Kostas Daniilidis
Petros Maragos
Nikos Paragios (Eds.)

LNCS 6314

Computer Vision – ECCV 2010

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

Springer

Lecture Notes in Computer Science 6314

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 IV

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 3-642-15560-X Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15560-4 Springer Berlin Heidelberg New York

Printed in Germany

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

Preface

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

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

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

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

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

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

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

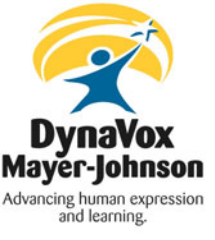


Table of Contents – Part IV

Spotlights and Posters W1

Scene and Object Recognition

Spotlights and Posters W2

Kernel Sparse Representation for Image Classification and Face Recognition

Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia

School of Computer Engineering, Nanyang Technological Univertiy, Singapore
{gaos0004,IvorTsang,asltchia}@ntu.edu.sg

Abstract. Recent research has shown the effectiveness of using sparse coding(Sc) to solve many computer vision problems. Motivated by the fact that kernel trick can capture the nonlinear similarity of features, which may reduce the feature quantization error and boost the sparse coding performance, we propose Kernel Sparse Representation(KSR). KSR is essentially the sparse coding technique in a high dimensional feature space mapped by implicit mapping function. We apply KSR to both image classification and face recognition. By incorporating KSR into Spatial Pyramid Matching(SPM), we propose KSRSPM for image classification. KSRSPM can further reduce the information loss in feature quantization step compared with Spatial Pyramid Matching using Sparse Coding(ScSPM). KSRSPM can be both regarded as the generalization of Efficient Match Kernel(EMK) and an extension of ScSPM. Compared with sparse coding, KSR can learn more discriminative sparse codes for face recognition. Extensive experimental results show that KSR outperforms sparse coding and EMK, and achieves state-of-the-art performance for image classification and face recognition on publicly available datasets.

1 Introduction

Sparse coding technique is attracting more and more researchers' attention in computer vision due to its state-of-the-art performance in many applications, such as image annotation [25], image restoration [20], image classification [28] *etc.* It aims at selecting the least possible basis from the large basis pool to linearly recover the given signal under a small reconstruction error constraint. Therefore, sparse coding can be easily applied to feature quantization in Bag-of-Word(BoW) model based image representation. Moreover, under the assumption that the face images to be tested can be reconstructed by the images from the same categories, sparse coding can also be used in face recognition [26].

BoW model [23] is widely used in computer vision [27,21] due to its concise representation and robustness to scale and rotation variance. Generally, it contains three modules: (i) Region selection and representation; (ii) Codebook generation and feature quantization; (iii) Frequency histogram based image representation. In these three modules, codebook generation and feature quantization are the most important portions for image presentation. The codebook is

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 1–14, 2010.

a collection of basic patterns used to reconstruct the local features. Each basic pattern is known as a visual word. Usually k-means is adopted to generate the codebook, and each local feature is quantized to its nearest visual word. However, such hard assignment method may cause severe information loss [3,6], especially for those features located at the boundary of several visual words. To minimize such errors, soft assignment [21,6] was introduced by assigning each feature to more than one visual words. However, the way of choosing parameters, including the weight assigned to the visual word and the number of visual words to be assigned, is not trivial to be determined.

Recently, Yang *et al.* [28] proposed the method of using sparse coding in the codebook generation and feature quantization module. Sparse coding can learn better codebook that further minimizes the quantization error than k-means. Meanwhile, the weights assigned to each visual word are learnt concurrently. By applying sparse coding to Spatial Pyramid Matching [13] (referred to as: ScSPM), their method achieves state-of-the-art performance in image classification.

Another application of sparse coding is face recognition. Face recognition is a classic problem in computer vision, and has a great potential in many real world application. It generally contains two stages. (i): Feature extraction; and (ii): Classifier construction and label prediction. Usually Nearest Neighbor (NN) [5] and Nearest Subspace(NS) [11] are used. However, NN predicts the label of the image to be tested by only using its nearest neighbor in the training data, therefore it can easily be affected by noise. NS approximates the test image by using all the images belonging to the same category, and assigns the image to the category which minimizes the reconstruction error. But NS may not work well for the case where classes are highly correlated to each other[26]. To overcome these problems, Wright *et al.* proposed a sparse coding based face recognition framework [26], which can automatically selects the images in the training set to approximate the test image. Their method is robust to occlusion, illumination and noise and achieves excellent performance.

Existing work based on sparse coding only seeks the sparse representation of the given signal in original signal space. Recall that kernel trick [22] maps the non-linear separable features into high dimensional feature space, in which features of the same type are easier grouped together and linear separable. In this case we may find the sparse representation for the signals more easily, and the reconstruction error may be reduced as well. Motivated by this, we propose Kernel Sparse Representation(KSR), which is the sparse coding in the mapped high dimensional feature space.

The contributions of this paper can be summarized as follows: (i): We propose the idea of kernel sparse representation, which is sparse coding in a high dimensional feature space. Experiments show that KSR greatly reduces the feature reconstruction error. (2): We propose KSRSPM for image classification. KSRSPM can be regarded as a generalized EMK, which can evaluate the similarity between local features accurately. Compared with EMK, our KSRSPM is more robust by using quantized feature other than the approximated high

dimensional feature. (3): We extend KSR to face recognition. KSR can achieve more discriminative sparse codes compared with sparse coding, which can boost the performance for face recognition.

The rest of this paper is organized as follows: In Section 2, we describe the details of KSR, including its objective function and its implementation. By incorporating KSR into SPM framework, we propose KSRSPM in Section 3. We also emphasize the relationship between our KSRSPM and EMK in details. Image classification performance on several public available datasets are also reported at the end of this section. In Section 4, we use KSR for face recognition. Results comparisons between sparse coding and KSR on Extended Yale B Face Dataset are listed in this section. Finally, we conclude our work in Section 5.

2 Kernel Sparse Representation and Implementation

2.1 Kernel Sparse Representation

For general sparse coding, it aims at finding the sparse representation under the given basis $U(U \in \mathbb{R}^{d\times k})$, while minimizing the reconstruction error. It equals to solving the following objective.

$$\begin{aligned} &\underset{U,v}{min}\|x - Uv\|^2 + \lambda\|v\|_1 \\ &subject \quad to: \quad \|u_m\|^2 \leq 1 \end{aligned} \tag{1}$$

where $U = [u_1, u_2, ..., u_k]$. The first term of Equation (1) is the reconstruction error, and the second term is used to control the sparsity of the sparse codes v. Empirically larger λ corresponds to sparser solution.

Suppose there exists a feature mapping function ϕ: $\mathcal{R}^d \rightarrow \mathcal{R}^K$, $(d < K)$. It maps the feature and basis to the high dimensional feature space: $x \rightarrow \phi(x)$, $U = [u_1, u_2, ..., u_k] \rightarrow \mathcal{U} = [\phi(u_1), \phi(u_2), \ldots, \phi(u_k)]$. We substitute the mapped features and basis to the formulation of sparse coding, and arrive at kernel sparse representation(KSR):

$$\underset{U,v}{min} \quad \|\phi(x) - \mathcal{U}v\|^2 + \lambda\|v\|_1 \tag{2}$$

where $\mathcal{U} = [\phi(u_1), \phi(u_2), \ldots, \phi(u_k)]$. In our work, we use Gaussian kernel due to its excellent performance in many work [22,2]: $\kappa(x_1, x_2) = exp(-\gamma\|x_1 - x_2\|^2)$. Note that $\phi(u_i)^T\phi(u_i) = \kappa(u_i, u_i) = exp(-\gamma\|u_i - u_i\|^2) = 1$, so we can remove the constraint on u_i. Kernel sparse representation seeks the sparse representation for a mapped feature under the mapped basis in the high dimensional space.

2.2 Implementation

The objective of Equation (2) is not convex. Following the work of [28,14], we optimize the sparse codes v and the codebook $\mathcal{U}$ alternatively.

Learning The Sparse Codes in New Feature Space. When the codebook U is fixed, the objective in Equation (2) can be rewritten as:

$$\begin{aligned} &\min_{v} \quad \|\phi(x) - \mathcal{U}v\|^2 + \lambda\|v\|_1 \\ =&\kappa(x,x) + v^T K_{UU} v - 2v^T K_U(x) + \lambda\|v\|_1 \\ =&L(v) + \lambda\|v\|_1 \end{aligned} \tag{3}$$

where $L(v) = 1 + v^T K_{UU} v - 2v^T K_U(x)$, K_{UU} is a $k * k$ matrix with $\{K_{UU}\}_{ij} = \kappa(u_i, u_j)$, and $K_U(x)$ is a $k * 1$ vector with $\{K_U(x)\}_i = \kappa(u_i, x)$. The objective is the same as that of sparse coding except for the definition of K_{UU} and $K_U(x)$. So we can easily extend the Feature-Sign Search Algorithm[14] to solve the sparse codes. As for the computational cost, they are the same except for the difference in calculating kernel matrix.

Learning Codebook. When v is fixed, we learn the codebook U. Due to the large amount of features, it is hard to use all the feature to learn the codebook. Following the work [28,2], we random sample some features to learn the codebook U, then use the learnt U to sparsely encode all the features. Suppose we randomly sample N features, then we rewrite the objective as follows (m, s, t are used to index the columns number of the codebook.):

$$\begin{aligned} f(U) &= \frac{1}{N}\sum_{i=1}^{N}[\|\phi(x_i) - \mathcal{U}v_i\|^2 + \lambda\|v_i\|_1] \\ &= \frac{1}{N}\sum_{i=1}^{N}[1 + \sum_{s=1}^{k}\sum_{t=1}^{k} v_{i,s}v_{i,t}\kappa(u_s,u_t) - 2\sum_{s=1}^{k} v_{i,s}\kappa(u_s,x_i) + \lambda\|v_i\|_1] \end{aligned} \tag{4}$$

Since U is in the kernel ($\kappa(u_i, .)$), it is very challenging to adopt the commonly used methods, for example, Stochastic Gradient Descent method [2] to find the optimal codebook. Instead we optimize each column of U alternatively. The derivative of $f(U)$ with respect to u_m is (u_m is the column to be updated):

$$\frac{\partial f}{\partial u_m} = \frac{-4\gamma}{N}\sum_{i=1}^{N}[\sum_{t=1}^{k} v_{i,m}v_{i,t}\kappa(u_m,u_t)(u_m - u_t) - v_{i,m}\kappa(u_m,x_i)(u_m - x_i)] \tag{5}$$

To find the optimal u_m, we set $\frac{\partial f}{\partial u_m} = 0$. However, it is not easy to solve the equation due to the terms with respect to $\kappa(u_m, .)$. As a compromise, we use the approximate solution to replace the exact solution. Similar to fixed point algorithm [12], in the n^{th} u_m updating iteration, we use the result of u_m in the $(n-1)^{th}$ updating iteration to compute the part in the kernel function. Denote the u_m in the n^{th} updating process as $u_{m,n}$, then the equation with respect to $u_{m,n}$ becomes:

$$\begin{aligned} \frac{\partial f}{\partial u_{m,n}} &\cong \frac{-4\gamma}{N}\sum_{i=1}^{N}[\sum_{t=1}^{k} v_{i,m}v_{i,t}\kappa(u_{m,n-1},u_t)(u_{m,n} - u_t) - v_{i,m}\kappa(u_{m,n-1},x_i)(u_{m,n} - x_i)] \\ &= 0 \end{aligned}$$

When all the remaining columns are fixed, it becomes a linear equation of $u_{m,n}$ and can be solved easily. Following the work [2], the codebook is initialized as the results of k-means.

3 Application I: Kernel Sparse Representation for Image Classification

In this Section, we apply kernel sparse representation in SPM framework, and propose the KSRSPM. On the one hand, KSRSPM is an extension of ScSPM [28] by replacing sparse coding with KSR. On the other hand, KSRSPM can be regarded as the generalization of Efficient Match Kernel(EMK) [2].

3.1 Sparse Coding for Codebook Generation

k-means clustering is usually used to generate the codebook in BoW model. In k-means, the whole local feature space $X = [x_1, x_2, \ldots, x_N]$ (where $x_i \in \mathbb{R}^{d\times 1}$) is split into k clusterings $S = [S_1, S_2, \ldots, S_k]$. Denote the corresponding clustering centers as $U = [u_1, u_2, \ldots, u_k] \in \mathbb{R}^{d\times k}$. In hard assignment, each feature is only assigned to its nearest cluster center, and the weight the feature contributing to that center is 1. The objective of k-means can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{U,S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - u_i\|^2 = \min_{U,V} \sum_{i=1}^{N} \|x_i - U v_i\|^2 \\ &subject \;\; to: \;\; Card(v_i) = 1, |v_i| = 1, v_i \succeq 0, \forall i. \end{aligned} \tag{6}$$

Here V is a clustering indices, $V = [v_1, v_2, \ldots, v_N]$ (where $v_i \in \mathbb{R}^{k\times 1}$). Each column of V indicates which visual word the local feature should be assigned to. To reduce the information loss in feature quantization, the constraint on v_m is relaxed. Meanwhile, to avoid each feature being assigned to too many clusters, the sparse constraint is imposed on v_m. Then, we arrive at the optimization problem of sparse coding:

$$\begin{aligned} &\min_{U,V} \sum_{i=1}^{N} \|x_i - U v_i\|^2 + \lambda \|v_i\|_1 \\ &subject \;\; to: \;\; |u_j| \leq 1, \forall j = 1, \ldots, k. \end{aligned} \tag{7}$$

3.2 Maximum Feature Pooling and Spatial Pyramid Matching Based Image Representation

Following the work of [28,4], we use maximum pooling method to represent the images. Maximum pooling uses the largest responses to each basic pattern to represent the region. More specifically, suppose one image region has D local features, and the codebook size is k. After maximum pooling, each image will be

represented by a k dimensional vector y, and the l^{th} entry is the largest response to the l^{th} basis vector of all the sparse codes in the selected region(v_D is the sparse codes of the D^{th} feature in this local region, and v_{Dl} is the l^{th} entry of v_D):

$$y_l = max\{|v_{1l}|, |v_{2l}|, \ldots, |v_{Dl}|\} \tag{8}$$

SPM technique is also used to preserve the spatial information. The whole image is divided into increasing finer regions, and maximum pooling is used in each subregion.

3.3 KSRSPM – An Generalization of Efficient Matching Kernel

Besides interpreted as an extension of ScSPM [28], KSRSPM can also be interpreted as a generalization of Efficient Matching Kernel (EMK) [2]. Let $X = [x_1, x_2, \ldots, x_p]$ be a set of local features in one image, and $V(x) = [v_1(x), v_2(x), \ldots, v_p(x)]$ are the corresponding clustering index vector in Equation (6). In BoW model, each image is presented by a normalized histogram $\bar{v}(X) = \frac{1}{|X|}\sum_{x\in X} v(x)$, which characterizes its visual word distribution. By using linear classifier, the resulting kernel function is:

$$K_B(X,Y) = \frac{1}{|X||Y|}\sum_{x\in X}\sum_{y\in Y} v(x)^T v(y) = \frac{1}{|X||Y|}\sum_{x\in X}\sum_{y\in Y}\delta(x,y) \tag{9}$$

where

$$\delta(x,y) = \begin{cases} 1, & v(x) = v(y) \\ 0, & otherwise \end{cases} \tag{10}$$

$\delta(x, y)$ is positive definite kernel, which is used to measure the similarity between two local features. However, such hard assignment based local feature similarity measuring method increases the information loss and reduces classification accuracy. Thus a continuous kernel is introduced to more accurately measure the similarity between local feature x and y:

$$K_S(X,Y) = \frac{1}{|X||Y|}\sum_{x\in X}\sum_{y\in Y} k(x,y) \tag{11}$$

Here $k(x, y)$ is positive definite kernel, which is referred to as local kernel. This is related to the normalized sum match kernel [19,9].

Due to the large amount of local features, directly using local kernel is both storage and computation prohibitive for image classification. To decrease the computation cost, Efficient Match Kernel(EMK) is introduced. Under the definition of finite dimensional kernel function [2], $k(x, y) = \phi(x)^T\phi(y)$, we can approximate $\phi(x)$ by using low dimensional features $\bar{v}_x$ in the space spanned by k basis vectors $H = [\phi(u_1), \phi(u_2), \ldots, \phi(u_k)]$:

$$\min_{H, v_x} \quad \|\phi(x) - Hv_x\|^2 \tag{12}$$

In this way, each image can be represented by $\bar{v}(X)_{new} = \frac{1}{|X|}H\sum_{x\in X} v_x$ beforehand. As a consequence, the computation speed can be accelerated.

EMK maps the local feature to high dimensional feature space to evaluate the similarity between local features more accurately, and uses the approximated feature Hv_x to construct the linear classifier for image classification. It can be summarized as two stages: (i): $x \xrightarrow{\phi} \phi(x)$: Map the feature to new feature space; (ii): $\phi(x) \xrightarrow{H} \bar{v}_x$: Reconstruct $\phi(x)$ by using the basis H.

Note that directly using original feature for image classification may cause overfitting [3]. To avoid this, and following the BoW model, we use v_x for image classification. We hope each $\phi(x)$ is only assigned several clusterings, so we add the sparse constraint in the objective of EMK:

$$\min_{H,v_x} \quad \|\phi(x) - Hv_x\|^2 + \lambda\|v_x\|_1 \tag{13}$$

This is the same as the objective of our kernel sparse representation. So EMK can be regarded as the special case of our KSRSPM at $\lambda = 0$. Compared with EMK, our KSRSPM uses the quantized feature indices for image classification, so it is more robust to the noise. What's more, by using maximum pooling, the robustness to intra-class and noise of our KSRSPM can be further strengthened.

3.4 Experiments

Parameters Setting. SIFT [16] is widely used in image recognition due to its excellent performance. For a fair comparison and to be consistent with previous work [28,13,2], we use the SIFT features under the same feature extraction setting. Specifically, we use dense grid sampling strategy and fix the step size and patch size to 8 and 16 respectively. We also resize the maximum side(width/length) of each image to 300 pixels[1]. After obtaining the SIFT, we use $\ell 2$-norm to normalize the feature length to 1. For the codebook size, we set $k = 1024$ in k-means, and randomly select $(5.0 \sim 8.0) * 10^4$ features to generate codebook for each data set. Following the work [28], we set $\lambda = 0.30$ for all the datasets. As for the parameter γ in the Gaussian kernel, we set γ to $\frac{1}{64}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}$ on Scene 15, UIUC-Sports, Caltech 256 and Corel 10 respectively. For SPM, we use top 3 layers and the weight for each layer is the same. We use one-vs-all linear SVM due to its advantage in speed [28] and excellent performance in maximum feature pooling based image classification. All the results for each dataset are based on six independent experiments, and the training images are selected randomly.

Scene 15 Dataset. Scene 15 [13] dataset is usually used for scene classification. It contains 4485 images, which are divided into 15 categories. Each category contains about 200 to 400 images. The image content is diverse, containing *suburb, coast, forest, highway, inside city, mountain, open country, street, tall building,*

[1] For UIUC-Sport dataset, we resize the maximum side to 400 due to the high resolution of original image.

office, bedroom, industrial, kitchen, living room and *store*. For fair comparison, we follow the same experimental setting [28,13]: randomly select 100 images each category as training data and use the remaining images as test data. The results are listed in Table 1.

Table 1. Performance Comparison on Scene 15 Dataset(%)

Method	Average Classification Rate
KSPM[13]	81.40±0.50
EMK[2]	77.89±0.85
ScSPM[28]	80.28±0.93
KSRSPM	**83.68±0.61**

Caltech 256. Caltech 256[2] is a very challenging dataset in both image content and dataset scale. First of all, compared with Caltech 101, the objects in Caltech 256 contains larger intra-class variance, and the object locations are no longer in the center of the image. Second, Caltech 256 contains 29780 images, which are divided into 256 categories. More categories will inevitably increase the inter-class similarity, and increase the performance degradation. We evaluate the method under four different settings: selecting 15, 30, 45, 60 per category as training data respectively, and use the rest as test data. The results are listed in Table 2.

Table 2. Performance Comparison on Caltech 256 dataset(%) (KC: Kernel codebook;)

Trn No.	KSPM[8]	KC[6]	EMK[2]	ScSPM[28]	**KSRSPM**
15	NA	NA	23.2±0.6	27.73±0.51	**29.77±0.14**
30	34.10	27.17±0.46	30.5±0.4	34.02±0.35	**35.67±0.10**
45	NA	NA	34.4±0.4	37.46±0.55	**38.61±0.19**
60	NA	NA	37.6±0.5	40.14±0.91	**40.30±0.22**

UIUC-Sport Dataset. UIUC-Sport [15] contains images collected from 8 kind of different sports: *badminton, bocce,croquet, polo, rock climbing, rowing, sailing* and *snow boarding*. There are 1792 images in all, and the number of images ranges from 137 to 250 per category. Following the work of Wu *et al.* [27], we randomly select 70 images from each category as training data, and randomly select another 60 images from each category as test data. The results are listed in Table 3.

Table 3. Performance Comparison on UIUC-Sport Dataset(%)

Method	Average Classification Rate
HIK+ocSVM[27]	83.54±1.13
EMK[2]	74.56±1.32
ScSPM[28]	82.74±1.46
KSRSPM	**84.92±0.78**

[2] www.vision.caltech.edu/Image_Datasets/Caltech256/

Table 4. Performance Comparison on Corel10 Dataset(%) (SMK:Spatial Markov Model)

Method	Average Classification Rate
SMK [17]	77.9
EMK [2]	79.90±1.73
ScSPM [28]	86.2±1.01
KSRSPM	**89.43±1.27**

Corel10 Dataset. Corel10 [18] contains 10 categories: *skiing, beach, buildings, tigers, owls, elephants, flowers, horses, mountains* and *food*. Each category contains 100 images. Following the work of Lu *et al.* [18], we randomly select 50 images as training data and use the rest as test data. The results are listed in Table 4.

Results Analysis. From Table 1-4, we can see that on Scene, UIUC-Sports, Corel10, KSRSPM outperforms EMK around $(5.7 \sim 10.4)\%$, and outperforms ScSPM around $(2.2 \sim 3.4)\%$. For Caltech 256, due to too many classes, the improvements are not very substantial, but still higher than EMK and ScSPM. We also list the confusion matrices of Scene, UIUC-Sports and Corel10 datasets in Figure 1 and Figure 2. The entry located in i^{th} row, j^{th} column in confusion matrix represents the percentage of class i being misclassified to class j. From the confusion matrices, we can see that some classes are easily be misclassified to some others.

Feature Quantization Error. Define Average Quantization Error (AverQE) as: $AverQE = \frac{1}{N}\sum_{i=1}^{N} \|\phi(x_i) - \mathcal{U}v_i\|_F^2$. It can be used to evaluate the information loss in the feature quantization process. To retain more information, we hope the feature quantization error can be reduced. We compute the AverQE of our kernel sparse representation (KSR) and Sparse coding (Sc) on all the features used for codebook generation, and list them in Table 5. From results we can see that kernel sparse representation can greatly decrease the feature quantization error.

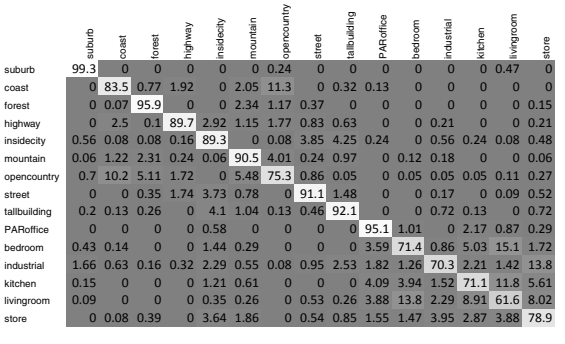


Fig. 1. Confusion Matrix on Scene 15 dataset(%)

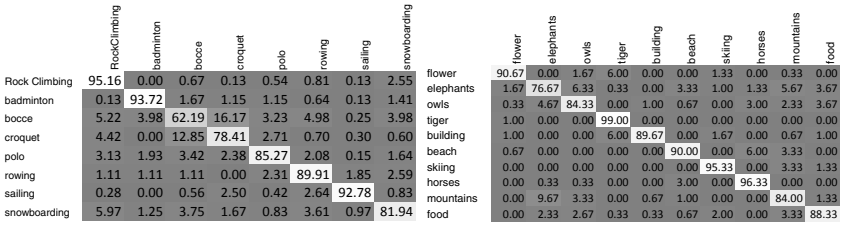


Fig. 2. Confusion Matrices on UIUC-Sports and Corel10(%)

Table 5. Average Feature Quantization Error on Different datasets

	Scene	Caltech 256	Sport	Corel
Sc	0.8681	0.9164	0.8864	0.9295
KSR	**9.63E-02**	**5.72E-02**	**9.40E-02**	**4.13E-02**

This may be the reason that our KSRSPM outperforms ScSPM. The results also agree with our assumption that sparse coding in high dimensional space can reduce the feature quantization error.

4 Application II: Kernel Sparse Representation for Face Recognition

4.1 Sparse Coding for Face Recognition

For face recognition, "If sufficient training samples are available from each class, it would be possible to represent the test samples as a linear combination of those training samples from the same class [26]".

Suppose there are N classes in all, and the training instances for class i are $A_i = [a_{i,1}, \ldots, a_{i,n_i}] \in \mathbb{R}^{d \times n_i}$, in which each column corresponds to one instance. Let $A = [A_1, \ldots, A_N] \in \mathbb{R}^{d \times \sum_{i=1}^{N} n_i}$ be the training set, and $y \in \mathbb{R}^{d \times 1}$ be the test sample. When noise e exists, the problem for face recognition [26] can be formulated as follows:

$$min \quad \|x_0\|_1 \qquad s.t. \qquad y = Ax^T + e = [A\ I][x^T\ e^T]^T = A_0 x_0 \tag{14}$$

sparse coding based image recognition aims at selecting only a few images from all the training instances to reconstruct the images to be tested. Let $\alpha_i = [\alpha_{i,1}, \ldots, \alpha_{i,n_i}](1 \leq i \leq N)$ be the coefficients corresponds to A_i in x_0. The reconstruction error by using the instances from class i can be computed as: $r_i(y) = \|y - A_i\alpha_i\|_2$. Then the test image is assigned to the category that minimizes the reconstruction error: $identity(y) = arg\ min_i\ \{r_1(y), \ldots, r_N(y)\}$.

4.2 Kernel Sparse Representation for Face Recognition

Kernel method can make the features belonging to the same category closer to each other [22]. Thus we apply kernel sparse representation in face recognition.

Firstly, the $\ell 1$ norm on reconstruction error is replaced by using $\ell 2$ norm(We assume that the noise may not be sparsely reconstructed by using the training samples). By mapping features to a high dimensional space: $y \rightarrow \phi(y)$, $A = [a_{1,1}, \ldots, a_{N,n_N}] \rightarrow \mathcal{A} = [\phi(a_{1,1}), \ldots, \phi(a_{N,n_N})]$, we obtain the objective of kernel sparse representation for face recognition:

$$min \quad \lambda\|x\|_1 + \|\phi(y) - \mathcal{A}x\|_2^2 \tag{15}$$

In which the parameter λ is used to balance the weight between the sparsity and the reconstruction error. Following the work of John Wright *et al.*, the test image is assigned to the category which minimizes the reconstruction error in the high dimensional feature space.

4.3 Evaluation on Extended Yale B Database

We evaluate our method on Extended Yale B Database [7], which contains 38 categories, 2414 frontal-face images. The cropped image size is 192×168. Following the work [26], we randomly select a half as training images in each category, and use the rest as test. The following five features are used for evaluation: RandomFace [26], LaplacianFace [10], EigenFace [24],FisherFace [1] and Downsample [26], and each feature is normalized to unit length by using $\ell 2$ norm. Gaussian kernel is used in our experiments: $\kappa(x_1, x_2) = exp(-\gamma\|x_1 - x_2\|^2)$. For Eigenfaces, Laplacianfaces, Downsample and Fisherfaces, we set $\gamma = 1/d$ where d is the feature dimension. For Randomfaces, $\gamma = 1/32d$.

The Effect of λ. We firstly evaluate λ by using 56D Downsample Feature. We list the results based on different λ in Table 6. When $\lambda \neq 0$, as λ decreases, the performance increases, and the proportion of non-zero elements in coefficients increases. But computational time also increases. When $\lambda = 0$, it happens to be the objective of Efficient Match Kernel, but the performance is not good as that in the case of $\lambda \neq 0$. This can show the effectiveness of the sparse term.

Result Comparison. Considering both the computational cost and the accuracy in Table 6, we set $\lambda = 10^{-5}$. The experimental results are listed in Table 7. All the results are based on 10 times independent experiments. Experimental results show that kernel sparse representation can outperform sparse coding in face recognition.

Table 6. The Effect of Sparsity Parameter: 56D Downsample Feature (Here sparsity is percentage of non-zeros elements in sparse codes)

λ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	0
sparsity(%)	0.58	0.75	0.88	2.13	4.66	8.35	16.69	-
reconstruction error	0.2399	0.1763	0.1651	0.1113	0.0893	0.0671	0.0462	-
time(sec)	0.0270	0.0280	0.0299	0.0477	0.2445	0.9926	6.2990	-
accuracy(%)	76.92	84.12	85.19	90.32	91.65	93.30	93.47	84.37

Table 7. Performance of Sparse Coding for Face Recognition(%)

	Feature Dimension	30	56	120	504
Eigen	Sc [26]	86.5	91.63	93.95	96.77
	KSR	**89.01**	**94.42**	**97.49**	**99.16**
Laplacian	Sc [26]	87.49	91.72	93.95	96.52
	KSR	**88.86**	**94.24**	**97.11**	**98.12**
Random	Sc [26]	82.6	91.47	95.53	98.09
	KSR	**85.46**	**92.36**	**96.14**	**98.37**
Downsample	Sc [26]	74.57	86.16	92.13	97.1
	KSR	**83.57**	**91.65**	**95.31**	**97.8**
Fisher	Sc [26]	86.91	NA	NA	NA
	KSR	**88.93**	NA	NA	NA

To further illustrate the performance of KSR, we calculate the similarity between the sparse codes of KSR and Sc in three classes(each classes contains 32 images). We list the results in Figure 3, in which the entry in (i, j) is the sparse codes similarity (normalized correlation) between image i and j. We know that a good sparse coding method can make the sparse codes belonging to same class more similar, therefore, the sparse codes similarity should be block-wise. From Figure 3 we can see that our KSR can get more discriminative sparse codes than sparse coding, which facilitates the better performance of the image recognition.

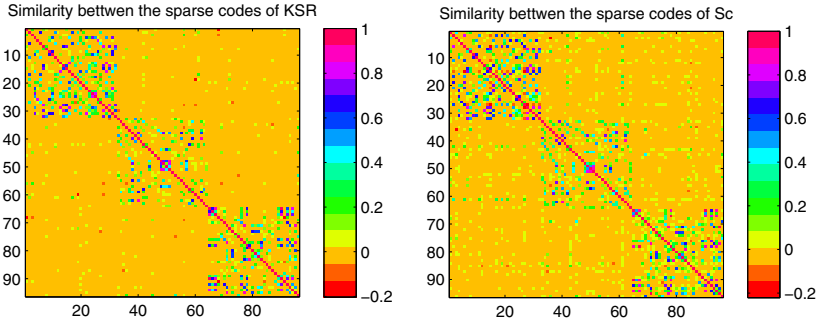


Fig. 3. Similarity between the sparse codes of KSR and Sc

5 Conclusion

In this paper, we propose a new technique: Kernel Sparse Representation, which is the sparse coding technique in a high dimensional feature space mapped by implicit feature mapping feature. We apply KSR to image classification and face recognition. For image classification, our proposed KSRSPM can both be regarded as an extension of ScSPM and an generalization of EMK. For face recognition, KSR can learn more discriminative sparse codes for face category

identification. Experimental results on several publicly available datasets show that our KSR outperforms both ScSPM and EMK, and achieves state-of-the-art performance.

References

1. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. TPAMI 19(7), 711–720 (1997)
2. Bo, L., Sminchisescu, C.: Efficient match kernels between sets of features for visual recognition. In: NIPS (2009)
3. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
4. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learnning mid-level features for recognition (2010)
5. Duda, R.O., Hart, P.E., Stock, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, Chichester (2001)
6. van Gemert, J.C., Geusebroek, J.M., Veenman, C.J., Smeulders, A.W.M.: Kernel codebooks for scene categorization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 696–709. Springer, Heidelberg (2008)
7. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. TPAMI 23(6), 643–660 (2001)
8. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. In: Technical Report (2007)
9. Haussler, D.: Convolution kernels on discrete structure. In: Technical Report (1999)
10. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.: Face recognition using laplacianfaces. TPAMI 27(3), 328–340 (2005)
11. Ho, J., Yang, M.H., Lim, J., Lee, K.C., Kriegman, D.J.: Clustering appearances of objects under varying illumination conditions. In: CVPR (2003)
12. Hyvärinen, A.: The fixed-point algorithm and maximum likelihood estimation for independent component analysis. Neural Process. Lett. 10(1) (1999)
13. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
14. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: NIPS, pp. 801–808 (2006)
15. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: ICCV (2007)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
17. Lu, Z., Ip, H.H.: Image categorization by learning with context and consistency. In: CVPR (2009)
18. Lu, Z., Ip, H.H.: Image categorization with spatial mismatch kernels. In: CVPR (2009)
19. Lyu, S.: Mercer kernels for object recognition with local features. In: CVPR, pp. 223–229 (2005)
20. Marial, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: ICCV (2009)
21. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)

22. Schölkopf, B., Smola, A.J., Müller, K.R.: Kernel principal component analysis. In: International Conference on Artificial Neural Networks, pp. 583–588 (1997)
23. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)
24. Turk, M., Pentland, A.: Eigenfaces for recognition. In: CVPR (1991)
25. Wang, C., Yan, S., Zhang, L., Zhang, H.J.: Multi-label sparse coding for automatic image annotation. In: CVPR (2009)
26. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. TPAMI 31(2), 210–227 (2009)
27. Wu, J., Rehg, J.M.: Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In: ICCV (2003)
28. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)

Every Picture Tells a Story: Generating Sentences from Images

Ali Farhadi[1], Mohsen Hejrati[2] , Mohammad Amin Sadeghi[2], Peter Young[1], Cyrus Rashtchian[1], Julia Hockenmaier[1], David Forsyth[1]

[1] Computer Science Department
University of Illinois at Urbana-Champaign
{afarhad2,pyoung2,crashtc2,juliahmr,daf}@illinois.edu
[2] Computer Vision Group, School of Mathematics
Institute for studies in theoretical Physics and Mathematics(IPM)
{m.a.sadeghi,mhejrati}@gmail.com

Abstract. Humans can prepare concise descriptions of pictures, focusing on what they find important. We demonstrate that automatic methods can do so too. We describe a system that can compute a score linking an image to a sentence. This score can be used to attach a descriptive sentence to a given image, or to obtain images that illustrate a given sentence. The score is obtained by comparing an estimate of meaning obtained from the image to one obtained from the sentence. Each estimate of meaning comes from a discriminative procedure that is learned using data. We evaluate on a novel dataset consisting of human-annotated images. While our underlying estimate of meaning is impoverished, it is sufficient to produce very good quantitative results, evaluated with a novel score that can account for synecdoche.

1 Introduction

For most pictures, humans can prepare a concise description in the form of a sentence relatively easily. Such descriptions might identify the most interesting objects, what they are doing, and where this is happening. These descriptions are rich, because they are in sentence form. They are accurate, with good agreement between annotators. They are concise: much is omitted, because humans tend not to mention objects or events that they judge to be less significant. Finally, they are consistent: in our data, annotators tend to agree on what is mentioned. Barnard *et al.* name two applications for methods that link text and images: **Illustration**, where one finds pictures suggested by text (perhaps to suggest illustrations from a collection); and **annotation**, where one finds text annotations for images (perhaps to allow keyword search to find more images) [1].

This paper investigates methods to generate short descriptive sentences from images. Our contributions include: We introduce a dataset to study this problem (section 3.1). We introduce a novel representation intermediate between images and sentences (section 2.1). We describe a novel, discriminative approach that produces very good results at sentence annotation (section 2.4). For illustration,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 15–29, 2010.

out of vocabulary words pose serious difficulties, and we show methods to use distributional semantics to cope with these issues (section 3.4). Evaluating sentence generation is very difficult, because sentences are fluid, and quite different sentences can describe the same phenomena. Worse, synecdoche (for example, substituting "animal" for "cat" or "bicycle" for "vehicle") and the general richness of vocabulary means that many different words can quite legitimately be used to describe the same picture. In section 3, we describe a quantitative evaluation of sentence generation at a useful scale.

Linking individual words to images has a rich history and space allows only a mention of the most relevant papers. A natural strategy is to try and predict words from image regions. The first image annotation system is due to Mori *et al.* [2]; Duygulu *et al.* continued this tradition using models from machine translation [3]. Since then, a wide range of models has been deployed (reviews in [4,5]); the current best performer is a form of nearest neighbours matching [6]. The most recent methods perform fairly well, but still find difficulty **placing** annotations on the correct regions.

Sentences are richer than lists of words, because they describe activities, properties of objects, and relations between entities (among other things). Such relations are revealing: Gupta and Davis show that respecting likely spatial relations between objects markedly improves the accuracy of both annotation and placing [7]. Li and Fei-Fei show that event recognition is improved by explicit inference on a generative model representing the scene in which the event occurs and also the objects in the image [8]. Using a different generative model, Li and Fei-Fei demonstrate that relations improve object labels, scene labels and segmentation [9]. Gupta and Davis show that respecting relations between objects and actions improve recognition of each [10,11]. Yao and Fei-Fei use the fact that objects and human poses are coupled and show that recognizing one helps the recognition of the other [12]. Relations between words in annotating sentences can reveal image structure. Berg *et al.* show that word features suggest which names in a caption are depicted in the attached picture, and that this improves the accuracy of links between names and faces [13]. Mensink and Verbeek show that complex co-occurrence relations between people improve face labelling, too [14]. Luo, Caputo and Ferrari [15] show benefits of associating faces and poses to names and verbs in predicting "who's doing what" in news articles. Coyne and Sproat describe an auto-illustration system that gives naive users a method to produce rendered images from free text descriptions (Wordseye; [16];`http://www.wordseye.com`).

There are few attempts to generate sentences from visual data. Gupta *et al.* generate sentences narrating a sports event in video using a compositional model based around AND-OR graphs [17]. The relatively stylised structure of the events helps both in sentence generation and in evaluation, because it is straightforward to tell which sentence is right. Yao *et al.* show some examples of both temporal narrative sentences (i.e. this happened, then that) and scene description sentences generated from visual data, but there is no evaluation [18].

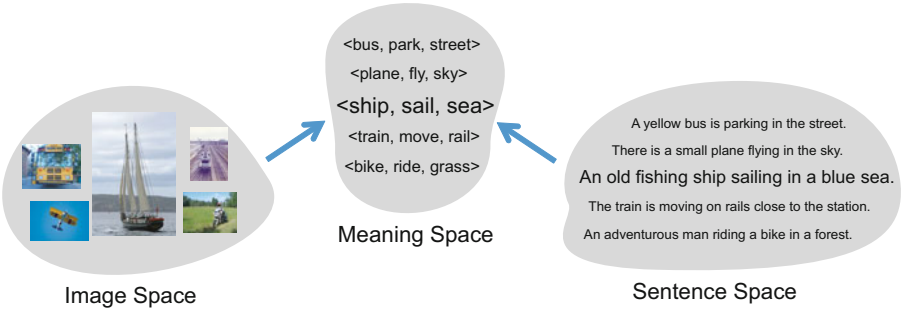


Fig. 1. There is an intermediate space of meaning which has different projections to the space of images and sentences. Once we learn the projections we can generate sentences for images and find images best described by a given sentence.

These methods generate a direct representation of what is happening in a scene, and then decode it into a sentence.

An alternative, which we espouse, is to build a scoring procedure that evaluates the similarity between a sentence and an image. This approach is attractive, because it is symmetric: given an image (resp. sentence), one can search for the best sentence (resp. image) in a large set. This means that one can do both illustration and annotation with one method. Another attraction is the method does not need a strong syntactic model, which is represented by the prior on sentences. Our scoring procedure is built around an intermediate representation, which we call the **meaning** of the image (resp. sentence). In effect, image and sentence are each mapped to this intermediate space, and the results are compared; similar meanings result in a high score. The advantage of doing so is that each of these maps can be adjusted discriminatively. While the meaning space could be abstract, in our implementation we use a direct representation of simple sentences as a meaning space. This allows us to exploit distributional semantics ideas to deal with out of vocabulary words. For example, we have no detector for "cattle"; but we can link sentences containing this word to images, because distributional semantics tells us that a "cattle" is similar to "sheep" and "cow", etc. (Figure 6)

2 Approach

Our model assumes that there is a space of *Meanings* that comes between the space of *Sentences* and the space of *Images*. We evaluate the similarity between a sentence and an image by (a) mapping each to the meaning space then (b) comparing the results. Figure 1 depicts the intermediate space of meanings. We will learn the mapping from images (resp. sentences) to meaning discriminatively from pairs of images (resp. sentences) and assigned meaning representations.

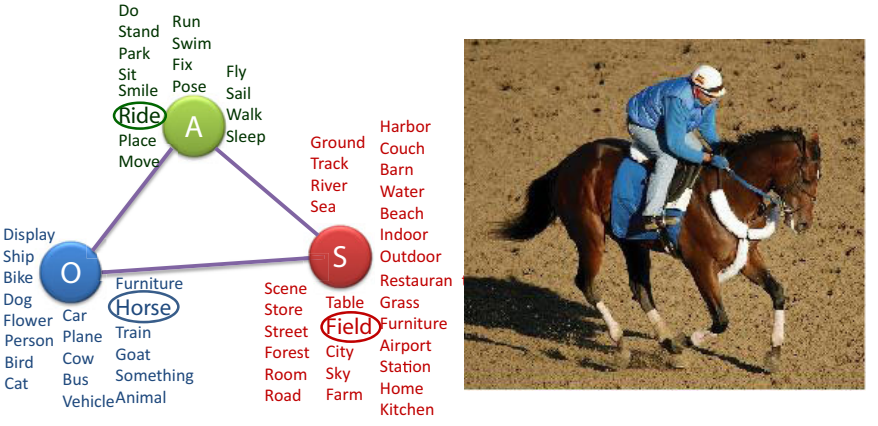


Fig. 2. We represent the space of the meanings by triplets of $\langle object, action, scene \rangle$. This is an MRF. Node potentials are computed by linear combination of scores from several detectors and classifiers. Edge potentials are estimated by frequencies. We have a reasonably sized state space for each of the nodes. The possible values for each nodes are written on the image. "O" stands for the node for the object, "A" for the action, and "S" for scene. Learning involves setting the weights on the node and edge potentials and inference is finding the best triplets given the potentials.

2.1 Mapping Image to Meaning

Our current representation of meaning is a triplet of $\langle object, action, scene \rangle$. This triplet provides a holistic idea about what the image (resp. sentence) is about and what is most important. For the image, this is the part that people would talk about first; for the sentence, this is the structure that should be preserved in the tightest summary. For each slot in the triplet, there is a discrete set of possible values. Choosing among them will result in a triplet. The mapping from images to meaning is reduced to learning to predict triplet for images. The problem of predicting a triplet from an image involves solving a (small) multi-label Markov random field. Each slot in the meaning representation can take a value from a set of discrete values. Figure 2 depicts the representation of the meaning space and the corresponding MRF. There is a node for objects which can take a value from a possible set of 23 nouns, a node for actions with 16 different values, and a node to scenes that can select each of 29 different values. The edges correspond to the binary relationships between nodes. Having provided the potentials of the MRF, we use a greedy method to do inference. Inference involves finding the best selection of the discrete sets of values given the unary and binary potentials.

We learn to predict triplets for images discriminatively. This requires having a dataset of images labeled with their meaning triplets. The potentials are computed as linear combinations of feature functions. This casts the problem of learning as searching for the best set of weights on the linear combination of feature functions so that the ground truth triplets score higher than any other triplet. Inference involves finding $argmax_y w^T \Phi(x, y)$ where Φ is the potential function, y is the triplet label, and w are the learned weights.

2.2 Image Potentials

We need informative features to drive the mapping from the image space to the meaning space.

Node Potentials. To provide information about the nodes on the MRF we first need to construct image features. Our *image features* consist of:

Felzenszwalb *et al.* detector responses. We use Felzenszwalb detectors [19] to predict confidence scores on all the images. We set the threshold such that all of the classes get predicted, at least once in each image. We then consider the max confidence of the detections for each category, the location of the center of the detected bounding box, the aspect ratio of the bounding box, and it's scale.

Hoiem *et al.* classification responses. We use the classification scores of Hoiem et. al [20] for the PASCAL classification tasks. These classifiers are based on geometry, HOG features, and detection responses.

Gist-based scene classification responses. We encode global information of images using gist [21]. Our features for scenes are the confidences of our Adaboost style classifier for scenes.

First we build node features by fitting a discriminative classifier (a linear SVM) to predict each of the nodes independently on the image features. Although the classifiers are being learned independently, they are well aware of other objects and scene information. We call these estimates *node features*. This is a number-of-nodes-dimensional vector and each element in this vector provides a score for a node given the image. This can be a node potential for object, action, and scene nodes. We expect similar images to have similar meanings, and so we obtain a set of features by matching our test image to training images. We combine these features into various other node potentials as below:

- by matching image features, we obtain the k-nearest neighbours in the training set to the test image, then compute the average of the node features over those neighbours, *computed from the image side*. By doing so, we have a representation of what the node features are for similar images.
- by matching image features, we obtain the k-nearest neighbours in the training set to the test image, then compute the average of the node features over those neighbours, *computed from the sentence side*. By doing so, we have a representation of what the sentence representation does for images that look like our image.
- by matching those node features derived from classifiers and detectors (above), we obtain the k-nearest neighbours in the training set to the test image, then compute the average of the node features over those neighbours, *computed from the image side*. By doing so, we have a representation of what the node features are for images that produce similar classifier and detector outputs.

– by matching those node features derived from classifiers and detectors (above), we obtain the k-nearest neighbours in the training set to the test image, then compute the average of the node features over those neighbours, *computed from the sentence side*. By doing so, we have a representation of what the sentence representation does for images that produce similar classifier and detector outputs.

Edge Potentials. Introducing a parameter for each edge results in unmanageable number of parameters. In addition, estimates of the parameters for the majority of edges would be noisy. There are serious smoothing issues. We adopt an approach similar to Good Turing smoothing methods to a) control the number of parameters b) do smoothing. We have multiple estimates for the edges potentials which can provide more accurate estimates if used together. We form the linear combinations of these potentials. Therefore, in learning we are interested in finding weights of the linear combination of the initial estimates so that the final linearly combined potentials provide values on the MRF so that the ground truth triplet is the highest scored triplet for all examples. This way we limit the number of parameters to the number of initial estimates.

We have four different estimates for edges. Our final score on the edges take the form of a linear combination of these estimates. Our four estimates for edges from node A to node B are:

– The normalized frequency of the word A in our corpus, $f(A)$.
– The normalized frequency of the word B in our corpus, $f(B)$.
– The normalized frequency of (A and B) at the same time, $f(A, b)$.
– $\frac{f(A,B)}{f(A)f(B)}$.

2.3 Sentence Potentials

We need a representation of the sentences. We represent a sentence by computing the similarity between the sentence and our triplets. For that we need to have a notion of similarity for objects, scenes and actions in text.

We used the Curran & Clark parser [22] to generate a dependency parse for each sentence. We extracted the subject, direct object, and any nmod dependencis involving a noun and a verb. These dependencies were used to generate the (object, action) pairs for the sentences. In order to extract the scene information from the sentences, we extracted the head nouns of the prepositional phrases (except for the prepositions "of" and "with"), and the head nouns of the phrase "X in the background".

Lin Similarity Measure for Objects and Scenes. We use the Lin similarity measure [23] to determine the semantic distance between two words. The Lin similarity measure uses WordNet synsets as the possible meanings of each words. The noun synsets are arranged in a heirarchy based on hypernym (is-a) and hyponym (instance-of) relations. Each synset is defined as having an information content based on how frequently the synset or a hyponym of the synset occurs in

a corpus (in the case, SemCor). The similarity of two synsets is defined as twice the information content of the least common ancestor of the synsets divided by the sum of the information content of the two synsets. Similar synsets will have a LCA that covers the two synsets, and very little else. When we compared two nouns, we considered all pairs of a filtered list of synsets for each noun, and used the most similar synsets. We filtered the list of synsets for each noun by limiting it to the first four synsets that were at least 10% as frequent as the most common synset of that noun. We also required the synsets to be physical entities.

Action Co-occurrence Score. We generated a second image caption data set consisting of roughly 8,000 images pulled from six Flickr groups. For all pairs of verbs, we used the likelihood ratio to determine if the two verbs co-occurring in the different captions of the same image was significant. We then used the likelihood ratio as the similarity score for the positively correlated verb pairs, and the negative of the likelihood ratio as the similarity score for the negatively correlated verb pairs. Typically, we found that this procedure discovered verbs that were either describing the same action or describing two actions that commonly co-occurred.

Node Potentials. We now can provide a similarity measure between sentences and objects, actions, and scenes using scores explained above. Below we explain our estimates of sentence node potentials.

- First we compute the similarity of each object, scene, and action extracted from each sentence. This gives us the the first estimates for the potentials over the nodes. We call this the *sentence node feature*.
- For each sentence, we also compute the average of sentence node features for other four sentences describing the same images in the train set.
- We compute the average of k nearest neighbors in the sentence node features space for a given sentence. We consider this as our third estimate for nodes.
- We also compute the average of the image node features for images corresponding to the nearest neighbors in the item above.
- The average of the sentence node features of reference sentences for the nearest neighbors in the item 3 is considered as our fifth estimate for nodes.
- We also include the sentence node feature for the reference sentence.

Edge Potentials. The edge estimates for sentences are identical to to edge estimates for the images explained in previous section.

2.4 Learning

There are two mappings that need to be learned. The map from the image space to the meaning space uses the image potentials and the map from the sentence space to the meaning space uses the sentence potentials. Learning the mapping from images to meaning involves finding the weights on the linear combinations of our image potentials on nodes and edges so that the ground truth triplets score

highest among all other triplets for all examples. This is a structure learning problem [24] which takes the form of

$$\min_{w} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i \in examples} \xi_i \quad (1)$$

subject to

$$w\Phi(x_i, y_i) + \xi_i \geq \max_{y \in \text{meaning space}} w\Phi(x_i, y) + L(y_i, y)\ \forall i \in \text{examples}$$

$$\xi_i \geq 0\ \forall i \in \text{examples}$$

where λ is the tradeoff factor between the regularization and slack variables ξ, Φ is our feature functions, x_i corresponds to our i^{th} image, and y_i is our structured label for the i^{th} image. We use the stochastic subgradient descent method [25] to solve this minimization.

3 Evaluation

We emphasize quantitative evaluation in our work. Our vocabulary of meaning is significantly larger than the equivalent in [8,9]. Evaluation requires innovation both in datasets and in measurement, described below.

3.1 Dataset

We need a dataset with images and corresponding sentences and also labels for our representations of the meaning space. No such dataset exists. We build our own dataset of images and sentences around the PASCAL 2008 images. This means we can use and compare to state of the art models and image annotations in PASCAL dataset.

PASCAL Sentence data set. To generate the sentences, we started with the 2008 PASCAL development kit. We randomly selected 50 images belonging to each of the 20 categories. Once we had a set of 1000 images, we used Amazon's Mechanical Turk to generate five captions for each image. We required the annotators to be based in the US, and that they pass a qualification exam testing their ability to identify spelling errors, grammatical errors, and descriptive captions. More details about the methods of collection can be found in [26]. Our dataset has 5 sentences for each image of the thousand images resulting in 5000 sentences. We also manually add labels for triplets of $\langle objects, actions, scenes \rangle$ for each images. These triplets label the main object in the image, the main action, and the main place. There are 173 different triplets in our train set and 123 in test set. There are 80 triplets in the test set that appeared in the train set. The dataset is available at `http://vision.cs.uiuc.edu/pascal-sentences/`.

3.2 Inference

Our model is learned to maximize the sum of the scores along the path identified by a triplet. In inference we search for the triplet which gives us the best

additive score, $argmax_y w^T \Phi(x_i, y)$. These models prefer triplets with combination of strong and poor responses over all mediocre responses. We conjecture that a multiplicative inference model would result in better predictions as the multiplicative model prefers all the responses to be reasonably good. Our multiplicative inference has the form of $argmax_y \prod w^T \Phi(x_i, y)$. We select the best triplet given the potentials on the nodes and edges greedily by relaxing an edge and solving for the best path and re-scoring the results using the relaxed edge.

3.3 Matching

Once we predict triplets for images and sentences we can score a match between an image and a sentence. If an image and a sentence predict very similar triplets, they should be projections of nearby points in the meaning space, and so they should have a high matching score. A natural score of the similarity of sentence triplets and image triples is the sum of ranks of sentence meaning and image meaning; the pair with smallest value of this sum is both strongly predicted by the image and strongly predicted by the sentence. However, this score is likely to be noisy, and is difficult to compute, because we must touch all pairs of meanings. We use a good, noise resistant approximation. To obtain the score, we:

- obtain the top k ranking triplets derived from sentences and compute the rank of each as an image triplet
- obtain the top k ranking triplets derived from images and compute the rank of each as a sentence triplet
- sum the sum of ranks for each of these sets, weighted by in the inverse rank of the triplet, so as to emphasize triplets that score strongly.

3.4 Out of Vocabulary Extension

We generate sentences by searching a pool of sentences for one that has a good match score to the image. We cannot learn a detector/classifier for each object/action/scene that exists. This means we need to score the similarity between the image and sentences that contain unfamiliar words. We propose using text information to attack this problem. For each unknown object we can produce a score of the similarity of that object with all of the objects in our vocabulary using distributional semantics methods explained in section 2.3 . We do the same thing for verbs and scenes as well. These similarity measures work as a crude guide to our model. For example, in Figure 6, we don't have a detector for "Volkswagen", "herd", "woman", and "cattle" but we can recognize them. our similarity measures provides a similarity distributions over things we know. This similarity distribution helps us to recognize objects, actions, and scenes for which we have no detector/classifier using objects/actions/scenes we know.

3.5 Experimental Settings

We divide our 1000 images to 600 training images and 400 testing images. We use 15 nearest neighbors in building potentials for images and sentences. For matching we use 50 closest triplets.

3.6 Mapping to the Meaning Space

Table 1 compares the results of mapping the images to the meaning space, predicting triplets for images. To do that, we need a measure of comparisons between pairs of triplets, the one that we predict and the ground truth triplets. One way of doing this is by simple comparisons of triplets. A prediction is correct if all three elements agree and wrong otherwise. We could also measure if any of the elements in the triplet match. Each score is insensitive to important aspects of loss. For example, predicting $\langle cat, sit, mat\rangle$ when ground truth is $\langle dog, sit, ground\rangle$ is not as bad as predicting $\langle bike, ride, street\rangle$. This implies that the penalty for confusing cats with dogs should be smaller than that for confusing cats with bikes. The same argument holds for actions and scenes as well. We also need our measure to take into account the amount of information a prediction conveys. For example, predicting $\langle object, do, scene\rangle$ is less favorable than $\langle cat, sit, mat\rangle$.

Tree-F1 measure. Tree-F1 measure: We need a measure that reflects two important interacting components, accuracy and specificity. We believe the right way to score error is to use taxonomy trees. We have taxonomy trees for objects, actions, and scenes and we can use them to measure the accuracy, relevance, and specificity of predictions. We introduce a novel measure, Tree-F1, which reflects how accurate and specific the prediction is. Given a taxonomy tree for, say, objects objects, we represent each prediction by the path from the root of the taxonomy tree to the predicted node. For example, if the prediction is cat we represent it as $Objects \Rightarrow animal \Rightarrow cat$. We can then report the standard F1 measure using the precision and recall. Precision is defined as the total number of edges on the path that matches the edges on the ground truth path divided by the total number of edges on the ground truth path and recall as the total number of edges on the predicted path which is in the ground truth path divided by the total number of edges in the path. For example, the measure for predicting dog when the ground truth is cat is 0.5 where the precision is 0.5 and recall is 0.5, the measure for predicting animal when the ground truth is cat is 0.66, and it is 0 for predicting bike when the ground truth is cat. The same procedure is applied to actions and scenes. The Tree-F1 measure for a triple is the mean of the three measures for objects, actions, and scenes. Table 1 shows Tree-F1 measures for several different experimental settings.

BLUE Measure. Similar to Machine translation approaches where reports of accuracy involves scores for the correctness of the translation and the correctness of the generated translation in terms of language and logic, we also consider another measure to check if the triplet we generate is logically valid or not. Analogous to the BLEU score in machine translation literature we introduce the "BLUE" score which measures this. For example, $\langle bottle, walk, street\rangle$ is not valid. For that, we check if the triplet ever appeared in our corpus or not. Table 1 shows these scores for the triplets predicted by several different experimental settings.

Table 1. Evaluation of mapping from the image space to the meaning space. "Obj" means when we only consider the potentials on the object node and use uniform potentials for other nodes and edges. "No Edge" means assuming a uniform potential over edges. "FW(A)" stands for fixed weights with additive inference model. This is the case where we use all the potentials but we don't learn any weights for them. "SL(A)" means using structure learning with additive inference model. "FW(M)" is similar to "FW(A)" with the exception that the inference model is multiplicative instead of additive. "SL(M)" is the structure learning with multiplicative inference.

	Obj	No Edge	FW(A)	SL(A)	FW(M)	SL(M)
Mean Tree-F1 for first 5	0.44	**0.52**	0.38	0.45	0.47	0.51
Mean BLUE for first 5	0.24	0.27	0.16	0.58	**0.76**	0.74
Mean Tree-F1 for first 5 objects	**0.59**	0.58	0.36	0.53	0.55	0.57
Mean Tree-F1 for first 5 actions	0.27	**0.52**	0.50	0.37	0.42	0.47
Mean Tree-F1 for first 5 scenes	0.28	0.48	0.28	0.44	0.46	**0.48**

4 Results

To evaluate our method we provide qualitative and quantitative results. There are two stages in our model. First we show the ability of our method to map from the image space to the meaning space. We then evaluate our results on predicting sentences for images, annotation. We also show qualitative results for finding images for sentences, illustration.

4.1 Mapping Images to Meanings

Table 1 compares several different experimental settings in terms of two measures explained above, Tree-F1 and BLUE. Each column in Table 1 corresponds to an experimental setting. We report average Tree-F1 and average BLUE measures for five top triplets for all images. We also breakdown the Tree-F1 to objects, actions, and scenes in bottom three rows of the table.

4.2 Annotation: Generating Sentences from Images

Figure 3 shows top 5 predicted triplets and top 5 generated sentences for example images in our test set. Quantitative evaluation of generated sentence is very challenging. We trained 2 individuals to annotate generated sentences. We ask them to annotate each generated sentence by either 1, 2, or 3. 1 means that the sentence is quite accurate with possible little mistakes about details in the sentence. 2 implies that the sentence have a rough idea about the image but it's not very accurate and 3 means that the sentence is not even remotely close to the image. We generate 10 sentences for each image. The total average of the scores given by these individuals is 2.33. The average number of sentences with score one per image is 1.48. The average number of sentences with score 2 per image is 3.8. 208 of 400 images have at least one sentence with score 1. 354 sentences out of 400 images have at least one sentence with score 2.

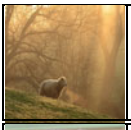
	(pet, sleep, ground) (dog, sleep, ground) (animal, sleep, ground) (animal, stand, ground) (goat, stand, ground)	see something unexpected. Cow in the grassfield. Beautiful scenery surrounds a fluffly sheep. Dog hearding sheep in open terrain. Cattle feeding at a trough.
	(furniture, place, furniture) (furniture, place, room) (furniture, place, home) (bottle, place, table) (display, place, table)	Refrigerator almost empty. Foods and utensils. Eatables in the refrigerator. The inside of a refrigerator apples, cottage cheese, tupperwares and lunch bags. Squash apenny white store with a hand statue, picnic tables in front of the building.
	(transportation, move, track) (bike, ride, track) (transportation, move, road) (pet, sleep, ground) (bike, ride, road)	A man stands next to a train on a cloudy day A backpacker stands beside a green train This is a picture of a man standing next to a green train There are two men standing on a rocky beach, smiling at the camera. This is a person laying down in the grass next to their bike in front of a strange white building.
	(display, place, table) (furniture, place, furniture) (furniture, place, furniture) (bottle, place, table) (furniture, place, home)	This is a lot of technology. Somebody's screensaver of a pumpkin A black laptop is connected to a black Dell monitor This is a dual monitor setup Old school Computer monitor with way to many stickers on it

Fig. 3. Generating sentences for images: We show top five predicted triplets in the middle column and top five predicted sentences in the right column

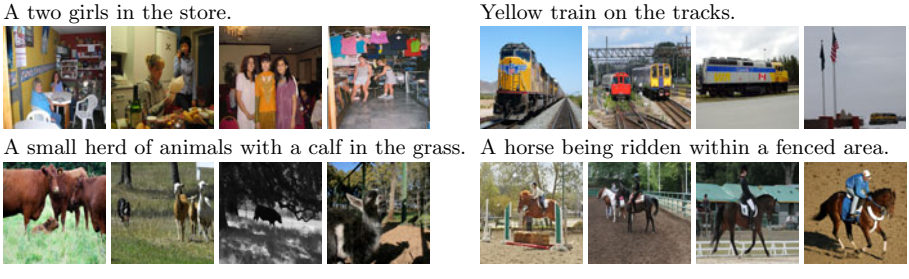


Fig. 4. Finding images for sentences: Once the matching in the meaning space is established we can generate sentences for images (annotation) and also find images that can be best describe by a sentence. In this picture we show four sentences with four 144 highest ranked images. We provide a list of 10 highest score images for each sentence for the test set in the supplementary material.

4.3 Illustration: Finding Images Best Described by Sentences

Not only our model can provide sentences that describe an image, but it also can find images which are best described by a given sentence. Once the connections to the meaning space is established, one could go in both directions, from images to sentences or the other way around. Figure 4 shows examples of finding images for sentences. For more qualitative results please see the supplementary material.

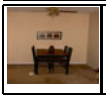
	A male and female giving pose for camera. A peaceful garden The food is ready on table.
	The two girls read to drive big bullet. Man with a goatee beard kneeling in front of a garden fence. Lone bicyclist sitting on a bench at a snowy beach.
	Black goat in a cage Horse behind a fence Wooly sheep standing next to a fence on a sunny day.

Fig. 5. Examples of failures in generating sentences for images.

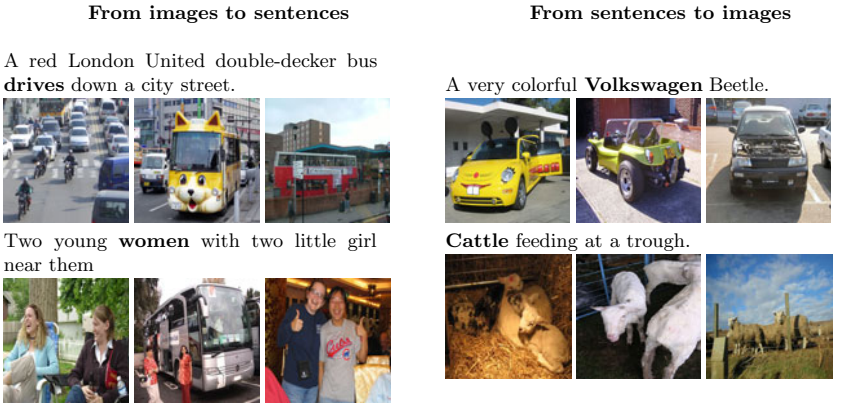


Fig. 6. Out of vocabulary extension: We don't have detectors for "drives", "women", "Volkswagen", and "Cattle". Despite this fact, we could recognize these objects/actions. Distributional semantics provide us with the ability to model unknown objects/actions/categories with their similarities to known categories. Here we show examples of sentences and images when we could recognize these unknowns for both generating sentences from images and finding images for sentences.

4.4 Out of Vocabulary Extension

Figure 6 depicts examples of the cases where we could successfully recognize objects/actions for which we have no detector/classifier. This is very interesting as the intermediate meaning space allows us to benefit from distributional semantics. This means that we can learn to recognize unknown objects/actions/scenes by looking at the patterns of responses from other similar known detector/classifiers.

5 Discussion and Future Work

Sentences are rich, compact and subtle representations of information. Even so, we can predict good sentences for images that people like. The intermediate

meaning representation is one key component in our model as it allows benefiting from distributional semantics. Our sentence model is oversimplified. We think an iterative procedure for going deeper in sentences and images would be the right direction. Once a sentence is generated for an image, it is much easier to check for adjectives and adverbs.

Acknowledgements

This work was supported in part by the National Science Foundation under IIS - 0803603 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program, in part by a gift from Google. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation or the Office of Naval Research. Ali Farhadi was supported by the Google PhD fellowship. We also would like to thank Majid Ashtiani for his help on cluster computing, and Hadi Kiapour, Attiye Hosseini for their help on evaluation.

References

1. Barnard, K., Duygulu, P., Forsyth, D.: Clustering art. In: CVPR, vol. II, pp. 434–441 (2001)
2. Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: WMISR (1999)
3. Duygulu, P., Barnard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
4. Datta, R., Li, J., Wang, J.Z.: Content-based image retrieval: approaches and trends of the new age. In: MIR 2005, pp. 253–262 (2005)
5. Forsyth, D., Berg, T., Alm, C., Farhadi, A., Hockenmaier, J., Loeff, N., Wang, G.: Words and pictures: Categories, modifiers, depiction and iconography. In: Object Categorization: Computer and Human Vision Perspectives, CUP (2009)
6. Phillips, P.J., Newton, E.: Meta-analysis of face recognition algorithms. In: ICAFGR (2002)
7. Gupta, A., Davis, L.: Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 16–29. Springer, Heidelberg (2008)
8. Li, L.J., Fei-Fei, L.: What, where and who? classifying event by scene and object recognition. In: ICCV (2007)
9. Li, L.J., Socher, R., Fei-Fei, L.: Towards total scene understanding:classification, annotation and segmentation in an automatic framework. In: CVPR (2009)
10. Gupta, A., Davis, L.: Objects in action: An approach for combining action understanding and object perception. In: CVPR (2007)
11. Gupta, A., Davis, A.K.,, L.: Observing human-object interactions: Using spatial and functional compatibility for recognition. Trans. on PAMI (2009)
12. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: CVPR (2010)
13. Berg, T.L., Berg, A.C., Edwards, J., Forsyth, D.A.: Who's in the picture. In: Advances in Neural Information Processing (2004)

14. Mensink, T., Verbeek, J.: Improving people search using query expansions: How friends help to find people. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 86–99. Springer, Heidelberg (2008)
15. Luo, J., Caputo, B., Ferrari, V.: Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In: NIPS (2009)
16. Coyne, B., Sproat, R.: Wordseye: an automatic text-to-scene conversion system. In: SIGGRAPH 2001 (2001)
17. Gupta, A., Srinivasan, P., Shi, J., Davis, L.: Understanding videos, constructing plots: Learning a visually grounded storyline model from annotated videos. In: CVPR (2009)
18. Yao, B.Z., Yang, X., Lin, L., Lee, M.W., Zhu, S.C.: I2t: Image parsing to text description. Proc. IEEE (2010) (in Press)
19. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR 2008 (2008)
20. Hoiem, D., Divvala, S., Hays, J.: Pascal voc 2009 challenge. In: PASCAL challenge workshop in ECCV (2009)
21. Oliva, A., Torralba, A.: Building the gist of a scene: the role of global image features in recognition. In: Progress in Brain Research, p. 2006 (2006)
22. Curran, J., Clark, S., Bos, J.: Linguistically motivated large-scale nlp with c&c and boxer. In: ACL, pp. 33–36
23. Lin, D.: An information-theoretic definition of similarity. In: ICML, 296–304 (1998)
24. Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C.: Learning structured prediction models: a large margin approach. In: ICML, pp. 896–903 (2005)
25. Ratliff, N., Bagnell, J.A., Zinkevich, M.: Subgradient methods for maximum margin structured learning. In: ICML (2006)
26. Rashtchian, C., Young, P., Hodosh, M., Hockenmaier, J.: Collecting image annotations using amazon's mechanical turk. In: NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (2010)

An Eye Fixation Database for Saliency Detection in Images

Subramanian Ramanathan[1], Harish Katti[2], Nicu Sebe[1],
Mohan Kankanhalli[2], and Tat-Seng Chua[2]

[1] Department of Information Engineering and Computer Science,
University of Trento, Italy
[2] School of Computing, National University of Singapore (NUS), Singapore
subramanian@disi.unitn.it

Abstract. To learn the preferential visual attention given by humans to specific image content, we present NUSEF- an eye fixation database compiled from a pool of 758 images and 75 subjects. Eye fixations are an excellent modality to learn semantics-driven human understanding of images, which is vastly different from feature-driven approaches employed by saliency computation algorithms. The database comprises fixation patterns acquired using an eye-tracker, as subjects free-viewed images corresponding to many semantic categories such as *faces* (human and mammal), *nudes* and *actions* (*look*, *read* and *shoot*). The consistent presence of fixation clusters around specific image regions confirms that visual attention is not subjective, but is directed towards *salient* objects and object-interactions.

We then show how the fixation clusters can be exploited for enhancing image understanding, by using our eye fixation database in an active image segmentation application. Apart from proposing a mechanism to automatically determine characteristic fixation seeds for segmentation, we show that the use of fixation seeds generated from multiple fixation clusters on the *salient* object can lead to a 10% improvement in segmentation performance over the state-of-the-art.

1 Introduction

The past decade has seen tremendous progress in the field of image understanding and retrieval. Breakthroughs have been achieved in robustly detecting and characterizing image objects [1,2], as well as in classifying scenes from image and video [3,4]. Nevertheless, computer vision's goal to 'enable computers to see what humans see' currently seems out of reach, and contemporary algorithms are focused on accurately interpreting and deriving a bag of keywords [5,6] for visual content.

Since human cognition is designed to process only limited information at any given time, our understanding of images is influenced by *what we attend to*, termed **visual attention**. Significant recent research has been devoted to understanding human visual attention. Through an urn model for object recall,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 30–43, 2010.

the authors in [7] demonstrate the inherent order of 'importance' assigned by human observers to scene objects. Most recently, the need for an eye-tracking database to train a model to predict where humans would look at in an image, is discussed in [8]. The database is motivated by the fact that human-observed 'regions of interest' are driven by top-down (task/semantics-based) as well as bottom-up (content/feature-based) processing, and generally don't match those predicted by image saliency computation methods [9,10,11,12,13,14].

In this paper, we present NUSEF- a database of eye-fixations compiled using an eye-tracker from a pool of 75 subjects and 758 images, spanning a large number of semantic categories. While [8] presents an eye-fixation database to learn what viewers attend to in everyday scenes, our database consists of a significant number of semantically *affective* (emotion-evoking) images. We believe that the analysis of visual attention for affective content can add a new dimension to eye-tracking research, and also offer interesting insights into how eye fixations are driven by image semantics- for *e.g.*, normal (*neutral*, *smiling*) faces are viewed differently from strongly expressive (*surprise*, *disgust*) faces and there are characteristic fixation patterns for images depicting actions (such as *look*, *read* and *shoot*) [15].

Our experimental results indicate that eye fixations are heavily influenced by image semantics and are consistently specific to *salient* (most important/meaningful) scene objects and object-interactions, which we call **attentional-bias**. Since the fixation data was acquired as subjects free-viewed images (*i.e*, in the absence of any pre-specified task), this observation is in contrast to the long-standing argument that top-down content processing by humans is subjective, and therefore, prone to extensive variability. Indeed, similar observations are also made in [16], where the authors argue that visual attention is essentially guided by recognized objects, with low-level saliency contributing only indirectly. We hope that this fixation database will particularly benefit members of the vision, multimedia, cognitive science and HCI communities.

Also, viewers exhibit exploratory behavior and attend to multiple regions-of-interest, as they observe salient objects. For *e.g.*, in *face* images, fixations are not concentrated around the center of the face but spread around the eyes, nose and mouth. We demonstrate how this phenomenon can be exploited for enhancing image understanding, using active image segmentation as an example. An algorithm for automatically segmenting the image region containing a fixation point is described in [17]. Employing the fixation point as a representative seed for the foreground object, the set of boundary edges around the fixated region are computed through energy minimization in polar space to produce promising results. While the authors claim that the fixation can be any random point in the object's interior, no methodology is provided to automatically select fixation points. On the contrary, a manually annotated point is taken as the fixation seed. Using acquired fixation patterns, we (i) propose a mechanism to automatically select the fixation seed and (ii) show how viewer's exploratory behavior can be exploited to generate multiple fixation seeds for segmentation, thereby contributing to a tremendous improvement in segmentation performance.

To summarize, the main contributions of this paper are the following:

1. A rich database of eye fixations for an image set spanning a comprehensive list of semantic categories, including a significant number of *affective* images. We believe that our eye fixation database, along with [8], will offer an excellent repository of ground truth data for visual attention and image understanding research.
2. Exploiting the attentional bias, or the clustering of fixations around the *salient* object, to automatically generate the fixation seed for active image segmentation.
3. Improving on the active segmentation performance achieved in [17] by 10%, upon generating multiple fixation seeds for segmentation within the *salient* object.

The paper outline is as follows. The next section describes acquisition, content, and other key characteristics of the eye fixation database. Section 3 discusses how attentional bias is exploited to improve the performance of active segmentation, along with the experimental results. We end with the main conclusions and directions for future work in Section 4.

2 Eye Fixation Database

The NUSEF (NUS Eye Fixation) database was acquired from undergraduate and graduate volunteers aged 18-35 years (μ=24.9, σ=3.4). The **ASL**^{TM} eye-tracker was used to non-invasively record eye fixations, as subjects free-viewed images. We chose a diverse set of 1024×728 resolution images, representative of various semantic concepts and capturing objects at varying scale, illumination and orientation, based on quality and aspect ratio constraints. Images comprised everyday scenes from *Flickr*, aesthetic content from *Photo.net*, *Google* images and emotion-evoking IAPS [18] pictures. The images [1] and Matlab code to visualize the image-wise and user-wise fixation characteristics have been made available at *http://mmas.comp.nus.edu.sg/NUSEF.html*.

2.1 Data Collection Protocol

From a collection of 1000 images, subjects were asked to view a random set of 400 images, over two passes, separated by a 10 minute interval. Each image was presented for 5 seconds and followed by a gray mask for 2 seconds, in order to destroy image persistence. The eye-tracker system consists of an infra-red sensing camera, placed alongside the computer monitor, at a distance of about 30 inches from the subject. Images were presented on a 17 inch LCD monitor, with a screen resolution of 96 dpi. Upon 9-point gaze calibration, the eye-tracker is accurate within the nearest 1^o visual angle at 3 feet viewing distance, which translates to an error radius of around 5 pixels on screen. The screen locations

[1] Except for copyrighted IAPS images, which may be obtained upon request from *http://csea.phhp.ufl.edu/media/*. IAPS-image IDs are provided, instead.

Table 1. Image distribution for NUSEF based on semantic category

Semantic Category	Image Description	Image Count
Face	Single or multiple human/mammal faces.	77
Portrait	Face and body of single human/mammal.	159
Nude		41
Action	Images with a pair of interacting objects (as in *look*, *read* and *shoot*).	60
Affect-variant group	Group of 2-3 images with varying affect.	46
Other concepts	Indoor, outdoor scenes, *world* images comprising living and non-living entities, *reptile*, *injury*.	375

that the subject observes (termed point-of-gaze), are sampled at 30 Hz, and processed to generate the coordinates and duration for every fixation. A fixation point represents the screen location where the point-of-gaze remains within 2^o visual angle for at least 100 milliseconds.

2.2 Image Content

The NUSEF database was compiled from images that were viewed by at least 13 subjects (containing a minimum of 50 fixations). Table 1 presents NUSEF's semantic category-based image distribution, while Table 2 compares our database to MIT's eye-tracking data [8]. Every image was viewed by an average of 25 subjects and over 57% of the images were viewed by more than 20 subjects. Therefore, the database provides statistically rich ground truth for image understanding applications.

Fig.1 shows the fixation patterns for various semantic image categories. Fixations are denoted by circles of varying sizes and gray-levels. The circle sizes

Table 2. Comparison between MIT database [8] and NUSEF in a nutshell

Database	# images	Average # viewers per image	Semantics	Remarks
MIT [8]	1003	15	Everyday scenes from *Flickr* and *LabelMe*	Fixations are found around faces, cars and text. Many fixations are biased towards the center.
NUSEF	758	25.3	Expressive face, nude, action, reptile and affect-variant group	Attentional-bias towards salient objects and object-interactions. Fixations are strongly influenced by scene semantics.

are indicative of the fixation duration at the point-of-gaze, while the gray-levels denote fixation starting time during the 5 second image presentation period. Evidently, a majority of the later fixations are around salient objects/regions even if early fixations may be influenced by other factors (image center, brightness, *etc.*). Low-level saliency drives visual attention in contextless indoor and outdoor scenes (Fig.1(a,b)). As also noted in [8], fixations are observed around specific regions like the eyes, nose and mouth for *faces* (Fig.1(c,d,e,f)). For *neutral* and *smiling* faces, attention is distributed almost equally between the upper (eyes) and lower (nose+mouth) halves of the face, while fixations are biased towards the lower half in highly expressive (*angry*, *surprise*, *disgust*) faces ((Fig.1(d)) (fixation statistics in [15]).

Semantic image categories unique to NUSEF include *nudes*, *actions* such as *look*, *read*, *shoot*, and *affect-variant groups*, which comprise a set of 2-3 images with similar content, but with each image inducing a different affect (*e.g.*, pleasant, neutral and unpleasant). Faces attract maximum attention in human and

Fig. 1. Exemplar images from various semantic categories (top) and corresponding gaze patterns (bottom) from NUSEF. Categories include Indoor (a) and Outdoor (b) scenes, *faces*- mammal (c) and human (d), *affect-variant group* (e,f), *action-look* (g) and *read* (h), *portrait*- human (i,j) and mammal (k), *nude* (l), *world* (m,n), *reptile* (o) and *injury* (p). Darker circles denote earlier fixations while whiter circles denote later fixations. Circle sizes denote fixation duration.

mammal *portraits* (Fig.1(i,j,k)), whereas most fixations occur on the body for *nudes* (Fig.1(l)). *Action* images (Fig.1(g,h)) are characterized by frequent fixation transitions between interacting objects, with more transitions occurring from the *action recipient* to the *action source* [15] (*e.g.* Man and book are *action source* and *recipient* respectively in Fig.1(h)). Affect-variant groups allow for a closer analysis of attentional bias, when objects are introduced/deleted in/from the image. The injured/missing eye in Fig.1(e) attracts the most attention, while the fixation distribution is more typical when the missing eye is replaced using image manipulation techniques in Fig.1(f). Fixations are observed around living beings in *world* images Fig.1(l,m), as well as unpleasant concepts such as *reptile* (Fig.1(o)) and *injury* (Fig.1(p)).

2.3 Analysis of Visual Attention Characteristics

Based on the fixation patterns observed for various semantic image categories, we summarize the following about human visual attention characteristics:

1. Human visual attention is undoubtedly influenced by image semantics. Except for contextless indoor and outdoor scenes, fixation clusters are clearly observed around *salient* objects/regions, and we term this phenomenon as attentional-bias. Concepts such as living beings, faces, *etc.* are *salient*, and generally attract considerable visual attention. Also, it appears that attentional-bias is independent of illumination, orientation as well as scale of the *salient* object/concept. This is evident from Fig.1(m,n), where over 90% of the total fixations are observed within 5% of the image area.
2. Scale of the object-of-focus and underlying semantics determine the *salient* image concept(s). Faces are *salient* in *portraits*, and within the face, the eyes, nose and mouth are *salient*. Unpleasant concepts such as reptiles, blood and injury, considerably influence visual attention whenever present. The fact that recognized concepts drive visual attention adds support to the theory that visual attention and object recognition are concurrent processes, and this is an interesting topic of research in the cognitive science community.
3. Visual attention patterns for *action* images are characterized by extensive fixation transitions between interacting objects. This inference is useful for characterizing actions, which otherwise cannot be detected using vision-based approaches. The observed fixation patterns are useful for developing a model to predict interesting regions in unknown images [8], or to localize the spatial locations of *salient* objects and actions [15].
4. Fixations around different 'regions of interest' confirm the exploratory behavior exhibited by the viewers, as they attend to *salient* content. This is particularly useful as human cognition can identify two content-wise dissimilar (due to differing color, texture *etc.*) image regions, as components of the same semantic entity. Overlap of the fixations corresponding to the two regions offer us vital cues, which can be exploited for enhancing automated image understanding. In the next section, we present one such example where the various fixation clusters observed on an object of interest are processed

to generate multiple fixation seeds for active image segmentation. Employing multiple fixation seeds instead of one for active segmentation is found to enhance segmentation performance tremendously.

3 Enhancing Active Image Segmentation with Multiple Fixations

Even as visual attention is specific to *salient* objects, all the fixations on the salient object are generally not restricted to a specific region. Instead, fixations tend to cluster around regions-of-interest within the salient object. If multiple, spatially overlapping, fixation clusters can be discovered from fixation patterns, information from the various clusters can be integrated to infer properties of the entire object. As an exemplar application, we demonstrate how statistically rich fixation data from NUSEF can be utilized to enhance fixation-based active segmentation.

A fixation-based image segmentation scheme, whose objective is to compute the enclosing contour containing the fixation point, has been recently proposed in [17]. Based on the premise that the human eye invariably fixates within the interior of an object, the algorithm attempts to find the set of boundary contours surrounding the fixation. Upon computing the probabilistic boundary edge map to determine the likelihood of an edge pixel being on an actual depth boundary through a combination of monocular, stereo and motion cues, the algorithm proceeds by transforming the edge-map onto polar space, with the fixation point at the pole. The polar space transformation is carried out in order to avoid the problem of graph-cut approaches preferring shorter contours over longer ones, so as to obtain the 'real' boundary contours.

Segmentation, then becomes the problem of finding the optimal cut through the polar edge map, so that edge pixels to the left of the cut are inside the fixation region, while those to the right are outside. An energy function is defined for assigning binary labels '0' and '1' to pixels inside and outside respectively, and the optimal segmentation is obtained as the graph-cut that minimizes the energy function.

3.1 Algorithm Analysis

While the segmentation procedure proposed in [17] is intuitive and the achieved segmentation performance is better than or comparable to other contemporary algorithms [19,20,21], the fixation-based active segmentation scheme suffers from the following shortcomings:

i. The active segmentation algorithm relies on a solitary fixation seed, which it considers to be representative of the foreground object (object-of-interest). This is not true of real fixation data as in general, humans tend to fixate at multiple locations on the object of interest (such as eyes, nose and mouth on a face). Intuitively, segmentation achieved from multiple fixations should

be more accurate and robust compared to the segmentation achieved using a solitary fixation.

ii. While the fixation point is assumed to be any random point within the interior of the object, there is no methodology provided to automatically select the fixation points. Instead, the algorithm requires the user to input the fixation point. Automatic selection of the fixation seed should be trivial with real fixation data, as most fixation points should lie within the *salient* object. At the least, the centroid of the fixation points can be safely assumed to lie within the foreground.

iii. In some cases, using multiple fixation seeds can enable a more accurate segmentation. The authors do not discuss how segments obtained from more than one fixation seed within the same object may be combined to generate the foreground segmentation.

To investigate the hypothesis that multiple fixations available from real eye-fixation data should enhance segmentation performance, and to exploit the

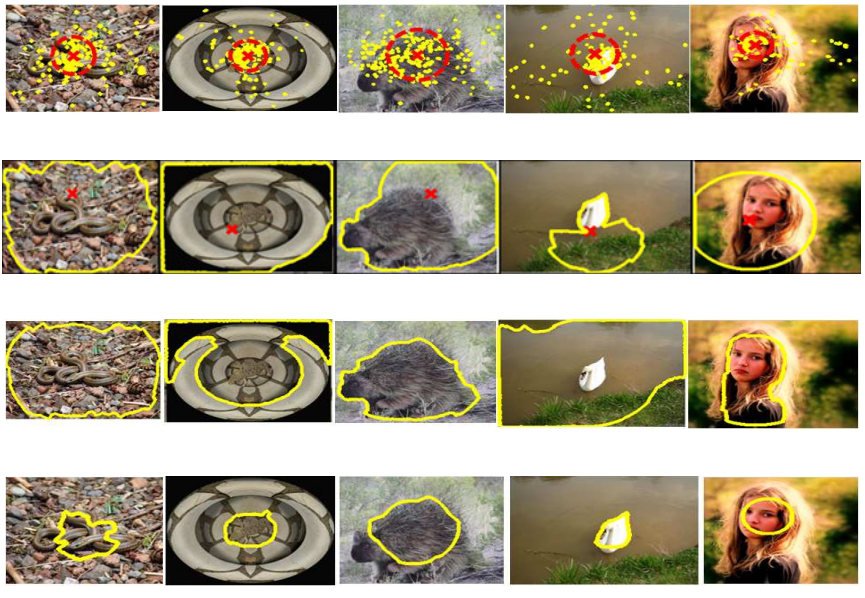
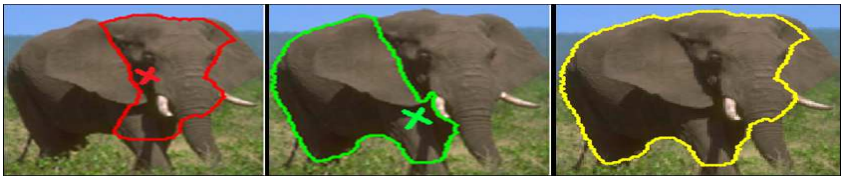


Fig. 2. Enhanced segmentation with multiple fixations. The first row shows the normalized fixation points (yellow). The red 'X' denotes centroid of the fixation cluster around the *salient* object, while the circle represents the mean radius of the cluster. Second row shows segmentation achieved with a random fixation seed inside the object of interest [17]. Third row contains segments obtained upon moving the segmentation seed to the fixation cluster centroid. Incorporating the fixation distribution around the centroid in the energy minimization process can lead to a 'tighter' segmentation of the foreground, as seen in the last row.



(a)

(b)

Fig. 3. More fixation seeds are better than one- Segments from multiple fixation clusters can be combined to achieve more precise segmentation as seen for the (a) *portrait* and (b) *face* images. The final segmentation map (yellow) is computed as the union of intersecting segments. Corresponding fixation patterns can be seen in Fig.1

fixation clusters around salient objects owing to attentional bias, we performed the following experiments:

(a) To determine whether the segmentation performance of [17] is indeed stable and accurate irrespective of the fixation location, we obtained the output segments for 20 randomly selected fixation seeds from within the hand-drawn segmentation maps for 80 NUSEF images. The baseline segmentation performance is determined as the mean value of the F-measure for the 20 segments obtained from the random seeds. The F-measure, which is used as a measure of the segmentation performance accuracy, is defined as:

$$F = 2PR/(P + R) \tag{1}$$

where P and R denote precision and recall respectively. P denotes the fraction of the segmentation output overlapping with the ground truth, while R represents fraction of the ground-truth overlapping with the output segment.

(b) Considering the set of all fixation points for a given image, a characteristic fixation seed is generated as the centroid of the largest fixation cluster. This allows for the fixation seed to be computed automatically from real fixation data, and since the NUSEF contains statistically rich fixation data, the segmentation output for this characteristic seed, should be more stable than that obtained with a random fixation. Also, as seen from Figs.2 and 3, the centroid of the largest fixation cluster generally lies within the *salient* object, and therefore, the segmentation output with the centroidal fixation seed should be comparable to that obtained in (a). As seen from Fig.2 (rows 2 and 3), using the centroidal seed can sometimes produce a more desirable

segmentation. The largest fixation cluster is computed as follows. In order to account for the fixation duration at every fixated location, each fixation is weighted by the minimum fixation duration in order to generate a corresponding number of 'normalized fixation points' within a Gaussian kernel around the fixation location (this is the inverse of how a fixation is computed). Agglomerative hierarchical clustering is then employed to remove outliers and retain 90% of the original points based on Euclidian distance from the cluster center.

(c) As fewer fixations are observed as we travel radially away from the centroid, the fixation distribution around the centroid can be used as a reliable estimate of the foreground expanse. We recomputed the output segmentation by

Algorithm 1. Pseudo-code for (a), (b), (c), (d)

Steps in (a)
- Using [17], obtain segments for 20 random fixation seeds chosen from within the ground-truth segmentation.
- Compute F as the mean of the F-measures for the 20 segments (using Eq. 1).

Steps in (b)
- (i) for all fixation points fp, compute $weight_{fp} = (fixation\_duration\_at\_fp)/100$ (min fixation duration). Sample $weight_{fp}$ points within a Gaussian kernel around fp to generate normalized fixation points.
- (ii) Employ hierarchical clustering to compute the biggest fixation cluster based on Euclidian distance criterion.
- Use the centroid of this cluster as the fixation seed and invoke [17] to obtain the segmentation output.
- Compute F using Eq. 1.

Steps in (c)
- Perform step (i) to compute the normalized fixation point locations.
- Perform step (ii) to compute the biggest fixation cluster.
- (iii) Compute the centroid and assign r_{mean} as the mean distance of all points from the cluster centroid.
- (iv) Use the centroid of this cluster as the fixation seed for [17].
- (v) for all edge pixels p beyond $2 * r_{mean}$ distance from the fixation centroid, reset the labeling cost as $U_p(l_p = 0) = D$ and $U_p(l_p = 1) = 0$. This initialization discourages segmentation algorithm from labeling pixels outside $2 * r_{mean}$ distance as being 'inside' the fixation region.
- (vi)Perform the energy minimization to obtain the segmentation output.
- Compute F using Eq. 1.

Steps in (d)
- Perform steps (i),(ii) to compute the biggest fixation cluster.
- Compute sub-clusters within this cluster such that minimum cluster size $> D_{min}$ and distance between cluster centers $> D_{min}$, again employing agglomerative clustering.
- Repeat steps (ii), (iii), (iv), (v), and (vi) for all sub-clusters.
- Integrate the segments obtained from the various clusters in the final segmentation map by computing the union of segments having more than 10% overlap.
- Compute F using Eq. 1.

incorporating this information in the energy minimization process. In particular, we re-initialize the labeling cost U(.), so that all edge pixels at a distance greater than r_t from the centroid are deemed to be outside the foreground, *i.e.*, $U_p(l_p = 0) = D$ and $U_p(l_p = 1) = 0\ \forall p$ such that, $r_p \geq r_t$. Setting $r_t = 2r_{mean}$, where r_{mean} is the mean cluster radius from the centroid, works well for most images in practice. Incorporating fixation distribution information in the energy minimization process leads to a 'tighter' and more accurate foreground segmentation for difficult cases where the foreground-background similarity is high (Fig 2, fourth row).

(d) Penalizing the spread of the 'inside' region beyond r_t can at times, force the graph-cut algorithm to limit the foreground boundary at textural edges. In such cases, integrating the segmentation maps obtained from sub-clusters within the main cluster can lead to the optimal segmentation (Fig 3). From the main fixation cluster, we again employ agglomerative clustering to discover all sub-clusters that have a minimum membership (at least 5% of the total fixations) and whose centroids are separated by a minimum distance (100 pixels). The segmentation map for each cluster is computed as in (c), and we compute the final segmentation map as the union of segments that have at least 10% overlap.

The pseudo-code summarizing the steps involved in (a), (b), (c) and (d) is provided in Algorithm 1.

3.2 Results and Discussion

Performance evaluation to evaluate the effect of (a), (b), (c) and (d) was done on 80 NUSEF images, each comprising only one *salient* object. The data essentially corresponded to the following semantic categories- *Face*, *portrait*, *world* and *nude*, and included a number of challenging cases, where the foreground and background are visually similar.

As mentioned previously, the F-measure is used for evaluating segmentation accuracy. For the baseline method, the mean F-measure for the segmentation outputs produced from 20 random seeds was computed, while in all of (b), (c) and (d), a single segmentation output is produced for which the F-measure is computed. The F-measure scores for segmentation procedures (a), (b), (c) and (d) are tabulated in Table 3.

Table 3. Performance evaluation for segmentation outputs from (a), (b), (c) and (d)

Procedure	F-measure (mean ± variance)
(a)	0.6 ± 0.05
(b)	0.59 ± 0.06
(c)	0.60 ± 0.04
(d)	**0.66± 0.04**

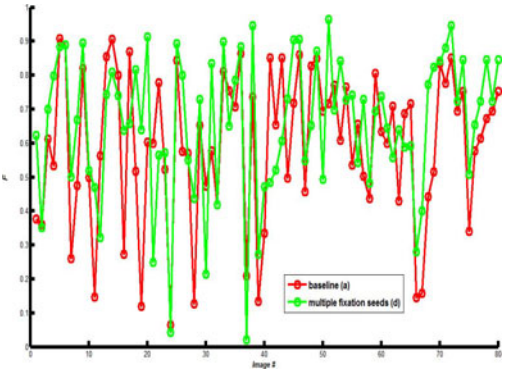


Fig. 4. F measure plot for 80 images. The legend is as follows - *red* baseline and *green* - Integration of segments obtained from multiple sub-clusters.

The F-measure scores for (a), (b) and (c) are found to be almost similar. While the fixation seeds for (a) were randomly picked from the hand-segmented ground truth, the seeds for (b) and (c) were automatically obtained from the fixation data. The fact that the segmentation performance obtained from all three procedures are comparable implies that our methodology for determining the fixation seed is valid. While incorporating the fixation distribution information in the segmentation framework can isolate the foreground more accurately for difficult cases (shown in Fig.2), it also causes the graph-cut algorithm to draw the boundaries along the edges closest to the fixation, sometimes leading to inefficient segmentation. Nevertheless, this deficiency can be overcome by considering overlapping segments obtained from multiple fixation clusters whose centers are sufficiently far away from one another, as in (d).

Fig.4 presents the F-measure plots for segmentation procedures (a) and (d). Clearly, the segmentation performance obtained using multiple fixation seeds is better than that obtained from a random fixation point for most images. This is because segments are conservatively computed in the multi-fixation seed case using the cluster spread as a cue, and then integrated to produce the final segmentation map. However, in some cases where spurious segments are picked up, the segmentation performance using multi-fixation seeds also falls. Overall, a significant 10% improvement in segmentation performance is obtained on using multiple seeds obtained from actual fixation data for segmentation as against a random fixation seed.

4 Conclusion and Future Work

This paper presents NUSEF- an eye fixation database acquired for images corresponding to many semantic categories, including *affective* content, in which visual attention is strongly driven by image semantics. The acquired fixation patterns confirm the hypothesis that eye fixations are influenced by *salient* image

content, and are largely independent of the viewer-specific preferences. We believe that this database would be particularly beneficial for visual attention and image understanding-related research. The fact that viewers show exploratory behavior while observing *salient* content, thereby generating clusters around interesting regions, is then exploited to enhance the segmentation performance achieved by the fixation-based active segmentation algorithm by as much as 10%.

Future work involves formalizing the segmentation procedure, which is currently based on certain heuristics. If fixation data can be efficiently used for object segmentation, it would benefit a number of vision and graphics applications such as content-based image retrieval and seam carving. Characterization of image data based on gaze patterns (*e.g.* action vs non-action images) is another direction for future work.

Acknowledgements

This research has been partially supported by the FP7 IP European project GLOCAL.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
2. 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. Int. J. Comput. Vision 81(1), 2–23 (2009)
3. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vision 42(3), 145–175 (2001)
4. Van Gemert, J.C., Geusebroek, J.M., Veenman, C.J., Snoek, C.G.M., Smeulders, A.W.M.: Robust scene categorization by learning image statistics in context. In: CVPR-SLAM Workshop (2006)
5. Zheng, Y.T., Neo, S.Y., Chua, T.S., Tian, Q.: Visual synset: a higher-level visual representation for object-based image retrieval. The Visual Computer 25(1), 13–23 (2009)
6. Uijlings, J.R.R., Smeulders, A.W.M., Scha, R.J.H.: Real-time bag of words, approximately. In: CIVR (2009)
7. Spain, M., Perona, P.: Some objects are more equal than others: Measuring and predicting importance. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 523–536. Springer, Heidelberg (2008)
8. Judd, T., Ehinger, K., Durand, F., Torralba, A.: Learning to predict where humans look. In: ICCV (2009)
9. Peters, R.J., Iyer, A., Itti, L., Koch, C.: Components of bottom-up gaze allocation in natural images. Vision Research 45(8), 2397–2416 (2005)
10. Valenti, R., Sebe, N., Gevers, T.: Image saliency by isocentric curvedness and color. In: ICCV (2009)
11. Liu, T., Sun, J., Zheng, N.N., Tang, X., Shum, H.Y.: Learning to detect a salient object. In: CVPR (2007)

12. Zhang, L., Tong, M.H., Marks, T.K., Shan, H., Cottrell, G.W.: SUN: A Bayesian framework for saliency using natural statistics. Journal of Vision 8(7), 1–20 (2008)
13. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. PAMI 20(11), 1254–1259 (1998)
14. Bruce, N., Tsotsos, J.: Saliency, attention, and visual search: An information theoretic approach. J. of Vision 9(3), 1–24 (2009)
15. Subramanian, R., Harish, K., Raymond, H., Chua, T.S., Kankanhalli, M.: Automated localization of affective objects and actions in images via caption text-cum-eye gaze analysis. In: ACM MM, pp. 729–732 (2009)
16. Einhuser, W., Spain, M., Perona, P.: Objects predict fixations better than early saliency. J. Vis. 8(14), 1–26 (2008)
17. Mishra, A., Aloimonos, Y., Fah, C.L.: Active segmentation with fixation. In: ICCV (2009)
18. Lang, P., Bradley, M., Cuthbert, B.: (iaps): Affective ratings of pictures and instruction manual. Technical report, University of Florida (2008)
19. Bagon, S., Boiman, O., Irani, M.: What is a good image segment? A unified approach to segment extraction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 30–44. Springer, Heidelberg (2008)
20. Alpert, S., Galun, M., Basri, R., Brandt, A.: Image segmentation by probabilistic bottom-up aggregation and cue integration. In: CVPR (2007)
21. Arbelaez, P., Cohen, L.: Constraine image segmentation from hierarchical boundaries. In: CVPR (2008)

Face Image Relighting Using Locally Constrained Global Optimization

Jiansheng Chen, Guangda Su, Jinping He, and Shenglan Ben

Department of Electronic Engineering, Tsinghua University, P.R. China
{jschenthu,susu,hjping}@mail.tsinghua.edu.cn,
bsl06@mails.tsinghua.edu.cn

Abstract. A face image relighting method using locally constrained global optimization is presented in this paper. Based on the empirical fact that common radiance environments are locally homogeneous, we propose to use an optimization based solution in which local linear adjustments are performed on overlapping windows throughout the input image. As such, local textures and global smoothness of the input image can be preserved simultaneously when applying the illumination transformation. Experimental results demonstrate the effectiveness of the proposed method comparing to some previous approaches.

Keywords: face image, relighting, illumination, optimization, ratio image.

1 Introduction

Photo-realistic re-rendering of images under lighting condition changes has attracted a lot of attention in the computer graphics and computer vision community during the past decade. Especially, the relighting of human face images has been very extensively studied because of its wide range applications in face recognition and film production. The challenge of such a problem resides in the complex and individual shape of the human face, the subtle and spatially varying reflectance properties of skin, and the extreme sensitivity of the human perception system towards the appearance of other people's faces [1].

The problem of image based face relighting can be stated as follows: given an image of a target face, determine the appearance of this face under a lighting condition different from that in the given image. In general, such a problem is difficult and the exact solution usually cannot be achieved. Assumptions have to be made on the properties of the target face, the constraints of the lighting condition, or the availability of additional information. In the inverse lighting based methods, the 3D model and the albedo of the target face are assumed to be known [1,2]. In the quotient image based method proposed by Shashua and Raviv, it is assumed that different people's faces share the same 3D shape and differ only in their albedo [3]. By approximating the face as a convex Lambertian object, face images under a wide variety of lighting conditions approximately lie on a low-dimensional linear subspace using the spherical harmonic representation [4,5]. Such a finding has been adopted in the 3D spherical harmonic basis morphable model based relighting method proposed in [6], as well as

K. Daniilidis, P. Maragos, and N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 44–57, 2010.

in the Markov Random Field (MRF) based relighting method proposed in [7]. In the face relighting system proposed in [8], the albedo of the reference subject and the lighting condition of the given target face image are known. Actually, the cross subject reflectance transfer in [8] also implicitly assumes that that shape of the reference face and target face are identical after image warping.

In this paper, we focus on the face image relighting problem using the following configuration. Suppose images of a reference face under two lighting conditions are available. Given an image of the target face under the first lighting condition, generate an image of the target face which looks as if it is taken under the second lighting condition. Intuitively, such a configuration is very close to the one studied in [8]. However, they differ in mainly two aspects. Firstly, the first lighting condition is unknown in our configuration. Secondly, the two reference images are really taken under different lighting conditions instead of been generated using a single reflectance model as is practiced in [8]. We will demonstrate that these differences change the characteristic of the problem so that the ratio image based method adopted in [8] cannot be directly applied. The objective of our method is to preserve facial details as well as minimizing artifacts while generating a photo-realistic relighting effect. The foundation of our method is the empirical fact that real life radiance environments are usually locally homogeneous. Hence neighboring pixels in the image can be combined to impose local constraints on small overlapping windows when applying global illumination transformation. As such, the face image relighting problem can be converted to a locally constrained global optimization problem, for which an efficient solution exists based on solving a large scale sparse linear system.

The rest of this paper is organized as follows. In the next section, we review previous work related to this topic. In section 3, we introduce a straightforward solution to this problem and explain why it cannot produce satisfactory results. Section 4 presents the proposed method and its implementation. Experimental results and comparisons are demonstrated in section 5. Section 6 concludes our work.

2 Related Work

Image re-rendering has been an active research topic in the field of computer vision and computer graphics. Marschner et. al. modeled the light as being emitted by a large sphere surround the object, and assumed the linearity of the light. With the known geometry and albedo of the object, a least-square system was used to find the distribution of light incident on the object in a given image. The ratios of synthesized images under different lighting conditions were then used to modify the original image to generate the relighting effect [2]. A system for measuring the geometry and albedo of human faces was further proposed in [9], in which a displacement-mapped subdivision surface is used to model face surface, and a BRDF measuring method was used to determine the albedo. The idea of the ratio image between synthesized images was later used in the film post-production system presented in [8] for solving the performance actor relighting problem in motion pictures. A high resolution and high dynamic range image database was used in this system for image synthesis. Shashua and Raviv proposed the notion of a quotient image for face relighting and recognition. By modeling human faces as Lambertian surfaces and by using the fact that the image space of Lambertian objects can be modeled using a low-dimensional representation,

they proposed a method calculating a object invariant signature, namely the quotient image, using a bootstrap dataset consisting of reference face images taken under three independent lighting conditions [3]. They also assumed fixed viewpoint, no cast shadow and no dense correspondence. Stoschek further combined the quotient image with image morphing to generate relit faces under changing poses [10]. Liu et. al. extended the idea of ratio image to the re-rendering problem due to people's expression change and proposed an expression ratio image [11]. The radiance environment map was adopted by Wen et. al. in face relighting under rotating lighting environments [12]. The spherical harmonics representation was used to approximate a radiance environment map from one or more images of a sphere. The radiance map based methods, however, ignore cast shadows and inter-reflections on faces. Using spherical harmonic to represent lighting functions on the surface of a sphere was independently proposed by Basri et. al. [5] and Ramamoorthi et. al. [13,14]. Both work analytically proved that under any lighting conditions, a nine-dimensional linear subspace accounts for most of the variability in the reflectance function for convex Lambertian surfaces. In the MRF based face relighting method, this nine-dimensional approximation was used for solving the lighting function for faces when the surface normal are available [7]. Lee et. al. further pointed out the existence of a physical setup of nine single light sources under which the face images well span the space of all face images under different lighting conditions [15].

Dense correspondence between reference images and the target face image is required in our method. In [8], easily detectable points such as eye corners and mouth corners were used as control points for the image warping in which a local histogram adjustment was also applied. In our method, we build a 105 point Active Appearance Model (AAM) for human faces. Different face images are aligned based on the AAM model fitted to them. As a generative parametric model describing both shape and appearance variations of objects, AAM was first proposed in [16], in which a Gauss-Newton process is used for the model fitting. Matthews and Baker proposed an efficient AAM fitting algorithm that did not require a linear relationship between the image difference and the model parameter difference [17]. This model has faster convergence and better fitting accuracy than the original AAM [16]. Donner et. al. proposed another fast AAM using canonical correlation analysis (CCA) that models the relation between the image difference and the model parameter difference for improving the convergence of the fitting algorithm [18]. In our work, a modified AAM fitting algorithm similar to the one proposed in [17] is used to perform dense correspondence between face images.

3 A Relighting Approach and Its Problems

The face relighting problem discussed in this paper is close to the one presented in the film post-production system in [8]. There are two human faces among which one is used as the reference face and the other is the target face to be relit. Two images of the reference face under two unknown lighting conditions, A and B, are given. We denote these two images as I_r^A and I_r^B respectively. An input image I_t^A of the target face taken under the lighting condition A or a lighting condition that is similar to A is given. The face relighting task is to generate an image I_t^B that appears as if it is an

image of the target face taken under the lighting condition B or a lighting condition that is very similar to B. Usually, the reference face and the target face are from different subjects. Nevertheless, same-subject face relighting also make sense considering the possible pose/expression/age variations [8]. Peers et. al. solved these two kinds of problem under the same framework and so do us.

Such a face relighting problem is practically meaningful in many scenarios besides the film production application discussed in [8]. For instance, in the field of face recognition, most databases, especially large scale databases consisting of millions of users, only contain face images taken under the standard frontal lighting condition. Only very few research oriented databases [19,20], which are usually small in size, include face images taken under various lighting conditions. By applying face relighting described above, the illumination variations in the research oriented databases can be transferred to other databases for generating large scale simulated face databases containing illumination variations. These simulated databases can possibly be used for large scale study of the illumination robustness of face recognition technologies.

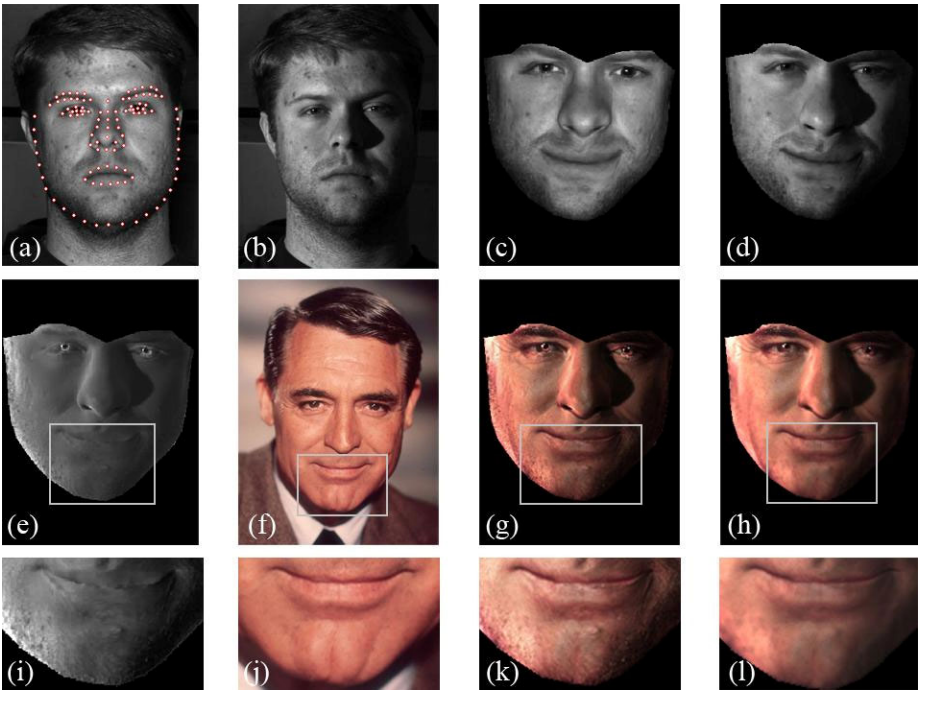


Fig. 1. Face relighting using a ratio image based method used in [8]. (a) The reference face image I_r^A fitted by an AAM model. (b) The reference image I_r^B. Reference images are from the Yale face database [19]. (c) The warped reference image $L(I_r^A)$. (d) The warped reference image $L(I_r^B)$. (e) The ratio image T. (f) The input image of the target face I_t^A. (g) The relighting result using unfiltered ratio image. (h) The relighting result using Gaussian filtered ratio image. (i)-(l) are the gamma-enhanced detail of the chin areas of images (e)-(h). (The image of Cary Grant is available at http://www.answers.com/topic/cary-grant-large-image.)

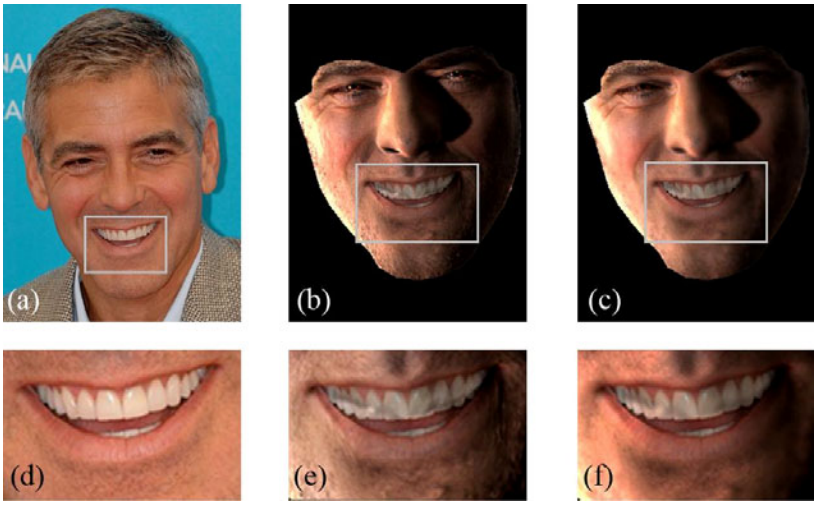


Fig. 2. Face relighting results for excessive expression change. Fig. 1(a) and Fig. 1(b) are used as references. (b) Relighting result using unfiltered ratio image. (c) Relighting result using Gaussian filtered ratio image. (d)-(f) are the gamma-enhanced detail of the mouth areas of (a)-(c). (The image of George Clooney is available at http://www.grrltv.com/2009/01/.)

A straightforward solution to the proposed problem is the ratio image based relighting method used in [8]. Firstly, corresponding facial landmark points are located using AAM fitting [17] on the three face images: I_r^A, I_r^B and I_t^A. Next, the two reference images, I_r^A and I_r^B are warped to the same pose and expression as the input target image I_t^A. This can be easily done by first applying triangulation and then using the piece-wise affine transformation or the thin-plate splines [21]. We denote the warped reference images as $L(I_r^A)$ and $L(I_r^B)$ respectively. A ratio image T is computed by dividing each pixel in $L(I_r^B)$ by the corresponding pixel in $L(I_r^A)$, or $T = L(I_r^B)/L(I_r^A)$. Divisions by zero can be replaced by small numbers. Finally, a relit image is generated as the Cartesian product between the ratio image and the input target image, or $I_t^B = T \otimes I_t^A$. A color image can be first decomposed into the hue, saturation, and gray-value components. The relighting process is applied to the gray-value component only. The relit color image is achieved by composing the original hue and saturation components, and the newly generated gray-value component.

Fig. 1 shows an example of the relighting method described above. Fig. 1(g) clearly reveals the problem of such an approach: although in the input image, Cary Grant is pretty clean-shaven, beards are quite obvious in the relit image. Artifacts besides the beards are also noticeable in the relit image. Basically, these artifacts are introduced by the ratio image as is shown in Fig. 1(i). Through Cartesian product, artifacts in the ratio image are directly transferred to the relighting result. In fact, the formation of these artifacts is quite complicated. Firstly, since the two reference images are not taken at the same time, the difference between the two images may not be solely caused by the lighting condition change. Variation in the albedo of the face surface caused by factors such as the skin condition change, the facial hairs movements and so on, may introduce false texture information into the ratio image. Such a

phenomenon may be further intensified by the specularity in certain facial areas. Secondly, errors are inevitable during the dense correspondence and face image warping, which will also bring about alignment artifacts. In the relighting system described in [8], these problems are negligible in key frames for which the references images are rendered from exactly the same reflectance field, leading to a strictly accurate dense correspondence when calculating the ratio image.

Applying low pass filtering to the ratio image is an handy solution to this problem. Fig. 1(h) shows the relighting result after filtering the ratio image using a Gaussian filter with a standard deviation $\sigma = 2.0$. The artifacts are somewhat smoothed out, but are still observable. Nevertheless, such a low pass filtering method has two critical problems. Firstly, it may suppress some high frequency textures in the target image. Secondly, it does not work for low frequency artifacts. Fig. 2 illustrates these problems in relighting a face image with excessive expression changes. In Fig. 2(f), the genuine beards of George Clooney are obviously blurred, and the low frequency artifacts on the teeth are still there.

To solve these problems, we propose to substitute the Cartesian product step with a locally constrained global optimization process which can globally adjust the illumination distribution of the target image while preserving local details and isolating artifacts in the ratio image.

4 The Proposed Method

Under the assumption that the human face is Lambertian, a face image can be described by the product of the albedo and the cosine angle between a point light source and the surface normal: $I(i) = \rho(i)n(i)^T s$, where $0 \leq \rho(i) \leq 1$ is the surface reflectance associated with pixel i, $n(i)$ is the surface normal direction associated with pixel i, and s is the light source direction whose magnitude is the light source intensity [3]. Let's study a small local window $\omega \subset I$. Suppose that the light source is fairly far away from the face and consider the empirical fact that human face surface is locally smooth. We have $n(i)^T s \approx n(j)^T s$ for any $i, j \in \omega$, or $\omega(i) \approx \kappa\rho(i)$ for any $i \in \omega$, where κ is a nonnegative constant associated with the local window. Such a linear relationship holds when the light source varies from s_1 to s_2. Accordingly we have $\omega_1(i) \approx \kappa_1\rho(i)$ and $\omega_2(i) \approx \kappa_2\rho(i)$. Subsequently, $\omega_1(i) \approx \alpha\omega_2(i)$, in which $\alpha = \kappa_1/\kappa_2$ is a nonnegative constant. That is to say, a local window of face image differs only in terms of a nonnegative multiple across lighting condition changes. This conclusion can be easily generalized to multiple independent light source scenarios. We name α as the *local relighting coefficient* associated with the local window ω.

Consider an ideal case in which two human faces, A and B, are spatially accurately aligned. Let ω_A and ω_B be two spatially corresponding local windows on the two human faces. Under the same lighting condition change, the local relighting coefficient associated with ω_A and ω_B should be equal. We further assume that after image warping, the reference face and the target face in the proposed relighting problem match exactly. As such, the local relighting coefficients calculated from the warped references images can be used to 'relight' the input target face image. A straightforward solution is: first divide the face images into small blocks, then for each block calculate the local relighting coefficient in the reference images, finally multiply these

coefficients to the input target image in a block wise manner to accomplish the relighting. However, this approach is nothing but a simple extension of the relighting method described in section 3. Moreover, such a block based process will create false edges between neighboring image blocks.

Alternatively, we propose a locally constrained global optimization method similar to the one used for image matting in [22]. For a local window ω_i centered at pixel i, minimize equation (1), in which α_i is the local relighting coefficient associated with ω_i; τ_i is the value of the ratio image at pixel i, or $\tau_i = T(i)$; and λ is a nonnegative weight balancing the two terms in (1). The value of τ_i can actually be regarded as a fair guess of the local relighting coefficient, α_i. Note that pixel values of the relighting result are now variables in the objective function. If the local window ω_i contains only one pixel, equation (1) has a trivial minimal solution: $I_t^B(i) = \tau_i I_t^A(i)$, leading to exactly the same relighting result as we have discussed in the last section. However, as long as the local windows contain more than one pixel, such a trivial solution no longer exists because of the local constraints among overlapping neighboring windows introduced implicitly by equation (1). In our implementation, the smallest local window size is 3x3 pixels.

$$\mathcal{F}_i = \sum_{j \in \omega_i} \left(I_t^B(j) - \alpha_i I_t^A(j) \right)^2 + \lambda(\alpha_i - \tau_i)^2 \tag{1}$$

Combining the objective functions of all the local windows, we achieve a global optimization objective shown in equation (2), in which i sums over all pixels in the input image. Notice that local windows now overlap with each other, thus constraints inside a local window will be naturally propagated to its neighbor widows, so that the image smoothness can be retained. At the same time, strong local structures, such as edges, can be reasonable preserved when local minimization is achieved.

$$\mathcal{F} = \sum_i \left(\sum_{j \in \omega_i} \left(I_t^B(j) - \alpha_i I_t^A(j) \right)^2 + \lambda(\alpha_i - \tau_i)^2 \right) = \sum_i \mathcal{F}_i \tag{2}$$

The image relighting problem can be solved by minimizing the objective function $\mathcal{F}$, of which the variables consist of the pixel values of the relighting result image I_t^B and the local relighting coefficients α_i. When the balancing weight λ is nonnegative, the objective function $\mathcal{F}$ is convex because it is a nonnegative sum of a bunch of quadratic terms on domain $\mathbb{R}^{2N}$, where N is the number of pixels in the input face image. Hence, any convex optimization method can be adopted here [23]. Nevertheless, we will demonstrate in the following an analytical solution for minimizing the objective function $\mathcal{F}$.

$$\arg\min_{\alpha, I_t^B} \mathcal{F} = \arg\min_{I_t^B} \sum_i \arg\min_{\alpha_i} \mathcal{F}_i \tag{3}$$

Equation (3) shows an equivalent expression for the optimization problem with the objective function $\mathcal{F}$. The basic idea is to first solve the optimal value of α_i, denoted as $\hat{\alpha}_i$, by setting the first derivative of $\mathcal{F}_i$ to zero. As such, $\hat{\alpha}_i$ are expressed as functions of the pixels values of I_t^B. Then the optimal I_t^B, denoted as $\hat{I}_t^B$, can be consequently solved by setting the first derivative of function $\mathcal{F}$, in which α_i are replaced by $\hat{\alpha}_i$, to zero. Equations (4)-(9) show the derivation steps.

$$\frac{\partial \mathcal{F}_i}{\partial \alpha_i}\Big|_{\alpha_i=\hat{\alpha}_i} = 2\hat{\alpha}_i \sum_{j\in\omega_i} \left(I_t^A(j)\right)^2 - 2\sum_{j\in\omega_i} I_t^A(j) I_t^B(j) + 2\lambda\hat{\alpha}_i - 2\lambda\tau_i = 0 \quad (4)$$

Therefore we have

$$\hat{\alpha}_i = \frac{\sum_{j\in\omega_i} I_t^A(j) I_t^B(j) + \lambda\tau_i}{\gamma_i + \lambda}, \quad (5)$$

in which

$$\gamma_i = \sum_{j\in\omega_i} \left(I_t^A(j)\right)^2 \quad (6).$$

Replace α_i by $\hat{\alpha}_i$ in function $\mathcal{F}$ and set its first derivative to zero, we have

$$\frac{\partial \mathcal{F}}{\partial I_t^B(k)}\Big|_{I_t^B=\hat{I}_t^B} = \sum_{i|k\in\omega_i} 2\left(\hat{I}_t^B(k) - \frac{I_t^A(k)}{\gamma_i+\lambda}\sum_{j\in\omega_i} I_t^A(j)\hat{I}_t^B(j)\right) - \sum_{i|k\in\omega_i} \frac{2\lambda\tau_i I_t^A(k)}{\gamma_i+\lambda} = 0 \quad (7)$$

By varying k for all the pixels in the input image, a linear system is formed.

$$S \cdot \hat{I}_t^B = U \quad (8)$$

U and $\hat{I}_t^B$ are column vectors with length N, and S is a $N \times N$ square matrix. The elements of U and S are expressed by equation (9) and equation (10), in which $\delta(k, j)$ is the Kronecker delta.

$$u_k = \lambda I_t^A(k) \sum_{i|k\in\omega_i} \frac{\tau_i}{\gamma_i+\lambda} \quad (9)$$

$$s_{k,j} = \sum_{i|(k\in\omega_i \cap j\in\omega_i)} \left(\delta(k,j) - \frac{I_t^A(k) I_t^A(j)}{\gamma_i+\lambda}\right) \quad (10)$$

The relighting result I_t^B can be directly solved from the linear system in equation (8). The convex nature of the original optimization problem ensures that equation (8) is solvable. The linear system is large in scale because the number of equations equals the number of pixels in the target image. However, close observation on equation (10) reveals that matrix S is not only symmetric but also very sparse. Most of the elements in S are zero except those whose index k and j correspond to pixels that can be covered by one single local window. For example, if the local window ω is of the size 3×3 pixels, each row (or column) of S contains no more than 25 nonzero elements. Solving large scale sparse linear systems has been very extensively studied. In our implementation, we use the PARDISO solver [24,25,26,27].

The following list summarizes major steps of the proposed face relighting algorithm. The algorithms inputs are: two reference face images I_r^A and I_r^B, and one target face image I_t^A to be relit.

1. Fit an AAM model to the three images for locating facial landmarks.
2. Warp the two references images to the target face image.
3. Calculate a ratio image T between the two warped reference images.
4. Apply a Gaussian filter with standard deviation σ to T.

5. Construct vector U and matrix S according to equations (9) and (10).
6. Solve the linear system in equation (8) to get the relighting result I_t^B.
7. If the inputs are color images, steps 1-6 are performed on their gray-value component. The final result is generated by composing the original hue and saturation components of I_t^A, and I_t^B as the new gray-value component.

5 Experimental Results and Comparisons

We repeat the experiment shown in Fig. 1 using the proposed face relighting algorithm. The result is shown in Fig. 3. Comparing to Fig. 1(g) and 1(h), the new relighting result shown in Fig. 3(b) is much smoother and is free of high frequency artifacts. Fig. 3(e) shows the chin area after relighting. The effect of the beard in the reference images is no long visible. The relighting result seems realistic and natural. At the same time, textures of the target face are very well preserved.

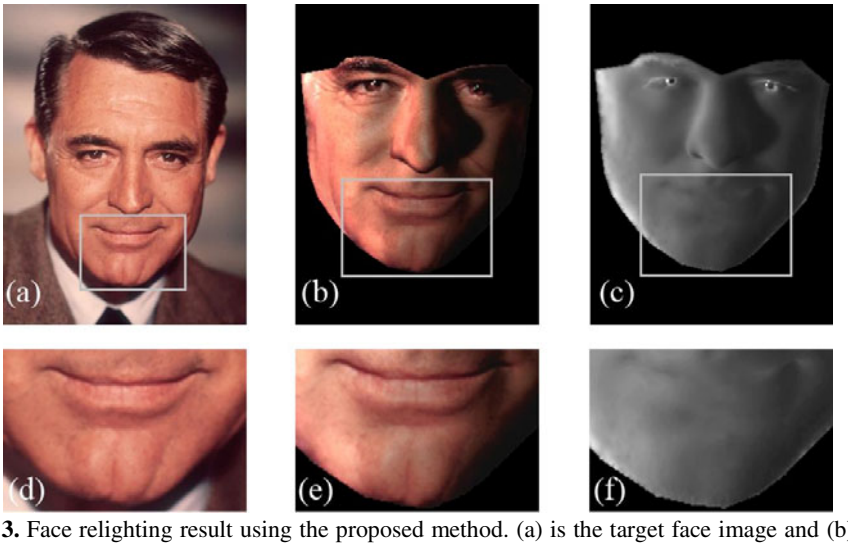


Fig. 3. Face relighting result using the proposed method. (a) is the target face image and (b) is the relighting results. (c) is the local relighting coefficients. (d)-(f) are the gamma-enhanced detail of the chin areas of images (a)-(c).

The local relighting coefficients α_i are very interesting. They are necessary for the problem analysis and formation, but are implicit when solving the relighting result I_t^B. Actually, when I_t^B is known, α_i can be calculated using equations (5) and (6). Fig. 3(c) shows the α_i thus calculated. Fig. 3(c) is similar to Fig. 1(e) except that it is much smoother. During the optimization, the first term on the right side of equation (1) forces the smoothness of α_i, while the second term pushes their values towards τ_i. In fact, Fig. 3(c) resembles the 'ideal' ratio images achieved in [8] very much. More relighting results are shown in Fig. 4, which also demonstrates the robustness of the proposed method towards expression change, accessory, skin color, small pose variation, and the choice of reference images.

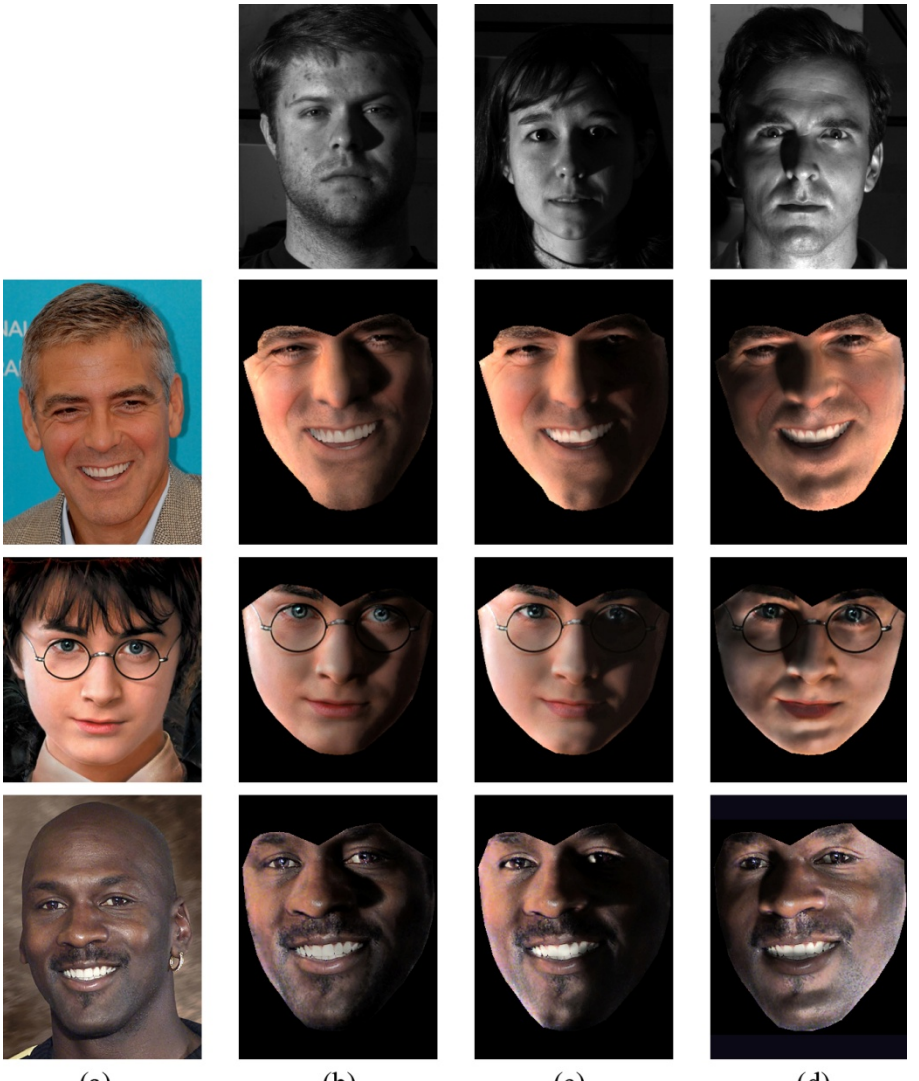


Fig. 4. More face relighting results. Images in column (a) are the target face images to be relit. The top most images in columns (b)-(d) are the I_r^B reference images from Yale database. Their corresponding frontal illumination images are used as I_r^A reference images. Relighting results are shown in the bottom three rows of (b)-(d). (The image of Harry Potter, acted by Daniel Radcliffe, is available at http://languageisavirus.com/harry-potter/layouts/harry-potter/harry-potter-chamber-of-secrets.jpg. The image of Micheal Jordan is available at http://www.mkphoto.net/i/people/sjordanface.jpg.)

There are mainly three parameters in the proposed algorithm: the Gaussian filter parameter σ, the balancing weight λ, and the size of the shifting window ω. The Gaussian filter helps to suppress high frequency artifacts or noises in the ratio image. Larger σ indicates a stronger low pass filtering effect which will also blur textures especially the edges. The weight λ balances the efforts for preserving image smoothness and texture, with the resemblances of illumination changes in the relighting result. An extreme case is that when $\lambda = 0$, the solution of the optimization problem is trivial. $\mathcal{F}$ is minimized when $\alpha_i = 0$, so that $I_t^B = I_t^A$. As such, the texture and smoothness of the target image are literally completely preserved, but no relighting effect is created. The size of the shifting window ω decides to what extent local constraints will be propagated to its neighbors. It also affects the sparsity of the linear system in equation (8). The larger the local window is, the less sparse the linear system will be, and the solving process will thus take longer time. Under the proposed optimization framework, the relighting result is not very sensitive to parameter choices. Empirical results show that our method can generate realistic relighting result for a wide range of parameter values. Fig. 5 compares the relighting results of Carey Grant under different parameter settings. To facilitate implementation, we empirically recommend that: $\sigma \in [0.2,\ 2.0]$, $\lambda \in [0.2,\ 1.5]$ and ω is 3×3 pixels or 5×5 pixels. Be default, we use $\sigma = 0.5$, $\lambda = 0.5$ and ω is 3×3 pixels.

$\sigma = 1.5$; $\lambda = 0.5$; ω=3×3 pixels | $\sigma = 0.5$; $\lambda = 1.5$; ω=3×3 pixels | $\sigma = 0.5$; $\lambda = 0.5$; ω=5×5 pixels | $\sigma = 1.0$; $\lambda = 0.2$; ω=3×3 pixels

Fig. 5. Face relighting results using different parameters

In the proposed method, the AAM fitting, Guassian filtering, image warping steps can be done very fast. The most time consuming step is to solve the large linear system in equation (8). We test the running time on a desktop PC equipped with a Pentium D 3.4GHz CPU and 2GB memory. Table 1 profiles the running time of the proposed method for input images of different sizes.

By including the reference images I_r^A and I_r^B in the bootstrap set, the quotient image based method proposed in [3] can also be applied to the face image relighting problem discussed in this paper. We test three bootstrap set configurations: i) single reference object without image warping (three images including I_r^A and I_r^B); ii) single reference object with image warping (three images including $L(I_r^A)$ and $L(I_r^B)$); iii) ten reference objects with image warping (thirty images including $L(I_r^A)$ and $L(I_r^B)$).

Table 1. Running time of the proposed method

Image Size (pixels)	Size of ω (pixels)	Total time (seconds)	Time for linear system solving (seconds)
200x200	3x3	2.1	1.5
300x300	3x3	6.9	5.8
400x400	3x3	12.5	10.2
200x200	5x5	5.7	5.1
300x300	5x5	19.8	18.7
400x400	5x5	45.4	43.1

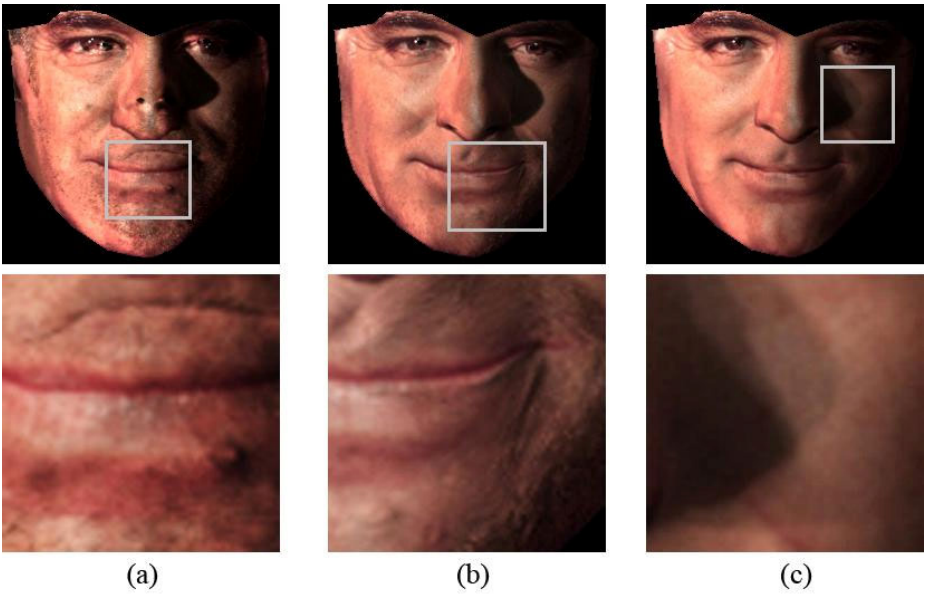


Fig. 6. Quotient image based relighting [3]. Columns (a)-(c) correspond to the bootstrap set configurations i)-iii). Images in the second row are gamma-enhanced details.

The relighting results are shown in Fig. 6 correspondingly. Without dense correspondence and image warping, there are a lot of artifacts caused by misalignments in Fig. 6(a). The formation of high frequency artifacts shown in Fig. 6(b) is similar to what we have explained for Fig. 1 and Fig. 2 in section 3. The best relighting result, which is shown in Fig. 6(c), is achieved when multiple bootstrap objects are used. However, cast shadows from multiple objects overlap with each other causing unrealistic relighting effects. This is due to the fact that cast shadows are ignored in the basic assumptions of the quotient image method [3].

6 Conclusions

Face image relighting is an interesting problem and has wide range applications in face recognition and film production. We study the face relighting problem in which reference face images are available. Based on the empirical fact that common radiance environments are locally homogeneous, we propose to use an optimization based solution in which local linear adjustments are performed on overlapping windows throughout the input image. The local constraints help to preserve texture information of the input image, and the global optimization ensures the overall image smoothness during the illumination transformation. We have demonstrated the effectiveness of our method by applying it to challenging real life face images. Experimental results show that our method is able to generate photo-realistic relighting effects. Also, the robustness of our method is ensured by its convex optimization nature.

The proposed method assumes that the reference face and the target face are similar in their shapes after image warping. Deviations from such an assumption lead to 'shape artifacts' in the relighting results. Apply certain strategies in the reference face selection may be a promising solution. Also, The AAM fitting may fail on images under harsh lighting conditions, and the overall running time is not satisfactory for large input images. All these problems require further research efforts.

Acknowledgments

The work in this paper was substantially supported by a Key Project of the Ministry of Public Security of China: "Processing and Validation of Digital Image and Video" (2005ZDGGQHDX005), a National 973 Project: "Fundamental Research on Multi-Domain Collaboration for Broadband Wireless Communications" (2007CB310600), and a research grant from Tsinghua University (053207002).

References

1. Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., Sagar, M.: Acquiring the Reflectance Field of a Human Face. In: Proceeding of ACM SIGGRAPH, pp. 145–156 (2000)
2. Marschner, S.R., Greenberg, D.P.: Inverse Lighting for Photography. In: Proceedings of Fifth Color Imaging Conference, pp. 262–265 (1997)
3. Shashua, A., Raviv, T.R.: The Quotient Image: Class-Based Re-Rendering and Recognition with Varying Illuminations. IEEE Trans. on PAMI 23(2), 129–139 (2001)
4. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: Proceedings of ACM SIGGRAPH, pp. 497–500 (2001)
5. Basri, R., Jacobs, D.W.: Lambertian Reflectance and Linear Subspaces. IEEE Trans. on PAMI 25(2), 218–233 (2003)
6. Zhang, L., Wang, S., Samaras, D.: Face synthesis and recognition from a single image under arbitrary unkown lighting using a spherical harmonic basis morphable model. In: Proceedings of IEEE CVPR, pp. 209–216 (2005)
7. Wang, Y., Liu, Z., Hua, G., Wen, Z., Zhang, Z., Samaras, D.: Face Re-Lighting from a Single Image Under Harsh Lighting Conditions. In: Proceedings of IEEE CVPR (2007)

8. Peers, P., Tamura, N., Matusik, M., Debevec, P.: Post-production Facial Performance Relighting using Reflectance Transfer. In: Proceedings of ACM SIGGRAPH (2007)
9. Marschner, S.R., Guenter, B., Raghupathy, S.: Modeling and Rendering for RealisticFacial Animation. In: Proceedings of Eurographics Rendering Workshop, pp. 231–242 (2000)
10. Stoschek, A.: Image-based re-rendering of faces for continuous pose and illumination directions. In: Proceedings of IEEE CVPR, pp. 582–587 (2000)
11. Liu, Z., Shan, Y., Zhang, Z.: Expressive expression mapping with ratio images. In: Proceedings of ACM SIGGRAPH, pp. 271–276 (2001)
12. Wen, Z., Liu, Z., Huang, T.: Face relighting with radiance environment maps. In: Proceedings of IEEE CVPR, pp. 157–165 (2003)
13. Ramamoorthi, R., Hanrahan, P.: On the Relationship between Radiance and Irradiance: Determining the Illumination from Images of Convex Lambertian Object. J. Optical Soc. Am. 18(10), 2448–2459 (2001)
14. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: Proceedings of ACM SIGGRAPH, pp. 497–500 (2001)
15. Lee, K.C., Ho, J., Kriegman, D.: Nine Points of Light: Acquiring Subspaces for Face Recognition under Variable Lighting. In: Proceedings of IEEE CVPR, pp. 357–362 (2001)
16. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. IEEE Trans on PAMI 23(6), 681–685 (2001)
17. Matthews, I., Baker, S.: Active Appearance Models Revisited. Int'l J. Computer Vision 60(2), 135–164 (2004)
18. Donner, R., Reiter, M., Langs, G., Peloschek, P., Bischof, H.: Fast Active Appearance Model Search Using Canonical Correlation Analysis. IEEE Trans. on PAMI 28(10), 1690–1694 (2006)
19. Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. on PAMI 23(6), 643–660 (2001)
20. Gao, W., Cao, B., Shan, S., Chen, X., Zhou, D., Zhang, X., Zhao, D.: The CAS-PEAL Large-Scale Chinese Face Database and Baseline Evaluations. IEEE Trans. on SMC, Part A 38(1) (2008)
21. Bookstein, F.L.: Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. IEEE Trans. on PAMI 11(6) (1989)
22. Levin, A., Lischinski, D., Weiss, Y.: A Closed Form Solution to Natural Image Matting. IEEE Trans. on PAMI 30(2), 1–15 (2008)
23. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
24. Schenk, O., Gärtner, K.: Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO. J. of Future Generation Computer Systems 20(3), 475–487 (2004)
25. Schenk, O., Gärtner, K.: On fast factorization pivoting methods for symmetric indefinite systems. Elec. Trans. Numer. Anal. 23, 158–179 (2006)
26. Schenk, O., Bollhoefer, M., Roemer, R.: On large-scale diagonalization techniques for the Anderson model of localization. SIAM Review 50, 91–112 (2008)
27. Schenk, O., Waechter, A., Hagemann, M.: Matching-based Preprocessing Algorithms to the Solution of Saddle-Point Problems in Large-Scale Nonconvex Interior-Point Optimization. J. of Comput. Opt. and App. 36(2-3), 321–341 (2007)

Correlation-Based Intrinsic Image Extraction from a Single Image

Xiaoyue Jiang[1], Andrew J. Schofield[1], and Jeremy L. Wyatt[2]

[1] School of Psychology, University of Birmingham, Birmingham, B15 2TT, UK
[2] School of Computer Science, University of Birmingham, B15 2TT, UK
{x.y.jiang,a.j.schofield}@bham.ac.uk, jlw@cs.bham.ac.uk

Abstract. Intrinsic images represent the underlying properties of a scene such as illumination (shading) and surface reflectance. Extracting intrinsic images is a challenging, ill-posed problem. Human performance on tasks such as shadow detection and shape-from-shading is improved by adding colour and texture to surfaces. In particular, when a surface is painted with a textured pattern, correlations between local mean luminance and local luminance amplitude promote the interpretation of luminance variations as illumination changes. Based on this finding, we propose a novel feature, local luminance amplitude, to separate illumination and reflectance, and a framework to integrate this cue with hue and texture to extract intrinsic images. The algorithm uses steerable filters to separate images into frequency and orientation components and constructs shading and reflectance images from weighted combinations of these components. Weights are determined by correlations between corresponding variations in local luminance, local amplitude, colour and texture. The intrinsic images are further refined by ensuring the consistency of local texture elements. We test this method on surfaces photographed under different lighting conditions. The effectiveness of the algorithm is demonstrated by the correlation between our intrinsic images and ground truth shading and reflectance data. Luminance amplitude was found to be a useful cue. Results are also presented for natural images.

1 Introduction

In standard imagery, pixel intensities depend on both the reflectance properties of objects in the scene and its illumination conditions. Images representing these underlying properties are called intrinsic images [1]. The extraction of such images can improve many computer vision methods such as: object recognition, light source estimation and shape-from-shading.

The extraction of intrinsic images is an ill-posed problem. A variety of cues have been proposed to constrain this problem. Early approaches were based on the Retinex theory of lightness constancy in humans [2]. This theory rests on the assumption that lighting changes are smooth whereas reflectance changes are abrupt; this difference can be used to distinguish illumination from reflectance. However, abutting flat surfaces at different orientations in a 3D world produce

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 58–71, 2010.

abrupt changes in illumination leading Shina and Adelson [3] to propose a 2-stage process wherein luminance junctions are classified as illumination or reflectance using local heuristics and then reclassified if necessary by a global analysis that reconstructs 3D shapes. The method works well in stylised stimuli where edges are easily defined. With supervised learning, Bell and Freeman [4] reconstructed shading and reflectance from classified steerable filter coefficients.

Another common approach is to use colour as a key for identifying illumination gradients, based on the assumption that hue is illumination invariant (see for example, [5], [6] and [7]). An illumination map can then be derived by reintegrating only those gradients that arise from illumination. However, hue is not entirely illumination-invariant: outdoor shadows are tinted blue [8] and hue is poorly specified in dark shadows. In addition hue based methods can be confused by small image features, although Tappen et al. [7] provides a reasonable solution to this problem by training a classifier to distinguish shadow and reflectance edges. Finlayson et al.'s colour based method [9] defines an illumination-invariant colour space to discriminate shadows from reflectance variations, but this requires a calibrated camera. Further, since humans can distinguish shadows from reflectance changes in monochrome images [10] colour cannot be the only cue that enables such a separation.

Like hue, certain texture properties are also invariant to illumination and therefore a potential cue for deriving intrinsic images. Shen et al. [11] applied texture consistency as a constraint for decomposing images into shading and reflectance. This algorithm identifies groups of pixels sharing illumination invariant texture features and adjusts those aspects that are not illumination invariant until the groups are more consistent. The result is to discount illumination to produce a reflectance map. However, this method uses a computationally expensive optimization procedure. Finally, intrinsic images can be extracted from image sequences using multiple images to constrain the problem [12,13,14]. Such algorithms produce good results but are limited by the need for multiple images.

We introduce a new algorithm to extract intrinsic images from single images. The method combines colour and texture with a new metric (luminance amplitude, see Section 2) which is used by humans to differentiate shading and reflectance [15]. We combine these metrics with a steerable filter decomposition and use inter-cue correlations to identify frequency/orientation components that belong to the shading and reflectance maps respectively. We test this approach on images for which we have ground truth data and compare our novel luminance amplitude cue with the more established texture and colour cues.

2 Basic Cues

Luminance amplitude: Assuming Lambertian reflectance, the intensity value $I(x, y)$ of every pixel in an image is the product of the incident lighting $L(x, y)$ and the reflectance $R(x, y)$ at that point: $I(x, y) = R(x, y) \times L(x, y)$. For surfaces with a painted texture we can measure the mean and variance (luminance amplitude) of pixel intensities in local regions. If a change in local mean intensity

is caused by a change in illumination then luminance amplitude should vary in the same direction. This occurs because illumination multiplies light and dark reflectances in the texture by a common factor. Thus correlated changes in mean luminance and luminance amplitude indicate illumination changes.

Colour and Texture: Unlike illumination changes, reflectance variations are characterized by complex variations in pixel values based on a number of potential cues. If the pattern of a texture changes (e.g. a change in granularity or dominant orientation), then any associated change in mean intensity, over a large enough patch, might reasonably be regarded as a change in surface reflectance $R(x, y)$. Colour is another diagnostic feature for reflectance changes. If hue and intensity vary together this is likely to signal a reflectance change. Thus positive correlations between colour and luminance or texture and luminance indicate reflectance changes.

3 Steerable Filter Based Feature Extraction

Our algorithm is based on the relationships between intensity, luminance amplitude, texture and colour, as described above. The overall framework for the algorithm is shown in Fig. 1. We use steerable filters to decompose the image into its constituent orientation/frequency bands. These filters provide a general framework that can decompose images and completely reconstruct the originals from the resulting components [16] or, as here, construct partial images from selected components. We apply the steerable filter bank S_L to the raw luminance values (luminance modulations LM) extracting a full set of luminance components $(LM_{ij}, (i = 1, ..., N; j = 1, ..., M)$, where N is the number of orientations, and M is the number of frequency bands in S_L). We also apply the filter bank to estimate variations in local amplitude (AM), texture (TM), and hue (HM). We then calculate the correlation between LM and AM, TM and HM in each orientation/frequency band. If a component of LM is positively correlated to AM but not TM or HM, it is deemed to convey shading information. The illumination

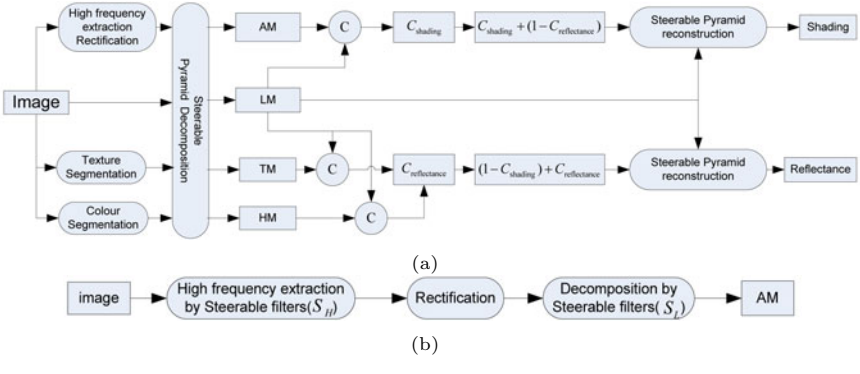


Fig. 1. (a) Flowchart of the overall algorithm, (b) Flowchart of AM extraction

(or shading) image is then constructed from only these components. Similarly, components of LM that are positively correlated to TM and HM but not AM are used to construct the reflectance image.

3.1 Extraction of AM

The equation for image intensity ($I = R \times L$) shows that, for fixed reflectance, intensity increases with increased illumination (L). Assuming that lighting is constant in a local region but that reflectance varies due to texture, convolving both sides of the lighting equation with a high-frequency filter $F_h(\cdot)$ produces the equation $F_h(I) = F_h(R) \times L$. Large scale variations in L will modulate the magnitude of the filter responses such that the envelope of $F_h(I)$ will be determined by L. Rectifying the output of $F_h(I)$ will demodulate this envelope which can be processed by further filtering with $F_l(\cdot)$,

$$F_l(r(F_h(I))) = F_l(r(F_h(R) \times L)) \tag{1}$$

where $r(x) = abs(x - \mu_x)$ is the rectification of signal x. The low-frequency filter $F_l(\cdot)$ detects low-frequency information in $r(F_h(I))$, hence Eq.1 can be written as $F_l(r(F_h(I))) \approx F_l(r(L))$. The response of $F_l(r(F_h(I)))$ is a measure of local amplitude which, if the texture is uniform, will be correlated with illumination.

In practice, the high-frequency part of the input image I_H is extracted by the steerable filters: reconstructing I_H from only the high-frequency responses. We then rectify I_H about its mean value, i.e. $r(I_H) = abs(I_H - \mu_{I_H})$, and apply steerable filters (S_L), as used for LM, to decompose $r(I_H)$ yielding components of AM that match those extracted for LM. The flowchart for extracting AM and an example AM component are shown in Fig. 1(b) and Fig. 2(c), respectively.

3.2 Extraction of TM

Texture modulation (TM) should represent transitions between different texture patterns. The extraction of TM relies on texture segmentation; a difficult problem in itself which we do not attempt to solve in full here. However, for our purposes segmentation based on Gabor-features works well. We calculate the Gabor responses to the images in different orientation and frequency bands, and then use the principal component analysis (PCA) to extract common texture features. Next we use fuzzy clustering to classify the responses into non-continuous regions of similar texture. We then create a simple texture map by block filling texture regions with their own mean intensity, see Fig. 2(f), before applying steerable filters (S_L) to extract components matching the LM signals, see Fig. 2(d).

3.3 Extraction of HM

Colour (more specifically hue) is an important feature for estimating reflectance as it is, more-or-less, illumination invariant. We derive a 4-dimentional intensity-free colour vector,

$$F_{colour}(x, y) = (r_{xy}/\|I_{xy}\|, g_{xy}/\|I_{xy}\|, b_{xy}/\|I_{xy}\|, h_{xy}) \tag{2}$$

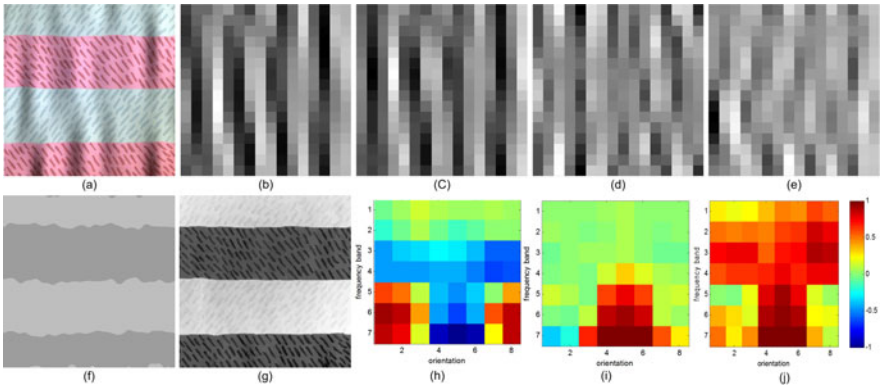


Fig. 2. Intermediate processing: (a) the input image, (b) one LM component of (a), (c-e) the matching AM, TM and HM components respectively, (f-g) texture and colour maps for (a), (h-j) pictorial representation of correlations between LM and AM, TM and HM respectively; x-axis shows component orientation, y frequency, cell colour indicates correlation coefficients

where (r_{xy}, g_{xy}, b_{xy}) represent RGB values, $\|I_{xy}\| = \sqrt{r_{xy}^2 + g_{xy}^2 + b_{xy}^2}$ is the norm of the RGB triple, and h_{xy} is hue. We apply the mean shift algorithm [17] to this vector so as to achieve colour segmentation, block filling each region in the hue map according to its mean luminance, see Fig. 2(g). Once again steerable filters (S_L) are used to extract HM components from the hue segmentation results, see Fig. 2(e).

4 Reconstruction of Shading and Reflectance

Extracting intrinsic images makes the implicit, but not always appropriate, assumption that luminance variations are either due to reflectance or illumination but not both. Our algorithm is based on this assumption but not bound by it. By estimating the correlation between each LM component and reflectance and shading respectively we can assign individual components to shading and reflectance in some proportion. Correlation coefficients C_{ij}^{la}, between the corresponding components LM_{ij} and AM_{ij} are used to measure the relationship between luminance and luminance amplitude, see Fig. 2(h). If LM_{ij} and AM_{ij} are positively related $(C_{ij}^{la} > 0)$ we presume that LM_{ij} results from illumination and retain some proportion of it when reconstructing the illumination map. If $C_{ij}^{la} < 0$, LM_{ij} is used in the reflectance map. Hence we rename C_{ij}^{la} as C_{ij}^{shd} because when positive this measure places the component into the shading map.

The correlation coefficient C_{ij}^{lt} between component LM_{ij} and TM_{ij} describes the relationship between luminance and texture. If $C_{ij}^{lt} > 0$, LM_{ij} should be treated as a reflectance change and used in the reflectance map. Because TM is extracted from texture blocks, see Fig. 2(f), its low-frequency components are more reliable than its high frequency components. We use the frequency index j

to weight coefficients C_{ij}^{lt} to reflect their reliability. The updated coefficient $\tilde{C}_{ij}^{lt}$ between LM_{ij} and TM_{ij} is,

$$\tilde{C}_{ij}^{lt} = C_{ij}^{lt} \times \frac{j}{M+1} \tag{3}$$

where $j = M$ indicates the lowest frequency band. Similarly, the correlation coefficient C_{ij}^{lh} between LM_{ij} and HM_{ij} measures the relationship between luminance and hue changes. If $C_{ij}^{lh} > 0$, LM_{ij} should be included in the reflectance map. As texture and colour are both positively related to reflectance changes we can combine their relationship with LM into a single measure (C_{ij}^{ref}) as follows,

$$C_{ij}^{ref} = \frac{\tilde{C}_{ij}^{lt} + C_{ij}^{lh}}{2} \tag{4}$$

In order to correctly divide LM components between shading and reflectance we need to consider their relationship with both properties. If we treat local amplitude, texture and colour as equally reliable, then the correlation coefficients for reconstructed shading ($C_{ij}^{rec\_shd}$) and reflectance ($C_{ij}^{rec\_ref}$) are given by Eq.5 and Eq.6, respectively.

$$C_{ij}^{rec\_shd} = C_{ij}^{shd} + (1 - C_{ij}^{ref}) \tag{5}$$

$$C_{ij}^{rec\_ref} = (1 - C_{ij}^{shd}) + C_{ij}^{ref} \tag{6}$$

However, because we cannot decompose the original image into infinitely narrow orientation and frequency bands, and because, in real lighting situations, some variations in intensity are caused by reflectance and shading together, some components LM_{ij} will strongly correlate with both reflectance and shading. In this situation, we need to assign a weight to each correlation coefficient according to the reliability of the texture and colour segmentation results. That is,

$$C_{ij}^{rec\_shd} = wC_{ij}^{shd} + (1 - w)(1 - C_{ij}^{ref}) \tag{7}$$

$$C_{ij}^{rec\_ref} = w(1 - C_{ij}^{shd}) + (1 - w)C_{ij}^{ref} \tag{8}$$

$$w = \frac{T_S}{T_S + T_R/k} \tag{9}$$

where T_R ($T_R \subset [0, 1]$) estimates the reliability of texture and colour segmentations and $T_S = 1 - T_R$ the reliability of the amplitude modulations. Although texture and colour segmentation are based (in principle) on illumination-invariant features, illumination changes can still influence the segmentation results. Therefore we use an illumination parameter k to adjust the reliability of these cues. k is the image's *key* value which is given by the global contrast of an image [18],

$$k = \frac{L_{max} - L_{av}}{L_{max} - L_{min}} \tag{10}$$

where L_{av}, L_{min} and L_{max} are the logarithmic average, minimum and maximum of the luminance respectively. More extreme (harsh, high contrast) lighting conditions produce bigger key values down-weighting texture and colour. Fig. 3(b) shows the key values of images under different lighting conditions.

After deciding the correlation coefficients for every LM component, estimates of shading and reflectance can be reconstructed as follows:

$$I_{shd} = S_L \otimes \begin{cases} C_{ij}^{rec\_shd} \times LM_{ij} & \text{if } C_{ij}^{rec\_shd} > 0 \\ 0 & \text{if} C_{ij}^{rec\_shd} \leq 0 \end{cases} \tag{11}$$

$$I_{ref} = S_L \otimes \begin{cases} C_{ij}^{rec\_ref} \times LM_{ij} & \text{if } C_{ij}^{rec\_ref} > 0 \\ 0 & \text{if} C_{ij}^{rec\_ref} \leq 0 \end{cases} \tag{12}$$

where $\otimes$ is the reconstruction of steerable filters S_L with weighted LM_{ij}. Due to the self-inverting characteristics of steerable filters, the same filters can be used for decomposition and reconstruction [16]. Correlation coefficients $C_{ij}^{rec\_shd}$ and $C_{ij}^{rec\_ref}$ determine how much each component will contribute to the relevant intrinsic image. More positive correlations produce stronger weights, but negative correlations produce zero weights.

5 Post-processing of the Reconstructed Images

As will be shown in Section 6, the reconstruction process described above is reasonably effective. However it is not perfect and we now outline some post processing steps that improve the final results.

5.1 DC Component of Shading Image

During the reconstruction process some LM components will be set to zero. Therefore the resulting images may lose their DC value. Although this DC value will not influence the overall appearance of the reconstructed images, it may affect subsequent processing. An alternative estimate for the reflectance image can be derived from the shading image as follows:

$$I_{\text{dRef}} = I_{\text{org}} / I_{shd} \tag{13}$$

where I_{org} is the original image. Therefore the problem of calculating the DC component can be transferred to an assessment of the reflectance image I_{dRef}. If the initial estimate of shading is accurate enough, then I_{dRef} should convey uniform intensity distributions within each texture class. Thus an evaluation of the texture consistency within I_{dRef} can be used as a cost function to optimize I_{shd}. The optimization problem is

$$E_{shd}(V_{DC}) = \arg\min_{T_i} \sum_i F_{txt}\left(\frac{I_{\text{org}}}{I_{\text{shd}} + V_{DC}}, T_i\right) \tag{14}$$

where V_{DC} is the DC value to be optimized. The function $F_{txt}(I, T_i)$ represents texture consistency evaluated for image I against the texture segmentation results $T_i (i = 1, ..., p)$, where p is the number of textures in the image I. We

model the texture distribution as a normal distribution $N(\mu_i, \sigma_i)$, hence texture consistency is defined as

$$F_{txt}(\hat{I}_{\text{dRef}}, T_i) = -\frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{(\hat{I}_{\text{dRef}}(T_i) - \mu_i)^2}{2\sigma_i^2}) \tag{15}$$

where $\hat{I}_{\text{dRef}} = I_{org}/(I_{shd} + V_{DC})$; $\hat{I}_{\text{dRef}}(T_i)$ is the region that belongs to texture T_i in image $\hat{I}_{\text{dRef}}$. Texture distributions are sampled from the most reliable regions of $\hat{I}_{dRef}$ where the corresponding shading value I_{shd} is high. The resulting shading image $\hat{I}_{shd} = I_{shd} + V_{DC}$ is a better estimate of true shading than I_{shd}.

5.2 Compensation of Reflectance in Regions of Deep Shadow

When the frequency of an AM component is greater than that of any shading in the image, it will represent small variations in reflectance not shading. Thus LM tends not to correlate with AM in high-frequency bands, but LM will still correlate well with TM and HM in these bands. Therefore high frequency components tend to be allocated to the reflectance image rather than the shading image. However, shadows tend to suppress the luminance range of textures. Looking ahead, this is seen in Fig. 5(d) where textured areas of the reconstructed reflectance image are slightly erased in regions that were in shadow (cf Fig. 2(a)).

We solve the above problem by locally amplifying the responses of high-frequency components using the optimised shading ($\hat{I}_{shd}$, Eq.14) to guide the adjustment of each LM_{ij} component separately. The compensation for LM_{ij} is,

$$\hat{LM}_{ij} = LM_{ij} \times F_{adj}(\alpha, \beta, t) = LM_{ij} \times \frac{t(\alpha + \beta)}{\alpha \hat{I}_{shd} + \beta} \tag{16}$$

where $\tilde{I}_{shd}$ is produced by normalizing $\hat{I}_{shd}$ to the range [0, 1]. The adjustment function $F_{adj}(\alpha, \beta, t)$ is in the range of $[t(\alpha/\beta+1), t]$, such that the darkest part is re-scaled to $t(\alpha/\beta+1)LM_{ij}$ and the lightest part to tLM_{ij}. Parameters α and β ($\alpha > 0, \beta > 0$) control the adjustments for the dark and light pixels, t controls the overall range of the adjustment. The objective function for optimizing these parameters is

$$E_{ref} = \arg\min \sum_k F_{txt}(\hat{LM}_{ij}, T_k) + \lambda E_{\text{cst}}(F_{adj}) \tag{17}$$

where $F_{txt}(\hat{LM}_{ij}, T_k)$ evaluates texture consistency as defined in Eq. 15. For each texture T_k, $\hat{LM}_{ij}$ is modelled as a normal distribution $N(\mu_k^{ij}, \sigma_k^{ij})$ based on the more reliable regions where $\hat{I}_{shd}$ is high (light regions). The function $E_{\text{cst}}(F_{adj})$ constrains the maximum rescaling produced by the adjustment function. If the estimated shading map is good, we only need to apply small adjustments and can constrain the ratio between α and β as Eq. 18. The interior-point algorithm [19] is used to solve this constrained optimization problem.

$$E_{\text{cst}}(F_{adj}) = \alpha/\beta \tag{18}$$

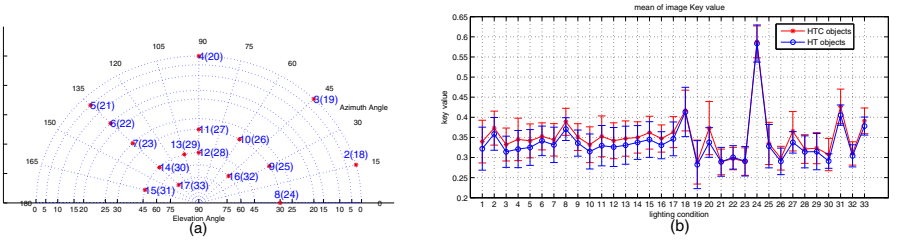


Fig. 3. (a) Asterisks show spotlight positions, numbers represent spot+diffuse conditions (those in brackets spotlight only conditions). (b) Mean key values for HT and HTC surfaces indexed by lighting condition.

6 Experimental Results

6.1 Test Set

In order to evaluate the proposed algorithm quantitatively we devised a test set containing images of 10 surfaces (5 surface shapes with two paint treatments) photographed under 33 lighting conditions. Surfaces were milled into small (57 x 64 mm) plastic blocks using a CNC Milling machine (Modella JWX-10, Roland Inc, Japan) and ArtCAM design software (Delcam plc, UK) providing multiple examples of the same surface. Surface profiles included highly coherent sinusoidal corrugations, random but oriented ripples and isotropic bumps, see Fig. 2 and 7. Two paint treatments (HTC and HT surface as described below) were applied using matte paints. Both treatments consisted of broad red and green stripes each textured with thin elements oriented differently for the two colours. For hue, texture, contrast (HTC) surfaces the darker green paint was chosen such that the green regions had lower contrast but higher mean reflectance than the red regions. For hue, texture (HT) surfaces the contrasts of the green and red regions were more similar.

Surfaces were places into the centre of a 1m diameter integrating sphere; standing vertically and facing forward towards a pair of cameras placed either side of azimuth angle 90^o, see Fig. 3(a). Here we use images from the right camera only. A bright white backlight (composed of 24 wide angle, 7 lm LEDs; NSPWR70BS, Nichia Inc, Japan) placed in the wall of the sphere behind the surfaces produced uniform diffuse illumination via reflections off the white internal surface of the sphere. A baffle placed behind the surface ensured that the backlight did not shine directly into the cameras. We placed an array of individual high brightness (29 cd), spotlight LEDs (Nichia NSPW500DS) at specific locations in the wall of the sphere, facing the surface. Only 16 of the spotlights were used in this study located as shown in Fig. 3(a). The spotlights produced a bluer light than the backlight.

We first photographed the objects under the diffuse light only (condition 1), then under the diffuse light with each of the spotlights in turn, conditions $l (l = 2, ..., 17)$. Finally we used each spotlight alone, conditions $l + 16$. We also produced a matte grey version of every surface to provide ground truth shading images. Ground truth reflectance images were obtained by dividing images taken

under the diffuse light by the shading ground truth for each surface. Images were taken from a larger database (http://www.bold.bham.ac.uk).

6.2 Evaluation of Extraction Method and Cue Combinations

We extracted intrinsic images for the test set while using different cues within the algorithm. We used correlations between the estimated shading or reflectance images and their respective ground truth images as a metric for assessing results. Fig. 4(a), (b) show the performance of different cues for the HTC treatment condition under all 33 lighting conditions, where H, T and A indicate that HM, TM and AM streams were 'turned on' respectively. The combination of all three cues (HTA in Fig. 4(a), (b)) is better than any cue alone or any combination of two cues. The results show that local amplitude (A in Fig. 4(a), (b)) has an important role in detecting illumination changes. When it is combined with either hue or texture (HA and TA in Fig. 4(a), (b)), it boosts performance relative to either of these cues alone.

Fig. 4(c) further summarize the results. It shows improved extraction of shading from diffuse+spot images as streams are activated. Illumination changes are much weaker in these images. Consequently, small correlation coefficients between cues make the assignment of LM components less accurate when only using one or two cues. Fig. 4(a) shows marked differences in performance for individual light sources. The poor performance for images under frontal spotlights (azimuth 90^o, conditions 20, 27 & 28) is caused by the same problem.

Conversely reflectance images are very good for diffuse+spot lighting and less good for spotlights only, see Fig. 4(c). We might expect reflectance estimates to be best for images that contain relatively little shading (Fig. 4(b) conditions 1-17)

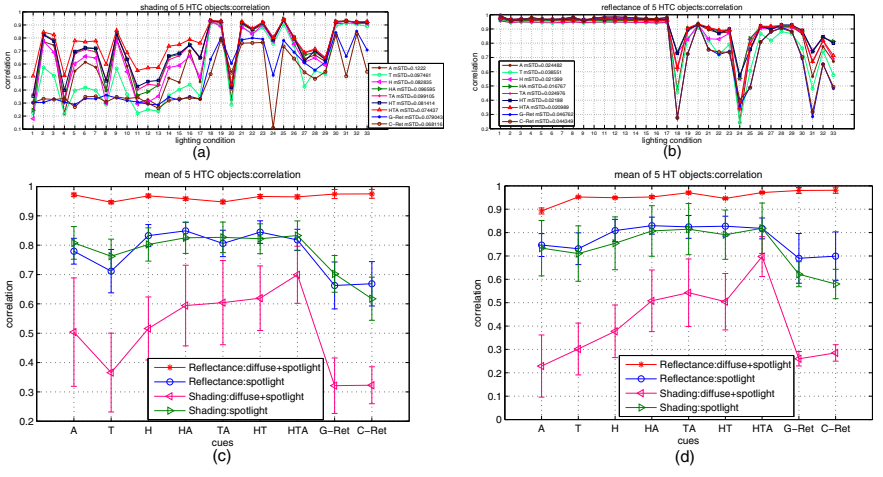


Fig. 4. Performance of different cues. Mean correlation against ground truth images for (a) shading and (b) reflectance estimates, which is the average of 5 HTC objects.(mSTD is the mean standard deviation of 5 objects across 33 lighting conditions). (c),(d) are correlation results on HTC and HT images respectively.

as only slight adjustments are needed in this case. Reflectance estimates are very poor (especially when only texture is used) for spotlights directed from the side (conditions 18, 24, & 31). These images are a special case where the light only glances on the surface much of which is in shadow, almost erasing any texture present. Fig. 4(d) summaries the results on HT surfaces. As we should expect AM alone worked better for HTC surfaces, where texture contrast and hence AM was negatively correlated with gross reflectance (LM) across texture boundaries, than HT surfaces where texture contrast did not vary. Combining cues remains helpful in this situation. Fig. 4(c),(d) also show the results of processing our stimuli with the grey-Retinex (G-Ret) and colour-Retinex (C-Ret) algorithms [20]. Our algorithm outperforms Retinex for these images.

6.3 Enhancement of Reconstructed Images

Based on the extraction of intrinsic images using all three streams (HTA) we tested the post-processing enhancements presented in Section 5. Recovering the DC component in the shading image does not alter its correlation with the ground truth stimuli but it does improve the estimate of reflectance obtained by dividing the original image by the reconstructed shading map (compare dRef and dRef-original in Fig. 5(f),(g) and examples in Fig. 5(b),(c)). Recovering the DC component of shading greatly improves it as a basis for further processing. The effects of enhancing reflectance is clearly seen by comparing Fig. 5(d) and (e). Improvements to the reconstructed reflectance images can be assessed more directly by comparing each reconstructed image (RecRef) with its compensated version (CompRef) here we see improved performance for spotlight images and the difficult side-lit cases (Fig. 5(f), (g), conditions 18, 24 & 31).

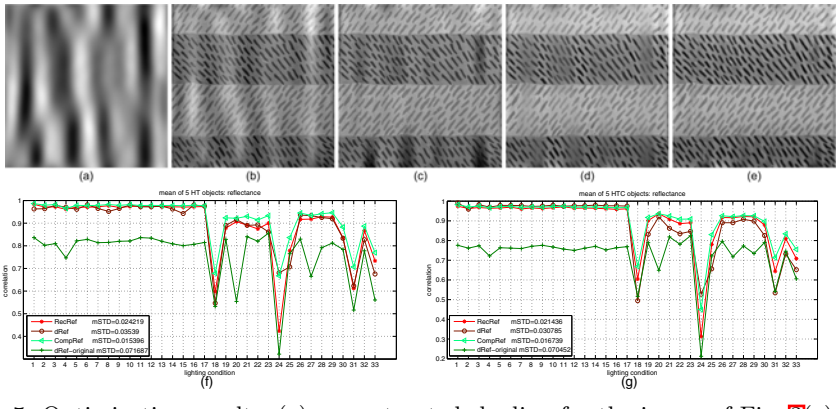


Fig. 5. Optimization results. (a) reconstructed shading for the image of Fig. 2(a), (b) reflectance derived from shading image (a). (c) reflectance derived from optimized shading. (d) reconstructed reflectance. (e) compensated reflectance. The mean correlation value between reflectance and ground truth for 5 different HT objects (f) and for 5 different HTC objects (g). mSTD is the mean stand deviation of the correlation for 5 objects across 33 lighting conditions.

6.4 Evaluation on Natural Images

Fig. 6 shows the performance of our algorithms compared to Retinex on the full MIT data set. The results for reflectance and shading have been averaged. The

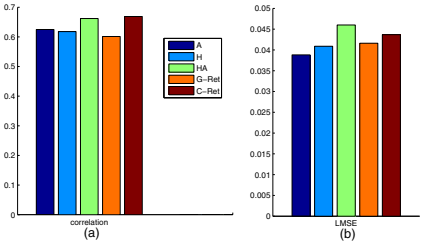


Fig. 6. Average results on the MIT data set

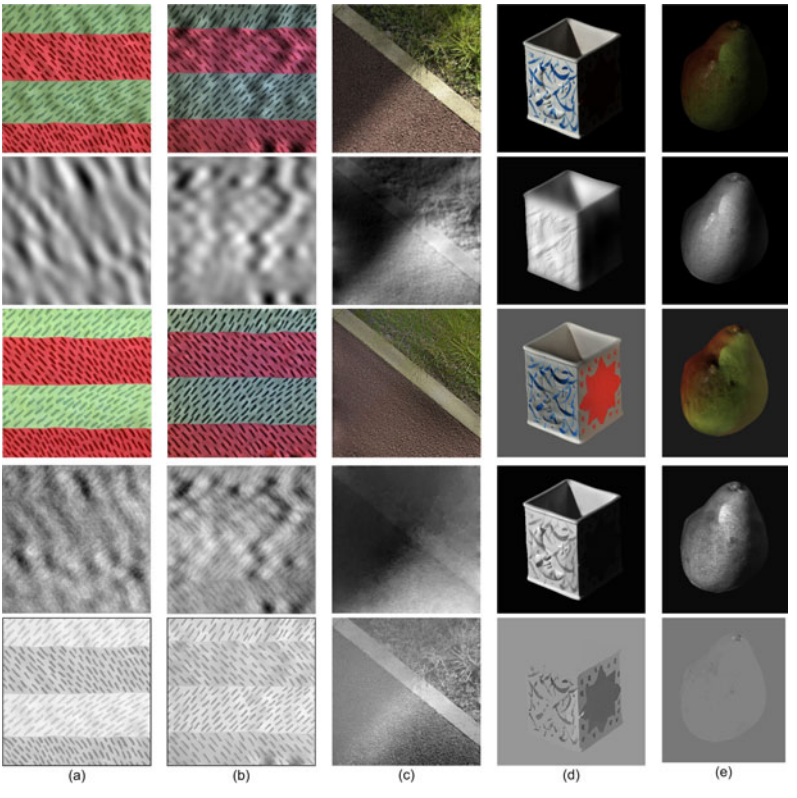


Fig. 7. Intrinsic images extracted by our algorithm and colour Retinex. Top to bottom in each panel: original image, shading and reflectance estimates from our algorithm and colour Retinex respectively. (a) isotropic ripples 1 with HTC texture, light condition $l = 14$; (b) isotropic ripples 2 with HT texture, $l = 19$; (c) a natural image. (d),(e) sample images from MIT data set.

LMSE measure [20], Fig. 6(b), favours our algorithm with only 'A' or 'H' (A low value is good.) while our correlation metric, Fig. 6(a), places our 'HA' algorithm equal to C-Retinex. (A high value is good.) We prefer our correlation metric as LMSE includes a term which can amplify the contrast of each local patch. This term is optimised on a local basis to reduce the MSE. Thus, it is possible to produce low LMSE scores from output images that do not visually match ground truth, as in the case of 'A' and 'H' only. The supplementary material provides a more detailed explanation.

Finally Fig. 7 shows some additional results based on our test set, a natural image where shadows fall across regions of different textures and colours, and two images from the MIT test set [20]. Example output from the colour Retinex method is also shown in Fig. 7. Our algorithm gives fairly accurate estimations for shading and reflectance. Although it does not perform well on the MIT 'box' stimulus (Fig. 7(d)), it outperforms Retinex on the 'pear' stimulus (Fig. 7(e)). From these results we see that our algorithm extracts low-frequency shading very well and is suited to natural textures. However the assignment of components is not accurate for high-frequency shading, due to the limitations of the frequency bandwidth of the steerable filters (about 1 octave). Furthermore, our algorithm currently deals with each component globally. This failing differentially affects high-frequency components which tend to arise from more local features.

7 Conclusion

In this paper, we propose an algorithm to extract shading and reflectance maps from a single input image. Based on results from human vision we propose local luminance amplitude (AM) as an effective cue for separating shading and reflectance along with texture and hue. We also introduce a multi-resolution framework to decompose images into components that can then separately contribute to the shading and reflectance maps. Correlation coefficients between luminance and the different cues (AM, texture, and hue) decide the weight for each component in each map. Experiments on images of rippled surfaces under different lighting conditions showed the effectiveness of the proposed algorithm in all but a few hard cases. In the proposed algorithm, we gave each component a global weight for each reconstruction, but shading or reflectance information may exist in only specific locations, such as at shadow edges. That is, correlations between different cues may be location-dependent. Thus the algorithm could be improved by adding a local correlation measure. Our method can produce two estimates for both shading and reflectance. One estimate is derived directly from the weighted components. The other can be derived by dividing the original image by the other reconstructed intrinsic image (e.g. shading=image/reflectance). Here, we used the reconstructed shading image to improve reflectance estimates, but did not provide an equivalent enhancement for shading. In future work, considering the four initial estimates together might improve estimates of shading and reflectance.

Acknowledgement. This project is supported by EPSRC grant EP/F026269/1.

References

1. Barrow, H.G., Tanenbaum, J.M.: Recovering intrinsic scene characteristics from images. Computer Vision systems, 3–26 (1978)
2. Land, E.H., McCann, J.J.: Lightness and retinex theory. Journal of the Optical Society of America A 61, 1–11 (1971)
3. Sinha, P., Adelson, E.: Recovering reflectance and illumination in a world of painted polyhedra. In: ICCV, pp. 156–163 (1993)
4. Bell, M., Freeman, W.T.: Learning local evidence for shading and reflectance. In: ICCV, vol. 1, pp. 670–677 (2001)
5. Funt, B.V., Drew, M.S., Brockington, M.: Recovering shading from color images. In: Sandini, G. (ed.) ECCV 1992. LNCS, vol. 588, pp. 124–132. Springer, Heidelberg (1992)
6. Olmos, A., Kingdom, F.A.A.: A biologically inspired algorithm for the reconvery of shading and reflectance images. Perception 33, 1463–1473 (2004)
7. Tappen, M.F., Freeman, W.T., Adelson, E.H.: Recovering intrinsic images from a single image. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1459–1472 (2005)
8. Parraga, C.A., Troscianko, T., Tolhurst, D.J.: Spatiochromatic properties of natural images and human vision. Current Biology 12, 483–487 (2002)
9. Finlayson, G.D., Hordley, S.D., Lu, C., Drew, M.: On the removal of shadows from images. IEEE Trans. Pattern Analysis and Machine Intelligence 28, 59–68 (2006)
10. Kingdom, F.A.A.: Perceiving light versus material. Vis. Res. 48, 2090–2105 (2008)
11. Shen, L., Tan, P., Lin, S.: Intrinsic image decomposition with non-local texture cues. In: IEEE Computer Vision and Pattern Recognition, pp. 1–7 (2008)
12. Weiss, Y.: Deriving intrinsic images from image sequences. In: ICCV, vol. 2, pp. 68–75 (2001)
13. Agrawal, A., Raskar, R., Chellappa, R.: Edge suppression by gradient field transformation using cross projection tensors. In: CVPR, vol. 2, pp. 2301–2308 (2006)
14. Matsushita, Y., Lin, S., Kang, S.B., Shum, H.Y.: Estimating intrinsic images from image sequences with biased illumination. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 274–286. Springer, Heidelberg (2004)
15. Schofield, A.J., Hesse, G., Rock, P.B., Georgeson, M.A.: Local luminance amplitude modulates the interpretation of shape-from-shading in textured surfaces. Vision Research 46, 3462–3482 (2006)
16. Simoncelli, E.P., Freeman, W.T.: The steerable pyramid: A flexible architecture for multi-scale derivative computation. In: ICIP, pp. 444–447 (1995)
17. Comanicu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24, 603–619 (2004)
18. Reinhard, E., Devlin, K.: Dynamic range reduction inspired by photoreceptor physiology. IEEE Trans. Visualization and Computer Graphics 11, 13–24 (2005)
19. Waltz, R.A., Morales, J.L., Nocedal, J., Orban, D.: An interior algorithm for nonlinear optimization that combines line search and trust region steps. Mathematical Programming 107, 391–408 (2006)
20. Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground-truth dataset and baseline evaluations for intrinsic image algorithms. In: ICCV (2009)

ADICT: Accurate Direct and Inverse Color Transformation

Behzad Sajadi, Maxim Lazarov, and Aditi Majumder

University of California, Irvine
{bsajadi,mlazarov,majumder}@uci.edu

Abstract. A color transfer function describes the relationship between the input and the output colors of a device. Computing this function is difficult when devices do not follow traditionally coveted properties like channel independency or color constancy, as is the case with most commodity capture and display devices (like projectors, camerass and printers). In this paper we present a novel representation for the color transfer function of any device, using higher-dimensional Bézier patches, that does not rely on any restrictive assumptions and hence can handle devices that do not behave in an ideal manner. Using this representation and a novel reparametrization technique, we design a color transformation method that is more accurate and free of local artifacts compared to existing color transformation methods. We demonstrate this method's generality by using it for color management on a variety of input and output devices. Our method shows significant improvement in the appearance of seamlessness when used in the particularly demanding application of color matching across multi-projector displays or multi-camera systems. Finally we demonstrate that our color transformation method can be performed efficiently using a real-time GPU implementation.

1 Introduction

A color transfer function matches colors in a device-dependent RGB space to those in the device-independent CIE XYZ space. For capture devices, the input is in the XYZ space and the output in the RGB space and vice-versa for display devices (like projectors and printers). When the domain and the range of this function are the RGB and the XYZ spaces respectively, it is a *direct transfer function*. When the domain and the range are switched, the function is an *inverse transfer function*. Let us consider two devices—a source and a target—with direct transfer functions T_s and T_t. The color given by (r_s, g_s, b_s) in the source device can be achieved by the input $(r_t, g_t, b_t) = T_t^{-1}T_s(r_s, g_s, b_s)$ in the target device. Here T_s is the direct transfer function and T_t^{-1} is the inverse transfer function. Accurate computation of $T_t^{-1}T_s$ is the goal of any color management system. For ideal devices with channel color constancy (constant chromaticity across all channel inputs) and no channel interdependencies, both the direct and inverse transfer functions, T_s and T_t, are linear 3×3 matrices. This allows easy inversion and concatenation to compute desired target inputs that create the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 72–86, 2010.

same image as in the source. However, current commodity devices like projector, cameras and printers deviate considerably from the ideal properties of channel independency and color constancy, making it difficult to compute $T_t^{-1}T_s$.

In this paper, we represent the color transfer functions of any non-ideal device using multiple higher-dimensional Bézier patches. Augmenting this with a novel reparametrization of colors in the device independent space, we design a new color transformation method (Section 3) with the following advantages.

Generality: The Bézier based representation of the color transfer function does not depend on ideal behavior of the device. Hence, it can be applied to devices with significant channel interdependencies (often due to the use of more than three primaries that do not form a basis), no color constancy, and non-monotonic channel transfer functions. It is also backward compatible to ideal devices and can be applied to both capture and display devices alike (Section 5).

Quality: Unlike existing color transformation methods (Section 2), our method assures (C^2) continuity, resulting in smoother transition from one color to another. It also handles non-monotonicity in the transfer function elegantly. Consequently, our method consistently shows greater accuracy and less local artifacts compared to the existing methods when tested on a variety of devices including projectors, cameras and printers (Section 5).

Application: The quality of any color matching method is particularly challenged in applications where images from multiple devices are placed in a spatially contiguous manner, where human ability to detect color differences increases [1]. This occurs when tiling multiple display devices (e.g. tiled display walls) or when stitching different parts of a scene captured with different cameras in a panorama (e.g. surveillance application). In these applications, our method shows significant improvement in color matching, especially when the devices differ significantly in color properties (Section 5). Thus, such applications no longer are restricted to use devices of same model or architecture to avoid large variations in color.

Efficiency: Our color transformation method is amenable to real-time implementation on GPU(Section 4).

2 Previous Work

Let $i_l, l \in \{r, g, b\}$, $0 \leq i_l \leq 1$, be a channel input in the device-dependent RGB space. The color gamut of an ideal device (exhibiting channel independence and color constancy) in the XYZ space is a parallelepiped spanned by three vectors (X_l, Y_l, Z_l), one per channel l. The color in the XYZ space corresponding to (i_r, i_g, i_b) in the RGB space is then given by

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{pmatrix} \begin{pmatrix} i_r \\ i_g \\ i_b \end{pmatrix} = M \begin{pmatrix} i_r \\ i_g \\ i_b \end{pmatrix} \tag{1}$$

Here, the forward transfer function T is the matrix M and can be estimated accurately by measuring the color output at only three points, namely $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. Also, T can be easily inverted by computing M^{-1}.

To match the response of the ideal device with that of the human eye, each channel usually has a non-linear channel transfer function, given by $h_l, l \in \{r, g, b\}$. Accounting for h_l, Equation 2 can be written as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M \begin{pmatrix} h_r(i_r) \\ h_g(i_g) \\ h_b(i_b) \end{pmatrix} \tag{2}$$

Here, to estimate T we need to reconstruct h_l in addition to M [2]. This involves measuring color output at k uniformly sampled channel inputs i_l when inputs to the other two channels are zero. So, for accurate representation of T, we need $3k$ samples. Inverting T involves two steps: (a) Applying M^{-1}; (b) Applying the inverse channel transfer functions h_l^{-1}. To assure invertibility, h_l is assumed to be monotonic, a property satisfied by most traditional devices like CRT displays. Most work on color management assume channel independence and additivity common in traditional displays [3,4].

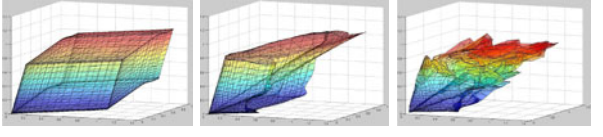


Fig. 1. Visualization of color gamuts of different projection technologies in CIE XYZ space. From left to right: LCD, LCoS and DLP.

Current commodity devices are far from such ideal devices. First, the chroma (defined by the chromaticity coordinates) is not constant across a channel. This means that the boundaries of the parallelepiped (and even iso-contours within it) are no longer straight lines, but wiggly curves. Second, the channels may not be independent. This means that the color transformation is non-additive, i.e. $T(I_1) + T(I_2) \neq T(I_1 + I_2)$, where I and $T(I)$ denote the colors in RGB and XYZ space. In this case, the color gamut ceases to be a parallelepiped. Usually, this is due to the fact that many consumer devices have more than 3 primaries that do not form a basis in the XYZ space [5]. This includes the DLP projectors that use an additional white channel and the printers that use an additional black ink. More recently multi-primary cameras and LED/laser projectors are being introduced [6,7]. Figure 1 shows the effect of such non-ideal behavior on the shape of the 3D gamuts in the XYZ space.

Though some attempts has been made to reconstruct T in non-ideal systems being within the domain of linear matrices [8], they can only tolerate small deviations from ideal devices. The simplest way to reconstruct the function T for non-ideal systems is to sample the three dimensional RGB space uniformly to create a piecewise linear approximation of T. The inversion of the function would then involve a linear interpolation of the function from its neighbors [9]. However, this assumes monotonic iso-contours in T and a dense sampling of the

domain. Assuming k samples per channel, k^3 samples are required for accurate estimation. However, k has to be relatively large ($k = 16$) when compared to our method ($k = 9$) to provide a dense sampling assuring more accurate local interpolation. Even for a relatively small $k = 16$, 16^3 measurements can be very time consuming. Most importantly, linear interpolation assures only C^0 continuity creating considerable visual artifacts (Section 5).

Hence, many current systems custom tailor T for specific devices. Windows Color Management System (WCS), designed jointly by Canon and Microsoft, is a good example [10] and uses different techniques to compute the transfer functions of different devices. For projectors, first the input RGB space is sampled uniformly and the corresponding XYZ values are measured. Let the sampled input values be $(r_i, g_i, b_i), 1 \leq i \leq n$ and the XYZ values (X_i, Y_i, Z_i). For each (X_i, Y_i, Z_i), the input (r'_i, g'_i, b'_i) that would result in (X_i, Y_i, Z_i) is predicted assuming an ideal device described by Equation 2. Next, the vector deviation $\bar{d}_i$ of the actual input (r_i, g_i, b_i) from the predicted input (r'_i, g'_i, b'_i) is computed and associated with the corresponding sampled input. To compute the input for desired XYZ values (X_d, Y_d, Z_d), first input (r'_d, g'_d, b'_d) is predicted assuming an ideal device using Equation 2. Then the deviation is linearly interpolated from the sampled $\bar{d}_i$s and added to (r'_d, g'_d, b'_d) to generate the final input (r_d, g_d, b_d). Unfortunately, in addition to C^1 discontinuity, it can result in non-monotonic output even though there is no non-monotonic iso-parametric curves or surfaces in the input. Further, it assumes channel color constancy that is not true in most commodity devices and results in severe color anomalies (Section 5).

Note that all the above related work, including our work, focus on color management techniques that are content agnostic, i.e. does not depend on the image content. Hence, the device once calibrated, can correct any image. However, our work is orthogonal to a body of literature on content-dependent color management schemes. These determine the best possible color mappings for a specific image based on the particular spatial distribution of the colors to achieve the most perceptually pleasing appearance in the new device [11,12,13,14,15]. Hence, in a content-dependent scheme the color mapping is unique for each image and needs to be recomputed for every image. Content dependent methods cannot achieve interactive rates for videos unless special hardware is used.

3 Algorithm

We present a new general way to represent the color transfer functions of a non-ideal device, both direct and inverse, using a set of Bézier patches. A critical aspect of this representation is a non-linear parametrization of the Bézier patches in the XYZ space (Section 3.1). Using this representation we design a new color transformation method for converting the RGB colors in a source device to those in a target device significantly different than the source device (Section 3.2).

3.1 Color Transfer Function

We do not assume channel chrominance constancy, channel independence, or monotonic channel transfer functions. However, we assume T to be a smooth function. Let us sample T at n different RGB points, i.e., for these n samples – $(r_1, g_1, b_1), (r_2, g_2, b_2) \ldots (r_n, g_n, b_n)$ – we know the corresponding outputs – $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2) \ldots (X_n, Y_n, Z_n)$. Our direct transfer function T is represented by a set of Bézier functions $\mathcal{F}_c(r, g, b)$, one for each XYZ channel, $c, c \in \{X, Y, Z\}$. Our inverse transfer function T^{-1} is represented by another set of Bézier functions $\mathcal{B}_l(X, Y, Z)$, one for each RGB channel, $l, l \in \{r, g, b\}$.

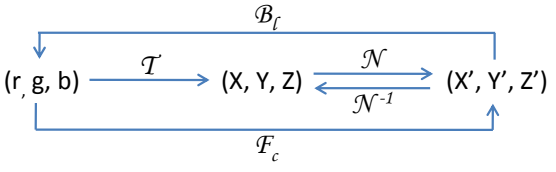


Fig. 2. Representation of color transfer functions

We first describe the reconstruction of $\mathcal{B}_l$. $\mathcal{F}_c$ is computed similarly, but the domain and range color spaces are switched (Figure 2). We reconstruct $\mathcal{B}_l$ for each RGB channel by *fitting* a Bézier to the measured n samples. This consists of two steps. We measure the X, Y, Z values for various input values of r, g, and b uniformly distributed in the RGB space. For every input $\{r_i, g_i, b_i\}$ let the measured output color be $\{X_i, Y_i, Z_i\}$. Using these samples, we build three (one for each channel) 3D Bézier volumes in 4D – $B_r(X, Y, Z)$, $B_g(X, Y, Z)$, and $B_b(X, Y, Z)$. In other words, our Bézier surfaces are height fields in the XYZ space, one for each of the three input channels.

Although the data points we collect are uniform in the RGB space, they need not be uniformly distributed in the XYZ space, limiting our ability to fit a Bézier surface to the XYZ colors. Hence we apply a non-linear function $\mathcal{N} : (X, Y, Z) \rightarrow (X', Y', Z')$ in order to make the distribution in the XYZ space uniform. We fit the Bézier surface in this modified XYZ space. The reparametrized XYZ space is sampled in a relatively uniform fashion, and the control points of the Bézier are placed in a regular grid (Figure 3), thus fixing three of the four coordinates of the control points. We use a linear least square method to fix the fourth coordinate so that the computed Bézier height field smoothly passes through the data set. Details of the non-linear reparametrization is available in Section 3.1.

The Bézier representation suits non-ideal devices due to the following:

- Since the Bézier is just a polynomial representation, we can represent non-constant higher order variations in the channel chrominance.
- Since we use a separate Bézier function to represent each of the different input channels, non-additive color transformations due to channel interdependencies can be easily handled.
- Higher order Bézier functions can be reconstructed from fewer samples than required for a piecewise linear representation.
- Bézier aids elegant handling of non-monotonicity in the color transfer function(Section 3.1).

Non-Linear Reparametrization. In this section, we describe the non-linear transformation $(X', Y', Z') = \mathcal{N}(X, Y, Z)$. $\mathcal{N}_l$ is a 3D non-linear function that can be complex and difficult to design. However, note we do not need perfect uniform parametrization, but a function that will yield close to uniform parametrization. Hence, we approximate the function by a channel dependent non-linear function, $\mathcal{N}_l$, resulting from a concatenation of a channel independent 3D linear transformation $\mathcal{L}$, and a channel dependent 1D non-linear function $\mathcal{K}_l$. Figure 3 shows the result of our reparametrization. This results in more accurate Bézier fitting and better interpolation at unsampled points.

3D Linear Function: We propose a 3×3 linear matrix whose columns are given by the XYZ values corresponding to the inputs $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ respectively. $\mathcal{L}$ is the inverse of this matrix. Thus, this is very similar to a standard gamut transformation matrix M^{-1} that allows us to align the XYZ basis with the RGB basis. This transformation yields an intermediate space (X_m, Y_m, Z_m) such that $(X_m, Y_m, Z_m)^T = \mathcal{L}(X, Y, Z)^T$. Following the linear transformation, the primary contribution to X_m, Y_m and Z_m is from r, g, and b respectively.

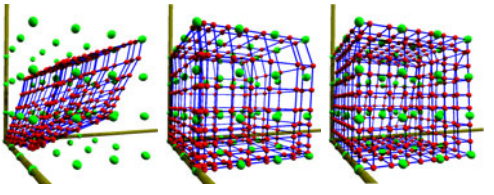


Fig. 3. The samples in XYZ (in red) space and the corresponding control points (in green) generated for B_r before applying $\mathcal{N}_l$ (left), after applying $\mathcal{L}$ (middle), and after applying $\mathcal{K}_l$ (right).

1D Non-Linear Function: Consider function c_r relating r to X_m, i.e. $X_m = c_r(r)$. Similarly we have $Y_m = c_g(g)$ and $Z_m = c_b(b)$. From the measured XYZ values we fit a function of the form $c_l = x^\gamma$ and choose it to be our 1D non-linear function $\mathcal{K}_l$, i.e. $\mathcal{K}_l = c_l^{-1}$.

Handling Non-Monotonicity. Let us first consider monotonicity in a 1D function $y = f(x)$ sampled n times and the (x_i, y_i) samples are in increasing order of x_i. Common way to make these samples monotonically increasing is to apply $y_i = max(y_{i-1}, y_i)$, $2 \leq i \leq n$. We expanded this simple idea to three dimensional sampling to make sure that the iso-contour of the sampled outputs are monotonic. Suppose we have the RGB values in the form of (r_i, g_j, b_k) where all of r_i, g_j, and b_ks are in the increasing order and their corresponding XYZ values after application of $\mathcal{F}$ are $(X_m(i,j,k), Y_m(i,j,k), Z_m(i,j,k))$. The following pseudo code makes the iso-parametric curves of this grid monotonic.

```
for i = 1 : n
  for j = 1 : n
    for k = 1 : n {
        if i > 1 {X_m(i, j, k) = max(X_m(i − 1, j, k), X_m(i, j, k))}
        if j > 1 {Y_m(i, j, k) = max(Y_m(i, j − 1, k), Y_m(i, j, k))}
        if k > 1 {Z_m(i, j, k) = max(Z_m(i, j, k − 1), Z_m(i, j, k))}    }
```

3.2 Color Transformation

The color transfer function of a non-ideal device comprises of $\mathcal{F}_c$, $\mathcal{B}_l$, and $\mathcal{N}$. $\mathcal{N}$ comprises of a matrix and inverse channel transfer functions and is hence similar the ideal device parameters. Hence, the nonlinearities are encoded in $\mathcal{F}_c$ and $\mathcal{B}_l$.

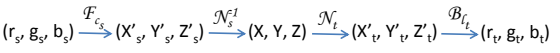


Fig. 4. Color Transformation Method

After the Bézier patches are computed, the color transformation from a source RGB color (r_s, g_s, b_s) to the device RGB color (r_t, g_t, b_t) is achieved using the pipeline in Figure 4. (r_s, g_s, b_s) is first converted to the source reparametrized space (X'_s, Y'_s, Z'_s) using $\mathcal{F}_{cs}$. This denotes the desired color in the device independent XYZ color space. To find the target (r_t, g_t, b_t) that produces this color, N_t is applied followed by evaluation of $\mathcal{B}_{lt}$ at the reparametrized (X'_t, Y'_t, Z'_t).

When matching colors across multiple devices, often the desired color is provided in the XYZ space – say (X_d, Y_d, Z_d). In this situation, achieving a color

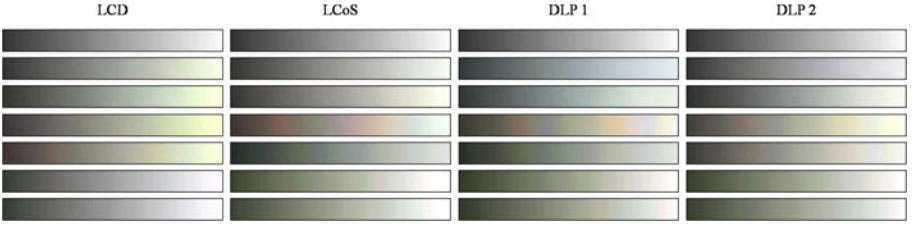


Fig. 5. Comparison of black-to-white gradient reproduction quality using, from top to bottom, target image, our method, method for ideal devices, WCS, Adobe CMM, and linear interpolation (for 9x9x9 and 16x16x16 RGB space samples) in different projection technologies. Note that our method provides the closest match and the smoothest color transitions. Please zoom-in to see the differences.

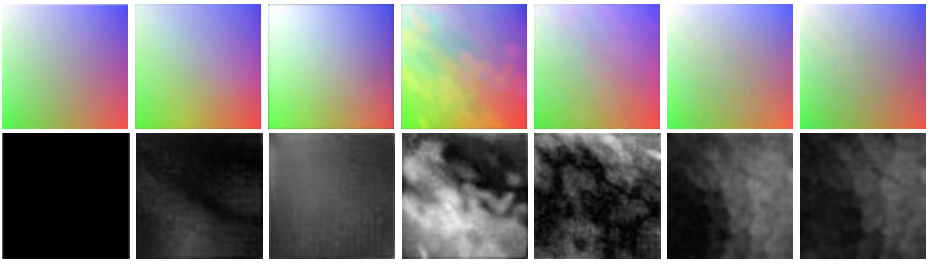


Fig. 6. Comparison of color gradient reproduction quality using different methods – from left to right: target image, our method, ideal devices method, WCS, Adobe CMM, linear interpolation using 9x9x9 and 16x16 samples. The difference of each method from the target image is shown in the second row. Look for color mismatches and artifacts in the top row in the brighter areas. Please zoom-in to see the differences.

Fig. 7. Color uniformity on a display made of 3 heterogeneous projectors (LCD, LCoS and DLP from left to right): Before any color matching (left), after color matching using ideal devices method (middle) and our method (right). Note that the method for ideal devices introduces a color mismatch in the middle projector by changing the color temperature. Our method balances the brightness better, making the left projector of comparable brightness as the other two. Please zoom-in to see the differences.

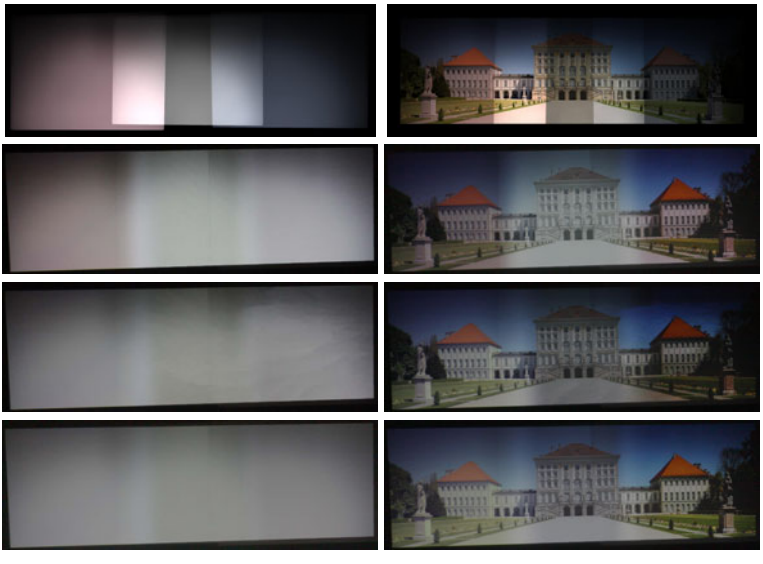


Fig. 8. Comparison of observed color uniformity of a white image and a natural image in a display made of 3 heterogeneous projectors (LCD and two DLPs from left to right). From top to bottom: before color matching, after matching using method for ideal devices, using linear interpolation, and using our method. Zoom in to see banding artifacts in the white image and the severe color temperature change on the right side of the road in the natural image for linear interpolation. Note that the method for ideal device fails to achieve the balancing of the color temperature.

matching involves evaluating the $\mathcal{B}_l$s of the different devices at the same desired (X_d, Y_d, Z_d) after reparametrization. Hence the input to reproduce (X_d, Y_d, Z_d) is given by $(\mathcal{B}_r(X'_d, Y'_d, Z'_d), \mathcal{B}_g(X'_d, Y'_d, Z'_d), \mathcal{B}_b(X'_d, Y'_d, Z'_d)$.

Note that the focus of our algorithm is to compute a target color when within the device gamut. Out-of-gamut colors can be first mapped to an in-gamut color using gamut mapping techniques [16,17] before applying our method.

4 Implementation

In our implementation we use a digital SLR camera (Canon Rebel XSi), that has an sRGB color gamut, as measuring device. [18,9] show that most devices have gamuts that lie within the sRGB gamut and hence our measurements are accurate. We use the RAW images assuring linearity of the camera channel transfer function and no color processing. Hence, to convert the measured colors in the sRGB space to the CIE XYZ space, we apply a standard 3×3 linear transformation. For reconstructing T, we use $k = 9$ (total of $9^3 = 729$ measurements) to sample the input space uniformly. Empirically, this provides good results even for devices with significant deviation from ideal devices – like the ones illustrated in Figure 1. Also, for all devices, a 3D Bézier of degree 3 assures adequate C^2 continuity (details in Section 5.1).

Real-time GPU implementation: The Bézier functions can be stored in a compact manner by storing the $(N+1)^3$ control points for a degree N 3D Bézier (64 control points per channel for our degree 3 Bézier, 192 points in total).

The Bézier is evaluated in runtime to achieve the color correction, implemented in real-time on the GPU using CUDA. We achieve about 70 fps on a 2GHz Xeon workstation with a mid-range GeForce 9600 GT GPU for a XGA (1024 x 768) image.

An alternative to evaluating a Bézier is to compute a 3D LUT for all color values by evaluating the Bézier at all these values which are then indexed by the pixel colors to achieve the color correction. This uses significantly greater storage ($256^3 \times 3$ bytes) but can be implemented on any graphics hardware.

5 Results

We have applied our method to correct colors across many different devices including projectors, cameras, and printers. We have matched colors of multiple devices to a desired image in sRGB space. To evaluate the error between the original (desired or source) images and the corrected image captured by a camera, we compute the pixelwise ΔE color difference (Euclidian distance in the CIE LAB space) between these two images and encode it as a gray image (higher gray values indicate higher deviation from the original image). We also summarize the mean, maximum and the standard deviation of these color differences in Tables 1 and 2. To normalize the brightness of the captured and original image, we scale the captured image by $\frac{M_d}{M_c}$ where M_d and M_c are the mean of the desired and captured image respectively, calculated over all the channels. M_d and M_c provide a measure of the overall brightness of the two images. We use a homography to geometrically align the captured images with the original image since our camera did not show any significant radial distortion [19,20]. Note that the error reports can be improved by using a more accurate color measuring device, such as colorimeter or spectroradiometer.

We compare our method with three different existing methods. First, we compare with the correction suitable for ideal devices as in Equation 2 achieved by

Table 1. The average, maximum, and standard deviation of ΔE color differences of the color-matched image from the original image using different methods. For our method (ADICT) and linear interpolation we also mention the number of samples used ($9 \times 9 \times 9 = 9^3$ or $16 \times 16 \times 16 = 16^3$) and if the reparametrization has been used or not (w R and wo R respectively).

Algorithm	LCD			LCoS			DLP 1			DLP 2		
Gray (Fig. 6)												
Our Method(9^3 w R)	2.96	8.22	1.69	3.45	7.59	1.98	4.91	9.08	1.93	5.12	10.32	2.03
Our Method(9^3 wo R)	3.65	10.41	1.76	3.98	8.26	2.12	5.56	13.47	2.01	5.96	14.83	2.46
Ideal Devices Method	5.28	14.86	1.89	4.41	11.37	2.56	5.02	11.70	1.95	10.61	29.48	3.52
WCS	6.29	17.61	1.97	7.76	19.17	3.50	8.68	23.04	2.32	10.25	23.97	3.16
Adobe CMM	4.93	14.77	1.86	6.73	14.24	2.99	6.97	18.63	2.02	9.46	19.28	2.82
Lin Interp.(9^3 wo R)	3.94	11.07	1.84	4.56	10.17	2.15	6.73	13.29	2.08	6.89	15.35	2.52
Lin Interp.(9^3 w R)	3.28	11.21	1.80	3.76	9.32	2.01	5.48	12.86	1.98	5.93	13.47	2.32
Lin Interp.(16^3 wo R)	3.06	9.85	1.76	3.52	8.75	1.95	5.05	11.43	1.96	5.58	11.48	2.13
Lin Interp.(16^3 w R)	2.93	8.75	1.71	3.47	8.50	1.92	4.81	9.93	1.95	5.18	11.07	2.11
Color (Fig. 5)												
Our Method(9^3 w R)	3.79	8.96	1.77	4.14	10.03	2.14	5.84	18.32	2.34	5.43	19.23	2.54
Our Method(9^3 wo R)	4.97	10.53	2.02	5.54	14.12	2.51	7.45	20.25	2.87	7.04	22.14	2.97
Ideal Devices Method	7.65	16.81	2.46	6.33	18.64	2.73	6.83	28.13	2.48	10.32	29.17	3.43
WCS	8.47	17.00	2.42	10.28	21.86	3.76	13.92	30.94	3.38	12.78	24.53	3.23
Adobe CMM	6.29	14.73	2.29	8.67	17.83	3.12	10.86	23.47	3.10	10.77	22.61	3.01
Lin Interp.(9^3 wo R)	5.12	11.04	2.12	5.58	13.79	2.55	8.25	19.64	2.45	7.94	23.12	2.94
Lin Interp.(9^3 w R)	4.48	11.52	2.01	4.86	11.69	2.36	7.10	18.67	2.38	7.12	20.98	2.84
Lin Interp.(16^3 wo R)	3.85	9.04	1.80	4.73	11.02	2.23	6.02	18.98	2.32	5.98	19.45	2.37
Lin Interp.(16^3 w R)	3.67	8.98	1.79	4.53	10.84	2.18	5.92	18.47	2.30	5.48	19.30	2.36

simply applying an inverse matrix multiplication followed by an inverse channel transfer function. Second, we compare with WCS, a commonly used color management system for Windows, described in Section 2. Third, we compare with Adobe CMM, another commonly used commercial color management system. Underlying principles of this system is not available in public domain, however, we can still compare the results of it with that of our method. Finally, we compare with a linear interpolation of the color transfer function as proposed in [9]. For all these methods, we use a sparse sampling of colors, 9^3, as used in our method. However, for the linear interpolation method, we also compare with a much denser sampling, 16^3, as proposed in [9]. Please note that capturing 9^3 and 16^3 images in our experiments with a high-end camera took about 2 hours and 11 hours respectively. For projectors in addition to the longer calibration time heating issues also come into picture when we capture 16^3 images.

To show the effectiveness of our non-linear parametrization (Section 3.1), we show that a version of our method where the reparametrization is not applied yields less accurate results. The advantage of our reparametrization is further emphasized by the reduction of error when it is applied for the linear interpolation method [9].

Note that after combining linear interpolation with our nonlinear reparameterization and with using 6 times more images we achieved statistically similar results compared to our method. However, we still see severe local artifacts in the linear interpolation results due to the lack of smoothness constraints. This is more pronounced in the whitish images (Figures 5 and 8).

Matching Different Projectors: We have used our method on three types of projectors: LCD (Epson EMP 74), LCoS (Canon Realis X700) and DLP (Sharp XG-PH50X and InFocus Screenplay 4800). Unlike three-primary LCD and LCoS, DLP projectors use a four-color filter wheel. Thus, they exhibit the greatest deviation from an ideal monotonic parallelepiped gamut due to strong contribution from the fourth 'white' primary. We use two desired images in sRGB space: (a) a smooth linear gradient from black to white (Figure 5) and (b) a color gradient image which shows smooth transition of colors from red, green, blue and white (Figure 6). Note that the existing methods show significant deviation from the desired image and also visual artifacts like blotches while our method yields smoother and more accurate colors; the error statistics in Table 1 emphasize this. In particular, the color matching is significantly improved by our method compared to the method for ideal devices when applied to devices that deviate significantly from ideal additive gamut (like the InFocus DLP projector).

The results of any color transformation method is best illustrated when used to match colors across spatially contiguous devices when humans are more sensitive in detecting color differences [1]. We used our method to achieve color matching across displays made of multiple projectors. We built a three-projector display with projectors of different technology and balanced their color using different methods for comparison. The remaining spatial variation of intensity after the color matching is corrected using methods by Majumder and Stevens [21].

Fig. 9. Comparison of printer-to-projector color matching for a black-to-white gradient. Note that our method provides the closest match and the smoothest color transitions.

In the first setup, we use an LCoS, an LCD and a DLP projector (Figure 7) and compare it with color matching method for ideal devices. In the second setup, we use an LCD projector with two DLP projectors and compare with the method for ideal devices and the linear interpolation technique in [9]. In both cases, existing methods show color mismatches or visual artifacts like blotching and banding while our method shows a seamless result, especially for flat white, the most testing pattern for demonstrating color matching. Note that multi-projector displays are usually never built using projectors of different technologies to avoid the difficult color matching problem. Also, LCD projectors that are close to ideal are the most common choice for multi-projector displays so that method for ideal devices can be used to achieve the color matching. Our result demonstrates that seamless displays made of projectors of different technologies are possible if a sophisticated color management algorithm as ours is applied.

Fig. 10. Comparison of printer-to-projector color matching for a natural image. Difference from input image is shown in the second row of images. From left to right are the printed image, our method, method for ideal devices, WCS, and Adobe CMM. Note the banding artifact and less vibrancy in color due to inaccurate color match in WCS, Adobe CMM and the method for ideal devices when compared to our method. Please zoom in to see the differences.

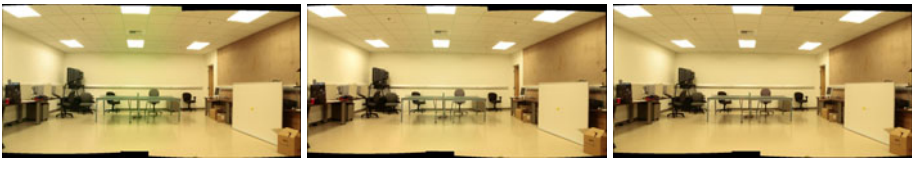


Fig. 11. Panorama generated from images captured by three different cameras, when using no color matching (left), inverse transformation for ideal devices (middle) and color matching using our method (right). The image on the left shows severe color mismatch. This is not corrected by the middle one – note the greenish tinge in the white of the floor and ceiling near the center and also lower saturation of the green color of the ping-pong table. These artifacts are completely removed by our method on the right. Please zoom in to see the differences.

Matching a Projector to a Printer: We demonstrate the use of our method for matching a target projected image to its source printed counterpart. To sample the color transfer function of the printer, we capture with the measurement camera a printed color chart of the 9x9x9 samples. Then apply the different methods to match the color (Figure 4). For this, we use a gray gradient (Figure 9) and a natural image (Figure 10). We find our method to provide the closest match devoid of any banding artifacts. The error statistics on the deviation from the original is summarized in Table 2.

Matching Different Cameras: We use our method to match color across three cameras (Sony DSC-W1, Sony DSC-F707, and Canon SLR 30D) that together capture a panorama (Figure 11). To find the correspondence between the RGB space of each camera and XYZ space, we capture multiple images using our measurement camera and the camera whose color has to be characterized. Following a homography based registration, this provides us millions of correspondences. We choose the appropriate ones to assure a close to uniform sampling in the RGB space.

Table 2. The average, maximum, and standard deviation of ΔE color difference of the captured image from the original image of our linear gradient (Fig. 9), in the first row, and natural scene (Fig. 10), in the second row, for projector-to-printer color matching.

Our			Ideal			WCS			Adobe CMM		
2.71	6.71	1.54	3.73	10.22	2.13	6.35	12.90	3.48	5.31	9.97	2.71
2.66	9.41	1.72	4.58	15.16	2.34	5.08	16.96	3.63	4.60	14.89	2.98

5.1 Discussion

Generality: Since our method works directly with the three channel input in which media is usually formatted, it can handle any device irrespective of the actual number of primaries used and the exact method of combining them within the device. Since, our method can be used for both direct and inverse color transformations, it can be used for both capture and display devices alike.

Backward Compatibility: When handling devices that are ideal or close to ideal, instead of sampling the output color at all k^3 samples, we can just measure the output color at k values for each channel, i.e. $3k$ measurements. Rest of the samples can be predicted using the additivity assumption. The rest of the our method remains unchanged. Hence, our method is backward compatible to ideal devices, as is demonstrated by the superior results on the near ideal LCD projector (Figure 6 and 5).

Superior Color Management: To illustrate the significantly better results of our method, we choose projectors since they are good examples of commodity devices with all kinds of anomalies (Figure 1). Almost all different projector technologies (LCD, LCoS, DLP) show channel color non-constancy. LCoS and DLP projectors are severely non-additive in nature. [18] presents extensive studies on projectors that show non-monotonic color responses. Same is shown in [22]. Non-monotonicity is also common in cameras [23]. Unlike linear interpolation that assumes monotonicity and WCS that does not preserve monotonicity, our Bezier based method handles non-monotonicity better. Further, unlike existing methods that can handle only additive gamuts or assure only C^0 continuity while handling non-additive gamuts, our method assures C^2 continuity for both additive and non-additive gamuts. Hence, our method yields superior results that all existing methods consistently.

Degree of the Bezier: We experimented with Beziers of degree up to 6. Cubic Beziers provided a good fitting that was improved marginally by using degree 4. We chose the cubic Bezier for faster GPU implementation. Degrees 5 and 6 showed some visual noise due to over fitting. However, the user can choose the degree that works best for the particular device.

Perceptual Plausibility: Please note that Even though statistically in some of the experiments the non-linear interpolation method with a higher sampling rate achieved similar results to our method still it shows severe local artifacts especially for white as can be seen in Figures 5 and 8. This is due to the fact

that our method uses a smooth non-linear interpolation which makes it devoid of these local artifacts. Also please note that these statistical results achieved in combination with our reparameterization and with 6 times more samples.

Sparse Sampling: The sparse sampling for reconstructing the color transfer function is also an additional advantage of our method. Our method requires $9^3 = 729$ samples, an order of magnitude smaller than the $16^3 = 4056$ samples required for the linear interpolations that provide somewhat comparable results. In case of some devices such as projectors we need to capture one image per sample. With a high-end camera it took about 2 hours to capture the 729 images while it takes 11 more than hours for 4056 samples.

6 Conclusion

We have presented a new general method for computing the direct and inverse color transformations for non-ideal devices. This can be extremely useful for addressing color management demands of commodity devices. Our Bézier representation of these functions is general, can be stored compactly, and evaluated in real-time using a GPU. Since our method does not make any assumptions on the nature of the color properties of the device, we have shown that it can be used to match colors across heterogeneous display and capture devices. In the future this work can be used as a foundation to explore color seamlessness algorithms for multi-camera or multi-projector systems.

References

1. Valois, R.L.D., Valois, K.K.D.: Spatial Vision. Oxford University Press, Oxford (1990)
2. Berns, R., Motta, R., Gorzynski, M.: Crt colorimetry, part i and ii: Theory and practice. Color Research and Application 18, 299–325 (1992)
3. Bala, R., Braun, K.: A camera-based method for calibrating projection color displays. In: 14th Color Imaging Conference (2006)
4. Bastani, B., Ghaffari, R., Funt, B.: Optimal linear rgb-to-xyz mapping for color display calibration. In: 12th Color Imaging Conference (2004)
5. Heckaman, R.L., Fairchild, M.D., Wyble, D.: The effect of dlp projector white channel on perceptual gamut. In: 13th Color Imaging Conference (2005)
6. Niven, G., Mooradian, A.: Low cost lasers and laser arrays for projection displays, pp. 1904–1907 (2006)
7. Kishimoto, J., Yamaguchi, M., Ohyama, N.: Evaluation of tone mapping for multiband high dynamic range images. In: ACM SIGGRAPH Talks (2008)
8. Wyble, D.R., Rosen, M.R.: Color management of dlp projectors. In: 12th Color Imaging Conference (2004)
9. Wallace, G., Chen, H., Li, K.: Color gamut matching for tiled display walls. In: Immersive Projection Technology Workshop (2003)
10. Tin, S.K.: Color characterization of projectors. US Patent 7148902 (2006)
11. Balasubramanian, R., de Queiroz, R., Eschbach, R.: Gamut mapping to preserve spatial luminance variations. Journal of Image Science and Technology 45, 436–482 (2001)

12. Horiuchi, T., Tominaga, S.: Color gamut mapping algorithm for preserving spatial ratios. In: 16th Color Imaging Conference (2008)
13. Nakauchi, S., Hatanaka, S., Usui, S.: Color gamut mapping based on a perceptual image difference measure. Color Research and Application 24, 280–290 (1999)
14. Kimmel, R., Shaked, D., Elad, M., Sobel, I.: Space dependent color gamut mapping: A variational approach. IEEE Transactions on image processing, 796–803 (2005)
15. McCann, J.J.: Lessons learned from mondrian applied to real images and color gamuts. In: 7th Color Imaging Conference (1999)
16. Montag, E.D., Fairchild, M.D.: Psychophysical evaluation of gamut mapping techniques using simple rendered images and artificial gamut boundaries. IEEE TIP 6, 977–989 (1997)
17. Morovic, J., Ronnier, L.M.: The fundamentals of gamut mapping: a survey. The Journal of Imaging Science and Technology 45, 283–290 (2001)
18. Majumder, A., Stevens, R.: Color nonuniformity in projection-based displays: Analysis and solutions. IEEE TVCG 10(2) (2003)
19. Sukthankar, R., Stockton, R., Mullin, M.: Smarter presentations: Exploiting homography in cameraprojector systems. In: IEEE ICCV (2001)
20. Raskar, R.: Immersive planar displays using roughly aligned projectors. In: IEEE VR (1999)
21. Majumder, A., Stevens, R.: Perceptual photometric seamlessness in tiled projection-based displays. In: ACM TOG, vol. 24 (2005)
22. Nayar, S.K., Peri, H., Grossberg, M.D., Belhumeur, P.N.: A projection system with radiometric compensation for screen imperfections. In: IEEE PROCAMS (2003)
23. Grossberg, M., Nayar, S.: Determining the camera response from images: What is knowable? In: IEEE PAMI, vol. 25, pp. 1455–1467 (2003)

Real-Time Specular Highlight Removal Using Bilateral Filtering*

Qingxiong Yang, Shengnan Wang, and Narendra Ahuja

University of Illinos, Urbana Champaign
http://vision.ai.uiuc.edu/~qyang6/

Abstract. In this paper, we propose a simple but effective specular highlight removal method using a single input image. Our method is based on a key observation - the maximum fraction of the diffuse color component (so called maximum diffuse chromaticity in the literature) in local patches in color images changes smoothly. Using this property, we can estimate the maximum diffuse chromaticity values of the specular pixels by directly applying low-pass filter to the maximum fraction of the color components of the original image, such that the maximum diffuse chromaticity values can be propagated from the diffuse pixels to the specular pixels. The diffuse color at each pixel can then be computed as a nonlinear function of the estimated maximum diffuse chromaticity. Our method can be directly extended for multi-color surfaces if edge-preserving filters (e.g., bilateral filter) are used such that the smoothing can be guided by the maximum diffuse chromaticity. But maximum diffuse chromaticity is to be estimated. We thus present an approximation and demonstrate its effectiveness. Recent development in fast bilateral filtering techniques enables our method to run over **200**$\times$ faster than the state-of-the-art on a standard CPU and differentiates our method from previous work.

1 Introduction

The spectral energy distribution of the light reflected from an object is the product of the spectral energy distribution of the illumination and the surface reflectance. Using the dichromatic reflection model [14], the reflected light can be separated into two components, due to specular and diffuse reflections, respectively. Specular reflection presents difficulties for many computer vision tasks, such as segmentation, detection and matching, since it captures source characteristics, creating a discontinuity in the omnipresent, object-determined diffuse part. For simplification, specularities are usually disregarded as outliers by methods that are based on the diffuse component analysis. Since the presence of specular reflection is inevitable in real world, and they do capture important scene information, e.g., surface shape and source characteristics, incorporation of specularity regions in the analysis is important.

Previous methods for separating reflection components can be separated into two categories by the number of images used. The first category uses multiple images taken

* The source code and the tested images are available on the author's website. The support of Hewlett-Packard under the Open-Innovation Research program is gratefully acknowledged.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 87–100, 2010.

under specific conditions (*e.g.*, viewpoint, lighting direction, *etc.*) or using single. [9] used multiple images captured from different polarization angles. Sato and Ikeuchi [13] employed the dichromatic model for separation by analyzing color signatures in many images captured with a moving light source. Lin and Shum [6] also changed the light source direction to produce two photometric images and used linear basis functions to separate the specular components. [11] again requires different illumination directions. These approaches are of restricted use in a general setting since the light source is usually fixed in the real world. A feasible solution is to change the view point instead of changing the illumination direction. Using multiple images taken from different viewing directions, Lee [4] presented a method for specular region detection and Lin [5] removed the highlights by treating the specular pixels as outliers, and matching the remaining diffuse parts in other views. However, this method may fail if the size of the highlight region is large, because then the large number of pixels involved can not be considered as outliers. These methods are moderately practical, since it may not always be possible to meet the required conditions in practice.

Highlight removal using a single image, as in the other category, is generally much more challenging. When dealing with multi-colored images, most single-image-based methods require color segmentation (*e.g.*, [3],[1]) which is known to be non-robust for complex textured images or requires user assistance for highlight detection [16]. It was therefore a significant advance when Tan and Ikeuchi [20] demonstrated that highlights from textured objects with complex multi-colored scenes can be effectively removed without explicit color segmentation. This method removes highlights by iteratively shifting chromaticity values towards those of the neighboring pixel having the maximum chromaticity in the neighborhood. The neighborhood is determined using a "pseudo-coded" diffuse image which has exactly the same geometrical profile as the diffuse component of the input image and can be generated by shifting each pixel's intensity and maximum chromaticity nonlinearly. Assuming that the specular intensity is either zero (for diffuse pixels) or a constant, Shen and Cai [15] introduced a fast highlight removal method using a modified "pseudo-coded" diffuse image, but this method only works for multi-color surfaces when the dominant highlight region is approximately uniform. Similar to the "pseudo-coded" diffuse image presented in [20], Mallick *et. al.* [8] proposed an SUV color space which separated the specular and diffuse components into S channel and UV channels. This SUV space was further used for highlight removal by iteratively eroding the specular channel using either a single image or video sequences [7]. This type of approaches may encounter problems due to discontinuities in surface colors, across which diffuse information cannot be accurately propagated.

Other approaches for single-image highlight removal analyze the distributions of image colors within a color space. Tan and Ikeuchi [19] related the specular pixels to diffuse pixels for every surface color by projecting image colors along the illumination color direction to a point of lowest observed intensity. As a result, the decomposition can be expressed in a close form, and can be solved directly for every pixel. Their experimental results contained noise caused by a number of factors, including image noise, color blending at edges, and multiple surface colors. By integrating the texture from outside the highlight to determine the candidate diffuse color for traditional

color-space technique, that is for each pixel, a set of candidate diffuse colors is obtained from a texture scale of 1×1, and is iteratively pruned as the texture scale increase, Tan et al. [17] showed appreciable improvements in diffuse-highlight separation. This method requires that there are enough repetitive textures locally and the highlight does not have similar color to the surface.

All above methods that use a single input image share the same problem that they are not capable for real-time applications, e.g., stereo matching for specular surfaces, and generally result in noticeable artifacts. In this paper, we propose a simple but effective specular highlight reduction method using a single input image. Our method is closely related to [20], in which, the diffuse color of a specular pixel is derived as a nonlinear function of its color (from input image) and the maximum fraction of the diffuse color components which is denoted as maximum diffuse chromaticity in the paper. The final step is estimating the maximum diffuse chromaticity value for every pixel which is a non-trivial problem, and a method which iteratively shifting chromaticity values towards those of the neighboring pixel having the maximum chromaticity in the neighborhood was proposed. Our method is the same as [20] except for the way of estimating the maximum diffuse chromaticity. Based on a key observation - the maximum diffuse chromaticity in local patches in colorful images generally changes smoothly, we estimate the maximum diffuse chromaticity values of the specular pixels by directly applying low-pass filter to the maximum fraction of the color components of the original image, such that the maximum diffuse chromaticity values can be propagated from the diffuse pixels to the specular pixels. Our method can be directly extended for multi-color surfaces if edge-preserving filters (e.g., bilateral filter) are used such that the smoothing can be guided by the maximum diffuse chromaticity. In practice, maximum diffuse chromaticity is unknown and is to be estimated. We thus present an approximation and demonstrate its effectiveness.

Our method have benefited a lot from the recent development in fast bilateral filtering techniques [12], [23], [24], which enables our method to run **200**$\times$ faster than [20][1] on average. Another advantage of our method is that image pixels are processed independently, allowing for parallel implementation. Our GPU implementation shows that our highlight removal method can process 1MB images at video rate on an NVIDIA Geforce 8800 GTX GPU.

Besides having the speed advantage, our method does not have the non-converged artifacts due to discontinuities in surface colors as presented in [20]. The use of low-pass filter guarantees that the estimated maximum diffuse chromaticities will be locally smooth, so are the estimated diffuse reflections. Nevertheless, since the theory of our method is heavily built upon [20], it shares most of the limitations with [20], *e.g.*, the input images have chromatic surfaces, the output of the camera is linear to the flux of the incident light, and the illumination chromaticity can be correctly measured/estimated.

2 Algorithm

In this section, we first briefly review the adopted reflection model (Sec. 2.1), and then present a real-time highlight removal method (Sec. 2.2).

[1] The source code is available on its author's homepage [18].

2.1 Reflection Model

Using standard diffuse+specular reflection models commonly used in computer graphics, the reflected light color ($\boldsymbol{J}$) captured by a RGB camera can be represented as a linear combination of diffuse ($\boldsymbol{J^D}$) and specular ($\boldsymbol{J^S}$) colors:

$$\boldsymbol{J} = \boldsymbol{J^D} + \boldsymbol{J^S}. \tag{1}$$

Let chromaticity be defined as the fraction of color component c

$$\sigma_c = \frac{J_c}{\sum_{c\in\{r,g,b\}} J_c}, \tag{2}$$

where $c \in \{r, g, b\}$, we define diffuse chromaticity Λ_c and illumination chromaticity Γ_c as follows:

$$\Lambda_c = \frac{J_c^D}{\sum_{c\in\{r,g,b\}} J_c^D}, \tag{3}$$

$$\Gamma_c = \frac{J_c^S}{\sum_{c\in\{r,g,b\}} J_c^S}. \tag{4}$$

Following the chromaticity definition in Eqn. (2), (3) and (4), we express the reflected light color J_c as

$$J_c = \Lambda_c \sum_{u\in\{r,g,b\}} J_u^D + \Gamma_c \sum_{u\in\{r,g,b\}} J_u^S. \tag{5}$$

Assumed that the illumination chromaticity can be measured (with a white reference) or estimated [21], using which the input image can be normalized such that $\Gamma_r = \Gamma_g = \Gamma_b = 1/3$ and $J_r^S = J_g^S = J_b^S = J^S$. Then the diffuse component can be written as

$$J_c^D = J_c - J^S, \tag{6}$$

according to Eqn. 5.

Following the chromaticity definition in Eqn. (2) and (3), we define maximum chromaticity as

$$\sigma_{max} = \max(\sigma_r, \sigma_g, \sigma_b) \tag{7}$$

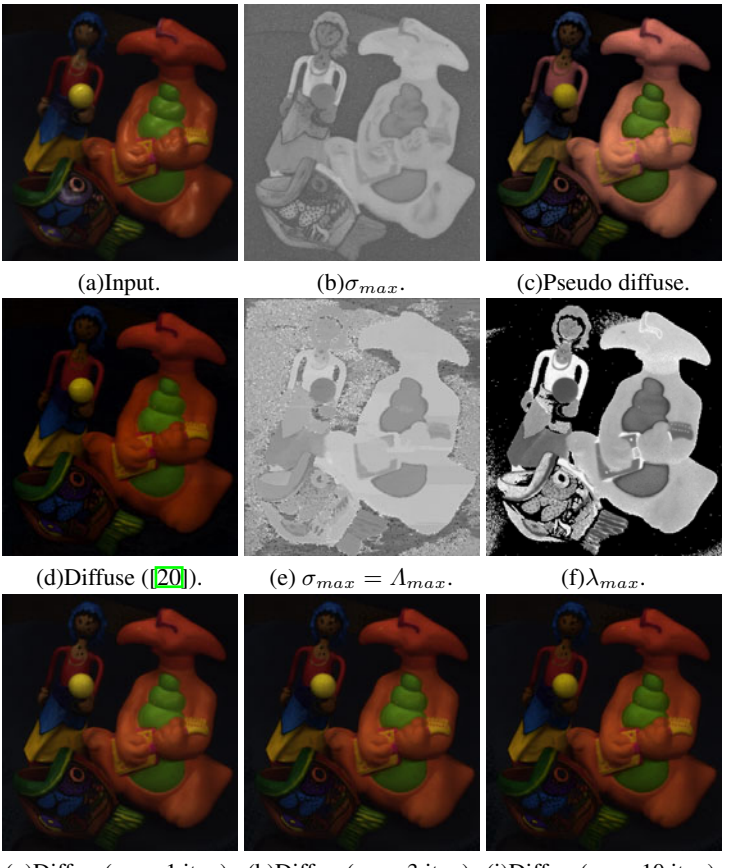
and maximum diffuse chromaticity be

$$\Lambda_{max} = \max(\Lambda_r, \Lambda_g, \Lambda_b). \tag{8}$$

Tan [20] shows that the diffuse component can be represented as a function of Λ_{max}

$$J_c^D(\Lambda_{max}) = J_c - \frac{\max_{u\in\{r,g,b\}} J_u - \Lambda_{max} \sum_{u\in\{r,g,b\}} J_u}{1 - 3\Lambda_{max}}. \tag{9}$$

Since surface materials may vary from point to point, Λ_{max} changes from pixel to pixel in real images but is limited from $\frac{1}{3}$ to 1.



(a)Input. (b)σ_{max}. (c)Pseudo diffuse.

(d)Diffuse ([20]). (e) $\sigma_{max} = \Lambda_{max}$. (f)$\lambda_{max}$.

(g)Diffuse(ours, 1 iter.). (h)Diffuse(ours, 3 iter.). (i)Diffuse(ours, 10 iter.).

Fig. 1. Highlight removal on real data set. (a) is the input image; (b) is the maximum chromaticity σ_{max} (Eqn. 2) computed from (a); (c) is the "pseudo-coded" diffuse image computed using Eqn. (9) by setting Λ_{max} to a constant 0.5; (d) is the extracted diffuse reflection using the method presented in [20][1]; (e) is the maximum chromaticity σ_{max} computed from (d) using Eqn. (2) and (7). Assuming (d) is the ground-truth diffuse reflection, then (e) is also the maximum diffuse chromaticity Λ_{max}; (f) is our approximation (λ_{max}, Eqn. 12) of the maximum diffuse chromaticity Λ_{max}; (g)-(i) are the extracted diffuse reflections using our method after 1, 3 and 10 iterations, respectively. As can be seen, our method generally converges after 3 iterations, and (h) and (i) are visually closed to (d).

Estimating the maximum diffuse chromaticity Λ_{max} for every pixel from a single image is a non-trivial problem. However, if it is set to a constant, then a "pseudo-coded" diffuse image which has exactly the same geometrical profile as the diffuse component of the input image can be obtained. In this case, the saturation values of all pixels are made constant and this "pseudo-coded" diffuse image is essentially a 2D image, while the ground-truth diffuse image is a 3D image. The 2D "pseudo-coded" diffuse image is just an approximation of ground truth, which will fail to preserve the feature

discriminability for surfaces having the same hue but different saturation. However, it is the best estimate we can get, and has been demonstrated to be effective for solving the highlight removal problem in [20]. Fig. 1 presents such an example by setting Λ_{max} to a constant 0.5. Fig. 1 (a) is the input image, (b) is the maximum chromaticity values σ_{max} computed from the input image (a), (c) is the "pseudo-coded" diffuse image, (d) presents diffuse reflection extracted using the method presented in [20] and (e) presents the maximum chromaticity values σ_{max} computed from (d) using Eqn. (2) and (7). Assume that the specular highlights are correctly removed from (d), (e) is also presented as the maximum diffuse chromaticity Λ_{max}.

2.2 Highlight Removal Using Bilateral Filter

According to Eqn. (9), the highlight removal problem can be reduced as the searching for the maximum diffuse chromaticity Λ_{max} which changes from pixel to pixel. However, as shown in Fig. 1 (d) and (e), the variance of Λ_{max} is very small in local patches when the surface colors are consistent. The maximum chromaticity σ_{max} in Fig. 1 (b) is the same as the maximum diffuse chromaticity Λ_{max} except for specular pixels, which cause the intensity/color discontinuities within local patches of the same surface color. Intuitively, applying low-pass filtering to the maximum chromaticity σ_{max} in Fig. 1 (b) will smooth out the variances due to specular highlights. However, there are two issues:

1. The smoothing filter should be edge-aware, such that the σ_{max} values of two pixels associated with different surface materials (Λ_{max} values are different) won't be blend together.
2. The diffuse pixels will be affected by the specular pixels after smoothing.

As a popular edge-aware operator, joint bilateral filter can be employed to smooth the maximum chromaticity σ_{max} using the maximum diffuse chromaticity Λ_{max} as the smoothing guidance. But Λ_{max} is to be estimated, thus we need to find a substitution or an approximation. Although the "pseudo-coded" diffuse image presented in [20] is free of specularity, it is not a good substitution for this problem because its color depends on both the surface geometry and material, while Λ_c is invariant to the surface geometry. Let

$$\sigma_{min} = \min(\sigma_r, \sigma_g, \sigma_b), \tag{10}$$

we approximiate Λ_c using λ_c computed as follows

$$\lambda_c = \frac{\sigma_c - \sigma_{min}}{1 - 3\sigma_{min}}. \tag{11}$$

The relationship between the approximated diffuse chromaticity λ_c and the real diffuse chromaticity Λ_c is captured in Theorem 1 and Theorem 2.

Theorem 1. *For any two pixels* $\boldsymbol{p}$ *and* $\boldsymbol{q}$*, if* $\Lambda_c(\boldsymbol{p}) = \Lambda_c(\boldsymbol{q})$*, then* $\lambda_c(\boldsymbol{p}) = \lambda_c(\boldsymbol{q})$.

Theorem 2. *For any two pixels* $\boldsymbol{p}$ *and* $\boldsymbol{q}$*, if* $\lambda_c(\boldsymbol{p}) = \lambda_c(\boldsymbol{q})$*, then* $\Lambda_c(\boldsymbol{p}) = \Lambda_c(\boldsymbol{q})$ *only if* $\Lambda_{min}(\boldsymbol{p}) = \Lambda_{min}(\boldsymbol{q})$.

Note that λ_c is just an approximation of Λ_c, which will fail for the specific case specified in Theorem 2. However, it is the best estimate we can get. Fig. 1 (f) presents the maximum values of the approximated diffuse chromaticity

$$\lambda_{max} = \max(\lambda_r, \lambda_g, \lambda_b) = \max(\frac{\sigma_r - \sigma_{min}}{1 - 3\sigma_{min}}, \frac{\sigma_g - \sigma_{min}}{1 - 3\sigma_{min}}, \frac{\sigma_b - \sigma_{min}}{1 - 3\sigma_{min}}) \tag{12}$$

computed from the input image presented in Fig. 1 (a).

Using the approximated maximum diffuse chromaticity defined in Eqn. (12) to guide the smoothing, the filtered maximum chromaticity σ_{max} can be computed as follows

$$\sigma^F_{max}(\boldsymbol{p}) = \frac{\sum_{\boldsymbol{q}\in\Omega} \mathcal{F}(\boldsymbol{p}, \boldsymbol{q})\mathcal{G}(\lambda_{max}(\boldsymbol{p}), \lambda_{max}(\boldsymbol{q}))\sigma_{max}(\boldsymbol{q})}{\sum_{\boldsymbol{q}\in\Omega} \mathcal{F}(\boldsymbol{p}, \boldsymbol{q})\mathcal{G}(\lambda_{max}(\boldsymbol{p}), \lambda_{max}(\boldsymbol{q}))}, \tag{13}$$

where $\mathcal{F}$ and $\mathcal{G}$ are spatial and range weighting functions which are typically Gaussian in the literature [22], [2].

The variances of the maximum chromaticity σ_{max} due to specular highlights will be reduced after filtering, and the filtered maximum chromaticity σ^F_{max} will be more closed to Λ_{max} than σ_{max} for the specular pixels. However, after smoothing, the diffuse pixels will be affected by the specular pixels too. According to Theorem 3, the filtered maximum chromaticity values σ^F_{max} of the diffuse pixels will be lower than the unfiltered values σ_{max}. As a result, to exclude the contribution of the specular pixels, we compare σ^F_{max} and σ_{max} and take the maximum value:

$$\sigma_{max}(\boldsymbol{p}) = \max(\sigma_{max}, \sigma^F_{max}(\boldsymbol{p})). \tag{14}$$

Theorem 3. *Assume $\Gamma_c = \frac{1}{3}$, then $\Lambda_{max} \geq \sigma_{max}$. Equality holds when the $\Lambda_{max} = \frac{1}{3}$.*

We then iteratively apply joint bilateral filter to σ_{max} such that the maximum diffuse chromaticity values can be gradually propagated from the diffuse pixels to the specular pixels. In practice, we compare the filtered values σ^F_{max} with σ_{max} after every iteration. The algorithm is believed to converge when their difference is smaller than a threshold (set to 0.03 in our experiments) at every pixel. Our method generally converges after $2 - 3$ iterations. Fig. 1 (g)-(i) present the extracted diffuse reflections after 1, 3 and 10 iterations, respectively. The proposed highlight removal algorithm is summarized in Algorithm 1.

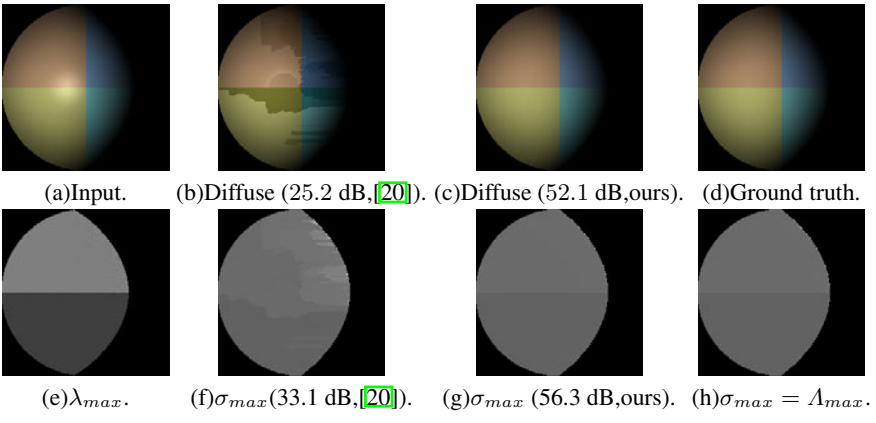
Algorithm 1. Highlight removal using a single RGB image

1: Compute σ_{max} at every pixel using the input image and store it as a grayscale image.
2: Compute λ_{max} at every pixel using the input image and store it as a grayscale image.
3: **repeat**
4: -Apply joint bilateral filter to image σ_{max} using λ_{max} as the guidance image (Eqn. 14), store the filtered image as σ^F_{max};
5: -For each pixel $\boldsymbol{p}$, $\sigma_{max}(\boldsymbol{p}) = \max(\sigma_{max}(\boldsymbol{p}), \sigma^F_{max}(\boldsymbol{p}))$;
6: **until** $\sigma^F_{max} - \sigma_{max} < 0.03$ at every pixel.

3 Experimental Results

To evaluate our method, we conducted experiments on a synthetic data set and several real images either used in the previous work [20] or captured by a Sony DFW-X700 camera with gamma correction off. Quantitative evaluation can only be performed on the synthetic image. For real images, we compared our results with images captured with polarizing filters over the camera and the light source. Comparison with the highlight removal method presented in [20] is also provided.

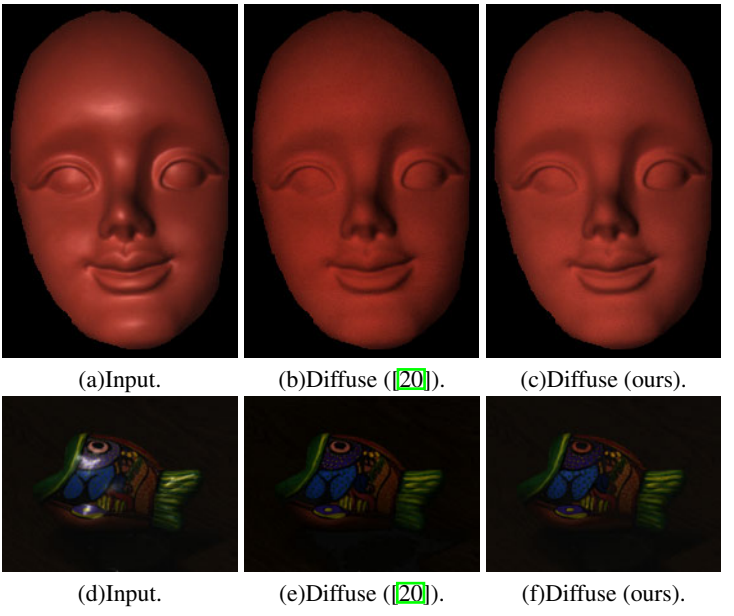
We first numerically compare our method with the method presented in [20] using a synthetic image (used in [6]) presented in Fig. 2 (a). It turns out that [20] performs poorly on this dataset. [20] propagates the maximum chromaticities from diffuse pixels to specular pixels by comparing the maximum chromaticities of two pixels each time, and the greater value is adopted. However, the propagation fails to stop at color edges, and separates incorrect diffuse components for this image. Our method uses a local patch instead of only two pixels, and it is thus more robust and propagates the diffuse chromaticity faster. For numerically evaluation, we compute peak signal-to-noise ratio (PSNR) from the extracted diffuse reflections and the ground-truth presented in Fig. 2 (d). PNSR larger than 40 dB often corresponds to almost invisible differences



(a)Input. (b)Diffuse (25.2 dB,[20]). (c)Diffuse (52.1 dB,ours). (d)Ground truth.

(e)λ_{max}. (f)σ_{max}(33.1 dB,[20]). (g)σ_{max} (56.3 dB,ours). (h)$\sigma_{max} = \Lambda_{max}$.

Fig. 2. Highlight removal on a synthetic image. From (a)-(d): input image, diffuse reflection extracted using [20][1], diffuse reflection extracted using our method and the ground truth. (e) is our approximation (Eqn. 12) of the maximum diffuse chromaticity Λ_{max}. Unlike the "pseudo-coded" diffuse image presented in [20], our approximation is invariant to surface geometry. (e) is used as the guidance image for the bilateral filtering process in Algorithm 1. (f)-(h) present the maximum chromaticity computed from (b)-(d), respectively. As can be seen in (b) and (f), [20] has two main problems for this data set: (i) pixels around highlight edges remain specular although there is no visible discontinuities around the highlight regions in the estimated maximum diffuse chromaticity as presented in (f); (ii) [20] propagates the maximum chromaticities from diffuse pixels to specular pixels. However, the fact that the propagation does not stop at color edges results in extracting incorrect diffuse reflections. Our method does not have these problems as can be seen in (c) and (g). The PSNR value computed from (b) and (d) is 25.2 dB, and 52.1 dB from (c) and (d).



(a)Input. (b)Diffuse ([20]). (c)Diffuse (ours).

(d)Input. (e)Diffuse ([20]). (f)Diffuse (ours).

Fig. 3. Highlight removal on the images provided in [20][1]. From left to right: input image, diffuse reflection extracted using [20], diffuse reflection extracted using our method. Visual comparison shows that our method is comparable to [20] for these data sets but much faster.

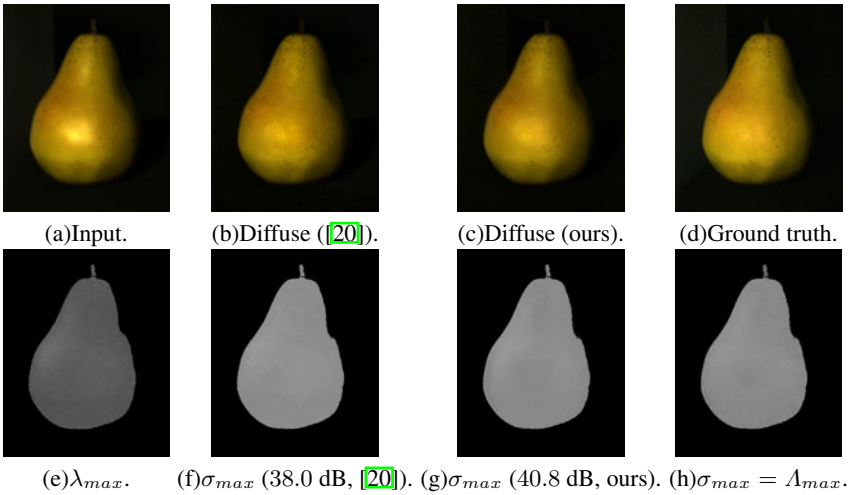
as suggested in [10]. The PSNR values obtained using [20] and our method are 25.2 and 52.1 dB, respectively, which agree with our subjective evaluation that there is no visible difference between the extracted diffuse reflection using our method (Fig. 2 (c)) and the ground truth (Fig. 2 (d)). Fig. 2 (e) presents our approximation (Eqn. 12) of the maximum diffuse chromaticity Λ_{max} which is used as the guidance image for the bilateral filtering process in Algorithm 1. Fig. 2 (f)-(h) present the maximum chromaticity computed from (b)-(d), respectively.

Fig. 1 and 3 compare our method with [20] using the images provided by the author of [20]. Visual comparison shows that the performance of the two methods is similar while no ground truth is available for quantitative comparison.

Finally, we conducted experiments on two real images with ground-truth diffuse reflections captured by polarizing filters over the camera and the light source. The experimental results are presented in Fig. 4 and 5. The ground-truth diffuse reflections are presented in Fig. 4 (d) and 5 (d). Fig. 4 (e) and 5 (e) present our approximations (Eqn. 12) of the maximum diffuse chromaticities Λ_{max} which are used as the guidance image for the bilateral filtering process in Algorithm 1. We next computed the maximum chromaticity σ_{max} using Fig. 4 (b)-(d) and 5 (b)-(d), respectively, and presented the results in Fig. 4 (f)-(h) and 5 (f)-(h). The dark pixels are excluded to avoid quantization noise. We next compared (f) and (g) in Fig. 4 and 5 with the ground-truth maximum diffuse chromaticity images presented in Fig. 4 (h) and Fig. 5 (h) since the diffuse reflection is represented as a function of the maximum diffuse chromaticity Λ_{max}. Accurate Λ_{max}

results in accurate diffuse reflection. We don't use the PSNR values computed from the ground-truth diffuse reflection in Fig. 4 (d) and Fig. 5 (d) because the amount of light captured with and without the polarizing filters are different. The PSNR values computed from Fig. 4 (f) and (g) are 38.0 and 40.8 dB, which shows that both [20] and our method are suitable for single-color objects. The PSNR values computed from Fig. 5 (f) and (g) are 21.2 and 33.1 dB. Hence, the estimated diffuse reflection using our method (Fig. 5 (g)) is not very accurate although it is a bit better than [20] (Fig. 5 (f)). However, if we exclude pixels with maximum chromaticity less than or equal to 0.37, which means that most of the pixels near neutral (our method is limited to chromatic surfaces) are excluded, the PSNR values obtained using [20] and our method are raised to 24.7 and 40.7 dB, respectively. In this case, the diffuse reflection computed using our method is comparable to the ground truth as PSNR larger than 40 dB often corresponds to almost invisible differences. However, the diffuse reflection extracted using the method presented in [20] is still of low quality as can be seen from the obtained PSNR values. Both visual and numerical comparison on real images show that our method is a bit more robust/accurate than [20], but both methods are invalid for grayscale surfaces.

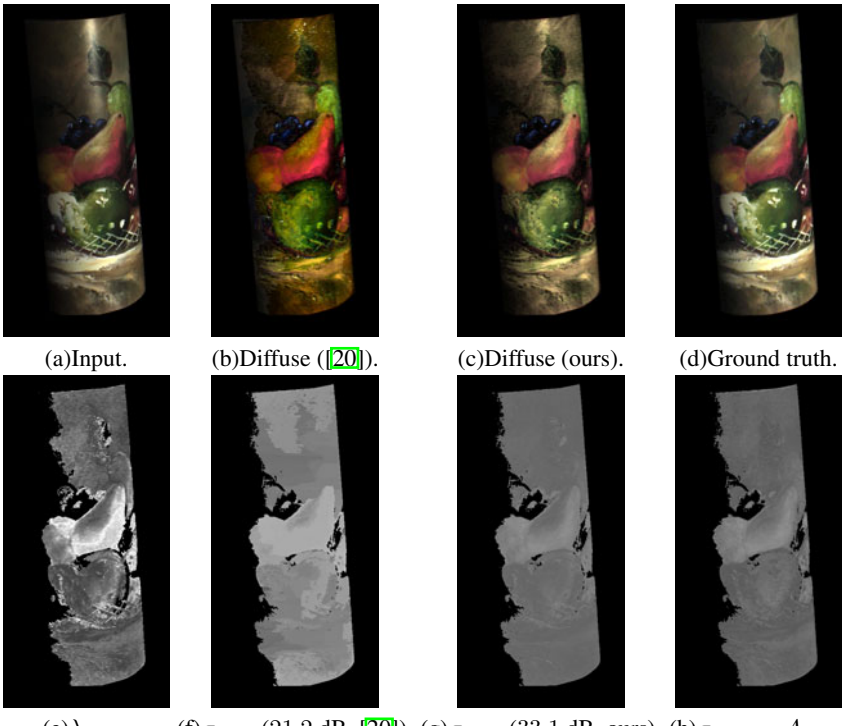
Fig. 6 compares the runtime of the method presented in [20] and ours. Note that the speeeup factor of our method is generally over 200.



(a)Input. (b)Diffuse ([20]). (c)Diffuse (ours). (d)Ground truth.

(e)λ_{max}. (f)σ_{max} (38.0 dB, [20]). (g)σ_{max} (40.8 dB, ours). (h)$\sigma_{max} = \Lambda_{max}$.

Fig. 4. Highlight removal on a low-textured image. From (a)-(d): input image, diffuse reflection extracted using [20][1], diffuse reflection extracted using our method and the ground truth. (e) is our approximation (Eqn. 12) of the maximum diffuse chromaticity Λ_{max}. (e) is used as the guidance image for the bilateral filtering process in Algorithm 1. (f)-(h) are the maximum chromaticity computed from (b)-(d), respectively. The numbers under (f)-(g) are PSNR values computed by comparing with the ground truth in (h), which numerically proves both [20] and our method are suitable for low-textured scenes. The images are gamma corrected for better illustration.



(a)Input. (b)Diffuse ([20]). (c)Diffuse (ours). (d)Ground truth.

(e)λ_{max}. (f)σ_{max} (21.2 dB, [20]). (g)σ_{max} (33.1 dB, ours). (h)$\sigma_{max} = \Lambda_{max}$.

Fig. 5. Highlight removal on a heavy-textured image. From (a)-(d): input image, diffuse reflection extracted using [20][1], diffuse reflection extracted using our method and the ground truth. (e) is our approximation (Eqn. 12) of the maximum diffuse chromaticity Λ_{max}. (e) is used as the guidance image for the bilateral filtering process in Algorithm 1. (f)-(h) are the maximum chromaticity computed from (b)-(d), respectively. The numbers under (f)-(g) are PSNR values computed by comparing with the ground truth in (h). Note that neither our method nor [20] is of high quality when comparable to the ground truth as many pixels are close to neutral. Nevertheless, the PSNR value computed from our method is still a bit higher than the value computed from [20]. Moreover, visually comparison shows that the color of extracted diffuse reflection using [20] is a bit less accurate. The images are gamma corrected for better illustration.

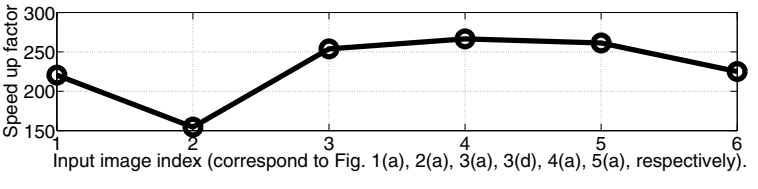


Fig. 6. Speed comparison. This figure shows that our method runs over 200× faster than [20] on a standard CPU except for the second data set, which is the synthetic image presented in Fig. 2 (a). But note that [20] obtains incorrect diffuse reflection from this data set, thus probably, the iterative algorithm presented in [20] stops before convergence.

4 Conclusions

We have proposed a new highlight removal model in the paper. Using a single color image, the highlight removal problem is formulated as an iterative bilateral filtering process which normally converges in 2 to 3 iterations. Unlike previous methods, the presented technique can process high-resolution images at video rate thus is suitable for real-time applications, *e.g.*, stereo matching for specular surfaces. Besides, this technique does not result in noticeable artifacts, and guarantees that the estimated diffuse reflections will be locally smooth.

References

1. Bajcsy, R., Lee, S., Leonardis, A.: Detection of diffuse and specular interface reflections and inter-reflections by color image segmentation. IJCV 17(3), 241–272 (1996)
2. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. In: Siggraph, vol. 21 (2002)
3. Klinker, G., Shafer, S., Kanade, T.: The measurement of highlights in color images. IJCV 2(1), 7–32 (1988)
4. Lee, S., Bajcsy, R.: Detection of specularity using color and multiple views. In: Sandini, G. (ed.) ECCV 1992. LNCS, vol. 588, pp. 99–114. Springer, Heidelberg (1992)
5. Lin, S., Li, Y., Kang, S., Tong, X., Shum, H.Y.: Diffuse-specular separation and depth recovery from image sequences. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 210–224. Springer, Heidelberg (2002)
6. Lin, S., Shum, H.Y.: Separation of diffuse and specular reflection in color images. In: CVPR, pp. 341–346 (2001)
7. Mallick, S.P., Zickler, T., Belhumeur, P.N., Kriegman, D.J.: Specularity removal in images and videos: A pde approach. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 550–563. Springer, Heidelberg (2006)
8. Mallick, S.P., Zickler, T.E., Kriegman, D.J., Belhumeur, P.N.: Beyond lambert: Reconstructing specular surfaces using color. In: CVPR, pp. II619–II626 (2005)
9. Nayar, S., Fang, X., Boult, T.: Separation of reflection components using color and polarization. IJCV 21(3) (1996)
10. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 568–580. Springer, Heidelberg (2006)
11. Park, J., Tou, J.: Highlight separation and surface orientation for 3-d specular objects. In: ICPR, pp. I331–I335 (1990)
12. Porikli, F.: Constant time o(1) bilateral filtering. In: CVPR (2008)
13. Sato, Y., Ikeuchi, K.: Temporal-color space analysis of reflection. JOSA 11(11), 2990–3002 (1994)
14. Shafer, S.: Using color to separate reflection components. Color Res. App. 10(4), 210–218 (1985)
15. Shen, H.L., Cai, Q.Y.: Simple and efficient method for specularity removal in an image. Applied Optics 48(14), 2711–2719 (2009)
16. Tan, P., Lin, S., Quan, L., Shum, H.Y.: Highlight removal by illumination-constrained inpainting. In: ICCV, p. 164 (2003)
17. Tan, P., Quan, L., Lin, S.: Separation of highlight reflections on textured surfaces. In: CVPR, pp. 1855–1860 (2006)

18. Tan, R.: Highlight removal from a single image, http://www.commsp.ee.ic.ac.uk/~rtan/code.html
19. Tan, R., Ikeuchi, K.: Reflection components decomposition of textured surfaces using linear basis functions. In: CVPR, pp. I125–I131 (2005)
20. Tan, R., Ikeuchi, K.: Separating reflection components of textured surfaces using a single image. PAMI 27(2), 178–193 (2005)
21. Tan, R.T., Nishino, K., Ikeuchi, K.: Illumination chromaticity estimation using inverse-intensity chromaticity space. In: CVPR, pp. 673–680 (2003)
22. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV, pp. 839–846 (1998)
23. Yang, Q., Tan, K.H., Ahuja, N.: Real-time o(1) bilateral filtering. In: CVPR (2009)
24. Yang, Q., Wang, S., Ahuja, N.: SVM for Edge-Preserving Filtering. In: CVPR (2010)

A Proof of Theorem 1

For any two pixels p and q, let

$$J_{min} = \min(J_r, J_g, J_b), \tag{15}$$
$$J^D_{min} = \min(J^D_r, J^D_g, J^D_b), \tag{16}$$
$$\tag{17}$$

according to the definition of λ_c in Eqn. 11, we obtain

$$\begin{aligned}
\lambda_c &= (\sigma_c - \sigma_{min})/(1 - 3\sigma_{min}) \qquad (18)\\
&= \Big(\frac{J_c}{\sum_{u\in\{r,g,b\}} J_u} - \frac{J_{min}}{\sum_{u\in\{r,g,b\}} J_u}\Big)\\
&\quad / \Big(\frac{\sum_{u\in\{r,g,b\}} J_u}{\sum_{u\in\{r,g,b\}} J_u} - \frac{3\cdot J_{min}}{\sum_{u\in\{r,g,b\}} J_u}\Big)\\
&= \Big(\frac{J_c - J_{min}}{\sum_{u\in\{r,g,b\}} J_u}\Big)/\Big(\frac{\sum_{u\in\{r,g,b\}}(J_u - J_{min})}{\sum_{u\in\{r,g,b\}} J_u}\Big)\\
&= \frac{J_c - J_{min}}{\sum_{u\in\{r,g,b\}}(J_u - J_{min})}\\
&= \frac{(J^D_c + J^S) - (J^D_{min} + J^S)}{\sum_{u\in\{r,g,b\}}((J^D_u + J^S) - (J^D_{min} + J^S))}\\
&= \frac{J^D_c - J^D_{min}}{\sum_{u\in\{r,g,b\}}(J^D_u - J^D_{min})}\\
&= \Big(\frac{J^D_c - J^D_{min}}{\sum_{u\in\{r,g,b\}} J^D_u}\Big)/\Big(\frac{\sum_{u\in\{r,g,b\}}(J^D_u - J^D_{min})}{\sum_{u\in\{r,g,b\}} J^D_u}\Big)\\
&= (\Lambda_c - \Lambda_{min})/\Big(\sum_{u\in\{r,g,b\}}(\Lambda_u - \Lambda_{min})\Big). \qquad (19)
\end{aligned}$$

For any two pixels p and q, if $\Lambda_c(p) = \Lambda_c(q)$, Eqn. (19) ensures that $\lambda_c(p) = \lambda_c(q)$.

B Proof of Theorem 2

For any two pixels p and q, assume $\lambda_c(p) = \lambda_c(q)$, from Equation (19) we obtain

$$\frac{\Lambda_c(p) - \Lambda_{min}(p)}{1 - 3\Lambda_{min}(p)} = \frac{\Lambda_c(q) - \Lambda_{min}(q)}{1 - 3\Lambda_{min}(q)}. \tag{20}$$

If $\Lambda_{min}(p) = \Lambda_{min}(q)$, from (20) we obtain $\Lambda_c(p) = \Lambda_c(q)$.

C Proof of Theorem 3

let $J_{max} = \max(J_r, J_g, J_b)$, according to the definition of σ_{max},

$$\sigma_{max} = \frac{J_{max}}{\sum_{u\in\{r,g,b\}} J_u}, \tag{21}$$

$$= \frac{J^D_{max} + J^S}{\sum_{u\in\{r,g,b\}} J^D_u + 3J^S} \tag{22}$$

$$= (\Lambda_{max} + \frac{J^S}{\sum_{u\in\{r,g,b\}} J^D_u})/(1 + \frac{3J^S}{\sum_{u\in\{r,g,b\}} J^D_u}) \tag{23}$$

$$= \Lambda_{max} + \frac{J^S}{\sum_{u\in\{r,g,b\}} J^D_u} - \sigma_{max}\frac{3J^S}{\sum_{u\in\{r,g,b\}} J^D_u} \tag{24}$$

$$= \Lambda_{max} + (1 - 3\sigma_{max})\frac{J^S}{\sum_{u\in\{r,g,b\}} J^D_u}. \tag{25}$$

According to its definition, $\sigma_{max} \geq 1$, thus

$$\sigma_{max} - \Lambda_{max} \leq 0, \tag{26}$$

and the quality holds when the $\Lambda_{max} = \sigma_{max} = \frac{1}{3}$.

Learning Artistic Lighting Template from Portrait Photographs

Xin Jin[1], Mingtian Zhao[2,3], Xiaowu Chen[1,⋆],
Qinping Zhao[1], and Song-Chun Zhu[2,3]

[1] State Key Laboratory of Virtual Reality Technology and Systems,
School of Computer Science and Engineering, Beihang University, Beijing, China
jinxin@vrlab.buaa.edu.cn,
chen@buaa.edu.cn
[2] Lotus Hill Institute, Ezhou, China
[3] Department of Statistics, University of California, Los Angeles, USA
{mtzhao,sczhu}@stat.ucla.edu

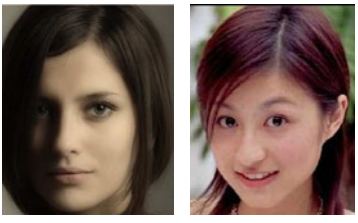
Abstract. This paper presents a method for learning artistic portrait lighting template from a dataset of artistic and daily portrait photographs. The learned template can be used for (1) classification of artistic and daily portrait photographs, and (2) numerical aesthetic quality assessment of these photographs in lighting usage. For learning the template, we adopt Haar-like local lighting contrast features, which are then extracted from pre-defined areas on frontal faces, and selected to form a log-linear model using a stepwise feature pursuit algorithm. Our learned template corresponds well to some typical studio styles of portrait photography. With the template, the classification and assessment tasks are achieved under probability ratio test formulations. On our dataset composed of 350 artistic and 500 daily photographs, we achieve a 89.5% classification accuracy in cross-validated tests, and the assessment model assigns reasonable numerical scores based on portraits' aesthetic quality in lighting.

1 Introduction

The word *photography*, first used in 1839 by Sir John Herschel, came from two Greek words *photos* (light) and *graphé* (drawing) [1]. Just like brushes and pigments of painters, light is the major tool of photographers for creating beautiful pictures. The art of lighting in photography has been so heavily influencing the aesthetics of photographs, that learning to manipulate lighting is the doorway to high-quality photography. Especially, understanding how light works and having an appreciation of good lighting is at the root position of photographer training [2]. The goal of this paper is to enable computers to *appreciate* good lighting, namely, to distinguish artistic photographs with outstanding lighting composition from daily (commonplace) ones (as shown in Fig.1), and to quantitatively assess their visual aesthetics quality in lighting usage.

⋆ Corresponding author.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 101–114, 2010.



(a) (b)

Fig. 1. Two portrait photographs(a) is usually considered as an artistic photograph, while (b) is often considered as a daily one

In recent literature, classification of images according to the quality of visual aesthetics, including photographs [3,4,5,6,7], paintings [8], etc., and numerical quality assessment of them [3,5], have attracted increasing interest. Most existing work towards this interest follows three steps: (1) collecting an image dataset according to specific objectives, and separating it into two subsets containing the "good" and "bad" images, respectively, (2) designing various features and extracting them from these collected images, and (3) training a classifier to automatically judge a test image's quality class, or fitting a model to data with consensus scores of aesthetics quality from various training sources in order to assess the test image numerically. In early work [3,4] only global features were used, then Datta et al. [5] considered local features within image regions, and a few later studies [6,7,8] also included features extracted between regions. However, most of them did not use content-oriented features (e.g., features specially designed for portrait, landscape), which limited their performance, because aesthetic metrics usually vary a lot over diverse image contents [2,9,10,11].

As an attempt for content-oriented design towards the problem, this paper focuses on analyzing the lighting of portrait photography. Although factors such as pose, personality, expression, etc., no doubt affect the aesthetics, the use of lighting itself plays a key role. According to professional studies on portrait photography [2,9,10,11], lights and shadows on the face, with their relative locations, their area ratios, etc., are the dominant facts of attraction, which also make the main contribution to 3D perception of the 2D image on the photograph. Fig.2 includes four typical lighting styles of artistic portrait photographs [2,9,10,11]. The *Rembrandt* style is usually implemented with sidelight. It is featured by a triangular highlight below one of the eyes, with its surrounding areas in dark shadows. The *Paramount* style has a butterfly shape for the shadow between nose and mouth, which is achieved with a key light in front of and above the model's face. The *Loop* style is named after the loop-shaped shadow below the nose. Light position for this style should be between those of Rembrandt and Paramount styles. And with a more extreme sidelight than that of the Rembrandt style, the *Split* style has half of the face in shadow. Such categorization of styles inspired us that there probably exists an *artistic lighting template* for each style, specified by the local contrast within the parts of the face. Thus we divide the area of a frontal face into 16 rectangular parts as shown in Fig.3. Choosing rectangles to represent facial parts is

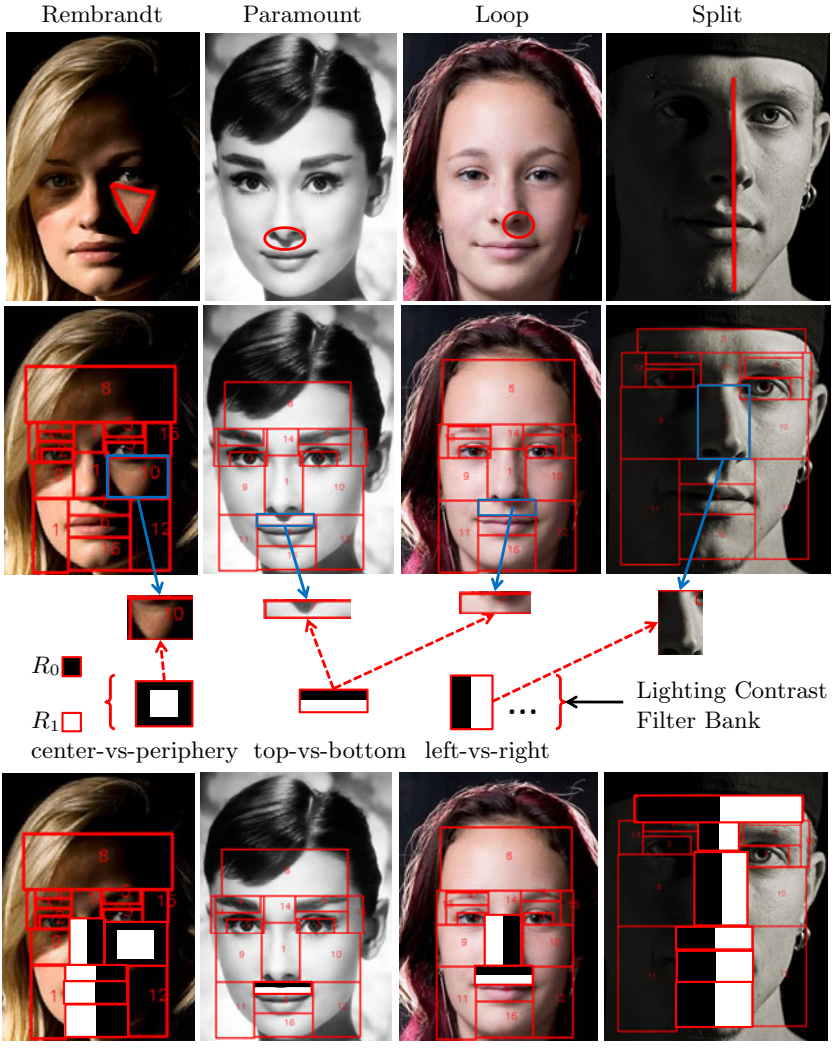
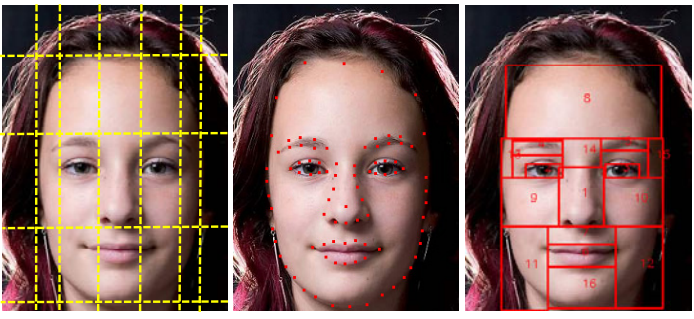


Fig. 2. Artistic lighting templates. Top row: four typical artistic lighting styles of portrait photography. Middle rows: Haar-like lighting contrast filters for measuring the local contrast. Large local contrast can usually be captured by one of the filters, for example, the Rembrandt style can be captured by the center-vs-periphery filter applied below one of the eyes. Bottom row: example lighting templates for the styles, designed manually for illustration.

for computational efficiency. We then adopt Haar-like features within each rectangle to capture its local lighting contrast, which are extracted by a bank of filters of different contrast types, target channels and statistics. The last row of Fig. 2 shows the corresponding example lighting templates with the most distinctive features for the above artistic lighting styles (note: features are selected manually for illustration).



(a) (b) (c)

Fig. 3. (a) A demonstration of professional photographers' common practice to assess portrait photographs' lighting quality. The face is coarsely divided into 3×5 grids according to the positions of hair, eyes, eyebrows, etc. Lights and shadows in these grids are the criteria for lighting usage. Inspired by professional photographers, we do face alignment (b) to detect feature points, then divide the face into 16 rectangular parts (c) corresponding to the nose, mouth, eyes, etc.

We learn an artistic portrait template for these styles composed of local lighting contrast features using a stepwise feature pursuit algorithm. The learning algorithm is built on a log-linear model of the probability distributions of artistic portraits, with the feature responses as factors. The learned template can be used for classifying artistic and daily portrait photographs in terms of lighting usage. Furthermore, we use human experiments to obtain consensus scores of portrait lighting to which we fit a regression model with our selected features for predicting the aesthetic quality of a portrait photograph's lighting.

The contributions of this paper include: (1) a learned common artistic lighting template of portrait photographs, (2) a method for modeling the lighting usage of portrait photographs, with content-oriented designs of features and template, and (3) evaluations of the strengths and weaknesses of our methods and designs by applying them on a dataset containing 350 artistic and 500 daily photographs.

2 Feature Design

We adopt Haar-like features to capture the lighting characteristics on important parts of the face. Each feature F consists of four components.

1. Spatial contrast type T, which can be left-vs-right, top-vs-bottom, or center-vs-periphery, as shown in Fig.2 (above the bottom row).
2. Rectangular region R on the image lattice. According to the type T, region R is divided into two equally sized subregions R_0 and R_1 as shown in black and white areas in Fig.2.
3. Target channel C of the image. Our adopted channels include the graylevel, the three channels L, a and b in the CIE 1976 (L^*,a^*,b^*) color space (note:

L differs slightly from graylevel although the two channels are heavily correlated), and the hue and saturation channels in the HSV color space. In order to capture the effect of staggered highlight under sidelight, we also involve the channel of graylevel gradients, as well as an edge channel (see component 4 below) including all edge pixels generated by a Canny edge detector [12].

4. Target statistic S of the image. We consider 3 types of statistics: (1) mean value μ, (2) histogram h, and (3) density ρ which was specially designed for the edge channel (i.e., the proportion of edge pixels).

With the above feature design, we define the local contrast r between the statistics μ, h and ρ of the two subregions R_0 and R_1 as

$$r_\mu = |\mu_1 - \mu_0| \ , \qquad r_h = \mathrm{JS}(h_1||h_0) \ , \qquad r_\rho = |\rho_1 - \rho_0| \ , \tag{1}$$

where $\mathrm{JS}(\cdot||\cdot)$ denotes the discrete Jensen-Shannon divergence [13]. We use r as the feature response of F in our model.

3 Template Learning

Let Ω_A denote the set of artistic portrait photographs, and Ω_D denote the set of daily ones, we would like to build a template for Ω_A against Ω_D, as well as a probability distribution upon this template. A template is a group of features which characterize the artistic photographs. Suppose the template is composed of a set of features $\{F_1, \cdots, F_K\}$, a probabilistic model for each image $\mathbf{I} \in \Omega_A$ can be defined in a log-linear form [14,15] as

$$p(\mathbf{I}) = q(\mathbf{I}) \prod_{k=1}^{K} \frac{1}{z_k} \exp\{\lambda_k r_k(\mathbf{I})\} \ , \tag{2}$$

in which $q(\mathbf{I})$ is the null distribution of $\mathbf{I}$ without any knowledge of the feature responses r_k, λ_k are weight parameters, and z_k are normalizing constants for the factors. For our case, we use the template to describe the new information of photographs in Ω_A compared with those in Ω_D, and leave the rest information in $q(\mathbf{I})$. In this way, $q(\mathbf{I})$ models the daily photographs.

We are interested in selecting an informative subset of the features for the template, rather than using the whole feature set, for two reasons: (1) features tend to be correlated, and (2) we prefer a simple template for both capability of generalization and computational efficiency. Since selecting an optimal subset of features at once is a non-trivial task, we adopt a stepwise feature pursuit algorithm [16,15] to select one feature at a time to construct $p(\mathbf{I})$.

Given a candidate feature set, we begin with an empty template corresponding to the daily photo distribution $p_0(\mathbf{I}) = q(\mathbf{I})$ at step 0. Then at each step t, we choose the max-gain feature

$$\begin{aligned} F_{(t)} &= \arg\max_{F_k} \mathrm{KL}(p_t(r_k(\mathbf{I}))||p_{t-1}(r_k(\mathbf{I}))) \\ &\approx \arg\max_{F_k} \mathrm{KL}(p_t(r_k(\mathbf{I}))||q(r_k(\mathbf{I}))) \\ &\approx \arg\max_{F_k} \mathrm{KL}(h_A(r_k(\mathbf{I}))||h_D(r_k(\mathbf{I}))) \end{aligned} \tag{3}$$

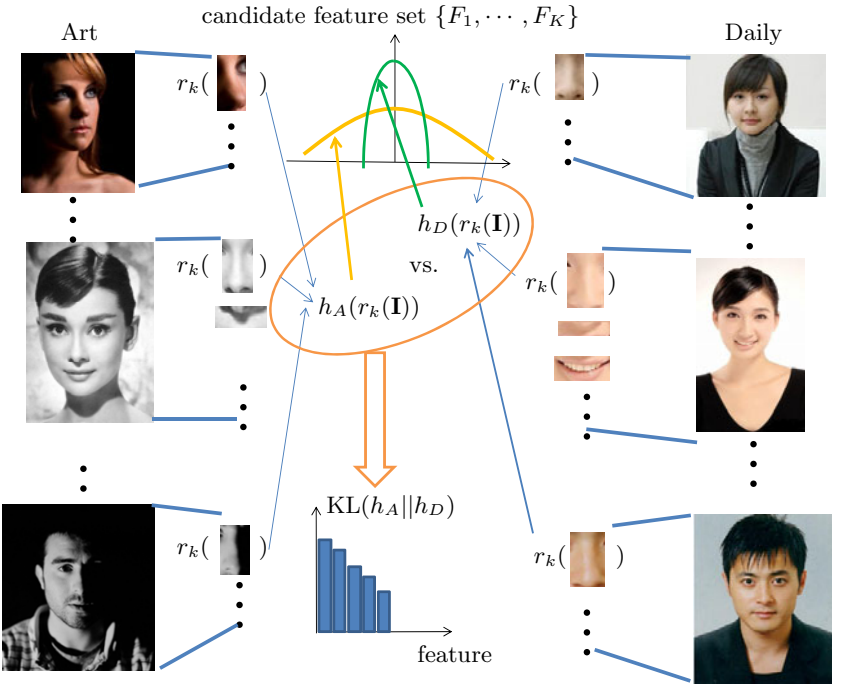


Fig. 4. Template learning. For each feature F_k in the candidate feature set, its responses $r_k(\mathbf{I})$ is calculated for all images in Ω_A and Ω_D, and two histograms $h_A(r_k(\mathbf{I}))$ and $h_D(r_k(\mathbf{I}))$ are obtained for the artistic and daily photographs, respectively. Then the KL divergence between the two histograms can be computed, as an approximation to the information gain of F_k on our dataset.

on our dataset, where $\mathrm{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence, and $h_A(\cdot)$ and $h_D(\cdot)$ are the histograms over Ω_A and Ω_D, respectively, as shown in Fig.4. In Eq.(3), the second approximation applies empirical estimates of marginal probabilities with instances in the dataset. In order for the first approximation, where we assume $p_{t-1}(r_k(\mathbf{I})) \approx q(r_k(\mathbf{I}))$, to be feasible, we apply local inhibition in the template learning process to reduce the correlations among the selected features in the pursuit steps. Noticing the fact that the rectangular regions have no or very little overlaps with each other, we simply assume independence of features in different regions, and use the inhibition strategy that selects only one feature from each rectangular part of the face. Meanwhile, for step t, the parameters $\lambda_{(t)}$ and $z_{(t)}$ can be computed by solving the system

$$\begin{aligned} \mathrm{E}_q\left[\frac{1}{z_{(t)}}\exp\{\lambda_{(t)}r_{(t)}(\mathbf{I})\}r_{(t)}(\mathbf{I})\right] &= \mathrm{E}_{p_t}[r_{(t)}(\mathbf{I})] = \mathrm{E}_f[r_{(t)}(\mathbf{I})] \\ \mathrm{E}_q[\exp\{\lambda_{(t)}r_{(t)}(\mathbf{I})\}] &= z_{(t)} \end{aligned} \tag{4}$$

with $\mathrm{E}_q[\cdot] \approx \mathrm{Mean}_{\Omega_D}(\cdot)$ and $\mathrm{E}_f[\cdot] \approx \mathrm{Mean}_{\Omega_A}(\cdot)$ as empirical estimates according to our dataset. The feature pursuit process stops when the Bayesian Information Criterion (BIC) of the model reaches its minimum [17].

4 Inference for Classification and Assessment

4.1 Classification

Since the template is built for artistic photographs against daily ones, besides modeling the former, it can also be used for classifying these two types. Given a test portrait photo $\mathbf{I}$, we can calculate a template matching score with

$$\text{MatchingScore}(\mathbf{I}) = \log \frac{p(\mathbf{I})}{q(\mathbf{I})} = \sum_{k=1}^{K} (\lambda_k r_k(\mathbf{I}) - \log z_k) \ . \tag{5}$$

This follows a probability ratio test formulation. If the matching score is greater than a learned threshold corresponding to a certain significance level (e.g., the equal error rate (EER) threshold), the test photo is classified as an art photo, otherwise it is classified as a daily one.

This classification algorithm differs from AdaBoost [18] although the two have similar formulations. In AdaBoost, each $r_k(\mathbf{I})$ is a binary classifier, while in the above method it is a continuous feature response. Using a continuous $r_k(\mathbf{I})$ augmented from binary classifier has two advantages: (1) it can usually lead to a model with relatively smaller number of features since $r_k(\mathbf{I})$ becomes more informative, and (2) the model can be extended to predict a meaningful continuous score (see the next section).

4.2 Numerical Assessment

In addition to classification, it is usually more useful to have a reasonable numerical assessment of an input photo on its aesthetic quality. This can be achieved by extending our learning algorithm to a regression framework.

In an empirical manner, we define the quality of a portrait photograph $\mathbf{I}$ as the probability p that it is better than another random chosen portrait photograph $\mathbf{J}$, namely,

$$\text{QualityScore}(\mathbf{I}) = p = \text{E}_{f(\mathbf{J})} \left[\mathbf{1}(\mathbf{I} \text{ wins against } \mathbf{J} \text{ in quality}) \right] \ , \tag{6}$$

where f is the distribution of all portrait photographs (either artistic or daily), and $\mathbf{1}(\cdot)$ is the indicator function. For predicting the score p, we randomly choose a fixed number n of photographs to compare with $\mathbf{I}$, then the number of wins of $\mathbf{I}$ should follow a binomial distribution $\text{binom}(n, p)$. We do such comparison experiments on many test images and obtain their scores, then estimate the effects of the features on the score using logistic regression [19], by fitting the model

$$\log \frac{p}{1-p} = \sum_{k=1}^{K} (\lambda_k r_k(\mathbf{I}) - \log z_k) = \lambda_0 + \sum_{k=1}^{K} \lambda_k r_k(\mathbf{I}) \ . \tag{7}$$

This model is able to output a score $p \in (0, 1)$ for the quality of test photographs.

5 Experiments

5.1 Data Collection

Prior to experiments on learning and evaluating the template, we collect a image dataset to work on. We collect 350 artistic portrait photographs from 3 sources: (1) masterpieces of portrait photography from famous photographers (e.g., Yousuf Karsh, Arnold Newman), (2) collections from professional photography websites (e.g., photo.net, portrait-photos.org), and (3) scanned copies from professional portrait photography books focusing on lighting [2,9,10,11].

The 500 daily photographs for comparison are obtained from 2 main sources: (1) popular photo hosting websites (e.g., flickr.com), and (2) image search engines' results for the keywords "face" and "daily life" (e.g., images.google.com).

The collected dataset tries to randomize irrelevant factors by spanning over various poses, ethnic groups, etc. All images in our dataset are aligned using AAM [20] to a standard frontal face before entering the algorithms. Sometimes manual corrections are needed for better accuracy after automatic alignment due to shadow effects.

5.2 The Learned Template

Fig.5 displays the set of candidate features and a learned template composed of the selected 12 most informative features. The 12 rectangular parts cover most of the area on the face. Most of the selected features are of the left-vs-right spatial type. This matches the common lighting strategies for portrait photography in studios: photographers usually change the directions of light in the horizontal dimension. The parts for nose and the area between mouth and nose are the top significant areas, which makes sense as these two parts are relatively complex in geometry and have various appearances under different illumination conditions. It is worth noticing that all color features (except one on saturation) are ignored by the template learning process, which in turn proves the conjecture that lighting is the major factor of artistic portrait photography.

5.3 Classification Results

Fig.6 and Fig.7 shows the performance of our method in classifying artistic and daily portrait photographs. Running 5-fold cross-validation, an average performance is displayed with the ROC curve in Fig.6. The average classification accuracy is 89.5% in these experiments using the EER threshold, which is quite good considering the relatively small training sample size. Look at the failure cases in Fig.7—some are indeed hard to classify unless information such as poses and expressions is involved.

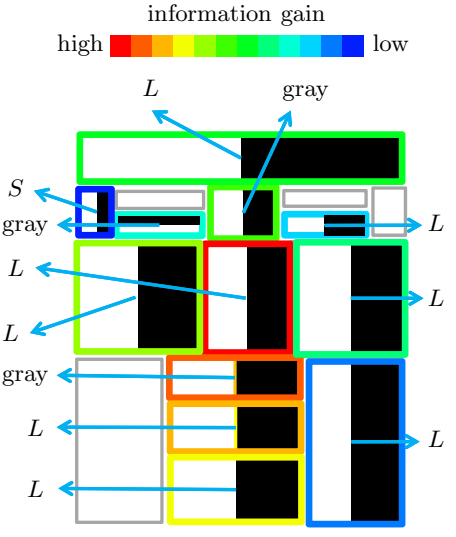
5.4 Numerical Assessment Results

For the numerical assessment task, training data with the quality of photographs are necessary for fitting the regression model (see Section 4.2). We obtain the

14 filters × 16 parts = 224 features

Contrast Type	Statistic	Channel
left-vs-right	mean	graylevel
left-vs-right	histogram	graylevel
left-vs-right	mean	gradient
left-vs-right	histogram	gradient
left-vs-right	density	edge
left-vs-right	mean	H
left-vs-right	mean	S
left-vs-right	histogram	S
left-vs-right	mean	L
left-vs-right	histogram	L
left-vs-right	mean	a
left-vs-right	mean	b
top-vs-bottom	mean	graylevel
⋮	⋮	⋮

(a) candidate feature set



(b) learned template

Fig. 5. (a) The feature set with 224 candidate features: for each of the 16 rectangular parts of the faces, we have 14 local contrast filters. (b) A learned common artistic portrait lighting template composed of the 12 most informative features, shown with ranks marked by colored boundaries corresponding to the legend. The target channel of the selected features are also marked. As for target statistics, all selected features are on means.

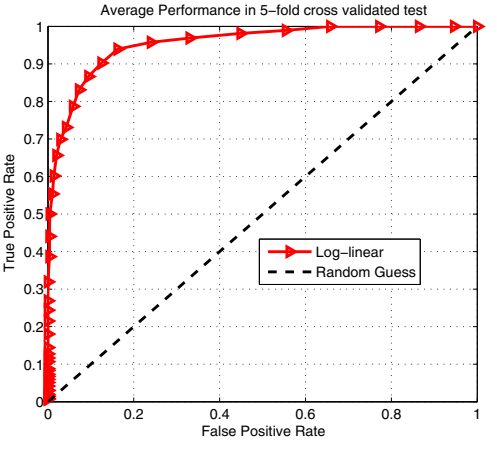


Fig. 6. The classification performance of the log-linear model with a fixed number of 12 features. The ROC curve displays the average performance in the 5-fold cross-validated tests.

Fig. 7. Examples of classification results

Table 1. Logistic Regression Coefficients

Feature	Est.	Std.Err.	z-score	p-value	Feature	Est.	Std.Err.	z-score	p-value
(I)	−2.04	0.14	−14.82	< 0.01	F_7	1.99	0.46	4.27	< 0.01
F_1	8.39	1.29	6.51	< 0.01	F_8	4.45	0.69	6.45	< 0.01
F_2	0.05	0.74	0.07	0.95	F_9	2.06	0.74	2.78	0.01
F_3	−3.47	1.98	−1.75	0.08	F_{10}	−1.88	0.66	−2.84	< 0.01
F_4	−0.09	0.86	−0.11	0.91	F_{11}	−0.22	0.33	−0.66	0.51
F_5	1.57	0.61	2.57	0.01	F_{12}	−4.04	0.77	−5.26	< 0.01
F_6	0.32	0.80	0.40	0.69					

consensus quality scores (i.e., winning probabilities against randomly chosen images) with human experiments.

Human Experiments. We randomly choose 50 photographs (either artistic or daily) from our dataset as training examples. We also choose 10 graduate students of various majors as test subjects to do the comparisons between photographs.

For each $\mathbf{I}$ in the 50 training images, another 50 random images $\mathbf{J}_1, \cdots, \mathbf{J}_{50}$ from the rest of the dataset are sampled with replacement. Each of the 50 pairs $(\mathbf{I}, \mathbf{J}_1), \cdots, (\mathbf{I}, \mathbf{J}_{50})$ is displayed to a random test subject, who will compared the two images and report their relative rank order in the quality of lighting. In this way, a total of 2500 comparisons are performed, and the numbers of wins and losses for the 50 training images are obtained for fitting the prediction model in Section 4.2. Here the replacement ensures the constant probability condition for assuming a binomial distribution [19].

Regression. Most features have favorable small p-values in the logistic regression (see Table 1). Fig.8 plots the fitted scores vs. the scores obtained from human experiments. Due to the small size of the training set, and the small

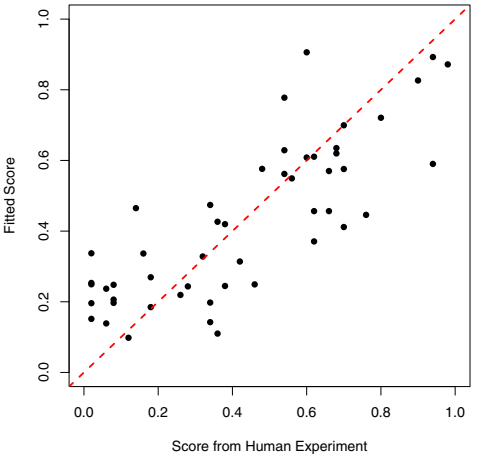


Fig. 8. Goodness-of-fit visualization of the logistic regression for quality assessment

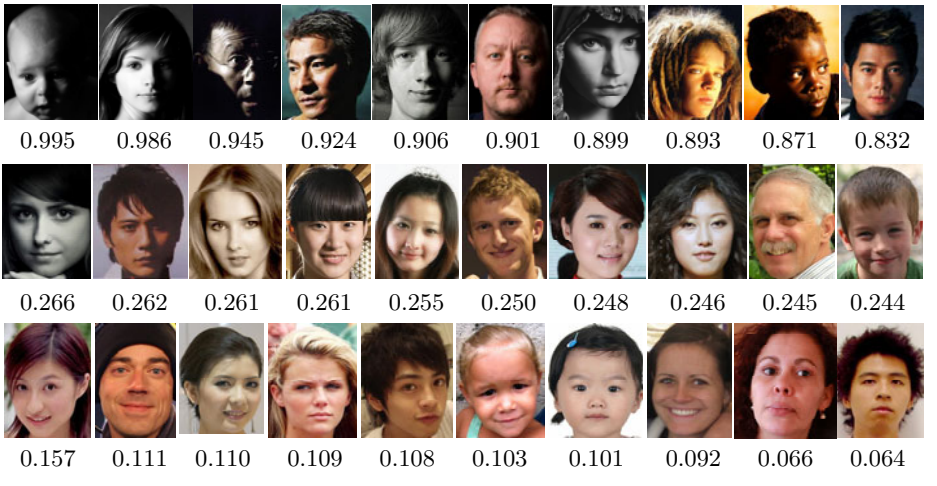


Fig. 9. Examples of numerical assessment: the top-10, middle-10, and bottom-10 photographs are displayed with predicted quality scores

number of Bernoulli trials, there are still a considerable residual deviance in the fitting (null deviance: 936.85 on 49 degrees of freedom, residual deviance: 398.50 on 37 degrees of freedom). But since no heteroskedasticity or non-normal effects are noticed [19], we believe that our empirical definition for the aesthetic quality and the experimental design should make good sense.

Fig.9 shows examples of prediction, which includes the top-10, middle-10, and bottom-10 photographs predicted by our regression model, for the rest of the photographs in our dataset besides the training examples. For the two photographs in Fig.1, the predicted scores are (a) 0.781 and (b) 0.157, respectively.

6 Discussions

In previous sections, we have demonstrated the capabilities of our template-based model for classification and assessment despite its simplicity. For further diagnosis of the method and potential improvements, a few important aspects need to be considered.

Template and Feature Capabilities. Our common template can be understood as an "average" template of various art lighting styles. In Fig.7, the results show that the learned common template is not quite suitable for the Paramount style, which however can deal with Rembrandt, Split and Loop styles very well. The reason for this might be that the learned features' contrast types in our template are mostly left-vs-right. In practice, photographers use a key light above and in front of the model to form a butterfly shaped shadow between mouth and nose. This can usually be better captured by top-vs-down features. But in the training process, this effect is often submerged by the other styles. This problem can be partially solved by learning a generative template for each style, in other words, we believe it is necessary to learn a mixture model for artistic portraits, which is more effective and efficient. Besides, if we trade computational efficiency for more complex feature shapes (e.g., triangle, ellipse), we can expect to model the styles like Rembrandt and Paramount better.

Local Inhibition Strategy. We apply local inhibition for reducing the correlations among selected features, which is necessary for the stepwise pursuit algorithm. But the inhibition strategy that allows only one feature in each rectangular part is imperfect, for example, many informative features might be missed during the process. If it can be done with an acceptable computational cost, a pre-computation for the feature correlations on our dataset is helpful. But the scalability of this method is limited by frequent modifications of the dataset and candidate features.

Interpreting the Regression Model. Look at the regression coefficients in Table 1. The coefficient for F_1 is approximately 8.39 with a standard error of 1.29. For this coefficient, the underlying meaning of the numbers is that the larger the local contrast is, the better the photo quality becomes. This does not make much sense as it contradicts with some photographs in our dataset, for example, the demo photo for the Paramount style. To better understand the contribution of each feature, Fig.10 displays part of the estimated coefficients (for the intercept and first two features only) using linear quantile regressions [21] which fit the response $\log(\#_{\text{win}}/\#_{\text{loss}})$, from 5% to 95% quantiles with a step of 5%. The plots reflect the non-linear contributions of the features. For example, F_2 has a more obvious positive contribution thus is more important for low-quantile (daily) photographs than for high-quantile (artistic) ones, although its average effect over various images is small (see the point estimate of its coefficient in Table 1). Similar non-linear effects for the other features are also observed.

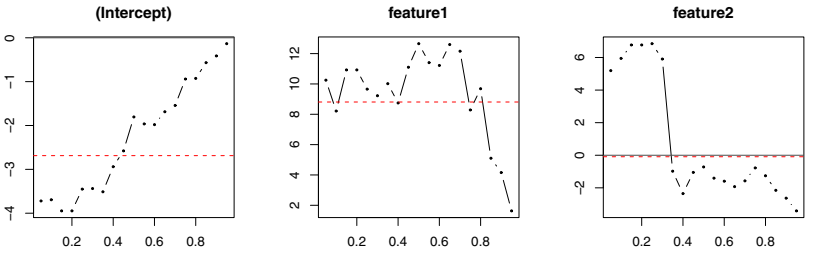


Fig. 10. Coefficient plots of linear quantile regressions, in which the predictors are feature values, and the response is $\log(\#_{\text{win}}/\#_{\text{loss}})$. The horizontal axis is the response quantile, and the vertical axis is the coefficient of each feature. The black curves are quantile regression estimates, and the dashed red lines are for ordinary least-squares regression estimates.

7 Conclusion

In this paper, we learn an artistic lighting template for classification and assessment of artistic and daily portrait photographs in lighting. Using Haar-like features applied on rectangular areas of the face, our method generates a template with justifiable interpretations. We use a log-linear probabilistic model to characterize artistic portrait photographs against daily ones, which gives satisfactory performance in both of the two tasks. Potential improvements of this method include more expressive features, better feature selection strategy with reduced assumptions, and more powerful statistical models, etc.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (90818003&60933006), National High-Tech (863) and Key Technologies R&D Programs of China (2009AA01Z331&2008BAH29B02), National Grand Fundamental Research (973) Program of China (2006CB303007), and Specialized Research Fund for the Doctoral Program of Higher Education (20091102110019).

References

1. Wikipedia, http://en.wikipedia.org/wiki/photography
2. Hurter, B.: The best of photographic lighting — techniques and images for digital photographers, 2nd edn. Amherst Media (2007)
3. Tong, H., Li, M., Zhang, H., He, J., Zhang, C.: Classification of digital photos taken by photographers or home users. PCM (1), 198–205 (2004)
4. Ke, Y., Tang, X., Jing, F.: The design of high-level features for photo quality assessment. In: CVPR, pp. 419–426 (2006)

5. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Studying aesthetics in photographic images using a computational approach. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, Part 3, vol. 3953, pp. 288–301. Springer, Heidelberg (2006)
6. Luo, Y., Tang, X.: Photo and video quality evaluation: Focusing on the subject. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 386–399. Springer, Heidelberg (2008)
7. Wong, L.K., Low, K.L.: Saliency-enhanced image aesthetics class prediction. In: ICIP (2009)
8. Li, C., Chen, T.: Aesthetic visual quality assessment of paintings. IEEE Journal of Selected Topics in Signal Processing 3, 236–252 (2009)
9. Hunter, F., Biver, S., Fuqua, P.: Light: Science and Magic: An Introduction to Photographic Lighting, 3rd edn. Focal Press (2007)
10. Grey, C.: Master Lighting Guide for Portrait Photographers. Amherst Media (2004)
11. Prakel, D.: Basics Photography: Lighting. AVA Publishing (2007)
12. Canny, J.F.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8, 679–714 (1986)
13. Lin, J.: Divergence measures based on the shannon entropy. IEEE Trans. Info. Theory 37 (1991)
14. Della Pietra, S., Della Pietra, V., Lafferty, J.: Inducing features of random fields. IEEE Trans. Pattern Anal. Mach. Intell. 19, 380–393 (1997)
15. Si, Z., Gong, H., Wu, Y.N., Zhu, S.C.: Learning mixed templates for object recognition. In: CVPR, pp. 272–279 (2009)
16. Friedman, J.H.: Exploratory projection pursuit. Journal of American Stat. Assoc. 82, 249–266 (1987)
17. Schwarz, G.: Estimating the dimension of a model. Ann. Statist. 6 (1978)
18. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
19. Faraway, J.J.: Extending the Linear Model with R. Taylor & Francis Group, Abington (2006)
20. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, Part 2, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
21. Koenker, R.: Quantile Regression. Cambridge University Press, Cambridge (2005)

Photometric Stereo from Maximum Feasible Lambertian Reflections

Chanki Yu, Yongduek Seo, and Sang Wook Lee

Department of Media Technology, Sogang University, Seoul, Korea
{ckyu,yndk,slee}@sogang.ac.kr

Abstract. We present a Lambertian photometric stereo algorithm robust to specularities and shadows and it is based on a maximum feasible subsystem (Max FS) framework. A Big-M method is developed to obtain the maximum subset of images that satisfy the Lambertian constraint among the whole set of captured photometric stereo images which include non-Lambertian reflections such as specularities and shadows. Our algorithm employs purely algebraic pixel-wise optimization without relying on probabilistic/physical reasoning or initialization, and it guarantees the global optimality. It can be applied to the image sets with the number of images ranging from four to hundreds, and we show that the computation time is reasonably short for a medium number of images (10~100). Experiments are carried out with various objects to demonstrate the effectiveness of the algorithm.

1 Introduction

Photometric stereo has been a subject of extensive research in computer vision since it was first introduced by Woodham [1]. Given a set of at least three images with different illumination directions, the estimation of surface orientation and albedo is a simple linear problem for Lambertian surfaces. In reality, however, complex object geometry and reflectance result in non-Lambertian behavior of reflections such as specularities and shadows. Although much research has been carried out to develop methods robust to non-Lambertian reflections using more than three images, their presence still poses serious problems to photometric stereo.

Probably the most comprehensive approach would be to account for all the reflections including Lambertian using an appropriate BRDF (bidirectional reflectance distribution function) model and locally estimate albedo and non-Lambertian reflectance parameters as well as surface orientations [10][9][12][5]. This approach can extract the richest information from the image set when the BRDF model can physically represent the image irradiance accurately. On the other hand, any limitations in the BRDF model can degrade the results and the local BRDF-based estimation cannot take cast shadows into account. Some methods use the dichromatic model for separating or discounting specular reflections [7][3], but they require that dielectric object surfaces have reasonably strong chromaticities. A reference object has also been used for obtaining BRDF and surface orientations, but shadows are not considered in the recovery [8].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 115–126, 2010.

The earliest photometric stereo approaches use photometric reasoning with 4 images to detect non-Lambertian reflections and recover albedo and shape by rejecting the images that are suspected to have non-Lambertian reflections. Coleman and Jain [2] used a set of 4 images, computed 4 albedos using combinations of 3 images and detected the presence of specular reflections if the computed albedos are substantially inconsistent. The combination that results in the lowest albedo value is taken as the set of Lambertian image irradiance. Barsky and Petrou extended this approach to shadows but with the help of color information [4]. For Lambertian photometric stereo with 4 light sources, Chandraker *et al*. employed a graph cut-based MRF (Markov Random Field) algorithm to detect light source visibility and recovered shape discarding shadows [6].

Four images hardly suffice to fully sort out non-Lambertian reflections for an object with complex shape and BRDF. Some robust photometric stereo algorithms have recently been developed that use a large number (>100) of dense images. Wu *et al*. employed two MRF inference algorithms (graph cut and tensor belief propagation) for dense photometric stereo [13], but the smoothness constraints on surface orientations diminish fine details and albedos are not recovered. Wu and Tang used EM (Expectation Maximization) to compute albedo while simultaneously clustering a set of initial normals obtained through ratio images [14]. The convergence of EM algorithms is well known, but their performance is influenced by initial conditions. They used the highest 50% intensities as numerators of the ratio images to perform plane fitting. Verbiest and Van Gool adopted an ML (Maximum Likelihood) framework and modeled their inlier map as an MRF with an associated Gibbs-prior distribution [15]. They also took the 50% highest intensities as inliers to provide the prior distribution for their EM optimization.

In this paper, we present a general extension of photometric stereo with from small to large number of images based on a maximum feasible subsystem (Max FS) framework. The Max FS problem is to find the maximum cardinality feasible subset in an infeasible set of linear constraints such as the Lambertian constraints [19]. By solving this problem, we can obtain the maximum subset of images that satisfy the Lambertian constraint among the whole set of images which include specular reflections and shadows. Until recently there has been little development of algorithms for actually solving the MAX FS problem, but a small number of methods are now becoming available, with more under development, inspired by several applications in fields such as radiation therapy planning, machine learning, signal processing, etc.

We use the Big-M method, a mixed-integer exact formulation, for solving the Max FS problem, and show that it is appropriate for the photometric stereo problem [20]. Unlike the dense photometric stereo methods mentioned above, the presented method is based on purely algebraic pixel-wise optimization without relying on the probabilistic/physical reasoning or initialization and it guarantees the global optimality. It requires only one parameter that can be determined once for a system depending on the image noise/distortion level. The presented algorithm does not require the redundancy that a dense image set can provide. Therefore, the number of images the presented algorithm can be as small as 4 and as large as hundreds. For a medium number of images (10~100), the computation is reasonably fast.

In multiview geometry, the SOI (sum of infeasibilities) has mainly been used for global minimization [18]. Recently, Li has developed a global optimization method for the algebraic DLT(Direct Linear Transformation) problem that has fixed bounded variables [17]. He suggeted an exact bilinear-MIP(Mixed Integer Programming)

formulation and obtained globally optimal solutions using an LP(Linear Programming)-based BnB(Branch-and-Bound) algorithm. Although the Li's work is a somewhat different approach to a geometric problem than our MaxFS/Big-M approach to a photometric problem, the objectives are the same.

It may be noted that RANSAC [16] has been used for photometric stereo and appearance-based approach [26][27]. The main difference between the RANSAC and the Max FS lies in the global optimality. The RANSAC finds some of the inliers, but not all. As RANSAC is a non-deterministic heuristic algorithm, it provides no guarantee to the optimality of its solution in terms of maximizing the feasible set's cardinality. Li performed some RANSAC experiments and observed significant variation of results that cannot be eliminated by further nonlinear refinement [17].

The rest of this paper is organized as follows. Section 2 presents a Max FS formulation of Lambertian photometric stereo, and experimental results are presented in Section 3, and we conclude in Section 4.

2 Max FS Formulation for Photometric Stereo

2.1 Lambertian Photometric Stereo

In photometric stereo, surface normal vector and albedo are estimated from images captured under multiple illumination directions. We consider only calibrated photometric stereo where light directions and intensities are known. Images are captured at a fixed viewpoint under orthographic projection.

For the Lambertian reflection model with k illumination directions, the image irradiance $\mathbf{I} = [\, I_1\, I_2 \ldots I_k\,]^{\mathrm{T}} \in \Re^k$) at a pixel is given by:

$$\boldsymbol{I} = \rho(\boldsymbol{L}\cdot\boldsymbol{N}) = \boldsymbol{L}\cdot\boldsymbol{n}\,, \tag{1}$$

where ρ is the albedo, $\boldsymbol{N}$ is the surface normal ($[N_x\ N_y\ N_z]^{\mathrm{T}} \in \Re^3$), $\boldsymbol{L}$ is the illumination unit vectors ($[\boldsymbol{L}_1\ \boldsymbol{L}_2 \ldots \boldsymbol{L}_k\,] \in \Re^{k\times 3}$). Attached shadows are not considered in Equation 1.

We estimate the scaled normal vector $\boldsymbol{n} \in \Re^3$ which is the normal vector $\boldsymbol{N}$ multiplied by the scalar value ρ. When a grey-scale image is captured in m bits, the scalar value ρ ranges from 0 to 2^m-1 and the scaled surface normal $\boldsymbol{n}$ has a range of value of $\boldsymbol{n}^{lb} \sim \boldsymbol{n}^{ub}$ ($\boldsymbol{n}^{ub} = 2^m - 1$, $\boldsymbol{n}^{lb} = -\boldsymbol{n}^{ub}$). When the $\boldsymbol{n}$ is estimated, $\boldsymbol{N}$ and ρ can be easily obtained as follows:

$$\boldsymbol{N} = \frac{\boldsymbol{n}}{|\boldsymbol{n}|}\,, \qquad \rho = |\boldsymbol{n}|\,. \tag{2}$$

In the Max FS formulation described below, Lambertian pixels are taken as inliers and non-Lambertian specular/shadowed pixels are as outliers. If we could select Lambertian pixels from input data, we could estimate the accurate scaled normal vector.

2.2 Maximum Feasible Subsystem Problem

The goal of a Max-FS algorithm is to find the largest cardinality set that constraints are feasible. In other words, it seeks a feasible subsystem containing the largest

number of inequalities for an infeasible linear system $\boldsymbol{Ax} \leq \boldsymbol{b}$ with real matrix $\boldsymbol{A} \in \mathfrak{R}^{k \times n}$, real vector $\boldsymbol{b} \in \mathfrak{R}^{k}$ and variable $\boldsymbol{x} \in \mathfrak{R}^{n}$. The Max-FS problem admits the following mixed integer linear programming (MILP) formulation by introducing a binary variable y_i for each of the inequalities:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & \sum_{i=1}^{k} y_i \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq b_i + M_i \, y_i, \quad \forall i \\ & \boldsymbol{x} \in \Re^n, \quad y_i \in \{0,1\}, \quad i = 1,...,k \ , \end{aligned} \tag{4}$$

where M_i is a large positive number that converts an infeasible inequality into a feasible one when $y_i = 1$. The case where $y_i = 0$ indicates that the inequality is feasible. Note that if $y_i = 1$, then the i^{th} constraint is automatically deactivated. This MILP formulation is known as the Big-M method [19][20].

The performance of the Big-M method varies depending on the constant M_i. With tighter M_i, the method produces better results. If there are fixed bounds on the variable $\boldsymbol{x}$ ($\boldsymbol{x}^{lb} \leq \boldsymbol{x} \leq \boldsymbol{x}^{ub}$), we can compute a proper M_i using the following equation and obtain a tight formulation:

$$M_i = \boldsymbol{max} \left\{ \sum_{j=1}^{n} a_{ij} x_j - b_i \right\}, \quad x_j^{lb} \leq x_j \leq x_j^{ub} \ , \ j = 1,...,n \, . \tag{5}$$

Generally, MILP problem are solved by the LP-based BnB(Linear Programming-based Branch and Bound) or the LP-based BnC (LP-based Branch and Cut). LP-base BnB/BnC guarantees the global optimality of its solution [17][21].

2.3 Max FS (Big-M MILP) Formulation for Photometric Stereo

By taking the pixels from Lambertian reflections as inliers and taking those from non-Lambertian reflections as outliers, we can formulate the photometric stereo problem with non-Lambertian reflections as a Max-FS problem. The set of input data ($\boldsymbol{D}$) is partitioned into the inlier-set $\boldsymbol{D_I}$ (Lambertian pixels) and the outlier-set $\boldsymbol{D_O}$ (specular and shadowed pixels) with $\boldsymbol{D_I} \subseteq \boldsymbol{D}, \boldsymbol{D_O} \subseteq \boldsymbol{D}, \boldsymbol{D_I} \cup \boldsymbol{D_O} = \boldsymbol{D}$ and $\boldsymbol{D_I} \cap \boldsymbol{D_O} = \boldsymbol{\emptyset}$.

A maximum noise magnitude (or tolerance) $\varepsilon > 0$ provides a bound for the equation $\boldsymbol{L}_i \cdot \boldsymbol{n}$ and image intensity I_i:

$$\left| \boldsymbol{L}_i \cdot \boldsymbol{n} - I_i \right| \leq \varepsilon \, . \tag{6}$$

The Max-FS formulation of Equation 6 using the big-M method is as follows:

$$\begin{aligned} \min_{\mathbf{n},\mathbf{y}} \quad & \sum_{i=1}^{k} y_i \\ \text{subject to} \quad & \left| \boldsymbol{L}_i \cdot \boldsymbol{n} - I_i \right| \leq \varepsilon + M_i \, y_i, \ \forall i \\ & y_i \in \{0,1\}, \quad i = 1,...,k \\ & n^{lb} \leq n_j \leq n^{ub} \ , \ j = 1,2,3 \, . \end{aligned} \tag{7}$$

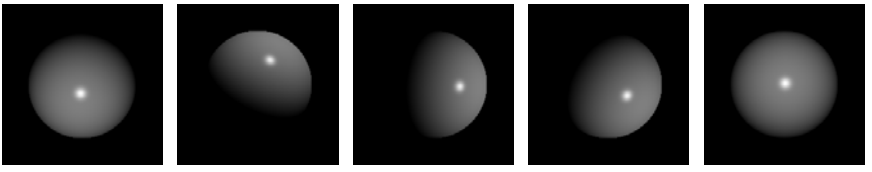


Fig. 1. Five synthetic images of a sphere

If $y_i = 1$, the i^{th} data is specular or shadowed pixel. If $y_i = 0$, on the other hand, the i^{th} constraint satisfies the Lambertian condition and the pixel is an inlier.

We can calculate the proper M_i from Equation 5. For an m-bit grey-scale image, however, the captured image intensities are limited to $n^{ub} = 2^m - 1$ and so is $|\boldsymbol{L}_i \cdot \boldsymbol{n} - I_i|$.

Therefore, we can determine the tight fixed bound considering the tolerance value (ε) as follows:

$$M_i = n^{ub} + \varepsilon, \quad \forall i . \tag{8}$$

The constraint equation ($|\boldsymbol{L}_i \cdot \boldsymbol{n} - I_i| \le \varepsilon + M_i y_i$) of Equation 7 can now be replaced by the following two inequalities:

$$\boldsymbol{L}_i \cdot \boldsymbol{n} - I_i \le \varepsilon + M_i y_i, \quad -\boldsymbol{L}_i \cdot \boldsymbol{n} + I_i \le \varepsilon + M_i y_i . \tag{9}$$

If attached shadows are considered, the Lambertian reflection becomes as follows:

$$\boldsymbol{I} = \boldsymbol{max}(\boldsymbol{L} \cdot \boldsymbol{n}, 0). \tag{10}$$

Intensities should always be non-negative. However, Equation 9 allows that a $\boldsymbol{L}_i \cdot \boldsymbol{n}$ have the small negative value ($-\varepsilon < \boldsymbol{L}_i \cdot \boldsymbol{n} < 0$) near the boundaries of attached shadows.

To solve the this problem, we add the non-negative constraints with respect to the tolerance value ε as follows:

$$\boldsymbol{L}_i \cdot \boldsymbol{n} \ge 2\varepsilon - M_i y_i . \tag{11}$$

This constraint prevents shadowed pixels from being chosen as a maximum feasible set. With the constraints shown in Equations 9 and 11, the photometric stereo with the maximum feasible Lambertian reflections is given as follows:

$$\begin{aligned} \min_{\mathbf{n},\mathbf{y}} \quad & \sum_{i=1}^{k} y_i \\ \text{subject to} \quad & \boldsymbol{L}_i \cdot \boldsymbol{n} - M_i y_i \le I_i + \varepsilon, \quad \forall i \\ & -\boldsymbol{L}_i \cdot \boldsymbol{n} - M_i y_i \le -I_i + \varepsilon, \quad \forall i \\ & -\boldsymbol{L}_i \cdot \boldsymbol{n} - M_i y_i \le -2\varepsilon, \quad \forall i \\ & n^{lb} \le n_j \le n^{ub}, \qquad j = 1, 2, 3, \\ & y_i \in \{0, 1\}, \qquad i = 1, \ldots, k . \end{aligned} \tag{12}$$

Note that our method does not require initial values of parameters except the tolerance value ε. Since it is a mixed-integer programming problem, we use a BnC algorithm in the GLPK (GNU Linear Programming Kit) as a solver. The GLPK package is written

in ANSI C and developed for solving large-scale linear programming (LP) and mixed integer linear programming (MILP) problems. An LP-based BnC algorithm consists of a cutting plane method and an LP-based BnB algorithm. It provides constraints by plane cutting to the BnB algorithm for faster convergence.

3 Experimental Results

3.1 Experiment with a Synthetic Sphere

For an experiment with known ground truth, we used images generated from a synthetic sphere. For simulation, a variant of the Torrance-sparrow model is adopted with the surface roughness 0.07. We sampled the illumination space for zenith angles up to 60° with approximately uniform projected solid angle and generated 196 images. We carried out experiments for several subsets of 196 images. The captured RGB images have a 24-bit color depth and we used simple grey-scale conversion I = (R+G+B)/3. The upper and lower bounds of the scaled normal vector elements are assigned to 255 and -255.

Table 1 shows the running times per pixel for 12, 24, 36, 48 and 60 random samples of 196 images. Table 2 shows the running times and the average angular errors of computed surface normals in degrees for several tolerance values. The error between the estimation and the ground truth of surface normal is computed as follows:

$$e = \frac{1}{N_p} \sum_{i=1}^{N_p} \left| \cos^{-1}\left(\boldsymbol{n}_i^g \cdot \boldsymbol{n}_i^e\right) \right|,$$

where N_p is the total number of pixels, $\boldsymbol{n}_i^g$ and $\boldsymbol{n}_i^e$ are the ground-truth and the estimated surface normal vectors at pixel i, respectively.

Table 1. Summary of running times (tolerance ε=5)

Number of images	12	24	36	48	60
Average running time per pixel [msec]	7.834	35.474	225.568	985.625	5,624.73

Table 2. Running times and angular errors for experiment with 12 and 24 images

- Average running time per pixel. [unit: msec]

	ε=1	ε=2	ε=3	ε=4	ε=5
12 images	10.073	9.319	8.762	8.305	7.834
24 images	45.657	40.577	37.863	36.051	35.232

- Average angular errors of recovered surface normals [unit: degree]

	ε=1	ε=2	ε=3	ε=4	ε=5
12 images	0.138^o	0.158^o	0.178^o	0.199^o	0.219^o
24 images	0.085^o	0.102^o	0.118^o	0.137^o	0.153^o

The experiments were run on an Intel Core™2 CPU 6300, 1.87GHz with 4GB RAM and our algorithm is implemented with C language using the GLPK library which provides functions for mixed-integer programming [21]. The scaled normal vector is estimated using a least-squares method with the inlier image set estimated by the Max-FS algorithm. For these medium-sized dataset, the computation is reasonably fast and the normal errors are very small.

3.2 Experiments with Real Objects

We test the performance of our algorithm using four data sets captured by equipment. Illumination sources are made of the white LED lights and they are calibrated. The raw image data obtained with Nikon D-200 camera are used, and we convert the raw images to linear 24-bit RGB images (8 bits per channel) for processing.

Figure 2 show the results from baseball images: (a-c) show 3 images among 19 images used by our algorithm, (e-g) show the 3 binary feasibility maps obtain from our

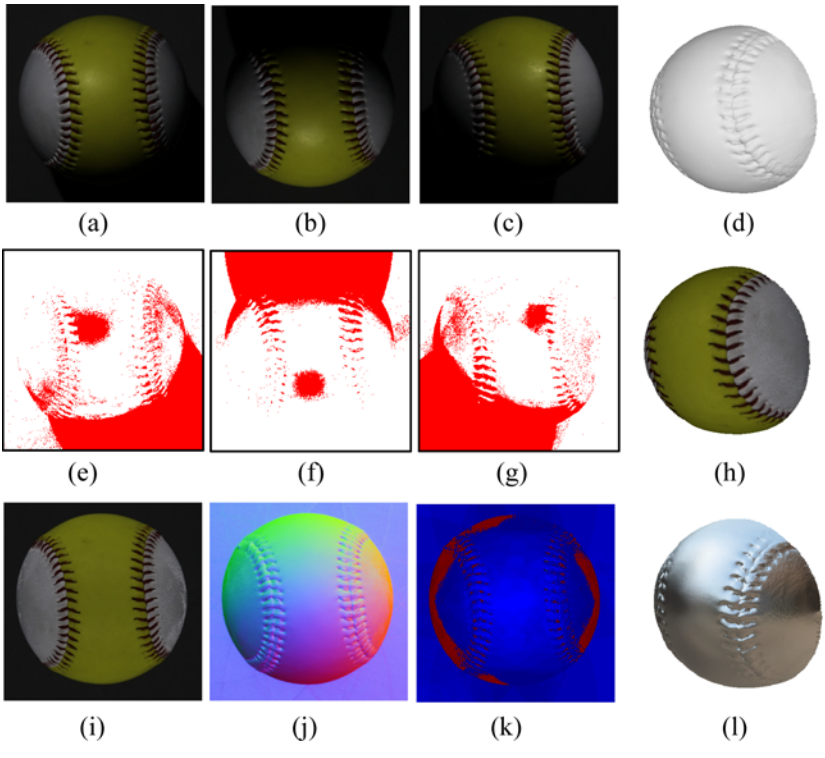


Fig. 2. (a~c) three images of a baseball, (e~g) binary feasibility maps for the images shown in (a~c), respectively, (i) estimated albedo, (j) color-encoded normal map, (k) inlier map, (d) rendered view of recovered 3D shape, (h) rendered view of 3D shape textured only with Lambertian albedo, and (l) environmental lighting effects shown on metallic surface.

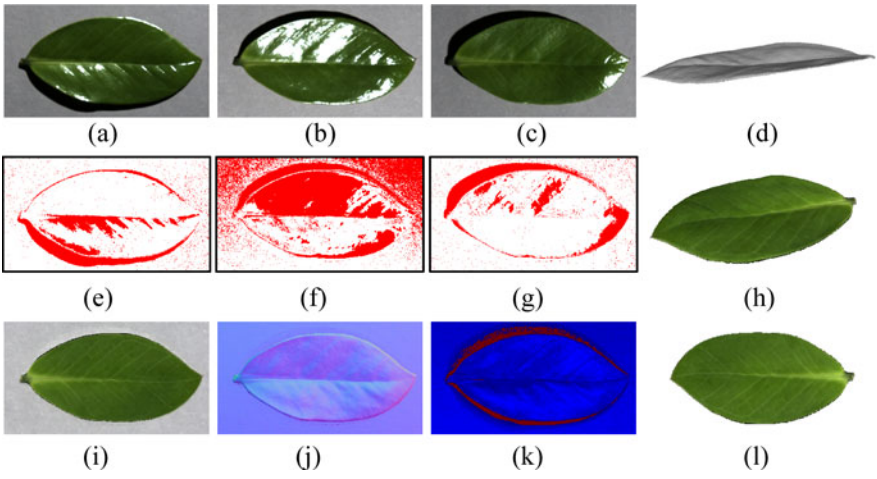


Fig. 3. (a~c) three images of a leaf, (e~g) binary feasibility maps for the images shown in (a~c), respectively, (i) estimated albedo, (j) color-encoded normal map, (k) inlier map, (d) rendered view of recovered 3D shape, and (h,l) rendered view of 3D shape textured only with Lambertian albedo.

Max-FS algorithm for the 3 images shown in (a~c), respectively, where inliers in each image are denoted in white and outliers in red, (i) shows estimated albedo, and (j) shows color-coded surface normals. This color-coded map of surface normals is obtained by the RGB encoding: [R, G, B] = 255 [$(n_x+1)/2$, $(n_y+1)/2$, n_z]. Figure 2 (k) shows the inlier map computed by the Max-FS algorithm. The regions with more than 50% of inliers are shown in blue, and those with less than 50% are in red, and those with less than 3 inliers are in green. (Only a few pixels are on green, but not clearly visible.) The amount of inliers is denoted by the brightness in each region. Figure 2 (d) shows the rendered view of the reconstructed surface, (h) shows the rendered view texture-mapped only with the Lambertian albedo shown in (i), and (l) shows the environmental lighting effects shown on metallic surface. The images (d), (h) and (l) show that the details of the complex concave regions of the baseball seams are recovered properly. We used open source code to recover the shape [22][23].

Figure 3 shows the results from a glossy leaf which has some anisotropic non-Lambertian reflectances. Figure 4 shows the results from a picture frame with complex shape and various materials, and the results from a highly glossy teapot is shown in Figure 5.

For the experiments with the baseball (Figure 2) and the teapot (Figure 5), we used 19 images with viewing direction $(\theta, \phi) = (0, 0)$, and the illumination angles were given as follows. For $\theta = 30°$ and 60°, ϕ ranges from 45° to 315° at an interval of 90°. For $\theta = 45°$, ϕ ranges from 30° to 330° at an interval of 30°. For the experiments with the leaf (Figure 3) and the picture frame (Figure 4), we used 15 images with viewing

direction $(\theta, \phi) = (0, 0)$, and the illumination angles were given as follows. For $\theta = 30°$, ϕ ranges from 45° to 315° at an interval of 90°. For $\theta = 45°$, ϕ ranges from 30° to 330° at an interval of 30°. The running times for the four real datasets are shown in Table 3.

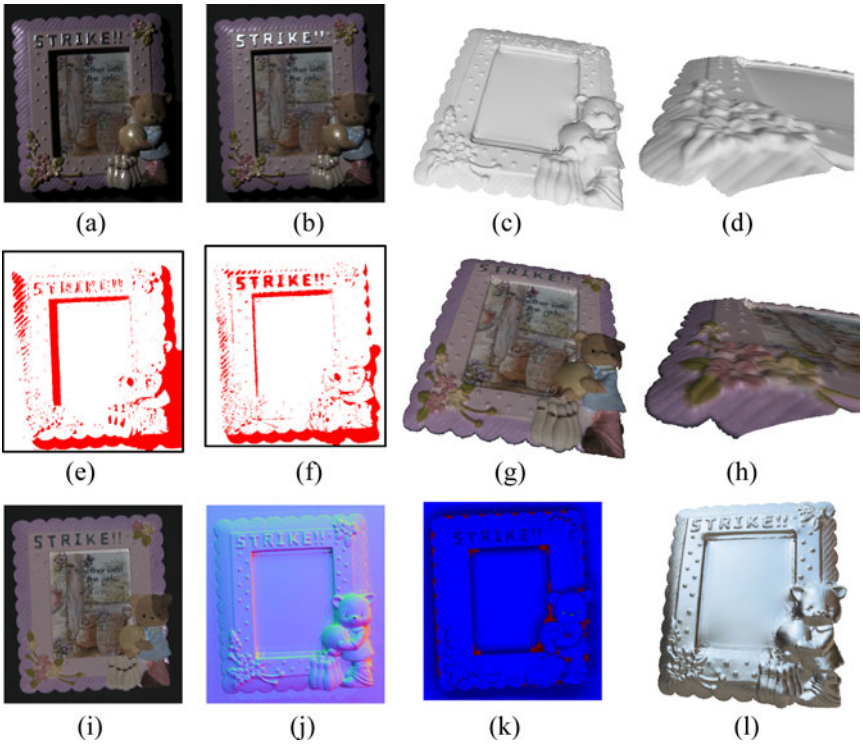


Fig. 4. (a, b) two images of a picture frame, binary feasibility maps for the images shown in (a, b), respectively, (i) estimated albedo, (j) color-encoded normal map, (k) inlier map, (c, d) rendered views of recovered 3D shape, (g, h) rendered views of 3D shape textured only with Lambertian albedo, and (l) environmental lighting effects shown on metallic surface.

Table 3. Summary of experiments with four datasets

Running time per pixel [unit: msec]

Data set	Baseball	Leaf	Picture frame	Teapot
# of images	19	15	15	19
tolerance ε	6	4	11	11
Average running time per pixel. [msec]	48.205	40.673	9.882	29.692

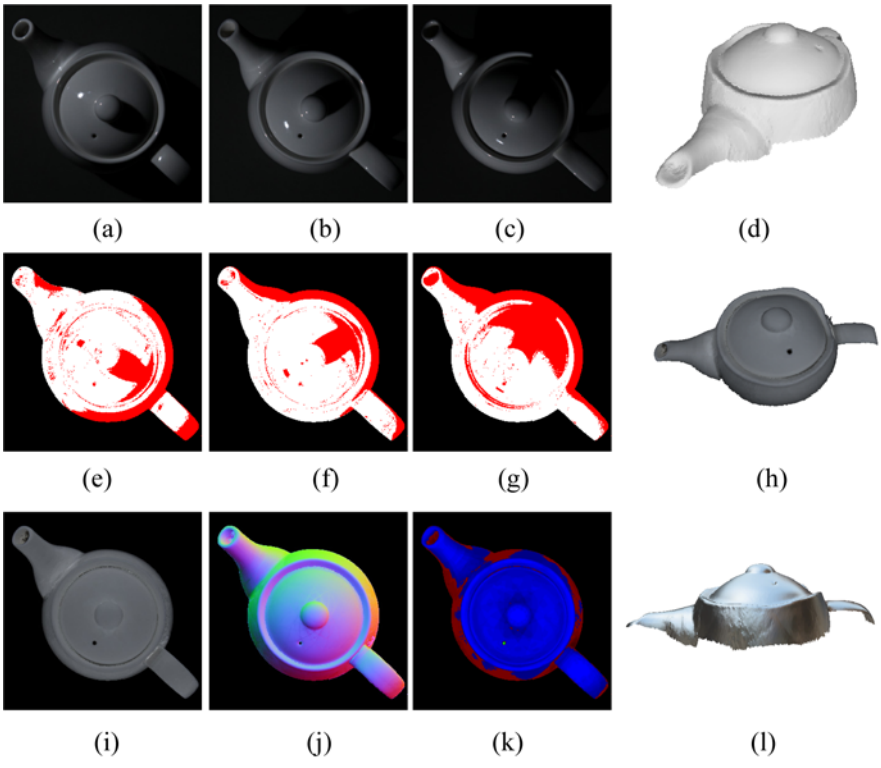


Fig. 5. (a~c) three images of a teapot, (e~g) binary feasibility maps for the images shown in (a~c), respectively, (i) estimated albedo, (j) color-encoded normal map, (k) inlier map, (d) rendered view of recovered 3D shape, (h) rendered view of 3D shape textured only with Lambertian albedo, and (l) environmental lighting effects shown on metallic surface

4 Conclusion

The recent development of Max FS methods and software can be very useful for many computer vision problems, and it is shown in this paper that they are highly suitable for distinguishing non-Lambertian reflections from Lambertian in an image set that has specular reflections are shadows. For the photometric stereo problem, the irradiance and albedo values captured images are strictly bounded and thus the Big-M method is very effective. The presented algorithm is highly efficacious in rejecting non-Lambertian components of reflections due to weak specular reflections and shadows. It requires only one parameter that represents the system noise and distortion and does not rely on probabilistic reasoning or initial conditions. The system noise can be possibly estimated automatically from captured images [25].

We have yet to experimentally compare our algorithm with one of the recent dense photometric stereo methods. The dense photometric stereo with more than a hundred images can benefit from surface normal averaging effect and thus higher signal-to-noise results can be obtained. If this effect is of interest, we may simply capture

several images for each light direction and average them for the similar results. This does not increase the computation time substantially.

Our future work includes the development of methods for very rough surfaces where non-Lambertian reflections from specularities are wide spread spatially. Rougher a surface is, fewer Lambertian reflections are observed in general. For the rough surface where less than three Lambertian reflections are exposed, we are interested in developing methods that utilize spatial constraints from the neighborhood to solve the two-image photometric stereo [24].

References

1. Woodham, R.: Photometric method for determining surface orientation from multiple images. Opt. Eng. 19(1), 139–144 (1980)
2. Coleman Jr., E., Jain, R.: Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. CGIP 18(4), 309–328 (1982)
3. Sato, Y., Ikeuchi, K.: Temporal-color space analysis of reflection. Journal of Optical Society of America A 11(11), 2990–3002 (1992)
4. Barsky, S., Petrou, M.: The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. PAMI 25(10), 1239–1252 (2003)
5. Nayar, S., Ikeuchi, K., Kanade, T.: Determining shape and reflectance of hybrid surfaces by photometric sampling. IEEE Trans. on Robotics and Automation 6(4), 418–431 (1990)
6. Chandraker, M., Agarwal, S., Kriegman, D.: Shadowcuts: Photometric stereo with shadows. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
7. Mallick, S.P., Zickler, T.E., Kriegman, D.J., Belhumeur, P.N.: Beyond Lambert: Reconstructing specular surfaces using color. In: IEEE International Conference on Computer Vision (2005)
8. Hertzmann, A., Seitz, S.: Shape and materials by example: a photometric stereo approach. In: IEEE Conference on Computer Vision and Pattern Recognition (2003)
9. Goldman, D., Curless, B., Hertzmann, A., Seitz, S.: Shape and spatially-varying brdfs from photometric stereo. In: IEEE International Conference on Computer Vision (2005)
10. Tagare, H., de Figueiredo, R.: A theory of photometric stereo for a class of diffuse non-lambertian surfaces. PAMI 13(2), 133–152 (1991)
11. Georghiades, A.S.: Incorporating the Torrance and Sparrow model of reflectance in uncalibrated photometric stereo. In: IEEE International Conference on Computer Vision (2003)
12. Chung, H.-S., Jia, J.: Efficient photometric stereo on glossy surfaces with wide specular lobes. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
13. Wu, T.-P., Tang, K.-L., Tang, C.-K., Wong, T.-T.: Dense photometric stereo: A markov random field approach. PAMI 28(11), 1830–1846 (2006)
14. Wu, T.-P., Tang, C.-K.: Dense photometric stereo by expectation maximization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 159–172. Springer, Heidelberg (2006)
15. Verbiest, F., Van Gool, L.: Photometric stereo with coherent outlier handling and confidence estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)

17. Li, H.: Consensus Set Maximization with guaranteed global optimality for robust geometry estimation. In: IEEE International Conference on Computer Vision (2009)
18. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. In: IEEE International Conference on Computer Vision (2005)
19. Chinneck, J.W.: Feasibility and infeasibility in optimization: algorithms and computational methods, 1st edn. Springer, Heidelberg (2007)
20. Parker, M.: A set covering approach to infeasibility analysis of linear programming problems and related issues. PhD thesis, University of Colorado at Denver (1995)
21. Makhorin, A.: GLPK (GNU linear programming kit) 4.1.6 (2004), http://www.gnu.org/software/glpk/glpk.html
22. Agrawal, A., Raskar, R., Chellappa, R.: What is the range of surface reconstructions from a gradient field? In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 578–591. Springer, Heidelberg (2006)
23. Agrawal, A., Raskar, R., Chellappa, R.: An algebraic approach to surface reconstructions from gradient fields? In: IEEE International Conference on Computer Vision (2006)
24. 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)
25. Liu, C., Szeliski, R., Kang, S.B., Zitnick, C.L., Freeman, W.T.: Automatic estimation and removal of noise from a single image. PAMI 30(2), 299–314 (2008)
26. Mukaigawa, Y., Ishii, Y., Shakunaga, T.: Analysis of photo-metric factors based on photometric linearization. Journal of Optical Society of America A 24(10), 3326–3334 (2007)
27. Higo, T., Matsushita, Y., Joshi, N., Ikeuchi, K.: A hand-held photometric stereo camera for 3-D modeling. In: IEEE International Conference on Computer Vision (2009)

Part-Based Feature Synthesis for Human Detection

Aharon Bar-Hillel[1,⋆], Dan Levi[1,⋆], Eyal Krupka[2], and Chen Goldberg[3]

[1] General Motors Advanced Technical Center Israel, Herzliya
{aharon.barhillel,dan.levi}@gm.com
[2] Israel Innovation Labs, Microsoft Israel R&D Center, Haifa
eyalk@microsoft.com
[3] Tel Aviv University
chen.goldberg@cs.tau.ac.il

Abstract. We introduce a new approach for learning part-based object detection through feature synthesis. Our method consists of an iterative process of feature generation and pruning. A feature generation procedure is presented in which basic part-based features are developed into a feature hierarchy using operators for part localization, part refining and part combination. Feature pruning is done using a new feature selection algorithm for linear SVM, termed Predictive Feature Selection (PFS), which is governed by weight prediction. The algorithm makes it possible to choose from $O(10^6)$ features in an efficient but accurate manner. We analyze the validity and behavior of PFS and empirically demonstrate its speed and accuracy advantages over relevant competitors. We present an empirical evaluation of our method on three human detection datasets including the current de-facto benchmarks (the INRIA and Caltech pedestrian datasets) and a new challenging dataset of children images in difficult poses. The evaluation suggests that our approach is on a par with the best current methods and advances the state-of-the-art on the Caltech pedestrian training dataset.

Keywords: Human detection, Feature selection, Part-Based Object Recognition.

1 Introduction

Human detection is an important instance of the object detection problem, and has attracted a great deal of research effort in the last few years [1,2,3,4,5,6]. It has important applications in several domains including automotive safety and smart video surveillance systems. From a purely scientific point of view, it incorporates most of the difficulties characterizing object detection in general-namely viewpoint, scale and articulation problems. Several approaches have been put forward, with established benchmarks [1,7] enabling competitive research.

It is widely acknowledged that a method's detection performance largely depends on the richness and quality of the features used, and the ability to combine

⋆ Both authors contributed equally to this work.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 127–142, 2010.

diverse feature families [3,8]. While some progress has been made by careful manual feature design [9,1,10], there is a growing tendency to automate the feature design and cue integration process. This can be done by feature selection from a very large feature family [2,11], or by kernel integration methods [8]. The idea of feature selection has been extended to 'feature mining' by introducing the notion of a dynamic hypothesis family to select from, see [3]. In this paper we take this notion one step further.

At an abstract level, we regard automatic feature synthesis as an iterative interplay of two modules: a 'hypothesis generator' and a 'hypothesis selector'. The 'hypotheses generator' gets a temporary classifier and a hypothesis family, and produces an extended hypothesis family, with new features which are conjectured to be helpful to the current classifier. The 'hypotheses selector' then prunes the new suggested feature set and learns a (hopefully better) classifier from it. The distinction between these two agents and the division of labor between them follows similar frameworks in the methodology of scientific discovery ('context of discovery' and 'context of justification' presented in [12]) or in certain forms of reinforcement learning (the actor-critic framework [13]).

This paper makes two main contributions, corresponding to the domains of 'feature generation' and 'feature selection' mentioned above. First, we suggest a part-based feature generation process, where parts are derived from natural images fragments such as those used in [14,15]. This process starts with basic global features and gradually moves toward more complicated features including localization information, part description refinement, and spatial/logical relations between part detections. More complex part-based features are generated sequentially, from parts proved to be successful in earlier stages. Second, we introduce a new feature selection method for Support Vector Machines (SVMs), termed the SVM Predictive Feature Selection (SVM-PFS), and use it in the pruning stages of the feature synthesis process. SVM-PFS iterates between SVM training and feature selection and provides accurate selection with orders of magnitude speedup over previous SVM wrappers. We provide a formal analysis of SVM-PFS and empirically demonstrate its advantages in a human detection task over alternatives such as SVM-RFE [16], boosting [17] or column generation [18].

We test our feature synthesis process on three human detection datasets, including the two current de-facto benchmarks for pedestrian detection (the INRIA [1] and Caltech [7] datasets) and a difficult dataset of children involved in various activities which we have collected ourselves. Our method is comparable to the best current methods on the INRIA and Children datasets. On the Caltech pedestrian training dataset we achieve a detection rate of 30% at 1 false alarm per image compared to at most 25% for competing methods.

1.1 Overview and Related Work

The learning process we term *Feature Synthesis* gets positive and negative image window examples as input and learns an image window classifier. *Feature Synthesis* is an iterative procedure in which iteration n is composed of two stages: *feature generation* resulting in a set of candidate features F_n, and *feature selection*

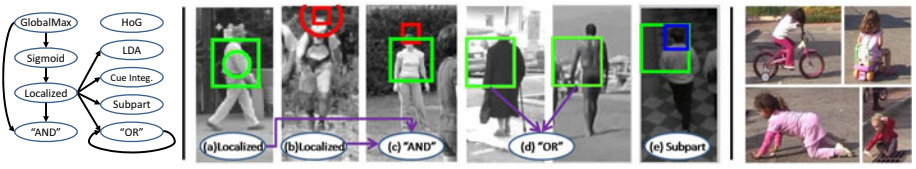


Fig. 1. Left: Feature types currently supported in our generation process. An arrow between A and B stands for 'A can be generated from B'. **Center:** Examples of features from our learned classifiers. a,b) Localized features. The rectangle denotes the fragment. The circle marks the 1-std of its location Gaussian. c) Detection example for a spatial "AND" feature composed of fragments a,b. d) Two fragments composing together a semantic "OR" feature. e) A subpart feature. The blue rectangle is the emphasized fragment quarter. **Right:** Typical images from the Children dataset.

resulting in a subset of selected features $S_n \subset F_n$ and a learned linear classifier C_n. In the *feature generation* stage a new set of features T_n is generated, and added to previously generated features. We experimented with two ways to construct the candidate set: the 'monotonic' way $F_n = F_{n-1} \cup T_n$ and the 'non-monotonic' version, in which we continue from the set of previously selected features: $F_n = S_{n-1} \cup T_n$. In the *feature selection* stage the PFS algorithm selects a subset of the candidate features $S_n \subset F_n$ with a fixed size M and returns the learned classifier C_n. From the second iteration on PFS is initialized with the previous selected features and their weights (S_{n-1}, C_{n-1}) directing its search for new useful features. The final classifier consists of the selected features S_n and learned classifier C_n at the final iteration.

While the framework described above can be applied to any feature type, we suggest a process in which the introduced feature sets T_n consist of part-based features with increasing complexity. Part-based representations have attracted a lot of machine vision research [14,19,4,20,21], and are believed to play an important role in human vision [22]. Such representations hold the promise of being relatively amendable to partial occlusion and object articulation, and are among the most successful methods applied to current benchmarks [4]. The parts used in our process are defined using natural image fragments, whose SIFT [9] descriptors are compared to the image in a dense grid [14,20]. Beginning with simple features corresponding to the global maximal response of a part in the image, we derive more complex features using several operators which can be roughly characterized as based on localization, refinement and combination.

Part localization adds a score to the feature reflecting the location of the part in an absolute framework (commonly referred to as a 'star model' [21]), or with respect to other parts (e.g. in [23]). Part refinement may take several forms: part decomposition into subparts [24], re-training of the part mask for increased discriminative power [4], or binding the SIFT descriptor of the part with additional, hopefully orthogonal, descriptors (e.g. of texture or color). Part combination may take the form of 'and' or 'or' operators applied to component parts, with and without spatial constraints. Applying 'and' operators corresponds to

simple monomials introducing non-linearity when no spatial constraints are imposed, and to 'doublets' [23] if such constraints exist. Applying 'or' operators can create 'semantic parts' [20] which may have multiple, different appearances yet a single semantic role, such as 'hand' or 'head'. While most of these feature forms have previously been suggested, here we combine them into a feature generation process enabling successive creation of feature sets with increasing complexity.

The feature synthesis approach proposed requires successive large-scale feature selection and classifier learning epochs. We chose SVM as our base classifier, based on theoretical considerations [25] as well as empirical studies [26]. Feature selection algorithms can be roughly divided into filters, governed by classifier-independent feature ranking, and wrappers, which are selecting features for a specific classifier and include repeated runs of the learner during selection. Typically the former are faster, while the latter are more accurate in terms of classification performance. While several wrapper methods for SVM have been described in the literature [27,16,28], all of them require at least one, and usually several SVM runs on the entire data with the full candidate feature set. For large datasets with thousands of examples and features this is prohibitive, as even a single computation of the Gram matrix takes $O(L^2N)$ where L is the number of examples and N is the number of features.

The algorithm we suggest for the *feature selection* stage, SVM-PFS, aims to match the accuracy of existing SVM wrapper methods, and specifically the SVM-RFE [16] method, but with a low computational cost like filter methods. SVM-RFE starts by training SVM with the full candidate feature set, then it removes the features with the lowest absolute weight in a backward elimination process. SVM-PFS uses a similar elimination process, but it avoids training SVM on the entire feature set. Instead SVM is only trained on small subsets of features, and the learned classifier is used to obtain weight predictions for unseen features. Our SVM-PFS analysis derives the predicted weight criterion from gradient considerations, bounds the weight prediction error, and characterizes the algorithm's behavior in the presence of large amounts of useless features. SVM-PFS has a speedup factor of order $Q/\log(Q)$ over SVM-RFE, where Q is the ratio between the sizes of the candidate and final feature sets. In our experiments SVM-PFS accuracy was comparable to SVM-RFE, while speedup factors of up to $\times 116$ were obtained. This speedup enables our large scale feature synthesis experiments.

There are several lines of work in the literature which are close to our approach in at least one respect. The Deformable Part Model (DPM) [4] shares the part-based nature of the model, and the search for more complex part-based representations. However, the learning technique they employ (latent SVM) is very different from ours, as well as the models learned. Unlike [4], our model typically includes tens to hundreds of parts in the final classifier, with various feature types extracted from them. The 'feature mining' approach [3] shares the attempt to automate hypothesis family formation, and the basic concepts of a dynamic feature set and generator-selector distinction. However, both the generator they employ (parameter perturbations in a single parametric family) and

the selector (boosting) are fundamentally different. In the feature selection literature, forward selection methods like [29] and specifically the column generation approach [18] are the most similar to ours. The latter uses a weight prediction score identical to ours, but in an agglomerative boosting-like framework. Our work is more inspired by RFE both in terms of theory and in adopting an elimination strategy. Though PFS and column generation use the same feature ranking score in intermediate steps, we show that their empirical behavior is very different, with the PFS algorithm demonstrating considerable advantage. Qualitative observations suggest that the reason is the inability of the agglomerative method to remove features selected early, which become redundant later.

We explain the *feature generation* stage in Section 2. Section 3 presents the PFS algorithm for feature selection, Section 4 provides the empirical evaluation of our method and Section 5 briefly discusses future work.

2 Part Based Feature Generation

As a preliminary stage to the first *feature generation stage* we sample a large pool of rectangular image fragments R from the positive training examples, in which the objects are roughly aligned. The fragments cover a wide range of possible object parts with different sizes and aspect ratios as suggested in [14]. Given an image I and a fragment r we compute a sparse set of its detected locations L^r, where each location $l \in L^r$ is an (x, y) image position. The detected locations are computed as in [20]. We compute a 128-dimensional SIFT descriptor [9] of the fragment, $S(r)$ and compare it to the image on a dense grid of locations using the inner product similarity $a^r(l) = S(r) \cdot S(l)$, where $S(l)$ is the SIFT descriptor at location l with the same size as r. From the dense similarity map we compute the sparse detections set L^r as the five top scoring local maxima.

In each *feature generation* stage we generate features of a new type, where each feature is a scalar function over image windows: $f : I \mapsto \mathbb{R}$. The generation function gets the type to generate as input, as well as the previous generated and selected features (F_n,S_n correspondingly) and the fragment pool R, and creates new features of the desired type. For most of the feature types, new features are generated by transforming features from other types, already present in F_n or S_n. The dependency graph in figure 1(Left) shows the generation transformations currently supported in our system. The feature generation order may vary between experiments, as long as it conforms with the dependency graph. In our reported experiments features were generated roughly in the order at which they are presented below, with minor variations. (See Section 4 for more details).

Most of the features we generate represent different aspects of object-part detection, computed using the detection map L^r of one or more fragments. We mark by R_n the set of all the fragments used by the current feature set S_n. Below we describe the feature types implemented and their generation process.

HoG Features. We start with non-part based features obtained by applying HoG descriptors [1] on the entire image window I. The generation of HoG features is independent of the learning state.

GlobalMax Features. Given a fragment r and an image I, $f(I)$ is its maximal appearance score over all the image detections: $f(I) = \max_{l \in L^r} a^r(l)$. One max feature is generated per $r \in R$.

Sigmoid features. We extend each *globalMax* feature by applying a sigmoid function to the appearance score to enhance discriminative power: $f(I) = \max_{l \in L^r} \mathcal{G}(a^r(l))$, where $\mathcal{G}(x) = 1/(1 + \exp(-20 \cdot (x - \theta)))$. Sigmoid function parameter θ was chosen as the *globalMax* feature quantization threshold maximizing its mutual information with the class labels [15].

Localized features. We extend each *sigmoid* feature g by adding a localization score: $f(I) = \max_{l \in L^r} \mathcal{G}(a^r(l)) \cdot \mathcal{N}(l; \mu, \sigma I_{2\times 2})$ where $\mathcal{N}$ a 2D Gaussian function of the detection location l. Such features represent location sensitive part detection, attaining a high value when both the appearance score is high and the position is close to the Gaussian mean, similar to parts in a star-like model [21]. Two localized features with $\sigma = 10, 18$ were generated per *sigmoid* feature $g \in F_n$, with μ being the original image location of the fragment r.

Subpart features. A spatial sub-part is characterized by a subset B of the spatial bins in the SIFT descriptor, and specifically we consider the four quarters of the SIFT, with 2×2 spatial bins each. Given a localized feature g we compute the subpart feature as $f(I) = g(I) \cdot S^T(r) \mid_B \cdot S(l_{\max}) \mid_B$ where $l_{\max} \in L^r$ is the argmax location of the maximum operation in $g(I)$. This puts an additional emphasis on the similarity between specific subparts in the detection. We generate four such features for each *localized* feature $g \in S_n$.

LDA features. Given a *localized* feature g, we use the SIFT descriptor $S(l_{\max})$ computed for all training images to train a Linear Discriminant Analysis (LDA) [30] part classifier. The result is a 128 dimensional weight vector w, replacing the original fragment SIFT used in the original localized feature. The LDA feature is hence computed as $f = \max_{l \in L^r} \mathcal{G}(w \cdot S(l)) \cdot \mathcal{N}(l; \mu, \sigma I_{2\times 2})$.

"OR" features. For every two *localized* features $g \in S_n$ and $g' \in F_n$ we generate an "OR" feature computed as $f = \max(g, g')$ if their associated fragments originated in similar image locations. Such "OR" features aim to represent semantic object parts with multiple possible appearances. "OR" features with more than two fragments can be created using a recursive "OR" application in which g is already an "OR" feature.

Cue-integration features. Given a localized feature g we compute the co-occurrence descriptor [10] $CO(l_{\max})$ in all training images and train an LDA part classifier using them. The co-occurrence descriptor expresses texture information additional to SIFT, and the feature is computed as an LDA feature, but with $CO(l)$ replacing $S(l)$. Similarly we generate features that integrate both channels by concatenating the SIFT and co-occurrence descriptors.

"AND" features. Given two features based on fragments r, r' we compute their co-detection scores by $f = \max_{l \in L^r, l' \in L^{r'}} a^r(l) \cdot a^{r'}(l') \mathcal{N}_{rel}(l - l') \cdot \mathcal{N}_{abs}((l + l')/2)$. $\mathcal{N}_{rel}, \mathcal{N}_{abs}$ are Gaussian functions preferring a certain spatial relation between the

fragments and a certain absolute location, respectively. We generate several hundred such features by choosing pairs in which the score correlation in positive images is higher than the correlation in negative images.

Recall that after each type of features is added, we run the *feature selection* stage which is explained in the next section.

3 Predictive Feature Selection

An SVM gets as input a labeled sample $\{\mathbf{x}_i, y_i\}_{i=1}^{L}$ where $\mathbf{x}_i \in \mathbb{R}^M$ are training instances and $y_i \in \{-1, 1\}$ are their labels, and learns a classifier $sign(\mathbf{w} \cdot \mathbf{x} - b)$ by solving the quadratic program

$$\min_{\mathbf{w} \in R^M} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i=1}^{L}\xi_i \quad s.t.\ \forall i\ \ y_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1 - \xi_i \tag{1}$$

Denoting the Gram matrix by $\mathbf{K}$ (i.e. $\mathbf{K}_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$), the dual problem is

$$\max_{\boldsymbol{\alpha} \in R^L} ||\boldsymbol{\alpha}||_1 - \frac{1}{2}(\boldsymbol{\alpha} \otimes \mathbf{y})^T \mathbf{K}(\boldsymbol{\alpha} \otimes \mathbf{y}) \quad s.t.\ \ 0 \leq \boldsymbol{\alpha} + \boldsymbol{\eta} \leq C,\ 0 \leq \boldsymbol{\eta},\ \boldsymbol{\alpha}^T\mathbf{y} = 0 \tag{2}$$

where $\mathbf{y}$ is the vector of all labels and $\otimes$ stands for the element-wise vector product. Due to Kuhn-Tucker conditions, the optimal weight vector $\mathbf{w}$ can be expressed as $\mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x}_i$, where α_i are the components of the dual optimum α. Specifically, if we denote by x_i^j the j-th feature of $\mathbf{x}_i$ and by $\mathbf{x}^j = (x_1^j, .., x_L^j)$ the entire feature column, then the weight of feature j is given by

$$w_j = \sum_{i=1}^{L} \alpha_i y_i x_i^j = \boldsymbol{\alpha} \cdot (\mathbf{y} \otimes \mathbf{x}^j) \tag{3}$$

Applying this equation to features which were *not* seen during the SVM training can be regarded as weight prediction. The SVM-PFS[1] algorithm (see Algorithm 1) is an iterative pruning procedure which uses the square of the predicted weight as its feature ranking score.

SVM-PFS keeps two feature sets: a working set S with M features, on which SVM is trained, and a candidate feature set F, initially including $N \gg M$ features. In each round, SVM is trained on S, and the resulting α vector is used to compute feature scores. The features in $F \backslash S$ are sorted and the least promising candidates are dropped. Then the working set S is re-chosen by drawing features from the previous S, and from $F \setminus S$, where the ratio between the two parts of the new S is controlled by a stability parameter c. Features from both S and $F \setminus S$ are drawn with a probability proportional to the feature score. The process ends when the candidate feature set size reaches M.

While the algorithm is relatively simple, there are several tricky details worth noting. First, as will be discussed, the weight predictions are 'optimistic', and the actual weights are a lower bound for them. Hence the scores for S, (which are real

[1] Code available at http://sites.google.com/site/aharonbarhillel/

Algorithm 1. The SVM-PFS algorithm

Input: L labeled instances $\{\mathbf{x}_i, y_i\}_{i=1}^{L}$ where $\mathbf{x}_i \in \mathbb{R}^N$, the allowed $M < N$, a fraction parameter $t \in (0,1)$, stability parameter $c \in (0,1)$.
Output: A feature subset $S = \{i_1, .., i_M\}$ and an SVM classifier working on $\mathbf{x}|_s$.

Initialization:
Set the set of candidate features $F = \{1, 2, .., N\}$.
Normalize all the feature columns such that $\forall j \;\; E\mathbf{x}^j = 0\,,\; \|\mathbf{x}^j\|_\infty = 1$.
Initialize the working set S by drawing M different random features from F.
While $|F| > M$ do

1. Train an SVM on $\{\mathbf{x}_i|_s, y_i\}_{i=1}^{L}$ and keep the dual weight vector $\boldsymbol{\alpha}$.
2. For each feature $j \in F$ compute its score $h^j = h(\mathbf{x}^j) = \left(\sum_{i=1}^{L} \alpha_i y_i x_i^j\right)^2$.
3. Sort the scores $\{h^j | j \in F \setminus S\}$ in descending order and drop the last features from $F \setminus S$: $F = F \setminus \{j \in F \setminus S \,|\, Rank(h^j) > (1-t)|F|\,\} \cup S$,
4. Choose a new working set S by $S = S_1 \cup S_2$, where
   (a) S_1 is chosen from S by drawing cM features without replacement according to $p(j) = h^j / \sum_{j \in S} h^j$.
   (b) S_2 is chosen from $F \setminus S$ by drawing $(1-c)M$ features without replacement according to $p(j) = h^j / \sum_{j \in F \setminus S} h^j$.

Return S and the last computed classifier.

weights) and for $F \setminus S$ (which are weight predictions) are considered separately in steps 3,4. A second element is the randomness in the choice of S, which is important in order to break the symmetry between nearly identical features and to reduce feature redundancy. The RFE algorithm, which is deterministic, indeed does not cope well with redundant features [31], and PFS does a better job in discarding them. Finally, the l_∞ feature normalization is also important for performance. Beyond making the features comparable, this normalization gives a fixed bound on the radius of the ball containing all examples, which is important for SVM generalization [32]. While the l_2 norm could also be used, l_∞ is preferable as it is biased toward dense features, which are more stable, and it was found superior in all our experiments.

The computational complexity of SVM-PFS is analyzed in detail in the appendix. For the common case of $L \gg N/M$ it is $O(L^2 M)$, which is $Q\log(Q)$-faster than SVM-RFE [16] with $Q = N/M$. We next present a theoretical analysis of the algorithm.

3.1 Analysis

Soft SVM minimizes the loss $L(f) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{L}[1 - y_i f(x_i)]_+$ [33], for a classifier $f(\mathbf{x}; \mathbf{w}, b) = \mathbf{w} \cdot \mathbf{x} - b$ over a fixed feature set. SVM-PFS, like SVM-RFE, tries to minimize the same SVM loss, but for classifiers $f(\mathbf{x}; \mathbf{w}, b, S) = \mathbf{w} \cdot \mathbf{x}\,|_S - b$ restricted to a feature subset S. While SVM-RFE uses real SVM weights and backward elimination, PFS extends this to weight predictions and a combination of forward selection (into S) and backward elimination (applied

to F). In this section we justify the usage of weight predictions in forward and backward selection, and elaborate on the stability of PFS when working with large sets with many useless features. Proofs are deferred to the appendix.

Forward Selection: For backward elimination, it is shown in [16] that removing the feature with the smallest actual weight is the policy which least maximizes the SVM loss (in the limit of infinitesimally small weights). Here we give a similar result for forward selection with weight prediction rather than actual weight. Let $f(\mathbf{x}; \mathbf{w}, b, \{1, .., M\})$ be a classifier trained using soft SVM on the feature set $\{\mathbf{x}^j\}_{j=1}^{M}$, and $\{\mathbf{x}^j\}_{j=M+1}^{N}$ be a set of additional yet unseen features. In forward selection, our task is to choose the feature index $l \in \{M+1, .., N\}$ whose addition enables maximal reduction of the loss. Formally, we say that feature $\mathbf{x}^l$ is 'optimal in the small weight limit' iff

$$\exists \epsilon_0 > 0 \;\; \forall \epsilon < \epsilon_0 \;\; l = \underset{j \in \{M+1,..,N\}}{\operatorname{argmin}} \; \min_{\mathbf{w},b} L(f(\mathbf{x}; (\mathbf{w}, \epsilon), b, \{1, .., M\} \cup j)) \qquad (4)$$

The following theorem states the conditions under which the feature with the highest predicted weight is optimal:

Theorem 1. *If both the primal and the dual SVM solutions are unique then* $argmax_{j \in \{M+1,..N\}} (\sum_{i=1}^{L} y_i \alpha_i x_i^j)^2$ *is an optimal feature in the sense of eq. 4.*

The theorem is proved by considering the derivative of SVM's value function w.r.t the parameter perturbation caused by adding a new feature. Note that the non-uniqueness of the primal and dual solutions is an exception rather than the rule for the SVM problem [34].

Backward elimination with weight predictions: PFS uses (noisy) forward selection in choosing S, but most of its power is derived from the backward elimination process of F. While the backward elimination process based on real weights was justified in [16], the utility of the same process with weight predictions heavily depends on the accuracy of these predictions. Let $(\mathbf{w}^{old}, \boldsymbol{\alpha}^{old})$ be the (primal, dual) SVM solution for a feature set S with M features and $(\mathbf{w}^{new}, \boldsymbol{\alpha}^{new})$ be the SVM solution for $S \cup \{M+1\}$. Using Eq. 3 to predict the weight of the new feature relies on the intuition that adding a single feature usually induces only slight changes to $\boldsymbol{\alpha}$, and hence the real weight $w^{real} = \boldsymbol{\alpha}^{new} \cdot (\mathbf{x}^{M+1} \otimes \mathbf{y})$ is close to the predicted $w^{pred} = \boldsymbol{\alpha}^{old} \cdot (\mathbf{x}^{M+1} \otimes \mathbf{y})$. The following theorem quantifies the accuracy of the prediction, again in the small weight limit.

Theorem 2. *Assume that adding the new feature* $\mathbf{x}^{M+1}$ *did not change the set of support vectors, i.e.* $SV = \{i : \alpha_i^{old} > 0\} = \{i : \alpha_i^{new} > 0\}$. *Denote by* $\hat{\mathbf{K}}$ *the signed Gram matrix, i.e.* $\hat{\mathbf{K}}_{i,j} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$, *and by* $\hat{\mathbf{K}}_{sv}$ *the sub-matrix of* $\hat{\mathbf{K}}$ *with lines and columns in* SV. *If* $\hat{\mathbf{K}}_{sv}$ *is not singular (which is the case if* $M > |SV|$ *and the data points are in a general position) then*

$$\frac{\lambda_{min}(\hat{\mathbf{K}}_{sv})}{\|\mathbf{u}\|^2 + \lambda_{min}(\hat{\mathbf{K}}_{sv})} w^{pred} \leq w^{real} = \frac{w^{pred}}{1 + u^T \hat{\mathbf{K}}_{sv}^{-1} u} \leq w^{pred} \qquad (5)$$

where $\mathbf{u} = \mathbf{y} \otimes \mathbf{x}^{M+1}|_{sv}$, *and* $\lambda_{min}(\hat{\mathbf{K}}_{sv})$ *is the smallest eigenvalue of* $\hat{\mathbf{K}}_{sv}$.

The assumptions of theorem 2 are rather restrictive, but they often hold when a new feature is added to a large working set (i.e. M is on the order of 10^3). In this case the weight of the new feature is small, and so are the changes to the vector $\boldsymbol{\alpha}$ and the support vector set. The theorem states that under these conditions w^{pred} upper bounds the real weight, allowing us to safely drop features with low predicted weight. Furthermore, as features are accumulated $\lambda_{min}(\hat{\mathbf{K}}_{sv})$ rises, improving the left hand bound in Eq. 5 and entailing better weight prediction. For small M, weight prediction can be improved by adding a small constant ridge to the Gram matrix diagonal, thus raising $\lambda_{min}(\hat{\mathbf{K}}_{sv})$. These phenomena are empirically demonstrated in Section 4.1.

Robustness to noise and initial conditions: In PFS a small subset of features is used to predict weights for a very large feature set, often containing mostly 'garbage' features. Hence it may seem that a good initial S set is required, and that large quantities of bad features in S will lead to random feature selection. The following theorem shows that this is not the case:

Theorem 3. *Assume S contains $M \gg L$ random totally uninformative features, created by choosing x_i^j independently from a symmetric distribution with moments $Ex = 0$, $Ex^2 = 1$, $Ex^4 = J$. Then with probability approaching 1 as $M \to \infty$ all the examples are support vectors and we have*

$$\forall i \quad \alpha_i = \frac{1}{M}(1 + \frac{C_1}{\sqrt{M}}\xi_i)$$

$$\forall \mathbf{x} \in S \quad h(\mathbf{x}) \propto \rho(\mathbf{x}, \mathbf{y})(1 + \frac{C_2}{\sqrt{M}}\xi_h)$$

where $\rho(w, z) = (1/\sigma(w)\sigma(z)) \cdot E[(w - Ew)(z - Ez)]$ is the Pearson correlation, $\xi_i, \xi_h \sim N(0, 1)$, and C_1, C_2 are $O(\sqrt{L}, \sqrt{J})$ and constant w.r.t M.

Theorem 3 states that if S contains many 'garbage' features, weight predictions tend toward the Pearson correlation, a common feature selection filter [35]. This guarantees robustness w.r.t to initial conditions and large amounts of bad features. While proved only for $M \gg L$, the convergence of $h(\mathbf{x})$ toward the Pearson coefficient is fast and observed empirically in the first PFS rounds.

4 Empirical Evaluation

In Section 4.1 we evaluate the PFS algorithm and compare it to several relevant feature selection methods. We then present human detection results for our full system in Section 4.2.

4.1 PFS Evaluation

PFS initial evaluation: In all our reported experiments, we use default values for the algorithm parameters of $c = 0.2$ and $t = 0.5$. We used a subset of MNIST, where the task is to discriminate the digits '5' and '8' to check the quality of

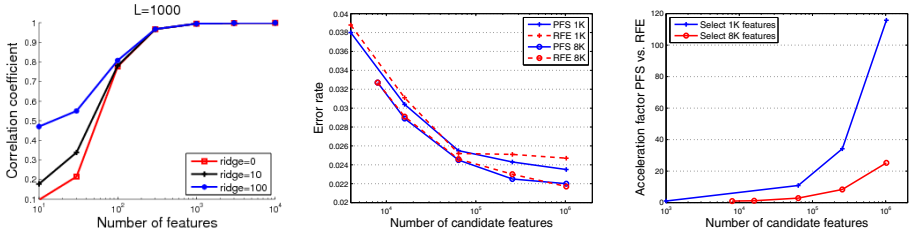


Fig. 2. Left: Correlation between predicted and actual weights as a function of working set size M, on an Mnist[36] subset. The features are randomly selected monomials that include two input pixels. Graphs are shown for three ridge (small constant added to the diagonal of the Gram matrix) values. Actual weights are computed by adding the relevant feature to the working set and re-training SVM. **Center:**Accuracy of SVM-RFE and SVM-PFS as a function of the candidate feature set size N for PK-GINA data set. **Right:**Ratio between actual run time of RFE vs. run time of PFS as a function of N for the PK-GINA dataset. PFS runs in less than an hour. The accuracy of both methods is very similar and both benefit from an increasingly larger candidate set. However, the training time of SVM-RFE increases considerably with feature set size, whereas SVM-PFS increases slowly (See appendix for analysis). PFS shows the greatest gain when selecting from a large candidate set.

weight prediction, on which PFS relies. The results, appearing in figure 2(Left) indicated highly reliable weight prediction if the number of features M in S is larger than 500. We then conducted a large scale experiment with the GINA-PK digit recognition dataset from the IJCNN-2007 challenge [37], in which examples are digits of size 28×28 pixels, and the discrimination is between odd and even digits. The features we used are maximum of monomials of certain pixel configurations over a small area. For example, for an area of 2×2 and a monomial of degree 2 the features have the form $\max_{i\in\{0,1\},j\in\{0,1\}}(P_{x_1+i,y_1+j}P_{x_2+i,y_2+j})$ where $P_{x,y}$ is the pixel value at (x, y) . We generated 1,024,000 features with random parameters. PFS and RFE were evaluated for choosing $M = 1000, 8000$ features out of $N = 8000, \ldots, 1,024,000$. The results of these experiments are shown in figure 2(Middle,Right). While the accuracy of PFS and RFE is very similar, PFS was up to $\times 116$ faster. Notably, our results ranked second best in the challenge (post challenge submission).

Comparison with other feature selection algorithms: We compared PFS to Mutual information max-min [15]+SVM, Adaboost [17], Column generation SVM [18] and SVM-RFE in the task of choosing $M = 500$ features among $N = 19,560$ using a subset of 1700 training samples from the INRIA pedestrian dataset. The feature set included *sigmoid* and *localized* features based on 4000 fragments and *HOG* features. The *per-window* INRIA-test results are presented in figure 3(a). PFS and RFE give the best results, but PFS was an order of magnitude faster. As a further baseline, note that SVM trained over 500 random features achieved a miss rate of 85% at 10^{-4} FPPW, and with 500 features selected using the Pearson correlation filter it achieved 72%.

4.2 Feature Synthesis for Human Detection

Implementation Details. We used 20,000 fragments as our basic fragment pool R, with sizes ranging from 12×12 to 80×40 pixels. In initial stages we generated a total of $20,000$ *globalMax*, $20,000$ *sigmoid*, $40,000$ *localized* and $3,780$ *HoG*. We then sequentially added the *subparts*, *LDA*, *"OR"*, *cue-integration* and *"AND"* features. In all our experiments we used PFS to choose $M = 500$ features. SVM was used with $C = 0.01$ and a ridge value of 0.01.

INRIA pedestrian [1] results. Following the conclusions of [7] we evaluated our method using both the full-image evaluation based on the PASCAL criteria and the traditional *per-window* evaluation. For the full image evaluation we used a 2-stage classifier cascade to speed up detection. The first stage consists of a HoG classifier [1] adjusted to return 200 detections per image, which are then handed to our classifier. Our non maxima suppression procedure suppresses the less confident window of each pair if their overlap area divided by their union area is greater than $\frac{3}{4}$ or if the less confident window is fully included in the more confident one. We experimented with several mixtures of monotonic and non monotonic steps of the generation process.

The results of our method (**FeatSynth**) are compared to other methods in figure 3(b,c). Further details regarding the tested algorithms and evaluation methodology appear in [6],[7] respectively. These results were obtained using monotonic generation steps in the initial stages (HoG, globalMax, sigmoid and localized features), followed by non monotonic generation steps of *subparts*, *LDA* and *cue-integration* features. At the full image evaluation FeatSynth achieved a 89.3% detection rate at 1 false positive per image (FPPI) outperforming all methods except for LatSvm-V2 [4] (90.7%). Combination features (*"OR"* and *"AND"*) did not add to the performance of this classifier and were therefore omitted. However, these features did contribute when all feature synthesis was done using only non-monotonic steps, which resulted in a slightly worse classifier. Figure 3(d) shows the gradual improvement of the detection rate for the non-monotone process at 1,1/3 and 1/10 FPPI.

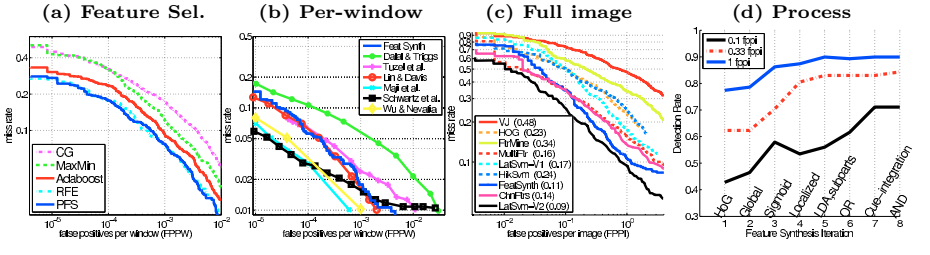


Fig. 3. INRIA pedestrian results. (a) Feature selection comparison on a sample of the INRIA dataset. **(b)** Per-window DET curve. **(c)** Full image evaluation on the 288 positive test images. **(d)** Feature synthesis process evaluation on full images. See text for details.

Caltech pedestrian [7] results. The Caltech pedestrian *training* dataset contains six sessions (0-5), each with multiple videos taken from a moving vehicle. We followed the evaluation methodology from [7], applying our detector on every 30^{th} frame, with a total of $4,306$ test images, an order of magnitude larger than the INRIA test. As before we used a 2-stage classifier cascade, with the same classifier trained on the INRIA dataset as the second stage classifier, and the Feature Mining classifier [3], which performs best at low miss rates, as the first stage. Figure 4 shows a comparison to 8 algorithms evaluated in [7] on different subsets of the dataset. We refer the reader to [7] for further details on the tested algorithms and evaluation methodology. In the overall evaluation (fig. 4(a)) FeatSynth achieved a 30% detection rate at 1 FPPI improving the current state-of-the-art, 25%, by MultiFtr [2,7]. FeatSynth also outperformed the other methods on 50-pixel or taller, un-occluded or partially occluded pedestrians (fig. 4(b)) and on all other dataset subsets except for near pedestrians, proving

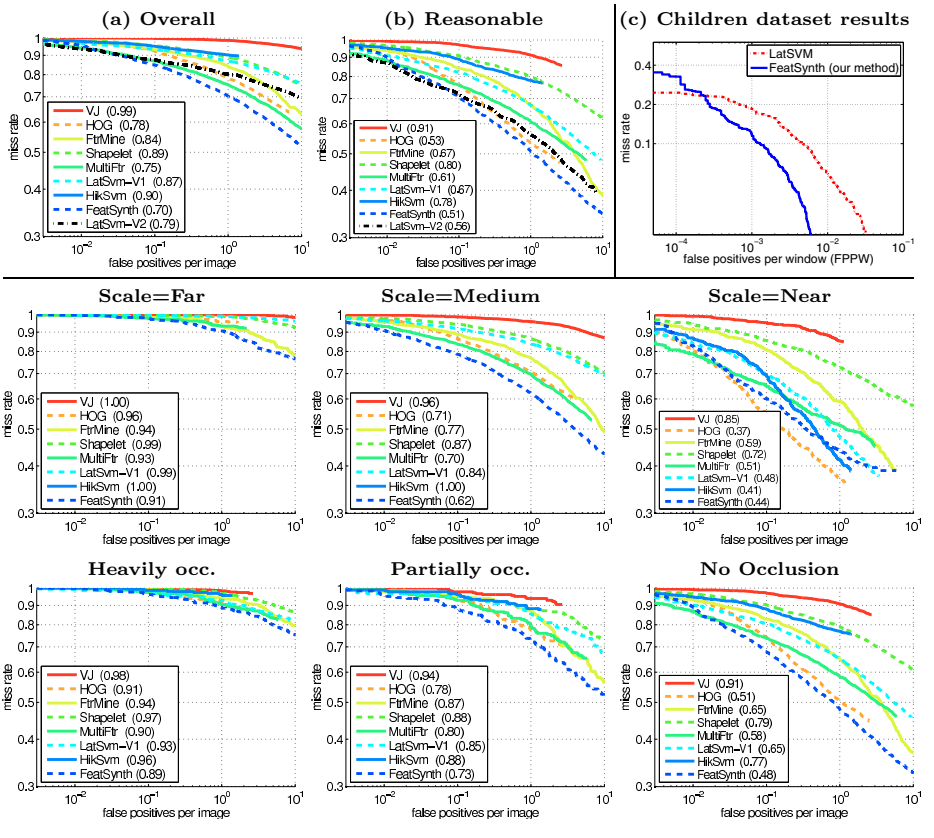


Fig. 4. Results on Caltech pedestrian training dataset and partitions (All except top-right) and child detection results (top-right). See text for details.

robustness to varying conditions. Since our classifier (as most other methods) was trained on the INRIA, this evaluation demonstrates its generalization power.

Child detection: We created a *Children* dataset of 109 short video clips, of which 82 contain 13 children performing various activities such as crawling, riding a bike or a toy car, sitting and playing. The other 27 clips were taken in the same sites but without the children. These data are mostly relevant for an automotive application of identifying children in the rear camera of a backing vehicle. Based on the videos we compiled a dataset of 2300 children's images, split evenly into train and test, where children from the training set do not appear in the test set. Half the negative images were extracted from the 27 non-children clips, and the other half from the INRIA negative images set. The dataset is rather challenging due to the high pose variance of the children (see figure 1(Left)), making it difficult for the template-based methods used for pedestrians to succeed. The results of our method appear in figure 4(Top-Right), and compared to the part-based method of [4] trained on the children data with 2 components.

5 Conclusions and Future Work

We presented a new methodology for part-based feature learning and showed its utility on known pedestrian detection benchmarks. The paradigm we suggest is highly flexible, allowing for fast exploration of new feature types. We believe that several directions may alleviate the method farther: enable more generation transformations, automate the search over feature synthesis order, introduce negative examples mining into the process, and introduce human guidance as a 'weak learner' into the loop.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
2. Wojek, C., Schiele, B.: A performance evaluation of single and multi-feature people detection. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 82–91. Springer, Heidelberg (2008)
3. Dollár, P., Tu, Z., Tao, H., Belongie, S.: Feature mining for image classification. In: CVPR (2007)
4. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
5. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR (2008)
6. Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human detection using partial least squares analysis. In: ICCV (2009)
7. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR (2009)
8. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV (2009)

9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
10. Haralick, R., Shanmugam, K., Dinstein, I.: Texture features for image classification. IEEE Transactions on Systems,Man, and Cybernetics 3(6) (1973)
11. Dollar, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: BMVC (2009)
12. Popper, K.: Objective knowledge: An Evolutionary Approach. Clarendon Press, Oxford (1972)
13. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)
14. Ullman, S., Sali, E., Vidal-Naquet, M.: A fragment-based approach to object representation and classification. In: Arcelli, C., Cordella, L.P., Sanniti di Baja, G. (eds.) IWVF 2001. LNCS, vol. 2059, p. 85. Springer, Heidelberg (2001)
15. Vidal-Naquet, M., Ullman, S.: Object recognition with informative features and linear classification. In: ICCV
16. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46 (2002)
17. Schapire, R.E., Singer, Y.: Improved boosting using confidence-rated predictions. Machine Learning 37, 297–336 (1999)
18. Bi, J., Zhang, T., Bennett, K.P.: Column-generation boosting methods for mixture of kernels. In: KDD (2004)
19. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale invariant learning. In: CVPR (2003)
20. Karlinsky, L., Dinerstein, M., Levi, D., Ullman, S.: Unsupervised classification and part localization by consistency amplification. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 321–335. Springer, Heidelberg (2008)
21. Bar-Hillel, A., Weinshall, D.: Efficient learning of relational object class models. IJCV (2008)
22. Tversky, B., Hemenway, K.: Objects, parts, and categories. Journal of Experimental Psychology: General 113(2), 169–197 (1984)
23. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: ICCV, vol. 1 (2005)
24. Ullman, S., Epshtein, B.: Visual classification by a hierarchy of extended fragments. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 321–344. Springer, Heidelberg (2006)
25. Vapnik, V.: The Nature Of Statistical Learning Theory. Springer, Heidelberg (1995)
26. Lee, J.W., Lee, J.B., Park, M., Sonq, S.H.: An extensive comparison of recent classification tools applied to microarray data. Computational statistics and Data Analysis 48(4), 869–885 (2005)
27. Rakotomamonjy, A.: Variable selection using svm-based criteria. JMLR, 1357–1370 (2003)
28. Weston, J., Elisseeff, A., Schoelkopf, B., Tipping, M.: Use of the zero norm with linear models and kernel methods. JMLR 3, 1439–1461 (2003)
29. Perkins, S., Lacker, K., Theiler, J.: Grafting: Fast incremental feature selection by gradient descent in function space. JMLR 3
30. Fukunaga, K.: Statistical Pattern Recognition, 2nd edn. Academic Press, San Diego (1990)
31. Xie, Z.X., Hu, Q.H., Yu, D.R.: Improved feature selection algorithm based on svm and correlation. In: NIPS, pp. 1373–1380 (2006)

32. Shivaswamy, P., Jebara, T.: Ellipsoidal kernel machines. In: AISTATS (2007)
33. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning; Data mining, Inference and Prediction. Springer, Heidelberg (2001)
34. Burges, J., Crisp, D.: Uniqueness of the svm solution. In: NIPS (1999)
35. Hall, M., Smith, L.: Feature subset selection: a correlation based filter approach. In: International Conference on Neural Information Processing and Intelligent Information Systems, pp. 855–858 (1997)
36. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324 (1998)
37. Guyon, I., Saffari, A., Dror, G., Cawley, G.C.: Agnostic learning vs. prior knowledge challenge. In: IJCNN (2007)

Improving the Fisher Kernel for Large-Scale Image Classification

Florent Perronnin, Jorge Sánchez, and Thomas Mensink

Xerox Research Centre Europe (XRCE)

Abstract. The Fisher kernel (FK) is a generic framework which combines the benefits of generative and discriminative approaches. In the context of image classification the FK was shown to extend the popular bag-of-visual-words (BOV) by going beyond count statistics. However, in practice, this enriched representation has not yet shown its superiority over the BOV. In the first part we show that with several well-motivated modifications over the original framework we can boost the accuracy of the FK. On PASCAL VOC 2007 we increase the Average Precision (AP) from 47.9% to 58.3%. Similarly, we demonstrate state-of-the-art accuracy on CalTech 256. A major advantage is that these results are obtained *using only SIFT descriptors and costless linear classifiers*. Equipped with this representation, we can now explore image classification on a larger scale. In the second part, as an application, we compare two abundant resources of labeled images to learn classifiers: ImageNet and Flickr groups. In an evaluation involving hundreds of thousands of training images we show that classifiers learned on Flickr groups perform surprisingly well (although they were not intended for this purpose) and that they can complement classifiers learned on more carefully annotated datasets.

1 Introduction

We consider the problem of learning image classifiers on large annotated datasets, *e.g.* using hundreds of thousands of labeled images. Our goal is to devise an image representation which yields high classification accuracy, yet which is efficient. Efficiency includes the cost of computing the representations, the cost of learning classifiers on these representations as well as the cost of classifying a new image.

One of the most popular approaches to image classification to date has been to describe images with bag-of-visual-words (BOV) histograms and to classify them using non-linear Support Vector Machines (SVM) [1]. In a nutshell, the BOV representation of an image is computed as follows. Local descriptors are extracted from the image and each descriptor is assigned to its closest visual word in a "visual vocabulary": a codebook obtained offline by clustering a large set of descriptors with k-means. There have been several extensions of this initial idea including[1] the soft-assignment of patches to visual words [2,3] or the use of spatial pyramids to take into account the image structure [4]. A trend in BOV

[1] An extensive overview of the BOV falls out of the scope of this paper.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 143–156, 2010.

approaches is to have multiple combinations of patch detectors, descriptors and spatial pyramids (where a combination is often referred to as a "channel"), to train one classifier per channel and then to combine the output of the classifiers [5,6,7,3]. Systems following this paradigm have consistently performed among the best in the successive PASCAL VOC evaluations [8,9,10].

An important limitation of such approaches is their scalability to large quantities of training images. First, the feature extraction of many channels comes at a high cost. Second, the learning of non-linear SVMs scales somewhere between $O(N^2)$ and $O(N^3)$ – where N is the number of training images – and becomes impractical for N in the tens or hundreds of thousands. This is in contrast with linear SVMs whose training cost is in $O(N)$ [11,12] and which can therefore be efficiently learned with large quantities of images [13]. However linear SVMs have been repeatedly reported to be inferior to non-linear SVMs on BOV histograms [14,15,16,17].

Several algorithms have been proposed to reduce the training cost. Combining Spectral Regression with Kernel Discriminant Analysis (SR-KDA), Tahir *et al.* [7] report a faster training time and a small accuracy improvement over the SVM. However SR-KDA still scales in $O(N^3)$. Wang *et al.* [14], Maji and Berg [15], Perronnin *et al.* [16] and Vedaldi and Zisserman [17] proposed different approximations for additive kernels. These algorithms scale linearly with the number of training samples while providing the same accuracy as the original non-linear SVM classifiers. Rather than modifying the classifiers, attempts have been made to obtain BOV representations which perform well with linear classifiers. Yang *et al.* [18] proposed a sparse coding algorithm to replace K-means clustering and a max- (instead of average-) pooling of the descriptor-level statistics. It was shown that excellent classification results could be obtained with linear classifiers – interestingly much better than with non-linear classifiers.

We stress that all the methods mentioned previously are inherently limited by the shortcomings of the BOV representation, and especially by the fact that the descriptor quantization is a lossy process as underlined in the work of Boiman *et al.* [19]. Hence, efficient alternatives to the BOV histogram have been sought. Bo and Sminchisescu [20] proposed the Efficient Match Kernel (EMK) which consists in mapping the local descriptors to a low-dimensional feature space and in averaging these vectors to form a fixed-length image representation. They showed that a linear classifier on the EMK representation could outperform a non-linear classifier on the BOV. However, this approach is limited by the assumption that the same kernel can be used to measure the similarity between two descriptors, whatever their location in the descriptor space.

In this work we consider the Fisher Kernel (FK) introduced by Jaakkola and Haussler [21] and applied by Perronnin and Dance [22] to image classification. This representation was shown to extend the BOV: it is not limited to the number of occurrences of each visual word but it also encodes additional information about the distribution of the descriptors. Therefore the FK overcomes some of the limitations raised by [19]. Yet, in practice, the FK has led to somewhat disappointing results – no better than the BOV.

The contributions of this paper are two-fold:

1. First, we propose several well-motivated improvements over the original Fisher representation and show that they boost the classification accuracy. For instance, on the PASCAL VOC 2007 dataset we increase the Average Precision (AP) from 47.9% to 58.3%. On the CalTech 256 dataset we also demonstrate state-of-the-art performance. A major advantage is that these results are obtained *using only SIFT descriptors and costless linear classifiers*. Equipped with this representation, we can then explore image classification on a larger scale.
2. Second, we compare two abundant sources of training images to learn image classifiers: ImageNet[2] [23] and Flickr groups[3]. In an evaluation involving hundreds of thousands of training images we show that classifiers learned on Flickr groups perform surprisingly well (although Flickr groups were not intended for this purpose) and that they can nicely complement classifiers learned on more carefully annotated datasets.

The remainder of this article is organized as follows. In the next section we provide a brief overview of the FK. In section 3 we describe the proposed improvements and in section 4 we evaluate their impact on the classification accuracy. In section 5, using this improved representation, we compare ImageNet and Flickr groups as sources of labeled training material to learn image classifiers.

2 The Fisher Vector

Let $X = \{x_t, t = 1 \dots T\}$ be the set of T local descriptors extracted from an image. We assume that the generation process of X can be modeled by a probability density function u_λ with parameters λ [4]. X can be described by the gradient vector [21]:

$$G_\lambda^X = \frac{1}{T} \nabla_\lambda \log u_\lambda(X). \tag{1}$$

The gradient of the log-likelihood describes the contribution of the parameters to the generation process. The dimensionality of this vector depends only on the number of parameters in λ, not on the number of patches T. A natural kernel on these gradients is [21]:

$$K(X, Y) = G_\lambda^{X\prime} F_\lambda^{-1} G_\lambda^Y \tag{2}$$

where F_λ is the Fisher information matrix of u_λ:

$$F_\lambda = E_{x \sim u_\lambda} \left[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)' \right]. \tag{3}$$

[2] http://www.image-net.org
[3] http://www.flickr.com/groups
[4] We make the following abuse of notation to simplify the presentation: λ denotes both the set of parameters of u as well as the estimate of these parameters.

As F_λ is symmetric and positive definite, it has a Cholesky decomposition $F_\lambda = L'_\lambda L_\lambda$ and $K(X,Y)$ can be rewritten as a dot-product between normalized vectors $\mathcal{G}_\lambda$ with:

$$\mathcal{G}^X_\lambda = L_\lambda G^X_\lambda. \tag{4}$$

We will refer to $\mathcal{G}^X_\lambda$ as the *Fisher vector* of X. We underline that *learning a kernel classifier using the kernel (2) is equivalent to learning a linear classifier on the Fisher vectors* $\mathcal{G}^X_\lambda$. As explained earlier, learning linear classifiers can be done extremely efficiently.

We follow [22] and choose u_λ to be a Gaussian mixture model (GMM): $u_\lambda(x) = \sum_{i=1}^K w_i u_i(x)$. We denote $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1 \dots K\}$ where w_i, μ_i and Σ_i are respectively the mixture weight, mean vector and covariance matrix of Gaussian u_i. We assume that the covariance matrices are diagonal and we denote by σ_i^2 the variance vector. The GMM u_λ is trained on a large number of images using Maximum Likelihood (ML) estimation . It is supposed to describe the content of *any* image. We assume that the x_t's are generated independently by u_λ and therefore:

$$G^X_\lambda = \frac{1}{T} \sum_{t=1}^T \nabla_\lambda \log u_\lambda(x_t). \tag{5}$$

We consider the gradient with respect to the mean and standard deviation parameters (the gradient with respect to the weight parameters brings little additional information). We make use of the diagonal closed-form approximation of [22], in which case the normalization of the gradient by $L_\lambda = F_\lambda^{-1/2}$ is simply a whitening of the dimensions. Let $\gamma_t(i)$ be the soft assignment of descriptor x_t to Gaussian i:

$$\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{j=1}^K w_j u_j(x_t)}. \tag{6}$$

Let D denote the dimensionality of the descriptors x_t. Let $\mathcal{G}^X_{\mu,i}$ (resp. $\mathcal{G}_{\sigma,i}$) be the D-dimensional gradient with respect to the mean μ_i (resp. standard deviation σ_i) of Gaussian i. Mathematical derivations lead to:

$$\mathcal{G}^X_{\mu,i} = \frac{1}{T\sqrt{w_i}} \sum_{t=1}^T \gamma_t(i) \left(\frac{x_t - \mu_i}{\sigma_i} \right), \tag{7}$$

$$\mathcal{G}^X_{\sigma,i} = \frac{1}{T\sqrt{2w_i}} \sum_{t=1}^T \gamma_t(i) \left[\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right], \tag{8}$$

where the division between vectors is as a term-by-term operation. The final gradient vector $\mathcal{G}^X_\lambda$ is the concatenation of the $\mathcal{G}^X_{\mu,i}$ and $\mathcal{G}^X_{\sigma,i}$ vectors for $i = 1 \dots K$ and is therefore $2KD$-dimensional.

3 Improving the Fisher Vector

3.1 L2 Normalization

We assume that the descriptors $X = \{x_t, t = 1 \dots T\}$ of a given image follow a distribution p. According to the law of large numbers (convergence of the sample average to the expected value when T increases) we can rewrite equation (5) as:

$$G_\lambda^X \approx \nabla_\lambda E_{x\sim p} \log u_\lambda(x) = \nabla_\lambda \int_x p(x) \log u_\lambda(x) dx. \tag{9}$$

Now let us assume that we can decompose p into a mixture of two parts: a background image-independent part which follows u_λ and an image-specific part which follows an image-specific distribution q. Let $0 \leq \omega \leq 1$ be the proportion of image-specific information contained in the image:

$$p(x) = \omega q(x) + (1 - \omega) u_\lambda(x). \tag{10}$$

We can rewrite:

$$G_\lambda^X \approx \omega \nabla_\lambda \int_x q(x) \log u_\lambda(x) dx + (1 - \omega) \nabla_\lambda \int_x u_\lambda(x) \log u_\lambda(x) dx. \tag{11}$$

If the values of the parameters λ were estimated with a ML process – i.e. to maximize (at least locally and approximately) $E_{x\sim u_\lambda} \log u_\lambda(x)$ – then we have:

$$\nabla_\lambda \int_x u_\lambda(x) \log u_\lambda(x) dx = \nabla_\lambda E_{x\sim u_\lambda} \log u_\lambda(x) \approx 0. \tag{12}$$

Consequently, we have:

$$G_\lambda^X \approx \omega \nabla_\lambda \int_x q(x) \log u_\lambda(x) dx = \omega \nabla_\lambda E_{x\sim q} \log u_\lambda(x). \tag{13}$$

This shows that the image-independent information is approximately discarded from the Fisher vector signature, a positive property. Such a decomposition of images into background and image-specific information has also been employed in BOV approaches by Zhang *et al.* [24]. However, while the decomposition is *explicit* in [24], it is *implicit* in the FK case.

We note that the signature still depends on the proportion of image-specific information ω. Consequently, two images containing the same object but different amounts of background information (*e.g.* same object at different scales) will have different signatures. Especially, small objects with a small ω value will be difficult to detect. To remove the dependence on ω, we can L2-normalize[5]

[5] Actually dividing the Fisher vector by any Lp norm would cancel-out the effect of ω. We chose the L2 norm because it is the natural norm associated with the dot-product.

the vector G_λ^X or equivalently $\mathcal{G}_\lambda^X$. We follow the latter option which is strictly equivalent to replacing the kernel (2) with:

$$\frac{K(X,Y)}{\sqrt{K(X,X)K(Y,Y)}} \tag{14}$$

To our knowledge, this simple L2 normalization strategy has never been applied to the Fisher kernel in a categorization scenario.

This is not to say that the L2 norm of the Fisher vector is not discriminative. Actually, $||\mathcal{G}_\lambda^X|| = \omega||\nabla_\lambda E_{x\sim q} \log u_\lambda(x)||$ and the second term may contain class-specific information. In practice, removing the dependence on the L2 norm (*i.e.* on both ω and $||\nabla_\lambda E_{x\sim q} \log u_\lambda(x)||$) can lead to large improvements.

3.2 Power Normalization

The second improvement is motivated by an empirical observation: as the number of Gaussians increases, Fisher vectors become sparser. This effect can be easily explained: as the number of Gaussians increases, fewer descriptors x_t are assigned with a significant probability $\gamma_t(i)$ to each Gaussian. In the case where no descriptor x_t is assigned significantly to a given Gaussian i (i.e. $\gamma_t(i) \approx 0$, $\forall t$), the gradient vectors $\mathcal{G}_{\mu,i}^X$ and $\mathcal{G}_{\sigma,i}^X$ are close to null (c.f. equations (7) and (8)). Hence, as the number of Gaussians increases, the distribution of features in a given dimension becomes more peaky around zero, as exemplified in Fig 1.

We note that the dot-product on L2 normalized vectors is equivalent to the L2 distance. Since the dot-product / L2 distance are poor measures of similarity on sparse vectors, we are left with two choices:

- We can replace the dot-product by a kernel which is more robust on sparse vectors. For instance, we can choose the Laplacian kernel which is based on the L1 distance[6].
- An alternative possibility is to "unsparsify" the representation so that we can keep the dot-product similarity.

While in preliminary experiments we did observe an improvement with the Laplacian kernel, a major disadvantage with this option is that we have to pay the cost of non-linear classification. Therefore we favor the latter option.

We propose to apply in each dimension the following function:

$$f(z) = \text{sign}(z)|z|^\alpha \tag{15}$$

where $0 \leq \alpha \leq 1$ is a parameter of the normalization. We show in Fig 1 the effect of this normalization. We experimented with other functions $f(z)$ such as $\text{sign}(z)\log(1+\alpha|z|)$ or $\text{asinh}(\alpha z)$ but did not improve over the power normalization.

[6] The fact that L1 is more robust than L2 on sparse vectors is well known in the case of BOV histograms: see *e.g.* the work of Nistér and Stewénius [25].

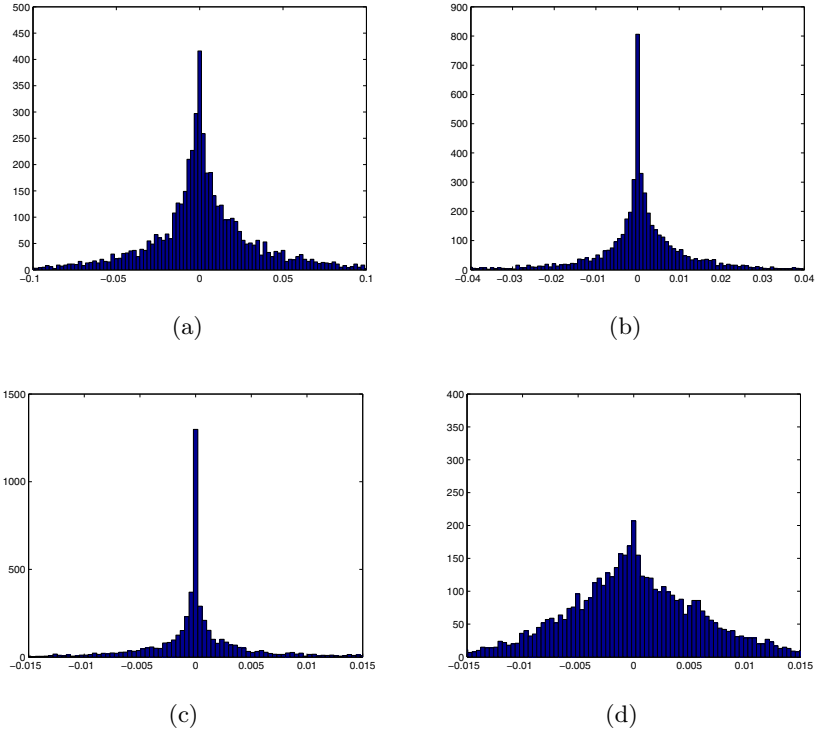


Fig. 1. Distribution of the values in the first dimension of the L2-normalized Fisher vector. (a), (b) and (c): resp. 16 Gaussians, 64 Gaussians and 256 Gaussians with no power normalization. (d): 256 Gaussians with power normalization ($\alpha = 0.5$). Note the different scales. All the histograms have been estimated on the 5,011 training images of the PASCAL VOC 2007 dataset.

The optimal value of α may vary with the number K of Gaussians in the GMM. In all our experiments, we set $K = 256$ as it provides a good compromise between computational cost and classification accuracy. Setting $K = 512$ typically increases the accuracy by a few decimals at twice the computational cost. In preliminary experiments, we found that $\alpha = 0.5$ was a reasonable value for $K = 256$ and this value is fixed throughout our experiments.

When combining the power and the L2 normalizations, we apply the power normalization first and then the L2 normalization. We note that this does not affect the analysis of the previous section: the L2 normalization on the power-normalized vectors sill removes the influence of the mixing coefficient ω.

3.3 Spatial Pyramids

Spatial pyramid matching was introduced by Lazebnik *et al.* to take into account the rough geometry of a scene [4]. It consists in repeatedly subdividing an image and computing histograms of local features at increasingly fine

resolutions by pooling descriptor-level statistics. The most common pooling strategy is to average the descriptor-level statistics but max-pooling combined with sparse coding was shown to be very competitive [18]. Spatial pyramids are very effective both for scene recognition [4] and loosely structured object recognition as demonstrated during the PASCAL VOC evaluations [8,9].

While to our knowledge the spatial pyramid and the FK have never been combined, this can be done as follows: instead of extracting a BOV histogram in each region, we extract a Fisher vector. We use average pooling as this is the natural pooling mechanism for Fisher vectors (c.f. equations (7) and (8)). We follow the splitting strategy adopted by the winning systems of PASCAL VOC 2008 [9]. We extract 8 Fisher vectors per image: one for the whole image, three for the top, middle and bottom regions and four for each of the four quadrants.

In the case where Fisher vectors are extracted from sub-regions, the "peakiness" effect will be even more exaggerated as fewer descriptor-level statistics are pooled at a region-level compared to the image-level. Hence, the power normalization is likely to be even more beneficial in this case.

When combining L2 normalization and spatial pyramids we L2 normalize each of the 8 Fisher vectors independently.

4 Evaluation of the Proposed Improvements

We first describe our experimental setup. We then evaluate the impact of the three proposed improvements on two challenging datasets: PASCAL VOC 2007 [8] and CalTech 256 [26].

4.1 Experimental Setup

We extract features from 32×32 pixel patches on regular grids (every 16 pixels) at five scales. In most of our experiments we make use only of 128-D SIFT descriptors [27]. We also consider in some experiments simple 96-D color features: a patch is subdivided into 4×4 sub-regions (as is the case of the SIFT descriptor) and we compute in each sub-region the mean and standard deviation for the three R, G and B channels. Both SIFT and color features are reduced to 64 dimensions using Principal Component Analysis (PCA).

In all our experiments we use GMMs with $K = 256$ Gaussians to compute the Fisher vectors. The GMMs are trained using the Maximum Likelihood (ML) criterion and a standard Expectation-Maximization (EM) algorithm. We learn linear SVMs with a hinge loss using the primal formulation and a Stochastic Gradient Descent (SGD) algorithm [12][7]. We also experimented with logistic regression but the learning cost was higher (approx. twice as high) and we did not observe a significant improvement. When using SIFT and color features, we train two systems separately and simply average their scores (no weighting).

[7] An implementation is available on Léon Bottou's webpage: http://leon.bottou.org/projects/sgd

Table 1. Impact of the proposed modifications to the FK on PASCAL VOC 2007. "PN" = power normalization. "L2" = L2 normalization. "SP" = Spatial Pyramid. The first line (no modification applied) corresponds to the baseline FK of [22]. Between parentheses: the absolute improvement with respect to the baseline FK. Accuracy is measured in terms of AP (in %).

PN	L2	SP	SIFT		Color		SIFT + Color	
No	No	No	47.9		34.2		45.9	
Yes	No	No	54.2	(+6.3)	45.9	(+11.7)	57.6	(+11.7)
No	Yes	No	51.8	(+3.9)	40.6	(+6.4)	53.9	(+8.0)
No	No	Yes	50.3	(+2.4)	37.5	(+3.3)	49.0	(+3.1)
Yes	Yes	No	55.3	(+7.4)	47.1	(+12.9)	58.0	(+12.1)
Yes	No	Yes	55.3	(+7.4)	46.5	(+12.3)	57.5	(+11.6)
No	Yes	Yes	55.5	(+7.6)	45.8	(+11.6)	56.9	(+11.0)
Yes	Yes	Yes	58.3	(+10.4)	50.9	(+16.7)	60.3	(+14.4)

4.2 PASCAL VOC 2007

The PASCAL VOC 2007 dataset [8] contains around 10K images of 20 object classes. We use the standard protocol which consists in training on the provided "trainval" set and testing on the "test" set. Classification accuracy is measured using Average Precision (AP). We report the average over the 20 classes. To tune the SVM regularization parameters, we use the "train" set for training and the "val" set for validation.

We first show in Table 1 the influence of each of the 3 proposed modifications individually or when combined together. The single most important improvement is the power normalization of the Fisher values. Combinations of two modifications generally improve over a single modification and the combination of all three modifications brings an additional increase. If we compare the baseline FK to the proposed modified FK, we observe an increase from 47.9% AP to 58.3% for SIFT descriptors. This corresponds to a +10.4% absolute improvement, a remarkable achievement on this dataset. To our knowledge, *these are the best results reported to date on PASCAL VOC 2007 using SIFT descriptors only.*

We now compare in Table 2 the results of our system with the best results reported in the literature on this dataset. Since most of these systems make use of color information, we report results with SIFT features only and with SIFT and color features[8]. The best system during the competition (by INRIA) [8] reported 59.4% AP using multiple channels and costly non-linear SVMs. Uijlings *et al.* [28] also report 59.4% but this is an optimistic figure which supposes the "oracle" knowledge of the object locations both in training and test images. The system of van Gemert *et al.* [3] uses many channels and soft-assignment. The system of Yang *et al.* [6] uses, again, many channels and a sophisticated Multiple Kernel Learning (MKL) algorithm. Finally, the best results we are aware of are those

[8] Our goal in doing so is not to advocate for the use of many channels but to show that, as is the case of the BOV, the FK can benefit from multiple channels and especially from color information.

Table 2. Comparison of the proposed Improved Fisher kernel (IFK) with the state-of-the-art on PASCAL VOC 2007. Please see the text for details about each of the systems.

Method	AP (in %)
Standard FK (SIFT) [22]	47.9
Best of VOC07 [8]	59.4
Context (SIFT) [28]	59.4
Kernel Codebook [3]	60.5
MKL [6]	62.2
Cls + Loc [29]	63.5
IFK (SIFT)	58.3
IFK (SIFT+Color)	60.3

of Harzallah *et al.* [29]. This system combines the winning INRIA classification system and a costly sliding-window-based object localization system.

4.3 CalTech 256

We now report results on the challenging CalTech 256 dataset. It consists of approx. 30K images of 256 categories. As is standard practice we run experiments with different numbers of training images per category: $ntrain = 15, 30, 45$ and 60. The remaining images are used for evaluation. To tune the SVM regularization parameters, we train the system with $(ntrain - 5)$ images and validate the results on the last 5 images. We repeat each experiment 5 times with different training and test splits. We report the average classification accuracy as well as the standard deviation (between parentheses). Since most of the results reported in the literature on CalTech 256 rely on SIFT features only, we also report results only with SIFT.

We do not provide a break-down of the improvements as was the case for PASCAL VOC 2007 but report directly in Table 3 the results of the standard FK of [22] and of the proposed improved FK. Again, we observe a very significant improvement of the classification accuracy using the proposed modifications. We also compare our results with those of the best systems reported in the literature. Our system outperforms significantly the kernel codebook approach of van Gemert *et al.* [3], the EMK of [20], the sparse coding of [18] and the system proposed by the authors of the CalTech 256 dataset [26]. We also significantly outperform the Nearest Neighbor (NN) approach of [19] when only SIFT features are employed (but [19] outperforms our SIFT only results with 5 descriptors). Again, to our knowledge, *these are the best results reported on CalTech 256 using only SIFT features.*

5 Large-Scale Experiments: ImageNet and Flickr Groups

Now equipped with our improved Fisher vector, we can explore image categorization on a larger scale. As an application we compare two abundant resources

Table 3. Comparison of the proposed Improved Fisher Kernel (IFK) with the state-of-the-art on CalTech 256. Please see the text for details about each of the systems.

Method	ntrain=15	ntrain=30	ntrain=45	ntrain=60
Kernel Codebook [3]	-	27.2 (0.4)	-	-
EMK (SIFT) [20]	23.2 (0.6)	30.5 (0.4)	34.4 (0.4)	37.6 (0.5)
Standard FK (SIFT) [22]	25.6 (0.6)	29.0 (0.5)	34.9 (0.2)	38.5 (0.5)
Sparse Coding (SIFT) [18]	27.7 (0.5)	34.0 (0.4)	37.5 (0.6)	40.1 (0.9)
Baseline (SIFT) [26]	-	34.1 (0.2)	-	-
NN (SIFT) [19]	-	38.0 (-)	-	-
NN [19]	-	42.7 (-)	-	-
IFK (SIFT)	34.7 (0.2)	40.8 (0.1)	45.0 (0.2)	47.9 (0.4)

of labeled images to learn image classifiers: ImageNet and Flickr groups. We replicated the protocol of [16] and tried to create two training sets (one with ImageNet images, one with Flickr group images) with the same 20 classes as PASCAL VOC 2007. To make the comparison as fair as possible we used as test data the VOC 2007 "test" set. This is in line with the PASCAL VOC "competition 2" challenge which consists in training on any "non-test" data [8].

ImageNet [23] contains (as of today) approx. 10M images of 15K concepts. This dataset was collected by gathering photos from image search engines and photo-sharing websites and then manually correcting the labels using the Amazon Mechanical Turk (AMT). For each of the 20 VOC classes we looked for the corresponding synset in ImageNet. We did not find synsets for 2 classes: person and potted plant. We downloaded images from the remaining 18 synsets as well as their children synsets but limited the number of training images per class to 25K. We obtained a total of 270K images.

Flickr groups have been employed in the computer vision literature to build text features [30] and concept-based features [14]. Yet, to our knowledge, they have never been used to train image classifiers. For each of the 20 VOC classes we looked for a corresponding Flickr group with a large number of images. We did not find satisfying groups for two classes: sofa and tv. Again, we collected up to 25K images per category and obtained approx. 350K images.

We underline that a perfectly fair comparison of Imagenet and Flickr groups is impossible (*e.g.* we have the same maximum number of images per class but a different total number of images). Our goal is just to give a rough idea of the accuracy which can be expected when training classifiers on these resources. The results are provided in Table 4. The system trained on Flickr groups yields the best results on 12 out of 20 categories (boldfaced) which shows that Flickr groups are a great resource of labeled training images although they were not intended for this purpose.

We also provide in Table 4 results for various combinations of classifiers learned on these resources. To combine classifiers, we use late fusion and

Table 4. Comparison of different training resources: I = ImageNet, F = Flickr groups, V = VOC 2007 trainval. A+B denotes the late fusion of the classifiers learned on resources A and B. The test data is the PASCAL VOC 2007 "test" set. For these experiments, we used SIFT features only. See the text for details.

Train	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow
I	**81.0**	66.4	**60.4**	71.4	**24.5**	67.3	**74.7**	62.9	36.2	36.5
F	80.2	**72.7**	55.3	**76.7**	20.6	**70.0**	73.8	**64.6**	**44.0**	**49.7**
V	75.7	64.8	52.8	70.6	30.0	64.1	77.5	55.5	55.6	41.8
I+F	81.6	71.4	59.1	75.3	24.8	69.6	75.5	65.9	43.1	48.7
V+I	82.1	70.0	62.5	74.4	28.8	68.6	78.5	64.5	53.7	47.4
V+F	82.3	73.5	59.5	78.0	26.5	70.6	78.5	65.1	56.6	53.0
V+I+F	82.5	72.3	61.0	76.5	28.5	70.4	77.8	66.3	54.8	53.0
[29]	77.2	69.3	56.2	66.6	45.5	68.1	83.4	53.6	58.3	51.1

Train	table	dog	horse	moto	person	plant	sheep	sofa	train	tv	mean
I	**52.9**	43.0	**70.4**	61.1	-	-	51.7	**58.6**	76.4	**40.6**	-
F	31.8	**47.7**	56.2	**69.5**	**73.6**	**29.1**	**60.0**	-	**82.1**	-	-
V	56.3	41.7	76.3	64.4	82.7	28.3	39.7	56.6	79.7	51.5	58.3
I+F	52.4	47.9	68.1	69.6	73.6	29.1	58.9	58.6	82.2	40.6	59.8
V+I	60.0	49.0	77.3	68.3	82.7	28.3	54.6	64.3	81.5	53.1	62.5
V+F	57.4	52.9	75.0	70.9	82.8	32.7	58.4	56.6	83.9	51.5	63.3
V+I+F	59.2	51.0	74.7	70.2	82.8	32.7	58.9	64.3	83.1	53.1	63.6
[29]	62.2	45.2	78.4	69.7	86.1	52.4	54.4	54.3	75.8	62.1	63.5

assign equal weights to all classifiers[9]. Since we are not aware of any result in the literature for the "competition 2", we provide as a point of comparison the results of the system trained on VOC 2007 (c.f. section 4.2) as well as those of [29]. One conclusion is that there is a great complementarity between systems trained on the carefully annotated VOC 2007 dataset and systems trained on more casually annotated datasets such as Flickr groups. Combining the systems trained on VOC 2007 and Flickr groups (V+F), we achieve 63.3% AP, an accuracy comparable to the 63.5% reported in [29]. Interestingly, we followed a different path from [29] to reach these results: while [29] relies on a more complex system, we rely on more data. In this sense, our conclusions meet those of Torralba *et al.* [31]: large training sets can make a significant difference. However, while NN-based approaches (as used in [31]) are difficult to scale to a very large number of training samples, the proposed approach leverages such large resources efficiently.

Let us indeed consider the computational cost of our approach. We focus on the system trained on the largest resource: the 350K Flickr group images. All the times we report were estimated using a single CPU of a 2.5GHz Xeon machine with 32GB of RAM. Extracting and projecting the SIFT features for the 350K training images takes approx. 15h (150ms / image), learning the GMM on a random subset of 1M descriptors approx. 30 min, computing the Fisher vectors

[9] We do not claim this is the optimal approach to combine multiple learning sources. This is just one reasonable way to do so.

approx. 4h (40ms / image) and learning the 18 classifiers approx. 2h (7 min / class). Classifying a new image takes 150ms+40ms=190ms for the signature extraction plus 0.2ms / class for the linear classification. Hence, the whole system can be trained and evaluated in less than a day on a single CPU. As a comparison, the system of [29] relies on a costly sliding-window object detection system which requires on the order of 1.5 days of training / class (using only the VOC 2007 data) and several minutes / class to classify an image[10].

6 Conclusion

In this work, we proposed several well-motivated modifications over the FK framework and showed that they could boost the accuracy of image classifiers. On both PASCAL VOC 2007 and CalTech 256 we reported state-of-the-art results *using only SIFT features and costless linear classifiers*. This makes our system scalable to large quantities of training images. Hence, the proposed improved Fisher vector has the potential to become a new standard representation in image classification.

We also compared two large-scale resources of training material – ImageNet and Flickr groups – and showed that Flickr groups are a great source of training material although they were not intended for this purpose. Moreover, we showed that there is a complementarity between classifiers learned on one hand on large casually annotated resources and on the other hand on small carefully labeled training sets. We hope that these results will encourage other researchers to participate in the "competition 2" of PASCAL VOC, a very interesting and challenging task which has received too little attention in our opinion.

References

1. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV SLCV Workshop (2004)
2. Farquhar, J., Szedmak, S., Meng, H., Shawe-Taylor, J.: Improving "bag-of-keypoints" image categorisation. Technical report, University of Southampton (2005)
3. Gemert, J.V., Veenman, C., Smeulders, A., Geusebroek, J.: Visual word ambiguity. In: IEEE PAMI (2010) (accepted)
4. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
5. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. IJCV 73 (2007)
6. Yang, J., Li, Y., Tian, Y., Duan, L., Gao, W.: Group sensitive multiple kernel learning for object categorization. In: ICCV (2009)

[10] These numbers were obtained through a personal correspondence with the first author of [29].

7. Tahir, M., Kittler, J., Mikolajczyk, K., Yan, F., van de Sande, K., Gevers, T.: Visual category recognition using spectral regression and kernel discriminant analysis. In: ICCV workshop on subspace methods (2009)
8. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results (2007)
9. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2008 (VOC 2008) Results (2008)
10. Everingham, M., Gool, L.V., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2009 (VOC 2009) Results (2009)
11. Joachims, T.: Training linear svms in linear time. In: KDD (2006)
12. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal estimate sub-gradient solver for SVM. In: ICML (2007)
13. Li, Y., Crandall, D., Huttenlocher, D.: Landmark classification in large-scale image collections. In: ICCV (2009)
14. Wang, G., Hoiem, D., Forsyth, D.: Learning image similarity from flickr groups using stochastic intersection kernel machines. In: ICCV (2009)
15. Maji, S., Berg, A.: Max-margin additive classifiers for detection. In: ICCV (2009)
16. Perronnin, F., Sánchez, J., Liu, Y.: Large-scale image categorization with explicit data embedding. In: CVPR (2010)
17. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. In: CVPR (2010)
18. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)
19. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
20. Bo, L., Sminchisescu, C.: Efficient match kernels between sets of features for visual recognition. In: NIPS (2009)
21. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: NIPS (1999)
22. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: CVPR (2007)
23. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
24. Zhang, X., Li, Z., Zhang, L., Ma, W., Shum, H.-Y.: Efficient indexing for large-scale visual search. In: ICCV (2009)
25. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
26. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
27. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60 (2004)
28. Uijlings, J., Smeulders, A., Scha, R.: What is the spatial extent of an object? In: CVPR (2009)
29. Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. In: ICCV (2009)
30. Hoiem, D., Wang, G., Forsyth, D.: Building text features for object image classification. In: CVPR (2009)
31. Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. In: IEEE PAMI (2008)

Max-Margin Dictionary Learning for Multiclass Image Categorization

Xiao-Chen Lian[1], Zhiwei Li[3], Bao-Liang Lu[1,2], and Lei Zhang[3]

[1] Dept. of Computer Science and Engineering, Shanghai Jiao Tong University, China
[2] MOE-MS Key Lab for BCMI, Shanghai Jiao Tong University, China
[3] Microsoft Research Asia
lianxiaochen@gmail.com, zli@microsoft.com,
bllu@sjtu.edu.cn, leizhang@microsoft.com

Abstract. Visual dictionary learning and base (binary) classifier training are two basic problems for the recently most popular image categorization framework, which is based on the bag-of-visual-terms (BOV) models and multiclass SVM classifiers. In this paper, we study new algorithms to improve performance of this framework from these two aspects. Typically SVM classifiers are trained with dictionaries fixed, and as a result the traditional loss function can only be minimized with respect to hyperplane parameters (w and b). We propose a novel loss function for a binary classifier, which links the hinge-loss term with dictionary learning. By doing so, we can further optimize the loss function with respect to the dictionary parameters. Thus, this framework is able to further increase margins of binary classifiers, and consequently decrease the error bound of the aggregated classifier. On two benchmark dataset, Graz [1] and the fifteen scene category dataset [2], our experiment results significantly outperformed state-of-the-art works.

Keywords: Bag of visual words, Dictionary learning, Max margin.

1 Introduction

Visual recognition is one of the fundamental challenges in computer vision, which targets at automatically assigning class labels to images based on their visual features. In recent years, many methods have been proposed [2,3,4,5], in which the framework that combines bag of visual words (BOV) model with SVM-based multiclass classifiers [3,4] has achieved state-of-the-art performance in various benchmark tasks [2,6,7]. To further improve the performance of this framework, we study two basic problems of it in this paper.

First, how to learn a better BOV model? A core issue of this framework is generating a dictionary that will be effective for classifier training. Most of existing approaches adopt unsupervised clustering manners, whose goals are to keep sufficient information for representing the original features by minimizing a reconstruction error or expected distortion (e.g. K-means [8], manifold learning [9] and sparse coding [4]). Due to the ignorance to supervisory information,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 157–170, 2010.

the histogram representations of images over the learned dictionary may not be optimal for a classification task. Therefore, a highly probably better choice is to incorporate discriminative information (i.e. class labels) into the dictionary construction process.

Second, how to train a better SVM classifier? SVM-based multiclass classifiers are usually constructed by aggregating results of a collection of binary classifiers. The most popular strategies are *one-vs.-one* where all pairs of classes are compared, and *one-vs.-all* where each class is compared against all others. The performance of the binary classifiers directly affects the performance of the aggregated classifier. Thus, a straightforward idea to improve the multiclass classifiers is improving the individual binary classifier.

Existing approaches typically deal with the above two problems separately: dictionaries are first generated and classifiers are then learned based on them. In this paper, we propose a novel framework for image classification which unifies the dictionary learning process with classifier training. The framework reduces the multiclass problem to a collection of one-vs-one binary problems. For each binary problem, classifier learning and dictionary generation are conducted iteratively by minimizing a unified objective function which adopts the maximum margin criteria. We name this approach Max-Margin Dictionary Learning (MMDL). We evaluate MMDL using two widely used classifier aggregation strategies: majority voting and Decision Directed Acyclic Graph (DDAG) [10]. Experimental results show that by embedding the dictionary learning into classifier training, the performance of the aggregated multiclass classifier is improved. Our results outperformed state-of-the-art results on Graz [1] and the fifteen scene category dataset [2].

2 Related Work

Supervised dictionary learning has attracted much attention in recent years. Existing approaches can be roughly categorized into three categories.

First, constructing multiple dictionaries, e.g. [11] wraps dictionary construction inside a boosting procedure and learns multiple dictionaries with complementary discriminative power, and[12] learns a category-specific dictionary for each category.

Second, learning a dictionary by manipulating an initial dictionary, e.g. merging visual words. The merging process could be guided by mutual information between visual words and classes [1], or trade-off between intra-class compactness and inter-class discrimination power [13]. The performance of such approaches is highly affected by the initial dictionary since only merging operation is considered in them. To ease this problem a large dictionary is required at the beginning to preserve as much discriminative abilities as possible, which is not guaranteed though.

Third, learning a dictionary via pursuing a descriptor-level discriminative ability, e.g. empirical information loss minimization method [14], randomized decision forests [15,16], and sparse coding-based approaches [17,18,19]. Most of these approaches are first motivated from coding of signals, where a sample (or

say signal) is only analogous to a local descriptor in an image rather than a whole image which is composed of a collection of local descriptors. Actually, this requirement is over strong since local descriptors of different objects are often overlapped (i.e. a white patch may appear both in the sky and on a wall).

Moreover, depending on whether dictionary learning and classifier training are unified in a single process or not, the above approaches can be further categorized to two categories. Most of them take two separate processes, e.g. [11,15,16,12,1,13,14], in which a dictionary is first learned and then a classifier is trained over it. Therefore, the objectives of the two processes are likely to be inconsistent. The other category of approaches takes a similar strategy as ours, that is, they combine the two processes by designing a hybrid generative and discriminative energy function. The discrimination criteria used include softmax discriminative cost functions [17,18] and Fisher's discrimination criterion [19]. However, existing approaches put the discrimination criteria on individual local descriptors rather than image-level representations, i.e. histogram representations of images.

After this paper was submitted, two additional related works were published, which also consider learning dictionary with image-level discriminative criteria. Yang *et al.* [20] used sparse coding for dictionary learning and put a classification loss in the model. Boureau *et al.* [21] used regularized logistic cost.

3 Max-Margin Dictionary Learning

In this section, we first introduce the motivation of incorporating max-margin criteria into dictionary learning process. Then the Max-Margin Dictionary Learning (MMDL) algorithm is described and the analysis on how max-margin criterion affects the dictionary construction is given. Finally, we describe the pipeline of the whole classification framework.

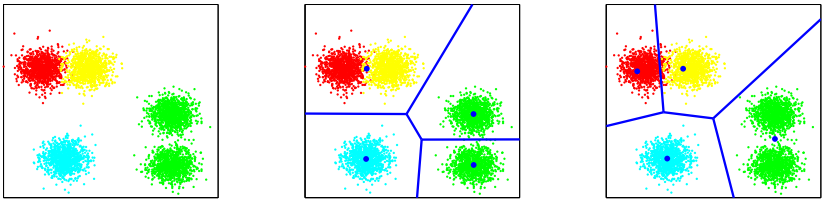
3.1 Problem Formulation

Suppose we are given a corpus of training images $\mathcal{D} = \{(I^d, c^d)\}_{d=1}^{D}$, where $I^d = \{x_1^d, x_2^d \ldots, x_{N_d}^d\}$ is the set of local descriptors (i.e. SIFT [22]) extracted from image d, and $c^d \in \{+1, -1\}$ is the class label associated with I^d. A dictionary learning method will construct a dictionary which consists of K visual words $V = \{v_1, v_2, \ldots, v_K\}$. A descriptor x_i^d from image d is quantized to a K-dimension vector ϕ_i^d where

$$\phi_i^d[k] = \begin{cases} 1, & k = \underset{w}{\operatorname{argmin}} \parallel x_i^d - v_w \parallel_2 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

for hard assignment and

$$\phi_i^d[k] = \frac{\exp(-\gamma \parallel x_i^d - v_k \parallel_2^2)}{\sum_{k'=1}^{K} \exp(-\gamma \parallel x_i^d - v_{k'} \parallel_2^2)}, \quad k = 1, \ldots, K \tag{2}$$



(a) (b) (c)

Fig. 1. (a) The 2-D synthetic data. Colors of points indicate their categories. (b) K-means results and Voronoi boundary of the words. Red and yellow points are grouped onto the same word and as a result cannot be distinguished. (c) A correct partition where categories information are completely preserved at the price of distortion. This figure should best be viewed in color

for soft assignment [23]. Then image d is represented by the histogram

$$\Phi^d = \frac{1}{N_d} \sum_{i=1}^{N_d} \phi_i^d, \tag{3}$$

and the set of couples $\{(\Phi^d, c^d)\}$ are used to train the classifiers.

Traditionally, the dictionary V is learned by minimizing the reconstruction error or overall distortion. For example, K-means clustering algorithm solves the following problem

$$\min_V \sum_{d=1}^{D} \sum_{i=1}^{N_d} \min_{k=1\ldots K} \parallel x_i^d - v_k \parallel_2^2 \tag{4}$$

However, as the learning process does not utilize the category information, the resulted histograms may not be optimal for classification. We illustrate the problem on a toy data shown in Figure 1(a). K-means groups the red and yellow clusters into one word (Figure 1(b)) and separates the two green clusters to two words because it only considers to minimizing the overall distortion. As a result, red and yellow clusters cannot be distinguished through their histogram representations. A correct partition is shown in Figure 1(c); although the dictionary has a bigger distortion, it is more discriminative than the dictionary obtained by K-means.

3.2 MMDL

Our solution to the above problem is to combine classifier training and dictionary learning together. Motivated by the loss function in SVM, we design the following objective function

$$\mathcal{L}(V, W) = \frac{1}{2} \parallel W \parallel_2^2 + C \sum_d \max\left(0, 1 - c_d \langle W, \Phi^d \rangle\right) \tag{5}$$

Algorithm 1. MMDL

Input: A training set $\mathcal{D} = \{(I^d, c^d)\}_{d=1}^{D}$; number of iteration T; convergence threshold ϵ

Initialize the dictionary V
repeat{Alternate dictionary and classifier learning}
  Fix V, $W = \underset{W'}{\operatorname{argmin}}\, \mathcal{L}(V, W')$
  $V_0 = V$; $\mathcal{L}_{\min} = \mathcal{L}(V_0, W)$; $V_{\min} = V_0$
  for $t = 1$ **to** T **do**
    $V_t = V_{t-1} - \lambda_t \nabla \mathcal{L}(V_t, W)$ (∇ denotes subgradient operators)
    if $\mathcal{L}(V_t, W) < \mathcal{L}_{\min}$
      $\mathcal{L}_{\min} = \mathcal{L}(V_t, W)$
      $V_{\min} = V_t$
    end if
  end for
  $V = V_{\min}$
until convergence rate of $\mathcal{L}(V, W)$ is below ϵ
return dictionary V and classifier W

where Φ^d is computed through Eq. (2) and (3), $W = (w_1, \ldots, w_K)^\top$ is a hyperplane classifier and C is the trade-off factor. It is noted that:

1) We omit the offset (i.e. b in a standard SVM classifier) since the L1-norm of Φ^d is always one.

2) In terms of learning a dictionary, the objective of Eq. 5 is different from Eq. 4. Eq. 4 minimizes the distortion of a dictionary, while Eq. 5 aims at finding a dictionary which minimizes a SVM loss.

3) For computational reason, we only support a linear SVM classifier in this framework. The main reason is that using a non-linear kernel in Eq. 5 makes learning difficult since the analytical forms of project functions are usually unknown and computing their derivatives is intractable. However, as later shown in experiments, using the dictionary learned by MMDL, linear SVM classifier outperforms the non-linear SVM classifiers that use dictionary learned by K-means.

By minimizing $\mathcal{L}(V, W)$, we obtain a dictionary V and a binary classifier W which are expected to be with a large margin. The minimization is proceeded as a two-phase iteration. In the first phase, the dictionary V is fixed, and the computation of W becomes a standard linear SVM problem, in which the first term punishing the model complexity and the hinge loss term punishing the training error. In the second phase, V is computed by fixing W. Eq. 5 links the dictionary learning and classifier training processes. In traditional BOV+SVM framework where the two processes are separated, the optimization of Eq. 5 involves only the first phase. While in MMDL, we can further minimize Eq. 5 by doing the second phase, and the margins are further increased. Due to the presence of both the non-linearity of Φ^d and the non-differentiability of the hinge loss, we apply subgradient method [24] which is widely used with non-differentiable objective functions.

The iteration used in subgradient method is quite similar to that of steepest descent, except the following two differences (refer to [24] for details): (1) As the

objective function may not have derivatives at all points, the search direction is the negative of the subgradient; (2) It may happen that the search direction is not a descent direction, therefore a list recording the lowest objective function value found so far is maintained.

Algorithm 1 depicts the complete MMDL algorithm. In line 3, W is computed by a standard SVM solver. In line 7, the dictionary V is updated according to the subgradient and the step size λ_t. The subgradient of a convex function f at point x_0 is a nonempty closed interval $[f^-(x_0), f^+(x_0)]$ where $f^-(x_0)$ and $f^+(x_0)$ are the left- and right-sided derivatives respectively. The interval reduces to a point when f is differentiable at x_0. In this case, $f^-(x_0) = f^+(x_0) = \partial f(x_0)$.

Denote $\langle W^\top, \Phi^d \rangle$ by $h^d(V)$, then the hinge loss term for image d is $\mathcal{L}^d = \max(0, 1 - c^d h^d(V))$. When $c^d h^d(V) < 1$, $\mathcal{L}^d = 1 - c^d h^d(V)$ is differentiable. Its subgradient at $v_k (k = 1 \ldots, K)$ equals to its derivative

$$\begin{aligned}
\frac{\partial \mathcal{L}^d}{\partial v_k} &= \frac{\partial}{\partial v_k}(-\frac{c^d w_k}{N_d} \sum_{i=1}^{N_d} \phi_i^d[k]) \\
&= -\frac{c^d w_k}{N_d} \sum_{i=1}^{N_d} \frac{2\gamma(x_i^d - v_k) \exp(-\gamma \parallel x_i^d - v_k \parallel_2^2)}{\left(\sum_{k'=1}^{K} \exp(-\gamma \parallel x_i^d - v_{k'} \parallel_2^2)\right)} \\
&\quad + \frac{c^d w_k}{N_d} \sum_{i=1}^{N_d} \frac{2\gamma(x_i^d - v_k) \exp(-\gamma \parallel x_i^d - v_k \parallel_2^2)^2}{\left(\sum_{k'=1}^{K} \exp(-\gamma \parallel x_i^d - v_{k'} \parallel_2^2)\right)^2} \\
&= -\frac{2c^d w_k}{N_d} \sum_{i=1}^{N_d} \gamma(x_i^d - v_k) \left(\phi_i^d[k] - (\phi_i^d[k])^2\right).
\end{aligned} \tag{6}$$

When $c^d h^d(V) \geq 1$, the subgradient $\nabla \mathcal{L}^d = 0$ for all $v_k (k = 1 \ldots, K)$, which means we pick the right-sided derivative of the hinge loss term. The visual word v_k is then updated by

$$v_k^{t+1} = v_k^t - \lambda_t \sum_{d \in \mathcal{X}} \frac{\partial \mathcal{L}^d}{\partial v_k^t} \tag{7}$$

where $\mathcal{X} = \{d \mid c^d h^d(V) < 1\}$ is the set of indices of the images that lie in the margin or are misclassified by W. We name these images as *effective images* because only these images are involved in the dictionary update equation.

Analysis. We examine the update rules to see how the *effective images* take effect in the dictionary learning process. For better understanding, we reformat Eq. (7) as:

$$v_k' = v_k + \sum_{d \in \mathcal{X}} \frac{2\gamma\lambda}{N_d} \sum_{i=1}^{N_d} s_i^d[k] \cdot t_i^d[k] \cdot p_i^d[k], \tag{8}$$

where

$$
\begin{aligned}
s_i^d[k] &= \mathrm{sign}(\mathrm{c^d w_k}) \\
p_i^d[k] &= \frac{x_i^d - v_k}{\parallel x_i^d - v_k \parallel_2^2} \\
t_i^d[k] &= w_k \left(\phi_i^d[k] - (\phi_i^d[k])^2\right) \parallel x_i^d - v_k \parallel_2^2 .
\end{aligned} \tag{9}
$$

Intuitively, the update of v_k is the net force of all local descriptors in effective images. Each descriptor x_i^d pushes or pulls v_k along a direction. The direction of the force is determined by $s_i^d[k]$ and $p_i^d[k]$. If $s_i^d[k] > 0$, it means that the k-th word is positive to correctly predicting the label of image d (i.e. the image favors larger $\phi_i^d[k]$); otherwise, it means that we expect that the image d should have smaller $\phi_i^d[k]$. As a result, when $s_i^d[k] > 0$, v_k will be pulled to be near to descriptor x_i^d, and when $s_i^d[k] < 0$, it will be pushed to be far away from x_i^d. Therefore moving v_k according to Eq. (8) will decreases the hinge loss $\mathcal{L}(V, W)$. The strength of x_i^d's force on v_k is determined by $t_i^d[k]$, which is proportional to w_k, a quadratic term $\phi_i^d[k] - (\phi_i^d[k])^2$ and $\parallel x_i^d - v_k \parallel_2^2$ (Euclidean distance between x_i^d and v_k). In the following, we give an intuitive explanation about $t_i^d[k]$.

From the feature selection's point of view, hyperplane W plays a role as visual word selector. If the absolute value of w_k is very small, it means the word is not important for the classifier, and thus the update to the corresponding v_k could be minor.

Before analyzing the latter two terms, we first note that $\phi_i^d[k]$ and $e_i^d[k] = \parallel x_i^d - v_k \parallel_2^2$ are both related to the distance between descriptor x_i^d and visual word v_k. The former one measures the relative distance from v_k to $x_i^d[k]$ compared with other visual words, while the latter is the absolute distance. We first consider the case when the force x_i^d exerts on v_k is pull. When $\phi_i^d[k]$ is very large, moving v_k for a distance may not increase the distortion too much. Therefore the quadratic term $\phi_i^d[k] - (\phi_i^d[k])^2$ will be small, indicating that x_i^d does not hold v_k strongly and allows other descriptors to move it. If $\phi_i^d[k]$ is quite small, v_k is relative far from x_i^d, and the quadratic term will also be small which means the force should be small as moving v_k close to x_i^d may cause large distortion. If e_i^d is large but other visual words are much far away from x_i^d, moving v_k close is acceptable. Otherwise x_i^d may not pull v_k over as the distortion may increase. Similar discussion can be made when the force is push.

3.3 Base Classifier and Aggregation Strategy

The hyperplane classifier W obtained during dictionary learning can be used as the base classifier. Although it is a linear classifier, in our experiments it outperforms the SVM classifiers with non-linear kernels which are trained based on unsupervisedly learned dictionaries.

Any strategy that aggregates binary classifiers can be used in our framework, e.g. majority voting (VOTE), DDAG and Error-Correcting Codes (ECC) [25]. In this paper we evaluate the combination of MMDL with VOTE and DDAG.

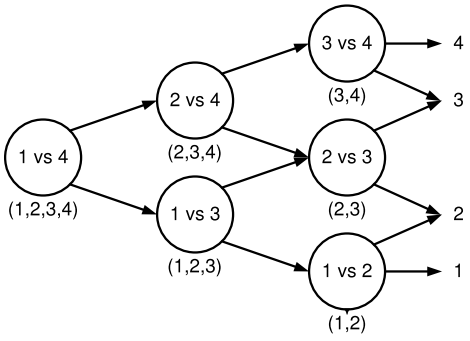


Fig. 2. A four-class DDAG. The list under each node contains the class labels that will remain when the evaluation reaches the node.

We use DDAG as an example to demonstrate how we aggregate MMDL base classifiers.

DDAG is a lightweight architecture which is efficient to evaluate. Besides, the theoretic advantage of DDAG is that when the base classifiers are hyperplanes, enlarging the margin of all nodes in a DDAG will lower the bound on the generalization error. For a C-class problem, DDAG has $C(C-1)/2$ nodes, each one distinguishing a pair of classes a and b. On each such node, MMDL learns a dictionary and a corresponding classifier using the subset of images labeled by a or b. The evaluation of a DDAG $\mathcal{G}$ on a test point is equivalent to operating on a list which initially contains all classes. The point x is first test by the node that corresponds to the first and last classes on the list and one of the class is eliminated from the list if the node prefers the other one. DDAG then proceeds to test the first and last classes on the new list. The process terminates when only one class remains in the list and x is judged to be that class (see Fig. 2).

3.4 Time Complexity

Let C be the number of categories, K be the size of each two-class dictionary, and L be the dimension of descriptor. Suppose the number of descriptors from each categories is N. The time complexity for learning all two-class dictionaries is $O(C \times (C-1) \times N \times K \times L \times T_s \times T_i)$, where T_s and T_i are the number of iterations for subgradient and two-phase iteration respectively. It is comparable to the complexity of learning a same size dictionary by K-means, i.e. $O(C \times N \times \frac{C(C-1)}{2} \times K \times L \times T)$, where T is the number of iterations for K-means to converge.

4 Experiments

In this section, we report results on two benchmark datasets: Graz-02 [1] and fifteen scene dataset [2]. We use a variant setting of SIFT to generate local

Table 1. A comparison of the pixel precision-recall equal error rates on Graz-02 dataset. Dictionary size is 200

	cars	people	bicycles
AIB200-KNN [1]	50.90	49.70	**63.80**
AIB200-SVM [1]	40.10	50.70	59.90
MMDL+HP	**54.27**	**55.81**	63.55

descriptors. In our implementation, a patch is divided into 2×2 subpatches rather than the 4×4 schema in the standard SIFT setting [22]. For each subpatch a 8-bin histogram of oriented gradients is calculated. Thus, our local descriptor is 32-d. We adopt this setting mainly for its computational efficiency and Uijlings *et al.* [26] reported that 2×2 SIFT performed marginally better but never worse than the 4×4 SIFT.

In all experiments, we perform processing in gray scale, even when color images are available. We initialize the dictionary V by randomly selecting descriptors from training data and set the parameters of MMDL as $C = 32$, $\gamma = 1\times10^{-3}$ and $\lambda_t = 1\times10^{-1}$ for all $t = 1\ldots,T$. The number of iterations T for subgradient method is set to be 40, and MMDL converges after about 30 iterations under the convergence threshold $\epsilon = 1 \times 10^{-4}$.

4.1 Object Localization

We first use Graz-02 dataset [1] to evaluate the performance of MMDL for object localization. Graz-02 contains three classes (bicycles, cars and people) with extreme variability in pose, scale and lighting. The task is to label image pixel as either belonging to one of the three classes or background. The baseline approach is another supervised dictionary learning method proposed in [1]. The measure of performance is pixel precision-recall error rate. We follow the same setup as in [1]: for each object, a dictionary that distinguishes foreground objects from background is constructed; when testing, a histogram of frequencies of visual words within the 80×80-pixel window centered at each pixel is computed. A SVM classifier is applied to classify the histogram and a confidence that the pixel belongs to foreground object is returned. Precision and recall are computed according to ground-truth segmentation provided by the dataset. The results when the sizes of dictionaries are 200 are reported in Table 1. MMDL+HP means that we directly used the learned hyperplane classifiers obtained during the dictionary learning. The performance of our approach is significantly better than the baseline approach on the first two classes, and is comparable with [1] on the last class.

4.2 Scene Category Classification

The second dataset we use is the fifteen scene dataset (scene15), which consists of fifteen kinds of scene images, e.g. highway, kitchen and street. As in [2,14], SIFT descriptors of 16×16 patches sampled over a grid with spacing of 8 pixels are computed. 100 images per class are randomly selected for training and the

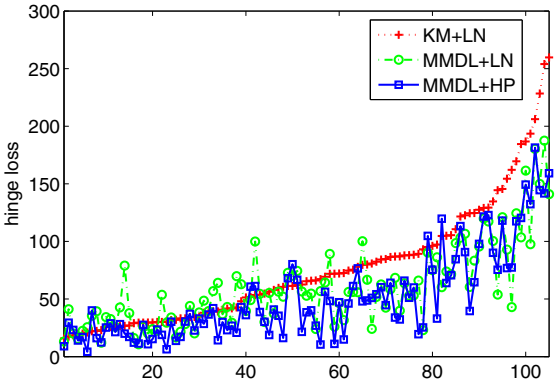


Fig. 3. Comparison of hinge losses on all binary problems obtained by MMDL and K-means on scene15. The hinge losses are computed on test data.

Table 2. Comparison of hinge losses for the top four most confused classes of scene15: *bedroom*, *kitchen*, *living room* and *industrial*

	KM+LN	MMDL+LN	MMDL+HP
bedroom vs. kitchen	137.25 ± 8.23	115.86 ± 3.56	$\mathbf{108.59 \pm 6.35}$
bedroom vs. living room	239.97 ± 9.55	206.62 ± 13.08	$\mathbf{189.93 \pm 32.36}$
bedroom vs. industrial	168.38 ± 2.87	$\mathbf{124.71 \pm 0.49}$	125.25 ± 5.89
kitchen vs. living room	193.47 ± 9.70	173.29 ± 14.23	$\mathbf{166.30 \pm 12.70}$
kitchen vs. industrial	133.34 ± 16.91	95.78 ± 7.56	$\mathbf{88.24 \pm 8.24}$
living room vs. industrial	222.74 ± 24.41	147.55 ± 33.33	$\mathbf{155.82 \pm 16.33}$

rest for testing. We train 105 binary classifiers, one for each pair of classes, with all possible combinations of dictionary learning algorithms and classifier settings. The dictionary learning algorithms are K-means (KM) and MMDL. The quantization of KM uses the soft assignment in Eq. 2 with the same γ as MMDL. Each binary problem use a dictionary with 50 visual words. The classifiers are SVM with linear kernel (LN) and histogram intersection kernel (HI), and the hyperplane-based classifier learned by MMDL (HP). For example, a name "KM+HI+DDAG" means that we adopt K-means to learn a dictionary, histogram intersection kernel to train SVM classifiers, and the DDAG approach to aggregate base classifiers. The experiments are repeated five times and the final result is reported as the mean and standard deviation of the results from the individual runs.

To show the superiority of MMDL over K-means, in Fig. 3 we plot the hinge losses of linear classifiers on all binary problems obtained by the K-means and MMDL. The x-coordinate is the indices of binary classifiers which are sorted in an order that their hinge loss produced by the corresponding KM+LIN method on test set are ascending. We also list the hinge losses of the top four most confused classes (bedroom, kitchen, living room and industrial) in Table 2. In

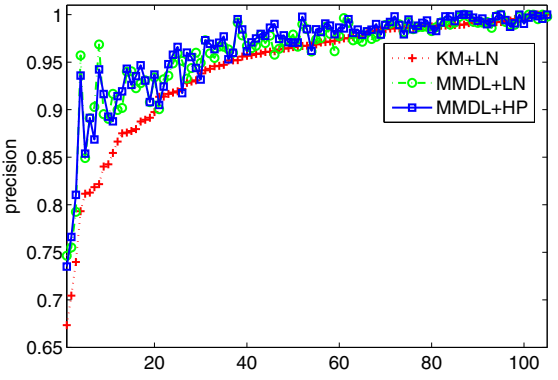


Fig. 4. Comparison of precisions on all binary problems obtained by MMDL and K-means on scene15

Table 3. Comparison of precisions (percentage) for the top four most confused classes of scene15: *bedroom*, *kitchen*, *living room* and *industrial*

	KM+LN	MMDL+LN	MMDL+HP
bedroom vs. kitchen	77.46 ± 2.38	80.04 ± 0.97	$\mathbf{82.69 \pm 1.11}$
bedroom vs. living room	66.70 ± 0.42	70.06 ± 3.26	$\mathbf{72.37 \pm 0.77}$
bedroom vs. industrial	81.96 ± 1.84	86.99 ± 1.43	$\mathbf{87.83 \pm 1.51}$
kitchen vs. living room	72.03 ± 1.43	75.86 ± 3.37	$\mathbf{78.53 \pm 2.26}$
kitchen vs. industrial	86.80 ± 1.92	86.96 ± 0.39	$\mathbf{89.60 \pm 1.41}$
living room vs. industrial	78.66 ± 2.73	$\mathbf{87.55 \pm 3.37}$	85.56 ± 0.94

Table 4. Comparison of precisions (percentage) for all classes of scene15

	KM+HI		MMDL+HP		MMDL+HI	
class	DDAG	VOTE	DDAG	VOTE	DDAG	VOTE
bedroom	34.5 ± 0.9	40.8 ± 1.8	47.1 ± 7.3	$\mathbf{58.0 \pm 5.2}$	46.0 ± 5.7	55.7 ± 5.7
suburb	88.2 ± 3.9	89.4 ± 2.5	91.7 ± 1.1	92.9 ± 2.6	92.9 ± 1.9	$\mathbf{93.9 \pm 1.8}$
kitchen	52.1 ± 1.4	57.0 ± 3.4	$\mathbf{71.2 \pm 0.5}$	69.4 ± 3.4	68.8 ± 6.7	69.1 ± 6.3
living room	49.7 ± 3.8	46.9 ± 4.2	53.3 ± 1.3	51.0 ± 5.0	$\mathbf{61.9 \pm 2.4}$	54.1 ± 3.8
coast	81.0 ± 5.3	82.6 ± 5.5	82.4 ± 1.2	84.6 ± 2.0	86.2 ± 2.8	$\mathbf{90.1 \pm 3.1}$
forest	90.2 ± 1.3	90.8 ± 1.6	$\mathbf{92.3 \pm 1.5}$	$\mathbf{92.3 \pm 1.5}$	90.6 ± 1.8	91.7 ± 1.2
highway	83.8 ± 3.8	84.4 ± 2.5	87.1 ± 1.8	87.1 ± 2.4	87.1 ± 2.5	$\mathbf{88.1 \pm 3.5}$
inside city	65.2 ± 3.9	66.8 ± 3.5	72.1 ± 3.8	72.8 ± 3.6	72.6 ± 1.7	$\mathbf{75.8 \pm 1.1}$
mountain	79.2 ± 1.8	78.5 ± 2.4	83.8 ± 1.3	82.1 ± 1.6	$\mathbf{84.4 \pm 1.1}$	82.5 ± 1.3
open country	68.4 ± 2.6	68.0 ± 2.9	71.5 ± 1.5	73.4 ± 3.2	$\mathbf{80.0 \pm 2.1}$	78.8 ± 2.3
street	84.0 ± 2.4	82.6 ± 2.9	$\mathbf{87.3 \pm 2.1}$	86.5 ± 1.4	86.1 ± 1.5	86.3 ± 2.1
tall building	82.9 ± 0.6	82.0 ± 0.7	77.9 ± 1.0	79.0 ± 5.6	$\mathbf{87.5 \pm 1.0}$	85.3 ± 0.2
office	77.4 ± 1.5	75.9 ± 2.0	82.9 ± 4.8	80.9 ± 3.0	$\mathbf{89.3 \pm 4.0}$	87.5 ± 5.8
store	64.0 ± 5.8	63.1 ± 6.2	68.2 ± 1.2	70.7 ± 3.3	$\mathbf{74.6 \pm 0.5}$	73.8 ± 0.5
industrial	42.3 ± 5.2	42.0 ± 2.8	42.2 ± 5.0	48.3 ± 4.8	55.5 ± 4.3	$\mathbf{58.5 \pm 5.2}$
average	69.5 ± 0.2	70.1 ± 0.1	74.1 ± 1.2	75.3 ± 2.1	77.6 ± 0.3	$\mathbf{78.1 \pm 0.7}$

Table 5. Comparison of average precisions (percentage) on secene15 dataset

	L = 2	L = 3
MMDL+SPM+HP+DDAG	78.34 ± 0.90	82.33 ± 0.39
MMDL+SPM+HP+VOTE	79.15 ± 0.76	83.21 ± 0.45
MMDL+SPM+HI+DDAG	82.23 ± 1.01	$\mathbf{85.98 \pm 0.68}$
MMDL+SPM+HI+VOTE	$\mathbf{82.66 \pm 0.51}$	$\mathbf{86.43 \pm 0.41}$
KM+SPM+HI+DDAG	77.48 ± 1.08	79.65 ± 0.59
KM+SPM+HI+VOTE	77.89 ± 0.50	80.17 ± 0.28
HG [27]	-	85.2
SPM [2]	80.1 ± 0.5	81.4 ± 0.5
ScSPM [4]	-	80.4 ± 0.9
sPACT [3]	-	83.3 ± 0.5

the similar way, we compare their precisions on all binary problems in Fig. 4 and Table 3. We can see that:

1) In terms of both the hinge loss and precision, MMDL based approach is significantly better than K-means based approaches.

2) For the four categories, which KM+LIN does not distinguish well (i.e. classification between the four classes), the improvements obtained by MMDL are significant. For all categories, MMDL outperforms K-means.

Table 4 shows the performance for each category with different dictionary and classifier settings. Our basic approaches, i.e. MMDL+HP+DDAG/VOTE, significantly outperform the baseline approaches (KM+HI+DDAG/VOTE), and with histogram intersection kernel, their performance is even better. With a 200 word universal dictionary, which is obtained by running K-means over SIFT descriptors of randomly sampled 400 images, the linear SVM achieved an average precision at 74.3%[1] which is also lower than our approaches. We also learned a 5250-word universal dictionary by K-means, whose size is equal to the total number of visual words used in MMDL approaches. Its result with histogram intersection kernel is 75.3%. An interesting observation is that without incorporating the max margin term into learning process, using a set of two-class dictionaries is worse than using a single dictionary with enough size. Two-class dictionaries are likely to over fit on training images, and their generalization capabilities are usually weak. While from table 4, we can see that MMDL can boost the performance, which is attributed to the incorporation of max margin criteria.

On scene15, the state-of-the-art results are obtained by applying spatial pyramid matching (SPM) mechanism [2]. We apply it to each binary classifier in our framework. Although our objective function of dictionary learning does not optimize for the SPM representation, our approach achieves the best results as shown in Table 5. To the best of our knowledge, it outperforms all results on

[1] The result is better than the result, $72.2 \pm 0.6\%$, reported in [2]

this dataset reported in recent years [3,4,2,27]. Actually, due to a characteristic of SPM mechanism (i.e. it is a "linear" transform indeed), it can be integrated in our loss function easily.

5 Conclusion

We have proposed a max-margin dictionary learning algorithm, which can be integrated in the training process of a linear SVM classifier to further increase the margin of the learned classifier, and consequently decrease the error bound of the aggregated multi-class classifier. Our preliminary experiment results on two benchmark datasets demonstrate the effectiveness of the proposed approach.

In the future, we are going to study how to directly apply non-linear kernel functions, e.g. histogram intersection kernel and χ^2 kernel, in the SVM classifier. Recently, using spatial information in image classification have drawn much attention. A common problem of these approaches is that the spatial constraints are predetermined and fixed during dictionary learning. We are designing a method that will automatically determine the spatial constraints under the guidance of supervised information.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (Grant No. 60773090 and Grant No. 90820018), the National Basic Research Program of China (Grant No. 2009CB320901), and the National High-Tech Research Program of China (Grant No. 2008AA02Z315).

References

1. Fulkerson, B., Vedaldi, A., Soatto, S.: Localizing objects with smart dictionaries. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 179–192. Springer, Heidelberg (2008)
2. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. CVPR, pp. 2169–2178 (2006)
3. Wu, J., Rehg, J.: Where am I: Place instance and category recognition using spatial PACT. In: Proc. CVPR (2008)
4. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Proc. CVPR (2009)
5. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proc. CVPR, vol. 2, pp. 524–531
6. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Comput. Vis. Image Underst. 106, 59–70 (2007)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, Results (2009), http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html

8. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. ICCV., vol. 2, pp. 1470–1477 (2003)
9. Jiang, Y.G., Ngo, C.W.: Visual word proximity and linguistics for semantic video indexing and near-duplicate retrieval. Comput. Vis. Image Underst. 113, 405–414 (2009)
10. Platt, J., Cristianini, N., Shawe-Taylor, J.: Large margin DAGs for multiclass classification. Advances in neural information processing systems 12, 547–553 (2000)
11. Zhang, W., Surve, A., Fern, X., Dietterich, T.: Learning non-redundant codebooks for classifying complex objects. In: Proceedings of the 26th Annual International Conference on Machine Learning (2009)
12. Perronnin, F.: Universal and adapted vocabularies for generic visual categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1243–1256 (2008)
13. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: Proc. ICCV, pp. 1800–1807 (2005)
14. Lazebnik, S., Raginsky, M.: Supervised learning of quantizer codebooks by information loss minimization. IEEE Trans. Pattern Anal. Mach. Intell. 31, 1294–1309 (2009)
15. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. Advances in neural information processing systems 19, 985 (2007)
16. Shotton, J., Johnson, J., Cipolla, M.: Semantic texton forests for image categorization and segmentation. In: Proc. CVPR (2008)
17. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Supervised dictionary learning. Advances in Neural Information Processing Systems 21 (2009)
18. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis. In: Proc. CVPR (2008)
19. Huang, K., Aviyente, S.: Sparse representation for signal classification. Advances in Neural Information Processing Systems 19, 609 (2007)
20. Yang, J., Yu, K., Huang, T.: Supervised Translation-Invariant Sparse Coding. In: Proc. CVPR (2010)
21. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning Mid-Level Features For Recognition. In: Proc. CVPR (2010)
22. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. ICCV., vol. 2, pp. 1150–1157 (1999)
23. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. CVPR (2008)
24. Shor, N., Kiwiel, K., Ruszcayǹski, A.: Minimization methods for non-differentiable functions. Springer, New York (1985)
25. Allwein, E., Schapire, R., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. The Journal of Machine Learning Research 1, 113–141 (2001)
26. Uijlings, J., Smeulders, A., Scha, R.: What is the Spatial Extent of an Object? In: Proc. CVPR (2009)
27. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.: Hierarchical Gaussianization for Image Classification. In: Proc. ICCV (2009)

Towards Optimal Naive Bayes Nearest Neighbor

Régis Behmo[1], Paul Marcombes[1,2], Arnak Dalalyan[2], and Véronique Prinet[1]

[1] NLPR / LIAMA, Institute of Automation, Chinese Academy of Sciences*
[2] IMAGINE, LIGM, Université Paris-Est

Abstract. Naive Bayes Nearest Neighbor (NBNN) is a feature-based image classifier that achieves impressive degree of accuracy [1] by exploiting 'Image-to-Class' distances and by avoiding quantization of local image descriptors. It is based on the hypothesis that each local descriptor is drawn from a class-dependent probability measure. The density of the latter is estimated by the non-parametric kernel estimator, which is further simplified under the assumption that the normalization factor is class-independent. While leading to significant simplification, the assumption underlying the original NBNN is too restrictive and considerably degrades its generalization ability. The goal of this paper is to address this issue.

As we relax the incriminated assumption we are faced with a parameter selection problem that we solve by hinge-loss minimization. We also show that our modified formulation naturally generalizes to optimal combinations of feature types. Experiments conducted on several datasets show that the gain over the original NBNN may attain up to 20 percentage points. We also take advantage of the linearity of optimal NBNN to perform classification by detection through efficient sub-window search [2], with yet another performance gain. As a result, our classifier outperforms — in terms of misclassification error — methods based on support vector machine and bags of quantized features on some datasets.

1 Introduction

With the advent in recent years of powerful blob and corner detectors and descriptors, the orderless bag of quantized features — also called bag of words (BoW) — has been the preferred image representation for image classification. The BoW owes its popularity to its relative simplicity and its ability to produce a compact, finite-dimensional representation that can be used as input of a state-of-the-art classifier such as support vector machine (SVM) or Adaboost. One can cite several highly competitive approaches that are essentially based on the BoW/SVM combination [3,4,5,6]. In this paper, we propose an alternative to mainstream methods based on parameter-optimized version of the NBNN.

In BoW representations, the quantization step results in a substantial loss of discriminative power of the visual features [6,1]. This loss was quantitatively measured in [1] and it is argued that the popularity enjoyed by the BoW/SVM combination is due to the efficiency of the SVM classifier, not to the representation itself. In simple words, most, but not all, of the information discarded by the feature quantization step is offset

* The first author is supported by a INRIA-Cordi grant. This work was partially supported by the Chinese Ministry of Science and Technology.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 171–184, 2010.

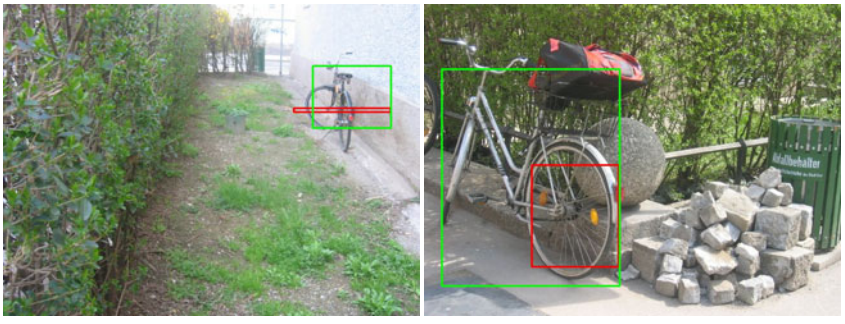


Fig. 1. Subwindow detection for the original NBNN (red) and for our version of NBNN (green). Since the background class is more densely sampled than the object class, the original NBNN tends to select an object window that is too small relatively to the object instance. As show these examples, our approach addresses this issue.

by the efficiency of the classifier. Naive Bayes Nearest Neighbor (NBNN) is a classifier introduced in [1] that was designed to address this issue: NBNN is non-parametric, does not require any feature quantization step and thus uses to advantage the full discriminative power of visual features. However, in practice, we observe that NBNN performs relatively well on certain datasets, but not on others. To remedy this, we start by analyzing the theoretical foundations of the NBNN. We show that this performance variability could stem from the assumption that the normalization factor involved in the kernel estimator of the conditional density of features is class-independent. We relax this assumption and provide a new formulation of the NBNN which is richer than the original one. In particular, our approach is well suited for optimal, multi-channel image classification and object detection. The main argument of NBNN is that the log-likelihood of a visual feature can be approximated by the distance to its nearest neighbor. In our formulation, this log-likelihood is approximately equal to an affine function of the nearest neighbor distance. The latter involves two affine coefficients that, in general, depend on properties of the training feature set. Our first contribution consists in a method to optimize these parameters by solving a linear problem that minimizes the cross-validated hinge loss. In addition, this new formulation generalizes well to optimal combinations of features of different types, here referred to as channels. The distance correction parameters also serve to balance each feature channel according to its relative relevance, as not all feature channels are equally useful to the problem at hand. As our last contribution, we show how to reformulate our classifier to perform object detection and classification by detection. In classification by detection (cf. [7] and the references therein), the aim is to classify images that contain an object embedded in a strongly cluttered background. Our solution consists in finding the image subwindow that maximizes a function that is linear in the image features. Due to this linearity, the optimal object location can be found by branch and bound subwindow search [2].

We conducted some experiments that reveal that affine distance correction improves NBNN performance by up to 20 percentage points. This indicates that our modified formulation is not merely a theoretical improvement, but is also of practical interest. Moreover, this gain is obtained with little computational overhead, compared to the

original NBNN formulation. Interesting results are also given concerning the relative efficiency of radiometry invariant SIFT features [8]: Opponent SIFT is the descriptor that performed worst in NBNN, but it becomes the most efficient descriptor in our formulation.

The idea of designing optimal combinations of different feature channels by cross-validation has been studied by several authors. In the present context, the most relevant reference is [9]. While the method in [9] was conceived with the idea of having just one descriptor per image (either a global texture descriptor or a bag of words), our method works best when the number of descriptors per image is large. In [4,10], an image is subdivided into a pyramid of regions at different scales, and each region represents a channel. This fundamentally differs from our work in that they use bags of words to represent each image subregion. The idea of considering each image region as a channel can be applied in our context without any modification. With respect to the sub-window search, the idea is that in a cluttered background, classification performs best when first locating the most likely object position. This is close to the concept of region of interest developed in [11]. The detection scheme we use is inspired by [2,12].

The remainder of this paper is organized as follows. Original NBNN as well as the modification we propose are summarized in Section 2. In section 3, the adaptation of the optimal NBNN formulation to the problem of object detection is presented. Experimental results on three real datasets are reported in section 4.

2 Parametric NBNN Classification

2.1 Initial Formulation of NBNN

In this section, we briefly recall the main arguments of NBNN described by Boiman *et al.* [1] and introduce some necessary notation.

In an image I with hidden class label c_I, we extract K_I features $(d_k^I)_k \in \mathbb{R}^D$. Under the naive Bayes assumption, and assuming all image labels are equally probable ($P(c) \sim cte$) the optimal prediction $\hat{c}_I$ of the class label of image I maximizes the product of the feature probabilities relatively to the class label:

$$\hat{c}_I = \arg\max_c \prod_{k=1}^{K_I} P(d_k^I|c). \tag{1}$$

The feature probability conditioned on the image class $P(d_k^I|c)$ can be estimated by a non-parametric kernel estimator, also called Parzen-Rosenblatt estimator. If we note $\chi^c = \left\{d_k^J | c_J = c, 1 \leq k \leq K_J\right\}$ the set of all features from all training images that belong to class c, we can write:

$$P(d_k^I|c) = \frac{1}{Z} \sum_{d \in \chi^c} \exp\left(\frac{- \parallel d_k^I - d \parallel^2}{2\sigma^2}\right), \tag{2}$$

where σ is the bandwidth of the density estimator. In [1], this estimator is further approximated by the largest term from the sum on the RHS. This leads to a quite simple expression:

$$\forall d,\ \forall c,\ -\log\left(P(d|c)\right) \simeq \min_{d' \in \chi^c} \parallel d - d' \parallel^2 . \tag{3}$$

The decision rule for image I is thus:

$$\hat{c}_I = \arg\max_c P(I|c) = \arg\min_c \sum_k \min_{d \in \chi^c} \parallel d_k^I - d \parallel^2 . \tag{4}$$

This classifier is shown to outperform the usual nearest neighbor classifier. Moreover, it does not require any feature quantization step, and the descriptive power of image features is thus preserved.

The reasoning above proceeds in three distinct steps: the naive Bayes assumption considers that image features are independent identically distributed given the image class c_I (equation 1). Then, the estimation of a feature probability density is obtained by a non-parametric density estimation process like the Parzen-Rosenblatt estimator (equation 2). NBNN is based on the assumption that the logarithm of this value, which is a sum of distances, can be approximated by its largest term (equation 3). In the following section, we will show that the implicit simplification that consists in removing the normalization parameter from the density estimator is invalid in most practical cases.

Along with the notation introduced in this section, we will also need the notion of point-to-set distance, which is simply the squared Euclidean distance of a point to its nearest neighbor in the set: $\forall \Omega \subset \mathbb{R}^D, \forall x \in \mathbb{R}^D, \tau(x, \Omega) = \inf_{y \in \Omega} \parallel x - y \parallel^2$. In what follows, $\tau(x, \chi^c)$ will be abbreviated as $\tau^c(x)$.

2.2 Affine Correction of Nearest Neighbor Distance for NBNN

The most important theoretical limitation of NBNN is that in order to obtain a simple approximation of the log-likelihood, the normalization factor $1/Z$ of the kernel estimator is assumed to be the same for all classes. Yet, there is no *a priori* reason to believe that this assumption is satisfied in practice. If this factor significantly varies from one class to another, then the approximation of the maximum a posteriori class label $\hat{c}_I$ by equation 4 becomes unreliable.

It should be noted that the objection that we raise does not concern the core hypothesis of NBNN, namely the naive Bayes hypothesis and the approximation of the sum of exponentials of equation 2 by its largest term. In fact, in the following we will essentially follow and extend the arguments presented in [1] using the same starting hypothesis.

Non-parametric kernel density estimation requires the definition of a smoothing parameter σ, also called bandwidth. We consider the general case of a sample of K points $\{x_k | 1 \leq k \leq K\}$ drawn from a probability measure defined on some D-dimensional feature space Ω. The density of this probability measure can be estimated by:

$$\forall x \in \Omega, f(x) = \frac{1}{Z} \sum_{k=1}^{K} \exp\left(-\frac{||x - x_k||^2}{2\sigma^2} \right). \tag{5}$$

The value of Z is obtained by normalization of the density function: $\int_\Omega f(x)dx = 1 \Leftrightarrow Z = K(2\pi)^{\frac{D}{2}}\sigma^D$. We retain the NBNN assumption that the likelihood of a feature is approximately equal to the value of the largest term from the sum on the right hand side of equation 5. Here we provide an argument that supports this assumption: it is known

that the convergence speed of the Parzen-Rosenblatt (PR) estimator is $K^{-4/(4+D)}$ [13]. This means that in the case of a 128-dimensional feature space, such as the SIFT feature space, in order to reach an approximation bounded by $1/2$ we need to sample 2^{33} points. In practice, the PR estimator does not converge and there is little sense in keeping more than just the first term of the sum.

Thus, the log-likelihood of a visual feature d relatively to an image label c is:

$$-\log\big(P(d|c)\big) = -\log\left\{\frac{1}{Z^c}\exp\left(-\frac{\tau^c(d)}{2(\sigma^c)^2}\right)\right\} = \frac{\tau^c(d)}{2(\sigma^c)^2} + \log(Z^c), \qquad (6)$$

where $Z^c = |\chi^c|(2\pi)^{\frac{D}{2}}(\sigma^c)^D$. Recall that $\tau^c(d)$ is the squared Euclidean distance of d to its nearest neighbor in χ^c. In the above equations, we have replaced the class-independent notation σ, Z by σ^c, Z^c since, in general, there is no reason to believe that parameters should be equal across classes. For instance, both parameters are functions of the number of training features of class c in the training set.

Returning to the naive Bayes formulation, we obtain:

$$\forall c, -\log\left(P(I|c)\right) = \sum_{k=1}^{K_I}\left(\frac{\tau^c(d_k^I)}{2(\sigma^c)^2} + \log(Z^c)\right) = \alpha^c\sum_{k=1}^{K_I}\tau^c(d_k^I) + K_I\beta^c, \qquad (7)$$

where $\alpha^c = 1/(2(\sigma^c)^2)$ and $\beta^c = \log(Z^c)$ is a re-parametrization of the log-likelihood 6 that has the advantage of being linear in the model parameters. The image label is then decided according to a criterion that is slightly different from equation 4:

$$\hat{c}_I = \arg\min_c\left(\alpha^c\sum_{k=1}^{K_I}\tau^c(d_k^I) + K_I\beta^c\right). \qquad (8)$$

We note that this modified decision criterion can be interpreted in two different ways: it can either be interpreted as the consequence of a density estimator to which a multiplicative factor was added, or as an unmodified NBNN in which an affine correction has been added to the squared Euclidean distance. In the former, the resulting formulation can be considered different from the initial NBNN. In the latter, equation 8 can be obtained from equation 4 simply by replacing $\tau^c(d)$ by $\alpha^c\tau^c(d) + \beta^c$ (since α^c is positive, the nearest neighbor distance itself does not change). This formulation differs from [1] only in the evaluation of the distance function, leaving us with two parameters per class to be evaluated.

At this point, it is important to recall that the introduction of parameters α^c and β^c does not violate the naive Bayes assumption, nor the assumption of equiprobability of classes. In fact, the density estimation correction can be seen precisely as an enforcement of these assumptions. If a class is more densely sampled than others (i.e: its feature space contains more training samples), then the NBNN estimator will have a bias towards that class, even though it made the assumption that all classes are equally probable. The purpose of setting appropriate values for α^c and β^c is to correct this bias.

It might be noted that deciding on a suitable value for α^c and β^c reduces to defining an appropriate bandwidth σ^c. Indeed, the dimensionality D of the feature space and the number $|\chi^c|$ of training feature points are known parameters. However, in practice,

choosing a good value for the bandwidth parameter is time-consuming and inefficient. To cope with this issue, we designed an optimization scheme to find the optimal values of parameters α^c, β^c with respect to cross-validation.

2.3 Multi-channel Image Classification

In the most general case, an image is described by different features coming from different sources or sampling methods. For example, we can sample SIFT features and local color histograms from an image. We observe that the classification criterion of equation 1 copes well with the introduction of multiple feature sources. The only difference should be the parameters for density estimation, since feature types correspond, in general, to different feature spaces.

In order to handle different feature types, we need to introduce a few definitions and adapt our notation. In particular, we define the concept of *channel*: a channel χ is a function that associates a set of finite-dimensional characteristics to an image I: $\forall I, \chi(I) \subset \mathbb{R}^{d_\chi}$. Channels can be defined arbitrarily: a channel can be associated to a particular detector/descriptor pair, but can also represent global image characteristics. For instance, an image channel can consist in a single element, such as the global color histogram.

Let us assume we have defined a certain number of channels $(\chi_n)_{1 \leq n \leq N}$, that are expected to be particularly relevant to the problem at hand. Adapting the framework of our modified NBNN to multiple channels is just a matter of changing notation. Similarly to the single-channel case, we aim here at estimating the class label of an image I:

$$\hat{c}_I = \arg\max_c P(I|c), \quad \text{with} \quad P(I|c) = \prod_n \prod_{d \in \chi_n(I)} P(d|c). \tag{9}$$

Since different channels have different features spaces, the density correction parameters should depend on the channel index: α^c, β^c will thus be noted α^c_n, β^c_n. The notation from the previous section are adapted in a similar way: we call $\chi^c_n = \bigcup_{J|c_J=c} \chi_n(J)$ the set of all features from class c and channel n and define the distance function of a feature d to χ^c_n by: $\forall d,\ \tau^c_n(d) = \tau(d, \chi^c_n)$. This leads to the classification criterion:

$$\hat{c}_I = \arg\min_c \sum_n \left(\alpha^c_n \sum_{d \in \chi_n(I)} \tau^c_n(d) + \beta^c_n |\chi_n(I)| \right). \tag{10}$$

Naturally, when adding feature channels to our decision criterion, we wish to balance the importance of each channel relatively to its relevance to the problem at hand. Equation 10 shows us that the function of relevance weighting can be assigned to the distance correction parameters. The problems of adequate channel balancing and nearest neighbor distance correction should thus be addressed in one single step. In the following section, we present a method to find the optimal values of these parameters.

2.4 Parameter Estimation

We now turn to the problem of estimating values of α^c_n and β^c_n that are optimal for classification. To simplify mathematical derivations, let us denote by $\mathbf{X}^c(I)$ the vector in $\mathbb{R}^{2N}$ defined by

$$X_n^c(I) = \sum_{d \in \chi_n(I)} \tau_n^c(d), \qquad X_{N+n}^c(I) = |\chi_n(I)|, \quad \forall n = 1, \ldots, N. \tag{11}$$

For every c, the vector $\mathbf{X}^c(I)$ can be considered as a global descriptor of image I. We also denote by $\boldsymbol{\omega}^c$ the $(2N)$-vector $(\alpha_1^c, \ldots, \alpha_N^c, \beta_1^c, \ldots, \beta_N^c)$ and by W the matrix that results from concatenation of vectors w^c for different values of c. Using these notation, the classifier we propose can be rewritten as:

$$\hat{c}_I = \arg\min_c \ (\boldsymbol{\omega}^c)^\top \mathbf{X}^c(I), \tag{12}$$

where $(\boldsymbol{\omega}^c)^\top$ stands for the transpose of $\boldsymbol{\omega}^c$. This is close in spirit to the winner-takes-all classifier widely used for the multiclass classification.

Given a labeled sample $(I_i, c_i)_{i=1,\ldots,K}$ independent of the sample used for computing the sets χ_n^c, we can define a constrained linear energy optimization problem that minimizes the hinge loss of a multi-channel NBNN classifier:

$$E(W) = \sum_{i=1}^{K} \max_{c: c \neq c_i} \left(1 + (\boldsymbol{\omega}^{c_i})^\top \mathbf{X}^{c_i}(I_i) - (\boldsymbol{\omega}^c)^\top \mathbf{X}^c(I_i)\right)_+, \tag{13}$$

where $(x)_+$ stands for the positive part of a real x. The minimization of $E(W)$ can be recast as a linear program since it is equivalent to minimizing $\sum_i \xi_i$ subject to constraints:

$$\xi_i \geq 1 + (\boldsymbol{\omega}^{c_i})^\top \mathbf{X}^{c_i}(I_i) - (\boldsymbol{\omega}^c)^\top \mathbf{X}^c(I_i), \quad \forall i = 1, \ldots, K, \ \forall c \neq c_i, \tag{14}$$

$$\xi_i \geq 0 \quad \forall i = 1, \ldots, K, \tag{15}$$

$$(\boldsymbol{\omega}^c)^\top \mathbf{e}_n \geq 0, \quad \forall n = 1, \ldots, N, \tag{16}$$

where $\mathbf{e}_n$ stands for the vector of $\mathbb{R}^{2N}$ having all coordinates equal to zero, except for the nth coordinate, which is equal to 1. This linear program can be solved quickly for a relatively large number of channels and images[1]. In practice, the number of channels should be kept small relatively to the number of training samples to avoid overfitting. The computational complexity of solving the aforementioned linear program is negligible w.r.t. the complexity of computing the global descriptors $\mathbf{X}^c$ based on the nearest neighbor search.

Our contribution at this point is two-fold. We have proposed a natural parametric version of NBNN that is designed to improve the predictive performance of NBNN. We have also integrated the possibility to optimally combine multiple feature channels in the classifier. Due to the fact that we estimate the distance correcting weights through the optimization of the hinge loss, the parameters α_n^c, β_n^c up-weight channels that are most relevant to classification.

3 Multi-channel Classification by Detection

Optimal NBNN was designed in the goal of classifying images with little background clutter, and it is bound to fail on images containing too many background features.

[1] Our implementation makes use of the GNU linear programming kit `http://www.gnu.org/software/glpk/`

Classification by detection (*cf.* [7] and the references therein) is tailored to this kind of situations. It consists in selecting the image region that is the most likely to contain the object instance. In this section, we adapt our optimal NBNN to the problems of classification by detection. We will see that the final formulation is mostly identical to the formulation we adopted for general image classification.

We shall adopt the experimental framework given by the annotated Graz-02 dataset [14]: object instances from class c are surrounded by background clutter, denoted $\overline{c}$. Keeping the initial naive Bayes as well as the class equiprobability assumptions, our goal is to maximize the probability of the joint object label c and position π inside the image conditioned on the image content. We further assume object positions are equiprobable ($P(\pi|c) = P(\pi) = cte$). The image class estimate now takes the following form: $(\hat{c}_I, \hat{\pi}_I) = \arg\max_{c,\pi} P(I|c,\pi)$. Following the same line of thought as in NBNN, we can expand the likelihood term under the naive Bayes assumption: $P(I|c,\pi) = \prod_n \prod_{d\in\chi_n(I)} P(d|c,\pi)$ for all c and π.

At this point, we make the additional assumption that a feature probability knowing the object class and position only depends on the point belonging or not to the object:

$$\forall n,\ d,\ c,\ -\log\left(P(d|c,\pi)\right) = \begin{cases} \tau_n^c(d) \text{ if } d \in \pi \\ \tau_n^{\overline{c}}(d) \text{ if } d \notin \pi. \end{cases} \tag{17}$$

In the above equation, we have written the feature-to-set distance functions τ_n^c and $\tau_n^{\overline{c}}$ without apparent density correction in order to alleviate the notation. We leave to the reader the task of replacing τ_n^c by $\alpha_n^c \tau_n^c + \beta_n^c$ in the equations of this section.

The image log-likelihood function is now decomposed over all features inside and outside the object: $E(I,c,\pi) \triangleq -\log\big(P(I|c,\pi)\big) = \sum_n \big(\sum_{d\in\pi} \tau_n^c(d) + \sum_{d\notin\pi} \tau_n^{\overline{c}}(d)\big)$. The term on the RHS can be rewritten:

$$E(I,c,\pi) = \sum_n \left\{ \sum_{d\in\pi} (\tau_n^c(d) - \tau_n^{\overline{c}}(d)) + \sum_d \tau_n^{\overline{c}}(d) \right\}. \tag{18}$$

Observing that the second sum on the RHS does not depend on π, we get $E(I,c,\pi) = E_1(I,c,\pi) + E_2(I,c)$, where $E_1(I,c,\pi) = \sum_n \sum_{d\in\pi} \big(\tau_n^c(d) - \tau_n^{\overline{c}}(d)\big)$ and $E_2(I,c) = \sum_n \sum_d \tau_n^{\overline{c}}(d)$. Let us define the optimal object position $\hat{\pi}^c$ relatively to class c as the position that minimizes the first energy term: $\hat{\pi}^c = \arg\min_\pi E_1(I,c,\pi)$ for all c. Then, we can obtain the most likely image class and object position by:

$$\hat{c}_I = \arg\min_c \left(E_1(I,c,\hat{\pi}^c) + E_2(I,c)\right), \qquad \hat{\pi}_I = \hat{\pi}^{\hat{c}_I}. \tag{19}$$

For any class c, finding the rectangular window $\hat{\pi}^c$ that is the most likely candidate can be done naively by exhaustive search, but it proves prohibitive. Instead, we make use of fast branch and bound subwindow search [2]. The method used to search for the image window that maximizes the prediction of a linear SVM can be generalized to any classifier that is linear in the image features, such as our optimal multi-channel NBNN.

In short, the most likely class label and object position for a test image I are found by the following algorithm:

```
declare variables ĉ, π̂
Ê = +∞
for each class label c do
  find π̂^c by efficient branch and bound subwindow search
  π̂^c = arg min_π E_1(I, c, π)
  if E_1(I, c, π̂^c) + E_2(I, c) < Ê then
    Ê = E_1(I, c, π̂^c) + E_2(I, c)
    ĉ = c
    π̂ = π̂^c
  end if
end for
return ĉ, π̂
```

4 Experiments

Our optimal NBNN classifier was tested on three datasets: Caltech-101 [15], SceneClass 13 [16] and Graz-02 [14]. In each case, the training set was divided into two equal parts for parameter selection. Classification results are expressed in percent and reflect the rate of good classification, per class or averaged over all classes.

A major practical limitation of NBNN and of our approach is the computational time necessary to nearest neighbor search, since the sets of potential nearest neighbors to explore can contain of the order of 10^5 to 10^6 points. We thus need to implement an appropriate search method. However, the dimensionality of the descriptor space can also be quite large and traditional exact search methods, such as kd-trees or vantage point trees [17] are inefficient. We chose Locality Sensitive Hashing (LSH) and addressed the thorny issue of parameter tuning by multi-probe LSH[2] [18] with a recall rate of 0.8. We observed that resulting classification performance are not overly sensitive to small variations in the required recall rate. However, computations speed is: compared to exhaustive naive search, the observed speed increase was more than ten-fold. Further improvement in the execution times can be achieved using recent approximate NN-search methods [19,20].

Let us describe the databases used in our experiments.

Caltech-101 (5 classes). This dataset includes the five most populated classes of the Caltech-101 dataset: faces, airplanes, cars-side, motorbikes and background. These images present relatively little clutter and variation in object pose. Images were resized to a maximum of 300×300 pixels prior to processing. The training and testing sets both contain 30 randomly chosen image per class. Each experiment was repeated 20 times and we report the average results over all experiments.

SceneClass 13. Each image of this dataset belongs to one of 13 indoor and outdoor scenes. We employed 150 training images per class and assigned the rest to the testing set.

Graz-02. This manually segmented dataset contains instances of three classes: bike, people or car. Each image belongs to just one class. The training and testing sets

[2] For which an open source implementation exists: http://lshkit.sourceforge.net/

are both composed of 100 images per class. This database is considered as challenging [21] since the objects of interest are not necessarily central or dominant. Furthermore, they are subject to significant pose variation and partial occlusion.

4.1 Single-Channel Classification

The impact of optimal parameter selection on NBNN is measured by performing image classification with just one feature channel. We chose SIFT features [22] for their relative popularity. Results are summarized in Tables 1 and 2.

Table 1. Performance comparison between the bag of words classified by linear and χ^2-kernel SVM, the NBNN classifier and our optimal NBNN

Datasets	BoW/SVM	BoW/χ^2-SVM	NBNN [1]	Optimal NBNN
SceneClass13 [16]	67.85 $\pm$0.78	76.7 $\pm$0.60	48.52 $\pm$1.53	75.35 $\pm$0.79
Graz02 [14]	68.18 $\pm$4.21	77.91 $\pm$2.43	61.13 $\pm$5.61	78.98 $\pm$2.37
Caltech101 [15]	59.2 $\pm$11.89	89.13 $\pm$2.53	73.07 $\pm$4.02	89.77 $\pm$2.31

In Table 1, the first two columns refer to the classification of bags of words by linear SVM and by χ^2-kernel SVM. In all three experiments we selected the most efficient codebook size (between 500 and 3000) and feature histograms were normalized by their L^1 norm. Furthermore, only the results for the χ^2-kernel SVM with the best possible value (in a finite grid) of the smoothing parameter are reported. In Table 2, we omitted the results of BoW/SVM because of their clear inferiority w.r.t. BoW/χ^2-SVM.

Table 2. Performance comparison between the bag of words classified by χ^2-kernel SVM, the NBNN classifier and our optimal NBNN. Per class results for Caltech-101 (5 classes) dataset.

Class	BoW/χ^2-SVM	NBNN [1]	Optimal NBNN
Airplanes	91.99 $\pm$ 4.87	34.17 $\pm$11.35	95.00 $\pm$ 3.25
Car-side	96.16 $\pm$ 3.84	97.67 $\pm$ 2.38	94.00 $\pm$ 4.29
Faces	82.67 $\pm$ 9.10	85.83 $\pm$ 9.02	89.00 $\pm$ 7.16
Motorbikes	87.80 $\pm$ 6.28	71.33 $\pm$19.13	91.00 $\pm$ 5.69
Background-google	87.50 $\pm$ 6.22	76.33 $\pm$22.08	79.83 $\pm$ 10.67

There are two lessons to be learned from these experiments: the first is that correcting the NBNN formulation proves to be an absolute necessity if we want use unquantized features to advantage. Indeed the gain produced by parameter selection is almost systematic and exceeds 15 percentage points (in average) for the SceneClass and Graz-02 datasets. Secondly, we observe that the accuracy of NBNN is comparable to the state-of-the-art classification procedures such as BoW/χ^2-SVM. It should also be noted unlike NBNN, BoW/χ^2-SVM involves a tuning parameter the choice of which is a delicate issue.

To our knowledge, the state-of-the-art reported in the literature are 73.4% [23] for SceneClass13 (with an experimental setting however different from ours, since the authors use half of the dataset for training and the other half for testing), and 82.7% [24] for Graz-02 (using 150 positive and 150 negative images for training, for each non-background class). Given the relatively small training set that we use, our results compare favorably.

4.2 Radiometry Invariance

In this experiment, results highlight the necessity of parametric density estimation to make best use of visual features. In [8], different radiometry invariants of SIFT are presented and their relative performances are evaluated. Our own experiments made with the initial formulation of NBNN concur with the conclusions of [8],

As we find that the most efficient descriptors are: rgSIFT, followed by cSIFT and transformed color SIFT (cf. Table 3). The order of these descriptors roughly corresponds to the conclusions of [8]. Experiments revealed that the performance exhibited by optimal NBNN reverse this sequence: opponentSIFT becomes one of the best descriptors, with 91.10% good classification rate, while rgSIFT performs worst, with 85.17%. Thus, a wrong evaluation of the feature space properties undermines the descriptor performance.

4.3 Multi-channel Classification

The notion of channel is sufficiently versatile to be adapted to a variety of different contexts. In this experiment, we borrow the idea developed in [4] to subdivide the image in different spatial regions. We consider that an image channel associated to a certain

Table 3. Caltech101 (5classes): Influence of various radiometry invariant features. Best and worst SIFT invariants are highlighted in blue and red, respectively.

Feature	BoW/χ^2-SVM	NBNN [1]	Optimal NBNN
SIFT	88.90 ±2.59	73.07 ±4.02	89.77 ±2.31
OpponentSIFT	89.90 ±2.18	72.73 ±6.01	91.10 ±2.45
rgSIFT	86.03 ±2.63	80.17 ±3.73	85.17 ±4.86
cSIFT	86.13 ±2.76	75.43 ±3.86	86.87 ±3.23
Transf. color SIFT	89.40 ±2.48	73.03 ±5.52	90.01 ±3.03

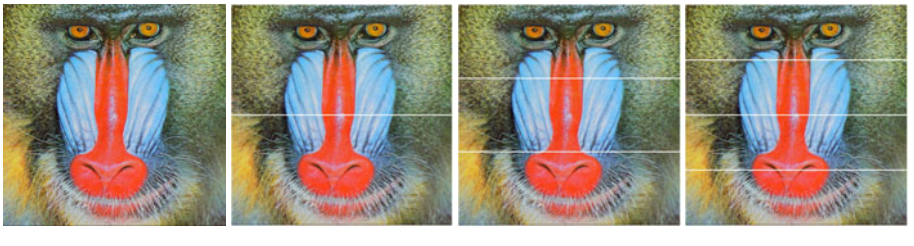


Fig. 2. Feature channels as image subregions: 1×1, 1×2, 1×3, 1×4

Table 4. Multi-channel classification, SceneClass13 dataset

Channels	#channels	NBNN	Optimal NBNN
1×1	1	48.52	75.35
$1 \times 1 + 1 \times 2$	3	53.59	76.10
$1 \times 1 + 1 \times 3$	4	55.24	76.54
$1 \times 1 + 1 \times 4$	5	55.37	78.26

image region is composed of all features that are located inside this region. In practice, image regions are regular grids of fixed size. We conducted experiments on the SceneClass13 dataset with 1 (1×1), 3 ($1 \times 1 + 1 \times 2$), 4 ($1 \times 1 + 1 \times 3$) and 5 ($1 \times 1 + 1 \times 4$) channels (see Fig. 2 for an illustration). Results are summarized in table 4. As we can see by comparing the first line with the subsequent lines, adding channels increases the rate of correct classification. Best performances are recorded in the experiment with the largest number of channels.

4.4 Classification by Detection

The Graz-02 dataset is a good example of the necessity of classification by detection for diminishing the importance of background clutter. In this set of experiments, the dataset is divided into just two classes: the positive class contains images of bicycles, while the negative class contains all other images. In this context, the estimated label of a test image I is given by:

$$\hat{c}_I = \text{sign}\left(E_2(I, back) - E(I, bike, \hat{\pi}^{bike})\right), \tag{20}$$

where we have retained notation from Section 3. The distance correction parameters that have to be determined for this problem are the α_n^c, β_n^c where c is in $\left\{bike, \overline{bike}, back\right\}$. For the sake of parameter selection, the sets of images from classes $bike$ and $\overline{bike}$ are obtained by decomposing each positive image in two complementary parts: the points located on a bicycle instance are in $bike$ while others are in $\overline{bike}$. Density estimation parameters were learned using the procedure described in Section 2.4.

We combined all five SIFT radiometry invariants already employed in Section 4.2. With classification by detection, we raised the classification rate of optimal NBNN from 78.70% to 83.60%, while classification by detection with NBNN achieved just 68.35%. Detection examples are shown in Fig. 1 and 3.

Our results by optimal NBNN (both for classification and classification by detection) are close to the results reported in [25, 21] and [26], where the rate of classification is 77%, 80% and 84% respectively (see Table 5 for details).

Table 5. Per-class classification rate for the Graz-02 database

Class	NBNN	Optimal NBNN	Optimal NBNN by detection	[21]	[25]	[26]
bike	68.35 ± 10.66	78.70 ± 4.67	83.6	80.5	77.8	84.4
people	45.10 ± 12.30	76.20 ± 5.85	–	81.7	81.2	–
car	42.40 ± 15.41	82.05 ± 4.88	–	70.1	70.5	79.9

Fig. 3. Subwindow detection for NBNN (red) and optimal NBNN (green). For this experiment, all five SIFT radiometry invariants were combined. (see Section 4.4)

It can be observed that the non-parametric NBNN usually converges towards an optimal object window that is too small relatively to the object instance. This is due to the fact that the background class is more densely sampled. Consequently, the nearest neighbor distance gives an estimate of the probability density that is too large. It was precisely to address this issue that optimal NBNN was designed.

5 Conclusion

In this paper, we proposed a parametric version of the NBNN classifier as well as a method for learning the parameters from a labeled set of images. The flexibility of this new classifier is exploited for defining its multi-channel counterpart and for adapting it to the task of object localization and classification by detection. Both in theory and in practice, it is shown that the new approach is much more powerful than the original NBNN in the case where the number of features per class is strongly class-dependent. Furthermore, the experiments carried out on some standard databases demonstrate that parametric NBNN can compete with other state-of-the-art approaches to object classification. The C++ implementation of the optimal NBNN is made publicly available at `http://code.google.com/p/optimal-nbnn/`.

Testing alternative strategies for parameter optimization step [27] and combining our approach with approximate nearest-neighbor search [19] are interesting avenues for future research.

References

1. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
2. Lampert, C., Blaschko, M., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR (2008)

3. Marszałek, M., Schmid, C., Harzallah, H., van de Weijer, J.: Learning object representations for visual object class recognition. In: Visual Recognition Challange workshop (2007)
4. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
5. Zhang, H., Berg, A.C., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: CVPR (2006)
6. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. International Journal of Computer Vision 87, 316–336 (2010)
7. Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. In: International Conference on Computer Vision (2009)
8. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. In: T-PAMI (2010)
9. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: ICCV (2007)
10. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: International Conference on Image and Video Retrieval, ICIVR (2007)
11. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: ICCV (2007)
12. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR (2009)
13. Stone, C.: Optimal uniform rate of convergence for nonparametric estimators of a density function or its derivatives. Recent advances in statistics (1983)
14. Marszałek, M., Schmid, C.: Accurate object localization with shape masks. In: CVPR (2007)
15. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. T-PAMI 28, 594–611 (2006)
16. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
17. Yianilos, P.N.: Data structures and algorithms for nearest neighbor search in general metric spaces. In: SODA: ACM-SIAM Symposium on Discrete Algorithms (1993)
18. Dong, W., Wang, Z., Josephson, W., Charikar, M., Li, K.: Modeling LSH for performance tuning. In: CIKM, pp. 669–678. ACM, New York (2008)
19. Muja, M., Lowe, D.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISAPP (2009)
20. Jégou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis & Machine Intelligence (to appear 2010)
21. Mutch, J., Lowe, D.G.: Object class recognition and localization using sparse features with limited receptive fields. Int. J. Comput. Vision 80, 45–57 (2008)
22. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2003)
23. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via pLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
24. Ling, H., Soatto, S.: Proximity distribution kernels for geometric context in category recognition. In: ICCV (2007)
25. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. PAMI 28 (2004/2006)
26. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: Neural Information Processing Systems, NIPS (2006)
27. Lee, Y., Lin, Y., Wahba, G.: Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. JASA 99, 67–81 (2004)

Weakly Supervised Classification of Objects in Images Using Soft Random Forests

Riwal Lefort[1,2], Ronan Fablet[2], and Jean-Marc Boucher[2]

[1] Ifremer/STH, Technopole Brest Iroise, 29280 Plouzane
[2] Telecom-Bretagne/LabSticc, Technopole Brest Iroise, 29280 Plouzane
{riwal.lefort,ronan.fablet,jm.boucher}@telecom-bretagne.eu

Abstract. The development of robust classification model is among the important issues in computer vision. This paper deals with weakly supervised learning that generalizes the supervised and semi-supervised learning. In weakly supervised learning training data are given as the priors of each class for each sample. We first propose a weakly supervised strategy for learning soft decision trees. Besides, the introduction of class priors for training samples instead of hard class labels makes natural the formulation of an iterative learning procedure. We report experiments for UCI object recognition datasets. These experiments show that recognition performance close to the supervised learning can be expected using the propose framework. Besides, an application to semi-supervised learning, which can be regarded as a particular case of weakly supervised learning, further demonstrates the pertinence of the contribution. We further discuss the relevance of weakly supervised learning for computer vision applications.

1 Introduction

The paper focuses on weakly supervised learning that includes both the supervised and semi-supervised learning. It considers probability vectors that indicate the prior of each class for each sample of the training set. The same notations are used throughout the paper. Let $\{x_n, \pi_n\}_n$ be the training dataset, where x_n is the feature vector for sample n and $\pi_n = \{\pi_{ni}\}_i$ is the prior vector for sample n, i indexing the classes. π_{ni} gives the likelihood for the example x_n to belong to class i. A supervised case corresponds to priors π_{ni} set to 0 or 1 whether or not sample n belongs to class i. The semi-supervised learning is also a specific case where training priors π_{ni} are given as 0 or 1 for the subset of the fully labelled training samples and as uniform priors for the remaining unlabelled samples.

Weakly supervised learning covers several cases of interest. Especially, in image and video indexing issue, object recognition dataset may involve training images labelled with the presence or the absence of each object category [1] [2] [3] [4] [5] [6]. Again such presence/absence dataset can be regarded as specific cases of training priors. As an other example, one can imagine an annotation by experts with some uncertainty measure [7]. This situation would be typical from photo-interpretation applications especially remote sensing applications [8].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 185–198, 2010.

A further generalization can be issued from expert-driven or prior automated analysis providing some confidence or uncertainty measure of the classification of objects or group of objects. This is typical from remote sensing applications. For instance, in the acoustics sensing of the ocean for fisheries management [9], images of fish schools are labelled with class proportions that lead to individual priors for each fish school. Such training dataset provided with class priors instead of hard class labels could also be dealt with when a cascade of classifiers or information could be processed before the final decision. In such cases, one could benefit from soft decisions to keep all relevant information in the cascade until the final decisions. This is typical of computer vision applications where multiple sources of information may be available [10] [11].

In this paper, we address weakly supervised learning using decision trees. The most common probabilistic classifiers are provided by generative models learnt using Expectation Maximization algorithm [12] [13]. These generative probabilistic classifiers are also used in ensemble classifiers [14] as in boosting schemes or with iterative classifiers [11]. In contrast, we investigate the construction of decision trees with weakly supervised dataset. Decision tree and random forest are among the most flexible and efficient techniques for supervised object classification [15]. However to our knowledge, previous works only deal with decision trees that consider hard input and probabilistic output [16]. The second contribution of this work is to develop an iterative procedure for weakly supervised learning. The objective is to iteratively refine the class priors of the training data in order to improve the classification performance of the classifier at convergence. We report different experiments to demonstrate the relevance of these contributions. The focus is given to examples demonstrating the genericity of the proposed weakly supervised framework, including applications to semi-supervised learning. In all cases, the proposed approach favourably compares to previous work, especially hard decision trees, generative classification models, and discriminative classification models.

This paper is organized as follows. In section 2, we present the weakly supervised learning of decision trees. In section 3, the iterative procedure for weakly supervised learning is detailed. The application to semi-supervised learning is presented in section 4 while experiments and conclusions are given in sections 5 and 6.

2 Decision Trees and Random Forest

2.1 Supervised Decision Trees and Random Forests

In supervised learning, the method consists in splitting the descriptor space into sub-sets that are homogeneous in terms of object classes. More precisely, the feature space is split based on the maximization of the gain of information. Different split criteria have been proposed such as the Gini criterion [17], the Shannon entropy [18] [19], or other on statistical tests such as ANOVA [20] or χ^2 test [21]. All of these methods have shown to lead to rather equivalent classification performances.

We focus here on the C4.5 decision trees which are among the most popular [19]. During the training step, at a given node of the tree, the procedure chooses descriptor d and associated split value S_d that maximize information gain G:

$$\arg \max_{\{d,S_d\}} G(S_d) \tag{1}$$

where gain G is issued from the Shannon entropy of object classes [19]:

$$\begin{cases} G = \left(\sum_m E^m \right) - E^0 \\ E^m = -\sum_i p_{mi} log(p_{mi}) \end{cases}. \tag{2}$$

where E^0 indicates the entropy at the parent considered node, E^m the entropy at children node m, and p_{mi} the likelihood of class i at node m.

A test sample is passed trough the tree and follows the test rules associated with each node. It is assigned to the class of the terminal node (or descriptor subspace) that it reaches.

Random forests combine a "bagging" procedure [22] and the random selection of a subset of descriptors at each node [23]. The random forest [15] can provide probabilistic outputs given by the posterior distribution of the class votes over the trees of the forest. Additional randomization-based procedures can be applied during the construction of the tree [24]. In some cases, they may lead to improve performances. Here, the standard random forests will be considered [15].

2.2 Weakly Supervised Learning of Soft Decision Trees

In this section, we introduce a weakly supervised procedure for learning soft decision trees. Let us denote by $\{x_n, \pi_n\}$ the provided weakly supervised dataset.

In contrast to the standard decision tree, any node of the tree is associated with class priors. In the weakly supervised setting, the key idea is to propagate class priors through tree nodes rather than class labels as in the supervised case. Consequently, given a constructed decision tree, a test sample will be passed trough the tree and be assigned the class priors of the terminal it will reache.

Let us denote by p_{mi} the class priors at node m of the tree. The key aspect of the weakly supervised learning of the soft decision tree is the computation of class prior p_{mi} at any node m. In the supervised case it consists in evaluating the proportion of each class at node m. In a weakly supervised learning context, real classes are unknown and class proportions can not be easily assessed. We propose to compute p_{mi} as a weighted sum over priors $\{\pi_{ni}\}$ for all samples attached to node m. For descriptor d, denoting x_n^d the instance value and considering the children node m_1 that groups together data such as $\{x_n^d\} < S_d$, the following fusion rule is then proposed:

$$p_{m_1 i} \propto \sum_{\{n\}|\{x_n^d\}<S_d} (\pi_{ni})^\alpha \tag{3}$$

For the second children node m_2 that groups data such as $\{x_n^d\} > S_d$, the equivalent fusion rule is suggested:

$$p_{m_2 i} \propto \sum_{\{n\}|\{x_n^d\}>S_d} (\pi_{ni})^{\alpha} \tag{4}$$

The considered power α weighs low-uncertainty samples, i.e. samples such that class priors closer to 1 should contribute more to the overall cluster mean p_{mi}. An infinite exponent values resorts to assigning the class with the greatest prior over all samples in the cluster. In contrast, an exponent value close to zero withdraws from the weighted sum low class prior. In practice, for α from 0.1 to 8, performances are more or less the same accuracy. After experiments, α is set to 0.8. This setting comes to give more importance to priors close to one. If $\alpha < 1$, high class priors are given a similar greater weight compared to low class priors. If $\alpha > 1$, the closer to one the prior the greater the weight.

Considering a random forest, the output from each tree t for a given test data x is a prior vector $p_t = \{p_{ti}\}$. p_{ti} is the prior for class i at the terminal node reached for tree t. The overall probability that x is assigned to class i, i.e. posterior likelihood $p(y = i|x)$, is then given by the mean:

$$p(y = i|x) = \frac{1}{T} \sum_{t=1}^{T} p_{ti} \tag{5}$$

where $y_n = i$ denotes that sample x_n is assigned to class i. A hard classification resorts to selecting the most likely class according to posteriors (5).

In this paper, experiments are carried out to fix the mean optimal number of tree per forest. Results show that 100 trees per forests are optimal on average. Furthermore, following the random forest process, there is no pruning.

3 Iterative Classification

3.1 Naive Iterative Procedure

The basic idea of the iterative scheme is that the class priors of the training samples can be refined iteratively from the overall knowledge acquired by the trained classifier such that these class priors finally converge to the real class of the training samples. The classifier at a given iteration can then be viewed as a filter that should reduce the noise or uncertainty on the class priors of training samples. Note that this iterative method is only applied to the training dataset.

Such an iterative procedure has previously been investigated in different contexts, especially with probabilistic generative classifier [11]. Theoretical results regarding convergence properties can hardly be derived [25] [26], though good experimental performances have been reported [27]. The major drawbacks of this approach are possible over-training effects and the propagation of early classification errors [28]. Bayesian models may contribute to solve for these over-training issues.

Table 1. Naive iterative procedure for weakly supervised learning (IP1)

Given an initial training data set $T_1 = \{x_n, \pi_n^1\}$ and M iterations,

1. For m from 1 to M
   - Learn a classifier C_m from T_m.
   - Apply the classifier C_m to T_m.
   - Update $T_{m+1} = \{x_n, \pi_n^{m+1}\}$ with $\pi_n^{m+1} \propto \pi_n^1 p(x_n|y_n = i, C_m)$.
2. Learn the final classifier using T_{M+1}.

The implementation of this naive iterative procedure proceeds as follows for weakly supervised learning. At iteration m given the weakly supervised dataset $\{x_n, \pi_n^m\}$, a random forest C_m can be learnt. The updated random forest could be used to process any training sample $\{x_n, \pi_n^m\}$ to provide an updates class prior π^{m+1}. This update of class prior π^{m+1} should exploit both the output of the random forest and the initial prior π^1. Here, the updated priors are given by: $\pi_n^{m+1} \propto \pi_n^1 p(x_n|y_n = i, C_m)$ where $y_n = i$ denotes the classe variable for sample n.

This algorithm is sketched in Tab. 1. In the subsequent, this procedure will be referred to as IP1 (Iterative Procedure 1).

3.2 Randomization-Based Iterative Procedure without over Training

A major issue with the above naive iterative procedure is that the random forest is repeatedly applied to the training data such that over-training effects may be expected. Such over-training effects should be avoided.

To this end, we propose a second iterative procedure. The key idea is to exploit a randomization-based procedure to distinguish at each iteration separate training and test subsets. More precisely, we proceed as follows. At iteration m, the training dataset $T_m = \{x_n, \pi_n^m\}$ is randomly split into a training dataset Tr_m and a test dataset Tt_m according to a given proportion β. Tr_m is exploited to build a weakly supervised random forest C_m. Samples in Tt_m are passed

Table 2. Randomization-based iterative procedure for weakly supervised learning (IP2)

Given a training data set $T_1 = \{x_n, \pi_n^1\}$ and M iterations,

1. for m from 1 to M
   - Randomly split T_m in two groups: $Tr_m = \{x_n, \pi_n^m\}$ and $Tt_m = \{x_n, \pi_n^m\}$ according to a split proportion β.
   - Learn a classifier C_m from subset Tr_m.
   - Apply classifier C_m to subset Tt_m.
   - Update $Tt_{m+1} = \{x_n, \pi_n^{m+1}\}$ with $\pi_n^{m+1} \propto \pi_n^1 p(x_n|y_n = i, C_m)$.
   - Update training dataset T_{m+1} as Tt_{m+1}: $T_{m+1} = \{Tr_m, Tt_{m+1}\}$.
2. Learn the final classifier using T_{M+1}.

through random forest C_m and updated class priors are issued from the same rule as previously: $\pi_n^{m+1} \propto \pi_n^1 p(x_n|y_n = i, C_m)$. β gives the proportion of training examples in the training set Tr_m while the remainder $(1 - \beta)$ training examples fall in the test set Tt_m. Setting β obeys to a trade-off: for a good assessment of random forest C_m, the number of samples in Tr_m must be high enough. But if β is too high, only very few samples will be updated at each iteration leading to a very slow convergence of the algorithm. In practice β is typically set to 0.75.

The algorithm is shown in the table 2. In the subsequent, this procedure will be denoted as IP2 (Iterative Procedure 2).

4 Application to Semi-supervised Learning

4.1 Related Work

Semi-Supervised Learning is reviewed in [28]. Four types of methods can be distinguished. The first type includes generative models often exploiting Expectation Maximization schemes that assess parameters of mono-modal Gaussian models [29] [28] or multi-modal Gaussian models [30]. Their advantages are the consistency of the mathematical framework with the probabilistic setting. The second category refers to discriminative models such as the semi-supervised support vector machine (S3VM) [31] [28]. Despite a mathematically-sound basis and good performances, S3VM are subject to local optimization issues and S3VM can be outperformed by other models depending on the dataset. Graph-based classifier is an other well known category in semi-supervised learning [32] [28]. The approach is close to the K-nearest-neighbour approach but similarities between examples are also taken in account. The principal drawback is that this classifier is mostly transductive: generalization properties are rather weak and performances decrease with unobserved data. The last family of semi-supervised models is formed by iterative schemes such as the self-training approach [33] or the co-training approach [34] that is applicable if observation features can be split into two independent groups. The advantage is the good performance reached by these methods and the simplicity of the approach. Their drawbacks mostly lie in the difficulties to characterize convergence properties.

4.2 Self Training with Soft Random Forests

A semi-supervised version of the iterative procedure proposed in the previous section can be derived. Following a self training strategy, it conststs in initially training a random forest from groundtruthed training samples only. Then, at each iteration, unlabelled data are processed by the current classifier and the K samples with the greatest class posteriors are appended to the training database to retrain the classifier. If should be stressed that in the standard implementation of semi-supervised learning with SVMs and random forest the new samples appended to training set at each iteration are assigned class labels. In contrast, we benefit from the proposed weakly supervised decision trees. This is expected to reduce the propagation of classification errors. The sketch of semi-supervised learning is given in table 3.

Table 3. Soft self-training procedure for semi-supervised learning

Given an initial training data set $T = \{T_L, T_U\}$, where T_L contains labelled data and T_U unlabelled data, and M iterations.

1. For m from 1 to M
   - Learn a classifier C_m from T_L.
   - Apply classifier C_m to T_U.
   - For each classes, transfer from T_U to T_L the most confident examples, with weak label, according to the probabilistic classification.
2. Generate the final classifier using T_L.

5 Experiments

5.1 Simulation Protocol

In this section, we compare four classification models: IP1 using soft random forests, IP2 using soft random forests, soft random forests alone, and the generative model proposed in [2] for weakly labelled data.

Given a supervised dataset, a weakly supervised training dataset is built. We distribute all the training examples in several groups according to predefined target class proportions (table 4). All the instances in a given group are assigned the class proportion of the group. In table 4, we show an example of target class proportions for a three-class dataset. In this example, we can create groups containing from one class (supervised learning) to three classes. For each case of class-mixture, different mixture complexities can be created: from one class dominating the mixture, i.e. the prior of one class being close to one, to equiprobable class, i.e. equal values for non-zeros class priors.

To evaluate the performances of the proposed weakly supervised learning strategies, we consider different reference datasets of the UCI machine learning repository so that reported experiments could be reproduced. The three considered datasets have been chosen to provide representative examples of the datasets to be dealt with in computer vision applications. We do not consider datasets with two classes because they do not allow us to generate complex class-mixtures. D1 is an image segmentation dataset containing 7 classes of texture and 330 instances per class. Each sample is characterized by a 19-dimensional real feature vector. D1 is a typical computer vision dataset drawn from a database of 7 outdoor images (brickface, sky, foliage, cement, window, path, grass). D2 is the classical Iris dataset containing 3 classes and 50 instances per class. Each objectis characterized by geometric features, i.e. length and width of the petals. D3 is the Synthetic Control Chart Time Series dataset, containing 6 classes of typical line evolutions, 100 instances per classes, and 5 quantified descriptors. An interesting property of this dataset is that the distribution of the features within each class is not unimodal and cannot be easily modelled using a parametric approach. This is particularly relevant for computer vision applications where objects classes often lead non-linear manifolds in the feature space. Dataset D3

Table 4. Example of training class priors for a dataset with 3 classes. Different cases are carried out: from the supervised labelling to the high complexity mixture.

Dataset with 3 classes, 1-class mixture labels:
$\begin{pmatrix}1\\0\\0\end{pmatrix}$ $\begin{pmatrix}0\\1\\0\end{pmatrix}$ $\begin{pmatrix}0\\0\\1\end{pmatrix}$ (supervised learning)
Dataset with 3 classes, 2-class mixture labels:
$\begin{pmatrix}0.8\\0.2\\0\end{pmatrix}$ $\begin{pmatrix}0.2\\0.8\\0\end{pmatrix}$ $\begin{pmatrix}0.6\\0.4\\0\end{pmatrix}$ $\begin{pmatrix}0.4\\0.6\\0\end{pmatrix}$ $\begin{pmatrix}0.8\\0\\0.2\end{pmatrix}$ $\begin{pmatrix}0.2\\0\\0.8\end{pmatrix}$ $\begin{pmatrix}0.6\\0\\0.4\end{pmatrix}$ $\begin{pmatrix}0.4\\0\\0.6\end{pmatrix}$ $\begin{pmatrix}0\\0.8\\0.2\end{pmatrix}$ $\begin{pmatrix}0\\0.2\\0.8\end{pmatrix}$ $\begin{pmatrix}0\\0.6\\0.4\end{pmatrix}$ $\begin{pmatrix}0\\0.4\\0.6\end{pmatrix}$
Dataset with 3 classes, 3-class mixture labels:
$\begin{pmatrix}0.8\\0.1\\0.1\end{pmatrix}$ $\begin{pmatrix}0.1\\0.8\\0.1\end{pmatrix}$ $\begin{pmatrix}0.1\\0.1\\0.8\end{pmatrix}$ $\begin{pmatrix}0.4\\0.2\\0.2\end{pmatrix}$ $\begin{pmatrix}0.2\\0.4\\0.2\end{pmatrix}$ $\begin{pmatrix}0.2\\0.2\\0.4\end{pmatrix}$

was also chosen to investigate the classification of time series depicting different behaviours (cyclic, increasing vs. decreasing trend, upward vs. downward shift). This is regarded as a mean to evaluate the ability to discriminate dynamic contents particularly relevant for video analysis, i.e. activity recognition, trajectory classification, event detection.

Cross validation over 100 tests allows assessing a mean classification rate. 90% of data are used to train classifier while the 10% remainders are used to test. Dataset is randomly split every test and the procedure that affects weak labels to the training data is carried out every test. A mean correct classification rate is extracted over the cross validation.

5.2 Experiments on Weakly Supervised Dataset

We report weakly supervised experiments in table 5 for the tree datasets. Results are provided as a function of the training dataset and as a function of the class mixture complexity, from the supervised learning (1-class mixture) to the maximum complexity mixture (mixture with all classes). Results are reported for the iterative procedures IP1 (section 3.1) and IP2 (section 3.2), the weakly supervised learning of soft random forests, the generative and discriminant models previously proposed in [2] and [9]. The later respectively exploit Gaussian mixtures and a kernel Fisher discrimination technique.

Overall, the iterative process IP2 outperforms the other models. Even if random forests alone are outperformed by the generative model with D2, the iterative procedure leads to improved classification. The explanation is that class priors are iteratively refined to finally resort to less fuzzy priors. Experiments with dataset D3, particularly stress the relevance of the introduction of soft decision trees. Due to the interlaced structure of the feature distribution for each class of dataset D3, the generative and discriminative models perform poorly. In contrast the weakly supervised random forests reach correct classification rates close to the supervised reference even with complex 6-class training mixtures. For instance, considering D3 and 6-classes mixture labels, the iterative procedure IP2 combined to soft forest reach 98.8% (vs. 100% in the supervised case) where the generative end discriminative models only reach 58.3% and 59.8% of correct classification.

Table 5. Classification performances for datasets D1, D2, and D3: the mean correct classification rate (%) is reported as a function of the complexity of the mixture label for the 5 classification models IP1 + soft trees, IP2+ soft trees, soft trees and random forest alone, a EM-based generative algorithm [2], and a discriminative-based algorithm [9]

Dataset, type of mixtures	IP1 + soft trees	IP2 + soft trees	soft trees	Naive bayes [2]	Fisher + K-pca [9]
D1, 1 classes mixture	-	-	**96.1%**	83.7%	89.7%
D1, 2 classes mixture	90.7%	**96.1%**	92.3%	83.6%	89.2%
D1, 3 classes mixture	88.7%	**95.9%**	91.2%	84.4%	89.5%
D1, 4 classes mixture	88.3%	**94.4%**	88.4%	83.7%	89.1%
D1, 5 classes mixture	85.0%	**94.1%**	88.8%	83.8%	89.1%
D1, 6 classes mixture	75.2%	**92.7%**	84.6%	83.1%	89.1%
D1, 7 classes mixture	55.1%	81.4%	62.6%	75.1%	**85.9%**
D2, 1 classes mixture	-	-	**97.3%**	94.6%	96.0%
D2, 2 classes mixture	**97.3%**	**97.3%**	90.6%	95.3%	87.3%
D2, 3 classes mixture	84.0%	**92.6%**	81.3%	85.3%	76.6%
D3, 1 classes mixture	-	-	**100%**	77.1%	66.8%
D3, 2 classes mixture	90.5%	**100%**	90.0%	62.2%	63.6%
D3, 3 classes mixture	91.3%	**99.5%**	89.3%	62.1%	61.5%
D3, 4 classes mixture	82.1%	**98.1%**	75.6%	45.5%	62%
D3, 5 classes mixture	74.6%	**97.3%**	82.1%	47.3%	59.1%
D3, 6 classes mixture	94.0%	**98.8%**	88.6%	58.3%	59.8%

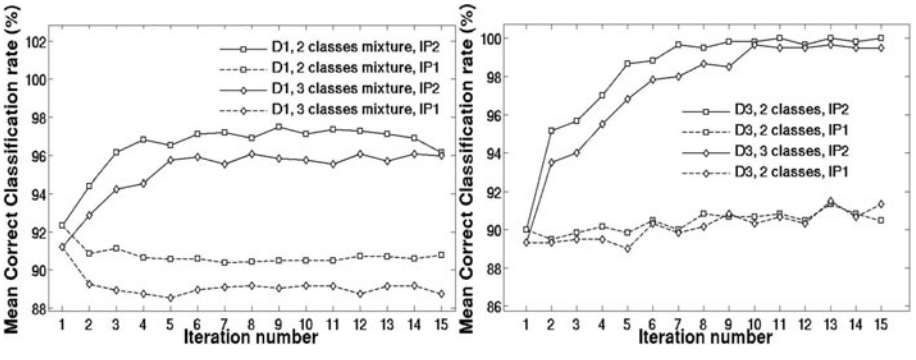


Fig. 1. Evolution of the performances of the iterative procedures IP1 and IP2 through iteration: dataset D1 (left), dataset D3 (right)

To further illustrate the behaviour of the iterative procedures IP1 and I2, we report in figure 1 the evolution of the mean correct classification rate (for the test set) as a function of the iteration for dataset D1 and D2 and several types of class mixtures. These plots state the relevance of the procedure IP2 compared to procedure IP1. Whereas the later does not lead to significant improvement over iterations, the gain in the correct classification rate can be up to 10% after the convergence of the IP2 procedure, for instance for dataset D2 and 2-class mixtures. The convergence is typically observed on a ten of iterations. These results can be explained by the fact that the IP2 procedure distinguishes at each iteration separate training and test set to update the random forests and the class priors.

5.3 Application to Fish School Classification in Sonar Images

Weakly supervised learning is applied to fisheries acoustics data [9] formed by a set of fish schools automatically extracted in sonar acoustics data. Fish schools are extracted as connected components using a thresholding-based algorithm. Each school is characterized by a X-dimensional feature vector comprising geometric (i.e. surface, width, height of the school) and acoustic (i.e. backscattered energy) descriptors. At the operation level, training samples would issue from the sonar in trawled areas, such any training school would be assigned the relative priors of each class. With a view to performing a quantitative evaluation, such weakly supervised situations are simulated from a groundtruthed fish school dataset. The later has been built from sonar images in trawled regions depicting only one species.

From results given in table 6, we perform a comparative evaluation based on the same methods than in section 5.2 as shown in table 5. Class proportions have been simulated as presented in table 4. Similar conclusions can be drawn. Overall the iterative procedure with soft random forests (IP2-SRF) outperforms the other techniques including the generative and discriminative models presented in [2] [9], except for the four-class mixture case where soft random forests alone perform better (58% vs. 55%). The operational situations typically involve mixtures between two or three species and the reported recognition performances (between 71% and 79%) are relevant w.r.t. ecological objectives in terms of species biomass evaluation and the associated expected uncertainty levels.

Table 6. Classification performances for sonar image dataset D4: the mean correct classification rate (%) is reported as a function of the complexity of the mixture label for the 5 classification models IP1 + soft trees, IP2+ soft trees, soft trees and random forest alone, a EM-based generative algorithm [2], and a discriminative-based algorithm [9].

Dataset, type of mixtures	IP1 + soft trees	IP2 + soft trees	soft trees	Naive bayes [2]	Fisher + K-pca [9]
D4, 1 classes mixture	-	-	**89.3%**	66.9%	69.9%
D4, 2 classes mixture	72.3%	**79.4%**	71.9%	52%	71.7%
D4, 3 classes mixture	62.9%	**70%**	68.3%	51.2%	65.9%
D4, 4 classes mixture	45.3%	55%	**58.7%**	47.9%	56.2%

5.4 Semi-supervised Experiments

Semi-supervised experiments have been carried out using a procedure similar to the previous section. Training and test sets are randomly built for a given dataset. Each training set is composed of labelled and unlabelled samples. We here report results for datasets D2 and D3 with the following experimental setting. For dataset D3 the training dataset contains 9 labelled examples (3 for each class) and 126 unlabelled examples (42 for each class). For dataset D3, we focus on a two-class example considering only samples corresponding to normal and cyclic pattern. Training datasets contain 4 labelled samples and 86 unlabelled samples per class. This particular experimental setting is chosen to illustrate the relevance of the semi-supervised learning when only very fulled labelled training

samples are available. In any case, the upper bound of the classification performances of a semi-supervised scheme is given by the supervised case. Therefore, only weak gain can be expected when a representative set of fully labelled samples is provided to the semi-supervised learning.

Five semi-supervised procedure are compared: three based on self-training (ST) strategies [28], with soft random forests (ST-SRF), with standard (hard) random forests (ST-RF), with a nave Bayes classifier (ST-NBC), a EM-based nave Bayes classifier (EM-NBC) [2] and the iterative procedure IP2 to soft random forest (IP2-SRF). Results are reported in figure 2.

These semi-supervised experiments first highlight the relevance of the soft random forests compared to their standard versions. For instance, when comparing both to a self-training strategy, the soft random forests lead to a gain of 5% of correct classification with dataset D3. This is regarded as a direct consequence of a reduced propagation of initial classification errors with soft decisions. The structure of the feature space for dataset D3 further illustrates as previously the flexibility of the random forest schemes compared to the other ones, especially generative models which behave poorly.

These experiments also demonstrate the relevance of the weakly supervised learning IP2-SRF in a semi-supervised context. The later favourably compares to the best self-training strategy (i.e. 90% vs 82.5% of correct classification for dataset D2 after 10 iterations). This can be justified by the relations between the two procedures. As mentioned in section 4, the self training procedure with soft random forests can be regarded as a specific implementation of the iterative procedure IP2. More precisely, the self-training strategy consists in iteratively appending unlabelled samples to the training dataset. At a given iteration, among the samples not yet appended to the training set, those with the greatest measures of the confidence in the classification are selected. Hence the classification decisions performed for the samples in the training set are never re-evaluated. In contrast

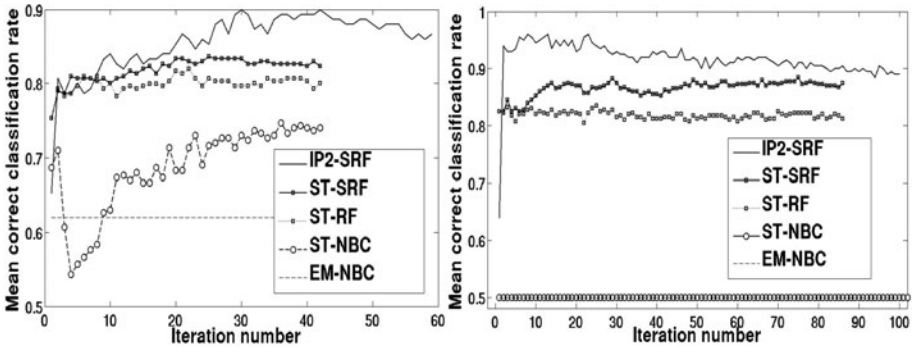


Fig. 2. Classification performances in semi-supervised contexts: dataset D2 (left) and dataset D3 (right) restricted to classes "standard patterns" and "cyclic pattern". Five approaches are compared: ST-SRF, ST-RF, ST-NBC, EM-NBC, and IP2-SRF (cf. text for details).

to this deterministic update of the training set, the weakly supervised iterative procedure IP2 exploits a randomization-based strategy where unlabelled sample are randomly picked to build at each iteration a training set. Therefore, soft classification decisions are repeatedly re-evaluated with respect to the updated overall knowledge. Then, the proposed entropy criterion (equation (3)) implies that fully labelled samples are also implicitly given more weight as in the soft training procedures. These different features support the better performances reported here for the iterative procedure IP2 combined to soft random forests.

6 Conclusion

We have presented in this paper methodological contributions for learning object class models in a weakly supervised context. The key feature is that classification information in the training datasets are given as the priors of the different classes for each training sample. This weakly supervised setting covers the supervised situations, the semi-supervised situations, and computer vision applications as the object recognition scheme that only specifies the presence or absence of each class in each image of the training dataset.

Soft random forests for weakly supervised learning: From a methodological point of view, a first original contribution is a procedure to learn soft random forests in a weakly supervised context. Interestingly the later is equivalent to the C4.5 random forest [15] if a supervised dataset is available such that recognition performances for low uncertainty priors can be granted. The second methodological contribution is an iterative procedure aimed at iteratively improving the learning model.

The experimental evaluation of these methodological contributions for several reference UCI datasets demonstrate their relevance compared to previous work including generative and discriminative models [2] [9]. We have also shown that these weakly supervised learning schemes are relevant for semi-supervised datasets for which they favourably compare to standard iterative techniques such as self-training procedures. The experiments support the statement widely acknowledged in pattern recognition that, when relevantly iterated, soft decisions perform better than hard decisions.

Weakly supervised learning for computer vision applications: The reference UCI datasets considered in the reported experiments are representative of different types of computer vision datasets (i.e. patch classification, object recognition, trajectory analysis). For these datasets, we have shown that recognition performances close to upper bounds provides by the supervised learning could be reached by the proposed weakly learning strategy even when the training set mostly high uncertainty class priors.

This is of particular importance as it supports the relevance of the weakly supervised learning to process uncertain or contradictory expert or automated preliminary interpretations. In any case, such a learning scheme should make in computer vision applications the construction of training datasets which is task

often a very tedious task. The later observation initially motivated the introduction of the semi-supervised learning and of the weakly supervised case restricted to presence and absence information. Our work should be regarded as a further development of these ideas to take into account more general prior information. As illustrated for instance by the fisheries acoustics dataset, we believe that this generalization may permit reaching relevant classification performances when the knowledge of presence and/or absence information only leads to unsatisfactory classification rates [35] [2].

Given the central role of the randomization-based sampling in the iterative procedure, future work will focus on its analysis both form experimental and theoretical points of view. The objectives will be to characterize the convergence of this procedure as well to evaluate different random sampling beyond the uniform sampling tester in the reported experiments. A stopping criteria might also be considered.

References

1. Crandall, D.J., Huttenlocher, D.P.: Weakly supervised learning of part-based spatial models for visual object recognition. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 16–29. Springer, Heidelberg (2006)
2. Ulusoy, I., Bishop, C.M.: Generative versus discriminative methods for object recognition. In: CVPR, vol. 2, pp. 258–265 (2005)
3. Ponce, J., Hebert, M., Schmid, C., Ziserman, A.: Toward Category-Level Object Recognition. LNCS, vol. 4170. Springer, Heidelberg (2006)
4. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for object recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 18–32. Springer, Heidelberg (2000)
5. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scaled-invariant learning of models for visual recognition. IJCV 71, 273–303 (2006)
6. Schmid, C.: Weakly supervised learning of visual models and its application to content-based retrieval. IJCV 56, 7–16 (2004)
7. Rossiter, J., Mukai, T.: Bio-mimetic learning from images using imprecise expert information. Fuzzy Set and Systems 158, 295–311 (2007)
8. van de Vlag, D., Stein, A.: Incorporing uncertainty via hierarchical classification using fuzzy decision trees. IEEE Transaction on GRS 45, 237–245 (2007)
9. Lefort, R., Fablet, R., Boucher, J.M.: Combining image-level and object-level inference for weakly supervised object recognition. Application to fisheries acoustics. In: ICIP (2009)
10. Maccormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. IJCV 39, 57–71 (2000)
11. Neville, J., Jensen, D.: Iterative classification in relational data. In: AAAI workshop on learning statistical models from relational data, pp. 42–49 (2000)
12. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse and other variants. Kluwer Academic Publishers, Dordrecht (1998)
13. Lachlan, G.M., Krishnan, T.: The EM algorithm and extentions. Wiley, Chichester (1997)
14. Kotsiantis, P., Pintelas, P.: Logitboost of simple bayesian classifier. Informatica Journal 29, 53–59 (2005)

15. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
16. Calonder, M., Lepetit, V., Fua, P.: Keypoint signatures for fast learning and recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 58–71. Springer, Heidelberg (2008)
17. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and regression trees. Chapman and Hall, Boca Raton (1984)
18. Quinlan, J.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
19. Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Francisco (1993)
20. Loh, W.Y., Shih, Y.Y.: Split selection methods for classification trees. Statistica Sinica 7, 815–840 (1997)
21. Kass, G.V.: An exploratory technique for invesgating large quantities of categorical data. Journal of applied statistics 29, 119–127 (1980)
22. Breiman, L.: Bagging predictors. Machine Learning 26, 123–140 (1996)
23. Ho, T.K.: Random decision forest. ICDAR (1995)
24. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning 36, 3–42 (2006)
25. Culp, M., Michailidis, G.: An iterative algorithm for extending learners to semi-supervised setting. In: The 2007 Joint Statistical Meetings (2007)
26. Haffari, G., Sarkar, A.: Analysis of semi-supervised learning with the yarowsky algorithm. In: 23rd Conference on Uncertainty in Artificial Intelligence (2007)
27. Macskassy, S.A., Provost, F.: A simple relational classifier. In: Proceedings of the second workshop on multi-relational data mining, pp. 64–76 (2003)
28. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning. MIT Press, Cambridge (2006)
29. Nigam, K., McCullum, A., Thrun, S., Mitchel, T.: Learning to classify text from labeled and unlabeled documents. In: AAAIJ (1998)
30. Nigam, K., McCallum, A., Thrun, S., Mann, G.: Text classification from labeled and unlabeled documents using em. Machine Learning 39, 103–134 (2000)
31. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML, pp. 200–209 (1999)
32. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)
33. Rosenberg, C., Hebert, M., Schneidermann, H.: Semi-supervised self-training of object detection models. In: WACV (2005)
34. Blum, A., Mitchel, T.: Combining labeled and unlabeled data with co-training. In: WCLT, pp. 92–100 (1998)
35. Fablet, R., Lefort, R., Scalabrin, C., Mass, J., Boucher, J.M.: Weakly supervised learning using proportion based information: an application to fisheries acoustic. In: International Conference on Pattern Recognition (2008)

Learning What and How of Contextual Models for Scene Labeling

Arpit Jain[1], Abhinav Gupta[2], and Larry S. Davis[1]

[1] University of Maryland College Park, MD, 20742
[2] Carnegie Mellon University, Pittsburgh, PA, 15213

Abstract. We present a data-driven approach to predict the importance of edges and construct a Markov network for image analysis based on statistical models of global and local image features. We also address the coupled problem of predicting the feature weights associated with each edge of a Markov network for evaluation of context. Experimental results indicate that this scene dependent structure construction model eliminates spurious edges and improves performance over fully-connected and neighborhood connected Markov network.

1 Introduction

Image understanding is one of the central problems in computer vision. Recently, there has been significant improvements in the accuracy of image understanding due to a shift from recognizing objects "in isolation" to context based recognition systems. Such systems improve recognition rates by augmenting appearance based models of individual objects with contextual information based on pair-wise relationships between objects. These relationships can be co-occurrence relationships or fine-grained spatial relationships. However, most approaches have employed brute force approaches to apply context - all objects are first (probabilistically) detected and connected in a massive network to which probabilistic inference methods are applied. First, this approach is clearly not scalable; but more important, it suffers from the serious drawback that it treats all pair-wise relationships in an image as equally important for image analysis. For example, consider the image shown in Figure 1, where our goal is to identify the unknown label of the region outlined in red (which we will refer to as the target), given the labels of other regions in the image. The regions labeled as building tend to force the label of the target towards building (two building regions co-occur more often than building and car) and the region labeled car tends to force the label of the target to be road, since car above road has higher probability in the contextual model than car above building. In the case of fully-connected models, the edges from the building regions to the target region outnumber the edges from other regions to the target and therefore the target is incorrectly labeled as building. If we had only utilized the relationship from the region associated with the car and ignored the relationships from other objects to predict the label of the target, then we would have labeled the target correctly.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 199–212, 2010.

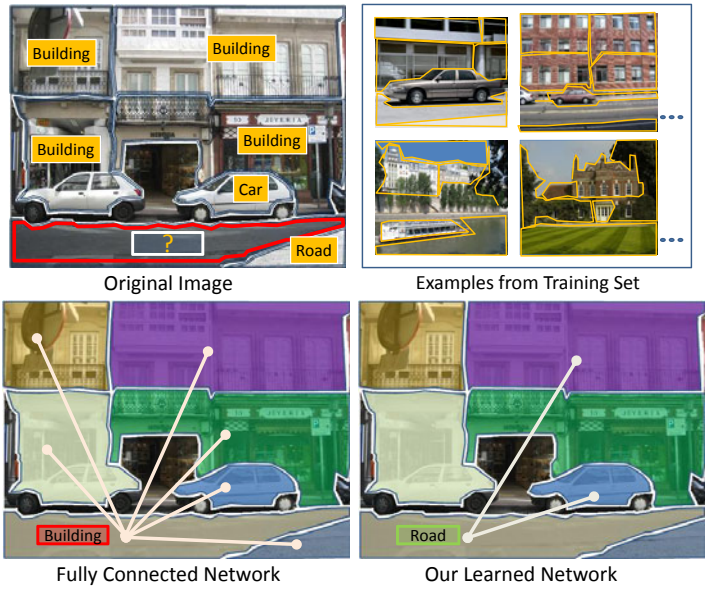


Fig. 1. An example from our dataset showing that all relations are not informative in a fully connected network and can lead to wrong labeling and how our proposed method learns "what" edges are important and removes dubious information.

Other approaches for applying context to image understanding have considered fixed structure Markov networks where only nodes corresponding to neighboring segments are linked by an edge [2]. Such approaches are based on the reasonable assumption that neighboring segments carry significant contextual information and might be sufficient for recognition. However, such neighborhood based connectivity schemes have two shortcomings: (1) Images are two-dimensional projection of the three-dimensional world. Two objects which are far in the 3D world might appear very close in the image plane and therefore some noisy relationships are included in the network. (2) The assumption that neighboring segments provide sufficient contextual information is too strong and does not hold in many cases. For example, in a sunset scene, the relationship between the appearance of the sky (orange) and the appearance of large bodies of water is useful for recognition of such bodies of water, even though there can be intervening land regions between them.

In this paper, we evaluate the importance of individual contextual-constraints and use a data-driven model for selection of **what** contextual constraints should be employed for solving a specific scene understanding problem, and for constructing a corresponding Markov-network. Unlike previous approaches that use fully connected or fixed structures based on neighborhood relationships, our approach predicts the structure of the Markov network (i.e., selects edges). Selection of edges is generally dependent on a combination of global and local factors

such as discriminativeness of regions. However, identifying the variables/factors associated with predicting the importance of a contextual edge a priori is difficult. Instead, we take a data driven approach to predict the importance of an edge, in which scenes similar to a "test" image are identified in the training dataset and utilized to predict which regions should be linked by an edge in the Markov network corresponding to the test image - referred to as edge prediction. Figure 1(a) shows an example of edge prediction from our test dataset. Our approach appropriately eliminates the edges from most of the building regions to the target and maintains the edge from the car. This leads to a correct labeling of the target.

To learn a data-driven(non-parametric) model of edge importance, we have to compute the importance of edges in the training data-set itself. This requires evaluating each edge in the training data-set with respect to other edges in the training data-set. Edges that represent consistent spatial-relationships between pairs-of-nouns are retained as informative edges and the rest are dropped. If a single 2D-spatial relationship was sufficient to represent constraints between a pair of nouns, then extracting consistent edges would be straight-forward. However, relationships between pairs of nouns are themselves scene-dependent (due to viewpoint, functional-context, etc.). For example, based on viewpoint, a road might be either below a car or around a car (see Figure 1(b)). Similarly, relationships are also based on function-context of an object. For example, a bottle can either be on the table or below the table based on its function (drinking vs. trash). Therefore, we cluster the relationships between pairs of nouns based on scene properties. For each cluster, we then learn feature-weights which reflect how much each feature of the vector of variables capturing spatial relationships is important for evaluating constraint/relationship satisfaction. For example, in a top-down view, road being "around" car is most important. Our approach not only learns the construction model for Markov networks, but also learns the feature weights which define how to evaluate the degree to which a relationship between a pair of nouns is satisfied. Again, instead of explicitly modeling the factors on which these feature weights depend, we utilize a data driven approach to produce pseudo-clusters of images and estimate **how** each contextual edge should be evaluated (See Figure 1(b)) in each cluster.

The contributions of our paper are: (1) A data driven approach for predicting **what** contextual constraints are important for labeling a specific image that uses only a subset of the relationships used in a fully-connected model. The resulting labeling are both more accurate and computed more efficiently compared to the fully-connected model. (2) A model for predicting **how** each contextual edge should be evaluated. Unlike previous approaches, which utilize a single spatial-relationship between a pair of objects (car above road), we learn a scene dependent model of context and can encode complex relationships (car above road from a standing person's viewpoint, but road around car from a top-down viewpoint).

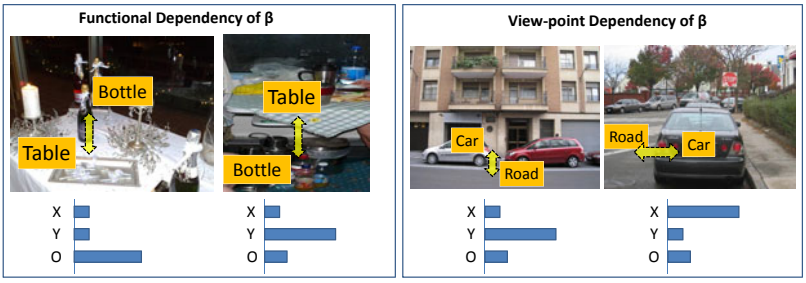


Fig. 2. The figure shows examples of how feature weights are a function of both local and global factors. Here we show how feature weights depend on function context and viewpoint. Pairwise features X,Y,O refer to differences in x-coordinates,difference in y-coordinates and overlap between two regions respectively.

2 Related Work

Recent research has shown the importance of context in many image and video understanding tasks [3,1,4,5,6]. Some of these tasks include segmentation and recognition. For object recognition, researchers have investigated various sources of context, including context from the scene [7], objects [4] and actions [22]. Scene based context harnesses global scene classification such as urban, landscape, kitchen etc to constraint the objects that can occur in the scene (for example, a car cannot occur in a kitchen). On the other hand, object based contextual approaches model object-object co-occurrence and spatial relationships to constraint the recognition problem (for example, car above road). Recent research suggests that the object-object based contextual models outperform the scene based contextual models [8]. Our work builds upon this and tries to improve how object-object relationships should be utilized on selected pairs of regions instead of all region pairs.

In most previous work, relationships are represented by graphical models such as belief networks [22] or CRFs [4], and the parameters of the graphical models are learned using graph cuts [23] or max-margin method [24]. One of the common problems with such approaches is determining "what" edges in the graphical model should be used for inference. While fully-connected networks provide the largest number of constraints, they are hard to evaluate and also include weak edges which can sometimes lead to higher belief entropy. Fixed structure approaches, such as neighborhood based MRF's[2], are computationally less demanding but ignore vital long range constraints. Other approaches such as [9] perform approximate inference by selecting fewer edges based on object co-occurrences and discriminability. There has been some work on learning the structure of a graphical model from the training dataset itself [10]. Here, the edges are learned/inserted based on the consistency of relationships throughout the dataset. However, most of the contextual relationships are scene based and might not hold true for all scenarios. In such situations, structure-learning approaches tend to drop the informative edges, since they are not consistent throughout. Instead, we

predict the relevant contextual relationships based on the scene being analyzed. In our approach, instead of learning a fixed structure from the training dataset, we learn the space of allowable structures and then predict a structure for a test image based on its global scene features and local features.

Our work is similar in spirit to "cautious" collective inference [11,12]. Here, instead of using all relationships, the relationships which connect discriminative regions are used for initial iterations and the number of relationships used are increased with each iteration. However, the confidence in the classification of a region is itself a subtle problem and might be scene-dependent. Instead, we learn a decision model for dropping the edges/relationships based on global scene parameters and local parameters. Our work is also related to the feature/kernel weighting problem [13]. However, instead of learning weights of features/kernel for recognition problems, we select features for a constraint satisfaction problem. Therefore, the feature weights are on pairwise features and indicate "how" the edge in a Markov network should be evaluated. This is similar to [1] in which the prior on possible relationships between pairs of nouns is learned, where each relationship is based on one pair-wise feature. However, this approach keeps the priors/weights fixed for a given pair of nouns whereas in our case we learn a scene-dependent weight function.

3 Overview

Given a set of training images with ground truth labeling of segments, our goal is to learn a model which predicts the importance of an edge in a Markov network given the global features of the image and local features of the regions connected by that edge. We also want to learn a model of image and class-specific pairwise feature weights to evaluate contextual edges. Instead of modeling the latent factors and using a parametric approach for computing edge importance and feature-weights, we use a data-driven non-parametric approach to model these. Learning a non-parametric model of edge-importance would require computing edge importance in the ensemble of Markov networks of the set of training images. Edge importance, however, itself-depends upon feature weights; feature weights determine if contextual constraints are satisfied or not. On the other hand, the feature weights, themselves, depend on the structure of the Markov networks in the training dataset, since only the images for which nouns are (finally) linked by an edge should be evaluated to compute the feature weights. We propose an iterative approach to these linked problems. We fix the feature weights to estimate the current edge-importance function, followed by fixing the edge-importance function to re-evaluate feature weights.

Learning. Figure 3 shows an overview of our iterative learning algorithm. Assume that at some iteration, we have some contextual edges in the training data-set and feature weights associated with each contextual edge. For example, in figure 3, out of the six occurrences of road and car, we have contextual edges in five cases with their corresponding weights. Based on the current feature weights, we first estimate how likely each edge satisfies the contextual relationship and its

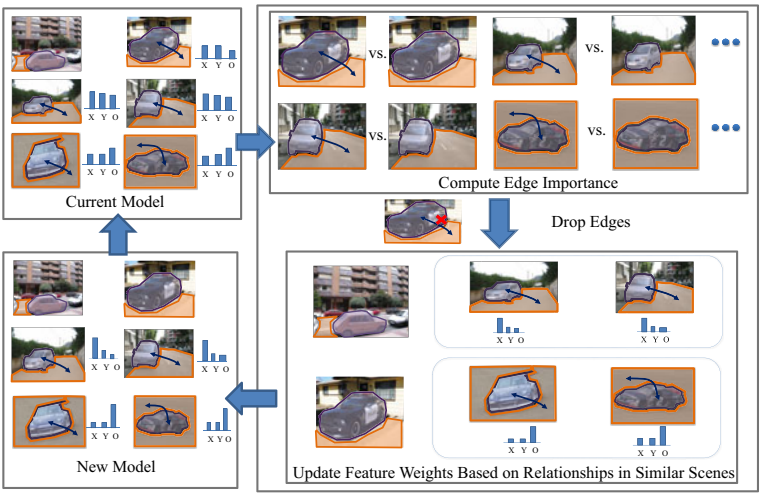


Fig. 3. Overview of our approach: we propose an iterative approach of **what** constitutes the space of important edges and **how** these edges can be evaluated. Our approach simultaneously learn the construction model $\mathcal{F}_e()$ and differential feature weights β

importance in identifying the labels of the regions. To compute the importance, we compare labeling performance with and without the edge in the Markov network. For example, in the first case the relative locations of the car and road are not coherent with other similar examples in the training dataset (the road is neither around/overlapping the car nor is it to the right of the car as in the other cases). Therefore, in this case the edge linking the car and road is not informative and the Markov network without the edge outperforms the Markov network with the edge.

After computing the importance of each edge, a few non-informative edges are eliminated. At this stage, we fix our edge importance function and utilize it to estimate the new pair-wise feature weights. For computing the new feature weights, we retrieve similar examples from the training dataset and analyze which pair-wise features are consistent throughout the set of retrieved samples. The weights of the consistent features are increased accordingly. In the example, we can see that for the images with a top-down viewpoint, the overlap feature becomes important since in the retrieved samples the road region was generally overlapping the car region. Once the feature weights are updated, we obtain a new non-parametric model of both edge-importance and feature weights. This new model is then used to evaluate the edge importance and drop further edges and recompute feature weights.

Inference. The inference procedure is illustrated in Figure 4. An image is first segmented into regions. For segmentation, we use the SWA algorithm [14] and stability analysis for estimating the stable segmentation level [15]. We then predict the importance of each edge based on global features and local features

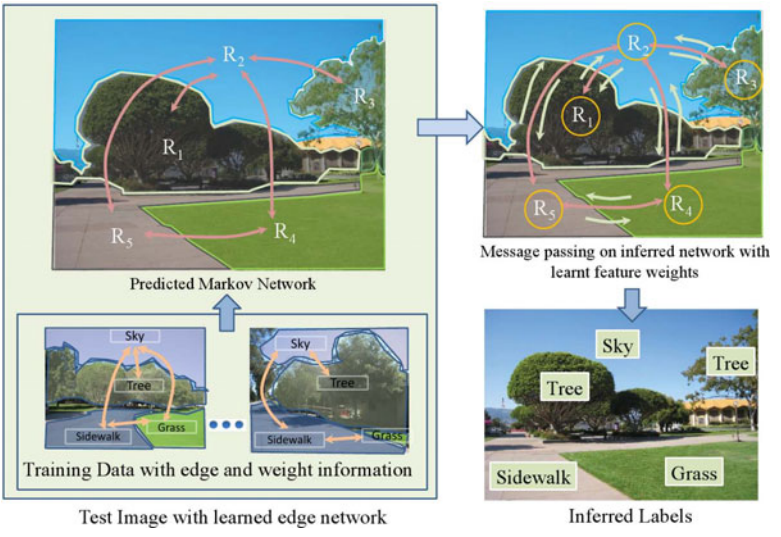


Fig. 4. Inference algorithm for our approach: Using the global and local features computed from the segmentation, we first predict the structure of the Markov network by matching possible edges to edges in the training set (locally weighted regression). We also estimate the feature weights β for each edge in the Markov network. Finally we use message passing to predict the labels.

of the regions connected by the edge. Based on the importance of edges, we construct a Markov network for inference. For each edge, we also compute feature weights that should be utilized to evaluate context on that edge. The labels are then predicted using the message passing algorithm over the constructed Markov network with the estimated feature weights.

4 Mathematical Formulation

We now more formally describe our approach to learn "what" edges constitute the space of efficient networks and "how" to evaluate these edges in those networks. Our motivation is that not all edges in the complete Markov network are informative. So, we want to include in our Markov network only those edges which are generally informative, given the image, and also predict the corresponding feature weights which describe how to evaluate the constraints specified by the selected edges. Formally, our goal is to learn two functions from training data: $\mathcal{F}_e(G^t, R_i^t, R_j^t)$ and $\beta(G^t, n_i^t, n_j^t)$; where $\mathcal{F}_e()$ evaluates whether there should be an edge between regions i and j of image t and $\beta()$ represents the vector of pair-wise feature weights. The function $\mathcal{F}_e()$ depends on the global scene features G^t and the local features of region i and j represented by R_i^t and R_j^t. On the other hand, the feature weights depend on the global scene features and the pair of noun classes for which the pair-wise features are being evaluated.

The functions are learned based on a cost function, which minimizes the cost of labeling in the training dataset. The task is to predict all the nouns in an image, and therefore our cost function can be formulated as follows: Suppose our vocabulary consists of m object-classes, and let y_i^t be an m-dimensional vector which represents the ground-truth annotation for region i in training image t. Function $f_A(R_i)$ evaluates the appearance of the region by comparing it with the appearance models of the m labels and returns an m-dimensional vector with appearance distance scores. The cost function is then formulated as:

$$\mathcal{C} = \sum_t (\sum_i y_i^t f_A(R_i) + \sum_{(j,k)\in\Lambda^t} y_{jk}^t F_C(G^t, R_{jk})) \tag{1}$$

In this cost function, y_{jk}^t is a m^2 dimensional vector which represents pair-wise ground truth annotations and F_C is a m^2 dimensional-vector representing how well the pair-wise features R_{jk} match the contextual parameters for all m^2 pairs of labels. Λ^t represents the set of chosen edges for an image t based on function $\mathcal{F}_e()$ and, therefore, we define Λ^t as: $\{(j,k) : \mathcal{F}_e(G^t, R_i^t, R_j^t) > \alpha\}$.

Contextual evaluation also requires feature-weighting, since all features are not equally important for contextual relationship evaluation. For example, while a difference in y-coordinate is important in evaluation of the contextual relationship between sky and water, the differences in x-coordinate is irrelevant. As discussed previously, these feature weights depend not only on the pair of nouns but also the global features of the scene. Therefore, if the function $f_{n_j,n_k}(G^t, R_{jk})$ represents the $(n_j, n_k)^t h$ element of F_C, we can write it as:

$$f_{n_j,n_k}(G^t, R_{jk}) = \sum_{l=1}^{L} \beta_{n_j,n_k}^l(G^t) C_{n_j,n_k}^l(G^t, R_{jk}^l) \tag{2}$$

where β^l represents the weight of the l^{th} pair-wise feature and is dependent on global scene features and the pair of nouns, and C^l is the context model which measures how well the l^{th} dimension of a pairwise feature R_{jk} satisfies the constraint learned for that dimension for the given pair of nouns.

Intuitively, equation 1 states that the cost can be minimized if: (1) We sum over the contextual constraints that have low cost, that is, Λ^t should only include informative edges. (2) the learned feature weights should be such that the dimensions which represent consistent relationships should have higher weight as compared to the other dimensions. Our goal is to minimize equation (1) with respect to all possible graphs in all training images and all possible weights. At that minima, we have a subset of edges for all the images in the training data-set and feature-weights at each edge. We then learn a non-parametric representation of $\mathcal{F}_e()$ and β based on the importance and weights estimated for the edges in the training dataset. As we can see, the estimation of β in training images depends on edges that are important in the training images and the evaluation of the importance of edges depends on β. Therefore, we employ an iterative approach where we fix β and learn the function $\mathcal{F}_e()$ and in the next step, based on the importance of edges in the training dataset, we re-estimate β.

4.1 Iterative Approach

Learning $\mathcal{F}_e()$. Given feature-weights β, we predict whether an edge is informative or not. The information carried by an edge (representing potential contextual constraints on the pair-wise assignment of nouns to the nodes at the two ends of the edge) is a measure of how important that generic edge type is for inferring the labels associated with the nodes connected by the edge. The information carried in an edge depends on both global and local factors such as viewpoint and discriminability. Instead, of discovering all the factors and then learning a parametric-function; we use a non-parametric representation of the importance function. However, we still need to compute the importance of each edge in the training data-set.

To compute the importance of an edge, we use the message-passing algorithm. The importance of an edge is defined as how much the message passing through the edge helps in bringing the belief of nodes connected by the edge towards their goal belief (ground-truth). Suppose that the goal beliefs at node i and j are y_i and y_j respectively. The importance of the edge between i to j is defined as:

$$I(i \leftrightarrow j) = \frac{1}{iter} \sum_{k=1}^{iter} (y_i.b^k_{\mathcal{N}_i} - y_i.b^k_{\mathcal{N}_{i-(i,j)}}) + (y_j.b^k_{\mathcal{N}_j} - y_j.b^k_{\mathcal{N}_{j-(i,j)}}) \quad (3)$$

where $b^k_{\mathcal{N}_i}$ is the belief at node i at iteration k computed using messages from all the nodes (fully-connected setting); $b^k_{\mathcal{N}_{i-(i,j)}}$ is the belief at node i at iteration k computed using messages from all the nodes except $i \leftrightarrow j$ (edge-dropped setting). *iter* is the total number of iterations of message passing algorithm.

Using this approach, the importance of each edge is computed based on the local message passing algorithm. It does not take into account the behavior of other similar edges (similar global scene features and connecting similar local regions) in the training dataset. For example, in a particular image from the set of beach scenes, the edge between sky and water might not be important; however if it is important in most other beach images, we want to increase the importance of that particular edge so that it is not dropped. We therefore update the importance of an edge by using the importance of the edges which have similar global and local features. This is followed by an edge dropping step, where the edges with low importance are dropped randomly to compute an efficient and accurate networks for the training images.

Learning β. Given the importance function of the edges $\mathcal{F}_e()$, we estimate β. As stated above, we use locally weighted regression for estimating β, therefore we need to estimate individual feature weights for all edges. Given the cost function in equation 1, a gradient descent approach is employed to estimate the feature weights of edges. We obtain the gradient as:

$$\frac{\partial \mathcal{C}}{\partial \beta^l_{n_j,n_k}} = C^l(G^t, R_{jk}) \quad (4)$$

where $\beta^l_{n_j,n_k}$ is the weight of l^{th} feature for edge (j,k). The above equation states that for a given pair of nouns, if the l^{th} dimension of pairwise feature is

consistent with the l^{th} dimension of pairwise features from similar images, then the value of $\beta^l_{n_j,n_k}$ should be increased. Therefore, the value of β is updated at each step using the gradient above and then normalized ($\sum_l \beta^l = 1$). Intuitively, this equation evaluates which contextual relationship is satisfied on average for a pair of nouns and increases it weight. For example, between sky and water the above relationship is always satisfied where as left/right has high variance (In images sky is sometimes on left and sometimes on right of water). Therefore, this equation increases the weight of dY (measuring above) and decreases the weight of dX (measuring left).

4.2 Inference

Given a segmentation of an image, we construct a Markov network for the image using the function $\mathcal{F}_e()$. For this construction, we first compute the global features, G, of the image and the local features of every region in the segmentation. A potential edge in the network is then predicted using simple locally weighted regression:

$$\mathcal{F}_e(G, R_j, R_k) = \sum_{t,j_t,k_t} W(G, G^t, R_j, R_{j_t}, R_k, R_{k_t}) M(j_t \leftrightarrow k_t) \tag{5}$$

where $W()$ is the weight function based on distances between local and global features of training and test data and $M()$ is an indicator function which predicts whether the edge was retained in the training data or not. The feature weights are also computed using locally weighted regression. The labels are then predicted using the message passing algorithm over the constructed Markov network with the estimated feature weights.

5 Experimental Results

We describe the experiments conducted on a subset of the LabelMe [19] dataset. We randomly selected 350 images from LabelMe and divided the set into 250 training and 100 test images. Our training data-set consists of images with segmentations and labels provided[1]. We used GIST features [18] as global features for scene matching. For appearance modeling, we use Hoeim's features [17] together with class specific metric learning used by [20]. The pairwise relation feature vocabulary consists of 5 contextual relationships[2]. In all experiments, we compare the performance of our approach to a fully-connected Markov network and a neighborhood based Markov network. We measure the performance of our annotation result as the number of pixels correctly labeled divided by total number of pixels in the image, averaged over all images.

In the training phase, we run inference on each training image and utilize the ground truth to evaluate the importance of each edge. At each iteration, a few unimportant edges are dropped and feature weights are re-estimated. Figure 6 (a)

[1] Grass, tree, field, building, rock, water, road, sky, person, car, sign, mountain, ground, sand, bison, snow, boat, airplane, sidewalk

[2] Contextual relations - above/below, left/right, greener, bluer, brighter

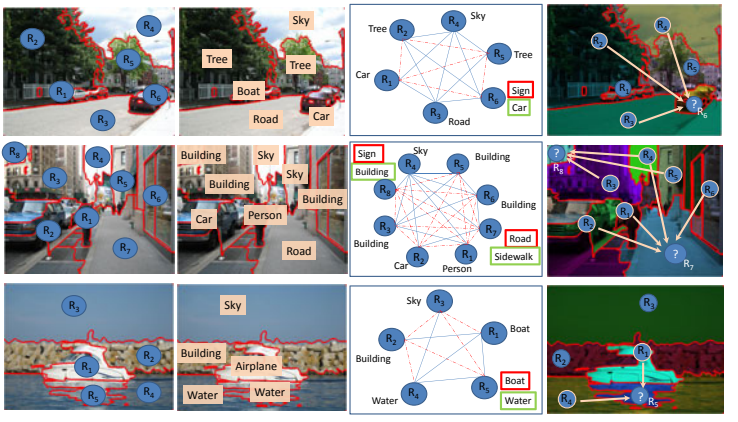


Fig. 5. A few qualitative examples from the LabelMe dataset of how constructing the network structure using our approach leads to an efficient Markov structure which also improves labeling performance

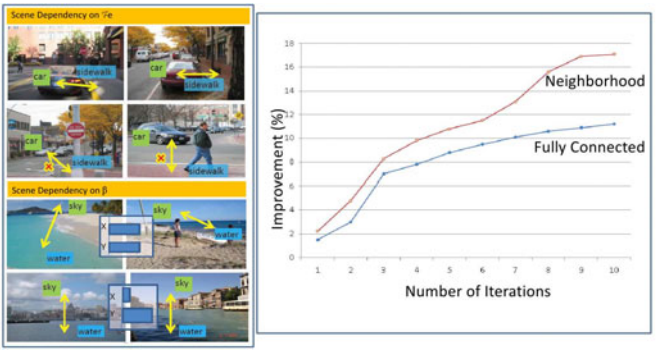


Fig. 6. (a) Scene dependency of $\mathcal{F}_e()$ and β. In the first case the edge between sidewalk and road is informative only when the car is parked or is nearby it. In the second case, in scenes like beaches, both x and y are important features of context; however when the viewpoint is such that water occupies most of the lower space, the importance of x decreases. (b) The graphs show the % improvement over the fully-connected and neighborhood based Markov network as the training continues.

show examples of how our approach captures the scene dependency of $\mathcal{F}_e()$ and β respectively. Fig 6 (b) shows the percentage improvement over a fully-connected network and a neighborhood based network with each iteration. The figure clearly shows that dropping edges not only provides computational efficiency, but also improves the matching scores in training due to the removal of spurious constraints or constraints which link regions which are not discriminative.

On test images, we first predict the Markov network using the learned $\mathcal{F}_e()$ and then utilize β to perform inference. Figure 7 show the performance of our approach compared to a fully-connected and a neighborhood connected

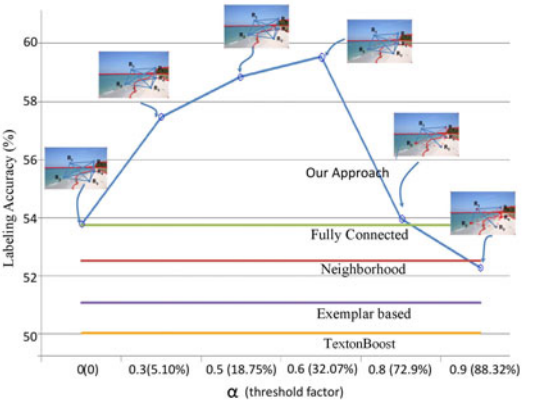


Fig. 7. The graph shows the improvement of our algorithm over the fully-connected network and neighborhood based network on the LabelMe dataset with an example of graph structures at different thresholds of $\mathcal{F}_e()$. The values in the parentheses shows the percentage of edges dropped at a given threshold of $\mathcal{F}_e()$

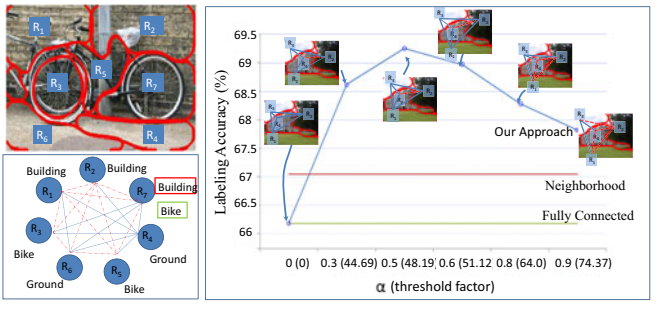


Fig. 8. (a) An example where our approach removed spurious edges and improved labeling performance. (b) Labeling accuracy of our algorithm compared to fully-connected and neighborhood connected Markov networks on the MSRC dataset with examples of graph structures at different thresholds of $\mathcal{F}_e()$.

Markov network on the LabelMe dataset at different thresholds of $\mathcal{F}_e()$. A higher threshold corresponds to dropping more edges. The values in parenthesis on the threshold axis shows the average percentage of edges dropped at that particular threshold. We also compared the performance of our approach to publicly available version of texton-boost (without CRF) on our LabelMe dataset and it yields approximately 50% as compared to 59% by our approach. It should be noted that this is similar to the performance of the local-appearance based model used in our approach. Therefore, our approach should also provide considerable improvement over the CRF version of the texton-boost as well. Above the performance chart, we show "what" edges are dropped at a given threshold. We also compare our approach to the exemplar based approach similar to [21] where the labels are transferred based on the edge matches in the training dataset.

Figure 5 shows representative examples where our approach performed better than the fully-connected network. The second column of the figure shows the result obtained using just the appearance model (likelihood term). The third column shows our network compared to a fully-connected Markov network. The label marked in red is the result obtained using the fully-connected network while the label in green is the result obtained using our approach. In the last column, we show the regions of interest (where our approach dropped spurious edges which led to improvement in performance). In the first example, if a fully-connected network is utilized, the label of the red car on the right side of road is forced to signboard (by other car). This is because the car-signboard relationship is stronger than car-car relation in the current spatial configuration and the bad appearance model predicts signboard as a more likely label. On the other hand, when the spurious edge is dropped, the labeling improves and the region is correctly labeled as a car. Similarly in the second example, we observe that the sidewalk is labeled as road in the fully-connected network (due to strong appearance likelihood and presence of buildings). On the other hand, the region labeled as person boosts the presence of sidewalk (people walk on sidewalks) and when spurious edges from buildings are dropped by our approach the labeling improves and the region is correctly labeled as sidewalk.

We additionally tested our algorithm on the MSRC dataset. The training and testing data of the MSRC dataset is the same as in [16]. The dataset has 21 object classes. It should be noted that MSRC is not an ideal dataset since the number of regions per image is very low and therefore there are not many spurious edges that can be dropped. However, this experiment is performed in order to compare the performance of our baseline to other state-of-the-art approaches. Figure 8(a) shows the performance of our algorithm compared to fully-connected and neighborhood connected networks on the MSRC dataset. Our results are comparable to the state of the art approaches based on image segmentation such as [16]. Figure 8(b) shows an example of one case where dubious information is passed along edges in the fully-connected network leading to wrong labeling. Region 7, in the fully connected network, was labeled as building. The is because the building-building and bike-building contextual relationship is stronger than bike-bike relationship. But when the link of the bike region with regions labeled as building was removed through our edge prediction, it was correctly labeled as bike.

Acknowledgement. This research is supported by ONR grant N000141010766. This material is based upon work supported by, or in part by, the U. S. Army Research Laboratory and the U. S. Army Research Office under grant number W911NF-09-1-0383. The authors would also like to thank Alexei Efros, Martial Hebert and Tomasz Malisiewicz for useful discussions on the paper.

References

1. Gupta, A., Davis, L.: Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 16–29. Springer, Heidelberg (2008)

2. Carbonetto, P., Freitas, N., Barnard, K.: A statistical model for general contextual object recognition. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 350–362. Springer, Heidelberg (2004)
3. Divvala, S., Hoiem, D., Hays, J., Efros, A.A., Hebert, M.: An Empirical Study of Context in Object Detection. In: CVPR 2009 (2009)
4. Galleguillos, C., Rabinovich, A., Belongie, S.: Object Categorization using Co-Occurrence, Location and Appearance. In: CVPR 2008 (2008)
5. Li, J., Fei-Fei, L.: What, where and who? Classifying event by scene and object recognition. In: ICCV 2007 (2007)
6. He, X., Zemel, R.: Latent topic random fields: Learning using a taxonomy of labels. In: CVPR 2008 (2008)
7. Murphy, K., Torralba, A., Freeman, W.: Using the Forest to See the Trees:A Graphical Model Relating Features, Objects and Scenes. In: NIPS 2003 (2003)
8. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in Context. In: ICCV 2007 (2007)
9. Torralba, A., Murphy, K.P., Freeman, W.T.: Contextual Models for Object Detection using Boosted Random Fields. In: Adv. in Neural Information Processing Systems (NIPS), pp. 1401–1408 (2005)
10. Friedman, N.: The Bayesian structural EM algorithm. In: UAI 1998 (1998)
11. McDowell, L.K., Gupta, K., Aha, D.: Cautious Inference in Collective Classification. In: AAAI 2007 (2007)
12. Neville, J., Jensen, D.: Iterative Classification in Relational Data. In: AAAI 2000 Workshop on Learning Statistical Models from Relational Data (2000)
13. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: More Efficiency in Multiple Kernel Learning. In: ICML 2007 (2007)
14. Galun, M., Sharon, E., Basri, R., Brandt, A.: Texture segmentation by multiscale aggregation of filter responses and shape elements. In: ICCV (2003)
15. Rabinovich, A., Lange, T., Buhmann, J., Belongie, S.: Model Order Selection and Cue Combination for Image Segmentation. In: CVPR 2006 (2006)
16. Shotton, J., Johnson, M., Cipolla, R.: Semantic Texton Forests for Image Categorization and Segmentation. In: CVPR 2008 (2008)
17. Hoiem, D., Efros, A.A., Hebert, M.: Geometric Context from a Single Image. In: ICCV 2005 (2005)
18. Oliva, A., Torralba, A.: Building the Gist of a Scene: The Role of Global Image Features in Recognition. In: Visual Perception 2006 (2006)
19. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. In: IJCV 2008 (2008)
20. Jain, P., Kapoor, A.: Probabilistic Nearest Neighbor Classifier with Active Learning, http://www.cs.utexas.edu/users/pjain/pknn/
21. Malisiewicz, T., Efros, A.: Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships. In: NIPS (2009)
22. Gupta, A., Davis, L.S.: Objects in Action:An Approach for Combining Action Understanding and Object Perception. In: CVPR 2007 (2007)
23. 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)
24. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., Singer, Y.: Large margin methods for structured and interdependent output variables. JMLR 6, 1453–1484 (2005)

Adapting Visual Category Models to New Domains

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell

UC Berkeley EECS and ICSI, Berkeley, CA
{saenko,kulis,mfritz,trevor}@eecs.berkeley.edu

Abstract. Domain adaptation is an important emerging topic in computer vision. In this paper, we present one of the first studies of domain shift in the context of object recognition. We introduce a method that adapts object models acquired in a particular *visual domain* to new imaging conditions by learning a transformation that minimizes the effect of domain-induced changes in the feature distribution. The transformation is learned in a supervised manner and can be applied to categories for which there are no labeled examples in the new domain. While we focus our evaluation on object recognition tasks, the transform-based adaptation technique we develop is general and could be applied to non-image data. Another contribution is a new multi-domain object database, freely available for download. We experimentally demonstrate the ability of our method to improve recognition on categories with few or no target domain labels and moderate to large changes in the imaging conditions.

1 Introduction

Supervised classification methods, such as kernel-based and nearest-neighbor classifiers, have been shown to perform very well on standard object recognition tasks (e.g. [4], [17], [3]). However, many such methods expect the test images to come from the same distribution as the training images, and often fail when presented with a novel *visual domain*. While the problem of *domain adaptation* has received significant recent attention in the natural language processing community, it has been largely overlooked in the object recognition field. In this paper, we explore the issue of domain shift in the context of object recognition, and present a novel method that adapts existing classifiers to new domains where labeled data is scarce.

Often, we wish to perform recognition in a *target* visual domain where we have very few labeled examples and/or only have labels for a subset of categories, but have access to a *source* domain with plenty of labeled examples in many categories. As Figure 1 shows, it is insufficient to directly use object classifiers trained on the source domain, as their performance can degrade significantly on the target domain. Even when the same features are extracted in both domains, and the necessary normalization is performed on the image and the feature vectors, the underlying cause of the domain shift can strongly affect the feature distribution and thus violate the assumptions of the classifier. Typical

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 213–226, 2010.

source domain

target domain

(a)

train	test	SVM-bow	NBNN [3]
source	source	54 ± 2	61 ± 1
source	target	20 ± 1	19 ± 1

(b)

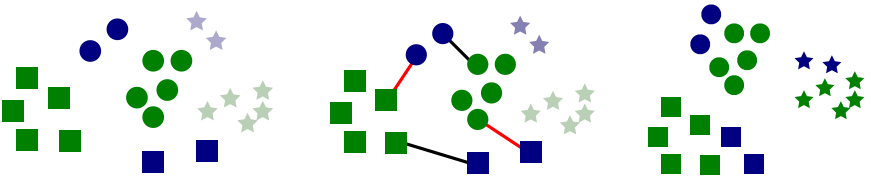
Fig. 1. (a) Example of extreme visual domain shift. (b) Degradation of the performance of two object classification methods (an SVM over a bag-of-words representation (SVM-bow) and the Naive Bayes nearest neighbor (NBNN) classifier of [3]) when trained and tested on these image domains (see Sec.4 for dataset descriptions). Classification accuracy is averaged over 31 object categories, and over 5 random 80%-20% splits into train/test data.

causes of visual domain shift include changes in the camera, image resolution, lighting, background, viewpoint, and post-processing. In the extreme case, all of these changes take place, such as when shifting from typical object category datasets mined from internet search engines to images captured in real-world surroundings, e.g. by a mobile robot (see Figure 1).

Recently, domain adaptation methods that attempt to transfer classifiers learned on a source domain to new domains have been proposed in the language community. For example, Blitzer et al. adapt sentiment classifiers learned on book reviews to electronics and kitchen appliances [2]. In this paper, we argue that addressing the problem of domain adaptation for object recognition is essential for two reasons: 1) while labeled datasets are becoming larger and more available, they still differ significantly from many interesting application domains, and 2) it is unrealistic to expect the user to collect many labels in each new domain, especially when one considers the large number of possible object categories. Therefore, we need methods that can transfer object category knowledge from large labeled datasets to new domains.

In this paper, we introduce a novel domain adaptation technique based on cross-domain transformations. The key idea, illustrated in Figure 2, is to learn a regularized non-linear transformation that maps points in the source domain (green) closer to those in the target domain (blue), using supervised data from both domains. The input consists of labeled pairs of inter-domain examples that are known to be either similar (black lines) or dissimilar (red lines). The output is the learned transformation, which can be applied to previously unseen test data points. One of the key advantages of our transform-based approach is that it can be applied over novel test samples from categories seen at training time, and can also generalize to new categories which were not present at training time.

We develop a general framework for learning regularized cross-domain transformations, and then present an algorithm based on a specific regularizer which results in a symmetric transform. This special case of transformations has



(a) Domain shift problem (b) Pairwise constraints (c) Invariant space

Fig. 2. The key idea of our approach to domain adaptation is to learn a transformation that compensates for the domain-induced changes. By leveraging (dis)similarity constraints (b) we aim to reunite samples from two different domains (blue and green) in a common invariant space (c) in order to learn and classify new samples more effectively across domains. The transformation can also be applied to new categories (lightly-shaded stars). This figure is best viewed in color.

previously been explored for metric learning, and we base the algorithm presented in this paper on the information theoretic metric learning method of [8]. Metric learning has been successfully applied to a variety of problems in vision and other domains (see [6,11,14] for some vision examples) but to our knowledge has not been applied to domain adaptation. In work subsequent to that reported in this paper, we have developed a variant of our method that learns regularized asymmetric transformations, which allows us to model more general types of domain shift[1].

Rather than committing to a specific form of the classifier, we only assume that it operates over (kernelized) distances between examples. Encoding the domain invariance into the feature representation allows our method to benefit a broad range of classification methods, from k-NN to SVM, as well as clustering methods. While we evaluate our technique on object recognition, it is a general adaptation method that could be applied to non-image data.

In the next section, we relate our approach to existing work on domain adaptation and transfer learning. Section 3 describes our general framework for domain adaptation and presents an algorithm based on symmetric transformations, i.e. metric learning. We evaluate our approach on a new dataset designed to study the problem of visual domain shift, which is described in Section 4, and show empirical results of object classifier adaptation on several visual domains in Section 5.

2 Related Work

The domain adaptation problem has recently started to gain attention in the natural language community. Daume III [7] proposed a domain adaptation approach that works by transforming the features into an augmented space, where the input features from each domain are copied twice, once to a domain-independent portion of the feature vector, and once to the portion specific to that domain.

[1] See the technical report [15] for details of the method; for comparison results using this method are shown in the tables below.

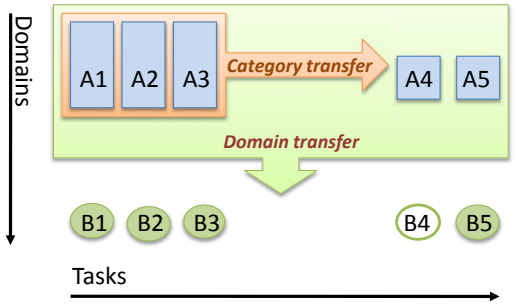


Fig. 3. Unlike category transfer methods, our method does not transfer structure between related tasks, but rather transfers the learned structure of the domain shift from tasks labeled in both domains (e.g. tasks 1,2,3 and 5 in the figure) to tasks unlabeled in the target domain (e.g. task 4), without requiring these tasks to be related.

The portion specific to all other domains is set to zeros. While "frustratingly" easy to implement, this approach only works for classifiers that learn a function over the features. With normalized features (as in our experimental results), the nearest neighbor classifier results are unchanged after adaptation. Structural correspondence learning is another method proposed for NLP tasks such as sentiment classification [2]. However, it is targeted towards language domains, and relies heavily on the selection of *pivot* features, which are words that frequently occur in both domains (e.g. "wonderful", "awful") and are correlated with domain-specific words.

Recently, several adaptation methods for the support vector machine (SVM) classifier have been proposed in the video retrieval literature. Yang et al. [18] proposed an Adaptive SVM (A-SVM) which adjusts the existing classifier $f^s(x)$ trained on the source domain to obtain a new SVM classifier $f^t(x)$. Cross-domain SVM (CD-SVM) proposed by Jiang et al. [13] defines a weight for each source training sample based on distance to the target domain, and re-trains the SVM classifier with re-weighted patterns. The domain transfer SVM (DT-SVM) proposed by Duan et al. [9] used multiple-kernel learning to minimize the difference between the means of the source and target feature distributions. These methods are specific to the SVM classifier, and they require target-domain labels for all categories. The advantage of our method is that it can perform transfer of domain-invariant representations to *novel* categories, with no target-domain labels, and can be applied to a variety of classifiers and clustering techniques.

Our approach can be thought of as a form of knowledge transfer from the source to the target domain. However, in contrast to many existing transfer learning paradigms (e.g. [16], [10], [12]), we do not presume any degree of relatedness between the categories that are used to learn the transferred structure and the categories to which the structure is transferred (see Figure 3).

Individual categories are related across domains, of course; the key point is that we are transferring the structure of the domain shift, not transferring structures common to related categories.

Finally, metric and similarity learning has been successfully applied to a variety of problems in vision and other domains (see [6,11,14,5] for some vision examples) but to our knowledge has not been used for domain adaptation.

3 Domain Adaptation Using Regularized Cross-Domain Transforms

We begin by describing our general domain adaptation model in the linear setting, then, in Section 3.1, show how both the linear and the kernelized version of the particular case of a symmetric transform used in our experiments can be implemented using the metric learning approach of [8].

In the following, we assume that there are two domains $\mathcal{A}$ and $\mathcal{B}$ (e.g., source and target). Given vectors $\boldsymbol{x} \in \mathcal{A}$ and $\boldsymbol{y} \in \mathcal{B}$, we propose to learn a linear transformation W from $\mathcal{B}$ to $\mathcal{A}$ (or equivalently, a transformation W^T to transform from $\mathcal{A}$ to $\mathcal{B}$). If the dimensionality of the vectors $\boldsymbol{x} \in \mathcal{A}$ is d_A and the dimensionality of the vectors $\boldsymbol{y} \in \mathcal{B}$ is d_B, then the size of the matrix W is $d_A \times d_B$. We denote the resulting inner product similarity function between $\boldsymbol{x}$ and the transformed $\boldsymbol{y}$ as

$$\text{sim}_W(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T W \boldsymbol{y}.$$

The goal is to learn the linear transformation given some form of supervision, and then to utilize the learned similarity function in a classification or clustering algorithm. To avoid overfitting, we choose a regularization function for W, which we will denote as $r(W)$ (choices of the regularizer are discussed below). Denote $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_A}]$ as the matrix of n_A training data points (of dimensionality d_A) from $\mathcal{A}$ and $Y = [\boldsymbol{y}_1, ..., \boldsymbol{y}_{n_B}]$ as the matrix of n_B training data points (of dimensionality d_B) from $\mathcal{B}$. We will discuss the exact form of supervision we propose for domain adaptation problems in Section 3.1, but for now assume that it is a function of the learned similarity values $\text{sim}_W(\boldsymbol{x}, \boldsymbol{y})$ (i.e., a function of the matrix $X^T W Y$), so a general optimization problem would seek to minimize the regularizer subject to supervision constraints given by functions c_i:

$$\begin{array}{ll} \min_W & r(W) \\ \text{s.t.} & c_i(X^T W Y) \geq 0, \quad 1 \leq i \leq c. \end{array} \tag{1}$$

Due to the potential of infeasibility, we can introduce slack variables into the above formulation, or write the problem as an unconstrained problem:

$$\min_W \; r(W) + \lambda \sum_i c_i(X^T W Y).$$

In this paper, we focus on a special case of this general transformation learning problem, one that employs a particular regularizer and constraints that are a function of the learned *distances*[2]

$$\mathrm{d}_W(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} - \boldsymbol{y})^T W (\boldsymbol{x} - \boldsymbol{y}).$$

The regularizer we consider here is $r(W) = \mathrm{tr}(W) - \log\det(W)$. Note that this regularizer can only be applied when the dimensionalities of the two domains are equal ($d_A = d_B$). This choice of regularizer and constraints has previously been studied as a Mahalanobis metric learning method, and is called *information-theoretic metric learning* (ITML) [14]; we stress, however, that the use of such a regularizer for domain adaptation is novel, as is our method for constructing cross-domain constraints, which we discuss in Section 3.1. We call this approach *symm* for short, since the learned transformation W is always symmetric positive definite.

The fact that W is required to be symmetric may be overly restrictive for some applications. We refer the reader to [15], where we develop an asymmetric version of our domain adaptation model with the regularizer $r(W) = \frac{1}{2}\|W\|_F^2$ and constraints that are functions of learned similarities $\mathrm{sim}_W(\boldsymbol{x}, \boldsymbol{y})$. This method, called *asymm* for short in this paper, can also handle the case when $d_A \neq d_B$.

3.1 Domain Adaptation Using Metric Learning

In this section, we describe our specific algorithm in detail. In using symmetric positive definite matrices, the idea is that the shift can be approximated as an arbitrary linear scaling and rotation of the feature space. We aim to recover this transformation by leveraging labeled data consisting of similarity and dissimilarity constraints between points in the two domains. Since the matrix W corresponding to the metric is symmetric positive semi-definite, we can think of it as mapping samples coming from two different domains into a common invariant space, in order to learn and classify instances more effectively across domains. Note that by factorizing W as $W = G^T G$, we can equivalently view the distance d_W between points $\boldsymbol{x}$ and $\boldsymbol{y}$ as $(G\boldsymbol{x} - G\boldsymbol{y})^T(G\boldsymbol{x} - G\boldsymbol{y})$; that is, the distance is simply the squared Euclidean distance after applying the linear transformation specified by G. The transformation G therefore maps data points from both domains into an invariant space. Because a linear transformation may not be sufficient, we optionally kernelize the distance matrix to learn non-linear transformations.

Generating Cross-Domain Constraints. Suppose that we want to recognize a total of n categories (tasks), with training data from category i denoted as $\mathcal{L}_i$ and consisting of $(\boldsymbol{x}, l)$ pairs of input image observations $\boldsymbol{x}$ and category labels l. There are two cases that we consider. In the first case, we have many

[2] Mathematically, to ensure that such constraints are a function of $X^T W Y$, we let $X = Y$ be the concatenation of data points in both domains. This is possible since the dimensionalities of the domains are identical.

labeled examples for each of the n categories in the source domain data, $\mathcal{L}^A = \mathcal{L}_1^A \cup ... \cup \mathcal{L}_n^A$, and a few labeled examples for each category in the target domain data, $\mathcal{L}^B = \mathcal{L}_1^B \cup ... \cup \mathcal{L}_n^B$. In the second case, we have the same training data $\mathcal{L}^A$, but only have labels for a subset of the categories in the target domain, $\mathcal{L}^B = \mathcal{L}_1^B \cup ... \cup \mathcal{L}_m^B$, where $m < n$. Here, our goal is to adapt the classifiers trained on tasks $m+1, ..., n$, which only have source domain labels, to obtain new classifiers that reduce the predictive error on the target domain by accounting for the domain shift. We accomplish this by applying the transformation learned on the first m categories to the features in the source domain training set of categories $m+1, ..., n$, and re-training the classifier.

To generate the similarity and dissimilarity constraints necessary to learn the domain-invariant transformation, we use the following procedure. We sample a random pair consisting of a labeled source domain sample $(\boldsymbol{x}_{\boldsymbol{i}}^A, l_i^A)$ and a labeled target domain sample $(\boldsymbol{x}_{\boldsymbol{j}}^B, l_j^B)$, and create a constraint

$$\begin{aligned} d_W(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B) &\leq u \quad \text{if} \quad l_i = l_j, \\ d_W(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B) &\geq \ell \quad \text{if} \quad l_i \neq l_j. \end{aligned} \tag{2}$$

We call these *class-based* constraints, and we use this procedure to construct a set $\mathcal{S}$ of pairs (i, j) of similarity constraints and $\mathcal{D}$ of dissimilarity constraints. Alternatively, we can generate constraints based not on class labels, but on information of the form "target domain sample $\boldsymbol{x}_i^A$ is similar to source domain sample $\boldsymbol{x}_j^B$". This is particularly useful when the source and target data include images of the same object, as it allows us to best recover the structure of the domain shift, without learning anything about particular categories. We refer to these as *correspondence* constraints.

It is important to generate constraints between samples of different domains, as including same-domain constraints can make it difficult for the algorithm to learn the domain shift. In fact, we show experimentally that creating constraints based on class labels without regard for domain boundaries, in the style of metric learning, does considerably worse than our method.

Learning W using ITML. As mentioned above, information-theoretic metric learning (ITML) formulates the problem as follows:

$$\begin{aligned} \min_{W \succeq 0} \quad & \mathrm{tr}(W) - \log \det W \\ \text{s. t.} \quad & d_W(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B) \leq u \quad (i,j) \in \mathcal{S}, \\ & d_W(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B) \geq \ell \quad (i,j) \in \mathcal{D}, \end{aligned} \tag{3}$$

where the regularizer $\mathrm{tr}(W) - \log \det W$ is defined only between positive semi-definite matrices. This regularizer is a special case of the *LogDet divergence*, which has many properties desirable for metric learning such as scale and rotation invariance [8]. Note that one typically adds slack variables, governed by a tradeoff parameter λ, to the above formulation to ensure that a feasible solution can always be found.

We follow the approach given in [8] to find the optimal W for (3). At each step of the algorithm, a single pair $(\boldsymbol{x}_i^A, \boldsymbol{x}_j^B)$ from $\mathcal{S}$ or $\mathcal{D}$ is chosen, and an update of the form

$$W_{t+1} = W_t + \beta_t W_t(\boldsymbol{x}_i^A - \boldsymbol{x}_j^B)(\boldsymbol{x}_i^A - \boldsymbol{x}_j^B)^T W_t$$

is applied. In the above, β_t is a scalar parameter computed by the algorithm based on the type of constraint and the amount of violation of the constraint. Such updates are repeated until reaching global convergence; typically we choose the most violated constraint at every iteration and stop when all constraints are satisfied up to some tolerance ϵ.

In some cases, the dimensionality of the data is very high, or a linear transformation is not sufficient for the desired metric. In such cases, we can apply *kernelization* to the above algorithm in order to learn high-dimensional metrics and/or non-linear transformations. Let $\bar{X} = [X\ Y]$ be the concatenated matrix of data points from both domains. It is possible to show that the updates for ITML may be written in terms of the kernel matrix by multiplying the updates on the left by $\bar{X}^T$ and on the right by $\bar{X}$, yielding

$$K_{t+1} = K_t + \beta_t K_t(\boldsymbol{e}_i^A - \boldsymbol{e}_j^B)(\boldsymbol{e}_i^A - \boldsymbol{e}_j^B)^T K_t,$$

where $\boldsymbol{e}_i^A$ is the standard basis vector corresponding to the index of $\boldsymbol{x}_i^A$ and $K_t = \bar{X}^T W_t \bar{X}$. $K_0 = \bar{X}^T \bar{X}$ corresponds to some kernel matrix over the concatenated input data when we map data points from both domains to a high-dimensional feature space. Furthermore, the learned kernel function may be computed over arbtirary points, and the method may be scaled for very large data sets; see [8,14] for details.

4 A Database for Studying Effects of Domain Shift in Object Recognition

As detailed earlier, effects of domain shift have been largely overlooked in previous object recognition studies. Therefore, one of the contributions of this paper is a database[3] that allows researchers to study, evaluate and compare solutions to the domain shift problem by establishing a multiple-domain labeled dataset and benchmark. In addition to the domain shift aspects, this database also proposes a challenging office environment category learning task which reflects the difficulty of real-world indoor robotic object recognition, and may serve as a useful testbed for such tasks. It contains a total of 4652 images originating from the following three domains:

Images from the web: The first domain consists of images from the web downloaded from online merchants (www.amazon.com). This has become a very popular way to acquire data, as it allows for easy access to large amounts of data that lends itself to learning category models. These images are of products shot at medium resolution typically taken in an environment with studio lighting

[3] Available at http://www.eecs.berkeley.edu/~mfritz/domainadaptation/.

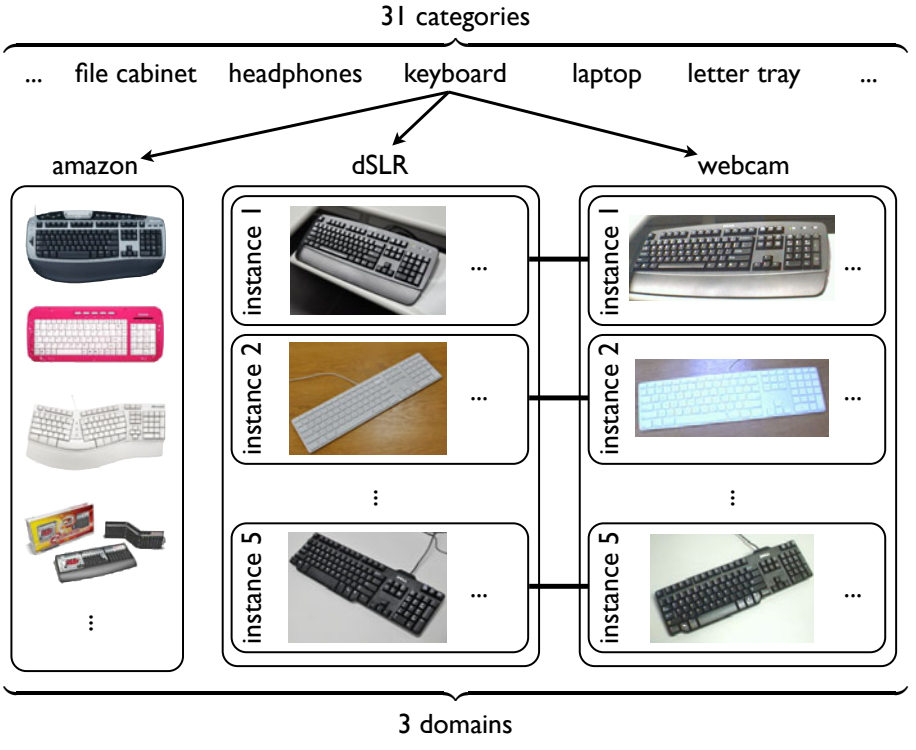


Fig. 4. New dataset for investigating domain shifts in visual category recognition tasks. Images of objects from 31 categories are downloaded from the web as well as captured by a high definition and a low definition camera.

conditions. We collected two datasets: *amazon* contains 31 categories[4] with an average of 90 images each. The images capture the large intra-class variation of these categories, but typically show the objects only from a canonical viewpoint. *amazonINS* contains 17 object *instances* (e.g. can of *Taster's Choice* instant coffee) with an average of two images each.

Images from a digital SLR camera: The second domain consists of images that are captured with a digital SLR camera in realistic environments with natural lighting conditions. The images have high resolution (4288x2848) and low noise. We have recorded two datasets: *dslr* has images of the 31 object categories, with 5 different objects for each, in an office environment. Each object was captured with on average 3 images taken from different viewpoints, for a

[4] The 31 categories in the database are: backpack, bike, bike helmet, bookcase, bottle, calculator, desk chair, desk lamp, computer, file cabinet, headphones, keyboard, laptop, letter tray, mobile phone, monitor, mouse, mug, notebook, pen, phone, printer, projector, puncher, ring binder, ruler, scissors, speaker, stapler, tape, and trash can.

total of 423 images. *dslrINS* contains 534 images of the 17 object instances, with an average of 30 images per instance, taken in a home environment.

Images from a webcam: The third domain consists of images of the 31 categories recorded with a simple webcam. The images are of low resolution (640x480) and show significant noise and color as well as white balance artifacts. Many current imagers on robotic platforms share a similarly-sized sensor, and therefore also possess these sensing characteristics. The resulting *webcam* dataset contains the same 5 objects per category as in *dSLR*, for a total of 795 images.

The database represents several interesting visual domain shifts. First of all, it allows us to investigate the adaptation of category models learned on the web to dSLR and webcam images, which can be thought of as in situ observations on a robotic platform in a realistic office or home environment. Second, domain transfer between the high-quality dSLR images to low-resolution webcam images allows for a very controlled investigation of category model adaptation, as the same objects were recorded in both domains. Finally, the amazonINS and dslrINS datasets allow us to evaluate adaptation of product instance models from web data to a user environment, in a setting where images of the same products are available in both domains.

5 Experiments

In this section, we evaluate our domain adaptation approach by applying it to k-nearest neighbor classification of object categories and instances. We use the database described in the previous section to study different types of domain shifts and compare our new approach to several baseline methods. First, we detail our image processing pipeline, and then describe the different experimental settings and elaborate on our empirical findings.

Image Processing: All images were resized to the same width and converted to grayscale. Local scale-invariant interest points were detected using the SURF [1] detector to describe the image. SURF features have been shown to be highly repeatable and robust to noise, displacement, geometric and photometric transformations. The blob response threshold was set to 1000, and the other parameters to default values. A 64-dimensional non-rotationally invariant SURF descriptor was used to describe the patch surrounding each detected interest point. After extracting a set of SURF descriptors for each image, vector quantization into visual words was performed to generate the final feature vector. A codebook of size 800 was constructed by k-means clustering on a randomly chosen subset of the amazon database. All images were converted to histograms over the resulting visual words. No spatial or color information was included in the image representation for these experiments.

In the following, we compare k-NN classifiers that use the proposed cross-domain transformation to the following baselines: 1) k-NN classifiers that operate in the original feature space using a Euclidean distance, and 2) k-NN classifiers

Table 1. Domain adaptation results for categories seen during training in the target domain

		No shift	Baseline Methods				Our Method	
domain A	domain B	knnAA	knnAB	knnBB	ITML(A+B)	ITML(B)	asymm	symm
webcam	dslr	0.34	0.14	0.20	0.18	0.23	0.25	**0.27**
dslr	webcam	0.31	0.25	0.23	0.23	0.28	0.30	**0.31**
amazon	webcam	0.33	0.03	0.43	0.41	0.43	**0.48**	0.44

Table 2. Domain adaptation results for categories not seen during training in the target domain

		Baseline Methods		Our Method	
domain A	domain B	knnAB	ITML(A+B)	asymm	symm
webcam	dslr	0.37	0.38	**0.53**	0.49
amazonINS	dslrINS	0.23	0.25	**0.30**	0.25

that use traditional supervised metric learning, implemented using the ITML [8] method, trained using all available labels in both domains. We kernelize the metric using an RBF kernel with width $\sigma = 1.0$, and set $\lambda = 10^2$. As a performance measure, we use accuracy (number of correctly classified test samples divided by the total number of test samples) averaged over 10 randomly selected train/test sets. $k = 1$ was used in all experiments.

Same-category setting: In this setting, each category has (a small number of) labels in the target domain (3 in our experiments) For the source domain, we used 8 labels per category for *webcam*/*dslr* and 20 for *amazon*.

We generate constraints between all cross-domain image pairs in the training set based on their class labels, as described in Section 3.1. Table 1 shows the results. In the first result column, to illustrate the level of performance without the domain shift, we plot the accuracy of the Euclidean k-NN classifier trained on the source domain $\mathcal{A}$ and tested on images from the same domain (knn_AA). The next column shows the same classifier, but trained on $\mathcal{A}$ and tested on $\mathcal{B}$ (knn_AB). Here, the effect of the domain shift is evident, as the performance drops for all domain pairs, dramatically so in the case of the *amazon* to *webcam* shift. We can also train k-NN using the few available $\mathcal{B}$ labels (knn_BB, third column). The fourth and the fifth columns show the metric learning baseline, trained either on all pooled training data from both domains ($ITML(A+B)$), or only on $\mathcal{B}$ labels ($ITML(B)$). Finally, the last two columns show the symmetric variant of our domain adaptation method presented in this paper (*symm*), and its asymmetric variant [15] (*asymm*). knn_BB does not perform as well because of the limited amount of labeled examples we have available in $\mathcal{B}$. Even the more powerful metric-learning based classifier fails to perform as well as the k-NN classifier using our domain-invariant transform.

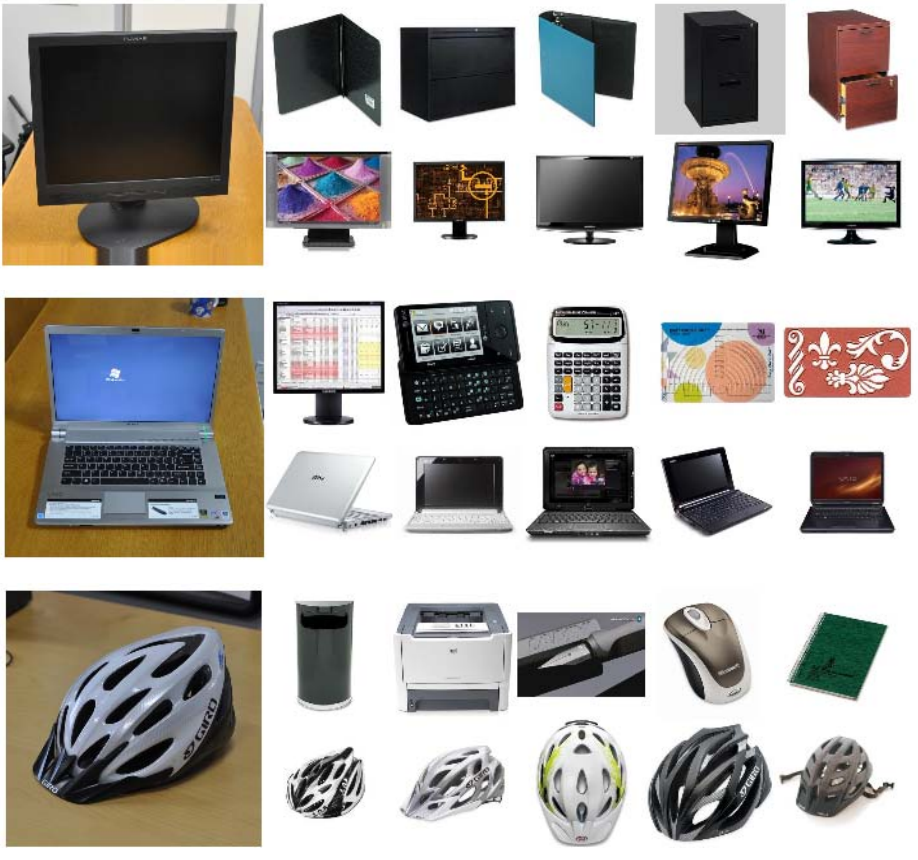


Fig. 5. Examples where our method succeeds in finding images of the correct category despite the domain shift. The large image on the right of each group is a *webcam* query image, while the smaller images are of the 5 nearest neighbors retrieved from the *amazon* dataset, using either the *knn_AB* baseline in Table 1 (top row of smaller images), or the learned cross-domain *symm* kernel (bottom row of smaller images).

The shift between dslr and webcam domains represents a moderate amount of change, mostly due to the differences in the cameras, as the same objects were used to collect both datasets. Since webcam actually has more training images, the reverse webcam-to-dslr shift is probably better suited to adaptation. In both these cases, *symm* outperforms *asym*, possibly due to the more symmetric nature of the shift and/or lack of training data to learn a more general tranformation. The shift between the amazon and the dslr/webcam domains is the most drastic (bottom row of Table 1) Even for this challenging problem, the

adapted k-NN classifier outperforms the non-adapted baselines, with *asymm* doing better than *symm*. Figure 5 show example images retrieved by our method from *amazon* for a query from *webcam*.

New-category setting: In this setting, the test data belong to categories for which we only have labels in the source domain. We use the first half of the categories to learn the transformation, forming correspondence constraints (Section 3.1) between images of the same object instances in roughly the same pose. We test the metric on the remaining categories. The results of adapting *webcam* to *dslr* are shown in the first row of Table 2. Our approach clearly learns something about the domain shift, significantly improving the performance over the baselines, with *asymm* beating *symm*. Note that the overall accuracies are higher as this is a 16-way classification task. The last row shows results on an instance classification task, tackling the shift from Amazon to user environment images. While the symmetric method does not improve on the baseline in this case (possibly due to limited training data, only 2 images per product in *amazon*), the asymmetric method is able to compensate for some of this domain shift.

In both of the above settings, our *symm* method outperforms the standard metric learning baseline $ITML(A+B)$. This clearly demonstrates the advantage of our approach of sampling class-based constraints using inter-domain pairs and, for new-category experiments, of using correspondence constraints.

6 Conclusion

We presented a detailed study of domain shift in the context of object recognition, and introduced a novel adaptation technique that projects the features into a domain-invariant space via a transformation learned from labeled source and target domain examples. Our approach can be applied to adapt a wide range of visual models which operate over similarities or distances between samples, and works both on cases where we need to classify novel test samples from categories seen at training time, and on cases where the test samples come from new categories which were not seen at training time. This is especially useful for object recognition, as large multi-category object databases can be adapted to new domains without requiring labels for all of the possibly huge number of categories. Our results show the effectiveness of our technique for adapting k-NN classifiers to a range of domain shifts.

References

1. 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)
2. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: ACL (2007)
3. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE, Los Alamitos (2008)

4. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR (2007)
5. Chechik, G., Sharma, V., Shalit, U., Bengio, S.: Large scale online learning of image similarity through ranking. In: Pattern Recognition and Image Analysis (2009)
6. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Proc. CVPR (2005)
7. Daume III, H.: Frustratingly easy domain adaptation. In: ACL (2007)
8. Davis, J., Kulis, B., Jain, P., Sra, S., Dhillon, I.: Information-theoretic metric learning. In: ICML (2007)
9. Duan, L., Tsang, I.W., Xu, D., Maybank, S.J.: Domain transfer svm for video concept detection. In: CVPR (2009)
10. Fink, M.: Object classification from a single example utilizing class relevance metrics. In: Proc. NIPS (2004)
11. Hertz, T., Bar-Hillel, A., Weinshall, D.: Learning distance functions for image retrieval. In: CVPR (2004)
12. Hertz, T., Hillel, A.B., Weinshall, D.: Learning a kernel function for classification with small training samples. In: International Conference on Machine Learning (ICML), pp. 401–408 (2006)
13. Jiang, W., Zavesky, E., Chang, S., Loui, A.: Cross-domain learning methods for high-level visual concept classification. In: ICIP (2008)
14. Kulis, B., Jain, P., Grauman, K.: Fast similarity search for learned metrics. IEEE PAMI 39(12), 2143–2157 (2009)
15. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Visual domain adaptation using regularized cross-domain transforms. Technical Report UCB/EECS-2010-106, EECS Department, University of California, Berkeley (July 2010)
16. Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. In: ICCV (2009)
17. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: ICCV (2007)
18. Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive svms. In: ACM Multimedia (2007)

Improved Human Parsing with a Full Relational Model

Duan Tran and David Forsyth

University of Illinois at Urbana-Champaign, USA
{ddtran2,daf}@illinois.edu

Abstract. We show quantitative evidence that a full relational model of the body performs better at upper body parsing than the standard tree model, despite the need to adopt approximate inference and learning procedures. Our method uses an approximate search for inference, and an approximate structure learning method to learn. We compare our method to state of the art methods on our dataset (which depicts a wide range of poses), on the standard Buffy dataset, and on the reduced PASCAL dataset published recently. Our results suggest that the Buffy dataset over emphasizes poses where the arms hang down, and that leads to generalization problems.

1 Introduction

In human parsing, we have a region of interest (ROI) containing a person, perhaps produced by a detector, and we must produce an accurate representation of the body configuration. This problem is an important part of activity recognition; for example, the ROI might be produced by a detector, but we must know what the arms are doing to label the activity. The representation produced is usually a stick figure, or a box model, but may be image regions or joint locations. All representations encode the configuration of body segments.

It is usual to represent pairwise spatial relations between locations structured into a kinematic tree, so that dynamic programming can be used for inference [10,6]. The joint relations encoded by the kinematic tree model are important, but there are other important relations. Limbs on the left side of the body usually look like those on the right. This cue should be important, because limbs are genuinely difficult to detect, particularly in the absence of an appearance model. Even the strongest recent methods have difficulty detecting forearms (e.g. [1], 32%, p8). Inference difficulties occur when one encodes relations between all pairs of segments, because finding the best parse now becomes max-cut. Approximate inference on sets of extended image segments can produce good parses for difficult images [16]. However, there is no evidence comparing the benefits of a full model against the cost of approximate inference.

In this paper we explore the advantages of representing a full set of relations for human parsing. We show strong quantitative evidence that the advantages of representing a full set of relations between segments outweigh the costs of approximate inference and approximate learning. We concentrate on upper body parsing, and show results on Buffy and Pascal dataset [9], and on a new dataset where the prior on body configuration is quite weak.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 227–240, 2010.

Fig. 1. A tree model of the upper body represents only relations that can be organized as a tree (always the kinematic relations in the natural tree, **top left**). By doing so, it omits relations that are important; for example, the left arm typically looks like the right arm. A full model (**bottom left** — we have indicated only some of the relations omitted by the tree model) encodes these relations, at the cost of approximate inference and approximate training. There is qualitative evidence in the literature that full models can yield good parses [16]; in this paper, we show quantitative evidence on two datasets that a full model offers strong advantages over a tree model. On the **right**, we show parses derived from a tree model (**top**) and a full model (**bottom**); note that the appearance constraints in the full model often help arms to be in the right place. This is confirmed by our quantitative data.

1.1 Related Work

For a tree structured kinematic model, and segments with known appearance, inference is by dynamic programming (the "pictorial structure model" [5]). A similar approach can be applied to informative local patches [15], or to joint locations using velocities at the joints, assuming known activity [25]. If segment appearance is unknown, it can be recovered from motion sequences [19], or by an iterative procedure of estimating appearance, then configuration, etc. [18]. The iterative procedure can produce good parses, but can fail if the search starts poorly. These methods can be costly, because the search space is a discretization of all possible segment configurations. Improvements result from estimating head location, then pruning the search space [9]; and from tuning the initial appearance model with spatial priors on segment locations and respecting

likely interactions between segment appearance models (the upper arm is often the same color as the upper body) [3]. Alternatively, local segment scores can be computed by appearance independent segment detectors; edge based limb detectors give limited performance [21], but a combination of HOG and color segmentation features beats the original iterative process [14].

There has been considerable experimental tuning of various tree models. A ten-body segment model (head, upper body, two for each arm and two for each leg) is now universal. The standard scoring procedure regards a prediction correct if its endpoints lie within 50% of the ground truth segment length from the true positions; this score is fairly generous, because body segments are long. Nonetheless, high scores are difficult to get. Ramanan [18] has published a widely used dataset for full body localization, on which the best method (Andriluka *et al.*, 2009 [1]) gets 55.2%. Ferrari *et al.* [9] published another for upper body localization, on which the best method (Eichner *et al.*, 2009 [3]) gets 80.3% in its best configuration on Buffy dataset. On a selected subset of PASCAL challenge images, this method gets 72.3%. The strongest methods all use carefully constructed and tuned segment detectors, with search space pruning.

The tree model presents real difficulties: limb occlusions seem to be correlated; tree models tend to prefer parses that superimpose both arms or both legs on one promising set of image segments; and a tree model cannot encode the tendency of the left arm (resp. leg) to look like the right arm (resp. leg). Fergus *et al.*were able to work with models encoding all relations between up to seven object parts using efficient pruning methods [8]. Tian *et al.*extend the tree model by inserting appearance constraints between segments (e.g. left lower leg looks like right lower leg); in this case, the tree model supplies a strong enough bound that exact inference using branch and bound is possible [28]. Sapp *et al.*demonstrate improvements in performance by making some terms in the tree model depend on an approximate estimate of global pose, obtained by matching the test image to the training dataset [22]. This suggests strong structural correlations appear in poses. Correlated limb occlusions can be dealt with using a mixture of trees without complicating inference [11]. Alternatively, Sigal *et al.*use a modification to the likelihood to avoid double counting in the case of occluded limbs [24]. Doubled limbs can be controlled with "repulsive" edges [3,13], or by converting the energy function into a posterior probability, drawing samples, and using some process to decide which is best (for example, rejecting parses where arms overlap) [5]. An alternative, which requires approximate inference, is to require that the model covers more of the image pixels [12].

Recent work has shown that human parses can be helped by identifying objects people might be using [2,32]. Yao *et al.*use a fixed set of quantized body poses, with exact segment locations depending probabilistically on (a) the type of the pose (and so on the activity label and nearby objects through this) and (b) other segment locations [32]. Their method can produce parses that are as good as, or better than, the state of the art, for a sports dataset of stylised poses.

Another important difficulty is determining which poses to work with. In our opinion, the performance of a parser should be as close to pose-independent as possible. That is, the parser should be tested (if not trained) on a dataset with a rich selection of poses at approximately even frequencies. This is complicated, because natural data

sources often have strong biases — as we shall see, TV actors in stills tend to have their arms by their sides. The result is very strong effects due to the prior, which can cause generalization problems. For these reasons, we have collected a further dataset that emphasizes rich upper body configurations.

2 Method

Our approach defines a search space in the image using a set of tuned body segment detectors. We then build an energy model that is regressed against actual loss for a set of parses of each training image. Our model scores appearance and spatial relations between all pairs of segments in the image. We then find the best parse by an approximate maximization procedure.

We have detectors for upper body, head and arm segments. Our detectors do not distinguish between upper and lower arms. We must choose a label (head, upper body, left/right upper/lower arm, null) for each response. For image $\mathcal{I}$, we build a scoring function $C(L;\mathcal{I})$ which evaluates a labelling L of the responses. We consider only labellings that are consistent, in the sense that we do not attempt to label head detector responses as upper bodies, etc. Write S_i for the i-th body segment in the model, D_j for the j-th detector response in the image, and $L(S_i)$ for the image segment labelled S_i by L. Our energy is a linear combination of unary and binary features, which we write as

$$C(L;\mathcal{I}) = \sum_{i \in \text{features}} w_i \phi_i(L;\mathcal{I}) = \mathbf{W}^T \Phi(L;\mathcal{I})$$

where each feature ϕ_i is either a unary feature (yielding $\phi_i = \phi_i(S_j, L(S_j);\mathcal{I})$) or a binary feature (yielding $\phi_i = \phi_i(S_j, S_k, L(S_j), L(S_k);\mathcal{I})$). We do *not* require that the set of binary features form a tree, but represent all pairs. Our features measure both spatial and appearance relations (section 2.4). The scoring function can be converted to an energy by $E(L) = -C(L)$; a probability model follows, though we do not use it.

2.1 Searching a Full Energy Model

Finding the best labelling involves solving a general zero-one quadratic form subject to linear constraints, and there is no exact algorithm. While approximate algorithms for MRF's could be applied, most labels are null and there is only one instance of each non-null label, meaning that expansion moves are unlikely to be successful. We use an approximate search procedure that relies on the proven competence of tree models.

The upper body detector is quite reliable, so there are relatively few false positives. This means we can search for a configuration at each upper body, then take the overall best configuration. Because tree models are quite reliable, we can use specialised tree models to produce arm candidates on each side of each given upper body, then evaluate all triples of right arm-torso-left arm. Finally, we use a local search to improve segments.

Obtaining arm candidates: We need to obtain candidates for left (resp. right) arm that have a good chance of being in the final configuration. We can do so by simplifying the

cost function, removing all terms apart from those referring to upper body, left (resp. right) upper arm and left (resp. right) lower arm. The resulting simplified cost function is easily maximised with dynamic programming. We keep the top 300 candidates found this way for each side.

Building good triples: We now have 300 candidates each for left (resp. right arm), and a set of head candidates. We obtain the top five triples by exhaustive evaluation of the whole cost function.

Polishing with local search: Limb detectors chatter, because they must respond to contrast at limb edges and limbs are narrow; it is usual to see more than one response near a segment. To counteract this effect, we polish each of the top five triples. We repeatedly fix five segments and search for the best candidate for the sixth, stopping when all segments have been visited without change. We now report the best of the polished five triples for each upper body.

Detection: We report the parse associated with the best upper body detector response. In principle, one could parse multiple people in an image by keeping all such parses, applying a threshold to the cost function, and using non-maximum suppression (to control chatter at the upper body detector); since most images in evaluation datasets contain single people, and since our focus is on "hard parses", we have not investigated doing so.

Complexity: With 6 human parts in the model, the exact solution will cost $O(T * H * LUA * LLA * RUA * RLA)$ where T, H are torso and head detections, LUA, LLA and RUA, RLA are left upper, lower arms (resp. right upper and lower arms) detections. While T and H are small (less than 10 each), LUA, LLA, RUA, RLA are quite large (normally more than 100 each after pruning by the closeness to the torso), this complexity is practically intractable. However, our approximate solution has complexity $O(T * H * LA * RA) - -LA, RA$: numbers of full left (resp. right arms) that we keep top 300 for each). This complexity is tractable, and though it is an approximation, it still proves its benefit of improving the performance. In fact, Our implementation in C just takes around 5 seconds for one parsing on a computer of Xeon 2.27HGz.

2.2 Training a Full Energy Model

We wish to train the energy model so that detections using that model are as good as possible. **Structure learning** is a method that use a series of correct examples to estimate appropriate weightings of features relative to one another to produce a score that is effective at estimating configuration (in general [26,27]; applied to parsing [29]). For a given image $\mathcal{I}$ and known $\mathbf{W}$ the best labelling is

$$\underset{L \in L(\mathcal{I})}{\arg\max} \mathbf{W}^T \Phi(L; \mathcal{I})$$

though we cannot necessarily identify it. We choose a loss function $\mathcal{L}(L_p, \hat{L})$ that gives the cost of predicting L_p when the correct answer is $\hat{L}$. Write the set of n examples as

$\mathcal{E}$, and $L_{p,i}$ as the prediction for the i'th example. Structure learning must now estimate a $\mathbf{W}$ to minimize the hinge loss as in [20,30,26]

$$\min \lambda \frac{1}{2} \| \mathbf{W} \|^2 + \frac{1}{n} \sum_{i \in examples} \xi_i$$

subject to the constraints

$$\forall i \in \mathcal{E},\ \mathbf{W}^T \Phi(\hat{L}; \mathcal{I}_i) + \xi_i \geq \max_{L_{p,i} \in L(\mathcal{I}_i)} (\mathbf{W}^T \Phi(L_{p,i}; \mathcal{I}_i) + \mathcal{L}(L_{p,i}, \hat{L}_i))$$
$$\xi_i \geq 0$$

At the minimum, we can choose the slack variables ξ_i to make the constraints equal. Therefore, we can move the constraints to the objective function, which is:

$$\lambda \frac{1}{2} \| \mathbf{W} \|^2 + \frac{1}{n} \sum_{i \in examples} \max_{L_{p,i} \in L(\mathcal{I}_i)} \left(\begin{array}{c} \mathbf{W}^T \Phi(L_{p,i}; \mathcal{I}_i) + \\ \mathcal{L}(L_{p,i}, \hat{L}_i) - \mathbf{W}^T \Phi(\hat{L}; \mathcal{I}_i) \end{array} \right)$$

Notice that this function is convex, but not differentiable. We use the cutting-plane method of [30], as implemented in SVM-Struct package[1]. Our approach is:

Start: we initialize $\mathbf{W}$, and prepare a pool of candidate labellings $\mathcal{C}_i^{(0)}$ for each example image using the search of section 2.1. Then, iterate multiple rounds of

1. for each example, compute the best (most violated constraint) labelling $L_{p,i} = \arg\max_{L \in \mathcal{C}_i^{(k)}} \mathbf{W}^T \Phi(L; \mathcal{I}_i) + \mathcal{L}(L_{p,i}, \hat{L}_i)$.
2. pass these labellings to SVM-Struct to form cutting planes to update $\mathbf{W}$.

This procedure will stop until there are no violated labelling found (in this case, the ground truth labelling is the highest score) or no significant change in the objective value when updating $\mathbf{W}$. We observe that the learning converges after 50-60 iterations.

Table 1. Summary of part detectors. Equal error rates (EER) are computed with 5-fold cross validation. The lower arm detector is not comparable to others as it tends to be dataset dependent. We operate the detectors at 92% recall and given a upper body candidate we keep the 300 best lower arm responses.

Detector	Size	Features	EER
Upper body	80x80	HOG	0.096 +/-0.005
Head	56x56	HOG	0.123+/0.012
Lower arm	30x30	HOG, SSIM	0.249+/-0.068

[1] http://svmlight.joachims.org/svm_struct.html

2.3 Part Detectors

We have detectors for upper body, head and arm segments, but do not distinguish between upper and lower arms for a total of three part detectors. The detectors are oriented, and we use a total of 25 orientations of $(-180^o..180^o)$ for arm detectors and 13 orientations of $(-90^o..90^o)$ for head detector and upper body detector. We use code from

Table 2. This table shows pairwise features (undirected links) to be computed. [D]: distance binning, [A]: appearance difference, [N]: angle, [O]: overlap

Parts	Upper body	Head	LUA	LLA	RUA
Upper body	-	-	-	-	-
Head	D,A,N,O	-	-	-	-
LUA	D,A,N,O	A,O	-	-	-
LLA	D,A,N,O	A,O	D,A,O	-	-
RUA	D,A,N,O	A,O	A,O	A,O	-
RLA	D,A,N,O	A,O	A,O	A,O	D,A,O

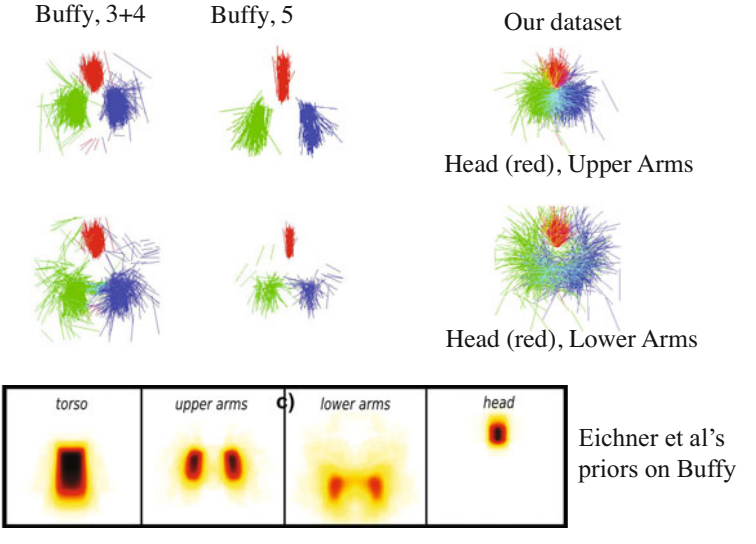


Fig. 2. In the Buffy dataset, upper arms hang by the sides of the body, and lower arms mostly do so as well. This very strong spatial prior can overcome contributions by other parts of a model, but impedes generalization. **Above:** Scatter plots of head and upper arm (**top row**) or lower arm (**bottom row**) sticks with respect to fixed upper body position for the Buffy 3 and 4 ground truth, Buffy 5 ground truth, and our ground truth. Notice how compact the prior configuration is for the Buffy datasets. Our dataset emphasizes a wide range of body configurations. **Below:** Part of figure 1, from [3], for reference, showing the location priors derived in that paper for the Buffy dataset; again, note the highly compact prior.

Table 3. Average PCP over body segments for a full model; for a tree model; and for Eichner and Ferrari [3], who use a tree model with a location prior to recover appearance. Performance of the full model is much better than performance of tree models, except for the model of Eichner and Ferrari applied to Pascal or to buffy_s256. However, for all models, training on Buffy_256 leads to strong generalization problems (performance on Buffy_256 is much better than performance on other test sets), most likely because of the quite strong bias in arm location. We believe that the very strong performance of Eichner and Ferrari on Buffy_s256 should be ascribed to the effects of that bias. Buffy_s5e3&Pascal appears to contain a similar, but smaller, bias (compare training on this and testing on Buffy_s256 with training on this and testing on our_test). We do not have figures for Eichner and Ferrari's method trained on Buffy_s2to6 and tested on Buffy_s256. Note that: Buffy_s5e256_sub and Buffy_s5e3&Pascal_sub are subsets of 150 examples randomly chosen for each dataset.

Model		Test set		
		our_test	Buffy_s5e256	Pascal
	Train set			
Full model	our_train	0.663	0.623	0.627
	Buffy_s5e256_sub	0.583	0.676	0.625
	Buffy_s5e3&Pascal_sub	0.613	0.628	
Tree model	our_train	0.608	0.552	0.565
	Buffy_s5e256_sub	0.545	0.629	0.599
	Buffy_s3&Pascal_sub	0.565	0.596	
Eichner&Ferraris	Buffy_s5e2to6	0.557		0.675
	Buffy_s5e34&Pascal	0.559	0.801	

Felzenszwalb[2] to compute HOG features [7] for upper body and head detectors. Arm segment detectors use HOG features and self-similarity features (from [23], using the implementation of V. Gulshan). This detector does not distinguish between upper and lower arms, because locally they are similar in appearance. Upper arms can be difficult to detect, because there may be little contrast between the segment and the body. To overcome this difficulty, we also use a *virtual* upper arm detector, obtained by joining points nearby the top of the upper body segment to the elbows of nearby lower arm segments.

Lower arms can be strongly oriented (i.e. long and thin), and our arm detector may respond more than once to a lower arm in a lateral view. Extending the support of the detector does not help, unless one searches an impractical number of orientations. We deal with this by expanding the search space: we add new lower arms to the pool of detector responses, made by fusing nearby arm detections at the same orientation.

All part detectors are linear SVM trained on cropped parts from our dataset and from some of Buffy_s5e3 dataset. We bootstrap upper body and head detectors on a subset of background images, and lower arm detector on subset training images (regions outside the subject box). Table 1 summarizes part detector parameters.

[2] http://people.cs.uchicago.edu/~pff/latent/

Fig. 3. Examples of stick-figure, upper body parses of figures in our dataset produced by the full model trained on our dataset **top** row, our tree model **top-center** and the code of Eichner *et al.***bottom center** (trained on buffy_2to6) and **bottom** (trained on buffy_3&4 and pascal), all applied to our dataset. Red: upper body; Green: head; Blue-Purple: left upper/lower arm; Green-Yellow: right upper-lower arm. Note doubled arms produced by the tree model and a tendency for Eichner *et al.*to produce hanging arms, most likely a result of the strong geometric prior in their training datasets.

2.4 Features

We use a binning scheme, after [18]. Binning takes a feature such as distance and quantizes the range to a set of discrete bins, then sets the bin into which a value falls to be one and all others zero. We find it helpful to antialias by splitting the vote among nearby bins.

Unary features are the detector score at the detection labelled with a part label (converted to a probability using the method of [17]), and a binned vector representing the part length. For virtual upper arms, we have no detector score and instead use the value of the detector response at the lower arm used to create the virtual upper arm.

Binary features are different for different pairs of parts. We use six parts: upper body (from chest to navel), head (from top forehead to chin), left upper arm (LUA), left lower arm (LLA), right upper arm (RUA), and right lower arm (LLA). For each pair, we compute features from *distance*, *appearance*, *angle*, or *overlap*, according to the scheme of table 2.

Distance features for a pair of segments consist of a binned vector representing distance between endpoints, concatenated with the actual distance. The **comparative appearance feature** is formed from a set of appearance vectors. The appearance vectors consist of normalized color histograms, normalized Gabor filter histograms [4], and a histogram of textons [31]. For each type of appearance vector, we compute the χ^2 distance between the vectors corresponding to the two segments to be compared. For speed, integral images of appearance features are precomputed over reoriented images. **Angle features** are given by a binned angle vector representing signed angle from segment 1 to segment 2 in the range $(-90^o..90^o)$ for the head-torso pair, and $(-180^o..180^o)$ for all others. **Overlap features** give the ratio of endpoint distances to segment length, with the ratio computed for each segment. There are a total of 707 features.

3 Experimental Results

We compare a full model to a tree model on three datasets, described below. The full model is trained as above. The tree model is trained in the same way, but with the weights of features representing relations not in the tree clamped at zero. Inference (and so training) of the tree does not require a polishing step, because dynamic programming is exact. The tree is the usual kinematic tree (figure 1).

3.1 Dataset

We describe results on three datasets. The first is the Buffy dataset of [9], in various partitions. This dataset has little variation in layout (figure 2). The second is the subset

Fig. 4. Examples of stick-figure, upper body parses of figures in the buffy produced by the full model trained on ours **top** row, and our tree model trained on ours **bottom**. Red: upper body; Green: head; Blue-Purple: left upper/lower arm; Green-Yellow: right upper-lower arm. Note doubled arms produced by the tree model, and the strong tendency for poses to have hanging arms.

Fig. 5. Examples of stick-figure, upper body parses of figures in our dataset produced by the full model trained on our dataset **top** row, our tree model **top-center** and the code of Eichner *et al.***bottom center** (trained on buffy_2to6) and **bottom** (trained on buffy_3&4 and pascal), all applied to our dataset. Red: upper body; Green: head; Blue-Purple: left upper/lower arm; Green-Yellow: right upper-lower arm. Note doubled arms produced by the tree model and a tendency for Eichner *et al.*to produce hanging arms, most likely a result of the strong geometric prior in their training datasets.

of Pascal images marked up and released by [3]. Human parsing results are usually intended to drive activity recognition, which is at its most interesting when the body takes unusual postures. Methods that work well on a dataset with a strong spatial bias may do so because (say) they are particularly good at some common poses; such methods may not be useful in practice. For this reason, we have created a third dataset of 593 images (346 training, 247 test), marked up with stick figures by hand. This dataset is built to have aggressive spatial variation in configuration (figure 2).

3.2 Results

We follow convention and measure performance with PCP (Percentage of Correctly estimated body Parts). In this method, a segment is correct if its endpoints lie within 50% of the length of the ground truth from the annotated location [9]. Since our method produces one parse for each upper body detector response, we apply non-maximum suppression to the score, to prevent effects from multiple nearby upper body detector

responses. As in Eichner *et al.* [3], we evaluate PCP only for stickmen whose upper body response overlaps the correct upper body.

On Buffy and Pascal, our method obtains 62.3% and 62.7%, respectively (compare 80.3% and 72.3%, Eichner *et al.* [3]). However, there are two difficulties with these dataset (especially for Buffy), both potentially quite serious. First, there is little variation in pose. Figure 2 shows a scatter plot of ground truth head and arm segments for overlaid upper body segments. Head, upper arm and lower arm segments all have relatively little scatter — most figures are upright. Second, the contrast for many frames is relatively low. Both issues suggest that careful detector engineering will produce improvements in performance by overcoming contrast difficulties. Detector engineering is a valuable contribution which is responsible for all advances on the buffy dataset, but it will not identify better or worse modelling techniques. Because the spatial configuration varies so little in the Buffy dataset, comparisons of modelling techniques on this dataset should be approached with caution.

On all three datasets, the full model significantly outperforms the tree model (table 3). This is most likely because appearance consistency constraints between upper arms help overcome relatively low contrast at the boundary. Typical results suggest that improvements occur because consistency in appearance (the left arm must look like the right) is a cue that helps parse, and possibly because the model is spatially more rigid than a tree model (figure 5). The value of these cues outweighs the cost of approximate inference and approximate learning. Our parser can be configured as a detector by applying non-maximum suppression to the parse score and thresholding.

4 Discussion

We have shown quantitative evidence that a full relational model of the body performs better at upper body parsing than the standard tree model, despite the need to adopt approximate inference and learning procedures. We have obtained our results on a new dataset where there is extensive spatial variation in body configuration. Our results suggest that appearance consistency constraints help localize upper arms better. Our method extends to a full body parse.

Acknowledgements

This work was supported in part by the National Science Foundation under IIS - 0534837 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program, and in part by the Vietnam Education Foundation through a fellowship to Duan Tran. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the NSF, the ONR, or the VEF.

References

1. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR (2009)
2. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for static human-object interactions (2010)

3. Eichner, M., Ferrari, V.: Better appearance models for pictorial structures. In: British Machine Vision Conference (2009)
4. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Learning to describe objects. In: CVPR (2009)
5. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV 61(1), 55–79 (2005)
6. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient matching of pictorial structures. In: CVPR (2000)
7. Felzenszwalb, P.F., McAllester, D.A., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
8. Fergus, R., Perona, P., Zisserman, A.: Object Class Recognition by Unsupervised Scale-Invariant Learning. In: CVPR (2003)
9. Ferrari, V., Marin, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: CVPR (2008)
10. Ioffe, S., Forsyth, D.: Finding people by sampling. In: ICCV, pp. 1092–1097 (1999)
11. Ioffe, S., Forsyth, D.: Human tracking with mixtures of trees. In: ICCV, pp. 690–695 (2001)
12. Jiang, H.: Human pose estimation using consistent max-covering. In: ICCV (2009)
13. Jiang, H., Martin, R.: Global pose estimation using non-tree models. In: CVPR (2008)
14. Johnson, S., Everingham, M.: Combining discriminative appearance and segmentation cues for articulated human pose estimation. In: MLVMA 2009 (2009)
15. Mori, G., Malik, J.: Estimating human body configurations using shape context matching. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 666–680. Springer, Heidelberg (2002)
16. Mori, G., Ren, X., Efros, A.A., Malik, J.: Recovering human body configurations: Combining segmentation and recognition. In: CVPR (2004)
17. Platt, J.: Probabilities for sv machines. In: Advances in Neural Information Processing (1999)
18. Ramanan, D.: Learning to parse images of articulated bodies. In: Advances in Neural Information Processing (2006)
19. Ramanan, D., Forsyth, D., Barnard, K.: Building models of animals from video. PAMI 28(8), 1319–1334 (2006)
20. Ratliff, N., Bagnell, J.A., Zinkevich, M.: Subgradient methods for maximum margin structured learning. In: ICML 2006 Workshop on Learning in Structured Output Spaces (2006)
21. Ronfard, R., Schmid, C., Triggs, B.: Learning to parse pictures of people. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, p. 700. Springer, Heidelberg (2002)
22. Sapp, B., Jordan, C., Taskar, B.: Adaptive pose prior for pictorial structure. In: CVPR (2010)
23. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: CVPR (2007)
24. Sigal, L., Black, M.J.: Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In: CVPR (2006)
25. Song, Y., Feng, X., Perona, P.: Towards detection of human motion. In: CVPR, pp. 810–817 (2000)
26. Taskar, B.: Learning Structured Prediction Models: A Large Margin Approach. PhD thesis, Stanford University (2004)
27. Taskar, B., Lacoste-Julien, S., Jordan, M.: Structured prediction via the extragradient method. In: Neural Information Processing Systems Conference (2005)
28. Tian, T.-P., Sclaroff, S.: Fast globally optimal 2d human detection with loopy graph models. In: CVPR (2010)

29. Tran, D., Forsyth, D.: Configuration estimates improve pedestrian finding. In: Advances in Neural Information Processing (2007)
30. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research (JMLR) 6, 1453–1484 (2005)
31. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. Int. J. Computer Vision 62(1-2), 61–81 (2005)
32. Yao, B., Fei-Fei, L.: Model mutual context of object and human pose in human-object interaction activities. In: CVPR (2010)

Multiresolution Models for Object Detection

Dennis Park, Deva Ramanan, and Charless Fowlkes

UC Irvine, Irvine CA 92697, USA
{iypark,dramanan,fowlkes}@ics.uci.edu

Abstract. Most current approaches to recognition aim to be scale-invariant. However, the cues available for recognizing a 300 pixel tall object are qualitatively different from those for recognizing a 3 pixel tall object. We argue that for sensors with finite resolution, one should instead use scale-variant, or multiresolution representations that adapt in complexity to the size of a putative detection window. We describe a multiresolution model that acts as a deformable part-based model when scoring large instances and a rigid template with scoring small instances. We also examine the interplay of resolution and context, and demonstrate that context is most helpful for detecting low-resolution instances when local models are limited in discriminative power. We demonstrate impressive results on the Caltech Pedestrian benchmark, which contains object instances at a wide range of scales. Whereas recent state-of-the-art methods demonstrate missed detection rates of 86%-37% at 1 false-positive-per-image, our multiresolution model reduces the rate to 29%.

1 Introduction

Objects appear at a continuous range of scales in unconstrained photographs of the world. This constitutes a significant mode of intra-class variability in detection problems. The dominant perspective in the recognition community is that one should strive for scale-invariant representations, e.g., by computing features with respect to an appropriately adapted coordinate frame, as in SIFT or scanning window detectors. While this is conceptually elegant, it ignores the fact that finite sensor resolution poses an undeniable limit to scale-invariance. Recognizing a 3-pixel tall object is fundamentally harder than recognizing a 300-pixel object or a 3000-pixel object.

This is perhaps most readily apparent in common demonstrations of the importance of context in recognition (e.g., [1]). For example, the same local patch of pixels may be identified as a car or phone depending on whether the surroundings look like a street scene or a person in an office. However, such demonstrations always involve a low-resolution, heavily-blurred image of the object in question. Given enough resolution, one *should* be able to recognize a toy-car held up to someone's ear despite the improbable context. This suggests that scene context itself should also be entered into detection in a scale-*variant* fashion with contextual cues only being used to increase the accuracy of recognizing small instances, where local image evidence is uninformative.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 241–254, 2010.

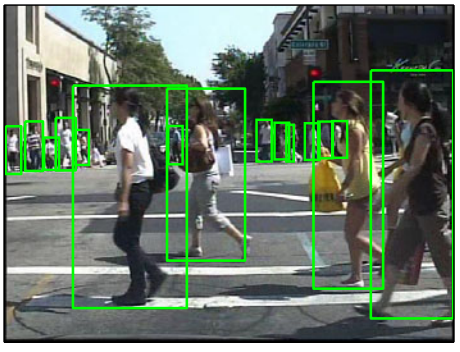


Fig. 1. An example test image in Caltech Pedestrian dataset and its ground truth annotations. The detection results of baselines and our algorithm on this image are shown in Fig 3. Note that people appear at a wide range of scales.

In this paper we propose that models for object detection should have a multiresolution structure which utilizes features ranging from detailed high-resolution parts, to whole object templates, to scene context cues. Furthermore, we treat these features in a scale dependent manner, so that high-resolution features are not used when detecting low-resolution instances.

We examine the interplay of resolution and context in the domain of pedestrian detection for autonomous vehicle navigation. Much of the recent successful work is based on template detection. We begin by asking a simple question - *what should the size of the template be*? On one hand, we want a small template that can detect small people, important for providing time for a vehicle to react. On the other hand, we want a large template than can exploit detailed features (of say, faces) to increase accuracy. Such questions are complicated by the fact a simple rigid template is not likely to accurately model both extremes, and that contextual cues should perhaps be overridden by high-confidence, large-scale detections. Using a well-known pedestrian benchmark [2], we demonstrate that contextual multiresolution models provide a significant improvement over the collective recent history of pedestrian models (as surveyed in [2]).

2 Related Work

There is storied tradition of advocating scale-invariance in visual recognition, from scale-invariant feature detectors [3,4,5] to scale-invariant object representations [6,7]. Unfortunately, such scale-invariant representations don't leverage additional pixel resolution for detecting large-scale instances.

Another family of representations deal with *multiscale* models that compute features at multiple scales. Such models are typically not multiresolution in that they do not adapt in complexity to the size of a putative detection. Examples include multiscale edge models [8] and object representations based on multi-scale wavelets [9,10]. Our approach is most similar to the multiscale part model

of [11] that defines both a low-resolution root template and high-resolution part filters. We extend the publically-available code to encode adaptive multiresolution models that act as rigid templates when scoring small-scale instances and flexible part-based models when scoring large-scale instances.

There is a quite large literature on pedestrian detection, dating back to the early scanning-window classifers of [9,12,13]. We refer the reader to the recent surveys [2,14] for an overview of contemporary approaches. Recent work has focused on models for handling pose variation [15,11,16,17,18], reducing complexity of learning [19,20]), and multicue combination [21,22]. To the best of our knowledge, there has been no past work on multiresolution representations of pedestrians.

3 Multiresolution Models

We will describe a family of multiresolution template models of increasing complexity. To establish notation, we begin with a description of a simple fixed-resolution template.

3.1 Fixed-Resolution Models

Let x denote an image window and $\Phi(x)$ denote its extracted features - say, histogram of oriented gradient (HOG) features [13]. Following an established line of work on scanning-window linear classifiers [23,13], we label x as a pedestrian if

$$f(x) > 0 \qquad \text{where} \qquad f(x) = w \cdot \Phi(x) \tag{1}$$

Such representations are trained with positive and negative examples of pedestrian windows – formally, a set of pairs (x_i, y_i) where $y_i \in \{-1, 1\}$. Popular training algorithms include SVMs [23,13] and boosting [24,25]. In our work, we will train w using a linear SVM:

$$w^* = \underset{w}{\operatorname{argmin}} \frac{1}{2} w \cdot w + C \sum_i \max(0, 1 - y_i w \cdot \Phi(x_i)) \tag{2}$$

One hidden assumption in such formalisms is that both the training and test data x_i is assumed to be scaled to a canonical size. For example, in Dalal and Triggs' [13] well-known detector, all training and test windows are scaled to be of size 128×64 pixels. The detector is used to find larger instances of pedestrians by scaling down in the image, implemented through an image pyramid. Formally speaking, the detector cannot be used to find instances smaller than 128×64. In practice, a common heuristic is to upsample smaller windows via interpolation, but this introduces artifacts which hurt performance [11,2].

In this paper, we define a feature representation $\Phi(x)$ that directly processes windows of varying size, allowing one to extract additional features (and hence build a more accurate model) when x is a large-size window.

3.2 Multiple Fixed-Resolution Models

Arguably the simplest method of dealing with windows of varying sizes is to build a separate model for each size. Assume that every window x arrives with a bit s that specifies whether it is "small" or "large". One can still write two templates as a single classifier $f(x, s) = w \cdot \Phi(x, s)$ where:

$$\Phi(x,s) = \begin{bmatrix} \phi_0(x) \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{if} \quad s = 0 \qquad \text{and} \qquad \Phi(x,s) = \begin{bmatrix} 0 \\ 0 \\ \phi_1(x) \\ 1 \end{bmatrix} \text{if} \quad s = 1 \tag{3}$$

Here, $\phi_0(x)$ and $\phi_1(x)$ represent two different feature representations extracted at two different scale windows - say for example, 50-pixel and 100-pixel tall people. Given training data triples (x_i, s_i, y_i) one could learn a single w that minimizes training error in (2) where $\Phi(x_i)$ is replaced by $\Phi(x_i, s_i)$.

It is straightforward to show that (2) reduces to *independent* SVM problems given the above multiresolution feature. It is equivalent to partitioning the dataset into small and large instances and training on each independently. This poses a problem since the detector scores for small and large detections need to be comparable. For example, one might expect that small-scale instances are harder to detect, and so such scores would generally be weaker than their large-scale counterparts. Comparable scores are essential to allow for proper non-max suppression between scales, contextual reasoning [26] and for ROC benchmark evaluation.

3.3 Multiscale Multiresolution Models

One mechanism of integrating two fixed-scale models is to also compute $\phi_0(x)$ for windows with $s = 1$. In other words, we can always resize a 100-pixel windows to 50-pixels and compute the resulting small-scale feature. This allows the large-resolution model to be *multiscale* in that features are computed multiple resolutions:

$$\Phi(x,s) = \begin{bmatrix} \phi_0(x) \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{if} \quad s = 0 \qquad \text{and} \qquad \Phi(x,s) = \begin{bmatrix} \phi_0(x) \\ 0 \\ \phi_1(x) \\ 1 \end{bmatrix} \text{if} \quad s = 1 \tag{4}$$

Note that because the coarse-scale features $\phi_0(x)$ are shared across both representations, the training problem no longer reduces to learning separate SVMs. In this case, distinct bias terms make scores for large and small instances comparable.

3.4 Multiresolution Part Models

One limitation of the above approach is that both small and large-scale models are encoded with a rigid template. Low-level descriptors such as HOG are

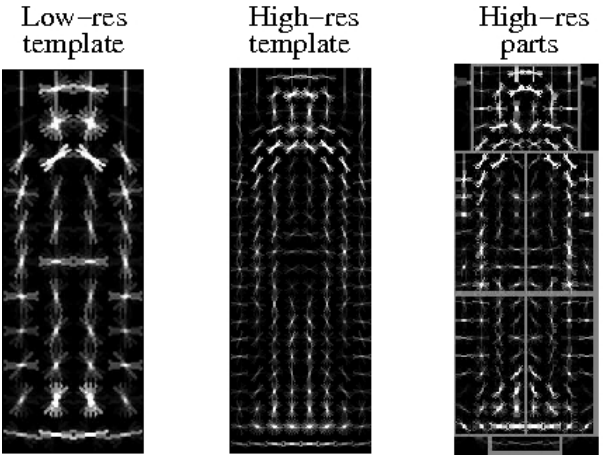


Fig. 2. Finding large-scale instances. One might use a low-resolution template (shown on the **left**). Alternatively, to exploit the extra resolution of large-scale instances, one might define a high-resolution template (**middle**). Edges capturing the boundary of the body and head are blurred out due to variation in the postures of pedestrians in the training data. A more successful approach is to explicitly model the deformation with a part model (shown on the **right**), which learns sharper part templates.

invariant to small scale image deformation due to the local spatial binning of gradient values. However, this binning occurs at a fixed-size neighborhood (in our case, a neighborhood of 4×4 pixels). On the other hand, object deformations (such as the articulation of a pedestrian) occur at a scale relative to the *size of the instance*. This means that a HOG descriptor is likely invariant to the pose deformations of a 50-pixel pedestrian, but not a 100-pixel tall pedestrian.

To model pose variations at larger scales, we augment our large-scale model with a latent parameter capturing pose variation. Following the work of [11], we add a latent parameter z that specifies the location of a collection of parts. Given the z, we define $\phi_1(x, z)$ to be a vector of vectorized-HOG features extracted at the given part locations, appended with the part offsets themselves. This allows the corresponding parameters from w to encode part templates and part deformation parameters that penalize/favor certain part deformations over others.

$$\Phi(x, s, z) = \begin{bmatrix} \phi_0(x) \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ if } \quad s = 0 \qquad \text{and} \qquad \Phi(x, s, z) = \begin{bmatrix} \phi_0(x) \\ 0 \\ \phi_1(x, z) \\ 1 \end{bmatrix} \text{ if } \quad s = 1 \quad (5)$$

The final classifier searches over latent values $f(x, s) = \max_z w \cdot \Phi(x, s, z)$:

$$f(x, s) = \begin{cases} w_0 \cdot \phi_0(x) + b_0 & \text{if} \quad s = 0 \\ w_0 \cdot \phi_o(x) + \max_z w_1 \cdot \phi_1(x, z) + b_1 & \text{if} \quad s = 1 \end{cases} \quad (6)$$

When scoring small instances, the above reduces to a standard linear template. When scoring large instances, the above requires a search over all part deformations, for the configuration that yields the maximum score. As in [11], we assume parts are independently positioned given a root location, equivalent to the standard "star" model assumptions in part-based models. This allows us to use dynamic programming to efficiently compute the max:

$$\max_z w_1 \cdot \phi_1(x, z) = \max_z \sum_j w_j \cdot \phi(x, z_j) + \sum_{j,k \in E} w_{jk} \cdot \phi(z_j, z_k) \tag{7}$$

where z_j is the location of part j, w_j is the template for part j, w_{jk} is a deformation model (spring) between part j and k, and E defines the edge structure in the star graph. We write $\phi(x, z_j)$ for the HOG feature extracted from location z_j and $\phi(z_j, z_k)$ for the squared relative offset between part j and k. Given training data triples (x_i, s_i, y_i), w can be trained with a latent SVM using the coordinate descent procedure outlined in [11] or the convex-concave procedure described in [27]. We use the publically-available coordinate descent code [28].

3.5 Latent Multiresolution Part Models

One limitation of the above model is that training data is still specified in terms of a fixed, discrete size s_i - all instances are either 50 or 100 pixels tall. Given a training window of arbitrary height x_i, one might resize it to 50 or 100 pixels by quantization. The correct quantization may be ambiguous for datasets such as PASCAL where many images of people are truncated to head and shoulder shots [29] – here a small bounding box may be better described with a truncated, high-resolution model. When the training data x_i is given as set of bounding box coordinates, [11] shows that one can significantly improve performance by estimating a latent location and scale of a "hidden" bounding box that sufficiently overlaps the given ground-truth bounding box.

We augment this procedure to also estimate the "hidden resolution" s_i of a training instance x_i. Training examples that are large will not have any low-resolution (e.g., 50-pixel tall) bounding boxes that overlap the given ground-truth coordinates. In these cases, the resolution is fixed to $s_i = 1$ and is no longer latent. Similarly, training instances that are very small will not have any high-resolution bounding boxes with sufficient overlap. However, there will be a collection of training instances of "intermediate" size that could be processed as low or high-resolution instances. The values of s_i will be treated as latent and estimated through the latent SVM framework: starting with a random initialization of latent s_i and z_i values, (1) a model/weight-vector w is trained through convex optimization, and (2) the model is used to relabel an example x_i with a latent resolution state s_i and part location z_i that produces the best score.

Relationship to mixture models: It is relevant to compare our model to the mixture models described in [23]. One might view our multiresolution model as a mixture of two models. However, there are a number of important differences from [23]. Firstly, our components share many parameters, while those in [23]

do not share any. For example, we use both low and high resolution instances to learn a low-res "root" template, while [23] only uses high-resolution instances. Secondly, the mixture component variable s_i is treated differently in our framework. At test time, this variable is *not* latent because we know the size of a putative window that is being scored. At train time, the variable is treated as latent for a subset of training instances whose resolution is ambiguous.

Extensions: Though we have described two-layer multi-resolution models, extensions to hierarchical models of three or more layers in straightforward. For example, the head part of a pedestrian may be composed of an eye, nose, and mouth parts. One would expect such a model to be even more accurate. Note that such a model is still efficient to score because the edge structure E is now a tree rather than a star model, which is still amenable to dynamic programming. Training a single resolution hierarchical part model poses a difficulty since it cannot exploit the many training and testing instances where the details, e.g., of the eyes and nose, are *not* resolvable. Our multiresolution formalism provides a framework to manage this complexity during both training and testing.

4 Multiresolution Contextual Models

We now augment our analysis of resolution to consider the effects of contextual reasoning. Our hypothesis, to be borne out by experiment, is that context plays a stronger role in detecting small-scale instances. Toward that end, we add a simple but effective contextual feature for pedestrian detection - ground plane estimation. Hoeim et. al. [1] clearly espouse the benefit of ground plane estimation for validating the observed locations and scales of putative detections. One approach would be to treat the ground plane as a latent variable to be estimated for each frame or video. We take a simpler approach and assume that the training and test data are collected in similar conditions, and so apply a ground-plane model learned from the training data at test time. We begin with the following assumptions:

1. The camera is aligned with the ground plane
2. Pedestrians have roughly the same height
3. Pedestrians are supported by a ground plane

Given the above and a standard perspective projection model, it is straightforward to show that there exists a linear relationship between the projected height of a detection (h) and the y-location of the lower edge of its bounding box in the image (y):

$$h = ay + b \tag{8}$$

Features: One reasonable contextual feature is to penalize the score of a detection in proportion to the squared deviation from the model:

$$(h - (ay + b))^2 = w_p \cdot \phi_p(x) \quad \text{where} \quad \phi_p(x) = \begin{bmatrix} h^2 & y^2 & hy & h & y & 1 \end{bmatrix}^T \tag{9}$$

where we have assumed the image features x include the location and height of the image window, and where model parameters w_p implicitly encode both the parameters of the ground plane and the amount to penalize detections which deviate from the ground plane model.

Our intuition says that low-resolution models should strongly penalize deviations because the local template will generate false positives due to its limited resolution. Alternately, the high-resolution model should not strongly penalize deviations because the local template is more accurate and the assumptions do not always hold (people are not all the same height). We investigate these possibilities experimentally using different encodings of our contextual features, including augmenting $\Phi(x, z, s)$ with a single set of perspective features $\phi_p(x)$ used across both low and high resolution models, or a separate set of features for each resolution ($\phi_p^0(x)$ and $\phi_p^1(x)$).

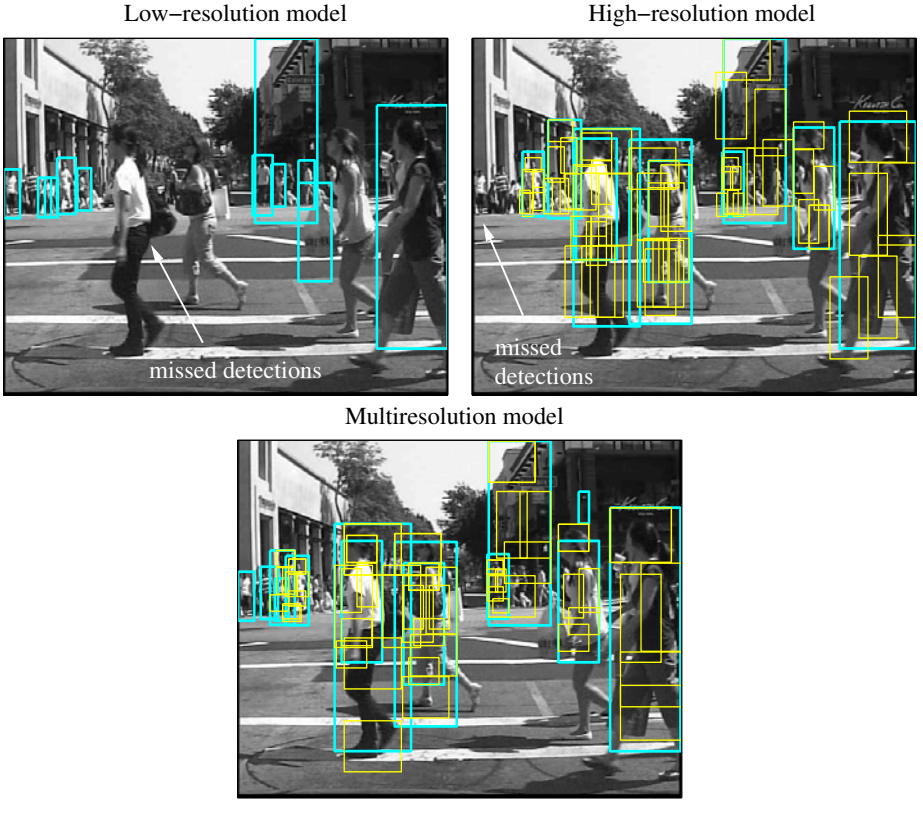


Fig. 3. On the **left**, we show the result of our low-resolution rigid-template baseline. One can see it fails to detect large instances. On the **right**, we show detections of our high-resolution, part-based baseline, which fails to find small instances. On the **bottom**, we show detections of our multiresolution model that is able to detect both large and small instances. The threshold of each model is set to yield the same rate of FPPI of 1.2.

5 Experimental Results

Implementation: We implemented our final context-augmented multiresolution model through fairly straightforward modification to the online multiscale part-based code [28]. In both the benchmark and diagnostic evaluation, we compare to the original code as a baseline. The contextual model decribed in the following results use scale-specific contextual features ($\phi_p^0(x)$ and $\phi_p^1(x)$), which we found slightly outperformed a single-scale contextual feature (though this is examined further in Sec.5.2).

5.1 Benchmark Results

We submitted our system for evaluation on the Caltech Pedestrian Benchmark [2]. The benchmark curators scored our system using a battery of 11 experiments

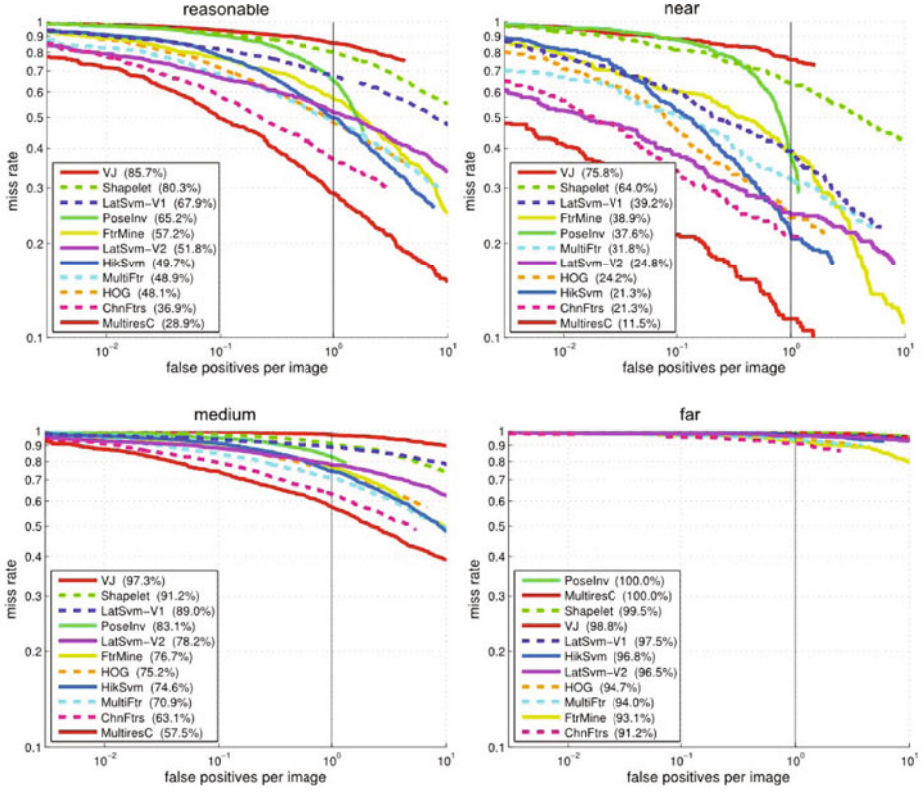


Fig. 4. Benchmark results. From the upper left graph in **clockwise** direction, we show the results for *reasonable*, *near*, *far* and *medium* experiments, evaluated on test instances with various heights ($h > 30$, $h > 80$, $h < 30$, and $30 < h < 80$ and $h < 30$, respectively). Our context-augmented multiresolution model, labeled as **MultiresC**, significantly outperforms all previous systems in 10 out of the 11 benchmark experiments (all but the 'far' experiment').

on a held-out testset, designed to analyze performance in different regimes depending on object scales, aspect ratios, and levels of occlusion (Fig. 4). The results are impressive - *our system outperforms all previously-reported methods*, across the entire range of FPPI (false positives per image) rates, in 10 out of 11 experiments. The sole experiment for which we do not win is the far-scale experiment, in which all detectors essentially fail. Even given our multiresolution model, finding extremely small objects is a fundamentally difficult problem because there is little information that can be extracted in such instances.

Our results are particularly impressive for the near-scale experiment, where we *halve* the previous-best miss rate at 1 FPPI [30]. Previous approaches, including the multiscale part-based model of [23], use fixed-resolution detectors that tend to be tuned for the small-scale regime so as to correctly fire on the set of small instances in this dataset. Our multiresolution model leverages the additional pixels available in large instances to significantly boost performance.

5.2 Diagnostic Experiments

To further analyze the performance of our system, we construct a set of diagnostic experiments by splitting up the publically-available Caltech Pedestrian training data into a disjoint set of training and validation videos. We defined this split pseudo-randomly, ensuring that similar numbers of people appeared in both sets. We compare to a high-resolution baseline (equivalent to the original part-based code [28]) and a low-resolution baseline (equivalent to a root-only model [13]), and a version of our multiresolution model without context. We

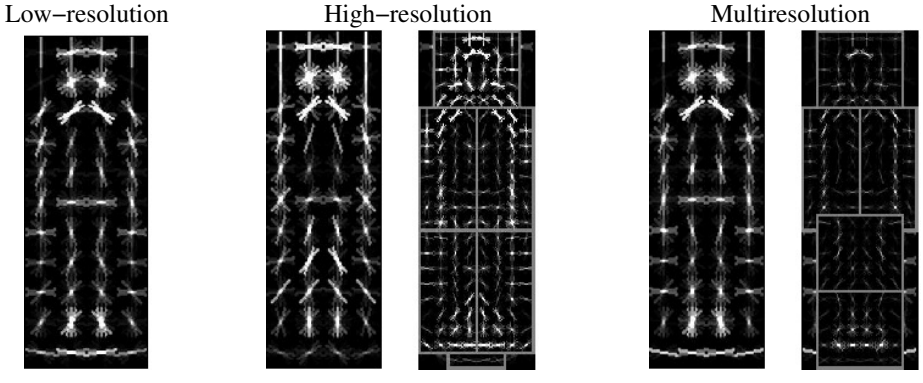


Fig. 5. On the **left**, we visualize our low-resolution rigid-template. In the **middle**, we visualize the high-resolution part-based template of [11] trained on Caltech pedestrians. Note the root templates look different, as only a small portion of the training data (of high enough resolution) is used to train the part-model. On the **right**, we visualize the multiresolution model. Note that the root component looks similar to the low-resolution model. Also note that the parts overall have weaker weights. This suggests that much of the overall score of the multiresolution model is given by the root score. However, it is still able to detect both small and large instances as shown in our results.

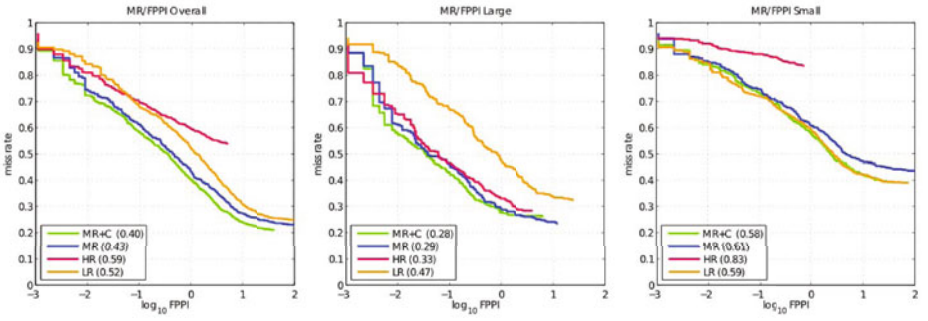


Fig. 6. Results of diagnostic experiments. We compare results to fixed resolution baselines, where "LR" is a low-resolution rigid template and "HR" is a high-resolution part-based model. On the **left**, we show results evaluated on the full set of test instances from validation data. In the **middle**, we show results for large-instance (> 90 pixels). On the **right**, we show the results on small-instances (< 90 pixels). The "LR" template performs well on small instances, while the "HR" template performs well on large instances. Our multiresolution "MR" model exploits the best of both, in the appropriate regimes. Our context-augmented model "MR+C" provides a small improvement overall, but a noticable improvement when detecting small instances at a higher FPPI rate.

visualize our baseline models in Fig. 5. All methods are trained and evaluated on the exact same data. To better interpret results, we threw out instances that were very small (< 30 pixels in height) or abnormal in aspect ratio (i.e. $h/w > 5$), as we view the latter as an artifact of annotating video by interpolation.

Overall: Overall, our multiresolution model outperforms baseline models. Our contextual model provides a small but noticeable improvement, reducing the missed detection rate from 43% to 40%. We shall see that the majority of this improvement comes from detecting small-scale instances. Somewhat surprisingly, we see that a simple rigid template outperform a more sophisticated part model - 52% MD compared to 59%. One can attribute this to the fact that the part-based model has a fixed resolution of 88 pixels (selected through cross-validation), and so cannot detect any instances which are smaller. This significantly hurts performance as more than 80% of instances fall in this small category. However, one may suspect that the part-model should perform better when evaluating results on test instances that are 88 pixels or taller.

Detecting large instances: When evaluating on large instances (> 90 pixels in height), our multiresolution model performs similarly to the high-resolution part-based model. Both of these models provide a stark improvement over a low-resolution rigid template. We also see that perspective context provides no observable improvement. One might argue that this is due to a weak contextual feature, but we next show that it does provide a strong improvement for small scale detections.

Detecting small instances: When evaluating on small instances (< 90 pixels in height), we see that the part-based model performs quite poorly, as it is

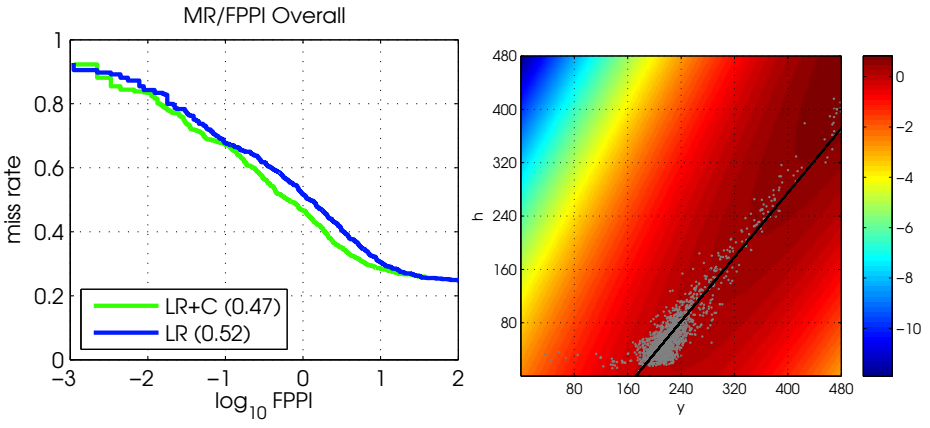


Fig. 7. We show the effectiveness of our perspective features on low-resolution models. Overall performance increases from 51% MD to 46% MD. We visualize our perspective features on the **right**. We plot the distribution of h and y (bounding box height and image-y locations) in the ground truth data, and plot the score $w_p \cdot \phi_p(x)$ as a function h and y. We also display the distribution of ground truth (visualized with a point cloud) along with its linear fit. We see that the learned contextual features penalize detections whose heights and image-y locations are not consistent with the ground plane.

unable to detect the majority of test instances which are small. Our multiresolution model performs slightly worse than a low-resolution model (61% compared to 59%). Perspective features provide a noticeable improvement for our multiresolution model, increasing performance from 61% MD to 58%.

Context features: To verify that our contextual features are indeed reasonable, we analyze the benefit of our contextual features on a low-resolution model. We see a noticeable reduction in the MD rate from 51% to 46%, suggesting our contextual features are indeed fairly effective. Their effect is diminished in our multiresolution model because the part-based model is able to better score large-scale instances, reducing the need for score adjustment using context.

6 Conclusion

We describe a simple but effective framework for merging different object representations, tuned for different scale-regimes, into a single coherent multiresolution model. Our model exploits the intuition that large instances should be easier to score, implying that one should adapt representations at the instance-level. We also demonstrate that context should be similarly adapted at the instance-level. Smaller objects are more difficult to recognize, and it is under this regime that one should expect to see the largest gains from contextual reasoning. We demonstrate impressive results on the difficult but practical problem of finding large and small pedestrians from a moving vehicle.

Acknowledgements

Funding for this research was provided by NSF grants 0954083 and 0812428, and a UC Labs research program grant.

References

1. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. International Journal of Computer Vision 80(1), 3–15 (2008)
2. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2009)
3. Lindeberg, T.: Scale-space theory in computer vision. Springer, Heidelberg (1994)
4. Lowe, D.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60, 91–110 (2004)
5. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. International Journal of Computer Vision 60, 63–86 (2004)
6. Fergus, R., Perona, P., Zisserman, A., et al: Object class recognition by unsupervised scale-invariant learning. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (2003) Citeseer
7. Dorko, G., Schmid, C.: Selection of scale-invariant parts for object class recognition. In: ICCV 2003 (2003) Citeseer
8. Mallat, S., Zhong, S.: Characterization of signals from multiscale edges. IEEE Transactions on pattern analysis and machine intelligence 14, 710–732 (1992)
9. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T.: Pedestrian detection using wavelet templates. In: IEEE CVPR, pp. 193–199 (1997)
10. Schneiderman, H., Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE Computer Society, Los Alamitos (1999/2000)
11. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Computer Vision and Pattern Recognition, Anchorage, USA (June 2008)
12. Gavrila, D.: Pedestrian detection from a moving vehicle. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 37–49. Springer, Heidelberg (2000)
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. I886–I893 (2005)
14. Enzweiler, M., Gavrila, D.M.: Monocular pedestrian detection: Survey and experiments. IEEE PAMI 31, 2179–2195 (2009)
15. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE PAMI 23, 349 (2001)
16. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE PAMI 30, 1713–1727 (2008)
17. Wang, X., Han, T.X., Yan, S.: An HOG-LBP human detector with partial occlusion handling. International Journal of Computer Vision (2009)
18. Lin, Z., Hua, G., Davis, L.S.: Multiple instance feature for robust part-based object detection. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 405–412 (2009)
19. Maji, S., Berg, A., Malik, J.: Classification using intersection kernel SVMs is efficient. In: IEEE Conf. on Computer Vision and Pattern Recognition (2008)

20. Schwartz, W., Kembhavi, A., Harwood, D., Davis, L.: Human detection using partial least squares analysis. International Journal of Computer Vision (2009)
21. Wojek, C., Walk, S., Schiele, B.: Multi-cue onboard pedestrian detection. In: IEEE Conf. on Computer Vision and Pattern Recognition (2009)
22. Gavrila, D.M., Munder, S.: Multi-cue pedestrian detection and tracking from a moving vehicle. International journal of computer vision 73, 41–59 (2007)
23. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE PAMI 99(5555)
24. Dollar, 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)
25. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: Proc. CVPR, pp. 1–8 (2007)
26. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models of multi-class object layout. In: ICCV (2009)
27. Yu, C., Joachims, T.: Learning structural SVMs with latent variables. In: Proceedings of the 26th Annual International Conference on Machine Learning. ACM, New York (2009)
28. http://people.cs.uchicago.edu/~pff/latent
29. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results (2007), http://www.pascal-network.org/challenges/VOC/voc2007/workshop
30. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: BMVC (2009)

Accurate Image Localization Based on Google Maps Street View

Amir Roshan Zamir and Mubarak Shah*

University of Central Florida, Orlando FL 32816, USA

Abstract. Finding an image's exact GPS location is a challenging computer vision problem that has many real-world applications. In this paper, we address the problem of finding the GPS location of images with an accuracy which is comparable to hand-held GPS devices.We leverage a structured data set of about 100,000 images build from Google Maps Street View as the reference images. We propose a localization method in which the SIFT descriptors of the detected SIFT interest points in the reference images are indexed using a tree. In order to localize a query image, the tree is queried using the detected SIFT descriptors in the query image. A novel GPS-tag-based pruning method removes the less reliable descriptors. Then, a smoothing step with an associated voting scheme is utilized; this allows each query descriptor to vote for the location its nearest neighbor belongs to, in order to accurately localize the query image. A parameter called *Confidence of Localization* which is based on the Kurtosis of the distribution of votes is defined to determine how reliable the localization of a particular image is. In addition, we propose a novel approach to localize groups of images accurately in a hierarchical manner. First, each image is localized individually; then, the rest of the images in the group are matched against images in the neighboring area of the found first match. The final location is determined based on the *Confidence of Localization* parameter. The proposed image group localization method can deal with very unclear queries which are not capable of being geolocated individually.

1 Introduction

Determining the exact GPS location of an image is a task of particular interest. As there are billions of images saved in online photo collections - like Flickr, Panoramio etc. - there is an extant resource of information for further applications [1,2]. For example, in Agarwal et al. [1], a structure from motion approach is employed to find the 3D reconstruction of Rome using GPS-tagged images of the city. Many such applications need some sort of information about the exact location of the images; however, most of the images saved on the online repositories are not GPS-tagged. A system that is capable of finding an exact

* The authors would like to thank Jonathan Poock for his valuable technical contributions and comments on various drafts of the submission, which have significantly improved the quality of the paper.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 255–268, 2010.

location using merely visual data can be used to find the GPS-tag of the images and thus make the huge number of non-GPS-tagged images usable for further applications.

However, there are many images which are incapable of being localized individually, due to their low quality, small size or noise. Many of these images are saved in albums or image groups; these groupings can act as clues to finding the exact location of the unclear image. For instance, images saved in online photo collections in an album usually have locations that are close to one another.

Visual localization of images is an important task in computer vision. Jacobs et al. [3] use a simple method to localize webcams by using information from satellite weather maps. Schindler et al. [4] use a data set of 30,000 images for geolocating images using a vocabulary tree [5]. The authors of [6] localize landmarks based on image data, metadata and other sources of information. Kalogerakis et al. [7] leverage images in a sequence to localize them in a global way. In their method, they use some travel priors to develop the chronological order of the images in order to find the location of images. Zhang et al. [8] perform the localization task by matching image key points and then applying a geometrical alignment. Hakeem et al. [9] find the geolocation and trajectory of a moving camera by using a dataset of 300 reference images. Although much research has been done in the area of localizing images visually, many other sources of information can be used alongside the visual data to improve the accuracy and feasibility of geolocation, such as used in Kalogerakis et al. [7]. To the best of our knowledge, image localization utilizing groups of images has not been investigated; as such, this paper claims to be the first to use the proximity information of images to aid in localization.

In our method, a query image is matched against a GPS-tagged image data set; the location tag of the matched image is used to find the accurate GPS location of the query image. In order to accomplish this, we use a comprehensive and structured dataset of GPS-tagged Google Maps Street View images as our reference database. We extract SIFT descriptors from these images; in order to expedite the subsequent matching process, we index the data using trees. The trees are then searched by a nearest-neighbor method, with the results preemptively reduced by a pruning function. The results of the search are then fed through a voting scheme in order to determine the best result among the matched images. Our proposed *Confidence of Localization* parameter determines the reliability of the match using the Kurtosis of the voting distribution function. Also, we propose a method for localizing group of images, in which each image in the query group is first localized as a single image. After that, the other images in the group are localized within the neighboring area of the detected location from the first step. A parameter called CoL_{group} is then used to select the rough area and associated corresponding accurate locations of each image in the query group. The proposed group localization method can determine the correct GPS location of images that would be impossible to geolocate manually. In the results section, we show how our proposed single and group image localization methods are significantly more accurate than the current methods.

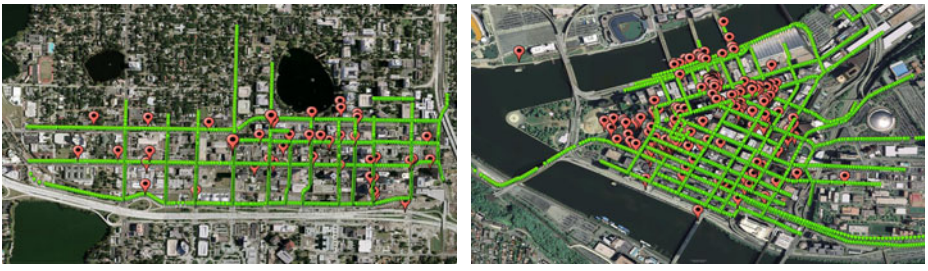


Fig. 1. We use a dataset of about 100,000 GPS-tagged images downloaded from Google Maps Street View for Pittsburg, PA (Right) and Orlando, FL (left). The green and red markers are the locations of reference and query images respectively.

2 Google Maps Street View Dataset

Different type of image databases have been used for localization tasks. In Hakeem et al. [9] a database of 300 GPS-tagged images is used, whereas Kalogerakis et al. [7] leverage a dataset of 6 million non-structured GPS-tagged images downloaded from internet, and Schindler et al. [4] use a data set of 30,000 street-side images. We propose using a comprehensive 360° structured image dataset in order to increase the accuracy of the localization task. The images extracted from Google Maps Street View are a very good example of such a dataset. Google Maps Street View is a very comprehensive dataset which consists of 360° panoramic views of almost all main streets and roads in a number of countries, with a distance of about 12m between locations. Using a dataset with these characteristics allows us to make the localization task very reliable, with respect to feasibility and accuracy; this is primarily due to the comprehensiveness and organization of the dataset. The following are some of the main advantages of using datasets such as Google Maps Street View:

• Query Independency: Since the images in the dataset are uniformly distributed over different locations, regardless of the popularity of a given location or object, the localization task is independent of the popularity of the objects in the query image and the location.

• Accuracy: As the images in the data set are spherical 360° views taken about every 12 meters, it is possible to correctly localize an image with a greater degree of accuracy than would be permitted by a sparser data set comprised of non-spherical images. The achieved accuracy is comparable to - and, in some cases, better than - the accuracy of hand-held GPS devices.

• Epipolar Geometry: The comprehensiveness and uniformity of the data set makes accurate localization possible without employing methods based on epipolar geometry [9]- methods which are usually computationally expensive and, in many cases, lacking in required robustness. Additionally, the camera's intrinsic parameters for both the query and the dataset images are not required in order to accurately localize the images.

• Secondary Applications: Using a structured database allows us to derive additional information, without the need for additional in-depth computation. For example, camera orientation can be determined as an immediate result of localization using the Google Maps Street View data set, without employing methods based on epipolar geometry. Since the data set consists of 360° views, the orientation of the camera can be easily determined just by finding which part of the 360° view has been matched to the query image - a task that can be completed without the need for any further processing. Localization and orientation determination are tasks that even hand-held GPS devices are not capable of achieving without motion information.

However, the use of the Google Maps Street View dataset introduces some complications as well. The massive number of images can be a problem for fast localization. The need for capturing a large number of images makes using wide lenses and image manipulation (which always add some noise and geometric distortions to the images) unavoidable. Storage limitations make saving very high quality images impossible as well, so a matching technique must be capable of dealing with a distorted, low-quality, large-scale image data set. The database's uniform distribution over different locations can have some negative effects - while it does make the localization task query-independent, it also limits the number of image matches for each query as well. For example, a landmark will appear in exactly as many images as a mundane building. This is in direct contrast to other current large scale localization methods like Kalogerakis et al. [7], which can have a large number of image matches for a location in their database - a fact especially true if a location is a landmark; this allows the localization task to still be successful on a single match. The small number of correct matches in our database makes the matching process critical, as if none of the correct matches - which are few in number - are detected, the localization process fails.

We use a dataset of approximately 100,000 GPS-tagged Google Street View images, captured automatically from Google Maps Street View web site from Pittsburgh, PA and Orlando, FL. The distribution of our dataset and query

Fig. 2. Sample Reference Images. Each row shows one placemark's side views, top view and map location.

images are shown in Fig. 1. The images in this dataset are captured approximately every 12 meters. The database consists of five images per placemark: four side-view images and one image covering the upper hemisphere view. These five images cover the whole 360° panorama. By contrast, Schindler et al.'s [4] dataset has only one side view. The images in their dataset are taken about every 0.7 meters, covering 20km of street-side images, while our dataset covers about 200km of full 360° views. Some sample dataset images are illustrated in Fig. 2.

3 Single Image Localization

Many different approaches for finding the best match for an image has been examined in the literature. Hakeem et al. [9] perform the search process by nearest-neighbor search among SIFT descriptors of a small dataset of about 300 reference images. Kalogerakis et al. [7] perform the task by calculating a number of low-level features - such as color histograms and texton histograms - for 6 million images while assuming that there is a very close match for the query image in their dataset. Schindler et al. [4] try to solve the problem by using the bag of visual words approach. In the results section, we show that the approach in Schindler et al. [4] cannot effectively handle large-scale datasets that are primarily comprised of repetitive urban features. In order to accurately localize images, we use a method based on a nearest-neighbor tree search, with pruning and smoothing steps added to improve accuracy and eliminate storage and computational complexity issues.

During training, we process the reference dataset by computing the SIFT descriptors [10] for all interest points detected by the SIFT detector [10,11]. Then, the descriptor vectors (and their corresponding GPS tags) are organized into a tree using FLANN [12] . As we show later, a well-tuned pruning method allows us to find very reliable descriptors; as such, we generally need to compute at most $\frac{1}{6}$ of the number of interest points that Schindler et al. [4]'s method

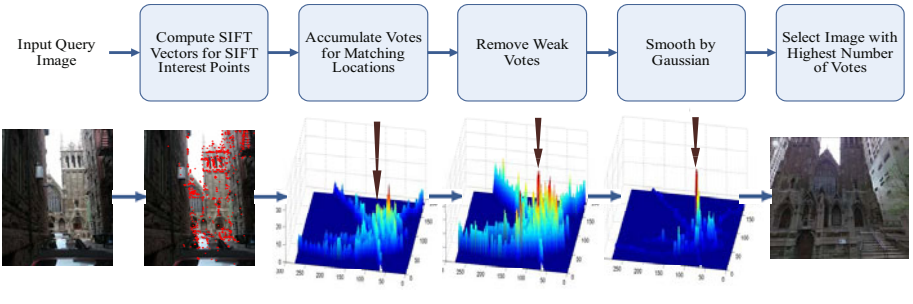


Fig. 3. Block diagram of localization of a query image. Lower row shows the corresponding results of each step for the image. Note the streets in the vote plots, as the votes are shown over the actual map. The dark arrow points toward the ground truth location. The distance between the ground truth and matched location is 17.8m.

requires. Fig. 3 shows the block diagram of the proposed method for localizing a query image. In the first step, the SIFT descriptors are computed for SIFT interest points in the same way as we process the dataset during training. Then, in the second step, the nearest-neighbors for each of the query SIFT vectors are found in the tree. Each of the retrieved nearest-neighbors vote for the image that they belong to. The votes can be shown as a plot over the actual map of the area covered by our reference dataset (as shown in third column of Fig. 3).

As noisy interest points are commonly detected in an image, a pruning step is essential. Lowe et al. [10] find reliable matches by setting a maximum threshold of 0.8 on the ratio of the distance between the query descriptor and the first and second nearest neighbors. For geolocation tasks in large-scale datasets, the pruning step becomes more important; this is primarily because many of the processed descriptors belong to non-permanent and uninformative objects (ie. vehicles, people, etc), or are detected on the ground plane - both cases where the descriptors become misleading for geolocation purposes. The massive number of descriptors in the dataset can add noisy, unauthenticated matches as well. Schindler et al. [4] find the more informative visual words by maximizing an information gain function, a process which requires reference images with significant overlap. Hakeem et al. [9] prune their dataset by setting the maximum SIFT threshold proposed in Lowe et al. [10] to 0.6 in order to keep more reliable matches. We propose using the following function in order to prune the matches:

$$V_{flag}(d_i) = \begin{cases} 1 & \frac{NN(d_i,1)}{NN(d_i,Min\{j\})} < 0.8 \\ & \forall j \rightarrow |Loc(NN(d_i,1)) - Loc(NN(d_i,j))| > D \\ 0 & otherwise \end{cases} , \tag{1}$$

where $V_{flag}(d_i)$ is the flag of the vote corresponding to the query descriptor d_i. If the flag is 0, the descriptor is removed in the pruning step; if the flag is 1, it participates in the voting. $NN(d_i, k)$ is the k_{th} nearest-neighbor of d_i. $Loc(NN(d_i, k))$ is the GPS location of the k_{th} nearest-neighbor to descriptor d_i and $||$ represents the actual distance between the two GPS locations of the nearest neighbor. At its core, Eq. 1 may appear to be the SIFT ratio [10]; the changes we have made mean that the descriptor in the denominator is dynamically determined, based on actual GPS distance. This is an important difference, as allowing this ratio to be determined dynamically creates a great advantage over the simple ratio between first and second nearest-neighbors used in Lowe et al. [10] and Hakeem et al. [9], in that it allows the localization task to handle repeated urban structures more accurately. The importance of this method becomes clearer by considering the reference images shown in Fig. 2. The windows of the skyscraper shown in the 3_{rd} column, 3_{rd} row of the figure are identical, leading to very close nearest-neighbor results for a query descriptor of this window (as shown in bottom left corner image in Fig. 4). While the SIFT ratio used in Lowe et al. [10] and Hakeem et al. [9] removes this descriptor in the pruning step, the proposed method retains it, as the location of all of the very similar nearest neighbors are close to each other. In other words, even though we cannot necessarily determine which of the windows shown in the query image

correspond to each of the windows in the skyscraper, they will still be voting for the correct location, as the GPS-tag of all these very similar nearest-neighbors point to one location. To explain it in a less-anecdotal way, Eq. 1 removes a descriptor only if the descriptor in the denominator does not belong to any of the nearby locations of the first nearest-neighbor AND the ratio is greater than 0.8. As can be seen in the 4_{th} column of Fig. 3, the votes around the ground truth location are mostly retained, whereas many of the incorrect votes are removed.

Since some of the objects in a query image may be in view of several reference images, we smooth the votes based on their locations in order to prevent the votes from being scattered using this equation:

$$V_{smoothed}(\lambda', \psi') = \sum_{\lambda=-\infty}^{+\infty} \sum_{\phi=-\infty}^{+\infty} e^{-(\frac{\lambda^2+\phi^2}{2\sigma'^2})} V(\lambda' - \lambda, \phi' - \phi) V_{flag}(\lambda' - \lambda, \phi' - \phi) \ , \tag{2}$$

where $V(\lambda, \phi)$ and $V_{flag}(\lambda, \phi)$ are the voting and flags function (respectively), for the GPS location specified by λ and ϕ, and the first coefficient is the 2D Gaussian function with a standard deviation of σ'. As each descriptor is associated with a GPS-tagged image, we can represent the voting function's parameter in terms of λ and ϕ. As can be seen in column 5 of Fig. 3, the smoothing step makes the peak which corresponds to the correct location more distinct.

As shown in the block diagram in Fig. 3, the location which corresponds to the highest peak is selected as the GPS location of the query image.

3.1 Confidence of Localization

There are several cases in which a query image may - quite simply - be impossible to localize. For instance, a query might come from an area outside of the region covered by the database; alternatively, the image might be so unclear or noisy that no meaningful geolocation information can be extracted from it. A parameter that can check for (and, consequently, prevent) these kind of positive errors is important. In probability theory, statistical moments have significant applications. The Kurtosis is a measure of whether a distribution is tall and slim or short and squat [13]. As we are interested in examining the behavior of the voting function in order to have a measure of reliability, we normalize it and consider it as a probability distribution function. Since the Kurtosis of a distribution can represent the peakedness of a distribution, we propose to use it as a measure of *Confidence of Localization*, since a tall and thin vote distribution with a distinct peak corresponds to a reliable decision for the location; correspondingly, a widely-spread one with a short peak represents a poor and unreliable localization. Our *Confidence of Localization* parameter is thus represented by the following equation:

$$CoL = Kurt(V_{smoothed}) = -3 + \frac{1}{\sigma^4} \sum_{\phi=-\infty}^{+\infty} \sum_{\lambda=-\infty}^{+\infty} [(\lambda - \mu_\lambda)^2 (\phi - \mu_\phi)^2] V_{smoothed}(\lambda, \phi), \tag{3}$$

where $V_{smoothed}$ is the vote distribution function (see Eq. 2). The above equation is the Kurtosis of the 2D vote distribution function, with random variables λ and ϕ, corresponding to the GPS coordinates. A high Kurtosis value represents a distribution with a clearer and more defined peak; in turn, this represents a higher confidence value. In the next section, we use this CoL parameter to localize a group of images.

4 Image Group Localization

We propose a novel hierarchical approach to localize image groups. The only assumption inherent in the proposed method is that all of the images in the group must have been taken within the radial distance R of each other; this radial distance R is a parameter that can be set in the method. In our approach, no information about the chronological history of the images is required.

To localize an image group consisting of images I_1 to I_N, we employ a hierarchical approach consisting of two steps:

• Step 1, Individual Localization of Each Image: In the first step of the approach, all of the images in the group are localized individually, independent from other images. In order to do this, we use the Single Image Localization method described previously in section 3; thus, each one of the single images in the group returns a GPS location.

• Step 2, Search in Limited Subsets: In the second step, N subsets of reference images which are within the distance R of each of the N GPS locations found in step 1 are constructed. Following that, a localization method - similar to the method defined in section 3 - is employed for localizing the images in the group; however, in this case, the dataset searched is limited to each of the N subsets created by the initial search. We define the CoL value for each of the secondary, sequential search processes done in each of the limited subsets as:

$$CoL_{group}(S) = \sum_{i=1}^{N} \frac{CoL_i}{N} \ , \tag{4}$$

where S represents each of the secondary search processes. Once the CoL_{group} value for each of the limited subsets is calculated, the subset that scores the highest value is selected as the rough area of the image group. From there, each query image is assigned the GPS location of the match that was found in that limited subset.

Since this proposed approach to image group localization requires multiple searches in each step, the computational complexity of the method is of particular interest. The number of necessary calculations for localizing a single query image in our method is dependent on the number of detected interest points in the image. If we assume C is a typical number representing the number of required calculations for localizing an image individually, the number of required calculations to localize a group of images using the proposed approach is:

$$CE_{group}(N, \delta) = C(N + \frac{(N-1)N}{\delta}) \ , \tag{5}$$

where N is the number of images in the group and δ is a constant that is determined by the size of the limited subsets used in the step 2 of section 4. δ ranges from 1 to ∞, where 1 means each limited subset is as large as the whole dataset and ∞ means each subset is extremely small. Since the number of required calculations to localize an image individually is C, the number of required calculations to localize N images individually will be $N \times C$, so the percentage increase in computational complexity using the proposed group method vs. the individual localization method is :

$$PE(N, \delta) = \frac{CE_{group}(N, \delta) - N \times C}{N \times C} \times 100 \ , \tag{6}$$

i.e.,

$$PE(N, \delta) = \frac{N - 1}{\delta} \times 100 \ , \tag{7}$$

For 4 and 50 - both typical values for N and δ, respectively - the increase in compuational complexity is 24%, garnering a roughly three-fold increase in system accuracy.

5 Experiments

Our test set consists of 521 query images. These images are all GPS-tagged, user-uploaded images downloaded from online photo-sharing web sites (Flickr, Panoramio, Picasa, etc.) for Pittsburgh, PA and Orlando, FL. Only indoor images, privacy-infringing images and irrelevant images (e.g. an image which only shows a bird in the sky), are manually removed from the test set. In order to ensure reliability of results, all the GPS tags of the query images are manually checked and refined, as the user-tagged GPS locations are usually very noisy and inaccurate. Fig. 4 depicts some of the images.

311 images out of the 521 query images are used as the test set for the single-image localization method; 210 images are organized in 60 groups of 2,3,4 and 5 images with 15 groups for each as the test set for group image localization method.

5.1 Single Image Localization Results

Fig. 5 shows the results of the localization task for the test set of 311 images. In order to avoid computational issues of indexing the large number of images in a single tree, we construct 5 individual trees spanning the whole dataset. The final nearest-neighbor selected is chosen from among the 5 nearest-neighbor results retrieved across each tree. In these experiments, the queries and reference images of both of the cities are used. In order to make the curves in Fig. 5 invariant with respect to differing test sets, we randomly divide the single image localization method's test set into ten smaller test sets; likewise, we divide the group image localization method's test set into 5 smaller test sets. The curves in Fig. 5 are the average of the result curves generated for each of the smaller test sets. As

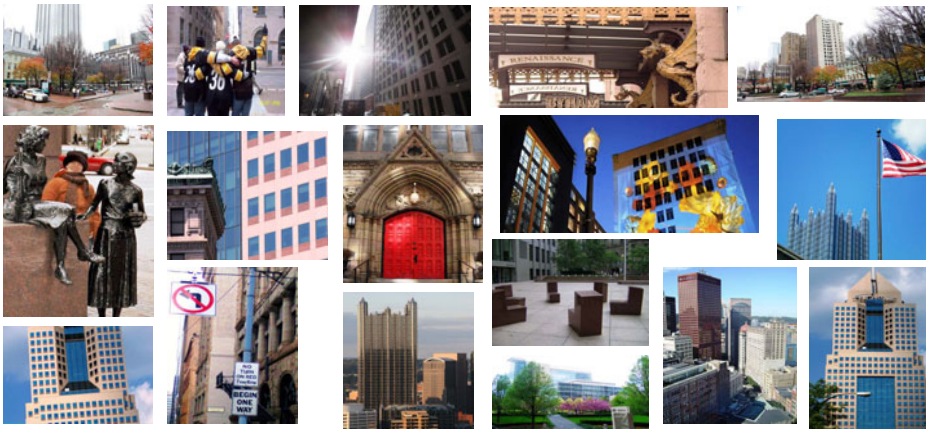


Fig. 4. Sample query images in our test set

can be seen in Fig. 5, all of the steps proposed in Fig. 3 improve the accuracy significantly. The smoothing step unifies the votes, leading to a more distinct correct peak, while attenuating the incorrect votes. Dynamic pruning removes the wrong matches, bringing about a more accurate localization task; this enables us to calculate and save fewer SIFT descriptors per image. By comparison, we have found (on average) 500 SIFT interest points per image; in Schindler et al. [4], the implementation used about 3000 interest points. As can be seen in Fig. 5, our method shows a significant improvement over the bag of visual words method used by Schindler et al. [4]. This is mostly due to the fact that, in the very similar and repeated structures of an urban area, the information lost in the quantization becomes critical. Additionally, the method proposed in Schindler et al. [4] requires reference images with significant overlap to maximize the information gain function, an assumption which can lead to significant issues in large scale localization. As can be seen in Fig. 5, about 60% of the test set is localized to within less than 100 meters of the ground truth; by comparison, this number for the method by Schindler et al. [4] is about 22%. However, our method fails when images are extremely cluttered with non-permanent objects (e.g. cars, people) or objects of low informative values (e.g. foliage).

In order to examine the performance of the proposed CoL function, the distribution of the CoL values of the localization of the test set consisting of 311 images is shown in Fig. 6 versus the error distance. The 311 CoL values are grouped into 8 bins based on the CoL values; the mean error of each of the bin members are shown on the vertical axis. As observed in the figure, higher CoL values - due to distinct peaks in the vote distribution function - correspond to lower error, meaning the localization is more reliable. Since theoretically the value of the Kurtosis is not limited, we normalize the CoL values and show them ranging from 0 to 1 on the plot.

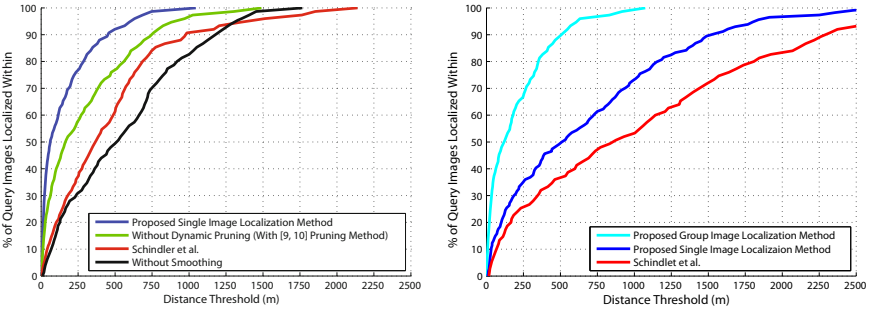


Fig. 5. The left figure shows the single image localization method results vs. Schindler et al.'s method, along with the curves representing the effect of each step. The right figure shows the localization results using the proposed image group localization method.

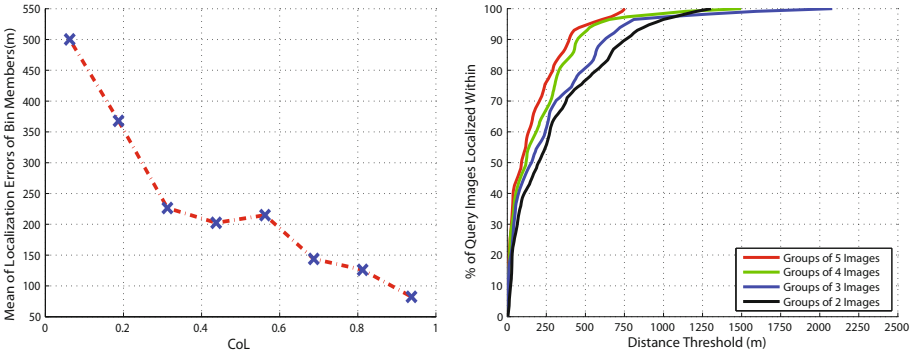


Fig. 6. The left figure shows the distribution of CoL values for the localization of the test set of 311 images. The CoL values are organized in 8 bins; the vertical axis shows the mean error value in meters for each bin. The right figure shows the breakdown of the results from the test set of the group image localization method based on the number of images in each group.

In order to show the importance of a parameter which represents the reliability of the localization task, we performed another experiment on CoL by using a test set of 62 query images. 34 of the images are from the Pittsburgh query set; 28 are from the Orlando query set. In this experiment, we grow one tree for each city, allowing the CoL function to determine the correct tree to use for each query. We localize each query image using each tree. Since a low CoL value for the tree to which the query image does not belong is expected, we select the location returned by the tree with higher CoL value as the final location of the query images. By this method, the proposed CoL parameter selected the correct location for 53 images out of the 64 test images - an accuracy of 82%. This

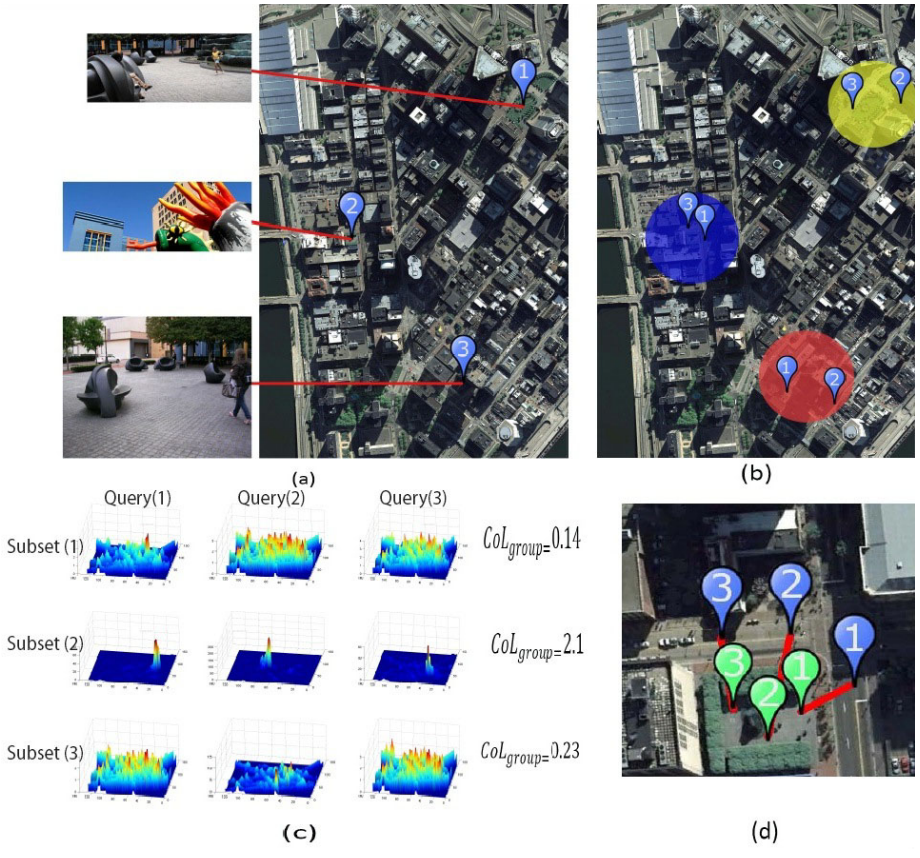


Fig. 7. An Example of Image Group Localization. (a):Query Images and Single Localization Results (b): Results of Search in Limited Subset. Each colored region is a different limited subset (c): Voting Surfaces and CoL_{group} for each query in each subset. (d): Blue Markers: Matched locations in the specific limited subset. Green markers represent the corresponding ground truth of queries. The red lines connect the ground truth with the respective correct match. The distances between the ground truth and final matched location are 10.2m, 15.7m and 11.4m, for queries 1, 2, and 3 respectively.

shows how a parameter representing the confidence of the localization can be of great assistance in preventing positive errors. More importantly, it can assist in extending the reference dataset as it may make reconstruction unnecessary.

5.2 Image Group Localization Results

Fig. 7 shows an example of localizing a set of images using the proposed method for geolocating image groups. The image group has 3 images, which are depicted on the left-hand side of Column (a). As discussed in Section 4, the first step of the proposed method is localization of images individually, resulting in a GPS

location for each image. Each query's individual localization is displayed on the map in Column (a). Column (b) shows the result of applying a search within the limited subset created by the initial search in step 1; the other two query images are localized around the initial points found in Column (a). Column (c) shows the voting surfaces for each query in each subset. As can be seen, Subset (2) has the most distinct peaks across all three queries; correspondingly, Subset (2) also has the highest CoL_{group} value and is thus selected as the correct set of matches. Finally, Column (d) shows an inset of the map corresponding to Subset (2) with the matched images represented by blue markers and the ground truth locations for the queries represented by green markers.

As discussed earlier, there are 210 images in our test set for group image localization. Most of the images were selected as they are (individually) very unclear and therefore challenging to localize; this was done in order to show how proximity information can be extremely helpful in localizing images that are incapable of being geolocated individually. We set the parameter R to 300 meters for our tests; this is a conservative assumption. This means that we assume that the images in one group are all taken within 300 meters of each other. The right column of Fig. 5, compares the performance of Schindler et al. [4]'s method, our proposed single image localization method, and the group image localization method. As can been seen, the use of proximity information results in a drastic improvement. The right plot in Fig. 6 shows the breakdown of the results of the test set from the group image localization method based on the number of images in the groups. As mentioned earlier, this set consists of groups of 2, 3, 4 and 5 images. As can be seen in Fig. 6, the accuracy of localization for groups with a larger number of images is greater, due to the fact that groups with a larger number of images will search more limited subsets. Consequently the chance of finding the correct location is higher.

6 Conclusion

In this paper we addressed the problem of finding the exact GPS location of images. We leveraged a large-scale structured image dataset covering the whole 360° view captured automatically from Google Maps Street View. We proposed a method for geolocating single images, specifically examining how the accuracy of current localization methods degenerates when applied to large-scale problems. First, we indexed the SIFT descriptors of the reference images in a tree; said tree is later queried by the SIFT descriptors of a query image in order to find each individual query descriptor's nearest neighbor. We proposed a dynamic pruning method which employed GPS locations to remove unreliable query descriptors if many similar reference descriptors exist in disparate areas. Surviving descriptors votes were then smoothed and then voted for the location their nearest neighbor reference descriptor belonged to. The reliability of the geolocation was represented by a proposed parameter called CoL, which was based on the Kurtosis of the vote distribution. Finally, a novel approach - using the proximity information of images - was proposed in order to localize groups of images. First, each image

in the image group was localized individually, followed by the localization of the rest of the images in the group within the neighborhood of the found location. Later, the location of each image within the rough area (Limited Subset) with the highest CoL_{group} value was selected as the exact location of each image.

References

1. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV (2009)
2. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. ACM Trans. Graph. 25, 835–846 (2006)
3. Jacobs, N., Satkin, S., Roman, N., Speyer, R., Pless, R.: Geolocating static cameras. In: ICCV (2007)
4. Schindler, G., Brown, M., Szeliski, R.: City-scale location recognition. In: CVPR, pp. 1–7 (2007)
5. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of the, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006, vol. 2, pp. 2161–2168. IEEE Computer Society, Los Alamitos (2006)
6. Crandall, D., Backstrom, L., Huttenlocher, D., Kleinberg, J.: Mapping the world's photos. In: International World Wide Web Conference (2009)
7. Kalogerakis, E., Vesselova, O., Hays, J., Efros, A., Hertzmann, A.: Image sequence geolocation with human travel priors. In: ICCV (2009)
8. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: 3DPVT 2006: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT 2006), pp. 33–40 (2006)
9. Hakeem, A., Vezzani, R., Shah, M., Cucchiara, R.: Estimating geospatial trajectory of a moving camera. In: International Conference on Pattern Recognition, vol. 2, pp. 82–87 (2006)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60 (2004)
11. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), http://www.vlfeat.org/
12. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISAPP (2009)
13. Balanda, K.P., MacGillivray, H.L.: Kurtosis: A critical review. The American Statistician 42, 111–119 (1988)

A Minimal Case Solution to the Calibrated Relative Pose Problem for the Case of Two Known Orientation Angles*

Friedrich Fraundorfer, Petri Tanskanen, and Marc Pollefeys

Computer Vision and Geometry Lab
Department of Computer Science
ETH Zürich, Switzerland
{fraundorfer,marc.pollefeys}@inf.ethz.ch,
tpetri@student.ethz.ch

Abstract. It this paper we present a novel minimal case solution to the calibrated relative pose problem using 3 point correspondences for the case of two known orientation angles. This case is relevant when a camera is coupled with an inertial measurement unit (IMU) and it recently gained importance with the omnipresence of Smartphones (iPhone, Nokia N900) that are equipped with accelerometers to measure the gravity normal. Similar to the 5-point (6-point), 7-point, and 8-point algorithm for computing the essential matrix in the unconstrained case, we derive a 3-point, 4-point and, 5-point algorithm for the special case of two known orientation angles. We investigate degenerate conditions and show that the new 3-point algorithm can cope with planes and even collinear points. We will show a detailed analysis and comparison on synthetic data and present results on cell phone images. As an additional application we demonstrate the algorithm on relative pose estimation for a micro aerial vehicle's (MAV) camera-IMU system.

1 Introduction

In this paper we investigate the case of computing calibrated relative pose for the case of two known orientation angles. This case is largely motivated by the availability of Smartphones (e.g. iPhone, Nokia N900) that are equipped with a camera and an inertial measurement unit (IMU). In the case of Smartphones the IMU mainly consists of accelerometers that allow to measure the earths gravity vector. From this measurement two orientation angles of the device and the embedded camera can be measured but usually not all three orientation angles. A similar situation arises when one is detecting vanishing points in the image. From a detected vanishing point it is also possible to compute two orientation angles [3], and we have the same case as with the accelerometer.

* This work was supported in parts by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant n.231855 (sFly) and by the Swiss National Science Foundation (SNF) under grant n.200021-125017.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 269–282, 2010.

For the relative pose problem this means that only one rotation angle and the three translation parameters are left to be computed from visual measurements. For this special case we will derive a simplified essential matrix and we will show that this leads to an algorithm that can compute the relative pose from three point correspondences only (similar to [5]), instead of the standard 5 point algorithm [10]. In analogy to the 5-point (6-point), 7-point, and 8-point algorithm for computing the essential matrix in the unconstrained case, we derive a 3-point, 4-point, and 5-point algorithm for the special case of two known orientation angles.

Reducing the number of point correspondences is of utmost importance when a RANSAC [1] scheme is used to cope with outliers in the data. The number of random samples to find one outlier free sample depends exponentially on the number of parameters to instantiate one hypothesis. The necessary number of samples to get an outlier free sample with a chance of 99% and an outlier ratio of 50% is 146 for the standard 5-point algorithm. The 3-point algorithm would only need 35 samples which is a speedup of a factor of 4. In R-RANSAC [8], where the termination criterion is, when the probability of missing a set of inliers larger than the largest support found so far, falls under a predefined threshold, a smaller sample size also improves the efficiency. In this sense the proposed 3-point algorithm will be much more efficient in computing relative pose than previous methods, which might be very important for using the method on Smartphones with limited computational power.

We also analyze the degeneracies of the new algorithm and we will show that it will work for planar scenes and even for the case of three collinear points.

In addition to a detailed description of the proposed algorithm, we will give a detailed performance analysis using synthetic data. We will test the algorithm under different levels of noise and more important under noise on the IMU measurements. For results on real data we show relative pose estimation on images from a Nokia N900 Smartphone. The IMU values of the N900 have a precision of 1 degree and we will show that this is good enough to be used for our algorithm. As an additional application we demonstrate the algorithm on relative pose estimation for the camera-IMU system of a micro aerial vehicle (MAV). Visual localization for MAV's is a very active field of research and the power and weight limitations of a MAV do not allow for the use of powerful computers. Therefore it is even more important to use very efficient algorithms, e.g. our proposed 3-point algorithm.

2 Related Work

The special case of knowing the full camera orientation is the case of pure translational motion. In this case the essential matrix can be computed linearly [3]. The case of knowing the translation and solving for the rotation only, using three points, was briefly described in [11]. The case of knowing the rotation partially, e.g. from vanishing points has already been investigated, e.g. for removing tilt in photographs [2] or rectifying images for visual localization [16]. However, the

knowledge of two rotations can directly be used to derive a simplified essential matrix that can be estimated from 3 point correspondences and the two rotation angles. This 3-point method has recently been investigated in [5] and this work is closely related to ours. They set up a polynomial equation system in the entries of the simplified essential matrix and use the Macaulay matrix method to solve it which gives 12 solutions for the essential matrix.

In our work we use a similar parameterization of the essential matrix, however we follow a different way of setting up and solving the polynomial system for the essential matrix. The method we propose leads to a 4th degree polynomial which results in up to 4 real solutions for the essential matrix, instead of 12 as in [5]. In terms of efficiency this is an important fact. The different essential matrices have to be verified with additional points (usually done within a RANSAC loop). In this sense our formulation with 4 solutions is much more efficient than the one with 12 solutions.

Our formulation is related to the 5-point algorithm [10] and to the 6-point algorithm [14] in the way the 4th degree polynomial is set up. In addition we also propose a linear 5-point algorithm and a 4-point algorithm which are analogies to the 8-point [6] and 7-point [4] algorithm for computing the essential matrix.

Other approaches that make use of IMU measurements perform Kalman filter fusion of the IMU measurements and the vision based pose estimates [9,12]. In our approach we propose a very tight coupling between IMU and vision measurements in a way that the IMU measurements simplify the vision based camera pose estimation. Another example of tight integration of IMU and visual measurements has also been proposed in [17]. IMU measurements have also been used in [15] together with GPS for large scale structure-from-motion. In this approach GPS and IMU values have been used as initial values for bundle adjustment.

3 Estimating the Essential Matrix for the Case of Two Known Orientation Angles

In this section we derive the parameterization of the essential matrix for the case of two known orientation angles. We identify additional linear constraints in the parameters of E which make it possible to compute a minimal solution for the essential matrix from 3-point correspondences. We will derive this algorithm in detail and give also a 4-point algorithm and a linear 5-point algorithm.

We will start with the definition of a simplified essential matrix where two rotations (pitch and roll) are zero. The essential matrix E can be written as a product of the translation and the three rotation matrices as follows:

$$E = [t]_\times (R_Y R_P R_R) \tag{1}$$

R_Y is the rotation matrix for the yaw axis, R_P is the rotation matrix for the pitch axis and R_R is the rotation matrix for the roll axis. $[t]_\times$ is the skew symmetric matrix form of the translation vector $t = (t_x, t_y, t_z)$.

$$R_Y = \begin{bmatrix} \cos(Y) & \sin(Y) & 0 \\ -\sin(Y) & \cos(Y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$R_R = \begin{bmatrix} \cos(R) & 0 & \sin(R) \\ 0 & 1 & 0 \\ -\sin(R) & 0 & \cos(R) \end{bmatrix} \tag{3}$$

$$R_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(P) & \sin(P) \\ 0 & -\sin(P) & \cos(P) \end{bmatrix} \tag{4}$$

With roll and pitch values zero the matrices R_P and R_R are reduced to identity matrices and the essential matrix E gets $E = [t]_\times R_Y$. This expanded gives the simplified essential matrix as written in (5).

$$E = \begin{bmatrix} t_z \sin(Y) & -t_z \cos(Y) & t_y \\ t_z \cos(Y) & t_z \sin(Y) & -t_x \\ -t_y \cos(Y) - t_x \sin(Y) & t_x \cos(Y) - t_y \sin(Y) & 0 \end{bmatrix} \tag{5}$$

By looking at the essential matrix in the form of (5) we can identify 3 linear relations.

$$E = \begin{bmatrix} E_{1,1} & E_{1,2} & E_{1,3} \\ E_{2,1} & E_{2,2} & E_{2,3} \\ E_{3,1} & E_{3,2} & E_{3,3} \end{bmatrix} \tag{6}$$

$$E_{3,3} = 0 \tag{7}$$

$$E_{1,2} = -E_{2,1} \tag{8}$$

$$E_{1,1} = E_{2,2} \tag{9}$$

Using these relations the essential matrix E can be expressed with 6 of its matrix entries reducing the degrees-of-freedom from 8 (up to scale) to 5.

$$E = \begin{bmatrix} E_{2,2} & -E_{2,1} & E_{1,3} \\ E_{2,1} & E_{2,2} & E_{2,3} \\ E_{3,1} & E_{3,2} & 0 \end{bmatrix} \tag{10}$$

In the following we will use the representation of (10) to solve for the essential matrix from point correspondences. The epipolar constraint $\mathbf{x}'^T E \mathbf{x} = 0$ can be written as shown in (11) for the 6 entries of E as written in (12).

$$[x'\ xy' - x'y\ yy' + xx'\ y'\ x\ y]E = 0 \tag{11}$$

$$E = [E_{1,3}\ E_{2,1}\ E_{2,2}\ E_{2,3}\ E_{3,1}\ E_{3,2}] \tag{12}$$

Stacking the constraint rows (11) leads to an equation system of the form

$$AE = 0, \tag{13}$$

where each point correspondence contributes one row. The essential matrix has also to fulfill two internal constraints, the $det(E) = 0$ constraint (14) and the trace constraint (15).

$$det(E) = 0 \tag{14}$$

$$EE^T E - \frac{1}{2} trace(EE^T)E = 0 \tag{15}$$

The condition of pitch and roll being zero can be met by knowing the two rotation angles (e.g. from an IMU or from vanishing points) and transforming the image coordinates from a general pose into one with zero pitch and zero roll angles. This can be done by multiplying the image coordinates of the first frame with a homography transform $H = R_P R_R$ which is seen by writing the three rotations explicitly in the epipolar constraint (16). In (17) it can easily be seen that to remove relative pitch and roll rotations, the necessary transformation is $R_P R_R$.

$$\mathbf{x}'([\boldsymbol{t}]_\times R_Y R_P R_R)\mathbf{x} = 0 \tag{16}$$

$$\mathbf{x}'([\boldsymbol{t}]_\times R_Y)(R_P R_R \mathbf{x}) = 0 \tag{17}$$

3.1 The Linear 5-Point Algorithm

The linear 5-point algorithm is a direct consequence of the epipolar constraints written as (13). The essential matrix E (12) has only 6 parameters and is defined up to scale. Every point correspondence gives a constraint in the form of (11). With 5 of these constraints (13) can be linearly solved for the entries of E. The solution has to be corrected so that E fulfills the $det(E) = 0$ constraint and the trace constraint. This is done be replacing E with E' such that the first two singular values are corrected to be identical. This is in analogy to the 8-point algorithm for computing the essential matrix and fundamental matrix [6]. Similar to the 8-point algorithm the linear 5-point algorithm can be used to find a least squared solution to an over-constrained system (13) if more than 5 point correspondences are used.

Differently to the 8-point algorithm, the case of points in a plane is not a degenerate case for the linear 5-point algorithm. It is shown in [13] that A has rank 6 if all 3D points are in a plane. As the linear 5-point algorithm only needs 5 linearly independent equations this is not a degenerate case.

3.2 The 4-Point Algorithm

The 4-point algorithm is an analogy to the 7-point algorithm for the unconstrained case of computing the essential matrix as described in [4]. It uses the $\det(E) = 0$ constraint for the estimation.

With 4 point correspondences A of the equation system (13) has rank 4. In this case the solution to (13) is a 2-dimensional null space of the form

$$E = aE_1 + E_2. \tag{18}$$

The two-dimensional null space E_1 and E_2 can be computed using SVD. The scalar parameter a can be computed using the $\det(E) = 0$ constraint. Expanding

the expression $\det(aE_1 + E_2) = 0$ leads to a cubic polynomial in a. There will be one or three solutions for a (complex solutions are discarded) which leads to one or three solutions for E by back-substitution into (18). The algorithm can be used for more than 4 point correspondences. In this case the null space E_1, E_2 is computed in the least squares sense from all the point correspondences.

3.3 The 3-Point Minimal Case Algorithm

The 3-point minimal case algorithm is an analogy to the 5-point minimal case solver [10] and as well to the 6-point minimal case solver [14] for the unconstrained case. It is minimal as it solves for the remaining 3 DOF (up to scale) using only 3 point correspondences. Similar to the 5-point and 6-point methods it uses the trace constraint and the $\det(E) = 0$ constraint to set up a polynomial equation system and to find the solution to E by solving it.

With 3 point correspondences the matrix A of (13) has rank 3. In this case the solution to (13) is a 3-dimensional null space of the form

$$E = aE_1 + bE_2 + E_3 \tag{19}$$

This is the same case as for the 6-point algorithm, where the solution also has the form of a 3-dimensional subspace.

To derive the solution we start by substituting (19) into the $\det(E) = 0$ constraint (14) and into the trace constraint (15), which gives 7 polynomial equations in a and b of degree 3. Our reduced essential matrix parameterization (12) has only 6 parameters, thus the trace constraint gives only 6 equations instead of 9. And the seventh equation comes from the $\det(E) = 0$ constraint. However the rank of the linear system of the 7 equation is only 6. There exists a linear dependency between the entries $E_{1,3}, E_{2,3}, E_{3,1}, E_{3,2}$ which can be verified by symbolic Gaussian elimination on the equations.

Considering this as a homogeneous linear system in the monomials of a and b this will give expressions in the monomials $a^3, a^2b, a^2, ab^2, ab, a, b^3, b^2, b, 1$. Next we set up a polynomial equation system using the 6 linearly independent equations. By performing Gaussian elimination and subsequent row operations a polynomial of 4th degree in b can be found. From Gaussian elimination we get (20).

$$\begin{array}{cccccccccc|l}
a^3 & a^2b & a^2 & ab^2 & ab & a & b^3 & b^2 & b & 1 & \\
\hline
1 & & & & & & . & . & . & . & \\
 & 1 & & & & & . & . & . & . & \\
 & & 1 & & & & . & . & . & . & \\
 & & & 1 & & & . & . & . & . & \\
 & & & & 1 & & . & . & . & . & < h > \\
 & & & & & 1 & . & . & . & . & < i >
\end{array} \tag{20}$$

With row operations on the polynomials $< h >$ and $< i >$ we can arrive at the desired 4th degree univariate polynomial.

$$< h >= poly(ab, b^3, b^2, b, 1) \tag{21}$$

$$< i >= poly(a, b^3, b^2, b, 1) \tag{22}$$

We eliminate the monomial ab by multiplying $< i >$ with b and subtracting it from $< h >$.

$$< k >=< h > -b < i >= poly(b^4, b^3, b^2, b, 1) \tag{23}$$

The resulting polynomial $< k >$ is a 4th degree polynomial in b. The solutions for b by solving $< k >$ (23) can be substituted into $< i >$ (22) which is linear in a and which gives one solution for a for each b. For each pair of (a, b) we compute an essential matrix using (19).

It can be shown (e.g. with symbolic software like the Gröbner basis package of Maple) that the 3-point problem can have up to 4 solutions. In this sense our solution with the 4th degree polynomial is optimal. The algorithm can also be used for more than 3 point correspondences. In this case the null space E_1, E_2, E_3 is computed in the least squares sense from all the point correspondences.

4 Degeneracies

In [13] it is shown that 6 points in a plane give 6 linearly independent equations of the form of (11), however any further point in the plane will not give another linearly independent equation. This means that algorithms that need more than 6 independent linear equations, e.g. the 7-point and 8-point algorithm will fail for the case of a plane. The 5-point and the 6-point algorithm and the 3-point, 4-point and linear 5-point as well are able to deal with the case of all the points in a plane.

The case of collinear points is another interesting case. It is a degenerate case for the 5-point algorithm but not for the 3-point, which can compute the essential matrix from 3 collinear points. Three points on the same line give three independent equations for the epipolar constraint (11), however any further points on the line will not give another linearly independent equations [13]. For the 3-point algorithm only three linearly independent equations in (11) are needed. The equation system (20) has rank 6 in this case which allows our method to solve the system. However the case where the 3D line is parallel to the baseline of the cameras is a degenerate case. There we observed that the rank of (20) drops to 3.

5 Experiments

5.1 Synthetic Data

In this experiment we evaluate the robustness of the new methods under image noise and noise on the IMU measurements and compare it to the standard 5-point method [11]. Robustness to noise from the IMU is especially important since errors in the two given angles can influence the outcome of the new algorithms.

The test scene consists of random 3D points that have a depth of 50% of the distance of the first camera to the scene. The baseline between the views is 10% of the distance to the scene and the direction is either parallel to the

scene (sideways) or along the z-axis of the first camera (forward). Additionally, the second camera was rotated around every axis. This is similar to Nistér's test scene described in [11].

We want to test the algorithms used for two cases, the minimal case when they are used for RANSAC and the least squares case when using many points for a polishing step. For the first series of experiments the minimal number of point correspondences necessary to get a solution is used. If the method resulted in multiple solutions the one with the smallest deviation of the true solution was chosen. For the minimal case experiments 500 trials were done per data point in the plots and the average translation and rotation error of the first quantile are plotted. This measure is a good performance criterion if the method is used for RANSAC where it is more important to find an acceptable solution with many inliers than to get consistent results over all trials. The less accurate solutions will result in less inliers and be automatically filtered out. The least squares plots show the mean value of 200 trials with 100 point correspondences per trial.

To analyze the robustness the computed essential matrices are decomposed in a relative translation direction and rotation. The translational error is the

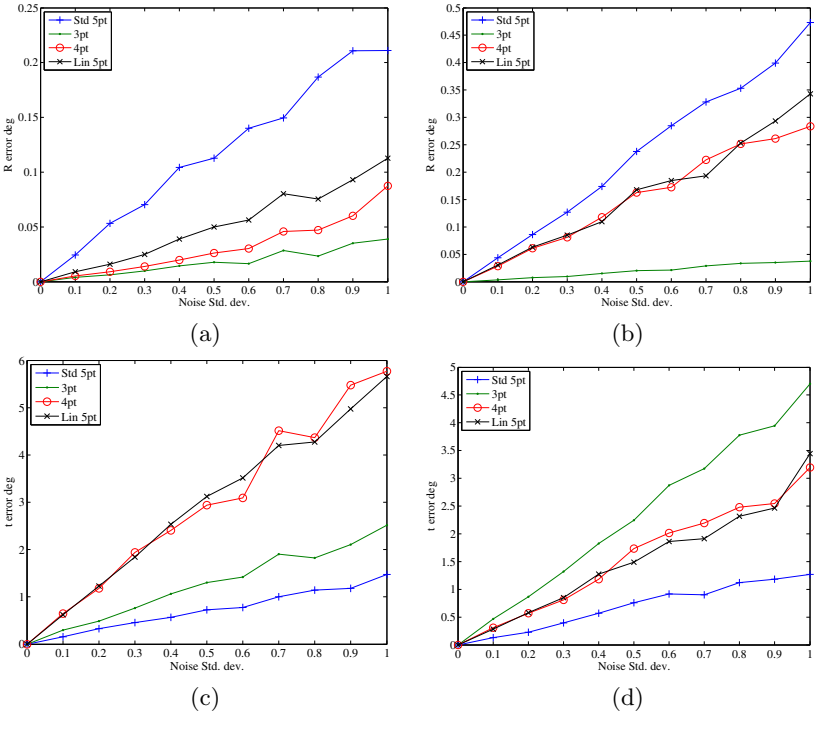


Fig. 1. Rotational and translational error for the minimal case algorithms in degrees over image noise standard deviation in pixel. (a) Rotational sideways (b) Rotational forward (c) Translational sideways (d) Translational forward.

angle between the true translation direction and the estimated direction. The rotational error is the smallest angle of rotation which is needed to bring the estimated rotation to the true value.

Figure 1 shows the results of the minimal cases for gradually increased image noise levels with perfect IMU data. As can be seen, the new methods produce slightly less robust translation estimates. One reason for that is that with only three resp. four image points the 3-point and 4-point methods are generally more sensitive to image noise. Notice that the 3-point method performs better for sideways motion than for forward motion. Regarding the rotational errors the new methods are more robust than the standard 5-point method, but it has to be noticed that the errors of all algorithms are very small. In addition to that the standard 5-point method has to estimate rotations around 3 axes whereas the new methods estimate only one.

The least squares results in Fig. 2 show that the new methods are as robust as the standard 5-point method for sideways motion. For forward motion, which is the weak point of the standard 5-point algorithm, the 3-point and linear 5-point methods perform better, but the 4-point method does not produce better results.

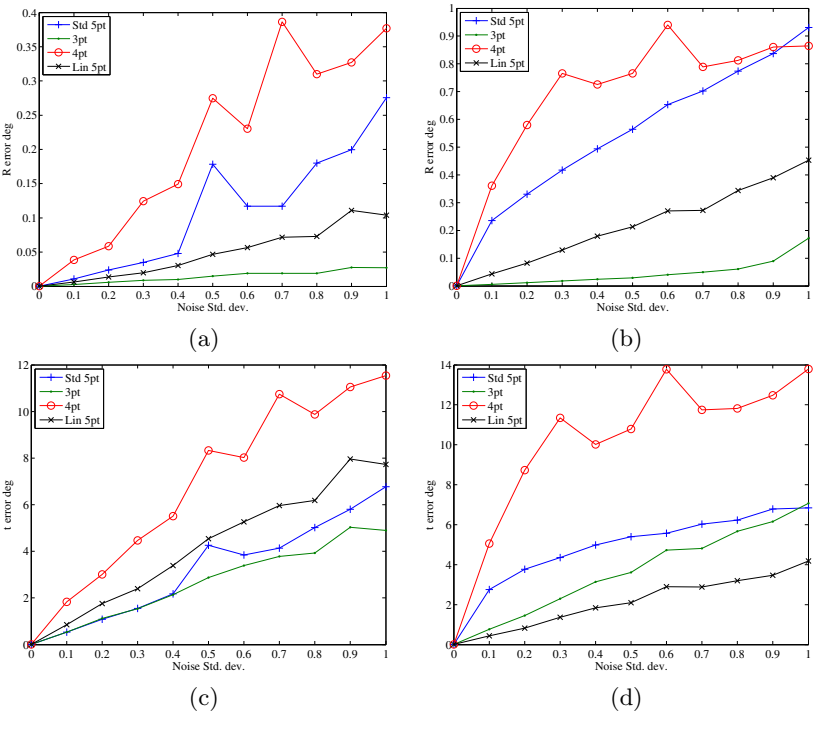


Fig. 2. Rotational and translational error (least squares case) in degrees over image noise standard deviation in pixel. (a) Rotational sideways (b) Rotational forward (c) Translational sideways (d) Translational forward.

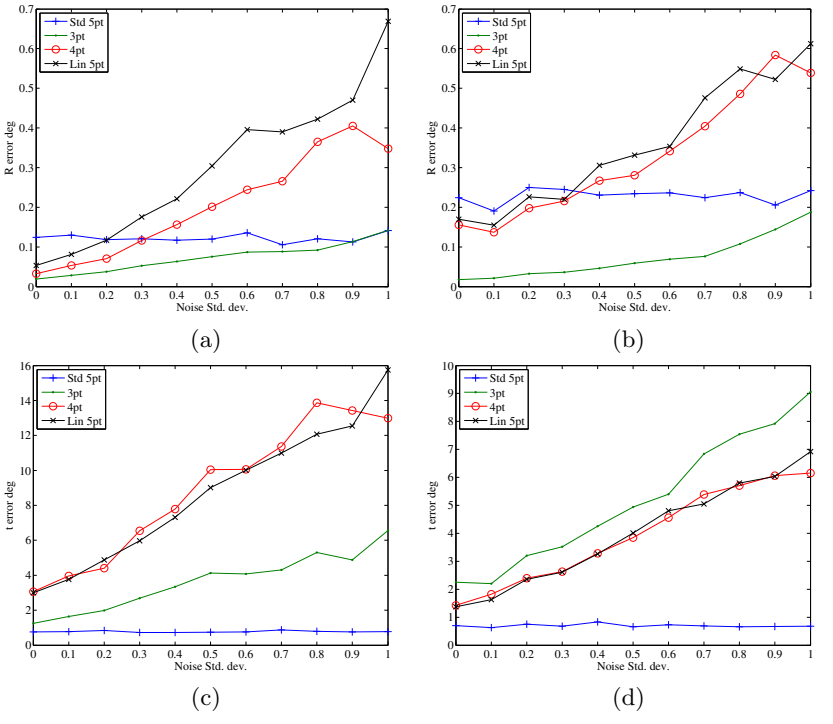


Fig. 3. Rotational and translational error for the minimal case algorithms in degrees over IMU noise standard deviation in degree (at 0.5 pixel image noise). (a) Rotational sideways (b) Rotational forward (c) Translational sideways (d) Translational forward.

Since in reality no exact data will be available about the attitude of the camera, it is important to examine how the new methods work when noise is added to the IMU data. Good accelerometers today have noise levels of around 0.02 degrees in the computed angles. Figure 3 shows the results for the minimal cases for increasing noise on the IMU data while assuming image noise with 0.5 pixel standard deviation. The standard 5-point method is not influenced by the noise because no IMU data is used. But for the other algorithms the error scales with the noise of the IMU data. This shows that it is important to have good measurements from the IMU because too big errors in the angles make it harder to compute the correct solution. The 3-point method gives acceptable results for sideways motion even for very noisy IMU data. The least squares results in Fig. 4 show a similar picture. For sideways motion and moderate noise the 3-point and the linear 5-point methods are as robust as the standard 5-point method, where for forward motion the linear 5-point is again the most robust of the presented methods. With more image points used for the estimation the new methods benefit from more robustness to image nose, but it is apparent that more image points do not help that much against noise and imprecisions from the IMU.

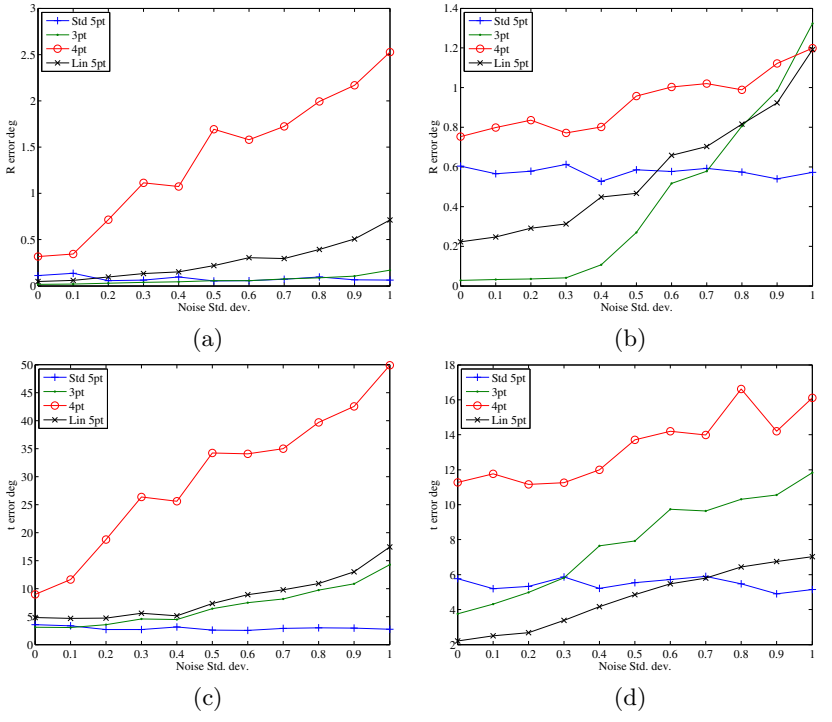
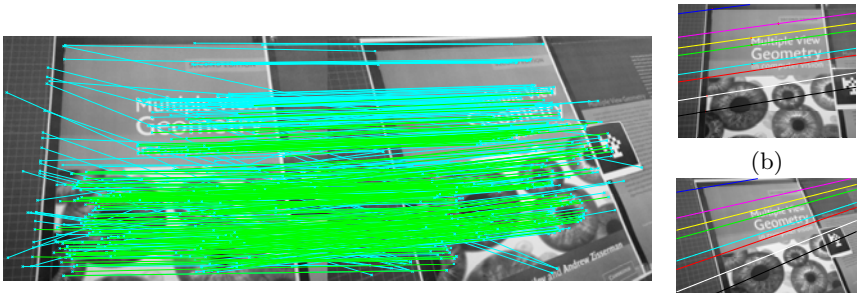


Fig. 4. Rotational and translational error (least squares case) in degrees over IMU noise standard deviation in degree (at 0.5 pixel image noise). (a) Rotational sideways (b) Rotational forward (c) Translational sideways (d) Translational forward.

The results of the synthetic experiments show that the new methods and especially the 3-point method can be used for example for faster RANSAC to get a set of consistent point correspondences with similar robustness as the standard 5-point method.

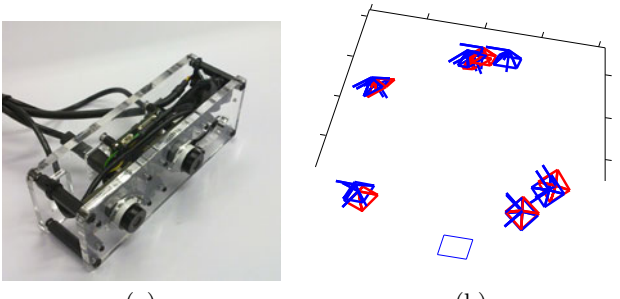
5.2 Real Data from N900 Smartphone

To demonstrate that the new 3-point method also works on currently available consumer Smartphones. We tested it with the Nokia N900 that has an accelerometer similar to other modern Smartphones like Apple's iPhone. The relative rotations to the plane perpendicular to the gravity vector were read out from the accelerometers and used to warp the extracted SIFT [7] features to a virtual ground plane. These transformed feature coordinates only differ in one rotation angle and translation and where used as input for the 3-point method. Fig. 5 shows an example of estimated epipolar geometry for a planar scene. The blue lines show the initial SIFT matches and the green lines the inliers after RANSAC.



(a) (b) (c)

Fig. 5. Epipolar geometry from N900 images and IMU readings. (a) Cyan lines indicate initial SIFT feature matches, green lines are inliers after RANSAC. Residual Sampson distance of inliers is 0.4 pixel. The quality of inliers depends largely on the accuracy of IMU readings. (b,c) Epipolar geometry visualized by epipolar lines.



(a) (b)

Fig. 6. Relative pose estimation for MAV camera-IMU system. (a) The camera-IMU system for a micro quadrotor. (b) Poses computed from the 3-point algorithm (red) compared to marker based pose estimation (blue).

5.3 Relative Pose Estimation for MAV Camera-IMU System

In this experiment we test the 3-point algorithm for relative pose estimation for a MAV camera-IMU system. The camera-IMU system consists of a stereo setup and an IMU with a nominal accuracy of 0.06 degrees. The camera-IMU system is designed to fit on a micro quadrotor. In the experiment we compute the pose of the camera-IMU relative to a reference pose for multiple different poses using SIFT [7] feature matches. To verify the results we also compute the poses using a marker based pose estimation method[1] without using IMU information. The computed poses are plotted in Fig. 6. The poses from the 3-point method are in red color, the poses from the marker based method are in blue. The blue square is the location of the marker on the floor. This experiment successfully demonstrates the practicability of our proposed 3-point method.

[1] AR Toolkit Plus: http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php

6 Conclusion

In this paper we presented three new algorithms (3-point, 4-point and linear 5-point) to compute the essential matrix for the case when two orientation angles are known. We showed that this is a practically relevant case as modern Smartphones (iPhone, Nokia N900) are equipped with an IMU that can measure two of the three orientation angles of the device. This is done by accelerometers which are able to measure the earths gravity normal and thus can measure two of its three orientation angles. This case is also practically relevant for the case of pure visual relative pose estimation for the case when a vanishing point can be detected in the images. Similar to an IMU a single vanishing point also gives two orientation angles. What's more, it is conceivable that in future, cameras will always be coupled with IMU's. The small scale of MEMS IMU's make a tight integration possible. This makes the proposed method even more practically relevant. In the experimental section we analyzed the accuracy of the proposed methods very carefully using synthetic data. A special focus was put on the case of noisy IMU measurements. We showed that the algorithm is sensitive to inaccurate IMU measurements, especially the translation part. However, we showed with real experiments using a Nokia N900 Smartphone, that the accuracy of 1° is sufficient for robust relative pose estimation. More experiments conducted with an IMU with higher precision (0.06°) showed how the proposed methods can reliably be used for relative pose estimation of an micro aerial vehicle. So far we did not explore the possibility of using the two orientation angles to unwarp the images before feature detection and matching. By unwarping, the perspective distortions between the two images could be removed. This would allow the use of simpler feature detectors, which need to be only scale and rotation invariant.

References

1. Fischler, M.A., Bolles, R.C.: RANSAC random sampling concensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of ACM 26, 381–395 (1981)
2. Gallagher, A.: Using vanishing points to correct camera rotation in images. In: Computer and Robot Vision, pp. 460–467 (2005)
3. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge (2000)
4. Hartley, R.: Projective reconstruction and invariants from multiple images. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(10), 1036–1041 (1994)
5. Kalantari, M., Hashemi, A., Jung, F., Guédon, J.P.: A new solution to the relative orientation problem using only 3 points and the vertical direction. arXiv 0905.3964 (2009)
6. Longuet-Higgins, H.: A computer algorithm for reconstructing a scene from two projections. Nature 293, 133–135 (1981)
7. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)

8. Matas, J., Chum, O.: Randomized ransac with sequential probability ratio test. In: Proc. IEEE International Conference on Computer Vision, vol. II, pp. 1727–1732 (2005)
9. Naroditsky, O., Zhu, Z., Das, A., Samarasekera, S., Oskiper, T., Kumar, R.: Videotrek: A vision system for a tag-along robot. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1101–1108 (2009)
10. Nistér, D.: An efficient solution to the five-point relative pose problem. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. II195–II202 (2003)
11. Nistér, D., Schaffalitzky, F.: Four points in two or three calibrated views: theory and practice. International Journal of Computer Vision, 211–231 (2006)
12. Oskiper, T., Zhu, Z., Samarasekera, S., Kumar, R.: Visual odometry system using multiple stereo cameras and inertial measurement unit. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
13. Philip, J.: Critical point configurations of the 5-,6-,7-, and 8-point algorithms for relative orientation. In: Technical Report TRITA-MAT-1998-MA-13, KTH (1998)
14. Pizarro, O., Eustice, R., Singh, H.: Relative pose estimation for instrumented, calibrated imaging platforms. In: Proceedings of the Seventh International Conference on Digital Image Computing: Techniques and Applications, pp. 601–612 (2003)
15. Pollefeys, M., Nister, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. International Journal of Computer Vision 78(2-3), 143–167 (2008)
16. Robertsone, D., Cipolla, R.: An image-based system for urban navigation. In: British Machine Vision Conference, pp. 1–10 (2004)
17. Steder, B., Grisetti, G., Grzonka, S., Stachniss, C., Rottmann, A., Burgard, W.: Learning maps in 3d using attitude and noisy vision sensors. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 644–649 (2007)

Bilinear Factorization via Augmented Lagrange Multipliers*

Alessio Del Bue[1], João Xavier[2], Lourdes Agapito[3], and Marco Paladini[3]

[1] Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy
[2] ISR - Istituto Superior Técnico, Av. Rovisco Pais, Lisbon, Portugal
[3] Queen Mary University of London, Mile end road, E1 4NS London, UK

Abstract. This paper presents a unified approach to solve different bilinear factorization problems in Computer Vision in the presence of missing data in the measurements. The problem is formulated as a constrained optimization problem where one of the factors is constrained to lie on a specific manifold. To achieve this, we introduce an equivalent reformulation of the bilinear factorization problem. This reformulation decouples the core bilinear aspect from the manifold specificity. We then tackle the resulting constrained optimization problem with Bilinear factorization via Augmented Lagrange Multipliers (BALM). The mechanics of our algorithm are such that only a projector onto the manifold constraint is needed. That is the strength and the novelty of our approach: it can handle seamlessly different Computer Vision problems. We present experiments and results for two popular factorization problems: Non-rigid Structure from Motion and Photometric Stereo.

1 Introduction

Many inference problems in Computer Vision fit the form of bilinear problems since often observations are influenced by two independent factors where each can be described by a linear model. For instance, in photometric stereo [2] the shape of the object and the light source direction interact bilinearly to influence the image intensity. In rigid structure from motion [16] the 3D shape of the object is pre-multiplied by the camera matrix to determine its image coordinates. In facial tracking the problem of separating head pose and facial expression can also be defined as a bilinear problem [1]. In non-rigid structure from motion [4] the 2D coordinates of features arise from a bilinear relation between the camera matrix and the time varying shape. The interaction between two factors has also been generalised to several problems [15] in a learning context. In all these

* This research has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013) ERC Starting Grant agreement 204871-HUMANIS. This work was also partially supported by FCT, under ISR/IST plurianual funding (POSC program, FEDER), grant SIPM-PTDC/EEA-ACR/73749/2006, grant MODI-PTDC/EEA-ACR/72201/2006. We thank J. Buenaposada for providing an implementation of [2].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 283–296, 2010.

problems the objective is to make inferences about both factors – the goal is their simultaneous estimation.

In this paper, we present a unified approach to solve different bilinear factorization problems in computer vision. The problem is formulated as a constrained optimization problem where one of the factors is constrained to lie on a specific manifold. Our key observation is that the main difference between different factorization problems is the manifold on which the solution lies. Thus, intuitively, it should be possible to construct an unified optimization framework in which a change of the manifold constraint just implies replacing an inner module of the algorithm (as opposed to an overall redesign of the optimization method from scratch). In this paper, we propose such a modular approach. To achieve this, we start by introducing an equivalent reformulation of the bilinear factorization problem. In loose terms, the reformulation decouples the core bilinear aspect from the manifold specificity. We then tackle the resulting constrained optimization problem via an Augmented Lagrange Multipliers (ALM) iterative algorithm. The mechanics of our algorithm are such that only a projector onto the manifold constraint is needed. That is the strength and the novelty of our approach: this framework can handle seamlessly different computer vision factorization problems. What will differ in each case is the projector of the solution onto the correct manifold.

In our experiments we show that we are able to deal with high percentages of missing data which has the practical implication that our approach can be used on data coming from real and not just controlled scenarios. We illustrate our unified approach by applying it to solve two popular computer vision problems: Non-rigid Structure-from-Motion (NRSfM) and Photometric Stereo (PS). To the best of our knowledge, this paper constitutes the first attempt to propose a unified optimisation framework for large scale bilinear factorization problems with given manifold constraints on one of the factors and provide a practical algorithm that can deal with missing data in the measurements.

2 Related Work

Bilinear models appear frequently in Computer Vision. However, it is in the area of Structure from Motion (SfM) that most of the efforts dedicated to solve this problem have come from. The wealth of research in this area is such that we cannot give an exhaustive review of the literature. Instead we will focus on describing what we believe are the two most important threads of research to solve the problem of low-rank matrix factorization in the case of missing data.

One line of research that dominates the literature are approaches that perform alternation of closed-form solutions to solve for the two factors of the matrix. The first of these approaches to solve the problem of missing data was proposed by Wiberg [18]. Since then many different solutions have been put forward. Buchanan and Fitzgibbon [5] provide a comprehensive review of these methods while proposing their own alternative approach. Their Damped Newton algorithm provides faster and more accurate solutions than standard alternation approaches. The common property of all these methods is that they only solve the

low-rank matrix factorization problem without imposing manifold constraints. The constraints are applied afterwards, once the low-rank matrix has been estimated. Crucially, the constraints are not imposed during the minimization.

On the other hand, a relatively recent set of algorithms have attempted to solve the problem by including explicitly the non-linear constraints given by the specific problem structure in the low-rank minimization. Marques and Costeira [12] introduced the concept of *motion manifold* in rigid SfM to obtain motion matrices that exactly satisfy the camera constraints. Similarly, Paladini et al. [13] propose an alternation algorithm associated with an optimal projector onto the *motion manifold* of non-rigid shapes. The practical implication of their algorithm is that it can deal with very high percentages of missing data. Shaji *et al.* [14] also propose to solve a non-linear optimisation problem directly on the product manifold of the Special Euclidean Group claiming better results than [5] in a rigid real sequence.

However, all these approaches are tailored to specific problems. Therefore, for different manifold constraints an overall redesign of the optimization method would be needed. The purpose of our work is to present a generic approach that is not problem dependent. In similar spirit, Chandraker and Kriegman [6] have proposed a globally optimal bilinear fitting approach for general Computer Vision problems. The key contribution of their approach is that they can prove convergence to a global minimiser using a branch and bound approach. However, the main drawback is that they are restricted to simple bilinear problems where the number of variables in one of the sets must be very small (for instance just 9 variables in one of their examples).

Our Bilinear factorization via Augmented Lagrange Multipliers (BALM) is designed to deal with large-scale optimisation problems with the inclusion of non-linear constraints. Our approach is not the first one to adopt the Augmented Lagrangian Multipliers (ALM) framework in the Computer Vision or related contexts. In perspective 3D reconstruction [11] ALM was used to enforce constraints on the perspective depths. In [10] ALM is successfully employed as a single matrix imputation algorithm which can deal with large scale problems.

3 Problem Statement

We denote by $Y \in \mathbb{R}^{n\times m}$ the measurement matrix. In this paper, we consider the general case of missing data. We let the finite set $\mathcal{O} := \{(i,j) : Y_{ij} \text{ is observed}\}$ enumerate the indices of the entries of Y which are available. The bilinear factorization problem we address is the following constrained optimization problem:

$$\begin{array}{l} \text{minimize } \sum_{(i,j)\in\mathcal{O}} \left(Y_{ij} - s_i^\top m_j\right)^2 \\ \text{subject to } M_i \in \mathcal{M}, \quad i = 1,\ldots,f, \end{array} \tag{1}$$

where $s_i^\top$ denotes the ith row of the matrix $S \in \mathbb{R}^{n\times r}$ and m_j denotes the jth column of the matrix $M = \begin{bmatrix} M_1 & \cdots & M_i & \cdots & M_f \end{bmatrix} \in \mathbb{R}^{r\times m}, \quad M_i \in \mathbb{R}^{r\times p}$.

The variable to optimize in (1) is (S, M). Here, f is the number of frames and we consider throughout the paper that $n \geq r \geq p$. For instance in the structure

from motion problem S would be the 3D structure and M the camera matrices, in photometric stereo S would be the lighting parameters and M the surface normals and albedo.

In words, problem (1) consists in finding the best rank r factorization SM of Y, given the available entries enumerated by $\mathcal{O}$ and subject to the constraints on M. More precisely, each submatrix $M_i \in \mathbb{R}^{r\times p}$ of M must belong to the manifold $\mathcal{M} \subset \mathbb{R}^{r\times p}$. Our aim in this paper is to construct an algorithm to solve problem (1) which takes advantage of the fact that the projector onto $\mathcal{M}$ is available (easily implementable). That is, we assume that, for a given $A \in \mathbb{R}^{r\times p}$, it is known how to solve the projection problem onto $\mathcal{M}$

$$\begin{array}{l} \text{minimize} \quad \|A - X\|^2, \\ \text{subject to } X \in \mathcal{M} \end{array} \tag{2}$$

where $\|X\|$ denotes the Frobenius norm of X. In the sequel, we let $p_{\mathcal{M}}(A)$ denote a solution of (2).

Problem reformulation. Let us define a new set of variables $z := \{Z_{ij} : (i,j) \notin \mathcal{O}\}$. Think of them as representing the non-observed entries of Y. We can introduce these variables in (1) and obtain the following equivalent optimization problem

$$\begin{array}{l} \text{minimize} \quad \|Y(z) - SM\|^2 \\ \text{subject to } M_i \in \mathcal{M}, \quad i = 1, \ldots, f, \end{array} \tag{3}$$

where the (i,j) entry of the matrix $Y(z)$ is defined as

$$(Y(z))_{ij} := \begin{cases} Y_{ij} \ , & \text{if } (i,j) \in \mathcal{O} \\ Z_{ij} \ , & \text{if } (i,j) \notin \mathcal{O} \end{cases}.$$

In words, $Y(z)$ is just Y where we fill-in the missing entries with z. Note that the variable to optimize in (3) is (z, S, M). Problem (3) is equivalent to (1) because once we fix (S, M) in (3) and minimize over z we fall back into (1). Finally, we clone M into a new variable $N = \begin{bmatrix} N_1 & \cdots & N_i & \cdots & N_f \end{bmatrix} \in \mathbb{R}^{r\times m}, \quad N_i \in \mathbb{R}^{r\times p}$, and transfer the manifold constraint to the latter. By doing so, we roughly separate the bilinear issue from the manifold restriction. This is our final reformulation:

$$\begin{array}{l} \text{minimize} \quad \|Y(z) - SM\|^2 \\ \text{subject to } M_i = N_i, \quad i = 1, \ldots, f \\ \qquad\qquad\quad N_i \in \mathcal{M}, \quad i = 1, \ldots, f. \end{array} \tag{4}$$

The variable to optimize in (4) is (z, S, M, N).

4 The BALM Algorithm

The main difficulty in the constrained optimization problem (4) are the equality constrains $M_i = N_i$. We propose to handle them through an augmented Lagrangian approach, see [9,3] for details on this optimization technique. In our

context, the augmented Lagrangian corresponding to (4) is given by

$$L_\sigma(z,S,M,N;R) = \|Y(z) - SM\|^2 - \sum_{i=1}^{f} \mathsf{tr}\left(R_i^\top (M_i - N_i)\right) + \frac{\sigma}{2}\sum_{i=1}^{f}\|M_i - N_i\|^2 .$$

where $\sigma > 0$ is the weight of the penalty term and R_i, $i = 1,\ldots,f$, denote Lagrange multipliers. We let $R = \begin{bmatrix} R_1 \cdots R_f \end{bmatrix}$. The optimization problem (4) can then be tackled by our Bilinear factorization via Augmented Lagrange Multipliers (BALM) algorithm detailed in Algorithm 1.

Algorithm 1. Bilinear factorization via Augmented Lagrange Multipliers (BALM)

set $k = 0$ and $\epsilon_{\mathsf{best}} = +\infty$
initialize $\sigma^{(0)}$, $R^{(0)}$, $\gamma > 1$ and $0 < \eta < 1$
initialize $z^{(0)}$, $S^{(0)}$ and $M^{(0)}$
repeat
    solve

$$\left(z^{(k+1)}, S^{(k+1)}, M^{(k+1)}, N^{(k+1)}\right) = \begin{array}{ll} \text{argmin} & L_{\sigma^{(k)}}(z,S,M,N;R^{(k)}) \\ \text{subject to} & N_i \in \mathcal{M}, \quad i = 1,\ldots,f, \end{array} \tag{5}$$

    using the iterative Gauss-Seidel scheme described in **Algorithm 2**
    compute $\epsilon = \left\|M^{(k+1)} - N^{(k+1)}\right\|^2$
    if $\epsilon < \eta\,\epsilon_{\mathsf{best}}$
        $R^{(k+1)} = R^{(k)} - \sigma^{(k)}\left(M^{(k+1)} - N^{(k+1)}\right)$
        $\sigma^{(k+1)} = \sigma^{(k)}$
        $\epsilon_{\mathsf{best}} = \epsilon$
    else
        $R^{(k+1)} = R^{(k)}$
        $\sigma^{(k+1)} = \gamma\sigma^{(k)}$
    endif
    update $k \leftarrow k + 1$
until some stopping criterion

Regarding the initialization of the BALM algorithm, we used $\sigma^{(0)} = 50$, $R^{(0)} = 0$, $\gamma = 5$ and $\eta = 1/2$ in all our computer experiments. With respect to $z^{(0)}$, $S^{(0)}$ and $M^{(0)}$, we feel that there is no universally good method, that is, the structure of $\mathcal{M}$ must taken into account. We discuss the initialization $\left(z^{(0)}, S^{(0)}, M^{(0)}\right)$ for several examples in the experimental section of this paper.

Clearly, solving the inner problem (5) at each iteration of the BALM method is the main computational step. Note that in (5) the optimization variable is (z,S,M,N) ($\sigma^{(k)}$ and $R^{(k)}$ are constants). To tackle (5) we propose an iterative Gauss-Seidel scheme which is described in Algorithm 2. We now show that each of the subproblems (6), (7) and (8) inside the Gauss-Seidel scheme are easily solvable.

Algorithm 2. Iterative Gauss Seidel scheme to solve for (5)

1: set $l = 0$ and choose $L_{\max}$
2: set $z^{[0]} = z^{(k)}$, $S^{[0]} = S^{(k)}$ and $M^{[0]} = M^{(k)}$
3: **repeat**
4: solve

$$N^{[l+1]} = \operatorname{argmin} \quad L_{\sigma^{(k)}}\left(z^{[l]}, S^{[l]}, M^{[l]}, N; R^{(k)}\right) \quad \text{subject to } N_i \in \mathcal{M}, \quad i = 1, \ldots, f, \tag{6}$$

5: solve

$$\left(S^{[l+1]}, M^{[l+1]}\right) = \operatorname{argmin} L_{\sigma^{(k)}}\left(z^{[l]}, S, M, N^{[l+1]}; R^{(k)}\right) \tag{7}$$

6: solve

$$z^{[l+1]} = \operatorname{argmin} L_{\sigma^{(k)}}\left(z, S^{[l+1]}, M^{[l+1]}, N^{[l+1]}; R^{(k)}\right) \tag{8}$$

7: update $l \leftarrow l + 1$
8: **until** $l = L_{\max}$
9: set $S^{(k+1)} = S^{[L_{\max}]}$, $M^{(k+1)} = M^{[L_{\max}]}$ and $N^{(k+1)} = N^{[L_{\max}]}$

4.1 Solving for (6)

It is straightforward to see (details omitted) that (6) decouples into f projections onto the manifold of constraints $\mathcal{M}$. More precisely, if we partition

$$N^{[l+1]} = \left[N_1^{[l+1]} \cdots N_i^{[l+1]} \cdots N_f^{[l+1]}\right] \in \mathbb{R}^{r \times m}, \quad N_i^{[l+1]} \in \mathbb{R}^{r \times p},$$

the solution of (6) is given by

$$N_i^{[l+1]} = p_{\mathcal{M}}\left(M_i^{[l]} - \frac{1}{\sigma^{(k)}} R_i^{(k)}\right), \quad i = 1, \ldots, f.$$

We recall that $p_{\mathcal{M}}$ stands for the projector onto $\mathcal{M}$, see (2), which we assume is available. This is the only part of the algorithm where the constraint manifold $\mathcal{M}$ plays a role. Thus, replacing $\mathcal{M}$ amounts to replace the projector $p_{\mathcal{M}}$. This is the modularity which we alluded to previously.

4.2 Solving for (7)

To simplify notation in this subsection, we let $Y = Y\left(z^{[l]}\right)$, $N = N^{[l+1]}$, $\sigma = \sigma^{(k)}$ and $R = R^{(k)}$. Solving (7) corresponds to solving

$$\text{minimize } \|Y - SM\|^2 + \tfrac{\sigma}{2} \textstyle\sum_{i=1}^{f} \left\|M_i - \left(N_i + \frac{1}{\sigma} R_i\right)\right\|^2.$$

Equivalently, in terms of the new variable $\widetilde{S} = \sqrt{\frac{2}{\sigma}}S$, we have

$$\text{minimize } \left\|\widetilde{Y} - \widetilde{S}M\right\|^2 + \|C - M\|^2 \tag{9}$$

where $\widetilde{Y} = \sqrt{\frac{2}{\sigma}}Y$ and $C = N + \frac{1}{\sigma}R$.

Now, any full row rank matrix $M \in \mathbb{R}^{r\times m}$ can be represented as $M = AQ^\top$ where $A \in \mathbb{R}^{r\times r}$ is nonsingular and $Q \in \mathbb{R}^{m\times r}$ is a Stielfel matrix ($Q^\top Q = I_r$). Plugging this representation into (9) produces the optimization problem

$$\begin{aligned} &\text{minimize } \left\|\widetilde{Y} - \widetilde{S}AQ^\top\right\|^2 + \left\|C - AQ^\top\right\|^2 \\ &\text{subject to } Q^\top Q = I_r \end{aligned} \tag{10}$$

with optimization variable $(\widetilde{S}, A, Q) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{r\times r} \times \mathbb{R}^{m\times r}$.

Introducing the new variable $\widehat{S} = \widetilde{S}A$, transforms (10) into

$$\begin{aligned} &\text{minimize } \left\|\begin{bmatrix}\widetilde{Y}\\ C\end{bmatrix} - \begin{bmatrix}\widehat{S}\\ A\end{bmatrix} Q^\top\right\|^2. \\ &\text{subject to } Q^\top Q = I_r \end{aligned} \tag{11}$$

For a given Q in (11), the optimal $(\widehat{S}, A)$ is

$$\begin{bmatrix}\widehat{S}\\ A\end{bmatrix} = \begin{bmatrix}\widetilde{Y}\\ C\end{bmatrix} Q$$

which, when plugged back into (11), leaves the maximization problem

$$\begin{aligned} &\text{maximize } \mathsf{tr}\left(Q^\top \begin{bmatrix}\widetilde{Y}\\ C\end{bmatrix}^\top \begin{bmatrix}\widetilde{Y}\\ C\end{bmatrix} Q\right) \\ &\text{subject to } Q^\top Q = I_r \end{aligned} \tag{12}$$

which can be solved through an eigenvalue decomposition ($\mathsf{tr}(X)$ denotes the trace of the square matrix X). Problem (7) optimizes jointly over (S,M). An alternative approach is to replace (7) by two least-squares problems: one over M (for fixed S) and the other over S (for fixed M).

4.3 Solving for (8)

After solving for $N^{[l+1]}$ and $\left(S^{[l+1]}, M^{[l+1]}\right)$, problem (8) updates the missing data. The solution of (8) is trivial: we just have to take $Z_{ij}^{[l+1]}$ as the (i,j)th entry of $S^{[l+1]}M^{[l+1]}$ for all $(i,j) \notin \mathcal{O}$.

Algorithm convergence. At best, the BALM algorithm can produce a local minimizer for (1). That is, we do not claim that BALM (algorithm 1) converges to a global minimizer. In fact, even the nonlinear Gauss-Seidel technique (algorithm 2) is not guaranteed to globally solve (5). This is the common situation when dealing with nonconvex problems. See [7] for some convergence results on augmented Lagrangian methods. We now apply our generic BALM algorithm to solve two different bilinear computer vision problems: the Non-Rigid Structure-from-Motion problem (NRSfM) and the Photometric Stereo (PS) problem.

5 Example 1: BALM for Non-rigid SfM

The problem of recovering the non-rigid 3D shape of a deforming object from a monocular video sequence given only 2D correspondences between frames was formulated as a factorization problem by Bregler *et al.* [4] in the case of an orthographic camera. The assumption is that the 3D shape can be represented as a linear combination of a set of d basis shapes with time varying coefficients t_{id}. If the image coordinates are referred to the centroid, the projection of the shape at frame i can be expressed as

$$Y_i = \begin{bmatrix} u_{i1} \dots u_{in} \\ v_{i1} \dots v_{in} \end{bmatrix}^\top = \left(\sum_{l=1}^{d} t_{il} B_l \right) Q_i = \begin{bmatrix} B_1 \dots B_d \end{bmatrix} (t_i \otimes Q_i) = S M_i \quad (13)$$

where Y_i is the $n \times 2$ measurement matrix that contains the 2D coordinates of n image points in frame i, B_l are the basis shapes of size $n \times 3$ and t_{il} are the time varying shape coefficients and Q_i is the projection matrix for frame i, which in the case of an orthographic camera is a 3×2 matrix that encodes the first two columns of a rotation matrix (therefore it is a Stiefel matrix). By stacking all the measurements for all the frames into a single matrix we have

$$Y = \begin{bmatrix} B_1 \dots B_d \end{bmatrix} \begin{bmatrix} t_1 \otimes Q_1 \dots t_f \otimes Q_f \end{bmatrix} = S \begin{bmatrix} M_1 \dots M_f \end{bmatrix} = SM. \quad (14)$$

Now, we have expressed the measurement matrix as a bilinear interaction between the shape matrix S of size $n \times 3d$ and the motion matrix M of size $3d \times 2f$. This form fit exactly the optimisation problem as presented in Eq. (11). Therefore, in the NRSFM case, the manifold constraint corresponds to $\mathcal{M} = \left\{ t \otimes Q \,:\, t \in \mathbb{R}^d,\, Q \in \mathbb{R}^{3\times 2}, Q^\top Q = I_2 \right\}$, or in other words, the two rows of the rotation matrix Q^T must be orthonormal (i.e. it is a Stiefel matrix). To apply our BALM algorithm, we need first to derive the projector onto $\mathcal{M}$. We now turn to this problem.

5.1 NRSfM Manifold Projector

In [13] Paladini *et al.* derived an exact globally optimal algorithm to project the motion matrices onto the non-rigid motion manifold. However, here, we discuss an alternative which provides an *approximate* projector onto $\mathcal{M}$. The advantage is that our proposed algorithm stills provides accurate estimates while being considerably faster (approximately 100 times in experimental tests).

Let $A \in \mathbb{R}^{3d\times 2}$ be given, and consider the partition $A = \begin{bmatrix} A_1^\top \, A_2^\top \cdots A_d^\top \end{bmatrix}^\top$, where $A_i \in \mathbb{R}^{3\times 2}$. We want to compute $p_{\mathcal{M}}(A)$, that is, we want to solve the optimization problem

$$\begin{aligned} &\text{minimize} \quad \|A - t \otimes Q\|^2 . \\ &\text{subject to } t \in \mathbb{R}^d \\ &\qquad\qquad Q^\top Q = I_2 \end{aligned} \quad (15)$$

If $A \in \mathcal{M}$, that is, if $A = t \otimes Q$ for some $t = (t_1, \dots, t_d)^\top \in \mathbb{R}^d$ and Stiefel matrix Q, then we would have the identity $\sum_{i=1}^{d} A_i A_i^\top = \|t\|^2 QQ^\top$. That is,

the left-hand side of the equation would reveal the underlying Q (up to a right multiplication by a 2×2 rotation). This motivates the following approach, for a generic A: compute $\mathcal{A} = \sum_{i=1}^{d} A_i A_i^\top$ and estimate Q as its dominant Stiefel, that is, corresponding to the 2 top eigenvectors of $\mathcal{A}$. Let $\widehat{Q}$ denotes this Stiefel matrix. Even when $A \in \mathcal{M}$ we do not have, in general, $\widehat{Q} = Q$. Rather, $\widehat{Q} = QR$ for a 2×2 orthogonal matrix R. Thus, we return to problem (15) and we solve for $t \in \mathbb{R}^d$ and R:

$$\begin{aligned} &\text{minimize} \quad \left\| A - t \otimes \left(\widehat{Q}R\right) \right\|^2 \\ &\text{subject to } t \in \mathbb{R}^d \\ &\qquad\qquad R \in \mathbb{R}^{2\times 2},\ R^\top R = I_2 \end{aligned} \quad . \tag{16}$$

For a fixed R, the optimal t is $t_i = \frac{1}{2}\mathsf{tr}\left(R^\top \widehat{Q}^\top A_i\right)$ with $i = 1, \ldots, d$. Plugging t_i into (16) gives the reduced problem over R

$$\begin{aligned} &\text{maximize} \ \sum_{i=1}^{d} \left(\mathsf{tr}\left(R^\top T_i\right)\right)^2 \\ &\text{subject to } R^\top R = I_2 \end{aligned} \tag{17}$$

where $T_i := \widehat{Q}^\top A_i$. Now, a 2×2 rotation matrix R must fall into one of the two following cases $\det(R) = 1$ or $\det(R) = -1$. In both cases the solution is similar. If we have $\det(R) = \pm 1$ we have that $R = \begin{bmatrix} c & \mp s \\ s & \pm c \end{bmatrix}$ for some $(c, s) \in \mathbb{R}^2$, $\|(c, s)\| = 1$. Using this representation in (17) yields

$$\begin{aligned} &\text{maximize} \ \begin{bmatrix} c & s \end{bmatrix} \left(\sum_{i=1}^{d} \begin{bmatrix} T_i(1,1) + T_i(2,2) \\ T_i(2,1) - T_i(1,2) \end{bmatrix} \begin{bmatrix} T_i(1,1) \pm T_i(2,2) \\ T_i(2,1) \mp T_i(1,2) \end{bmatrix}^\top \right) \begin{bmatrix} c \\ s \end{bmatrix} \\ &\text{subject to } \|(c, s)\| = 1 \end{aligned} \tag{18}$$

which can be solved by an eigenvalue decomposition. After examining separately the two cases (i.e. $\pm$), we pick the best.

6 Example 2: BALM for Photometric Stereo

Basri et al. [2] derived a bilinear approximation of the image brightness given by luminance variations. This derivation is based on a spherical harmonics representation of lighting variations and it allows to frame PS as a factorization problem with manifold constraints on one of the bilinear factors. Given a set of images of a Lambertian object with varying illumination, it is possible to extract the dense normal to the surface of the object z, the albedo ρ and the lighting directions l. For a 1^{st} order spherical harmonics approximation, the brightness at image pixel j at frame i can be modelled as $Y_{ij} = l_i^\top \ \rho_j [1 \ z_j^\top]^\top = S_i M_j$, where $l_i \in \mathbb{R}^4$, $\rho_j \in \mathbb{R}$, $z_j \in \mathbb{R}^3$ with $z_j^\top z_j = 1$. A compact matrix form can be obtained for each pixel Y_{ij} as:

$$Y = \begin{bmatrix} Y_{11} & \ldots & Y_{1n} \\ \vdots & \ddots & \vdots \\ Y_{f1} & \ldots & Y_{fn} \end{bmatrix} = \begin{bmatrix} l_1^\top \\ \vdots \\ l_f^\top \end{bmatrix} \left[\rho_1 \begin{bmatrix} 1 \\ z_1 \end{bmatrix} \ldots \rho_n \begin{bmatrix} 1 \\ z_n \end{bmatrix} \right] = SM \tag{19}$$

where a single image i is represented by the vector $Y_i = \left[y_{i1} \dots y_{in} \right]$. Manifold constraints are given by the surface normal constraints which implies $M^{4\times f}$ lying on the manifold defined by: $\mathcal{M} = \left\{ \rho \left[1 \;\; z^\top \right]^\top : \rho \in \mathbb{R}, z \in \mathbb{R}^3, z^\top z = 1 \right\}$. The matrix $S^{n\times 4}$ now contains the collection of lighting directions which which combines the first-order spherical harmonics at each frame i.

6.1 Photometric Stereo Manifold Projector

We now derive the projector onto the manifold. That is, for a given $a \in \mathbb{R}^4$, we show how to solve the associated optimization problem

$$\begin{array}{l} \text{minimize} \;\; \left\| a - \rho \left[1 \;\; z^\top \right]^\top \right\|^2 . \\ \text{subject to } z^\top z = 1 \end{array} \tag{20}$$

The variable to optimize is $(\rho, z) \in \mathbb{R} \times \mathbb{R}^3$. Consider the partition $a = \left[\alpha \, \beta^\top \right]^\top$ with $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^3$. We can rewrite (20) as

$$\begin{array}{l} \text{minimize} \;\; \rho^2 - \alpha\rho - \rho\beta^\top z \, . \\ \text{subject to } z^\top z = 1 \end{array} \tag{21}$$

We denote by $(\rho^\star, z^\star)$ the solution of (21). If $\beta = 0$ then $\rho^\star = \alpha/2$ and $z^\star$ can be any unit-norm vector. If $\beta \neq 0$ then optimizing (21) over z (for a fixed ρ) gives

$$z^\star = \frac{\rho}{|\rho|} \frac{\beta}{\|\beta\|}. \tag{22}$$

Now, if $\alpha \geq 0$ (respectively $\alpha < 0$) then it is clear that $\rho^\star \geq 0$ ($\rho^\star < 0$). Inserting this constraint into (21) leaves a quadratic problem with an easy solution: $\rho^\star = (\alpha + \|\beta\|)/2$ (respectively $\rho^\star = (\alpha - \|\beta\|)/2$).

7 Experiments

7.1 Synthetic Experiments: NRSfM

In our synthetic experiments[1] we used a 3D motion capture sequence showing a deforming face, captured using a VICON system tracking a subject wearing 37 markers. The 3D points were then projected synthetically onto an image sequence 74 frames long using an orthographic camera. To test the performance we computed the relative 3D reconstruction error, which we defined as the Frobenius norm of the difference between the recovered 3D shape $\mathtt{S}$ and the ground truth 3D shape $\mathtt{S}_{GT}$, computed as: $||\mathtt{S} - \mathtt{S}_{GT}||/||\mathtt{S}_{GT}||$. We subtract the centroid of each shape and align them with Procrustes analysis. In the experiments with noise, zero mean additive Gaussian noise was applied with standard deviation $\sigma = \mathrm{n} \times \mathrm{s}/100$ where n is the noise percentage and s is defined as $\max(Y)$ in pixels. In all experiments the number of basis shapes was fixed to $k = 5$. The trials for each level of noise were averaged over 10 runs.

[1] For additional experiments, videos and the code please check: http://www.isr.ist.utl.pt/~adb/the-balm/

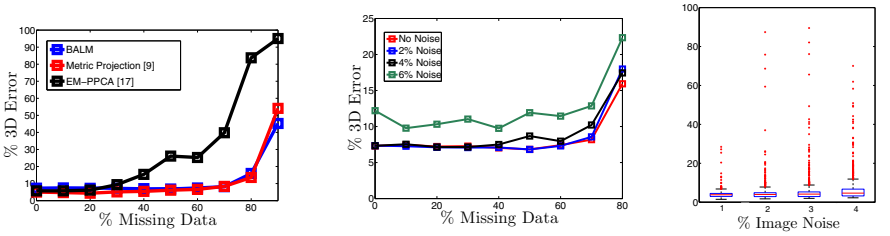


Fig. 1. Synthetic experiment results showing comparison with NRSfM methods (left), robustness of the BALM method with different ratios of missing data and noise (centre) and a boxplot for the convergence of the algorithm for increasing noise levels (right)

Figure 1 (left) shows a comparison between our proposed BALM algorithm and two state of the art methods: Torresani *et al.*'s [17] EM-PPCA and Metric Projections (MP) [13] in the absence of noise and for increasing levels of missing data. While in the case of full data the performance of the algorithms is comparable, BALM and MP clearly outperform EM-PPCA in all cases of missing data higher than 30%. Notice that BALM, has a similar performance to MP, which is a specific NRSfM algorithm. Another interesting fact is the extreme resilience of BALM to missing data. In Figure 1 (centre) we evaluate BALM's performance with respect to noise in the image measurements of up to 6% and missing data ratios of up to 90% in a combined test. The plot shows robustness to noise even for increasing levels of missing data. We also performed a set of convergence tests, in the full data case, for varying levels of noise to evaluate convergence empirically. In this case we used a synthetically generated 3D shape to ensure a known global minimum. Figure 1 (right) shows a boxplot of the 3D error, on the vertical axis, for 1000 trials of the BALM algorithm for each level of noise and no missing data. The algorithm achieves an overall median error close to zero for all the noise levels. Most of the experiments are between 0% and 20% 3D error with very few local minima reaching higher errors.

Regarding run times, in an experiment with 60% missing data, the convergence time was $16s$ for BALM, $10s$ for EM-PPCA [17] and $10m$ for MP [13]. Although the runtimes for BALM and EM-PPCA are comparable, BALM systematically outperforms EM-PPCA in the case of missing data (see Figure 1 (left)). All implementations are in Matlab. However, EM-PPCA runs with partial MEX code while BALM is not optimised. For runtime evaluation we used a Desktop PC AMD X2 2.6Ghz with 4GB of RAM.

Regarding the initialization of the BALM algorithm, the projection matrices, the mean shape and the missing data tracks are first initialised using [12]. We then used Torresani *et al.*'s initialisation [17] to estimate the configuration weights and the basis shapes given the residual of the first rigid solution.

7.2 Real Data: NRSfM

We tested our method on a real sequence of a cushion being bent. We tracked 90 points and we simulated a missing data ratio of 40% by eliminating data points

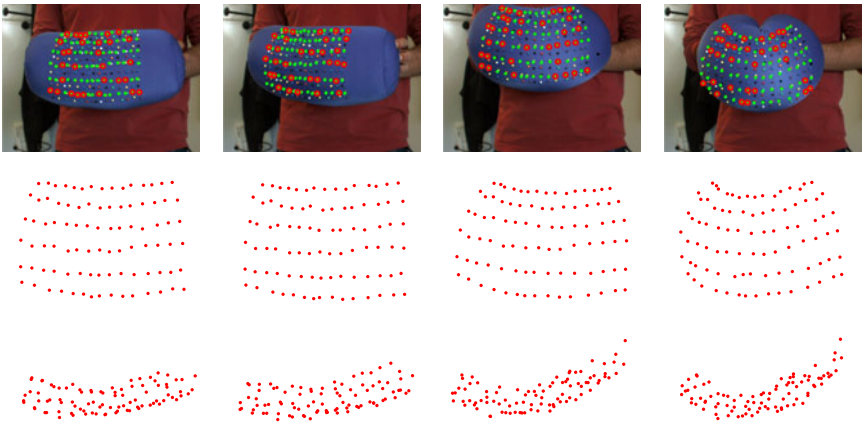


Fig. 2. Cushion sequence with 40% missing data. First row shows four image samples with missing points highlighted with a red circle. Second and third rows show a frontal and side view of the 3D reconstruction using BALM.

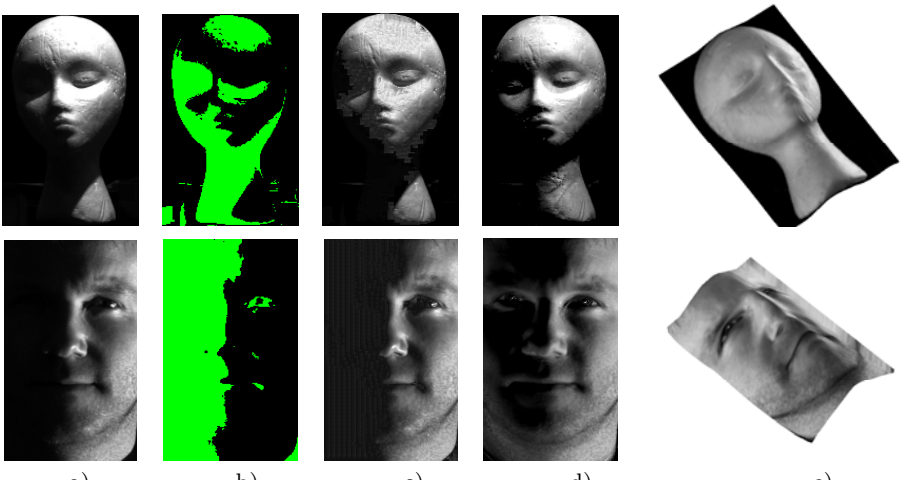


Fig. 3. Real photometric stereo results for the 46-frames long Sculpture sequence with 48% missing data (1^{st} row) and 49-frames long YaleB10 sequence with 31% missing data (2^{nd} row). Each image shows in a) a selected original frame from the sequence; in b) an image mask where green pixels represents missing entries; in c) the image used as initialisation for BALM; in d) the resulting optimised image with a Lambertian model and in e) the reconstructed 3D surface.

manually. Figure 2 shows 4 selected frames and their respective 3D reconstructions (frontal and top view). The bending is clearly observable in the 3D shape plots where BALM shows robustness given the high percentage of missing data.

7.3 Real Data: Photometric Stereo

We present results for the BALM factorization using the photometric projector as presented in Section 6.1. The aim is to extract the 3D surface, albedo and luminance parameters with significant occlusions in the input image data. The occlusions are defined as the darkest and brightest (saturated) pixels in the image sequence since in these areas the Lambertian model will not be satisfied. An initialisation for the missing entries in Y is given by the inpainitng technique [8] which fills the image holes given the known parts of the image. In such a way we exploit image pixels which may resemble the current illumination in the image. The initialisation for $\left(S^{(0)}, M^{(0)}\right)$ is then given by a simple rank-4 SVD on the "inpainted" Y. These affine low-rank components are then normalized as in [2] to comply with the equal norm constraints of the spherical harmonics model. Figure 3 shows the results for two image frames selected from the sequence. The first row of Figure 3 (Sculpture sequence) shows that even with large occlusions the initialisation with inpainting copes well if the overall texture of the object is quite homogenous (although errors can still be noticed on the top of the head and the darkest areas). Note that the final optimised image (Figure 3(d)) reveals some further details in the neck area which were hidden in the original frame. The sequence on the second row, taken from the YaleB database sequence, shows an extreme occlusion in which half of face of the the subject is not visible. In this case inpainting clearly fails to provide a usable initialisation but still the reconstructed shape resembles the subject.

8 Conclusions

We have provided a novel and general optimisation framework for a broad range of bilinear problems in Computer Vision with manifold constraints on the space where the data lies. Our results match state of the art methods in NRSfM and show a considerable improvement in performance for the PS problem. Our approach can deal with a number of entries in the data matrix Y in the order of 10^6 and more. This feature, together with the robustness to missing data, render the BALM algorithm a preferable choice for bilinear modelling in large-scale inference scenarios.

References

1. Bascle, B., Blake, A.: Separability of pose and expression in facial tracking and animation. In: Proc. 6th International Conference on Computer Vision, Bombay, India, pp. 323–328 (1998)
2. Basri, R., Jacobs, D., Kemelmacher, I.: Photometric stereo with general, unknown lighting. International Journal of Computer Vision 72(3), 239–257 (2007)
3. Bertsekas, D.: Constrained optimization and Lagrange multiplier methods. Academic Press, New York (1982)
4. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina, June 2000, pp. 690–696 (2000)

5. Buchanan, A.M., Fitzgibbon, A.: Damped newton algorithms for matrix factorization with missing data. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, San Diego, California, vol. 2, pp. 316–322 (2005)
6. Chandraker, M., Kriegman, D.: Globally optimal bilinear programming for computer vision applications. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8 (2008)
7. Conn, A., Gould, N., Sartenaer, A., Toint, P.: Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints. SIAM Journal on Optimization 6(3), 674–703 (1996)
8. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. on Image Processing 13(9), 1200–1212 (2004)
9. Hestenes, M.: Multiplier and gradient methods. Journal of Optimization Theory and Applications 4(5), 303–320 (1969)
10. Lin, Z., Chen, M., Wu, L., Ma, Y.: The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. UIUC Technical Report UILU-ENG-09-2215 (2009)
11. Mai, F., Hung, Y.S.: Augmented lagrangian-based algorithm for projective reconstruction from multiple views with minimization of 2d reprojection error. Journal of Signal Processing Systems (2009)
12. Marques, M., Costeira, J.P.: Estimating 3D shape from degenerate sequences with missing data. In: Computer Vision and Image Understanding (2008)
13. Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., Agapito, L.: Factorization for Non-Rigid and Articulated Structure using Metric Projections. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Miami, Florida, pp. 2898–2905 (2009)
14. Shaji, A., Chandran, S., Suter, D.: Manifold Optimisation for Motion Factorisation. In: 19th International Conference on Pattern Recognition (ICPR 2008), pp. 1–4 (2008)
15. Tenenbaum, J., Freeman, W.: Separating style and content with bilinear models. Neural Computation 12(6), 1247–1283 (2000)
16. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization approach. International Journal of Computer Vision 9(2) (1992)
17. 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, 878–892 (2008)
18. Wiberg, T.: Computation of principal components when data are missing. In: COMPSTAT 1976, Proc. Comput. Stat., 2nd Symp., Berlin, pp. 229–236 (1976)

Piecewise Quadratic Reconstruction of Non-Rigid Surfaces from Monocular Sequences*

João Fayad[1], Lourdes Agapito[1], and Alessio Del Bue[2]

[1] Queen Mary University of London. Mile End Road, London E1 4NS, UK
[2] Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy

Abstract. In this paper we present a new method for the 3D reconstruction of highly deforming surfaces (for instance a flag waving in the wind) viewed by a single orthographic camera. We assume that the surface is described by a set of feature points which are tracked along an image sequence. Most non-rigid structure from motion algorithms assume a *global* deformation model where a rigid mean shape component accounts for most of the motion and the deformation modes are small deviations from it. However, in the case of strongly deforming objects, the deformations become more complex and a global model will often fail to explain the intricate deformations which are no longer small linear deviations from a strong mean component. Our proposed algorithm divides the surface into overlapping patches, reconstructs each of these patches individually using a quadratic deformation model and finally registers them imposing the constraint that points shared by patches must correspond to the same 3D points in space. We show good results on challenging motion capture and real video sequences with strong deformations where global methods fail to achieve good reconstructions.

1 Introduction

The recovery of the 3D structure of a deformable object from a sequence of images acquired with a single camera is an inherently ill-posed problem since different shapes can give rise to the same image measurements. The problem becomes particularly challenging when no initial model or prior information is known about the observed surface and the only input information is a set of 2D correspondences between points along the sequence, a problem defined as Non-Rigid Structure from Motion (NRSfM). Most NRSfM methods are based on the low rank basis shape model defined by Bregler *et al.* [1] in which the deformable 3D shape is represented as a linear combination of a set of basis shapes with time varying coefficients. This model has allowed the development of a family of algorithms which have been a successful extension of classical rigid factorization algorithms to the non-rigid case [2,3,4,5,6,7,8,9]. However, so far, these algorithms have been demonstrated on simple sequences where the

* This work was partially funded by the European Research Council under ERC Starting Grant agreement 204871-HUMANIS.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 297–310, 2010.

deformations are only small deviations from a rigid component. In the case of strongly deforming objects, the deformations become more complex and a global model will often fail to explain those intricate deformations which are no longer small linear deviations of a strong mean component.

In this paper we argue that in such cases a *local* piecewise reconstruction of the object can achieve accurate reconstructions where the global methods fail. Our proposed algorithm divides the surface into overlapping patches, reconstructs each of these patches individually and finally registers all the patches together imposing the constraint that points shared by patches must correspond to the same 3D points in space. Our method is generic in the sense that it does not rely on any specific reconstruction However, in our experiments, we have found that the quadratic deformations model recently proposed by Fayad *et al.* [10] provides the best local reconstructions. There are two important advantages of the quadratic deformations model. Firstly, it assumes that locally the object deforms according to three fixed modes of deformation (linear, quadratic and cross terms) that account for stretching, sheering, bending, twisting. Our experiments on real data reveal that this model is well suited to encode local deformations, even in cases when the surface overall has strong deformations. Secondly, the parameters of the model have a physical meaning and therefore prior knowledge about the way in which the object deforms can be injected into the model by fixing parameters to certain values or imposing priors. For instance if the surface is known to be inextensible the values of the parameters that account for the stretching can be fixed to zero.

We show results on challenging motion capture and real video sequences with strong deformations and a very small amount of camera rotation (which adds to the difficulty of obtaining accurate reconstructions) and where we show that global methods fail to provide good results. To the best of our knowledge, this is one of the first attempts to model complex deformations using a NRSfM approach.

1.1 Related Work

The reconstruction of deformable surfaces from monocular video sequences is a problem that has received increased attention over the last decade. It is a hard problem which still remains unsolved because it is inherently ill-posed. The approaches developed so far to deal with the ambiguities can be classified according to the amount of *a priori* knowledge that they require. This can range from the strongest assumption of a known model [11], to which the current measurements are fit, to the model-free approach championed by NRSfM methods [1,12,6,8,4,9] or intermediate assumptions such as the requirement of a reference image in which the shape is known *a priori* [13,14].

In this paper we focus on the model-free NRSfM approaches where the only input is a set of point correspondences between images. Most approaches stem from the seminal work of Bregler *et al.* [1] who introduced the linear basis shape model. Since then, the focus has been on overcoming the problems caused by ambiguities and degeneracies by proposing different optimization schemes and

the use of generic 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 [12,6,8]. Torresani *et al.* [7] formulate the problem using Probabilistic Principal Components Analysis introducing priors as a Gaussian distribution on the deformation weights. Other approaches focus on ensuring that the solution lies on the correct motion manifold where the metric constraints are exactly satisfied [15]. The linear basis shape model has also allowed the formulation of closed form solutions both for the affine [5] and perspective [4,9] cases. However, closed form solutions are known to be very sensitive to noise [3,7] and cannot deal with missing data.

It is only very recently that NRSfM algorithms have departed from the low rank linear basis model. In their dual formulation, Akhter *et al.* [16] propose the use of the DCT basis to express the trajectory of each 3D point, which has the advantage that the basis is object independent and does not need to be estimated. Rabaud and Belongie [17] assume that only small neighbourhoods of shapes are well modelled by a linear subspace and they adopt a manifold learning technique to constrain the number of degrees of freedom of the object. Fayad *et al.* [10] recently proposed the use of a quadratic deformation model to explain strong deformations of a non-rigid object. While this model is physically grounded and can explain well natural deformations, the severe drawback of their approach is that they use it as a global model. This is clearly only true when the entire object behaves according to a quadratic model (for instance a bending motion but with a single bending point), which severely restricts its use. More recently, discriminative and generative methods have been combined to reconstruct 3D deformable surfaces [18]. A new method has also been proposed to solve the 3D surface reconstruction and point matching problems simultaneously [19].

Other local to global methods include the work of Salzmann *et al.* [20] who proposed a machine learning approach in which a local deformation model was learned using motion capture data. However, the models are specialised to the specific surface that the learning was carried on and require additional training data. In work developed in parallel to ours, and similar in spirit, Taylor *et al.* [21] have also proposed a piecewise approach to solve the NRSfM problem using locally-rigid SfM to reconstruct a soup of rigid triangles. Our approach is closely related to the work of Varol *et al.* [22] who also propose a model free piecewise reconstruction of non-rigid surfaces but based on planar patches with overlapping points. An initial planar reconstruction is estimated for each patch from the homographies estimated from pairwise correspondences. Patches are then merged and finally the 3D point cloud is refined after fitting it to a mesh assuming temporal smoothness constraints. The strength of their approach is that matching is only required between pairs of consecutive views instead of requiring long tracks. However, their method can suffer when the planar assumption is not satisfied, as would be the case in the reconstruction of highly deformable surfaces. In contrast, while our method does require long tracks to reconstruct the patches, the quadratic model can cope with much stronger local deformations, which is, ultimately, the focus of this work.

2 Piecewise Non-Rigid Structure from Motion

In this paper we propose a piecewise approach to the estimation of the 3D shape of a deformable object observed with an orthographic camera. Our only assumption is that we have a sufficient number of features tracked across an image sequence. We focus on the case where the overall deformations of the object are too complex to be explained by a single global deformation model. Instead, our insight is that the quadratic deformation model [10] is a good local model that can be used to reconstruct local patches. These can then be registered together to provide a global reconstruction imposing that the overlapping points are the same 3D points in space. The first step in our approach is to divide the 3D surface into patches. Note that throughout the paper we use the word patches to mean surface patches whether or not they are planar.

Algorithm 1. Piecewise Reconstruction of Highly Deformable Surfaces

Input: 2D correspondences of points tracked through the sequence.
Output: 3D reconstruction of the global surface for every frame.
1: Divide surface into N regular patches $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_N\}$
2: Reconstruct individual patches using the quadratic deformation model.
3: Align individual 3D reconstructions.
4: Final optimization.

2.1 Division of the Surface into Patches

The aim of this work is to provide a fully automatic method to deal with any type of 3D non-rigid surfaces, whether planar, such as a piece of paper, or non planar such as a beating heart or a torso. However, often some *a priori* information exists about the nature of the object being observed. Here, we provide a solution to the division of the surface into regular patches in three different situations: when a reference 3D shape is known for an image in the sequence, when the surface is known to be a planar shape but a reference image is not available and finally in the general case where no *a priori* knowledge is available about the surface. In all our experiments we divide the object into a set of regular patches.

Known reference shape: A number of recent approaches to non-rigid shape reconstruction from monocular sequences rely on the assumption that the shape of the object is known in some reference image [13,14]. For instance, often the surfaces of interest are sheets of paper or cloth and it is reasonable to assume that they are viewed in a planar configuration in the first frame. If this assumption were satisfied, then it would be simple to divide the surface into regular patches, based on a division of the known surface and the corresponding image points.

Planar surfaces: In some situations, we know in advance that the surface being reconstructed is a deforming plane (a sheet of paper or a flag waving in the wind), but a reference image for that shape is not known. In this case, we propose a method based on the isometric low-dimensional mapping method

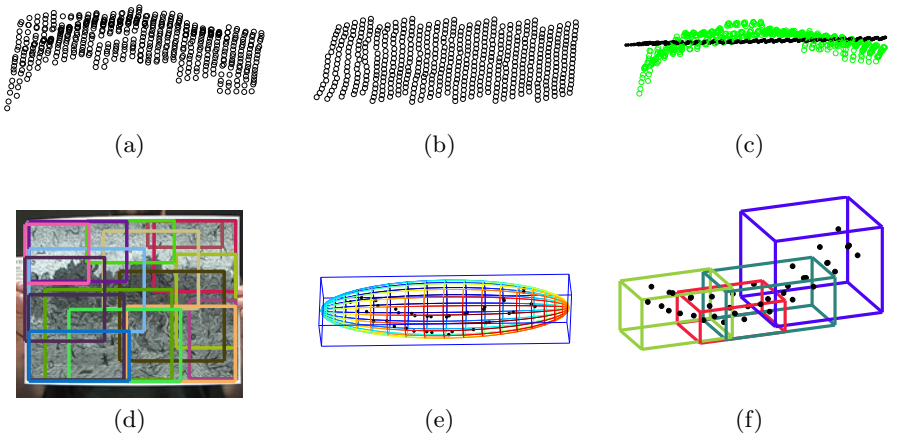


Fig. 1. (a) Reconstructed mean shape of the flag shape (see Figure 5) using rigid factorization. (b) Result of applying Isomap to the surface. (c) Side view of the shape before and after Isomap. (d) Division of a planar surface into patches. (e) Volume bounding box. (f) Division of volume into pieces.

Isomap [23]. First we reconstruct a *mean* shape of the surface by applying Tomasi and Kanade's rigid factorization algorithm [24] to a few frames or to the entire sequence. Since the object is non-rigid, this average rigid surface will not be planar. Therefore it would not be an easy task to divide it into regular patches. However, we can use Isomap [23] to compute an isometric low-dimensional embedding (the 2D flat surface) of the higher dimensional data (the deformed 3D surface). In other words, Isomap will find an isometric mapping of the deformed *mean* surface, obtained by rigid factorization, onto a 2D plane. Figure 1(a)-(c) illustrates the process. Due to noise in the data and to the sparseness of the 2D tracks the embedding will not be exactly isometric. However, it is a good enough representation to use for the division of the surface into regular overlapping pieces.

No *a priori* knowledge: Finally, if there is no *a priori* information about the shape of the object we proceed as follows. First we apply the rigid factorization algorithm [24] to some frames of the sequence to obtain a *mean* shape. An ellipsoid is then fitted to the mean shape in order to estimate the volume of the object. Finally a bounding box of that volume is computed and divided into regular overlapping pieces. Figure 1 shows this process being applied to a cylindrical surface.

3 Reconstruction of Individual Patches

Once the surface has been divided into a set of regular patches or pieces, each of these can be reconstructed in 3D using any state of the art non-rigid reconstruction algorithm. However, we propose to use the quadratic deformation model [10].

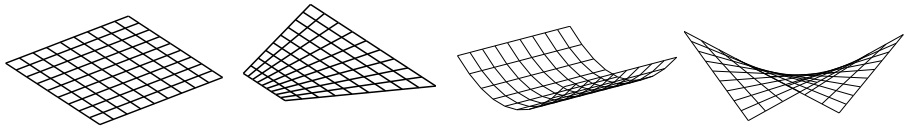


Fig. 2. Quadratic deformation modes applied to a synthetic planar patch

Intuitively, the quadratic model can encode bending, stretching, shearing and twisting modes of deformation which are natural ways in which objects deform locally. We now give a short overview of this model.

3.1 Quadratic Deformation Model

The quadratic deformation model was first defined in the context of computer graphics [25] to produce realistic simulations of deformable objects for computer games. It was recently applied to the problem of NRSfM by Fayad *et al.* [10]. The model encapsulates three modes of deformation (linear, quadratic and cross-terms) which are combined with time varying coefficients to give the NRSfM shape in each frame. The modes are built from the matrix $\mathtt{S}^{\mathtt{L}}$ that contains the 3D coordinates of p points on the surface in what can be defined as the rest shape. Additionally, $\mathtt{S}^{\mathtt{L}}$ needs to be described in a reference frame aligned with the principal deformation axis, and centered at the centroid of the object. The linear mode is exactly the $3 \times p$ matrix $\mathtt{S}^{\mathtt{L}}$, the quadratic mode $\mathtt{S}^{\mathtt{Q}}$ contains the squared values of the coordinates and the cross-terms mode $\mathtt{S}^{\mathtt{C}}$ contains their bilinear products. The shape matrix $\mathtt{S}$ is then built by stacking the three modes:

$$\mathtt{S}^{\mathtt{L}} = \begin{bmatrix} X_1 \dots X_p \\ Y_1 \dots Y_p \\ Z_1 \dots Z_p \end{bmatrix} \mathtt{S}^{\mathtt{Q}} = \begin{bmatrix} X_1^2 \dots X_p^2 \\ Y_1^2 \dots Y_p^2 \\ Z_1^2 \dots Z_p^2 \end{bmatrix} \mathtt{S}^{\mathtt{C}} = \begin{bmatrix} X_1Y_1 \dots X_pY_p \\ Y_1Z_1 \dots Y_pZ_p \\ Z_1X_1 \dots Z_pX_p \end{bmatrix} \mathtt{S} = \begin{bmatrix} \mathtt{S}^{\mathtt{L}} \\ \hline \mathtt{S}^{\mathtt{Q}} \\ \hline \mathtt{S}^{\mathtt{C}} \end{bmatrix} \quad (1)$$

The deformed shape at every frame i is then defined as:

$$\mathtt{S}_i = \begin{bmatrix} \mathtt{L}_i & \mathtt{Q}_i & \mathtt{C}_i \end{bmatrix} \mathtt{S} = \mathtt{D}_i \mathtt{S} \quad (2)$$

where $\mathtt{L}_i$, $\mathtt{Q}_i$ and $\mathtt{C}_i$ are the 3×3 deformation coefficient matrices, for frame i, associated with the linear, quadratic and cross-terms deformations. Notice that the shape matrix $\mathtt{S}$ is fixed for all the frames while the quadratic deformation matrix $\mathtt{D}_i$ varies frame-wise. Figure 2 shows examples of deformations allowed by this model applied to a planar surface.

Assuming the non-rigid motion is observed by an orthographic camera, the 2D coordinates of a generic point j in frame i can be written as:

$$\mathbf{w}_{ij} = \mathtt{R}_i \, \mathtt{D}_i \, \mathbf{S}_j + \mathbf{t}_i \quad (3)$$

where $\mathtt{R}_i$ and $\mathbf{t}_i$ are respectively the truncated 2×3 rotation matrix and the 2D translation vector for frame i, and $\mathbf{S}_j$ the 9-vector with the three shape modes of point j.

3.2 Non-linear Optimization

The model parameters are estimated by minimizing the image reprojection error of all the observed points given by the following cost-function:

$$\min_{\mathbf{q}_i,\mathbf{t}_i,\mathtt{L}_i,\mathtt{Q}_i,\mathtt{C}_i} \mathcal{J}(\mathbf{q}_i,\mathbf{t}_i,\mathtt{L}_i,\mathtt{Q}_i,\mathtt{C}_i) = \min_{\mathbf{q}_i,\mathbf{t}_i,\mathtt{L}_i,\mathtt{Q}_i,\mathtt{C}_i} \sum_{i,j}^{f,p} ||\mathbf{w}_{ij} - \mathtt{R}_i(\mathbf{q}_i)\,[\mathtt{L}_i\,\mathtt{Q}_i\,\mathtt{C}_i]\,\mathbf{S}_j - \mathbf{t}_i||^2 \tag{4}$$

where $\mathtt{R}_i(\mathbf{q}_i)$ indicates that, internally, the rotations are parameterised using quaternion vectors $\mathbf{q}_i$, which are therefore the parameters being optimized.

To prevent ambiguities, temporal smoothness priors are added to the cost function to penalise camera poses and deformation coefficients that vary too much from one frame to the next. The smoothness terms are given by

$$\lambda \sum_{i=2}^{f} |||[\mathtt{L}_i\,\mathtt{Q}_i\,\mathtt{C}_i] - [\mathtt{L}_{i-1}\,\mathtt{Q}_{i-1}\,\mathtt{C}_{i-1}]|||^2 + \gamma \sum_{i=2}^{f} ||\mathbf{t}_i - \mathbf{t}_{i-1}||^2 + \rho \sum_{i=2}^{f} ||\mathbf{q}_i - \mathbf{q}_{i-1}||^2 \quad . \tag{5}$$

These prior terms are added to the cost function $\mathcal{J}$ which is then optimised using bundle adjustment [26]. The deformation parameters in $[\mathtt{L}_i\,\mathtt{Q}_i\,\mathtt{C}_i]$ are initialised to encode a rigid object. This is achieved imposing $\mathtt{L}_i = \mathtt{I}_{3\times 3}$ and $\mathtt{Q}_i = \mathtt{C}_i = \mathtt{0}$. The rotations are initialized as $\mathtt{R} = \mathtt{W}\,\mathrm{pinv}(\mathtt{S}^{\mathtt{L}})$, where $\mathtt{W}$ is the $2F \times P$ stack of all the image measurements. These matrices are then projected to the orthonormal subspace by forcing its singular values to be unitary. Regularization weights γ and ρ are determined empirically and fixed for all the experiments.

3.3 Estimation of the Rest Shape

Recall that the rest shape is fixed for the entire sequence. What causes the deformations of the object are the frame by frame changes in the deformation matrices $[\mathtt{L}_i\,\mathtt{Q}_i\,\mathtt{C}_i]$. It is also important to notice that the rest shape is not equivalent to the average shape observed throughout the sequence. In fact, the rest shape can be thought of as the least deformed instance of the observed shape, when no forces are being applied to it. For instance in the case of a sheet of paper or a piece of cloth the rest shape would be a plane.

Similarly to Fayad *et al.* we assume that the observed sequence contains some initial frames where the object does not deform much. We have found that this does not impose restrictions on the type of sequences we can deal with. We run rigid factorization [24] on these frames and recover the mean shape. We have found that as few as 5 frames are sufficient to obtain a good reconstruction of the rest shape.

***A Priori* Knowledge.** When there is prior knowledge available about the surface being observed this should be taken into account. For instance when the surface is known to be planar, a similar process to the one described in Section 2.1 can be used to flatten an initial estimate of the surface obtained via rigid factorization. In this case we apply Isomap [23] to the reconstructed

surface and recover its lower dimensional planar embedding. The coordinates of the rest shape can then be optimised within the non-linear scheme described in Equation (4) to account for noise in the initial estimation.

4 From Local Patches to a Global Reconstruction

4.1 Connecting Patches

The algorithm described in the previous section allows us to reconstruct the set of 3D patches $\mathcal{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_N\}$ independently using the quadratic deformation model. At this point it is important to remind the reader that any other NRSfM algorithm could be used for the individual patches. However, in Section 5 we will show that we have obtained best results with the quadratic model.

Since the patches are reconstructed independently, each in their own reference frame, they will be reconstructed at different depths. This results in a set of unconnected patches (see Figure 3) which need to be aligned to reconstruct the entire surface. The pieces are registered by using the constraints provided by the overlapping regions, as shared points between two patches must have the same 3D coordinates. Since the patches are reconstructed from the same set of images optimizing image reprojection error the only misalignments that can occur are along the depth direction. In other words, the reconstructions of two different patches will only be misaligned along the Z coordinate of their translation vectors. Additionally, the relative depth of each patch can only be determined up to a sign flip, an ambiguity that cannot be resolved under orthography. This ambiguity manifests itself as a difference in sign in the Z coordinate of the translation vectors between two patches.

The translation ambiguity is solved by registering the Z coordinates of the shared points between pairs of patches. In practice, to minimize errors, we register the Z coordinate of the centroids of both patches. To solve the sign ambiguity a reference patch is chosen. For each other patch the sign of the translation vector which provides the reconstruction that is more consistent with the reference patch is then selected. The division of the surface into patches was performed in a way that every patch overlaps with at least another patch. Thus, starting

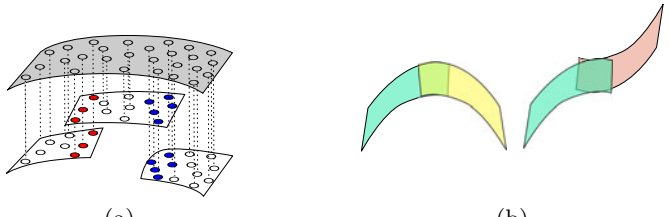


Fig. 3. (a) Reconstruction of shared points in different patches differ by a translation on the Z axis. (b) Representation of the ambiguity on the sign of the Z coordinate of the reconstructions.

from any patch, we can register patches one by one until they form a consistent 3D point cloud representing the non-rigid shape.

Overlapping regions between patches will have multiple 3D reconstructions of the same 2D point. However, a one to one match between 2D and 3D points is desired. Thus, after registration is performed, the final step is to merge the multiple representations of a point into a single 3D point. This is done by averaging all the 3D representations of a given point.

4.2 Final Optimization

Once individual patches are reconstructed and initially aligned, a final global optimization step is used to refine the results. This refinement is achieved by imposing the constraint that shared points must have the same 3D coordinates. This can be done by applying the original cost function defined in Equation 4 to all the patches and adding a prior term that penalises reconstructions in which the 3D coordinates of shared points between patches are distant:

$$\sum_{i,j}^{f,p} \sum_{n \in \Theta_j} \left|\left| \mathbf{w}_{ij}^{(n)} - \hat{\mathbf{w}}_{ij}^{(n)} \right|\right|^2 + \eta \sum_{k \in \Theta_j / \{n\}} \left|\left| \hat{\mathbf{X}}_{ij}^{(n)} - \hat{\mathbf{X}}_{ij}^{(k)} \right|\right|^2, \tag{6}$$

where $\mathbf{w}_{ij}^{(n)}$ are the 2D coordinates of point j in frame i in patch (n), Θ_j is the set of N patches that contain point j, and $\hat{\mathbf{X}}_{ij}^{(n)}$ are the 3D coordinates of point j in frame i reconstructed from patch (n) using the quadratic model described in Section 3.1. The global optimization step, as in the local case, is done using bundle adjustment [26].

5 Experiments

Our approach aims at reconstructing highly deformable sequences where usual NRSfM methods fail[1]. To be able to provide quantitative results and to allow comparisons with other methods, we have chosen to use a challenging example of a motion capture (MOCAP) sequence of a flag/cloth waving in the wind [27]. This sequence is particularly difficult as it contains strong, rapidly varying deformations appearing through the whole surface. We show some frames of the *MOCAP flag sequence*[2] with added texture in Figure 5.

Local vs. Global: Our first set of experiments was designed to show that current NRSfM models based on global models fail to achieve good reconstructions on a sequence of an object undergoing strong, agile or complex deformations. In Figure 4 we show ground truth 3D data together with some examples of

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

[2] Note that the apparent stripe-like structure of the flag is not due to our piecewise reconstruction. It is present in the ground truth 3D data as a consequence of the regular way in which the markers were placed.

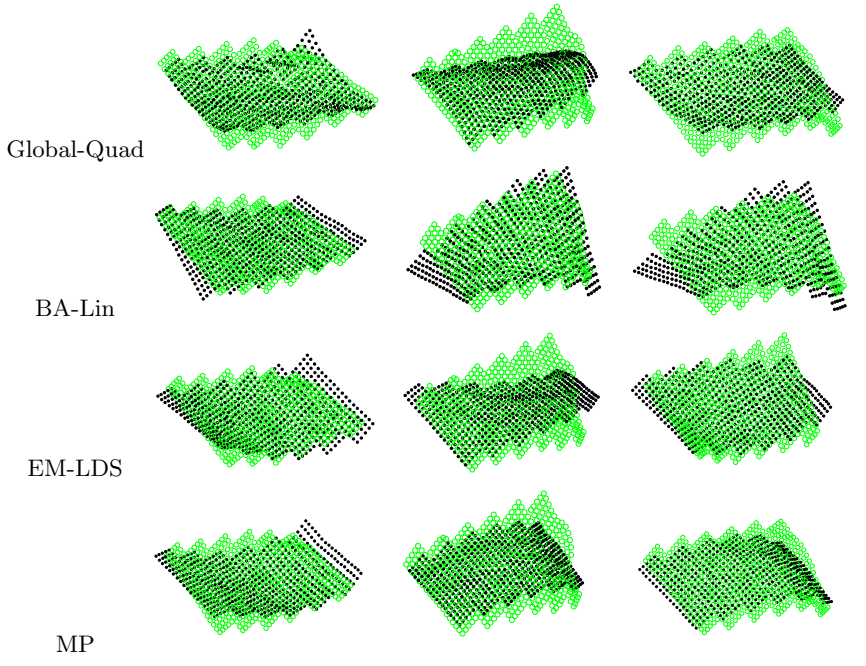


Fig. 4. Reconstructions of the flag sequence from using the Global-Quad, BA-lin, EM-LDS and MP methods. Ground truth is represented by green circles while reconstructions are represented as black dots.

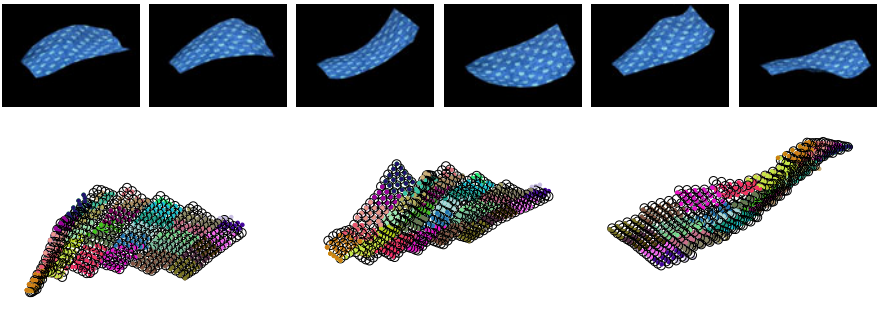


Fig. 5. Top: Some frames of the *MOCAP flag* [27] sequence with added texture. Bottom: Reconstruction of 3 frames of the flag sequence with our new piecewise quadratic deformations model. Ground truth is presented as black circles, reconstructed points are shown as coloured dots where the colour indicates the patch they belong to.

3D reconstructions obtained using 4 different global SfM methods: 1) (Global-Quad) original global formulation of the quadratic model [10], 2)(BA-Lin) linear combination of basis shape model with Bundle Adjustment optimization [28], 3) (EM-LDS) NRSfM method proposed by Torresani *et al.* [7] and 4) Metric

Projections method [15]. Table 1 (right) shows the reconstruction error given by the different algorithms. These experiments reveal that state of the art NRSfM methods based on global models fail to reconstruct this highly deforming object.

Justification of quadratic model as best local model: In this section we justify our choice of the quadratic deformation model as the most adequate local model to express strong, natural local deformations. In Table 1 (middle column) we show the 3D reconstruction error (measured with respect to ground truth values) averaged over all the patches in the flag for each of the algorithms mentioned in the previous section. The 3D error is defined as $||\hat{\mathtt{X}}_i - \mathtt{X}_i^{GT}||/||\mathtt{X}_i^{GT}||$ averaged through all the frames i, where $\hat{\mathtt{X}}$ is the 3D reconstruction and $\mathtt{X}_i^{GT}$ is the groud truth data. We remove the centroid of these shapes on every frame and register them using Procrustes analysis before computing the 3D error.

It is clear that the quadratic model outperforms all the other methods (4.05% error vs. errors between 15% and 29%). Each reconstruction algorithm was ran with its *out of the box* initialization. In the left column of Table 1 we show the average patch 3D errors when the mean shape for algorithms (BA-Lin) and (EM-LDS) and the rest shape for the (Global-Quad) algorithm were initialized with the known ground truth flat shape given by the motion capture data. This experiment shows that *a priori* knowledge of the 3D shape of the surface improves the reconstructions. The quadratic model continues to outperform others by an order of magnitude (3.18% error vs. errors between 15% and 19%).

Piecewise quadratic reconstruction on MOCAP sequences (flag and cylinder): Applying the piecewise quadratic deformation model to the *MOCAP flag* sequence results in the reconstructions show in Figure 5 where the coloured points are the reconstructed points (colour encodes the patch they belong to) and the circles are the ground truth values. The rest shape was initialised from rigid factorization of 5 frames followed by flattening of the shape using Isomap. The object was divided into 36 overlapping patches. Patch size ranges from 21 to 75 points, with an average size of 54.2 points, with the total number of points in the object being 540. A pair of overlapping patches share, on average, 17.6 points. The 3D reconstruction error can be found in Table 1 (right column). Results show that in this challenging sequence, our model is able to provide a very accurate reconstruction, with only 3.25% of 3D error. Recall that the other NRSfM methods gave errors ranging between 15% and 26%. In Figure 6 we show reconstructions (cyan dots) and ground truth values (black circles) of the *MOCAP cylinder* used in [10]. We report an average 3D error of 1.97% compared to a 3D error of 5% reported in [10]. Therefore the piecewise approach greatly improves the results of the global algorithm. In this sequence the object was divided into 4 overlapping pieces, with two having 16 points and the other two 19 points, from a total of 39 points. A pair of overlapping pieces share, on average, 7.8 points.

Piecewise quadratic reconstruction of real sequence: Figure 7 (top and middle rows) shows the reconstruction of a real sequence showing a paper bending [22]. Reconstructed points are represented in different colours showing the

Table 1. 3D Reconstruction error for different NRSfM methods on the flag sequence

Algorithm	Patch GT init (%)	Patch Own init (%)	3D error whole Flag (%)
Global-Quad [10]	3.18	4.05	15.79
BA-Lin [28]	17.48	16.51	26.29
EM-LDS [7]	15.34	15.85	17.09
MP [15]	-	29.77	18.57
Piecewise-Quad	-	-	**3.25**

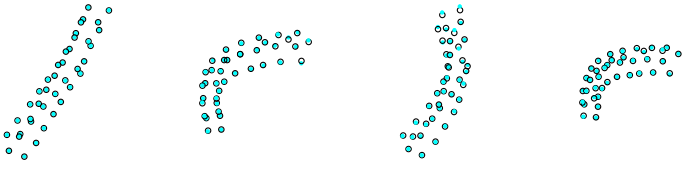


Fig. 6. Results of the reconstruction of the *(MOCAP cylinder)* sequence used in [10]. Blue dots are reconstructed points and black circles are ground truth values.

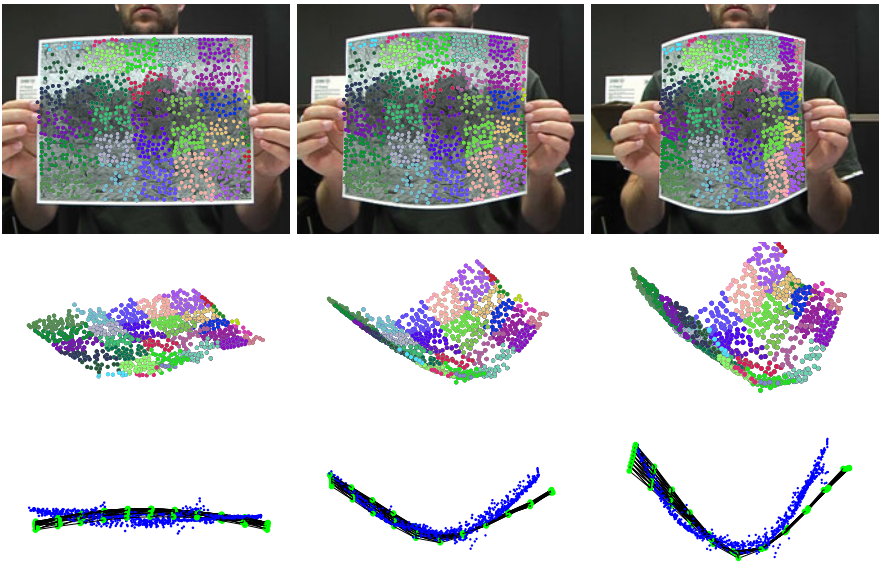


Fig. 7. Reconstruction of a real sequence of a paper bending [22]. The different colours show the different patches. Top: 2D reprojection of the points. Centre: 3D reconstructions with our piecewise reconstruction. Bottom: Comparison of our reconstruction (blue point cloud) with Varol *et al.*'s method [22] (mesh with green vertices).

36 patches used in the reconstruction. In this case, the size of the patches ranges between 38 and 167 points, with an average size of 113 points, from a total of 871 points. A pair of overlapping patches share on average 31.47 points. The rest shape was obtained running rigid factorization on 8 frames and then using Isomap to obtain the 2D embedding plane. We also provide a qualitative comparison with the mesh obtained with Varol *et al.*'s method [22] (Figure 7, bottom row). When the deformation is strongest our reconstruction provides a more realistic curved shape, whereas Varol *et al.*'s appears to be a piecewise planar approximation. Moreover, the video provided as supplementary material shows that Varol *et al.*'s reconstruction has strong flickering while ours is smooth.

6 Conclusions and Future Work

We have proposed a new piecewise NRSfM algorithm to reconstruct highly deformable surfaces. We view this new algorithm as a step forward towards modelling of realistic complex deformations within the NRSfM paradigm. So far, the models proposed in NRSfM are global. Our piecewise reconstruction model is based on the quadratic deformation model. Our experimental results show that the quadratic model outperforms other methods as a local model and, as part of a piecewise model, we have reported very low 3D reconstruction errors on complex MOCAP and real sequences. In future work we will address the issue of dealing with missing data due to self-occlusions.

References

1. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: CVPR (2000)
2. Brand, M.: Morphable models from video. In: CVPR (2001)
3. Brand, M.: A direct method for 3D factorization of nonrigid motion observed in 2D. In: CVPR (2005)
4. Xiao, J., Kanade, T.: Uncalibrated perspective reconstruction of deformable structures. In: ICCV (2005)
5. Xiao, J., Chai, J., Kanade, T.: A closed-form solution to non-rigid shape and motion recovery. IJCV (2006)
6. Del Bue, A., Lladó, X., Agapito, L.: Non-rigid metric shape and motion recovery from uncalibrated images using priors. In: CVPR (2006)
7. Torresani, L., Hertzmann, A., Bregler., C.: Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. PAMI (2008)
8. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-fine low-rank structure-from-motion. In: CVPR (2008)
9. 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)
10. Fayad, J., Del Bue, A., Agapito, L., Aguiar, P.M.Q.: Non-rigid structure from motion using quadratic deformation models. In: BMVC (2009)
11. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH (1999)

12. Aanæs, H., Kahl, F.: Estimation of deformable structure and motion. In: Workshop on Vision and Modelling of Dynamic Scenes, Copenhagen, Denmark (2002)
13. Salzmann, M., Fua, P.: Reconstructing sharply folding surfaces: A convex formulation. In: CVPR (2009)
14. Perriollat, M., Hartley, R., Bartoli, A.: Monocular template-based reconstruction of inextensible surfaces. In: BMVC (2008)
15. Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., Agapito, L.: Factorization for non-rigid and articulated structure using metric projections. In: CVPR (2009)
16. Akhter, I., Sheikh, Y.A., Khan, S., Kanade, T.: Nonrigid structure from motion in trajectory space. In: Neural Information Processing Systems (2008)
17. Rabaud, V., Belongie, S.: Re-thinking non-rigid structure from motion. In: CVPR (2008)
18. Salzmann, M., Urtasun, R.: Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In: CVPR (2010)
19. Shaji, A., Varol, A., Torresani, L., Fua, P.: Simultaneous Point Matching and 3D Deformable Surface Reconstruction. In: CVPR (2010)
20. Salzmann, M., Urtasun, R., Fua, P.: Local deformation models for monocular 3d shape recovery. In: CVPR (2008)
21. Taylor, J., Jepson, A.D., Kutulakos, K.N.: Non-rigid structure from locally-rigid motion. In: CVPR (2010)
22. Varol, A., Salzmann, M., Tola, E., Fua, P.: Template-free monocular reconstruction of deformable surfaces. In: ICCV (2009)
23. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science (2000)
24. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization approach. In: IJCV (1992)
25. Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless deformations based on shape matching. In: SIGGRAPH (2005)
26. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment - a modern synthesis. In: ICCV (2000)
27. White, R., Crane, K., Forsyth, D.: Capturing and animating occluded cloth. In: ACM Trans. on Graphics (2007)
28. Del Bue, A., Smeraldi, F., Agapito, L.: Non-rigid structure from motion using ranklet–based tracking and non-linear optimization. In: IVC (2007)

Extrinsic Camera Calibration Using Multiple Reflections

Joel A. Hesch[1], Anastasios I. Mourikis[2], and Stergios I. Roumeliotis[1,⋆]

[1] University of Minnesota, Minneapolis MN 55455, USA
{joel,stergios}@cs.umn.edu
[2] University of California, Riverside CA 92521, USA
mourikis@ee.ucr.edu

Abstract. This paper presents a method for determining the six-degree-of-freedom (DOF) transformation between a camera and a base frame of interest, while concurrently estimating the 3D base-frame coordinates of unknown point features in the scene. The camera observes the reflections of fiducial points, whose base-frame coordinates are known, and reconstruction points, whose base-frame coordinates are unknown. In this paper, we examine the case in which, due to visibility constraints, none of the points are directly viewed by the camera, but instead are seen via reflection in multiple planar mirrors. Exploiting these measurements, we *analytically* compute the camera-to-base transformation and the 3D base-frame coordinates of the unknown reconstruction points, without *a priori* knowledge of the mirror sizes, motions, or placements with respect to the camera. Subsequently, we refine the analytical solution using a maximum-likelihood estimator (MLE), to obtain high-accuracy estimates of the camera-to-base transformation, the mirror configurations for each image, and the 3D coordinates of the reconstruction points in the base frame. We validate the accuracy and correctness of our method with simulations and real-world experiments.

1 Introduction

Extrinsic calibration – the task of computing the six-degrees-of-freedom (DOF) transformation between the camera's frame of reference and a base frame of interest – is a prerequisite for many vision-based tasks. For example, mobile robots often rely on cameras to detect and locate obstacles during their operation. When this is the case, accurate knowledge of the camera-to-body transformation is necessary for precise navigation. Estimating this transformation is often not a trivial matter: one common problem is that the robot's chassis may not lie within the camera's direct field of view (see Fig. 1), which means that one cannot apply calibration methods that rely on direct observations of known points on the robot body. This is only one example application where the camera

⋆ This work was supported by the University of Minnesota (Digital Technology Center), the University of California Riverside (Bourns College of Engineering), and the National Science Foundation (IIS-0643680, IIS-0811946, IIS-0835637).

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 311–325, 2010.

extrinsic parameters must be computed without a direct line-of-sight to any of the available fiducial points. In this work, we show that in these cases one can exploit observations of the fiducial points through reflections in multiple mirrors, to extrinsically calibrate the camera.

The objective of our work is to design an automated procedure for determining the 3D transformation between the camera frame and a *base frame* of interest, by utilizing the measurements of fiducial points, whose position in the base frame is known *a priori*. We examine the scenario in which the known points are not directly visible to the camera, but can only be observed through reflections in multiple mirrors. We maneuver the mirrors to provide the camera with multiple views of the fiducial points; however, no prior information about the mirrors' sizes or motions with respect to the camera is assumed. Instead, the configurations of the mirrors and the camera-to-base transformation are both treated as unknowns to be computed from the measurements. In addition to these quantities, in this paper we show how the images recorded by the camera through the mirror reflections can be used to estimate the positions of additional points in the scene, whose locations were not known *a priori*.

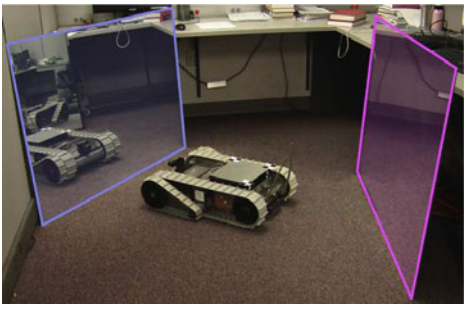


Fig. 1. A mobile robot views its reflection in two mirrors (front highlighted in blue, back highlighted in purple). The robot visually tracks point features to estimate the camera-to-base frame transformation, and a 3D point-cloud representation of its chassis.

Thus, the problem we address is that of jointly estimating the camera's configuration, mirrors' configurations, and scene structure, using observations of points visible only though a number of reflections. The main contribution of this work is an algorithm for *analytically* computing all the unknown quantities, given the available measurements. The analytically computed estimates are subsequently refined by a maximum likelihood estimator (MLE), implemented by an iterative nonlinear minimization process, to obtain the estimates with the highest precision possible while accounting for measurement noise. In addition to the theoretical importance of an analytical solution, its practical utility is demonstrated by both our simulation results and our real-world experiments. These tests demonstrate that using the analytical solution to seed the MLE results in accurate estimates, which can be computed in a small number of iterations.

2 Related Work

Extrinsic camera calibration has been widely studied for the case in which known points are *directly observed* by the camera [1,2,3]. Unfortunately, in many realistic scenarios, the known points may not lie within the camera's field of view

(see Fig. 1). This motivates studying the more limiting scenarios, in which the points of interest can only be observed through reflection using one or more planar mirrors. The literature in this field is substantially sparser. We note that catadioptric systems in which one or more mirrors are employed to reconstruct a scene, such as those presented in [4,5,6,7] are not directly relevant here. First, in these methods the location of the mirrors is assumed to be known in advance. Second, in these systems each point is observed multiple times in each image (directly, as well as through reflections). In our method each point is only observed *once* per image, via its reflection in the moving planar mirrors.

A system which employs a moving planar mirror for 3D scene reconstruction was introduced by Jang et al. [8]. By exploiting a combination of known markers on a moving mirror and vanishing points in the reflections, they first solved for the position of the mirror with respect to the camera, and subsequently determined the 3D scene based on synthetic stereo from multiple reflections. In contrast to this approach, we do not utilize known mirror markers, since doing so would introduce constraints on the mirror motions (i.e., the markers must always be visible to the camera). This enhances the flexibility of our method, but it renders our problem more challenging, since multiple images are required to compute the mirror configurations.

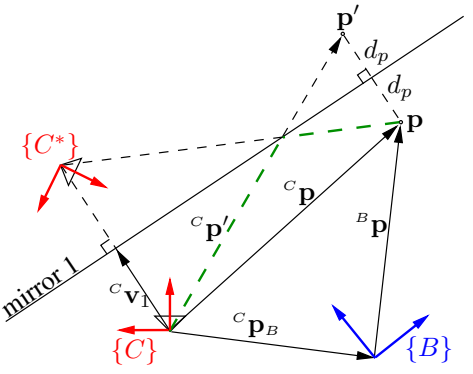


Fig. 2. The camera observes ${}^C\mathbf{p}'$, which is the reflection of $\mathbf{p}$. The base frame is $\{B\}$, while $\{C\}$ is the camera frame, and $\{C^*\}$ is the equivalent *imaginary* camera frame which lies behind the mirror. ${}^C\mathbf{v}_1$ is the shortest vector from $\{C\}$ to the mirror. The distance d_p is measured from $\mathbf{p}$ to the mirror, along $\mathbf{v}_1$. The vector ${}^C\mathbf{p}_B$ is the origin of $\{B\}$ with respect to $\{C\}$, while ${}^B\mathbf{p}$ and ${}^C\mathbf{p}$ denote the coordinates of $\mathbf{p}$ expressed in $\{B\}$ and $\{C\}$, respectively. The dashed green line is the path of the reflection.

Kumar et al. [9] presented a vision system that utilized a moving planar mirror to determine the transformations between multiple cameras with non-overlapping fields of view. Each camera in turn viewed the reflection of a calibration grid, whose position with respect to the cameras was fixed. To solve the problem, each camera was required to view the calibration pattern from five vantage points. Subsequently, the measurement constraints were transformed into a set of linear equations which were solved for the unknown transformations. In contrast to [9], the method presented here requires observations of only three fiducial points, and is also applicable in cases where reflection in a single mirror is not sufficient to make the points visible to the camera.

Finally, in our previous work we addressed the problem of extrinsic camera calibration using a *single* mirror, presenting both an analytical solution [10], and

an MLE to obtain estimates of the camera-to-base transformation [11]. We now extend this single-mirror extrinsic calibration method and address the multi-mirror case, as described in the following sections.

3 Problem Formulation

Our main goal in this work is to simultaneously determine: (i) the six-DOF transformation between the camera frame, $\{C\}$, and a base frame of interest, $\{B\}$, and (ii) the 3D base-frame coordinates of N_r "reconstruction points." To this end, we assume that N_f fiducial points, whose coordinates in the base frame are known *a priori*, are observed in N_c images. We address the most limiting scenario, in which the points do not lie in the camera's direct field of view, but are only visible via reflection in N_v mirrors (each point is reflected N_v times, and each point is only observed once in each image). Since the placement of the mirrors in each image is unknown, in addition to the camera configuration and the positions of the reconstruction points, it is necessary to jointly estimate the configurations of the mirrors in all the images. In what follows, we start by presenting the model describing the camera measurements.

3.1 Measurement Model

The camera observes each point, $\mathbf{p}$, via its reflection $\mathbf{p}'$, as shown in Fig. 2. The measurement model which describes this observation is divided in two components: (i) the camera projection model and (ii) the expression which describes the geometric relationship between $\mathbf{p}'$ and $\mathbf{p}$ as a function of the mirror and camera configurations.

Single-mirror constraint: In the single-mirror scenario, we obtain two equations from geometry (see Fig. 2):

$$^{C}\mathbf{p}' = {}^{C}\mathbf{p} + 2d_p \frac{^{C}\mathbf{v}_1}{\|^{C}\mathbf{v}_1\|}, \quad d_p = \|^{C}\mathbf{v}_1\| - \frac{^{C}\mathbf{v}_1^T}{\|^{C}\mathbf{v}_1\|} {}^{C}\mathbf{p}, \tag{1}$$

where $^{C}\mathbf{p}'$ is the vector from the origin of $\{C\}$ to the reflected point $\mathbf{p}'$, $^{C}\mathbf{p}$ is vector from $\{C\}$ to $\mathbf{p}$, $^{C}\mathbf{v}_1$ is the mirror vector, which is the shortest vector form the origin of $\{C\}$ to the reflective surface, and d_p is the distance between the mirror and the point $\mathbf{p}$ measured along the direction of $^{C}\mathbf{v}_1$. In order to simplify the notation, we refer to $^{C}\mathbf{v}_1$ as $\mathbf{v}_1$ in the remainder of the paper. In addition to the two geometric constraints derived from Fig. 2, we also exploit the coordinate transformation between $^{C}\mathbf{p}$ and $^{B}\mathbf{p}$, i.e.,

$$^{C}\mathbf{p} = {}^{C}_{B}\mathbf{R}\, {}^{B}\mathbf{p} + {}^{C}\mathbf{p}_B, \tag{2}$$

where ${}^{C}_{B}\mathbf{R}$ is the matrix which rotates vectors from $\{B\}$ to $\{C\}$, and $^{C}\mathbf{p}_B$ is the origin of $\{B\}$ with respect to $\{C\}$. We substitute (2) into (1), and rearrange the terms to obtain

$$^{C}\mathbf{p}' = \left(\mathbf{I}_3 - 2\frac{\mathbf{v}_1\mathbf{v}_1^T}{\mathbf{v}_1^T\mathbf{v}_1}\right) {}^{C}\mathbf{p} + 2\,\mathbf{v}_1 = \mathbf{M}_1\,({}^{C}_{B}\mathbf{R}\, {}^{B}\mathbf{p} + {}^{C}\mathbf{p}_B) + 2\,\mathbf{v}_1, \tag{3}$$

where $\mathbf{M}_1 = \left(\mathbf{I}_3 - 2\left(\mathbf{v}_1\mathbf{v}_1^T/\mathbf{v}_1^T\mathbf{v}_1\right)\right)$ is the Householder transformation matrix corresponding to the mirror reflection. Equation (3) is equivalently expressed in homogeneous coordinates as

$$\begin{bmatrix} {}^C\mathbf{p}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_1 & 2\,\mathbf{v}_1 \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^C_B\mathbf{R} & {}^C\mathbf{p}_B \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{b}_1 \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{p} \\ 1 \end{bmatrix}, \tag{4}$$

where the pair, $\mathbf{A}_1 = \mathbf{M}_1\,{}^C_B\mathbf{R}$ and $\mathbf{b}_1 = \mathbf{M}_1\,{}^C\mathbf{p}_B + 2\mathbf{v}_1$, defines a composite homogeneous/reflection transformation, which converts ${}^B\mathbf{p}$ into ${}^C\mathbf{p}'$.

N_v-mirror constraint: The single-mirror case is readily extended to the N_v-mirror case, by noting that each additional mirror in the system adds a reflection transformation parameterized by the corresponding mirror vector. Hence, the geometric relationship for a base-frame point observed through N_v mirrors is

$$\begin{bmatrix} {}^C\mathbf{p}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{N_v} & 2\,\mathbf{v}_{N_v} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \cdots \begin{bmatrix} \mathbf{M}_1 & 2\,\mathbf{v}_1 \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^C_B\mathbf{R} & {}^C\mathbf{p}_B \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{N_v} & \mathbf{b}_{N_v} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{p} \\ 1 \end{bmatrix}, \tag{5}$$

where $\{\mathbf{A}_{N_v}, \mathbf{b}_{N_v}\}$ is a homogeneous transformation comprising the N_v mirror vectors and the camera-to-base transformation. Their structure can be defined recursively by expanding (5), i.e.,

$$\begin{aligned} \mathbf{A}_{N_v} &= \mathbf{M}_{N_v}\cdots\mathbf{M}_1\,{}^C_B\mathbf{R} = \mathbf{M}_{N_v}\mathbf{A}_{N_v-1} && (6)\\ \mathbf{b}_{N_v} &= \mathbf{M}_{N_v}\cdots\mathbf{M}_1\,{}^C\mathbf{p}_B + 2\,\mathbf{M}_{N_v}\cdots\mathbf{M}_2\mathbf{v}_1 + \cdots \\ &\quad + 2\,\mathbf{M}_{N_v}\mathbf{M}_{N_v-1}\mathbf{v}_{N_v-2} + 2\,\mathbf{M}_{N_v}\mathbf{v}_{N_v-1} + 2\,\mathbf{v}_{N_v} \\ &= \mathbf{M}_{N_v}\mathbf{b}_{N_v-1} + 2\,\mathbf{v}_{N_v}. && (7) \end{aligned}$$

We extend this recursive structure to include the camera-to-base transformation:

$$\mathbf{A}_0 = {}^C_B\mathbf{R}, \quad \mathbf{b}_0 = {}^C\mathbf{p}_B, \tag{8}$$

which will simplify the discussion of our analytical solution (see Sect. 4.2).

Perspective projection model: The reflected point, $\mathbf{p}'$, is observed by the camera whose intrinsic camera parameters are assumed to be known [12]. The normalized image coordinates of the measurement are described by the perspective projection model:

$$\mathbf{z} = \frac{1}{p_3}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + \boldsymbol{\eta} = \mathbf{h}({}^C\mathbf{p}') + \boldsymbol{\eta}, \quad {}^C\mathbf{p}' = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^T, \tag{9}$$

where $\boldsymbol{\eta}$ is the pixel noise, which is modeled as a zero-mean white Gaussian process with covariance matrix $\sigma_\eta^2\mathbf{I}_2$. Equations (5) and (9) define the measurement model that expresses the point's observed image coordinates $\mathbf{z}$, as a function of the vector ${}^B\mathbf{p}$, the *unknown* camera-to-base transformation $\{{}^C_B\mathbf{R}, {}^C\mathbf{p}_B\}$, and the *unknown* configurations of the mirrors with respect to the camera, $\mathbf{v}_1, \ldots, \mathbf{v}_{N_v}$.

4 Camera-to-Base Transformation Analytical Solution

We address the problem of obtaining an analytical solution for all the unknown quantities in two steps: first, we obtain an analytical solution for the camera-to-base frame transformation, as well as for the mirrors' configurations. Once these quantities have been determined, we subsequently obtain an analytical solution for the 3D positions of the reconstruction points, as explained in Sect. 5.

4.1 Relationship to PnP

To obtain the analytical solution for the camera and mirror configurations, we exploit the similarity of our problem to the n-point perspective pose estimation problem (PnP). Specifically, in the standard formulation of PnP we seek the six-DOF transformation $\{{}^C_B\mathbf{R}, {}^C\mathbf{p}_B\}$ between a camera frame $\{C\}$, and a base frame $\{B\}$, given perspective measurements of N_f fiducial points, ${}^B\mathbf{p}_i$, $i = 1, \ldots, N_f$:

$$\mathbf{z}_i = \mathbf{h}({}^C\mathbf{p}_i) + \boldsymbol{\eta}_i, \quad \text{where} \quad \begin{bmatrix} {}^C\mathbf{p}_i \\ 1 \end{bmatrix} = \begin{bmatrix} {}^C_B\mathbf{R} & {}^C\mathbf{p}_B \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{p}_i \\ 1 \end{bmatrix}. \tag{10}$$

By comparison of (10) to (5) and (9), the relationship between the mirror-based calibration and PnP problems becomes evident. Specifically, while in the PnP the only unknowns are $\{{}^C_B\mathbf{R}, {}^C\mathbf{p}_B\}$, in the mirror-based calibration problem we have additional unknowns, corresponding to the mirror configurations. All these unknowns, however, are "encoded" in the pair $\{\mathbf{A}_{N_v}, \mathbf{b}_{N_v}\}$, which appears in (5) and (9) in the same way as the pair $\{{}^C_B\mathbf{R}, {}^C\mathbf{p}_B\}$ does in (10). This similarity allows us to use the solution of the PnP, which is a well-studied problem, as a first step towards solving the multi-mirror calibration problem. Specifically, we first exploit the similarity to PnP to solve for the pair $\{\mathbf{A}_{N_v}, \mathbf{b}_{N_v}\}$, and next utilize (6) and (7) to solve for the camera and mirror configurations, as explained in Sect. 4.2.

In order to compute the pair $\{\mathbf{A}_{N_v}, \mathbf{b}_{N_v}\}$, we need to take the special properties of the matrix $\mathbf{A}_{N_v}$ into consideration. Specifically, $\mathbf{A}_{N_v}$ is the product of N_v Householder reflection matrices and one rotation matrix. As a result, $\mathbf{A}_{N_v}$ is unitary, and when N_v is even it is a rotation matrix (its determinant is equal to $+1$). Therefore, when N_v is even we can directly apply a PnP solution method to obtain $\mathbf{A}_{N_v}$ and $\mathbf{b}_{N_v}$. Any algorithm is suitable here; in the experiments presented in this paper, $N_f = 3$, and we solve the corresponding P3P problem using the solution presented by Fischler and Bolles [2].

When N_v is odd, the determinant of $\mathbf{A}_{N_v}$ is equal to -1, and therefore we cannot directly employ a PnP solution method. However, we can use a very simple transformation to bring the problem to a form in which the PnP algorithms can be directly applied. Specifically, we can transform $\mathbf{A}_{N_v}$ into a rotation matrix by applying an additional *known* reflection of our choice. For instance, if we change the sign of the y coordinates of all points in the image, this corresponds to applying a reflection across the xz-plane in the camera frame. Thus, the measurement equation in this case becomes $\mathbf{z} = \mathbf{h}({}^{\breve{C}}\mathbf{p}) + \boldsymbol{\eta}$, where

$$\begin{bmatrix} {}^{\breve{C}}\mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} (\mathbf{I}_3 - 2\mathbf{e}_2\mathbf{e}_2^T) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{C}\mathbf{p}' \\ 1 \end{bmatrix} = \begin{bmatrix} (\mathbf{I}_3 - 2\mathbf{e}_2\mathbf{e}_2^T) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A}_{N_v} & \mathbf{b}_{N_v} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{B}\mathbf{p} \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} {}^{\breve{C}}_{B}\mathbf{R} & {}^{\breve{C}}\mathbf{p}_B \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{B}\mathbf{p} \\ 1 \end{bmatrix}, \tag{11}$$

where $\mathbf{e}_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$. Note that the matrix ${}^{\breve{C}}_{B}\mathbf{R}$ is a rotation matrix, not a reflection. Thus, after negating the y-coordinate of all the points in the image, we can solve the PnP to obtain solution(s) for the unknown transformation $\{{}^{\breve{C}}_{B}\mathbf{R}, {}^{\breve{C}}\mathbf{p}_B\}$. Subsequently, given the PnP solution, we recover $\mathbf{A}_{N_v}$ and $\mathbf{b}_{N_v}$ through the following relationship which follows directly from (11)

$$\begin{bmatrix} (\mathbf{I}_3 - 2\mathbf{e}_2\mathbf{e}_2^T) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} {}^{\breve{C}}_{B}\mathbf{R} & {}^{\breve{C}}\mathbf{p}_B \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{N_v} & \mathbf{b}_{N_v} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{12}$$

4.2 Analytical Solution for the Camera and Mirror Configurations

We next describe how we use the computed $\{\mathbf{A}_{N_v}, \mathbf{b}_{N_v}\}$, to analytically solve for the camera-to-base transformation and the configuration of the mirrors. Before presenting our analytical solution, we discuss the conditions necessary for such a solution to exist.

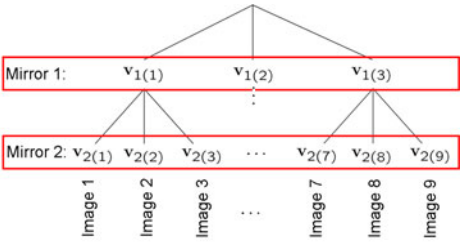


Fig. 3. Mirror configurations for the 2-mirror case depicted as a ternary tree

We start by noting that, in order to compute a discrete set of solutions to the PnP problem in the preceding section, at least three non-collinear points are required. For a unique solution, at least four points in a general configuration are necessary [3]. By analogy, in the mirror-based calibration problem at least $N_f = 3$ known points are needed in order to obtain solutions for $\{\mathbf{A}_{N_v}, \mathbf{b}_{N_v}\}$, and when $N_f \geq 4$ the solution is unique in each image (barring degenerate configurations).

We now examine the number of unknowns in the system, and compare it with the number of available constraint equations. Note that, $\mathbf{A}_{N_v}$ is a unitary matrix, and only has 3 degrees of freedom. Thus determining the pair $\{\mathbf{A}_{N_v}, \mathbf{b}_{N_v}\}$ from $N_f \geq 3$ points in each image only provides us with 6 independent constraint equations (3 for the unitary matrix $\mathbf{A}_{N_v}$, and 3 for the vector $\mathbf{b}_{N_v}$), which we can utilize to solve for the camera and mirror configurations. Thus, from N_c images, we can obtain $6N_c$ independent constraints.

On the other hand, if N_v mirrors are used, then each of the mirrors introduces 3 unknowns in the system (the elements of vector $\mathbf{v}_i$). Moreover, the 6-DOF camera-to-base transformation introduces 6 additional unknowns. Therefore, if in each of the N_c images each mirror moves to a new configuration, the total number of unknowns is equal to $3N_vN_c+6$. Thus, if $N_v > 1$, moving each mirror

to a new configuration in each image results in a problem where the number of unknowns, $3N_vN_c + 6$, is larger than the number of constraints available, $6N_c$. Therefore, some restrictions on the mirrors' motion must be imposed, for a unique solution to exist.

Our strategy is to move the mirrors in a specific order so as to isolate different system unknowns in each observation. Fig. 3 depicts our approach for the two-mirror case. For each configuration of mirror 1, we move mirror 2 in three different locations and record an image. Each "leaf node" in the ternary tree corresponds to one image, and tracing the path from the leaf node to the root of the tree specifies the mirror configurations for that image ($\mathbf{v}_{\ell(m)}$, denotes the m-th configuration of mirror ℓ). The mirror-configuration tree helps visualize the order in which the mirrors move and the unknown quantities are computed.

In the two-mirror case, the total number of mirror 1 configurations is three, and the total number of mirror 2 configurations is nine. In the general case of N_v mirrors, this strategy results in $6 + 3 \times (3 + 3^2 + \ldots + 3^{N_v}) = 6 + \frac{9}{2}(3^{N_v} - 1)$ unknowns, and 6×3^{N_v} constraints, which is an over-determined problem. We stress that, even though the problem is overdetermined, using a smaller number of images is not possible. At least three different configurations of each mirror are necessary, in order to compute a unique solution for the camera-to-base transformation. If only two configurations per mirror are used, then a continuum of solutions exists [10]. This dictates the proposed mirror motion strategy.

We next describe our algorithm for determining all the mirror-configuration vectors, $\mathbf{v}_{\ell(m)}$, as well as the transformation $\{{}^C_B\mathbf{R}, {}^C\mathbf{p}_B\}$. This algorithm is a recursive one: first, all the mirror configurations for mirror N_v are determined, then we proceed to mirror $N_v - 1$, and so on.

Determining the configuration of the N_v-th mirror: Specifically, once $\{\mathbf{A}_{N_v(m)}, \mathbf{b}_{N_v(m)}\}$, $m = 1, \ldots, 3^{N_v}$, are obtained using the PnP solution, we exploit the structure of (6) and (7) to compute the vectors $\mathbf{v}_{N_v(m)}$, $m = 1, \ldots, 3^{N_v}$. We proceed to compute these vectors in sets of three, corresponding to those images for which the configurations of mirrors 1 through $N_v - 1$ remain fixed.

To demonstrate the procedure, we focus on $\mathbf{v}_{N_v(m)}$, $m = 1, 2, 3$, which are the first three configurations for mirror N_v. In this case

$$\mathbf{A}_{N_v(m)} = \mathbf{M}_{N_v(m)}\mathbf{A}_{N_v-1(1)}, \quad m = 1, 2, 3, \tag{13}$$

where $\mathbf{M}_{\ell(m)}$ denotes the Householder reflection matrix for the m-th configuration of mirror ℓ. For each pair (m, m') of mirror-N_v configurations, if we let $\mathbf{r}_{mm'}$ be a unit vector perpendicular to both $\mathbf{v}_{N_v(m)}$ and $\mathbf{v}_{N_v(m')}$, we obtain

$$\begin{aligned}\mathbf{A}_{N_v(m)}\mathbf{A}^T_{N_v(m')}\mathbf{r}_{mm'} &= \mathbf{M}_{N_v(m)}\mathbf{A}_{N_v-1(1)}\mathbf{A}^T_{N_v-1(1)}\mathbf{M}^T_{N_v(m')}\mathbf{r}_{mm'} \\ &= \mathbf{M}_{N_v(m)}\mathbf{M}^T_{N_v(m')}\mathbf{r}_{mm'} = \mathbf{r}_{mm'},\end{aligned} \tag{14}$$

where we exploited $\mathbf{A}_{N_v-1(1)}\mathbf{A}^T_{N_v-1(1)} = \mathbf{I}_3$, and $\mathbf{v}^T_{N_v(m)}\mathbf{r}_{mm'} = \mathbf{v}^T_{N_v(m')}\mathbf{r}_{mm'} = 0$. The above result states that $\mathbf{r}_{mm'}$ is the eigenvector of $\mathbf{A}_{N_v(m)}\mathbf{A}^T_{N_v(m')}$ corresponding to the unit eigenvalue. Since $\mathbf{A}_{N_v(m)}\mathbf{A}^T_{N_v(m')}$ is a known matrix, we

can compute $\mathbf{r}_{mm'}$ up to sign. Moreover, since the vectors $\mathbf{r}_{mm'}$ and $\mathbf{r}_{mm''}$, for $m' \neq m''$, are both perpendicular to $\mathbf{v}_{N_v(m)}$, we can use the following expressions to obtain the vectors $\mathbf{v}_{N_v(m)}$, $m = 1, 2, 3$, up to scale:

$$\mathbf{v}_{N_v(1)} = c_{N_v(1)}\,\mathbf{r}_{13} \times \mathbf{r}_{12}, \quad \mathbf{v}_{N_v(2)} = c_{N_v(2)}\,\mathbf{r}_{21} \times \mathbf{r}_{23}, \quad \mathbf{v}_{N_v(3)} = c_{N_v(3)}\,\mathbf{r}_{13} \times \mathbf{r}_{23}, \tag{15}$$

where $c_{N_v(m)}$, $m = 1, 2, 3$ are unknown scalars. In order to determine these scalars, we note that they appear linearly in (7), along with the vector $\mathbf{b}_{N_v-1(1)}$. Using these equations we can formulate the over-determined linear system:

$$\begin{bmatrix} \mathbf{M}_{N_v(1)} & 2\mathbf{r}_{13} \times \mathbf{r}_{12} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ \mathbf{M}_{N_v(2)} & \mathbf{0}_{3\times 1} & 2\mathbf{r}_{21} \times \mathbf{r}_{23} & \mathbf{0}_{3\times 1} \\ \mathbf{M}_{N_v(3)} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} & 2\mathbf{r}_{13} \times \mathbf{r}_{23} \end{bmatrix} \begin{bmatrix} \mathbf{b}_{N_v-1(1)} \\ c_{N_v(1)} \\ c_{N_v(2)} \\ c_{N_v(3)} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{N_v(1)} \\ \mathbf{b}_{N_v(2)} \\ \mathbf{b}_{N_v(3)} \end{bmatrix}. \tag{16}$$

The solution of this system provides us with the scale factors $c_{N_v(m)}$, $m = 1, 2, 3$, which, in turn, allows us to fully determine the vectors $\mathbf{v}_{N_v(m)}$, $m = 1, 2, 3$, using (15). Following the same procedure, we determine the configuration, $\mathbf{v}_{N_v(m)}$, of mirror N_v for the remaining images, $m = 4, \ldots, 3^{N_v}$.

Dealing with multiple solutions: Up to this point we have assumed that each of the PnP solutions was unique, in order to simplify the presentation of the analytical solution. However, in the general case multiple PnP solutions may exist (e.g., up to 4 admissible ones when $N_f = 3$). In that case, we compute an analytical solution by following the above procedure for each of the PnP solutions, and select the solution which yields the minimum residual in the solution of (16). If the measurements were noise-free, we would expect only one solution to have zero error, since the nonlinear system we are solving is over-constrained. In the realistic case where noise is present, we have found that choosing the solution with the minimum error is a suitable way of rejecting invalid solutions.

Solving for the remaining mirror vectors, and the camera-to-base transformation: Once a solution for the vectors $\mathbf{v}_{N_v(m)}$ has been computed, we proceed to eliminate the unknowns corresponding to mirror N_v from the problem, using [see (6) and (7)]:

$$\mathbf{A}_{N_v-1(j)} = \mathbf{M}^{-1}_{N_v(m)} \mathbf{A}_{N_v(m)} \tag{17}$$

$$\mathbf{b}_{N_v-1(j)} = \mathbf{M}^{-1}_{N_v(m)} \left(\mathbf{b}_{N_v(m)} - 2\,\mathbf{v}_{N_v(m)} \right). \tag{18}$$

for $m = 1, \ldots, 3^{N_v}$, $j = \lceil m/3 \rceil$ (where $\lceil \cdot \rceil$ denotes the round-up operation). Note that since three different images (three different values of m) correspond to the same index j, we can obtain three estimates of $\mathbf{A}_{N_v-1(j)}$ and $\mathbf{b}_{N_v-1(j)}$. Due to the presence of noise, these estimates will generally be slightly different. To obtain a single, "average" estimate for $\mathbf{A}_{N_v-1(j)}$ and $\mathbf{b}_{N_v-1(j)}$, we employ a least-squares procedure similar to the one presented in [10]. Proceeding recursively as above, we can compute all mirrors' configurations. Moreover, as explained in Sect. 3.1, the camera-to-base transformation $\mathbf{A}_0$ and $\mathbf{b}_0$, can be obtained using the same process.

5 Analytical Solution for Scene Reconstruction

In the previous sections, we discussed how to analytically determine the camera-to-base transformation and the mirror configurations using the observations of the fiducial points. We now turn our attention to computing the 3D coordinates of all reconstruction points, whose coordinates are *not known a priori*. We describe an analytical method to compute their coordinates, which will be subsequently refined in an MLE for determining a more precise estimate while accounting for measurement noise. We assume that each image contains the reflections of at least three fiducial points and N_r reconstruction points. The fiducial points are utilized to determine the mirror vectors as well as the camera-to-base transformation (see Sect. 4.2), and the observations of the N_r reconstruction points are utilized to compute a 3D point-cloud representation of important objects in the scene (e.g., the robot chassis) with respect to the base frame.

Consider a single reconstruction point, ${}^B\mathbf{p}$, observed via reflection through N_v mirrors, for example, in N_c images. If we denote the measured unit-vector direction towards the reflected point as ${}^C\hat{\mathbf{p}}'_j$, then we obtain

$$s_j\, {}^C\hat{\mathbf{p}}'_j = \mathbf{A}_{N_v(j)}\, {}^B\mathbf{p} + \mathbf{b}_{N_v(j)}, \quad j = 1, \ldots, N_c, \tag{19}$$

where the scalar s_j is the *unknown* distance to the reflected point in image j. This measurement model is equivalent to the perspective projection model defined in Sect. 3.1. Equation (19) is linear in the unknowns s_j and ${}^B\mathbf{p}$. When ${}^B\mathbf{p}$ is observed in at least two images (i.e., $N_c \geq 2$), we can form an overdetermined set of linear equations ($N_c + 3$ unknowns and $3N_c$ constraints), which is solved to obtain the distance to the reconstruction point in each image, as well as the point's coordinates in the base frame:

$$\begin{bmatrix} \mathbf{A}_{N_v(1)} & -{}^C\hat{\mathbf{p}}'_1 & \cdots & \mathbf{0}_{3\times 1} \\ \mathbf{A}_{N_v(2)} & \mathbf{0}_{3\times 1} & \cdots & \mathbf{0}_{3\times 1} \\ \vdots & & \ddots & \vdots \\ \mathbf{A}_{N_v(N_c)} & \mathbf{0}_{3\times 1} & \cdots & -{}^C\hat{\mathbf{p}}'_{N_c} \end{bmatrix} \begin{bmatrix} {}^B\mathbf{p} \\ s_1 \\ \vdots \\ s_{N_c} \end{bmatrix} = \begin{bmatrix} -\mathbf{b}_{N_v(1)} \\ -\mathbf{b}_{N_v(2)} \\ \vdots \\ -\mathbf{b}_{N_v(N_c)} \end{bmatrix}. \tag{20}$$

6 MLE Refinement of Analytical Solutions

After the analytical solutions for the mirror configurations, camera-to-base transformation, and position of the reconstruction points have been computed, we refine them by applying maximum likelihood estimation. The vector of all unknown parameters is given by:

$$\mathbf{x} = \begin{bmatrix} {}^C\mathbf{p}_B^T & {}^C\bar{q}_B^T & {}^C\mathbf{v}_{1(1)}^T & \cdots & {}^C\mathbf{v}_{N_v(3^{N_v})}^T & {}^B\mathbf{p}_1^T & \cdots & {}^B\mathbf{p}_{N_r}^T \end{bmatrix}^T, \tag{21}$$

where N_r is the number of reconstruction points and ${}^C\bar{q}_B$ is the unit quaternion of rotation between frames $\{B\}$ and $\{C\}$. We use $\mathcal{Z}$ to denote the set of all available measurements, and the likelihood of the measurements is given by

$$L(\mathcal{Z};\mathbf{x}) = \prod_{i=1}^{N_p}\prod_{j=1}^{N_c} p(\mathbf{z}_{ij};\mathbf{x}) = \prod_{i=1}^{N_p}\prod_{j=1}^{N_c} \frac{1}{2\pi\sigma_\eta^2}\exp\Big[-\frac{(\mathbf{z}_{ij}-\mathbf{h}_{ij}(\mathbf{x}))^T(\mathbf{z}_{ij}-\mathbf{h}_{ij}(\mathbf{x}))}{2\sigma_\eta^2}\Big], \quad (22)$$

where $\mathbf{h}_{ij}(\mathbf{x})$ is the measurement function defined in (9), and $N_p = N_r + N_f$ is the total number of reconstruction and fiducial points. Maximizing the likelihood, in the presence of i.i.d. Gaussian noise, is equivalent to minimizing the following non-linear least-squares cost function:

$$J(\mathbf{x}) = \sum_{i,j}(\mathbf{z}_{ij} - \mathbf{h}_{ij}(\mathbf{x}))^T(\mathbf{z}_{ij} - \mathbf{h}_{ij}(\mathbf{x})), \quad (23)$$

which is done iteratively using the Levenberg-Marquardt (LM) algorithm for estimating the parameter vector in (21).

7 Simulations

In this section, we present simulation results that demonstrate the feasibility of computing the camera-to-base transformation using the proposed approach.

We evaluate the accuracy of the analytically computed camera-to-base transformation (see Sect. 4.2) as well as the uncertainty in the MLE estimates. In particular, we investigate how the performance is affected by pixel noise, and by the distance between the camera and the mirrors. We consider a *base* case, in which three fiducial points placed at the corners of a right triangle, with sides measuring $20 \times 20 \times 20\sqrt{2}$ cm, are observed in 200 images. The points are seen via their reflections in two planar mirrors, which are placed at distances of 0.5 m in front of and behind the camera, and rotated by 30 deg in two directions.

In Fig. 4, we plot the errors in the position and attitude estimates of the analytical solution along with those of the MLE. To evaluate the analytical solution's accuracy, we depict the RMS error for the least-accurate axis averaged over 100 Monte-Carlo runs. The MLE accuracy is shown by the standard deviation (1σ) for the least certain axis computed from the covariance estimate.

- As expected, by increasing the pixel noise, the accuracy of the analytical solution, as well as of the MLE decreases.
- As the distance between the camera and the mirrors increases, the accuracy also decreases. We note that the effect is more pronounced than in the single-mirror scenario [10], since in the two-mirror simulation *both* mirrors are moving farther away from the camera, and the effective depth to the scene is increasing at twice the rate.

Note that using the analytical solution as an initial guess for the MLE enables the latter to converge to the correct minimum 100% of the time for non-singular measurement configurations. On average, fewer iterations were required (typically 4) when compared to using a naïve initial guess (typically 18). This shows that the availability of a precise analytical solution improves the speed and robustness of the overall estimation process.

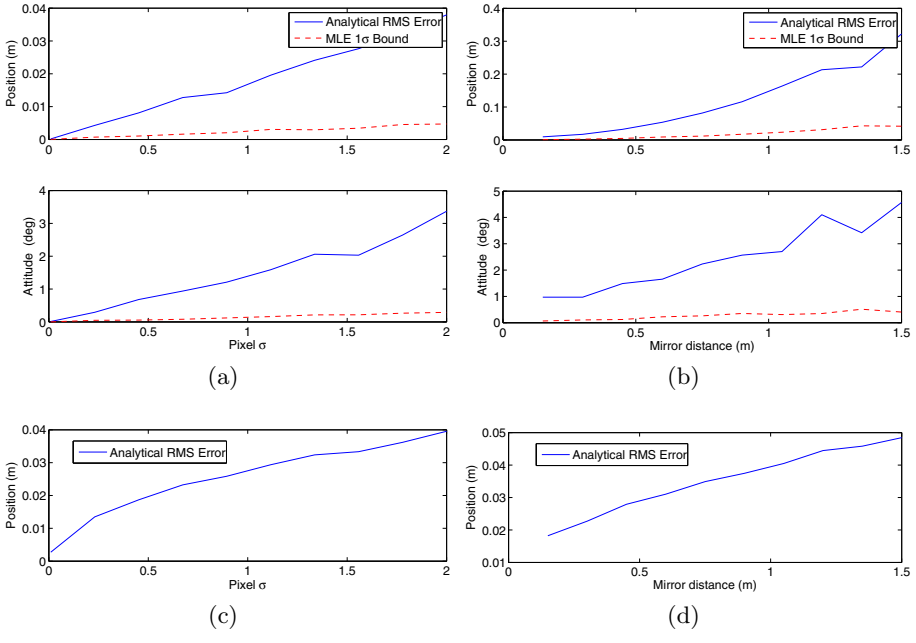


Fig. 4. Two-mirror case: Analytical solution and MLE accuracy for attitude and position plotted versus: (a) pixel noise and (b) mirror distance. Plots (c) and (d) depict the average reconstruction error for the least-accurately reconstructed point versus pixel noise and distance.

In order to evaluate the accuracy of the 3D reconstruction, we randomly populated the simulation environment with 60 points placed near the fiducial markers. The average RMS error (over 100 simulations), for the least-accurately reconstructed point, is plotted versus pixel noise and mirror distance [see Fig. 4(c) and (d)]. Note that even for large distances, or pixel disturbances, the analytical reconstruction error is 5 cm or less.

8 Experiments

The proposed method was also evaluated in real-world experiments to assess its performance and effectiveness in practice. In particular, we consider the case of two mirrors with a camera-equipped mobile robot to compute the transformation between the camera frame of reference and the robot-body frame (see Fig. 1). Frame $\{B\}$ is right-handed with its x-axis pointing towards the front of the robot and its z-axis pointing upwards. The rear-right fiducial marker coincides with the origin of $\{B\}$, while the other two markers lie on its x- and y-axis, respectively. Due to the relative placement of the camera and the fiducials, they cannot be observed directly by the camera nor can they be seen in the reflection in the

(a) (b)

Fig. 5. (a) Image recorded during experimentation. Two reflections of the robot are visible which provide different viewpoints of the chassis. (b) The same image with the single mirror and two-mirror reflections highlighted in blue and purple, respectively. The fiducial points are only visible through the two-mirror reflection.

front mirror [see Fig.s 5(a) and 5(b)]. Instead, the markers were only visible via their double reflections, first in the rear mirror, then in the front mirror.

The camera was connected to the robot's on-board computer via Firewire, and recorded 900 gray-scale images at 10 Hz with resolution 1024×768 pixels. During the experiment the markers were tracked using the Kanade-Lucas-Tomasi Feature Tracker (KLT) [13]. The front mirror was moved continuously through the image sequence, while the rear mirror was moved in three configurations (total 300 images per configuration). The analytically computed camera-to-base transformation is ${}^{C}\mathbf{p}_{B} = \begin{bmatrix} 11.26 & -4.48 & -52.48 \end{bmatrix}^{T}$ cm, and ${}^{C}\mathbf{q}_{B} = \begin{bmatrix} -0.5005 & 0.5063 & -0.4931 & 0.4998 \end{bmatrix}^{T}$, which is very close to the manually determined estimate.

We initialized the MLE with the analytically computed quantities (both the camera-to-base transformation and the mirror vectors in all images), and the Levenberg-Marquardt minimization converged after three iterations. The final estimate for translation and orientation was ${}^{C}\mathbf{p}_{B} = \begin{bmatrix} 10.54 & -4.42 & -53.01 \end{bmatrix}^{T}$ cm, and ${}^{C}\mathbf{q}_{B} = \begin{bmatrix} -0.5026 & 0.5040 & -0.4931 & 0.5000 \end{bmatrix}^{T}$, respectively. The corresponding 3σ bounds computed from the diagonal components of the MLE estimated covariance were $\begin{bmatrix} 9.75 & 6.92 & 6.44 \end{bmatrix}$ mm in position and $\begin{bmatrix} 0.445 & 0.684 & 0.356 \end{bmatrix}$ deg in orientation.

9 Conclusions and Future Work

In this paper, we presented a method for point-based extrinsic camera calibration and 3D scene reconstruction in the challenging case when the points of interest lie outside the camera's direct field of view. To address this issue, we utilize one or more moving planar mirrors to extend the area which the camera can view. We do not assume prior knowledge about the mirror size or placement with respect to the camera. Instead, the only information we exploit are the reflections of

fiducial points, whose coordinates are known *a priori*, and reconstruction points, whose coordinates are unknown and must be calculated from the measurements. We introduced an analytical approach to determine the mirror configurations, the camera-to-base transformation, and the base-frame coordinates of the reconstruction points. Subsequently, we refined the analytically computed quantities using an MLE to produce high-accuracy estimates, along with a measure of the uncertainty in each parameter's estimate. We carried out simulation trials to verify the correctness of the proposed algorithm, as well as to evaluate its sensitivity to various system parameters. Furthermore, we validated the real-world performance of our approach, demonstrating its effectiveness and reliability in practical implementations.

In our ongoing work, we are investigating the feasibility of multi-mirror strategies for complete robot-body 3D reconstruction. Furthermore, we plan to extend this method to the case where no fiducial points are available (i.e., none of the points' coordinates are known *a priori*), but are estimated along with the camera-to-base transformation and the mirror configurations.

References

1. Merritt, E.L.: Explicitly three-point resection in space. Photogrammetric Engineering XV, 649–655 (1949)
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applicatlons to image analysis and automated cartography. Communications of the ACM 24, 381–395 (1981)
3. Haralick, R.M., Lee, C.N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. Int. Journal of Computer Vision 13, 331–356 (1994)
4. Gluckman, J., Nayar, S.K.: Planar catadioptric stereo: Geometry and calibration. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Ft. Collins, CO, pp. 22–28 (1999)
5. Jang, G., Kim, S., Kweon, I.: Single camera catadioptic stereo system. In: Proc. of the Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, Beijing, China (2005)
6. Ramsgaard, B.K., Balslev, I., Arnspang, J.: Mirror-based trinocular systems in robot-vision. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Barcelona, Spain, pp. 499–502 (2000)
7. Nayar, S.K.: Sphereo: Determining depth using two specular spheres and a single camera. In: Proc. of the SPIE Conf. on Optics, Illumination, and Image Sensing for Machine Vision, pp. 245–254 (1988)
8. Jang, K.H., Lee, D.H., Jung, S.K.: A moving planar mirror based approach for cultural reconstruction. Computer Animation and Virtual Worlds 15, 415–423 (2004)
9. Kumar, R.K., Ilie, A., Frahm, J.M., Pollefeys, M.: Simple calibration of non-overlapping cameras with a mirror. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Anchorage, AK (2008)

10. Hesch, J.A., Mourikis, A.I., Roumeliotis, S.I.: Mirror-based extrinsic camera calibration. In: Proc. of the Int. Workshop on the Algorithmic Foundations of Robotics, Guanajuato, Mexico, pp. 285–299 (2008)
11. Hesch, J.A., Mourikis, A.I., Roumeliotis, S.I.: Determining the camera to robot-body transformation from planar mirror reflections. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Nice, France, pp. 3865–3871 (2008)
12. Bouguet, J.Y.: Camera calibration toolbox for matlab (2006)
13. Shi, J., Tomasi, C.: Good features to track. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Washington, DC, pp. 593–600 (1994)

Probabilistic Deformable Surface Tracking from Multiple Videos

Cedric Cagniart[1], Edmond Boyer[2], and Slobodan Ilic[1]

[1] Technische Universität München
[2] Grenoble Universités - INRIA Rhône-Alpes
{cagniart,slobodan.ilic}@in.tum.de, edmond.boyer@inrialpes.fr

Abstract. In this paper, we address the problem of tracking the temporal evolution of arbitrary shapes observed in multi-camera setups. This is motivated by the ever growing number of applications that require consistent shape information along temporal sequences. The approach we propose considers a temporal sequence of independently reconstructed surfaces and iteratively deforms a reference mesh to fit these observations. To effectively cope with outlying and missing geometry, we introduce a novel probabilistic mesh deformation framework. Using generic local rigidity priors and accounting for the uncertainty in the data acquisition process, this framework effectively handles missing data, relatively large reconstruction artefacts and multiple objects. Extensive experiments demonstrate the effectiveness and robustness of the method on various 4D datasets.

1 Introduction

Inferring shapes and their temporal evolutions from image data is a central problem in computer vision. Applications range from the visual restitution of live events to their analysis, recognition and even synthesis. The recovery of shapes using multiple images has received considerable attention over the last decade and several approaches can build precise static 3D models from geometric and photometric information, sometimes in real time. However, when applied to temporal sequences of moving objects, they provide temporally inconsistent shape models by treating each frame independently hence ignoring the dynamic nature of the observed event.

Most methods interested in tracking deformable surfaces in multi-camera systems deform a reference template mesh to fit observed geometric cues as well as possible at each time frame. These cues appear in the literature as photo-consistent models, visual hulls, or even silhouette data directly. Recent works suggest that even without considering photometric information, this geometric data is in many cases sufficiently constraining [1,2,3]. It is however subject to background segmentation and reconstruction errors which needs to be handled in the tracking process. Using strong deformation priors, e.g. articulated models, can help increase robustness but does not extend well to more complex scenes involving several objects whose nature is not necessarily known beforehand. As

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 326–339, 2010.

such scenes require more generic and thus weaker deformation models, it becomes necessary to look into the uncertainty of the data acquisition process and to introduce more robust algorithms modelling its errors.

In this paper, we take these uncertainties into account by embedding the shape tracking within a probabilistic framework. In this way, the need for strong priors is relaxed thus allowing for more complex scenes without sacrifying robustness. The approach considers as input a sequence of independently reconstructed surfaces and iteratively deforms a reference mesh to fit them. The problem is cast as a Bayesian maximum-likelihood estimation where the joint probability of the deformation parameters, i.e. motion, and of the observed data is to be maximized. In order to robustly handle the association between the observations and the reference mesh, latent variables are introduced to identify the mesh region each observation is drawn from, while accounting for possible outliers. We iteratively solve for the motion parameters and posterior probabilities of the latent variables using the Expectation-Maximization algorithm [4].

The remainder of this paper is organized as follows : Section 2 gives an overview previous works that deal with surface tracking in multi-camera environments. In Section 3 we detail our contribution. The corresponding results are presented in Section 4. We conclude the paper by discussing the limitations of our approach and the openings for future work.

2 Related Works

Most of the existing literature dealing with surface tracking in multi-camera environments has to do with the marker-less capture of human performances. For the common case where only one actor is captured, most methods use strong prior knowledge on the deformation of the observed object in the form of articulated models. The works by Gall et al. [5,6] use silhouette and appearance information in a particle filtering framework to infer an optimal skeletal pose. Vlasic et al. [1] first optimize for the pose using the visual hull, then refine the shape estimate from the silhouettes. The works by Mundermann, Corraza et al. [3,7] use a variant of the ICP algorithm [8] to fit an articulated model to the visual hull. The more generic framework used by Aguiar et al. [9] relies on the preservation of Laplacian coordinates of a coarse tetrahedral mesh whose deformation is guided by silhouettes and photometric information. Skeletons on one side and the preservation of volume on the other showed to be priors strong enough for these algorithms to neglect the uncertainty in the input data. However, such strong deformation priors are no longer usable when dealing with objects of arbitrary nature.

To track surfaces in less constrained scenes, it is necessary to relax the deformation priors and thus to handle the noise in the input data. Treating the task as the registration of point sets is more generic but most of the non-rigid extensions to the ICP algorithm[8] lack robustness when confronted with outliers because of the determinism in the choice of point assignments. Among the recent approaches addressing the problem in a probabilistic framework, the works by Horaud et al. address articulated tracking [10] and the registration of rigid and articulated point

sets [11], while the *Coherent Point Drift* algorithm by Myronenko et al. [12] treats arbitrary deformations by regularizing the displacement field. These approaches all use the Expectation-Maximization algorithm to iteratively re-evaluate smooth assignments between the model and the data.

The method we present in this paper uses as input 3D data acquired with a multi-camera setup. It can handle complex scenes involving numerous objects of arbitrary nature by using generic surface deformation priors. It also handles the noise inherent to visual data acquisition by modeling the uncertainty in the observation process and by using the Expectation-Maximization algorithm. The following sections detail the algorithm.

3 Method

3.1 Parametrization and Deformation Framework

In the absence of prior knowledge on the nature of the observed surface, it is challenging to use noisy and sometimes incomplete information to infer meaningful measurements of motion and deformation. A possible way of establishing rigidity priors on the surface is to use the first mesh of a sequence as reference, and then to deform it across time to fit the observed data while penalizing locally non-rigid deformations with respect to its reference pose.

The framework presented in our previous work [2] does so by arbitrarily splitting the original geometry in surface elements called patches and by creating a corresponding coarser control structure in which the reference mesh is embedded. The idea is to regularly distribute patches of a maximal fixed geodesic radius on the surface and to associate to each patch P_k a rotation matrix $\mathbf{R_k}$ and the position of it's center of mass $\mathbf{c_k}$. These parameters encode a rigid transformation with respect to the world coordinates and allow for each vertex v whose position in the reference mesh was $\mathbf{x^0}(v)$ to define its new position as predicted by P_k as:

$$\mathbf{x_k}(v) = \mathbf{R_k}(\mathbf{x^0}(v) - \mathbf{c_k^0}) + \mathbf{c_k}. \tag{1}$$

This effectively decouple the parametrization of the deformation from the complexity of the original geometry. The deformed mesh is computed by linearly blending the predictions made by different patches for each vertex as given by Eq. 2. The weighting functions α_k are simply Gaussians of the euclidean distance to the center of mass of P_k and their support is the union of P_k and its neighbouring patches N_i. They are normalised to add up to 1.

$$\mathbf{x}(v) = \sum_k \alpha_k(v)\mathbf{x_k}(v). \tag{2}$$

3.2 Problem Formulation

Given a set of observed 3D points and an estimate of the current pose of the mesh, we are faced with a parameter estimation problem where the log-likelihood of the joint probability distribution of data and model must be maximized:

$$\max_{\Theta} \ \ln \ P(\mathcal{Y}, \Theta), \tag{3}$$

where:

- $\mathcal{Y} = \{y_i\}_{i=1:m}$ is the set of observed 3D points $\{\mathbf{y_i}\}_{i=1:m}$ and their normals.
- $\Theta = \{\mathbf{R_k}, \mathbf{c_k}\}_{k=1:N_p}$ are the parameters encoding the deformation.
- N_p is the number of patches.

We introduce prior knowledge on the range of possible shape deformations in the form of $E_r(\Theta) = -\ln P(\Theta)$. This energy is modelled by a simple term penalizing local non-rigid deformations of the surface with respect to a reference pose. It is directly linked to the patch-based representation and simply tries to enforce the predicted positions $\mathbf{x_k}(v)$ and $\mathbf{x_l}(v)$ of a vertex v by two neighbouring patches P_k and $P_l \in N_k$ to be consistent.

$$E_r(\Theta) = \frac{1}{2} \sum_{P_l} \sum_{P_k \in N_l} \left[\sum_{v \in P_k \cup P_l} (\alpha_k(v) + \alpha_l(v)) \|\mathbf{x_k}(v) - \mathbf{x_l}(v)\|^2 \right]. \tag{4}$$

Eq.3 can be rewritten using the fact that $P(\mathcal{Y}, \Theta) = P(\mathcal{Y}|\Theta)P(\Theta)$ and leads to solving the following optimization problem:

$$\min_{\Theta} E_r(\Theta) - \ln\ P(\mathcal{Y}|\Theta). \tag{5}$$

3.3 Bayesian Model

We approximate the pdf $P(\mathcal{Y}|\Theta)$ with a mixture of distributions parametrized by a common covariance σ^2, where each component corresponds to a patch. This requires to introduce latent variables z_i for each observation $y_i \in \mathcal{Y}$, where $z_i = k$ means that y_i was generated by the mixture component associated with P_k. We also increase the robustness of our model to outliers by introducing a uniform component in the mixture to handle points in the input data that could not be explained by the patches. This uniform component is supported on the scene's bounding box and we index it with $N_p + 1$.

$$P(y_i|\Theta, \sigma) = \sum_{k=1}^{N_p+1} \Pi_k P(y_i|z_i = k, \Theta, \sigma), \tag{6}$$

where the $\Pi_k = p(z_i = k|\Theta, \sigma)$ represent probabilities on the latent variables marginalized over all possible values of y_i. In other words they are prior probabilities on model-data assignments. We define them as constants $p(z_i = k)$ that add up to 1, using the expected proportion of outlier surface in the observations and the ratios of patch surfaces in the reference mesh.

The patch mixture component with index k must encode a distance between the position $\mathbf{y_i}$ and the patch P_k while accounting for the alignment of normals. For computational cost reasons, we model this distance by looking for each patch P_k in its different predicted poses (this means the positions $\{\mathbf{x_l}(v)\}_{l \in \{k\} \cup N_k, v \in P_k}$ and corresponding normals as shown in Fig. 1) for the closest vertex with a

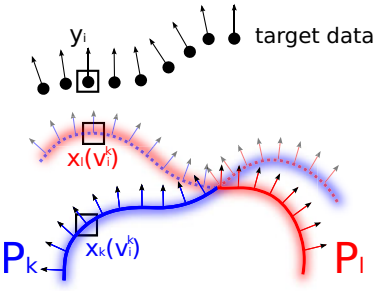


Fig. 1. A point/normal y_i with position $\mathbf{y_i}$ from the observed data is associated to v_i^k, the closest vertex with a compatible normal among all the predictions for the patch P_k. In this case v_i^k is selected because of its position and normal in the prediction made by the neighbouring patch P_l.

compatible normal v_i^k. We consider two points and normals to be compatible when their normals form an angle smaller than a threshold.

$$\forall k \in [1, N_p], \quad P(y_i|z_i = k, \Theta, \sigma) \sim \begin{cases} \mathcal{N}(\mathbf{y_i}|\mathbf{x}(v_i^k), \sigma) & \text{if } v_i^k \text{ exists} \\ \epsilon & \text{otherwise,} \end{cases} \tag{7}$$

where ϵ encodes for a negligible uniform distribution defined on the scene's bounding box.

3.4 Expectation-Maximization

The variables z_i can not be observed but we can use their posterior distributions (Eq. 8) in the EM algorithm first presented by Dempster et al. [4].

$$P(z_i = k|y_i, \Theta, \sigma) = \frac{\Pi_k P(y_i|z_i = k, \Theta, \sigma)}{\sum_{l=1}^{N_p+1} \Pi_l P(y_i|z_i = l, \Theta, \sigma)}. \tag{8}$$

The idea is to replace $P(\mathcal{Y}|\Theta, \sigma)$ with the marginalization over the hidden variables of the joint probability.

$$\ln P(\mathcal{Y}|\Theta, \sigma) = \ln \sum_Z q(Z) \frac{P(\mathcal{Y}, Z|\Theta, \sigma)}{q(Z)}, \tag{9}$$

where $q(Z)$ is a positive real valued function who sums up to 1. The concavity of the log function allows to write a bound on the function of interest:

$$-\ln P(\mathcal{Y}|\Theta, \sigma) \leq -\sum_Z q(Z) \ln \frac{P(\mathcal{Y}, Z|\Theta, \sigma)}{q(Z)}. \tag{10}$$

It can be shown that given a current estimate (Θ^t, σ^t), it is is optimal to choose $q(Z) = P(Z|Y, \Theta^t, \sigma^t)$ in that the bounding function then touches the bounded

function at (Θ^t, σ^t). This means that the bounding function should be the expected complete-data log-likelihood conditioned by the observed data:

$$-\ln P(\mathcal{Y}|\Theta,\sigma) \leq const - E_Z[\ln P(\mathcal{Y}, Z|\Theta,\sigma)|Y]. \tag{11}$$

We rewrite $P(\mathcal{Y}, Z|\Theta,\sigma)$ by making the approximation that the observation process draws the y_i's in $\mathcal{Y}$ from the distribution in an independent identically distributed way:

$$P(\mathcal{Y}, Z|\Theta,\sigma) = \prod_{i=1}^{m} P(y_i, z_i|\Theta,\sigma) \tag{12}$$

$$= \prod_{k=1}^{N_p+1} \prod_{i=1}^{m} \left[P(y_i, z_i = k|\Theta,\sigma)\right]^{\delta_k(z_i)}. \tag{13}$$

The choice made for $q(z)$ then allows to write:

$$-\ln P(\mathcal{Y}|\Theta,\sigma) \leq const - \sum_{k=1}^{N_p+1} \sum_{i=1}^{m} P(z_i = k|y_i, \Theta^t, \sigma^t) \ln P(y_i|z_i = k, \Theta, \sigma). \tag{14}$$

We use the Expectation-Maximization algorithm to iteratively re-evaluate the (Θ, σ) and the posterior probability distributions on the latent variables $\{z_i\}$.

In the E - Step step the posterior $P(z_i|y_i, \Theta^t, \sigma^t)$ functions are evaluated using the current estimation Θ^t, σ^t and the corresponding predicted local deformations of the mesh. They represent weights in the soft assignments of the data to the model. The process amounts to the computation of a $m \times (N_p + 1)$ matrix whose lines add up to 1. This is an extremely parallel operation as all the elements of this matrix can be evaluated independently, except for the normalization step that has to be done by line.

The M - Step requires to minimize the bounding function obtained by evaluating the data-model assignment weights in the E-Step:

$$\Theta^{t+1}, \sigma^{t+1} = \text{argmin} \left[const + E_r(\Theta) - \sum_{k=1}^{N_p+1} \sum_{i=1}^{m} P(z_i = k|y_i, \Theta^t, \sigma^t) \ln P(y_i|z_i = k, \Theta, \sigma) \right] \tag{15}$$

In this bounding function, both data terms and rigidity terms are squared distances between 3D points. Instead of completely minimizing the bounding function, we just run one iteration of the Gauss-Newton algorithm, which amounts to minimizing the quadratic approximation of the objective function around Θ^t.

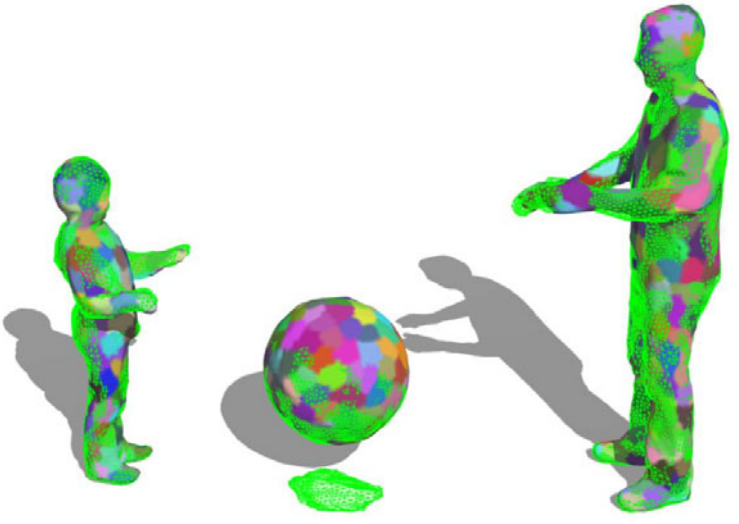


Fig. 2. Ball Sequence involving multiple objects. Note that the wrong geometry on the floor, coming from the shadows, does not affect the tracking results. It is classified as outlier by EM and the ball is not attracted to it.

4 Results

4.1 Multi-object Tracking and Outlier Rejection

The algorithm presented in more generic than the available state of the art methods and allows to track surfaces in complex scenes. We show our results on two of these sequences to demonstrate the clear advantages of our approach. We also provide timing estimates in Table 1 to give a rough idea of its computational complexity.

Ball Sequence. The first of these sequence is the *ball* dataset from INRIA-Perception. It consists of 275 photo-consistent meshes. It involves three distinct object and can not be treated with articulated models. The significant overlap in the silhouettes makes it necessary to run a 3D reconstruction and use point clouds as input data to reduce ambiguity. In Figure 2 we show a particularly difficult frame in which the wrong segmentation of shadows in the original images has resulted in the creation of outlying geometry. The data term presented in [2] does not account for this possibility and simply tries to minimize the distance between two point clouds. Our approach in contrast handles the outlying geometry by progressively reducing its weight in the function optimized by the M-Step.

BasketBall Sequence. We recorded the Basketball sequence in our own multi-camera studio. It is 1364 frames (about 55sec) long and consists of meshes independently reconstructed by a voxel-carving method. It displays a basketball

player dribbling a ball. The interactions between the two objects are fast and complex as the ball bounces between the legs and is sometimes held close to the body for many frames. The results presented in Figure 3 and the accompanying video show two things : firstly, our algorithm can recover these difficult motions and deformations. Secondly, it can cope with the numerous artefacts in the input data : missing limbs, occlusions and self intersecting geometry.

4.2 Human Performance Capture

We also ran our algorithm on standard datasets available to the community to compare it to previous works. We used as input the results of a precise 3D reconstruction algorithm in one case, and noisy voxel carving in the other. As we show in this section, our algorithm performs consistently well in both these situations.

Tracking Using Photo-consistent Meshes As Input. The Surfcap Data from University of Surrey consists of a series of temporally inconsistent meshes obtained by the photo-consistency driven graph-cut method of Starck et al.[13]. Except for some rare reconstruction artefacts in a couple of frames, these are overall very clean and smooth meshes. Because of their extremely high resolution, these meshes were down-sampled to roughly 10k vertices and fed to our algorithm. We present in this paper and the associated video our results on six sequences. They show a hip-hop dancer whose moves are very challenging to track because they contain fast motions and large deformations. In Figure 4, our results on the *Flashkick* dataset show that we can cope with extremely fast deformations such as a backflip. In Figure 5 we present our results on the *Pop* sequence in which the intricate and ambiguous motion of crossing arms is handled properly. Additionally Figure 7 shows a quantitative evaluation of the overlap error between the reprojected silhouettes from our result and the original silhouettes. The error is given as the ratio of erroneous pixels and total number of pixels in the original silhouette. In the presented results we performed an additional optimization that minimizes this reprojection error and keeps it approximately at a constant value of 5%.

Tracking Using Voxel Carving As Input. We used the multi-view image data made public by the MIT CSAIL group to run a very simple voxel carving algorithm. The resulting visual hulls, although only a coarse approximation of the true shape, were enough to drive the deformation of the provided template mesh through the sequences. We ran our algorithm on four of the available sequences and refined the result using silhouette fitting. We compared the silhouette reprojection error to the meshes obtained by Vlasic et al. in [1] and display our results in Figure 8. We also show our results after silhouette fitting on the *Samba* dataset. In this specific sequence, a woman in a skirt dances. Skirts are difficult to handle for methods deforming a reference mesh as the interpolated surface between the bottom of the skirt and the legs does not exist and has to undergo severe compression and stretching. We show in Figure 6 that our approaches still manages to produce visually convincing results.

Fig. 3. Results on the Basketball Sequence. Note that wrong geometry, missing data and fast motion have a limited impact on our tracking algorithm.

Fig. 4. The Flashkick sequence exhibits very fast motion

Fig. 5. The Pop sequence involves a very ambiguous situation when the arms cross

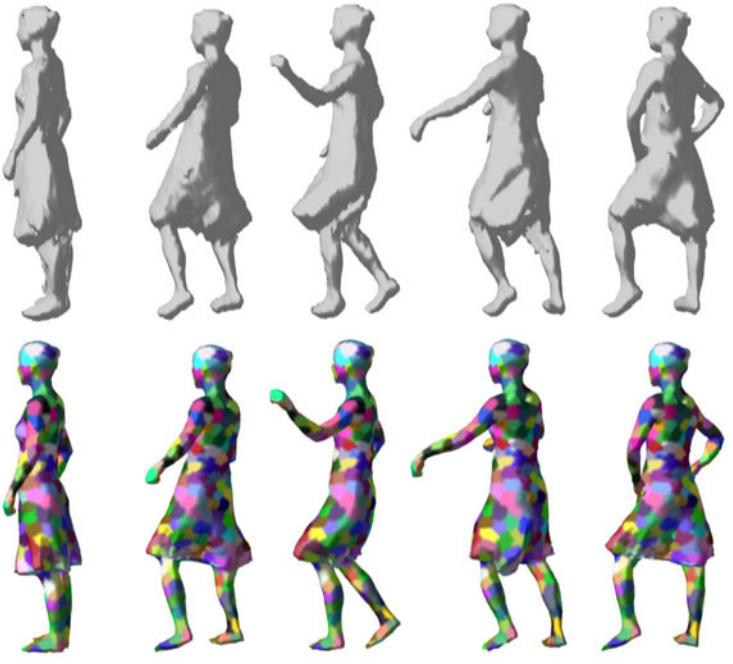
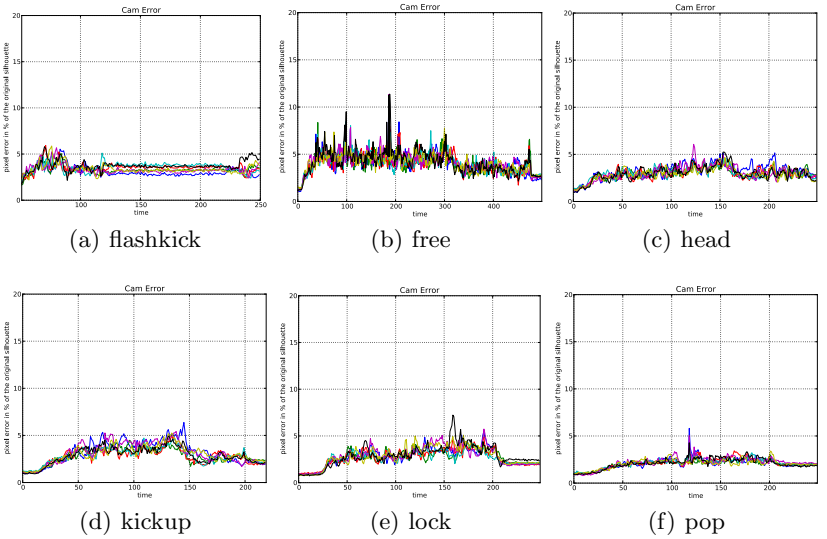


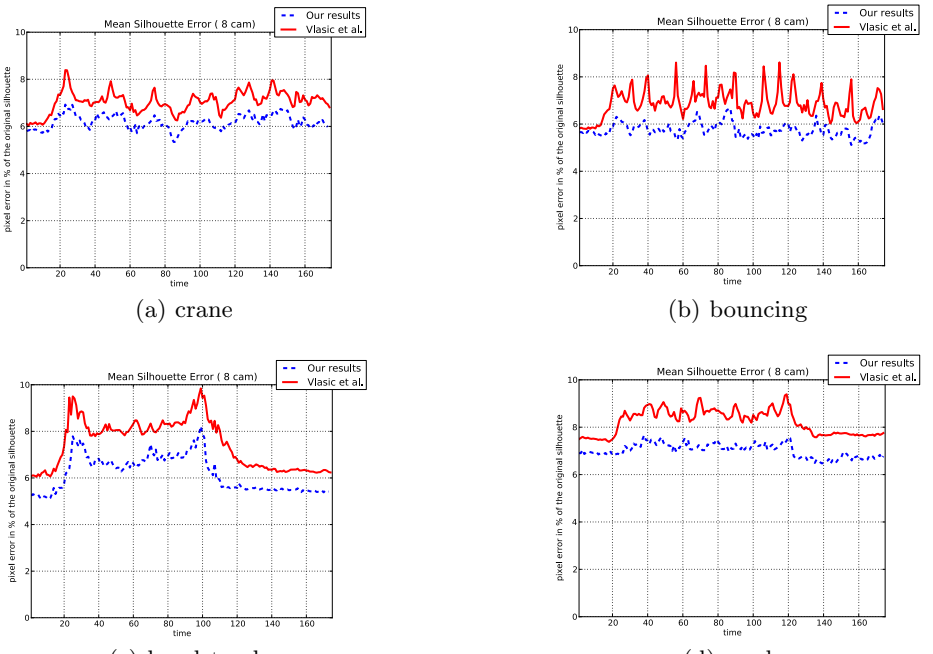
Fig. 6. Results on the Samba sequence show the tracking of a skirt using visual hull reconstructions



(a) flashkick (b) free (c) head

(d) kickup (e) lock (f) pop

Fig. 7. Silhouette reprojection error of our deformed model in percentage of the original silhouette area. Each color represents a camera.



(a) crane (b) bouncing

(c) handstand (d) samba

Fig. 8. Comparison of our numerical results with the method of Vlasic et al.[1]. Although we perform numerically better, it should be noted that their results are temporally smoothed, which can explain the difference in performance.

Table 1. Average timings on standard sequences for the EM procedure (without silhouette refinement), obtained on a 2.5Ghz quad-core machine with target point clouds of roughly 10k vertices. These measurements were obtained by looking at times when files were written to the hard-drive and do not constitute a precise performance evaluation. However they give a rough idea of the computational complexity of our method.

Sequence	Length	Reference Mesh Vertex Count	Average Time Per Frame
Flashkick	200	5445	24 sec
Free	500	4284	25 sec
Head	250	5548	29 sec
Kickup	220	5580	23 sec
Lock	250	5301	24 sec
Pop	250	5596	16 sec
Handstand	174	5939	29 sec
Bouncing	174	3848	29 sec
Crane	174	3407	11 sec
Samba	150	5530	12 sec

5 Discussion

The prediction mechanism for neighbouring patches in the computation of associations described in subsection 3.3 is the key to our method, as it encodes for multiple hypothesis on the position of the patch. More specifically, it gives a chance to the surface to locally quickly return to its rest pose by propagating the information from correctly registered parts of the mesh to parts where the current approximation of the deformation is erroneous.

Topology changes. Although this framework assumes very little on the nature of the tracked objects, it can not handle variations in the topological nature of the reference surface. The reference frame has to be topologically suitable, that is it has to be split wherever the surface might split during the sequence. In other terms, a small amount of geometry disappearance (self-intersection) can be handled, but there can't be any creation of geometry.

The i.i.d. assumption can be considered as problematic in that the observation process is a multi-camera setup in which parts of the surface, thus patches occlude each other. This clearly biases the drawing of samples in the distribution of 3D data. For example in Figure 3, when the arms and body are joined, the local density of points in the input data doesn't double, which clearly indicates that the data generation by two overlapping patches on the arm and the body is not independent. In that sense our method and Equation 12 are only approximations.

6 Conclusion

We proposed a probabilistic method for temporal mesh deformation which can effectively cope with noisy and missing data. We deform a reference mesh and fit it to independently reconstructed geometry obtained from multiple cameras. The imperfection of background segmentation and reconstruction algorithms results in the creation of wrong or missing geometry. Using generic local rigidity priors on the tracked surface, we propose a Bayesian framework which takes into account uncertainties of the acquisition process. We perform a maximum-likelihood estimation where the joint probability of the deformation parameters and the observed data is maximized using the Expectation-Maximization algorithm. We showed on a large number of multi-view sequences that our method is robust to reconstruction artefacts and numerically as precise as state of the art methods based on skeletal priors. Moreover, this effectiveness is achieved with a much more generic deformation model that allows to process complex sequences involving several objects of unknown nature.

References

1. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. In: ACM SIGGRAPH 2008 (2008)
2. Cagniart, C., Boyer, E., Ilic, S.: Free-from mesh tracking: a patch-based approach. In: IEEE CVPR (2010)

3. Mundermann, L., Corazza, S., Andriacchi, T.P.: Accurately measuring human movement using articulated icp with soft-joint constraints and a repository of articulated models. In: CVPR (2007)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, series B (1977)
5. Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.P.: Motion capture using joint skeleton tracking and surface estimation. In: IEEE CVPR 2009 (2009)
6. Gall, J., Rosenhahn, B., Brox, T., Seidel, H.P.: Optimization and filtering for human motion capture. IJCV 87 (2010)
7. Corazza, S., Mündermann, L., Gambaretto, E., Ferrigno, G., Andriacchi, T.P.: Markerless motion capture through visual hull, articulated icp and subject specific model generation. IJCV 87 (2010)
8. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE PAMI 14 (1992)
9. de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.P., Thrun, S.: Performance capture from sparse multi-view video. In: ACM SIGGRAPH 2008 (2008)
10. Horaud, R.P., Niskanen, M., Dewaele, G., Boyer, E.: Human motion tracking by registering an articulated surface to 3-d points and normals. IEEE PAMI 31 (2009)
11. Horaud, R.P., Forbes, F., Yguel, M., Dewaele, G., Zhang, J.: Rigid and articulated point registration with expectation conditional maximization. IEEE PAMI (2010)
12. Myronenko, A., Song, X.: Point-set registration: Coherent point drift. IEEE PAMI (2010)
13. Starck, J., Hilton, A.: Surface capture for performance based animation. In: IEEE Computer Graphics and Applications 27(3) (2007)

Theory of Optimal View Interpolation with Depth Inaccuracy

Keita Takahashi*

The University of Tokyo, IRT Research Initiative, Hongo 7-3-1,
Bunkyo-ku, 113-8656 Tokyo, Japan
keita.takahashi@ieee.org

Abstract. Depth inaccuracy greatly affects the quality of free-viewpoint image synthesis. A theoretical framework for a simplified view interpolation setup to quantitatively analyze the effect of depth inaccuracy and provide a principled optimization scheme based on the mean squared error metric is proposed. The theory clarifies that if the probabilistic distribution of disparity errors is available, optimal view interpolation that outperforms conventional linear interpolation can be achieved. It is also revealed that under specific conditions, the optimal interpolation converges to linear interpolation. Furthermore, appropriate band-limitation combined with linear interpolation is also discussed, leading to an easy algorithm that achieves near-optimal quality. Experimental results using real scenes are also presented to confirm this theory.

1 Introduction

Free-viewpoint image synthesis is the process of combining a set of multi-view images to generate an image which can be seen from a new viewpoint where no camera was actually located [11,16,6]. The quality of free-viewpoint images is greatly affected by the inaccuracies of camera calibration, geometry estimation, and other precedent procedures. However, to our knowledge, there are few studies that rigorously analyze the quantitative quality of resulting images in the presence of such inaccuracies.

This paper presents a theoretical framework for the free-viewpoint image synthesis problem to quantitatively analyze the effect of depth inaccuracy and provide a principled optimization scheme based on the mean squared error (MSE) metric. For simplicity of analysis, the scope of discussion is limited to a fundamental view interpolation setup where the new viewpoint is located between two given input views. I first formulate the relation between the accuracy of depth and the resulting quality of view interpolation. Based on the formulation, I then derive an optimal view interpolation scheme that minimizes the mean squared error (MSE) of the synthesized image, and discuss an appropriate band-limitation combined with conventional linear interpolation. Furthermore, I reveal that the

* This research was supported by National Institute of Information and Communication Technology (NICT).

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 340–353, 2010.

optimal interpolation converges to linear interpolation under specific conditions. Experimental results using real scenes are presented to validate this theory.

A key finding of this work is that the optimal interpolation which outperforms conventional linear interpolation can be achieved if the probabilistic distribution of disparity errors is available. This work also gives a theoretical base for the use of linear interpolation as an approximation of the optimal interpolation, although in previous works linear interpolation was used heuristically without sufficient theoretical justification. In addition, the band-limitation scheme presented in this work achieves near-optimal quality just combined with linear interpolation.

2 Background

Depth inaccuracy is one of the main factors that degrades the quality of free-viewpoint image synthesis and view interpolation. It results from not only the fundamental limitation of geometry estimation methods but also practical reasons such as computational time and the rendering algorithm. For example, depth information is quantized into discrete values in layer-based rendering methods [10,14,12]. In communication systems, depth inaccuracy is induced by lossy data compression.

The tradeoff between depth inaccuracy and camera interval has been studied from the perspective of the sampling problem, resulting in the limiting condition for aliasing-free view interpolation referred to as minimum sampling curve [2,15,8]. If necessary, anti-alias filtering can be used to enforce the limiting condition. However, in practice, the quality of the interpolated view tends to degrade *continuously* as the depth inaccuracy increases, which cannot be captured using the yes-no-question of with/without aliasing artifacts. Furthermore, the anti-alias filter often worsens the result because it filters out some of the valid signal components with the aliasing component [13]. By contrast, I adopt the MSE instead of aliasing for the quality metric to overcome these limitations.

In the context of video or multi-view image compression, depth inaccuracy is quantitatively considered for analyzing coding performance [4,9]. This approach uses the Fourier domain analysis to derive the MSE of the inter-image prediction. It was applied to the view interpolation problem [13], but the discussion was limited to the case where the new viewpoint is located at the midpoint between the two given views. Inspired by those works, this research presents a more comprehensive framework for theoretical analysis of the view interpolation problem, resulting in a principled optimization scheme.

3 Theory

Figure 1 illustrates the configuration used throughout this paper. Let $v_L(x, y)$ and $v_R(x, y)$ be the input images located on the left and right in parallel, respectively. $v(x, y)$ is the target image I want to synthesize from these images by view interpolation. The target viewpoint internally divides the baseline connecting the left and right viewpoints into the ratio $r : 1 - r$, where $r \in [0, 1]$. Let

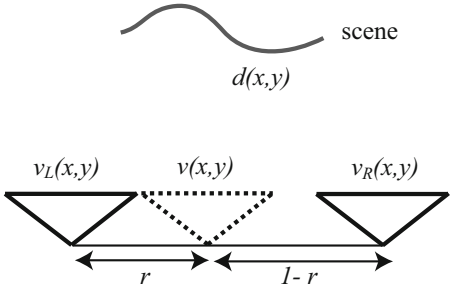


Fig. 1. Configuration

$d(x, y)$ be a pixel-wise depth map where (x, y) denotes pixel positions on the target image and the depth values are represented in terms of disparities (pixel shifts) between the left and right images. I assume that the depth information is provided with certain amount of errors, and analyze the effect of depth errors on the result of view interpolation. Depth estimation itself, i.e., how to estimate $d(x, y)$, is out of scope of this paper.

3.1 View Interpolation with Disparity Error

First, I define a model that describes the relation between the input and target images. Assume that the target scene is a diffusive surface without occlusions. Basically, the left and right images, $v_L(x, y)$ and $v_R(x, y)$, are associated with the target image $v(x, y)$ using the depth map $d(x, y)$. In addition, noise terms, denoted as $n_L(x, y)$ and $n_R(x, y)$, are introduced to compensate the incompleteness of the above assumption. Consequently, the model is described as:

$$\begin{aligned} v_L(x, y) &= v\left(x - rd(x, y), y\right) + n_L(x, y), \\ v_R(x, y) &= v\left(x + (1-r)d(x, y), y\right) + n_R(x, y). \end{aligned} \tag{1}$$

The noise terms include occlusions, non-diffusive reflections, and other components that are present in $v_L(x, y)$ or $v_R(x, y)$ but unpredictable from $v(x, y)$. They are referred to as *unpredictable noise components* in this paper.

The view interpolation process consists of two steps; disparity shifting is applied to the input images, and these images are combined linearly to produce the target image. These steps should be formulated by taking the presence of the depth inaccuracy into account. Let ξ be the disparity error considered as a probabilistic variable, and apply disparity shifting of $(d(x, y)+\xi)$ pixels to the input images to obtain $v'_L(x, y, \xi)$ and $v'_R(x, y, \xi)$ as follows:

$$\begin{aligned} v'_L(x, y, \xi) &= v_L\left(x + r\left(d(x, y)+\xi\right), y\right) &&= v\left(x + r\xi, y\right) + n'_L(x, y) \\ v'_R(x, y, \xi) &= v_R\left(x - (1-r)\left(d(x, y)+\xi\right), y\right) &&= v\left(x - (1-r)\xi, y\right) + n'_R(x, y) \end{aligned} \tag{2}$$

where $n'_L(x, y)$ and $n'_R(x, y)$ are disparity-shifted versions of $n_L(x, y)$ and $n_R(x, y)$, respectively. If $\xi = 0$, $v'_L(x, y, \xi)$ and $v'_R(x, y, \xi)$ are equal to $v(x, y)$, except the

unpredictable noise components. By combining these disparity-shifted images, the result of view interpolation is obtained:

$$\hat{v}(x,y,\xi) = f_L(x,y) \circ v'_L(x,y,\xi) + f_R(x,y) \circ v'_R(x,y,\xi), \tag{3}$$

where $\circ$ denotes convolution, and $f_L(x,y)$ and $f_R(x,y)$ are spatially invariant linear filters, which are referred to as *combining filters* in this paper. It should be noted that *the combining filters integrate the functions of weighting coefficients and prefilters*. This unified formalization leads to a comprehensive optimization scheme. The synthesized image is written as $\hat{v}(x,y,\xi)$ because it is an estimate of the true target image $v(x,y)$ with the disparity error ξ.

In many previous works [3,7,5,8,12][1], linear interpolation with respect to the distance from the target viewpoint

$$\hat{v}(x,y,\xi) = (1-r) \cdot v'_L(x,y,\xi) + r \cdot v'_R(x,y,\xi) \tag{4}$$

was adopted without sufficient theoretical justification. This interpolation is straightforward and often produces reasonable results, however, it is just a special case of (3) with

$$f_L(x,y) = (1-r) \cdot \delta(x,y), \;\; f_R(x,y) = r \cdot \delta(x,y) \tag{5}$$

where $\delta(x,y)$ is Kronecker's delta function. Here, $f_L(x,y)$ and $f_R(x,y)$ degenerate into weighting coefficients without the function of prefilters. By contrast, I seek truly optimal forms for $f_L(x,y)$ and $f_R(x,y)$ based on the minimization of the MSE of the resulting image. It is clarified that under specific conditions, the optimal interpolation converges to linear interpolation as a limit.

3.2 MSE of View Interpolation

In this subsection, I derive the MSE of view interpolation by taking the expectation of the estimation error over the probabilistic disparity error ξ.

The theory is constructed in the frequency domain to easily deal with pixel shifts and filtering convolutions. Substitution of (2) into (3) and Fourier transform over (x,y) yields

$$\begin{aligned}\hat{V}(\omega_x,\omega_y,\xi) = \{&F_L(\omega_x,\omega_y)e^{jr\xi\omega_x} + F_R(\omega_x,\omega_y)e^{j(r-1)\xi\omega_x}\}V(\omega_x,\omega_y) \\ &+F_L(\omega_x,\omega_y)N'_L(\omega_x,\omega_y) + F_R(\omega_x,\omega_y)N'_R(\omega_x,\omega_y),\end{aligned} \tag{6}$$

where ω_x and ω_y ($\omega_x, \omega_y \in [-\pi,\pi]$) are the angular frequencies of x and y, and j denotes the imaginary unit. $\hat{V}$, V, F_L, F_R, N'_L, and N'_R are the Fourier transforms of $\hat{v}$, v, f_L, f_R, n'_L, and n'_R, respectively. The estimation error from the ground truth $V(\omega_x,\omega_y)$ is

[1] Although those works consider more general configurations, their interpolation schemes are equivalent with (5) when the configuration is simplified into that in Fig. 1. Another choice is angular penalty [1], which becomes equivalent with (5) when the object is located far from the cameras.

$$E(\omega_x, \omega_y, \xi) = V(\omega_x, \omega_y) - \hat{V}(\omega_x, \omega_y, \xi). \tag{7}$$

The expectation of the squared error is obtained by

$$\Phi(\omega_x, \omega_y) = \int_{-\infty}^{\infty} p(\xi) \, \|E(\omega_x, \omega_y, \xi)\|^2 \, d\xi, \tag{8}$$

where $p(\xi)$ denotes the probability distribution of ξ. I assume $p(\xi)$ is constant over the image to simplify the discussion. Integration of $\Phi(\omega_x, \omega_y)$ over the entire spectra results in the MSE of view interpolation.[2]

$$\text{MSE} = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \Phi(\omega_x, \omega_y) \, d\omega_x d\omega_y. \tag{9}$$

To further investigate the characteristics of $\Phi(\omega_x, \omega_y)$, (6) and (7) are substituted into (8), resulting in

$$\Phi(\omega_x, \omega_y) = G(\omega_x, \omega_y)\|V(\omega_x, \omega_y)\|^2 + \|\bar{N}(\omega_x, \omega_y)\|^2, \tag{10}$$

$$\begin{aligned} G() = 1 &+ \|F_L()\|^2 + \|F_R()\|^2 - F_L()\psi(r\omega_x) - F_L^*()\psi(-r\omega_x) - F_R()\psi((r-1)\omega_x) \\ &- F_R^*()\psi((1-r)\omega_x) + F_L()F_R^*()\psi(\omega_x) + F_L^*()F_R()\psi(-\omega_x), \end{aligned} \tag{11}$$

$$\|\bar{N}()\|^2 = \|F_L()N_L'()\|^2 + \|F_R()N_R'()\|^2. \tag{12}$$

In (11) and (12), (ω_x, ω_y) is abbreviated as (), and * denotes a complex conjugate. ψ is defined as

$$\psi(z) = \int_{-\infty}^{\infty} p(\xi) e^{j\xi z} d\xi. \tag{13}$$

It is assumed that $n_L'(x, y)$ and $n_R'(x, y)$ have no correlation with $v(x, y)$ and with each other, thereby cross spectra between them are ignored.

As shown by (10), $\Phi(\omega_x, \omega_y)$ consists of three terms. $\|V(\omega_x, \omega_y)\|^2$ denotes the power spectral density of the target image, $G(\omega_x, \omega_y)$ serves as a *gain* that characterizes the magnification factor of $\|V(\omega_x, \omega_y)\|^2$ to produce $\Phi(\omega_x, \omega_y)$, and $\|\bar{N}(\omega_x, \omega_y)\|^2$ summarizes the unpredictable noise components as (12). The disparity error, characterized by $p(\xi)$, affects the gain $G(\omega_x, \omega_y)$ via ψ, as can be seen from (11) and (13). The combining filters, $F_L(\omega_x, \omega_y)$ and $F_R(\omega_x, \omega_y)$, affect both $G(\omega_x, \omega_y)$ and $\|\bar{N}(\omega_x, \omega_y)\|^2$ and are the subjects of optimization, which is described next.

3.3 Derivation of Optimal View Interpolation

The essential idea of optimization is to determine such combining filters, $F_L(\omega_x, \omega_y)$ and $F_R(\omega_x, \omega_y)$, that minimize the MSE defined by (9). This is

[2] Parseval's formula proves that (9) is equivalent with the mean of squared errors calculated in the pixel (spatial) domain.

equivalent to minimizing $\Phi(\omega_x, \omega_y)$ for all frequencies. The optima, denoted as $\hat{F}_L(\omega_x, \omega_y)$ and $\hat{F}_R(\omega_x, \omega_y)$, should satisfy

$$\frac{\partial \Phi(\omega_x, \omega_y)}{\partial \hat{F}_L^*(\omega_x, \omega_y)} = 0, \quad \frac{\partial \Phi(\omega_x, \omega_y)}{\partial \hat{F}_R^*(\omega_x, \omega_y)} = 0. \tag{14}$$

Substitution of (10)–(12) into (14) leads to simultaneous equations

$$\begin{pmatrix} 1+\theta_L & \psi(-\omega_x) \\ \psi(\omega_x) & 1+\theta_R \end{pmatrix} \begin{pmatrix} \hat{F}_L(\omega_x, \omega_y) \\ \hat{F}_R(\omega_x, \omega_y) \end{pmatrix} = \begin{pmatrix} \psi(-r\omega_x) \\ \psi((1-r)\omega_x) \end{pmatrix}, \tag{15}$$

$$\text{where} \quad \theta_L = \frac{\|N_L(\omega_x, \omega_y)\|^2}{\|V(\omega_x, \omega_y)\|^2}, \; \theta_R = \frac{\|N_R(\omega_x, \omega_y)\|^2}{\|V(\omega_x, \omega_y)\|^2} \tag{16}$$

whose unique solution is given as

$$\begin{aligned} \hat{F}_L(\omega_x, \omega_y) &= \frac{(1+\theta_R)\psi(-r\omega_x) - \psi(-\omega_x)\psi((1-r)\omega_x)}{(1+\theta_L)(1+\theta_R) - \psi(\omega_x)\psi(-\omega_x)} \\ \hat{F}_R(\omega_x, \omega_y) &= \frac{(1+\theta_L)\psi((1-r)\omega_x) - \psi(\omega_x)\psi(-r\omega_x)}{(1+\theta_L)(1+\theta_R) - \psi(\omega_x)\psi(-\omega_x)} \end{aligned} \tag{17}$$

I refer to $\hat{F}_L(\omega_x, \omega_y)$ and $\hat{F}_R(\omega_x, \omega_y)$ as *the optimal combining filters*, and view interpolation with those filters as *the optimal view interpolation*.

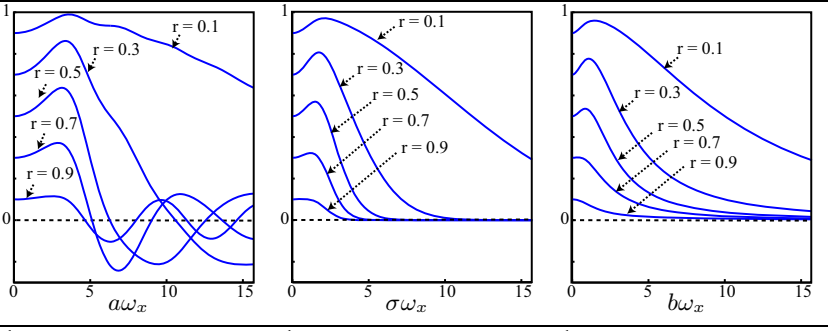
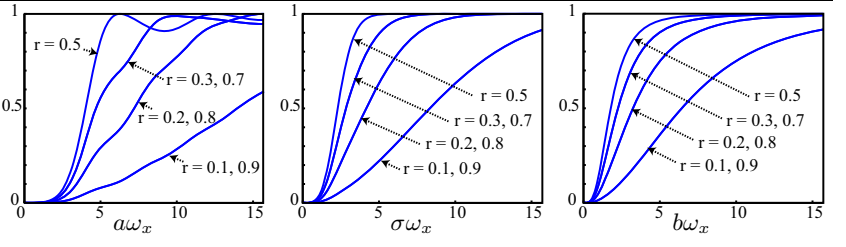
In the remainder, I assume that the unpredictable noise components, $N_L(\omega_x, \omega_y)$ and $N_R(\omega_x, \omega_y)$, are small relative to $V(\omega_x, \omega_y)$, and set $\theta_L = \theta_R = 0$. Because it is difficult to accurately estimate $N_L(\omega_x, \omega_y)$ and $N_R(\omega_x, \omega_y)$, this assumption greatly simplifies further analysis. Note that under this assumption, $\hat{F}_L$, $\hat{F}_R$, and G can be written as functions of ω_x without relation to ω_y.

Several interesting observations on (17) are obtained. First, the optimal combining filters are determined by the disparity errors characterized by $p(\xi)$ and the viewpoint of the target image denoted by r, *without* relation to the spectral shape of the target image $V(\omega_x, \omega_y)$. Second, several boundary conditions that are naturally required by definition are satisfied; $(\hat{F}_L, \hat{F}_R) = (1, 0)$ for $r = 0$, $(\hat{F}_L, \hat{F}_R) = (0, 1)$ for $r = 1$, and $\hat{F}_L = \hat{F}_R$ for $r = 1/2$ can be easily confirmed. Finally, when $\psi(z)$ is even, $\hat{F}_L$ and $\hat{F}_R$ are symmetric with respect to $r = 1/2$, which means that $\hat{F}_L$ for r and $\hat{F}_R$ for $1 - r$ are the same.

3.4 Examples

Let three probability distributions, uniform, Gaussian, and Laplacian, be assumed for $p(\xi)$ as typical models of the disparity error. See Table 1 for a summary. The forms of $p(\xi)$ and $\psi(z)$ are listed in the second and third rows, respectively. In the fourth row, each graph illustrates the frequency response of $\hat{F}_L$ with different values of r. Note that the horizontal axes are scaled by a, σ, or b for the uniform, Gaussian, or Laplacian distributions, respectively. The responses of $\hat{F}_R$ can be obtained by just replacing r with $1 - r$ because F_L and F_R are symmetric. As can be seen from these graphs, the optimal filter emphasizes middle

Table 1. Optimal combining filter ($\hat{F}_L(\omega_x)$) and gain term ($G(\omega_x)$) for disparity errors ($p(\xi)$) following uniform, Gaussian, and Laplacian distributions (a, σ, $b \geq 0$)

	uniform	Gaussian	Laplacian
$p(\xi)$	$\begin{cases} 1/(2a) & (-a \leq \xi \leq a) \\ 0 & (\text{otherwise}) \end{cases}$	$\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\xi^2}{2\sigma^2}\right)$	$\frac{1}{2b} \exp\left(-\frac{\|\xi\|}{b}\right)$
$\psi(z)$	$\frac{\sin(az)}{az}$	$\exp\left(-\frac{(\sigma z)^2}{2}\right)$	$\frac{1}{1+(bz)^2}$
$\hat{F}_L(\omega_x)$			
$G(\omega_x)$			

frequencies and reduces high frequencies. The bottom row shows the gain term $G(\omega_x)$ for each distribution, whose form is obtained by using (11).

As mentioned before, $G(\omega_x)$ is the gain between the original image and the view interpolation error in the frequency domain; the smaller $G(\omega_x)$ means less error. *The optimal combing filters are to minimize $G(\omega_x)$ for all frequencies because they are designed to minimize the MSE.* At least, they should be better than linear interpolation, whose combining filters in the frequency domain are given by $F_L(\omega_x) = 1 - r$ and $F_R(\omega_x) = r$. The comparison between the optimal and linear interpolations for the three distribution models are shown in Fig. 2. It is clear that $G(\omega_x)$ of the optimal interpolation is always smaller than that of linear interpolation.

Another important observation is that the gain of the optimal interpolation never exceeds 1.0, which means that *the optimal interpolation keeps the error below the original signal for all frequencies.* By contrast, linear interpolation amplifies the error in high frequencies with the gain that exceeds 1.0. However, if linear interpolation is combined with appropriate band-limitation, near-optimal

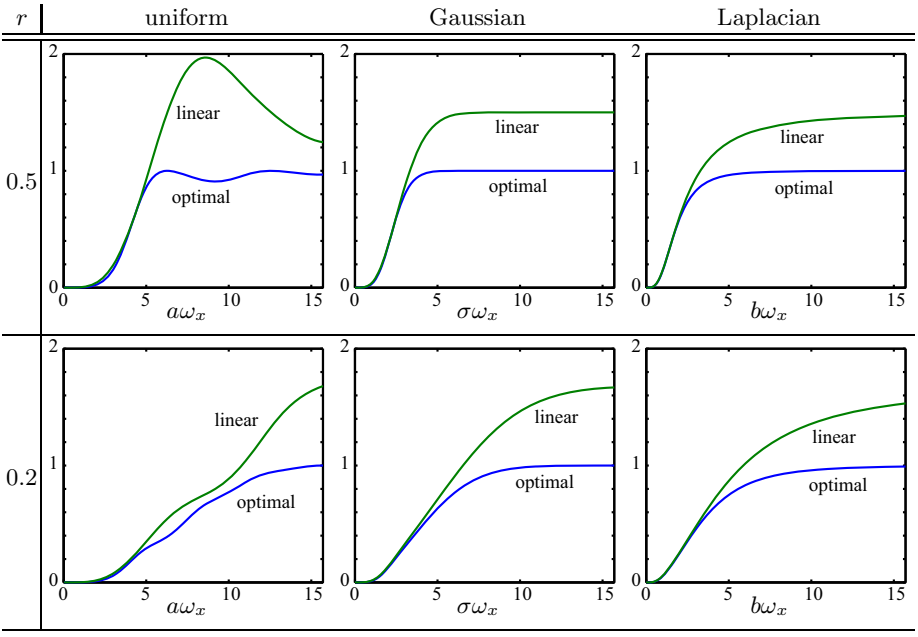


Fig. 2. Comparison of gain term $G(\omega_x)$ between optimal and linear interpolations

interpolation can be achieved. Let $G_{lin}(\omega_x)$ be the gain term of linear interpolation. If the input images are band-limited to the spectral range where $G_{lin}(\omega_x) \leq 1.0$, the resulting gain becomes $\min\{G_{lin}(\omega_x), 1.0\}$, avoiding the overshoots in high frequencies. This method, referred to as *band-limited linear interpolation*, has a practical advantage in that the implementation is much easier than the naive implementation of the optimal interpolation. The spectral components where $G_{lin}(\omega_x) \leq 1.0$ are kept without any change and other components are just reduced to 0. The cut-off frequency of this method is derived from the shape of the gain term $G(\omega_x)$, while the band-limitation in [2,8] was based on the anti-alias condition.

3.5 Relation with Linear Interpolation

Finally, I investigate the theoretical relation between the optimal and linear interpolations. I give a theoretical basis to linear interpolation used just *heuristically* without sufficient justification.

I first discuss the three distribution models mentioned above. As seen from the third row of Table 1, $\psi(z)$ can be written as a function of kz, where k takes a, σ, or b, respectively. k represent the magnitude of the disparity error; a larger k corresponds to a larger error. For all of those distributions, $\psi(z)$ can be expanded around $kz = 0$ as:

$$\psi(z) = 1 - A \cdot (kz)^2 + O\left((kz)^4\right), \tag{18}$$

where A takes 1/6, 1/2, and 1 for the uniform, Gaussian, and Laplacian distributions, respectively. Substituting (18) into (17),

$$\lim_{k\omega_x \to 0} \hat{F}_L(\omega_x, \omega_y) = 1 - r, \quad \lim_{k\omega_x \to 0} \hat{F}_R(\omega_x, \omega_y) = r, \tag{19}$$

can be proven. This result can be confirmed from the fourth row of Table 1; $\hat{F}_L$ reaches $1-r$ as $k\omega_x$ reaches 0 for all distribution models. The physical meaning is straightforward. *For small disparity errors ($k \sim 0$) or low frequencies ($\omega_x \sim 0$), the optimal interpolation converges to conventional linear interpolation.* Equation (19) also holds true when $p(\xi)$ is represented as a weighted sum of those three distributions.

As shown by (8)–(10), the final MSE depends on the spectral shape of the target image $V(\omega_x, \omega_y)$. It should be noted that $V(\omega_x, \omega_y)$ of a natural image tends to be low-frequency oriented (most of the image energy resides in low frequencies). In addition, as seen in Fig. 2, the difference in $G(\omega_x)$ between the optimal and linear interpolations is marginal for small $k\omega_x$. Consequently, the difference in MSE between the optimal and linear interpolations would be quite small for practically small values of k, which would justify the use of linear interpolation as an approximation of the optimal interpolation.

The discussion above can be partly extended to general distributions. Equation (13) can be rewritten as

$$\psi(z) = \int_{-\infty}^{\infty} p(\xi) \sum_{n=0}^{\infty} \frac{(j\xi z)^n}{n!} d\xi = \sum_{n=0}^{\infty} \left\{ \int_{-\infty}^{\infty} p(\xi) \xi^n d\xi \right\} \frac{(jz)^n}{n!} = \sum_{n=0}^{\infty} \mu_{\xi,n} \frac{(jz)^n}{n!}, \tag{20}$$

where $\mu_{\xi,n}$ is the n-th order moment of ξ. $\mu_{\xi,0} = 1$ by definition of the probability distribution $p(\xi)$. $\mu_{\xi,1}$ is the mean of ξ, and let it be 0. $\mu_{\xi,2}$ is the variance and described as σ_ξ^2. If all of the higher order moments are convergent, Equation (20) can be approximated around $z = 0$ as

$$\psi(z) \sim 1 - \sigma_\xi^2 z^2/2 \tag{21}$$

Substitution of (21) into (17) leads to

$$\lim_{\omega_x \to 0} \hat{F}_L(\omega_x, \omega_y) = 1 - r, \quad \lim_{\omega_x \to 0} \hat{F}_R(\omega_x, \omega_y) = r. \tag{22}$$

Unlike (19), (22) gives no information about k. This is reasonable because no explicit form is assumed for $p(\xi)$, thereby there is no parameter such as k that characterizes $p(\xi)$. However, also for this case, the optimal interpolation converges to linear interpolation as ω_x reaches 0.

4 Experiment

Figure 3 illustrates the flow of the experiment (See Fig. 1 for the configuration). A complete MATLAB implementation is included in the supplementary material.

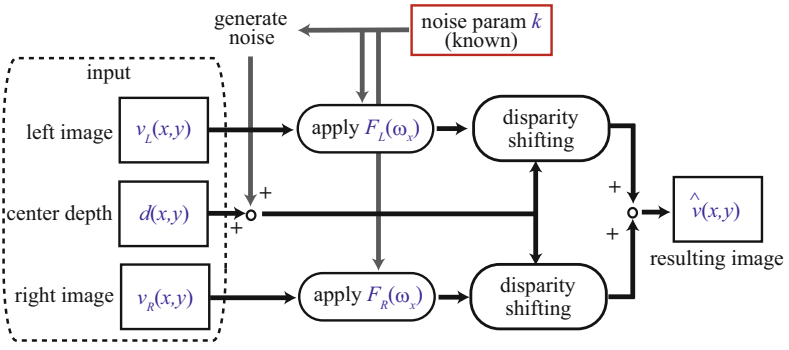


Fig. 3. Flow of experiment. Artificial noise is added to disparity map to simulate depth inaccuracy. Its parameter is used to design combining filters, F_L and F_R.

The input is a pair of stereo images, denoted as $v_L(x, y)$ and $v_R(x, y)$, respectively, and a pixel-wise depth map from the target viewpoint, denoted as $d(x, y)$. The pixel value of $d(x, y)$ represents the disparity (pixel shift) that is measured between the viewpoints of $v_L(x, y)$ and $v_R(x, y)$.

First, artificial noise is added to $d(x, y)$ to simulate the disparity error, yielding $d'(x, y)$. The disparity error is random, but the distribution $p(\xi)$ is assumed to be known. Next, combining filters are applied to the input images. The optimal filters are derived from (17) using the known distribution of the disparity error, $p(\xi)$. Although this filtering operation is applicable as convolution in the spatial domain, it is implemented in the frequency domain because the purpose of the experiments is to confirm the theory. To be precise, the left image after filtering operation, $\dot{v}_L(x, y)$, is obtained as

$$\dot{v}_L(x,y) = f_L(x,y) \circ v_L(x,y) = \mathcal{F}^{-1}\left[F_L(\omega_x,\omega_y) \cdot \mathcal{F}\left[v_L(x,y)\right]\right], \tag{23}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent Fourier transform and its inverse. For linear interpolation, the filtering operation is just multiplying the input images by a weighting coefficient, $1 - r$ or r, and is implemented in the spatial domain. The band-limitation introduced in Section 3.4 is also implemented as an option.

Then, disparity shifting is performed over the filtered left image $\dot{v}_L(x, y)$ using the noisy disparity map $d'(x, y)$, yielding

$$\ddot{v}_L(x,y) = \dot{v}_L\left(x + r \cdot d'(x,y),\ y\right). \tag{24}$$

In this operation, cubic spline interpolation is adopted to read fractional pixel positions. $\ddot{v}_R(x, y)$ is also obtained in the same way. Finally, the resulting image from the target viewpoint is obtained as

$$\hat{v}(x,y) = \ddot{v}_L(x,y) + \ddot{v}_R(x,y). \tag{25}$$

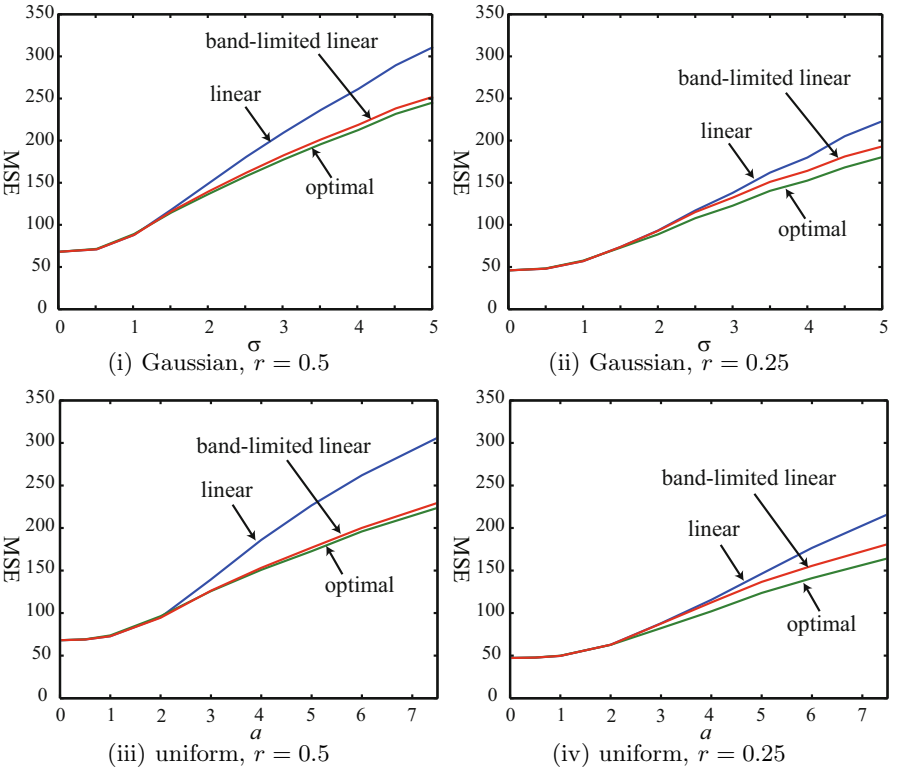


Fig. 4. Profiles of MSE (y axes) against magnitude of disparity error (x axes) obtained with *cones* dataset. Optimal interpolation outperforms linear interpolation as a whole. Band-limited linear interpolation achieves near-optimal quality.

$\hat{v}(x, y)$ is compared with the ground truth $v(x, y)$ to evaluate the accuracy. No occlusion handling was performed because it is out of scope of this paper.[3]

Two real image datasets, *cones* and *teddy*, from Middlebury stereo website[4] are used for experiments. All images are 450×375 pixels and converted into 8-bit grayscale. The target viewpoint is set to the position of *im2*, whose corresponding disparity map *disp*2 is also available. When *im0* and *im4* are used for $v_L(x, y)$ and $v_R(x, y)$, respectively, r equals 0.5. Instead, when *im1* and *im5* are used, r becomes 0.25. In both cases, the accuracy of the original depth information is 1/4 pixel length in terms of disparities between the left and right images.[5] The

[3] The procedure described here seems a little different from that of (2) and (3); the disparity shifting and combing filtering are performed in the opposite order. The order is changed to avoid hole-filling problems. These two procedures are completely equivalent in the theoretical model.

[4] http://vision.middlebury.edu/stereo/

[5] Consequently, inherent quantization errors are included in the input depth data. Those relatively small errors are ignored in the experiment.

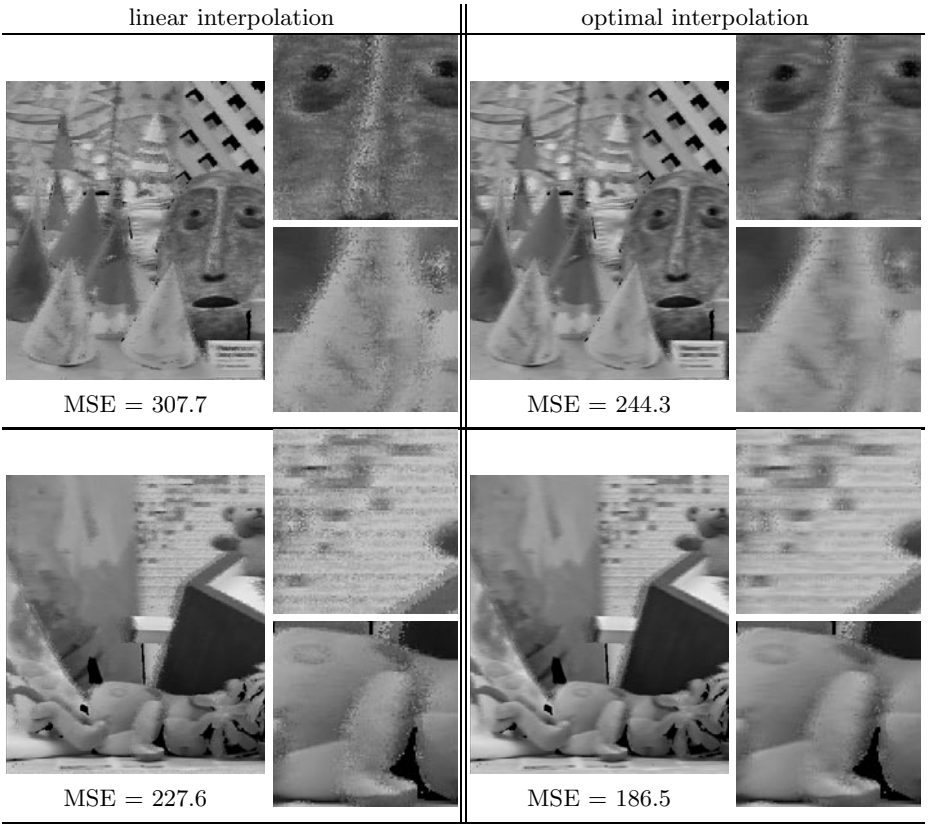


Fig. 5. Resulting images and close-ups by linear (left) and optimal (right) interpolations. Disparity errors are Gaussian of $\sigma = 5$ and viewpoint is set to $r = 0.5$. Optimal interpolation produces more plausible results.

resulting image is compared with the ground truth *im2* for evaluation of MSE, which are calculated in floating point precision without considering the pixels with void disparities and 50 pixels from both sides.

Figure 4 shows the MSE profiles against the magnitude of the disparity error obtained with *cones* dataset. The disparity error is Gaussian for the upper row and uniform for the lower row. It is obvious that the optimal interpolation outperforms linear interpolation as a whole. As the theory states, the difference between the optimal and linear interpolations decreases as the disparity error decreases. Sometimes, linear interpolation outperforms the optimal interpolation by a very narrow margin. However, as the disparity error increases, the advantage of the optimal interpolation becomes significant. Another important observation is that band-limited linear interpolation achieves comparable quality with the optimal interpolation.

Figure 5 shows several resulting images by the linear and optimal interpolations with large disparity errors (Gaussian with $\sigma = 5$). These images are low

quality because of the very noisy disparities used, but the optimal interpolation produces more plausible results than linear interpolation in visual quality. More results are included in the supplementary material.

It should be noted that the proposed theory starts with an occlusion-free diffusive surface model (see (1)), and the noise terms are actually ignored (θ_R and θ_L in (16) are assumed to be 0) in the optimization. However, the optimal interpolation works well for real scenes that contain non-diffusive reflections and occlusions, indicating that the proposed theory successfully captures the essential property of real image data despite its simplicity.

5 Conclusions

A theoretical framework for view interpolation problem to analyze the quantitative quality of the resulting image in the presence of depth inaccuracy and provide a principled optimization scheme based on the MSE metric was proposed. The theory clarified that if the probabilistic distribution of the disparity error is available, the optimal view interpolation that outperforms conventional linear interpolation can be achieved. It was also revealed that the optimal interpolation converges to linear interpolation as the disparity inaccuracy decreases. The theory was confirmed by experiment using real image data.

The main drawback of the optimal interpolation scheme is that it requires the exact shape of the disparity error distribution $p(\xi)$, which may be infeasible in practice. Furthermore, the advantage of optimal interpolation over linear interpolation is marginal unless the disparity inaccuracy is considerably large. Consequently, when the depth information is accurate to some extent, linear interpolation is a realistic choice. If the depth information is very noisy, band-limited linear-interpolation can be adopted, because it can achieve near-optimal quality only with simple band limitation. Although the cut-off frequency also depends on the shape of $p(\xi)$, this parameter might be tuned interactively.

Future work will include several directions. First, the theory will be extended to deal with more general configurations, e.g., 2-D configurations of input cameras and the target viewpoint not located on the baseline. Second, the probabilistic distribution of the disparity error should be studied from real data. For example, most stereo matching algorithms produce depth maps with spatially varying errors, which conflicts with the assumption that $p(\xi)$ is space invariant. In addition, efficient implementations or approximations for the optimal combining filters should be considered.

Finally, it should be noted that view interpolation is essentially a very complex problem. As seen from [17], many technical elements including accurate camera calibration, stable depth/correspondence estimation, and appropriate handling of occlusion boundaries, contribute to the final rendering quality. Meanwhile, the theory presented in this paper is focused only on a single aspect of the problem: *how to blend input image pixels to synthesize new output pixels when the correspondences are established with some amount of errors*, which is a common issue for any view interpolation algorithm. I believe the theory can be extended

to deal with other aspects in the future and it leads to a solid mathematical framework for general view interpolation problems.

References

1. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: ACM SIGGRAPH Papers pp. 425–432 (2001)
2. Chai, J., Tong, X., Chany, S.C., Shum, H.Y.: Plenoptic sampling. In: ACM Trans. Graphics (Proc. ACM SIGGRAPH), pp. 307–318 (2000)
3. Chen, S.E., Williams, L.: View interpolation for image synthesis. In: Proc. ACM SIGGRAPH, pp. 279–288 (1993)
4. Girod, B.: The efficiency of motion-compensating prediction for hybrid coding of video sequences. IEEE Journal SAC SAC 5(7), 1140–1154 (1987)
5. Gortler, S.-J., Crzeszczuk, R., Szeliski, R., Cohen, M.-F.: The lumigraph. In: Proc. ACM SIGGRAPH, pp. 43–54 (1996)
6. Kubota, A., Smolic, A., Magnor, M., Tanimoto, M., Chen, T., Zhang, C.: Special issue on multi-view imaging and 3dtv. IEEE Signal Processing Magazine 24(6), 10–111 (2007)
7. Levoy, M., Hanrahan, P.: Light field rendering. In: Proc. ACM SIGGRAPH, pp. 31–42 (1996)
8. Lin, Z., Shum, H.Y.: A geometric analysis of light field rendering. Intl. Journal of Computer Vision 58(2), 121–138 (2004)
9. Ramanathan, P., Girod, B.: Rate-distortion analysis for light field coding and streaming. EURASIP SP:IC 21(6), 462–475 (2006)
10. Shade, J.W., Gortler, S.J., He, L.W., Szeliski, R.: Layered depth images. In: Proc. ACM SIGGRAPH, pp. 231–242 (1998)
11. Shum, H.Y., Kang, S.B., Chan, S.C.: Survey of Image-Based Representations and Compression Techniques. IEEE Trans. CSVT 13(11), 1020–1037 (2003)
12. Taguchi, Y., Takahashi, K., Naemura, T.: Real-time all-in-focus video-based rendering using a network camera array. In: Proc. 3DTV-Conference, pp. 241–244 (2008)
13. Takahashi, K., Naemura, T.: Theoretical model and optimal prefilter for view interpolation. In: Proc. IEEE ICIP, pp. 1528–1531 (2008)
14. Tong, X., Chai, J., Shum, H.Y.: Layered lumigraph with lod control. The Journal of Visualization and Computer Animation 13(4), 249–261 (2002)
15. Zhang, C., Chen, T.: Spectral analysis for sampling image-based rendering data. IEEE Trans. CSVT 13(11), 1038–1050 (2003)
16. Zhang, C., Chen, T.: A survey on image-based rendering - representation, sampling and compression. EURASIP SP:IC 19(1), 1–28 (2004)
17. Zitnick, C., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.: High quality video interpolation using a layered representation. In: ACM SIGGRAPH Papers, pp. 600–608 (2004)

Practical Methods for Convex Multi-view Reconstruction

Christopher Zach and Marc Pollefeys

ETH Zürich, Universitätstrasse 6, CH-8092 Zürich

Abstract. Globally optimal formulations of geometric computer vision problems comprise an exciting topic in multiple view geometry. These approaches are unaffected by the quality of a provided initial solution, can directly identify outliers in the given data, and provide a better theoretical understanding of geometric vision problems. The disadvantage of these methods are the substantial computational costs, which limit the tractable problem size significantly, and the tendency of reducing a particular geometric problem to one of the standard programs well-understood in convex optimization. We select a view on these geometric vision tasks inspired by recent progress made on other low-level vision problems using very simple (and easy to parallelize) methods. Our view also enables the utilization of geometrically more meaningful cost functions, which cannot be represented by one of the standard optimization problems. We also demonstrate in the numerical experiments, that our proposed method scales better with respect to the problem size than standard optimization codes.

1 Introduction

Globally optimal methods in multiple view geometry and 3D computer vision are appealing tools to estimate geometric entities from visual input. So far most research in this field has been focused on the formulation of geometric vision problems in terms of a standardized optimization taxonomy, e.g. as linear or higher order cone programs. With very few exceptions, generic optimization codes are utilized for the respective numerical experiments. The emphasis on global optimal formulations lies on L_∞-based objective functions, i.e. minimizing the maximum over a set of convex or quasi-convex functions with respect to the unknowns. The initially intriguing L_∞-based objective can be easily converted into a simple cost function combined with a large number of (convex) constraints, which subsequently enables tractable solvers to be applied. The decision of utilizing an L_∞-based objective function has two important consequences: first, it induces particular (and often unrealistic) assumptions on the noise characteristic of the observed measurements; and second, the typically encountered quasi-convex nature of the optimization problem implies, that the solution procedure only indirectly provides the unknown variables of interest (through a sequence of feasibility problems). Hence, more robust and efficient alternative formulations of important tasks in multiple view geometry are desired.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 354–367, 2010.

Further, formulating geometric vision task in terms of general optimization problems has the advantage of having a well-understood theory and mature software implementations available, but such an approach also limits the range of multi-view problems and objective functions to those standard optimization problems. In this work we propose a more direct view on a class of geometric vision problems not taking the route through one of the standard convex programs. Our view on these problems is inspired by recent advances on convex formulations or relaxations of low-level vision problems. Our contribution is two-fold: we demonstrate the applicability of optimization methods primarily utilized in signal and image processing for geometric vision problems, and we extend recent convex models for multi-view reconstruction by a new cost function better approximating the squared reprojection error while still preserving the convexity of the problem.

2 Related Work

2.1 Global Optimization in Multiple View Geometry

Global optimization in multiple view geometry has gained a lot of interest in the recent years. In particular, L_∞-based (or min-max) formulations are popular (i) due to the well-understood relation with fractional programming leading to linear or second order cone programs, and (ii) due to the good accuracy provided by the solution for geometric applications if no outliers are present.

The first exposition of L_∞ minimization for geometric computer vision problems is given in [1], where the authors propose the L_∞ cost function for multi-view reconstruction tasks. The relation between quasi-convex functions and L_∞ optimization for multi-view geometry was independently discovered in [2] and [3]. Quasi-convex functions (i.e. functions with convex sublevel sets) can be effectively minimized by a bisection search for the opimal value, thus solving a sequence of convex feasibility problems. Additional convex constraints can be also provided. These approaches present structure and motion recovery given known camera rotations as the prototypical application.

Sim and Hartley [4] discuss an L_∞ view on the problem of estimating camera centers (again under the assumption of known rotations) given the directions of the baseline between cameras. The 3D scene structure is not explicitly modeled as unknown parameter subject to minimization, therefore the problem size is substantially reduced. Placing camera centers given a set of relative directions is similar to the graph embedding problem for motion recovery [5,6]. In [4] the degeneracy of embedding formulations for linear camera motions is addressed by utilizing the trifocal tensor to incorporate the relative scales of baselines. Removing the 3D structure from the problem formulation reduces the size of the optimization problem substantially, but also leads to minimization of quite abstract cost functions (e.g. the angular deviation between given and estimated baseline directions) instead of image-related quantities like the reprojection error.

The high computational costs of L_∞ optimization has lead to investigations to reduce the respective run-time. [7] describes an interior point algorithm

exploiting the same sparsity pattern in the underlying problem as it is also found in sparse bundle adjustment methods. The observation that the objective value of a min-max problem is only dependent on a (potentially small) subset of error terms can be utilized to formulate faster methods for L_∞ optimization [8]. In this approach only a subset of data points is considered for optimization (thus making the problem smaller), but the residuals are evaluated for all data points. If all residuals are less or equal to the objective value, then the procedure can be stopped; otherwise additional data points with large residuals are added in further minimization steps.

L_∞ optimization has the potential disadvantage of being susceptible to outliers in the input data. It can be shown that some of the inputs attaining the maximum error in min-max optimization are guaranteed to be outliers under suitable assumptions, hence these outliers can be iteratively detected and removed by L_∞ optimization [9]. Alternatively, L_∞ (min-max) objective functions can be replaced by L_1 (min-sum) cost functions, leading directly to formulations much less affected by outliers in the data. Straightfoward L_1-based optimization of geometric vision problems similar to many of the L_∞ approaches outlined above usually leads to sum-of-fractions type of optimization problems, which are extremely difficult to solve. The recent approaches described in [10,11] (and reviewed in more detail in Sec. 3) aim on directly minimizing the number (i.e. L_0-norm) of outliers for a particular inlier criterion represented by suitable (either linear or second order cone) constraints. If all data points are inliers (i.e. the residuals are less than a given threshold), the objectve value is 0. Convexification of the L_0 norm yields an L_1-like objective function and thereby to respective linear or second order cone programs.

We present our method for practical convex optimization in multi-view geometry on the problem of structure and motion recovery given the global camera rotations. This raises the question of how these rotations can be determined. Global rotations can be efficiently computed from pair-wise relative orientations between images by utilizing the consistency relation between relative and absolute camera rotations, $R_j = R_{ij}R_i$. [12] and [13] present different methods to obtain consistent global rotations. For full structure and motion computation, [13] employs the L_∞ framework of [2], but uses only a small subset of scene points in order to accelerate the minimization procedure and to guarantee an outlier-free set of data points.

2.2 Non-smooth Convex Optimization and Proximal Methods

Even a convex optimization problem can be difficult to solve efficiently, especially if the objective is a non-smooth function. A class of methods successfully applied in signal and image processing are based on proximal calculus: for a convex function f and $\gamma > 0$ the mapping

$$\operatorname{prox}_{\gamma f}(\bar{x}) = \arg\min_x \left\{ f(x) + \frac{1}{2\gamma}\|x - \bar{x}\|_2^2 \right\} \tag{1}$$

is called the proximity operator. It generalizes the notion of projection operators (for which f is the hard indicator function for a convex set S). In Eq. 1 the objective itself is called the Moreau envelope of function f of index γ. Sometimes $\text{prox}_{\gamma f}$ is difficult to compute directly, but the proximity operator for the conjugate function, $\text{prox}_{\gamma f^*}$, can be determined efficiently. In these cases we can utilize Moreau's decomposition,

$$x = \text{prox}_{\gamma f}(x) + \gamma \, \text{prox}_{f^*/\gamma}(x/\gamma) \tag{2}$$

We refer to [14] for a recent, compact exposition of proximal calculus and its importance in image processing applications.

Proximal methods, in particular the forward-backward algorithm [14] and Douglas-Rachford splitting [15,16], allow the efficient minimization of structured, convex problems of the form $\min_x f_1(x) + f_2(x)$. In brief, the Douglas-Rachford splitting iterates

$$\begin{aligned} \hat{x}^{(n)} &= \text{prox}_{\gamma f_2}(x^{(n)}) \\ x^{(n+1)} &= x^{(n)} + \text{prox}_{\gamma f_1}(2\hat{x}^{(n)} - x^{(n)}) - \hat{x}^{(n)}, \end{aligned} \tag{3}$$

and $\hat{x}^{(n)}$ is known to converge to the solution of $\min_x f_1(x) + f_2(x)$. See also e.g. [17] for the connections between Douglas-Rachford, augmented Lagrangian and split Bregman methods. In Section 4.3 we apply the Douglas-Rachford splitting on a problem with f_2 being the indicator function of a hyper-plane (i.e. $\text{prox}_{\gamma f_2}$ amounts to project the argument into a linear subspace), and f_1 further decomposing into many independent objectives.

3 Convex L_1 Reconstruction with Known Rotations

In this section we review robust structure and motion computation with known camera rotations based on convex optimization. Since all the camera centers and 3D points are mutually dependent in the objective function, we choose this application in order to demonstrate our method on a larger scale problem. Other classical problems addressed by global optimization in multi-view geometry are optimal point triangulation and camera resectioning, which involve a much smaller number of unknowns. In the following we assume that global camera rotations are given (e.g. a set of pairwise relative rotations can be upgraded to consistent global ones via eigenvalue decomposition [13]). Let i denote the index of the cameras and j be the index of 3D points, respectively. The set of global rotations R_i for each camera is assumed to be known. Further, let $u_{ij} = (u_{ij}^1, u_{ij}^2, 1)^T$ be the projection of the (unknown) 3D point X_j into image i, i.e. $u_{ij} \propto R_i X_j + T_i$, where T_i and X_j are the translation vectors and 3D point positions yet to be determined. We assume that the image coordinates u_{ij} are normalized, i.e. premultiplied by the inverse camera intrinsics. With known rotations, the relationship between 3D structure, camera poses and image measurements are essentially linear (up to the unknown depth). The full projection function

$$\hat{u}_{ij} = \frac{R_i X_j + T_i}{(R_i X_j + T_i)_3}$$

is nonlinear, but e.g. the squared reprojection error is quasi-convex and amenable for L_∞ optimization (e.g. [2]). We focus on the L_1 setting where a minimizer of the sum of some deviation is sought. The intention is to increase robustness in presence of gross outliers by utilizing an L_1 objective function. [10,11] use a quasi L_∞ model by assigning zero cost, whenever the projection of a 3D point X_j lies within a neighborhood of a user-specified radius σ (where the neighborhood is induced either by the Euclidean norm [11] or by the maximum norm [10]. Consequently, no cost is attributed in the objective function, if X_j lies within a (generalized) cone induced by the camera center $C_i = -R_i^T T_i$ and the observed image point u_{ij} (see Figure 1).

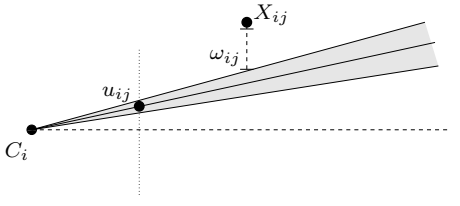


Fig. 1. The cone induced by the camera center C_i and the observed image point u_{ij}. Points X_{ij} residing within the shaded cone are considered as inliers and have no cost in the objective function, whereas outliers are penalized.

Denoting $X_{ij} = (X_{ij}^1, X_{ij}^2, X_{ij}^3)^T = R_i X_j + T_i$, the condition of X_j being in the respective cone with radius σ reads as

$$\|u_{ij} - X_{ij}^{1,2}/X_{ij}^3\|_p \leq \sigma$$

or equivalently,

$$\|u_{ij} X_{ij}^3 - X_{ij}^{1,2}\|_p \leq \sigma X_{ij}^3, \tag{4}$$

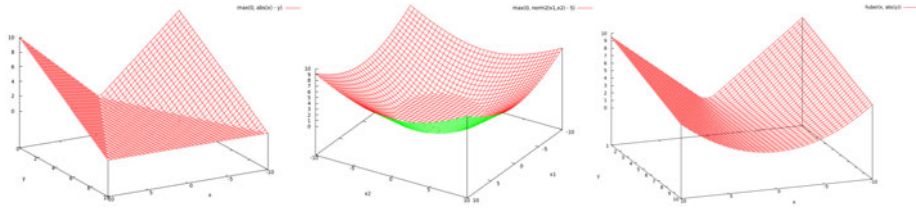
where we also employ the cheirality constraint of X_j being in front of the camera i, i.e. $X_{ij}^3 \geq 0$. The underlying norm can be the L_1 norm ($p = 1$), the Euclidean one ($p = 2$), or the maximum norm ($p = \infty$). Observe that the constraint Eq. 4 corresponds to a second order cone ($p = 2$) and the intersection of affine linear half-spaces, respectively.

If T_i's and X_j's can be determined such that Eq. 4 is fulfilled for all i and j, then all reprojection errors are less or equal to σ (either using L_2 or L_∞ distances in the image). If this is not the case, one can measure the infeasibility s_{ij} of the projected 3D point [11],

$$\|u_{ij} X_{ij}^3 - X_{ij}^{1,2}\|_p \leq \sigma X_{ij}^3 + s_{ij},$$

or the necessary offset vectors in object space to move X_{ij} onto the cone, $\omega_{ij} \in \mathbb{R}^2$ [10],

$$\|u_{ij} X_{ij}^3 - X_{ij}^{1,2} + \omega_{ij}\|_p \leq \sigma X_{ij}^3.$$



(a) $g_1(x, y)$ (b) $g_2(x, y)$ for $y = 5$ (c) $H(x, y)$

Fig. 2. Surface plots for the two pan functions g_1 (a) and g_2 (for a fixed value of y, (b)), and the bivariate Huber cost function (c)

Since nonzero values of s_{ij} or ω_{ij} correspond to outlier measurements in the image, it is reasonable to search for sparse solutions in terms of s_{ij} and ω_{ij}, respectively, i.e. to minimize the L_0 norm of s_{ij} (or ω_{ij}). Convexification of the L_0 norm yields the following L_1 objective function and constraints (using the offset vector formulation):

$$\begin{aligned} \min_{T_i, X_j, \omega_{ij}} \sum_{ij} \|\omega_{ij}\|_1 \quad & \text{s.t.} \\ \left\| u_{ij} X_{ij}^3 - X_{ij}^{1,2} + \omega_{ij} \right\|_p \leq \sigma X_{ij}^3 \quad & \forall ij \\ X_{ij} = R_i X_j + T_i & . \end{aligned} \tag{5}$$

If $p = \infty$ (and also for $p = 1$) this is a linear program, and for $p = 2$ one obtains a second order cone program, which can be solved by suitable convex optimization codes.

In order to avoid the degenerate solution $T_i = 0$ and $X_j = 0$ for all cameras and 3D points in the optimization problem Eq. 5 and to avoid a 4-parameter family of solutions (arbitrary global translation and scale), one has to enforce suitable cheirality constraints (e.g. $X_{ij}^3 \geq 1$ [10]) or fix the reference frame [11]. Utilizing the cheirality constraint implicitly selects the smallest feasible reconstruction with respect to its scale, since the objective function in Eq. 5 is reduced by decreasing the global scale.

4 Our Approach

Generic optimization of Eq. 5 using a linear or second order cone programming toolbox turns out to be not efficient in practice. One reason for the inefficiency is the introduction of auxiliary variables, either s_{ij} or ω_{ij}. Another difficulty for generic optimization codes is the large number of non-local constraints. Hence, we propose to directly optimize a non-differentiable objective function without the need for additional unknowns.

4.1 The Cost Functions

In order to reformulate Eq. 5 in more general terms, we define and analyze several convex functions, which are used in Sec. 4.2 to derive suitable proximal-point problems forming the basis of the numerical scheme.

Notations. We introduce the indicator function $\imath_S(x)$ returning 0 if $x \in S$ and ∞ otherwise. In particular, $\imath_{\mathbb{R}_0^+}$ and $\imath_{\mathbb{R}_0^-}$ denote the indicator functions for non-negative (respectively non-positive) real numbers. For a convex, lower semi-continuous function f, let f^* be its conjugate function, $f^*(z) = \max_x zx - f(x)$.

The "Pan" Functions. We define the pan function $g_d : \mathbb{R}^d \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ as

$$g_d(x, y) = \max\{0, \|x\|_2 - y\}. \tag{6}$$

For a particular value of $y \geq 0$ the shape of $g_d(\cdot, y)$ is a truncated L_1 cost function resembling the cross-section of a pan-like shape (see also Fig. 2(a) and (b)). It can be also viewed as two-sided variant of a hinge loss function. This function is convex, but not differentiable at $\|x\|_2 = y$.

As a first step to derive the conjugate function of g_d, we observe that

$$g_d(x, y) = \max_{\|z\|_2 \leq 1} \{z^T x - \|z\|_2 y\}. \tag{7}$$

Omitting the subscript in $\|\cdot\|_2 = \|\cdot\|$, if $\|x\| \leq y$ we have $z^T x - \|z\| y \leq \|z\|\,\|x\| - \|z\| y = \|z\|(\|x\| - y)$ (by the Cauchy-Schwarz inequality), but the second factor is non-positive by assumption, hence maximizing over z in the unit disc yields $z = 0$ with objective value 0. $\|x\| > y$: observe that $\|z\| y$ is independent of the direction of z and $z^T x$ is maximal if $z \parallel x$, i.e. $z = kx$ for some $k \in [0, \|x\|^{-1}]$ (since $\|z\| \leq 1$). Overall, $z^T x - \|z\| y = k\|x\|^2 - k\|x\| y = k\|x\|(\|x\| - y)$, hence $k = \|x\|^{-1}$ (i.e. $z = x/\|x\|$) maximizes that expression with value $\|x\| - y$. Overall, both definitions Eqs. 6 and 7 are again equivalent.

Finally, we can convert Eq. 7 into a bilinear form by introducing the additional variable v,

$$g_d(x, y) = \max_{\|z\| \leq 1, v \leq -\|z\|} \{z^T x + vy\}. \tag{8}$$

Note that $g_d(x, y)$ is ∞ for $y < 0$, since v is not bounded from below. For given $x \in \mathbb{R}^d$ and $y > 0$ the maximization always gives $v = -\|z\|$, since the objective can be increased whenever $v < -\|z\|$. Thus, the definitions in Eqs. 7 and 8 are equal, and Eq. 8 allows us to directly read off the corresponding conjugate function $g_d^*(z, v) = \imath_{C_d}(z, v)$ with $C_d \equiv \{(z, v) : \|z\| \leq 1, \|z\| \leq -v\}$. Thus, we can reduce the computation of $\text{prox}_{\gamma g_d}$ essentially to the projection into the set C_d in view of Moreau's decomposition Eq. 2. Projecting into C_d can be done in closed form and distiction of cases.

The Bivariate Huber Cost Function. Instead of having a combined L_∞/L_1 cost function as described in the previous section, one can also consider penalizing a squared residual for inliers as defined by the respective 3D cone, and an L_1 penalizier for outliers. Define the bivariate Huber cost function $H : \mathbb{R}\times\mathbb{R}_0^+ \to \mathbb{R}_0^+$ by

$$H(x,y) = \begin{cases} \frac{x^2}{2y} & |x| \le y \\ |x| - y/2 & |x| \ge y \end{cases} \tag{9}$$

for $y \ge 0$ (see also Fig. 2(a)). We denote this function as bivariate Huber cost since it also takes the inlier threshold (here y) as additional parameter. Unlike the squared reprojection error, which is only a quasi-convex function, $H(x,y)$ is convex in $\mathbb{R} \times \mathbb{R}_0^+$. The conjugate function of the univariate Huber cost (i.e. as a function only of x) is readily derived as

$$H^*(z;y) = \frac{y}{2}z^2 + \imath_{[-1,1]}(z). \tag{10}$$

Note that partial conjugation with respect to x is not sufficient for our purpose, since $\frac{y}{2}z^2$ is not a bilinear expression in the primal and dual variables. We combine $H^*(z;y)$ with $\imath_{\mathbb{R}_0^+}(y)$ corresponding to the constraint $y \ge 0$, and obtain

$$\begin{aligned} H(x,y) = H(x;y) + \imath_{\mathbb{R}_0^+}(y) &= \max_{z\in[-1,1]} \left(zx - \frac{y}{2}z^2 \right) + \max_{v\le 0} vy \\ &= \max_{\substack{z\in[-1,1] \\ v' \le -z^2/2}} zx + v'y, \end{aligned}$$

where we substituted $v' = v - z^2/2$ and used the fact $\imath_{\mathbb{R}_0^+}(y) = \max_{v\le 0} vy$. Note that the last line allows us to immediately identify the conjugate function of H with respect to both arguments as

$$H^*(z,v) = \imath_{[-1,1]}(z) + \imath_{\mathbb{R}_0^-}(v + z^2/2). \tag{11}$$

The feasible region for (z,v) is the region K below the parabola $-z^2/2$ intersected by $[-1,1]\times\mathbb{R}$. Finding the closest point in K requires to determine the nearest point on a parabola segment, which leads to solving a 3rd order polynomial. Luckily, this cubic polynomial is in depressed form (i.e. of the form $x^3 + px + q = 0$) and is known to have only one real root. Hence, projecting an arbitrary pair (z,v) into the set K is tractable although not extremely cheap (due to the necessary computation of cubic roots).

The cost functions and the respective conjugates introduced in this section allow the efficient application of proximal methods to the multi-view reconstruction problem as discussed in the following.

4.2 Application to Multi-view Reconstruction

Our formulation of the multi-view reconstruction approach with known camera rotations follows very closely the model Eq. 5. In our experiments we observed

that solely fixing the gauge freedom by setting two translation vectors can still result in quasi-degenerate solutions. The utilized L_1 penalizer on the offset variables measures deviations in object space (in contrast to image space) and therefore induces a strong prior towards solutions collapsing many of the scene points and camera centers into a single point. Further, strictly enforcing the cheirality constraint for all 3D points is very restrictive and does not cope well with outlier correspondences ultimately triangulated behind one of the cameras. Therefore, we slightly modify the convex formulation of Eq. 5 into

$$\begin{aligned} \min_{T_i, X_j, \omega_{ij}, \rho_{ij}} \sum_{ij} \|\omega_{ij}\|_1 + \sum_{ij} [\rho_{ij}]_+ \quad \text{s.t.} \\ \left\| u_{ij}^{1,2} X_{ij}^3 - X_{ij}^{1,2} + \omega_{ij}^{1,2} \right\| \leq \sigma(X_{ij}^3 + \rho_{ij}) \\ X_{ij}^3 + \rho_{ij} \geq 1 \qquad\qquad X_{ij} = R_i X_j + T_i, \end{aligned} \tag{12}$$

where $[\cdot]_+ \equiv \max\{0, \cdot\}$. Hence, we look for sparse correction values ω_{ij} and ρ_{ij} such that all corrected 3D points lie inside the cone induced by the measured image projection u_{ij} (first inequality constraint) and fulfill the cheirality constraint (second inequality constraint). By observing the following equivalence,

$$g_d(x, y) = \min_{s \in \mathbb{R}^d : \|x+s\|_2 \leq y} \|s\|_2,$$

we can use the pan function introduced in Sec. 4.1 to eliminate the auxiliary variables ω_{ij}, and obtain the equivalent problem (illustrated first for $d = 1$, i.e. the anisotropic variant of the inlier cone):

$$\min_{T_i, X_j, \rho_{ij}} \sum_{ij} \sum_{l=1}^{2} g_1\left(u_{ij}^l \bar{X}_{ij}^3 - X_{ij}^l, \sigma \bar{X}_{ij}^3\right) + \sum_{ij} \imath_{[1,\infty)}(\bar{X}_{ij}^3) + \sum_{ij} [\rho_{ij}]_+, \tag{13}$$

with $X_{ij} = R_i X_j + T_i$ and $\bar{X}_{ij}^3 = X_{ij}^3 + \rho_{ij}$. The choice $d = 2$ leads to a similar (now isotropic) problem using g_2 instead of g_1:

$$\min_{T_i, X_j, \rho_{ij}} \sum_{ij} g_2\left(u_{ij}^{1,2} \bar{X}_{ij}^3 - X_{ij}^{1,2}, \sigma \bar{X}_{ij}^3\right) + \sum_{ij} \imath_{[1.\infty)}(\bar{X}_{ij}^3) + \sum_{ij} [\rho_{ij}]_+. \tag{14}$$

Both problems are convex minimization tasks with non-differentiable objective functions. Observe that the arguments of g_d and $\imath$ are linear in the unknowns T_i, X_j and ω_{ij}.

Finally, instead of having just zero cost for inliers (as in the two objectives above), squared deviations from the observed image points over the distance can be penalized by utilizing the bivariate Huber function,

$$\min_{T_i, X_j, \rho_{ij}} \sum_{ij} \sum_{l=1}^{2} H\left(u_{ij}^l \bar{X}_{ij}^3 - X_{ij}^l, \sigma \bar{X}_{ij}^3\right) + \sum_{ij} \imath_{[1,\infty)}(\bar{X}_{ij}^3) + \sum_{ij} [\rho_{ij}]_+. \tag{15}$$

This formulation essentially approximates the squared reprojection error (i.e. squared numerator and denominator) by a convex quadratic-over-linear function

for inlier points. Consequently, the 3D points in the corresponding solution tend to stay closer to the observed image measurements. More importantly, inlier 3D points are attracted by a unique minimum and the numerical procedure converges faster in practice.

At the first glance nothing is gained by such reformulation other than moving the (linear or second order cone) constraints into the objective (Eqs. 13 and 14), and allowing for a refined cost for inlier points (Eq. 15). But the problems are very structured: the objective function is a sum of many convex functions only taking very few arguments, therefore depending only on a small subset of the unknowns. Hence, a variant of Douglas-Rachford splitting can be applied as decribed in the following section.

4.3 Numerical Scheme

The objective functions in the previous section can be more generally written as

$$\min_{\mathcal{X}} \sum_k h_k(L_k \mathcal{X}), \tag{16}$$

where $\mathcal{X}$ denotes all unknowns T_i, X_j and ρ_{ij}, the h_k are convex functions and L_k are matrices of appropriate dimensions. Similar to dual decomposition methods we can introduce local unknowns x_k for each h_k and explicitly enforce global consistency (see also [18]),

$$\min_{\mathcal{X}, \mathcal{Y}_k} \underbrace{\sum_k h_k(\mathcal{Y}_k)}_{\equiv f_1} + \underbrace{\sum_k \imath\{L_k \mathcal{X} = \mathcal{Y}_k\}}_{\equiv f_2}. \tag{17}$$

Application of Douglas-Rachford splitting amounts to solving $\text{prox}_{\gamma f_1}$ and $\text{prox}_{\gamma f_2}$ (recall Eq. 3). The first proximity operator, $\text{prox}_{\gamma f_1}$ decouples into independent problems $\text{prox}_{\gamma h_k}$, in particular (referring to Eq. 15, with analogous expressions for Eqs. 13 and 14) the term $h_k(L_i \mathcal{X})$ is one of

$$h_k(L_i \mathcal{X}) = \begin{cases} H\big(u^l_{ij} \bar{X}^3_{ij} - X^l_{ij}, \sigma \bar{X}^3_{ij}\big) & l \in \{1, 2\} \\ \imath_{[1,\infty)}(\bar{X}^3_{ij}) \\ [\rho_{ij}]_+ \end{cases}$$

For h_k equal to g_d or H we can utilize the derivations from Section 4.1 in order to determine $\text{prox}_{\gamma g_d}$ or $\text{prox}_{\gamma H}$ efficiently. If $h_k = \imath_{[1,\infty)}(\cdot)$, the proximity operator can be easily derived as clamping operation into the feasible domain $[1, \infty)$, and for $h_k = [\cdot]_+$ the respective proximity operator is given by

$$\text{prox}_{\gamma [\cdot]_+}(\bar{x}) = \max(0, \bar{x} - \gamma).$$

$\text{prox}_{\gamma f_2}(\mathcal{X}, (\mathcal{Y}_k)_k)$ corresponds to finding the closest point $(\hat{\mathcal{X}}, (\hat{\mathcal{Y}}_k)_k)$ satisfying the linear constraints $L_k \hat{\mathcal{X}} = \hat{\mathcal{Y}}_k$ for all k, therefore $\text{prox}_{\gamma f_2}$ is a projection operation into a linear subspace. Following [19] a particular instance of a Douglas-Rachford approach called simultaneous-direction method of multipliers (SDMM)

can be stated: choose $\gamma > 0$ and arbitrary initial values $y_k^{(0)}$ and $z_k^{(0)}$, and iterate for $n = 0, 1, \ldots$:

$$\begin{aligned} x^{(n)} &= \left(\sum_k L_k^T L_k \right)^{-1} \sum_k L_k^T \left(y_k^{(n)} - z_k^{(n)} \right) \\ y_k^{(n+1)} &= \operatorname{prox}_{\gamma h_k} \left(L_k x^{(n)} + z_k^{(n)} \right) \quad \forall k \\ z_k^{(n+1)} &= z_k^{(n)} + L_k x^{(n)} - y_k^{(n+1)} \quad \forall k. \end{aligned} \tag{18}$$

$x^{(n)}$ converges to a solution of Eq. 16. Note that the L_k are very sparse matrices, and the inverse of $\sum L_k^T L_k$ can be efficiently found using sparse Cholesky factorization. Note that a projection into a linear subspace is always uniquely defined, hence $\sum L_k^T L_k$ must have full rank.

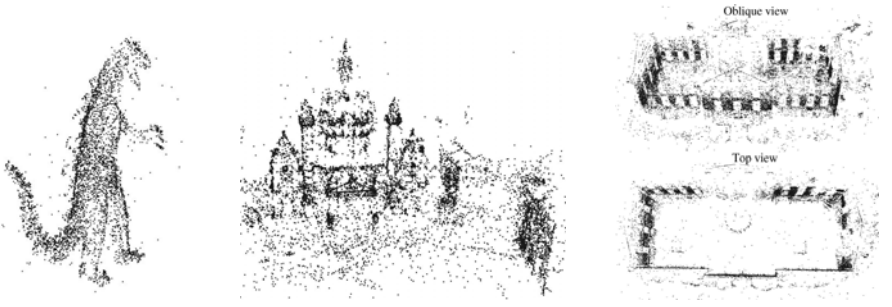
Since the method is iterative, a suitable stopping criterion is required. We selected the following one: the current 3D structure and the one obtained after a significant number of iterations (our choice is 1000) are normalized in their sizes, and the maximum change in the respective 3D point positions is used as stopping criterion. If no 3D point was moved more than ε, then the update iterations terminate. We use $\varepsilon = 10^{-8}$ in our implementation. Further, we set $\gamma = 0.1$ in all experiments.

5 Numerical Results

Most work in convex and quasi-convex optimization in multiple view geometry uses the "Dinosaur" data-set[1] consisting of camera poses and tracked 2D points. We use the rotations known from the given projection matrices, and determine the 3D structure and camera centers using the convex formulations described in Sec. 4.2. We obtain numerical and timing results for several optimization codes: (i) we utilize an easy-to-use simplex-based linear program solver[2] to optimize the problem Eq. 5 (denoted as "Simplex" in the evaluation Table 1); (ii) we also experiment with an interior point algorithm for semi-definite programs (DSDP [20], which has also a direct interface to specify LP cones, indicated by DSDP in Table 1)); and finally we implemented the update equations Eq. 18 for the described cost functions to minimize the respective energy. Table 1 summarizes the obtained timing and accuracy results, where we set the inlier radius σ to one pixel. Of particular interest in the evaluation is the dependence of the run-time on the data-set size: generally, the iterative proximal methods scale better with the data-set size. Both generic LP solvers were not able to complete the full data-set (due to numerical instabilities for the Simplex method and excessive memory consumption of the semi-definite code DSDP).

These performance numbers need to be compared with the close to two hours reported in [11] for the full data-set. Note that [10] does not indicate the respective run-time performance. The Huber function based model Eq 15 is strictly

[1] Available from http://www.robots.ox.ac.uk/~vgg/data/data-mview.html
[2] http://lpsolve.sourceforge.net/



(a) "Dinosaur" (b) "Church" (c) "Building"

Fig. 3. Reconstruction result for the "Dinosaur" data set (36 images, $\approx$ 5.000 3D points), "Church" data set (58 views, $\approx$ 12.500 3D points), and "Building" (128 images, $\approx$ 27.000 3D points) using the energy function Eq 15. No bundle adjustment is applied.

Table 1. Accuracy and run-time results for the "Dinosaur" sequence for increasing data-set sizes. Results are shown for Simplex and interior-point based (DSDP) solvers optimizing the model Eq. 5, and proposed iterative methods for all energy models

# views	# 3D points	Method	Run-time	L_∞ inlier error	L_2 inlier error	# of outliers
8	843	Simplex	9.5s	0.913	1.17	33/2465
		DSDP	2m27s	0.430	0.480	33/2465
		Anisotr.	11.4s	0.622	0.712	33/2465
		Isotropic	4s	0.466	0.511	32/2465
		Huber cost	5.4s	0.223	0.235	48/2465
16	1915	Simplex	1m43s	0.892	1.10	165/5905
		DSDP	24m37s	0.389	0.434	165/5905
		Anisotr.	26.1s	0.705	0.800	165/5905
		Isotropic	13.5s	0.668	0.721	165/5905
		Huber cost	13.7s	0.256	0.271	207/5905
24	3205	Simplex	19m26s	0.895	1.10	294/10416
		DSDP	103m36s	0.369	0.411	294/10416
		Anisotr.	1m44s	0.895	0.756	294/10416
		Isotropic	1m33s	0.784	0.838	311/10416
		Huber cost	1m16s	0.262	0.278	371/10416
36	4983	Anisotr.	1m51s	0.650	0.734	383/16432
		Isotropic	45s	0.483	0.533	421/16432
		Huber cost	46s	0.269	0.286	539/16432

convex for inlier points (and therefore has a unique minimizer), which could be the explanation for reaching the stopping criterion faster than the other energy models Eqs. 13 and 14.

Since the Huber cost model Eq. 15 penalizes deviations from the ray induced by the image measurements, the final reprojection error of the 3D points are consistently smaller than for the combined L_∞/L_1 cost models. The last column in Table 1 depicts the number of reported outlier measurements outside the respective σ radius in the image. This value is stated to show the equivalence of the

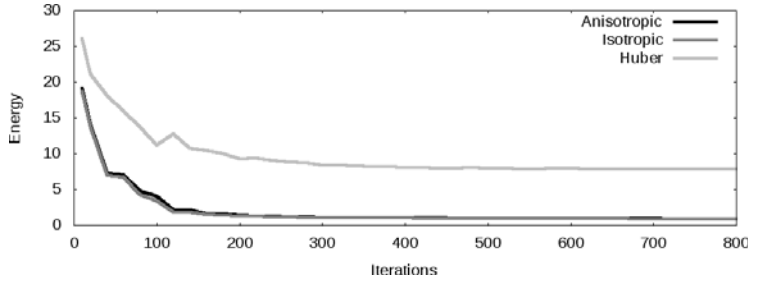


Fig. 4. Temporal evolution of the objective value with respect to iterations for the full Dinosaur dataset. 100 iterations corresponds to 0.6s (anisotropic and isotropic objectives) and 1.1s (Huber cost function) of run-time.

iterative method for Eq. 13 with the linear programming formulation. Further, it shows that the smaller reprojection error in the Huber cost model is compensated by a larger number of reported outliers (which is a result of the energy model and not induced by the implementation). We also applied the proposed method on real-world data sets consisting of 58 and 128 views, respectively (see Fig. 3(b) and (c)). Consistent rotations are estimated via relative poses from SIFT feature matches. The iterative Huber approach (for $\sigma = 2$ pixels) requires 12.5 and 40 minutes to satisfy the convergence test, and about 30% is spent in approximate column reordering for the sparse Cholesky factorization. The mean reprojection errors for inliers are 1.08 and 0.53 pixels, respectively. Figure 4 depicts the convergence rate of the objective value for the Dinosaur data set with respect to the number of iterations. Since the objective function is rather flat near the global minimum, small changes in the objective value do not necessarily imply small updates in the variables, and the termination criterion is achieved much later.

6 Conclusion

In this work we present a different view on convex problems arising in mutiple-view geometry, and propose a non-standard optimization method for these problems. By looking at classical optimization tasks in geometric vision from a general convex optimization perspective we can formulate interesting new geometric cost functions and also provide practical minimization procedures.

Future work needs to explore other applications in geometric computer vision potentially taking advantage of the proposed formulation and the associated numerical method, e.g. for quasi-convex problems. Finally, GPU-based implementations of the numerical methods are expected to result in significant reductions of the run-time for the proposed methods.

Acknowledgements. We would like to thank Manfred Klopschitz and Alexander Schwing for their valuable support and feedback.

References

1. Hartley, R., Schaffalitzky, F.: L_∞ minimization in geometric reconstruction problems. In: Proc. CVPR (2004)
2. Kahl, F.: Multiple view geometry and the L_∞ norm. In: Proc. ICCV, pp. 1002–1009 (2005)
3. Ke, Q., Kanade, T.: Quasiconvex optimization for robust geometric reconstruction. In: Proc. ICCV, pp. 986–993 (2005)
4. Sim, K., Hartley, R.: Recovering camera motion using L_∞ minimization. In: Proc. CVPR (2006)
5. Govindu, V.M.: Combining two-view constraints for motion estimation. In: Proc. CVPR, pp. 218–225 (2001)
6. Brand, M., Antone, M., Teller, S.: Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 262–273. Springer, Heidelberg (2004)
7. Seo, Y., Lee, H.L.S.W.: Sparse structures in L-Infinity norm minimization for structure and motion reconstruction. In: Proc. ECCV, pp. 780–793 (2006)
8. Seo, Y., Hartley, R.: A fast method to minimize L_∞ error norm for geometric vision problems. In: Proc. CVPR (2007)
9. Sim, K., Hartley, R.: Removing outliers using the L_∞ norm. In: Proc. CVPR (2006)
10. Dalalyan, A., Keriven, R.: L_1-penalized robust estimation for a class of inverse problems arising in multiview geometry. In: NIPS (2009)
11. Seo, Y., Lee, H.L.S.W.: Outlier removal by convex optimization for l-infinity approaches. In: Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology, pp. 203–214 (2009)
12. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Proc. CVPR, pp. 684–691 (2004)
13. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: Proc. CVPR (2007)
14. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Modeling and Simulation 4, 1168–1200 (2005)
15. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. 16, 964–979 (1978)
16. Combettes, P.L., Pesquet, J.C.: A douglas-rachford splitting approach to nonsmooth convex variational signal recovery. IEEE J. Selected Topics Signal Processing 1, 564–574 (2007)
17. Esser, E.: Applications of lagrangian-based alternating direction methods and connections to split bregman. UCLA CAM Report TR09-31 (2009)
18. Setzer, S., Steidl, G., Teuber, T.: Deblurring Poissonian images by split Bregman techniques. Journal of Visual Communication and Image Representation (2009)
19. Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing (2009) arXiv:0912.3522v2
20. Benson, S.J., Ye, Y., Zhang, X.: Solving large-scale sparse semidefinite programs for combinatorial optimization. SIAM Journal on Optimization 10, 443–461 (2000)

Building Rome on a Cloudless Day

Jan-Michael Frahm[1], Pierre Fite-Georgel[1], David Gallup[1], Tim Johnson[1], Rahul Raguram[1], Changchang Wu[1], Yi-Hung Jen[1], Enrique Dunn[1], Brian Clipp[1], Svetlana Lazebnik[1], and Marc Pollefeys[1,2]

[1] University of North Carolina at Chapel Hill, Department of Computer Science
[2] ETH Zürich, Department of Computer Science

Abstract. This paper introduces an approach for dense 3D reconstruction from unregistered Internet-scale photo collections with about 3 million images within the span of a day on a single PC ("cloudless"). Our method advances image clustering, stereo, stereo fusion and structure from motion to achieve high computational performance. We leverage geometric and appearance constraints to obtain a highly parallel implementation on modern graphics processors and multi-core architectures. This leads to two orders of magnitude higher performance on an order of magnitude larger dataset than competing state-of-the-art approaches.

1 Introduction

Recent years have seen an explosion in consumer digital photography and a phenomenal growth of community photo-sharing websites. More than 80 million photos are uploaded to the web every day,[1] and this number shows no signs of slowing down. More and more of the Earth's cities and sights are photographed each day from a variety of cameras, viewing positions, and angles. This has created a growing need for computer vision techniques that can provide intuitive and compelling visual representations of landmarks and geographic locations. In response to this challenge, the field has progressed quite impressively. Snavely et al. [1] were the first to demonstrate successful structure from motion (SfM) from Internet photo collections. Agarwal et al. [2] have performed camera registration and sparse 3D reconstruction starting with 150,000 images in a day on 62 cloud computers with 500 cores. Li et al. [3] have presented a system that combines appearance and multi-view geometry constraints to process tens of thousands of images in little more than a day on a single computer. There also exist techniques for accurate reconstruction of dense 3D models from community photo collections [4,5], but they are currently much slower and more computationally intensive than the SfM approaches.

While the above systems represent significant advances for 3D reconstruction, they do not yet measure up to the needs of city-scale modeling as, for example, a query for "Rome" on Flickr.com returns about 3 million images. This paper proposes a highly efficient system performing camera registration and *dense*

[1] http://royal.pingdom.com/2010/01/22/internet-2009-in-numbers

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 368–381, 2010.

Fig. 1. Example models of our method from Rome (left) and Berlin (right) computed in less than 24 hours from automatically registered subsets of photo collections of 2.9 million and 2.8 million images respectively

geometry estimation for city-scale reconstruction from millions of images on a single PC (no cloud computers = "cloudless"). The proposed system brings the computation of models from Internet photo collections on par with state-of-the-art performance for reconstruction from video [6] by extending the capabilities of each step of the reconstruction pipeline to efficiently handle the variability and complexity of large-scale, unorganized, heavily contaminated datasets.

Our method efficiently combines 2D appearance and color constraints with 3D multi-view geometry constraints to estimate the geometric relationships between millions of images. The resulting registration serves as a basis for dense geometry computation using fast plane sweeping stereo [7] and a new method for robust and efficient depthmap fusion [8]. We take advantage of the appearance and geometry constraints to achieve parallelization on graphics processors and multi-core architectures. All timings in the paper are obtained on a PC with dual Intel quadcore Xeon 3.33 Ghz processors, four NVidia 295GTX commodity graphics cards, 48 GB RAM and a 1 TB solid state hard drive for data storage. The major steps of our method are:

1) Appearance-based clustering with small codes (Sec. 3.1): Similarily to Li et al. [3] we use *gist* features [9] to capture global image appearance. The complexity of the subsequent geometric registration is reduced by clustering the gist features to obtain a set of *canonical* or *iconic* views [3]. In order to be able to fit several million gist features in GPU-memory, we compress them to compact binary strings using a locality sensitive scheme [10,11,12]. We then cluster them based on Hamming distance using the k-medoids algorithm [13] implemented on the GPU. To our knowledge, this is the first application of small codes in the style of [12] outside of proof-of-concept recognition settings, and the first demonstration of their effectiveness for large-scale clustering problems.

2) Geometric cluster verification (Sec. 3.2) is used to identify in each cluster a "core" set images with mutually consistent epipolar geometry using a fast

RANSAC method [14]. Out of the core images an iconic image is selected and all remaining cluster images are verified to match to it, and the ones found to be inconsistent are removed. Finally, we select a single iconic view as the best representative of the cluster. Additionally, we take advantage of user-provided geo-location of the images if it is available (approximately 10% of the images in the Internet photo collection have this information), providing geo-location for a large fraction of our clusters ($> 66\%$).

3) Local iconic scene graph reconstruction (Sec. 3.3) establishes a skeleton registration of the iconic images in the different locations. We use vocabulary tree search [15] and clustering based on geo-location and image appearance to identify neighboring iconics. Both of these strategies typically lead to sets of locally connected images corresponding to different geographically separated sites of the city. We dub these sets *local* iconic scene graphs. These graphs are extended by registering *all* the additional views from the iconic clusters linked to them. Our registration method uses incremental SfM combined with periodic bundle adjustment, achieving accuracies on par with existing methods [2,16].

5) Dense model computation (Sec. 3.4) uses all registered views in the local iconic scene graphs to compute dense scene geometry for the captured sites. Taking advantage of the initial appearance-based image grouping method, we use fast plane sweeping stereo to obtain depthmaps from each iconic cluster. To minimize the computational load we perform visibility-based view selection for the dense depthmap computation. Then we apply a novel extension to a depthmap fusion method [17] to obtain a watertight scene representation from the noisy but redundant depthmaps.

2 Previous Work

Our method is the first technique performing dense modeling from unorganized Internet-scale photo collections consisting of millions of images. There exist scalable systems for dense 3D urban reconstruction from millions of video frames [6,18]. However, modeling from video is inherently much more efficient as it takes advantage of spatial proximity between the camera positions of successive frames, whereas the spatial relationships between images in a community photo collection are unknown a priori, and in fact, 40% to 60% of images in such collections turn out to be irrelevant clutter [3].

The first approach for organizing unordered image collections was proposed by Schaffalitzky and Zisserman [19]. Sparse 3D reconstruction of landmarks from Internet photo collections was first addressed by the *Photo Tourism* system [20], which achieves high-quality results through exhaustive pairwise image matching and frequent global bundle adjustment. Neither one of these steps is very scalable, so in practice, the Photo Tourism system can be applied to a few thousand images at most. Aiming at scalability, Snavely et al. [16] construct *skeletal sets* of images whose reconstruction approximates the full reconstruction of the whole dataset. However, computing these sets still requires initial exhaustive pairwise

image matching. Agarwal et al. [2] parallelize the matching process and use approximate nearest neighbor search and query expansion [21] on a cluster of 62 machines each one comparable to our single PC. With that single PC, we tackle an order of magnitude more data in the same amount of time.

The speed of our approach is a result of efficient early application of 2D appearance-based constraints, similarly to the approach of Li et al. [3]. Our system takes each of the methods used by [3] to the next level to successfully process two orders of magnitude more data by parallelizing the computation on graphics processors and multi-core architectures. As a prerequisite for efficient camera registration, our method initially clusters the dataset by appearance and selects one *iconic image* for each appearance cluster. This is the opposite of the approach of Simon et al. [22], who treat scene summarization as a by-product of 3D reconstruction and select canonical views through clustering the 3D camera poses. While our method of image organization is initially looser than that of [22], it provides a powerful pre-selection mechanism for advancing the reconstruction efficiency significantly.

After selecting the iconic images, the next step of our system is to discover the geometric relationships between them and register them together through SfM. Li et al. [3] used a vocabulary tree [15] to rapidly find related iconics. Our system can use a vocabulary tree in the absence of geo-location information for the iconics. If this geo-location is available, we use it to identify potentially related views similarly to [23]. This approach is more efficient since it avoids building the vocabulary tree. Note that the methods of [24,25] are also applicable to discovering spatial relationships in large collections of data.

To perform SfM on the set of iconic images, Li et al. [3] first partitioned the iconic scene graph into multiple connected components. In contrast, we do not cut the iconic scene graph, as such an approach is prone to excessive fragmentation of scene models. Instead, we use a growing strategy combined with efficient merging and periodic bundle adjustment to obtain higher-quality, more complete models. This strategy is reminiscent of techniques for out-of-core bundle-adjustment [26] that take advantage of the uneven viewpoint distribution in photo collections by locally optimizing clusters of nearby views, and then connecting the local solutions into a global one.

Given the registered viewpoints recovered by SfM, we next perform multi-view stereo to get dense 3D models. The first approach demonstrating dense modeling from photo collections was proposed by Goesele et al. [4]. It uses per-pixel view selection and patch growing to obtain a set of surface elements, which are then regularized into a Poisson surface model. However, this approach does not make it easy to provide textures for the resulting models. Recently, Furukawa et al. [5] proposed a dense reconstruction method from large-scale photo collections using view clustering to initialize the PMVS approach [27]. The method of [5] computes a dense model from approximately 13,000 images in about two days on a single computer assuming known camera registration. Our proposed method uses an extended version of Yang and Pollefeys stereo [7] combined with novel multi-layer depthmap fusion [8]. While achieving comparable quality, it computationally

outperforms [4,5] by running on a single PC within less than two hours for 64,000 images.

3 The Approach

In this section we will describe our system. Section 3.1 discusses the initial appearance-based clustering, and Section 3.2 discusses our method for efficient geometric verification of the resulting clusters. Section 3.3 explains our SfM scheme, and Section 3.4 explains dense 3D model generation.

3.1 Appearance-Based Clustering with Small Codes

Similarily to Li et al. [3], we begin by computing a global appearance descriptor for every image in the dataset. We generate a gist feature [9] for each image by computing oriented edge responses at three scales (with 8, 8 and 4 orientations, respectively), aggregated to a 4×4 spatial resolution. To ensure better grouping of views for our dense reconstruction method, we concatenate the gist with a subsampled RGB image at 4×4 spatial resolution. Both the gist and the color parts of the descriptor are rescaled to have unit norm. The combined descriptor has 368 dimensions, and is computed on the eight GPU cores at a rate of 781Hz[2].

The next step is to cluster the gist descriptors to obtain groups of images consistent in appearance. We aim at a GPU-based implementation, given the inherent parallelism in the distance computation of clustering algorithms like k-means and k-medoids. Since it is impossible to efficiently cluster up to 2.8 million 368-dimensional double-precision vectors in the GPU memory of 768 MB, we have chosen to compress the descriptors to much shorter binary strings, such that the Hamming distances between the compressed strings approximate the distances between the original descriptors. To this end, we have implemented on the GPU the locality sensitive binary code (LSBC) scheme of Raginsky and Lazebnik [11], in which the ith bit of the code for a descriptor vector $\mathbf{x}$ is given by $\varphi_i(\mathbf{x}) = \mathrm{sgn}[\cos(\mathbf{x} \cdot \mathbf{r}_i + b_i) + t_i]$, where $\mathbf{r} \sim \mathrm{Normal}(0, \gamma I)$, $b_i \sim \mathrm{Unif}[0, 2\pi]$, and $t_i \sim \mathrm{Unif}[-1, 1]$ are randomly chosen code parameters. As shown in [11], as the number of bits in the code increases, the *normalized* Hamming distance (i.e., Hamming distance divided by code length) between two binary strings $\varphi(\mathbf{x})$ and $\varphi(\mathbf{y})$ approximates $(1 - K(\mathbf{x}, \mathbf{y}))/2$, where $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma \|\mathbf{x}-\mathbf{y}\|^2/2}$ is a Gaussian kernel between $\mathbf{x}$ and $\mathbf{y}$. We have compared the LSBC scheme with a simple locality sensitive hashing (LSH) scheme for unit norm vectors where the ith bit of the code is given by $\mathrm{sgn}(\mathbf{x} \cdot \mathbf{r}_i)$ [10]. As shown in the recall-precision plots in Figure 2, LSBC does a better job of preserving the distance relationships of our descriptors than the LSH scheme of [10].

We have found that $\gamma = 4.0$ works well for our data, and that the code length of 512 offers the best tradeoff between approximation accuracy and memory usage. To give an idea of the memory savings afforded by this scheme, at 32 bytes per dimension, each original descriptor takes up 11,776 bytes, while the

[2] http://www.cs.unc.edu/~jmf/Software.html

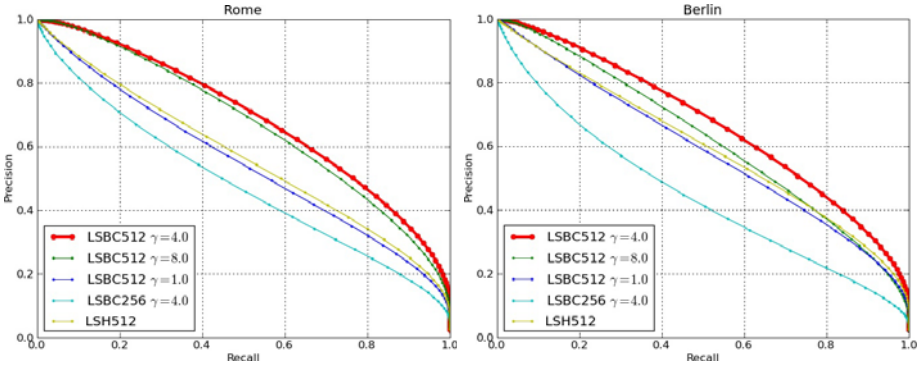


Fig. 2. Comparison of LSH [10] and LSBC [11] coding schemes with different settings for γ and code size on Rome data (left) and Berlin data (right). These plots show the recall and precision of nearest-neighbor search with Hamming distance on binary codes for retrieving the "true" k nearest neighbors according to Euclidean distance on the original gist features (k is our average cluster size, 28 for Rome and 26 for Berlin). For our chosen code size of 512, the LSBC scheme with $\gamma = 4$ outperforms LSH.

corresponding binary vector takes up only 64 bytes, thus achieving a compression factor of 184. With this amount of compression, we can cluster up to about 4 million images on our memory budget of 768 MB, versus only a few hundred thousand images in the original gist representation.

For clustering the binary code vectors with the Hamming distance, we have implemented the k-medoids algorithm [13] on the GPU. Like k-means, k-medoids alternates between updating cluster centers and cluster assignments, but unlike k-means, it forces each cluster center to be an element of the dataset. For every iteration, we compute the Hamming distance matrix between the binary codes of all images and those that correspond to the medoids. Due to the size of the dataset and number of cluster centers, this distance matrix must be computed

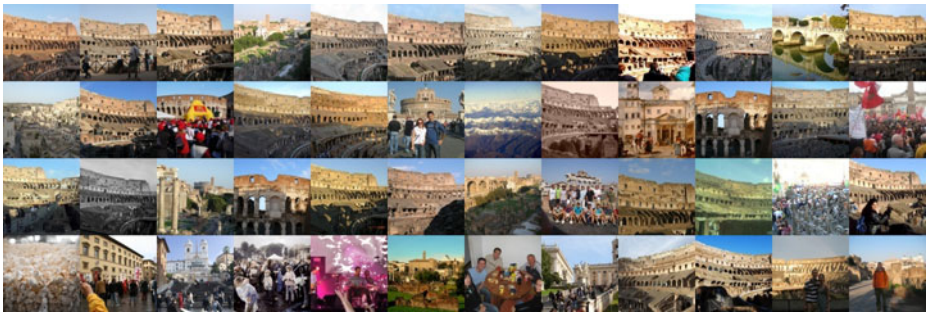


Fig. 3. Images closest to the center of one cluster from Rome

piecewise, as it is too large to store. An example of a typical cluster output by our system is shown in Figure 3.

An open problem for clustering in general is how to initialize the cluster centers, as the initialization can have a big effect on the end results. We found that images with available geo-location information (typically $10-15\%$ of our datasets) provide a good sampling of the points of interest (see Figure 4). Thus, we first cluster the codevectors of images with available geo-location into k_{geo} clusters initialized randomly. The resulting centers are used together with additional k_{rand} random centers to initialize the clustering of the complete dataset (in all our experiments $k_{geo} = k_{rand}$). From Table 2 it can be seen that we gain about 20% more geometrically consistent images by this initialization strategy.

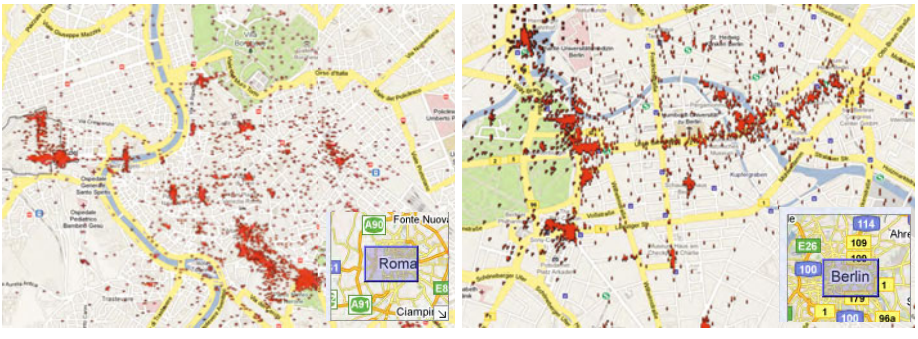


Fig. 4. Geo-tag density map for Rome (left) and Berlin (right)

3.2 Geometric Verification

The clusters obtained in the previous step consist of images that are visually similar but may be geometrically and semantically inconsistent. Since our goal is to reconstruct scenes with stable 3D structure, we next enforce geometric consistency for images within a cluster. A cluster is deemed to be consistent if it has at least n images with a valid pairwise epipolar geometry. This is determined by selecting an initial subset of n images (those closest to the cluster medoid) and estimating the two-view geometry of all the pairs in this subset while requiring at least m inliers (in all our experiments we use $n = 4$, $m = 18$). Inconsistent images within the subset are replaced by others until n valid images are found, or all cluster images are exhausted and the cluster is rejected.

The computation of two-view epipolar geometry is performed as follows. We extract SIFT features [28] using an efficient GPU implementation,[3] processing 1024×768 images at approximately 10.5 Hz on a single GPU including disk access. In the interest of computational efficiency and memory bandwidth, we limit the number of features extracted to 4000 per image. Next, we calculate the putative SIFT matches for each image pair. This computationally demanding

[3] http://www.cs.unc.edu/~ccwu/siftgpu

process (which could take a few seconds per pair on the CPU) is cast as a matrix multiplication problem on multiple GPUs (with a speedup of three orders of magnitude to 740 Hz), followed by a subsequent distance ratio test [28] to identify likely correspondences.

The putative matches are verified by estimation of the fundamental matrix with ARRSAC [14] using a 7-point algorithm [29]. ARRSAC is a robust estimation framework designed for efficient real-time operation. Similarly to generic RANSAC, its performance significantly degrades for small inlier ratios. However, we have observed that of all registered images in the three datasets, a significant fraction has inlier ratios above 50% (e.g., for San Marco, this fraction is 72%). This makes intuitive sense: it has been observed [16,3] that community photo collections contain a tremendous amount of viewpoint overlap and redundancy, which is particularly pronounced at the scale at which we operate. We use this to our advantage by limiting the maximum number of tested hypotheses to 400 in ARRSAC, which corresponds to inlier ratio of approximately 50%. To improve registration performance, we take the solution deemed the most promising by the SPRT test of ARRSAC, and perform *post hoc* refinement. The latter enables us to recover a significant fraction of solutions with less than 50% inlier ratio. Comparing the number of images registered by the standard ARRSAC and by our modified procedure shows a loss of less than 3% for Rome and less than 5% for Berlin, while having an approximately two- to five-fold gain in speed.

We choose a representative or "iconic" image for each verified cluster as the image with the most inliers to the other $n - 1$ top images. Afterwards all other cluster images are only verified with respect to the iconic image. Our system processes all the appearance-based clusters independently using 16 threads on 8 CPU cores and 8 GPU cores. In particular, the process of putative matching is distributed over multiple GPUs, while the robust estimation of the fundamental matrix utilizes the CPU cores. This enables effective utilization of all available computing resources and gives a significant speedup to about 480 Hz for verification. An example of a verified cluster is shown in Figure 5.

Whenever possible, we geo-locate the verified clusters using images with user-provided geo-tags. Our geo-location procedure evaluates pairwise geographic distances of all geo-tagged images in the iconic cluster. Then it performs a weighted voting on the locations of all images within a spatial proximity of the most central image as defined by the pairwise distances. This typically provides geo-location for about two thirds of the iconic clusters.

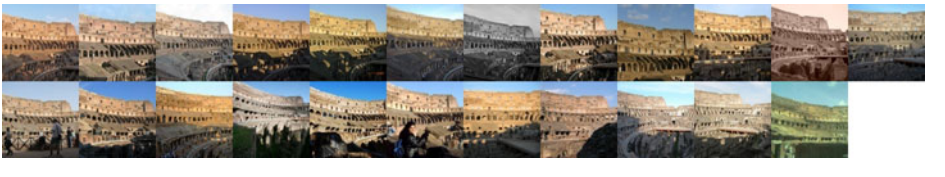


Fig. 5. The cluster of Figure 3 following geometric verification

3.3 Local Iconic Scene Graph Reconstruction

After identifying the geometrically consistent clusters, we need to establish pairwise relationships between the iconics. Li et al. [3] introduced the *iconic scene graph* to encode these relationships. We use the same concept but identify multiple *local* iconic scene graphs corresponding to the multiple geographic sites within each dataset. This keeps the complexity low despite the fact that our sets of iconics are comparable in size to the entire datasets of [3].

We experimented with two different schemes for efficiently obtaining candidate iconic pairs for geometric verification. The first scheme is applicable in the absence of any geo-location. It is based on building a vocabulary tree index for the SIFT features of the iconics, and using each iconic to query for related images. The drawback of this scheme is that the mapping of the vocabulary tree has to be rebuilt specifically for each set of iconics, imposing a significant overhead on the computation. The second scheme avoids this overhead by using geo-location of iconic clusters. In this scheme, the candidate pairs are defined as all pairs within a certain distance s of each other (in all our experiments set to $s = 150$ m). As for the iconics lacking geo-location, they are linked to their l nearest neighbors ($l = 10$ in all experiments) in the binarized gist descriptor space (the distance computation uses GPU-based nearest-neighbor search as in the k-medoids clustering). We have found this second scheme to be more efficient whenever geo-location is available for a sufficient fraction of the iconics (as in our Rome and Berlin datasets). For both schemes, all the candidate iconic pairs are geometrically verified as described in Section 3.2, and the pairs with a valid epipolar geometry are connected by an edge. Each connected set of iconics obtained in this way is a *local iconic scene graph*, usually corresponding to a distinct geographic site in a city.

Next, each local iconic scene graph is processed independently to obtain a camera registration and a sparse 3D point cloud using an incremental approach. The algorithm picks the pair of iconic images whose epipolar geometry given by the essential matrix (computed similarly to the fundamental matrix in Section 3.2) has the highest inlier number and delivers a sufficiently low reconstruction uncertainty, as computed by the criterion of [30]. Obtaining a metric two-view reconstruction requires a known camera calibration, which we either obtain from the EXIF-data of the iconics (there are 34% EXIF based calibrations for the Berlin dataset and 40% for Rome), or alternatively we approximate the calibration by assuming a popular viewing angle for the camera model. The latter estimate typically approximates the true focal length within the error bounds of successfully executing the five-point method [31]. To limit drift after inserting i new iconics, the 3D sub-model and camera parameters are optimized by a sparse bundle adjustment [32]. The particular choice of i is not critical and in all our experiments we use $i = 50$. If no new images can be registered into the current sub-model, the process starts afresh by picking the next best pair of iconics not yet registered to any sub-model. Note that we intentionally construct multiple sub-models that may share some images. We use these images to merge newly completed sub-models with existing ones whenever sufficient 3D matches exist.

Fig. 6. Sample input images (courtesy of Flickr users: jdn, _parris_, Rictor Norton & David Allen, Felpa_Boy, Andreas Solberg, xiquinhosilva, HarshLight,stevesheriw), local iconic scene graph and 3D model

The merging step again uses ARRSAC [14] to robustly estimate a similarity transformation based on the identified 3D matches.

In the last stage of the incremental reconstruction algorithm, we complete the model by incorporating non-iconic images from iconic clusters of the registered iconics. This process takes advantage of the feature matches between the non-iconic images and their respective iconics known from the geometric verification (Section 3.2). The 2D matches between the image and its iconic determine 2D-3D correspondences between the image and the 3D model into which the iconic is registered, and ARRSAC is once again used to determine the camera pose. Detailed results of our 3D reconstruction algorithm are shown in Figure 6, and timings are given in Table 1.

3.4 Dense Geometry Estimation

Once the camera poses have been recovered, the next step is to recover the surface of the scene, represented as a polygonal mesh, and to reconstruct the surface color, represented as a texture map. We use a two-phase approach for surface reconstruction: first, recover depthmaps for a selected number of images, and second, fuse the depthmaps into a final surface model.

One of the major challenges of stereo from Internet photo collections is appearance variation. Previous approaches [4,5] take great care to select compatible views for stereo matching. We use the clustering approach from Section 3.1 to cluster all images registered in the local iconic scene graph. Since our gist descriptor encodes color, the resulting clusters contain images similar in color. The availability of images similar in color within a spatially confined area enables us

Table 1. Computation times (hh:mm hrs) for the reconstruction for the Rome and Berlin dataset using geo-tags, and the San Marco dataset without geo-tags

Dataset	Gist & Clustering	SIFT & Geom. verification	Local iconic scene graph	Dense	total time
Rome & geo	1:35 hrs	11:36 hrs	8:35 hrs	1:58 hrs	23:53 hrs
Berlin & geo	1:30 hrs	11:46 hrs	7:03 hrs	0:58 hrs	21:58 hrs
San Marco	0:03 hrs	0:24 hrs	0:32 hrs	0:07 hrs	1:06 hrs

Table 2. Number of images for the Rome dataset with the use of geo-location (Rome & geo) and without geo-location (Rome), the Berlin dataset with geo-tags (Berlin & geo) and without (Berlin), and the San Marco dataset without the use of geo-location, column 4 shows the number of iconic images, column 5 gives the total number of geometrically consistent images, column 6 gives the number of images registered in the 64 largest models, column 7 lists the number of registered views in the largest model

		LSBC	#images			
Dataset	total	clusters	iconics	verified	3D models	largest model
Rome & geo	2,884,653	100, 000	21,651	306788	63905	5671
Rome	2,884,653	100, 000	17874	249689	-	-
Berlin & geo	2,771,966	100, 000	14664	124317	31190	3158
San Marco	44, 229	4,429	890	13604	1488	721

to use traditional stereo methods. We use GPU-accelerated plane sweep stereo [33] with a 3×3 normalized cross-correlation matching kernel. Our stereo deploys 20 matching views, and handles occlusions (and other outliers) through taking the best 50% of views per pixel as suggested in [34]. We have found that within a set of 20 views, non-identical views provide a sufficient baseline for accurate depth computation.

We adapted the vertical heightmap approach of [17] for depthmap fusion to handle more geometrically complex scenes. This method is intended to compute watertight approximate surface models. It assumes that the vertical direction of the scene is known beforehand. For community photo collections, this direction can be easily obtained using the approach of [35] based on the assumption that most photographers will keep the camera's x-axis perpendicular the vertical direction. The heightmap is computed by constructing an occupancy grid over a volume of interest. The resolution of the heightmap is determined by the median camera-to-point distance, which is representative of the scene-scale and the accuracy of the depth measurements. All points below the heightmap surface are considered full and all points above are considered empty. Each vertical column of the grid is computed independently. For each vertical column, occupancy votes are accumulated from the depthmaps. Points between the camera center and the depth value receive empty votes, and points beyond the depth value receive a full vote with a weight that falls off with distance. Then a height value is determined that minimizes the number of empty votes above and the number of full votes below. Our extension is to allow the approach to have multiple connected "segments" within the column, which provides higher-quality mesh models while maintaining the regularization properties of the original approach. A polygonal mesh is then extracted from the heightmap and texture maps are generated as the mean of all images observing the geometry (refer to [8] for details). The heightmap model is robust to noise and it can be computed very efficiently on the GPU. Runtimes are provided in Table 1, and Table 2 shows the number of views registered in the the dense 3D models.

4 Conclusions

This paper demonstrates the first system able to deliver dense geometry for city-scale photo collections within the span of a day on a single PC. This system extends to the scale of millions of images state-of-the-art methods for appearance-based clustering [3], robust estimation [14], and stereo fusion [8]. To successfully handle reconstruction problems of this magnitude, we have incorporated novel system components for clustering of small codes, geo-location of iconic images through their clusters, efficient incremental model merging, and stereo view selection through appearance-based clustering. Beyond making algorithmic changes, we were able to significantly improve performance by leveraging the constraints from appearance clustering and location independence to parallelize the processing on modern multi-core CPUs and commodity graphics cards.

Acknowledgements. We like to thank NSF grant IIS-0916829, DOE grant DE-FG52-08NA28778, Lockheed Martin, SPAWAR, Nvidia, NSF CAREER award IIS-0845629, Microsoft Research Faculty Fellowship, and Xerox for their support.

References

1. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. In: SIGGRAPH, pp. 835–846 (2006)
2. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building Rome in a day. In: ICCV (2009)
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. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: ICCV (2007)
5. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Towards internet-scale multi-view stereo. In: Proceedings of IEEE CVPR (2010)
6. Pollefeys, M., Nister, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. IJCV Special Issue on Modeling Large-Scale 3D Scenes (2008)
7. Yang, R., Pollefeys, M.: Multi-resolution real-time stereo on commodity graphics hardware. In: CVPR, pp. 211–217 (2003)
8. Gallup, D., Pollefeys, M., Frahm, J.M.: 3d reconstruction using an n-layer heightmap. In: DAGM (2010)
9. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
10. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of the ACM 51, 117–122 (2008)
11. Raginsky, M., Lazebnik, S.: Locality sensitive binary codes from shift-invariant kernels. In: NIPS (2009)
12. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large databases for recognition. In: CVPR (2008)
13. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, Chichester (1990)
14. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 500–513. Springer, Heidelberg (2008)
15. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
16. Snavely, N., Seitz, S.M., Szeliski, R.: Skeletal sets for efficient structure from motion. In: CVPR (2008)
17. Gallup, D., Frahm, J.M., Pollefeys, M.: A heightmap model for efficient 3d reconstruction from street-level video. In: 3DPVT (2010)
18. Cornelis, N., Cornelis, K., Van Gool, L.: Fast compact city modeling for navigation pre-visualization. In: CVPR (2006)

19. 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)
20. Snavely, N., Seitz, S.M., Szeliski, R.: Modeling the world from Internet photo collections. IJCV 80, 189–210 (2008)
21. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
22. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: ICCV (2007)
23. Strecha, C., Pylvanainen, T., Fua, P.: Dynamic and scalable large scale image reconstruction. In: CVPR (2010)
24. Chum, O., Matas, J.: Web scale image clustering: Large scale discovery of spatially related images. Technical Report, CTU-CMP-2008-15 (2008)
25. Philbin, J., Zisserman, A.: Object mining using a matching graph on very large image collections. In: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (2008)
26. Ni, K., Steedly, D., Dellaert, F.: Out-of-core bundle adjustment for large-scale 3d reconstruction. In: ICCV (2007)
27. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. Trans. PAMI (2009)
28. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
29. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
30. Beder, C., Steffen, R.: Determining an initial image pair for fixing the scale of a 3D reconstruction from an image sequence. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 657–666. Springer, Heidelberg (2006)
31. Nistér, D.: An efficient solution to the five-point relative pose problem. Trans. PAMI 26, 756–770 (2004)
32. Lourakis, M., Argyros, A.: The design and implementation of a generic sparse bundle adjustment software package based on the Levenberg-Marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH (2004)
33. Kim, S., Gallup, D., Frahm, J., Akbarzadeh, A., Yang, Q., Yang, R., Nister, D., Pollefeys, M.: Gain adaptive real-time stereo streaming. In: International Conference on Computer Vision Systems, ICVS (2007)
34. Kang, S., Szeliski, R., Chai, J.: Handling occlusions in dense multi-view stereo. In: CVPR (2001)
35. Szeliski, R.: Image alignment and stitching: A tutorial. Microsoft Research Technical Report (2004)

Camera Pose Estimation Using Images of Planar Mirror Reflections

Rui Rodrigues, João P. Barreto, and Urbano Nunes

Institute of Systems and Robotics, Dept. of Electrical and Computer Engineering,
University of Coimbra, 3030 Coimbra, Portugal
{rrodrigues,jpbar,urbano}@isr.uc.pt

Abstract. The image of a planar mirror reflection (IPMR) can be interpreted as a virtual view of the scene, acquired by a camera with a pose symmetric to the pose of the real camera with respect to the mirror plane. The epipolar geometry of virtual views associated with different IPMRs is well understood, and it is possible to recover the camera motion and perform 3D scene reconstruction by applying standard structure-from-motion methods that use image correspondences as input. In this article we address the problem of estimating the pose of the real camera, as well as the positions of the mirror plane, by assuming that the rigid motion between N virtual views induced by planar mirror reflections is known. The solution of this problem enables the registration of objects lying outside the camera field-of-view, which can have important applications in domains like non-overlapping camera network calibration and robot vision. We show that the positions of the mirror planes can be uniquely determined by solving a system of linear equations. This enables to estimate the pose of the real camera in a straightforward closed-form manner using a minimum of $N = 3$ virtual views. Both synthetic tests and real experiments show the superiority of our approach with respect to current state-of-the-art methods.

1 Introduction

It is well known that the image of a planar mirror reflection (IPMR) is equivalent to the image that would be acquired by a virtual camera located behind the mirror. Such virtual camera has a pose symmetric to the pose of the real camera with respect to the plane of reflection, and presents the same intrinsic parameters [1]. We can use a planar mirror for observing a scene from a viewpoint different from the actual viewpoint of the imaging device. This has been explored in the past for building planar catadioptric systems (PCS) able to provide multi-view imagery while using a single static camera, [2][3][4]. All these works assumed a specific mirror configuration for computing the rigid transformations between virtual views and achieve extrinsic calibration.

Gluckman and Nayar were the first authors studying the epipolar geometry between virtual views induced by planar mirror reflections with the mirrors being in an arbitrary configuration [5]. They proved that in general a pair of IPMRs is related by a fundamental matrix, and that the rigid displacement between the corresponding virtual cameras is always a planar motion. Since in a planar motion the translation is orthogonal to the rotation axis, then the fundamental matrix between IPMRs has only 6 degrees of freedom (DOF) [6].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 382–395, 2010.

Without lost of generality, let's consider a sequence of IPMRs obtained by moving a planar mirror in front of a static camera, c.f. Fig. 1(a). Since there is an epipolar geometry relating the virtual views, it is possible to recover the virtual camera motion and compute the scene structure using a suitable structure-from-motion (SfM) approach [6],[7]. However, this approach, neither provides the pose of the real camera, nor determines the location of the mirror planes. This article proposes a robust closed-form method for accurately estimation of the 3D pose of the real camera using the rigid displacement between $N \geq 3$ virtual views.

Knowing the pose of the real camera enables to determine the rigid transformation between the camera and scene objects lying outside of its FOV, which can be useful for many application scenarios. Hesch *et al.* consider a camera mounted on a robotic platform, and describe a method that uses planar mirror reflections for estimating the rigid transformation relating camera and robot reference frames [8]. They track points in hundreds of IPMRs acquired while moving the robot in front of a mirror, and use a complicated iterative ML estimator to find the transformation. Still in the context of hand-eye calibration, Mariotinni *et al.* propose several geometric methods for localizing the real camera and the mirror planes in stereo PCS [9]. However, all the methods require the real camera and the virtual views to simultaneously observe a part of the scene, which somehow undermines the usefulness of the approach.

Closely related to our work is the article by Sturm *et al.* [10], that determines the position of an object lying outside the camera FOV. They observe that the planar motion between pairs of virtual views [5] can be described as a *fixed-axis rotation*, with the rotation axis being the intersection of the two mirror planes. Thus, they propose a constructive approach that, for a certain virtual view, estimates the fixed-axes of rotation in a closed form manner. It determines the mirror position by fitting a plane to the estimated 3D lines, and finally it recovers the real camera pose by reflecting the virtual view with respect to the mirror. The article proves for the first time that the position of mirrors and camera can be uniquely determined from a minimum of $N = 3$ virtual views and, in addition, it discusses singular configurations. The main difference to our work is that we recover the mirror position in a single estimation step by solving a new system of linear equations. The experimental results show that our formulation significantly improves accuracy and increases robustness.

Another related work is the one of Kumar *et al.* [11], concerning the calibration of camera networks with non-overlapping FOV. The extrinsic calibration of non-overlapping nodes is achieved by registering them with respect to a reference object that is observed through planar mirror reflections. The poses of the real cameras are obtained by solving a system of linear equations derived from orthogonality relations between axes of different reference frames. The linear estimator proposed by Kumar *et al.* requires a minimum of $N = 5$ views and provides a closed form solution that is sub-optimal. This solution is used as an initial estimate for a subsequent refinement step using bundle adjustment. We present tests, with both synthetic and real data, that show that the estimations obtained with this approach are substantially less accurate and robust than the ones achieved with our algorithm.

Notation: Matrices are represented in sans serif font, e.g. M, vectors in bold,e.g. $\mathbf{Q}$ $\mathbf{q}$, and scalars in italic, e.g. d. We do not distinguish between a linear transformation

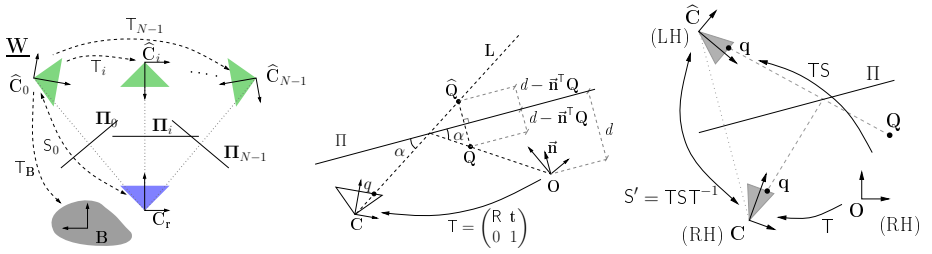


Fig. 1. The geometry of IPMR: (a) The object $\mathbf{B}$ is observed by camera $\mathbf{C}_r$ through N planar mirror reflections, with each mirror plane $\boldsymbol{\Pi}_i$ giving rise to a virtual camera $\widehat{\mathbf{C}}_i$. The $N-1$ rigid transformations T_i (and possibly the pose of the object T_B) are known 'a priori'. We recover the position of mirror $\boldsymbol{\Pi}_0$ and localize the real camera $\mathbf{C}_r$ by reflecting $\widehat{\mathbf{C}}_0$. (b) The 3D point $\mathbf{Q}$ is seen through a planar mirror reflection. The line of back-projection $\mathbf{L}$ always goes trough point $\widehat{\mathbf{Q}}$, that is symmetric to $\mathbf{Q}$ with respect to the mirror plane $\boldsymbol{\Pi}$. (c) The virtual camera $\widehat{\mathbf{C}}$ and the real camera $\mathbf{C}$ are symmetric to each other with respect to the mirror Π. Remark that any symmetry transformation causes a change in the handiness of the coordinate systems.

and the matrix representing it. If nothing is said 3D points are represented in non-homogeneous coordinates, and the vector product is carried using the skew symmetric matrix, e.g. $\mathbf{t} \times \mathbf{n} = [\mathbf{t}]_\times \mathbf{n}$. The symbol$\widehat{\ }$signals virtual entities induced by planar mirror reflections, e.g. $\widehat{\mathbf{Q}}$, and vectors topped with an arrow are versors with unitary norm, e.g. $\vec{\mathbf{n}}$.

2 Projection Model for Images of Planar Mirror Reflections

This section shows how to model the projection of a camera seeing a scene through a planar mirror reflection. We re-formulate some background concepts that have been introduced in the past [5],[9],[11], [10].

2.1 Projection of a 3D Point

Consider the scheme of Fig. 1(b) showing a camera with projection center $\mathbf{C}$ and a 3D planar mirror $\boldsymbol{\Pi}$. Both entities are expressed with respect to a world reference frame with origin in $\mathbf{O}$. The mapping of world coordinates into camera coordinates is carried by a 4×4 matrix T in the special euclidean group *se(3)* [12].

$$\mathsf{T} = \begin{pmatrix} \mathsf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix}$$

with R being a rotation matrix, and $\mathbf{t}$ a 3×1 translation vector. Plane $\boldsymbol{\Pi}$ is uniquely defined by its normal, represented by the unitary vector $\vec{\mathbf{n}}$, and the scalar euclidean distance d. A generic point $\mathbf{X}$ lies on the plane $\boldsymbol{\Pi}$ *iff* it satisfies the following equation

$$\vec{\mathbf{n}}^\mathsf{T} \mathbf{X} = d$$

Let $\mathbf{Q}$ be the coordinates of a 3D point that is observed through a planar mirror reflection. The reflection follows Snell's law, and $\mathbf{Q}$ is projected into the image point $\mathbf{q}$. It is easy to see that the back-projection line $\mathbf{L}$ always goes through point $\widehat{\mathbf{Q}}$, that is the symmetric of $\mathbf{Q}$ with respect to $\boldsymbol{\Pi}$. Since every point on $\mathbf{L}$ is imaged into $\mathbf{q}$, we can use the symmetry to go around the Snell's reflection. Thus, the projection of $\mathbf{Q}$ into $\mathbf{q}$ can be expressed in homogeneous coordinates by

$$\mathbf{q} \sim \mathsf{K}\,(\mathsf{I}\,0)\;\mathsf{T}\begin{pmatrix}\widehat{\mathbf{Q}}\\ 1\end{pmatrix}, \tag{1}$$

with I being the 3×3 identity matrix, and K the matrix of intrinsic parameters [6].

From Fig. 1(b) it follows that the world coordinates of the symmetric point $\widehat{\mathbf{Q}}$ are

$$\widehat{\mathbf{Q}} = \mathbf{Q} + 2\,(d - \vec{\mathbf{n}}^{\mathsf{T}}\,\mathbf{Q})\vec{\mathbf{n}}$$

The equation can be re-written in the following matrix form:

$$\begin{pmatrix}\widehat{\mathbf{Q}}\\ 1\end{pmatrix} = \mathsf{S}\begin{pmatrix}\mathbf{Q}\\ 1\end{pmatrix} \tag{2}$$

with S being a *symmetry transformation* induced by $\boldsymbol{\Pi}$

$$\mathsf{S} = \begin{pmatrix}\mathsf{I} - 2\vec{\mathbf{n}}\vec{\mathbf{n}}^{\mathsf{T}} & 2d\vec{\mathbf{n}}\\ 0 & 1\end{pmatrix} \tag{3}$$

2.2 Symmetry Matrices

Let's denote by *ss(3)* the set of all 4×4 symmetry matrices S whose structure is given in equation 3. The top-left 3×3 sub-matrix of any element in *ss(3)* is always a Householder matrix [13]. Taking into account the properties of the Householder matrices, it is straightforward to prove that the following holds:

(i) The symmetry transformations are involutionary: $\mathsf{S}^{-1} = \mathsf{S}\,, \forall_{\mathsf{S}\in ss(3)}$

(ii) The product of two symmetry transformations is a rigid transformation: $\mathsf{S}_1\,\mathsf{S}_2 \in se(3)\,, \forall_{\mathsf{S}_1\,\mathsf{S}_2\,\in\,ss(3)}$ (*ss(3)* is not an algebraic group)

(iii) In general the product of a symmetry transformation S by a rigid transformation T is not an element of *ss(3)*

2.3 The Virtual Camera

Consider the projection equation 1. Replacing $\widehat{\mathbf{Q}}$ by the result of equation 2 yields:

$$\mathbf{q} \sim \mathsf{K}\,(\mathsf{I}\,0)\;\mathsf{T}\,\mathsf{S}\begin{pmatrix}\mathbf{Q}\\ 1\end{pmatrix} \tag{4}$$

We can think on $\mathsf{T}\,\mathsf{S}$ as being a transformation that maps world coordinates into coordinates in a certain reference frame $\widehat{\mathbf{C}}$. It follows from equation 4, that the IPMR can be

modeled as the image that would be directly acquired by a *virtual camera* placed in $\widehat{\mathbf{C}}$ and having intrinsic parameters K [5]. It is also easy to realize that this virtual camera $\widehat{\mathbf{C}}$ and the *real camera* $\mathbf{C}$ are symmetric to each other with respect to the mirror plane $\boldsymbol{\Pi}$, c. f. Fig 1(c). The transformation from the real camera $\mathbf{C}$ to the virtual camera $\widehat{\mathbf{C}}$ is carried by the symmetry matrix S':

$$\mathsf{S}' = \mathsf{T}\,\mathsf{S}\,\mathsf{T}^{-1} \tag{5}$$

Since S' is involutionary, it also maps the virtual camera into the real camera. Moreover, and as pointed out by Kumar et al. [11], if the reference frame $\mathbf{C}$ is right-handed, then the reference frame $\widehat{\mathbf{C}}$ is left-handed, and vice-versa.

3 Assumptions and Problem Formulation

Without loss of generality, let's consider a situation where an object, lying outside the FOV of a static camera, is observed through planar mirror reflections obtained by freely moving the mirror in front of the camera [1]. As discussed in section 2.3, for each position of the mirror plane $\boldsymbol{\Pi}_i$, with $i = 0, 1, \ldots, N-1$, there is a virtual camera $\widehat{\mathbf{C}}_i$ that models the projection, c. f. Fig. 1(a). For convenience, it will be assumed that the world coordinate system is coincident with the reference frame of the virtual camera $\widehat{\mathbf{C}}_0$, and that geometric entities are expressed in world coordinates by default.

The rigid motions T_i of the virtual cameras $\widehat{\mathbf{C}}_i$, as well as the pose of the object T_B, are assumed to be known 'a priori'. They can be computed from image data using a suitable SfM approach [5] [6]. The choice of the most suitable method for the problem at hands, as well as eventual adaptations to take into account the constraints of planar camera motion [5], are beyond the scope of the work. Our objective is to estimate the pose of the real camera $\mathbf{C}_r$, using the rigid displacements T_i. Remark that, if the position of the real camera is known, then the mirror plane $\boldsymbol{\Pi}_i$ can be easily determined by finding the orthogonal plane that bisects the line that joins the centers of $\mathbf{C}_r$ and $\widehat{\mathbf{C}}_r$. On the other hand, if the position of the mirror plane $\boldsymbol{\Pi}_i$ is known, then the real camera can be determined by a symmetry transformation of $\widehat{\mathbf{C}}_i$ (equation 3). Therefore, we can solve the formulated problem by either estimating directly the pose of the real camera, or by finding the position of one of the mirror planes. We will pursue the latter strategy and look for the position of $\boldsymbol{\Pi}_\mathbf{0}$.

Finally, remark in Fig. 1(c) that the system of coordinates $\mathbf{O}$ is right-handed, while the virtual camera reference frame is left-handed. Since most implementations of popular SfM algorithms provide as output rigid transformations that preserve the handiness, it will be henceforth assumed that the scene is represented using left-handed coordinates. In this manner, the transformations between the object $\mathbf{B}$ and the virtual camera $\widehat{\mathbf{C}}_i$ can be carried by an element of *ss(3)*, at the expenses of considering a symmetry transformation between the object and the real camera, c.f. Fig 1(a).

[1] This is not geometrically equivalent to keeping the mirror stationary and moving the camera in an unconstrained manner. In this case the rigid displacement between virtual views is not necessarily a planar motion.

4 Searching for the Mirror Planes

As pointed out by Gluckman and Nayar, the rigid transformation between two virtual cameras is always a planar motion [5]. This section derives the constraints that each planar motion T_i imposes on the position of the planar mirrors $\boldsymbol{\Pi}_0$ and $\boldsymbol{\Pi}_i$.

Consider the scheme of Fig. 1(a), with the reference frame of camera $\widehat{\mathbf{C}}_0$ being the world coordinate system in which the different planes are represented. Let $\widehat{\mathbf{Q}}_0$ and $\mathbf{Q}_r$ be the coordinates of the same 3D point expressed with respect to $\widehat{\mathbf{C}}_0$ and $\mathbf{C}_r$ respectively. It follows that,

$$\begin{pmatrix} \mathbf{Q}_r \\ 1 \end{pmatrix} = \mathsf{S}_0 \begin{pmatrix} \widehat{\mathbf{Q}}_0 \\ 1 \end{pmatrix}$$

with S_0 being the symmetry transformation defined by plane $\boldsymbol{\Pi}_0$ (equation 3). If $\widehat{\mathbf{Q}}_i$ represents the same 3D point in the coordinate system of $\widehat{\mathbf{C}}_i$, it comes from equation 5 that

$$\begin{pmatrix} \mathbf{Q}_r \\ 1 \end{pmatrix} = \mathsf{T}_i \mathsf{S}_i \mathsf{T}_i^{-1} \begin{pmatrix} \widehat{\mathbf{Q}}_i \\ 1 \end{pmatrix}$$

By equaling the two equations above, replacing $\widehat{\mathbf{Q}}_i$ by $\mathsf{T}_i \widehat{\mathbf{Q}}_0$, and considering the properties of the symmetry matrices, we conclude that

$$\mathsf{T}_i = \mathsf{S}_0 \mathsf{S}_i$$

Considering explicitly the rotation R_i, the translation $\mathtt{t}_i$, and expressing the symmetries in terms of the plane parameters, it yields

$$\begin{pmatrix} \mathsf{R}_i & \mathbf{t}_i \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathsf{I} - 2\vec{\mathbf{n}}_0\vec{\mathbf{n}}_0^\mathsf{T} & 2d_0\vec{\mathbf{n}}_0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathsf{I} - 2\vec{\mathbf{n}}_i\vec{\mathbf{n}}_i^\mathsf{T} & 2d_i\vec{\mathbf{n}}_i \\ 0 & 1 \end{pmatrix} \tag{6}$$

4.1 Geometric Properties

Taking into account the result of equation 6, we can state the following property that relates the rotation R_i between virtual cameras, to the normals $\vec{\mathbf{n}}_0$ and $\vec{\mathbf{n}}_i$ of the corresponding mirror planes.

Statement 1: Let $\vec{\boldsymbol{\omega}}_i$ and θ_i denote, respectively, the direction of the rotation axis and the rotation angle of R_i. If α_i is the angle between the normals to the mirror planes such that $\vec{\mathbf{n}}_0^\mathsf{T}\,\vec{\mathbf{n}}_i = \cos(\alpha_i)$, then the following equalities hold:

$$\vec{\boldsymbol{\omega}}_i = \tfrac{\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0}{\sin(\alpha_i)} \bigwedge \theta_i = 2\,\alpha_i$$

Proof: Let R'_i be a rotation by an angle of $2\alpha_i$ around an axis with unit direction $(\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0)\sin(\alpha_i)^{-1}$. By applying Rodrigues' formula [12], it follows that:

$$\mathsf{R}'_i = \mathsf{I} + \frac{\sin(2\alpha_i)}{\sin(\alpha_i)}[\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0]_\times + \frac{(1-\cos(2\alpha_i))}{\sin(\alpha_i)^2}[\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0]_\times^2$$

We conclude after some algebraic manipulation that:

$$[\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0]_\times = \vec{\mathbf{n}}_0\vec{\mathbf{n}}_i^\mathsf{T} - \vec{\mathbf{n}}_i\vec{\mathbf{n}}_0^\mathsf{T}$$

$$[\vec{\mathbf{n}}_i \times \vec{\mathbf{n}}_0]_\times^2 = \cos(\alpha_i)\,(\vec{\mathbf{n}}_0\vec{\mathbf{n}}_i^\mathsf{T} + \vec{\mathbf{n}}_i\vec{\mathbf{n}}_0^\mathsf{T}) - \vec{\mathbf{n}}_0\vec{\mathbf{n}}_0^\mathsf{T} - \vec{\mathbf{n}}_i\vec{\mathbf{n}}_i^\mathsf{T}$$

Replacing the above results in the expression of R'_i, and performing some simplifications, it yields that

$$\mathsf{R}'_i = \mathsf{I} + 4\cos(\alpha_i)(\vec{\mathbf{n}}_0\vec{\mathbf{n}}_i^\mathsf{T}) - 2\,(\vec{\mathbf{n}}_0\vec{\mathbf{n}}_i^\mathsf{T} + \vec{\mathbf{n}}_i\vec{\mathbf{n}}_0^\mathsf{T})$$

Finally, taking into account that $\cos(\alpha_i) = \vec{\mathbf{n}}_0^\mathsf{T}\,\vec{\mathbf{n}}_i$, it follows that

$$\mathsf{R}'_i = (\mathsf{I} - 2\vec{\mathbf{n}}_0\vec{\mathbf{n}}_0^\mathsf{T})\,(\mathsf{I} - 2\vec{\mathbf{n}}_i\vec{\mathbf{n}}_i^\mathsf{T})$$

Thus, R'_i and R_i are the same rotation matrix (see equation 6), and the correctness of the statement has been proved. □

The next result was originally stated in [5], and it is presented in here for the sake of completeness

Statement 2: The rigid transformation T_i is a planar motion, with the translation component $\mathbf{t}_i$ being orthogonal to the rotation axis $\vec{\boldsymbol{\omega}}_i$.

Proof: From equation 6, and taking into account that $\vec{\mathbf{n}}_i^\mathsf{T}\,\vec{\mathbf{n}}_0 = \cos(\frac{\theta_i}{2})$, we can write the translation as

$$\mathbf{t}_i = 2(d_0 - 2\,d_i\,\cos(\frac{\theta_i}{2}))\,\vec{\mathbf{n}}_0 + 2\,d_i\,\vec{\mathbf{n}}_i \tag{7}$$

Since $\vec{\boldsymbol{\omega}}_i$ is orthogonal to both $\vec{\mathbf{n}}_0$ and $\vec{\mathbf{n}}_i$ (property 1) it is obvious that $\mathbf{t}_i^\mathsf{T}\,\vec{\boldsymbol{\omega}}_i, = 0$, which proves the orthogonality statement. □

4.2 Linear Constraints

As stated in section 3, it is sufficient to estimate the position of one mirror plane for solving for the pose of the real camera and position of remaining planes. It is shown in here that each planar motion T_i gives rise to a pair of independent linear constraints on the position of the plane $\boldsymbol{\Pi}_0$, c.f. Fig. 1(a). Henceforth, the rigid transformation is expressed in terms of $\vec{\boldsymbol{\omega}}_i$, θ_i and $\mathbf{t}_i$, while the planes are parametrized by their normals $\vec{\mathbf{n}}$ and scalar distances d.

Consider the transpose of equation 7 with both sides being multiplied by $\vec{\mathbf{n}}_0$. It arises after some algebraic manipulation that

$$\mathbf{t}_i^\mathsf{T}\,\vec{\mathbf{n}}_0 - 2\,d_0 + 2\,\cos(\frac{\theta_i}{2})\,d_i = 0 \tag{8}$$

A different equation can be derived by considering the cross product between $\mathbf{t}_i$ and $\vec{\mathbf{n}}_0$. From equation 7 and property 1, we obtain that

$$[\mathbf{t}_i]_\times\,\vec{\mathbf{n}}_0 - 2\,\sin(\frac{\theta_i}{2})\,\vec{\boldsymbol{\omega}}_i\,d_i = 0 \tag{9}$$

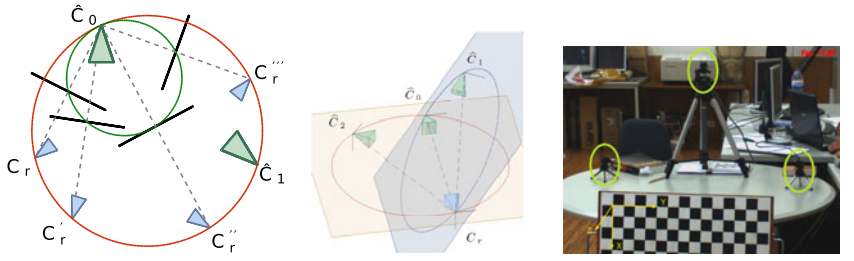


Fig. 2. (a) The space of possible solutions. $\widehat{\mathbf{C}}_0$ and $\widehat{\mathbf{C}}_1$ are two virtual cameras. The figure shows the planes $\boldsymbol{\Pi}_0$ satisfying equation 10, as well as the corresponding solutions for the real camera $\mathbf{C}_r$. The red circle is the locus of camera centers, while the green circle is the locus of points where $\boldsymbol{\Pi}_0$ bisects $\mathbf{C}_r\widehat{\mathbf{C}}_0$. It can be verified that there are an infinite number of solutions for the pose of the real camera that are consistent with the two virtual cameras. (b) Determining the real camera $\mathbf{C}_r$ for the case of $N = 3$. Consider the two pairs of virtual cameras $\widehat{\mathbf{C}}_0$, $\widehat{\mathbf{C}}_1$ and $\widehat{\mathbf{C}}_0$, $\widehat{\mathbf{C}}_2$. Each pair has a space of possible solutions for the real camera. The two loci intersect in two points: the common virtual camera $\widehat{\mathbf{C}}_0$, and the correct solution for the real camera $\mathbf{C}_r$. (c) Camera cluster setup used to evaluate the algorithms performance using IPMRs.

Equations 8 and 9 provide 4 linear constraints on the unknown plane parameters $\vec{\mathbf{n}}_0$, d_0 and d_i. However, and since $[\mathbf{t}_i]_\times$ is a singular matrix, only 3 constraints are independent. It is interesting to see that equation 9 implicitly enforces orthogonality between the rotation axis $\vec{\boldsymbol{\omega}}_i$ and vectors $\vec{\mathbf{n}}_0$ (property 1) and $\mathbf{t}_i$ (property 2).

By solving equation 8 in order to d_i, and replacing the result in equation 9, it arises

$$([\mathbf{t}_i]_\times + \tan(\frac{\theta_i}{2})\,\vec{\boldsymbol{\omega}}_i\,\mathbf{t}_i^{\mathsf{T}})\vec{\mathbf{n}}_0 - 2\tan(\frac{\theta_i}{2})\,\vec{\boldsymbol{\omega}}_i\,d_0 = 0 \tag{10}$$

Equation 10 provides 2 independent linear constraints on the parameters of plane $\boldsymbol{\Pi}_0$. Since the camera pose and mirror planes can be uniquely recovered by finding the position of one of those planes, the constrains of equation 10 constitute a minimal formulation of the problem in terms of the number of unknowns. However, these two independent constraints are insufficient to determine the 3 DOF of the mirror plane. As pointed out by Sturm and Bonfort [10] the problem is under-determined when using just a pair of virtual views. In this case the real camera can be at any location in a circular locus on the plane orthogonal to the axis of the relative rotation. This is illustrated in Fig. 2(a).

5 Determining the Real Camera from $N \geq 3$ Virtual Cameras

Three virtual cameras, $\widehat{\mathbf{C}}_0$, $\widehat{\mathbf{C}}_1$ and $\widehat{\mathbf{C}}_2$, define two independent planar motions T_1 and T_2. Each motion gives rise to two independent linear constraints on the parameters $\vec{\mathbf{n}}_0$ and d_0 (equation 10). Since we have a total of 4 equations and just 3 unknowns, then it is possible to uniquely determine the plane $\boldsymbol{\Pi}_0$ and estimate the pose of the real camera. Fig. 2(b) provides a geometric insight about localizing the real camera using $N = 3$ virtual cameras.

5.1 The System of Linear Equations

Consider N virtual cameras defining $N-1$ independent planar motions, c.f. Fig. 1(a). From section 4.2 follows that each motion T_i gives rise to a set of linear constraints on the parameters of the mirror plane $\boldsymbol{\Pi}_0$. Stacking the constraints of the $N-1$ motions leads to a system of linear equations that can be solved using a DLT estimation approach. We consider two alternative formulations for the system of equations: *Method 1* that is based on the result of equation 10, which becomes singular whenever $\theta_i = \pi$; and *Method 2* that uses the constraints derived in equations 8-9.

Method 1. The system of equations can be written as

$$\underbrace{\begin{pmatrix} \mathsf{A}_1 \\ \vdots \\ \mathsf{A}_{N-1} \end{pmatrix}}_{\mathsf{A}} \begin{pmatrix} \vec{\mathbf{n}}_0 \\ d_0 \end{pmatrix} = 0 \tag{11}$$

where each planar motion T_i defines a sub-matrix A_i with dimension 3×4 (equation 10)

$$\mathsf{A}_i = \left([\mathbf{t}_i]_\times + \tan(\tfrac{\theta_i}{2}) \vec{\boldsymbol{\omega}}_i \mathbf{t}_i^\mathsf{T} \; -2\tan(\tfrac{\theta_i}{2}) \vec{\boldsymbol{\omega}}_i \right)$$

Method 2. This second method uses the linear constraints derived in equations 8-9. Remark that the formulation is non-minimal in the sense that involves the scalar parameters d_i. The system of equations can be written in the following form

$$\underbrace{\begin{pmatrix} \mathsf{B}_1 & \mathbf{b}_1 & 0 & \dots & 0 \\ \mathsf{B}_2 & 0 & \mathbf{b}_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathsf{B}_{N-1} & 0 & 0 & \dots & \mathbf{b}_{N-1} \end{pmatrix}}_{\mathsf{B}} \begin{pmatrix} \vec{\mathbf{n}}_0 \\ d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-1} \end{pmatrix} = 0 \tag{12}$$

with B_i being 4×4 sub-matrices

$$\mathsf{B}_i = \begin{pmatrix} \mathbf{t}_i^\mathsf{T} & -2 \\ [\mathbf{t}_i]_\times & 0 \end{pmatrix}$$

and $\mathbf{b}_i$ vectors with dimension 4

$$\mathbf{b}_i = \begin{pmatrix} 2\cos(\frac{\theta_i}{2}) \\ -2\sin(\frac{\theta_i}{2}) \vec{\boldsymbol{\omega}}_i \end{pmatrix}$$

5.2 Outline of the Algorithm

1. Let T_i be the transformations mapping $\widehat{\mathbf{C}}_0$ into $\widehat{\mathbf{C}}_i$, with $i = 1, 2, \dots N-1$. For each T_i obtain the rotation axis $\vec{\boldsymbol{\omega}}_i$, the rotation angle θ_i and the translation vector $\mathbf{t}_i$

2. Build the $3(N-1) \times 4$ matrix A of equation 11 (for *method 2* build the $4(N-1) \times (N+3)$ matrix B of equation 12)
3. Apply SVD to find a least square solution for the system of equations (the vector result is denoted by $\mathbf{x}$).
4. Compute plane $\boldsymbol{\Pi}_0$ by making $\vec{\mathbf{n}}_0 = \frac{\mathbf{x}_{1\ldots3}}{||\mathbf{x}_{1\ldots3}||}$ and $d_0 = \frac{x_4}{||\mathbf{x}_{1\ldots3}||}$
5. Determine the symmetry matrix S_0 using $\vec{\mathbf{n}}_0$ and d_0 (equation 3). This matrix maps $\widehat{\mathbf{C}}_0$ into $\mathbf{C}_r$, enabling the localization of the real camera, c.f. Fig. 1(a).
6. Compute the position of remaining mirror planes $\boldsymbol{\Pi}_i$, by finding the plane orthogonal to the line $\mathbf{C}_r\bar{\mathbf{C}}_i$ that bisects it.

5.3 Singular Configurations

It has already been referred that the linear equations of method 1 present an algebraic singularity for the case of $\theta_i = \pi$. In addition, the null spaces of matrices A (*method 1*) and B (*method 2*) become multi-dimensional whenever all the mirror planes intersect into a single line in 3D. This is a singular configuration for which there are multiple solutions for the real camera pose. It can occur by either rotating the mirror around a fixed-axis, or by translating the mirror in a manner that all the reflection planes are parallel between them (the intersection line is at infinity). This degeneracy has been originally observed by Sturm and Bonfort [10].

6 Performance Evaluation with Synthetic Data

This section evaluates the accuracy of the algorithm outlined in section 5.2. The two linear formulations (*method 1* and *method 2*) are compared against the closed-form linear estimation approaches proposed by Sturm [10] and Kumar *et al.* [11]. For the latter we used the code implementation made publicly available by the authors, while for the former we re-implemented the method following the steps in the paper.

In order to have reliable ground truth we defined a simulation environment for generating synthetic data. We consider a working volume with a static camera. A set of N planes, simulating the mirrors, are generated assuming a convenient random uniform distribution. Each plane gives rise to a virtual camera using the symmetry transformation of equation 3. The positions of the virtual cameras are disturbed both in translation and rotation. The direction of the translation and rotation axis are drawn from a random uniform distribution, while the amount of disturbance is given by a zero mean normal distribution with variable standard deviation (the noise power). For convenience the errors in translation are relative errors defined as a percentage of the modulus of the true translation vector. The disturbed pose of the virtual cameras are the input to the different methods. The translation and rotation angle of the relative rigid motion between the estimate and ground truth are used as error metrics. We performed tests for different noise conditions. For each test the simulation was ran 100 times and the root mean square error was computed. The results are exhibited in Fig. 3.

According to the synthetic experiments the two linear methods presented in this paper outperform the linear approach proposed by Kumar *et al.* [11]. This is not surprising

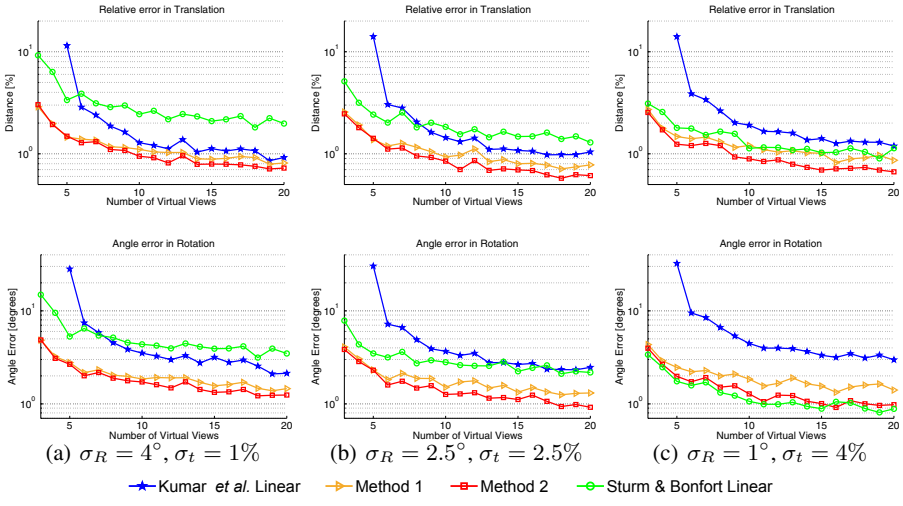


Fig. 3. Synthetic experiment for comparing the performance in estimating the real camera pose for an increasing number N of images of planar mirror reflections. The top row shows the RMS of the relative error in translation, while the bottom row concerns the angular RMS error in rotation. Each column concerns results assuming different standard deviations σ_R and σ_t for the additive noise in rotation and translation.

because the latter provides a sub-optimal solution in the sense that it estimates the rigid displacement between camera and object without taking into account the rotation constraints. The approach by Sturm and Bonfort is in general less accurate than both *method 1* and *method 2*, and seems to be highly sensitive to noise in the rotation between virtual views. However, it presents better results in the estimation of the real camera rotation for the case of $\sigma_R = 1^\circ$. This behavior can be explained by the fact that their algorithm fits the mirror to the estimated fixed rotation axes by first determining the normal orientation of the plane, and then its distance to the origin. Under negligible noise in rotation, the direction of the axes is accurately determined, and the normal to the mirror is correctly computed. Since the real camera rotation only depends on the mirror orientation, the final estimation error is very small. When the noise in rotation increases, the estimation of the direction of the fixed rotation axes becomes inaccurate, and the final results suffer a quick degradation.

Our two methods carry the estimation of the mirror position by simultaneously solving for the orientation and distance to the origin. This leads to a better trade-off in terms of sensitivity to noise. It also worth mentioning that during the simulations there were situations for which both Sturm's and Kumar's algorithm diverged in the sense that the output estimate was too off from the ground truth. In the case of Sturm's approach this happened occasionally for $N = 3$, while for Kumar's approach this happened quite often for $N \leq 6$. The situations of divergence were discarded and the random generation of data was repeated to produce the results of Fig. 3.

Somehow surprising is the fact that *method 2* systematically over-performs the minimal formulation of *method 1*. From synthetic results and experiences with real data,

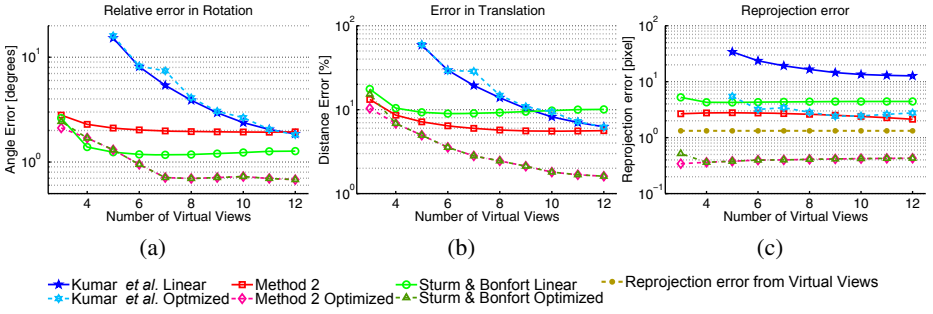


Fig. 4. Extrinsic calibration of a camera cluster with 3 nodes (Fig. 2(c)). The graphics compare the errors in estimating the relative displacement between camera nodes using different approaches. Since the cameras have overlapping FOV's, we determined an approximate ground truth by using a conventional extrinsic calibration approach [6].

method 2 remained the most accurate approach. In order to not overload the next graphs only the results using *Method 2* will be shown.

7 Experiments in Extrinsic Calibration

In this experiment we use IPMRs to perform the extrinsic calibration of a small cluster with 3 camera nodes. Fig. 2(c) shows the setup where a calibration grid is placed outside of the FOV of the cameras. Several images of the grid are acquired with the help of a mirror, and the corresponding virtual cameras are localized in the grid reference frame by homography factorization [7]. This is performed independently for each camera node. The virtual views are used as input for the different methods that return a pose estimate for the real camera. Since we are using a common object, the 3 cameras are registered in the same reference frame which enables extrinsic calibration. The results are subsequently refined by optimizing the re-projection error with respect to the mirror positions and real camera pose.

Fig. 4 shows the difference between the reference ground truth and the extrinsic calibration from mirror reflections. The graphics compare the performance of the different methods for an increasing number of virtual views $N = 3, \ldots, 12$. Since we collected a total of 12 images per node, we provide the RMS error taking into account the estimation results for all possible N image combinations. This real experiment confirms the conclusions reached in synthetic environment: Kumar *et al.* algorithm is in general inferior to both our method and Sturm's approach. It needs a larger number of views to provide accurate estimations, and it diverges for $N \leq 8$ views. For $N < 4$ views Sturm's approach also diverges in the cases that the mirrors are close to a singular parallel configuration. Table 1 shows the number of diverging cases for all methods. These situations were not considered in the statistical treatment of the results shown in Fig. 4. It is also important to refer that, since the determination of motion from planar homographies typically presents a good accuracy in rotation, this is an experiment that favors Sturm's approach, explaining the excellent performance in estimating the camera rotation.

Table 1. The table summarizes the number of situations for which each method fails in providing a suitable initial estimate. Our approach shows an impressive robustness even when the number of virtual views is small and/or are close to a singular parallel configuration.

Number of views	3	4	5	6	7	8	9	10	11	12
Total number of combinations	220	495	792	924	792	495	220	66	12	1
Method 2	0	0	0	0	0	0	0	0	0	0
Sturm and Bonfort Linear	8	2	0	0	0	0	0	0	0	0
Kumar *et al.* Linear			756	441	146	19	0	0	0	0

Fig. 4 also shows the optimized results from each method. It can be observed that Kumar's accuracy is not brilliant for $N \leq 8$ because of the poor closed-form initialization. Also Sturm's initialization for N=3 does not provide good convergence results because of the inaccuracy in the translation estimate. For $N \geq 3$ both Sturm's and our method converge towards the same solution.

8 Conclusions

This article provides a geometric insight about IPMRs and shows that the position of the mirror planes can be recovered in a straightforward manner by solving a system of linear equations. This leads to a new closed-form estimation method that is able to recover the pose of the real camera from a minimum of $N = 3$ views. Extensive experimental results clearly show the superiority of our approach with respect to the state-of-the-art. Our approached proved to have a well balanced sensitivity to noise in translation and rotation, a good behavior under situations of quasi-singularity, and an excellent accuracy even when the number of images is minimum.

Acknowledgments

The authors are grateful to the Portuguese Science Foundation by generous funding through grant PTDC/EEA-ACR/72226/2006. João P. Barreto also acknowledges the support through grant PTDC/EEA-ACR/68887/2006.

References

1. Baker, S., Nayar, S.K.: A theory of single-viewpoint catadioptric image formation. Int. Journal of Computer Vision 35(2), 175–196 (1999)
2. Goshtasby, A., Gruver, W.: Design of a single-lens stereo camera system. Pattern Recognition (1993)
3. Inaba, M., Hara, T., Inoue, H.: A stereo viewer based on a single camera with view-controlmechanisms. In: Intelligent Robots and Systems (1993)
4. Ramsgaard, B., Balslev, I., Arnspang, J.: Mirror-based trinocular systems in robot-vision. Pattern Recognition 4, 499–502 (2000)
5. Gluckman, J., Nayar, S.: Catadioptric stereo using planar mirrors. IJCV (2001)
6. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, New York (2003)

7. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-D Vision: From Images to Geometric Models. Springer, Heidelberg (2003)
8. Hesch, J., Mourikis, A., Roumeliotis, S.: Determining the camera to robot-body transformation from planar mirror reflections. In: IROS (2008)
9. Mariottini, G., Scheggi, S., Morbidi, F.: Planar catadioptric stereo: Single and multi-view geometry for calibration and localization. In: ICRA (2009)
10. Sturm, P., Bonfort, T.: How to compute the pose of an object without a direct view? In: ACCV, vol. II, pp. 21–31 (2006)
11. Kumar, R., Ilie, A., Frahm, J.M., Pollefeys, M.: Simple calibration of non-overlapping cameras with a mirror. In: CVPR, pp. 1–7 (2008)
12. Murray, R.M., Sastry, S.S., Zexiang, L.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Inc, Boca Raton (1994)
13. Golub, G.H., Van Loan, C.F.: Matrix computations, 3rd edn. Johns Hopkins Univ. Press, Baltimore (1996)

Element-Wise Factorization for N-View Projective Reconstruction

Yuchao Dai[1,2], Hongdong Li[2,3], and Mingyi He[1]

[1] School of Electronics and Information, Northwestern Polytechnical University
Shaanxi Key Laboratory of Information Acquisition and Processing, Xi'an China
[2] Australian National University, Australia
[3] Canberra Research Lab, NICTA, Australia

Abstract. Sturm-Triggs iteration is a standard method for solving the *projective factorization* problem. Like other iterative algorithms, this method suffers from some common drawbacks such as requiring a good initialization, the iteration may not converge or only converge to a local minimum, etc. None of the published works can offer any sort of global optimality guarantee to the problem. In this paper, an optimal solution to projective factorization for structure and motion is presented, based on the same principle of low-rank factorization. Instead of formulating the problem as *matrix factorization*, we recast it as *element-wise factorization*, leading to a convenient and efficient semi-definite program formulation. Our method is thus *global*, where no initial point is needed, and a globally-optimal solution can be found (up to some relaxation gap). Unlike traditional projective factorization, our method can handle real-world difficult cases like missing data or outliers easily, and all in a unified manner. Extensive experiments on both synthetic and real image data show comparable or superior results compared with existing methods.

1 Introduction

Tomasi-Kanade factorization [1] is probably one of the most remarkable works in multi-view structure-from-motion (SFM) research. This algorithm is not only of significant theoretical importance, but also of striking elegance and computational simplicity. Given a multi-view measurement matrix $\mathtt{M}$, it simultaneously solves for the (stacked) camera projection matrix $\mathtt{P}$ and the 3D structure $\mathtt{X}$, via a simple *matrix factorization* $\mathtt{M} = \mathtt{PX}$ through a single Singular Value Decomposition (SVD). Its elegance also comes from the fact that it treats all points and all camera frames *uniformly*, no any "privileged" or "preferred" frames and points.

Tomasi-Kanade's factorization algorithm was developed for *affine* camera cases. We revisit the *projective* generalization of the factorization method in this paper. In particular, we are motivated by the most popular algorithm-of-choice for *projective factorization*–the iterative Sturm-Triggs method[1] [2] [3]. Projective

[1] We exclude the non-iterative version of Sturm-Triggs method reported in [2], as it crucially relies on accurate depth estimation from epipolar geometry.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 396–409, 2010.

imaging process can be compactly written as $\mathtt{\Lambda} \odot \mathtt{M} = \mathtt{PX}$ where $\mathtt{\Lambda}$ matrix is a properly stacked but unknown *projective depth matrix*, $\odot$ denotes the element-wise (Hadamard) product. Since both the depths $\mathtt{\Lambda}$ and the right-hand factorization $\mathtt{P}$ and $\mathtt{X}$ are unknowns, a natural approach to solve this is through iterative alternation till converge. Many other projective generalizations, such as [4] [5] [6] share a similar computational pattern in terms of iteration.

Like any other iterative algorithm, the Sturm-Triggs iteration and its extensions have some common drawbacks. For example, iterative algorithms all require a good initial point to start with; the iteration procedure may not converge; even if it does converge (theoretically or empirically), it may only converge to a local minimum; global optimality can hardly be guaranteed.

Unfortunately, for the particular method of Sturm-Triggs iteration and the alikes, all the aforementioned drawbacks did have been observed in all occasions. Indeed, [7] pointed out that the iterative Sturm-Triggs method with row- and column- normalization is not guaranteed to converge in theory. To salvage this, they proposed a column-wise only normalization and derived a provably-convergent iterative algorithm (called *column-space* method) [7]. Oliensis and Hartley also observed situations where the iterations fell into a limiting cycle and never converged [8]. Hartley and Zisserman [9] concluded that the popular choice of initialization–assuming all depths to be one–works only when the ratios of true depths of the different 3D point $\mathbf{X}_j$ remain approximately constant during a sequence. As a result, to make Sturm-Triggs iteration work, the true solution has to be rather close to the affine case.

Even worse, a recent complete theoretical analysis delivers even more negative message [8], which shows that (1) the simplest Sturm-Triggs iteration without normalization (called SIESTA w.o. balancing) will always converge to the trivial solution; (2) paper [10]'s provably-convergent iteration method will generally converge to a useless solution; (3) applying both row-wise and column-wise normalization may possibly run into unstable state during iteration. The authors also provided a remedy, i.e. a regularization-based iterative algorithm (called CIESTA) that can converge to a stable solution, albeit the solution is biased (towards all depths being close to one).

Having mentioned the above negative points, we however argue that Sturm-Triggs algorithm (and its variants) are useful in practice. After a few iterations they often return a much improved and useful result. In fact, many of the issues discussed in [8] are theoretically driven. However when the actual camera-point configuration is far away from affine configuration, Sturm-Triggs algorithm tends to produce a bad solution. It would be nicer if a projective-factorization algorithm can be made free from these theoretical drawbacks and can at the same time be useful in practice.

In this paper, we propose a closed-form solution to projective factorization, which is based on the similar idea of low-rank factorization and stays away from all the above mentioned theoretical traps. Our algorithm is global; no initial guess is needed. Given a complete measurement matrix the result will be globally optimal (at most up to some relaxation gap). Additionally, it deals

with missing data and outliers all in unified framework. The outlier-extension of our method has an intimate connection with the recent proposed Compressive Sensing theory and algorithms. Nevertheless, the main theory and algorithms of our method stand independently, and do not depend on compressive sensing theory.

2 Element-Wise Factorization

2.1 Preliminaries

Consider n stationary 3D points $\mathbf{X}_j = [x_j, y_j, z_j, 1]^T, j = 1, \cdots, n$ observed by all m projective cameras $\mathtt{P}_i, i = 1, \cdots, m$. Under projective camera model, the j-th 3D point $\mathbf{X}_j$ is projected onto image point $\mathbf{m}_{ij} = [u_{ij}, v_{ij}, 1]^T$ by $\mathbf{m}_{ij} = \frac{1}{\lambda_{ij}}\mathtt{P}_i\mathbf{X}_j$, where λ_{ij} is a scale factor, commonly called "projective depth" [9]. It is easy to see that $\lambda_{ij} = 1$ when the camera reduces to an affine camera.

Collecting all the image measurements over all frames, we form a *measurement matrix* $\mathtt{M} = [\mathbf{m}_{ij}]$ of size $(3m \times n)$. Now the above relationship is compactly written as

$$\mathtt{M} = \left[(\frac{1}{\lambda_{ij}})\right]_{\sharp} \odot (\mathtt{PX}), \tag{1}$$

where $\mathtt{P} \in \mathbb{R}^{3m\times 4}$ and $\mathtt{X} \in \mathbb{R}^{4\times n}$ are properly stacked projection matrix and structure matrix. Note that each row of the inverse depth matrix is repeated 3 times. We use a subscript of "$\sharp$" to denote such a triple copy.

Define $\mathtt{W} = \mathtt{PX}$ as the **rescaled (re-weighted) measurement matrix**, we can equivalently re-write the above equation as:

$$\mathtt{W} = \Lambda \odot \mathtt{M} = \mathtt{PX}, \tag{2}$$

where $\Lambda = [(\lambda_{ij})]_{\sharp} \in \mathbb{R}^{3m\times n}$, i.e. a triple copy. As seen from the equation, matrix $\mathtt{W}$ must have a rank at most 4.

The problem of projective factorization seeks to simultaneously solve for the unknown depths Λ, the unknown cameras $\mathtt{P}$ and the unknown structure $\mathtt{X}$. Compared with affine factorization, this is a much harder problem, mainly because depths are not known *a priori*. Of course, one could compute these projective depths beforehand, by other means, e.g, via fundamental matrices or trifocal tensors, via a common reference plane, etc. However, such approaches diminish the elegance of the factorization algorithm, as they no longer treat points and frames uniformly.

The Sturm-Triggs type iterative algorithms solve the problem through *alternation*: (1) fix Λ, solve for $\mathtt{P}$ and $\mathtt{X}$ via SVD factorization; (2) fix $\mathtt{P}$ and $\mathtt{X}$, solve for Λ via least squares; (3) Alternate between the above two steps till convergence. Usually, to avoid possible trivial solutions (e.g., all depths being zero, or all but 4 columns of the depth matrix are zeros, etc.), some kind of row-wise and column-wise normalization (a.k.a *balancing*) is necessary.

2.2 Element-Wise Factorization

As we have explained earlier, though many of the existing iterative projective factorization algorithms do produce sufficiently good results, there is however, no any theoretic justification. In other words, the optimality of such iteration procedures is not guaranteed. In this subsection, we present a closed-form solution to projective factorization.

Our main idea. We repeat the basic equation here: $\mathtt{W} = \Lambda \odot \mathtt{M} = \mathtt{PX}$. Recall that $\mathtt{M}$ is the only input, and the task is to solve for both Λ and $\mathtt{W}$. Note that $\odot$ denotes element-wise product, therefore we can view the problem as an *element-wise factorization* problem, in the following sense:

– Given measurement matrix $\mathtt{M}$, find two matrices Λ and $\mathtt{W}$ such that $\mathtt{W} = \Lambda \odot \mathtt{M}$.

At a first glance, this seems to be an impossible task, as the system is severely under-constrained. However, for the particular problem of projective factorization, we have extra conditions on the unknown matrices which may sufficiently constrain the system. Roughly speaking, these extra conditions (to be listed below) are expected to supply the system with sufficient constraints, making the element-wise factorization problem well-posed and hence solvable. These constraints are in fact very mild, reasonable, and not restrictive.

– All visible points' projective depths must be positive. This is nothing but the well-known and very common *cheirality* constraint. In other words, visible point must lie in front of the camera.
– The re-scaled measurement matrix $\mathtt{W}$ has rank at most 4. This is true for noise-free case (we will further relax this in the actual computation).
– The rank of Λ is also at most 4. This is easy to see, since $[\lambda_{ij}]$ is a sub-matrix of $\mathtt{W}$; hence, $\mathrm{rank}(\mathtt{W}) \leq 4 \Rightarrow \mathrm{rank}(\Lambda) \leq 4$.
– All the rows and columns of Λ have been normalized to have (average) unit sum. The row-sum and column-sum constraints play two roles: (1) rule out trivial solutions;(2) rule out scale ambiguity in the factorization.

Formulation. Mathematically, the element-wise factorization is formulated as:

$$\begin{aligned} &\mathtt{Find\ W}, \Lambda, \ \mathtt{such\ that,} \\ &\mathtt{W} = \Lambda \odot \mathtt{M}, \\ &\mathrm{rank}(\mathtt{W}) \leq 4, \\ &\textstyle\sum_i \lambda_{ij} = m, j = 1, \cdots, n, \\ &\textstyle\sum_j \lambda_{ij} = n, i = 1, \cdots, m, \\ &\lambda_{ij} > 0. \end{aligned} \tag{3}$$

There is one more theoretical issue left, however. One would ask: will the unit row-sum and column-sum constraints be too restrictive such that no feasible solution of Λ matrix can be found? This is a reasonable question to ask, because in the traditional iterative projective factorization scenarios, Mahamud et al [7] and Oliensis et al [8] both showed that applying the row-wise and column-wise normalization during the iteration may hinder the convergence.

However, we show that this is not a problem at all for our algorithm, thank to Sinkhorn's famous theorem regarding *doubly stochastic matrix* [11]. A square nonnegative matrix is called doubly stochastic if the sum of the entries in each row and each column is equal to one. Sinkhorn proved the following theorem, which gives the diagonal equivalence between doubly stochastic matrix and any arbitrary positive matrix.

Theorem 1. *[11] Any strictly positive matrix* $\mathtt{A}$ *of order* n *can always be normalized into a doubly stochastic matrix by the following diagonal scaling,* $\mathtt{D}_1\mathtt{A}\mathtt{D}_2$*, where* $\mathtt{D}_1$ *and* $\mathtt{D}_2$ *are diagonal matrices of order* n *with strictly positive diagonal entries. Such two diagonal matrices are unique up to scale for a given positive matrix* $\mathtt{A}$*.*

This theorem can be naturally generalized to non-square positive matrices, and we have the following result:

Corollary 1. *[12] Any strictly positive matrix* $\mathtt{A}$ *of size* $m \times n$ *can always be rescaled to* $\mathtt{D}_1\mathtt{A}\mathtt{D}_2$ *whose row-sums all equal to* n *and column-sums all equal to* m*, where* $\mathtt{D}_1$ *and* $\mathtt{D}_2$ *are respectively* $m \times m$ *and* $n \times n$ *positive diagonal matrices. Such* $\mathtt{D}_1$ *and* $\mathtt{D}_2$ *are unique up to scale for any given* $\mathtt{A}$*.*

In our context, this result suggests that the row-wise and column-wise normalization conditions are not restrictive, because the entire set of (positive) projective depth matrices is reachable from a row- and column-normalized positive matrix. Furthermore, in Appendix we will show that for general configurations, the rank = 4 constraint provides sufficient constraints for solving the problem.

3 Implementation

3.1 Rank Minimization

Noise is inevitable in real measurements, which will consequently increase the actual rank of $\mathtt{W}$. To accommodate noise, we slightly modify the problem formulation, and pose it as a rank minimization problem:

$$\begin{aligned} &\texttt{Minimize } \mathrm{rank}(\mathtt{W}), \texttt{ subject to},\\ &\mathtt{W} = \mathtt{\Lambda} \odot \mathtt{M},\\ &\textstyle\sum_i \lambda_{ij} = m, j = 1, \cdots, n,\\ &\textstyle\sum_j \lambda_{ij} = n, i = 1, \cdots, m,\\ &\lambda_{ij} > 0. \end{aligned} \tag{4}$$

Once the problem is solved, we can use $\mathtt{\Lambda}$ as the estimated depth matrix, and $\mathtt{W}$ as the rescaled measurement matrix. Subsequently they can be fed into a single SVD, or be used to initialize a bundle adjustment process.

3.2 Trace Minimization

To solve rank minimization problem exactly is intractable in general [13]. To overcome this, *nuclear-norm* has been introduced as the tightest convex surrogate of rank. The nuclear norm of $\mathtt{X} \in \mathbb{R}^{m\times n}$ is defined as $\|\mathtt{X}\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i$, where σ_i is the ith singular value of $\mathtt{X}$.

Recently, using nuclear norm minimization to solve rank minimization problem has received considerable attention, in particular in the research of *compressive sensing*. One surprising result is that for a large class of matrices satisfying some "incoherency" or "restricted isometry" properties, nuclear norm minimization actually gives an exact solution. In other words, the relaxation gap is zero.

In this paper, we simply use the nuclear norm only as a convex surrogate (a relaxation) to the rank function, mainly for the purpose of approximately solving our projective factorization problem. Our main contribution of this work lies more in the new element-wise factorization formulation, than the actual computational implementation. We however appreciate the significance of the results due to compressive sensing. Thanks to these results, we at least can say, our algorithm may produce the exact and globally optimal solution, when certain conditions are satisfied.

Using the nuclear norm, we replace the original objective function $\text{rank}(\mathtt{W})$ with $\|\mathtt{W}\|_*$. Furthermore, the nuclear norm minimization $\min \|\mathtt{W}\|_*$ can be rewritten as an equivalent SDP (semi-definite programming) problem:

$$\begin{aligned} &\min_{\mathtt{W}} \frac{1}{2}(\text{tr}(\mathtt{X}) + \text{tr}(\mathtt{Y})) \\ &\text{s.t.} \begin{pmatrix} \mathtt{X} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{Y} \end{pmatrix} \succeq 0 \end{aligned}$$

Such an equivalence is grounded on the following theorem (ref. [14]).

Theorem 2. *Let* $\mathtt{A} \in \mathbb{R}^{m \times n}$ *be a given matrix, then* $\text{rank}(\mathtt{A}) \leq r$ *if and only if there exist two symmetric matrices* $\mathtt{B} = \mathtt{B}^\top \in \mathbb{R}^{m \times m}$ *and* $\mathtt{C} = \mathtt{C}^\top \in \mathbb{R}^{n \times n}$ *such that* $\text{rank}(\mathtt{B}) + \text{rank}(\mathtt{C}) \leq 2r$ *and* $\begin{bmatrix} \mathtt{B} & \mathtt{A} \\ \mathtt{A}^\top & \mathtt{C} \end{bmatrix} \succeq 0$.

Piecing everything together, we finally reach a **trace minimization** problem:

$$\begin{aligned} &\min_{\mathtt{W}} \frac{1}{2}(\text{tr}(\mathtt{X}) + \text{tr}(\mathtt{Y})) \\ &\text{s.t.} \quad \begin{pmatrix} \mathtt{X} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{Y} \end{pmatrix} \succeq 0 \\ &\mathtt{W} = \mathtt{\Lambda} \odot \mathtt{M}, \\ &\textstyle\sum_i \lambda_{ij} = m, j = 1, \cdots, n, \\ &\textstyle\sum_j \lambda_{ij} = n, i = 1, \cdots, m, \\ &\lambda_{ij} > 0. \end{aligned} \tag{5}$$

This is a standard semi-definite programming (SDP), thus can be solved efficiently using any off-the-shelf SDP solvers. In all our experiments, we simply used SeDumi and SDPT3 [15] as the solvers, mainly for theory-validation purpose. Note however that, these state-of-the-art SDP solvers still cannot solve large scale problems, due to excessive memory and computational requirement. A better choice is those fast algorithms specially designed for large-scale nuclear norm minimization problems, and many of them can be found in recent compressive sensing literature (see e.g. [16],[17]).

4 Extensions

4.1 Dealing with Missing Data

In most real-world structure-from-motion applications, *missing data* are inevitable, due to e.g. self-occlusion, points behind cameras (i.e., cheirality) etc. Missing data lead to an incomplete measurement matrix, but simple SVD cannot directly perform on an incomplete matrix. This constitutes a major drawback of factorization-based methods.

For affine (camera) factorization, many missing data handling ideas have been proposed. Buchanan and Fitzgibbon [18] summarized existing methods, and classified them into four categories (1) closed form method,(2) imputation method, (3)alternation method and (4) direct nonlinear minimization.

Unfortunately, relatively less works were reported for projective factorization with missing data. A few related works are e.g. [19] [20] [21] [22]. Most existing works either rely on iteration or alternation, or assume the depths are precomputed by other means (reducing to affine case).

Our new element-wise factorization, on the other hand, offers a unified treatment to the missing data problem. With little modification, our SDP formulation can be extended to solve both complete case and missing data case. A similar work was reported elsewhere but is restricted to affine camera model [23].

Given an incomplete measurement matrix $\mathtt{M} = [\mathbf{m}_{ij}]$ with missing data, we define a 0-1 *mask matrix* Ω as

$$\Omega = [\omega_{ij}], \ \ \mathtt{where}\ \ \omega_{ij} = \begin{cases} \mathbf{1} \in \mathbb{R}^3, & \mathtt{if}\ \ \mathbf{m}_{ij} \text{ is available,} \\ \mathbf{0} \in \mathbb{R}^3, & \mathtt{if}\ \ \mathbf{m}_{ij} \text{ is missing.} \end{cases} \tag{6}$$

With these notations, the projective imaging process with missing data can be written as:

$$\Lambda \odot \mathtt{M} = \Omega \odot \mathtt{W}.$$

Now our task becomes:

– Given an incomplete $\mathtt{M}$, find a completed low-rank $\mathtt{W}$ such that $\Lambda \odot \mathtt{M} = \Omega \odot \mathtt{W}$.

Note that at those missing positions, we do not need to estimate the corresponding depths, so we set $\lambda_{ij} = 1$ whenever $\omega_{ij} = \mathbf{0}$.

Applying the nuclear norm heuristics, our SDP formulation for projective factorization with missing data is:

$$\begin{array}{l} \min\limits_{\mathtt{W}} \dfrac{1}{2}(\mathrm{tr}(\mathtt{X}) + \mathrm{tr}(\mathtt{Y})) \\ \text{s.t.} \quad \begin{pmatrix} \mathtt{X} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{Y} \end{pmatrix} \succeq 0, \\ \Lambda \odot \mathtt{M} = \Omega \odot \mathtt{W}, \\ \lambda_{ij} > 0, \ \text{if } \omega_{ij} = \mathbf{1}, \\ \lambda_{ij} = 1, \ \text{if } \omega_{ij} = \mathbf{0}, \\ \sum_i \lambda_{ij} = m, j = 1, \cdots, n, \\ \sum_j \lambda_{ij} = n, i = 1, \cdots, m. \end{array} \tag{7}$$

Once this SDP converges, the resultant $\mathtt{W}$ is a completed $3m \times n$ full matrix with no entries missing. Moreover, we can even read out a *completed* projective depth matrix just as the sub-matrix of $\mathtt{W}$ formed by every the third rows of $\mathtt{W}$.

4.2 Dealing with Pure Outliers

Another recurring practical issue in real SFM applications is the outlier problem. Different from the missing data case, for the outlier case, we know that some of the entries of the given measurement matrix $\mathtt{M}$ are contaminated by gross errors (i.e., wrong matches), but we do not know where they are. We assume there are only a small portion of outliers and they are randomly distributed in $\mathtt{M}$, in other words, the outliers are sparse.

By conventional factorization methods, there is no easy and unified way to deal with outliers. Most published works are based on some pre-processing using RANSAC [19]. However, we show how our element-wise factorization formulation can handle this problem nicely and uniformly, if certain compressive sensing conditions are satisfied ([24],[25]).

Denote the actual measurement matrix as $\mathtt{M}$, which contains some outliers at unknown positions. Denote the underlying outlier-free measurement matrix as $\hat{\mathtt{M}}$. Then we have $\mathtt{M} = \hat{\mathtt{M}} + \mathtt{E}$, where $\mathtt{E}$ gives the outlier pattern. Now, list the basic projective imaging equation as $\mathtt{W} = \mathtt{\Lambda} \odot (\mathtt{M} - \mathtt{E})$, the task is to simultaneously find the optimal $\mathtt{W}$,$\mathtt{\Lambda}$ and the outlier pattern $\mathtt{E}$, such that $\mathtt{W}$ has the lowest rank and $\mathtt{E}$ is as sparse as possible. To quantify the sparseness of $\mathtt{E}$, we use its element L_1-norm $\|\mathtt{E}\|_1 = \sum_{i,j} |\mathtt{E}_{i,j}|$ as a relaxation of its L_0-norm[2]. Combined with the nuclear-norm heuristics for $\mathtt{W}$, the objective function is chosen as $\|\mathtt{W}\|_* + \mu\|\mathtt{E}\|_1$, where μ is a trade-off parameter (we used 0.4 in our experiments).

The final minimization formulation for factorization with outliers becomes:

$$\begin{aligned} &\min_{\mathtt{W},\mathtt{E}} \frac{1}{2}(\mathrm{tr}(\mathtt{X}) + \mathrm{tr}(\mathtt{Y})) + \mu\|\mathtt{E}\|_1 \\ &\text{s.t.} \quad \begin{pmatrix} \mathtt{X} & \mathtt{W} \\ \mathtt{W}^\top & \mathtt{Y} \end{pmatrix} \succeq 0 \\ &\mathtt{W} = \mathtt{\Lambda} \odot (\mathtt{M} - \mathtt{E}) \\ &\textstyle\sum_i \lambda_{ij} = m, j = 1, \cdots, n \\ &\textstyle\sum_j \lambda_{ij} = n, i = 1, \cdots, m \\ &\lambda_{ij} > 0, \forall i, j. \end{aligned} \tag{8}$$

It is worth noting that, in the presence of missing data or outliers, the performance of our algorithm is problem-dependent. The ratio and the (spatial) distribution of outliers or missing data all affect the final result. But this is also true for most other algorithms.

5 Experimental Results

To evaluate the performance of the proposed method, we conducted extensive experiments on both synthetic data and real image data. We tested complete

[2] We use the fact that $\|\mathtt{E}\|_0 = \|\mathtt{\Lambda} \odot \mathtt{E}\|_0$, since $\lambda_{ij} > 0$.

measurement case, as well as missing data case and outlier case. Reprojection error in the image plane and relative projective depth error (if the ground-truth is known) are used to evaluate the algorithm performance.

Relative depth error is defined as follows

$$\varepsilon = \frac{\|\hat{\Lambda}_{GT} - \hat{\Lambda}_{Recover}\|}{\|\hat{\Lambda}_{GT}\|}, \tag{9}$$

where $\hat{\Lambda}_{GT}$ is the ground truth projective depth matrix after column-sum and row-sum balancing and $\hat{\Lambda}_{Recover}$ is the projective depth matrix recovered after balancing.

5.1 Synthetic Experiments

In all the synthetic experiments, we randomly generated 50 points within a cube of $[-30, 30]^3$ in space, while 10 perspective cameras were simulated. The image size is set as 800×800. The camera parameters are set as follows: the focal lengths are set randomly between 900 and 1100, the principal point is set at the image center, and the skew is zero. We added realistic Gaussian noise to all simulated measurements.

We first tested for the complete measurement case, i.e., the input measurement matrix `M` is complete. In all of our experiments, the SDP solver output results in less than 20 iterations (even including the experiments for missing data and outlier cases), and cost less than 0.5 seconds per iteration on a modest 1.6GHz Core-Duo laptop with memory 2GB using SDPT3 as solver.

Synthetic images: large depth variations. We simulated cases where the true depths are widely distributed and not close to one, which are commonly encountered in real world applications of structure from motion especially in large scale reconstructions.

We defined the depth variation as $r = \max_{ij}(\lambda_{ij}) / \min_{ij}(\lambda_{ij})$, i.e. the ratio between the maximal depth and the minimal depth. We tested two cases, one is that all the depth variations are within [1,5), the other is that all the depth variations are within [5,20]. As we expect, our method outperforms all state-of-the-art iterative methods by a significant margin. Figure 1 illustrates error histograms for the two cases. From Fig. 1(a) and Fig. 1(c), we observe that our algorithm produces reprojection error less than 2 pixels while SIESTA (with balancing) outputs reprojection error up to 100 pixels for small and modest depth variations. From Fig. 1(b) and Fig. 1(d), we observe that our algorithm produces reprojection error less than 14 pixels while SIESTA outputs error up to 250 pixels for large depth variation. This can be explained that our method is a closed-form solution and does not depend on initialization. However all other algorithms highly depend on initialization, where affine camera model is widely used as initialization which is not the case for large depth variation.

Synthetic images: missing data. To evaluate the performance of our algorithm on measurements with missing data, we generated synthetic data sets

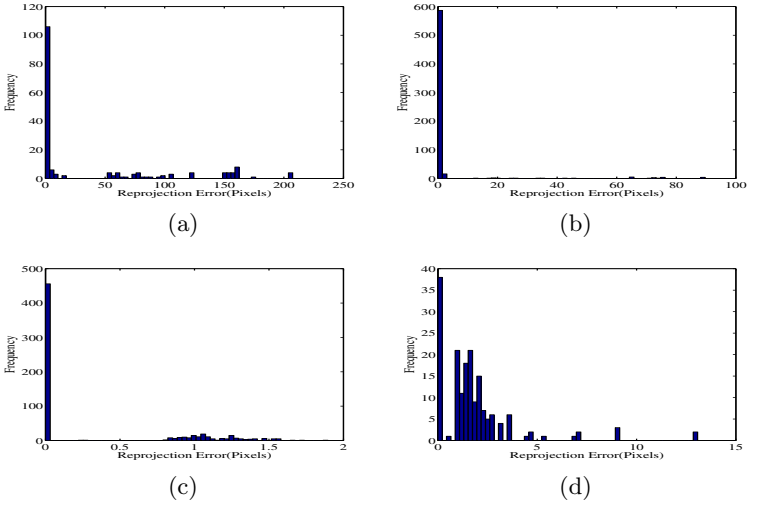


Fig. 1. Performance comparison between SIESTA and proposed method under various level of depth variations. Clearly, our method is much more superior. (a) Histogram of reprojection error by the normalized SIESTA ($r < 5$); (b) Histogram of reprojection error by the normalized SIESTA ($5 \leq r < 20$); (c) Histogram of reprojection error by our method ($r < 5$); (d) Histogram of reprojection error by our method ($5 \leq r < 20$).

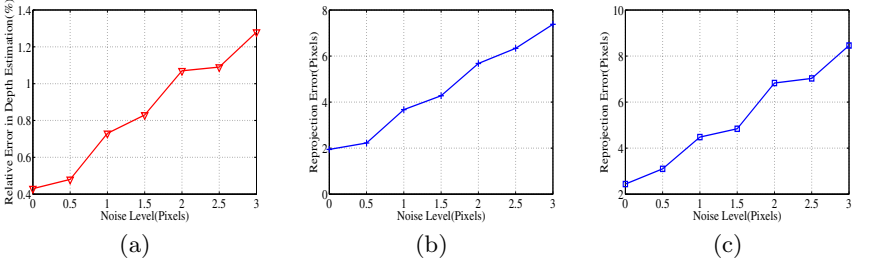


Fig. 2. Performance of the proposed method for missing data case. (a) Relative error in depth estimation under various level of Gaussian noise. (b) Reprojection error on visible points; (c) Reprojection error on all points.

with dimension 20×50 as before followed by removing 20% of 2D points in the measurement matrix randomly to simulate missing data case. The relative depth error and reprojection error for both visible points and missing points are plotted against different Gaussian noise levels in Fig. 2.

Synthetic images: outliers. To illustrate the performance of our algorithm for projective factorization with outliers, we generated the following illustrative example. The configuration is 10 cameras observing 20 points leading to measurement matrix of 20×20. The outlier pattern is generated according to uniform distribution with 5% positions are outliers. The results are shown in

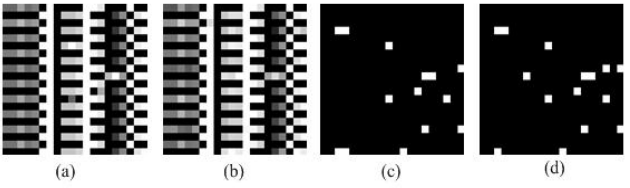


Fig. 3. Performance of the proposed method dealing with outliers in measurement matrix. (a) Input Measurement Matrix. (b) Recovered Measurement Matrix. (c) Recovered Outlier Pattern. (d) Ground Truth Outlier Pattern where white denotes outlier.

Fig. 4. Real image sequences used in experiments. (a) Corridor, (b) Teabox, (c) Chair.

Fig. 3. From the figure, we conclude that our method recovers the outlier pattern successfully.

5.2 Real Image Experiments

Real images: complete data. We first tested our method on real images with complete measurement. Some of the real images used in our experiments are shown in Fig. 4. Reprojection errors for these images are shown in Table 1.

Real images: missing data. We tested our method on real images with missing data. A small portion of the Dinosaur data with dimension 18×20 is used, as our current SDP solver can only solve toy-size problems. The Dinosaur sequence [18] is conventionally used as an example for affine factorization. Here we however solve it as a projective factorization with missing data problem. Our experiment is mainly for theory validation purpose. Fig. 5 illustrates the effect of our projective depth estimation at a 4×4 image patch.

Table 1. Performance Evaluation for Real Images with Complete Measurements

Dataset / Method	SIESTA [8]	CIESTA [8]	Col-space [7]	Our method
Corridor	0.3961	0.3955	0.3973	0.3961
Teabox	4.4819e-4	4.4872e-4	4.8359e-4	4.4819e-4
Chair	1.3575	1.3723	1.3441	1.3385

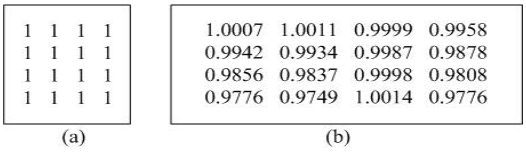


Fig. 5. Depth estimation from affine factorization and projective factorization. (a) Affine camera sets all the depths to be 1s (b) Depths estimated by our method.

6 Conclusion

In this paper, we have proposed a new element-wise factorization framework for non-iterative projective reconstruction. We formulate the problem as an SDP and solve it efficiently and (approximately) globally. Our results are comparable or superior to other iterative methods when these methods work. When they no longer work, ours still works.

Future work will address drawbacks of the current implementation, in particular the scalability issue of the standard SDP solver. We will also consider non-rigid deformable motion, degenerate cases, and cases combining missing data and outliers. Theoretical analysis about our missing-data and outlier-handling procedure is also planned.

Acknowledgements

The first author appreciated the China Scholarship Council for supporting his visit to ANU (Oct 2008 to Oct 2009). The authors would like to thank Prof. Richard Hartley for his immeasurable discussion. This work was supported partially by National Natural Science Foundation of China under key project 60736007, Natural Science Foundation of Shaanxi Province under 2010JZ011, ARC through its SRI in Bionic Vision Science and Technology grant to Bionic Vision Australia. NICTA is funded by the Australian Government as represented by the DBCDE and by the ARC.

References

1. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vision 9, 137–154 (1992)
2. 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)
3. Triggs, B.: Factorization methods for projective structure and motion. In: CVPR, pp. 845–851. IEEE Computer Society, Los Alamitos (1996)
4. Ueshiba, T., Tomita, F.: A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 296–310. Springer, Heidelberg (1998)

5. Heyden, A., Berthilsson, R., Sparr, G.: An iterative factorization method for projective structure and motion from image sequences. Image Vision Comput. 17, 981–991 (1999)
6. Chen, Q., Medioni, G.: Efficient iterative solution to m-view projective reconstruction problem. In: CVPR, pp. 55–61 (1999)
7. Mahamud, S., Hebert, M.: Iterative projective reconstruction from multiple views. In: CVPR, pp. 430–437 (2000)
8. Oliensis, J., Hartley, R.: Iterative extensions of the Sturm/Triggs algorithm: Convergence and nonconvergence. IEEE Trans. Pattern Anal. Mach. Intell. 29, 2217–2233 (2007)
9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
10. Mahamud, S., Hebert, M., Omori, Y., Ponce, J.: Provably-convergent iterative methods for projective structure from motion. In: CVPR, pp. 1018–1025 (2001)
11. Sinkhorn, R.: A relationship between arbitrary positive matrices and doubly stochastic matrices. Ann. Math. Statist., 876–879 (1964)
12. Sinkhorn, R.: Diagonal equivalence to matrices with prescribed row and column sums. In: The American Mathematical Monthly, pp. 402–405 (1967)
13. Recht, B., Fazel, M., Parrilo, P.: Guarranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. Technical report, California Institute of Technology (2007)
14. Fazel, M., Hindi, H., Boyd, S.: A rank minimization heuristic with application to minimum order system approximation. In: Proc. Am. Control Conf., pp. 4734–4739 (2001)
15. Toh, K., Todd, M., Tutuncu, R.: Sdpt3 — a matlab software package for semidefinite programming. Optimization Methods and Software, 545–581 (1999)
16. Cai, J.F., Candes, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion (2008), http://arxiv.org/abs/0810.3286
17. Ma, S., Goldfarb, D., Chen, L.: Fixed point and bregman iterative methods for matrix rank minimization. Math. Program., Ser. A (2009)
18. Buchanan, A.M., Fitzgibbon, A.W.: Damped newton algorithms for matrix factorization with missing data. In: CVPR, pp. 316–322 (2005)
19. Martinec, D., Pajdla, T.: Structure from many perspective images with occlusions. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 355–369. Springer, Heidelberg (2002)
20. Tang, W., Hung, Y.: A subspace method for projective reconstruction from multiple images with missing data. Image Vision Comput 24, 515–524 (2006)
21. Martinec, D., Pajdla, T.: 3D reconstruction by fitting low-rank matrices with missing data. In: CVPR, pp. 198–205 (2005)
22. Martinec, D., Pajdla, T.: Robust rotation and translation estimation in multiview reconstruction. In: CVPR, pp. 1–8 (2007)
23. Olsson, C., Oskarsson, M.: A convex approach to low rank matrix approximation with missing data. In: SCIA, pp. 301–309 (2009)
24. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis (2009), http://arxiv.org/abs/0912.3599
25. Chandrasekaran, V., Sanghavi, S., Parrilo, P., Willsky, A.: Rank-sparsity incoherence for matrix decomposition (2009), http://arxiv.org/abs/0906.2220v1

Appendix

In this appendix, we will show that our formulation (i.e., (5)) is well-posed, meaning that any true solution is indeed the exact unique solution of the formulation. We assume the cameras and points are generically configured, image measurements are complete and noise-free—hence the rank is identically 4.

In the main body of our paper (subsection-2.2), we have already shown that applying the row- and column- normalization places no restriction to the solution space, given that all (visible) projective depths are positive.

Next, we need to show that enforcing the two rank conditions on both $\mathtt{W}$ and Λ provides sufficient constraints for solving the element-wise factorization problem.

Denote the image coordinate as $\mathbf{m}_{ij} = [u_{ij}, v_{ij}, 1]^T$, the projective depth as λ_{ij}, according to the rank-4 constraint on projective depth matrix Λ, the jth column of Λ is expressed as

$$\Lambda_j = a_{1j}\Lambda_1 + a_{2j}\Lambda_2 + a_{3j}\Lambda_3 + a_{4j}\Lambda_4.$$

Since $\mathtt{W}$ is expected to have rank 4 in general case, we have

$$\mathtt{W}_j = b_{1j}\mathtt{W}_1 + b_{2j}\mathtt{W}_2 + b_{3j}\mathtt{W}_3 + b_{4j}\mathtt{W}_4, \tag{10}$$

where $\mathtt{W}_j, j = 5, \cdots, n$ denotes the $j$the column of $\mathtt{W}$.

Substitute the image coordinates into the above equations, we have

$$\begin{aligned}
u_{ij}(a_{1j}\lambda_{i1} + a_{2j}\lambda_{i2} + a_{3j}\lambda_{i3} + a_{4j}\lambda_{i4}) &= b_{1j}u_{i1}\lambda_{i1} + b_{2j}u_{i2}\lambda_{i2} + b_{3j}u_{i3}\lambda_{i3} + b_{4j}u_{i4}\lambda_{i4} \\
v_{ij}(a_{1j}\lambda_{i1} + a_{2j}\lambda_{i2} + a_{3j}\lambda_{i3} + a_{4j}\lambda_{i4}) &= b_{1j}v_{i1}\lambda_{i1} + b_{2j}v_{i2}\lambda_{i2} + b_{3j}v_{i3}\lambda_{i3} + b_{4j}v_{i4}\lambda_{i4} \\
a_{1j}\lambda_{i1} + a_{2j}\lambda_{i2} + a_{3j}\lambda_{i3} + a_{4j}\lambda_{i4} &= b_{1j}\lambda_{i1} + b_{2j}\lambda_{i2} + b_{3j}\lambda_{i3} + b_{4j}\lambda_{i4},
\end{aligned}$$

which implies that $a_{1j} = b_{1j}, a_{2j} = b_{2j}, a_{3j} = b_{3j}, a_{4j} = b_{4j}$. This can be explained as the rank-4 constraint on Λ is included under the rank-4 constraint on $\mathtt{W}$.

Then we have

$$\frac{u_{ij}}{v_{ij}} = \frac{b_{1j}\lambda_{i1}u_{i1} + b_{2j}\lambda_{i2}u_{i2} + b_{3j}\lambda_{i3}u_{i3} + b_{4j}\lambda_{i4}u_{i4}}{b_{1j}\lambda_{i1}v_{i1} + b_{2j}\lambda_{i2}v_{i2} + b_{3j}\lambda_{i3}v_{i3} + b_{4j}\lambda_{i4}v_{i4}}$$

Let $\eta_{ij} = \frac{u_{ij}}{v_{ij}}$, we obtain

$$\lambda_{i1}b_{1j}(u_{i1} - \eta_{ij}v_{i1}) + \lambda_{i2}b_{2j}(u_{i2} - \eta_{ij}v_{i2}) + \lambda_{i3}b_{3j}(u_{i3} - \eta_{ij}v_{i3}) + \lambda_{i4}b_{4j}(u_{i4} - \eta_{ij}v_{i4}) = 0$$

There are $m(n-4)$ equations while the number of variables is $4m+4(n-4)$, thus the problem is well-posed. In our SDP implementation, we use "min (rank)" (as opposed to enforcing a hard constraint of "rank=4") to solve a relaxed version.

Learning Relations among Movie Characters: A Social Network Perspective

Lei Ding and Alper Yilmaz

Photogrammetric Computer Vision Lab
The Ohio State University
dinglei@cse.ohio-state.edu, yilmaz.15@osu.edu

Abstract. If you have ever watched movies or television shows, you know how easy it is to tell the good characters from the bad ones. Little, however, is known "whether" or "how" computers can achieve such high-level understanding of movies. In this paper, we take the first step towards learning the relations among movie characters using visual and auditory cues. Specifically, we use support vector regression to estimate local characterization of adverseness at the scene level. Such local properties are then synthesized via statistical learning based on Gaussian processes to derive the affinity between the movie characters. Once the affinity is learned, we perform social network analysis to find communities of characters and identify the leader of each community. We experimentally demonstrate that the relations among characters can be determined with reasonable accuracy from the movie content.

1 Introduction

During recent years, researchers have devoted countless efforts on object detection and tracking to understand the scene content from motion patterns in videos [12,7,1,6]. Most of these efforts, however, did not go beyond analyzing or grouping trajectories, or understanding individual actions performed by tracked objects [11,8,2]. The computer vision community, generally speaking, did not consider analyzing the video content from a sociological perspective, which would provide systematic understanding of the roles and social activities performed by actors based on their relations.

In sociology, the social happenings in a society are conjectured to be best represented and analyzed using a social network structure [22]. The social network structure provides a means to detect and analyze communities in the network, which is one of the most important problems studied in modern sociology. The communities are generally detected based on the connectivity between the actors in a network. In context of surveillance, a recent research reported in [23] takes advantage of social networks to find such communities. The authors use a proximity heuristic to generate a social network, which may not necessarily represent the *social structure* in the scene. The communities in the network are then detected using a common social network analysis tool referred to as the *modularity algorithm* [17]. In a similar fashion, authors of [10] generate social

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 410–423, 2010.

relations based on the proximity and relative velocity between the actors in a scene, which are later used to detect groups of people in a crowd by means of clustering techniques.

In this paper, we attempt to construct social networks, identify communities and find the leader of each community in a video sequence from a sociological perspective using computer vision and machine learning techniques. Due to the availability of visual and auditory information, we chose to perform the proposed techniques on theatrical movies, which contain recordings of social happenings and interactions. The generality of relations among the characters in a movie introduces several challenges to analysis of the movie content: (1) it is not clear which actors act as the key characters; (2) we do not know how the low-level features relate to relations among characters; (3) no studies have been carried on how to synthesize high-level relational information from local visual or auditory cues from movies.

In order to address these challenges, our approach first aligns the movie script with the frames in the video using closed captions [5]. We should note that, the movie script is used only to segment the movie into scenes and provide a basis for generating the *scene-character relation matrix*. Alternatively, this information can be obtained using video segmentation [24] and face detection and recognition techniques [3]. A unique characteristic of our proposed framework is its applicability to an *adversarial social network*, which is a highly recognized but less researched topic in sociology [22], possibly due to the complexity of defining adversarial relations alongside friendship relations. Without loss of generality, an adversarial social network contains two disjoint rival communities $C_1 \bigcup C_2 = \{c_1, c_2, \cdots, c_N\}$ composed of actors, where members within a community have friendly relations and across communities have adversarial relations. In our framework, we use visual and auditory information to quantify adverseness at the scene level, which serves as soft constraints among the movie characters. These soft constraints are then systematically integrated to learn inter-character affinity. The adverse communities in the resulting social network are discovered by subjecting the inter-character affinity matrix to a generalized modularity principle [4], which is shown to perform better than the original modularity [17]. Social networks commonly contain leaders who have the most important roles in their communities. The importance of an actor is quantified by computing degree, closeness, or betweenness centralities [9]. More recently, eigenvector centrality has been proposed as an alternative [19]. In this paper, due to its intrinsic relation to the proposed learning mechanism, we adopt the eigenvector centrality to find leaders in the two adverse communities. An illustration of the communities and their leaders discovered by our approach is given in Figure 1 for the movie titled *G.I. Joe: The Rise of Cobra (2009)*.

The remainder of the paper is organized as follows. We start with providing the basics of the social network framework in the next section, which is followed by a discussion on how we construct the social networks from movies in Section 3. The methodology used to analyze these social networks is described in Section

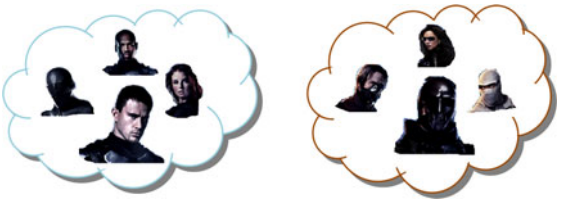


Fig. 1. Pictorial representation of communities in the movie titled *G.I. Joe: The Rise of Cobra (2009)*. Our approach automatically detects the two rival communities (*G.I. Joe* and *Cobra*), and identifies their leaders (*Duke* and *McCullen*) visualized as the upscaled characters at the front of each community.

4, and is evaluated on a set of movies in Section 5. Our contributions in this paper can be summarized as follows:

- Proposal of principled methods for learning adversarial and non-adversarial relations among actors, which is new to both computer vision and sociology communities;
- Understanding these relations using a modified modularity principle for social network analysis;
- A dataset of movies, which contain scripts, closed captions and visual and auditory features, for further research in high-level video understanding.

2 Social Network Representation

Following a common practice in sociology, we define interactions between the characters in a movie using a social network structure. In this setting, the characters are treated as the vertices $V = \{v_i : v_i \text{ represents } c_i\}$[1] with cardinality $|V|$ and their interactions are defined as edges $E = \{(v_i, v_j)|v_i, v_j \in V\}$ between the vertices in a graph $G(V, E)$. The resulting graph G is a fully-connected weighted graph with an affinity matrix K of size $|V| \times |V|$.

In this social setting, the characters may have either adversarial or non-adversarial relations with each other. These relations can be exemplified between the characters in a war movie as non-adversarial (collaborative) within respective armies, and adversarial (competing) across the armies. Sociology and computer vision researchers often neglect adversarial relations and only analyze non-adversarial relations, such as spatial proximity relationship, friendship and kinship. The co-occurrence of both relations generates an adversarial network, which exhibits a heterogeneous social structure. Technically, adversarial or non-adversarial relation between the characters c_i and c_j can be represented by a real-valued weight in the affinity matrix $K(c_i, c_j)$, which will be decided by the proposed affinity learning method.

[1] While conventionally v is used to represent a vertex in a graph, we will also use c in this paper, and both v and c point to the same character in the movie.

A movie $\mathcal{M}$ is composed of non-overlapping M scenes, $\mathcal{M} = s_1 \bigcup s_2 \bigcup \cdots \bigcup s_M$, where each scene contains interactions among a set of movie characters. Appearance of a character in a scene can be encoded in a scene-character relation matrix denoted by $A = \{A_{i,j}\}$, where $A_{i,j} = 1$ if $c_j \in s_i$. It can be obtained by searching for speaker names in the script. This representation is reminiscent of the actor-event graph in social network analysis [22]. While the character relations in A can be directly used for construction of the social network, we will demonstrate later that the use of visual and auditory scene features can lead to a better social network representation by learning the inter-character affinity matrix K.

Temporal Segmentation of Movie into Scenes In order to align visual and auditory features with the movie script, we require temporal segmentation of the movie into scenes, which will provide start and stop timings for each scene. This segmentation process is guided by the accompanying movie script and closed captions. The script is usually a draft version with no time tagging and lacks professional editing, while the closed captions are composed of lines d_i, which contain timed sentences uttered by characters. The approach we use to perform this task can be considered as a variant of the alignment technique in [5] and is summarized in Figure 2:

1. Divide the script into scenes, each of which is denoted as s_i. Similarly, closed captions are divided into lines d_i.
2. Define $\mathcal{C}$ to be a cost matrix. Compute the percentage p of the words in closed caption d_j matched with scene s_i while respecting the order of words. Set the cost as $\mathcal{C}_{i,j} = 1 - p$.
3. Apply dynamic time warping to $\mathcal{C}$ for estimating start t_1^i and stop times t_2^i of s_i, which respectively correspond to the smallest and largest time stamps for closed captions matched with s_i.

Due to the fact that publicly available scripts for movies are not perfectly edited, the temporal segmentation may not be precise. Regardless, our approach is robust to such inaccuracies in segment boundaries. A potential future modification of temporal segmentation can include a combination of the proposed approach with other automatic scene segmentation techniques, such as [24].

3 Learning Social Networks

Adversarial is defined "to have or involve antagonistic parties or opposing interests" between individuals or groups of people [16]. In movies or more generally in real life environments, adversarial relations between individuals are exhibited in the words they use, tones of their speech and actions they perform. Considering that a scene is the smallest segment in a movie which contains a continued event, low-level features generated from the video and audio of each scene can be used to quantify adversarial and non-adversarial contents. In the following text, we conjecture that the character members of the same community co-occur more often in non-adversarial scenes than in adversarial ones, and learn the social network formed by movie characters based on both the scene-character relations and scene contents.

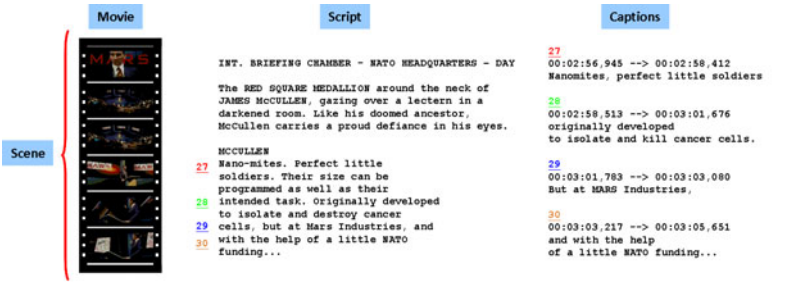


Fig. 2. Temporal segmentation of a movie into scenes. The colored numbers in the middle block indicate matched sentences in the closed captions shown on the right.

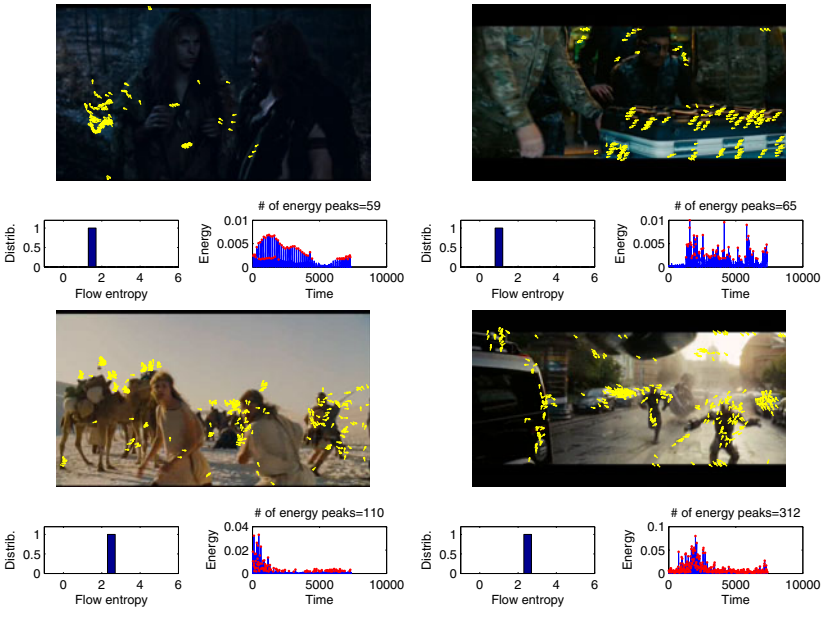


Fig. 3. Visual and auditory characteristics of adversarial scenes. Top row: non-adversarial scenes from *Year One (2009)* and *G.I. Joe: The Rise of Cobra (2009)*; Bottom row: adversarial scenes from these two movies. Optical flow vectors are superimposed on the frames and computed features are shown as plots for a temporal window of 10 video frames, including entropy distribution of optical flow vectors and detected energy peaks (red dots in energy signals).

3.1 Scene Level Features and Scene Characterization

Movie directors often follow certain rules, referred to as the film grammar or cinematic principles in the film literature, to emphasize the adversarial content in scenes. Typically, adversarial scenes contain abrupt changes in visual and

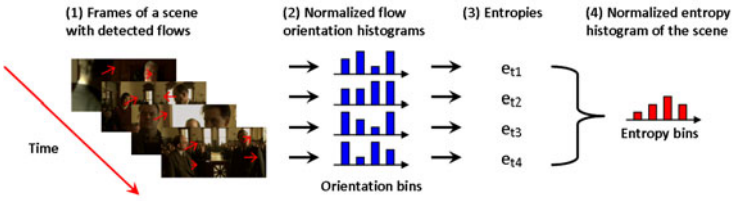


Fig. 4. Generation of the normalized entropy histogram from orientation distributions of optical flows detected from a scene

auditory contents, whereas these contents change gradually in non-adversarial scenes. We should, however, note that these clues can be dormant depending on the director's style. In our framework, we handle such dormant relations by learning a robust support vector regressor from a training set.

The visual and auditory features, which quantify adversarial scene content, can be extracted by analyzing the disturbances in the video [18]. In particular for measuring visual disturbance, we follow the cinematic principles and conjecture that for an adversarial scene, the motion field is nearly evenly distributed in all directions (see Figure 3 for illustration). For generating the optical flow distributions, we use the Kanade-Lucas-Tomasi tracker [20] within the scene bounds and use good features to track. Alternatively, one can use dense flow field generated by estimating optical flow at each pixel [15]. The visual disturbance in the observed flow field can be measured by entropy of the orientation distribution as shown in Figure 4. Specifically, we apply a moving window of 10 frames with 5 frames overlapping in the video for constructing the orientation histograms of optical flows. We use histograms of optical flow vectors weighted by the magnitude of motion. The number of orientation bins is set to 10 and the number

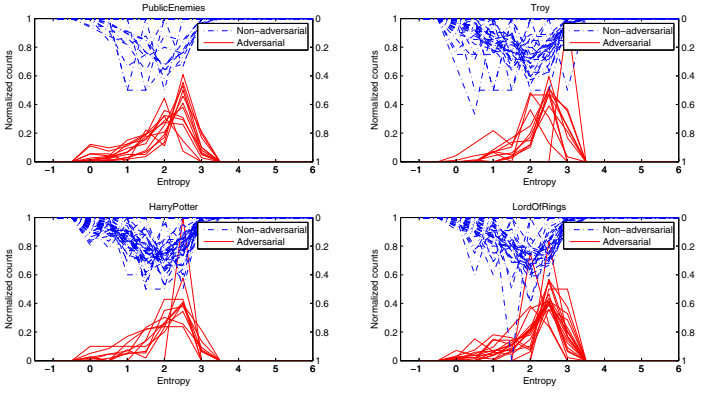


Fig. 5. Visualization of entropy histogram feature vectors extracted from four example movies. The two classes (adversarial and non-adversarial) have distinct patterns, in that adversarial scenes tend to consistently produce strong peaks in high entropies. Best viewed in color.

of entropy bins in the final feature vector is set to 5. As can be observed in Figure 5, flow distributions generated from adversarial scenes tend to be uniformly distributed and thus, they consistently have more high-entropy peaks compared to non-adversarial scenes. This observation serves as the basis for distinguishing the two types of scenes.

Auditory features extracted from the accompanying movie audio are used together with the visual features to improve the performance. We adopt a combination of temporal and spectral auditory features discussed in [13,18]: energy peak ratio, energy entropy, short-time energy, spectral flux and zero crossing rate. Specifically, these features are computed for sliding audio frames that are 400 ms in length. The means of these features over the duration of the scene constitute a feature vector. A sample auditory feature (energy peaks) is shown in Figure 3 for both adversarial and non-adversarial scenes. It can be observed that adversarial scenes have more peaks in energy signals, which are moving averages of squared audio signals.

The visual and auditory features provide two vectors per scene (5 dimensional visual and 5 dimensional auditory), which are used to estimate a real value $\beta_i \in [-1, +1]$ for quantifying the adverseness of the scene s_i. Broadly speaking, the more negative the β_i is the more adversarial the scene is, and vise versa. In order to facilitate estimation of β_i, we use support vector regression (SVR) [21], which has been successfully used to solve various problems in recent computer vision literature. We apply a radial basis function to both the visual and auditory feature vectors, which leads to two kernel matrices $\mathcal{K}_v$ and $\mathcal{K}_a$ respectively. The two kernel bandwidths can be chosen by using cross-validation. The joint kernel is then computed as the multiplication kernel: $\hat{\mathcal{K}} = \mathcal{K}_v \mathcal{K}_a$. Due to space limitations, we skip the details of the SVR and refer the reader to [21]. The final decision function is written as: $\beta_i = g(s_i) = \sum_{j=1}^{L} (\alpha_j - \alpha_j^*) \hat{\mathcal{K}}_{l_j,i} + b$, where the coefficient b is offset, α_i and α_i^* are the Lagrange multipliers for labeling constraints, L is the number of labeled examples, and l_j is the index for the j^{th} labeled example.

The training for support vector regression is achieved by using a set of scenes labeled as adversarial ($\beta_i = -1$) and non-adversarial ($\beta_i = +1$). We define a non-adversarial scene in the training and test sets as a scene which contains character members from only one group. Conversely, a scene in which the members of rival groups co-occur is labeled as adversarial. Considering that the adverseness of a scene is sometimes unclear and involves high-level semantics instead of pure observations, the stated approach avoids the subjectiveness in scene labeling. The labeling of scenes s_i in the novel movie $\mathcal{M}$ is then achieved by estimating corresponding β_i using the regression learned from labeled scene examples from other movies in a dataset.

3.2 Learning Inter-character Affinity

Let c_i be character i, and $\mathbf{f} = (f_1, \cdots, f_N)^T$ be the vector of community memberships containing ± 1 values, where f_i refers to the membership of c_i. Let $\mathbf{f}$ distribute according to a zero-mean identity-covariance Gaussian process $P(\mathbf{f}) = (2\pi)^{-N/2} \exp^{-\frac{1}{2}\mathbf{f}^T\mathbf{f}}$. In order to model the information contained in the

scene-character relation matrix A and the aforementioned adverseness of each scene β_i, we assume the following distributions: (1) if c_i and c_j occur in a non-adversarial scene k ($\beta_k \geq 0$), we assume $f_i - f_j \sim \mathcal{N}(0, \frac{1}{\beta_k^2})$; (2) if c_i and c_j occur in an adversarial scene k ($\beta_k < 0$), we assume $f_i + f_j \sim \mathcal{N}(0, \frac{1}{\beta_k^2})$.

Therefore, if $\beta_i = 0$, then the constraint imposed by a scene becomes inconsequential, which corresponds to the least confidence in the constraint. On the other hand, if $\beta_i = \pm 1$, the corresponding constraint becomes the strongest. Because of the distributions we use, none of the constraints is hard, making our method relatively flexible and insensitive to prediction errors. Applying the Bayes' rule, the posterior probability of $\mathbf{f}$ given the constraints is defined by:

$$P(\mathbf{f}|A,\beta) \propto \exp\left(-\frac{1}{2}\mathbf{f}^T\mathbf{f} - \sum_{k:\beta_k \geq 0} \sum_{c_i,c_j \in s_k} \frac{(f_i - f_j)^2 \beta_k^2}{2} - \sum_{k:\beta_k < 0} \sum_{c_i,c_j \in s_k} \frac{(f_i + f_j)^2 \beta_k^2}{2}\right).$$

It can be verified that $P(\mathbf{f}|A,\beta) \propto \exp(-\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f})$ is a Gaussian process with zero mean. Using $K_{i,j} = E\{f_i f_j | A, \beta\}$ as the learned affinity between c_i and c_j, it follows that $K = M^{-1}$, where

$$M_{i,j} = \begin{cases} \sum_{k:c_i,c_j \in s_k, \beta_k < 0} \beta_k^2 - \sum_{k:c_i,c_j \in s_k, \beta_k \geq 0} \beta_k^2 & \text{if } i \neq j \\ 1 + \sum_{l \neq i} \sum_{k:c_i,c_l \in s_k} \beta_k^2 & \text{if } i = j \end{cases}.$$

The resulting K is symmetric and positive definite. However, unlike an affinity matrix from a Gaussian kernel, it may contain negative values. The proposed approach has two special cases:

- In the case when $\beta_i = 1$, then the aforementioned learning mechanism reduces to a co-occurrence based approach which is a traditional tool in social network analysis [17,4]. Specifically, $M_{i,j}$, for $i \neq j$, represents the minus value of the number of scenes where c_i and c_j occur together. This reduced scheme does not utilize the video/audio feature based prediction of adverseness, and serves as a natural baseline in this paper.
- If we use fixed variance parameters in the assumed distributions instead of the learned ones, our affinity learning method reduces to the affinity propagation approach proposed in [14].

4 Social Network Analysis

In this section, we deal with grouping the movie characters into communities and finding the leader of each community. A common approach to detecting communities from a social network is to cluster vertices of the corresponding graph using the modularity-cut [17], which has been recently used in context of surveillance [23]. For social environments, a recent study reported in [4] has shown that community detection performance of [17] can be increased by considering a generalized objective referred to as the *max-min modularity*. Their proposed algorithm, however, assumes unweighted edges and is not directly suitable for our social networks which contain weighted edges of learned strength.

In our design, we first generate a *principal affinity matrix* K' by the following rules: $K'_{i,j} = K_{i,j}$ for $K_{i,j} > 0$, and $K'_{i,j} = 0$ for other entries. We then generate a *complementary affinity matrix* K'' by the following rules: $K''_{i,j} = -K_{i,j}$ for $K_{i,j} < 0$, and $K''_{i,j} = 0$ for other entries. The matrix K'' represents the *unrelatedness* between vertices in the network in terms of community memberships. Adopting the strategy in [4] and using K' and K'', we formulate the max-min modularity criterion as $Q_{MM} = Q_{max} - Q_{min}$ for:

$$Q_{max} = \frac{1}{2m'} \sum_{i,j} (K'_{ij} - \frac{k'_i k'_j}{2m'})(f_i f_j + 1) \triangleq \frac{1}{2m'} \sum_{i,j} B'_{i,j}(f_i f_j + 1),$$

$$Q_{min} = \frac{1}{2m''} \sum_{i,j} (K''_{ij} - \frac{k''_i k''_j}{2m''})(f_i f_j + 1) \triangleq \frac{1}{2m''} \sum_{i,j} B''_{i,j}(f_i f_j + 1),$$

where $m' = \frac{1}{2}\sum_{ij} K'_{ij}$, $k'_i = \sum_j K'_{ij}$, $m'' = \frac{1}{2}\sum_{ij} K''_{ij}$, $k''_i = \sum_j K''_{ij}$ and the term $\frac{k'_i k'_j}{2m'}$ represents the expected edge strength between the characters c_i and c_j [17]. Based on this observation, we note that $K'_{i,j} - \frac{k'_i k'_j}{2m}$ measures how much the connection between two characters is stronger than what would be expected between them, and serves as the basis for keeping the two characters in the same community. In this formulation, the max-min modularity Q_{MM} roots from the conditions for a good network division that (1) edge strength across communities should be smaller than expected, and (2) unrelated characters within a community should be minimal. These conditions can be realized by maximizing Q_{MM}. Using standard eigen-analysis, it follows that the eigenvector $\mathbf{u}$ of $\frac{1}{2m'}B' - \frac{1}{2m''}B''$ with the largest eigenvalue maximizes a relaxed version of Q_{MM}. The resulting eigenvector solution contains real values, and we threshold them at the 0 level to obtain the desired community memberships. That is, we let $f_i = +1$ if $u_i \geq 0$, and $f_i = -1$ if otherwise.

Once the communities in the movie are extracted, their leaders can be computed by analyzing the centrality of each character in the community. In our case, since the communities correspond to two adversarial social groups, their expected leaders relate to the *hero* or the *villain* in the movie. Let the centrality score, x_i for the i^{th} movie character be proportional to the sum of the scores of all vertices which are connected to it: $x_i = \frac{1}{\lambda}\sum_{j=1}^{N} K'_{i,j} x_j$, where N is the total number of characters in the movie and λ is a constant. It follows from this notation that the centralities of characters satisfy $K'\mathbf{x} = \lambda\mathbf{x}$ in the vector form. It can be shown that the eigenvector with largest eigenvalue provides the desired centrality measure [19]. Therefore, if we let the eigenvector of K' with the largest eigenvalue be $\mathbf{v}$, the leaders of the two communities are given by $\arg\max_{i:u_i \geq 0} v_i$ and $\arg\max_{i:u_i < 0} v_i$ respectively.

5 Experiments

For qualitative and quantitative evaluation of the proposed approach, we generate a dataset of 10 movies which contains recent and classical theatrical movies

that cover a range of genres including action, adventure, fantasy and drama.[2] The movies in our dataset broadly contain two rival communities with a designated leader for each community. For each movie with statistics tabulated in Table 1, the dataset contains visual and auditory features, movie script, and closed caption data, all of which are temporally aligned.

Table 1. Statistics of movies in our dataset which includes the number of scenes in the movie, the number of lines in closed caption data, the total number of characters in the movie and the number of characters in one of the two communities

Movies enumerated in footnote 2	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
# of scenes	198	151	238	226	116	51	105	297	188	199
# of captions in lines	1143	1585	1063	1155	1337	1515	1293	1262	1099	1402
# of characters in total	11	7	10	10	7	7	6	8	10	9
# of characters in community 1	6	4	6	5	3	3	4	6	7	7

In the following discussion, we analyze social networks with accompanying affinity matrices generated from

- the character co-occurrence information reflected in matrix A (co-occurrence), which is more traditional in sociology;
- in addition to co-occurrence, scene adverseness characterizations β_i which are learned from video and audio contents using the proposed approach.

In order to evaluate the contribution of these features, we provide comparisons of collective use of visual and auditory features with their individual use in extraction of communities and their leaders (details are discussed in Section 3). Due to space limitations, in Figure 6, we only show graphical representations of the social networks for ten movies learned from both visual and auditory features using the proposed approach. The color codes in the figure reflect the strength of affinity between characters. We observe that inter-community connections tend to be weaker than certain intra-community ones.

In this paper, the affinity between the characters are strongly related to the adverseness of the scenes in which they appear. This relation suggests validation of how effective the support vector regression (SVR) is for estimating the scene adverseness. In order to facilitate this, we compute the mean square error (MSE) as our error measure over all the scenes in each movie, and average the resulting MSEs over all ten movies. When both the visual and auditory features are used, the MSE is estimated as 0.61. In contrast, when only one of the features is used MSE increases to 0.80 for visual only and 0.77 for auditory only. These

[2] The movies in or dataset are (1) *G.I. Joe: The Rise of Cobra (2009)*; (2) *Harry Potter and the Half-Blood Prince (2009)*; (3) *Public Enemies (2009)*; (4) *Troy (2004)*; (5) *Braveheart (1995)*; (6) *Year One (2009)*; (7) *Coraline (2009)*; (8) *True Lies (1994)*; (9) *The Chronicles of Narnia: The Lion, the Witch and the Wardrobe (2005)*; (10) *The Lord of the Rings: The Return of the King (2003)* . The dataset is available at `http://dpl.ceegs.ohio-state.edu/resources.php`.

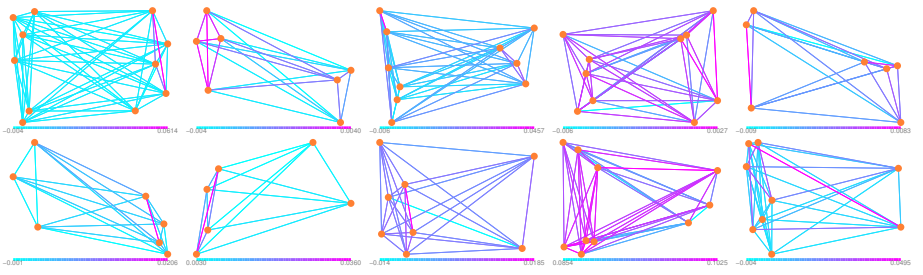


Fig. 6. Social networks generated using the proposed approach for the ten movies in our dataset. Characters (vertices) are placed on the left and right with respect to communities they belong to. The strength of affinity is indicated by pinkness of the edges: the stronger the edge is the pinker it is. Best viewed in color.

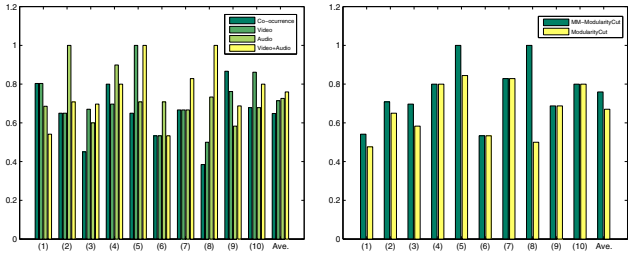


Fig. 7. Accuracy of social network analysis in $F1$ measures. Left: comparison of four approaches, where the proposed one is video+audio; Right: comparison of two modularity algorithms (max-min vs. original), with the proposed video+audio approach.

numbers translate into accuracy rates for predicting if a scene is adversarial or non-adversarial. Respectively, the accuracy rates are computed as 81.6%, 78.2% and 78.7% for collective feature use, visual only and auditory only. These numbers reflect that the adverseness estimates of scenes can be further utilized to infer the relations among the movie characters.

The accuracy of community detection relates to how precise the assignment of the characters is into each one of the community. Considering that a community is a set of characters, the accuracy can be measured using the precision and recall values of predicted assignments given the ground truth. For each community these two values can be combined into an $F1$ measure, which is the harmonic mean of precision and recall. This measure takes into account the possible imbalance in the size of communities and has been widely adopted. Considering that the movies in our dataset contains more than one community, we report the average $F1$ measure over detected communities as the final detection accuracy for each movie. From the quantitative evaluations shown in Figure 7, for four movies visual features help enhance performance appreciably. Overall, auditory features improve the performance slightly more than the visual features when they are used independently. Their combination, however, provides

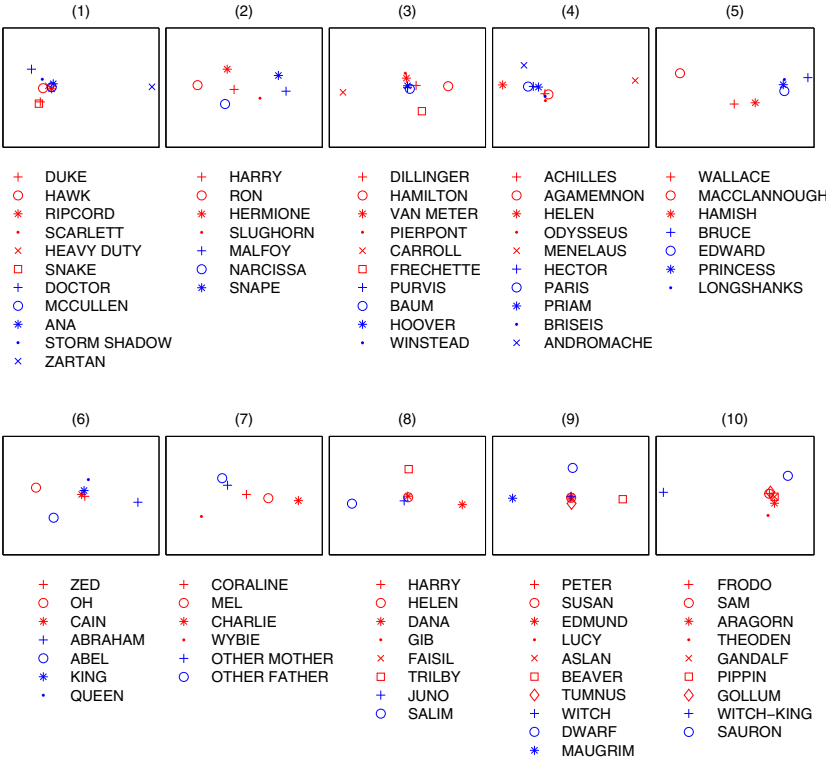


Fig. 8. 2D visual maps of character relations. Red and blue stand for the two communities respectively according to our ground truth labeling. Best viewed in color.

the best performance, which on the average leads to an $F1$ measure of 76.0%. This score, when compared to using only the character co-occurrence to generate the social network, improves the grouping performance by 11.1%. In the same figure, we also show that the modified max-min modularity, when compared to the traditional modularity computed from K, improves the $F1$ measure by 8.9%.

As discussed in Section 4, the community assignment of characters is realized by analyzing the eigenspace of $\frac{1}{2m'}B' - \frac{1}{2m''}B''$. In order to visualize this assignment process, we map the characters in the movie into coordinates defined by the two eigenvectors with highest eigenvalues. This mapping provides an optimal way to visualize the inter-character relations in two dimensions. In the figure, we illustrate the ground truth in red and blue colors respectively for the two communities[3]. As can be observed, the characters who belong to separate communities tend to lie apart.

As discussed in Section 4, the eigenvector of K' with the highest eigenvalue provides the leaders of communities. In Table 2, we tabulate these leaders with their pictures for the two rival communities for each movie. The predicted leaders

[3] In movie (10), *Gollum* has a good personality except for when he is close to *the ring*. The ring changes the good behavior of the characters to bad except for *Frodo*.

Table 2. Community leaders discovered using the proposed framework. The names in bold face refer to correct ones, whereas those in italics are not.

Movies	(1)	(2)	(3)	(4)	(5)
Community 1	 *Hawk*	 **Harry**	 **Dillinger**	 **Achilles**	 *MacClan.*
Community 2	 **McCullen**	 **Snape**	 **Purvis**	 *Androm.*	 **Longsha.**
Movies	(6)	(7)	(8)	(9)	(10)
Community 1	 **Zed**	 **Coraline**	 *Trilby*	 *Susan*	 **Frodo**
Community 2	 *Abraham*	 **OtherMo.**	 **Salim**	 **Witch**	 **WitchKing**

who correspond to the true leaders in the movie are shown in bold face, while incorrect leaders are shown in italics. Overall, it can be observed that many of the leaders are successfully discovered by our framework.

6 Conclusions and Future Work

In this paper, we have presented the first work on learning the relations among characters from movies using a social network approach. We have used visual and auditory features to characterize the adverseness of each scene in a movie. By using an affinity learning procedure, we incorporate the scene adverseness, and make informed decisions in constructing and analyzing the corresponding social network. Extensive analysis on a set of 10 movies has validated the effectiveness of our framework in high level understanding of social interactions. The proposed framework also contributes to sociology by leveraging computer vision and machine learning techniques. Although we present our framework on analysis of movies, it is possible to apply it to other problem domains, such as video surveillance, in which suspicious behaviors can be related to the interactions between objects in a scene.

References

1. Ali, S., Basharat, A., Shah, M.: Chaotic invariants for human action recognition. In: ICCV (2007)
2. Alon, J., Athitsos, V., Yuan, Q., Sclaroff, S.: A unified framework for gesture recognition and spatiotemporal gesture segmentation. IEEE Trans. on PAMI 31(9), 1685–1699 (2009)
3. Arandjelović, O., Zisserman, A.: Automatic face recognition for film character retrieval in feature-length films. In: CIVR (2005)

4. Chen, J., Zaiane, O., Goebel, R.: Detecting communities in social networks using max-min modularity. In: SDM (2009)
5. Cour, T., Jordan, C., Miltsakaki, E., Taskar, B.: Movie/script: Alignment and parsingof video and text transcription. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 158–171. Springer, Heidelberg (2008)
6. Ding, L., Fan, Q., Hsiao, J., Pankanti, S.: Graph based event detection from realistic videos using weak feature correspondence. In: ICASSP (2010)
7. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV (2003)
8. Fathi, A., Mori, G.: Action recognition by learning mid-level motion features. In: CVPR (2008)
9. Freeman, L.: Centrality in social networks: Conceptual clarification. Social Networks 1(3), 215–239 (1979)
10. Ge, W., Collins, R., Ruback, B.: Automatically detecting the small group structure of a crowd. In: WACV (2009)
11. Jiang, H., Fels, S., Little, J.: A linear programming approach for multiple object tracking. In: CVPR (2007)
12. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV (2003)
13. Lin, J., Wang, W.: Weakly-supervised violence detection in movies with audio and video based co-training. In: PCM (2009)
14. Lu, Z., Carreira-Perpinan, M.A.: Constrained spectral clustering through affinity propagation. In: CVPR (2008)
15. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI (1981)
16. Merriam-Webster: Merriam-webster online dictionary (2010), http://www.merriam-webster.com/dictionary
17. Newman, M.E.J.: Modularity and community structure in networks. PNAS 103(23), 8577–8582 (2006)
18. Rasheed, Z., Shah, M.: Movie genre classification by exploiting audio-visual features of previews. In: ICPR (2002)
19. Ruhnau, B.: Eigenvector-centrality? a node-centrality. Social Networks 22(4), 357–365 (2000)
20. Shi, J., Tomasi, C.: Good features to track. In: CVPR (1994)
21. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing 14(3), 199–222 (2004)
22. Wasserman, S., Faust, K., Iacobucci, D.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge (1994)
23. Yu, T., Lim, S.N., Patwardhan, K., Krahnstoever, N.: Monitoring, recognizing and discovering social networks. In: CVPR (2009)
24. Zhai, Y., Shah, M.: Video scene segmentation using markov chain monte carlo. IEEE Trans. on Multimedia 8(4), 686–697 (2006)

What, Where and How Many? Combining Object Detectors and CRFs

Ľubor Ladický, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip H.S. Torr*

Oxford Brookes University
http://cms.brookes.ac.uk/research/visiongroup

Abstract. Computer vision algorithms for individual tasks such as object recognition, detection and segmentation have shown impressive results in the recent past. The next challenge is to integrate all these algorithms and address the problem of scene understanding. This paper is a step towards this goal. We present a probabilistic framework for reasoning about regions, objects, and their attributes such as object class, location, and spatial extent. Our model is a Conditional Random Field defined on pixels, segments and objects. We define a global energy function for the model, which combines results from sliding window detectors, and low-level pixel-based unary and pairwise relations. One of our primary contributions is to show that this energy function can be solved efficiently. Experimental results show that our model achieves significant improvement over the baseline methods on CamVid and PASCAL VOC datasets.

1 Introduction

Scene understanding has been one of the central goals in computer vision for many decades [1]. It involves various individual tasks, such as object recognition, image segmentation, object detection, and 3D scene recovery. Substantial progress has been made in each of these tasks in the past few years [2,3,4,5,6]. In light of these successes, the challenging problem now is to put these individual elements together to achieve the grand goal — *scene understanding*, a problem which has received increasing attention recently [6,7]. The problem of scene understanding involves explaining the whole image by recognizing all the objects of interest within an image and their spatial extent or shape. This paper is a step towards this goal. We address the problems of *what*, *where*, and *how many*: we recognize objects, find their location and spatial extent, segment them, and also provide the number of instances of objects. This work can be viewed as an integration of object class segmentation methods [3], which fail to distinguish between adjacent instances of objects of the same class, and object detection approaches [4], which do not provide information about background classes, such as grass, sky and road.

The problem of scene understanding is particularly challenging in scenes composed of a large variety of classes, such as road scenes [8] and images in the PASCAL VOC

* This work is supported by EPSRC research grants, HMGCC, the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. P. H. S. Torr is in receipt of Royal Society Wolfson Research Merit Award.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 424–437, 2010.

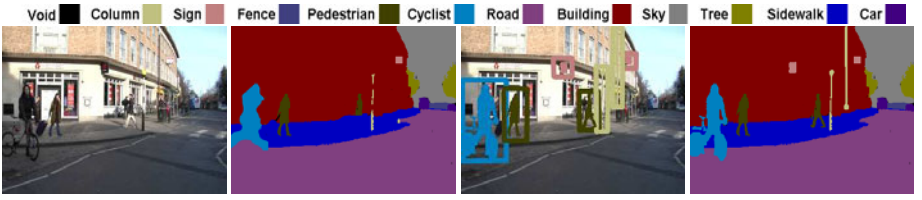


Fig. 1. A conceptual view of our method. (a) An example input image. (b) Object class segmentation result of a typical CRF approach. (c) Object detection result with foreground/background estimate within each bounding box. (d) Result of our proposed method, which jointly infers about objects and pixels. Standard CRF methods applied to complex scenes as in (a) underperform on the "things" classes, e.g. inaccurate segmentation of the bicyclist and persons, and misses a pole and a sign, as seen in (b). However, object detectors tend to perform well on such classes. By incorporating these detection hypotheses (§2.2), shown in (c), into our framework, we aim to achieve an accurate overall segmentation result as in (d) (§3.3). (**Best viewed in colour**)

dataset [9]. For instance, road scene datasets contain classes with specific shapes such as person, car, bicycle, as well as background classes such as road, sky, grass, which lack a distinctive shape (Figure 1). The distinction between these two sets of classes — referred to as *things* and *stuff* respectively — is well known [10,11,12]. Adelson [10] emphasized the importance of studying the properties of *stuff* in early vision tasks. Recently, these ideas are being revisited in the context of the new vision challenges, and have been implemented in many forms [12,13,14,15]. In our work, we follow the definition by Forsyth *et al*. [11], where *stuff* is a homogeneous or reoccurring pattern of fine-scale properties, but has no specific spatial extent or shape, and a *thing* has a distinct size and shape. The distinction between these classes can also be interpreted in terms of localization. *Things*, such as cars, pedestrians, bicycles, can be easily localized by bounding boxes unlike *stuff*, such as road, sky[1].

Complete scene understanding requires not only the pixel-wise segmentation of an image, but also an identification of object instances of a particular class. Consider an image of a road scene taken from one side of the street. It typically contains many cars parked in a row. Object class segmentation methods such as [3,8,16] would label all the cars adjacent to each other as belonging to a large car segment or blob, as illustrated in Figure 2. Thus, we would not have information about the number of instances of a particular object—car in this case. On the other hand, object detection methods can identify the number of objects [4,17], but cannot be used for background (*stuff*) classes.

In this paper, we propose a method to jointly estimate the class category, location, and segmentation of objects/regions in a visual scene. We define a global energy function for the Conditional Random Field (CRF) model, which combines results from detectors (Figure 1(c)), pairwise relationships between mid-level cues such as superpixels, and low-level pixel-based unary and pairwise relations (Figure 1(b)). We also show that, unlike [6,18], our formulation can be solved efficiently using graph cut based move

[1] Naturally what is classified as things or stuff might depend on either the application or viewing scale, *e.g*. flowers or trees might be things or stuff.

(a) (b) (c)

Fig. 2. (a) Object class segmentation results (without detection), (b) The detection result, (c) Combined segmentation and detection. Object class segmentation algorithms, such as [3], label all the cars adjacent to each other as belonging to one large blob. Detection methods localize objects and provide information about the number of objects, but do not give a segmentation. Our method jointly infers the number of object instances and the object class segmentation. See §2.3 for details. (**Best viewed in colour**)

making algorithms. We evaluate our approach extensively on two widely used datasets, namely Cambridge-driving Labeled Video Database (CamVid) [8] and PASCAL VOC 2009 [9], and show a significant improvement over the baseline methods.

Outline of the paper. Section 1.1 discusses the most related work. Standard CRF approaches for the object segmentation task are reviewed in Section 2.1. Section 2.2 describes the details of the detector-based potential, and its incorporation into the CRF framework. We also show that this novel CRF model can be efficiently solved using graph cut based algorithms in Section 2.3. Implementation details and the experimental evaluation are presented in Section 3. Section 4 discusses concluding remarks.

1.1 Related Work

Our method is inspired by the works on object class segmentation [3,6,8,16], foreground (*thing*) object detection [4,17], and relating *things* and *stuff* [12]. Whilst the segmentation methods provide impressive results on certain classes, they typically underperform on *things*, due to not explicitly capturing the global shape information of object class instances. On the other hand, detection methods are geared towards capturing this information, but tend to fail on *stuff*, which is amorphous.

A few object detection methods have attempted to combine object detection and segmentation sub-tasks, however they suffer from certain drawbacks. Larlus and Jurie [19] obtained an initial object detection result in the form of a bounding box, and then refined this rectangular region using a CRF. A similar approach has been followed by entries based on object detection algorithms [4] in the PASCAL VOC 2009 [9] segmentation challenge. This approach is not formulated as one energy cost function and cannot be applied to either cluttered scenes or *stuff* classes. Furthermore, there is no principled way of handling multiple overlapping bounding boxes. Tu *et al.* [15] also presented an effective approach for identifying text and faces, but leave much of the image unlabelled. Gu *et al.* [20] used regions for object detection instead of bounding boxes, but

were restricted to using a single over-segmentation of the image. Thus, their approach cannot recover from any errors in this initial segmentation step. In comparison, our method does not make such *a priori* decisions, and jointly reasons about segments and objects.

The work of layout CRF [21] also provides a principled way to integrate things and stuff. However, their approach requires that things must conform to a predefined structured layout of parts, and does not allow for the integration of arbitrary detector responses. To our knowledge, the only other existing approaches that attempt to jointly estimate segmentation and detection in one optimization framework are the works of [6,18]. However, the minimization of their cost functions is intractable and their inference methods can get easily stuck in local optima. Thus, their incorporation of detector potentials does not result in a significant improvement of performance. Also, [6] focussed only on two classes (cars and pedestrians), while we handle many types of objects (*e.g.* 20 classes in the PASCAL VOC dataset). A direct comparison with this method was not possible as neither their code nor their dataset because ground truth annotations are not publicly available at the time of publication.

2 CRFs and Detectors

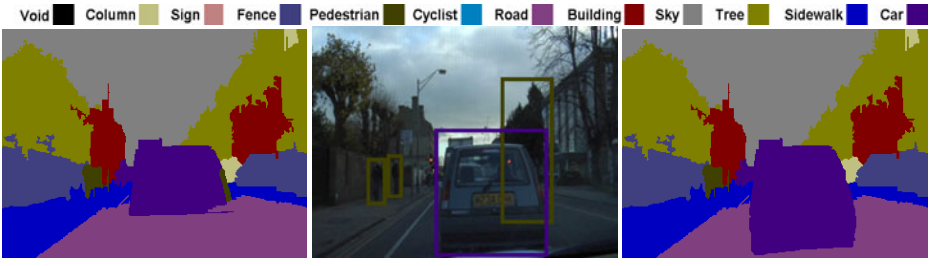
We define the problem of jointly estimating segmentation and detection in terms of minimizing a global energy function on a CRF model. Our approach combines the results from detectors, pairwise relationships between superpixels, and other low-level cues. Note that our framework allows us to incorporate any object detection approach into any pixel or segment based CRF.

2.1 CRFs for Labelling Problems

In the standard CRF formulation for image labelling problems [3] we represent each pixel as random variable. Each of these random variables takes a label from the set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$, which may represent objects such car, airplane, bicycle. Let $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ denote the set of random variables corresponding to the image pixels $i \in \mathcal{V} = \{1, 2, \ldots, N\}$. A clique c is a set of random variables $\mathbf{X}_c$ which are conditionally dependent on each other. A labelling $\mathbf{x}$ refers to any possible assignment of labels to the random variables and takes values from the set $\mathbf{L} = \mathcal{L}^N$.

The posterior distribution $\Pr(\mathbf{x}|\mathbf{D})$ over the labellings of the CRF can be written as: $\Pr(\mathbf{x}|\mathbf{D}) = \frac{1}{Z}\exp(-\sum_{c\in\mathcal{C}}\psi_c(\mathbf{x}_c))$, where Z is a normalizing constant called the *partition function*, $\mathcal{C}$ is the set of all cliques, and $\mathbf{D}$ the given data. The term $\psi_c(\mathbf{x}_c)$ is known as the potential function of the clique $c \subseteq \mathcal{V}$, where $\mathbf{x}_c = \{x_i : i \in c\}$. The corresponding Gibbs energy is given by: $E(\mathbf{x}) = -\log\Pr(\mathbf{x}|\mathbf{D}) - \log Z = \sum_{c\in\mathcal{C}}\psi_c(\mathbf{x}_c)$. The most probable or Maximum a Posteriori (MAP) labelling $\mathbf{x}^*$ of the random field is defined as: $\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{L}}\Pr(\mathbf{x}|\mathbf{D}) = \arg\min_{\mathbf{x}\in\mathbf{L}}E(\mathbf{x})$.

In computer vision labelling problems such as segmentation or object recognition, the energy $E(\mathbf{x})$ is typically modelled as a sum of unary, pairwise [3,22], and higher order [23] potentials. The unary potentials are based on local feature responses and capture the likelihood of a pixel taking a certain label. Pairwise potentials encourage neighbouring pixels in the image to take the same label. Similarly, a CRF can be defined



(a) (b) (c)

Fig. 3. (a) Segmentation without object detectors, (b) Object detections for car and pedestrian shown as bounding boxes, (c) Segmentation using our method. These detector potentials act as a soft constraint. Some false positive detections (such as the large green box representing person) do not affect the final segmentation result in (c), as it does not agree with other strong hypotheses based on pixels and segments. On the other hand, a strong detector response (such as the purple bounding box around the car) correctly relabels the road and pedestrian region as car in (c) resulting in a more accurate object class segmentation. (**Best viewed in colour**)

over segments [24,25] obtained by unsupervised segmentation [26,27] of the image. Recently, these models have been generalized to include pixels and segments in a single CRF framework by introducing higher order potentials [16]. All these models successfully reason about pixels and/or segments. However, they fail to incorporate the notion of object instances, their location, and spatial extent (which are important cues used by humans to understand a scene) into the recognition framework. Thus, these models are insufficient to address the problem of scene understanding. We aim to overcome these issues by introducing novel object detector based potentials into the CRF framework.

2.2 Detectors in CRF Framework

MAP estimation can be understood as a soft competition among different hypotheses (defined over pixel or segment random variables), in which the final solution maximizes the weighted agreement between them. These weighted hypotheses can be interpreted as potentials in the CRF model. In object class recognition, these hypotheses encourage: (i) variables to take particular labels (unary potentials), and (ii) agreement between variables (pairwise). Existing methods [16,24,25] are limited to such hypotheses provided by pixels and/or segments only. We introduce an additional set of hypotheses representing object detections for the recognition framework[2].

Some object detection approaches [4,19] have used their results to perform a segmentation within the detected areas[3]. These approaches include both the true and false positive detections, and segment them assuming they all contain the objects of interest. There is no way of recovering from these erroneous segmentations. Our approach overcomes this issue by using the detection results as hypotheses that can be rejected

[2] Note that our model chooses from a set of given detection hypotheses, and does not propose any new detections.

[3] As evident in some of the PASCAL VOC 2009 segmentation challenge entries.

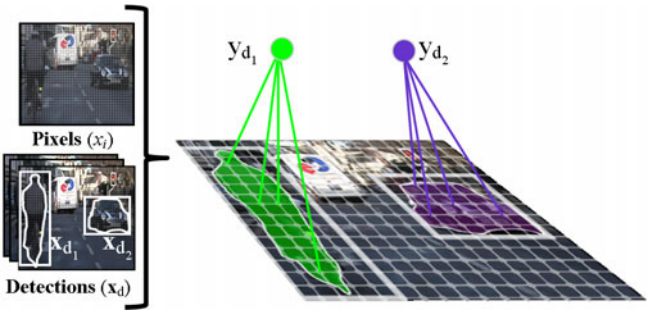


Fig. 4. Inclusion of object detector potentials into a CRF model. We show a pixel-based CRF as an example here. The set of pixels in a detection d_1 (corresponding to the bicyclist in the scene) is denoted by $\mathbf{x}_{d_1}$. A higher order clique is defined over this detection window by connecting the object pixels $\mathbf{x}_{d_1}$ to an auxiliary variable $y_{d_1} \in \{0, 1\}$. This variable allows the inclusion of detector responses as soft constraints. (**Best viewed in colour**)

in the global CRF energy. In other words, all detections act as soft constraints in our framework, and must agree with other cues from pixels and segments before affecting the object class segmentation result. We illustrate this with one of our results shown in Figure 3. Here, the false positive detection for "person" class (shown as the large green box on the right) does not affect the segmentation result in (c). Although, the true positive detection for "car" class (shown as the purple box) refines the segmentation because it agrees with other hypotheses. This is achieved by using the object detector responses[4] to define a clique potential over the pixels, as described below.

Let $\mathcal{D}$ denote the set of object detections, which are represented by bounding boxes enclosing objects, and corresponding scores that indicate the strength of the detections. We define a novel clique potential ψ_d over the set of pixels $\mathbf{x}_d$ belonging to the d-th detection (*e.g.* pixels within the bounding box), with a score H_d and detected label l_d. Figure 4 shows the inclusion of this potential graphically on a pixel-based CRF. The new energy function is given by:

$$E(\mathbf{x}) = E_{pix}(\mathbf{x}) + \sum_{d \in \mathcal{D}} \psi_d(\mathbf{x}_d, H_d, l_d), \tag{1}$$

where $E_{pix}(\mathbf{x})$ is any standard pixel-based energy. The minimization procedure should be able to reject false detection hypotheses on the basis of other potentials (pixels and/or segments). We introduce an auxiliary variable $y_d \in \{0, 1\}$, which takes value 1 to indicate the acceptance of d-th detection hypothesis. Let ϕ_d be a function of this variable and the detector response. Thus the detector potential $\psi_d(.)$ is the minimum of the energy values provided by including ($y_d = 1$) and excluding ($y_d = 0$) the detector hypothesis, as given below:

$$\psi_d(\mathbf{x}_d, H_d, l_d) = \min_{y_d \in \{0,1\}} \phi_d(y_d, \mathbf{x}_d, H_d, l_d). \tag{2}$$

[4] This includes sliding window detectors as a special case.

We now discuss the form of this function $\phi_d(\cdot)$. If the detector hypothesis is included ($y_d = 1$), it should: (a) Encourage consistency by ensuring that labellings where all the pixels in $\mathbf{x_d}$ take the label l_d should be more probable, *i.e.* the associated energy of such labellings should be lower; (b) Be robust to partial inconsistencies, *i.e.* pixels taking a label other than l_d in the detection window. Such inconsistencies should be assigned a cost rather than completely disregarding the detection hypothesis. The absence of the partial inconsistency cost will lead to a hard constraint where either all or none of the pixels in the window take the label l_d. This allows objects partially occluded to be correctly detected and labelled.

To enable a compact representation, we choose the potential ψ_d such that the associated cost for partial inconsistency depends only on the number of pixels $N_d = \sum_{i \in \mathbf{x_d}} \delta(x_i \neq l_d)$ disagreeing with the detection hypothesis. Let $f(\mathbf{x}_d, H_d)$ define the strength of the hypothesis and $g(N_d, H_d)$ the cost taken for partial inconsistency. The detector potential then takes the form:

$$\psi_d(\mathbf{x}_d, H_d, l_d) = \min_{y_d \in \{0,1\}} (-f(\mathbf{x}_d, H_d) y_d + g(N_d, H_d) y_d). \tag{3}$$

A stronger classifier response H_d indicates an increased likelihood of the presence of an object at a location. This is reflected in the function $f(\cdot)$, which should be monotonically increasing with respect to the classifier response H_d. As we also wish to penalize inconsistency, the function $g(\cdot)$ should be monotonically increasing with respect to N_d. The number of detections used in the CRF framework is determined by a threshold H_t. The hypothesis function $f(\cdot)$ is chosen to be a linear truncated function using H_t as:

$$f(\mathbf{x}_d, H_d) = w_d |\mathbf{x}_d| \max(0, H_d - H_t), \tag{4}$$

where w_d is the detector potential weight. This ensures that $f(\cdot) = 0$ for all detections with a response $H_d \leq H_t$. We choose the inconsistency penalizing function $g(\cdot)$ to be a linear function of the number of inconsistent pixels N_d of the form:

$$g(N_d, H_d) = k_d N_d, \quad k_d = \frac{f(\mathbf{x}_d, H_d)}{p_d |\mathbf{x}_d|}, \tag{5}$$

where the slope k_d was chosen such that the inconsistency cost equals $f(\cdot)$ when the percentage of inconsistent pixels is p_d.

Detectors may be applied directly, especially if they estimate foreground pixels themselves. However, in this work, we use sliding window detectors, which provide a bounding box around objects. To obtain a more accurate set of pixels $\mathbf{x}_d$ that belong to the object, we use a local colour model [28] to estimate foreground and background within the box. This is similar to the approach used by submissions in the PASCAL VOC 2009 segmentation challenge. Any other foreground estimation techniques may be used. See §3 for more details on the detectors used. Note that equation (1) could be defined in a similar fashion over superpixels.

2.3 Inference for Detector Potentials

One of the main advantages of our framework is that the associated energy function can be solved efficiently using graph cut [29] based move making algorithms (which

outperform message passing algorithms [30,31] for many vision problems). We now show that our detector potential in equation (3) can be converted into a form solvable using $\alpha\beta$-swap and α-expansion algorithms [2]. In contrast, the related work in [6] suffers from a difficult to optimize energy. Using equations (3), (4), (5), and $N_d = \sum_{i \in \mathbf{x_d}} \delta(x_i \neq l_d)$, the detector potential $\psi_d(\cdot)$ can be rewritten as follows:

$$\begin{aligned}\psi_d(\mathbf{x}_d, H_d, l_d) &= \min(0, -f(\mathbf{x}_d, H_d) + k_d \sum_{i \in \mathbf{x}_d} \delta(x_i \neq l_d)) \\ &= -f(\mathbf{x}_d, H_d) + \min(f(\mathbf{x}_d, H_d), k_d \sum_{i \in \mathbf{x}_d} \delta(x_i \neq l_d)). \end{aligned} \tag{6}$$

This potential takes the form of a Robust P^N potential [23], which is defined as:

$$\psi_h(\mathbf{x}) = \min(\gamma_{max}, \min_l(\gamma_l + k_l \sum_{i \in \mathbf{x}} \delta(x_i \neq l))), \tag{7}$$

where $\gamma_{max} = f(\cdot), \gamma_l = f(\cdot), \forall l \neq d$, and $\gamma_d = 0$. Thus it can be solved efficiently using $\alpha\beta$-swap and α-expansion algorithms as shown in [23]. The detection instance variables y_d can be recovered from the final labelling by computing y_d as:

$$y_d = \arg \min_{y'_d \in \{0,1\}} (-f(\mathbf{x}_d, H_d) y'_d + g(N_d, H_d) y'_d). \tag{8}$$

3 Experimental Evaluation

We evaluated our framework on the CamVid [8] and PASCAL VOC 2009 [9] datasets.

CamVid. The Cambridge-driving Labeled Video Database (CamVid) consists of over 10 minutes of high quality 30 Hz footage. The videos are captured at 960 × 720 resolution with a camera mounted inside a car. Three of the four sequences were shot in daylight, and the fourth sequence was captured at dusk. Sample frames from the day and dusk sequences are shown in Figures 1 and 3. Only a selection of frames from the video sequences are manually annotated. Each pixel in these frames was labelled as one of the 32 candidate classes. We used the same subset of 11 class categories as [8,32] for experimental analysis. We have detector responses for the 5 *thing* classes, namely Car, Sign-Symbol, Pedestrian, Column-Pole, and Bicyclist. A small number of pixels were labelled as *void*, which do not belong to one of these classes and are ignored. The dataset is split into 367 training and 233 test images. To make our experimental setup the same as [8,32], we scaled all the images by a factor of 3.

PASCAL VOC 2009. This dataset was used for the PASCAL Visual Object Category segmentation contest 2009. It contains 14,743 images in all, with 20 foreground (*things*) classes and 1 background (*stuff*) class. We have detector responses for all foreground classes. Each image has an associated annotation file with the bounding boxes and the object class label for each object in the image. A subset of these images are also annotated with pixel-wise segmentation of each object present. We used only these images for training our framework. It contains 749 training, 750 validation, and 750 test images.

3.1 CRF Framework

We now describe the baseline CRF formulation used in our experiments. Note that any CRF formulation based on pixels or segments could have been used. We use the Associative Hierarchical CRF model [16], which combines features at different quantization levels of the image, such as pixels, segments, and is a generalization of commonly used pixel and segment-based CRFs. We have a base layer of variables corresponding to pixels, and a hierarchy of auxiliary variables, which encode mid-level cues from and between segments. Furthermore, it assumes that pixels in the same segment obtained using unsupervised segmentation methods, are highly correlated, but are not required to take the same label. This allows us to incorporate multiple segmentations in a principled approach.

In our experiments we used a two level hierarchy based on pixels and segments. Three segmentations are used for the CamVid dataset and six for the PASCAL VOC 2009 dataset; these were obtained by varying parameters of the MeanShift algorithm [26], similar to [16,32].

Pixel-based potentials. The pixel-based unary potential is identical to that used in [16,32], and is derived from *TextonBoost* [3]. It estimates the probability of a pixel taking a certain label by boosting weak classifiers based on a set of shape filter responses. Shape filters are defined by triplets of feature type, feature cluster, and rectangular region and their response for a given pixel is the number of features belonging to the given cluster in the region placed relative to the given pixel. The most discriminative filters are found using the Joint Boosting algorithm [14]. Details of the learning procedure are given in [3,16]. To enforce local consistency between neighbouring pixels we use the standard contrast sensitive Potts model [22] as the pairwise potential on the pixel level.

Segment-based potentials. We also learn unary potentials for variables in higher layers (*i.e.* layers other than the base layer), which represent segments or super-segments (groups of segments). The segment unary potential is also learnt using the Joint Boosting algorithm [14]. The pairwise potentials in higher layers (*e.g.* pairwise potentials between segments) are defined using a contrast sensitive (based on distance between colour histogram features) Potts model. We refer the reader to [16] for more details on these potentials and the learning procedure.

3.2 Detection-Based Potentials

The object detections are included in the form of a higher order potential over pixels based on detector responses, as detailed in §2.2. The implementation details of this potential are described below. In order to jointly estimate the class category, location, and segmentation of objects, we augment the standard CRF using responses of two of the most successful detectors[5]: (i) histogram-based detector proposed in [17]; and (ii) parts-based detector proposed in [4]. Other detector methods could similarly be incorporated into our framework.

In [17], histograms of multiple features (such as bag of visual words, self-similarity descriptors, SIFT descriptors, oriented edges) were used to train a cascaded classifier

[5] We thank the authors of [4,17] for providing their detections on the PASCAL VOC 2009 dataset.

(a) (b) (c)

Fig. 5. (a) Segmentation without object detectors, (b) Object detection results on this image showing pedestrian and sign/symbol detections, (c) Segmentation using all the detection results. Note that one of the persons (on the left side of the image) is originally labelled as bicyclist (shown in cyan) in (a). This false labelling is corrected in (c) using the detection result. We also show that unary potentials on segments (traffic light on the right), and object detector potentials (traffic light on the left) provide complementary information, thus leading to both the objects being correctly labelled in (c). Some of the regions are labelled incorrectly (the person furthest on the left) perhaps due to a weak detection response. (**Best viewed in colour**)

composed of Support Vector Machines (SVM). The first stage of the cascade is a linear SVM, which proposes candidate object windows and discards all the windows that do not contain an object. The second and third stages are more powerful classifiers using quasi-linear and non-linear SVMs respectively. All the SVMs are trained with ground truth object instances [9]. The negative samples (which are prohibitively large in number) are obtained by bootstrapping each classifier, as follows. Potential object regions are detected in the training images using the classifier. These potential object regions are compared with the ground truth, and a few of the incorrect detections are added to the training data as negative samples. The SVM is then retrained using these negative and the positive ground truth samples.

In [4] each object is composed of a set of deformable parts and a global template. Both the global template and the parts are represented by HOG descriptors [33], but computed at a coarse and fine level respectively. The task of learning the parts and the global template is posed as a latent SVM problem, which is solved by an iterative method. The negative samples are obtained by bootstrapping the classifier, as described above.

Both these methods produce results as bounding boxes around the detected objects along with a score, which represents the likelihood of a box containing an object. A more accurate set of pixels belonging to the detected object is obtained using local foreground and background colour models [28]. In our experiments we observed that the model is robust to change in detector potential parameters. The parameter p_d (from equation (5)) can be set anywhere in the range $10\% - 40\%$. The parameter H_t (which defines the detector threshold, equation (4)) can be set to 0 for most of the SVM-based classifiers. To compensate the bias towards foreground classes the unary potentials of background class(es) were weighted by factor w_b. This bias weight and the detector potential weight w_d were learnt along with the other potential weights on the validation set using the greedy approach presented in [16]. The CRF was solved efficiently using the graph cut based α-expansion algorithm [2,23].

Table 1. We show quantitative results on the CamVid test set on both recall and intersection vs union measures. 'Global' refers to the overall percentage of pixels correctly classified, and 'Average' is the average of the per class measures. Numbers in bold show the best performance for the respective class under each measure. Our method includes detectors trained on the 5 "thing" classes, namely Car, Sign-Symbol, Pedestrian, Column-Pole, Bicyclist. We clearly see how the inclusion of our detector potentials ('Our method') improves over a baseline CRF method ('Without detectors'), which is based on [16]. For the recall measure, we perform better on 8 out of 11 classes, and for the intersection vs measure, we achieve better results on 9 classes. Note that our method was optimized for intersection vs union measure. Results, where available, of previous methods [8, 32] are also shown for reference.

	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Sidewalk	Bicyclist	**Global**	**Average**
Recall[6]													
[8]	46.2	61.9	89.7	68.6	42.9	89.5	**53.6**	46.6	0.7	60.5	22.5	69.1	53.0
[32]	**84.5**	72.6	**97.5**	72.7	34.1	**95.3**	34.2	45.7	8.1	77.6	28.5	**83.8**	59.2
Without detectors	79.3	76.0	96.2	74.6	**43.2**	94.0	40.4	47.0	**14.6**	81.2	31.1	83.1	61.6
Our method	81.5	**76.6**	96.2	**78.7**	40.2	93.9	43.0	**47.6**	14.3	**81.5**	**33.9**	**83.8**	**62.5**
Intersection *vs* Union[7]													
[32]	**71.6**	60.4	**89.5**	58.3	19.4	86.6	**26.1**	35.0	7.2	63.8	22.6	-	49.2
Without detectors	70.0	**63.7**	**89.5**	58.9	17.1	86.3	20.0	**35.8**	9.2	**64.6**	23.1	-	48.9
Our method	71.5	**63.7**	89.4	**64.8**	**19.8**	**86.8**	23.7	35.6	**9.3**	**64.6**	**26.5**	-	**50.5**

3.3 Results

Figures 2, 3 and 5 show qualitative results on the CamVid dataset. Object segmentation approaches do not identify the number of instances of objects, but this information is recovered using our combined segmentation and detection model (from y_d variables, as discussed in §2.3), and is shown in Figure 2. Figure 3 shows the advantage of our soft constraint approach to include detection results. The false positive detection here (shown as the large green box) does not affect the final segmentation, as the other hypotheses based on pixels and segments are stronger. However, a strong detector hypothesis (shown as the purple box) refines the segmentation accurately. Figure 5 highlights the complementary information provided by the object detectors and segment-based potentials. An object falsely missed by the detector (traffic light on the right) is recognized based on the segment potentials, while another object (traffic light on the left) overlooked by the segment potentials is captured by the detector. More details are provided in the figure captions. Quantitative results on the CamVid dataset are shown in Table 1. For the recall measure, our method performs the best on 5 of the classes, and shows near-best ($< 1\%$ difference in accuracy) results on 3 other classes. Accuracy of "things" classes improved by 7% on average. This measure does not consider false positives, and creates a bias towards smaller classes. Therefore, we also provide results with the intersection *vs* union measure in Table 1. We observe that our method shows improved results on almost all the classes in this case.

[6] Defined as $\frac{\text{True Positive}}{\text{True Positive + False Negative}}$.

[7] Defined as $\frac{\text{True Positive}}{\text{True Positive + False Negative + False Positive}}$; also used in PASCAL VOC challenges.

Table 2. Quantitative analysis of VOC 2009 test dataset results [9] using the intersection vs union performance measure. Our method is ranked **third** when compared the 6 best submissions in the 2009 challenge. The method UOCTTI_LSVM-MDPM is based on an object detection algorithm [4] and refines the bounding boxes with a GrabCut style approach. The method BROOKESMSRC_AHCRF is the CRF model used as an example in our work. We perform better than both these baseline methods by 3.1% and 7.3% respectively. Underlined numbers in bold denote the best performance for each class.

	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining table	Dog	Horse	Motor bike	Person	Potted plant	Sheep	Sofa	Train	TV/monitor	**Average**
BONN_SVM-SEGM	**83.9**	64.3	21.8	21.7	**32.0**	**40.2**	**57.3**	49.4	**38.8**	5.2	**28.5**	22.0	19.6	33.6	45.5	33.6	27.3	**40.4**	18.1	33.6	**46.1**	**36.3**
CVC_HOCRF	80.2	**67.1**	**26.6**	**30.3**	31.6	30.0	44.5	41.6	25.2	5.9	27.8	11.0	23.1	**40.5**	**53.2**	32.0	22.2	37.4	**23.6**	40.3	30.2	34.5
UOCTTI_LSVM-MDPM	78.9	35.3	22.5	19.1	23.5	36.2	41.2	50.1	11.7	8.9	**28.5**	1.4	5.9	24.0	35.3	33.4	**35.1**	27.7	14.2	34.1	41.8	29.0
NECUIUC_CLS-DTCT	81.8	41.9	23.1	22.4	22.0	27.8	43.2	**51.8**	25.9	4.5	18.5	18.0	**23.5**	26.9	36.6	**34.8**	8.8	28.3	14.0	35.5	34.7	29.7
LEAR_SEGDET	79.1	44.6	15.5	20.5	13.3	28.8	29.3	35.8	25.4	4.4	20.3	1.3	16.4	28.2	30.0	24.5	12.2	31.5	18.3	28.8	31.9	25.7
BROOKESMSRC_AHCRF	79.6	48.3	6.7	19.1	10.0	16.6	32.7	38.1	25.3	5.5	9.4	25.1	13.3	12.3	35.5	20.7	13.4	17.1	18.4	37.5	36.4	24.8
Our method	81.2	46.1	15.4	24.6	20.9	36.9	50.0	43.9	28.4	**11.5**	18.2	**25.4**	14.7	25.1	37.7	34.1	27.7	29.6	18.4	**43.8**	40.8	32.1

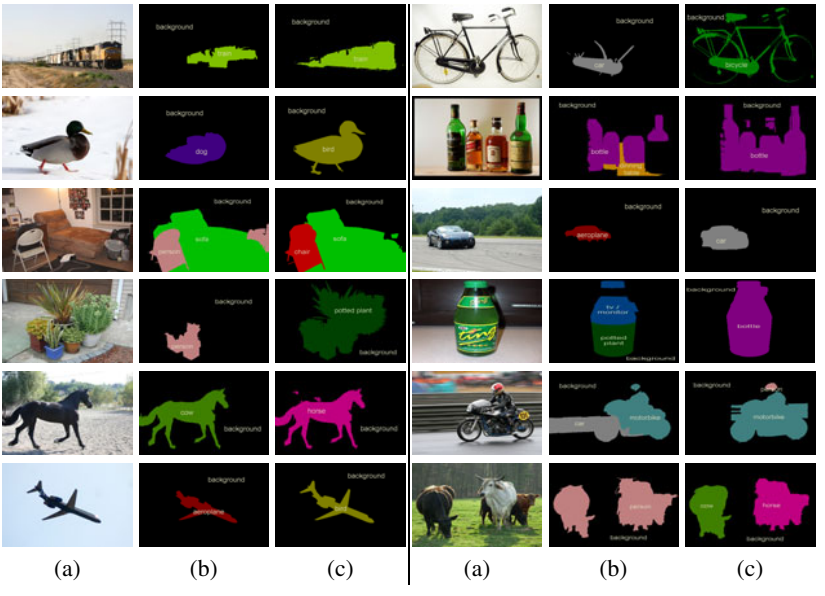


Fig. 6. (a) Original test image from PASCAL VOC 2009 dataset [9], (b) The labelling obtained by [16] without object detectors, (c) The labelling provided by our method which includes detector based potentials. Note that no groundtruth is publicly available for test images in this dataset. Examples shown in the first five rows illustrate how detector potentials not only correctly identify the object, but also provide very precise object boundaries, e.g. bird (second row), car (third row). Some failure cases are shown in the last row. This was caused by a missed detection or incorrect detections that are very strong and dominate all other potentials. (**Best viewed in colour**)

Qualitative results on PASCAL VOC 2009 test set are shown in Figure 6. Our approach provides very precise object boundaries and recovers from many failure cases. For example, bird (second row), car (third row), potted plant (fourth row) are not only correctly identified, but also segmented with accurate object boundaries. Quantitative

results on this dataset are provided in Table 2. We compare our results with the 6 best submissions from the 2009 challenge, and achieve the third best average accuracy. Our method shows the best performance in 3 categories, and a close 2nd/3rd in 10 others. Note that using the detector based work (UOCTTI_LSVM-MDPM: 29.0%) and pixel-based method (BROOKESMSRC_AHCRF: 24.8%) as examples in our framework, we improve the accuracy to 32.1%. Both the BONN [34] and CVC [35] methods can be directly placed in our work, and should lead to an increase in performance.

4 Summary

We have presented a novel framework for a principled integration of detectors with CRFs. Unlike many existing methods, our approach supports the robust handling of occluded objects and false detections in an efficient and tractable manner. We believe the techniques described in this paper are of interest to many working in the problem of object class segmentation, as they allow the efficient integration of any detector response with any CRF. The benefits of this approach can be seen in the results; our approach consistently demonstrated improvement over the baseline methods, under the intersection *vs* union measure.

This work increases the expressibility of CRFs and shows how they can be used to identify object instances, and answer the questions: "*What object instance is this?*", "*Where is it?*", and "*How many of them?*", bringing us one step closer to complete scene understanding.

References

1. Barrow, H.G., Tenenbaum, J.M.: Computational vision. IEEE 69, 572–595 (1981)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23, 1222–1239 (2001)
3. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
4. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
5. Hoiem, D., Efros, A., Hebert, M.: Closing the loop on scene interpretation. In: CVPR (2008)
6. Gould, S., Gao, T., Koller, D.: Region-based segmentation and object detection. In: NIPS (2009)
7. Li, L.-J., Socher, R., Fei-Fei, L.: Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: CVPR (2009)
8. Brostow, G.J., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
9. Everingham, M., et al.: The PASCAL Visual Object Classes Challenge (VOC) Results (2009)
10. Adelson, E.H.: On seeing stuff: the perception of materials by humans and machines. In: SPIE, vol. 4299, pp. 1–12 (2001)

11. Forsyth, D.A., et al.: Finding pictures of objects in large collections of images. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, Part II, vol. 1065, pp. 335–360. Springer, Heidelberg (1996)
12. 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)
13. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV (2007)
14. Torralba, A., Murphy, K., Freeman, W.T.: Sharing features: Efficient boosting procedures for multiclass object detection. In: CVPR, vol. 2, pp. 762–769 (2004)
15. Tu, Z., et al.: Image parsing: Unifying segmentation, detection, and recognition. IJCV (2005)
16. Ladicky, L., Russell, C., Kohli, P., Torr, P.H.S.: Associative hierarchical crfs for object class image segmentation. In: ICCV (2009)
17. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV (2009)
18. Wojek, C., Schiele, B.: A dynamic conditional random field model for joint labeling of object and scene classes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 733–747. Springer, Heidelberg (2008)
19. Larlus, D., Jurie, F.: Combining appearance models and markov random fields for category level object segmentation. In: CVPR (2008)
20. Gu, C., Lim, J., Arbelaez, P., Malik, J.: Recognition using regions. In: CVPR (2009)
21. Winn, J., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: CVPR (2006)
22. Boykov, Y., Jolly, M.-P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV, vol. 1, pp. 105–112 (2001)
23. Kohli, P., Ladicky, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. In: CVPR (2008)
24. He, X., Zemel, R.S., Carreira-Perpiñán, M.Á.: Learning and incorporating top-down cues in image segmentation. In: CVPR, vol. 2, pp. 695–702 (2004)
25. Yang, L., Meer, P., Foran, D.J.: Multiple class segmentation using a unified framework over mean-shift patches. In: CVPR (2007)
26. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space. PAMI (2002)
27. Shi, J., Malik, J.: Normalized cuts and image segmentation. PAMI 22, 888–905 (2000)
28. Rother, C., Kolmogorov, V., Blake, A.: GrabCut. In: SIGGRAPH, pp. 309–314 (2004)
29. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. PAMI 26, 1124–1137 (2004)
30. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. In: CVPR (2004)
31. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. PAMI 28, 1568–1583 (2006)
32. Sturgess, P., Alahari, K., Ladicky, L., Torr, P.H.S.: Combining appearance and structure from motion features for road scene understanding. In: BMVC (2009)
33. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
34. Li, F., Carreira, J., Sminchisescu, C.: Object recognition as ranking holistic figure-ground hypotheses. In: CVPR (2010)
35. Gonfaus, J.M., Boix, X., van de Weijer, J., Bagdanov, A.D., Serrat, J., Gonzalez, J.: Harmony potentials for joint classification and segmentation. In: CVPR (2010)

Visual Recognition with Humans in the Loop

Steve Branson[1], Catherine Wah[1], Florian Schroff[1], Boris Babenko[1], Peter Welinder[2], Pietro Perona[2], and Serge Belongie[1]

[1] University of California, San Diego
{sbranson,cwah,gschroff,bbabenko,sjb}@cs.ucsd.edu
[2] California Institute of Technology
{welinder,perona}@caltech.edu

Abstract. We present an interactive, hybrid human-computer method for object classification. The method applies to classes of objects that are recognizable by people with appropriate expertise (*e.g.*, animal species or airplane model), but not (in general) by people without such expertise. It can be seen as a visual version of the *20 questions game*, where questions based on simple visual attributes are posed interactively. The goal is to identify the true class while minimizing the number of questions asked, using the visual content of the image. We introduce a general framework for incorporating almost any off-the-shelf multi-class object recognition algorithm into the visual 20 questions game, and provide methodologies to account for imperfect user responses and unreliable computer vision algorithms. We evaluate our methods on *Birds-200*, a difficult dataset of 200 tightly-related bird species, and on the *Animals With Attributes* dataset. Our results demonstrate that incorporating user input drives up recognition accuracy to levels that are good enough for practical applications, while at the same time, computer vision reduces the amount of human interaction required.

1 Introduction

Multi-class object recognition has undergone rapid change and progress over the last decade. These advances have largely focused on types of object categories that are easy for humans to recognize, such as motorbikes, chairs, horses, bottles, *etc.* Finer-grained categories, such as specific types of motorbikes, chairs, or horses are more difficult for humans and have received comparatively little attention. One could argue that object recognition as a field is simply not mature enough to tackle these types of finer-grained categories. Performance on basic-level categories is still lower than what people would consider acceptable for practical applications (state-of-the-art accuracy on Caltech-256[1] is $\approx 45\%$, and $\approx 28\%$ in the 2009 VOC detection challenge [2]. Moreover, the number of object categories in most object recognition datasets is still fairly low, and increasing the number of categories further is usually detrimental to performance [1].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 438–451, 2010.

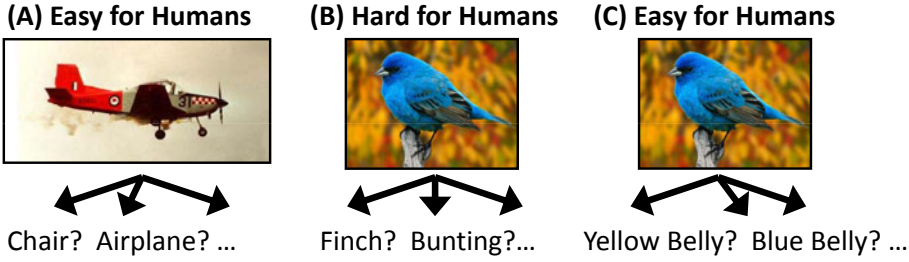


Fig. 1. Examples of classification problems that are easy or hard for humans. While basic-level category recognition (left) and recognition of low-level visual attributes (right) are easy for humans, most people struggle with finer-grained categories (middle). By defining categories in terms of low-level visual properties, hard classification problems can be turned into a sequence of easy ones.

On the other hand, recognition of finer-grained subordinate categories is an important problem to study – it can help people recognize types of objects they don't yet know how to identify. We believe a hybrid human-computer recognition method is a practical intermediate solution toward applying contemporary computer vision algorithms to these types of problems. Rather than trying to solve object recognition entirely, we take on the objective of minimizing the amount of human labor required. As research in object recognition progresses, tasks will become increasingly automated, until eventually we will no longer need humans in the loop. This approach differs from some of the prevailing ways in which people approach research in computer vision, where researchers begin with simpler and less realistic datasets and progressively make them more difficult and realistic as computer vision improves (*e.g.*, Caltech-4 → Caltech-101 → Caltech-256). The advantage of the human-computer paradigm is that we can provide usable services to people in the interim-period where computer vision is still unsolved. This may help increase demand for computer vision, spur data collection, and provide solutions for the types of problems people outside the field want solved.

In this work, our goal is to provide a simple framework that makes it as effortless as possible for researchers to plug their existing algorithms into the human-computer framework and use humans to drive up performance to levels that are good enough for real-life applications. Implicit to our model is the assumption that lay-people generally cannot recognize finer-grained categories (*e.g.*, Myrtle Warbler, Thruxton Jackaroo, *etc.*) due to imperfect memory or limited experiences; however, they do have the fundamental visual capabilities to recognize the parts and attributes that collectively make recognition possible (see Fig. 1). By contrast, computers lack many of the fundamental visual capabilities that humans have, but have perfect memory and are able to pool knowledge collected from large groups of people. Users interact with our system by answering simple yes/no or multiple choice questions about an image or object, as shown in Fig. 2. Similar to the *20-questions game*[1], we observe that the

[1] See for example http://20q.net

Fig. 2. Examples of the visual 20 questions game on the 200 class Bird dataset. Human responses (shown in red) to questions posed by the computer (shown in blue) are used to drive up recognition accuracy. In the left image, computer vision algorithms can guess the bird species correctly without any user interaction. In the middle image, computer vision reduces the number of questions to 2. In the right image, computer vision provides little help.

number of questions needed to classify an object from a database of C classes is usually $O(\log C)$ (when user responses are accurate), and can be faster when computer vision is in the loop. Our method of choosing the next question to ask uses an information gain criterion and can deal with noisy (probabilistic) user responses. We show that it is easy to incorporate any computer vision algorithm that can be made to produce a probabilistic output over object classes.

Our experiments in this paper focus on bird species categorization, which we take to be a representative example of recognition of tightly-related categories. The bird dataset contains 200 bird species and over 6,000 images. We believe that similar methodologies will apply to other object domains.

The structure of the paper is as follows: In Section 2, we discuss related work. In Section 3, we define the hybrid human-computer problem and basic algorithm, which includes methodologies for modeling noisy user responses and incorporating computer vision into the framework. We describe our datasets and implementation details in Section 4, and present empirical results in Section 5.

2 Related Work

Recognition of tightly related categories is still an open area in computer vision, although there has been success in a few areas such as book covers and movie posters (*e.g.*, rigid, mostly flat objects [3]). The problem is challenging because the number of object categories is larger, with low interclass variance, and variability in pose, lighting, and background causes high intraclass variance. Ability to exploit domain knowledge and cross-category patterns and similarities becomes increasingly important.

There exist a variety of datasets related to recognition of tightly-related categories, including Oxford Flowers 102 [4], UIUC Birds [5], and STONEFLY9 [6]. While these works represent progress, they still have shortcomings in scaling to large numbers of categories, applying to other types of object domains, or achieving performance levels that are good enough for real-world applications. Perhaps most similar in spirit to our work is the Botanist's Field Guide [7], a system for plant species recognition with hundreds of categories and tens of

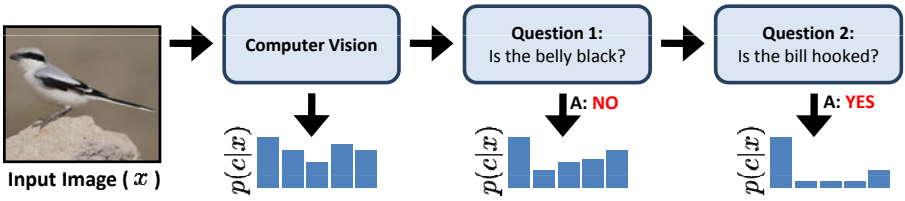


Fig. 3. Visualization of the basic algorithm flow. The system poses questions to the user, which along with computer vision, incrementally refine the probability distribution over classes.

thousands of images. One key difference is that their system is intended primarily for experts, and requires plant leaves to be photographed in a controlled manner at training and test time, making segmentation and pose normalization possible. In contrast, all of our training and testing images are obtained from Flickr in unconstrained settings (see Fig. 4), and the system is intended to be used by lay people.

There exists a multitude of different areas in computer science that interleave vision, learning, or other processing with human input. Relevance feedback [8] is a method for interactive image retrieval, in which users mark the relevance of image search results, which are in turn used to create a refined search query. Active learning algorithms [9,10,11] interleave training a classifier with asking users to label examples, where the objective is to minimize the total number of labeling tasks. Our objectives are somewhat similar, except that we are querying information at runtime rather than training time. Expert systems [12,13] involve construction of a knowledge base and inference rules that can help non-experts solve a problem. Our approach differs due to the added ability to observe image pixels as an additional source of information. Computationally, our method also has similarities to algorithms based on information gain, entropy calculation, and decision trees [14,15,16].

Finally, a lot of progress has been made on trying to scale object recognition to large numbers of categories. Such approaches include using class taxonomies [17,18], feature sharing [19], error correcting output codes (ECOC) [20], and attribute based classification methods [21,22,23]. All of these methods could be easily plugged into our framework to incorporate user interaction.

3 Visual Recognition with Humans in the Loop

Given an image x, our goal is to determine the true object class $c \in \{1...C\}$ by posing questions based on visual properties that are easy for the user to answer (see Fig. 1). At each step, we aim to exploit the visual content of the image and the current history of question responses to intelligently select the next question. The basic algorithm flow is summarized in Fig. 3.

Let $\mathcal{Q} = \{q_1...q_n\}$ be a set of possible questions (*e.g.*, IsRed?, HasStripes?, *etc.*), and $\mathcal{A}_i$ be the set of possible answers to q_i. The user's answer is some

Algorithm 1. Visual 20 Questions Game

1: $U^0 \leftarrow \emptyset$
2: **for** $t = 1$ to 20 **do**
3: $\quad j(t) = \max_k \mathrm{I}(c; u_k | x, U^{t-1})$
4: $\quad$ Ask user question $q_{j(t)}$, and $U^t \leftarrow U^{t-1} \cup u_{j(t)}$.
5: **end for**
6: Return class $c^* = \max_c p(c|x, U^t)$

random variable $a_i \in \mathcal{A}_i$. We also allow users to qualify each response with a confidence value $r_i \in \mathcal{V}$, (*e.g.*, $\mathcal{V} = \{\mathsf{Guessing}, \mathsf{Probably}, \mathsf{Definitely}\}$). The user's response is then a pair of random variables $u_i = (a_i, r_i)$.

At each time step t, we select a question $q_{j(t)}$ to pose to the user, where $j(t) \in 1...n$. Let $j \in \{1...n\}^T$ be an array of T indices to questions we will ask the user. $U^{t-1} = \{u_{j(1)}...u_{j(t-1)}\}$ is the set of responses obtained by time step $t-1$. We use maximum information gain as the criterion to select $q_{j(t)}$. Information gain is widely used in decision trees (*e.g.* [15]) and can be computed from an estimate of $p(c|x, U^{t-1})$.

We define $\mathrm{I}(c; u_i | x, U^{t-1})$, the expected information gain of posing the additional question q_i, as follows:

$$\mathrm{I}(c; u_i | x, U^{t-1}) = \mathbb{E}_u \left[\mathrm{KL} \left(p(c|x, u_i \cup U^{t-1}) \parallel p(c|x, U^{t-1}) \right) \right] \quad (1)$$

$$= \sum_{u_i \in \mathcal{A}_i \times \mathcal{V}} p(u_i | x, U^{t-1}) \big(\mathrm{H}(c|x, u_i \cup U^{t-1}) - \mathrm{H}(c|x, U^{t-1}) \big) \quad (2)$$

where $\mathrm{H}(c|x, U^{t-1})$ is the entropy of $p(c|x, U^{t-1})$

$$\mathrm{H}(c|x, U^{t-1}) = -\sum_{c=1}^{C} p(c|x, U^{t-1}) \log p(c|x, U^{t-1}) \quad (3)$$

The general algorithm for interactive object recognition is shown in Algorithm 1. In the next sections, we describe in greater detail methods for modeling user responses and different methods for incorporating computer vision algorithms, which correspond to different ways to estimate $p(c|x, U^{t-1})$.

3.1 Incorporating Computer Vision

When no computer vision is involved it is possible to pre-compute a decision tree that defines which question to ask for every possible sequence of user responses. With computer vision in the loop, however, the best questions depend dynamically on the contents of the image.

In this section, we propose a simple framework for incorporating any multi-class object recognition algorithm that produces a probabilistic output over classes. We can compute $p(c|x, U)$, where U is any arbitrary sequence of responses, as follows:

$$p(c|x, U) = \frac{p(U|c, x) p(c|x)}{Z} = \frac{p(U|c) p(c|x)}{Z} \quad (4)$$

where $Z = \sum_c p(U|c)p(c|x)$. Here, we make the assumption that $p(U|c, x) = p(U|c)$; effectively this assumes that the types of noise or randomness that we see in user responses is class-dependent and not image-dependent. We can still accommodate variation in responses due to user error, subjectivity, external factors, and intraclass variance; however we throw away some image-related information (for example, we lose ability to model a change in the distribution of user responses as a result of a computer-vision-based estimate of object pose).

In terms of computation, we estimate $p(c|x)$ using a classifier trained offline (more details in Section 4.3). Upon receiving an image, we run the classifier once at the beginning of the process, and incrementally update $p(c|x, U)$ by gathering more answers to questions from the user. One could imagine a system where a learning algorithm is invoked several times during the process; as categories are weeded out by answers, the system would use a more tuned classifier to update the estimate of $p(c|x)$. However, our preliminary experiments with such methods did not show an advantage[2]. Note that when no computer vision is involved, we simply replace $p(c|x)$ with a prior $p(c)$.

3.2 Modeling User Responses

Recall that for each question we may also ask a corresponding confidence value from the user, which may be necessary when an attribute cannot be determined (for example, when the associated part(s) are not visible). We assume that the questions are answered independently given the category:

$$p(U^{t-1}|c) = \prod_{i}^{t-1} p(u_i|c) \tag{5}$$

The same assumption allows us to express $p(u_i|x, U^{t-1})$ in Equation 2 as

$$p(u_i|x, U^{t-1}) = \sum_{c=1}^{C} p(u_i|c)p(c|x, U^{t-1}) \tag{6}$$

It may also be possible to use a more sophisticated model in which we estimate a full joint distribution for $p(U^{t-1}|c)$; in our preliminary experiments this approach did not work well due to insufficient training data.

To compute $p(u_i|c) = p(a_i, r_i|c) = p(a_i|r_i, c)p(r_i|c)$, we assume that $p(r_i|c) = p(r_i)$. Next, we compute each $p(a_i|r_i, c)$ as the posterior of a multinomial distribution with Dirichlet prior $\mathrm{Dir}\big(\alpha_r p(a_i|r_i) + \alpha_c p(a_i|c)\big)$, where α_r and α_c are constants, $p(a_i|r_i)$ is a global attribute prior, and $p(a_i|c)$ is estimated by pooling together certainty labels. In practice, we use a larger prior term for *Guessing* than *Definitely*, $\alpha_{guess} > \alpha_{def}$, which effectively down weights the importance of any response with certainty level *Guessing*.

[2] See supplementary material (`http://www.vision.caltech.edu/visipedia/birds200.html`) for more details.

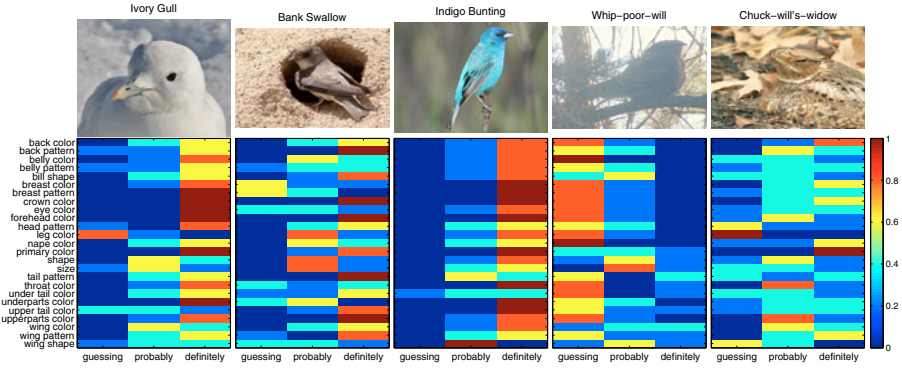


Fig. 4. Examples of user responses for each of the 25 attributes. The distribution over {*Guessing, Probably, Definitely*} is color coded with blue denoting 0% and red denoting 100% of the five answers per image attribute pair.

4 Datasets and Implementation Details

In this section we provide a brief overview of the datasets we used, methods used to construct visual questions, computer vision algorithms we tested, and parameter settings.

4.1 Birds-200 Dataset

Birds-200 is a dataset of 6033 images over 200 bird species, such as Myrtle Warblers, Pomarine Jaegars, and Black-footed Albatrosses – classes that cannot usually be identified by non-experts. In many cases, different bird species are nearly visually identical (see Fig. 8).

We assembled a set of 25 visual questions (list shown in Fig. 4), which encompass 288 binary attributes (*e.g.*, the question HasBellyColor can take on 15 different possible colors). The list of attributes was extracted from whatbird.com[3], a bird field guide website.

We collected "deterministic" class-attributes by parsing attributes from whatbird.com. Additionally, we collected data of how non-expert users respond to attribute questions via a Mechanical Turk interface. To minimize the effects of user subjectivity and error, our interface provides prototypical images of each possible attribute response. The reader is encouraged to look at the supplementary material for screenshots of the question answering user-interface and example images of the dataset.

Fig. 4 shows a visualization of the types of user response results we get on the Birds-200 dataset. It should be noted that the uncertainty of the user responses strongly correlates with the parts that are visible in an image as well as overall difficulty of the corresponding bird species.

[3] `http://www.whatbird.com/`

When evaluating performance, test results are generated by randomly selecting a response returned by an MTurk user for the appropriate test image.

4.2 Animals with Attributes

We also tested performance on the Animals With Attributes (AwA) [21], a dataset of 50 animal classes and 85 binary attributes. We consider this dataset less relevant than birds (because classes are recognizable by non-experts), and therefore do not focus as much on this dataset.

4.3 Implementation Details and Parameter Settings

For both datasets, our computer vision algorithms are based on Andrea Vedaldi's publicly available source code [24], which combines vector-quantized geometric blur and color/gray SIFT features using spatial pyramids, multiple kernel learning, and per-class 1-vs-all SVMs. We added features based on full image color histograms and vector-quantized color histograms. For each classifier we used Platt scaling [25] to learn parameters for $p(c|x)$ on a validation set. We used 15 training examples for each Birds-200 class and 30 training examples for each AwA class. Bird training and testing images are roughly cropped.

Additionally, we compare performance to a second computer vision algorithm based on attribute classifiers, which we train using the same features/training code, with positive and negative examples set using whatbird.com attribute labels. We combined attribute classifiers into per-class probabilities $p(c|x)$ using the method described in [21].

For estimating user response statistics on the Birds-200 dataset, we used $\alpha_{guess} = 64$, $\alpha_{prob} = 16$, $\alpha_{def} = 8$, and $\alpha_c = 8$ (see Section 3.2).

5 Experiments

In this section, we provide experimental results and analysis of the hybrid-human computer classification paradigm. Due to space limitations, our discussion focuses on the Birds dataset. We include results (see Fig. 9) from which the user can verify that trends are similar on Birds-200 and AwA, and we include additional results on AwA in the supplementary material.

5.1 Measuring Performance

We use two main methodologies for measuring performance, which correspond to two different possible user-interfaces:

- **Method 1:** We ask the user exactly T questions, predict the class with highest probability, and measure the percent of the time that we are correct.
- **Method 2:** After asking each question, we present a small gallery of images of the highest probability class, and allow the user to stop the system early. We measure the average number of questions asked per test image.

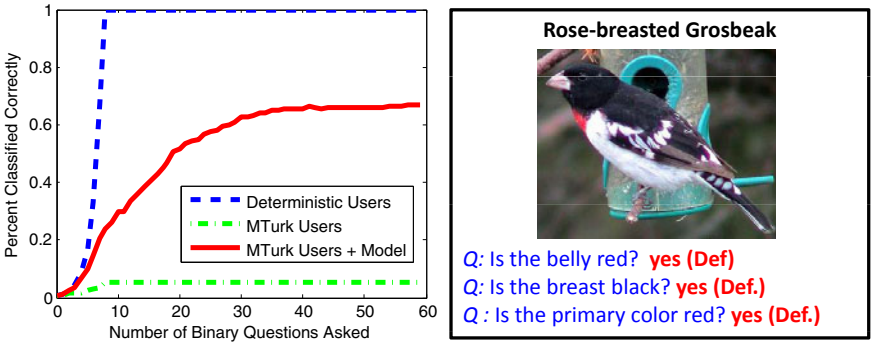


Fig. 5. Different Models of User Responses: *Left:* Classification performance on Birds-200 (Method 1) without computer vision. Performance rises quickly (blue curve) if users respond deterministically according to whatbird.com attributes. MTurk users respond quite differently, resulting in low performance (green curve). A learned model of MTurk responses is much more robust (red curve). *Right:* A test image where users answer several questions incorrectly and our model still classifies the image correctly.

For the second method, we assume that people are perfect verifiers, *e.g.*, they will stop the system if and only if they have been presented with the correct class. While this is not always possible in reality, there is some trade-off between classification accuracy and amount of human labor, and we believe that these two metrics collectively capture the most important considerations.

5.2 Results

In this section, we present our results and discuss some interesting trends toward understanding the visual 20 questions classification paradigm.

User Responses are Stochastic: In Fig. 5, we show the effects of different models of user responses without using any computer vision. When users are assumed to respond deterministically in accordance with the attributes from whatbird.com, performance rises quickly to 100% within 8 questions (roughly $\log_2(200)$). However, this assumption is not realistic; when testing with responses from Mechanical Turk, performance saturates at around 5%. Low performance caused by subjective answers are unavoidable (*e.g.*, perception of the color brown vs. the color buff), and the probability of the correct class drops to zero after any inconsistent response. Although performance is 10 times better than random chance, it renders the system useless. This demonstrates a challenge for existing field guide websites. When our learned model of user responses (see Section 3.2) is incorporated, performance jumps to 66% due to the ability to tolerate a reasonable degree of error in user responses (see Fig. 5 for an example). Nevertheless, stochastic user responses increase the number of questions required to achieve a given accuracy level, and some images can never be classified correctly, even when asking all possible questions. In Section 5.2, we discuss the reasons why performance saturates at lower than 100% performance.

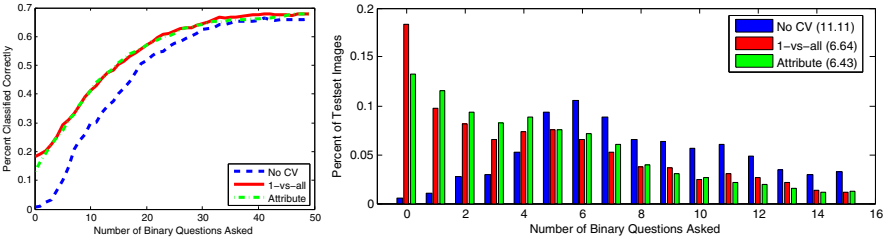


Fig. 6. Performance on Birds-200 when using computer vision: Left Plot: comparison of classification accuracy (Method 1) with and without computer vision when using MTurk user responses. Two different computer vision algorithms are shown, one based on per-class 1-vs-all classifiers and another based on attribute classifiers. Right plot: the number of questions needed to identify the true class (Method 2) drops from 11.11 to 6.43 on average when incorporating computer vision.

Computer Vision Reduces Manual Labor: The main benefit of computer vision occurs due to reduction in human labor (in terms of the number of questions a user has to answer). In Fig. 6, we see that computer vision reduces the average number of yes/no questions needed to identify the true bird species from 11.11 to 6.43 using responses from MTurk users. Without computer vision, the distribution of question counts is bell-shaped and centered around 6 questions. When computer vision is incorporated, the distribution peaks at 0 questions but is more heavy-tailed, which suggests that computer vision algorithms are often good at recognizing the "easy" test examples (examples that are sufficiently similar to the training data), but provide diminishing returns toward classifying the harder examples that are not sufficiently similar to training data. As a result, computer vision is more effective at reducing the average amount of time than reducing the time spent on the most difficult images.

User Responses Drive Up Performance: An alternative way of interpreting the results is that user responses drive up the accuracy of computer vision algorithms. In Fig. 6, we see that user responses improve overall performance from $\approx 19\%$ (using 0 questions) to $\approx 66\%$.

Computer Vision Improves Overall Performance: Even when users answer all questions, performance saturates at a higher level when using computer vision ($\approx 69\%$ vs. $\approx 66\%$, see Fig. 6). The left image in Fig. 7 shows an example of an image classified correctly using computer vision, which is not classified correctly without computer vision, even after asking 60 questions. In this example, some visually salient features like the long neck are not captured in our list of visual attribute questions. The features used by our vision algorithms also capture other cues (such as global texture statistics) that are not well-represented in our list of attributes (which capture mostly color and part-localized patterns).

Different Questions Are Asked With and Without Computer Vision: In general, the information gain criterion favors questions that 1) can be answered reliably, and 2) split the set of possible classes roughly in half. Questions

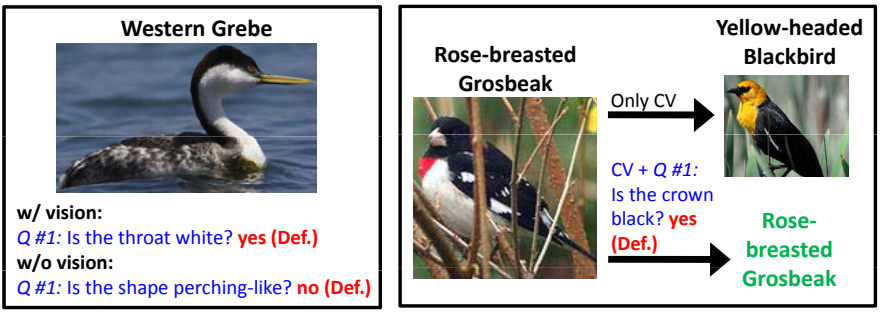


Fig. 7. Examples where computer vision and user responses work together: *Left:* An image that is only classified correctly when computer vision is incorporated. Additionally, the computer vision based method selects the question HasThroatColorWhite, a different and more relevant question than when vision is not used. In the right image, the user response to HasCrownColorBlack helps correct computer vision when its initial prediction is wrong.

like HasShapePerchingLike, which divide the classes fairly evenly, and HasUnderpartsColorYellow, which tends to be answered reliably, are commonly chosen.

When computer vision is incorporated, the likelihood of classes change and different questions are selected. In the left image of Fig. 7, we see an example where a different question is asked with and without computer vision, which allows the system to find the correct class using one question.

Recognition is Not Always Successful: According the the Cornell Ornithology Website[4], the four keys to bird species recognition are 1) size and shape, 2) color and pattern, 3) behavior, and 4) habitat. Bird species classification is a difficult problem and is not always possible using a single image. One potential advantage of the visual 20 questions paradigm is that other contextual sources of information such as behavior and habitat can easily be incorporated as additional questions.

Fig. 8 illustrates some example failures. The most common failure conditions occur due to 1) classes that are nearly visually identical, 2) images of poor viewpoint or low resolution, such that some parts are not visible, 3) significant mistakes made by MTurkers, or 4) inadequacies in the set of attributes we used.

1-vs-all Vs. Attribute-Based Classification: In general, 1-vs-all classifiers slightly outperform attribute-based classifiers; however, they converge to similar performance as the number of question increases, as shown in Fig. 6 and 9. The features we use (kernelized and based on bag-of-words) may not be well suited to the types of attributes we are using, which tend to be localized and associated with a particular part. One potential advantage of attribute-based methods is computational scalability when the number of classes increases; whereas 1-vs-all methods always require C classifiers, the number of attribute classifiers can

[4] http://www.allaboutbirds.org/NetCommunity/page.aspx?pid=1053

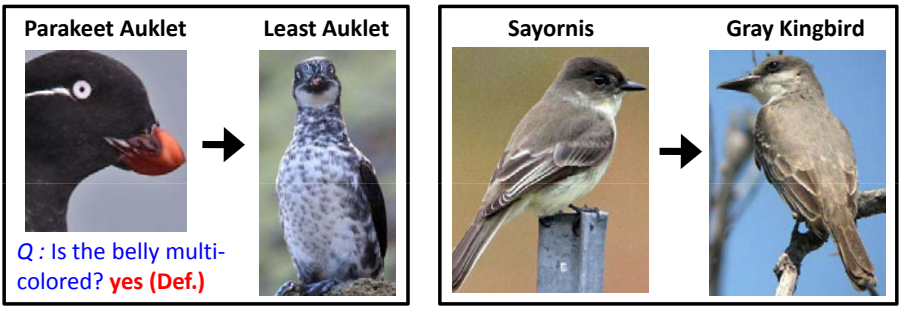


Fig. 8. Images that are misclassified by our system: *Left:* The Parakeet Auklet image is misclassified due to a cropped image, which causes an incorrect answer to the belly pattern question (the Parakeet Auklet has a plain, white belly, see Fig. 2). *Right:* The Sayornis and Gray Kingbird are commonly confused due to visual similarity.

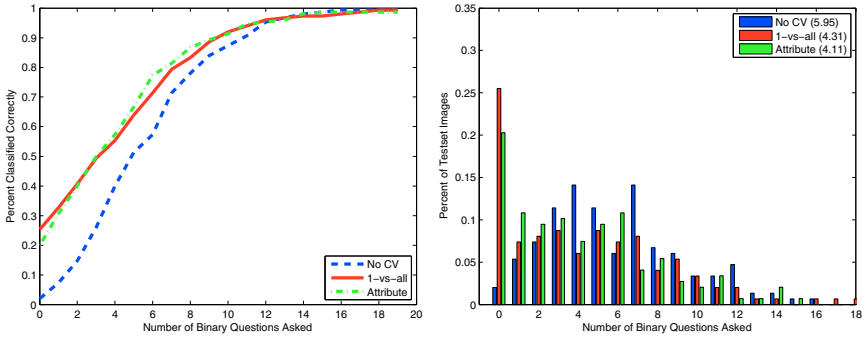


Fig. 9. Performance on Animals With Attributes: Left Plot: Classification performance (Method 1), simulating user responses using soft class-attributes (see [21]). Right Plot: The required number of questions needed to identify the true class (Method 2) drops from 5.94 to 4.11 on average when incorporating computer vision.

be varied in order to trade-off accuracy and computation time. The table below displays the average number of questions needed (Method 1) on the Birds dataset using different number of attribute classifiers (which were selected randomly):

200 (1-vs-all)	288 attr.	100 attr.	50 attr.	20 attr.	10 attr.
6.43	6.72	7.01	7.67	8.81	9.52

6 Conclusion

Object recognition remains a challenging problem for computer vision. Furthermore, recognizing tightly related categories in one shot is difficult even for humans without proper expertise. Our work attempts to leverage the power of both

human recognition abilities and that of computer vision. We presented a simple way of designing a hybrid human-computer classification system, which can be used in conjunction with a large variety of computer vision algorithms. Our results show that user input significantly drives up performance; while it may take many years before object recognition algorithms achieve reasonable performance on their own, incorporating human input can produce usable recognition systems. On the other hand, having computer vision in the loop reduces the amount of required human labor to successfully classify an image. Finally, we showed that incorporating models of stochastic user responses leads to much better reliability in comparison to deterministic field guides generated by experts.

We believe our work opens the door to many interesting sub-problems. The most obvious next step is to explore other types of domains. While we were able to extract a set of reasonable attributes/questions for the bird dataset, this may be more difficult for other domains; one possible topic for future work is to find a more principled way of discovering a set of useful questions. Alternative types of user input, such as asking the user to click on the location of certain parts, could also be investigated. Lastly, while we used off-the-shelf computer vision algorithms in this work, it may be possible to improve them to better suit the challenges of tightly-related category recognition, such as algorithms that incorporate a part-based model.

Acknowledgments

Funding for this work was provided by NSF CAREER Grant #0448615, NSF Grant AGS-0941760, ONR MURI Grant N00014-06-1-0734, ONR MURI Grant #N00014-08-1-0638, Google Research Award. The authors would like to give special thanks to Takeshi Mita for his efforts in constructing the birds dataset.

References

1. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
2. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL VOC Challenge 2009 Results (2009)
3. Nister, D., Stewenius, H.: Recognition with a vocabulary tree. In: CVPR (2006)
4. Nilsback, M., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conf. on Comp. Vision, Graphics & Image Proc., pp. 722–729 (2008)
5. Lazebnik, S., Schmid, C., Ponce, J.: A maximum entropy framework for part-based texture and object recognition. In: ICCV, vol. 1, pp. 832–838 (2005)
6. Martınez-Munoz, et al.: Dictionary-free categorization of very similar objects via stacked evidence trees. In: CVPR (2009)
7. Belhumeur, P., Chen, D., Feiner, S., Jacobs, D., Kress, W., Ling, H., Lopez, I., Ramamoorthi, R., Sheorey, S., White, S., Zhang, L.: Searching the world's herbaria: A system for visual identification of plant species. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 116–129. Springer, Heidelberg (2008)

8. Zhou, X., Huang, T.: Relevance feedback in image retrieval: A comprehensive review. Multimedia Systems 8, 536–544 (2003)
9. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. JMLR 2, 45–66 (2002)
10. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Active learning with gaussian processes for object categorization. In: ICCV, pp. 1–8 (2007)
11. Holub, A., Perona, P., Burl, M.: Entropy-based active learning for object recognition. In: Workshop on Online Learning for Classification (OLC), pp. 1–8 (2008)
12. Neapolitan, R.E.: Probabilistic reasoning in expert systems: theory and algorithms. John Wiley & Sons, Inc., New York (1990)
13. Beynon, M., Cosker, D., Marshall, D.: An expert system for multi-criteria decision making using Dempster Shafer theory. Expert Systems with Applications 20 (2001)
14. Tsang, S., Kao, B., Yip, K., Ho, W., Lee, S.: Decision trees for uncertain data. In: International Conference on Data Engineering, ICDE (2009)
15. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
16. Dembo, A., Cover, T., Thomas, J.: Information theoretic inequalities. IEEE Transactions on Information Theory 37, 1501–1518 (1991)
17. Sivic, J., Russell, B., Zisserman, A., Freeman, W., Efros, A.: Unsupervised discovery of visual object class hierarchies. In: CVPR, pp. 1–8 (2008)
18. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. In: CVPR, pp. 1–8 (2008)
19. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: CVPR, vol. 2 (2004)
20. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286 (1995)
21. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
22. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
23. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and Simile Classifiers for Face Verification. In: ICCV (2009)
24. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV (2009)
25. Platt, J.: Probabilities for SV machines. In: NIPS, pp. 61–74 (1999)

Localizing Objects While Learning Their Appearance

Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari

Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland
{deselaers,bogdan,ferrari}@vision.ee.ethz.ch

Abstract. Learning a new object class from cluttered training images is very challenging when the location of object instances is unknown. Previous works generally require objects covering a large portion of the images. We present a novel approach that can cope with extensive clutter as well as large scale and appearance variations between object instances. To make this possible we propose a conditional random field that starts from generic knowledge and then progressively adapts to the new class. Our approach simultaneously localizes object instances while learning an appearance model specific for the class. We demonstrate this on the challenging Pascal VOC 2007 dataset. Furthermore, our method enables to train any state-of-the-art object detector in a weakly supervised fashion, although it would normally require object location annotations.

1 Introduction

In weakly supervised learning (WSL) we are given a set of images, each containing one or more instances of an unknown object class. In contrast to the fully supervised scenario, the location of objects is *not* given. The task is to learn a model for this object class, which can then be used to determine whether a test image contains the class and possibly even to localize it (typically up to a bounding-box). In this case, the learned model is asked to do more than what the training examples teach.

WSL has become a major topic in recent years [1,2,3,4,5,6,7] to reduce the manual labeling effort to learn object classes. In the traditional paradigm, each new class is learned from scratch without any knowledge other than what was engineered into the system. In this paper, we explore a scenario where generic knowledge about object classes is first learned during a *meta-training stage* when images of many different classes are provided along with the location of objects. This generic knowledge is then used to support the learning of a new class *without* location annotation (fig. 1). Generic knowledge makes WSL easier as it rests on a stronger basis.

We propose a conditional random field (CRF) to simultaneously localize object instances and learn an appearance model for the new class. The CRF aims at selecting one window per image containing an instance of the new object class. We alternate between localizing the objects in the training images and learning class-specific models that are then incorporated into the next iteration. Initially

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 452–466, 2010.

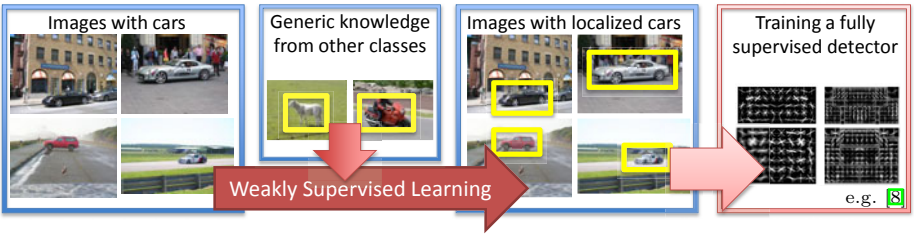


Fig. 1. Learning scenario. Starting from weakly supervised images we localize the object instances of a new class while learning an appearance model. Generic knowledge is used to start WSL on a stronger basis. Our method can be used as a pre-processing step to any fully supervised object detector.

the CRF employs generic knowledge to guide the selection process as it *reduces the location ambiguity*. Over the iterations the CRF progressively adapts to the new class, learning more and more about its appearance and shape. This strategy enables our method to learn from very cluttered images containing objects with large variations in appearance and scale, such as the PASCAL VOC 2007 [9] (fig. 4,5).To the best of our knowledge, no earlier method has been demonstrated capable of learning from PASCAL07 in a WSL scenario (but on easier datasets such as Caltech4 [3] or Weizmann horses [10]).

The main contribution of this paper is a novel method to jointly localize and learn a new class from WS data. Therefore, in sec. 7 we directly evaluate performance as the percentage of instances of the new class which it localizes in their WS training images, and compare to two existing methods [11,12] and various baselines. Moreover, we also demonstrate an application of our method: we train the fully supervised model of Felzenszwalb et al. [8] from objects localized by our method, evaluate it on the PASCAL07 test set, and compare its performance to the original model trained from ground-truth bounding-boxes.

Related Work. We focus here on WSL methods to learn object classes (i.e. requiring no object locations). Many approaches are based on a *bag-of-words* for the entire image [13,14]. Although they have demonstrated impressive classification performance [9], they are usually unable to localize objects.

There are several WSL methods that achieve localization, such as part-based [2,3], segmentation-based [1,4,11,5,6,15], and others [12,16,7]. However, most methods have been demonstrated on datasets such as Caltech4 [2,3,1,4,6,16,7] and Weizmann horses [10,6,15], where objects are rather centered and occupy a large portion of the image, there is little scale/viewpoint variation, and limited background clutter. This is due to the difficulty of spotting the recurring object pattern in challenging imaging conditions.

There are a few exceptions [11,12,17]. [11] attempts to segment out regions similar across many images from the difficult LabelMe dataset [18], but reports that it is very hard to find small objects such as cars in it. [12] is related to our approach as it also finds one window per image. It iteratively refines windows initialized from the most discriminative local features. This fails when the objects

occupy a modest portion of the images and for classes such as horses, for which local texture features have little discriminative power. [17] clusters windows of similar appearance using link analysis techniques. Both [12] and [17] experiment on (part of) the PASCAL VOC 06 dataset. We quantitatively compare to [11,12] in sec. 7.

Our use of generic knowledge is related to *transfer learning* [19,20], where learning a new class is helped by labeled examples of other classes. There are relatively few works on transfer learning for visual recognition. Lando and Edelman [21] learn a new face from just one view, supported by images of other faces. Fei-Fei [22] sequentially updates a part-based classifier trained on previous object classes to fit a new class from very few examples. Stark et al. [23] transfer shape knowledge between related classes in a manually controlled manner. Tommasi et al. [24] use the parameters of the SVM for a known class as a prior for a new, related class. These works reduce the number of images necessary to learn a new class, improving generalization from only a few examples [20]. In this paper instead, we reduce the *degree of supervision* (i.e. no object locations). As another difference, the works above transfer knowledge from one class to another, whereas our generic knowledge provides a background against which it is easier to learn *any* new class. Our generic knowledge conveys how to localize new classes. Automatically *localizing* instances of the new class in their training images is a central objective of our work.

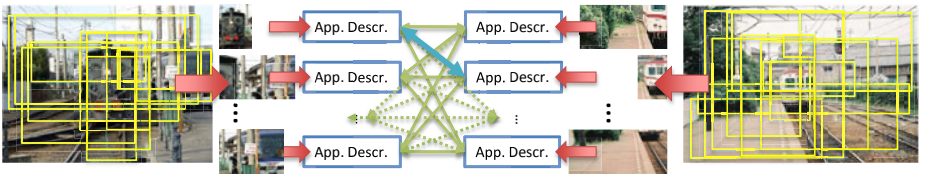
Plan of the Paper. Our new CRF model is described in sec. 2. In sec. 3 and 4 we explain how it is used to localize instances of a new object class in WS training images while learning a model of the new class. Sec. 5 details the generic knowledge that is incorporated into the process and how it is obtained. Sec. 6 describes the image cues we use and in sec. 7 we experimentally evaluate the method.

2 The CRF Model to Localize a New Class

The goal of this paper is to simultaneously localize objects of a new target class in a set of training images and learn an appearance model of the class. As we make no assumption about object locations, scales, or overall shape (aspect-ratio), any image window can potentially contain an object of the target class. We select one window per image by optimizing an energy function defined globally over all training images. Ideally the energy is minimal when all selected windows contain an object of the same class.

Configuration of Windows L. The set of training images $\mathcal{I} = (I_1, \ldots, I_N)$ is represented as a fully connected CRF. Each image I_n is a node which can take on a state from a discrete set corresponding to all image windows. The posterior probability for a configuration of windows $L = (l_1, \ldots, l_N)$ can be written as

$$p(L|\mathcal{I}, \Theta) \propto \exp\left(\sum_n \rho_n \Phi(l_n|I_n, \Theta) + \sum_{n,m} \rho_n \rho_m \Psi(l_n, l_m|I_n, I_m, \Theta)\right) \quad (1)$$



$$\Psi(l_n, l_m | I_n, I_m, \Theta) = \alpha_\Lambda \Lambda(l_n, l_m, \theta_\Lambda) + \sum_f \alpha_{\Gamma_f} \Gamma_f(l_n, l_m | I_n, I_m, \theta_{\Gamma_f})$$

Fig. 2. The pairwise potential. Two images with candidate windows (yellow). Appearance descriptors are extracted for each window (arrows). The pairwise potential Ψ is computed for every pair of windows between the two images, as a linear combination of appearance similarity cues Γ_f and the aspect-ratio similarity Λ.

where each l_n is a window in image I_n; Θ are the parameters of the CRF; ρ_n is the responsibility of image I_n, weighting its impact on the overall energy (sec. 4.3).

The Unary Potential Φ measures how likely an image window l_n is to contain an object of the target class

$$\Phi(l_n; I_n) = \alpha_\Omega \Omega(l_n | I_n, \theta_\Omega) + \alpha_\Pi \Pi(l_n | I_n, \theta_\Pi) + \sum_f \alpha_{\Upsilon_f} \Upsilon_f(l_n | I_n, \theta_{\Upsilon_f}) \qquad (2)$$

It is a linear combination of: (a) Ω, the likelihood [25] that l_n contains an object of *any* class, rather than background (sec. 5.1); (b) Π, a model of the overall shape of the windows, specific to the target class (sec. 4.2); (c) Υ_f, appearance models, one for each image cue f, specific to the target class (sec. 4.1). The scalars α weight the terms.

The Pairwise Potential Ψ measures the similarity between two windows, assessing how likely they are to contain objects of the same class (fig. 2).

$$\Psi(l_n, l_m | I_n, I_m, \Theta) = \alpha_\Lambda \Lambda(l_n, l_m, \theta_\Lambda) + \sum_f \alpha_{\Gamma_f} \Gamma_f(l_n, l_m | I_n, I_m) \qquad (3)$$

It is a linear combination of: (a) Λ, a prior on the shape similarity between two windows l_n, l_m, depending only on states l_n, l_m (sec. 5.2); (b) a term Γ_f measuring the appearance similarity between l_n and l_m according to multiple cues f that depends on the image content (sec. 5.3). The scalars α weight the terms. Fig. 2 illustrates the computation of the pairwise potential for every pair of windows between two images.

The Parameters $\boldsymbol{\theta_\Omega}, \boldsymbol{\theta_\Lambda}, \boldsymbol{\theta_{\Gamma_f}}$ and the weights α are learned from meta-training data (sec. 5). The class-specific models Π and Υ and the image responsibilities ρ_n are initially unknown and set to uniform. Over the learning iterations they are progressively adapted to the target class (sec. 4).

Note that our model connects nodes (windows) *between* images, rather than elements *within* an image as typically done for CRFs in other domains (e.g. pixels in segmentation [26], body parts in human pose estimation [27]).

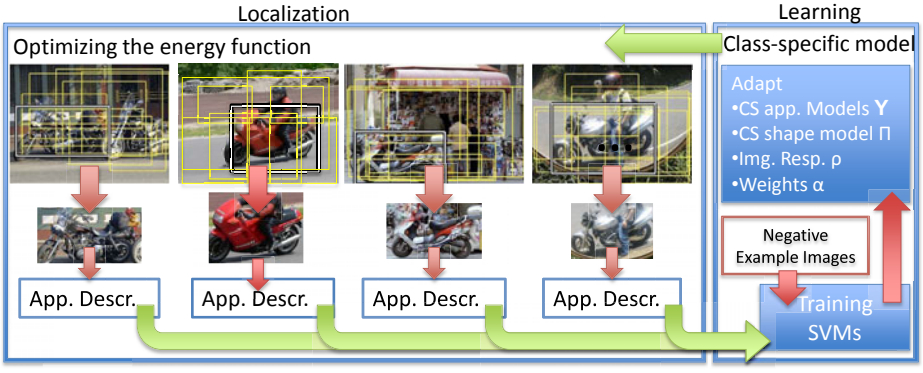


Fig. 3. Localization and learning. The localization and learning stages are alternated. *Localization:* one window (black/white) is selected among the candidate windows (yellow) for each image. *Learning:* a model Υ specific to the target class is (re)-trained from the appearance descriptors of the selected windows and a set of negative training windows. Other CRF components are adapted to the class and the CRF is updated.

3 Localization and Learning

When given a set of images $\mathcal{I}$ of a target class the goal is to localize its object instances. The *localization* and *learning* stages are alternated, optimizing one while keeping the other fixed (fig. 3).

3.1 Localization

Localizing objects corresponds to finding the configuration L^* that maximizes eq. (1):

$$L^* = \arg\max_L \{p(L|\mathcal{I}, \Theta)\} \tag{4}$$

The selected windows L^* are the most likely to contain instances of the same object class (according to our model).

Optimizing our fully connected model is NP-hard. We approximate the global optimum using the tree-reweighted message passing algorithm TRW-S [28]. TRW-S also returns a lower bound on the energy. When this coincides with the returned solution, we know it found the global optimum. In our experiments, TRW-S finds it in 93% of the cases, and in the others the lower bound is only 0.06% smaller on average than the returned energy. Thus we know that the obtained configurations L^* are very close to the global optimum.

3.2 Learning

Based on the selected windows L^*, we adapt several characteristics of the CRF to the target class: (a) the class-specific appearance models Υ_f, (b) the class-specific shape model Π, (c) the image responsibilities ρ_n, and (d) the weights α of the cues (details in sec. 4).

The localization and learning stages *help each other*, as better localizations lead to better class-specific models, which in turn sharpen localization. Similar EM-like optimization schemes [8] are commonly used to learn in the presence of latent variables (in our case L^*).

4 Adaptation

During the learning stage (sec. 3.2), the CRF is progressively adapted from generic to class-specific. For this adaptation, an additional negative image set $\mathcal{N}$ is used, which does not contain any object of the target class.

4.1 Class-Specific Appearance Models $\boldsymbol{\Upsilon_f}$

Any model trainable from annotated object windows could be used here (e.g. [29,8,30,14]). We train a separate SVM θ_{Υ_f} for each appearance cue f.

Since usually not all selected windows L^* contain an object of the target class, these SVMs are iteratively trained. First, the SVM θ_f is trained to separate all windows L^* from windows randomly sampled from $\mathcal{N}$. Then, the SVM θ_f is used to score each training window $l_n^* \in L^*$. The top scored $\kappa\%$ windows are then used to retrain θ_f. This is repeated ten times.

4.2 Class-Specific Shape Model Π

The parameters θ_Π are learned as the distribution of the aspect-ratio of the selected windows L^*.

4.3 Image Responsibilities $\boldsymbol{\rho_n}$

emphasize images where the model is confident of having localized an object of the target class. We set ρ_n proportional to the score of the class-specific appearance model: $\rho_n \propto \sum_f \alpha_{\Upsilon_f} \Upsilon_f(l_n|I_n, \theta_\Upsilon)$. This reduces the impact of particularly difficult images and makes the model more robust to outliers.

4.4 Unary Appearance Cue Weights $\boldsymbol{\alpha_{\Upsilon_f}}$

Not all classes can be discriminated equally well using the same cues (e.g. motorbikes can be recognized well using texture patches, mugs using shape/gradient features, and sheep using color). We adapt the weights α_{Υ_f} of the class-specific appearance models Υ_f for the cues f. We use the top-scored $\kappa\%$ selected windows to train a linear SVM w to combine their appearance scores $\Upsilon_f(l_n|I_n, \theta_{\Upsilon_f})$. Then, we update $\alpha_{\Upsilon_f} \leftarrow \alpha_{\Upsilon_f} + \lambda w_f$. The scalar λ controls the adaptation rate.

4.5 Pairwise Appearance Cue Weights $\boldsymbol{\alpha_{\Gamma_f}}$

We proceed analogously to the previous paragraph. The SVM w is trained to combine the scores $\Gamma_f(l_n, l_m|I_n, I_m)$ between all pairs of the top $\kappa\%$ selected windows.

The objectness Ω, the shape similarity Λ, and the appearance similarity Γ_f are not explicitly adapted to the target class but only implicitly through weights $\alpha_{\Upsilon_f}, \alpha_{\Gamma_f}$ and image responsibilities ρ_n.

5 Generic Knowledge: Initializing Θ

Initially the model parameters Θ carry only generic knowledge. They are learned in a meta-training stage to maximize the localization performance on a set of meta-training images $\mathcal{M}$ containing objects of known classes annotated with bounding-boxes.

5.1 Objectness Ω

We use the objectness measure $\Omega(l|I, \theta_\Omega)$ of [25], which quantifies how likely it is for a window l to contain an object of *any* class. Objectness is trained to distinguish windows containing an object with a well-defined boundary and center, such as cows and telephones, from amorphous background windows, such as grass and road. Objectness combines several image cues measuring distinctive characteristics of objects, such as appearing different from their surroundings, having a closed boundary, and sometimes being unique within the image.

We use objectness as a location prior in our CRF, by evaluating it for all windows in an image I and then sampling 100 windows according to their scores $\Omega(l|I, \theta_\Omega)$. These form the set of states for node I (i.e. the candidate windows the CRF can choose from).

This procedure brings two advantages. First, it greatly reduces the computational complexity of CRF inference, which grows with the square of the number of states (there are $\simeq 10^8$ windows in an image [30]). Second, the sampled windows and their scores Ω attract the CRF toward selecting objects rather than background windows. In a WSL setup this avoids trivial solutions, e.g. where all selected windows cover a chunk of sky in airplane training images [7]. In sec. 7 we evaluate objectness quantitatively. For more details about objectness see [25].

5.2 Pairwise Shape Similarity Λ

θ_Λ is learned as the Bayesian posterior $\Lambda(l_n, l_m, \theta_\Lambda) = p(l_n \overset{c}{=} l_m|\mathrm{SS}(l_n, l_m))$ from many window pairs containing the same ($l_n \overset{c}{=} l_m$) and different classes. $\mathrm{SS}(l_n, l_m)$ measures the aspect ratio similarity of the windows l_n and l_m. In practice this learns that instances of the same class have similar aspect-ratios.

5.3 Pairwise Appearance Similarity Γ_f

We compute the similarity between two windows l_n, l_m in images I_n, I_m as the SSD $||l_n^f(I_n) - l_m^f(I_m)||^2$ between their appearance descriptors $l_n^f(I_n)$ and $l_m^f(I_m)$. This measures how likely they are to contain instances of the same class, according to cue f. The pairwise potentials Γ_f are directly defined as $\Gamma_f(l_n, l_m|I_n, I_m) = ||l_n^f(I_n) - l_m^f(I_m)||^2$.

5.4 Weights α

To learn the weights α between the various terms of our model, we perform a multi-stage grid search.

First, we learn the weights α_Ω, α_Λ, and α_{Γ_f} for objectness Ω, shape similarity Λ, and appearance similarity Γ_f so that the windows L^* returned by the localization stage (sec. 3.1) best cover the meta-training bounding-boxes $\mathcal{M}$ (according to the criterion in sec. 7.1). These weights are determined using only the localization stage, not the adaptation stage, as they contain no class-specific knowledge.

With these weights fixed, we proceed to determine the remaining weights α_Π and α_{Υ_f} for the class-specific shape model Π and the class-specific appearance models Υ_f. These are learned using the full method (sec. 3.1 and 3.2).

5.5 Kernel of the SVMs Υ_f

We evaluated linear and intersection kernels for the class-specific appearance models Υ_f and found the latter to perform slightly better.

5.6 Percentage κ of Images

With weights α and the SVM kernels fixed, we determine the optimal percentage κ of selected windows to use for the iterative training in sec. 4.1.

The remaining parameters, the class-specific appearance models Υ_f, the class-specific shape model Π, and the image responsibilities ρ_n are not learned from meta-training data. They are initially unknown and set uniformly.

6 Appearance Cues

We extract 4 appearance descriptors f from each candidate window and use them to calculate the appearance similarity Γ_f and the class-specific appearance score Υ_f.

GIST [31] is based on localized histograms of gradient orientations. It captures the rough spatial arrangement of image structures, and has been shown to work well for describing the overall appearance of a scene. Here instead, we extract GIST from each candidate *window*.

Color Histograms (CH) provide complementary information to gradients. We describe a window with a single histogram in the LAB color space.

Bag of Visual Words (BOW) are de-facto standard for many object recognition tasks [30,13,12,14]. We use the SURF descriptors [32,30] and quantize them into 2000 words using k-means. A window is described by a BOW of SURF.

Histograms of Oriented Gradients (HOG) also are an established descriptor for object class recognition [8,29].

7 Experiments: WS Localization and Learning

We evaluate the central ability of our method: localizing objects in weakly supervised training images. We experiment on datasets of varying difficulty.

Caltech4 [3]. We use 100 random images for each of the four classes in this popular dataset (airplanes, cars, faces, motorbikes). The images contain large, centered objects, and there is limited scale variation and background clutter.

As meta-training data $\mathcal{M}$ we use 444 train+val images from 6 PASCAL07 classes (bicycle, bird, boat, bus, dog, sheep) with bounding-box annotations. $\mathcal{M}$ is used to learn the parameters for initializing our CRF (sec. 5). This is done only once. The same parameters are then reused in all experiments.

Pascal06 [33,12]. For comparison, we run our method on the training subsets used by [12][1]. These include images for each aspect of 6 classes: car, bicycle, bus, motorbike, cow, and sheep. Up to four aspects are considered per class, totaling 14 training sets (see [12] for details). Although PASCAL VOC06 images are challenging in general, these subsets are easier and contain many large objects. As meta-training data $\mathcal{M}$ we use 471 train+val images from 6 PASCAL07 classes (aeroplane, bird, boat, cat, dog, horse).

Pascal07-6x2 [9]. For the detailed evaluation of the components of our method below, we use all images from 6 classes (aeroplane, bicycle, boat, bus, horse, and motorbike) of the PASCAL VOC 2007 train+val dataset from the left and right aspect each. Each of the 12 class/aspect combination contains between 28 and 67 images for a total of 538 images. As negative set $\mathcal{N}$ we use 2000 random images taken from train+val not containing any instance of the target class. This dataset is very challenging, as objects vary greatly in location, scale, and appearance. Moreover, there is significant viewpoint variation within an aspect (fig. 4, 5). We report in detail on these classes because they represent compact objects on which fully supervised methods perform reasonably well [9] (as opposed to classes such as 'potted plant' where even fully supervised methods fail). As meta-training data $\mathcal{M}$ we use 799 train+val images from 6 other PASCAL07 classes (bird, car, cat, cow, dog, sheep).

Pascal07-all [9]. For completeness, we also report results for *all* class/aspect combinations in PASCAL07 with more than 25 images (our method, as well as the competitors and baselines to which we compare, fails when given fewer images). We use the same meta-training data as for PASCAL07-6x2. In total, the PASCAL07-all set contains 42 class/aspect combinations, covering all 14 classes not used for meta-training.

[1] Provided to us by the authors of [12].

7.1 Localizing Objects in Their Weakly Supervised Training Images

We *directly* evaluate the ability of our method to localize objects in a set of training images $\mathcal{I}$ only known to contain a target class (sec. 7). Tab. 1 shows results for two baselines, two competing methods [11,12] and for several variants of our method. We report as CorLoc the percentage of images in which a method correctly localizes an object of the target class according to the PASCAL-criterion (window intersection-over-union > 0.5). No location of any object in $\mathcal{I}$ is given to any method beforehand. The detailed analysis in the following paragraphs focuses on the Caltech4, PASCAL06, and PASCAL07-6x2 datasets. The last paragraph discusses results on the PASCAL07-all dataset.

Baselines. The 'image center' baseline simply picks a window in the image center by chopping 10% off the width/height from the image borders. This is useful to assess the difficulty of a dataset. The 'ESS' baseline is based on bag-of-visual-words. We extract SURF features [32] from all images of a dataset, cluster them into 2000 words, and weight each word by the log of the relative frequency of occurrence in positive vs negative images of a class (as done by [13,12,30]). Hence, these feature weights are class-specific. For localization, we use Efficient Subwindow Search (ESS) [30] to find the window with the highest sum of weights in an image[2].

The image center baseline confirms our impressions about the difficulty of the datasets. It reaches about 70% CorLoc on Caltech4 and PASCAL06-[12], but fails on PASCAL07. The trend is confirmed by ESS.

Competitors. We compare to the method from [11] using their implementation[3]. This method does not directly return one window per image. It determines 30 topics roughly corresponding to object classes. A topic consists of a group of superpixels in each training image. For each topic, we put a bounding-box around its superpixels in every image, and then evaluate its CorLoc performance. We report the performance of the topic with the highest CorLoc. This method achieves a modest CorLoc on the challenging PASCAL07-6x2, but found the object in about half the images of the easier Caltech4 and PASCAL06-[12].

As a second competitor we reimplemented the method from [12], which directly returns one window per image. It works quite well on Caltech4 and on their PASCAL06-[12] subset, where the objects occupy a large portion of the images. On the much harder PASCAL07-6x2 it performs considerably worse since its initialization stage does not lock onto objects[4]. Overall, this method performed better than [11] on all three datasets.

[2] Baseline suggested by C. Lampert in personal communications.

[3] http://www.di.ens.fr/~russell/projects/mult_seg_discovery/index.html

[4] Unfortunately, we could not obtain the source code from the authors of [12]. We asked them to process our PASCAL07-6x2 training sets and they confirmed that their method performs poorly on them.

Table 1. Results. The first block reports results for the baselines and the second for the competitors [12,11]. Rows (a)-(c): results for our method using only the localization stage. Rows (d)-(e): results for the full method using the localization and learning stages. All results until row (e) are given in CorLoc. Rows (f)-(g) report the performance of the objectness measure Ω (see main text). Column (Color) shows the colors used for visualization in figs. 4, 5.

			Pascal07		
Method	Caltech4	Pascal06-[12]	6×2	all	Color
image center	66	76	23	14	
ESS	43	33	23	10	
Russel et al. [11]	40	58	20	13	
Chum et al. [12]	57	67	29	15	
this paper – localization only					
(a) random windows	0	0	0	0	
(b) single cue (GIST)	72	64	35	21	
(c) all cues	70	77	39	22	
this paper – localization and learning					
(d) learning Υ_f	85	84	48	22	
(e) **full adaptation**	**87**	**82**	**50**	**26**	
objectness measure Ω					
(f) hit-rate	100	99	89	85	
(g) signal-to-noise	29	31	16	14	

Localization Only (a)-(c). Here we stop our method after the localization stage (sec. 3.1), without running the learning stage (sec. 3.2). In order to investigate the impact of generic knowledge, we perform experiments with several stripped-down versions of our CRF model. Models (a) and (b) use only GIST descriptors in the pairwise similarity score Γ_f. (a) uses 100 random candidate windows with uniform scores in Ω; (b) uses 100 candidate windows sampled from the objectness measure Ω (sec. 5.1). While method (a) is not able to localize any object, (b) already performs quite well.

By adding the remaining appearance cues Γ_f in setup (c), results improve further and all baselines and competitors are outperformed. Using only the localization stage, our method already localizes more than 70% of the objects in Caltech4 and Pascal06-[12], and 39% of the objects in Pascal07-6x2.

Localization and Learning (d)-(e). Here we run our full method, iteratively alternating localization and learning. In setup (d), we learn only appearance models Υ_f specific to the target class. In setup (e), all parameters of the CRF are adapted to the target class. The considerable increase in CorLoc shows that the learning stage helps localization. The full method (e) substantially outperforms all competitors/baselines on all datasets, and in particular reaches about twice their CorLoc on Pascal07-6x2. Overall, it finds most objects in Caltech4 and Pascal06-[12], and half of those in Pascal07-6x2 (fig. 4, 5).

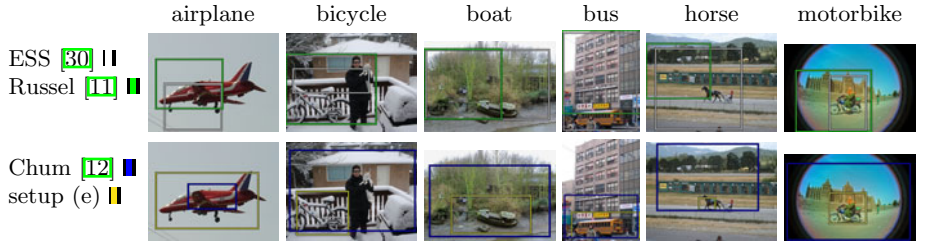


Fig. 4. Comparison to baselines and competitors. Example objects localized by different methods in their weakly supervised training images (i.e. only object presence is given for training, no object locations). Top row: the ESS baseline [30] and the method from [11]. Bottom row: the method from [12] and our method in setup (e). Our method localizes object visibly better than both baselines and competitors, especially in difficult images.

As tab. 1 shows, each variant improves over the previous one. Showing that (*i*) the generic knowledge elements we incorporate are important for a successful initial localization (setups (a)-(c)) and (*ii*) the learning stage successfully adapts the model to the target class (setups (d)-(e)).

Pascal07-all. For completeness, we report in tab. 1 also results over the PASCAL07-all set, which contains 42 class/aspect combinations, including many for which even fully supervised methods fail (e.g. 'potted plant'). Compared to PASCAL07-6x2, CorLoc drops by about half for all methods, suggesting that WS learning on *all* PASCAL07 classes is beyond what currently possible. However, it is interesting to notice how the relative performance of our method (setup (e)) compared to the competitors [11,12] is close to what observed in PASCAL07-6x2: our method performs about twice as well as them.

Objectness Measure (f)-(g). We also evaluate the 100 windows sampled from Ω. The percentage (f) of objects of the target class covered by a sampled window gives an upper-bound on the CorLoc that can be achieved by our method. As the table shows, most target objects are covered. The percentage (g) of sampled windows covering an object of the target class gives the signal-to-noise ratio that enters the CRF model. This ratio is much higher than when considering all image windows.

8 Experiments: Localizing Objects in New Test Images

Our method enables training a fully-supervised object detector from weakly supervised data, although it would normally require object location annotations. To demonstrate this point, we train the fully supervised object detector of [8][5] from objects localized using setup (e) and compare its performance to the original model trained from ground-truth bounding-boxes.

[5] The source code is available at `http://people.cs.uchicago.edu/~pff/latent/`

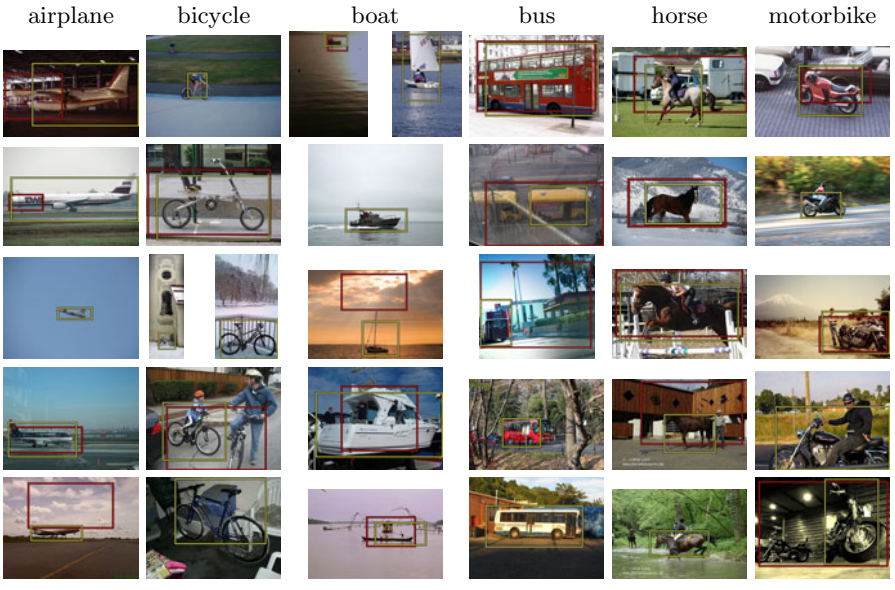


Fig. 5. Example results comparing our method in setup (c) to setup (e). If only is visible, both setups chose the same window. The learning stage in setup (e) leads to more correctly localized objects.

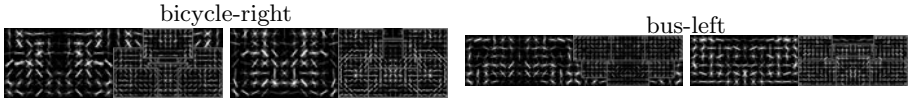


Fig. 6. Models [8] trained from the output of our method (left) and from ground-truth bounding-boxes (right)

We perform this experiment for all 12 class/aspect combinations in PASCAL07-6x2. The detection performance for each class/aspect is measured by the average precision (AP) on the entire PASCAL 2007 test set (4952 images). We report below the mean AP over the 12 class/aspect combinations (mAP). As usual in a test stage, no information is given about the test images, not even whether they contain the object. Notice how this test set is entirely disjoint from the train+val set used for training and meta-training.

The mAP resulting from models trained in a weakly supervised setting from the output of our method is 0.16, compared to 0.33 of the original fully supervised models. Therefore, our method enables to train [8] *without* ground-truth bounding-box annotations, while keeping the detection performance on the test set at about 48% of the model trained from ground-truth bounding-boxes. We consider this a very encouraging result, given that we are not aware of previous methods demonstrated capable of localizing objects on the PASCAL07 test set when trained in a weakly supervised setting. Fig. 6 visually compares two

models trained from the output of our method to the corresponding models trained from ground-truth bounding-boxes.

9 Conclusion

We presented a technique for localizing objects of an unknown class and learning an appearance model of the class from weakly supervised training images. The proposed model starts from generic knowledge and progressively adapts more and more to the new class. This allows to learn from highly cluttered images with strong scale and appearance variations between object instances. We also demonstrated how to use our method to train a fully supervised object detector from weakly supervised data.

References

1. Arora, H., Loeff, N., Forsyth, D., Ahuja, N.: Unsupervised segmentation of objects using efficient learning. In: CVPR (2007)
2. Crandall, D.J., 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)
3. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR (2003)
4. Galleguillos, C., Babenko, B., Rabinovich, A., Belongie, S.: Weakly supervised object localization with stable segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 193–207. Springer, Heidelberg (2008)
5. Todorovic, S., Ahuja, N.: Extracting subimages of an unknown category from a set of images. In: CVPR (2006)
6. Winn, J., Jojic, N.: LOCUS: learning object classes with unsupervised segmentation. In: ICCV (2005)
7. Nguyen, M., Torresani, L., de la Torre, F., Rother, C.: Weakly supervised discriminative localization and classification: a joint learning process. In: ICCV (2009)
8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI (2009) (in press)
9. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 Results (2007)
10. Borenstein, E., Ullman, S.: Learning to segment. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 315–328. Springer, Heidelberg (2004)
11. Russell, B., Efros, A., Sivic, J., Freeman, W., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
12. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: CVPR (2007)
13. Dorkó, G., Schmid, C.: Object class recognition using discriminative local features. Technical Report RR-5497, INRIA - Rhone-Alpes (2005)
14. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. In: IJCV (2007)

15. Cao, L., Li, F.F.: Spatially coherent latent topic model for concurrent segmentation and classification of objects and scene. In: ICCV (2007)
16. Lee, Y.J., Grauman, K.: Shape discovery from unlabeled image collections. In: CVPR (2009)
17. Kim, G., Torralba, A.: Unsupervised detection of regions of interest using iterative link analysis. In: NIPS (2009)
18. Russel, B.C., Torralba, A.: LabelMe: a database and web-based tool for image annotation. IJCV 77, 157–173 (2008)
19. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.: Self-taught learning: transfer learning from unlabeled data. In: ICML (2007)
20. Thrun, S.: Is learning the n-th thing any easier than learning the first? In: NIPS (1996)
21. Lando, M., Edelman, S.: Generalization from a single view in face recognition. In: Technical Report CS-TR 95-02, The Weizmann Institute of Science (1995)
22. 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 Workshop of Generative Model Based Vision (2004)
23. Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. In: ICCV (2009)
24. Tommasi, T., Caputo, B.: The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories. In: BMVC (2009)
25. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: CVPR (2010)
26. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: interactive foreground extraction using iterated graph cuts. In: SIGGRAPH, vol. 23, pp. 309–314 (2004)
27. Ramanan, D.: Learning to parse images of articulated bodies. In: NIPS (2006)
28. Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. PAMI 28, 1568–1583 (2006)
29. Dalal, N., Triggs, B.: Histogram of Oriented Gradients for Human Detection. In: CVPR (2005)
30. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. PAMI (2009) (in press)
31. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
32. Bay, H., Ess, A., Tuytelaars, T., van Gool, L.: SURF: Speeded up robust features. CVIU 110, 346–359 (2008)
33. Everingham, M., Van Gool, L., Williams, C.K.I., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2006 (VOC2006) (2006)

Monocular 3D Scene Modeling and Inference: Understanding Multi-Object Traffic Scenes

Christian Wojek[1,2], Stefan Roth[1], Konrad Schindler[1,3], and Bernt Schiele[1,2]

[1] Computer Science Department, TU Darmstadt
[2] MPI Informatics, Saarbrücken
[3] Photogrammetry and Remote Sensing Group, ETH Zürich

Abstract. Scene understanding has (again) become a focus of computer vision research, leveraging advances in detection, context modeling, and tracking. In this paper, we present a novel probabilistic 3D scene model that encompasses multi-class object detection, object tracking, scene labeling, and 3D geometric relations. This integrated 3D model is able to represent complex interactions like inter-object occlusion, physical exclusion between objects, and geometric context. Inference allows to recover 3D scene context and perform 3D multiobject tracking from a mobile observer, for objects of multiple categories, using only monocular video as input. In particular, we show that a joint scene tracklet model for the evidence collected over multiple frames substantially improves performance. The approach is evaluated for two different types of challenging onboard sequences. We first show a substantial improvement to the state-of-the-art in 3D multi-people tracking. Moreover, a similar performance gain is achieved for multi-class 3D tracking of cars and trucks on a new, challenging dataset.

1 Introduction

Robustly tracking objects from a moving observer is an active research area due to its importance for driver assistance, traffic safety, and autonomous navigation [1,2]. Dynamically changing backgrounds, varying lighting conditions, and the low viewpoint of vehicle-mounted cameras all contribute to the difficulty of the problem. Furthermore, to support navigation, object locations should be estimated in a global 3D coordinate frame rather than in image coordinates.

The main goal of this paper is to address this important and challenging problem by proposing a new *probabilistic 3D scene model*. Our model builds upon several important lessons from previous research: (1) robust tracking performance is currently best achieved with a *tracking-by-detection* framework [3]; (2) short term evidence aggregation, typically termed *tracklets* [4], allows for increased tracking robustness; (3) the objects should not be modeled in isolation, but in their *3D scene context*, which puts strong constraints on the position and motion of tracked objects [1,5]; and (4) *multi-cue combination* of scene labels and object detectors allows to strengthen weak detections, but also to prune inconsistent false detections [5]. While all these different components have been shown to boost performance individually, in the present work we for the first time integrate them all in a single system. As our experiments show, the proposed probabilistic 3D scene model significantly outperforms the current state-of-the-art. Fig. 1

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 467–481, 2010.

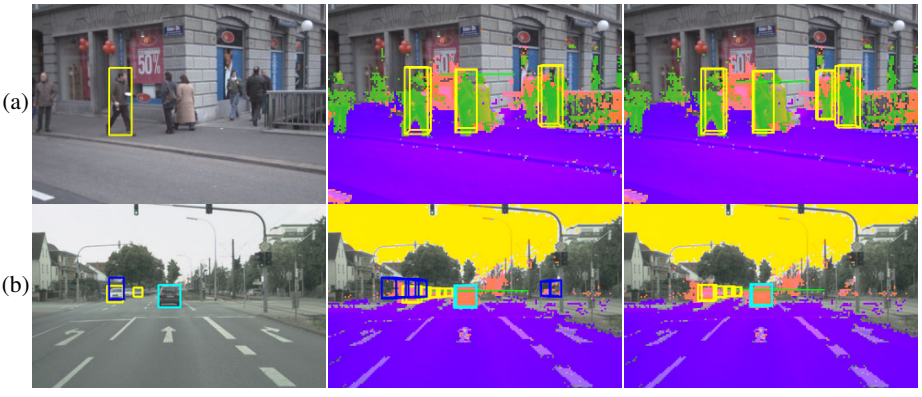


Fig. 1. Our system performs 3D inference to reinforce weakly detected objects and to prune false positive detections by exploiting evidence from scene labeling and an object detector. *(left)* Detector input; *(middle)* single-frame 3D inference with overlaid scene labeling and horizon estimate; *(right)* multi-frame tracking results (all results at 0.1 FPPI). See Sec. 6 for a detailed discussion.

shows example results for two different types of challenging onboard sequences. Our system is able to robustly track a varying number of targets in 3D world coordinates in highly dynamic scenes. This enables us to use a single camera only instead of relying on stereo cameras as in previous work (e.g., [1,2]).

Despite using only monocular input, the proposed model allows to constrain object detections to geometrically feasible locations and enforces physically plausible 3D dynamics. This improves object detection results by pruning physically implausible false positives and strengthening weak detections along an object's trajectory. We demonstrate that accumulating scene evidence over a small number of frames with help of a 3D scene model significantly improves performance. As exact inference is intractable we employ reversible-jump Markov Chain Monte Carlo (RJMCMC) sampling to approximate per-frame distributions. Further improvement can be achieved by performing long-term data association with a Hidden Markov Model (HMM).

2 Related Work

Our work builds on recent advances in scene understanding by pixel-wise labeling, 3D scene analysis and tracking. The use of scene context has been investigated in the computer vision literature in several ways. Torralba [6] proposes to employ Gabor filter-bank responses in a bottom-up fashion in order to gain prior information on likely 2D object positions. More recently, Shotton et al. [7] use a strong joint-boosting classifier with context reasoning based on a CRF framework to provide a local, per pixel classification of image content. Ess et al. [8] and Brostow et al. [9] particularly address traffic scene understanding. [8] uses 2D Walsh-Hadamard filter-bank responses together with stereo depth information to infer traffic situations, while [9] leverages 3D point clouds to improve 2D scene segmentation. Tu et al. [10] use MCMC sampling techniques to combine top-down discriminative classifiers with bottom-up generative models for 2D

image understanding. Common to these approaches is the goal of 2D image understanding. Our work includes scene labeling as a cue, but its ultimate goal is to obtain a 3D model of the observed world.

This paper is most similar to work by Hoiem et al. [5] and Ess et al. [1]. [5] combines image segmentation and object detections in order to infer the objects' positions in 3D. Their work, however, is limited to single images and does not exploit temporal information available in video. [1] extends [5], but requires a stereo camera setup to achieve robust tracking of pedestrians from a mobile platform. Similarly, [2] tracks pedestrians for driver assistance applications and employs a stereo camera to find regions of interest and to suppress false detections. Note, however, that stereo will yield only little improvement in the far field, because a stereo rig with a realistic baseline will have negligible disparity. Thus, further constraints are needed, since appearance-based object detection is unreliable at very small scales. Therefore, we investigate the feasibility of a monocular camera setup for mobile scene understanding. Another system that uses monocular sequences is [11]. Contrary to this work, we tightly couple our scene model and the hypothesized positions of objects with the notion of scene tracklets, and exploit constraints given by a-priori information (e.g., approximate object heights and camera pitch). Our experiments show that these short-term associations substantially stabilize 3D inference and improve robustness beyond what has previously been reported. Our experimental results show that the proposed approach outperforms the stereo system by Ess et al. [1].

Tracking-by-detection, with an offline learned appearance model, is a popular approach for tracking objects in challenging environments. Breitenstein et al. [12], for instance, track humans based on a number of different detectors in image coordinates. Similarly, Okuma et al. [3] track hockey players in television broadcasts. Huang et al. [13] track people in a surveillance scenario from a static camera, grouping detections in neighboring frames into *tracklets*. Similar ideas have been exploited by Kaucic et al. [4] to track vehicles from a helicopter, and by Li et al. [14] to track pedestrians with a static surveillance camera. However, none of these tracklet approaches exploit the strong constraints given by the size and position of other objects, and instead build up individual tracks for each object. In this paper we contribute a probabilistic scene model that allows to jointly infer the camera parameters and the position of *all* objects in 3D world coordinates by using only monocular video and odometry information. Increased robustness is achieved by extending the tracklet idea to *entire scenes* toward the inference of a global scene model.

Realistic, but complex models for tracking including ours are often not amenable to closed-form inference. Thus, several approaches resort to MCMC sampling. Khan et

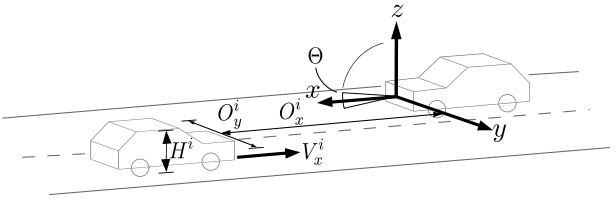


Fig. 2. Visualization of the 3D scene state $\mathbf{X}$ in the world coordinate system. The camera is mounted to the vehicle on the right.

al. [15] track ants and incorporate their social behavior by means of an MRF. Zhao et al. [16] use MCMC sampling to track people from a static camera. Isard&MacCormick [17] track people in front of relatively uncluttered backgrounds from a static indoor camera. All three approaches use rather weak appearance models, which prove sufficient for static cameras. Our model employs a strong object detector and pixel-wise scene labeling to cope with highly dynamic scenes recorded from a moving platform.

3 Single-Frame 3D Scene Model

We begin by describing our 3D scene model for a *single image*, which aims at combining available prior knowledge with image evidence in order to reconstruct the 3D positions of all objects in the scene. For clarity, the time index t is omitted when referring to a single time step only. Variables in image coordinates are printed in lower case, variables in 3D world coordinates in upper case; vectors are printed in bold face.

The posterior distribution for the 3D scene state $\mathbf{X}$ given image evidence $\mathcal{E}$ is defined in the usual way, in terms of a prior and an observation model:

$$P(\mathbf{X}|\mathcal{E}) \propto P(\mathcal{E}|\mathbf{X})P(\mathbf{X}) \tag{1}$$

The 3D state $\mathbf{X}$ consists of the individual states of all objects $\mathbf{O}^i$, described by their relative 3D position $(O_x^i, O_y^i, O_z^i)^\top$ w.r.t. the observer and by their height H^i. Moreover, $\mathbf{X}$ includes the internal camera parameters $\mathbf{K}$ and the camera orientation $\mathbf{R}$.

The goal of this work is to infer the 3D state $\mathbf{X}$ from video data of a monocular, forward facing camera (see Fig. 2). While in general this is an under-constrained problem, in robotic and automotive applications we can make the following assumptions that are expressed in the prior $P(\mathbf{X})$: The camera undergoes no roll and yaw w.r.t. the platform, its intrinsics $\mathbf{K}$ are constant and have been calibrated off-line, and the speed and turn rate of the platform are estimated from odometer readings. Furthermore, the platform as well as all objects of interest are constrained to stand on a common ground plane (i.e., $O_z^i = 0$). Note that under these assumptions the ground plane in camera-centric coordinates is fully determined by the pitch angle Θ. As the camera is rigidly mounted to the vehicle, it can only pitch a few degrees. To avoid degenerate camera configurations, the pitch angle is therefore modeled as normally distributed around the pitch of the resting platform as observed during calibration: $\mathcal{N}(\Theta; \mu_\Theta, \sigma_\Theta)$. This prior allows deviations arising from acceleration and braking of the observer. This is particularly important for the estimation of distant objects as, due to the low camera viewpoint, even minor changes in the pitch may cause a large error for distance estimation in the far field.

Moreover, we assume the height of all scene objects to follow a normal distribution around a known mean value, which is specific for the respective object class c_i, $\mathcal{N}(H^i; \mu_H^{c_i}, \sigma_H^{c_i})$. This helps to prune false detections that are consistent with the ground plane, but are of the wrong height (e.g., background structures such as street lights). The overall prior is thus given as

$$P(\mathbf{X}) \propto \mathcal{N}(\Theta; \mu_\Theta, \sigma_\Theta) \cdot \prod_i \mathcal{N}(H^i; \mu_H^{c_i}, \sigma_H^{c_i}) \tag{2}$$

Next, we turn to the observation model $P(\mathcal{E}|\mathbf{X})$. The image evidence $\mathcal{E}$ is comprised of a set of potential object detections and a scene labeling, i.e., category labels densely

estimated for every pixel. As we will see in the experiments, the combination of these two types of image evidence is beneficial as object detections give reliable but rather coarse bounding boxes, and low level cues enable more fine-grained data association by penalizing inconsistent associations and supporting consistent, but weak detections.

For each object our model fuses object appearance given by the object detector confidence, geometric constraints, and local evidence from bottom-up pixel-wise labeling:

$$P(\mathcal{E}|\mathbf{X}) \propto \prod_i \Psi_D\left(\mathbf{d}^{a(i)}\right) \cdot \Psi_G\left(\mathbf{O}^i, \Theta; \mathbf{d}^{a(i)}\right) \cdot \Psi_L^i\left(\mathbf{X}; \mathbf{l}\right) \tag{3}$$

Here, $a(i)$ denotes the association function, which assigns a candidate object detection $\mathbf{d}^{a(i)}$ to every 3D object hypothesis $\mathbf{O}^i$. Note that the associations between objects and detections are established as part of the MCMC sampling procedure (see Sec. 3.2). The appearance potential Ψ_D maps the appearance score of detection $\mathbf{d}^{a(i)}$ for object i into the positive range. Depending on the employed classifier, we use different mappings – see Sec. 5 for details.

The geometry potential Ψ_G models how well the estimated 3D state $\mathbf{O}^i$ satisfies the geometric constraints due to the ground plane specified by the camera pitch Θ. Denoting the projection of the 3D position $\mathbf{O}^i$ to the image plane as $\mathbf{o}^i$, the distance between $\mathbf{o}^i$ and the associated detection $\mathbf{d}^{a(i)}$ in x-y-scale-space serves as a measure of how much the geometric constraints are violated. We model Ψ_G using a Gaussian

$$\Psi_G(\mathbf{O}^i, \Theta; \mathbf{d}^{a(i)}) = \mathcal{N}(\mathbf{o}^i; \mathbf{d}^{a(i)}, \boldsymbol{\sigma}_G + \bar{\boldsymbol{\sigma}}_G) \;, \tag{4}$$

where we split the kernel bandwidth into a constant component $\boldsymbol{\sigma}_G$ and a scale-dependent component $\bar{\boldsymbol{\sigma}}_G$ to account for inaccuracies that arise from the scanning stride of the sliding-window detectors.

The scene labeling potential Ψ_L^i describes how well the projection $\mathbf{o}^i$ matches the bottom-up pixel labeling. For each pixel j and each class c the labeling yields a classification score $l^j(c)$. Similar to Ψ_D, the labeling scores are normalized pixel-wise by means of a softmax transformation in order to obtain positive values.

It is important to note that this cue demands 3D scene modeling: To determine the set of pixels that belong to each potential object, one needs to account for inter-object occlusions, and hence know the objects' depth ordering. Given that ordering, we proceed as follows: each object is back-projected to a bounding box $\mathbf{o}^i$, and that box is split into a visible region $\boldsymbol{\delta}^i$ and an occluded region $\boldsymbol{\omega}^i$. The object likelihood is then defined as the ratio between the cumulative score for the expected label e and the cumulative score of the pixel-wise best label $k \neq e$, evaluated over the visible part of $\mathbf{o}^i$:

$$\Psi_L^i(\mathbf{X}; \mathbf{l}) = \left(\frac{\sum_{j \in \boldsymbol{\delta}^i} l^j(e) + \tau}{\epsilon|\boldsymbol{\omega}^i| + \sum_{j \in \boldsymbol{\delta}^i} l^j(k) + \tau} \right)^{\alpha} \;, \tag{5}$$

where the constant τ corresponds to a weak Dirichlet prior; $\epsilon|\boldsymbol{\omega}^i|$ avoids highly occluded objects to have a large influence with little available evidence; and α balances the relative importance of detector score and pixel label likelihood.

Importantly, $P(\mathbf{X}|\mathcal{E})$ is not comparable across scene configurations with different numbers of objects. We address this with a reversible jump MCMC framework [18].

3.1 Inference Framework

To perform inference in the above model, we simulate the posterior distribution $P(\mathbf{X}|\mathcal{E})$ in a Metropolis-Hastings MCMC framework [19]. At each iteration s new scene samples $\mathbf{X}'$ are proposed by different *moves* from the proposal density $Q(\mathbf{X}'; \mathbf{X}^{(s)})$. Since our goal is to sample from the equilibrium distribution, we discard the samples from an initial burn-in phase. Note that the normalization of the posterior does not have to be known, since it is independent of $\mathbf{X}$ and therefore cancels out in the posterior ratio.

3.2 Proposal Moves

Proposal moves change the current state of the Markov chain. We employ three different move types: *diffusion moves* to update the last state's variables, *add moves* and *delete moves* to change the state's dimensionality by adding or removing objects from the scene. Add and delete moves are mutually reversible and trans-dimensional. At each iteration, the move type is selected randomly with fixed probabilities q_{Add}, q_{Del} and q_{Dif}.

Diffusion moves change the current state by sampling new values for the state variables. At each diffusion move, object variables are updated with a probability of $q_{\mathbf{O}}$, while Θ is updated with a probability of q_{Θ}.

To update objects we draw the index i of the object to update from a uniform distribution and then update $\mathbf{O}^i$. Proposals are drawn from a multi-variate normal distribution centered at the position of the previous state and with diagonal covariance.

To update the camera pitch Θ proposals are generated from a mixture model. The first mixture component is a broad normal distribution centered at the calibrated pitch for the motionless platform. For the remaining mixture components, we assume distant objects associated with detections at small scales to have the class' mean height and use $\mathbf{d}^{a(i)}$ to compute their distance by means of the theorem of intersecting lines. Then the deviation between the detected bounding box and the object's projection in the image allows one to estimate the camera pitch. We place one mixture component around each pitch computed this way and assign mixture weights proportional to the detection scores to put more weight on more likely objects.

Add moves add a new object $\mathbf{O}^{N+1}$ to the chain's last state, where N is the number of objects contained in $\mathbf{X}^{(s)}$. As this move is trans-dimensional (i.e., the number of dimensions of $\mathbf{X}^{(s)}$ and $\mathbf{X}'$ do not match) special consideration needs to be taken when the posterior ratio $\frac{P(\mathbf{X}'|\mathcal{E})}{P(\mathbf{X}^{(s)}|\mathcal{E})}$ is evaluated. In particular, $P(\mathbf{X}^{(s)}|\mathcal{E})$ needs to be made comparable in the state space of $P(\mathbf{X}'|\mathcal{E})$. To this end, we assume a constant probability $\bar{P}(\mathbf{O}^{N+1})$ for each object to be part of the background. Hence, posteriors of states with different numbers of objects can be compared in the higher dimensional state space by transforming $P(\mathbf{X}^{(s)}|\mathcal{E})$ to

$$\hat{P}(\mathbf{X}^{(s)}|\mathcal{E}) = P(\mathbf{X}^{(s)}|\mathcal{E})\bar{P}(\mathbf{O}^{N+1}) \tag{6}$$

To efficiently explore high density regions of the posterior we use the detection scores in the proposal distribution. A new object index n is drawn from the discrete set of all K detections $\{\bar{d}\}$, which are not yet associated with an object in the scene, according to $Q(\mathbf{X}'; \mathbf{X}^{(s)}) = \frac{\psi_D(\bar{d}^n)}{\sum_k \psi_D(\bar{d}^k)}$. The data association function is updated by letting $a(N+1)$ associate the new object with the selected detection. For distant objects (i.e., detections

at small scales) we instantiate the new object at a distance given through the theorem of intersecting lines and the height prior, whereas for objects in the near-field a more accurate 3D position can be estimated from the ground plane and camera calibration.

Delete moves remove an object $\mathbf{O}^n$ from the last state and move the associated detection $\mathbf{d}^{a(n)}$ back to $\{\bar{d}\}$. Similar to the add move, the proposed lower dimensional state $\mathbf{X}'$ needs to be transformed. The object index n to be removed from the scene is drawn uniformly among all objects currently in the scene, thus $Q(\mathbf{X}'; \mathbf{X}^{(s)}) = \frac{1}{N}$.

3.3 Projective 3D to 2D Marginalization

In order to obtain a score for a 2D position $\mathbf{u}$ (including scale) from our 3D scene model, the probabilistic framework suggests marginalizing over all possible 3D scenes $\mathbf{X}$ that contain an object that projects to that 2D position:

$$P(\mathbf{u}|\mathcal{E}) = \int \max_i \left(\left[\mathbf{u} = \mathbf{o}^i \right] \right) P(\mathbf{X}|\mathcal{E}) \, \mathrm{d}\mathbf{X} \, , \tag{7}$$

with $[expr]$ being the Iverson bracket: $[expr] = 1$ if the enclosed expression is true, and 0 otherwise. Hence, the binary function $\max_i([\cdot])$ detects whether there exists *any* 3D object in the scene that projects to image position $\mathbf{u}$. The marginal is approximated with samples $\mathbf{X}^{(s)}$ drawn using MCMC:

$$P(\mathbf{u}|\mathcal{E}) \approx \frac{1}{S} \sum_{s=1}^{S} \max_i \left(\left[\mathbf{u} = \mathbf{o}^{i,(s)} \right] \right) \, , \tag{8}$$

where $\mathbf{o}^{i,(s)}$ denotes the projection of object $\mathbf{O}^i$ of sample s to the image, and S is the number of samples. In practice $\max_i([\cdot])$ checks whether any of the 3D objects of sample s projects into a small neighborhood of the image position $\mathbf{u}$.

4 Multi-frame Scene Model and Inference

So far we have described our scene model for a single image in static scenes only. For the extension to video streams we pursue a two-stage tracking approach. First, we extend the model to neighboring frames by using greedy data association. Second, the resulting *scene tracklets* are used to extend our model towards long-term data association by performing *scene tracking* with an HMM.

4.1 Multi-frame 3D Scene Tracklet Model

To apply our model to multiple frames, we first use the observer's estimated speed V_{ego} and turn (yaw) rate to roughly compensate the camera's ego-motion. Next, we use a coarse dynamic model for all moving objects to locally perform association, which is refined during tracking. For initial data associations objects that move substantially slower than the camera (e.g., people) are modeled as standing still, $V_x^i = 0$. For objects with a similar speed (e.g., cars and trucks), we distinguish those moving in the same direction as the observers from the oncoming traffic with the help of the detector's class

label. The former are expected to move with a similar speed as the observer, $V_x^i = V_{ego}$, whereas the latter are expected to move with a similar speed, but in opposite direction, $V_x^i = -V_{ego}$. The camera pitch Θ_t can be assumed constant for small time intervals.

For a given frame t we associate objects and detections as described in Sec. 3.2. In adjacent frames we perform association by finding the detection with maximum overlap to each predicted object. Missing evidence is compensated by assuming a minimum detection likelihood anywhere in the image. We define the scene tracklet posterior as

$$P(\mathbf{X}_t|\mathcal{E}_{-\delta t+t:t+\delta t}) \propto \prod_{r=t-\delta t}^{t+\delta t} P(\hat{\mathbf{X}}_r|\mathcal{E}_r), \tag{9}$$

where $\hat{\mathbf{X}}_r$ denotes the predicted scene configuration using the initial dynamic model just explained.

4.2 Long Term Data Association with Scene Tracking

While the above model extension to scene tracklets is feasible for small time intervals, it does not scale well to longer sequences, because greedy data association in combination with a simplistic motion model will eventually fail. Moreover, the greedy formalism cannot handle objects leaving or entering the scene.

We therefore introduce an explicit data association variable $\mathcal{A}_t$, which assigns objects to detections in frame t. With this explicit mapping, long-term tracking is performed by modeling associations over time in a hidden Markov model (HMM). Inference is performed in a sliding window of length w to avoid latency as required by an online setting:

$$\begin{aligned} P(\mathbf{X}_{1:w}, \mathcal{A}_{1:w}|\mathcal{E}_{-\delta t+1:w+\delta t}) = P(\mathbf{X}_1|\mathcal{A}_1, \mathcal{E}_{-\delta t+1:1+\delta t}) \\ \prod_{k=2}^{w} P(\mathcal{A}_k|\mathcal{A}_{k-1}) P(\mathbf{X}_k|\mathcal{A}_k, \mathcal{E}_{-\delta t+k:k+\delta t}) \end{aligned} \tag{10}$$

The emission model is the scene tracklet model from Sec. 4.1, but with explicit data association $\mathcal{A}_k$. The transition probabilities are defined as $P(\mathcal{A}_k|\mathcal{A}_{k-1}) \propto P_e^\eta P_l^\lambda$. Thus, P_e is the probability for an object to enter the scene, while P_l denotes the probability for an object to leave the scene. To determine the number η of objects entering the scene, respectively the number λ of objects leaving the scene, we again perform frame-by-frame greedy maximum overlap matching. In Eq. (10) the marginals $P(\mathbf{X}_k, \mathcal{A}_k|\mathcal{E}_{-\delta t+1:w+\delta t})$ can be computed with the sum-product algorithm. Finally, the probability of an object being part of the scene is computed by marginalization over all other variables (cf. Sec. 3.3):

$$P(\mathbf{u}_k|\mathcal{E}_{-\delta t+1:w+\delta t}) = \sum_{\mathcal{A}_k} \int \max_i \left(\left[\mathbf{u}_k = \mathbf{o}_k^i \right] \right) P(\mathbf{X}_k, \mathcal{A}_k|\mathcal{E}_{-\delta t+1:w+\delta t}) \, \mathrm{d}\mathbf{X}_k \tag{11}$$

In practice we approximate the integral with MCMC samples as above, however this time only using those that correspond to the data association $\mathcal{A}_k$. Note that the summation over $\mathcal{A}_k$ only requires to consider associations that occur in the sample set.

5 Datasets and Implementation Details

For our experiments we use two datasets: (1) *ETH-Loewenplatz*, which was introduced by [1] to benchmark pedestrian tracking from a moving observer; and (2) a new multi-class dataset we recorded with an onboard camera to specifically evaluate the challenges targeted by our work including realistic traffic scenarios with a large number of small objects, objects of interest from different categories, and higher driving speed.

ETH-Loewenplatz. This publicly available pedestrian benchmark[1] contains 802 frames overall at a resolution of 640×480 pixels of which every 4$^{\text{th}}$ frame is annotated. The sequence, which has been recorded from a driving car in urban traffic at ≈15 fps, comes with a total of 2631 annotated bounding boxes. Fig. 4 shows some examples.

MPI-VehicleScenes. As the above dataset is restricted to pedestrians observed at low driving speeds, we recorded a new multi-class test set consisting of 674 images. The data is subdivided into 5 sequences and has been recorded at a resolution of 752×480 pixels from a driving car at ≈15 fps. Additionally ego-speed and turn rate are obtained from the car's ESP module. See Fig. 5 for sample images. 1331 front view of cars, 156 rear view of cars, and 422 front views of trucks are annotated with bounding boxes. Vehicles appear over a large range of scales from as small as 20 pixels to as large as 270 pixels. 46% of the objects have a height of ≤ 30 pixels, and are thus hard to detect[2].

Object detectors. To detect potential object instances, we use state-of-the-art object detectors. For *ETH-Loewenplatz* we use our motion feature enhanced variant of the HOG framework [20]. SVM margins are mapped to positive values with a soft-clipping function [21].

For our new *MPI-VehicleScenes* test set we employ a multi-class detector based on traditional HOG-features and joint boosting [22] as classifier. It can detect the four object classes *car front, car back, truck front* or *truck back*. The scores are mapped to positive values by means of class-wise sigmoid functions. Note that for our application it is important to explicitly separate front from back views, because the motion model is dependent on the heading direction. This detector was trained on a separate dataset recorded from a driving car, with a similar viewpoint as in the test data.

Scene labeling. Every pixel is assigned to the classes *pedestrian, vehicle, street, lane marking, sky* or *void* to obtain a scene labeling. As features we use the first 16 coefficients of the Walsh-Hadamard transform extracted at five scales (4-64 pixels), along with the pixels' (x, y)-coordinates to account for their location in the image. This algorithm is trained on external data and also employs joint boosting as classifier [23].

Experimental setup. For both datasets and all object classes we use the same set of parameters for our MCMC sampler: $q_{Add} = 0.1$, $q_{Del} = 0.1$, $q_{Dif} = 0.8$, $q_{\mathbf{O}} = 0.8$, $q_{\Theta} = 0.2$. For the HMM's sliding window of Eqn. 10 we choose a length of $W = 7$ frames. Our sampler uses 3,000 samples for burn-in and 20,000 samples to approximate

[1] `http://www.vision.ee.ethz.ch/~aess/dataset/`
[2] The data is publicly available at `http://www.mpi-inf.mpg.de/departments/d2`

the posterior and runs without parallelization at about 1 fps on recent hardware. By running multiple Markov chains in parallel we expect a possible speed-up of one or two orders of magnitude. As we do not have 3D ground truth to assess 3D performance, we project the results back to the images and match them to ground truth annotations with the PASCAL criterion ($intersection/union > 50\%$).

Baselines. As baselines we report both the performance of the object detectors as well as the result of an extended Kalman filter (EKF) atop the detections. The EKFs track the objects independently, but work in 3D state space with the same dynamic models as our MCMC sampler. To reduce false alarms in the absence of an explicit model for new objects entering, tracks are declared valid only after three successive associations. Analogous to our system, the camera ego-motion is compensated using odometry. Best results were obtained, when the last detection's score was used as confidence measure.

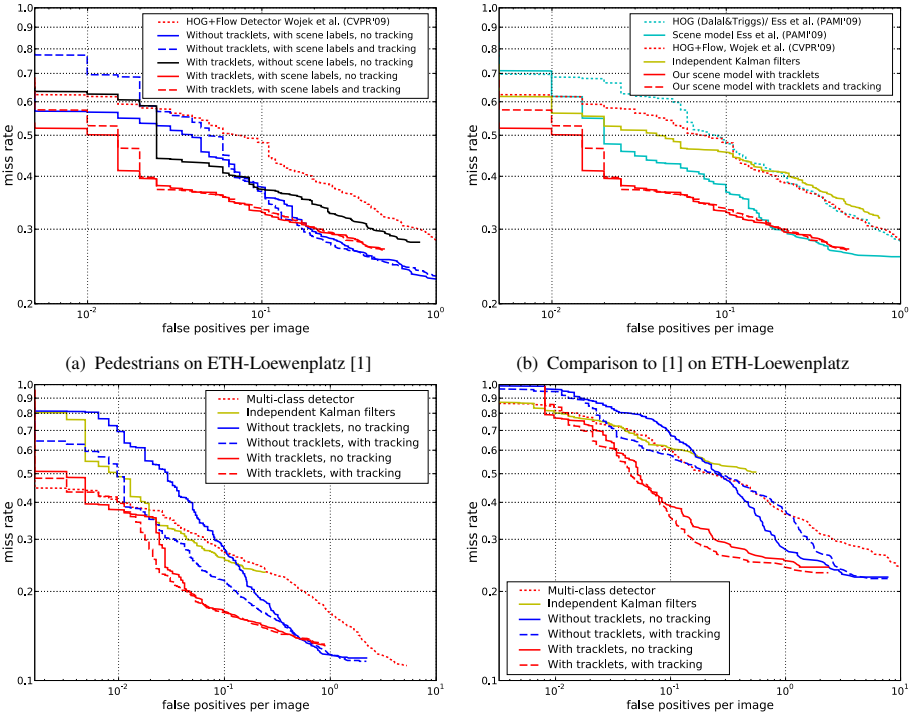
6 Experimental Results

We start by reporting our system's performance for pedestrians on *ETH-Loewenplatz*. Following [1] we consider only people with a height of at least 60 pixels. The authors kindly provided us with their original results to allow for a fair comparison.[3]

In the following we analyze the performance at a constant error rate of 0.1 false positive per image (FPPI). At this error rate the detector (dotted red curve) achieves a miss rate of 48.0%, cf. Fig. 3(a). False detections typically appear on background structures (such as trees or street signs, cf. Fig. 4(a)) or on pedestrians' body parts. When we perform single frame inference (solid blue curve) with our model we improve by 10.4%; additionally adding tracking (dashed blue curve) performs similarly (improvement of 11.6%; see Fig. 4, Fig. 1(a)), but some false positives in the high precision regime are reinforced. When we omit scene labeling but use scene tracklets (black curve) of two adjacent frames our model achieves an improvement of 10.8% compared to the detector. When pixel-labeling information is added to obtain the full model (solid red curve), we observe best results with an improvement of 15.2%. Additionally performing long-term data association (dashed red curve) does not further improve the performance for this dataset: recall has already saturated due to the good performance of the detector, whereas the precision cannot be boosted because the remaining false positives happen to be consistent with the scene model (e.g., human-sized street signs).

Fig. 3(b) compares the system's performance to EKFs and state-of-the-art results by Ess et al. [1]. When we track detections with EKFs (yellow curve) we gain 2.5% compared to the detector, but add additional false detections in the high precision regime, as high-scoring false positives on background structures are further strengthened. Compared to their detector (HOG, [21], dotted cyan curve), the system in [1] achieves an improvement of 11.1% using stereo vision (solid cyan curve), while our monocular approach gains 15.2% over the detector used in our system [20]. We obtain a miss rate of 32.8% using monocular video, which clearly demonstrates the power of the proposed

[3] The original results published in [1] were biased *against* Ess et al., because they did not allow detections slightly < 60 pixels to match true pedestrians ≥ 60 pixels, discarding many correct detections. We therefore regenerated all FPPI-curves.



(a) Pedestrians on ETH-Loewenplatz [1]

(b) Comparison to [1] on ETH-Loewenplatz

(c) Cars on MPI-VehicleScenes

(d) Trucks on MPI-VehicleScenes

Fig. 3. Results obtained with our system. The first row shows results for *pedestrians* on ETH-Loewenplatz and compares to the state-of-the-art. The second row shows results for *truck* and *car* on our new *MPI-VehicleScenes* dataset. Figure best viewed in color.

approach using multi-frame scene tracklets in conjunction with local pixel-labeling. Some example results of our system are depicted in Fig. 1 and 4. Our scene tracklet model allows to stabilize horizon estimation compared to a single-frame model, see Fig. 1(a). Moreover, consistent detections and scene labels boost performance, especially when geometry estimation is more difficult, such as for example in the absence of a sufficient number of objects with confident detections, cf. Fig. 4(b),(c).

Next, we turn to the evaluation on our new *MPI-VehicleScenes* dataset. We note, that *cars rear* are detected almost perfectly, due to the fact that there are only few instances at rather similar scales. Moreover, the test dataset does not contain rear views of trucks. Hence, we will focus on the classes *car front* and *truck front*. In the following, when we refer to cars or trucks this always concerns front views.

For cars the detector achieves a miss rate of 27.0% (see Fig. 3(c)). Independent EKFs improve results by 1.1% to a miss rate of 25.9%. However, in the high precision regime some recall is lost. False positives mainly occur on parts of actual cars, such as on head lights of cars in the near-field, and on rear views of cars – see Fig. 5(a). Thus, in the single-frame case of our approach the false detections are often strengthened rather than weakened by the scene geometry, cf. Fig. 1(b), and in some cases even wrongly

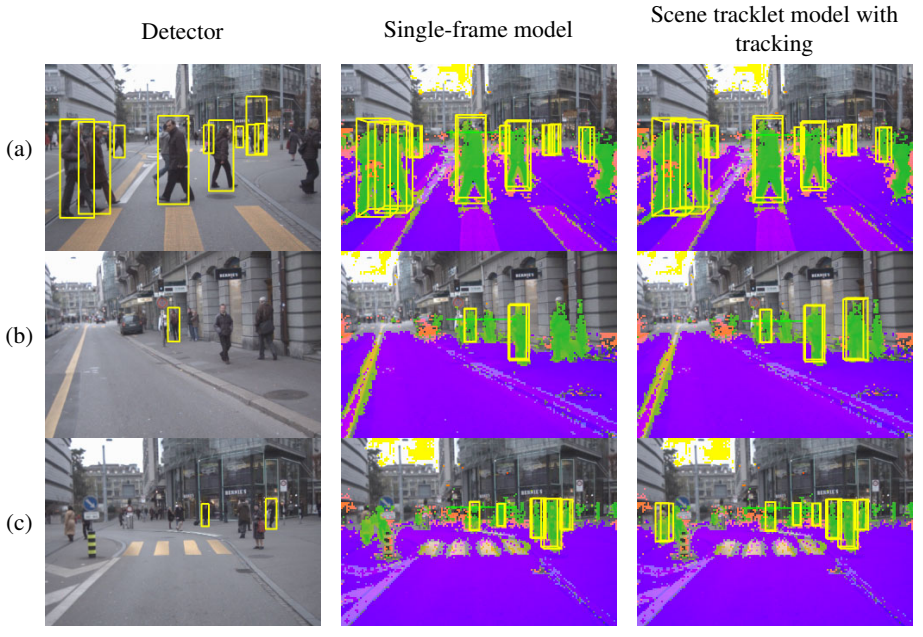


Fig. 4. Sample images showing typical results of our model along with MAP scene labels at a constant error rate of 0.1 false positives per image. *Street* pixels appear in purple, *lane markings* in light purple, *sky* in yellow, *pedestrians* in green and *vehicles* in orange. *Void* (background) pixels are not overlayed. The light green line denotes the estimated horizon.

bias geometry estimation, thus lowering the scores for correct objects. A drop in high precision performance is the result (27.8% miss rate at 0.1 FPPI). This drop can partially be recovered to a miss rate of 21.8%, when an HMM is added for longer-term tracking.

When scene tracklets are employed, many false hypotheses are discarded because of the gross mismatch between their expected and observed dynamics. Consequently, scene tracklets boost performance significantly, resulting in an improvement of 9.9% in miss rate. Adding long-term tracking with the HMM again only slightly improves result over scene tracklets (by 0.1%). Therefore we conclude that the critical source of improvement is *not* to track objects over extended periods of time, but to enforce a *consistent scene interpretation with short tracklets*, by tightly coupling tracklet estimation with geometry fitting and scene labeling.

Finally, we also report results for trucks, cf. Fig. 3(d). For this class our detector has a higher miss rate of 59.4%. This is caused by a significantly higher intra-class variation among trucks and by the fact that the frontal truck detector often fires on cars due to the high visual similarity of the lower part – an example is shown in Fig. 1(b). As a consequence, independent EKFs do not yield an improvement (miss rate 60.9%), as already observed for cars. Similarly, our model using single-frame evidence is not able to disambiguate the classes when both detectors fire, resulting in a miss rate of 67.9%. Though HMM tracking improves this to 57.6%.

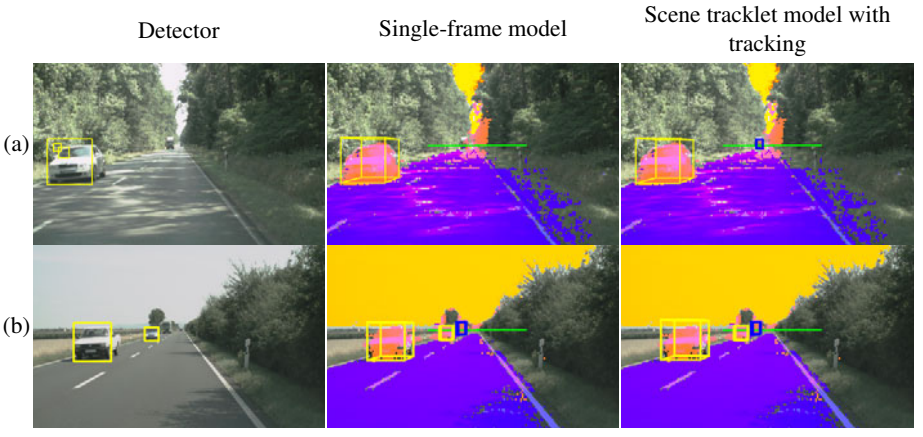


Fig. 5. Example images showing typical results of our model on the *MPI-VehicleScenes* dataset at a constant error rate of 0.1 false positives per image. For color description see Fig. 4.

As in the previous examples, our scene tracklet model is able to suppress many false detections through evidence aggregation across a small number of frames (miss rate 38.6%). Also, weak detections on small scale objects are strengthened, thus recall is improved – cf. Fig. 5(a),(b). Compared to the detector, we improve the miss rate by 20.8%, respectively by 23.9% when also adding HMM tracking.

Discussion. Overall, our experiments for two datasets and four different object classes indicate that our scene tracklet model is able to exploit scene context to robustly infer both the 3D scene geometry and the presence of objects in that scene from a monocular camera. This performance is mainly due to the use of a strong *tracking-by-detection* framework which employs *tracklets* on a scene level thereby leveraging evidence from a number of consecutive frames. The tight coupling with the observation model allows to exploit *3D scene context* as well as to *combine multiple cues* of a detector and from scene labeling. Long-term tracking with an HMM only results in minor additional improvement. In all cases, independent extended 3D Kalman filters cannot significantly improve the output of state-of-the-art object detectors on these datasets, and are greatly outperformed by the integrated state model. On the new multi-class *MPI-VehicleScenes* dataset we outperform state-of-the-art detection by 10.0% for cars, respectively 23.9% for trucks at 0.1 FPPI.

Comparing to other work that integrates detection and scene modeling, we also outperform [1] by 3.8% at 0.1 FPPI for the case of pedestrians, even though we do not use stereo information. At a recall of 60% our model reduces the number of false positives by almost a factor of 4.

7 Conclusion

We have presented a probabilistic 3D scene model, that enables multi-frame tracklet inference on a scene level in a tracking-by-detection framework. Our system performs

monocular 3D scene geometry estimation in realistic traffic scenes, and leads to more reliable detection of objects such as pedestrians, cars, and trucks. We exploit information from object (category) detection and low-level scene labeling to obtain a *consistent 3D description of an observed scene*, even though we only use a single camera. Our experimental results show a clear improvement over top-performing state-of-the-art object detectors. Moreover, we significantly outperform basic Kalman filters and a state-of-the-art stereo camera system [1].

Our experiments underline the observation that objects are valuable constraints for the underlying 3D geometry, and vice versa (cf. [1,5]), so that a joint estimation can improve detection performance.

In future work we plan to extend our model with a more elaborate tracking framework with long-term occlusion handling. Moreover, we aim to model further components and objects of road scenes such as street markings and motorbikes. It would also be interesting to explore the fusion with complementary sensors such as RADAR or LIDAR, which should allow for further improvements.

Acknowledgement. We thank Andreas Ess for providing his data and results.

References

1. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: Robust multi-person tracking from a mobile platform. PAMI 31 (2009)
2. Gavrila, D.M., Munder, S.: Multi-cue pedestrian detection and tracking from a moving vehicle. In: IJCV, vol. 73 (2007)
3. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
4. Kaucic, R., Perera, A.G., Brooksby, G., Kaufhold, J., Hoogs, A.: A unified framework for tracking through occlusions and across sensor gaps. In: CVPR (2005)
5. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: IJCV, vol. 80 (2008)
6. Torralba, A.: Contextual priming for object detection. In: IJCV, vol. 53 (2003)
7. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
8. Ess, A., Müller, T., Grabner, H., Van Gool, L.: Segmentation-based urban traffic scene understanding. In: BMVC (2009)
9. Brostow, G., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using SfM point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
10. Tu, Z., Chen, X., Yuille, A.L., Zhu, S.C.: Image parsing: Unifying segmentation, detection, and recognition. IJCV 63 (2005)
11. Shashua, A., Gdalyahu, Y., Hayun, G.: Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In: IVS (2004)
12. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Van Gool, L.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV (2009)
13. 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)

14. Li, Y., Huang, C., Nevatia, R.: Learning to associate: HybridBoosted multi-target tracker for crowded scene. In: CVPR (2009)
15. Khan, Z., Balch, T., Dellaert, F.: MCMC-based particle filtering for tracking a variable number of interacting targets. PAMI 27 (2005)
16. Zhao, T., Nevatia, R., Wu, B.: Segmentation and tracking of multiple humans in crowded environments. PAMI 30 (2008)
17. Isard, M., MacCormick, J.: BraMBLe: a Bayesian multiple-blob tracker. In: ICCV (2001)
18. Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82 (1995)
19. Gilks, W., Richardson, S., Spiegelhalter, D. (eds.): Markov Chain Monte Carlo in Practice. Chapman and Hall, Boca Raton (1995)
20. Wojek, C., Walk, S., Schiele, B.: Multi-cue onboard pedestrian detection. In: CVPR (2009)
21. Dalal, N.: Finding People in Images and Videos. PhD thesis, Institut National Polytechnique de Grenoble (2006)
22. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. PAMI 29 (2007)
23. Wojek, C., Schiele, B.: A dynamic conditional random field model for joint labeling of object and scene classes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 733–747. Springer, Heidelberg (2008)

Blocks World Revisited: Image Understanding Using Qualitative Geometry and Mechanics

Abhinav Gupta, Alexei A. Efros, and Martial Hebert

Robotics Institute, Carnegie Mellon University

Abstract. Since most current scene understanding approaches operate either on the 2D image or using a surface-based representation, they do not allow reasoning about the physical constraints within the 3D scene. Inspired by the "Blocks World" work in the 1960's, we present a *qualitative* physical representation of an outdoor scene where objects have volume and mass, and relationships describe 3D structure and mechanical configurations. Our representation allows us to apply powerful global geometric constraints between 3D volumes as well as the laws of statics in a qualitative manner. We also present a novel iterative "interpretation-by-synthesis" approach where, starting from an empty ground plane, we progressively "build up" a physically-plausible 3D interpretation of the image. For surface layout estimation, our method demonstrates an improvement in performance over the state-of-the-art [9]. But more importantly, our approach automatically generates **3D parse graphs** which describe qualitative geometric and mechanical properties of objects and relationships between objects within an image.

1 Introduction

What does it mean to understand a visual scene? One popular answer is simply *naming* the objects present in the image – "building", "road", etc. – and possibly locating them in the image. However this level of understanding is somewhat superficial as it tells us little about the underlying structure of the scene. To really understand an image it is necessary to probe deeper – to acquire some notion of the geometric scene layout, its free space, walkable surfaces, qualitative occlusions and depth relationships, etc. Object naming has also a practical limitation: due to the heavy-tailed distribution of object instances in natural images, a large number of objects will occur too infrequently to build usable recognition models, leaving parts of the image completely unexplained. To address these shortcomings, there has been a recent push toward more geometric, rather than semantic, approaches to image understanding [8,18]. The idea is to learn a mapping between regions in the image and planar surfaces in the scene. The resulting labeled image can often be "popped-up" into 3D by cutting and folding it at the appropriate region boundaries, much like a children's pop-up book. However this process has two limitations. First, since surface orientation labels are being estimated per region or per super-pixel, it is difficult to enforce global consistency, which often leads to scene models that are physically

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 482–496, 2010.

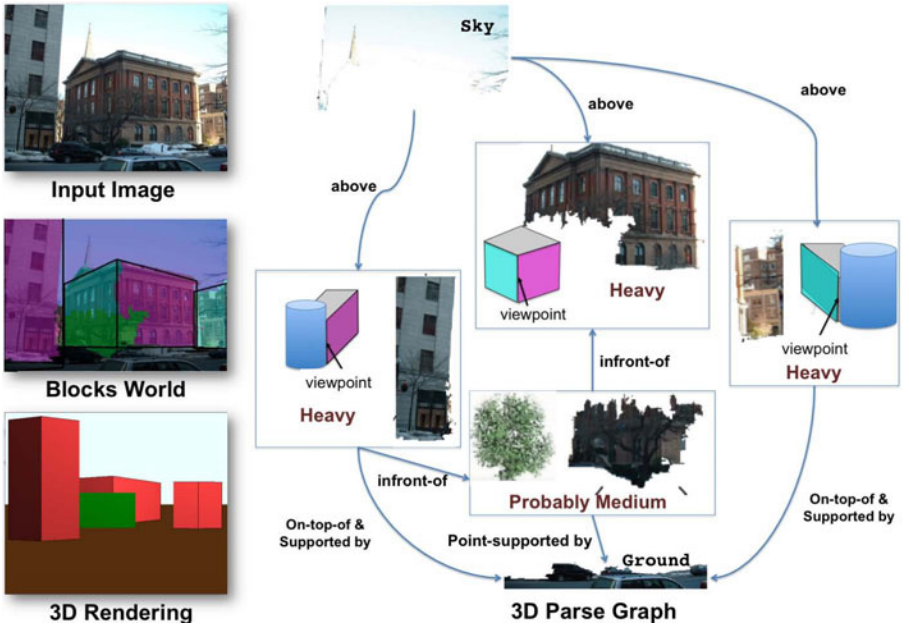


Fig. 1. Example output of our automatic scene understanding system. The 3D parse graph summarizes the inferred object properties (physical boundaries, geometric type, and mechanical properties) and relationships between objects within the scene. See more examples on project webpage.

impossible or highly unlikely. Second, even if successful, these pop-up models (also known as "billboards" in graphics) lack the physical substance of a true 3D representation. Like in a Potemkin village, there is nothing behind the pretty façades!

This paper argues that a more physical representation of the scene, where objects have volume and mass, can provide crucial high-level constraints to help construct a globally-consistent model of the scene, as well as allow for powerful ways of understanding and interpreting the underlying image. These new constraints come in the form of geometric relationships between 3D volumes as well as laws of *statics* governing the behavior of forces and torques. Our main insight is that the problem can be framed *qualitatively*, without requiring a metric reconstruction of the 3D scene structure (which is, of course, impossible from a single image). Figure 1 shows a real output from our fully-automatic system.

The paper's main contributions are: (a) a novel qualitative scene representation based on volumes (blocks) drawn from a small library; (b) the use of 3D geometry and mechanical constraints for reasoning about scene structure; (c) an iterative Interpretation-by-Synthesis framework that, starting from the empty ground plane, progressively "builds up" a consistent and coherent interpretation of the image; (d) a top-down segmentation adjustment procedure where partial scene interpretations guide the creation of new segment proposals.

Related Work: The idea that the basic physical and geometric constraints of our world (so-called *laws of nature*) play a crucial role in visual perception goes back at least to Helmholtz and his argument for "unconscious inference". In computer vision, this theme can be traced back to the very beginnings of our discipline, with Larry Roberts arguing in 1965 that *"the perception of solid objects is a process which can be based on the properties of three-dimensional transformations and the laws of nature"* [17]. Roberts' famous Blocks World was a daring early attempt at producing a complete scene understanding system for a closed artificial world of textureless polyhedral shapes by using a generic library of polyhedral block components. At the same time, researchers in robotics also realized the importance of physical stability of block assemblies since many block configurations, while geometrically possible, were not physically stable. They showed how to generate plans for the manipulation steps required to go from an initial configuration to a target configuration such that at any stage of assembly the blocks world remained stable [1]. Finally, the *MIT Copy Demo* [21] combined the two efforts, demonstrating a robot that could visually observe a blocks world configuration and then recreate it from a pile of unordered blocks (recently [2] gave a more sophisticated reinterpretation of this idea, but still in a highly constrained environment).

Unfortunately, hopes that the insights learned from the blocks world would carry over into the real world did not materialize as it became apparent that algorithms were too dependent on its very restrictive assumptions (perfect boundary detection, textureless surfaces, etc). While the idea of using 3D geometric primitives for understanding real scenes carried on into the work on generalized cylinders and resulted in some impressive demos in the 1980s (e.g., ACRONYM [16]), it eventually gave way to the currently dominant appearance-based, semantic labeling methods, e.g., [19,5]. Of these, the most ambitious is the effort of S.C.Zhu and colleagues [23] who use a hand-crafted stochastic grammar over a highly detailed dataset of labelled objects and parts to hierarchically parse an image. While they show impressive results for a few specific scene types (e.g., kitchens, corridors) the approach is yet to be demonstrated on more general data.

Most related to our work is a recent series of methods that attempt to model geometric scene structure from a single image: inferring qualitative geometry of surfaces [8,18], finding ground/vertical "fold" lines [3], grouping lines into surfaces [22,13], estimating occlusion boundaries [10], and combining geometric and semantic information [9,14,4]. However, these approaches do not model the global interactions between the geometric entities within the scene, and attempts to incorporate them at the 2D image labeling level [15,12] have been only partially successful. While single volumes have been used to model simple building interiors [6] and objects (such as bed) [7], these approaches do not model geometric or mechanical inter-volume relationships. And while modeling physical constraints has been used in the context of dynamic object relationships in video [20], we are not aware of any work using them to analyze static images.

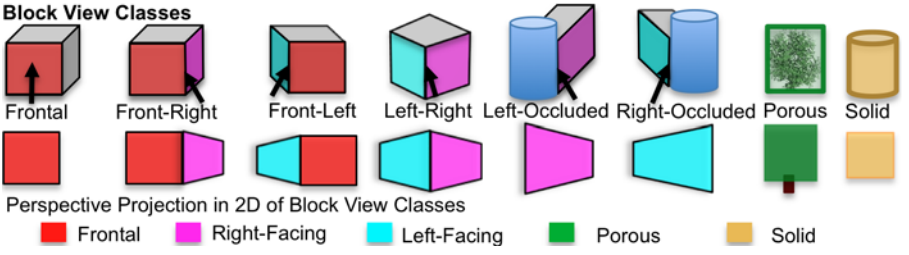


Fig. 2. Catalog of the possible block view classes and associated 2D projections. The 3D blocks are shown as cuboids although our representation imposes no such constraints on the 3D shape of the block. The arrow represents the camera viewpoint.

2 Overview

Our goal is to obtain a rich physical understanding of an outdoor scene (under camera assumptions similar to [9]) in which objects have volume and mass, and inter-object relationships are governed by the geometry and mechanics of the 3D world. Unfortunately, from a single image, it is next to impossible to estimate a precise, metric 3D representation for a generic scene. Our proposed solution is to represent 3D objects *qualitatively*, as one or more convex "blocks". We define a block as an image region represented by one of a small class of geometric primitives and qualitative density distribution(described below). The block is our basic unit of reasoning within the 3D scene. While a block is purposefully kept somewhat under-constrained to allow 3D reasoning even with a large degree of uncertainty, it contains enough information to produce a reasonable 3D rendering under some assumptions (see Figures 1 and 8).

2.1 Block Representation

Geometrically, we want to qualitatively represent the 3D space occupied by a block with respect to the camera viewpoint. Using the convexity assumption, we can restrict the projection of each block in the image to one of the eight *block-view classes* shown in Figure 2. These classes correspond to distinct aspects of a block over possible viewpoints. Figure 3(a) shows a few examples of extracted blocks in our test dataset.

We also want to represent the gravitational force acting on each block and the extent to which a block can support other blocks, which requires knowing the density of the block. Estimating density using visual cues alone is a hard problem. But it turns out that there is enough visual regularity in the world to be able to coarsely estimate a *density class* of each block: "light"(e.g. trees and bushes), "medium" (e.g. humans) and "high-density" (e.g. buildings). These three classes span the spectrum of possible densities and each class represents order-of-magnitude difference with respect to the other classes.

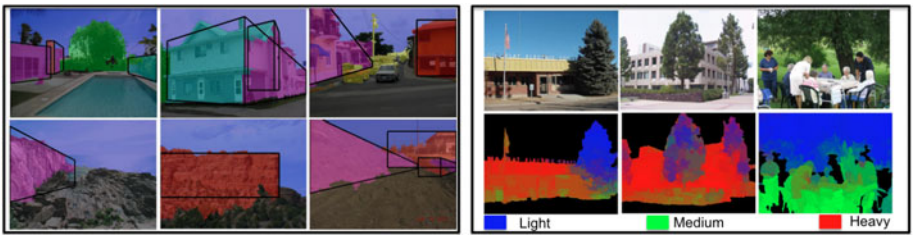


Fig. 3. (a) Examples of extracted blocks in the wide variety of images. (b) Examples of super-pixel based density estimation of our approach.

2.2 Representing Relationships

Instead of worrying about absolute depth, we only encode pairwise depth relationships between blocks, as defined by the painters' algorithm. For a given pair of blocks $\mathcal{B}_i$ and $\mathcal{B}_j$, there are three possible depth relationships: "infront of","behind" and "no-relationship". Block $\mathcal{B}_i$ is "infront of" $\mathcal{B}_j$ if $\mathcal{B}_i$ transitively occludes $\mathcal{B}_j$ in the current viewpoint and vice-versa.

To represent the mechanical configuration of the scene we use support relationships between pair of blocks. For a given pair of blocks $\mathcal{B}_i$ and $\mathcal{B}_j$, there are three possible support relationships: "supports","supported by" and "no-relationship". Figure 1 shows examples of the depth and support relationships extracted by our approach (see edges between the blocks).

2.3 Geometric and Mechanical Constraints

Having a rich physical representation (blocks with masses instead of popped-up planes) of the scene can provide additional powerful global constraints which are vital for successful image understanding. These additional constraints are:

Static Equilibrium. Under the static world assumption, the forces and torques acting on a block should cancel out (Newton's first law). We use this to derive constraints on segmentation of objects, and estimating depth and support relationships. For example, in Figure 4(c), the orange segment is rejected since it leads to the orange block which is physically unstable due to unbalanced torques.

Support Force Constraint. A supporting object should have enough strength to provide contact reactionary forces on the supported objects. We utilize density to derive constraints on the support relationships and on the relative strengths of the support*ing* and support*ed* bodies.

Volume Constraint. All the objects in the world must have finite volumes and cannot inter-penetrate each other. Figure 4(b) shows an example of how this constraint can help in rejecting bad segmentation hypotheses (red and yellow surfaces in the image cannot belong to different objects) since that would lead to 3D intersection of blocks.

Block Representation

Type	Classes
View Classes	"frontal", "front-right", "front-left", "left-right", "left-occluded"," right-occluded", "porous", "solid-non planar"
Density Classes	"light", "medium" and "heavy"

Relationships Representation

Type	Classes
Depth	"infront", "behind", "no-relationship"
Support	"Supports", "Supported-by"," no-relationship"

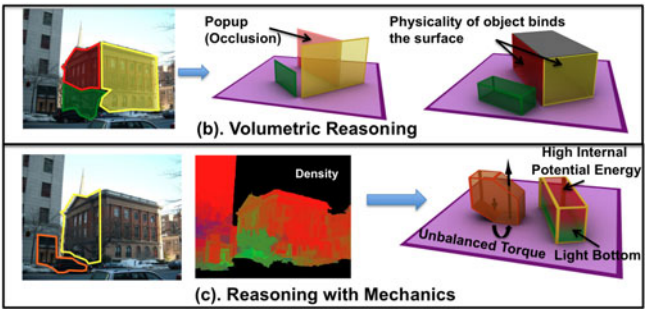


Fig. 4. (a) Our Representation (b)Role of Volume Constraint: The finite volume constraint binds the two surfaces together whereas in pop-ups the relative locations of surfaces is not constrained. (c) Role of Static Constraints: Unbalanced torque leads to rejection of the orange segment. The low internal stability (lighter bottom and heavier top) of yellow segment leads to its rejection.

Depth Ordering Constraint. The depth ordering of the objects with respect to the camera viewpoint should be consistent with the projected regions in the image as explained in Section 3.6.

3 Assembling Blocks World: Interpretation by Synthesis

We need to use the constraints described earlier to generate a physically plausible scene interpretation. While one can apply these constraints to evaluate all possible interpretations of the scene, such an approach is computationally infeasible because of the size of the hypotheses space. Similarly, probabilistic approaches like Bayesian networks where all attributes and segmentations are inferred simultaneously are also infeasible due to the large number of variables with higher-order clique constraints (physical stability has to be evaluated in terms of multiple bodies simultaneously interacting with each other). Instead, we propose an iterative "interpretation by synthesis" approach which grows the image interpretation by adding regions one by one, such that confident regions are interpreted first in order to guide the interpretation of other regions. Our approach is inspired by robotics systems which perform block assembly to reach a target configuration of blocks [11]. One nice property of our approach is that, at any given stage in the assembly, the partial interpretation of the scene satisfies all the geometrical and mechanical constraints described above.

3.1 Initialization

Before we begin assembling a blocks world from an input image, we need to generate an inventory of hypotheses that are consistent with the image. We use a multiple segmentation approach to generate a large number of regions that can correspond to blocks. We use the occlusion boundary segmenter of [10],

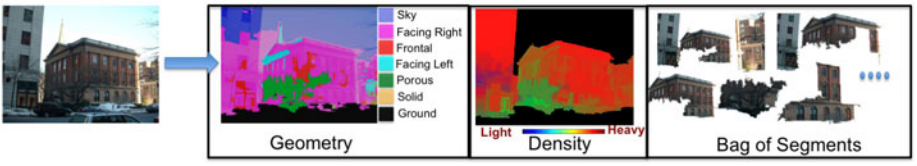


Fig. 5. Initialization: We use multiple segmentation to generate block hypothesis and a super-pixel based classifier for surface geometry and material-density.

which starts with a set of superpixels and iteratively coarsens the segmentation by combining regions selected by a local classifier. Since our goal is to generate multiple possible segmentations, we implement a randomized version of the same where the coarsening is done multiple times with different seeds and different parameters and to generate 17 possible segmentations for each image.

We estimate the local surface layout labels using [8] to estimate the view class of each block hypothesis. To estimate the density of each block hypothesis, we implemented a superpixel-based local classifier which learns a mapping between image features (same as in [8]) and object densities. We learn the density classifier on a set of training images labeled with the three density classes. Figure 3(b) shows a few examples of per-superpixel density estimation for a few images. A more detailed quantitative analysis of our density classifier is provided in Section 4. Figure 5 shows the initialization of our approach. Using the "ground" and "sky" regions from the estimated surface layout, we initialize our empty blocks world consisting of ground and sky. We are now ready to start adding blocks.

3.2 Searching Block Configurations

The goal is to assemble a blocks world that is consistent with the input image. However, which block hypotheses should be used and the order in which they should be added are not known *a priori*. We employ a search strategy where for a current blocks world $\mathcal{W}_t$, we propose k possible block hypotheses. These k hypotheses are selected based on a set of local criteria: confidence in surface layout geometry, density estimation, internal physical stability (heavy-bottom and light top, c.f. Section 3.5), and support likelihood (estimated based on contact with support surfaces). Each of these k hypotheses is then provisionally "placed" into the current blocks world and scored using a cost function described in the next section. The hypothesis with the lowest cost is accepted and the corresponding block is added to reach a new configuration $\mathcal{W}_{t+1}$. The process of generating new hypothesis and cost estimation is then repeated for $\mathcal{W}_{t+1}$, until all the regions in the image have been explained. For all experiments in the paper we use $k = 4$.

3.3 Evaluating Proposals

Given a candidate block $\mathcal{B}_i$, we want to estimate its associated mechanical and geometrical properties and its relationship to the blocks already placed in the scene to minimize the following cost function:

$$\mathcal{C}(\mathcal{B}_i) = \mathcal{F}_{geometry}(\mathcal{G}_i) + \sum_{S \in ground,sky} \mathcal{F}_{contacts}(\mathcal{G}_i, S) + \mathcal{F}_{intra}(\mathcal{S}_i, \mathcal{G}_i, d) + \sum_{j \in blocks} \mathcal{F}_{stability}(\mathcal{G}_i, S_{ij}, \mathcal{B}_j) + \mathcal{F}_{depth}(\mathcal{G}_i, S_{ij}, \mathcal{D}), \quad (1)$$

where $\mathcal{G}_i$ represents the estimated block-view class, $\mathcal{S}_i$ corresponds to the region associated with the block, d is the estimated density of the block, S_{ij} represent support relationships and $\mathcal{D}$ represents partial-depth ordering obtained from depth relationships. $\mathcal{F}_{geometry}$ measures the agreement of the estimated block view class with the superpixel-based surface layout estimation [8] and $\mathcal{F}_{contacts}$ measures the agreement of geometric properties with ground and sky contact points (Section 3.4). $\mathcal{F}_{intra}$ and $\mathcal{F}_{stability}$ measure physical stability within a single block and with respect to other blocks respectively (Section 3.5). Finally, $\mathcal{F}_{depth}$ measures the agreement of projection of blocks in the 2D image plane with the estimated depth ordering (Section 3.6).

Minimizing the cost function over all possible configurations is too costly. Instead, Figure 6 illustrates our iterative approach for evaluating a block hypothesis by estimating its geometric and mechanical properties such that the cost function $\mathcal{C}$ is approximately minimized. We first estimate the block-view class of the new block by minimizing $\mathcal{F}_{geometry}$ and $\mathcal{F}_{contacts}$ (Figure 6c). Using the block-view class, we compute the stability of the blocks under various support relationships by minimizing $\mathcal{F}_{intra}$ and $\mathcal{F}_{stability}$ (Figure 6d). We then use the estimated block-view class and support relationships to estimate a partial depth ordering of the blocks in the image . This minimizes the final term in our cost function, $\mathcal{F}_{depth}$ (Figure 6e). Block-geometric properties, physical stability analysis and partial depth ordering of blocks in the scene provide important cues for improving segmentation. Therefore, after computing these properties, we perform a segmentation adjustment step (Figure 6f). The final computed score is then returned to the top-level search procedure which uses it to select the best block to add. We now discuss each of the steps above in detail.

3.4 Estimating Geometry

Estimating the geometric attributes of a block involves inferring the block view class (Figure 2) and the 2D location of a convex corner, which we call "foldedge" using cues from the surface layout estimation and ground and sky contact points. Let us assume that we are adding block $\mathcal{B}_i$ with the associated segment $\mathcal{S}_i$ in the 2D image plane. We need to estimate the block-view class $\mathcal{G}_i$ and the foldedge location f_i given the surface-geometric labels for superpixels g and the evidence from ground and sky contact points in the image plane (C_i^G and C_i^S). This can be written as:

$$P(\mathcal{G}_i, f_i | g, \mathcal{S}_i, C_i^G, C_i^S) \propto P(g|\mathcal{G}_i, f_i, \mathcal{S}_i) P(C_i^G|\mathcal{G}_i, f_i) P(C_i^S|\mathcal{G}_i, f_i). \quad (2)$$

The first term indicates how well the surface layout matches the hypothesized block view class, and the second and third terms indicate the agreement between

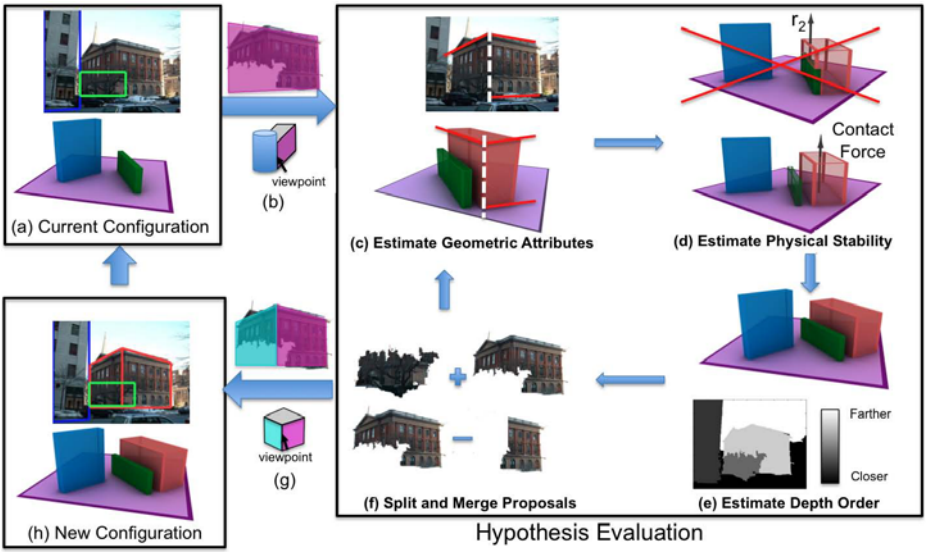


Fig. 6. Our approach for evaluating block hypothesis and estimating the associated cost of placing a block

block view class and ground/sky contact points, respectively. We compute the first term as

$$P(g|\mathcal{G}_i, f_i, \mathcal{S}_i) = \frac{1}{Z} \sum_{s \in \mathcal{S}_i} P(g_s|\mathcal{G}_i, f_i, s), \tag{3}$$

where s is a super-pixel in segment $\mathcal{S}_i$. $P(g_s|\mathcal{G}_i, f_i, s)$ represents the agreement between the predicted block geometry, $(\mathcal{G}_i, f_i)$ and the result of the surface layout estimation algorithm for superpixel s. For example, if the block is associated with "front-right" view class and the superpixel is on the right of the folding edge, then $P(g_s|\mathcal{G}_i, f_i, s)$ would be the probability of the superpixel being labeled right-facing by the surface layout estimation algorithm.

For estimating the contact points likelihood term, we use the constraints of perspective projection. Given the block geometry and the folding edge, we fit straight lines l_g and l_s to the the ground and sky contact points, respectively, and we verify if their slopes are in agreement with the surface geometry: for a frontal surface, l_g and l_s should be horizontal, and for left- and right-facing surfaces l_g and l_s should intersect on the horizon line.

3.5 Estimating Physical Stability

Our stability measure (Figure 6d) consists of three terms. (1) **Internal Stability:** We prefer blocks with low potential energies, that is, blocks which have heavier bottom and lighter top. This is useful for rejecting segmentations which

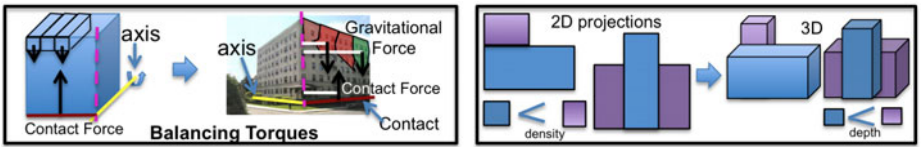


Fig. 7. (a) Computation of torques around contact lines. (b) Extracted depth constraints are based on convexity and support relationships among blocks.

merge two segments with different densities, such as the lighter object below the heavier object shown on Figure 4(c). For computing internal stability, we rotate the block by a small angle, $\delta\theta$, (clockwise and anti-clockwise) around the center of each face; and compute the change in potential energy of the block as:

$$\Delta P_i = \sum_{c \in \{light, medium, heavy\}} \sum_{s \in \mathcal{S}_i} p(d_s = c) m_c \delta h_s, \tag{4}$$

where $p(d_s = c)$ is the probability of assigning density class c to superpixel s, δh_s is the change in height due to the rotation and m_c is a constant representing the density class. The change in potential energy is a function of three constants. Using constraints such as $\rho_{hm} = \frac{m_{heavy}}{m_{medium}} > 1$ and $\rho_{lm} = \frac{m_{light}}{m_{medium}} < 1$, we compute the expected value of ΔP_i with respect to the ratio of densities (ρ_{hm} and ρ_{lm}). The prior on ratio of densities for the objects can be derived using density and the frequency of occurrence of different materials in our training images. (2) **Stability:** We compute the likelihood of a block being stable given the density configuration and support relations. For this, we first compute the contact points of the block with the supporting block and then compute the torque due to gravitational force exerted by each superpixel and the resultant contact force around the contact line (Figure 7a). This again leads to torque as a function of three constants and we use similar qualitative analysis to compute the stability. (3) **Constraints from Block Strength:** We also derive constraint on support attributes based on the densities of the two blocks possibly interacting with each other. If the density of the supporting block is less than density of the supported block; we then assume that the two blocks are not in physical contact and the block below occludes the contact of the block above with the ground.

3.6 Extracting Depth Constraints

The depth ordering constraints (Figure 6(e)) are used to guide the next step of refining the segmentation by splitting and merging regions. Computing depth ordering requires estimating pairwise depth constraints on blocks and then using them to form global depth ordering. The rules for inferring depth constraints are shown in Figure 7(b). These pairwise constraints are then used to generate a global partial depth ordering via a simple constraint satisfaction approach.

3.7 Creating Split and Merge Proposals

This final step involving changes to the segmentation (Figure 6f) is crucial because it avoids the pitfalls of previous systems which assumed a fixed, initial segmentation (or even multiple segmentations) and were unable to recover from incorrect or incomplete groupings. For example, no segmentation algorithm can group two regions separated by an occluding object because such a merge would require reasoning about depth ordering. It is precisely this type of reasoning that the depth ordering estimation of Section 3.6 enables. We include segmentation in the interpretation loop and use the current interpretation of the scene to generate more segments that can be utilized as blocks in the blocks world.

Using estimated depth relationships and block view classes we create **merge proposals** where two or more non-adjacent segments are combined if they share a block as neighbor which is estimated to be in front of them in the current viewpoint. In that case, the shared block is interpreted as an occluder which fragmented the background block into pieces which the merge proposal attempts to reconnect. We also create additional merge proposals by combing two or more neighboring segments. **Split proposals** divide a block into two or more blocks if the inferred properties of the block are not in agreement with confident individual cues. For example, if the surface layout algorithm estimates a surface as frontal with high-confidence and our inferred geometry is not frontal, then the block is divided to create two or more blocks that agree with the surface layout. The split and merge proposals are then evaluated by a cost function whose terms are based on the confidence in the estimated geometry and physical stability of the new block(s) compared to previous block(s). In our experiments, approximately 11% of the blocks are created using the resegmentation procedure.

4 Experimental Results

Since there has been so little done in the area of qualitative volumetric scene understanding, there are no established datasets, evaluation methodologies, or even much in terms of relevant previous work to compare against. Therefore, we will present our evaluation in two parts: 1) qualitatively, by showing a few representative scene parse results in the paper, and a wide variety of results on the project webpage[1]; 2) quantitatively, by evaluating *individual components* of our system and, when available, comparing against the relevant previous work.

Dataset: We use the dataset and methodology of Hoiem et. al [9] for comparison. This dataset consists of 300 images of outdoor scenes with ground truth surface orientation labeled for all images, but occlusion boundaries are only labelled for 100 images. The first 50 (of the 100) are used for training the surface segmentation [8] and occlusion reasoning [10] of our segmenter. The remaining 250 images are used to evaluate our blocks world approach. The surface classifiers are trained and tested using five-fold cross-validation just like in [9].

[1] http://www.cs.cmu.edu/~abhinavg/blocksworld

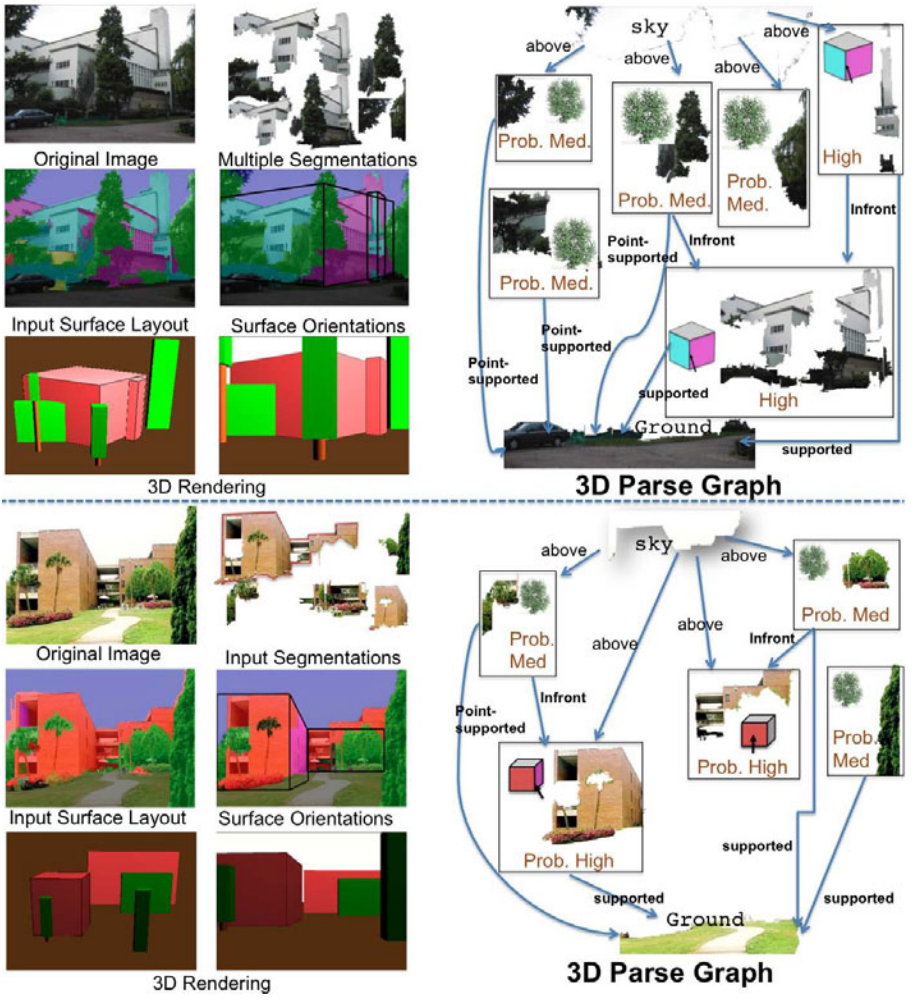


Fig. 8. Some qualitative results obtained using our approach. (Top) Our approach combines the two faces of building separated due to presence of occluder. (Bottom) Our approach correctly labels the right face by rejecting bad segments using mechanics.

Qualitative: Figure 8 shows two examples of complete interpretation automatically generated by the system and a 3D toy blocks world generated in VRML. In the top example, the building is occluded by a tree in the image and therefore none of the previous approaches can combine the two faces of the building to produce a single building region. In a pop-up based representation, the placement of the left face is unconstrained due to the contact with ground not being visible. However, in our approach volumetric constraints aid the reasoning process and combine the two faces to produce a block occluded by the tree. The bottom example shows how statics can help in selecting the best blocks and improve block-view estimation. Reasoning about mechanical constraints rejects

Fig. 9. Additional results to show the qualitative performance on wide variety of scenes

the segment corresponding to the whole building (due to unbalanced torque). For the selected convex block, the cues from ground and sky contact points aid in proper geometric classification of the block. Figure 9 shows a few other qualitative examples with the overlaid block and estimated surface orientations.

Quantitative: We evaluate various components of our system separately. It is not possible to quantitatively compare the performance of the entire system because there is no baseline approach. For surface layout estimation, we compare against the state-of-the-art approach [9] which combines occlusion boundary reasoning and surface layout estimation (we removed their recognition component from the system). On the main geometric classes ("ground","vertical" and "sky"), our performance is nearly identical, so we focus on vertical sub-classes (frontal, right-facing, left-facing, porous and solids). For this comparison, we discard the super-pixels belonging to ground and sky and evaluate the performance over the vertical super-pixels. With this evaluation metric, [9] has an average performance of 68.8%. whereas our approach performs at 73.72%. Improving vertical subclass performance on this dataset is known to be extremely hard; in fact the two recent papers on the topic [15,12] show no improvement over [9].

We compare the segmentation performance to [9] on 50 images whose ground truth (segmented regions and occlusion boundaries) is publicly available [10]. For

Table 1. Quantitative Evaluation

	Surface Layout	Segmentation	Density Class.
Hoiem et. al (CVPR 2008)	68.8%	0.6532	-
This paper	73.72%	0.6885	69.32%

Fig. 10. Failure Cases: We fail to recover proper geometry when [8] is confident about the wrong predictions. Our approach also hallucinates blocks whenever there is a possible convex corner. For example, in the second image the wrong surface layout predicts a false corner and our approach strengthens this volumetric interpretation.

comparing the block segmentation performance we use the Best Spatial Support (BSS) metric. We compute the best overlap score of each ground truth segment and then average it over all ground-truth segments to obtain the BSS score. As can be seen, our approach improves the segmentation performance of [9] by approximately 5.4%. We also evaluated the importance of different terms in the cost function. Without the ground/sky contact term, the surface layout performance falls by 1.9%. The removal of physical stability terms cause the surface layout and segmentation performance to fall by 1.5% and 1.8% respectively.

Acknowledgement. This research was supported by NSF Grant IIS-0905402 and Guggenheim Fellowship to Alexei Efros.

References

1. Blum, M., Griffith, A., Neumann, B.: A stability test for configuration of blocks. In: TR-AI Memo (1970)
2. Brand, M., Cooper, P., Birnbaum, L.: Seeing physics, or: Physics is for prediction. In: Physics-Based Modeling in Computer Vision (1995)
3. Delage, E., Lee, H., Ng., A.Y.: A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In: CVPR (2006)
4. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
5. Gupta, A., Davis, L.: Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 16–29. Springer, Heidelberg (2008)
6. Hedau, V., Hoiem, D., Forsyth, D.: Recovering the spatial layout of cluttered rooms. In: ICCV (2009)
7. Hedau, V., Hoiem, D., Forsyth, D.: Thinking inside the box: Using appearance models and context based on room geometry. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV. Part IV, LNCS, vol. 6314, Springer, Heidelberg (2010)
8. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. In: IJCV (2007)
9. Hoiem, D., Efros, A.A., Hebert, M.: Closing the loop on scene interpretation. In: CVPR (2008)
10. Hoiem, D., Stein, A., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from a single image. In: ICCV (2007)

11. Ikeuchi, K., Suehiro, T.: Toward an assembly plan from observation: Task recognition with polyhedral objects. In: Robotics & Automation (1994)
12. Lazebnik, S., Raginsky, M.: An empirical bayes approach to contextual region classification. In: CVPR (2009)
13. Lee, D., Hebert, M., Kanade., T.: Geometric reasoning for single image structure recovery. In: CVPR (2009)
14. Nedovic, V., Smeulders, A., Redert, A., Geusebroek, J.: Stages as models of scene geometry. In: PAMI (2010)
15. Ramalingam, S., Kohli, P., Alahari, K., Torr, P.: Exact inference in multi-label crfs with higher order cliques. In: CVPR (2008)
16. Brooks, R., Creiner, R., Binford, T.: The acronym model-based vision system. In: IJCAI (1979)
17. Roberts, L.: Machine perception of 3-d solids. In: PhD. Thesis (1965)
18. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. PAMI (2009)
19. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
20. Siskind, J.: Visual event classification via force dynamics. In: AAAI (2000)
21. Winston, P.H.: The mit robot. In: Machine Intelligence (1972)
22. Yu, S., Zhang, H., Malik., J.: Inferring spatial layout from a single image via depth-ordered grouping. In: CVPR Workshop (2008)
23. Zhu, S., Mumford, D.: A stochastic grammar of images. In: Found. and Trends. in Graph. and Vision (2006)

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 IV, LNCS 6314, pp. 497–510, 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

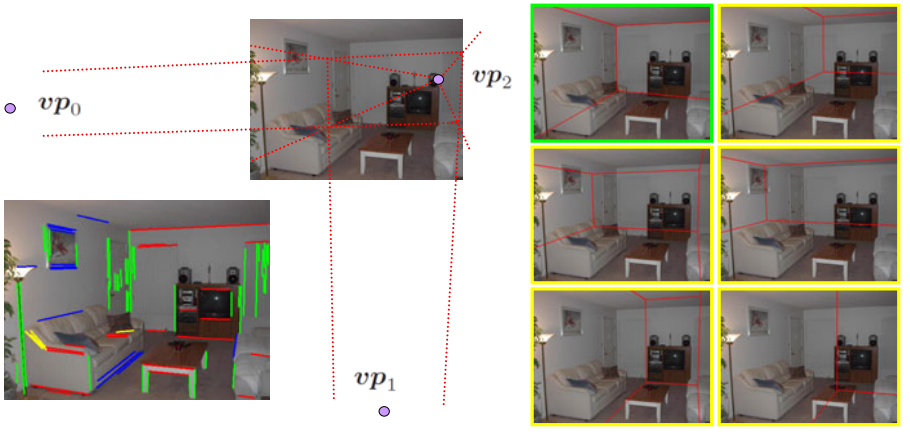


Fig. 2. Lower-Left: We have 3 groups of lines (shown in R, G, B) corresponding to the 3 vanishing points respectively. There are also "outlier" lines (shown in yellow) which do not belong to any group. **Upper-Left**: A candidate "box" specifying the boundaries between the ceiling, walls and floor is 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 $\mathbf{\Psi}$ and $\mathbf{E}^0$.

Algorithm 1. Overall Learning Procedure

```
Input: {(x_i, y_i)}_{i=1}^m, C, ε_final
Output: w
Cons ← ∅
ε ← ε_0
repeat
    for i = 1 to m do
        find (ŷ, ĥ) by solving (5) using Algorithm 2
        if the constraint in (4) corresponding to (ŷ, ĥ) is violated more than ε then
            add the constraint to Cons
        end if
    end for
    update w by solving the QP given Cons
    for i = 1 to m do
        update h_i by solving (6)
    end for
    if # new constraints in last iteration is less than threshold then
        ε ← ε/2
    end if
until ε < ε_final and # new constraints in last iteration is less than threshold
```

3.2 Approximate Inference

Because the joint feature mapping $\mathbf{\Psi}$ 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.
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.

[3] If all 5 faces are present, there are 8 boundaries between them.

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 ($\boldsymbol{h} = \mathbf{GT}$)**: 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

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).

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 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

[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.

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

[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.

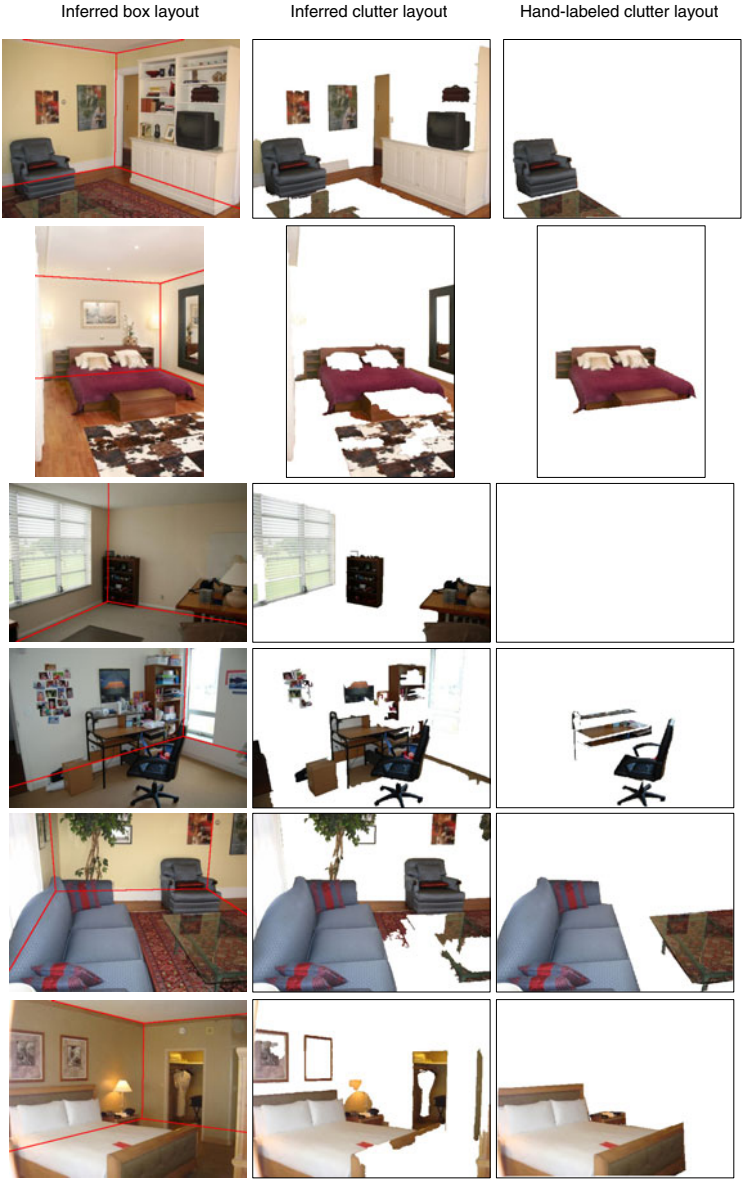


Fig. 4. Sample results for comparing the recovered clutters by our method and the hand-labeled clutters in the dataset. The 1st and 2nd column are recovered box and clutter layouts by our method. The 3rd column (right) is the hand-labeled clutter layouts. Our method usually recovers more objects as "clutter" 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.

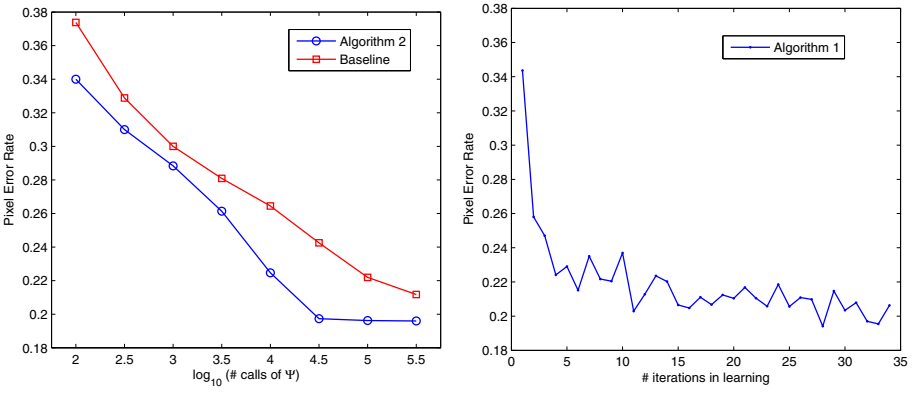


Fig. 5. Left: Comparison between the inference method described in Algorithm 2 and the baseline inference method that evaluates hypotheses independently. **Right:** Empirical convergence evaluation for the learning procedure.

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 $\mathbf{\Psi}$ 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 $\mathbf{\Psi}$.

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, 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

[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.

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 (2010) (to appear)
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)
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. In: IVC, vol. 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)

Visual Tracking Using a Pixelwise Spatiotemporal Oriented Energy Representation

Kevin J. Cannons, Jacob M. Gryn, and Richard P. Wildes

Department of Computer Science and Engineering
York University
Toronto, Ontario, Canada
{kcannons,jgryn,wildes}@cse.yorku.ca

Abstract. This paper presents a novel pixelwise representation for visual tracking that models both the spatial structure and dynamics of a target in a unified fashion. The representation is derived from spatiotemporal energy measurements that capture underlying local spacetime orientation structure at multiple scales. For interframe motion estimation, the feature representation is instantiated within a pixelwise template warping framework; thus, the spatial arrangement of the pixelwise energy measurements remains intact. The proposed target representation is extremely rich, including appearance and motion information as well as information about how these descriptors are spatially arranged. Qualitative and quantitative empirical evaluation on challenging sequences demonstrates that the resulting tracker outperforms several alternative state-of-the-art systems.

1 Introduction

Tracking of objects in image sequences is a well-studied problem in computer vision that has seen numerous advances over the past thirty years. There are several direct applications of "following a target" (e.g., surveillance and active camera systems); furthermore, many computer vision problems rely on visual trackers as an initial stage of processing (e.g., activity and object recognition). Between the direct applications of target tracking and the evolution of visual tracking into a basic stage for subsequent processing, there is no shortage of motivation for the development of robust visual trackers.

Even given this strong motivation, to date a general purpose visual tracker that operates robustly across all real-world settings has not emerged. One key challenge for visual trackers is illumination effects. Under the use of many popular representations (e.g., colour), the features' appearance changes drastically depending on the lighting conditions. A second challenge for visual trackers is clutter. As the amount of scene clutter increases, so to does the chance that the tracker will be distracted away from the true target by other "interesting" scene objects (i.e., objects with similar feature characteristics). Finally, trackers often experience errors when the target exhibits sudden changes in appearance or velocity that violate the underlying assumptions of the system's models.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 511–524, 2010.

In this work, it is proposed that the choice of representation is key to meeting the above challenges. A representation that is invariant to illumination changes will be better able to track through significant lighting effects. A feature set that provides a rich characterization will be less likely to confound the true target with other scene objects. Finally, a rich representation allows for greater tracker resilience to sudden changes in appearance or velocity because as one component of the representation experiences a fast change, other components may remain more consistent. In the current approach, a pixelwise spatiotemporal oriented energy representation is employed. This representation uniformly captures both the spatial and dynamic properties of the target for a rich characterization, with robustness to illumination and amenability to on-line updating.

Visual trackers can be coarsely divided into three general categories: (i) discrete feature trackers (ii) contour-based trackers, and (iii) region-based trackers [9]. Since the present contribution falls into the region-tracker category, only the most relevant works in this class will be reviewed. Some region trackers isolate moving regions of interest by performing background subtraction and subsequent data association between the detected foreground "blobs" [30,27,28]. Another subclass of region trackers collapses the spatial information across the target support and uses a histogram representation of the target during tracking [11,16,5,10]. The work in [10] presents the most relevant histogram tracker to the current approach because both share a similar energy-based feature set. A final subcategory of region-based trackers retains spatial organization within the tracked area by using (dense) pixelwise feature measurements. Various feature measurements have been considered [25,13,20,23]. Further, several approaches have been developed for updating/adapting the target representation on-line [19,20,23,3]. This final subcategory of region trackers is of relevance to the present work, as it maintains a representation of dense pixelwise feature measurements with a parameterized model of target motion.

Throughout all categories of trackers, a relatively under-researched topic is that of identifying an effective representation that models both the spatial and dynamic properties of a target in a uniform fashion. While some tracking-related research has sought to combine spatial and motion-based features (e.g., [7,24]), the two different classes of features are derived separately from the image sequence, which has potential for making subsequent integration challenging. A single exception is the tracker noted above that derived its features from spatiotemporal oriented energy measurements, albeit ultimately collapsing across spatial support [10], making it more susceptible to clutter (e.g., background and foreground share similar overall feature statistics, yet would be distinguished by spatial layout) and "blind" to more complex motions (e.g., rotation).

In light of previous research, the main contributions of the present paper are as follows. (i) A novel oriented energy representation that retains the spatial organization of the target is developed for visual tracking. Although similar oriented energy features have been used before in visual trackers [10] and other areas of image sequence processing (e.g., [2,15,29,26,12,31]), it appears that these features have never been deployed in a pixelwise fashion to form the

fundamental features for tracking. (ii) A method is derived for instantiating this representation within a parametric flow estimation tracking algorithm. (iii) The discriminative power of the pixelwise oriented energy representation is demonstrated via a direct comparison against other commonly-used features. (iv) The overall tracking implementation is demonstrated to perform better than several state-of-the-art algorithms on a set of challenging video sequences during extensive qualitative and quantitative comparisons.

2 Technical Approach

2.1 Features: Spatiotemporal Oriented Energies

Video sequences induce very different orientation patterns in image spacetime depending on their contents. For instance, a textured, stationary object yields a much different orientation signature than if the very same object were undergoing translational motion. An efficient framework for analyzing spatiotemporal information can be realized through the use of 3D, (x, y, t), oriented energies [2]. These energies are derived from the filter responses of orientation selective bandpass filters that are applied to the spatiotemporal volume representation of a video stream. A chief attribute of an oriented energy representation is its ability to encompass both spatial and dynamic aspects of visual spacetime, strictly through the analysis of 3D orientation. Consideration of spatial patterns (e.g., image textures) is performed when the filters are applied within the image plane. Dynamic attributes of the scene (e.g., velocity and flicker) are analyzed by filtering at orientations that extend into the temporal dimension.

The aforementioned energies are well-suited to form the feature representation in visual tracking applications for four significant reasons. (i) A rich description of the target is attained due to the fact that oriented energies encompass both target appearance and dynamics. This richness allows for a tracker that is more robust to clutter both in the form of background static structures and other moving targets in the scene. (ii) The oriented energies are robust to illumination changes. By construction, the proposed feature set provides invariance to both additive and multiplicative image intensity changes. (iii) The energies can be computed at multiple scales, allowing for a multiscale analysis of the target attributes. Finer scales provide information regarding motion of individual target parts (e.g., limbs) and detailed spatial textures (e.g., facial expressions, clothing logos). In a complementary fashion, coarser scales provide information regarding the overall target velocity and its gross shape. (iv) The representation is efficiently implemented via linear and pixelwise non-linear operations [14], with amenability to real-time realizations on GPUs [31].

The desired oriented energies are realized using broadly tuned 3D Gaussian second derivative filters, $G_2(\theta, \gamma)$, and their Hilbert transforms, $H_2(\theta, \gamma)$, where θ specifies the 3D direction of the filter axis of symmetry, and γ indicates the scale within a Gaussian pyramid [14]. To attain an initial measure of energy, the filter responses are pixelwise rectified (squared) and summed according to

$$E(\mathbf{x}; \theta, \gamma) = [G_2(\theta, \gamma) * I(\mathbf{x})]^2 + [H_2(\theta, \gamma) * I(\mathbf{x})]^2, \quad (1)$$

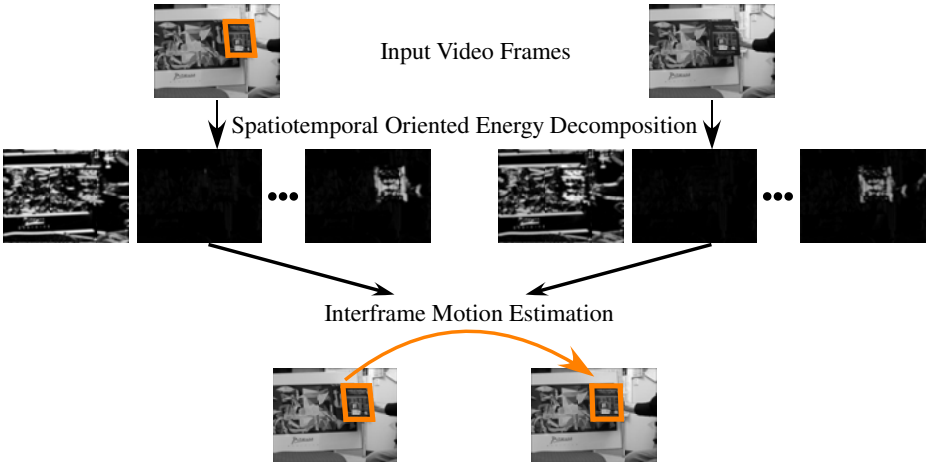


Fig. 1. Overview of visual tracking approach. (top) Two frames from a video where a book is being tracked. The left image is the first frame with a crop box defining the target's initial location. The right image is a subsequent frame where the target must be localized. (middle) Spacetime oriented filters decompose the input into a series of channels capturing spatiotemporal orientation; left-to-right the channels for each frame correspond roughly to horizontal static structure, rightward and leftward motion. (bottom) Interframe motion is computed using the oriented energy decomposition.

where $\mathbf{x} = (x, y, t)$ are spatiotemporal image coordinates, I is an image, and $*$ denotes the convolution operator. It is the bandpass nature of the G_2 and H_2 filters during the computation of (1) that leads to the energies' invariance to additive image intensity variations.

The initial definition of local energy measurements, (1), is dependent on image contrast (i.e., it will increase monotonically with contrast). To obtain a purer measure of the relative contribution of orientations irrespective of image contrast, pixelwise normalization is performed,

$$\hat{E}(\mathbf{x}; \theta, \gamma) = \frac{E(\mathbf{x}; \theta, \gamma)}{\sum_{\tilde{\gamma}} \sum_{\tilde{\theta}} E\left(\mathbf{x}; \tilde{\theta}, \tilde{\gamma}\right) + \epsilon} \quad , \tag{2}$$

where ϵ is a constant introduced as a noise floor and to avoid numerical instabilities when the overall energy content is small. Additionally, the summations, (2), consider all scales and orientations at which filtering is performed (here, the convention is to use˜for variables of summation). The representation's invariance to multiplicative intensity changes is a direct result of this normalization, (2).

2.2 Target Representation

Depending on the tracking architecture being employed, pixelwise energy measurements, (2), can be manipulated to define various target representations (e.g.,

collapsed to form an energy histogram, parameterized by orientation and scale [10]). The present approach retains the target's spatial organization by defining the representation in terms of a pixelwise template for tracking based on parametric registration of the template to the image across a sequence [21,4,6]. In particular, the template is initially defined as

$$T(x, y, \theta, \gamma) = \hat{E}(x, y, t_0; \theta, \gamma) \tag{3}$$

for energies measured at some start time, t_0, and spatial support, (x, y), over some suitably specified region. Thus, the template is indexed spatially by position, (x, y), and at each position it provides a set of $\theta \times \gamma$ energy measurements that indicate the relative presence or absence of spacetime orientations. It will be shown in Sec. 3 that retaining pixelwise organization leads to significantly better performance than collapsing over target support, as in [10].

As an illustrative example, Fig. 1 shows a sample oriented energy decomposition where a book is moving to the left in front of a cluttered background. Consideration of the first channel shows that it responds strongly to horizontal static structures both on the book and in the background. The second channel corresponds roughly to rightward motion and as such, results in negligible energy across the entire image frame. Significantly, note how the third channel tuned roughly to leftward motion yields strong energy responses on the book and small responses elsewhere, effectively differentiating between target and background.

Finally, note that the target representation, (3), is in contrast to standard template tracking-based systems that typically only utilize a single channel of intensity features during estimation [4,6]. Further, even previous approaches that have considered multiple measurements/pixel make use of only spatially derived features (e.g., [20]), which will be shown in Sec. 3 to significantly limit performance in comparison to the current approach.

2.3 Robust Motion Estimation

Tracking using a pixelwise template approach consists of matching the template, T, to the current frame of the sequence so as to estimate and compensate for the interframe motion of the target. In the present approach, both the template, T, and the image frame, I, are represented in terms of oriented energy measurements, (2). To illustrate the efficacy of this representation, an affine motion model is used to capture target interframe motion, as applicable when the target depth variation is small relative to the camera-to-target distance [25,22,23]. Further, the optical flow constraint equation (OFCE) [17] is used to formulate a match measure between features (oriented energies) that are aligned by the motion model. Under a parametric model, the OFCE can be written as

$$\nabla^{\top} \hat{E}\, \mathbf{u}(\mathbf{a}) + \hat{E}_t = 0 \ , \tag{4}$$

where $\nabla^{\top} \hat{E} = \left(\hat{E}_x, \hat{E}_y\right)$ are the first-order spatial derivatives of the image energy measurements, (2), for some specific orientation, θ, and scale, γ, $\hat{E}_t$ is

the first order temporal derivative, $\mathbf{u} = (u, v)^\top$ is the flow vector, and $\mathbf{a} = (a_0, a_1, \ldots a_5)^\top$ are the six affine motion parameters for the local region. The affine motion model is explicitly defined as

$$\mathbf{u}(x, y; \mathbf{a}) = (a_0 + a_1 x + a_2 y, a_3 + a_4 x + a_5 y)^\top . \tag{5}$$

The affine parameters, $\mathbf{a}$, are estimated by minimizing the error in the constraint equation, (4), summed over the target support. Significantly, in the present approach the target representation spans not just a single image plane, but multiple feature channels (orientations and scales) of spatiotemporal oriented energies. As a result, the error minimization is performed across the target support and over all feature channels. To measure deviation from the optical flow constraint, a robust error metric, $\rho(\eta, \sigma)$, is utilized [6]. The robust metric is beneficial for occlusion events, imprecise target delineations that include background pixels, and target motion that deviates from the affine motion model (e.g., non-rigid, articulated motion). With the above considerations in mind, the affine motion parameters, defining the interframe target motion, are taken as

$$\arg\min_{\mathbf{a}} \sum_{\tilde{\mathbf{x}}} \sum_{\tilde{\theta}} \sum_{\tilde{\gamma}} \rho \left[\nabla^\top \hat{E} \left(\tilde{x}, \tilde{y}, t; \tilde{\theta}, \tilde{\gamma} \right) \mathbf{u}(\mathbf{a}) + \hat{E}_t \left(\tilde{x}, \tilde{y}, t; \tilde{\theta}, \tilde{\gamma} \right), \sigma \right], \tag{6}$$

where summations are across target support, $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y})$, as well as all feature channel orientations, $\tilde{\theta}$, and scales, $\tilde{\gamma}$. In the present implementation, the Geman-McClure error metric [18] is utilized with σ the robust metric width, as suggested in [6]. The minimization to yield the motion estimate, (6), is performed using a gradient descent procedure [6]. To increase the capture range of the tracker, the minimization process is performed in a coarse-to-fine fashion [4,6].

The affine parameters estimated using (6) bring the target template into alignment with the closest matching local set of oriented energy features that are derived from the current image frame. At the conclusion of processing each frame, the position of the target is updated via the affine motion estimates, $\mathbf{a}$, forming a track of the target across the video sequence. After target positional updating has been completed, the next video frame is obtained and the motion estimation process between the template and the new image data is performed. This process is repeated until the end of the video is reached.

2.4 Template Adaptation

Tracking algorithms require a means of ensuring that the internal target representation (i.e., the template) remains up-to-date with the true target characteristics in the current frame, especially when tracking over long sequences. For the proposed tracker, template adaptation is necessary to ensure that changes in target appearance (e.g., target rotation, addition/removal of clothing accessories, changing facial expression) and dynamics (e.g., speeding up, slowing down, changing direction) are accurately represented by the current template. In the present implementation, a simple template update scheme is utilized that

computes a weighted combination of the aligned, optimal candidate oriented energy image features in the current frame, C^i, and the previous template

$$T^{i+1}(\mathbf{x}, \theta, \gamma) = \alpha T^i(\mathbf{x}, \theta, \gamma) + (1 - \alpha) C^i(\mathbf{x}, \theta, \gamma) \quad , \tag{7}$$

where α is a constant adaptation parameter controlling the rate of the updates (c.f., [19]). Although this update mechanism is far from the state-of-the-art in adaptation [20,23,3], the implementation achieves competitive results due to the overall strength of the pixelwise oriented energy feature set.

To summarize, Fig. 1 provides an overview of the entire system.

3 Empirical Evaluation

Three experiments were performed on the resulting system to assess its ability to track affine deformations, determine the power of the pixelwise spatiotemporal oriented energy representation, and compare its performance against alternative trackers. For all three experiments, unless otherwise stated, the following parameters were used. For the representation, 10 orientations were selected as they span the space of 3D orientations for the highest order filters that were used (i.e., H_2). The particular orientations selected were the normals to the faces of an icosahedron, as they evenly sample the sphere. Energies were computed at a single scale, corresponding to direct application of the oriented filters to the input imagery. For motion estimation, coarse-to-fine processing operated over 4 levels of a Gaussian pyramid built on top of the oriented energy measurements. Templates were hand initialized and updated with $\alpha \approx 0.9$. Video results for all experiments are available in supplemental material and online [1].

Experiment 1: Tracking affine deformations. This experiment illustrates the ability of the proposed system to estimate a wide range of affine motions when tracking a planar target (book) against a similarly complicated texture background; see Fig. 2. The target undergoes severe deformations including significant rotation, shearing, and scaling. While other feature representations (e.g., pixel intensities) also might perform well in these cases, the experiment documents that the spatiotemporal oriented energy approach, in particular, succeeds when experiencing affine deformations and that the motion estimator itself is capable of achieving excellent performance.

It is seen that the system accurately tracks the book throughout all tested cases. Performance decreases slightly when the book undergoes significant rotation. This drop is not alarming because part of the oriented energy representation encompasses the spatial orientation of the target, which is clearly changing during a rotation. Despite this "appearance change" (under the proposed feature representation), success is had for three reasons. (i) Template updates adjust the internal template representation to more accurately represent the target in the current frame. (ii) The filters used in computing the oriented energies (G_2 and H_2) are broadly tuned and allow for inexact matches. (iii) The tracker can utilize other aspects of the rich representation (e.g., motion) that remain

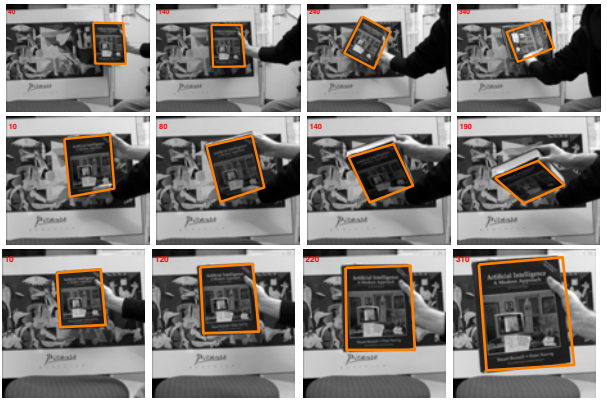


Fig. 2. Tracking through affine deformations. (row 1) Translation and subsequent rotation as the book rotates in plane over 90°. (row 2) Shearing as book rotates significantly out-of-plane. (row 3) Scaling as book moves toward camera. Orange box indicates tracked target.

relatively constant throughout the rotation. It is expected that if a more elaborate template update scheme were used, the results could be further improved.

Experiment 2: Feature set comparison. This experiment provides a comparison between the proposed spatiotemporal oriented energy features and two alternatives. In all cases, the motion estimation and template updates are identical (i.e., according to Sections 2.3 and 2.4). The single differentiating factor is the feature set. In the first system, pixelwise spatiotemporal oriented energies were used (10 orientations, as above) while the second tracker simply employed pixelwise raw image intensities. The third feature representation was purely spatial oriented energies, computed at four orientations (0°, 45°, 90°, and 135°), so as to span the space of 2D orientations for the highest order filters.

Five difficult, publicly available video sequences were used to demonstrate the points of this experiment. All five videos with ground truth can be downloaded [1,3]. The videos are documented in Table 1; results are shown in Fig. 3. To

Table 1. Experiments 2 and 3 video documentation. See Figs. 3 and 4 for images.

Occluded Face 2 [3]: Facial target. In plane target rotation. Cluttered background. Appearance change via addition of hat. Significant occlusion by book and hat.
Sylvester [23]: Hand-held stuffed animal target. Fast erratic motion, including out-of-plane rotation/shear. Illumination change across trajectory.
Tiger 2 [3]: Hand-held stuffed animal target. Small target with fast erratic motion. Cluttered background/foreground. Occlusion as target moves amongst leaves.
Ming [23]: Facial target. Variable facial expression. Fast motion. Significant illumination change across trajectory.
Pop Machines [original]: Similar appearing targets with crossing trajectories. Low quality surveillance video. Harsh lighting. Full occlusion from central pillar.

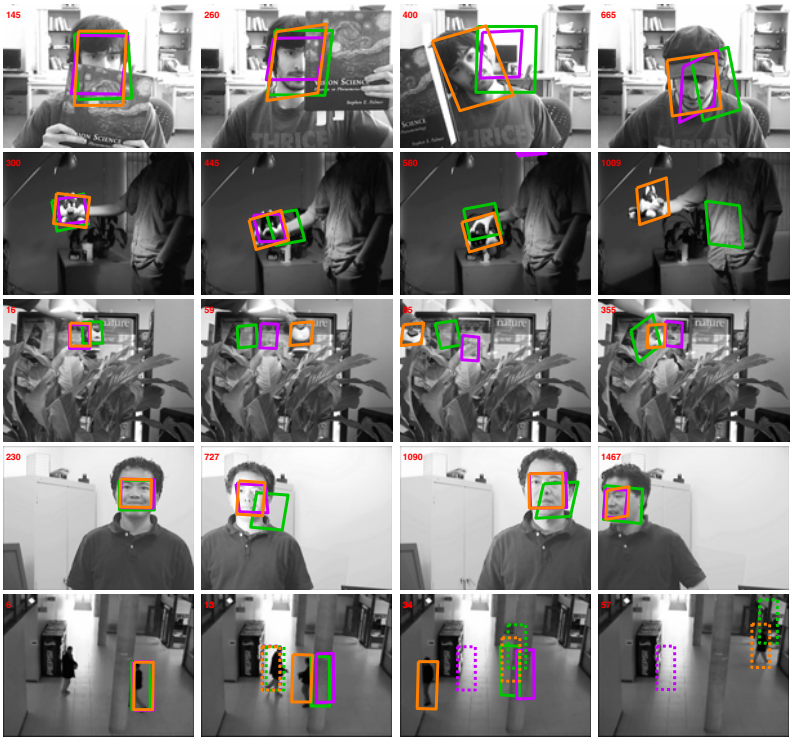


Fig. 3. Feature comparisons. Frame numbers are shown in the top left corner of each image. Top-to-bottom by row, shown are *Occluded Face 2*, *Sylvester*, *Tiger 2*, *Ming*, and *Pop Machines* of Table 1. Orange, purple, and green boxes are for spatiotemporal oriented, purely spatial oriented, and raw intensity features, resp.

ensure fair comparison with previous literature, the initial tracking boxes were set to the ground truth coordinates for the selected start frames, where available. For *Pop Machines*, trackers were initialized at the onset of each target's motion.

This second experiment clearly illustrates the fact that the choice of feature representation is critical in overcoming certain challenges in tracking including, illumination changes, clutter, appearance changes, and multiple targets with similar appearance. With reference to Fig. 3, illumination changes are problematic for the pure intensity features, as can be seen in the results from the *Ming* (Frames 727 and 1090) and *Sylvester* (Frame 445) sequences. Bandpass filtering, (1), and normalization, (2), allows both energy-based approaches to track, relatively unaffected, through these illumination variations. The *Occluded Face 2* (Frame 400) and *Tiger 2* (Frames 59 and 85) videos demonstrate that the purely spatially-based features (both raw intensity and orientation) can easily be distracted by complicated cluttered scenery, especially when the target undergoes a slight change in appearance (e.g., rotation of head, partial occlusion by foliage, motion blur). The addition of motion information in the spatiotemporal

approach provides added discriminative power to avoid being trapped by clutter. Appearance changes caused by out of plane rotations in *Sylvester* (Frame 580) and the addition of a hat in *Occluded Face 2* (Frame 665) are also problematic for the raw intensity features and the spatial oriented energies; however, motion information allows the spatiotemporal approach to succeed during the appearance changes. Notice also that when the motion changes rapidly (*Sylvester*, *Tiger 2*), the spatiotemporal approach can still maintain track, as the spatial component of the representation remains stable while the motion component adapts via update, (7). Finally, in *Pop Machines* the spatiotemporal energy representation is able to achieve success where the alternatives at least partially lose track of both targets. In this case, motion information is critical in distinguishing the targets, given their similar appearance. Success is had with the proposed approach even as the targets cross paths and with the pillar providing further occlusion.

Experiment 3: Comparison against alternative trackers. In this experiment, analyses are conducted that show the proposed spatiotemporal oriented energy tracker (**SOE**) meets or exceeds the performance of several alternative strong trackers. The trackers considered are the multiple instance learning tracker (**MIL**) [3], the incremental visual tracker (**IVT**) [23], and a tracker that uses a similar oriented energy representation, but that is spatially collapsed across target support to fit within the mean shift framework (**MS**) [10]. The parameters for the competing algorithms were assigned values that were recommended by the authors or those that provided superior results. The videos used for this experiment are the same ones used in Exp. 2.

Figure 4 shows qualitative tracker results. For *Occluded Face 2*, **SOE** and **IVT** provide very similar qualitative results; whereas, **MIL** becomes poorly localized during the later stages of the video. The collapsing of spatial arrangement information in conjunction with a loose initial target window limits the performance of **MS**, as it is distracted onto the background. Also problematic for **MIL** and **MS** is that they only estimate translation, while the target rotates. In *Sylvester*, **IVT** experiences a complete failure when the target suddenly rotates toward the camera (rapid appearance change). **MIL** follows the target throughout the entire sequence, but at times the lighting and appearance changes (caused by out-of-plane rotations) move the tracking window partially off-target. **MS** also tracks the target throughout the sequence, but allows its target window to grow gradually too large due to a relatively unstructured background and no notion of target spatial organization. **SOE** performs best due to the robustness of its features to illumination changes and their ability to capitalize on motion information when appearance varies rapidly. In *Tiger 2*, **SOE** struggles somewhat relative to **MIL**. Here, the small target combined with rapid motion makes it difficult for the employed coarse-to-fine, gradient-based motion estimator to obtain accurate updates. These challenges make this sequence favorable to trackers that make use of a "spotting approach" (e.g., **MIL**). The result for **SOE** is that it lags behind during the fastest motions; although, it "catches-up" throughout. With *Ming*, **SOE** and **IVT** provide accurate tracks that are qualitatively very similar. **MIL** cannot handle the large scale changes that the target undergoes throughout this video and as such,

often ends up only tracking a fraction of the target. **MS** again follows the target but with a tracking window that grows too large. Finally, in *Pop Machines*, since the two individuals within the scene look very similar and walk closely to one another, **MIL** has difficulty distinguishing between them. For much of the video sequence, both of the **MIL** tracking windows are following the same individual. On the other hand, **MS** and **IVT** cannot surmount the full occlusions caused by the foreground pillar. Only **SOE**'s feature representation, which encompasses target dynamics and spatial organization, is capable of distinguishing between the targets and tracking them both to the conclusion of the video.

In comparing the performance of the proposed **SOE** to the previous approach that made use of spatiotemporal oriented energy features for tracking, **MS**, the benefits of maintaining spatial organization (as provided by **SOE**, but not **MS**) are well documented. **MS** shows a tendency to drift onto non-target locations that share similar feature characteristics with the target when they are collapsed across support regions (e.g., occluding book in *Occluded Face 2*, backgrounds in *Ming* and *Sylvester*). In contrast, **SOE** does not exhibit these problems, as it maintains the spatial organization of the features via its pixelwise representation and the targets are distinguished from the non-target locations on that basis.

Figure 5 shows quantitative error plots for the trackers considered. Since **MIL** is stochastic, it was run 5 times and its errors averaged [3]. Ground truth for *Occluded Face 2*, *Sylvester* and *Tiger 2* were available previously [3]; ground truth was manually obtained for the *Ming* and *Pop Machines* videos. The plots largely corroborate the points that were observed qualitatively. For instance, the minor failure of **MIL** near the end of *Occluded Face 2* is indicated by the rapid increase in error. Also in *Occluded Face 2*, a transient increase in error (Frames 400 — 600) can be observed for **SOE** and **IVT** as they accurately track the target's rotation whereas the ground truth is provided more coarsely as target translation [3]. Similarly, the complete failure of **IVT** near frame 700 of *Sylvester* is readily seen. Also, the tendency of **SOE** to lag and recover in *Tiger 2* is captured in the up/down trace of its error curve, even as it remains generally below that of **IVT** and **MS**, albeit above **MIL**. For *Ming*, the excellent tracks provided by **SOE** and **IVT** are visible. It can also be seen that **MS** experiences a sudden failure as it partially moves off the target near the beginning of the sequence. However, **MS** eventually re-centers itself and provides centers of mass comparable to **MIL**. In the plots for *Pop Machines*, the upward ramps for **IVT**, **MIL**, and **MS** show how the errors slowly increase when the tracking windows fall off of a target as it continues to move progressively away. In contrast, **SOE** enjoys a relatively low error throughout for both targets.

To summarize the quantitative plots in Fig. 5, the center of mass pixel distance error was averaged across all frames, yielding the summary statistics in Table 2. Although the proposed **SOE** does not attain the best performance for every video, it is best in three cases (with one tie) and second best in the remaining two cases (trailing the best by only 3 pixels in one case). **IVT** also scores two top places, in one case tied with, and in the other only slightly better than **SOE**. Further, all trackers except **SOE** experience at least one complete failure where

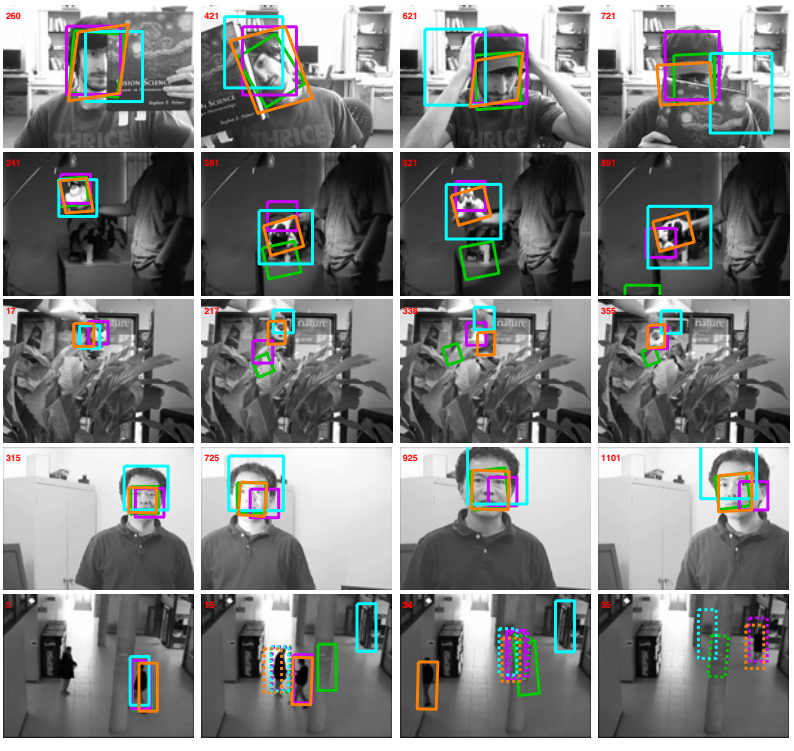


Fig. 4. Comparison to alternative trackers. Top-to-bottom as in Fig. 3. Orange, green, purple, and teal show results for proposed **SOE**, **IVT**, **MIL**, and **MS** trackers, resp.

the tracking window falls off target and does not re-establish a track before the end of the sequence. Overall, these results argue for the superior performance of **SOE** in comparison to the alternatives considered.

4 Discussion and Summary

The main contribution of the presented approach to visual tracking is the introduction of a novel target representation in terms of pixelwise spatiotemporal oriented energies. This representation uniformly captures both the spatial and temporal characteristics of a target with robustness to illumination to yield an uncommonly rich feature set, supporting tracking through appearance and illumination changes, erratic motion, complicated backgrounds and occlusions. A limitation of the current approach is its lack of explicit modeling of background motion, e.g., as encountered with an active camera, which may make the temporal components of the target features less distinctive compared to the background. In future work, various approaches can improve the system in this regard (e.g., background stabilization [8] and automatic selection of a subset of spatiotemporal features distinguishing the target vs. the background [11]).

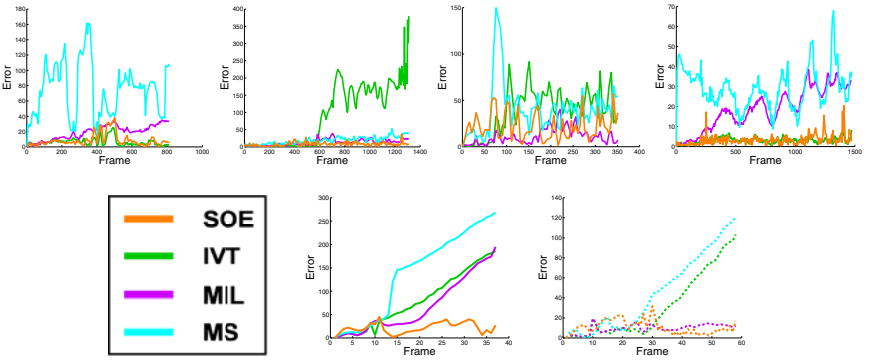


Fig. 5. Quantitative results for Experiment 3. Each plot shows the Euclidean pixel error between the ground truth and tracker center of mass. Row 1, left-to-right, results for *Occluded Face 2*, *Sylvester*, *Tiger 2*, and *Ming*. Row 2, left-to-right, *Pop Machines* target 1 (starting on right) and target 2 (starting on left).

Table 2. Summary of quantitative results. Values listed are pixel distance errors for the center of mass points. Green and red show best and second best performance, resp.

Algorithm	Occluded Face 2	Sylvester	Tiger 2	Ming	Pop Machines
SOE (proposed)	9	8	22	3	13
IVT	6	92	39	3	49
MIL	19	13	11	19	26
MS	75	19	40	29	76

The proposed approach has been realized in a software system for visual tracking that uses robust, parametric motion estimation to capture frame-to-frame target motion. Evaluation of the system on a realistic set of videos confirms the approach's ability to surmount significant tracking challenges (multiple targets, illumination and appearance variation, fast/erratic motion, clutter and occlusion) relative to a variety of alternative state-of-the-art trackers.

References

1. http://www.cse.yorku.ca/vision/research/oriented-energy-tracking
2. Adelson, E., Bergen, J.: Spatiotemporal energy models for the perception of motion. JOSA A 2(2), 284–299 (1985)
3. Babenko, B., Yang, M., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR, pp. 983–990 (2009)
4. Bergen, J., Anandan, P., Hanna, K., Hingorani, R.: Hierarchical model-based motion estimation. In: Sandini, G. (ed.) ECCV 1992. LNCS, vol. 588, pp. 237–252. Springer, Heidelberg (1992)
5. Birchfield, S., Rangarajan, S.: Spatiograms versus histograms for region-based tracking. In: CVPR, vol. 2, pp. 1158–1163 (2005)
6. Black, M., Anandan, P.: The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. CVIU 63(1), 75–104 (1996)

7. Bogomolov, Y., Dror, G., Lapchev, S., Rivlin, E., Rudzsky, M.: Classification of moving targets based on motion and appearance. In: BMVC, pp. 142–149 (2003)
8. Burt, P., Bergen, J., Hingorani, R., Kolczynski, R., Lee, W., Leung, A., Lubin, J., Shvayster, H.: Object tracking with a moving camera. In: Motion Wkshp, pp. 2–12 (1989)
9. Cannons, K.: A review of visual tracking. Technical Report CSE-2008-07, York University, Department of Computer Science and Engineering (2008)
10. Cannons, K., Wildes, R.: Spatiotemporal oriented energy features for visual tracking. In: Yagi, Y., Kang, S.B., Kweon, I.S., Zha, H. (eds.) ACCV 2007, Part I. LNCS, vol. 4843, pp. 532–543. Springer, Heidelberg (2007)
11. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. PAMI 25(5), 564–575 (2003)
12. Derpanis, K., Sizintsev, M., Cannons, K., Wildes, R.: Efficient action spotting based on a spacetime oriented structure representation. In: CVPR (2010)
13. Elgammal, A., Duraiswami, R., Davis, L.: Probabilistic tracking in joint feature-spatial spaces. In: CVPR, pp. 781–788 (2003)
14. Freeman, W., Adelson, E.: The design and use of steerable filters. PAMI 13(9), 891–906 (1991)
15. Granlund, G., Knutsson, H.: Signal Processing for Computer Vision. Kluwer, Dordrecht (1995)
16. Hager, G., Dewan, M., Stewart, C.: Multiple kernel tracking with SSD. In: CVPR, vol. 1, pp. 790–797 (2004)
17. Horn, B.: Robot Vision. MIT Press, Cambridge (1986)
18. Huber, P.: Robust Statistical Procedures. SIAM Press, Philadelphia (1977)
19. Irani, M., Rousso, B., Peleg, S.: Computing occluding and transparent motions. IJCV 12(1), 5–16 (1994)
20. Jepson, A., Fleet, D., El-Maraghi, T.: Robust on-line appearance models for visual tracking. PAMI 25(10), 1296–1311 (2003)
21. Lucas, B., Kanade, T.: An iterative image registration technique with application to stereo vision. In: DARPA IUW, pp. 121–130 (1981)
22. Meyer, F., Bouthemy, P.: Region-based tracking using affine motion models in long image sequences. CVGIP: Image Understanding 60(2), 119–140 (1994)
23. Ross, D., Lim, J., Lin, R., Yang, M.: Incremental learning for robust visual tracking. IJCV 77, 125–141 (2008)
24. Sato, K., Aggarwal, J.: Temporal spatio-velocity transformation and its applications to tracking and interaction. CVIU 96(2), 100–128 (2004)
25. Shi, J., Tomasi, C.: Good features to track. In: CVPR, pp. 593–600 (1994)
26. Sizintsev, M., Wildes, R.: Spatiotemporal stereo via spatiotemporal quadric element (stequel) matching. In: CVPR, pp. 493–500 (2009)
27. Stauffer, C., Grimson, W.: Learning patterns of activity using real-time tracking. PAMI 22(8), 747–757 (2000)
28. Takala, V., Pietikainen, M.: Multi-object tracking using color, texture and motion. In: ICCV (2007)
29. Wildes, R., Bergen, J.: Qualitative spatiotemporal analysis using an oriented energy representation. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 768–784. Springer, Heidelberg (2000)
30. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Real-time tracking of the human body. PAMI 19(7), 780–785 (1997)
31. Zaharescu, A., Wildes, R.: Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 563–576. Springer, Heidelberg (2010)

A Globally Optimal Approach for 3D Elastic Motion Estimation from Stereo Sequences

Qifan Wang, Linmi Tao, and Huijun Di

State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Computer Science and Technology, Tsinghua University, China
{wqfcr618,tao.linmi,ajon98}@gmail.com

Abstract. Dense and markerless elastic 3D motion estimation based on stereo sequences is a challenge in computer vision. Solutions based on scene flow and 3D registration are mostly restricted to simple non-rigid motions, and suffer from the error accumulation. To address this problem, this paper proposes a globally optimal approach to non-rigid motion estimation which simultaneously recovers the 3D surface as well as its non-rigid motion over time. The instantaneous surface of the object is represented as a set of points which is reconstructed from the matched stereo images, meanwhile its deformation is captured by registering the points over time under spatio-temporal constraints. A global energy is defined on the constraints of stereo, spatial smoothness and temporal continuity, which is optimized via an iterative algorithm to approximate the minimum. Our extensive experiments on real video sequences including different facial expressions, cloth flapping, flag waves, etc. proved the robustness of our method and showed the method effectively handles complex nonrigid motions.

1 Introduction

3D elastic motion estimation is one of the long lasting challenges in computer vision. The most popular approach to motion capture is to attach distinctive markers to deformable objects, and tracked in image sequences acquired by two or more calibrated video cameras. The tracked markers are then used to reconstruct the corresponding 3D motion. The limitation to these distinctive marker based technologies is that the number of distinctive markers is relatively rather sparse comparing to the number of points of a reconstructed 3D surface in [1,3].

Markerless motion capture methods based on computer vision technology offer an attractive alternative. Two main streams of researches have been implemented in the past decades. On one hand, approaches based on scene flow [6,7,8,9,11] have been proposed to independently estimate local motions between adjacent and transfer into a long trajectory. Obviously, error accumulation is the main problem of these approaches. On the other hand, registration among reconstructed surfaces is also a way to get the trajectory by tracing the motion of vertices [14,15], but the error in 3D surface reconstruction will leads to the fail of the tracking, which means the errors in spatial surface reconstruction will be accumulated in temporal tracking. As a result, both the approaches are

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 525–538, 2010.

limited to handle slow and simple non-rigid motion, in short time. In this paper, we propose a novel probabilistic framework to estimate markless non-rigid 3D motion from calibrated stereo image sequences.

1.1 Related Work

The most popular approach (such as [7,8]) on this topic obtain the scene flow to establish the 3D motion field. Scene flow was introduced by Vedula *et al.* [10] as a 3D optical field which is naturally another form of 3D displacement field. These methods recover the scene flow by coupling the optical flow estimation in both cameras with dense stereo matching between the images. Work [9,11] first compute the optical flow in each image sequence independently, then couple the flow for the 3D motion. Others such as [6,13] directly estimate both 3D shape and its motion. A variational method is proposed in [12], this work proposed one formulation that does both reconstruction and scene flow estimation. Scene flow estimation is performed by alternatively optimizing the reconstruction and the 2D motion field. An efficient approach for scene flow estimation is proposed in [8], which decouple the position and velocity estimation steps, and to estimate dense velocities using a variational approach.

Although existing scene flow algorithms have achieved exciting results, these approach suffers from two limitations since they do not exploit the redundancy of spatio-temporal information. First, scene flow methods estimate the 3D motion based on local consistency (optical flow), which restrict the algorithms in handling little deformation. Second, local motions are independently calculated between adjacent frames and then concatenated into long trajectories, leading to error accumulation over time. Recently, work by Di *et al.* [2,17] achieve groupwise shape registration on the whole image sequence which utilize a dynamic model to obtain the 2D elastic motion. These works gain some remarkable results without error accumulation.

Method based on registration among reconstructed 3D shapes is also proposed to estimate 3D motions. 3D active appearance models (AAMs) are often used for facial motion estimation [4]. Parametric models which are used to encode facial shape and appearance are fitted to several images. Most recently, Bradley *et al.* [14] deploy a camera array and multi-view reconstruction to capture the panoramic geometry of garments during human motion. Work by Furukawa *et al.* [15] uses a polyhedral mesh with fixed topology to represent the geometry of the scene. The shape deformation is captured by tracking its vertices over time with a rigid motion model and a regularized nonrigid deformation model for the whole mesh. An efficient framework for 3D deformable surface tracking is proposed in [5], this work reformulate the SOCP feasibility problem into an unconstrained quadratic optimization problem which could be solved efficiently by resolving a set of sparse linear equations.

These registration methods achieve impressive effort especially in garment motion capture. However, the reconstruction and registration process are performed separately. In this case, solving 3D shape reconstruction and motion tracking alone, ignoring their interrelationships, is rather challenging, which leads imprecise motion estimation. One way to improve these methods is to draw dependency between the reconstruction and tracking by integrate them within a unified model.

1.2 Proposed Approach and Contribution

This paper addresses elastic motion estimation from a synchronized and calibrated stereo sequences, which is treated as a joint tracking and reconstruction problem. A novel generative model - Symmetric Hidden Markov Models(SHMMs) is proposed to model the spatio-temporal information with stereo constraint of whole sequences. The main contribution of this paper is: the proposed globally optimal approach can handle fast, complex, and highly nonrigid motions without error accumulation over a large number of frames. This involves several key ingredients: (a) a generative model, which fully exploits the spatio-temporal information to simultaneously recover the 3D surface and obtain its motion over time; (b) nonrigid transformation functions, which effectively describe the elastic deformation; (c) a common spatial structure - a mean shape, which is automatically learned and utilized to establish the correspondence among the shapes in each image(details in Sec. 2).

2 Symmetric Hidden Markov Models and Problem Formulation

2.1 Basic Idea

The problem of 3D elastic motion estimation consists of two subproblems: shape reconstruction and motion tracking. On one hand, the method for 3D shape reconstruction has been fully developed (based on stereo vision), and an intuitive idea is to solve the problem of 3D motion estimation in two steps: first to reconstruct surface frame by frame and then to track points on the surface over time. On the other hand, the optical flow based methods for 2D motion tracking have matured as well. Straightforwardly, scene flow approaches coupled optical flow with disparity for 3D motion estimation between adjacent frames. Both the approaches obtained dense 3D motion, however, suffered from the accumulation of both reconstruction and tracking errors.

Inspiring by the work on Di *et al.* [2,17], we assimilate its idea of groupwise shape registration which avoids error propagation. A straightforward approach is: first independently apply [2] to each of the stereo sequences which obtains the 2D elastic motion in both views, and then reconstruct the 3D motion via stereo matching. In this way, reconstruction and registration are still performed separately which lead to imprecise result(discussed in Sec. 4.2). Our idea is to couple the two image sequences together with stereo constraint by integrating a stereo term into an unified tracking model–Symmetric Hidden Markov Models(SHMMs). This stereo term bridges the left and right sequences, through which spatio-temporal information from both sequences could pass to each other(see fig.1). In other words, our globally optimal approach fully exploits the spatio-temporal information in both views via stereo constraint, therefore draws dependency between the registration and reconstruction.

We define a 2D shape in each image which is represented as a set of sparse points, say L points. In each image, these L points are clustered from the edges by assuming they are the L centers of a Gaussian Mixture Models(GMMs) that generates the edges. Inspired by [19], a 2D mean shape is introduced into SHMMs as a common spatial structure, which is also represented by L points, and the spatial constraints associated to SHMMs are imposed by registering the mean shape to 2D shape in each image

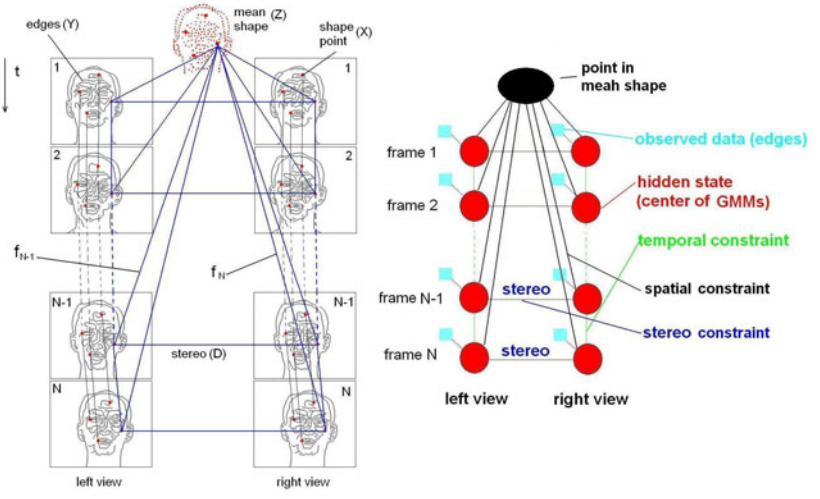


Fig. 1. Our basic idea: Symmetric Hidden Markov Model with stereo, spatial and temporal constraints

through a smooth nonrigid transformation f (see fig.1). Our tracking is performed by the registering of 2D shapes to our mean shape over time in both views. Meanwhile, the reconstruction is accomplished through the matching of the 2D shapes in stereo image pairs. In short, our task is simultaneously registering all the 2D shapes to the mean shape in both image sequences through smooth nonrigid transformations.

2.2 Symmetric Hidden Markov Models

Since we are working on the whole stereo image sequences, we first rectify the stereo images and estimate the disparity field using a stereo algorithm [18] for each stereo image pairs as the stereo constraint in SHMMs. As mentioned in sec. 2.1, the shape is represented as L points. Then these L points come into being L trajectories or Hidden Markov Models(HMMs) along the time domain in each image sequence, and therefore form L Symmetric Hidden Markov Models(SHMMs) in the stereo sequences. In our SHMMs, L points representing the shape are L hidden states; the edges extracted in each frame [16] are our observations.

Our SHMM is displayed in right fig.1, a hidden state stands for a center of GMM while the edges are treated as its observation. Each hidden state belongs to a HMM while the rest of the hidden states in the same HMM delegate the corresponding states in the temporal domain. A couple of corresponding HMMs in the stereo sequences enforced by the stereo constraints form a SHMM. The correspondence among all hidden states in one SHMM is founded by registering them to a identical point in the mean shape via transformations. Note that each SHMM essentially represents a trajectory of a 3D point. 3D shape deformation can be achieved when all L SHMMs are inferred, and the dense 3D motion is obtained through the TPS(see Sec. 2.3.2).

2.3 Problem Formulation

Before giving the formulation of our SHMMs, let us introduce the following notation for better understanding. Assume that there are N frames in each image sequence, t denotes the index of frames, $t \in \{1, 2, \ldots, N\}$, k denotes the viewpoint, $k \in \{l, r\}$; Let $I_{k,t}$ be the t^{th} image in the left($k = l$) or right($k = r$) sequence; $D = \{D_t\}$ be the disparity maps drew from the stereo image pairs; $Y_{k,t} = \{Y^i_{k,t} | i = 1, 2, \ldots, N_{k,t}\}$ be the edge point set of image $I_{k,t}$, where $N_{k,t}$ is the number of edge points.

Now we want to obtain the 3D shape motion. From the stereo vision we know that a 3D point could be reconstructed by two corresponding points in stereo images; The two corresponding points also could be seen as the projection of a 3D point onto two images. Since we have rectified the images sequences, the corresponding points in two views have the same y. So we define our 3D points(shape) as $X_t = \{[x^j_{l,t}, x^j_{r,t}, y^j_t] | j = 1, 2, \ldots, L\}$ which is clustered from edge set Y_t, where L is the number of the points in the shape. Actually $X^j_{l,t} = [x^j_{l,t}, y^j_t]$ is the 2D projection in the left image and $X^j_{r,t} = [x^j_{r,t}, y^j_t]$ is its corresponding point in the right image. Let V^j_t be the velocity of X^j_t; $Z = \{Z^j | j = 1, 2, \ldots, L\}$ be the common spatial structure - mean shape. The variable $X^j_{k,t}$ stands for the position of a hidden state in a SHMM. Each shape $X_{k,t}$ in our SHMMs matches to the mean shape Z via a smooth nonrigid transformation, $f = \{f_{k,t}\}$(see fig.1). We can simply write the equation:

$$X_{k,t} = f_{k,t}(Z) \tag{1}$$

note that X^j_t and V^j_t are 3D vector denoting the position and velocity of the points; $X^j_{k,t}$ is the 2D projection of X^j_t; Y^i_t and Z^j are 2D vector denoting the position of the edge points and mean shape; $f_{k,t}$ is $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ transformation.

Our problem is: Given disparity map sets D and edge point sets Y, we want to obtain an optimal solution of mean shape Z, nonrigid transformation f and the velocity V. Then each shape X can be directly obtained by eqn. 1.

The global energy based on our SHMMs includes four terms: a data term $E_d(Z, f; Y)$, which models the inherent relations between observation Y and hidden states X under GMM; a spatial term $E_{sp}(f)$, which enforces the smoothness of the transformations; a temporal term $E_t(Z, f, V)$, which encodes the temporal continuity by modeling the kinematics motion of points; and a stereo term $E_{st}(Z, f; D)$, which embeds the stereo constraint of the hidden states in stereo images. Note that the position of hidden states X can be represented as $f(Z)$ in terms of eqn. 1. Now we write the global energy as:

$$\begin{aligned} E(Z, f, V; D, Y) = E_d(Z, f; Y) + \alpha E_{sp}(f) \\ + \beta E_t(Z, f, V) + \gamma E_{st}(Z, f; D). \end{aligned} \tag{2}$$

α, β and γ are weight parameters that control the proportion of each term in the global energy.

Date Term. The data term encodes the relation between observation and hidden state node in all frames and it is defined as:

$$E_d(Z, f; Y) = \sum_{k,t} \sum_{i=1}^{N_{k,t}} \sum_{j=1}^{L} \left(Q(m_{k,t}^i = j) \| Y_{k,t}^i - f_{k,t}(Z^j) \|^2 / {\sigma_{k,t}^j}^2 \right) \tag{3}$$

where $\sigma_{k,t}^j$ is the variance of Gaussian distribution, $k \in \{l, r\}$, $t \in \{1, 2, \ldots, N\}$; $m_{k,t}^i$ is a discrete variable introduced to denote an index of the Gaussian mixture, which generates the i^{th} edge point $Y_{k,t}^i$ and $m_{k,t}^i \in \{1, 2, \ldots, L\}$; $Q(m_{k,t}^i = j)$ is the probability of $m_{k,t}^i = j$, where $\sum_{j=1}^{L} Q(m_{k,t}^i = j) = 1$.

The reasoning of the data term is depicted below: In a single image(say $I_{k,t}$), as mentioned in section 2.1, the observation is considered as a GMM with L centers. Then the complete density function of $Y_{k,t}^i$ is then given by:

$$P(Y_{k,t}^i | m_{k,t}^i = j, f_{k,t}(Z^j)) = \sum_{j=1}^{L} Q(m_{k,t}^i = j) N(Y_{k,t}^i, f_{k,t}(Z^j), {\sigma_{k,t}^j}^2) \tag{4}$$

where $N(X, \mu, \sigma)$ denote the Gaussian distribution on X with mean μ and variance σ. Note that $f_{k,t}(Z^j)$ essentially represent the j^{th} center $X_{k,t}^j$ of the GMMs in terms of eqn. 1. Assuming that all the edge points are independent and identically distributed (i.i.d.) with distribution eqn.4. Therefore, our data energy term eqn. 3 comes from the negative log likelihood of the joint $posterior\ \ probability$ of all the edge points. This term indicates how well the observation and the hidden states fit the GMMs.

Spatial Term. Spatial smoothness is considered in a way that each shape should be smoothly transformed from the mean shape through a smooth nonrigid transformation $f : \mathbb{R}^2 \to \mathbb{R}^2$, the smoothness of the transformation f can be measured by:

$$\|\triangle f\|^2 = \int \int \left[(\frac{\partial^2 f}{\partial x^2})^2 + 2(\frac{\partial^2 f}{\partial x \partial y})^2 + (\frac{\partial^2 f}{\partial y^2})^2 \right] \tag{5}$$

which is one of popular choices and invariant under rotations and translations, where $\triangle$ is the $Laplace\ \ operator$. When minimizing the functional related to eqn. 5, the optimal f will be the well known thin-plate spline (TPS) [20] and the measure in eqn. 5 will be the bending energy of a thin plate of infinite extent. And the $E_{sp}(f)$ term can be written as:

$$E_{sp}(f) = \sum_{k,t} \|\triangle f_{k,t}\|^2 \tag{6}$$

when considering all the transformation function f, where $k \in \{l, r\}$, $t = 1, 2, \ldots, N$. This term enforces the spatial smoothness of the transformations, which transfer the mean shape to each shape. Dense points matching between the stereo images and among the temporal images can be achieved by defining a dense mesh in the mean shape and warping it to all the images via these transformations f.

Temporal Term. In order to enforce the temporal constraint, we define the temporal term $E_t(Z, f, V)$ as:

$$E_t(Z, f, V) = \sum_{t=1}^{N} \sum_{j=1}^{L} \left(tr(S_t^j - S_{t-1}^j A) \Psi^{-1} (S_t^j - S_{t-1}^j A)^T / {\tau_t}^2 \right) \tag{7}$$

where tr stands for the trace operation of a matrix. The symbol S_t^j is a vector which combines the position of a hidden point with its velocity.

$$S_t^j \equiv [X_t^j \;\; V_t^j] \tag{8}$$

A and Ψ are both matrixes, their form will be given in the reasoning of eqn. 7. τ_t is standard deviation of a normal distribution.

Inspired by the idea of Kalman filter [21], the temporal continuity is enforced by modeling the state transition in the tracking of each HMM. The overall motion may not be easily described, but if we focus on one particular particle on the object, its motion however can be defined under kinematics. Without loss of generalization, we assume that between the $(t-1)^{th}$ and t^{th} frame the j^{th} HMM undergoes a constant acceleration that is normally distributed, with zero mean and standard deviation τ_t. From kinematics we conclude that

$$[X_t^j \;\; V_t^j] = [X_{t-1}^j \;\; V_{t-1}^j] A + [a_p \;\; a_v] G \tag{9}$$

where X_t^j is the position of the 3D point at frame t, V_t^j is its velocity. a_p and a_v are the point's accelerations of displacement and velocity, respectively. Under an ideal case of exact constant acceleration, the random acceleration variable a_p and a_v are perfectly correlated, i.e. their correlation equals to 1(a_p=a_v). But in practice, they may not be perfectly correlated, and this is why two separated accelerations a_p and a_v rather that one are used in eqn. 9. The matrixes A and G are defined as

$$A = \begin{bmatrix} 1 & 0 \\ \triangle t & 1 \end{bmatrix} \qquad G = \begin{bmatrix} \frac{\triangle t^2}{2} & 0 \\ 0 & \triangle t \end{bmatrix} \tag{10}$$

The $\triangle t$ is taken as 1 here as frame index is used as time, both A and G are 2×2 matrixes. Therefore the distribution of $[a_p \;\; a_v]G$ is a Gaussian with zero mean and covariance ${\tau_t}^2 \Psi$. Then eqn 9 can be written as

$$\begin{aligned} &P(X_t^j, V_t^j, X_{t-1}^j, V_{t-1}^j) \\ &= N(S_t^j - S_{t-1}^j A, 0, {\tau_t}^2 \Psi) \end{aligned} \tag{11}$$

where Ψ is calculated from G

The temporal energy term eqn. 7 directly comes from the negative log likelihood of *posteriori* (eqn. 11) by considering all the HMMs together. This term enforces the temporal continuity via modeling the kinematics motion of points.

Stereo Term. So far we haven't model the relationship between the stereo sequences. Since the two 2D shapes $X_{l,t}, X_{r,t}$ at any time t are the two projections of the same 3D shape X_t, their deformations are inherently related. In order to model this inherent spatio-temporal relationship, we encode stereo constraint as the stereo term $E_{st}(Z, f; D)$ to force a restriction of the motion between the corresponding stereo points.

$$E_{st}(Z, f; D) = \sum_{t=1}^{N} \sum_{j=1}^{L} \rho_d(X_{l,t}^j, X_{r,t}^j, D_t^j) \tag{12}$$

where function ρ_d punish the inconsistent matching of the hidden state between the stereo images under the disparity map, D_t^j gained from D_t is the disparity between $X_{l,t}^j$ and $X_{r,t}^j$. Here we choose a broadly used quadratic cost function, similar to the one used in [24]:

$$\rho_d(P_l, P_r, d) = ||P_l - P_r| - d|^2 \tag{13}$$

This term plays a crucial role in the model that it bridges the information of the two sequences so that shape matching between two views and registration along the time domain could be simultaneously achieved.

The combination of eqn. 3, 6, 7 and 12 is our symmetric model. We now describe an iterative optimization algorithm to minimize the global energy eqn. 2.

3 Inference and Optimization

3.1 Inference Under EM Algorithm

Directly minimizing the global energy in eqn.2 is intractable, as many terms are coupled together. Using the same divide-and-conquer fashion in [22], the optimization problems can be split into two slightly simpler sub-problems. The idea is that we first treat X and V as a whole S(defined in eqn.8) and minimize $E(Z, f, V; D, Y)$ (eqn.2) w.r.t. S, then find f and Z which achieve the optimal X(eqn.1) by solving a fitting problem. In this regard we have $X = SB$, where $B = [1\ \ 0]^T$. Then the two sub-problems are given as

$$\begin{aligned} SP1: \min_{S} & \sum_{k,t} \sum_{j=1}^{L} \sum_{i=1}^{N_{k,t}} (Q(m_{k,t}^i = j) \| Y_{k,t}^i - S_{k,t}^j B \|^2 / {\sigma_{k,t}^j}^2) \\ & + \sum_{t=1}^{N} \sum_{j=1}^{L} tr(S_t^j - S_{t-1}^j A) \Psi^{-1} (S_t^j - S_{t-1}^j A)^T / {\tau_t}^2 \\ & + \gamma \sum_{t=1}^{N} \sum_{j=1}^{L} \| \|S_{l,t}^j B - S_{r,t}^j B\| - D_t^j \|^2 \end{aligned} \tag{14}$$

$$SP2: \min_{f,Z} \sum_{k,t} \left(\alpha \| \triangle f_{k,t} \|^2 + \sum_{j=1}^{L} \| f_{k,t}(Z^j) - S_{k,t}^j B \|^2 \right) \tag{15}$$

SP1 is a symmetric trajectory tracking problem, and SP2 is a groupwise shape registration problem. The solution of these two sub-problems can be obtained by an iterative deterministic annealing algorithm under EM framework present in [17]. We refer to sec. 4.2 and 4.3 of [17] for full details on how to achieve the optimal solution of these two energies. Note that S in $SP2$ is obtained from $SP1$ in each iteration and β in eqn.2 is merged into τ_t.

3.2 Outlier and Missing Data Handling

In our SHMMs, if one hidden state $X^j_{k,t}$ is inferred inaccurately due to the outliers (noises and occlusions) or data missing, it will be temporally inconsistent with the others in the same HMM, spatially inconsistent with Z (mapping from Z to X will be non-smooth) and symmetrically inconsistent with the corresponding state in the SHMM. Although constraints of stereo, temporal continuity and spatial smoothness will help in pulling $X^j_{k,t}$ away from the outliers and missing data, too strict constraints may introduce a bias in the estimation of $X^j_{k,t}$. During the EM iteration, the temporal and stereo parameter $1/\tau_{k,t}$ and γ in eqn.14 together with spatial parameter α in eqn.15 will be decreased from an initial value to a small value. Thus towards the end, the spatiotemporal and stereo constraints will have a very tiny bias effect.

In order to account for outliers and missing data, occlusions O is first computed from disparity D using the similar principle in sec. 3.3 [7]. A hidden binary variable $w^i_{k,t}$ is further introduced so that $w^i_{k,t} = 0$ if the edge point $Y^i_{k,t}$ is an outlier, and $w^i_{k,t} = 1$ if $Y^i_{k,t}$ is generated by the GMM. As mentioned in section 2.3.1, $Q(m^i_{k,t} = j)$ is the posterior probability that $Y^i_{k,t}$ is generated by $X^j_{k,t}$, then we have $\sum_{j=1}^{L} Q(m^i_{k,t} = j) = P(w^i_{k,t} = 1|Y^i_{k,t})$, for occlusion, $P(w = 0|Y \in O) = 1$. By introducing $w^i_{k,t}$, outliers can be suppressed and missing data can be handled. The complete EM algorithm for the SHMMs is shown in Table 1. ρ is the correlation between a_p and a_v.

Table 1. Our full iterative algorithm

Initialize f,Z,Y,D
Initialize parameters α,λ,ρ
Begin: Deterministic Annealing
Calculate new $Q(m)$,σ and τ
Symmetric trajectory tracking problem: Solve $SP1$
Groupwise Shape Registration: Solve $SP2$
Update Z and f
Decrease annealing parameters
Repeat until converge
End

4 Experimental Results and Discussion

4.1 Implementation and Datasets

Given the stereo video sequences mentioned above, the edge features and disparity maps, which serve as the inputs, are obtained by using edge detection algorithm [16] for all

Table 2. Characteristics of the six datasets: N, L, E and M are the numbers of frames, shape points, average edge points and vertices on the mesh we use to reconstruct the 3D surface; w and h are the width and the height of input images in pixels; The initial parameters: ρ is fixed to 0.1; α, λ and γ are 5, 0.2 and 10, they decrease during the iterations

Data	$face1$	$face2$	$cloth$	$flag$	$sphere$	$balloon$
N	60	40	50	30	120	20
L	500	500	500	700	300	250
E	4140	4000	5210	6540	3100	2500
M	5200	5200	6600	7010	6200	3200
w	640	640	1024	1024	640	320
h	480	480	768	768	480	240

images and by applying stereo algorithm [18] to all stereo image pairs respectively. Although [18] is not the best stereo algorithms according to the Middlebury benchmarks [23], it achieves fast speed in implementation. The output of the proposed approach is a set of 4D vectors (m_x, m_y, m_z, t), denoting the 3D motion field over time. Our algorithm is implemented in MATLAB and run with almost same speed on a 2.8GHZ CPU. The speed of the algorithm depends on the number of the points L representing the shape, number of frames N and the number of input edge points. For instance, it needs about one and a half hour on a stereo sequence where L equals 500, N equals 40 and there are about 4000 edge points in each image, average 68 seconds per frame.

Six real datasets are used for the experiments: waving $flag$, flapping $cloth$, inflating $balloon$, deforming $sphere$(courtesy of J. Wang, K. Xu [25]), talking $face1$ and $face2$ with different expression. The characteristics of these datasets and the parameter values used in our experiments are given in Table. 2. The motions in $face1$ are slow, but the mouth and head motions in $face1$ are challenging. Motions are fast in $flag$ and $balloon$, but relatively simple. The $sphere$ deforms quickly and dramatically, which makes it hard to track the points on its highly deforming surface. $Cloth$ and $face2$ are quite challenging datasets involving complex motions during their deformation. In $cloth$, the textures are weak compared to the others and the motions are very fast. Moreover, there are occlusions in some part of the video due to some folds. Motions in $face2$ are relatively slow than in $cloth$, but with different facial expressions, the mouth, eye and eyebrow shapes change distinctly and irregularly which makes motion estimation difficult especially in these regions.

4.2 Results and Evaluation

Left most of Fig.2 gives some result on $face1$ from some frames in the left stereo sequences along with the estimated deformation field. The center and right most of Fig.2 demonstrates a inflating $balloon$ and a deforming $sphere$. The red points are some of the tracked particles during the inflating motion and the quick deformation. Scene flow methods which use the intensity as their observation and assume the local consistency of the motion failed on the large scale motion especially when the size or shape of the tracking object varies greatly. However, our method makes use of the

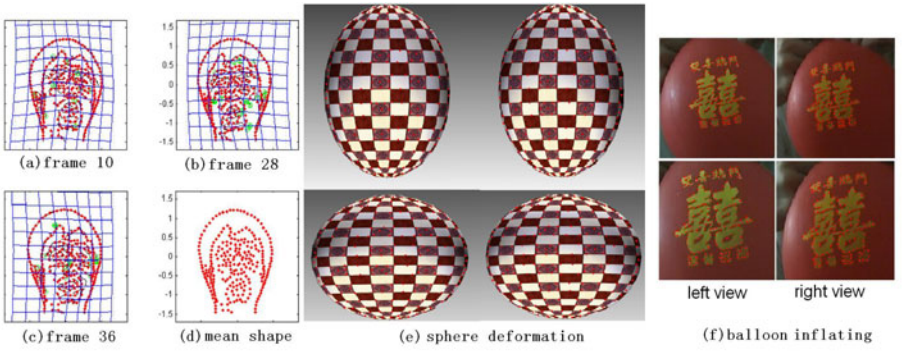


Fig. 2. Several achieved results (a)-(c): edge maps of a face sequence along with estimated deformation field; (d): learned mean shape; (e)-(f): sphere deformation and balloon inflation, the time interval between the top and the bottom stereo image pairs is 0.3s

explicit geometric information among the edges and allows us dealing with large scale motion(see Fig.4 for detail).

Left of Fig.3 shows our result on dataset $flag$, $cloth$ and $face2$, including a sample image from left sequence, the corresponding 3D surface with and without texture mapping and the estimated motion field which is rendered by line segments connecting the positions of sample points in the previous frame (red) to the current ones (green). Textures are mapped onto the reconstructed surface by averaging the back-projected textures from the corresponding images, which is a good way to visually assess the quality of the results, since textures will only look sharp and clear when the estimated shape and motion are accurate throughout the sequence. As shown by the figure, the reconstructed surfaces with sharp images looking close to the originals. Of course, there are some unshapely regions. For instance, the mouth of $face2$ and some part of the fold structure of the $cloth$. Overall however, our algorithm has been able to accurately capture the cloth's and face's complicated shape and motion. Right of Fig.3 gave two texture-mapped mean shapes computed by our method and [2] respectively. In the selected region (red rectangle) the mean shape calculated by the method [2] is much more blurred than the one calculated by our approach, which indicates the registration error of [2] is much bigger than that of our method.

Whereas there are numerous datasets with ground truth for various algorithms in computer vision, the 3D motion estimation problem is probably not mature enough to deserve a proper evaluation benchmark. In order to evaluate our method and get a comparison with other algorithms, we test our algorithm on two typical datasets with ground truth: we manually align 57 points across the stereo sequence of $face2$(use 40 frames)(landmark: 1-8: right eye, 9-16: left eye, 17-29: nose, 30-41: mouth, 42-49: right eyebrow, 50-57: left eyebrow) since face deformation is highly irregular which makes motion estimation difficult and we also use the synthetic, textured $sphere$ in [25] as our test data since its quick and dramatic deformation. We compared three methods on each dataset: (a) our globally optimal method based on SHMM; (b) tracking both sequences using our method in [2], then using stereo information to match shape; (c) pixel-wise

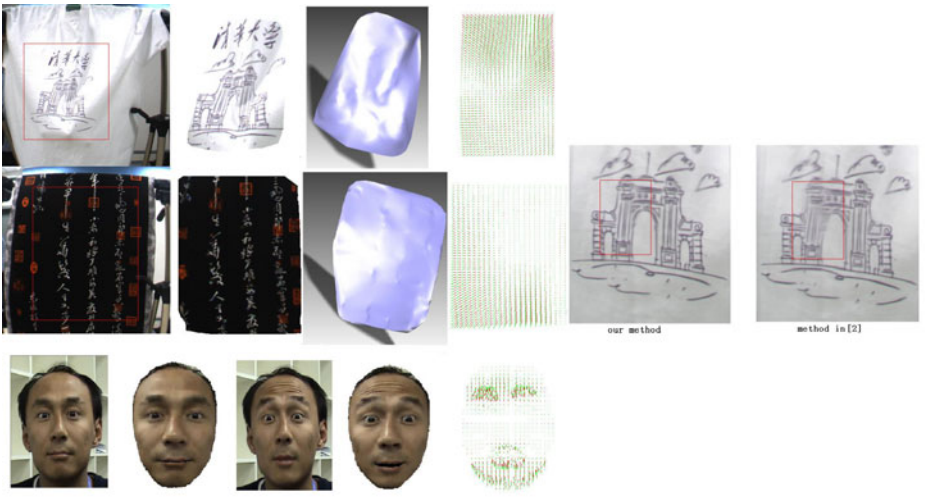


Fig. 3. From left to right in the left top and left mid($cloth$ and $flag$): an input image, a reconstructed surface with and without texture-mapping, and the corresponding motion field; $face2$ is shown at the left bottom of the figure. Our texture-mapped model is indeed very close to the corresponding input image, but there are moderate flaws in some places, in particular in the mouth region of $face2$ dueing to the complex expression and in some folded area of $cloth$. A comparison of the texture-mapped mean shape computed by our method and [2].

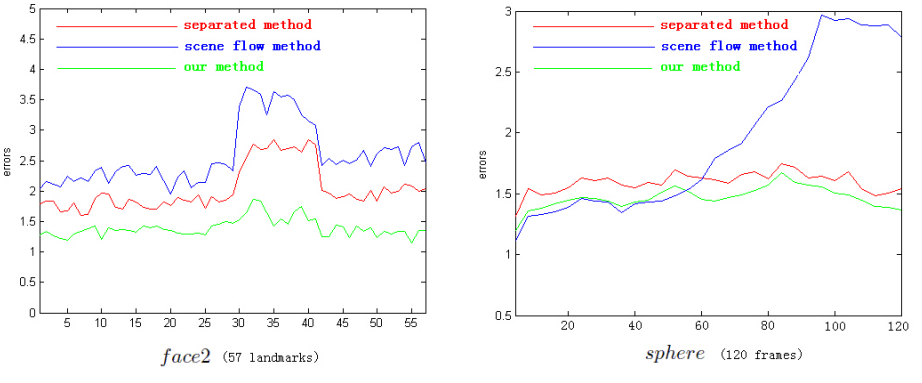


Fig. 4. Comparison of three methods on ground truth data in terms of RMS errors in pixels. Left: landmark-based RMS errors in $face2$. Right: time-based RMS errors in $sphere$.

scene flow algorithm(we simply re-implemented the method in [7]), which couples the optical flow estimation in both image sequence with dense stereo matching between the images.

Fig.4 gives the comparison of the three methods by computing the RMS errors in terms of pixels based on the ground truth data. The results showed that our approach achieved more accurate motion than the other two methods without error accumulation.

Our approach outperforms (b) because method (b) estimated the 3D motion in a separated way. In other words, (b) didn't combine the stereo and the motion together, the reconstructing and tracking were achieved separately. Although scene flow methods gained almost the same accuracy as ours in the first few frames in the $sphere$ data(see Fig.4 right), with the passage of time, the accumulated errors becomes notable since this method only computed the adjacent flow and then concatenated them into long trajectories. Suffering from error accumulation, scene flow methods could not achieve the same accuracy as our approach.

5 Conclusions

In this paper, we proposed a globally optimal approach for 3D nonrigid motion estimation from a stereo image sequences. We embed spatio-temporal information with stereo constraints of whole sequences into a novel generative model - Symmetric Hidden Markov Models. A global energy of the model is defined on the constraints of stereo, spatial smoothness and temporal continuity, which is optimized via an iterative algorithm to approximate the minimum.

Our approach is inspired by Di $et\ al.$ [2] on 2D groupwise shape registration. Experiments on real video sequences showed that our approach is able to handle fast, complex, and highly nonrigid motions without error accumulation. However, our method has two main limitations. First, our method cannot handle the topological interchange of shapes, for instance the object surface comes in contact with itself, e.g. a sleeve touches the torso in a cloth flapping. The extracted edges in these situations can tangle up or be covered with each other, which is not handled in the clustering process. Second, large occlusion leads the disparity range being comparable to the size of the objects in the image, which will not only cause many multi-resolution stereo algorithms hard to obtain accurate disparity but also make our approach difficult to matching shapes between the stereo images. A feasible scheme for dealing with this limitation is to extend our work to multi-view based motion capture [14,15].

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China under Grant Nos. 60873266 and 90820304. The authors would like to thank Kun Xu and Liang Li for their valuable suggestions and helps towards this research.

References

1. White, R., Crane, K., Forsyth, D.: Capturing and animating occluded cloth. ACM Transactions on Graphics (2007)
2. Di, H., Tao, L., Xu, G.: A Mixture of Transformed Hidden Markov Models for Elastic Motion Estimation. IEEE Trans. PAMI 31(10), 1817–1830 (2009)
3. Park, S.I., Hodgins, J.K.: Capturing and animating skin deformation in human motion. ACM ToG 25(3) (2006)

4. Koterba, S.C., Baker, S., Matthews, I., Hu, C., Xiao, J., Cohn, J., Kanade, T.: Multi-view AAM fitting and camera calibration. In: Proc. ICCV (2005)
5. Jianke, Z., Steven, C.H., Zenglin, X., Lyu, M.R.: An Effective Approach to 3D Deformable Surface Tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 766–779. Springer, Heidelberg (2008)
6. Carceroni, R.L., Kutulakos, K.N.: Multi-view scene capture by surfel sampling: From video streams to non-rigid 3D motion, shape and reflectance. IJCV (2002)
7. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: Proc. ICCV (2007)
8. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient Dense Scene Flow from Sparse or Dense Stereo Data. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 739–751. Springer, Heidelberg (2008)
9. Vedula, S., Baker, S., Kanade, T.: Image-based spatiotemporal modeling and view interpolation of dynamic events. ACM ToG (2005)
10. Vedula, S., Baker, S.: Three-dimensional scene flow. IEEE Trans. PAMI (2005)
11. Li, R., Sclaroff, S.: Multi-scale 3D scene flow from binocular stereo sequences. In: WACV/MOTION (2005)
12. Pons, J.-P., Keriven, R., Faugeras, O.: Modelling dynamic scenes by registering multi-view image sequences. In: Proc. CVPR (2005)
13. Neumann, J., Aloimonos, Y.: Spatio-temporal stereo using multi-resolution subdivision surfaces. In: IJCV (2002)
14. Bradley, D., Popa, T., Sheffer, A., Heidrich, W., Boubekeur, T.: Markerless Garment Capture. ACM Trans. on SIGGRAPH (2008)
15. Furukawa, Y., Ponce, J.: Dense 3D Motion Capture from Synchronized Video Streams. In: Proc. CVPR (2008)
16. Canny, J.: A computational approach to edge detection. PAMI (1986)
17. Di, H., Iqbal, R.N., Xu, G., Tao, L.: Groupwise shape registration on raw edge sequence via a spatio-temporal generative model. In: CVPR (2007)
18. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. IJCV (2006)
19. Chui, H., Rangarajan, A., Zhang, J., Leonard, C.: Unsupervised learning of an atlas from unlabeled point-sets. IEEE Trans. PAMI (2004)
20. Bookstein, F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Trans. PAMI (1989)
21. Forsyth, D., Ponce, J.: Chapter tracking with linear dynamic models, computer vision: A modern approach. Prentice Hall, Inc., Englewood Cliffs (2003)
22. Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. CVIU (2003)
23. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV (2002)
24. Jian, S., Heung-Yeung, S., Nanning, Z.: Stereo Matching Using Belief Propagation. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 510–524. Springer, Heidelberg (2002)
25. Wang, J., Xu, K., Zhou, K., Lin, S., Hu, S., Guo, B.: Spherical Harmonics Scaling. In: Pacific Conference on Computer Graphics and Applications (2006)

Occlusion Boundary Detection Using Pseudo-depth

Xuming He and Alan Yuille

Department of Statistics, UCLA,
8145 Math Science Building, Los Angeles, CA, USA
{hexm,yuille}@stat.ucla.edu

Abstract. We address the problem of detecting occlusion boundaries from motion sequences, which is important for motion segmentation, estimating depth order, and related tasks. Previous work by Stein and Hebert has addressed this problem and obtained good results on a benchmarked dataset using two-dimensional image cues, motion estimation, and a *global boundary model* [1]. In this paper we describe a method for detecting occlusion boundaries which uses depth cues and local segmentation cues. More specifically, we show that crude scaled estimates of depth, which we call *pseudo-depth*, can be extracted from motion sequences containing a small number of image frames using standard SVD factorization methods followed by weak smoothing using a Markov Random Field defined over super-pixels. We then train a classifier for occlusion boundaries using pseudo-depth and local static boundary cues (adding motion cues only gives slightly better results). We evaluate performance on Stein and Hebert's dataset and obtain results of similar average quality which are better in the low recall/high precision range. Note that our cues and methods are different from [1] – in particular we did not use their sophisticated global boundary model – and so we conjecture that a unified approach would yield even better results.

Keywords: Occlusion boundary detection, Depth cue, Markov Random Field.

1 Introduction

Occlusion boundary detection, which detects object boundaries that occludes background in motion sequences, is important for motion segmentation, depth order estimation, and related tasks. For example, although there has been much recent progress in estimating dense motion flow [2,3,4,5] the estimation errors typically occur at the boundaries. Recently Stein and Hebert [1] developed a method for occlusion boundary detection using machine learning methods which combines two-dimensional image cues, motion estimation, and a sophisticated global boundary model. Their method gave good quality results when evaluated on a benchmarked database.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 539–552, 2010.

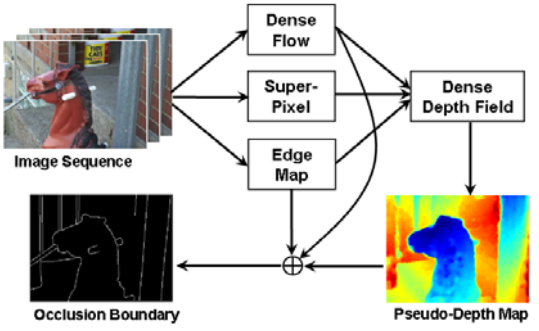


Fig. 1. Overview of our approach. Step 1 computes the dense motion flow. Step 2 estimates the pseudo-depth using SVD and weak smoothness. Step 3 trains a classifier for detecting occlusion boundaries in terms of the local edge map, the motion-flow, and the pseudo-depth. Best seen in color.

In this paper we argue that occlusion boundaries often occur at depth discontinuities and so depth cues can be used to detect them. We use a motion estimation algorithm to find the correspondence between different image frames and hence estimate crude scaled estimates of depth, which we call *pseudo-depth*. The discontinuities in pseudo-depth typically occur at occlusion boundaries and so provide detection cues which can be combined with local image segmentation cues. We note that the relationship of occlusion boundaries to depth has long been realized in the binocular stereo community [6,7] and indeed earlier work has described how it can apply to motion sequences (e.g., [8]). More recently, some motion estimation algorithms [5] do introduce some depth knowledge in an implicit form of motion smoothing.

In this paper, see Figure (1), we proceed in the following steps. *Step 1:* estimate the dense motion flow from the input image sequence. We perform this estimation using a novel algorithm (submitted elsewhere) but other motion flow algorithms that perform well on the Middlebury dataset [2] would probably be sufficient. *Step 2:* estimate *pseudo-depth* by the Singular-Value-Decomposition (SVD) technique [9,10] from the motion flow. We call this pseudo-depth since it is: (a) very noisy, (b) only known up to a scaling factor, and (c) only valid as depth if there is a single rigid motion in the image. We perform weak smoothing of the depth using a Markov Random Field (MRF) defined over super-pixels [11] (extending a method reported in [12]). We observe, see figure (1), that the pseudo-depth captures the rough depth structure and, in particular, tends to have discontinuities at occlusion boundaries. *Step 3:* train a classifier for occlusion boundaries which takes as input the motion flow, the local edge map, and the pseudo-depth map. In practice, we obtain good results using only the local edge map and the pseudo-depth map.

The contribution of this paper is to show that we can obtain results comparable to Stein and Hebert's [1] using only pseudo-depth cues and static edge cues (i.e. the Berkeley edge detector [13]). We describe the background material in

section (2) and how we estimate motion flow in section (3). Section (4) describes how we estimate pseudo-depth which is our main cue for occlusion boundary detection. Section (5) describes how we train a classifier to detect occlusion boundaries using pseudo-depth, static edge cues, and motion cues. Section (6) shows that our classifier achieves state of the art results, based on pseudo-depth and static edge cues alone, and gives comparisons to the results in [1].

2 Background

There is an enormous computer vision literature on motion estimation that can mostly be traced back to the classic work of Horn and Schunk [14]. Most of them uses a measurement term based on the optical flow constraint and smoothness terms on the velocity field to resolve the ambiguities in local measurement. The effectiveness and efficiency of algorithms for estimating velocity were improved by the use of coarse-to-fine multi-scale techniques [3] and by the introduction of robust smoothness [15] to improve performance at velocity boundaries.

Earlier researchers have discussed how motion and depth cues could be combined to detect surface discontinuities (e.g., [8]) but their technique relies on pixel-level MRF and line processes. The importance of occlusion boundaries has been realized in the binocular stereo community [7,6]. In visual motion analysis, many efforts have been made to address the problem of modeling motion boundaries in motion estimation (see [16,17] and reference therein). Some work (e.g., [18]) has attempted to estimate motion boundaries using image segmentation and explicitly modeling regions that appear or disappear due to occlusion, but they have not been systematically evaluated on benchmark datasets. Stein and Hebert [1] proposed methods for detecting occlusion boundaries but do not use explicit depth cues.

There is an extensive literature on methods to estimate depth from sequences of images [19]. This is a highly active area and there has been recent successful work on estimating three-dimensional structure from sets of photographs [20], or dense depth estimation from optical flows [21,22,23]. In this paper, our goal is only to obtain rough dense depth estimates from motion so we rely on fairly simple techniques such as the SVD factorization method [9,10] and a scaled orthographic camera model instead of the more sophisticated techniques and camera models described in those approaches.

There has also been recent work on estimating depth from single images [12,24,25], some of which explicitly address occlusion [25]. There seems to be no direct way to compare our results to theirs. But we adapt techniques used in these papers, for example performing a pre-segmentation of the image into superpixels [11] and then smoothing the depth field by defining a Markov Random Field (MRF) on superpixels [12].

3 Step 1: Motion Flow Estimation

We compute motion flow between image sequences using our own motion flow algorithm (submitted elsewhere and to be publicly available). But the rest of

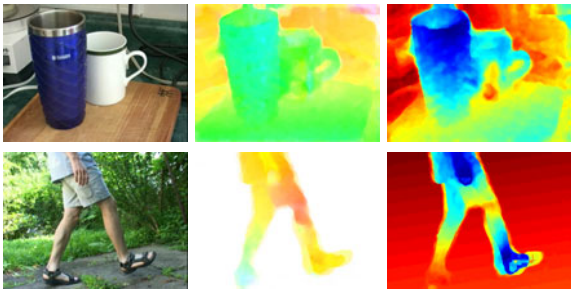


Fig. 2. Two examples of dense flow estimation (based on Middlebury flow-color coding, middle panel) and dense pseudo-depth estimation (right panel). Observe that the pseudo-depth has discontinuities at the boundary of the walking figure (lower panels) even though the images contain multiple motions. Best seen in color.

this paper does not critically depend on which motion flow algorithm is used. We obtained similar results using motion code publicly available (e.g., [5]). So we believe that motion flow results by, for example, other algorithms that perform well on the Middlebury dataset [2] may yield motion flow that is sufficiently accurate to be used as input to our approach.

More specifically, we compute dense motion flow for a sequence of three images $\{\mathbf{I}_{-m}, \mathbf{I}_0, \mathbf{I}_m\}$, where $m \geq 3$ indexes the image frame. The middle image $\mathbf{I}_0$ is the reference image for which we will evaluate the pseudo-depth and estimate the occlusion boundaries. The size of m is determined by the following considerations. We require that the images $\mathbf{I}_m, \mathbf{I}_{-m}$ must be sufficiently far apart in time to enable us to estimate the pseudo-depth reliably but they must be close enough in time to ensure that we can obtain the motion flow accurately. In practice, our default setting for the Stein and Hebert database [1] was $m = 7$ but we reduced m for short image sequences. We show two typical results of the motion flow in Figure (2)(middle panels).

We compute the motion flow $\mathbf{V}_b$ from $\mathbf{I}_0$ to $\mathbf{I}_{-m}$ (backwards) and $\mathbf{V}_f$ from $\mathbf{I}_0$ to $\mathbf{I}_m$ (forwards). These motion flows $\mathbf{V}_b$ and $\mathbf{V}_f$ are used to specify the correspondence between pixels in the three images. We represent the pixel coordinates in the reference image $\mathbf{I}_0$ by $\mathbf{x}$ with corresponding pixels coordinates $\mathbf{x} - \mathbf{V}_b$ and $\mathbf{x} + \mathbf{V}_f$ in $\mathbf{I}_{-m}$ and $\mathbf{I}_m$ respectively.

4 Step 2: Dense Pseudo-depth Estimation

We use the motion flow field to estimate dense pseudo-depth by a two-stage process. Firstly, we formulate the problem in terms of quadratic minimization which can be solved by the standard Singular Value Decomposition (SVD) approach, yielding a noisy estimation of pseudo-depth. Secondly, we obtain a smoothed estimate of pseudo-depth by decomposing the image into super-pixels [11], defining

a Markov Random Field (MRF) on the pseudo-depth for the super-pixels and imposing a weak smoothness assumption.

Our method depends on three key assumptions that: (i) occlusion (and pseudo-depth) boundaries can only occur at the boundaries of super-pixels, (ii) the pseudo-depth within each super-pixel can be modeled as planar, and (iii) the parameters of the pseudo-depth planes at neighboring super-pixels are weakly smooth (i.e. is usually very similar but can occasionally change significantly – for example, at occlusion boundaries). Figure 3 shows two examples of smoothed pseudo-depth fields.

4.1 Pseudo-depth from Motion

We use corresponding pixels $\mathbf{x} - \mathbf{V}_b$, $\mathbf{x}$, and $\mathbf{x} + \mathbf{V}_f$, supplied by the motion-flow algorithm, to estimate pseudo-depth for all pixels in the reference image $\mathbf{I}_0$. We assume that the camera geometry can be modeled as scaled-orthographic projection [19] (This is a reasonable assumption provided the the camera has the same direction of gaze in all three images).

We also assume that the motion of the viewed scene can be modeled as if it is perfectly rigid. This rigidity assumption is correct for many images in the dataset [1] but is violated for those which contain moving objects such as cats or humans, for example see Figure (2)(lower panels). Interestingly the pseudo-depth estimation results are surprisingly insensitive to these violations and, in particular, discontinuities in the pseudo-depth estimates often occur at the boundaries of these moving objects.

More formally, we assume that the pixels $\mathbf{x} = \{(x_\mu, y_\mu) : \mu \in \mathbf{L}\}$ in the reference image (where $\mathbf{L}$ is the image lattice) correspond to points $\mathbf{X} = \{(x_\mu, y_\mu, z_\mu) : \mu \in \mathbf{L}\}$ in three-dimensional space, where the $\{z_\mu : \mu \in \mathbf{L}\}$ are unknown and need to be estimated. We assume that the other two images $\mathbf{I}_{-m}$ and $\mathbf{I}_m$ are generated by these points $\mathbf{X}$ using scaled orthographic projection with unknown camera projection matrices $\mathbf{C}_{-m}, \mathbf{C}_m$. Hence the positions of these points $\mathbf{X}$ in images $\mathbf{I}_{-m}, \mathbf{I}_m$ is given by $\Pi(\mathbf{X}; \mathbf{C}_{-m})$ and $\Pi(\mathbf{X}; \mathbf{C}_m)$ respectively, where $\Pi(\mathbf{X}; \mathbf{C}) = \mathbf{C}\mathbf{X}$ is the projection.

Our task is to estimate the projection parameters $\mathbf{C}_{-m}, \mathbf{C}_m$ and the pseudo-depths $\{z_\mu\}$ so that the projections best agree with the correspondences between $\mathbf{I}_{-m}, \mathbf{I}_0, \mathbf{I}_m$ estimated by the motion flow algorithm. It can be formulated as minimizing a quadratic cost function:

$$E[\{z_\mu\}; \mathbf{C}_{\{-m,m\}}] = \sum_\mu |\mathbf{x}_\mu + \mathbf{v}_\mu^f - \Pi(\mathbf{X}_\mu; \mathbf{C}_m)|^2 + |\mathbf{x}_\mu - \mathbf{v}_\mu^b - \Pi(\mathbf{X}_\mu; \mathbf{C}_{-m})|^2. \quad (1)$$

As is well known, this minimization can be solved algebraically using singular value decomposition to estimate $\{z_\mu^*\}$ and $\mathbf{C}_{-m}^*, \mathbf{C}_m^*$ [9,10]. For the scaled orthographic approximation there is only a single ambiguity $\{z_\mu^*\} \mapsto \{\lambda z_\mu^*\}$ where λ is an unknown constant (but some estimate of λ can be made using knowledge of likely values of the camera parameters). We do not attempt to estimate λ and instead use the method described in [10] which implicitly specifies a default value for it.

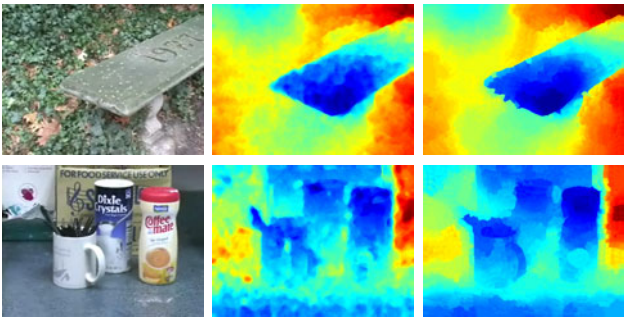


Fig. 3. Two examples of the estimated pseudo-depth field. Left panels: the reference images. Middle panels: the pseudo-depth estimated using SVD. Right panels: the pseudo-depth after weak smoothing. Best seen in color.

4.2 Weak Smoothness to Improve the Pseudo-Depth Estimates

The pseudo-depth estimates provided by SVD are noisy, particularly at places where the motion flow is noisy. We improve these estimates by weak smoothing using a Markov Random Field (MRF) model which discourages smoothing across depth discontinuities. This smoothing must be "weak" in order to avoid smoothing across the occlusion boundaries.

To define this MRF we first decompose the reference image into super-pixels using a spectral clustering method [11]. This gives roughly 1000 superpixels for the reference image (which is usually of size 240×320). We assume that each super-pixel corresponds to a planar surface in pseudo-depth. This assumption is reasonable since the size of the super-pixels is fairly small (also, by definition, the intensity properties of super-pixels are fairly uniform so it would be hard to get more precise estimates of their pseudo-depth). We also assume that neighboring super-pixels have planar surfaces which have similar orientations and pseudo-depth except at motion occlusion boundaries. This method is similar to one reported in [12] who also used a MRF defined on super-pixels for depth smoothing.

More precisely, let $\mathbf{X}_i = \{(x_{ir}, y_{ir}, z_{ir})\}$ be the set of points (indexed by r) in the i^{th} superpixel (with their pseudo-depths estimated as above). We assume a parametric planar form for each super-pixel – $a_i(x_{ir}-x_{i0})+b_i(y_{ir}-y_{i0})+z_{ir}-c_i = 0$ – and express the parameters as $\mathbf{d}_i = (a_i, b_i, c_i)$.

Next we define an MRF whose nodes are the super-pixels and whose state variables are the parameters $D = \{\mathbf{d}_i\}$ (i.e. a super-pixel i has state $\mathbf{d}_i$). The MRF has unary potential terms which relate the state $\mathbf{d}_i$ to the three-dimensional position estimates $\mathbf{X}_i$ and pairwise potential terms which encourage neighboring super-pixels to have similar values of $\mathbf{d}$, but which are robust to discontinuities (to prevent smoothing across occlusion boundaries). This gives an MRF specified by a Gibbs distribution with energy:

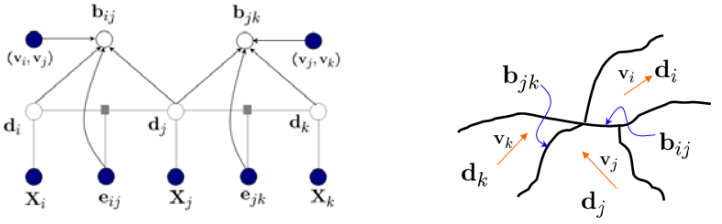


Fig. 4. Left panel: A graphical representation of the Markov random field model for the depth field and occlusion boundary detector. Circular nodes are random variables, rectangular nodes are data-dependent functions, and shaded nodes are observed. Right panel: An illustration of neighboring superpixels and the corresponding random variables defined on the superpixels and their boundaries.

$$E(D|\mathbf{X}, \mathbf{e}) = \sum_i E_u(\mathbf{d}_i, \mathbf{X}_i) + \sum_{i,j:\ j \in N(i)} E_p(\mathbf{d}_i, \mathbf{d}_j, e_{ij}), \tag{2}$$

where $\mathbf{e} = \{e_{ij}\}$ is a static edge cue [13] and e_{ij} is the static edge probability between super-pixels i and j. $N(i)$ is the neighborhood of node i. Figure 4 shows the graphical representation of our model. The unary and pairwise terms are defined as follows.

The unary term at super-pixel i depends on the 3D position estimates $\mathbf{X}_i$ at the points within the super-pixel. We use an L1 norm to penalize the deviation of these points from the plane with parameters $\mathbf{d}_i$ (L1 is chosen because of its good robustness properties) which gives:

$$E_u(\mathbf{d}_i, \mathbf{X}_i) = \alpha \sum_r ||\mathbf{c}_{ir}^T \mathbf{d}_i + z_{ir}||_{l_1} \tag{3}$$

where $\mathbf{c}_{ir} = (x_{ir} - x_{i0}, y_{ir} - y_{i0}, -1)^T$ is a constant vector for each point in the super-pixel i. The pairwise energy function also uses the L1 norm to penalize the differences between the parameters $\mathbf{d}_i$ and $\mathbf{d}_j$ at neighboring pixels, but this penalty is reduced if there is a strong static edge between the super-pixels. This gives:

$$E_p(\mathbf{d}_i, \mathbf{d}_j, e_{ij}) = (1 - \beta e_{ij})||\mathbf{d}_i - \mathbf{d}_j||_{l_1}. \tag{4}$$

where β is a coefficient modulating the strength of the edge probability e_{ij}.

4.3 Inferring Pseudo-depth Using the Weak Smoothness MRF

We now perform *inference on the MRF* to estimate the best state $\{\mathbf{d}_i^*\} = \arg\min E(D|\mathbf{X}, \mathbf{e})$ by minimizing the energy. This energy is convex so we can solve for the minimum by performing coordinate descent using Linear Programming (LP) [26] to compute each descent step. At each step, a superpixels's depth variables are updated given its neighbor information and we sequentially update

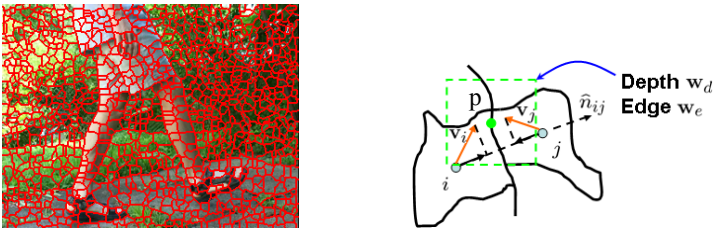


Fig. 5. Left: An instance of super-pixeled image. Right: Local cues for occlusion boundary detection. Best seen in color.

all the nodes in the field until the changes are below a fixed threshold. A few iterations, typically about 5, suffice for convergence.

We keep updating the random field for several iterations until the change is small. More specifically, we solve the following minimization problem using LP at each node:

$$\mathbf{d}_i = \arg\min_{\mathbf{d}_i} \alpha \sum_r ||\mathbf{c}_{ir}^T \mathbf{d}_i + z_{ir}||_{l_1} + \sum_{j \in N(i)} (1 - \beta e_{ij}) ||\mathbf{d}_i - \mathbf{d}_j||_{l_1} \qquad (5)$$

where $\mathbf{c}_{ir}$ are constants in E_u as in Eqn (3).

5 Step 3: Occlusion Boundary Detection

We now address the final task of occlusion detection. We use the super-pixel map of the image and attempt to classify whether the boundary between two super-pixels is, or is not, an occlusion boundary. To achieve this we train a classifier whose input is the static edge map, the estimated motion flow field, and the estimated pseudo-depth. The ground truth can be supplied from a labeled dataset, for example [1].

Three types of local cues are evaluated in our method: 1) the pseudo-depth estimates; 2) the static boundary/edge map; 3) the averaged motion estimates within each superpixel. More specifically, for a pixel $\mathbf{x}_p$ on the superpixel boundary b_{ij}, the pseudo-depth cue is a patch $w_d(p)$ of the pseudo-depth map centered at $\mathbf{x}_p$, the edge cue is a patch $w_e(p)$ of the static edge probability map at the same location (e.g. supplied by Berkeley boundary detection [13] or [27]). For the motion cue, we compute the relative motion in the following way. For superpixels i and j, their average velocity v_i and v_j are computed first. Then they are projected onto the unit vector connecting the centers of mass of the two superpixels, denoted by $\hat{n}_{ij}$. See Figure 5 for an illustration of those cues.

The output of the classifier is a probability that the pixel $\mathbf{x}_p$ is on occlusion boundary. Let $b(\mathbf{x}_p)$ denote this event. The classifier output can be written as $P(x_p|w_d(p), w_e(p), \hat{n}_{ij}^T v_i, \hat{n}_{ij}^T v_j)$. To decide if a superpixel boundary b_{ij} is an

occlusion boundary, we apply the classifier to all the points $\mathbf{x}_p \in b_{ij}$ and average the outputs of the classifier. More specifically, we set

$$P(b_{ij}|D, \mathbf{e}, v_{i,j}) = \frac{1}{|\mathbf{x}_{b_{ij}}|} \sum_{\mathbf{x}_p \in \mathbf{x}_{b_{ij}}} P(b(\mathbf{x}_p)|w_d(p), w_e(p), \mathbf{v}_{i,j}) \quad (6)$$

where $\mathbf{x}_{b_{ij}}$ is the set of pixel sites on the boundary b_{ij}, and $\mathbf{v}_{i,j}$ denotes $(\hat{n}_{ij}^T v_i, \hat{n}_{ij}^T v_j)$. The classifier can be any binary classifier with probability output. In our experiments we report results using a Multilayer Perceptron (MLP) with 15 hidden units. But we also experimented with a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel, which gave similar results. We refer to [28] for details of these techniques.

6 Experiments

6.1 Dataset and Setup

We evaluated our approach on the CMU dataset, which includes 30 image sequences of real-world scenes [1]. We use half of the sequences as training data and the other half as test data (selected randomly). We swap the training and test sets to obtain test results on all the images. In each case, we treat a third of the training data as a validation set.

We segment each reference image into approximately 1000 superpixels using spectral clustering [11]. We align the ground truth occlusion boundaries to the superpixels using the method described in [1]. This step introduces a small amount of errors, particularly on objects with detailed structure. We treat this aligned superpixel labeling as the ground truth in our evaluation (like [1]). We set the parameters of the pseudo-depth random field using the validation set and searching over the range $[0, 1]$ with step size 0.1. This yields values $\alpha = 1.0$ and $\beta = 0.9$ which we use in the following evaluation. We use a Multilayer Perceptron with 15 hidden nodes (we also tested an SVM with RBF kernel – both give similar performance), trained using all the positive examples and 1/3 of the negative examples selected randomly.

6.2 Experimental Results

The experimental results are summarized in Figure 6, which shows the precision-recall curves of our occlusion boundary detector with different settings and the state of the art. The precision (Pr) and recall (Rc) are computed in terms of the superpixel boundary segments. We also show the error rates with threshold 0.5, and F measure computed at 50% recall rate in Table 1. The F measure is defined by the harmonic mean of the precision and recall rate, i.e., $F = 2/(1/Pr + 1/Rc)$.

The left column in Figure 6 shows the detection results with the static edge cue and pseudo-depth cue. Observe that the pseudo-depth cue by itself is extremely useful at low recall values hence validating our use of it. The pseudo-depth is also complimentary to the static edge cue: it has high precision at low recall region

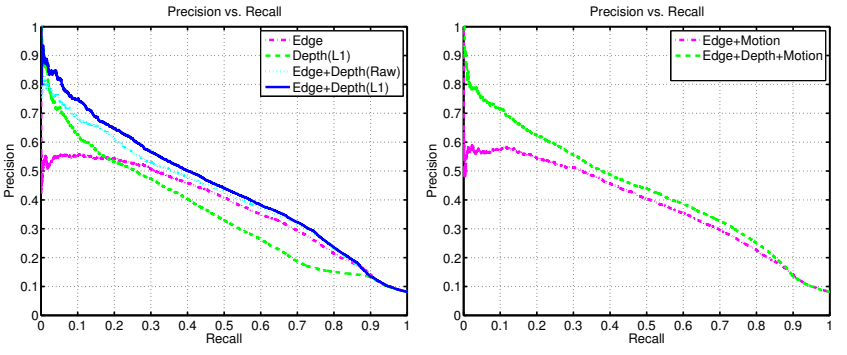


Fig. 6. Left panel: The precision-recall curve of our method with pseudo-depth and edge cues separately and in combination. Right panel: The precision-recall curve of our models including the motion cues. Observe that the motion cue does not contribute much if the other cues are used by comparing two plots. Best seen in color.

Table 1. A summary of average error rates and F measures for occlusion detection using different combinations of cues. Depth(raw) is the direct pseudo-depth output of SVD. Depth(L1) is the weakly smoothed pseudo-depth estimate. Adding motion cue to "Edge+Depth" does not provide significantly different results.

	Edge only	Depth(L1)	Edge+Motion	Edge+Depth(raw)	Edge+Depth(L1)
Error Rate	8.29	8.30	8.27	8.20	7.98
F-Score	44.93	39.78	44.73	46.01	46.89

while the static edge cue works better in the higher recall region. Combining both achieves the best performance. The smoothed pseudo-depth information provides better performance than the raw pseudo-depth map (provided by SVD), which demonstrates the benefits of weakly smoothing the pseudo-depth. The right column in the plot examines the improvements in performance due to the motion cues. We notice that adding the motion cue achieves only slightly better results by comparing two plots in Figure 6, and performance is similar to the model without the motion cue, as shown by the precision-recall curve. This might be caused by the relatively simple motion cue we use. We note that direct comparisons with the methods of Stein and Hebert's performance [1] is not completely precise because we used a different procedure for estimating super-pixels, but visual inspection suggests that their super-pixels are very similar to ours.

We can compare our results to those of Stein and Hebert's shown in Figure 7. Observe that our method gives generally comparable results to their "state-of-the-art" and outperforms them in the high precision regime. Moreover, their performance is significantly helped by their sophisticated global boundary model, which we do not use. Figure 8 illustrates the differences between our methods and the difference between the cues that are used.

Figure 9 shows a few examples of dense pseudo-depth fields and occlusion boundary detection results. We can see that our approach handles both static

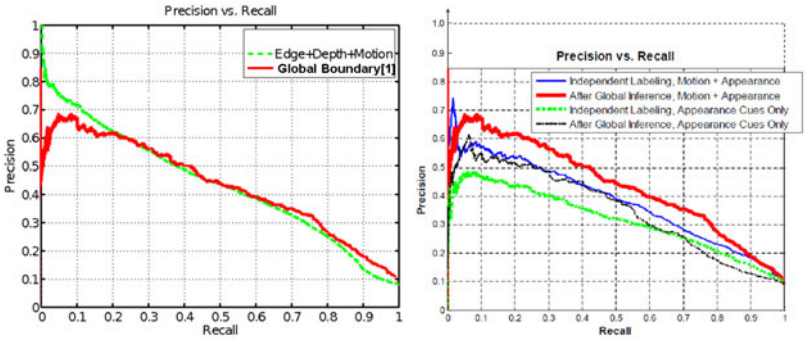


Fig. 7. Left panel: we compare our results with the best result reported by the global boundary model in [1]. Right panel: the precision-recall curve from [1] for comparison. Observe that our performance is better in the high precision regime and that their results rely heavily on their global boundary model which we do not use. Best seen in color.

	Classifier Input	Appearance	Motion	Depth	Global Boundary
Our Method	Pixel level	Edge Map	Motion Field	Pseudo-Depth	No
Stein and Hebert [1]	Super-pixel level	Super-pixel Statistics (edge, color, length and area ratio – 20 features.)	Motion Statistics	No	Boundary Consistency (Corners, T and X junctions – 50 features)

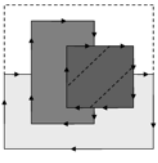


Fig. 8. Left panel: We contrast the cues used in our paper with those reported in [1]. We rely on pseudo-depth, static edges, and motion (but motion adds little). Stein and Hebert use static edges, motion cues, and a global boundary process. We classify individual pixels while they classify super-pixel boundaries directly. Right panel: This illustrates the surface consistency cues used in the global boundary process in [1] which, presumably, would improve our results.

scenes (row 1-4) and dynamic scenes (row 5-7) well. In the static scenes, the pseudo-depth estimation provides a sharp 3D boundary, which makes occlusion boundary detection much easier than using image cues only. The "pseudo-depth" in image sequences containing moving objects is also very informative for the occlusion detection task because it helps indicate depth boundaries even though the pseudo-depth values within the moving objects are highly inaccurate.

Note that the evaluation of occlusion boundaries are performed only at super-pixel boundaries [1] so there may be some errors introduced. But visual inspection shows that almost all the occlusion boundaries do occur at super-pixel boundaries.

Finally, note that our pseudo-depth smoothing method is successful at filling in small regions (super-pixels) where the motion estimation is noisy and hence the depth estimated by SVD is also noisy. But this smoothing cannot compensate for serious errors in the motion flow estimation. It will obviously not compensate

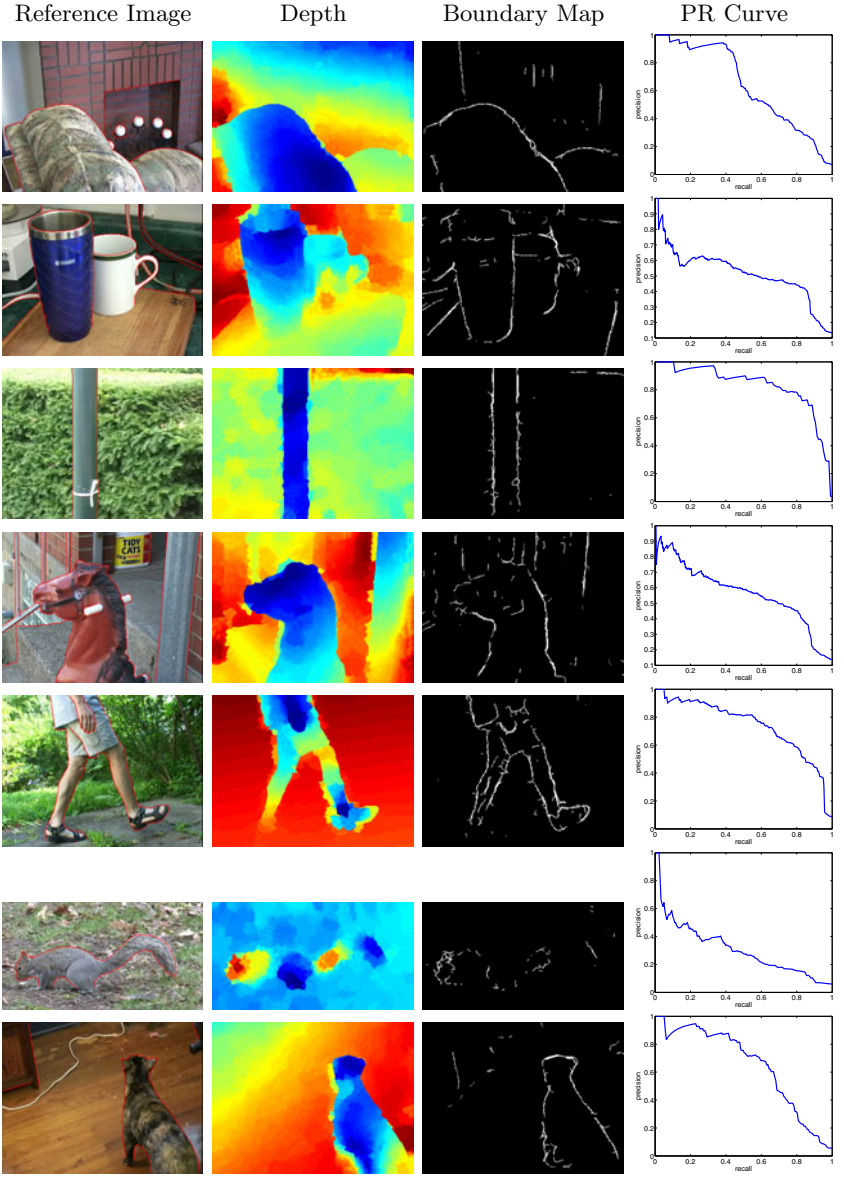


Fig. 9. Example of occlusion boundary detection results on the CMU dataset. First column: the reference frame with the ground truth boundaries overlaid. Second column: the estimated pseudo-depth field. Third column: the confidence map of the detected occlusion boundaries. Fourth column: Precision-Recall curves for the corresponding individual image sequences. Best seen in color.

for pseudo-depth estimation errors caused by moving objects, which may require some type of segmentation and separate depth estimation within each segmented region. We investigated whether the presence of moving objects could be detected from the eigenvalues of the SVD, as described in [29], but this was not successful– for almost all image sequences we typically only found two significant eigenvalues independent of whether the sequences contained moving objects. More research is needed here.

Our algorithm runs in approximately 10 seconds with all parts of the code, except the motion estimation, implemented in Matlab. So it should be straightforward to speed this up to real time performance. Stein and Hebert do not report computation time [1].

7 Conclusion and Discussion

This paper shows that crude estimation of depth, which we call pseudo-depth, provides useful cues for estimating occlusion boundaries particularly in combination with static edge cues. We show that pseudo-depth can be estimated efficiently from motion sequences and that the discontinuities in pseudo-depth occur at occlusion boundaries. We train a classifier for occlusion boundary detection with input from pseudo-depth, edge cues, and motion cues. We show that pseudo-depth and edge cues give good results comparable with the state of the art [1] when evaluated on benchmarked datasets. But that enhancing the cue set to include the motion separately does not give significant improvements. We note that the methods we use do not exploit global surface consistency constraints which are used extensively in [1] as global boundary models. Hence we conjecture that even better results can be obtained if these surface cues are combined with pseudo-depth and edge cues.

Acknowledgments. We acknowledge the support from the NSF 0736015. We appreciate conversations with Shuang Wu and George Papandreou.

References

1. Stein, A., Hebert, M.: Occlusion boundaries from motion: Low-level detection and mid-level reasoning. International Journal of Computer Vision 82, 325–357 (2009)
2. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV (2007)
3. Anandan, P.: A computational framework and an algorithm for the measurement of visual motion. International Journal of Computer Vision 2, 283–310 (1989)
4. Roth, S., Black, M.J.: On the spatial statistics of optical flow. International Journal of Computer Vision 74, 1 (2007)
5. Wedel, A., Cremers, D., Pock, T., Bischof, H.: Structure- and motion-adaptive regularization for high accuracy optic flow. In: ICCV (2009)
6. Geiger, D., Ladendorf, B., Yuille, A.: Occlusions and binocular stereo. International Journal of Computer Vision 14, 211–226 (1995)

7. Belhumeur, P., Mumford, D.: A bayesian treatment of the stereo correspondence problem using half-occluded regions, pp. 506–512 (1992)
8. Gamble, E., Geiger, D., Poggio, T., Weinshall, D.: Integration of vision modules and labeling of surface discontinuities. IEEE Transactions on Systems, Man and Cybernetics 19, 1576–1581 (1989)
9. Kontsevich, L.L., Kontsevich, M.L., Shen, A.K.: Two algorithms for reconstructing shapes. Optoelectronics, Instrumentation and Data Processing 5, 75–81 (1987)
10. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. International Journal of Computer Vision 9, 2 (1992)
11. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV (2003)
12. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3d scene structure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. 31, 824–840 (2009)
13. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. Pattern Anal. Mach. Intell. 26, 530–549 (2004)
14. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
15. Black, M., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. CVIU 63, 1 (1996)
16. Fleet, D.J., Black, M.J., Nestares, O.: Bayesian inference of visual motion boundaries. In: Exploring artificial intelligence in the new millenniu, pp. 139–173 (2003)
17. Zitnick, C.L., Jojic, N., Kang, S.B.: Consistent segmentation for optical flow estimation. In: ICCV (2005)
18. Barbu, A., Yuille, A.: Motion estimation by swendsen-wang cuts. In: CVPR (2004)
19. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
20. Agarwal, S., Snavely, N., Simon, I., Seitz, S.M., Szeliski, R.: Building rome in a day. In: ICCV (2009)
21. Xiong, Y., Shafer, S.A.: Dense structure from a dense optical flow sequence. Comput. Vis. Image Underst. 69, 222–245 (1998)
22. Ernst, F., Wilinski, P., van Overveld, C.W.A.M.: Dense structure-from-motion: An approach based on segment matching. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 217–231. Springer, Heidelberg (2002)
23. Calway, A.: Recursive estimation of 3d motion and surface structure from local affine flow parameters. IEEE Trans. Pattern Anal. Mach. Intell. 27, 562–574 (2005)
24. Russell, B.C., Torralba, A.: Building a database of 3d scenes from user annotations. In: CVPR (2009)
25. Hoiem, D., Stein, A., Efros, A., Hebert, M.: Recovering occlusion boundaries from a single image. In: ICCV (2007)
26. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
27. Konishi, S., Yuille, A., Coughlan, J., Zhu, S.C.: Statistical edge detection: learning and evaluating edge cues. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 57–74 (2003)
28. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
29. Costeira, J., Kanade, T.: A multibody factorization method for independently moving-objects 29, 159–179 (1998)

Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera

Wongun Choi and Silvio Savarese

Department of Electrical and Computer Engineering
University of Michigan, Ann Arbor, USA
{wgchoi,silvio}@umich.edu

Abstract. Tracking multiple objects is important in many application domains. We propose a novel algorithm for multi-object tracking that is capable of working under very challenging conditions such as minimal hardware equipment, uncalibrated monocular camera, occlusions and severe background clutter. To address this problem we propose a new method that jointly estimates object tracks, estimates corresponding 2D/3D temporal trajectories in the camera reference system as well as estimates the model parameters (pose, focal length, etc) within a coherent probabilistic formulation. Since our goal is to estimate stable and robust tracks that can be univocally associated to the object IDs, we propose to include in our formulation an interaction (attraction and repulsion) model that is able to model multiple 2D/3D trajectories in space-time and handle situations where objects occlude each other. We use a MCMC particle filtering algorithm for parameter inference and propose a solution that enables accurate and efficient tracking and camera model estimation. Qualitative and quantitative experimental results obtained using our own dataset and the publicly available ETH dataset shows very promising tracking and camera estimation results.

1 Introduction

Designing algorithms for tracking objects is critical in many applications such as surveillance, autonomous vehicle and robotics. In many of these applications it is desirable to detect moving humans or other targets as well as identify their spatial-temporal trajectories. Such information can enable the design of activity recognition systems for interpreting complex behaviors of individuals and their interaction with the environment. This can also provide crucial information to help an autonomous system to explore and interact with complex environments.

Among the key desiderata of an ideal tracking system researchers have identified the ability to: i) estimate stable and accurate tracks and uniquely associate them to a specific object; ii) associate tracks to 2D/3D-temporal trajectories in the 3D scene; iii) work with the minimal hardware equipment (e.g., single camera-vs-stereo cameras; no laser data); iv) work with a moving camera. Meeting all these desiderata is extremely challenging. For instance estimating stable tracks is difficult as objects are often subject to occlusions (they cross each other

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 553–567, 2010.

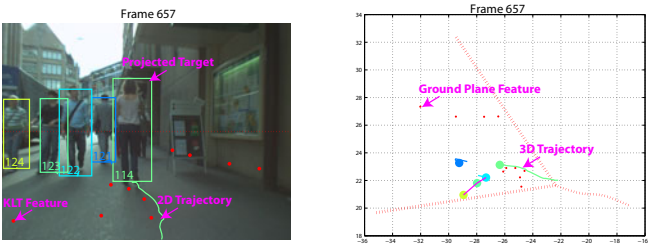


Fig. 1. Example result on ETH Seq.#2. Left: targets(colored bounding boxes) and the trajectories automatically produced by our algorithm in the image plane. Right: estimated location of targets (filled circles) on the 3D coordinate. Our algorithm not only tracks targets in the 3D coordinate system, but also estimates ego-motion of the camera by using ground plane features (red dots) and automatically discovers group of people in the scene (magenta link). Our algorithm is capable to estimate the 3D trajectories from a monocular moving camera.

in the image plane), illumination conditions can change in time, the camera motion can disturb the tracking procedure. Estimating tracks (trajectories) in the 3D world (or camera) reference system is also very hard as estimating 3D world-2D image mapping is intrinsically ambiguous if only one camera is available and camera parameters are unknown. Structure from motion (SFM) techniques are often inadequate to estimate motion parameters in that: i) the reconstruction is noisy and unreliable if small base-line is considered, ii) cluttered dynamic scene elements violate the SFM assumption of static background, iii) the procedure is computationally expensive and can be hardly implemented in real time.

Inspired by the work of [1] wherein a method for integrating multiple cues (such as odometry, depth estimation, and object detection) into a cognitive feedback loop was proposed, we present a new framework for tackling most of the issues introduced above in a coherent probabilistic framework. Specifically, our goals are to: i) solve the multi-object tracking problem by using a single uncalibrated moving camera; ii) handle complex scenes where multiple pedestrians are moving at the same time and occluding each other; iii) estimate the 2D/3D-temporal trajectories within the camera reference system.

The key contribution of our work relies on the fact that we simultaneously estimate the camera parameters (such as focal length and camera pose) and track objects (such as pedestrians) as they move in the scene (Fig.1). Tracks provide cues for estimating camera parameters by using their scale and velocity in the image plane; at the same time, camera parameters can help track objects more robustly as critical prior information becomes available. This, in turn, allows us to estimate object 3D trajectories in the camera reference system. Inspired by [2], we utilize a simplified camera model that allows to find a compact (but powerful) relationship between the variables (targets and camera parameters) via camera projection constraints. The identification of a handful of feature tracks associated with the static background allows us to add additional constraints to the camera model. Eventually, we frame our problem as a maximum-posterior

problem in the joint variable space. In order to reduce the (otherwise extremely) large search space caused by the high dimensionality of the representation, we incorporate MCMC particle filtering algorithm which finds the best explanation in sequential fashion. Notice that, unlike previous methods using MCMC, our method is the first that uses MCMC for efficiently solving the joint camera estimation and multi-target problem.

The second key contribution is that we obtain robust and stable tracking results (i.e. uniquely associate object identities to each track) by incorporating interaction models. Interaction between targets have been largely ignored in the object tracking literature, due to the high complexity in modeling moving targets and the consequential computational complexity. The independent assumption is reasonable when the scene is sparse (only few objects exists in the scene). In a crowded scene, however, the independent motion model often fails to account for the target's deviation from the prediction, e.g. if a collision is expected, targets will change their velocity and direction rapidly so as to avoid a collision. Thus, modeling interactions allows us to disambiguate occlusions between targets and better associate object labels to underlying trajectories. This capability is further enhanced by the fact that our trajectories are estimated in 3D rather than in the image plane. Our interaction models are coherently integrated in the graphical model introduced above.

We validate our theoretical results in a number of experiments using our own dataset [3] and the publicly available ETH dataset [1]. Our dataset contains several sequences of multiple humans observed under challenging conditions (moving camera with shakes, occlusions, etc). Our algorithm shows very promising results (all the results were superior than the detector baseline). Also, we evaluate our system functionalities and report detection rates by turning on and off interaction/repulsion models and camera estimation capabilities. Such results confirm our intuition that both camera models and interaction models play a critical role into the construction of stable tracks. Moreover, experiments with the ETH dataset show that our method outperforms (in terms of tracks detection accuracy) the state-of-the-art results [1]. Anecdotal examples on both datasets demonstrate that our algorithm is capable to estimate the 3D trajectories of multiple targets in the camera reference system as well as estimate the (moving) camera trajectory in a given world reference system.

2 Related Work

Multi-target tracking has received a large amount of interest among computer vision researchers. Tracking algorithms based on appearance infromation [4,5,6] are often able to track targets very well when the scene is uncluttered and the camera is static. However, as the complexity of the scene increases (complex background, crowded scene, etc), these algorithms suffer from the well-known tracker drift problem [7]. Recent improvement in object detection [8,9] makes it possible to apply detection algorithms which can effectively reduce the amount of error accumulated during tracking [10,1,11,12]. However, nearly none of these

algorithms [10,12,11] take advantage of the scene geometry or the interplay between the scene and the camera model to improve the tracking capabilities, especially for pruning out unlikely target trajectories, such as a floating human. Furthermore, methods relying on detection results [8,9] are still prone to high degree of false alarms which result in false trajectory initializations. Not only does this make the system unreliable, but also increases the complexity of the correspondence problem – a critical component of the multi-target tracking algorithm. Recently, [1] proposed a mobile tracking system that can simultaneously carry out detection and tracking by incorporating various sources of information (depth map, visual odometry). While this method demonstrates that putting together cues into a cognitive loop greatly helps reduce the false alarm rate, it leverages on the usage of stereo cameras and other specific hardware components. Multi-target tracking can be also aided by considering interaction between targets [13,14,15]. The usage of such interaction model, however, is mainly limited to the repulsion models which cannot explain the targets moving as a group. Moreover, such interaction models have never been incorporated into framework for simultaneous camera and scene estimation, and object tracking.

3 Multi-target Tracking Model

3.1 Overall Method

Given a video sequence, our goal is to jointly track multiple moving or static targets (e.g. cars, pedestrians), identify their trajectories in 3D with respect to the camera reference system, and estimate camera parameters (focal length, viewing angle, etc). We model each target as a hidden variable Z_i in 3D space whose trajectory in time must be estimated and separated from all other trajectories. We argue that estimating trajectories in 3D is more robust than estimating trajectories in the image plane because we can impose a number of priors in actual 3D space as we shall see next.

Such trajectories in 3D are estimated by measuring their projections onto 2D image plane which represent our observation variables X_i (Fig.2). Given the observations, tracks Z_i in 3D are estimated by jointly searching the most plausible explanation for both camera and all the existing targets' states using the projection characterized by the camera model (Sec.3.3). Clearly the projection from Z_i to observation X_i is a function of camera parameters. Thus, we introduce a simplified camera model (Sec.3.3) which allows us to reduce the number of parameters that are required to be estimated. We assume rough initial camera parameters are given and can be better estimated by the detected targets in the image plane. All camera parameters at time t are collected in the variable Θ_t. Moreover, as an important contribution of our work, we do not assume that targets are moving independently but their motion may be interrelated. Thus we introduce an interaction model which allows us to better estimate the states of all target. Our interaction model is composed of repulsion and attraction model (Sec.3.6). We assume i) targets cannot take the same location in the 3D coordinate and cannot collide with each other (*repulsion model*), and ii) targets that

have moved in a coherent fashion (as a group) up to time t are likely to move as a group after time t as well (*attraction model*).

In our work, we assume that the following information can be extracted from the video sequence: i) Visible targets' location and bounding box can be detected at each frame with some number of false alarms. Target detections are used to initiate tracks and gather evidence for existing targets. We use the state-of-the-art object detector [9] for detecting targets (sec.3.2). ii) Rough trajectories in the image plane are available. This additional piece of information is used as a complementary cue to better locate targets in the image plane and it is useful when the target is not properly detected by the detector. We use mean-shift algorithm [4] to obtain them (sec.3.2). iii) Feature points from static background can also be identified and tracked. Background features help the algorithm estimate the camera parameters' variations in time. For this task, KLT tracker [16] is incorporated in our algorithm. Given above cues, the targets are automatically identified/tracked/terminated using our coherent multi-target model.

3.2 Track Initiation, Termination and Correspondence

Target initiation and termination. As detection results are given by the detector, our multi-target tracking algorithm automatically initiates targets. If there exists a detection that is not matching any track, the algorithm initiates a target hypothesis. If enough matching detections for the hypothesis are found in N_i consecutive frames, the algorithm will recognize the hypothesis as a valid track and begins tracking the target. Conversely, if no enough detections are found for the same target within N_t consecutive frames, the track is automatically terminated.

Correspondence. Target correspondence is a very challenging problem by itself. For simplicity, we use the Hungarian algorithm [17] which is based on the overlap ratio between existing targets and detections. We employ two independent sources of information to solve the correspondence problem: affinity matrices of prediction and appearance tracking. The first one is constructed using the image plane prediction of i^{th} target $\hat{X}_{it} = E[X_{it}|Z_{i(t-1)}, \Theta_{t-1}]$ in time t, where t indicates the time dependency at instant (time stamp) t. By computing the negative log of pairwise overlap ratio between the predictions $\hat{X}_{it}$ and detections X_{jt}, $A(X_{it}, X_{jt}) = -log(\frac{Intersection(X_{it}, X_{jt})}{Union(X_{it}, X_{jt})})$, we construct a pairwise affinity matrix between detections and predictions. The second one leverages on the mean-shift tracker [4] (this cue is also used in the estimation). When a new target hypothesis is created, an individual mean-shift tracker is assigned to each target and applied to each frame until the target tracking is terminated. The appearance model (color histogram) is updated only when there is a supporting (matching) detection to avoid tracker-drift. Similarly to the prediction-detection affinity matrix, we compute another affinity matrix between mean-shift output Y_{it} and detections. Given the two affinity matrices, we sum the two matrices to calculate the final matrix which will be the input of Hungarian algorithm. In following sections, we assume the correspondence is given by this algorithm, so Z_{it} and its observation X_{it} are assumed to be matched.

3.3 Camera Model and KLT Features

Camera Model. Due to the inherent uncertainty in the camera projection matrix, it is very challenging to infer the exact location of an object in 3D given image plane location and camera parameters. To mitigate this problem, we set a number of assumptions on the underlying geometry and camera configuration similarly to [2]. We additionally assume that the camera follows forward motion only. With these assumptions, the camera parameters can be represented by following variables: focal length f_θ, height h_θ, horizontal center point u_θ, horizon position v_θ, panning angle ϕ_θ, absolute velocity r_θ, and 3D location (x_θ, z_θ) with respect to the reference system associated to the initial frame. Thus, the projection function f_P can be defined

$$X = f_P(\hat{Z};\Theta) = \begin{bmatrix} \frac{f_\theta x_z}{z_z} + u_\theta \\ \frac{f_\theta h_\theta}{z_z} + v_\theta \\ \frac{f_\theta h_z}{z_z} \end{bmatrix}, \quad \hat{Z} = f_P^{-1}(X;\Theta) = \begin{bmatrix} \frac{h_\theta(u_x - u_\theta)}{v_x - v_\theta} \\ \frac{f_\theta h_\theta}{v_x - v_\theta} \\ \frac{h_x h_\theta}{v_x - v_\theta} \end{bmatrix}, \quad Z = \begin{bmatrix} R(\phi_\theta) & 0 \\ 0 & 1 \end{bmatrix} \hat{Z} + \begin{bmatrix} x_\theta \\ z_\theta \\ 0 \end{bmatrix} \tag{1}$$

where $X = [u_X, v_X, h_X]^T$ and $Z = [x_Z, z_Z, h_Z]^T$; u_X, v_X, and h_X are the (bottom) center point location and height of an observation in the image plane respectively; x_Z, z_Z, and h_Z are the location and the height of the object in world coordinate. Here $\hat{Z}$ denotes the location of the target in current camera coordinate system, and Z denotes the state of the target in global reference system.

KLT for Camera Motion. In order to track multiple targets reliably, it is crucial to get a good estimate of the camera's extrinsic parameters (panning, location, and velocity). At that end, we use KLT features [16] as additional observations. Suppose we can extract feature points τ_t which are lying on the ground plane. Then by applying the inverse projection f_P^{-1} and forward projection f_P on τ_{t-1} with camera parameters in each time frame Θ_{t-1}, Θ_t, we can obtain the expected location of $\hat{\tau}_t$. By comparing the difference between τ_t and $\hat{\tau}_t$, we can infer the amount of camera's motion in the time. As we will show in the experimental section, this feature improves the tracking performance significantly. This is inspired by SLAM procedures such as [18,19].

3.4 Target Class Model

We use state-of-the-art object detector [9] for detecting targets. Despite its excellent performance, [9] still yields false detections in the challenging experimental setting we work with. In order to differentiate such false detections, we introduce one more multinomial hidden variable c in the target states Z_t, which indicates the object class of the target being tracked. If one target's c variable is set to be 0, then the target is not a valid object and thus can be removed. To guide the algorithm estimate the category of the class, we assign a height prior to each variable describing an object class. This follows a normal distribution with a particular mean and variance. Non-object class are described by a uniform distribution.

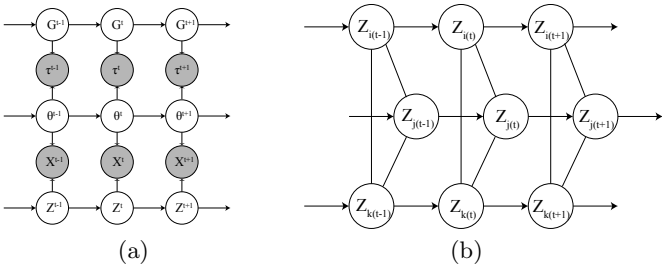


Fig. 2. Graphical model describing underlying model for multi-object tracking. Shaded nodes represent observable variables and empty nodes shows the hidden variables. Panel (a) shows overall relationship between observations X, Y, τ, camera parameters Θ, ground features' state G and targets' states Z. Here, we dropped the mean-shift observation Y on the graph to avoid clutter. Panel (b) shows the interaction between targets (i, j, k). The interaction is modelled by the undirected edges between targets.

Thus, if any observation yields a very unlikely large or small height for a certain target class (such as 1 meter for humans), than the algorithm will automatically reject out this observation. Not only does this help the algorithm to reduce the number of false alarms, but it also helps the camera parameter estimation to be more robust since it essentially rejects out outliers in the estimation process.

3.5 Sequential Tracking Model with Independent Assumption

In this section, we discuss in details the probabilistic relationship between the hidden states $\Omega_{t-1} = [Z_{t-1}, \Theta_{t-1}, G_{t-1}]$ and all the observations $\chi_t = [X_t, Y_t, \tau_t]$. Given the evidences χ_t and the estimates Ω_{t-1} at a previous time stamp, we can compute the posterior distribution $P(\Omega_t|\chi^t)$. Here, we use superscripts for denoting all the history up to time t and subscripts for current time t variables. Notice that Z_t, G_t, X_t, Y_t, and τ_t collects variables for each individual target state, ground feature state, target observation, and feature observation respectively. Following the basic Bayesian sequential model, the posterior distribution $P(\Omega_t|\chi^t)$ can be factorized as follows :

$$P(\Omega_t|\chi^t) \propto P(\Omega_t, \chi_t|\chi^{t-1}) = P(\chi_t|\Omega_t) \int P(\Omega_t|\Omega_{t-1}) P(\Omega_{t-1}|\chi^{t-1}) d\Omega_{t-1} \quad (2)$$

Here, the first term $P(\chi_t|\Omega_t)$ represents the *observation model* and the term $P(\Omega_t|\Omega_{t-1})$ explains the *motion model*. Based on the conditional independence assumption represented in Fig.2, each term can be further factorized into :

$$P(\chi_t|\Omega_t) = P(X_t, Y_t|Z_t, \Theta_t) P(\tau_t|G_t, \Theta_t) \quad (3)$$

$$P(\Omega_t|\Omega_{t-1}) = P(Z_t|Z_{t-1}) P(\Theta_t|\Theta_{t-1}) P(G_t|G_{t-1}) \quad (4)$$

Target Model. The targets' state $Z_t = \{Z_{it}\}_{i=1}^N$ is composed of 6 variables, (x, z) 3D location, (v^x, v^z) velocity, h height, and c class indicator. Given this

parameterization, we formulate the motion model for targets as $P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_{it}|Z_{i(t-1)})$ where

$$P(Z_{it}|Z_{i(t-1)}) \propto P(m_{it}|m_{i(t-1)})P(h_{it}|h_{i(t-1)})P(c_{it}|c_{i(t-1)})P(h_{it}|c_{it}) \quad (5)$$

Here, the variables x, z, v^x, v^z were substituted with m to make the notation more compact. The first term $P(m_{it}|m_{i(t-1)})$ is modeled as a simple first order linear dynamic motion model with an additive gaussian noise. The second term $P(h_{it}|h_{i(t-1)})$ is modeled as $P(h_{it}|h_{i(t-1)}) \sim N(h_{i(t-1)}, \sigma_h)$ to allow some degree of variation in height. $P(c_{it}|c_{i(t-1)})$ is modeled as a indicator function $I(c_{it} = c_{i(t-1)})$, since we do not allow the target's class to be changing in time. The targets' height prior, $P(h_{it}|c_{it})$, is represented either as a normal distribution with mean and standard deviation (h_{c_k}, σ_{c_k}) when $c = k$ or as a uniform distribution p_{c_0} when $c = 0$ (no object). In our MCMC particle filter implementation, this uniform-gaussian mixture formulation plays an "outliers-rejection" role similar to RANSAC, since it will "push out" targets from class k which are not consistent with the "consensus" to maximize the posterior distribution. Observations are modeled using the forward projection f_P: $X_{it} = f_P(Z_{it}, \Theta_t) + W$, where W is gaussian noise. Similarly, we assume that mean shift tracker can be modelled as $Y_{it} = f_P(Z_{it}, \Theta_t) + V$, again V is gaussian noise.

Ground feature Model. As stated in Sec.3.3, we use KLT tracker to track stationary features on the ground so as to get a robust estimate of the camera motion. This can be achieved by introducing the hidden state G_{it} which captures the true location of a ground feature in 3D. Let τ_{it} be the ground feature tracked in the image plane at time t, and $\hat{\tau}_{it}$ the projection of G_{it} into the image plane at t. This indicates the expected location of the feature G_{it} at t. G_{it} is composed of three variables x, z, α (its 3D location and a binary indicator variable, such variable encodes whether the feature is static and lies on the ground or not) and τ_{it} have two variables u, v (its location in the image plane). Assuming the ground plane features are static, the motion model of G_{it} will have a simple form of indicator function, $P(G_{it}|G_{i(t-1)}) = I(G_{it} = G_{i(t-1)})$.

The relationship between the state and observation $(P(\tau_t|G_t, \Theta_t))$ can be modelled using the camera projection function f_P if the feature is truely static and lying on the ground plane ($\alpha = 1$). However, if either the feature is moving or the feature is not on the ground plane ($\alpha = 0$), the projection function f_P does not model the correct relationship between τ_{it} and G_{it}. Thus, the observation process is modeled as $P(\tau_t|G_t, \Theta_t) \sim N(f_P(G_{it}, \Theta_t), \Sigma_G)$ if α_i is 1, otherwise $P(\tau_t|G_t, \Theta_t) \sim unif(p_G)$. Similar to the class variable in target model, those features that are not consistent with the majority of other features will be automatically filtered out.

Camera Model. In order to deal with camera motion, we also model camera motion parameters. Note that the camera parameters are coupled with the target and feature observations and cannot be directly observed. The temporal relationship between camera parameters is simply represented as a linear dynamic model $x_{t\theta} = x_{(t-1)\theta} - r_{(t-1)\theta} * sin(\phi_{(t-1)\theta}) * dt$ and $z_{t\theta} = z_{(t-1)\theta} + r_{(t-1)\theta} * cos(\phi_{(t-1)\theta}) * dt$

(we defined the positive value of ϕ for the left direction so there appears minus sign on x_t). We inject uncertainty in the velocity parameter by adding gaussian noise. The uncertainty of the other camera parameters ($f_\theta, h_\theta, u_\theta, v_\theta$ and ϕ_θ) are just modeled as additive gaussian noise.

3.6 From Independent to Joint Target Model

In real world crowded scenes, targets rarely move independently from each other. Targets rarely occupy the same physical space (*repulsion model*). Moreover, once human targets form a group, they typically tend to move together in subsequent time frames (*group model*). In this work, we employ two interaction models between targets (repulsion and group model) to aid the tracking alogrithm. However, since these two interactions cannot occur at the same time, we introduce a hidden variable β_{ijt} that lets us select the appropriate interaction model (mode variable).

The interaction models are modeled as pairwise potentials between current targets' states, thus forming a Markov Random Field as shown on fig.2. Thus the targets' motion model $P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_{it}|Z_{i(t-1)})$ can be substituted by: $\prod_{i<j} \psi(Z_{it}, Z_{jt}; \beta_{ijt}) \prod_{i<j} P(\beta_{ijt}|\beta_{ij(t-1)}) \prod_{i=1}^{N} P(Z_{it}|Z_{i(t-1)})$ where $\psi(Z_{it}, Z_{jt}; \beta_{ijt})$ is the pairwise potential.

Mode variable. In order to model transitions between interactions, we describe the transition probability $P(\beta_{ijt}|\beta ij(t-1))$ as p_β if $\beta_{ijt} = \beta ij(t-1)$, and as $1-p_\beta$, otherwise. In our implementation, p_β is set to be 0.9. Again, this variable is automatically estimated given observations. Thus,

$$\psi(Z_{it}, Z_{jt}; \beta_{ijt}) = \begin{cases} \psi_g(Z_{it}, Z_{jt}), \ if \ \beta_{ijt} = 1 \\ \psi_r(Z_{it}, Z_{jt}), \ otherwise \end{cases} \tag{6}$$

Repulsion model. In order to push away targets that are too close, we model the repulsion potential as $\psi_r(Z_{it}, Z_{jt}) = e^{\frac{-1}{c_r r_{ij}}}$ where r_{ij} denotes the distance between two targets in the 3D space and c_r is a parameter controlling the repulsion force between those. This pairwise potential has larger values as two targets are located far away, and has a value closer to 0 when two targets are nearby.

Group Motion Model. The assumption here is that, if two targets are moving together while keeping the same distance (group movement), they will tend to keep the same relative location in consecutive time frames as well. This can be modelled as $p_{it} - p_{jt} \approx p_{i(t-1)} - p_{j(t-1)}$, which is in turn equivalent to $v_{it} \approx v_{jt}$, where p_{it} is the target's location in 3D and v_{it} is the velocity component of Z_{it}. Thus, we model the group motion potential as a $\psi_g(Z_{it}, Z_{jt}) = e^{-c_g * \|v_{it} - v_{jt}\|}$, where c_g is a parameter controlling the similarity of velocities. Since groups of targets are also defiend by the distance among each others, we enforce that the distance r_{ij} between two targets should be close enough in order to be considered as a group. This can be modeled by multiplying ψ_g with a soft step function and obtain $\psi_g(Z_{it}, Z_{jt}) = \frac{1}{1+e^{s_g(r_{ij}-t_g)}} \dot{e}^{-c_g * \|v_{it} - v_{jt}\|}$, where s_g is a parameter regulating the slope of soft step function and t_g is a distance threshold.

4 Tracking Multi-target by MCMC Particle Filter

Considering the complexity of the given probabilistic formulation, it is extremely challenging to design an analytical inference method for estimating the Maximum-a-Posteri solution. This challenge is due to the presence of: 1) the high nonlinearity of projection function(EQ.1); 2) the MRF induced by pairwise potential; 3) the non-gaussian nature of the posterior and prior distribution. Instead of relying on an analytical solution, we employ a sampling based sequential filtering algorithm (the Monte-Carlo Markov-Chain (MCMC) Particle Filter [13]). Inspired by [13], we employ MCMC sampling scheme to propagate the posterior distribution in the particle filtering framework. In each frame, we keep the number of samples without weights and thus approximate the prior distribution with N dirac samples $P(\Omega_{t-1}|\chi^{t-1}) \approx \{\Omega^r_{t-1}\}^N_{r=1}$. Subsequently the final posterior distribution in time t can be approximated by following equation

$$P(\Omega_t|\chi^t) \approx cP(X_t|Z_t,\Theta_t)P(\tau_t|G_t,\Theta_t)\sum_{r=1}^{N} P(Z_t|Z^{(r)}_{t-1})P(G_t|G^{(r)}_{t-1})P(\Theta_t|\Theta^{(r)}_{t-1}) \quad (7)$$

As a condition for the construction of an MCMC method, we need to design a Markov chain over the joint space of Ω. This has the same stationary distribution as the posterior distribution $P(\Omega_t|\chi^t)$. First, we define the *proposal distribution* as a combination of 1) weighted sampling from all existing targets, features, and camera and 2) appropriate random perturbation to the chosen node (additive gaussian or switching state). The proposal density can be represented as follows: 1) sample one hidden state(camera, target, feature) with probability $p_i = \frac{w_i}{\sum_{k=0}^{M} w_k}$. 2) if the camera Θ is selected, sample from a multinomial normal distribution to get new sample $\Theta'_t = \Theta^{(s)}_t + \mu$, where μ is the gaussian sample. 3) If a target Z_i is chosen: i) sample from a multinomial normal distribution and add it to x, z, v_x, v_z, h; ii) switch the class variable c_i by p^f_c; iii) switch interaction mode β_{ijt} for all j by p^f_β. 4) If a feature G_i is selected: i) sample from a multinomial normal distribution and add it to x, z; ii) switch the indicator variable α_i by p^f_α. Here we assign higher weight w_i onto the camera's state since camera parameters are coupled with all states and so this estimation process requires a larger number of trials. Since this proposal distribution is symmetric (the probability to move from Ω'_t to $\Omega^{(s)}_t$ and from $\Omega^{(s)}_t$ to Ω'_t is the same), we can drop the proposal distribution term from the acceptance ratio a. Thus a can be written as :

$$a = \begin{cases} \frac{\prod_{i=1}^{n} P(X_{it}|Z'_{it},\Theta'_t)\prod_{i=1}^{m} P(\tau_{it}|G'_{it},\Theta'_t)P(\Omega'_t|\chi^{t-1})}{\prod_{i=1}^{n} P(X_{it}|Z^{(s)}_{it},\Theta^{(s)}_t)\prod_{i=1}^{m} P(\tau_{it}|G^{(s)}_{it},\Theta^{(s)}_t)P(\Omega^{(s)}_t|\chi^{t-1})} & ,when\ the\ camera\ is\ chosen \\ \frac{P(X_{kt}|Z'_{kt},\Theta'_t)P(\Omega'_t|\chi^{t-1})}{P(X_{kt}|Z^{(s)}_{kt},\Theta^{(s)}_t)P(\Omega^{(s)}_t|\chi^{t-1})} & ,when\ a\ target\ is\ chosen \\ \frac{\prod_{i=1}^{n} P(\tau_{it}|G'_{it},\Theta'_t)P(\Omega'_t|\chi^{t-1})}{\prod_{i=1}^{n} P(\tau_{it}|G^{(s)}_{it},\Theta^{(s)}_t)P(\Omega^{(s)}_t|\chi^{t-1})} & ,when\ a\ feature\ is\ chosen \end{cases} \quad (8)$$

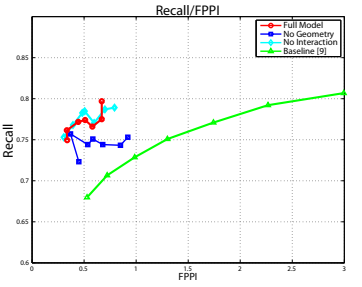


Fig. 3. Our full model obtains the best Recall rates when compared to the baseline detector [9]. Note that the effect of τ is quite significant. To obtain plots, we run the algorithm with different threshold values for the detector.

where $P(\Omega_t|\chi^{t-1}) \approx \sum_{r=1}^{N} P(Z_t|Z_{t-1}^{(r)})P(G_t|G_{t-1}^{(r)})P(\Theta_t|\Theta_{t-1}^{(r)})$. Note that the computation of other observation likelihood is not necessary in the case of sampling a target or ground feature. Even though the computation of prediction term $P(\Omega_t|\chi^{t-1})$ involves several multiplication and summation operations. The majority of target's state remains unchanged during a sampling iteration. This enables an efficient implementation by caching unchanged priors.

5 Experimental Results and Implementation Details

In order to evaluate the performance of our tracking algorithm, we applied it to two datasets: our own dataset and ETH moving vehicle dataset [1]. During all experiments, we assumed rough initial camera configuration is given (focal length, camera height, horizon). Since cameras are not calibrated in our dataset, camera configuration is initialized to some reasonable value. For ETH dataset, we use the calibration information provided by the authors to initiate the algorithm.

Detector. Human targets are detected using the part-based detector [9] which is trained on the VOC 2006 dataset. As a benchmark, we report the recall/FPPI (False Positive Per Image) measure of the detector [9] along with our result in Fig.3.

Mean-shift. Appearance-based tracks are obtained using the mean-shift tracker with a similarity threshold of 0.94 in order to avoid false correspondences.

Feature selection. We extract 1300 KLT features to cover the visible area for every frame. After extracting KLT features, we select candidate ground plane features by rejecting out those lying on a target's prediction area or above the horizon. Among those candidates, a maximum number of 10 features were used in MCMC to reduce the computational burden. In practice these were sufficient to obtain robust estimation of the camera parameters.

MCMC Implementation. In the actual implementation of MCMC particle filter, we incorporated a burn-in and thinning scheme. We ignored the inital

Table 1. The recall/FPPI of ETH[1] algorithm is measured at the point having similar FPPI value to our algorithm. The performance of our algorithm was comparable or superior to ETH algorithm for Seq.#2. Notice in Seq.#3, as the number of false positives increases drastically, our algorithm was not able to correctly differentiate between true and false positives.

Recall/FPPI on ETH dataset			
Method		Seq.#2	Seq.#3
Our Algorithm	Recall	0.556 0.541 0.519	0.339 0.421 0.497
	FPPI	0.792 0.442 0.267	2.792 1.608 0.647
ETH [1]	Recall	0.498 0.404 0.338	0.673 0.616 0.484
	FPPI	0.781 0.431 0.262	2.772 1.593 0.638

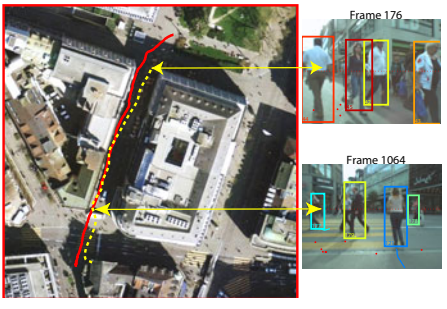


Fig. 4. The camera's trajectory estimated by our algorithm (yellow) for Seq.#2 and by [1] (red). The trajectories are overlaid onto the satellite image by rotating and rescaling with the same factor in (x,z) direction. Notice our trajectory was obtained without using stereo cameras or SFM.

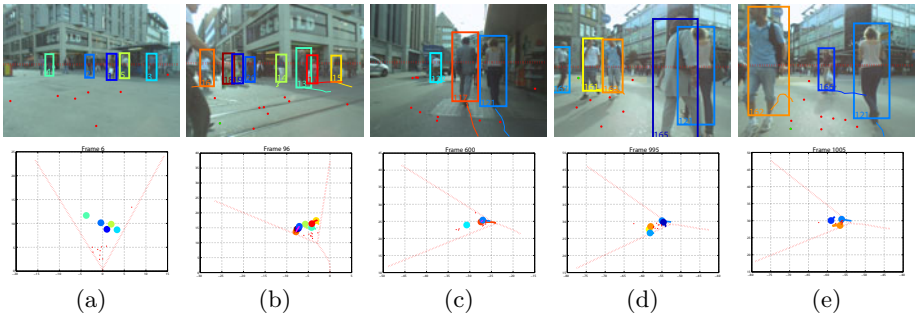


Fig. 5. Examples of tracking subsequences obtained by our algorithm in Seq.#2. Top: trajectories in the image plane; bottom: trjectory estimates in 3D space along with the camera's location and viewing angle estimates.

number of samples to avoid wrong estimation. In each experiment, we set 2000 of burn-in samples and selected one out of 100 samples.

Semi-static Camera. Firstly, we show the performance of our algorithm using our own dataset. Our dataset is ideal to test sequences with semi-static camera mation [3]. It is composed of nine short video sequences recorded at 30FPS by a hand-held video camera. These contain random shakes, sudden camera panning, and multiple number of pedestrians. The dataset contains 4749 frames and 3685 pedestrian annotations in total (every 10th frame is manually annotated). In our experiment, we ignored every other frame, so the tracking algorithm is applied at 15 fps. We report the quantitative measure of the performance by the recall/FPPI rate. Fig.3 shows the overall recall/FPPI curve obtained by our algorithm and the baseline detector [9]. In order to evaluate the effect of the tracked ground features and interaction model to the final performance, we show recall/FPPI curves when no features τ were tracked (blue) and no interaction models were used (cyan).

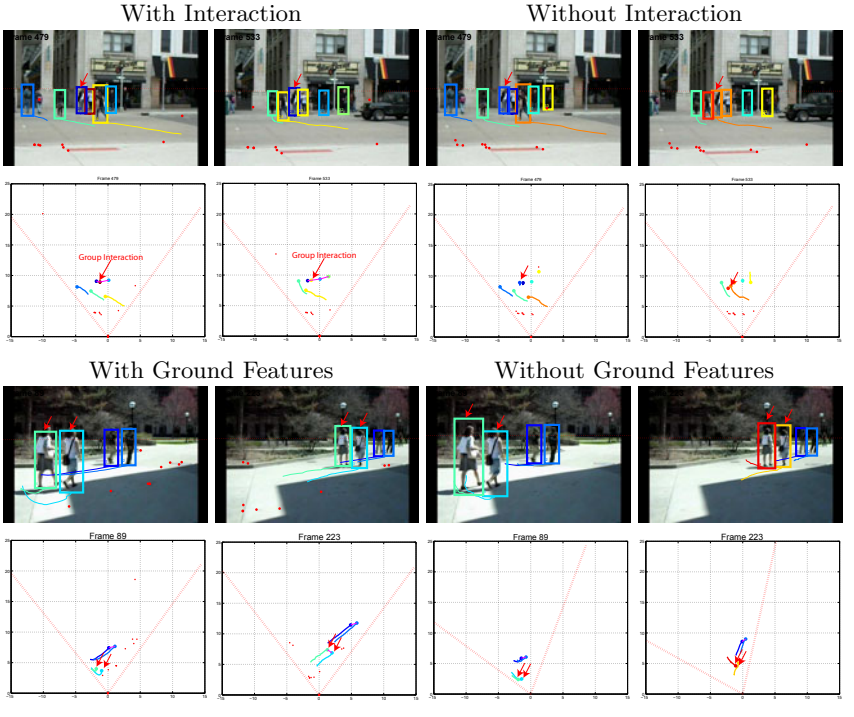


Fig. 6. Tracking comparison. Upper two rows: example of tracking results with and without the interaction model. Note that the group interaction (magenta link) prevents possible ID switch (red arrows) between two similar targets after occlusion. Bottom two rows: tracking results with and without ground features. Ground features not only helps the algorithm estimate the camera motion robustly but also generates better trajectories. Note that the ID of two targets (red arrows) are not maintained (if ground features are not used) due to poor camera motion estimation.

ETH dataset. To show the versatility of our algorithm, we also applied our algorithm on the ETH dataset [1]. ETH dataset is taken by a stereo pair of cameras mounted on a small cart which navigates through busy downtown environment. We evaluated our algorithm using only left camera images for tracking. Among five sequences listed in [1], we applied our algorithm on the "Seq#2" and "Seq#3". Both sequences contain large number of pedestrians walking around a downtown area. In both sequences, our algorithm was working better or as well as [1]. Unlike [1], we used only the single (left) camera sequence throughout the experiments so the performance of tracking algorithm was solely relied on better estimation capability of our algorithm. Quantitative results are reported in Table 1. Following the evaluation criteria of [11,1], we report the number of pedestrians, the number of trajectories, the number of mostly hit trajectories, mostly missed trajectories, the number of false alarm, and the number of ID switch for the Seq.#2 as following: 33, 47, 28, 8, 3, 2. Since [1] did not report exact frame numbers, we choose 350 to 800 frames. Following [1], we also counted as a new trajectory if a person is occluded for more than 10 frames. Overall, about 60% of the trajectories were covered and most of the missed trajectories belonged to small people. Qualitative results are reported in Fig. 4, 5 and 6. For additional results, please visit our project's webpage [3].

6 Conclusion

In this paper, we presented a fully automatic multi-target tracking algorithm. Different sources of information were integrated into one coherent probabilistic framework and all the variable was estimated in a joint fashion. Our framework has very flexible structure, so other additional cues can be incorporated for further stabilization. Combining other geometric cues is our plan for investigation.

Acknowledgment. This work is supported through a grant from Ford Motor Company via the Ford-U of M Innovation Alliance (Award #N011537). We also would like to thank Jeffrey Remillard for his valuable feedbacks throughout this project.

References

1. Ess, A., Leibe, B., Schindler, K., van Gool, L.: A mobile vision system for robust multi-person tracking. In: CVPR (2008)
2. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. In: CVPR (2006)
3. Project-webpage (2010), http://www.eecs.umich.edu/vision/mttproject.html
4. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI (2002)
5. Avidan, S.: Ensemble tracking. PAMI (2007)
6. Yin, Z., Collins, R.: On-the-fly object modeling while tracking. In: CVPR (2007)
7. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. PAMI 26, 810–815 (2004)

8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. In: PAMI (2009)
10. Okuma, K., Taleghani, A., Freitas, N.D., Freitas, O.D., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
11. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors (2007)
12. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV (2009)
13. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets (2005)
14. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV (2009)
15. Scovanner, P., Tappen, M.: Learning pedestrian dynamics from the real world. In: ICCV (2009)
16. Tomasi, C., Kanade, T.: Detection and tracking of point features. In: Carnegie Mellon University Technical Report (1991)
17. Kuhn, H.W.: The hungarian method for the assignment problem. In: Naval Research Logistics Quarterly (1955)
18. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. PAMI 29, 1052–1067 (2007)
19. Smith, P., Reid, I., Davison, A.: Real-time monocular slam with straight lines. In: BMVC (2006)

Joint Estimation of Motion, Structure and Geometry from Stereo Sequences

Levi Valgaerts[1], Andrés Bruhn[1], Henning Zimmer[1], Joachim Weickert[1], Carsten Stoll[2], and Christian Theobalt[2]

[1] Mathematical Image Analysis Group, Saarland University, Saarbrücken, Germany
{valgaerts,bruhn,zimmer,weickert}@mia.uni-saarland.de
[2] Max-Planck Institute for Informatics, Saarbrücken, Germany
{stoll,theobalt}@mpi-inf.mpg.de

Abstract. We present a novel variational method for the simultaneous estimation of dense scene flow and structure from stereo sequences. In contrast to existing approaches that rely on a fully calibrated camera setup, we assume that only the intrinsic camera parameters are known. To couple the estimation of motion, structure and geometry, we propose a joint energy functional that integrates spatial and temporal information from two subsequent image pairs subject to an unknown stereo setup. We further introduce a normalisation of image and stereo constraints such that deviations from model assumptions can be interpreted in a geometrical way. Finally, we suggest a separate discontinuity-preserving regularisation to improve the accuracy. Experiments on calibrated and uncalibrated data demonstrate the excellent performance of our approach. We even outperform recent techniques for the rectified case that make explicit use of the simplified geometry.

1 Introduction

For many tasks in computer vision, such as vehicle navigation, motion capture and dynamic rendering, it is essential to recover the three-dimensional displacement field of a scene. This so called *scene flow* represents the real 3D motion of objects – as opposed to optical flow that only describes the projection of this motion on the 2D image plane [23]. Since depth information is required to determine 3D motion, scene flow can not be computed without estimating the scene structure as well. In contrast to structure from motion, scene flow does not relate to a static world. Instead, objects in the scene are allowed to move freely and in a non-rigid fashion. Thus, for estimating scene flow, stereo sequences are required that provide two views per time instance.

Existing scene flow algorithms often treat stereo and motion independently. In fact, most of them rely on a sequential computation of the scene flow and structure [23,19,26,20,24]. However, to improve the quality of the estimation it is important that 3D motion and shape estimation are coupled. This can be achieved by exploiting the spatial and temporal dependencies in the image sequence [26,12,4,18,6]. Among those methods that solve for the scene flow and structure simultaneously, variational approaches play a major role. Some of these techniques parameterise the problem directly in 3D space [6]. Others are based on optical flow computation [26,12,18] and have consistently improved their results in the wake of increasing optical flow accuracy.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 568–581, 2010.

All of the afore mentioned methods have one aspect in common: they assume that the cameras have been calibrated beforehand. However, in order to deal with general stereo setups without requiring an explicit calibration step, it would be desirable to jointly estimate the scene flow, the scene structure *and* the stereo geometry.

In this paper we thus propose a variational scene flow method for *uncalibrated* stereo sequences. We do this by integrating the spatial and temporal information from two stereo pairs in a global energy functional while simultaneously estimating the unknown stereo geometry in consecutive time steps. Assuming that the internal camera parameters are known, our method allows to recover the dense scene structure and the dense scene flow up to a scale factor. Apart from this novel generalised model, we make two additional contributions: First, within the multiresolution framework required to handle large displacements, we introduce a tensor-based notation for linearised constraints. This notation allows to normalise these constraints such that deviations from the model can be interpreted as geometrical distances. Secondly, we propose a regularisation strategy that penalises discontinuities in the different displacement fields separately. This makes sense, since motion and depth continuities do not necessarily coincide. Our experiments clearly demonstrate the benefits of both contributions and show the favourable performance of our method compared to recent techniques for the rectified case.

Related Work. In the context of scene flow estimation, closely related to our work are the methods [26,12,18], which jointly compute spatial and temporal motion fields by minimising a single energy. In particular the method of Huguet and Devernay [12] uses similar data constraints as our approach. However, it applies a joint smoothness term to all displacement fields. A more adequate separate treatment of the smoothness term is proposed by Wedel *et al.* [25] who decouple the estimation of structure and motion to achieve real-time performance. However, in their case, the separate smoothness term does not yield more accurate results than their preceding work with joint regularisation [24]. All of the previous approaches are based on rectified sequences and do not consider a suitable constraint normalisation. Apart from these methods that parameterise the displacements in terms of image coordinates, there are also techniques that work directly in 3D space. Such techniques include methods based on reprojection errors [6], space carving and nonlinear optimisation [4], deformable meshes [9] and Markov Random Fields [13]. Moreover, all these methods rely on a previous calibration step, since they involve the use of projection matrices.

In the context of optical flow estimation, the work of Valgaerts *et al.* [22] and Zimmer *et al.* [27] are closest related to our approach. While the first one shows the benefit of jointly estimating dense displacements and the underlying stereo geometry, the second one proposes a normalisation of the data constraints to penalise a geometrically meaningful distance. In our approach we extend both ideas to scene flow and unify them by normalising both data and stereo constraints.

Paper Organisation. In Sect. 2 we derive our variational model for the uncalibrated case. Important issues like incremental computation and constraint normalisation are then discussed in Sect. 3. While Sect. 4 is dedicated to the alternating minimisation of the proposed energy, our results and a comparison to the literature are presented in Sect. 5. The paper concludes with a summary in Sect. 6.

2 A Scene Flow Model for Uncalibrated Stereo Sequences

In the following we consider the classical four-frame case depicted in Fig. 1. It consists of two consecutive image pairs of a stereo sequence: the left image $g_{1\mathrm{l}}(\boldsymbol{x})$ and the right image $g_{1\mathrm{r}}(\boldsymbol{x})$ at time t and the left image $g_{2\mathrm{l}}(\boldsymbol{x})$ and right image $g_{2\mathrm{r}}(\boldsymbol{x})$ at time $t+1$. Here $\boldsymbol{x} = (x, y)^\top$ denotes the location in a rectangular image domain $\Omega \subset \mathbb{R}^2$ that is assumed to be the same for all images. We furthermore assume that the sequence has been recorded by a single fixed stereo rig, i.e. there exists a common fundamental matrix F that describes the epipolar geometry [7] of the stereo pairs at time t and $t+1$.

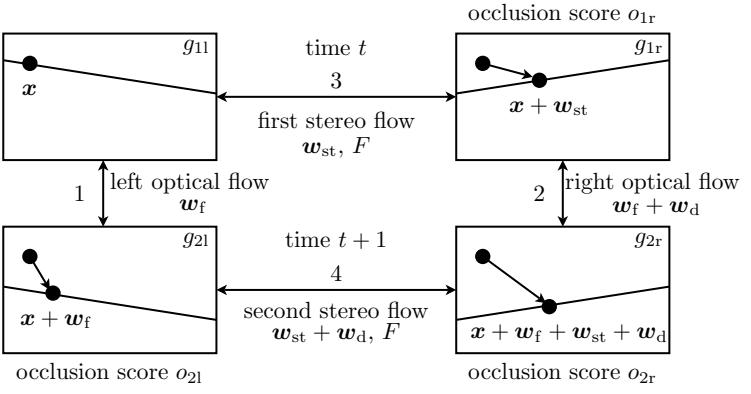


Fig. 1. The correspondences between the four frames of a binocular stereo sequence

In contrast to previous variational methods that start out from a rectified stereo sequence [24,12], our method assumes a general stereo geometry with unknown fundamental matrix. As a consequence, the stereo correspondences do not take on the form of a scalar valued disparity but of a 2-dimensional displacement field that we will refer to as *stereo flow*. In total, we consider four types of correspondences in our model: two optical flows between consecutive frames of the same camera (left, right) and two stereo flows between the left and right frame at the same time instance (t, $t+1$). Exploiting the dependencies in Fig. 1, these correspondences can be parameterised by six unknown functions with respect to the reference image $g_{1\mathrm{l}}(\boldsymbol{x})$: the first stereo flow $\boldsymbol{w}_{\mathrm{st}} = (u_{\mathrm{st}}, v_{\mathrm{st}})^\top$, the left optical flow $\boldsymbol{w}_{\mathrm{f}} = (u_{\mathrm{f}}, v_{\mathrm{f}})^\top$ and the difference flow $\boldsymbol{w}_{\mathrm{d}} = (u_{\mathrm{d}}, v_{\mathrm{d}})^\top$ that can be interpreted as a change in optical flow or a change in stereo flow. Moreover, we have seven degrees of freedom from the fundamental matrix F, which restricts points to lie on corresponding epipolar lines, as shown in Fig. 1. These degrees of freedom arise from the fact that F is a 3×3 matrix of rank 2 that is defined up to a scale factor. For given intrinsic camera parameters, knowing the fundamental matrix is sufficient to recover projection matrices (P_1, P_2) for the left and the right image sequence [10]. Together with the stereo flow $\boldsymbol{w}_{\mathrm{st}}$ at time t, these matrices allow to reconstruct a reference image point up to a scale in the camera coordinate system. To obtain a reconstruction at time $t+1$ and the scene flow relative to the cameras, the left optical flow $\boldsymbol{w}_{\mathrm{f}}$ and the flow change $\boldsymbol{w}_{\mathrm{d}}$ have to be known additionally.

Since we are interested in a joint computation of the 3D motion, structure and geometry, that are parameterised by $(\boldsymbol{w}_{\mathrm{f}}, \boldsymbol{w}_{\mathrm{st}}, \boldsymbol{w}_{\mathrm{d}})^{\top}$ and F, we propose to minimise a global energy functional that combines the spatial and temporal information of the different views while imposing geometric consistency. This functional has the form

$$\mathcal{E} = \int_{\Omega} (\mathcal{E}_{\mathrm{D}} + \mathcal{E}_{\mathrm{E}} + \mathcal{E}_{\mathrm{S}}) \ \mathrm{d}\boldsymbol{x} \ , \tag{1}$$

where $\mathcal{E}_{\mathrm{d}}$ is the data term that models the assumption that certain image features remain constant between the four frames, $\mathcal{E}_{\mathrm{E}}$ is the epipolar term that relates the stereo views by the unknown epipolar geometry, and $\mathcal{E}_{\mathrm{S}}$ is the smoothness term that assumes the solution to be piecewise smooth. In the following we will detail on the different terms.

2.1 Data Constraints

Let us now derive the four constraints that model the relation between the four input images w.r.t. the reference image. For simplicity, let us assume for the moment that the brightness of corresponding image points remains constant between all frames [11]. Following the enumeration of constraints in Fig. 1 we obtain the expressions

$$\mathcal{E}_{\mathrm{D1}} = \Psi\left(|g_{2\mathrm{l}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}}) - g_{1\mathrm{l}}(\boldsymbol{x})|^2\right) \ , \tag{2}$$

$$\mathcal{E}_{\mathrm{D2}} = \Psi\left(|g_{2\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + \boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}}) - g_{1\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{st}})|^2\right) \ , \tag{3}$$

$$\mathcal{E}_{\mathrm{D3}} = \Psi\left(|g_{1\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{st}}) - g_{1\mathrm{l}}(\boldsymbol{x})|^2\right) \ , \tag{4}$$

$$\mathcal{E}_{\mathrm{D4}} = \Psi\left(|g_{2\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + \boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}}) - g_{2\mathrm{l}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}})|^2\right) \ . \tag{5}$$

The first two terms correspond to an optical flow constraint between two time instances, while the last two terms arise from a stereo correspondence at consecutive time steps. As in [12] we choose to penalise all constraints separately since outliers for optical flow and stereo do not necessarily occur in the same location. As penalty function Ψ we choose the regularised L_1 norm $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ with $\epsilon = 0.001$ as proposed e.g. in [2]. In our final model we include the gradient constancy assumption to cope with varying illumination and extend the expressions above to RGB colour images. Then the first term (2) becomes

$$\mathcal{E}_{\mathrm{D1}} = \Psi\left(\sum_{i=1}^{3}\left(\left|g^i_{2\mathrm{l}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}}) - g^i_{1\mathrm{l}}(\boldsymbol{x})\right|^2 + \gamma\left|\nabla g^i_{2\mathrm{l}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}}) - \nabla g^i_{1\mathrm{l}}(\boldsymbol{x})\right|^2\right)\right) \ , \tag{6}$$

where $\gamma \geq 0$ is a weighting factor, the symbol $\nabla = (\partial_x, \partial_y)^{\top}$ denotes the spatial gradient operator, and g^1, g^2, and g^3 represent the three RGB colour channels. The constraints $\mathcal{E}_{\mathrm{D2}}$, $\mathcal{E}_{\mathrm{D3}}$ and $\mathcal{E}_{\mathrm{D4}}$ are extended in the same way.

2.2 Occlusion Scores

In order to handle situations, where parts of the scene become occluded due to motion or a change of camera viewpoint, we additionally introduce occlusion scores. For instance, the score $o_{1\mathrm{r}} : \Omega \rightarrow \{0, 1\}$ takes on the value 1 for points in the reference image

g_{1l} that are visible in g_{1r}, and 0 otherwise. Once the fundamental matrix is known and the projection matrices (P_1, P_2) have been computed, the values of o_{1r} can be determined by projecting the reconstruction at time t back on the image plane using $P2$. Of all the points that reproject onto the same location, the one that lies closest to the optical centre of $P2$ will be marked as visible. This technique is also known as *Z-buffering*. The scores o_{2l} and o_{2r} for the image pairs (g_{1l}, g_{2l}) and (g_{1l}, g_{2r}) are determined analogously by reprojection on time $t+1$ with P_1 and P_2, respectively. The four data terms are multiplied by the occlusion scores to switch them off where the constancy assumptions can not be fulfilled. This yields the final data term

$$\mathcal{E}_{\mathrm{D}} = o_{2l}\,\mathcal{E}_{\mathrm{D1}} + o_{1r}\,o_{2r}\,\mathcal{E}_{\mathrm{D2}} + o_{1r}\,\mathcal{E}_{\mathrm{D3}} + o_{2l}\,o_{2r}\,\mathcal{E}_{\mathrm{D4}} \quad . \tag{7}$$

Note that each term has to be multiplied by the occlusion scores of the images that occur in the according data constraint, since the reappearance of points in g_{2r} that are occluded in g_{1r} or g_{2l} is not noticed by the reference image.

2.3 Epipolar Constraints

Let us now model the geometric relation between the left and right images of the stereo pairs (g_{1l}, g_{1r}) and (g_{2l}, g_{2r}). To this end we introduce two terms that relate the unknown flows and the fundamental matrix F via the respective epipolar constraints [16]:

$$\mathcal{E}_{\mathrm{E1}} = \Psi\left(\left((\boldsymbol{x}+\boldsymbol{w}_{\mathrm{st}})_{\mathrm{h}}^{\top} F\,(\boldsymbol{x})_{\mathrm{h}}\right)^2\right) \text{ , and} \tag{8}$$

$$\mathcal{E}_{\mathrm{E2}} = \Psi\left(\left((\boldsymbol{x}+\boldsymbol{w}_{\mathrm{f}}+\boldsymbol{w}_{\mathrm{st}}+\boldsymbol{w}_{\mathrm{d}})_{\mathrm{h}}^{\top} F\,(\boldsymbol{x}+\boldsymbol{w}_{\mathrm{f}})_{\mathrm{h}}\right)^2\right) \quad . \tag{9}$$

Here the subscript h denotes the use of homogeneous coordinates, i.e. $(\boldsymbol{x})_{\mathrm{h}} = (x, y, 1)^{\top}$. Both terms $\mathcal{E}_{\mathrm{E1}}$ and $\mathcal{E}_{\mathrm{E2}}$ are soft constraints that penalise deviations of a point from its epipolar line. The use of Ψ increases the robustness of the estimation of F with respect to outliers. While the first epipolar term can be modelled completely in accordance with [22], the second epipolar constraint is much more complicated: Although it is linear in $\boldsymbol{w}_{\mathrm{st}}$ and $\boldsymbol{w}_{\mathrm{d}}$, it is quadratic with respect to the left optical flow $\boldsymbol{w}_{\mathrm{f}}$. This makes the minimisation of the corresponding energy difficult. To nevertheless obtain a linear expression in all flows we thus propose to introduce an auxiliary variable $\boldsymbol{w}_{\mathrm{a}} = (u_{\mathrm{a}}, v_{\mathrm{a}})^{\top}$, which is assumed to be close to $\boldsymbol{w}_{\mathrm{f}}$, and split up the epipolar constraint such that $\boldsymbol{w}_{\mathrm{f}}$ and $\boldsymbol{w}_{\mathrm{a}}$ take on symmetric roles. In this way we can approximate term (9) via

$$\begin{aligned}\mathcal{E}_{\mathrm{E2}} = \Psi\Big(& \tfrac{1}{2}\left((\boldsymbol{x}+\boldsymbol{w}_{\mathrm{f}}+\boldsymbol{w}_{\mathrm{st}}+\boldsymbol{w}_{\mathrm{d}})_{\mathrm{h}}^{\top} F\,(\boldsymbol{x}+\boldsymbol{w}_{\mathrm{a}})_{\mathrm{h}}\right)^2 \\ & + \tfrac{1}{2}\left((\boldsymbol{x}+\boldsymbol{w}_{\mathrm{a}}+\boldsymbol{w}_{\mathrm{st}}+\boldsymbol{w}_{\mathrm{d}})_{\mathrm{h}}^{\top} F\,(\boldsymbol{x}+\boldsymbol{w}_{\mathrm{f}})_{\mathrm{h}}\right)^2\Big) + \mu\left(|\boldsymbol{w}_{\mathrm{f}}-\boldsymbol{w}_{\mathrm{a}}|^2\right) \quad ,\end{aligned} \tag{10}$$

where μ is the weight of the additional similarity term that is required to couple $\boldsymbol{w}_{\mathrm{a}}$ and $\boldsymbol{w}_{\mathrm{f}}$. Introducing the weights β_1 and β_2 we obtain the final epipolar term

$$\mathcal{E}_{\mathrm{E}} = \beta_1\,\mathcal{E}_{\mathrm{E1}} + \beta_2\,\mathcal{E}_{\mathrm{E2}} \quad . \tag{11}$$

To avoid the trivial solution we additionally impose the constraint $\|F\|_{\mathrm{Frob}}^2 = 1$ on the Frobenius norm of the fundamental matrix F as proposed in [14].

2.4 Smoothness Constraints

Let us finally detail on the design of the smoothness term. Its task is to regularise the problem in locations where the remaining terms do not guarantee a unique solution (aperture problem) or to fill in information in the presence of outliers, e.g. occlusions. Because there often exists an overlap between the discontinuities of $\boldsymbol{w}_{\mathrm{f}}$, $\boldsymbol{w}_{\mathrm{st}}$ and $\boldsymbol{w}_{\mathrm{d}}$, the authors of [12] suggested a joint piecewise smoothness assumption on all flows. With our method, however, we want to cover the general case where the flow and stereo discontinuities do not necessarily coincide, e.g. for different in-plane motions. Therefore we propose a separate penalisation of the flow gradients:

$$\mathcal{E}_{\mathrm{S1}} = \Psi\left(|\nabla \boldsymbol{w}_{\mathrm{f}}|^2\right) \;, \mathcal{E}_{\mathrm{S2}} = \Psi\left(|\nabla \boldsymbol{w}_{\mathrm{st}}|^2\right) \text{, and } \mathcal{E}_{\mathrm{S3}} = \Psi\left(|\nabla \boldsymbol{w}_{\mathrm{d}}|^2\right) \quad , \tag{12}$$

with $|\nabla \boldsymbol{w}_*|^2 := |\nabla u_*|^2 + |\nabla v_*|^2$, where $*$ stands for f, st or d. The penalisation via the subquadratic function Ψ, as defined before, equals total variation (TV) regularisation [21]. This gives rise to the smoothness term

$$\mathcal{E}_{\mathrm{S}} = \alpha_1 \, \mathcal{E}_{\mathrm{S1}} + \alpha_2 \, \mathcal{E}_{\mathrm{S2}} + \alpha_3 \, \mathcal{E}_{\mathrm{S3}} \;\; , \tag{13}$$

where α_1, α_2, α_3 are positive weights that balance the smoothness assumptions for the three displacement fields.

3 Linearisation and Normalisation

Substituting all data, epipolar and smoothness terms into (1) we obtain an energy functional that is rather complicated. Moreover, it is non-convex, since the unknown flows appear implicitly in the arguments of the data term. A common strategy to resolve this problem is to perform an incremental computation of the unknowns within a coarse-to-fine multiscale approach. This can either be done by a fixed point iteration on the Euler-Lagrange equations [2] or by a series of energies that approximate the original model on every resolution level [17]. In the following we stick to the second strategy and discuss how the corresponding energy for each level can be derived. Assuming that solutions $\boldsymbol{w}_{\mathrm{f}}$, $\boldsymbol{w}_{\mathrm{st}}$, $\boldsymbol{w}_{\mathrm{d}}$ and $\boldsymbol{w}_{\mathrm{a}}$ are available from a coarser scale, we aim at expressing the total energy in terms of the increments $d\boldsymbol{w}_{\mathrm{f}} = (du_{\mathrm{f}}, dv_{\mathrm{f}})$, $d\boldsymbol{w}_{\mathrm{st}} = (du_{\mathrm{st}}, dv_{\mathrm{st}})$, $d\boldsymbol{w}_{\mathrm{d}} = (du_{\mathrm{d}}, dv_{\mathrm{d}})$, and $d\boldsymbol{w}_{\mathrm{a}} = (du_{\mathrm{a}}, dv_{\mathrm{a}})$. This allows us to introduce a tensor notation which offers two advantages: (i) The convexity of the resulting energy functional in the flow increments becomes explicit, and (ii) a normalisation strategy can be applied that makes deviations from the model assumptions interpretable in a geometric way.

3.1 Linearisation in the Data Term

Let us first discuss the differential form of the data term by the example of the simplified data constraint from expression (3). Using a first order Taylor expansion to linearise this expression with respect to all increments we obtain the approximation

$$\begin{aligned} & g_{2\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + d\boldsymbol{w}_{\mathrm{f}} + \boldsymbol{w}_{\mathrm{st}} + d\boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}} + d\boldsymbol{w}_{\mathrm{d}}) - g_{1\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{st}} + d\boldsymbol{w}_{\mathrm{st}}) \\ \approx \;\; & g_{2\mathrm{r}} + \partial_x g_{2\mathrm{r}} \cdot (du_{\mathrm{f}} + du_{\mathrm{st}} + du_{\mathrm{d}}) + \partial_y g_{2\mathrm{r}} \cdot (dv_{\mathrm{f}} + dv_{\mathrm{st}} + dv_{\mathrm{d}}) \\ & - g_{1\mathrm{r}} - \partial_x g_{1\mathrm{r}} \cdot (du_{\mathrm{st}}) - \partial_y g_{1\mathrm{r}} \cdot (dv_{\mathrm{st}}) \;\; . \end{aligned} \tag{14}$$

Rearranging the terms and using the following abbreviations

$$g_{2z} = g_{2\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + \boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}}) - g_{1\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{st}}) \,, \tag{15}$$

$$g_{2\mathrm{r}x} = \partial_x g_{2\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + \boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}}) \,, \quad g_{2xz} = \partial_x g_{2z} \,, \tag{16}$$

$$g_{2\mathrm{r}y} = \partial_y g_{2\mathrm{r}}(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + \boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}}) \,, \quad g_{2yz} = \partial_y g_{2z} \;\; , \tag{17}$$

we can rewrite the linearised term in (14) as inner product

$$\boldsymbol{g}_2^\top \boldsymbol{d} = g_{2\mathrm{r}x} du_{\mathrm{f}} + g_{2\mathrm{r}y} dv_{\mathrm{f}} + g_{2xz} du_{\mathrm{st}} + g_{2yz} dv_{\mathrm{st}} + g_{2\mathrm{r}x} du_{\mathrm{d}} + g_{2\mathrm{r}y} dv_{\mathrm{d}} + g_{2z} \;\; , \tag{18}$$

where the two vectors are defined as $\boldsymbol{g}_2 := (g_{2\mathrm{r}x}, g_{2\mathrm{r}y}, g_{2xz}, g_{2yz}, g_{2\mathrm{r}x}, g_{2\mathrm{r}y}, g_{2z})^\top$ and $\boldsymbol{d} := (du_{\mathrm{f}}, dv_{\mathrm{f}}, du_{\mathrm{st}}, dv_{\mathrm{st}}, du_{\mathrm{d}}, dv_{\mathrm{d}}, 1)^\top$. The equation $\boldsymbol{g}_2^\top \boldsymbol{d} = 0$ can be seen as a multidimensional extension of the classical optical flow constraint [11]. Inserting it as squared argument into the penaliser Ψ yields the robustified quadratic form

$$\mathcal{E}_{\mathrm{D2}} = \Psi\left((\boldsymbol{g}_2^\top \boldsymbol{d})^2\right) = \Psi\left(\boldsymbol{d}^\top J_2 \, \boldsymbol{d}\right) \;\; , \tag{19}$$

where $J_2 = \boldsymbol{g}_2 \, \boldsymbol{g}_2^\top$ is a 7×7 matrix that provides coupling between all increments. By analogy to the motion tensor notation in optical flow estimation [3], we denote J_2 as *scene flow tensor*. The linearisation of the three remaining data constraints is carried out accordingly, and results in the 7×7 scene flow tensors J_1, J_3 and J_4. Missing dependencies between the variables give rise to zero tensor entries. Including the gradient constancy assumption and extending it to RGB colour images as in equation (6) is straightforward and leads to a weighted sum of the corresponding tensors [27].

3.2 Treatment of the Epipolar Term

The first epipolar term $(\boldsymbol{x}+\boldsymbol{w}_{\mathrm{st}}+d\boldsymbol{w}_{\mathrm{st}})_{\mathrm{h}}^\top F(\boldsymbol{x})_{\mathrm{h}}$ is already linear in the increment $d\boldsymbol{w}_{\mathrm{st}}$. As in the case of the data terms we can thus define the vector $\boldsymbol{d}_1 = (du_{\mathrm{st}}, dv_{\mathrm{st}}, 1)^\top$ and write the argument of the first epipolar term (8) as a quadratic form

$$\mathcal{E}_{\mathrm{E1}} = \Psi\left(\boldsymbol{d}_1^\top E_1 \, \boldsymbol{d}_1\right) \;\; . \tag{20}$$

The corresponding epipolar tensor E_1 of size 3×3 is defined as $(a_1, b_1, q_1)^\top (a_1, b_1, q_1)$, where a_1 and b_1 are the coefficients of the epipolar line $\boldsymbol{l} = F(\boldsymbol{x})_{\mathrm{h}}$, and q_1 is the scaled distance of the point $\boldsymbol{x}$ to this line [22]. However, care has to be taken with respect to symmetry when introducing the flow increments in the second epipolar term (10). The expanded differential variant of its argument reads

$$\begin{aligned}
&\tfrac{1}{4}\big(\,(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + d\boldsymbol{w}_{\mathrm{f}} + \boldsymbol{w}_{\mathrm{st}} + d\boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}} + d\boldsymbol{w}_{\mathrm{d}})_{\mathrm{h}}^\top \, F\,(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{a}})_{\mathrm{h}}\,\big)^2 \\
+&\tfrac{1}{4}\big(\,(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{a}} + d\boldsymbol{w}_{\mathrm{a}} + \boldsymbol{w}_{\mathrm{st}} + d\boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}} + d\boldsymbol{w}_{\mathrm{d}})_{\mathrm{h}}^\top \, F\,(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}})_{\mathrm{h}}\,\big)^2 \\
+&\tfrac{1}{4}\big(\,(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{a}} + d\boldsymbol{w}_{\mathrm{a}})_{\mathrm{h}}^\top \, F^\top\,(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + \boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}})_{\mathrm{h}}\,\big)^2 \\
+&\tfrac{1}{4}\big(\,(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{f}} + d\boldsymbol{w}_{\mathrm{f}})_{\mathrm{h}}^\top \, F^\top\,(\boldsymbol{x} + \boldsymbol{w}_{\mathrm{a}} + \boldsymbol{w}_{\mathrm{st}} + \boldsymbol{w}_{\mathrm{d}})_{\mathrm{h}}\,\big)^2 \;\; ,
\end{aligned} \tag{21}$$

where we have additionally included the last two terms with the transposed fundamental matrix to ensure a symmetrical treatment of the left and right flow increments. This is

required since as opposed to the first epipolar constraint variations can occur in both the left and the right image position. Since all terms of expression (21) are linear in the increments, the second epipolar term can be written as

$$\begin{aligned}\mathcal{E}_{\mathrm{E2}} = \quad & \Psi\left(\tfrac{1}{4}\,\boldsymbol{d}_2^\top E_2\,\boldsymbol{d}_2 + \tfrac{1}{4}\,\boldsymbol{d}_3^\top E_3\,\boldsymbol{d}_3 + \tfrac{1}{4}\,\boldsymbol{d}_4^\top E_4\,\boldsymbol{d}_4 + \tfrac{1}{4}\,\boldsymbol{d}_5^\top E_5\,\boldsymbol{d}_5\right) \\ & + \mu\left(|\boldsymbol{w}_{\mathrm{f}} + d\boldsymbol{w}_{\mathrm{f}} - \boldsymbol{w}_{\mathrm{a}} - d\boldsymbol{w}_{\mathrm{a}}|\right)^2 \quad , \end{aligned} \tag{22}$$

where we have defined the following vectors:

$$\boldsymbol{d}_2 = (du_{\mathrm{f}} + du_{\mathrm{st}} + du_{\mathrm{d}}, dv_{\mathrm{f}} + dv_{\mathrm{st}} + dv_{\mathrm{d}}, 1)\,, \quad \boldsymbol{d}_3 = (du_{\mathrm{a}}, dv_{\mathrm{a}}, 1)\,, \tag{23}$$

$$\boldsymbol{d}_4 = (du_{\mathrm{a}} + du_{\mathrm{st}} + du_{\mathrm{d}}, dv_{\mathrm{a}} + dv_{\mathrm{st}} + dv_{\mathrm{d}}, 1)\,, \quad \boldsymbol{d}_5 = (du_{\mathrm{f}}, dv_{\mathrm{f}}, 1)\ . \tag{24}$$

As in the case of the first epipolar tensor, the entries of the other epipolar tensors $E_i = (a_i, b_i, q_i)^\top (a_i, b_i, q_i)$, for $2 \leq i \leq 5$, are related to the coefficients of the epipolar lines.

3.3 Constraint Normalisation

In [27] the authors demonstrate that the linearised brightness constancy assumption for optical flow can be interpreted geometrically as a weighted distance of the estimated flow to the line described by the optical flow constraint. Equivalently, the multidimensional brightness constancy constraint in (18) can be considered as the weighted distance of the scene flow to the hyperplane described by $\boldsymbol{g}_2^\top \boldsymbol{d} = 0$. To obtain the actual distance to the hyperplane we have to normalise the constraint by dividing it by the magnitude of the hyperplane normal. Since the last entry of $\boldsymbol{d}$ is constant, this normal vector is given by the first six components of $\boldsymbol{g}_2$, i.e. $\boldsymbol{n} = (g_{2rx}, g_{2ry}, g_{2xz}, g_{2yz}, g_{2rx}, g_{2ry})^\top$. Now it becomes explicit why it is desirable to penalise the actual distance to the hyperplane: Unlike the original constraint this distance does not scale with the magnitude of the derivatives contained in $\boldsymbol{g}_2$. This prevents overweighting at unreliable structures such as noise or occlusions that typically manifest themselves in large image gradients. The corresponding normalised quadratic form is given by

$$\frac{1}{|\boldsymbol{n}|^2 + \zeta^2}\,(\boldsymbol{g}_2^\top \boldsymbol{d})^2 \;=\; \boldsymbol{d}^\top \left(\frac{J_2}{\sum_{i=1}^{6} (J_2)_{ii} + \zeta^2}\right) \boldsymbol{d} \;=\; \boldsymbol{d}^\top \widehat{J}_2\, \boldsymbol{d}\ , \tag{25}$$

where $\zeta = 0.1$ is a constant that avoids division by zero, and $\widehat{J}_2$ denotes the normalised version of J_2. We apply the same normalisation strategy to the remaining data terms. For the extension to the gradient constancy and colour images we refer to [27] .

Our normalisation idea is, however, not restricted to the scene flow tensors only. By normalising the epipolar tensors as well we obtain a widely used geometrical error measure from computer vision: the distance to the epipolar lines [16]. Analogously to (25), we can derive the normalisation factor for the epipolar tensors. It reads

$$|\boldsymbol{n}_i|^2 + \zeta^2 \;=\; \textstyle\sum_{j=1}^{2} (E_i)_{jj} + \zeta^2 \;=\; a_i^2 + b_i^2 + \zeta^2\ . \tag{26}$$

Division by this factor then results in the normalised epipolar tensors $\widehat{E}_i$, for $1 \leq i \leq 5$.

4 Minimisation and Numerical Solution

By combining all terms derived in Sect. 3, we obtain the following differential form of our energy that has to be minimised at each level of the coarse-to-fine approach:

$$\begin{aligned}
&\mathcal{E}(d\boldsymbol{w}_{\mathrm{f}}, d\boldsymbol{w}_{\mathrm{st}}, d\boldsymbol{w}_{\mathrm{d}}, d\boldsymbol{w}_{\mathrm{a}}, F) = \\
&\int_{\Omega} \Big(o_{2\mathrm{l}}\, \Psi\Big(\boldsymbol{d}^{\top} \widehat{J}_1\, \boldsymbol{d}\Big) + o_{1\mathrm{r}}\, o_{2\mathrm{r}}\, \Psi\Big(\boldsymbol{d}^{\top} \widehat{J}_2\, \boldsymbol{d}\Big) + o_{1\mathrm{r}}\, \Psi\Big(\boldsymbol{d}^{\top} \widehat{J}_3\, \boldsymbol{d}\Big) + o_{2\mathrm{l}}\, o_{2\mathrm{r}}\, \Psi\Big(\boldsymbol{d}^{\top} \widehat{J}_4\, \boldsymbol{d}\Big) \\
&\quad + \beta_1\, \Psi\Big(\boldsymbol{d}_1^{\top} \widehat{E}_1\, \boldsymbol{d}_1\Big) + \beta_2\, \Psi\Big(\tfrac{1}{4}\, \boldsymbol{d}_2^{\top} \widehat{E}_2\, \boldsymbol{d}_2 + \tfrac{1}{4}\, \boldsymbol{d}_3^{\top} \widehat{E}_3\, \boldsymbol{d}_3 + \tfrac{1}{4}\, \boldsymbol{d}_4^{\top} \widehat{E}_4\, \boldsymbol{d}_4 + \tfrac{1}{4}\, \boldsymbol{d}_5^{\top} \widehat{E}_5\, \boldsymbol{d}_5\Big) \\
&\quad + \alpha_1\, \Psi\Big(|\nabla(\boldsymbol{w}_{\mathrm{f}} + d\boldsymbol{w}_{\mathrm{f}})|^2\Big) + \alpha_2\, \Psi\Big(|\nabla(\boldsymbol{w}_{\mathrm{st}} + d\boldsymbol{w}_{\mathrm{st}})|^2\Big) + \alpha_3\, \Psi\Big(|\nabla(\boldsymbol{w}_{\mathrm{d}} + d\boldsymbol{w}_{\mathrm{d}})|^2\Big) \\
&\quad + \beta_2\, \mu\, \Big(|\boldsymbol{w}_{\mathrm{f}} + d\boldsymbol{w}_{\mathrm{f}} - \boldsymbol{w}_{\mathrm{a}} - d\boldsymbol{w}_{\mathrm{a}}|^2\Big) \Big) \mathrm{d}\boldsymbol{x} \;\; , \;\; \text{with} \;\; \|F\|^2_{\mathrm{Frob}} = 1 \;\; . \qquad (27)
\end{aligned}$$

Note that this energy is convex in the flow increments $d\boldsymbol{w}_{\mathrm{f}}$, $d\boldsymbol{w}_{\mathrm{st}}$, $d\boldsymbol{w}_{\mathrm{d}}$ and the auxiliary variable $d\boldsymbol{w}_{\mathrm{a}}$, since only squared arguments and convex penaliser functions are used. In order to minimise it under the given constraint $\|F\|^2_{\mathrm{Frob}} = 1$, we follow [22] and use the method of the Lagrange multipliers. We thus obtain the Lagrangian

$$\mathcal{L}(d\boldsymbol{w}_{\mathrm{f}}, d\boldsymbol{w}_{\mathrm{st}}, d\boldsymbol{w}_{\mathrm{d}}, d\boldsymbol{w}_{\mathrm{a}}, F, \lambda) = \mathcal{E}(d\boldsymbol{w}_{\mathrm{f}}, d\boldsymbol{w}_{\mathrm{st}}, d\boldsymbol{w}_{\mathrm{d}}, d\boldsymbol{w}_{\mathrm{a}}, F) + \lambda(1 - \boldsymbol{f}^{\top} \boldsymbol{f}) \;\; , \qquad (28)$$

where λ is the Lagrangian multiplier, and $\boldsymbol{f}$ is a vector that contains all 9 entries of F. This formulation suggests an alternating minimisation with two steps:

(i) Minimising the Lagrangian with respect to the flow increments leads to the corresponding Euler-Lagrange equations. By discretising them via finite difference approximations, one ends up with a nonlinear system of equations due to the robust function Ψ. To ensure fast convergence, we solve this system with a bidirectional multigrid framework based on a nonlinear point coupled Gauß-Seidel solver [3]. In the coarse-to-fine pyramid we use a downsampling factor of $\eta = 0.9$, while the images are warped onto the reference image using Coons patches based on bicubic interpolation [5].

(ii) Differentiation of the Lagrangian with respect to the fundamental matrix results in an eigenvalue problem [22] that is nonlinear due to Ψ and the normalisation weights (26). To solve this eigenvalue problem we apply a reweighted total least squares method in which the weights and the arguments of Ψ are fixed iteratively. We would like to point out that this step of the minimisation estimates the fundamental matrix from the *dense* correspondences of both stereo pairs.

The alternating computation of the flow increments and the fundamental matrix works as follows: The Euler-Lagrange equations are solved with a current estimate of the fundamental matrix. Using the newly computed flows, the fundamental matrix is updated by solving the eigenvalue problem. We extract a pair of camera matrices and perform a dense scene reconstruction by triangulation [10]. After recomputing the occlusion scores, the Euler-Lagrange equations are then solved again. This iterative process is repeated until convergence. We initialise the occlusion scores with 1 and compute the first iteration with disabled epipolar constraints.

Table 1. Evaluation of different methods on the rectified sphere sequence. Runtime on Intel Core2 1.86 GHz: ∼420 seconds. Parameters: $\alpha_1 = 2$, $\alpha_2 = 1.5$, $\alpha_3 = 0.3$, $\beta_1 = \beta_2 = 0.1$, $\gamma = 0.1$, $\mu = 1$.

Method	RMSE			AAE
	$(u_\mathrm{f}, v_\mathrm{f}, u_\mathrm{d}, v_\mathrm{d})$	$(u_\mathrm{f}, v_\mathrm{f})$	$(u_\mathrm{st}, v_\mathrm{st})$	$(u_\mathrm{f}, v_\mathrm{f})$
Our method initialised with [8]	**1.76**	**0.63**	3.8	1.17
Our method	1.78	**0.63**	5.5	**1.16**
Wedel *et al.* [24] with ground truth	2.40	0.65	–	1.40
Wedel *et al.* [24] (87%)	2.45	0.66	**2.9**	1.50
Huguet and Devernay [12]	2.51	0.69	3.8	1.75
Wedel *et al.* [24] (100%)	2.55	0.77	10.9	2.76

5 Experiments

We evaluate the performance of our method on synthetic stereo sequences with ground truth and on real world images. To assess the quality we compute the root mean square error RMSE of the scene flow $(u_\mathrm{f}, v_\mathrm{f}, u_\mathrm{d}, v_\mathrm{d})$, the optical flow $(u_\mathrm{f}, v_\mathrm{f})$ and the stereo flow $(u_\mathrm{st}, v_\mathrm{st})$, as well as the absolute angular error AAE of the optical flow, see [24]. As a quality measure for the fundamental matrix we use the error d_F according to [7]. It is determined by using the estimated fundamental matrix to randomly create a large number (100,000) of correspondences and the ground truth fundamental matrix to establish their epipolar lines. After computing the average distance between all points and lines, the roles of the matrices are reversed to obtain a symmetric measure in pixel units.

In a first experiment we consider the synthetic sphere sequence of Huguet and Devernay [12] (*http://devernay.free.fr/vision/varsceneflow/*), which is composed of four 512×512 images of a textured sphere with rotating hemispheres. Despite the fact that this sequence is rectified, and thus constitutes a special case with vanishing vertical components of the stereo flow, it is a good benchmark for comparison against existing techniques. Additionally it requires to estimate large stereo displacements which pose a challenge to variational methods. In this context we follow the idea of [24] and [12], and initialise $(u_\mathrm{st}, v_\mathrm{st})$ with a dedicated method for large displacements. To this end, we use a variant of the recent optical flow technique of [1] with constraint normalisation and SIFT matches [15] as prior. For consistency we also included results for initialisation with the belief propagation algorithm of [8], as used by Huguet and Devernay. However, this initialisation is only applicable for rectified images.

Table 1 compares our results with those of the variational methods of Huguet and Devernay [12] and Wedel *et al.* [24] and lists the errors computed within the sphere. With a substantial improvement in the RMSE for $(u_\mathrm{f}, v_\mathrm{f}, u_\mathrm{d}, v_\mathrm{d})$ and in the AAE we consistently outperform the other approaches for the scene flow, although these methods are specifically tailored to the rectified case. The lower RMSE of the method of Wedel *et al.* for $(u_\mathrm{st}, v_\mathrm{st})$ is due to the fact it uses sparse stereo correspondences that do not provide results in occluded regions. However, the accuracy of their estimated scene flow is significantly lower than ours. This even holds if they use *ground truth* for the stereo correspondences. The good performance of our method is also reflected in the accurate estimation of the stereo geometry: We obtain a subpixel precision of $d_F = 0.019$.

In a second experiment we evaluate the performance for a general stereo geometry. To this end we generated a synthetic sequence of four frames with ground truth (available at *http://www.mia.uni-saarland.de/valgaerts/eccv10/sceneflow*). It is similar to the one of the previous experiment: A textured sphere with rotating hemispheres is positioned against a plane in the background as shown in Fig. 2. To demonstrate the benefits of the different design steps in our model we start from a variant that performs a joint regularisation of the flows as in [12] and does not include constraint normalisation. We then refine the model by subsequently adding the normalisation and the separate regulariser. Table 2 lists the progressively improving results. The errors are computed in the non-occluded regions of the whole image domain. The AAE is not listed because it is not defined for the zero flow in the background. In Fig. 2 the computed flow fields are shown together with the obtained occlusion scores, the 3D reconstruction and the scene flow. As one can see, the estimated displacements resemble the ground truth very well. Again, this is confirmed by a subpixel precision of $d_F = 0.021$ for the stereo geometry.

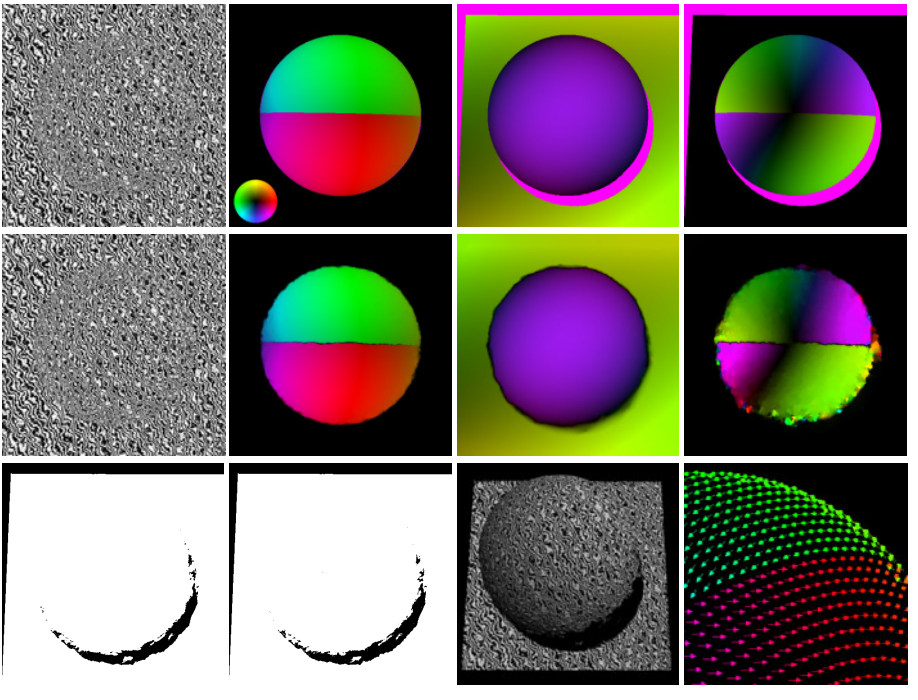


Fig. 2. Results for the general sphere sequence (image size 512×512). **Top Row: (a)** Left frame at first time step. **(b) + (c) + (d)** Ground truth of left optical flow, first stereo flow and flow change. Colour encodes the direction, brightness the magnitude (see colour circle). Occlusions are coloured pink. **Middle Row: (e)** Left frame at second time step. **(f) + (g) + (h)** Estimated left optical flow, first stereo flow and flow change. **Bottom Row: (i) + (j)** Estimated occlusion scores $o_{1\mathrm{r}}$ and $o_{2\mathrm{r}}$. **(k)** Estimated scene reconstruction. **(l)** Estimated scene flow. Runtime: ∼420 seconds. Parameters: $\alpha_1 = 1.5$, $\alpha_2 = 2$, $\alpha_3 = 0.8$, $\beta_1 = \beta_2 = 0.03$, $\gamma = 0.1$, $\mu = 1$.

Table 2. Evaluation of different variants of our method on the general sphere sequence

Method	RMSE $(u_\mathrm{f}, v_\mathrm{f}, u_\mathrm{d}, v_\mathrm{d})$	RMSE $(u_\mathrm{f}, v_\mathrm{f})$	RMSE $(u_\mathrm{st}, v_\mathrm{st})$
joint regularisation	0.67	0.64	2.08
joint regularisation + normalisation	0.63	**0.59**	1.86
separate regularisation + normalisation	**0.61**	**0.59**	**1.61**

Fig. 3. Results for real world sequences (image size 470×340). **Top Row: (a) + (b)** *Smiling*, left frames at consecutive time steps. **(c) + (d)** *Closing Mouth*, left frames at consecutive time steps. **Bottom Row: (e)** Reconstruction and overlayed scene flow for *Smiling*. Increasing magnitude from green to red. **(f)** Close-up *Smiling*. **(g)** Close-up *Closing Mouth*. Runtime: $\sim$260 seconds. Parameters: $\alpha_1 = 15$, $\alpha_2 = 20$, $\alpha_3 = 15$, $\beta_1 = \beta_2 = 0.5$, $\gamma = 30$, $\mu = 1$.

For our last experiment we have recorded two uncalibrated stereo sequences to test the performance of our method on real world data. The results are shown in Fig. 3 for the sequences *Smiling* and *Closing Mouth*. As one can verify in both cases the 3D structure and the motion of the face are captured well and look very realistic. We emphasise that these two results are obtained from only four frames. Additional real world results can be found at *http://www.mia.uni-saarland.de/valgaerts/eccv10/sceneflow*.

6 Conclusions

We have presented a general approach for the dense estimation of scene flow, scene structure and geometry from uncalibrated stereo sequences. Our contributions are threefold: (i) We generalise the classical four-frame case to arbitrary stereo setups by embedding epipolar constraints into a joint energy functional with data and smoothness terms. (ii) We introduce a tensor notation which allows us to normalise the data and stereo constraints such that they become geometrically interpretable. (iii) We present a separate robustfication of the smoothness terms to handle scenarios where flow discontinuities do not coincide. Our evaluation has demonstrated that the proposed approach is not only more general than existing methods but also more accurate: Even without explicitly knowing the stereo geometry, we outperform recent techniques that have been specifically designed for the rectified case. Furthermore, the stereo geometry is estimated with sub-pixel precision and reconstructions for real world data show that scene structure and motion are determined with high quality. This clearly demonstrates the benefit of a joint computation of flow, structure and geometry.

Acknowledgements. We gratefully acknowledge partial funding by the Deutsche Forschungsgemeinschaft (*WE 2602/6-1*), and by the International Max-Planck Research School. We thank Pascal Gwosdek, Jennifer Metzger and Sebastian Volz for their help.

References

1. Brox, T., Bregler, C., Malik, J.: Large displacement optical flow. In: Proc. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 41–48. IEEE Computer Society Press, Miami (2009)
2. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optic flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
3. Bruhn, A., Weickert, J.: Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In: Proc. Tenth International Conference on Computer Vision, vol. 1, pp. 749–755. IEEE Computer Society Press, Beijing (2005)
4. Carceroni, R.L., Kutulakos, K.N.: Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape and reflectance. International Journal of Computer Vision 49(2-3), 175–214 (2002)
5. Coons, S.A.: Surfaces for computer aided design of space forms. Tech. Rep. MIT/LCS/TR-41, Massachusetts Institute of Technology, Cambridge, MA (1967)
6. Courchay, J., Pons, J.P., Monasse, P., Keriven, R.: Dense and accurate spatio-temporal multi-view stereovision. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) Computer Vision – ACCV 2009. LNCS, vol. 5995, pp. 11–22. Springer, Heidelberg (2010)
7. Faugeras, O., Luong, Q.T., Papadopoulo, T.: The Geometry of Multiple Images. MIT Press, Cambridge (2001)
8. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. International Journal of Computer Vision 40(1), 41–54 (2006)
9. Furukawa, Y., Ponce, J.: Dense 3d motion capture from synchronized video streams. In: Proc. 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Anchorage (2008)

10. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
11. Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
12. Huguet, F., Devernay, F.: A variational method for scene flow estimation from stereo sequences. In: Proc. Eleventh International Conference on Computer Vision. IEEE Computer Society Press, Rio de Janeiro (2007)
13. Isard, M., MacCormick, J.: Dense motion and disparity estimation via loopy belief propagation. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3852, pp. 32–41. Springer, Heidelberg (2006)
14. Longuet-Higgins, H.C.: A computer algorithm for reconstructing a scene from two projections. Nature 293, 133–135 (1981)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
16. Luong, Q.T., Faugeras, O.D.: The fundamental matrix: theory, algorithms, and stability analysis. International Journal of Computer Vision 17(1), 43–75 (1996)
17. Mémin, E., Pérez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. IEEE Transactions on Image Processing 7(5), 703–719 (1998)
18. Min, D.B., Sohn, K.: Edge-preserving simultaneous joint motion-disparity estimation. In: Proc. 18th International Conference on Pattern Recognition, Hong Kong, pp. 74–77 (2006)
19. Patras, I., Alvertos, N., Tziritas, G.: Joint disparity and motion field estimation in steroscopic image sequences. In: Proc. 13th International Conference on Pattern Recognition, Vienna, Austria, vol. 1, pp. 359–362 (1996)
20. Pons, J.P., Keriven, R., Faugeras, O.D.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. International Journal of Computer Vision 72(2), 179–193 (2007)
21. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
22. Valgaerts, L., Bruhn, A., Weickert, J.: A variational model for the joint recovery of the fundamental matrix and the optical flow. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 314–324. Springer, Heidelberg (2008)
23. Vedula, S., Baker, S., Rander, P., Collins, R.T., Kanade, T.: Three-dimensional scene flow. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(3), 475–480 (2005)
24. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 739–751. Springer, Heidelberg (2008)
25. Wedel, A., Vaudrey, T., Meissner, A., Rabe, C., Brox, T., Franke, U., Cremers, D.: An evaluation approach for scene flow with decoupled motion and position. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) Dagstuhl Seminar. LNCS, vol. 5604, pp. 46–69. Springer, Heidelberg (2009)
26. Zhang, Y., Kambhamettu, C.: On 3d scene flow and structure estimation. In: Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 778–785. IEEE Computer Society Press, Kauai (2001)
27. Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., Seidel, H.P.: Complementary optic flow. In: Cremers, D., Boykov, Y., Blake, A., Schmidt, F.R. (eds.) EMMCVPR 2009. LNCS, vol. 5681, pp. 207–220. Springer, Heidelberg (2009)

Dense, Robust, and Accurate Motion Field Estimation from Stereo Image Sequences in Real-Time

Clemens Rabe, Thomas Müller, Andreas Wedel, and Uwe Franke

Daimler Research, Sindelfingen

Abstract. In this paper a novel approach for estimating the three dimensional motion field of the visible world from stereo image sequences is proposed. This approach combines dense variational optical flow estimation, including spatial regularization, with Kalman filtering for temporal smoothness and robustness. The result is a dense, robust, and accurate reconstruction of the three-dimensional motion field of the current scene that is computed in real-time. Parallel implementation on a GPU and an FPGA yields a vision-system which is directly applicable in real-world scenarios, like automotive driver assistance systems or in the field of surveillance. Within this paper we systematically show that the proposed algorithm is physically motivated and that it outperforms existing approaches with respect to computation time and accuracy.

1 Introduction

Estimating the three-dimensional motion vector field from stereo image sequences remains as one of the fundamental computational challenges and is a key task in computer vision. Different variants of this problem arise in the estimation of ego motion [1], object motion [2], human motion [3], and motion segmentation [4]. Knowledge of the surrounding motion field is the basis for a wide range of technical applications, e.g. in automotive driver assistance systems or in the field of surveillance. Therefore, and especially in safety relevant applications, robustness, density of information, accuracy as well as real-time capability are of utmost importance.

Some approaches known from literature use sparse feature-based tracking methods and increase the robustness by temporal integration using filters. Others apply space-related regularizers for a dense computation from only two consecutive frames. In this paper, we combine such a dense variational approach on the image domain with Kalman filters at every single pixel to establish temporal smoothness of the dense three-dimensional motion field.

In this paper, we present a new algorithm called *Dense6D*, that estimates the motion field by fusing dense stereo and optical flow information. A real world example of such an estimate is shown in Fig. 1. An improvement in accuracy and robustness with respect to standard approaches known from literature is also achieved with *Variational6D*, which replaces the optical flow component

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 582–595, 2010.

Fig. 1. *left:* typical traffic scene. *right:* motion field, estimated by the Dense6D algorithm proposed in this paper. The color encodes the velocity (from green to red) of the observed points.

of Dense6D by a variational scene flow method. Throughout the paper, the term *scene flow* will denote the three-dimensional motion field consisting of the optical flow and the disparity change along the optical flow vectors between two consecutive frames.

1.1 Related Work

Due to the importance of the problem, a lot of different approaches to image based motion field estimation have been proposed in the last three decades. Most of them can be classified into the following main strategies:

- model based approaches
- sparse feature tracking methods using multiple image frames
- dense scene flow computation from two consecutive frames

The estimation of motion vectors involves the reconstruction of the three dimensional scene via stereo matching and the estimation of point correspondences between two or more consecutive images. Both problems are classical ill-posed problems in the sense that merely matching of similar intensities will typically not give rise to a unique solution. The three mentioned strategies choose different ways to overcome the ill-posedness.

The model based approaches, like in [3] or [2] use physically constrained object or human models to make the problem well-posed; however, the need of a model disqualifies the model based approaches in a large variety of situations.

The feature tracking and scene flow appraoches use regularization to make the problem well-posed. This regularization is either formulated in the time domain for the tracking of features in [5] or [6] or in the spatial domain, imposing smoothness of the motion field between two consecutive frames like in [7] or [8]. In this paper we revisit both, feature tracking and dense scene flow computation,

and suggest the use of Kalman filters for every image pixel to reconstruct a dense and robust three-dimensional motion field of the depicted scene.

Scene flow computation methods are mainly based on the classic optical flow algorithm by Horn and Schunck in [9], where the flow field is computed as the minimizer of an energy functional that assumes constant image intensities and a smooth flow field. This framework has been improved in [10] to cope with flow discontinuities and outliers and in [11] to cope with large flow vectors. In recent years, several real-time optical flow methods have been proposed, e.g. in [12], [13].

Joint motion and disparity estimation for the scene flow computation was introduced in [14]. In [8] the motion and disparity estimation steps were decoupled in order to achieve real-time capability without loosing accuracy.

On the other hand, the application of Kalman filters [15] in real-time motion field estimation was proposed in [16] and later as *6D-Vision* in [17], using the well-known KLT-tracker [5] and a dense stereo disparity field as input. However, this method only yields sparse information. To build a vision system which provides a dense, robust and accurate motion field in real-time, we suggest to replace the tracker by a dense variational optical or scene flow algorithm. Despite the computational complexity and the large volume of data, special computation schemes for the filtering process, implemented on modern graphic processing units (GPUs), are used to ensure real-time capability (25 Hz in our implementation).

1.2 Outline

In Chapter 2 we will shortly revisit some approaches to optical flow and scene flow computation, which are then used as input for the filtering process. In Chapter 3 we will introduce the concept of Kalman filters for motion estimation and the novel dense filtered motion estimation approaches based on stereo, optical flow and scene flow. Chapter 4 will experimentally demonstrate the new approaches and systematically evaluate the gain in robustness and accuracy over existing methods. Chapter 5 concludes this contribution with a summary and an outlook on future work.

2 Two-Frame Motion Field Estimation

2.1 Combination of Optical Flow and Stereo

For the estimation of the motion field of the environment, we consider the two stereo image pairs $I_{\{1,2\}}^{\{L,R\}} : \Omega \rightarrow \mathbb{R}$ on the image domain $\Omega = \{\boldsymbol{x}\} \subset \mathbb{R}^2$. Throughout this paper, we will assume that the camera system is calibrated, and the taken images are preprocessed by a rectification module that performs a lens-correction and establishes a standard stereo configuration.

Having determined the optical flow $\boldsymbol{u} : \Omega \rightarrow \mathbb{R}^2$, an inverse transformation together with known depth at both time instants can yield the three-dimensional motion field information of actual interest.

When searching the whole consecutives image for corresponding single gray values, so that (with $\boldsymbol{u} \equiv \boldsymbol{u}(\boldsymbol{x})$)

$$\rho(\boldsymbol{x}, \boldsymbol{u}) = I_2(\boldsymbol{x} + \boldsymbol{u}) - I_1(\boldsymbol{x}) = 0\,, \tag{1}$$

this obviously leads to an ill-posed problem. In [9] Horn and Schunck proposed to overcome this by minimizing a global energy functional on the whole image domain, consisting of a data consistency term and a regularization term,

$$E[\boldsymbol{u}] = \int_{\Omega} \left\{ \lambda \left| \rho(\boldsymbol{x}, \boldsymbol{u}(\boldsymbol{x})) \right|^n + \sum_{i=x,y} \left| \boldsymbol{\nabla} u_i(\boldsymbol{x}) \right|^n \right\} \mathrm{d}\boldsymbol{x} \tag{2}$$

with $n = 2$. The parameter $\lambda \in \mathbb{R}^+$ weights between the data and the regularization. This method leads to dense, accurate results and yields real-time performance on modern hardware.

The case $n = 2$ is quite simple to compute, but suffers from blurring effects around flow edges and over-weights outliers. Therefore, we use the computation method proposed by Wedel et al. in [18] based on [13], where $n = 1$ leads to improved results. This method solves the two terms in Eq. (2) by introducing an additional coupling term and an iterative solution scheme on a coarse-to-fine grid. The data term part is solved directly point-wise by a thresholding step, while the regularizing smoothness term is solved by a dual approach, proposed by Chambolle in [19].

In our work, the dense semi-global-matching (SGM) method by Hirschmüller [20] is used for the estimation of the disparity images $d_{\{1,2\}} : \Omega \rightarrow \mathbb{R}^+$ (see Fig. 2). The algorithm is available on dedicated parallel hardware (FPGA). Therefore, the disparity computation does not effect the real-time performance of our motion field estimation in a negative way.

With the knowledge of the optical flow field $\boldsymbol{u}(\boldsymbol{x})$ between two consecutive frames and both depth images $Z_{\{1,2\}}(\boldsymbol{x}) \sim 1/d_{\{1,2\}}(\boldsymbol{x})$, the three-dimensional

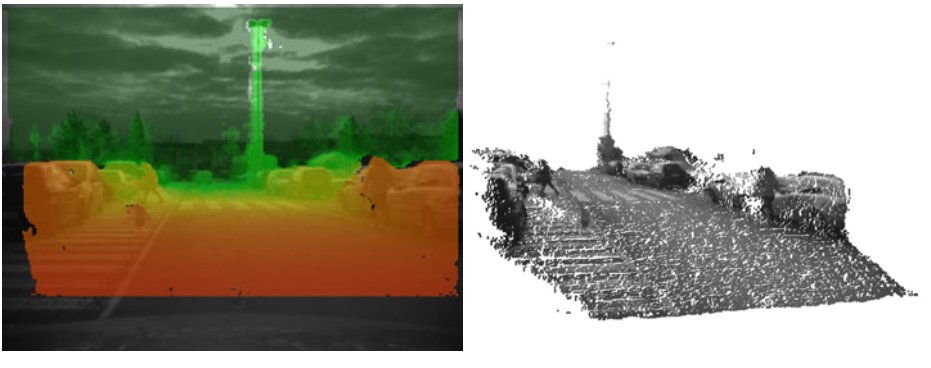


Fig. 2. *left:* traffic scene with SGM stereo computation. The color encodes the distance from near (red) to far (green). *right:* three-dimensional visualization of the corresponding scene.

motion field can be determined. However, this straight-forward differential approach usually leads to insufficient results, due to noisy depth measurements.

2.2 Variational Scene Flow

Noisy depth measurements lead to a noisy motion field estimation along the optical rays. To reduce this noise, we use a global variational approach where the disparity change is regularized together with the optical flow [8]. A decoupled depth measurement is used for the first frame and yields the disparity field $d_1(\boldsymbol{x})$, related to the left image, while the disparity change $\dot{d}: \Omega \rightarrow \mathbb{R}$ is estimated via a global optimization scheme together with the optical flow $\boldsymbol{u}$. The functional to minimize is defined as

$$E\left[\boldsymbol{u}, \dot{d}\right] = \int_{\Omega} \left\{ R\left(\boldsymbol{u}(\boldsymbol{x}), \dot{d}(\boldsymbol{x})\right) + \sum_{i=x,y} \left|\boldsymbol{\nabla} u_i(\boldsymbol{x})\right| + \left|\boldsymbol{\nabla} \dot{d}(\boldsymbol{x})\right| \right\} \mathrm{d}\boldsymbol{x} \tag{3}$$

with the data term

$$R\left(\boldsymbol{u}, \dot{d}\right) = \lambda_{\mathrm{L}} \left|\rho_{\mathrm{L}}(\boldsymbol{u})\right| + \lambda_{\mathrm{R}} \left|\rho_{\mathrm{R}}\left(\boldsymbol{u}, \dot{d}\right)\right| + \lambda_2 \left|\rho_2\left(\boldsymbol{u}, \dot{d}\right)\right| \tag{4}$$

and the residuals (with $d_1 \equiv d_1(\boldsymbol{x})$ and the unity vector in x direction $\boldsymbol{e}_x$)

$$\rho_{\mathrm{L}}(\boldsymbol{u}) = I_2^{\mathrm{L}}(\boldsymbol{x} + \boldsymbol{u}) - I_1^{\mathrm{L}}(\boldsymbol{x}) \tag{5}$$

$$\rho_{\mathrm{R}}\left(\boldsymbol{u}, \dot{d}\right) = I_2^{\mathrm{R}}\left(\boldsymbol{x} + \boldsymbol{u} - \left(d_1 + \dot{d}\right)\boldsymbol{e}_x\right) - I_1^{\mathrm{R}}(\boldsymbol{x} - d_1 \boldsymbol{e}_x) \tag{6}$$

$$\rho_2\left(\boldsymbol{u}, \dot{d}\right) = I_2^{\mathrm{R}}\left(\boldsymbol{x} + \boldsymbol{u} - \left(d_1 + \dot{d}\right)\boldsymbol{e}_x\right) - I_2^{\mathrm{L}}(\boldsymbol{x} + \boldsymbol{u})\ . \tag{7}$$

For the numerical computation, the data term and the regularizations are coupled by an additional term to establish an iterative solution scheme with a coarse-to-fine approach. The data term can then be solved point-wise by implementing $|x| \approx \sqrt{x^2 + \varepsilon}, \varepsilon \ll 1$, linearizing the residuals in Eqs. (5) - (7) and performing gradient descend steps. The regularization terms in the optical flow $\boldsymbol{u}$ as well as in the disparity change $\dot{d}$ are solved by the former mentioned dual approach. This method provides better results, compared to the approach with optical flow and stereo, but demands higher computational costs.

3 Temporal Integration of the Motion Field

To increase the robustness and accuracy of the estimated motion field, we suggest a temporal integration using Kalman filters. In this section, the underlying Kalman filter model is explained in detail for the proposed Dense6D and Variational6D algorithms.

3.1 Model

In our pinhole stereo camera system, the 3d structure of the observed scene is immediately reconstructed by a stereo algorithm. In this configuration, the left image point $\boldsymbol{x} = (x, y)^\top$ is the projection of a world point $\tilde{\boldsymbol{X}} = (X, Y, Z, 1)^\top$. Expressed in homogeneous coordinates, this relation is given by

$$\begin{pmatrix} x & y & d & 1 \end{pmatrix}^\top \backsimeq \boldsymbol{\Pi} \cdot \tilde{\boldsymbol{X}} \tag{8}$$

with the positive disparity $d \equiv d(\boldsymbol{x})$. The extended projection matrix $\boldsymbol{\Pi}$ is given as

$$\boldsymbol{\Pi} = \begin{pmatrix} f_x & 0 & x_0 & 0 \\ 0 & f_y & y_0 & 0 \\ 0 & 0 & 0 & b \cdot f_x \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{R}_\mathrm{c} & \boldsymbol{t}_\mathrm{c} \\ \mathbf{0}^\top & 1 \end{pmatrix} \tag{9}$$

with f_x and f_y as the focal lengths in pixel, $(x_0, y_0)^\top$ as the principal point in pixel and b as the base width of the stereo camera system. The rotation matrix $\mathbf{R}_\mathrm{c}$ and the translation vector $\boldsymbol{t}_\mathrm{c}$ describe the extrinsic orientation of the camera system to the world coordinate system. To determine the three-dimensional world position for an observed image point $\boldsymbol{x}$ with known disparity d, Eq. (8) has to be inverted.

Having established a correspondence over time for an observed image point by an optical flow or feature tracking algorithm, the 3d motion can be calculated directly from the reconstructed 3d points. However, such an approach suffers heavily from the immanent measurement noise and thus does not yield robust results. Therefore, we present here a method to estimate the 3d position and 3d motion of a point using a Kalman filter [15]. Due to the recursive nature of the Kalman filter, the estimation is improved continuously with each measurement, by updating the state vector and its associated covariance matrix. This eliminates the need to save a history of measurements and is computationally

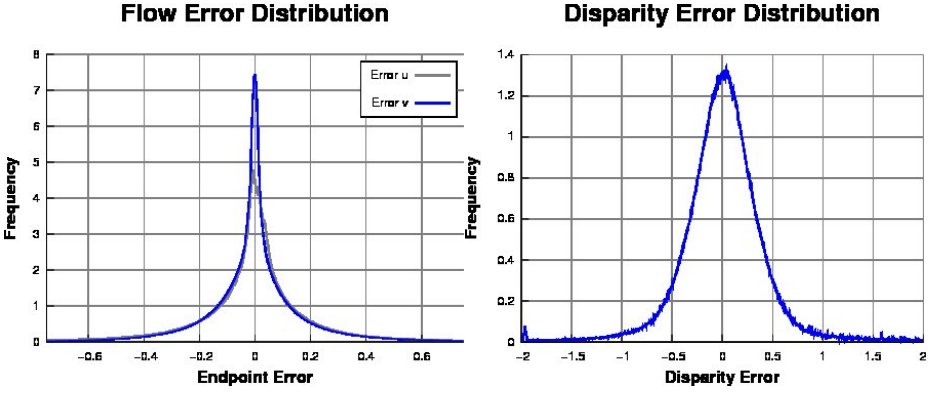


Fig. 3. *left:* Dense optical flow error distribution, *right:* SGM stereo error distribution, both related to ground truth data from synthetic sequences

highly efficient. Additionally, Kalman filters provide measurement uncertainties which can be regarded when the motion field is evaluated for further applications. Looking at the error distributions of the input data shown in Fig. 3, the application of the Kalman filter is justified.

The state vector of the Kalman filter is defined as $\boldsymbol{\xi} = \left(X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}\right)^\top$, the combination of the 3d position and the 3d velocity vector. The system model describes the propagation of the state vector $\boldsymbol{\xi}_t$ of the previous time step $t-1$ to the current one t and assumes a linear motion. It is given by the linear equation system

$$\tilde{\boldsymbol{\xi}}_t = \begin{pmatrix} \mathbf{R}_\mathrm{e} & \Delta t \cdot \mathbf{R}_\mathrm{e} \\ \mathbf{0} & \mathbf{R}_\mathrm{e} \end{pmatrix} \boldsymbol{\xi}_{t-1} + \begin{pmatrix} \boldsymbol{t}_\mathrm{e} \\ \mathbf{0} \end{pmatrix} \tag{10}$$

with $\mathbf{R}_\mathrm{e}$ and $\boldsymbol{t}_\mathrm{e}$ as the rotation and the translation components of the inverse motion of the observer, and Δt as the time between both frames.

The measurement model of the Kalman filter describes the relation between the measurement vector $\boldsymbol{z} = (x, y, d)^\top$ and the state vector $\boldsymbol{\xi}$. Here, only the position components of the state vector are directly measured, and the relation between the measured projection $\boldsymbol{z}$ and the reconstructed 3d point is given by Eq. (8). Since the measurement model must be formulated in the euclidean space rather than the projective space, the measurement model is non-linear:

$$\tilde{\boldsymbol{z}} = w \cdot \begin{pmatrix} x & y & d & 1 \end{pmatrix}^\top = \boldsymbol{\Pi} \cdot \begin{pmatrix} X & Y & Z & 1 \end{pmatrix}^\top \tag{11}$$

$$\boldsymbol{z} = \frac{1}{w} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \tilde{\boldsymbol{z}} \tag{12}$$

Therefore, extended Kalman filters have to be applied.

3.2 Filtered Tracks and Stereo: 6D-Vision

In [6], 2000 Kanade-Lucas-Tomasi (KLT) features are used to generate measurements which are temporally integrated by Kalman filters with the model equations mentioned previously. The current measurement vector $\boldsymbol{z}_t$ is determined by

$$\boldsymbol{z}_t = \begin{pmatrix} \boldsymbol{x}_t \\ d_t\left(\boldsymbol{x}_t\right) \end{pmatrix}, \quad \boldsymbol{x}_t = \boldsymbol{x}_{t-1} + \boldsymbol{u}\left(\boldsymbol{x}_{t-1}\right) \tag{13}$$

with $\boldsymbol{u}\left(\boldsymbol{x}_{t-1}\right)$ as the optical flow related to the previous position $\boldsymbol{x}_{t-1}$ of the feature, computed for example by the Lucas-Kanade method [5], and the corresponding disparity $d_t(\boldsymbol{x}_t)$ at the new image position $\boldsymbol{x}_t$.

Note that $\boldsymbol{x}_{t-1}$ in Eq. (13) depicts the old *measured* image position at the previous frame, not the projection of the filtered state $\boldsymbol{\xi}_{t-1}$. That means the image position of the features is only determined by the feature tracker, while the filtering only influences the velocity and the disparity estimation. This way, undesired low pass filtering effects of the Kalman filter are avoided. Together

with a multiple-filter approach, that reduces the settling time of a filter by running multiple differently initialized filters in parallel, the 6D-Vision motion field estimation method provides robust results in real-time, even in real world scenarios.

3.3 Filtered Dense Optical Flow and Stereo: Dense6D

The information provided by the 6D-Vision approach is only sparse. However, to utilize as much information as possible from a stereo image sequence, we replace the feature tracker in the measurement step by a dense optical flow algorithm (*Dense6D*). Modern parallel hardware, an NVIDIA graphics adapter with CUDA capability in our implementation, together with sophisticated numerical computation schemes at the filtering process, enables us to assign Kalman filters to every single pixel of the input image sequence (of the size 640 px × 480 px) and to apply them in real-time (at 25 Hz). For numerical stability, the implementation is based on the well-known U-D factorization proposed by Bierman et al. [21]. A code generator takes advantage of the sparse measurement matrix and produces the actual CUDA implementation.

At the beginning of the computation step from image I_{t-1} to I_t, every pixel $\boldsymbol{x}_{t-1}$ on the discrete pixel grid is associated with one Kalman filter $\mathcal{K}_{t-1}(\boldsymbol{x}_{t-1})$ and one sub-pixel component $\boldsymbol{s}_{t-1}(\boldsymbol{x}_{t-1})$) After having determined the dense optical flow field from I_{t-1} to I_t, and after having updated the filters during the filtering step, $\mathcal{K}_{t-1}(\boldsymbol{x}_{t-1}) \rightarrow \mathcal{K}_t(\boldsymbol{x}_{t-1})$, the updated Kalman filter field $\mathcal{K}_t(\boldsymbol{x}_{t-1})$ must be warped along the sub-pixel accurate optical flow $u(\boldsymbol{x}_{t-1})$, to receive the filter field $\mathcal{K}_t(\boldsymbol{x}_t)$ on the new discrete pixel positions $\boldsymbol{x}_t$. The updates of the positions and the sub-pixel components are given by

$$\boldsymbol{x}_t = \lfloor \boldsymbol{x}_{t-1} + \boldsymbol{s}_{t-1}(\boldsymbol{x}_{t-1}) + \boldsymbol{u}(\boldsymbol{x}_{t-1}) + 0.5 \text{ px} \rfloor \tag{14}$$

$$\boldsymbol{s}_t(\boldsymbol{x}_t) = [\boldsymbol{s}_{t-1}(\boldsymbol{x}_{t-1}) + u(\boldsymbol{x}_{t-1}) + 0.5 \text{ px}] \bmod 1 \text{ px} - 0.5 \text{ px} \tag{15}$$

At every time step the sub-pixel component is updated due to the sub-pixel accurate optical flow, which is always taken from the discrete position of the pixel grid, since exact optical flow information is only available at these points.

During the resampling step it is possible that not every pixel $\boldsymbol{x}_t$ of the current image is referred by a flow vector $\boldsymbol{u}(\boldsymbol{x}_{t-1})$. In this case a new filter has to be created with predefined initial values and associated with the empty pixel. Another option is to initialize the filter based on the states and the covariances of the surrounding filters.

On the other hand, if one pixel $\boldsymbol{x}_t$ of the current image is referred by more than one flow vectors $\boldsymbol{u}(\boldsymbol{x}_{t-1})$, one either has to decide which one of the filters will be used with the corresponding pixel for the next frame, or has to combine them to a new one. In this case, the covariances of the concurring filters can weight between them. It is also reasonable to use the depth information, so that the filter with the smallest Z value in the filter state can survive, while the other ones are deleted.

For performance reasons, our implementation does not perform such an extended analysis, but generates the target image on a first-come first-serve basis.

3.4 Filtered Variational Scene Flow

If the dense optical flow method is replaced by a variational scene flow scheme as proposed in Sec. 2.2, the estimation of the disparity change $\dot{d}(\boldsymbol{x})$ from Eq. (3) can be used as an additional measurement. In this case, the measurement vector for the Kalman filter is

$$\boldsymbol{z}_t = \begin{pmatrix} \boldsymbol{x}_t \\ d_t(\boldsymbol{x}_t) \\ d_{t-1}(\boldsymbol{x}_{t-1}) + \dot{d}(\boldsymbol{x}_{t-1}) \end{pmatrix}, \tag{16}$$

while the new matrix

$$\mathbf{H} = \begin{pmatrix} f_x & 0 & x_0 & 0 \\ 0 & f_y & y_0 & 0 \\ 0 & 0 & 0 & b \cdot f_x \\ 0 & 0 & 0 & b \cdot f_x \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{R}_c & \boldsymbol{t}_c \\ \mathbf{0}^\top & 1 \end{pmatrix} \tag{17}$$

replaces the extended projection matrix $\mathbf{\Pi}$ in Eq. (11). The Kalman filter weights between the two disparity measurements regarding the measurement covariance matrix.

4 Evaluation

In our experiments, we compare the following motion field estimation techniques described in this paper:

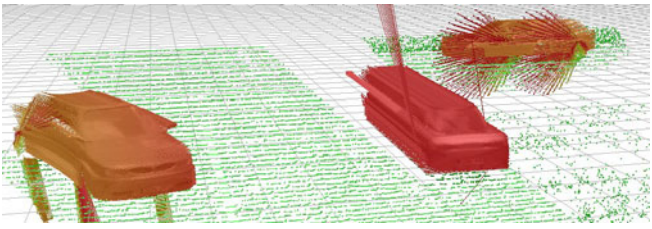
1. Differential motion field estimation from optical flow and stereo (Sec. 2.1)
2. Variational scene flow from two frames (Sec. 2.2)
3. the Kalman filtered method, using dense optical flow and stereo (Dense6D, introduced in Sec. 3.3)
4. the filtered variational scene flow approach (Variational6D, introduced in Sec. 3.4).

4.1 Evaluation with Ground Truth Information

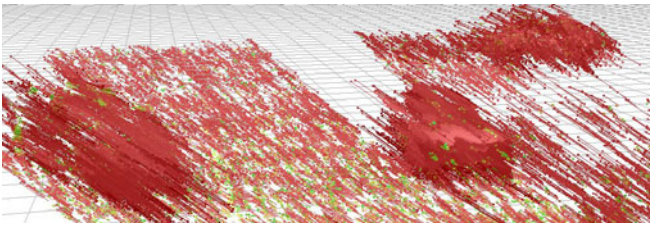
In the first experimental part, we analyze our vision system on a synthetic stereo image sequence rendered with Povray [22]. The experiments are conducted on a sequence with an image resolution of 640 px × 480 px × 12 bit and 150 frames.

The evaluation platform consists of an Intel Quad-Core 3 GHz processor and an NVIDIA GeForce 285 GTX graphics adapter. On this configuration, the dense optical flow calculation is performed in 24 ms, whereas the dense scene flow computation takes 65 ms. The 640 × 480 Kalman filters are processed in 12 ms. This enables us to achieve a framerate of 25 Hz for the Dense6D algorithm and about 10 Hz for the Variational6D approach.

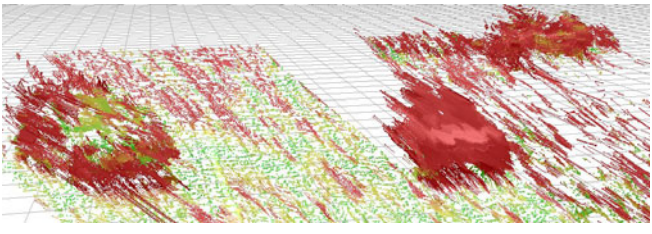
In our exemplary sequence, the camera moves through an artificial traffic scene containing crossing and turning vehicles. Fig. 4 shows the motion vector fields for



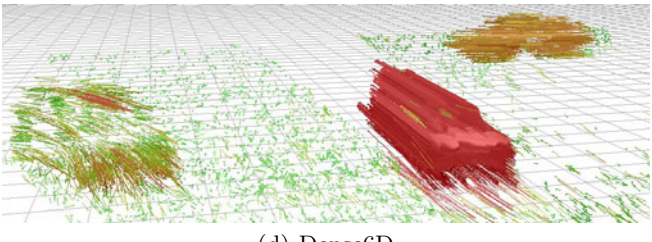
(a) Ground truth



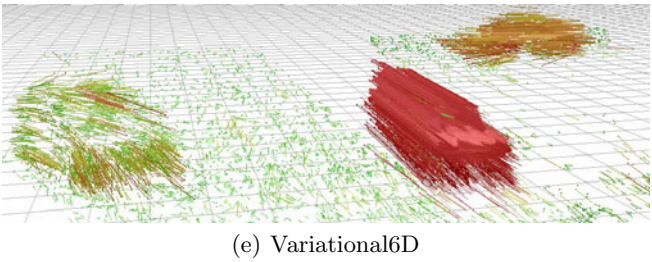
(b) Direct combination of optical flow and stereo



(c) Scene flow

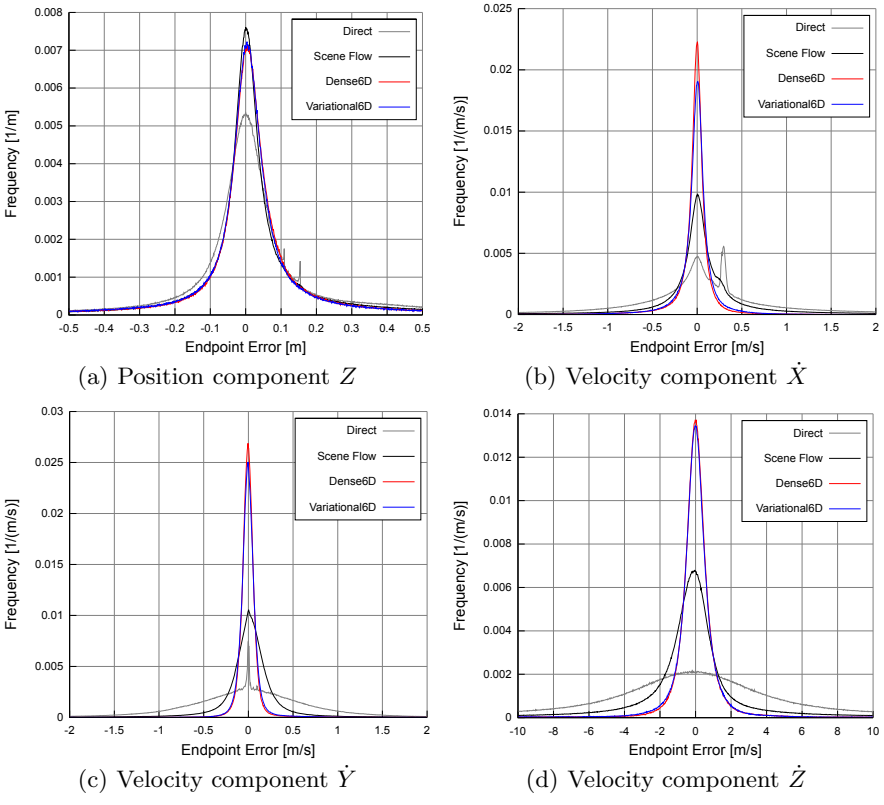


(d) Dense6D



(e) Variational6D

Fig. 4. Estimated motion field of the described methods. The color encodes the velocity: green encodes 0.0 $^{m}/_{s}$, red encodes 8.0 $^{m}/_{s}$. The vectors show the predicted 3d position in 0.250 s (Figures (a), (d), (e)) resp. 0.050 s (Figures (b), (c)).



(a) Position component Z (b) Velocity component $\dot{X}$

(c) Velocity component $\dot{Y}$ (d) Velocity component $\dot{Z}$

Fig. 5. Error distributions of the Z position and the velocity components calculated from the direct combination of optical flow and stereo (gray), the scene flow (black), the Dense6D method (red), and the Variational6D method (blue)

Table 1. Median error (ME) and root mean square error (RMS) of the Z position and velocity components of the four evaluated methods

	Z [m]		$\dot{X}$ [m/s]		$\dot{Y}$ [m/s]		$\dot{Z}$ [m/s]	
Method	ME	RMS	ME	RMS	ME	RMS	ME	RMS
Direct	0.0010	2.749	0.0462	42.0093	0.0004	15.370	0.4374	141.442
Scene Flow	0.0080	2.807	0.0179	22.7186	0.0172	11.470	-0.1173	67.520
Dense6D	0.0104	1.068	-0.0065	0.3623	-0.0044	0.339	0.0107	2.538
Variational6D	0.0085	1.282	-0.0007	0.3712	-0.0040	0.319	-0.0044	2.537

one frame of the sequence. The left vehicle performs a turning maneuver, while the remaining two cars are moving linearly. The vehicle in the middle moves at a constant speed, whereas the car coming from the right performs a constant deceleration. Obviously, the Dense6D and Variational6D algorithms outperform the differential approaches.

Since the three-dimensional ground truth position and motion fields are available, the error distributions for $Z, \dot{X}, \dot{Y}$ and $\dot{Z}$ can be determined. Accumulated over the whole image Ω and the whole sequence $[0, T]$, the error distributions are shown in Fig. 5. In addition, the median (ME) of the error distribution and the root mean squared (RMS) error is computed for the quantities $Z, \dot{X}, \dot{Y}$ and $\dot{Z}$.

One can clearly see from Fig. 5 and Tab. 1 that the proposed Dense6D algorithm outperformes the scene flow computation method with respect to accuracy and robustness.

Overall, we are not able to detect significant advantages regarding the accuracy of one of the new proposed algorithms against the other one in this paper. Therefore, due to the less complex computation scheme of the Dense6D algorithm against Variational6D, we suggest to use the first one in future motion field estimation tasks.

However, as one can see from the results presented in this chapter, both new approaches outperform dense real-time methods known from literature by far.

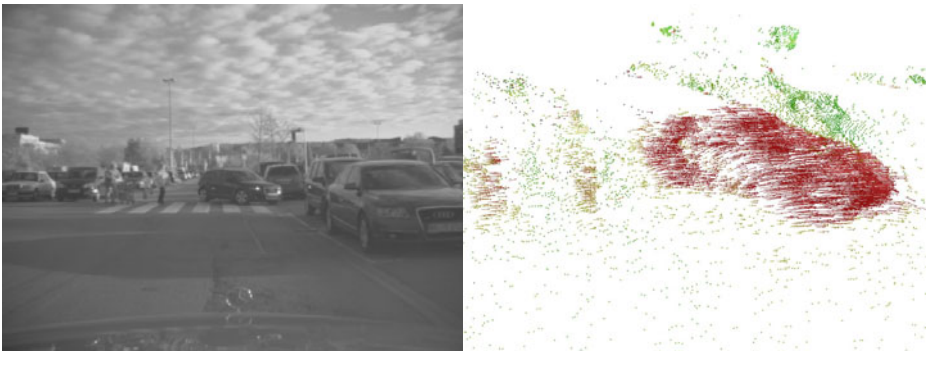


Fig. 6. *left:* typical traffic scene. *right:* corresponding 3d motion field, estimated by the Dense6D algorithm proposed in this paper. The color encodes the velocity (from green to red) of the observed points.

4.2 Real World Results

The two proposed new estimation methods are directly applicable in real-world scenarios, being able to build the basis for robust reliable object detection and segmentation. Fig. 6 shows the estimated motion vector field of a turning vehicle at a distance of about 30 m. Here, the observer was moving at a speed of about 3 $^{\mathrm{m}}/_{\mathrm{s}}$. In Fig. 7, the motion field of multiple pedestrians is shown. Here, the observer was also moving at about 1 $^{\mathrm{m}}/_{\mathrm{s}}$, including a strong turning maneuver. For visualization purposes, the motion of the camera was compensated using inertial sensor data. Dense6D is currently implemented in our research car and can become a key element in future safety driver assistance systems.

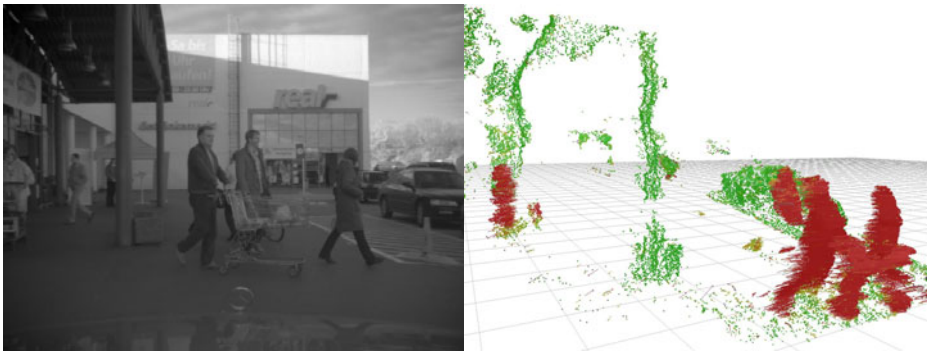


Fig. 7. *left:* typical traffic scene. *right:* corresponding 3d motion field, estimated by the Dense6D algorithm proposed in this paper. The color encodes the velocity (from green to red) of the observed points.

5 Conclusions

In this paper, we have proposed two approaches to dense, robust, and accurate motion field estimation in real-time. We have combined dense variational optical or scene flow estimation techniques with Kalman filters, assuming a linear motion model. Evaluation of the relevant error quantities compared to synthetic ground truth data show that these approaches lead to far better results in real-time than known by literature so far. This is apparently similar in real world scenarios.

The next future work will include a multi-filter implementation on the GPU and the consideration of flow uncertainties in the filtering process.

References

1. Badino, H.: A robust approach for ego-motion estimation using a mobile stereo platform. In: Jähne, B., Mester, R., Barth, E., Scharr, H. (eds.) IWCM 2004. LNCS, vol. 3417, pp. 198–208. Springer, Heidelberg (2004)
2. Barth, A., Franke, U.: Where will the oncoming vehicle be the next second? In: Intelligent Vehicles Symposium. IEEE, Los Alamitos (2008)
3. Rosenhahn, B., Brox, T., Weickert, J.: Three-dimensional shape knowledge for joint image segmentation and pose tracking. International Journal of Computer Vision 73, 243–262 (2007)
4. Wedel, A., Pock, T., Braun, J., Franke, U., Cremers, D.: Duality tv-l1 flow with fundamental matrix prior. In: Image and Vision Computing, Auckland, New Zealand (2008)
5. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. Seventh International Joint Conference on Artificial Intelligence, Vancouver, Canada (1981)
6. Rabe, C., Franke, U., Gehrig, S.: Fast detection of moving objects in complex scenarios. In: Intelligent Vehicles, Istanbul, Turkey, DaimerChrysler AG, pp. 398–403 (2007)

7. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. In: International Conference on Computer Vision (1999)
8. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparce or dense stereo data. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 196–209. Springer, Heidelberg (2008)
9. Horn, B.K.P., Schunk, B.G.: Determining optical flow. Artificial Intelligence 17, 185–203 (1981)
10. Memin, E., Perez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. IEEE Transactions on Image Processing 7, 703–719 (1998)
11. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
12. Bruhn, A., Weickert, J.: Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In: Proc. Tenth International Conference on Computer Vision, vol. 1, pp. 749–755 (2005)
13. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l1 optical flow. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 214–223. Springer, Heidelberg (2007)
14. Patras, I., Hendriks, E., Tziritas, G.: A joint motion/disparity estimation method for the construction of stereo interpolated images in stereoscopic image sequences. In: Proc. 3rd Annual Conference of the Advanced School of Computing and Imaging, Heijen, The Netherlands (1997)
15. Kalman, R.E.: A new approach to linear filtering and prediction problems. ASME-Journal of Basic Engineering 82, 35–45 (1960)
16. Franke, U., Rabe, C.: Kalman filter based depth from motion with fast convergence. In: Proc. of the 2005 IEEE Intelligent Vehicles Symposium (2005)
17. Franke, U., Rabe, C., Badino, H., Gehrig, S.: 6d-vision: Fusion of stereo and motion for robust environment perception. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 216–223. Springer, Heidelberg (2005)
18. Wedel, A., Cremers, D., Pock, T., Bischof, H.: Structure- and motion-adaptive regularization for high accuracy optical flow. In: International Conference on Computer Vision (2009)
19. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision 20, 89–97 (2004)
20. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: CVPR (2005)
21. Bierman, G.J.: Factorization Methods for Discrete Sequential Estimation. Academic Press Inc., London (1977)
22. Rabe, C., Vaudrey, T.: University of Auckland. enpeda. image sequence analysis test site, EISATS (2010), http://www.mi.auckland.ac.nz/EISATS

Estimation of 3D Object Structure, Motion and Rotation Based on 4D Affine Optical Flow Using a Multi-camera Array

Tobias Schuchert[1,2] and Hanno Scharr[2]

[1] Fraunhofer Institute of Optronics, System Technologies and Image Exploitation, Karlsruhe, Germany
tobias.schuchert@iosb.fraunhofer.de
[2] Institute for Chemistry and Dynamics of the Geosphere, ICG-3: Phytosphere, Forschungszentrum Jülich, Germany
h.scharr@fz-juelich.de

Abstract. In this paper we extend a standard affine optical flow model to 4D and present how affine parameters can be used for estimation of 3D object structure, 3D motion and rotation using a 1D camera grid. Local changes of the projected motion vector field are modelled not only on the image plane as usual for affine optical flow, but also in camera displacement direction, and in time. We identify all parameters of this 4D fully affine model with terms depending on scene structure, scene motion, and camera displacement. We model the scene by planar, translating, and rotating surface patches and project them with a pinhole camera grid model. Imaged intensities of the projected surface points are then modelled by a brightness change model handling illumination changes. Experiments demonstrate the accuracy of the new model. It outperforms not only 2D affine optical flow models but range flow for varying illumination. Moreover we are able to estimate surface normals and rotation parameters. Experiments on real data of a plant physiology experiment confirm the applicability of our model.

1 Introduction

Object structure and motion estimation from camera image sequences is a typical and well explored computer vision topic and many different solutions exist for different application prerequisites. We target at a typical plant physiology lab situation (see e.g. [1]), where e.g. growth, i.e. divergence of the motion vector field or curvature production in terms of spatial derivatives of the rotation vector field, of plant organs are parameters of interest. In order to analyse derivatives of the motion field, motion and structure of plant organs – here leaves of seedlings and small plants – need to be measured in high spatial (sub-millimetre) and temporal resolution (several minutes). Highest accuracy is thus a prerequisite here, as derivatives of the motion field are the final signal of interest and rigid motion of leaves is much larger than motion due to e.g. growth. Such measurements help unravelling bio-chemical processes underlying plant growth (see e.g. [2]) and

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 596–609, 2010.

thus give hints for seed, feed, and food production or plant breeding. However, calculation time is less of an issue, as analyses may be calculated off-line.

A typical lab setup uses a single camera on a moving stage looking downward on the plant, instead of using multiple cameras. The advantage of such a setup is that the moving stage allows to take images from many equidistant camera positions, typically at several mm or even sub-mm distances depending on object size. Further calibration needs only be done for a single camera and the camera may be moved away when not needed as plants should not be shaded by measurement equipment. One loop through all camera positions takes much shorter (seconds) than time between two acquisitions at the same position (minutes). Consequently we may regard the acquired data as if it came from a synchronized, fine spaced 1D grid of cameras. This camera grid equidistantly samples a 4D space spanned by the sensor coordinates x and y, camera position s and time t.

Such 4D data has already been exploited in literature (e.g. [3,4]). There 4D optical flow with affine components is calculated and all components are interpreted in terms of 3D translation, 3D position and surface normals of the imaged object. Good performance is reported for translating objects, however rotating objects lead to severe errors. Here we present a solution to this problem by modelling rotation and extending the affine flow to s- and t-derivatives of the flow-field. The model from [3,4] is valid for instantaneously moving cameras observing moving surfaces. This is unlike preceding work (e.g. [5,6,7,8]) where either an observed surface moves, or a camera, but not both. The patch-based affine model also differs from other frameworks for motion and stereo analysis, where point-based models are applied (e.g. [9,10,11,12]) or global minimization in 3D scene space is addressed (e.g. [13,14]).

1.1 Approach

The standard 2D affine optical flow model (cmp. e.g. [15])

$$\nabla I \left[\begin{pmatrix} u_x \\ u_y \end{pmatrix} + \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \right] + I_t = 0 \tag{1}$$

defines parameters in image coordinates, i.e. flow parameters. Here, the meaning of the flow parameters $u_\cdot$ and $a_{\cdot\cdot}$ will be explained in world coordinates and parameters of imaged moving surface patches.

Following [4] an image sequence may be interpreted as data in a 3D space where a brightness change constraint defines a linear model for intensity changes due to apparent local object motion. This motion is called optical flow. When the data is acquired by a single fixed camera, i.e. x-y-t-space, visible motion may be explained by object motion. When acquired by a moving camera looking at a fixed scene, i.e. x-y-s-space, displacements (then usually called disparities) are anti-proportional to local depth. This is known as structure from camera motion (e.g. [16]). Here, we interpret the camera position s as additional dimension of the data. Hence all image sequences acquired by a 1D camera grid can be combined to sample a 4D-volume in x-y-s-t-space. Brightness changes in this space are modelled as total differential of the intensity data.

The presented 4D fully affine optical flow model can be seen as an extended version of (1). But here, affine modelling not only covers linear changes in local pixel coordinates Δx, and Δy, but also in camera motion direction and time, i.e. additional Δs and Δt terms. In order to explain 3D structure and 3D motion in world coordinates by the estimated flow parameters, 3D dynamic surface patches are projected into the image by a pinhole camera (cmp. Sec. 2). A crucial point here is the correct handling of neighbor locations. We model it by back-projection of the pixel grid to the surface in the scene (see Secs. 2.4 and 2.5). A detailed derivation can be found in Section 2.

In order to evaluate the model we use a parameter estimation procedure as proposed in [4]. It is a total least squares (TLS) estimation scheme ideally suited for estimation when Gaussian noise is present. No robust statistics or regularization terms are applied. Such terms may conceal model errors and therefore are not suitable for model evaluation. Adaptations needed here are presented in Section 3.

Quantitative experiments (Section 4) use synthetic data with ground truth available. For systematic evaluation of accuracies we use pinhole-projected 32bit-float sinusoidal patterns suppressing otherwise unavoidable quantization noise. For more realistic scenes with ground truth available we use simple geometric structures moving in a known way rendered by POV-Ray [17] in 8bit-integer accuracy. We compare motion results to range flow [18,3] in order to give an intuition of the accuracies achievable using a simple TLS estimator. In contrast to our model, range flow needs depth information as input and estimates 3D translation only.

An experiment with real data showing a small tobacco leaf visually demonstrates the increased accuracy compared to other methods. Only the new method yields plausible results.

1.2 Contribution

The current paper is an extension to a series of papers [3,4]. Our contributions are the following: (1) Derivation of 4D affine flow parameters (Δs- and Δt-terms). (2) Back-projection of the pixel grid to an imaged surface respecting camera position and time. (3) Modelling rotational motion of surface patches. (4) A thorough model evaluation focusing on rotation effects.

Although our scheme can readily be used to estimate motion and shape deformations of plant leaves, we do not aim at presenting a final *method* yet. In our view a method not only needs accurate modelling but also needs well adapted discretization, estimation schemes, regularization etc. Here we focus on modelling, only.

2 Model Derivation

2.1 Surface Patch Model

Following [4] we model a surface patch at world coordinate position (X_0, Y_0, Z_0) as a function of time t by

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X_0 + U_X t + \Delta X \\ Y_0 + U_Y t + \Delta Y \\ Z_0 + U_Z t + Z_X \Delta X + Z_Y \Delta Y \end{pmatrix} \tag{2}$$

where Z_X and Z_Y are surface slopes in X- and Y-direction for time $t = 0$ and $\boldsymbol{U} = (U_X, U_Y, U_Z)$ is the velocity of the patch. The surface normal is then $\boldsymbol{n} = (Z_X, Z_Y, -1)$.

2.2 Rotation

We define rotation of a surface patch by angular velocity vector $\boldsymbol{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)^{\mathrm{T}}$ located at its central point $\boldsymbol{X}_0 = (X_0, Y_0, Z_0)^{\mathrm{T}}$. Velocity $\boldsymbol{U}$ of points on the surface patch is then determined by

$$\boldsymbol{U} = \boldsymbol{N} + \boldsymbol{\Omega} \times \Delta\boldsymbol{X} \tag{3}$$

with translational velocity $\boldsymbol{N} = (N_X, N_Y, N_Z)^T$, distance to the rotation center $\Delta\boldsymbol{X}$ and angular velocity $\boldsymbol{\Omega}$.

Equation (3) defines rotation around the surface patch center. For general rotational motion this is not sufficient, as the true center of rotation may not coincide with the patch center. This leads to accelerated motion of the patch center

$$\boldsymbol{U} = \boldsymbol{N} + \boldsymbol{A}t + \boldsymbol{\Omega} \times \Delta\boldsymbol{X}. \tag{4}$$

with acceleration $\boldsymbol{A}$. We address constant acceleration only, whereas acceleration coming from rotation is non-constant. This could be modelled by estimation of the true rotation center introducing 3 additional parameters analogue to (3).

2.3 Projective Camera Model

We use pinhole cameras at world coordinate positions $(s, 0, 0)$, looking into Z-direction

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X - s \\ Y \end{pmatrix} . \tag{5}$$

Sensor coordinates x, y are aligned with world coordinates X, Y. Camera position space is sampled equidistantly using a 1D camera grid. We combine data acquired by all cameras into one 4D data set equidistantly sampling the continuous intensity function $I(x, y, s, t)$.

2.4 Pixel-Centered View

Parameter estimation at a 4D pixel $\boldsymbol{x_0} = (x_0, y_0, s_0, t_0)$ is done using the acquired image data $I(\boldsymbol{x}) := I(x, y, s, t)$ in a local neighborhood Λ, with $\boldsymbol{x} := (x, y, s, t)^T$. Consequently we need to know surface position $\boldsymbol{X}$ for each 4D pixel, i.e. $\boldsymbol{X}(\boldsymbol{x})$. Using (5) we know

$$\begin{pmatrix} X(\boldsymbol{x}) \\ Y(\boldsymbol{x}) \end{pmatrix} = \frac{Z(\boldsymbol{x})}{f} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} s \\ 0 \end{pmatrix} . \tag{6}$$

In order to derive an expression for $Z(\boldsymbol{x})$ we fit a tangent plane with surface normal $\boldsymbol{n} = (Z_X, Z_Y, -1)^T$ to the point $\boldsymbol{X}(\boldsymbol{x})$. The intersection of this tangent plane with the Z-axis is then $Z(0,0,0,t)$, and $Z(0,0,0,t) = Z(\mathbf{0}) + Z_t t$ for a constantly translating plane, where $\mathbf{0} := (0,0,0,0)^T$. Consequently $Z(\boldsymbol{x})$ can be expressed as

$$Z(\boldsymbol{x}) = Z(\mathbf{0}) + Z_X X(\boldsymbol{x}) + Z_Y Y(\boldsymbol{x}) + Z_t t \Leftrightarrow Z(\boldsymbol{x}) = \frac{Z(\mathbf{0}) + Z_X s + Z_t t}{1 - Z_X \frac{x}{f} - Z_Y \frac{y}{f}} \quad (7)$$

where we used (6) to substitute $X(\boldsymbol{x})$ and $Y(\boldsymbol{x})$. Combining (6) and (7) yields

$$\boldsymbol{X}(\boldsymbol{x}) = \frac{Z(\mathbf{0}) + Z_X s + Z_t t}{f - Z_X x - Z_Y y} \begin{pmatrix} x \\ y \\ f \end{pmatrix} + \begin{pmatrix} s \\ 0 \\ 0 \end{pmatrix}. \quad (8)$$

Equation (8) extends the formulation in [4], where $\boldsymbol{X}$ only depends on local image coordinates x and y.

2.5 Projecting the Pixel Grid to the Surface

A pixel $\boldsymbol{x}$ in the local neighborhood Λ used for parameter estimation is given by $\boldsymbol{x} = \boldsymbol{x_0} + \Delta\boldsymbol{x} = (x_0 + \Delta x, y_0 + \Delta y, s_0 + \Delta s, t_0 + \Delta t)^T$. The surface patch center $\boldsymbol{X_0}$ by definition is the projection of the neighborhood center point $\boldsymbol{x_0}$ to the surface. Thus neighbor points of $\boldsymbol{X_0}$ on the surface given by $\Delta\boldsymbol{X} = (\Delta X, \Delta Y, \Delta Z)$ are projections of the cameras pixel grids to the surface. We need to derive $\Delta\boldsymbol{X}(\Delta\boldsymbol{x})$. To stay on the surface, we model $\Delta Z = Z_X \Delta X + Z_Y \Delta Y$, cmp. (2). From (2) we know

$$\Delta\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{X}_0 - \boldsymbol{U}t \quad (9)$$

where $\boldsymbol{X}$ is a point on the surface at a given point in time t, $\boldsymbol{X}_0$ is the surface patch center at time $t_0 = 0$, and $\Delta\boldsymbol{X}$ is the distance between $\boldsymbol{X}$ and the point $\boldsymbol{X}_0 + \boldsymbol{U}t$ where the patch center moved to. The distance between $\boldsymbol{X}$ and $\boldsymbol{X_0}$ can be expressed by the linearisation

$$\boldsymbol{X} - \boldsymbol{X}_0 = \frac{\partial \boldsymbol{X}}{\partial x}\Delta x + \frac{\partial \boldsymbol{X}}{\partial y}\Delta y + \frac{\partial \boldsymbol{X}}{\partial s}\Delta s + \frac{\partial \boldsymbol{X}}{\partial t}\Delta t\,. \quad (10)$$

Partial derivatives of $\boldsymbol{X}(\boldsymbol{x})$ can be derived from (8).

2.6 Brightness Change Model

Point correspondences in our multi-dimensional data set are derived via an estimation analogue to common structure-from-camera-motion or optical-flow methods. Thus we employ a differential brightness constraint modelling intensity changes $\mathrm{d}I$ of a surface element for the 4D data set $I(x, y, s, t)$. $\mathrm{d}I$ equals

$$I_x \mathrm{d}x + I_y \mathrm{d}y + I_s \mathrm{d}s + I_t \mathrm{d}t = I(g_1 + g_{1,x}\Delta x + g_{1,y}\Delta y + g_2 t)\mathrm{d}t\,. \quad (11)$$

We denote $I_* = \frac{\partial I}{\partial *}$ for derivatives of the image intensities I. In the following we use notation $g = (g_1 + g_{1,x}\Delta x + g_{1,y}\Delta y + g_2 t)$. The left hand side of (11) is derived from dI by chain rule. The right hand side of (11) models spatially varying brightness changes. It boils down to a local spatio-temporal series expansion of varying illumination times bidirectional reflectance distribution function (BRDF). We refer to [3] for a detailed derivation.

2.7 A 4D-Affine Model

We combine the above equations in order to derive the new 4D optical flow model and a geometric interpretation of its parameters. Again following [4] we project the moving surface patch model to the sensor plane by substituting (2) in (5) and calculate the differentials $\mathrm{d}x$ and $\mathrm{d}y$ for a *given surface location* (i.e. for constant ΔX and ΔY)

$$\begin{pmatrix} \mathrm{d}x \\ \mathrm{d}y \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} (U_X - U_Z \frac{x}{f})\mathrm{d}t - \mathrm{d}s \\ (U_Y - U_Z \frac{y}{f})\mathrm{d}t \end{pmatrix} . \tag{12}$$

From (2) we know that Z depends on the unknown ΔX and ΔY: $Z = Z_0 + U_Z \Delta t + Z_X \Delta X + Z_Y \Delta Y$. We therefore rephrase f/Z using (9) and (10)

$$f/Z = -\nu - b_1 \Delta x - b_2 \Delta y - b_3 \Delta s - b_4 \Delta t \tag{13}$$

$$\begin{aligned} &\text{with} \quad \nu = -\frac{f}{Z_0}, \quad b_1 = \frac{Z_X}{Z_0 c}, \quad b_2 = \frac{Z_Y}{Z_0 c}, \quad b_3 = \frac{f}{Z_0}\frac{Z_X}{Zc}, \quad b_4 = \frac{f}{Z_0}\frac{Z_t}{Zc} \\ &\text{and} \quad c = 1 - Z_X \frac{x}{f} - Z_Y \frac{y}{f} . \end{aligned} \tag{14}$$

The remaining substitution steps are then as follows: first in (4) substitute $\Delta \boldsymbol{X}$ by (9)–(10), then in (12) substitute $\boldsymbol{U}$ by (4) and $\frac{f}{Z}$ by (13). Finally substitute in the brightness change model (11) $\mathrm{d}x$ and $\mathrm{d}y$ by (12). Ignoring higher order terms yields representations of the elements of the 4 dimensional optical flow model

$$\begin{aligned} &\nabla I \left[\begin{pmatrix} u_x \mathrm{d}t + \nu \mathrm{d}s \\ u_y \mathrm{d}t \end{pmatrix} + \begin{pmatrix} a_{11}\mathrm{d}t + b_1 \mathrm{d}s & a_{12}\mathrm{d}t + b_2 \mathrm{d}s & a_{13}\mathrm{d}t + b_3 \mathrm{d}s & a_{14}\mathrm{d}t + b_4 \mathrm{d}s \\ a_{21}\mathrm{d}t & a_{22}\mathrm{d}t & a_{23}\mathrm{d}t & a_{24}\mathrm{d}t \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \\ \Delta t \end{pmatrix} \right] \\ &+ I_s \mathrm{d}s + I_t \mathrm{d}t = I g \mathrm{d}t \end{aligned} \tag{15}$$

with parameters

$$\begin{aligned}
u_x &= -\nu \left(N_X - \tfrac{x_0}{f} N_Z \right) \\
u_y &= -\nu \left(N_Y - \tfrac{y_0}{f} N_Z \right) \\
a_{11} &= -\nu \left[\Omega_Y \tfrac{\partial Z}{\partial x} - \Omega_Z \tfrac{\partial Y}{\partial x} - \tfrac{x_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial x} - \Omega_Y \tfrac{\partial X}{\partial x} \right) - \tfrac{N_Z}{Z_0} \right] - b_1 \left(N_X - \tfrac{x_0}{f} N_Z \right) \\
a_{12} &= -\nu \left[\Omega_Y \tfrac{\partial Z}{\partial y} - \Omega_Z \tfrac{\partial Y}{\partial y} - \tfrac{x_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial y} - \Omega_Y \tfrac{\partial X}{\partial y} \right) \right] - b_2 \left(N_X - \tfrac{x_0}{f} N_Z \right) \\
a_{13} &= -\nu \left[\Omega_Y \tfrac{\partial Z}{\partial s} - \Omega_Z \tfrac{\partial Y}{\partial s} - \tfrac{x_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial s} - \Omega_Y \tfrac{\partial X}{\partial s} \right) \right] - b_3 \left(N_X - \tfrac{x_0}{f} N_Z \right) \\
a_{14} &= -\nu \left[\Omega_Y \tfrac{\partial Z}{\partial t} - \Omega_Z \tfrac{\partial Y}{\partial t} - \tfrac{x_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial t} - \Omega_Y \tfrac{\partial X}{\partial t} \right) + \left(A_X - \tfrac{x_0}{f} A_Z \right) \right] - b_4 \left(N_X - \tfrac{x_0}{f} N_Z \right) \\
a_{21} &= -\nu \left[\Omega_Z \tfrac{\partial X}{\partial x} - \Omega_X \tfrac{\partial Z}{\partial x} - \tfrac{y_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial x} - \Omega_Y \tfrac{\partial X}{\partial x} \right) \right] - b_1 \left(N_Y - \tfrac{y_0}{f} N_Z \right) \\
a_{22} &= -\nu \left[\Omega_Z \tfrac{\partial X}{\partial y} - \Omega_X \tfrac{\partial Z}{\partial y} - \tfrac{y_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial y} - \Omega_Y \tfrac{\partial X}{\partial y} \right) - \tfrac{N_Z}{Z_0} \right] - b_2 \left(N_Y - \tfrac{y_0}{f} N_Z \right) \\
a_{23} &= -\nu \left[\Omega_Z \tfrac{\partial X}{\partial s} - \Omega_X \tfrac{\partial Z}{\partial s} - \tfrac{y_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial s} - \Omega_Y \tfrac{\partial X}{\partial s} \right) \right] - b_3 \left(N_Y - \tfrac{y_0}{f} N_Z \right) \\
a_{24} &= -\nu \left[\Omega_Z \tfrac{\partial X}{\partial t} - \Omega_X \tfrac{\partial Z}{\partial t} - \tfrac{y_0}{f} \left(\Omega_X \tfrac{\partial Y}{\partial t} - \Omega_Y \tfrac{\partial X}{\partial t} \right) + \left(A_Y - \tfrac{y_0}{f} A_Z \right) \right] - b_4 \left(N_Y - \tfrac{y_0}{f} N_Z \right)
\end{aligned} \tag{16}$$

The partial derivatives of world coordinates in (16) can be derived from (8), $b_\cdot$ and ν are given in (14).

2.8 The Range Constraint, Z_t, b_4, and Why (8) Still Holds under Rotation

Flow parameter b_4 (see (14)) depends on Z_t, the partial t-derivative of Z. We are not explicitly interested in Z_t, thus we want to express it using parameters we are interested in. We know that $U_Z := dZ/dt$ and thus that the time derivative of the first equation in (7) yields the *range constraint* known from [18]

$$Z_t = U_Z - Z_X U_X - Z_Y U_Y \tag{17}$$

valid for *translating* planes, i.e. for constant surface slopes Z_X and Z_Y. Obviously surface slopes change when a surface rotates and the range constraint becomes

$$Z_t = U_Z - Z_X U_X - Z_Y U_Y - X Z_{X,t} - Y Z_{Y,t} \tag{18}$$

where $Z_{X,t}$ and $Z_{Y,t}$ are t-derivatives of Z_X and Z_Y.

Equation (7) was derived for constant Z_X and Z_Y. For rotating surfaces we approximate them via first order Taylor expansions $Z_X(t) = Z_X(0) + Z_{X,t}t$ and $Z_Y(t) = Z_Y(0) + Z_{Y,t}t$ and derive for (7)

$$\begin{aligned} & Z(\boldsymbol{x}) = Z(\mathbf{0}) + Z_X(t)X(\boldsymbol{x}) + Z_Y(t)Y(\boldsymbol{x}) + Z_t t \\ \Leftrightarrow\ & Z(\boldsymbol{x}) = \tfrac{1}{c}\left(Z(\mathbf{0}) + Z_X(\mathbf{0})s + (Z_t + Z_{X,t}X(\boldsymbol{x}) + Z_{Y,t}Y(\boldsymbol{x}))t\right) \end{aligned} \tag{19}$$

Substituting Z_t using (18) yields

$$Z(\boldsymbol{x}) = \frac{1}{c}\left(Z(\mathbf{0}) + Z_X s + (U_Z - Z_X U_X - Z_Y U_Y)t\right) = \frac{Z(\mathbf{0}) + Z_X s + \tilde{Z}_t t}{c} \tag{20}$$

where now $\tilde{Z}_t$ is *defined* by the standard range constraint (17). We conclude that (8) still holds for a first order model of rotational motion, if $\tilde{Z}_t$ ignores surface slope changes due to rotation. Consequently Z_t in (14) also becomes $\tilde{Z}_t$.

3 Parameter Estimation

We calculate image derivatives by optimized 5-tab derivative filter sets presented in [19] and then estimate parameters in three steps. **1.** We solve for 4D affine optical flow parameters ν, $b_1, \ldots, b_4$, u_x, u_y, $a_{11}, \ldots, a_{24}$ and brightness change parameters $g_1, g_{1,x}, g_{1,y}, g_2$ using a usual local total least squares estimator (see [4] for details). **2.** We solve for depth Z_0, and surface normals Z_X and Z_Y, and Z_t using (14), where focal length f has to be known e.g. from a calibration step. This allows to calculate c from (14) and partial derivatives of world coordinates from (8). **3.** We solve for translation $\boldsymbol{N}$, and rotation $\boldsymbol{\Omega}$ if desired, using the equations in (16) or – as reference methods – using the *Range Flow* method from [21]. Acceleration $\boldsymbol{A}$ can only be estimated up to 1 degree

of freedom as we have only 2 equations (the ones for a_{14} and a_{24}) for 3 parameters A_X, A_Y, A_Z. (16) and (14) being an overdetermined system of equations, there are several ways to solve for $\boldsymbol{N}$ and $\boldsymbol{\Omega}$ using a standard least squares estimation scheme. For our experiments we select the following submodels by selecting some or all equations from (16) and (14), or by removing terms when parameters like rotation or acceleration are not estimated. This is equivalent to not modelling these parameters or setting them to zero. We use the following submodels

2D OF trans. estimates $\boldsymbol{N}$ only, using equations for $u_x, u_y, a_{11}, a_{12}, a_{21}, a_{22}$ (i.e. the method from [3]).
2D OF rot. estimates $\boldsymbol{N}$ and $\boldsymbol{\Omega}$ using equations for $u_x, u_y, a_{11}, a_{12}, a_{21}, a_{22}$.
2D OF trans. and ... and *2D OF rot. and ...* using additional equations indicated by ...
4D OF trans. estimates $\boldsymbol{N}$ only, using all 11 equations containing motion information, 10 from (16) and 1 from (14), i.e. the one for b_4.
4D OF rot. estimates $\boldsymbol{N}$ and $\boldsymbol{\Omega}$ only, using the 11 equations.
4D OF estimates $\boldsymbol{N}$, $\boldsymbol{\Omega}$, and $\boldsymbol{A}$ using the 11 equations.

Parameters that are not solved for are set to zero. *4D OF* uses two equations more than *2D OF rot. and* a_{13}, a_{23}, b_4 but estimates $\boldsymbol{A}$ in addition. We therefore get identical results for $\boldsymbol{N}$ and $\boldsymbol{\Omega}$ using the two models. Thus we do not show results for *4D OF* in the experiments below.

4 Experiments

In a first experiment we use synthetic sinusoidal sequences for systematic error analysis. Then the different models are compared on more realistic data with ground truth, i.e., a moving cube rendered with POV-Ray [17]. Finally we show results for a rotating plant leaf.

4.1 Sinusoidal Pattern

For systematic error analysis we render a surface patch with sinusoidal pattern, where geometry and intensities mimic typical settings used in our actual lab experiments with plants. The 32-bit float intensity values are in the range $[50; 150]$. Input sequences are generated with surface patch parameters $Z_0 = 100$ mm, $Z_X = 0.6$, and $Z_Y = -0.5$, and motion parameters $\boldsymbol{N} \approx (0.0073, -0.0037, -0.3)^{\mathrm{T}}$ mm/frame and $\omega = 0.003$ degree/frame around rotational axis $\boldsymbol{v} = (2, 3, 2)^{\mathrm{T}}$, i.e., $\boldsymbol{\Omega} = \omega \boldsymbol{v}$. In each experiment we vary only one of these parameters. The synthetic sensor contains 501×501 pixels with width and height 0.0044 mm. The focal length of the projective camera is set to $f = 12$ mm. We generate data for 9 cameras, positioned horizontal as a 1D, equidistantly spaced camera grid with displacement of 0.5 mm. In order to keep optical flow in camera displacement direction below 1 pixel/displacement, we $preshift$ the data by 13 pixel/displacement. The effective image size shrinks to

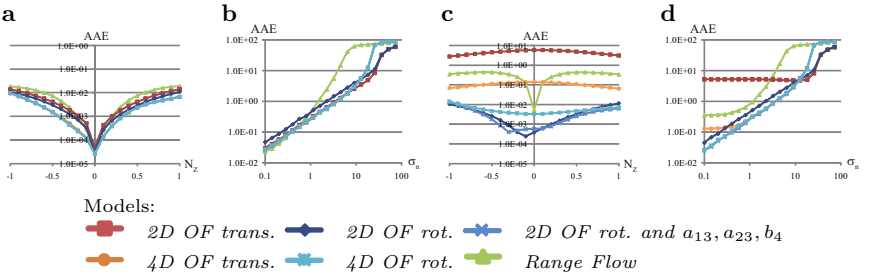


Fig. 1. Average angular error (AAE) versus increasing N_Z and σ_n. **a**, **b**: data without rotation, **c**, **d**: with rotation.

301×301 pixel due to border effects. Neighbourhood Λ is implemented by a Gaussian filter with size $65 \times 65 \times 5 \times 5$ and standard deviations $19 \times 19 \times 1 \times 1$ in x, y, s, t-directions. In order to compare performance of models, we use the average angular error [20]

$$AAE = \frac{1}{N} \sum_{i=1}^{N} \arccos\left(\boldsymbol{r}_t(i)^{\mathrm{T}} \boldsymbol{r}_e(i)\right) \tag{21}$$

for N pixel with a minimum border distance of 60 pixel, true motion $\boldsymbol{r}_t$ and estimated motion $\boldsymbol{r}_e$, with $\boldsymbol{r} = (\boldsymbol{N}^{\mathrm{T}}, 1)^{\mathrm{T}}$ for translation and $\boldsymbol{r} = (\boldsymbol{\Omega}^{\mathrm{T}}, 1)^{\mathrm{T}}$ for rotation. Figure 1 shows average angular errors of translational motion estimates for sequences without (**a**,**b**) and with (**c**,**d**) rotation. We show errors for increasing translational motion N_Z in Figs. 1**a** and **c** and for increasing standard deviation of noise σ_n in Figs. 1**b** and **d**. Figures 1**a** and **b** demonstrate that all models perform almost equally well for sequences without rotation. Models using more affine parameters (*4D OF trans./rot.*, and *2D OF rot. and* a_{13}, a_{23}, b_4) perform best. *Range Flow* performs only slightly better for low noise sequences. In case of rotation (Figs. 1**c** and **d**) *Range Flow* and the translational models yield high errors compared to rotational models. However, comparing rotational models, *4D OF rot.* performs worst. This indicates that modelling $\boldsymbol{A}$ in the equations for a_{14} and a_{24} ((16)) or not using a_{14} and a_{24} (i.e. *2D OF rot. and* a_{13}, a_{23}, b_4) is beneficial. In case of noise rotational models using 4D affine terms (*2D OF rot. and* a_{13}, a_{23}, b_4 and *4D OF rot.*) show best performance up to $\sigma_n = 10$. *4D OF trans.* shows considerable better performance than *Range Flow*, despite for $N_Z = 0$ and no noise, and performs as good as the best rotational models for $1 < \sigma_n < 10$. In Fig. 2 we compare average angular errors of translational (Fig. 2**a**) and rotational (Fig. 2**b**–**d**) motion parameters for different rotation models. Translational models and *Range Flow* are shown for reference in Fig. 2**a**. Figures 2**c** and **d** show angular errors of rotational parameters for a sequence with rotation and increasing N_Z and σ_n, respectively, i.e. , the $\boldsymbol{\Omega}$ counterparts of Figs. 1**c** and **d**.

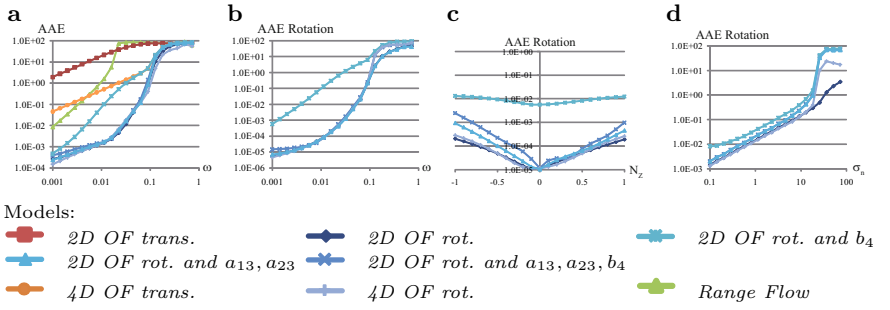


Fig. 2. **a**: AAE of $\boldsymbol{N}$ versus ω and **b**–**d**: AAE of $\boldsymbol{\Omega}$ versus **b**: ω, **c**: N_Z and **d**: σ_n.

The figures demonstrate that incorporating the affine parameters a_{14} and a_{24} in *4D OF rot.* without modelling of acceleration significantly increases errors. Figures 2**a** and **b** show average angular errors of translational and rotational parameters for sequences with increasing ω. Rotational models without a_{14} and a_{24} perform similar and up to three orders of magnitude better than *Range Flow* and the translational models.

We conclude that modelling rotation yields almost always significantly lower or at least similar errors as the translational models and *Range Flow*. Using a_{14} and a_{24} without modelling acceleration $\boldsymbol{A}$ should be avoided.

4.2 Synthetic Cube

The synthetic cube sequence allows us to compare models on more realistic data with ground truth available. The cube centre is at $Z = 600$mm, moves with $\boldsymbol{N} = (-0.2, 0, -1)^T$ mm/frame, and rotates around its Y-axis with $\omega = 0.4$ degrees/frame. It is covered with a noise pattern in order to make local estimation reliable. Neighbourhood Λ is the same as for the sinusoidal sequences. The 1D camera grid contains 9 cameras with a displacement of 5 mm. Figures 3**a**–**d** show first and last frame of the central camera, two regions where errors are evaluated, and ground truth motion, respectively.

Structure estimation accuracies for different choices of Λ and optical flow types are given in Tab.1. We see that using surface normals, i.e. affine terms $b_.$, and data from more than one point in time in the estimation improves accuracy by more than 1 order of magnitude. Accuracy is then comparable to typical laser scanning range sensors, e.g. Sick IVP Ruler E600 with 0.2mm resolution.

Table 1. Average surface distances in mm (mean $\pm$ std. deviation). 'Left' and 'right' refer to the areas of interest depicted in Fig.3**c**.

Neighbourhood Λ	flow type	Error 'left'	Error 'right'
$65 \times 65 \times 1 \times 1$	not affine	2.77 ± 0.91	3.10 ± 1.34
$65 \times 65 \times 1 \times 1$	affine	0.15 ± 0.07	0.29 ± 0.16
$65 \times 65 \times 5 \times 5$	affine	0.08 ± 0.03	0.21 ± 0.11

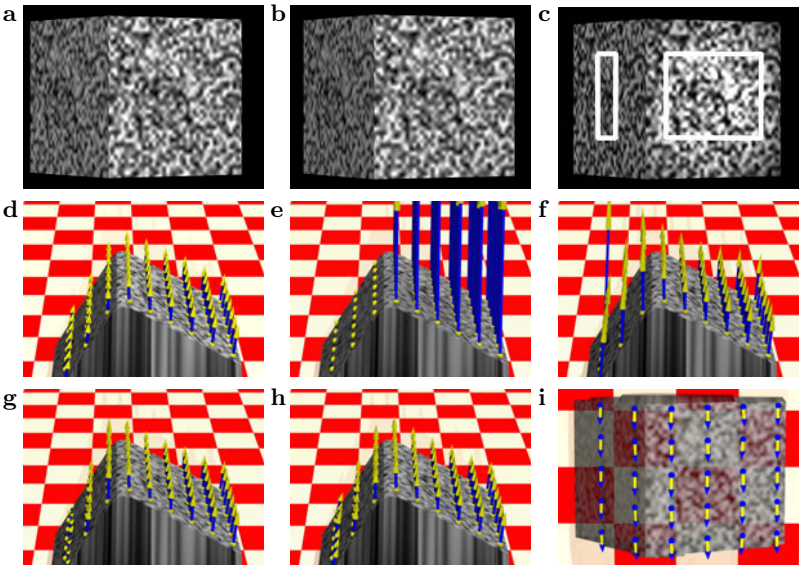


Fig. 3. Cube moving towards camera with rotation. Top row: First (**a**) and last (**b**) input frame, and central frame with evaluation areas (**c**). **d**: ground truth motion. Motion estimates $\boldsymbol{U}$ with amplified errors. **e**: *2D OF trans.*, **f**: *Range Flow*, **g**: *4D OF trans.*, and **h**: *2D OF rot. and* a_{13}, a_{23}, b_4. **i**: Rotational motion $\boldsymbol{\Omega}$ estimated via *2D OF rot. and* a_{13}, a_{23}, b_4.

Motion estimates of two translational models, one rotational model and *Range Flow* are shown in Figs. 3e–**h**. The errors are amplified by a factor of 5 for better comparison of the models. The estimates of *2D OF trans.* clearly show large errors, where estimates on the right side of the cube point in Z-direction, estimates on the left side of the cube are not visible because they point inwards the cube. Estimates of *Range Flow* are more accurate, but distorted near borders of the cube. Models *4D OF trans.* and *2D OF rot. and* a_{13}, a_{23}, b_4 yield more accurate results. Estimates of the translational model are still distorted, mainly on the left side of the cube. Estimation results of the rotational model best match the ground truth. Fig. 3**i** shows a rendered top view of the cube with estimation results of rotational motion using the model *2D OF rot. and* a_{13}, a_{23}, b_4. The estimates clearly recover the true motion.

Table 2 shows angular errors for the regions depicted in Fig. 3c which quantitatively confirm the visual impression of the rendered results. *2D OF trans.* performs better when b_4, a_{13} and a_{23}, or all three terms are additionally used for estimation. Otherwise estimates are heavily distorted. The same is true for translation estimates with models also estimating rotation. Rotation estimates are equally well for all rotation models. Errors of *Range Flow* are lower than for *2D OF trans.*, but significantly higher than for models incorporating more affine terms.

Table 2. Average angular error (AAE) and standard deviations in degrees of translational and rotational motion parameters of regions on left and right side of the cube (see Fig. 3**c**). Errors or standard deviations above 1° (AAE) are indicated in red, below 0.1° (AAE) in green.

motion model	affine parameters	AAE left region		AAE right region	
		translation	rotation	translation	rotation
Translation	2D OF	112 ± 1.01	n/a	19.8 ± 0.67	n/a
	2D OF + b_4	1.22 ± 0.43	n/a	0.32 ± 0.20	n/a
	2D OF + a_{13},a_{23}	1.72 ± 1.28	n/a	1.06 ± 0.57	n/a
	2D OF + a_{13},a_{23},b_4	0.91 ± 0.64	n/a	0.58 ± 0.33	n/a
	4D OF	0.91 ± 0.64	n/a	0.58 ± 0.33	n/a
Translation and rotation	2D OF	6.85 ± 6.24	0.018 ± 0.009	1.73 ± 1.37	0.004 ± 0.011
	2D OF + b_4	0.52 ± 0.19	0.017 ± 0.009	0.27 ± 0.17	0.004 ± 0.011
	2D OF + a_{13},a_{23}	0.93 ± 0.37	0.017 ± 0.009	0.19 ± 0.11	0.004 ± 0.011
	2D OF + a_{13},a_{23},b_4	0.67 ± 0.29	0.017 ± 0.009	0.22 ± 0.13	0.005 ± 0.011
Range Flow		8.86 ± 0.69	n/a	1.88 ± 0.31	n/a

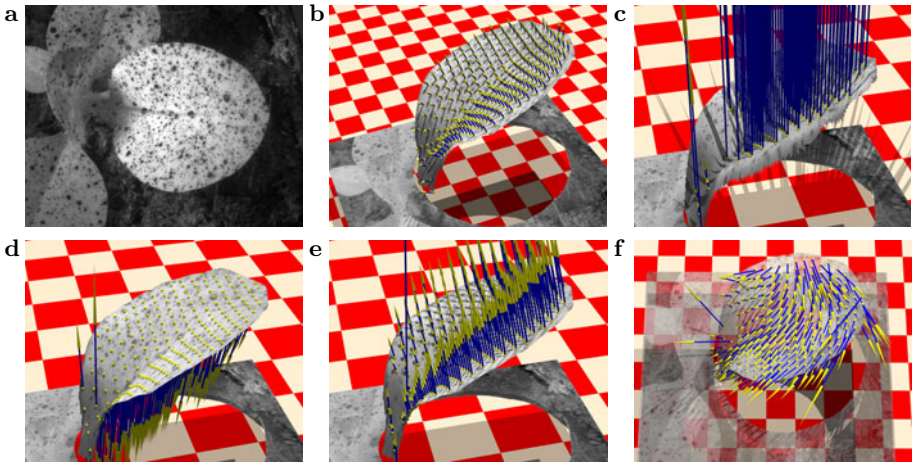


Fig. 4. Plant Leaf Sequence. **a**: Central frame of central camera. **b**: Estimated structure and surface normals. Motion estimates for **c**: *Range Flow*, **d**: *2D OF trans.* and **e**: proposed new model *2D OF rot and* a_{13},a_{23} *and* b_4. **f**: estimated rotational velocity.

4.3 Plant Leaf

Figure 4**a** shows one frame of a tobacco plant leaf input sequence. The leaf is textured with watercolour to reduce errors coming from the aperture problem (cmp. [1]). The scene is illuminated by directed infrared light emitting diodes from top causing illumination changes on the leaves. The maximal width of the leaf is approximately 20 mm. Images are taken by a movingstage-based 1D camera grid with 9 positions at 1 mm distance (see Sec. 1). Sampling rate of the camera per position is one image every 2 minutes. Sensor size is 1600×1200 pixel. Neighbourhood Λ is implemented using a Gaussian filter with size 121×121×5×5 and standard deviation 41 × 41 × 1 × 1 in x, y, s, t-direction.

The big leaf on the right rotates upward around its node where it is attached to the stem (i.e. approx. around the Y-axis). This results in a visible motion towards the camera and to the left. Moreover the leaf unrolls along its midvein and folds its sides up. Figure 4**b** shows estimated structure and surface normals. Visibly the true structure is well recovered. Translation estimates for the presented models are shown in Fig. 4**c**–**e**. *Range Flow* [21] significantly overestimates the motion (Fig. 4**c**). With the purely translational motion model *2D OF trans.* [3] estimation results are heavily corrupted (Fig. 4**d**). This model apparently interprets shrinkage of the projected leave length in x-direction due to rotation as being caused by motion away from the camera. The rotational model *2D OF rot and* a_{13},a_{23} *and* b_4 yields a severely improved motion vector field, even though motion still seems to be overestimated. Figure 4**f** shows estimated rotational motion vectors. Rotation around the node is well visible. Unrolling and folding of the leaf can be recovered by analysing changes in the rotation vector field. Making this possible was the main goal of the presented work (cmp. Sec. 1).

5 Summary and Conclusions

In this paper we presented a 4D affine optical flow model and how the parameters of this model can be explained by real world parameters. Based on a rigid surface patch we modelled translation, acceleration and rotation. The rotational model improves estimation results in almost all cases and additionally allows to estimate rotational parameters which is of high interest for understanding plant physiology. Synthetic experiments showed that modelling acceleration is not sufficient to estimate rotation reliably and should therefore not be used if rotation occurs in the sequence. The 4D affine model and its explanation of real world parameters improved accuracy of motion estimates on synthetic and real data compared to *Range Flow* and previous *2D OF* affine models. In order to increase accuracy further and cope with different scenarios than plant leaf estimation, the main focus in future research will be on developing a more sophisticated estimator. Furthermore the estimator should be able to handle large motions coming from camera displacement. This is a prerequisite for using full camera arrays, like e.g. [22], instead of moving stages, and adapting the affine optical flow model to other application areas.

References

1. Biskup, B., Küsters, R., Scharr, H., Walter, A., Rascher, U.: Quantification of plant surface structures from small baseline stereo images to measure the three-dimensional surface from the leaf to the canopy scale. In: Nova Acta Leopoldina, vol. 357(96), pp. 31–47 (2009)
2. Walter, A., Christ, M.M., Barron-Gafford, G.A., Grieve, K.A., Paige, T., Murthy, R., Rascher, U.: The effect of elevated co2 on diel leaf growth cycle, leaf carbohydrate content and canopy growth performance of populus deltoides. Global Change Biology 8, 1207–1219 (2005)

3. Schuchert, T., Scharr, H.: Simultaneous estimation of surface motion, depth and slopes under changing illumination. In: Hamprecht, F.A., Schnörr, C., Jähne, B. (eds.) DAGM 2007. LNCS, vol. 4713, pp. 184–193. Springer, Heidelberg (2007)
4. Schuchert, T., Scharr, H.: An affine optical flow model for dynamic surface reconstruction. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) Dagstuhl Seminar. LNCS, vol. 5604, pp. 70–90. Springer, Heidelberg (2009)
5. Longuet-Higgins, H., Prazdny, K.: The interpretation of a moving retinal image. Proceedings of The Royal Society of London B 208, 385–397 (1980)
6. Kanatani, K.: Structure from motion without correspondence: general principle. In: Proc. Image Understanding Workshop, pp. 10711–10716 (1985)
7. Adiv, G.: Determining 3-d motion and structure from optical flow generated by several moving objects. PAMI 7, 384–401 (1985)
8. Subbarao, M., Waxman, A.: Closed form solutions to image flow equations for planar surfaces in motion. CVGIP: Graphical Models and Image Processing 36, 208–228 (1986)
9. Szeliski, R.: A multi-view approach to motion and stereo. In: CVPR (1999)
10. Vedula, S., Baker, S., Rander, P., Collins, R., Kanade, T.: Three-dimensional scene flow. PAMI 27, 475–480 (2005)
11. Carceroni, R., Kutulakos, K.: Multi-view 3d shape and motion recovery on the spatio-temporal curve manifold. In: ICCV, vol. (1), pp. 520–527 (1999)
12. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, D.: Efficient dense scene flow from sparse or dense stereo data. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 739–751. Springer, Heidelberg (2008)
13. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. IJCV 72(2), 179–193 (2007)
14. Kolev, K., Klodt, M., Brox, T., Cremers, D.: Continuous global optimization in multiview 3d reconstruction. IJCV 84, 80–96 (2009)
15. Fleet, D., Weiss, Y.: Optical flow estimation. In: Mathematical models for Computer Vision: The Handbook. Springer, Heidelberg (2005)
16. Matthies, L.H., Szeliski, R., Kanade, T.: Kalman filter-based algorithms for estimating depth from image sequences. IJCV 3, 209–236 (1989)
17. Cason, C.: Persistence of vision ray tracer (POV-Ray), version 3.6, Windows (2005)
18. Spies, H., Jähne, B., Barron, J.: Range flow estimation. CVIU 85, 209–231 (2002)
19. Scharr, H.: Optimal filters for extended optical flow. In: Jähne, B., Mester, R., Barth, E., Scharr, H. (eds.) IWCM 2004. LNCS, vol. 3417, pp. 14–29. Springer, Heidelberg (2007)
20. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. IJCV 12(1), 43–77 (1994)
21. Schuchert, T., Aach, T., Scharr, H.: Range flow for varying illumination. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 509–522. Springer, Heidelberg (2008)
22. ViewPLUS: ProFUSION 25 (2008), http://www.viewplus.co.jp/products/profusion25/ProFUSION25-e.pdf

Efficiently Scaling Up Video Annotation with Crowdsourced Marketplaces

Carl Vondrick, Deva Ramanan, and Donald Patterson

Department of Computer Science
University of California, Irvine, USA
{cvondric,dramanan,djp3}@ics.uci.edu

Abstract. Accurately annotating entities in video is labor intensive and expensive. As the quantity of online video grows, traditional solutions to this task are unable to scale to meet the needs of researchers with limited budgets. Current practice provides a temporary solution by paying dedicated workers to label a fraction of the total frames and otherwise settling for linear interpolation. As budgets and scale require sparser key frames, the assumption of linearity fails and labels become inaccurate. To address this problem we have created a public framework for dividing the work of labeling video data into micro-tasks that can be completed by huge labor pools available through crowdsourced marketplaces. By extracting pixel-based features from manually labeled entities, we are able to leverage more sophisticated interpolation between key frames to maximize performance given a budget. Finally, by validating the power of our framework on difficult, real-world data sets we demonstrate an inherent trade-off between the mix of human and cloud computing used vs. the accuracy and cost of the labeling.

1 Introduction

Sorokin and Forsyth [1] made the influential observation that *image* labeling can be crowdsourced at low costs through platforms such as Amazon's Mechanical Turk (MTurk). This approach has revolutionized *static* data annotation in vision, and enabled almost all large-scale image data sets collected since then to be labeled [2,3,4]. Contemporary computer vision research has subsequently demonstrated the value of massive data sets of labeled images such as the results from ImageNet [2], LabelMe [3], and TinyImages [5].

The same does not hold true for video despite a corresponding abundance of data, such as that from webcams and public-domain archival footage [6]. We believe that this is due to the *dynamic* nature of video data which makes frame by frame labeling necessary but inefficient for manual labor. Inspired by popular successes such as [7,8,9], we focus on cost effective video annotation with MTurk. We show the results of a year's worth of experiments on learning how to use MTurk to effectively label video. This has resulted in our release of *vatic* (Video Annotation Tool from Irvine, California), the first open platform for monetized,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 610–623, 2010.

Fig. 1. An example of the difficult problem that our interactive system addresses. The red boxed player becomes totally occluded while many players quickly change pose from standing to a prone position. The referees commonly enter and leave the scene. The ball exists in the pile of people, but even a state-of-the-art vision algorithm is unable to determine its position.

crowdsource video labeling, and a set of "best practices" for creating video-labeling tasks on a crowdsourced marketplace. Our hope is that our findings will spur innovation in the creation of affordable, massive data sets of labeled video.

The contributions made in this paper are motivated by our desire to uncover best-practices for monetized crowdsourced video labeling. In section 2, we present insights into the design of a user-interface in which workers track a single object through a continuous video shot (to solve the problem in Fig.1). In section 3, we analyze trade-offs particular to balancing computer and human effort in video annotation by extending work that minimized labeling cost only along the dimension of human effort [8,10]. Although the "Turk philosophy" is to completely replace difficult computer tasks (such as video labeling) with human effort, this is clearly *not* efficient given the redundancy of video. In contrast to VideoLabelMe [7], we show that one can interpolate *nonlinear* least-cost paths with efficient dynamic programming algorithms based on image data and user annotated endpoints. In section 4, we analyze the total cost of labeling for various combinations of human workers and cloud-computing CPU cycles. Our final contribution is the release of a simple, reusable, and open-source platform for research video labeling.

2 Mechanical Turk

Amazon's Mechanical Turk is an online labor marketplace that connects employers who have small tasks that are difficult for computers but trivial for humans and workers. Employers create *Human Intelligence Tasks* (HITs) and set their

prices (typically very low) before posting them to the Mechanical Turk servers. Workers browse offered jobs and accept those that interest them. Task completion is not guaranteed and is the result of typical market dynamics [11]. Upon employer validation of completed work, Amazon releases escrowed payments to the workers. Our system uses MTurk to locate workers who can annotate video. We create HITs by first dividing a large video into smaller sequences shot by a single camera. This is accomplished using a standard scene recognition algorithm [12]. We require workers to use our user interface to label the video.

2.1 User Interface

Our user interface (UI) is an interactive browser-based video player that guides workers in labeling a single entity in a sequence of video that is rendered by the client in real-time. Just-in-time compiler optimizations make this feasible in JavaScript.

Nonetheless due to the wide range of systems used by workers, we carefully manage frame caching to enable fast response even on low bandwidth connections. When the video player initializes in the client's browser, the UI immediately requests the first frame of the video from our servers. Subsequent frames are asynchronously buffered only until the next key frame. As many MTurk workers will open the task, but not accept it, we minimize bandwidth overhead by not downloading the entire video sequence. When the user presses "Play", the next frame in the buffer will appear on the screen, the previous frame is discarded from memory, and the next frame to be added to the buffer will begin downloading.

Our approach requires that all frames are individually rendered from the video sequences and saved to our server as JPEG encoded files. While this decompression requires significantly more storage, we are able to optimize the server to efficiently serve these JPEG images. Our custom video player removed artifacts introduced by competing implementations such as the Flash video codec. Long load times and compression artifacts initially prevented effective labeling by workers. Providing a JavaScript based solution also enables much wider participation across platforms and without the need for software licensing (as a MATLAB based solution would require).

Our video player (shown in Fig. 2) presents the user with a frame from the video sequence and instructs the user to draw a bounding box around a entity selected from a predefined list. In the case of basketball footage that we tested, we requested that all players, referees and the ball be tracked. After drawing the initial box, the video starts to play. The player software will automatically pause on regularly spaced key frames and prompt the user to label again. Workers annotated a single entity over the entire duration of the sequence.

As entities were annotated by multiple workers in parallel, the resulting tracks were merged into the video stream so that subsequent workers labeling the same video, viewed progressively more densely labeled video. Labeling was complete when multiple workers agreed that there were no more entities left to label.

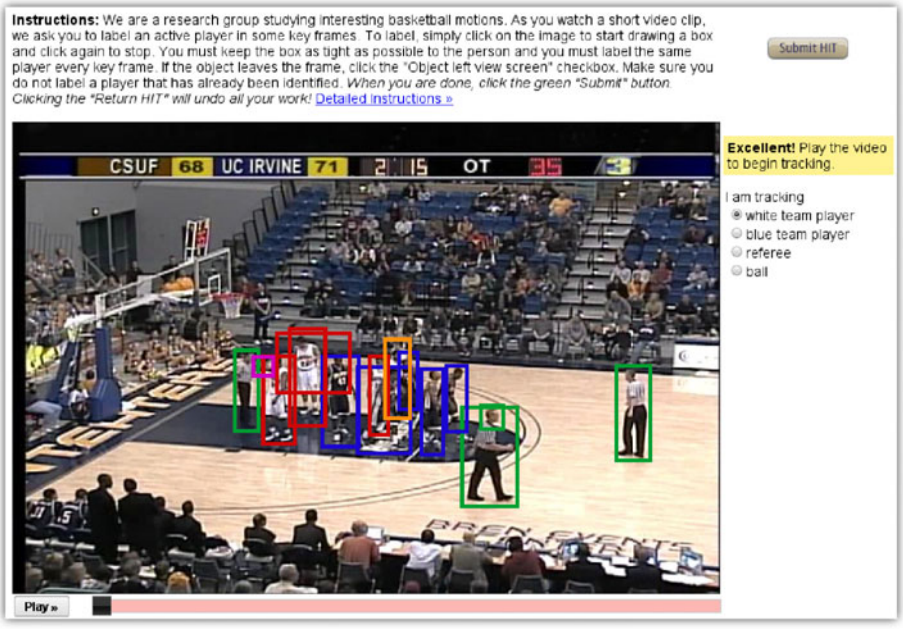


Fig. 2. Our video labeling user interface. All previously labeled entities are shown and the box the user is currently working with is bright orange.

Displaying other workers' labels unintentionally fostered a sense of community engagement that some of workers expressed in unsolicited comments.

> "Maybe it's more bizarre that I keep doing these hits for a penny. I must not be the only one who finds them oddly compelling–more and more boxes show up on each hit." — Anonymous subject

Mechanical Turk does not necessarily ensure quality work is produced. In fact, as a result of the low price of most HITs, many workers attempt to satisfy the HIT with the least amount of effort possible. Therefore it is very important that HITs are structured to produce desired results in a somewhat adversarial environment. One of the key criteria for the design of the UI is to make sure that producing quality work is no harder than doing the minimal amount of work to convince the UI that the HIT is completed. A second important criteria is to build into the evaluation process of a HIT an analysis of the validity of the work. A typical approach is to have multiple workers complete the same task until a statistical test demonstrates consensus on a single answer. A final important criteria is to design the interface so that it is difficult to successfully write an automated bot to get through the UI.

By requiring the user to annotate every key frame or explicitly say there is nothing left to annotate, we reduce the ease with which a worker can just "click-through" the interface without actually annotating anything. If they have to stop

at every key-frame anyway, they may as well do the annotation. Additionally, we automatically rejected any annotations that were degenerate. Typical examples we encountered were to label the first frame and then never move the box, or to have boxes that were very small or very large. It was clear that many of these degenerate tasks were attempted by malicious bots. To ensure the validity of our experiments, we manually validated all labellings, although the production system can test for statistical overlap of multiple workers before accepting the job. In our experiments we found that 35% of the labels had to be discarded as a result of attempts to trick our interface.

2.2 Dense Labeling Protocol

We instruct a worker to label one unlabeled entity in each sequence. After the worker finishes one job, their work is sent back to the server. Once sufficient non-degenerate labels are received to ensure confidence, the data is visually added to the video sequence so that subsequent labeling of the video reflects the new labels. A different worker can then realize that this entity is already labeled and decide to label another entity.

We decided to divide the labor in this manner because it introduces more diversity across workers. Each worker will always be shown a random sequence, so a single worker cannot do all the work for one sequence. Consequently, if a worker provides poor annotations, or shows a systematic bias, they will not taint all entities in a sequence with inadequate performance.

Complex videos of humans typically feature people suddenly appearing from occlusion, and entering and leaving the view frame. If the worker cannot find any unlabeled entities in the initial frame, the worker indicates that all entities of interest are currently labeled. The video player will then advance forward a few frames and ask the user to search for entities that may have entered the frame or appeared from occlusion. This cycle repeats itself until an unlabeled entity is found, or the end of the video is reached. If the video is exhausted and no people have been found, the user implicitly votes to finish the video. After enough people vote, the server stops spawning HITs for this video.

2.3 User Instructions

There are many possible points of failure with non-expert and non-malicious workers. For example, a user may change entities that they are labeling in the middle of the sequence, they may draw bounding boxes which are too large or too small, or they may improperly handle occlusions, entrances and exits of entities from the frame. In order to try to reduce poor labellings, the first time a worker views our HIT they are shown detailed instructions on how to operate our UI. However, workers do not invest time reading instructions because it takes time that they could otherwise be using to complete a different HIT. We attempt to get users started as quickly as possible by showing examples of accepted and rejected work, alongside detailed text descriptions. Workers can quickly dismiss the written instructions and at any point can return to them.

2.4 Video Server Cloud

We developed our server entirely in Python 2.6 and Cython, both open source and under OSI-approved licenses, so that the system can be deployed on enterprise-scale clusters and compute clouds. While MATLAB is the dominant tool for computer vision work, its license prevents cheap large-scale clustering that such massive amounts of annotation data require. During our experiments, we successfully distributed computation across 20 virtual CPUs using functional programming techniques.

3 Tracking and Interpolation

Vital to our analysis is the ability to properly interpolate between a sparse set of annotations. Our labeling tool requests that a worker labels the enclosing bounding box of an entity every T frames. Offline, we interpolate the object path between these key frames using a variety of algorithms. Because this is done offline, we can afford to use computationally expensive schemes. We define b to be the coordinates of a bounding box:

$$b = \begin{bmatrix} x_1 & x_2 & y_1 & y_2 \end{bmatrix}^T \tag{1}$$

We write b_t for the bounding box coordinate of an entity at time t. Without loss of generality, let us define the keyframe times to be time 0 and time T. We define the interpolation problem to be: Given b_0 and b_T, estimate b_t for $0 < t < T$.

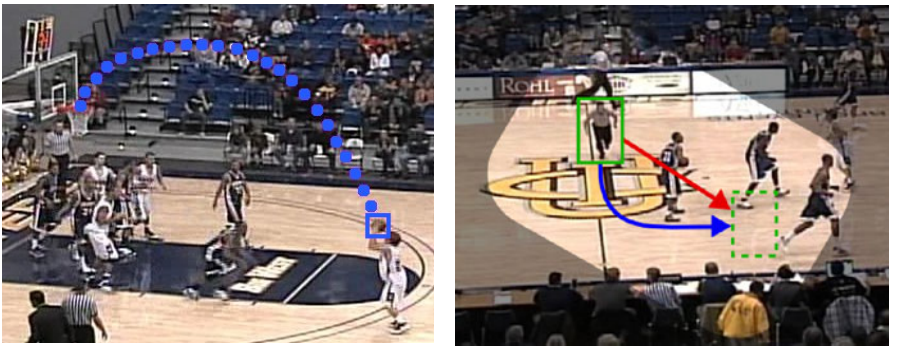
3.1 Linear Interpolation

The simplest approach is linear interpolation:

$$b_t^{lin} = \left(\frac{t}{T}\right) b_0 + \left(\frac{T-t}{T}\right) b_T \quad \text{for} \quad 0 \leq t \leq T \tag{2}$$

[7] makes the astute point that constant velocity in 3D does not project to constant velocity in 2D due to perspective effects. Assuming the tracked entity is planar, they describe a homography-preserving shape interpolation scheme that correctly models perspective projection. However, we found simpler linear interpolation to work well for many common scenes where there does not exhibit much depth variation.

Both of these interpolation algorithms are very efficient since they avoid the need to process any pixel data. However, both assume constant velocity which clearly does not hold for many entities (Fig.3a). We describe a dynamic-programming-based interpolation algorithm that attempts to *track* the location of the entity given the constraints that the track must start at b_0 and end at b_T. While there are many trackers available [13], we chose a simple method so we can focus on the economics of MTurk video annotation.



(a) Basketballs do not move linearly (b) The error tube on the linear path

Fig. 3. Nonlinear motion requires more sophisticated interpolation strategies for estimating entity position given the end locations. We employ a visual tracker in the error tube of the linear path in order to find the actual path (in blue) through visual analysis.

3.2 Discriminative Object Templates

To score a putative interpolation path, we need a visual model of the tracked object. We use all the annotated keyframes within a single video shot to build such a model. Let us assume N such keyframes exist, which in turn yield N bounding boxes which contain the entity of interest. Our first approach was to construct an average pixel-based template, and score it with sum-of-squared differences (SSD) or normalized correlation. We also experimented with more sophisticated templates built on histogram of oriented (HOG) [14] features. We found poor results with both.

We believe the suboptimal results arose from the fact that such templates are not designed to find objects in the cluttered backgrounds that we encountered. To compensate for this fact, we extract an *extremely* large set of "negative" bounding boxes collected from the N keyframes, making sure that they do not overlap the entity of interest in those frames. We then attempted to score a putative bounding box by competing an object template with an average background template. We again found poor results, this time due to the fact that our video backgrounds are complex and are poorly modeled with an average template.

Finally, we converged on the approach of learning a discriminative classifier trained to produce high scores on positive bounding boxes and low scores on the negatives. For each bounding box b_n we compute a feature descriptor composed of HOG and color histogram features:

$$\phi_n(b_n) = \begin{bmatrix} HOG \\ RGB \end{bmatrix} \tag{3}$$

We use a RGB color histogram with 8 bins for each dimension. Before extracting features, we resize the image patch at b_n to the "canonical" object size estimated

from the average of the N labeled bounding boxes. Given a collection of features along with labels $y_n \in \{-1, 1\}$ identifying them as positives or negatives, we learn a linear SVM weight vector w that minimizes the following loss function:

$$w^* = \text{argmin} \frac{1}{2} w \cdot w + C \sum_{n}^{N} \max(0, 1 - y_n w \cdot \phi_n(b_n)) \tag{4}$$

We use liblinear [15], which appears to be fastest linear SVM solver available. For typical size problems, training took a few seconds.

3.3 Constrained Tracking

Let us write the constrained endpoints given by the keyframes b_0^* and b_T^*. We wish to use the template w to construct a low-cost path $b_{0:T} = \{b_0 \dots b_T\}$ subject to the constraints that $b_0 = b_0^*$ and $b_T = b_T^*$. We score a path by its smoothness and the local classifier scores:

$$\underset{b_{1:T}}{\text{argmin}} \sum_{t=1}^{T} U_t(b_t) + P(b_t, b_{t-1}) \tag{5}$$

$$s.t. \quad b_0 = b_0^* \quad \text{and} \quad b_T = b_T^* \tag{6}$$

We define the unary cost U_t to the negative SVM score plus the deviation from the linear interpolation path. In order to reduce the penalty on occlusions, we truncate the cost by α_2:

$$U_t(b_t) = \min\left(-w \cdot \phi_t(b_t) + \alpha_1 ||b_t - b_t^{lin}||^2, \alpha_2\right) \tag{7}$$

We define the pairwise cost to be proportional to the change in position:

$$P(b_t, b_{t-1}) = \alpha_3 ||b_t - b_{t-1}||^2 \tag{8}$$

Note that the constraints in (6) can be removed simply re-defining the local costs to be:

$$U_0(b_0) = \text{inf} \quad \text{for} \quad b_0 \neq b_0^* \qquad \text{and} \qquad U_T(b_T) = \text{inf} \quad \text{for} \quad b_T \neq b_T^* \tag{9}$$

3.4 Efficient Optimization

Given K candidate bounding boxes in a frame, a naive approach for computing the minimum cost path would take time $O(K^T)$. It is well known that one can use dynamic programming to solve the above problem in $O(TK^2)$ by the following recursion [16]:

$$cost_t(b_t) = U_t(b_t) + \min_{b_{t-1}} cost_{t-1}(b_{t-1}) + P(b_t, b_{t-1}) \tag{10}$$

where $cost_t(b_t)$ represents the cost of the best path from $t = 0$ to b_t. We initialize $cost_0(b_0) = U_0(b_0)$. By keeping track of the argmin, one can reconstruct the minimum cost path ending at any node.

We limit K by restricting the set of putative bounding boxes at time t to lie within some fixed deviation from b_t^{lin}, both in terms of position and scale (Fig.3b). We found this pruning greatly improved running time and accuracy. We also note that the above recursion can be written as a min-convolution [17], allowing us to compute the optimum in $O(TK)$ using distance-transform speed-ups:

$$cost_t(b_t) = U(b_t) + \min_{b_{t-1}} cost_{t-1}(b_{t-1}) + \alpha_2 ||b_t - b_{t-1}||^2 \tag{11}$$

4 Results

We validate our online labeling framework by placing three different data sets of varying difficulty on MTurk. First, we look at "easy" videos of high-contrast entities moving in uncluttered backgrounds with little occlusion. We use YouTube sports footage of athletes performing running drills (as in Fig.5). Second, we look at a "marginally hard" task of annotating basketball players who tend to undergo a fair number of occlusions in quite cluttered backgrounds (as in Fig.1 and Fig.4). Finally, we consider the task of annotating "very hard" entities such as a basketball (as in Fig.3), which is hard to track due to frequent occlusions by players and the existence of large amounts of motion blur relative to its small image size. We use these data sets to examine cost trade-offs between automation and manual labeling as a function of the difficulty of the data. Our labeled basketball video data is unique for its size and complexity, and we plan to make it available to the community for further research on activity analysis.

We employ our previously described user interface to have workers annotate every fifth frame of our video sets, including a two-hour (210,000 frame) basketball game. We use this dense set of MTurk annotations as ground truth. We then hold out different intervals of the ground truth in order to determine how well the tracking and interpolation methods predict the missing annotations. We

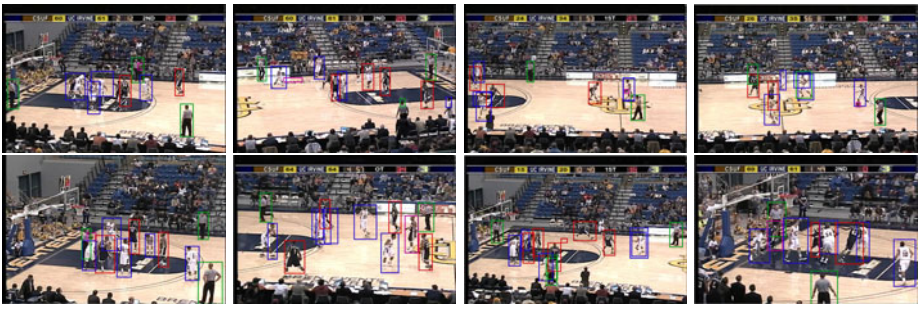


Fig. 4. Example of annotations from MTurk workers on difficult basketball footage. Red boxes are labeled as white team, blue as blue team, green as referee, and purple as the basketball. Worker quality is good, but not perfect.

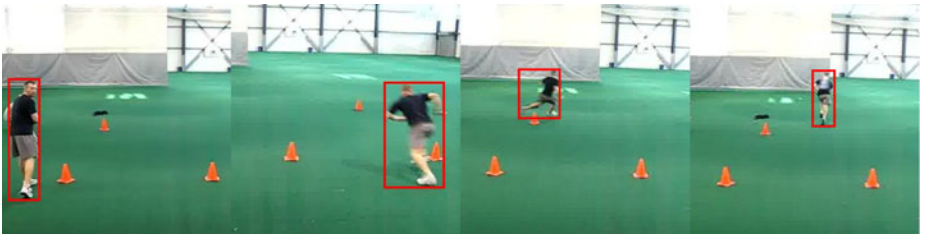
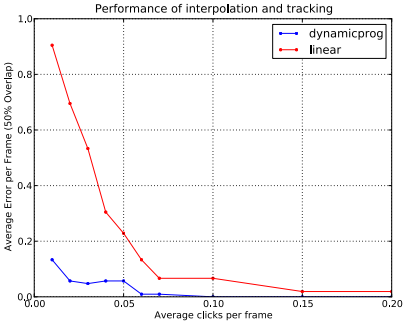
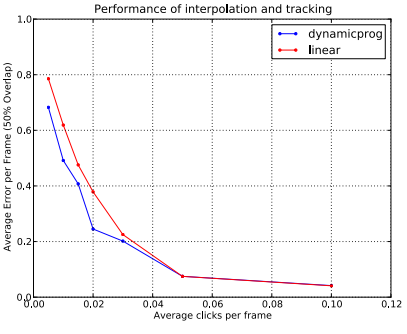


Fig. 5. An example of our "sports drill" data set containing nonlinear motion on uncluttered backgrounds

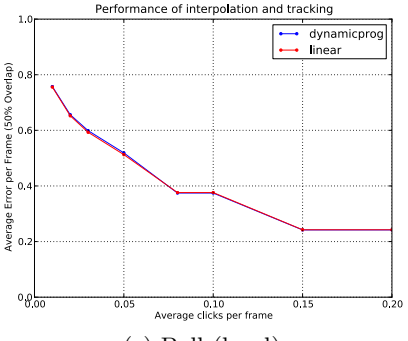
score our predictions with the same criteria as the PASCAL challenge: a prediction must overlap with the actual annotation by at least 50% to be considered a detection.



(a) Field drills (easy)

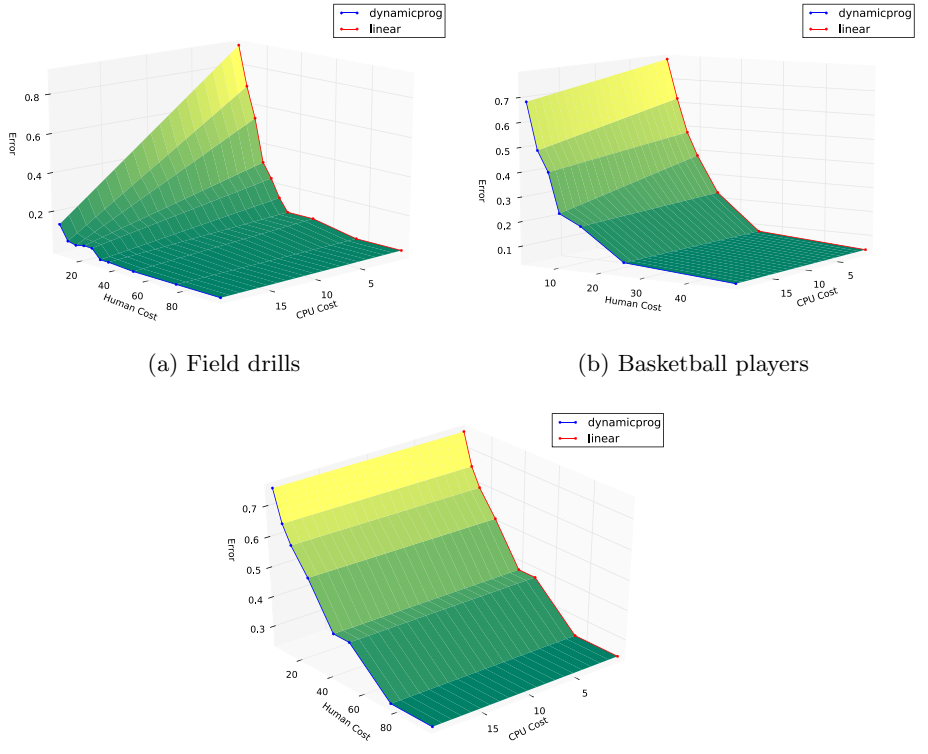


(b) Basketball players (medium)



(c) Ball (hard)

Fig. 6. Performance of dynamic programming and 2D linear interpolation on easy, marginally-difficult, and very-difficult data sets. Dynamic programming excels when features are easily extracted, such as in the field drill. But, dynamic programming performs equally well as linear interpolation when the object is highly occluded (such as a basketball).



(a) Field drills

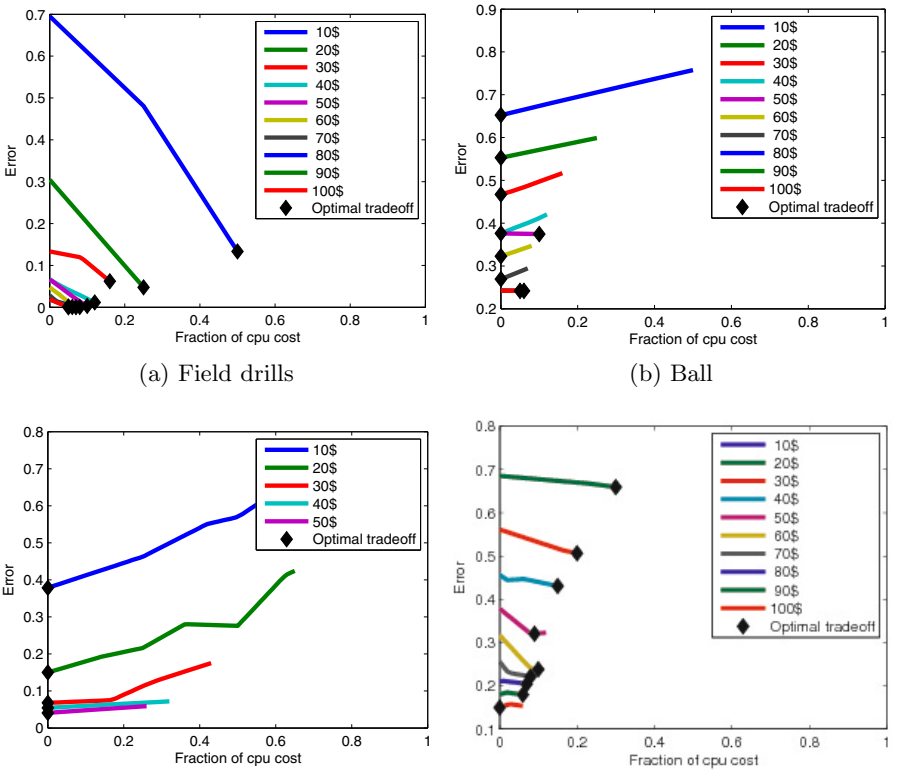
(b) Basketball players

(c) Ball

Fig. 7. Cost trade-off between human effort and CPU cycles. As the total cost increases, performance will improve. Cost axes are in dollars.

4.1 Diminishing Returns

We first confirm our hypothesis that the "Turk philosophy" (human computation is cheap and subsumes automated methods) does not hold for video because it is wasteful for users to annotate every frame. Fig. 6 shows a diminishing returns property in which increased human labeling (x-axis) results in smaller and smaller reductions in error rates (y-axis). Moreover, the rate of the diminishing is affected both by the difficulty of the video (easy, medium and hard) and the choice of interpolation algorithm (linear interpolation in red and dynamic programming in blue). For easy videos, we can achieve 10% error with a user annotation rate of 0.05 clicks per frame. For medium-difficultly videos, we require at least 0.1 clicks per frame regardless of the mode of interpolation. Finally, for difficult videos, we need 0.2 clicks per frame for the best accuracy. Our results suggest that interpolation of any kind can exploit the redundancy in video to reduce annotation effort by an order of magnitude compared to a naive, "brute-force" MTurk approach.



(a) Field drills (b) Ball

(c) Basketball players (d) Basketball players (alt cost)

Fig. 8. We show the cost trade-off between human effort and CPU cycles for different dollar amounts on different videos. In the "easy" field-drill video (**a**), the optimal trade-off is to maximize CPU usage for a fixed dollar amount. In the "very-difficult" ball footage (**b**), the optimal trade-off is to minimize CPU usage for a fixed dollar amount, essentially reducing to linear interpolation. Our most interesting trade-off results occur for the basketball footage. At our current MTurk rate of 2 cents per shot (**c**), the optimal trade off is to minimize CPU usage and use linear interpolation with frequent annotations. At a proposed MTurk rate of 10 cents per shot (suggested by many annotators) and half the CPU cost (likely achievable in the near future) (**d**), the optimal trade off is to maximize CPU usage for a fixed dollar amount.

4.2 CPU vs. Human Cost

We now consider how to optimally divide CPU effort vs human effort (frequency of annotation) so as to maximize track accuracy. We make three reasonable assumptions: linear interpolation is free in terms of CPU effort; tracking-based interpolation requires a fixed amount of computation effort regardless of annotation frequency; and we can smoothly increase CPU effort from 0% to 100% by choosing to linearly interpolate or use a dynamic programming tracker for each

key frame interval. If we wish α CPU effort then we randomly choose to use a dynamic programming tracker by flipping an α-biased coin.

We use rates from Amazon's Elastic Compute Cloud (EC2) platform to compute a monetized CPU cost. Our tracking algorithm will take 102 hours on EC2 to fully process a two hour basketball game. At US$0.17 per hour, the tracking costs US$17.34. We compute a monetized human cost by subtracting the amount we paid MTurk workers. We compensated workers US$0.02 per HIT to label every fifth frame, with a total of $483.20 to have MTurk label every object in every frame. We note, however, that such a rate does not appear to be sustainable – many workers loudly complained about this rate and demanded at least $0.10 per HIT. Finally, we compute the tracking error produced by various dollar amounts devoted to human labeling versus CPU usage. Not surprising, as we spend more money, label error decreases.

4.3 Performance Cost Trade-Off

We now consider our motivating question: how should one divide human effort versus CPU effort so as to maximize track accuracy given a X$? A fixed dollar amount can be spent only on human annotations, purely on CPU, or some combination. We express this combination as a diagonal line in the ground plane of the 3D plot in Fig. 7. We plot the tracking accuracy as a function of this combination for different X$ amounts in Fig. 8. We describe the trade-off further in the caption of Fig. 8.

5 Conclusion

Our motivation thus far has been the use of crowdsource marketplaces as a cost-effective labeling tool. We argue that they also provide an interesting platform for research on *interactive* vision. It is clear that state-of-the-art techniques are unable to automatically interpret complex visual phenomena. Our hypothesis is that by allowing a modest amount of human intervention, one can *can* successfully deploy vision algorithms which incrementally can measure and quantify progress for such difficult scenarios. To demonstrate this, our analysis in this paper focused on basketball sports footage. We note that there has been relatively little work in this domain, compared to tennis or soccer, perhaps because of clutter and the many occlusions. Indeed, we know of no algorithm that can correctly track the players in Fig. 1.

Acknowledgments. Funding for this research was provided by NSF grants 0954083 and 0812428. We thank the thousands of MTurk workers for their participation.

References

1. Sorokin, A., Forsyth, D.: Utility data annotation with amazon mechanical turk. Urbana 51, 61820 (2008)
2. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proc. CVPR, pp. 710–719 (2009)

3. Russell, B., Torralba, A., Murphy, K., Freeman, W.: LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision 77, 157–173 (2008)
4. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and Simile Classifiers for Face Verification. In: IEEE International Conference on Computer Vision, ICCV (2009)
5. Torralba, A., Fergus, R., Freeman, W.: 80 million tiny images: a large dataset for non-parametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 1958–1970 (2008)
6. http://www.moviearchive.org (2010)
7. Yuen, J., Russell, B., Liu, C., Torralba, A.: LabelMe video: Building a Video Database with Human Annotations (2009)
8. Vijayanarasimhan, S., Grauman, K.: Whats It Going to Cost You?: Predicting Effort vs. Informativeness for Multi-Label Image Annotations. In: CVPR (2009)
9. Liu, C., Freeman, W., Adelson, E., Weiss, Y.: Human-assisted motion annotation. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8 (2008)
10. Vijayanarasimhan, S., Jain, P., Grauman, K.: Far-Sighted Active Learning on a Budget for Image and Video Recognition. In: CVPR (2010)
11. Ross, J., Irani, L., Silberman, M.S., Zaldivar, A., Tomlinson, B.: Who are the crowdworkers? shifting demographics in mechanical turk. In: alt.CHI session of CHI 2010 Extended Abstracts on Human Factors in Computing Systems (2010)
12. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision 42, 145–175 (2001)
13. Avidan, S.: Ensemble tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 261–271 (2007)
14. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. I: 886–893 (2005)
15. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research 9, 1871–1874 (2008)
16. Bellman, R.: Some problems in the theory of dynamic programming. Econometrica: Journal of the Econometric Society, 37–48 (1954)
17. Felzenszwalb, P., Huttenlocher, D.: Distance transforms of sampled functions. Cornell Computing and Information Science Technical Report TR2004-1963 (2004)

Robust and Fast Collaborative Tracking with Two Stage Sparse Optimization

Baiyang Liu[1,2,⋆], Lin Yang[2], Junzhou Huang[1], Peter Meer[3], Leiguang Gong[4], and Casimir Kulikowski[1]

[1] Department of Computer Science, Rutgers University, NJ, USA
[2] Deptartment of Radiology, UMDNJ-Robert Wood Johnson Medical School, NJ, USA
[3] Department of Electrical and Computer Engineering, Rutgers University, NJ, USA
[4] IBM T.J. Watson Research, NY, USA

Abstract. The sparse representation has been widely used in many areas and utilized for visual tracking. Tracking with sparse representation is formulated as searching for samples with minimal reconstruction errors from learned template subspace. However, the computational cost makes it unsuitable to utilize high dimensional advanced features which are often important for robust tracking under dynamic environment. Based on the observations that a target can be reconstructed from several templates, and only some of the features with discriminative power are significant to separate the target from the background, we propose a novel online tracking algorithm with two stage sparse optimization to jointly minimize the target reconstruction error and maximize the discriminative power. As the target template and discriminative features usually have temporal and spatial relationship, dynamic group sparsity (DGS) is utilized in our algorithm. The proposed method is compared with three state-of-art trackers using five public challenging sequences, which exhibit appearance changes, heavy occlusions, and pose variations. Our algorithm is shown to outperform these methods.

1 Introduction

Tracking is to estimate the state of the moving target in the coming observed sequences. This topic is interesting for many industrial applications, such as surveillance, traffic monitoring, vehicle navigation, video indexing, etc. Accurate tracking of a general object in a dynamic environment is difficult due to the following challenges [1,2]:

- Dynamic appearance changes due to illumination, rotation, and scaling;
- 3D pose variations and information loss due to the projection;
- Partial and full object occlusions;
- Complex background clutters;
- Similar objects from the same class which lead to landmark ambiguity.

⋆ This research is completed when the author is a research assistant in the Department of Radiology in the UMDNJ-Robert Wood Johnson Medicial School.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 624–637, 2010.

Current tracking techniques can be categorized as discriminative or generative methods. Discriminative methods formulate the tracking as a classification problem[3,4,5,6]. The trained classifier is used to discriminate the target from background and can be online updated during the tracking procedure [7,8]. The generative methods represent the target observations as an appearance model [9]. The tracking problem is formulated as searching for the region with the highest probability generated from the appearance model [10,11,12,13,14,15,16]. It was proposed to update the target appearance model incrementally for adapting to dynamic environmental changes and target appearance variations. Generative models and discriminative models are combined and a one step forward prediction based collaborative tracking are proposed in [17].

Recently, sparse representations have been utilized in many areas [18,19,20,21] and successfully applied for tracking [22]. The tracking problem is formulated as finding a sparse approximation in the template subspace Φ. For candidate sample y, the general sparse problem can be formulated as

$$x_0 = argmin_x ||x||_0 \text{ subject to } ||y - \Phi x|| < \epsilon \tag{1}$$

where $||.||_0$ denotes the zero norm which represents the number of nonzero components and ϵ is the level of reconstruction error. However, it is well known that the l_0 optimization problem is NP-hard and there is no efficient algorithm to find the global optimum solution other than exhaustive search.

One class of algorithms tries to seek the sparsest solution by performing basis pursuit (BP) based l_1 minimization as

$$x_1 = argmin_x \ ||y - \Phi x|| + \tau |x|_1 \tag{2}$$

using linear programming instead of l_0 minimization in (1) [23]. This method is applied to solve l_1 minimization with none-negative constraints in [22]. The results are found to be efficient and adaptive to appearance changes, especially occlusion. However, there are still several problems exist:

- It is computationally expensive for very high dimensional data, which makes it unsuitable to use advanced image features for fast tracking applications.
- The background pixels in the target templates do not lie on the linear template subspace. The scale of the reconstruction error from background pixels is often larger than that from the target pixels, which might affect the accuracy of the sparse representation. It is therefore more reasonable to build the target template subspace *from the pixels belonging to the object*.
- The non-negative constraints, although can provide very good results when there are outliers, are vulnerable to complete tracking failures if wrong templates are selected.
- Temporal correlation between target templates and spatial relations among adjacent image features are not considered.
- Since the sparse parameter τ in (2) has no physical meaning, it is therefore difficult to tune up the parameter.

We observed that the target can usually be represented by templates sparsely and only part of the features, which can discriminate the target and background, are necessary to identify the target. Motivated by [22], considering existing problems and our observations, we proposed a robust and fast tracking algorithm with two stage sparse optimization. The algorithm starts from feature selection by solving a dynamic group sparsity (DGS) [24] optimization problem. The DGS is then performed on the selected feature space for sparse reconstruction of the target. These two sparsity problems are optimized jointly and the final results are obtained by Bayesian inference. According to our knowledge, this is the first study reporting fast and robust tracking algorithm using *two stage sparsity optimization*. The contributions of this paper are:

- A unified online updated sparse tracking framework which is targeted to use very high dimensional image features.
- The location adjacent features and time adjacent target templates tend to be selected as a group in our sparse representation, which provides more robust tracking results.
- The sparse parameters do have physical meaning and therefore are easy to be tuned.
- The algorithm is efficient. It is at least three times faster than the most current literature on sparse representation based tracking.
- Pose variation, appearance changes, and heavy occlusions are handled in our algorithm.

The paper is organized as follows: The related work is explained in Section 2. The tracking algorithm using two stage sparsity is presented in Section 3. Section 4 presents the experimental results. Finally, Section 5 concludes the paper.

2 Related Work

As online learning, sparse representation and dynamic group sparsity are intensively used in our algorithm, in this section we will give a brief review. Online adaptive tracking method has been intensively investigated in the recent literature. Grabner et al [7] propose to update the feature selection incrementally using the training samples gathered from current tracking result, which may lead to potential target drifting because of accumulated errors. Semi-online boosting [25] was proposed to incrementally update the classifier using unlabeled and labeled data together to avoid the target drifting. Multiple Instance Learning boosting method (MILBoosting) [4] put all samples into bags and labels them with bag labels. The positive bag is required to contain at least one real positive, while the negative bags have only negative samples. The drifting problem is handled in their method since the true target included in positive bag is learned implicitly. The target is represented as a single online learned appearance model in incremental visual tracking (IVT) [14]. As single appearance model is argued to be not sufficient to present the target in a dynamic environment, multiple appearance models are also proposed to be incrementally learned during the

tracking in [26]. Online updating is proven to be an important step in adaptive tracking and is also used in our algorithm.

Sparse representation was introduced for tracking in [22]. The target candidate is represented as a linear combination of the learned template set composed of both target templates and the trivial template which has only one nonzero element. The assumption is that good target candidate can be sparsely represented by both the target templates and the trivial templates. This sparse optimization problem is solved as a l_1 minimization problem with nonenegative constraints.

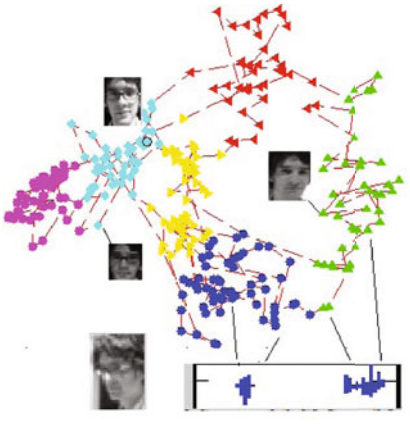


Fig. 1. The group structure of template feature vectors which can be clustered into six groups. The consecutive templates are connected with edges.

Another well known class of sparse optimization algorithms is the iterative greedy pursuit. The earliest algorithms including the matching pursuit [27] and orthogonal matching pursuit [28]. The subspace pursuit [29] and the compressive sampling matching pursuit [30] were proposed to reach similar theoretical recovery guarantees as the BP while reduce computational complexity. However, the nonzero components of the solution are not randomly distributed and tend to be clustered. Motivated by this prior, dynamic group sparsity (DGS) recovery algorithm is proposed in [24]. The algorithm includes five main steps in each iteration: 1) pruning the residue estimation; 2) merging the support sets; 3) estimating the signal by least square; 4) pruning the signal estimation and 5)updating the signal/ residue estimation and support set. The algorithm is similar to that of SP/CoSaMP [29,30] except considering the effect of neighbors also in the pruning process. DGS optimization also provides more robust result by forcing group representation which can eliminate wrong templates that do not fall in the same linear space as its neighbors. In Figure 1, we show the group structure of the consecutive learned templates in one of our testing tracking sequences. The image features are projected to two dimensional vector and clustered into six groups. In the bottom of Figure 1, we can tell that the target is sparsely represented by two groups. In other words,if one of the templates in the group is selected, its temporal adjacent templates tend to be selected too in our sparse representation using DGS.

3 Tracking with Two Stage Sparsity

We start this section from Bayesian tracking framework. The tracking algorithm is formulated as a two stage sparse optimization that is optimized jointly. The final results are obtained by Bayesian inference.

3.1 Bayesian Tracking Framework

Let affine parameters $\chi_t = (x, y, s, r, \theta, \lambda)$ represent the target state in the t-th frame, where x and y are the coordinates, s and r are the scale and the aspect, θ is the rotation angle, λ is the skew. The tracking problem can be formulated as an estimation of the state probability $p(\chi_t|z_{1:t})$, where z represents the observation in the previous t frames. Sequential Bayesian tracking based on Markovian assumption estimates and propagates the probability by recursively performing prediction

$$p(\chi_t|z_{1:t-1}) = \int p(\chi_t|\chi_{t-1})p(\chi_{t-1}|z_{1:t-1})d\chi_{t-1} \tag{3}$$

and updating

$$p(\chi_t|z_{1:t}) \propto p(z_t|\chi_t)p(\chi_t|z_{1:t-1}). \tag{4}$$

The transition model $p(\chi_t|\chi_{t-1})$ is constrained by assuming a Gaussian distribution $\mathcal{N}(\chi_t|\chi_{t-1}, \sigma)$. The observation model $p(z_t|\chi_t)$ represents the likelihood of z_t being generated from state χ_t.

In our algorithm, N candidate samples are generated based on the state transition model $p(\chi_t|\chi_{t-1})$. The state variables are considered as independent of each other. Each candidate sample I_i with state χ_t^i is reconstructed from the template library Φ using dynamic group sparsity (DGS). The likelihood $p(z_t|\chi_t^i) = exp(-\epsilon_i)$ where $\epsilon_i = min_\alpha||\Phi\alpha - I_i||$ is the optimized reconstruction error of I_i and α represents the sparse coefficients. Instead of solving the optimization problem in the full feature space, we propose to perform the sparse optimization in selected feature space with discriminative power. This enables us to use advanced high dimensional features without sacrificing the efficiency of the algorithm. Once the tracking state is confirmed, new samples are extracted and used to online update the training set and template library. The final result is obtained by maximizing $p(\chi_t|z_{1:t})$.

3.2 Two Stage Sparse Representation

Given the learned target template library $\Phi \in \mathbb{R}^{p\times m}$, where m is the number of templates and p is the dimension of the features. Let $\Phi_1 = [\Phi, I]$ and $\alpha_1 = \begin{bmatrix} \alpha \\ f \end{bmatrix}$ where α represents the sparse coefficient vector and f denotes the occlusion, the candidate sample y is sparsely reconstructed from Φ by minimizing the l_2 errors and finding α with K_1 nonzero components and f with K_2 nonzero components using greedy method:

$$\alpha_1 = argmin_{\alpha,f}\ ||\Phi_1\alpha_1 - y||_2, \text{while } ||\alpha||_0 \leq K_1 \text{ and } ||f||_0 \leq K_2. \tag{5}$$

Equation (5) can be solved efficiently when the dimension of the feature space and candidate searching space are small. However, it is computationally expensive for very high dimensional data, which make it unsuitable if advanced image

Algorithm 1. Tracking with two stage sparsity optimization

Input: Target's initial state χ_0, sparsity parameter K_0 for feature selection, K_1 and K_2 for target template and trivial template.

Initialize: Construct n training samples $\{X \in \mathbb{R}^{n\times p}, L \in \mathbb{R}^{n\times 1}\}$, where X is the sample matrix, L is the label and p is the dimension of the feature vector.

1. For each frame $t = 1 : T$ in the video where T is the total number of frames:
2. Perform DGS to solve $w^* = argmin_w ||Xw - L||_2$,
   subject to: $|w|_0 \leq K_0$ (when $t = 1$ we will use the initializations).
3. Construct diagonal matrix W, $W_{i,i} = \begin{cases} 1, w_i^* \neq 0 \\ 0, otherwise; \end{cases}$
4. Generate N candidate samples y_i in state χ_t^i.
5. For each $y_i, i = 1 : N$
6. Let $W' \in \mathbb{R}^{K_0\times p}$ as the matrix contains all non-zero rows of W,
7. $\Phi' = W'\Phi$, $y_i' = W'y_i$, and $f' = W'f$,
8. perform DGS to solve
9. $(\alpha^*, f^*) = argmin_{\alpha,f} \left|\left|[\Phi' \ W'] \begin{bmatrix} \alpha \\ f \end{bmatrix} - y_i'\right|\right|_2$, subject to: $\begin{matrix} ||\alpha||_0 \leq K_1 \\ ||f||_0 \leq K_2. \end{matrix}$
10. $\epsilon_i = ||\Phi'\alpha^* - y_i'||_2$.
11. $p(z_t|\chi_t^i) = exp(-\epsilon_i)$.
12. end for
13. $\chi_t^* = argmax_{\chi_t} p(\chi_t|z_{1:t})$.
14. Update the training set and template library with tracking results.
15. end for

features are used. Because only some of the features, which can discriminate the target and background, are necessary to identify the target, we argued that the effective dimension of the feature space can be decreased to K_0 dimension with diagonal matrix W. The number of nonzero components in W is not larger than K_0. The i-th feature is activated if W_{ii} is nonzero. Given n available samples $X \in \mathbb{R}^{n\times p}$ and their labels $L \in \mathbb{R}^{n\times 1}$, The joint sparse solution can be found:

$$\begin{aligned}(\alpha_1, W) = argmin_{\alpha_1,W} \ &\lambda||W\Phi_1\alpha_1 - Wy||_2 \\ &+\beta F(W, X, L) + \tau_1||\alpha_1||_1 + \tau_2||diag(W)||_1 \end{aligned} \tag{6}$$

where $F(W, X, L)$ is the loss function in the selected feature space for training dataset and samples in current frame. The τ_1 and τ_2 are the sparse parameters. As we explained before, the parameters τ_1 and τ_2 in (6) have no direct physical meaning and therefore it is difficult to tune their values. In our algorithm, we apply greedy algorithm to directly solve the original l_0 minimization problem for sparse representation. In this way (6) can be rewritten as:

$$\begin{aligned}(\alpha_1, W) = argmin_{\alpha_1,W} \ &\lambda||W\Phi_1\alpha_1 - Wy||_2 + \beta F(W, X, L), \\ \text{subject to: } &||diag(W)||_0 \leq K_0, ||\alpha||_0 \leq K_1 \text{ and } ||f||_0 \leq K_2. \end{aligned} \tag{7}$$

As it is hard to find an optimum solution for (6) when both α_1 and W are unknown, we solve (7) using two stage dynamic group sparsity optimization with greedy method. The first stage is to select the sparse set of features that

are most discriminative in separating the target from the background. Then the generative likelihood of each sample is estimated in the second stage with sparse representation. The details of the algorithm are shown in Algorithm 1. We will explain each stage in the following sections.

Feature selection. Given a set of training data $X = \{x_i \in \mathbb{R}^{1\times p}\}$ with $L = \{l_i\}, i = 1 \ldots n$ as the labels. The term $F(W, X, L)$ in equation 6 is defined as

$$F(W, X, L) = e^{-\sum_{i=1}^{n}(x_i w)l_i}, \tag{8}$$

where $w \in \mathbb{R}^{p\times 1}$ is a sparse vector. The j-th feature is selected if $w_j \neq 0$. The solution to minimize $F(W, X, L)$ can be found by solving the following sparse problem

$$w^* = argmin_w \, ||Xw - L||, \text{subject to: } ||w||_0 \leq K_0 \tag{9}$$

where K_0 is the max number of features will be selected. Here we want to emphasize that using greedy method for optimization, the parameter K_0 does have physical meaning corresponding to the number of features we plan to select. Considering Haar-like features, we do have the spatial relationship between neighborhood features. For example, if a small patch is occluded, the features extracted from this region will tend to be treated as a group in sparse optimization. Let $N_w(i, j)$ as the value of j-th neighbor of i-th feature, the support set is pruned based on Z

$$z_i = w_i^2 + \sum_{j=1}^{\tau} \theta_j^2 N_w^2(i, j), i = 1 \ldots p \tag{10}$$

in DGS taking the neighborhood relationship into consideration, where θ is the weight of neighbors. With the optimal w found by DGS, The diagonal matrix W can be constructed as

$$W_{j,j} = \begin{cases} 1, w_j^* \neq 0 \\ 0, otherwise; \end{cases} \tag{11}$$

Benefiting from the sparse solution to (9), we will be able to use advanced high dimensional features without sacrificing the efficiency of the algorithm. The other benefit is the object selection in the target region. The target templates usually contain some background features which are not linear. By doing discriminative feature selection, features from background pixels in the target templates are eliminated. The target template library is therefore more efficient and robust.

Sparse Reconstruction. After we calculate the weighting matrix W, the α and f in equation (6) can be found in the second stage

$$(\alpha, f) = argmin_{\alpha,f} \, ||W\Phi_1\alpha_1 - Wy||, \text{ subject to: } ||\alpha||_0 \leq K_1 \text{ and } ||f||_0 \leq K_2. \tag{12}$$

where $\Phi_1 = [\Phi, I]$ and $\alpha_1 = \begin{bmatrix} \alpha \\ f \end{bmatrix}$. Let $W' \in \mathbb{R}^{K_0 \times p}$ as the matrix contains all nonzero rows of W. We define $\Phi' = W'\Phi$ and $y' = W'y$. Please notify that in this step we already reduced the feature dimension from $p \times m$ to $K_0 \times m$ where m is the number of templates in the target library. In this stage the following equation is solved

$$(\alpha^*, f^*) = argmin_{\alpha,f} \left|\left| [\Phi',\ W'] \begin{bmatrix} \alpha \\ f \end{bmatrix} - y' \right|\right|, \text{ subject to: } \begin{matrix} ||\alpha||_0 \leq K_1 \\ ||f||_0 \leq K_2 \end{matrix}. \quad (13)$$

Here the sparsity parameters K_1 and K_2 have clear physical meaning, where K_1 controls the sparsity of a target template representation and K_2 controls the tolerance of occlusion. Then the likelihood of the testing sample y as target is $e^{-|\Phi'\alpha^* - y'|_2}$ and the final result is obtained by maximizing the $p(\chi_t|z_{1:t})$.

As we have already shown in Figure 1, the target templates have group structure and the temporally consecutive templates are likely to fall into the same group. The correct target sample can be reconstructed by sparse grouped templates. In our algorithm, we take into consideration the relationship between the template neighbors and tend to select grouped templates. This lead to a sparse vector in global but dense in local grouped consecutive templates. The l_1 minimization algorithm with non-negative constraints in [22] provides very sparse representation in template reconstruction coefficients, but it is vulnerable to outliers, namely, one single mistake in a template library can lead to complete tracking failure. For example, if a background sample is added into the template incorrectly, in an static background, it probably will have high matching likelihood since they are static most of time and can often find the perfect reconstruction. We avoid this problem in our algorithm by forcing a group selection of sparse coefficients. Since the outlier template is not in the same linear space as its neighbors, this can prevent it from being selected as it will lead to large reconstruction errors where even a standalone matching has a high score.

Once the tracking result is confirmed, the template library is incrementally updated as [22]. The samples with high likelihood and near the target are added to the training set as positive while the others are added as negative samples. This procedure is repeated for each frame in a whole sequence. The joint optimization of the two stage sparsity problem thus provides a fast, robust and accurate tracking result.

4 Experiments

The proposed tracking algorithm is evaluated using five challenging sequences with 3217 frames in total. The method is compared with three latest state-of-art tracking methods named L1 tracker(L1) [22], Incremental Visual Tracking (IVT) [14], Multiple Instance Learning(MIL) [4]. The tracking results of the compared algorithms are obtained by running the binaries or source code provided by their authors using the same initial positions. The source code of L1, IVT, MIL can

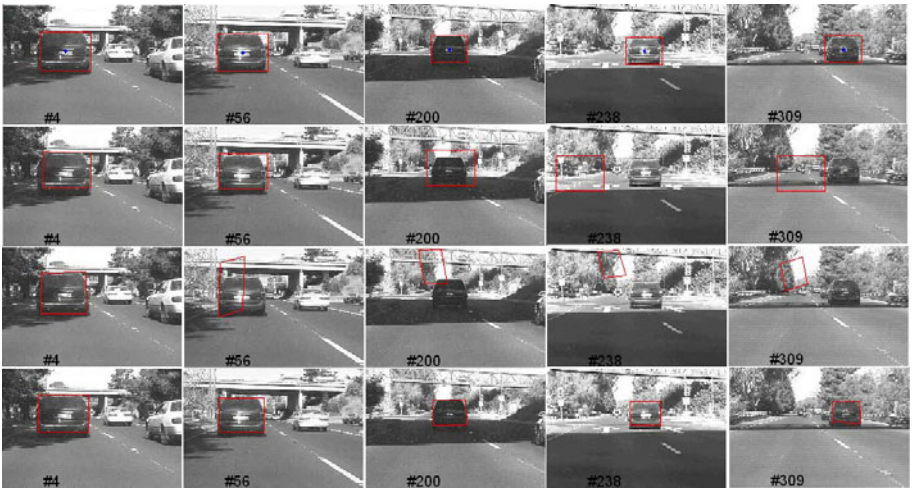


Fig. 2. The tracking results of a car sequence in an open road environment. The vehicle was driven beneath a bridge which led to large illumination changes. Results from our algorithm, MIL, L1, and IVT are given in the first, second, third, and fourth row, respectively.

be obtained from the URLs [1] [2] [3]. The first, second, third and fourth sequences were obtained from [14], and the fifth sequence was downloaded from [4].

In Section 4.1 we present the visual evaluation of the comparative tracking results. Several frames in five sequences are shown in the figures. Detailed quantitative evaluation of the comparative tracking are presented in Section 4.2. The tacking error-time curves of four sequences are plotted. Both visual and quantitative results demonstrate that our method provides more robust and accurate tracking results.

4.1 Visual Evaluation of Comparative Experiment Results

The first sequence was captured in an open road environment. The tracking results of the 4, 56, 200, 238, 309 are presented in Figure 2. The L1 starts to show some drifting on the 56-th frame. The MIL starts to show some target drifting (on the 200-th frame) and finally loses the target (the 238-th frame). IVT can track this sequence quite well. The target was successfully tracked using our proposed algorithm during the entire sequence.

The second sequence is to track a moving face. The 2, 47, 116, 173, and 222 frames are presented in Figure 4.1. The L1 algorithm fails to track the target when there are both pose and scale changes, shown in the 116-th frame. The MIL method can roughly capture the position of the object, but does have some target

[1] http://www.ist.temple.edu/ hbling/code_data.htm
[2] http://www.cs.toronto.edu/ dross/ivt/
[3] http://vision.ucsd.edu/ bbabenko/project_miltrack.shtml

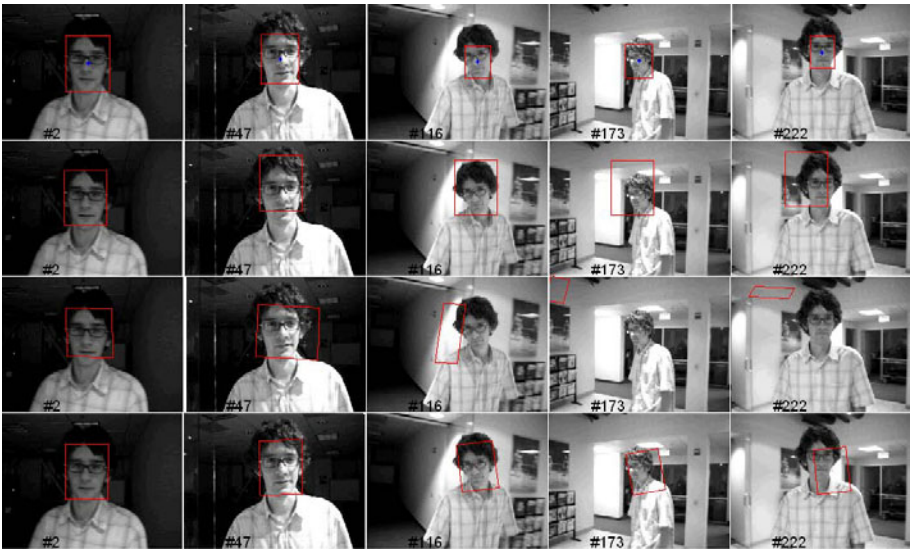


Fig. 3. The tracking results of a moving face sequence, which has large pose variation, scaling, and illumination changes. The order of the row sequences is the same as Figure 2.

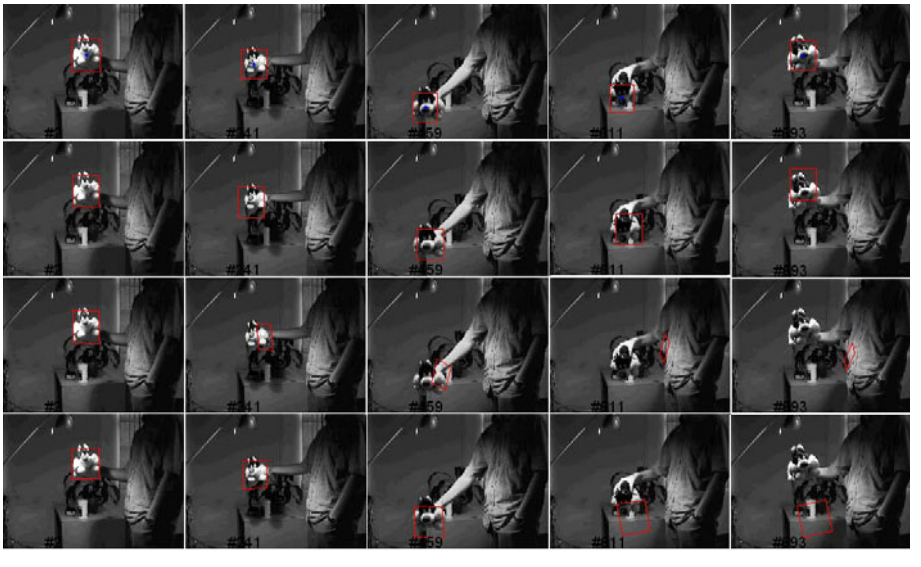


Fig. 4. The tracking results of a plush toy moving around under different pose and illumination conditions. The order of the rows is the same as in Figure 2.

drift problems, especially in the 173-th and 222-th frame. Our proposed two stage sparse tracking algorithm can track the moving face accurately through the whole sequence while the IVT produces some errors, especially on the 222-th frame.

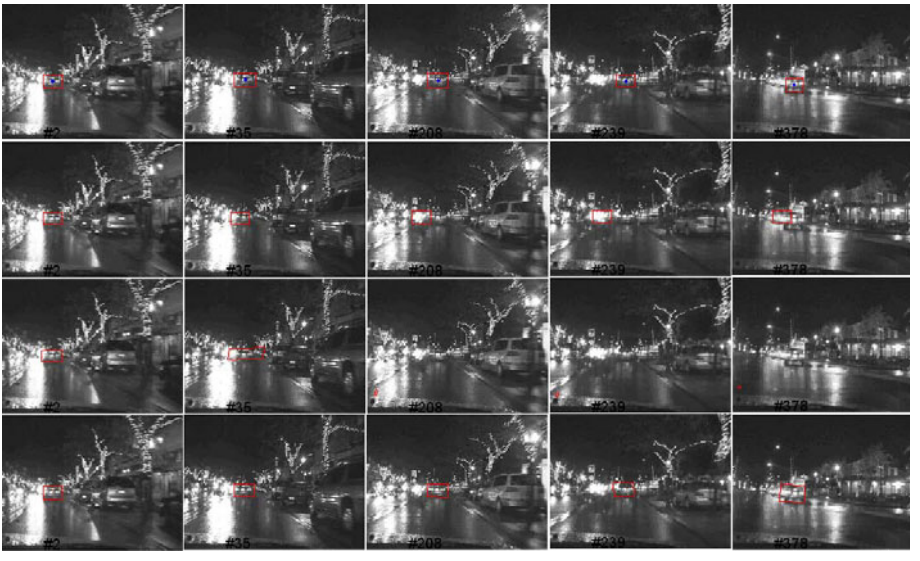


Fig. 5. The tracking results of the car sequence in a dark environment. This sequence has low resolution and poor contrast, which introduce some landmark ambiguity. The order of the row sequences is the same as Figure 5.

The third image sequence with frame 2, 241, 459, 611, and 693 is shown in Figure 6. The L1 method starts to have some drifting problem from roughly the 200-th frames, shown in the 241-th and 459-th frame. The MIL algorithm provides very good tracking results in this sequence. IVT fails to follow the object on the 611-th frame after major pose variation and can not be recovered. Our algorithm provides robust and accurate tracking result for this long sequence.

In the fourth sequence, the vehicle was driven in a very dark environment and captured from another moving vehicle. The 2, 35, 208, 239, 378 frames are presented in Figure 4.1. The L1 algorithm starts to fail to track the target from the 35-th frame. The MIL can roughly capture the position of the object before, but starts to have target drift problem from the 208-th frame distracted by light. IVT can track the target through the whole video sequence but it is not as accurate as our results, which can be found in the 378-th frame.

The results of the fifth sequence are shown in Figure 6. In this sequence we show the robustness of our algorithm in handling occlusion. The frame indexes are 10, 427, 641, 713, and 792. Starting from the 641-th frame, our method perform consistently better compared with the other methods.

4.2 Quantitative Evaluation of Comparative Experimental Results

For fair comparison, the tracking error e in each frame is measured as $e = \epsilon/d$, where ϵ is the offset of center from the ground truth and the d is the diagonal length of the target rectangle. For perfect tracking, the e should be equal to zero

Fig. 6. The tracking results of a face sequence, which includes a lot of pose variations, partial or full occlusions. The order of the row sequences is the same as Figure 2.

for each frame. In Table 1, we compared the quantitative e using our proposed algorithm with L1, MIL and IVT.

The best result in each column is shown in bold in Table 1. The missing column represents the number of frames where the $e > 1$. For a fair comparison, we do not count these failing frames when computing the *overall mean and variance* in the 7-th and the 8-th columns in Table 1. Measured by the public open benchmark, on average our algorithm only has 7% of drifting errors and never misses one single frame in the five tracking sequences which contain thousands of frames in total. In Figure 7 we present the tracking error-time curve. We can see that except for the fifth sequence, in which we obtain similar results as IVT (IVT will intend to shrink the window to very small size but won't lose the center of the target, as shown in Figure 6), our algorithm does outperform the other methods. The method is computationally efficient. Even using a MATLAB implementation, it can process two frames/second.

Table 1. The overall quantitative tracking performance comparison of proposed robust tracking method with two stage sparse optimization, L1 [22], MIL [4], and IVT [14].

	Mean					Overall				
	Seq1	Seq2	Seq3	Seq4	Seq5	Mean	Variance	Median	Max	Missing
L1	1.10	1.31	0.89	3.22	0.21	0.38	0.26	1.34	5.09	828
MIL	1.02	0.34	0.17	1.16	0.12	0.31	0.29	0.46	3.82	55
IVT	0.04	0.09	1.15	**0.07**	0.13	0.08	0.08	0.05	5.82	470
Proposed Method	**0.03**	**0.08**	**0.16**	0.08	**0.12**	**0.07**	**0.06**	**0.04**	**0.34**	**0**

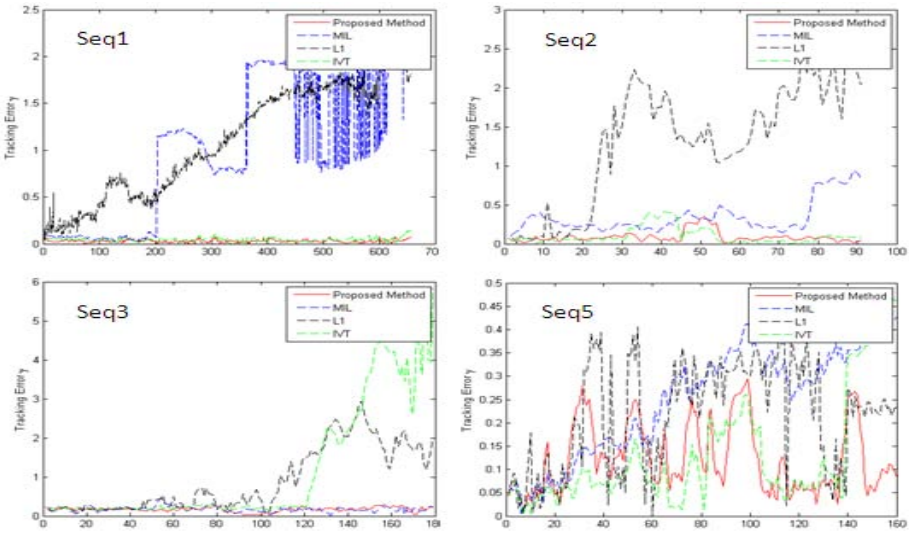


Fig. 7. The tracking accuracy e for each frame in four different sequences

5 Conclusion

We have proposed an online robust and fast tracking algorithm using a two stage sparse optimization approach. No shape or motion priors are required for this algorithm. Both the training set and the template library models are online updated. Two stage sparse optimization is solved jointly by minimizing the target reconstruction error and maximizing the discriminative power by selecting a sparse set of features. The experimental results demonstrate the effectiveness of our method in handling a number of challenging sequences.

Acknowledgement

This research is supported, in part, by UMDNJ Foundation Funding #66-09.

References

1. Yang, M., Wu, Y., Hua, G.: Context-aware visual tracking. PAMI 31, 1195–1209 (2009)
2. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38, 13–32 (2006)
3. Avidan, S.: Ensemble tracking. PAMI 29, 261–271 (2007)
4. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR (2009)
5. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV (2009)
6. Hess, R., Fern, A.: Discriminatively trained particle filters for complex multi-object tracking. In: CVPR (2009)
7. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR, vol. 1, pp. 260–267 (2006)

8. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: British Machine Vision Conference, vol. 1, pp. 47–55 (2006)
9. Matthews, I., Baker, S.: Active appearance models revisited. IJCV 60, 135–164 (2004)
10. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. IJCV 26, 329–342 (1998)
11. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. PAMI 24, 603–619 (2002)
12. Matthews, L., Ishikawa, T., Baker, S.: The template update problem. PAMI 26, 810–815 (2004)
13. Porikli, F., Tuzel, O., Meer, P.: Covariance tracking using model update based on Lie algebra. In: CVPR, vol. 1, pp. 728–735 (2006)
14. Ross, D., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. IJCV 77, 125–141 (2008)
15. Xue, M., Zhou, S.K., Porikli, F.: Probabilistic visual tracking via robust template matching and incremental subspace update. In: IEEE International Conference on Multimedia and Expo, pp. 1818–1821 (2007)
16. Zhou, S.K., Chellappa, R., Moghaddam, B.: Visual tracking and recognition using appearance-adaptive models in particle filters. ITIP 13, 1491–1506 (2004)
17. Yang, L., Georgescu, B., Zheng, Y., Meer, P., Comaniciu, D.: 3D ultrasound tracking of the left ventricle using one-step forward prediction and data fusion of collaborative trackers. In: CVPR (2008)
18. Gu, J., Nayar, S.K., Grinspun, E., Belhumeur, P.N., Ramamoorthi, R.: Compressive structured light for recovering inhomogeneous participating media. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 845–858. Springer, Heidelberg (2008)
19. Cevher, V., Sankaranarayanan, A., Duarte, M.F., Reddy, D., Baraniuk, R.G., Chellappa, R.: Compressive sensing for background subtraction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 155–168. Springer, Heidelberg (2008)
20. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis. In: CVPR (2008)
21. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. PAMI 31, 210–227 (2009)
22. Mei, X., Ling, H.: Robust visual tracking using l_1 minimization. In: ICCV (2009)
23. Donoho, D.: Compressed sensing. IEEE Transactions on Information Theory 52, 1289–1306 (2006)
24. Huang, J., Huang, X., Metaxas, D.: Learning with dynamic group sparsity. In: ICCV (2009)
25. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
26. Yu, Q., Dinh, T.B., Medioni, G.: Online tracking and reacquisition using co-trained generative and discriminative trackers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 678–691. Springer, Heidelberg (2008)
27. Mallat, S., Zhang, Z.: Matching pursuits with timefrequency dictionaries. IEEE Transactions on Signal Processing 41, 3397–3415 (1993)
28. Tropp, J., Gilbert, A.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on Information Theory 53, 4655–4666 (2007)
29. Dai, W., Milenkovic, O.: Subspace pursuit for compressive sensing: Closing the gap between performance and complexity. CoRR abs/0803.0811 (2008)
30. Needell, D., Tropp, J.: Cosamp: Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis (2008)

Nonlocal Multiscale Hierarchical Decomposition on Graphs

Moncef Hidane, Olivier Lézoray, Vinh-Thong Ta, and Abderrahim Elmoataz

Université de Caen Basse-Normandie, ENSICAEN, CNRS, GREYC Image Team,
6 Boulevard Maréchal Juin, F-14050 Caen Cedex France
{moncef.hidane,olivier.lezoray,vinhthong.ta,
abderrahim.elmoataz-billah}@unicaen.fr

Abstract. The decomposition of images into their meaningful components is one of the major tasks in computer vision. Tadmor, Nezzar and Vese [1] have proposed a general approach for multiscale hierarchical decomposition of images. On the basis of this work, we propose a multiscale hierarchical decomposition of functions on graphs. The decomposition is based on a discrete variational framework that makes it possible to process arbitrary discrete data sets with the natural introduction of nonlocal interactions. This leads to an approach that can be used for the decomposition of images, meshes, or arbitrary data sets by taking advantage of the graph structure. To have a fully automatic decomposition, the issue of parameter selection is fully addressed. We illustrate our approach with numerous decomposition results on images, meshes, and point clouds and show the benefits.

1 Introduction

It is well-accepted by now that vision is inherently a multiscale phenomenon. A visual task comes to representing and interpreting visual scenes with its singularities. As a consequence, a mathematical multiscale representation is essential to analyze images and decomposing an image into its meaningful components is one of the major tasks in computer vision and image processing. Images usually contain two types of main components: structure and texture. Structure is mainly the sharp edges in the image, which separate different objects. Texture is in general a repetitive pattern or the presence of oscillations (that can be noise). A typical image decomposition problem that is often considered is image denoising [2] where an image f is separated into a clean part u and a noisy part v. A popular scheme that proposed such a decomposition was the total variation decomposition of Rudin, Osher and Fatemi (ROF) [3] which was introduced as an effective denoising tool:

$$\inf_u \left\{ \int_\Omega |\nabla u| + \lambda \|v\|_{L^2}^2, f = u + v \right\} \tag{1}$$

where λ is a scale parameter. With the ROF model, as pointed out by Meyer [2], image denoising leads to image decomposition where the minimization of the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 638–650, 2010.

previous functional leads to a decomposition of f into a part u that extracts the edges of f, and a part v that captures the texture. Moreover, denoising at different scales λ generates a multiscale representation. In [1], Tadmor, Nezzar and Vese proposed a new multiscale image decomposition which offers a hierarchical and adaptive representation for different features in images. An image is hierarchically decomposed into the sum of simpler atoms u_k, where u_k extracts more refined information from the previous scale u_{k-1}. To this end, the atoms u_k are obtained as dyadically scaled minimizers of the ROF functional at increasing λ_k scales. Recently, the integration of nonlocal interactions in images [4] has shown to be very effective for capturing repetitive patterns in images and has led to numerous developments for image denoising (see [5] an reference therein). As it has been shown by Elmoataz *et al.* in [6], the formalism of difference equations on graphs provides a unifying view of local and nonlocal processing of functions on graphs (from images to high dimensional data). However, the integration of nonlocal interactions [7] has been few investigated for image decomposition.

Contributions. In this work, we propose to generalize the multiscale hierarchical decomposition of Tadmor, Nezzar and Vese (TNV) [1] proposed for images to a multiscale hierarchical decomposition of functions on arbitrary graphs. This naturally enables, first, to integrate nonlocal information in the decomposition process and second, to generalize it to the decomposition of functions on graphs. With the proposed approach, it is then possible to obtain a decomposition integrating nonlocal information for images, point clouds, or meshes (see [8] for an alternative aproach) once a graph representation is associated to the data under consideration.

Paper organization. In Section 2, we recall the multiscale hierarchical decomposition of images introduced in [1]. The formulation in this latter section is continuous. In Section 3, we introduce the discrete variational of [6] and give the numerical algorithms used for the solution. We then move on to the extension of the multiscale hierarchical decomposition to general data lying on graphs using the former framework. The approaches taken to construct weighted graphs are given. We then address the issue of parameter selection and give a general selection procedure. In the last Section, we illustrate the capabilities of the decomposition by applying it to different kinds of data defined on graphs. Grayscale, color, synthetic and natural images are considered as well as 3-dimensional meshes and point clouds.

2 Multiscale Hierarchical Decomposition of Images

The Tadmore Nezzar Vese multiscale hierarchical decomposition [1] builds upon the total variation minimization introduced in [3]. Given a noisy image f defined on continuous domain Ω, the authors in [3] consider the minimization (1) in order to recover the original denoised image. The starting point in [1] is an alternative point of view about the standard ROF denoising methodology. Inspired by the work of Meyer [2], Eq. (1) is interpreted as a decomposition: the

original noise-free image f is decomposed into a cartoon part u and an oscillatory or textured part v. The cartoon part should capture only the geometric features of the original image (i.e. sharp edges and boundaries) while the texture one should capture the repeated and meaningful small patterns. The trade-off between geometry and texture is dictated by the parameter λ which now plays the role of a scale separation parameter. The small meaningful patterns known as texture are in fact scale dependent: what might be seen as texture at a fixed scale could in fact contain important details when considered under a refined scale. To overcome this possible limitation, the authors in [1] propose to iterate the process in (1). Starting with an initial image f and an initial scale λ_0, a first decomposition is performed on f. Let u_0 be the regularized image and v_0 the residual so that $f = u_0 + v_0$. Then, another decomposition is performed, this time on the residual v_0 with a refined scale $\lambda_1 > \lambda_0$ leading to the following decomposition: $v_0 = u_1 + v_1$. We now have $f = u_0 + u_1 + v_1$. For a fixed index k and a sequence of refined scales $\lambda_0 > \lambda_1 > \ldots > \lambda_k$ we obtain the following decomposition:

$$f = u_0 + \ldots + u_k + v_k \ . \tag{2}$$

Results about the convergence as $k \longrightarrow \infty$ as well as upper and lower bounds for the choice of λ are provided in [1] for the geometric sequence $\lambda_i = 2^i$. Unfortunately, it is difficult to explicitly compute these bounds. Tadmor et al [1] argue that the choice of the parameter λ_0 is not important. If λ_0 is too small, the first decomposition levels, will be very smooth and details will not be reconstructed while λ_i is not large enough. While this is true for images with 4-adjacency grid graphs, it may be problematic for high dimensional data living on graphs mainly for two aspects: the dimension of the data can be high as well the nature of the processing (local or nonlocal). Therefore, a bad choice of λ_0 can produce higher computational cost for the decomposition since more levels are required. In [9], the hierarchical decomposition has been successfully applied to deblurring and denoising images as well as for segmentation.

3 Multiscale Hierarchical Decomposition on Graphs

In this Section, we recall the discrete variational framework introduced in [6] for the processing of general data defined on arbitrary graphs. Based on the ideas presented in [1], we use this framework to produce a multiscale decomposition of general data defined on graphs.

3.1 Digital Variational Framework

Definitions and Notations. Let $G = (V, E, w)$ be a general weighted graph consisting of a set of vertices $V = \{\alpha_1, \ldots, \alpha_N\}$ and set of weighted edges $E \subset V \times V$ with a similarity measure $w : V \times V \rightarrow [0, 1]$. For an edge (α, β) connecting the vertices α and β, $w(\alpha, \beta)$ represents a similarity measure between the vertices often based on an *a priori* distance measure. w is supposed symmetric and

satisfies $w(\alpha,\beta) = 0$ if $(\alpha,\beta) \notin E$. The graph G is assumed to be undirected, with no self-loops and no multiple-edges.

Let $f : V \to \mathbb{R}$ be a real-valued function. To measure the regularity of f, the authors in [6] use the following difference operator df:

$$(df)(\alpha,\beta) \stackrel{def}{=} \partial_\beta f(\alpha) \stackrel{def}{=} \sqrt{w(\alpha,\beta)}(f(\beta) - f(\alpha)),\ \forall(\alpha,\beta) \in E\ . \tag{3}$$

The discrete gradient operator is then defined at vertex α as follows:

$$\begin{aligned}\nabla_w f(\alpha) &\stackrel{def}{=} [\partial_\beta f(\alpha)\ :\ \beta \sim \alpha]^T \\ &= [\partial_{\beta_1} f(\alpha), \ldots, \partial_{\beta_k} f(\alpha)]^T,\ \forall(\alpha,\beta_i) \in E\ .\end{aligned} \tag{4}$$

Its Euclidean norm represents a local measure of the variation of f at a given vertex:

$$\begin{aligned}|\nabla_w f(\alpha)| &= \sqrt{\sum_{\beta\sim\alpha} (\partial_\beta f(\alpha))^2} \\ &= \sqrt{\sum_{\beta\sim\alpha} w(\alpha,\beta)(f(\beta) - f(\alpha))^2}\ .\end{aligned} \tag{5}$$

Another important operator is the nonlocal curvature graph operator which is defined as follows [6]:

$$\kappa_w^f(\alpha) = \tfrac{1}{2} \sum_{\beta\sim\alpha} \gamma_w^f(\alpha,\beta)(f(\alpha) - f(\beta))\ , \tag{6}$$

$$\gamma_w^f(\alpha,\beta) = w(\alpha,\beta)(|\nabla_w f(\beta)|^{-1} + |\nabla_w f(\alpha)|^{-1})\ . \tag{7}$$

Discrete Energy. Let f^0 be a real function defined on a weighted graph $G = (V,E,w)$. Usually f^0 is a corrupted version of a clean function g. The problem of recovering g from f^0 is a typical inverse problem. A standard method to solve this inverse problem is to consider it as a variational one. The variational formulation consists in the minimization of an energy functional involving a regularization term and a fidelity term. The energy considered here is :

$$E_w(f, f^0, \lambda) = \sum_{\alpha\in V} |\nabla_w f(\alpha)| + \tfrac{\lambda}{2}\|f - f^0\|^2\ , \tag{8}$$

where

$$\|f - f^0\|^2 = \sum_{\alpha\in V} |f(\alpha) - f^0(\alpha)|^2\ . \tag{9}$$

The first term of the right hand side in (8) is the regularization one, while the second one forces the solution to be closed to the initial data. The factor λ dictates the trade-off between these two terms. Then the minimization consists in finding:

$$\operatorname{arginf}\{E_w(f, f^0, \lambda),\ f : V \to \mathbb{R}\}\ . \tag{10}$$

For $w = 1$ the energy is the one proposed in [10], in the case of images modeled by 4 or 8-adjacency grid graphs. For $w \neq 1$ the formulation becomes an adaptive version of the digital TV filter [10]. If f^0 has its values in $\mathbb{R}^n$ then we consider an energy regularization for each component. In this case, the use of a global similarity measure plays the role of a correlation between the different channels and thus avoids the drawbacks of marginal processing [11].

3.2 Numerical Resolution

Restoration/Decomposition Equations. In this Section, we derive the restoration/decomposition equations that will serve to decompose general data defined on graphs. The discrete formulation of the energies leads to a set of algebraic equations which are the discrete equivalent of the Euler-Lagrange equation. Since the energy (8) is strictly convex, the optimal solution is given by taking the derivative:

$$\frac{\partial E_w(f, f^0, \lambda)}{\partial f(\alpha)} = 0, \quad \forall \alpha \in V. \tag{11}$$

In practice, in order to avoid zero division, we replace the Euclidean norm of the discrete gradient operator by a regularized version:

$$|\nabla_w f(\alpha)|_\epsilon = \sqrt{|\nabla_w f(\alpha)|^2 + \epsilon^2}\ ,$$

where ϵ is a small fixed number, typically $\epsilon = 10^{-4}$. This amounts to consider the following energy function:

$$E_w(f, f^0, \lambda) = \sum_{\alpha \in V} |\nabla_w f(\alpha)|_\epsilon + \tfrac{\lambda}{2}\|f - f^0\|^2\ . \tag{12}$$

This regularized energy is still strictly convex, and thus the solution is again given by taking the derivatives. In the sequel, we will drop the ϵ subscript, and $|\nabla_w f|$ will mean $|\nabla_w f|_\epsilon$.
The restoration/decomposition equations are then written (see [6]):

$$2\kappa_w^f(\alpha) + \lambda(f(\alpha) - f^0(\alpha)) = 0, \quad \forall \alpha \in V. \tag{13}$$

Algorithms. We use the linearized Gauss Jacobi method to find an approximate solution:

$$\begin{cases} f^{(0)} = f^0 \\ f^{(t+1)}(\alpha) = \dfrac{\lambda f^0(\alpha) + \sum_{\beta \sim \alpha} \gamma_w^{f^{(t)}}(\alpha, \beta) f^{(t)}(\beta)}{\lambda + \sum_{\beta \sim \alpha} \gamma_w^{f^{(t)}}(\alpha, \beta)}, \forall \alpha \in V, \end{cases} \tag{14}$$

where γ_w^f is defined in (7). Starting with the initial data f^0, each iteration of (14) relates the new value $f^{(t+1)}(\alpha)$ to $f^0(\alpha)$ and to a weighted average of the filtered data in the neighborhood of α. The contribution of f^0 is constant and dictated

by the scale parameter λ. As the decomposition of the successive residuals is performed, the scales are getting greater and the fidelity term has more impact on the decomposition.

For an unweighted graph, the algorithm (14) corresponds to the one proposed in [10] for the restoration of noisy images.

3.3 Digital Multiscale Hierarchical Decomposition on Graphs

In this Section, we propose a digital version of the multiscale hierarchical decomposition TNV, based on the framework introduced above. This digital version allows us to extend the TNV methodology to arbitrary graphs, including meshes and point clouds. The graph processing has another important advantage: it makes it possible to naturally consider nonlocal interactions between data [6] and thus enforces the nonlinearity of the decompositions.

Let us first review some aspects of the construction of weighted graphs. Data defined on graphs fall into two categories: organized and unorganized. For the former, the graph structure is known *a priori* while for the latter, a neighborhood graph should be considered. In this paper, we use the k-nearest neighbors graph where the nearest neighbors are selected according to a distance measure between vertices. Let V be a set of vertices, $E \subset V \times V$ a set of edges and f^0 a real function defined on V. The function f^0 is used to define a distance measure $d(\alpha, \beta)$ between each adjacent vertices α and β. For example, if f^0 is the function that assigns to each vertex its coordinates in the Euclidean space, we can use the Euclidean distance as a distance measure. If the graph represents an image domain and f^0 is the intensity function, one can consider the Euclidean distance between the intensity components for each pixel. If the data are organized, one can introduce feature vectors for each vertex and consider the distances between these feature vectors. A classical example is the introduction of patches around each pixel in the case of image denoising (see [6]). The graph now consists in of set of vertices, a set of edges and a distance measure d. A similarity measure w is defined, based on the distance d. w should be a non-increasing function that maps the range space of d to the interval $[0, 1]$. The ones we use in this paper are:

1) $w(\alpha, \beta) = \frac{1}{1+d(\alpha,\beta)}$

2) $w(\alpha, \beta) = e^{\frac{-d^2(\alpha,\beta)}{\sigma^2}}$

The estimation of σ is addressed later in this paper. In particular, we will consider local estimation of σ for each pixel in the case of image decomposition.

We have now a weighted graph $G = (V, E, w)$ and a function f^0 on V. We perform a first decomposition using equation (8) at an initial scale λ_0 and then iterate the process as in TNV, following the same geometric progression of scales. The graph framework extends the TNV method. It has the following advantages, in contrast to the TNV approach:

1) The nonlocal processing is embedded in the graph structure.
2) It introduces adaptation within the graph weights.
3) It allows the decomposition of any data on graphs (meshes, curves, ...).

The decomposition equations are already in digital form consisting in a system of nonlinear equations (13) for each decomposition step. For a fixed index k, we end up with the same decomposition as in (2). Once general graph topologies and weights are considered, our approach generalizes and extends the TNV approach.

3.4 Parameter Selection

In this Section, we propose a method to select the parameters of the digital multiscale hierarchical decomposition. The parameters that we consider here are the initial scale λ_0 and σ, the kernel bandwidth in the case of an exponential weight. For images, we propose a local estimation of σ at each vertex.

Image Data. For the nonlocal decomposition of images, we consider neighborhood graphs, mainly the k-nearest neighbors graph. In this case, we look for the k-nearest neighbors inside a window surrounding each pixel. This latter graph is coupled with a 4-adjacency grid graph, which will be denoted as k-NNG$_4$ in the sequel. For images we take $k = 25$ or $k = 10$. This choice allows for nonlocal processing while keeping the associated computational cost low. The edge weights $w(\alpha, \beta)$ play an important role in controlling the diffusion around each pixel. An efficient weight function should incorporate local informations embedded in the graph structure and the data. The exponential weight function is a good candidate for this. For each pixel, a 5×5 patch window is considered. More generally, for a $N \times N$ patch surrounding a pixel α, we take σ_α to be an estimation of the standard deviation of the intensity values in the patch. Such an approach amounts to consider the intensities as local independent random variables and to estimate their standard deviation. This standard deviation will serve to control the diffusivity of the regularization. We use an empirical estimation for the variance. If α is a pixel and B_α is the patch of size N_α surrounding α, then the estimate is:

$$\sigma_\alpha = \sqrt{\frac{1}{N_\alpha} \sum_{k \in B_\alpha} (f^0(k) - M_\alpha)^2} \ , \tag{15}$$

where M_α is the empirical mean computed inside the patch:

$$M_\alpha = \frac{1}{N_\alpha} \sum_{k \in B_\alpha} f^0(k) \ . \tag{16}$$

Then, $w(\alpha, \beta) = e^{\frac{-d^2(\alpha,\beta)}{\sigma_\alpha \sigma_\beta}}$. Other estimates can be used, in particular the median-based estimate used in [12].

For the parameter λ_0, a first approach could consist in using the analytical formula given in [10]. However, as the aim of the first minimization is to isolate the cartoon part, the parameter λ_0 can hardly be tied to an unconstrained

minimization as in the denoising case. Our aim here is the decomposition of the image and we do this by considering a global standard deviation estimated as in (15). This time we consider the entire image and let λ_0 be a multiple of $\frac{1}{\sigma}$. In our numerical experiments, we take $\lambda_0 = \frac{1}{2\sigma}$, so the texture below scale $\frac{1}{\lambda_0} = 2\sigma$ is unresolved in the first residual v_0. Here again more robust estimates can be used. For the case of color images, we use a single regularization for each color channel while the 1-Laplace operator is the same for all the components thus taking into account the correlation between the different channels.

Meshes. In the case of meshes, the graph structure is known *a priori*. The function to regularize is the one that maps each vertex to its coordinates in the Euclidean space. The decomposition can be seen as a preprocessing task. Indeed, it has the advantage of separating different detail levels in a structure generally containing several thousands of points. To introduce adaptation in the decomposition, we use the weight function $w(\alpha, \beta) = \frac{1}{1+d(\alpha,\beta)}$. Here $d(\alpha, \beta)$ is the Euclidean distance between α and β.

Point Clouds. The same methodology can be applied to unorganized sets of data. This time, the graph structure has to be constructed. When considering a neighborhood graph, one should check that the resultant graph is connected. If the k-nearest neighbors graph is choosen, the value of k has to fit this requirement.

4 Results

In this Section, we show some results of the proposed digital multiscale hierarchical decomposition on graphs. The approach is illustrated on grayscale and color images as well as on meshes and point clouds. The reader may consult the electronic version of the paper to accurately see the details in the provided decompositions.

4.1 Images

First, we illustrate the approach with a grayscale image. Figure 1 shows the 5 first levels of decomposition of an initial noisy finger image. Each reconstruction step is shown in a column in the second to fifth rows and the original image is in the first row. Level 1 corresponds to the first regularization u_0, level 2 to $u_0 + u_1$, until level 5, that shows $\sum_{i=0}^{4} = u_i$. Last column shows the last residual, in this case $v_4 + 128$. Results provided in the first row of Figure 1 correspond to the use of an unweighted 4-adjacency grid graph, i.e. $w(\alpha, \beta) = 1$ for all $(\alpha, \beta) \in E$ with λ_0 set to 0.02. One can see that this configuration produces blurred edges in the first steps of the decomposition. All the next rows correspond to results with our proposed extension. Third row presents results with a weighted 8-adjacency grid graph. The distance between two adjacent vertices is evaluated pixel-wise, an exponential weight is used with $\sigma = 40$, and λ_0 is set to 0.02. The use of

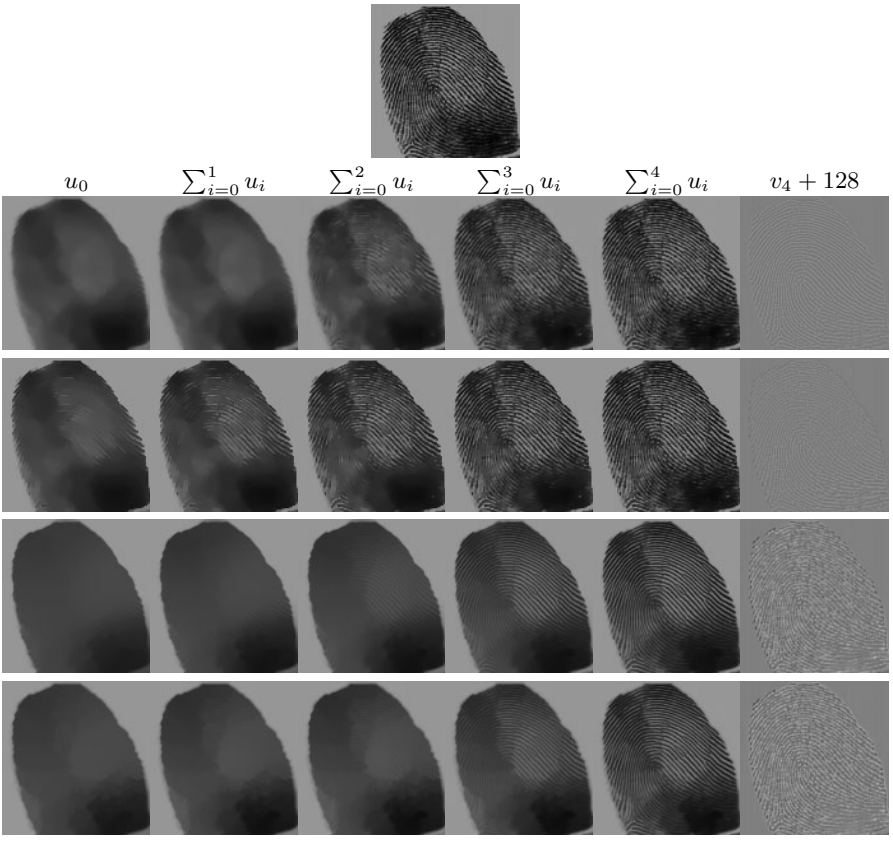


Fig. 1. Decomposition of a noisy finger image. Fist row: original image. Second row: 4-adjacency unweighted grid graph. Third row: 8-adjacency grid graph with exponential weight. Fourth row: 25-NNG$_4$ with exponential weight and 5×5 patches. Fifth row: 25-NNG$_4$ with exponential weight and 5×5 patches, σ_α is evaluated at each pixel, λ_0 is evaluated automatically. See text for more details.

a weighted graph enables an adaptive decomposition with less blur but much more texture is present in the first levels. Fourth row presents results with a weighted $25-\text{NNG}_4$. The distance between two adjacent pixels is evaluated by introducing 5×5 patches around each pixel. The weight function is exponential with σ evaluated globally and $\lambda_0 = \frac{1}{2\sigma}$. This time with nonlocal weights, few texture is preserved in the first levels and the edges of the structure are sharper. One can also see that the noise part in the initial image has not been obtained at the last level of the decomposition and can be found in the residual. The last row of Figure 1 uses the same graph structure as in the former one. The same exponential weight is used but this time, σ is local to each pixel and evaluated as detailed in Section 3.4 as well as for λ_0. We see that in this latter case, the texture has completely disappeared from the first level and only the geometric

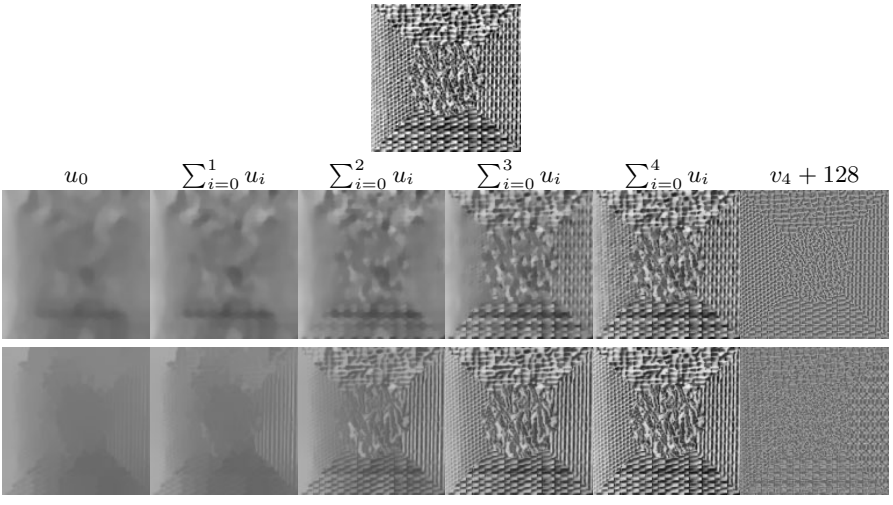


Fig. 2. Decomposition of a textured image. Fist row: initial image. Second row: 4-adjacency grid graph, unweighted. Third row: 10-NNG$_4$, exponential weight, σ_u is evaluated at each pixel, λ_0 is evaluated automatically. See text for more details.

part has been captured. Also, the reconstruction is more homogeneous, and each decomposition brings a distinct part of the texture, making the scale separation more explicit. This illustrates how a careful selection of the parameters (initial λ_0 and graph weights) is important to obtain an accurate nonlocal decomposition.

Figure 2 shows the results of the decomposition of a textured image. The initial image (first row) contains 5 different textures, each one with a different direction. The purpose here is to isolate the different texture regions across the levels. We also would like to verify that the coarser scale textures are reconstructed first. Second row presents a 5-step decomposition using a 4-adjacency grid graph with unweighted edges and $\lambda_0 = 0.005$. First level fails to separate the different textured regions, but some scale separation properties start appearing in level 3. Third row presents results obtained by considering a 10-NNG$_4$, with weighted edges (exponential), σ is computed locally as in Section 3.4 inside a 5×5 patch, and λ_0 is selected as in Section 3.4. In this example, the first level succeeds to capture an important aspect of the geometry. The reconstruction is faster and the scale separation is more explicit than in the previous row. Finally, the same comparison is performed on a color image in Figure 3. The decomposition has been considered for each color channel, with the same 1-Laplace operator for all the channels. Results in the first row are obtained with $\lambda_0 = 0.05$ on an unweighted 4-adjacency grid graph. Second row presents a nonlocal decomposition with parameters automatically estimated (inside a 5×5 patch for σ_α). Here again, the first level of the nonlocal adaptive decomposition succeeds in capturing the geometrical part of the initial image. Due to the variety of textures present in the initial image, the reconstruction in the case of the nonlocal decomposition is slower and one should look at finer scales in

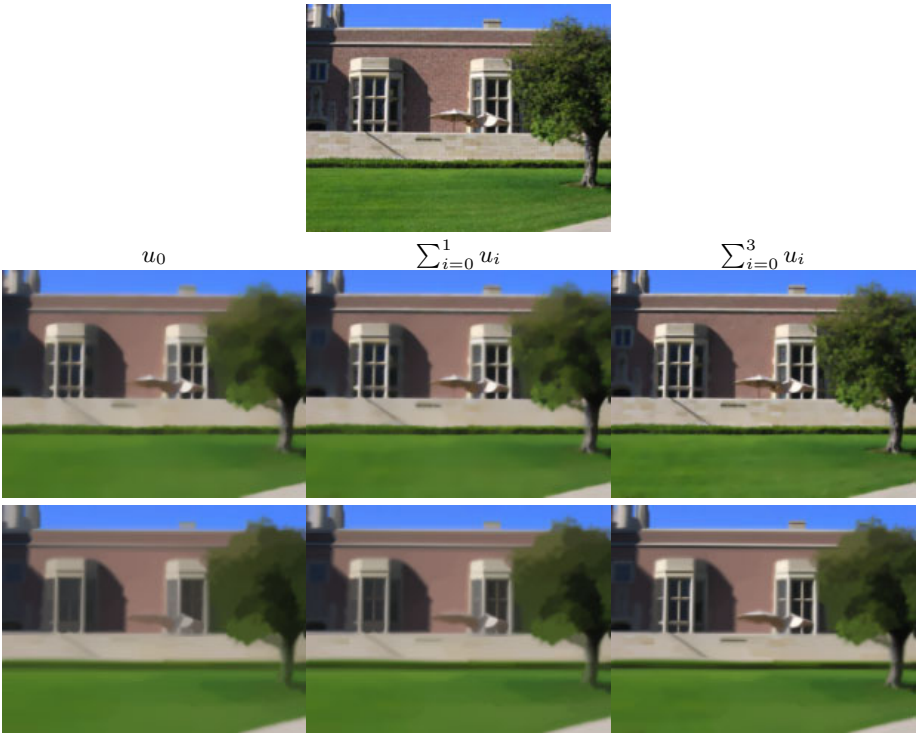


Fig. 3. Decomposition of a color image. Fist row: initial image. Second row: 4-adjacency grid graph, unweighted. Third row: 10-NNG$_4$, exponential weight, σ_α is evaluated at each pixel, and λ_0 is evaluated automatically.

order to completely recover the initial image. Indeed, with our nonlocal approach, repetitive structures (such as the window patterns) are considered as texture and removed from the first level that contains only the sole structure part of the image.

4.2 Meshes and Point Clouds

Here we provide numerical experiments illustrating the application of the multiscale hierarchical decomposition on graphs to meshes and point clouds. For meshes, the graph structure is given *a priori*. Figure 4 shows the results of a 10-step decomposition performed on a 3-dimensional hand mesh. The parameter λ_0 was set to 0.1, the distance between two vertices is the Euclidean distance, and the edges are weighted with: $w(\alpha, \beta) = \frac{1}{1+d(\alpha,\beta)}$. Rows 1 and 2 show two different views (interior and exterior) of the same hand mesh. The decomposition performs well on the first level by simplifying the mesh structure. Different details are recovered as the decompositions are performed. Finally, in Figure 5, we illustrate the denoising capability of the digital decomposition by applying it to

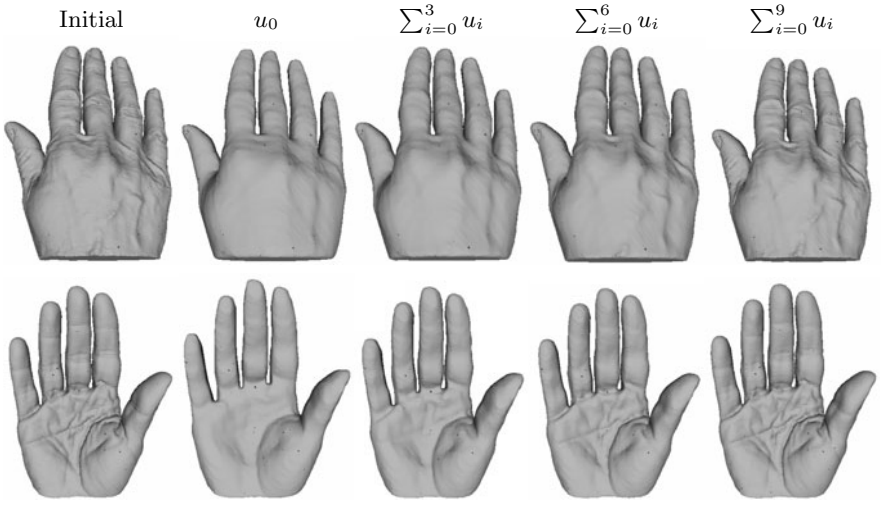


Fig. 4. Decomposition of a hand mesh. Initial scale $\lambda_0 = 0.1$. See text for more details.

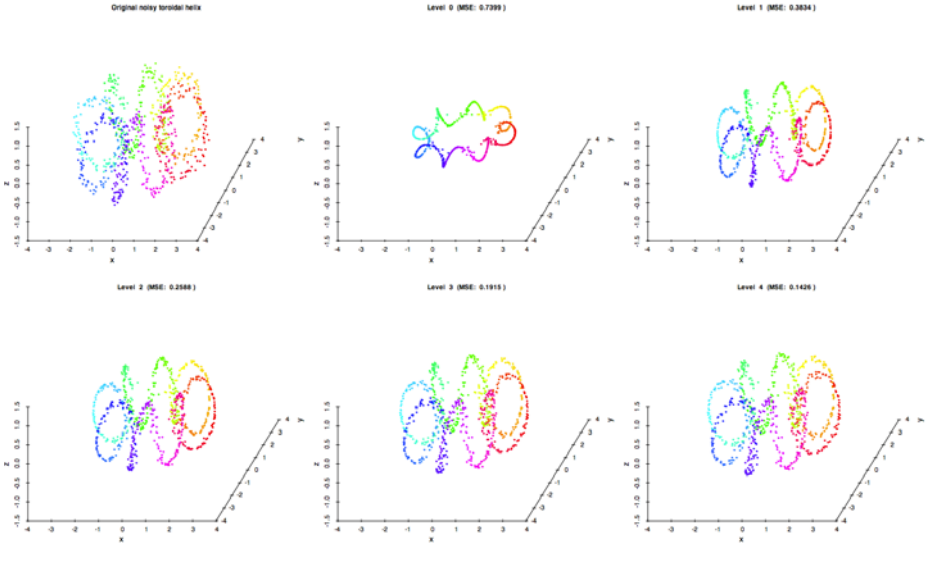


Fig. 5. Decomposition of a noisy toroidal helix. Initial scale $\lambda_0 = 0.1$, 20-nearest neighbors graph. See text for more details.

a noisy toroidal helix. A 20-nearest neighbors graph has been considered, λ_0 was set to 0.1, the distance between two vertices is the Euclidean distance and the edges are weighted with: $w(\alpha, \beta) = \frac{1}{1+d(\alpha,\beta)}$ where σ was set to the maximum of all the distances between two connected vertices. The first decomposition has

a strong simplification impact and reveals the structure of the helix. A denoised toroidal helix is recovered as the successive decompositions are performed.

5 Conclusion

In this paper, we generalized the hierarchical multiscale decomposition of Tadmor, Nezzar and Vese [1] to general data defined on graphs. The decomposition is based on a discrete variational framework. For images, the introducton of non-local interactions enables a finer decomposition. Moreover, since the proposed formulation considers arbitrary graphs, unusual domains such as meshes point clouds can be considered. Finally, we also have addressed the crucial problem of parameter selection to have a fully automatic decomposition.

References

1. Tadmor, E., Nezzar, S., Vese, L.: A multiscale image representation using hierarchical (BV, L2) decompositions. Multiscale Modeling and Simulation 2, 554–579 (2004)
2. Meyer, Y.: Oscillating Patterns in Image Processing and Nonlinear Evolution Equations. University Lecture Series. American Mathematical Society, Boston (2001)
3. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
4. Buades, A., Coll, B., Morel, J.M.: Nonlocal image and movie denoising. International Journal of Computer Vision 76, 123–139 (2008)
5. Buades, A., Coll, B., Morel, J.M.: Image denoising methods. A new non-local principle 52, 113–147 (2010)
6. Elmoataz, A., Lézoray, O., Bougleux, S.: Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. IEEE Transactions on Image Processing 17, 1047–1060 (2008)
7. Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. Multiscale Modeling and Simulation 7, 1005–1028 (2008)
8. Ohtake, Y., Belyaev, A., Seidel, H.-P.: A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In: Proceedings of the Shape Modeling International, p. 153 (2003)
9. Tadmor, E., Nezzar, S., Vese, L.: Multiscale hierarchical decomposition of images with applications to deblurring, denoising and segmentation. Communications in Mathematical Sciences 6, 281–307 (2008)
10. Chan, T.F., Osher, S., Shen, J.: The digital TV filter and nonlinear denoising. IEEE Transactions on Image Processing 10, 231–241 (2001)
11. Lezoray, O., Elmoataz, A., Bougleux, S.: Graph regularization for color image processing. Computer Vision and Image Understanding 107, 38–55 (2007)
12. Kervrann, C.: An adaptive window approach for image smoothing and structures preserving. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 132–144. Springer, Heidelberg (2004)

Adaptive Regularization for Image Segmentation Using Local Image Curvature Cues

Josna Rao[1], Rafeef Abugharbieh[1], and Ghassan Hamarneh[2]

[1] Biomedical Image & Signal Computing Lab,
University of British Columbia, Canada
[2] Medical Image Analysis Lab, Simon Fraser University, Canada
{josnar,rafeef}@ece.ubc.ca, hamarneh@cs.sfu.ca

Abstract. Image segmentation techniques typically require proper weighting of competing data fidelity and regularization terms. Conventionally, the associated parameters are set through tedious trial and error procedures and kept constant over the image. However, spatially varying structural characteristics, such as object curvature, combined with varying noise and imaging artifacts, significantly complicate the selection process of segmentation parameters. In this work, we propose a novel approach for automating the parameter selection by employing a robust structural cue to prevent excessive regularization of trusted (i.e. low noise) high curvature image regions. Our approach autonomously adapts local regularization weights by combining local measures of image curvature and edge evidence that are gated by a signal reliability measure. We demonstrate the utility and favorable performance of our approach within two major segmentation frameworks, graph cuts and active contours, and present quantitative and qualitative results on a variety of natural and medical images.

1 Introduction

Regularization plays a crucial role in improving the robustness and applicability of image segmentation techniques. Through the use of weighted regularization terms in conjunction with data fidelity terms, images plagued by high levels of deterioration, i.e. noise or poor edge contrast, are prevented from causing excessive irregularities and inaccuracies in the resultant segmentation. The vast majority of existing segmentation methods are predominantly based on parameter-laden optimization procedures designed to produce 'optimal' segmentations at their minimum. These methods commonly involve a highly sensitive tradeoff between the aforementioned regularization (smoothing) terms and data fidelity terms. Depending on how differently these competing energy terms are weighted, the resulting segmentation can greatly differ. Examples of widely used optimization-based segmentation methods with this sensitive tradeoff include active contours techniques [1,2,3,4], graph cut methods [5], optimal path approaches [6] and numerous variations thereof. In fact, addressing the issue of how to best balance competing cost terms is of great importance to many related algorithmic

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 651–665, 2010.

formulations in computer vision. More generally, this tradeoff is seen in likelihood versus prior in Bayesian methods [7] and loss versus penalty in machine learning [8].

Determining the optimum balance between regularization and adherence to image content has predominantly been done empirically and in an ad-hoc manner. However, natural and medical images commonly have objects which exhibit complicated and spatially varying boundary behavior, and often suffer from significant inhomogeneous image artifacts, e.g. the spatially varying bias field commonly observed in magnetic resonance (MR) images [9]. Compensating for such image deteriorations by uniformly increasing the level of regularization, until the most degraded region of the image is properly regularized, may result in excessive smoothing in those regions that do not require that much regularization. Subsequently, this results in a loss in segmentation accuracy, particularly for objects with highly curved boundaries. This commonly results in a painstaking and unreliable parameter-tweaking process.

Most reported approaches to segmentation keep a uniform level of regularization across the image or along an object boundary, i.e. one that does not vary spatially and is determined empirically. As addressed in McIntosh and Hamarneh [10], adapting the regularization weights *across a set of images* is necessary for addressing the variability present in real image data. Although an optimal regularization weight can be found for a single image in a set [10], the same weight may not be optimal for all regions of that image. In [11], a max-margin approach is used to learn the optimal parameter setting. In [12], Kolmogorov et al. solved the optimization problem for a range of parameters.

In recent years, spatially adaptive regularization has been acknowledged as a necessary requirement for improving the accuracy of energy-minimizing segmentations. In an earlier work [13], we proposed an adaptive regularization framework based on estimating the level of image reliability through local data cues reflecting both structure gradient and noise. Our approach in [13] demonstrated a clear advantage to spatially adaptive reliability-based regularization when compared to standard uniform regularization methods. Erdem and Tari [14] proposed a method for modulating diffusivity for Mumford-Shah segmentation approaches through the use of data-driven local cues and contextual feedback, specifically focusing on edge (gradient) consistency, edge continuity, and texture cues. Kokkinos et al. [15] proposed a spatially adaptive texture estimation measure through an amplitude/frequency modulation model of images that allows for a probabilistic discrimination between edges, textured and smooth regions. In [15], a texture cue, a loosely defined context-based classifier cue, and an intensity cue were used to distinguish between texture edges and edges between different objects. Only the latter edges were then used to dampen the curve evolution and define the segmentation boundary. Malik et al. [16] proposed Normalized Cuts to regularize segmentation in textured regions through the use of local texture and orientation cues. Gilboa et al. [17] presented a graph-cut based segmentation framework with spatially varying regularization through edge weights in the graph using a gradient magnitude-based cue.

These previous spatially adaptive methods focused on modulating regularization through local gradient direction and magnitude, texture, and noise estimation cues. In this paper, we advocate the need to integrate, for the first time, curvature cues into a spatially-adaptive regularization scheme. Object boundaries typically exhibit details of various scales, i.e. parts of the boundary can be smooth while other parts exhibit highly curved features. It is therefore inappropriate to enforce the same level of regularization in these different regions of varying degrees of curvature. In [18,19,20], for example, it was observed that high curvature points are anatomically and structurally important and thus form a good basis for feature matching over a set of data for image registration. Therefore, such high curvature parts of an object boundary should not be excessively regularized, otherwise important geometrical details are lost. The key idea of our approach is to decrease the regularization in reliable high curvature regions of the object to be segmented. To this end, we propose a new regularization scheme where structural curvature information calculated from the image is used to control the regularization and to better preserve the object shape in the presence of poor image quality.

It is important to distinguish our proposed curvature based spatial modulation of regularization from earlier works incorporating curvature, which fall under one of two classes. One class uses the *curvature of an evolving contour as an internal energy* to locally control the contour evolution in order to smoothen high curvature contour segments, e.g. [1,21]. The other class treats *image curvature as an external energy* in order to attract the evolving contour to high curvature regions in the image, e.g. [1,18]. In contrast, our proposed method uses estimates of *local image curvature to modulate the spatial regularization* by adaptively balancing the relative contributions of internal vs. external energies in the optimization process.

In summary, we propose a local image curvature-based structural cue that is robust to noise, is computed automatically, and does not require any prior knowledge or preprocessing steps. In order to showcase the utility of our approach, we incorporate this structural cue into two popular segmentation frameworks, graph cuts [22,23] and active contours [24]. We validate our method on real natural and medical images, and compare its performance against two alternative approaches for regularization: using the best possible spatially uniform (fixed) weight, and using a curvature-oblivious spatially adaptive regularization cue based on a signal reliability approach [13]).

2 Methods

Our regularization technique focuses on energy-minimizing segmentation, where the objective is to find a segmentation $C(x, y)$ that labels every pixel p in an image $I(x, y) : \Omega \subset \mathbf{R^2} \rightarrow \mathbf{R}$, e.g., object vs. background. We use an adaptive regularization weight $w(x, y) \in [0, 1]$ that varies across the image, and incorporate this weight into a general-form energy functional as follows:

$$E(C(x,y), w(x,y)) = w(x,y)E_{int}(C(x,y)) + (1 - w(x,y))\, E_{ext}(C(x,y)|I) \quad (1)$$

where E_{int} is the internal cost term contributing to the regularization of the segmentation in order to counteract the effects of image artifacts. E_{ext} is the external cost term contributing to the contour's attraction to desired image features, e.g. edges.

Our novel approach to balancing internal and external energy terms employs data cues that autonomously gauge and adapt the required level of regularization in local image regions. This required level of regularization should be different in two distinct scenarios: (a) high curvature boundary, and (b) poor image quality. The first situation requires low regularization to prevent loss of structural details, and the second situation requires high regularization to prevent erratic segmentation behavior due to noise. We set $w(x, y)$ in (1) such that reliable high curvature regions will have less regularization.

2.1 Local Image Curvature Cue

Let $I(x, y; \sigma) = G_\sigma(x, y) * I_o(x, y)$ be a smoothened image where $I_o(x, y)$ is the original image and σ is the Gaussian scale parameter. The unit vector tangent to the iso-intensity contour $C_I(x, y; \sigma)$ and passing through a point (x, y) is given as:

$$\mathbf{t}(x, y; \sigma) = \frac{1}{\sqrt{I_{x,\sigma}^2(x, y) + I_{y,\sigma}^2(x, y)}} \begin{bmatrix} I_{y,\sigma}(x, y) \\ -I_{x,\sigma}(x, y) \end{bmatrix} \tag{2}$$

where $I_{x,\sigma}$ and $I_{y,\sigma}$ are the image derivatives along x and y, respectively, at scale σ. Denoting the Hessian matrix of $I(x, y; \sigma)$ by $H_\sigma(x, y)$, the local image curvature $K(x, y; \sigma)$ can be calculated as [25,26]:

$$K(x, y; \sigma) = \left| \mathbf{t}^T H_\sigma \mathbf{t} \right|. \tag{3}$$

Note that we used the absolute value on the right hand side of (3) since we are not concerned with differentiating between convex and concave curvature. We follow the method in [27] where equation (3) is enhanced to have a stronger response near edges by multiplication with the gradient magnitude raised to some power, which we chose as 2. The enhanced curvature estimate becomes

$$\tilde{K}(x, y; \sigma) = \left| \frac{I_{y,\sigma}^2 I_{xx,\sigma} - 2 I_{x,\sigma} I_{y,\sigma} I_{xy,\sigma} + I_{x,\sigma}^2 I_{yy,\sigma}}{\sqrt{I_{x,\sigma}^2 + I_{y,\sigma}^2}} \right|. \tag{4}$$

To determine the curvature values of differently sized structures in the image, we automate the scale selection process by using the normalized scale coordinates of [27]. As the amplitude of the image spatial derivatives decreases with increasing scale, to compare the curvature values across different scales, the curvature must be scale-normalized. $\tilde{K}_{norm}$ is determined through scale-normalized coordinates $\xi = x/\sigma$. The normalized derivative operator with respect to ξ becomes $\partial_\xi = \sigma \partial_x$. Substituting the scale normalized coordinates into (4) results in the following normalized rescaled curvature:

$$\tilde{K}_{norm}(x, y; \sigma) = \sigma^3 \tilde{K}(x, y; \sigma). \tag{5}$$

After the curvature values at each scale have been normalized, the final curvature cue at every pixel is determined by selecting the scale at which $\tilde{K}_{norm}$ assumes a maximum value [27]:

$$K_m(x,y) = \max_{\sigma} \tilde{K}_{norm}(x,y,\sigma). \tag{6}$$

The curvature measure (6) is sensitive to noise and might inaccurately give rise to a strong response at non-structure, high-noise regions of the image. Following the concept of cue gating, as proposed in [16], where gradient information is suppressed in high texture regions, we thus define a noise-gated curvature cue, $K_G(x,y)$, that suppresses our curvature cue in high noise regions as follows:

$$K_G(x,y) = (1 - N(x,y))\, K_m(x,y). \tag{7}$$

$N(x,y)$ is a noise measure calculated using local image spectral flatness as follows [13]:

$$N(x,y) = \frac{\exp\left(\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \ln S\left(\omega_x,\omega_y\right) d\omega_x d\omega_y\right)}{\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} S\left(\omega_x,\omega_y\right) d\omega_x d\omega_y} \tag{8}$$

where $S(\omega_x,\omega_y) = |F(\omega_x,\omega_y)|^2$ is the power spectrum of the image, $F(\omega_x,\omega_y)$ is the Fourier transform of the image and (ω_x,ω_y) are the two spatial radian frequencies. $N(x,y)$ and $K_m(x,y)$ are normalized to the range $[0,1]$. This noise measure responds best to white noise-like patterns (i.e. occupying a very wide and flat spectrum).

2.2 Curvature-Based Regularization

Our noise-gated curvature cue in (7) is used to augment our noise-gated edge evidence $E_G(x,y)$ (proposed in [13]), which we also normalize to $[0,1]$:

$$E_G(x,y) = (1 - N(x,y))\, |\nabla I(x,y)|\,. \tag{9}$$

Both noise-gated, local image cues, curvature K_G and edge E_G, are now used to control $w(x,y)$ in (1). A meaningful way for setting $w(x,y)$ should satisfy the following requirements: (i) in highly trusted (noise-gated) edge evidence, little regularization is needed, regardless of the curvature strength; and (ii) in regions with low edge evidence, we set the regularization to be inversely proportional to the trusted (noise-gated) curvature, such that high curvature regions are not overly regularized. Note that 'high' curvature or edge evidence means a value close to 1 as all our cues are normalized. Thus, we form the adaptive weight as follows:

$$w(x,y) = 1 - E_G(x,y)^{(1-K_G(x,y))}. \tag{10}$$

If $E_G(x,y)$ is large (approaching 1), the exponent has little effect on the resulting weight, and requirement (i) is satisfied. If $E_G(x,y)$ is low and $K_G(x,y)$ is non-zero, the noise-gated edge evidence will be raised to a power $(1 - K_G(x,y)) \approx 0$, resulting in a lower $w(x,y)$, satisfying requirement (ii). Note that the detrimental effects from noise are handled by this model through the noise-gating of the cues. We refer to $E_G(x,y)^{(1-K_G(x,y))}$ as the curvature-modulated image reliability measure.

2.3 Incorporation of Texture Cue

In many natural images, large gradients and large curvature values can arise from texture edges rather than from edges representing object boundaries. To prevent texture edges from being included in the final edge set of the image, we must ensure greater regularization occurs in textured regions. We employ a texture measure $T(x,y)$ from Erdem and Tari [14] that estimates the probability of a pixel being near a texture edge:

$$T(x,y) = 1 - \exp\left(-\gamma(\min(\rho^1(x,y), \rho^2(x,y)))\right) \tag{11}$$

where γ is a decay rate parameter and $\rho^1(x,y)$ and $\rho^2(x,y)$ represent the p-values returned from the Wilcoxon Mann-Whitney test for sampled distributions lying between regions to the left and right of (x,y), and above and below of (x,y). If texture exists around (x,y), the differences between the distributions will be large and the resulting minimum p-values will be low, producing a low $T(x,y)$.

We incorporate the texture cue into our framework by modifying (9) to form the noise- and texture-gated edge evidence term as follows:

$$E_{G,T}(x,y) = T(x,y)\,(1 - N(x,y))\,|\nabla I(x,y)|\,. \tag{12}$$

Incorporating (12) into our spatially adaptive weight produces:

$$w(x,y) = 1 - E_{G,T}(x,y)^{(1-K_G(x,y))}. \tag{13}$$

2.4 Structural Cue Modulated Graph Cuts Segmentation

We first incorporated our adaptive weights[1] $w(p)$ into a graph cuts (GC) based segmentation [22,23]. The segmentation energy in this case becomes:

$$E(f) = \sum_{p,q\in\mathcal{N}} w(p)E_{int}(f_p, f_q) + \sum_{p\in P}(1 - w(p))E_{ext}(f_p) \tag{14}$$

where $f \in \mathcal{L}$ is the labeling for all pixels $p \in P$, $\mathcal{L}$ is the space of all possible labellings, and P is the set of pixels in image I. In GC, E_{int} is the interaction penalty between pixel pairs (i.e. the penalty of assigning labels f_p and f_q to neighboring pixels p and q), E_{ext} measures how well label f_p fits pixel p given the observed data, and $\mathcal{N}$ is the set of interacting pairs of pixels. $E_{ext}(f_p)$ is proportional to the difference between the intensity of p and the mean intensity of seeds labelled with f_p. $E_{int}(f_p, f_q) = 0$ if $f_p = f_q$ and 1 otherwise.

2.5 Structural Cue Modulated Active Contours Segmentation

We also implemented our proposed adaptive regularization within the popular active contours without edges (AC) segmentation framework by Chan and

[1] We use p to reflect graph vertices representing an image pixel at (x,y).

Vese [24]. The segmentation $C(x,y)$ in (1) is represented here via a Lipschitz function, $\phi(x,y) : \Omega \rightarrow \mathbf{R}$, where pixels interior to the zero-level set of ϕ are labeled as object and exterior pixels as background. The segmentation energy, $E(\phi)$ is given by:

$$\begin{aligned} E(\phi(x,y)) = \mu \int_{\Omega} \delta\left(\phi(x,y)\right) \left|\nabla\phi(x,y)\right| dxdy + \\ +\lambda_1 \int_{\Omega} \left|I(x,y) - c_1\right|^2 H\left(\phi(x,y)\right) dxdy + \\ +\lambda_2 \int_{\Omega} \left|I(x,y) - c_2\right|^2 \left(1 - H\left(\phi(x,y)\right)\right) dxdy \end{aligned} \tag{15}$$

where the first term is the internal (regularization) energy equal to the contour length, $\delta(z)$ is the dirac function and $H(z)$ is the Heaviside function. The latter two terms in (15) are external (data) terms, and c_1 and c_2 are the averages of $I(x,y)$ inside and respectively outside the zero-level set of ϕ [24]. λ_1, λ_2 and μ are constants that control the balance between smoothing and data adherence of the contour. We modified (15) to incorporate spatially adaptive regularization by replacing λ_1, λ_2 and μ with an adaptive convex weighting as follows:

$$\begin{aligned} E(\phi(x,y)) = \int_{\Omega} w(x,y)\delta(\phi(x,y)) \left|\nabla\phi(x,y)\right| dxdy + \\ + \int_{\Omega} \left(1 - w(x,y) + \epsilon\right) \left|I(x,y) - c_1\right|^2 H(\phi(x,y))dxdy + \\ + \int_{\Omega} \left(1 - w(x,y) + \epsilon\right) \left|I(x,y) - c_2\right|^2 \left(1 - H\left(\phi(x,y)\right)\right) dxdy \end{aligned} \tag{16}$$

where ϵ prevents a zero data force term (i.e. to prevent impeding curve evolution). We selected $\epsilon = 0.1$.

We minimize (16) with respect to $\phi(x,y)$ to determine the corresponding Euler-Lagrange equation for $\phi(x,y)$, which is derived in the supplementary material (see also [28]). We then solve for $\phi(x,y)$ iteratively by parameterizing the gradient descent with an artificial time $t \geq 0$ to produce the PDE for $\phi(t,x,y)$ as

$$\begin{aligned} \frac{\partial\phi}{\partial t} = \delta(\phi(x,y))\nabla w(x,y) \cdot \frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|} + w(x,y)\delta(\phi(x,y))\mathrm{div}\left(\frac{\nabla\phi(x,y)}{|\nabla\phi(x,y)|}\right) \\ -(1 - w(x,y) + \epsilon)\delta(\phi(x,y))\left[\left|I(x,y) - c_1\right|^2 - \left|I(x,y) - c_2\right|^2\right] = 0 \end{aligned} \tag{17}$$

where $\phi(0,x,y)$ represents the initial contour provided to the method.

3 Results and Discussion

Using MATLAB code on a PC with 3.6 GHz Intel Core Duo processor and 2GB of RAM, we ran a series of tests using a GC wrapper [22], and an implementation of AC [29], both of which were modified as proposed in Sections 2.4 and 2.5. We tested various natural images where structural features play an important

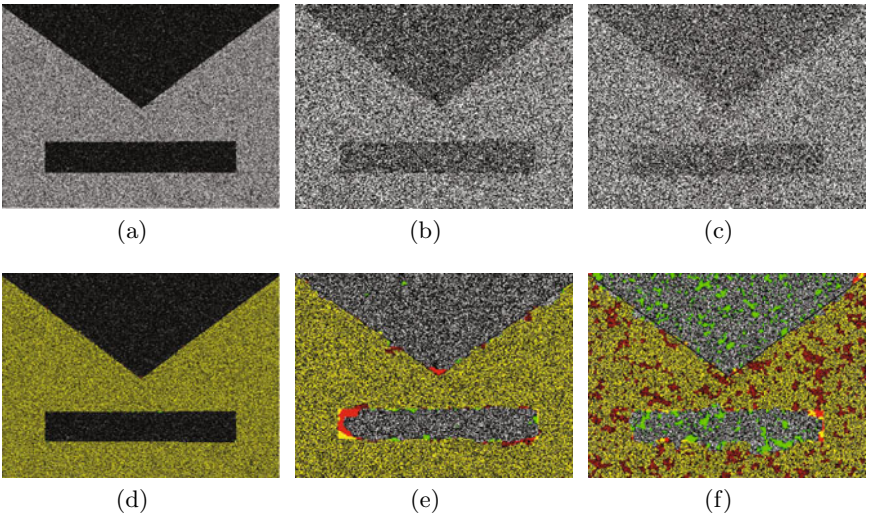
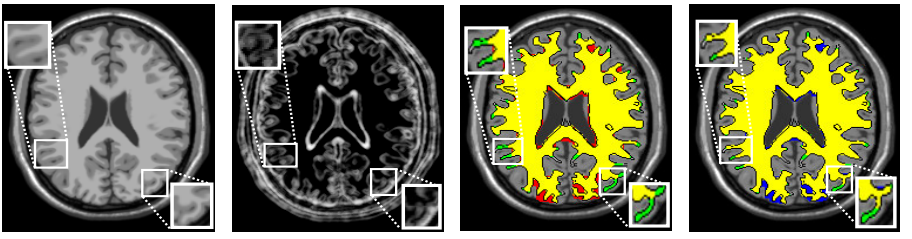


Fig. 1. (color figure) Segmentation of grey object in synthetic image corrupted by AWGN with increasing standard devation. (a), (b), (c) Original images with std. dev. of 0.05, 1.05, and 1.90, respectively. (d), (e), (f) Corresponding segmentations from the proposed adaptive weight (*green*) and the least-error fixed weight (*red*) where *yellow* regions are where the segmentations overlap. At the high noise level of (c), the segmentation (f) begins to form holes and inaccuracies.

role and which are available at the McGill Calibrated Color Database [30]. We also tested on magnetic resonance imaging (MRI) data from BrainWeb [31]. To demonstrate the advantage of our method, we compared against segmentation results from using the least-error fixed regularization weight, and against segmentation results from using a spatially adaptive regularization weight solely based in image reliability without any curvature-modulation [13]. For quantitative analysis, we used a set of 18 coronal brain MRI slices with ground truth segmentations and performed ANOVA to ascertain the improvements in segmentation accuracy afforded by our method. Computationally, the proposed method required less than a minute to calculate the regularization weights for a 768×576 image. For GC, a low number of random seeds (0.3% of image pixels for each label) were selected automatically by using the ground truth. For AC, we used an initial contour of a 50×50 square placed in the center of images, and used the same intial contour for comparison tests against the alternate methods.

We first analyze the robustness of the spatially adaptive regularization weight (Section 2.2). Figs. 1(a), 1(b), and 1(c) show synthetic images corrupted by increasing levels of average white Gaussian noise (AWGN). The GC adaptive weight segmentation for the images corrupted by noise levels of 0.05 and 1.05 std. dev. (Figs. 1(d) and 1(e), respectively) adheres to the corners of the object and does not leak outside of the object, unlike the fixed weight segmentation in red. At an extremely high noise level of 1.90 std. dev. shown in Fig. 1(c),



(a) (b) (c) (d)

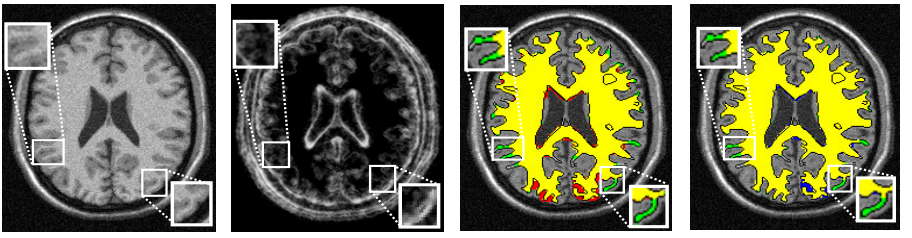
Fig. 2. (color figure) Segmentation of MR data from BrainWeb using GC with curvature-modulated regularization. (a) T1 slice with 20% intensity non-uniformity. (b) Curvature-modulated reliability calculated by our proposed method. Black intensities corresponds to 0 (low reliability/high regularization) and white to 1. Note higher reliability in cortical folds. (c) Comparison of segmentations from the proposed adaptive weight (*green*) to the least-error fixed weight (*red*), and (d) to the non-structural image reliability weight (*blue*). *Yellow* regions are where the segmentations overlap, and ground truth contour is shown in *black*. Proposed weights result in better segmentation of high curvature cortical folds (see *green*) with minimal leakage into the background, unlike other methods (see *red* and *blue* leakage regions).

the resulting adaptive weight segmentation (Fig. 1(f)) begins to show holes and degradation. Analysis of the Dice similarity coefficient between the adaptive weight GC segmentations and the the ground truth, for the synthetic image of Fig. 1 over various noise levels, showed that segmentation accuracy begins to drop at noise levels greater than 1.75 std. dev.

We next present results of GC segmentation with our proposed regularization framework on MR images from BrainWeb [31]. Fig. 2(a) shows a T1 image with an intensity inhomogeneity of 20%. High curvature-modulated reliabilty in the cortical folds (Fig. 2(b)) results in lower regularization in these regions. The overlayed GC segmentations (Fig. 2(c)) using the adaptive regularization weight versus the least-error fixed weight shows greater segmentation accuracy in high curvature regions. Additionally, the proposed method shows improvements over the existing non-structural image reliability framework (Fig. 2(d)).

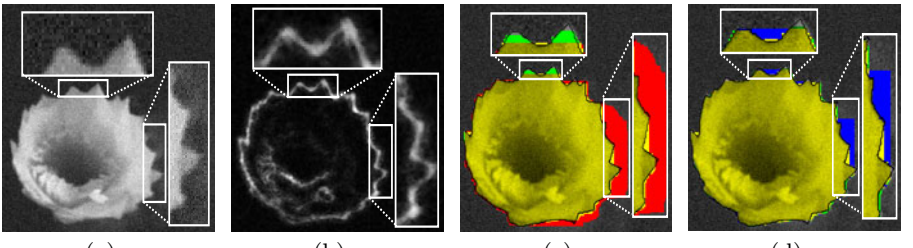
Fig. 3(a) shows the same T1 image of Fig. 2(a) but with a noise level of 7%. The resulting curvature-modulated reliability map (Fig. 3(b)) is not corrupted by the noise and still enforces greater regularization in high curvature cortical folds, as seen in the resultant segmentation comparisons of Fig. 3(c) and Fig. 3(d). At higher noise levels, our proposed curvature modulation results in a more accurate segmentation than the standard least-error uniform weight, and even more accurate than the noise-cue image reliability approach.

We also tested GC with our proposed regularization framework on a series of natural images, such as the flower shown in (Fig. 4(a)), where this image has been corrupted by AWGN with a standard deviation of 0.3. From this image, we produced the curvature-modulated reliability mapping in (Fig. 4(b)). The higher curvature-modulated reliability in the petal tip regions allows for a more



(a) (b) (c) (d)

Fig. 3. (color figure) Segmentation of noisy MR data from BrainWeb using GC with curvature-modulated regularization. (a) T1 slice with 7% noise level. (b) Curvature-modulated reliability. (c) Comparison of segmentations from the proposed adaptive weight (*green*) to the least-error fixed weight (*red*), and (d) to the non-structural image reliability weight (*blue*). Even in a high noise case, cortical folds are correctly segmented with proposed weights.



(a) (b) (c) (d)

Fig. 4. (color figure) GC segmentation of flower image . (a) Original image with AWGN of standard deviation 0.3. (b) Curvature-modulated reliability (higher in petal tip and crevice regions) (c) Comparison of segmentations from the proposed adaptive weight (*green*) to the least-error fixed weight (*red*), and (d) to the non-structural image reliability weight (*blue*) with overlapping regions in *yellow*. The proposed weights provided the best segmentation of the petal tips and had the least amount of leakage.

accurate segmentation when compared to the least-error fixed weight segmentation (Fig. 4(c)) and the non-structural image reliability weight segmentation (Fig. 4(d)) which, as expected, required higher regularization in the detailed petal tip regions, resulting in leakage into the background.

We demonstrate the AC segmentation with our regularization framework on the dandelion image of Fig. 5(a). Iterations were run until the contour evolution converged (at most 700 iterations). The low curvature-modulated reliability (Fig. 5(b)) in regions outside the flower prevents the resulting segmentation from including objects in the background, unlike the fixed weight segmentation (Fig. 5(c)) which leaked into the background of the image (see the red region). Additionally, the proposed method segments the petal tips more correctly than the non-structural image reliability segmentation as shown in Fig. 5(d) (where our segmentation in green captures all petal tips).

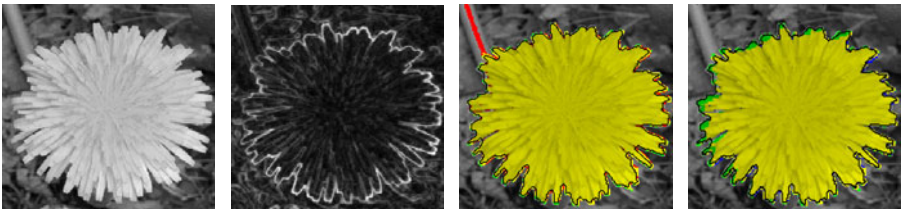


Fig. 5. (*Color figure, refer to e-copy*). Active Contours segmentation of a natural image. (a) Original image. (b) Curvature-modulated reliability calculated by our proposed method. (c) Comparison of segmentations from the proposed adaptive weight (*green*) to the least-error fixed weight (*red*), and (d) to the non-structural image reliability weight (*blue*). *Yellow* regions are where segmentations overlap. In (c), high regularization in the background prevents the segmentation from the proposed weights from leaking, unlike the fixed-weight method in *red*. In (d), only our segmentation in *green* captures all petal tip details.

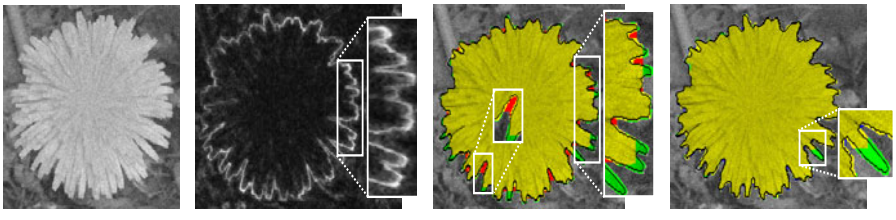


Fig. 6. (color figure) GC segmentation of natural image (a) Original image corrupted by AWGN with standard deviation of 0.3. (b) Curvature-modulated reliability calculated by our proposed method. (c) Comparison of segmentations from the proposed adaptive weight (*green*) to the least-error fixed weight (*red*), and (d) to the non-structural image reliability weight (*blue*) with overlapping regions in *yellow*. Proposed method provides best segmentation of high curvature petal tips and crevices with minimal leakage into the background.

We segmented the same dandelion again but with corruption by AWGN of standard deviation 0.3 (image values normalized to range between 0 and 1), as shown in Fig. 6(a). The curvature-modulated reliability (Fig. 6(b)) produces lower regularization weights in the petal tips and petal crevices. In Fig. 6(c), the fixed-weight segmentation excessively regularizes in the petal region, resulting in leakage (shown in red). Our method does not leak into the background and is able to capture the petal tips (shown in green). Similarly, in Fig. 6(d), the non-structural image reliability segmentation misses a few petal tips, which our method captured.

We demonstrate the ability of the texture-modulated weight defined in (13) (Section 2.3) to segment the textured image of Fig. 7(a) where we set the parameter γ in (11) to 0.1. The curvature modulated reliability (Fig. 7(b)) is erroneously

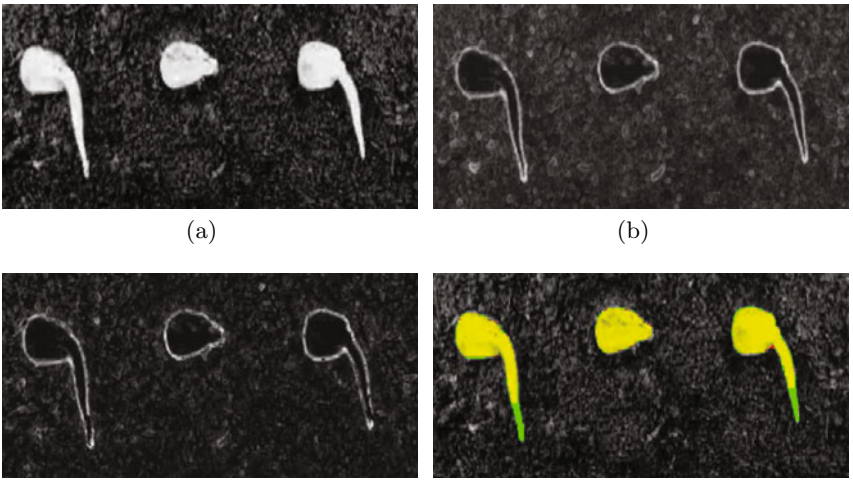


Fig. 7. (color figure) GC segmentation of textured natural image using the curvature-and-texture modulated weight. (a) Original image. (b) Curvature-modulated reliability calculated by our proposed method with no texture gating. (c) Curvature-and-texture modulated reliability. (d) Comparison of segmentations from the proposed adaptive weight (*green*) to the least-error fixed weight (*red*) with overlapping regions in *yellow*. Incorporation of a texture cue reduces leakage into the background, and proposed curvature cue reduces regularization in the protrusion region of the plant.

large for regions with texture. The curvature-and-texture modulated reliability shown in Fig. 7(c) is lower for the texture edges. The resulting GC segmentation is shown in Fig. 7(d). The higher curvature regions of the plant seedlings are accurately segmented by the adaptive weight due to lower regularization in these regions.

Quantitatively, we found significant improvements with the proposed method on the set of 18 brain MRI slices. Using the AC segmentation method with our proposed regularization framework to segment for cortical white matter, and validating with ground truth data, we found our method to produce an average Dice similarity of 78.4% (standard deviation of 0.0402) compared to 72.52% for the least-error fixed weight segmentation (std of 0.0728) and 60.68% for the non-structural image reliability segmentation (std of 0.1481). Our proposed method was significantly more accurate than the alternate methods with p-values $\ll$ 0.05. We performed GC segmentations of the cortical white matter on the same dataset and found an average Dice similarity of 89.91% (std of 0.0317) from our proposed method, compared to 86.20% (std of 0.0486) for the least-error fixed weight segmentation and 88.90% (std of 0.0331) for the non-structural image reliability segmentation. Again, our proposed method was significantly more accurate with all p-values $\ll$ 0.05. For each slice, we averaged 25 GC segmentations with random seed selections to determine the Dice similarity for

that slice. In addition, we investigated the effect of removing the noise-gating of the curvature measure. We found the resulting segmentation to be 25% less accurate when tested on the image of Fig. 6(a).

4 Conclusion

The key goal of our proposed method was to prevent excessive regularization of structurally important regions in energy minimization based segmentation. We presented a novel local curvature-based structural cue for modulating regularization. This cue was made robust to noise through gating by local signal reliability. Unlike current methods that employ curvature either as an internal or an external energy term, we use curvature to balance the competing internal and external energy terms. Accordingly, highly curved yet reliable boundary regions are spared from regularization.

We incorporated our proposed regularization framework into graph cuts and active contours for image segmentation. We demonstrated superior performance when compared to non-contextual regularization weights, as well as to adaptive, but curvature-oblivious, regularization cues. Quantitative and qualitative tests demonstrated that curvature-controlled regularization improves accuracy since it is common for image data to contain object boundaries of varying degrees of curvature. Future work will focus on extending our proposed cues and regularization framework to 3D image data, as well as adopting measures for other types of noise, e.g. spatially correlated non-Guassian noise.

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. of Comput. Vision 1, 321–331 (1988)
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. of Comput. Vision 22, 61–79 (1997)
3. Osher, S.J., Paragios, N.: Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer, Heidelberg (2003)
4. Pluempitiwiriyawej, C., Moura, J.M.F., Wu, Y.J.L., Ho, C.: STACS: New active contour scheme for cardiac MR image segmentation. IEEE TMI 24, 593–603 (2005)
5. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. Int. J. Comput. Vision 70, 109–131 (2006)
6. Barrett, W.A., Mortensen, E.N.: Interactive live-wire boundary extraction. Medical Image Analysis 1, 331–341 (1997)
7. Akselrod-Ballin, A., Galun, M., Gomori, M.J., Brandt, A., Basri, R.: Prior knowledge driven multiscale segmentation of brain MRI. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part II. LNCS, vol. 4792, pp. 118–126. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-75759-7_15
8. Zhao, P., Yu, B.: Stagewise lasso. Journal of Machine Learning Research 8, 2701–2726 (2007)
9. Samsonov, A.A., Johnson, C.R.: Noise-adaptive nonlinear diffusion filtering of MR images with spatially varying noise levels. Magnetic Resonance in Medicine 52, 798–806 (2004)

10. McIntosh, C., Hamarneh, G.: Is a single energy functional sufficient? Adaptive energy functionals and automatic initialization. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part II. LNCS, vol. 4792, pp. 503–510. Springer, Heidelberg (2007)
11. 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)
12. Kolmogorov, V., Boykov, Y., Rother, C.: Applications of parametric maxflow in computer vision. In: ICCV, vol. 8 (2007)
13. Rao, J., Hamarneh, G., Abugharbieh, R.: Adaptive contextual energy parameterization for automated image segmentation. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Wang, J.-X., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5875, pp. 1089–1100. Springer, Heidelberg (2009)
14. Erdem, E., Tari, S.: Mumford-Shah regularizer with contextual feedback. Journal of Mathematical Imaging and Vision 33, 67–84 (2009)
15. Kokkinos, I., Evangelopoulos, G., Maragos, P.: Texture analysis and segmentation using modulation features, generative models, and weighted curve evolution. IEEE PAMI 31, 142–157 (2009)
16. Malik, J., Belongie, S., Leung, T.K., Shi, J.: Contour and texture analysis for image segmentation. Int. J. of Comput. Vision 43, 7–27 (2001)
17. Gilboa, G., Darbon, J., Osher, S., Chan, T.: Nonlocal convex functionals for image regularization. UCLA CAM-report, 06–57 (2006)
18. Cohen, I., Ayache, N., Sulger, P.: Tracking points on deformable objects using curvature information. In: Sandini, G. (ed.) ECCV 1992. LNCS, vol. 588, pp. 458–466. Springer, Heidelberg (1992)
19. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. Int. J. of Comput. Vision 37, 151–172 (2000)
20. Glocker, B., Komodakis, N., Paragios, N., Navab, N.: Approximated curvature penalty in non-rigid registration using pairwise MRFs. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Wang, J.-X., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnação, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5875, Springer, Heidelberg (2009)
21. Evans, L.C., Spruck, J.: Motion of level sets by mean curvature. I. Journal of Differential Geometry 33, 635–681 (1991)
22. Bagon, S.: MATLAB wrapper for graph cuts (2006), http://www.wisdom.weizmann.ac.il/~bagon
23. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE PAMI 20, 1222–1239 (2001)
24. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. Image Processing 10, 266–277 (2001)
25. Kitchen, L., Rosenfeld, A.: Gray-level corner detection. Pattern Recognition Letters 1, 95–102 (1982)
26. Donias, M., Baylou, P., Keskes, N.: Curvature of oriented patterns: 2-D and 3-D estimation from differential geometry. In: ICIP, pp. I: 236–240 (1998)
27. Lindeberg, T.: On scale selection for differential operators. In: Scan. Conf. on Image Analalysis, pp. 857–866 (1993)

28. Rao, J., Abugharbieh, R., Hamarneh, G.: Adaptive regularization for image segmentation using local image curvature cues. Technical Report TR 2010-08, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada (2010)
29. Wu, Y.: Chan Vese active contours without edges (2009), http://www.mathworks.com/matlabcentral/fileexchange/23445
30. Olmos, A., Kingdom, F.A.A.: McGill calibrated colour image database (2004), http://tabby.vision.mcgill.ca
31. Cocosco, C.A., Kollokian, V., Kwan, R.K.S., Evans, A.C.: BrainWeb: Online interface to a 3D MRI simulated brain database. In: NeuroImage, vol. 5, Academic Press, London (1997)

A Static SMC Sampler on Shapes for the Automated Segmentation of Aortic Calcifications

Kersten Petersen[1], Mads Nielsen[1,2], and Sami S. Brandt[2]

[1] Department of Computer Science, University of Copenhagen, Denmark
[2] Synarc Imaging Technologies, Denmark

Abstract. In this paper, we propose a sampling-based shape segmentation method that builds upon a global shape and a local appearance model. It is suited for challenging problems where there is high uncertainty about the correct solution due to a low signal-to-noise ratio, clutter, occlusions or an erroneous model. Our method suits for segmentation tasks where the number of objects is not known a priori, or where the object of interest is invisible and can only be inferred from other objects in the image. The method was inspired by shape particle filtering from de Bruijne and Nielsen, but shows substantial improvements to it. The principal contributions of this paper are as follows: (i) We introduce statistically motivated importance weights that lead to better performance and facilitate the application to new problems. (ii) We adapt the static sequential Monte Carlo (SMC) algorithm to the problem of image segmentation, where the algorithm proves to sample efficiently from high-dimensional static spaces. (iii) We evaluate the static SMC sampler on shapes on a medical problem of high relevance: the automated quantification of aortic calcifications on X-ray radiographs for the prognosis and diagnosis of cardiovascular disease and mortality. Our results suggest that the static SMC sampler on shapes is more generic, robust, and accurate than shape particle filtering, while being computationally equally costly.

1 Introduction

Shape segmentation is a fundamental problem in many research fields including computer vision, medical image analysis, and geophysics. When the image is afflicted by clutter, occlusions or a low signal-to-noise ratio, object boundaries are hardly visible, and different segmentation results may appear equally likely. Examples range from the segmentation of the prostate in magnetic resonance (MR) images [12] to inferring the shape of underground bodies of salt from seismographs [12]. The problem is even more challenging if the object of interest is invisible and can only indirectly be inferred from other objects in the image. For instance, a country road might only be discernible in the context of surrounding objects (*e.g.* trees) or areas (*e.g.* corn fields or lakes). Another example is the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 666–679, 2010.

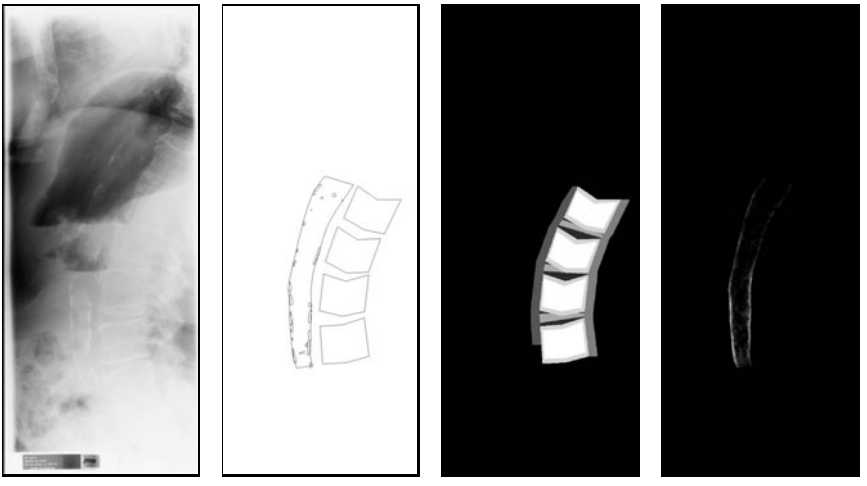


Fig. 1. (a) A lateral X-ray radiograph of the lumbar vertebrae and the aorta. (b) The manual annotations from a trained radiologist for the aorta, aortic calcifications and the vertebrae L1 to L4. (c) A possible shape template for the vertebrae. In the spirit of [5], it encompasses five anatomical regions of interest and a background class. (d) The shape template for the aorta. It is derived by averaging over manually annotated calcifications from several images, and expresses our prior knowledge about the aorta regions that are likely to contain calcifications [3].

case of invisible objects such as a window sill or a water surface which might only be identified from visible small-scale objects, such as rain drops or reflections.

Most existing methods will not succeed in segmenting the images listed above. Global shape models, for instance, are often too constrained to adapt to the desired object. Moreover, they have to be initialized close to the final solution [20] or require automatic object recognition [2]. A global appearance model without a spatial prior does not perform sufficiently well either. It struggles if the object of interest shows many appearance changes or has pixel-wise feature resemblance to other objects in the image. Several authors have instead developed algorithms that incorporate both shape and appearance information. Examples are hierarchical approaches that combine global shape and appearance models, using snakes [2] or Markov processes [16] for local deformations. However, these optimization-based methods have the drawback that they do not naturally cope with multi-modal distributions. Depending on the method, they usually return a local or the global mode, which both can be suboptimal, when a lot of image noise is present [12].

On the contrary, a sampling-based method, known as shape particle filtering [3,4,5], achieves competitive results on many complex segmentation tasks. It uses a global shape model to enforce spatial consistency and a statistical pixel classifier for modeling local and non-linear appearance variations. As the name implies, it uses particle filtering for sampling, which prevents it from getting stuck in local maxima and eliminates the need for manual initialization. The

method is guaranteed to converge if sufficiently many samples are drawn, and it can be applied to problems where the number of objects is unknown a priori [6].

The main idea of shape particle filtering can be summarized as follows. In the training phase, two models are created: (i) A global shape model, usually a point distribution model (PDM) [7], is built from training shapes. (ii) A pixel classifier is trained using local image descriptors with corresponding labels to distinguish the appearance of prominent structures in the image. Furthermore, a function is defined to construct a probabilistic template on the basis of a given shape from the shape model. It is important that the shape template encodes prior knowledge about the spatial distribution of objects in the image that are suitable for segmenting the object of interest. Figure 1 depicts examples of shape templates for segmenting the lumbar vertebrae L1 to L4 and the aorta on a standard radiograph. During testing samples from the global shape model are drawn and a probabilistic shape template for each of them is constructed. To obtain the sample weights for the particle filtering scheme, for each class label the pixel-wise probabilities from the shape template are compared with the respective probabilities from the pixel classifier.

One problem, however, is that this probability comparison appears in various disguises throughout the literature of shape particle filtering. Comparing [3] and [5] for instance, the importance weights differ considerably, since they lack a clear statistical interpretation. Thus, it might be tricky to adapt shape particle filtering to a new segmentation task. We also observe that the underlying sampling algorithm, particle filtering, does not exploit the static nature of the images, but is tailored to dynamic processes. For this reason it requires many samples to approximate the complicated target distribution accurately. Third, the MAP point estimate is arguably not the best choice for summarizing the target distribution. Particularly in segmentation tasks, where we condition shape samples on each other, a Bayesian approach, in which we propagate all the samples, seems more reasonable.

In this paper, we propose a segmentation algorithm that is based on shape particle filtering, but addresses the raised points of concern. First, we propose a statistically-driven formulation of the target distribution, which can be used in stochastic sampling to perform shape segmentation. Second, we substitute particle filtering by a state-of-the-art method for efficiently sampling in high-dimensional spaces. It is called sequential Monte Carlo (SMC) sampler [15] and is proven to produce consistent estimates of the target distribution. It usually performs better than standard MCMC methods which can easily get trapped in local modes and are hard to analyze [8]. We show, how to adapt the SMC sampler to the task of shape segmentation and introduce two adaptation schemes to improve the accuracy and robustness of the algorithm. More specifically, we vary the number of samples to gain computational efficiency, and we adapt the forward Markov kernel of the SMC sampler over time. Finally, we describe the target distribution with all available samples instead of just one.

We evaluate the static SMC sampler on shapes on an important medical problem. The goal is to automatically segment calcifications in the abdominal aorta on standard radiographs, which are important for the prognosis and diagnosis of cardiovascular disease (CVD) and mortality [21]. Although CT images are more valuable to identify and quantify atherosclerosis, standard X-ray radiographs are ubiquitous, inexpensive and fast. They can be of great help for treatment planning, study of drug effects and diagnosis. The main problem of X-ray radiographs, however, is the poor image quality, as illustrated in Figure 1. Calcifications can only be located inside the aorta, but the aorta itself is invisible and can only be estimated from the distribution of the calcifications and from the orientation and shape of the spine. We demonstrate the performance of the sampling algorithms for vertebrae and aorta segmentation on radiographs, both because it is a valuable test to check the method's accuracy and robustness, but also because it is a crucial step for solving a medical problem of interest.

The paper is structured as follows: In Section 2, we formulate the problem of shape segmentation. Section 3 introduces SMC samplers. In Section 4, we discuss the necessary steps to adapt the SMC sampler for shape segmentation. Section 5 shows experimental results and in Section 6 and 7 we conclude with a discussion of the proposed method.

2 Problem Definition

Our objective is to segment a shape governed by parameters $\theta \in \Theta$ in a given image. We approach this problem by using probabilistic shape templates to moderate the output of a pixel classifier. Specifically, we define K class labels representing regions of interest in the image and construct the shape template by specifying the probability for each pixel to belong to class $C_k \in \mathcal{Y}$, where $\mathcal{Y}$ denotes the space of the class labels and $k = 1, \ldots, K$. Each shape may consist of a different number of pixels. To compare shapes statistically, we sample N points from each shape template.

Let the input data for the pixel classifier be given by $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$, where $\mathbf{x}_n$ is an element of the feature space $\mathcal{X}$. Moreover let $\mathbf{u} = (u_1, \ldots, u_N)^{\mathrm{T}} \in \mathcal{Y}^N$ denote a latent variable vector of associated target values [13,14]. Then the marginal posterior distribution of the parameters is given by

$$\begin{aligned} p(\theta|\mathcal{D}) &= \sum_{\mathbf{u}\in\mathcal{Y}^N} p(\mathbf{u}, \theta|\mathcal{D}) \\ &\propto \sum_{\mathbf{u}\in\mathcal{Y}^N} p(\mathcal{D}|\mathbf{u}, \theta) p(\mathbf{u}, \theta) \end{aligned} \tag{1}$$

where we use the short-hand notation $\sum_{\mathbf{u}\in\mathcal{Y}^N} \equiv \sum_{u_1\in\mathcal{Y}} \sum_{u_2\in\mathcal{Y}} \cdots \sum_{u_N\in\mathcal{Y}}$.

Assuming that the input data $\mathcal{D}$ is iid, we can express the likelihood function $p(\mathcal{D}|\mathbf{u}, \theta)$ as the product of the outputs from a pixel classifier $h : \mathcal{X} \to \mathcal{Y}$:

$$p(\mathcal{D}|\mathbf{u},\theta) = \prod_{n=1}^{N} p(\mathbf{x}_n|u_n,\theta) = \prod_{n=1}^{N} h(\mathbf{x}_n(\theta)) \ . \tag{2}$$

The prior $p(\mathbf{u},\theta)$ can be written as

$$p(\mathbf{u},\theta) = p(\mathbf{u}|\theta)p(\theta) = p(\theta)\prod_{n=1}^{N} p(u_n|\theta) \ , \tag{3}$$

where we assume conditional independence of $p(\mathbf{u}|\theta)$ and regard the conditional probability $p(u_n|\theta)$ as the probability map of the shape template.
Substituting equations (2) and (3) in (1), we get:

$$\begin{aligned} p(\theta|\mathcal{D}) &\propto p(\theta) \sum_{\mathbf{u}\in\mathcal{Y}^N} \prod_{n=1}^{N} h(\mathbf{x}_n(\theta))p(u_n|\theta) \\ &= p(\theta)\left(\sum_{u_1\in\mathcal{Y}} h(\mathbf{x}_1(\theta))p(u_1|\theta)\right)\left(\sum_{\mathbf{u}_{\setminus u_1}\in\mathcal{Y}^{N-1}} \prod_{n=2}^{N} h(\mathbf{x}_n(\theta))p(u_n|\theta)\right) \\ &= p(\theta)\prod_{n=1}^{N}\sum_{u_n\in\mathcal{Y}} h(\mathbf{x}_n(\theta))p(u_n|\theta) \\ &\equiv \pi(\theta|\mathcal{D}) \ , \end{aligned} \tag{4}$$

where we use the short-hand notation $\mathbf{u}_{\setminus u_1}$ to denote latent variable vector $\mathbf{u}$ without the first element u_1.

As the normalization factor $p(\mathcal{D})$ is unknown, we can express the conditional distribution $p(\theta|\mathcal{D})$ only up to a constant Z, or

$$p(\theta|\mathcal{D}) = \frac{\pi(\theta|\mathcal{D})}{Z}. \tag{5}$$

Analytical integration of $\pi(\theta|\mathcal{D})$ in a high-dimensional space Θ is generally intractable, but we can numerically integrate by Monte Carlo methods such as sequential Monte Carlo.

3 Static SMC Sampler Approach

The static sequential Monte Carlo sampler (SMC sampler) [15] generalizes SMC methods [11] such as particle filtering. It enables efficient sampling from a sequence of distributions $\{\pi_t\}_{t=1}^{T}$ that are defined on a common measurable space

$(\Omega, \mathcal{F})$. Similar to simulated annealing, the idea is to bridge a tractable proposal distribution π_1 to the target distribution $\pi_T = \pi$ through a sequence of artificial distributions $\{\pi_t\}_{t=1}^{T-1}$. In each iteration the samples are drawn using a standard SMC method.

The static SMC sampler provides asymptotically consistent samples and causes significantly less degeneracy of the weights than the standard SMC methods. Assuming that two subsequent distributions π_{t-1} and π_t are sufficiently close to each other, this scheme propagates samples $\{\theta^{(l)}\}_{l=1}^{L}$ to regions of high density in a sound way. The idea is to first sample from smoothly evolving artificial distributions π_t instead of the complicated target distribution π_T.

In the formulation of an SMC sampler one technical trick is needed to ensure a feasible computation of the importance weights. In an iterative importance sampling scheme the importance weights can only be computed exactly for $t = 1$. In all subsequent iterations an auxiliary variable technique is needed to compensate for the discrepancy between the proposal and the target distribution in the weight computation. An artificial backward Markov kernel $J_{t-1}(\theta_t, \theta_{t-1})$ was introduced in [15], which can be regarded as the counterpart to the forward Markov kernel $K_t(\theta_{t-1}, \theta_t)$ used in standard SMC methods. The importance weight w_t of sample θ_t in iteration t of the SMC sampler can then be defined by

$$\frac{w_t}{w_{t-1}} \propto \frac{\pi_t(\theta_t)}{\pi_{t-1}(\theta_{t-1})} \frac{J_{t-1}(\theta_t, \theta_{t-1})}{K_t(\theta_{t-1}, \theta_t)}. \tag{6}$$

The challenge of applying an SMC sampler to a given problem lies in modeling the intermediate distributions π_t and the two Markov kernels K_t and J_{t-1}. They have to be chosen in a way that the samples mix well and the weight variance is small. Let us discuss our choices for the static SMC sampler on shapes.

4 Static SMC Sampler on Shapes

4.1 Artificial Distributions

Inspired by [17], we define the artificial target distributions $\{\pi_t\}_{t=1}^{T}$ by the geometric path between the proposal distribution $\pi_1 = p(\theta)$ and the target distribution $\pi_T = \pi(\theta|\mathcal{D})$ in (4) such that

$$\pi_t(\theta) = \pi(\theta|\mathcal{D})^{\beta_t} p(\theta)^{1-\beta_t} \tag{7}$$

Algorithm 1. Static SMC Sampler on Shapes (Training)

Require: Training features and labels $\mathcal{D}_{\text{train}}$; training shapes θ_{train}.
Ensure: Trained models $p(\theta)$ and h.

Train a global shape model $p(\theta)$ based on θ_{train}.
Train a pixel classifier h based on $\mathcal{D}_{\text{train}}$.

Algorithm 2. Static SMC Sampler on Shapes (Testing)

Require: Testing features $\mathcal{D}_{\text{test}}$ for given image; a probabilistic shape template $p(u|\theta)$; N evaluation positions for the mean shape of $p(\theta)$; schedules $\{L_t\}_{t=1}^{T}$ and $\{M_t\}_{t=1}^{T}$; threshold α for the desired sample survival rate; output of algorithm 1
Ensure: Monte Carlo estimate of $p(\theta|\mathcal{D}_{\text{test}})$.

1. Initialization:

Set $\beta_0 = 0$ and $L_{t+1} = L_t$.
for $l = 1, \ldots, L_{\text{dense}}$ **do**
  Draw dense samples $\theta_{\text{dense}}^{(l)} \sim p(\theta)$.
end for

2. Sampling and Resampling:

for $t = 1$ to T **do**
  for l that index repeated occurrences in $\{\theta_t'^{(l)}\}_{l=1}^{L_t}$ **do**
    if $t = 1$ **then**
      Draw samples $\theta_t^{(l)} \sim p(\theta)$.
    else
      Train $p_{\text{local}}(\theta)$ using the M_t nearest samples from $\theta_{\text{dense}}^{(l)}$ to $\theta_{t-1}'^{(l)}$.
      Draw samples $\theta_t^{(l)} \sim p_{\text{local}}(\theta)$.
    end if
    Warp the N evaluation positions w.r.t. the mean shape to sample $\theta^{(l)}$.
    Obtain $\{\mathbf{x}_n\}_{n=1}^{N} \subset \mathcal{D}_{\text{test}}$ and $\mathbf{u} = (u_1, \ldots, u_N)^{\mathrm{T}}$ using these warped positions.
    Evaluate $\pi_t(\theta^{(l)})$ using (4) and (7).
  end for
  Optimize β_t such that the sample survival rate equals α.
  Evaluate sample weights $\{w_t^{(l)}\}_{l=1}^{L_t}$ using (9) and normalize them.
  Resample $\{\theta_t^{(l)}, w_t^{(l)}\}_{l=1}^{L_t}$ to obtain $\{\theta_t'^{(l)}, 1/L_{t+1}\}_{l=1}^{L_{t+1}}$.
end for

with annealing parameter $0 \leq \beta_1 < \ldots < \beta_T = 1$. To ensure that $\pi_t \approx \pi_{t-1}$, the number of annealing iterations T should be set to the maximal value that is computationally still feasible. As a starting point for tuning the values of $\{\beta_t\}_{t=1}^{T}$ one may choose a cooling schedule with a slow density increase in the beginning and a steeper raise towards the end. This facilitates samples to gradually move towards high density regions. Popular choices are logarithmic, quadratic or piecewise linear cooling schedules, as proposed in [15] and [17]. We prefer to construct the schedule by optimizing β_t in each iteration, such that the effective sample size equals a user-specified value [9].

4.2 Forward Markov Kernel

After the resampling phase the static SMC sampler applies the forward Markov kernel K_t to draw new samples θ_t given the samples θ_{t-1} from the previous iteration. As in [5], we only compute new weights for repeated resampled samples, whereas those that occur for the first time remain untouched. In other words, well fitting samples are rather propagated than perturbed through K_t.

While our scheme uses the same kernel as shape particle filtering for $t = 1$, it differs for subsequent time steps. In shape particle filtering, $\{K_t\}_{t\geq 2}$ is identical to K_1, except that the mean differs and the shape variance is scaled by a constant factor. A fixed value for the scaling factor, however, is problematic, since it should lead to a good mixture of samples in initial iterations, while not heavily distorting samples in later iterations. Another problem is that samples from this diffusion kernel may be far away from the shape manifold, when the mean for $\{K_t\}_{t\geq 2}$ is distant from the mean for K_1.

Our idea is to instead generate a large number of samples $\{\theta\}_{l=1}^{L_{\text{dense}}}$ from K_1 before we iterate over the artificial distributions. As a guideline L_{dense} should generally be set to a much larger value than the number of samples for which we evaluate the importance weights. We use these dense samples to generate the forward Markov kernel for iterations $t > 1$. More specifically, we propose to build a local shape model from the M_t nearest shapes of a repeated sample θ_{t-1}, which is then used to draw the new sample θ_t. This encourages sample distortions in directions that are plausible under the global shape model. Note that parameter M_t is also iteration dependent. It should be decreased over time to achieve a good mixture of samples while ensuring small weight variance.

4.3 Backward Markov Kernel

We define the backward Markov kernel J_{t-1} as an approximation to the optimal backward Markov kernel that minimizes the variance of the unnormalized importance weights. Let us suppose that two subsequent artificial distributions are similar to each other and K_t is a valid MCMC kernel of the invariant distribution π_t. Then according to [15], a good choice for the backward Markov kernel is

$$J_{t-1}(\theta_t, \theta_{t-1}) = \frac{\pi_t(\theta_{t-1})K_t(\theta_{t-1}, \theta_t)}{\pi_t(\theta_t)} \,. \tag{8}$$

which simplifies the importance weights to

$$\frac{w_t}{w_{t-1}} \propto \frac{\pi_t(\theta_{t-1})}{\pi_{t-1}(\theta_{t-1})} \,. \tag{9}$$

4.4 Resampling

The SMC sampler incorporates a resampling stage that accounts for the potentially increasing discrepancy between π_{t-1} and π_t. In contrast to shape particle filtering, however, it is not necessarily applied after each iteration. The effective number of samples (ESS) has to fall below a user-specified threshold to initiate a resampling phase. We propose to use systematic resampling, but any of the other commonly applied techniques is possible as well [10].

The SMC sampler on shapes also resamples for adapting the number of samples L_t used in each iteration t. In initial iterations we generate a relatively large number of samples in order to explore the shape space and approximately hit

potential modes of the target distribution. After several iterations though, we lower the number of samples to be computationally more efficient. We assume that a subset of samples approximates the target distribution sufficiently well when these samples reside in regions of high density.

5 Evaluation

We test the proposed approach on 57 randomly picked lateral spine radiographs taken from a combined osteoporosis–atherosclerosis screening program. The training of the models for the aorta and vertebrae segmentation is performed on 78 separate images. The full dataset contains both healthy and fractured vertebrae as well as aortas with no till many visible calcifications. The original radiographs have been scanned at a resolution of 45 μm per pixel. A medical expert outlined the aorta and placed six landmark points on each of the vertebrae L1 to L4.

5.1 Vertebrae Segmentation

In the vertebrae setting, we generate an anatomical template that consists of six regions of interest, the five vertebral classes defined in [4] and a background class. The shape–pose model explains 90% of the vertebrae shape variation. A random forest classifier [1] is trained on 100,000 sampled points from the five

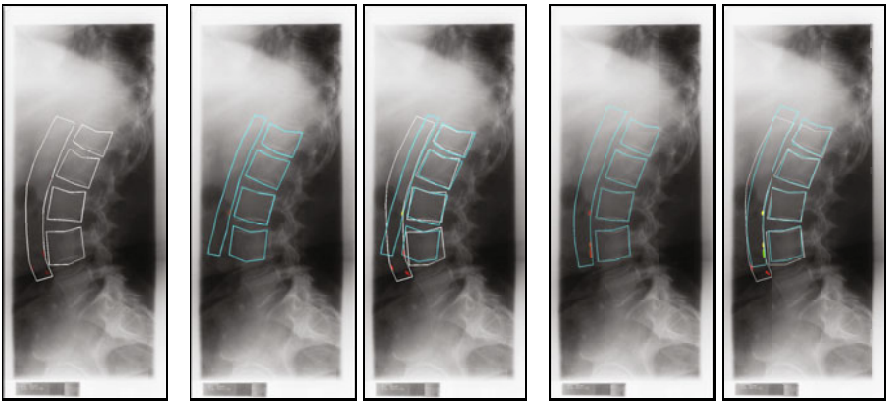


Fig. 2. This image sequence illustrates a typical result obtained from both sampling algorithms. (a) The manual annotations from a medical expert. (b) Result of shape particle filtering. (c) The overlay of the manual annotations and the result of shape particle filtering. (d) Result of the static SMC sampler on shapes. (e) The overlay of the manual annotations and the result from the SMC sampler on shapes using shape-dependent features. The calcifications are colored as follows: Green denotes true positives, red false negatives and yellow false positives.

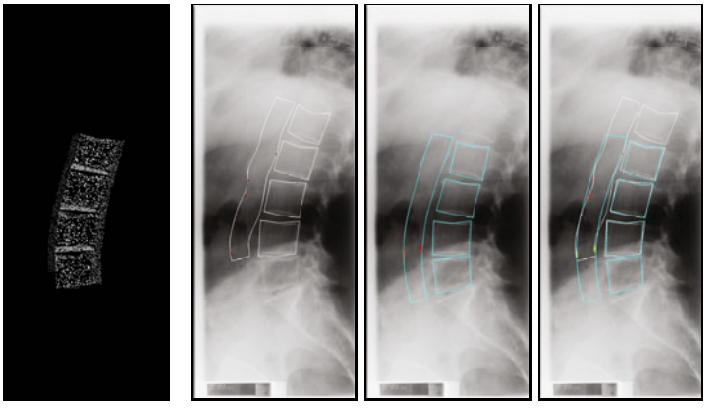


Fig. 3. (a) The evaluation points for a vertebrae sample. We draw the same number of points from each template class, as we assume that they equally contribute in finding the correct vertebrae. Thin plate splines are used to warp these points to the individual samples. (b)-(d) A case with a vertebra shift for the static SMC sampler on shapes. It is a common problem for the vertebra segmentation and downgrades the aorta and vertebrae results for both segmentation methods. The individual panels have the same explanation as in the Figure 2.

shape-related foreground classes using Gaussian derivative features up to third order on scales 0.17, 0.56 and 1.7 mm.

The vertebrae settings for the shape particle filtering algorithm have been reported in [4]. The SMC sampling scheme is defined as follows: We use $T = 50$ artificial distributions and set the desired sample survival rate to $\alpha = 0.3$. These settings cause β_T to be approximately 0.3 instead of 1, which is intended, as it saves computational time and the obtained vertebrae samples are more diverse. The final samples are more likely to stem from different local modes instead of only one large mode, which might be suboptimal under the influence of a lot of image noise. We decrease L_t logarithmically in 5 steps from 3000 to 350 followed by 45 iterations with 200 samples. Parameter M_t falls logarithmically from 400 to 5 and L_{dense} is set to 100,000. We evaluate each vertebrae sample at 5000 positions per vertebrae class (see Figure 3).

In our experiments, the median of the mean distance between the manually annotated vertebrae and the vertebrae MAP estimate obtained from shape particle filtering is 1.9 mm, whereas the static SMC sampler on shapes yields 1.59 mm. State-of-the-art vertebrae segmentation methods [19,18] achieve a mean distance of 0.93 for healthy and 2.27 mm for fractured vertebrae contours on a different dataset. Figure 2 shows a typical result of our method in comparison to shape particle filtering. A common problem for this data set are shifts in the vertebral levels, as illustrated in Figure 3. However, strategies to alleviate this problem are outside the scope of this paper.

5.2 Aorta Segmentation

The anatomical template for the aorta is adopted from [3] and it defines two classes: calcified and non-calcified. We sample 100,000 points from the aorta template to train a random forest classifier. The setup for the pose–shape model is analogous to the vertebrae case. To infer an aorta from vertebrae shapes we train a conditional model of the aorta given the vertebrae.

In shape particle filtering, we compute the MAP estimate of the vertebrae samples and regress a mean aorta shape using the trained conditional model. The aorta samples for the first iteration are drawn from the aorta shape model around this aorta mean shape. For the static SMC scheme, we propagate the samples in a Bayesian way. Instead of calculating a single point estimate the whole set of samples is passed on to the aorta stage and taken as the input of the conditional model. We obtain a set of mean aorta samples with associated covariance matrices, which we use to construct our proposal distribution, a mixture of Gaussians. We set $T = 20$, $\alpha = 0.3$ and $L_{\text{dense}} = 10,000$. The number of samples L_t decreases logarithmically from 750 to 100 in 5 steps and remains at 100 for the residual iterations. M_t falls logarithmically from 100 to 5. We evaluate the aorta samples at all warped points from the mean of the global aorta shape model.

The results for the aorta segmentation show a similar trend as for the vertebrae stage. The median of the mean distance between the manually annotated aorta and the aorta from shape particle filtering is 4.77 mm, whereas the static SMC sampler on shapes obtains 3.95 mm. The corresponding aorta area overlaps are 0.51 and 0.62 respectively. However, in contrast to the aorta weights of shape particle filtering, our weight formulation can also benefit from more descriptive features that depend on the aorta and the calcification shape. If we extend our appearance features by shape-related calcification features (*e.g.* area, length of principal components, eccentricity, or perimeter) and features that measure the alignment of the aorta and the calcifications, the aorta area overlap increases to 0.71 and the mean distance decreases to 2.83 mm.

6 Discussion

On the basis of this study, our conclusion is that the static SMC sampler on shapes yields better results than shape particle filtering. Let us discuss the reasons for its higher performance.

First, the static SMC sampler on shapes uses sound importance weights in contrast to shape particle filtering, where the weights lack a clear statistical interpretation. The weights in [3] or [4] for instance do not directly account for the shape prior, which explains several unlikely aorta shape segmentations. Furthermore, the convolved formulation of the aorta weights does not support shape-dependent features, whereas our method benefits considerably, as we have shown in Section 5.2. A third problem of shape particle filtering is that the drawn shape samples are not comparable, as they are not evaluated at the same

relative positions. Thus, the sample weights are corrupted, when the size or the ratio of the different classes in the shape template varies.

Second, the underlying static SMC sampler seems to be more efficient than particle filtering in exploring high-dimensional static spaces with potentially isolated modes. Image segmentation seems to exhibit a complicated target distribution, such that we benefit, when we gradually guide samples from the proposal distribution to regions of high density in the target distribution. At least our method performs better than shape particle filtering for the vertebrae segmentation, although both weight formulations resemble each other for the vertebrae case. The difference can also be attributed to the changed diffusion kernel. A local shape model of decreasing size seems to be more appropriate for obtaining samples that cover most of the solution space in the initial iterations, but explore narrow peaks in later iterations. Compared to shape particle filtering, our diffusion kernel should lead to fewer overshoots in complicated target distributions, while achieving a better mixture of samples. The iterative adaptation of the sample size on the other side should not affect the performance of the segmentation algorithm. It helps reducing the computational cost of our method.

A third reason is that we summarize the target distribution by all the final samples, not just the MAP point estimate as in shape particle filtering. By following this Bayesian idea, we benefit especially multi-stage segmentation models. For instance, shape particle filtering unavoidably misplaces the aorta sample in Figure 2, as it is conditioned on an inexact vertebrae sample. However, the static SMC sampler on shapes can find a more plausible aorta based on a suboptimal vertebra sample.

One may think that the SMC sampler on shapes is difficult to calibrate, as it requires two trained models and several parameters to be set. However, we suspect that the settings of our examples should be a reasonable starting point for many other segmentation tasks. The number of artificial distribution T and the number of samples $\{L_t\}_{t=1}^{T}$ should be set to the maximal values that are computationally still feasible. Both the static SMC sampler on shapes and shape particle filtering required for instance roughly 15 minutes per aorta or vertebrae segmentation on a 2.4 GHz Intel core under MATLAB. The number of dense samples L_{dense} should generally exceed $\{L_t\}_{t=1}^{T}$ and the schedule for $\{M_t\}_{t=1}^{T}$ may roughly start at $\sqrt{L_{\text{dense}}}$ and decrease to a small number like 5. We achieved satisfactory results by setting $\alpha = 0.3$, which leads to a resampling of several unlikely shapes without risking a fast degeneracy of the estimates. We assume that the SMC sampler on shapes is not harder to adjust than a standard Markov Chain Monte Carlo algorithm, where one typically needs to estimate the autocorrelation and the number of 'burn in' samples.

In this paper, we experimented with a linear point distribution model (PDM) [7], but more sophisticated shape models is likely to generate more plausible samples from the shape manifold. We plan to investigate the influence of shape models on additional segmentation tasks in the future. Similarly, the design of the shape template and the pixel classifier can play an important role for the performance of our method. The shape template should define as many regions

of consistent appearance as possible without imposing too many constraints on the hypothesis space. Also, careful design of the pixel classifier and the features may have substantial impact on the performance of the segmentation algorithm.

Further improvements can be expected from the design of the forward Markov kernel. In future work, we plan to investigate the potential of various combinations of MCMC kernels. Similar to the rib segmentation on chest radiographs [6], it may be worthwhile analyzing MCMC kernels that incorporate prior knowledge about the potential modes of the target distribution based on the current sample and its weight. For instance, in the segmentation of the vertebrae, we could combine the current MCMC kernel an MCMC kernel that suggests samples at the potential side modes, which are presumably located one vertebra up or down.

7 Conclusion

In summary, the static SMC sampler inherits many advantages of shape particle filtering and in addition improves its accuracy and robustness. We enhanced the method by three contributions: First, we introduced a theoretically sound formulation of the importance weights. Second, we suggested to replace particle filtering by an SMC sampler. And third, instead of adhering to the MAP estimate, we proposed to propagate the whole target distribution represented by its Monte Carlo estimates. We adapt the forward Markov kernel and the number of samples over time to obtain better convergence to the modes and save computational time. We demonstrated our algorithm on the segmentation of the aorta and the vertebrae, and saw that it can be an accurate and robust tool to proceed in this technically demanding but clinically valuable task.

References

1. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
2. Brejl, M., Sonka, M.: Object localization and border detection criteria design in edge-based image segmentation: Automated learning from examples. IEEE Trans. Med. Imaging 19(10), 973–985 (2000)
3. de Bruijne, M.: Shape particle guided tissue classification. In: Golland, P., Rueckert, D. (eds.) MMBIA (2006)
4. de Bruijne, M., Nielsen, M.: Image segmentation by shape particle filtering. In: ICPR 2004, vol. 3, pp. 722–725 (2004)
5. de Bruijne, M., Nielsen, M.: Shape particle filtering for image segmentation. In: Barillot, C., Haynor, D.R., Hellier, P. (eds.) MICCAI 2004. LNCS, vol. 3216, pp. 168–175. Springer, Heidelberg (2004)
6. de Bruijne, M., Nielsen, M.: Multi-object segmentation using shape particles. In: IPMI, pp. 762–773 (2005)
7. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models—their training and application. Comput. Vis. Image Underst. 61(1), 38–59 (1995)
8. Del Moral, P., Doucet, A., Jasra, A.: Sequential monte carlo samplers. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68(3), 411–436 (2006)

9. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 126–133 (August 2002)
10. Douc, R., Cappe, O.: Comparison of resampling schemes for particle filtering. In: Pan, Y., Chen, D.-x., Guo, M., Cao, J., Dongarra, J. (eds.) ISPA 2005. LNCS, vol. 3758, pp. 64–69. Springer, Heidelberg (2005)
11. Doucet, A., de Freitas, N., Gordon, N.: Sequential Monte Carlo methods in practice. Springer, New York (2001)
12. Fan, A.C.: Curve Sampling and Geometric Conditional Simulation. PhD thesis, Massachussetts Institute of Technology (February 2008)
13. Hansson, M., Brandt, S., Gudmundsson, P.: Bayesian probability maps for evaluation of cardiac ultrasound data. In: PMMIA (2009)
14. Hansson, M., Brandt, S., Gudmundsson, P., Lindgren, F.: Evaluation of cardiac ultrasound data by bayesian probability maps. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Kuno, Y., Wang, J., Pajarola, R., Lindstrom, P., Hinkenjann, A., Encarnacão, M.L., Silva, C.T., Coming, D. (eds.) ISVC 2009. LNCS, vol. 5876, pp. 1073–1084. Springer, Heidelberg (2009)
15. Johansen, A.M., Del Moral, P., Doucet, A.: Sequential monte carlo samplers for rare events. Technical report, University of Cambridge, Department of Engineering, Trumpington (2005)
16. Kervrann, C., Heitz, F.: A hierarchical markov modeling approach for the segmentation and tracking of deformable shapes. Graphical Models and Image Processing 60(3), 173–195 (1998)
17. Neal, R.M.: Annealed importance sampling. Statistics and Computing 11(2), 125–139 (2001)
18. Roberts, M.G., Cootes, T.F., Adams, J.E.: Robust active appearance models with iteratively rescaled kernels. In: Proc. BMVC, vol. 1, pp. 302–311 (2007)
19. Roberts, M.G., Cootes, T.F., Pacheco, E., Oh, T., Adams, J.E.: Segmentation of lumbar vertebrae using part-based graphs and active appearance models. In: Yang, G.-Z., Hawkes, D., Rueckert, D., Noble, A., Taylor, C. (eds.) MICCAI 2009. LNCS, vol. 5762, pp. 1017–1024. Springer, Heidelberg (2009)
20. Smyth, P.P., Taylor, C.J., Adams, J.E.: Automatic measurement of vertebral shape using active shape models. In: Image and Vision Computing, pp. 705–714. BMVA Press (1996)
21. Wilson, P., Kauppila, L., O'Donnell, C., Kiel, D., Hannan, M., Polak, J., Cupples, L.: Abdominal aortic calcific deposits are an important predictor of vascular morbidity and mortality. Circulation (103), 1529–1534 (2001)

Fast Dynamic Texture Detection

V. Javier Traver[1,2], Majid Mirmehdi[3], Xianghua Xie[4], and Raúl Montoliu[2,5]

[1]DLSI, [2]iNIT, [5]DICC, Univ. Jaume I, Castellón, Spain
{vtraver,montoliu}@uji.es
[3]Dept. Comp. Science, Univ. of Bristol, Bristol, UK
majid@cs.bris.ac.uk
[4]Dept. Comp. Science, Univ. of Wales Swansea, Swansea, UK
x.xie@swansea.ac.uk

Abstract. Dynamic textures can be considered to be spatio-temporally varying visual patterns in image sequences with certain temporal regularity. We propose a novel and efficient approach to explore the violation of the brightness constancy assumption, as an indication of presence of dynamic texture, using simple optical flow techniques. We assume that dynamic texture regions are those that have poor spatio-temporal optical flow coherence. Further, we propose a second approach that uses robust global parametric motion estimators that effectively and efficiently detect motion outliers, and which we exploit as powerful cues to localize dynamic textures. Experimental and comparative studies on a range of synthetic and real-world dynamic texture sequences show the feasibility of the proposed approaches, with results which are competitive to or better than recent state-of-art approaches and significantly faster.

Keywords: Dynamic texture; optical flow; brightness constancy assumption; global parametric motion.

1 Introduction

Dynamic textures (DTs) can be considered to be spatio-temporally varying visual patterns in image sequences, such as fire, waterfalls, crops in wind, shaking leaves, and many other instances of moving, structured or unstructured patterns. There has been a multitude of research works in DT segmentation, detection, recognition and synthesis, for example [1,2,3]. DT analysis has been found useful in a number of real-world applications, such as particle measurements [4], smoke detection in surveillance scenarios [5], and facial expression recognition [6].

DTs not only exhibit complex appearance but also commonly lack distinctive local features, for example due to transparency and challenging spatio-temporal variability. Linear models, particularly auto-regressive (AR) techniques, have been widely used in modeling DTs [7,8,9]. Most of the proposed AR models are first-order, which may prevent oscillations and higher-order temporal dependencies be correctly captured, but higher-order AR models such as [8] can improve the accuracy of synthesized sequences by capturing complex patterns over multiple frames. A DT constancy constraint was introduced in [10] as an analogy to the brightness constancy assumption for DTs. Very recently, layered models [11] have been proposed to deal with multiple different DTs

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 680–693, 2010.

in a single sequence. However, in general, AR models have been found mainly successful on synthetic DT sequences, in which the spatial extent of the DTs remains largely constant. For some DTs, such as smoke and fire, it is difficult for AR techniques to generalize. Furthermore, these methods rely on costly learning procedures and complicated mathematical and conceptual schemes [12].

Fractals [13], local binary patterns (LBP) [6], and spatio-temporal multiresolution histograms [14] have also been applied to DT analysis. These are illustrative examples of attempts to properly model the spatio-temporal nature of DTs. Campisi et al. [13] extend to the temporal dimension the self-similarity model which is known to be present in natural images. Zhao and Pietikäinen [6] use LBP in three planes (XY, YT, XT), allowing them to consider the spatio-temporal domain. Lu et al. [14] use histograms of velocity and acceleration at four levels of an image pyramid, applying spatio-temporal Gaussian filters to perform spatial and temporal multiresolution analysis.

Recently, Markov random fields have been used to model DT spatio-temporal dynamics, e.g. [15,16]. For example, in [16], the authors adopted two three-state hidden Markov models to model DT and non-DT moving objects. However, their focus was on one particular type of object, i.e. swaying leaves. Several other authors have also proposed specific models for specific DTs, e.g. steam [17], smoke [5,18], fire [19], or general fluids [20]. For example, steam is considered to blur image details and a supervised method using wavelets and other local features are applied in [17] for DT classification. Similarly, smoke smoothes image edges, a scenario which can be detected by monitoring the abrupt change of some local energy measure [5] just after the appearance of DT. Flickering and turbulence phenomena of smoke are exploited in [18]. Although some of these methods such as [21,18] perform in real-time, they are clearly not applicable to other kinds of DTs. An added restriction is that these techniques often assume the background scene to be known in advance.

Amongst many other motion cues [22,23], optical flow has been particularly useful in DT analysis. For example, in [24,1], optical flow is shown to be effective in discriminating different types of DTs. Regarding the optical flow, it is common to rely on the brightness constancy assumption (BCA) which states that a change of brightness at an image location has motion as its only cause. However, this has been found insufficient in dealing with DTs in real-world sequences [25], and alternative flow models to BCA, such as gradient constancy, color constancy and brightness conservation, were recently explored in [25], where an approach based on level-sets was formulated to segment image regions obeying either the BCA or some of these alternative flow models.

In contrast, a different, simpler route is explored in this paper. Since DT pixels do not follow the BCA, the dynamic texture can be located by detecting those pixels at which the estimated optical flow is not correct. In order to detect these optical flow "failures", two alternative approaches, as instances of a proposed general scheme for DT detection, are investigated (Sect. 2). One method is based on the fact that optical flow in DTs will exhibit changes in a local spatio-temporal neighborhood (Sect. 2.1). The other approach is based on using the motion outliers as detected by a robust global parametric 2D motion estimator (Sect. 2.2). In comparison to the state-of-the-art (Sect. 3), we obtain similarly accurate, if not better, results with methods which are both conceptually simpler and substantially faster (Sect. 4).

Algorithm 1. DT detection for pixel (i, j) at time t

Input: $\mathbf{v}$: past value,
$\mathbf{x}$: new value,
λ: weight of past and new information, and
$\mathcal{DT}$: DT probability at previous time $t-1$
Output: updated $\mathbf{v}$ and
$\mathcal{DT}(i,j)$: updated DT probability at current time t
1: $\varepsilon \Leftarrow$ evidenceOfDT($\mathbf{v}$, $\mathbf{x}$)
2: $\mathbf{v} \Leftarrow$ updateVisualCue($\mathbf{v}$, $\mathbf{x}$, ε)
3: $\mathcal{DT}(i,j) \Leftarrow$ temporalSmooth($\mathcal{DT}(i,j)$, ε, λ)
4: $\mathcal{DT}(i,j) \Leftarrow$ spatialSmooth($\mathcal{DT}$, i, j)

2 Proposed Methods

A general framework to detect dynamic textures in video sequences is proposed. It has as its core procedure the approach shown in Algorithm 1, which encapsulates a simple idea and admits a number of reasonable variants. For each pixel, information is kept on past visual data $\mathbf{v}$, and new visual cues $\mathbf{x}$ is computed at a given time step t. With $\mathbf{v}$ and $\mathbf{x}$, spatio-temporal evidence ε of DT is gained, and used to update the DT likelihood map $\mathcal{DT}$ at the corresponding pixel (i, j). The evidence ε is also used to update $\mathbf{v}$ with the current value $\mathbf{x}$. Then, in order to disregard short-time noisy detections and get more stable DT regions over time, the $\mathcal{DT}$ is temporally filtered. In particular, we use:

$$\mathcal{DT}(i,j) \Leftarrow \lambda \cdot \mathcal{DT}(i,j) + (1-\lambda) \cdot \varepsilon, \tag{1}$$

where the value for $\lambda \in (0, 1)$ can be chosen as a tradeoff between stability and reactivity of the detection. As a last step, the dynamic texture map $\mathcal{DT}$ is spatially smoothed with a $k \times k$ Gaussian kernel (where $k = 25$ was determined empirically). This spatial filtering is intended to remove small regions and provide smoother and more compact DT regions. The map is finally thresholded at 0.5 (which is also established empirically) to get a binary DT mask.

From this general description, a number of specific methods can be instantiated by defining the various elements of the algorithm: the visual information used for $\mathbf{v}$ and $\mathbf{x}$, how the evidence for DT is predicted, and how the updating of $\mathbf{v}$ is done (if at all). Two possible approaches, with their associated merits and shortcomings, are presented in this paper.

2.1 Optical-Flow-Based DT Detection (OFDT)

A characteristic of DT is that its visual appearance changes over time, and hence, detecting temporal changes is a reasonable approach to detect DT. In our first approach, based on optic flow, and referred to as OFDT, the evidenceOfDT($\mathbf{v}$, $\mathbf{x}$) function can be defined as

$$H(\theta_s - \mathcal{S}(\mathbf{v}, \mathbf{x})), \tag{2}$$

where $\mathcal{S}(\cdot, \cdot) \in [0, 1]$ is a similarity measure, θ_s is a similarity threshold, and $H(x)$ is the Heaviside (step) function (i.e. $H(x) = 1$ for $x > 0$, and $H(x) = 0$ for $x < 0$).

Smooth approximations of the step function can also be defined. Therefore, the evidence for DT is high when similarity between past and current visual data is low, which means a change is detected.

One visual cue which has often been used to characterize and recognize DTs is the optical flow, which can also be used to detect DT. Here, the values used for $\mathbf{v}$ and $\mathbf{x}$ are the two components of the flow vector, i.e. $\mathbf{v} = (v_1, v_2) = (v_x, v_y)$, and $\mathbf{x} = (x_1, x_2) = (v'_x, v'_y)$. One of the many possible similarity measures is

$$\mathcal{S}_{\mathsf{OFDT}}(\mathbf{v}, \mathbf{x}) = \frac{1}{2} \sum_{i=1}^{2} \exp\left(-\gamma_i \cdot \delta_i^2\right), \tag{3}$$

where $\delta_i = v_i - x_i$ is the difference in each component, and γ_i weights the squared difference δ_i^2 proportionally to a measure of the local variance of the corresponding flow component. The greater the local variance, the more importance is given to the difference. One way to set γ_i is given below. Other measures besides (3) were explored, in all cases seeking their normalization in $[0, 1]$ to set θ_s more easily.

The function used to update $\mathbf{v}$, $\mathsf{updateVisualCue}(\mathbf{v}, \mathbf{x}, \varepsilon)$, is simply its assignment to the current value $\mathbf{x}$ when $\varepsilon = 1$. Other sensible definitions are possible, such as a weighted sum between past, $\mathbf{v}$, and new data, $\mathbf{x}$.

The key observations behind the OFDT method are: (1) the BCA does not hold on DT; and (2) the flow computed assuming BCA exhibits a weak temporal and spatial coherence in DT locations. Therefore, DT can be detected by detecting optical flow "failures". The weak temporal coherence is captured by the temporal change detection, while the lack of spatial coherence is captured by the local measure of flow variance. Both procedures, temporal change detection and spatial variance, are coupled through the similarity measure $\mathcal{S}$. Let (μ_i, σ_i) denote the mean and standard deviation of the local optical flow component $i \in \{1, 2\}$. Then, γ_i is set as the relative standard deviation, $\gamma_i = \frac{\sigma_i}{|\mu_i|}$. The use of the *relative* standard deviation here reflects the idea that the importance of the variance of the optical flow depends on its magnitude. For example, this allows us to capture subtle DT in regions of small flow that would be undetected otherwise. The values (μ_i, σ_i) are computed on 5×5 windows and, to make these local computations faster, the concept of integral images [26] is used.

The charm of the OFDT approach is that no alternative motion models are needed at all, and no complex procedures, such as those based on level sets, are really required. This clearly contrasts with the conceptual and computational complexity of previous recent approaches, e.g. [25], summarized in Sect. 3. Our approach shows that just conventional optical flow methods relying on the BCA can be used, e.g. the Lucas-Kanade and Horn-Schunck methods.

Other approaches for change detection not based on optical flow can be those that exploit appearance cues. For instance, RGB values can be considered for $\mathbf{v}$ and $\mathbf{x}$, and a color-based similarity measure can be defined for $\mathcal{S}$. Despite the simplicity of such an approach, tests with this appearance-based DT detection yields very good results, provided that the camera does not move; otherwise, changes in appearance may easily happen as a consequence of camera motion and non-uniform scene. The proposed OFDT method is more flexible than this and can deal with egomotion conditions, as demonstrated later in Sect. 4.

2.2 Motion Outliers-Based DT Detection (MODT)

In this section, we present a second approach to DT detection, referred to as MODT and also based on Algorithm 1, which exploits the global motion outliers in the image sequence. Global image motion can be estimated with parametric 2D techniques which, unlike optical flow methods, can deal better with larger image deformations and can be very robust to the presence of a large amount of motion outliers, i.e. image locations not following the motion of the main part of the image. Estimating the motion parameters $\boldsymbol{\mu}$ of a given motion model $\mathbf{f}(\mathbf{p};\boldsymbol{\mu})$ is stated in [27] as minimizing error measure

$$E(\boldsymbol{\mu}) = \sum_{\mathbf{p}} \rho(\mathrm{DFD}_{\boldsymbol{\mu}}(\mathbf{p})), \tag{4}$$

where

$$\mathrm{DFD}_{\boldsymbol{\mu}}(\mathbf{p}) = I(\mathbf{f}(\mathbf{p};\boldsymbol{\mu}), t+1) - I(\mathbf{p}, t) \tag{5}$$

is the displaced frame difference considering the geometric transformation corresponding to the motion model $\mathbf{f}(\mathbf{p};\boldsymbol{\mu})$ which maps location $\mathbf{p}$ to another position for a given motion parameter vector $\boldsymbol{\mu}$, ρ is an M-estimator (such as the Tukey's biweight), and $I(\mathbf{p}, t)$ is the gray-level value of image I at location $\mathbf{p} = (i, j)$ and time t.

To solve the M-estimator problem, iterative reweigthed least squared (IRLS) is used so that the problem is converted into a weighted least-squares problem, $E(\boldsymbol{\mu}) = \sum_{\mathbf{p}} w(\mathbf{p}) \cdot \mathrm{DFD}^2_{\boldsymbol{\mu}}(\mathbf{p})$. The weights $w(\mathbf{p})$ are defined as $w(\mathbf{p}) = \frac{\psi(r(\mathbf{p}))}{r(\mathbf{p})}$, with ψ being the influence function (the derivative of the ρ function), and $r(\mathbf{p}) = \mathrm{DFD}_{\boldsymbol{\mu}}(\mathbf{p})$ is the residual. The minimization of this error is performed using an incremental and multiresolution scheme that deals with larger motions and prevents falling into local minima. At each level of the multiresolution pyramid, the IRLS process is applied.

In our case, we are not interested in the recovery of the motion parameters (which such methods can estimate very accurately) but rather, in their ability to detect motion outliers. We considered the affine motion model, and used four levels in the pyramid. The key idea is that DT pixels can simply be identified with motion outliers. The weights $w(\mathbf{p}) = w(i, j) \in [0, 1]$ represent how well a pixel (i, j) supports the parametric motion model or not. Therefore, the visual cue used for $\mathbf{x}$ in Algorithm 1 is just $w(i, j)$ and the evidence function for DT is simply $1 - \mathbf{x}$. Note, in this instantiation of the proposed framework, the past visual information $\mathbf{v}$ is not used, since only the frame-to-frame motion outliers are considered, so no updating of $\mathbf{v}$ is required either.

2.3 Comparison of OFDT and MODT

OFDT and MODT identify DT locations by detecting "motion failures" either as spatio-temporally irregular optic flow or as outliers of global motion. In comparison to OFDT, MODT is even simpler, since only the weights $w(i, j)$, directly provided as a by-product of the general-purpose global parametric motion estimator, are used. MODT is as fast or faster than OFDT (depending on the particular way the optical flow is computed). Additionally, MODT is generally very effective given that the parametric 2D motion is estimated on a frame-to-frame basis and no analysis is done on the temporal change of outliers, as it is done on the temporal change of the optical flow in OFDT. For instance,

jerky camera motions are dealt with more robustly by MODT, since the frequent flow changes induced by egomotion can be misdetected by OFDT as DT. It is possible to explore, however, how the influence on past information, which is considered in OFDT within $\mathcal{S}$, could be removed by redefining $\mathcal{S}$ so that it considers only the spatial inhomogeneity of the optical flow, not its temporal change. This is important, since the temporal change might be due to a camera moving at non-constant speed rather than to genuine DT's dynamics. Future work will look into this possibility.

Additionally, it can be noticed that, in MODT, no similarity measure $\mathcal{S}$ has to be defined and no critical parameter has to be set (results are quite insensitive to the single parameter, λ). Since OFDT is a local method, it offers more flexibility, but in its current form its performance still depends on the particular choice of the similarity measure, the estimation of the flow vectors (which can be wrongly noisy in non-DT regions), and on having to properly set the parameters in the specific optic flow method used. This latter difficulty has also been experienced by other authors [4].

3 The Approach by Fazekas et al. [25]

The methods in Sect. 2 propose two approaches to exploit the spatio-temporal irregularity of optical flow in DT regions. In contrast, since DT does not follow the BCA, three alternative flow assumptions are explored in [25]: gradient constancy (GC), color constancy (CC) and brightness conservation (BC). These represent different attempts to account for illumination changes to capture the dynamics of DTs. Then, the DT detection problem is posed as that of minimizing the functional $F(u, v, \tilde{u}, \tilde{v}, C) = G(L_1; \Omega_1) + G(L_2; \Omega_2) + G(S; \Omega_1 \cup \Omega_2) + \nu|C|$, where $G(H; \Omega) = \int_\Omega H(u, v, \tilde{u}, \tilde{v})\, \mathrm{d}x\, \mathrm{d}y$; the first term integrates a Lagrangian $L_1(u, v)$ for the flow (u, v) following the BCA over a non-DT region Ω_1; the second term integrates a Lagrangian $L_2(\tilde{u}, \tilde{v})$ for the flow $(\tilde{u}, \tilde{v})$ obeying an alternative assumption (GC, CC, BC) over DT region Ω_2; the third term seeks the smoothness $S(u, v, \tilde{u}, \tilde{v})$ of both flows over the whole image; and the last term aims at penalizing long contours C separating regions Ω_1 and Ω_2, with ν being a scaling parameter. The unknown discontinuity set makes the direct minimization of the functional F hard. Thus, the authors reformulate the problem as a level-set functional

$$F_{LS}(u, v, \tilde{u}, \tilde{v}, \phi) = \int \left[T_1(u, v; \phi) + T_2(\tilde{u}, \tilde{v}; \phi) + T_3(u, v, \tilde{u}, \tilde{v}) + T_4(\phi)\right] \mathrm{d}x\, \mathrm{d}y, \tag{6}$$

where $T_1 = E_1(u, v) \cdot H(\phi)$, $T_2 = (\gamma E_2(\tilde{u}, \tilde{v}) + \rho)H(-\phi)$, $T_3 = S(u, v, \tilde{u}, \tilde{v})$, and $T_4 = \nu|\nabla H(\phi)|$. E_1 and E_2 are energy functions for the BCA and some of the alternative assumptions (GC, CC, BC) respectively, ϕ is an indicator function the sign of which distinguishes Ω_1 from Ω_2, $H(\cdot)$ is the Heaviside function, and S measures the flow smoothness. Parameter γ weights one kind of flow against the other; ρ is required to prevent Ω_2 to become the whole image, which may happen since the alternative flow assumptions are more general than the BCA; ν is used to adjust the smoothness of the contour C; parameters α, $\tilde{\alpha}$, $\tilde{\beta}$ are used in the smoothness terms for BC and CC; parameter λ is used in the definition of the energy terms E_2; and the Heaviside function uses a parameter ϕ_0. About a dozen parameters have to be set.

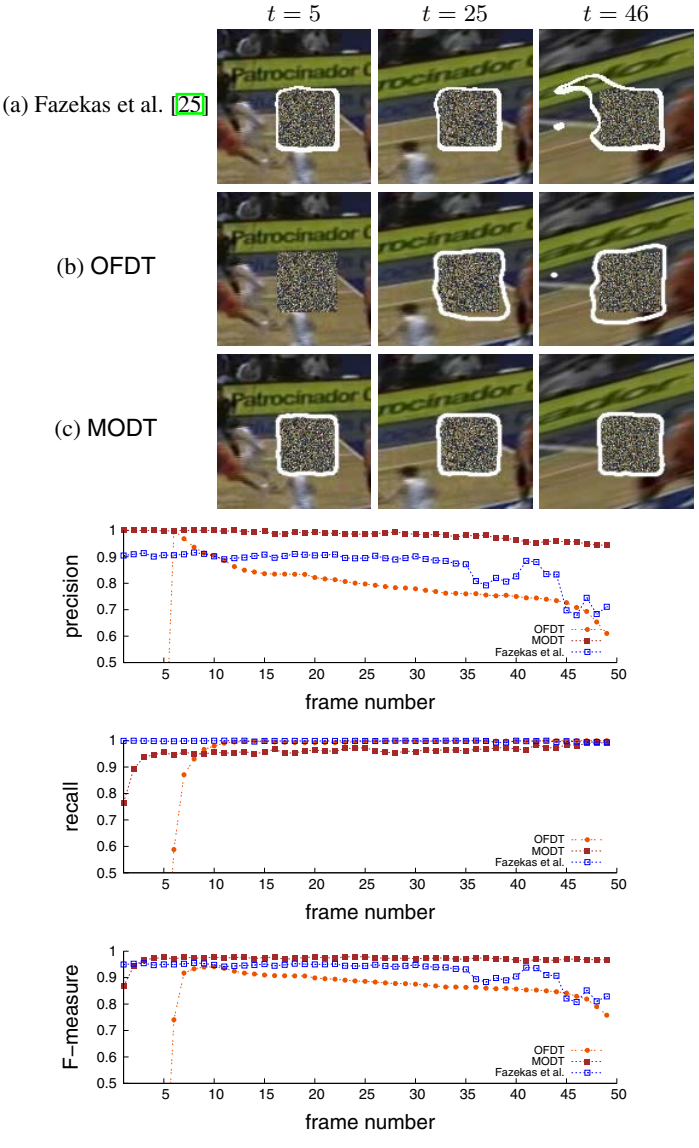
In addition to this sophisticated level-set approach, Fazekas et al. [25] considered a simpler non-level-set based approximation to DT, consisting of using the residual of the optical flow (v_x, v_y), defined as $r(v_x, v_y) = (I(x + v_x, y + v_y, t + 1) - I(x, y, t))^2$, and then setting a threshold for it. This method is reported to be very sensitive to the choice of the threshold for the residual, and fails under several conditions such as significant egomotion. They suggest yet another idea consisting of comparing this residual $r = r(v_x, v_y)$ with the no-flow residual $r_0 = r(0, 0)$, so that DT is flagged when $r_0 - r < \theta_r$, the threshold θ_r being obtained from a simple parametric classifier. It is assumed that DT and non-DT regions are linearly separable in this residual difference space. However, it is possible that one of the regions dominates the other, causing the parametric thresholding to be no longer effective. This idea has been recently revisited [28] by including a spatiotemporal median filter of optic flow residual maps. An adaptive way to set the threshold for these maps is also suggested, which relies on the DT occupying a "significant" part of the data in the first n frames. However, no analysis is provided on how effective this adaptive threshold is.

4 Results

Short synthetic videos have been generated so that ground-truth dynamic texture maps are available, which allows a quantitative comparison. Additionally, qualitative comparisons are performed with video sequences of real DTs. Our DT detection results are depicted with the contours of the connected components in the DT map $\mathcal{DT}$. To distinguish exterior from interior contours, they are depicted in white and black, respectively. Thus, a black contour within a white one means a region that is *not* DT.

The outlier identification for the MODT method uses a robust parametric 2D motion estimation algorithm [27]. Our proposed methods are compared to the level-set-based system by Fazekas et al. [25] using an implementation provided by its authors. No comparison is performed with their fast approximations since they are less accurate than their level-set procedure, and we are interested in using their most accurate results. We have focussed our comparative analysis to learning-free, motion-based methods and in particular to studying whether alternative flow models to the BCA are actually required. Therefore, we make no comparison with non-motion based or learning-based methods, such as AR or Markov-based models, which are outside the scope of our work here.

For MODT, the parameter $\lambda = 0.5$ is set for all tests. For OFDT, the Lucas-Kanade method for optical flow computation was applied, on 5×5-sized windows, after Gaussian averaging and downsampling. The similarity threshold was $\theta_s = 0.95$, and the temporal parameter was $\lambda = 0.9$ for the synthetic sequences and $\lambda = 0.7$ for the real ones. For Fazekas et al. [25], the parameters were in the values set in their software and their paper [25]. For the synthetic sequences, only the size of the images was changed accordingly, and the number of levels in the Gaussian image pyramid was set to 3 instead of 4, as required for the smaller image sizes used (128×128). For the real sequences, the results were not recomputed, but directly taken from the web page [29] associated with [25]. In this case, interior and exterior contours are both in red.



(d) Precision, recall and F-measure

Fig. 1. Exp. 1: background moving affinely, random static DT

4.1 Synthetic Sequences

Quantitative assessment based on ground-truth DT masks, uses the number of true positives p (DT correctly classified as such), false positives $\bar{p}$ (non-DT misclassified as DT) and false negatives $\bar{n}$ (DT misclassified as non-DT). From these, precision π and recall ρ are computed, respectively, as $\pi = \frac{p}{p+\bar{p}}$ and $\rho = \frac{p}{p+\bar{n}}$. The F-measure, $F = \frac{2 \cdot \pi \cdot \rho}{\pi + \rho}$,

combines π and ρ to summarize the performance as a single value. All these measures range in $[0, 1]$, and the higher, the better.

Influence of background motion (Experiment 1). A background image is moved affinely, while a rectangular-shaped dynamic texture is built fully randomly (the value for each pixel in each frame is independently drawn from a uniform distribution). Results at three frames of the 50-frame sequence are shown in Fig. 1. It can be seen that while all approaches behave well most of the time, at the end of the sequence, the Fazekas et al. [25] approach and OFDT have many false positives. OFDT does not detect the DT at the beginning since, in (1), $\lambda = 0.9$ weights past information much more than the new, and the DT likelihood map is initialized to 0. The evolution of π, ρ and F over time is given in Fig. 1(d). Generally, MODT is more precise than Fazekas et al. [25] and OFDT. Both of these latter methods, and particularly [25], have higher recall, at the expense of more false detections. The overall behavior, as captured by F, is best for MODT. This example illustrates how the global approach of MODT is more robust against egomotion conditions than the optical flow alternatives OFDT and Fazekas et al. [25]. As a quantitative guide, average and standard deviations of π, ρ and F over the sequence are collected in Table 1 for this and the following experiments.

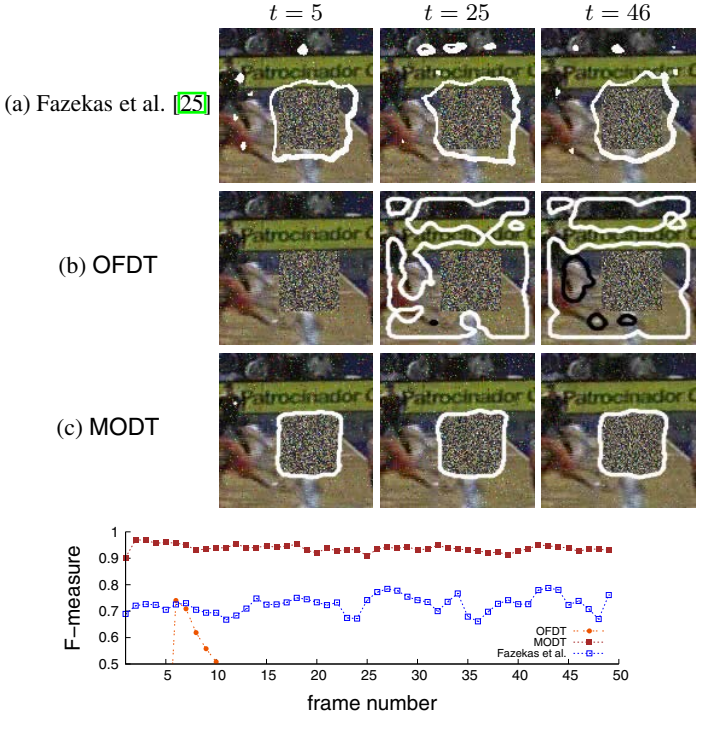


Fig. 2. Exp. 2: background with 5% random noise, random static DT

Table 1. Average (μ) and standard deviation (σ) for precision (π), recall (ρ) and F-measure (F) for all the synthetic experiments

		π		ρ		F	
Exp.	Method	μ_π	σ_π	μ_ρ	σ_ρ	μ_F	σ_F
1	[25]	0.87	0.06	1.00	0.00	0.93	0.04
	OFDT	0.70	0.27	0.86	0.33	0.77	0.29
	MODT	0.96	0.14	0.94	0.14	0.97	0.02
2	[25]	0.57	0.04	1.00	0.00	0.73	0.03
	OFDT	0.24	0.14	0.87	0.33	0.37	0.16
	MODT	0.87	0.13	0.97	0.14	0.94	0.01
3	[25]	0.70	0.04	1.00	0.00	0.82	0.02
	OFDT	0.60	0.24	0.86	0.33	0.70	0.26
	MODT	0.73	0.11	0.95	0.14	0.84	0.01

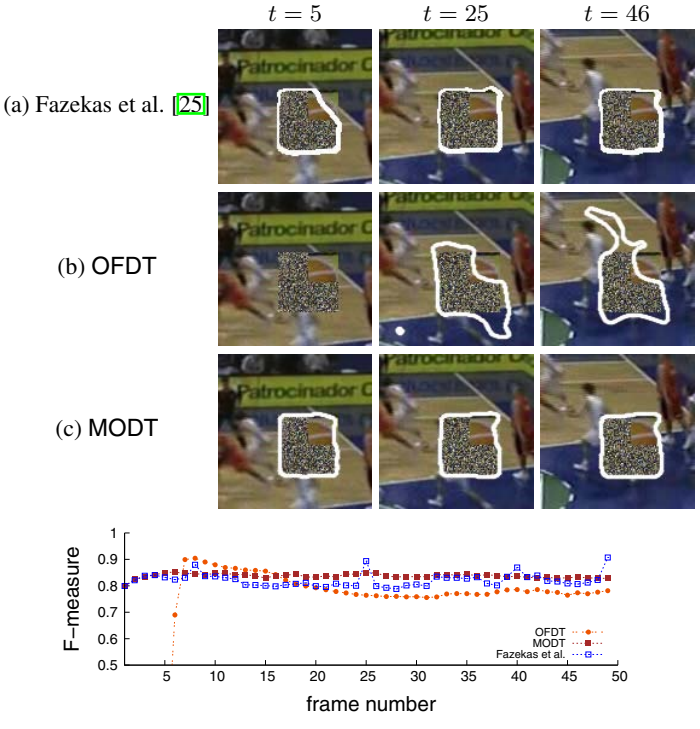
Influence of image noise (Experiment 2). Random noise (5%) is added to each frame of a sequence with a static background, and a DT region is generated randomly, as before. As shown in Fig. 2, the local methods (OFDT and Fazekas et al. [25]) exhibit many false positives, while the misclassifications are much fewer in MODT. This better performance by MODT is quantitatively reflected in the F-measure (Fig. 2(d) and Table 1). OFDT results are poorer than Fazekas et al. [25], possibly because Fazekas et al. [25] compute the flow with the Horn-Schunk (HS) method which, unlike Lucas-Kanade (LK), enforces global smoothness of the flow. However, in agreement with [4], we found it harder to select the right parameters for HS than for LK.

Influence of independent motion (Experiment 3). For this experiment, the background translates at a constant speed. The values of three-quarters of a rectangle (in an L-shaped form) is set randomly and the remaining top-right quarter has constant contents. This constant region is equivalent to an independently moving object. As illustrated in Fig. 3, MODT fails to detect the true extent of the DT region, as it misdetects the constant region as DT. The reason is that, given the global parametric motion of the background, both the true DT and the constant region are indistinguishably treated as motion outliers. While Fazekas et al. [25] works slightly better in this case, its behavior is unstable and surprisingly poor since, given its local nature and its elaborate design, it should deal with this situation more successfully. In fact, Fig. 3(b) shows the results of the (also local) OFDT approach, which exhibits quite promising results. The F-measure for the three methods is compared in Fig. 3(d) and Table 1.

Even though not shown here, tests of OFDT with a variational optical flow computation method were conducted and, when applied to these synthetic sequences, yielded extremely good results. While this indicates that OFDT in its current form is still sensitive to how the optic flow is estimated, it also provides strong evidence of the validity and interest of the approach.

4.2 Real Sequences

In order to test the approaches with sequences of real dynamic textures (fire, water, steam, smoke, etc.), the DynTex database [30] has been used. Results for three arbitrarily selected frames of some of these sequences are shown in Figs. 4–6. Fig. 4 illustrates



(d) F-measure

Fig. 3. Exp. 3: translating background, L-shaped random DT and static constant region

the results for a waterfall sequence while the camera is moving horizontally. It can be seen that the proposed approaches, despite their simpler design, work as well as, if not better than, Fazekas et al. [25], at higher frame rate (see Table 2).

The results for a smoke sequence are shown in Fig. 5. While all approaches behave reasonably well, they all have problems with detecting dense (smoke) regions, an issue related with the aperture problem. However, similarly to the synthetic examples, Fazekas et al. [25] tends to miss more true DT and misdetects some non-DT in comparison to OFDT and MODT. At $t = 100$, OFDT misdetects some regions in the wall as DT (lower-right part of the image), possibly because of illumination changes, a situation which is dealt with more robustly by MODT and Fazekas et al. [25].

Finally, Fig. 6 illustrates an example where Fazekas et al. [25] offers results better than MODT but worse than OFDT. MODT does not fare as well due to the large degree of motion outliers or their uneven distribution. On the other hand, the duck is correctly left undetected as DT most of the time by all methods, but all of them have problems when it moves faster at the end of the sequence. The likely reason for this is that, as in the third synthetic example, MODT, as a global parametric motion estimation method, is unable to distinguish independently moving objects from true DT. Indeed, OFDT as a local method, in this sequence (Fig. 6(c)) and others, can deal with these kinds of scenarios as well or better than Fazekas et al. [25].

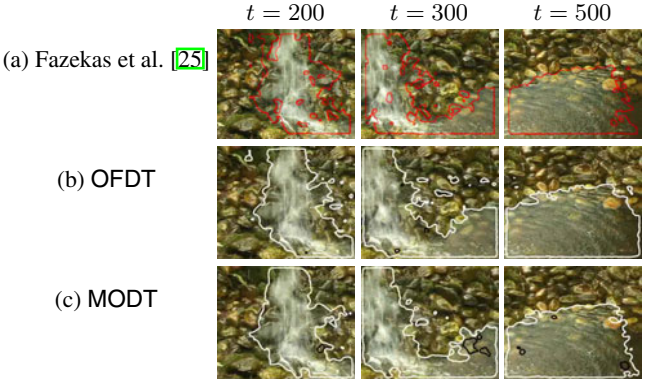


Fig. 4. Results with real sequence 6481i10.avi with the three methods

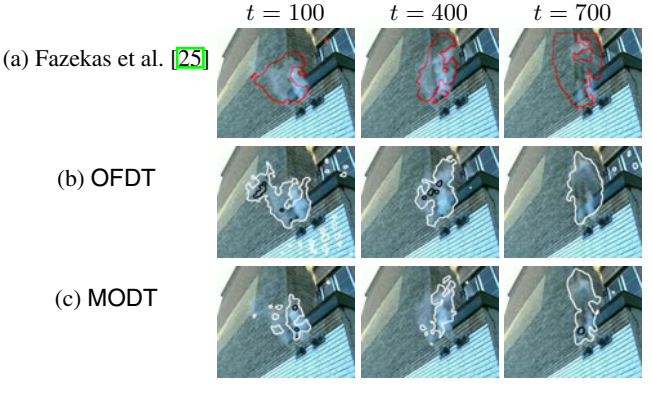


Fig. 5. Results with real sequence 648ea10.avi with the three methods

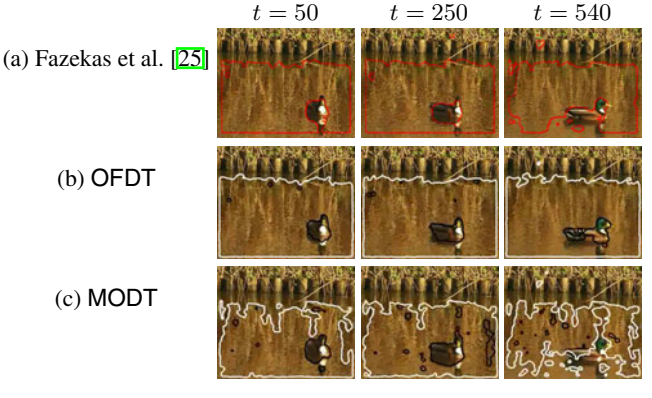


Fig. 6. Results with real sequence 644ce10.avi with the three methods

Table 2. Computation times for the three methods

Figure	DynTex sequence	# frames	Fazekas et al. [25]	OFDT	MODT
4	6481i10.avi	670	$>$ 5 hours	9.1 min.	4.3 min.
5	648ea10.avi	884	$>$ 7 hours	8.8 min.	5.3 min.
6	644ce10.avi	540	$>$ 4 hours	6.0 min.	3.8 min.
Average time per frame (s)			29	0.6	0.4

The algorithms were timed running on an Intel Pentium 1.7 GHz PC for sequences of 352×288-sized frames. Both OFDT and MODT perform significantly faster than Fazekas et al. [25] as shown in Table 2. These times are only approximate measurements of unoptimized implementations (including even I/O operations).

5 Conclusions

In comparison to recent DT methods, we proposed two simpler and much faster approaches. Unlike recently suggested in [25], we show that an alternative flow model to the BCA is *not* required. The key observation is that locations of DT can be detected as violations of the BCA either as a lack of spatio-temporal coherence of locally computed optical flow (method OFDT), or as motion outliers detected with some robust global parametric motion estimation (method MODT). Competitive results with both synthetic and real sequences were presented. Future work for improved DT detection is to explore the combination of the robustness of the global parametric method with the flexibility of the local optical flow approach in a joint framework.

Acknowledgements. V. J. Traver was funded by grant JC2008-00397 from the Spanish Ministerio de Ciencia e Innovación. We thank the Vista research team at Irisa/Inria Rennes for the use of Motion2D software, the authors of [25] for their code, and the Spanish research programme Consolider Ingenio-2010 for grant CSD2007-00018.

References

1. Chetverikov, D., Péteri, R.: A brief survey of dynamic texture description and recognition. In: Proc. Intl. Conf. Computer Recognition Systems, pp. 17–26 (2005)
2. Péteri, R., Chetverikov, D.: Dynamic texture recognition using normal flow and texture regularity. In: IbPRIA, pp. 223–230 (2005)
3. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. IJCV 51, 91–109 (2003)
4. Atcheson, B., Heidrich, W., Ihrke, I.: An evaluation of optical flow algorithms for background oriented schlieren imaging. Experiments in Fluids 46, 467–476 (2009)
5. Vezzani, R., Calderara, S., Piccinini, P., Cucchiara, R.: Smoke detection in video surveillance: The use of ViSOR. In: ACM IVR, pp. 289–297 (2008)
6. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE-PAMI 29, 915–928 (2007)
7. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: CVPR, vol. 2, pp. 58–63 (2001)

8. Hyndman, M., Jepson, A., Fleet, D.: Higher-order autoregressive models for dynamic textures. In: BMVC (2007)
9. Chan, A.B., Vasconcelos, N.: Variational layered dynamic textures. In: CVPR (2009)
10. Vidal, R., Ravichandran, A.: Optical flow estimation and segmentation of multiple moving dynamic textures. In: CVPR, pp. 516–521 (2005)
11. Chan, A., Vasconcelos, N.: Layered dynamic textures. IEEE-PAMI 31, 1862–1879 (2009)
12. Doretto, G., Cremers, D., Favaro, P., Soatto, S.: Dynamic texture segmentation. In: ICCV, vol. 2, pp. 1236–1242 (2003)
13. Campisi, P., Maiorana, E., Neri, A., Scarano, G.: Video texture modelling and synthesis using fractal processes. IET Image Processing 2, 1–17 (2008)
14. Lu, Z., Xie, W., Pei, J., Huang, J.: Dynamic texture recognition by spatiotemporal multiresolution histograms. In: IEEE Workshop. on Motion & Video Computing, vol. 2, pp. 241–246 (2005)
15. Ghanem, B., Ahuja, N.: Extracting a fluid dynamic texture and the background from video. In: CVPR (2008)
16. Toreyin, B., Cetin, A.: HMM based method for dynamic texture detection. In: IEEE 15th. Signal Processing and Communications Applications (2007)
17. Ferrari, R.J., Zhang, H., Kube, C.R.: Real-time detection of steam in video images. PR 40, 1148–1159 (2007)
18. Xiong, X., Caballero, R., Wang, H., Finn, A.M., Lelic, M.A., Peng, P.Y.: Video-based smoke detection: Possibilities, techniques, and challenges. In: IFPA (2007)
19. Toreyin, B., Cetin, A.: On-line detection of fire in video. In: CVPR (2007)
20. Corpetti, T., Memin, E., Pérez, P.: Dense estimation of fluid flows. IEEE-PAMI 24, 365–380 (2002)
21. Toreyin, B.U., Dedeoglu, Y., Cetin, A.E.: Wavelet based real-time smoke detection in video. In: EUSIPCO (2005)
22. Rahman, A., Murshed, M.: Real-time temporal texture characterisation using block based motion co-occurrence statistics. In: ICIP, pp. III: 1593–1596 (2004)
23. Bouthemy, P., Hardouin, C., Piriou, G., Yao, J.: Mixed-state auto-models and motion texture modeling. JMIV 25, 387–402 (2006)
24. Fazekas, S., Chetverikov, D.: Analysis and performance evaluation of optical flow features for dynamic texture recognition. Signal Processing: Image Comm. 22, 680–691 (2007)
25. Fazekas, S., Amiaz, T., Chetverikov, D., Kiryati, N.: Dynamic texture detection based on motion analysis. IJCV 82, 48–63 (2009)
26. Viola, P.A., Jones, M.J.: Robust real-time face detection. IJCV 57, 137–154 (2004)
27. Odobez, J., Bouthemy, P.: Robust multiresolution estimation of parametric motion models. Int. J. Visual Communication and Image Representation 6, 348–365 (1995)
28. Chetverikov, D., Fazekas, S., Haindl, M.: Dynamic texture as foreground and background. In: MVA (2010), doi:10.1007/s00138-010-0251-6 (Published online: February 21, 2010)
29. Fazekas, S., Amiaz, T., Chetverikov, D., Kiryati, N.: (Dynamic texture detection and segmentation), http://vision.sztaki.hu/~fazekas/dtsegm
30. Péteri, R., Huskies, M., Fazekas, S. (DynTex: a comprehensive database of dynamic textures), http://www.cwi.nl/projects/dyntex

Finding Semantic Structures in Image Hierarchies Using Laplacian Graph Energy

Yi-Zhe Song[1], Pablo Arbelaez[2], Peter Hall[1], Chuan Li[1], and Anupriya Balikai[1]

[1] MTRC, University of Bath, Bath, UK, BA2 7AY
[2] University of California at Berkeley - Berkeley, CA 94720
{yzs20,pmh,cl249,ab368}@cs.bath.ac.uk, arbelaez@eecs.berkeley.edu

Abstract. Many segmentation algorithms describe images in terms of a hierarchy of regions. Although such hierarchies can produce state of the art segmentations and have many applications, they often contain more data than is required for an efficient description. This paper shows Laplacian graph energy is a generic measure that can be used to identify semantic structures within hierarchies, independently of the algorithm that produces them. Quantitative experimental validation using hierarchies from two state of art algorithms show we can reduce the number of levels and regions in a hierarchy by an order of magnitude with little or no loss in performance when compared against human produced ground truth. We provide a tracking application that illustrates the value of reduced hierarchies.

1 Introduction

Hierarchical descriptions of images have long been recognized as being valuable to computer vision, the literature on how to build them and use them is vast. Ideally, hierarchies reflect assemblies that comprise real world objects, but in practice they can often be very large and complex. There are significant practical advantages to be had by simplifying hierarchical descriptions, for example we can expect gains in memory efficiency, speed and the hierarchies might be more semantically meaningful. Yet these advantages will be conferred only if the quintessential character of the object is retained by the simplification process. This paper provides a general purpose method to filter complex hierarchies into simpler ones, independent of the way in which the hierarchies are formed, with little or no loss in performance when benchmarked against ground truth data.

There are many reasons for making hierarchal descriptions and many ways to make them; the literature is vast, making a full review impossible here. In any case, we emphasize, *this paper is not about segmentation per se, nor it is about making hierarchies — it is about filtering hierarchies.* Since our purpose is extracting semantic structures from hierarchies rather than proposing algorithms for constructing new ones, we bypass the large literature on hierarchical segmentation and review only a few representatives of successful approaches.

Sieves [4] are a well established example. They are built using morphological operators to generate a tree rooted around gray level extrema in an image. Sieves

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 694–707, 2010.

are related to maximally stable extremal regions (MSER) which are made by filtering a hierarchy comprising binary regions in which each level is indexed by a gray level threshold [16]. The filtering criterion is stability, which is defined as the rate at which a region changes area with respect to the control parameter (threshold). Sieve trees are very complex, MSER trees are simpler by comparison; yet both have found applications and both address the important issue of segmentation, which is a major theme in this paper.

Mean-shift [5] is amongst the best known of the recent segmentation algorithms. A recent interesting development from Paris and Durand [18] observes that thresholds in feature space density lead directly to image space segmentations, and uses the notion of stability in feature space to produce a hierarchal description. Their definition of stability differs from that used to build MSER trees, but there is a common spirit of persistence as control variables change.

Normalized cuts [21] is another of the most widely used and influential approaches to segmentation. This approach is principled, resting as it does on spectral graph theory. Yet, it tends to produce arbitrary divisions across coherent regions in ways that are not intuitive to humans, breaking large areas such as the sky, for example. In response, there is now a sizable literature on various additions and modifications to suit specific circumstances. These include the popular multi-scale graph decompositions [6] which are directly related to hierarchical descriptions because smaller objects are children to larger ones.

The connected segmentation tree (CST) [2], which it has its roots in the early work by Ahuja [1], is specifically designed to yield semantically meaningful hierarchies. The CST takes into account the photometric properties, spatial organization, and structure of objects. It is very successful in identifying taxonomies amongst objects are therefore demonstrates the value of simple hierarchical descriptions.

The most successful boundary detectors to date are rooted in the probability of boundary (Pb) maps introduced by Martin *et al* [14]. The Pb maps compared very well against human produced ground truth using the Berkeley Segmentation Dataset (BSDS) [15], and recent improvements include multiscale analysis [19] and the use of global image information [17]. The latter are of particular interest because global-Pb lead to state of the art region hierarchies [3].

We simplify a hierarchy solely by the removal of levels, typically reducing their number by one or even two orders of magnitude. Others also simplify hierarchies: MSER simplifies a hierarchy in which thresholds make levels [16]; Kokkonis and Yuille [11] use a heuristic that estimates the cost of completing a given graph to reach a goal graph within an A^* search for structure coarsening; computational geometry and computer graphics offer many examples related to mesh simplification.

The **contribution** of this paper is to generically filter hierarchical descriptions with little or no loss of descriptive power compared to human ground truth, and with the exceptions of MSER and CST all the above hierarchies are typical in being large and complex. In particular, our contributions are three-fold:

- The extension of the notion of Laplacian energy from spectral graph theory to non-connected graphs.
- Its application as a measure of graph complexity, to finding meaningful segmentations.
- Extensive quantitative and qualitative evaluation proving that our approach preserves the semantic quality of the input hierarchy while reducing considerably its complexity.

Another important aspect of our method is that it filters hierarchies *after* they have been constructed. This means that we can apply our method to many different hierarchies. The reduced hierarchies we output have a sensible semantic interpretation in terms of objects and object parts.

Our method is fully explained in Section 3 , but broadly it considers each level to be a segmented partition of an image. Nodes of the graph at any level are the segmented regions which form a region adjacency graph (RAG) [23,25] by a neighboring relationship. We compute the complexity of the graph on each level using Laplacian graph energy and keep levels whose complexity is smaller than either of the neighboring levels. We make no attempt to simplify the graph within a level. The value of our filtering is demonstrated by experiment in Section 4. We continue by developing our intuition regarding Laplacian graph energy.

2 Laplacian Graph Energy as a Complexity Measure

Graph complexity can be measured in several ways [8] and is of value to applications including but not limited to embedding [20], classification [22], and the construction of prototypes [24]. The complexity measure we use is based on *Laplacian graph energy* defined by Gutman and Zhou [10]. Laplacian graph energy is attractive in the context of this paper because it favors the selection of regular graphs, and particularly favors polygonal graphs.

Let G be a unweighted graph of n vertices and m edges, i.e., a (n, m)-graph and A be its adjacency matrix. Let d_i be the degree of the ith vertex of G and D be the corresponding degree matrix, where $D(i, i) = d_i$. Then $L = D - A$ is the Laplacian, and the LE is defined [10] to be

$$\mathcal{LE}(G) = \sum_{i=1}^{n} \left| \lambda_i - \frac{2m}{n} \right| \tag{1}$$

In which: the λ_i are eigenvalues of the Laplacian matrix and $2m/n$ is the average vertex degree. Gutman and Zhu [10] prove that $\mathcal{LE}(G)$ falls into the interval

$$\mathcal{I}[G] = [2\sqrt{M}, 2M] \tag{2}$$

in which

$$M = m + \frac{1}{2} \sum_{i}^{n} \left(d_i - 2\frac{m}{n} \right)^2 \tag{3}$$

Gutman and Zhu [10] also prove that bipartite graphs are at both ends of the interval; that with $m = n/2$ at the lower bound and $m = n^2/2$ at the upper bound.

Our aim now is to characterize the behavior of LE as a graph changes, typically because of a computer vision algorithm. A graph can change in the number of nodes, arcs, and also in arc permutation. The effect of addition and deletion of arcs on graph energy is at best difficult to predict, it can go up or down [7]. We aim to show that in a set of graphs with a fixed number of nodes and arcs, the LE of the polygonal graph is likely to be a lower bound.

We begin by noticing that if we set ν to be the variance of node degree, for a unweighted graph G, then we have $M = m + n\nu/2$. The interval containing the LE for G can now be expressed as

$$\mathcal{I}[\nu; m, n] = \left[2\left(m + \frac{n\nu}{2}\right)^{1/2}, 2\left(m + \frac{n\nu}{2}\right)\right]. \tag{4}$$

which shows that the interval is parametrized by variance. The variance is zero if and only if G is a regular graph, in which case $\nu = 0$ and the interval is $\mathcal{I}[0; m, n] = [2\sqrt{m}, 2m]$; we note this interval depends only on the number of arcs. If the graph has $\nu > 0$, then the corresponding interval bounds rise and the interval widens.

Suppose a fixed number of arcs m and nodes n. If $m < n - 1$, then this graph is disconnected; but in practice we compute LE for connected components only (see Section 3). We assume $m \geq n$ from this point. For such an (n, m) graph, the variance in degree node depends solely on how the arcs are distributed and only regular graphs have zero variance. Allowing for permutations, there is at most one regular graph in the set. The LE for such a regular graph falls into the smallest interval, $\mathcal{I}[0; m, n]$, taken over the whole (n, m) family. If the variance rises then the LE is drawn from an interval with a lower bound greater than $2\sqrt{m}$ and an upper bound greater than $2m$. The left of Figure 1 provides an illustration of how variance, the bounding interval and LE relate to each other, when considering a family of graphs where $n = m = 8$. It shows that the LE for each graph (solid black) approximates the M value (solid red), and is bounded by $[2\sqrt{M}, 2M]$ (dotted red). Graphs with large variance (to the right edge of the figure) have higher, wider intervals, as indicated by the vertical line.

We can extend our intuition further by considering graphs of fixed n but an increasing number of edges, summarized in the right of Figure 1. It shows the output from a simulation in which arcs were randomly placed over a graph of eight nodes by thresholding a symmetric random matrix. Threshold values were chosen so that one new arc was added at each step, starting with isolated nodes and building to a complete graph. The Figure shows an LE trajectory, in red, having one main peak. This occurs on the most irregular graph. The black curve plots our modified LE, defined in Section 3. It shows the effect of allowing for graphs comprising many connected components. The first peak corresponds to graphs with many small components. Notice that when the graph comprises a single component, our modified energy corresponds to the standard LE, seen where and the two curves coincide. The graph with the lowest energy is to the

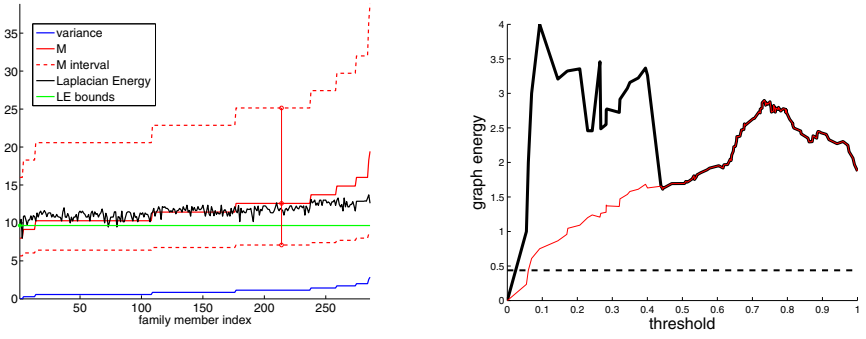


Fig. 1. Left: The relation between variance, the bounding interval and LE for graphs randomly drawn from an (m, n) family, where $(m = 8, n = 8)$. The family is ordered into sets of equal variance. The contour of LE through the (m, n) polygon is drawn in green. **Right:** The modified LE for random graphs (see Section 3) picked from $n = 8$ family, formed by adding one edge to each previous member. Note the two main peaks, the right peak corresponds to a single connected component. The standard LE is shown in red.

far left of this region, it is closest to a (m, n) polygon. The complete graph is the rightmost. This is empirical evidence that LE is minimal for polygonal graphs.

In summary, Laplacian graph energy is a broad measure of graph complexity. Regular structures which tend to be visually meaningful, such as polygons, exhibit lower Laplacian graph energy than structures comprising randomly selected arcs.

3 Using Laplacian Graph Energy to Filter Hierarchies

We suppose a full hierarchical description comprises a collection of N distinct levels, our problem is to determine $M << N$ levels needed in a reduced hierarchy. These M levels must preserve the semantic content of the full hierarchy. We begin our account by being more concrete about the hierarchies we have in mind. Image primitives, which are connected regions, reside at the bottom level of the hierarchy and partition the input image into a RAG — so nodes are synonymous with regions and only neighboring regions can be adjacent. A combination of primitives makes a parent region that is larger in size and which resides on the level directly above its children. The union of all regions at any level partition the image and also constitute a RAG, but in addition we have links between levels that specify child-parent relations. We constrain the RAG so that only children of a common parent can be adjacent, similar to the CST [2]. We assume that parent regions can be combined in recursive fashion, thus generating new levels. Such combination continues until the production algorithm halts; the halting criterion is algorithmic dependent but a level comprising a single region which covers the whole image provides a universal terminating case. It follows that

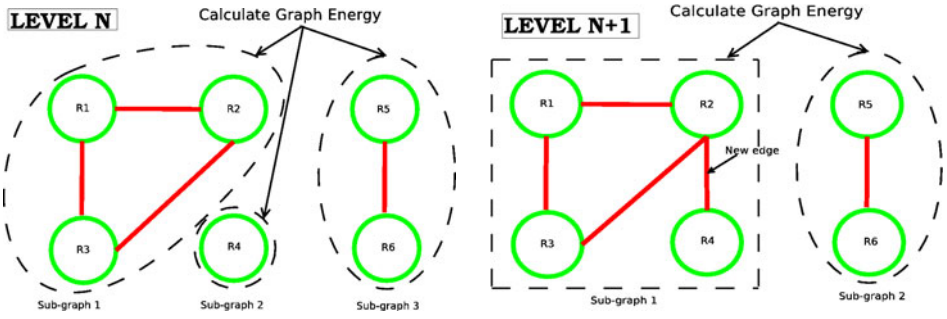


Fig. 2. Illustrating how graph energy is calculated on two levels of the hierarchy. Each node R_i corresponds to a region primitive. When the level increases, sub-graphs merge to create larger ones while the number of connected components falls.

we can represent any such hierarchy by a collection of levels, each one being a distinct partition of the input image, and each region in each level will bottom out into a distinct collection of region primitives. Moreover, each region the full hierarchy is partitioned by its children, its grand-children and eventually by its ancestral image primitives. Notice that we can represent such a hierarchy via an image map and therefore all arcs are implicitly specified.

Our principle in solving the above problem is to choose those levels that are lower in complexity than their neighbors, which follows the intuition developed in Section 2. We measure complexity via Laplacian graph energy, as defined above in Section 2. Note that rather than simplifying the hierarchy as a whole [9], we select levels of the hierarchy that exhibit lower Laplacian graph energy. Two modifications to the standard definition of Laplacian graph energy were proposed. First, we propose to use a weighted matrix A in which the element in row i and column j is given by

$$a_{ij} = \exp(-\frac{w_{ij}}{w_{max}}) \tag{5}$$

where w_{ij} is the average boundary strength between region i and region j, and w_{max} is a decay factor, set to the maximum over all w_{ij}. Thus our adjacency matrix is akin to the similarity matrix used in Normalized cuts [21].

Secondly, we introduce an extension to the standard definition of Laplacian graph energy that we call the *component-wise* Laplacian graph energy (cLGE). Such extension is motivated by the fact that we consider a scene to comprise a set of independent objects; within a hierarchy, these are defined by child-parent relationships. For a graph with K disconnected components, we define the cLGE to be

$$\xi = k \sum_{i=1}^{K} \frac{\mathcal{LE}(G_i)}{|n_i|} \tag{6}$$

in which G_i is ith connected component of $|n_i|$ nodes and k is the number of nodes in the whole graph. This is, at root, the sum of individual component

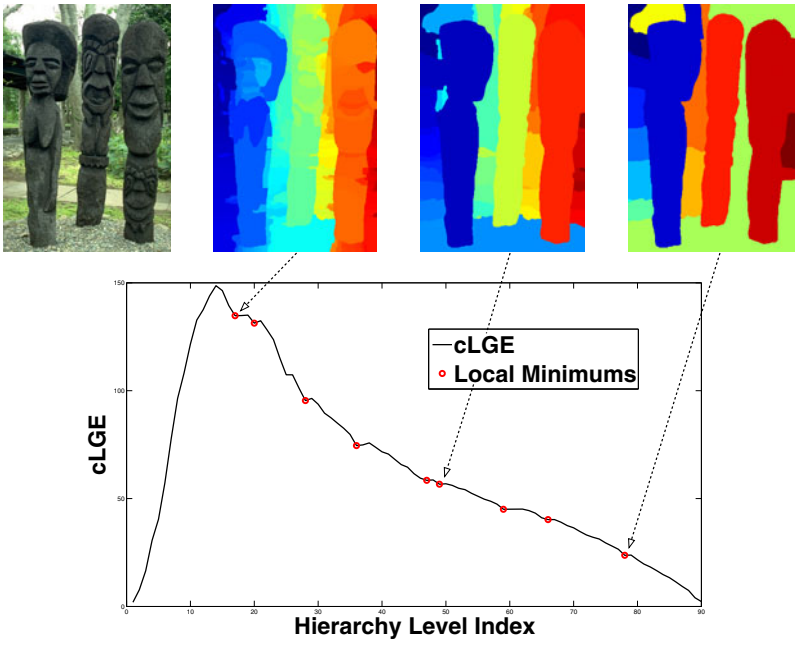


Fig. 3. Graph energy (equation 6) as a function of level index. Local minimum correspond to levels that are less complex compared to neighboring levels.

energies, but in which each is normalized by the number of nodes it contains. The scale factor k is used so that in the case of a single connected component our expression returns the original graph energy exactly.

We compute the cLGE at every level in the hierarchy independently using graphs built from the primitives at the lowest level; hence k in Equation 6 is the total number of image primitives. At the bottom level of the hierarchy, each primitive is a 1-node sub-graph on its own, whereas the top level forms a single connected graph. At intermediate levels, as segmentations become coarser, subgraphs are merged to create larger ones, and so the number of disconnected components will fall. Figure 2 illustrates how we compute cLGE over two levels with a simple graph of 6 nodes, each of which represents a region primitive. In this way, we numerically construct the function $\xi(z)$ where z is the level index. As z rises the number of regions falls, and each region covers a larger number of primitives.

As seen in Figure 3, cLGE for the level as a whole can rise or fall, depending on the way these primitives are connected. Following the intuition developed above (Section 2), $\xi(z)$ falls as individual connected components tend towards regular graphs which have minimal cLGE, so we keep those levels at which cLGE is locally minimal (circled in red in Figure 3). In the same figure, segmentations corresponding to selected local minimums are also shown, where finer visual details are retained in lower levels and semantic objects emerge at a higher level.

The different shapes of the plots in Figures 1 and 3 is explained by the number of primitives and edges, and the fact that the former figure uses un-weighted graphs whereas the latter uses weighted graphs

4 Results

This section presents both quantitative and qualitative results, beginning with **quantitative results** in Tables 1 and 2.

Both tables were constructed by evaluating both full and reduced hierarchies, we used the Berkeley Segmentation Dataset (BSDS) [15] as a foil against which to assess the retention of semantic information. We obtained benchmarks against not only the boundary models of images introduced in the original BSDS [15], but also against that of regions as well. Two state of the art segmentation hierarchies were benchmarked. One is due to [3], which is premised on global-Pb (gPb) edge maps, oriented watershed transform (owt) and ultrametric contour maps (ucm) which offers a convenient duality between boundary maps and hierarchical image segmentations. We refer to their algorithm as gPb-owt-ucm. As a comparison basis, we also include benchmark results of quad-trees with 8 levels [12], denoted as quad-tree-8.

The other algorithm is a topological approach to mean-shift authored by Paris and Durand [18], but with both owt and ucm implemented over its edge map representation, here referred to as Paris-owt-ucm. It is worth noting that Paris and Durand [18] obtained a F-measure of 0.61 on the original BSDS boundary benchmark, after applying the *-owt-ucm algorithm from Arbeláez *et al*, we observe an increase in performance signified by a F-measure of 0.63. Each of these algorithms yields hierarchies with hundreds of levels, yet experiments show that

Table 1. Boundary Benchmarks on the BSDS. Four new algorithms were benchmarked together with gPb-owt-ucm which is State-of-the Art. Results show little or no downgrade on F-measures of the cLGE filtered hierarchies (denoted *-cLGE) when compared to the originals, gPb-owt-ucm and Paris-owt-ucm [18]. Benchmark scores of a randomly filtered hierarchy (gPb-owt-ucm-M) are also given where a clear decrease on F-measures against gPb-owt-ucm-cLGE can be seen. Results of benchmarking quad-trees [12] with 8 levels are also included as a direct comparison basis.

Method	ODS	OIS	AP
human	0.79	0.79	-
gPb-owt-ucm	0.71	0.74	0.77
gPb-owt-ucm-cLGE	**0.71**	**0.72**	**0.77**
gPb-owt-ucm-M	0.67	0.57	0.69
Paris-owt-ucm	0.63	0.66	0.71
Paris-owt-ucm-cLGE	0.63	0.66	0.71
Paris-owt-ucm-M	0.61	0.65	0.48
quad-tree-8	0.37	0.39	0.26

Table 2. Region Benchmarks on the BSDS. We follow [3] to obtain region benchmarks for each of the four algorithms in Table 1. Again, *-cLGE delivered on-par benchmark scores against the originals on region covering criteria (leftmost three columns), Probabilistic Rand Index (PRI) and Variation of Information (VI). The right most columns shows the average number of nodes (AN) and levels (AL), demonstrating an order of magnitude improvement in nearly all cases.

Method	ODS	OIS	Best	PRI	VI	AN (AL)
human	0.73	0.73	-	0.87	1.16	-
gPb-owt-ucm	0.58	0.64	0.74	0.81	1.68	16267 (80)
gPb-owt-ucm-cLGE	**0.58**	**0.60**	**0.66**	**0.79**	**1.78**	**829 (8)**
gPb-owt-ucm-M	0.53	0.58	0.64	0.77	2.04	1349 (8)
Paris-owt-ucm	0.52	0.60	0.69	0.78	2.12	28448 (124)
Paris-owt-ucm-cLGE	0.52	0.60	0.68	0.78	2.03	5870 (28)
Paris-owt-ucm-M	0.50	0.60	0.68	0.77	2.07	6339 (28)
quad-tree-8	0.33	0.39	0.47	0.71	2.34	21845 (8)

each provides a high quality segmentation at some level within their representation, when compared to human segmented ground truth. Unfortunately, neither of them provide any method by which to choose these optimal level or levels: a method such as ours, which automatically picks semantic levels, is therefore potentially very useful.

In each case we create a full hierarchy of levels by thresholding the ucm output by the particular algorithm. We aim to demonstrate that our filtering technique is able to reduce the number of levels in full hierarchies which is usually in their hundreds down to only tens, yet retain semantic information. Columns of Tables 1 and 2 (boundary benchmarks and region benchmarks respectively) are exactly the same as these used by [3] apart from an extra column in Table 2; ODS, OIS and AP stand for Optimal Dataset Scale (best scale for the entire dataset), Optimal Image Scale (best scale per image) and Average Precision respectively, whereas Best (Best Covering Criteria), PRI (Probabilistic Rand Index) and VI (Variation of Information) are three different measures common in the literature to measure region segmentation quality instead of boundaries.

The right-most column in Table 2 provides the average number of nodes (AN) and the average number of levels (AL) for each of the benchmarked hierarchies across all of the 100 BSDS testing images. We refer to our reduced hierarchy by a graph energy suffix, *-cLGE. To introduce a control measure we also filtered by picking $M << N$ levels at random, with M determined via cLGE; we refer to these cases with the suffix *-M. Over the 100 testing images in BSDS, gPb-owt-ucm hierarchies contain an average of **80 levels**, whereas our reduced gPb-owt-ucm-cLGE only contains an average of **8 levels** which is an order of magnitude better. Similarly, Paris-owt-ucm has **124 average levels** whereas Paris-owt-ucm-cLGE reduce that to **28 levels**, again an order of magnitude improvement. The average number of regions is reduced from **16267** to **829** for gPb-owt-ucm when cLGE is used to select levels, and to **1349** when the M levels are randomly selected. Again we see an order of magnitude improvement, and we

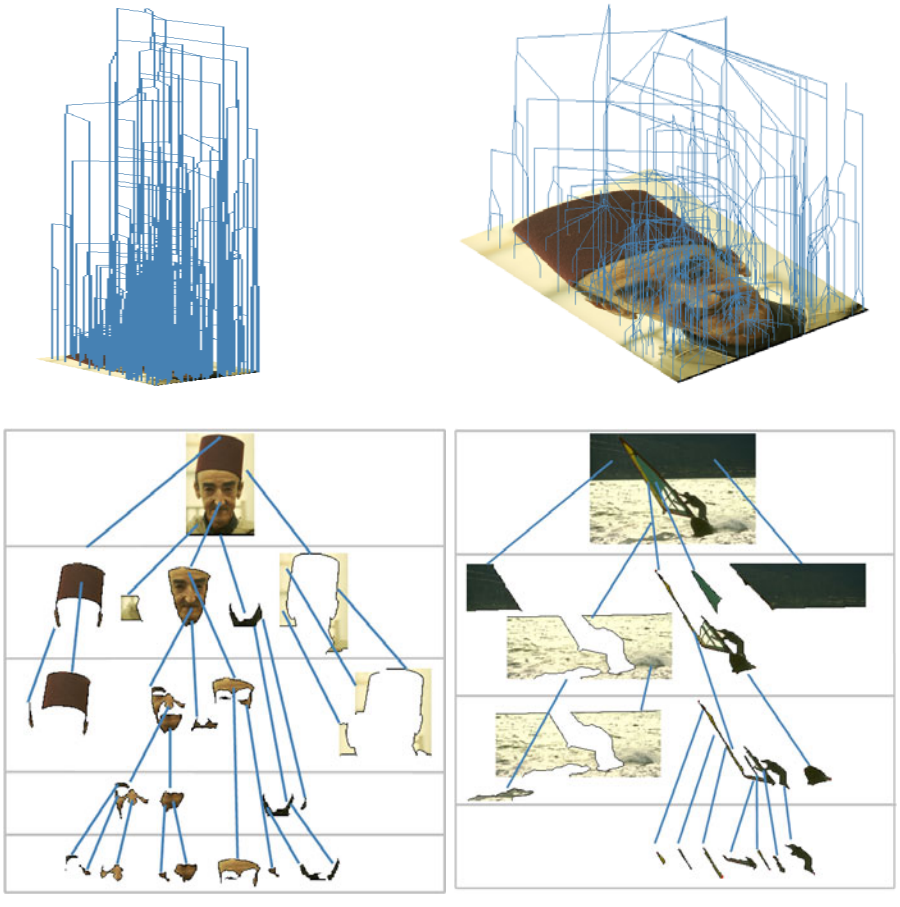


Fig. 4. The qualitative value of simplifying a hierarchy by removing levels. The top row visualizes merging regions as a tree, left over all levels of a gPb-owt-ucm hierarchy, right over the levels that remain after our filtering; bottom shows how objects are broken into useful parts (the original images are included for visualization purpose only).

conclude cLGE provides a non-random selection of levels. For Paris-owt-ucm we reduce the number of regions by about 1/5.

Despite many fewer levels and nodes, the table of boundary benchmarks, Table 1, shows cLGE filtered hierarchies retain the F-measures of the original, a similar story can be told in the region benchmark table, Table 2, with identical scores on ODS and fairly close ones on other measurements. In all cases, cLGE out performs our control of random selection. Overall, we see that cLGE retains benchmark scores of the original, while only keeping a small subset of its content.

In the rest of this section, we provide some **qualitative** results of the reduced gPb-owt-ucm hierarchies (gPb-owt-ucm-cLGE). In Figure 4, we offer visualizations of the original gPb-owt-ucm hierarchy and that of the reduced hierarchy (gPb-owt-ucm-cLGE), where a dramatic decrease in the number of levels is

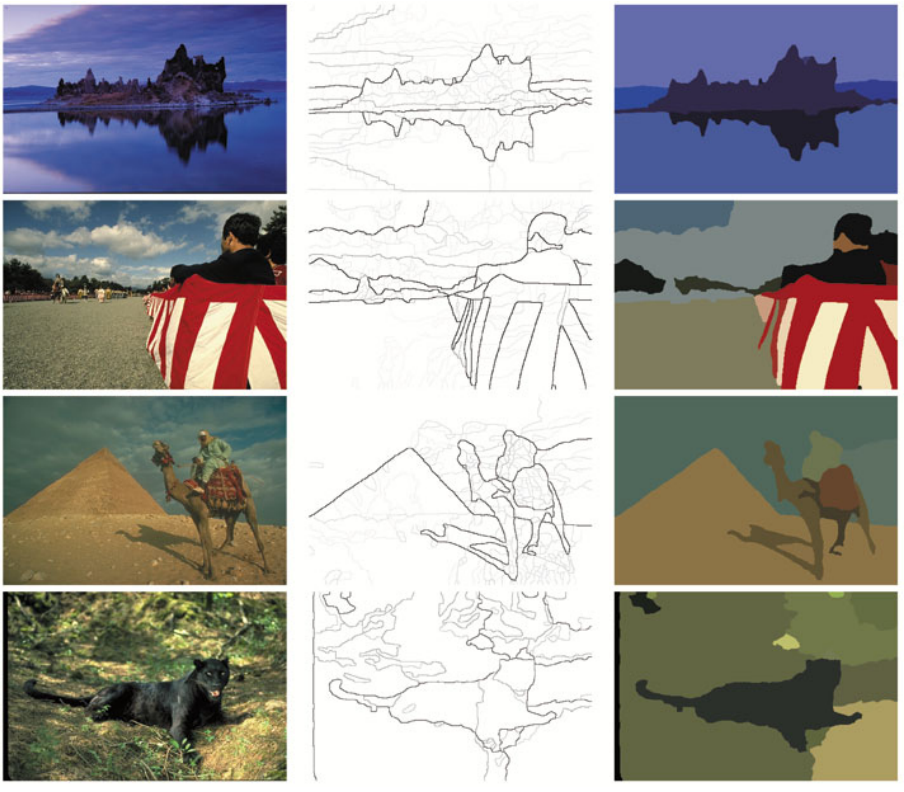


Fig. 5. Object-level image segmentations can be obtained independent of ground truth data. Left column: Original color images; Middle column: Reduced hierarchies represented as Ultrametric Contour Maps; Right column: Object-level segmentations chosen using the last local minimum on graph energy, which are the top-levels of the reduced hierarchies.

visible which in-turn made reasonable visualization possible. The bottom of the same figure shows gPb-owt-ucm-cLGE in terms of how nodes are broken down on two images. It is worth noting that because we only filter hierarchies, the segmentation results will depend on the quality of original hierarchies.

Finally, in Figure 5, we illustrate gPb-owt-ucm-cLGE as Ultrametric Contour Maps and show how a single object-level segmentation can be automatically chosen without the use of a threshold or by appeal to human ground truth data. For instance, [3] relies on ground-truth data to obtain thresholds. Although the threshold corresponding to ODS can be generalized to other images, the best results are obtained by OIS which is image-dependent. To yield a single object-level segmentation for a given image, we simply choose the level of the hierarchy corresponding to the last local minimum on graph energy, that is the top level in our gPb-owt-ucm-cLGE hierarchies. Such segmentations are the coarsest in the hierarchy.

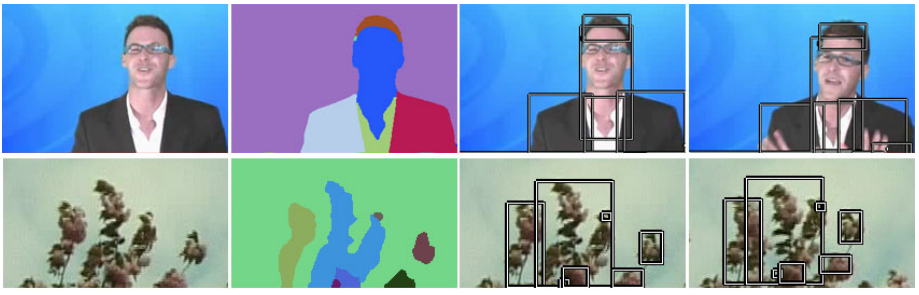


Fig. 6. Two examples of stable tracking using our reduced hierarchy over Berkeley's UCM maps (gPb-owt-ucm); see supplementary material for the full set of videos. Left to right: frame one, a segmentation map, typical frames from the video tracking regions.

With regards to runtime, because we work on regions rather than pixels, our graphs are relatively small in size and sparse in nature. This in turn made Eigenvalue decomposition less of a problem. In practice, our current Matlab implementation takes around 20 seconds per image on a Intel Core2Duo 2.6GHz machine with 4GB of RAM. Code will also be made available on-line[1].

5 Application to Tracking

Here we show the value of reduced hierarchies to tracking, in particular memory and complexity is improved, with marginal gain in accuracy. To describe a video, a hierarchy should be stable across the entire sequence. We evaluate a hierarchy's stability by the temporal stability of its regions. If a region is stable, it changes little over time and can be tracked more easily.

Given a video, we build a hierarchy from the first frame, and track every single region using the standard KLT tracker [13]. Figure 6 shows identifiable objects are tracked over time, these regions have come from a reduced hierarchy and qualitatively demonstrate the regions are semantic. The filtered hierarchy only consumes about one-tenth of the memory (which is the fraction of total regions in the reduced hierarchy compared to the full) and takes about one-tenth of the tracking time. We ran the experiment on several scenes and find the filtered hierarchies keep good regions that can be stably tracked. Stable regions can be seen in both Figure 6 and the supplementary material.

6 Conclusion

In this paper, we have introduced *component-wise Laplacian graph energy*, cLGE, as a complexity measure useful to filter image description hierarchies. cLGE is a measure of graph complexity that is simple to compute. We showed that

[1] http://www.cs.bath.ac.uk/Song

- cLGE operates over two state of the art image hierarchies, which lends support to our claim of algorithmic independence;
- we reduce the number of levels by an order or magnitude with little or no effect on the semantic quality of the result.
- the reduction in data leads to a description that benefits applications, as demonstrated by our tracking example.

We measured the semantic quality of an hierarchy using the widely used Berkeley Segmentation Dataset (BSDS) [15]; apart from the original boundary benchmarks, experiments were also conducted on a new extension on regions. Both experiments show little or no loss of semantic quality of the graph energy filtered hierarchies when compared to the originals.

Despite the good filtering performance of cLGE, the quality of the end result will depend on the quality of the original hierarchies. We have, though, shown that the filtered gPb-owt-ucm hierarchies, largely retain their performance; in addition they provide a solid basis for tracking because they are stable over time, and visualizations are reminiscent of CSTs [2].

Future work includes accessing how graph energy can be used to generate hierarchies of a more semantic fashion, possibly by recursively applying it to individual regions rather than the whole image. The question of whether an overall objective function can be optimized across the layers can also be a future research direction. Applications such as more efficient and accurate object classification and matching are being considered too.

References

1. Ahuja, N.: A transform for multiscale image segmentation by integrated edge and region detection. TPAMI 18(12), 1211–1235 (1996)
2. Ahuja, N., Todorovic, S.: Connected segmentation tree - a joint representation of region layout and hierarchy. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (June 2008)
3. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empitical evaluation. In: Computer Vision and Pattern Recognition (2009)
4. Bangham, J.A., Harvey, R.W., Ling, P.D., Aldridge, R.V.: Morphological scale-space preserving transforms in many dimensions. Journal of Electronic Imaging 5, 283–299 (1996)
5. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. TPAMI 24(5), 603–614 (2002)
6. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 1124–1131 (June 2005)
7. Day, J., So, W.: Singular value inequality and graph energy. Electronic Journal of Linear Algebra 16, 291–299 (2007)
8. Escolano, F., Hancock, E.R., Lozano, M.A.: Polytopal graph complexity, matrix permanentsm, and embedding. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 237–246. Springer, Heidelberg (2008)

9. Fisher, D.: Iterative optimization and simplification of hierarchical clusterings. J. Artif. Int. Res. 4(1), 147–179 (1996)
10. Gutman, I., Zhou, B.: Laplacian energy of a graph. Linear Algebra and its applications 414, 29–37 (2006)
11. Kokkinos, I., Yuille, A.: Hop: Hierarchical object parsing. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 802–809 (June 2009)
12. Liu, J., Yang, Y.: Multiresolution color image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 16, 689–700 (1994)
13. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI 1981), pp. 674–679 (April 1981)
14. Martin, D., Fowlkes, C., Malik, K.: Learning to detect natural image boundaries using local brightness, color and texture cues. TPAMI 26(5), 530–549 (2004)
15. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. 8th Int'l Conf. Computer Vision., vol. 2, pp. 416–423 (2001)
16. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide basline stereo from maximally stable extremal regions. In: BMVC, pp. 384–393 (2002)
17. Maire, M.: Arbeláez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize juncations in natural images. In: CVPR (2008)
18. Paris, S., Durand, F.: A topological approach to hierarchical segmentation using mean shift. In: CVPR, pp. 1–8. IEEE Computer Society Press, Los Alamitos (2007)
19. Ren, X.: Multi-scale improves boundary detection in natural images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 533–545. Springer, Heidelberg (2008)
20. Robles-Kelly, A., Hancock, E.R.: A riemannian approach to graph embedding. Pattern Recognition 40, 1042–1056 (2007)
21. Shi, J., Malik, J.: Normalized cuts and image segmentation. TPAMI, 888–905 (2000)
22. Shokoufandeh, A., Dickinson, S.J., Siddiqi, K., Zucker, S.W.: Indexing using a spectral encoding of topological structure. In: International Conference on Pattern Recognition, pp. 491–497 (1999)
23. Tu, P., Saxena, T., Hartley, R.: Recognizing objects using color-annotated adjacency graphs. In: Shape, Contour and Grouping in Computer Vision, pp. 246–263 (1999)
24. White, D., Wilson, R.: Mixing spectral representations of graphs. In: International Conference on Pattern Recognition (2006)
25. Worthington, P., Hancock, E.: Region-based object recognition using shape-from-shading. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 455–471. Springer, Heidelberg (2000)

Semantic Segmentation of Urban Scenes Using Dense Depth Maps

Chenxi Zhang, Liang Wang, and Ruigang Yang

Center for Visualization and Virtual Environments, University of Kentucky, USA

Abstract. In this paper we present a framework for semantic scene parsing and object recognition based on dense depth maps. Five view-independent 3D features that vary with object class are extracted from dense depth maps at a superpixel level for training a classifier using randomized decision forest technique. Our formulation integrates multiple features in a Markov Random Field (MRF) framework to segment and recognize different object classes in query street scene images. We evaluate our method both quantitatively and qualitatively on the challenging Cambridge-driving Labeled Video Database (CamVid). The result shows that only using dense depth information, we can achieve overall better accurate segmentation and recognition than that from sparse 3D features or appearance, or even the combination of sparse 3D features and appearance, advancing state-of-the-art performance. Furthermore, by aligning 3D dense depth based features into a unified coordinate frame, our algorithm can handle the special case of view changes between training and testing scenarios. Preliminary evaluation in cross training and testing shows promising results.

1 Introduction

Scene parsing, which refers to the process of simultaneously classifying and segmenting objects in an image, is one of the fundamental problems of computer vision. A successful scene parsing system is of great benefit to a variety of vision applications, such as object recognition, automatic driver assistance and 3D urban modeling. In this work, we focus on semantic segmentation of urban scenes from a monocular video sequence filmed at street level and propose an effective algorithm to address this particular problem. The most distinct feature that differentiates our approach from existing solutions lies in the use of dense depth maps recovered via multi-view stereo matching techniques as cues to achieve accurate scene parsing.

While the task of segmentation traditionally relies on color information alone, using depth information has some obvious advantages. Firstly it is invariant to lighting and/or texture variation; secondly it is invariant to camera pose and perspective change. Therefore using depth can potentially enable successful segmentation independent of illumination or view, significantly expanding the range of operation conditions. Recently, advances in structure from motion (SFM) techniques make it easier to obtain depth ques from video sequences. As a result there

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 708–721, 2010.

is notable progress in performing semantic segmentation using 3D cues. A pioneer work in using depth for urban scene segmentation is [9], in which the authors demonstrated that semantic segmentation is possible based solely on *sparse* 3D point clouds obtained from structure from motion techniques. Given the success of [9], a natural question raised is whether *dense* 3D information can perform equally well, or, even better on this challenging task. Our experiments indicate that this is true for street scene segmentation and recognition.

1.1 Related Work

Recently, many efforts have been made to achieve accurate semantic segmentation and classification. Traditional approaches employ 2D appearance information, such as color, texture, shape [10,3,12] and have achieved impressive results. However, a drawback of appearance based features is that they may change dramatically under different imaging conditions. For example, in day time and night, summer and winter, a scene may have different appearances. With recent advances in 3D imaging, 3D structure information has been exploited for semantic segmentation and recognition [9,11]. Specifically, when the input is a video sequence rather than a still image, the available motion based cues contain a large amount of information that can be used for segmentation and recognition. They are invariant to appearance changes. In [3], Liu et al. proposed a novel nonparametric approach for scene parsing using dense scene alignment. Their method is based on the alignment of testing image and its best matching image in the database by SIFT flow. However their formulation requires a large amount of training data of around 2700 fully annotated images [2].

The approaches most related to ours are [9] and [11]. Our method is inspired by the work of [9] with the following distinctions: First, we propose to use *dense (per-pixel)* depth map information for street scene segmentation, while in [9] they proposed to utilize sparse structure from motion point clouds; and in [11] they combined structure from motion and appearance descriptors together. Second, although both of the two previous approaches have achieved impressive results, some of the features, such as height above camera, is defined as the relative height between object and camera, therefore dependent on camera pose. Note there are two types of common camera configuration: front-view camera (as used in [9]) and side-view camera (as used in [11]). Training using one of the datasets and testing on the other is likely to lead to fail in both approaches. In our approach, because we only use per-pixel depth information without any dependence on camera pose or appearance, our segmentation algorithm can be easily formulated in a *view-independent* fashion. Applying our approach we can get satisfactory segmentation results from a video sequence using training data captured under a different configuration from testing data. Finally, while it is shown in [9] that motion-derived information leads to results comparable to existing state-of-the-art appearance based techniques, we demonstrate in this paper that semantic segmentation using only dense depth information outperforms both sparse structure from motion based method and appearance based method, or even the combination of sparse depth with appearance.

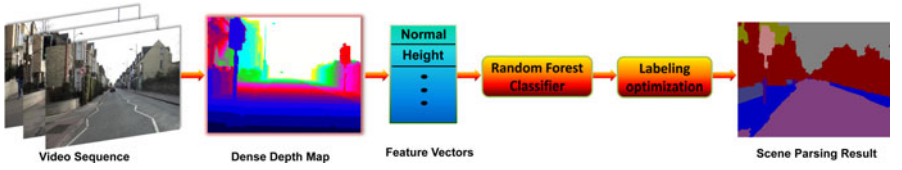


Fig. 1. Overview of our framework

1.2 Overview and Contributions

Figure 1 shows an overview of our scene parsing pipeline. The proposed algorithm starts from generating dense depth maps by plane swiping method. An over-segmentation is applied to video sequences and 3D information is obtained using dense depth maps. Five discriminative 3D features are extracted from dense depth maps and combined together to build a randomized decision forest classifier to obtain accurate semantic scene segmentation. The primary contribution is that we demonstrate a scene parsing algorithm that uses only dense 3D depth information to *outperform the combination of sparse 3D features and appearance*. Moreover, we demonstrate that by transferring those 3D features to a common coordinate system that is independent on camera pose, view-independent semantic scene segmentation can be achieved. Training in one type of camera view and application in a changed camera view is now possible.

In summary, our dense depth based segmentation algorithm lends itself well for real-world applications in which the viewpoints, lightings and object textures are likely to be significantly different from those captured in the training database.

2 Depth Maps Recovery

Stereo reconstruction of dense depth maps from a video sequence has long been a research topic in computer vision. Recently, there have been great advances that are based on high-quality stereo matching algorithms and effective 3D modeling pipelines [19,15]. As depth recovery is not the primary focus of this work, we simply modify existing techniques to compute the scene depth information from video.

Given an input video sequence $\{I^t\}$ captured at street level, we first employ the SFM software released by Zhang et.al. [20] to estimate camera parameters. Then, our stereo matching module takes as input camera poses and the corresponding frames from the video and produces a dense depth map D^t for each frame. The stereo matching pipeline used in our paper consists of a depth initialization step followed by a depth refinement process.

For depth initialization, we model stereo matching as an energy minimization problem. The global energy function contains three terms, i.e., a data term, a smoothness term and a segmentation term. The data term measures how well the depth map agrees with the input images under the color consistency assumption. In this paper, we use the standard plane-sweep approach as described in [4,17] to compute data costs. In our implementation, the plane-sweep stereo

is solved on 17 consecutive images where the middle one is the reference view. The smoothness term incorporates the assumption that the scene is piecewise smooth and penalizes assigning different depth values to neighboring pixels. We use the truncated linear model [6] to define our smoothness cost. In order to better handle textureless areas, we incorporate the segmentation information into our MRF stereo framework as a soft constraint. We first segment each reference frame using mean-shift algorithm [5]. Each color segment is treated as a nonfronto-parallel plane in 3D and a robust plane fitting method [7] is applied to estimate the plane parameters. Similar to [16], the segmentation term is modeled to penalize depth assignment that departs from that given by the pixel's corresponding plane parameter. For each frame I^t, we use belief propagation (BP) [6] to estimate an initial depth map $\widetilde{D^t}$ by minimizing our energy function.

In depth initialization step, we compute the disparity map $\widetilde{D^t}$ for each frame without considering the temporal consistency among depth maps. To address this issue, during the depth refinement step a multi-view fusion algorithm [13] is applied to refine the depth maps returned by BP. Depth maps after refinement are smooth both spatially and temporally and contain less visual artifacts.

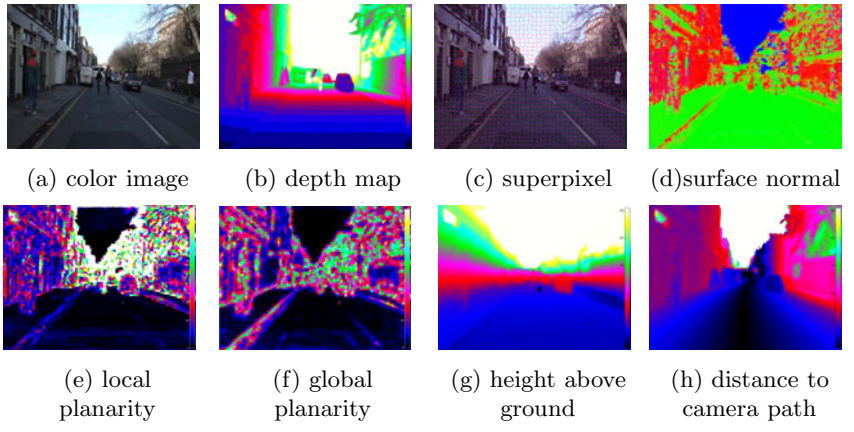
3 Semantic Segmentation from Dense Depth Maps

After recovering dense depth maps from videos, our algorithm starts by over-segment each image into homogeneous pixel clusters, i.e. *superpixels*, then extract feature vectors. These vectors are used for training and classifications. Raw classified outputs are combined via a pairwise Markov Random Field (MRF) for final segmentation. Details are presented below.

3.1 Image Over-Segmentation

Over-segmentation of image into superpixels is a common preprocessing step for image parsing algorithms. We choose to use over-segmentation as one of the preprocessing steps in semantic segmentation due to the following reasons: First, each superpixel in 2D images can be approximately viewed as a patch in 3D. Some features we employed in this work can be well defined over a patch, *e.g.*, surface normal, surface planarity etc. Second, over-segmentation can increase the chances that the boundaries of different object classes are extracted. In this regard, pixel-wise classification may result in less consistent boundaries. Finally, using over-segmentation can reduce the computational complexity of the system, since by counting each superpixel as one sample, the number of total samples are largely reduced as compared to pixel-wise training and testing.

We applied a geometric-flow based algorithm named "TurboPixels" [1] to achieve dense image over-segmentation. This recent technique can produce superpixels with uniform size and shape, maintain connectivity and compactness, and preserve original image edges. The choice for superpixel size is a key issue. On one hand, using large superpixels may bring in the risk of a superpixel spanning across multiple semantic objects. On the other hand, a small superpixel



(a) color image (b) depth map (c) superpixel (d)surface normal

(e) local planarity (f) global planarity (g) height above ground (h) distance to camera path

Fig. 2. Visualization of different steps and features in our algorithm

may contain insufficient points to precisely define a good feature. In our experiment, we set the initial number of superpixel as 6800 for image of size 960×720, roughly 100 pixels per superpixel.

3.2 Features from Dense Depth Map

For each superpixel, we extract five features in 3D space to train our classifier. These features are computed based on the 3D points within each superpixel. All five features are invariant of appearance changes. The five features are surface normal, height above ground, surface local planarity, surface neighboring planarity, and distance to camera path, which are denoted as $F_n^i, F_h^i, F_l^i, F_g^i, F_d^i$ for each superpixel i. Brostow et.al. [9] defined some similar features based on sparse 3D points and projected the features from 3D point clouds to 2D image plane. The way we compute our 3D features is different from theirs, and dense depth maps allow us to estimate these features in a more principled way.

Surface normal (F_n^i): We compute surface normal F_n^i of a 3D patch by fitting a least square plane to the set of 3D points within a superpixel. Similar method has been used in [11].

Height above ground (F_h^i): An object's height information is normally fixed and invariant to change of driving direction, thus can be used as a good feature for classification. [9] used height above camera center as one of the structure features. However, this feature is not invariant to the car on which the camera is mounted, or, large camera position changes. We instead use height above ground as our feature. Our algorithm requires a process to estimate ground plane parameters from depth map. In our implementation we use an iterative RANSAC method to estimate ground plane. At the beginning only 3D points whose normal are close to the up direction in camera coordinate system are used as samples. To avoid including points from sidewalk into plane fitiing, after

each RANSAC procedure we decrease the error threshold T_h used by RANSAC by half and use inliers from previous iteration to fit a new plane. Here T_h is simply a value that controls the point to plane distance. RANSAC will treat a point as an inlier if the distance from the point to plane is smaller than T_h. After a few iterations (6-8 in our implementation) the algorithm terminates and the final plane parameters are treated as ground plane parameters. We find this method works fairly well for our data, where nearly half of the pixels in images are dominated by ground scene. For a superpixel i, height above ground F_h^i is computed as the average distance to ground within the corresponding 3D patch.

Surface local and neighboring planarities: We in work define two types of surface planarity. One is the local planarity of a 3D patch (F_l^i), which corresponds to a superpixel in the image. The other is the neighboring planarity (F_g^i), that is, measuring the variance of a 3D patch orientation with respect to its neighboring patches. The local planarity is computed by using RANSAC based least square plane fitting and calculating the sum of square distances from points to the plane. The neighboring planarity is defined as the average difference of a 3D patch's surface normal with respect to its neighbors' surface normals. These features are useful for splitting planar and non planar objects, for example, building facades and plants.

Distance to camera path(F_d^i): Inspired by [9], we can take advantage of the distance to camera path to separate objects which are horizontally distanced from the camera. We compute the minimum distance from the centroid of the 3D patch to the camera path. The camera path is estimated by fitting a quadratic curve to the camera trajectory. This approach is more accurate and robust for computing objects' distances to camera path, and works well on various scenarios.

3.3 Randomized Decision Forest

Randomized decision forest is a well-known machine learning technique that has been employed in many computer vision tasks [14]. We use this technique to train our classifier based on features derived from the depth maps. In our implementation, we choose the proportion of training data used at each split node to be 0.66. A total number of 80 random decision trees are built for training. We experimentally find these parameters work well for achieving satisfactory recognition rate. We also apply the idea in [10] to balance the number of classes used for training, thus achieve a better class average performance.

3.4 Graph-Cut Based Optimization

We construct a pairwise Markov Random Field (MRF) for each image I by building a graph $G = \langle V, E \rangle$, where each node $v_i \in V$ in the graph represents a superpixel and each edge $e_{ij} \in E$ denotes the neighboring relationship between superpixels. The labeling problem is equal to assign a label $l_i \in L$ to each node

$v_i \in V$. The optimal assignment L_{assign} can be achieved by minimizing the energy:

$$E(L_{assign}) = \sum_{v_i \in V} \psi_i(l_i) + \lambda \sum_{e_{ij} \in E} \phi_{ij}(l_i, l_j) \tag{1}$$

Data term $\psi_i(l_i)$ and smoothness term $\phi_{ij}(l_i, l_j)$ are defined in the following paragraphs. They are computed from the feature responses and the randomized decision forest based classifier. After the costs are computed, a graph-cut optimization [18] is applied to obtain the global optimal labeling configuration.

Data term. The feature responses $F_n^i, F_h^i, F_l^i, F_g^i, F_d^i$ of each superpixel i are collected and applied rank normalization before passed to randomized decision forest for training. Specifically, given the samples for a feature F for all the superpixels as $F^1, F^2, ..., F^n$, where n is the number of superpixels, we first find the low-to-high order statistics $F^{(1)}, F^{(2)}, ..., F^{(n)}$ and then replace each image's feature value by its corresponding normalized rank, as:

$$\widetilde{F^i} = \frac{rank(x_i) - 1}{n - 1} \tag{2}$$

where F^i is the feature value for the i'th sample. The procedure uniformly maps all the features to the range of $[0, 1]$. When there are multiple samples with the same feature value, they are assigned the average rank of that value. We applied this data processing approach based on the fact that there are some scale factors between the features' measurements of different scenes. In addition, normalized rank is effective to compensate some inaccuracies induced by depth map generation and 3D features computation. After rank normalization, the feature responses of each superpixel are passed to randomized decision forest with corresponding ground-truth labels to build the classifier. When testing, the feature responses of each testing sample are passed to the classifier and a posterior probability distribution $P_i(l_i|F_n^i, F_h^i, F_l^i, F_g^i, F_d^i)$ which represents the probabilities the testing sample belongs to each category $l_i \in L$ is returned. We define the data term in MRF as :

$$\psi_i(l_i) = -logP_i(l_i|F_n^i, F_h^i, F_l^i, F_g^i, F_d^i) \tag{3}$$

Smoothness term. For a superpixel v_i and each of its neighbor superpixel v_j, the smoothness cost is defined as:

$$\psi_{ij}(l_i, l_j) = [l_i \neq l_j] \cdot \frac{1}{\delta \|c_i - c_j\|_2 + 1} \tag{4}$$

where $\|c_i - c_j\|_2$ is the L2-Norm of RGB color difference between neighbor superpixels. The penalty is inversely proportional to the color difference between neighbor superpixels. The more similar the colors of two neighbor superpixels are, the less likely they belong to different categories. In our experiment, the value of λ is set to be 1.9 and δ to be 0.1.

3.5 Temporal Multi-view Fusion

The temporally redundant information in video can be used to enhance the accuracy of scene parsing. In Xiao et al.'s work [11], they utilized multi-view information by defining a Markov Random Field for the entire sequence and imposing smoothness terms on superpixels in different images. The way we take advantage of multi-view segmentation consistency is different from theirs. The output of randomized decision forest classifier is a posterior probability distribution which represents the probabilities that a certain testing superpixel belongs to each class. Although we prefer to deem each superpixel belonging to one class, there are some thin structures, such as a column pole, which are far from filling the whole superpixel. In this case, a pixel-wise refinement is needed to achieve more accurate results. Moreover, for those superpixels which are misclassified, temporal fusion is able to increase the chance of making refinement based on neighbor frames' classification results. The fused probability distribution of each pixel can be represented as: $p_r(c) = \sum_{j \in N} w_j p_j(c), c = 1, 2, ..., k$ where $p_r(c)$ represents the probability of a pixel in reference view belonging to class c, $p_j(c)$ is the probability of correspondence pixel in neighbor view j belonging to class c. N represents the set of neighbor frames and k is the number of classes. The fused probability distribution is computed as weighted average of neighbor frames' probability distributions, where weights w_j are determined by how far the neighbor frame j is from reference view r. We define w_j as a Gaussian function:

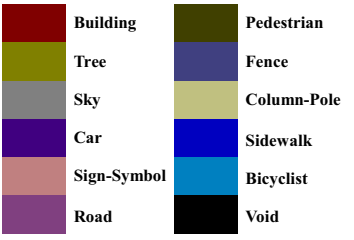
$$w_j = \exp\left(-\frac{(j-r)^2}{\sigma}\right) \tag{5}$$

where σ is set to 10 in our implementation.

After pixel-wise temporal fusion, we applied a superpixel level refinement by aggregating the refined probability distributions of all pixels within a superpixel. In this way we incorporate segmentation consistency across adjacent views.

3.6 Cross Training and Testing

We propose the idea of cross training and testing based on the fact that using per-pixel depth information is independent on camera pose and appearance. In our five 3D features, only surface normal is dependent on camera pose because we compute each 3D patch(a superpixel in 2D)'s normal in camera coordinate system. For cross view training and testing, surface normals in training and testing datasets should be transformed to a common coordinate system. We define the common coordinate system to be as the following: taking the surface normal of ground and camera moving direction(vertical to ground surface normal) as y axis and z axis, x axis can be obtained by the cross product of them. Note that since we only need to transform surface normals, the origin of the coordinate system does not matter. Any normal calculated in the camera coordinate can be rotated to the common coordinate frame, enabling cross training and testing.



Sequence	# of Frames	Dataset
0016E5	204	Day Train
0006R0	101	Day Train
0001TP1	62	Dusk Train
Seq05VD	171	Day Test
0001TP2	62	Dusk Test

Fig. 3. (a)Labeled colors for 11 object classes. (b)Split of sequence as training or testing data.

4 Experiments

We use the challenging CamVid database [8] to evaluate our algorithm's performance. The database includes four high quality video sequences at 30 fps with total duration about 10 minutes. The labeled ground truth images are extracted from the four original video sequences at a rate of 1 fps, with a part of one of the sequences at 15 fps. The image resolution is 960 × 720. Camera extrinsic and intrinsic parameters are also provided in the database. 32 semantic object classes are defined which include fixed objects, types of road surface, moving objects (such as vehicles and people) and ceilings (such as sky, tunnel, archway). Same as in [9], we use 11 dominant categories: Building, Tree, Sky, Car, Sign-Symbol, Road, Pedestrian, Fence, Column-Pole, Sidewalk and Bicyclist. Labeled colors for each object class are shown in Figure 3(a). A quantitative comparison to the state of art is provided. In addition, in order to show our algorithm's compatibility of view-independent training and testing, we carried out another test on images captured by a side-view camera provided by Google. Even using our classifier trained by the CamVid database which is mostly composed of front-view video sequences, we still get decent classification results.

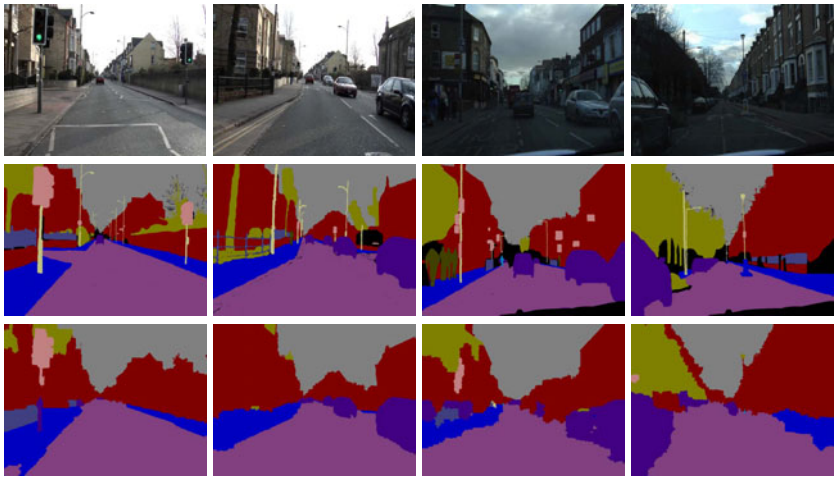
4.1 Evaluation Using the CamVid Database

For comparison with the results from [9], we split the training labeled frames into two groups in the same way as in [9], shown in Figure 3(b). Two groups of the labeled training data, 0016E5 and 0006R0, are used for day sequence training data, and another group 0005VD are used for day sequence testing. The first half of the dusk sequence (0001TP) are used for training, and the second half are used for testing.

We train our classifier using randomized decision forest based on the five dense depth based 3D features and carry out the same set of experiments as in [9]. Table 1 shows the quantitative testing result. In terms of global accuracy(i.e. pixel-wise percentage accuracy), we achieve 82.1% in comparison to 69.1% of combined structure from motion and appearance based approach and 61.8% of solely structure from motion based approach in [9]. Our algorithm also

Table 1. Comparison of Pixel-wise percentage accuracy with [9]. Our dense depth map based approach gives best result on 7 classes. 'Global' is the percentage of pixels correctly classified. 'Average' is the average value of per-class accuracies.

Alg	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Sidewalk	Bicyclist	Average	Global
Motion [9]	43.9	46.2	79.5	44.6	19.5	82.5	24.4	**58.5**	0.1	**61.8**	18	43.6	61.8
Appearance [9]	38.7	60.7	90.1	**71.1**	**51.4**	88.6	**54.6**	40.1	1.1	55.5	23.6	52.3	66.5
Combined [9]	46.2	**61.9**	89.7	68.6	42.9	89.5	53.6	46.6	0.7	60.5	22.5	53.0	69.1
Depth	**85.3**	57.3	**95.4**	69.2	46.5	**98.5**	23.8	44.3	**22.0**	38.1	**28.7**	**55.4**	**82.1**



(Seq05VD #390) (Seq05VD #270) (0001TP #525) (0001TP #1770)

Fig. 4. Scene Parsing result samples. From top to bottom: test image, ground truth, dense depth map inferred segmentation. Note that our algorithm solely using dense depth map is able to achieve accurate segmentation and recognition of street scene.

performs well on most of per-class accuracies and outperforms combined motion and appearance based approach in 7 classes out of 11 classes, which are Building, Sky,Car, Sign-symbol, Road,Column-Pole and Bicyclist. There are two categories (Sidewalk, Pedestrian) that our approach's performances are not as good as in [9]. Both can be attributed to the lack of high-quality depth maps, in which the depth difference between the ground and sidewalk is small. Pedestrian is moving so its depth map is usually wrong. In addition, there are not sufficient samples for the Pedestrian class. Applying superpixel based training on such small classes may face the problem of insufficient samples compared with pixel-wise training approach. Figure 4 shows the qualitative results achieved by our approach.

Table 2. Comparison of pixel-wise percentage accuracy with [9] in illumination variation test

Algorithm	Day Train - Dusk Test	Dusk Train - Day Test
Mot&Struct	45.5%	59.4%
Appearance	21.7%	50.5%
Depth	**63.4**%	**69.2**%

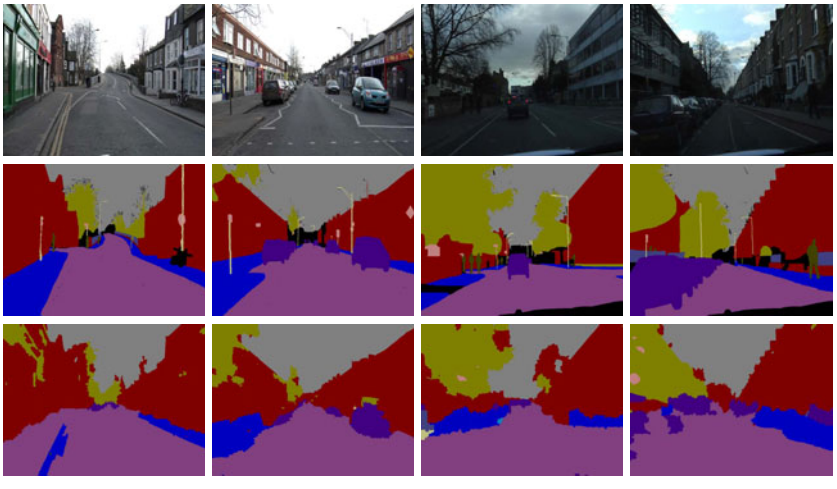


Fig. 5. Robustness to lighting condition changes. In the left figure, from top to bottom: test image, ground truth, segmentation results. The left two columns are examples of training by a dusk sequence and testing on day sequence. The right two columns show results when the training and the test sequence are swapped. Table 2 shows the overall accuracy and comparison to numbers reported in [9].

Testing on illumination variation. One expected advantage of our approach is its independence of appearance or illumination changes. We carried out a test training in one lighting condition(day/dusk) and testing on the other(dusk/day). [9] also carried out the same test using sparse 3D features and appearance features. We compare our results with theirs in Table 2 and Figure 5. As expected, our method has significantly improved the global classification accuracy. The improvement from dusk-training-day-testing is less than that from day-training-dusk-testing, probably due to the poor depth maps generated at dusk. Using active sensors, such as LiDAR scanners can alleviate this problem.

Multi-View Temporal Fusion Test. We test our multi-view temporal fusion algorithm using test sequence Seq05VD. A neighborhood of 40 frames of the reference frame are used for fusion. Some results are shown in Figure 6. It can be seen that fusion improves the overall consistency of the segmentation, which is particularly pronounced when viewing the sequence as a video.

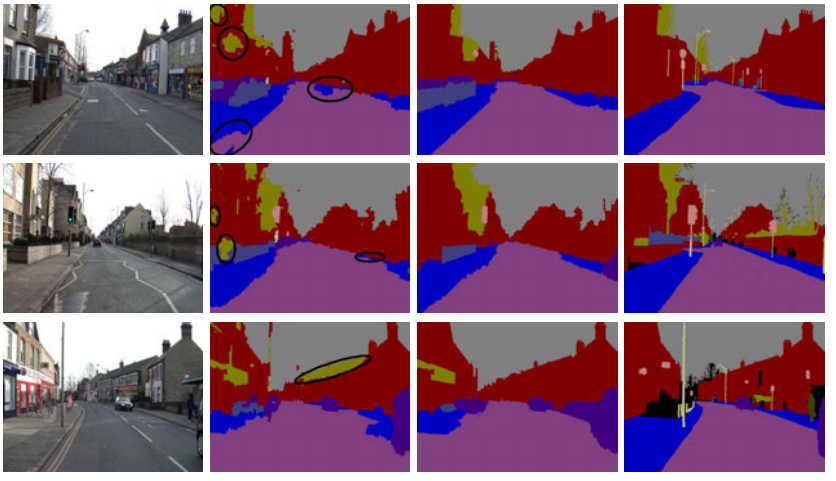


Fig. 6. Refinement of segmentation by multi-view fusion. From left to right: test image, result before fusion, result after fusion, ground truth. Based on temporal consistency of video sequence, some inaccurate classifications in reference view are refined by multi-view fusion, as circled in black.

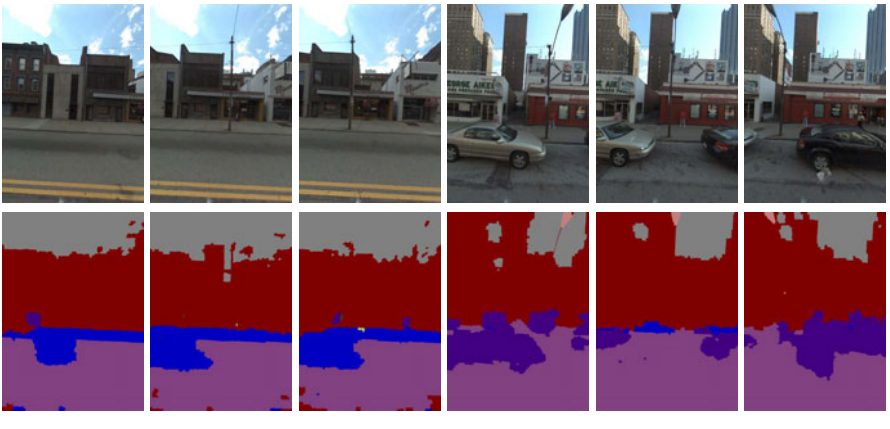


Fig. 7. Preliminary result of changed-view training and testing. From the side-view video sequence,we can see that our algorithm trained based on front-view sequences still obtained acceptable results.

Quantitatively, however, we found that the overall accuracy just improved by 1%. Further investigation in this topic is needed.

4.2 Cross Training Test

As mentioned above, we use per-pixel depth information which is independent of camera pose. It is reasonable to carry out a crossing training and testing

experiment between two types of camera configuration: front-view(used in [9]) and side-view cameras (used in [11]). We use CamVid(front-view) database as training images and Google Street View images(side-view) for testing. All the 3D features in both training and testing database are transferred to the unified coordinate system. In practice, it is necessary to apply a scale change to the testing data due to the scale ambiguity in structure from motion. We use a person's height as the reference. Our results are shown in Figure 7. It is visually comparable to the results from [11] in which both training and testing are performed with Google Street side-view images. We currently do not have access to ground truth labeled dataset, so a quantitative comparison is not available.

5 Conclusion

In this paper, we presented a novel framework for semantic segmentation and recognition based solely on dense depth maps. Our main contribution is that we have shown that dense depth maps generated by stereo contain plenty 3D information for scene parsing and can outperform segmentation using sparse 3D features or appearance features, or even the combination of both. Our method does not rely on any appearance information, making it robust against lighting changes. In addition our full 3D metric representation is independent of camera configurations, therefore we can use one set of front-view or side-view video for training and the other set for testing. This is not possible using appearance information dependent on camera poses and illuminations.

The accuracy of our approach depends on the quality of depth map. While we have applied state-of-the-art algorithms to calculate depth map, the quality of depth map is still quite fragile. Our next step is to apply laser range scan data that have significantly better accuracy and consistency. We expect to see large performance improvement with better input data. In addition, one future area of work is the bias against less frequently appearing objects, such as thin column poles. This is mainly due to lack of sufficient training examples, which naturally lead to a less statistically significant labeling for objects in these classes. One possibility to preserve the classification of less frequent object classes could be to include context information that may boost the significance of objects in certain cases.

Acknowledgements. The authors thank the anonymous reviewers and area chair for their constructive feedbacks. Thanks to University of Cambridge and Google for providing street view datasets. This project is sponsored in part by NSF grant HCC-0448185, CPA-0811647, and CNS-0923131.

References

1. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J.: Turbopixels: Fast superpixels using geometric flows. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(12), 2290–2297 (2009)
2. Russell, B.C., Torralba, A.: Labelme: a database and web-based tool for image. Int. J. of Computer Vision 77(1)

3. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2009)
4. Collins, R.T.: A space-sweep approach to true multi-image matching. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 358–365 (1996)
5. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 25(5), 603–619 (2002)
6. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Int. J. of Computer Vision 70(1) (October 2006)
7. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
8. Brostow, G., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition letters 20(2), 88–97 (2009)
9. Browtow, G., Shotton, J., Fauqueur, J., Cipolla, R.: Segmentation and recognition using structure from motion point clouds. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 44–57. Springer, Heidelberg (2008)
10. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2008)
11. Xiao, J., Quan, L.: Multiple view semantic segmentation for street view images. In: Proc. of Intl. Conf. on Computer Vision (2009)
12. Li, L., Li, F.: What, where and who? classifying events by scene and object recognition. In: Proc. of Intl. Conf. on Computer Vision (2007)
13. Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.M., Yang, R., Nister, D., Pollefeys, M.: Real-time visibility-based fusion of depth maps. In: Proc. of Intl. Conf. on Computer Vision (2007)
14. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning 36(1), 3–42 (2006)
15. Pollefeys, M., Nister, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. Int. J. of Computer Vision 78(2), 143–167 (2008)
16. Sun, J., Li, Y., Kang, S.B., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2005)
17. Yang, R., Pollefeys, M.: Multi-resolution real-time stereo on commodity graphics hardware. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2003)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)
19. Zhang, G., Jia, J., Wang, T.T., Bao, H.: Recovering consistent video depth maps via bundle optimization. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2008)
20. Zhang, G., Qin, X., Hua, W., Wang, T.T., Heng, P.A., Bao, H.: Robust metric reconstruction from challenging video sequences. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (2007), http://www.zjucvg.net/acts/acts.html

Tensor Sparse Coding for Region Covariances

Ravishankar Sivalingam, Daniel Boley,
Vassilios Morellas, and Nikolaos Papanikolopoulos

Department of Computer Science & Engineering,
University of Minnesota, Minneapolis, MN 55455, USA
{ravi,boley,morellas,npapas}@cs.umn.edu

Abstract. Sparse representation of signals has been the focus of much research in the recent years. A vast majority of existing algorithms deal with vectors, and higher–order data like images are usually vectorized before processing. However, the structure of the data may be lost in the process, leading to poor representation and overall performance degradation. In this paper we propose a novel approach for sparse representation of positive definite matrices, where vectorization would have destroyed the inherent structure of the data. The sparse decomposition of a positive definite matrix is formulated as a convex optimization problem, which falls under the category of determinant maximization (MAXDET) problems [1], for which efficient interior point algorithms exist. Experimental results are shown with simulated examples as well as in real–world computer vision applications, demonstrating the suitability of the new model. This forms the first step toward extending the cornucopia of sparsity-based algorithms to positive definite matrices.

Keywords: Positive definite matrices, region covariances, sparse coding, MAXDET optimization.

1 Introduction

In the past decade there has been a deluge of research on sparse representations of signals [2,3,4] and recovery of such sparse signals from noisy and/or undersampled observations [5,6]. Much of the work has been associated with vector-valued data, and higher–order signals like images (2-D, 3-D, or higher) have been dealt with primarily by vectorizing them and applying the aforementioned vector methods. See [7] for a review of a few examples of sparse representation in computer vision and pattern recognition applications. However, more recently some researchers have realized the advantages of maintaining the higher–order data in their original form [8] to preserve some inherent ordering, which may be lost upon vectorization.

One such data type consists of $n \times n$ symmetric positive semi-definite matrices ($\mathbf{S}_+^n$). The kernel matrix in many popular 'kernelized' machine learning algorithms [9] belongs to this class. In medical imaging, the revolutionary new field of Diffusion Tensor Imaging (DTI) represents each voxel in a 3-D brain scan as a 3×3 positive definite matrix, called the diffusion tensor, whose principal

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 722–735, 2010.

eigenvector gives the direction of water diffusion in that region. More recently in the image processing and computer vision community, a new feature known as the region covariance descriptor has emerged [10,11], which represents an image region by the covariance of n-dimensional feature vectors at each pixel in that region. This is currently being used in conjunction with machine learning algorithms for human detection and tracking, object recognition, texture classification, query-based retrieval of image regions, and much more [12].

In this paper we propose a novel approach for sparse representation of positive definite matrices, named *tensor sparse coding*[1]. The sparse decomposition of a positive definite matrix in terms of a given dictionary is formulated as a convex optimization problem, which belongs to the class of MAXDET problems [1] and for which efficient interior point methods are available. We believe that this extension of sparse coding techniques to the space of positive definite matrices will benefit the development of sparsity-related algorithms tailored to these problem domains as well. This forms the first step toward extending the cornucopia of sparsity-based algorithms to this new class of data points, and all algorithms that primarily use the sparse coding stage follow readily from our approach.

The rest of the paper is organized as follows: In the remainder of this section, we provide a brief description about region covariances, and related work on these descriptors. Section 2 describes the problem statement, and our tensor sparse coding approach is explained in Sect. 3. Experiments on both synthetic and actual datasets are shown in Sect. 4, wrapping up with our conclusions and future research directions in Sect. 5.

1.1 Region Covariance Descriptors

Region covariances were introduced by Tuzel et al. [10] as a novel region descriptor for object detection and classification. Given an image $\mathcal{I}$, let ϕ define a mapping function that extracts an n-dimensional feature vector z_i from each pixel $i \in \mathcal{I}$, such that

$$\phi(I, x_i, y_i) = z_i \ , \tag{1}$$

where $z_i \in \mathbf{R}^n$, and (x_i, y_i) is the location of the i^{th} pixel. A given image region R is represented by the $n \times n$ covariance matrix C_R of the feature vectors $\{z_i\}_{i=1}^{|R|}$ of the pixels in region R. Thus the region covariance descriptor is given by

$$C_R = \frac{1}{|R| - 1} \sum_{i=1}^{|R|} (z_i - \mu_R)(z_i - \mu_R)^T \ , \tag{2}$$

where, μ_R is the mean vector,

$$\mu_R = \frac{1}{|R|} \sum_{i=1}^{|R|} z_i \ . \tag{3}$$

[1] From the 'tensor' in 'diffusion tensor' [13].

The feature vector z usually consists of color information (in some preferred color–space, usually RGB) and information about the first and higher order spatial derivatives of the image intensity, depending on the application intended.

Although covariance matrices can be positive semi–definite in general, the covariance descriptors themselves are regularized by adding a small constant multiple of the identity matrix, making them strictly positive definite. Thus, the region covariance descriptors belong to $\mathbf{S}^n_{++}$, the space of $n \times n$ positive definite matrices which forms a connected Riemannian manifold. Given two covariance matrices C_i and C_j, the Riemannian distance metric $d_{\text{geo}}(C_i, C_j)$ gives the length of the geodesic connecting these two points on this manifold. This is given by [13],

$$d_{\text{geo}}(C_i, C_j) = \left\| \log \left(C_i^{-1/2} C_j C_i^{-1/2} \right) \right\|_F \quad , \tag{4}$$

where $\log(\cdot)$ represents the matrix logarithm and $\| \cdot \|_F$ is the Frobenius norm. Many existing classification algorithms for region covariances use the geodesic distance in a K-nearest-neighbor framework. The geodesic distance can also be used with a modified K-means algorithm for clustering.

Methods for fast computation of region covariances using *integral images* [11] enable the use of these compact features for many practical applications that demand real–time performance. For texture characterization, spatial derivatives are suitable features [10], whereas for face recognition, region covariances are constructed from outputs of a bank of Gabor filters [14]. Hu et al. [15] use covariance descriptors for probabilistic tracking using particle filtering. Palaio and Batista [16] also perform multi–object tracking using region covariances and particle filters. In [17], Paisitkriangkrai et al. boost the covariance features to improve the classification accuracy. In [12], Tuzel et al. use LogitBoost on the covariance descriptors for pedestrian detection. Sivalingam et al. [18] learn a modified distance metric over the manifold from pairwise constraints, for semi-supervised clustering.

2 Problem Statement

We begin with a known dictionary consisting of k $n \times n$ positive definite matrices $\mathcal{A} = \{A_i\}_{i=1}^k$, where each $A_i \in \mathbf{S}^n_{++}$ is referred to as a dictionary atom. Given a positive definite matrix S, our goal is to represent the new matrix as a linear combination of the dictionary atoms, *i.e.*,

$$S = x_1 A_1 + x_2 A_2 + \ldots + x_k A_k = \sum_{i=1}^{k} x_i A_i \quad , \tag{5}$$

where $x = (x_1, x_2, \ldots, x_k)^T$ is the vector of coefficients.

Since only a non-negative linear combination of positive definite matrices is guaranteed to yield a positive definite matrix, we impose a non-negativity constraint on the coefficient vector x, $x \in \mathbf{R}^k_+$.

It is to be noted that the given matrix S need not always be exactly representable as a sparse non-negative linear combination of the dictionary atoms. Hence, we will aim to find the best approximation $\hat{S}$ to S, by minimizing the residual approximation error in some sense. Clearly, we require the approximation $\hat{S}$ to be positive definite,

$$\hat{S} \succeq 0 \Longrightarrow x_1 A_1 + x_2 A_2 + \ldots + x_k A_k \succeq 0 \ . \tag{6}$$

Although this would be ensured by construction, due to the non-negativity of x and the strictly positive definite dictionary atoms, we leave this constraint in for reasons explained later in the discussion.

We further require that the representation be sparse, *i.e.*, S is to be represented by a sparse linear combination of the dictionary atoms. To this effect, we impose a constraint on the ℓ_0 "pseudo-norm" of x,

$$\|x\|_0 \leq T \ , \tag{7}$$

where T is a pre-defined parameter, denoting the maximum number of non-zero elements of x.

3 Approach

3.1 The LogDet Divergence

If X^{-1} and Y^{-1} are the covariance matrices of two multivariate Gaussians P_X and P_Y with the same (or zero) mean, then the KL-divergence between the two distributions [19] is given by,

$$KL\left(P_Y \| P_X\right) = \frac{1}{2} D_{\mathrm{ld}}(Y^{-1}, X^{-1}) = \frac{1}{2} D_{\mathrm{ld}}(X, Y) \ , \tag{8}$$

where $D_{\mathrm{ld}}(\cdot)$ is the LogDet (or Burg matrix) divergence [20], given by,

$$D_{\mathrm{ld}}(X, Y) = tr\left(XY^{-1}\right) - \log\det\left(XY^{-1}\right) - n \ . \tag{9}$$

Here n is the dimension of the matrices X and Y, and $tr(\cdot)$ denotes the trace of the matrix. Note that, in general, the divergence is asymmetric, *i.e.,* $D_{\mathrm{ld}}(X, Y) \neq D_{\mathrm{ld}}(Y, X)$.

Further, there exists a bijection between regular exponential families and a large class of Bregman divergences, called regular Bregman divergences [21]. For example, the squared-error loss function which is minimized in vector sparse coding methods comes from the squared Euclidean distance, which is the Bregman divergence corresponding to the multivariate Gaussian distribution. Thus, the minimization of a squared error objective function corresponds to the assumption of Gaussian noise. The Wishart distribution [22], which is a distribution over $n \times n$ positive definite matrices, with positive definite parameter matrix Θ and degrees of freedom $p \geq n$, is given by

$$\Pr(X|\Theta, p) = \frac{|X|^{(p-n-1)/2} \exp\left(-\frac{1}{2} tr(\Theta^{-1} X)\right)}{2^{pn/2} |\Theta|^{p/2} \Gamma_n(p/2)} \ , \tag{10}$$

where $|\cdot|$ is the determinant. The Bregman divergence corresponding to the Wishart distribution is the LogDet divergence $D_{\mathrm{ld}}(X, \Theta)$ [23]. If we assume that the positive definite matrix is drawn from a Wishart distribution, we can minimize the LogDet divergence between the matrix and its approximation. Further, the LogDet divergence (9) is convex in X (but not in Y) and hence is a perfect candidate for our problem formulation.

3.2 Formulation

Motivated by the above-mentioned reasons, we define our optimization problem as one which tries to minimize the LogDet divergence $D_{\mathrm{ld}}(\hat{S}, S)$ between the approximation $\hat{S}$ and the given matrix S.

$$D_{\mathrm{ld}}(\hat{S}, S) = tr\left(\left(\sum_{i=1}^{k} x_i A_i\right) S^{-1}\right) - \log\det\left(\left(\sum_{i=1}^{k} x_i A_i\right) S^{-1}\right) - n \ . \quad (11)$$

For numerical stability, we ensure that the arguments are also symmetric. Since the trace and the log det are invariant under a similarity transformation, we map $X \mapsto S^{-1/2} X S^{1/2}$, where X is the argument.

$$\begin{aligned} D_{\mathrm{ld}}(\hat{S}, S) &= tr\left(S^{-1/2}\left(\sum_{i=1}^{k} x_i A_i\right) S^{-1/2}\right) \\ &\quad - \log\det\left(S^{-1/2}\left(\sum_{i=1}^{k} x_i A_i\right) S^{-1/2}\right) - n \qquad (12) \\ &= tr\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) - \log\det\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) - n \ , \qquad (13) \end{aligned}$$

where $\hat{A}_i = S^{-1/2} A_i S^{-1/2}$. Therefore,

$$D_{\mathrm{ld}}(\hat{S}, S) = \sum_{i=1}^{k} x_i \ tr\hat{A}_i - \log\det\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) - n \ . \quad (14)$$

We discard n from the objective function as it is a constant.

The best approximation $\hat{S}$ would result in an exactly positive semidefinite residual $E = S - \hat{S}$, so that incrementing any x_i is not possible without pushing the residual to be indefinite, *i.e.*, leading to $\hat{S} \not\succeq S$, since subtracting even the "smallest" positive definite matrix from a positive semidefinite matrix will make it indefinite. Therefore, the minimum eigenvalue of the residual $\lambda_{\min}(S - \hat{S})$ should be as close to zero as possible. Hence we impose the constraint

$$\hat{S} \preceq S \quad \text{or} \quad x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \preceq I_n \ , \quad (15)$$

where I_n is the $n \times n$ identity matrix. Combining with (6), we get

$$0 \preceq x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \preceq I_n \ . \quad (16)$$

Since the constraint (7) is non-convex, a convex relaxation of this constraint involves minimizing the ℓ_1 norm of x instead of the ℓ_0 pseudo-norm. Under certain assumptions [24], the ℓ_1 penalty has been proven to yield the same (or similar) results as minimizing $\|x\|_0$ for sparse decompositions.

Combining all the above constraints with the objective function we wish to minimize, we have the following optimization problem:

$$\min_x \quad \sum_{i=1}^{k} x_i \, tr\hat{A}_i - \log\det\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) + \lambda \|x\|_1 \tag{17}$$

$$\text{s.t.} \quad x \geq 0 \tag{18}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \succeq 0 \tag{19}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \preceq I_n \ , \tag{20}$$

where $\lambda \geq 0$ is a parameter which represents a trade–off between a sparser representation and a closer approximation. Further, since the x_i's are non–negative, the ℓ_1 norm simply becomes the sum of the components of x, *i.e.*,

$$\|x\|_1 = \sum_{i=1}^{k} x_i \ , \tag{21}$$

yielding the optimization problem :

$$\min_x \quad \sum_{i=1}^{k} x_i \left(tr\hat{A}_i + \lambda\right) - \log\det\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) \tag{22}$$

$$\text{s.t.} \quad x \geq 0 \tag{23}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \succeq 0 \tag{24}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \preceq I_n \ . \tag{25}$$

Concurrent with other vector sparse coding techniques, we may express this optimization problem in an alternate form which puts a different form of constraint on the sparsity of x. Instead of a penalty term $\lambda\|x\|_1$ in the objective function, we may enforce the sparsity by adding the constraint $\|x\|_1 \leq T$ resulting in the following variation of the above problem:

$$\min_x \quad \sum_{i=1}^{k} x_i tr\hat{A}_i - \log\det\left(\sum_{i=1}^{k} x_i \hat{A}_i\right) \tag{26}$$

$$\text{s.t.} \quad x \geq 0 \tag{27}$$

$$\sum_{i=1}^{k} x_i \leq T \tag{28}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \succeq 0 \tag{29}$$

$$x_1\hat{A}_1 + x_2\hat{A}_2 + \ldots + x_k\hat{A}_k \preceq I_n \ . \tag{30}$$

We denote the optimization problem defined by (22)–(25) as Type I, and that defined by (26)–(30) as Type II.

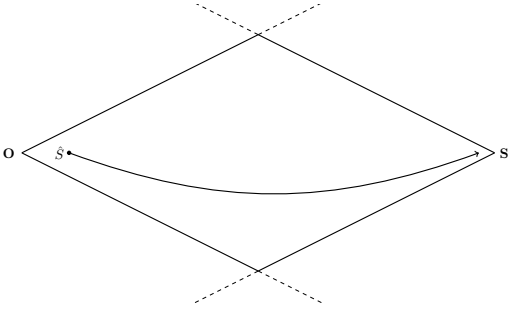


Fig. 1. The feasible set consists of the region of intersection of two positive semidefinite cones, one centered at the origin O, and the other an inverted cone centered at S. $\hat{S}$ lies in the strict interior of this cone, and is pushed towards S by the log det term in the objective. The linear term serves as a regularizer on the coefficients x_i.

3.3 The MAXDET Problem

The above formulations of tensor sparse coding fall under a general class of optimization problems known as determinant maximization, or MAXDET, problems [1], of which semi-definite programming (SDP) and linear programming (LP) are special cases. The MAXDET problem is defined as:

$$\min_{x} \quad c^T x + \log\det G(x)^{-1} \tag{31}$$

$$\text{s.t.} \quad G(x) \triangleq G_0 + x_1 G_1 + \ldots + x_k G_k \succ 0 \tag{32}$$

$$F(x) \triangleq F_0 + x_1 F_1 + \ldots + x_k F_k \succeq 0 \ , \tag{33}$$

where $x \in \mathbf{R}^k$, $G_i \in \mathbf{S}^n$ and $F_i \in \mathbf{S}^N$. These problems are convex, well-behaved, and efficient interior point methods exist for solving them. Note that the $G(x)$ inside the log det term also explicitly appears as a constraint in the standard form of the MAXDET problem, leading to our inclusion of the same in our formulation.

Thus, we have formulated two variations of our tensor sparse coding problem (Type I and II), both of which are convex and of the standard MAXDET form. The approximation $\hat{S}$ lies inside the intersection of the two positive semidefinite cones, one centered at the origin and the inverted positive semidefinite cone centered at S, which forms a closed convex set (See Fig. 1). The $-\log\det$ term in the objective function pushes the approximation $\hat{S}$ toward S, motivating a better approximation. We use `CVX` [25] to solve the MAXDET optimization problem.

4 Experiments

4.1 Numerical Example

Our first set of experiments were run on a synthetic data set, comprised of precision (inverse of covariance) matrices. We start with an $n \times n$ covariance matrix C

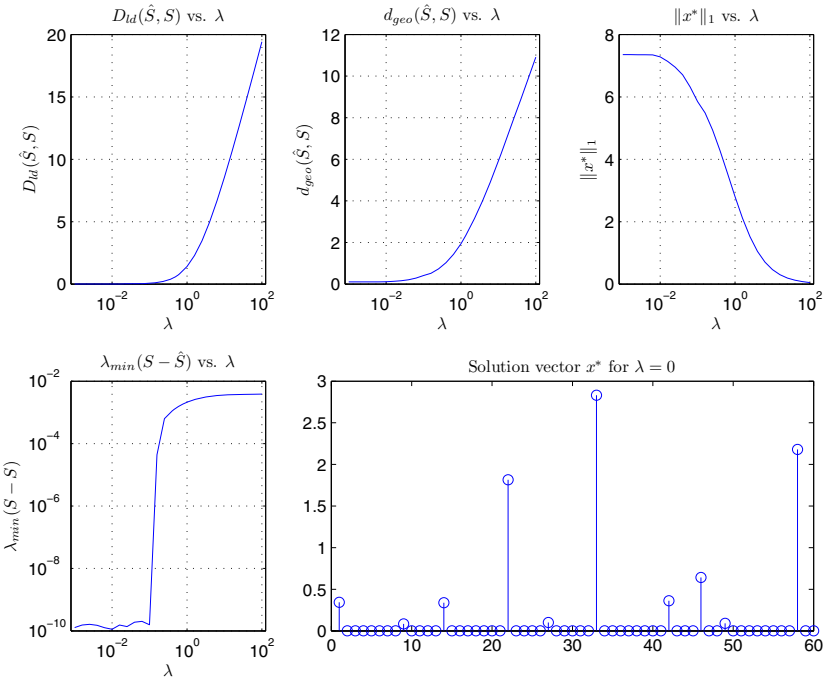


Fig. 2. Plot of the various quantities vs. λ for $n = 5$, $k = 60$. We show $D_{\text{ld}}(\hat{S}, S)$, $d_{\text{geo}}(\hat{S}, S)$, $\|x^*\|_1$, as well as $\lambda_{\min}(S - \hat{S})$, plotted in logarithmic scale. The λ values are varied logarithmically. The solution vector x^* in the unconstrained case is also shown on the right, and is observed to be sparse even without explicitly enforcing any sparsity.

and generate sets of samples from a multivariate Gaussian distribution $\mathcal{N}(0, C)$. There are $O(n^2)$ samples per set, from which we compute the inverse covariance for each of these sets. These precision matrices forms our data set. We select k of these matrices to form our dictionary $\mathcal{A} = \{A_i\}_{i=1}^k$. The sample point S to be sparse-coded is also generated in this manner. The precision matrix of a multivariate Gaussian distribution follows a Wishart distribution [22], and therefore our optimization problem is well suited to this model. The quantities we consider to represent the performance of the reconstruction are the LogDet Divergence $D_{\text{ld}}(\hat{S}, S)$, the geodesic distance $d_{\text{geo}}(\hat{S}, S)$, the ℓ_1 norm of the optimal coefficient vector $\|x^*\|_1$ and the minimum eigenvalue of the residual $\lambda_{\min}(S - \hat{S})$.

Effect of normalization. In vector sparse-coding and dictionary learning, the dictionary atoms are usually normalized to have unit length. In a similar fashion, we tried different ways to normalize the atoms in our dictionary, *viz.*, by spectral norm, $\|A_i\|_2$, Frobenius norm, $\|A_i\|_F$, or by trace, $tr(A_i)$. Since all matrix norms are equivalent, we only get a proportional change in the quality of approximation, as is expected. Throughout the rest of this section, we adhere to normalization by spectral norm.

Effect of sparsity constraints. Figure 2 shows the effect of varying λ on the quality of reconstruction, under the Type I problem. The geodesic distance can be seen to vary in a smooth and similar fashion to the LogDet divergence, reaffirming our choice of objective function. We also show the actual solution vector x^* for $\lambda = 0$, where it can be seen that even the unconstrained case results in a sparse solution vector. This is due to the fact that we require a non-negative coefficient vector, and it is widely noted in the vector-domain that non-negative decompositions result in sparsity, under certain conditions [26,27,28].

4.2 Classification Experiments

We evaluate the tensor sparse coding algorithm in a classification framework, where the training data is used as a dictionary $\mathcal{A}$, and the test point S is approximated by a sparse non-negative linear combination of the dictionary atoms. In all the following experiments, we use the Type I objective function for sparse coding, with $\lambda = 10^{-3}$.

The datasets used are comprised of region covariance descriptors from various applications such as human appearance clustering, texture classification and face recognition. The classification is performed in 4 different ways as follows:

- Geodesic KNN – K-nearest-neighbor classification with $K = 5$, using the Riemannian geodesic distance.
- Kernel SVM classification – Using the multi-class SVM approach, with the kernel matrix computed as

$$K(C_i, C_j) = \exp\left(-\frac{d^2_{\text{geo}}(C_i, C_j)}{2\sigma^2}\right) \quad , \tag{34}$$

  with $\sigma = 1$, we perform classification of the test set with the help of the software `LIBSVM` [29] for the SVM classification.
- $SC + WLV$ – In this method, the coefficient vector x is used as a weight vector to vote for the different class labels. In other words, the label k^* of S is computed as

$$k^* = \arg\max_k \sum_{A_i \in \mathcal{C}_k} x_i \quad , \tag{35}$$

  where $\mathcal{C}_k$ denotes class k. Each dictionary atom A_i votes with its own class label, and its vote is weighted by the corresponding coefficient x_i. The class which gets the highest vote is assigned as the class label of S, hence the name *Weighted Label Voting* (WLV).
- $SC + REC$ – Another method involving sparse coding is adapted from [30], where after the sparse coefficient vector is obtained, the positive definite matrix is reconstructed from atoms (and corresponding coefficients) from each class in the dictionary separately. The class which gives the minimum residual reconstruction error (REC), in terms of the LogDet divergence, is assigned to the new descriptor S.

Table 1. Classification accuracy for the *Cam5* dataset

Classifier	Mean Accuracy (%)	Std. Dev (%)
Geodesic KNN	62.76	2.59
Kernel SVM	72.59	4.94
SC + WLV	75.54	3.17
SC + REC	77.20	3.06

As mentioned earlier, much of the relevant literature on region covariances use the geodesic KNN for classification. Also, the SVM is a powerful and popular classifier in computer vision applications. Hence our choice of these two algorithms to compare our results. The geodesic KNN and the kernel SVM classification are performed directly on the covariance descriptors. The last two methods involve sparse coding, and since our problem formulation is derived under the LogDet divergence and corresponds to the precision matrix, we perform the sparse coding over the inverse of the covariance descriptors.

Human Appearance Descriptors. We use a subset of the 18-class *Cam5* dataset from [18], from which we choose the 16 classes which contain at least 10 data points each. From each of these 16 classes, we select 5 points for training and 5 for testing. The dictionary $\mathcal{A}$ is therefore comprised of $k = 80$ atoms. The descriptors are 5×5 covariances computed from the $\{\mathrm{R,G,B},I_x,I_y\}$ features at each pixel corresponding to the human foreground blobs. The classification accuracy for this dataset averaged over 100 random train-test splits is shown in Table 1. The sparse coding results provide a notable increase in accuracy compared to the KNN or SVM techniques.

Gabor-based Region Covariances for Face Recognition. We compare the classification performance of the $SC + WLV$ and $SC + REC$ methods with the geodesic KNN for the process of face recognition. We test over a subset of the FERET face database [31], using similar pre-processing as in Pang et al. [14]. The images from 10 subjects are taken from the grayscale FERET database, and correspond to the two letter codes *'ba'*, *'bd'*, *'be'*, *'bf'*, *'bg'*, *'bj'*, and *'bk'*. In each experiment, 3 of these are taken as training images, and the remaining 4 as test images, yielding a total of $\binom{7}{3} = 35$ different train-test splits.

The images are convolved with Gabor filters with 8 orientations $u = 0, \ldots, 7$, and up to 3 scales $v = 0, 1, 2$. The Gabor filters are constructed with the same parameters as explained in [14]. Let $g_{uv}(x, y)$ denote the Gabor-filter output at orientation u and scale v. Let $v_{\max}$ be the maximum scale of the Gabor filter in a dataset. We compute 3 datasets of region covariances for each value of $v_{\max} = 0, 1, 2$, comprised of different sets of features, as follows:

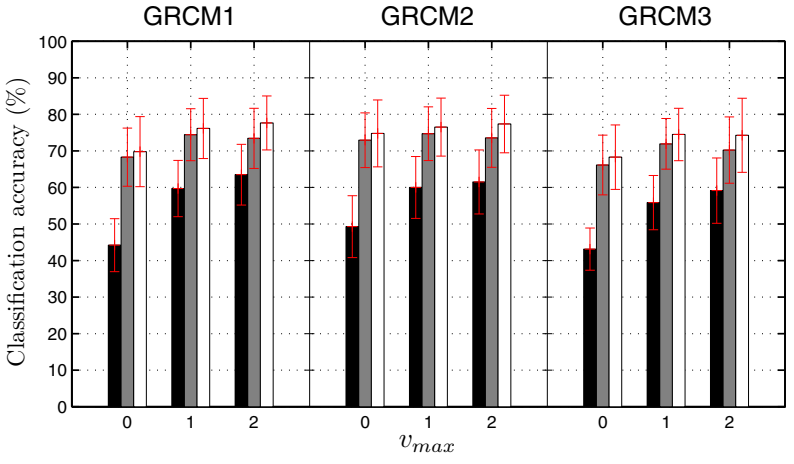


Fig. 3. Classification accuracy for 10 classes with the geodesic KNN (black), $SC{+}WLV$ (gray) and the $SC + REC$ (white) classifiers, for the Gabor-based region covariance datasets GRCM1, GRCM2, and GRCM3. The results are averaged over 35 trials, and 1σ standard deviation bars are shown.

- GRCM1 – $\{\ x,\ y,\ g_{00}(x,y), \ldots, g_{7v_{\max}}(x,y)\ \}$
- GRCM2 – $\{\ x,\ y,\ I,\ g_{00}(x,y), \ldots, g_{7v_{\max}}(x,y)\ \}$
- GRCM3 – $\{\ g_{00}(x,y), \ldots, g_{7v_{\max}}(x,y)\ \}$

yielding a total of 9 different datasets. For each of these 9 datasets, we average over the 35 runs of distinct train-test splits. The classification accuracies for the geodesic KNN and the two tensor sparse coding classification algorithms are shown in Fig. 3. It can be seen that even with fewer feature dimensions, the tensor sparse coding outperforms the KNN classifier significantly. The kernel SVM performs very poorly ($< 30\%$ accuracy) on this dataset, and hence is not shown.

Texture Classification. We now use the region covariances for texture classification, on the Brodatz dataset [32]. We use the training images in the database used to construct the 5-texture (*'5c'*, *'5m'*, *'5v'*, *'5v2'*, *'5v3'*), 10-texture (*'10'*, *'10v'*) and 16-texture (*'16c'*, *'16v'*) mosaics. From each image, 32×32 blocks are cut out, and a 5×5 region covariance descriptor is computed for each block using the grayscale intensities and absolute values of the first- and second-order spatial derivatives, $\{I, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|\}$.

Each image is 256×256 pixels, yielding 64 data points per image. For a k-class problem, we get $64k$ data points, where $k = 5$, 10, or 16. In each case, 5 data points from each class are used to construct the dictionary $\mathcal{A}$, $|\mathcal{A}| = 5k$, and the remaining $59k$ points are used for testing. The classification results are averaged over 20 random train-test splits, and are shown in Fig. 4. The sparse-coding-based methods consistently beat the KNN classifier, and is competitive

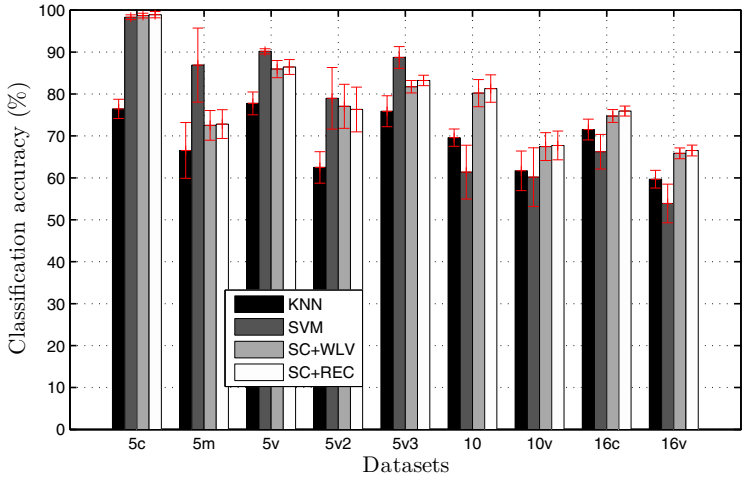


Fig. 4. Texture classification results on the Brodatz dataset, consisting of five 5-class, two 10-class and two 16-class problems. The results are averaged over 20 trials, and 1σ standard deviation bars are also shown.

with the SVM classifier. In fact, as number of classes increases, the sparse coding methods overtake the SVM classifier.

5 Conclusions and Future Work

We have proposed a novel sparse coding technique for positive definite matrices, which is convex and belongs to the standard class of MAXDET optimization problems. The performance of the tensor sparse coding in terms of accuracy of reconstruction, sparsity of the decomposition, as well as variations for different input parameters is analyzed. Results are shown not only for synthetic data but also for data sets from real-world computer vision applications, demonstrating the suitability of our model. In classification performance, the algorithms based on tensor sparse coding beat the state-of-the-art methods by a reasonable margin.

This work opens the door for the many sparsity-related algorithms to the space of positive definite matrices, and many techniques that require only a sparse coding step follow through readily from our work. Future work involves applying the above techniques to areas such as Diffusion Tensor Imaging. We are currently working on developing dictionary learning techniques over the positive definite matrix data, so that we may also learn a suitable dictionary in a data-driven manner, depending on the application at hand.

Acknowledgments. We are thankful to Professor Zhi-Quan (Tom) Luo, Ajay Joshi and Anoop Cherian (University of Minnesota) for their thoughtful input.

This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract #911NF-08-1-0463 (Proposal 55111-CI) and the National Science Foundation through grants #CNS-0324864, #CNS-0420836, #IIP-0443945, #IIP-0726109, #CNS-0708344, #CNS-0821474, #IIP-0934327, #IIS-0534286, and #IIS-0916750.

References

1. Vandenberghe, L., Boyd, S., Wu, S.: Determinant Maximization with Linear Matrix Inequality Constraints. SIAM Journal on Matrix Analysis and Applications (19), 499–533 (1998)
2. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. Roy. Statist. Soc. Ser. B 58(1), 267–288 (1996)
3. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least Angle Regression. Annals of Statistics 32(2), 407–499 (2004)
4. Tropp, J., Gilbert, A.: Signal Recovery from Random Measurements via Orthogonal Matching Pursuit. IEEE Transactions on Information Theory 53(12), 4655–4666 (2007)
5. Donoho, D.: Compressed Sensing. IEEE Transactions on Information Theory 52(4), 1289–1306 (2006)
6. Candès, E., Wakin, M.: An Introduction to Compressive Sampling. IEEE Signal Processing Magazine 25(2), 21–30 (2008)
7. Wright, J., Yi, M., Mairal, J., Sapiro, G., Huang, T.S., Yan, S.: Sparse Representation for Computer Vision and Pattern Recognition. Proceedings of the IEEE 98(6), 1031–1044 (2010)
8. Hazan, T., Polak, S., Shashua, A.: Sparse Image Coding Using a 3D Non-negative Tensor Factorization. In: International Conference of Computer Vision, pp. 50–57 (2005)
9. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to Kernel-based Learning Algorithms. IEEE Transactions on Neural Networks 12(2), 181–201 (2001)
10. 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. 697–704. Springer, Heidelberg (2006)
11. Porikli, F., Tuzel, O.: Fast Construction of Covariance Matrices for Arbitrary Size Image Windows. In: Proceedings of IEEE International Conference on Image Processing, pp. 1581–1584 (2006)
12. Tuzel, O., Porikli, F., Meer, P.: Pedestrian Detection via Classification on Riemannian Manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(10), 1713–1727 (2008)
13. Pennec, X., Fillard, P., Ayache, N.: A Riemannian Framework for Tensor Computing. International Journal of Computer Vision 66(1), 41–66 (2005)
14. Pang, Y., Yuan, Y., Li, X.: Gabor-Based Region Covariance Matrices for Face Recognition. IEEE Transactions on Circuit and Systems for Video Technology 18(7), 989–993 (2008)
15. Hu, H., Qin, J., Lin, Y., Xu, Y.: Region Covariance based Probabilistic Tracking. In: 7th World Congress on Intelligent Control and Automation, pp. 575–580 (2008)
16. Palaio, H., Batista, J.: Multi-object Tracking using an Adaptive Transition Model Particle Filter with Region Covariance Data Association. In: 19th International Conference on Pattern Recognition, pp. 1–4 (2008)

17. Paisitkriangkrai, S., Shen, C., Zhang, J.: Fast Pedestrian Detection Using a Cascade of Boosted Covariance Features. IEEE Transactions on Circuits and Systems for Video Technology 18(8), 1140–1151 (2008)
18. Sivalingam, R., Morellas, V., Boley, D., Papanikolopoulos, N.: Metric Learning for Semi-supervised Clustering of Region Covariance Descriptors. In: Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras, pp. 1–8 (2009)
19. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-Theoretic Metric Learning. In: Proceedings of International Conference on Machine Learning, pp. 209–216 (2007)
20. Kulis, B., Sustik, M., Dhillon, I.: Learning Low-Rank Kernel Matrices. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 505–512 (2006)
21. Banerjee, S., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman Divergences. Journal of Machine Learning Research 6, 1705–1749 (2005)
22. Wishart, J.: The Generalised Product Moment Distribution in Samples from a Normal Multivariate Population. Biometrika 20A(1/2), 32–52 (1928)
23. Wang, S., Jin, R.: An Information Geometry Approach for Distance Metric Learning. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, pp. 591–598 (2009)
24. Tropp, J.: Just Relax: Convex Programming Methods for Identifying Sparse Signals in Noise. IEEE Transactions on Information Theory 52(3), 1030–1051 (2006)
25. Grant, M., Boyd, S.: CVX: Matlab Software for Disciplined Convex Programming, ver. 1.21 (2010), http://cvxr.com/cvx
26. Donoho, D., Tanner, J.: Sparse Nonnegative Solution of Underdetermined Linear Equations by Linear Programming. Proceedings of the National Academy of Sciences 102(27), 9446–9451 (2005)
27. Donoho, D., Stodden, V.: When does Non-negative Matrix Factorization give a Correct Decomposition into Parts? In: Thrun, S., Saul, L., Sch'olkopf, B. (eds.) Advances in Neural Information Processing Systems, vol. 16, pp. 1141–1148. MIT Press, Cambridge (2004)
28. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems, vol. 13, pp. 556–562. MIT Press, Cambridge (2000)
29. Chang, C.-C., Lin, C.-J.: LIBSVM: A Library for Support Vector Machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
30. Wright, J., Yang, A., Ganesh, A., Sastri, S.S., Ma, Y.: Robust Face Recognition via Sparse Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(2), 210–227 (2009)
31. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET Evaluation Methodology for Face-recognition Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)
32. Randen, T., Husoy, J.H.: Filtering for Texture Classification: A Comparative Study. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(4), 291–310 (2009)

Improving Local Descriptors by Embedding Global and Local Spatial Information

Tatsuya Harada*, Hideki Nakayama, and Yasuo Kuniyoshi

Graduate School of Information Science and Technology, The University of Tokyo,
7-3-1 Hongo Bunkyo-ku, Tokyo 113-8656, Japan
{harada,nakayama,kuniyosh}@isi.imi.i.u-tokyo.ac.jp
http://www.isi.imi.i.u-tokyo.ac.jp/

Abstract. In this paper, we present a novel problem: "Given local descriptors, how can we incorporate both local and global spatial information into the descriptors, and obtain compact and discriminative features?" To address this problem, we proposed a general framework to improve any local descriptors by embedding both local and global spatial information. In addition, we proposed a simple and powerful combination method for different types of features. We evaluated the proposed method for the most standard scene and object recognition dataset, and confirm the effectiveness of the proposed method from the viewpoint of speed and accuracy.

Keywords: Local Auto Correlation, Weight Maps, Probabilistic Linear Discriminant Analysis, Scene and Object Recognition.

1 Introduction

Generic image recognition is one of the important problems of computer vision. However, generic image recognition has not yet been put to practical use, though specific detection techniques such as face detection and person detection are at the production level. The difficulty for generic image recognition is that the images should be recognized even if they appear on different scales and cluttered backgrounds when shown from different perspective views. We focus on images with "spatial biases" as the first step in generic image recognition. If a person takes pictures of objects and scenes, the composition of the pictures has some common properties. For example, the objects are arranged in the center of the picture. In this manner, "spatial biases" occur in the images which the person took for the same purpose. Therefore, image features with spatial information are effective in many cases.

Let us briefly review the descriptors and the features for generic image recognition. Self Similarity [1] and Geometric Blur [2] are descriptors containing spatial information. These descriptors represent local spatial structures on interesting points or grid points in the image. The SIFT [3] is the standard descriptor which

* PRESTO, JST.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 736–749, 2010.

represents local spatial information. On the other hand, the HOG feature [4] is formed by concatenating gradient histograms in each cell, and represents global spatial information. The Gist feature [5] is implemented by dividing the image into grid regions, calculating spatial envelopes in each cell, and finally combining spatial envelopes to form one image feature. Therefore, Gist represents a global spatial structure. The PHOG [6] or PHOW [7] features consist of a weighted concatenation of the HOG or Bags of Words (BoW) features [8] over each image sub-region at each resolution level. They also represent global spatial information. Notice that BoW itself discards spatial information, but PHOW which incorporates spatial information into BoW obtains better results on some object recognition datasets. The above mentioned descriptors and features have been proven experimentally to have good performance for functions such as object recognition, scene recognition and human detection. This fact reveals the importance of designing good features with both local and global information of the image.

In this paper, we present a novel problem: "Given local descriptors, how can we incorporate both local and global spatial information into the descriptors, and obtain compact and discriminative features?" For this problem, we propose a general framework to improve any local descriptors by embedding both local and global spatial information, and improve recognition performance for the task where spatial information is essential.

In addition, by applying the proposed framework to many descriptors, we can obtain multiple features from the image. In this situation, a classifier, which combines multiple features, is also important. Recently, Multiple Kernel Learning (MKL) (for example [9]) has attracted attention as a powerful classifier combining weighted multiple kernel machines. MKL is based on a kernel method, and thus is faced with the problem of learning time for a large amount of training data in exchange for high classification performance. Furthermore, a nearest neighbor approach acquires high classification performance without learning time, and in general, it needs a huge amount of classification time, because an input pattern is compared with all the training patterns or all the training local descriptors. To solve this problem the hashing technique is usually employed. In this paper, we propose a simple classifier "Naive Bayes Probabilistic Linear Discriminant Analysis (PLDA)" which combines many features based on a Naive Bayes scheme. This classifier does not need an optimization process to assign weights to each feature. In addition, it does not need to perform a comparison with all patterns, but with a small number of prototypes. For this reason, the proposed classifier is fast both in learning and classification processes.

Our proposed framework is inspired by many previous studies. The calculation of local spatial information is based on Higher-order Local Auto Correlation (HLAC) features [10]. Improving any local descriptors is inspired by the Covariance [11] and GLC [12] features. The calculation of global spatial information is based on Fisher Weight Maps and Eigen Weight Maps [13]. Our technical contribution is to generalize those techniques, and propose a new framework to incorporate local and global spatial information into arbitrary local descriptors.

Most descriptors are improved substantially by applying our method (see Section 6). To the best of our knowledge, no one has proposed a general framework to improve any local descriptors by incorporating both local and global spatial information. Furthermore, the Naive Bayes PLDA is a new approach in combining different types of features. This approach is simple and fast, and obtains good results.

2 Outline of the Proposed Method

In this section, we explain the outline of the proposed method. Initially, an input image is partitioned into spatial grids (cell). M is the number of partitioned cells. In each cell, we calculate features representing the cell from local descriptors. We call these features region features. We extract K kinds of features, such as texture, shape and color. The k-th feature of j-th region in the image I_i is denoted by $\boldsymbol{f}_{ij}^{(k)} \in \mathbb{R}^{d^{(k)}}$. For the image I_i, one feature $\boldsymbol{f}_i$ is obtained by concatenating all region features.

$$\boldsymbol{f}_i^{(k)} = (\boldsymbol{f}_{i1}^{(k)T} \cdots \boldsymbol{f}_{iM}^{(k)T})^T, \tag{1}$$

$$\boldsymbol{f}_i = (\boldsymbol{f}_i^{(1)T} \cdots \boldsymbol{f}_i^{(K)T})^T. \tag{2}$$

Now, we consider C classes $\{\omega_l\}_{l=1}^C$, and use Bayes decision rule to classify the image feature $\boldsymbol{f}_i$.

$$c = \arg\max_l \{P(\omega_l|\boldsymbol{f}_i)\} \Rightarrow \boldsymbol{f}_i \in \omega_c. \tag{3}$$

Using Bayes' theorem, and assuming that the prior probability is the same value for all the classes, the decision rule becomes

$$c = \arg\max_l \{p(\boldsymbol{f}_i|\omega_l)\} \Rightarrow \boldsymbol{f}_i \in \omega_c. \tag{4}$$

Here, the problem is how to estimate the probability density $p(\boldsymbol{f}_i|\omega_l)$, and how to handle the high dimensional image feature $\boldsymbol{f}_i$, which is generated by combining many kinds of features. Direct application of the feature $\boldsymbol{f}_i$ is inadvisable because of dimensionality. Therefore, we assume all the k type features $\boldsymbol{f}_i^{(k)}$ are independent conditioning on the class ω_l, and convert the problem into the estimation of each probability density in low dimensional space.

$$p(\boldsymbol{f}_i|\omega_l) = p\big((\boldsymbol{f}_i^{(1)T} \cdots \boldsymbol{f}_i^{(K)T})^T|\omega_l\big), \tag{5}$$

$$= p\big(\boldsymbol{f}_i^{(1)}|\omega_l\big)p\big(\boldsymbol{f}_i^{(2)}|\omega_l\big) \cdots p\big(\boldsymbol{f}_i^{(K)}|\omega_l\big), \tag{6}$$

$$= \prod_{k=1}^{K} p\big(\boldsymbol{f}_i^{(k)}|\omega_l\big). \tag{7}$$

The log of Eqn. 7 can be written in the form

$$\ln p(\boldsymbol{f}_i|\omega_l) = \sum_{k=1}^{K} \ln p\big(\boldsymbol{f}_i^{(k)}|\omega_l\big). \tag{8}$$

The problem is simplified to estimate the likelihood $p(\boldsymbol{f}_i^{(k)}|\omega_l)$ for each k-th feature following the naive Bayes approach. The assumption of conditional independence is a very strict assumption, but the naive Bayes approach has been proved to show higher performance than expected [14]. In fact, in generic object recognition, the Naive Bayes Nearest Neighbor (NBNN) approach [15] achieved satisfactory results, hence we expect our approach to achieve good performance in image recognition.

Although the problem is divided into the estimation of $p(\boldsymbol{f}_i^{(k)}|\omega_l)$, the feature $\boldsymbol{f}_i^{(k)}$ is still a high dimensional vector, since the k-th feature consists of M region features. Obviously, the k-th feature can be divided into region features with conditional independence assumptions for all sorts of features.

$$p(\boldsymbol{f}_i^{(k)}|\omega_l) = \prod_{m=1}^{M} p(\boldsymbol{f}_{im}^{(k)}|\omega_l). \tag{9}$$

In this assumption, we discard spatial information between the cells. Using spatial information enhances classification performance for images with a strong alignment of objects [6]. For this reason, we consider the weighted sum of region features to implicitly represent spatial information.

$$\boldsymbol{g}_i^{(k)} = w_1^{(k)}\boldsymbol{f}_{i1}^{(k)} + w_2^{(k)}\boldsymbol{f}_{i2}^{(k)} + \cdots + w_M^{(k)}\boldsymbol{f}_{iM}^{(k)}, \tag{10}$$

$$= \sum_{m=1}^{M} w_m^{(k)}\boldsymbol{f}_{im}^{(k)}, \tag{11}$$

where $w_m^{(k)} \in \mathbb{R}$ is a weight for the m-th region of the k-th feature in the image I_i. Let $F_i^{(k)} \in \mathbb{R}^{M\times d^{(k)}}$ denote the $M \times d^{(k)}$ matrix where the M row vectors are the region features.

$$F_i^{(k)T} = (\boldsymbol{f}_{i1}^{(k)} \cdots \boldsymbol{f}_{iM}^{(k)}). \tag{12}$$

Using this matrix, Eqn. 11 can be simplified as follows:

$$\boldsymbol{g}_i^{(k)} = F_i^{(k)T}\boldsymbol{w}^{(k)}, \tag{13}$$

where $\boldsymbol{w}^{(k)} \in \mathbb{R}^M$ is the region weight vector $\boldsymbol{w}^{(k)} = (w_1^{(k)} \dots w_M^{(k)})^T$.

The region weight is not limited to one weight vector. We can prepare some region weight vectors, and obtain the new feature vectors by concatenating $\{\boldsymbol{g}_{ij}^{(k)} = F_i^{(k)T}\boldsymbol{w}_j^{(k)}\}_{j=1}^{M'}$, which are the weighted region features.

$$\boldsymbol{g}_i^{(k)'} = (\boldsymbol{g}_{i1}^{(k)T} \dots \boldsymbol{g}_{iM'}^{(k)T})^T \tag{14}$$

Taking the dimensionality reduction into consideration, the number of region weight vectors is generally $M' \ll M$.

In addition, if $\boldsymbol{g}_i^{(k)'}$ is still a high dimensional vector, we use principal component analysis (PCA) for dimensionality reduction. Let $\boldsymbol{h}_i^{(k)}$ be the transformed

vector by using PCA for $\boldsymbol{g}_i^{(k)'}$. Therefore, the final classification rule of Eqn. 8 becomes:

$$c = \arg\max_l \sum_{k=1}^{K} \ln p\left(\boldsymbol{h}_i^{(k)} | \omega_l\right) \Rightarrow \boldsymbol{h}_i^{(k)} \in \omega_c. \tag{15}$$

We can reduce the problem to the estimation of the region weights and the proper probability distributions. In Sections 4 and 5, we explain the implementation of these methods. In Section 5, we use PLDA to estimate the probability distribution. Therefore, we call the proposed method for the combined multiple features the "Naive Bayes PLDA" method. More importantly, up to this point we have not mentioned the selection of the region features. In the next section, we explain how to build the region features including local spatial information from any local descriptors.

3 Local Spatial Information

In this section, we consider the generation of the region features including local spatial information. To this purpose, we calculate the local auto correlation of arbitrary local descriptors. Let $\boldsymbol{\phi}(\boldsymbol{r}_i)$ be the local descriptor at the reference point $\boldsymbol{r}_i$ and $\boldsymbol{a}_j$ be the displacement vector. Then the first-order auto correlation matrix is obtained by:

$$\Phi(\boldsymbol{a}_j) = \frac{1}{N_J} \sum_{i \in J} \boldsymbol{\phi}(\boldsymbol{r}_i) \boldsymbol{\phi}(\boldsymbol{r}_i + \boldsymbol{a}_j)^T, \tag{16}$$

where J is a region of the image and N_J is the number of local descriptors in the region J. Noticing that 0-th local auto-correlation in the region is the mean of the local descriptors, we have:

$$\bar{\boldsymbol{\phi}} = \frac{1}{N_J} \sum_{i \in J} \boldsymbol{\phi}(\boldsymbol{r}_i). \tag{17}$$

The local auto-correlation of any local descriptors is considered to be a type of Higher-order Local Auto-Correlation Features [10]. By using the elements of the mean and the local auto-correlation matrix, we obtain the region feature:

$$\boldsymbol{f} = (\bar{\boldsymbol{\phi}}^T \ \eta^T(\Phi(\boldsymbol{0})) \ \xi^T(\Phi(\boldsymbol{a}_1)) \ \cdots \ \xi^T(\Phi(\boldsymbol{a}_{n_a})))^T, \tag{18}$$

where $\eta(\cdot)$ returns a column vector consisting of the elements of the upper triangular portion of the input matrix, $\xi(\cdot)$ returns a column vector consisting of all the elements of the input matrix, and n_a is the number of displacement vectors. Selection of the displacement vectors is limited in the local area according to [10], and the number of displacement vectors is five. Figure 1 shows the five kinds of displacement vector in this paper. Although we can calculate the higher order auto-correlations, we usually use up to the first or second order, because the feature dimension exponentially increases with the increase in the order.

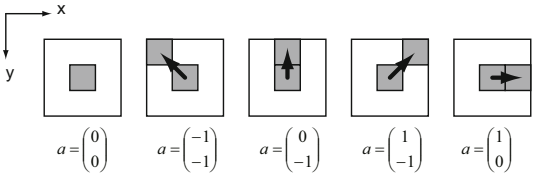


Fig. 1. Displacement vectors of local auto-correlation

Let the dimension of the local descriptor be d, the dimension of the region feature becomes $d + d(d+1)/2 + (n_a - 1)d^2$. In this way, the dimension of the region feature increases as the square of d. If we use the high dimensional local descriptor, it is hard to calculate the region features. Therefore, according to the dimension of descriptors, we can select the calculation of the region feature as follows:

Mean. $\boldsymbol{f} = \bar{\boldsymbol{\phi}}$ This is the first statistic of the local descriptor. The dimension is the same as the descriptor d.

Mean + Local Auto-Correlation at $\boldsymbol{a} = 0$. This is the special case of Eqn. 18, $\boldsymbol{f} = (\bar{\boldsymbol{\phi}}^T \ \eta^T(\Phi(\mathbf{0})))^T$. We call this feature the GLC (Generalized Local Correlation) feature [12]. The dimension is $d + d(d+1)/2$.

Mean + all Local Auto-Correlation. This is the same as Eqn. 18.

The above region features, except the Mean plus all Local Auto-Correlations, do not include spatial information. However, since they calculate the statistics of the local descriptors in the region, we believe these features include meaningful information.

4 Global Spatial Information

In the face recognition, Eigenfaces [16] and Fisherfaces [17] are well known methods for weighting the regions in the image. However, these methods can be applied to images consisting of the scalar values at the pixel such as the brightness, and cannot be applied to the image where each pixel is described by the vector. For this reason, the Fisher Weight Maps (FWM) and Eigen Weight Maps (EWM) are employed as a region weighting method [13]. The original weight maps are defined to weight each pixel in the image. In general, images have different scales and aspect ratios, and the pixel-wise weight maps are not directly utilized in the generic images. Therefore, to absorb the variety of scales and aspect ratios, all images are divided by a regular grid, and weight maps are applied to these regions.

Now, we have the labeled training samples $\{(\boldsymbol{f}_i^{(k)}, y_i)\}_{i=1}^N$. Let $\tilde{\Sigma}_W$ be the within-class covariance matrix of region features, and $\tilde{\Sigma}_B$ be the between-class covariance matrix. The Fisher criterion is given by $J(\boldsymbol{w}) = \frac{tr\tilde{\Sigma}_B}{tr\tilde{\Sigma}_W}$. The traces of $\tilde{\Sigma}_W$ and $\tilde{\Sigma}_B$ are given by:

$$tr\tilde{\Sigma}_W = \boldsymbol{w}^T \Sigma_W \boldsymbol{w}, \tag{19}$$

$$tr\tilde{\Sigma}_B = \boldsymbol{w}^T \Sigma_B \boldsymbol{w}, \tag{20}$$

where

$$\Sigma_W = \frac{1}{N}\sum_{l=1}^{C}\sum_{i\in\omega_l}(F_i^{(k)} - M_l)(F_i^{(k)} - M_l)^T, \tag{21}$$

$$\Sigma_B = \frac{1}{N}\sum_{l=1}^{C} n_l(M_l - M)(M_l - M)^T, \tag{22}$$

M_l is the mean of $F_i^{(k)}$ belonging to the class ω_l, and M is the mean of total data set. By maximizing the Fisher criterion under the condition $\boldsymbol{w}^T\Sigma_W\boldsymbol{w} = 1$, we can obtain the weight vector $\boldsymbol{w}$ as the eigen vector of the generalized eigenvalue problem.

$$\Sigma_B\boldsymbol{w} = \lambda\Sigma_W\boldsymbol{w}, \tag{23}$$

where λ is the eigen value corresponding to the eigen vector $\boldsymbol{w}$. We select the M' largest eigen values $\lambda_1, \cdots, \lambda_{M'}$, and the corresponding eigen vectors $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{M'}$, and calculate the weighted feature vector by using Eqn. 13 and Eqn. 14. These weight vectors are called Fisher Weight Maps (FWM). Because this method uses the matrix consisting of the region features, it is considered to be the generalized Fisher discriminant analysis.

In the case of the presence of clutter or occlusion, there would be difficulty establishing recognition if we use the global spatial information. However, the method automatically weights the discriminative regions, and ignores less discriminative regions. If unobservable regions are less important, this method is expected to work properly in the presence of clutter or occlusion.

5 Classifier

For the estimation of the probability density, Probabilistic Linear Discriminant Analysis is employed. Some variations of PLDA have been proposed [18] [19] [20]. In this paper, reference [18] is utilized whose solution is similar to that of Linear Discriminant Analysis (LDA). We note that the density estimation can be replaced by [19] and [20].

Suppose that N training samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$ are given, and all training samples belong to one of the C classes $\omega_1, \ldots, \omega_C$. In addition, assuming that the test sample $\boldsymbol{x}^t$ belonging to one of the C classes is given, we want to decide the class of the test sample. Let $\boldsymbol{u} = A^{-1}(\boldsymbol{x} - \boldsymbol{m})$ ($A \in \mathbb{R}^{d\times d'}$, $\boldsymbol{m} \in \mathbb{R}^d$) be the Affine transformation that transforms the input vector to the latent variable $\boldsymbol{u}$. Let $\boldsymbol{u}^t$ be the latent variable of the test sample, and $\{\boldsymbol{u}_i\}_{i=1}^N$ be the latent variables of the training samples. The probability that $\boldsymbol{u}^t$ belongs to the class ω_j is given by:

$$p(\boldsymbol{u}^t|\omega_j) = \mathcal{N}\left(\boldsymbol{u}^t|\frac{n_j\Psi}{n_j\Psi + I}\bar{\boldsymbol{u}}_j, I + \frac{\Psi}{n_j\Psi + I}\right), \tag{24}$$

where n_j is the number of samples belonging to the class ω_j, $\bar{\boldsymbol{u}}_j$ is the mean of the samples $\bar{\boldsymbol{u}}_j = \frac{1}{n_j}\sum_{\boldsymbol{u}_i\in\omega_j}\boldsymbol{u}_i$ belonging to the class ω_j, and $\Psi \in \mathbb{R}^{d'\times d'}$ is the diagonal matrix.

Here, we show the calculation of the parameters $\boldsymbol{m}$, A, and Ψ in Eqn. 24. At first, the between-class covariance matrix $S_b \in \mathbb{R}^{d\times d}$ and the within-class covariance matrix $S_w \in \mathbb{R}^{d\times d}$ are given by:

$$S_w = \frac{1}{N}\sum_l \sum_{i\in\omega_l} (\boldsymbol{x}_i - \boldsymbol{m}_l)(\boldsymbol{x}_i - \boldsymbol{m}_l)^T, \tag{25}$$

$$S_b = \frac{1}{N}\sum_l n_l(\boldsymbol{m}_l - \boldsymbol{m})(\boldsymbol{m}_l - \boldsymbol{m})^T, \tag{26}$$

where n_l is the number of samples in class ω_l, N is the number of the total training samples $N = \sum_l n_l$, $\boldsymbol{m}_l = \frac{1}{n_l}\sum_{i\in\omega_l} \boldsymbol{x}_i$ is the mean of the samples in class ω_l, and $\boldsymbol{m} = \frac{1}{N}\sum_i \boldsymbol{x}_i$ is the mean of the total training samples. Next, we calculate the transformation $\boldsymbol{y} = W^T\boldsymbol{x}$, $\boldsymbol{y} \in \mathbb{R}^{d'}$, $W \in \mathbb{R}^{d\times d'}$ that maximizes the between-class covariance matrix to the within-class covariance matrix. This process is the same as LDA, and the optimal projection matrix is given by the solution of generalized eigenvalue problem:

$$S_b W = S_w W \Lambda, \tag{27}$$

where Λ is the diagonal matrix with eigenvalues. The dimension d' of the discriminant space is given by $d' \leq \min(C-1, d)$.

Now we diagonalize the between-class covariance matrix and the within-class covariance matrix ($\Lambda_b = W^T S_b W$, $\Lambda_w = W^T S_w W$). From these diagonalizations, the parameters $\boldsymbol{m}$, A, and Ψ are given by:

$$\boldsymbol{m} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i, \tag{28}$$

$$A = W^{-T}\left(\frac{n}{n-1}\Lambda_w\right)^{1/2}, \tag{29}$$

$$\Psi = \max\left(0, \frac{n-1}{n}\frac{\Lambda_b}{\Lambda_w} - \frac{1}{n}\right), \tag{30}$$

where $n = N/C$.

To solve the PLDA, we calculate only the d dimensional generalized eigenvalue problem in Eqn. 27. Calculation complexity depends on the dimension. It is true that S_b and S_w depend on the number of samples, but it is easy to modify the calculation of correlation matrices into incremental manner. Moreover, because Eqn. 24 is the uni-modal Gaussian distribution, the classification rule in Eqn. 15 is simplified.

6 Experiment

6.1 Setup

We selected the following descriptors and features.

HLAC. We used at most second order HLAC features [10]. As a preprocessing step, we extracted edges by using the Canny operator, and obtains the

binary images. The HLAC features were extracted for the binary images. The dimension of at most second order HLAC features was 25.

Color HLAC. The Color HLAC features were one of the HLAC features [10] except that the RGB values were utilized as the local descriptor. We used at most first order Color HLAC features. The dimension of at most first order Color HLAC features was 45.

HOG. The HOG (Histograms of Oriented Gradients) [4] was implemented by dividing the image window into small spatial cells. In each cell, the histogram of oriented gradients was calculated. All histograms were concatenated to represent the image. We considered the histogram in each cell as the descriptor. We used an unsigned gradient HOG descriptor, and used a 20° bin size. The dimension of the local descriptor was 9 in our setting.

SIFT. We used the densely sampled SIFT descriptor [3] for each 5 pixels. We did not use the orientation normalization, and calculates the gray SIFT with $r = 2$ and $r = 8$. The dimension of the SIFT was 128. 128 dimensions were too high to calculate the local auto-correlation. Because the SIFT descriptor consists of 4×4 cells, we considered the histogram of each cell as the local descriptor. The dimension of the histogram in the cell was 8. Therefore, since we did not use the SIFT directly, we denote the SIFT- with the "-" mark.

Self Similarity (SS). The Self Similarity [1] calculates the correlation between the reference point and the surrounding points around the reference point. We used the angle bin = 8, and the radial interval = 3. The dimension of this descriptor was 24.

PHOG. The PHOG (Pyramid Histogram of Oriented Gradients) [6] is a global feature, and therefore we did not calculate the GLC and the local auto-correlation, but used the weight maps to reduce the dimensionality.

Gist. The Gist [5] is also a global feature. For this reason, we did not calculate the GLC and the local auto-correlation, but used the weight maps to reduce the dimensionality. We calculated both the gray Gist and RGB Gist for the 6 directions and 6 scales. The dimension of the gray Gist and the RGB Gist were 36 and 108 respectively.

We denote the descriptor, which is improved by embedding Global and Local Spatial (GLS) information, as (descriptor) + GLS. In the same manner, we denote the descriptor with GLC and FWM as (descriptor) + GLC + FWM.

We tested the experiments on the standard workstation (XeonW5590 (3.33 GHz) $\times 2 = 8$ core, 48GB ram) using Matlab. We did not use special acceleration techniques, such as MEX, for the implementation of both learning and classification.

6.2 Scene Classification

We experimented with a commonly used scene classification benchmark dataset by Lazebnik et al., [21] (LSP15). LSP15 consists of gray images of OT8 [5] plus seven additional classes. OT8 consists of 2,688 color images of eight classes. Each class has 260 to 410 sample images. In all, it has 4,492 gray images. We randomly

Table 1. Classification results on LSP15 with single features

Feature	Grid	LDA dim	Maps dim	PCA dim	Classification rate [%]	Learn [sec]	Classify [sec]
HOG (baseline)	2x2	14	4	36	54.8 ± 1.4	0.04	0.02
	4x4	14	16	100	61.5 ± 1.8	0.11	0.02
	6x6	14	36	100	**62.3 ± 0.6**	0.28	0.02
	8x8	14	64	100	61.2 ± 1.1	1.35	0.02
HOG +GLC +FWM	2x2	14	4	216	67.6 ± 1.0	0.20	0.02
	4x4	14	8	300	72.9 ± 1.3	0.64	0.02
	6x6	14	8	300	74.2 ± 0.5	0.64	0.02
	8x8	14	8	300	**74.5 ± 0.7**	0.71	0.02
HOG +GLS (proposed)	2x2	14	4	300	72.6 ± 1.0	24.29	0.04
	4x4	14	5	500	75.9 ± 0.9	35.99	0.04
	6x6	14	5	500	**77.3 ± 0.8**	36.18	0.04
	8x8	14	5	500	77.1 ± 0.7	36.35	0.05
SIFT- (baseline)	2x2	14	4	32	56.6 ± 1.2	0.04	0.02
	4x4	14	16	60	**63.1 ± 0.9**	0.10	0.02
	6x6	14	36	80	62.5 ± 0.7	0.23	0.02
	8x8	14	64	120	61.1 ± 1.4	1.02	0.03
SIFT- +GLC +FWM	2x2	14	4	176	56.9 ± 1.6	0.15	0.02
	4x4	14	8	350	66.9 ± 0.9	0.55	0.03
	6x6	14	8	350	69.0 ± 1.3	0.60	0.03
	8x8	14	8	350	**70.4 ± 0.8**	0.70	0.03
SIFT- +GLS (proposed)	2x2	14	4	600	66.2 ± 1.3	14.39	0.03
	4x4	14	4	600	73.1 ± 1.0	14.50	0.04
	6x6	14	4	600	74.3 ± 0.9	14.65	0.04
	8x8	14	4	600	**75.3 ± 0.7**	15.00	0.04
SS (baseline)	2x2	14	4	96	64.3 ± 1.3	0.07	0.02
	4x4	14	16	100	**65.9 ± 0.7**	0.47	0.02
	6x6	14	36	100	63.3 ± 0.9	4.93	0.03
	8x8	14	64	100	60.9 ± 1.4	40.51	0.04
SS +GLC +FWM	2x2	14	2	400	**80.0 ± 0.6**	2.04	0.03
	4x4	14	2	400	79.4 ± 0.7	2.24	0.03
	6x6	14	2	400	78.7 ± 0.9	2.40	0.03
	8x8	14	2	400	78.3 ± 0.9	2.72	0.03
SS+GLS (proposed)	2x2	14	2	400	**80.8 ± 0.7**	36.03	0.04
	4x4	14	2	400	80.4 ± 0.7	36.43	0.04

chose 100 training images for each class in LSP15. We used the remaining samples as test data, and calculated the mean of the classification rate for each class. This score was averaged over many trials replacing the training and test samples randomly. This is the methodology used in previous studies.

Initially, we tested the performance of the framework of embedding global and local spatial information into any descriptors. We compared (descriptor) + GLS with the baseline features and (descriptor) + GLC + FWM. The baseline features were obtained by concatenating all mean descriptors in each grid. FWM was not applied to the baseline features. The dimensions of PCA were selected to get the best performance for all features. The dimensions of weight maps were also selected to get the best performance for (descriptor) + GLC + FWM and (descriptor) + GLS. We used the simple LDA as the classifier in order to compare the performances of the features themselves.

Table 1 shows these results on LSP15 with the single features. The bold number means the best score in each feature. We can see that GLS improves the classification performance significantly for all descriptors. We changed the classifier to PLDA. SIFT- (2 and 8 scales) + GLS + PLDA obtained 80.1 [%], which is comparable to SIFT + hard quantization + intersection kernel

Table 2. Classification results on Caltech101 with single features

Feature	Grid	LDA dim	Maps dim	PCA dim	Classification rate [%]	Learn [sec]	Classify [sec]
HOG (baseline)	2x2	36	4	36	22.1 ± 1.5	0.1	0.1
	4x4	101	16	120	35.3 ± 1.4	0.2	0.4
	6x6	101	36	120	36.9 ± 0.8	0.3	0.4
	8x8	101	64	120	**37.0 ± 1.0**	1.4	0.4
HOG +GLC +FWM	2x2	101	4	216	30.8 ± 0.9	0.3	0.4
	4x4	101	8	300	40.1 ± 1.6	0.8	0.4
	6x6	101	8	300	41.0 ± 0.6	0.8	0.4
	8x8	101	8	300	**41.2 ± 1.5**	0.9	0.4
HOG +GLS (proposed)	2x2	101	4	300	41.7 ± 1.3	25.1	0.4
	4x4	101	8	300	44.3 ± 1.2	33.9	0.4
	6x6	101	8	300	44.3 ± 1.0	34.2	0.4
	8x8	101	8	300	**45.2 ± 0.6**	34.4	0.4
SIFT- (baseline)	2x2	32	4	32	28.1 ± 1.3	0.1	0.1
	4x4	101	16	120	42.9 ± 1.2	0.2	0.4
	6x6	101	36	120	**44.8 ± 1.7**	0.3	0.4
	8x8	101	64	120	44.4 ± 0.8	1.1	0.4
SIFT- +GLC +FWM	2x2	101	4	176	31.7 ± 1.3	0.2	0.4
	4x4	101	8	350	44.8 ± 0.7	0.9	0.4
	6x6	101	8	350	44.6 ± 0.6	0.9	0.4
	8x8	101	8	350	**47.2 ± 1.4**	1.0	0.4
SIFT- +GLS (proposed)	2x2	101	4	600	46.3 ± 1.0	15.1	0.5
	4x4	101	4	600	50.2 ± 1.2	15.4	0.5
	6x6	101	4	600	52.6 ± 1.1	15.6	0.5
	8x8	101	4	600	**53.4 ± 1.0**	16.2	0.5
SS (baseline)	2x2	96	4	96	40.3 ± 1.4	0.1	0.4
	4x4	101	16	120	**43.8 ± 1.4**	0.5	0.4
	6x6	101	36	120	42.6 ± 1.5	5.3	0.5
	8x8	101	64	120	42.3 ± 1.3	42.4	0.4
SS +GLC +FWM	2x2	101	4	350	50.3 ± 1.0	16.5	0.4
	4x4	101	6	350	50.7 ± 0.9	34.7	0.4
	6x6	101	6	350	**51.7 ± 1.4**	35.0	0.4
	8x8	101	6	350	51.4 ± 1.4	35.5	0.4
SS+GLS (proposed)	2x2	101	4	350	**52.6 ± 0.8**	64.7	0.5
	4x4	101	6	350	52.5 ± 1.3	100.4	0.5

(80.1 [%]), but inferior to SIFT + sparse codes + intersection kernel (84.3 [%]) [22] [23].

We evaluated the combination of multiple features on the LSP15. We used four features (HOG + GLS, SIFT-(2 and 8 scales) + GLS, SS + GLS, gray Gist), and combined them with the Naive Bayes PLDA. In LSP15, our method obtained the comparable score (86.6 [%]) to the state-of-the-art methods (Xiao et al. [24] (88.1 [%]), Zhou et al. [25] (85.2 [%]), Nakayama et al. [12] (84.1 [%]), Bosch et al. [26] (83.7 [%]), Lazebnik et al. [21] (81.4 [%])).

The Naive Bayes PLDA calculates the classifiers of each feature independently, and requires no optimization process to weight each classifier. The learning cost of PLDA is almost the same as LDA. On classification, the Naive Bayes PLDA only sums the log likelihoods of each classifier. The calculation times of LDA on both learning and classification are shown in Table 1. Learning finished within 1 minutes for all features, and classification finished within 0.1 second for all features. Therefore, GLS + (Naive Bayes) PLDA approach is fast, and obtains good performance for the scene classification.

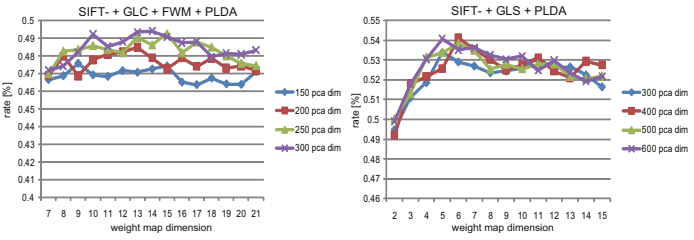


Fig. 2. Effect of the dimension of the weight maps and the PCA dimension on Caltech-101 dataset. The left figure shows the result with the single SIFT- + GLC + FWM. Right figure shows the result with the single SIFT- + GLS (proposed).

6.3 Object Recognition

Caltech-101 [27] is the de-facto standard object recognition dataset. This dataset consists of images from 101 object categories and one background class, and contains from 31 to 800 images per category. This dataset has large intra-class variety. The spatial information is essential for the image recognition, because the images in the same category are well centered. To evaluate classification performance, we followed the most standard methodology. 15 images are randomly selected from all 102 categories for training, and another random 15 for testing.

We tested the performance of the framework of embedding global and local spatial information into any descriptors in the same manner as the scene classification experiment. Table 2 shows these results on Caltech-101 with the single features. The bold number means the best score in each feature. We can see also that (descriptor) + GLS improves the classification performance significantly for all descriptors even in the object recognition dataset. We also checked (descriptor) + GLS + PLDA. SIFT- (2 and 8 scales) + GLS + PLDA obtained 55.4 [%], which is comparable to SIFT + spatial pyramid + hard quantization + kernel SVM (56.4 [%]), but inferior to SIFT + spatial pyramid + sparse codes + max pooling + kernel SVM (67.0 [%]) [22] [23].

We evaluated the performance of the combination of multiple features. We use eight features (HLAC (1/1 and 1/2 resolutions), Color HLAC, HOG + GLS, SIFT- (2 and 8 scales)+ GLS, SS + GLS, PHOG + FWM, gray Gist + FWM, RGB Gist + FWM). Our classification rate achieves 66.4 ± 1.0 [%]. This performance is lower than the state-of-the-art methods (Lin et al. [28] (75.8 ± 1.1 [%]), Boiman et al. [15] (72.8 ±0.39 [%]), Bosch et al. [7] (70.4 ± 0.7 [%])), but has comparable results with Frome et al. [29] (63.2 [%]) , Zhang et al. [30] (59.1 ± 0.56 [%]), and Lazebnik et al. [21] (56.4 [%]). It should be noted that our classification method is very simple and does not use the optimization of weight for each feature, and the learning and classification times are about 150 [sec] and 10 [msec/frame]. Our combination method shows a good trade-off between the computational costs and the classification performance for Caltech-101.

Finally, we evaluated the effect of the dimension of the weight maps and the PCA dimension on the Caltech-101. Figure 2 shows the results of the SIFT- +

GLC + FWM, and SIFT- + GLS with 8×8 grid cells using PLDA. Since there are no spaces to show all results, we picked out only two typical results. We can see that the peak classification performances are achieved around 10 dimensions with the SIFT- + GLC + FWM and SIFT- + GLS. These results show the effectiveness of dimensionality reduction of the weight maps.

7 Conclusions

In this paper, we proposed a general framework to improve any local descriptors by embedding both local and global spatial information. To incorporate local spatial information, we calculated the local auto-correlation of the densely sampled local descriptors, and generated the region features. Then we calculated the weighted sum of the region features by using discriminative weight maps to embed global spatial information. We also proposed a simple classifier "Naive Bayes PLDA" which combined many features based on a Naive Bayes scheme. Experimental results show that our method is very simple and fast, and boosts all descriptors substantially.

Here, we calculated the spatial correlation of the same descriptor. Our idea can be easily extended to the spatial correlation of different descriptors by which the conditional independence of Naive Bayes PLDA can be relaxed. The performance of GLC is improved by using the information geometory based metric [31]. By following this idea, we will also invent the proper similarity measure for our framework.

References

1. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: CVPR (2007)
2. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondence. In: CVPR (2005)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
5. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
6. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: ACM CIVR (2007)
7. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: ICCV (2007)
8. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV International Workshop on Statistical Learning in Computer Vision (2004)
9. Varma, M.: Learning the discriminative power-invariance trade-off. In: ICCV (2007)
10. Otsu, N., Kurita, T.: A new scheme for practical, flexible and intelligent vision systems. In: Proc. IAPR Workshop on Computer Vision (1988)

11. 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)
12. Nakayama, H., Harada, T., Kuniyoshi, Y.: Dense sampling low-level statistics of local features. In: ACM CIVR (2009)
13. Shinohara, Y., Otsu, N.: Facial expression recognition using fisher weight maps. In: IEEE FG (2004)
14. Domingos, P., Pazzani, M.J.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning 29, 103–130 (1997)
15. Boiman, O., Shechtman, E., Irani, M. In: defense of nearest-neighbor based image classification. In: CVPR (2008)
16. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: CVPR (1991)
17. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. on PAMI 19, 711–720 (1997)
18. Ioffe, S.: Probabilistic linear discriminant analysis. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 531–542. Springer, Heidelberg (2006)
19. Yu, S., Yu, K., Tresp, V., Kriegel, H.P., Wu, M.: Supervised probabilistic principal component analysis. In: ACM SIGKDD (2006)
20. Prince, S.J., Elder, J.H.: Probabilistic linear discriminant analysis for inferences about identity. In: ICCV (2007)
21. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
22. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR (2010)
23. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)
24. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: Sun database: Large scale scene recognition from abbey to zoo. In: CVPR (2010)
25. Zhou, X., Cui, N., Li, Z., Liang, F., Huang, T.S.: Hierarchical gaussianization for image classification. In: ICCV (2009)
26. Bosch, A., Zisserman, A., Munoz, X.: Scene classification using a hybrid generative/discriminative approach. IEEE Trans. on PAMI 30, 712–727 (2008)
27. 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 Workshop on Generative-Model Based Vision (2004)
28. Lin, Y.Y., Tsai, J.F., Liu, T.L.: Efficient discriminative local learning for object recognition. In: ICCV (2009)
29. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV (2007)
30. Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: CVPR (2006)
31. Nakayama, H., Harada, T., Kuniyoshi, Y.: Global gaussian approach for scene categorization using information geometry. In: CVPR (2010)

Detecting Faint Curved Edges in Noisy Images

Sharon Alpert, Meirav Galun, Boaz Nadler, and Ronen Basri*

Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot 76100, Israel

Abstract. A fundamental question for edge detection is how faint an edge can be and still be detected. In this paper we offer a formalism to study this question and subsequently introduce a hierarchical edge detection algorithm designed to detect faint curved edges in noisy images. In our formalism we view edge detection as a search in a space of feasible curves, and derive expressions to characterize the behavior of the optimal detection threshold as a function of curve length and the combinatorics of the search space. We then present an algorithm that efficiently searches for edges through a very large set of curves by hierarchically constructing difference filters that match the curves traced by the sought edges. We demonstrate the utility of our algorithm in simulations and in applications to challenging real images.

1 Introduction

This paper addresses the problem of detecting faint edges in noisy images. Noisy images with low signal to noise ratio (SNRs) are common in a variety of domains in which pictures are captured under limited visibility. Examples include electron microscopy (EM) images taken under certain protocols (e.g., cryo-EM), fingerprint images with low tissue contrast, photos acquired under poor lighting, etc. Edges are important since they mark the boundaries of shapes and provide cues to their relief and surface markings. Extracting edges from such images is important, therefore, to allow proper interpretation of their content. Moreover, the study of edge detection under such extreme visual conditions may potentially lead also to better algorithms for handling photographs of natural scenes.

Noise poses a challenge to edge detection, because it can change the contrast along edges, and even lead to local contrast reversals. Smoothing the image (e.g., with a Gaussian filter) reduces the noise, but it may also weaken the contrast across the edges and blend adjacent edges. An ideal filter would match the curve traced by an edge – such a filter can smooth along both sides of the edge to reduce the effect of noise (the longer the filter is, the more the noise is attenuated), while maintaining the contrast across the edge. Utilizing such ideal filters, however,

* Research was conducted in part while RB was at TTI-C. At the Weizmann Inst. research was conducted at the Moross Laboratory for Vision and Motor Control and supported in part by the IFSR.We thank Pedro Felzenswalb for sharing his insights with us.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 750–763, 2010.

is problematic, since the curves traced by the edges are unknown a-priori. Edge detection, therefore, can be viewed as a search in the space of possible curves. Below we use this view to offer a formalism to study a basic question: how high the contrast of an edge has to be to enable its detection. We further propose a method to detect faint edges by searching through a very large set of curves.

Existing edge detection methods use various strategies to overcome noise. Methods that use isotropic smoothing (e.g. [1]) are limited, since aggressive smoothing tends to smear the edges. Anisotropic diffusion methods [2] too face difficulties dealing with low SNRs, as they are typically initialized by local image gradients, whose estimation in noisy images may be unreliable. Recent methods use a variety of filter banks to improve the detection of faint edges, e.g., rectangular filters [3], curvelets [4], shearlets [5], and beamlets [6]. For example, [3,6] use rectangular filters of varying lengths and orientations. These filters are optimized for straight edges and their ability to detect faint curved edges is limited. Other methods compare histograms of intensities and textures in two adjacent half-disks [7,8]. Also related is [9]'s compositional approach to salient curve detection. Finally, [10] considers the statistical problem of detecting the presence of a single monotone edge emanating from a given pixel, but does not consider estimating its exact path.

Below we study the problem of faint edge detection by first characterizing the minimal detectable contrast as a function of curve length and the combinatorics of the set of considered curves. We further show detection limits when *all* monotone (and consequently also general) curves are considered. Subsequently, we introduce a method to detect curved edges at very low SNRs. The method utilizes a hierarchical quadtree data structure to efficiently search through a very large (superpolynomial) set of feasible curves and find the curves that elicit optimal responses. The data structure is analogous to that of the beamlet transform [11], but we use it to search through a much larger set of curves (monotone or non-self intersecting vs. mere straight line segments in each tile), thereby smoothing the noise adaptively along curves of varying shapes as opposed to only straight line segments. We demonstrate the utility of our method by simulations and by applying it to real images.

2 Minimal Detectable Contrast

To study the problem of edge detection in noisy images we view edge detection as a search in a collection of acceptable curves. We address the following question: given the level of noise in the image, what curves can safely be discarded as ones that do not mark an edge. Our aim is to derive a threshold that optimally discard such curves. Naturally such a threshold should depend on the length of the considered curves, and should comply with the following trade off. On one hand we expect the threshold to decrease with curve length since by averaging along longer curves noise is more aggressively attenuated. On the other hand, if the number of curves in our considered collection grows with their lengths then so may the number of false detections; hence we may need to increase

the threshold to control for this growth. This interplay determines the rate of decay of the threshold. Finally, this derivation can tell us the minimal detectable contrast, i.e., whether or not very faint edges can be detected *at all*.

2.1 Derivation

To fix the rate of false detections we consider a pure noise image, $I(x, y)$, with $N = n^2$ pixels where each pixel is distributed as $\mathcal{N}(0, \sigma^2)$. Consider a curved filter of width w measuring the average value of pixels along a curve of length L (i.e., we assume the number of i.i.d. pixel measurements that enter one filter application is wL and hold w fixed in this analysis). Suppose the edges can trace any of N_L different curves. With each potential curve Γ_i, $1 \leq i \leq N_L$ we associate a response R_i corresponding to the average value obtained by the corresponding curved filter. Clearly, R_i is a random variable distributed as $R_i \sim \mathcal{N}(0, \sigma_L^2)$, where $\sigma_L^2 = \sigma^2/(wL)$. Let $T = T(L, N_L)$ denote the *detection threshold*, i.e., an edge of length L is discarded if $|R_i| < T$. Since our image contains only noise each such detection is a false detection. Let $R_{\max} = \max_{1 \leq i \leq N_L} |R_i|$. To fix the rate of false detections, we set T to satisfy $P(R_{\max} \leq T) \geq 1 - \delta$, for a small constant δ $(0 < \delta \leq 0.5)$ independent of L.

To derive the threshold we assume that all R_i's are independent and subsequently that $T \gg \sigma_L$. Under these assumptions we obtain

$$P(R_{\max} \leq T) = [P(|R_i| \leq T)]^{N_L} . \tag{1}$$

Using properties of Q-functions (tails of Gaussians) we can approximate (1) as

$$P(|R_i| \leq T) \approx 1 - \sqrt{\frac{2}{\pi}} \frac{\sigma_L}{T} \exp\left(-\frac{T^2}{2\sigma_L^2}\right) . \tag{2}$$

Consequently, $P(R_{\max} \leq T) \gtrapprox 1 - \delta$ implies

$$\frac{\delta}{N_L} \gtrapprox \sqrt{\frac{2}{\pi}} \frac{\sigma_L}{T} \exp\left(-\frac{T^2}{2\sigma_L^2}\right) . \tag{3}$$

Thus, by taking the natural logarithm, substituting for σ_L, and ignoring small terms we obtain a lower bound for the threshold, i.e., $T \geq \sigma \sqrt{\frac{2 \ln(N_L/\delta)}{wL}}$. We set the threshold conservatively to this lower bound, i.e.,

$$T(L, N_L) \stackrel{def}{=} \sigma \sqrt{\frac{2 \ln N_L}{wL}} \tag{4}$$

while ignoring the effect of δ as it is set to a small constant. This expression is similar to the one derived in [3], with the exception that there N_L was set to N reflecting the number of straight edges of length L considered by that method.

A key assumption to our derivation is that the filter responses of the N_L feasible curves are statistically independent. In practice, this assumption may not hold, as curves may intersect or even partly overlap. As a result the corresponding

threshold may be lower. However, despite this simplification we show simulations on large sets of curves demonstrating a good fit to our predictions.

Equipped with an expression for the threshold we can proceed to determine the detection limits of faint edges for different sets of curves. We consider both the set of monotone curves (curves whose orientations at all points lie in a single quadrant) and the set of general curves with no self-intersections. For the monotone curves we assume that $L \lessapprox \sqrt{N}$ and for general curves that $L \lessapprox N$ since these are the longest such curves that can be obtained in an image. We are interested in two quantities that characterize the detectability of faint edges. The decay rate of the threshold as a function of curve length is captured by the ratio

$$\rho_\alpha = \frac{T(L, N_L)}{T(\alpha L, N_{\alpha L})} = \sqrt{\frac{\alpha \ln N_L}{\ln N_{\alpha L}}} \tag{5}$$

(with a constant $\alpha > 0$ typically set to $\alpha \in \{2, 4\}$), and the limit $T^\infty = \lim_{L,N\to\infty} T$ expresses the limiting value of the threshold. We distinguish between two cases. When $T^\infty > 0$ edges with contrast lower than T^∞ cannot be detected reliably. In this case longer filters do not improve detection. Conversely, when $T^\infty = 0$ (e.g., when $\liminf_{L\to\infty} \rho_\alpha > 1$) the threshold decays polynomially as $T(L, N_L) = O(1/L^{\log_\alpha \rho_\alpha})$. In this case in theory even the faintest edge can be detected, provided that it is sufficiently long.

2.2 Lower Bound for the Full Set of Curves

A basic question is to determine whether very faint edges can be detected if we consider the full set of general, non-self intersecting curves. Obviously this set is exponential in L, since the number of monotone curves in a 4-connected lattice is $2N \cdot 2^L$, and monotone curves form a subset of the non-self intersecting curves. However, while our analysis above implies that T^∞ does not vanish for exponential sets of curves, it is based on the independence assumption.

The following argument suggests that indeed T^∞ is strictly positive. We prove this by deriving a lower bound on T^∞ for the subset of monotone curves. We show this on a 4-connected lattice, but the result immediately extends to lattices with a larger number of connections. As before we consider a noise image. To derive the bound we consider a greedy approach to selecting the monotone curve of highest response emanating from a given point p_0. Let $\Omega \subset \mathbb{R}^2$ denote a 4-connected lattice in 2D, and let $I(x, y)$ denote the i.i.d. random variable at pixel $(x, y) \in \Omega$, $I(x, y) \sim \mathcal{N}(0, \sigma^2)$. Beginning at $p_0 = (x_0, y_0) \in \Omega$, suppose at step i we reach the point $p_i = (x_i, y_i)$, we proceed according to $p_{i+1} = \arg\max(I(x_i+1, y_i), I(x_i, y_i+1))$. Clearly, at every step we choose the maximum of two random variables which are independent of all those previously considered variables. Consequently, the distribution of the random variable at each pixel on this selected monotone curve is determined by $X = \max(Y_1, Y_2)$ where $Y_1, Y_2 \sim \mathcal{N}(0, \sigma^2)$. Such a curve of length L (and width 1) generates a sequence of L i.i.d random variables, whose mean and variance are $\sigma/\sqrt{\pi}$ and $\sigma^2(1 - 1/\pi)$, respectively (e.g., [12]). Therefore, the mean and the variance of the response

associated with the selected *curve* are $\sigma/\sqrt{\pi}$ and $\sigma^2(1-1/\pi)/L$, respectively. Moreover, using the strong law of large numbers, as $L \to \infty$ the probability that the response will obtain the value $\sigma/\sqrt{\pi}$ approaches 1. This shows that for any reasonable value of δ (the false detection rate) T^∞ must be strictly positive (and $\geq \sigma/\sqrt{\pi}$). Consequently, faint edges with contrast lower than T^∞ cannot be detected unless we allow accepting a considerable number of false positives. Note finally that the main arguments in this section extend also to other (non-gaussian), i.i.d. noise.

3 The Beam-Curve Pyramid

As very faint edges cannot be detected when the full set of curves is considered, we turn to constructing an edge detection algorithm that searches through a very large subset of curves while maintaining the detectability of very faint edges. Below we describe the basic principles behind our algorithm and provide expressions for its detection threshold according to the derivations in Sec. 2.

3.1 Construction

Let $\Omega \subset \mathbb{R}^2$ be the discrete two-dimensional grid of image pixels. We associate with Ω a system of square tiles of different areas that are arranged in a quadtree as follows. We use $j = 0, 1, 2, ...$ to denote scale. At every scale j we cover Ω with a collection of tiles of size $(2^j + 1) \times (2^j + 1)$ pixels such that each two adjacent tiles share a common side (see Figure 1). The tiles of different scales are aligned such that each tile of scale $j + 1$ is subdivided into four sub-tiles of scale j.

To each pair of points p_1 and p_2 on different sides of a tile $S^{(j)}$ of scale j we associate a unique curve which we refer to as *beam-curve*. At the finest scale $j = j_0$ the beam-curve is the straight line connecting p_1 and p_2. At coarser scales $j > j_0$ the beam-curve connecting p_1 and p_2 is constructed recursively from beam-curves of scale $j - 1$ according to pre-specified rules. These rules include constraints, specifying the set of curves that can be considered to form beam-curves, and a rule for selecting the optimal beam-curve between p_1 and p_2 among the feasible curves. For optimality we typically choose the curve of highest filter response.

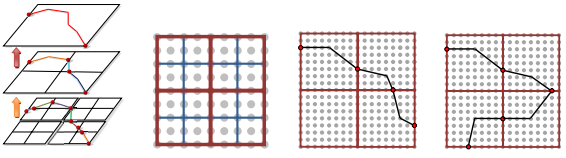


Fig. 1. From left to right: the beam-curve pyramid is a quad-tree structure (left) made of tiles of size $2^j + 1$, $j = 0, 1, 2...$ (second panel). A monotone (third panel) and a general beam-curves (right) at scale $j = 4$ are constructed by stitching curves from the next finer scale ($j = 3$). Here the base level is $j_0 = 2$.

Below we consider two types of constraint rules. *general, simple beam-curves* in a tile S traverse up to four contiguous sections within the four sub-tiles of S, where the traversal through the sub-tiles is either clockwise or counter-clockwise (Figure 1). In particular, these beam-curves are non-self intersecting and lie completely within their respective tile. *Monotone beam-curves* are curves that are monotone with respect to the coordinate axes. A curve is monotone if its tangent vectors at all points along the curve lie within one quadrant. Note that monotonicity depends on the choice of coordinate system.

As we show in Sec. 3.2 below, the construction of the beam-curve pyramid allows us to search through a superpolynomial set of curves, $N_L = O(NL^{\log L})$. This set of curves is a much larger superset of the straight line segments used in both [11,3]. Still beam-curves do not include various curves such as closed curves, spirals, and very windy curves. Our method represents such curves as a concatenation of (usually few) beam-curves and improves their detection by smoothing along each of the constituent beam-curves.

While a superpolynomial number of curves is scanned with this algorithm, the number of beam-curves stored in the pyramid and the cost of its construction are polynomial. The number of beam-curves at every scale is roughly $6N$ where N denotes the number of pixels in the image (see Appendix for details). The total number of beam curves therefore is $O(N \log N)$ [11]. The cost of constructing a full pyramid of general beam-curves is $O(N^{5/2})$, and the cost of constructing a pyramid of monotone curves is $O(N^2)$. Our implementation below focuses on monotone beam-curves. While these complexities may be high for certain practical applications, they can be reduced considerably by terminating the pyramid construction at a fixed scale or sparsifying the beam-curves through pruning. Speedup can also be gained by utilizing a parallel implementation.

3.2 Detection Thresholds

Below we apply our analysis of Sec. 2 to compute the detection thresholds in the beam-curve algorithm. In particular, we show that while for both monotone and general beam-curves the algorithm searches through a superpolynomial set of curves, the detection threshold decays polynomially with curve length.

Monotone beam-curves. Monotone beam-curves at each scale $j > 0$ can pass through up to three sub-tiles of scale $j-1$ in either clockwise or counter-clockwise order. The number of pairs of crossings in both directions is roughly 2^{2j-1} and so for two fixed endpoints the number of possible curves at scale J connecting the endpoints is $\prod_{j=1}^{J} 2^{2j-1}$. Since the total number of endpoint pairs at scale J is $6N$, the total number of possible beam-curves is

$$N_L = 6N \prod_{j=1}^{J} 2^{2 \cdot j-1} = 6N \cdot 2^{J^2}, \tag{6}$$

where $L = 2^J$ is roughly the mean length of curves at scale J. Hence

$$N_L = 6NL^{\log_2 L}, \tag{7}$$

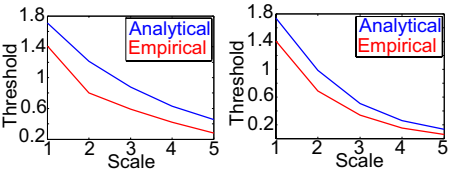


Fig. 2. Comparison of the threshold obtained with our analysis for monotone (left) and general (right) beam-curves with an empirical study

indicating that the set of monotone beam-curves is superpolynomial yet sub-exponential in L, i.e., asymptotically $L^p \ll N_L \ll 2^L$ for any fixed p. Plugging this into (4) we get that $T^\infty = 0$. The ratio

$$\rho_2 = \sqrt{\frac{2(\ln(6N) + J^2 \ln 2)}{\ln(6N) + (J^2 + 2J + 1)\ln 2}} \tag{8}$$

is about $\sqrt{2}$ for short curves ($\ln^2 L \ll \ln N$) and somewhat lower for longer curves ($\ln^2 L \approx \ln N$). By differentiating (8) w.r.t. J we see that for typical values of $10^4 \leq N \leq 10^6$ ρ_2 obtains minimal values of about $1.25 - 1.27$. In summary, for monotone beam-curves the threshold undergoes a polynomial decay indicating that very faint curves can be detected at a sufficiently large scale. Figure 2(left) shows a plot of $T(L, N_L)$ for a $N = 1000 \times 1000$ image.

General beam-curves. Next we consider general beam curves. Consider the curves obtained at level $j > 0$ connecting points on the sides of tiles. At scale j there are $2^j + 1$ pixels in each side of a tile. Such a curve can go through the four sub-tiles of level $j - 1$ in either clockwise or counter-clockwise order. At every crossing from one sub-tile to the next we have $2^{j-1} + 1$ possible crossing pixels, yielding roughly 2^{3j-2} triplets of crossings in both directions. Applying this recursively, the total number of curves considered by the beam-curve pyramid between any two fixed endpoints at scale J is roughly $\prod_{j=1}^{J} 2^{3j-2}$. Since the total number of pairs of endpoints at scale J is $6N$, the total number of possible beam curves at scale J is

$$N_L = 6N \prod_{j=1}^{J} 2^{3j-2} = 6N \cdot 2^{\frac{1}{2}(3J^2 - J)}, \tag{9}$$

where $L = 4^J$ is approximately the average curve length at scale J. As in the case of monotone beam-curves (9) implies that N_L grows with L at a superpolynomial rate. Plugging this into (4) we get that $T^\infty = 0$. The ratio

$$\rho_4 = 2\sqrt{\frac{2\ln(6N) + (3J^2 - J)\ln 2}{2\ln(6N) + (3J^2 + 5J + 2)\ln 2}} \tag{10}$$

is about 2 for short curves ($\ln^2(L) \ll \ln(N)$) and somewhat lower for longer curves ($\ln^2 L \approx \ln N$). By differentiating (10) we see that for the typical values

of $10^4 \leq N \leq 10^6$ ρ_4 obtains minimal values of about $1.71 - 1.76$. Figure 2 (right) shows a plot of $T(L, N_L)$ for a $N = 1000 \times 1000$ image.

4 Algorithm

We present an edge detection algorithm that is based on constructing a beam-curve pyramid as described in Section 3 and on applying the adaptive threshold derived in Section 3.2. The algorithm includes the following main steps:

1. **Initialization:** Construct the bottom level of the beam-curve pyramid by computing straight edge responses in 5×5 ($j_0 = 2$) tiles.
2. **Pyramid construction:** Construct level $j+1$ given level j. Obtain curved responses by stitching up to 3 (for monotone curves) or 4 (for general beam-curves) sub-curves from level j and store for every beam pair the curve of maximal response provided it exceeds the low threshold $\alpha T(L, N_L)$.
3. **Edge selection:** In post processing –
   (a) Discard curves whose associated response falls below the threshold $T(L, N_L)$.
   (b) Discard curves made of short scattered edges.
   (c) Apply spatial non-maximal suppression.
   (d) Apply inter-level suppression.

We next explain these steps in more detail.

Initialization: We begin at level $j_0 = 2$ with tiles of size 5×5 and associate a straight edge response with each pair of points on different sides of each tile. The mean intensity of a straight line γ connecting two points p_1 and p_2 is

$$F(\gamma) = \frac{1}{L(\gamma)} \int_{p_1}^{p_2} I(p)dp, \tag{11}$$

where we define the length as $L(\gamma) = \|p_2 - p_1\|_\infty$. We use the ℓ_∞ norm since it correctly accounts for the number of pixel measurements used to compute the mean. The mean is calculated by utilizing bi-cubic interpolation to achieve sub-pixel accuracy. We further calculate both F and L using the trapezoidal rule so that the end points are counted with weight $1/2$.

We next define a response filter, $R(\gamma)$, for a line γ between p_1 and p_2 as follows. If p_1 and p_2 fall on opposite sides of a tile the filter forms the shape of a parallelogram with

$$R(\gamma) = \left| \frac{\sum_{s=1}^{w/2} \left(L(\gamma^{+s}) F(\gamma^{+s}) - L(\gamma^{-s}) F(\gamma^{-s}) \right)}{\sum_{s=1}^{w/2} \left(L(\gamma^{+s}) + L(\gamma^{-s}) \right)} \right|, \tag{12}$$

where γ^s is the *offset line* connecting $p_1 + (s, 0)$ with $p_2 + (s, 0)$ (or $p_1 + (0, s)$ with $p_2 + (0, s)$) if the points lie on a vertical (respectively horizontal) side, $-w/2 \leq s \leq w/2$ integer. Otherwise, if p_1 and p_2 fall respectively on horizontal and vertical sides the filter forms the shape of a general quadrangle (see Fig. 3).

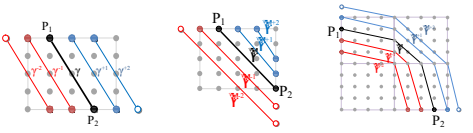


Fig. 3. Straight line filters of width $w = 4$ in a 5×5 tile forming a parallelogram (left) and a general quadrangle (middle). Notice that offset curves can exceed beyond the boundaries of a tile. Right: Stitching three straight filters at level $j_0 = 2$ to produce a monotone curve at level 3.

The response is computed as in (12), where now the offset lines connect $p_1 + (s, 0)$ with $p_2 \pm (0, s)$, depending on which of the four sides each of the points resides on. Note that the offset lines may fall partly outside a tile. In addition, every corner point is considered twice, once as lying on a horizontal side and once on a vertical side.

Pyramid construction: Once a level $j \geq j_0$ in a pyramid is computed we proceed to constructing level $j + 1$. For each pair of points p_1 and p_2 on two different sides of a tile at level $j + 1$ we consider all the curves that begin at p_1 and end at p_2 that can be obtained by stitching up to three curve segments of level j while preserving monotonicity (or up to four segments in the case of general curves, see Figure 3). We then store for each such pair the curve that elicits the highest response, provided that the response exceeds the low threshold $\alpha T(L, N_L)$, where L is the length of the curve (defined below) and $0 \leq \alpha \leq 1$ is constant. We use a low threshold at this stage to allow weak edges to concatenate to produce longer curves that can potentially pass the (high) threshold. For the stitching we consider two curved segments γ_1 connecting p_1 with p_2 and γ_2 connecting p_2 with p_3 on two adjacent tiles at level j. We define the mean intensity of $\gamma = \gamma_1 \cup \gamma_2$ by

$$F(\gamma) = \frac{1}{L(\gamma)} \left(L(\gamma_1) F(\gamma_1) + L(\gamma_2) F(\gamma_2) \right), \tag{13}$$

where $L(\gamma) = L(\gamma_1) + L(\gamma_2)$. Note that due to the use of the trapezoidal rule the point p_2 is counted exactly once. For the response we stitch the corresponding offset curves. We then compute their lengths and means (using (13)) and finally apply (12) to obtain a response. Note that by utilizing differences of means our algorithm deviates from the requirements of dynamic programming, since in some cases the optimal curve at a level $j + 1$ may be composed of sub-optimal curves at level j.

Edge selection: After constructing the pyramid we perform a top-down scan to select the output edges. This is needed since high contrast edges can also give rise to responses that exceed threshold in adjacent locations and in curves that include the real edge as a sub-curve. We consider only curved edges whose response exceeds the (high) threshold $T(L, N_L)$. We further apply a local statistical significance test to distinguish curves whose contrast is consistent along the curve from curves made of short, scattered edges. For each curve γ we denote

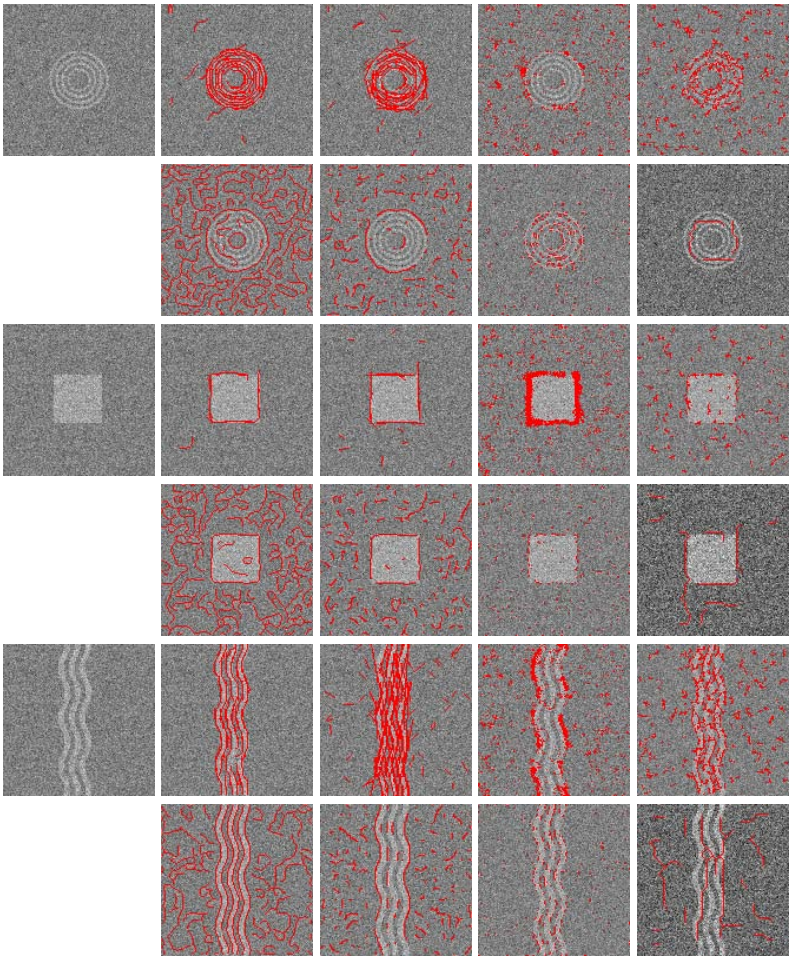


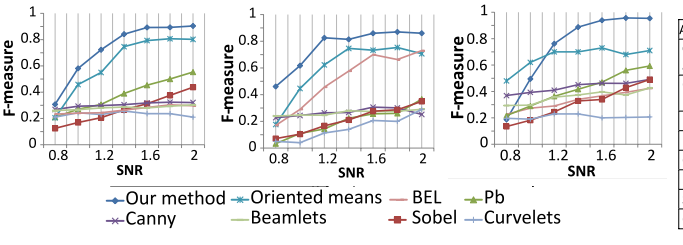
Fig. 4. Simulation examples: Three of the 63 simulation images are shown (left column) including simple patterns in significant noise (SNR 1.8). Each pair of rows shows the result of applying various edge detection algorithms to one of the noisy images. From left to right: our detection algorithm, oriented means, BEL, brightness gradients (top row), Canny, Beamlets, Sobel, and Curvelets (bottom row).

by σ_{local} the average of the two empirical standard deviations of the intensity profiles of the offset curves on the two sides of γ. Hence σ_{local} is an estimate of the local noise in analogy to the global noise σ. We then remove edges for which the response falls below $c\sigma_{local}T(L, N_L)/\sigma$ (typically $c \approx 0.8$). We follow this by spatial non-maximum suppression. In each tile we process the remaining curves in descending order of their responses. We start by accepting the curve of highest response. Then, for each subsequent curve, we accept it if its offset curves do not overlap with previously selected curves. If however they partially overlap with more salient curves we discard the overlapping portion and compute the

statistical significance of the remaining portion as a single curve. We conclude this part by applying inter-level non-maximum suppression. Proceeding from top to bottom, for every curve γ detected at some level J we consider its sub-curves at levels $j_0 \leq j < J$. Finally, we remove curves in those levels whose symmetric median Hausdorff distance to a sub-curve of γ falls below a certain threshold. We then output the collection of curved edges that survived this selection process.

5 Experiments

We evaluate our algorithm both in simulations and on real images and show results on challenging images acquired under unfavorable photography conditions. In all runs we restricted the scope of the algorithm to monotone beam-curves; we noticed at times that general beam-curves tend to produce wiggly curves due to nearby noise. For simulations we prepared 63 binary images of size 129×129 each containing either of three patterns, (1) four concentric circles of widths 4, 7, and 12 pixels separated by 3 pixel gaps, (2) a square of width 20, 40, or 60 pixels, and (3) three sine patterns of widths 3, 5, and 7 separated by 4 pixel gaps. We next scaled the intensities in each image by a factor τ and added i.i.d. zero mean Gaussian noise with standard deviation σ, thus producing images with SNR τ/σ. We compare our algorithm ($\sigma = 13, \alpha = 0.5, c = 0.75$) to several other algorithms including Matlab implementations of Canny [1] (Smoothing with std 2) and Sobel, Local brightness gradients (PB) [13], Boosted edge learning (BEL) [7], oriented means [3], curvelets [4] and our implementation of beamlets [6]. For evaluation we used the F-measure [14], $F = 2PR/(P + R)$, which trades between precision P and recall R. Figure 4 shows examples of the three patterns along with detection results for the various algorithms. Our algorithm managed to detect nearly all the edges with very few false positives. The results are summarized in Fig. 5. It can be seen that our method came in first in nearly all conditions. A notable exception is the case of sine patterns at very low SNR. With such patterns our method is limited by the monotonicity assumption. Still, the method was able to detect sine patterns at slightly higher SNRs better than any of the other



Algorithm	Circles	Square	Sine	Avg.
Our algorithm	**0.73**	**0.76**	**0.74**	**0.74**
Oriented means	0.62	0.60	0.67	0.63
BEL	0.27	0.51	0.33	0.37
PB	0.38	0.28	0.4	0.35
Canny	0.30	0.27	0.43	0.33
Beamlets	0.28	0.26	0.36	0.30
Sobel	0.27	0.21	0.31	0.26
Curvelets	0.23	0.15	0.21	0.20

Fig. 5. Simulation results: F-measures obtained with various edge detection algorithms as a function of SNR, from left to right, for the circle, square, and sine patterns. The table on the right shows the average F-measures (average obtained with SNR ranging from 0.8 to 2 in 0.2 intervals).

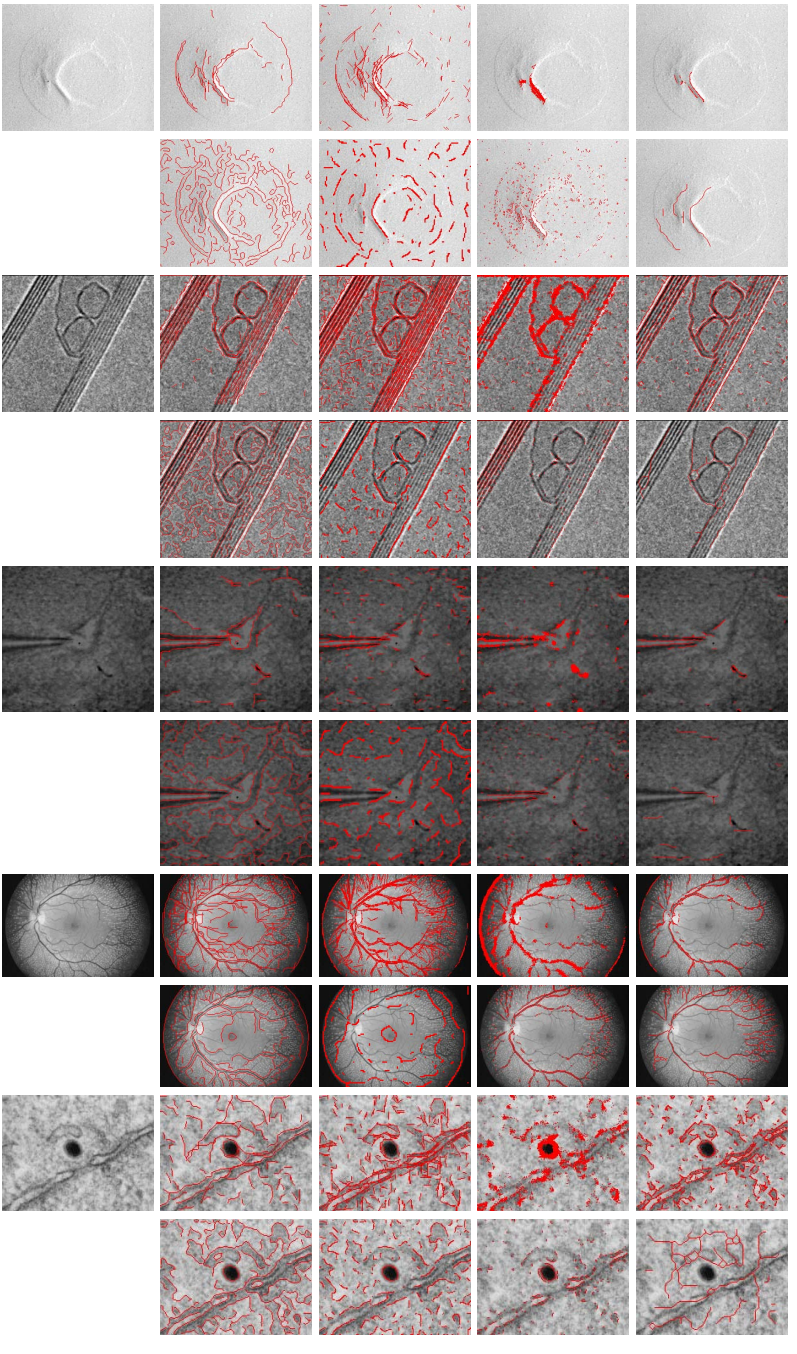


Fig. 6. Real images: Each pair of rows shows the original image and a comparison of our results with those obtained by other algorithms. Results are given in the same order as in Fig. 4.

methods. It should be noted however that some of the compared algorithms are not designed specifically to handle significant amounts of noise.

We next tested our algorithm on the 100 *gray level* images of the Berkeley segmentation dataset and human annotation benchmark [13]. For this test we associated with each curve point a confidence value computed as in [8] with a disc of radius 8. With an F-measure of 0.61, our method was ranked on par with the other leading edge detection methods that rely solely on intensity gradients (e.g., [13] and [3] reported respectively F-measures of 0.60 and 0.61), but was inferior to methods that incorporate texture (F-measures between 0.63 and 0.68 were reported, e.g., by [8,7,15]). This may reflect the relative importance of texture vs. noise in natural images. The median runtime of our algorithm on these images was roughly 8 minutes. Finally, Figures 6 shows the application of our method to challenging images of various sources. To assess the quality of each method one should note not only the accuracy of the true detections, but also the number of associated false detections.

6 Conclusion

We studied the problem of edge detection in noisy images viewing the problem as search in a space of feasible curves. We showed that the combinatorics of the search space plays a crucial role in the detection of faint edges and subsequently developed an algorithm that searches through a very large set of curves, but while maintaining detectability. In future work we hope to further investigate useful shape priors for edges and incorporate those into our formalism.

References

1. Canny, J.: A computational approach to edge detection. TPAMI 8, 679–698 (1986)
2. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. TPAMI 12, 629–639 (1990)
3. Galun, M., Basri, R., Brandt, A.: Multiscale edge detection and fiber enhancement using differences of oriented means. In: ICCV (2007)
4. Geback, T., Koumoutsakos, P.: Edge detection in microscopy images using curvelets. BMC Bioinformatics 10, 75 (2009)
5. Yi, S., Labate, D., Easley, G.R., Krim, H.: A shearlet approach to edge analysis and detection. T-IP 18, 929–941 (2009)
6. Mei, X., Zhang, L.: p., Li, P.x.: An approach for edge detection based on beamlet transform. In: ICIG, pp. 353–357 (2007)
7. Dollar, P., Tu, Z., Belongie, S.: Supervised learning of edges and object boundaries. In: CVPR, pp. 1964–1971 (2006)
8. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. TPAMI 26, 530–548 (2004)
9. Felzenszwalb, P., McAllester, D.: The generalized a* architecture. JAIR 29 (2007)
10. Arias-Castro, E., Candes, E., Helgason, H., Zeitouni, O.: Searching for a trail of evidence in a maze. Annals of Statistics 36, 1726–1757 (2008)

11. Donoho, D., Huo, X.: Beamlets and multiscale image analysis, multiscale and multiresolution methods. In: Engelfriet, J. (ed.) Simple Program Schemes and Formal Languages. LNCS, vol. 20, pp. 149–196. Springer, Heidelberg (1974)
12. Nadarajah, S., Kotz, S.: Exact distribution of the max/min of two gaussian random variables. TVLSI930 16, 210–212 (2008)
13. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its applications to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV, pp. 416–423 (2001)
14. Van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Dept. of Computer Science. University of Glasgow (1979)
15. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR, pp. 1–8 (2008)

Appendix: Complexity Analysis

We denote the number of pixels by $N = n \times n$ and assume further that $n = 2^k + 1$ for some integer $k \geq 0$, $k \approx \log n$. At scale j, $j = 0, 1, ..., k$ a row of tiles includes 2^{k-j} square tiles of size $(2^j + 1) \times (2^j + 1)$. Consequently, the total number of tiles at scale j is $2^{2(k-j)}$. A beam curve connects two pixels on the perimeter of a tile, and we exclude curves that connect two pixels that lie on the same side of a tile. The number of beam curves in a tile therefore is $2(2^j+1) \times 3(2^j+1) = 6(2^j+1)^2 \approx 3 \cdot 2^{2j+1}$. The total number of beam curves at scale j is $2^{2(k-j)} \times 3 \cdot 2^{2j+1} = 3 \cdot 2^{2k+1} \approx 6N$. The total number of beam curves at all scales is $\sum_{j=0}^{k} 6N \approx 6N \log N$.

Next we analyze the time complexity of constructing the beam curves. We begin with the general curves. At every scale j we construct a beam curve by connecting up to four sections in either clockwise or counter-clockwise order. The cost of connecting four sections in both orientations is $2(2^{j-1}+1)^3 \approx 2^{3(j-1)+1}$. Since the total number of beam curves at scale j is $6N$ we get $6N \times 2^{3(j-1)+1} = 3N \cdot 2^{3j-1}$. Summing this over all scales we get the complexity $1.5N \sum_{j=0}^{k} 2^{3j} \approx 1.5N \cdot 2^{3(k+1)} = 12N \cdot 2^{3k} = 12N^{5/2}$.

For the monotone curves at each scale j we construct a beam curve by connecting up to three sections in either clockwise or counter-clockwise order. The cost of connecting three sections in both orientations is $2(2^{j-1}+1)^2 \approx 2^{2(j-1)+1}$. Since the total number of beam curves at scale j is $6N$ we get $6N \times 2^{2(j-1)+1} = 3N \cdot 2^{2j}$. Summing this over all scales we get the complexity $3N \sum_{j=0}^{k} 2^{2j} \approx 3N \cdot 2^{2(k+1)} \approx 12N^2$.

Spatial Statistics of Visual Keypoints for Texture Recognition

Huu-Giao Nguyen, Ronan Fablet, and Jean-Marc Boucher

Institut Telecom / Telecom Bretagne / LabSTICC
CS 83818 - 29238 Brest Cedex 3 - France
Université européenne de Bretagne
{huu.nguyen,ronan.fablet,jm.boucher}@telecom-bretagne.eu

Abstract. In this paper, we propose a new descriptor of texture images based on the characterization of the spatial patterns of image keypoints. Regarding the set of visual keypoints of a given texture sample as the realization of marked point process, we define texture features from multivariate spatial statistics. Our approach initially relies on the construction of a codebook of the visual signatures of the keypoints. Here these visual signatures are given by SIFT feature vectors and the codebooks are issued from a hierarchical clustering algorithm suitable for processing large high-dimensional dataset. The texture descriptor is formed by cooccurrence statistics of neighboring keypoint pairs for different neighborhood radii. The proposed descriptor inherits the invariance properties of the SIFT w.r.t. contrast change and geometric image transformation (rotation, scaling). An application to texture recognition using the discriminative classifiers, namely: k-NN, SVM and random forest, is considered and a quantitative evaluation is reported for two case-studies: UIUC texture database and real sonar textures. The proposed approach favourably compares to previous work. We further discuss the properties of the proposed descriptor, including dimensionality aspects.

1 Introduction

The analysis of the texture content of images is among the critical issues for numerous application domains ranging from multimedia applications including content-based image and video indexing [1], automated scene analysis, archaeology [2] to more environment-oriented domains such as geosciences and remote sensing [3]. From the early 1970s, numerous advances have been reported in the definition of efficient while compact descriptors of visual textures. In the literature, the analysis of image textures initially mostly relied on a statistical analysis. Visual textures were regarded as the realization of random fields which could be characterized from relevant statistics such as covariance statistics [4,5], co-occurrence statistics [6,7,8,9] or statistics of the response to scale-space filters such as Gabor and wavelet analysis [10].

More recently, a renewed interest in texture analysis emerged from the development of texture descriptors invariant to geometric and photometric transformations of the images. Visual keypoints initially proposed for object recognition

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 764–777, 2010.

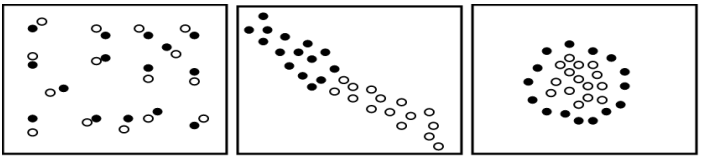


Fig. 1. Examples of difference spatial distributions of two marked-points

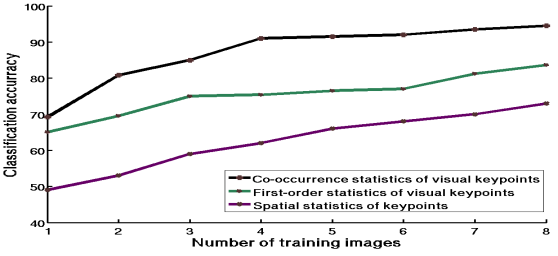


Fig. 2. Relevance of the combination of visual signatures of the keypoints to descriptors of their spatial organization: correct classification rate as a function of the number of training images for the UIUC texture database using only spatial statistics of the keypoint set (magenta), only statistics of the visual signatures of the keypoints (green), second-order ccoccurrence statistics of the visual keypoint set (black) jointly describing the visual content and the spatial organization.

were shown to be particularly efficient for texture analysis [11,12]. The resulting texture characterization inherits the robustness of visual keypoints in terms of invariance to geometric image distortions and contrast change. Among the most popular descriptors is the SIFT descriptor based on the characterization of gradient orientations in scale invariant regions [13]. The application to texture recognition generally consists in learning classification models from these signatures of the visual keypoints [11,12].

This typical approach however generally ignores the spatial organization of the visual signatures. As sketched by Fig.1, for similar relative occurrences of visual signatures, different spatial patterns revealing differences in visual content of the textures may be observed. In this work, we aim at jointly characterizing the local visual signatures of the textures and their spatial layout. Formally, we exploit spatial statistics to propose novel texture descriptors. Here, a texture is regarded as the realization of 2D marked point process referring to the set of the visual keypoints along with the associated visual descriptors, e.g., SIFT feature vectors. Texture descriptors are extracted as second-order co-occurrence statistics of the multivariate 2D point process. Applied to supervised texture recognition for two different texture databases, UIUC textures and real sonar textures, these descriptors favorably compare to previous work [7,8,9,10,11]. As illustrated by Fig.2, we show the relevance of the combination of statistics of the visual signatures to statistics of the spatial layout of the keypoints.

This paper is organised as follows. In Section 2, an overview of the proposed approach is described. We present in Section 3 the proposed textural features based on spatial statistics of visual keypoints. Section 4 discusses the application to texture recognition. In Section 5, texture recognition performances are reported for two texture databases. The main contributions of proposed approach with respect to previous work are further discussed in Section 6.

2 Overview of Proposed Method

The sketch of the proposed approach is reported in Fig.3. A texture is regarded as the realization of a marked point process of visual keypoints. Descriptive statistics of this marked point process will form a set of texture descriptors.

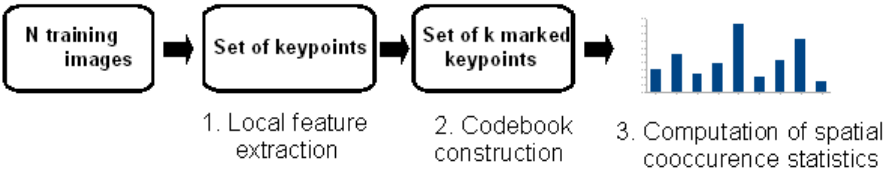


Fig. 3. Principal steps of our method for the extraction of a texture descriptor jointly characterizing the spatial layout of visual keypoints and their visual signatures

The initial step consists in detecting and characterizing visual keypoints. Various local descriptors have been proposed for object and texture recognition. The SIFT descriptor achieves a good and stable performance in the domain of texture classification [14]. SIFT keypoints [13] correspond to local extrema of difference-of-Gaussian (DoG) filters at different scales. Each pixel in the DoG images is compared to its 26 neighbors in a 3x3x3 neighborhood that spans adjacent DoG images. Each keypoint is characterized by the distribution of the orientations of the gradient of the intensity in the sixteen 4x4 windows around the considered point. This description ensures contrast invariance and partial invariance to affine transform. Orientation being quantized over eight values, the resulting SIFT feature vector is 128-dimensional, denoted by $X(s_i) = \{x_{i1}, ..., x_{i128}\}$. As an output of this first step, a texture sample can be regarded as the realization of a marked point process, corresponding to the extracted keypoints with marks given by the SIFT feature vectors.

Our goal is to characterize the spatial patterns formed by the visual keypoints for a given texture image. To this end, we consider the joint density that pairs of neighboring keypoints s_1 and s_2 occur with given visual signatures x and y

$$p\left(X(s_1) = x, X(s_2) = y, s_2 \in V(s_1)\right), \tag{1}$$

where $V(s_1)$ specifies the neighborhood of point s_1. This density is parameterized by the visual signatures and the neighborhood structure. Considering

different neighborhood sizes, this density conveys second-order information on the relations between the visual signatures and their spatial layout.

The dimensionality of the visual signatures, i.e. the 128-dimensional SIFT vectors, makes untractable the non-parametric computation of the above second-order co-occurrence density. Rather than investigating parametric models such as Poisson point processes [15], a non-parametric approach based on an initial adaptive quantization of the SIFT feature space is proposed. We first build a codebook of visual keypoints from their feature vetors using a k-means-like method. Any visual keypoint is then associated with a discrete mark corresponding to the assigned category of visual signatures, denoted by $m_{s_i} = p$ where $p = \{1, ..., k\}$. We then resort to a discrete approximation of the continuous density (Eq.1) for cooccurrences of visual words. The resulting second-order cooccurrence statistics are exploited for texture recognition.

The different steps of this approach are detailed in the subsequent.

3 Co-occurrence Statistics of Visual Keypoints

Modeling and characterizing spatial point patterns is an active area of research, especially in environment-related sciences [16]. The general goal is to reveal underlying phenomena from the statistical analysis of some spatially-referenced observations or measures. In our case, a set of spatial points with some associated discrete signatures is regarded as the realization of a multivariate 2D point process, for which relevant descriptive statistics should be defined.

3.1 Spatial Point Process

A *spatial point process* $\mathbb{S}$ is defined as a locally finite random subset of a given bounded region $B \subset \mathbb{R}^2$, and a realization of such a process is a spatial point pattern $s = \{s_1, ..., s_n\}$ of n points contained in B. Considering a realization of a point process, the moments of the random variable are meaningful descriptive quantities, such as the expected number $\mu(B)$ of points falling in region B [17]. This first-order moment is associated with the intensity measure ρ of $\mathbb{S}$:

$$\mu(B) = E\{\#B\} = \int_B \rho(s)ds \tag{2}$$

where $E\{.\}$ denotes the number of expected points falling in B, $\rho(s)ds$ is the probability that one point falls in an infinitesimally small area ds around s. The normalized first-order moment $K() = \mu(B)/|B|$, where $|B|$ is the area of region B, is a popular descriptive spatial statistics, known as Ripley's K-function [16].

Actually, the organization of spatially stationary point process is only provided by the higher-order moments. The covariance structure of the count variables is measured by the second-order factorial moment $\mu^{(2)}$ of $\mathbb{S}$:

$$\mu^{(2)}(B_1 \times B_2) = E\{ \sum_{s_1, s_2 \in S}^{\#} \mathbb{I}_{B_1}(s_1)\mathbb{I}_{B_2}(s_2)\} = \int_{B_1 \times B_2} \rho^{(2)}(s_1, s_2)ds_1 ds_2 \tag{3}$$

where $\sum^{\#}$ is the sum runs only over pairs of points, $\mathbb{I}[.]$ is an indicator function that takes the value 1 when s_i falls in the region B_i. Second-order density $\rho^{(2)}(s_1, s_2)ds_1ds_2$ is interpreted as the density of the pair of points s_1 and s_2 in infinitesimally small areas ds_1 and ds_2. Density function $\rho^{(2)}$ states the correlation of pairs of points [5] . Considering a stationary and translation-invariant point process, density $\rho^{(2)}(s_1, s_2)$ only depends on distance $\|s_1 - s_2\|$.

3.2 Descriptive Statistics of the Spatial Patterns of Visual Keypoints

The above second-order moment of spatial point process can be extended to multivariate spatial point patterns. In our case, each visual keypoint is associated with a discrete mark m_{s_i}. Let $\Psi = \{s_i; m_i\}$ be a multivariate marked point process. Extending (Eq.3), the second-order characteristics of Ψ are characterized by the factorial moment measure $\mu_f^{(2)}(B_1 \times B_2)$ for regions B_1 and B_2:

$$\mu_f^{(2)}(B_1 \times B_2) = E\{ \sum_{[s_1;m_1],[s_2;m_2]\in\Psi}^{\#} f(m_1, m_2)\mathbb{I}_{B_1}(s_1)\mathbb{I}_{B_2}(s_2)\} \tag{4}$$

where f is an arbitrary measurable non-negative function. Following [17], considering a translation-invariant and stationary point process, spatial sets $B_1 \times B_2$ are parameterized as $\{s_1 \in B\} \times \{s_2 \in D(s_1, r)\}$ where $D(s_1, r)$ is the disk of center s_1 and radius r. Focusing on cooccurrence statistics, i.e. function f set as $\delta_i().\delta_j()$ with i, j mark indices and δ the Kronecker function, we resort to the following second-order cooccurrence moment:

$$\mu_{ij}^{(2)}(., r) = E\{ \sum_{[s_1;m_1],[s_2;m_2]\in\Psi}^{\#} \delta_i(m_1).\delta_j(m_2)\mathbb{I}(\|s_1 - s_2\| \leq r)\} \tag{5}$$

Function $\mu_{ij}^{(2)}(., r)$ states the number of points of type j in a disk of radius r centered at a point of the spatial pattern of type i. These statistics express the covariance structure of the spatial pattern for points of type i and j (Fig.4a).

In practice, to exploit the above second-order descriptive statistics of keypoints in image, edge effects have to be dealt with. The area of the analysis region affects the accurate counting-based estimation of the spatial statistics. Several corrections for edge effects for points located near the boundary of the image have been proposed in the literature[16,18].

We then consider a normalized version of the second-order cooccurrence statistics (Eq.5) denoted by $\Gamma_{ij}^I(r)$ whose estimate is given by:

$$\Gamma_{ij}^I(r) = \frac{1}{N}\sum_{p=1}^{N}\frac{1}{A(s_p)}\sum_{j=1,j\neq i}^{N}\delta_i(m_p)\delta_j(m_q)\mathbb{I}[\|s_p - s_q\| \leq r] \tag{6}$$

where $A(s_p)$ is the actual area of the circular study region of radius r for keypoint s_p with p, q mark indices, N is the total number of keypoints in the image.

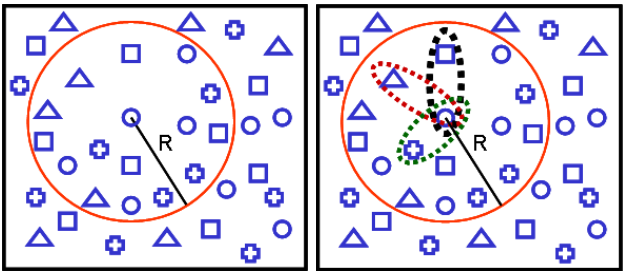


Fig. 4. Second-order spatial statistics for multivariate point process: (a) spatial statistics for any pair of keypoints within a circular region centered at a point of the pattern, (b) cooccurrence statistics for pairs of keypoints associated with given visual categories within a circular region centered at a point of the pattern.

3.3 Scaling Effects

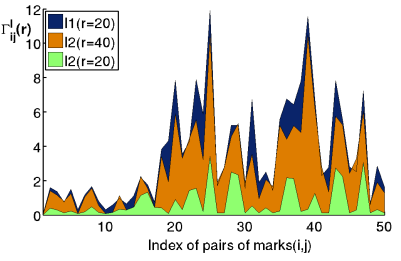
The radius value in the computation of the proposed second-order cooccurrence statistics can be viewed as a scale-space parameter for the analysis of the spatial patterns of visual keypoints. The effects of image scaling should then be further analyzed to reach invariance to such image transformations.

Considering two images of a given texture sample at two different scales α_1 and α_2, the first-order moments of the point process in these two images are related as follows:

$$\mu_1(r_1) \approx \mu_2(r_2) \tag{7}$$

$$\pi(\alpha_1 r)^2.K_1(\alpha_1 r) \approx \pi(\alpha_2 r)^2.K_2(\alpha_2 r) \tag{8}$$

with $r_1 = \alpha_1 r$ and $r_2 = \alpha_2 r$ and r the reference radius at a reference scale set to 1. As the detection of the visual keypoints in the images is scale-invariant, this property should be held in practice in our case. Interestingly, for an infinite circular region, the first-order moments of the two texture samples refer to the average point densities per surface unit and they only differ up to a factor depending on the rate α_1/α_2. This property then provides the mean for scale adaption. We proceed as follows. We first set a reference scale, i.e. $\alpha_1 = 1$ corresponding to expected mean point density per surface unit $K_1(\infty)$. For any texture sample, we estimate the associated scale factor α_2 with respect to the reference scaling from relation (Eq.8). The texture descriptor is then formed by the normalized second-order cooccurrence statistics for radius values $\{\alpha_2 r_i\}$, where the r_i's are predefined radius values at the reference scale. In practice, the r_i's are set according to a logarithmic sampling. Fig.5 clearly illustrates the benefit for the scale adaption in the computation of the cooccurrence statistics when comparing two texture samples differing by their scale factor.



(c)

Fig. 5. Scaling effect on second-order cooccurrence statistics: images of the same texture at two scale factors (left), comparison of cooccurrence statistics without (blue vs. green) and with (orange vs. blue) scale adaption for the computation of the second-order cooccurrence statistics (right). The plot depicts the value of $\Gamma_{ij}(r)$ as a function of an index assigned to the pair of discrete marks (i, j).

3.4 Dimensionality of the Feature Space

Given a codebook of keypoints with k visual words, the feature vector issued from the cooccurrence statistics in Eq.6 is $N_r k^2$-dimensional, where N_r is the number of radius r of circular study region. For instance, considering a codebook of 40 visual words and $N_r = 20$, the feature space is 32000-dimensional. Such a high-dimensional description may affect recognition performance.

To address this issue, we suggest determining categories of pairs of visual keypoints to reduce the dimension of the second-order cooccurrence statistics. The codebook of keypoint pairs is issued from the application of a classical k-means method to the set of pairs of categorized keypoints within the training images. The size k^* of this codebook of pairs of visual keypoints is chosen to be typically in the order of k, the size of the codebook of visual words. Let us denote by $M(s_i, s_j) = \{1, ..k^*\}$ the category assigned to the pair of keypoints s_i and s_j. The second-order cooccurrence statistics (Eq.6) are then computed as:

$$\Gamma^I_u(r) = \frac{1}{N} \sum_{p=1}^{N} \frac{1}{A(s_p)} \sum_{j=1, j\neq i}^{N} \delta_u(M(s_i, s_j)) \mathbb{I}[\|s_p - s_q\| \leq r] \tag{9}$$

for a given category u of the codebook of pairs of visual keypoints. This procedure can be regarded as an adapted quantization of the normalied cooccurrence statistics defined by Eq.(1)

4 Texture Classification

Given the textural features defined in the previous section, supervised texture classification is addressed using k nearest-neighbor (k-NN), support vector machines (SVM) and random forest (RF) classifiers:

- Nonparametric k-NN classifier is considered in this work because of its simplicity and its computational efficiency.

Table 1. Three different similarity measures, where $m_i = \frac{h_i + k_i}{2}$

Euclidean distance	χ^2 distance	Jeffrey divergence
$d(H,K) = \sum_i \lvert h_i - k_i \rvert^2$	$d(H,K) = \sum_i \frac{(h_i - m_i)^2}{m_i}$	$d(H,K) = \sum_i \left(h_i log \frac{h_i}{m_i} + k_i log \frac{k_i}{m_i} \right)$

- Regarding SVM classifiers [19], a one-versus-all strategy is exploited to train a multi-class SVM with a Gaussian kernel.
- RF classifier relies on the construction of an ensemble of classification trees using some form of randomization. A texture sample is classified by sending it down every tree and aggregating the reached leaf distributions. RF classifier uses a voting rule to assign a class to an unknown texture sample [20].

SVM and k-NN classifiers here require the definition of a similarity measure in the considered feature space. Three different distances reported in Tab.1 accounting for the characteristics of the cooccurrence statistics are investigated.

Randomization-based learning strategy: Randomization-based machine learning procedures are particularly appealing for classification from very large training dataset and high-dimensional feature space [19,20]. In our case, a huge dataset of keypoints is used as input for the construction of the codebook of keypoint signatures from their SIFT feature vectors. For instance, a typical texture image involves 4.000 keypoints and 100 640x480 UIUC texture images leads to over 4.10^5 samples points in the 128-dimensional feature space defined by the SIFT descriptors. Adapted clustering techniques are required to perform the initial determination of the categories of visual keypoints. We suggest using a hierarchical clustering algorithm [21]. It relies on an agglomerative algorithm to generate a clustering solution. A hierarchical subtree is first built for each cluster. It then re-agglomerates these clusters to build a final hierarchical tree.

The initial clustering of visual keypoints might be regarded as a critical step in the proposed procedure. We can turn it into an advantage to build a randomized set of classifiers and improve the robustness to this initial clustering step. We proceed as follows. First, we carry out a random subsampling of visual keypoints within the training dataset to select a subset tractable for the clustering scheme. Typically, 10^5 sample keypoints and up to 150 clusters can be considered. Given extracted categories of visual keypoints, second-order cooccurrence statistics are computed for a predefined set of radius values for each texture sample in the training set and a classifier is trained in the resulting feature space. From repeated initial random subsampling steps, a randomized ensemble of classifiers is built. Regarding the recognition step with a trained classifier ensemble, the classification of an unknown texture results from a simple voting rule.

5 Experiments

5.1 UIUC Texture Classification

The first experiment relies on 1000 640x480 texture images of UIUC database. This database involves 25 texture classes and each class contains 40 images

with strongly varying viewpoint, scale and illumination conditions. Examples are reported in Fig.5a. The evaluation involves the computation of classification performances for model learning with N_t training texture samples per class. Training images are randomly selected among the 40 samples per class. The remaining $40-N_t$ images per class are used as test images. The random selection of training samples is repeated 200 times to evaluate the mean and the standard deviation of the correct classification rate.

For comparison purposes, a set of texture descriptors such as Gabor filter [10], co-occurrence matrix [9], spatial statistics of the keypoints [15], bag-of-keypoints(BoK) [22], Zhang's method [11], Xu's method [7] and Varma's method [8] were selected to evaluate the relevance of our contribution compared to the state-of-the-art techniques. The results on UIUC database of Varma's method was reported in [8]. For the other methods, we report the performance with the best parameter setting. BoK was tested with $k = \{60, 120, 150\}$ classes. For cooccurrence matrices, the following neighborhood types were considered: [0,1], [1,1]. Gabor features were computed for the frequencies $f = \{0, 4, 8\}$ and the orientation $\theta = \{0, \frac{\pi}{3}, \frac{\pi}{2}, \frac{3\pi}{4}\}$. We also tested different parameter settings for Xu's method: density level $ind = \{1, 8\}$, dimension of MFS $f = \{26, 64\}$ and iteration level $ite = \{8, 10\}$. SVM classifiers and Jeffrey divergence are used.

The parameter setting for our approach is as follows. A set of 10^5 random sampling keypoints is exploited for each hierarchical clustering step. The numbers of categories of visual keypoints k and of visual keypoint pairs k^* are respectively set to $k = 150$ and $k^* = 60$. $N_{r_{ref}} = 10$ circular study regions with reference radius values $r_{ref} = \{10, 20, 40, 60, 80, 120, 150, 190, 220, 240\}$ are considered for the computation of the second-order cooccurrence statistics. In Tab.2, we report the following results for our approach: cooccurrence statistics $\Gamma^I_{ij}(r)$ in Eq.6, cooccurrence statistics with dimensional feature reduction $\Gamma^I_u(r)$ in Eq.9.

Mean classification rates and standard deviations over 200 random selections are reported in Tab.2 as a function of the number of training images. The proposed descriptor favourably compares to the other approaches. Observing the result in the case of 20 training samples, our proposed method reaches up to

Table 2. Classification rates and standard deviations over 200 random selections on UIUC texture database

N_t	1	5	10	15	20
Gabor filter	31.22 ± 3.14	45.14 ± 2.54	57.37 ± 1.93	61.25 ± 1.52	67.78 ± 1.28
Cooccurrence matrix	44.17 ± 2.93	62.25 ± 2.34	70.33 ± 1.75	73.67 ± 1.53	79.17 ± 1.37
Spatial statistic	48.69 ± 2.85	69.25 ± 2.45	75.42 ± 1.66	82.66 ± 1.28	87.34 ± 1.34
BoK	67.25 ± 2.75	76.38 ± 2.15	81.12 ± 1.45	86.35 ± 1.20	91.28 ± 1.15
Xu [7]	61.14 ± 2.90	$83.33 \pm 2.07.$	89.68 ± 1.65	91.34 ± 1.45	93.85 ± 1.31
Varma [8]	–	85.35 ± 1.69	91.64 ± 1.18	94.09 ± 0.98	95.40 ± 0.92
Zhang [11]	72.53 ± 2.45	88.62 ± 1.33	93.17 ± 1.15	95.33 ± 0.98	96.67 ± 0.93
$\Gamma^I_{ij}(r)$	75.43 ± 2.65	89.22 ± 1.47	93.48 ± 0.98	96.21 ± 0.66	97.17 ± 0.42
$\Gamma^I_u(r)$	75.66 ± 1.65	91.67 ± 0.93	94.33 ± 0.78	96.54 ± 0.53	97.34 ± 0.25

97.34% of correct classification. Our descriptor gets a gain greater than 6% compared with BoK and than 3% compared with local fractal feature extraction methods. Our descriptor is shown to be slightly more robust and stable than Zhang's method , 97.34 ± 0.25 w.r.t. 96.67 ± 0.93. Greater gain in performances are observed when only few training images are used. From 5 training images per class, the proposed approach gets a correct classification greater than 91.67% whereas all the other methods are below 88.62%. Besides, while reducing the computational complexity, the dimension reduction technique also leads to a more robust texture recognition when few training samples are available.

5.2 Real Sonar Textures

Sonar imaging provides a remote sensing tool to observe and characterize the physical properties of the seafloor and is increasingly used for a variety of applications such as environmental monitoring, marine geosciences and ecology, as well as oil industry or defense [3,23]. Sonar images are issued from the measurements of the backscattered echo of the seabed for successive sonar swaths corresponding to various incidence angles. In Fig.6a, an example of sidescan sonar images with incident angles from -85° to +85° was obtained from a DF1000 sonar. The different seabeds here correspond to different textural features. For a given seabed type, the mean backscatter clearly depends on the incidence angle [3,23]. Especially, for vertical incidences, poor discrimination among seabed types can be expected. Besides, textural patterns may also vary depending on incidence angles as shown in Fig.6(b,c), where in the specular domain [5°, 40°] a loss in contrast is observed for sand ripples compared with the sector [80°, 85°].

We used a database of 180 sonar texture images involving 6 different seabed classes which are extracted from sidescan sonar images, e.g. Fig.6. Each class comprises 30 texture images. Sonar texture samples are 256 x 256 images with strong variations of incidence angles, scale and illumination conditions. We randomly choose $N = \{1, 5\}$ samples of each class to build a training dataset, the

(a) [5°,40°](b) [80°,85°](c)

Fig. 6. A sidescan sonar image(a) and textures of a seabed type sample of sand ripples for two angular sectors(b,c) (Rebent, IFREMER)

Table 3. Correct classification rate for the sonar texture database using second-order cooccurrence statistics (Eq.10) with different classifiers (k-NN, SVM, RF) and similarity measures (Euclidean,χ^2,Jeffrey divergence).

	k-NN			SVM			RF
	Euclidean	χ^2	Jeffrey	Euclidean	χ^2	Jeffrey	default
3 samples	89.2%	88.9%	91.3%	88.7%	89.4%	91.6%	91.8%
5 samples	94.2%	93.6%	95.1%	93.8%	94.7%	96.4%	96.4%

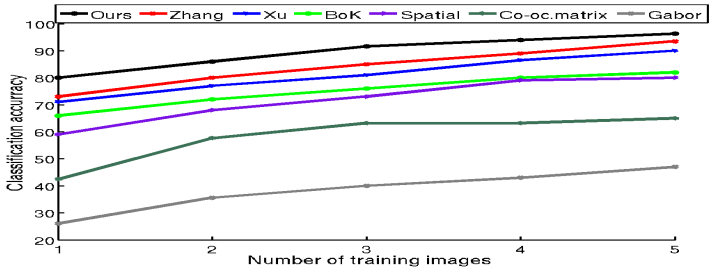


Fig. 7. Average classification rate on sonar images database

other images being used as the query images. For these experiments, the following parameter setting is used: k=50, $r = \{10, 20, 40, 80, 120\}$ and $k^* = 35$. For comparison purposes, except for Varma's method, we used all methods similarly to the experiment conducted with the UIUC texture database.

We first compare the performances of the different classifiers and similarity measures. Jeffrey divergence improves the classification performance with an approximate gain greater than 2% compared to χ^2 and Euclidean distance when 3 (or 5) training images are considered in Tab.3. Differences in the performance among the classifiers, k-NN, SVM and RF, are not obvious for this dataset. However, RF may be slightly more robust and stable than k-NN or SVM.

Regarding the comparison of the proposed descriptor to previous work, the mean correct classification rate is reported for each approach as a function of the number of training samples (Fig.7). The proposed method reaches up to 96.4% of correct classification when 5 training images are considered. It favourably compares to previous work for which the best score is 93.5%(Zhang's method). More than 6% of correct classification is gained when one image per class is used for training. These results demonstrate the robustness of the proposed descriptor to geometric and photometric image distortions.

6 Discussion and Future Developments

We have proposed a novel texture descriptor based on the statistical characterization of the spatial patterns of visual keypoints in texture images. The key feature of our approach is its joint description of the visual signatures of the

texture and of their spatial organization. We further discuss these contributions with respect to previous work.

- *Spatial information for texture recognition:* Numerous previous studies have investigated the use of spatial information for texture description [7,8,9]. The focus is generally given to the characterization of local texture patterns in a neighbourhood of a point. As an example Varma et al.[8] and Xu et al. [7] proposed a local fractal feature based on the analysis of the distribution of intensities as a function of the distance to a center point. In contrast, our approach encodes spatial information not at a local scale but at some object-related scale. The proposed descriptors aim at describing the spatial patterns formed by the set of visual keypoints, not the spatial variations of the intensity in a neighboorhood of each keypoint.

These two aspects should be regarded as complementary. In the reported results, key points extracted as local DOG extrema and visual signatures provided by the SIFT descriptor were considered. It should be noted that any other local visual signature, including for instance the local fractal feature, could be considered in a similar manner. In future work, the evaluation of the robustness and the distinctiveness of other texture descriptors may be investigated such as GLOH[14], SURF[24], DAISY[25], CS-LBP (see [14,26] for a review).

- *Image modelling from spatial point processes:* It might be noted that spatial point process were previously investigated for texture analysis. For instance, Linnett et al.[15] modeled the spatial distribution of the grey-levels by a 2D Poisson process. More recently, cooccurrence statistics of visual keypoints were investigated for different applications, namely scene categorization [27,28], robot navigation [29]. However, those methods are again aimed at characterizing the variations of image intensities in local neighbourhoods. It should be noted that the link to spatial statistics also provide an unbiased estimation of the cooccurrence statistics based on a correction of edge effects.

Recently Gibbs point process models have also been applied to extract geometric objects in texture images by Lafarge et al. [30]. The Gibbs model provide an a priori model for the spatial organization of elementary geometric objects. Such a model cannot be applied to texture recognition. In contrast, our approach aims at deriving a feature vector for texture recognition based on spatial statistics. It should be noted that considered spatial statistics can be regarded as the sufficient statistics defining a multivariate log-Gaussian Cox process [5].

- *Invariance and dimensionality issues:* Local descriptors have emerged as a powerful tool for invariant texture characterization and classification compared to the early feature developed for texture analysis such as Gabor features and cooccurrence matrices. Approaches based on descriptive statistics of a set of visual keypoints may benefit from the robustness of their visual signatures in terms of invariance to photometric and geometric image transformations while providing a more compact representation of the information. In that case, each texture image is associated with a feature vector such that the size of the training database equals the number of training images. The BoK method, i.e. the distribution of the occurrences of the visual words in each texture sample, is a first

example. As reported here, the associated recognition performances are however degraded compared to the classical keypoint classifier. The method described here also falls in this category. In contrast to BoK, both the visual signatures and the spatial patterns are characterized from second-order cooccurrence statistics. These statistics could convey scale-invariance. Improved texture recognition performances were initially obtained at the expense of the dimensionality of the feature space compared to BoK, i.e. $N_r k^2$ vs. k where N_r is the number of circular analysis region and k and the number of visual words. We have shown that dimensionality could be downsized up to the range of $N_r k$ simultaneously to a more robust recognition of texture sample.

In future work, other lower-dimensional representation could be investigated, especially from parametric and semi-parametric models of the covariance functions underlying the considered second-order cooccurrence statistics [5].

References

1. Ndjiki-Nya, P., Makai, B., Blattermann, G., Smolic, A., Schwarz, H., Wiegand, T.: A content-based video coding approach for rigid and non-rigid textures. In: IEEE Conf. on Im. Proc., ICIP, pp. 3169–3172 (2006)
2. Zalesny, A., der Maur, D.A., Paget, R., Vergauwen, M., Gool, L.V.: Realistic textures for virtual anastylosis. In: CVPR Workshop, vol. 1, pp. 14–20 (2003)
3. Karoui, I., Fablet, R., Boucher, J.M., Augustin, J.M.: Seabed segmentation using optimized statistics of sonar textures. IEEE Trans. on Geos. and Rem. Sens. 47(6), 1621–1631 (2009)
4. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC, pp. 384–393 (2002)
5. Møller, J., Syversveen, A., Waagepetersen, R.: Log gaussian cox processes. Scandinavian Journal of Statistics 25(3), 451–482 (1998)
6. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, B.: Discovering objects and their location in images. In: ICCV, vol. 1, pp. 370–377 (2005)
7. Xu, Y., Ji, H., Fermuller, C.: Viewpoint invariant texture description using fractal analysis. IJCV 83(1), 85–100 (2009)
8. Varma, M., Garg, R.: Locally invariant fractal features for statistical texture classification. In: ICCV, vol. 1, pp. 1–8 (2007)
9. Haralick, R.: Statistical and structural approaches to textures. Proceedings of the IEEE 67, 786–804 (1979)
10. Randen, T., Husøy, J.H.: Filtering for texture classification: A comparative study. IEEE Trans. on PAMI 21, 291–310 (1999)
11. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. IJCV 73, 213–238 (2007)
12. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. IEEE Trans. on PAMI 27, 1265–1278 (2005)
13. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
14. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. on PAMI 27(10), 1615–1630 (2005)

15. Linnett, L., Carmichael, D., Clarke, S.: Texture classification using a spatial-point process model. IEE Vision, Image and Signal Processing 142, 1–6 (1995)
16. Goreaud, F., Pélissier, R.: On explicit formulas of edge effect correction for ripley's k-function. Journal of Vegetation Science 10, 433–438 (1999)
17. Stoyan, D., Stoyan, H.: Fractals, random shapes and point fields. Wiley, Chichester (1994)
18. Nguyen, H.-G., Fablet, R., Boucher, J.-M.: Invariant descriptors of sonar textures from spatial statistics of local features. In: ICASSP, pp. 1674–1677 (2010)
19. Kotsiantis, S., Zaharakis, I., Pintelas, P.: Machine learning: a review of classification and combining techniques. Artificial Intelligence Review 26, 159–190 (2006)
20. Breiman, F.: Random forests. Machine learning 45, 5–32 (2001)
21. Zhao, Y., Karypis, G.: Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery 10, 141–168 (2005)
22. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3024, pp. 1–22. Springer, Heidelberg (2004)
23. Chenadec, G.L., Boucher, J.M., Lurton, X.: Angular dependence of k-distributed sonar data. IEEE Trans. on Geos. and Rem. Sens. 45, 1124–1235 (2007)
24. 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)
25. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide baseline stereo. IEEE Trans. on PAMI (2009)
26. Heikkilä, M., Pietikäinen, M., Schmid, C.: Description of interest regions with local binary patterns. Pattern Recognition 42(3), 425–436 (2009)
27. Ling, H., Soatto, S.: Proximity distribution kernels for geometric context in category recognition. In: ICCV, pp. 1–8 (2007)
28. Savarese, S., Winn, J.M., Criminisi, A.: Discriminative object class models of appearance and shape by correlatons. In: CVPR, vol. 2, pp. 2033–2040 (2006)
29. Cummins, M., Newman, P.: FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. IJRR 27, 647–665 (2008)
30. Lafarge, F., Gimel'farb, G., Descombes, X.: Geometric feature extraction by a multi-marked point process. IEEE Trans. on PAMI 99(1) (2009)

BRIEF: Binary Robust Independent Elementary Features

Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua

EPFL, Lausanne, Switzerland
firstname.lastname@epfl.ch

Abstract. We propose to use binary strings as an efficient feature point descriptor, which we call BRIEF. We show that it is highly discriminative even when using relatively few bits and can be computed using simple intensity difference tests. Furthermore, the descriptor similarity can be evaluated using the Hamming distance, which is very efficient to compute, instead of the L_2 norm as is usually done.

As a result, BRIEF is very fast both to build and to match. We compare it against SURF and U-SURF on standard benchmarks and show that it yields a similar or better recognition performance, while running in a fraction of the time required by either.

1 Introduction

Feature point descriptors are now at the core of many Computer Vision technologies, such as object recognition, 3D reconstruction, image retrieval, and camera localization. Since applications of these technologies have to handle ever more data or to run on mobile devices with limited computational resources, there is a growing need for local descriptors that are fast to compute, fast to match, and memory efficient.

One way to speed up matching and reduce memory consumption is to work with short descriptors. They can be obtained by applying dimensionality reduction, such as PCA [1] or LDA [2], to an original descriptor such as SIFT [3] or SURF [4]. For example, it was shown in [5,6,7] that floating point values of the descriptor vector could be quantized using very few bits per value without loss of recognition performance. An even more drastic dimensionality reduction can be achieved by using hash functions that reduce SIFT descriptors to binary strings, as done in [8]. These strings represent binary descriptors whose similarity can be measured by the Hamming distance.

While effective, these approaches to dimensionality reduction require first computing the full descriptor before further processing can take place. In this paper, we show that this whole computation can be shortcut by *directly* computing binary strings from image patches. The individual bits are obtained by comparing the intensities of pairs of points along the same lines as in [9] but without requiring a training phase. We refer to the resulting descriptor as BRIEF.

Our experiments show that only 256 bits, or even 128 bits, often suffice to obtain very good matching results. BRIEF is therefore very efficient both to

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 778–792, 2010.

compute and to store in memory. Furthermore, comparing strings can be done by computing the Hamming distance, which can be done extremely fast on modern CPUs that often provide a specific instruction to perform a XOR or bit count operation, as is the case in the latest SSE [10] instruction set.

This means that BRIEF easily outperforms other fast descriptors such as SURF and U-SURF in terms of speed, as will be shown in the Results section. Furthermore, it also outperforms them in terms of recognition rate in many cases, as we will demonstrate using benchmark datasets.

2 Related Work

The SIFT descriptor [3] is highly discriminant but, being a 128-vector, is relatively slow to compute and match. This can be a drawback for real-time applications such as SLAM that keep track of many points as well as for algorithms that require storing very large numbers of descriptors, for example for large-scale 3D reconstruction.

There are many approaches to solving this problem by developing faster to compute and match descriptors, while preserving the discriminative power of SIFT. The SURF descriptor [4] represents one of the best known ones. Like SIFT, it relies on local gradient histograms but uses integral images to speed up the computation. Different parameter settings are possible but, since using only 64 dimensions already yields good recognition performances, that version has become very popular and a *de facto* standard. This is why we compare ourselves to it in the Results section.

SURF addresses the issue of speed but, since the descriptor is a 64-vector of floating points values, representing it still requires 256 bytes. This becomes significant when millions of descriptors must be stored. There are three main classes of approaches to reducing this number.

The first involves dimensionality reduction techniques such as Principal Component Analysis (PCA) or Linear Discriminant Embedding (LDE). PCA is very easy to perform and can reduce descriptor size at no loss in recognition performance [1]. By contrast, LDE requires labeled training data, in the form of descriptors that should be matched together, which is more difficult to obtain. It can improve performance [2] but can also overfit and degrade performance.

A second way to shorten a descriptor is to quantize its floating-point coordinates into integers coded on fewer bits. In [5], it is shown that the SIFT descriptor can be quantized using only 4 bits per coordinate. Quantization is used for the same purpose in [6,7]. It is a simple operation that results not only in memory gain but also in faster matching as computing the distance between short vectors can then be done very efficiently on modern CPUs. In [6], it is shown that for some parameter settings of the DAISY descriptor, PCA and quantization can be combined to reduce its size to 60 bits. However, in this approach the Hamming distance cannot be used for matching because the bits are, in contrast to BRIEF, arranged in blocks of four and hence cannot be processed independently.

A third and more radical way to shorten a descriptor is to binarize it. For example, [8] drew its inspiration from Locality Sensitive Hashing (LSH) [11] to

turn floating-point vectors into binary strings. This is done by thresholding the vectors after multiplication with an appropriate matrix. Similarity between descriptors is then measured by the Hamming distance between the corresponding binary strings. This is very fast because the Hamming distance can be computed very efficiently with a bitwise XOR operation followed by a bit count. The same algorithm was applied to the GIST descriptor to obtain a binary description of an entire image [12]. Another way to binarize the GIST descriptor is to use non-linear Neighborhood Component Analysis [12,13], which seems more powerful but probably slower at run-time.

While all three classes of shortening techniques provide satisfactory results, relying on them remains inefficient in the sense that first computing a long descriptor then shortening it involves a substantial amount of time-consuming computation. By contrast, the approach we advocate in this paper directly builds short descriptors by comparing the intensities of pairs of points without ever creating a long one. Such intensity comparisons were used in [9] for classification purposes and were shown to be very powerful in spite of their extreme simplicity. Nevertheless, the present approach is very different from [9] and [14] because it does *not* involve any form of online or offline training.

3 Method

Our approach is inspired by earlier work [9,15] that showed that image patches could be effectively classified on the basis of a relatively small number of pairwise intensity comparisons. The results of these tests were used to train either randomized classification trees [15] or a Naive Bayesian classifier [9] to recognize patches seen from different viewpoints. Here, we do away with both the classifier and the trees, and simply create a bit vector out of the test responses, which we compute after having smoothed the image patch.

More specifically, we define test τ on patch $\mathbf{p}$ of size $S \times S$ as

$$\tau(\mathbf{p}; \mathbf{x}, \mathbf{y}) := \begin{cases} 1 & \text{if } \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} , \tag{1}$$

where $\mathbf{p}(\mathbf{x})$ is the pixel intensity in a smoothed version of $\mathbf{p}$ at $\mathbf{x} = (u, v)^\top$. Choosing a set of n_d $(\mathbf{x}, \mathbf{y})$-location pairs uniquely defines a set of binary tests. We take our BRIEF descriptor to be the n_d-dimensional bitstring

$$f_{n_d}(\mathbf{p}) := \sum_{1 \le i \le n_d} 2^{i-1} \, \tau(\mathbf{p}; \mathbf{x}_i, \mathbf{y}_i) \ . \tag{2}$$

In this paper we consider $n_d = 128$, 256, and 512 and will show in the Results section that these yield good compromises between speed, storage efficiency, and recognition rate. In the remainder of the paper, we will refer to BRIEF descriptors as BRIEF-k, where $k = n_d/8$ represents the number of *bytes* required to store the descriptor.

When creating such descriptors, the only choices that have to be made are those of the kernels used to smooth the patches before intensity differencing and

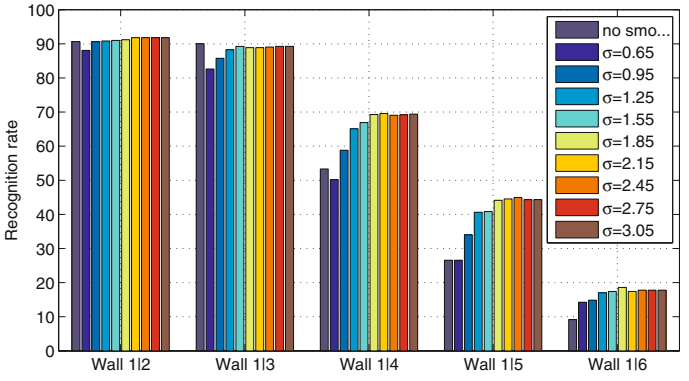


Fig. 1. Each group of 10 bars represents the recognition rates in one specific stereo pair for increasing levels of Gaussian smoothing. Especially for the hard-to-match pairs, which are those on the right side of the plot, smoothing is essential in slowing down the rate at which the recognition rate decreases.

the spatial arrangement of the $(\mathbf{x}, \mathbf{y})$-pairs. We discuss these in the remainder of this section.

To this end, we use the Wall dataset that we will describe in more detail in section 4. It contains five image pairs, with the first image being the same in all pairs and the second image shot from a monotonically growing baseline, which makes matching increasingly more difficult. To compare the pertinence of the various potential choices, we use as a quality measure the *recognition rate* in image pairs that will be precisely defined at the beginning of section 4. In short, for both images of a pair and for a given number of corresponding keypoints between them, it quantifies how often the correct match can be established using BRIEF for description and the Hamming distance as the metric for matching. This rate can be computed reliably because the scene is planar and the homography between images is known. It can therefore be used to check whether points truly correspond to each other or not.

3.1 Smoothing Kernels

By construction, the tests of Eq. 1 take only the information at single pixels into account and are therefore very noise-sensitive. By pre-smoothing the patch, this sensitivity can be reduced, thus increasing the stability and repeatability of the descriptors. It is for the same reason that images need to be smoothed before they can be meaningfully differentiated when looking for edges. This analogy applies because our intensity difference tests can be thought of as evaluating the sign of the derivatives within a patch.

Fig. 1 illustrates the effects of increasing amounts of Gaussian smoothing on the recognition rates for variances of Gaussian kernel ranging from 0 to 3. The more difficult the matching, the more important smoothing becomes to achieving good performance. Furthermore, the recognition rates remain relatively constant

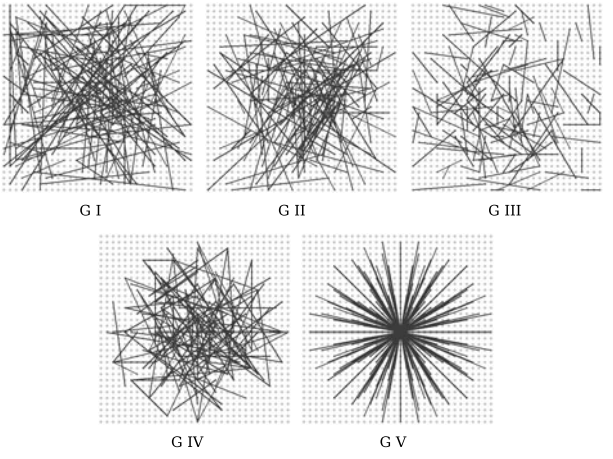


Fig. 2. Different approaches to choosing the test locations. All except the righmost one are selected by random sampling. Showing 128 tests in every image.

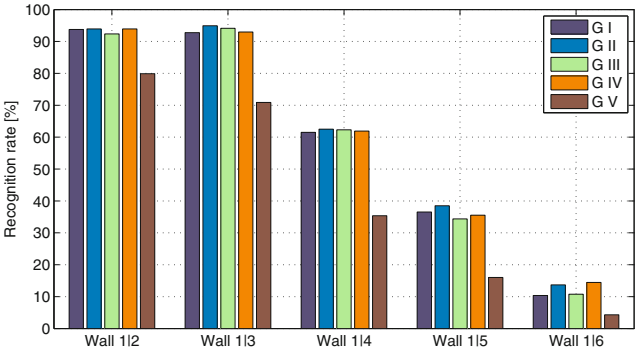


Fig. 3. Recognition rate for the five different test geometries introduced in section 3.2.

in the 1 to 3 range and, in practice, we use a value of 2. For the corresponding discrete kernel window we found a size of 9×9 pixels be necessary and sufficient.

3.2 Spatial Arrangement of the Binary Tests

Generating a length n_d bit vector leaves many options for selecting the n_d test locations $(\mathbf{x}_i, \mathbf{y}_i)$ of Eq. 1 in a patch of size $S \times S$. We experimented with the five sampling geometries depicted by Fig. 2. Assuming the origin of the patch coordinate system to be located at the patch center, they can be described as follows.

I) $(\mathbf{X}, \mathbf{Y}) \sim$ i.i.d. Uniform$(-\frac{S}{2}, \frac{S}{2})$: The $(\mathbf{x}_i, \mathbf{y}_i)$ locations are evenly distributed over the patch and tests can lie close to the patch border.

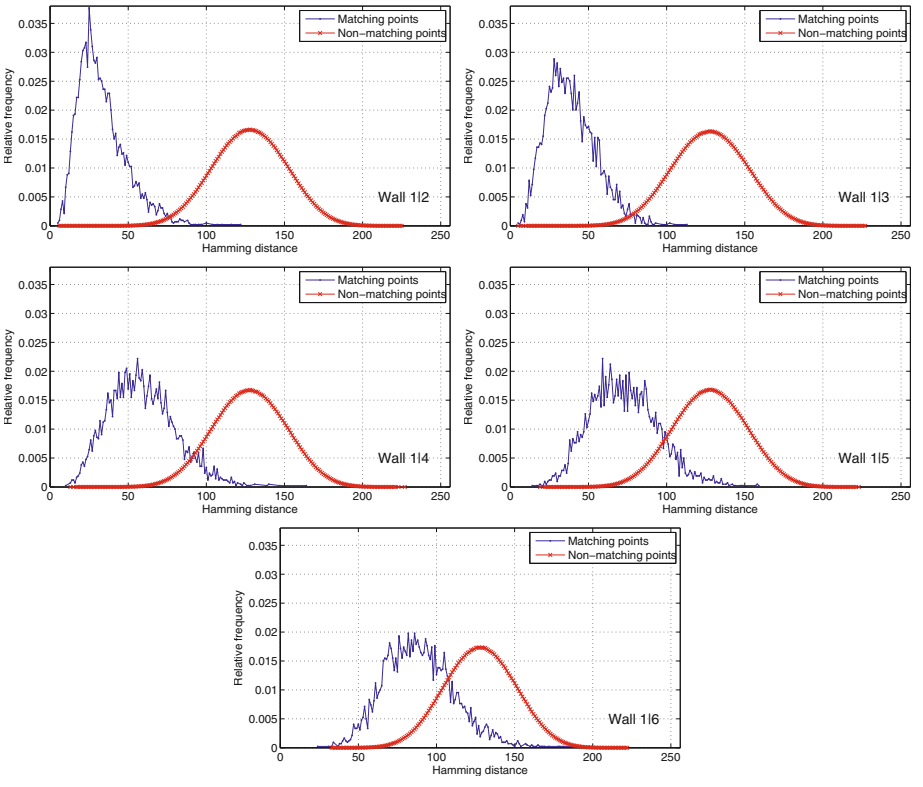


Fig. 4. Distributions of Hamming distances for matching pairs of points (thin blue lines) and for non-matching pairs (thick red lines) in each of the five image pairs of the Wall dataset. They are most separated for the first image pairs, whose baseline is smaller, ultimately resulting in higher recognition rates.

II) $(\mathbf{X}, \mathbf{Y}) \sim$ i.i.d. Gaussian$(0, \frac{1}{25}S^2)$: The tests are sampled from an isotropic Gaussian distribution. Experimentally we found $\frac{s}{2} = \frac{5}{2}\sigma \Leftrightarrow \sigma^2 = \frac{1}{25}S^2$ to give best results in terms of recognition rate.

III) $\mathbf{X} \sim$ i.i.d. Gaussian$(0, \frac{1}{25}S^2)$, $\mathbf{Y} \sim$ i.i.d. Gaussian$(\mathbf{x}_i, \frac{1}{100}S^2)$: The sampling involves two steps. The first location $\mathbf{x}_i$ is sampled from a Gaussian centered around the origin while the second location is sampled from another Gaussian centered on $\mathbf{x}_i$. This forces the tests to be more local. Test locations outside the patch are clamped to the edge of the patch. Again, experimentally we found $\frac{S}{4} = \frac{5}{2}\sigma \Leftrightarrow \sigma^2 = \frac{1}{100}S^2$ for the second Gaussian performing best.

IV) The $(\mathbf{x}_i, \mathbf{y}_i)$ are randomly sampled from discrete locations of a coarse polar grid introducing a spatial quantization.

V) $\forall i : \mathbf{x}_i = (0, 0)^\top$ and $\mathbf{y}_i$ is takes all possible values on a coarse polar grid containing n_d points.

For each of these test geometries we compute the recognition rate and show the result in Fig. 3. Clearly, the symmetrical and regular G V strategy loses out against all random designs G I to G IV, with G II enjoying a small advantage over the other three in most cases. For this reason, in all further experiments presented in this paper, it is the one we will use.

3.3 Distance Distributions

In this section, we take a closer look at the distribution of Hamming distances between our descriptors. To this end we extract about 4000 matching points from the five image pairs of the Wall sequence. For each image pair, Fig. 4 shows the normalized histograms, or distributions, of Hamming distances between corresponding points (in blue) and non-corresponding points (in red). The maximum possible Hamming distance being $32 \cdot 8 = 256$ bits, unsurprisingly, the distribution of distances for non-matching points is roughly Gaussian and centered around 128. As could also be expected, the blue curves are centered around a smaller value that increases with the baseline of the image pairs and, therefore, with the difficulty of the matching task.

Since establishing a match can be understood as classifying pairs of points as being a match or not, a classifier that relies on these Hamming distances will work best when their distributions are most separated. As we will see in section 4, this is of course what happens with recognition rates being higher in the first pairs of the Wall sequence than in the subsequent ones.

4 Results

In this section, we compare our method against several competing approaches. Chief among them is the latest OpenCV implementation of the SURF descriptor [4], which has become a *de facto* standard for fast-to-compute descriptors. We use the standard SURF64 version, which returns a 64-dimensional floating point vector and requires 256 bytes of storage. Because BRIEF, unlike SURF, does not correct for orientation, we also compare against U-SURF [4], where the U stands for upright and means that orientation also is ignored [4].

To this end, we use the six publicly available test image sequences depicted by Fig. 5. They are designed to test robustness to

- viewpoint changes (Wall[1], Graffiti[1], Fountain[2]),
- compression artifacts (Jpeg[1]),
- illumination changes (Light[1]),
- and image blur (Trees[1]).

For each one, we consider 5 image pairs by matching the first of 6 images to the other five they contain. Note that, the Wall and Graffiti scenes being planar,

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

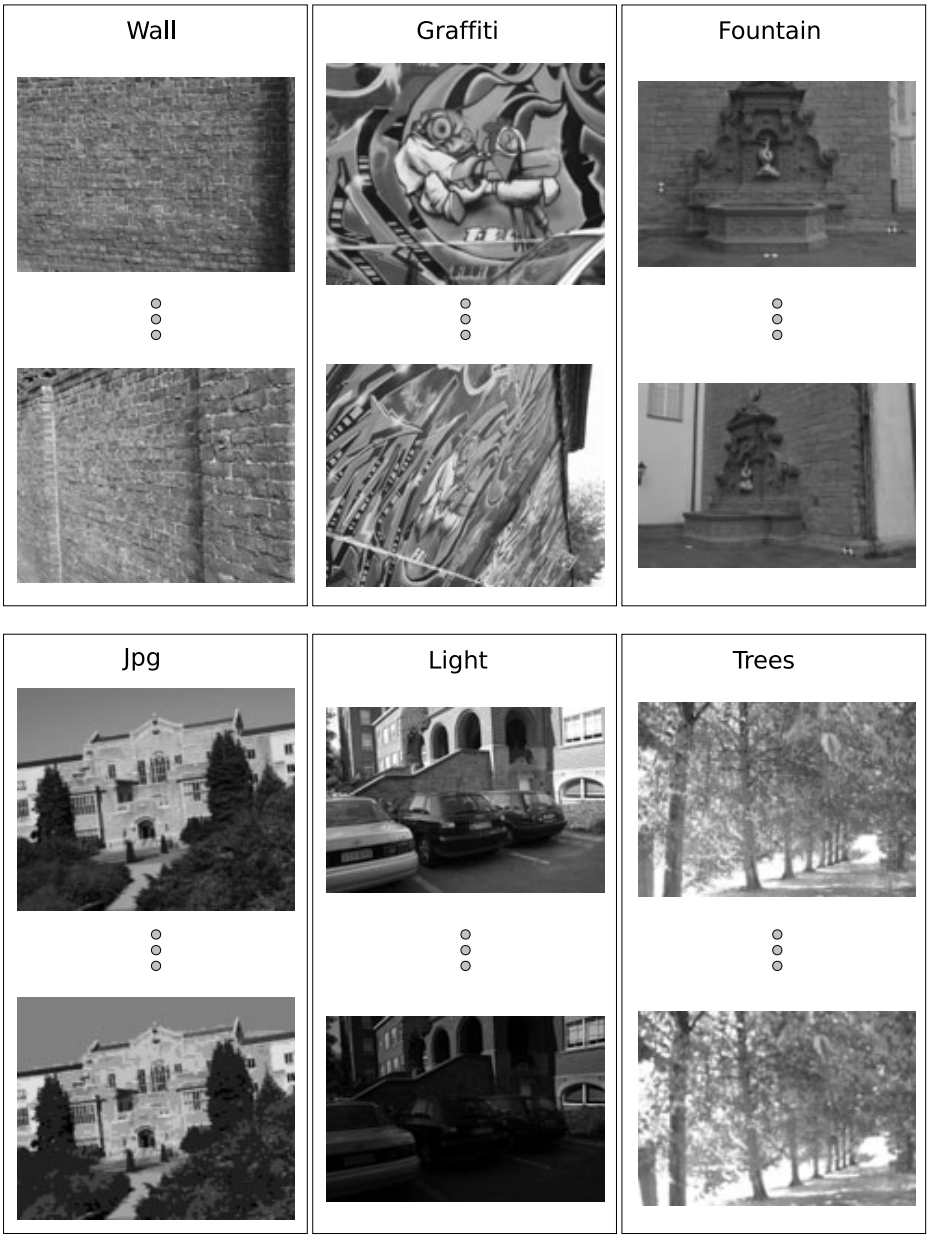


Fig. 5. Data sets used for comparison purposes. Each one contains 6 images and we consider 5 image pairs by matching the first one against all others.

the images are related by homographies that are used to compute the ground truth. The viewpoints for Jpeg, Light, and Trees being almost similar, the images are also taken to be related by a homography, which is very close to being

the identity. By contrast, the Fountain scene is fully three-dimensional and the ground truth is computed from laser scan data. Note also that the 5 pairs in Wall and Fountain are sorted in order of increasing baseline so that pair 1|6 is much harder to match than pair 1|2, which negatively affects the performance of *all* the descriptors considered here.

For evaluation purposes, we rely on two straightforward metrics, elapsed CPU time and recognition rate. The former simply is averaged measured wall clock time over many repeated runs. Given an image pair, the latter is computed as follows:

- Pick N interest points from the first image, infer the N corresponding points in the other from the ground truth data, and compute the $2N$ associated descriptors using the method under consideration.
- For each point in the first set, find the nearest neighbor in the second one and call it a match.
- Count the number of correct matches n_c and take the recognition rate to be $r = n_c/N$.

Although this may artificially increase the recognition rates, only the absolute recognition rate numbers are to be taken with caution. Since we apply the same procedure to all descriptors, and not only ours, the relative rankings we obtain are still valid and speak in BRIEF's favor. To confirm this, we detected SURF-points in both images of each test pair and computed their (SURF- or BRIEF-) descriptors, matched these descriptors to their nearest neighbor, and applied a standard left-right consistency check. Even though this setup now involves outliers, BRIEF continued to outperform SURF in the same proportion as before.

In the remainder of this section we will use these metrics to show that the computational requirements of BRIEF are much lower than those of all other methods while achieving better recognition rates than SURF on all sequences except Graffiti. This is explained both by the fact that this dataset requires strong rotation invariance, which BRIEF does not provide, and by the very specific nature of the Graffiti images. They contain large monochrome areas on which our intensity difference tests are often uninformative. In other words, this data set clearly favors descriptors based on gradient histograms, as has already been noted [7]. When comparing recognition rates against those of U-SURF, BRIEF still does better on Wall, Fountain, Trees, and similarly on Light and Jpg.

In other words, on data sets such as those that involve only modest amounts of in-plane rotation, there is a cost not only in terms of speed but also of recognition rate to achieving orientation invariance, as already pointed out in [4]. This explains in part why both BRIEF and U-SURF outperform SURF. Therefore, when not actually required, orientation correction should be avoided. This is an important observation because there are more and more cases, such as when using a mobile phone equipped with an orientation sensor, when orientation invariance stops being a requirement. This is also true in many urban contexts where photos tend to be taken at more or less canonical orientations and for mobile robotics where the robot's attitude is known.

Recognition Rate as a Function of Descriptor Size. Since many practical problems involve matching a few hundred feature points, we first use $N = 512$ to compare the recognition rate of BRIEF using either 128, 256, or 512 tests, which we denote as BRIEF-16, BRIEF-32, and BRIEF-64. The trailing number stands for the number of bytes required to store the descriptor. Recall that since both SURF and U-SURF return 64 floating point numbers, they require 256 bytes of storage and are therefore at least four times bigger.

As shown in Fig. 6, BRIEF-64 outperforms SURF and U-SURF in all sequences except Graffiti while using a descriptor that is four times smaller. Unsurprisingly, BRIEF-32 does not do quite as well but still compares well against SURF and U-SURF. BRIEF-16 is too short and shows the limits of the approach.

To show that this behavior is *not* an artifact for the number N of feature points used for testing purposes, we present similar recognition rates for values of N ranging from 512 to 4096 in Fig. 7. As could be expected, the recognition rates drop as N increases for *all* the descriptors but the rankings remain unchanged.

In Fig. 8, we use the wall data set to plot recognition rates as a function of the number of tests. We clearly see a saturation effect beyond 200 tests for the easy cases and an improvement up to 512 for the others. This tallies with the results of Fig. 6 showing that BRIEF-32 (256 tests) yields near optimal results for the short baseline pairs and that BRIEF-64 (512 tests) is more appropriate for the others.

Influence of Feature Detector. To perform the experiments described above, we used SURF keypoints so that we could run both SURF, U-SURF, and BRIEF on the same points. This choice was motivated by the fact that SURF requires an orientation and a scale and U-SURF a scale, which the SURF detector provides.

However, in practice, using the SURF detector in conjunction with BRIEF would negate part of the considerable speed advantage that BRIEF enjoys over SURF. It would make much more sense to use a fast detector such as [16]. To test the validity of this approach, we therefore recomputed our recognition rates on the Wall sequence using CenSurE keypoints[3] instead of SURF keypoints. As can be seen in Fig. 9-left, BRIEF works even slightly better for CenSurE points than for SURF points.

Orientation Sensitivity. BRIEF is not designed to be rotationally invariant. Nevertheless, as shown by our results on the 5 test data sets, it tolerates small amounts of rotation. To quantify this tolerance, we take the first image of the Wall sequence with $N = 512$ points and match these against points in a rotated version of itself, where the rotation angle ranges from 0 to 180 degrees.

Fig. 9-right depicts the recognition rate of BRIEF-32, SURF, and U-SURF. Since the latter does not correct for orientation either, its behavior is very similar or even a bit worse than that of BRIEF: Up to 10 to 15 degrees, there is little degradation followed by a precipitous drop. SURF, which attempts to

[3] Center Surrounded Extrema, or CenSurE for short, is implemented in OpenCV 2.0 under the alias name *Star detector*, following the shape of the CenSurE detector.

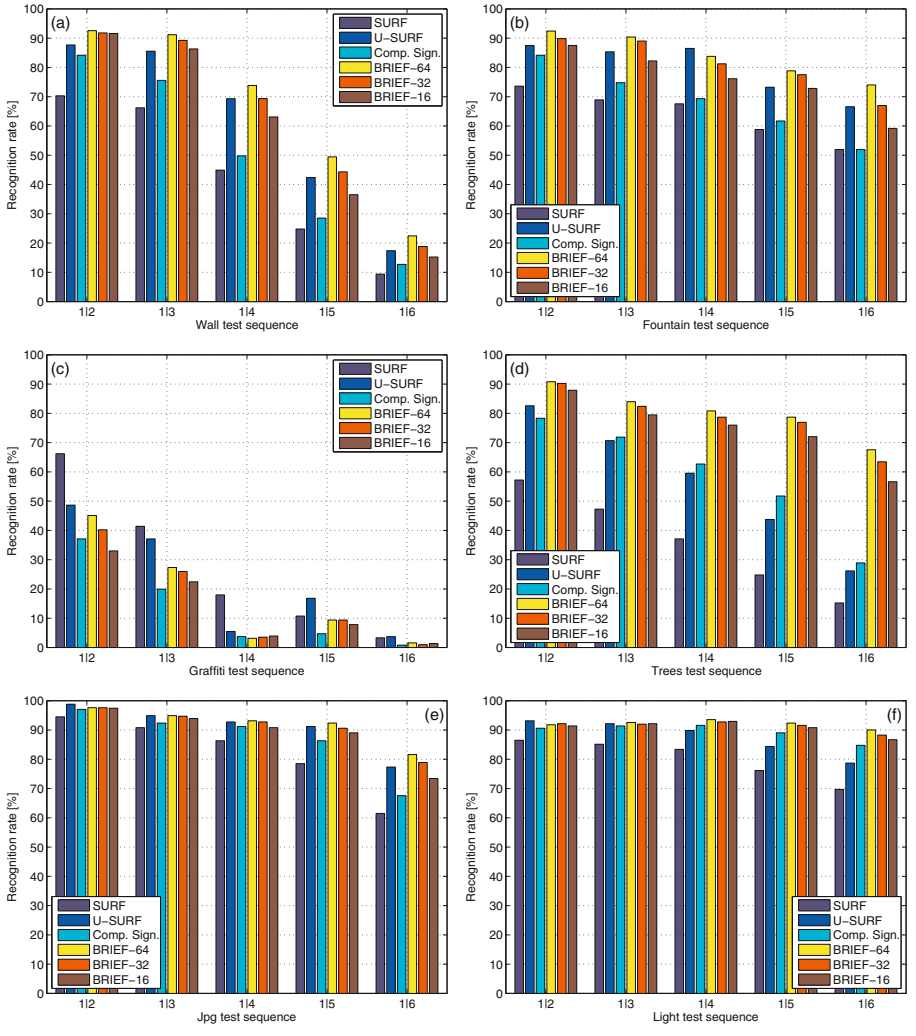


Fig. 6. Recognition rates on (a) Wall (b) Fountain. (c) Graffiti (d) Trees (e) Jpg (f) Light. The trailing 16, 32, or 64 in the descriptor's name is its length in bytes. It is much shorter than those of SURF and U-SURF, which both are 256. For completeness, we also compare to a recent approach called Compact Signatures [7] which has been shown to be very efficient. We obtained the code from OpenCV's SVN repository.

compensate for orientation changes, does better for large rotations but worse for small ones, highlighting once again that orientation-invariance comes at a cost.

To complete the experiment, we plot a fourth curve labeled as O-BRIEF-32, where the "O" stands for orientation correction. In other words, we run BRIEF-32 on an image rotated using the orientation estimated by SURF. O-BRIEF-32 is not meant to represent a practical approach but to demonstrate

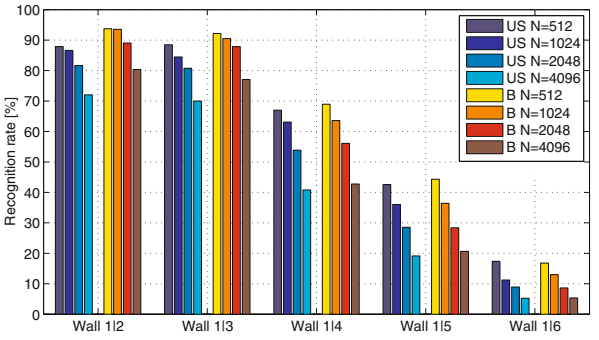


Fig. 7. Influence of the number N of keypoints being considered on the recognition rates. For each of the five image pairs of Wall, we plot on the left side four sets of rates for N being 512, 1024, 2048, and 4096 when using U-SURF, and four equivalent sets when using BRIEF-32. The first are denoted as US and the second as B in the color chart. Note that the recognition rates consistently decrease when N increases but that, for the same N, BRIEF-32 outperforms U-SURF, except in the last image pair where the recognition rate is equally low for both.

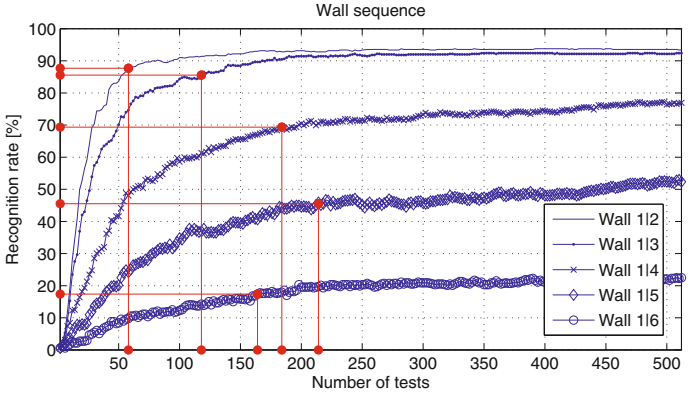


Fig. 8. Recognition rate as a function of the number of tests on Wall. The vertical and horizontal lines denote the number of tests required to achieve the same recognition rate as U-SURF on respective image pairs. In other words, BRIEF requires only 58, 118, 184, 214, and 164 bits for Wall 1|2, ..., 1|6, respectively, which compares favorably to U-SURF's $64 \cdot 4 \cdot 8 = 2048$ bits (assuming 4 bytes/float).

that the response to in-plane rotations is more a function of the quality of the orientation estimator rather than of the descriptor itself, as evidenced by the fact that O-BRIEF-32 and SURF are almost perfectly superposed.

Estimating Speed. In a practical setting where either speed matters or computational resources are limited, not only should a descriptor exhibit the highest possible recognition rates but also be computationally as cheap as

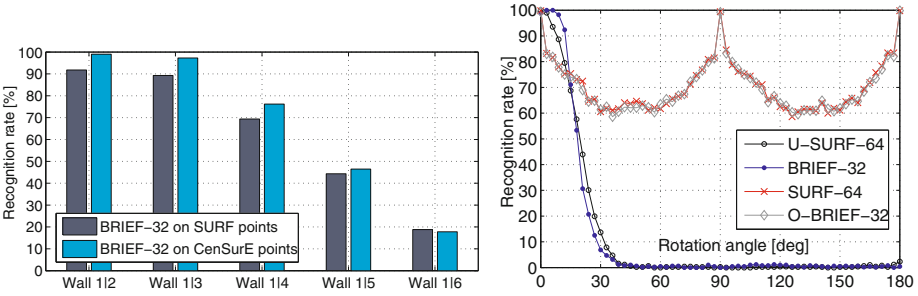


Fig. 9. Left: Using CenSurE keypoints instead of SURF keypoints. BRIEF works slightly better with CenSurE than with SURF keypoints. Right: Recognition rate when matching the first image of the Wall dataset against a rotated version of itself, as a function of the rotation angle.

possible. Matching a number of points between two images typically involves three steps:

1) Detecting the feature points.
2) Computing the description vectors.
3) Matching, which means finding the nearest neighbor in descriptor space.

For affine-invariant methods such as SURF, the first step can involve a costly scale-space search for local maxima. In the case of BRIEF, any fast detector such as CenSurE [16] or FAST [17] can be used. BRIEF is therefore at an advantage there.

The following table gives timing results for the second and third steps for 512 keypoints, measured on a 2.66 GHz/Linux x86-64 machine, in milliseconds:

	BRIEF-16	BRIEF-32	BRIEF-64	SURF-64
Descriptor computation	8.18	8.87	9.57	335
Matching (exact NN)	2.19	4.35	8.16	28.3

As far as building the descriptors is concerned, we observe a 35- to 41-fold speed-up over SURF where the time for performing and storing the tests remains virtually constant. U-SURF being about 1/3 faster than SURF [4], the equivalent number should be an 23- to 27-fold speed increase. Because BRIEF spends by far the most CPU time with smoothing, approximate smoothing techniques based on integral images may yield extra speed. For matching, we observe a 4- to 13-fold speed-up over SURF. The matching time scales quadratically with the number of bits used in BRIEF but the absolute values remain extremely low within the useful range. Furthermore, in theory at least, these computation times could be driven almost to zero using the POPCNT instruction from SSE4.2 [10]. Because only the latest Intel Core i7 CPUs support this instruction, we were unable to exploit it and used a straight-forward SSE2/SSE4.1 implementation instead.

5 Conclusion

We have introduced the BRIEF descriptor that relies on a relatively small number of intensity difference tests to represent an image patch as a binary string.[4] Not only is construction and matching for this descriptor much faster than for other state-of-the-art ones, it also tends to yield higher recognition rates, as long as invariance to large in-plane rotations is not a requirement.

It is an important result from a practical point of view because it means that real-time matching performance can be achieved even on devices with very limited computational power. It is also important from a more theoretical viewpoint because it confirms the validity of the recent trend [18,12] that involves moving from the Euclidean to the Hamming distance for matching purposes.

In future work, we will incorporate orientation and scale invariance into BRIEF so that it can compete with SURF and SIFT in a wider set of situations. Using fast orientation estimators, there is no theoretical reason why this could not be done without any significant speed penalty.

References

1. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1615–1630 (2004)
2. Hua, G., Brown, M., Winder, S.: Discriminant Embedding for Local Image Descriptors. In: International Conference on Computer Vision (2007)
3. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. Computer Vision and Image Understanding 20, 91–110 (2004)
4. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: Speeded Up Robust Features. Computer Vision and Image Understanding 10, 346–359 (2008)
5. Tuytelaars, T., Schmid, C.: Vector Quantizing Feature Space With a Regular Lattice. In: International Conference on Computer Vision (2007)
6. Winder, S., Hua, G., Brown, M.: Picking the Best Daisy. In: Conference on Computer Vision and Pattern Recognition (2009)
7. Calonder, M., Lepetit, V., Konolige, K., Bowman, J., Mihelich, P., Fua, P.: Compact Signatures for High-Speed Interest Point Description and Matching. In: International Conference on Computer Vision (2009)
8. Shakhnarovich, G.: Learning Task-Specific Similarity. PhD thesis, Massachusetts Institute of Technology (2005)
9. Ozuysal, M., Calonder, M., Lepetit, V., Fua, P.: Fast Keypoint Recognition Using Random Ferns. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 448–461 (2010)
10. Intel: SSE4 Programming Reference: software.intel.com/file/18187 (2007)
11. Gionis, A., Indik, P., Motwani, R.: Similarity Search in High Dimensions Via Hashing. In: International Conference on Very Large Databases (2004)
12. Torralba, A., Fergus, R., Weiss, Y.: Small Codes and Large Databases for Recognition. In: Conference on Computer Vision and Pattern Recognition (2008)
13. Salakhutdinov, R., Hinton, G.: Learning a Nonlinear Embedding by Preserving Class Neighbourhood Structure. In: International Conference on Artificial Intelligence and Statistics (2007)

[4] The BRIEF code being very simple, we will be happy to make it publicly available.

14. Taylor, S., Rosten, E., Drummond, T.: Robust feature matching in 2.3s. In: IEEE CVPR Workshop on Feature Detectors and Descriptors: The State of the Art and Beyond (2009)
15. Lepetit, V., Fua, P.: Keypoint Recognition Using Randomized Trees. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1465–1479 (2006)
16. Agrawal, M., Konolige, K., Blas, M.: Censure: Center Surround Extremas for Realtime Feature Detection and Matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 102–115. Springer, Heidelberg (2008)
17. 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)
18. Shakhnarovich, G., Viola, P., Darrell, T.: Fast Pose Estimation With Parameter-Sensitive Hashing. In: International Conference on Computer Vision (2003)

Multi-label Feature Transform for Image Classifications

Hua Wang, Heng Huang, and Chris Ding

Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019, USA
huawang2007@mavs.uta.edu, heng@uta.edu, chqding@uta.edu

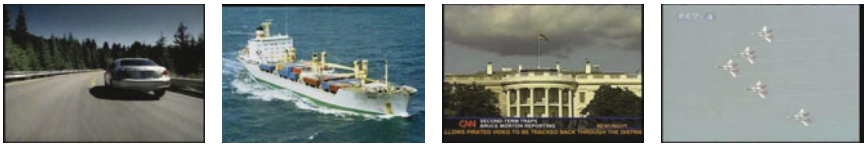
Abstract. Image and video annotations are challenging but important tasks to understand digital multimedia contents in computer vision, which by nature is a *multi-label multi-class* classification problem because every image is usually associated with more than one semantic keyword. As a result, label assignments are no longer confined to class membership indications as in traditional single-label multi-class classification, which also convey important characteristic information to assess object similarity from knowledge perspective. Therefore, besides implicitly making use of label assignments to formulate label correlations as in many existing multi-label classification algorithms, we propose a novel Multi-Label Feature Transform (MLFT) approach to also explicitly use them as part of data features. Through two transformations on attributes and label assignments respectively, MLFT approach uses kernel to implicitly construct a *label-augmented feature vector* to integrate attributes and labels of a data set in a balanced manner, such that the data discriminability is enhanced because of taking advantage of the information from both data and label perspectives. Promising experimental results on four standard multi-label data sets from image annotation and other applications demonstrate the effectiveness of our approach.

Keywords: Multi-label classification, Feature transformation, Image annotation.

1 Introduction

Automatically annotating image and video is a key task to understand digital multimedia contents for browsing, searching, and navigation. In the real world, an image or a video clip is usually attached with several different semantic keywords, *e.g.*, all the images in Fig. 1 are annotated with more than one keyword. This poses so-called *multi-label multi-class* classification problems, which refer to problems where each object can be assigned to multiple classes. Multi-label problems are more general than traditional *single-label* (multi-class) problems, in which each object is assigned to exactly one class. Driven by its broad applications in diverse domains, such as image/video annotation, gene function annotation, and text categorization, *etc.*, multi-label classification is receiving increasing attentions in recent years.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 793–806, 2010.



(a) car, road. (b) sea, ship. (c) building, sky. (d) sky, plane.

Fig. 1. Sample images from TRECVID 2005 dataset. Each image is annotated with multiple semantic keywords.

An important difference between single-label classification and multi-label classification is that the classes in single-label classification are assumed to be mutually exclusive while those in multi-label classification are normally interdependent from one another. For example, "sea" and "ship" tend to appear in a same image, while "fire" typically does not appear together with "ice". Thus, many multi-label classification algorithms have been developed to make use of label correlations to improve the overall multi-label classification performance. Two ways are popularly used to employ label correlations: incorporating label correlations into the existing label propagation learning algorithms, as either part of graph weight [10,2] or an additional constraint [17,15]; or utilizing label correlations to seek a more discriminative subspace [16,18,8]. Besides, there also exist many other methods based on different mechanisms, such as matrix factorization[12], maximizing label entropy [19], Bayesian model [6], and so on.

Our perspectives and motivations. One common aspect of many existing multi-label classification algorithms is that they all attempt to leverage correlations among classes, which are typically computed from label assignments on training data. Although this paradigm of implicitly using label assignments has shown their strength, it would be also favorable to explicitly use them as part of data attributes for classification [5]. In multi-label classification, multiple labels may be associated with a single object, hence the number of common labels shared by two different objects is no longer restricted to be one. Label assignments thus turn out a similarity measurement among data objects.

Therefore, in this work, we propose a novel Multi-Label Feature Transform (MLFT) approach to use label assignments as both part of data attributes (explicit usage) and label correlations (implicit usage) for enhanced multi-label classification performance. We first transform the original data attributes (via proposed Multi-label Kernel Laplacian Embedding (MLKLE) method) and corresponding label assignments (via proposed Correlative Kernel Transform (CKT) method) from their native spaces to two new (sub)spaces with similar dimensionality. Then the transformed feature vectors from these two types of data are integrated together to form a new *label augmented feature vector* in the spanned hyperspace. Because the dimensionalities of the two transformed feature vectors are close, the vector concatenation thereby space spanning is balanced. Most importantly, the label-augmented feature vector not only preserves original attributes information, but also captures label correlations through label

assignments implicitly and explicitly. As a result, data points in the spanned feature space are more discriminable, by which succeeding classification can be conducted more effectively. As an additional advantage, MLFT averts possible difficulties to directly employ state-of-the-art single-label classification methods to solve multi-label problems, such that their powerful classification capabilities can be utilized on multi-label data.

Although originated from simple vector concatenation, the proposed MLFT does not need to explicitly construct label-augmented feature vector due to introducing kernel. Moreover, using a more discriminative kernel, label-augmented feature vectors are mapped to a more linearly separable high-dimensional space such that classifications can be carried out much easier.

2 Multi-label Feature Transform

For a classification task with n data points and K classes, each data point $\mathbf{x}_i \in \mathbb{R}^p$ is associated with a subset of class labels represented by a binary vector $\mathbf{y}_i \in \{0,1\}^K$ such that $\mathbf{y}_i(k) = 1$ if $\mathbf{x}_i$ belongs to the kth class, and 0 otherwise. Meanwhile, we also have the pairwise similarities $W \in \mathbb{R}^{n \times n}$ among the n data points with W_{ij} indicating how closely $\mathbf{x}_i$ and $\mathbf{x}_j$ are related. W may be computed from attributes data or directly obtained from experimental observations. Suppose the number of labeled data points is $l(< n)$, our goal is to predict labels $\{\mathbf{y}_i\}_{i=l+1}^n$ for the unlabeled data points $\{\mathbf{x}_i\}_{i=l+1}^n$. We write $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$.

2.1 Outlines of MLFT Approach

In multi-label classification, each data point may be assigned to multiple classes at the same time, *i.e.*, $\sum_k \mathbf{y}_i(k) \geq 1$. Thus, the overlap of labels assigned to two data points computed by $\mathbf{y}_i^T \mathbf{y}_j$ is an integer ranged from 0 to K, which induces an affinity relationship between $\mathbf{x}_i$ and $\mathbf{x}_j$. This is different from the single-label case, where the number of overlapped labels of two data points is restricted to be either 0 or 1. Namely, $\mathbf{y}_i$ not only indicates the class membership for a data point, but also contains important attribute information to assess the similarity between data points. Therefore, we propose MLFT approach to construct a new label-augmented feature vector $\mathbf{z}_i \in \mathbb{R}^r$ to integrate the characteristic information conveyed by both $\mathbf{x}_i$ and $\mathbf{y}_i$, such that $\mathbf{z}_i$ are more separable to achieve enhanced classification performance.

A naive construction of $\mathbf{z}_i$ can be the simple concatenation of $\mathbf{x}_i$ and $\mathbf{y}_i$, which, however, suffers from two critical problems. Firstly, the original features $\mathbf{x}_i$ are often compromised by noise during the generation process, and thereby might not be very discriminable. Secondly, and more importantly, in many data sets from real computer vision applications, such as image annotation, the number of features of a data set is usually much greater than that of classes, *i.e.*, $p \gg K$. That is, directly concatenating $\mathbf{x}_i$ and $\mathbf{y}_i$ causes unbalance problem. Therefore, we need to transform $\mathbf{x}_i$ and $\mathbf{y}_i$ into two (sub)spaces with close dimensionalities, and also eliminate the irrelevant features.

Let $U \in \mathbb{R}^{p\times(K-1)}$ be a linear transformation for $\mathbf{x}_i$ (*i.e.*, the dimensionality of feature space is reduced to the number of class minus 1, $K-1 \ll p$), the input feature vector $\mathbf{q}_i^x \in \mathbb{R}^{K-1}$ is computed by $\mathbf{q}_i^x = U^T\mathbf{x}_i$. Similarly, we denote $\mathbf{p}_i^y \in \mathbb{R}^K$ as the input label vector, which is transformed from $\mathbf{y}_i$ and computed by a kernel function $\mathbf{p}_i^y = \phi(\mathbf{y}_i)$. Thus, the dimensionality of $\mathbf{z}_i$ is $r = 2K-1$. U is computed by Multi-label Kernel Laplacian Embedding (MLKLE) as detailed in Section 3, and $\phi(\cdot)$ is obtained by Correlative Kernel Transform (CKT) as introduced Section 4. The label-augmented feature vector $\mathbf{z}_i$ is thus constructed as follows:

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{q}_i^x \\ \mathbf{p}_i^y \end{bmatrix} . \tag{1}$$

2.2 Implicit Construction of Label-Augmented Feature Vector

Once the label-augmented feature vector $\mathbf{z}_i$ is computed, any traditional single-label classification method can be used to carry out classification. In this work, we use support vector machine (SVM) because of its elegant theoretical foundation and powerful classification capability. A special benefit of using SVM with kernel is that the construction of the label-augmented feature vector $\mathbf{z}_i$ can be conveniently interpreted from kernel perspective. To be more specific, let $\mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j)$ be a radial basis function (RBF) kernel on $\mathbf{z}_i$:

$$\begin{aligned} \mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j) &= \exp\left(-\gamma\|\mathbf{z}_i - \mathbf{z}_j\|^2\right) = \exp\left(-\gamma\|\mathbf{q}_i^x - \mathbf{q}_j^x\|^2\right)\exp\left(-\gamma\|\mathbf{p}_i^y - \mathbf{p}_j^y\|^2\right) \\ &= \mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x)\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y), \end{aligned} \tag{2}$$

where $\mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x)$ and $\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y)$ are the kernels with respect to input feature vectors and input label vectors respectively. Therefore, the construction of $\mathbf{z}_i$ in Eq. (1) indeed can be seen as a multiplicative kernel, which comprises information from both data attributes and label assignments. Similarly, $\mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j)$ can be seen as an additive kernel when linear kernel is used, and the same for other popular kernels used in SVM.

Therefore, instead of explicitly transforming $\mathbf{y}_i$, we focus on devising a discriminative kernel as described in Section 4. By introducing kernel, explicit construction of label-augmented feature vector $\mathbf{z}_i$ in Eq. (1) is no longer needed. Instead, we may use a more delicate kernel to incorporate additional useful information for better classification.

We outline the classification procedures by the proposed MLFT approach in Table 1 and will describe the details of each step in the rest of this paper. As can be seen, we concentrate on the data preparation phase, because it is the most essential part to boost classification performance.

3 Multi-label Kernel Laplacian Embedding

As analyzed in Section 2.1, two problems, indiscriminability of $\mathbf{x}_i$ and unbalanced cardinalities of attributes space and label space, prevent us from using direct concatenation of $\mathbf{x}_i$ and $\mathbf{y}_i$. Therefore, we first solve these two difficulties for $\mathbf{x}_i$

Table 1. Classification using the proposed MLFT approach

Data preparation: (a) Initialize unlabeled data points $\{\mathbf{x}_i\}_{i=l+1}^n$ to get the initial labels $\{\hat{\mathbf{y}}_i\}_{i=l+1}^n$. (Section 5) (b) Compute the linear transformation U using proposed MLKLE method from $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^l$ and $\{(\mathbf{x}_i, \hat{\mathbf{y}}_i)\}_{i=l+1}^n$. Project all the data points including those unlabeled into the embedding subspace to obtain $\{\mathbf{q}_i^x\}_{i=1}^n$ as in Eq. (5). (Section 3) (c) Compute kernel matrix $\mathcal{K}_z$ on label-augmented feature vectors $\{\mathbf{z}_i\}_{i=1}^n$ as in Eq. (15) by $\{\mathbf{q}_i^x\}_{i=1}^n$ and implicit transform of $\{\mathbf{y}_i\}_{i=1}^l$ using proposed CKT method. (Section 4)
Training: Training K SVM classifiers, one for each class.
Testing: Classify $\{\mathbf{z}_i\}_{i=l+1}^n$ using the trained classifiers to obtain the predicted labels $\{\mathbf{y}_i\}_{i=l+1}^n$, one class at a time.

and propose a Multi-label Kernel Laplacian Embedding (MLKLE) method to reduce the dimensionality of $\mathbf{x}_i$ and seek its intrinsic structure with irrelevant patterns pruned. This produces a transformation $U \in \mathbb{R}^{p\times(K-1)}$ by maximizing the following optimization objective:

$$J = \mathbf{tr}\left(\frac{U^T X \mathcal{K} X^T U}{U^T X (D-W) X^T U}\right), \tag{3}$$

where $\mathcal{K}$ is a $n \times n$ kernel matrix, and $D = \mathrm{diag}\,(d_1, \ldots, d_n)$, $d_i = \sum_j W_{ij}$. Thus, $L = D - W$ is the graph Laplacian [3]. $\mathcal{K}$ and W are defined later in Eq. (11) and Eq. (13) respectively. The solution to this problem is well established in mathematics by resolving the following generalized eigenvalue problem:

$$X \mathcal{K} X^T \mathbf{u}_k = \lambda_k X (D-W) X^T \mathbf{u}_k, \tag{4}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$ are the result eigenvalues and $\mathbf{u}_k$ are the corresponding eigenvectors. Hence, $U = [\mathbf{u}_1, \ldots, \mathbf{u}_{K-1}]$.

Using U, $\mathbf{x}_i$ can be projected into the embedding space by

$$\mathbf{q}_i^x = U^T \mathbf{x}_i \ . \tag{5}$$

Because many modern data sets in real life come in from multiple data sources, we often have both *attributes* information about data objects (vector data X) and various *pairwise similarities* between data objects (graph data W) for a same data set at the same time. Since these two types of data both convey valuable information for class membership inference, many existing embedding algorithms designed for only one type of input data are usually insufficient, especially the correlations between these two types of data are not exploited. In contrast, the true power of the proposed MLKLE in Eq. (3) lies in that it integrates both *attributes* and *pairwise relationships* in an integral optimization objective, such

that the discriminability of projected data in the shared low-dimensional subspace is doubly reinforced due to taking advantage of the information from the both data sources. In the rest of this section, we derive the optimization objective in Eq. (3).

3.1 PCA Laplacian Embedding

For attribute data X, Principle Component Analysis (PCA) [9] maximizes the data variance in the embedding space to retain the most information. Let $Q^T = [\mathbf{q}_1^x, \ldots, \mathbf{q}_n^x] \in \mathbb{R}^{(K-1)\times n}$ be the projected feature matrix, PCA aims to find the projection by maximizing

$$J_{\text{PCA}} = \mathbf{tr}\left(Q^T X^T X Q\right) \quad . \tag{6}$$

For pairwise relationships W, Laplacian embedding preserves the same relationships and maximizes the smoothness with respect to the intrinsic manifold of the data set by minimizing [7]

$$J_{\text{Lap}} = \mathbf{tr}\left(Q^T (D - W) Q\right) \quad . \tag{7}$$

Combining Eq. (6) and Eq. (7), we can construct an additive objective as:

$$J = \alpha J_{\text{Lap}} - (1 - \alpha) J_{\text{PCA}}, \tag{8}$$

where $0 < \alpha < 1$ is a tradeoff parameter. In practice, however, optimal α is hard to choose. Thus, instead of using a trace *difference* as in Eq. (8), we formulate the objective as a trace *quotient* so that α is removed:

$$J_{\text{PCA-Lap}} = \mathbf{tr}\left(\frac{Q^T X^T X Q}{Q^T (D - W) Q}\right) \quad . \tag{9}$$

We call Eq. (9) as PCA Laplacian embedding, which integrates both attributes and pairwise relationship in a same embedding space.

3.2 Kernel Laplacian Linear Embedding (KLE)

PCA Laplacian embedding in Eq. (9) is a purely unsupervised embedding, while label information, though useful, is not leveraged. We notice that $X^T X$ in Eq. (9) is a linear kernel, which could be replaced by a more discriminative kernel $\mathcal{K}$. Besides the supervision information from training data, label correlations can also be incorporated via $\mathcal{K}$, therefore we defer the definition of $\mathcal{K}$ now and will give its detailed implementation later by Eq. (11) in Section 3.3. Replacing $X^T X$ by $\mathcal{K}$, we have a new objective as:

$$\max_{Q} \mathbf{tr}\left(\frac{Q^T \mathcal{K} Q}{Q^T (D - W) Q}\right) \quad . \tag{10}$$

Using linear embedding $Q^T = U^T X$, the optimization objective in Eq. (3) is derived. We call it as Kernel Laplacian Embedding (KLE).

3.3 Label Correlation Enhanced Kernel Laplacian Embedding

Because label correlations are important information contained exclusively in multi-label data and useful for multi-label classification, it would be beneficial to take advantage them in KLE. Equipped with the kernel matrix $\mathcal{K}$ and graph Laplacian $L = D - W$ in Eq. (3), we can easily incorporate label correlations into the optimization objective. We first denote $C \in \mathbb{R}^{K \times K}$ as the label correlation matrix, which will be defined later by Eq. (17) in Section 6. C is a symmetric matrix and its entry C_{kl} captures the correlation between the kth and lth classes.

Correlation Enhanced Kernel. Instead of using the simplest linear kernel XX^T in Eq. (9), we may use a multiplicative kernel to carry more information:

$$\mathcal{K}_{ij} = \mathcal{K}_x(\mathbf{x}_i, \mathbf{x}_j)\mathcal{K}_y(\mathbf{y}_i, \mathbf{y}_j), \tag{11}$$

where $\mathcal{K}_x$ is a kernel with respect to data attributes X, and $\mathcal{K}_y$ is a kernel with respect to label assignments Y.

Same as most existing related works, $\mathcal{K}_x$ is constructed using the Gaussian kernel $\mathcal{K}_x(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma\right)$, where σ is fine tuned by 5-fold cross validation in our evaluations.

$\mathcal{K}_y$ encodes label correlations via assessing the label similarities between data points. The simplest way to measure the label overlap of two data points computes $\mathbf{y}_i^T \mathbf{y}_j$. The bigger the overlap is, the more similar the two data points are. The problem of this straightforward similarity is that it treats all the classes independent and can not exploit the correlations among them. In particular, it will give zero similarity whenever two data points do not share any labels. However, data points with no common label can still be strongly related if their attached labels are highly correlated. Therefore, instead of computing the label similarity by the dot product, we compute it by $\mathbf{y}_i^T\, C\, \mathbf{y}_j$ with normalization:

$$\mathcal{K}_y(\mathbf{y}_i, \mathbf{y}_j) = \frac{\mathbf{y}_i^T\, C\, \mathbf{y}_j}{\|\mathbf{y}_i\|\, \|\mathbf{y}_j\|} \quad . \tag{12}$$

We call $\mathcal{K}$ as the *correlation enhanced kernel*. In order to compute $\mathcal{K}_y$, we need to initialize the unlabeled data point first, which will be detailed later in Section 5.

Correlation Enhanced Pairwise Similarities. Many existing embedding algorithms construct the pairwise similarity matrix (W) only from data attributes (X), while label correlations are overlooked, which, however, is useful in multi-label classification. Therefore, we propose a *Correlation Enhanced Pairwise Similarity* scheme to make use of them as follows:

$$W = W_{\mathrm{X}} + \beta W_{\mathrm{L}}, \tag{13}$$

where W_{X} can be constructed from attribute data or obtained directly from experimental observations, and W_{L} is the label similarity matrix. β is a parameter determining how much the pairwise relationship should be biased by label similarities, and empirically selected as $\beta = \sum_{i,j,i\neq j} W_{\mathrm{X}}(i,j) / \sum_{i,j,i\neq j} W_{\mathrm{L}}(i,j)$.

As $\mathcal{K}_x$ and $\mathcal{K}_y$ defined above readily assess the similarities between data points from data and knowledge perspectives respectively, we define:

$$W_{\mathrm{X}}(i,j) = \begin{cases} \mathcal{K}_x(i,j) & i \neq j, \\ 0 & i = j; \end{cases} \qquad W_{\mathrm{L}}(i,j) = \begin{cases} \mathcal{K}_y(i,j) & i \neq j, \\ 0 & i = j \ . \end{cases} \tag{14}$$

Finally, when $\mathcal{K}$ and W are defined as in Eq. (11) and Eq. (13), we call Eq. (3) as the proposed Multi-Label Kernel Laplacian Embedding (MLKLE) method, which is summarized in Table 2.

Table 2. Algorithm of MLKLE

Input:
$X \in \mathbb{R}^{p \times n}$: centralized feature matrix
$Y \in \mathbb{R}^{K \times n}$: label matrix

Steps:
1 Construct $\mathcal{K}$ as in Eq. (11) and W as in Eq. (13).
2 Compute $X\mathcal{K}X^T$ and $X(D-W)X^T$.
3 Resolve the generalized eigenvalue problem as in Eq. (4). Construct the projection matrix U by the eigenvectors corresponding to the $(K-1)$ leading eigenvalues.

Output:
$U \in \mathbb{R}^{p \times (K-1)}$: the projection matrix to project data from original feature space $\mathbb{R}^p$ to the embedding space $\mathbb{R}^{(K-1)}$.

4 Correlative Kernel Transformation

Because the construction of label-augmented feature vector $\mathbf{z}_i$ can be seen as a multiplicative kernel $\mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j)$ computed individually from $\mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x)$ and $\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y)$ when using SVM for classification, we do not need to explicitly define the kernel function $\phi(\cdot)$ any longer. Therefore, instead of using the simple vector concatenation as in Eq. (1), we may devise a more discriminative kernel, which also provides another opportunity to incorporate more useful information for improved classification accuracy, such as the label correlations of a multi-label data set. Thus, in this work, we use a multiplicative kernel as:

$$\mathcal{K}_z(\mathbf{z}_i, \mathbf{z}_j) = \mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x)\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y), \tag{15}$$

where $\mathcal{K}_q(\mathbf{q}_i^x, \mathbf{q}_j^x) = \mathbf{q}_i^{x\,T}\mathbf{q}_j^x$ is a linear kernel on transformed input feature vectors and let

$$\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y) = \mathcal{K}_y \ . \tag{16}$$

We call the implicit transformation from $\mathbf{y}_i$ to $\mathbf{p}_i^y$ using $\mathcal{K}_p(\mathbf{p}_i^y, \mathbf{p}_j^y)$ defined in Eq. (16) as Correlative Kernel Transformation (CKT).

5 Initialization of Unlabeled Data

The ultimate goal of our algorithm is to predict labels $\mathbf{y}$ for unlabeled data points $\mathbf{x}_i$. Using our classification framework, we first need to initialize them to compute label-augmented feature vector $\mathbf{z}_i$ or the hybrid kernel $\mathcal{K}_z$. We can use any classification method to get the initialized labels $\hat{\mathbf{y}}_i$ for unlabeled data points. Although the initializations are not completely correct, a big portion of them are (assumed to be) correctly predicted. Our classification framework will self-consistently amend the incorrect labels. In this work, we use K-nearest neighbor (KNN) method for initialization because of its simplicity and clear intuition ($K = 1$ is used in this work and we abbreviate it as 1NN).

Another important point of the initialization step lies in that it provides an opportunity to make use of existing multi-label classification algorithms, *i.e.*, through the initialization step, the proposed MLFT approach can naturally benefit from the advantages of previous related works.

6 Motivation and Formulation of Label Correlations

Label correlations play a significant role in multi-label classification tasks, which are routinely utilized in most, if not all, existing multi-label classification algorithms as a primary mechanism to improve the overall classification performance. In this work, we also attempt to use them from the following three perspectives: attribute kernel as in Eq. (11), pairwise similarity as in Eq. (13) and label augmentation as in Eq. (15).

As the number of shared data points belonging to two classes measures how closely they are related, we use the cosine similarity to quantify label correlations. Let $Y = [\mathbf{y}_{(1)}, \ldots, \mathbf{y}_{(K)}]$, $\mathbf{y}_{(k)} \in \mathbb{R}^n (1 \leq k \leq K)$ is an n-vector, which is a class-wise label indicator vector for the kth class. Note that $\mathbf{y}_{(k)}$ is different from the data-point-wise label indicator vectors $\mathbf{y}_i$, which is a K-vector. We define the label correlation matrix, $C \in \mathbb{R}^{K \times K}$, to characterize label correlations as following:

$$C(k,l) = \cos(\mathbf{y}_{(k)}, \mathbf{y}_{(l)}) = \frac{\langle \mathbf{y}_{(k)}, \mathbf{y}_{(l)} \rangle}{\|\mathbf{y}_{(k)}\| \, \|\mathbf{y}_{(l)}\|} \ . \tag{17}$$

Using the TRECVID 2005 data set[1] with LSCOM-Lite annotation scheme [13], the label correlations defined in Eq. (17) is illustrated in Fig. 2. The high correlation value between "person" and "face" depicted in Fig. 2 shows that they are highly correlated, which perfectly agree with the common sense in real life due to the simplest fact that everybody has a face. Similar observations can be also found for "outdoor" and "sky", "waterscape-waterfront" and "boat-ship", "studio" and "TV-computer scree", "road" and "car", *etc.*, which concretely confirm the correctness of the formulation of label correlations defined in Eq. (17) at semantic level.

[1] http://www-nlpir.nist.gov/projects/trecvid/

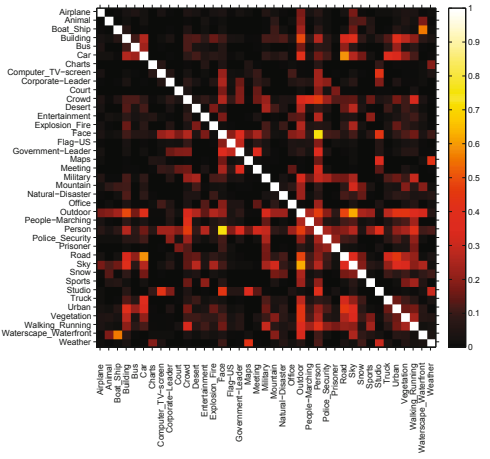


Fig. 2. Pairwise label correlations of 39 keywords in LSCOM-Lite on TRECVID 2005.

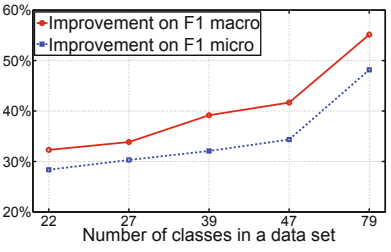


Fig. 3. Improvements (%) on F1 macro/micro average by MLFT algorithm over SVM using original features grows with the number of classes of multi-label data sets.

7 Experimental Evaluations

We evaluate the proposed MLFT algorithm using the following three standard multi-label image data sets.

TRECVID 2005 data set contains 137 broadcast news videos, which are segmented into 61901 sub-shots and labeled with 39 concepts according to LSCOM-Lite annotations [13]. We randomly sample the data set such that each concept (label) has at least 100 video key frames. We use 384-dimensional block-wise (64) color moments (mean and variable of each color band) as features.

Mediamill data set [14] includes 43907 sub-shots labeled with 101 classes (lexicon concepts), where each image are characterized by a 120-dimensional vector. Eliminating the classes containing less than 1000 data points, we have 27 classes in experiments. We randomly pick up 2609 sub-shots such that each class has at least 100 data points.

Corel natural scene data set [1] contains 2407 images represented by a 294-dimensional vector, which are labeled with 6 semantic concepts (labels).

Besides image annotation, we also extend our evaluation of the proposed algorithm to one more application in bioinformatics and use the following broadly used data set.

Yeast data set [4] is formed by micro-array expression data and phylogenetic profiles with 2417 genes. Each gene is expressed as a 107-dimensional vector, which is associated with at most 190 biological functions (labels) simultaneously. Filtering out the minor classes with small number of labeled genes, we end up with 14 labels.

Obviously, the number of features of every data set is much greater than the corresponding number of labels.

7.1 Evaluation Metrics for Multi-label Classification

The conventional classification performance metrics in statistical learning, *precision* and *F1 score*, are used to evaluate the proposed algorithms. For every class, the precision $p^{(k)}$ and F1 score $F_1^{(k)}$ for the kth class are computed following the standard definition for a binary classification problem. To address the multi-label scenario, as recommended in [11], macro average and micro average are used to assess the overall performance across multiple labels.

7.2 Multi-label Classification Performance

We use standard 5-fold cross validation to evaluate the classification performance of the proposed MLFT algorithm, and compare the experimental results with the most recent multi-label classification methods. We choose two label propagation based approaches: (1) Multi-Label Gaussian harmonic Function (MLGF) [17] method and (2) Semi-supervised learning by Sylvester Equation (SMSE) [2]; and two dimensionality reduction based approaches: (3) multi-label dimensionality reduction via Dependence Maximization (MDDM) [18] method and (4) Multi-Label Least Square (MLLS) [8] method. We use LIBSVM[2] to implement SVM throughout this paper, including the final classification step in the proposed MLFT approach. For MLGF and SMSE methods, we follow the detailed algorithms described in [17,2]. For MDDM, we use KNN for classification after dimensionality reduction, where 1NN classifiers are run one class at a time. We tried different 1NN, 3NN, 5NN, and the results are similar. Due to the limited space, we only report 1NN results. For MLLS, we use the codes posted on the authors' website [8].

We also run SVM directly using the original features of data sets, and report the classification results as baseline. The classification is conducted one class at a time, where every class is treated as a binary classification problem.

As mentioned in Section 2, instead of using MLKLE and CKF to obtain the balanced input vector $\mathbf{q}_i^x$ and $\mathbf{p}_i^y$ to construct label-augmented feature vector $\mathbf{z}_i$ using Eq. (1), we can also simply concatenate the original feature vectors $\mathbf{x}_i$ and label vector $\mathbf{y}_i$, we call this method as Naive Multi-label Feature Transform (NMLFT) method and report its results. In order to alleviate the unbalanced problem, we replace $\mathbf{p}_i^y$ by $\mu\mathbf{y}_i$, and empirically select $\mu = \sqrt{\frac{\sum_{i,j,i\neq j}\|\mathbf{y}_i-\mathbf{y}_j\|^2}{\sum_{i,j,i\neq j}\|\mathbf{x}_i-\mathbf{x}_j\|^2}}$ as SVM computes the decision hyperplane using Euclidean distance.

Table 3 presents the classification performance comparisons of the seven compared methods by 5-fold cross validation on the four multi-label data sets, which show that the proposed MLFT constantly outperforms the other methods. This demonstrates that the mapping of training data points from the original feature space into the transformed label augmented feature space through the proposed MLFT algorithm is generalizable to the test data, and the classification performance is hence improved for multi-label classification tasks. In addition, the

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 3. Performance evaluation of the seven compared methods on the four multi-label data sets by 5-fold cross validation

Datasets	Evaluation metrics		Compared methods						
			SVM	MLGF	SMSE	MDDM	MLLS	NMLFT	MLFT
TRECVID 2005 (39 classes)	Macro avg.	Precision	0.269	0.108	0.107	0.366	0.272	0.243	0.421
		F1 score	0.236	0.151	0.150	0.370	0.275	0.330	0.398
	Micro avg.	Precision	0.252	0.107	0.107	0.352	0.279	0.339	0.420
		F1 score	0.371	0.167	0.165	0.491	0.375	0.483	0.527
Mediamill (27 classes)	Macro avg.	Precision	0.301	0.204	0.205	0.385	0.307	0.376	0.395
		F1 score	0.302	0.206	0.213	0.389	0.314	0.380	0.431
	Micro avg.	Precision	0.297	0.201	0.199	0.382	0.304	0.369	0.388
		F1 score	0.459	0.304	0.301	0.541	0.470	0.518	0.572
Yeast (14 classes)	Macro avg.	Precision	0.657	0.564	0.670	0.794	0.689	0.747	0.824
		F1 score	0.114	0.107	0.109	0.147	0.129	0.132	0.154
	Micro avg.	Precision	0.826	0.652	0.675	0.854	0.827	0.828	0.885
		F1 score	0.139	0.124	0.128	0.160	0.147	0.149	0.168
Corel natural scene (6 classes)	Macro avg.	Precision	0.582	0.341	0.362	0.661	0.588	0.642	0.691
		F1 score	0.542	0.410	0.415	0.667	0.545	0.651	0.687
	Micro avg.	Precision	0.591	0.404	0.410	0.669	0.593	0.650	0.693
		F1 score	0.581	0.421	0.431	0.675	0.585	0.662	0.690

experimental results also show that MLFT is always superior to NMLFT, which testifies that the balanced label augmentation using the transformed input vectors produced by the proposed MLKLE and CKT is indispensable for an effective feature transformation.

In addition, Fig. 4 shows the class-wise classification performance measured by precision on TRECVID 2005 data set (as an example, because we can not show the results on all the data sets due space limit). The results show that, besides the overall performance as listed in Table 3, the proposed MLFT approach consistently outperform the other approaches in most of the individual functional classes, which again confirms the effectiveness of the proposed algorithms.

7.3 Label Enhancement in Multi-label Classification

A more careful analysis shows that the performance improvements of MLFT algorithm (measured by macro/micro average of F1 scores) over those of SVM using original features grow with the number of classes in a data set. The results are summarized in Fig. 3. Because F1 score is a balanced measurement over precision and recall via the harmonic mean, it is more representative for classification performance assessment. Therefore, we tentatively conclude that the benefit of using label augmentation is proportional to the number of classes in a multi-label data set. When the number of labels in a data set is larger, more label correlations are included in $\mathbf{z}_i$ to amend incorrect predictions in initialization. The extreme case is in single-label classification or even binary classification,

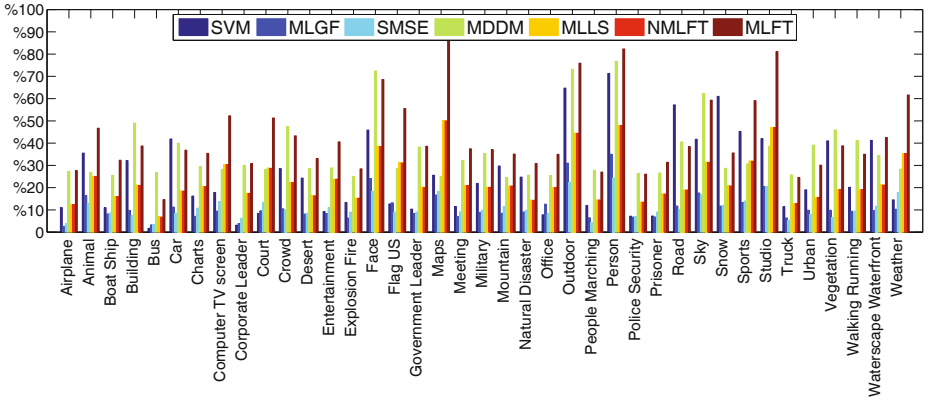


Fig. 4. Class-wise precisions of compared approaches on TRECVID 2005 data set

MLFT algorithm usually does not benefit from the augmentation by the label vector $\mathbf{y}_i$ because everything depends solely on exactly one label prediction. On the other hand, as shown in Fig. 3, as long as the number of classes of a data set is not small, MLFT algorithm always exhibits satisfying classification performance.

8 Conclusions

In this work, we first revealed that label assignments in multi-label classification not only indicate class membership of data points, but also convey very important characteristic information to assess similarity among data points from *knowledge* perspective. We then proposed a novel Multi-Label Feature Transform (MLFT) approach to use label assignments explicitly as part of data attributes and implicitly to formulate label correlations. Through two transformations on data attributes (via MLKLE) and label assignments (via CKT) respectively, the transformed input feature vector and input label vector have similar dimensionality, such that they can be integrated to form the label-augmented feature vector in a balanced manner. Data discriminability in the spanned space is thereby enhanced by taking advantage of the information coming from both data and knowledge perspectives. Moreover, although the proposed MLFT approach is originated from simple vector concatenation, by introducing kernel utility we may avert the explicit construction of label-augmented feature vector and thereby use a discriminative kernel to utilize data in a more flexible and effective manner. Extensively empirical evaluations are conducted on five standard multi-label data sets. Promising experimental results have demonstrated the effectiveness of our approach.

Acknowledgments. This research is supported by NSF-CCF 0830780, NSF-CCF 0939187, NSF-CCF 0917274, NSF-DMS 0915228, NSF-CNS 0923494.

References

1. Boutell, M., Luo, J., Shen, X., Brown, C.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004)
2. Chen, G., Song, Y., Wang, F., Zhang, C.: Semi-supervised Multi-label Learning by Solving a Sylvester Equation. In: Jonker, W., Petković, M. (eds.) SDM 2008. LNCS, vol. 5159, Springer, Heidelberg (2008)
3. Chung, F.: Spectral graph theory. Amer. Mathematical Society, Providence (1997)
4. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Proc of NIPS (2001)
5. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 22–30. Springer, Heidelberg (2004)
6. Griffiths, T., Ghahramani, Z.: Infinite latent feature models and the Indian buffet process. In: Proc. of NIPS (2006)
7. Hall, K.: An r-dimensional quadratic placement algorithm. Management Science, 219–229 (1970)
8. Ji, S., Tang, L., Yu, S., Ye, J.: Extracting shared subspace for multi-label classification. In: Proc of SIGKDD (2008)
9. Jolliffe, I.: Principal component analysis. Springer, Heidelberg (2002)
10. Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: Proc of CVPR, pp. 1719–1726 (2006)
11. Lewis, D., Yang, Y., Rose, T., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
12. Liu, Y., Jin, R., Yang, L.: Semi-supervised multi-label learning by constrained non-negative matrix factorization. In: Proc. of AAAI, p. 421 (2006)
13. Naphade, M., Kennedy, L., Kender, J., Chang, S., Smith, J., Over, P., Hauptmann, A.: LSCOM-lite: A light scale concept ontology for multimedia understanding for TRECVID 2005. Tech. rep., Technical report, IBM Research Tech. Report, RC23612, W0505-104 (2005)
14. Snoek, C.G.M., Worring, M., van Gemert, J.C., Geusebroek, J.M., Smeulders, A.W.M.: The challenge problem for automated detection of 101 semantic concepts in multimedia. In: Proc. of ACM Multimedia (2006)
15. Wang, H., Huang, H., Ding, C.: Image Annotation Using Multi-label Correlated Greens Function. In: Proc. of ICCV, pp. 2029–2034 (2009)
16. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proc of SIGIR (2005)
17. Zha, Z., Mei, T., Wang, J., Wang, Z., Hua, X.: Graph-based semi-supervised learning with multi-label. In: Proc. of IEEE ICME, pp. 1321–1324 (2008)
18. Zhang, Y., Zhou, Z.: Multi-Label Dimensionality Reduction via Dependence Maximization. In: Proc of AAAI, pp. 1503–1505 (2008)
19. Zhu, S., Ji, X., Xu, W., Gong, Y.: Multi-labelled classification using maximum entropy method. In: Proc. of SIGIR, pp. 274–281 (2005)

Author Index