Kostas Daniilidis
Petros Maragos
Nikos Paragios (Eds.)

LNCS 6316

Computer Vision – ECCV 2010

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

6 Part VI

ECCV 2010
Crete-Greece

Springer

Lecture Notes in Computer Science 6316

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 VI

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	3-642-15566-9 Springer Berlin Heidelberg New York
ISBN-13	978-3-642-15566-6 Springer Berlin Heidelberg New York

springer.com

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

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

Gold Sponsors

Google

Microsoft Research

technicolor

Silver Sponsors

Adobe

DynaVox Mayer-Johnson
Advancing human expression and learning.

ERCIM
European Research Consortium
for Informatics and Mathematics

GE

IBM

Johnson Controls

POINT GREY

SIEMENS

UNIVERSITY of HOUSTON

Table of Contents – Part VI

Visual Learning

Spotlights and Posters R2

Constrained Spectral Clustering via Exhaustive and Efficient Constraint Propagation

Zhiwu Lu and Horace H.S. Ip

Department of Computer Science, City University of Hong Kong, Hong Kong
AIMtech Centre, City University of Hong Kong, Hong Kong
lzhiwu2@student.cityu.edu.hk, cship@cityu.edu.hk

Abstract. This paper presents an exhaustive and efficient constraint propagation approach to exploiting pairwise constraints for spectral clustering. Since traditional label propagation techniques cannot be readily generalized to propagate pairwise constraints, we tackle the constraint propagation problem inversely by decomposing it to a set of independent label propagation subproblems which are further solved in quadratic time using semi-supervised learning based on k-nearest neighbors graphs. Since this time complexity is proportional to the number of all possible pairwise constraints, our approach gives a computationally efficient solution for exhaustively propagating pairwise constraint throughout the entire dataset. The resulting exhaustive set of propagated pairwise constraints are then used to adjust the weight (or similarity) matrix for spectral clustering. It is worth noting that this paper first clearly shows how pairwise constraints are propagated independently and then accumulated into a conciliatory closed-form solution. Experimental results on real-life datasets demonstrate that our approach to constrained spectral clustering outperforms the state-of-the-art techniques.

1 Introduction

Cluster analysis is largely driven by the quest for more robust clustering algorithms capable of detecting clusters with diverse shapes and densities. It is worth noting that data clustering is an ill-posed problem when the associated objective function is not well defined, which leads to fundamental limitations of generic clustering algorithms. Multiple clustering solutions may seem to be equally plausible due to an inherent arbitrariness in the notion of a cluster. Therefore, any additional supervisory information must be exploited in order to reduce this degeneracy of possible solutions and improve the quality of clustering. The labels of data are potential sources of such supervisory information which has been widely used. In this paper, we consider a commonly adopted and weaker type of supervisory information, called pairwise constraints which specify whether a pair of data belongs to the same cluster or not.

There exist two types of pairwise constraints, known as *must-link* constraints and *cannot-link* constraints, respectively. We can readily derive such pairwise constraints from the labels of data, where a pair of data with the same label

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 1–14, 2010.

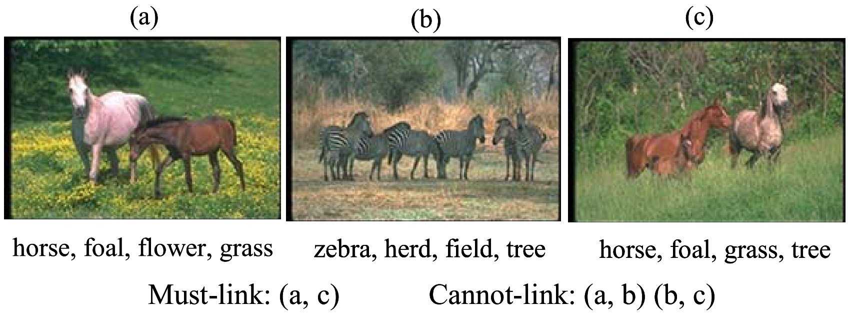


Fig. 1. The must-link and cannot-link constraints derived from the annotations of images. Since we focus on recognizing the objects of interests in images, these constraints are formed without considering the backgrounds such as tree, grass, and field.

denotes must-link constraint and cannot-link constraint otherwise. It should be noted, however, that the inverse may not be true, i.e. in general we cannot infer the labels of data from pairwise constraints, particularly for multi-class data. This implies that pairwise constraints are inherently weaker but more general than the labels of data. Moreover, pairwise constraints can also be automatically derived from domain knowledge [1,2] or through machine learning. For example, we can obtain pairwise constraints from the annotations of the images shown in Fig. 1. Since we focus on recognizing the objects of interests (e.g. horse and zebra) in images, the pairwise constraints can be formed without considering the backgrounds such as tree, grass, and field. In practice, the objects of interest can be roughly distinguished from the backgrounds according to the ranking scores of annotations learnt automatically by an image search engine.

Pairwise constraints have been widely used for constrained clustering [1,2,3,4,5], and it has been reported that the use of appropriate pairwise constraints can often lead to the improved quality of clustering. In this paper, we focus on the exploitation of pairwise constraints for spectral clustering [6,7,8,9] which constructs a new low-dimensional data representation for clustering using the leading eigenvectors of the similarity matrix. Since pairwise constraints specify whether a pair of data belongs to the same cluster, they provide a source of information about the data relationships, which can be readily used to adjust the similarities between the data for spectral clustering. In fact, the idea of exploiting pairwise constraints for spectral clustering has been studied previously. For example, [10] trivially adjusted the similarities between the data to 1 and 0 for must-link and cannot-link constraints, respectively. This method only adjusts the similarities between constrained data. In contrast, [11] propagated pairwise constraints to other similarities between unconstrained data using Gaussian process. However, as noted in [11], this method makes certain assumptions for constraint propagation specially with respect to two-class problems, although the heuristic approach for multi-class problems is also discussed. Furthermore, such constraint propagation is formulated as a semi-definite programming (SDP) problem in [12]. Although the method is not limited to two-class problems, it incurs extremely large computational cost for solving the SDP problem. In [13], the constraint propagation is also formulated as a constrained optimization problem, but only must-link constraints can be used for optimization.

To overcome these problems, we propose an exhaustive and efficient constraint propagation approach to exploiting pairwise constraints for spectral clustering, which is not limited to two-class problems or using only must-link constraints. Specifically, since traditional label propagation techniques [14,15,16] cannot be readily generalized to propagate pairwise constraints, we tackle the constraint propagation problem inversely by decomposing it to a set of independent label propagation subproblems. Furthermore, we show that through semi-supervised learning based on k-nearest neighbors graphs, the set of label propagation subproblems can be solved in quadratic time $O(kN^2)$ with respect to the data size N $(k \ll N)$. Since this time complexity is proportional to the total number of all possible pairwise constraints (i.e. $N(N-1)/2$), our constraint propagation approach can be considered computationally efficient. It is worth noting that our approach incurs much less computational cost than [12], given that SDP-based constraint propagation has a time complexity of $O(N^4)$.

The resulting exhaustive set of propagated pairwise constraints can be exploited for spectral clustering through adjusting the similarity matrix with this information. The experimental results on image and UCI datasets demonstrate that our approach outperforms the state-of-the-art techniques. It is worth noting that our approach can be seen as a very general constraint propagation technique, which has the following advantages:

(1) This is the first constraint propagation approach that clearly shows how pairwise constraints are propagated independently and then accumulated into a *conciliatory closed-form solution.*

(2) Our approach is not limited to two-class problems or using only must-link constraints. More importantly, *our approach allows soft constraints*, i.e., the pairwise constraints can be associated with confidence scores like [17,18].

(3) The exhaustive set of pairwise constraints obtained by our approach can also potentially be used to improve the performance of other machine learning techniques by adjusting the similarity matrix.

The remainder of this paper is organized as follows. In Section 2, we propose an exhaustive and efficient constraint propagation approach. In Section 3, we exploit the exhaustive set of propagated pairwise constraints for spectral clustering. In Section 4, our approach is evaluated on image and UCI datasets. Finally, Section 5 gives the conclusions drawn from experimental results.

2 Exhaustive and Efficient Constraint Propagation

Given a dataset $\mathcal{X} = \{x_1, ..., x_N\}$, we denote a set of must-link constraints as $\mathcal{M} = \{(x_i, x_j) : z_i = z_j\}$ and a set of cannot-link constraints as $\mathcal{C} = \{(x_i, x_j) : z_i \neq z_j\}$, where z_i is the label of data x_i. Our goal is to exploit the two types of

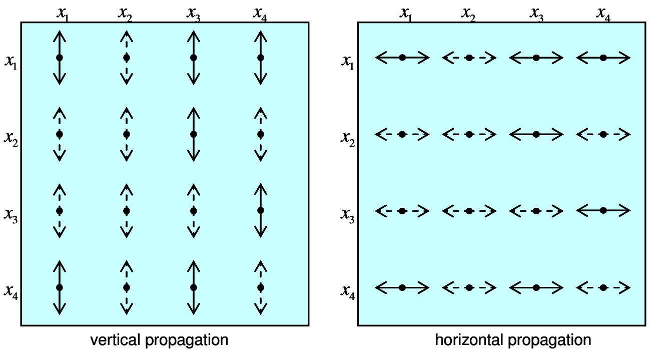


Fig. 2. The vertical and horizontal propagation of pairwise constraints. Each arrow denotes the direction of constraint propagation. The solid arrow means that the pairwise constraint is provided initially, while the dashed arrow means that the pairwise constraint is newly generated during constraint propagation.

pairwise constraints for spectral clustering on the dataset $\mathcal{X}$. As we have mentioned, the pairwise constraints can be used to adjust the similarities between data so that spectral clustering can be performed with the adjusted similarity matrix. In previous work [10], only the similarities between the constrained data are adjusted, and thus the pairwise constraints exert very limited effect on the subsequent spectral clustering. In the following, we propose an exhaustive and efficient constraint propagation technique that spreads the effect of pairwise constraints throughout the entire dataset, thereby enabling the pairwise constraints to exert a stronger influence on the subsequent spectral clustering.

A main obstacle of constraint propagation lies in that the cannot-link constraints are not transitive. In this paper, however, we succeed in propagating both must-link and cannot-link constraints. We first represent these two types of pairwise constraints using a single matrix $Z = \{Z_{ij}\}_{N\times N}$:

$$Z_{ij} = \begin{cases} +1, & (x_i, x_j) \in \mathcal{M}; \\ -1, & (x_i, x_j) \in \mathcal{C}; \\ 0, & \text{otherwise}. \end{cases} \tag{1}$$

Here, we have $|Z_{ij}| \leq 1$ for soft constraints [17,18]. Since we can directly obtain the pairwise constraints from the above matrix Z, the pairwise constraints have been represented using Z without loss of information. We make further observations on Z column by column. It can be observed that the j-th column $Z_{.j}$ actually provides the initial configuration of a *two-class semi-supervised learning problem* with respect to x_j, where the "positive class" contains the data that must appear together with x_j and the "negative class" contains the data that cannot appear together with x_j. More concretely, x_i can be initially regarded as coming from the positive (or negative) class if $Z_{ij} > 0$ (or < 0), but if x_i and x_j are not constrained (i.e. $Z_{ij} = 0$) thus x_i is initially unlabeled. This configuration of a two-class semi-supervised learning is also suitable for soft constraints. The semi-supervised learning problem with respect to x_j can be solved by the label propagation technique [14]. Since the other columns of Z can be handled similarly, we can decompose the constraint propagation problem into N independent label propagation subproblems which can then be solved in parallel. The vertical propagation of pairwise constraints is illustrated in Fig. 2.

However, it is also possible that a column contains no pairwise constraints (for example, see the second column in Fig. 2). That is, the entries of this column may all be zeros, and for such cases, no constraint propagation will occur along this column. We deal with this problem through horizontal constraint propagation (see Fig. 2), which is performed after the vertical constraint propagation. The horizontal propagation can be done similar to the vertical propagation discussed above. The only difference is that we now consider Z row by row, instead of column-wise. More significantly, through combining the vertical and horizontal constraint propagation, we succeed in propagating the pairwise constraints to any pair of data. That is, the semi-supervised learning for constraint propagation could not break down if one type of constraints is missing for some data.

The two sets of constraint propagation subproblems can be solved efficiently through semi-supervised learning based on k-nearest neighbors graphs. Let $\mathcal{F} = \{F = \{F_{ij}\}_{N\times N} : |F_{ij}| \leq 1\}$. In fact, each matrix $F \in \mathcal{F}$ denotes a set of pairwise constraints with the associated confidence scores. That is, $F_{ij} > 0$ is equivalent to $(x_i, x_j) \in \mathcal{M}$ while $F_{ij} < 0$ is equivalent to $(x_i, x_j) \in \mathcal{C}$, with $|F_{ij}|$ being the confidence score (i.e. probability) of $(x_i, x_j) \in \mathcal{M}$ or $(x_i, x_j) \in \mathcal{C}$. Particularly, $Z \in \mathcal{F}$, where Z collects the initial pairwise constraints. Given the affinity (or similarity) matrix A for the dataset $\mathcal{X}$, our algorithm for constraint propagation is summarized as follows:

(1) Form the weight matrix W of a graph by $W_{ij} = \frac{A(x_i,x_j)}{\sqrt{A(x_i,x_i)}\sqrt{A(x_j,x_j)}}$ if x_j ($j \neq i$) is among the k-nearest neighbors (k-NN) of x_i and $W_{ij} = 0$ otherwise. Set $W = (W + W^T)/2$ to ensure that W is symmetric.

(2) Construct the matrix $\bar{\mathcal{L}} = D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th row of W.

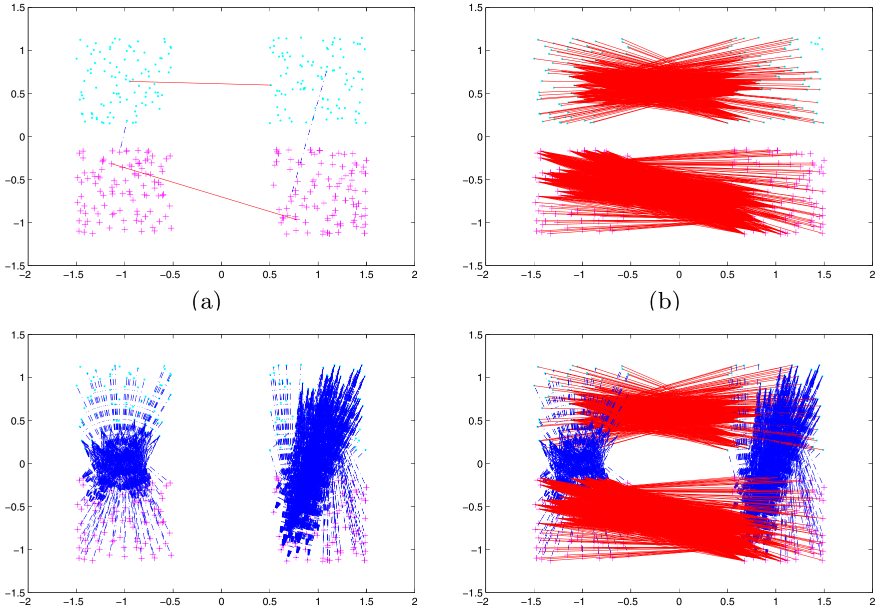
(3) Iterate $F_v(t+1) = \alpha\bar{\mathcal{L}}F_v(t) + (1-\alpha)Z$ for vertical constraint propagation until convergence, where $F_v(t) \in \mathcal{F}$ and α is a parameter in the range $(0,1)$.

(4) Iterate $F_h(t+1) = \alpha F_h(t)\bar{\mathcal{L}} + (1-\alpha)F_v^*$ for horizontal constraint propagation until convergence, where $F_h(t) \in \mathcal{F}$ and F_v^* is the limit of $\{F_v(t)\}$.

(5) Output $F^* = F_h^*$ as the final representation of the pairwise constraints, where F_h^* is the limit of $\{F_h(t)\}$.

Below we give a convergence analysis of the above constraint propagation algorithm. Since the vertical constraint propagation in Step (3) can be regarded as label propagation, its convergence has been shown in [14]. More concretely, similar to [14], we can obtain $F_v^* = (1-\alpha)(I-\alpha\bar{\mathcal{L}})^{-1}Z$ as the limit of $\{F_v(t)\}$. As for the horizontal constraint propagation, we have

$$\begin{aligned} F_h^T(t+1) &= \alpha\bar{\mathcal{L}}^T F_h^T(t) + (1-\alpha){F_v^*}^T \\ &= \alpha\bar{\mathcal{L}}F_h^T(t) + (1-\alpha){F_v^*}^T. \end{aligned} \tag{2}$$



(a) (b)

(c) (d)

Fig. 3. The illustration of our constraint propagation: (a) four pairwise constraints and ideal clustering of the data; (b) final constraints propagated from only two must-link constraints; (c) final constraints propagated from only two cannot-link constraints; (d) final constraints propagated from four pairwise constraints. Here, must-link constraints are denoted by solid red lines, while cannot-link constraints are denoted by dashed blue lines. Moreover, we only show the propagated constraints with predicted confidence scores > 0.1 in Figs. 3(b)-3(d).

That is, the horizontal propagation in Step (4) can be transformed to a vertical propagation which converges to ${F_h^*}^T = (1-\alpha)(I-\alpha\bar{\mathcal{L}})^{-1}{F_v^*}^T$. Hence, our constraint propagation algorithm has the following closed-form solution:

$$\begin{aligned} F^* = F_h^* &= (1-\alpha)F_v^*(I-\alpha\bar{\mathcal{L}}^T)^{-1} \\ &= (1-\alpha)^2(I-\alpha\bar{\mathcal{L}})^{-1}Z(I-\alpha\bar{\mathcal{L}})^{-1}, \end{aligned} \tag{3}$$

which actually accumulates the evidence to reconcile the contradictory propagated constraints for certain pairs of data. As a toy example, the propagated constraints given by the above equation are explicitly shown in Fig. 3. We can find that the propagated constraints obtained by our approach are consistent with the ideal clustering of the data.

Finally, we give a complexity analysis of our constraint propagation algorithm. Through semi-supervised learning based on k-nearest neighbors graphs ($k \ll N$), both vertical and horizontal constraint propagation can be performed in quadratic time $O(kN^2)$. Since this time complexity is proportional to the total number of all possible pairwise constraints (i.e. $N(N-1)/2$), our algorithm can be considered computationally efficient. Moreover, our algorithm incurs significantly less computational cost than [12], given that constraint propagation based on semi-definite programming has a time complexity of $O(N^4)$.

3 Fully Constrained Spectral Clustering

It should be noted that the output F^* of our constraint propagation algorithm represents an exhaustive set of pairwise constraints with the associated confidence scores $|F^*|$. Our goal is to obtain a data partition that is fully consistent with F^*. Here, we exploit F^* for spectral clustering by adjusting the weight matrix W as follows:

$$\tilde{W}_{ij} = \begin{cases} 1-(1-F^*_{ij})(1-W_{ij}), & F^*_{ij} \geq 0; \\ (1+F^*_{ij})W_{ij}, & F^*_{ij} < 0. \end{cases} \tag{4}$$

In the following, $\tilde{W}$ will be used for constrained spectral clustering. Here, we need to first prove that this matrix can be regarded as a weight matrix by showing that $\tilde{W}$ has the following nice properties.

Proposition 1. *(i) $\tilde{W}$ is nonnegative and symmetric; (ii) $\tilde{W}_{ij} \geq W_{ij}$ (or $< W_{ij}$) if $F^*_{ij} \geq 0$ (or < 0).*

Proof. The above proposition is proven as follows:

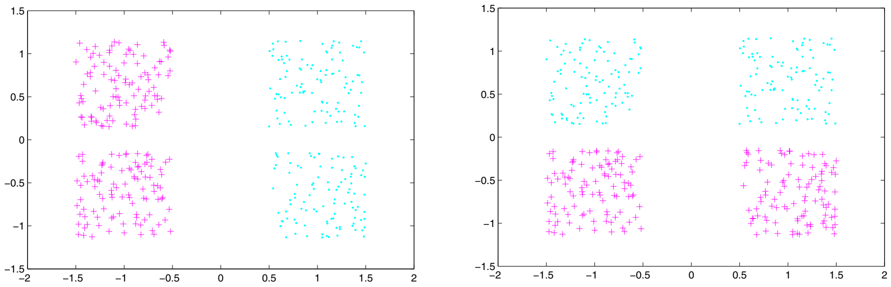
(i) The symmetry of both W and F^* ensures that $\tilde{W}$ is symmetric. Since $0 \leq W_{ij} \leq 1$ and $|F^*_{ij}| \leq 1$, we also have: $\tilde{W}_{ij} = 1-(1-F^*_{ij})(1-W_{ij}) \geq 1-(1-W_{ij}) \geq 0$ if $F^*_{ij} \geq 0$ and $\tilde{W}_{ij} = (1+F^*_{ij})W_{ij} \geq 0$ if $F^*_{ij} < 0$. That is, we always have $\tilde{W}_{ij} \geq 0$. Hence, $\tilde{W}$ is nonnegative and symmetric.

(ii) According to (4), we can consider $\tilde{W}_{ij}$ as a *monotonically increasing function* of F^*_{ij}. Since $\tilde{W}_{ij} = W_{ij}$ when $F^*_{ij} = 0$, we thus have: $\tilde{W}_{ij} \geq W_{ij}$ (or $< W_{ij}$) if $F^*_{ij} \geq 0$ (or < 0).

This proves that $\tilde{W}$ can be used as a weight matrix for spectral clustering. More importantly, according to Proposition 1, the new weight matrix $\tilde{W}$ is derived from the original weight matrix W by increasing W_{ij} for the must-link constraints with $F^*_{ij} > 0$ and decreasing W_{ij} for the cannot-link constraints with $F^*_{ij} < 0$. This is entirely consistent with our original motivation of exploiting pairwise constraints for spectral clustering.

After we have incorporated the exhaustive set of pairwise constraints obtained by our constraint propagation into a new weight matrix $\tilde{W}$, we then perform spectral clustering with this weight matrix. The corresponding algorithm is summarized as follows:

(1) Find K largest nontrivial eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_K$ of $\tilde{D}^{-1/2}\tilde{W}\tilde{D}^{-1/2}$, where $\tilde{D}$ is a diagonal matrix with its (i,i)-element equal to the sum of the i-th row of the weight matrix $\tilde{W}$.



(a)

(b)

Fig. 4. The results of constrained clustering on the toy data using four pairwise constraints given by Fig. 3(a): (a) spectral learning [10]; (b) our approach. The clustering obtained by our approach is consistent with the ideal clustering of the data.

(2) Form $E = [\mathbf{v}_1, ..., \mathbf{v}_K]$, and normalize each row of E to have unit length. Here, the i-th row $E_{i.}$ is the low-dimensional feature vector for data x_i.

(3) Perform k-means clustering on the new feature vectors $E_{i.} (i = 1, ..., N)$ to obtain K clusters.

The clustering results on the toy data (see Fig. 3(a)) by the above algorithm are shown in Fig. 4(b). We can find that the clustering obtained by our approach is consistent with the ideal clustering of the data, while this is not true for spectral learning [10] without using constraint propagation (see Fig. 4(a)). In the following, since the pairwise constraints used for constrained spectral clustering (CSC) is obtained by our exhaustive and efficient constraint propagation (E^{2}CP), the above associated clustering algorithm is denoted as E^{2}CSC (or E^{2}CP directly) to distinguish it from other CSC algorithms.

4 Experimental Results

In this section, we conduct extensive experiments on real-life data to evaluate the proposed constrained spectral clustering algorithm. We first describe the experimental setup, including the clustering evaluation measure and the parameter selection. Moreover, we compare our algorithm with other closely related methods on two image datasets and four UCI datasets, respectively.

4.1 Experimental Setup

For comparison, we present the results of affinity propagation (AP) [11], spectral learning (SL) [10] and semi-supervised kernel k-means (SSKK) [4], which are three closely related constrained clustering algorithms. SL and SSKK adjust only the similarities between the constrained data, while AP and our E^{2}CP propagate the pairwise constraints throughout the entire dataset. Here, it should be noted that AP cannot directly address multi-class problems and we have to take into

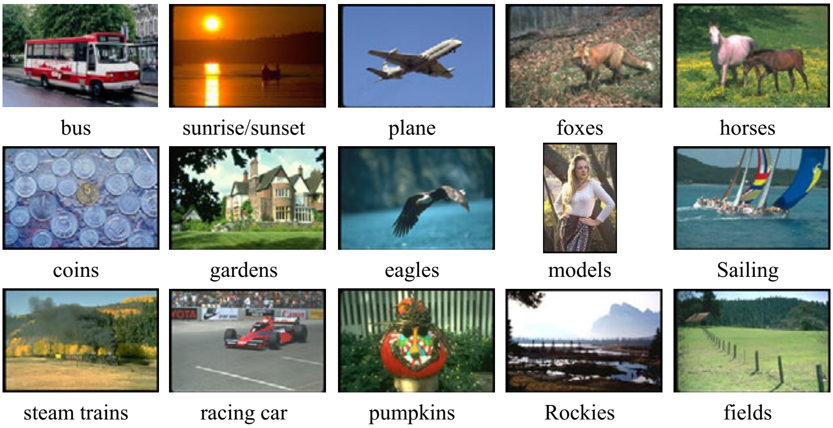


Fig. 5. Sample images from 15 categories of the Corel dataset

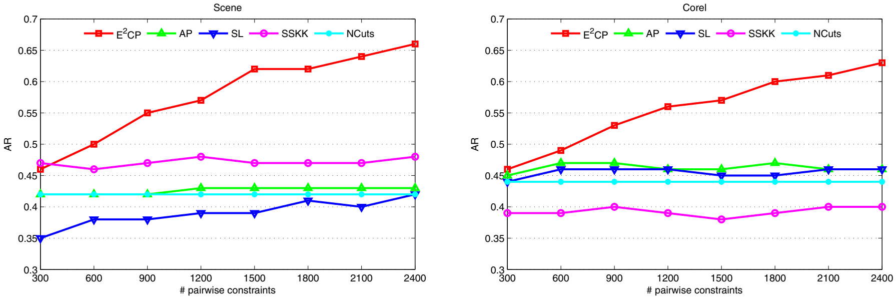
account the heuristic approach discussed in [11]. We also report the baseline results of normalized cuts (NCuts) [8], which is effectively a spectral clustering algorithm but without using pairwise constraints.

We evaluate the clustering results with the adjusted Rand (AR) index [19,20,21], which has been widely used for the evaluation of clustering algorithms. The AR index measures the pairwise agreement between the computed clustering and the ground truth clustering, and takes a value in the range [-1,1]. A higher AR index indicates that a higher percentage of data pairs in the obtained clustering have the same relationship (musk-link or cannot-link) as in the ground truth clustering. In the following, each experiment is randomly run 25 times, and the average AR index is obtained as the final clustering evaluation measure.

We set $\alpha = 0.8$ and $k = 20$ for our E^{2}CP algorithm. The k-NN graph constructed for our constraint propagation is also used for the subsequent spectral clustering. To ensure a fair comparison, we adopt the same k-NN graph for the other algorithms. Here, we construct the graph with different kernels for image and UCI datasets. That is, the spatial Markov kernel [15] is defined on the image datasets to exploit the spatial information, while the Gaussian kernel is used for the UCI datasets as in [11]. For each dataset, different numbers of pairwise constraints are randomly generated using the ground-truth cluster labels.

4.2 Results on Image Datasets

We select two different image datasets. The first one contains 8 scene categories from MIT [22], including four man-made scenes and four natural scenes. The total number of images is 2,688. The size of each image in this Scene dataset is 256×256 pixels. The second dataset contains images from a Corel collection. We select 15 categories (see Fig. 5), and each of the categories contains 100 images. In total, this selected set has 1,500 images. The size of each image in this dataset is 384×256 or 256×384 pixels.



(a) (b)

Fig. 6. The clustering results on the two image datasets by different clustering algorithms with a varying number of pairwise constraints

For these two image datasets, we choose two different feature sets which are introduced in [23] and [15], respectively. That is, as in [23], the SIFT descriptors are used for the Scene dataset, while, similar to [15], the joint color and Gabor features are used for the Corel dataset. These features are chosen to ensure a fair comparison with the state-of-the-art techniques. More concretely, for the Scene dataset, we extract SIFT descriptors of 16×16 pixel blocks computed over a regular grid with spacing of 8 pixels. As for the Corel dataset, we divide each image into blocks of 16×16 pixels and then extract a joint color/texture feature vector from each block. Here, the texture features are represented as the means and standard deviations of the coefficients of a bank of Gabor filters (with 3 scales and 4 orientations), and the color features are the mean values of HSV color components. Finally, for each image dataset, we perform k-means clustering on the extracted feature vectors to form a vocabulary of 400 visual keywords. Based on this visual vocabulary, we then define a spatial Markov kernel [15] as the weight matrix for graph construction.

In the experiments, we provide the clustering algorithms with a varying number of pairwise constraints. The clustering results are shown in Fig. 6. We can find that our E^{2}CP generally performs the best among the five clustering methods. The effectiveness of our exhaustive constraint propagation approach to exploiting pairwise constraints for spectral clustering is verified by the fact that our E^{2}CP consistently obtains better results. In contrast, SL and SSKK perform unsatisfactorily, and, in some cases, their performance has been degraded to those of NCuts. This may be due to that by merely adjusting the similarities only between the constrained images, these approaches have not fully utilized the additional supervisory or prior information inherent in the constrained images, and hence can not discover the complex manifolds hidden in the challenging image datasets. Although AP can also propagate pairwise constraints throughout the entire dataset like our E^{2}CP, the heuristic approach discussed in [11] may not address multi-class problems for the challenging image datasets, which leads to unsatisfactory results. Moreover, another important observation is that the improvement in the clustering performance by our E^{2}CP with respect to NCuts becomes more obvious when more pairwise constraints are provided, while this

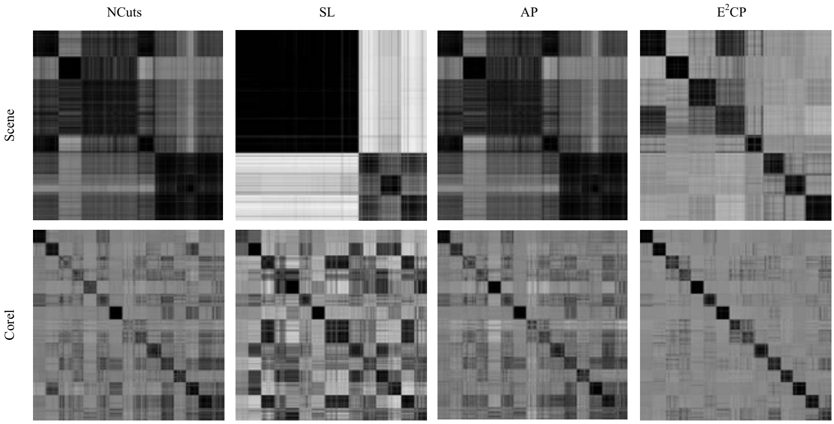


Fig. 7. Distance matrices of the low-dimensional data representations for the two image datasets obtained by NCuts, SL, AP, and E^{2}CP, respectively. For illustration purpose, the data are arranged such that images within a cluster appear consecutively. The darker is a pixel, the smaller is the distance.

is not the case for AP, SL or SSKK. In other words, the pairwise constraints has been exploited more exhaustively and effectively by our E^{2}CP.

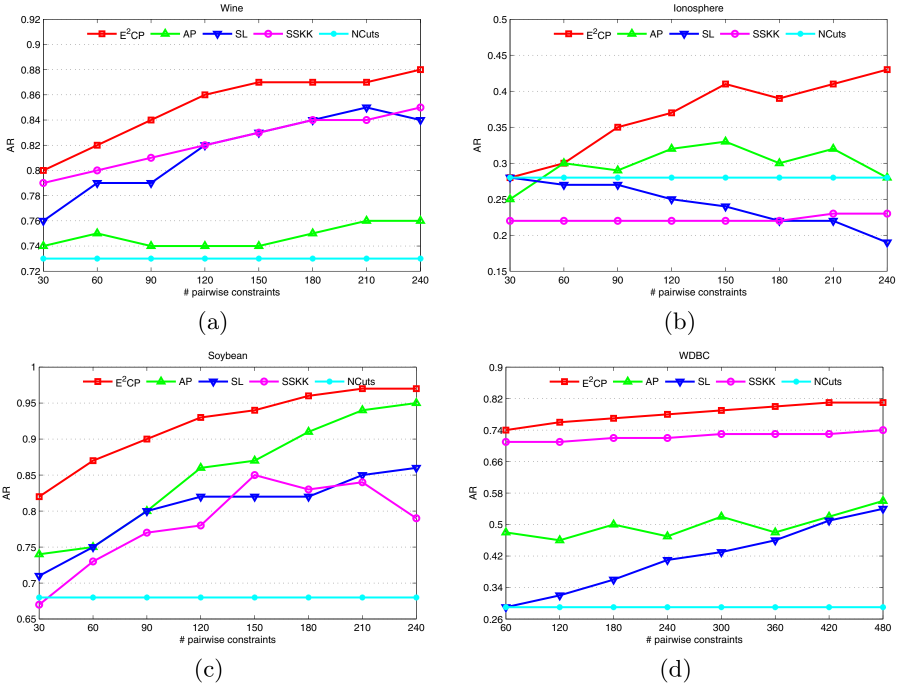
To make it clearer how our E^{2}CP exploits the pairwise constraints for spectral clustering, we show the distance matrices of the low-dimensional data representations obtained by NCuts, SL, AP, and E^{2}CP in Fig. 7. We can find that the block structure of the distance matrices of the data representations obtained by our E^{2}CP on the two image datasets is significantly more obvious, as compared to those of the data representations obtained by NCuts, SL, and AP. This means that after being adjusted by our E^{2}CP, each cluster associated with the new data representation becomes more compact and different clusters become more separated. Hence, we can conclude that our E^{2}CP does lead to better spectral clustering through our exhaustive constraint propagation.

The pairwise constraints used here are actually very sparse. For example, the largest number of pairwise constraints (i.e. 2,400) used for constrained clustering are generated with only 2.6% of the images in the Scene dataset. Here, images from the same cluster form the must-link constraints while images from different clusters form the cannot-link constraints. Through our exhaustive constraint propagation, we obtain 3,611,328 pairwise constraints with nonzero confidence scores from this sparse set of pairwise constraints. That is, we have successfully propagated 2,400 pairwise constraints throughout the entire dataset.

It is noteworthy that the running time of our E^{2}CP is comparable to that of the constrained clustering algorithms without using constraint propagation (e.g. SL and NCuts). Moreover, as for the two constraint propagation approaches, our E^{2}CP runs faster than AP, particularly for multi-class problems. For example, the time taken by E^{2}CP, AP, SL, SSKK, and NCuts on the Scene dataset is 20, 42, 15, 17, and 12 seconds, respectively. We run all the five algorithms (Matlab code) on a PC with 2.33 GHz CPU and 2GB RAM.

4.3 Results on UCI Datasets

We further conduct experiments on four UCI datasets, which are described in Table 1. The UCI data are widely used to evaluate clustering and classification algorithms in machine learning. Here, as in [11], the Gaussian kernel is defined on each UCI dataset for computing the weight matrix during graph construction. The experimental setup on the UCI datasets is similar to that for the image datasets. The clustering results are shown in Fig. 8.



(a) (b)

(c) (d)

Fig. 8. The clustering results on the four UCI datasets by different clustering algorithms with a varying number of pairwise constraints

Table 1. Four UCI datasets used in the experiment. The features are first normalized to the range [-1, 1] for all the datasets.

Datasets	Wine	Ionosphere	Soybean	WDBC
# samples	178	351	47	569
# features	13	34	35	30
# clusters	3	2	4	2

Again, we can find that our E^{2}CP performs the best in most cases. Moreover, the other three constrained clustering approaches (i.e. AP, SL, and SSKK) are shown to have generally benefited from the pairwise constraints as compared to NCuts. This observation is different from that on the image datasets. As we have mentioned, this may be due to that, considering the complexity of the image datasets, a more exhaustive propagation (like our E^{2}CP) of the pairwise constraints is needed in order to fully utilize the inherent supervisory information provided by the constraints. Our experimental results also demonstrated that an exhaustive propagation of the pairwise constraints in the UCI data through our E^{2}CP leads to improved clustering performance over the other three constrained clustering approaches (i.e. AP, SL, and SSKK).

5 Conclusions

We have proposed an exhaustive and efficient constraint propagation approach to exploiting pairwise constraints for spectral clustering. The challenging constraint propagation problem for both the must-link and cannot-link constraints is decomposed into a set of independent label propagation subproblems, which can then be solved efficiently and in parallel through semi-supervised learning based on k-nearest neighbors graphs. The resulting exhaustive set of propagated pairwise constraints with associated confidence scores are further used to adjust the weight matrix for spectral clustering. It is worth noting that this paper first clearly shows how pairwise constraints are propagated independently and then accumulated into a conciliatory closed-form solution. Experimental results on image and UCI datasets demonstrate clearly that by exhaustively propagating the pairwise constraints throughout the entire dataset, our approach is able to fully utilize the additional supervisory or prior information inherent in the constrained data for spectral clustering and then achieve superior performance compared to the state-of-the-art techniques. For future work, our approach will also be used to improve the performance of other graph-based methods by exhaustively exploiting the pairwise constraints.

Acknowledgements

The work described in this paper was supported by a grant from the Research Council of Hong Kong SAR, China (Project No. CityU 114007) and a grant from City University of Hong Kong (Project No. 7008040).

References

1. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: ICML, pp. 577–584 (2001)
2. Klein, D., Kamvar, S., Manning, C.: From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In: ICML, pp. 307–314 (2002)

3. Basu, S., Bilenko, M., Mooney, R.: A probabilistic framework for semi-supervised clustering. In: KDD, pp. 59–68 (2004)
4. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: A kernel approach. In: ICML, pp. 457–464 (2005)
5. Lu, Z., Peng, Y.: A semi-supervised learning algorithm on Gaussian mixture with automatic model selection. Neural Processing Letters 27, 57–66 (2008)
6. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 14, pp. 849–856 (2002)
7. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17, 395–416 (2007)
8. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
9. Veksler, O.: Star shape prior for graph-cut image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 454–467. Springer, Heidelberg (2008)
10. Kamvar, S., Klein, D., Manning, C.: Spectral learning. In: IJCAI, pp. 561–566 (2003)
11. Lu, Z., Carreira-Perpinan, M.: Constrained spectral clustering through affinity propagation. In: CVPR (2008)
12. Li, Z., Liu, J., Tang, X.: Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In: ICML, pp. 576–583 (2008)
13. Yu, S., Shi, J.: Segmentation given partial grouping constraints. IEEE Trans. on Pattern Analysis and Machine Intelligence 26, 173–183 (2004)
14. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems, vol. 16, pp. 321–328 (2004)
15. Lu, Z., Ip, H.: Image categorization by learning with context and consistency. In: CVPR, pp. 2719–2726 (2009)
16. Lu, Z., Ip, H.: Combining context, consistency, and diversity cues for interactive image categorization. IEEE Transactions on Multimedia 12, 194–203 (2010)
17. Law, M., Topchy, A., Jain, A.: Clustering with soft and group constraints. In: Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, pp. 662–670 (2004)
18. Law, M., Topchy, A., Jain, A.: Model-based clustering with probabilistic constraints. In: Proceedings of SIAM Data Mining, pp. 641–645 (2005)
19. Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification 2, 193–218 (1985)
20. Lu, Z., Peng, Y., Xiao, J.: From comparing clusterings to combining clusterings. In: AAAI, pp. 665–670 (2008)
21. Lu, Z., Peng, Y., Ip, H.: Gaussian mixture learning via robust competitive agglomeration. Pattern Recognition Letters 31, 539–547 (2010)
22. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
23. Bosch, A., Zisserman, A., Muñoz, X.: Scene classification via pLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)

Object Recognition with Hierarchical Stel Models

Alessandro Perina[1,2], Nebojsa Jojic[2], Umberto Castellani[1], Marco Cristani[1,3], and Vittorio Murino[1,3]

[1] University of Verona
[2] Microsoft Research
[3] Italian Institute of Technology

Abstract. We propose a new generative model, and a new image similarity kernel based on a linked hierarchy of probabilistic segmentations. The model is used to efficiently segment multiple images into a consistent set of image regions. The segmentations are provided at several levels of granularity and links among them are automatically provided. Model training and inference in it is faster than most local feature extraction algorithms, and yet the provided image segmentation, and the segment matching among images provide a rich backdrop for image recognition, segmentation and registration tasks.

1 Introduction

It is well understood that image registration, segmentation and recognition are related tasks [17,23,18,3], and yet, the engineering paradigm suggests the decomposition of the general vision problem into components, first to be considered (and even applied) in isolation, and then, sometimes, combined as modules.

In some cases, the modular approach is highly successful. For example, algorithms for registration of multiple images of a static scene have recently matured to the point where they can be directly used in a variety of applications (e.g., photosynth.net). The registration algorithms typically do not attempt to solve the recognition or the segmentation problems, and are not readily applicable to registering images of different scenes or objects so that they can be used as modules in recognition algorithms. Still, the feature extraction stage, e.g. SIFT, in these technologies has found its way to object recognition research, but not as a tool for image registration. Under the assumption that registration of images of similar (but not identical) objects would be hard, the image features are compared as if they do not have a spatial configuration, i.e., as bags of visual words (BOW) [1] randomly scattered across the image.

The initial success of BOW models was extended when the researchers attempted to encode at least some spatial information in the models, even if the required spatial reasoning would be short of full image registration. Such models are often computationally expensive. For example, [2] forms vocabularies from pairs of nearby features called "doublets" or "bigamy". Besides taking cooccurrences into account this approach benefits from some geometric invariance,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 15–28, 2010.

but it is expensive even when feature pairs are considered, and the cost grows exponentially for higher order statistics. In [4] a codebook of local appearances is learned in way that allows reasoning about which local structures may appear on objects of a particular class. However, this process has to be supervised by human-specified object positions and segmentations. Generative part-based models like [6,23] are in principle learnable from unsegmented images, but are computationally expensive as they solve combinatorial search problems. Among the more computationally efficient approaches, the spatial pyramid method [7] stands out. The images are recursively subdivided into rectangular blocks, in a fixed, image-independent way, and the bag-of-words models are applied separately in these blocks. Image similarity is then defined based on the feature histogram intersections. This representation is combined with a kernel-based pyramid matching scheme [8], which efficiently computes approximate global geometric correspondence between sets of features in two images. Having defined an image kernel, or a similarity measure for two images, a variety of off-the-shelf learning algorithms can be used for classification (e.g., the nearest neighbor method, which simply labels the unlabeled test image with the label of the most similar labeled image). While the spatial pyramid indirectly registers images for computation of such a kernel, this registration is limited by the use of a fixed block-partition scheme for all images.

In this paper, we propose a related approach to defining image similarities, which can guide object recognition, but also segmentation and registration tasks. The similarities between two different images are broken down to different regions, but these regions are not rigidly defined by a pyramid kernel, nor do they require combinatorial matching between images as in [11]. Instead, they are computed using a novel hierarchical model based on the probabilistic index map/stel models [10,9,5,18], which consider the segmentation task as a joint segmentation of an image collection, rather than individual images, thus avoiding a costly combinatorial matching of segments across images. Our new hierarchical stel model (HSM) also contains multiple levels of segmentation granularity, linked across the hierarchy, and provides a rich backdrop for image segmentation, registration and recognition tasks, as any new image can be segmented in various class-specific ways under under this set of generative models. In particular, we propose a similarity kernel based on the entire stel hierarchy across all classes and granularity levels, and we demonstrate that the computation of this kernel for two test images implicitly matches not only image segments, but even the object parts at a much finer granularity than that evident in a segmentation under any class model. Not only that such use of HSM leads to high recognition rates, but it also provides surprisingly accurate unsupervised image segmentation, and unusually informative registration of entirely different images.

2 The Basic Probabilistic Index Map/Stel Model

The basic probabilistic index map, PIM [10], or as it is also called, structure element (*stel*) model, assumes that each pixel measurement x_i, with its 2-D coordinate i, has an associated discrete variable s_i, which takes a label from the interval

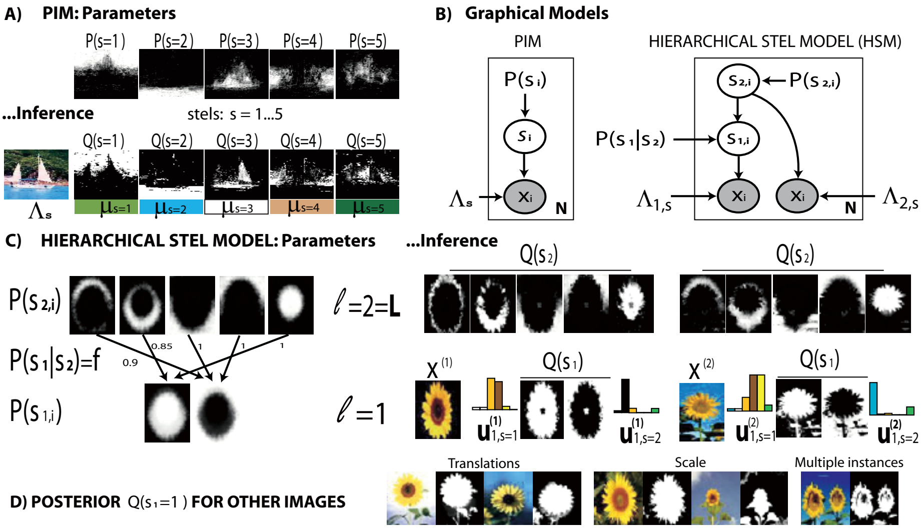


Fig. 1. PIM and Hierarchical stel model (HSM) illustration

$[1, S]$. Such a labeling splits the image into S stels so that s-th stel is a collection of pixel coordinates i, which may be scattered across the image, or grouped together into coherent blobs, and for which the index s_i is set to the desired stel label s, i.e., $\Omega(s) = \{i|s_i = s\}$. Fig. 1A shows some examples of stels: $\Omega(s = 2)$ represents the sea, $\Omega(s = 3)$ the schooner. The stel assignments are almost exclusively considered in a probabilistic fashion. In the simplest case, the distribution over possible assignments of image coordinates to stels is modeled by a set of location-specific distributions $P_i(s_i)$ that describe which image coordinates are more likely to belong to particular stels *a priori*. Such a probabilistic index maps ties the stel partitions in different images of the same type. The posterior stel distribution $Q(s_i)$ describes how this prior belief about class-specific image partition gets altered given the pixel measurements in a particular image (see Fig. 1A). The image evidence that the model detects is the image self-similarity within a stel: the pixels with the same stel label s are expected to follow a tight distribution over image measurements, defined by parameters Λ_s. Each distribution Λ_s can be modeled, for example, as a Gaussian $\Lambda_s = (\mu_s, \sigma_s)$ (in Fig.1 we only show the means μ_s) or in other more complex ways [18,9]. The collection $\{\Lambda_s\}$ of all stel parameters, organized by the stel index, is referred to as a palette. The palette for two different images of the same class can be completely different. Instead of local appearance similarity, the model insists on consistent segmentation through the stel prior. For example stel $\Omega(3)$ in all images of pedestrians may capture the lower part of the background and $\Omega(1)$ the torso of the pedestrian in the foreground (Fig. 3). Differences in local appearance of these parts are explained away as differences in the palettes associated with the images. Moreover, the stel prior is easily learned from a collection of images starting from a noninformative initialization, which allows for efficient segmentation of new images in a fashion consistent with the joint segmentation of the training images. Another view of this model is that captures correlated changes of pixels, as in [24], but in a much more computationally efficient way.

This basic model is easily enriched with transformation variables [18,9] which alleviate the requirement for rough pre-alignment of images. However, even the basic model has a remarkable ability to deal with somewhat misaligned images without the help of extra variables. For example, Fig. 1C-bottom illustrates the basic PIM model of the sunflower category, in which the images undergo significant transformations (scale, translations, multiple instances). Without help with accounting for these transformations explicitly, the prior $P(\{s_i\})$ is soft after learning, but strong enough to tie the segmentations together into consistent stels. Of course, this robustness to image transformation is limited. In case of very fine image segmentations with large number of stels, and/or very large image transformations, and/or a sparse training set, the part correspondence may be highly unreliable. Adding transformation variables could help in such cases, but in this paper we advocate an even more efficient approach that follows a traditional computer vision concept: coarse-to-fine hierarchies.

3 Hierarchical Stel Model (HSM)

Modeling transformation variables is inherently expensive in any model. The cost of dealing with image translation is of the order $N \log N$, where N is the number of pixels, but if we also need to take care of scale, rotation, or even affine transformations, the expense may accumulate quickly. In this paper, our goal is to extend the natural ability of stel models to capture all but the largest transformations. If for instance, the model is not sensitive to the transformations present in the fairly well-aligned Caltech database, then the extra transformation variables only need to model coarse translation in large images (relative to the object size), and capture scale at several coarse levels.

To achieve such an increased invariance to image transformation, we consider stel models at multiple levels of granularity so that the more refined models are linked to the coarser models. This modification confers two advantages to the stel models:

- If the alignment at some level of granularity is failing, the coarser levels may still be useful.
- The higher quality of the alignment of stels at a coarse granularity will guide the alignment at a finer granularity, making these more useful.

Hierarchical stel model captures a hierarchy of stel partitions at L different granularity levels indexed by ℓ: $\Omega^{\ell}(s) = \{i|s_{\ell,i} = s\}$. The index label s can be chosen from sets of different cardinality for stels at different levels of hierarchy. For example, in Fig. 1C we show two levels of hierarchical stel model with two stels in level $\ell = 1$ and five in level $\ell = 2$. The stel partitions are linked hierarchically by distributions $P(s_{\ell,i} = a|s_{\ell+1,i} = b) = f^{\ell}_{a,b}$ which are *not* spatially varying. In Fig. 1C this linking conditional distributions are defined by a 5×2 table of conditional probabilities $f^{1}_{a,b}$, but only a few strongest weights are illustrated by arrows. The image $\{x_i\}$ is linked to each of these stel assignments directly, as if it was generated L times[1] (Fig. 1B).

Given the prior $P^{\ell+1}(\{s_i\})$ for level $\ell+1$ in the same form as in the basic site-specific PIM/stel model of the previous section, the prior for the level below satisfies:

$$P_i^{\ell}(s_{\ell,i}=a)=\sum_b P_i^{\ell+1}(s_{\ell+1,i}=b)\cdot f_{a,b}^{\ell}. \tag{1}$$

In this way, each successive level provides a coarser set of stels, created by (probabilistic) grouping of stels from the previous level according to $f_{a,b}^{\ell}$; only at the finest granularity the stel prior is location-specific, as in the previous section,

$$P(\{s_{L,i}\}_{i=1}^{N})=\prod_i P_i(s_{L,i}). \tag{2}$$

As before, the conditional links between the image observation and the stel assignment at $P(x_i|s_{\ell,i}=s)$ depend only on the s-th palette entry at the hierarchy level ℓ, and not on the pixel coordinate, thus allowing the palette to affect the appearance of all the stel's pixels in concert. For added flexibility, the palette entries capture a mixture of colors. Image colors in the dataset are clustered around 32 color centers, and the real-valued pixel intensities are replaced by discrete indices to these centers in all our experiments. Each palette entry $\Lambda_{\ell,s}$ is thus a histogram consisting of 32 probabilities $\{u_{\ell,s}(k)\}$, and

$$P(x_i=k|s_{\ell,i}=s)=u_{\ell,s}(k). \tag{3}$$

The joint probability over all variables in the model is

$$P=\prod_i P(s_{L,i})\prod_{\ell=0}^{L-1} f_{s_{\ell,i},s_{\ell+1,i}}^{\ell}\prod_{\ell=0}^{L} p(x_i|s_{\ell,i}) \tag{4}$$

where level $\ell=0$ trivially reduces to a bag of words representation as the stel variables across the image are constant $s_{0,i}=1$. Following the same strategy as [10] we can easily write the free energy $F=\sum Q\log\frac{Q}{P}$ for this graphical model assuming a factorized posterior $Q=\prod_{\ell,i}Q(s_{\ell,i})$, take appropriate derivatives, and derive the following inference rules for minimizing the free energy for a single image given the prior over stel hierarchy:

$$Q(s_{\ell,i}=s)\propto P(s_{\ell,i}=s)\cdot u_{\ell,s}(x_i)\quad u_{\ell,s}(k)\propto\sum_i Q(s_{\ell,i}=s)\cdot[x_i=k], \tag{5}$$

where [] is an indicator function. The above updates are image-specific; each image has in fact its own palette of histograms which allows images with very different colors to be segmented following the same stel prior (Fig. 1C).

[1] The motivation for multiple generation of x_i from multiple levels of hierarchy comes from the observation that modeling multiple paths from hidden variables to the data, or, for that matter, among hidden variables in the higher levels, alleviates local minima problems in learning [19].

Given a collection of images indexed by t, and the posterior distributions $Q(s_\ell^t)$ computed as above, the hierarchical stel distribution is updated as

$$f_{a,b}^{\ell} \propto \sum_{t,i} Q(s_{\ell+1,i}^t = b) \cdot Q(s_{\ell,i}^t = a) \qquad P(s_{L,i} = s) \propto \sum_t Q(s_{L,i}^t = s). \tag{6}$$

These updates are iterated and the model is learned in an unsupervised way form a collection of images. As the result, all images are consistently segmented into stels at multiple levels of hierarchy. As the palettes are image-specific in the model, the images can have completely different colors and still be consistently segmented. The hierarchical representation of stels reduces the errors in segmentation, and provides a rich information about part correspondence for image comparison, and, therefore, recognition.

4 Hierarchical Stel Kernel (HSK)

The HSM can be trained for many different image classes indexed by c. A pair of images (whether they are in one of the training sets for the stel models or not) can be segmented into stels under any of the resulting models $P_c(\{s_{\ell,i}\})$ by iterating the two equations (5). The pair of resulting posterior distributions $Q_c(s_{\ell,i}^A), Q_c(s_{\ell,i}^B)$ for each combination of class c and granularity level ℓ provides a coarse correspondence for regions in the two images (Fig. 2).

This rich information can be used in numerous ways, but we limit our analysis and experiments here to one of the simplest approaches, inspired by the spatial pyramid match kernel [7], which propose course-to-fine spatial feature matching schema based on comparing histograms of image features in different parts of the image and weighting and accumulating evidence of feature sharing. As in [7], we compute image features in images and represent them using the same codebook of 300 visual words. But, instead of partitioning each image image using the same set of rectangular blocks of different sizes, we use the image-specific segmentations induced by HSM models. Then similarity in image features in two different images is considered important if these features tend to be within the same posterior stel under many models.

Specifically, the feature indices $k \in [1, 300]$ are assigned to locations on a grid that covers every fifth pixel along both image dimensions. In a given image, within the s-th stel under the model of class c, at a hierarchy level ℓ an unnormalized histogram of image features $h_{c,\ell,s}(k)$ is computed as

$$h_{c,\ell,s}(k) = \sum_i Q_c(s_{\ell,i}) \cdot n_{i,k} \tag{7}$$

where $n_{i,k}$ is equal to 1 if a feature of index k is present at location i, 0 otherwise. Given two images A and B, their histogram similarities within the corresponding stels are defined by the histogram intersection kernel [8] defined as

$$K(A,B) = \min_k(h_{c,\ell,s}^A(k), h_{c,\ell,s}^B(k)), \tag{8}$$

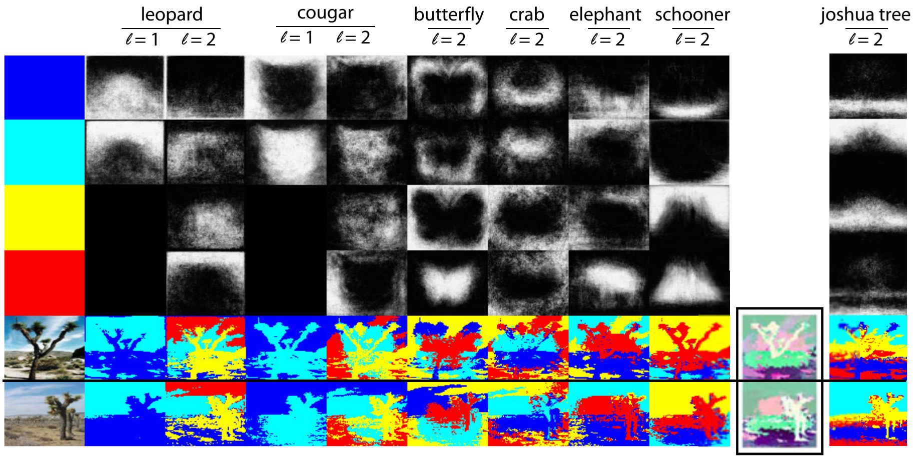


Fig. 2. Segmentations of two images from the Joshua tree category under various stel models trained on Caltech 101 images. The prior stel distributions are illustrated on top. The stels are assigned different colors (blue, light blue, yellow and red), to illustrate the mode of each posterior stel assignment, which is based both on the prior and on the image evidence. Although none of the individual segmentations under the leopard, cougar, butterfly, crab, elephant, and schooner models fits these models very well, the two images are for the most part consistently segmented under these models: If the different stel assignments a pixel gets under these different models are considered a discrete multi-dimensional label, and if these multi-dimensional labels of all pixels are projected through a random matrix onto 3D colors, so that the similar consistent labels across models and levels of hierarchy result in a similar color, then the two joshua tree images end up colored as shown in the rectangular box. This illustrates that the tree bark has consistent stel assignment in two images more often than not, and similar correspondence among other parts of the two scenes are visible. In contrast, a single segmentation, even under the model trained on Joshua tree images (the last column), does not provide a refined part correspondence.

because this provides computational advantages. To compute a single measure of similarity for two images under all stels of level ℓ, we sum all the similarities, weighting more the matches obtained in finer segments:

$$K_c^{HSK}(A,B) = \sum_{l=0}^{L} \frac{1}{2^{L-\ell}} \cdot \sum_s \min_k (h^A_{c,\ell,s}(k), h^B_{c,\ell,s}(k)), \tag{9}$$

In multi class classification tasks, we define the hierarchical stel kernel (HSK) as the sum of the kernels for individual classes $K^{HSK} = \sum_c K_c^{HSK}$. There are two reasons for this operation. First, when image similarities are computed for classification tasks, one or both images may not be labeled as belonging to a particular class, and so considering all classes simultaneously is needed. Second, even if one of the images belongs to a known class (an exemplar used in classification, for instance) and the other's class is to be predicted, multiple segmentations of the images under different class models provides useful additional alignment information (Fig. 2). When insufficient data is used for training stel models (e.g., 15 training images for Caltech101), the segmentation under any given class may be noisy, and so pulling multiple segmentations may help. Natural images share similar structure: Consider for example portraits of dogs and humans, or structure of different classes of natural scenes, where the background is broken into horizontal stripes in images of schooners and cars alike. Thus, using many stel tessellations under many classes reinforces proper alignment of image parts.

Furthermore, as Fig. 5B illustrates, the alignment becomes finer than under any single model, even than the finest level of stel hierarchy under the model for the *correct* class. To illustrate this, we note that because the posterior $Q(s)$ tends to be peaky, i.e. close to 0 or 1 for most pixels, for any class we have

$$\begin{aligned} K_c^{HSK}(A,B) &\approx \sum_{l=0}^{L} \frac{1}{2^{L-\ell}} \cdot \sum_{i,j} \min_k(n^A_{k,i}, n^B_{k,j}) \times \big(\sum_s \min_{A,B}(Q(s^A_{\ell,i}=s), Q(s^B_{\ell,j}=s)\big) \\ &= \sum_{i,j} F_{i,j} \times M_{i,j} \qquad (10) \end{aligned}$$

where $M_{i,j} = \sum_{\ell=0}^{L} \frac{1}{2^{L-\ell}} \big(\sum_s \min_{A,B}(Q(s^A_{\ell,i} = s), Q(s^B_{\ell,j} = s)\big)$ represents the level of expected similarity between the i-th pixel in image A and j-th pixel in image B based simply on how often the stel labels for these two pixels are shared across the hierarchy, and $F_{i,j} = \min_k(n^A_{k,i}, n^B_{k,j})$ represents feature similarities (i.e., matches) between the coordinate i in one image and coordinate j in the other, independently of any segmentation. Finally we can write

$$K^{HSK} = \sum_{i,j} F_{i,j} \times \sum_c M^c_{i,j}. \qquad (11)$$

Here we have that $F_{i,j} > 0$ if in locations i and j the same feature index is present. This feature match is more rewarded through weight $\sum_c M^c_{i,j}$ if i and j share the same stels across different models and granularity levels. Figure 5 illustrates these two components, $F_{i,j}$ and $\sum_c M^c_{i,j}$, of the similarity kernel on the pixel level. First, in Fig. 5A we show how combining three arbitrary classes creates enough context not only to find the corresponding segment for pixel i in the first image, but to actually refine this matching across pixels j in the second. For the selected i, marked by a square, $\sum_c M^c_{i,j}$ is represented as an image over coordinates j in the second image. In the second image, as well as in match maps $\sum_c M^c_{i,j}$, the cross represents the pixel $j = i$ so that the misalignment of the two faces is evident. While the inference under the face class may be sufficient to roughly match large regions among the images, the stel segmentations based on three classes' segmentations narrow down the correspondence of the marked pixel (right eye) to the eye regions of the face in the second image and a spurious match in the background which happened to have a similar color to the facial region. For easier visualization we illustrated only three select stels from the three classes. In Fig. 5B for this example, and several more, we show what happens when all stels and all classes are used as in the equations above. For two facial images, the supplemental video shows correspondence

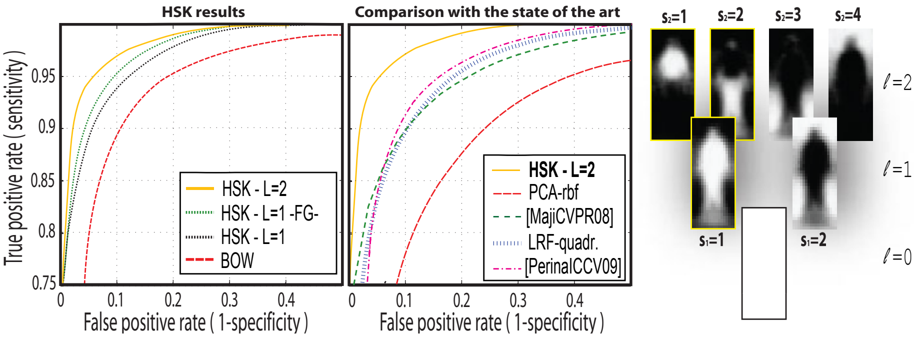


Fig. 3. Pedestrian classification. Left: ROC plots comparing HSM/HSK and other approaches. Right: the learned HSM parameters.

of various pixels in the same manner (The pixel in the first image is marked by a cursor, and the mapping in the second image is shown as a heat map).

Finally in Fig. 5C, we show jointly the mapping of three pixels i_1, i_2, i_3 in the first image by placing the appropriate match maps M in the R, G, and B channels of the image. As the result, when the entire stel hierarchy under all classes is used to evaluate $\sum M$, the regions around the eyes, and especially around the right eye in the second image are colored red, while the regions in the lower part of the face, especially lips, are colored green, and the background elements are colored blue, indicating that the entire stel model hierarchy can localize the face parts beyond the granularity of any single model and any single level of hierarchy. For comparison, M obtained for the face class only and butterfly class only are shown. To illustrate in the same manner the spatial pyramid kernel [7], we compute similar decomposition into the expected matching of pixels based on block image segmentation, and the feature matching of pixels. The complete kernel under both HSM and the spatial pyramid is the sum over all pixels of the product $M_{i,j} \cdot F_{i,j}$ and so these products are also illustrated in the figure.

Inference and learning complexity in stel models is linear in the number of image coordinates, stels and classes. The total computation time is considerably faster than SIFT feature computation. Furthermore, the quality of image matching does not decay much if we use only 30 out of 101 classes.

5 Experiments

We evaluated our approach on Caltech28, Calteh101 and Daimler pedestrian datasets. We compared with the classification results provided by the datasets' creators and with the other feature organization paradigms, namely Bag of words (BW), Stel organization (SO) and Spatial Pyramids (SPK), as well as other state-of-the art methods. We considered both classification and unsupervised segmentation tasks. We used support vector machines as discriminative classifiers, feeding the kernels as input.

5.1 Pedestrian Classification: Daimler Dataset

We evaluated our method on pedestrian classification using the procedure of [12]. We trained a hierarchical stel model with $S_1 = 2$ and $S_2 = 4$ on the training set for each class (See Fig. 3 for an illustration). Having trained HSM on the training data, stel inference can be performed on test images, so that pairwise similarities (the kernel matrix) can be computed for all pairs of images (training and test). For the feature code book, we used the dictionary of Haar wavelets [13]. Given input images of size 18 x 36 and their posterior distributions $Q(s_1^t)$ and $Q(s_2^t)$, we compute w_l^t convolving the image x^t with wavelets of scales 4 x 4 (l=1) and 8 x 8 (l=2). We only encoded the magnitude in the feature vectors. As described above, image features and stel segmentations are used to compute the kernel matrix and this matrix is fed to a standard SVM classification algorithm. The ROC plots are shown in Fig. 3. As expected, results improve as we go from L = 0 (AUC, Area under the curve, 0.954) to a multi-level setup (L > 0). We repeated the classification only keeping into account the foreground wavelet coefficients. When L=1 the accuracy is significantly improved by considering only the foreground, but for L=2 it does not, as the hierarchical stel kernel already reaches impressive performance without emphasizing foreground in classification. Though matching at the highest pyramid level seems to account for most of the improvement (AUC 0.9751), using all the levels together confers a statistically significant benefit (AUC 0.9854). The ROC plot on the right of figure 3 compares HSK with several recent approaches including [12] which reviews standard pedestrian classification algorithm and features, [15] which uses a hybrid generative-discriminative approach based on PIM [10], and [14] which employs spatial pyramids kernel on a multi-level version of the HOG descriptor [16].

5.2 Unsupervised Segmentation and Supervised Recognition of Caltech 28 Images

Caltech 28 [17] is composed of 28 classes of objects among the subset of Caltech 101 categories that contain more than 60 images. The chosen categories contain objects with thin regions (e.g. flamingo, lotus), peripheral structures (e.g. cup), objects that are not centered (e.g. leopards, dalmatians, Joshua trees). None of the chosen classes contains background artifacts that make them easily identifiable. For each class, we randomly selected 30 images for training and 30 images for testing. To serve as discrete features to match, we extracted SIFT features from 15x15 pixel windows computed over a grid with spacing of 5 pixels. These features were mapped to W=300 codewords as discussed in Section 4. We trained a hierarchical model for each class using $S_1 = 3$ and $S_2 = 5$ and then

Table 1. Classification accuracies on Caltech 28

HSK L=1	HSK L=1	HSK L=2	[9]	SPK [7]	BW	[17]
$S_1 = 3$	$S_1 = 5$	$S_1 = 3, \ S_2 = 5$	-	L=2	-	-
73,15%	74,57%	**78,10%**	65,12%	65,43%	56,01%	69%

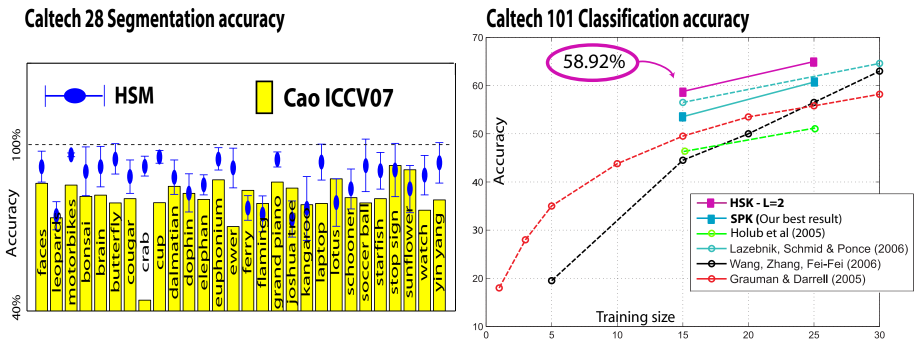


Fig. 4. Classification results for the Caltech experiments. On the left we report the segmentation accuracy for each class of Caltech 28 obtained by [17] (yellow bars) and by HSM (blue dots with confidence level). On the right, we compare recognition rates on Caltech 101 images with related spatial-reasoning methods using similar local features.

performed inference on the test images. We calculated the kernel between all pairs of images as discussed in Section 4 and the used a standard SVM that uses the class labels and kernels to determine the missing class labels of images in the test set. We compared the results of several set ups of HSK and with: *i*) the bag of words classifier BW, *ii*) the spatial pyramid kernel (SPK, [7]), and *iii*) a classifier based on the single level stel partition (SO, S=5, [9]). All the methods are compared using the same core-kernel (histogram intersection) and the same feature dictionary. First, we compared these related methods repeating the classification 10 times with a randomly chosen training-testing partition. Then we performed t-tests and found:

$$BW <<^{1\cdot 10^{\;3}} SPK <<^{3\cdot 10^{\;3}} HSK >>^{5\cdot 10^{\;4}} SO >>^{4\cdot 10^{\;3}} BW^{2} \qquad (12)$$

Where $>>^p$ stands for greater with statistical significance with p-value equal to p. HSK's advantage here is due to the segmentations provided by HSM, which explain away a lot of object transformations (see Fig.1C, bottom) and capture meaningful object partitions. Mean classification accuracies are summarized in table 1. As a further test on Caltech 28 we tackled image segmentation, simply using the posterior stel segmentation induced by the coarsest level of HSM ($S_1 = 2$). Each class of images is fit independently as described in Section 3. After training, the posterior stel distributions are used as image segmentations. We compared our results with [17], which provides the manual labeling of pixels. In figure 4 we compare the segmentation accuracy over different classes. The overall test accuracy of our unsupervised method is 79,8%, outperforming the supervised method of [17] with test accuracy of 69%.

[2] SO and SPK have been found statistically equal.

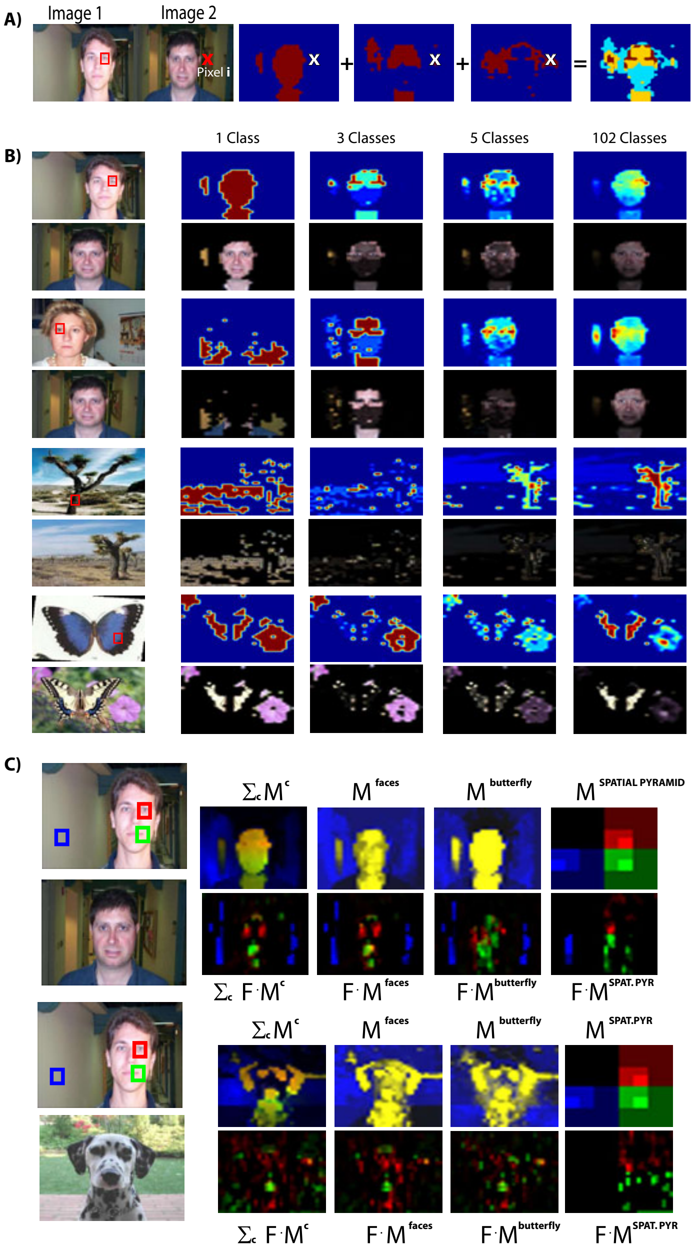


Fig. 5. Image correspondences implicitly captured by the hierarchical stel kernel. In A and B, the pairs of images are shown with the pixel of interest in the first image labeled by a square. In B, for each pair, the stel-based match matrix M, which is only based on color stel models, is shown as averaged under 1,3,5, and 102 classes randomly selected from Caltech 101. Below each M matrix we show it multiplied with the target image. C illustrates the correspondence of multiple points for two image pairs.

5.3 Recognition Rates on Caltech 101

Our final set of experiment is on the Caltech 101 dataset. For the sake of comparison, our experimental setup is similar to [7]. Namely, we randomly select 30 images from each category: 15 of them are used for training and the rest are used for testing. We compare our method to only those recognition approaches that do not combine several other modalities. Results are reported in figure 4 The successfully recognized classes include the ones with rotation artifacts, and the natural scenes (like joshua tree and okapi), where segmentation is difficult. The least successful classes are animals, similarly to [7]. This is likely not due to problems of segmentation, but discretized feature representation [20]. Since our goal is mainly to compare our representation with SPK we report the results we have obtained using the SPK authors's implementation of the feature extraction and quantization. Note that due to a random selection of images, we did not recreate the exact classification result of SPK, but our HSK similarity measure outperforms both our implementation of the SPK and the best published SPK result.

6 Conclusions

We propose a new generative model, and a new image similarity kernel based on a linked hierarchy of stel segmentation. The goal of our experiments was primarily to demonstrate the spatial reasoning that can be achieved with our method, and which goes beyond block comparisons, and even beyond segment matching and closer to registration of very different images. Therefore we compared our method using the same discretized features as in the literature describing efficient spatial reasoning approaches. However, we expect that the better local feature modeling may improve classification performance, as for example, [20] proposes. Still, even with current discretized features, the hierarchical stel models can be used efficiently and with high accuracy in segmentation and classification tasks. We expect that our image representation will find its applications in multikernel approaches but may also find other applications due to its ability to combine image recognition, segmentation, and registration. For example [21,22] are based on SPK and could be easily used with our method.

References

1. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR 2005 (2005)
2. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in images. In: ICCV 2005 (2005)
3. Russell, B.C., Efros, A., Sivic, S., Freeman, W.T., Zisserman, A.: Segmenting Scenes by Matching Image Composites. In: NIPS 2009 (2009)
4. Leibe, B., et al.: An implicit shape model for combined object categorization and segmentation. In: Ponce, J., Hebert, M., Schmid, C., Zisserman, A. (eds.) Toward Category-Level Object Recognition. LNCS, vol. 4170, pp. 508–524. Springer, Heidelberg (2006)

5. Ferrari, V., Zissermann, A.: Learning Visual Attributes. In: NIPS 2007 (2007)
6. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 18–32. Springer, Heidelberg (2000)
7. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR 2006 (2006)
8. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features. In: ICCV 2005 (2005)
9. Jojic, N., Perina, A., Cristani, M., Murino, V., Frey, B.J.: Stel component analysis: Modeling spatial correlations in image class structure. In: CVPR 2009 (2009)
10. Jojic, N., Caspi, Y.: Capturing image structure with probabilistic index maps. In: CVPR 2004 (2004)
11. Russell, B., et al.: Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In: CVPR 2006 (2006)
12. Munder, S., Gavrila, D.: An experimental study on pedestrian classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1863–1868 (2006)
13. Papageorgiou, C., Poggio, T.: A trainable system for object detection. Int. J. Comput. Vision 38, 15–33 (2000)
14. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR 2008 (2008)
15. Perina, A., et al.: A Hybrid Generative/discriminative Classification Framework Based on Free-energy Terms. In: ICCV 2009 (2009)
16. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: ICCV 2005 (2005)
17. Cao, L., Fei-Fei, L.: Spatially Coherent Latent Topic Model for Concurrent Segmentation and Classification of Objects and Scenes. In: ICCV 2007 (2007)
18. Winn, J., Jojic, N.: LOCUS: Learning Object Classes with Unsupervised Segmentation. In: ICCV 2005 (2005)
19. Jojic, N., Winn, J., Zitnick, L.: Escaping local minima through hierarchical model selection: Automatic object discovery, segmentation, and tracking in video. In: CVPR 2006 (2006)
20. Boiman, O., Shechtman, E.: In Defense of Nearest-Neighbor Based Image Classification. In: CVPR 2008 (2008)
21. Yang, J., Yuz, K., Gongz, Y., Huang, T.: Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification. In: CVPR 2009 (2009)
22. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Active Learning with Gaussian Processes for Object Categorization. In: ICCV 2007 (2007)
23. Sun, M., Su, H., Savarese, S., Fei-Fei, L.: A multi-view probabilistic model for 3d object classes. In: CVPR 2009 (2009)
24. Stauffer, C., Miller, E., Tieu, K.: Transform-invariant image decomposition with similarity templates. In: NIPS 2002 (2002)

MIForests: Multiple-Instance Learning with Randomized Trees[⋆]

Christian Leistner, Amir Saffari, and Horst Bischof

Institute for Computer Graphics and Vision, Graz University of Technology,
Inffeldgasse 16, 8010 Graz, Austria
{leistner,saffari,bischof}@icg.tugraz.at
http://www.icg.tugraz.at

Abstract. Multiple-instance learning (MIL) allows for training classifiers from ambiguously labeled data. In computer vision, this learning paradigm has been recently used in many applications such as object classification, detection and tracking. This paper presents a novel multiple-instance learning algorithm for randomized trees called *MIForests*. Randomized trees are fast, inherently parallel and multi-class and are thus increasingly popular in computer vision. MIForest combine the advantages of these classifiers with the flexibility of multiple instance learning. In order to leverage the randomized trees for MIL, we define the hidden class labels inside target bags as random variables. These random variables are optimized by training random forests and using a fast iterative homotopy method for solving the non-convex optimization problem. Additionally, most previously proposed MIL approaches operate in batch or off-line mode and thus assume access to the entire training set. This limits their applicability in scenarios where the data arrives sequentially and in dynamic environments. We show that MIForests are not limited to off-line problems and present an on-line extension of our approach. In the experiments, we evaluate MIForests on standard visual MIL benchmark datasets where we achieve state-of-the-art results while being faster than previous approaches and being able to inherently solve multi-class problems. The on-line version of MIForests is evaluated on visual object tracking where we outperform the state-of-the-art method based on boosting.

1 Introduction

In recent years, visual object classification and detection has made significant progress. Besides novel methods for image representations, one important factor was the development and application of advanced machine learning methods. Traditional supervised learning algorithms require labeled training data where each instance (*i.e.*, data sample or feature vector) has a given label. In practice, the labels are usually provided by a human labeler. However, especially for positive classes it is often hard to label the samples so that they can be best

[⋆] This work has been supported by the Austrian FFG project MobiTrick (825840) and Outlier (820923) under the FIT-IT program.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 29–42, 2010.

exploited by the learning algorithm. For example, in case of object detection bounding boxes are usually cropped around the target object and provided as positive training samples. The decision where exactly to crop the object and at which size is up to the human labeler and it is often not clear if those patches are best suited for the learner. Additionally, it would also ease the labeling effort if the *exact* object location had not to be marked. By contrast, it would be desired to provide the learner only a rough position of the target object and leave it on its own how to incorporate the information in order to deliver best classification results. For standard supervised learning techniques it is hard to resolve such ambiguously labeled data. In contrast, multiple-instance learning (MIL) [1,2] naturally can perform this task. In particular, in multiple-instance learning, training samples are provided in form of bags, where each bag consists of several instances. Labels are only provided for the bags and not for the instances. The labels of instances in positive bags are unknown whereas all instances in negative bags can be considered as being negative. For positive bags, the only constraint is that at least one of the instances is positive. Recently, multiple instance learning has enjoyed increasing popularity, especially in computer vision, because in practice data is often provided in a similar manner. Applying MIL in the above example, the rough object position would correspond to a bag and patches inside the bag to instances. During training, MIL would find those patches that lead to best classification results and leave out the others.

While multiple-instance learning has been used in many applications such as text-categorization [3], drug activity recognition [2] or computer security problems [4], especially computer vision is one of the most important domains where multiple instance-learning algorithms have been recently applied. For instance, many authors applied MIL to image retrieval [5,6] or image categorization tasks [7]. Another computer vision application where multiple-instance learning can be used is to tackle the alignment problem when training appearance-based detectors based on boosting [8], speed-up classifier cascades [9] or even action recognition [10] and semantic segmentation [11]. In case of object tracking, it is mostly hard to decide which patches to use for updating the adaptive appearance model. If the tracker location is not precise, errors may accumulate which finally leads to drifting. Recently, Babenko *et al.* [12] demonstrated that using MIL for tracking leads to much more stable results. For most of these vision tasks SVM variants or boosting have been used.

In this paper, we present a novel multiple-instance learning algorithm based on random forests (RF) [13][1]. The motivation for developing such an algorithm has several reasons: RFs have demonstrated to be better or at least comparable to other state-of-the-art methods in both classification [13] and clustering [14]. Caruana *et al.* [15] showed that RFs outperform most state-of-the-art learners on high dimensional data problems. Especially, the speed in both training and evaluation is one of their main appealing properties. Additionally, RFs can easily be parallelized, which makes them interesting for multi-core and GPU

[1] Note that we consider "random forests" and "randomized trees" to be the same and use the term interchangeably throughout the paper.

implementations [16]. RFs are inherently multi-class, therefore it is not necessary to build several binary classifiers for solving multi-class problems. Finally, compared to boosting and other ensemble methods, RFs are more robust against label noise [13]. These advantages of random forests have also led to increased interest in the computer vision domain. For instance, recently Gall and Lempinsky [17] presented an efficient object detection framework based on random forests. Shotton *et al.* [18] presented a real-time algorithm for semantic segmentation based on randomized trees. Bosch and Zisserman used RFs for object categorization [19]. Randomized trees have also successfully applied to visual tracking, either in batch mode using keypoints [20] or on-line using tracking-by-detection [21].

The main contribution of this work is an algorithm that extends random forests to multiple-instance learning. We thus call the method *MIForests*. MIForests bring the advantages of random forests to multiple-instance learning, where usually different methods have been applied. In turn, extending random forests in order to allow for multiple-instance learning allows vision tasks where RFs are typically applied to benefit from the flexibility of MIL. MIForests are very similar to conventional random forests. However, since the training data is provided in form of bags, during learning the real class labels of instances inside bags are unknown. In order to find the hidden class labels, we consider them as random variables defined over a space of probability distributions. We disambiguate the instance labels by iteratively searching for distributions that minimize the overall learning objective. Since this is a non-convex optimization problem, we adopt an approach based on deterministic annealing, which provides a fast solution and thus preserves the speed of random forests during training. The evaluation speed of MIForests is identical to standard random forests.

Although there have been proposed numerous approaches to the MIL problem, most of them operate in off-line or batch mode. Off-line methods assume having access to the entire training data which eases optimization and typically yields good classifiers. In practice, however, learners often have limited access to the problem domain due to dynamic environments or streaming data sources. In computer vision, this is *e.g.* the case in robot navigation or object tracking. For such problems off-line learning does not work anymore and on-line methods have to be applied. In this paper, we take this into account and show how MIForests can be extended to on-line learning.

In the experimental section, we compare MIForests with other popular MIL algorithms both on benchmark data sets and on multi-class image classification problems, where we show that MIForests can achieve state-of-the-art results without splitting multi-class problems into several binary classifiers. We evaluate the on-line extension of MIForests on object tracking and compare it to the state-of-the-art methods.

In Section 2, we present a brief overview on previous multiple-instance learning methods and RFs. In Section 3, we derive our new multiple-instance learning algorithm for random forests and present an on-line extension. Experimental results on standard visual MIL datasets, comparisons to other MIL approaches and tracking results of our approach are presented in Section 4. Finally, in Section 5, we give some conclusions and ideas for future work.

2 Related Work

In traditional supervised learning training data is provided in form of $\{(\mathbf{x}_1, y_1) \ldots (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i$ is an instance and, in the binary case, $y_i \in \{-1, +1\}$ the corresponding label. In multiple instance learning training samples are given in bags $B_i, i = 1, \ldots, n$, where each bag may consist of an arbitrary number of instances $B_i = \{x_i^1, x_i^2, \ldots, x_i^{n_i}\}$. Negative bags B_i^- consist of only negative instances. Ambiguity is introduced into learning by the constraint that for positive bags B_i^+, it is only guaranteed that there exist at least one positive instance (also called *witness* of the bag). There is no information about other instances in the bag. In fact, they might not even belong to the negative class. The task is to learn either a bag classifier $f : B \rightarrow \{-1, 1\}$ or an instance classifier $f : \mathbb{R}^d \rightarrow \{-1, 1\}$. However, bag classification can be obtained automatically from instance classification, *e.g.*, by using the *max* operator $p_i = \max_j \{p_{ij}\}$ over posterior probability estimates p_{ij} for the j-th instance of the i-th bag.

There exists a vast amount of literature and many different approaches on how to solve the MIL problem. Here, we briefly review some of the most popular ones. The most naïve approach is to simply ignore the MIL setting and train a supervised classifier on all instances with the bag label. Blum and Kalai [22], for instance, showed that one can achieve reasonable results when training an instance classifier that is robust to class label noise. As we will show later in the experimental part, RFs are also promising candidates for such a naïve approach. Many MIL methods work by adapting supervised learners to the MIL constraints, mostly using SVM-type learners. For example, Andrews *et al.* [3] proposed two different types of SVM-MIL approaches mi-SVM and MI-SVM. They differ basically on their assumptions, *i.e.*, the first method tries to identify the labels of all instances in a bag while the latter one finds only the witness and ignores all others. Another SVM-based approach MICA [23] tries to find the witness using linear programming. There also exist some boosting-based methods, *e.g.*, [8]. Wang and Zucker [24] trained a nearest neighbor algorithm using Hausdorff distance. Other popular approaches are based on the diverse-density assumption, for example [25,26], which more directly tries to address the MIL problem via finding a more appropriate feature representation for bags. In MILES, Chen *et al.* [7,27] trained a supervised SVM on data mapped into a new feature space based on bag similarities. There exist also approaches for training decision trees in a MIL fashion, *e.g.*, [28].

2.1 Random Forests

Random Forests (RFs) were originally proposed by Amit *et al.* [29], extended by Breiman [13] and consist of ensembles of M independent decision trees $f_m(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{Y} = \{1, \ldots, K\}$. For a forest $\mathcal{F} = \{f_1, \cdots, f_M\}$ the predictive confidence can be defined as $F_k(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} p_m(k|\mathbf{x})$, where $p_m(k|\mathbf{x})$ is the estimated density of class labels of the leaf of the m-th tree, where sample $\mathbf{x}$ resides. A decision is made by simply taking the maximum over all individual probabilities of the trees for a class k with $C(\mathbf{x}) = \arg\max_{k\in\mathcal{Y}} F_k(\mathbf{x})$. [13] showed that the generalization error of random forests is upper bounded by $GE \leq \bar{\rho}\frac{1-s^2}{s^2}$, where $\bar{\rho}$ is the mean correlation between pairs of trees in the forest and s is the strength of the ensemble (*i.e.*, the expected value of the margin over the entire distribution). In order to decrease the correlation of the trees, each tree is provided with a slightly different subset of training data by subsampling with replacement from the entire training set, *a.k.a* bagging. Trees are trained recursively, where each split node randomly selects binary tests from the feature vector and selects the best according to an impurity measurement such as the entropy $H(I) = -\sum_{i=1}^{K} p_i^j \log(p_i^j)$, where p_i^j is the label density of class i in node j. The recursive training continues until a maximum depth is reached or no further information gain is possible.

3 Multiple-Instance Random Forests

In the following, we introduce a novel multiple instance learning algorithm using randomized trees called *MIForests*. MIForests deliver multi-class instance classifiers in form of $F(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{Y} = \{1, \ldots, K\}$. Hence, during learning for each bag there is guaranteed that it has at least one instance from the target class but it may also consist of instances of some or all other classes $\{1, \ldots, K\}$. This makes MIForests different to most previous MIL algorithms that only yield binary classifiers and require to handle a multi-class problem by a sequence of binary ones. One obvious way to design RFs capable of solving MIL tasks is to adopt MIL versions for single decision trees [28]. However, strategies developed for common decision trees are hard to apply for RFs due to the random split nature of their trees. For example, improper regularization of trees of a RF on the node level can decrease the diversity $\bar{\rho}$ among trees and thus increase the overall generalization error [13]. Thus, in order to perform multiple instance learning with random forests one has to find an optimization strategy that preserves the diversity among the trees.

We formulate multiple instance learning as an optimization procedure where the labels of the instances become the optimization variables. Therefore, the algorithm tries to uncover the true labels of the instances in an iterative manner. Given such labels, one can train a supervised classifier which then can be used to classify both instances and bags. Let $B_i, i = 1, \ldots, n$ denote the i-th bag in the training set with label y_i. Each bag consists of n_i instances: $\{\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{n_i}\}$. We write the objective function to optimize as

$$\begin{aligned}(\{y_i^j\}^*, F^*) =& \arg\min_{\{y_i^j\}, F(\cdot)} \sum_{i=1}^{n}\sum_{j=1}^{n_i} \ell(F_{y_i^j}(\mathbf{x}_i^j)) \qquad (1)\\ &\text{s.t. } \forall i : \sum_{j=1}^{n_i} \mathbb{I}(y_i = \arg\max_{k\in\mathcal{Y}} F_k(\mathbf{x}_i^j)) \geq 1.\end{aligned}$$

The objective in this optimization procedure is to minimize a loss function $\ell(\cdot)$ which is defined over the entire set of instances by considering the condition that at least one instance in each bag has to be from the target class. Note that $\mathbb{I}(\cdot)$ is an indicator function and $F_k(\mathbf{x})$ is the confidence of the classifier for the k-th class, *i.e.*, $F_k(\mathbf{x}) = p(k|\mathbf{x}) - \frac{1}{K}$. Often the loss function depends on the classification margin of an instance. In the case of Random Forests, the margin can be written as [13]

$$m(\mathbf{x}, y) = p(y|\mathbf{x}) - \max_{\substack{k\in\mathcal{Y}\\ k\neq y}} p(k|\mathbf{x}) = F_y(\mathbf{x}) - \max_{\substack{k\in\mathcal{Y}\\ k\neq y}} F_k(\mathbf{x}). \tag{2}$$

Note that for a correct classification $m(\mathbf{x}, y) > 0$ should hold. Overall, it can easy be seen that Eq. (1) is a non-convex optimization problem because a random forest has to be trained and simultaneously a suitable set of labels y_i^j has to be found. Due to the integer values of the labels y_i^j, this problem is a type of integer programming and is usually difficult to solve. In order to solve this non-convex optimization problem without loosing too much of the training speed of random forests, we use a fast iterative optimization procedure based on deterministic annealing (DA).

3.1 Optimization

DA [30] is a homotopy method which is able to fast minimize non-convex combinatorial optimization problems. The main idea is to extend a difficult optimization problem with an easier one by adding a convex entropy term and solve this first. In particular, one tries to minimize the entropy $\mathcal{H}$ of the distribution p in form of

$$p^* = \arg\min_{p\in\mathcal{P}} E_p(\mathcal{F}(y)) - T\mathcal{H}(p), \tag{3}$$

where $\mathcal{P}$ is a space of probability distributions and $\mathcal{F}(y)$ is our objective function. The optimization problem is than gradually deformed to its original form using a cooling parameter T, *i.e.*, $T_0 > T_1 > \ldots > T_\infty = 0$. Due to its speed and simplicity, DA-based optimization has been applied to many problems, among them also multiple-instance learning though in context with SVMs, *i.e.*, see [31]. Furthermore, due to the induced randomness in deterministic annealing, it fits to the nature of randomized trees and was recently also used for solving semi-supervised learning problems [32]. For further details on DA we refer the reader to [30].

In order to optimize our MIL objective function (Eq. (1)), we propose the following iterative strategy: In the first iteration, we train a naïve RF that ignores the MIL constraint and uses the corresponding bag labels for instances inside that bag. Then, after the first iteration, we treat the instance labels in target bags as binary variables. These random variables are defined over a space of probability distributions $\mathcal{P}$. We now search a distribution $\hat{\mathbf{p}} \in \mathcal{P}$ for each bag which solves our optimization problem in Eq. (1). Based on $\hat{\mathbf{p}}$ each tree randomly selects the instance labels for training. Hence, based on the optimization of $\hat{\mathbf{p}}$ we try to identify the real but hidden labels of instances.

We reformulate the objective function given in Eq. (1) so that it is suitable for DA optimization

$$\mathcal{L}_{DA}(F, \hat{\mathbf{p}}) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \sum_{k=1}^{K} \hat{p}(k|\mathbf{x}_i^j) \ell(F_k(\mathbf{x}_i^j)) + T \sum_{i=1}^{n} H(\hat{\mathbf{p}}_i), \tag{4}$$

where T is the temperature parameter and

$$H(\hat{\mathbf{p}}_i) = - \sum_{j=1}^{n_i} \sum_{k=1}^{K} \hat{p}(k|\mathbf{x}_i^j) \log(\hat{p}(k|\mathbf{x}_i^j)) \tag{5}$$

is the entropy over the predicted distribution inside a bag. It can be seen that the parameter T steers the importance between the original objective function and the entropy. If T is high, the entropy dominates the loss function and the problem can be easier solved due to the convexity. If $T = 0$ the original loss dominates (Eq. (1)). Hence, DA first solves the easy task of entropy minimization and then by continuously decreasing T from high values to zero gradually solves the original optimization problem, *i.e.*, finding the real but hidden instance labels y and simultaneously training an instance classifier.

In more detail, for a given temperature level, the learning problem can be written as

$$\begin{aligned} (F^*, \hat{\mathbf{p}}^*) =& \arg\min_{\hat{\mathbf{p}}, F(\cdot)} \mathcal{L}_{DA}(F, \hat{\mathbf{p}}) \\ \text{s.t. } \forall i :& \sum_{j=1}^{n_i} \mathbb{I}(y_i = \arg\max_{k \in \mathcal{Y}} F_k(\mathbf{x}_i^j)) \geq 1. \end{aligned} \tag{6}$$

We split this optimization problem up into a two-step convex optimization problem analog to an alternating coordinate descent approach. In the first step, the objective function $\mathcal{F}$ is optimized by fixing the distribution $\hat{\mathbf{p}}$ and optimizing the learning model. In the second step, the distribution p^* over the bags according to the current entropy level is adjusted. Note that both individual steps are convex optimization problems. For a given distribution over the bag samples, we randomly choose a label according to $\hat{\mathbf{p}}$. We repeat this process independently for every tree f in the forest. Hence, in the limit, we will exactly maintain the same distribution over the unlabeled samples as given by $\hat{\mathbf{p}}$. Let $\{\hat{y}_{ij}\}$ be the randomly drawn labels according to the distribution $\hat{\mathbf{p}}$ for m-th tree. The optimization problem for the m-th tree becomes

$$\begin{aligned} f_m^* =& \arg\min_{f} \sum_{i=1}^{n} \sum_{j=1}^{n_i} \ell(f_{\hat{y}_i^j}(\mathbf{x}_i^j)) \\ \text{s.t. } \forall i :& \sum_{j=1}^{n_i} \mathbb{I}(y_i = \arg\max_{k \in \mathcal{Y}} f_k(\mathbf{x}_i^j)) \geq 1. \end{aligned} \tag{7}$$

Algorithm 1. MIForests

```
Require: Bags $\{\mathcal{B}_i\}$
Require: The size of the forest: $M$
Require: A starting heat parameter $T_0$
Require: An ending parameter $T_{min}$
Require: A cooling function $c(T, m)$
Set: $\forall i : \hat{y}_i^j = y_i$
Train the RF: $\mathcal{F} \leftarrow$ trainRF($\{\hat{y}_i^j\}$).
Init epochs: $m = 0$.
while $T_{m+1} \geq T_{min}$ do
    Get the temperature: $T_{m+1} \leftarrow c(T_m, m)$.
    Set $m \leftarrow m + 1$.
    $\forall \mathbf{x}_i^j \in \mathcal{B}_i, k \in \mathcal{Y}$ : Compute $p^*(k|\mathbf{x}_i^j)$
    for $t$ from 1 to $M$ do
        $\forall \mathbf{x}_i^j \in \mathcal{B}_i$ : Select random label, $\hat{y}_i^j$ according to $p^*(\cdot|\mathbf{x}_i^j)$
        Set the label for instance with highest $p^*(\cdot|\mathbf{x}_i^j)$ equal to bag label
        Re-train the tree:
        $f_m \leftarrow$ trainTree($\{\hat{y}_i^j\}$).
    end for
end while
Output the forest $\mathcal{F}$.
```

Since the margin maximizing loss function is convex, this loss function is also convex. In order to not violate the MIL constraint, after having randomly selected instance labels for a bag, we always set the instance with the highest probability according to $\hat{\mathbf{p}}$ equal to the bag label. At this stage we train all the trees in the forest by the formulation given above.

After we trained the random forest, we enter the second stage where we find the optimal distribution according to

$$\hat{\mathbf{p}}^* = \arg\min_{\hat{\mathbf{p}}} \sum_{i=1}^{n} \sum_{j=1}^{n_i} \sum_{k=1}^{K} \hat{p}(k|\mathbf{x}_i^j) \ell(F_k(\mathbf{x}_i^j)) + T \sum_{i=1}^{n} H(\hat{\mathbf{p}}_i). \tag{8}$$

The optimal distribution is found by taking the derivative *w.r.t* p and setting it to zero. We depict all detailed steps of the method in Algorithm 1.

3.2 On-Line MIForests

MIForests as introduced above are trained off-line using a two-step optimization procedure as given in Eq. (4), where in one step the objective function $\mathcal{F}$ is optimized and in the second step the distribution $\hat{\mathbf{p}}$ over the bags, respectively. In order to modify the algorithm so that it is suitable for on-line learning, *i.e.*, the bags B_i arrive sequentially, one has to change both optimization steps to operate in on-line mode. In the following, we show how to train the randomized trees on-line in order to optimize $\mathcal{F}$ and also how $\hat{\mathbf{p}}$ can be optimized on-line to disambiguate the class labels inside positive bags.

Bagging, necessary to build the tree ensemble, can be easily done on-line by modeling the sequentially arriving samples with a Poisson distribution initialized with a constant value λ [33]. On-line learning of the decision trees is less trivial due to their recursive split nature. However, as we recently showed [21] the pure recursive training of the trees can be circumvented by using a tree-growing procedure similar to evolving-trees [34]. In more detail, the algorithm starts with trees consisting only of root nodes and randomly selected node tests f_i and thresholds θ_i. Each node estimates an impurity measure based on the Gini index ($G_i = \sum_{i=1}^{K} p_i^j (1 - p_i^j)$) on-line, where p_i^j is the label density of class i in node K. Then, after each on-line update the possible information gain ΔG during a potential node split is measured. If ΔG exceeds a given threshold β, the node becomes a split node; *i.e.*, it is not updated any more and generates two child leaf nodes. The growing proceeds until a maximum depth is reached. Even when the tree has grown to its full size, all leaf nodes are further on-line updated. The method is simple to implement and has shown to converge fast to its off-line counterpart. For further details we refer the reader to [21].

Besides on-line training of the randomized trees, we also have to perform the deterministic annealing on-line. This means we have to estimate $\hat{p}$ on-line by examining the sequentially arriving samples. Therefore, if a new bag B_i arrives, we initialize a new distribution $\hat{p_i}$ over its instances using the current confidence output of $\mathcal{F}_t$. Then, we iteratively apply the optimization of $\mathcal{F}_t$ and $\hat{p_i}$ only for the current bag B_i following the same two-step procedure and annealing schedule as in the off-line case (Eq. (7),Eq. (8)). Afterwards, B_i is discarded and the training proceeds with the next bag B_{i+1}. We skip the algorithm box due to lack of space.

4 Experiments

The purpose of this section is to evaluate the proposed algorithms on standard MIL machine learning benchmark datasets and to demonstrate their performance on typical computer vision problems such as object tracking. Note that, in general, we abstain from any data set or feature engineering procedures, since the main purpose is to compare the different learning methods.

4.1 Benchmark Datasets

We first evaluate our proposed MIForests on popular benchmark datasets used in most studies of multiple-instance learning algorithms, *i.e.*, the *Musk1 and Musk2* drug activity datasets proposed by Dietterich [2] and the *Tiger, Elephant and Fox* image datasets proposed by Andrews *et al.* [3][2]. For sanity check we also tested common random forests [13], *i.e.*, ignoring the MIL constraint. For all learners we used 50 trees with a maximum depth of 20. As cooling schedule we used a simple exponential function in form of $T_t = e^{-t \cdot C}$, where t is the current

[2] Sample C++ code is available at `http://www.ymer.org/amir/software/milforests`

Table 1. Results and comparisons in terms of percent classification accuracy on popular MIL benchmark datasets. We report the average over 5 runs. Best methods with the error margin are marked in bold face.

Method	Elephant	Fox	Tiger	Musk1	Musk2
RandomForest [13]	74	60	77	85	78
MIForest	**84**	**64**	82	85	82
MI-Kernel [3]	**84**	60	**84**	**88**	**89**
MI-SVM [36]	81	59	**84**	78	84
mi-SVM [36]	82	58	79	87	84
MILES [7]	81	62	80	**88**	83
SIL-SVM	**85**	53	77	**88**	**87**
AW-SVM [31]	82	**64**	83	86	84
AL-SVM [31]	79	63	78	86	83
EM-DD [26]	78	56	72	85	85
MILBoost-NOR [8]	73	58	56	71	61

iteration and the constant $C = \frac{1}{2}$. We determined these settings empirically and kept them fixed over all experiments.

As can be observed, the performance of the individual approaches varies highly depending on the data set. The experiments show that MIForests achieve state-of-the-art performance and are even outperforming several SVM-based approaches and those based on boosting. Especially for the vision problems, we are always among the best. Also the naïve RF approach yields surprisingly good performance, especially on *Fox* and *Musk1*; however, it cannot take pace with the performance of its MIForest counterpart. One explanation for this might be that RFs are less susceptible to noise compared to other learning methods, which is necessary for the naïve approach [22]. Compared to its most similar SVM variant (AL-SVM), MIForest outperforms it on two datasets, draws on one and performs worse on two. Finally, it has to be mentioned that especially for [31] and [35] better results can be achieved by incorporating prior knowledge into the learners, *e.g.*, how many “real” positives exist inside bags; which however also holds for MIForests.

4.2 Corel Dataset

Here, we evaluate our proposed methods on the Corel-1000 and Corel-2000 image dataset for region-based image classification. The data set consists of 2000 images with 20 different categories. Each image corresponds to a bag consisting of instances obtained via oversegmentation. It is thus a typical MIL problem. In order to allow for fair comparison we used the same data settings and features as proposed by Chen *et al.* [7]. For the results we used the same settings as in our previous experiments. In contrast to most other approaches, we did not train 20 1-vs.-all classifiers, but trained one multi-class forest, which is usually a more difficult task. We compare MIForests with MILES, the original algorithm proposed on this data set [7]. Since MILES is a binary algorithm we trained

Table 2. Results and comparisons on the COREL image categorization benchmark. Additionally, we depict the training times in seconds.

Method	Corel-1000	Corel-2000	1000 Images	2000 Images
MIForest	59	66	4.6	22.0
MILES	58	67	180	960

20 1-vs.-all MILES classifiers and depict the results in Table 2. As can be seen, MIForests achieve competitive results for multi-class scenarios, however, being much faster. We measured the average time on a standard Core Duo machine with 2.4 Ghz.

4.3 Object Tracking

A recent dominating trend in tracking called "tracking by detection" has shown to deliver excellent results at real-time speeds. In these methods, usually an appearance-based classifier is trained with a marked object at the first frame versus its local background [37]. The object is then tracked by performing re-detection in the succeeding frames. In order to handle rapid appearance and illumination changes, recent works, *e.g.*, [38], use on-line self-updating of the classifiers. However, during this process it is not clear where to select the positive and negative updates necessary for self-updating. If the samples are selected wrongly, slight errors can accumulate over time (*a.k.a* label jitter) and cause drifting. Recently, Babenko *et al.* [12] demonstrated that label jitter can be handled by formulating the update process using an on-line MIL boosting algorithm. Using MIL, the allowed positive update area around the current tracker can be increased and the classifier resolves the ambiguities by itself, yielding more robust tracking results. See [12] for a more detailed discussion about the usefulness of MIL for tracking. In the following, we demonstrate that on-line MIForests can also give excellent tracking results, outperforming the state-of-the-art tracker based on boosting.

We focus on tracking arbitrary objects; so there is no prior knowledge about the object class available except its initial position. We use eight publicly available sequences including variations in illumination, pose, scale, rotation and appearance, and partial occlusions. The sequences *Sylvester* and *David* are taken from [39] and *Face Occlusion 1* is taken from [40], respectively. *Face occlusion 2*, *Girl*, *Tiger1,Tiger2* and *Coke* are taken from [12]. All video frames are gray scale and of size 320 × 240. To show the real accuracy of the compared tracking methods, we use the overlap-criterion of the VOC Challenge [41], which is defined as $A_{overlap} = R_T \cap R_{GT} / R_T \cup R_{GT}$, where R_T is the tracking rectangle and R_{GT} the groundtruth. Since we are interested in the alignment accuracy of our tracker and the tracked object, rather than just computing the raw distance we measure the accuracy of a tracker by computing the average detection score for the entire video. Note that values between 0.5 and 0.7 are usually acceptable results, values larger than 0.7 can be considered as almost perfect.

The main purpose of the tracking experiments is the comparison of the influence of the different on-line learning methods. Hence, we use simple Haar-like features for representation, did not implement any rotation or scale search and avoid any other engineering methods, although these things would definitely improve the overall results. For MIForests, we used 50 trees with depth 10 and the same annealing schedule as in the ML experiments. Overall, we generate 500 features randomly. As [12] for all boosting methods, we used 50 selectors with each 250 weak classifiers which results in a featurepool of size 12500.

In Table 3 we depict detailed results for all tracking sequences compared to MILBoost [12], SemiBoost (OSB) [42], on-line AdaBoost (OAB)[38] and on-line random forests (ORF) [21]. As can be seen, MIForests perform best on seven tracking sequences. Remarkably, we are able to outperform MILBoost, which is currently known to be amongst the best tracking methods, on 6 out of 8 sequences, draw on 1 and are slightly worse on 1. The resulting tracking videos can be found in the supplementary material.

Table 3. Tracking results on the benchmark sequences measured as average detection window and ground truth overlap over 5 runs per sequence. Best performing method is marked in bold face.

Method	*sylv*	*david*	*faceocc2*	*tiger1*	*tiger2*	*coke*	*faceocc1*	*girl*
MIForest	0.59	**0.72**	**0.77**	**0.55**	**0.53**	**0.35**	**0.77**	**0.71**
MILBoost	**0.60**	0.57	0.65	0.49	**0.53**	0.33	0.60	0.53
OSB	0.46	0.31	0.63	0.17	0.08	0.08	0.71	0.69
OAB	0.50	0.32	0.64	0.27	0.25	0.25	0.47	0.38
ORF	0.53	0.69	0.72	0.38	0.43	**0.35**	0.71	0.70

5 Conclusion

In this paper, we presented a new multiple-instance learning method based on randomized trees (MILForest). We define the labels of instances inside positive bags as random variables and use a deterministic-annealing style procedure in order to find the true but hidden labels of the samples. In order to account for the increasing number of data and leverage the usage of our method in streaming data scenarios, we also showed how to extend MILForests for on-line learning. We demonstrated that MILForests are competitive to other methods on standard visual MIL benchmark datasets while being faster and inherently multi-class. We demonstrated the usability of the on-line extension on the task of visual object tracking where we outperformed state-of-the-art methods. In future work, we plan to test our algorithm on other vision applications such as object detection and categorization.

References

1. Keeler, J., Rumelhart, D., Leow, W.: Integrated segmentation and recognition of hand-printed numerals. In: NIPS (1990)
2. Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple-instance problem with axis-paralle rectangles. In: Artifical Intelligence (1997)

3. Andrews, S., Tsochandaridis, I., Hofman, T.: Support vector machines for multiple-instance learning. Adv. Neural. Inf. Process. Syst. 15, 561–568 (2003)
4. Ruffo, G.: Learning single and multiple instance decision trees for computer security applications. PhD thesis (2000)
5. Zhang, M.L., Goldman, S.: Em-dd: An improved multi-instance learning technique. In: NIPS (2002)
6. Vijayanarasimhan, S., Grauman, K.: Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In: CVPR (2008)
7. Chen, Y., Bi, J., Wang, J.: Miles: Multiple-instance learning via embedded instance selection. In: IEEE PAMI (2006)
8. Viola, P., Platt, J., Zhang, C.: Multiple instance boosting for object detection. In: NIPS (2006)
9. Zhang, C., Viola, P.: Multiple-instance pruning for learning efficient cascade detectors. In: NIPS (2008)
10. Stikic, M., Schiele, B.: Activity recognition from sparsely labeled data using multi-instance learning. In: Choudhury, T., Quigley, A., Strang, T., Suginuma, K. (eds.) LoCA 2009. LNCS, vol. 5561, pp. 156–173. Springer, Heidelberg (2009)
11. Vezhnevets, A., Buhmann, J.: Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In: CVPR (2010)
12. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR (2009)
13. Breiman, L.: Random forests. In: Machine Learning (2001)
14. Moosmann, F., Triggs, B., Jurie, F.: Fast discriminative visual codebooks using randomized clustering forests. In: NIPS, pp. 985–992 (2006)
15. Caruana, R., Karampatziakis, N., Yessenalina, A.: An empirical evaluation of supervised learning in high dimensions. In: ICML (2008)
16. Sharp, T.: Implementing decision trees and forests on a gpu. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 595–608. Springer, Heidelberg (2008)
17. Gall, J., Lempinsky, V.: Class-specific hough forests for object detection. In: CVPR (2009)
18. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image catergorization and segmentation. In: CVPR (2008)
19. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: ICCV (2007)
20. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. In: CVPR (2006)
21. Saffari, A., Leistner, C., Godec, M., Santner, J., Bischof, H.: On-line random forests. In: OLCV (2009)
22. Blum, A., Kalai, A.: A note on learning from multiple instance examples. In: Machine Learning, pp. 23–29 (1998)
23. Mangasarian, O., Wild, E.: Multiple-instance learning via successive linear programming. Technical report (2005)
24. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: ICML (2000)
25. Maron, O., Lozano-Perez, T.: A framework for multiple-instance learning. In: NIPS (1997)
26. Zhang, Q., Goldman, S.: Em-dd: An improved multiple instance learning technique. In: NIPS (2001)
27. Foulds, J., Frank, E.: Revisiting multi-instance learning via embedded instance selection. LNCS. Springer, Heidelberg (2008)

28. Blockeel, H., Page, D., Srinivasan, A.: Multi-instance tree learning. In: ICML (2005)
29. Geman, Y.A.D.: Shape quantization and recognition with randomized trees. Neural Computation (1996)
30. Rose, K.: Deterministic annealing, constrained clustering, and optimization. In: IJCNN (1998)
31. Gehler, P., Chapelle, O.: Deterministic annealing for multiple-instance learning. In: AISTATS (2007)
32. Leistner, C., Saffari, A., Santner, J., Bischof, H.: Semi-supervised random forests. In: ICCV (2009)
33. Oza, N., Russell, S.: Online bagging and boosting. In: Proceedings Artificial Intelligence and Statistics, pp. 105–112 (2001)
34. Pakkanen, J., Iivarinen, J., Oja, E.: The evolving tree—a novel self-organizing network for data analysis. Neural Process. Lett. 20, 199–211 (2004)
35. Bunescu, R., Mooney, R.: Multiple instance learning for sparse positive bags. In: ICML (2007)
36. Zhou, Z.H., Sun, Y.Y., Li, Y.F.: Multi-instance learning by treating instances as non-i.i.d. samples. In: ICML (2009)
37. Avidan, S.: Ensemble tracking. In: CVPR, vol. 2, pp. 494–501 (2005)
38. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR (2006)
39. Ross, D., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. In: IJCV (2008)
40. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR (2006)
41. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object class challenge 2007. In: VOC (2007)
42. Grabner, H., Leistner, C., Bischof, H.: On-line semi-supervised 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)

Manifold Valued Statistics, Exact Principal Geodesic Analysis and the Effect of Linear Approximations

Stefan Sommer[1], François Lauze[1], Søren Hauberg[1], and Mads Nielsen[1,2]

[1] Dept. of Computer Science, Univ. of Copenhagen, Denmark
sommer@diku.dk
[2] Nordic Bioscience Imaging, Herlev, Denmark

Abstract. Manifolds are widely used to model non-linearity arising in a range of computer vision applications. This paper treats statistics on manifolds and the loss of accuracy occurring when linearizing the manifold prior to performing statistical operations. Using recent advances in manifold computations, we present a comparison between the non-linear analog of Principal Component Analysis, Principal Geodesic Analysis, in its linearized form and its exact counterpart that uses true intrinsic distances. We give examples of datasets for which the linearized version provides good approximations and for which it does not. Indicators for the differences between the two versions are then developed and applied to two examples of manifold valued data: outlines of vertebrae from a study of vertebral fractures and spacial coordinates of human skeleton end-effectors acquired using a stereo camera and tracking software.

Keywords: manifolds, Riemannian metrics, linearization, manifold valued statistics, Principal Geodesic Analysis (PGA), Geodesic PCA.

1 Introduction

This paper treats the effect of linearization when using the non-linear analog of Principal Component Analysis, Principal Geodesic Analysis (PGA, [1]), to estimate the variability in sets of manifold valued data. Until recently, PGA has been performed by linearizing the manifold, which distorts intrinsic distances, but with the introduction of more powerful computational tools [2], PGA can now be computed with true intrinsic distances. We show how simple and fast indicators allow us to approximate the differences between linearized PGA and exact PGA with true intrinsic distances and evaluate the effect of the linearization.

As a test case for the indicators, we perform a comparison between two manifold valued datasets: outlines of vertebrae from a study of vertebral fractures, and human skeleton end-effectors in spatial coordinates recorded using a stereo camera and tracking software. We will show that linearized PGA provides a reasonable approximation in only one of the experiments and that the indicators allow us to predict this before doing the time-intensive computation of exact PGA with intrinsic distances.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 43–56, 2010.

1.1 Motivation

A wide variety of problems in computer vision possess non-linear structure and are therefore naturally modeled using Riemannian geometry. In diffusion tensor imaging [3,4,5], for image segmentation [6] and registration [7], shape spaces [8], and human motion modeling [9,10], Riemannian manifolds have been used to enforce consistency in data, provide dimensionality reduction, and define more accurate metrics. The wide applicability of manifolds in modeling problems has created the need for statistical tools for manifold data.

Generalizing linear statistical operations to manifolds [1,11,12,13] provides examples of the theoretical and computational problems arising when departing from familiar Euclidean spaces. The tools developed when pursuing this have been used successfully for a range of computer vision applications, and the area is the subject of active research [2,13]. Depending on the level of approximation used in the computations, manifold statistics can be hard to carry out in practice because operations such as finding distances and performing optimization do not admit the closed-form solutions often found in Euclidean spaces [1].

One way of doing manifold statistics is projecting the set of manifold valued data points to the tangent space of a mean point of the manifold. The vector space structure of the tangent space brings back convenient Euclidean statistics, but the distortion of the distances between the data points inherent in the linearization may however lead to sub-optimal solutions to the statistical problems. In contrast to this, some statistical operations can be carried out with true intrinsic manifold distances giving a true picture of the data [2,13]. This, however, often comes at the cost of increased computational complexity and requires conditions on the locality of data.

Because of the trade-offs between convenient linearization and exact modeling, we seek for ways to evaluate the extent of the distortion between the linearized data and true manifold data; we are interested in determining if performing statistics with intrinsic distances offers significant advantages over the linearized approach. Such knowledge has the potential of saving substantial computation time and to improve results of statistical operations.

1.2 Related Work

The mathematical aspects of manifolds are covered extensively in the literature with [14,15] providing good references. Numerical and computational aspects of interest in a general setting are considered in the theoretical papers [16,17] while more specific shape related applications are proposed in [18,19,20].

Both the mathematical community, e.g. [11], and more applied fields, computer vision in particular [1,12], have worked with different aspect of statistics on manifolds. A recent wave of interest by statisticians [21,13] has created new methods with strong links to tools developed in computer vision [13].

The manifold generalization of linear PCA, PGA, was first introduced in [22], but it was formulated in the form most widely used in [1]. It has subsequently been used for several applications. To mention a few, the authors in [1,4] study variations of medial atoms, [23] uses a variation of PGA for facial classification, [24] presents examples on motion capture data, and [20] applies PGA to vertebrae outlines. The algorithm presented in [1] for computing PGA with linearization has been most widely used. In contrast to this, [24] computes PGA as defined in [22] without approximations, but only for a specific manifold, the Lie group SO(3). By using ODE formulations of geodesics and taking derivatives, [2] provides algorithms for computing PGA without approximations on wide classes of manifolds.

Geodesic PCA (GPCA, [13,21]) is in many respects close to PGA but optimizes for the placement of the center point and minimizes projection residuals along geodesics instead of maximizing variance in geodesic subspaces. GPCA uses no linear approximation, but it is currently only computed on spaces where explicit formulas for geodesics exist and on quotients of such spaces.

1.3 Content and Outline

In the next section, we discuss the benefits of using manifolds in modeling, manifold valued statistics, and linearization. Then, in section 3, we consider in detail the specific case of Principal Geodesic Analysis and use synthetic examples to explain the differences between linearized PGA and exact PGA with true intrinsic distances. We progress to developing indicators of these differences, and, in section 4, we compare linearized and intrinsic PGA on real-life examples of manifold valued datasets and analyze the power of the indicators. The paper thus contributes by

(1) developing simple and fast indicators of the difference between linearized PGA and exact PGA that show the effect of linearization,
(2) giving examples of the differences between linearized PGA and exact PGA on real-life datasets from computer vision,
(3) and showing the power of the indicators when applied to the datasets.

2 Manifolds and Manifold Valued Statistics

The interest in manifolds as modeling tools arises from the non-linearity apparent in a variety of problems. We will in the following exemplify this by considering the pose of a human skeleton captured by e.g. a tracking system or motion capture equipment. Consider the position of a moving hand while the elbow and the rest of the body stay fixed. The hand cannot move freely as the length of the lower arm restricts it movement. Linear vector space structure is not present; if we multiply the position of the hand by a scalar, the length of the arm would in general change in order to accommodate the new hand position. Even switching to an angular representation of the pose of the elbow joint will not help; angles have inherent periodicity, which is not compatible with vector space structure.

Though the space of possible hand positions is not linear, it has the structure of a manifold since it possesses the property that it locally can be approximated by a vector space. Furthermore, we can, in a natural way, equip it with a Riemannian metric [14], which allows us to make precise notions of length of curves on the space and intrinsic acceleration. This in turns defines the Riemannian manifold equivalent of straight lines: geodesics. The length of geodesics connecting points defines a distance metric on the manifold.

2.1 Benefits from Modeling Using Manifolds

The main advantages of introducing manifolds in modeling are as follows: consistency in representation, dimensionality reduction, and accuracy in measurements. Consistency ensures the modeled object satisfies the requirements making up the manifold; when moving the position of the hand on the manifold, we are certain the length of the lower arm is kept constant. Such requirements reduce the number of degrees of freedom and hence provide dimensionality reduction. Consistency and dimensionality reduction are therefore closely linked.

Accuracy is connected to the distance measure defined by the Riemannian metric. A reasonable measure of the distance between two positions of the hand will be the length of the shortest curve arising when moving the hand between the positions. Such a curve will, in this example, be a circular arc, and, in the manifold model, the distance will be the length of the arc. In the vector space model, however, the distance will be the length of the straight line connecting the hand positions and, hence, will not reflect the length of an allowed movement of the hand. The manifold model therefore gives a more accurate distance measure.

2.2 Linearizing the Manifold

By linearizing the manifold to the tangent space of a mean point, we can in many applications ensure consistency, but not accuracy, in statistical operations. Let M be a manifold and $\{x_1, \ldots, x_N\}$ a dataset consisting of points on the manifold. An intrinsic mean [11] is defined as a solution to the optimization problem

$$\mu = \operatorname{argmin}_q \sum_{i=1}^{N} d(x_i, q)^2 \tag{1}$$

with $d(x_i, q)$ denoting the manifold distance between the ith data point and the mean candidate q.

Each point p of a manifold has a connected linear space called the tangent space and denoted T_pM. The dimension of T_pM is equal to the dimension of the manifold, which, as in the vector space case, specifies the number of degrees of freedom. Vectors in the tangent space are often mapped back to the manifold using the exponential map, Exp_p, which maps straight lines trough the origin of T_pM to geodesics on M passing p.

If we consider the tangent space of an intrinsic mean, $T_\mu M$, we can represent x_i by vectors w_i in $T_\mu M$ such that $\operatorname{Exp}_\mu w_i = x_i$.[1] The map that sends $x_i \in M$

[1] See Figure 1 for an example of a 2-dimensional manifold with sampled elements of the tangent space of the mean and corresponding points on the manifold.

to $w_i \in T_\mu M$ is called the logarithm map and denoted Log_μ. The vector space structure of $T_\mu M$ allows us to use standard statistical tools on $\{w_1, \ldots, w_N\}$. We could for example infer some distribution in $T_\mu M$, sample a vector v from it, and project the result back to a point p on the manifold so that $p = \mathrm{Exp}_\mu v$. It is important to note that consistency is ensured in doing this; p will be on the manifold and hence satisfy the encoded requirements. Turning to the example of hand positions, we have found a consistent way of sampling hand positions without violating the fixed length of the lower arm.

The above procedure can be seen as a way of linearizing the manifold around the intrinsic mean μ because the tangent space $T_\mu M$ provides a first order approximation of the manifold around μ. Yet, distances between vectors in $T_\mu M$ do not always reflect the manifold distances between the corresponding points on the manifold: distances between w_i and the origin of $T_\mu M$ equal the distances $d(x_i, \mu)$, but the inter-point distances $d(x_i, x_j)$ are not in general equal to the tangent space distances$\|w_i - w_j\|$. Accuracy may therefore be lost as a result of the approximation. In short, linearization preserves consistency but may destroy accuracy.

3 Principal Geodesic Analysis

Principal Component Analysis (PCA) is widely used to model the variability of datasets of vector space valued data and provide linear dimensionality reduction. PCA gives a sequence of linear subspaces maximizing the variance of the projection of the data or, equivalently, minimizing the reconstruction errors. The kth subspace is spanned by an orthogonal basis $\{v^1, \ldots, v^k\}$ of principal components v^i.

PCA is dependent on the vector space structure and hence cannot be performed on manifold valued datasets. Principal Geodesic Analysis was developed to overcome this limitation. PGA centers its operations at a point $\mu \in M$ with μ usually being an intrinsic mean of the dataset $\{x_1, \ldots, x_N\}$, and finds geodesic subspaces, which are images $S = \mathrm{Exp}_\mu V$ of linear subspaces V of the tangent space $T_\mu M$. A projection operator π_S is defined by letting $\pi_S(x)$ be a point in S closest to x. The kth geodesic subspace S_k is then given as $\mathrm{Exp}_\mu(V_k)$, $V_k = \mathrm{span}\,\{v^1, \ldots, v^k\}$, where the principal directions v^i are given recursively by

$$\begin{aligned} v^i &= \mathrm{argmax}_{\|v\|=1, v \in V_{i-1}^\perp} \frac{1}{N} \sum_{j=1}^{N} d(\mu, \pi_{S_v}(x_j))^2 \ , \\ S_v &= \mathrm{Exp}_\mu(\mathrm{span}\,(V_{i-1}, v)) \ . \end{aligned} \tag{2}$$

The term being maximized is the sample variance, the expected value of the squared distance to μ. PGA therefore extends PCA by finding geodesic subspaces in which variance is maximized.

Since the projection $\pi_{S_k}(x)$ is hard to compute, PGA is traditionally approximated by linearizing the manifold. The data $x_1, \ldots, x_N$ are projected to $T_\mu M$ using Log_μ, and regular PCA is performed on $w_i = \mathrm{Log}_\mu x_i$. Equation (2) then becomes

$$v^i \approx \operatorname{argmax}_{\|v\|=1, v\in V_{i-1}^{\perp}} \frac{1}{N}\sum_{j=1}^{N}\left(\langle w_j, v\rangle^2 + \sum_{l=1}^{k-1}\langle w_j, v^l\rangle^2\right) . \tag{3}$$

We can define a normal distribution $\mathcal{N}$ in $T_\mu M$ using the result of the PCA procedure, and, in doing so, we have performed the procedure described in section 2.2. We will refer to PGA with the approximation as *linearized* PGA. PGA as defined by (2) without the approximation will be referred to as *exact* PGA. Advances in manifold computations allow exact PGA to be computed on the Lie group SO(3) [24] and, more recently, on wide classes of manifolds [2].

Replacing maximization of the sample variances $d(\mu, \pi_{S_v}(x_j))^2$ by minimization of the squared reconstruction errors $d(x_j, \pi_{S_v}(x_j))^2$, we obtain another manifold extension of PCA and thus an alternate definition of PGA:

$$v^i = \operatorname{argmin}_{\|v\|=1, v\in V_{i-1}^{\perp}} \frac{1}{N}\sum_{j=1}^{N} d(x_j, \pi_{S_v}(x_j))^2 . \tag{4}$$

In contrast to vector space PCA, the two definitions are not equivalent. It can be shown that, in some cases, solutions to (2) will approach parts of the manifold where the cost function is non differentiable, a problem we have not encountered when solving for (4). We are currently working on a paper giving a theoretical treatment of this phenomenon and other differences between the definitions. The latter formulation is chosen for Geodesic PCA to avoid similar instabilities of variance maximization [13]. In correspondence with this, we will use (4) in the rest of the paper, but we stress that this choice is made only to avoid instabilities in (2) and that all computations presented can be performed using the former definition with only minor changes to the optimization algorithms [2].

3.1 Linearized PGA vs. Exact PGA

Computing the projection map π_S is particularly time-intensive causing the computation of exact PGA to last substantially longer than linearized PGA. To give an example, computing linearized PGA for one of the datasets later in this paper takes 5 seconds with a parallelized Matlab implementation, and computing exact PGA for the same example requires approximately 10 minutes. This time penalty makes it is worth considering the actual gain of computing exact PGA. We will in this section give examples of low dimensional manifolds on which it is possible visually to identify the differences between the methods.

We consider surfaces embedded in $\mathbb{R}^3$ and defined by the equation

$$S_c = \{(x,y,z) | cx^2 + y^2 + z^2 = 1\} \tag{5}$$

for different values of the scalar c. For $c > 0$, S_c is an ellipsoid and equal to the sphere $\mathbb{S}^2$ in the case $c = 1$. The surface S_0 is a cylinder and, for $c < 0$, S_c is an hyperboloid. Consider the point $p = (0,0,1)$ and note that $p \in S_c$ for all c. The curvature of S_c at p is equal to c. Note that in particular for the cylinder case

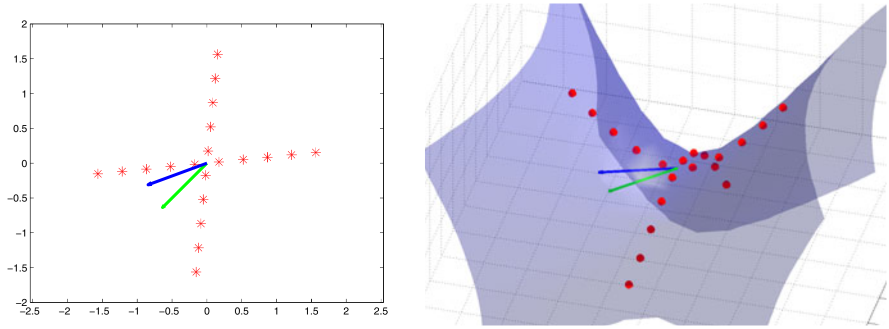


Fig. 1. T_pS_{-2} with sampled points and first principal components (blue exact PGA, green linearized PGA) (left) and S_{-2} with projected points and first principal components (blue exact PGA ([2]), green linearized PGA) (right)

the curvature is zero; the cylinder locally has the geometry of the plane $\mathbb{R}^2$ even though it informally seems to curve.

We evenly distribute 20 points along two straight lines through the origin of the tangent space T_pS_c, project the points from T_pS_c to the surface S_c, and perform linearized and exact PGA. Since linearized PCA amounts to Euclidean PCA in T_pS_c, the first principal component divides the angle between the lines for all c. In contrast to this, the corresponding residuals and the first principal component found using exact PGA are dependent on c. Table 1 shows the angle between the principal components found using the different methods, the average squared residuals and differences between squared residuals for different values of c. Let us give a brief explanation of the result. The symmetry of the sphere and the dataset causes the effect of curvature to even out in the spherical case S_1. The cylinder S_0 has local geometry equal to $\mathbb{R}^2$ which causes the equality between the methods in the $c = 0$ case. The hyperboloids with $c < 0$ are non-symmetric causing a decrease in residuals as the first principal component approaches the hyperbolic axis. This effect increases with curvature causing the the first principal component to align with this axis for large negative values of c.

It is tempting to think that increasing absolute curvature causes increasing differences between the methods. Yet, redoing the experiment with the lines rotated by $\pi/4$ making them symmetric around the x and y axes will produce vanishing differences. Curvature in itself, therefore, does not necessarily imply

Table 1. Differences between methods for selected values of c

c:	**1**	**0.5**	**0**	**-0.5**	**-1**	**-1.5**	**-2**	**-3**	**-4**	**-5**
angle (°):	0.0	0.1	0.0	3.4	14.9	22.2	24.8	27.2	28.3	28.8
lin. sq. res.:	0.251	0.315	0.405	0.458	0.489	0.508	0.520	0.534	0.539	0.541
exact sq. res.:	0.251	0.315	0.405	0.458	0.478	0.482	0.485	0.489	0.491	0.492
diff (%):	0.0	0.0	0.0	0.1	2.3	5.1	6.7	8.4	8.9	9.0

large differences, and the actual differences are hence dependent on both curvature and the dataset.

3.2 The Difference Indicators

The projection π_S is in (3) approximated using the orthogonal projection in the tangent space $T_\mu M$. We let τ_S denote the difference in residuals arising when using the two projections and aim at approximating τ_S to give an estimate of the gain in precision obtained by using true projections. The subspaces optimizing (4) and (3) will in general differ due to the different projection methods and the fact that residuals are approximated by tangent space distances in (3). We let ρ denote the difference in residuals between the projection of the data to the two subspaces, and we aim at approximating ρ to indicate the gain in accuracy when computing exact PGA.

We start by giving precise definitions for τ_S and ρ before deriving the indicators $\tilde{\tau}_S$ and σ of their values. The term indicators is used to emphasize expected correlation between the values of e.g. τ_S and the indicator $\tilde{\tau}_S$ but with no direct expression for the correlation.

Assume $v_1, \ldots, v_{k-1}$ are principal components and let $v \in T_\mu M$ be such that $v_1, \ldots, v_{k-1}, v$ constitues an orthonormal basis. Let the geodesic subspace S_v be given by $\mathrm{Exp}_\mu \mathrm{span}\,\{v_1, \ldots, v_{k-1}, v\}$, and let $w_j = \mathrm{Log}_\mu x_j$ for each element of the dataset $\{x_1, \ldots, x_N\}$. We denote by $\hat{\pi}_{S_v}(x_j)$ the point on the manifold corresponding to the orthogonal tangent space projection of w_j, i.e.

$$\hat{\pi}_S(x_j) = \mathrm{Exp}_\mu \left(\langle w_j, v \rangle v + \sum_{l=1}^{k-1} \left\langle w_j, v^l \right\rangle v^l \right) , \tag{6}$$

and define the average projection difference

$$\tau_S = \frac{1}{N} \sum_{j=1}^{N} \left(d(x_j, \hat{\pi}_{S_v}(x_j))^2 - d(x_j, \pi_{S_v}(x_j))^2 \right) . \tag{7}$$

Let now v be an exact PGA principal geodesic component computed using (4) and let $\hat{v}$ be a linearized PGA principal component computed using (3). We let S_v and $S_{\hat{v}}$ denote the geodesic subspaces corresponding to v and $\hat{v}$. The average residual difference is then given by

$$\rho = \frac{1}{N} \sum_{j=1}^{N} \left(d(x_j, \pi_{S_{\hat{v}}}(x_j))^2 - d(x_j, \pi_{S_v}(x_j))^2 \right) . \tag{8}$$

Note that both τ_S and ρ are positive since π_{S_v} minimizes residuals and v minimizes (4).

3.3 The Projection Difference

Since $\pi_{S_v}(x_j)$ is the point in S_v closest to x_j, the differences expressed in each term of (7) measure the difference between $f(\hat{\pi}_{S_v}(x_j))$ and $f(y_j)$ with $y_j \in S_v$

minimizing the map $f(y) = d(x_j, y)^2$. The gradient $\nabla_y f$ vanishes in such a minimum leading us to approximate the difference by the norm of the gradient at $\hat{\pi}_{S_v}(x_j)$. The gradient is readily evaluated since it is given by the component of $-2\mathrm{Log}_{\hat{\pi}_{S_v}(x_j)}(x_j)$ in the tangent space of S_v [11]. We use this to approximate τ_S by

$$\tau_{S_v} \approx \tilde{\tau}_{S_v} = \frac{2}{N} \sum_{j=1}^{N} \|\nabla_{\hat{\pi}_{S_v}(x_j)} f\| \tag{9}$$

and note that each term of the sum, and therefore the entire indicator $\tilde{\tau}_{S_v}$, is inexpensive to compute.

3.4 The Residual Difference

We now heuristically derive an indicator σ that is correlated with ρ. The correlation will be confirmed later by the experiments. Assume for a moment that distances in the tangent space $T_\mu M$ approximate the true manifold distances well. The residual sums $\frac{1}{N}\sum_{j=1}^{N} d(x_j, \pi_{S_{\hat{v}}}(x_j))^2$ and $\frac{1}{N}\sum_{j=1}^{N} d(x_j, \pi_{S_v}(x_j))^2$ will then be close to identical since v is chosen to minimize the latter sum, and $\hat{v}$ is chosen to minimize the sum of tangent space residuals. The difference ρ will therefore be close to zero. Conversely, assume that distances in the tangent space differ greatly from the true manifold distances. On constant curvature spaces like the sphere S_1, these distance differences will generally be uniformly distributed causing the linearized principal component $\hat{v}$ to be close to v and ρ therefore close to zero. On the contrary, the distance differences will vary on spaces with non-constant curvature like S_{-1} where $\hat{v}$ in general is far from v causing ρ to be large. We therefore expect ρ to be correlated with the standard deviation σ of the differences between the tangent space residual approximations and the actual orthogonal projection residuals,

$$\sigma = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\Big(\|w_j - \mathrm{Log}_\mu(\hat{\pi}_{S_{\hat{v}}})\| - d(x_j, \hat{\pi}_{S_{\hat{v}}}(x_j)) - \mu\Big)^2}\,, \tag{10}$$

with μ the mean value of the scalars $\|w_j - \mathrm{Log}_\mu(\hat{\pi}_{S_{\hat{v}}})\| - d(x_j, \hat{\pi}_{S_{\hat{v}}}(x_j))$. We use σ, which again is fast to compute, to indicate the size of ρ.

4 Experiments

We present experiments on the synthetic data of section 3.1 and on two real-life datasets for two purposes: the experiments will show examples where computing exact PGA results in increased accuracy as well as examples where linearized PGA performs well, and the power of the indicators developed in section 3 will be explored.

When investigating the correlation between the indicator $\tilde{\tau}_{S_{\hat{v}}}$ and the projection difference $\tau_{S_{\hat{v}}}$, we let $\hat{v}$ be the first principal component computed using linearized PGA. In addition, we compare the residual difference ρ with the indicator σ.

4.1 Synthetic Data

We test the indicators on the manifolds S_c with the synthetic data described in section 3.1. Figure 2 shows τ_S as a function of the indicator $\tilde{\tau}_{S_{\hat{v}}}$ and ρ as a function of the indicator σ for each value of c. For both graphs, we see correlation between the indicators and actual differences. For $c = 1$ and $c = 0.5$, σ is relatively high compared to ρ stressing that the indicators only give approximations and that, if full precision is required, exact PGA should be computed.

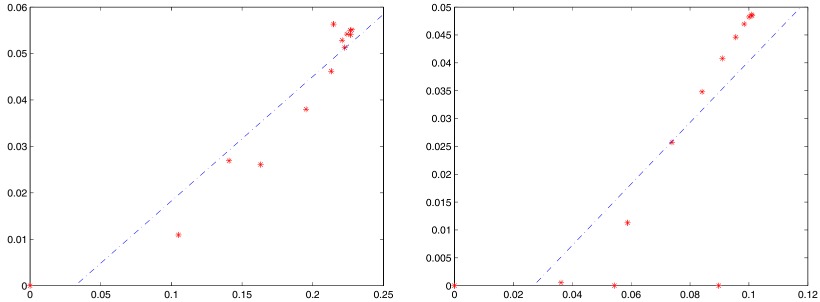


Fig. 2. Synthethic data: Projection difference $\tau_{S_{\hat{v}}}$ as a function of the indicator $\tilde{\tau}_{S_{\hat{v}}}$ with the broken line fitted to the points (left) and residual difference ρ as a function of the indicator σ with the broken line fitted to the points (right)

4.2 Vertebrae Outlines

In this experiment, we consider outlines of vertebrae obtained in a study of vertebral fractures. The dataset of 36 lateral X-rays have been manually annotated by medical experts to identify the outline of the vertebra of each image. To remove variability in the number and placement of points, a resampling is performed to ensured constant inter-point distances. With this equidistance property in mind, the authors in [20] define a submanifold of $\mathbb{R}^{2n}$ on which the outlines naturally reside. We give a brief review of the setup but refer to the paper for details. The equidistance constraint is encoded using a map $F : \mathbb{R}^{2n} \to \mathbb{R}^{n-2}$ with components

$$F^i(P_1, ..., P_n) = d_{i+2,i+1} - d_{i+1,i}, \quad i = 1, .., n-2 \tag{11}$$

with n the number of points and $d_{i,j} = (x_i - x_j)^2 + (y_i - y_j)^2$ the squared distances between points P_i and P_j. The constraint is satisfied for a vertebra outline $c = \{P_1, \ldots, P_n\}$ if $F(c) = 0$. An additional constraint is added to remove scaling effects by ensuring the outline reside on the unit sphere. The preimage $A_n = F^{-1}(0)$ is then a submanifold of $\mathbb{R}^{2n}$, the space of equidistant vertebra outlines. We choose 8 random outlines from the dataset and perform linearized PGA and exact PGA. The experiment consists of 20 such selections,

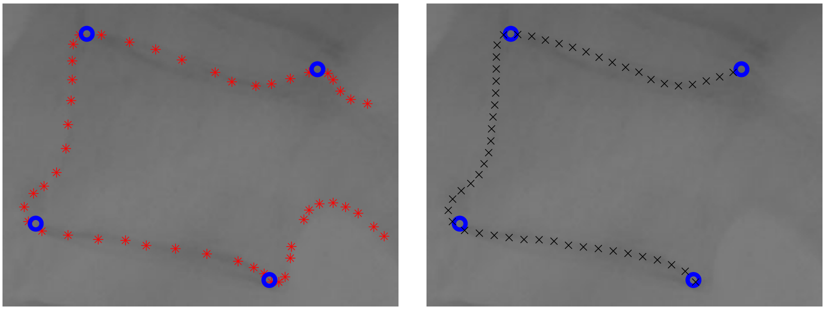


Fig. 3. Manually annotated vertebrae outline (left) and resampled outline (right)

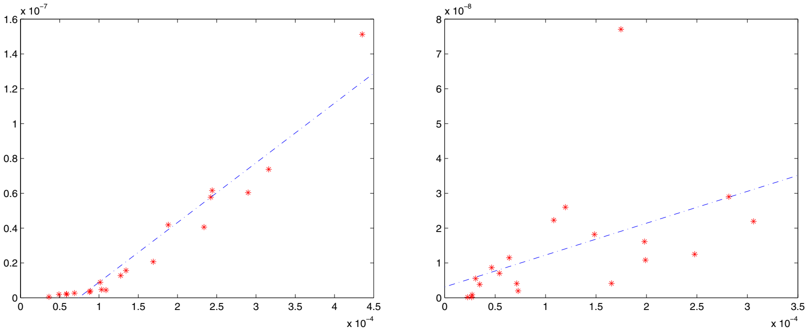


Fig. 4. Vertebrae outlines: Projection difference $\tau_{S_{\hat{v}}}$ as a function of the indicator $\tilde{\tau}_{S_{\hat{v}}}$ (left) and residual difference ρ as a function of the indicator σ (right)

and, for each selection, the entities $\tau_{S_{\hat{v}}}$, $\tilde{\tau}_{S_{\hat{v}}}$, ρ and σ are computed and plotted in Figure 4. Though we visually see correlation between the indicators and their respective associated values in the figures, not only are the correlations low, as the indicators and their values have significantly different orders of magnitude, but in reality, both the indicators and the associated values are in the order of the computation tolerance, i.e close to zero from a numerical point of view. As small indicators should imply small values, we can conclude that the indicators works as required and that, for the example of vertebra outlines, doing statistics on the manifold A_n is helpful in keeping the data consistent, i.e. the equidistance constraint satisfied, but provides little added accuracy.

4.3 Human Poses

In this experiment, we consider human poses obtained using tracking software. A consumer stereo camera[2] is placed in front of a test person, and the tracking

[2] http://www.ptgrey.com/products/bumblebee2/

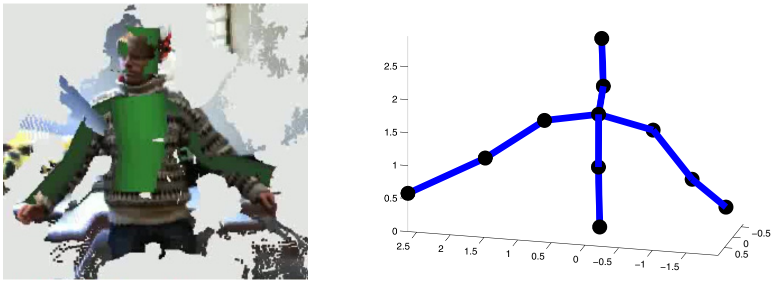


Fig. 5. Camera output superimposed with tracking result (left) and a tracked pose with 11 end-effectors marked by thick dots (right)

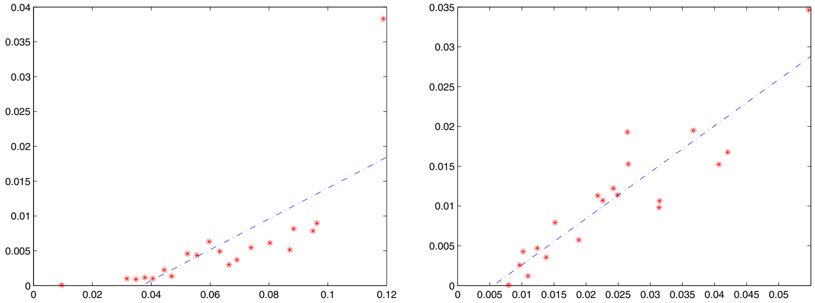


Fig. 6. Human poses: Projection difference $\tau_{S_{\hat{v}}}$ as a function of the indicator $\tilde{\tau}_{S_{\hat{v}}}$ (left) and residual difference ρ as a function of the indicator σ (right)

software described in [10] is invoked in order to track the pose of the persons upper body. The recorded poses are represented by the human body end-effectors; the end-points of each bone of the skeleton. The placement of each end-effector is given spatial coordinates so that an entire pose with k end-effectors can be considered a point in $\mathbb{R}^{3k}$. To simplify the representation, only the end-effectors of a subset of the skeleton are included, and, when two bones meet at a joint, their end-points are considered one end-effector. Figure 5 shows a human pose with 11 end-effectors marked by thick dots.

The fact that bones do not change length in short time spans gives rise to a constraint for each bone; the distance between the pair of end-effectors must be constant. We incorporate this into a pose model with b bones by restricting the allowed poses to the preimage $F^{-1}(0)$ of the map $F : \mathbb{R}^{3k} \rightarrow \mathbb{R}^{b}$ given by

$$F^{i}(x) = \|e_{i_1} - e_{i_2}\|^2 - l_i^2 \ , \tag{12}$$

where e_{i_1} and e_{i_2} denote the spatial coordinates of the end-effectors and l_i the constant length of the ith bone. In this way, the set of allowed poses constitute a $3k - b$-dimensional implicitly represented manifold.

We record 26 poses using the tracking setup, and, amongst those, we make 20 random choices of 8 poses and perform linearized PGA and exact PGA. For each experiment, $\tau_{S_{\hat{v}}}$, $\tilde{\tau}_{S_{\hat{v}}}$, ρ, and σ are computed and plotted in Figure 6. The indicators provide a good picture of the projection and residual differences, which are significantly greater than for the vertebra experiment. The indicators and the corresponding true values are now at the same order of magnitude, and the correlation between the indicators and the values they correspond to is therefore significant. The maximal increase in average squared residuals is 1.53 percent with individual squared point residuals changing up to 30.7 percent.

5 Conclusion

In this paper, we have explored the differences between exact PGA and its widely used simplification, linearized PGA. We have developed simple indicators of the loss of accuracy when using the linearized PGA instead of exact PGA. As shown on real-life examples of manifold valued datasets, these indicators provide meaningful insight into the accuracy of the linearized method. The experiments, in addition, show that linearization is in some cases a good and fast approximation, but exact PGA offers better accuracy for other applications.

We are currently working on deriving formal arguments for the correlation between σ and ρ. In the future, we plan to apply the developed indicators to the many uses of PGA, which have previously been computed using the linearized approach, to test whether exact PGA can provide significant increases in accuracy and hence more precise modeling. In order to make better decisions on whether to use linearized or exact PGA, it will be useful to find thresholds for the values of $\tilde{\tau}_{S_{\hat{v}}}$ and σ dependent on the sought for precision. Future research will hopefully lead to such thresholds.

References

1. Fletcher, P., Lu, C., Pizer, S., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE Transactions on Medical Imaging 23, 995–1005 (2004)
2. Sommer, S., Lauze, F., Nielsen, M.: The differential of the exponential map, jacobi fields, and exact principal geodesic analysis (2010) (submitted)
3. Fletcher, P.T., Joshi, S.: Riemannian geometry for the statistical analysis of diffusion tensor data. Signal Processing 87, 250–262 (2007)
4. Fletcher, P.T., Joshi, S.: Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In: Sonka, M., Kakadiaris, I.A., Kybic, J. (eds.) CVAMIA/MMBIA 2004. LNCS, vol. 3117, pp. 87–98. Springer, Heidelberg (2004)
5. Pennec, X., Fillard, P., Ayache, N.: A riemannian framework for tensor computing. Int. J. Comput. Vision 66, 41–66 (2006)
6. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22, 61–79 (1995)

7. Pennec, X., Guttmann, C., Thirion, J.: Feature-based registration of medical images: Estimation and validation of the pose accuracy. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 1107–1114. Springer, Heidelberg (1998)
8. Kendall, D.G.: Shape manifolds, procrustean metrics, and complex projective spaces. Bull. London Math. Soc. 16, 81–121 (1984)
9. Sminchisescu, C., Jepson, A.: Generative modeling for continuous Non-Linearly embedded visual inference. In: ICML, pp. 759–766 (2004)
10. Hauberg, S., Sommer, S., Pedersen, K.S.: Gaussian-like spatial priors for articulated tracking. In: Daniilidis, K. (ed.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 425–437. Springer, Heidelberg (2010)
11. Karcher, H.: Riemannian center of mass and mollifier smoothing. Communications on Pure and Applied Mathematics 30, 509–541 (1977)
12. Pennec, X.: Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. J. Math. Imaging Vis. 25, 127–154 (2006)
13. Huckemann, S., Hotz, T., Munk, A.: Intrinsic shape analysis: Geodesic PCA for riemannian manifolds modulo isometric lie group actions. Statistica Sinica 20, 1–100 (2010)
14. do Carmo, M.P.: Riemannian geometry. Mathematics: Theory & Applications. Birkhauser, Boston (1992)
15. Lee, J.M.: Riemannian manifolds. Graduate Texts in Mathematics, vol. 176. Springer, New York (1997); An introduction to curvature
16. Dedieu, J., Nowicki, D.: Symplectic methods for the approximation of the exponential map and the newton iteration on riemannian submanifolds. Journal of Complexity 21, 487–501 (2005)
17. Noakes, L.: A global algorithm for geodesics. Journal of the Australian Mathematical Society 64, 37–50 (1998)
18. Klassen, E., Srivastava, A.: Geodesics between 3D closed curves using Path-Straightening. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 95–106. Springer, Heidelberg (2006)
19. Schmidt, F., Clausen, M., Cremers, D.: Shape matching by variational computation of geodesics on a manifold. In: Pattern Recognition, pp. 142–151. Springer, Berlin (2006)
20. Sommer, S., Tatu, A., Chen, C., Jørgensen, D., de Bruijne, M., Loog, M., Nielsen, M., Lauze, F.: Bicycle chain shape models. In: MMBIA/CVPR 2009, pp. 157–163 (2009)
21. Huckemann, S., Ziezold, H.: Principal component analysis for riemannian manifolds, with an application to triangular shape spaces. Advances in Applied Probability 38, 299–319 (2006)
22. Fletcher, P., Lu, C., Joshi, S.: Statistics of shape via principal geodesic analysis on lie groups. In: CVPR 2003, vol. 1, p. I-95 – I-101 (2003)
23. Wu, J., Smith, W., Hancock, E.: Weighted principal geodesic analysis for facial gender classification. In: Progress in Pattern Recognition, Image Analysis and Applications, pp. 331–339. Springer, Berlin (2008)
24. Said, S., Courty, N., Bihan, N.L., Sangwine, S.: Exact principal geodesic analysis for data on so(3). In: EUSIPCO 2007 (2007)

Stacked Hierarchical Labeling

Daniel Munoz, J. Andrew Bagnell, and Martial Hebert

The Robotics Institute
Carnegie Mellon University
{dmunoz,dbagnell,hebert}@ri.cmu.edu

Abstract. In this work we propose a hierarchical approach for labeling semantic objects and regions in scenes. Our approach is reminiscent of early vision literature in that we use a decomposition of the image in order to encode relational and spatial information. In contrast to much existing work on structured prediction for scene understanding, we bypass a global probabilistic model and instead directly train a hierarchical inference *procedure* inspired by the message passing mechanics of some approximate inference procedures in graphical models. This approach mitigates both the theoretical and empirical difficulties of learning probabilistic models when exact inference is intractable. In particular, we draw from recent work in machine learning and break the complex inference process into a hierarchical series of simple machine learning subproblems. Each subproblem in the hierarchy is designed to capture the image and contextual statistics in the scene. This hierarchy spans coarse-to-fine regions and explicitly models the mixtures of semantic labels that may be present due to imperfect segmentation. To avoid cascading of errors and overfitting, we train the learning problems in sequence to ensure robustness to likely errors earlier in the inference sequence and leverage the stacking approach developed by Cohen *et al.*

1 Introduction

The challenging problem of segmenting and labeling an image into semantically coherent regions can be naturally modeled as a hierarchical process to interpret the scene [23]. Typically, a graphical model is used where each node represents the labels present in some region of the image with dependencies that tie together multiple regions [3,6]. The nodes at the bottom of the hierarchy provide low-level discriminative information, while nodes higher up resolve ambiguities using global information. While these representations seem intuitive, learning the optimal components of the model is practically intractable due to complex dependencies. Furthermore, even with simplified representations exact inference remains intractable [13] and prohibits learning these models. Although training with exact inference is infeasible, a natural alternative is to use *approximate* inference. However, as we discuss in the next section, these approximations during learning can lead to undesirable behavior [16]. Therefore, we move away from a representation for which training is intractable and toward an approach which relies on effective components that are simple to train.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 57–70, 2010.

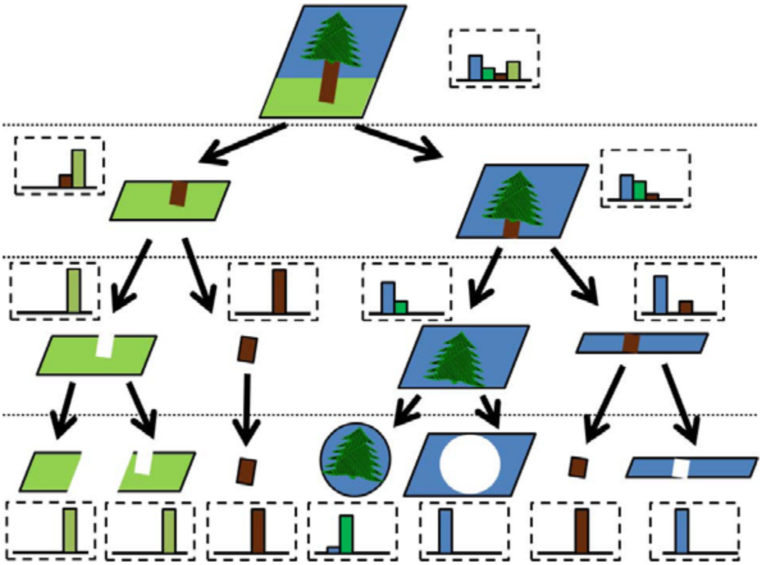


Fig. 1. A synthetic example of our hierarchical labeling process. Given an image and its hierarchical decomposition of regions, we sequentially predict the proportion of labels present (drawn in the dashed boxes) using image features and previous predictions.

In this work, we model low-level information combined with higher-order reasoning using a hierarchical representation. Our approach is similar to previous structured models with the key difference that we no longer attempt the intractable task of finding the mode of the joint posterior distribution using a generic approximate inference algorithm. Instead we simplify the problem into a series of subproblems that are specifically trained to perform well for our task. That is, we train these subproblems to model the relations present in the image so that the overall prediction is correct. One major advantage of this approach is that test-time structured prediction is simply a sequence of predictions. Our contribution is a novel hierarchical algorithm for labeling semantic regions.

An idealized example of our approach is depicted in Fig. 1. We represent the inference process as a series of predictions along the hierarchy from coarse to fine. Given an image, we first create a hierarchy of regions that range from very large regions in the image (including the image itself as one region at the top) down to small regions (*e.g.*, superpixels) at the bottom. We do not rely on each region to contain one label; instead we explicitly model the label proportions in each region. Starting with the entire image, we train a classifier[1] to predict the proportions of labels in the image. As we further discuss in Sect. 3, these predictions are passed to the child level and are used to train another classifier

[1] In this work, we refer to a classifier as an algorithm that predicts a distribution over labels, instead of a single label.

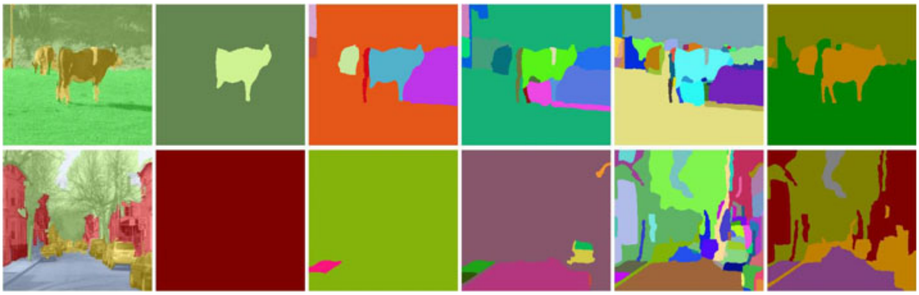


Fig. 2. Hierarchy analysis for two images. From left to right: input image with ground truth overlaid, the segmentation maps for L2 (second level), L4, L6, L8, and most likely label for each region in L8.

over the child subregions. The procedure is repeated until the leaves are reached. Since we model label proportions over regions: we are robust to imperfect segmentation, we can use features defined over large regions, and we do not make hard commitments during inference.

Figure 2 illustrates four levels from the hierarchy on two images from the Stanford Background Dataset (SBD) [8]. Ideally the leaves in the hierarchy are regions that contain only one label, but as Fig. 2 also illustrates, this assumption is not always true, especially for more complex scenes. With our hierarchical approach, we demonstrate state-of-the-art performance on SBD and MSRC-21 [26] with the added benefit of drastically simpler computations over global methods.

2 Background

2.1 Motivation

Random field models in vision have proven to be an effective tool and are also attractive due to their convex nature (assuming no latent states) [19]. Furthermore, although exact inference is NP-hard over these models, there has been much recent progress towards efficient *approximate* inference techniques [12,14]. However, correctly optimizing these convex models requires *exact* inference during learning. Unfortunately, when exact inference cannot be performed, converging to the optimum is no longer guaranteed [16]. For example, Kulesza and Pereira [16] demonstrate a case where learning with bounded approximate inference can prevent the learning procedure from ever reaching a feasible zero empirical risk solution. Similarly in another example, they show that learning with loopy belief propagation can diverge.

As we are forced to use approximate inference during learning, the learned model (*e.g.*, parameters) is tightly tied to the chosen inference procedure in both theory [29] and in practice [17]. However, learning the best model for the chosen inference procedure is often still difficult. Due to this fundamental limitation when training with approximate inference techniques, we move away from the global probabilistic interpretation used in hierarchical random field formulations, such as [18]. Instead, in a manner inspired by inference procedures over graphical models, we propose a novel method using iterative classifiers that are trained to encode interactions between levels but correspond to no explicit joint probability distribution.

2.2 Related Work

Our hierarchical formulation resembles early directed graphical models from Bouman and Shapiro [3] and Feng *et al.* [6] for scene analysis. Whereas these approaches rely on tree-based interactions to enable tractable learning, we no longer train a graphical model and are not restricted in the types of contextual cues that we can use. Instead we focus on maximizing what we ultimately care about: predicting correct labelings. This idea is analogous to the difficult and non-convex problem of maximizing the marginals [11]. The notion of training the inference algorithm to make correct predictions is also similar to Barbu [2] for image denoising, in which a model is trained knowing that an inaccurate, but fast, inference algorithm will be used. In our approach we break up the complex structured prediction problem into a series of simpler classification problems, inspired by recent works in machine learning focused on sequence prediction [4,5]. In the vision setting, this notion of a series of classification problems is similar to Auto-context [27], in which pixel classifiers are trained in series using the previous classifier's predictions with pairwise information to model contextual cues. In our work, we go beyond typical site-wise representations that require entities to contain one label. Because we model label proportions, we can use features defined over large regions to better represent the context, rather than an aggregation of site-wise labels. Furthermore, the hierarchy provides spatial support context between levels and naturally propagates long-range interactions that may be hard to capture with pairwise interactions. We build on the forward sequential learning approach used and analyzed in [28,10,25] to prevent cascading errors and leverage the sequential stacking idea to minimize cascaded overfitting [30,4,15].

3 Stacked Hierarchical Labeling

3.1 Overview

Given an image and its hierarchical region representation, we train a series of classifiers, from coarse to fine, to predict the label proportions in each region in the level. After a level has been trained, the predicted labels are passed to the child regions to be used as features that model contextual relationships. Figure 3 illustrates (on test data) how the probabilities for the respective labels increase and become more precise along three levels in the lower half of the hierarchy. Our approach is robust to the quality of the segmentation at each level as we explicitly model that regions may contain multiple labels. Therefore, depending on how

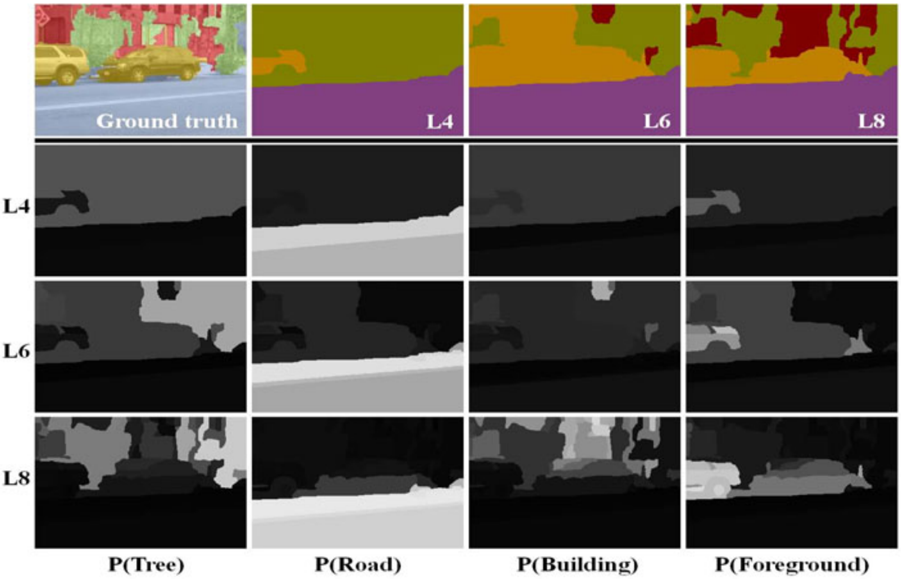


Fig. 3. Refinement in label predictions down the hierarchy. Row 1: Test image (with ground truth overlaid) and predictions for three levels in the hierarchy. Rows 2-4: The respective level's label probability maps, where white indicates high probability.

the hierarchy is constructed, our algorithm will learn how regions for different labels are split between levels. We create the hierarchy using the technique from Arbelaez *et al.* [1,22].

The next subsections describe each component of the training procedure. We first introduce the notations and describe the basic classifier used at each level (Sect. 3.2). We then describe how predictions from a parent region are incorporated as features (Sect. 3.3) and how classifiers are trained across levels in the hierarchy to finalize the procedure (Sect. 3.4).

3.2 Modeling Heterogeneous Regions

We denote by $\mathcal{K}$ the set of possible labels, L the number of levels in the hierarchy, $\mathcal{T}$ the set of training images, $\mathcal{I}_I$ the image data for image I, $\mathcal{R}_I$ its set of regions in the hierarchy, and $\mathcal{R}_{I,\ell}$ the set of regions at level ℓ. For each region $r \in \mathcal{R}_I$, we define Y_r to be the random variable that represents the label of the region. For each level ℓ, we train a probabilistic classifier to match the empirical label distribution of $r \in \mathcal{R}_{I,\ell}$ across all training images. For its simplicity, we use a generalized maximum entropy classifier q_{ϕ_ℓ}, where $\phi_\ell : \mathbb{R}^d \rightarrow \mathbb{R}$ is a function that defines the distribution:

$$q_{\phi_\ell}(Y_r = a|\mathcal{I}_I) = \frac{\exp(\phi_\ell(f_I(r,a)))}{\sum_{k\in\mathcal{K}} \exp(\phi_\ell(f_I(r,a)))}, \tag{1}$$

Algorithm 1. `train_maxent`

```
Inputs: Dataset of region features with true distributions $\mathcal{D} = \{(f_I(r,k), p_{I,r,k})\}_{I,r,k}$
where $p_{I,r,k} = p_I(Y_r = k)$, Step size $\alpha_t$, Number of iterations $T$.
$\phi = 0$
for $t = 1 \dots T$ do
    $\mathcal{A} = \emptyset$
    for $(f_I(r,k), p_{I,r,k}) \in \mathcal{D}$ do
        if $\beta_I(r,k) \neq 0$ then
            $\mathcal{A} \leftarrow \mathcal{A} \cup \{(f_I(r,k),\ \beta_I(r,k))\}$
        end if
    end for
    $h_t$ = train_multi_class_regressor($\mathcal{A}$)
    $\phi \leftarrow \phi + \alpha_t h_t$ // (or, line-search instead of constant $\alpha_t$)
end for
Return: MaxEnt classifier $\phi$
```

and $f_I : \mathcal{R}_I \times \mathcal{L} \rightarrow \mathbb{R}^d$ are the feature functions that extract (label-specific) features describing the region from image data $\mathcal{I}_I$, such a texture and color (see Appendix). In the following subsection, we discuss how predictions from parent regions are appended to this vector to model context.

At each level, we match the distributions by minimizing the cross entropy of the empirical label distributions p and the classifier q, which reduces to:

$$\phi_\ell^* = \arg\max_{\phi_\ell} \sum_{I \in \mathcal{T}} \sum_{r \in \mathcal{R}_{I,\ell}} \sum_{k \in \mathcal{K}} p_I(Y_r = k) \log q_{\phi_\ell}(Y_r = k | \mathcal{I}_I). \tag{2}$$

This is a standard maximum log-likelihood estimation where the samples are weighted by $p_I(Y_r = k)$, *i.e.*, the number of pixels labeled k in the region divided by its area. The optimization may be performed through standard convex optimization (*e.g.*, gradient ascent) and provides competitive performance in our experiments; however, we found using a non-linear model further improves performance. We train a non-linear model in a boosting manner through Euclidean functional gradient ascent [24]; the following describes the optimization but it is not specific to our hierarchical procedure.

The functional gradient of the inner term in (2) is $\beta_I(r,k)\delta_{f_I(r,k)}$, where

$$\beta_I(r,k) = p_I(Y_r = k) - q_{\phi_\ell}(Y_r = k | \mathcal{I}_I), \tag{3}$$

and δ_x is the Dirac delta function centered at feature value x. As a form of boosting, we train a new function h to match the functional gradient residuals and add it to ϕ. The residuals indicate how to update the function when evaluated at the respective feature locations so that the predicted and ground truth distributions match. We repeat this procedure until convergence and then return $\phi = \sum_t \alpha_t h_t$, where α_t is the step size. We refer to [24] for more details. The training algorithm is given in Algorithm 1. In our experiments we train a separate Random Forest for each class as the multi-class regressor h.

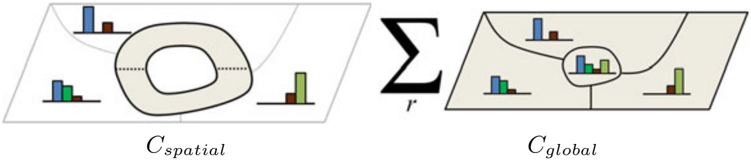


Fig. 4. Illustration of the context features described in Sect. 3.3. Gray indicates the pixels that are being used to compute the feature.

Partial Labelings. Ideally all pixels in the training set are assigned a label; however, most datasets contain many images with unlabeled pixels (such as MSRC-21). We assume that if a class is labeled in the image, then all instances of that class are labeled in that image. In the case a region r is partially labeled, we propose for the classifier to match the proportions of the classes actually present ($\hat{\mathcal{K}}_r$) and to not penalize the predictions made for the classes not present ($\bar{\mathcal{K}}_r$), as the unlabeled pixels may actually contain classes from $\mathcal{K}$. We do this by treating (2) as a negative loss function and by only penalizing the terms with labels in $\hat{\mathcal{K}}_r$ and ignoring the remaining labels, *i.e.*, setting $p_I(Y_r = a) = 0, \forall a \in \bar{\mathcal{K}}_r$ discards losses over the labels not present.

3.3 Context Features

In addition to the image features computed at each level, we need to define the information that is passed from one classifier to the next. It is this information that ties together the individual classifiers trained at each level to yield the global image interpretation generator. Intuitively, using the label distribution predicted by the parent's classifier will make training the child's level distribution predictor an easier problem. At each level, we receive probabilities from the parent level regions. Since this information is of variable length per image, specifically $|\mathcal{R}_{I,\ell-1}| \times |\mathcal{K}|$, we need to summarize it into a fixed-length vector that can be used as input to a generic classifier. For each region in $\mathcal{R}_\ell$, we define three types of contextual features that are computed using the predictions from the regions in $\mathcal{R}_{\ell-1}$. For the first, each region simply uses its parent region's label predictions ($C_{parent} \in \mathbb{R}^{|\mathcal{K}|}$). The next two are illustrated in Fig. 4. The second is the weighted average of the neighboring region's probabilities. The weights are the areas of the region's dilated mask that overlaps with the respective neighbors. In order to describe spatial layout, we compute the averages above and below the region separately ($C_{spatial} \in \mathbb{R}^{2|\mathcal{K}|}$). The third is the weighted average (by size) of the probabilities across all regions in the parent level ($C_{global} \in \mathbb{R}^{|\mathcal{K}|}$); this feature is duplicated for all regions in the current level. These context features are then appended to the respective region's f_I image-based feature vector.

3.4 Hierarchical Stacking

The MaxEnt classifier described in Sect. 3.2 is the basic component used at each level in the hierarchy. Collectively training the classifiers is prone to two problems

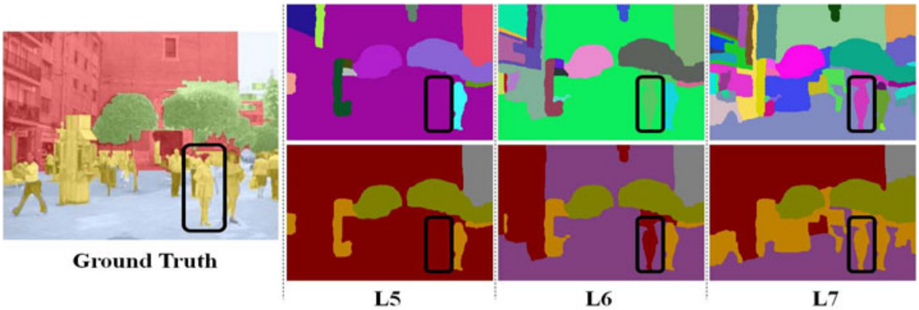


Fig. 5. Example of test-time error recovery. Left: test image with ground truth overlaid. Top: segmentation maps for L5, L6, L7. Bottom: most likely label per region.

of cascading errors. *First*, if we train each level's classifier independently using the parent regions' ground truth, we will have cascading errors at test-time due each classifier being trained with perfect contextual information. Therefore, we have to train the hierarchical procedure in the same way it is executed at test-time: in sequence, using the predictions from the previous level. After predicting the label distributions for each region in a level, we pass this information to the child level, similar to what is done during inference over a graphical model. Similar to other hierarchical methods [18], we pass these predicted per-class probabilities for each region as a vector from which the children construct the context features as described above. Ideally, high levels in the hierarchy can represent the type of environment which "primes" the lower levels with a smaller set of labels to consider. *Second*, now using predictions from the same data used for training is prone to a cascade of errors due to overfitting as subsequent levels will rely heavily on still optimistically correct context. While parent predictions are important, we also want to learn how to recover from mistakes that will be made at test time by trading off between the parent probabilities and image features. To achieve this robust training, we use the idea of stacking [30,4] when training the classifier. Figure 7 illustrates how stacking addresses the overfitting behavior on the MSRC-21 dataset.

Stacking trains a sequence of classifiers where the outputs of one classifier are treated as additional features and are used to train another classifier. In order to avoid overfitting, the outputs are predicted on data that was not used to train the classifier. Obtaining held-out predictions is achieved in a manner similar to cross-validation where the training data is split into multiple subsets that multiple classifiers train on. Because the predictions are made on unseen data, the procedure simulates the test-time behavior and ideally learns how to correct earlier mistakes. An example of this correcting behavior during test-time is illustrated in Fig. 5. In L5, the person is part of a very large region for which label *building* is most confident. In L6, the person is segmented out from its large parent region; however, the most likely label for this region incorrectly follows from the parent's label (*building*). In L7, the region recovers from this error and is correctly labeled *foreground*.

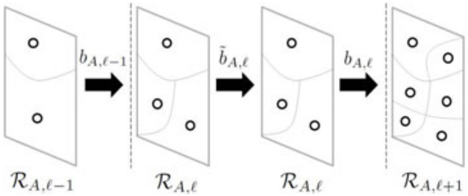


Fig. 6. Predictions between levels during learning/inference for image A

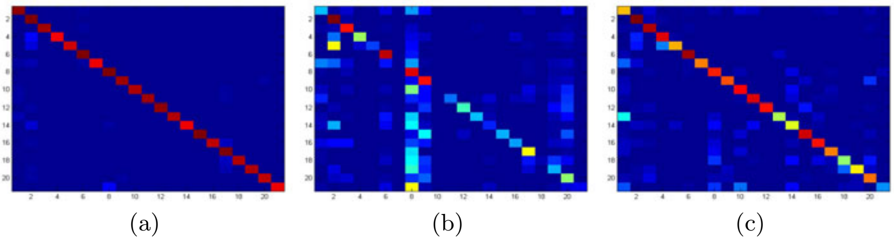
We now describe the stacking procedure in detail (Fig. 6). For each image $I \in \mathcal{T}$, we receive the predictions for each parent region in $\mathcal{R}_{I,\ell-1}$; we denote this $|\mathcal{R}_{I,\ell-1}| \times |\mathcal{K}|$ set of predictions per image I as $b_{I,\ell-1}$. Using $b_{I,\ell-1}$, we compute the context features (Sect. 3.3) for each region in $\mathcal{R}_{I,\ell}$ and append them to its image features f_I. We then generate held-out predictions for all regions at level ℓ (across all training images) by training temporary classifiers on subsets of regions and predicting on the held-out regions. That is, to generate the predictions for regions $\mathcal{R}_{A,\ell}$ in image A, we train a classifier $\tilde{\phi}_{A,\ell}$ over the regions $\cup_{I\in\mathcal{T}\setminus A}\mathcal{R}_{I,\ell}$ and then classify the held-out regions $\mathcal{R}_{A,\ell}$ to generate predictions $\tilde{b}_{A,\ell}$. This process is repeated $|\mathcal{T}|$ times to generate predictions across all images[2]. This stacking procedure is done solely during training to generate predictions to compute the context features. Therefore, we train a final classifier $\tilde{\phi}_\ell$ across all regions at level ℓ to be used at test time. The main idea is that the temporary classifiers simulate the behavior $\tilde{\phi}_\ell$ will have on the unseen test data. Since these classifiers use predictions from the parent level, we refer to them as inter-level classifiers.

One potential problem occurs when a large region at level $\ell - 1$ is split into many small regions at level ℓ. In that case, the context feature $C_{spatial}$ for most of the offspring regions is uninformative because it uses the predictions only from the one parent region without capturing any context. To address this problem, we apply a second round of stacking. In that second round, a new classifier is learned and new predictions $b_{I,\ell}$ are generated by using the same procedure as described above, with the one critical difference that $C_{spatial}$ is computed by using the predictions at level ℓ generated from the classifier learned in the first round, $\tilde{b}_{I,\ell}$, rather than by using the predictions from the previous level, $b_{I,\ell-1}$. In addition, we also append each region's respective prediction from the first round, $\tilde{b}_{I,\ell}$. The resulting set of predictions $b_{I,\ell}$ from this intra-level stacking are then passed to the next level and classifier ϕ_ℓ is saved for test-time. The two-stage process is then repeated for the next level. In practice, we do not do the second stage at the top level.

3.5 Inference

Given a test image I and its hierarchy of regions, inference proceeds in the same cascading manner. At level ℓ, we receive the parent level probabilities $b_{I,\ell-1}$ to

[2] In practice, we hold out 10% of the training images instead of just one.



(a) (b) (c)

Fig. 7. Confusion matrices on MSRC-21 dataset. Performance on training set without stacking (a), and performance on testing set without (b) and with (c) stacking.

create the context features and then use the inter-level classifier $\tilde{\phi}_\ell$ to predict $\tilde{b}_{I,\ell}$. Next, we use $b_{I,\ell-1}$ and $\tilde{b}_{I,\ell}$ to create the same context features from the second stage and then predict $b_{I,\ell}$ with the intra-level classifier ϕ_ℓ. Therefore, performing inference over the hierarchy requires $2L-1$ predictions (since there are no intra-level predictions for the first level).

4 Experiments

We evaluate our algorithm on the MSRC-21 and Stanford Background datasets and demonstrate that we can achieve high performance predictions as with other structured models, even though we never explicitly model global configurations. In both experiments we use the same set of standard image features, mostly computed from the STAIR Vision Library [9], and the same set of learning parameters used to train the hierarchy; see the Appendix for specific details.

4.1 MSRC-21

The MSRC-21 dataset [26] contains a variety of outdoor environments with 21 possible classes; we use the standard evaluation split from [26]. Although not ideal for our hierarchical regions, we use the image-based region features from the flat CRF model of [7] and demonstrate favorable quantitative performance compared to this and other similar recent work. As illustrated in Table 1, we compare with related models that are structured [31,20], use hierarchical regions [21], and sequentially trained (over sites) [27]; "Hier." is our hierarchical approach and "Leaf" is a site-wise classifier trained only over the leaf regions without any context. Although the hierarchical CRF model of [20] demonstrates superior performance, it should be noted that their pixel-wise classifier can obtain an overall accuracy of 80%, which suggests the use of much more discriminative features. In Fig. 8, we quantify the hierarchy's refinement in labeling by plotting, at each level, the accuracies if we assign the regions' pixels their most probable label.

Table 1. Performances on the MSRC-21 dataset. *Overall* is the total number of pixels correct and *Average* is the mean across the columns. *Averaged over 5 different splits.

	Overall	*Average*	Building	Grass	Tree	Cow	Sheep	Sky	Airplane	Water	Face	Car	Bicycle	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
[7]*	77	64	72	95	81	66	71	93	74	70	70	69	72	68	55	23	83	40	77	60	50	50	14
[31]	75	65	77	93	70	58	64	92	57	70	61	69	67	74	70	47	80	53	73	53	56	47	40
[21]	–	67	30	71	69	68	64	84	88	58	77	82	91	90	82	34	93	74	31	56	54	54	49
[27]	75	69	69	96	87	78	80	95	83	67	84	70	79	47	61	30	80	45	78	68	52	67	27
[20]	86	75	80	96	86	74	87	99	74	87	86	87	82	97	95	30	86	31	95	51	69	66	9
Leaf	74	60	72	96	85	74	70	91	63	58	65	59	69	58	32	22	84	25	83	55	33	54	4
Hier.	78	71	63	93	88	84	65	89	69	78	74	81	84	80	51	55	84	80	69	47	59	71	24

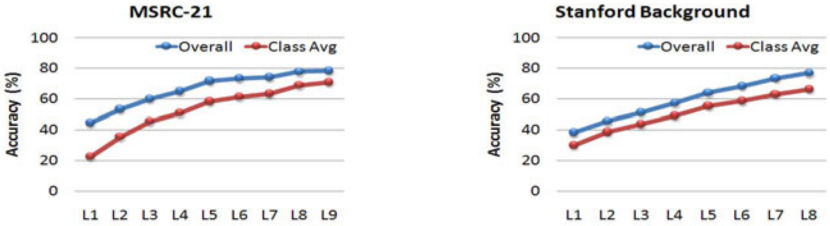


Fig. 8. Accuracies when assigning regions, at each level, their most probable label

4.2 Stanford Background Dataset

We also evaluate our approach on the recent dataset from [8]. This dataset contains densely labeled images containing eight semantic labels. All results were averaged over five random trials, using the splits described in [8].

Table 2 contains the performances of two structured models and our hierarchical approach. We achieve comparable performance with the global energy model used in [8] while never explicitly modeling the global configurations. Holding segmentation and *image* feature extraction time constant, our hierarchical inference typically takes 12 s/image (10 s of which is spent on computing the *contextual* features), whereas the global energy approach can widely vary from 30 s to 10 min to converge. In Fig. 8, we see a similar label refinement.

4.3 Confident Predictions

Another benefit of our approach over MAP inference techniques (*e.g.*, graph-cuts) is that we never make hard decisions and always predict a distribution of labels. Therefore, when eventually assigning a label to a region, we can extract a notion of confidence in the labeling. We define a labeling as *confident* when the most likely label is 0.2 higher than the runner-up, and otherwise *uncertain*. For example, in Fig. 9, the cars are *confident* in the labeling, but the trees in front of the building are *uncertain*. On MSRC-21, our *confident* predictions constitute 79% of the data and achieve an overall accuracy of 89%, while the *uncertain*

Table 2. Performances on the Stanford Background Dataset

	Overall	*Average*	Sky	Tree	Road	Grass	Water	Bldg.	Mtn.	Fgnd.
[8] Pixel CRF	74.3	66.6	93.9	67.1	90.3	83.3	55.4	71.4	9.3	62.2
[8] Region Energy	76.4	65.5	92.6	61.4	89.6	82.4	47.9	82.4	13.8	53.7
Leaf	72.8	58.0	89.7	58.3	85.8	69.8	15.8	78.1	1.5	64.9
Hierarchy	76.9	66.2	91.6	66.3	86.7	83.0	59.8	78.4	5.0	63.5

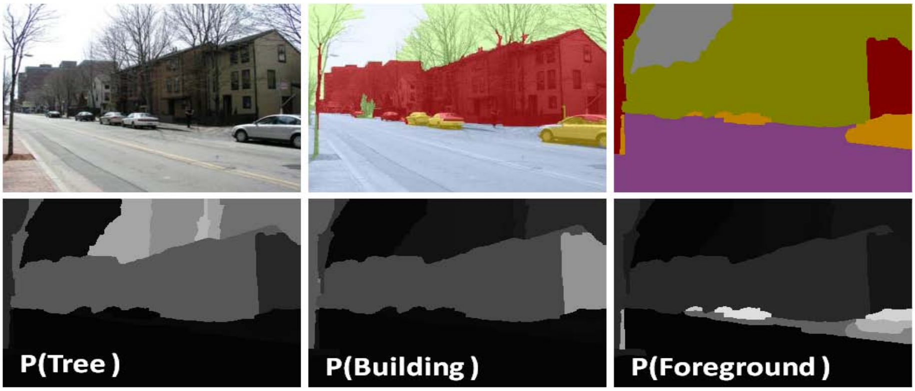


Fig. 9. The ambiguity in ground truth label (top, middle) is correctly modeled in our predictions (bottom row), resulting in a labeling for the building that is *uncertain*

accuracy is 37%. On SBD, our *confident* predictions constitute 87% of the data and achieve an overall accuracy of 82%, while the *uncertain* accuracy is 40%. These numbers indicate that we make most errors when the labeling is *uncertain*.

5 Conclusion

We propose an alternative to the graphical model formulation for structured prediction in computer vision. Our approach is based on training a sequence of simple subproblems that are designed to use context, bypassing the difficulties of training typical structured models. Specifically, we designed an algorithm to train these subproblems in a hierarchical procedure that a) captures the context over large regions b) explicitly models that regions contain mixed labels and c) is trained to follow the same procedure during test-time. Our experiments demonstrate this simple approach is able to capture context and make high performance predictions without a probabilistic model over global configurations.

Acknowledgments. D. Munoz is supported by a QinetiQ North America Robotics Fellowship. This work is supported in part by the ONR MURI grant

N00014-09-1-1052, Reasoning in Reduced Information Spaces. We thank A. Grubb and S. Ross for valuable discussions and B. Becker and A. Collet for CPU resources.

References

1. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR (2009)
2. Barbu, A.: Training an active random field for real-time image denoising. IEEE Trans. on Image Processing 18(11) (2009)
3. Bouman, C.A., Shapiro, M.: A multiscale random field model for bayesian image segmentation. IEEE Trans. on Image Processing 3(2) (1994)
4. Cohen, W.W., Carvalho, V.R.: Stacked sequential learning. In: IJCAI (2005)
5. Daume III, H., Langford, J., Marcu, D.: Search-based structured prediction. Machine Learning Journal 75(3) (2009)
6. Feng, X., Williams, C.K.I., Felderhof, S.N.: Combining belief networks and neural networks for scene segmentation. IEEE T-PAMI 24(4) (2002)
7. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. IJCV 80(3) (2008)
8. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
9. Gould, S., Russakovsky, O., Goodfellow, I., Baumstarck, P., Ng, A.Y., Koller, D.: The stair vision library, v2.3 (2009), http://ai.stanford.edu/~sgould/svl
10. Heitz, G., Gould, S., Saxena, A., Koller, D.: Cascaded classification models: Combining models for holistic scene understanding. In: NIPS (2008)
11. Kakade, S., Teh, Y.W., Roweis, S.: An alternate objective function for markovian fields. In: ICML (2002)
12. Kohli, P., Ladicky, L., Torr, P.H.: Robust higher order potentials for enforcing label consistency. IJCV 82(3) (2009)
13. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? IEEE T-PAMI 26(2) (2004)
14. Komodakis, N., Paragios, N., Tziritas, G.: Mrf energy minimization and beyond via dual decomposition. IEEE T-PAMI (in press)
15. Kou, Z., Cohen, W.W.: Stacked graphical models for efficient inference in markov random fields. In: SDM (2007)
16. Kulesza, A., Pereira, F.: Structured learning with approximate inference. In: NIPS (2007)
17. Kumar, S., August, J., Hebert, M.: Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 153–168. Springer, Heidelberg (2005)
18. Kumar, S., Hebert, M.: A hierarchical field framework for unified context-based classification. In: ICCV (2005)
19. Kumar, S., Hebert, M.: Discriminative random fields. IJCV 68(2) (2006)
20. Ladicky, L., Russell, C., Kohli, P., Torr, P.: Associative hierarchical crfs for object class image segmentation. In: ICCV (2009)
21. Lim, J.J., Arbelaez, P., Gu, C., Malik, J.: Context by region ancestry. In: ICCV (2009)

22. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR (2008)
23. Ohta, Y., Kanade, T., Sakai, T.: An analysis system for scenes containing objects with substructures. In: Int'l. Joint Conference on Pattern Recognitions (1978)
24. Ratliff, N., Silver, D., Bagnell, J.A.: Learning to search: Functional gradient techniques for imitation learning. Autonomous Robots 27(1) (2009)
25. Ross, S., Bagnell, J.A.: Efficient reductions for imitation learning. In: AIStats (2010)
26. 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 81(1) (2009)
27. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3d brain image segmentation. T-PAMI 18(11) (2009)
28. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV 57(2) (2004)
29. Wainwright, M.J.: Estimating the "wrong" graphical model: Benefits in the computation-limited setting. JMLR 7(11) (2006)
30. Wolpert, D.H.: Stacked generalization. Neural Networks 5(2) (1992)
31. Zhang, L., Ji, Q.: Image segmentation with a unified graphical model. T-PAMI 32(8) (2010)

A Image Features

For the top-level in the hierarchy, we use Gist[3] computed from 64x64 rescaled images at 2 scales with 8 and 4 orientations, Pyramid Histogram of Oriented Gradients[4] with 2 levels, 8 histogram bins and 4 orientations, and a color histogram over $CIELab$ colorspace with 10 bins over L and 20 bins over a and b along with the mean and std. per channel. For the remaining levels in the hierarchy, we primarily use the region appearance features from [7,9]. These features consist of filters, color and bounding box statistics, location, and the weighted average of neighboring regions' features. In addition, we also count the number of vertices on the convex hull enclosing each region and use a hierarchy-based descriptor to model relative relocation. This descriptor consists of the orientation ($[-\pi, \pi]$) and length of the vector extending from the centroid of the parent to the child, normalized by the diagonal of the parent's bounding box.

B Hierarchy

The hierarchy is created by thresholding the scale value from 256 at an interval of -30. During functional gradient boosting, the step size is $\alpha_t = \frac{1.5}{\sqrt{t}}$ for each level while we increase the number of iterations T at each level to handle the increasing amount of data as regions were split: 10, 12, 15, 17, 20, 20, 25, 30. The Random Forest[5] regressor consisted of 10 trees and each tree required at least 15 samples to split a node. We found the entire process is resilient to changes in these parameters.

[3] http://lear.inrialpes.fr/software

[4] http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html

[5] http://code.google.com/p/randomforest-matlab/

Fully Isotropic Fast Marching Methods on Cartesian Grids

Vikram Appia and Anthony Yezzi

Georgia Institute of Technology, GA, USA

Abstract. The existing Fast Marching methods which are used to solve the Eikonal equation use a locally continuous model to estimate the accumulated cost, but a discontinuous (discretized) model for the traveling cost around each grid point. Because the accumulated cost and the traveling (local) cost are treated differently, the estimate of the accumulated cost at any point will vary based on the direction of the arriving front. Instead we propose to estimate the traveling cost at each grid point based on a locally continuous model, where we will interpolate the traveling cost along the direction of the propagating front. We further choose an interpolation scheme that is not biased by the direction of the front. Thus making the fast marching process truly isotropic. We show the significance of removing the directional bias in the computation of the cost in certain applications of fast marching method. We also compare the accuracy and computation times of our proposed methods with the existing state of the art fast marching techniques to demonstrate the superiority of our method.

Keywords: Fast Marching Methods, Isotropic Fast Marching, Segmentation, Tracking, FMM, Eikonal Equation, minimal cost path.

1 Introduction

A large number of computer vision applications such as segmentation, tracking, optimal path planning *etc*. use the minimal cost path approach. The Fast Marching Method which is widely used to solve the minimal path problem was first introduced by Sethian [1,10] and Tsitsiklis [11]. Cohen and Kimmel [4,5] later noticed that the minimal cost problem satisfies the Eikonal equation,

$$\|\nabla u\| = \tau. \tag{1}$$

For the Eikonal equation 1 defined on a Cartesian Grid, $\tau(x)$ would be the traveling cost at a given grid point and $u(x)$, the accumulated cost. Since we solve the Eikonal equation numerically on Cartesian Grids, it is impossible to find the exact solution. Some modifications have been suggested in [6,7] to improve the accuracy of the Fast Marching method. Authors in [6,8,9,11] also suggest using an 8-connected neighbor scheme to improve accuracy. All these techniques use a locally continuous model to estimate the accumulated cost, but assume the traveling cost to be constant (discretized) around each grid point. Only [6] interpolates τ by shifting it to the center of the grid with a nearest neighbor interpolation, but it still assumes a discretized shifted grid for τ. In this paper we propose to use a locally continuous model to estimate τ as well.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 71–83, 2010.

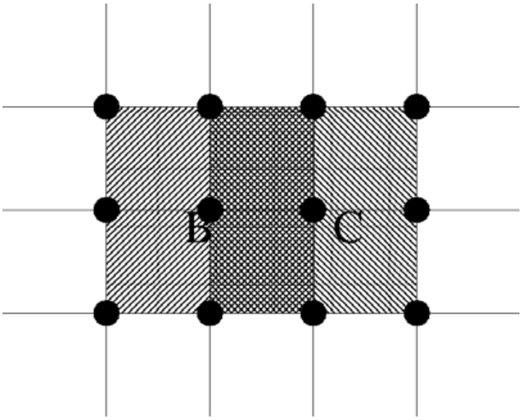


Fig. 1. Overlap in the influence areas of τ_B and τ_C

For the geometry shown in Figure 1, the Fast Marching Method uses linear approximation to compute the accumulated cost at the point C, but it uses a constant traveling cost τ_C for each of the four grid cells containing the point C. The influence area of the cost function given at a grid point will include all the four quadrants around it. Thus, there is an overlap in the areas of influence of the grid points B and C. This means the value of u_C will vary depending on the direction from which the front is arriving. Ideally, for isotropic fast marching, the accumulated cost should be independent of the direction of the arriving front. For the image shown in Figure 2, we use the traveling cost, $\tau(x) = I(x)$, where I(x) is the intensity at each pixel. The accumulated cost in traveling from point A to B should be equal to the cost in traveling from B to A. But, due to the dependence on the direction of marching, there will be a difference in the accumulated costs. Figure 2 compares the minimal path obtained using back propagation from end point B to the source point A with the minimal path obtained by reversing the direction of front propagation. The difference in the two paths highlights the error caused by the directional dependence of the Fast Marching method.

In this paper we propose two methods to overcome the above-mentioned shortcomings. The first method uses a linear/bilinear model locally to estimate τ along the direction of the propagating front within each grid cell. Here we use a continuous model to estimate τ and also take the direction of arrival into consideration. We also discuss how the scheme can be made truly isotropic by removing any bias due to the marching direction. We call this method the Interpolated Fast Marching Method and it is discussed in detail in Section 2. In the second method we calculate u on an upsampled grid. In upsampling the grid, τ in the neighborhood of each grid point becomes constant, which eliminates the need to estimate τ using a continuous model. We will use the value of τ from the direction of arriving front. The upsampled version of the 4 and 8-connected neighbor schemes are discussed in Section 3. Finally, in Section 4 we describe a few numerical experiments conducted to highlight the significance of making the fast marching method independent of direction and we test the accuracy of the proposed methods.

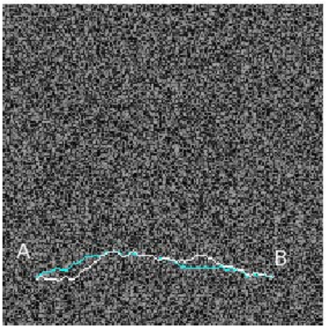


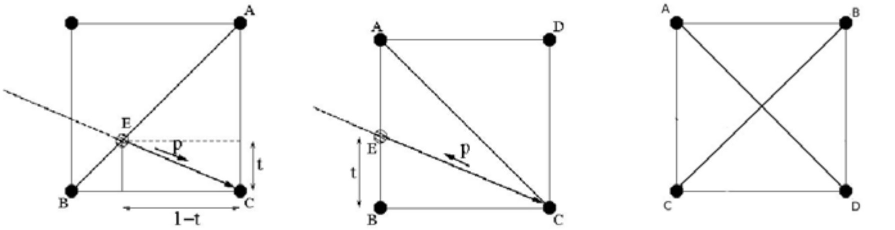
Fig. 2. Image with random noise

2 Interpolated Fast Marching Method

For interpolated Fast Marching scheme we will assume τ to be continuous around each grid point and use linear/bilinear interpolation to estimate the value of the local traveling cost within each grid cell. Here we will derive the equations for the linear and bilinear Interpolated Fast Marching schemes. To estimate the traveling cost in a grid cell, the bilinear scheme will use the value of τ from all the grid points for a given quadrant. Since only 2 neighbors are used in each quadrant to calculate u in a 4-connected neighbor scheme, we only discuss the 8-connected neighbor scheme with bilinear interpolation.

2.1 Linear Interpolation

4-Connected Neighbors Scheme. Consider a front arriving at the grid point C from the quadrant AB and intersecting $\overline{AB}$ at E as shown in Figure 3(a). We will use the linear interpolation of the local traveling cost along the path $\overrightarrow{EC}$ to compute u_C. Thus the accumulated cost at C will be,



(a) 4-Connected Neighbors Scheme (b) 8-Connected Neighbors Scheme (c) Isotropic triangulation of a Grid Cell

Fig. 3. Triangulation of Grid cells

$$u_C = \min_{0 \leq t \leq 1} \Big\{ u_B(1-t) + u_A t + \int_0^1 \tau(p) \sqrt{t^2 + (1-t)^2} dp \Big\}. \tag{2}$$

Substituting, $\tau(p) = \tau_C + (\tau_A - \tau_C)p(1-t) + (\tau_B - \tau_C)pt, 0 \leq p \leq 1$, in (2) we get,

$$u_C = \min_{0 \leq t \leq 1} \left\{ u_B(1-t) + u_A t + \sqrt{t^2 + (1-t)^2} \left(\frac{\tau_A + \tau_C}{2} + \frac{\tau_B - \tau_A}{2} t \right) \right\}. \tag{3}$$

We get the necessary optimality condition to obtain the minimum of u_C by solving $\frac{du_C}{dt} = 0$, which yields,

$$\begin{aligned} &u_A - u_B + \sqrt{t^2 + (1-t)^2} \left(\frac{\tau_B - \tau_A}{2} t \right) \\ &+ \frac{2t-1}{\sqrt{t^2 + (1-t)^2}} \left(\frac{\tau_A + \tau_C}{2} + \frac{\tau_B - \tau_A}{2} t \right) = 0. \end{aligned} \tag{4}$$

8-Connected Neighbors Scheme. The geometry for 8-connected neighbors is shown in Figure 3(b). Using linear interpolation to estimate the local traveling cost along $\overrightarrow{EC}$, the accumulated cost, u_C, will be,

$$u_C = \min_{0 \leq t \leq 1} \Big\{ u_B(1-t) + u_A t + \int_0^1 \tau(p) \sqrt{1 + t^2} dp \Big\}. \tag{5}$$

Substituting, $\tau(p) = \tau_C + (\tau_B - \tau_C)p + (\tau_A - \tau_B)pt, 0 \leq p \leq 1$, in (5) we get,

$$u_C = \min_{0 \leq t \leq 1} \left\{ u_A t + u_B(1-t) + \sqrt{1 + t^2} \left(\frac{\tau_B + \tau_C}{2} + \frac{\tau_A - \tau_B}{2} t \right) \right\}. \tag{6}$$

Again the minimizer of u_C can be obtained by solving $\frac{du_C}{dt} = 0$. Thus we have,

$$u_A - u_B + \sqrt{1 + t^2} \left(\frac{\tau_A - \tau_B}{2} \right) + \frac{t}{\sqrt{1 + t^2}} \left(\frac{\tau_B + \tau_C}{2} + \frac{\tau_A - \tau_B}{2} t \right) = 0. \tag{7}$$

Isotropic Linear Interpolation Scheme. Figure 3(a) and 3(b) show the triangulation of a grid cell for the 4 and 8 neighbor schemes respectively. Depending on the front direction one of the quadrant/octant will be chosen to estimate the accumulated cost. But this will induce a directional bias. To overcome this directional bias, we will have to consider all possible triangulations shown in Figure 3(c). In effect the accumulated cost across a grid cell must be the minimum of the solutions obtained using the 4 and 8 neighbor schemes. This would make the scheme completely unbiased to direction and we call this scheme the Iso-Linear scheme.

2.2 Bilinear Interpolation

8-Connected Neighbors Scheme. The bilinear interpolation to estimate the local traveling cost along $\overrightarrow{EC}$ is given by,

$$\tau(p) = \tau_A(p)(pt) + \tau_B(p)(1-pt) + \tau_C(1-p)(1-pt) + \tau_D(1-p)(pt).$$

It is inherently independent of any directional bias within a grid cell. Substituting, this value of $\tau(p)$ for $0 \le p \le 1$, in (5) we get,

$$u_C = \min_{0\le t\le 1} \left\{ u_A t + u_B(1-t) + \sqrt{1+t^2}\left(\frac{\tau_B+\tau_C}{2} + \frac{\tau_A-\tau_B}{3}t + \frac{\tau_D-\tau_C}{6}t\right)\right\}. \tag{8}$$

We will again solve $\frac{du_C}{dt} = 0$, which yields,

$$\begin{aligned} &u_A - u_B + \sqrt{1+t^2}\left(\frac{\tau_A-\tau_B}{3} + \frac{\tau_D-\tau_C}{6}\right) \\ &+ \frac{t}{\sqrt{1+t^2}}\left(\frac{\tau_B+\tau_C}{2} + \frac{\tau_A-\tau_B}{3}t + \frac{\tau_D-\tau_C}{6}t\right) = 0. \end{aligned} \tag{9}$$

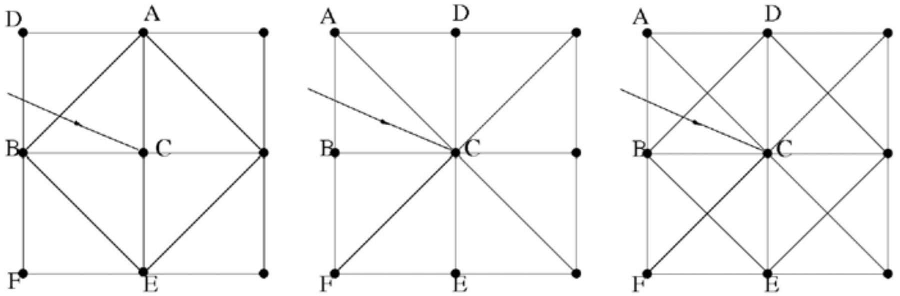
Algebraic manipulations on (4), (7) and (9) will yield quartic equations. We used the Ferrari and Newton methods to solve these quartic equations. We compared the solutions from both techniques and found that they generate equally accurate solutions. Since Newton's method has a quadratic convergence, three iterations were sufficient for convergence. Fixing the number of iterations in each update step also ensures that we have the same computation complexity in each update. This makes the technique suitable to implement on hardware. The solution to Newton's method has fewer (logical and mathematical) operations in comparison to finding the Ferrari (analytic) solution; hence using Newton's method is computationally efficient. We compare the computation times of the two methods on a 500x500 grid in the Table 1. Here we call the 4 and 8-connected neighbor linear Interpolated Fast Marching schemes, Linear-4 and Linear-8 respectively and the 8-connected neighbor bilinear Interpolated Fast Marching scheme, Bilinear-8. The computation times were measured on a laptop with a 1.73 GHz Processor.

Table 1. Comparison of computation times

	Linear-4	Linear-8	Bilinear-8
Analytic (Ferrari)	1.51s	2.83s	3.23s
Newton's Method	0.51s	0.52s	0.65s

2.3 Marching Forward Loop

We will still follow the main loop as explained in the basic Fast Marching method [10]. But, when a *trial* point is *accepted* in the min heap structure we will compute the value of u from both the quadrants/octants which include the newly *accepted* point and replace the newly calculated u with the minimum of the two solutions and the existing value of u (if the point is marked as *trial*).



(a) 4-Connected Neighbors Scheme (b) 8-Connected Neighbors Scheme (c) Isotropic Fast Marching Scheme

Fig. 4. B is the newly *accepted* grid point and u_C is to be computed

Consider the example in Figure 4(a) where B is the newly *accepted* point and the accumulated cost at neighbor C is to be computed. As opposed to the basic fast marching technique, u_C does not solely depend on u_A,u_B,u_E and the local traveling cost, τ_C, but it also depends on the costs at all the other 8-connected neighbors. Thus, using the quadrant containing the minimum of u_A and u_E will not necessarily guarantee the minimum solution to (3). Hence we have to consider both the quadrants that contain B. If the front also arrives at C from the other two quadrants, they will be considered when the corresponding neighbors become *accepted*. The same argument can be extended to the 8-connected neighbor case shown in Figure 4(b). Here we only need to calculate u_C from the two octants containing $\overline{AB}$ and $\overline{FB}$ once point B is *accepted*. For the front arriving at point C as shown in Figure 2(c), we will consider the possibilities of the front arriving from $\overline{AB}$,$\overline{BD}$ and $\overline{DA}$.

We depart from the traditional Fast Marching method only in the update procedure for the accumulated cost, but follow the same main (outer) loop. Thus the parallel algorithm explained in Bronstein et al.[2], can be extended for the implementation on hardware.

3 Upsampled Fast Marching Method

Figure 5 shows that there is no overlap in the influence areas of τ on the upsampled grid. Here the solid circles are the grid points from the original grid. Since the traveling cost is constant in each grid cell, there is no directional bias in the calculation of u. We will compute u on the upsampled grid and then downsample the output on the original grid.

3.1 4-Connected Neighbors Scheme

In the upsampled grid, τ is constant in each quadrant around a grid point. Again the constant traveling cost within each grid cell makes this scheme isotropic. Depending on the direction of the front we will choose the value of τ in calculating u. For example, if

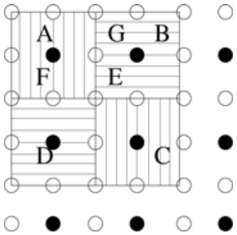


Fig. 5. No overlap in the influence areas of τ_A, τ_B, $\tau_C$and τ_D

the front arrives at E from the north-west then we would use τ_A (Figure 5). At the point G we would use τ_A for a front arriving from the west and τ_B for a front arriving from the east. We would use τ_A to calculate u_A irrespective of the direction of the arriving front. Since the value of τ is constant along the direction of the front at a sub-pixel level, it is not necessary to assume a locally continuous model in interpolating τ. Thus, the accumulated cost at E with the front arriving from the north-west would be,

$$u_E = \min_{0 \le t \le 0.5} \left\{ u_F t + u_G (0.5 - t) + \tau_A \sqrt{t^2 + (0.5 - t)^2} \right\} \tag{10}$$

This minimization leads to the closed form solution,

$$u_E = \begin{cases} \frac{\left(u_F + u_G + \sqrt{\delta}\right)}{2} & \text{if } \delta \ge 0 \\ \min(u_F, u_G) + \frac{\tau_A}{2} & \text{otherwise} \end{cases}$$

where, $\delta = \frac{\tau_A^2}{2} - (u_F - u_G)^2$.

3.2 8-Connected Neighbors Scheme

As in the case with 4-connected neighbors, τ is constant in each octant around a grid point in the upsampled grid. We note that there will be exactly one point in each octant that corresponds to a point in the original grid. We will use the corresponding value of τ to compute u.

By following the procedure described in Section 2.3, we calculate u only from the two octants that contain the newly *accepted* point. If F is the newly *accepted* point, we will calculate u_E in the octants containing $\overline{FA}$ and $\overline{FD}$ (Figure 5). The solution will be the minimum of the two values obtained. Thus, for a front arriving from north-west, the accumulated cost at E will be,

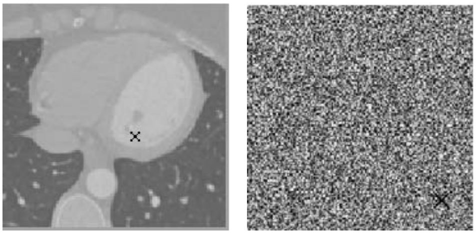
$$u_E = \min_{0 \le t \le 0.5} \left\{ u_A t + u_F (0.5 - t) + \tau_A \sqrt{0.5 + t^2} \right\} \tag{11}$$

giving the closed form solution,

$$u_E = \begin{cases} u_F + \frac{\tau_A}{2} & \text{if } u_F \le u_A \\ u_A + \sqrt{2}\frac{\tau_A}{2} & \text{if } \tau_A \le 2\sqrt{2}(u_F - u_A) \\ u_F + \frac{\sqrt{\tau_A^2 - 4(u_F - u_A)^2}}{2} & \text{otherwise} \end{cases}$$

4 Numerical Experiments

We conducted a few experiments to compare the proposed methods to the basic Fast Marching Method (FMM) [10], Tsitsiklis scheme [11], Shifted-Grid Fast Marching (SGFM) [6] and Multi-stencil Fast Marching (MSFM) [7]. We also compare the upsampled 4 and 8-connected neighbor Fast Marching schemes with the upsampled version of the SGFM scheme (upSG).

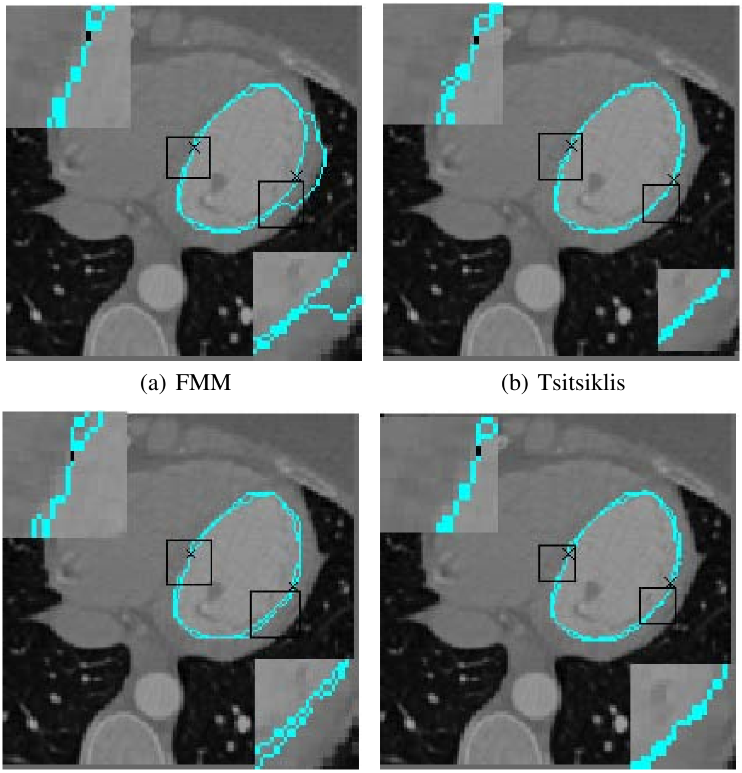


(a) Cardiac Data (b) Random noise

Fig. 6. Test Images

In the first experiment we pick a random point, marked by the 'x' in the images shown in Figure 6, and compute u at every point of the image. We then compute the total cost in propagating a front from each point of the image back to the point marked by the 'x'. We take the average of the difference (error) across the entire image. The numerical values are listed in the Table 2, under the column labeled Average Back Propagation Error (ABPE). We used the cost function, $\tau(x) = \frac{1}{1+|\nabla I|^2}$ for the cardiac image and $\tau(x) = I(x)$ for the random noise image.

In Figure 7 we present the results of segmenting the left ventricles in a 2D cardiac slice. To segment the image we pick a point on the boundary of the object and compute the saddle points as described in [5]. From each saddle point we then obtain two minimal paths back to the initial point; these paths will give the segmentation of the object. The minimal paths were obtained using a sub-pixel level back propagation scheme. We then choose the saddle point which minimizes the Chan-Vese [3] energy of the obtained segmentation. Images in Figure 7 show the overlay of segmentation curves initialized with 2 different user given points on the boundary. We see that the segmentation curves are not consistent and they depend on the initialization. This is mainly due to the difference in the marching direction in each case and weak image features at certain locations. We highlight certain regions in these images to compare the segmentation obtained from the different methods.



(a) FMM

(b) Tsitsiklis

(c) SGFM

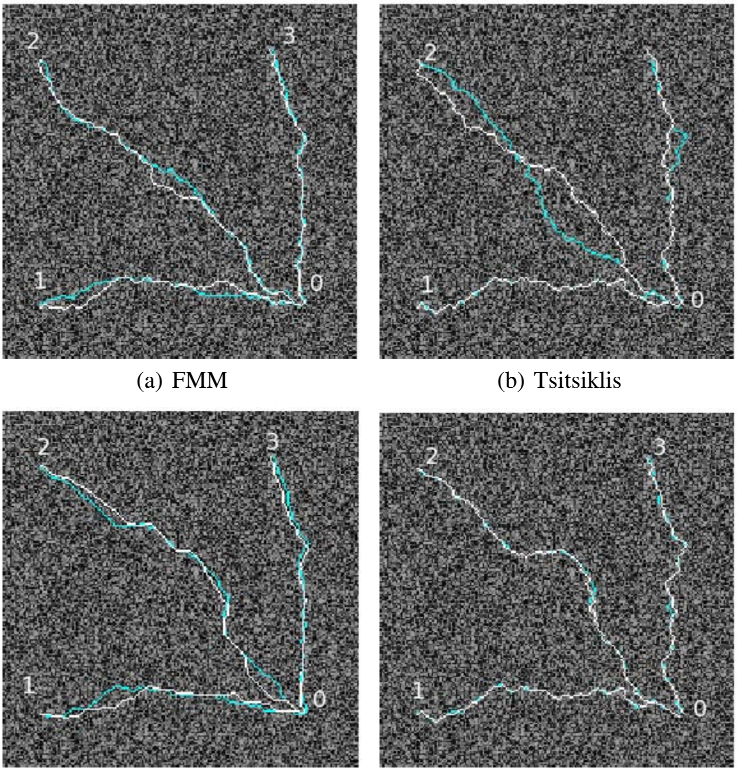
(d) Iso-Linear

Fig. 7. A comparison of segmentation

In the images shown in Figure 8, we compare the minimal paths obtained in traveling from point ‘0’ to points ‘1’,‘2’ and ‘3’ with the corresponding paths obtained by reversing the direction. We see that using interpolated FMM gives consistent paths, even in the absence of any strong image feature. The results are in accordance to the Average Back Propagation Errors listed in Table 2. The ABPE for the Tsitsiklis scheme is the highest and accordingly the paths obtained with the Tsitsiklis scheme show a lot of variation. Although the SGFM shows lower average error there are variations in the obtained minimal paths. This is because the interpolation of the cost function in SGFM is equivalent to image smoothing for the τ ($\tau(x) = I(x)$) used in this example. This decreases the corresponding average error, but it also decreases the difference in the geodesic distances of the various paths. Thus with the change in the marching direction, the back propagation takes different paths between two given points.

In the next example we compare the accuracy of the various techniques for two cost functions on a 50x50 grid,

$$\tau_1(x, y) = 1/20\sqrt{(sin\tfrac{x}{20}cos\tfrac{y}{20})^2 + (cos\tfrac{x}{20}sin\tfrac{y}{20})^2},$$
$$\tau_2(x, y) = 1/10\sqrt{(sin\tfrac{x}{20}cos\tfrac{y}{10})^2 + (cos\tfrac{x}{20}sin\tfrac{y}{10})^2}.$$



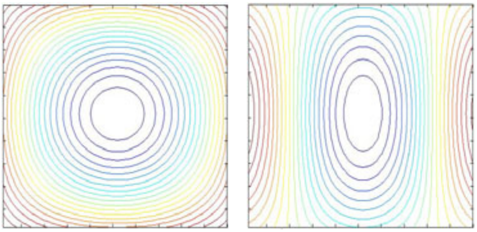
(a) FMM

(b) Tsitsiklis

(c) SGFM

(d) Iso-Linear

Fig. 8. A comparison of tracking



(a) Iso-contour: u_1

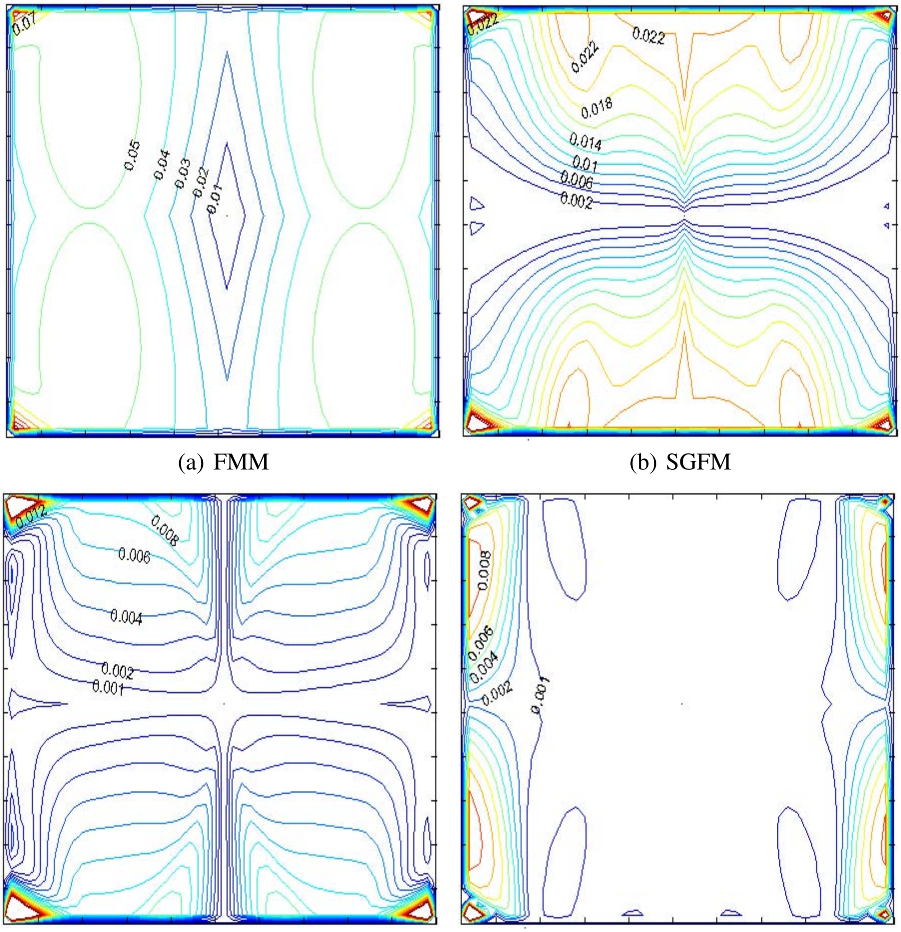
(b) Iso-contour: u_2

Fig. 9. Iso-contours

The iso-contours of $u_{analytic}$ are shown in Figure 9. The geodesics from the center $(26, 26)$ of the grid will be straight lines for τ_1 and curved for τ_2. Since, we have the analytic solution for these cost functions, we can compare the L_1, L_2 and L_∞ norms for each method.

Table 2. Error norms for τ_1 and τ_2, Average Back Propagation Errors and Computation times

	τ_1			τ_2			ABPE		Time
	L_1	L_2	L_∞	L_1	L_2	L_∞	I_1	I_2	(s)
FMM	$2.46\text{x}10^{-2}$	$6.73\text{x}10^{-4}$	0.0380	$4.37\text{x}10^{-2}$	$2.07\text{x}10^{-3}$	0.1060	0.0725	0.3901	0.27
Tsitsiklis	$2.14\text{x}10^{-2}$	$4.89\text{x}10^{-4}$	0.0281	$3.81\text{x}10^{-2}$	$1.57\text{x}10^{-3}$	0.0825	0.1007	0.4348	0.26
MSFM	$2.36\text{x}10^{-2}$	$6.07\text{x}10^{-4}$	0.0349	$4.23\text{x}10^{-2}$	$1.94\text{x}10^{-3}$	0.1007	0.0825	0.3572	0.29
SGFM	$2.33\text{x}10^{-3}$	$6.32\text{x}10^{-6}$	0.0051	$1.25\text{x}10^{-2}$	$2.14\text{x}10^{-4}$	0.0580	0.0022	0.0277	0.33
Linear4	$1.10\text{x}10^{-2}$	$1.71\text{x}10^{-4}$	0.0285	$1.69\text{x}10^{-2}$	$4.01\text{x}10^{-4}$	0.0875	0.0122	0.1036	0.51
Linear8	$2.25\text{x}10^{-3}$	$6.82\text{x}10^{-6}$	0.0046	$4.46\text{x}10^{-3}$	$3.43\text{x}10^{-5}$	0.0596	0.0028	0.0355	0.52
IsoLinear	$2.25\text{x}10^{-3}$	$6.82\text{x}10^{-6}$	0.0046	$4.03\text{x}10^{-3}$	$3.11\text{x}10^{-5}$	0.0596	0.0109	0.0911	0.91
Bilinear8	$2.74\text{x}10^{-3}$	$9.42\text{x}10^{-6}$	0.0052	$5.01\text{x}10^{-3}$	$4.10\text{x}10^{-5}$	0.0607	0.0028	0.0101	0.65
Up4	$1.79\text{x}10^{-3}$	$7.60\text{x}10^{-6}$	0.0101	$3.14\text{x}10^{-3}$	$2.89\text{x}10^{-5}$	0.0655	0.0451	0.1919	1.37
Up8	$2.99\text{x}10^{-4}$	$1.96\text{x}10^{-7}$	0.0014	$1.54\text{x}10^{-3}$	$7.81\text{x}10^{-6}$	0.0289	0.0011	0.0221	1.42
UpSG	$1.96\text{x}10^{-3}$	$4.15\text{x}10^{-6}$	0.0035	$1.20\text{x}10^{-2}$	$1.94\text{x}10^{-4}$	0.0566	0.0015	0.0141	1.42



(a) FMM

(b) SGFM

(c) Iso-Linear

(d) Upsampled-8

Fig. 10. Iso-contours of errors for τ_2

$$L_1 = \text{mean}(|u - u_{analytic}|),$$
$$L_2 = \text{mean}(|u - u_{analytic}|^2),$$
$$L_\infty = \max(|u - u_{analytic}|).$$

The numerical errors in using cost functions τ_1 and τ_2 are listed in Table 2. Notice that the error norms show significant improvement for the proposed methods, especially in the case with curved geodesics (τ_2). The iso-contours of the errors for τ_2 while using FMM, SGFM, Iso-Linear and up8 are shown in Figure 10.

We also enlist the computation times for each of these methods on a 500x500 grid in the last column of Table 2. All computation times were measured on a laptop with a 1.73 GHz Processor.

5 Conclusion

In this paper we present techniques to make the fast marching method independent of the marching direction and thus improve the accuracy of the Fast Marching Method. One approach interpolates the local traveling cost along the front and the other computes u on an upsampled grid. We also showed that combining the 8 and 4-connected neighbor schemes further reduces the inaccuracy by considering all possible directions of the arrival of the front. We have compared both our approaches to the existing Fast Marching techniques and we have shown a significant improvement over them. Although both our approaches have higher computation times, they can be implemented efficiently on hardware and they are practical solutions to eliminate the inaccuracies of existing techniques.

Acknowledgements

This work was partially supported by NIH grant R01-HL-085417, NSF grant CCF-0728911 as well as an EmTech seed grant.

References

1. Adalsteinsson, D., Sethian, J.A.: A fast level set method for propagating interfaces. Journal of Computational Physics 118, 269–277 (1994)
2. Bronstein, A.M., Bronstein, M.M., Devir, Y.S., Kimmel, R., Weber, O.: Parallel algorithms for approximation of distance maps on parametric surfaces (2007)
3. Chan, T., Vese, L.: An active contour model without edges. In: Nielsen, M., Johansen, P., Fogh Olsen, O., Weickert, J. (eds.) Scale-Space 1999. LNCS, vol. 1682, pp. 141–151. Springer, Heidelberg (1999)
4. Cohen, L., Kimmel, R.: Global minimum for active contour models: A minimal path approach. International Journal of Computer Vision 24, 57–78 (1997)
5. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: A minimal path approach. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 666 (1996)
6. Danielsson, P.E., Lin, Q.: A modified fast marching method. In: SCIA, pp. 1154–1161 (2003)

7. Hassouna, M.S., Farag, A.A.: Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(9), 1563–1574 (2007)
8. Kim, S., Folie, D.: The group marching method: An o(n) level set eikonal solver
9. Polymenakos, L.C., Bertsekas, D.P., Tsitsiklis, J.N.: Implementation of efficient algorithms for globally optimal trajectories. IEEE Transactions on Automatic Control 43, 278–283 (1998)
10. Sethian, J.A.: Level Set Methods and Fast Marching Methods. Cambridge University Press, Cambridge (1999)
11. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. IEEE Transactions on Automatic Control 40(9), 1528–1538 (1995)

Clustering Complex Data with Group-Dependent Feature Selection

Yen-Yu Lin[1,2], Tyng-Luh Liu[1], and Chiou-Shann Fuh[2]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2] Department of CSIE, National Taiwan University, Taipei, Taiwan
{yylin,liutyng}@iis.sinica.edu.tw, fuh@csie.ntu.edu.tw

Abstract. We describe a clustering approach with the emphasis on detecting coherent structures in a complex dataset, and illustrate its effectiveness with computer vision applications. By complex data, we mean that the attribute variations among the data are too extensive such that clustering based on a single feature representation/descriptor is insufficient to faithfully divide the data into meaningful groups. The proposed method thus assumes the data are represented with various feature representations, and aims to uncover the underlying cluster structure. To that end, we associate each cluster with a boosting classifier derived from *multiple kernel learning*, and apply the cluster-specific classifier to feature selection across various descriptors to best separate data of the cluster from the rest. Specifically, we integrate the multiple, correlative training tasks of the cluster-specific classifiers into the clustering procedure, and cast them as a joint constrained optimization problem. Through the optimization iterations, the cluster structure is gradually revealed by these classifiers, while their discriminant power to capture similar data would be progressively improved owing to better data labeling.

1 Introduction

Clustering is a technique to partition the data into groups so that *similar* (or *coherent*) objects and their properties can be readily identified and exploited. While such a goal is explicit and clear, the notion of *similarity* is often not well defined, partly due to the lack of a universally applicable *similarity measure*. As a result, previous research efforts on developing clustering algorithms mostly focus on dealing with different scenarios or specific applications. In the field of vision research, performing data clustering is essential in addressing various tasks such as object categorization [1,2] or image segmentation [3,4]. Despite the great applicability, a fundamental difficulty hindering the advance of clustering techniques is that the intrinsic cluster structure is not evidently revealed in the feature representation of complex data. Namely, the resulting similarities among data points do not faithfully reflect their true relationships.

We are thus motivated to consider establishing a clustering framework with the flexibility of allowing the data to be characterized by multiple descriptors. The generalization aims to bridge the gap between the resulting data similarities and their underlying relationships. Take, for example, the images shown in

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 84–97, 2010.

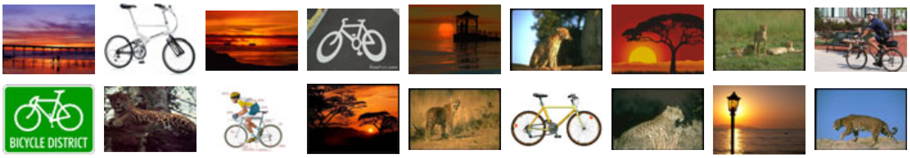


Fig. 1. Images from three different categories: `sunset`, `bicycle` and `jaguar`

Fig. [1]. Without any ambiguities, one can easily divide them into three clusters. Nevertheless, say, in an object recognition system, color related features are required to separate the images in category `sunset` from the others. Analogously, shape and texture based features are respectively needed for describing categories `bicycle` and `jaguar`. This example not only illustrates the importance of using multiple features but also points out that the optimal features for ensuring each cluster coherent often vary from cluster to cluster.

The other concept critical to our approach is *unsupervised feature selection*. It is challenging due to the absence of data labels to guide the relevance search, e.g., [5,6]. To take account of the use of multiple descriptors for clustering, our formulation generalizes unsupervised feature selection to its *cluster/group-dependent* and *cross feature space* extensions. To that end, each cluster is associated with a classifier learned with multiple kernels to give a good separation between data inside and outside the cluster, and data are dynamically assigned to appropriate clusters through the progressive learning processes of these cluster-specific classifiers. Iteratively, the learned classifiers are expected to facilitate the revealing of the intrinsic cluster structure, while the progressively improved clustering results would provide more reliable data labels in learning the classifiers.

Specifically, we integrate the multiple, correlative training processes of the cluster-specific classifiers into the clustering procedure, and realize the unified formulation by 1) proposing a general constrained optimization problem that can accommodate both fully unlabeled and partially labeled datasets; and 2) implementing *multiple kernel learning* [7] in a *boosting* way to construct the cluster-specific classifiers. Prior knowledge can thus be conveniently exploited in choosing a proper set of visual features of diverse forms to more precisely depict the data. Indeed our approach provides a new perspective of applying multiple kernel learning, which typically addresses supervised applications, to both unsupervised and semisupervised ones. Such a generalization is novel in the field. Different from other existing clustering techniques, our method can not only achieve better clustering results but also have access to the information regarding the commonly shared features in each cluster.

2 Related Work

Techniques on clustering can vary considerably in many aspects, including assuming particular principles for data grouping, making different assumptions about cluster shapes or structures, and using various optimization techniques for problem solving. Such variations however do not devalue the importance of clustering being a fundamental tool for unsupervised learning. Instead, clustering methods such as *k-means*, *spectral clustering* [3,8], *mean shift* [4] or *affinity propagation* [9] are constantly applied in more effectively solving a broad range of computer vision problems.

Although most clustering algorithms are developed with theoretic support, their performances still depend critically on the feature representation of data. Previous approaches, e.g., [5,6], concerning the limitation have thus suggested to perform clustering and feature selection simultaneously such that relevant features are emphasized. Due to the inherent difficulty of unsupervised feature selection, methods of this category often proceed in an iterative manner, namely, the steps of feature selection and clustering are carried out alternately.

Feature selection can also be done cluster-wise, say, by imposing the *Gaussian mixture models* on the data distribution, or by learning a distance function for each cluster via re-weighting feature dimensions such as the formulations described in [10,11]. However, these methods typically assume that the underlying data are in a single feature space and in form of vectors. The restriction may reduce the overall effectiveness when the data of interest can be more precisely characterized by considering multiple descriptors and diverse forms, e.g., bag-of-features [12,13] or pyramids [14,15].

Xu et al. [16] instead consider the large margin principle for measuring how good a data partitioning is. Their method first maps the data into the kernel-induced space, and seeks the data labeling (clustering) with which the maximum margin can be obtained by applying SVMs to the then labeled data. Subsequently, Zhao et al. [17] introduce a cutting-plane algorithm to generalize the framework of maximum margin clustering from binary-class to multi-class.

The technique of *cluster ensembles* by Strehl and Ghosh [18] is most relevant to our approach. It provides a useful mechanism for combining multiple clustering results. The ensemble partitioning is optimized such that it shares as much information with each of the elementary ones as possible. Fred and Jain [19] introduce the concept of *evidence accumulation* to merge various clusterings to a single one via a voting scheme. These methods generally achieve better clustering performances. Implicitly, they also provide a way for clustering data with multiple feature representations: One could generate an elementary clustering result for each data representation, and combine them into an ensemble one. However, the obtained partitioning is optimized in a global fashion, neglecting that the optimal features are often cluster-dependent.

Finally, it is possible to overcome the unsupervised nature of clustering by incorporating a small amount of labeled data in the procedure so that satisfactory results can be achieved, especially in complex tasks. For example, Xing et al. [20] impose *side information* for metric learning to facilitate clustering, while Tuzel et al. [2] utilize pairwise constraints to perform semisupervised clustering in a kernel-induced feature space.

3 Problem Definition

We formalize and justify the proposed clustering technique in this section. Prior to that, we need to specify the notations adopted in the formulation.

3.1 Notations

Given a dataset $D = \{\mathbf{x}_i\}_{i=1}^N$, our goal is to partition D into C clusters. We shall use a *partition matrix*, $Y = [y_{ic}] \in \{0, 1\}^{N\times C}$, to represent the clustering result, where $y_{ic} = 1$ indicates that $\mathbf{x}_i$ belongs to the cth cluster, otherwise $y_{ic} = 0$. Besides, let $\mathbf{y}_{i,:}$ and $\mathbf{y}_{:,c}$ denote the ith row and cth column of Y respectively.

To tackle complex clustering tasks, we consider the use of multiple descriptors to more precisely characterize the data. These descriptors may result in diverse forms of feature representations, such as vectors [21], bags of features [22], or pyramids [14]. To avoid directly working with these varieties, we adopt a strategy similar to that in [15,23], where kernel matrices are used to provide a uniform representation for data under various descriptors. Specifically, suppose M kinds of descriptors are employed to depict each sample, i.e., $\mathbf{x}_i = \{\mathbf{x}_{i,m} \in \mathcal{X}_m\}_{m=1}^M$. For each descriptor m, a non-negative distance function $d_m : \mathcal{X}_m \times \mathcal{X}_m \rightarrow \mathbb{R}$ is associated. The corresponding kernel matrix K_m and kernel function k_m can be established by

$$K_m(i, j) = k_m(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma_m d_m^2(\mathbf{x}_{i,m}, \mathbf{x}_{j,m})\right), \tag{1}$$

where γ_m is a positive constant. By applying the procedure to each descriptor, a kernel bank Ω of size M is obtained, i.e., $\Omega = \{K_m\}_{m=1}^M$. The kernel bank will serve as the information bottleneck in the sense that data access is restricted to referencing only the M kernels. This way our method can conveniently work with various descriptors without worrying about their diversities.

3.2 Formulation

The idea of improving clustering performances for complex data is motivated by the observation that the optimal features for grouping are often cluster-dependent. Our formulation associates each cluster with a classifier to *best* interpret the relationships among data and the cluster. Specifically, a cluster-specific classifier is designed to divide the data so that its members would share certain common features, which are generally distinct from the rest. Furthermore, the *goodness* of the clustering quality about a resulting cluster can be explicitly measured by the induced loss (namely, the *degree* of difficulty) in learning the specific classifier. It follows that the proposed clustering seeks an optimal data partitioning with the minimal total loss in jointly learning all the C cluster-specific classifiers.

As one may notice that our discussion so far would lead to a cause-and-effect dilemma: While the data labels are required in learning the cluster-specific classifiers, they in turn can only be determined through the clustering results implied by these classifiers. We resolve this difficulty by incorporating the learning processes of the cluster-specific classifiers into the clustering procedure, and cast the task as the following constrained optimization problem:

$$\min_{Y,\{f_c\}_{c=1}^{C}} \sum_{c=1}^{C} \text{Loss}(f_c, \{\mathbf{x}_i, y_{ic}\}_{i=1}^{N}) \tag{2}$$

$$\text{subject to } Y \in \{0,1\}^{N\times C}, \tag{3}$$

$$\mathbf{y}_{i,:}\mathbf{e}_C = 1, \text{ for } i = 1, 2, ..., N, \tag{4}$$

$$\ell \leq \mathbf{e}_N^\top \mathbf{y}_{:,c} \leq u, \text{ for } c = 1, 2, ..., C, \tag{5}$$

$$\mathbf{y}_{i,:} = \mathbf{y}_{j,:}, \text{ if } (i,j) \in S, \tag{6}$$

$$\mathbf{y}_{i,:} \neq \mathbf{y}_{j,:}, \text{ if } (i,j) \in S', \tag{7}$$

where $\{f_c\}_{c=1}^{C}$ are the cluster-specific classifiers. $\mathbf{e}_C$ and $\mathbf{e}_N$ are column vectors, whose elements are all one, of dimensions C and N respectively.

We now give justifications for the above constrained optimization problem. Our discussions focus first on the part of constraints. With (3) and (4), Y is guaranteed to be a valid partition matrix. Since in practical applications most clusters are rarely of extreme sizes, we impose the desired upper bound u and lower bound ℓ of the cluster size in (5). The remaining constraints (6) and (7) are optional so that our method can be extended to address semisupervised learning. In that case, (6) and (7) would provide a set of pairwise instance-level constraints, each of which specifies either a pair of data points must reside in the same cluster or not. S in (6) and S' in (7) are respectively used to denote the collections of these *must-links* and *cannot-links*.

Assuming that all the constraints are satisfied, the formulation would look for optimal data partitioning Y^* such that, according to (2), the total induced loss of all the cluster-specific classifiers is minimized. That is, the proposed clustering approach would prefer that data residing in each cluster are well separated from the rest by the cluster-specific classifier (and hence yields a small loss), which is derived by coupling a discriminant function with an optimal feature selection to achieve the desired property. This implies that most of the data in an arbitrary cluster c would share some coherent characteristics implicitly defined by the optimal feature selection in forming f_c^*. The proposed optimization elegantly connects the unsupervised clustering procedure with the supervised learning of the specific classifiers. By jointly addressing the two tasks, our method can uncover a reasonable cluster structure even for a complex dataset.

4 Optimization Procedure

To deal with the cause-and-effect factor in (2), we consider an iterative strategy to solve the constrained optimization problem. At each iteration, the cluster-specific classifiers $\{f_c\}_{c=1}^{C}$ and the partition matrix Y are alternately optimized. More specifically, $\{f_c\}$ are first optimized while Y is fixed, and then their roles are switched. The iterations are repeated until the loss cannot be further reduced.

4.1 On Learning Cluster-Specific Classifiers

Notice that in the constrained optimization problem (2) the cluster-specific classifiers $\{f_c\}$ only appear in the objective function, and there is no correlation among them once Y is fixed. Thus, these classifiers can be optimized independently by minimizing their corresponding loss function. That is, f_c can be derived by considering only the binary labeled data $\{\mathbf{x}_i, y_{ic}\}_{i=1}^N$.

Our choice of selecting a suitable supervised learning methodology for constructing the classifiers is based on two key requirements stemming from the properties related to the classifiers and the iterative training process. First, the cluster-specific classifiers should be generated by using information from multiple kernels, i.e., via multiple kernel learning [7]. Second, the *degree of data fitting* in the classifiers can be conveniently controlled. The latter requirement arises due to the expected phenomenon that the data labels $\{y_{ic}\}_{i=1}^N$ would be noisy during the earlier iterations, and then progressively become more accurate through the iterative optimization. By addressing this effect, we can significantly alleviate the possibility of overfitting or underfitting in learning the classifiers. Having taken the two into account, we consider each f_c a boosting classifier. In what follows, we describe the two main elements in learning such classifiers.

The Pool of Weak Learners. We adopt a similar strategy proposed in [24]. To begin with, the discriminant power of each kernel is transferred into a set of weak learners, called *dyadic hypercuts* [25]. We then construct the pool of weak learners by including the dyadic hypercuts generated from all the kernels in Ω. The procedure naturally enables a boosting algorithm to learn classifiers that inherit the discriminant power from the multiple kernels.

A dyadic hypercut h is specified by three parameters: a positive sample $\mathbf{x}_p$, a negative sample $\mathbf{x}_n$, and a kernel function k_m. (Note that the positive and negative samples here depend on labels $\{y_{ic}\}_{i=1}^N$.) The model for prediction is

$$h(\mathbf{x}) = a \cdot \operatorname{sign}(k_m(\mathbf{x}_p, \mathbf{x}) - k_m(\mathbf{x}_n, \mathbf{x}) - \theta) + b, \tag{8}$$

where a and b are real values, and θ is used for thresholding. The size of the set of weak learners is $|\mathcal{H}| = N^+ \times N^- \times M$, where N^+ (N^-) is the number of positive (negative) training data, and M is the number of kernels.

Loss Function for Boostig. Among the many choices of loss function for learning boosting classifiers, we have implemented two of the most popular ones, i.e., *ExpLoss* and *LogLoss* [26,27], to test our method. In our experiments, *LogLoss* leads to better performances, and is thus adopted. It follows that in (2) we have

$$\text{Loss}(f_c, \{\mathbf{x}_i, y_{ic}\}_{i=1}^N) = \sum_{i=1}^N \ln\left(1 + \exp\left(-\tilde{y}_{ic} f_c(\mathbf{x}_i)\right)\right), \tag{9}$$

where $\tilde{y}_{ic} = 2y_{ic} - 1$ is to convert a binary label $y_{ic} \in \{0, 1\}$ in partition matrices to $\tilde{y}_{ic} \in \{-1, 1\}$ for boosting models. With the pool of weak learners generated from the kernel bank Ω and the loss function (9), all cluster-specific classifiers $\{f_c\}$ can be learned one by one via *LogitBoost* [27].

4.2 On Assigning Data into Clusters

Once the cluster-specific classifiers are fixed, we illustrate that how the partition matrix Y in (2) can be optimized by *binary integer programming* (BIP) [28]. For the ease of our discussion, the canonical form of a BIP problem is given below

$$\min_{\mathbf{z}} \ \mathbf{d}^\top \mathbf{z} \tag{10}$$

$$\text{subject to} \ A\mathbf{z} \leq \mathbf{b} \text{ and } A_{eq}\mathbf{z} = \mathbf{b}_{eq}, \tag{11}$$

$$z_i \in \{0, 1\}. \tag{12}$$

It suffices to show the proposed constrained optimization can be transformed to the above form. To rewrite the objective function (2) as the inner product (10), we let $\mathbf{z} \equiv vec(Y) = [y_{11} \ \cdots y_{1C} \cdots \ y_{ic} \ \cdots \ y_{NC}]^\top$, the *vectorization* of partition matrix Y and set the column vector $\mathbf{d} = [d_{ic}]$ as

$$d_{ic} = \ln\left(1 + \exp\left(-f_c(\mathbf{x}_i)\right)\right) + \sum_{c'=1 \ \& \ c' \neq c}^{C} \ln\left(1 + \exp\left(f_{c'}(\mathbf{x}_i)\right)\right). \tag{13}$$

The definitions of $\mathbf{d}$ and $\mathbf{z}$ would lead to

$$\mathbf{d}^\top \mathbf{z} = \mathbf{d}^\top vec(Y) = \sum_{c=1}^{C} \sum_{i=1}^{N} \ln\left(1 + \exp\left(-\tilde{y}_{ic} f_c(\mathbf{x}_i)\right)\right). \tag{14}$$

Indeed the derivation of (14) is based on (4). For each sample $\mathbf{x}_i$, there is one and only one element whose value is 1 in the vector $\mathbf{y}_{i,:} = [y_{i1} \ \cdots \ y_{iC}]$. And no matter which element equals to 1, we have

$$\sum_{c=1}^{C} d_{ic} y_{ic} = \sum_{c=1}^{C} \ln\left(1 + \exp\left(-\tilde{y}_{ic} f_c(\mathbf{x}_i)\right)\right). \tag{15}$$

Now, summing over all the data on the both sides of (15) gives (14). We are left to express the constraints (3)–(7) into (11) and (12). Since the derivations related to (3)–(6) are straightforward, we focus on the reduction of constraint (7). To represent $\mathbf{y}_{i,:} \neq \mathbf{y}_{j,:}$, we consider additional auxiliary variables, $\mathbf{p} \in \{0, 1\}^{C \times 1}$ and $\mathbf{q} \in \{0, 1\}^{C \times 1}$, and the following three constraints

$$\mathbf{y}_{i,:} - \mathbf{y}_{j,:} = \mathbf{p} - \mathbf{q}, \ \mathbf{p} + \mathbf{q} \leq \mathbf{e}_C, \text{ and } \mathbf{e}_C^\top \mathbf{p} + \mathbf{e}_C^\top \mathbf{q} = 2. \tag{16}$$

It can be verified that $\mathbf{y}_{i,:} \neq \mathbf{y}_{j,:}$ if and only if the constraints in (16), which are all conformed to (11), hold. Thus, our discussion justifies that when $\{f_c\}$ are fixed, the constrained optimization problem (2) can be effectively solved by BIP to obtain a new data partitioning Y.

4.3 Implementation Details

In solving the constrained optimization, we begin by providing an initial Y derived by randomly splitting the data into clusters of similar sizes. As it turns out the proposed optimization procedure is robust against different initializations, and converges fast. (Further details will be discussed in the next section.)

It is useful to progressively adjust the data fitting power in learning the classifiers, since the reliability of the data labeling is expected to improve through the iterations. Specifically, say, at iteration t, we set the number of weak learners in f_c as `base` + t*`step_size`, where the value of `step_size` is decided by the tradeoff between the convergence speed and the risk of overfitting. In all our experiments, we have `base` = 5 and `step_size` = 2. Also note that the boosting classifiers tend to perfectly classify the training data, and underestimate the LogLoss (9). This can be resolved by *leave-one-out* estimation: The induced loss of sample $\mathbf{x}_i$ in (9) is evaluated by the classifier learned with the rest of the data. (For computational issue, we implement ten-fold cross-validation.)

Being a special case of integer programming, BIP is still *NP-hard*. A practical implementation of an appropriate methodology such as *branch-and-bound* or *cutting plane* would require a feasible initialization to reduce BIP into a series of linear programs, and thus speed up the underlying optimization process. In our case, we design a greedy scheme to find an initial set of data labels. We first assume an upper bound on the cluster size. Then, among those undecided samples we identify the next possible sample labeling such that the assignment yields the smallest loss and would not cause the size of the target cluster to exceed the upper bound. The process is repeated until the data labeling is completed. Given the initialization, we apply MOSEK [29] to efficiently solving the BIP problems. For example, it takes less than one second when $(N, C) = (600, 20)$.

5 Experimental Results

We carry out two sets of experiments: visual object categorization and face image grouping. The image data used in the experiments are complex and display rich variations caused by various factors. They nevertheless provide a good test bed to demonstrate the importance of using multiple feature representations. In the first experiment, we compare our approach with state-of-the-art clustering algorithms and discuss the convergence issue. In the second experiment, we show the advantages of our approach in the aspects of performing cluster-dependent, cross-space feature selection and incorporating partially labeled data.

5.1 Visual Object Categorization

Dataset. The Caltech-101 dataset [30], collected by Fei-Fei et al., is used in our experiments of object categorization. Following the setting in [1], we select the same twenty object categories from the Caltech-101 dataset, and randomly pick 30 images from each category to form a set of 600 images. The large and diverse intraclass variations make clustering over the dataset very challenging.

Descriptors, Distances and Kernels. We consider five different image descriptors and their corresponding distance function. Via (1), they yield the following kernels (denoted below in bold and in abbreviation):

Table 1. The performances in form of [ACC (%) / NMI] by different clustering methods. **Top**: each kernel is considered individually. **Bottom**: all kernels are used jointly.

kernel	k-means	Affinity Prop.	Spectral Clus.	Ours
GB	68.0 / 0.732	52.5 / 0.578	69.5 / 0.704	**75.0** / **0.742**
SIFT	62.5 / 0.680	59.8 / 0.638	62.5 / 0.668	**69.6** / **0.706**
SS	**65.7** / **0.659**	55.7 / 0.574	63.3 / 0.655	62.1 / 0.639
C2	37.8 / 0.417	47.5 / 0.517	**57.7** / **0.585**	51.2 / 0.550
PHOG	53.3 / 0.547	43.3 / 0.464	**61.0** / **0.624**	55.2 / 0.569

kernels	CE + k-means	CE + Affinity Prop.	CE + Spectral Clus.	Ours
All	73.8 / 0.737	63.3 / 0.654	77.3 / 0.758	**85.7** / **0.833**

- **GB**. For a given image, we randomly sample 400 edge pixels, and characterize them by the *geometric blur* descriptor [12]. With these image features, we adopt the distance function suggested in equation (2) of the work by Zhang et al. [22] to obtain the kernel.
- **SIFT**. The kernel is analogously constructed as is the kernel GB, except that the features are described with the *SIFT* descriptor [13].
- **SS**. We consider the *self-similarity* descriptor [31] over an evenly sampled grid of each image, and use k-means clustering to generate *visual words* from the resulting local features of all images. Then the kernel is built by matching *spatial pyramids*, which are introduced in [14].
- **C2**. Mutch and Lowe [21] have proposed a set of features that emulate the visual system mechanism. We adopt these biologically inspired features to depict images and construct an RBF kernel.
- **PHOG**. We also use the *PHOG* descriptor [15] to capture image features. Together with the χ^2 distance, the kernel is established.

Quantitative Results. In all the experiments, we set the number of clusters to the number of classes in ground truth, and evaluate clustering performances with the two criteria: *clustering accuracy* (ACC) [6], and *normalized mutual information* (NMI) [18]. The output ranges of the two criteria are both $[0, 1]$. The larger the values, the better the clustering results are. Our approach starts from a random initialization of data partitioning Y. We run our algorithm 20 times with different random partitionings, and report the average performance. Besides, we respectively set ℓ and u in (5) as $\lfloor 0.8k_1 \rfloor$ and $\lceil 1.2k_2 \rceil$, where k_1 and k_2 are the minimal and the maximal cluster sizes in the dataset respectively.

We first evaluate our method in the cases that each descriptor is used *individually*, and compare it with three popular clustering methodologies: k-means, affinity propagation [9], and spectral clustering [8]. The implementations for the three techniques are as follows. k-means works on data in Euclidean space, so we use *multidimensional scaling* [32] to recover the feature vectors of data from their pairwise distances. Affinity propagation detects representative exemplars (clusters) by considering similarities among data. We set the pairwise similarities as the negative distances. Spectral clustering and our approach both take a kernel matrix as input. The outcomes by the four clustering algorithms are shown in

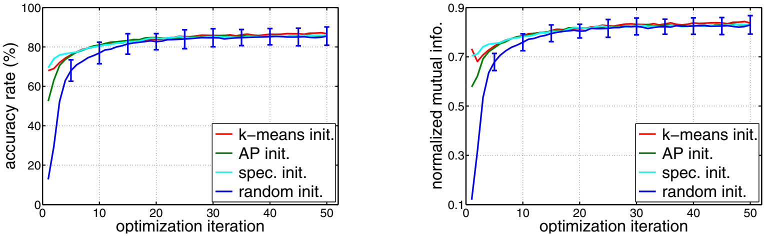


Fig. 2. With different initializations, the clustering accuracy (**left**) and normalized mutual information (**right**) of the proposed approach along the iterative optimization

Table 1 (**top**). In this setting, the proposed method outperforms k-means and affinity propagation, and is competitive with spectral clustering.

When multiple kernels are used simultaneously, we compare the proposed framework with cluster ensembles (CE) [18]. In particular, our implementation of cluster ensembles is to combine the five separately generated clustering results by one of the following three techniques: k-means, affinity propagation and spectral clustering. We report the results in Table 1 (**bottom**). First of all, our approach achieves significant improvements of 10.7% (= 85.7% − 75.0%) in ACC and 0.091 (= 0.833 − 0.742) in NMI over the best results obtained with a single kernel. It suggests that these kernels tend to complement one another, and our method can exploit this property to yield better clustering results. Furthermore, unlike that cluster ensembles relies on merging multiple clustering results in a global fashion, our approach performs cluster-dependent feature selection over multiple descriptors to recover the cluster structure. The quantitative results show that our method can make the most of multiple kernels, and improves the performances from 77.3% to 85.7% in ACC and from 0.758 to 0.833 in NMI.

Pertaining to the convergence issue, we evaluate our algorithm with 23 different initializations, including 20 random data partitionings and three meaningful clustering results by applying k-means, affinity propagation and spectral clustering to kernel GB, respectively. The clustering performances through the iterative optimization procedure are plotted in Fig. 2. It can be observed that the proposed optimization algorithm is efficient and robust: It converges within a few iterations and yields similar performances with diverse initializations.

5.2 Face Image Grouping

Dataset. The CMU PIE database [33] is used in our experiments of face image grouping. It comprises face images of 68 subjects. To evaluate our method for cluster-dependent feature selection, we divide the 68 people into four equal-size disjoint groups, each of which contains face images from 17 subjects reflecting a certain kind of variations. See Fig. 3 for an overview.

Specifically, for each subject in the first group, we consider only the images of the frontal pose (C27) taken in varying lighting conditions (those under the

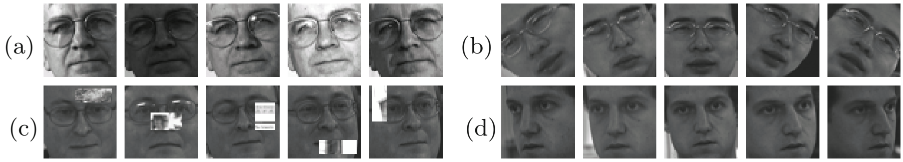


Fig. 3. Four kinds of intraclass variations caused by (a) different lighting conditions, (b) in-plane rotations, (c) partial occlusions, and (d) out-of-plane rotations

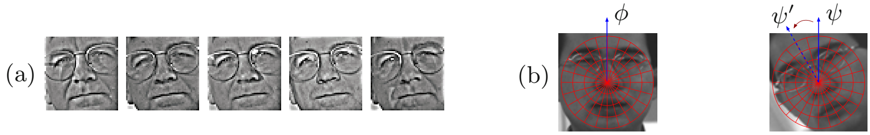


Fig. 4. (a) Images obtained by applying the delighting algorithm [34] to the five images in Fig. 3a. (b) Each image is divided into 96 regions. The distance between the two images is obtained when circularly shifting causes ψ' to be the new starting radial axis.

directory "`lights`"). For subjects in the second and third groups, the images with near frontal poses (C05, C07, C09, C27, C29) under the directory "`expression`" are used. While each image from the second group is rotated by a randomly sampled angle within $[-45°, 45°]$, each from the third group is instead occluded by a non-face patch, whose area is about ten percent of the face region. Finally, for subjects in the fourth group, the images with out-of-plane rotations are selected under the directory "`expression`" and with the poses (C05, C11, C27, C29, C37). All images are cropped and resized to 51×51 pixels.

Descriptors, Distances and Kernels. With the dataset, we adopt and design a set of visual features, and establish the following four kernels.

- **DeLight**. The data representation is obtained from the delighting algorithm [34], and the corresponding distance function is set as $1 - \cos\theta$, where θ is the angle between a pair of samples under the representation. Some delighting results are shown in Fig. 4a. It can be seen that variations caused by different lighting conditions are significantly alleviated under the representation.
- **LBP**. As is illustrated in Fig. 4b, we divide each image into $96 = 24 \times 4$ regions, and use a rotation-invariant *local binary pattern* (LBP) operator [35] (with operator setting $LBP_{8,1}^{riu2}$) to detect 10 distinct binary patterns. Thus an image can be represented by a 960-dimensional vector, where each dimension records the number of occurrences that a specific pattern is detected in the corresponding region. To achieve rotation invariant, the distance between two such vectors, say, $\mathbf{x}_i$ and $\mathbf{x}_j$, is the minimal one among the 24 values computed from the distance function $1 - \text{sum}(\min(\mathbf{x}_i, \mathbf{x}_j))/\text{sum}(\max(\mathbf{x}_i, \mathbf{x}_j))$ by circularly shifting the starting radial axis for $\mathbf{x}_j$. Clearly, the base kernel is constructed to deal with variations resulting from rotations.

Table 2. The performances of cluster ensembles and our approach in different settings

method	kernel(s)	dataset (number of classes)				
		All (68)	Lighting (17)	Rotation (17)	Occlusion (17)	Profile (17)
Ours	DeLight	40.2 / 0.628	**91.4** / **0.974**	21.0 / 0.435	25.5 / 0.508	23.0 / 0.487
	LBP	**47.3** / **0.672**	71.1 / 0.886	**59.9** / **0.744**	30.0 / 0.500	**28.2** / **0.512**
	RsLTS	39.3 / 0.647	35.4 / 0.518	32.9 / 0.495	**61.4** / **0.757**	27.6 / 0.492
	RsL2	31.6 / 0.628	50.9 / 0.685	27.6 / 0.464	19.5 / 0.352	28.4 / 0.509

CE	All	55.4 / 0.746	92.6 / 0.975	43.8 / 0.657	55.4 / 0.695	29.8 / 0.535
Ours	All	**61.9** / **0.822**	**93.6** / **0.985**	**57.8** / **0.730**	**64.8** / **0.781**	**31.6** / **0.554**

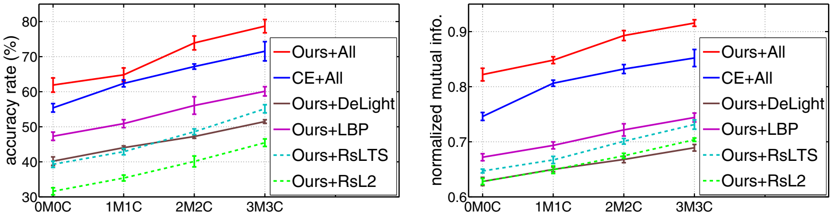


Fig. 5. The performances of cluster ensembles and our approach w.r.t. different amounts of **m**ust-links and **c**annot-links per subject and different settings of kernel(s)

- **RsL2**. Each sample is represented by its pixel intensities in raster scan order. Euclidean (L^2) distance is used to correlate two images.
- **RsLTS**. The same as RsL2 except that the distance function is now based on the *least trimmed squares* (LTS) with 20% outliers allowed. The kernel is designed to take account of the partial occlusions in face images.

Quantitative Results. We report the performances of applying our approach to the four kernels one by one in the third column of Table 2 (**top**). Besides, we also record the performances with respect to each of the four groups in the last four columns of the same table. (Each group is named according to the type of its intraclass variation.) Note that each result in the last four columns is computed by considering only the data in the corresponding group. No additional clustering is performed. As expected, each of the four kernels generally yields good performances in dealing with a specific kind of intraclass variations. For example, the kernel DeLight achieves a satisfactory result for subjects in the Lighting group, while LBP and RsLTS yield acceptable outcomes in Rotation and Occlusion groups respectively. However, none of them is good enough for dealing with the whole dataset. Still the results reveal that if we could choose proper features for each subject, it would lead to substantial improvements.

To verify the point, we apply the proposed clustering technique to the four kernels simultaneously, and compare it with cluster ensembles, which is used to merge the four clustering results derived by implementing our approach with single kernel. In Table 2 (**bottom**), it shows that using multiple kernels in our approach can achieve remarkable improvements over the best result obtained from using a single kernel (i.e. LBP), and also significantly outperforms the foregoing setting for cluster ensembles.

Indeed performing clustering with this dataset is hard, due to the large subject number and the extensive intraclass variations. We thus randomly generate one **m**ust-link and one **c**annot-link for each subject, and denote the setting of semisupervised clustering as `1M1C`. Analogously, we also have `0M0C` (i.e. unsupervised), `2M2C` and `3M3C`. Combining different amounts of pairwise constraints and different settings of kernel(s), the performances with respect to ACC and NMI of our approach are shown in Fig. 5. It is clear that by introducing only a few constraints, our approach can achieve considerable gains in performance.

6 Conclusion

We have presented an effective approach to clustering complex data that considers cluster-dependent feature selection and multiple feature representations. Specifically, we incorporate the supervised training processes of cluster-specific classifiers into the unsupervised clustering procedure, cast them as a joint optimization problem, and develop an efficient technique to accomplish it. The proposed method is comprehensively evaluated with two challenging vision applications, coupled with a number of feature representations for the data. The promising experimental results further demonstrate its usefulness. In addition, our formulation provides a new way of extending the multiple kernel learning framework, which is typically used in tackling supervised-learning problems, to address unsupervised and semisupervised applications. This aspect of generalization introduces a new frontier of applying multiple kernel learning to handling the ever-increasingly complex vision tasks.

Acknowledgments. We want to thank the anonymous reviewers for their valuable comments. This work is supported in part by grant 97-2221-E-001-019-MY3.

References

1. Dueck, D., Frey, B.: Non-metric affinity propagation for unsupervised image categorization. In: ICCV (2007)
2. Tuzel, O., Porikli, F., Meer, P.: Kernel methods forweakly supervised mean shift clustering. In: ICCV (2009)
3. Shi, J., Malik, J.: Normalized cuts and image segmentation. TPAMI (2000)
4. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. TPAMI (2002)
5. Roth, V., Lange, T.: Feature selection in clustering problems. In: NIPS (2003)
6. Ye, J., Zhao, Z., Wu, M.: Discriminative k-means for clustering. In: NIPS (2007)
7. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. JMLR (2004)
8. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS (2001)

9. Frey, B., Dueck, D.: Clustering by passing messages between data points. Science (2007)
10. Cheng, H., Hua, K., Vu, K.: Constrained locally weighted clustering. In: VLDB (2008)
11. Domeniconi, C., Al-Razgan, M.: Weighted cluster ensembles: Methods and analysis. TKDD (2009)
12. Berg, A., Malik, J.: Geometric blur for template matching. In: CVPR (2001)
13. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
14. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
15. Bosch, A., Zisserman, A., Muñoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR (2007)
16. Xu, L., Neufeld, J., Larson, B., Schuurmans, D.: Maximum margin clustering. In: NIPS (2004)
17. Zhao, B., Wang, F., Zhang, C.: Efficient multiclass maximum margin clustering. In: ICML (2008)
18. Strehl, A., Ghosh, J.: Cluster ensembles – A knowledge reuse framework for combining multiple partitions. JMLR (2002)
19. Fred, A., Jain, A.: Combining multiple clusterings using evidence accumulation. TPAMI (2005)
20. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. In: NIPS (2002)
21. Mutch, J., Lowe, D.: Multiclass object recognition with sparse, localized features. In: CVPR (2006)
22. Zhang, H., Berg, A., Maire, M., Malik, J.: SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: CVPR (2006)
23. Lin, Y.-Y., Liu, T.-L., Fuh, C.-S.: Local ensemble kernel learning for object category recognition. In: CVPR (2007)
24. Lin, Y.-Y., Tsai, J.-F., Liu, T.-L.: Efficient discriminative local learning for object recognition. In: ICCV (2009)
25. Moghaddam, B., Shakhnarovich, G.: Boosted dyadic kernel discriminants. In: NIPS (2002)
26. Collins, M., Schapire, R., Singer, Y.: Logistic regression, AdaBoost and Bregman distances. ML (2002)
27. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Annals of Statistics (2000)
28. Wolsey, L.: Integer Programming. John Wiley & Sons, Chichester (1998)
29. The MOSEK Optimization Software, http://www.mosek.com/index.html
30. 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)
31. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: CVPR (2007)
32. Cox, T., Cox, M.: Multidimentional Scaling. Chapman & Hall, London (1994)
33. Sim, T., Baker, S., Bsat, M.: The CMU pose, illumination, and expression database. TPAMI (2005)
34. Gross, R., Brajovic, V.: An image preprocessing algorithm for illumination invariant face recognition. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, Springer, Heidelberg (2003)
35. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. TPAMI (2002)

On Parameter Learning in CRF-Based Approaches to Object Class Image Segmentation

Sebastian Nowozin[1], Peter V. Gehler[2], and Christoph H. Lampert[3]

[1] Microsoft Research Cambridge, UK
[2] ETH Zurich, Switzerland
[3] Institute of Science and Technology, Austria

Abstract. Recent progress in per-pixel object class labeling of natural images can be attributed to the use of multiple types of image features and sound statistical learning approaches. Within the latter, Conditional Random Fields (CRF) are prominently used for their ability to represent interactions between random variables. Despite their popularity in computer vision, *parameter learning* for CRFs has remained difficult, popular approaches being cross-validation and *piecewise training*.

In this work, we propose a simple yet expressive tree-structured CRF based on a recent hierarchical image segmentation method. Our model combines and weights multiple image features within a hierarchical representation and allows simple and efficient globally-optimal learning of $\approx 10^5$ parameters. The tractability of our model allows us to pose and answer some of the open questions regarding parameter learning applying to CRF-based approaches. The key findings for learning CRF models are, from the obvious to the surprising, i) multiple image features always help, ii) the limiting dimension with respect to current models is the amount of training data, iii) piecewise training is competitive, iv) current methods for max-margin training fail for models with many parameters.

1 Introduction

Computer vision increasingly addresses high-level vision tasks such as scene understanding, object class image segmentation, and class-level object recognition. Two drivers of this development have been the abundance of digital images and the use of statistical machine learning models. Yet, it remains unclear what classes of models are suited best to these tasks. *Random field models* [1,2] have found many applications due to their ability to concisely express dependencies between multiple random variables, making them attractive for many high-level vision tasks. *Parameter learning* in these rich models is essential to find from a large set of possible candidates the model instance that best explains the observed data and generalizes to unseen data. Despite the importance of parameter learning, current applications of random fields in computer vision sidestep many issues, making assumptions that are intuitive, but largely heuristic. The reason for this gap between principled modeling and use of heuristics is the intractability of many random field models, which makes it necessary use approximations.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 98–111, 2010.

To shed light on the currently used practices we take the task of object class image segmentation and propose a simple, yet expressive hierarchical multi-scale CRF model in which parameter learning can be analyzed in isolation.

In our model, parameter learning *is* tractable, allowing us to experimentally address the following open questions regarding conditional random fields for object class image segmentation: 1. What is the effect of combining multiple image features on the resulting model performance? 2. How does the size of the training set and the accuracy of optimizing the training objective influence the resulting performance? 3. Is it better to learn the models part-by-part (*piecewise*) or jointly? 4. Does maximum margin training offer an advantage over maximum likelihood estimation?

Outline. We first describe random fields in Section 2. In Section 3 we discuss the current computer vision literature on parameter learning in CRFs. Our novel model is introduced in Section 4 and we report experiments in Section 5.

2 Learning Random Fields

In this section we review basic results about random field models, factor design and define the problems that need to be solved to perform prediction and parameter learning.

2.1 Random Field Models and Factor Graphs

Discrete random field models, also known as Markov networks, are a popular model to describe interacting variables [2]. In particular we will focus on *conditional random fields* (CRF) [3,4]. For a set $Y = \{Y_1, \dots, Y_V\}$ of random variables, each taking values in a label set $\mathcal{Y} = \{1, \dots, C\}$, a set of observation variables $X = \{X_1, \dots, X_W\}$, and a parameter vector $\boldsymbol{w} \in \mathbb{R}^D$, a conditional random field specifies a probability distribution as

$$p(Y = \boldsymbol{y} | X = \boldsymbol{x}, \boldsymbol{w}) = \frac{1}{Z(\boldsymbol{x}, \boldsymbol{w})} \exp(-E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})), \tag{1}$$

where $E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w})$ is an *energy function* and $Z(\boldsymbol{x}, \boldsymbol{w}) = \sum_{\boldsymbol{y} \in \mathcal{Y}^V} \exp(-E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}))$ is a normalizing constant known as *partition function* [1]. The energy function is specified in terms of *log-potential functions*, also known as *log-factors*. Let $\mathcal{F} \subseteq 2^V \times 2^W$ be a set of subsets of the variables. Then $\mathcal{F}$ specifies a factorization of (1), or equivalently an additive decomposition of the energy function as

$$E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}) = \sum_{F \in \mathcal{F}} E_F(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w}), \tag{2}$$

where $\boldsymbol{y}_F$ and $\boldsymbol{x}_F$ denote the restrictions of Y and X to the elements appearing in F, respectively. The energy function E_F operates only on the variables appearing in the set F.

The factorization is often given implicitly by means of an undirected graphical model [1]. For all practical purposes, it is more convenient to directly specify $\mathcal{F}$ used in (2) in terms of a *factor graph* [5]. For each element $F \in \mathcal{F}$, a factor graph contains a *factor node* (drawn as ■), which is connected to all *variable nodes* (drawn as ○) that are members of F. The factor graph compactly defines $\mathcal{F}$ in (2). An example is shown in Figure 2 (page 103).

In order to fully specify the random field model, the form of the individual terms $E_F(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w})$ in the summation (2) has to be defined. Each term corresponds to one factor F in the factor graph and specifies the local interactions between a small set of random variables. In practice the different factors have one of a few different roles such as incorporating observations into the model or enforcing a consistent labeling of the variables. Therefore, *clique templates* [4] (also known as *parameter tying*) are used, replicating parameters across groups of factors with the same purpose. We let $T = \{1, \dots, |T|\}$ denote a small set of different factor purposes and split the parameter vector as $\boldsymbol{w} = (\boldsymbol{w}_1^\top, \dots, \boldsymbol{w}_{|T|}^\top)^\top$, then the energy of each factor can be written as $E_F^{t(F)}(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w}_{t(F)})$, where $t(F)$ is the type of the factor. As an additional notation, let $\boldsymbol{\mu}_F \in \{0,1\}^{\mathcal{Y}^F}$ be a set of binary indicator variables indexed by $\boldsymbol{y}_F \in \mathcal{Y}^F$ and let $\mu_F(\boldsymbol{y}_F) \in \{0,1\}$ be one if $Y_F = \boldsymbol{y}_F$, zero otherwise. Let the scalar $\theta_{F,\boldsymbol{y}_F}(\boldsymbol{x}_F, \boldsymbol{w}_{t(F)}) = E_F^{t(F)}(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w}_{t(F)})$ be the evaluated energy when $Y_F = \boldsymbol{y}_F$. By suitably concatenating all μ_F, θ_F we can rewrite the energy (2) as the inner product $\langle \boldsymbol{\theta}(\boldsymbol{x}, \boldsymbol{w}), \boldsymbol{\mu} \rangle$. Because this form is linear, the distribution (1) is an *exponential family distribution* [1] with *sufficient statistics* $\boldsymbol{\mu}$ and so called *canonical parameters* $\boldsymbol{\theta}(\boldsymbol{x}, \boldsymbol{w})$.

What is left to do is to give the form of the *feature function* $\boldsymbol{\theta}_F(\boldsymbol{x}_F, \boldsymbol{w}_{t(F)})$ for all factor types $t(F) \in T$. As we will see below an important requirement for efficient parameter learning is that the energy function is *linear* in $\boldsymbol{w}$. The energy function $E_F^{t(f)}$ related to one factor F is already a linear function in the output of the feature function $\boldsymbol{\theta}_F : \mathcal{X}^F \times \mathbb{R}^{D_{t(F)}} \to \mathbb{R}^{\mathcal{Y}^F}$. Therefore, the energy will only be linear in $\boldsymbol{w}$ if we make the feature function also a linear function in its second argument $\boldsymbol{w}$. To this end, we will write $\boldsymbol{\theta}_F(\boldsymbol{x}_F, \boldsymbol{w}_{t(F)}) = H_F^{t(F)}(\boldsymbol{x}_F)\boldsymbol{w}_{t(F)}$, where $H_F^{t(F)}(\boldsymbol{x}_F)$ is a linear map from $\mathbb{R}^{D_{t(F)}}$ onto $\mathbb{R}^{\mathcal{Y}^F}$, mapping the parameters $\boldsymbol{w}_{t(F)}$ to energies. Due to the identity $E_F^{t(F)}(\boldsymbol{y}_F; \boldsymbol{x}_F, \boldsymbol{w}_{t(F)}) = \langle \boldsymbol{\theta}_F(\boldsymbol{x}_F, \boldsymbol{w}_{t(F)}), \mu_F(\boldsymbol{y}_F) \rangle = \langle H_F^{t(F)}(\boldsymbol{x}_F)\boldsymbol{w}_{t(F)}, \mu_F(\boldsymbol{y}_F) \rangle = \langle \boldsymbol{w}_{t(F)}, \phi(\boldsymbol{x}_F, \boldsymbol{y}_F) \rangle$ we can make explicit the linearity in *both* $\boldsymbol{w}_{t(F)}$ and $\mu_F(\boldsymbol{y}_F)$, where $\phi(\boldsymbol{x}_F, \boldsymbol{y}_F) = \mu_F(\boldsymbol{y}_F) H_F^{t(F)}(\boldsymbol{x}_F)$ is also known as joint feature map in the CRF literature. *Why is this important?* Linearity in $\boldsymbol{w}$ leads to convex learning problems (so that local optimality implies global optimality); linearity in μ leads to an exponential family distribution.

2.2 Inference Problems

The random field model is now fully specified and we can consider inference and learning tasks. The two tasks of our interest are the test-time prediction task, labeling an image with a likely segmentation, and the parameter learning task in which we have fully annotated training data and want to estimate a good parameter vector $\boldsymbol{w}$. In computer vision, predictions are most often made by solving an energy minimization problem as follows.

Problem 1 (MAP-MRF Labeling Problem). Given an observation $\boldsymbol{x}$ and a parameter vector $\boldsymbol{w}$, find the $\boldsymbol{y} \in \mathcal{Y}^V$ that maximizes the aposteriori probability $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w})$, that is, solve

$$\boldsymbol{y}^* = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}^V} p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{w}) = \operatorname*{argmin}_{\boldsymbol{y} \in \mathcal{Y}^V} E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}).$$

For general factor graphs this problem is NP-hard [2].

To address the parameter learning problem we use the principle of *maximum likelihood* to find a point estimate for $\boldsymbol{w}$. We now define the estimation problem but in Section 5.4 make connections to maximum-margin procedures.

Problem 2 (Regularized CML Estimation (CMLE)). Given a set of N fully observed independent and identically distributed (iid) instances $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1,\dots,N}$ and given a prior $p(\boldsymbol{w})$ over $\mathbb{R}^D$, find $\boldsymbol{w}^* \in \mathbb{R}^D$ with maximum regularized conditional likelihood, that is, solve

$$\begin{aligned} \boldsymbol{w}^* &= \operatorname*{argmax}_{\boldsymbol{w} \in \mathbb{R}^D} \; p(\boldsymbol{w}) \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{x}_n, \boldsymbol{w}) \\ &= \operatorname*{argmax}_{\boldsymbol{w} \in \mathbb{R}^D} \; \left[\frac{1}{N} \log p(\boldsymbol{w}) - \frac{1}{N} \sum_{n=1}^{N} \left(E(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{w}) + \log Z(\boldsymbol{x}_n, \boldsymbol{w}) \right) \right]. \end{aligned} \quad (3)$$

From the fact that $E(\boldsymbol{y}_n; \boldsymbol{x}_n, \boldsymbol{w})$ is a linear function in $\boldsymbol{w}$ it follows [2, section 20.3.2] that the log-likelihood (3) is a concave differentiable function in $\boldsymbol{w}$ and therefore $\boldsymbol{w}^*$ can be found using gradient descent. In the case that $\log p(\boldsymbol{w})$ is strictly concave in $\boldsymbol{w}$, (3) has a unique maximizer. Despite this, it is hard to solve Problem 2 for general factor graphs. The reason is that evaluating (3) for a given $\boldsymbol{w}$ requires computing the partition function $Z(\boldsymbol{x}_n, \boldsymbol{w})$ for each sample, a task involving summation of an exponential number of terms.

In our model presented in Section 4 we therefore consider *tree-structured* factor graphs. These are by definition acyclic and the partition function can be computed efficiently by rearranging the exponential number of terms as a recursion along the tree. This algorithm for computing $\log Z(\boldsymbol{x}_n, \boldsymbol{w})$ and $\nabla_{\boldsymbol{w}} \log Z(\boldsymbol{x}_n, \boldsymbol{w})$ is known as *sum-product algorithm* [5]. Likewise, for tree-structured factor graphs we can efficiently solve the MAP-MRF problem by the *max-product* algorithm.

3 Literature Review

Literature on CRF-based object class segmentation. CRF-based approaches to object class image segmentation can be distinguished by what kind of factors they use (unary, pairwise, higher-order factors), the model capacity, that is, how many free parameters they have, how the model structure is defined (pixel grid, superpixels, etc.) and how the parameter learning is performed.

Regarding the *representation*, the main lines are pixel- or pixel-blocks based approaches [6,7,8,9,10,11], superpixel-based representations [12,13,14], superpixel hierarchies [15,16], and hybrid (both pixels and superpixels) representations [17,18,19].

For parameter learning, most works cited before use a form of piecewise training or cross validation on one to five hand-chosen parameters. Models in which joint parameter learning is performed are rare and often use an approximation, such as loopy BP in [14,11], pseudolikelihood in [9], and contrastive divergence in [10]. Pincipled max-margin learning is performed in [6,12,19].

Literature on comparing learning methods for CRFs. Because we address the effect of different parameter learning methods, let us summarize existing comparisons of parameter learning methods. Kumar et al. [20] compare a large number of approximate CRF learning methods on a synthetic binary low-level vision task with four parameters. Similar experiments on the same dataset have been done by Korc and Förstner [21]. The excellent study of Parise and Welling [22] compares learning methods for generative binary non-vision MRF models with fixed, non-replicated structure. Finley and Joachims [23] compare learning methods for intractable MRF models advocating max-margin learning on relaxations.

Importance of tree-based models. Many early models for low-level vision were based on tree-structured generative MRFs (for an extensive survey see [24]), where the structure of the tree is fixed and simple, such as a quad-tree on a 2D grid. The use of tree-structured models for high-level vision tasks is much less common. One reason is that we now have efficient algorithms for MAP inference for certain potential functions for graphs of arbitrary structure. This offers more modeling freedom on the graph structure while restricting the potential function class. But recently there seems to be reconsideration of tree-based hierarchical models for high-level vision tasks where the tree structure is *adapted to the image content* [15,16,25]. Infact, even the more complex hybrid models listed above [17,18,19] base their multi-scale structure on a hierarchical tree of superpixels. Whereas obviously tree-based models are a restricted model class, the ability to learn arbitrary potential functions and the adapted nature of the tree structure to the image content offer drastic improvements over the early tree-based models considered before [24].

Lim et al. [25] is closest to our approach: a segmentation hierarchy is used as a multi-scale model for object class image segmentation. For each image region a linear classifier is learned, using features derived from the hierarchy. The main drawbacks of the otherwise sensible approach are the lack of pairwise interactions between image regions and the use of an adhoc test-time prediction function.

4 Model

We now define a tree-structured model for object class image segmentation. The model is naturally multi-scale and adapted to the image content. Due to

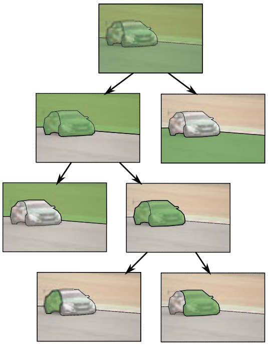


Fig. 1. Illustration of a hierarchical UCM segmentation. The hierarchy ranges from a superpixel partitioning at the leaf level to the entire image at the root. Each node's image region is shaded in green. (Figure best viewed in color.)

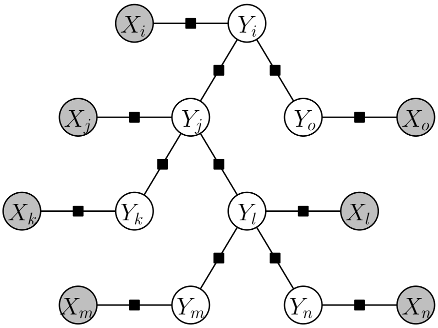


Fig. 2. Tree-structured factor graph CRF induced by the hierarchical segmentation. Each shaded segment r in Figure 1 has an observation variable X_r (drawn shaded) and a class variable Y_r. Factors (drawn as ■) encode interactions.

its tree structure, test-time image labeling as well as joint parameter learning are tractable. The tractability allows us to answer for the first time important questions regarding modeling choices, such as: What is the required image granularity for object class image segmentation? How to parametrize and learn the factors? What limits the current model performance? Is joint parameter learning superior to piecewise training?

The model is based on the recent *ultrametric contour maps* (UCM) hierarchical segmentation method of Arbeláez [26]. We use the UCM segmentation to define a tree structured factor graph. The factors are then suitably parametrized such that parameter estimation from training data can be performed. This idea is illustrated in Figures 1 and 2. In Figure 1 we illustrate the output of the UCM method: a segmentation tree that recursively partitions the image into regions. The leaves of the segmentation tree form a superpixel segmentation of the image, whereas interior nodes represent larger image regions. Ideally object instances – such as the car in the Figure 1 – are eventually represented by a single interior node. We use the structure of the segmentation tree to define a factor graph as shown in Figure 2. The shaded nodes correspond to image information observed for each image region, whereas the white nodes represent the class variables to be predicted, one for each region. The factor nodes (drawn as ■) link both observation and class variables, as well as pairs of class variables.

Because the hierarchical model structure is based on the UCM segmentation, it is naturally adapted to the image content. Moreover, it is a multi-scale representation of the image [26]. Our factor-graph can concisely represent a probability distribution over all possible labelings.

In next three subsections we discuss the choice of superpixel granularity, how to parametrize factors and how to perform training and prediction.

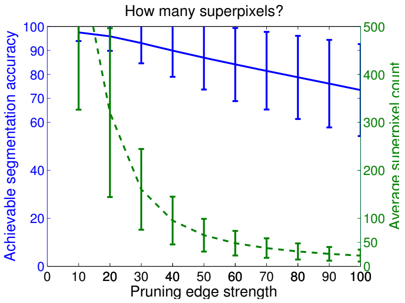


Fig. 3. Upper bound on the achievable VOC 2009 segmentation accuracy as a function of the preserved UCM edge strength. The left axis (solid, blue) shows the accuracy, the right axis (dashed, green) shows the mean number of superpixels per image. For each curve one unit standard deviations over the 749 training images is shown.

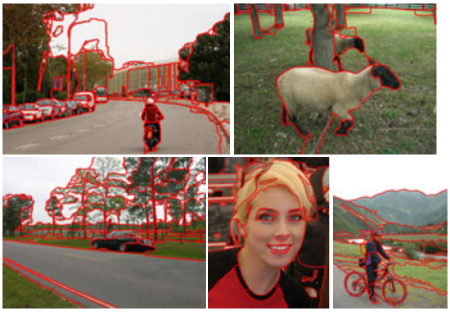


Fig. 4. Visualization of the superpixels of the hierarchical segmentation. Shown are examples from the VOC 2009 segmentation set, with the chosen edge pruning parameter of 40, leading to an average of ≈ 100 superpixels and ≈ 200 tree nodes per image.

4.1 Experiment: How Many Superpixels?

When using a fixed precomputed representation of the image such as superpixels, it is fair to ask how much representational power is lost in the process: because we associate one discrete random variable with each superpixel, an error on this representational level cannot be corrected later.

To determine this trade-off, we produce UCM segmentations using the code of Arbeláez [26] for the 749 images in the PASCAL VOC 2009 segmentation challenge [27]. By thresholding the obtained UCM maps at increasing values we obtain a set of successively coarser hierarchical segmentations. For each threshold we evaluate the maximum achievable accuracy if we could label all leaves of the segmentation tree knowing the ground truth pixel labeling.

The results are shown in Figure 3. Even with a relatively small average number of superpixels the segmentation accuracy is above 70%. While this number appears to be low, it can be put into perspective by recognizing that the currently best state-of-the-art segmentation models applied to the VOC 2009 data set – including non-CRF approaches and methods trained on substantially more training data – achieve 25 – 36% using the same evaluation measure [27]. Gould et al. [13] carried out a similar experiment on the MSRC and Sowerby data sets, and their results agree with our observations. For the following experiments we choose a pruning edge strength of 40, yielding an average of ≈ 100 superpixels per image and a maximum achievable accuracy of ≈ 90%. For this choice, Figure 4 shows typical example segmentations for the VOC 2009 images. For each image shown, the achievable accuracy is between 89.7% and 90.3%.

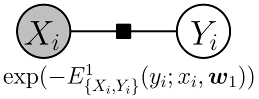


Fig. 5. Unary energy $E^1_{\{X_i,Y_i\}}(y_i;x_i,\boldsymbol{w}_1)$

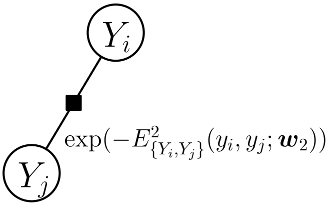


Fig. 6. Pairwise data-independent energy $E^2_{\{Y_i,Y_j\}}(y_i,y_j;\boldsymbol{w}_2)$

4.2 Features and Factors

We now describe how to parametrize the factors used in our model, starting with the unary observation factors.

Unary observation factors. The most important factors, the unary observation factors, describe the interaction between the image content and the variables of interest. We use multiple image features representing appearance statistics based on shape, color and texture to span a rich feature space describing an image region. As shown in Figure 5 and described in Section 2.1, the unary energy takes the following general form

$$E^1_{\{X_i,Y_i\}}(y_i;x_i,\boldsymbol{w}_1) = \langle \theta^1_{\{X_i\}}(x_i,\boldsymbol{w}_1), \boldsymbol{\mu}_{\{Y_i\}}\rangle = \langle H^1_{\{X_i\}}(x_i)\boldsymbol{w}_1, \boldsymbol{\mu}_{\{Y_i\}}\rangle.$$

Within this form, we define $H^1_{\{X_i\}}(x_i)$ as the concatenation of multiple image features. In particular, we define $H^1_{\{X_i\}}(x_i) = (f_{\mathrm{SIFT}}(x_i), f_{\mathrm{QHOG}}(x_i), f_{\mathrm{QPHOG}}(x_i), f_{\mathrm{STF}}(x_i))^\top$, where each f_a is an image feature related to the image region associated with X_i. As image features $f_a : \mathcal{X} \to \mathbb{R}^{D_a}$ we use the following: $a \in A = \{\mathrm{SIFT}, \mathrm{QHOG}, \mathrm{QPHOG}, \mathrm{STF}\}$, where SIFT are normalized bag-of-words histograms of quantized scale-invariant feature points ($D_{\mathrm{SIFT}} = 512$). The QHOG features are soft-quantized histogram of oriented gradient vectors of the image content within a bounding box of the image region X_i ($D_{\mathrm{QHOG}} = 512$). Similarly, the QPHOG features are soft-quantized pyramid of histogram of oriented gradient features of the black-and-white mask describing the image region X_i ($D_{\mathrm{QPHOG}} = 512$). The STF features are normalized histograms of semantic texton forest responses within the image regions [28] ($D_{\mathrm{STF}} = 2024$). For the above features $\boldsymbol{w}_1 \in \mathbb{R}^{D\times\mathcal{Y}}$, where $D = \sum_{a\in A} D_a = 3560$, such that $\boldsymbol{w}_1$ in total has $C \cdot D$ elements. The SIFT and STF features model general image statistics in the region X_i, whereas the QHOG and QPHOG features are responses to a template of shapes and appearances obtained by clustering the training data. If the hierarchical segmentation contains a region that describes an object instance, we hope to obtain a high response in these features. More details regarding the features used are available in the supplementary materials.

Data-independent pairwise factor. The pairwise factor shown in Figure 6 models the interaction of labels (Y_i, Y_j), where i and j form a children-parent pair in the hierarchical segmentation. If for example, y_i is labeled with a class, then y_j is likely to be labeled with the same class. We consider two possible energies of the form shown in Figure 6, the first one having the commonly used form

$$E^{2,P}_{\{Y_i,Y_i\}}(y_i, y_j; \boldsymbol{w}_{2,P}) = \langle \boldsymbol{w}_{2,P}, \boldsymbol{\mu}_{\{Y_i,Y_j\}}\rangle,$$

where we set $H^{2,P}_{\emptyset}$ to the identity operator, such that $\boldsymbol{w}_{2,P} \in \mathbb{R}^{\mathcal{Y}\times\mathcal{Y}}$ is a simple table of energy values for each possible configuration (y_i, y_j). This setting contains the *generalized Potts* model for pairwise interactions as a special case. Note that unlike in random fields defined on a pixel grid we do not assume regular/submodular/attractive energies and also do not require symmetry of the matrix $\boldsymbol{w}_{2,P}$. This is important because the role of child and parent variable is known; for instance, a children-parent region labeling of (“*car*″, “*background*″) is more likely to occur than (“*background*″, “*car*″). We consider a second type of energy as a baseline: the constant energy, making all variables $Y_i \in Y$ independent. We define it as parameter-less energy $E^{2,\text{constant}}_{\{Y_i,Y_i\}}(y_i, y_j) = 0$.

4.3 Training and Testing

Training. For solving Problem 2 we use the LBFGS method from the minFunc package of Mark Schmidt[1] and for the inference we use libDAI [29]. In the experiments we state the number of LBFGS iterations used.

For each instance in the training set, we set as ground truth label $\boldsymbol{y} \in \mathcal{Y}^V$ not the discrete labeling vector but the actual distribution $\boldsymbol{\mu}_V \in [0, 1]^{\mathcal{Y}^V}$ of pixel labels within each image region. This faithfully represents the actual ground truth information and reduces to the discrete label case if all pixels within a region have the same label.

For the prior distribution over the parameters $\boldsymbol{w}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{w}_3$ we choose a multivariate Normal distribution, such that $p(\boldsymbol{w}_1; \sigma) = \mathcal{N}(\mathbf{0}, \sigma^2 I)$, $p(\boldsymbol{w}_2; \tau) = \mathcal{N}(\mathbf{0}, \tau^2 I)$, and $p(\boldsymbol{w}_3; \tau) = \mathcal{N}(\mathbf{0}, \tau^2 I)$. This leads to two hyper-parameters (σ, τ) to be selected by model selection.

Test-time prediction. For a given test image $\boldsymbol{x}$ and trained model $\boldsymbol{w}^*$ we find the MAP labeling $\boldsymbol{y}^* = \operatorname{argmin}_{\boldsymbol{y}\in\mathcal{Y}^V} E(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{w}^*)$. In $\boldsymbol{y}^*$ we have one label per hierarchical image region, whereas the original task is to label each pixel with a unique label. It could therefore be the case that two region labels contradict each other in their pixel assignments. We could enforce consistency by assigning infinite energies to children-parent labelings of the form (y_c, y_p) where $y_c \neq y_p$ and $y_p \neq$ “background”. However, inconsistent labelings are absent in the training data and hence the model parameters are already chosen such that inconsistent labelings are unlikely. Experiments confirm this: on holdout data less than 0.7% of all children-parent links are inconsistently labeled. Therefore, for making test-time predictions we label each pixel with the label of its largest region that is not assigned a background label. In case no such region exist, the background label is assigned.

[1] `http://people.cs.ubc.ca/~schmidtm/Software/minFunc.html`

Unary features	seg-val	Train time	D
SIFT	6.13%	22h01m	11,193
QHOG	8.40%	19h30m	11,193
QPHOG	7.35%	36h03m	11,193
STF	6.76%	39h36m	42,945
QHOG,QPHOG	10.92%	24h35m	21,945
SIFT,QHOG,QPHOG	14.54%	26h17m	32,697
all features	15.04%	41h39m	75,201

Fig. 7. The result of feature combination at the unary factors

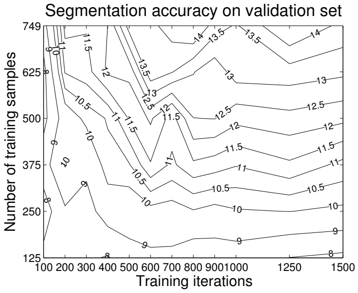


Fig. 8. VOC 2009 validation accuracy as the training set size and number of LBFGS iterations vary

5 Experiments

Throughout the experiments section we use the PASCAL VOC 2009 dataset [27]. The segmentation challenge contains 1499 annotated images (749 training, 750 validation), labeling each pixel with either "background" or one of 20 object classes, such as car, person, bottle, etc. The dataset is widely accepted to be difficult and realistic. We report the official PASCAL VOC2009 segmentation challenge performance measure [27] which is the average over 20 object classes of the intersection/union metric. Except for the final challenge evaluation, all models are trained on the segmentation `train` set (749 images) and we report the performance on the segmentation `val` set (750 images).

5.1 Quantifying the Effect of Feature Combination

For high level vision tasks such as object recognition, image classification and segmentation it is now well accepted that the combination of multiple image features is essential for obtaining good performance [30]. On the other hand, the use of multiple image features leads to models with many parameters and thus a possibly higher estimation error or overfitting.

We verify our model by evaluating the performance of individual features versus their combination. We do not perform model selection and fix $\sigma = 1000$, $\tau = 1000$. We train using 700 LBFGS iterations on the segmentation `train` set and report the performance on the segmentation `val` set.

Table 7 reports the results. As expected, combining multiple features is essential to obtain reasonable performance levels. Combining the three SIFT, QHOG, and QPHOG features doubles the performance of each individual one.

Moreover, we find that adding any reasonable image feature never decreased the performance. This shows that our model can combine multiple image features in a robust way, and a high dimensionality D of the parameter space does not lead to overfitting. We submitted a model trained using all features on the segmentation `trainval` dataset to the VOC2009 challenge. Some good and erroneous segmentations of this model are shown in Figure 9. A discussion of the

Table 1. VOC 2009 segmentation accuracy on validation set for the best performing unary-only model, the best piecewise-trained model, and the jointly-trained model

Model	seg-val	Training time
Unary only,	9.98%	2h15m
Piecewise, Potts	14.50%	(2h15)+10h28m
Joint	14.54%	26h17m

challenge results and how other CRF-based approaches fared can be found in the supplementary materials.

5.2 Training Set Size and Learning Tradeoff

For any machine learning model, there exists a tradeoff between the expressivity of the model, the scalability to large training sets and the feasibility of optimization [31]. This experiment determines what the limiting dimension of our model is: the model class, the training set size or the training procedure. We train using the SIFT, QHOG and QPHOG features as we vary the training set size and the LBFGS iterations.[2] We evaluate each model on the validation set.

The results are shown in Figure 8. Up to about 600 LBFGS iterations the performance increases with more iterations. This is true for all training set sizes, but eventually the performance levels off when enough iterations are used. Uniformly the performance increases when more training samples are used. This indicates that the model has enough expressive power to achieve high accuracy but is currently limited by the small amount of annotated training data.

5.3 Piecewise versus Joint Parameter Learning

Piecewise training [32] is a two-step procedure where in the first step the factor graph is decomposed into disjoint subgraphs and each subgraph is trained individually. In the second step the learned weights are fixed and the factors joining the subgraphs are jointly trained. Piecewise training is an effective approximation and has been extensively used. Despite this, it has so far not been studied how much is lost compared to joint training of the model.

To quantify what is lost we use CMLE training with 700 iterations on the SIFT, QHOG and QPHOG features. We first produce a model without pairwise potentials (Unary only) by selecting $\sigma \in \{10, 100, 1000\}$ for best performance on the validation set. The learned parameters are fixed and the pairwise energy $E^{2,P}$ is used to retrain, selecting $\tau \in \{10, 100, 1000\}$ for best performance on the validation set (Piecewise, Potts). The canonical competitor to this piecewise-trained model is a jointly trained model (Joint), with $\sigma, \tau = 1000$ fixed.

[2] The training set size is within $\{125, 250, 375, 500, 625, 749\}$, the training iterations within $\{100, 200, \ldots, 1000, 1250, 1500\}$.

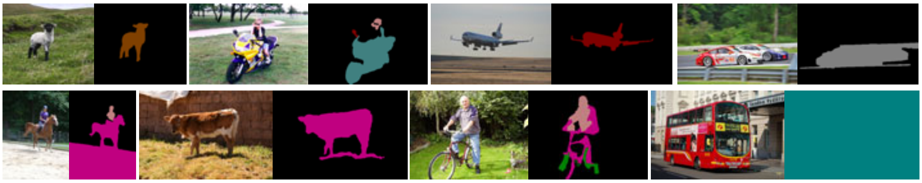


Fig. 9. VOC test predictions. Top: success, bottom row: typical failures (background labeled, wrong label, clutter, entire image labeled).

The results are shown in Table 1. The training time is reduced, but it is surprising that the loss due to piecewise training of the unary energies is negligible.

5.4 Maximum Likelihood versus Max-Margin

So far we have estimated the parameters of our models using the principle of maximum likelihood. An alternative method to estimate $\boldsymbol{w}$ from training data is the *maximum margin principle* [33], recently applied to learn structured prediction models in computer vision [34,6,12,19] using the structured SVM formulation.

We use the standard margin-rescaling structured SVM formulation [33], which we describe in the supplementary materials. The use of the structured SVM entails the choice of a semi-metric $\Delta(\boldsymbol{y}_n, \boldsymbol{y})$ and the parameter C_{svm}. For $\Delta : \mathcal{Y}^V \times \mathcal{Y}^V \to \mathbb{R}_+$ we choose the same function as [12], weighting the regions by their relative sizes, something that is not possible in standard CMLE training.

We evaluate the structured SVM against CMLE with 500 LBFGS iterations. For the structured SVM we use the popular cutting plane training procedure [33], solved using the Mosek QP solver. We evaluate $C_{\text{svm}} \in \{10^{-5}, 10^{-4}, \dots, 1\}$ for the structured SVM model and $(\sigma, \tau) \in \{100, 1000\} \times \{1, 10, 100, 1000\}$ for CMLE and report the best achieved performance on the validation set using the SIFT,QHOG,QPHOG features and the data-independent pairwise Potts factor. For larger values of C_{svm} the cutting-plane training procedure failed; we describe this in detail in the supplementary materials.

The results shown in Table 2 show that the CMLE training procedure requires less time and outperforms the structured SVM model consistently. It is unclear and remains to be examined whether this is due to the failure of the structured SVM optimization procedure for large values of C_{svm} or because of an inferior estimator.

Table 2. Results of maximum likelihood training and structured support vector machine training. See main text for details.

	Accuracy CMLE	Training time CMLE	Accuracy SVM	Training time SVM
Potts	13.65%	24h11m	13.21%	165h10m

6 Conclusions and Future Work

We draw the following conclusions for the class of tree-structured/hierarchical CRF based approaches to object class image segmentation:

- Current CRF models are limited by the amount of training data and available image features; more of both consistently leads to better performance,
- Piecewise training of unary observation factors is competitive with joint training and reduces the required training time considerably,
- Max-margin training is not well-tested within computer vision; current methods are slow and unstable in case of many parameters.

This work provides recommendations for the tractable, tree-structured case on a popular high-level vision task. In the future we plan to provide a larger study examining whether our conclusions extend to general intractable CRF models learned using approximate inference. Additionally, we would like to examine other high-level data-driven vision tasks.

References

1. Lauritzen, S.L.: Graphical Models. Oxford University Press, Oxford (1996)
2. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge (2009)
3. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML (2001)
4. Sutton, C., McCallum, A.: An introduction to conditional random fields for relational learning. In: Introduction to Statistical Relational Learning. MIT Press, Cambridge (2007)
5. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47, 498–519 (2001)
6. Szummer, M., Kohli, P., Hoiem, D.: Learning CRFs using graph cuts. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 582–595. Springer, Heidelberg (2008)
7. Winn, J.M., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: CVPR (2006)
8. 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 81 (2007)
9. Kumar, S., Hebert, M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In: ICCV (2003)
10. He, X., Zemel, R.S., Carreira-Perpiñán, M.Á.: Multiscale conditional random fields for image labeling. In: CVPR (2004)
11. Schnitzspan, P., Fritz, M., Schiele, B.: Hierarchical support vector random fields: Joint training to combine local and global features. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 527–540. Springer, Heidelberg (2008)
12. Nowozin, S., Lampert, C.H.: Global connectivity potentials for random field models. In: CVPR (2009)

13. Gould, S., Rodgers, J., Cohen, D., Elidan, G., Koller, D.: Multi-class segmentation with relative location prior. IJCV 80, 300–316 (2008)
14. Batra, D., Sukthankar, R., Chen, T.: Learning class-specific affinities for image labelling. In: CVPR (2008)
15. Reynolds, J., Murphy, K.: Figure-ground segmentation using a hierarchical conditional random field. In: CRV (2007)
16. Plath, N., Toussaint, M., Nakajima, S.: Multi-class image segmentation using conditional random fields and global classification. In: ICML (2009)
17. Kohli, P., Ladický, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. In: CVPR (2008)
18. Ladický, L., Russell, C., Kohli, P.: Associative hierarchical crfs for object class image segmentation. In: ICCV (2009)
19. Munoz, D., Bagnell, J.A., Vandapel, N., Hebert, M.: Contextual classification with functional max-margin markov networks. In: CVPR (2009)
20. Kumar, S., August, J., Hebert, M.: Exploiting inference for approximate parameter learning in discriminative fields: An empirical study. In: Rangarajan, A., Vemuri, B.C., Yuille, A.L. (eds.) EMMCVPR 2005. LNCS, vol. 3757, pp. 153–168. Springer, Heidelberg (2005)
21. Korc, F., Förstner, W.: Approximate parameter learning in conditional random fields: An empirical investigation. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 11–20. Springer, Heidelberg (2008)
22. Parise, S., Welling, M.: Learning in Markov random fields: An empirical study. In: Joint Statistical Meeting, JSM 2005 (2005)
23. Finley, T., Joachims, T.: Training structural SVMs when exact inference is intractable. In: ICML (2008)
24. Willsky, A.S.: Multiresolution markov models for signal and image processing. Proceedings of the IEEE (2002)
25. Lim, J.J., Gu, C., Arbeláez, P., Malik, J.: Context by region ancestry. In: ICCV (2009)
26. Arbeláez, P.: Boundary extraction in natural images using ultrametric contour maps. In: Workshop on Perceptual Organization in Computer Vision (2006)
27. Everingham, M., Gool, L.V., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2009 Results (2009), http://www.pascal-network.org/challenges/VOC/voc2009/workshop/
28. Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)
29. Mooij, J.M.: libDAI: A free/open source C++ library for discrete approximate inference methods (2008), http://www.libdai.org/
30. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV (2009)
31. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. In: NIPS (2007)
32. Sutton, C.A., McCallum, A.: Piecewise training for undirected models. In: UAI (2005)
33. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. JMLR 6, 1453–1484 (2005)
34. Blaschko, M.B., Lampert, C.H.: Learning to localize objects with structured output regression. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 2–15. Springer, Heidelberg (2008)

Exploring the Identity Manifold: Constrained Operations in Face Space

Ankur Patel and William A.P. Smith

Department of Computer Science, The University of York
{ankur,wsmith}@cs.york.ac.uk

Abstract. In this paper, we constrain faces to points on a manifold within the parameter space of a linear statistical model. The manifold is the subspace of faces which have maximally likely distinctiveness and different points correspond to unique identities. We show how the tools of differential geometry can be used to replace linear operations such as warping and averaging with operations on the surface of this manifold. We use the manifold to develop a new method for fitting a statistical face shape model to data, which is both robust (avoids overfitting) and overcomes model dominance (is not susceptible to local minima close to the mean face). Our method outperforms a generic non-linear optimiser when fitting a dense 3D morphable face model to data.

1 Introduction

Linear statistical models have been used to model variation in 2D [5] and 3D [3] shape, appearance and texture. These models are generative in nature, in the sense that instances similar to those used to train the model can be computed from a low dimensional parameter vector. Faces have proven a particularly suitable class to model using such approaches.

Perhaps the best known statistical face model is the Active Appearance Model (AAM) [5] which combines a linear model of 2D shape and 2D appearance. Rather than model appearance, the 3D Morphable Model of Blanz and Vetter [3] models the shape and texture which give rise to appearance via a model of image formation. Xiao et al.[17] have used a 3D model in conjunction with a 2D appearance model to enforce geometric constraints on the 2D shape generated.

Applying these models to face processing tasks requires a means to fit the model to observed data. This data may take many forms, such as the appearance of a face in one [3,5,17] or more [1,6] images, a noisy and incomplete 3D scan [2] or the location of a sparse set of feature points in an image [8]. Often this fitting process is underconstrained, prone to converge on local minima and computationally expensive. For these reasons, there is strong motivation for developing additional constraints to reduce the search space of the fitting process.

The most common method for learning such models from data, Principal Components Analysis (PCA), is based on the assumption that faces form a Gaussian cloud in a high dimensional space. The principal axes of this cloud

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 112–125, 2010.

are estimated from a training sample, allowing any face to be approximated in terms of a small number of parameters.

Psychological results [16,11] have shown that this parameter space has an interesting perceptually-motivated interpretation: *identity* relates to direction in parameter space while *distinctiveness* is related to vector length (or equivalently distance from the mean). The reason for this is that increasing the length of a parameter vector simply exaggerates its differences from the average linearly, in other words its *features*, whereas rotating a parameter vector changes the *mix* of features present in the face. This is the justification for using angular difference in face space as a measure of dissimilarity for face recognition.

This decomposition also allows a useful probabilistic interpretation. Under the Gaussian assumption, each model parameter is independent and distributed according to a Gaussian distribution. This means that all faces lie on or near the surface of a hyper-ellipsoid in parameter space, with the probability density over the parameter vector lengths following a chi-square distribution. In other words, distinctiveness is subject to a statistical prior with the distinctiveness of most samples clustered around the expected length.

1.1 Contribution

In this paper, we use these observations to motivate a representation for faces which decomposes face appearance into identity and distinctiveness subspaces. We focus on statistical models of 3D face shape, though all of our results are equally applicable for any parametric face representation. We use ideas from differential geometry to develop tools which operate in the identity subspace, i.e. which retain constant distinctiveness. We provide empirical justification for constraining samples to have fixed distinctiveness, determined by the expected vector length.

We propose a new algorithm for fitting a statistical face model to data. Many such methods have been proposed, the details being dependent on the precise nature of the model and data. However, this inevitably involves a non-linear optimisation over the model parameters.

Examples include Cootes's [5] original algorithm for fitting AAMs to images which assumes that the relationship between error and optimal additive parameter updates is constant. Matthews and Baker's [9] inverse compositional algorithm avoided this assumption allowing faster and more robust convergence. In the domain of fitting 3D morphable models to single 2D images, Blanz and Vetter's [3] approach was to use a stochastic optimisation process in an analysis-by-synthesis framework in the hope of finding a global minimum. Careful initialisation and regularisation is required to obtain stable performance. Romdhani et al.[14] proposed an alternative approach which used additional features such as edges and specularities as part of the error term. The hope was to obtain a globally convex objective function, allowing local optimisation methods to arrive at the global optimum. All these approaches must trade off satisfaction of a model-based prior against quality of fit. To ensure robust performance, these approaches must favour the prior, resulting in model dominance.

Our approach operates via gradient descent on the manifold of equal distinctiveness. In other words, we solve for identity and assume distinctiveness takes its expected value. We show how the method naturally lends itself to a coarse-to-fine optimisation strategy and how the result avoids overfitting or local minima in which generic non-linear optimisers become stuck.

2 Statistical Modelling

Consider a sample of 3-dimensional face meshes which are in dense correspondence (i.e. the same point on every face has the same vertex index). The ith shape is represented by a vector of p vertices $\mathbf{s}_i = (x_1, y_1, z_1, \dots, x_p, y_p, z_p) \in \mathbb{R}^{3p}$. Given m such shape vectors, we use principal components analysis to obtain an orthogonal coordinate system spanned by the m eigenvectors P_i. Any shape vector $\mathbf{s}$ may now be represented as a linear combination of the average shape and the model eigenvectors:

$$\mathbf{s} = \bar{\mathbf{s}} + \sum_{i=1}^{m} c_i P_i, \tag{1}$$

where $\mathbf{c} = [c_1 \ \dots \ c_m]^T$ is a vector of parameters. We stack the eigenvectors to form a matrix $\mathbf{P}$, such that we may write: $\mathbf{s} = \bar{\mathbf{s}} + \mathbf{Pc}$. The PCA eigenvalues λ_i provide a measure of how much of the variance of the training data is captured by each eigenvector. We may choose to retain $n < m$ model dimensions, such that a certain percentage of the cumulative variance is captured. We discuss the effect of the number of model dimensions and empirically evaluate their stability in Section 2.2.

Our interest in this paper is to explore how shape samples drawn from a population distribute themselves in parameter space and how we can use this knowledge to constrain operations. We define the vector $\mathbf{b} = [c_1/\sqrt{\lambda_1} \ \dots \ c_n/\sqrt{\lambda_n}]^T$ as the variance-normalised parameter vector. This vector is distributed according to a multivariate Gaussian with zero mean and unit variance, i.e. $\mathbf{b} \sim \mathcal{N}(0, \mathbf{I}_n)$. This is the prior constraint typically used in the model fitting process to ensure that solutions remain plausible. It is maximised by a zero vector, which corresponds to the mean sample.

However, another interpretation based on the parameter vector length is possible. The squared norm of $\mathbf{b}$ corresponds to the square of the Mahalanobis distance of $\mathbf{c}$ from the mean:

$$||\mathbf{b}||^2 = D_M^2(\mathbf{c}) = \sum_{i=1}^{n} \left(\frac{c_i}{\sqrt{\lambda_i}} \right)^2 . \tag{2}$$

Since we assume each parameter follows a Gaussian distribution, the parenthesised terms are independent, normally distributed random variables with zero mean and unit variance. The sum of the square of such variables follows a chi-square distribution with n degrees of freedom, i.e. $||\mathbf{b}||^2 \sim \chi_n^2$. This distribution has expected value n and variance $2n$.

These two apparently contradictory distributions suggest that the mean face is the most probable sample but has a highly improbable vector length. For example, a model with 100 dimensions would have an expected vector length of 100 and over 99% of parameter vectors would have lengths between 70 and 130. The probability of a vector length less than 50 is negligibly small.

2.1 Identity as Direction

From the discussion above, it is clear that valid members of the class will occupy a subset of parameter space. These points will lie close to the surface of a hyperellipsoid, the diameters of which are determined by the eigenvalues of the data and the variance of the distance of samples from the manifold determined by the number of model dimensions. It is worth noting that as the number of dimensions increases, so the variance increases and the distance of samples from the manifold increases. Hence, the validity of assuming points lie on the surface of the hyperellipsoidal manifold breaks down as the number of model dimensions increases. Nevertheless, psychological results show us that the dimensionality of face space is relatively small (Meytlis and Sirovich [10] suggest 100 dimensions is sufficient, even using a crude eigenface model).

The analysis of data on a hyperellipsoidal manifold is extremely complex. Therefore, without loss of generality, we transform the manifold to a hypersphere by scaling each dimension by its corresponding standard deviation. By constraining faces to lie on the surface of this manifold, we maintain equal distinctiveness and ensure that only faces with the most probable distinctiveness can be generated. For the remainder of this paper, we therefore represent parameter vectors with squared Mahalanobis length n as unit vectors in $\mathbb{R}^n$: $\mathbf{x} = \frac{1}{\sqrt{n}} \left[\frac{c_1}{\sqrt{\lambda_1}} \ \cdots \ \frac{c_n}{\sqrt{\lambda_n}} \right]^T$.

A unit vector in n-dimensional space $\mathbf{x} \in \mathbb{R}^n$, may be considered as a point lying on the hyperspherical manifold $x \in S^{n-1}$. The two are related by $\mathbf{x} = \Phi(x)$ where $\Phi : S^{n-1} \mapsto \mathbb{R}^n$ is an embedding. If $v \in T_b S^{n-1}$ is a vector in the tangent space to S^{n-1} at a base point $b \in S^{n-1}$, the *exponential map*, denoted Exp_b of v is the point on S^{n-1} along the geodesic in the direction of v at distance $\|v\|$ from b. The inverse of the exponential map is the log map, denoted Log_b.

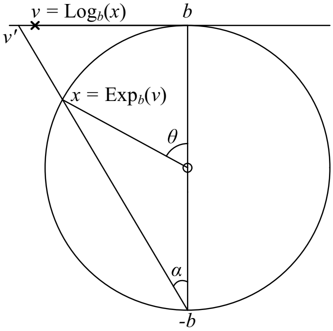


Fig. 1. Computing log and exponential maps using a stereographic projection for the S^1 manifold

The geodesic distance (i.e. angular difference) between two points on the unit hypersphere $x_1, x_2 \in S^{n-1}$ can be expressed in terms of the log map, i.e. $d(x_1, x_2) = \|\mathrm{Log}_{x_1}(x_2)\| = \arccos(\Phi(x_1) \cdot \Phi(x_2))$.

We propose a novel implementation of the exponential and log maps for a unit hypersphere which is both simple and efficient. We do so using a stereographic projection. The log map of a point x at basepoint b is calculated as follows. We define the tangent vector $v' \in T_b S^{n-1}$ as the stereographic projection of x from $-b$ to the tangent space to S^{n-1} at b. This tangent vector has the correct direction but incorrect magnitude. To obtain the log map of x, we rescale v' giving v, such that $\|v\| = d(b, x)$. The exponential map is computed by reversing this process, i.e. by applying an inverse stereographic projection to the rescaled tangent vector. Figure 1 clarifies the geometry involved for the S^1 case.

In practice, we represent points on both the hyperspherical manifold and the tangent space as vectors embedded in $\mathbb{R}^n$. Hence, our proposed implementation of the log map of x at base point b is computed as:

$$\Phi_T\left(\mathrm{Log}_b(x)\right) = \mathbf{b} + \frac{\theta(\mathbf{v}' - \mathbf{b})}{\|\mathbf{v}' - \mathbf{b}\|}, \tag{3}$$

where $\mathbf{b} = \Phi(b)$ and $\mathbf{x} = \Phi(x)$ are both unit vectors in $\mathbb{R}^n$,

$$\mathbf{v}' = \frac{2(\mathbf{b} + \mathbf{x})}{\|\mathbf{b} + \mathbf{x}\| \cos\alpha} - \mathbf{b}, \qquad \alpha = \arccos\left(\frac{4 + \|\mathbf{b} + \mathbf{x}\|^2 - \|\mathbf{x} - \mathbf{b}\|^2}{4\|\mathbf{b} + \mathbf{x}\|}\right), \tag{4}$$

and $\theta = \arccos(\mathbf{b} \cdot \mathbf{x})$.

The result is a point in the tangent space $T_b S^{n-1}$ embedded in $\mathbb{R}^n$ according to an arbitrary embedding $\Phi_T : T_b S^{n-1} \mapsto \mathbb{R}^n$. A similar expression can be derived for the exponential map. These expressions hold for unit vectors in any number of dimensions. In the remaining sections, we use the log and exponential map to derive useful operations on the manifold.

2.2 Empirical Evaluation: χ^2 Prediction

Before we consider applications of processing data on the manifold described above, we provide some empirical assessment of how well the theoretically predicted manifold adheres to real world data. In order for all plausible data samples to lie on or near the manifold, the assumption of parameter vector lengths following the chi-squared distribution must hold. In turn, the distribution of faces along each eigenvector must follow a Gaussian distribution. In practice, these eigenvectors are estimated from a sparse sample of a high dimensional space. In the case of a dense 3D face shape model, observations typically consist of tens of thousands of vertices while the training set typically comprises only hundreds of samples.

Clearly, the validity of the estimated manifold depends on the quality of the estimated eigenvectors and therefore the size and diversity of the training set. We empirically evaluate how well unseen data adheres to our assumptions. This allows us to determine how many model dimensions can be safely retained.

Our empirical test is conducted as follows. From a pool of 100 face meshes [15], we randomly select 75. We build a PCA model and project each of the remaining 25 out-of-sample data onto the model eigenvectors. We repeat this process 80 times, giving a total of 2000 out-of-sample parameter vectors. We analyse the

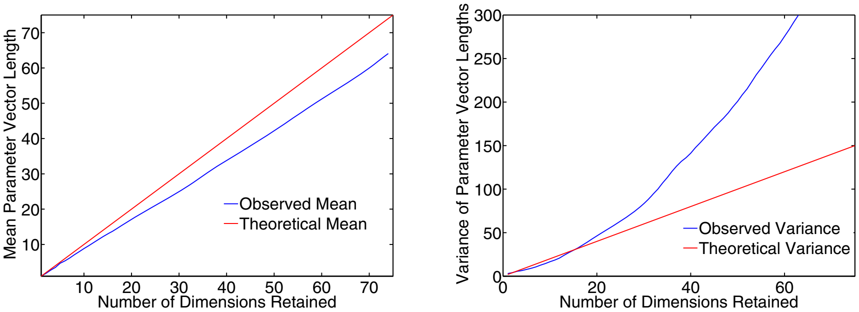


Fig. 2. Predicted versus observed mean (left) and variance (right) of out-of-sample parameter vector lengths

mean and variance of the squared-Mahalanobis length of these vectors and measure how well they agree with the predicted chi-square distribution. We would expect the mean and variance to grow linearly with the number of model dimensions retained. As can be seen in Figure 2, the observed mean lengths are close to, but smaller than, the theoretical prediction. On the other hand, the variance is only close to the predicted value for up to approximately 20 dimensions. Beyond this, the variance increases rapidly meaning many points will lie a significant distance from the manifold surface. We believe this is an effect of the sparsity of the training data. A much larger training set would allow this effect to be studied further. Nevertheless, we can see that for a modest number of dimensions, real world data does follow the statistical prediction reasonably well.

2.3 Empirical Evaluation: Manifold Approximation

The second empirical evaluation necessary to justify our approach, is to assess the error induced by forcing all samples to lie on the manifold, i.e. enforcing a hard constraint on vector length. Given an out-of-sample face, $\mathbf{s}$, the optimal parameter vector (in a least squares sense) is given by $\mathbf{c}^* = \mathbf{P}^T(\mathbf{s} - \bar{\mathbf{s}})$. Substituting $\mathbf{c}^*$ back into (1), we can obtain $\mathbf{s}^{\text{mod}}$, the shape which minimises $\|\mathbf{s}^{\text{mod}} - \mathbf{s}\|^2$. However, this shape is not constrained by the model prior and is almost always an overfit to the data. We compare this optimal model-based reconstruction to the shape, $\mathbf{s}^{\text{man}}$, obtained by projecting $\mathbf{c}^*$ to the closest point on the hyperspherical manifold:

$$\mathbf{c}^{\text{man}} = \frac{\sqrt{n}}{D_M(\mathbf{c}^*)}\mathbf{c}^*. \tag{5}$$

Over the 10 out-of-sample faces in the BFM [12] the mean Euclidian error of $\mathbf{s}^{\text{mod}}$ for a $n = 99$ parameter model was 1.128mm. By projecting to the S^{n-1} hypersphere, the mean Euclidian error of $\mathbf{s}^{\text{man}}$ increased to 1.89mm. The optimal choice of $n-1$ dimensional subspace (with respect to Euclidian error) would be to simply retain the first $n-1$ eigenvectors of the PCA model. For our data, this gives a mean Euclidian error of only 1.133mm. However, the purpose of our choice of manifold is to enforce *plausibility*. This is reflected in the fact that error in the surface normals of the approximated faces (which in turn determines appearance), *reduces* when projecting to the manifold. For our data, the mean angular error drops from 5.92° for $\mathbf{s}^{\text{mod}}$ to 5.48° for $\mathbf{s}^{\text{man}}$. In other words, by constraining faces to be more plausible, we reduce appearance error.

3 Plausibility-Preserving Warps and Averages

Warping between faces or, more generally, computing weighted combinations of two or more faces has applications in animation and in the production of stimuli for psychological experiments [11]. The most obvious way to warp between two shapes that are in dense correspondence is to linearly warp each vertex from its position in one shape to its position in the other. Equivalently, this can be approximated by linearly warping between the two vectors of PCA parameters. However, in either case the intermediate faces will not correspond to plausible faces. Since the manifold of maximally probable distinctiveness is curved, any linear warp will include faces that do not lie on the manifold, with the least plausible face occurring halfway along the warp.

Face-antiface warps provide a particularly interesting special case. An antiface is the antipodal point of a source face on the manifold. Perceptually, antifaces appear "opposite" in some sense to the original face. The vector connecting a face to its antiface in parameter space passes through the mean. A linear warp between a face and antiface is therefore well-defined but will include implausible faces for the duration of the warp. There is a further problem with such linear warps. Psychological studies have shown that there is a perceptual discontinuity as the face trajectory crosses the mean [11]. In other words, as identity flips from face to antiface, the perceptual effect of a small movement through face space is exaggerated.

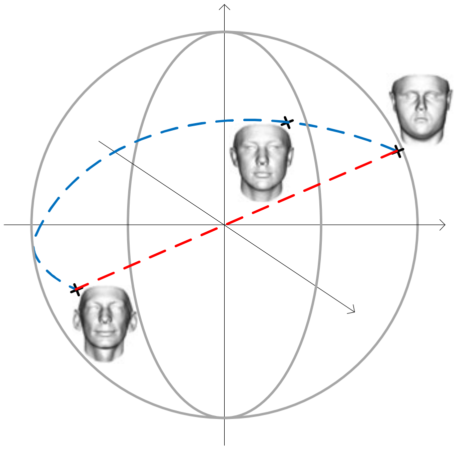


Fig. 3. Warping between face and antiface on the S^2 manifold. Linear warp is shown in red, one of the possible geodesic warps is shown in blue.

Instead, we propose warps which take place across the surface of the manifold, following the geodesic curve between the two source faces. Another way to view these warps is as a rotation of a unit vector in $\mathbb{R}^n$. All intermediate faces in this case have equal distinctiveness and are equally plausible. In the case of antifaces, there is no single geodesic warp connecting face to antiface. In fact, there are an infinite number of valid warps, all of length π. Any such warp will smoothly vary identity from the source face to its antiface, via a series of faces with uniform distinctiveness. One way to conceptualise this is that we can set off from a point on the hyperspherical manifold in any direction and reach the antiface after travelling a distance π.

An interesting result of this observation is that we can choose any intermediate face as a target which will be visited on the warp from face to antiface. This gives us a way to specify one of the infinite face-antiface warps and may also have interesting applications in generating stimuli for psychological studies. This idea is demonstrated in Figure 3 for the S^2 manifold, which shows the difference between a plausibility-preserving and linear warp.

For a source face x_{src} and intermediate target face x_{tar}, we can define a unit vector in the tangent space, $v \in T_{x_{src}}S^{n-1}$, from x_{src} in the direction of x_{tar}: $v = \frac{\text{Log}_{x_{src}}(x_{tar})}{d(x_{src},x_{tar})}$. A geodesic warp from x_{src} to x_{tar} is therefore given by following this vector by a distance specified by the warping parameter w:

$$x_{war} = \text{Exp}_{x_{src}}\left(w\frac{\text{Log}_{x_{src}}(x_{tar})}{d(x_{src},x_{tar})}\right). \tag{6}$$

When $w = 0$ we obtain the source face, i.e. $x_{war} = x_{src}$, and when $w = d(x_{src}, x_{tar})$ we obtain the target face, i.e. $x_{war} = x_{tar}$. If we set $w = \pi$ we obtain the antiface to x_{src}. Intermediate faces are obtained when $w \in (0, \pi)$.

We show an example warp from face to antiface via an intermediate target face in Figure 5 using the 199 parameter BFM [12]. Note that the effect is of smooth variation of identity, with each of the intermediate faces containing significant detail. We contrast this with a linear warp through the mean face which results in implausibly smooth intermediate faces and no transition through intermediate identities. In Figure 4 we plot the parameter vector lengths for the linear and plausibility-preserving warps.

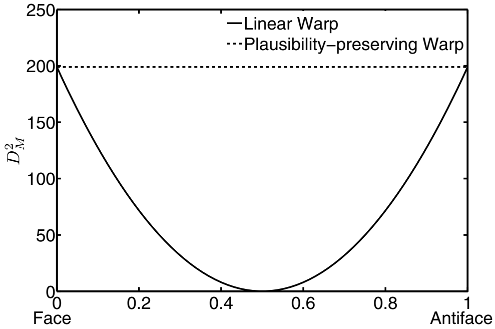


Fig. 4. Vector length or ‘plausibility’ is plotted throughout a warp between a face and antiface (see Figure 5)

3.1 Averages

Given $u > 2$ source faces, $x_1, \ldots, x_u \in S^{n-1}$, we wish to compute a plausible average face which captures characteristics of each of the source faces. The linear

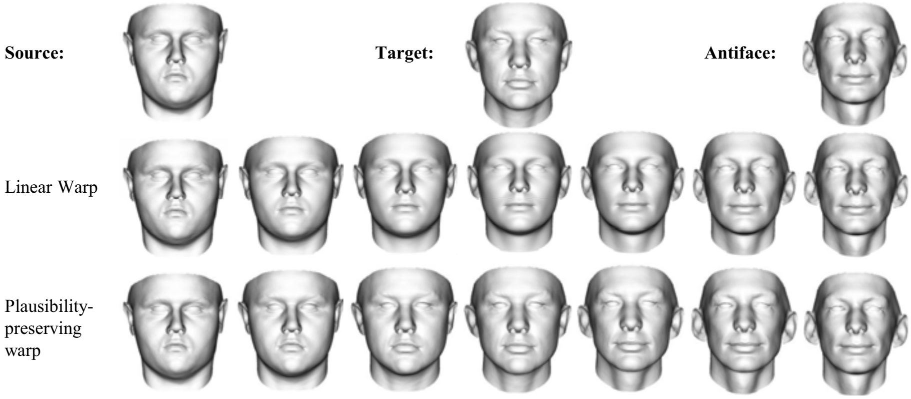


Fig. 5. Linear versus plausibility-preserving warp from face to antiface

or Euclidian mean of the parameter vectors minimises the sum of square error in $\mathbb{R}^n$ from the average to each of the source faces. This is the *extrinsic mean* and will not lie on the manifold. The result is that the face is implausibly smooth and lacking in features. We propose the use of the *intrinsic* or Karcher mean. For $u = 2$, this can be found using the warping equation given above with $w = 0.5$. For $u > 2$, this is the point $x_\mu \in S^{n-1}$ which minimises the total squared geodesic distance to each of the source faces:

$$x_\mu = \underset{x \in S^{n\;\;1}}{\arg\min} \sum_{i=1}^{u} d(x, x_i)^2. \tag{7}$$

This point cannot be found analytically, so we solve it as an iterative optimisation using the gradient descent method of Pennec [13]. We initialise our estimate as one of the source data points, i.e. $x_\mu^{(0)} = x_1$. The estimated intrinsic mean is then iteratively updated as follows:

$$x_\mu^{(j+1)} = \mathrm{Exp}_{x_\mu^{(j)}} \left(\frac{1}{u} \sum_{i=1}^{u} \mathrm{Log}_{x_\mu^{(j)}}(x_i) \right). \tag{8}$$

This process converges rapidly, typically within 5 iterations. In Figure 6 we compare our plausibility-preserving averages with linear averaging of the 74 dimensional parameter vectors obtained using the USF data [15]. Notice that each of the Euclidian averages appears unrealistically smooth, whereas the averages computed on the manifold clearly show the presence of distinct features present in the source faces (for example, the broader nostrils of face 1 are visible in the first three averages but not the fourth).

4 Model Fitting on the Manifold of Plausible Faces

The most powerful application of the identity manifold is to use it for the purpose of constraining the process of fitting a model to data. Suppose the function

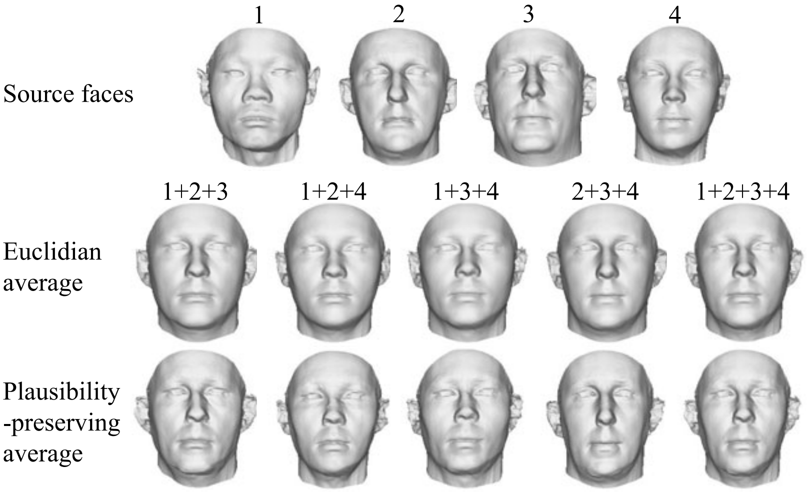


Fig. 6. Linear versus plausibility-preserving averages

$\varepsilon : S^{n-1} \mapsto \mathbb{R}$ is an objective function which evaluates the quality of fit of a face represented by a point on the plausibility manifold to some observed data. This function could take any form, for example the difference between predicted and observed appearance in an analysis-by-synthesis framework or the error between a sparse set of feature points. We pose model fitting as finding the point on the manifold which minimises this error, i.e.:

$$x^* = \arg \min_{x \in S^{n\ \ 1}} \varepsilon(x). \tag{9}$$

In doing so, we ensure that plausibility is enforced as a hard constraint. Note also that the optimisation is more heavily constrained since the dimensionality of the hypersphere is 1 less than the parameter space.

4.1 Local Optimisation

We can perform gradient descent on the surface of the manifold to find a local minimum in the error function. The fact that our manifold is hyperspherical has some interesting implications for such an approach. We must first compute the gradient of the objective function in terms of a vector on the tangent plane: $\nabla\varepsilon(x) \in T_x S^{n-1}$. To do so, we compute the gradient in terms of a vector in $\mathbb{R}^n$ and project the result to the tangent plane as follows:

$$\nabla\varepsilon(x) = \mathrm{Log}_x \left(\Phi^{-1} \left(\frac{\mathbf{x} - \mathbf{g}}{\|\mathbf{x} - \mathbf{g}\|} \right) \right), \tag{10}$$

where $\mathbf{x} = [x_1 \ \dots \ x_n]^T = \Phi(x)$. The gradient $\mathbf{g} = [\partial_{x_1}\varepsilon(x) \ \dots \ \partial_{x_n}\varepsilon(x)]^T$ is approximated by using finite differences to calculate the partial derivatives:

$$\partial_{x_i}\varepsilon(x) \approx \frac{\varepsilon(x'_i) - \varepsilon(x)}{\epsilon}, \tag{11}$$

where $x'_i = \Phi^{-1}([x_1 \ \dots \ x_i + \epsilon \ \dots \ x_n])$.

With a means to compute the gradient, we can iteratively minimise the objective function by adapting the gradient descent algorithm to operate on the surface of a manifold:

$$x^{(t+1)} = \mathrm{Exp}_{x^{(t)}}\left(-\gamma\nabla\varepsilon(x^{(t)})\right), \tag{12}$$

where γ is the step size. Note that as γ varies, the point $\mathrm{Exp}_x\left(-\gamma\nabla\varepsilon(x)\right) \in S^{n-1}$ traces out a great circle about the hypersphere. This is the search space for the one-dimensional line search at each iteration of gradient descent.

4.2 Coarse-to-Fine Model Fitting

The difficulty with our approach is choosing an unbiased initialisation. Existing methods for fitting statistical models to data typically commence from an initialisation of the mean (i.e. zero parameter vector), e.g. [3,5]. However, this point lies far from the plausibility manifold and is therefore unsuitable in our case.

We tackle this problem and also reduce susceptibility to becoming trapped in local minima by proposing a coarse-to-fine algorithm which iteratively increases the number of model dimensions considered in the optimisation.

Consider in the simplest case a 1-dimensional model. Only two points strictly satisfy the plausibility constraint in this case and the problem therefore reduces to a binary decision:

$$\mathbf{x}^{(1)} = \begin{cases} [\ 1\ ] & \text{if } \varepsilon(\Phi^{-1}([\ 1\ ])) < \varepsilon(\Phi^{-1}([\ -1\ ])) \\ [\ -1\ ] & \text{otherwise} \end{cases}, \tag{13}$$

We use this result to initialise the solution in two dimensions, initially setting the second parameter to zero: $\mathbf{x}^{(n)}_{\text{init}} = \left[\mathbf{x}^{(n-1)} \mid 0\right]$. We then perform gradient descent, which in the two parameter case means optimising a single angular parameter. We continue this process, incrementally adding dimensions to the optimisation, each time setting the new parameter to zero and then performing gradient descent on the new manifold using this as an initialisation. Hence, the result of a local optimisation in n dimensions is used as the initialisation for optimisation in $n+1$ dimensions ensuring that the solution is already constrained to the right region of the manifold.

The nature of the hyperspherical manifold can be used to inform the step size used in the gradient descent optimisation. We assume that the result in n dimensions has restricted the solution to the correct hemisphere of the hypersphere. Travelling in the direction of the negative gradient reduces the error. To travel in this direction whilst remaining in the same hemisphere means the maximum arc distance that can be moved is $\frac{\pi}{2}$. Hence, the result in n dimensions is given by $\mathbf{x}^{(n)} = h(d^*)$, where

$$h(d) = \mathrm{Exp}_{\varPhi\ {}^{1}(\mathbf{x}_{\text{init}}^{(n)})}\left(d \frac{-\nabla\varepsilon\left(\varPhi^{-1}(\mathbf{x}_{\text{init}}^{(n)})\right)}{\left\|\nabla\varepsilon\left(\varPhi^{-1}(\mathbf{x}_{\text{init}}^{(n)})\right)\right\|} \right). \tag{14}$$

The arc distance d determines how far we travel along the great circle implied by the gradient of the objective function. Since we wish to constrain our solution to the same hemisphere, d must lie in the interval $(0, \frac{\pi}{2})$ and we hence find d^* using golden section search [7] to solve: $d^* = \arg\min_{d} h(d),\ 0 < d < \frac{\pi}{2}$. Multiple iterations of gradient descent can be used each time a dimension is added to the optimisation. In our results we use four iterations per dimension.

4.3 Model Fitting Example

For our experimental evaluation, we use the algorithm described above to fit our 3D morphable shape model to unseen data. We choose as an objective function the angular error between surface normals at each vertex of the model. This is an interesting choice of objective function for two reasons. First, the search landscape of the objective function is littered with local minima. Second, the fitted result is likely to have lower perceptual error than a least squares fit directly to the vertices. Whilst such a least squares fit gives minimal geometric error, the result is often a gross over-fit which does not resemble the input face. Minimising the surface normal error is a non-linear problem which is related to minimising appearance error, as undertaken by analysis-by-synthesis of image data [3].

From an input face shape, represented by p vertices, we compute surface normals at each vertex by averaging face normals of faces adjacent to the vertex. If $\mathbf{N}^i$ is the surface normal at vertex i, our objective function is the sum of squared angular errors between input and model surface normals:

$$\varepsilon(x) = \sum_{i=1}^{p} \left(\arccos(\mathbf{n}^i(\varPhi(x)) \cdot \mathbf{N}^i)\right)^2, \tag{15}$$

where $\mathbf{n}^i([x_1\ \dots\ x_n])$ is the surface normal of the ith vertex of the shape given by: $\overline{\mathbf{s}} + \mathbf{Pc}$, where the parameter is vector is computed by transforming the unit vector back to the hyperellipse: $\mathbf{c} = \sqrt{n}\left[x_1\sqrt{\lambda_1}\ \dots\ x_n\sqrt{\lambda_n}\right]^T$.

We compare our manifold optimisation with direct optimisation of (15) using a generic optimiser based on the BFGS Quasi-Newton method with a cubic line search [4]. Note that the generic optimiser converges close to the mean if all parameters are optimised simultaneously. We therefore take the same coarse-to-fine approach as for the manifold fitting, whereby we iteratively increase the number of dimensions considered in the optimisation.

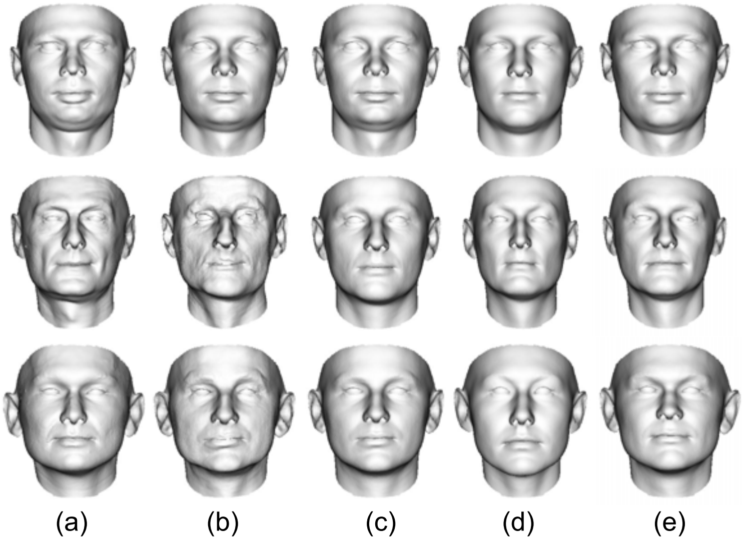


Fig. 7. Model fitting result: (a) input unseen face; (b) least squares fit to vertices; (c) parameter vector of (b) rescaled to manifold; (d) BFGS optimisation; (e) manifold optimisation. All the results are for a $n = 99$ parameter model.

In Figure 7 we show results on the BFM [12] data. Column (a) shows input faces which are not in the morphable model training set. A simple linear least squares fit of the model to the vertices of the unseen faces yields the result in column (b). Whilst this result is optimal in terms of the Euclidian error between input and reconstructed vertices, the result is an overfit and, in particular, the face in row 2 is clearly implausible. Rescaling the parameter vector obtained by least squares to the closest point on the manifold yields the result shown in column (c). While this face is now plausible, it lacks any of the distinguishing features of the input faces. Column (d) shows the result of using a generic non-linear optimiser to solve (15). Because of local minima close to the mean, these faces are implausibly smooth. Finally, our manifold fitting result is shown in column (e). Note that this result represents a trade off between over and under-fitting. The mean angular error of the surface normals for the out-of-sample faces in the BFM using (d) is 7.23°, while using the proposed method the error is 5.33°. Our result outperformed the generic non-linear optimiser for all of the BFM faces.

5 Conclusions

We have shown how a number of useful operations can be performed on the manifold of equally distinctive faces. This provides a new way to constrain operations involving the parameters of a statistical model. In particular, we have shown how to constrain the process of fitting a model to data and how a coarse-to-fine strategy avoids local minima. Matlab implementations are available at (http://www.cs.york.ac.uk/~wsmith/ECCV2010.html). In future work, we intend to apply our model fitting strategy to more demanding objective functions and to experiment with other sources of data besides faces.

References

1. Amberg, B., Blake, A., Fitzgibbon, A., Romdhani, S., Vetter, T.: Reconstructing high quality face-surfaces using model based stereo. In: Proc. ICCV (2007)
2. Blanz, V., Scherbaum, K., Seidel, H.P.: Fitting a morphable model to 3D scans of faces. In: Proc. ICCV (2007)
3. Blanz, V., Vetter, T.: Face recognition based on fitting a 3D morphable model. IEEE Trans. Pattern Anal. Mach. Intell. 25(9), 1063–1074 (2003)
4. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms. Journal of the Institute of Mathematics and Its Applications 6(1), 76–90 (1970)
5. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 484–498. Springer, Heidelberg (1998)
6. Hu, C., Xiao, J., Matthews, I., Baker, S., Cohn, J., Kanade, T.: Fitting a single active appearance model simultaneously to multiple images. In: Proc. BMVC (2004)
7. Kiefer, J.: Sequential minimax search for a maximum. Proceedings of the American Mathematical Society 4(3), 502–506 (1953)
8. Knothe, R., Romdhani, S., Vetter, T.: Combining PCA and LFA for surface reconstruction from a sparse set of control points. In: Proc. Int. Conf. on Automatic Face and Gesture Recognition, pp. 637–644 (2006)
9. Matthews, I., Baker, S.: Active appearance models revisited. Int. J. Comput. Vis. 60(2), 135–164 (2004)
10. Meytlis, M., Sirovich, L.: On the dimensionality of face space. IEEE Trans. Pattern Anal. Mach. Intell. 29(7), 1262–1267 (2007)
11. O'Toole, A.J., Vetter, T., Volz, H., Salter, E.M.: Three-dimensional caricatures of human heads: Distinctiveness and the perception of facial age. Perception 26(6), 719–732 (1997)
12. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D face model for pose and illumination invariant face recognition. In: Proc. IEEE Intl. Conf. on Advanced Video and Signal based Surveillance (2009)
13. Pennec, X.: Probabilities and statistics on Riemannian manifolds: basic tools for geometric measurements. In: Proc. IEEE Workshop on Nonlinear Signal and Image Processing (1999)
14. Romdhani, S., Vetter, T.: Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In: Proc. CVPR, vol. 2, pp. 986–993 (2005)
15. Sarkar, S.: USF humanid 3D face database (2005)
16. Valentine, T.: A unified account of the effects of distinctiveness, inversion, and race in face recognition. Quarterly Journal of Experimental Psychology A 43(2), 161–204 (1991)
17. Xiao, J., Baker, S., Matthews, I., Kanade, T.: Real–time combined 2D+3D active appearance models. In: Proc. CVPR, pp. 535–542 (2004)

Multi-label Linear Discriminant Analysis

Hua Wang, Chris Ding, and Heng Huang

Department of Computer Science and Engineering, University of Texas at Arlington,
Arlington, TX 76019, USA
huawang2007@mavs.uta.edu, chqding@uta.edu, heng@uta.edu

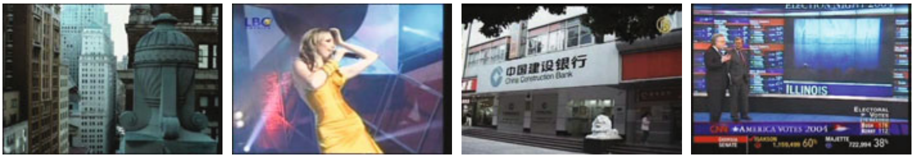
Abstract. Multi-label problems arise frequently in image and video annotations, and many other related applications such as multi-topic text categorization, music classification, *etc.* Like other computer vision tasks, multi-label image and video annotations also suffer from the difficulty of high dimensionality because images often have a large number of features. Linear discriminant analysis (LDA) is a well-known method for dimensionality reduction. However, the classical Linear Discriminant Analysis (LDA) only works for single-label multi-class classifications and cannot be directly applied to multi-label multi-class classifications. It is desirable to naturally generalize the classical LDA to multi-label formulations. At the same time, multi-label data present a new opportunity to improve classification accuracy through label correlations, which are absent in single-label data. In this work, we propose a novel Multi-label Linear Discriminant Analysis (MLDA) method to take advantage of label correlations and explore the powerful classification capability of the classical LDA to deal with multi-label multi-class problems. Extensive experimental evaluations on five public multi-label data sets demonstrate excellent performance of our method.

Keywords: Multi-label classification, Multi-label linear discriminant analysis, Image annotation.

1 Introduction

Image and video annotation has been an active research topic in recent years due to its potentially large impact on both image/video understanding and web/database image/video retrieval. In a typical image annotation problem, each picture is usually associated with several different conceptual classes. For example, the picture in Figure 1(a) is annotated with "building", "outdoor", and "urban", and similarly other pictures in Figure 1 are also associated with more than one semantic concepts. In machine learning, such problems that require each data point to be assigned to multiple different categories are called as *multi-label* classification problem. In contrast, in traditional *single-label* classification, also known as *single-label multi-class* classification, each data point belongs to only one category. Multi-label multi-class classification is more general than single-label multi-class classification, and recently has stimulated a slew of multi-label learning algorithms [16,5,7,10,3,9,17,4,15].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 126–139, 2010.



(a) building, outdoor, urban (b) face, person, entertainment (c) building, outdoor, urban (d) TV screen, person, studio

Fig. 1. Sample images from TRECVID 2005 data set. Each image is annotated with several different semantic words (listed under each images). When they are used as test images during cross-validations, our new proposed MLDA methods can correctly predict all of them. But other previous methods can only predict the first or second labels of each image. They cannot predict 'urban' in (a), 'entertainment' in (b), 'urban' in (c), 'person' and 'studio' in (d).

An important difference between single-label classification and multi-label classification lies in that, classes in the former are assumed mutually exclusive, while those in the latter are typically interdependent from one another. That is, in multi-label classification, class memberships can be inferred from one another through label correlations, which provide an important opportunity to improve classification accuracy. As a result, a multi-label classification method should make use of label correlations for improved classification performance.

High dimensionality of typical image data makes dimensionality reduction an important step to achieve efficient and effective image annotation. Among various dimensionality reduction methods in statistical learning, Linear Discriminant Analysis (LDA) is well known and widely used due to its powerful classification capability. However, LDA by nature is devised for single-label classification, therefore it can not be directly used in image annotation. The main difficulty to apply the classical LDA in multi-label classification is how to measure the inter and intra class scatters, which are clearly defined in single-label classification but become obscure in multi-label case. Because a data point with multiple labels belongs to different classes at the same time, how much it should contribute to the between-class and within-class scatters remains unclear. Therefore, it is desirable to generalize the classical LDA to deal with multi-label classification problem, and meanwhile, incorporate mutual correlations among labels.

In this paper, we propose a novel Multi-label Linear Discriminant Analysis (MLDA) method to explore the powerful classification capability of LDA in multi-label tasks and take advantage of label correlations. We first review the classical LDA and point out the computation ambiguity when using traditional single-label definitions of the scatter matrices in multi-label classification. After that, we introduce the details of our proposed MLDA method with empirical validations.

2 Difficulties of Classical LDA in Multi-Label Classification

Given a data set with n samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ and K classes, where $\mathbf{x}_i \in \mathbb{R}^p$ and $\mathbf{y}_i \in \{0,1\}^K$. $\mathbf{y}_i(k) = 1$ if $\mathbf{x}_i$ belongs to the k-th class, and 0 otherwise. Let input data be partitioned into K groups as $\{\pi_k\}_{k=1}^K$, where π_k denotes the sample set of the k-th class with n_k data points. We write $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and

$$Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^T = \left[\mathbf{y}_{(1)}, \ldots, \mathbf{y}_{(K)}\right], \tag{1}$$

where $\mathbf{y}_{(k)} \in \{0,1\}^n$ is the class-wise label indication vector for the k-th class.

2.1 Review of Classical LDA

Classical LDA seeks a linear transformation $G = \mathbb{R}^{p\times r}$ that maps $\mathbf{x}_i$ in the high p-dimensional space to a vector $\mathbf{q}_i \in \mathbb{R}^r$ in a lower $r(< p)$-dimensional space by $\mathbf{q}_i = G^T\mathbf{x}_i$. In classical LDA, the *between-class*, *within-class*, and *total-class* scatter matrices are defined as follows [2]:

$$S_b = \sum_{k=1}^K n_k \left(\mathbf{m}_k - \mathbf{m}\right)\left(\mathbf{m}_k - \mathbf{m}\right)^T, \tag{2}$$

$$S_w = \sum_{k=1}^K \sum_{\mathbf{x}_i \in \pi_k} \left(\mathbf{x}_i - \mathbf{m}_k\right)\left(\mathbf{x}_i - \mathbf{m}_k\right)^T, \tag{3}$$

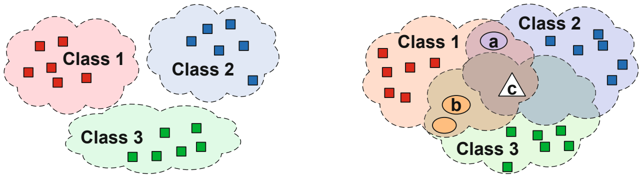
$$S_t = \sum_{i=1}^n \left(\mathbf{x}_i - \mathbf{m}\right)\left(\mathbf{x}_i - \mathbf{m}\right)^T, \tag{4}$$

where $\mathbf{m}_k = \frac{1}{n_k}\sum_{\mathbf{x}_i\in\pi_k}\mathbf{x}_i$ is the class mean (class centroid) of the k-th class, $\mathbf{m} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$ is the global mean (global centroid), and $S_t = S_b + S_w$. The optimal G is chosen such that the between-class distance is maximize whilst the within-class distance is minimized in the low-dimensional projected space, which leads to the standard LDA optimization objective [2] as follows:

$$\arg\max_G J = \mathbf{tr}\left(\frac{G^T S_b G}{G^T S_w G}\right). \tag{5}$$

2.2 Ambiguity Caused by Data with Multiple Labels in Classical LDA

Classical LDA for single-label classification is summarized in Eqs. (2–5), where the scatter matrices, S_w, S_b, and S_t, are well-defined as per the spatial distribution of data points as in Figure 2(a). However, in multi-label case, these definitions become obscure, because decision regions overlap among one another and decision boundaries turn out ambiguous as in Figure 2(b). Besides the data



(a) Single-label data. (b) Multi-label data.

Fig. 2. (a) In single-label classification, every data point distinctly belongs to only one class. (b) In multi-label classification, some data points may belong to multiple classes, denoted as ovals and triangles, which cause the ambiguity in scatter matrices calculations.

points belonging to only one class denoted by squares, some data points could also belong to multiple classes, as denoted by ovals for those belonging to two classes and triangles for those belonging to all three classes. How much a data point with multiple labels should contribute to the data scatters is not defined, therefore the scatter matrices defined in Eqs. (2–4) can not be computed.

3 Multi-label Linear Discriminant Analysis (MLDA)

Classical LDA deals with single-label problems, where data partitions are mutually exclusive, *i.e.*, $\pi_i \cap \pi_j = \varnothing$ if $i \neq j$, and $\sum_{k=1}^{K} n_k = n$. This, however, is no longer held in multi-label case. In this section, we propose a novel Multi-label Linear Discriminant Analysis (MLDA) method to explore the powerful classification capability of classical LDA in multi-label classification tasks. We first solve the ambiguity problem revealed in Section 2, and then leverage label correlations to enhance classification performance. Our method is a natural generalization of classical LDA.

3.1 Class-Wise Scatter Matrices

The ambiguity when using traditional single-label definitions of scatter matrices in multi-label classification prevents us from directly applying classical LDA to solve multi-label problems. Therefore, instead of defining the scatter matrices from data point perspective as in Eqs. (2–4), we propose to compute them by class-wise, such that the structural variances of training data are represented more lucidly and the scatter matrices are easier to be constructed. Moreover, the ambiguity, how much a data point with multiple labels should contribute to the scatter matrices, is avoided, and label correlations can be incorporated. The *class-wise between-class scatter matrix* is defined as:

$$S_b = \sum_{k=1}^{K} S_b^{(k)}, \; S_b^{(k)} = \left(\sum_{i=1}^{n} Y_{ik} \right) (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T, \tag{6}$$

the *class-wise within-class scatter matrix* S_w is defined as:

$$S_w = \sum_{k=1}^{K} S_w^{(k)}, \; S_w^{(k)} = \sum_{i=1}^{n} Y_{ik} \left(\mathbf{x}_i - \mathbf{m}_k\right) \left(\mathbf{x}_i - \mathbf{m}_k\right)^T , \tag{7}$$

and the *class-wise total-class scatter matrix* S_t is defined as:

$$S_t = \sum_{k=1}^{K} S_t^{(k)}, \; S_t^{(k)} = \sum_{i=1}^{n} Y_{ik} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T, \tag{8}$$

where $\mathbf{m}_k$ is the mean of class k and $\mathbf{m}$ is the *multi-label global mean*, which are defined as follows:

$$\mathbf{m}_k = \frac{\sum_{i=1}^{n} Y_{ik}\mathbf{x}_i}{\sum_{i=1}^{n} Y_{ik}}, \quad \mathbf{m} = \frac{\sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik}\mathbf{x}_i}{\sum_{k=1}^{K}\sum_{i=1}^{n} Y_{ik}}. \tag{9}$$

Theorem 1. *When applied into single-label classification, the multi-label scatter matrices, S_b, S_w, and S_t, defined in Eqs. (6–8), are reduced to their corresponding counterparts in classical LDA as defined in Eqs. (2–4).*

From the above definitions, the Theorem 1 can be easily obtained. Most importantly, in classical LDA, $S_t = S_b + S_w$, which is still held in multi-label classifications.

Theorem 2. *For multi-label class-wise scatter matrices, $S_b^{(k)}$, $S_w^{(k)}$, and $S_t^{(k)}$ as defined in Eqs. (6–8), the following relationship is held:*

$$S_t^{(k)} = S_b^{(k)} + S_w^{(k)}. \tag{10}$$

Therefore, $S_t = S_b + S_w$.

Proof. According to Eq. (9), we have $\sum_{i=1}^{n} Y_{ik}\mathbf{m}_k = \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i$. Thus,

$$\sum_{i=1}^{n} Y_{ik}\mathbf{m}_k\mathbf{m}_k^T = \sum_{i=1}^{n} Y_{ik}\mathbf{m}_k\mathbf{x}_i^T \;\text{ and }\; \sum_{i=1}^{n} Y_{ik}\mathbf{m}_k\mathbf{m}_k^T = \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{m}_k^T. \tag{11}$$

From Eqs. (6–8) and using Eq. (11), we have:

$$S_t^{(k)} = \sum_{i=1}^{n} Y_{ik}\mathbf{x}_i\mathbf{x}_i^T + \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m}_k\mathbf{m}^T - \sum_{i=1}^{n} Y_{ik}\mathbf{m}\mathbf{m}_k^T = S_b^{(k)} + S_w^{(k)} \tag{12}$$

□

3.2 Multi-label Correlations

Multi-label data provide a new opportunity to improve classification accuracy through label correlations, which are absent in single-label data. Typically, the label correlation between two classes is formulated as following [15]:

$$C_{kl} = \cos(\mathbf{y}_{(k)}, \mathbf{y}_{(l)}) = \frac{\langle \mathbf{y}_{(k)}, \mathbf{y}_{(l)} \rangle}{\|\mathbf{y}_{(k)}\| \, \|\mathbf{y}_{(l)}\|}. \tag{13}$$

Thus, $C \in \mathbb{R}^{K \times K}$ is a symmetric matrix. Apparently, $C = I$ for single-label data. Namely, no label correlations can be utilized in single-label classification.

In multi-label classification, a data point may belong to several different classes simultaneously, hence the data points assigned to two different classes may overlap. Statistically, the bigger the overlap is, the more closely the two classes are related. Namely, class memberships in multi-label classification be inferred from one another through label correlations. Specifically, the *correlated labels assignments* are computed as:

$$Y^c = YC. \tag{14}$$

Several existing multi-label classification algorithms used label correlations to boost classification performance [16,1,3,15]. Using TRECVID 2005 data set with LSCOM-Lite annotation scheme [11], label correlations defined in Eq. (13) are illustrated in Figure 3. The high correlation value between "person" and "face" shows that they are highly correlated, which perfectly agree with the common sense in real life for the simplest fact that everybody has a face. Similar observations, such as "outdoor" and "sky", "waterscape-waterfront" and "boat-ship", "road" and "car", *etc.*, can also be seen in Figure 3, which concretely confirm the correctness of the formulation of label correlations defined in Eq. (13) from semantic perspective.

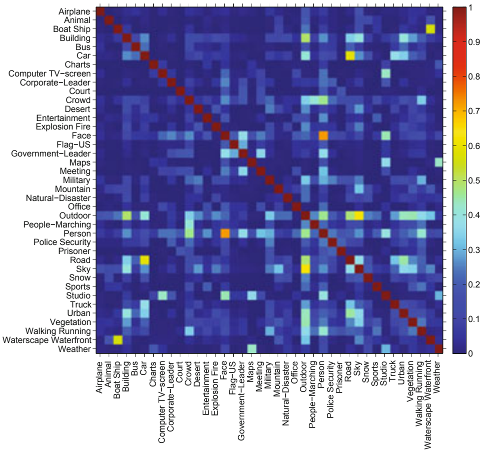


Fig. 3. Correlations between all pairs of 39 keywords in LSCOM-Lite on TRECVID 2005 data set

We replace Y by YC in Eqs. (6–9) in calculation of class-wise scatter matrices to incorporate label correlations. Theorems 1 still holds, because in single-label classification $C = I$ thereby $YC = Y$. Theorems 2 also holds, because we introduce C in both sides of equations.

3.3 Over-Counting Correction

Our further analysis on the class-wise scatter matrices in Eqs. (6–8) shows that the data points with multiple labels are over-counted in the scatter matrices calculations. For example, because data point $\mathbf{a}$ in Figure 2(b) has two labels for class 1 and class 2, it is used in both $S_b^{(1)}$ and $S_b^{(2)}$. Because $S_b = S_b^{(1)} + S_b^{(2)} + S_b^{(3)}$, data point $\mathbf{a}$ is used twice in the between-class scatter matrix S_b. Similarly, data point $\mathbf{c}$ is used three times in both S_b and S_w. In general, data point $\mathbf{x}_i$ with multiple labels is used $\sum_{k=1}^{K} \mathbf{y}_i\left(k\right)$ times in the scatter matrices, which are over-counted compared to data points associated with only one single label.

We correct the over-counting problem by introducing the normalized matrix $Z = \left[\mathbf{z}_1, \ldots, \mathbf{z}_n\right]^T \in \mathbb{R}^{n \times K}$:

$$\mathbf{z}_i = \mathbf{y}_i C / \|\mathbf{y}_i\|_{\ell 1}, \tag{15}$$

where $\|\cdot\|_{\ell 1}$ is the $\ell 1$-norm of a vector. A similar normalization could be as following:

$$\mathbf{z}_i = \mathbf{y}_i C / \|\mathbf{y}_i C\|_{\ell 1}, \tag{16}$$

such that $\sum_{k=1}^{K} \mathbf{z}_i\left(k\right) = 1$ and every data point has same importance in scatter matrices calculation. However, this is not reasonable for multi-label data when label correlations are considered, because a data point with multiple labels is generally believed to convey more information than that with only one single label. For example, in image annotation for natural scene pictures, a picture annotated with labels "Antarctica + penguin" is likely to contain more information than another one annotated with only label "Antarctic". Note that, $\sum_{k=1}^{K} \mathbf{z}_i\left(k\right) \geq 1$ when the *correlated normalized weight* in Eq. (15) is used, *i.e.*, the more labels a data point are associated with, the more important it is. Therefore, instead of using Eq. (16), in this work, we use the normalization in Eq. (15) to deal with the over-counting problem in multi-label data.

By replacing Y by Z in Eqs. (6–9), we have the final MLDA *scatter matrices.* Again, Theorems 1 and 2 can be similarly proved.

3.4 MLDA for Multi-label Classification

Now we write the scatter matrices in a more compact matrix form and summarize our MLDA method. First, let

$$\tilde{X} = X - \mathbf{m}\mathbf{e}^T, \tag{17}$$

where $\mathbf{e} = \left[1, \ldots, 1\right]^T$. Eq. (17) centers input data in multi-label sense, which is different from data centering in classical LDA for single-label classification where $\tilde{X} = X\left(I - \mathbf{e}\mathbf{e}^T/n\right)$.

We define $W = \operatorname{diag}(w_1, \ldots, w_K)$, where $w_k = \sum_{i=1}^{n} Z_{ik}$ is the weight of the k-th class in data scatters. Obviously, in single-label classification, $w_k = n_k$ is the number of data points in the k-th class. Thus,

$$S_b = \tilde{X} Z W^{-1} Z^T \tilde{X}^T. \tag{18}$$

Let $L = \operatorname{diag}(l_1, \ldots, l_n)$, where $l_i = \sum_{i=1}^{K} Z_{ik}$. Clearly, in single-label classification, $L = I$, because each data point only belongs to one class. Thus,

$$S_t = \tilde{X} L \tilde{X}^T. \tag{19}$$

Finally, the optimization objective of our proposed MLDA method is defined in a similar way to classical LDA using trace of matrix ratio as following:

$$\arg\max_G \mathbf{tr}\left(\frac{G^T S_b G}{G^T S_w G}\right). \tag{20}$$

In real life applications, the number of features of a data set is often greater than that of training samples, thus S_w could be singular. As a result, in our implementation, we solve the eigenvalue problem $S_w^+ S_b \mathbf{v}_k = \lambda_k \mathbf{v}_k$, where S_w^+ is the pseudo-inverse of S_w. G is thus constructed by taking the eigenvectors corresponding to the r largest eigenvalues, and the classification can be carried out on the projected data.

4 Connections to Related Works

We review several most recent related multi-label classification methods which also use dimensionality reduction. First of all, many of these algorithms involve XYY^TX^T by certain forms in their optimization objectives, we thereby examine it in some details.

First, because $X\mathbf{y}_{(k)} = \sum_{\mathbf{x}_i \in \pi_k} Y_{ik}\mathbf{x}_i = w_k \mathbf{m}_k$, the following is held:

$$XYY^TX^T = \textstyle\sum_{k=1}^{K} w_k^2 \mathbf{m}_k \mathbf{m}_k^T. \tag{21}$$

When the input data are properly centered as in Eq. (17), the between-class scatter matrix can be written as $S_b = \sum_{k=1}^{K} w_k \mathbf{m}_k \mathbf{m}_k^T$, thus XYY^TX^T is a coarse approximation of S_b. They are equivalent only if every class has same number of data points, *i.e.* $n_i = n_j$, $\forall i, j$.

Second, but more important, they treat the classes in a multi-label data set as independent, thereby label correlations, C, is not employed, though they are very important to enhance classification performance.

MLSI. Yu *et al.* [16] extended unsupervised latent semantic indexing (LSI) to make use of supervision information, called Multi-label informed Latent Semantic Indexing (MLSI) method using (in our notation)

$$\begin{aligned} &\arg\max_G \mathbf{tr}\left(G^T\left((1-\beta)XX^TXX^T + \beta XYY^TX^T\right)G\right) \\ &\qquad s.t. \quad G^TXX^TG = I. \end{aligned} \tag{22}$$

The first term is the original LSI objective. The second term is the supervised regularizer, which implicitly approximates S_b with deficiencies as analyzed above.

MDDM. Zhang *et al.* [17] proposed Multi-label Dimensionality reduction via Dependence Maximization (MDDM) method to identify a lower-dimensional subspace by maximizing the dependence between the original features and associated class labels through (in our notation)

$$\max_{G} \mathbf{tr}\left(G^T X H Y Y^T H X^T G\right), \tag{23}$$

where $H = I - \mathbf{e}\mathbf{e}^T/n$ is the centering matrix in single-label sense such that XH has zero mean. However, the correct data centering in multi-label classification should be as in Eq. (17) and is different from XH. Ignoring H, Eq. (23) is same as Eq. (21), which simulates S_b without taking advantage of its full potentials.

MLLS. Ji *et al.* [3] suggested Multi-Label Least Square (MLLS) method to extract a common structure (subspace) shared among multiple labels. The optimization objective is (in our notation):

$$\begin{aligned} &\max_{G} \mathbf{tr}\left(G^T \left(I - \alpha M\right)^{-1} \left(M^{-1} X Y Y^T X^T M^{-1}\right) G\right) \\ &M = \frac{1}{n} X X^T + (\alpha + \beta) I. \end{aligned} \tag{24}$$

Eq. (24) still fundamentally relies on XYY^TX^T to use label information, though more complicated.

We will compare the proposed MLDA method with these related algorithms in next evaluation section.

We notice another two recent works in [8] and [4] have close titles with our paper. However, the former attempts to solve multi-label classification implicitly through QR-decomposition in the null space of S_w, which is far more complicated than our method. Most importantly, label correlations are not considered in this work. The latter incorporates discriminative dimensionality reduction into Support Vector Machine (SVM), and thereby fundamentally different from the proposed MLDA method. In summary, our MLDA method present a generic framework for solving multi-label problems, which naturally incorporates label correlations inherent in multi-label data.

5 Experimental Results

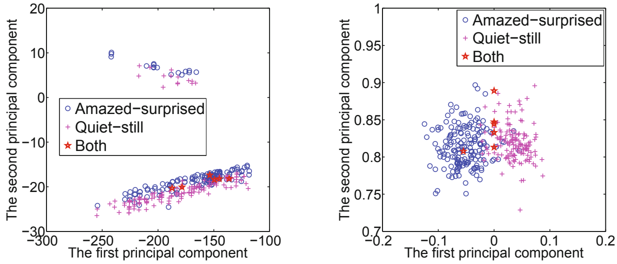
To evaluate the performance of multi-label classification methods, we use both basic image features (such as pixel values and moments of colors) and SIFT features in image classifications. We validate the proposed MLDA methods using the following standard multi-label data sets for image annotation.

TRECVID 2005[1] data set contains 61901 images and labeled with 39 concepts (labels). As most previous works, we randomly sample the data such that each concept has at least 100 images.

MSRC[2] data set is provided by the computer vision group at Microsoft Research Cambridge, which has 591 images annotated by 22 classes.

[1] http://www-nlpir.nist.gov/projects/trecvid

[2] http://research.microsoft.com/en-us/projects/objectclassrecognition/default.htm



(a) Visualization on 2D plane in the original space ($p = 72$).

(b) Visualization on 2D plane in the reduced subspace ($l = 5$) by MLDA.

Fig. 4. Visualization on 2D plane for the data points from the two classes in music emotion data set, in original space and projected space by MLDA, respectively

For these two image data sets, we divide each image into 64 blocks by a 8×8 grid and compute the first and second moments (mean and variance) of each color band to obtain a 384-dimensional vector as features. For MSRC data, we also employ SIFT features to measure similarities between images. We use MSRC (SIFT) to refer this data.

Mediamill [12] data set includes 43907 sub-shots with 101 classes, where each image is characterized by a 120-dimensional vector. Eliminating the classes containing less than 1000 samples, we have 27 classes. We randomly select 2609 sub-shots such that each class has at least 100 labeled data points.

In order to justify the generic applicability of our method, we also evaluate all methods on two following data sets from different applications.

Music emotion [13] data set comprises 593 songs with 6 emotions (labels). The dimensionality of the data points is 72.

Yahoo data set described in [14] came from the "yahoo.com" domain. Each web page is described as a 37187-dimensional feature vector. We use the "science" topic because it has maximum number of labels, which contains 6345 web pages with 22 labels.

5.1 Discriminative Capability of MLDA

We first evaluate the discriminative capability of the proposed MLDA method. We randomly pick up two classes from music emotion data set, "amazed-surprised" and "quiet-still", and visualize the data points from these two classes in the original space ($p = 72$) on 2D plane using the first two principal component coordinates as shown in Figure 4(a). It is obvious that the data points are mingled together and it is difficult to find a linear decision boundary with high classification accuracy. We then run MLDA on the whole data set with all six labels, and transform the data points into the obtained projection subspace ($l = K - 1 = 5$), in which we visualize the same data points on 2D plane as shown in Figure 4(b). Apparently, the

Table 1. Performance evaluations of six compared methods by 5-fold cross validations

Data	Evaluation metrics		Compared methods					
			LDA-C1	SVM	MLSI	MDDM	MLLS	MLDA
TREC05	Macro average	Precision	0.282	0.269	0.247	0.366	0.248	**0.420**
		F1 score	0.190	0.286	0.275	0.370	0.276	**0.399**
	Micro average	Precision	0.274	0.252	0.234	0.352	0.241	**0.418**
		F1 score	0.408	0.399	0.293	0.491	0.295	**0.528**
MSRC	Macro average	Precision	0.291	0.274	0.252	0.370	0.255	**0.431**
		F1 score	0.201	0.295	0.287	0.392	0.290	**0.410**
	Micro average	Precision	0.288	0.262	0.253	0.363	0.255	**0.420**
		F1 score	0.415	0.406	0.301	0.504	0.302	**0.533**
MediaMill	Macro average	Precision	0.337	0.302	0.207	0.385	0.206	**0.410**
		F1 score	0.349	0.322	0301	0.418	0.311	**0.430**
	Micro average	Precision	0.335	0.297	0.207	0.382	0.205	**0.388**
		F1 score	0.518	0.398	0.341	0.440	0.340	**0.443**
Music emotion	Macro average	Precision	0.507	0.434	0.329	0.509	0.311	**0.614**
		F1 score	0.453	0.418	0.323	0.506	0.471	**0.618**
	Micro average	Precision	0.504	0.501	0.328	0.507	0.308	**0.613**
		F1 score	0.477	0.441	0.339	0.518	0.475	**0.626**
Yahoo (Science)	Macro average	Precision	0.458	0.414	0.396	0.463	0.421	**0.501**
		F1 score	0.227	0.302	0.296	0.481	0.443	**0.498**
	Micro average	Precision	0.447	0.416	0.395	0.458	0.420	**0.499**
		F1 score	0.226	0.218	0.209	0.484	0.519	**0.544**
MSRC (SIFT)	Macro average	Precision	0.415	0.408	0.428	0.520	0.424	**0.612**
		F1 score	0.367	0.358	0.381	0.471	0.376	**0.531**
	Micro average	Precision	0.408	0.403	0.412	0.515	0.407	**0.597**
		F1 score	0.612	0.611	0.620	0.671	0.617	**0.698**

data points are clearly separated according to their class membership now, which demonstrates that the projection subspace produced by MLDA is indeed more discriminative. In addition, MLDA significantly reduces the data dimensionality (from 72 to 5), such that the computational complexity of the subsequent classification is largely reduced.

5.2 Classification Performance

We use standard 5-fold cross validation to evaluate the classification performance of the proposed MLDA method, and compare the results to the three related multi-label classification methods, MLSI, MDDM, and MLLS discussed in Section 4. K-Nearest Neighbor (KNN) classifier ($K = 1$ in this work) is used for classification after dimensionality reduction by MLSI, MDDM, and MLDA methods. We also tested $K = 3, 5$ and the results are similar to $K = 1$. Because of the limited space, we only show the results of $K = 1$. Euclidean distance is used to decide neighborhood in KNN. KNN is conducted one class at a time, where a binary classification is conducted for each class. Note that, we choose KNN because it is the most widely used classification method following standard LDA. Because multi-label problem is already addressed in the dimensionality reduction step in MLSI, MDDM and our method, the subsequent classification method, such as KNN in our evaluations, do not need to take care of multi-label issue any longer. MLLS has its own classification mechanism. Following the standard way, we select $l = K - 1$ as the dimensionality of the projected subspace. For MLSI, the parameter β is set as 0.5 as recommended in [16]. For MDDM, we use the same linear kernel as in the experimental evaluation in [17]. For MLLS, we use the codes posted at the authors' web site [3], which fine tunes the parameters based on F1 scores.

LDA-C1. We report the classification performance of classical LDA as a reference. Because classical LDA is inherently a single-label classification method, we do both dimensionality reduction and classification one class at a time. For every class, the classification is done as a binary classification problem, which thereby implicitly treats all the classes isolated.

Support Vector Machine (SVM). We use SVM classification results as a baseline. Similar to LDA-C1, we run SVM one class at a time, and for every class the classification is done as a binary classification problem. SVM is implemented by LIBSVM[3] (Matlab version).

The conventional classification performance metrics in statistical learning, *precision* and *F1 score*, are used to evaluate the compared methods. Precision and F1 score are computed for every class following the standard definition for a binary classification problem. To address multi-label classification, class-wise macro average and micro average are used to assess the overall performance across multiple labels [6]. In multi-label classification, the macro average is the mean of the values of a standard class-wise metric over all the labels, thus attributing equal weights to every class. The micro average is obtained from the summation of contingency matrices for all binary classifiers. The micro average metric gives equal weight to all classifications, which can be seen as a weighted average that emphasizes more on the accuracy of categories with more positive samples.

Table 1 presents the classification performance comparisons by 5-fold cross validation, which show that the proposed MLDA method generally outperforms all other methods, sometimes significantly. We achieve about 10% improvement on average over all the data sets. To be more specific, Figure 1 shows four images from TRECVID 2005. Only the proposed MLDA method can correctly annotate all images. All other methods only predict part of the labels. These results quantitatively demonstrate the effectiveness of our method, and justify the utility of the class-wise scatter matrices and label correlations.

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

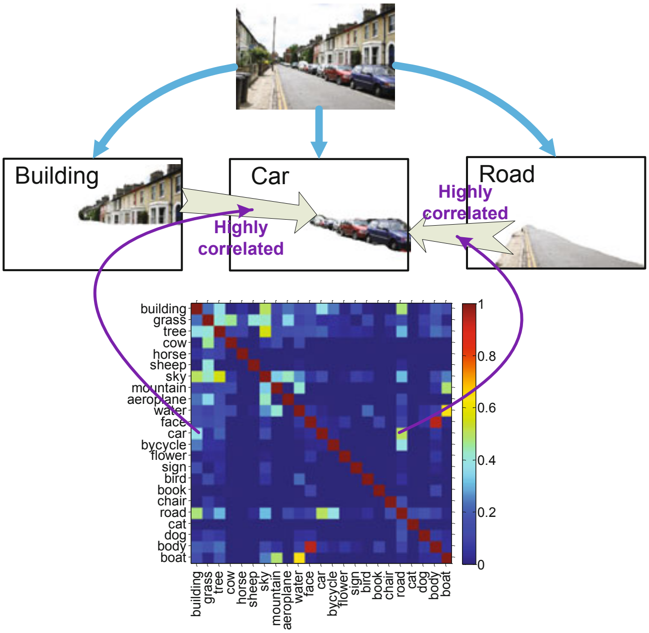


Fig. 5. An example image from MSRC data (Top). The label "car" can only be correctly annotated by our MLDA method, because "car" has high correlations with "building" and "road" (appearing in the image). The bottom panel visualizes the label correlation matrix.

5.3 Label Transfer via Label Correlations

A more careful examination on the classification results in Section 5.2 shows that, for the sample image shown in the top panel of Figure 5 from MSRC data set, the label "car" can only be correctly annotated by the proposed MLDA method, while two other labels, "building" and "road", generally can be correctly annotated by most of the compared methods. By scrutinizing label correlations of MSRC data set, defined by Eq. (13) and illustrated in the bottom panel of Figure 5, we can see that "car" is highly correlated with both "building" and "road". Therefore, label "car" is transferred to the sample image from its annotated labels through label correlations, which concretely corroborates the usefulness of label correlations to boost multi-label classification performance.

6 Conclusions

In this work, we proposed a novel Multi-label Linear Discriminant Analysis (MLDA) method to naturally generalize classical LDA for multi-label classification. We reformulated the scatter matrices from class perspective, such that the new class-wise scatter matrices solved the computation ambiguity to use traditional single-label definitions of scatter matrices in multi-label classification, and incorporated label correlations from multi-label data. We examined three closely related multi-label classification methods and showed the advantages of our method theoretically. Encouraging results in extensive experimental evaluations supported our proposed methods and theoretical analysis empirically.

Acknowledgments. This research is supported by NSF-CCF 0830780, NSF-CCF 0939187, NSF-CCF 0917274, NSF-DMS 0915228, NSF-CNS 0923494.

References

1. Chen, G., Song, Y., Wang, F., Zhang, C.: Semi-supervised Multi-label Learning by Solving a Sylvester Equation. In: Proc. of SDM (2008)
2. Fukunaga, K.: Introduction to statistical pattern recognition (1990)
3. Ji, S., Tang, L., Yu, S., Ye, J.: Extracting shared subspace for multi-label classification. In: Proc. of SIGKDD, pp. 381–389 (2008)
4. Ji, S., Ye, J.: Linear Dimensionality Reduction for Multi-label Classification. In: Proc. of IJCAI, pp. 1077–1082 (2009)
5. Kang, F., Jin, R., Sukthankar, R.: Correlated label propagation with application to multi-label learning. In: Proc. of CVPR, pp. 1719–1726 (2006)
6. 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)
7. Liu, Y., Jin, R., Yang, L.: Semi-supervised multi-label learning by constrained non-negative matrix factorization. In: Proc. of AAAI, p. 421 (2006)
8. Park, C., Lee, M.: On applying linear discriminant analysis for multi-labeled problems. Pattern Recognition Letters 29(7), 878–887 (2008)
9. Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of continuous multi-label problems. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 792–805. Springer, Heidelberg (2008)
10. Qi, G., Hua, X., Rui, Y., Tang, J., Mei, T., Zhang, H.: Correlative multi-label video annotation. In: Proc. of ACM Multimedia, pp. 17–26 (2007)
11. Smeaton, A., Over, P., Kraaij, W.: Evaluation campaigns and TRECVid. In: Proc. of MIR, p. 330 (2006)
12. 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, pp. 421–430 (2006)
13. Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: Multilabel classification of music into emotions. In: Proc. of ISMIR (2008)
14. Ueda, N., Saito, K.: Single-shot detection of multiple categories of text using parametric mixture models. In: Proc. of SIGKDD, pp. 626–631 (2002)
15. Wang, H., Huang, H., Ding, C.: Image Annotation Using Multi-label Correlated Greens Function. In: Proc. of ICCV (2009)
16. Yu, K., Yu, S., Tresp, V.: Multi-label informed latent semantic indexing. In: Proc. of SIGIR, p. 265 (2005)
17. Zhang, Y., Zhou, Z.: Multi-Label Dimensionality Reduction via Dependence Maximization. In: Proc. of AAAI, pp. 1503–1505 (2008)

Convolutional Learning of Spatio-temporal Features

Graham W. Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler

Courant Institute of Mathematical Sciences, New York University
New York, USA
{gwtaylor,fergus,yann,bregler}@cs.nyu.edu

Abstract. We address the problem of learning good features for understanding video data. We introduce a model that learns latent representations of image sequences from pairs of successive images. The convolutional architecture of our model allows it to scale to realistic image sizes whilst using a compact parametrization. In experiments on the NORB dataset, we show our model extracts latent "flow fields" which correspond to the transformation between the pair of input frames. We also use our model to extract low-level motion features in a multi-stage architecture for action recognition, demonstrating competitive performance on both the KTH and Hollywood2 datasets.

Keywords: unsupervised learning, restricted Boltzmann machines, convolutional nets, optical flow, video analysis, activity recognition.

1 Introduction

While the dominant methodology for visual recognition from images and video relies on hand-crafted features, there has been a growing interest in methods that learn low-level and mid-level features, either in supervised [1], unsupervised [2,3,4], or semi-supervised settings [5]. In recent years, feature-learning methods have focused on learning multiple layers of feature hierarchies to extract increasingly abstract representations at each stage. This has been generally done by composing modules of the same architecture such as Restricted Boltzmann Machines (RBM) [2], autoencoders [3], or various forms of encoder-decoder networks [4,6,7] each of which are trained unsupervised and therefore can take advantage of large amounts of unlabeled image data. The resulting "deep architectures" are then globally trained discriminatively, with the idea that the first phase of unsupervised feature learning has provided an initialization that is much more salient for high-level tasks than the usual random initialization.

Most of the above methods do not exploit the pictorial nature of the input, and have been applied to relatively small image patches (typically less than 64×64 pixels), because they do not scale well with the size of the input. This can be addressed by using a *convolutional architecture* [1], which exploits the fact that salient motifs can appear anywhere in the image. This idea has been recently used in the context of RBMs [8,9]. By employing successive stages of weight-sharing

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 140–153, 2010.

and feature-pooling, deep convolutional architectures can achieve stable latent representations at each layer, that preserve locality, provide invariance to small variations of the input, and drastically reduce the number of free parameters.

To date, most of the work on unsupervised feature extraction has focused on static images but little attention has been given to learning about the way that images from videos change over time. The few works that address the problem (e.g. [10,6]) are trained on isolated patches (not convolutionally), and suffer from the same limitations as static methods. In this paper, we propose a model that can extract motion-sensitive features from pairs of images (i.e. neighbouring frames of video). The features can capture both static and dynamic content. Our model is trained convolutionally which enables it to work on high-resolution images. We first apply it to synthetic data and show that it learns to represent flow-like features when the type of transformations are restricted. We then use it to extract useful features for human activity recognition in a multi-stage architecture that achieves state-of-the-art performance on the KTH actions dataset. Results are also shown on the challenging Hollywood2 action recognition dataset.

2 Related Work

Our work extends the Gated RBM (GRBM) model proposed by Memisevic and Hinton [10]. The GRBM is able to extract distributed, domain-specific representations of image patch transformations. Due to its tensor parameterization, it is not practical to apply this model to patches larger than about $(N = 32) \times 32$ since the number of parameters grows as $O(N^4)$. Therefore, it has only been applied to low-resolution synthetic images of shifting pixels or PCA-reduced samples of low-resolution video. While the model has been shown to improve digit classification by learning the types of transformations to which the classifier should remain invariant, we are not aware of is application to a discriminative task on real video. Memisevic and Hinton have recently proposed a factored form of the GRBM [11] that drastically reduces the number of free parameters by replacing the three-way weight tensor with three low-rank matrices. In the present work, we take an alternative convolutional approach to scaling up the model, which achieves the additional benefit of translation invariance. Sutskever and Hinton [12] proposed a type of temporal RBM for video. Using synthetic videos of bouncing balls, they trained a model which was then able to generate similar videos, but did not apply their work to discriminative tasks. The signal from the past only provides a type of "temporal bias" to the hidden variables, which is fundamentally different from our third-order RBM, where past inputs modulate the interactions between the current input and the latent feature representation.

Building on the rapidly growing literature on sparse over-complete decompositions of image patches [13], Cadieu and Olshausen [6] have proposed a two-layer probabilistic model that learns complex motion features from video. In contrast to our model, they explicitly separate static amplitude and dynamic phase at the first layer. The second layer then learns high-order dependencies among the phase variables. Dean et al. [14] have recently proposed learning spatio-temporal descriptors by recursively applying the feature-sign sparse coding algorithm [15] to 3D patches of videos extracted at detected interest points. Like our work, their descriptors are adaptive, but their method is trained at the patch level.

State-of-the-art methods for activity recognition use engineered motion and texture descriptors extracted around interest points detected by spatio-temporal corner detectors. The descriptors are then vector-quantized, pooled over time and space into a "bag", and fed to an SVM classifier. Among the best performing methods are 1) Laptev et al.'s spatio-temporal interest points (STIP) [16] used in conjunction with the "HOG/HOF" descriptor that computes histograms of spatial gradients and optic flow accumulated in local space-time neighbourhoods [17]; 2) Dollar et al.'s "Cuboids" approach [18] used in conjunction with several different descriptor types; and 3) Willems et al.'s approach [19] which uses the determinant of the Hessian as a saliency measure and computes a weighted sum of Haar wavelet responses within local rectangular sub-volumes.

In contrast to these approaches, we perform a type of implicit, rather than explicit interest point detection and focus on learning descriptors rather than hand-crafting them. We also bypass the quantization step in favor of several additional layers of feature extraction that provide a distributed representation of each video. Jhuang et al. [20] propose an approach similar in spirit to ours, using multiple levels of feature detectors at increasing spatio-temporal scale. However, like [17,18,19], they forgo learning until the very last stage: low and mid-level features are engineered.

3 Unsupervised Learning of Spatio-temporal Features

We first describe a related approach, the gated Restricted Boltzmann Machine, which models image patches but does not scale to realistic-sized images or video. We then describe our convolutional model.

3.1 The Gated Restricted Boltzmann Machine (GRBM)

The gated Restricted Boltzmann Machine [10] differs from other conditional RBM architectures (e.g. [21,12]) in that its inputs change the effective *weights* of the model instead of simply adjusting the effective *biases* of visible or latent variables (see Figure 1(left)). This is achieved by defining an energy function that captures third-order interactions among three types of binary stochastic variables: inputs, $\mathbf{x}$, outputs, $\mathbf{y}$, and latents, $\mathbf{z}$:

$$E\left(\mathbf{y},\mathbf{z};\mathbf{x}\right) = -\sum_{ijk} W_{ijk} x_i y_j z_k - \sum_k b_k z_k - \sum_j c_j y_j \tag{1}$$

where W_{ijk} are the components of a parameter tensor, $\mathbf{W}$, which is learned. To model affine and not just linear dependencies, biases $\mathbf{b}$ and $\mathbf{c}$ are included.

When learning from video, $\mathbf{x}$ and $\mathbf{y}$ are image patches (expressed as vectors) at identical spatial locations in sequential frames, and $\mathbf{z}$ is a latent representation

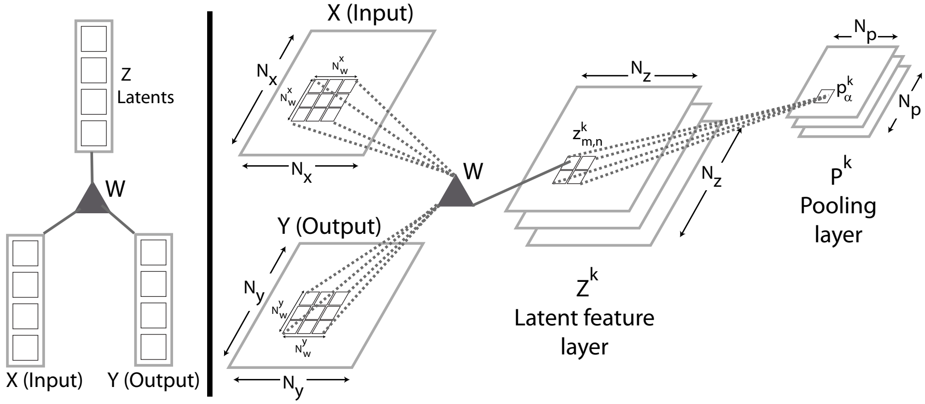


Fig. 1. Left: A gated RBM. Right: A convolutional gated RBM using probabilistic max-pooling.

of the transformation between $\mathbf{x}$ and $\mathbf{y}$. The energy of any joint configuration $\{\mathbf{y}, \mathbf{z}; \mathbf{x}\}$ is converted to a conditional probability by normalizing:

$$p(\mathbf{y}, \mathbf{z}|\mathbf{x}) = \exp\left(-E(\mathbf{y}, \mathbf{z}; \mathbf{x})\right) / Z(\mathbf{x}) \tag{2}$$

where the "partition function", $Z(\mathbf{x}) = \sum_{\mathbf{y},\mathbf{z}} \exp\left(-E(\mathbf{y}, \mathbf{z}; \mathbf{x})\right)$ is intractable to compute exactly since it involves a sum over all possible configurations of the output and latent variables. However, we do not need to compute this quantity to perform either inference or learning. Given an input-output pair of image patches, $\{\mathbf{x}, \mathbf{y}\}$, it follows from Eq. 1 and 2 that

$$p(z_k = 1|\mathbf{x}, \mathbf{y}) = \sigma(\sum_{ij} W_{ijk} x_i y_j + b_k) \tag{3}$$

where $\sigma(z) = 1/(1 + \exp(-z))$ is the logistic.

Maximizing the marginal conditional likelihood, $p(\mathbf{y}|\mathbf{x})$, over parameters $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ is difficult for all but the smallest models due to the intractability of computing Z. Learning, however, still works well if we approximately follow the gradient of another function called the contrastive divergence (CD) [22].

3.2 The Convolutional Gated Restricted Boltzmann Machine (convGRBM)

GRBMs represent the input and output as a vector, and thus ignore the pictorial structure of images. Weights that encode a particular local transformation must be re-learned to detect that same transformation at multiple locations. We now describe a form of GRBM that shares weights at all locations in an image. Inference is performed efficiently through convolution, so we refer to the model as a convolutional GRBM (convGRBM). The model is illustrated in Figure 1(right).

In our description of the GRBM, we suggested that $\mathbf{x}$ and $\mathbf{y}$ were time-adjacent patches from video, but they could have been any arbitrary vectors. Here, we assume that $\mathbf{x}$ is a $N_x \times N_x$ binary image and $\mathbf{y}$ is a $N_y \times N_y$ binary image. We assume square, binary images to simplify the presentation but provide details of using real-valued images in the supplemental material. In the GRBM we had K binary latent variables. Now we have K $N_z \times N_z$ binary latent feature maps ($\mathbf{z} = \{\mathbf{z}^k\}_{k=1}^K$). Let m and n be spatial indices to each 2D feature map, such that a single feature is described as $z^k_{m,n}$. The indices m and n not only index a particular 2D feature, but they also define 1) an $N^y_w \times N^y_w$ local region in $\mathbf{y}$ from which this feature receives input, and 2) a $N^x_w \times N^x_w$ region of $\mathbf{x}$ which modulates the interaction between all K features at location m, n and the $N^y_w \times N^y_w$ local region in $\mathbf{y}$. Alternatively, we can think of each of the K features at index m, n as contributing a local log-linear patch model between the $N^x_w \times N^x_w$ pixels in $\mathbf{x}$ and the $N^y_w \times N^y_w$ pixels in $\mathbf{y}$ where the location of these local regions is specified by m, n. The number of local autoregressive models that can be "blended" is exponential in the number of feature maps.

For the remainder of our discussion, we will make two assumptions: 1) the input and output images are the same dimensions, $N_x = N_y$ (this holds true for neighbouring frames in video); and 2) the filter dimensions in the input and the output are the same, $N^x_w = N^y_w$. These assumptions are not necessary, but they greatly simplify bookkeeping and therefore the presentation that follows.

The convGRBM has the following energy function:

$$E\left(\mathbf{y}, \mathbf{z}; \mathbf{x}\right) = -\sum_{k=1}^{K} \sum_{m,n=1}^{N_z} \sum_{r,s=1}^{N^y_w} z^k_{m,n} \gamma(\mathbf{x})^k_{r,s,m,n} y_{m+r-1,n+s-1} - \sum_{k=1}^{K} b_k \sum_{m,n=1}^{N_z} z^k_{m,n} - c \sum_{i,j=1}^{N_y} y_{i,j} \tag{4}$$

where we use a per-map bias, b_k, for the latent variables and single output bias, c. Eq. 4 is similar to the energy function of a convolutional RBM [8], except that what was previously a filter weight with 3 indices: r, s, k has been replaced by a *conditional* filter weight, $\gamma(\mathbf{x})^k_{r,s,m,n} = \sum_{u,v}^{N^x_w} W^k_{r,s,u,v} x_{m+u-1,n+v-1}$, with 5 indices. The additional indices m, n denote the local region in $\mathbf{x}$ which modulates the filter. Note that while m, n index the entire feature map, u, v and r, s index within the local regions of $\mathbf{x}$ and $\mathbf{y}$, respectively.

As in the GRBM, the probability of jointly observing $\mathbf{y}$ and $\mathbf{z}$ given $\mathbf{x}$ is given by Eq. 2. The conditional distributions for $\mathbf{z}|\mathbf{y}, \mathbf{x}$ and $\mathbf{y}|\mathbf{z}, \mathbf{x}$ naturally follow:

$$p(z^k_{m,n} = 1|\mathbf{x}, \mathbf{y}) = \sigma(\sum_{r,s=1}^{N^y_w} \gamma(\mathbf{x})^k_{r,s,m,n} y_{m+r-1,n+s-1} + b_k) \tag{5}$$

$$p(y_{i,j} = 1|\mathbf{x}, \mathbf{z}) = \sigma(\sum_{k=1}^{K} \sum_{r,s=1}^{N^y_w} \hat{\gamma}(\mathbf{x})^k_{r',s',i+r-1,j+s-1} \hat{z}^k_{i+r-1,j+s-1} + c) \tag{6}$$

where $r' = N^y_w - r + 1$ and $s' = N^y_w - s + 1$ represent a "flipping" of the filter indices (i.e. correlation rather than convolution), and $\hat{\mathbf{z}}$ is the result of zero-padding $\mathbf{z}$ such that its first $N_w^y - 1$ rows and columns are zero. Note that in Eq. 5 an output unit $y_{i,j}$ makes a bottom-up contribution to several elements (m, n) in all K feature maps. Therefore, in top-down reconstruction (Eq. 6) we must ensure that each output unit receives input from all feature map elements to which it has contributed, through the same conditional filter weight that was used bottom-up. To account for border effects, it is convenient to define $\hat{\gamma}(\mathbf{x})$ as a zero-padded version of $\gamma(\mathbf{x})$ whose dimensions are $N_w^y \times N_w^y \times N_y \times N_y \times K$.

As with convolutional RBMs, we can express both inference (Eq. 5) and reconstruction (Eq. 6) in terms of convolution operations (see the supplemental material for details). While inference in a convolutional RBM requires a single 2D convolution of the data with the filters, inference in the convGRBM requires a 2D convolution of the output and data for each element of the conditioning window: i.e. $N_w^x \times N_w^x$ convolutions. The same holds true for reconstruction (replacing data with feature maps). Note, however, that a fully-connected (i.e. non-convolutional) GRBM requires $N_x \times N_x$ more operations during inference than a standard RBM. Restricting connections to be local clearly makes a huge difference in efficiency, especially when the ratio of pixels to filter size is high.

Probabilistic Max Pooling. Most object recognition systems use a pooling operation that combines nearby values in input or feature space through a max, average or histogram operator. This provides the system with some invariance to small local distortions and reduces the computational burden. Traditional pooling layers, however, are designed for feed-forward architectures like convolutional nets and do not support generative models such as RBMs that include top-down feedback. Lee et al. [8] thus introduced probabilistic max-pooling in the context of convolutional RBMs. We adopt their approach, and summarize it here.

Recall that we have K feature maps connected to the visible input and output. We introduce a layer on top of the feature maps, called the pooling layer, which also has K maps, each connected 1-1 to a feature map. However, the maps of the pooling layer are reduced in spatial resolution by a constant factor C in each dimension (e.g. 2 or 4). More precisely, each feature map $\mathbf{z}^k$ is partitioned into non-overlapping $C \times C$ blocks, and each block is connected to exactly one binary unit, p_α^k, in the pooling layer (i.e. $N_p = N_z/C$). Here, we have adopted the notation of [8] where α indexes the pooling units and also define a block formally as $B_\alpha \triangleq \{(m, n) : z_{m,n} \text{ belongs to the block } \alpha\}$.

The connection between pooling unit p_α and the features in block B_α is constrained such that *at most* one of the features in a block is on, and if any of the features in block B_α is on, then p_α must be on, otherwise p_α is off. This leads to a modified, constrained, energy function:

$$E(\mathbf{y}, \mathbf{z}; \mathbf{x}) = -\sum_{k=1}^{K} \sum_{\alpha} \sum_{(m,n) \in B_\alpha} \sum_{r,s=1}^{N_w^y} z_{m,n}^k \gamma(\mathbf{x})_{r,s,m,n}^k y_{m+r-1,n+s-1}$$

$$-\sum_{k=1}^{K} b_k \sum_{m,n=1}^{N_z} z_{m,n}^k - c \sum_{i,j=1}^{N_y} y_{i,j} \quad \text{subject to:} \sum_{(m,n) \in B_\alpha} z_{m,n}^k \leq 1, \forall k, \alpha. \qquad (7)$$

Changing the energy function results in a change to the inference procedure. Note that each unit in feature map k receives the following bottom-up signal from the input and output:

$$I(z^k_{m,n}) \triangleq \sum_{r,s=1}^{N^y_w} \gamma(\mathbf{x})^k_{r,s,m,n} y_{m+r-1,n+s-1} + b_k. \tag{8}$$

Due to the factorial form of Eq. 7, we can sample each of the blocks independently as a multinomial function of their inputs:

$$p(z^k_{m,n} = 1|\mathbf{x}, \mathbf{y}) = \Omega^{-1}\exp\left(I(z^k_{m,n})\right), \qquad p(p^k_\alpha = 0|\mathbf{x}, \mathbf{y}) = \Omega^{-1} \tag{9}$$

where the normalization constant is $\Omega = 1 + \sum_{(m',n')\in B_\alpha} \exp\left(I(z^k_{m',n'})\right)$.

4 Experiments on Synthetic Data: NORB

One way to evaluate third-order RBMs is by experimenting in a domain where optical flow is controlled and regular (e.g. the “shifting pixels” experiments of [10]). In this section, we describe a domain for experimentation that is of increased complexity yet still controlled. The “Small NORB” dataset [23] has 5 object categories (humans, airplanes, cards, trucks, animals), and 5 different object instances for each training and test. Each object instance has 18 azimuths, 9 camera-elevations, and 6 illuminations, for a total of 24300 training samples and 24300 test samples. Traditionally NORB has been used to evaluate object recognition. Since our goal is to extract useful “transformation” features from pairs of images we use the dataset differently than intended.

The azimuth, elevation, and illumination changes in the NORB dataset are at fixed intervals and corresponding labels for each image are available. Therefore, we created synthetic “videos” where an object underwent forward or reverse transformation in one of the dimensions while the others were held fixed. Before generating the videos, we downsampled each image to 32×32 pixels, and preprocessed it using local contrast normalization (LCN) as described in [24]. The LCN operation involves a 9×9 smoothing filter, so each resulting image is 24×24.

We then trained a convGRBM with real-valued outputs and 20 binary feature maps. The filter dimensions were $N^x_w = N^y_w = 9$. The model trained on all azimuth changes of ±20°, and all camera elevation changes of ±10°. It was trained for 30 complete passes through the training set, using standard CD(1) learning. Figure 2 shows the result of performing 10 “image analogies”. Each analogy is represented by a group of six small greyscale images and one larger “optical flow’ image. To perform an analogy, the model is presented with a pair of images each from an object instance it has never seen before, and asked to apply the same inferred transformation to a random target image, also which it has never seen before. We can also visualize the “flow” implicit in the hidden units and conditional on the pair, by drawing, for each input pixel, an arrow to the output pixel to which it is most strongly connected according to the learned

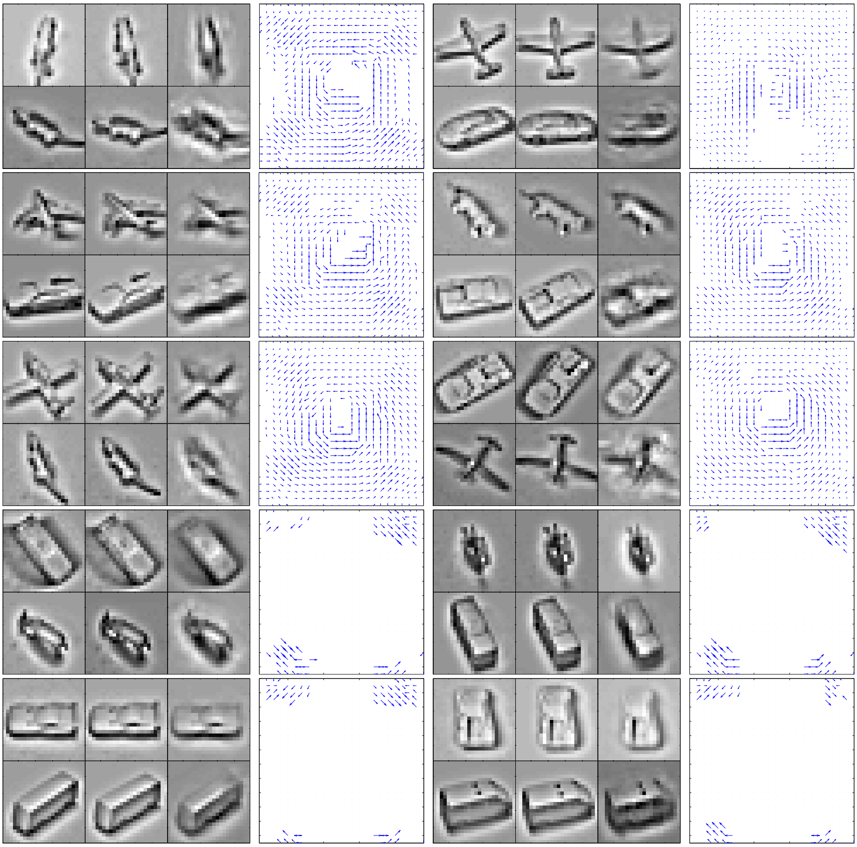


Fig. 2. Analogies. Each group of six greyscale images from left to right, top to bottom represent: input image; output image; model's reconstruction of output; random target image; ground truth of random target (i.e. by searching for the example that corresponds to the transformation between image and output); inferred transformation applied to targets. Examples 1-6 show changes in azimuth; 7-10 show changes in camera elevation. A representation of inferred "max" flow fields is shown for each example.

filters, W (marginalized over the binary feature maps). Much information is potentially lost in this representation [10]: the transformation encoded by the feature maps can be much richer than what is expressed by optical flow alone.

5 Experiments on Human Activity Recognition

Recognition of human activity from video is a challenging problem that has received an increasing amount of attention from the computer vision community in recent years. The ability to parse high-level visual information has wide-ranging

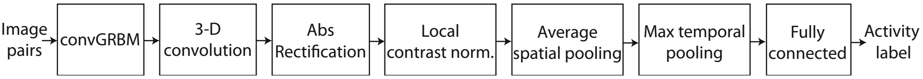


Fig. 3. An overview of our multi-stage architecture for human activity recognition. See text for a description of each stage.

applications that include surveillance and security, the aid of people with special needs and the understanding and interpretation of non-verbal communication.

We approach the problem with a multi-stage architecture (see Figure 3) that combines convolutional and fully-connected layers. At the lowest layer, a convolutional GRBM extracts features from every successive pair of frames. We observe that most features are motion-sensitive, but others capture static information. This is particularly useful in providing context in more challenging datasets [25] and will aid in applying our method to other tasks, such as scene recognition from video. A subset of the feature maps inferred from the KTH actions dataset are shown in Figure 4. The features are extremely diverse: many capture limb movement, others capture edge content, and one seems particularly apt at segmenting person from background (we note that the background is generally uniformly textured in KTH).

To capture mid-level spatio-temporal cues, we apply a traditional (i.e. feed-forward) convolutional layer that uses 3D spatio-temporal filters. A connectivity table indicates which of the 3D convolutional layer output maps are connected to each convGRBM pooling map. Our convolutional layer is a 3D extension of the architecture advocated by Jarrett et al. [7]: filtering, followed by a tanh nonlinearity, followed by absolute value rectification, followed by a local contrast normalization layer, followed by average pooling and subsampling. Both the abs($\cdot$) and tanh($\cdot$) are performed element-wise, so their extension to 3D is straightforward. The LCN and pooling/subsampling layers each employ a filtering operation, which we perform in 3D instead of 2D.

The output of the second convolutional layer is a series of 3D feature maps. To cope with variable-length sequences, we perform an additional max pooling in the temporal dimension. This ensures that the mid-level features can be reduced to a vector of consistent size. This representation is followed by one or more fully-connected layers (we use 1 or 2 in our experiments). The topmost layer is a softmax (multinomial) layer corresponding to discrete activity labels, and intermediate layers use a tanh nonlinearity. The convGRBM is trained unsupervised using CD, while the upper layers are trained by backpropagation. We do not backpropagate through the first layer following unsupervised training, though this could be done to make the low-level features more discriminative.

5.1 KTH Actions Dataset

The KTH actions dataset [26] is the most commonly used dataset in evaluating human action recognition. It consists of 25 subjects performing six actions: walking, jogging, running, boxing, hand waving, and hand clapping under 4 scenarios (outdoors, outdoors with scale variation, outdoors with different clothes and indoors). Each sequence is further divided into shorter "clips" for a total of 2391 sequences. We use the original evaluation methodology: assigning 8 subjects to a training set, 8 to a validation set, and the remaining 9 subjects to a test set so that our results are directly comparable to the recent survey by Wang et al. [27].

Preprocessing. We maintained the original frame rate (25fps) and spatial resolution 160×120 in all of our experiments. All videos then underwent 3D local contrast normalization (an extension of [24]).

Unsupervised Learning. We trained a convGRBM with $N_z = 32$ feature maps and a pooling factor of $C = 4$. Filter sizes were $N_x^x = N_x^y = 16$. We chose 16 as it was a number amenable to GPU-based computing, and it was close to the minimal patch size (18×18) suggested by Wang et al. [27]. We have not tried other patch sizes. Weights were updated in "mini-batches" of 128 pairs of subsequent frames (the order of pairs was randomly permuted as to balance the mini-batches). We made 30 complete passes over all videos in the training set.

Supervised Learning. We trained a convolutional net with 128 9×9×9 filters (randomly initialized) on top of the features extracted by the convGRBM. Each feature map of the convolutional net received input from 4 randomly chosen pooling maps from the first layer. Architectural choices were motivated by a desire to extract mid-level spatio-temporal features; the local connectivity used is standard practice [1]. The nonlinearities we used were identical to those in [7] with the exception of extending contrast normalization and downsampling to 3D: LCN was performed using a 9×9×9 smoothing filter, followed by 4×4×4 average downsampling. We also tried a more traditional network architecture which did not use absolute value rectification and LCN. We found that it slightly decreased accuracy (by about 1%; less drastic than reported in [7] for static object recognition). The pooling layer was then subjected to a further max-pooling over time, the output was vectorized and connected to one or two fully-connected layers. All layers (except the convGRBM) used online backpropagation[1]. We made 30 complete passes through the training set.

Table 1 compares our approach to the prior art using dense sampling (i.e. no interest-point detection) and K-means quantization. We report mean accuracy over all six actions. Our method, to the best of our knowledge, gives the best mean accuracy on KTH amongst methods that do not use interest-point detection. The currently best performing method [17] uses the STIP interest-point detector and HOG/HOF or HOF descriptors (91.8 and 92.1%, respectively). Due to the high ratio of background pixels to subject pixels in KTH, and the limited number of actions (that don't require context information), interest-point methods tend to perform extremely well on KTH. Evidence already indicates that dense-sampling outperforms interest-points on more challenging datasets [27].

[1] The choice of using online learning here was simply a matter of convenience due to variable sequence lengths. Since the convGRBM is trained on pairs of frames (rather than whole sequences) it is easier to train in mini-batches.

Table 1. KTH action dataset: classification performance using dense sampling. Integers preceding a module indicate the number of feature maps in that module. Superscripts indicate filter sizes or downsampling ratio (chosen by context). convGRBM is our proposed method, trained unsupervised. F_{CSG} is a standard convolutional layer: a set of convolution filters (C) followed by a sigmoid/tanh nonlinearity (S), and gain coefficients (G). R/N/P_A is abs rectification, followed by local contrast normalization, followed by average pooling. The number of fully-connected layers are either 1 which corresponds to logistic regression (log_reg) or 2, which corresponds to a multi-layer perceptron (mlp).

Prior Art	Accuracy	Convolutional architectures	Accuracy
HOG3D-KM-SVM	85.3	32convGRBM$^{16\times16}$-128F$^{9\times9\times9}_{\mathrm{CSG}}$-R/N/P$^{4\times4\times4}_{A}$-log_reg	88.9
HOG/HOF-KM-SVM	86.1	32convGRBM$^{16\times16}$-128F$^{9\times9\times9}_{\mathrm{CSG}}$-R/N/P$^{4\times4\times4}_{A}$-mlp	**90.0**
HOG-KM-SVM	79.0	32F$^{16\times16\times2}_{\mathrm{CSG}}$-R/N/P$^{4\times4\times4}_{A}$-128F$^{9\times9\times9}_{\mathrm{CSG}}$-R/N/P$^{4\times4\times4}_{A}$-log_reg	79.4
HOF-KM-SVM	88.0	32F$^{16\times16\times2}_{\mathrm{CSG}}$-R/N/P$^{4\times4\times4}_{A}$-128F$^{9\times9\times9}_{\mathrm{CSG}}$-R/N/P$^{4\times4\times4}_{A}$-mlp	79.5

To demonstrate the advantage of low-level feature extraction with convGRBMs, we have replaced the first layer with a standard 3D convolutional layer (32F$^{16\times16\times2}_{\mathrm{CSG}}$ - see Table 1). By using filters of size $16\times16\times2$ and a $4\times4\times4$ pooling layer, we have matched the architecture of the convGRBM as best as possible to perform this comparison. The entire network is trained by backpropagation. We note that this fully feed-forward approach performs considerably worse.

5.2 Hollywood2 Dataset

The Hollywood2 dataset [25] consists of a collection of video clips containing 12 classes of human action extracted from 69 movies. It totals approximately 20.1 hours of video and contains approximately 150 samples per action. It provides a more realistic and challenging environment for human action recognition by containing varying spatial resolution, camera zoom, scene cuts and compression artifacts.

Table 2. Hollywood2 dataset: average precision (AP) using dense sampling

Method	AP
Prior Art [27]:	
HOG3D+KM+SVM	45.3
HOG/HOF+KM+SVM	**47.4**
HOG+KM+SVM	39.4
HOF+KM+SVM	45.5
convGRBM+SC+SVM	**46.6**

Performance is evaluated as suggested by Marszalek et al. [25]: by computing the average precision (AP) for each of the action classes and reporting mean AP over all actions. Following [27], we downsampled the spatial resolution of every video clip (which varies between clips) by a factor of 2. Videos were then zero-padded to have a constant spatial resolution. We did no temporal downsampling. All videos then underwent 3D local contrast normalization.

Similar to the KTH dataset, we trained a convGRBM with $N_z = 32$ feature maps and a pooling factor of $C = 4$. Filter sizes were $N_x^x = N_x^y = 16$. The convGRBM was trained for 50 complete passes over all videos in the training dataset and used a sparsity regularization term in the CD updates [28] that encouraged the hidden units to have a mean activation of 0.1.

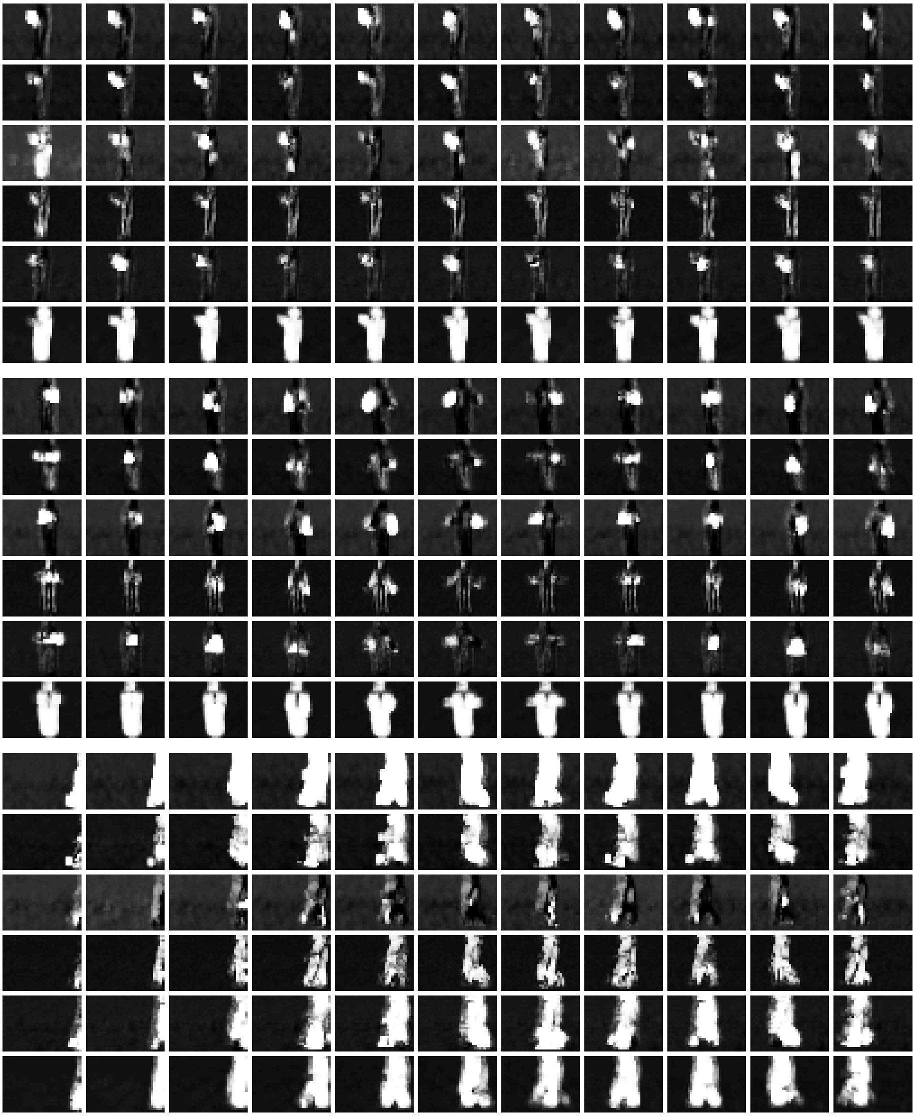


Fig. 4. Feature maps inferred from the KTH actions dataset. A subset of 6 (4 × 4 max-pooled) feature maps (of 32 total) inferred from sequences of the same subject performing different activities: boxing (rows 1-6), hand-clapping (rows 7-12) and walking (rows 13-18). Rows correspond to features, columns correspond to frames. We show person 1, scenario 1 and sequence 1. We display real-valued probabilities of activation rather than stochastic choices. We also downsample the frame rate by a factor of 4 for display. From the hand-clapping example, we see that features 1 and 3 are sensitive to motion in opposite directions (note how features 1 and 3 localize opposite hands), feature 4 seems to be sensitive to edges, and feature 6 learns to segment the subject from the background. Remaining activities are shown in the supplemental material.

Instead of applying a convolutional network to extract mid-level features, we sampled the feature maps of the convGRBM with a stride of 8 pixels in each direction, and formed local temporal groups of 10 frames. We then used the method described in [29] to learn a dictionary of 4000 basis vectors, and encode the temporal groups as sparse linear coefficients of the bases. Each video then yielded a varying number of sparse vectors (given different lengths) so we applied max-pooling over the temporal dimension. A SVM (with RBF kernel) was then trained (per-activity) on the top-level representation. Since Hollywood2 videos may contain more than one activity, this approach allowed us to avoid training a separate 3D convolutional net per-activity.

We achieve a mean AP of 46.6% using dense sampling, learned convGRBM low-level features and sparse coding with 4000 elements. To the best of our knowledge, the only superior published result is 47.4% which uses dense sampling with HOG/HOF features and quantization [27]. However, our result outperforms other popular methods such as Cuboids (45.0%) and Willems et al. (38.2%) (published in [27]). We also expect that an approach that combined our learned features with HOG/HOF descriptors could perform well.

6 Conclusion

Gated RBMs extract latent representations that are useful for video understanding tasks. However, they do not scale well to realistic resolutions and must learn separate feature detectors at all locations in a frame. In the spirit of recent work exploring convolutional deep architectures, we have introduced the convolutional gated RBM. We showed that it learned to represent optical flow and performed image analogies in a controlled, synthetic environment. In a more challenging setting, human activity recognition, it extracted useful motion-sensitive features, as well as segmentation and edge-detection operators that allowed it to perform competitively against the state-of-the-art as part of a multi-stage architecture.

Acknowledgments. We thank the Office Of Naval Research (ONR N000140910789, ONR N000140910076), and Google for supporting this research.

References

1. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE 86, 2278–2324 (1998)
2. Hinton, G., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Comput. 18, 1527–1554 (2006)
3. Larochelle, H., Erhan, D., Courville, A., Bergstra, J., Bengio, Y.: An empirical evaluation of deep architectures on problems with many factors of variation. In: ICML, pp. 473–480 (2007)
4. Ranzato, M., Poultney, C., Chopra, S., LeCun, Y.: Efficient learning of sparse representations with an energy-based model. In: NIPS, pp. 1137–1144 (2006)
5. Nair, V., Hinton, G.: 3D object recognition with deep belief nets. In: NIPS, pp. 1339–1347 (2009)

6. Cadieu, C., Olshausen, B.: Learning transformational invariants from natural movies. In: NIPS, pp. 209–216 (2009)
7. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: ICCV, pp. 2146–2153 (2009)
8. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: ICML, pp. 609–616 (2009)
9. Norouzi, M., Ranjbar, M., Mori, G.: Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. In: CVPR (2009)
10. Memisevic, R., Hinton, G.: Unsupervised learning of image transformations. In: CVPR (2007)
11. Memisevic, R., Hinton, G.: Learning to represent spatial transformations with factored higher-order Boltzmann machines. Neural Comput. (2010)
12. Sutskever, I., Hinton, G.: Learning multilevel distributed representations for high-dimensional sequences. In: AISTATS (2007)
13. Olshausen, B., Field, D.: Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Res. 37, 3311–3325 (1997)
14. Dean, T., Corrado, G., Washington, R.: Recursive sparse spatiotemporal coding. In: Proc. IEEE Int. Workshop on Mult. Inf. Proc. and Retr. (2009)
15. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: NIPS, pp. 801–808 (2007)
16. Laptev, I., Lindeberg, T.: Space-time interest points. In: ICCV, pp. 432–439 (2003)
17. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
18. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS (2005)
19. Willems, G., Tuytelaars, T., Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 650–663. Springer, Heidelberg (2008)
20. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. In: ICCV (2007)
21. He, X., Zemel, R., Carreira-Perpiñán, M.: Multiscale conditional random fields for image labeling. In: CVPR, pp. 695–702 (2004)
22. Hinton, G.: Training products of experts by minimizing contrastive divergence. Neural Comput. 14, 1771–1800 (2002)
23. LeCun, Y., Huang, F., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. In: CVPR (2004)
24. Pinto, N., Cox, D., DiCarlo, J.: Why is real-world visual object recognition hard? PLoS Comput. Biol. 4 (2008)
25. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR, pp. 2929–2936 (2009)
26. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: ICPR, pp. 32–36 (2004)
27. Wang, H., Ullah, M., Kläser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC, pp. 127–138 (2009)
28. Lee, H., Ekanadham, C., Ng, A.: Sparse deep belief net model for visual area V2. In: NIPS, pp. 873–880 (2008)
29. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: ICML, pp. 689–696 (2009)
30. Freund, Y., Haussler, D.: Unsupervised learning of distributions of binary vectors using 2-layer networks. In: Proc. NIPS, vol. 4 (1992)

Learning Pre-attentive Driving Behaviour from Holistic Visual Features

Nicolas Pugeault* and Richard Bowden

Centre for Vision, Speech and Signal Processing,
University of Surrey, UK
{n.pugeault,r.bowden}@surrey.ac.uk
http://www.ee.surrey.ac.uk/CVSSP/

Abstract. The aim of this paper is to learn driving behaviour by associating the actions recorded from a human driver with pre-attentive visual input, implemented using holistic image features (GIST). All images are labelled according to a number of driving–relevant contextual classes (eg, road type, junction) and the driver's actions (eg, braking, accelerating, steering) are recorded. The association between visual context and the driving data is learnt by Boosting decision stumps, that serve as input dimension selectors. Moreover, we propose a novel formulation of GIST features that lead to an improved performance for action prediction. The areas of the visual scenes that contribute to activation or inhibition of the predictors is shown by drawing activation maps for all learnt actions. We show good performance not only for detecting driving–relevant contextual labels, but also for predicting the driver's actions. The classifier's false positives and the associated activation maps can be used to focus attention and further learning on the uncommon and difficult situations.

1 Introduction

The objective of this manuscript is to learn the relationship between behaviour and visual stimulus in the context of driving. This is an extremely complex task due to variability in both the visual domain as well as the actions performed by the driver. Such actions are arguably dependant upon high level reasoning and context. However, we demonstrate that pre-attentive vision based upon simple holistic descriptors can account for the majority ($\sim 80\%$) of a driver's actions using minimal training ($< 1\%$).

The act of driving require little active attention for an experienced driver, allowing extended driving periods of several hours while at the same time having a conversation, thinking about an itinerary, etc. Indeed, this fact is a source of hazard, as an inattentive driver is less likely to react to unexpected emergencies. This article studies how pre-attentive visual perception can be used to learn aspects of driving behaviour by observing a human driver, releasing attention for other tasks such as tracking, traffics sign recognition, planning, etc. The

* Corresponding author.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 154–167, 2010.

learning is performed by recording the driver's actions (eg, braking, steering) at each frame together with a coarse labelling of each frame according to a set of driving contextual categories (eg, motorway, junction, pedestrian crossing). We choose to use holistic image features (so-called GIST) as a functional equivalent to pre-attentive vision in humans. GIST are a class of visual descriptors that encode a global representation of a visual scene's content, as opposed to local image features. This holistic aspect, together with the low resolution it requires, is consistent with the visual signal processed by the periphery of the retina in the absence of (relevant) gaze fixation. This is in stark contrast with feature–based methods that rely on high resolution extraction of sparse descriptors, and therefore belong to attentive vision.

Holistic representations of visual scenes have received a lot of attention during the last decade [1,2,3,4]. The rationale behind the use of holistic image descriptors for visual context description is that they are insensitive to the small variations that abound in complex scenes and hamper classification based on local features. This is especially critical in urban scenes, where the amount of visual information and variability is enormous. The original version of the GIST was proposed by Oliva & Torralba, who compared two descriptors based on the Fourier transform of image intensity [1]. The first one was based on the Fourier transform computed on the whole image (DST); the second is based on a windowed Fourier transform (WDST), localised on a coarse 8 × 8 grid. The latter was shown to contain more information than the first, and was used to define a set of perceptual properties (roughness, ruggedness, etc.) that allow for scene classification. In later publications by the same authors, the Fourier transform was replaced with steerable [2,5], or Gabor wavelets [3], computed over varying scale and orientation and averaged over grids of varying sizes. The dimension of the feature vector was in some case reduced using PCA [6,3]. Renninger & Malik studied how human subjects could identify visual scenes even after very brief exposures ($<$ 70ms), and proposed a GIST–like model as an explanation of those results [6]. Douze et al. compared GIST descriptors with bag-of-words approaches for image search, using the INRIA 'Holidays' and 'Copydays' datasets, and found that GIST descriptors yield lower performances than state of the art bag-of-word approaches, yet with a considerably lower computational and memory cost [4]. Siagan & Itti, used similar descriptors for the identification of indoor and outdoor scenes in a mobile robotics context [3,7]. Their implementation differs insofar as they use different filter banks, including centre-surround colour sensitive filters, and the resulting feature vectors were post-processed using PCA and ICA. Ackerman & Itti used spectral image information for steering a robotic platform on a path following scenario on two simple tracks [8]; in contrast, we consider a large database of real urban scenes. Kastner et al. [9] use a GIST variant for road type context detection, limited to the three categories 'highway', 'country road' and 'inner city'; their main contribution was the hierarchical principal component classification (HPCC).

In contrast, in this article we attempt to detect 13 contextual labels of varying difficulty pertaining to scene environment, road type, junction type along with

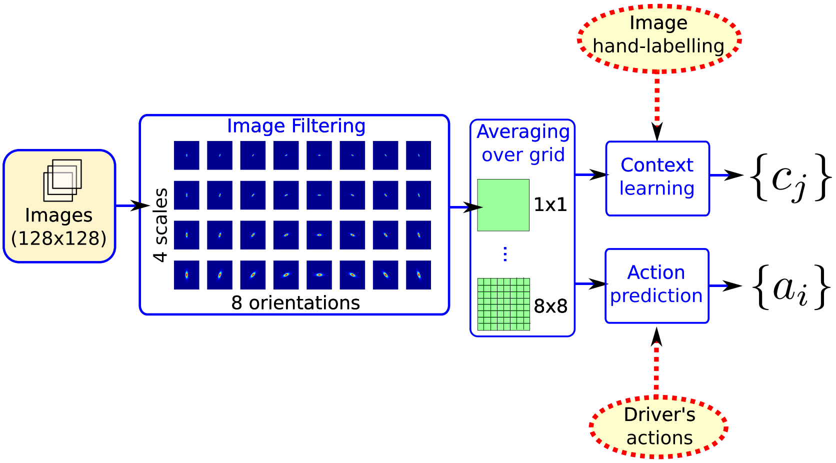


Fig. 1. Overview of the pre-attentive driving behaviour learning framework

some other attributes. Moreover, we learn relations between the visual context and five of the driver's actions: the activation of each of the three pedals, plus steering. We then show how these classifiers can be reversed to provide activation maps that determine the salient visual information that influences each action. The framework we propose is illustrated in Fig. 1: images are first resized and the contrast is normalized, then they are convolved with a filter bank, and the response is averaged over a grid; this forms the GIST descriptor. Then, two experts are learnt from these descriptors: the first one learns to detect contextual categories using hand labelled training samples; the second learns to predict the driver's actions. In this graph, the red dotted arrows represent information that is only provided at the training stage.

2 Methods

In this section we describe the learning framework illustrated in Fig 1: first, in section 2.1 we describe the GIST descriptor used, and propose a novel formulation of the descriptor; second, in section 2.2 we briefly discuss the learning algorithm.

2.1 Holistic Image Descriptors (GIST)

GIST are holistic image descriptor that encode a whole visual scene in one feature vector, generated by a coarse scale local filtering of a low resolution version of the image. The exact implementation varies in the literature, and the exact type of filters used does not seem to bear a major effect on the performance for context detection. In this work, we start by downscaling the images to 128×128 and normalizing the contrast, before filtering the resulting image with a bank of

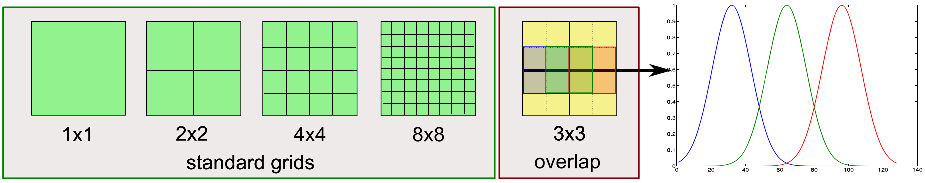


Fig. 2. Illustration of the grid averaging process. The left hand side shows the standard GIST grids, for sizes ranging from 1×1 to 8×8. The middle shows the effective cells for 2×2 grid with overlap: the green, red and blue square represent three overlapping squares. On the right, the graph shows an horizontal slice of this last grid, with overlapping Gaussians.

Gabor filters tuned to 8 different orientations and 4 scales; this results in $p = 32$ jets. The data size is then reduced by averaging the jets over a coarse grid laid over the image. Here again, the size of the grid used vary in the literature (we investigate the effect of this parameter in section 3.3); Oliva and Torralba reported a better performance of 4×4 versus 1×1 grids for context detection [1]. In this article we consider grids of size 1×1, 2×2, 4×4 and 8×8, separately and in combination (see Fig. 2).

One issue with this classical implementation is that the GIST vector can be very sensitive to small shifts of the features that lie close to the grid's boundaries. We propose an alternative sampling procedure based on overlapping smoothed cells. In this approach, adjacent rows of cells are overlapping by 50%, leading to an effective number of 144 cells for a 8×8 grid (see Fig. 2). Each cell's data vector $\mathbf{H} = (h_1, \cdots, h_p)$ is computed by averaging each jet $F_k, k \in \{1, \cdots, p\}$ according to a Gaussian kernel of variance one quarter of the grid cell's width:

$$h_k(x_0, y_0, s) = Q \sum_{x,y} F_k(x, y) \exp\left[-\left(\frac{x - x_0}{s/4}\right)^2 + \left(\frac{y - y_0}{s/4}\right)^2\right], \qquad (1)$$

where (x_0, y_0) is the centre of the grid cell, s is the cell width in pixels and Q is a normalization constant. The overlapping grid cells and the Gaussian smoothing are used to reduce the GIST vector sensitivity to small displacements at the grid's boundaries, and is shown to significantly improve performance on action prediction.

We will dispense with the additional PCA and/or ICA post-processing that is commonplace in the GIST literature (eg, [3]). Although reducing the feature dimension can be useful for some processes, we will rely on the boosted classifier to reduce dimensionality selectively through feature selection for each target category.

2.2 Classification

We use Boosting for learning both contextual labels and actions, as it has been shown to be successful for input selection and recognition [10,11]. We use a variant called *GentleBoost*, that has been shown to be more robust to noisy datasets [12]. Boosting is based on combining the weighted responses of a population of simple classifiers (called 'weak learners') into one robust classifier. The weak learners $l_i = (d_i, \tau_i, s_i)$ we used are simple decision stumps, each one applying a threshold τ on one of the feature vector's dimension d

$$R(l, \mathbf{v}) = \begin{cases} +s & \text{if} \quad v_d > \tau \\ -s & \text{otherwise} \end{cases}, \tag{2}$$

where, $s = \{-1, +1\}$ encodes the sign of the threshold that is applied. For each round of boosting i, the input dimension that best separates positive and negative examples is chosen, and the weights are updated. The classifier is therefore described by $\mathbf{L} = \{(l_1, w_1), \ldots (l_i, w_i), \ldots, (l_N, w_N)\}$, and the response is given by:

$$R(\mathbf{L}, \mathbf{v}) = \sum_{i=1}^{N} w_i \cdot R(l_i, \mathbf{v}). \tag{3}$$

As the number of weak learners is lower than the number of input dimensions, the learning process is effectively performing feature selection from the high dimensional input, and the weight of each weak learner provides a cue of the relative importance of each input towards the decision. In the following, and unless stated otherwise, the classifier was always trained using 1,000 samples from the dataset (0.7%), with half of the training set containing positive examples, and half negative examples. This positive/negative ratio was enforced to ensure that a sufficient number of positive examples were shown to the classifier, even for infrequent categories. Unless otherwise stated, the classifiers are evaluated on the rest of the dataset (ie, $> 99\%$ of the data).

2.3 Activation

In order to focus attention and direct higher level processes to relevant areas of the image, we need to evaluate which parts of the visual scene the predictors are tuned to, and whether they contribute to the activation or the inhibition of the action. We experimented with different ways to formalise what the predictors are responding to, and settled on reprojecting the Gaussian smoothing kernel in section 2.1 for each weak learner, weighted by this learner's weight. Thus the activation map is given by the mixture of Gaussians:

$$A(\mathbf{v}) = \sum_{i}^{|\mathbf{L}|} \left(w_i \cdot R(l_i, \mathbf{v}) \cdot G(l_i) \right), \tag{4}$$

for all weak learners l_i. In this equation $G(l)$ is the Gaussian kernel centred at the GIST grid cell l_i is associated with, with a variance of one fourth of the cell's width. The resulting map provides, for all images, an illustration of which image areas activate or inhibit each action.

Figure 9 shows the activation maps for each action for several example scenes, where the image is overlaid by green for excitation and red for inhibition.

3 Results

We evaluated the learning on a sequence taken from an instrumented car. The sequence contains 158,668 images for a total of about 3 hours of data, encompassing a variety of driving situations and settings. The dataset is illustrated in Fig. 3. The driver's actions were recorded from the car for each frame in the sequence.

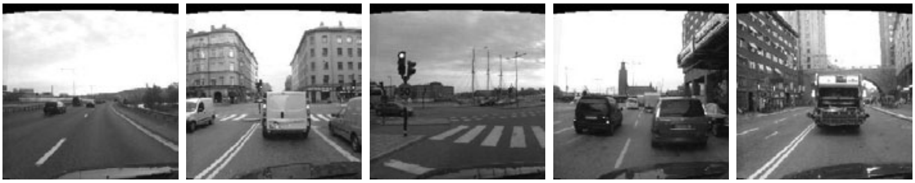


Fig. 3. Some example images taken from the 158,668 in the sequence

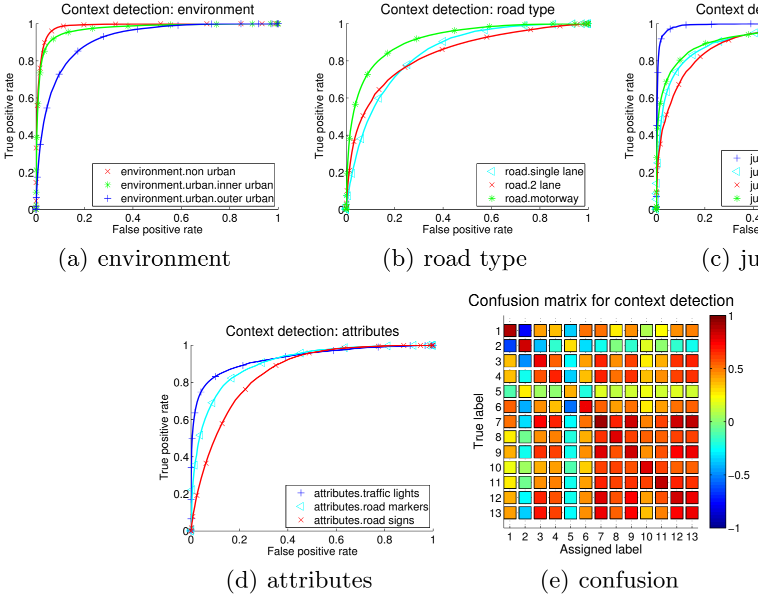
3.1 Learning Context Classes

Context information was provided in the form of a coarse labelling of each frame in the sequence pertaining to 13 classes. The number of frames labelled for each class is recorded in Table 1. The context classes are separated in four categories: *environment*, *road*, *junction* and *attributes*.

Table 1. Context labels associated to all images in the sequence (total: 158,668 frames)

Index	Category	Label	Count
1	environment	non-urban	47,923
2	environment	inner-urban	82,424
3	environment	outer-urban	28,321
4	road	single lane	31,269
5	road	two lanes	86,879
6	road	motorway	38,880
7	junction	roundabout	2,007
8	junction	crossroads	17,366
9	junction	T-junction	7,895
10	junction	pedestrian crossings	29,865
11	attributes	traffic lights	21,799
12	attributes	road markers	6,462
13	attributes	road signs	3,387

We trained an ensemble of Boosted decision stumps for each context class, using 100 rounds of Boosting on 1,000 frames chosen randomly; the performance was then evaluated on the rest of the dataset (more than 150,000 frames). Fig. 4 shows receiver operating characteristic (ROC) curves for all context classes, grouped by category. The confusion matrix is drawn in Fig. 4(e).



(a) environment (b) road type (c) junction

(d) attributes (e) confusion

Fig. 4. (a–d) ROC curves for the detection of different type of contextual information (see Table 1); (e) confusion matrix. All plots are for a combination of all overlapping grids for 100 rounds of Boosting, averaged over 10 runs.

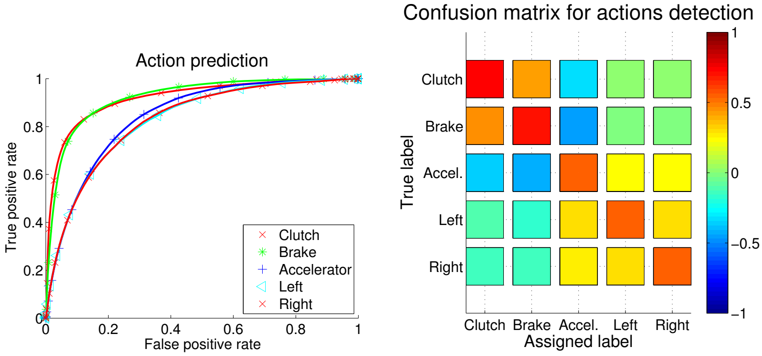
All classes are detected with good performance (note that all detectors are processed independently, without enforcing mutual exclusivity). The detection of the *environment* classes performs especially well, and the best performance is reached for distinction between ‘inner urban’ and ‘non urban’. The lower detection performance for ‘outer urban’ is likely to be due to the somewhat fuzzier definition of the class; this is confirmed by the higher confusion value between ‘non urban’ and ‘outer urban’. This high performance is consistent with published results in the literature. These categories are obviously global context categories and high performance validates other researchers’ findings that GIST–type descriptors perform well for context recognition.

However, the performance is surprisingly high for other (more difficult) categories which make less use of global context. For the *road* category, confusion values are high between the ‘single lane’ and ‘inner urban’ classes, and the ‘motorway’ and ‘non urban’ classes, which are naturally consistent with expectations. The detectors for *junction* and *attributes* show a good performance for all classes (the very high performance on the ‘roundabout’ class may be due to the relatively low number of examples in the database). The confusion matrix shows a large confusion between all *junction* and *attributes* classes, and the ‘inner city’ class. This is consistent with the reality of traffic settings, and it should be noted that traffic lights (for example) are fundamentally *local* visual events, and therefore what is detected in this case is the visual context in which they are *likely* to occur, which is indeed a town centre intersection.

3.2 Learning Driving Actions

In a second experiment, we learnt to predict driver's actions from the gist features. The actions we considered are the pressing of one of the three pedals (Accelerator, Brake and Clutch) and the action of steering left or right. The actions were discretised, and therefore the amplitude of each action was disregarded for this experiment. Note that observation of the data revealed that the actions of pressing the clutch or the brake were binary actions anyway.

The classifier used was GentleBoost with decision stumps as weak learners; it was trained for 100 rounds with 1,000 randomly selected data points (less than 1% of the dataset).



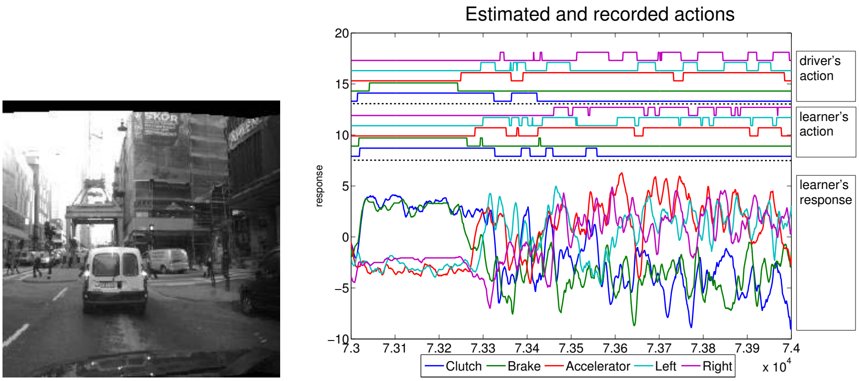
(a) ROC curves

(b) Confusion matrix

Fig. 5. Performance of the action prediction: (a) ROC analysis, and (b) confusion matrix. The results are for 100 rounds of boosting on the combined grids GIST descriptors; the training is done with 1,000 random frames, and tested with the rest of the dataset.

The action prediction performance is recorded in Fig. 5: the 'clutch' and 'brake' actions are predicted well (with 80% true positives for 10% false negatives); the two predictions also share a strong confusion value. This effect is driven by the large number of cases where the driver brings the car to a stop, pressing concurrently both brake and clutch. The performance when predicting the accelerator pedal and steering left or right is lower (80% false positives for 30% false negatives) but still good considering the large variability in the data. There is positive confusion values between steering and acceleration, which is consistent with good driving technique. The positive confusion between left and right steering is likely to come from the intersection situations, where steering left or right is equally plausible from visual information only.

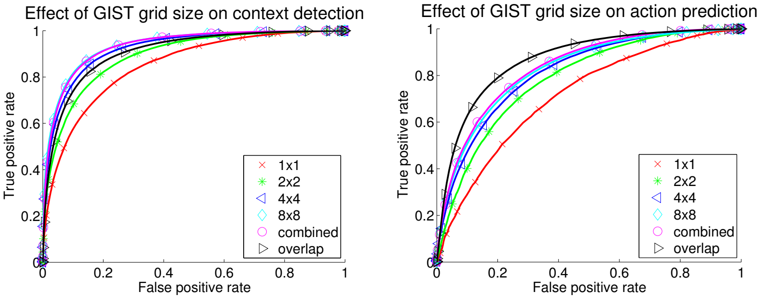
Fig 6 illustrates the quality of the action prediction on a short subsequence: the graph show curves for each action, for the driver and for the learnt response potential and final decision, respectively from top to bottom. The classifier was trained for 100 rounds on 1,000 frames taken randomly out of the 158,668. The classifier's response was smoothed using a 5–points moving average to remove the



(a) one image

(b) actions

Fig. 6. Illustration of the driver's and the system's elicited actions, on a short sub-sequence (1,000 frames). (a) first image in the sequence; (b) from top to bottom: the driver's action, the system's elicited actions, and the system's raw response. The predictor's response was smoothed using a 5–points moving average.



(a) GIST grid — context

(b) GIST grid — actions

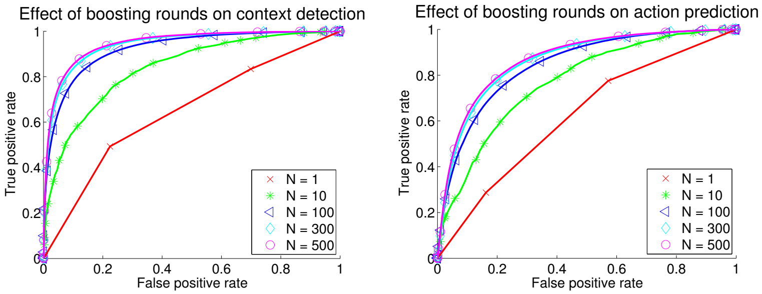
Fig. 7. Analysis of the effect of the GIST grid size on performance for (a) context detection and (b) action prediction

isolated outliers. The prediction for 'Brake', 'Clutch' and 'Accelerator' (accelerator pedal) is of very good quality for the whole sub-sequence. The prediction for steering 'Left' or 'Right' is not as reliable, but follows nonetheless the same patterns as the driver's.

3.3 Evaluation of the System's Parameters

We evaluated the influence of the GIST grid size and of the number of Boosting rounds on the detectors' performance, the results are displayed as ROC curves in Fig. 7 and 8. These ROC curves show the average performance over all classes and over 10 successive trainings of the detectors, each time with 1,000 randomly selected samples, and evaluated on the rest of the dataset. Figs. 7 show the performance for different GIST grids: 1 × 1, 2 × 2, 4 × 4, 8 × 8, and combinations of them all with and without overlapping. Each curve was obtained for 100 rounds of boosting. The best performance was obtained for using 8 × 8 grid and no additional performance was gained when using jointly a combination of all grids. The performance remained very good when using a 4 × 4 grid but dropped when using coarser histograms. When using overlapping smoothed grids, the performance for the context detection task was not improved compared to the 8 × 8 grid (Fig 7(a)); on the other hand, the performance for action prediction was significantly improved (Fig. 7(b)). This is likely to be due to less reliance upon global context and localised higher variability in the aspects of visual scenes relevant for predicting actions; eg, the position and the shape of the vehicle being followed can change to large extent. The non-overlapping grid used in classical GIST implementations make the feature vector sensitive to changes at the grid's boundaries, whereas an overlapping grid is less affected.

Fig. 8 shows the performance obtained for varying the number of rounds of Boosting, using an overlapping smoothed grid. No significant improvement was obtained by rising from 300 to 500 rounds, and 100 rounds yielded good performance. The performance for a single round of boosting was given as a baseline for a single decision stump's performance. Similar results were obtained when using other grids.

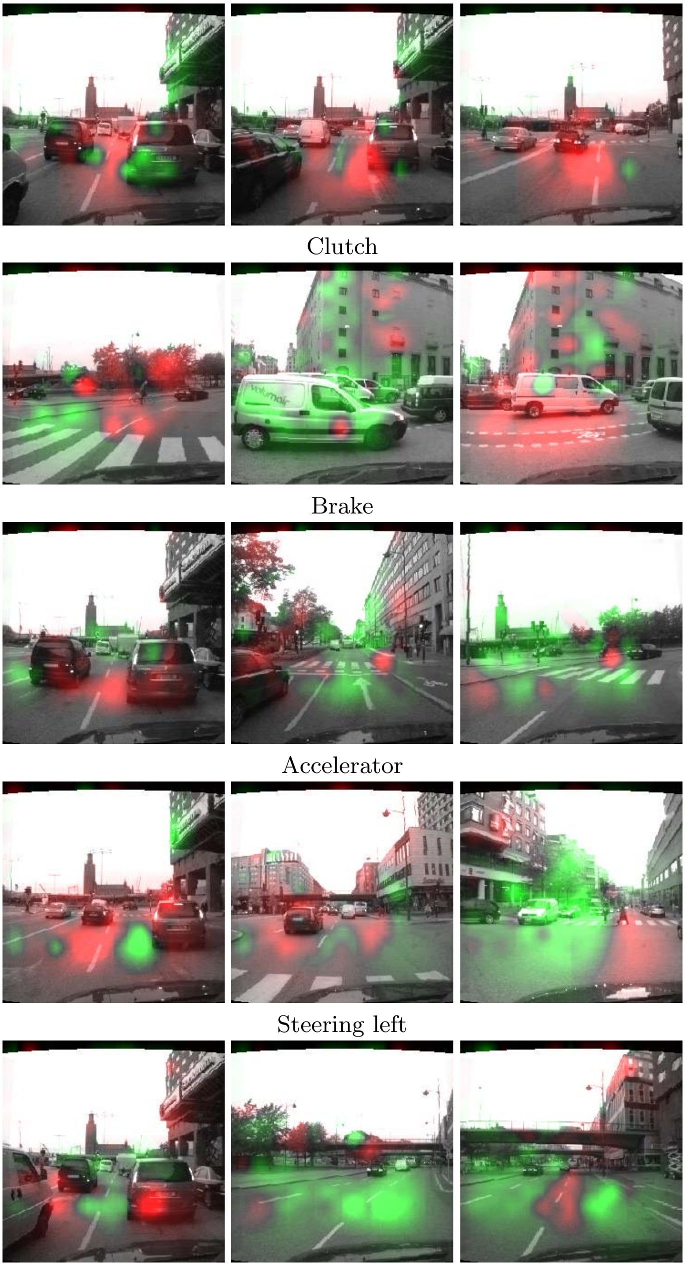


(a) Boosting rounds — context (b) Boosting rounds — actions

Fig. 8. Analysis of the effect of the number of rounds of Boosting on performance for (a) context detection and (b) action prediction

3.4 Predictors' Activation

In order to get a better insight in what rules the system learns from the driver, we use the classifier inversion described in section 2.3 to identify what parts of the visual scenes activate the different action predictors. In Fig 9, the activation maps for three different situations are shown, for all actions. On those maps, the original image is overlaid with green on the excitatory areas and red on the inhibitory areas.



Clutch

Brake

Accelerator

Steering left

Steering right

Fig. 9. Activation maps on selected frames for each action predicted by the system; green shows activation and red inhibition. Areas of empty road activate acceleration and steering towards them, while inhibiting braking and pressing the clutch. Conversely, other vehicles on the road inhibit acceleration and steering towards them, while exciting braking, pressing the clutch and steering away from them—see text.

We expect the clutch pedal to be depressed when reducing speed to a minimum or when the car stops. On the left image, we can see that the clutch is activated by the presence of another car immediately in front. On the middle image the car is further away, and the same image area, now empty of cars, inhibits the 'clutch' predictor; a similar inhibition pattern is visible in the right image where the road ahead is free. The second row shows consistent activation patterns for the 'brake' action: on the left, the empty road in front inhibits the predictor whereas the pedestrian crossing area activates it. On the middle, the presence of a car immediately in front leads to a strong excitation, whereas on the right an empty space yields a strong inhibition. As expected, the 'accelerator' activation is the opposite of the 'brake': activated by empty spaces and inhibited by other vehicles in front. The activation maps for steering actions are somewhat more difficult to interpret, as expected from the lower prediction performance. The 'left' and 'right' actions appear to be activated by obstacles and to promote veering away from them (see left images). They also seem to react to the vehicle's position in its lane, as evidenced by the sharp inhibition of steering generated on the central white line (see the bottom–right image).

4 Discussion

In this article we attempt to model driving behaviour by learning the relationship between a human driver's actions and holistic image descriptors. Supervision comes in two forms: first, a coarse labelling of the images in terms of a variety of driving–relevant contextual categories; second, a frame per frame record of the driver's actions when faced with this situation. We use GIST features as an equivalent to human pre-attentive vision, for encoding the visual input, and attempt to learn, for all images, both the associated labels and the driver's actions.

The GIST descriptor is a generic approach for holistic image features, and has several free parameters. Experimenting with different type of grids for the GIST descriptor, we found that the best performance was obtained for a 8 × 8 grid. Moreover, the small difference in performance between 8 × 8 and 4 × 4 grids make in unlikely for finer grids to increase performance notably, for a high computational cost. Instead, we proposed an overlapping grid smoothed using Gaussian functions, that lead to a significant performance improvement for action prediction (see Fig. 7(b)).

We found that the optimal performance is reached with a relatively low number of rounds of Boosting for both context detection and action prediction (100 rounds); this is a large dimension reduction compared to the original feature vector (6,496 for the combined overlapping grids). Therefore, the relatively high dimensionality of the original feature vector is not an issue after the training stage as each classifier only uses a small carefully selected proportion of it. Those dimensions and their respective contribution to the classifier's response can be reprojected in the image domain as discussed in section 2.3, and produces the activation maps shown in Fig. 9.

The high performance of pre-attentive vision for detecting the environment class (‘non urban’, ‘outer urban’, or ‘inner urban’) is consistent with previous results in the literature. Very good performance was also obtained when detecting more complex aspects of the driving context such as T–junctions, pedestrian crossings, or even traffic lights (see Fig. 4).This shows that holistic features do carry a large amount of visual information relevant for interpreting driving scenarios. Moreover, the success in detecting what are essentially *local* events (eg, traffic lights) shows the high contextual prior that permeates most driving visual scenes: the presence of an intersection in an urban setting, for example, is a strong predictor for the presence of a traffic light, or road markings.

The performance with which the driver’s actions can be predicted from holistic image features, is a more unexpected result (see Figs. 5 and 6). Indeed, the system does not have insight into the driver’s intentions and lacks any formal knowledge of the highway code. The fact that the driver’s actions can be predicted at all, only from transient holistic image features, illustrates the intuition that most of a driver’s actions are completely determined by the context in which he is, and only a small fraction is determined by intention, attentive vision and high–level reasoning. These cases are of special importance for learning an attentional model of the driver’s behaviour: we expect the false positives to be the instances in the dataset where the driver’s pre-attentive actions were inhibited by higher–level considerations. If we consider the case of crossing traffic at an intersection, pre-attentive vision may learn to slow down before the intersection, but the driver will then need to actively assess whether the way is free or if he needs to stop. The activation maps shown in Fig. 9 provides us with a useful indication of which parts of the scene are relevant for taking a decision; together with the driver’s gaze, they provide a way to focus the attention of a higher level feature–based learning on the most promising parts of the visual scene. Therefore, the learning of a more complex model can be bootstrapped by the activation maps at false positives and the driver’s gaze can be combined to learn the attentive components of driving. In this context, the pre-attentive model serves as a filter to focus attentional learning towards the rare instances where it is required, and the aspects of the scenes that may be of importance.

5 Conclusion

Holistic image descriptors have received a lot of attention in the recent year, both from the computer vision and the psychology communities, as a good model for fast, pre-attentive vision, and a good feature for scene identification. We used such GIST features for learning driving behaviour from a human driver, and obtained very good results both for the detection of visual context labels and for the prediction of the driver’s actions. The fairly high performance of the action prediction illustrates the fact that only a small proportion of the driving actions require formal understanding of the driver’s intentions or the highway code. This is a vivid illustration of the strong priors at work during normal driving behaviour, and of how much information pre-attentive perception can carry, as 80% of a driver's actions can be predicted. Such a performance allows to focus attention on learning the more complex rules that underlie the 10–20% of problematic cases.

Acknowledgments. This research was supported by the EU project DIPLECS (FP 7 ICT project no. 215078).

References

1. 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)
2. Torralba, A.: Contextual priming for object detection. International Journal of Computer Vision 53, 169–191 (2003)
3. Siagian, C., Itti, L.: Rapid biologically-inspired scene classification using features shared with visual attention. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 300–312 (2007)
4. Douze, M., Jégou, H., Sandhwalia, H., Amsaleg, L., Schmid, C.: Evaluation of gist descriptors for web-scale image search. In: CIVR 2009: Proceedings of the ACM International Conference on Image and Video Retrieval (2009)
5. Torralba, A., Oliva, A., Castelhano, M., Henderson, J.: Contextual guidance of attention in natural scenes: The role of global features on object search. Psychological Review 113, 766–786 (2006)
6. Renninger, L., Malik, J.: When is scene identification just texture recognition? Vision Research 44, 2301–2311 (2004)
7. Siagian, C., Itti, L.: Biologically inspired mobile robot vision localization. IEEE Transactions on Robotics 25, 861–873 (2009)
8. Ackerman, C., Itti, L.: Robot steering with spectral image information. IEEE Transactions in Robotics 21, 247–251 (2005)
9. Kastner, R., Schneider, F., Michalke, T., Fritsch, J., Goerick, C.: Image–based classification of driving scenes by a hierarchical principal component classification (HPCC). In: IEEE Intelligent Vehicles Symposium, pp. 341–346 (2009)
10. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55, 119–139 (1997)
11. Viola, P., Jones, M.: Robust real–time object detection. International Journal of Computer Vision 57, 137–154 (2001)
12. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. The Annals of Statistics 28, 337–407 (2000)

Detecting People Using Mutually Consistent Poselet Activations[*]

Lubomir Bourdev[1,2], Subhransu Maji[1], Thomas Brox[1], and Jitendra Malik[1]

[1] University of California at Berkeley
[2] Adobe Systems, Inc., San Jose, CA
{lbourdev,smaji,brox,malik}@eecs.berkeley.edu

Abstract. Bourdev and Malik (ICCV 09) introduced a new notion of parts, poselets, constructed to be tightly clustered both in the configuration space of keypoints, as well as in the appearance space of image patches. In this paper we develop a new algorithm for detecting people using poselets. Unlike that work which used 3D annotations of keypoints, we use only 2D annotations which are much easier for naive human annotators. The main algorithmic contribution is in how we use the pattern of poselet activations. Individual poselet activations are noisy, but considering the spatial context of each can provide vital disambiguating information, just as object detection can be improved by considering the detection scores of nearby objects in the scene. This can be done by training a two-layer feed-forward network with weights set using a max margin technique. The refined poselet activations are then clustered into mutually consistent hypotheses where consistency is based on empirically determined spatial keypoint distributions. Finally, bounding boxes are predicted for each person hypothesis and shape masks are aligned to edges in the image to provide a segmentation. To the best of our knowledge, the resulting system is the current best performer on the task of people detection and segmentation with an average precision of 47.8% and 40.5% respectively on PASCAL VOC 2009.

1 Introduction

Detecting people in images is hard because of the variation in visual appearance caused by changes in clothing, pose, articulation and occlusion. It is widely accepted that a representation based on parts is necessary to tackle the challenge of detecting people in images. But how shall we define parts?

Historically, the most common choice has been to use basic anatomical structures such as torso, left upper arm, left lower arm, and in a probabilistic framework such as pictorial structures [1], these become nodes in a graphical model and the conditional independence assumption inherent in the tree structure make inference tractable. Other approaches that look for good scoring parts in the right spatial relationships may be found in [2,3,4,5,6].

[*] This work was supported by Adobe Systems, Inc., a Google Fellowship., the German Academic Exchange Service (DAAD), and ONR MURI N00014-06-1-0734.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 168–181, 2010.

While these parts are quite natural in constructing kinematic models of a moving person, they are not necessarily the most salient features for visual recognition. A limb, modeled as a pair of parallel line segments, is quite difficult to detect reliably; there are false positives all over an image. In contrast, a visual conjunction such as "half of a frontal face and a left shoulder" may be a perfectly good discriminative visual pattern. This is perhaps the reason why the best performing approaches on people detection tend not to be based on first detecting anatomical parts. Leading this trend was work on pedestrian detection [7,8] using a multi-scale sliding window paradigm; other examples of such "appearance-based" techniques include [9,10,11,4]. Currently the best performing system on the task of people detection is by Felzenszwalb et al. [12] who generalized the approach to allow an intermediate layer of "parts" that can now be shifted with respect to each other, rendering the overall model deformable. The templates for these parts emerge as part of the overall discriminative training. The latest version, dubbed *Latent SVM* by the authors, has an additional mixture model on top permitting a rudimentary treatment of aspect.

Bourdev and Malik [14] introduced a new notion of parts as *poselets*, where the key idea is to define parts that are tightly clustered both in configuration space (as might be parameterized by the locations of various joints), and in appearance space (as might be parameterized by pixel values in an image patch). Finding such parts requires extra annotation, and [14] introduced a new dataset, H3D, consisting of images of people annotated with 3D keypoints making use of Taylor's algorithm [15]. The poselets themselves are created by a search procedure. A patch is randomly chosen in the image of a randomly picked person (the *seed* of the poselet), and other examples are found by searching in images of other people for a patch where the configuration of keypoints is similar to that in the seed (see figures 1, 6, and 7 in [14]). Given a set of examples of a poselet, which are, by construction, tightly clustered in configuration space, HOG features [7] are computed for each of the associated image patches. These are positive examples for training a linear Support Vector Machine. At test time, a multi-scale sliding window paradigm is used to find strong activations of the different poselet filters. These are combined by voting using a Max Margin Hough Transform for the torso/bounding box of a person.

In this paper, we present a better way to define and use poselets. We start with a critique of the approach in [14]:

The use of 3D keypoint annotations: While these carry more information than 2D annotations, they come at a cost in terms of annotation expense. The H3D annotation environment requires some degree of skill, and about 1-2 minutes per image. If we only mark keypoints in 2D, the task becomes much simpler and portable to unskilled labor of the type available on Amazon Mechanical Turk. While individual annotations become less informative, the ability to collect many more for a given amount of time and money is a great advantage. Additionally this makes the poselet idea applicable to other object categories where lifting to 3D using the Taylor algorithm is not even possible.

The use of Hough Transform voting: While such techniques have been used in computer vision from early days, and are natural baselines before trying more complex approaches, they provide less flexibility than one might like. Essentially this is a star model [16], in the graphical model sense, with part positions being referred to a center (a torso in [14]). However there may be no common target that all parts predict reliably. Each poselet makes good predictions only about local structure – a feet poselet does not know if the person is sitting or standing, and a face poselet cannot know if the person is occluded by, say, a table. Instead, we should look at pairwise consistency of poselets. A left shoulder poselet and a frontal face poselet may be uncertain in their prediction of the visible bounds, but they are certain on where the shoulders are, which makes it easier to tell if they refer to the same person.

In the following sections we propose solutions to these limitations which significantly increase performance.

2 Overview of Our Approach

The first step is to train poselets using only 2D keypoint annotations. Ignoring 3D information becomes possible by a new distance function for comparing 2D keypoint configurations. This simplification of annotation allowed us to augment the training set of [14] by annotation of the people category of PASCAL VOC 2009 training and validation images. The larger amount of training data leads to better initial poselet detectors. The experiments in this paper are based on training 500 such detectors, and we select from these, in a greedy fashion, the 100 or 200 best performing poselets that maximize coverage of the different examples in the training set.

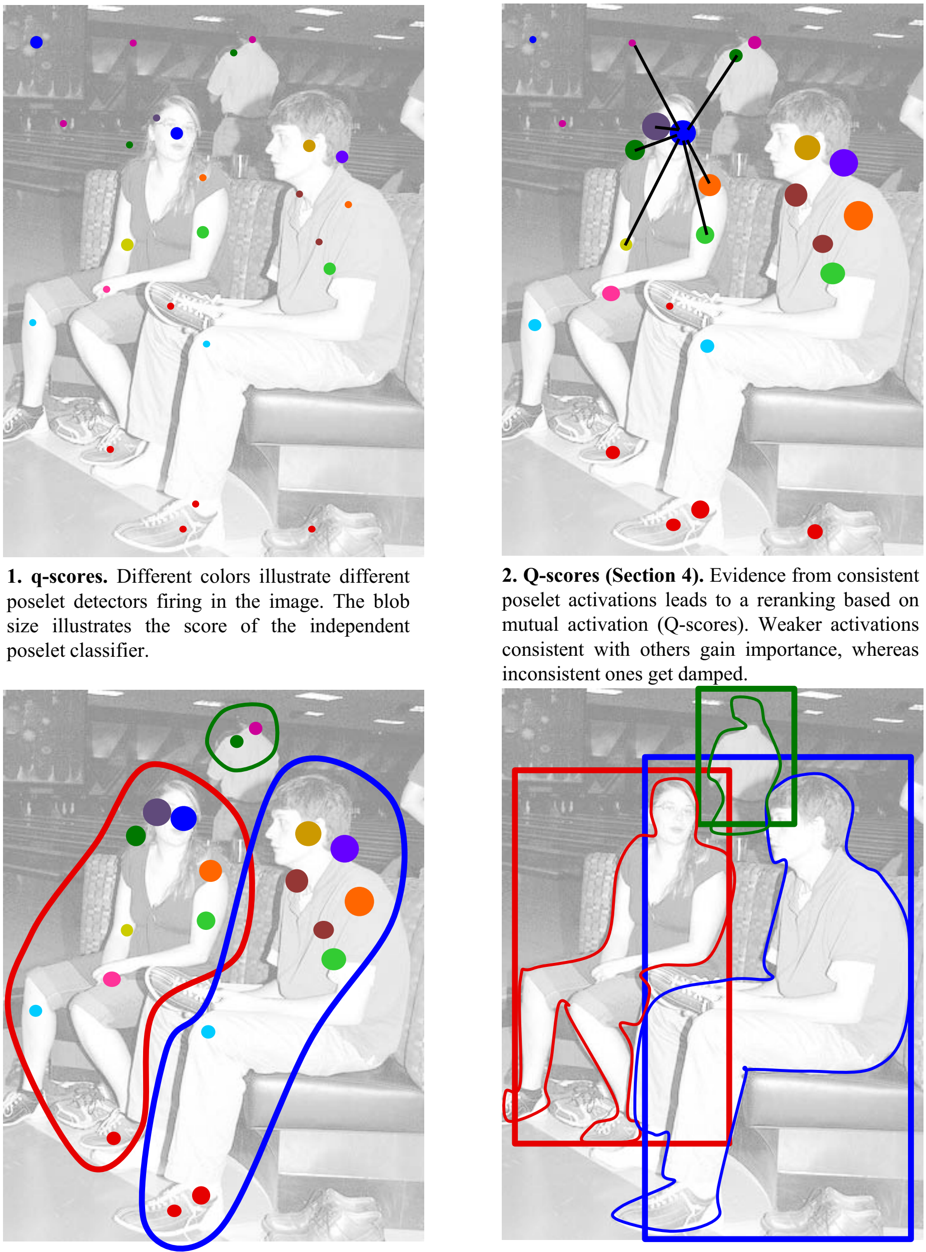
At test time, a multi-scale sliding window paradigm is used to find strong activations of the different poselet filters. In the overview figure for our algorithm, the results of this stage are shown as Fig. 1.1. We need to cluster these activations together if they correspond to the same hypothesized person in the image, predict a score for this person hypothesis, as well as an associated figure/ground segmentation and a bounding box.

The key insight here is that if two poselet activations are consistent, they will make similar predictions of the keypoints of the person, because two consistent true positive activations detect parts of the *same* person.

At training time, we can measure the empirical keypoint distributions (Fig. 2) associated with true activations of various poselet types, and at test time, we measure consistency between two poselet activations i and j using the symmetrized KL-divergence of their empirical keypoint distributions $\mathcal{N}_i^k$ and $\mathcal{N}_j^k$:

$$D_{SKL}(\mathcal{N}_i^k, \mathcal{N}_j^k) = D_{KL}(\mathcal{N}_i^k || \mathcal{N}_j^k) + D_{KL}(\mathcal{N}_j^k || \mathcal{N}_i^k) \tag{1}$$

$$d_{i,j} = \frac{1}{K} \sum_k D_{SKL}(\mathcal{N}_i^k, \mathcal{N}_j^k) \tag{2}$$



1. q-scores. Different colors illustrate different poselet detectors firing in the image. The blob size illustrates the score of the independent poselet classifier.

2. Q-scores (Section 4). Evidence from consistent poselet activations leads to a reranking based on mutual activation (Q-scores). Weaker activations consistent with others gain importance, whereas inconsistent ones get damped.

3. Clustering (Section 5). Activations are merged in a greedy manner starting with the strongest activation. Merging is based on pairwise consistency.

4. Bounding boxes (Section 6) and segmentations (Section 7). We predict the visible bounds and the contour of the person using the poselets within the cluster.

Fig. 1. Schematic overview with manually marked activations to illustrate the method we propose in this paper

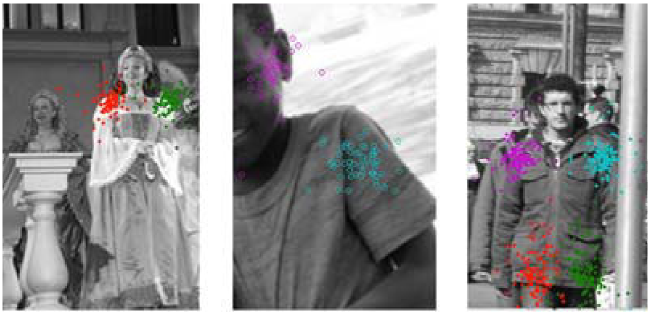


Fig. 2. Empirical keypoint distribution: locations of the shoulders (left), shoulder and ear (middle), and shoulders and hips (right) over true positive poselet activations

Since we represent these keypoint distributions as 2D Gaussians, D_{SLK} has a closed-form solution, and the summation is over all the K common keypoints in the two annotations.

The step from Fig. 1.1 to Fig. 1.2 illustrates an additional layer in the detector that uses the context of other poselet activations. This can be regarded as a feed-forward network, where the first layer generates poselet activations whose scores are independent (we call them **q-scores**) and the second layer combines all these to result in context-improved rescoring **Q-scores**. Alternatively, the **q** to **Q** stage can also be regarded as a star model applied to each poselet activation. The number of poselet activations stays the same, but the score of each activation is changed.

The activations are then clustered together to form people detections; cf. Fig. 1.3. We use a saliency based agglomerative clustering with pairwise distances based on consistency of the empirical keypoint distributions predicted by each poselet. Activations that have low score and are not consistent enough to be merged with one of the existing clusters get removed.

Fig. 1.4 illustrates the final step of predicting bounding boxes from the poselets in each cluster. Alternatively, we can predict segmentations from the clustered poselets.

3 Training and Selecting Poselets

We used the H3D training set (750 annotations), the PASCAL VOC 09 training set (2819 annotations for which we added keypoints), and 240 annotations we added manually from Flickr. We doubled this set by mirroring the images horizontally. Our training algorithm consists of the following steps:

1. Collecting patches. We select 500 random windows from the training set (*seed* windows), sampling with uniform distribution over scale and position while keeping a fixed aspect ratio of 1.5. For each seed window we extract patches from other training examples that have similar local keypoint configuration. Following [14], we compute a similarity transform that aligns the keypoints of each annotated image of a person with the keypoint configuration within the seed window and we discard any annotations whose residual error is too high. In the absence of 3D annotation, we propose the following distance metric:

$$D(P1, P2) = D_{proc}(P1, P2) + \lambda D_{vis}(P1, P2), \tag{3}$$

where D_{proc} is the Procrustes distance between the common keypoints in the seed and destination patch and D_{vis} is a visibility distance, set to the intersection over union of the keypoints present in both patches. D_{vis} has the effect of ensuring that the two configurations have a similar aspect, which is an important cue when 3D information is not available. Note that we allow rotations as part of the similarity transformation during alignment, which helps augment the useful part of the training set for a poselet.

2. Classifier training. We construct HOG features [7] from the collected patches and from random negative example patches and we train linear SVM classifiers. One important difference from [14] is that instead of using all patches as training examples we only use the nearest 250 training examples. Given the size of our training set, this ensures that all the training patches are sufficiently close to the "seed" patch. Otherwise, what may happen is that, e.g., as we collect more examples for a profile face detector, we will eventually start including examples of frontal faces, and they will end up dominating the classifier. Following standard practice, we bootstrap the initially trained SVMs by scanning over images that contain no people, collecting hard false positives and retraining. This process culminates in 500 trained poselet classifiers.

3. Finding true and false positives. We do a scanning pass over our training set and collect the top $2N$ activations of each poselet, where N is the number of annotated people. We assign labels (true positive, false positive, unknown) to each poselet activation. To assign a label we use the bounds of the patches we extracted in step 1. We partition the bounds into two classes: the top-rank patches (the training patches) are treated as ground truth; the lower-rank patches are treated as secondary ground truth. Any activation that has intersection over union overlap of more than 0.35 with a ground truth is assigned a true positive label. If the overlap with a secondary ground truth is less than 0.1 or none, it is assigned a false positive label. All other cases remain unlabeled.

4. Collecting information on each poselet.

(a) We fit a logistic over the positive and negative activations and the associated scores to convert SVM scores into probabilities q_i.

(b) We set a threshold for the SVM score that ensures 90% of the positive and unlabeled examples are above the threshold. This allows each poselet's detection rate to match the frequency of the pattern it has learned to detect.

(c) We fit a model for the keypoint predictions conditioned on each poselet by observing the keypoint distributions of the true positive activations of each poselet type. An example is shown in Fig. 2. We model the distributions using a 2D Gaussian associated with each keypoint.

(d) We fit the prediction of the visible bounds of the human relative to the poselet in a similar way using the true positive activations. We find the mean and variance of x_{min}, y_{min}, x_{max}, y_{max} of the visible bounding box.

5. Poselet selection. The 500 poselet detectors trained in the previous stages are based on randomly selected seed windows, and as a consequence some of these will be redundant and others will correspond to rare patterns. This suggests that we could select a smaller subset that could provide nearly equivalent or even better performance to the whole set (analogous to feature selection for classifiers). We treat this as a "set cover" problem, and solve it using a greedy strategy. For every positive example in the training set, we determine which poselets "cover" it, in the sense that the poselet has an above threshold activation which overlaps it sufficiently (step 3 above). We first pick the poselet that covers the most examples, then incrementally add poselets that cover the most not yet covered examples. Once there is no poselet that can cover any previously uncovered example, we select the poselet that covers the most examples covered by only one previous poselet, etc.

4 Exploiting Context among Poselets

When examining poselet activations, it becomes clear that they are far from perfect. This is but to be expected; the low level signal captured by HOG features is often ambiguous. Sometimes there is just not enough training data, but sometimes there are also "near-metamers"; patterns that can be distinguished by a human observer using additional context, but are almost indistinguishable given the HOG signal inside the image patch. For example, a back-facing head-and-torso pattern is similar in appearance to a front-facing one, and thus a back-facing poselet will often fire on front-facing people as well; see Fig. 3. Another example is a left leg, which in isolation looks very similar to a right leg.

One can resolve these ambiguities by exploiting context – the signal within a patch may be weak, but there is strong signal outside the patch or at a different

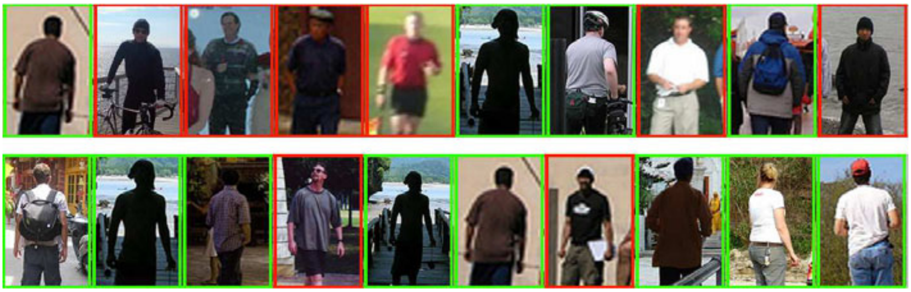


Fig. 3. The top 10 activations of a poselet trained to find back-facing people. **Top row:** Sorted by q-score. **Bottom row:** Sorted by Q-score. The correct and false activations have a green or red bounding box, respectively. The Q-scores are computed using the context of other activations, e.g. frontal faces, to disambiguate front-facing from back-facing people. Without context we make 6 mistakes (top) whereas using context we make only two mistakes (bottom).

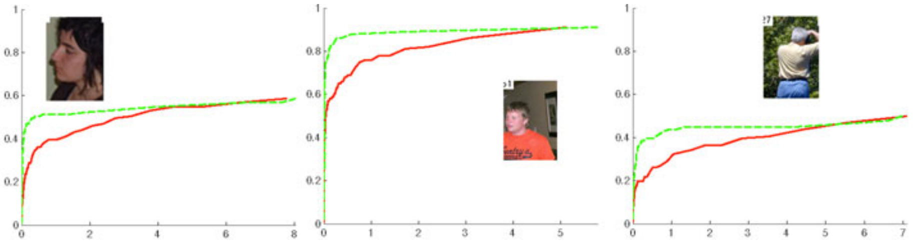


Fig. 4. ROC curves for activations of three poselets computed on our test set. Red continuous lines use q score and green dashed lines use Q score.

resolution. We use the pattern of neighboring poselet activations for disambiguation. For example if a frontal face poselet fires strongly, we can infer that we are more likely to have a front-facing head-and-shoulder pattern, rather than a back-facing one. The oval shape of a wheel sometimes triggers a face detector, but we can suppress the detection if there is no torso underneath.

We refer to the score of a poselet activation based only on its classifier as **q-score** and one that uses other nearby activations as **Q-score**. For each activation i we construct a context feature vector F_i of size the number of poselet types. The pth entry of F_i is the maximum q-score q_j over all activations j of poselet p that are consistent with activation i (or zero if none). We train a linear SVM on the context feature vectors of activations in the training set using their true and false positive labels. We then train a logistic to convert the SVM score into a probability Q_i. The result is what we call Q-score.

We treat two activations i and j as consistent if the symmetrized KL divergence, as defined in (2), $d_{i,j} < \tau$. We set τ as the threshold that best separates distances among consistent activations from distances among inconsistent activations on the training set. For all pairs of labeled activations on the training set we can determine whether they are consistent or not - namely, two activations are consistent if they are both true positives and share the same annotation.

Fig. 3 shows examples of the top activations of our back-facing pedestrian sorted by q-score and below them the corresponding top activations sorted by Q-score. Fig. 4 shows typical ROC cuves with q-scores vs Q-scores for the same poselet. Clearly, the mutual context among activations helps to obtain a better ranking. It is worth noting that Q-scores are assigned to the same activations as the q-scores. While the ranking is changed, the localization of the activation stays the same.

5 Clustering Poselet Activations

Our earlier approach in [14] is build upon the Max Margin Hough Transform from [17] in order to group poselet activations to consistent people detections. This comes with the assumption that the object has a stable central part and the relative position of all other parts has very small variance – an assumption that is not satisfied for articulated objects, such as people. We propose an alternative clustering algorithm:

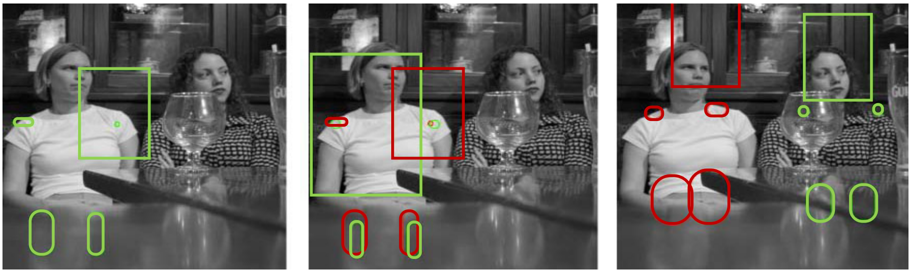


Fig. 5. Examples of poselet activations during clustering. The activation bounding boxes and the predictions of the hips and shoulders are shown. **Left:** We start with the highest probability activation, which for this image is a left shoulder. **Center:** Example of two compatible activations which will be placed in the same cluster. **Right:** Example of incompatible activations which will end up in separate clusters.

1. Initialize the set of clusters that correspond to person detection hypotheses $M = \{\emptyset\}$.
2. Successively take the poselet activation a_i with the highest score Q_i:
   (a) Find the closest cluster $m_j = \text{argmin}_{m_j \in M}\, d(a_i, m_j)$, where the distance d from a_i to cluster m_j is estimated using average linkage.
   (b) If $d(a_i, m_j) < \tau$ then $m_j \leftarrow merge(m_j, a_i)$, i.e. we merge i into an existing cluster. Otherwise, if $|M| < t$ then $M \leftarrow \{M \cup a_i\}$, i.e. we form a new cluster.

In the end the poselet activations are grouped into clusters each corresponding to a person detection hypothesis. In addition some poselets with low scores that are inconsistent with any clusters are marked as false positives and are discarded. The parameter t is a tradeoff between speed and false positive rate. We set $t = 100$, i.e. we collect at most 100 person hypotheses from each image.

This algorithm is a form of greedy clustering starting with the highest- probability poselet activations. Compared to other schemes such as spectral clustering or agglomerative clustering, the proposed algorithm has computational advantages because it processes the most salient information first. The algorithm runs in linear time. We do not spend compute cycles measuring distances between low scoring detections, and the algorithm can be terminated at any time with a good list of the most-salient-so-far hypothesis M. Furthermore, by starting with the highest probability detections we are less likely to be mislead by false positives. Fig. 5 shows examples of merging compatible activations (center) and forming a new cluster (right).

6 Locating and Scoring People Hypotheses

Given a cluster of poselet activations, we can predict the location of the torso, as well as a visible bounding box. We can also compute a score S, which is a measure of how likely the cluster corresponds to a person as opposed to being a false positive.

1. Torso prediction. The human torso is a more stable region to predict than the visible bounding box. Thus, before applying non-maximum suppression, we first predict torsos and derive visible bounds from these predictions. The torso can be predicted from the poselet activations within a cluster. If we use context, we also include all compatible activations that might not be in the cluster. We predict the locations of the hips and shoulders as the average prediction of each poselet activation, weighted by the score of the activation. These four keypoints define the torso of the person, which we parameterize using (x,y) location, length and angle. We use a fixed aspect ratio of 1.5.

2. Non-maximum suppression. We use agglomerative clustering to merge clusters whose intersection-over-union of torso bounds is greater than 0.6.

3. Visible bounds prediction. For each activation in the merged clusters we compute its prediction for the expected visible bounds x_{min}, y_{min}, x_{max} and y_{max} and the associated variances. We then perform mean shift for each of the four estimates independently and pick the dominant mode. Mean shift allows us to take into account the variance of the prediction, which is important. A frontal face poselet, for example, has a very reliable prediction for y_{min}, but is very unreliable for y_{max} since sometimes the legs of the person may be occluded.

4. Improving the predicted bounds. The above generative bounding box prediction is not very accurate and we enhance it using a linear regression similar to [12]. Specifically we transform $[x_{min} y_{min} x_{max} y_{max}]$ with a 4x4 regression matrix T. To train T, we perform steps 1, 2, and 3 on the training set, we match the bounds predictions to the ground truths using intersection over union overlap of 0.45 and collect the true positives. We then fit T using the predicted bounds and the associated ground truth bounds.

5. Computing the score of a poselet cluster. We follow [14] to predict the score S of the poselet cluster, i.e., we train a linear discriminative classifier with positivity constraints on its weights to predict the scores based on the activations within the cluster. We can use q-scores or Q-scores here, and we will show a comparison in Section 8. For our positive examples we use detections on the training set whose bounds intersection over union overlap is over 0.5. For negative examples we use detections that do not intersect the truth or whose overlap is less than 0.1. Our feature vector has the dimensionality of the number of poselet types. The feature value for each poselet type is the maximum of all activations of that poselet type within the cluster.

7 Object Segmentation by Contour Alignment

While prediction of bounding boxes is a reasonable proxy for the object detection problem, the final localization task is actually the segmentation of the detected object. From training examples with segmentation masks available we can derive a figure/ground predictor for each poselet. We use a simple shape model for each poselet by just averaging the masks of all examples in the training images after keypoint alignment.

At test time, we can derive a shape prior for the object by integrating the mask predictions $\phi_i : \mathbb{R}^2 \to [0,1]$ of all poselet activations $i = 1, ..., n$ assigned to one cluster. Rather than just averaging the masks, we weight them by the activation scores Q_i

$$p_{\text{in}}(x,y) = \frac{\sum_{i=1}^{n} Q_i \chi_i(x,y) \phi_i(x,y)}{\sum_{i=1}^{n} \chi_i(x,y)}, \tag{4}$$

where $\chi_i : \mathbb{R}^2 \to \{0,1\}$ denotes the indicator function for the poselet's support in the image. As we are interested in a binary segmentation, the soft masks are thresholded at $\theta_m = 0.07$. This value has been optimized for the PASCAL VOC 09 validation set.

The above procedure yields an a priori decision on which pixels belong to an object given the detection of certain poselets. It so far ignores further indication from the image. In order to get a more precise localization of object boundaries we align them to contours in the image. We use the state-of-the-art boundary predictor from [18] to obtain an edge map $f : \mathbb{R}^2 \to [0,1]$ of the image. Moreover, we extract the silhouette $g : \mathbb{R}^2 \to \{0,1\}$ of the predicted binary mask. We then estimate the deformation field $(u,v) : \mathbb{R}^2 \to \mathbb{R}$ that minimizes

$$E(u,v) = \int_{\mathbb{R}^2} |f(x,y) - g(x+u, y+v)| + \alpha \left(|\nabla u|^2 + |\nabla v|^2\right) dxdy. \tag{5}$$

The parameter $\alpha = 50$ determines the amount of flexibility granted to the deformation field. We use a coarse-to-fine numerical scheme known from optical flow estimation to compute the minimizer of (5) [19]. Warping the initial binary mask with the optimum deformation field (u,v) yields a mask that is aligned with boundaries in the image.

For segmenting the whole image, we paste the aligned binary masks from all clusters into the image domain, ignoring clusters with an overall score $S \leq 12$. Since we run the segmentation for only one category, the ordering of the single detections has no effect.

8 Experiments

Table 1 investigates the effect of the amount of poselets showing results using 10, 40, 100, and 200 selected poselets. Clearly, more poselets first help improving the detection performance, but the improvement saturates between 100 and 200 poselets. Table 1 also shows the positive effect of exploiting mutual context between poselet activations. The AP with Q-scores is consistently larger.

As an important baseline comparison to our previous detection in [14], we evaluated our new detector, using 200 poselets, on the task of detecting human torsos on the H3D test set. The ROC curves are shown in Fig. 6. The new ROC curve outperforms the one from [14] over the entire range. In [14] we required 256 poselets and we also scanned the horizontally flipped version of the image, which has the effect of doubling the poselets to 512.

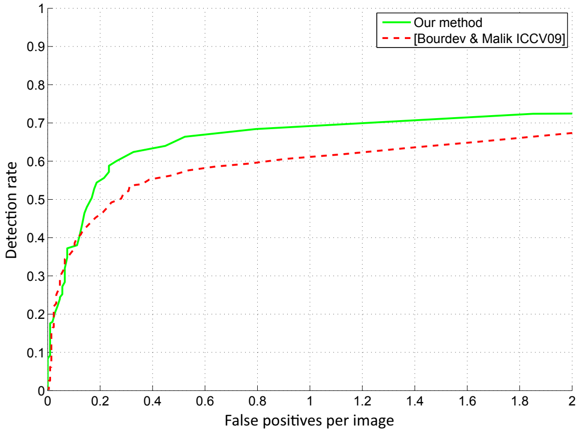


Fig. 6. Performance on the human torso detection task on the H3D test set

Table 1. AP on PASCAL VOC 2007 test set for various numbers of poselets using q-scores or Q-scores as described in Section 4

Num. poselets	q-scores	Q-scores
10	36.9%	37.8%
40	43.7%	44.3%
100	45.3%	45.6%
200	45.7%	46.9%

Table 2. Our performance on PASCAL VOC compared to the currently best results reported for the detection and segmentation tasks on the person category. The segmentation results were produced with 200 poselets.

	Detection				Segmentation		
	100 poselets	200 poselets	[12]	[13]	masks only	alignment	[20]
VOC 2007	45.6%	46.9%	36.8%	43.2%			
VOC 2008	54.1%	52.6%	43.1%		41.9%	43.1%	41.3%
VOC 2009	47.8%	46.9%		43.8%	39.4%	40.5%	38.9%

Finally we provide results on the person category of the recent PASCAL VOC challenges. As reported in Table 2, we have the best results reported to date, both in the detection and the segmentation challenge. Our results are reported for the competitions 4 and 6 because our method requires 2D keypoint annotations.

Table 2 also shows the impact of aligning the mask predictions to boundaries in the image. It is relatively small in quantity, as the performance is mainly due to the detector. The visual effect is much larger, as segmentations align well with true object boundaries. Some example detections and segmentations are shown in Fig. 7 and Fig. 8.

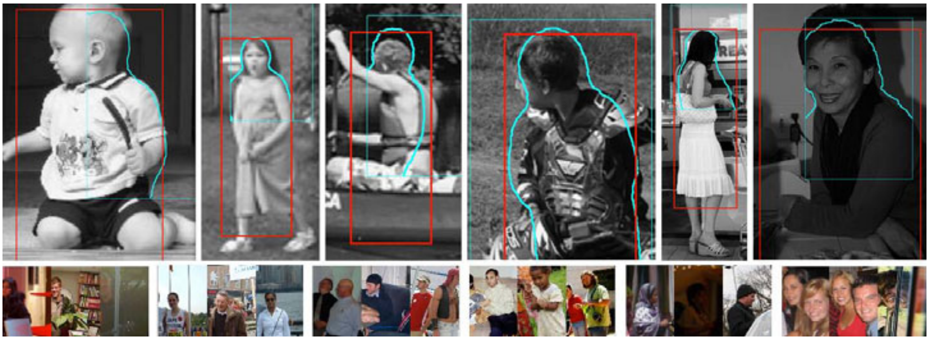


Fig. 7. Detection examples. The person's bounding box is shown in red. The highest probability poselet activation is shown in a cyan bounding box and a figure/ground outline. Below each image we show three training examples from the activated poselet.

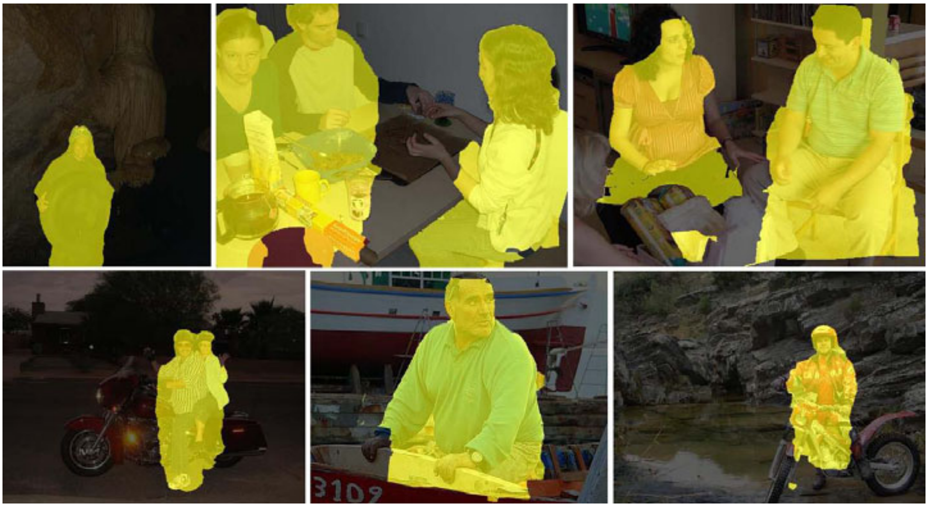


Fig. 8. Segmentation examples. The top middle example shows a typical limitation in case of occlusion by other objects.

9 Conclusion

It is possible to view the poselets approach in a natural sequence of increasing complexity from (1) Dalal and Triggs' [7] single holistic model to (2) Felzenszwalb et al.'s [12] parametric part model on to (3) poselets. In [12], certain fixed choices are made: one root filter, six part filters, two components. The poselet framework can be thought of as being in the spirit of nonparametric statistics – models with greater flexibility which, as more training data becomes available, are expected to have superior performance. These performance improvements do not come at an inordinate expense in terms of running time. On a 3GHz Macbook Pro our

Matlab implementation with 40 poselets runs in about 27 seconds per image, where a large part of the time is spent for HOG computation. We conclude by noting that the approach described in this paper for detecting people is equally applicable to other object categories. This is the subject of ongoing research.

References

1. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61, 55–79 (2005)
2. Ren, X., Berg, A.C., Malik, J.: Recovering human body configurations using pairwise constraints between parts. In: ICCV, pp. 824–831 (2005)
3. Ramanan, D.: Learning to parse images of articulated bodies. In: NIPS (2006)
4. Ferrari, V., Marin-Jimenez, M., Zisserman, A.: Progressive search space reduction for human pose estimation. In: CVPR (2008)
5. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR (2009)
6. Bergtholdt, M., Kappes, J., Schmidt, S., Schnörr, C.: A study of parts-based object class detection using complete graphs. IJCV 87, 93–117 (2010)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
8. Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., Poggio, T.: Pedestrian detection using wavelet templates. In: CVPR, pp. 193–199 (1997)
9. 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)
10. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV workshop on statistical learning in computer vision, pp. 17–32 (2004)
11. Gavrila, D.M.: A Bayesian, exemplar-based approach to hierarchical shape matching. PAMI 29, 1408–1421 (2007)
12. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI (2009) (published online)
13. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models (2010), Project website: `http://people.cs.uchicago.edu/~pff/latent`
14. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3d human pose annotations. In: ICCV (2009)
15. Taylor, C.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. CVIU 80, 349–363 (2000)
16. Crandall, D., Felzenszwalb, P., Huttenlocher, D.: Spatial priors for part-based recognition using statistical models. In: CVPR, pp. 10–17 (2005)
17. Maji, S., Malik, J.: Object detection using a max-margin hough tranform. In: CVPR (2009)
18. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: an empirical evaluation. In: ICCV (2009)
19. 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)
20. Yang, Y., Hallman, S., Ramanan, D., Fowlkes, C.: Layered object detection for multi-class segmentation. In: CVPR (2010)

Disparity Statistics for Pedestrian Detection: Combining Appearance, Motion and Stereo

Stefan Walk[1], Konrad Schindler[1,2], and Bernt Schiele[1,3]

[1] Computer Science Department, TU Darmstadt
[2] Photogrammetry and Remote Sensing Group, ETH Zürich
[3] MPI Informatics, Saarbrücken

Abstract. Pedestrian detection is an important problem in computer vision due to its importance for applications such as visual surveillance, robotics, and automotive safety. This paper pushes the state-of-the-art of pedestrian detection in two ways. First, we propose a simple yet highly effective novel feature based on binocular disparity, outperforming previously proposed stereo features. Second, we show that the combination of different classifiers often improves performance even when classifiers are based on the same feature or feature combination. These two extensions result in significantly improved performance over the state-of-the-art on two challenging datasets.

1 Introduction

Pedestrian detection has been an active research area and significant progress has been reported over the years. An important lesson from previous research is that combining complementary cues is vital to improve state-of-the-art performance. Gavrila&Munder [1] and Ess et al. [2] combine appearance with stereo cues to detect pedestrians from moving vehicles, with the stereo components as modules for candidate generation and post-verification. Dalal et al. [3] and Wojek et al. [4] combine appearance and motion features in a sliding window framework, significantly improving performance. Despite impressive advances reported in the literature, state-of-the-art detectors seldom satisfy application requirements and leave ample room for improvement.

This paper advances pedestrian detection in two ways: first, we contribute a novel feature for pedestrian detection in stereo images, which we use in combination with standard appearance and motion cues. Despite its simplicity, the new feature yields significant improvements in detection performance. Second, we explore the potential of classifier combination for pedestrian detection. While the combination of different features [1,2,3,4] has been key to recent progress, the combination of different classifiers for the *same* feature has not been explored in the context of pedestrian detection to the best of our knowledge. The benefit of both contributions is analyzed and discussed in detail using two different recent pedestrian datasets.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 182–195, 2010.

2 Related Work

Early work on pedestrian detection by Papageorgiou and Poggio [5] and Viola et al. [6] used wavelet features. Viola et al. used a cascade of boosted classifiers, while Papageorgiou and Poggio use an SVM with a quadratic kernel. In [6] temporal information is included by taking intensity differences to adjacent frames (shifted to multiple directions).

Many techniques have been published since then, greatly improving performance – the datasets used to evaluate the early works are essentially solved now. Enzweiler&Gavrila [7] and Dollár et al. [8] recently published surveys on monocular pedestrian detection. For datasets with strong pose variations, such as sport scenes, articulated models like [9] provide best performance. For "standard" pedestrians, which are in an upright pose (as they are when standing or walking), monolithic global descriptors applied in a sliding window framework are still state of the art [8,4].

A pedestrian detector usually consists of candidate generation, followed by feature extraction for the candidate windows, classification of the feature vector, and then non-maximum suppression to prevent multiple detections on a single pedestrian. The most popular method of generating candidate windows is the sliding-window framework, where the scale/position space is sampled with fixed strides. Other work (e.g. [1]) utilizes some method of region-of-interest generation in order to reduce the number of candidate windows and filter out negative samples at an early stage.

The dominant appearance features are variants of the HOG descriptor [10,11,12] and different flavors of generalized Haar wavelets [6,8]. To encode motion information, [6] encodes wavelets on temporal intensity differences. [10,13] encode differences of optical flow into local histograms, similar to HOG.

Stereo information is commonly used in separate modules of pedestrian detection and tracking systems [1,2]. [1] use stereo information in two ways: first, they identify regions of interests in the disparity maps; after the pedestrian detection step, hypotheses are verified by cross correlation between the two images – if there is no object at the estimated disparity level, the correlation measure is low and the hypothesis is rejected. In the model used by [2], the disparity map is used for ground plane estimation, to ensure that detections have a reasonable size (using a prior on human height), and to verify that a pedestrian detection has consistent depth. Rohrbach et al. [14] use the depth field generated by a dense stereo matcher as input for the HOG descriptor to build HOG-like histograms on the depth gradient. Rapus et al. [15] utilize a low-resolution 3D camera (time-of-flight principle), and extract multiple features, including gradients and Fourier coefficients, from intensity and depth to detect pedestrians.

The most wide-spread classifiers are statistical learning techniques to separate positive and negative instances in the feature space. Popular algorithms are support vector machines [16,10,17,18] and variants of boosting [6,19,20,4]. Duin&Tax [21] perform experiments regarding the combination of multiple classifiers on a digit recognition dataset. They found that, while combining complementary features provides the largest gain, combining different classifiers trained

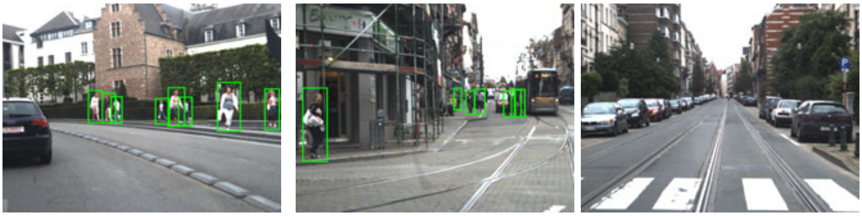


Fig. 1. Sample images from the new auxiliary training set. The last image is from the negative set.

on the *same* features can also help. We show that this also holds in the pedestrian detection setting.

3 Datasets

We use two different challenging datasets for our tests. Both databases have been recorded from a moving car in scenarios with many pedestrians: *ETH-Loewenplatz* [2,22,23] and *TUD-Brussels* [4]. Since we want to build a detector that utilizes both motion and stereo information, we are constrained in our choice of training data. We use two datasets: *TUD-MotionPairs* [4] and a new, auxiliary dataset to train the stereo-based component of our detector.

ETH-Loewenplatz. Our first test set consists of a video sequence of 800 consecutive stereo frames taken from a moving car, with annotations every 4 frames. In total it contains 2631 annotations, however we scan only for pedestrians bigger or equal to 48 pixels in size, which leaves us with 1431 annotations for evaluation.

TUD-Brussels. The second test set has 508 annotated frames recorded from a moving car. It originally had 1326 pedestrian annotations, but there were some small pedestrians missing. We supplemented those, resulting in a total of 1498 pedestrian annotations, with 1235 of them at least 48 pixels high. The dataset allows for optic flow estimation, but there is no published stereo information. However the authors kindly provided us with stereo pairs for this dataset.

TUD-MotionPairs. This dataset is used for training and contains 1776 pedestrian annotations in 1092 images, including the following frame for each annotated frame (to compute optical flow). The images are recorded in a pedestrian zone from a handheld camera, with pedestrians seen from multiple viewpoints. 192 image pairs without pedestrians, partly taken from a handheld camera and partly from a moving car, serve as negative set.

Auxiliary Training set. As *TUD-MotionPairs* does not contain stereo information, we have created a new dataset to train our stereo classifiers. The new dataset contains 2570 annotations in 824 frames in the positive set, with stereo and motion information available. However, most of the pedestrians in this set are small (2033 of them are smaller than the detection window,resulting in suboptimal quality). The negative set contains 321 frames, again with motion and stereo information. The images have a resolution of 640x480 pixels and were recorded from a moving car. Sample images are shown in Figure 1.

4 Baseline Features and Classifiers

The set of features and classifiers we use as baselines includes HOG [10] and HOF [3] as features, and SVMs and MPLBoost [24,25] as classifiers. The same features and classifiers were used recently in [4].

HOG. Dalal&Triggs proposed using histograms of oriented gradients in [10]. In HOG, every pixel votes for its gradient orientation into a grid of histograms using trilinear (spatial and orientation) interpolation. Local normalization is employed to make the feature robust against changes in illumination. Interpolation and histogramming makes the feature robust with regard to small changes in pose.

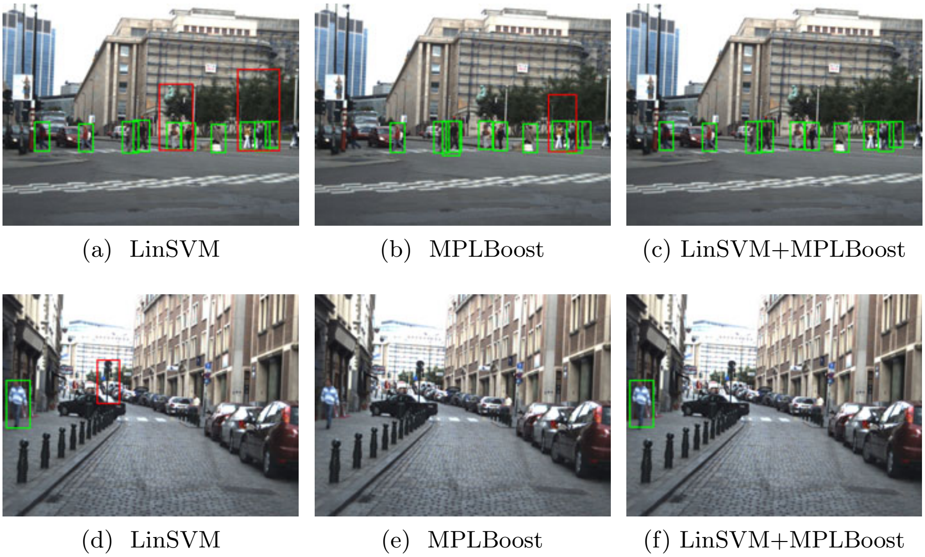
HOF. Histograms of Flow were introduced in [3] to encode motion information from optical flow. We use a reduced variant of the original IMHcd scheme with 2x2 blocks. Our version is on par with the original HOF in terms of performance. Flow fields are estimated with the publicly available optical flow implementation by Werlberger et al. [26].

SVM. Support Vector Machines are currently the standard for binary classification in computer vision. Linear SVMs learn a hyperplane that optimally separates negative and positive samples in high-dimensional feature space. Kernel SVMs are also possible, however their high computation time makes them intractable for sliding-window detection with high-dimensional feature vectors. An exception to this are histogram intersection kernels (HIKSVMs), for which an approximation can be evaluated in constant time [27].

MPLBoost. MPLBoost is an extension to AdaBoost[6], where K strong classifiers are learnt jointly, with each strong classifier focusing on a subset of the feature space. The final confidence is the maximum over the K classifiers, so only one of them needs to correctly identify a positive sample. Unless noted otherwise, we use $K = 4$ strong classifiers.

For training, negative samples are first randomly drawn from the negative training set to create an initial classifier. With this classifier the negative training images are scanned for *hard negatives* that get misclassified. These are added to the negative set and the classifier is retrained. We repeat this *bootstrapping* step twice to ensure that the result is minimally influenced by the random choice of the initial negative set.

The feature/classifier *components* we are using throughout the paper were previously studied in our paper [4]. Due to optimizations and changes in training procedure, there are some differences. Figure 5(b) compares the implementations. The three dotted lines compare the "old" HOG-detector (red dotted line) and HOG+Haar-detector (green dotted line) with our HOG-implementation (blue dotted line). Similarly the "new" HOG+HOF-feature (blue solid line) performs similar to or better than the previous HOG+HOF+Haar-feature (green


(a) LinSVM (b) MPLBoost (c) LinSVM+MPLBoost

(d) LinSVM (e) MPLBoost (f) LinSVM+MPLBoost

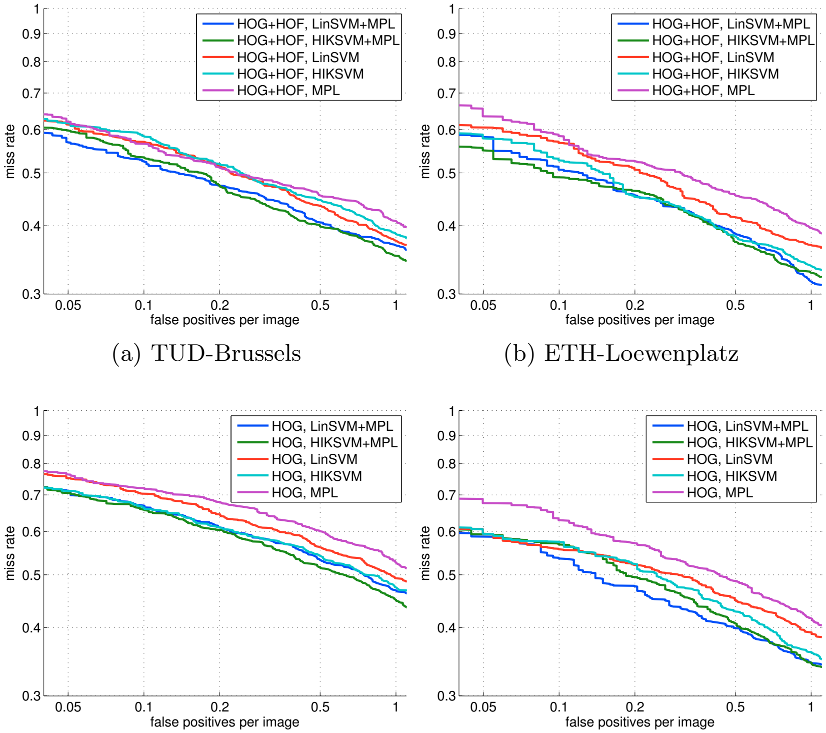
Fig. 2. MPLBoost and SVMs perform well but tend to have different false positives (a,b,d,e – red boxes correspond to false positives). By combining both classifiers the false positive rate can be reduced (c,f).

solid line) and HOG+HOF-feature (red solid line). Note that we do not use Haar features as in [4] we found them not to be beneficial in all cases.

5 Combination of Classifiers

It is well-established that utilizing a combination of complementary cues significantly boosts detection performance. E.g. Gavrila et al. [1] use shape, texture, and stereo cues to build a detection system while Wojek et al. [4] use multiple features (including appearance and motion information) to boost detection performance. Rohrbach et al. [14] fuse classifiers separately trained on intensity and depth. In these cases, the complementarity of the classifiers results from the cues being from different sources (such as stereo and motion information) or from the sources being encoded into different features. However, those are not the only sources of complementary information.

In [4], we noticed that MPLBoost and SVMs, while both giving good performance, tend to produce different false positives using the *same* feature set. For true positives, different classifiers are likely to give a positive answer, while for false positives the classifiers do not necessarily agree. See figure 2 for examples where LinSVM and MPLBoost (for the feature set HOG+HOF) produce different false positives (2(a,d) and (b,e) respectively). This gives a strong hint that by combining SVM and an MPLBoost classifiers, one can reduce the false positive rate. See figure 2(c,f) where such a combination eliminated false positives. This combination is described in the following.



(a) TUD-Brussels

(b) ETH-Loewenplatz

(c) TUD-Brussels

(d) ETH-Loewenplatz

Fig. 3. Results using classifier combination on *TUD-Brussels* and *ETH-Loewenplatz* with HOG+HOF and HOG alone as features. The single-component detectors are on par with the best published ones on *TUD-Brussels* from [4] (figure 5(b)), combining multiple classifiers yields a noticeable improvement.

Starting from the above observation, this paper explores the possibility to combine classifiers not only for different features but also for the same feature. The combination of classifiers for the same features is especially interesting as it is "cheap": Feature extraction is computationally expensive and often the bottleneck in today's systems. When combining classifiers on the same feature space, the feature vector has to be computed only once.

Classifiers are already combined at the training stage, which influences the bootstrapping phase: a window gets registered as a hard sample if it's hard for the *combined* classifier, enabling the classifiers to focus on data that is problematic for the final detector. This results in slightly better performance than training them separately. The combinations that we study in this section are linear SVM+MPLBoost and HIKSVM+MPLBoost, both trained on the same feature space, HOG+HOF. Combining a linear SVM with an HIKSVM did not show any improvement and thus is not reported here.

As noted before, one can expect classifier combination to improve classification if the combined classifiers have complementary characteristics. A (confidence-rated) classifier is a mapping from the feature vector space to a score. For an imperfect (but better than chance) classifier, the *probability density functions* (*pdf* s) of the positive and negative classes are overlapping. Under the reasonable assumption that the mean of the positive pdf is higher than the mean of the negative pdf, we can – without loss of generality – rescale the mapping so that the means of the positive and negative pdfs are at +1 and -1, respectively. Classification errors (caused by the overlap of the pdfs) can then be expected to decrease when the variance decreases. The variance σ^2_{x+y} of a weighted sum $\alpha x + \beta y$ of classifiers x and y ($\alpha + \beta = 1$) for a given class is $\sigma^2_{x+y} = \alpha^2\sigma^2_x + 2\alpha\beta\sigma^2_{xy} + \beta^2\sigma^2_y$ with σ^2_{xy} being the covariance. If this is lower than σ^2_x and σ^2_y, the combination can be expected to be beneficial.

Results are shown in figure 3 for the two test sets. For comparison, results for individual classifiers are shown as well. For *TUD-Brussels* and the feature combination HOG+HOF (Fig. 3(a)) the two combined classifiers (blue and green curves) clearly improve performance over the individual classifiers (red, cyan, violet curves). For *ETH-Loewenplatz* (Fig. 3(b)) the improvement of the combinations (blue, green curves) over the individual classifiers is also visible.

At 0.1 false positives per image the best combined classifier for HOG+HOF (Linear SVM + MPLBoost) has 4.2% more recall than the best single component classifier on *TUD-Brussels*, and 3.7% more recall on *ETH-Loewenplatz*. Using only HOG as feature, a smaller improvement can be observed over the best individual classifier for *TUD-Brussels* (see Fig. 3(c)) while on *ETH-Loewenplatz* the improvement is substantial at higher false positive rates: 5% improvement at 0.2 fppi (see Fig. 3(d)).

The results reported so far have been obtained by averaging classifier scores as a confidence measure of the combined classifier. This gives both components equal weight. To see if performance improves when the weights are learned instead, we employ a linear SVM as a top-level classifier with the lower level classifier confidences as inputs. Here, 5-fold cross validation on the training set is used to train the top-level classifier without overfitting: we train on 80% of the training data and evaluate the component classifiers on the remaining 20%, with the cross-validation scores being the feature vectors for the top-level classifier. The final component classifiers are then trained using the whole training set. However, there is no significant improvement over equal weights, which is not surprising, as the classifiers work about equally well. As training takes significantly longer with this approach (≈ 6 times), we do not use it in the rest of the paper. In the context of combining SVM kernels, [28] found that if the kernels are comparable in performance, averaging works well, while learning the combination is important when there are uninformative components, which agrees with our experience.

6 Utilizing Stereo Information

In the previous section, we showed that different classifiers on the same feature set can be combined to form a better classifier. However, the combination of different kinds of features promises a greater possible gain in information and consequently also in performance. One prominent source of information that is complementary to appearance and motion is binocular vision. Using a stereo image pair, we can extract disparity and depth information, which turns out to improve performance considerably.

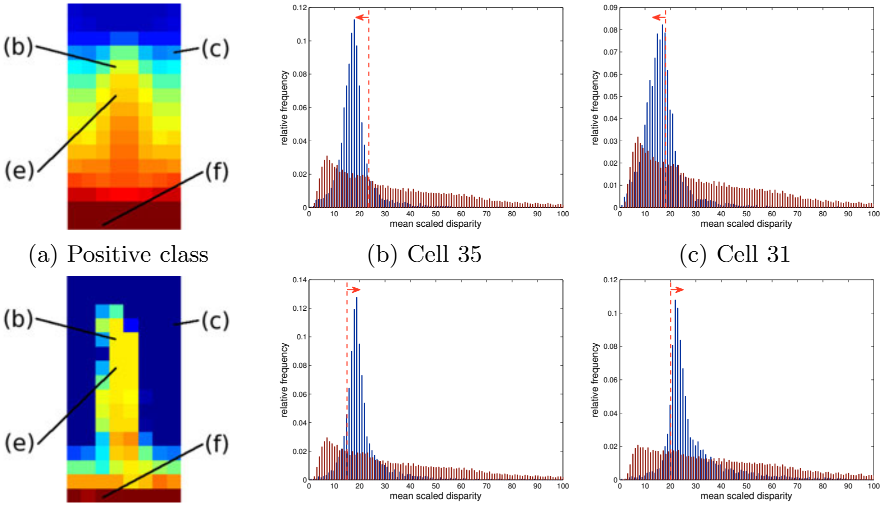
HOS-feature. As a first stereo feature, we use a HOG/HOF-like feature. In [14], Rohrbach et al. computed the HOG descriptor on the depth field, which is inversely proportional to the disparity field, because its gradients are – in theory – invariant to the position of the pedestrian in the world. The gradients in the disparity image are not invariant (they are nonlinearly scaled). However, HOG is designed to provide invariance against scale changes in "intensity" (in this case, disparity). This becomes problematic only for very small disparities, where the nonlinearity are noticeable. On the other hand, using the depth also has its problems: since $Z \propto \frac{1}{d}$, small errors in disparity result in large errors of the depth map; moreover, pixels with disparity 0 have infinite depth and require special handling when building the descriptor, otherwise a single pixel can cause an infinite entry in the histogram. If we directly compute gradients on the disparity map, no special handling is required.

We have experimented with standard HOG descriptors (encoding small-range gradients in depth or disparity) and also with a variant of HOF on the disparity field, where we treat the disparity field like a vector field with the disparity as the x-coordinate and the y coordinate set to 0. The only relevant orientation bins here are the left and right bins: For every pixel, it is encoded if the pixels that are 8 pixels (in the L_∞ norm) away in horizontal, vertical or diagonal direction have a smaller or greater distance to the camera, weighted by the difference. This scheme in principle encodes less information than the full HOG descriptor, however stereo algorithms are not that accurate on a small scale, so long-range differences are more stable. Experimentally we did not observe any significant difference between the performance of this encoding and the encoding proposed by [14]. Therefore, in the following we use the HOF-like descriptor on the disparity field (termed HOS in the following) with a linear SVM as the classifier.

Disparity statistics (DispStat) feature. The disparity field has an interesting invariant property: in the pinhole camera model, the disparity d at a given point is $d = \frac{fB}{Z} \propto \frac{1}{Z}$ with the focal length f, the baseline B, and the depth Z. The observed height h of an object of height H is $h = \frac{fH}{Z} \propto \frac{1}{Z}$

This means that the ratio of disparity and observed height is inversely proportional to the 3D object height; for objects of fixed size that ratio is constant. The heights of pedestrians are not identical, but very similar for most pedestrians. We can therefore, during sliding window search, divide the disparity values by the appropriate scale level determined by the layer of the image pyramid – e.g. for a reference height of 96 pixels and a scaled detection window of 64 pixels, disparities will be multiplied by 1.5. The scaled disparities of positive (pedestrian) samples will then follow a narrow distribution.[1]

[1] If the camera setup is different between the training and test images, the ratio between height and disparity has to be adapted accordingly.



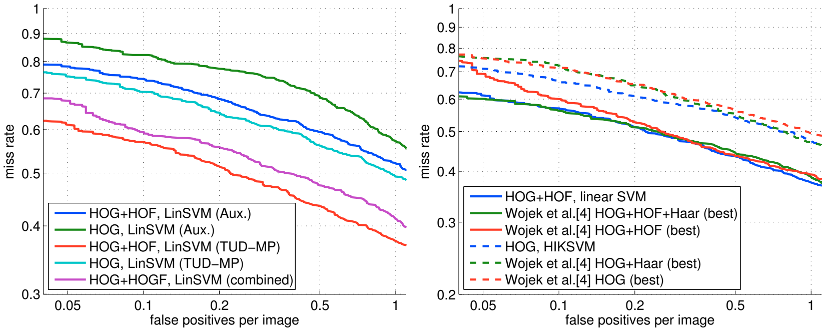
(a) Positive class (b) Cell 35 (c) Cell 31

(d) Sample instance (e) Cell 51 (f) Cell 122

Fig. 4. Visualization of the Disparity Statistics feature. (a) is a color map of the median of the feature values over all positive samples (symmetric because training images get mirrored), (d) of an example training instance. Warmer color corresponds to bigger disparity/nearer points. Clearly, the feature is able to encode information like the pedestrian standing on the ground plane and the area around the upper body being more likely to be behind the pedestrian.

This observation enables us to design a very simple and surprisingly effective feature. We divide the detection window into 8x8 pixel cells (the same as the HOG cell size, for computational efficiency). For each cell, the mean of the scaled disparities is computed. The concatenation of all 8×16 mean values from the 64×128 pixel window is the feature vector. For this feature, we use MPLBoost as classifier with $K = 2$ (more clusters did not help) and 100 boosting rounds.

Figure 4 visualizes the feature. In figure 4(a), the cell-wise median of all positive training samples is shown, 4(d) shows one particular positive training sample. One can immediately see different pieces of information captured by the new descriptor: the surrounding background is typically further away than the person, and the person usually stands on an approximately horizontal ground plane. In figure 4(b,c,e,f) statistics from example cells are shown along with weak classifier boundaries from the MPLBoost classifier. Displayed are the relative per-class frequencies of the disparity values. For the positive class, all 5140 training instances (including mirrored samples) are plotted, to plot the negative class 5 images were sampled densely, with the same parameters as in the sliding window search, resulting in 721900 samples (training of course uses all 321 images of the negative set). The dashed red line shows the weak classifier threshold, with arrows to the right signaling a lower bound, and arrows to the left an upper bound. Note that they are *weak* classifiers – they are only required to work better



(a) Different training sets

(b) Comparison with [4]

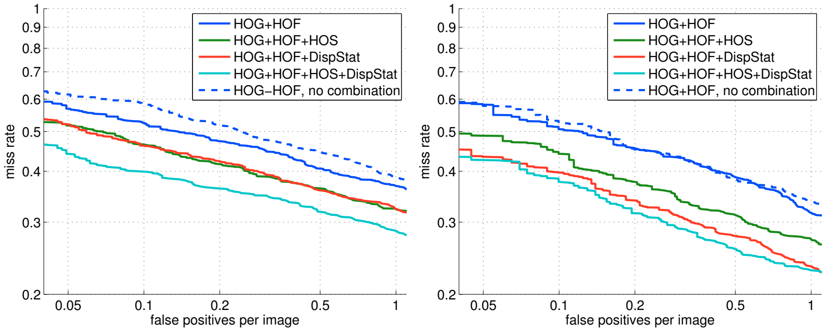
Fig. 5. (a) *TUD-MotionPairs (TUD-MP)* is a better training set than the auxiliary training set *(Aux.)* for appearance and motion information, however it contains no stereo information. Even combining *TUD-MotionPairs* with the auxiliary training set results in inferior performance for our detector when using appearance and motion as cues. In (b), one can see that our components are as least as good as the ones shown in [4].

than chance, so it does not matter if they miss-classify a portion of the training set. Even though the distributions overlap, making learning a non-trivial task, it is obvious that the class distributions are different and something can be learned from this data.

In figure 4(b) and (e), the disparity range for the upper body is evaluated by the weak classifiers, meaning the classifiers learn the size of a pedestrian (since the observed height is fixed – the height of the bounding box under evaluation – the *scaled* disparity relates inversely proportional to a height in 3D).

In 4(c), one weak classifier learned that the area to the right of the pedestrian usually is not closer to the camera than the pedestrian itself (note that the maximum of the distribution is at a lower disparity than the maxima of the distributions for (b) and (e)). However, the distribution here is not as narrow, because it is not uncommon that pedestrians stand next to other objects in a similar depth range. Figure 4(f) visualizes a weak classifier testing that the pedestrian stands on a ground plane, meaning that the cell under the pedestrian is closer to the camera than the pedestrian itself. Note that learning the pedestrian size and the ground plane assumption is completely data-driven.

Combining classifiers for different cues. Finding a dataset to train a detector using depth, motion, and appearance is not trivial: The public designated training sets we are aware of don't have both stereo and motion information available. Our new training set, the *auxiliary training set*, has this, however it is not as good as *TUD-MotionPairs* for appearance and motion, as can be seen in figure 5(a). The detector using HOG+HOF with a linear SVM has over 15% less recall when trained on this set (compare blue and red curves). Even joining the datasets for training results in inferior performance (violet curve).



(a) TUD-Brussels

(b) ETH-Loewenplatz

Fig. 6. Results using stereo information on *TUD-Brussels* and *ETH-Loewenplatz*

To address this problem, we train different components on different datasets, and combine the components with an additional classifier stacked on top, which operates on the outputs of the components. In this section, we take the best combined classifier for appearance and motion (linear SVM + MPLBoost on HOG+HOF trained on *TUD-MotionPairs*) as one component. To combine the appearance/motion with the stereo components, a linear SVM is trained on top of the component outputs to provide the final score. The top-level SVM and the stereo-based classifiers are trained jointly using 5-fold cross validation on the auxiliary training set. To generate dense disparity maps, we used the algorithm of Zach et al. [29].

Results. As can be seen in figure 6, our new feature/classifier combination improves performance significantly. Best results from figure 3 are reproduced for reference: the dotted blue lines are the best performing individual classifier (HOG+HOF); the solid blue lines are the best performing combined classifier. On *TUD-Brussels* (Fig. 6(a)), the new disparity statistics feature combined with our HOG+HOF-classifier (red curve) performs as good as the HOS feature combined with HOG+HOF (green curve), resulting in an improvement of 6.4% recall at 0.1 fppi over the detector using HOG+HOF alone (blue curve). Combining both stereo features (cyan curve), the improvement is 12.6% over the HOG+HOF detector (solid blue curve), and more than 18% better than HOG+HOF with a linear SVM (dashed blue curve), which in turn is slightly better than the best reported result in the literature for this dataset [4] (c.f. figure 5(b)). The improvements are consistent over a wide range of false positive rates.

On *ETH-Loewenplatz* (Fig. 6(b)), adding HOS (green curve) results in an improvement of 6.6% at 0.1 fppi over HOG+HOF (blue curve). Using DispStat in addition to HOG+HOF (red curve) results in a higher improvement than HOS resulting in 11% improvement at 0.1 fppi. Further combining DispStat with HOS (cyan curve) in addition to HOG+HOF improves recall by another 2%. These results clearly show that DispStat is the stronger feature than HOS

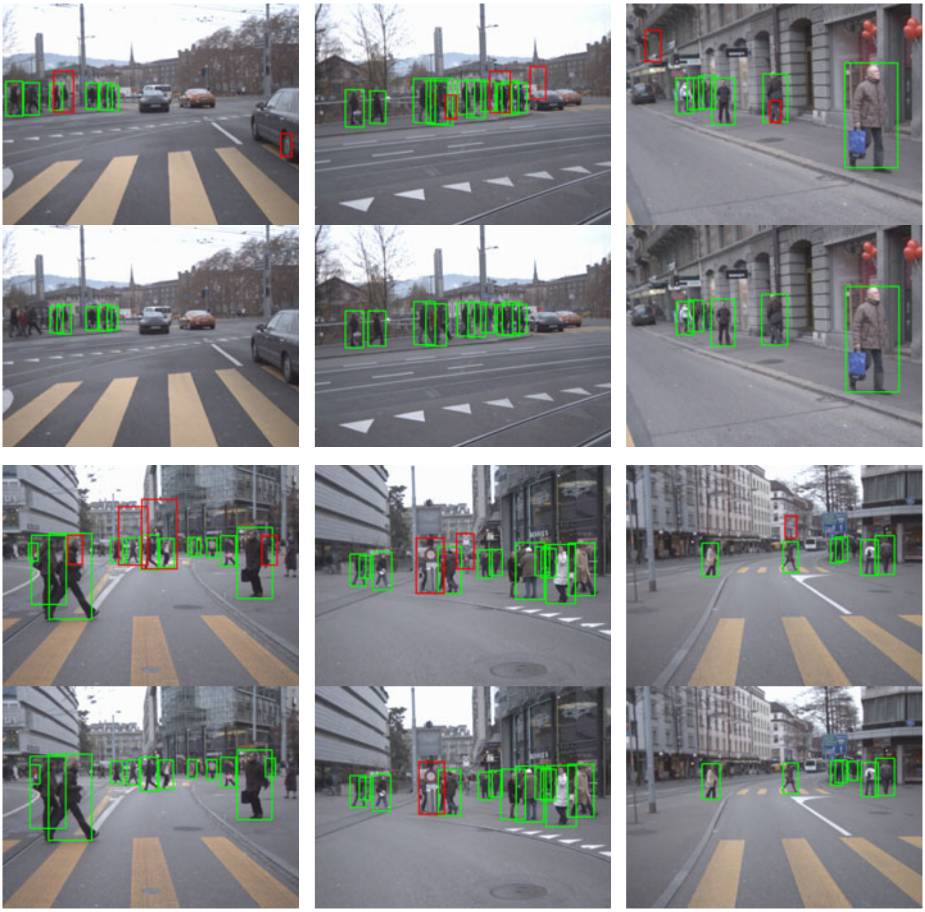


Fig. 7. Sample results using stereo information

for this dataset. Compared to the best single-classifier detector with HOG+HOF as features (dashed blue), the overall improvement is 15%.

Comparing to state-of-the-art performance by [23] (they use a complete system integrating stereo, ground-plane estimation and tracking) our combined detector outperforms their best performance. In their evaluation scheme (pedestrians larger than 60 pixels) we outperform their system by about 5% at 0.1 fppi. This clearly underlines the power of the contributions of this paper to improve the state-of-the-art in pedestrian detection.

Figure 7 show sample results using stereo information. In every pair, the upper image shows the HOG+HOF detector with HIKSVM+MPLBoost, the lower the full detector including HOS and the DispStat feature. Both detectors are shown at the point where they reach 70% recall, so differences are to be seen in the amount of false positives. The stereo features are especially good at eliminating false positives at the wrong scale, or not standing on the ground plane. “Typical” false positives, like car wheels (top left) and body parts (top right, bottom left) are easily filtered out, as well as detections having moving pedestrian as "legs" (bottom left). False positives on objects that are similar in 3d to a pedestrian are still an issue, for example the trash can with a traffic sign in the middle image in the lower row. Since the disparity field suffers from artifacts and missing information at the image border, some pedestrians (e.g. at the left border of the upper left image pair) are missed, however it detects others that the monocular detector misses (as both are tuned to get 70% recall). Also note that in the lower left image the HOG+HOF detector overestimates the size of the pedestrian at the right image border, causing a false positive and a missed detection, while the detector using stereo features correctly estimates the size and position of the pedestrian.

7 Conclusion

This paper consists of two contributions for pedestrian detection. First, we show that combining different classifiers trained on the same feature space can perform better than using a single classifier. Second, we introduce a new feature, called DispStat, for stereo, enabling the classifier to learn scene geometry information (like pedestrian height and the ground plane assumption) completely data-driven, without any prior knowledge. Combining those two contributions, we outperform the best published result on *TUD-Brussels* by over 12%, in combination with an adaptation of HOG for disparity fields similar to [14], this increases to over 18%. We verified these results on a second challenging dataset, *ETH-Loewenplatz*, where the performance of DispStat is even better, outperforming the HOS feature.

Acknowledgments. The authors thank Christopher Zach for providing his implementation of [29] and Christian Wojek for code and valuable discussion.

References

1. Gavrila, D.M., Munder, S.: Multi-cue pedestrian detection and tracking from a moving vehicle. IJCV 73, 41–59 (2007)
2. Ess, A., Leibe, B., Schindler, K., van Gool, L.: A mobile vision system for robust multi-person tracking. In: CVPR (2008)
3. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
4. Wojek, C., Walk, S., Schiele, B.: Multi-cue onboard pedestrian detection. In: CVPR (2009)
5. Papageorgiou, C., Poggio, T.: A trainable system for object detection. IJCV 38, 15–33 (2000)
6. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: ICCV (2003)
7. Enzweiler, M., Gavrila, D.M.: Monocular pedestrian detection: Survey and experiments. In: PAMI (2009)

8. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR (2009)
9. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR (2009)
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
11. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
12. Wang, X., Han, T.X., Yan, S.: A HOG-LBP human detector with partial occlusion handling. In: ICCV (2009)
13. Dalal, N.: Finding People in Images and Videos. PhD thesis, Institut National Polytechnique de Grenoble (2006)
14. Rohrbach, M., Enzweiler, M., Gavrila, D.M.: High-level fusion of depth and intensity for pedestrian classification. In: Denzler, J., Notni, G., Süße, H. (eds.) DAGM 2009. LNCS, vol. 5748, pp. 101–110. Springer, Heidelberg (2009)
15. Rapus, M., Munder, S., Baratoff, G., Denzler, J.: Pedestrian recognition using combined low-resolution depth and intensity images. In: IEEE Intelligent Vehicles Symposium (2008)
16. Shashua, A., Gdalyahu, Y., Hayun, G.: Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In: IVS (2004)
17. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: CVPR (2007)
18. Lin, Z., Davis, L.S.: A pose-invariant descriptor for human detection and segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 423–436. Springer, Heidelberg (2008)
19. Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. In: CVPR (2006)
20. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet part detectors. IJCV 75, 247–266 (2007)
21. Duin, R.P.W., Tax, D.M.J.: Experiments with classifier combining rules. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, p. 16. Springer, Heidelberg (2000)
22. Ess, A., Leibe, B., Schindler, K., van Gool, L.: Moving obstacle detection in highly dynamic scenes. In: ICRA (2009)
23. Ess, A., Leibe, B., Schindler, K., Van Gool, L.: Robust multi-person tracking from a mobile platform. PAMI 31(10), 1831–1846 (2009)
24. Babenko, B., Dollár, P., Tu, Z., Belongie, S.: Simultaneous learning and alignment: Multi-instance and multi-pose learning. In: ECCV workshop on Faces in Real-Life Images (2008)
25. Kim, T.K., Cipolla, R.: MCBoost: Multiple classifier boosting for perceptual coclustering of images and visual features. In: NIPS (2008)
26. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. In: BMVC (2009)
27. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR (2008)
28. Gehler, P.V., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV (2009)
29. Zach, C., Frahm, J.M., Niethammer, M.: Continuous maximal flows and Wulff shapes: Application to MRFs. In: CVPR (2009)

Multi-stage Sampling with Boosting Cascades for Pedestrian Detection in Images and Videos

Giovanni Gualdi, Andrea Prati, and Rita Cucchiara

University of Modena and Reggio Emilia*, Italy
{giovanni.gualdi,andrea.prati,rita.cucchiara}@unimore.it

Abstract. Many works address the problem of object detection by means of machine learning with boosted classifiers. They exploit sliding window search, spanning the whole image: the patches, at all possible positions and sizes, are sent to the classifier. Several methods have been proposed to speed up the search (adding complementary features or using specialized hardware). In this paper we propose a statistical-based search approach for object detection which uses a Monte Carlo sampling approach for estimating the likelihood density function with Gaussian kernels. The estimation relies on a multi-stage strategy where the proposal distribution is progressively refined by taking into account the feedback of the classifier (i.e. its response). For videos, this approach is plugged in a Bayesian-recursive framework which exploits the temporal coherency of the pedestrians. Several tests on both still images and videos on common datasets are provided in order to demonstrate the relevant speedup and the increased localization accuracy with respect to sliding window strategy using a pedestrian classifier based on covariance descriptors and a cascade of Logitboost classifiers.

Keywords: fast pedestrian detection, fast object detection, boosting classifiers, stochastic object detection, statistical object detection, Monte Carlo sampling, multi stage object detection.

1 Introduction and Related Works

Object detection and recognition in images and videos are problems that have been strongly addressed in computer vision in the past years. Some object classes (such as faces, pedestrians, vehicles, characters) received special attention by the research community, since their peculiarities can be "easily" modeled with machine learning techniques and classifiers can be efficiently exploited.

These classifiers are applied on image patches (or "windows") of a given size and in case the object is searched on a whole image, a sliding window search (e.g. [1,2,3]) is normally proposed. The algorithm passes to the window-based classifier all possible windows of an image; the approach has the drawback of brute force methods, that is the high computational load due to the number of windows to check, that grows quadratically in each dimension to span over (typically

* Giovanni Gualdi and Rita Cucchiara are with DII; Andrea Prati is with DISMI.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 196–209, 2010.

three, i.e. image coordinates and scale) [4]. Obviously, this computational load grows up if the search is performed on all the frames of a video. Consequently, several works focus on the reduction of the computational burden, following three main streams: (a) pruning the set of sliding windows by exploiting other cues (e.g. motion [5], depth [6], geometry and perspective [7], or whatever cue that is different from the appearance cue used by the detector itself); (b) speeding up with hardware-optimized implementations (such as GPUs [8]); (c) efficiently exploring the sub-window space through optimal solution algorithms [4,9].

In this paper, we address a new search paradigm to overcome the problem of the sliding window search in a general-purpose manner that does not conflict with all the other aforementioned optimizations (either hardware or software). The proposed method exploits a Monte Carlo sampling to provide an incremental estimation of a likelihood function and our innovative contribution is the use of the response/confidence of the classifier to build such likelihood function. In practice, this response is employed to increasingly draw samples on the areas where the objects are potentially present and avoiding to waste search time over other regions. Although we focus on cascade of boosting classifiers, where the classification is achieved by passing through the stages of the cascade, the proposal could be extended to any classifier that provides a classification confidence.

Mimicking the search of human vision, also [10,11] tackle the problem of optimized object detection. [10] explores the maximization of information gain: although it obtains speed-ups that are comparable to ours, two limitations are suffered: a slight degradation of performances w.r.t. sliding window detection (instead we obtain higher accuracy) and single-target detection (conversely our method is intrinsically multi-target). [11] proposes a deterministic (grid-distributed), multi-stage (coarse-to-fine) detection: successful detections at coarse resolutions yield to refined searches at finer resolutions. We also propose a multi-stage approach; however [11] binarizes the response of the classifier at each stage, while we propose to exploit its continuity, in order to be able to find true detections even when at earlier stages no successful detections are found.

When dealing with videos, the retrieved likelihood function is then plugged into a Bayesian recursive context, through a particle filter. Although this technique is often exploited for object tracking [12,13,14], our proposal does not aim to that achievement, rather it exploits the recursive framework to exploit the temporal coherency of the objects in order to further increase efficiency and accuracy of object detection. When the target distribution is multi-modal (due to ambiguity, clutter or presence of multiple targets), the particle filters suffer of the problem of *sample depletion*, and there are several extensions to handle multi-target tracking [15,16,17] or multi-modal posteriors, such as the *mixture particle filter* [18], where the different targets correspond to the modes of the mixture pdf. This approach has been refined in the *boosted particle filter* [17], where a cascaded Adaboost is used to guide the particle filter. The proposal distribution is a mixture model that incorporates information from both the Adaboost and the dynamical model of the tracked objects. Differently from other methods, we do not generate new particle filters together with the entrance of new objects in the scene: indeed this approach can quickly degrade the performance due to the increase of the number of targets. On the opposite, our proposal is capable to handle a variable number of objects thanks to a quasi-random sampling procedure and a measurement model that is shared among the objects of interest.

Although our proposal is independent on the adopted classifier, on the employed features and on the target class, in this paper we focus on pedestrian detection where many accurate classifiers have been proposed and benchmarked. Dealing with pedestrian classification, a wide range of features has been proposed; among them, Haar wavelets [19], Histogram of Gradients (HoG) [2], a combination of the two [20], Shapelet [21], Covariance descriptors [3], etc.. Over these features, the most typical classifiers are SVMs (typically linear or histogram intersection kernels SVMs [22]) or the boosting algorithms (e.g. AdaBoost [1], LogitBoost [23], MPL Boost [24]) assembled in rejection cascades: this architecture benefits of the property to use a very reduced portion of the rejection cascade when classifying those patches whose appearance strongly differs from the trained model, reducing therefore the computational load. Conversely, the number of stages to pass through increases in a way that is proportional to the appearance similarity of the patch with the target model. An example of such classifier, that we adopt in the present work, is the covariance-descriptor LogitBoost pedestrian classifier proposed by Tuzel *et al.*[3].

Summarizing, the contribution of our work is two-fold. Firstly, by exploiting a known implementation of a boosting cascade classifier, we propose a new object detection approach (in particular, pedestrians) that challenges the typical sliding windows approach: we claim that, by exploiting the only features used by the classifier itself, it is possible to drive a more efficient exploration of the state space of an image. Secondly, we demonstrate that the data obtained by such method can be easily plugged into a Bayesian-recursive filter, in order to exploit the temporal coherency of the moving objects (pedestrians) in videos to improve detection in very cluttered environments. Results demonstrate a significant speedup together with a higher precision in the object localization.

Moreover, with regards to the literature (especially work in [13] that proposes a multi-stage sampling for object tracking) we proposed the following innovations: (i) we perform detection of multiple objects through a single likelihood model; (ii) we handle object entrances and exits; (iii) a new measurement for likelihood is proposed; (iv) a variable number of particles and variable covariances avoid "over-focusing" on true detections; (v) the likelihood is computed exploiting a portion of the samples instead of the whole set.

2 Pedestrian Detection Using a Cascade of LogitBoosts

For pedestrian detection, in [3] the authors proposed the use of a cascade classifier with a cue given by the covariance matrix of a 8-dimensional set of features F (defined over each pixel of I):

$$F = \left[x, y, \left| I_x \right|, \left| I_y \right|, \sqrt{I_x^2 + I_y^2}, \left| I_{xx} \right|, \left| I_{yy} \right|, \arctan \frac{\left| I_y \right|}{\left| I_x \right|} \right]^T \tag{1}$$

where x and y are the pixel coordinates, I_x,I_y and I_{xx},I_{yy} are respectively the first and the second-order derivatives of the image. Then, for any (rectangular) patch of I, the covariance matrix of the set of features F can be computed and used as "covariance descriptor" in the classifier. This descriptor lies on a Riemannian manifold, and in order to apply any classifier in a successful manner, it should be mapped over an Euclidean space. Without digging into details, a detection over a single patch involves the mapping of several covariance matrices (approx. 350) onto the Euclidean space via the inverse of the exponential map [3]: $\log_\mu(Y) = \mu^{\frac{1}{2}} \log\left(\mu^{-\frac{1}{2}} Y \mu^{\frac{1}{2}}\right) \mu^{\frac{1}{2}}$. This operator maps a covariance matrix from the Riemannian manifold to the Euclidean space of symmetric matrices, defined as the space tangent to the Riemannian manifold in μ, that is the weighted mean of the covariance matrix of the positive training samples. The mapping computation requires at least one SVD of an 8x8 matrix, and since such operation is computationally demanding, it is necessary to optimize the detection process.

To this aim, Tuzel *et al.*adopt a cascade of boosting classifiers and specifically, a set of LogitBoost classifiers based on logistic regressors in a rejection cascade manner. One of the advantages of such structure is computational: given the task of pedestrian detection on real world images and defined the set of windows (or bounding boxes) to test, only a small portion of them will run through the whole set of the LogitBoost classifiers; in fact, most of the patches are typically very dissimilar to the trained pedestrian model and will be rejected at the earlier stages of the cascade, reducing therefore the overall load of detection process.

Given a window w, defined by the 3-dimensional vector (w_x, w_y, w_s) (being respectively coordinates of the window center and window scale w.r.t. a given size; we assume constant aspect ratio), we introduce the *detection response* R as

$$R(w) = \frac{P(w)}{M} \tag{2}$$

where P is the index of the last cascade which provides a positive classification for w and M is the total number of cascades. Given the structure of rejection cascades, the higher the degree of response $R(w)$ is, the further w reached the end of the cascade, the more similar it is to the pedestrian model (up to the extreme of $R = 1$, that means successful classification). Tests over large sets of images in standard benchmarks show that the cascade of LogitBoost classifiers with covariance descriptors rejects most of the negative samples (80% of negative patches) within the first $\frac{1}{5}$ of the cascade (i.e. $R(w) < 0.2$ for 80% of generic negative patches).

Pedestrian detection at frame level is usually performed with a sliding window approach, i.e. a complete scanning of the "Sliding Windows Set" (SWS), that contains the windows at all possible window states (w_x, w_y, w_s). The cardinality of the SWS depends on the size of the image, on the range of scales to check and on the degree of coarseness for the scattering of the windows: regarding this latter parameter, to obtain a successful detection process, the SWS must be rich enough so that at least one window targets each pedestrian in the image. To be more precise, every classifier has a degree of sensitivity to small translations and

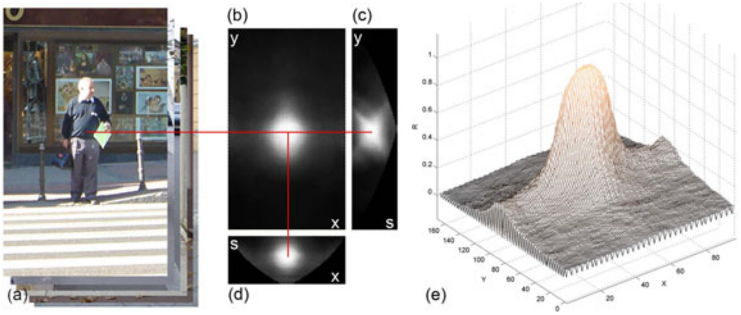


Fig. 1. Region of support for the cascade of LogitBoost classifiers trained on INRIA pedestrian dataset, averaged over a total 62 pedestrian patches; (a) a positive patch (pedestrian is 48x144); (b-d) response of the classifier: (b) fixed w_s (equal to 48x144), sliding w_x, w_y; (c) fixed w_x (equal to x of patch center), sliding w_s, w_y; (d) fixed w_y (equal to y of patch center), sliding w_x, w_s; (e) 3D plot of the response in (b).

scale variations, i.e. the response of the classifier in the close neighborhood (both in position and scale) of the window encompassing a pedestrian, remains positive ("region of support" of a positive detection). Having a sufficiently wide region of support allows to uniformly prune the SWS, up to the point of having at least one window targeting the region of support of each pedestrian in the frame. Vice versa, a too wide region of support could generate de-localized detections [4].

On this regard, an important advantage of the covariance descriptors is its relatively low degree of sensitivity to small translations and scale variations, i.e. its region of support over the positive detections was demonstrated to be higher with respect to many other descriptors (especially w.r.t. HoG). Its size depends on the training data, and the cascade of LogitBoost classifiers trained on the INRIA pedestrian dataset [3] shows a radius of such region of approximately 15% of the window size and 20% in the window scale (Fig. 1).

3 Multi-stage Sampling-Based Detection

The covariance-based pedestrian detector of [3] quantizes uniformly the state space with sliding windows, incurring in the two-fold problem of large waste in computational time searching over areas where pedestrians are not present and need of a redundant SWS to find every pedestrian in the scene.

Our objective is to provide a non-uniform quantization and to model the detection as an estimation of the states given the observations; we aim at estimating the modes of the continuous density function $p\left(\boldsymbol{X}|\boldsymbol{Z}\right)$, where $\boldsymbol{X} = (w_x, w_y, w_s)$ is the state and $\boldsymbol{Z}$ corresponds to the image. In section 3.1 we introduce an approximation of the likelihood function, progressively improved through a multi-stage sampling-based process. Such procedure has the advantage to provide a global view of the landscape of the likelihood function and, at the same time, to support efficient sample placement. The likelihood allows pedestrian detection within the single image. In section 3.2 we deal with pedestrian detection in videos, plugging the likelihood approximation method into a Bayesian-recursive filter.

3.1 Multi-stage Kernel-Based Density Estimation on a Single Image

Let's not consider any a prior information in the image (such as motion, geometry, depth, etc.) in order to provide a general solution. Consequently, the state pdf can be assumed proportional to the measurement likelihood function, i.e. $p(\boldsymbol{X}|\boldsymbol{Z}) \propto p(\boldsymbol{Z}|\boldsymbol{X})$.

The measurement likelihood function is estimated by iteratively refining it through m stages based on the observations. Algorithm 1 shows the complete procedure. The initial distribution $q_0(\boldsymbol{X})$ is set to a uniform distribution on the state space and it is sampled, extracting the first S_1 set of N_1 samples (see line 1 of Algorithm 1 and yellow points in the exemplar image of Fig. 2). Each sample s represents a state (w_x, w_y, w_s) in the domain of the windows. Scattering samples according to a uniform distribution is somehow similar to the sliding window strategy, though the samples are not equally distributed and their locations are not deterministically defined: indeed, the N_1 samples could also be grid-distributed without affecting the bottom line of the proposed method; instead, the key point here is N_1 be significantly lower than the cardinality of a typical SWS (see experiments in Section 4). The rationale is that part of these samples will fall in the basin of attraction of each region of support of the pedestrians in the image and will provide an initial rough estimation of the measurement function. Being driven by the previous measurements, at any stage i, the distribution q_i is progressively refined, to perform new sampling. This growing confidence over the proposal makes it possible to decrease, from stage to stage, the number of N_i to sample (see Fig. 2), differently from [13], where N_i is constant over stages.

The N_1 samples drawn from $q_0(\boldsymbol{X})$ (line 6) will be used to provide a first approximation of the measurement density function p_1, through a Kernel Density Estimation (KDE) approach with Gaussian kernel, generating a mixture of N_1 Gaussians: for each j-th component, mean, covariance and weight are defined as follows: the mean $\mu_i^{(j)}$ is set to the sample value $s_i^{(j)} = \left(w_{x,i}^{(j)}, w_{y,i}^{(j)}, w_{s,i}^{(j)}\right)$; the covariance matrix $\Sigma_i^{(j)}$ is set to a covariance Σ_i (line 8), which, at any given stage i, is constant for all samples. The work in [13] proposed to determine the Σ for each sample as a function of its k-nearest neighbors; this strategy yielded fairly unstable covariance estimations when applied to our context: indeed, given the low number of samples used in our method, k is to be kept pretty low (to maintain a significance over the covariance estimation), and this makes the estimation quite dependent on the specific randomized sample extraction. We preferred to assign an initial Σ_1 proportional to the size of the region of support of the classifier, and decrease the Σ_i of the following stages: this has the effect of incrementally narrowing the samples scattering, obtaining a more and more focused search over the state space.

Algorithm 1. Measurement Step

```
1: Set $q_0(\boldsymbol{X}) = U(\boldsymbol{X})$
2: Set $S = \emptyset$
3: for $i = 1$ to $m$ do
4:     begin
5:     Draw $N_i$ samples from $q_{i-1}(\boldsymbol{X})$:
6:         $S_i = \left\{ s_i^{(j)} | s_i^{(j)} \sim q_{i-1}(\boldsymbol{X}),\ \ j = 1, \ldots, N_i \right\}$
7:     Assign a Gaussian kernel to each sample:
8:         $\mu_i^{(j)} = s_i^{(j)} \quad ; \quad \Sigma_i^{(j)} = \Sigma_i$
9:     Compute the measurement on each sample $s_i^{(j)}$:
10:        $l_i^{(j)} = R^{\lambda_i}\left(\mu_i^{(j)}\right)$ with $R^{\lambda_i} \in [0, 1]$
11:    Obtain the measurement density function at step $i$:
12:        $p_i(\boldsymbol{Z}|\boldsymbol{X}) = \sum_{\pi_i^{(j)} \neq 0} \pi_i^{(j)} \cdot \mathcal{N}\left(\mu_i^{(j)}, \Sigma_i^{(j)}\right)$
13:    where: $\pi_i^{(j)} = \frac{l_i^{(j)}}{\sum_{k=1}^{N_i} l_i^{(k)}}$
14:    Compute the new proposal distribution:
15:        $q_i(\boldsymbol{X}) = (1 - \alpha_i) q_{i-1}(\boldsymbol{X}) + \alpha_i \frac{p_i(\boldsymbol{Z}|\boldsymbol{X})}{\int p_i(\boldsymbol{Z}|\boldsymbol{X}) d\boldsymbol{X}}$
16:    Retain only the samples with measurement value 1:
17:        $\widetilde{S}_i = \left\{ s_i^{(j)} \in S_i | R\left(\mu_i^{(j)}\right) = 1,\ \ j = 1, \ldots, N_i \right\}$
18:        $S = S \bigcup \widetilde{S}_i$
19:    end
20: Run variable-bandwidth meanshift (Non-Maximal-Suppression) over $S$. Obtain the
    set of modes $\mathcal{M}_1$
21: Prune the modes in $\mathcal{M}_1$ that do not represent reliable detection (see text). Obtain
    the new set of modes $\mathcal{M}_2$
22: Assign a Gaussian Kernel to each modes $\omega^{(j)} \in \mathcal{M}_2$ and compute the final likeli-
    hood function:
23:        $p(\boldsymbol{Z}|\boldsymbol{X}) \propto \sum_{\forall \omega^{(j)} \in \mathcal{M}_2} \mathcal{N}\left(\omega^{(j)}, \overline{\Sigma}\right)$
```

Finally, the response R of the classifier (eq. 2) is exploited, in a novel way, to determine the weight $\pi_i^{(j)}$ of the j-th component. The intention is that those samples falling close to the center of any region of support (i.e., close to the mode/peak of the distribution) might receive higher weight with respect to the others, so that the proposal distribution q_i, that is partly determined by p_i, will drive the sampling of the next stage more toward portion of the state space where the classifier yielded high responses. Conversely, sampling must not be wasted over areas with low response of the classifier. In other words, these weights must act as attractors which guide the samples toward the peaks. This is accomplished by connecting the weights $\pi_i^{(j)}$ to the response R of the pedestrian detector in the sample location $\mu_i^{(j)}$ (line 10).

The exponent λ_i used to compute the measurement is positive and increases at every stage: at early stages, $\lambda_i \in (0; 1)$, therefore the response of the samples

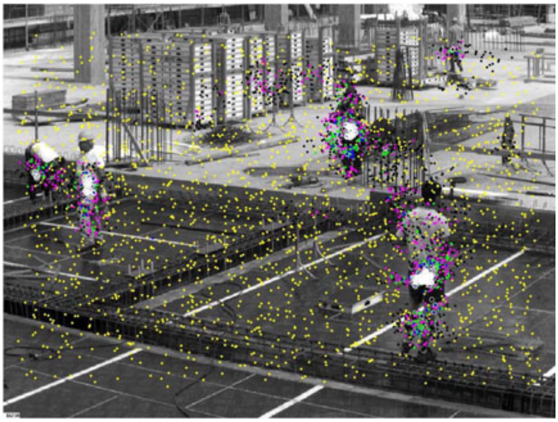


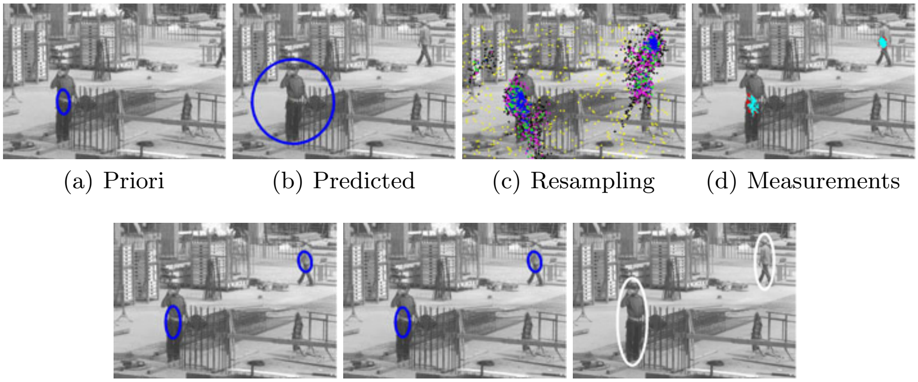
Fig. 2. Distribution of samples across the stages: $m = 5$ and $(2000, 1288, 829, 534, 349) = 5000$ samples. Stage order is yellow, black, magenta, green and blue. White circles represent the samples triggering a successful pedestrian classification.

is quite flattened, in order to treat fairly equally all range of not null responses; at later stages λ_i grows beyond 1, so that only the best responses will be held in account, while the others will be nullified. This behavior is clearly shown in Fig. 2 where the samples at subsequent stages (even if less numerous) are concentrated in the peaks of the distribution (i.e. where the response of the pedestrian detector is higher).

The Gaussian mixture of line 12 in Algorithm 1 is used as a partial estimation $p_i(\boldsymbol{Z}|\boldsymbol{X})$ of the likelihood function. This estimation is linearly combined with the previous proposal distribution $q_{i-1}(\boldsymbol{X})$ to obtain the new proposal distribution (line 15), where α_i is called *adaptation rate.*

The process is iterated for m stages and at the end of each stage only the samples of S^i that triggered a successful human detection (i.e. $R = 1$) are retained (line 17) and added to the final set of samples S (line 18). The samples retained in S are shown with white circles in Fig. 2. The number m of iterations can be fixed or adjusted according with a suitable convergence measure.

The non-maximal suppression is accomplished using a *variable-bandwidth mean-shift* suited to work on Gaussian mixtures [13], that provides a mixture of Gaussians representing in a compact way the final modes of the distribution. All those modes that contain less than τ_1 detections, or that contain less than $\tau_1/2$ strong detections are suppressed. Given the classification confidences provided by each LogitBoost classifiers of the cascade, a detection is considered strong if the minimum confidence is higher than a threshold τ_2. Numerical values are given in Sec. 4. The survived modes are considered successful detections and the derived mixture corresponds to the final likelihood function $p(\boldsymbol{Z}|\boldsymbol{X})$ (line 23).



(a) Priori (b) Predicted (c) Resampling (d) Measurements

(e) Likelihood (f) Posterior (g) Detections

Fig. 3. Multi-stage sampling in the context of Bayesian recursive filtering. In (c) the yellow dots represents the quasi-random sampling. The coloring is consistent with Fig. 2. The man on the upper-right corner is out of the influence of the predicted pdf, but the uniform component of eq. 5 allows some samples to fall within the region of support of that person and to act as attractors for the samples in the next stages. In (d), red dots represents successful detections, cyan dots are successful detections with high detection confidence.

3.2 Kernel-Based Bayesian Filtering on Videos

We extend here the previous method to the context of videos, by propagating the modes in a Bayesian-recursive filter. Differently from tracking approaches, the conditional density among frames (observations in time) is not used here to solve *data association*. Instead, the recursive nature of particle filtering exploits temporal coherence of pedestrians only to further improve detection. In the sequential Bayesian filtering framework, the conditional density of the state variable given the measurements is propagated through prediction and update stages as:

$$p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}\right) = \int p\left(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}\right) p\left(\boldsymbol{X}_{t-1}|\boldsymbol{Z}_{1:t-1}\right) d\boldsymbol{X}_{t-1} \tag{3}$$

$$p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t}\right) = \frac{p\left(\boldsymbol{Z}_t|\boldsymbol{X}_t\right) p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}\right)}{\int p\left(\boldsymbol{Z}_t|\boldsymbol{X}_t\right) p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}\right) d\boldsymbol{X}_t} \tag{4}$$

The priori $p\left(\boldsymbol{X}_{t-1}|\boldsymbol{Z}_{1:t-1}\right)$ is propagated from the posteriori at the previous frame and for the first frame $p\left(\boldsymbol{X}_0|\boldsymbol{Z}_0\right)$ no prior assumptions are made and uniform distribution is employed. The predicted pdf is obtained (eq. 3) as the product of the priori with the motion model and then marginalizing on $\boldsymbol{X}_{t-1}$. Since in complex scenes correct motion model is unknow [25], we applied a zero-order function with Gaussian noise of fixed covariance.

Fig. 3 depicts the different steps of this procedure. The priori is convolved with white noise which has the only effect of increasing its covariance (producing the

Table 1. Benchmark

		# images	**img size**	**# peds**	**peds size**	**avg peds/img**
Tests on Images	INRIA [2]	288	333x531-1280x960	582	80-800px	2.02
	Graz02 [27]	310	640x480	620	55-410px	2.00
Tests on CWS Videos	Video 1	148	800x600 @ 1 fps	340	55-350px	2.30
	Video 2	114		398		3.49
	Video 3	68		83		1.22

predicted pdf - Fig. 3(b)). Differently from the case of single images, where q_0 is uniform, in videos, at each time t (i.e. frame), $q_0(\boldsymbol{X}_t)$ is obtained by applying a *quasi-random sampling* [26] to the predicted distribution p:

$$q_0(\boldsymbol{X}_t) = \beta \cdot p(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}) + (1-\beta) \cdot U(\boldsymbol{X}_t) \tag{5}$$

where β decides the amount of random sampling. The random sampling is crucial to detect new pedestrians entering the scene (Fig. 3(c)). Given q_0, the procedure described in the previous section is used to iteratively estimate the likelihood $p(\boldsymbol{Z}_t|\boldsymbol{X}_t)$ (Fig. 3(e)). Any newly detected likelihood mode is confirmed as a new-entry pedestrian detection. The quasi-random sampling is applied only to the proposal distribution q_0 (the proposal of the first stage of the multi-stage sampling). The likelihood and the predicted are multiplied to obtain (unless a normalization factor) the posterior pdf (see eq. 4).

4 Experimental Results

We performed extensive experimentation of the proposed multi-stage boosting method (*MSBoost* or MSB hereinafter) both on images and videos with fairly high resolution (rarely less than 640x480, up to 1280x960): in these conditions the sliding window (SW) can be very demanding, and the benefit of MSBoost is highlighted. Additionally, we are also considering a large range of scales since the considered images contain people of quite diverse sizes. Finally, tests on videos were carried out considering no other information than appearance (neither motion nor scene geometry). Experimental results are obtained on the benchmark reported in Table 1. In order to compare with the state of the art we used publicly available datasets which also provide ground-truth annotations. In the case of images, we have used the Graz02 dataset [27] and the well-known INRIA dataset [2]. Regarding the videos, we used 3 video clips taken from construction working sites (CWS), that contain on average 19 entrances/exits per video.

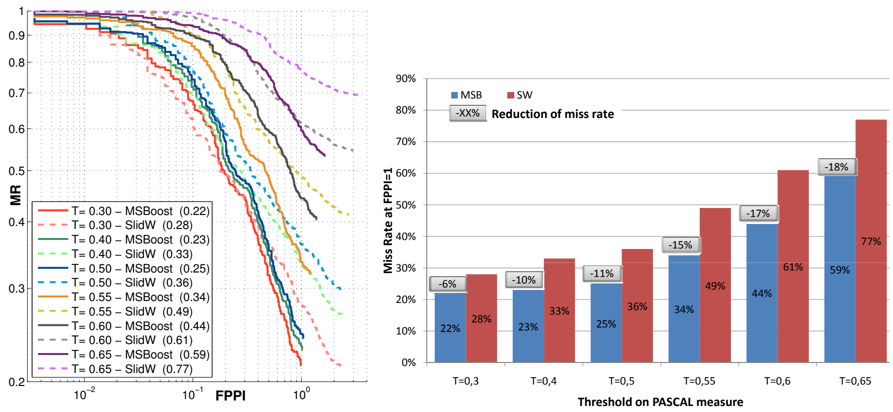
The accuracy of pedestrian detection is measured at object level in terms of the matching of the bounding box found by the detector (BB_{dt}) with the bounding box in the ground truth (BB_{gt}). A matching is found using the measure defined in the PASCAL object detection challenges [28] which states that the ratio between the area of overlap of BB_{dt} with BB_{gt} and the area of merge of the two BBs must be greater than a given threshold T, that is typically set to

50%; however, in some experiments we test the detection at lower and higher values of T, in order to better evaluate the localization accuracy of the detection of MSBoost w.r.t. SW. Throughout all tests, multiple detections of the same ground-truthed person, as well as a single detection matching multiple ground-truthed people, are affecting the performance in terms of recall and precision.

Regarding our approach, most of the tests has been performed using a total number of 5000 particles, divided over the $m = 5$ stages as follows: $N_i = NP \cdot e^{\gamma \cdot (i-1)}$, where $NP = 2000$ represents the initial number of particles (i.e., N_1), whereas γ is a constant factor (equal to 0.44 in our tests) which ensures that the number of particles diminishes over the stages in an exponential way. A similar approach is followed also for λ_i and Σ_i, which are the exponent for the measurement and the covariance for the Gaussian kernels, respectively. The starting values are 0.1 and $diag(7, 14, 0.125)$ (obtained considering the region of support, and with normalized scales) and the exponential constant are 1 and -0.66, respectively. Finally, the thresholds for the non-maximal suppression (see end of Section 3.1) have been set to $\tau_1 = 4.0$ and $\tau_2 = 4.0$. The first test on single images (INRIA dataset) aims at showing that MSBoost yields higher accuracy than SW in the detection localization; we measure detection performances varying the threshold T of the PASCAL challenge: the higher is T, the higher is the precision in detection accuracy required on the detector. In these tests MSBoost employs 15000 particles, while the SW uses a fixed position stride (10.9% of window size), employing on average 101400 (6.8 times more) windows per image, with peaks of 364000 (24.2 times more). Results are shown in Fig. 4. Fig. 4(a) highlights the trend of MR vs FPPI at different T: as expected, regardless of the detection paradigm, the higher is T, the higher is the MR. However, at any T, MSBoost shows lower MR than sliding window: moreover, as shown in Fig. 4(b), at increasing T, MSBoost decreases its performance in a lower degree w.r.t. SW; in other words, the detection localization of the former is higher and is achieved through the information gain obtained through the multi-stage sampling.

In the second test on single images (Graz02) we compared MSBoost vs SW at different number of windows. The scale stride is set to 1.2, the number of particles employed in MSBoost is 5000, while the number of windows in SW is 5000, 10000, 15000, 20000, 30000 and 50000 (corresponding respectively to a position stride of 15.6%, 10.9%, 9.8%, 8.2%, 7.1% and 5% of window size). The non-maximal suppression for SW is performed with mean shift and $\tau_2 = 2$. Tab. 2(a) shows the results achieved in terms of *False Positives Per Image* (FPPI) and *Miss Rate* (MR) as suggested in [29]. MSBoost with 5000 particles achieves a FPPI comparable to SW with 15000 windows, yielding an 8% lower MR.

Regarding the experiments on videos, we firstly aim at validating the usefulness of the Bayesian-recursive approach; we compared the FPPI and MR obtained on Video 1, by using the SW with 10000 windows, a non-recursive approach (Section 3.1 on each single frame) with 2500 particles and the Bayesian-recursive (Section 3.2) with varying number of particles (5000, 2500 and 1250); see Tab. 2(b). The Bayesian-recursive approach with 1250 particles yields similar FPPI and better MR w.r.t. non-recursive with 2500 particles. Moreover, it



(a) (b)

Fig. 4. Results on INRIA dataset at different values of T, the threshold on the bounding box matching. Number in brackets represent the MR at FPPI=1.

Table 2. Summary of results

(a) On Graz02 dataset

	FPPI	MR
SW (5000)	0.39	0.76
SW (10000)	0.66	0.57
SW (15000)	**0.73**	**0.51**
SW (20000)	1.08	0.46
SW (30000)	1.30	0.40
SW (50000)	1.66	0.37
MSB (5000)	**0.74**	**0.43**

(b) On Video 1

	FPPI	MR
SW (10000)	0.54	0.54
MSB (2500)	0.29	0.34
MSB rec (1250)	0.30	0.29
MSB rec (2500)	0.45	0.14
MSB rec (5000)	0.56	0.13
MSB rec (5000, no-exp decay)	0.56	0.16

(c) On Videos 1,2,3

	FPPI	MR
Video 1	0.56	0.13
Video 2	0.98	0.55
Video 3	0.42	0.78

obtains overall better performance than SW with 10000 windows. Eventually, we also evaluated the usefulness of the exponential decay of particles, as disabling it slightly reduces the performances. Then, to further validate our approach we tested Bayesian-recursive MSBoost on two other videos from the CWS dataset which contain several heavy occlusions of the pedestrians (see Table 1). Results are summarized in Tab. 2(c).

Regarding the computational load, when dealing with cascades of strong classifiers, the time cannot be considered simply proportional to the number of employed windows or samples. In fact, in traditional classifiers the classification time for each window is constant while in cascaded classifiers the time is reduced if the input is rejected at an intermediate level of the cascade. In this sense, the information gain across the multiple stages of the MSBoost produces samples that are increasingly closer to the positive classification and therefore the average number of strong classifiers that are successfully passed increases, raising the computation time: this is testified by the average R of MSBoost that is higher than the one of SW (0.26 and 0.08 respectively). Nevertheless, MSBoost ends up being definitely faster because: (a) the time to prepare the integral image and the tensors for each patch (that is a fixed time for each window, regardless of its appearance or of the use of MSBoost or SW), is on average 8.92 times the average classification time of a random negative patch; (b) merging overhead and classification time, the per-particle computational load of MSBoost is 1.9 times higher than the per-window computational load of SW; (c) the experimental results demonstrate that MSBoost achieves higher detection accuracy with a number of particles that is from 3 to 10 times lower than the number of windows employed with SW. Thus, the measured computation time for MSBoost is from 1.8 to 5.4 times lower than for SW. This increase of performance is almost independent on the number of objects to be detected. On average MSBoost takes about 1 second to perform 5000 detections using a C++ implementation on a dual-core off-the-shelf PC, also by exploiting the intrinsic parallelization of the algorithm. The complete approach can process about 0.75 frames per second (fps) with 5000 particles, which can be proportionally increased by reducing the number of particles (e.g., it becomes about 3 fps with 1250 particles which give good results on Video 1 of CWS).

5 Conclusions

The work introduces a novel method to avoid the brute force strategy of sliding window for (pedestrian) detection in both images and videos; the proposed method works within the domain of appearance used by the classifier itself, exploiting the response of the boosting cascade to drive an efficient spanning of the state space and using a multi-stage sampling based strategy. The derived measurement function can be plugged in a kernel-based Bayesian filtering to exploit temporal coherence of pedestrian in videos. Experimental results show a gain in computational load maintaining same accuracy of sliding window approach.

References

1. Viola, P.A., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. IJCV 63, 153–161 (2005)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
3. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE T-PAMI 30, 1713–1727 (2008)
4. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE T-PAMI 31 (2009)
5. Tao, J., Odobez, J.M.: Fast human detection from videos using covariance features. In: Workshop on VS at ECCV (2008)
6. Ess, A., Leibe, B., Schindler, K., van Gool, L.: Robust multiperson tracking from a mobile platform. IEEE T-PAMI 31, 1831–1846 (2009)
7. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. IJCV 80, 3–15 (2008)

8. Wojek, C., Dorkó, G., Schulz, A., Schiele, B.: Sliding-windows for rapid object class localization: A parallel technique. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 71–81. Springer, Heidelberg (2008)
9. Lehmann, A., Leibe, B., Van Gool, L.: Feature-centric efficient subwindow search. In: ICCV (2009)
10. Butko, N., Movellan, J.: Optimal scanning for faster object detection. In: IEEE Conference on CVPR 2009, pp. 2751–2758 (2009)
11. Zhang, W., Zelinsky, G., Samaras, D.: Real-time accurate object detection using multiple resolutions. In: IEEE Conference on ICCV 2007, pp. 1–8 (2007)
12. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE T-PAMI 25, 564–575 (2003)
13. Han, B., Zhu, Y., Comaniciu, D., Davis, L.S.: Visual tracking by continuous density propagation in sequential bayesian filtering framework. IEEE T-PAMI 31 (2009)
14. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. IJCV 29, 5–28 (1998)
15. Hue, C., Le Cadre, J.P., Perez, P.: Tracking multiple objects with particle filtering. IEEE Transactions on Aerospace and Electronic Systems 38, 791–812 (2002)
16. Isard, M., MacCormick, J.: Bramble: A bayesian multiple-blob tracker. In: ICCV, pp. 34–41 (2001)
17. Okuma, K., Taleghani, A., de Freitas, N., 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)
18. Vermaak, J., Doucet, A., Pérez, P.: Maintaining multi-modality through mixture tracking. In: ICCV, pp. 1110–1116 (2003)
19. Papageorgiou, C., Poggio, T.: A trainable system for object detection. IJCV 38, 15–33 (2000)
20. 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)
21. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: CVPR, pp. 1–8 (2007)
22. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR, pp. 1–8 (2008)
23. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Annals of Statistics 28, 337–407 (2000)
24. Babenko, B., Dollár, P., Tu, Z., Belongie, S.: Simultaneous learning and alignment: Multi-instance and multi-pose learning. In: Faces in Real-Life Images (2008)
25. Han, B., Comaniciu, D., Zhu, Y., Davis, L.: Incremental density approximation and kernel-based bayesian filtering for object tracking. In: CVPR (2004)
26. Philomin, V., Duraiswami, R., Davis, L.: Quasi-random sampling for condensation. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 134–149. Springer, Heidelberg (2000)
27. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. IEEE T-PAMI 28, 416–431 (2006)
28. Ponce, J., Berg, T., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, C., Torralba, A., Williams, C., Zhang, J., Zisserman, A.: In: Dataset issues in object recognition, pp. 29–48. Springer, Heidelberg (2006)
29. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR, pp. 304–311 (2009)

Learning to Detect Roads in High-Resolution Aerial Images

Volodymyr Mnih and Geoffrey E. Hinton

Department of Computer Science, University of Toronto,
6 King's College Rd., Toronto, Ontario,
M5S 3G4, Canada
{vmnih,hinton}@cs.toronto.edu

Abstract. Reliably extracting information from aerial imagery is a difficult problem with many practical applications. One specific case of this problem is the task of automatically detecting roads. This task is a difficult vision problem because of occlusions, shadows, and a wide variety of non-road objects. Despite 30 years of work on automatic road detection, no automatic or semi-automatic road detection system is currently on the market and no published method has been shown to work reliably on large datasets of urban imagery. We propose detecting roads using a neural network with millions of trainable weights which looks at a much larger context than was used in previous attempts at learning the task. The network is trained on massive amounts of data using a consumer GPU. We demonstrate that predictive performance can be substantially improved by initializing the feature detectors using recently developed unsupervised learning methods as well as by taking advantage of the local spatial coherence of the output labels. We show that our method works reliably on two challenging urban datasets that are an order of magnitude larger than what was used to evaluate previous approaches.

1 Introduction

Having up-to-date road maps is crucial for providing many important services. For example, a city requires accurate road maps for routing emergency vehicles, while a GPS-based navigation system needs the same information in order to provide the best directions to its users. Since new roads are constructed frequently keeping road maps up-to-date is an important problem.

At present, road maps are constructed and updated by hand based on high-resolution aerial imagery. Since very large areas need to be considered, the updating process is costly and time consuming. For this reason automatic detection of roads in high-resolution aerial imagery has attracted a lot of attention in the remote sensing community. Nevertheless, despite over 30 years of effort [1], at the time of writing there was no commercial automatic or semi-automatic road detection system on the market [2,3] and, to the best of our knowledge, no published method has been shown to work reliably on large datasets of high-resolution urban imagery.

Much of the published work on automatic road detection follows an ad-hoc multi-stage approach [1,4,5]. This generally involves establishing some a priori criteria for the appearance of roads and engineering a system that detects objects that satisfy the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 210–223, 2010.

established criteria. For example, roads are often characterized as high-contrast regions with low curvature and constant width, with a typical detection strategy involving edge detection, followed by edge grouping and pruning. While some of these approaches have exhibited good performance on a few sample images, the way in which they combine multiple components often results in the need to tune multiple thresholds and such methods have not been shown to work on large real-world datasets.

In this paper we follow a different approach, where the system *learns* to detect roads from expert-labelled data. Learning approaches are particularly well-suited to the road detection task because it is a rare example of a problem where expert-labelled data is abundant. It is easy to obtain hundreds of square kilometers of high-resolution aerial images and aligned road maps. In fact, most universities have libraries dedicated solely to geographic data of this kind.

Learning-based approaches to road detection are not new – several attempts at predicting whether a given pixel is road or not road given features extracted from some context around it have been made [6,7,8,9]. While showing some promise, these approaches have also failed to scale up to large challenging datasets. We believe that previous learning-based approaches to road detection have not worked well because they suffer from three main problems. First, very little training data is used, likely because ground truth for training and testing is typically obtained by manually labelling each pixel of an aerial image as road or non-road making it infeasible to use a lot of training data. Second, either a very small context is used to extract the features, or only a few features are extracted from the context. Finally, predictions for each pixel are made independently, ignoring the strong dependencies between the road/non-road labels for nearby pixels.

We propose a large-scale learning approach to road detection that addresses all three problems as follows:

- We use synthetic road/non-road labels that we generate from readily available vector road maps. This allows us to generate much larger labelled datasets than the ones that have been used in the past.[1]
- By using neural networks implemented on a graphics processor as our predictors we are able to efficiently learn a large number of features and use a large context for making predictions.
- We introduce a post-processing procedure that uses the dependencies present in nearby map pixels to significantly improve the predictions of our neural network.

Our proposed approach is the first to be shown to work well on large amounts of such challenging data. In fact, we perform an evaluation on two challenging urban datasets covering an area that is an order of magnitude larger than what was used to evaluate any previous approach. We also show that a previous learning based approach works well on some parts of the datasets but very poorly on others. Finally, we show that all three of our proposed enhancements are important to obtaining good detection results.

[1] Dollar et al. [10] proposed a similar approach to generating ground truth data but still used very little training data.

2 Problem Formulation

Let S be a satellite/aerial image and let M be a corresponding road map image. We define $M(i, j)$ to be 1 whenever location (i, j) in the satellite image S corresponds to a road pixel and 0 otherwise. The goal of this paper is to learn $p(M(i, j)|S)$ from data.

In a high-resolution aerial image, a single pixel can represent a square patch of land that is anywhere between several meters and tens of centimeters wide. At the same time one is typically interested in detecting roads in a large area such as an entire town or city. Hence, one is generally faced with the problem of making predictions for millions if not billions of map pixels based on an equally large number of satellite image pixels. For these reasons, the probability that $M(i, j) = 1$ has typically been modeled as a function of some relatively small subset of S that contains location (i, j) instead of the entire image S [7,10]. In this paper we model

$$p(N(M(i, j), w_m)|N(S(i, j), w_s)), \tag{1}$$

where $N(I(i, j), w)$ denotes a $w \times w$ patch of image I centered at location (i, j). Hence, we learn to make predictions for a $w_m \times w_m$ map patch given a $w_s \times w_s$ satellite image patch centered at the same location, where $w_m < w_s$. This allows us to reduce the required computation by both limiting the context used to make the predictions and by reusing the computations performed to extract features from the context.

2.1 Data

While high-resolution aerial imagery is easy to obtain, per pixel road/non-road labels are generally not available because most road maps come in a vector format that only specifies the centreline of each road and provides no information about road widths. This means that in order to obtain per-pixel labels one must either label images by hand or generate approximate labels from vector data. The hand labelling approach results in the most accurate labels, but is tedious and expensive. In this paper we concentrate on using approximate labels.

Our procedure for generating per-pixel labels for a given satellite image S is as follows. We start with a vector road map consisting of road centreline locations for a region that includes the area depicted in S. We rasterize the road map to obtain a mask C for the satellite image S. In other words, $C(i, j)$ is 1 if location (i, j) in satellite image S belongs to a road centreline and 0 otherwise.

We then use the mask C to define the ground truth map M as

$$M(i, j) = e^{-\frac{d(i,j)^2}{\sigma^2}}, \tag{2}$$

where $d(i, j)$ is the Euclidean distance between location (i, j) and the nearest nonzero pixel in the mask C, and σ is a smoothing parameter that depends on the scale of the aerial images being used. $M(i, j)$ can be interpreted as the probability that location (i, j) belongs to a road given that it is $d(i, j)$ pixels away from the nearest centreline pixel. This soft weighting scheme accounts for uncertainty in road widths and centreline locations. In our experiment σ was set such that the distance equivalent to $2\sigma + 1$ pixels roughly corresponds to the width of a typical two-lane road.

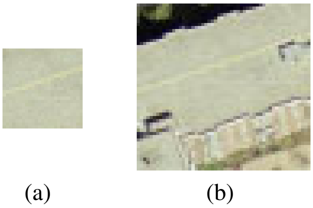

(a) (b)

Fig. 1. The rooftop of an apartment building. a) Without context. b) With context.

3 Learning to Detect Roads

Our goal is to learn a model of (1) from data. We use neural networks because of their ability to scale to massive amounts of data as well as the ease with which they can be implemented on parallel hardware such as a GPU. We model (1) as

$$f(\phi(N(S(i,j), w_s))), \tag{3}$$

where ϕ is feature extractor/pre-processor and f is a neural network with a single hidden layer and logistic sigmoid hidden and output units. To be precise,

$$f(\mathbf{x}) = \sigma(\mathbf{W}_2^T \sigma(\mathbf{W}_1^T \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2), \tag{4}$$

where $\sigma(\mathbf{x})$ is the elementwise logistic sigmoid function, $\mathbf{W}$'s are weight matrices and $\mathbf{b}$'s are bias vectors. We now describe the pre-processing function ϕ, followed by the training procedure for f.

3.1 Pre-processing

It has been pointed out that it is insufficient to use only local image intensity information for detecting roads [7]. We illustrate this point with Figure 1. The aerial image patch depicted in sub-figure 1(a) resembles a patch of road, but with more context, as shown in sub-figure 1(b), it is clearly the roof of an apartment building. Hence, it is important to incorporate as much context as possible into the inputs to the predictor.

The primary aim of the pre-processing procedure is to reduce the dimensionality of the input data in order to allow the use of a large context for making predictions. We apply Principal Component Analysis to $w_s \times w_s$ RGB aerial image patches and retain the top $w_s \cdot w_s$ principal components. The function ϕ is then defined as the projection of $w_s \times w_s$ RGB image patches onto the top $w_s \cdot w_s$ principal components. This transformation reduces the dimensionality of the data by two thirds while retaining most of the important structure. We have experimented with using alternative colour spaces, such as HSV, but did not find a substantial difference in performance.

It is possible to augment the input representation with other features, such as edge or texture features, but we do not do so in this paper. We have experimented with using edge information in addition to image intensity information, but this did not improve performance. This is likely due to our use of an unsupervised learning procedure for

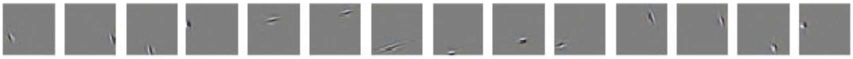


Fig. 2. Some of the filters learned by the unsupervised pretraining procedure

initializing, or pretraining, the neural network. In the next section we will describe how this procedure discovers edge features independently by learning a model of aerial image patches.

3.2 Training Procedure

At training time we are presented with N map and aerial image patch pairs. Let $\mathbf{m}^{(n)}$ and $\mathbf{s}^{(n)}$ be vectors representing the nth map and aerial image patches respectively, and let $\hat{\mathbf{m}}^{(n)}$ denote the predicted map patch for the nth training case. We train the neural network by minimizing the total cross entropy between ground truth and predicted map patches given by

$$-\sum_{n=1}^{N}\sum_{i=1}^{w_m^2}\left(m_i^{(n)}\log\hat{m}_i^{(n)}+(1-m_i^{(n)})\log(1-\hat{m}_i^{(n)})\right), \tag{5}$$

where we use subscripts to index vector components. We used stochastic gradient descent with momentum as the optimizer.

Unsupervised Pretraining. Traditionally neural networks have been initialized with small random weights. However, it has recently been shown that using an unsupervised learning procedure to initialize the weights can significantly improve the performance of neural networks [11,12]. Using such an initialization procedure has been referred to as *pretraining*.

We pretrain the neural network f using the procedure of Hinton and Salakhutdinov [11], which makes use of Restricted Boltzmann Machines (RBMs). An RBM is a type of undirected graphical model that defines a joint probability distribution over a vector of observed variables **v** and a vector of latent variables **h**. Since our neural network has real-valued inputs and logistic hidden units, in order to apply RBM-based pretraining, we use an RBM with Gaussian visible and binary hidden units. The joint probability distribution over **v** and **h** defined by an RBM with Gaussian visible and binary hidden units is

$$p(\mathbf{v},\mathbf{h})=e^{-E(\mathbf{v},\mathbf{h})}/Z,$$

where Z is a normalizing constant and the energy $E(\mathbf{v},\mathbf{h})$ is defined as

$$E(\mathbf{v},\mathbf{h})=\sum_i v_i^2-\left(\sum_i c_i v_i+\sum_k b_k h_k+\sum_{i,k} w_{ik} v_i h_k\right). \tag{6}$$

While maximum likelihood learning in RBMs is generally intractable, efficient approximate learning can be performed by approximately minimizing a different objective function known as Contrastive Divergence [13].

We train an RBM on the PCA representations of aerial image patches by approximately minimizing Contrastive Divergence using stochastic gradient descent with momentum. In order to encourage a sparse model of aerial images, i.e. one where only a few components of **h** are nonzero, we fix the hidden unit biases b_k to a large negative value[2], as proposed by Norouzi et al. [14]. This encourages the hidden units to be off unless they get a large input from the visible units. Once the RBM was trained, we initialized the weight matrix W_1 and bias vector b_1 from Equation 4 with the RBM weights w and b. We found that encouraging sparseness sped up learning and improved generalization.

Some selected filters learned by the pretraining procedure are shown in Figure 2. The vast majority of the filters learned to ignore colour, but the few filters that were colour sensitive were low-frequency, opposing red-green or blue-yellow filters. Many of the colour-neutral filters are oriented, high-frequency edge filters. We believe this is why augmenting the inputs with edge information did not improve road detection performance.

Adding Rotations. When training the neural network f we found that it is useful to rotate each training case by a random angle each time it is processed. Since many cities have large areas where the road network forms a grid, training on data without rotations will result in a model that is better at detecting roads at certain orientations. By randomly rotating the training cases the resulting models do not favor roads in any particular orientation.

4 Incorporating Structure

Figure 3(a) shows predictions for a small map patch made by our neural network. There are two obvious problems with these predictions – there are both gaps in the predicted roads and disconnected blotches of road pixels. Given our prior knowledge about the structure of road networks it would be safe to conclude that the blotches in Figure 3(a) are false positives while the gaps are false negatives. Previous learning-based approaches to road detection along with the method described in Section 3 make such mistakes because they make predictions independently for all pixels.

In order to take advantage of the structure present in nearby road/non-road labels we introduce a post-processing step. The goal is to improve the prediction for a given map pixel using nearby predictions. We treat this as a supervised learning problem and train a neural network to predict a $w_m \times w_m$ map patch from a $w_c \times w_c$ patch of predictions. To be precise, let $\hat{M}$ be the predictions of neural network f for map image M. Then let f_p be a neural network of the same functional form as f that predicts $N(M(i,j), w_m)$ based on $N(\hat{M}(i,j), w_c)$. The prediction of f_p for map image M is then denoted by $\hat{M}_p$.

The neural network f_p is trained using stochastic gradient descent to minimize cross entropy between the ground truth map patches and the predictions as given by Equation (5). We do not use pretraining when training f_p, as this did not improve performance.

[2] In this paper, we set b_k to -4.

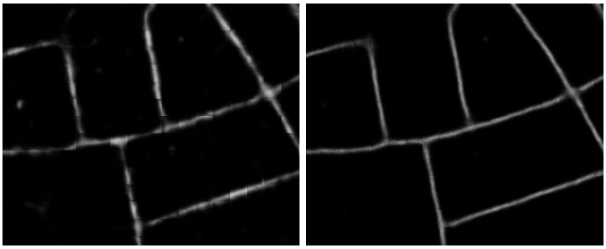


Fig. 3. (a) Predictions before post-processing. (b) Predictions after post-processing.

As with training of the neural network f, we randomly rotate each training case before it is processed in order to remove a bias towards roads in some orientations.

The post-processing procedure is similar to the approach employed by Jain and Seung [15] for natural image denoising. They train a convolutional neural network to predict small noise-free patches of natural images given larger patches that had noise added to them. Since our post-processing procedure repeatedly applies a local filter at fixed intervals over a larger image, it can be seen as a type of convolutional neural network where the convolution is followed by subsampling. Jain and Seung show that this kind of neural network architecture can be seen as performing approximate inference in a special kind of Markov Random Field model [15]. Jain and Seung also show that this approach outperforms approaches based on Markov Random Fields on the image denoising task.

Figure 3(b) shows the result of applying the post-processing procedure to the predictions from figure 3(a). The process clearly removes disconnected blotches, fills in the gaps in the roads, and generally improves the quality of the predictions. While we do not do so in this paper, the post-processing procedure can be applied repeatedly, with each application receiving the predictions made by the previous application as input. This process propagates confident predictions along the predicted road network.

5 Experiments

We performed experiments on two datasets consisting of urban aerial imagery at a resolution of 1.2 meters per pixel. We will refer to the datasets as URBAN1 and URBAN2. Dataset URBAN1 covers a large metropolitan area with both urban and suburban regions. It consist of a training set that covers roughly 500 square kilometers, a separate test set of 50 square kilometers, and a separate small validation set that was used for model selection. Dataset URBAN2 is only used for testing and consists of 28 square kilometers of aerial imagery of a city different from the one covered in URBAN1. When generating the ground truth pixel labels as described in Section 2.1, the smoothing parameters σ was set to 2 pixels. This makes the area within one standard deviation of a pixel roughly 20 feet in diameter, which is approximately the width of a typical two lane road.

We made predictions for 16×16 map patches from 64×64 colour RGB aerial image patches, which corresponds to $w_m = 16$ and $w_s = 64$. The neural network f had 4096 input units, 12288 hidden units, and 256 output units. For the post-processing procedure, we set w_c to 64 and used 4096 hidden units in the neural net f_p. Hence f_p had 4096 input units, 4096 hidden units, and 256 output units[3]. All inputs to the neural networks were shifted and rescaled to have mean 0 and standard deviation 1.

Although our method is not overly sensitive to the parameter values, we present them here for completeness. We used stochastic gradient descent with minibatches of size 64 and momentum of 0.9 for training the neural networks. We used a learning rate of 0.0005 and L_2 weight decay of 0.0002. When training Restricted Boltzmann Machines we used the contrastive divergence approximation to the gradient [13]. Once again, we used stochastic gradient descent with minibatches of size 64 and momentum of 0.9. We used a learning rate of 0.001 and L_2 weight decay of 0.0002. We made between 10 and 20 passes through the training set when training the neural networks and RBMs.

Since the models we have just described all have millions of parameters and the training set for dataset URBAN1 consists of over 1.2 million training cases, training our models would normally take months on a single core CPU or weeks on a multi-core machine. We were able to train our best model in less than 3 days on a consumer GPU. This included pretraining and training of neural network f and training of the post-processing neural network f_p. Since the training procedures for neural networks and RBMs are easily expressed in terms of elementary matrix operations, porting them to the GPU was trivial. In both cases, we obtained speedups of more than an order of magnitude over the same algorithms running on a modern four-core CPU[4]. In order to implement the required algorithms on the GPU, we first created a GPU-based matrix library for Python. The CUDAMat library as well as our implementations of neural networks and RBMs are now available as open-source software [16].

5.1 Metrics

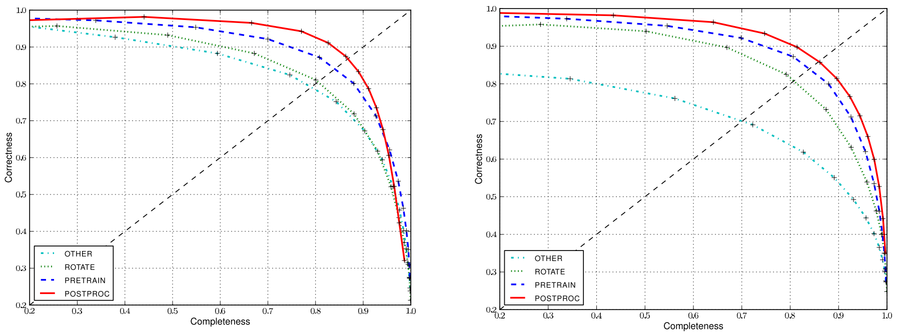
The most common metrics for evaluating road detection systems are correctness and completeness [17]. The *completeness* of a set of predictions is the fraction of true roads that were correctly detected, while the *correctness* is the fraction of predicted roads that are true roads. Since the road centreline locations that we used to generate ground truth are often noisy we compute relaxed completeness and correctness scores. Namely, in our experiments completeness represents the fraction of true road pixels that are within ρ pixels of a predicted road pixel, while correctness measures the fraction of predicted road pixels that are within ρ pixels of a true road pixel. Relaxing the completeness and correctness measures in this manner is common practice when evaluating road detection systems [17]. In this paper we set ρ to 3 pixels.

5.2 Results

Since our models provide us with road/non-road probabilities for map pixels, we need to select a threshold to make concrete predictions. For this reason we evaluate our models

[3] Multiples of 64 were used because using arrays with dimensions that are multiples of 64 can help reduce the number of idle cores on the GPU.

[4] CPU implementations used parallel linear algebra routines and MATLAB.



(a) Results for URBAN1

(b) Results for URBAN2

Fig. 4. Completeness/correctness curves on URBAN1 and URBAN2

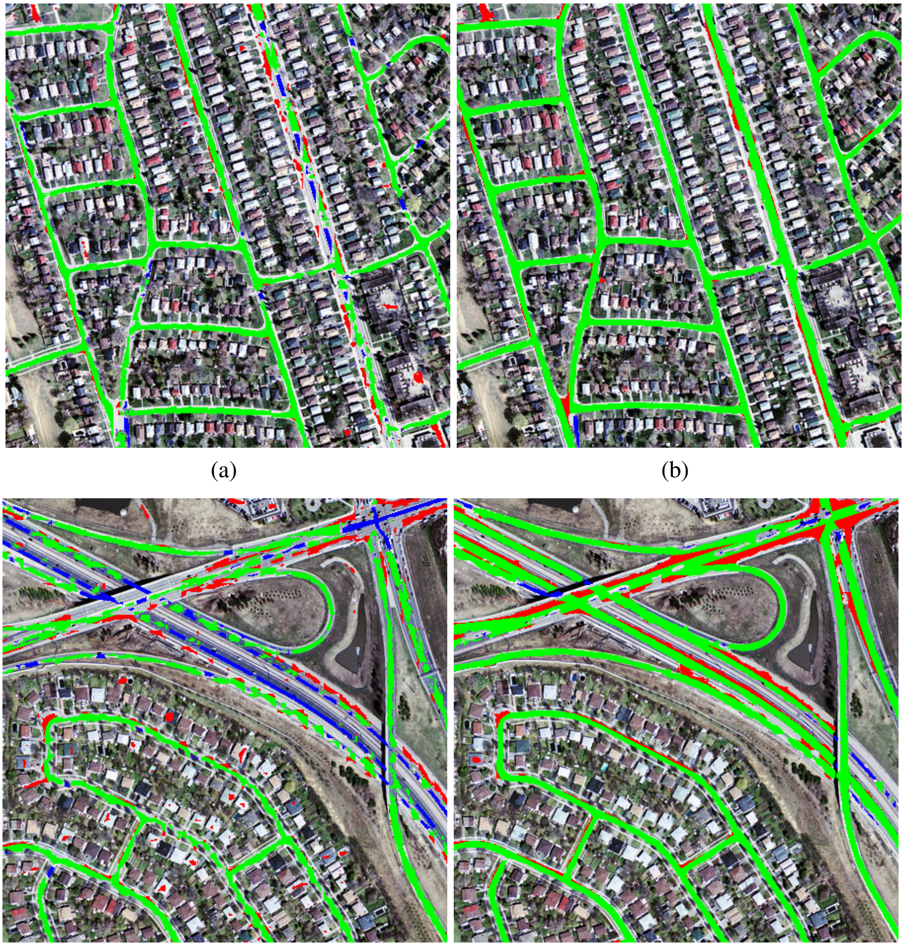
using completeness/correctness curves. Figure 4 shows completeness/correctness curves for the four models we evaluated on both datasets.

To compare to previous approaches, we evaluate a model, labelled OTHER, that uses a smaller context of size 24 and does not use rotated training data, pretraining, or post-processing. This approach has been used in several road detection systems [6,7,9], but with far less training data. The model OTHER is also an example of the kind of road detection system that can be trained on a modern CPU in the time it takes us to train our best model on a GPU.

We compare OTHER to three new models that used a context size of 64 and were trained as described above. The model ROTATE did not utilize pretraining or post-processing and is meant to show the performance of using a large context with rotated training data. The model PRETRAIN is a pretrained version of ROTATE. Finally, the model POSTPROC is the model PRETRAIN followed by our post-processing procedure.

The large difference in the performance of the model OTHER on the two datasets can be explained by the structure of their road networks. Many cities have large areas where the road network consists of a grid at some orientation, resulting in roads having two dominant orientations. Indeed, large parts of the cities in URBAN1 and URBAN2 consist of grids, however, the orientation of the grids is different between the two datasets. Since the model OTHER is trained on patches of URBAN1 without randomly rotating them, the model strongly favors roads in orientations similar to those in URBAN1. Since the dominant orientations of roads in URBAN2 are different, the performance of OTHER on URBAN2 is much worse than on URBAN1. This gap in performance shows that any approach that learns to detect roads from patches without incorporating rotations into the data or rotation invariance into the model is likely to work very poorly unless it is trained and tested on very similar conditions. This effect also highlights the importance of evaluating road detection systems on large datasets with a wide variety of road types and orientations.

Since the remaining three models randomly rotate each training case before processing it, our models exhibit similar performance on URBAN1 and URBAN2, suggesting that they are robust to significant variations between training and testing data. The



(a)

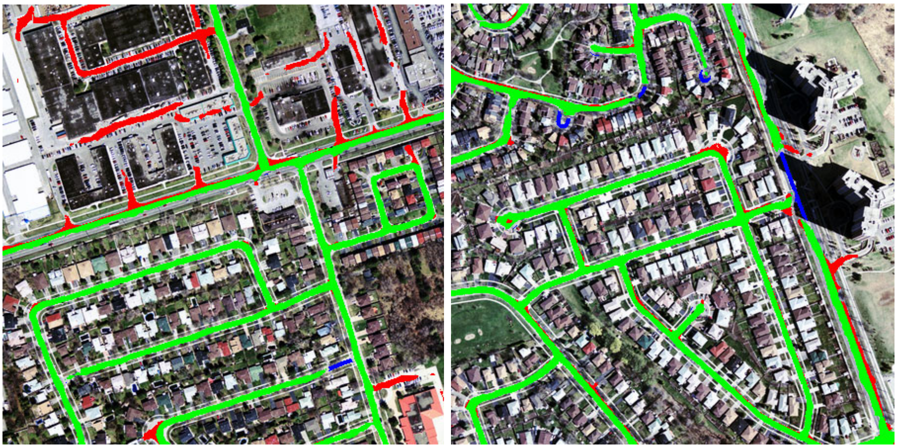
(b)

(c)

(d)

Fig. 5. a) and c) Visualization of the predictions made by OTHER. b) and d) Visualizations of the predictions made by POSTPROC. See the electronic version for colour. True positives are shown in green, false positives are shown in red, false negatives are shown in blue, and the background colour is used for true negatives. We used the threshold that corresponds to the break-even point on the completeness/correctness curves.

results also show that unsupervised pretraining significantly improves road detection performance. If we compare the models by their break-even points, i.e. the points on the curves where completeness equals correctness, then unsupervised pretraining improves both completeness and correctness by about 0.05 on both datasets. The post-processing procedure further improves completeness and correctness on both datasets by approximately another 0.02.



(a) (b)

Fig. 6. Failure modes of the model POSTPROC. See the electronic version for colour.

Figure 5 presents a qualitative comparison between the typical predictions of the models OTHER and POSTPROC on the URBAN1 test set. Figure 5(a) shows that while OTHER is able to detect two-lane suburban roads quite well, the model often has problems with bigger roads. Figure 5(b) shows that the model POSTPROC is able to deal with wider roads. Figures 5(c) and 5(d) show the predictions of OTHER and POSTPROC respectively for an area that includes a highway interchange. The model OTHER clearly has trouble detecting the highway while POSTPROC does not.

To get a better understanding of the kinds mistakes our best model makes, POSTPROC consider Figure 6. It shows predictions made by the POSTPROC model on two regions taken from the URBAN1 test set. Figure 6(a) shows some typical examples of false positive detections. Most of the false positives are in fact paved regions that cars drive on. Since only named streets tend to be included in road maps, things like alleys and parking lots are not included and hence end up being labelled as false positives, if detected.

Figure 6(b) shows some examples of typical false negative detections, which tend to be caused by rare road types or conditions. For example, while our model is able to deal with shadows and occlusions caused by small objects, such as trees, it is unable to deal with shadows and occlusions caused by large buildings. One possible way of dealing with such problems is modifying the post-processing procedure to receive predictions as well as a satellite image patch of the same area as input. This should allow the post-processor to learn to fill in such gaps based on appearance.

We stress that our evaluation was performed on challenging urban data and covered an area roughly an order of magnitude larger than the areas used to evaluate previous work on road detection. We believe that our approach is the first to be shown to work reliably on real-world data on a large scale.

6 Related Work

Most of the prior work on road detection, starting with the initial work of Bajcsy and Tavakoli [1], follows an ad-hoc approach. A popular approach involves first extracting edges or other primitives and then applying grouping and pruning techniques to obtain the final road network. Laptev et al. [5] use scale space theory to extract a coarse road network and then apply a ribbon snake model to refine the road network, while Mena and Malpica [18] use segmentation followed by skeleton extraction. Another common strategy involves tracking roads from either expert-provided or automatically extracted starting points [19,4].

One of the earliest attempts to learn to detect roads in aerial imagery is due to Boggess [7]. A neural network was used to predict road/non-road labels for a pixel given a small (5×5 pixels) aerial image context. Not surprisingly such a small context is not sufficient for detecting roads in a wide variety of settings. Subsequent attempts to use neural networks for road detection [6,9] did not achieve significant improvements over the results of Boggess as they also relied on a small context (9×9 pixels being the largest) for prediction and used very little training data.

Dollar et al. [10] presented some results on road detection for their general approach to learning object boundaries. They extract tens of thousands of predefined features (such as Haar filter responses) from a large context around each pixel and use a probabilistic boosting tree to make predictions. However, they only offer a proof-of-concept qualitative evaluation on three small images. While our approach shares many of the same characteristics, the key difference is that we learn the features and exploit the dependencies among the labels.

There is a vast literature on methods for exploiting dependencies among pixel labels to which our post-processing procedure is related. He et al. [20] applied Conditional Random Fields (CRFs) to the image labelling problem after extending them to the image domain. In the road detection literature, active contour models are often used to incorporate prior knowledge about the structure of road networks for improved detection results [5,21]. Porway et al. [22] used a grammar to model relationships between objects such as cars, trees, and roofs for the purpose of parsing aerial images. As we have already mentioned, our post-processing step is similar to the approach of Jain and Seung [15] to image denoising. One advantage of this type of approach over using MRFs and CRFs with unrestricted potentials is that it avoids the need for performing approximate inference by directly learning a mapping.

7 Future Directions

The Gaussian-binary RBM that was used to initialize the feature-detecting layer of the neural network is not a very good generative model of images because it assumes that the pixels are independent given the features. A better generative model would include an explicit representation of the covariance structure of the image. This has been shown to improve discriminative performance for an object recognition task.

Most of the “errors” in the current system are due to the ambiguous nature of the labelling task. Our system often finds real roads that are simply not large enough to be labelled as roads by an expert. The use of vector maps that lack road width information also means that our system is penalized for correctly finding road pixels in wide roads such as highways. In addition to hurting the test performance, errors of this type hurt the training because the network is trying to fit inconsistent labels. A better way to handle ambiguous labels during training is to view the labels extracted from the map as noisy versions of an underlying set of true labels. This allows the neural network to override labels that are clearly incorrect during training. On an object recognition task, explicitly modeling the label noise greatly improves performance when a substantial proportion of the labels are incorrect.

8 Conclusions

We have presented an approach for automatically detecting roads in aerial imagery using neural networks. By using synthetic road/non-road labels and a consumer GPU board we were able to efficiently train much larger neural networks on much more data than was feasible before. We also showed how unsupervised pretraining and supervised post-processing substantially improves the performance of our road detector. The resulting road detection system works reliably on two large datasets of challenging urban data. To the best of our knowledge, no other published road detection system has been shown to work well on challenging urban data on such a scale.

References

1. Bajcsy, R., Tavakoli, M.: Computer recognition of roads from satellite pictures. IEEE Transactions on Systems, Man, and Cybernetics 6, 623–637 (1976)
2. Baltsavias, E.P.: Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. ISPRS Journal of Photogrammetry and Remote Sensing 58, 129–151 (2004)
3. Mayer, H.: Object extraction in photogrammetric computer vision. ISPRS Journal of Photogrammetry and Remote Sensing 63, 213–222 (2008)
4. Hu, J., Razdan, A., Femiani, J.C., Cui, M., Wonka, P.: Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints. IEEE Transactions on Geoscience and Remote Sensing 45, 4144–4157 (2007)
5. Laptev, I., Mayer, H., Lindeberg, T., Eckstein, W., Steger, C., Baumgartner, A.: Automatic extraction of roads from aerial images based on scale space and snakes. Machine Vision and Applications 12, 23–31 (2000)
6. Bhattacharya, U., Parui, S.K.: An improved backpropagation neural network for detection of road-like features in satellite imagery. International Journal of Remote Sensing 18, 3379–3394 (1997)
7. Boggess, J.E.: Identification of roads in satellite imagery using artificial neural networks: A contextual approach. Technical report, Mississippi State University (1993)
8. Huang, X., Zhang, L.: Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. International Journal of Remote Sensing 30, 1977–1987 (2009)
9. Mokhtarzade, M., Zoej, M.J.V.: Road detection from high-resolution satellite images using artificial neural networks. International Journal of Applied Earth Observation and Geoinformation 9, 32–40 (2007)

10. Dollar, P., Tu, Z., Belongie, S.: Supervised learning of edges and object boundaries. In: CVPR 2006: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1964–1971 (2006)
11. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science 313, 504–507 (2006)
12. Larochelle, H., Bengio, Y., Louradour, J., Lamblin, P.: Exploring strategies for training deep neural networks. Journal of Machine Learning Research 10, 1–40 (2009)
13. Hinton, G.: Training products of experts by minimizing contrastive divergence. Neural Computation 14, 1771–1800 (2002)
14. Norouzi, M., Ranjbar, M., Mori, G.: Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning. In: CVPR (2009)
15. Jain, V., Seung, S.: Natural image denoising with convolutional networks. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems, vol. 21, pp. 769–776 (2009)
16. Mnih, V.: Cudamat: a CUDA-based matrix class for python. Technical Report UTML TR 2009-004, Department of Computer Science, University of Toronto (2009)
17. Wiedemann, C., Heipke, C., Mayer, H., Jamet, O.: Empirical evaluation of automatically extracted road axes. In: Empirical Evaluation Techniques in Computer Vision, pp. 172–187 (1998)
18. Mena, J.B., Malpica, J.A.: An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery. Pattern Recognition Letters 26, 1201–1220 (2005)
19. Geman, D., Geman, D., Jedynak, B., Jedynak, B., Syntim, P.: An active testing model for tracking roads in satellite images. IEEE Transactions on Pattern Analysis and Machine Intelligence 18, 1–14 (1995)
20. He, X., Zemel, R.S., Carreira-Perpiñán, M.Á.: Multiscale conditional random fields for image labeling. In: CVPR 2004: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 695–702 (2004)
21. Peng, T., Jermyn, I., Prinet, V., Zerubia, J.: An extended phase field higher-order active contour model for networks and its application to road network extraction from vhr satellite images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 509–520. Springer, Heidelberg (2008)
22. Porway, J., Wang, K., Yao, B., Zhu, S.C.: A hierarchical and contextual model for aerial image understanding. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2008)

Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry

Varsha Hedau[1], Derek Hoiem[2], and David Forsyth[2]

[1] Department of Electrical and Computer Engineering
[2] Department of Computer Science,
University of Illinois at Urbana Champaign
{vhedau2,dhoiem,daf}@uiuc.edu

Abstract. In this paper we show that a geometric representation of an object occurring in indoor scenes, along with rich scene structure can be used to produce a detector for that object in a single image. Using perspective cues from the global scene geometry, we first develop a 3D based object detector. This detector is competitive with an image based detector built using state-of-the-art methods; however, combining the two produces a notably improved detector, because it unifies contextual and geometric information. We then use a probabilistic model that explicitly uses constraints imposed by spatial layout – the locations of walls and floor in the image – to refine the 3D object estimates. We use an existing approach to compute spatial layout [1], and use constraints such as objects are supported by floor and can not stick through the walls. The resulting detector (a) has significantly improved accuracy when compared to the state-of-the-art 2D detectors and (b) gives a 3D interpretation of the location of the object, derived from a 2D image. We evaluate the detector on beds, for which we give extensive quantitative results derived from images of real scenes.

1 Introduction

We spend much of our lives in a box. We eat, work, and sleep in areas that are limited by orthogonal planes, populated with carefully arranged furniture. Yet, despite their importance and rich structure, such environments are highly challenging for current recognition methods, largely because the near-field objects do not conform to orthographic projective assumptions.

In this paper, we propose an approach to *think inside the box*, building tightly constrained models of appearance and interactions of objects in a way that reflects the dominant structure of the indoor scene. Our assumption is that the objects are aligned with the dominant directions of the scene. This allows us to integrate global scene orientation cues in an object appearance model, which considerably simplifies the otherwise challenging view invariant object detection. Using the perspective inside the room, we recover an approximate 3D localization of an object, which further facilitates incorporating even richer spatial interactions between objects and room's global geometry.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 224–237, 2010.

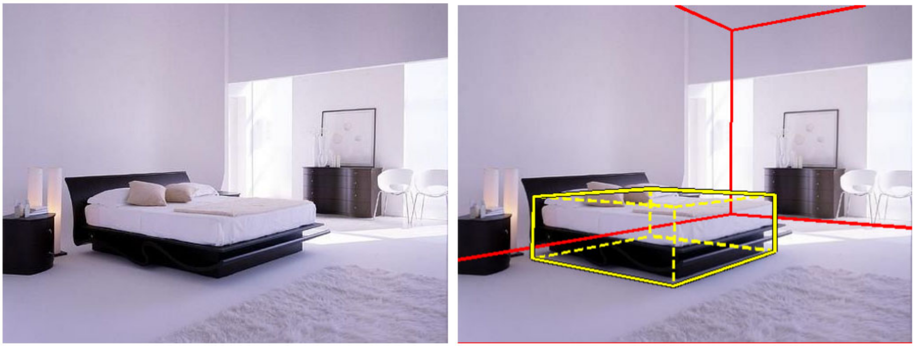


Fig. 1. We "think inside the box" to detect objects, which are modeled as axis aligned cuboids (shown in yellow) with the scene. The scene is represented as a box layout from Hedau et al. [1] (shown in red). By using the surrounding scene perspective to help model appearance, we can improve detection and localize the object in 3D. Furthermore, we show that supplying more information about the spatial interactions with the scene layout produces better detection.

We build on our earlier work [1] to obtain estimates of the room layout, which is modeled by a 3D oriented 'box' and a pixel labeling of major surfaces. Our focus is to use that layout information to improve object recognition in two key ways. First, we adapt the 2D sliding window detector strategy to a 3D domain, searching for parts in frontal-rectified features and stitching them together with a sliding 3D cuboid. Second, we model the relations of the objects with respect to the room, encoding soft constraints of size, visibility, and likely position within the room. In experiments on bed recognition for indoor scenes, we demonstrate that both of these innovations yield significant improvements.

1.1 Related Work

Our work builds on a wide range of techniques from literature on object recognition, 3D scene modeling, and contextual reasoning. In object recognition, the idea of sliding window detection with statistical templates has long been a mainstay [2,3,4,5] due to its simplicity and effectiveness for many categories. Within this framework, Dalal and Triggs [6] demonstrate that spatially local histograms of gradient (HOG) are effective features for pedestrian detection. More recently, Felzenszwalb et al. [7] extend this model to allow deformable latent parts, each modeled by its own HOG-based statistical template detector. We extend these ideas to 3D. Our object models are cuboids, composed of 3D planar parts whose orientations are defined with respect to the dominant orientations of the room. Like recent work [6,7], we detect these parts using HOG-based detectors, but our gradient images are frontally rectified such that rectangles in 3D become rectangles in the rectified image. This modification makes our detector invariant to viewpoint and is necessary for the near field case of indoor scenes, where object orientation changes rapidly with its location in the image. Similar to 2D sliding window detectors, we search for the objects by scanning a 3D cuboid at increments of ground plane position and scale.

Several recent works (e.g. [8,9]) also explore 3D-based object models, typically modeling objects as a collection of affine-transformed parts that have some spatial relation to each other or to the object center. These methods require complicated processes to align parts across views and instances and to build categorical spatial models. Because we annotate the corners of objects in training, our training and inference processes are very simple but effective. We have found our 3D-based features to be complementary to existing 2D models and show that a combination outperforms either alone.

Our work also adds to numerous efforts in image-based 3D scene estimation [10,11,12,13,14] and contextual reasoning [15]. Many previous approaches such as [16,17,18] use context in form of rough geometric constraints such as relative location and depth estimates, to improve object recognition in 2D. Our goal is to recover full 3D spatial extent of an object coherent with the surroundings, which requires stricter and richer constraints. The planar parts of our object cuboids are oriented 3D rectangles, which were shown to be useful for structure modeling in [19,20]. We build on our indoor spatial layout method [1], which estimates the principal orthogonal vanishing points, a 3D box layout, and a pixel labeling of the room surfaces. We use this method to obtain the orientation and our initial estimate of room layout.

Our probabilistic contextual reasoning most closely resembles works by Hoiem et al. [21] and Leibe et al. [22]. Like Hoiem et al. [21], we softly enforce size consistency through probabilistic inference. However, our 3D object models allow us to avoid making assumptions of roughly level cameras and orthographic projection, which is crucial for estimating object size in the near-field. Leibe et al. [22] detect and track objects from a moving video while constraining the objects to lie within the recovered street corridor. Because our scene estimates are recovered from a single image, we marginalize over the layouts while softly constraining that objects should lie within the room. Additionally, we model the spatial position of objects with respect to the walls.

1.2 Overview of Our Approach

Fig. 2 shows the overview of our approach. We start by detecting vanishing points corresponding to 3 orthogonal directions of the world using the vanishing point method from [1]. This gives us the orientation of object cuboids. By assuming that objects rest on floor we generate object candidates at several scales and translations by sliding a cuboid on floor planes at several different heights below camera. Fig. 2(c) shows some sample candidates obtained by our approach. Object cuboid is represented in terms of its planar sides or ‘faces’. We detect objects by searching for their axis aligned faces using rectified gradient features as shown in Fig. 2(d). Fig. 2(e) shows several detected horizontal and vertical cuboid faces parallel to the room orientation. These responses are combined together to score the object cuboids which are further refined probabilistically by using the object and scene layout interaction. Fig. 2(f) shows the detected object.

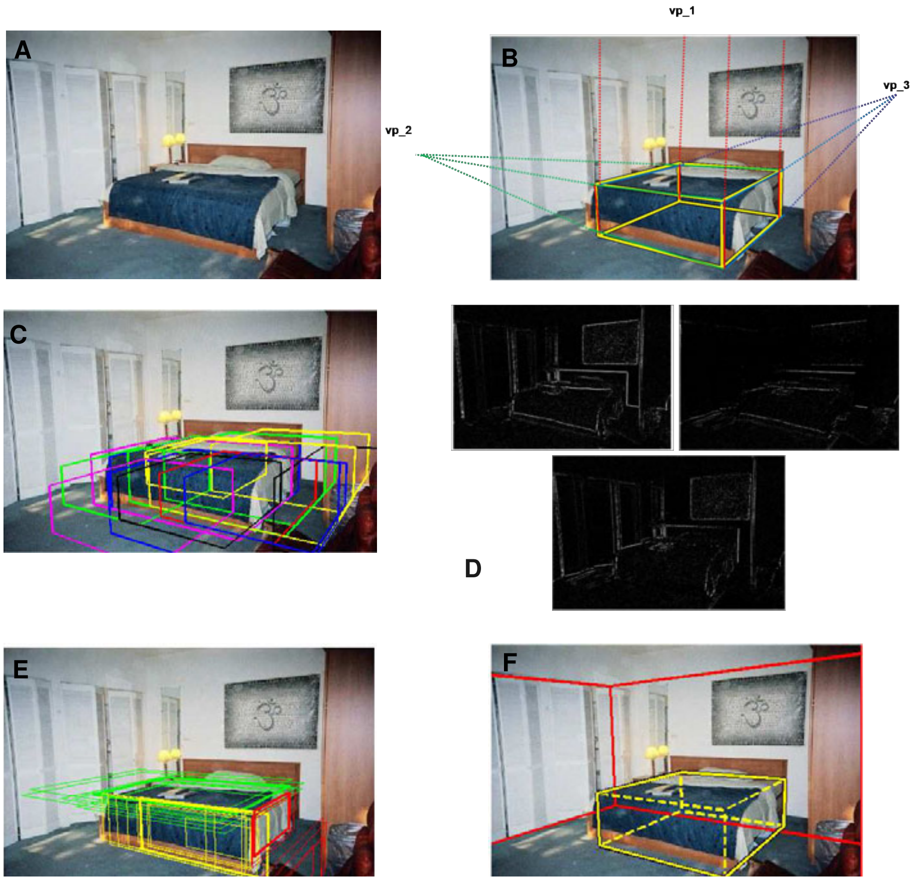


Fig. 2. Our algorithm takes original image and estimates the vanishing points of the three orthogonal directions of the world using method of [1]. This fixes the orientation of objects. We then generate many object candidates by sliding a cuboid in 3D. A sample of candidates are shown with different colors in (c). We detect cuboids by searching for their axis aligned 'faces' using rectified gradient features(shown in (d)). Some face samples with high response in each of the three orientation are shown in (e) with red, green and yellow. Bed cuboid detected by this procedure is further refined with the constraints provided by box layout of scene (shown in red) using a simple probabilistic model. The highest scoring bed cuboid is shown in yellow.

2 Detecting Objects

Classical sliding window approaches provide 2D localization of objects and cannot be easily used to predict 3D location and extent. Treating the objects as 2D planar cardboards has a disadvantage of knowing very little about their spatial interaction with the background. Indoor scenes are highly structured, and this information can be used to obtain a reasonable 3D localization of an object, which is our target in this paper. By assuming that the faces of object cuboids are parallel to the walls, the orientation of objects can be obtained from the rooms orientation given by vanishing points. Following this we build a searching strategy by sliding a cuboid in 3D to obtain object hypotheses and looking for consistent projected gradients in the image to score these hypotheses effectively.

We model objects as cuboid shaped boxes resting on floor in 3D. Most of the objects in indoor settings can be approximated by cuboid-like structure, e.g., beds and other furniture. Also, in a typical setting, the objects are parallel to room walls. Towards detecting objects in indoor scenes, we thus first estimate vanishing points corresponding to the three orthogonal directions of the room, which also serve as the vanishing points for the objects. These vanishing points fix the object orientation with respect to the camera. To estimate its translation, we generate object hypotheses constrained according to the vanishing points. For each of these hypotheses, we extract specialized features using perspective cues, and score them using a function learned from training images. In this paper we evaluate our method on beds, but, in principle our modeling procedure can also be extended to other similar objects such as chairs, cupboards, and tables.

2.1 Generating Object Hypotheses

To estimate the orientation of object cuboids in the image, we first estimate the vanishing points corresponding to the three orthogonal directions of the room. We use the method of Hedau et al. [1] for vanishing point estimation, which builds upon the method of [23]. The method detects long straight lines in the image. The lines corresponding to principal orthogonal directions in 3D should intersect at the corresponding vanishing points. The method thus accumulates votes for intersection points of these lines based on angular distance between the lines and the points. A triplet of points that gathers maximum votes and satisfies the orthogonality constraints gives three vanishing points. To speed up the search for this triplet, the space of points is quantized with variable bin size increasing as one goes away from the center of the image. Using the vanishing points one can estimate the rotation of the camera with respect to the room (and objects), as well as camera intrinsic parameters [23]. We next describe how we generate object hypotheses using the information of vanishing points.

Given the vanishing points corresponding to the three orthogonal directions of the room, $\{\overline{vp}_i\}_{i=1}^{3}$, assuming a camera with zero skew and square pixels, one can estimate the camera intrinsic matrix K and its rotation with respect to the room, R. Let us consider the coordinate system centered at the camera optical center whose axes are along the room directions: x-axis along the room width (left to right), y-axis along room height (bottom to top), and z-axis along room depth (towards the camera). For a point $\overline{X}$ in this coordinate system, its homogeneous coordinate projection in image plane $\overline{x}$ can be computed using the following projection relation.

$$c\overline{x} = KR\overline{X} \tag{1}$$

We make the following assumptions about the objects:

1. The object planes are parallel to the walls. It is possible to search angles that are not parallel to any wall, but we keep to the three principal directions in this work.
2. The object base touches the floor. This is true for most furniture in a room. Given a reference base corner point $\overline{X}$ of the object in 3D, the other k corner points $\{\overline{X}_i\}_{i=1}^{k}$ can be computed for given dimensions of the object cuboid.

To generate object hypotheses, we first fix the camera height h_c to an arbitrary value. Any object base corner point lying on the floor $\overline{X}$, should thus satisfy $\overline{X}^T n + h_c = 0$, where $n = (0, 1, 0)$ is the normal to the floor plane. We use this constraint to fix the reference base corner point of the object; the other corners are computed using object dimensions. The projections of these corners in the image can be computed using equation (1). We generate these object hypotheses for different discrete values of camera heights and object dimensions. For our experiments in Sec. 4 we vary camera height from 2.5 ft. to 8.5 ft. with 1 foot increments and use the aspect ratios of $2.5 \times 6 \times 5$ and $2.5 \times 7 \times 6$ ft. for beds in 3D. Note that the extent of floor plane for a camera height is bounded by the horizon line, the vanishing line joining the horizontal vanishing points. We use this constraint to limit the number of generated hypotheses. We typically get $4K$ to $30K$ object hypotheses per image.

2.2 Scoring Object Hypotheses

Part based approaches to modeling objects have shown good results. We model an object cuboid $\overline{c}$ as a collection of its faces, i.e., $\overline{c} = \{f_i\}_{i=1}^{F}$, where F is the number of faces. Given the configuration of the object, some faces are occluded, hence we attach a face visibility variable $\overline{v} = \{v_i\}_{i=1}^{F}$. Since the perspective distortion of each face is known, we extract features from each face after correcting for this distortion, denoted by $G = \{\overline{g}_i\}_{i=1}^{F}$ (see features described next). We independently score each face using a linear function on respective features, $s(f_i) = \overline{w}_i^t \overline{g}_i$, where $\overline{w}_i$ is weight vector learned from linear SVM. To deal with variations in object dimensions and for better localization, we allow the individual faces to deform slightly. For this, we modify the score of each face by the best scoring face in the neighboring object hypotheses $f_j \in \mathcal{N}(f_i)$. The final score of an object hypothesis $\overline{c}$ is thus given by

$$scr(\overline{c}) = \frac{\sum_i v_i \max_{f_j \in \mathcal{N}(f_i)} s(f_j)}{\sum_i v_i} \tag{2}$$

Features. Standard histogram of oriented gradients (HOG) features implicitly assume that the 2D image projection is the natural coordinate frame in which to view objects. Our method supposes that a 3D coordinate frame oriented with the room is better. We bin the gradients of image with respect to the direction of each pair of vanishing points. We use 6 orientation bins in our experiments. Fig. 2 (D) shows gradients binned in directions of each pair of vanishing points. Efficient

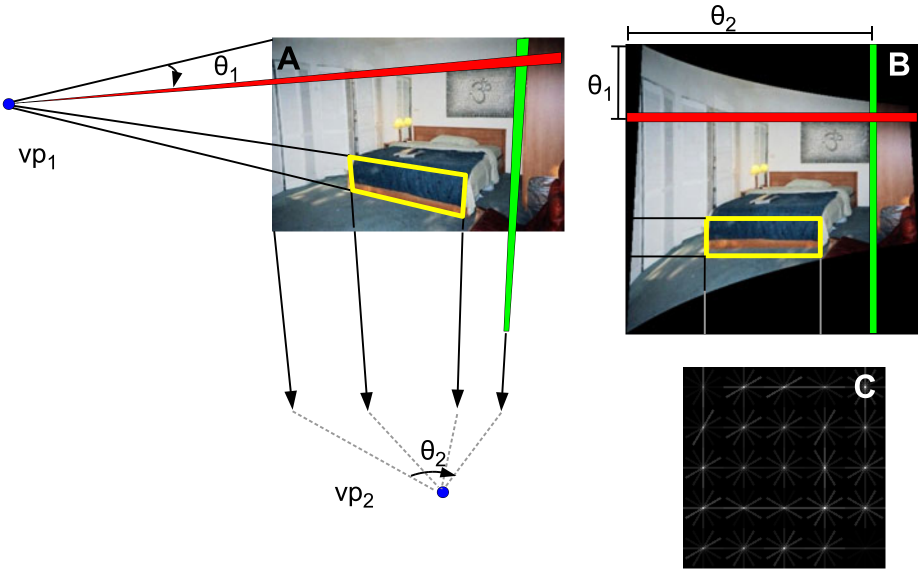


Fig. 3. Computing rectified HOG. We construct the orientation histograms with respect to the surrounding scene orientation (vanishing points directions). Fig. 2(D) shows gradients binned in direction of each pair of vanishing points. We further rectify these gradient images by using a transformation induced by indexing pixels in original image by the angles they subtend at the vanishing points. Note that gradient images are rectified and the original images are shown here only for simplicity. Such a rectification allows efficient computation of HOG in rectangular regions via integral images.

computation of HOG features is possible in rectangular windows however the projection of oriented rectangles in 3D are not rectangular in images. For this reason we compute histogram of gradients for a face in rectified coordinates corresponding to its vanishing points where it is frontal. Each face is divided into 5 × 5 cells and local normalization is done as described in [7]. Fig. 3(A,B) illustrates this rectification for a face with vanishing points vp_1, vp_2 and the corresponding HOG features are shown in Fig. 3(C). For simplicity we show the rectification on the original image however in principle gradient images, Fig. 2(D) are rectified. For computing face score HOG features computed with respect to the vanishing points of that face are used. Apart from HOG features we also use the line based features which is count of number of line pixels consistent with the orientation and average object label confidence, obtained from surface label estimates of [1] for each face.

Integrating the scores of a 2D Detector. There has been noticeable progress in 2D recognition approaches. We show how our cuboid detector can easily benefit from the state of art 2D methods [7]. Towards this we add the score of detector [7] in the bounding box of the cuboid to the cuboids score. Sec. 4 shows that we obtain a improved detector by incorporating the information from 2D detector.

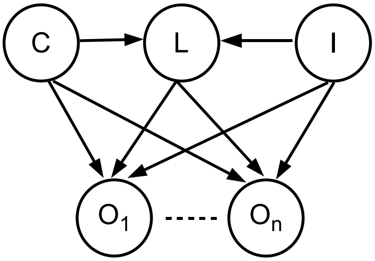


Fig. 4. Joint model of objects and the spatial layout. Objects are independant given layout and camera, leading to simple inference. Several spatial constraints such as objects can not penetrate walls and tend to occur at certain distances from wall can be encoded via this model (read text for more details). We show that incorporating such constraints leads to significant improvement in detecting objects.

3 Modeling the Interaction between Objects and Spatial Layout

Objects live in the scene and thus have to follow certain constraints due to the structure of the scene. These constraints can be used to help improve object detection. Towards this we propose to explicitly model the spatial interactions of objects with scene in a simple probabilistic framework. For scene structure we use our previous work, [1]. This work describes spatial layout of scene in terms of (a) a box layout that defines extent of walls, floor and ceiling, and (b) surface layout that gives pixel labeling of different surfaces as walls, floor, ceiling and objects. We have obtained the the spatial layout estimates on our images using the trained models from [1]. The choice of this spatial layout representation is intuitive for reasoning about spatial interaction between objects and the scene.

The box layout provides extent of walls and floor. The objects inside the box can not cross the boundaries of the walls. Also some objects tend to appear in certain spatial configurations with respect to the box. For instance, beds inside the rooms tend to be close to the walls. Thus knowing the extent of walls and floor provide important information about the placement of objects. Similarly an estimate of location of different objects inside the image can be used to refine the extents of wall floor boundaries.

Towards joint reasoning of objects and layout we propose a simple generative model, shown in Fig. 4. Here, $\{O_i\}_{i=1}^{N}$, $O_i \in \{0, 1\}$ are the object variables, O_i is whether a particular object is present or not, N is the number of objects, L is the box layout of the scene, C is the camera height and I is the image evidence. We consider all the detections left after doing a soft non-max suppression on the output of our cuboid detector (Sec. 2). The non-max suppression step greedily selects the highest the scoring detections while rejecting the ones that overlap more than a certain threshold with the existing selected detections. We use the thresold of 0.85 in our experiments. In this paper we have used beds as objects, however our framework is general enough to be applicable to other objects as well. The joint distribution over of objects, layout and camera can be written as

$$P(O_1, \ldots, O_N, L, C|I) = P(C)P(L|C, I) \prod_{i=1}^{N} P(O_i|L, C, I) \tag{3}$$

Here, $P(C)$ is the prior on camera height, assumed to be a Gaussian with mean $\mu = 5.5$ ft. (about eye level) and standard deviation $\sigma = 3$ ft. $P(L|C, I)$ is the layout likelihood conditioned on the camera which is estimated using layout scores obtained from the layout detector of [1] and features such as box layout height and depth given the camera height. $P(O_i|L, C, I)$ is the object likelihood conditioned on the layout and camera modeled as a logistic function given by,

$$P(O_i|L, C, I) = 1/(1 + \exp(-w^T \phi(O_i, L, C))) \tag{4}$$

where $\phi(O_i, L, C)$ is the feature set, consisting of (1) Scores from our object detector (described in Sec. 2); (2) Inferred object height given the camera height and horizon; and (3) Object-layout interaction features (described next). Objects are assumed to be independent given layout and the camera after non-max suppression, which leads to simple inference. We compute object marginals over a discrete set of sample values for camera heights and box layout. In our experiments we marginalize over top 100 layouts returned by the method of [1].

3.1 Interaction Features

We propose the object and layout interaction features which model 3D spatial constraints. This is possible due the 3D localization of objects provided by our object detector and the 3D extent of walls, floor obtained from [1]. As interaction features we use (a) overlap between object's footprint and the floor as an indicator of the extent of object sticking outside the floor i.e. into the walls; (b) distance between object and the walls which is computed as distance between the object and the nearest wall boundary, capturing the tendency of objects to occur at fairly consistent positions with respect to the layout.

Each of the above conditional likelihood is trained using logistic regression. The outputs of logistic regression are well calibrated probabilites. Inference is exact and straightforward on the above model.

Table 1. Average Precision (AP) for beds. Our 3D Cuboid detector is comparable with the state-of-art 2D object template detection method of Felzenszwalb et al. [7]. The combination of two detectors results in improvement in performance over each. The precise 3D extent of object provided by our cuboid detector facilitates incorporation of richer scene context which improves object detection significantly further.

Method	1.Cuboid detector	2. Felzenszwalb et al.	1+2	1+2+scene layout
Average Precision	0.513	0.542	0.596	0.628

4 Experiments

We evaluate our object detector and the joint layout model for beds on a dataset of 310 images of indoor scenes collected from Flickr and LabelMe [24]. We have labeled ground truth corners of beds in these images. We split this set randomly into 180 training and 130 test images.

Cuboid face detectors are trained using only bed images. We train one common detector for all the vertical faces of a cuboid and one for horizontal face. Faces that have overlap less than 50% with the ground truth are used as negative samples for the detector. We train the face detector using a linear SVM. For a face we allow some deformation by choosing its score as the maximum amongst all the faces having more than 75% overlap with it. Fig. 7 shows the precision-recall curves for our bed detector. Precision is the number of correct detections, and recall is the number of objects that are retrieved. We compare

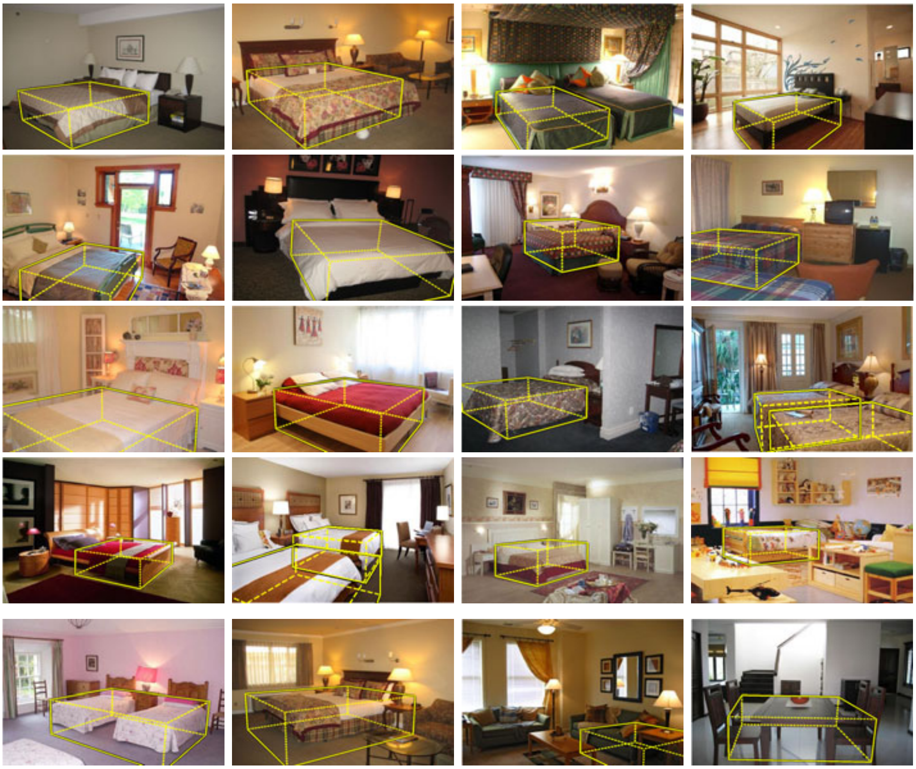


Fig. 5. Examples of high scoring detection selected from the first 100 top ranked detections of our algorithm on the test set images. First four rows show true positives, ground truth positive that are detected as positive and last row shows examples of false positives negatives that are detected as positives. Many false positives such as the the dining table and the sofa are due to high response of our detector to the strong oriented gradients in these areas.

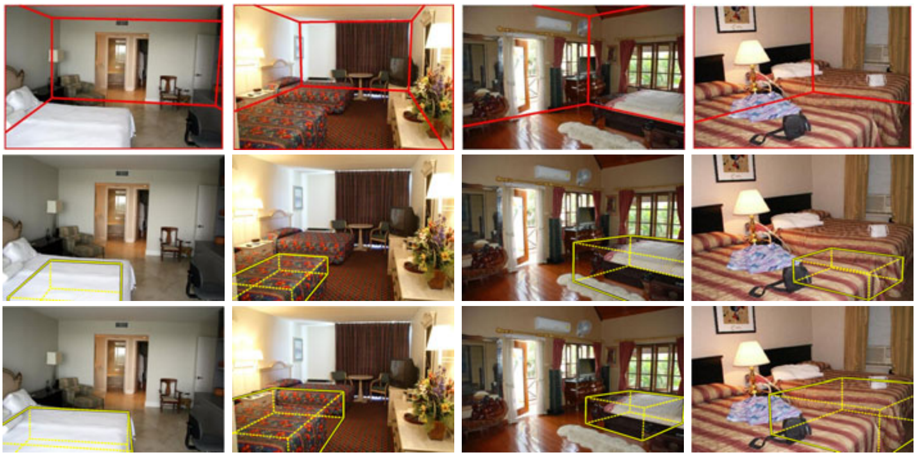
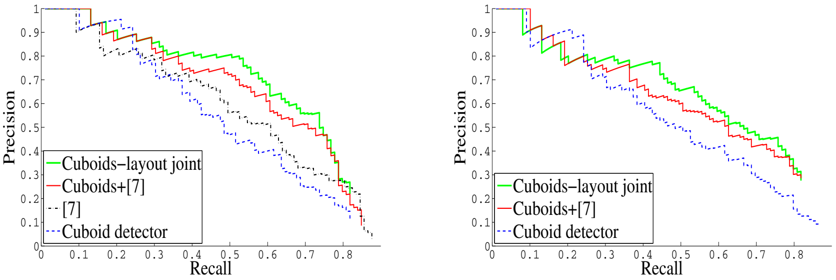


Fig. 6. Examples of improved object detections by joint modeling of objects and the scene layout. First row shows the best ranked box layout of scene obtained from Hedau et al. [1]. Second row shows the highest scoring beds by our cuboid detector in each image. Third row shows the highest scoring best detection obtained from our joint model. Note that first row shows only the best box layout for an image ranked by the box layout detector of [1], the bed detection is however obtained using marginal estimates of objects over discrete sample set of multiple high scoring box layouts. Notice how the joint model captures the tendency of beds occurring close to wall and the camera height prior prunes out the the detections with wrong scale estimates

our detector with the state of art 2D detector of Felzenszwalb et al. [7], which we train on our dataset. We use evaluation criteria similar to VOC Challenge. Precision recall curves are obtained for bounding box of the detected cuboids in order to compare with our baseline [7], which outputs bounding boxes. Average precision(AP) is computed over the entire test set. Our cuboid detector for beds has AP of 0.513 vs. 0.542 for the well-engineered 2D detector of [7].

To evaluate the additional information that is captured by our cuboid detector as compared to the 2D detector of [7] we combine the detection scores of this detector with our cuboid detector scores. For this we simply add to our score, the score of this detector in the bounding box of the cuboid. We obtain an improvement of 0.05 AP over [7] and 0.08 AP on our original cuboid detector (see Table 1). This suggests that 2D and cuboid detectors each have information to contribute to the other. We also show precision-recall curves in Fig. 7(b) computed using overlaps of the projected convex hull of the cuboids. Note that this is a stricter localization criterion as it also requires the object faces to overlap. We get similar improvement in performance by adding the score of 2D detector to our cuboid detector.

In Fig. 5 we show selected high ranked true positives (first four rows) and false positives (last row) of our cuboid detector. The cuboid detector often accurately localizes the bed in the images. It confuses other objects that have strong oriented



(a) **(b)**

Fig. 7. Precision-Recall curves for bed cuboid detector trained on our dataset of indoor images, (computed as bounding box overlap in **(a)**). We compare our method (blue curve) with the state of art 2D object template detection method of Felzenszwalb et al. [7] (black curve). Better performance is achieved by combining the scores of 2D detector with our cuboids suggesting some amount of complementary information provided by each. Cuboid detector is further improved by using the interactions with scene layout via a joint model (green curve). In **(b)**, we show the precision-recall curves computed using overlap of convex hull of cuboids. Here we achieve results similar to **(a)**. Note that this is a stricter criterion for evaluating localization. We can not compute this measure for [7] since its output is a bounding box.

gradients on them as beds (5th row, 3rd and 4th column). As seen Fig. 5, the detector is robust to cropping (3rd row, 1st col.), occlusion (4th row, 2nd col.), and clutter (4th row, 4th col.).

Finally we evaluate the performance of our joint object layout model. We achieve an AP of 0.628 from marginal estimates of objects obtained from our joint model. Fig. 6 shows several examples of improved object detections obtained by joint reasoning of the box layout, camera and the object cuboid. Notice how the interaction features of object and box layout helps to push the beds closer to the walls. The camera height prior helps in pruning out the detects with unlikely dimensions in 3D.

5 Conclusion

We have developed a detector to locate objects of a specific geometry in an indoor scene, while using object geometry, scene geometry, and their mutual arrangement. Using just a single image, the detector computes object localization in 3D that includes its location, orientation and extent, which is a lot more information when compared to 2D object detectors. The 2D localization performance of the detector is comparable to the state-of-the-art. When we combine our detector with a state-of-the-art 2D detector, there is a significant boost in performance, which indicates that the geometric constraints are highly informative.Furthermore, the visual results indicate that the detector can localize the object nicely, upto the level of its individual parts.

Such a 3D object detector can be used for generating a complete 3D layout of an image, which can in-turn aid graphics applications such as free space estimation, 3D walkthroughs, and image editing. In this paper, have demonstrated the concept of a sliding cuboid detector for a single object category, i.e., beds. However, in principle, the algorithm and the techniques discussed in this paper can also be extended to other objects such chair, sofa, table, dresser etc. Each of these objects can be modeled as a cuboid or as a cuboid with attached back rest, for instance chair and sofa. Likewise, our contextual framework could be extended to include other objects and people, with the goal of producing a complete, coherent parse of an image.

Acknowledgements. This work was supported in part by the National Science Foundation under IIS -0916014 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program.

References

1. Hedau, V., Hoiem, D., Forsyth, D.A.: Recovering the spatial layout of cluttered rooms. In: Proc. ICCV (2009)
2. Sung, K.K., Poggio, T.: Example based learning for view-based human face detection. Technical report, Cambridge, MA, USA (1994)
3. Rowley, H.A., Baluja, S., Kanade, T.: Neural network-based face detection. In: CVPR, p. 203. IEEE Comp. Society, Los Alamitos (1996)
4. Schneiderman, H., Kanade, T.: A statistical model for 3-d object detection applied to faces and cars. In: CVPR (2000)
5. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV 57 (2004)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
7. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI 99 (2009)
8. Hoiem, D., Rother, C., Winn, J.: 3d layoutcrf for multi-view object class recognition and segmentation. In: CVPR (2007)
9. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: ICCV, Kyoto, Japan (2009)
10. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Learning 3-d scene structure from a single still image. In: PAMI (2008)
11. Lee, D., Hebert, M., Kanade, T.: Geometric reasoning for single image structure recovery. In: Proc. CVPR (2009)
12. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV 75 (2007)
13. Barinova, O., Konushin, V., Yakubenko, A., Lee, K., Lim, H., Konushin, A.: Fast automatic single-view 3-d reconstruction of urban scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 100–113. Springer, Heidelberg (2008)
14. Gupta, A., Efros, A.A., Hebert, M.: Blocks world revisited: Image understanding using qualitative geometry and mechanics. In: ECCV (2010)
15. Hoiem, D., Efros, A.A., Hebert, M.: Closing the loop on scene interpretation. In: Proc. CVPR (2008)

16. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
17. Sudderth, E., Torralba, A., Freeman, W.T., Wilsky, A.: Depth from familiar objects: A hierarchical model for 3D scenes. In: Proc. CVPR (2006)
18. Heitz, G., Gould, S., Saxena, A., Koller, D.: Cascaded classification models: Combining models for holistic scene understanding. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) NIPS, pp. 641–648. MIT Press, Cambridge (2008)
19. Yu, S., Zhang, H., Malik, J.: Inferring spatial layout from a single image via depth-ordered grouping. In: CVPR Workshop (2008)
20. Tu, Z., Chen, X., Yuille, A.L., Zhu, S.C.: Image parsing: Unifying segmentation, detection, and recognition. IJCV 63, 113–140 (2005)
21. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: CVPR (2006)
22. Leibe, B., Schindler, K., Cornelis, N., van Gool, L.: Coupled object detection and tracking from static cameras and moving vehicles. PAMI 30, 1683–1698 (2008)
23. Rother, C.: A new approach to vanishing point detection in architectural environments. IVC 20 (2002)
24. Russell, B., Torralba, A., Murphy, K., Freeman, W.: Labelme: A database and web-based tool for image annotation. IJCV 77 (2008)

A Structural Filter Approach to Human Detection

Genquan Duan[1], Haizhou Ai[1], and Shihong Lao[2]

[1] Computer Science & Technology Department, Tsinghua University, Beijing, China
ahz@mail.tsinghua.edu.cn
[2] Core Technology Center, Omron Corporation, Kyoto, Japan
lao@ari.ncl.omron.co.jp

Abstract. Occlusions and articulated poses make human detection much more difficult than common more rigid object detection like face or car. In this paper, a Structural Filter (SF) approach to human detection is presented in order to deal with occlusions and articulated poses. A three-level hierarchical object structure consisting of words, sentences and paragraphs in analog to text grammar is proposed and correspondingly each level is associated to a kind of SF, that is, Word Structural Filter (WSF), Sentences Structural Filter (SSF) and Paragraph Structural Filter (PSF). A SF is a set of detectors which is able to infer what structures a test window possesses, and specifically WSF is composed of all detectors for words, SSF is composed of all detectors for sentences, and so as PSF. WSF works on the most basic units of an object. SSF deals with meaningful sub structures of an object. Visible parts of human in crowded scene can be head-shoulder, left-part, right-part, upper-body or whole-body, and articulated human change a lot in pose especially in doing sports. Visible parts and different poses are the appearance statuses of detected humans handled by PSF. The three levels of SFs, WSF, SSF and PSF, are integrated in an embedded structure to form a powerful classifier, named as Integrated Structural Filter (ISF). Detection experiments on pedestrian in highly crowded scenes and articulated human show the effectiveness and efficiency of our approach.

1 Introduction

Human detection has attracted much attention and significant progresses have been achieved in [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12]. However, highly accurate and real time human detection is still far from reality. There are mainly two difficulties for human detection: 1) Humans are highly articulated objects which change a lot in view, pose, size, position, etc. 2) Lots of things, including all around, may cause occlusions, like accessories (backpacks, briefcases, bags, etc.), or other persons. Especially in crowded scenes, humans always obscure each other.

Various algorithms are proposed for object detection to deal with occlusions or articulated poses. Deformable part model based on HOG features combined with a latent SVM was proposed in [2] for object detection, in which a root filter and several parts models are learned for each object category that can

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 238–251, 2010.

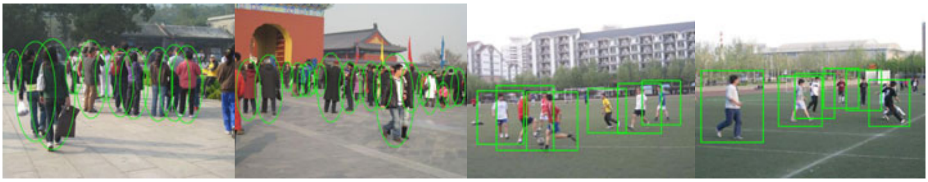


Fig. 1. Some combined results of Structural Filter approach to detect occluded pedestrian and articulated human

detect objects with some pose changes. A generic approach based on pictorial structure model was proposed in [7] to estimate human poses where a classifier is learned for each part and it infers locations of each part by a graph model. "Bags-of-words" method is widely applied to category [13] and detection [4][14] in computer vision. The approach in [14] is able to represent objects sparsely. Implicit Shape Model was proposed in [4] to detect pedestrians in crowed scenes in a bottom up way by a collection of visual words.

Holistic detectors are often limited when some parts are missing and it is even impractical to learn a holistic detector for objects with very large deformations. Therefore some approaches turn to parts/components to handle occlusions. Multiple occluded humans in [3] were detected by a Bayesian combination of part detectors where three types of body parts, head-shoulder, torsos and legs, are used. This approach was extended in [6] where a part hierarchy of an object class is defined and each part is a sub-region of its parent. There are also some component based methods to detect object through integrating part detectors by matching isomorphic graphs [15]. This kind of approach is more robust to occlusions where holistic object detectors will fail. But a critical issue here is how to integrate part detectors because parts tend to be less discriminative and part detectors are prone to producing more false positives. Some approaches rely on geometrical constraints of parts to handle false positives. But parts are easily missed due to occlusions, which often makes the constraints invalid.

Inspired by previous works in [3][6][15], in considering the relations among local regions, we propose a novel way to integrate part detectors, named as Structural Filter (SF) for object detection. Our aim is to handle occluded and articulated human detection in one framework and some combined results are shown in Fig. 1. A SF is a set of detectors which is able to infer what structures a test window possesses, where the structures could be *words*, *sentences* or *paragraphs*, corresponding to Word Structural Filter (WSF), Sentence Structural Filter (SSF) and Paragraph Structural Filer (PSF) respectively. A test window is positive if at least one detector in PSF provides a positive decision at last. We carry out some experiments on partially occluded pedestrian detection and articulated (multi-pose) human detection to demonstrate the effectiveness and efficiency of our approach.

The rest of this paper is organized as follows. The following section gives related work; Section 3 and Section 4 presents hierarchical structures of objects and Structural Filter separately; some experiments are carried out on pedestrian detection in crowded scenes and multi-pose human detection in section 5; and the discussion and the conclusion are given in the last two section.

2 Related Work

The first thing for human detection with occlusions and articulated poses is to model humans. Various models have been proposed to represent humans such as pictorial structure model [8], star model [7], multiple tree model [9], non-tree model [10] and part hierarchy model [6].

Pictorial structure [8] was proposed to represent humans by a joint configuration of parts in which an articulated model with 14 joints and 15 body parts was used and classifiers for each part using simple image features (first and second Gaussian derivatives) were learned. More discriminative detector for each part was proposed based on star model in [7]. In order to capture additional dependencies between body parts, multiple tree models was used in [9] to alleviate the limitations of a single tree-structured model. Non-tree model was proposed in [10] to enforce any type of constraints. These four typical models are proposed for pose estimation problem.

A part hierarchy model was proposed in [6] for detection and segmentation of partially occluded objects, in which parts are placed in specific locations and each part is a sub-region of its parent. Placing parts in specific locations is a convenient method for detection problem and provides the potential for sharing weak features. Detector ensemble [15] was proposed for face detection in heavy occlusions where sub-structures are applied to make each part more discriminative.

Following the works in [3][6][15], we build up a hierarchical structure of human and propose a Structure Filter approach to integrate part detectors to handle occlusions and articulated poses in one framework. The proposed hierarchical structure contains three levels, *words*, *sentences* and *paragraphs*, which combines the strengths of the approaches in [3][6][15]. The main differences are: 1) Parts are totally independent in [3] and each part is a sub-region of its parent in [6]. While in our method, *words* are basic units of objects. *Sentences*, consisting of *words*, are common sub structures of objects. *Paragraphs* are also composed of *words* and cover a set of *sentences*. *Paragraphs* correspond to the appearance statuses of detected objects, for example visible parts or particular poses in human detection. 2) Sub structures are also mentioned in [15] where a detector is learned for each sub structure and a detector ensemble which consists of a set of sub-structures gives a positive decision if at least one sub-structure is positive. While in our method, in addition to the two level structures, *words* and *sentences*, which are similar to [15], we add a *paragraph* level structure to learn a more robust detector to handle occlusions and articulated poses. 3) In our framework, a *word* is a general concept, which is a component of an object in a specific position and it can be a part, a component or a block. Furthermore, we propose a Structural Filter (SF) approach to integrate part detectors.

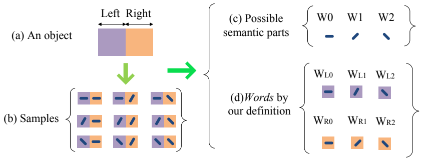


Fig. 2. The *words* and general semantic parts of an object. (a) An object with two regions, left and right. (b) Some samples of this object. (c) Possible general semantic parts. (d) *Words* defined in this paper.

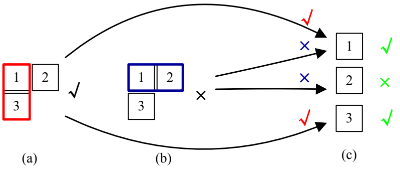


Fig. 3. SSF works interdependently. The red/blue blocks are two *sentences* shown in (a)/(b). Suppose a test window possesses the red structure but not the blue one, then the test result is that *word* "1" and word "3" are possessed as shown in (c).

Our contributions are summarized in three folds:

1) A three-level hierarchical object structure consisting of *words*, *sentences* and *paragraphs* in analog to text grammar is proposed for object detection.

2) A Structural Filter approach is proposed to integrate part detectors.

3) The proposed Structural Filter approach is a more general framework for object (rigid/non rigid) detection based on *words* (/parts/regions).

3 Three-Level Object Structure

3.1 Three-Level Object Structure

Words. A *word* is a component of an object in a specific block. In fact, the instance of *word* can be a part, a component or a block (of an area), which is similar as in [3] [6] [15]. Fig. 2 illustrates the difference of our *word* from general semantic part. It is worth mentioning that: 1) For humans, semantic "part" like head, leg, torso etc. may appear in different blocks due to no-rigid movement; 2) One block may contain several different parts of an object. In this paper, location is used as the first priority. One block may contain several parts for the Structural Filter approach to handle.

Sentences are sub-structures of an object which consist of *words*. A *word* is relatively less discriminative. Some of the *words* form a sub-structure which will be more discriminative as in [15]. Fig. 3 shows how SSF works interdependently.

Paragraphs corresponding to the appearance statuses of detected objects, are composed of *words* and cover a subset of *sentences*. Objects may show different statuses in different scenes. For example, parts of a pedestrian may be invisible in crowded scenes. The statuses of detected pedestrians can be head-shoulder, left-part, right-part, upper-body or whole-body.

3.2 Problem Formulation

Suppose an object O consists of N_W *words* which are denoted as a set $W = \{w_1, w_2, ..., w_{N_W}\}$. The *sentences* are $S = \{s_1, s_2, ..., s_{N_S}\}$ where N_S is the total

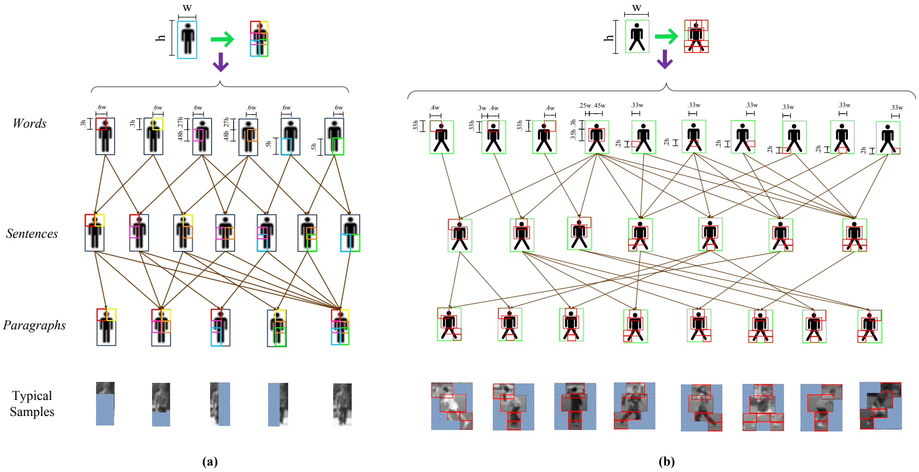


Fig. 4. Hierarchical structures of pedestrian and articulated human. (See Section 3.3 for details.)

number and each element $s_i (1 \leq i \leq N_S)$ is a subset of W. Similarly, *paragraphs* are represented as a set $P = \{p_1, p_2, ..., p_{N_P}\}$ where N_P is the number of the appearance statuses of detected objects and $p_i (1 \leq i \leq N_P)$ cover a set of S. *Sentences* are common sub-structures of an object which make *words* more discriminative and are used for inferences of *paragraphs*.

Each structure ϕ, either at *word* level, or *sentence* level or *paragraph* level, is associated with a detector with the detection rate $d(\phi)$ and the false positive rate $f(\phi)$. Our problem is to use a Structural Filter (SF) approach to integrate all these detectors. Each structure ϕ also has a missing tolerance parameter of parts, denoted as σ_ϕ, for integration.

3.3 Hierarchical Structures of Pedestrian and Articulated Human

As in [3][6], the simplest way to achieve *words* is to partition the sample space into some blocks according to heuristic knowledge. Hierarchical structures of pedestrian and articulated human are shown in Fig. 4 (a) and (b): 1^{st} row shows a sample space of pedestrian or articulated human; $2^{nd}/3^{rd}/4^{th}$ row shows *words*/*sentences*/*paragraphs* designed by prior knowledge; and 5^{th} row shows typical examples. The arrows between *words* and *sentences* show that *sentences* consist of *words*. Similarly, the arrows between *sentences* and *paragraphs* show that *paragraphs* cover a set of *sentences*.

Hierarchical structures of pedestrian. Pedestrians are relatively in strong cohesiveness. So we just evenly partition the sample space into six *words* shown in Fig. 4 (a). To deal with occlusions, we have defined five *paragraphs* of pedestrians, head-shoulder, upper-body, left-body, right-body and whole-body.

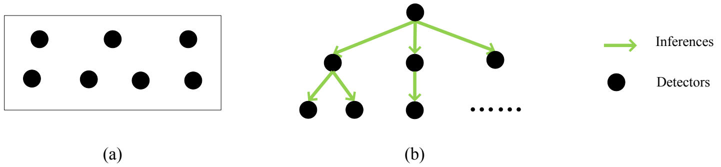


Fig. 5. Two typical methods to organize detectors, set method in (a) and tree method in (b)

Hierarchical structures of articulated human. Articulated (multi-pose) humans are more flexible than pedestrians. As a detection problem, all poses of humans as a whole are too difficult to deal with. We pay attention to a subset of poses where humans stand up on ground like walk, run etc. Mainly taking into account the varieties of heads and feet, we partition articulated human sample space into 10 *words* and define 8 *paragraphs* as shown in Fig. 4 (b).

4 Structural Filter Approach

4.1 The Definition of Structural Filters

A Structural Filter (SF) is a set of detectors which is able to infer what structures a test window possesses. Word Structural Filter (WSF) is composed of all the detectors for *words*, Sentence Structural Filter (SSF) is composed of all detectors for *sentences*, and so as Paragraph Structural Filer (PSF).

4.2 Three Level SFs: WSF/SSF/PSF

We adopt Real Adaboost [16] and Associated Pairing Comparison Features (APCFs) [1] to learn a cascade detector [17] for each structure (*word*, *sentence* or *paragraph*). APCF describes invariance of color and gradient of an object to some extent and it contains two essential elements, Pairing Comparison of Color (PCC) and Pairing Comparison of Gradient (PCG). A PCC is a Boolean color comparison of two granules and a PCG is a Boolean gradient comparison of two granules in which a granule is a square window patch. See [1] for details.

There are typically two methods to organize detectors in each SF of different levels: 1) The set method, where detectors are organized as a set and give decisions separately as shown in Fig. 5 (a). With the set method, all detectors involved are processed. 2) The tree method, where detectors are organized as a tree as shown in Fig. 5 (b). With the tree method, child nodes will be processed only if their parent node gives a negative decision. The tree method is much faster than the set method in decision making since only parts of its detectors are used. The tree method also provides the possibilities of sharing of weak features. For example, if the whole-body is visible, there is no need to test on head-shoulder detector or other detectors.

WSF and SSF tend to describe parts of objects. Each detector in WSF or SSF gives a decision independently. So detectors in WSF or SSF are organized as a set method. Organizing detectors in PSF as a tree method or a set method depends on the object to be detected. Fig. 4 (a) shows 5 paragraphs of pedestrian where left-part, right-part and upper-body are sub-regions of whole-body and head-shoulder is a sub-region of upper-body, so detectors in PSF for pedestrians are organized with a tree method. Fig. 4 (b) shows 8 paragraphs of articulated human where there is no any paragraph which is a sub-region of another one. So detectors in PSF for articulated human are organized with a set method.

4.3 Integrated Structural Filter

To construct a final human detector, the three level SFs, WSF, SSF and PSF, are integrated together to form a powerful classifier, which is called Integrated Structural Filter (ISF). The integration can be represented as sequences of WSF, SSF and PSF, for example, WSF$\Longrightarrow$SSF$\Longrightarrow$PSF, PSF$\Longrightarrow$SSF$\Longrightarrow$WSF or PSF$\Longrightarrow$WSF$\Longrightarrow$PSF$\Longrightarrow$SSF$\Longrightarrow$PSF. Each SF (WSF, SSF or PSF) in a sequence is called a *stage*.

Structural Filter inference is the inference by one stage of ISF, which can be summarized as three steps:

Step 1. Suppose that η is currently the stage to be dealt with, where η is one of WSF, SSF and PSF. Let Ω denote the set containing all passed *words* before the processing of η. Note that at the very beginning, Ω contains all words.

Step 2. A detector in η will be carried out if $|\omega| \leq \sigma_\phi$, in which ϕ is the structure associated to this structure, σ_ϕ is the missing tolerance and $\omega = \{w|w \in \phi, w \notin \Omega\}$. If this detector gives a positive decision, then push the structure ϕ into the passed structure set κ. Note that: 1) If the detectors in η are organized as set method, all detectors will be considered. 2) Else the detectors in η are organized as tree method. If the root detector gives positive decision, then its child detectors will be ignored and otherwise they will be considered.

Step 3. After the processing of η , update the passed *word* set $\Omega = \{\alpha|\alpha \in \phi, \phi \in \kappa\}$.

After each stage, we can obtain the passed *word* set. Actually we concern about the final decision of a test object's appearance statuses, so the last stage is always PSF and the statuses of a test object can be easily inferred by the passed structure set of the last stage.

Integration of SFs. Two simple methods for the integration are: 1) Bottom-Up method, which may be implemented by WSF$\Longrightarrow$SSF$\Longrightarrow$PSF, is similar to sub-structure in [15]. Bottom-Up method can depict parts of objects well, and is particularly efficient to deal with occlusion and share weak features. But it needs more time to discard negatives. 2) Top-Down method, which may be implemented by containing only one stage, PSF, gives the last decision directly. Top-Down method can discard negatives fast. But it is not easy to share features in Top-Down method.

In order to take advantages of both Bottom-Up method and Top-Down method, three level SFs, WSF, SSF and PSF, are integrated in an embedded structure

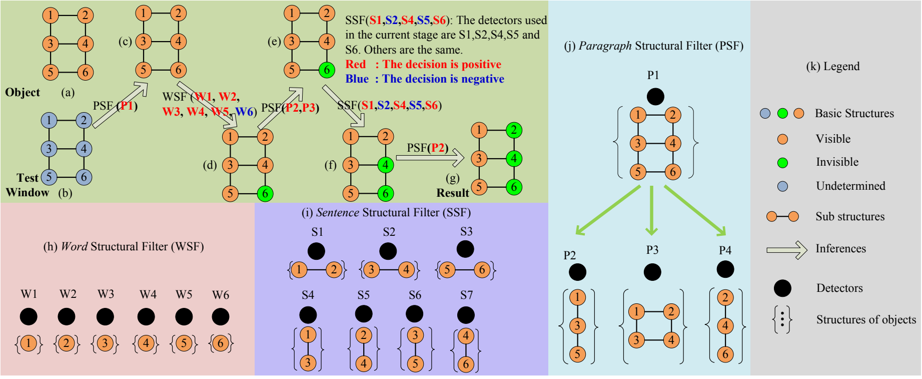

Object (a)
Test Window (b)
PSF (P1)
(c)
WSF (W1, W2, W3, W4, W5, W6)
(d)
PSF(P2,P3)
(e)
SSF(S1,S2,S4,S5,S6): The detectors used in the current stage are S1,S2,S4,S5 and S6. Others are the same.
Red : The decision is positive
Blue : The decision is negative
SSF(S1,S2,S4,S5,S6)
(f)
PSF(P2)
(g) Result
(h) Word Structural Filter (WSF)
W1 W2 W3 W4 W5 W6
(i) Sentence Structural Filter (SSF)
S1 S2 S3
S4 S5 S6 S7
(j) Paragraph Structural Filter (PSF)
P1
P2 P3 P4
(k) Legend
Basic Structures
Visible
Invisible
Undetermined
Sub structures
Inferences
Detectors
Structures of objects

Fig. 6. An example of ISF. (See Section 4.3 for details.)

through five stages, PSF⟹WSF⟹PSF⟹SSF⟹PSF, to form an ISF for both pedestrian detection and articulated human detection. Here the three PSFs are different sets of detectors in different stages. PSF in the 1^{st} stage is to discard negatives quickly. PSF in the 3^{rd} stage is to integrate the detection results of WSF and to provide passed *words* for SSF. PSF in the 5^{th} stage gives the final decision, positive or negative. Missing tolerances of *words* are applied when two consecutive stages of SFs are integrated. A detector in a SF will be involved only if missing *words* are within tolerance. A testing sample is positive if at least one detector in PSF gives a positive decision at last. **The learning algorithm** for ISF is summarized in Table 1.

An example of ISF is shown in Fig. 6. An illustrative "object" is shown in (a) which consists of six *words*. The missing tolerance for each word, sentence and paragraph is assumed to be zero. (h) (i) and (j) show WSF, SSF and PSF respectively, where detectors in WSF and SSF are organized by set mothod and PSF are organized by tree method. A test window (b) is processed by ISF with the flow shown in (c)-(g). For example, WSF (W1, W2, W3, W4, W5, W6) are applied from (c) to (d) where the used detectors are W1, W2, W3, W4, W5 and W6. **Red/Blue** means that a detector gives **positive/negative** decision.

Table 1. Learning algorithm for ISF

```
Input: Word set W; Sentence set S; Paragraph set P; Sample set $R = \{(x_i, y_i)|x_i \in \chi, y_i = \pm 1\}$ where $\chi$ is instance space; Five stages of ISF, PSF⟹WSF⟹PSF⟹SSF⟹PSF.
Initialize: Each detector in each stage of ISF is NULL.
For each stage $\psi$ in ISF ($\psi$ is WSF, SSF or PSF)
  * The structure set for $\psi$ is denoted as $\zeta$ ($\zeta$ is $\mathbf{W}$, $\mathbf{S}$ or $\mathbf{P}$)
  * For each structure $\phi$ in $\zeta$
    – Select $R^{'}$ ($R^{'} \subseteq R$). Enumerate each sample $\mathbf{x} \in R$. The passed word set $\Omega$ of $\mathbf{x}$ is inferred by all the previous stages. If missing words are within tolerance, add $\mathbf{x}$ into $R^{'}$.
    – Learn detector $\rho_\phi$ on sample set $R^{'}$ by algorithm in [1] and add $\rho_\phi$ to $\psi$.
Output: The learned ISF.
```

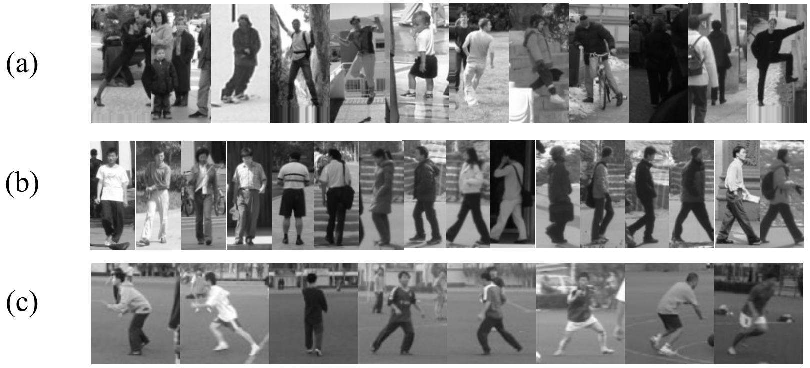


Fig. 7. Positives for pedestrian detection and articulated human detection. Images in (a) and (b) are from INRIA dataset and our collected dataset respectively. Both (a) and (b) are for pedestrian detection. (c) shows positives of articulated human.

The final decision is that the test window (b) is positive and its structure is the structure associated to P2 which is shown in (j).

5 Experiment

Experiments are done for partially occluded pedestrian detection and articulated human detection in cluttered backgrounds. We compare our ISF, which contains five stages, PSF⟹WSF⟹PSF⟹SSF⟹PSF with other state-of-the-art algorithms. In our experiments, the missing tolerance of *words* is set to 0 for PSF⟹WSF and PSF⟹SSF for both pedestrian detection and articulated human detection, while for WSF⟹PSF and SSF⟹PSF it is set to 1 for pedestrian detection and to 2 for articulated human detection. During the training for cascade classifiers, the detection rate is set to 0.998 and false positive rate is set to 0.33 for each layer of the detector associated to each structure, *word*, or *sentence* or *paragraph*, which guarantees that the ISF achieves high detection rate and low false positive rate. All experiments are conducted on an Intel Core(TM)2 2.33GHz PC with 2G memory.

5.1 Occluded Pedestrian Detection

INRIA [5] **dataset** is a popular public dataset for pedestrian detection. The database has 2416 64×128 people images for training and 1126 64×128 for testing. They are downscaled to 24×58 in our experiment. Some positives are shown in Fig. 7 (a). We compare ISF with other state-of-the-art algorithms by False Positive Per Window (FPPW). The ROC curve is given in Fig. 8, in which the x-axis is False Positives Per Window (FPPW), that is, FalsePos/(TrueNeg+FalsePos); and the y-axis is the detection rate, that is, TruePos/(FalseNeg+TruePos) or

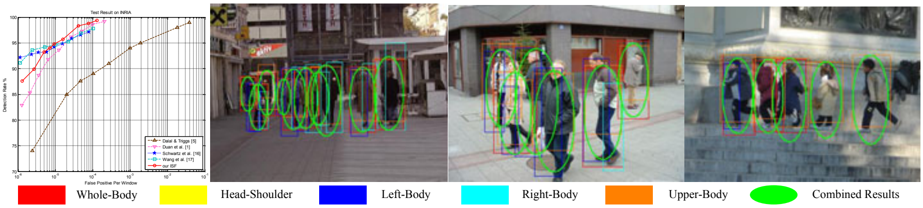


Fig. 8. Evaluation of ISF on INRIA dataset

1-missing rate. The result achieved by ISF improves the whole body detector [1] about 5% at FPPW=10^{-6} which is comparable to the results achieved in [18][19].

ETHZ [20] dataset consists of four video sequences (640 × 480 pixels at 15 frames/second), one for training and three for testing and only the three testing ones are used in our experiment. We collect 18474 positive samples of 24 × 58 for learning a robust ISF, which contains 9594 front/rear, 4440 left profile and 4440 right profile samples. Some positives are shown in Fig. 7 (b). We compare ISF with the methods in [20] and [18] by False Positive Per Image (FPPI) which is a better criterion for evaluating detector performance pointed out in [21]. In order to show the efficiency and effectiveness of ISF, we also train one stage PSF which is a Top-Down method mentioned in Section 4.3 on the same positive set.

When the intersection between a detection response and a ground-truth box is larger than 50% of their union, we consider it to be a successful detection. Only one detection per annotation is counted as correct. We obtain the ROC curves and some results shown in Fig. 9. Our ISF achieves better results than [20] and [18] on the first two sequences (Seq.#1 and Seq.#2), but the method in [18] achieves better results than ours on the third sequence (Seq.#3). The main reason is perhaps due to significant light changes in Seq.#3 for which our used features (APCFs) are somewhat sensitive. After the comparison of ISF and PSF, we can find that ISF achieves more accurate results with less false positives than PSF in general. The average cost time of ISF on ETHZ dataset is about 1.4s but that of PSF is about 2.6s. So ISF is much faster than PSF.

Furthermore, there are two things should be mentioned: One is that we do not use any additional cues like depth maps, ground-plane estimation, and occlusion reasoning, which are used in [20]. The other one is that there are some problems existed in ETHZ dataset which may affect the evaluation result as shown in Fig. 10. Some shadows of pedestrian are regarded as non-positives which are very hard for any pedestrian detector to identify and some pedestrians no longer exist in a scene are still labeled as positives.

More experiment results on USC SET B [3], Dataset S1 of PETS2009 [22] and our own collected dataset are given in Fig. 12.

5.2 Articulated Human Detection

We have labeled 11482 positive samples of 58 × 66 for articulated human detection. Typical positives are shown in Fig. 7 (c). Since currently there is no

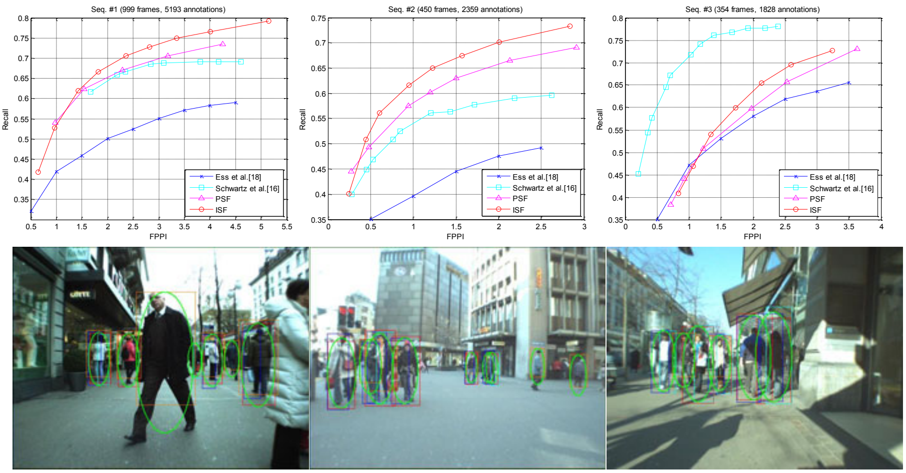


Fig. 9. Evaluation of ISF on ETHZ dataset

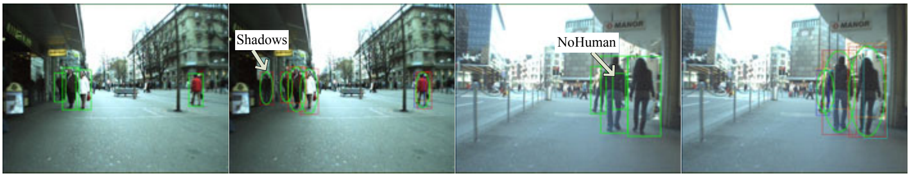


Fig. 10. Some groundtruths of ETHZ (in the 1^{st} and 3^{rd} columns) and our results (in the 2^{nd} and 4^{th} columns)

public available dataset for articulated human detection, we have labeled 170 images of 816 × 612 size with 874 humans for evaluation. Most of them are doing sports (playing football or basketball), so their poses differ a lot and are complex enough.

To compare with ISF, we have also trained PSF. The ROC curve and some results are shown in Fig. 11. This figure shows that ISF achieves more accurate results with less false positives than PSF. The average cost time of ISF is 1.8s and that of PSF is 9.2s. ISF is much faster than PSF.

6 Discussion

Feature sharing. One holistic detector is rather limited to handle occlusions of pedestrians. It is also difficult or impractical to train a usable holistic detector for articulated human due to the diversity. In our experiment, we show that our proposed SF approach is faster and more accurate than the approaches in which part detectors or specific poses detectors are fused simply. The intrinsic reason

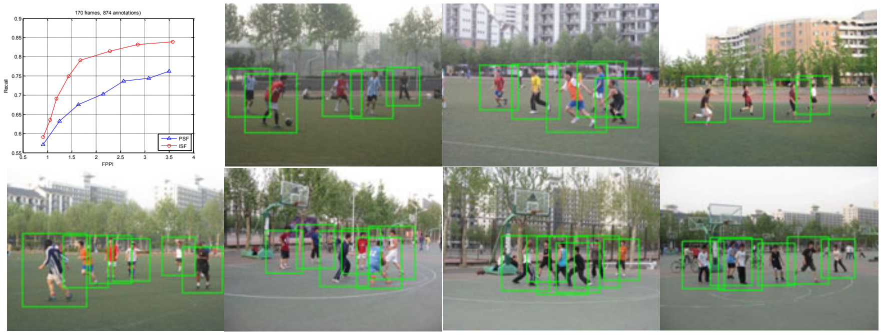


Fig. 11. Evaluation of ISF on our own collected dataset

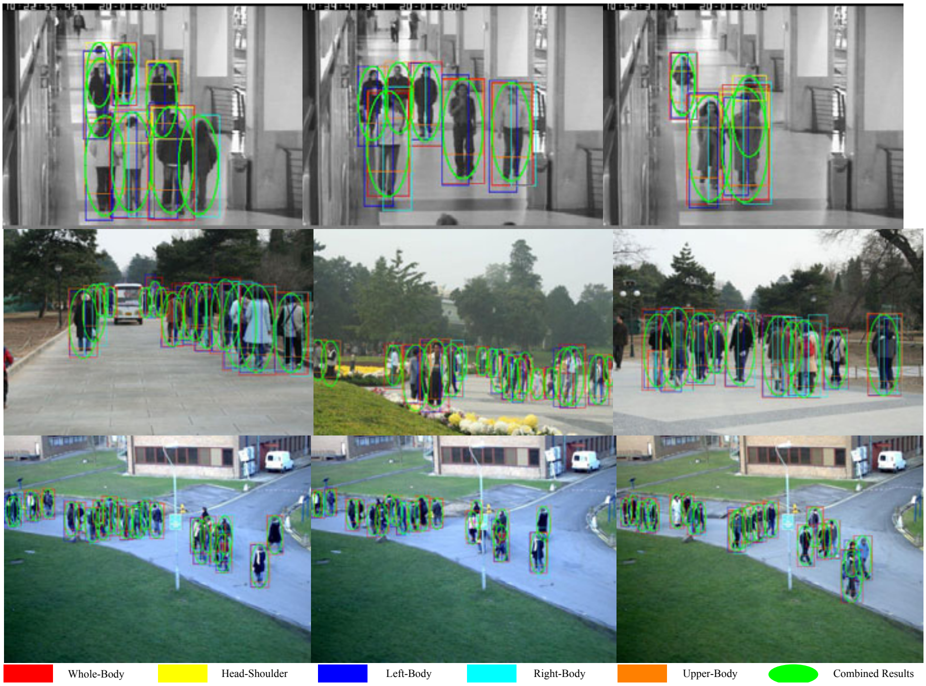


Fig. 12. Results of pedestrian detection on USC SET B(1^{st} line), our own collected dataset (2^{nd} line) and Dataset S1 of PETS 2009 (3^{rd} line)

lies on feature sharing. To explicitly define feature sharing, we first suppose two regions A and B, and region C is the shared area of A and B. Feature sharing means that A and B share the weak features in C. In our designed hierarchical object structures, a significant advantage of *words*, *sentences* and *paragraphs* is that they provide the potential to share weak features. For example, the weak features in head-shoulder can be shared with upper-body and whole-body.

Take the detectors learned for pedestrian detection as an example. There are 13417 weak features in PSF without any feature sharing, while there are 10714 weak features in ISF with feature sharing. Mainly due to the number of features in ISF is less than that in PSF, our ISF is faster than PSF. Feature sharing is of great benefit to our SF indeed. The experiment in the previous section also proves that ISF is more accurate than PSF, which in other words means that our SF approach has explored more discriminative features.

Relation with discriminative models (DM) and generative models (GM). We have proposed hierarchical structures and SF for object detection. In one hand, the detectors are learned by Boosting algorithm. From this point, our model is of DM. Detectors of different parts or poses in a traditional DM are independent but they are related to each other in our model. In another hand, the proposed hierarchical structures formulate one kind of object. From this point, our model is of GM. In fact, we have fused the DM of parts and GM of body structure in our approach.

7 Conclusion

In this paper, we present a SF approach to human detection. The three level SFs are WFS, SSF and PSF which correspond to a hierarchical structure of object, *words*, *sentences* and *paragraphs.* The approach can deal with occlusions and non rigid object detection. In a sense, it is a general framework for object (rigid/non rigid) detection based on *words* (/parts/regions). Experiment results on pedestrian detection in highly crowded scenes and articulated human detection demonstrate its effectiveness and efficiency.

There are some further works to be done to improve our SF approach. Currently *words* and *sentences* of an object are manually designed according to heuristic knowledge. It is hard to generalize this method for more complex objects therefore automatically learning of *words* and *sentences* is expected in the future.

Although the approach is proposed for human detection, we argue that it can be easily extended to other object detection problem, and also to multiple object categorization problems.

Acknowledgements

This work is supported in part by National Basic Research Program of China (2006CB303102), National High-Tech Research and Development Plan of China (2009AA01Z336), and it is also supported by a grant from Omron Corporation.

References

1. Duan, G., Huang, C., Ai, H., Lao, S.: Boosting associated pairing comparison features for pedestrian detection. In: 9th Workshop on Visual Surveillance (2009)
2. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)

3. Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: ICCV (2005)
4. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: CVPR (2005)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
6. Wu, B., Nevatia, R., Li, Y.: Segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. In: CVPR (2008)
7. Andriluka, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. In: CVPR (2009)
8. 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, pp. 700–714. Springer, Heidelberg (2002)
9. Wang, Y., Mori, G.: Multiple tree models for occlusion and spatial constraints in human pose estimation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 710–724. Springer, Heidelberg (2008)
10. Jiang, H., Martin, D.R.: Global pose estimation using non-tree models. In: CVPR (2008)
11. Lin, Z., Hua, G., Davis, L.: Multiple instance feature for robust part-based object detection. In: CVPR (2009)
12. Dollr, 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)
13. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
14. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 113–127. Springer, Heidelberg (2002)
15. Dai, S., Yang, M., Wu, Y., Katsaggelos, A.: Detector ensemble. In: Computer Vision and Pattern Recognition, CVPR (2007)
16. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning 37, 297–336 (1999)
17. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
18. Schwartz, W.R., Kembhavi, A., Harwood, D., Davis, L.S.: Human detection using partial least squares analysis. In: ICCV (2009)
19. Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: ICCV (2009)
20. Ess, A., Leibe, B., Gool, L.V.: Depth and appearance for mobile scene analysis. In: ICCV (2007)
21. Tran, D., Forsyth, D.: Configuration estimates improve pedestrian finding. In: NIPS (2007)
22. PETS 2009 (2009), http://www.cvg.rdg.ac.uk/PETS2009/

Geometric Constraints for Human Detection in Aerial Imagery

Vladimir Reilly, Berkan Solmaz, and Mubarak Shah

Computer Vision Lab, University of Central Florida, Orlando, USA
{vsreilly,bsolmaz,shah}@eecs.ucf.edu

Abstract. In this paper, we propose a method for detecting humans in imagery taken from a UAV. This is a challenging problem due to small number of pixels on target, which makes it more difficult to distinguish people from background clutter, and results in much larger searchspace. We propose a method for human detection based on a number of geometric constraints obtained from the metadata. Specifically, we obtain the orientation of groundplane normal, the orientation of shadows cast by humans in the scene, and the relationship between human heights and the size of their corresponding shadows. In cases when metadata is not available we propose a method for automatically estimating shadow orientation from image data. We utilize the above information in a geometry based shadow, and human blob detector, which provides an initial estimation for locations of humans in the scene. These candidate locations are then classified as either human or clutter using a combination of wavelet features, and a Support Vector Machine. Our method works on a single frame, and unlike motion detection based methods, it bypasses the global motion compensation process, and allows for detection of stationary and slow moving humans, while avoiding the search across the entire image, which makes it more accurate and very fast. We show impressive results on sequences from the VIVID dataset and our own data, and provide comparative analysis.

1 Introduction

In recent years improvements in electronics and sensors have allowed for development and deployment of Unmanned Aerial Vehicles (UAVs) on greater and greater scale, in a wide variety of applications, including surveillance, military, security, and distaster relief operations. The large amount of video data obtained from these platforms, requires automated video analysis tools, whose capabilities must include object detection, tracking, classification and finally scene and event analysis. While a number of methods and systems exist for detecting and tracking vehicles in UAV video (e.g. [1] [2]), the same cannot be said about human detection.

State of the art human detection methods such as [3] [4] [5] [6] [7], are designed to deal with datasets containing imagery taken from the ground, either in surveillance or consumer imagery scenario. People in that type of imagery are

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 252–265, 2010.

fairly large (e.g. 128x64 in the case of INRIA dataset). Also the camera in such scenarios is generally oriented with the ground plane. In our case, the humans are much smaller as seen in Figure 1. On average they are about 24x14 pixels in size, and have no visible parts, this makes part detection methods such as [4] and [6] inapplicable. Bag of feature models such as [5] also have great difficulty due to a very small number of interest points that can be found. Another issue is that since the camera is mounted on a moving aerial platform, the imaged size and visible distinguishing features of a person can be reduced even further when the camera is at a high elevation angle. Also, the moving aerial platform introduces a large number of possible orientations at which a human can appear in the scene. Due to lack of good distinguishing features of the human body in aerial imagery, a brute force image search generates many false detections, and is also quite slow. Hence, previous two works that specifically deal with aerial imagery ([8] and [9]), opt to constrain the search with preliminary processing.

A very popular approach is to constrain the search using motion as in [10], or Xiao et. al. [8]. They assume that only moving objects are of interest, and adopt a standard aerial surveillance pipeline. First, they compensate for global camera motion, then they detect moving objects, and finally classify each moving object as either a person or vehicle using the combination of histograms of oriented gradients (HOG) and a support vector machine proposed in [3]. The problem with the motion constraint, is that since people are viewed from far away, their motion is very subtle and difficult for the system to pick up. Of course, if people are stationary, then the system cannot detect them at all. If there are shadows present in the scene, then a number of additional problems arise. It is difficult to localize the human, since its shadow is part of the moving blob, which also makes the blobs more similar to each other making it more difficult to track them. See Figure 8 for examples of these failures.

Miller et. al. avoid the moving object assumption [9], by assuming that at least one Harris corner feature point will be detected on the human in each frame. This generates a large number of candidates which are then suppressed through tracking of the Harris corners in global reference frame. Each corner is then classified using a OT-MACH filter. If a track contains more human classifications than 20% of total track length, all points within track are labelled as human. The problem with the above approach is the large number of potential human candidates; they report 200 for a 320x240 image, and the need for a sophisticated tracker to filter them out.

We propose a very different approach. In particular we constrain the search by assuming that humans are upright shadow casting objects. We utilize directed low level computer vision techniques based on a set of geometric scene constraints derived from the metadata of the UAV platform. Specifically, we utilize the projection of the ground plane normal to find blobs normal to the ground plane, these give us an initial set of potential human candidates. Similarly we utilize the projection of shadow orientation to obtain a set of potential shadow candidates. We then obtain a refined set of human candidates, which are pairs of shadow and normal blobs that are of correct geometric configuration, and relative size.

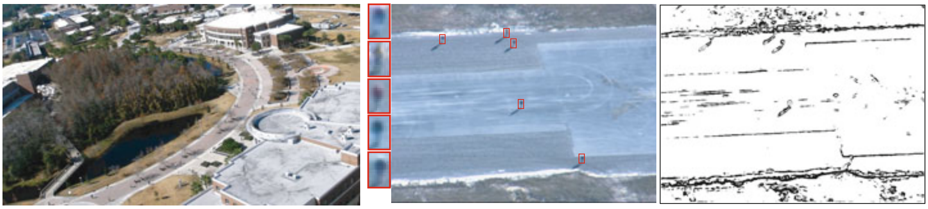


Fig. 1. On the left, are frames from some of the sequences, also examples of humans. The humans are only around 24x14 pixels in size, and are difficult to distinguish from the background. On the right, still image shadow detection using techniques from [11], pixels belonging to humans, and large parts of background were incorrectly labelled as gradient that belongs to shadow.

This is once again done based on projected directions, as well as the ratio of assumed projected human height and projected shadow length.

Once the refined set of candidates has been obtained, we extract wavelet features from each human candidate, and classify it as either human or clutter using a Support Vector Machine (SVM). Note that the main idea behind our geometric constraints is to improve the performance of any detection method by avoiding full frame search. Hence other models, features, and classification schemes suitable for aerial imagery can be used. Additionally, our method can be used to alleviate object localization problems associated with motion detection in presence of strong shadow.

The advantage of our constraints is that they do not require motion detection, registration, and tracking, which are time consuming, and can have their own problems. Additionally our method does not suffer degraded performance in presence of strong shadows. A slight disadvantage is that to get the full benefit, a strong shadow is necessary. However the initial set of candidates which we generate without using the shadow still performs better than brute force full-frame search (see section 4).

In absence of metadata, a static image shadow detector can be used to find the shadows in the image. For this purpose we extend the geometry detection method to work as a novel shadow detection method described in section 3.3. We found that standard shadow detection methods such as [11] and [12] perform poorly on real data (see Figure 1). The methods are based on obtaining illumination invariant (shadow-less) images, and comparing edges between these and original images. Since the humans and their shadows look similar in our data, the illumination invariant images would remove parts of shadows, humans and strong background gradients.

The main contribution of this paper is a novel method constraining human detection in aerial video, as well as a shadow detection method. In future work we will extend it to other object types. Our use of shadow is somewhat counterintuitive, since instead of treating it as a nuisance, we actually use it to help with the detection.

2 Ground-Plane Normal and Shadow Constraints

2.1 Metadata

The imagery obtained from the UAV has the following metadata associated with most of the frames. It has a set of aircraft parameters *latitude*, *longitude*, *altitude*, which define the position of the aircraft in the world, as well as *pitch*, *yaw*, *roll* which define the orientation of the aircraft within the world. Metadata also contains a set of camera parameters *scan*, *elevation*, *twist* which define the rotation of the camera with respect to the aircraft, as well as *focal length*, and *time*. We use this information to derive a set of world constraints, and then project them into the original image.

2.2 World Constraints

The Shadow is generally considered to be a nuisance in object detection, and surveillance scenarios. However, in the case of aerial human detection, the shadow information augments the lack of visual information from the object itself, especially in the cases where the aerial camera is close to being directly overhead. We employ three world constraints.

- The person is standing upright perpendicular to the ground plane.
- The person is casting a shadow.
- There is a geometric relationship between person's height and the length of their shadow. See Figure 2.

Given *latitude*, *longitude*, and *time*, we use the algorithm described in [13], to obtain the position of the sun relative to the observer on the ground. It is defined by the azimuth angle α (from the north direction), and the zenith angle γ (from the vertical direction). Assuming that the height of the person in the world is k we find the length of the shadow as $l = \frac{k}{\tan(\gamma - 90)}$, where γ is the zenith angle of the sun. Using the azimuth angle α we find the groundplane projection of the vector pointing to the sun, and scale it with the length of the shadow $\mathbf{S} = \langle l\cos(\alpha), l\sin(\alpha), 0\rangle$.

2.3 Image Constraints

Before we can use our world constraints for human detection, we have to transform them from the world coordinates to the image coordinates. To do this we use the metadata to obtain the projective homography transformation that relates image coordinates to the ground plane coordinates. For an excellent review of the concepts used in this section see [14].

We start by converting the spherical *latitude* and *longitude* coordinates of the aircraft to the planar Universal Transverse Mercator coordinates of our world $X_w = \mathit{east}$, and $Y_w = \mathit{north}$. Next, we construct a sensor model that transforms any image point $\mathbf{p}' = (x_i, y_i)$ to the corresponding world point $\mathbf{p} = (X_w, Y_w, Z_w)$. We do this by constructing the following sensor transform.

$$\Pi_1 = T^a_{Zw} T^e_{Xw} T^n_{Yw} R^y_{Zw} R^p_{Xw} R^r_{Yw} R^s_{Za} R^e_{Xa} R^t_{Ya}, \tag{1}$$

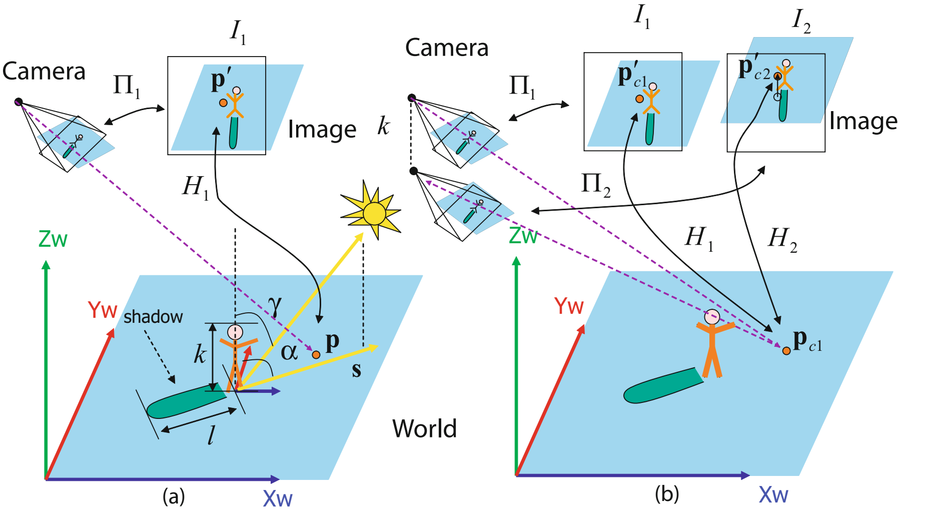


Fig. 2. Left, the sensor model Π_1 maps points in camera coordinates into world coordinates (since the transformation between image and camera coordinates is trivial we do not show it in the image).**X** corresponds to East direction, **Y** to North, **Z** to vertical direction. Vector **S** is pointing from an observer towards the sun along the ground. It is defined in terms of α - azimuth angle between northern direction and the sun. Zenith angle γ is between vertical direction and the sun. The height of a human is k, and the length of the shadow is l. We place the image plane into the world, and raytrace through it to find the world coordinates of the image points (we project from the image plane to the ground plane). We compute a homography H_1 between image points and their corresponding world coordinates on groundplane. Right, illustrates how we obtain the projection of the groundplane normal in the original image. Using a lowered sensor model Π_2 we obtain another homography H_2, which maps points in camera coordinates to a plane above the ground plane. Mapping a world point $\mathbf{p}_{c1}$ using H_1, and H_2, gives two image points $\mathbf{p}'_{c1}$, and $\mathbf{p}'_{c2}$. Vector from $\mathbf{p}'_{c1}$ to $\mathbf{p}'_{c2}$ is the projection of the normal vector.

where T^a_{Zw}, T^e_{Xw}, and T^n_{Yw} are translations for aircraft position in the world - *altitude*, *east*, and *north* respectively. R^y_{Zw}, R^p_{Xw}, and R^r_{Yw} are rotations for the aircraft - yaw, pitch and roll respectively. R^s_{Za}, R^e_{Xa} and R^t_{Ya} are rotation transforms for camera - scan, elevation, and tilt, respectively.

We transform 2D image coordinates $\mathbf{p}' = (x_i, y_i)$ into 3D camera coordinates $\hat{\mathbf{p}}' = (x_i, y_i, -f)$, where f is the *focal length* of the camera. Next, we apply the sensor transform from equation 1, and raytrace to the ground plane (see Figure 2 **(a)**).

$$\mathbf{p} = RayTrace(\Pi_1 \hat{\mathbf{p}}'). \tag{2}$$

Ray tracing requires geometric information about the environment, such as the world height at each point, this can be obtained from the digital elevation map of the area - DEM. In our case, we assume the scene to be planar, and project the points to the ground plane at zero altitude $Z_w = 0$.

For any set of image points $\mathbf{p'} = (x_i, y_i)$, raytraycing gives a corresponding set of ground plane point $\mathbf{p} = (X_w, Y_w, 0)$. Since we are assuming only one plane in the scene we only need correspondences of four image corners. We then compute a homography, H_1, between the two sets of points, such that $\mathbf{p} = H_1\mathbf{p'}$. Homography, H_1, will orthorectify the original frame, and align it with the North Direction. Orthorectification removes perspective distortion from the image and allows the measurement of world angles in the image. We use the inverse of the homography H_1^{-1} to project the shadow vector defined in world coordinates into the image coordinates. (see Figure 4 **(a)**).

$$\mathbf{S'} = \mathbf{S}H_1^{-1}. \tag{3}$$

Now, we obtain the projected ground plane normal (refer to Figure 2 **(b)**). We generate a second sensor model, where we lower the camera along the normal direction Z_w, by k, which is the assumed to be a person's height.

$$\Pi_2 = (T^a_{Zw} - [I|k])T^e_{Xw}T^n_{Yw}R^y_{Zw}R^p_{Xw}R^r_{Yw}R^s_{Za}R^e_{Xa}R^t_{Ya}. \tag{4}$$

Using the above sensor model Π_2 we obtain a second homography H_2 using the same process that was used for obtaining H_1. We now have two homographies, H_1 maps the points from the image to the ground plane, and H_2 maps the points from the image to a virtual plane parallel to the ground plane that is exactly k units above the ground plane. We select the center point of the image $\mathbf{p}'_{c1} = (x_c, y_c)$, and obtain its ground plane coordinates $\mathbf{p}_{c1} = H_1\mathbf{p}'_c$. Then we map it back to the original image using H_2, $\mathbf{p}'_{c2} = H_2^{-1}\mathbf{p}_c$. The projected normal is then given by

$$\mathbf{Z'} = p'_{c2} - p'_{c1}. \tag{5}$$

We compute the ratio between the projected shadow length and the projected person height as

$$\eta = \frac{|\mathbf{S'}|}{|\mathbf{Z'}|}. \tag{6}$$

3 Human Detection

3.1 Constraining the Search

In order to avoid the search over the entire frame, the first step in our human detection process is to constrain the search space of potential human candidates. We define the search space as a set of blobs oriented in direction of shadow, and direction of normal. To do so we utilize the image projection of the world constraints derived in the previous section - the projected orientation of the normal to the ground plane $\mathbf{Z'}$, the projected orientation of the shadow $\mathbf{S'}$, and the ratio between the projected person height, and projected shadow length η. See Figure 3.

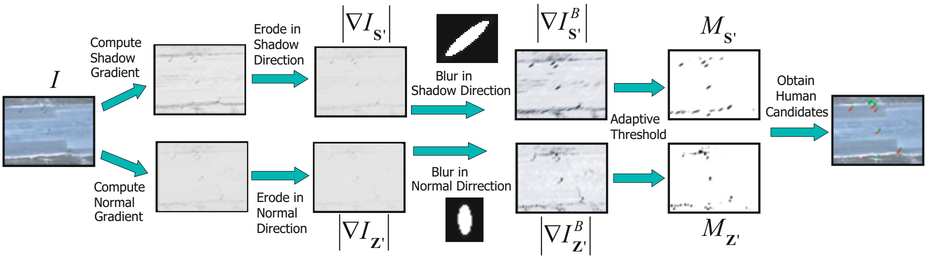

I
Compute Shadow Gradient
Compute Normal Gradient
Erode in Shadow Direction
Erode in Normal Direction
$|\nabla I_{\mathbf{S}'}|$
$|\nabla I_{\mathbf{Z}'}|$
Blur in Shadow Direction
Blur in Normal Dirrection
$|\nabla I^B_{\mathbf{S}'}|$
$|\nabla I^B_{\mathbf{Z}'}|$
Adaptive Threshold
$M_{\mathbf{S}'}$
$M_{\mathbf{Z}'}$
Obtain Human Candidates

Fig. 3. This figure illustrates the pipeline of applying image constraints to obtain an initial set of human candidates

Given a frame I, we compute gradient oriented in the direction of the shadow by applying a 2D Gaussian derivative filter,

$$G(x,y) = \cos(\theta)2xe^{-\frac{x^2+y^2}{\sigma^2}} + \sin(\theta)2ye^{-\frac{x^2+y^2}{\sigma^2}}, \tag{7}$$

θ is the angle between the vector of interest and the x axis,and take its absolute value. To further suppress gradient not oriented in the direction of the shadow vector we perform structural erosion along a line in the direction of the shadow orientation:

$$|\nabla I_{\mathbf{S'}}| = erode(\nabla I, \mathbf{S'}). \tag{8}$$

We obtain $|\nabla I_{\mathbf{Z'}}|$ using the same process. Next, we smooth the resulting gradient images with an elliptical averaging filter whose major axis is oriented along the direction of interest:

$$I^B_{\mathbf{S}'} = |\nabla I_{\mathbf{S}'}| * G_{\mathbf{S}'}, \tag{9}$$

where $B_{\mathbf{S'}}$ is an elliptical averaging filter, whose major axis is oriented along the shadow vector direction, this fills in the blobs. We obtain $I^B_{\mathbf{Z'}}$ using $G_{\mathbf{Z'}}$. Next, we apply an adaptive threshold to each pixel to obtain shadow and normal blob maps.

$$M_{\mathbf{S}'} = \begin{cases} 1 \text{ if } I^B_{\mathbf{S'}} > t \cdot mean(I^G_{\mathbf{S'}}) \\ 0 \text{ otherwise}, \end{cases} \tag{10}$$

See Figure 4 for resulting blob maps overlayed on the original image. We obtain $M_{\mathbf{Z'}}$ using the same method. From the binary blob maps we obtain a set of shadow and object candidate blobs using connected components. Notice that a number of false shadow and object blobs were initially detected, and later removed.

3.2 Exploiting Object Shadow Relationship

The initial application of the constraints does not take into account the relationship between the object candidates and their shadows, and hence generates many false positives. Our next step is to relate the shadow and human blob maps, and

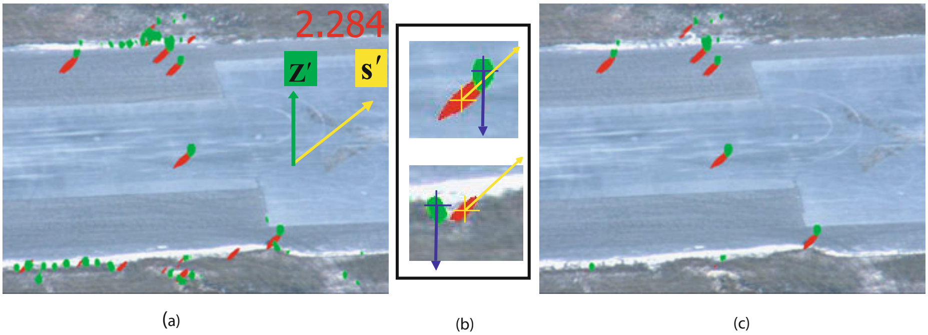


Fig. 4. (a) shows shadow blob map $M_{\mathbf{S}'}$ (shown in red), and normal blob map $M_{\mathbf{Z}'}$ (shown in green), overlayed on the original image. Notice there are false detections at the bottom of the image. Yellow arrow is the projected sun vector $\mathbf{S'}$, the projected normal vector $\mathbf{z'}$ is shown in green, and the ratio between the projected normal and shadow lengths is 2.284 (b) shows example candidates being refined. A valid configuration of human and shadow blobs (top) results in an intersection of the rays, and is kept as a human candidate. An invalid configuration of blobs (bottom) results in the divergence of the rays, and is removed from the set of human candidates. (c) shows refined blob maps after each normal blob was related to its corresponding shadow blob.

to remove shadow-human configurations that do not satisfy the image geometry which we derived from the metadata. We search every shadow blob, and try to pair it up with a potential object blob, if the shadow blob fails to match any object blobs, it is removed. If an object blob never gets assigned to a shadow blob it is also removed.

Given a shadow blob, $M^i_{\mathbf{S}'}$, we search in an area around the blob for a potential object blob $M^j_{\mathbf{Z}'}$. We allow one shadow blob to match to multiple normal blobs, but not vice versa,since the second case is not very likely to be observed. The search area is determined by major axis lengths of $M^i_{\mathbf{S}'}$ and $M^j_{\mathbf{Z}'}$. For any object candidate blob, $M^j_{\mathbf{Z}'}$ that falls within the search area, we ensure that it is in the proper geometric configuration relative to the shadow blob (see Figure 4 **(b)**) as follows. We make two line segments l^i, and l^j, each defined by two points as follows $l^i = \{c_i, c_i + Q\mathbf{S}'\}$, and $l^j = \{c_j, c_j - Q\mathbf{Z}'\}$. Where c_i, and c_j are centroids of shadow and object candidate blobs respectively, and Q is a large number. If the two line segments intersect, then the two blobs exhibit correct object shadow configuration.

We also check to see if the lengths of the major axes of $M^i_{\mathbf{S}'}$ and $M^j_{\mathbf{Z}'}$ conform to the projected ratio constraint η. If they do then we accept the configuration.

Depending on the orientation of the camera in the scene, it is possible for the person and shadow gradients to have the same orientation. In that case the shadow and object candidate blobs will merge, the amount of merging depends on the similarity of orientations $\mathbf{S}'$ and $\mathbf{Z}'$. Hence, we accept the shadow object pair if

$$\frac{M^i_{\mathbf{S}'} \cap M^j_{\mathbf{Z}'}}{M^i_{\mathbf{S}'} \cup M^j_{\mathbf{Z}'}} > q(1 - abs(\mathbf{S}' \cdot \mathbf{Z}')), \tag{11}$$

where q was determined empirically. For these cases the centroid of the person candidate blob is not on the person. Therefore for these cases we perform localization, where we obtain a new centroid by moving along the shadow vector $\mathbf{S}'$, as follows

$$\tilde{c} = c + \frac{m}{2}(1 - \frac{1}{\eta})\frac{\mathbf{S}'}{\|\mathbf{S}'\|}, \tag{12}$$

where m is the length of the major axis of shadow blob $M^i_{\mathbf{S}'}$.

3.3 Constraints without Metadata

Having all of the metadata, quickly provides a set of strict constraints for a variety of camera angles, and time of day. However, there may be cases when the metadata is either unavailable, or worse, is incorrect. In such cases it is acceptable to sacrifice some of the generality, and computation time to obtain a looser set of constraints that still perform well. Assuming that humans are vertical in the image, and ignoring the ratio between the size of humans and their shadows, we can still exploit the orientation of the shadow in the image, as well as the relationship between humans and their shadows, as described below.

We find the orientation of the shadow in the image in the following manner. We quantize the search space of shadow angle θ between 0° and 360°, in increments of d (we used 5 in our experiments). Keeping the normal orientation fixed, and ignoring shadow to normal ratio, we find all human candidates in image I for every orientation θ using technique described in sections 3.1 & 3.2 (see Figure 5). We track the candidates across different θ. Similar angles θ will detect the same human candidates. Therefore, each human candidate C_i has a set Θ_i for which it was detected, and a set O_i which is a binary vector, where each element corresponds to whether the shadow and human blobs overlapped. Then, the set of orientations for which it was detected due to overlap is Θ^o_i, and the set of orientations for which it was detected without overlap is $\Theta^{\bar{o}}_i$ (see Figure 5). We remove any candidate which has been detected over less than p orientations, since a human is always detected as a candidate if shadow and normal orientations are similar, and the resulting blobs overlap according to equation 11 (as in 5 (b) & (f)). Here p depends on quantization, we found that it should encompass at least 70°.

If there are two or more humans casting shadows on planes parallel to the ground plane (poles will work for the task as well), their orientations will be consistent. We find the optimal shadow orientation $\hat{\theta}$ by treating each $\Theta^{\bar{o}}_i$ as a sequence and then finding the longest common consecutive subsequence β among all $\Theta^{\bar{o}}$. Subsequence β must span at least 20° but no more than 40°. Finally, the optimal orientation $\hat{\theta} = mean(\beta)$. If we cannot find such a subsequence then there are either no shadows, or the orientation of the shadow is the same as the orientation of the normal, so we set $\hat{\theta}$ to our assumed normal. Figure 5

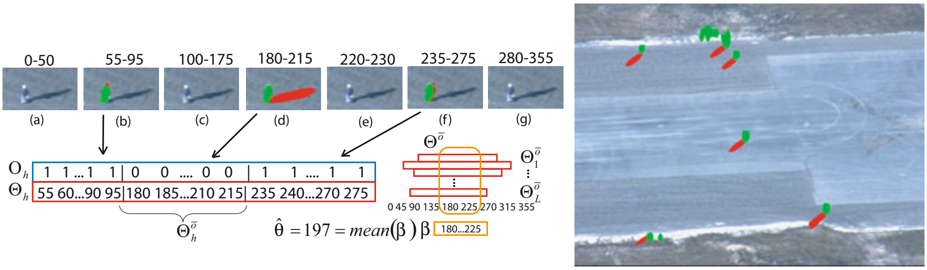


Fig. 5. (The flow chart shows our method for finding optimal shadow orientation for a given image in the absence of metadata. Top row shows human candidate responses obtained for different shadow orientations. A human candidate is then described by a vector of orientations for which it was detected, and a binary overlap vector. Optimal orientation $\hat{\theta}$ is the average of longest common consecutive non-overlapping subsequence of orientations among all human candidates. The image on the rights shows refined human candidate blobs for an automatically estimated shadow orientation of 35°, without metadata. Corresponding metadata derived value of θ for this frame is 46.7°. Blobs that were detected using metadata can be seen in fig. 4.

shows an example frame for which human candidates, were detected using the automatically estimated shadow orientation. There is a 10° difference between estimated orientation, and orientation derived from the metadata. This is the same frame as in Figure 4, qualitative examination of the shadow blobs, seems to indicate that the estimated orientation is more accurate than the one derived from the metadata, however the computation time of obtaining it is much larger. In practice this issue can be dealt with in the following manner. The angle can be estimated in the initial frame, and in subsequent frames it can be predicted and updated using a Kalman filter.

3.4 Object Candidate Classification

Wavelets have been shown to be useful in extracting distinguishing features from imagery. So in the final step of our method, we classify each object candidate as either a human or non-human using a combination of wavelet features and SVM (Figure 6). We chose wavelet features over HOG because we obtained higher classification rate on a validation set. We suspect that this is due to the fact that in the case of HOG, the small size of chips does not allow for the use of optimal overlapping grid parameters reported in [3], giving too coarse sampling. We apply Daubechies 2 wavelet filter to each chip, where the low-pass, and high-pass filters for a 1-D signal are defined as

$$\phi_1(x) = \sqrt{2}\sum_{k=0}^{3} c_k \phi_0(2x-k),\ \psi_1(x) = \sqrt{2}\sum_{k=0}^{3}(-1)^{k+1} c_{3-k}\phi_0(2x-k), \quad (13)$$

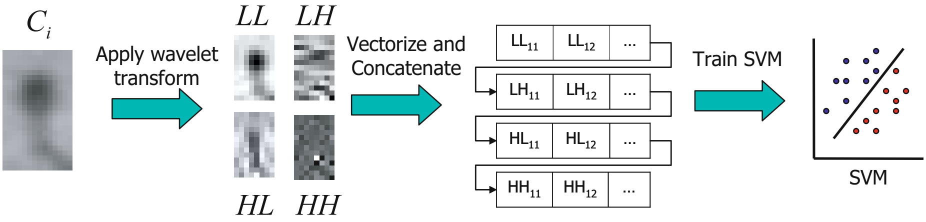


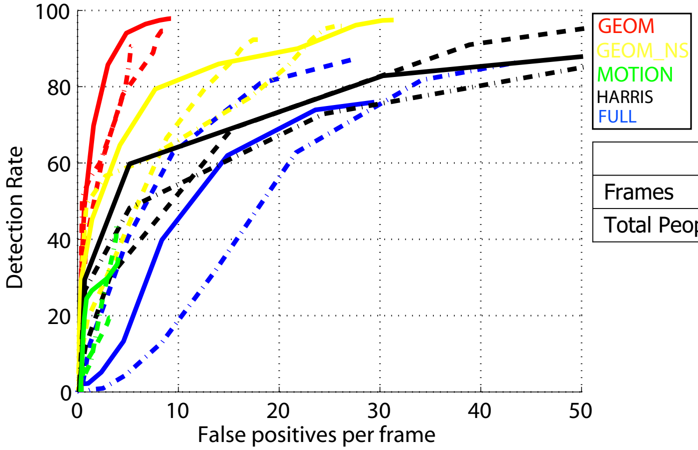
Fig. 6. Object candidate classification pipeline. Four wavelet filters (LL, LH, HL, HH) produce scaled version of original image, as well as gradient like features in horizontal vertical and diagonal directions. The resulting outputs are vectorized, normalized, and concatenated to form a feature vector. These feature vectors are classified using SVM.

here $c = (\frac{(1+\sqrt(3))}{4\sqrt(2)}, \frac{(3+\sqrt(3))}{4\sqrt(2)}, \frac{(3-\sqrt(3))}{4\sqrt(2)}, \frac{(1-\sqrt(3))}{4\sqrt(2)})$, are the Daubechies 2 wavelet coefficients, and ϕ_0 is either row or column of original image, and . In the case of the 2D image, the 1D filters are first applied along x, and then y directions. This gives to four outputs LL, LH, HL, HH. Where LL is a scaled version of the original image, and LH, HL, and HH, correspond to gradient like features along horizontal, vertical and diagonal directions. We used only one level, since adding more did not improve the performance. We vectorize the resulting outputs, normalize their values to be in the $[0, 1]$ range, and concatenate them into a single feature vector. We train a Support Vector Machine [15] on the resulting feature set using the RBF kernel. We use 2099 positive and 2217 negative examples $w \times h$: 14×24 pixels in size.

During the detection stage, we compute the centroid of the remaining object candidate blobs $M^i_{\mathbf{Z}'}$, extract a $w \times h$ chip around each centroid, extract wavelet features, and classify the resulting vector using SVM. If focal length data is available then the chip size could be selected automatically based on the magnitude, and orientation of the projected normal $|\mathbf{Z'}|$. Note, that this would amount to the use of absolute scale information, which would require a minor change in the geometry portion of the method to account for the effect of perspective distortion. The change amounts to computation of multiple shadow, and normal vector *magnitudes* for different regions of the image. However, since the sequences in the VIVID 3 dataset do not have correct focal length information, the size of the people in the images is approximately the same, and there is generally little perspective distortion in aerial video, we selected the $w \times h$ to be equal to the size of chips in the training set.

4 Results

We performed both qualitative and quantitative evaluation of the algorithm. Qualitative evaluation is shown on sequences from VIVID3 and 4 as well as some of our own data. The data contains both stationary and moving vehicles and people, as well as various clutter in the case of VIVID4. Vehicles cast a shadow, and



	Seq1	Seq2	Seq3
Frames	1191	1006	823
Total People	4892	2000	3098

Fig. 7. SVM confidence ROC curves for sequences 1 (dashed-dotted), 2 (dashed), and 3 (solid). Our Geometry based method with shadow, object-shadow relationship refinement, and centroid localization is shown in red. Yellow curves are for our geometry based method without the use of object-shadow relationship refinement, or centroid localization. A standard full frame detector (HOG) is shown in blue. Green shows results obtained from classifying blobs obtained through registration, motion, detection, and tracking, similar to [8]. Black curves are for our modified implementation of [9], which uses Harris corner tracks.

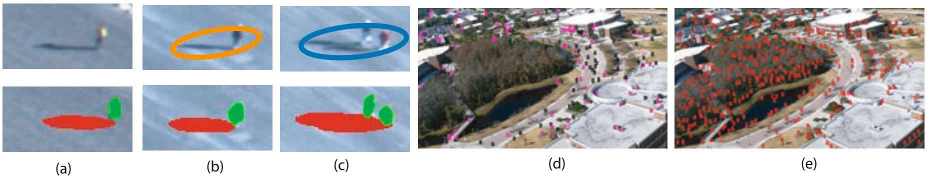


Fig. 8. (a) (b) and (c) compare motion detection (top row), and our geometry based method (bottom row). (a) Human is stationary and was not detected by the motion detector. (b) Moving blob includes shadow, the centroid of blob is not on the person. (c) Two moving blobs were merged by the tracker because of shadow overlap, centroid is not on either person. By contrast our method correctly detected and localized the human candidate (green). (d) and (e) compare geometry constrained human detection, and full frame HOG detection. Human candidates that were discarded by the wavelet classifier as clutter are shown in magenta, candidates that were classified as human are shown in **black**. Unconstrained full frame detection (e) generates many false positives.

are usually detected as candidates, these are currently filtered out in the classification stage, however we plan to extend the geometry method for vehicle detection as well. For quantitative evaluation we evaluated our detection methods on three sequences from the DARPA VIVID3 dataset of 640x480 resolution, and compared the detection against manually obtained groundtruth. We removed the frames where people congregated into groups. We used the following evaluation criteria *Recall* vs False Positives Per Frame (FPPF). Recall is defined as $\frac{TP}{TP+FN}$, where FN is number of false negatives, TP is the number of true positives in the frame. To evaluate the accuracy of the geometry based human candidate detector method, we require the centroid of the object candidate blob to be within w pixels of the centroid blob, where w is 15. We did not use the PASCAL measure of 50% bounding box overlap, since in our dataset the humans are much smaller, and make up a smaller percentage of the scene. In INRIA set inroduced in [3], an individual human makes up 6% of the image, in our case the human makes up about 0.1%. Under these circumstances small localization errors, result in large area overlap difference, hence we feel that the centroid distance measure is more meaningful for aerial data. Figure 7 compares ROC curves for our geometry based method with and without the use of object-shadow relationship refinement, and centroid localization, conventional full frame detection method (we used HOG detection binaries provided by the authors), and standard motion detection pipeline of registration, detection, and tracking. Figure 8 shows qualitative detection results. Conventional full frame detection is not only time consuming, (our MATLAB implementation takes several hours per 640x480 frame), but it also generates many false positives. By contrast preprocessing the image using geometric constraints to obtain human candidates, is not only much faster (6 seconds per frame), but gives far better results. Geometric constraints with the use of shadow based refinement, and centroid localization provide the best performance. However even without these additional steps, the geometric constraint based only on the projection of the normal still give superior results to full frame, as well as motion constrained detection. Motion based detection suffers from problems discussed in section 1, and shown in Figure 8. Which is why the green ROC curves in Figure 7 are very short. We implemented a part of [9] method, where instead of using the OT-Mach filter, we used our wavelet SVM combination for classification. These ROC curves are shown in black. We suspect that the poor performance is caused by poor tracking results. They simply used a greedy approach based on euclidian distance between the corners without any motion model. Therefore if a track contains corners belonging to both people and background, the 20% track length classification heuristic would introduce many false positives.

5 Conclusions

We proposed a novel method for detecting pedestrians in UAV surveillance imagery. This is a difficult problem due to very small size of humans in the image, and a large number of possible orientations. Our method takes advantage of the metadata information provided by the UAV platform to derive a series of geometric constraints, and to project them into the imagery. In cases when metadata is not available we proposed a method for estimating the constraints directly form image data. The constraints are then used to obtain candidate out of plane objects which are then classified as either human or non-human. We evaluated the method on challenging data from the VIVID 3 dataset, and obtained results superior to both full frame search, motion constrained detection, and Harris tracks constrained detection [9].

Acknowledgement

This research was funded in parts by Harris Corporation and US government.

References

1. Cheng, H., Butler, D., Basu, C.: ViTex: Video to tex and its application in aerial video surveillance. In: CVPR (2006)
2. Xiao, J., Cheng, H., Han, F., Sawhney, H.: Geo-spatial aerial video processing for scene understanding and object tracking. In: CVPR (2008)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1 (2005)
4. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
5. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: CVPR (2005)
6. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 69–82. Springer, Heidelberg (2004)
7. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: CVPR (2007)
8. Xiao, J., Yang, C., Han, F., Cheng, H.: Vehicle and person tracking in UAV videos. In: Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.) RT 2007 and CLEAR 2007. LNCS, vol. 4625, pp. 203–214. Springer, Heidelberg (2008)
9. Miller, A., Babenko, P., Hu, M., Shah, M.: Person tracking in UAV video. In: Stiefelhagen, R., Bowers, R., Fiscus, J.G. (eds.) RT 2007 and CLEAR 2007. LNCS, vol. 4625, pp. 215–220. Springer, Heidelberg (2008)
10. Bose, B., Grimson, E.: Improving object classification in far-field video. In: CVPR (2004)
11. Xu, L., Qi, F., Jiang, R.: Shadow removal from a single image. Intelligent Systems Design and Applications 2 (2006)
12. Finlayson, G., Hordley, S., Lu, C., Drew, M.: On the removal of shadows from images. IEEE PAMI 28 (2006)
13. Reda, I., Anreas, A.: Solar position algorithm for solar radiation applications. NREL Report No. TP-560-34302 (2003)
14. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004), ISBN: 0521540518
15. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001), Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Handling Urban Location Recognition as a 2D Homothetic Problem

Georges Baatz[1], Kevin Köser[1], David Chen[2],
Radek Grzeszczuk[3], and Marc Pollefeys[1]

[1] Department of Computer Science, ETH Zurich, Switzerland
{gbaatz,kevin.koeser,marc.pollefeys}@inf.ethz.ch
[2] Department of Electrical Engineering, Stanford University, Stanford, CA, USA
dmchen@stanford.edu
[3] Nokia Research at Palo Alto, CA, USA
radek.grzeszczuk@nokia.com

Abstract. We address the problem of large scale place-of-interest recognition in cell phone images of urban scenarios. Here, we go beyond what has been shown in earlier approaches by exploiting the nowadays often available 3D building information (e.g. from extruded floor plans) and massive street-view like image data for database creation. Exploiting vanishing points in query images and thus fully removing 3D rotation from the recognition problem allows then to simplify the feature invariance to a pure homothetic problem, which we show leaves more discriminative power in feature descriptors than classical SIFT. We rerank visual word based document queries using a fast stratified homothetic verification that is tailored for repetitive patterns like window grids on facades and in most cases boosts the correct document to top positions if it was in the short list. Since we exploit 3D building information, the approach finally outputs the camera pose in real world coordinates ready for augmenting the cell phone image with virtual 3D information. The whole system is demonstrated to outperform traditional approaches on city scale experiments for different sources of street-view like image data and a challenging set of cell phone images.

1 Introduction

In recent years, due to the ubiquitousness of cell phones and cameras, the demand for real-time localization and augmentation of virtual (3D) information arose and several systems have been proposed to solve the location recognition problem [3,1,2,6,8,9,10] or the closely related image retrieval problem [4,5,16,17,18]. A commonly used scheme that we also follow extracts local features (e.g. [12,11]) from a collection of reference images, vector-quantizes the feature descriptors to visual words and stores images as documents of these words in a database. Then for a query image techniques from web text search are applied to find the closest documents in the database, followed by a reranking of the result list based on geometric considerations.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 266–279, 2010.

We specifically look at the problem of place-of-interest recognition and camera pose estimation in urban scenarios, where we want to see how far we can get with visual information only. However, in contrast to general object recognition or image retrieval scenarios that cannot assume much about geometry and image content, we propose a tailored solution to the localization problem from cell phone images in a city. Here, often

- massive amounts of calibrated street level data are available for training[1]
- rough 3D city models exist[2]
- facades are planar and structures are vertically and horizontally aligned
- the camera's focal length is known approximately
- repetitive architectural elements appear that make 1-to-1 matching difficult

By projecting the offline training views to the surfaces, we can completely factorize out rotation from the recognition problem (in photometric matching and geometric verification). This enables the storage of gravity-aligned orthophotos (facade parts) in the database as opposed to densely sampling the space of all possible viewing poses. Query images can be transformed accordingly by finding the vertical and horizontal vanishing points of the given building. For recognition, matching and verification this reduces the problem to finding purely homothetic transformations, i.e. a scale and 2D offset on the building's surface. We show that this increases the discriminative power as compared to previous approaches on the one hand and allows to replace the computationally expensive RANSAC verification with a stratified homothetic parameter estimation, i.e. we perform three subsequent 1D estimates for distance, horizontal and vertical offset with respect to the building surface. Here the algorithm was designed in a way that e.g. window-to-window matches support the correct distance estimate through their scale ratio even if the match is from a different window instance on the facade's window grid. After having obtained the distance from the facade, horizontal and vertical offsets can be computed in the same way and we observe that using this reranking strategy is very effective in boosting the correct document to the first positions of the tested short list. As a side effect, we obtain the 6 DOF camera pose in absolute coordinates.

The key novel contributions are the orthophoto representation in the database allowing also for a more discriminative feature descriptor (upright SIFT), the homothetic verification scheme for repetitive structures and the exploitation of 3D building geometry so as to provide an absolute camera pose. In the next section we will relate the approach to previous work, before we go into details of the overall system and demonstrate its performance on different sources of cell phone and street level data.

[1] Nowadays several sources for image data taken from vehicles exist, e.g. Google's "Street View" or Microsoft's "Streetside". We use Earthmine's "3D street level imagery" for database creation and Navteq's "Enhanced 3D City Models" for testing.

[2] In this contribution we use extruded building outlines from Sanborn data, for more info see http://www.sanborn.com/products/citysets.asp

2 Previous Work

Location recognition at the city scale is closely related to image search and large scale object recognition for which a huge amount of previous work exist. A commonly used approach builds on top of the bag-of-features approach of [4] and the scalable vocabulary trees (SVT) of [5]. In the image retrieval scenario, usually the camera intrinsics and object geometry are unknown. It can therefore be difficult to find strong geometrical constraints for filtering the initial visual-word based results, although recent approaches look at (locally) consistent orientations and feature shapes [16,17,18] and exploit that pictures are usually not taken upside down. Location recognition approaches [9,8,6] usually know the intrinsic parameters of the camera, but do not exploit dense 3D models of the scene since these are difficult to obtain for larger environments.

The closest earlier works to ours are probably by Robertson and Cipolla [3], Wu et al. [2] and Schindler et al. [1]. The first one uses vanishing points, but works purely in 2D with local patch matching on a relatively small set of images (<100) and does not obtain 6 DOF pose in the city coordinate system since 3D information is missing. The concept of rectifying features according to vanishing points has been presented recently in [10], where the authors focused on single images. Exploiting 3D geometry has been proposed in [13] and [14], however these approaches require depth information for both images to be matched. Building on top of that, [2] uses 3D information from local reconstructions of streets of houses for database creation, but can only handle query images taken at fronto-parallel perspective relative to the building and cannot cope with out-of-plane rotations. In the field of systems using image data only [1] presented a large scale recognition system with impressive results also based upon a vocabulary tree. However, only 2D image data is used and in our experiments we show that in urban scenarios with mainly building facades 3D rotation invariant matching and recognition outperforms 2D methods. Another difference is that both of the two latter methods need RANSAC for geometric verification which can become inefficient with repetitive urban structures and high fractions of mismatches. In contrast we provide a simple stratified voting scheme for verification.

While the trend in the last years went towards building bigger and bigger databases and generating even synthetic views to sample the space of all possible points of view [6], we go into a different direction and represent only the building facades (upright orthophotos). An interesting effect of the technique is that it enables the usage of upright features, for which the feature orientation is obtained from vertical building axes, avoiding multiple descriptors for the same keypoint, avoiding potential bias of standard SIFT descriptors towards the bins of canonical orientations and allows distinguishing local structures differing by rotation. It has already been observed in face recognition [15] that exploiting the knowledge of aligned patches and reducing the invariance requirements can increase the recognition performance. Already for the SURF detector [11], rotation invariance could be disabled, however this was mainly motivated by performance reasons, while we show that leveraging rotation information helps recognition.

3 Offline Creation of the Recognition System

Data Acquisition and Selection. For creating the database we exploit two sources of information (see Figure 1):

- Calibrated image data: Panoramic images captured by a vehicle driving systematically through the streets. For each of these images camera position and orientation is known from GPS and sensor data.
- 2D Building floorplans as available from land registration or fire insurance companies as well as building heights. The 2D maps can be extruded to piecewise planar 3D models approximating the buildings (see Figure 1) and each of these buildings is assigned a place-of-interest ID.

For the dataset of San Francisco, panoramic images have been taken roughly every 10 meters and 14896 places of interest have been covered.

Sparse Representation of all Places-of-Interest of a City. Up to noise, resolution and model inaccuracies all panoramic images that see the same parts of a facade should give rise to the same descriptors, so there is a huge redundancy in the captured panoramic images. While it might be beneficial to fuse multiple views of the same features, we leave the optimal redundant sampling of the facades from multiple overlapping panoramas for future work. Instead we use the following strategy to obtain a close to minimal representation of the buildings:

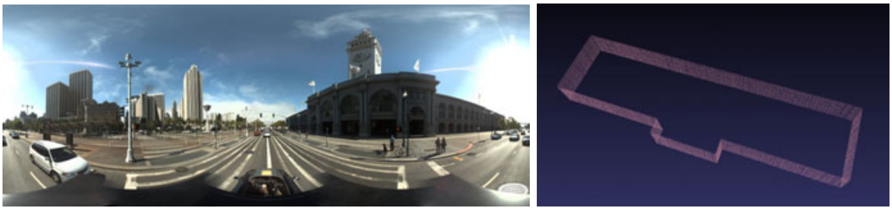


Fig. 1. Left: Panoramic image near the San Francisco Ferry Building grabbed by Vehicle. Right: Extruded building outline of Ferry Building.

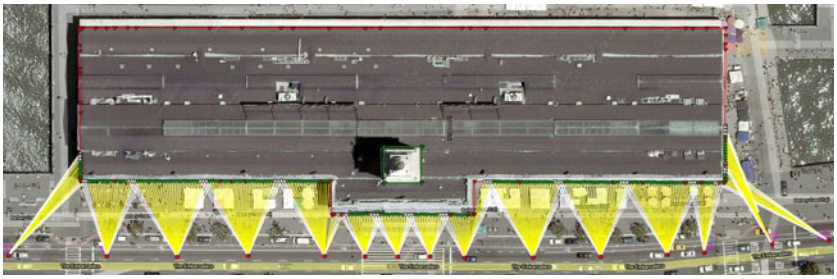


Fig. 2. Bird's eye view of Ferry Building. Portions of the panoramic images that are used to sparsely cover all facades of the POI are highlighted.

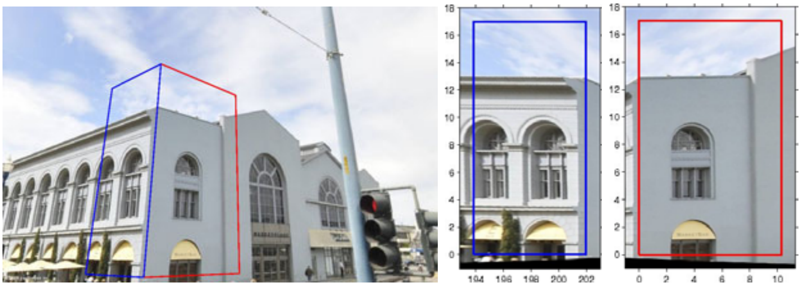


Fig. 3. Left: Building geometry projected into an image. Right: Two orhtophotos genarated from this image with overlaid geometry. The axes show the known scale in meters.

For each POI, we find the panoramic images within 50m distance to the building outline and extract perspective images with a 60° field of view every 20°. We prune those that look away from the POI or see it at a very oblique angle. The others are selected or rejected so as to represent all the POI surface subject to minimal overlap and maximal orthophoto resolution, when projecting the view onto the facade (see Figure 2). We obtain 58601 perspective images on the San Francisco dataset.

Geometric Rectification. Using the building height information we extrude the building outlines to 3D. We then project the reference images onto these 3D surfaces and render synthetic orthoviews. Since the scene geometry is roughly known for each of the calibrated panoramas, the image data can be projected onto the approximate geometry (see Figure 3). For each of the planar facade parts we generate orthophotos and use GPU-SIFT[3] to extract DoG keypoints and SIFT descriptors. Generally, for descriptor computation, previous approaches estimate keypoint orientations from the local gradient histogram. Rotating the local patch however in a way that the dominant peak is in the zero degree direction potentially makes the descriptors less discriminative, since all of them might have now significant mass in the zero degree descriptor bins and purely rotated local patches can no longer be distinguished. Instead, we project the gravity direction onto the facade and align the keypoints with this direction (upright SIFT). Effectively, by computing a gravity-compatible orthophoto, we remove all effects of 3D rotation and perspective from the image data[4]. Matching such features reduces the 6 DOF perspective recognition problem to a homothetic problem involving only scale and offset ambiguities in the 2D plane.

Scalable Vocabulary Tree Indexing. Based upon the extracted descriptors we use hierarchical k-means clustering to learn a vector quantization and build a visual vocabulary. We choose a random subset of 16M descriptors from the whole set of about 130M. We build a tree with the following parameters: split

[3] C. Wu: "SiftGPU" (Version 0.5.360) http://cs.unc.edu/ccwu/siftgpu

[4] Apart from image resolution issues due to interpolation.

factor $k = 10$, depth $d = 6$ which leads to one million leaf nodes. We then index the bags of features using an inverted file system (IFS) for fast retrieval.

4 Recognition of Places of Interest

Removing 3D Rotation Effects from Query Image. The incoming query image is assumed to come from a calibrated camera for which we expect to roughly know whether it was held more in landscape or in portrait orientation, so that we can correctly assign vanishing points to real-world directions. We detect line segments in the image using a method based on [19], estimate vanishing points as intersections of these lines, followed by a subsequent refinement step. Since the camera calibration is known, we can backproject the presumed vanishing points to rays in 3D space, which should be orthogonal. Every pair of points that does not fulfill this orthogonality constraint is no longer considered for rectification.

In case there are still multiple pairs of vanishing points left, we try to reduce the number of candidate pairs further. We estimate the importance of a plane by taking into account the number of lines on it and the closeness of lines corresponding to different vanishing points. We stretch the lines by 15% on both ends and then count the number of intersecting lines. For the plane with the highest number and all those within 95% of it, we generate an orthoview while discarding all the other planes.

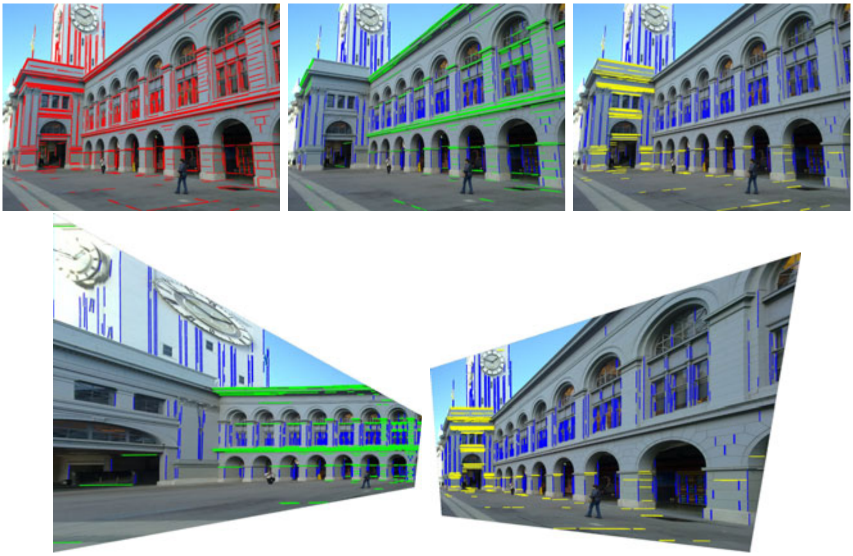


Fig. 4. Top row. Left: Query image with detected line segments. Middle and right: Lines belonging to the same vanishing point have been given the same color. Each image shows only the lines corresponding to one pair of orthogonal vanishing points. Bottom row: Two rectifications of the query image according to the two chosen pairs of vanishing points.

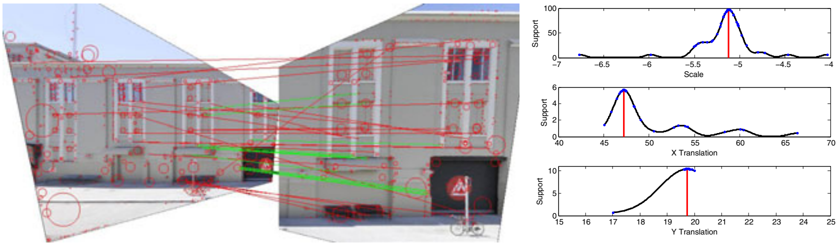


Fig. 5. Our voting scheme is illustrated using two images of Academy of Art University. Red circles indicate the scale of features, red lines are the raw correspondences and green lines are the final inliers. In the X Translation plot, note the secondary local maxima occurring at a 6m interval. They correspond to the repeating window structure. In the Y Translation plot, there is only one local maximum, since there is no vertical repetition. Also note that all but one scale inlier support the right y-offset, even though some of them vote for the wrong x-offset.

The vertical of the rectified images (see Figure 4) becomes the vanishing point (interpreted as a ray) which is closest to the known gravity vector. On these images, we then compute upright SIFT features which are used to query the vocabulary tree. The top 50 candidates are further examined by geometric verification.

Geometric Verification Voting Scheme. So far, ranking only used frequencies of visual words for POI identification. As usual, geometrical verification of the feature configurations can be used to improve the ranking. Unlike previous approaches, who usually perform RANSAC, we leverage the fact that we are solving a homothetic problem.

Since we are matching orthophotos, we may observe differences in scale and offset that translate to the camera distance and position with respect to the facade. First we observe that for all true correspondences $\{(S_{\mathrm{facade},j}, S_{\mathrm{query},j})\}$ the scale ratios $\rho_i := \sigma_{\mathrm{query},i}/\sigma_{\mathrm{facade},i}$ should be equal up to some tolerance. When swapping the roles of the images, the same argument applies for the inverse ratios, since the problem is symmetric. Consequently, we transfer it to the logarithmic domain, and require the differences of logarithmic scale ratios to agree up to a threshold $\log t$ that depends on the expected scale estimation uncertainty of the SIFT detector:

$$|\log \rho_i - \log \rho_j| \leq \log t\,. \tag{1}$$

In order to determine the scale ratio with the most support, we use a technique inspired by kernel density estimation [20]: every scale ratio contributes a Gaussian probability density function with mean $\log \rho_i$ and standard deviation $\log t$. We then consider the sum of all these contributions and find its maximum (more precisely, the argmax). All the datapoints within a certain distance (e.g. $2\log t$) are considered inliers.

Using the estimated scale ratio, we transform the feature coordinates of both images to a common scale. Since we know the true scale of the database image, we can have all the coordinates expressed in meters. Truly matching feature points now differ only by a global translation. The x and y components of this translation are estimated independently. We define the coordinate differences $\xi_i := x_{\mathrm{query},i} - x_{\mathrm{facade},i}$ and $\nu_i := y_{\mathrm{query},i} - y_{\mathrm{facade},i}$. As before, true correspondences should exhibit a consistent coordinate difference:

$$|\xi_i - \xi_j| \leq d \qquad \text{and} \qquad |\nu_i - \nu_j| \leq d\,. \tag{2}$$

Since all of the coordinates are expressed in terms of a known unit, we can again derive in a principled way a reasonable value for translation tolerance d, completely independently of image resolutions. We vote for x- and y-displacement separately using the same scheme as before (without transforming to log-space). The intersection of the two resulting inlier sets constitutes the final inlier set of the geometric verification (see Figure 5) and its cardinality is used to generate a new ranking of all the candidates under consideration.

This scheme has several advantages over previous approaches: RANSAC on top of an essential matrix, affine or projective transformation estimates 5, 6 or 8 parameters respectively. In contrast, our approach only needs to determine three degrees of freedom total, which means that the search space is smaller. On top of that, each degree of freedom is estimated separately further reducing the search space, which increases reliability and efficiency. In fact, we can afford exhaustively testing every hypothesis rather than sampling just some of them.

Every feature correspondence provides three constraints (scale, x- and y-coordinate). Thus, a single correspondence is enough to generate a complete hypothesis. Earlier, RANSAC-based approaches usually ignore scale and require outlier-free subsets of 5, 3 or 4 correspondences respectively. In order to hit such a set reliably, one needs to draw a number of samples which is essentially exponential in the number of required correspondences.

Finally, even wrong correspondences can still contain partial information about the solution. For instance, if one window in an image gets matched to the wrong window in the other image, this correspondence will likely vote for the right scale ratio and possibly for one correct coordinate.

Pose Estimation from 2D-2D Correspondences. Since we used vanishing points to rectify the original query image, we obtain the camera orientation with respect to the facade directly from the vanishing points. Since the rectified image plane is parallel to the facade, the only remaining parameters are those obtained in the previous section: Since we know the facade texture in meters the scale ratio can directly be used to compute a (perpendicular) distance $\mathrm{pos_z}$ of the camera from the facade. Assuming the camera is calibrated with focal length 1 pixel and principal point at zero, then

$$\mathrm{pos_z} = \mathrm{res_{facade}} \cdot \sigma_{\mathrm{facade}} / \sigma_{\mathrm{query}}, \tag{3}$$

where $\mathrm{res_{facade}}$ represents the resolution of the orthophoto in pixel/meter. The cell phone's $\mathrm{pos_x}$-offset (parallel to the facade) can directly be computed from the feature position

$$\mathrm{pos_x} = \mathrm{res_{facade}} \cdot (x_{\mathrm{facade}} - \sigma_{\mathrm{facade}}/\sigma_{\mathrm{query}} \cdot x_{\mathrm{query}}), \tag{4}$$

and $\mathrm{pos_y}$ in an analogous way. The local camera orientation with respect to the wall is simply the inverse vanishing point rotation. Finally, the relative coordinates with respect to the facade can be converted to absolute world coordinates using the facade's pose in the world.

5 Experiments

Upright SIFT versus Traditional SIFT. In order to test whether the SIFT descriptor's discriminative power improves if we do not rotate it into the dominant gradient orientation a simple experiment has been run (see Figure 6) on the image sequences for descriptor evaluation provided by [7]. Here we warp all 5 images of such a sequence to the first image, so that orientations are the same for corresponding SIFT keypoints.[5] Features at the same position ±50% feature size, same scale ±20% and same orientation ±30° are assumed to be a geometrical ground truth correspondence, other features are assumed to be not in correspondence. By comparing every descriptor of image 1 to every descriptor in the other images we generate the precision-recall diagram for the three sequences bark, wall and graffiti (see Figure 6) as has been done in [7]. In all of these sequences upright produces a significantly higher precision for a given recall fraction of the geometrical ground truth matches. A possible explanation is that when rotating the SIFT descriptor to the dominant orientation some gradient orientation histogram entries are more likely to obtain responses than others (e.g. those of the dominant orientation). This makes it more difficult to distinguish local regions that mostly differ by a rotation whereas this is possible using upright SIFT.

Vanishing Point Detection. For 31034 Earthmine images, we ran the vanishing point detection algorithm. In order to measure the error, we computed the angles between the directions that were found and the horizontals/verticals of known building surfaces. The distribution of these angles is shown in Figure 7. 75% of the time, the vanishing points are estimated correctly up to 2 degrees, the median error is 0.9°.

Recognition. Different variants of recognition pipelines are compared:

- **Affine.** This is our reference implementation. The SVT and IFS are trained and built on the raw survey images. As feature descriptor we use standard SIFT. For geometric verification we use the affine model.
- **Masked.** Same as before, except that for survey images we use geometric models to discard all features that do not lie on a building. This variant uses the same regions of the original images as the following variants. Its interest lies in testing how discarding background features affects recognition.

[5] For this experiment, we used A. Vedaldi and B. Fulkerson's vlfeat (v0.94 available from http://vlfeat.org) for detector and descriptor in this experiment.

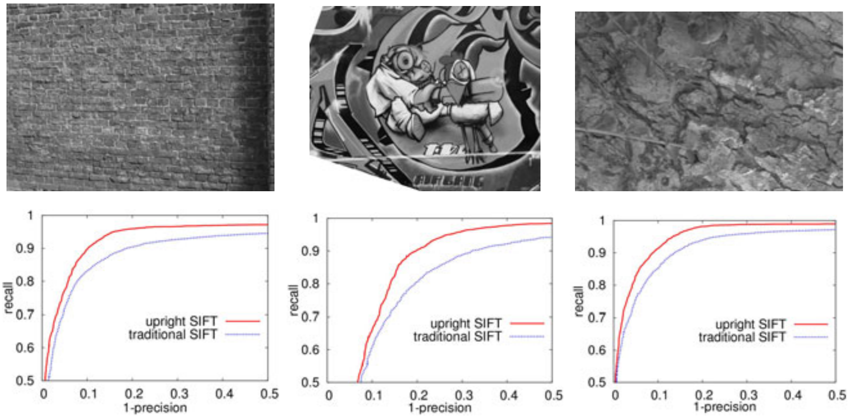


Fig. 6. Upright-SIFT vs. traditional SIFT with orientation estimation: All 5 images of the wall, graffiti and bark sequences [7] are warped to the first image of their sequence before DoG keypoints are extracted. We now compare the descriptiveness of upright-SIFT (with zero-orientation) and standard SIFT which estimates orientation from the local gradient histogram [12]. For a given precision (fraction of correct matches within all obtained matches) we get a higher recall rate (fraction of correct matches with respect to the set of geometrical ground truth correspondences.

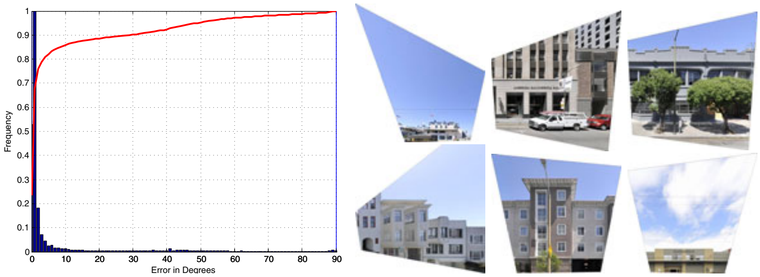


Fig. 7. Left: Histogram of orientation errors from vanishing points in degrees (blue) and cumulative curve (red), histogram scaled to the range [0, 1]. Right: Some rectified cell phone images.

– **Rectified.** Survey images are rectified using known 3D models of the buildings and query images are rectified using estimated vanishing points. The feature descriptor is still standard SIFT. Geometric verification is our proposed 3-degrees-of-freedom plane alignment using stratified histogram voting.
– **Upright.** Survey and query images are rectified as before, but in addition we use upright SIFT. Geometric verification is again 3DOF plane alignment.

We evaluated each of these four implementations on three different query sets:

– **Earthmine.** This dataset consists of 31,034 Earthmine images that were *not* selected for the training set. However, they stem from the same day and have been taken under the same conditions as the training set so that they must be considered as very easy. The images were automatically chosen such that they point towards a building. Whether or not this building is partially or completely occluded by vegetation was not a factor.

- **Navteq.** This dataset consists of 182 images, sampled at angles of 70° to 120° degrees (with respect to driving direction) and 0° to 20° (tilt) from panoramic image data from Navteq, where panoramic images have been chosen such that buildings could be seen reasonably well. This data has been taken more than one year later than the Earthmine training data and with different equipment.
- **Cellphone.** This dataset consists of 1180 images taken by various people with different camera phones (Nokia N95, N97, N900, N86) having between 5

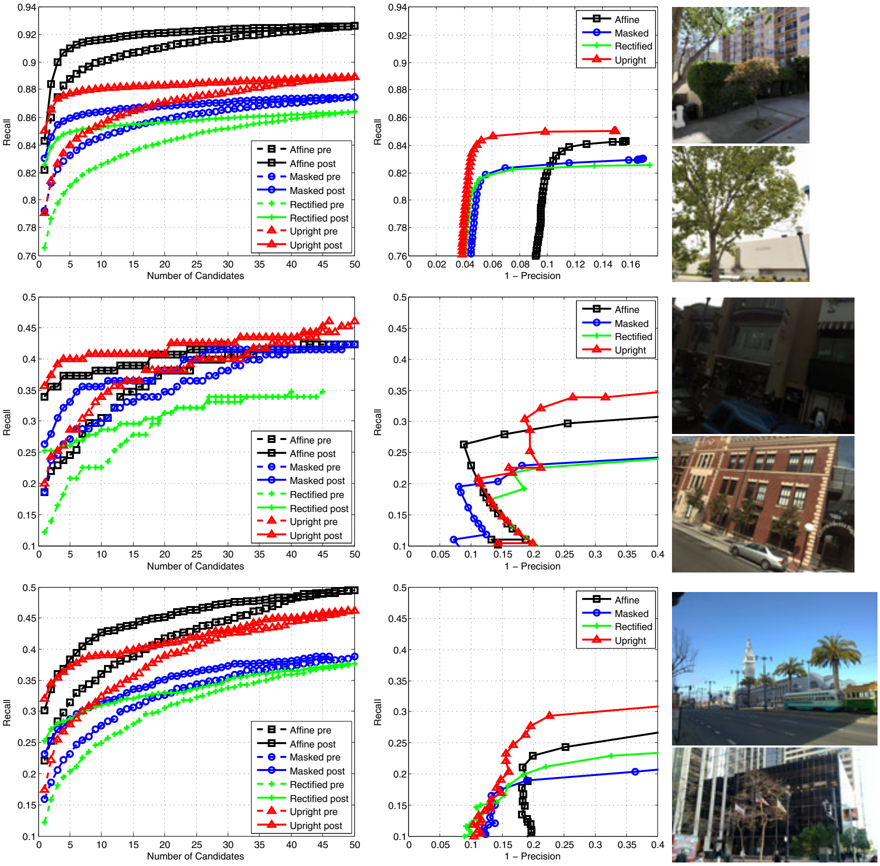


Fig. 8. Left column: Frequency of correct building being among top n candidates. Middle column: Precision-vs.-recall curve based on the number of inliers for accepting a candidate answer. Right column: Sample query images. Top row: Earthmine. Middle row: Navteq. Bottom row: Cellphone.

and 8 megapixel resolution. These images are from pedestrians' perspective partially under extreme angles and constitute the most challenging dataset.

We examined how frequently a correct building is returned as one of the top n candidates for n ranging from 1 to 50. This information was recorded for both the ranking before and after geometric verification and for all combinations of implementations and query sets. The results are shown in Figure 8. Since we are targeting augmented reality applications, we are mainly interested in the percentages for the top ranked image. These numbers are summarized in Table 1.

Table 1. Frequency of the top-ranked image being correct. For each dataset the best percentage has been highlighted.

	Affine	Masked	Rectified	Upright
Earthmine	84.3%	83.0%	82.6%	**85.0%**
Navteq	33.9%	26.3%	25.2%	**35.7%**
Cellphone	30.2%	23.2%	25.2%	**32.1%**

We observe that the performance is generally better on the Earthmine query set than on the other two, which is to be expected since these images come from the same source as the database images.

We notice that *Affine* generally outperforms *Masked*. The difference between the two is that the database for the former contains features from both buildings and surroundings, while the latter uses only features from buildings. This indicates that features from the surroundings help recognition rather than distract. This is probably the main reason why the pre-verification curves of the other two methods are lower than *Affine*. They suffer from the same disadvantage as *Masked*: having ignored the features from the surroundings.

With respect to the pre-verification curves, *Rectified* does slightly worse than *Masked*. On the other hand, the post-verification curve for *Rectified* is flatter. This means that rectifying the images may hurt performance in the SVT part, but it allows for a stronger geometric verification (3DOF homothetic vs. affine).

It also paves the way for using upright SIFT. As already stated before, upright SIFT is more discriminative because it can distinguish image patches that differ only by a rotation. We see that already the pre-verification curve for our proposed method (*Upright*) is higher than for *Masked* and *Rectified*. Combined with the strong 3DOF verification, it outperforms the other methods on all three datasets with respect to the top-ranked candidate (see Table 1). On top of that, this advantage gets bigger on the more challenging datasets.

We have seen that *Affine* has the highest pre-verification curve due to the inclusion of background features. Even though *Upright* is the better overall system, combining the advantages of both methods might yield even better results. We plan to address this in future work.

We also examined the precision-recall trade-off. The number of inliers for the top candidate is compared to a threshold. If the number is below, the system returns "no-answer", otherwise it returns the top candidate. By setting this threshold to lower values, one achieves a higher recall (how often a query gets a correct answer), but also lower precision (how often an answer is actually correct). By choosing a higher threshold these spurious matches can be reduced at the cost of losing some correct matches as well.

For all three query sets *Masked* and *Rectified* share a similar precision-recall curve with a better precision than *Affine*, but a worse recall. For the Earthmine and Cellphone datasets, *Upright* is clearly the better choice, while for Navteq it depends on how one wants to trade precision for recall.

6 Conclusion

We presented an approach for recognizing places of interest in cell phone images. By exploiting approximate 3D city models it was possible to convert street level data to an orthophoto-like representation of the facades of the city. In this representation also the gravity direction is known which enabled the use of upright SIFT features which have been proven more discriminative than classical SIFT on the standard feature descriptor test sets as well as in the location recognition pipeline. The given system can be seen as 3D rotation invariant matching and allowed for estimating homothetic transformations between a rectified cell phone image and a building facade, where the parameters scale and 2D offset of the homothetic transformation can be estimated separately. This allows for an efficient 1D voting scheme related to kernel density estimation and the resulting reranking has been shown to be very effective in boosting the true image to a top position in the reranked list.

Acknowledgments. The authors would like to thank Friedrich Fraundorfer for valuable and helpful discussions as well as Ramakrishna Vedantham and Sam Tsai for help with the software and database infrastructure. They also would like to thank the following peolple from Navteq: Bob Fernekes, Jeff Bach, Alwar Narayanan and from Earthmine: John Ristevski, Anthony Fassero.

References

1. Schindler, G., Brown, M., Szeliski, R.: City-Scale Location Recognition. In: CVPR 2007 (2007)
2. Wu, C., Fraundorfer, F., Frahm, J.-M., Pollefeys, M.: 3D model search and pose estimation from single images using VIP features. In: Workshop on Search in 3D, CVPR 2008 (2008)
3. Robertson, D., Cipolla, R.: An image based system for urban navigation. In: BMVC 2004 (2004)
4. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: ICCV 2003 (2003)
5. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR 2006 (2006)
6. Irschara, A., Zach, C., Frahm, J.-M., Bischof, H.: From structure-from-motion point clouds to fast location recognition. In: CVPR 2009 (2009)

7. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10) (2005)
8. Zhang, W., Kosecka, J.: Image based localization in urban environments. In: 3DPVT 2006 (2006)
9. Zhu, Z., Oskiper, T., Samarasekera, S., Kumar, R., Sawhney, H.S.: Real-time global localization with a pre-built visual landmark database. In: CVPR 2008 (2008)
10. Cao, Y., McDonald, J.: Viewpoint Invariant Features from Single Images using 3D Geometry. In: IEEE Workshop on Applications of Computer Vision 2009 (2009)
11. Bay, H., Ess, A., Tuytelaars, T., van Gool, L.: SURF: Speeded Up Robust Features. Computer Vision and Image Understanding 110(3) (2008)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2) (2004)
13. Köser, K., Koch, R.: Perspectively Invariant Normal Features. In: Workshop on 3D Representation for Recognition, ICCV 2007 (2007)
14. Wu, C., Clipp, B., Li, X., Frahm, J.-M., Pollefeys, M.: 3D Model Matching with Viewpoint Invariant Patches (VIPs). In: CVPR 2008 (2008)
15. Dreuw, P., Steingrube, P., Hanselmann, H., Ney, H.: SURF-Face: Face Recognition Under Viewpoint Consistency Constraints. In: BMVC 2009 (2009)
16. Jegou, H., Douze, M., Schmid, C.: Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
17. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object Retrieval with Large Vocabularies and Fast Spatial Matching. In: CVPR 2007 (2007)
18. Perdoch, M., Chum, O., Matas, J.: Efficient Representation of Local Geometry for Large Scale Object Retrieval. In: CVPR 2009 (2009)
19. Kosecka, J., Zhang, W.: Video Compass. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 476–490. Springer, Heidelberg (2002)
20. Bishop, C.M.: Pattern Recognition and Machine Learning, p. 123, Section 2.5.1 (2006) ISBN 0-387-31073-8

Recursive Coarse-to-Fine Localization for Fast Object Detection

Marco Pedersoli, Jordi Gonzàlez, Andrew D. Bagdanov, and Juan J. Villanueva

Dept. Ciències de la Computació & Centre de Visió per Computador,
Edifici O, Campus UAB 08193 Bellaterra (Cerdanyola) Barcelona, Spain
{marcopede,poal,bagdanov,juanjo}@cvc.uab.es

Abstract. Cascading techniques are commonly used to speed-up the scan of an image for object detection. However, cascades of detectors are slow to train due to the high number of detectors and corresponding thresholds to learn. Furthermore, they do not use any prior knowledge about the scene structure to decide where to focus the search. To handle these problems, we propose a new way to scan an image, where we couple a recursive coarse-to-fine refinement together with spatial constraints of the object location. For doing that we split an image into a set of uniformly distributed neighborhood regions, and for each of these we apply a local greedy search over feature resolutions. The neighborhood is defined as a scanning region that only one object can occupy. Therefore the best hypothesis is obtained as the location with maximum score and no thresholds are needed. We present an implementation of our method using a pyramid of HOG features and we evaluate it on two standard databases, VOC2007 and INRIA dataset. Results show that the Recursive Coarse-to-Fine Localization (RCFL) achieves a 12x speed-up compared to standard sliding windows. Compared with a cascade of multiple resolutions approach our method has slightly better performance in speed and Average-Precision. Furthermore, in contrast to cascading approach, the speed-up is independent of image conditions, the number of detected objects and clutter.

Keywords: Object Detection, Machine Learning, SVM.

1 Introduction

Many improvements and enhancements have been developed on object detection. However, the state of the art for detection is still far from the level necessary for real applications in terms of both speed and accuracy [1]. These two aspects are highly correlated: the newest and best performing methods for object detection, where multiple features [2,3,4], multiple and non-linear kernels [5,3] or deformable models [6] are employed, rely on high computational power. All these approaches are based on the concept of moving a classifier around over all possible scales and positions, scanning the image and searching for maximal detection responses, which is commonly called Sliding Windows (SW). However, standard SW is based on a brute-force approach.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 280–293, 2010.

Techniques to avoid the complete image scans have been proposed in the literature. In [7,8] the authors avoid a dense scan of the image by localizing the object as maximal response in a transformed space of local feature voting. Unfortunately this approach considers every feature independently from the rest, thus rendering it too sensitive to background clutter. Lampert et al. [9] proposed an interesting solution based on a branch-and-bound search over intervals representing bounding box positions and dimensions. However, the method depends on the existence and quality of the bound.

Other methods are able to speed-up SW scanning. A common approach is to decompose the base classifier into a cascade of rejecting classifiers, where the first one is fast but not very effective and the last is very accurate but computationally expensive. The first real-time classifier based on this strategy was proposed in [10], where a pedestrian is localized searching the best match in a hierarchy of human silhouette models. Cascades of classifiers are often based on Adaboost [11,12], where at each level of the cascade a new *strong* classifier is created by adding more and more *weak* classifiers. The main drawbacks of Adaboost cascades are the complexity of training, which can last for days on a standard PC due to the high number of detectors to be built, the selection of features and the learning of rejection thresholds for each cascade level.

For this reason, especially when dealing with large databases of images such as VOC2007 [13] or the INRIA person dataset [14], fast training methods are essential. Following this trend, the use of cascades based on SVMs with a hierarchy of features of increasing discriminative power [15] or with a hierarchy of kernels from linear to quasi-linear and non linear [3,5] has been proposed. These approaches, however, require computing all possible windows in the image, not use neither prior knowledge nor spatial constraints to guide the search, and require complex hierarchies of detectors.

In order to overcome the limitations mentioned above we present a method based on a single detector built on a multiresolution pyramid of dense features that benefits from spatial constraints to reduce the search space. In the search process, the image is divided into possible locations or neighborhoods, each one containing at most one object instance. Subsequently, for each location, a recursive coarse-to-fine localization refinement is applied based on the response of the detector at each resolution. The cost of a local search is thus reduced from linear (i.e. proportional to the number of possible locations) to logarithmic time.

Our method extends other contributions to object detection which also apply some kind of multiresolution strategy. The authors in [15] propose a cascade of detectors at different resolutions to speed up the sliding windows scan. Due to the use of multiple and separate detectors, they do not employ multiple resolution features in the same classifier, which reduces their discriminative power. In [16] the authors introduced a human detector based on the joint use of multiresolution gradient features. In this method the multi-resolution pyramid is used for better discriminative power but not for improving the search speed and object localization as in ours. The authors in [6] propose a 2-level dyadic pyramid for the object model: the first for the whole object and the second for parts. As

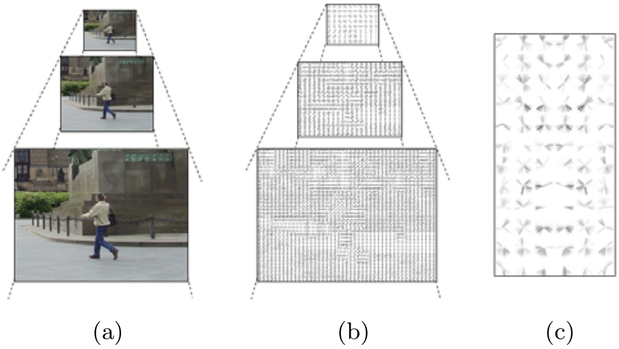


Fig. 1. Sliding window components: (a) Pyramid of images I_s: computed by repeated smoothing and sub-sampling of the original image. (b) Pyramid of features H_s: from every scale of the pyramid of images, the corresponding matrix of features is extracted. (c) Object model M: a $h \times w$ matrix of f-dimensional weight vectors.

in our method, no further feature computation is necessary because the same features are used for both multi-scale and multiresolution. But, in contrast to our method, they do not exploit local search to speed-up the scan process. Recently, the same authors propose in [17] a cascade algorithm for their detector. The method is similar to ours in the sense that it decomposes a single classifier into partial scores and uses these to prune hypotheses. However, their pruning method is based on thresholding object parts, while ours is based on recursive object localization refinements and so no threshold is necessary.

Our work uses features at different resolutions in the same classifier and exploits a greedy localization refinement to speed-up the image scan. Our implementation of the Recursive Coarse-To-Fine Localization (RCFL) based on HOGs does require no thresholds and runs twelve times faster than standard SW. In contrast to cascade approaches, the speed up is constant and independent of (i) the quality of the detector, (ii) the complexity of the image and (iii) the number of objects in the image.

2 The Image Scanning Approach

In this section we first describe the standard SW as a vectorial convolution between an object model and image features. Next, this formulation is extended to describe RCFL.

2.1 Sliding Windows

In SW, as described in [14], an object model is scanned over a pyramid of features representing an image. The pyramid of features is a set of matrices $H_s(x, y)$ (see Fig. 1 (b)), where each element is an f-dimensional feature vector. Each matrix

H_s is built from a smoothed and sub-sampled version $I_s(x, y)$ of the original image at a certain scale s, as shown in Fig. 1 (a). The object model for a linear classifier is an $h \times w$ matrix $M(x, y)$, where each element is an f-dimensional weight vector, as shown in Fig. 1 (c). The scale sampling of the pyramid of features is established by a parameter λ defining the number of levels in an octave, that is the number of levels we need to go down in the pyramid to get twice the feature resolution of the previous one.

The response D_s, or score, of the object model centered at position (x, y) and scale s is defined as:

$$D_s(x, y) = \sum_{\hat{x},\hat{y}} M(\hat{x}, \hat{y}) \cdot H_s(\hat{x} + x - w/2, \hat{y} + y - h/2), \tag{1}$$

where $\hat{x} \in \{0, 1, \dots, w - 1\}$, $\hat{y} \in \{0, 1, \dots, h - 1\}$. Note that the symbol $(- \cdot -)$ represents the scalar product because each element M_s and H_s are f-dimensional vectors. In this way, D_s is a pyramid of matrices of the same size as H_s, but where each element is a scalar that represents the response of the object model in the corresponding position and scale. Each element of $D_s(x, y)$ is converted to the corresponding image bounding box center $B_s(x, y)$:

$$B_s(x, y) = (2^{\frac{s}{\lambda}} kx, 2^{\frac{s}{\lambda}} ky) \tag{2}$$

$$\equiv k2^{\frac{s}{\lambda}}(x, y), \tag{3}$$

where k is the size of the feature at level $s = 0$ in pixels. For the sake of simplicity, in the following we will use the notation of Eq. (3), i.e. coordinate-wise scalar multiplications, as equivalent to notation in Eq. (2). Therefore, Eq. (3) describes SW in terms of image coordinates, which is more natural.

The same conversion of Eq. (2) is also applied for the bounding box size (w, h). In this way, we obtain all the necessary information to associate each score $D_s(x, y)$ with the corresponding image bounding box. Applying a Non-Maximum-Suppression (NMS) like in [18], we obtain the bounding box of the final detection.

2.2 Recursive Coarse-to-Fine Localization

In RCFL the object is searched in space but at different resolutions, from coarse to fine. The final score of the detector is now the sum of partial scores, one for each resolution. For this reason, the object model is a dyadic pyramid composed of l levels, where each level d is a matrix M_d of weight vectors. An example of a 3-level pyramid model for the class person in shown in Fig. 2, while an example of recursive localization refinement is shown in Fig. 3.

The computation of the partial score R_s^d for a resolution level d of the object model pyramid at a position (x, y) and scale s of the pyramid of features is then:

$$R_s^d(x, y) = \sum_{\hat{x}_d,\hat{y}_d} M_d(\hat{x}_d, \hat{y}_d) \cdot H_{s+\lambda d}(\hat{x}_d + (x - \frac{w}{2})2^d, \hat{y}_d + (y - \frac{h}{2})2^d), \tag{4}$$

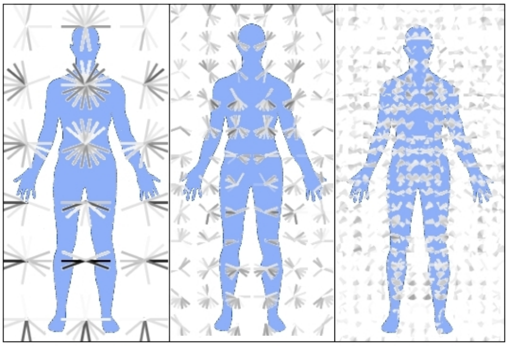


Fig. 2. HOG pyramid model M for the class person with $w = 3$, $h = 6$ and $l = 3$. The low resolution features ($d = 0$) give a general coarse representation of the human silhouette, while the high resolution ($d = 2$) focuses more on details.

where $\hat{x}_d \in \{0, 1, \ldots, w2^d - 1\}$, $\hat{y}_d \in \{0, 1, \ldots, h2^d - 1\}$. When $d = 0$ this is exactly Eq. (1). When the resolution level d is greater than 0, it is necessary to move-down λd levels in the feature pyramid to reach the corresponding resolution level. For each H_{s+d}, the search space is split into adjacent neighborhoods Δ_δ:

$$\Delta_\delta(x, y) = \{(\hat{x}, \hat{y}) | \hat{x} = x + d_x, \hat{y} = y + d_y\}, \tag{5}$$

where $d_x, d_y \in \{-\delta, -\delta + 1, \ldots, \delta - 1, \delta\}$ and δ is the radius of the neighborhood. The neighborhood represents all the locations where an object can be found. While in SW the number of hypotheses corresponds to the number of possible locations of the object, in RCFL the number of hypotheses corresponds to the number of neighborhoods. We define Π_s^0 for each (x, y) and scale s as the location that maximizes the partial score R_s^0 over the neighborhood Δ_δ :

$$\Pi_s^0(x, y) = \underset{(\hat{x},\hat{y}) \in \Delta_\delta(x,y)}{\arg\max} R_s^0(\hat{x}, \hat{y}). \tag{6}$$

Notice that (x, y) is the location of the center of the neighborhood at the coarse resolution at scale s, while Π_s^0 is the location of the object estimated by M_0. Since we optimize the score of $R_{0,s}$ over the neighborhood Δ_δ, it is not necessary to compute each (x, y). To select the correct sub-sampling of (x, y) is necessary that all locations be scanned at least once, which implies a sampling of $(\hat{\delta} x, \hat{\delta} y)$ with $\hat{\delta} \leq \delta$. The optimal position at levels $d > 0$ is recursively defined as a refinement of the position at $d - 1$:

$$\Pi_s^d(x, y) = \underset{(\hat{x},\hat{y}) \in \Delta_1(2\Pi_s^{d\quad 1}(x,y))}{\arg\max} R_s^d(\hat{x}, \hat{y}). \tag{7}$$

For $d > 0$ the neighborhood is fixed to Δ_1 because between the level d and $d + 1$ the feature resolution doubles and setting the maximum displacement to 1 allows

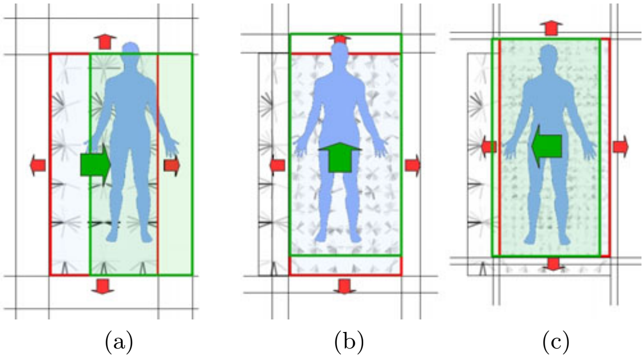

(a) (b) (c)

Fig. 3. Example of RCFL for detection. In (a), at a certain position (x, y) and scale s (red box) of the pyramid of features H, the best location $\Pi_s^0(x, y)$ (green box) for the low resolution model of the object M_0 is searched in the local neighborhood $\Delta_\delta(x, y)$. In (b), the same procedure is repeated for the next resolution level $s + \lambda d$, using as center of the neighborhood the best location computed at low resolution $\Pi_s^0(x, y)$. The process is recursively repeated for all feature resolution levels. In (c), the location obtained at the finest resolution $\Pi_s^2(x, y)$ is the location of the final detection and can be converted to pixels using Eq.(9).

refinement of the object model location at the new resolution. Recall our notational convention for coordinate-wise scalar multiplication, so that $2\Pi_s^{d-1}(x, y)$ represents a doubling of the coordinates for the object estimate at resolution $d - 1$. Knowing the optimal position of the object model at each level d, we calculate the total score $D_s(x, y)$ as:

$$D_s(x, y) = \sum_{\hat{d}} R_s^{\hat{d}}(\Pi_s^{\hat{d}}(x, y)), \tag{8}$$

where $\hat{d} = \{0, 1, \ldots, l - 1\}$. The computation of the bounding box of each score $D_s(x, y)$ is similar to the standard sliding windows. However, now (x, y) represents the location of the detection at the coarsest level. To obtain the location at the finest level it is necessary to convert the coordinates at Π_s^{l-1}. The center of the bounding box B, for the position (x, y) and scale s is thus:

$$B_s(x, y) = k2^{\frac{s+\lambda(l\quad 1)}{\lambda}} \Pi_s^{l-1}(x, y). \tag{9}$$

The final detection is computed like in normal SW by applying NMS.

3 Learning

Given a set of input data $\{x_1, , x_n\}$ and the associated labels $\{y_1, , y_n\}$, we find a parameter vector w of a function y that minimizes the regularized empirical risk:

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i y). \tag{10}$$

In our problem the input data x_i is a set of multiple resolution features (extracted from the pyramid H_s defined in previous section) associated to an image region, while the output data y_i is a binary label indicating whether the object is present in the region. The estimated output y depends on the relative position of each feature level with respect to the previous level. We introduce a structured latent variable h that is a vector of tuples (x_d, y_d) representing the relative position of a certain level d with respect to the previous $d-1$. Using latent variables allows us to obtain a better alignment of the object model with training data, which is useful to improve detector performance as shown in [6]. The estimated output is:

$$y = \max_h \langle w, f(x,h) \rangle, \tag{11}$$

where $f(x,h)$ is a function that maps the input features x to the corresponding latent variable h. In our case:

$$\langle w, f(x,h) \rangle = \sum_d R_s^d(x + x_d, y + y_d). \tag{12}$$

From Eq. (4) we see that w corresponds to the flattened version of M, our object model. Instead of computing the current maximum of f we compute the coarse-to-fine refinement approximation of this, which is Eq.(8):

$$\max_h \langle w, f(x,h) \rangle \approx D_s(\hat{x}, \hat{y}), \tag{13}$$

where $\hat{x}, \hat{y}, s$ corresponds to object location and scale at the lowest resolution. In contrast to normal SVM optimization, y is no longer linear in w, therefore the empirical risk is no longer convex, and standard optimization techniques can not be used. However$f$ is still convex in w since it is a maximum of linear functions. Thus, the empirical risk is is convex for $y_i = -1$ but concave for $y_i = 1$. In order to optimize this function we use a stochastic gradient descent, where learning is divided into two iterative steps: the optimization of w with h fixed for the positive examples and the estimation of the best h using the computed w [6].

Another problem of the learning is the number of negatives examples. While positive examples are costly to obtain and thus their number is always quite limited, the number of negative examples can be very high and can be obtained from images not containing the object to detect. This very high number of negative examples can help to boost performance, but it can make the learning process prohibitive in terms of time and memory. To solve this we use a cutting plane technique consisting of an iterative algorithm that first estimates w using a subset of the full training set and then selects the most violated constraints that will be added to the training set of the next estimation of w. This allows much faster learning and assures that the algorithm converges to the solution obtained with the full set of examples.

4 Discussion

RCFL scans the image in two ways at the same time. On one hand, it scans the image spatially, searching as a standard SW for the object. On the other hand, it scans the image in the resolution space, from coarse to fine. The number of hypotheses to scan is established at the coarsest level of the pyramid model as a set of neighborhood regions uniformly distributed over the image. Subsequent levels of the pyramid object model refine the hypotheses to the location with highest score inside each neighborhood.

In contrast to previous methods based on cascades [19,20,15,11], there is only one classifier to train. The only assumption required for the validity of the model is that the object has an appearance that can be modeled in a top-down manner. That is, global views of an object contain most of the relevant information needed to support reliable recognition [21], although specific details may be helpful for further refinement.

In cascade approaches, for each sliding window location the score of the classifier is used to evaluate whether to discard the location or continue to the next classifier of the cascade. This means that the score provided by a small number of features has to be precise enough to take a difficult choice that will greatly affect overall detector performance. Therefore, to obtain reliable decisions, the detector has to be conservative, discarding only a fraction of hypotheses. In contrast, our method does not require a binary decision about continuing in the search, but a localization decision about where to focus the search. This is much easier to take, because it is just about finding a local maxima in a small neighborhood of hypotheses. This is what allows RCFL to perform as fast as cascades without sacrificing accuracy and without any rejection threshold, as shown hereafter.

5 Implementation Details

Our aim is to evaluate the performance of the RCFL framework in terms of speed and performance. For this reason we implemented a RCFL based on HOG features, which are widely used in SW-based object detection [14,5]. However, the proposed framework is not limited to only HOG so it can be applied to any (and multiple) features.

Features. We used the HOG feature implementation proposed in [18]. The features for each square region are 31-dimensional: 9 contrast insensitive features, 18 contrast sensitive features and 4 representing the overall gradient of four neighbor regions. In contrast to the standard HOG proposed in [14], these features are not composed of 50% overlapping blocks, which saves memory space and simplifies the representation.

Object model definition. The object model has few parameters to tune. We set only the parameters of the lowest resolution object representation. All the rest follow from the use of a dyadic pyramid representation. The number of HOG features for the object representation is a trade-off between better discrimination (high number of HOGs) and the capability to detect small objects (low number of HOGs). The aspect ratio of the object model is chosen based on the mean aspect ratio of the training bounding boxes.

Positive examples. We convert an image containing positive examples into a pyramid of features (as described in section 2) and then search over space and scale for the best fit between the object model bounding box and the training example bounding box using the overlap ratio as defined in [13]. If the overlap o is greater than 0.5, the example is taken as a positive sample and added to T_p, otherwise it is discarded.

Negative examples. Negatives examples T_n are extracted from images not containing the object class. As for the positives, the image is converted to a pyramid of features. The examples are initially drawn using a uniform distribution over both space and scale. Subsequently, they are selected based on the cutting plane technique explained before.

SVM training. Positive T_p and negative T_n pyramids of features are flattened to vectors and used to train a linear SVM using libSVM [22]. The result of this is a weighted sum of support vectors. Since the kernel is lineal, these are summed up into a single vector of weights w. This is then converted back to the dyadic pyramid representation, resulting in our object model M.

6 Experiments

We evaluate our RCFL detector on two different and complementary databases: VOC2007 [1] and the INRIA person dataset [2]. We use VOC2007 as reference to evaluate our method on 20 different object classes and to obtain the most general detector configuration. Then, we test our method on the INRIA dataset, where many state-of-the-art methods are evaluated in terms of accuracy and speed.

6.1 Neighborhood Radius, Resolution Levels and Speed-Up Factor

The neighborhood radius δ and resolution levels l are the two most important parameters that influence the final performance of the detector. While for resolution levels greater than zero δ is forced to be 1 to ensure coherence of representation of the model over resolutions, for the zero level δ is free and greatly affects the speed-accuracy trade-off. Using a neighborhood of radius δ for level zero corresponds to scanning $q = (2\delta + 1)^2$ locations at the first level and subsequently 9 locations for the next levels. So, a model of l levels requires $q + 9l$ evaluations instead of $q4^l$ as in standard SW working at the finest level. However, the cost of scanning a location is proportional to the object model resolution which doubles at each level of the pyramid model. So, the ratio between the cost of brute-force search and the recursive localization approach is:

$$g(l, q) = \frac{q4^{l-1}}{\sum_d \frac{9}{4^d} + \frac{q}{4^{l-1}}} \tag{14}$$

where $d = \{0, 1, \ldots, l-2\}$. The computational cost of RCFL, compared to standard SW, is reduced proportionally to the number of levels of the object model l

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

[2] http://pascal.inrialpes.fr/data/human/

Table 1. Average-Precision computed on positive examples of train+validation of the VOC2007 database for different number of levels of resolution

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table
l=1	22.4	39.2	10.5	3.6	17.4	37.5	36.8	23.4	15.5	20.8	33.6
l=2	28.3	43.3	11.5	4.5	29.0	45.7	39.3	28.8	16.0	27.4	36.3
l=3	28.0	37.3	9.6	3.6	22.1	45.8	36.7	26.6	14.8	35.2	31.6

	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean	speed
l=1	19.2	45.4	36.5	23.6	16.2	19.3	33.3	26.5	44.7	26.3	1.0
l=2	24.7	42.9	38.0	22.1	16.3	27.7	34.1	31.3	47.7	29.7	3.2
l=3	26.7	43.9	37.7	21.5	15.1	27.2	30.6	28.2	46.0	28.1	12.2

and the neighborhood locations q. In experiments l is bounded by the resolutions available in images of the object and the memory space needed for the training. For the choice of δ we have to consider that a neighborhood must contain a unique hypothesis for an object. Therefore, to correctly localize partially overlapping bounding boxes it is necessary that, within a neighborhood, all possible detections overlap each other enough to be grouped together by NMS.

We computed the distribution of overlapping instances in the VOC2007 database for all classes and evaluated that a maximum overlapping of 0.2 assures fusing 99% of the object instances correctly. Limiting the minimum resolution for the lower resolution model to 15 HOG cells assures a minimum overlapping of 0.2 by setting $\delta \leq 1$.

6.2 Levels of Resolutions

We also must establish how many levels of feature resolution are best for our problem. For this, we evaluate the RCFL method against all classes of the PASCAL VOC2007 database. Because we are interested only on the relative performance of different configurations, we used only positive examples. In order to make the comparison fair, we choose the number of features for the maximum resolution level to be the same for all configurations.

Detection results are reported in Table 1. Mean values show that the best performance in terms of average-precision is the configuration with 2 resolution levels. However, the speed-up of this configuration is only 3.2 times. Moving to 3 resolution levels the performance is still good, but the speed-up is increased to 12.2 times. This makes this configuration an optimal trade-off between performance and speed and it will be the configuration used also in all the following experiments.

6.3 Comparison with Cascades

Although interesting methods implementing cascades based on HOG have been developed in recent years [11,15,17], all the methods use different HOG implementation, different parameter configurations and different learning strategies. We implemented our own cascade detector and to allow a best comparison we keep the same configuration based on three levels of feature resolution. The cascade is similar to [15], but we improve the learning strategy by joining all features from different levels into a single SVM optimization, exactly the same used for RCFL and

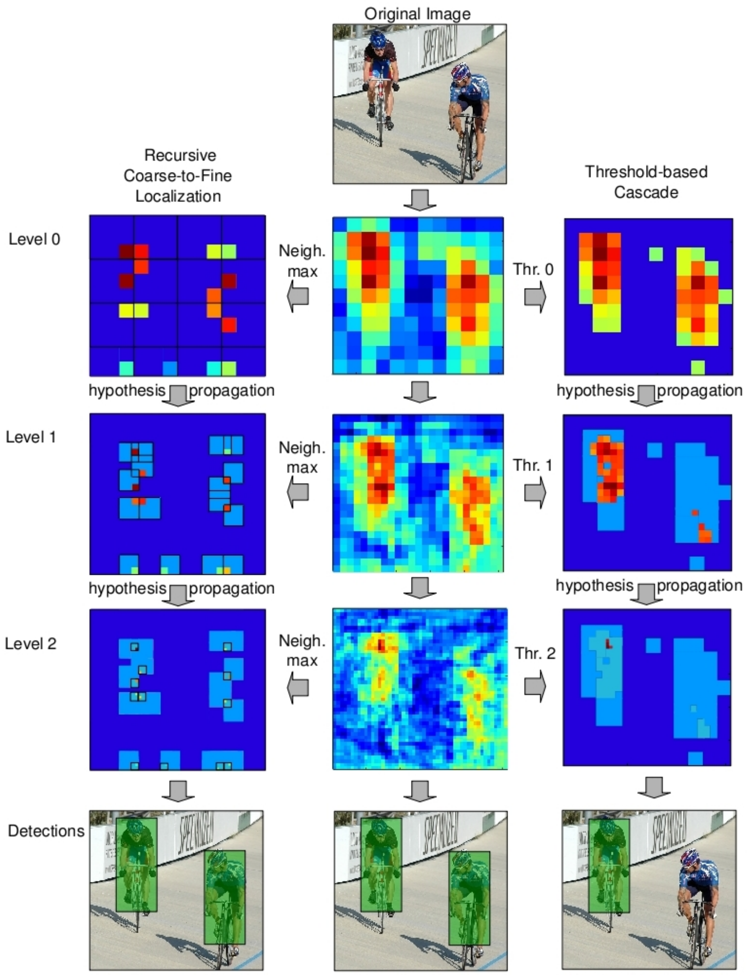


Fig. 4. Example of scan over resolutions with three methods: left column *RCFL*, central column *Exact*, right column *Cascade*. In the cascade the threshold is too high and prunes good hypothesis loosing the second cyclist. In RCFL both detections are made because the algorithm requires that hypotheses over all space are kept.

explained in section 3. Using same learning and features assures that changes in accuracy or speed are totally due to the method, not to implementation details nor different learning strategies. To select the optimal thresholds we used the method proposed in [17].

We compare the two methods also with a brute force approach in all classes of VOC2007. In this experiment we train the detectors only with the training set, while the validation set is used for threshold learning. The threshold value is set so the resulting precision-recall curve of the cascade detector should reach the

Table 2. Average-Precision computed on positive examples of train of the VOC2007 database. *Exact* shows the results of a brute force method; *Cascade* represents the result of a cascade method with thresholds chosen for obtaining the same performance as exact up to precision equals recall; thresholds are computed using the validation set of VOC2007; *Speed* is the average speed-up per class achieved for Cascade; *RCFL* is our method using three resolution levels.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table
Exact	24.1	41.3	11.3	3.9	20.8	36.8	35.4	25.5	16.0	19.4	21.2
Cascade	24.1	38.7	12.9	3.9	19.9	37.3	35.7	25.9	16.0	19.3	21.2
Speed	9.3	9.8	9.3	9.9	3.9	18.1	13.8	17.3	9.5	12.1	6.4
RCFL	23.6	39.4	12.9	2.7	19.7	39.2	34.5	25.9	17.0	21.6	23.1

	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean	speed
Exact	23.0	42.9	39.8	24.9	14.6	14.3	33.0	22.8	37.4	25.4	1.0
Cascade	23.0	40.2	41.5	24.9	14.6	15.1	33.2	23.0	42.2	25.6	10.9
Speed	3.3	17.6	20.1	3.6	6.4	19.0	15.0	9.8	2.8		
RCFL	24.1	42.0	41.1	25.3	14.2	15.8	29.6	22.5	41.0	25.8	12.2

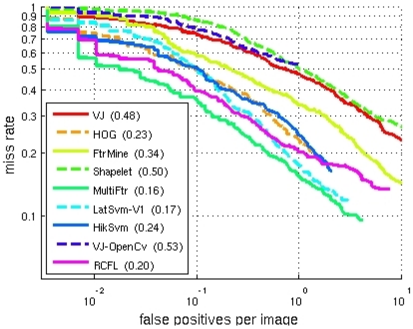
precision-equals-recall point without any pruning. Results are reported in Table 2. For the cascade we also report per-class speed-up, while the final speed-up is the average of all classes. It is interesting to notice that both speed-up methods, not only improve speed, but also in some case average-precision. This is due to the pruning of false positives. Notice that even without any threshold expressly tuned for it, RCFL obtain an average performance slightly better than the cascade and more important, the speed-up is constant, while for cascades it can vary a lot and it is unpredictable. This demonstrates that recursive localization is a suitable strategy for pruning hypotheses because (i) it obtains same or better performance than cascades in most of the classes (ii) it assures that speed does not depend on object class or image which is very important for real-time applications (iii) no threshold computation is necessary.

Fig. 4 show the pipeline of the pruning strategy of RCFL and Cascade of the class person for a certain scale. Although both strategies use exactly the same observations (central column), in this example Cascade is not able to detect an obect due to a too low partial score in the second level. RCFL is not affected by this problem because no thresholds are used for the pruning which is done in a spatial manner.

6.4 INRIA Pedestrian Dataset

The INRIA database is the standard benchmark for human detection [14]. Evaluation is done on a per-image basis. This is equivalent to a precision recall curve, but for certain tasks it is preferred because it gives an absolute measure of the average number of false positives to expect per-image (FPPI).

A comparison of RCFL with other methods is shown in Fig. 5 (a). The configuration of the detector is the same as in previous experiments with 3 resolution levels. RCFL reduces the standard HOG miss-rate by 3 points at 10^0 FPPI, by 10 points at 10^{-1} FPPI and by 14 points at 10^{-2} FPPI. Globally, two methods



(a)

Method	Features	Classifier	M-R	Time
HikSvm[16]	HOG-like	HIK SVN	0.24	140.0
Shapelet[23]	Gradients	AdaBoost	0.50	60.0
FtrMine[12]	Haar	AdaBoost	0.34	45.0
MultiFtr[4]	HOG+Haar	AdaBoost	**0.16**	18.9
HOG[14]	HOG	lin. SVN	0.23	13.3
LatSvm[6]	HOG	lat. SVM	0.17	6.3
Haar[19]	Haar	AdaBoost	0.48	7.0
RCFL	HOG	lin. SVM	0.20	**1.2**

(b)

Fig. 5. (a) False Positive Per-Image in the INRIA database. All curves but RCFL HOG are drawn using the data provided by [1]. (b) Comparison of different pedestrian detectors [1]. *M-R* represents the miss-rate at 10^0 false positive per-image. *Time* represents the seconds to compute an image of 640 × 480 pixels. RCFL reduces the miss-rate of the HOG detector performing much faster than any other method.

perform better than RCFL. However, *MultiFtr* uses multiple and complementary features to improve the HOG results while *LatSvm* learns the object deformations using latent SVM.

Table 5 (b) summarizes the main characteristics of each method. HOG RCFL performs better than all comparable methods, but with a higher speed. Our method takes 1.2 seconds to process an image: around 1 second is used for feature computation and only 0.2 for the scan. In contrast to most methods, where the most significant part of the time is used for scanning, with RCFL this scanning time is reduced to a small fraction.

7 Conclusions

In this paper we introduced a new method to speed-up object detection. The method join prior information about the search of object hypotheses with a coarse-to-fine localization to optimally distribute the computation necessary to detect objects. Compared to cascades approaches, our method obtains similar detection performance, assures a constant speed-up independent of object class and image conditions and do not need any threshold to prune hypotheses. Finally, the great generality of the idea behind RCFL allows it to be applied to most of current object detector methods: from deformable models [6] to bag of words pyramids [5], but also multiple features [3].

Acknowledgements. This work has been supported by the Spanish Research Programs Consolider-Ingenio 2010:MIPRCV (CSD200700018) and Avanza I+D ViCoMo (TSI-020400-2009-133); and by the Spanish projects TIN2009-14501-C02-01 and TIN2009-14501-C02-02. Thanks to Ivan Huerta for his precious help.

References

1. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR (2009)
2. Schwartz, W.R., Kembhavi, A., Harwood, D., Davis, L.S.: Human detection using partial least squares analysis. In: ICCV (2009)
3. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV (2009)
4. 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)
5. Harzallah, H., Jurie, F., Schmid, C.: Combining efficient object localization and image classification. In: ICCV (2009)
6. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR (2008)
7. Mikolajczyk, K., Leibe, B., Schiele, B.: Multiple object class detection with a generative model. In: CVPR, pp. 26–36 (2006)
8. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV 77, 259–289 (2008)
9. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR (2008)
10. Gavrila, D., Philomin, V.: Real-time object detection for smart vehicles. In: ICCV, pp. 87–93 (1999)
11. Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. In: CVPR, pp. 1491–1498 (2006)
12. Dollar, P., Tu, Z., Tao, H., Belongie, S.: Feature mining for image classification. In: CVPR (2007)
13. Everingham, M., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge, VOC 2007 Results (2007)
14. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
15. Zhang, W., Zelinsky, G., Samaras, D.: Real-time accurate object detection using multiple resolutions. In: ICCV (2007)
16. Maji, S., Berg, A., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR (2008)
17. Felzenszwalb, P., Girshick, R., McAllester, D.: Cascade object detection with deformable parts models. In: CVPR (2010)
18. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI 31 (2009)
19. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
20. 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)
21. Torralba, A.: How many pixels make an image? Visual Neuroscience 26, 123–131 (2009)
22. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2005)
23. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: CVPR (2007)

A Local Bag-of-Features Model for Large-Scale Object Retrieval

Zhe Lin and Jonathan Brandt

Adobe Systems, Inc.
{zlin,jbrandt}@adobe.com

Abstract. The so-called bag-of-features (BoF) representation for images is by now well-established in the context of large scale image and video retrieval. The BoF framework typically ranks database image according to a metric on the global histograms of the query and database images, respectively. Ranking based on global histograms has the advantage of being scalable with respect to the number of database images, but at the cost of reduced retrieval precision when the object of interest is small. Additionally, computationally intensive post-processing (such as RANSAC) is typically required to locate the object of interest in the retrieved images. To address these shortcomings, we propose a generalization of the global BoF framework to support scalable local matching. Specifically, we propose an efficient and accurate algorithm to accomplish local histogram matching and object localization simultaneously. The generalization is to represent each database image as a family of histograms that depend functionally on a bounding rectangle. Integral with the image retrieval process, we identify bounding rectangles whose histograms optimize query relevance, and rank the images accordingly. Through this localization scheme, we impose a weak spatial consistency constraint with low computational overhead. We validate our approach on two public image retrieval benchmarks: the University of Kentucky data set and the Oxford Building data set. Experiments show that our approach significantly improves on BoF-based retrieval, without requiring computationally expensive post-processing.

1 Introduction

We address the problem of retrieving images containing an object of interest, specified by a visual query, from a large image database. We are interested not only in ranking the database images but also locating the relevant objects in the top matching images.

Perhaps the most common and effective approach to large-scale image retrieval is the bag-of-features (BoF) framework (see, for example [15,11,13,2]). The BoF representation for an image is a global histogram of visual word occurrences where each "visual word" is a quantized local feature descriptor. The set of all possible visual words, or visual vocabulary, is learnt via various clustering algorithms, such as k-means [15,2], hierarchical k-means (HKM) [11], and approximate k-means (AKM) [13].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 294–308, 2010.

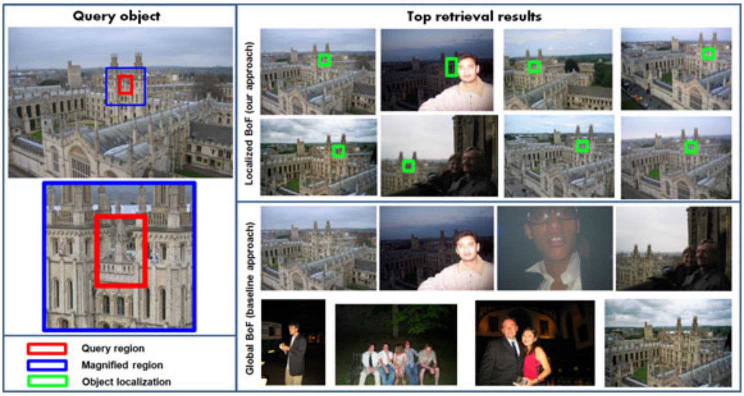


Fig. 1. An example of small object retrieval. Left: a query image with a region of interest. Right: the top 8 retrieved images using our approach and the baseline Global BoF approach. Red rectangle represents the query object of interest, and Green rectangles represent the returned object bounding boxes using our approach. In contrast, the baseline method cannot return bounding boxes and need additional totally different criterion (*e.g.* RANSAC) to localize objects.

Large vocabularies, typically containing a million or more visual words, tend to be more discriminative therefore more effective in locating specific objects. The large vocabulary size results in sparse histograms for particular images which can be efficiently represented and searched using inverted files [15].

Since the BoF representation contains no spatial information, post-processing to verify the spatial consistency of the retrieved images tends to improve retrieval accuracy provided the underlying spatial model is appropriate. Approaches to spatial verification include spatial neighborhood counting [15], as well as RANSAC-based spatial matching [13].

Retrieval based on the global BoF representation, although being very scalable, has the shortcoming that objects become difficult to retrieve as the amount of surrounding clutter in an image increases. That is, small objects are hard to find using a global histogram representation. Post-processing based spatial verification partially addresses this, but with an added computational cost that effectively limits the total number of images to be considered for post-processing. We propose a generalization of the global BoF framework to support scalable local matching without post-processing. By *local matching*, we mean matching the query to a locally bounded BoF, as opposed to a global BoF, which limits the effect of clutter, and also localizes the object. The generalization is to represent each database image as a family of histograms that depend functionally on a bounding rectangle. Integral with the image retrieval process, we identify bounding rectangles whose consequent local histograms optimize query relevance, and rank the images accordingly.

Ideally, we aim to localize the best region (namely, the one that has the maximum similarity to the query) in all the database images. As a simplification, we constrain our problem to the set of all possible subrectangles. Each image is therefore represented as a BoF histogram parameterized by a subrectangle. We can certainly go beyond rectangles through a post refinement process as in [19].

In order to maintain scalability, we use a spatial quantization-based indexing mechanism to compute sparse feature energies (norms) offline, and compute similarities over a coarse grid of rectangles to the query online. Integral images, enabled by a binary approximation of the BoF model, allow the localized similarities to be computed efficiently, and a full BoF comparison is done for the final ranking. In this way, we are able to match a query BoF against a broad set of sub-rectangle BoFs for each of the database images.

An example of the effectiveness of our algorithm for small object retrieval is shown in Fig. 1, where our approach returns more consistent results than traditional global BoF methods, and can localize objects simultaneously.

2 Related Work

Most common approaches to object localization in the BoF retrieval framework include neighborhood counting [15], and RANSAC-based methods [5,13]. Neighborhood counting uses the total number of neighboring word correspondences to rerank images. It is largely dependent on the size of the neighborhood and cannot capture spatial relationship in wider configurations. The RANSAC-based approaches can capture wider spatial consistency but are typically limited to near planar objects in order to avoid an overly complex spatial model, and are applied only to top hundreds of images [13, 2] due to RANSAC's computation complexity. During re-ranking, RANSAC-based verification computes similarities as the number of inliers, which is very different from the ranking criterion used in the first-phase BoF retrieval process (*i.e.* BoF similarity).

There have been approaches grouping pairs of or multiple local features in a larger spatial neighborhood as a new 'feature', *e.g.* the geometric min-Hash [1], bundling features [17], and multi-samples [18], to increase feature discriminative power. These approaches capture visual word co-ocurrence information in an early stage of retrieval, but still need an additional post-processing to localize objects in the top retrieved images.

BoF matching can also be formulated as a voting framework [2], where each matching pair of features between query and database will generate a vote (score) to be accumulated to query-to-database image distances. A fast weak geometric consistency scheme is introduced in [2] by voting for rotation angles and log-scale ratios during the first-phase retrieval process, but this model does not provide localization capability and cannot be easily extended to localize objects in arbitrarily rotated images.

We propose a local BoF model to simultaneously rank images and localize relevant objects under arbitrary rotations, significantly different viewpoints, and in the presence of clutter. By localized BoF matching, the model encodes weak spatial constraints implicitly during the retrieval process to improve the ranking accuracy and localize objects simultaneously. The model is fully integrated with an inverted file-based search to support large-scale object search and localization.

The localization process uses a simple greedy optimization method due to the potentially large scale nature of our problem. Although it is not guaranteed to find the global optimum, we have found through experiment that the retrieval accuracy of our approach is nearly identical to the result obtained by exhaustive search, as can be seen in Table 1. Also, our optimization method could be replaced with branch-and-bound [7] to guarantee a global optimum, but we found the added computational expense to be unnecessary.

Our approach is closely related to [6] in formulating the problem as a combination of image retrieval and object localization. Lampert [6] applied the branch-and-bound search to problems of subimage retrieval in large image and video sets, but the approach differs from our method in that it does not leverage the fast inverted file for localization. In this way our method is more scalable than [6].

In the context of a complete retrieval systems, our method can be regarded either as an improved BoF matching for the first-phase retrieval process [11], or as an improved weak spatial verification alternative to RANSAC-based schemes. Our contributions are three-fold:

1. A computationally efficient local BoF model for re-ranking database images and localizing objects in a large number images.
2. A local spatial pyramid model for combining the local BoF model and the spatial pyramid-based representation.
3. An efficient, integrated system for local BoF model and inverted file index for large scale object retrieval.

3 Local BoF Retrieval

3.1 Global BoF Model

The BoF representation begins with detection of local image features and extraction of each of the features as high dimensional descriptors $f \in \mathcal{R}^n$. Each of the descriptors is quantized according to a quantization function, $\mathcal{C} : \mathcal{R}^n \rightarrow \{1, 2, 3, \ldots, V\}$, to generate a set of "visual words" representing the image. The global BoF representation for an image is the normalized histogram of visual words, typically using either the L_1 or the L_2 norm, with components weighted by term frequency-inverse document frequency (TF-IDF). (Term frequency (TF) $\tau(i)$ is defined as the number of occurrences of word i in an image, and the inverse document frequency (IDF) α_i is defined as $\alpha_i = \log \frac{N}{N_i}$, where N is the total number of images and N_i is the number of images containing the word i.)

Let q denote the BoF for a query image and d denote the BoF for a database image. The relevance of d to query q is the distance, $D(q, d) = \|q - d\|_p^p$, where $p \in \{1, 2\}$ [11, 13, 2]. For search, the database images are ranked in ascending order of the distance to the query.

Since the BoF becomes very sparse when the vocabulary is large, the distance D can be evaluated efficiently by considering only the non-zero elements of q and d. For the L_2 norm, the simplification is as follows [15]:

$$D(q,d) = \|q-d\|_2^2 = 2 - 2 \sum_{i|q_i \neq 0 \wedge d_i \neq 0} q_i d_i. \tag{1}$$

We define the L_2 norm-based BoF similarity as:

$$S(q,d) := \sum_{i|q_i \neq 0 \wedge d_i \neq 0} q_i d_i. \tag{2}$$

In case of L_1 norm, the simplification is as follows (see [11] for the derivation):

$$D(q,d) = \|q-d\|_1^1 = 2 - \sum_{i|q_i \neq 0 \wedge d_i \neq 0} \left(q_i + d_i - |q_i - d_i|\right). \tag{3}$$

We define the L_1 norm-based BoF similarity as:

$$S(q,d) := \sum_{i|q_i \neq 0 \wedge d_i \neq 0} \left(q_i + d_i - |q_i - d_i|\right). \tag{4}$$

In any case, the search relevance of a database image I_d to a query image I_q is the BoF similarity, $\mathcal{S}(I_q, I_d) = S(q,d)$.

3.2 Local BoF Model

We can extend the global BoF model (denoted Global BoF) to a local model (Local BoF) by introducing a parameterization on the database BoF representation d. Specifically, let the Local BOF representation be a function $d(R)$ of a rectangle $R \in \mathbf{R}$, where R is parameterized by its bounding top/bottom/left/right image coordinates (t, b, l, r). $\mathbf{R}$ denotes the set of all subrectangles in an image. That is, for any database image, and for any subrectangle of the image, $d(R)$ is the normalized histogram of visual words occurring inside the subrectangle.

We define the image similarity as the global maximum of BoF similarity over the set of all possible subrectangles for the image.

$$\mathcal{S}(I_q, I_d) = \max_{R \in \mathbf{R}} S(q, d(R)), \tag{5}$$

$$R^*(I_d) = \arg\max_{R \in \mathbf{R}} S(q, d(R)), \tag{6}$$

where $S(q, d(R))$ is the localized object similarity, and $R^*(I_d)$ is the detected bounding box for image I_d. Note that R^* is not unique in general. We take a smallest one among the set of rectangles of equal similarity value.

We can solve the above problem by brute force simply by evaluating the similarity for all possible rectangles in all images as in the sliding window approach to object detection. We can also reduce the number of rectangles to consider by utilizing the branch-and-bound approach [7,6]. However, by exploiting the sparsity of BoF vectors and the inverted file index storage representation, we can achieve the goal even more efficiently.

The approach is to fit the similarity equations (Eq. 2 and 4) into an integral image computation framework. Integral images have been widely used in the object detection literature, *e.g.* Vedaldi *et al.* [16] used the integral image idea to improve the efficiency of object category detection significantly. Specifically, by converting from sum-over-word index to sum-over-feature form, and also factor the BoF normalization term out of the summation. We analyze L_1 and L_2 cases separately here. Let $\tilde{q}$ and $\tilde{d}$ denote the (*unnormalized*) TF-IDF weighted BoF histograms. Consequently, $q = \tilde{q}/\|\tilde{q}\|$ and $d = \tilde{d}/\|\tilde{d}\|$.

***L*₂ Case.** Eq. 2 can be rewritten as follows:

$$S(q,d) = \sum_{i|\tilde{q}_i \neq 0 \wedge \tilde{d}_i \neq 0} \frac{\tilde{q}_i \tilde{d}_i}{\|\tilde{q}\|\|\tilde{d}\|} = \frac{1}{\|\tilde{q}\|\|\tilde{d}\|} \sum_{i|\tilde{q}_i \neq 0 \wedge \tilde{d}_i \neq 0} \tilde{q}_i \tilde{d}_i. \tag{7}$$

Similarly, the localized similarity $S(q, d(R))$ can be written as follows:

$$S(q,d(R)) = \frac{1}{\|\tilde{q}\|\|\tilde{d}(R)\|} \sum_{i|\tilde{q}_i \neq 0 \wedge \tilde{d}_i(R) \neq 0} \tilde{q}_i \tilde{d}_i(R). \tag{8}$$

Since $\|\tilde{q}\|$ is constant,

$$S(q,d(R)) \propto \frac{1}{\|\tilde{d}(R)\|} \sum_{f|\tilde{q}_{\mathcal{C}(f)} \neq 0 \wedge f \in R} \tilde{q}_{\mathcal{C}(f)} \alpha_{\mathcal{C}(f)}, \tag{9}$$

where f denotes a feature in image I_d, and $f \in R$ means the feature f is located inside the region R. α_i is the IDF weight for word i.

From Eq. 9, we can see that the similarity is represented as the sum over votes from individual feature points in database images. For an arbitrary subrectangle, it is now straightforward to use the inverted file to accumulate the summation term in Eq. 9 for non-zero query words, and use an integral image to rapidly evaluate the term for an arbitrary subrectangle. The integral image $\mathcal{G}_{q,d}(x,y)$ of the summation term for query q and image d can be written as:

$$\mathcal{G}_{q,d}(x,y) = \sum_{f|\tilde{q}_{\mathcal{C}(f)} \neq 0 \wedge f \in (0,y,0,x)} \tilde{q}_{\mathcal{C}(f)} \alpha_{\mathcal{C}(f)}. \tag{10}$$

Under the binary TF histogram assumption, $\mathcal{G}_{q,d}(x,y)$ simplifies to the form:

$$\mathcal{G}_{q,d}(x,y) = \sum_{f|\tilde{q}_{\mathcal{C}(f)} \neq 0 \wedge f \in (0,y,0,x)} \frac{\alpha^2_{\mathcal{C}(f)}}{\tau_d(\mathcal{C}(f))}, \tag{11}$$

where $\tau_d(\mathcal{C}(f))$ is the TF of word $\mathcal{C}(f)$ in I_d, which distributes the contribution of multiple features f corresponding to the same visual word uniformly so that to ensure the binary assumption of the global TF histogram of I_d.

But in order to evaluate the full similarity in Eq. 9 we need an approximation for $\|\tilde{d}(R)\|$ since the L_2 norm does not accumulate linearly. For very large vocabularies, the L_2 norm of a BoF vector can be approximated as the square root of the L_1 norm [2]. (This follows from the observation that for large vocabularies, almost all TF histogram entries are either 1 or 0.) Hence, we replace the L_2 norm, $\|\tilde{d}(R)\|_2$, with the L_1 norm, $\|\tilde{d}(R)\|_1$, which can be computed efficiently for any subrectangle using an integral image. And, the integral image $\mathcal{H}_d(x,y)$ of $|\tilde{d}(R)|_1$ can be written as:

$$\mathcal{H}_d(x,y) = \sum_{f|f\in(0,y,0,x)} \frac{\alpha_{\mathcal{C}(f)}}{\tau_d(\mathcal{C}(f))}. \tag{12}$$

***L_1* Case.** Eq. 4 can be rewritten as follows:

$$S(q,d) = \sum_{i|\tilde{q}_i\neq 0\wedge\tilde{d}_i\neq 0} \left(\frac{\tilde{q}_i}{|\tilde{q}|} + \frac{\tilde{d}_i}{|\tilde{d}|} - \left|\frac{\tilde{q}_i}{|\tilde{q}|} - \frac{\tilde{d}_i}{|\tilde{d}|}\right|\right). \tag{13}$$

Similarly, the localized similarity $S(q,d(R))$ can be written as follows:

$$S(q,d(R)) = \sum_{i|\tilde{q}_i\neq 0\wedge\tilde{d}_i(R)\neq 0} \alpha_i \left(\frac{\tau_q(i)}{|\tilde{q}|} + \frac{\tau_{d(R)}(i)}{|\tilde{d}(R)|} - \left|\frac{\tau_q(i)}{|\tilde{q}|} - \frac{\tau_{d(R)}(i)}{|\tilde{d}(R)|}\right|\right), \tag{14}$$

where $\tau_q(i)$ and $\tau_{d(R)}(i)$ are the TFs of word i for q and $d(R)$, respectively, and α_i is the IDF weight.

We can again exploit the fact that for large vocabularies, most TF histogram entries are 0 or 1, and therefore we can approximate the BoF with its binary counterpart, where all non-zero entries are replaced by the IDF weights similar to the binary assumptions used in [4,1]. Under this assumption, $\tau_q(i) = 1$ and $\tau_{d(R)}(i) = 1$ for all i such that $\tilde{q}_i \neq 0 \wedge \tilde{d}_i \neq 0$. Breaking Eq. 14 into two cases, $|\tilde{q}| >= |\tilde{d}(R)|$ and $|\tilde{q}| < |\tilde{d}(R)|$, will remove the absolute sign and the results of the two cases can be combined by using the max operator:

$$S(q,d(R)) = \frac{2}{\max(|\tilde{q}|,|\tilde{d}(R)|)} \sum_{i|\tilde{q}_i\neq 0\wedge\tilde{d}_i(R)\neq 0} \alpha_i, \tag{15}$$

or, dropping the constant,

$$S(q,d(R)) \propto \frac{1}{\max(|\tilde{q}|,|\tilde{d}(R)|)} \sum_{f|\tilde{q}_{\mathcal{C}(f)}\neq 0\wedge f\in R} \alpha_{\mathcal{C}(f)}. \tag{16}$$

The above simplification results in factoring out of norms and a summation over features f, which is exactly what needed for integral image-based framework. Specifically, the norm $|\tilde{q}|$ is fixed with respect to R, while $|\tilde{d}(R)|$ and the summation term can be computed efficiently for all R using the integral images. Similar to the L_2 case, the integral image $\mathcal{G}_{q,d}(x,y)$ of the summation term for query q and image d can be written as:

$$\mathcal{G}_{q,d}(x,y) = \sum_{f|\tilde{q}_{\mathcal{C}(f)}\neq 0\wedge f\in(0,y,0,x)} \frac{\alpha_{\mathcal{C}(f)}}{\tau_d(\mathcal{C}(f))}, \tag{17}$$

where τ_d denotes the TF histogram as in the L_2 case. The L_1 norm $|\tilde{d}(R)|$ is computed efficiently using the integral image $\mathcal{H}_d(x, y)$ in Eq. 12.

Although the binary TF histogram assumption does not take advantage of the full histogram information during the retrieval process, we can rerank the retrieved images according to their exact histograms based on Eq. 14, rather than the binarized approximation.

Another important detail in forming the integral images is to spatially distribute multiple instances of a particular word while not violating the binarization assumption. We have found that if we uniformly distribute the whole vote α_i to different instances, *i.e.*. if there are K instances of word i, each instance gets a vote of α_i/K, we do not introduce a spatial bias by arbitrarily selecting a particular word instance, while respecting the binarization assumption. This is accomplished by the presence of $\tau_d(\mathcal{C}(f))$ in Eq. 11, 12 and 17.

In contrast to generic object category detection where slight shifts and scalings of the window greatly affect the classification scores due to feature misalignment, in our problem, a coarse grid can be used without affecting accuracy. We have found that a 80×80 or 160×160 grid is sufficient for queries larger than 200×200 pixels. If the grid is 80×80, and the image size is 480×640, memory or storage requirement for 1 million images is only 96MB which is negligible compared to the size of the vocabulary and inverted file. In this case, the integral images $\mathcal{G}_{q,d}$ and $\mathcal{H}_d$ are defined on the grid, instead of at all pixels.

3.3 Optimization

Given integral images of norms and similarities between query and database images, we need an efficient optimization scheme for Eq. 5. Here, we simply use a greedy search (see Algorithm 1). In each iteration, we sequentially optimize individual coordinate in the order of (t, b, l, r), and stop the iteration process when the returned bounding rectangle in the current iteration is the same as in the previous iteration or the maximum iteration limit is reached. From experiments, we found that our approach finds global optima in about 66% of the cases and the process generally converges in less than 3 iterations as shown in Fig. 4.

3.4 Local BoF Algorithm

We follow the same general image retrieval framework as described, for example, in [15,13,11]. For training and indexing, we (1) extract local interest regions and descriptors for all database images, (2) construct the visual vocabulary by clustering, (3) quantize all descriptors into visual words, and (4) construct an inverted file, indexed on the visual words, and including feature geometry with the index. During the testing stage, we (1) extract interest regions and descriptors in query image, (2) compute distances (or similarities) between query and all database images using the inverted file, (3) apply our Local BoF-based search to localize and rerank the top K results.

The Local BoF retrieval algorithm is briefly described in Algorithm 2. We assume a feature quantizer is given and all features are indexed based on the

Algorithm 1. Greedy Query Localization: (n_x, n_y) is the grid width and height. M is the maximum iterations. $S(u, v, w, z) =: S(q, d(R))$, $R = (u, v, w, z)$.

```
$(t, b, l, r) \leftarrow (0, n_y - 1, 0, n_x - 1)$
for $i = 1$ to $M$ do
  $t' \leftarrow \arg\max_{j=0,\ldots,b-1} S(j, b, l, r)$ and $b' \leftarrow \arg\max_{j=t'+1,\ldots,n_y-1} S(t', j, l, r)$
  $l' \leftarrow \arg\max_{j=0,\ldots,r-1} S(t', b', j, r)$ and $r' \leftarrow \arg\max_{j=l'+1,\ldots,n_x-1} S(t', b', l', j)$
  if $(t, b, l, r) = (t', b', l', r')$ then
    break
  end if
  $(t, b, l, r) \leftarrow (t', b', l', r')$
end for
return $R^* \leftarrow (t, b, l, r)$ and $\mathcal{S} \leftarrow S(t, b, l, r)$
```

Algorithm 2. Local BoF Retrieval

```
/*——————Offline——————*/
Quantize local descriptors and construct the inverted file.
for each database image $\{I_i\}_{i=1,2\ldots N}$ do
  Compute $\mathcal{H}_{d_i}$ using Eq. 12.
end for
/*——————Online——————*/
Given the query BoF $q$, use the Global BoF method to rank the images.
for all top-$K$ images $\{I_{T_j}\}_{j=1,\ldots,K}$ on the ranked list do
  Compute $\mathcal{G}_{q,d_{T_j}}$ based on Eq. 11 or Eq. 17.
  Compute $R^*(I_{T_j})$ and $\mathcal{S}(I_q, I_{T_j})$ using Algorithm 1.
end for
for all top-$K$ images $I_{T_1}, \ldots, I_{T_k}$ on the ranked list do
  Given $R^*(I_{T_j})$, recompute $\mathcal{S}(I_q, I_{T_j})$ using the non-binarized BoF.
end for
Rerank the top $K$ images based on $\mathcal{S}(I_q, I_{T_j})$.
return $R^*(I_{T_j})$ and the reranked image list.
```

quantizer, and the indices are organized into an inverted file. Offline, we compute integral norm images over the coarse uniform grid for both binary and full BoFs. Online, we first compute the BoF for the query region, sort the images based on the standard BoF algorithm, and then perform the local BoF optimization to estimate the optimal rectangle, compute similarities, and rank images.

3.5 Local Spatial Pyramid Model (LSPM)

Inspired by Lazebnik *et al.* [8], we can extend the Local BoF model by imposing a weak spatial consistency constraint using a local spatial pyramid model. Specifically, we decompose the query region into different spatial quantization levels $P \times P$ ($P = 1, 2...$). In each pyramid level, we compute the similarity vote for each grid cell in this spatial quantization and average them, and average the similarities at all pyramid levels to obtain the pyramid-based Local BoF similarity. For data sets such as the Oxford Building data set, where objects are mostly upright in the images, more levels of the spatial pyramid are more discriminative and hence result in better average precision. In our experiments we found that $P = 2$ is a good tradeoff between accuracy and complexity.

4 Results

We test the approach on two image retrieval data sets: the University of Kentucky data set(Ukbench)[1] [11], and the Oxford Building (5K) data set (Oxbuild)[2] [13]. Ukbench contains 10200 images of 2550 objects where each object has exactly four images. The evaluation metric is the average number of correct top-4 images for all 10200 queries. Oxbuild contains 5062 images of Flickr images and 55 standard test queries of 11 landmarks. The performance is evaluated as the mean average precision (mAP) score. We implemented our own retrieval system consisting of affine invariant region detection [10], SIFT [9] description, and hierarchical quantization methods, but for fair comparison to other approaches, our results here are based on the same features as [13] and [11] which are publicly available at the data set URLs. We used a fixed grid size (grid spacing meaning the size of one grid cell) of 80 × 80 pixels and performed reranking for top K images (where K=400 for Oxbuild and K=20 for Ukbench) in all experiments except the ones in Sec. 4.4, where the effects of these parameters are tested.

4.1 Results on Oxbuild

Since our method is aimed at improving retrieval on smaller queries, we have evaluated its performance as a function of query region size. For Oxbuild, we perform two types of 'query resize' experiments: (1) performance w.r.t. the resize ratio to the original query rectangle, *i.e.* test mAP values by varying the standard 55 query rectangles by fixing their center points and scales them uniformly by a set of constant factors ranging from 0 to 1; (2) performance w.r.t. the area (pixel size) of query subrectangle, *i.e.* test mAP values by choosing fixed-size query subretangles (the same number of pixels for all queries) with the same center and aspect ratio to the original query rectangles. Note that we resize query rectangles instead of the underlying query images. Fig. 2 (top) shows the comparison results of those two experiments for the L_1 case. As can be seen from the figure, both versions of our approach consistently outperform the Global BoF approach, improving mAP on average by 12.7% across all query resize ratios and 13.6% across all absolute query subrectangle sizes (pixels). In general, LSPM (level 2) showed better performance than Local BoF for larger scales due to its better discriminative power. For smaller queries, the advantage of using LSPM is not obvious due to the sparseness of the query features. More interestingly, the smaller the absolute query size, the more benefit is observed using our localized algorithms as shown in Fig. 2 (top-right).

[1] http://www.vis.uky.edu/~stewe/ukbench/

[2] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/index.html

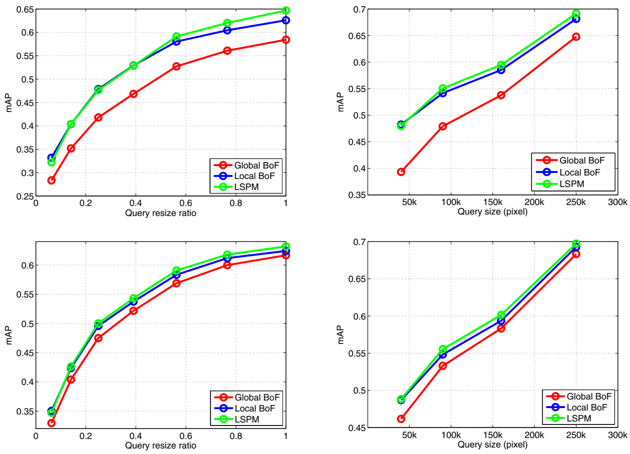


Fig. 2. Performance evaluation on Oxbuild for different approaches (Global BoF, Local BoF and LSPM) w.r.t. the query size. Top-Left: comparison w.r.t. the query size in ratios to the original query regions (the L_1 case). Top-Right: comparison w.r.t. the query size in pixels (the L_1 case). Bottom-Left: comparison w.r.t. the query size in ratios to the original query regions (the L_2 case). Bottom-Right: comparison w.r.t. the query size in pixels (the L_2 case).

Fig. 2 (bottom) shows the same experiments for the L_2 case. We can observe similar consistent improvement of our approach over the baseline but the improvement is less than the L_1 case, *i.e.* on average by 4.2% across all query resize ratios and 3.8% across all absolute query sizes (pixels). This is probably because the assumption that the L_2 norm becomes similar to the square root of the L_1 norm is less accurate due to the repeatitive structures in the data set.

In comparison to previous approaches, mAP of our approach of using the original 55 queries is 0.647 which is significantly better than the L_1 Global BoF (0.582) and L_2 BoF (0.618). And, our local BoF obtained almost identical result to the Global BoF with RANSAC-based reranking [13]. Note that our approach is significantly faster than RANSAC-based verification, and can be applied to thousands of the database images in less than 50ms during retrieval time (see Fig. 5 (right)). Another advantage of our approach is that it is not limited to rigid, mostly planar objects as in the RANSAC-based approach.

4.2 Results on UKbench

We performed the same query-size experiments for Ukbench. We resized each of the original query regions (entire image regions) by fixing its center to the center of the original image since there are no query regions are given. Since the vocabulary tree structure is not provided, we use our own HKM algorithm and the SIFT features (provided on the data set web page) to build a hierarchical vocabulary of 6 levels with the branching factor 10, and obtained the top-4 score of 3.29 which is the same as the best result of [11].

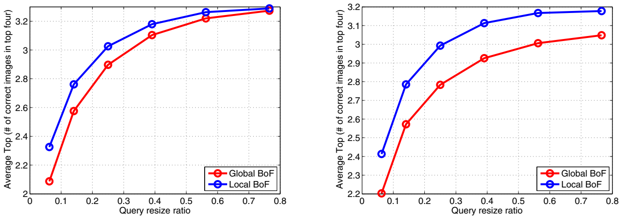


Fig. 3. Performance evaluation on Ukbench: average top-4 w.r.t. the query size. Left: the L_1 case. Right: the L_2 case.

Fig. 3 shows the results of the standard Global BoF and our Local BoF approaches. The improvement of our approach over the Global BoF is most significant for smaller queries, *i.e.* in general the smaller the query the more improvement we achieved. Specifically, the absolute improvement of the top-4 rate is 0.19 for the L_1 case and 0.21 for the L_2 case which are significant considering the strict true positive criterion used for the data set. For larger queries, our approach achieved relatively smaller improvement in retrieval because most of the images in this data set are close-up shots of objects. Comparing the L_1 and L_2 cases, the improvement for L_2 is more consistent over all query sizes.

4.3 Analysis of the Optimization Approach

We evaluated the number of greedy iterations needed for convergence during the local BoF search process using Oxbuild. As shown in Fig. 4 (left), retrieval performance improves with increasing iterations, but the improvement slows significantly after 2 iterations. Surprisingly all of the optimization for 22000 test images over 55 queries coverage in less than 4 iterations. For about 95% images, the process converges in only 2 iterations. Fig. 4 (right) validates proximity of our solutions to the global optimum when changing the area overlap ratio γ[3] to the globally optimum rectangle. Note that in about 66% of cases out of 22000 total localization tasks, our approach achieved the exact global optimum.

As shown in Table 1, we also compared the retrieval performance of our greedy-based approach and the globally optimum-based approach (branch-and-bound or brute-force search) with respect to the different query resize ratio β. Evidently, the greedy approach results in no significant degradation in mAP.

4.4 Analysis of the Algorithm Parameters

We first analyze the effect of changing the grid size (grid spacing) from 20 × 20 to 320 × 320, for a range of query sizes, using Oxbuild. From Fig. 5 (left) we can observe that the retrieval performance of the LSPM tends to be increasing and

[3] The area overlap ratio $\gamma(R_1, R_2)$ between two rectangles R_1 and R_2 is defined as: $\gamma = \frac{A(R_1 \bigcap R_2)}{A(R_1 \bigcup R_2)}$, where $A(Q)$ is the area of region Q.

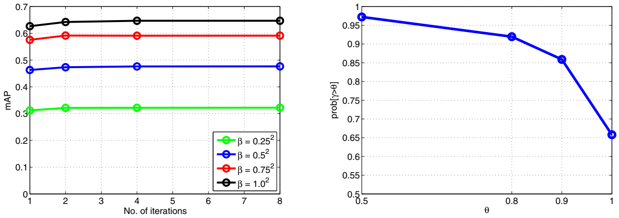


Fig. 4. Analysis of optimization on Oxbuild. Left: mAP w.r.t. the number of greedy iterations, β denotes the query resize ratio. Right: comparison of the greedy solution with the global optima: $Prob(\gamma > \theta)$ is computed as the percentage of localization tasks where the greedy solution rectangle overlaps with the global optimum rectangle when changing the area overlap ratio threshold θ.

Table 1. Performance (mAP) comparison of our greedy solution and global optimum localization-based approaches on Oxbuild

mAP	$\beta = 0.25^2$	$\beta = 0.5^2$	$\beta = 0.75^2$	$\beta = 1.0^2$
Greedy solution	0.322	0.476	0.591	0.647
Global optimum	0.329	0.481	0.591	0.644

leveling off with smaller grid sizes. While for larger grid sizes, the degradation in accuracy is more significant for small-size queries. This is reasonable because more precise localization can be achieved using smaller grids, and when the query size is smaller or similar to the grid size, LSPM becomes too sparse.

We also evaluated the performance by varying K, the number of top images to rerank, for a range of query sizes, using the same data set. Fig. 5 (middle) shows the mAP of the Local BoF and the LSPM w.r.t. K and query size. It is interesting to find that all curves level off with increasing reranking images. We can also observe a consistent mAP improvement of our approaches when moving from $K = 25$ to $K = 800$, *i.e.* the LSPM improves by 9.7% and the Local BoF improves by 8.4% on average. This figure indicates that the method is robust to K. Also, for most query sizes, the LSPM outperformed the Local BoF by about 2% on average but the LSPM resulted in a lower mAP than the local BoF for the smallest query due to the relatively coarser grid size (80×80).

4.5 Complexity and Scalability

Our approach only needs to store feature locations (x, y) (1 byte per feature coordinate) as the geometric information in the inverted file and does not require storing affine geometry parameters. Indexing 1 million images, averaging 500 features per image, requires about 1GB. In addition to the inverted file, our method requires storing L_1 or L_2 norm integrals of all database image BoFs. If each image has 48 grids, storing each integral requires 100 bytes. For 1 million images, this only amounts to 100MB, which is insignificant compared to the size

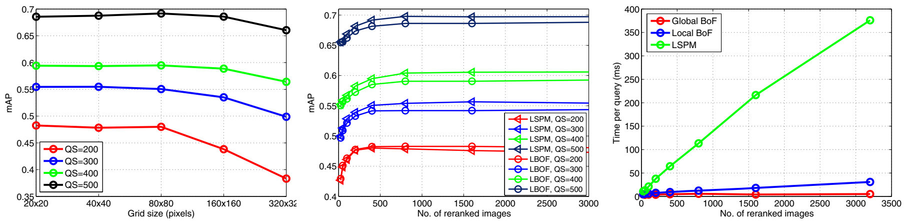


Fig. 5. Analysis of parameters using Oxbuild. Left: mAP w.r.t. the grid size (grid spacing in pixels) using the LSPM, 'QS=n' means the query region size is $n \times n$ pixels. Middle: mAP w.r.t. the number of reranked images, 'LBOF' stands for Local BoF and 'QS=n' means the query size is $n \times n$ pixels. Right: the query time comparison w.r.t. the number of reranked images.

of the inverted file. At retrieval time, our Local BoF is only slightly slower than the Global BoF approach when the optimization was run for the top 400 images, see Fig. 5 (right), and is only twice as slow when run on the top 1000 images. Typically the average query (or retrieval) time (ignoring the query feature extraction) for reranking the top 3200 images is less than 30ms compared to around 5ms for the global BoF approach. From the Local BoF curve, we can predict that our approach can spatially verify, rerank, and localize objects for 100k images in less than 1 second which is significantly faster than RANSAC-based spatial verification for the same number of reranking images. The speed is mainly due to the combination of a coarse grid, the integral image-based computation enabled by binary BoF approximation and greedy optimization.[4]

5 Conclusions

We have presented a local BoF model and its optimization method for efficient object retrieval. Our new contributions include (1) the generalization of the Global BoF to spatially localized models, Local BoF and LSPM, for reranking images and localizing objects in a unified framework, (2) the integration of the localized models with an inverted file index for efficient object retrieval in large image sets. Efficiency was achieved by introducing the binary BoF approximation. We have demonstrated consistent improvement over the baseline BoF on the average retrieval precision using our method. We have also shown that the method is much faster than alternative methods such as RANSAC.

Our local BoF framework is a general module that can be combined easily with numerous already published techniques including (1) different local features and their sample combinations [18], (2) varying quantization methods, such as HKM [11], AKM [13], Soft AKM [14], (3) specific cues, such as query expansion [13], (4) RANSAC [13], (5) gravity vector assumptions [13, 12], and (6) compression-based schemes, such as [12, 3]. We expect that combination with

[4] All the reported times are measured on a 3GHz machine with 16GB RAM.

such techniques will further improve object retrieval accuracy for large sets of images using the Local BoF framework.

References

1. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: Finding a (thick) needle in a haystack. In: CVPR, pp. 17–24 (2009)
2. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
3. Jegou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: CVPR, pp. 1169–1176 (2009)
4. Jegou, H., Douze, M., Schmid, C.: Packing bag-of-features. In: ICCV, pp. 1–8 (2009)
5. Ke, Y., Sukthankar, R., Huston, L.: Efficient near-duplicate detection and subimage retrieval. In: ACM Multimedia, pp. 869–876 (2004)
6. Lampert, C.H.: Detecting objects in large image collections and videos by efficient subimage retrieval. In: ICCV, pp. 1–8 (2009)
7. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR, pp. 1–8 (2008)
8. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
10. Matas, J., Chum, O., Urba, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: BMVC, pp. 384–396 (2002)
11. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
12. Perdoch, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: CVPR, pp. 9–16 (2009)
13. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR, pp. 1–8 (2007)
14. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR, pp. 1–8 (2008)
15. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV, pp. 1470–1477 (2003)
16. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV, pp. 1–8 (2009)
17. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling features for large scale partial-duplicate web image search. In: CVPR, pp. 25–32 (2009)
18. Wu, Z., Ke, Q., Sun, J., Shum, H.Y.: A multi-sample, multi-tree approach to bag-of-words image representation. In: ICCV, pp. 1–8 (2009)
19. Yeh, T., Lee, J.J., Darrell, T.: Fast concurrent object localization and recognition. In: CVPR, pp. 280–287 (2009)

Velocity-Dependent Shutter Sequences for Motion Deblurring

Scott McCloskey

Honeywell ACS Labs, Golden Valley, MN, USA
scott.mccloskey@honeywell.com

Abstract. We address the problem of high-quality image capture of fast-moving objects in moderate light environments. In such cases, the use of a traditional shutter is known to yield non-invertible motion blur due to the loss of certain spatial frequencies. We extend the flutter shutter method of Raskar et al. to fast-moving objects by first demonstrating that no coded exposure sequence yields an invertible point spread function for all velocities. Based on this, we argue that the shutter sequence must be dependent on object velocity, and propose a method for computing such velocity-dependent sequences. We demonstrate improved image quality from velocity-dependent sequences on fast-moving objects, as compared to sequences found using the existing sampling method.

1 Introduction

In challenging photographic situations, where ambient illumination is low or subject/camera motion is high, blur is a significant image quality problem for both consumer photographic and computer vision applications. Both optical and motion blur have been studied in the literature, and the limits of de-blurring are well understood. With respect to motion blur, it is well-known that the use of a traditional open/closed shutter results in motion blur that is non-invertible. That is, the use of a traditional shutter precludes the recovery of a sharp image information at certain spatial frequencies, and images processed by de-convolution will contain significant reconstruction artifacts. In order to address this shortcoming, recent work in computational photography has advocated the use of non-traditional capture methods to ensure invertibility of blur in the captured images. The use of coded exposure has demonstrated an ability to produce images with good contrast at all spatial frequencies without significant artifacts.

The fundamental idea of the flutter shutter approach is to open and close the shutter several times during image capture in order to produce an image with invertible motion blur. The motion blur is considered to be invertible if the associated modulation transfer function (MTF - the Fourier magnitude of the point spread function [PSF]) is greater than zero for all spatial frequencies. For such blurs, the deconvolution process is well-posed and the sharp image can be recovered from the blurred camera image, as illustrated in Fig. 1.

Though the shutter's fluttering pattern is one determinant, the effective PSF also depends on the motion of the object As with other current work with motion blur, we assume that the object follows a linear trajectory with constant

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 309–322, 2010.

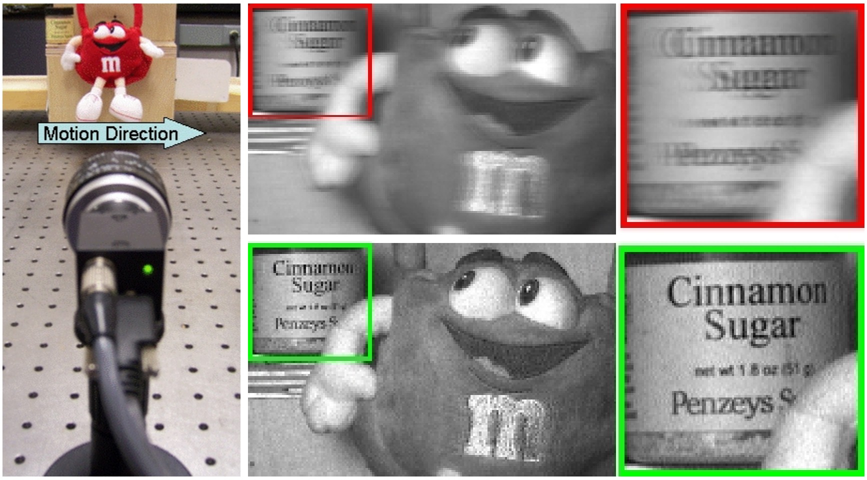


Fig. 1. We employ an off-the-shelf camera to capture coded exposure imagery of high-speed motion using velocity-dependent shutter sequences. The coded exposure image (top right) is then de-blurred to give a sharp image (bottom right) without artifacts.

(unknown) velocity, and that either blur is uniform or that blurred regions have been segmented in advance. Under these assumptions, a given flutter pattern can generate any of a family of PSFs, depending on the object's velocity and motion direction. Though this dependency has been mentioned in [1], we contribute the first analytic characterization of the relationship, and demonstrate that no fluttering pattern can generate a family consisting entirely of invertible PSFs. We therefore argue that fluttering patterns must be designed and selected for a particular velocity. We provide theoretical motivation for our algorithm to generate velocity-dependent fluttering patterns, which is shown to improve image quality on reconstructions of fast-moving objects. We also consider (previously ignored) read-out noise due to the use of an electronic shutter.

2 Related Work

Numerous methods in the category of blind deconvolution [2] have been presented to mitigate the effects of motion or optical blur in images. Though these methods may be successful relative to certain aesthetic objectives, they are fundamentally limited by the blurred input images. The image of an object moving along a straight line with constant velocity is equivalent to a sharply-focused image convolved with a 1D rectangular point spread function. The magnitude of the Fourier transform of such a PSF (known as the Modulation Transfer Function or MTF) is small at middle and high spatial frequencies, and goes to zero at several frequencies. As a result, contrast of a motion-blurred object will be significantly muted at the middle and high spatial frequencies, and there will be no contrast at a number of *lost frequencies* (the frequencies at which the MTF vanishes). These spatial frequencies are lost when captured through a traditional shutter, and post-processing the image cannot recover them and may instead introduce artifacts. Such images can, however, be processed by learning methods that use outside information (priors, etc.) to produce visually pleasing images [3,4]. While successful in that regard, hallucinating image content is inappropriate for many applications, e.g. forensics, that require the recovery of the true scene.

Given the incremental improvements of camera sensitivity, researchers have begun to use computational photographic methods to achieve fundamental new gains. Hasinoff and Kutulakos [5,6] propose light-efficient photography as a faster way of capturing images with large depth of field from multiple images. Telleen *et al.* [7] combine multiple, poorly-exposed images from a hand-held camera to produce low-noise images of stable image regions. Ben-Ezra and Nayar [8] use a hybrid camera to simultaneously acquire high-resolution/low frame rate and low-resolution/high frame rate videos; the point spread function estimated from the low resolution video is used to deblur the high resolution video. Synthetic apertures [9] have been shown capable of acquiring both scene radiance and depth in a single image; scene depth can subsequently be used to deblur optically-defocused regions of the scene, increasing depth of field. Levin *et al.* [10] acquire and process uniformly motion-blurred images with an invertible PSF by moving the camera during exposure, assuming a priori knowledge of the motion direction.

We extend the fluttering shutter method of Raskar, *et al.* [11] and do not require a priori knowledge of the motion direction. The flutter shutter approach chooses a shutter timing pattern with the intent of optimally preserving image content at all spatial frequencies, and preserving those frequencies at a nearly uniform level of contrast. Because the effective PSF is zero-padded, the MURA pattern [12] is not necessarily optimal. Raskar's shutter timing pattern is found by dividing the acquisition time into several *chops* of uniform duration, and by assigning a label of open or closed shutter to each of the chops subject to a constraint on the total exposure. Representing the timing pattern as a binary sequence (with 1s and 0s corresponding to open and closed shutter chops, respectively), the search for an optimal sequence is carried out by random sampling and a fitness function computed on the Fourier magnitude of the binary sequence, *which is assumed to be the effective MTF*. Agrawal and Xu [13] present a method to determine chop sequences that are optimal with respect to both invertibility and ease of blur estimation via alpha matting, but continue to conflate the PSF and the chop sequence. As we demonstrate in the next section, the equivalence of the binary chop sequence and the PSF only holds at a particular velocity and, at all other velocities, the invertibility of the effective PSF cannot be guaranteed.

3 Velocity Dependence

The fundamental notion behind the fluttering shutter concept is that the open/close sequence of the shutter should be chosen to give a PSF that passes all spatial frequencies with nearly uniform levels of contrast. In addition to the

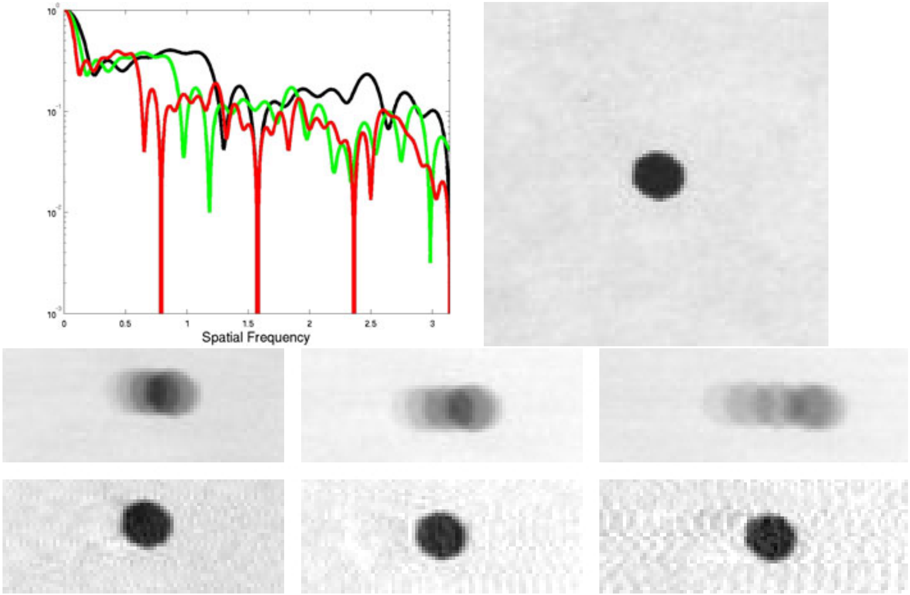


Fig. 2. For a particular fluttering sequence, the effective PSF/MTF depends on subject velocity. (Top Left) The effective MTFs of motion through a particular fluttering shutter for object velocities of 3 pixels per ms (black), $4p/ms$ (green), and $6p/ms$ (red). (Top Right) Reference image of stationary target. (Lower Images) Coded exposure images [top] of a dot moving left to right and de-blurred images [bottom]. Though the reconstructed image quality is good for object speeds of $3p/ms$ (bottom left), there are lost frequencies at $6p/ms$ (bottom right) and the reconstruction has noticeable artifacts.

binary chop sequence $S(t), t \in \{0, 1, ...N-1\}$[1], the fluttering pattern is specified by the duration t_{chop} of each chop. As such, the exposure time of the flutter shutter image is $t_{chop} \sum S(t)$ and the total acquisition time (the time from the start of the first open chop to the end of the last) is Nt_{chop}. By convention, the fluttering pattern is chosen to have an acquisition time no greater than twice the exposure time, i.e. no fewer than half of the $S(t)$ are open shutter chops. In order to implement an arbitrary chop sequence on a particular camera, it is necessary for the camera to support open and closed shutter periods as short as t_{chop}, a constraint that we discuss further in Section 4.

Though the fluttering sequence is one determinant, the effective PSF of motion blur also depends on the object's velocity on the image sensor. Though the velocity on the image sensor depends both on the object's real-world velocity and its distance from the camera, it is the image velocity that determines the PSF. Because of the PSF's dependence on velocity, a particular fluttering sequence

[1] By convention, the chop sequence begins and ends with a 1 representing open shutter chops, i.e. $S(0) = S(N-1) = 1$.

defines a family of PSFs, as illustrated in Fig. 2. A notable member of this family, which we refer to as the 'nominal' PSF, is effective when the object moves over a range of N pixels during the course of exposure with a fluttering sequence composed of N chops. In this case the effective PSF (call it B_N) is equal to a scaled version of the chop sequence S,

$$B_N(t) = \frac{S(t)}{\sum S(t)},\ \textit{for}\ t = 0, 1, ...N-1. \tag{1}$$

In this case, which has been considered in [11,13], the Fourier transform $\widehat{B_N}$ of the PSF B_N will be the same as that of the chop sequence S (up to a scale factor). Presuming that S was chosen to preserve all frequencies, this nominal PSF is invertible and the sharp image can be recovered.

In the general case, however, the PSF is a stretched version of the chop sequence and may not be invertible. In fact, no chop sequence can generate a family consisting of invertible PSFs for all velocities. In particular, if the object moves over $2N$ pixels during the course of exposure, the effective PSF will be

$$B_{2N} = \frac{1}{2\sum S(t)} * \left[S(0)\ S(0)\ S(1)\ S(1)...S(N-1)\ S(N-1)\right], \tag{2}$$

where * represents an element-wise multiplication of the sequence. As we will now demonstrate, B_{2N} suffers lost frequencies that cannot be recovered post-hoc.

Lemma 1. *Let S be an arbitrary chop sequence of length N. The effective PSF B_{2N} for an object that moves over $2N$ pixels during exposure will have a lost frequency at $k = \frac{N}{2}$.*

Proof. Let $A = \frac{1}{2\sum S(t)}$.

$$\begin{aligned}
\widehat{B_{2N}}(k) &= A\sum_{t=0}^{N-1} S(t)\left(e^{-i\frac{2\pi}{N}k(2t+1)} + e^{-i\frac{2\pi}{N}k2t}\right)\\
&= A\sum_{t=0}^{N-1} S(t)e^{-i\frac{2\pi}{N}kt}\left(e^{-i\frac{2\pi}{N}k(t+1)} + e^{-i\frac{2\pi}{N}kt}\right)\\
\widehat{B_{2N}}(\tfrac{N}{2}) &= A\sum_{t=0}^{N-1} S(t)e^{-i\pi t}\left(e^{-i\pi(t+1)} + e^{-i\pi t}\right)\\
&= A\sum_{t=0}^{N-1} S(t)e^{-i\pi t}0 = 0\square
\end{aligned} \tag{3}$$

It can similarly be shown that PSFs of the general form

$$B_{\kappa N} = \frac{1}{\kappa\sum S(t)} * \left[\underbrace{S(0)...S(0)}_{\kappa\ times}, ...\underbrace{S(N-1)...S(N-1)}_{\kappa\ times}\right], \tag{4}$$

will have $\kappa - 1$ lost frequencies. As well, it can be shown that for any object that moves over more than $2N$ pixels, the MTF will have at least one zero in the

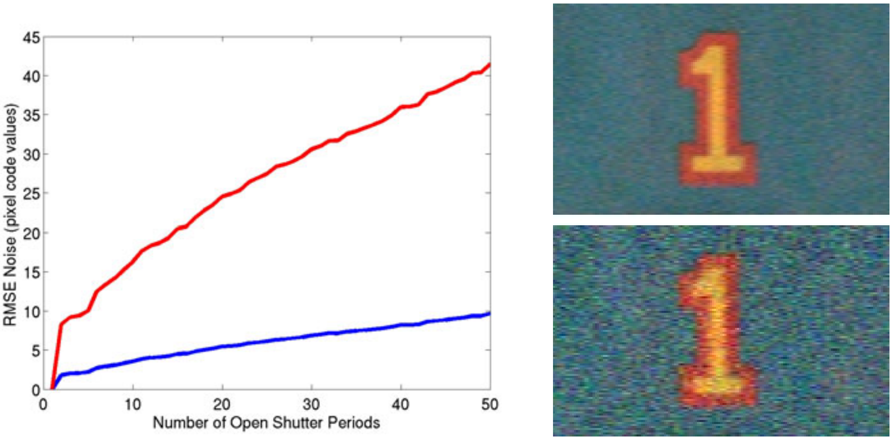


Fig. 3. Use of an electronic shutter for coded exposure imposes read-out noise proportional to the number of open shutter periods. (Left) Plots shows root mean squared error (RMSE) due to read-out noise in the captured image (blue), and the de-blurred image (red). For all captures, the exposure time is fixed at 30ms. (Top right) Reconstructed patch derived from an image captured with a physical shutter (taken from [11]). (Bottom right) Reconstructed image patch derived from an image with simulated read-out noise corresponding to 13 open chop periods, i.e. representing the electronic shutter implementation of the sequence given in [11].

range $0 \leq k \leq \frac{N}{2}$. The implication of this result is that the invertibility of coded exposure blur depends on the velocity; one cannot expect a flutter sequence designed for motion over N pixels to perform well when the velocity produces an effective motion of $2N$ pixels. We demonstrate the image quality implications of this in the Fig. 2 and the experiments of Section 6.

In order to capture images with invertible motion blur, the shutter's fluttering pattern must be selected according to the object velocity. The use of an inappropriate fluttering pattern may cause artifacts in the processed images, as illustrated in Fig. 2. As such, it is necessary to compute different fluttering patterns for specific velocities. On naive way to ensure invertibility of blur is to shorten the duration t_{chop} of each chop to compensate for higher than expected velocity. There are two problems with this approach, namely that (1) the exposure time would be reduced (and noise increased) and (2) the camera hardware may not support the shortened chop duration. Before describing our method to determine velocity-dependant fluttering patterns for each exposure time, we first discuss hardware limitations on the fluttering sequence.

4 Hardware Considerations

Though the original flutter shutter work [11] was demonstrated with a custom-made camera, more recent work [13] (including our own) has employed off-the-shelf cameras from Point Gray Research. Several of their cameras support flutter shutter image acquisition through an external trigger and multiple-exposure capture mode. This mode captures a single image whose exposure is accumulated over a pre-set number of pulses of variable duration. Because the camera lacks a physical shutter, the CCD sensor is cleared at the beginning of each pulse and at the end of a pulse the charge is added to the accumulated exposure. This transfer imposes read-out noise at the end of each open shutter period of a flutter shutter capture, a fact that has not been noted elsewhere. The noise level in the coded exposure image is proportional to the number of open shutter periods, as shown by the blue line in Fig. 3. Because de-convolving the flutter shutter PSF from the captured image amplifies noise, images de-blurred from those captured with more open shutter periods will have still more noise, as shown by the red line. The two images in Fig. 3 illustrate the difference in reconstructed image quality between a physical shutter implementation (top) and electronic shutter implementation (bottom) of the sequence given in [11]. In order to avoid such noise-related degradation, we bias our shutter finding method to favour sequences with fewer open shutter periods, as described in the next section.

A second result of the lack of a physical shutter is that there are constraints on both the minimum open shutter pulse length and the minimum time between pulses. These constraints depend on the image format and frame rate; for the Flea$^{\circledR}$2 camera used in our experiments, the 800-by-600 pixel grayscale image mode at 15Hz is the least restrictive, requiring pulse lengths of at least $1\mu s$ and at least $1.21ms$ between pulses. This second constraint is quite restrictive in light of the use of randomly-sampled binary chop sequences. In the event that the chop sequence contains the subsequence 101, this means that t_{chop} cannot fall below $1.21ms$, meaning that (for example) the minimum acquisition time of the 52 tap sequence given in [11] is 62ms and the minimum exposure time is 31ms. In the event that the image velocity of an object is 4 pixels per ms (a velocity that we consider in our experiments), the effective PSF would be 248 pixels long. Because of edge effects in the de-convolution, less than half of our 800-by-600 images could be recovered in this scenario.

5 Shutter Finding Method

Though it is possible to employ rejection sampling to find sequences without a 101 subsequence, this strategy would be extremely inefficient, as the frequency of random binary strings without a 101 subsequence decreases exponentially with the sequence length. For 32 element chop sequences, more than 98% of all sequences have a 101 substring, and for 52 element sequences the proportion is more than 99.9%. Use of rejection sampling, therefore, would add a factor of 50 to 1000 to the time required to find a suitable sequence. Instead of attempting to find a good sequence by sampling random sequences and rejecting those that can't be implemented, our method *constructs* a near optimal sequence that respects given hardware constraints.

We attempt to find an optimal PSF with respect to reconstructed image quality. In previous work [11], the minimum contrast in the MTF and variance of the MTF are mentioned as optimization criteria. We add a third criteria, mean contrast in the MTF and, when targeting a camera with an electronic shutter, a fourth term (number of open shutter periods) to limit read-out noise. Numerically, the fitness of a given PSF is a weighted sum of these terms,

$$F(B) = w_1\ min_k(|\hat{B}(k)|) + w_2\ var_k(|\hat{B}(k)|) + w_3\ mean_k(|\hat{B}(k)|) + w_4\ C, \tag{5}$$

where C represents the number of open shutter pulses.

In order to find reasonable values for these weights, we have simulated all PSFs for $N = 16$ and measured the RMSE of the reconstructed image in the presence of Gaussian noise (including a noise component proportional to C). By computing these errors for 5 random images from the Corel dataset, we find the optimal weights in the least-squares sense, and set $w_1 = 0.1$, $w_2 = -0.2$, $w_3 = 3.4$, and $w_4 = -0.1$.

Because Lemma 1 tells us that no single sequence can be expected to produce a good coded exposure image for all velocities, our method determines a unique flutter sequence for each combination of subject velocity and total required exposure time. Sequences can be completely specified by the open shutter segments, each segment having a duration and start time. The segments are constrained to be non-overlapping, have durations and spacings that respect hardware constraints, and have a total open shutter duration that equals the required exposure time. Our method builds a flutter pattern by first determining the segment durations and then determining each segment's start time. As we will show, the choice of segment durations (without start times) gives an upper bound to the contrast at all spatial frequencies; it determines the envelope of the MTF.

Lemma 2. *Let B be an arbitrary PSF of length N, let $B^1, B^2, ...B^C$ represent N-length PSFs such that $B(t) = \sum_{c=1}^{C} B^c(t)$, and let $\overleftarrow{B^1}, \overleftarrow{B^2}, ...\overleftarrow{B^C}$ represent shifted versions of the B^c such that its first non-zero entry appears at $t = 0$ (see Fig. 4). The sum of the MTFs of the $\overleftarrow{B^c}$ is an upper bound of the MTF of B.*

Proof.

$$\begin{aligned} \|\widehat{B}(k)\| = \quad & \|\sum_{t=0}^{N-1} B(t)e^{-i\frac{2\pi}{N}kt}\| \quad = \|\sum_{t=0}^{N-1}\sum_{c=1}^{C} B^c(t)e^{-i\frac{2\pi}{N}kt}\| \\ \leq \quad & \sum_{c=1}^{C} \|\sum_{t=0}^{N-1} B^c(t)e^{-i\frac{2\pi}{N}kt}\| \\ = & \sum_{c=1}^{C} \|\widehat{B^c}(k)\| = \sum_{c=1}^{C} \|\widehat{\overleftarrow{B^c}}(k)\| \equiv \lceil \|\widehat{B}(k)\| \rceil \square \end{aligned} \tag{6}$$

We denote this final quantity $\lceil \|\widehat{B}\| \rceil$ for future reference. This insight allows us to significantly limit the search space of potential flutter sequences, investigating only those composed of chop durations that might result in an invertible PSF.

Our algorithm produces a shutter sequence given the required exposure, subject velocity (given in pixels per ms), and hardware constraints on the shortest

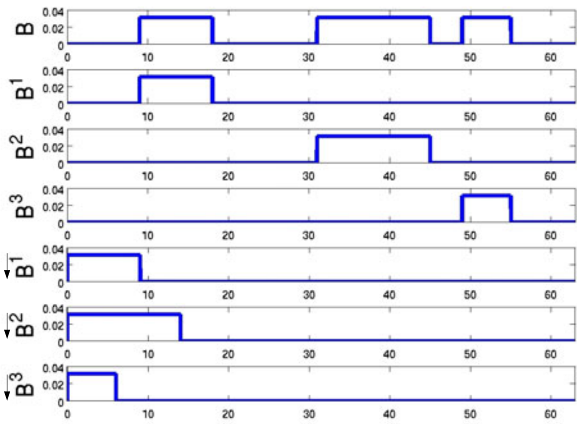


Fig. 4. PSF B (top row), its decomposition into chops (rows 2-4), and the shifted versions of these chops (rows 5-7). Lemma 2 shows that the MTF of B is bounded at each spatial frequency by the sum of the MTFs of the shifted chops.

chop duration c_{chop} and shortest period between open shutter periods c_{gap}. We find an optimal shutter sequence by first ranking each combination of open shutter durations and then by exploring arrangements of these in priority order, as in Algorithm 1. The search is terminated either when the fitness of the current best shutter sequence is greater than that of the envelope of the next combination of open shutter chop durations or when an optional timeout is reached. We describe the steps in greater detail in the remainder of this section.

Our method first determines all partitions of the required exposure time into sets of valid open shutter chop durations. We take t_{chop} to be the larger of either the minimum integration constraint or the value $\frac{1}{v}$, where v is the object's image velocity expressed in pixels per ms. The set of potential open chop lengths is taken to be all integer multiples of this shortest duration, and we compute all partitions of the exposure time into open shutter chops from this set. For each such partition, we compute the MTF envelope $\lceil \|\widehat{B}\| \rceil$ and measure the fitness of that envelope. This gives a ranking of each partition based on the *potential* fitness of a PSF corresponding to a shutter timing with that set of chop durations.

In the order of this ranking, we consider each partition and attempt to find the matching set of open shutter chop start times that produces the best PSF. We do this (in the BestSequenceOfPartition function) by starting with a random set of start times, such that the open shutter chops do not overlap and the required time between them is maintained. We then pursue a hill climbing strategy of computing several deformations of the current start times, selecting the one with the best fitness, and iterating. We repeat this process for several random sets of start times, and keep the sequence corresponding to the overall best PSF.

The ranked list of partitions is traversed until the fitness of the envelope corresponding to the next best partition is less than the fitness of the current

```
Input: Exposure time $T$, velocity $V$, constraints $c_{gap}$ and $c_{chop}$
Output: Shutter sequence $S$ and $t_{chop}$
$t_{chop}$ = max(1/V, min($c_{chop}$, $c_{gap}$));
numOpenChops = T/$t_{chop}$;
find all partitions of numOpenChops;
sort partitions by decreasing $\lceil \|\widehat{B}\| \rceil$;
$S$ = zeros(1, 2*numOpenChops);
foreach partition $P$ do
    Compute $\lceil \|\widehat{B}\| \rceil$ for $P$;
    if $\lceil \|\widehat{B}\| \rceil \leq$ Fitness(S) then
        break;
    end
    $\hat{S}$ = BestSequenceOfPartition($P$, $V$);
    if Fitness(S,V) $\leq$ Fitness($\hat{S}$,V) then
        $S = \hat{S}$;
    end
end
```

Algorithm 1. Overall shutter finding method

best PSF. Additionally, a time limit can be incorporated to produce a result within a budgeted amount of computation. At the completion of this process, the timing sequence with the highest PSF fitness is stored for use when the given object velocity and exposure time are required.

For our experiments, we have used this method to produce fluttering patterns for a wide range of combinations of subject velocity and required exposure time. These fluttering patterns are pre-computed and stored on the computer controlling the camera.

6 Experiments

In order to validate the claim that our velocity-dependant fluttering sequences provide increased image quality relative to the existing sampling method, we have carried out a number of experiments on real moving objects. Our coded exposure images are processed using the example code from [11] to produce a sharp image of the moving object. Because of the issues described in Sec. 4, we cannot use the 52 chop sequence given in [11] for comparison. Instead, we employ Raskar's sampling method to determine new fluttering patterns with $t_{chop} \geq 1.25ms$, the shortest chop length allowable by the camera. For relatively short exposure times, the search space of potential binary sequences is small, and can be searched exhaustively. For a $4ms$ exposure time, for instance, using $t_{chop} = 1.33ms$ requires three open chops and the search space has only $2^6 = 128$ elements, of which the optimal sequence is 1101. We use this to capture an image of an object moving at 4 pixels per ms, and present the de-blurred results in Fig. 5 (left column).

Because the object moves through more than one pixel per t_{chop}, the extent of the PSF is greater than the sequence's nominal PSF of [0.33 0.33 0 0.33],

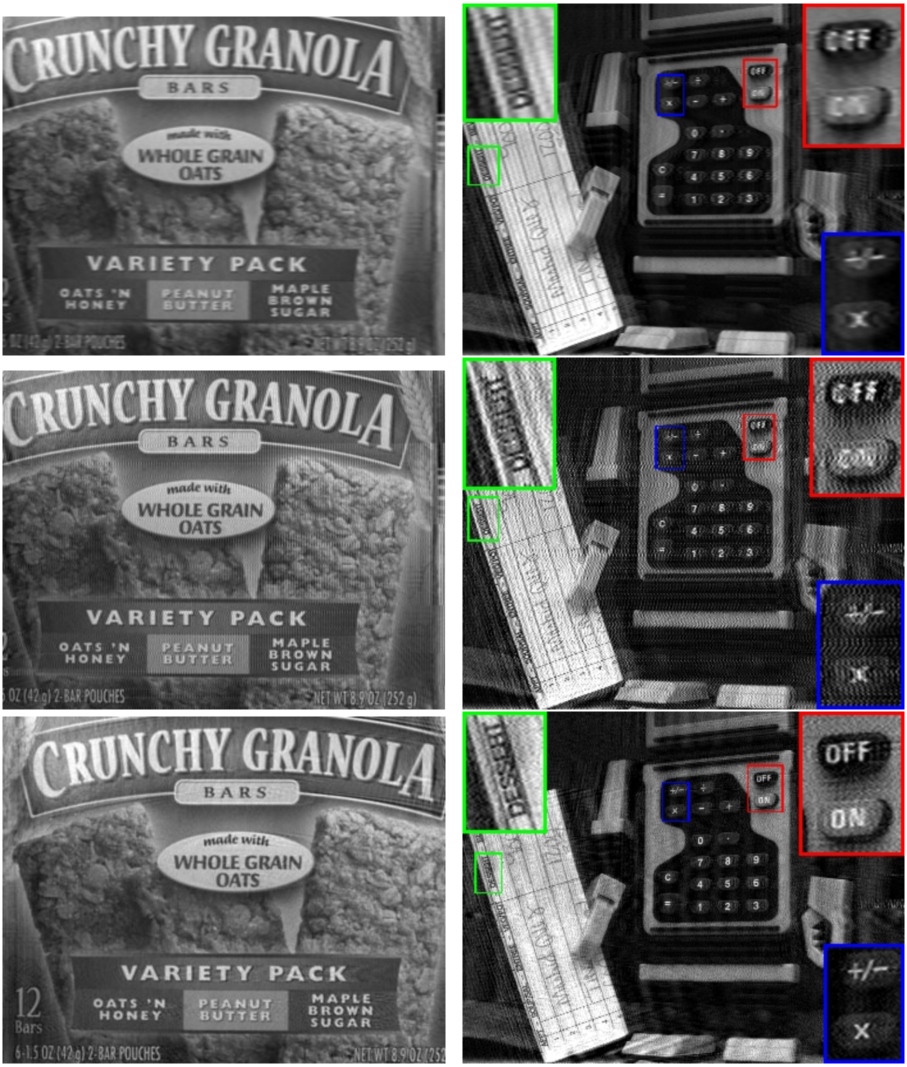


Fig. 5. Comparing de-blurred results to the existing method. (**Top row**) De-blurred flutter shutter image using the sampling and de-blurring methods of [11]. (**Centre row**) The same image, de-blurred using the effective PSF. (**Bottom row**) De-blurred flutter shutter image, acquired using a fluttering sequence determined by our method. Images in the left column are $4ms$ exposures of an object moving at 4 pixels per ms; image in the bottom row has 32 pixels of blur. Images in the right column (with annotated insets) are $8ms$ exposures of an object moving at 4 pixels per ms; image in the bottom row has 60 pixels of blur.

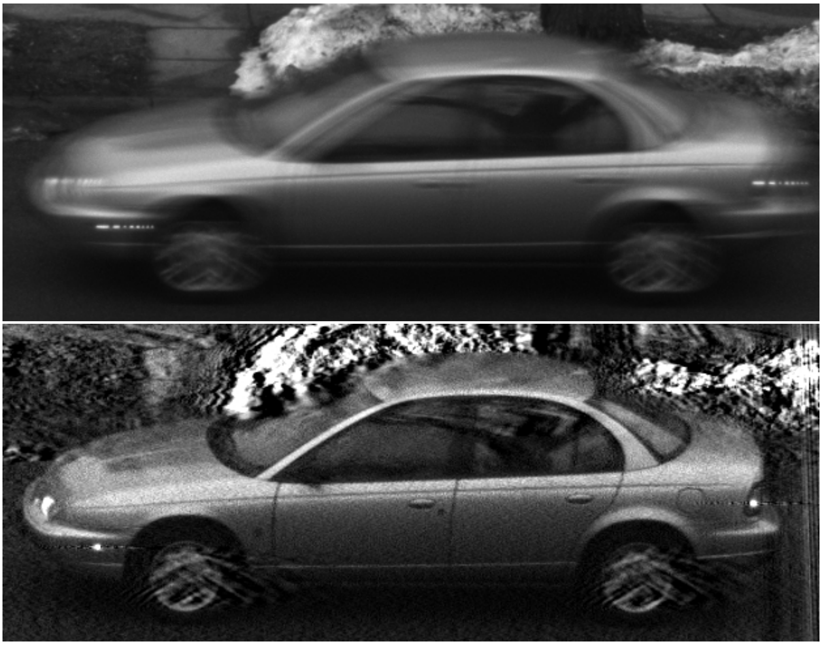


Fig. 6. (Top) Flutter shutter image captured of a car driving down a residential street, with 43 pixels of blur. (Bottom) Reconstructed image, in which details on the car are preserved; note that the static background has artifacts from the deconvolution. In this case, the street's speed limit serves as a strong prior on object velocity, obviating the need for explicit pre-capture velocity estimation. Exposure time is 20ms with velocity of 1.1 pixel per ms; shutter sequence is 1001001001100110001000111100011111111, with $t_{chop} = 1ms$.

and there are thus two choices for de-blurring. The approach taken in [11] is to re-sample the image to a smaller one in which the nominal PSF is the effective PSF. The nominal PSF is then de-convolved from this image, and the result is re-sampled in order to produce an image of the same size as the input. In this case, where the effective PSF is more than 4 times the length of the nominal PSF, the de-blurred image (top row of Fig. 5) has soft edges due to the final up-sampling by a factor of 4. In order to avoid this re-sampling step, we could instead de-convolve the effective PSF from the input image directly, as shown in Fig. 5 (middle row). As predicted by Lemma 1, however, this effective PSF is non-invertible and the resulting image has noticeable artifacts.

The de-blurred result of our shutter sequence is shown in Fig. 5 (bottom row), and avoids the artifacts due to lost frequencies while still preserving sharp edges in the scene. The shutter sequence used to capture the coded exposure image was 10000001111111100000000100000011100000111, with $t_{chop} = 0.25ms$.

Fig. 5 (right column) shows a capture with an 8ms exposure time and object velocity of 4 pixels per ms. The fluttering pattern determined by sampling is 110110000101 with $t_{chop} = 1.33ms$, and our computed shutter sequence is 1111000001111100000111110000111110000100001111 with $t_{chop} = 0.33ms$. As before, the de-blurred image resulting from our shutter sequence maintains high-frequency information without significant lost frequency artifacts, whereas the fluttering pattern derived from sampling gives either soft edges or significant reconstruction artifacts, depending on the de-blurring approach. All three images in this column show artifacts at occlusion edges, similar to the images of [11].

Though we have argued for, and demonstrated the benefits of, velocity-dependant shutter sequences for motion capture, we have not presented a pre-capture velocity estimation method for shutter selection. While such a method would be helpful, it is not necessary in all situations. Fig. 6 shows, for example, the captured and de-blurred images of a car driving down a residential street. In this case, the street's posted speed limit serves as a strong prior on a vehicle's velocity, obviating the need for explicit motion estimation before image capture. It is unlikely that any vehicle will travel at twice the posted speed limit, meaning that the lost frequencies predicted by Lemma 1 are an unlikely problem. One can imagine, however, that a shutter sequence providing optimal reconstructions of a residential street will perform poorly on a highway.

7 Conclusions and Future Work

We have presented a method for determining velocity-dependant shutter sequences to capture coded exposure imagery of fast-moving objects. We have demonstrated that these shutter sequences produce higher quality de-blurred imagery than those determined by the existing random sampling method. This algorithm is motivated by the (heretofore unnoted) observation that a particular shutter sequence gives rise to a *family* of PSFs, with the effective PSF determined by the object's velocity. We contribute an analytic proof that no shutter sequence can be devised that produces a family of invertible PSFs and that, in particular, a shutter sequence will produce non-invertible blur when the velocity is more than twice a nominal velocity. Our method for determining the optimal shutter sequence for a given combination of exposure time and object velocity is based on a priority search over the space of potential sequences, and features a termination condition that ensures optimality. We have also noted, measured, and incorporated a term in our optimisation to account for the fact that implementations of the flutter shutter based on electronic shutters incur read-out noise proportional to the number of open shutter periods.

Throughout these experiments, we have used manual estimation of the object velocity in order to select the appropriate fluttering sequence. In order to apply our method in unconstrained settings, this step should be performed automatically before initiating image capture. Though this step is non-trivial, we expect that the large body of literature on tracking and motion estimation will yield a workable approach, given two facts. First, real-world moving objects have inertia which precludes sudden changes of direction and velocity. Second, we note that all cameras already have meters that estimate a quantity (illumination) that potentially changes much quicker than velocity. It should also be noted that many cameras/lenses already have sensors that provide real-time motion estimates for optical image stabilisation, and that accurate velocity estimation obviates the need for explicit blur estimation from the image, as the shutter pattern and estimated velocity combined determine the extent of the PSF.

Acknowledgements

This material is based upon work supported by the U.S. Army Biometrics Task Force under Contract No. W91CRB-09-C-0013. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the U.S. Army. Flea®2 is a trademark of Point Grey Research, Inc. All brand names and trademarks used herein are for descriptive purposes only and are the property of their respective owners.

References

1. Agrawal, A., Raskar, R.: Optimal single image capture for motion deblurring. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2560–2567 (2009)
2. Haykin, S.: Blind Deconvolution. Prentice-Hall, Englewood Cliffs (1994)
3. Jia, J.: Single image motion deblurring using transparency. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
4. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. In: ACM SIGGRAPH (2008)
5. Hasinoff, S.W., Kutulakos, K.N.: Light-efficient photography. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 45–59. Springer, Heidelberg (2008)
6. Hasinoff, S.W., Kutulakos, K.N., Durand, F., Freeman, W.T.: Time-constrained photography. In: ICCV (2009)
7. Telleen, J., Sullivan, A., Yee, J., Wang, O., Gunawardane, P., Collins, I., Davis, J.: Synthetic shutter speed imaging. Computer Graphics Forum 26, 591–598 (2007)
8. Ben-ezra, M., Nayar, S.K.: Motion deblurring using hybrid imaging. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 657–664 (2003)
9. Levin, A., Fergus, R., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. In: ACM SIGGRAPH (2007)
10. Levin, A., Sand, P., Cho, T.S., Durand, F., Freeman, W.T.: Motion-invariant photography. In: ACM SIGGRAPH (2008)
11. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. ACM Trans. Graph. 25, 795–804 (2006)
12. Caroli, E., Stephen, J., Cocco, G., Natalucci, L., Spizzichino, A.: Coded aperture imaging in x- and gammaray astronomy. Space Science Reviews 45, 349–403 (1987)
13. Agrawal, A., Xu, Y.: Coded exposure deblurring: Optimized codes for psf estimation and invertibility. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2066–2073 (2009)

Colorization for Single Image Super Resolution

Shuaicheng Liu[1], Michael S. Brown[1], Seon Joo Kim[1], and Yu-Wing Tai[2]

[1] National University of Singapore
[2] Korea Advanced Institute of Science and Technology

Abstract. This paper introduces a new procedure to handle color in single image super resolution (SR). Most existing SR techniques focus primarily on enforcing image priors or synthesizing image details; less attention is paid to the final color assignment. As a result, many existing SR techniques exhibit some form of color aberration in the final upsampled image. In this paper, we outline a procedure based on image colorization and back-projection to perform color assignment guided by the super-resolution luminance channel. We have found that our procedure produces better results both quantitatively and qualitatively than existing approaches. In addition, our approach is generic and can be incorporated into any existing SR techniques.

Keywords: Super resolution, colorization, image upsampling.

1 Introduction and Related Work

Image super resolution (SR) refers to techniques that estimate a high-resolution (HR) image from a single low-resolution (LR) image input. Strategies to address the image SR problem are typically categorized into three broad methods: interpolation based methods, reconstruction based methods, and learning based methods.

Interpolation based techniques (e.g., [1,2,3,4]) have their roots in sampling theory and interpolate the HR image directly from the LR input. While these approaches tend to blur high frequency details resulting in noticeable aliasing artifacts along edges, they remain popular due to their computational simplicity. Reconstruction based approaches (e.g., [5,6,7,8,9,10,11,12,13]) estimate an HR image by enforcing priors in the upsampling process. Such priors are commonly incorporated into a back-projection framework to reduce artifacts around edges while constraining the estimated HR image against the LR input. Learning based techniques estimate high frequency details from a training set of HR images that encode the relationship between HR and LR images (e.g., [14,15,16,17,18,19,20,21]). These approaches synthesize missing details based on similarities between the input LR image and the examples in the training set based on patch similarities. Hybrid approaches that combine elements of reconstruction and learning based methods have also been proposed (e.g., [22,23]).

While these existing SR techniques have successfully demonstrated ways to enhance image quality through priors or detail hallucination – how to handle color in the SR process has received far less attention. Instead, two simple approaches

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 323–336, 2010.

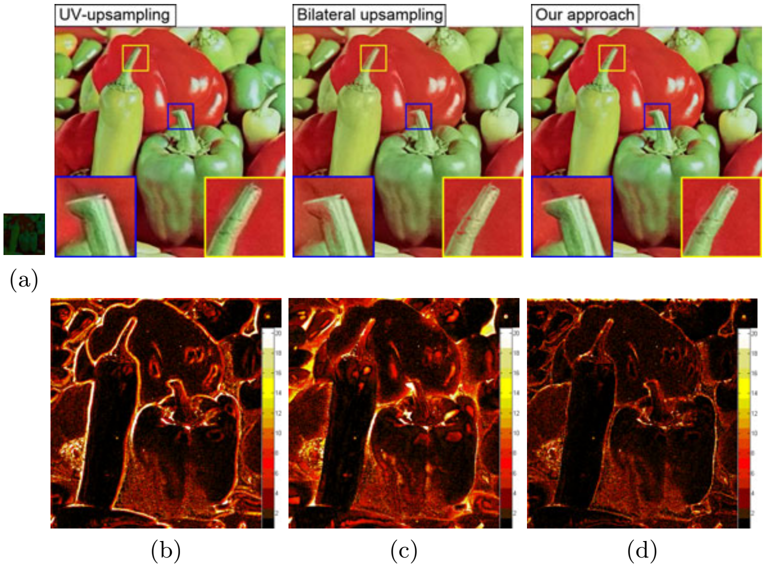


Fig. 1. (a) LR chrominance input. Results using bicubic interpolation of the UV channels (b), using joint-bilateral upsampling [25] (c), and our result (d). Color difference maps (bottom) are computed based on the CIEDE2000 color difference formula ([26,27]).

are commonly used to assign color. The first approach is to perform color assignment using simple upsampling of the chrominance values. This approach, used extensively in both reconstruction-based and learning-based SR (e.g. [12,13,19,24]), first transforms the input image from RGB to another color space, most notably YUV. Super resolution is applied only to the luminance channel, Y. The chrominance channels, U and V, are then upsampled using interpolation methods (e.g. bilinear, bicubic) and the final RGB is computed by recombining the new SR luminance image with the interpolated chrominance to RGB. The second approach, used primarily in learning-based techniques (e.g. [14,15,16]), is to use the full RGB channels in patch matching for detail synthesis, thus directly computing an RGB output.

These two existing approaches for SR color assignment have drawbacks. The basis for the UV-upsampling approach is that the human visual system is more sensitive to intensities than color and can therefore tolerate the color inaccuracies in this type of approximation. However, color artifacts along the edges, are still observable, especially under large magnification factors as shown in Fig. 1. Performing better upsampling of the chrominance, by weighted average [28] or joint-bilateral filtering [25], can reduce these artifacts as shown in Fig. 1(c), but not to the same extent as our algorithm (Fig. 1(d)). In addition, techniques such as joint-bilateral upsampling requires parameter-tuning to adjust the Gaussian window size and parameters of the bi-lateral filter's spatial and range components to obtain optimal results.

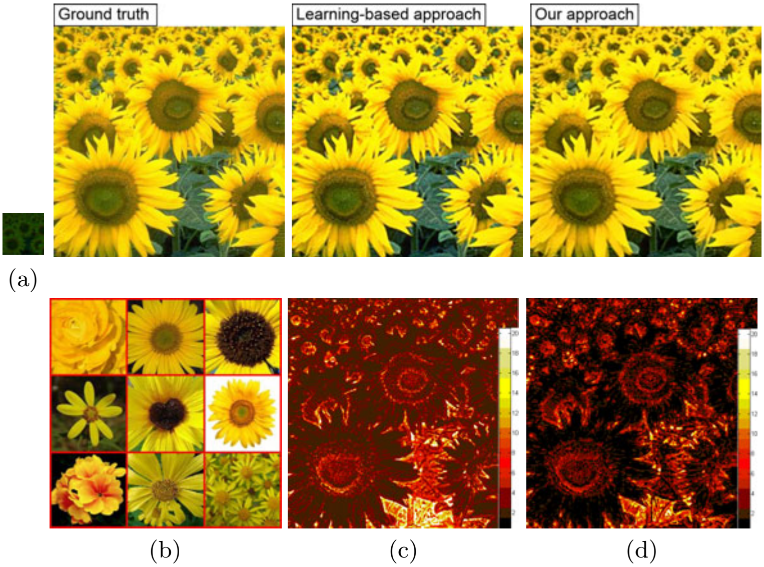


Fig. 2. (a) LR chrominance input, (b) ground truth image (top) and training images (bottom), (c) result using learning based SR [16], (d) our result. Color difference maps are computed based on the CIEDE2000 color difference formula ([26,27]).

For learning-based techniques, the quality of the final color assignment depends heavily on the similarity between the training data and the input image. The techniques that perform full RGB learning can exhibit various color artifacts when suitable patches cannot be found in the the training data. Approaches that apply learning-based on the luminance channel in tandem with UV-upsampling can still exhibit errors when the estimated SR luminance images contains contrast shifts due to training set mismatches. Since back-projection is often not used in learning-based techniques, this error in the SR luminance image can lead to color shifts in the final RGB assignment. Fig. 2 shows examples of the color problems often found in learning-based approaches.

In this paper, we propose a new approach to reconstruct colors when performing single image super resolution. As with chrominance upsampling, our approach applies super resolution only to the luminance channel. Unique to our approach, however, is the use of image colorization [29,30] to assign the chrominance values. To do this, we first compute a chrominance map that adjusts the spatial locations of the chrominance samples supplied by the LR input image. The chrominance map is then used to colorize the final result based on the SR luminance channel. When applying our approach to learning-based SR techniques, we also introduce a back-projection step to first normalize the luminance channel before image colorization. We show that this back-projection procedure has little adverse impact on the synthesized details. Our approach not only shows improvements both visually and quantitatively, but is straight-forward to implement and requires no

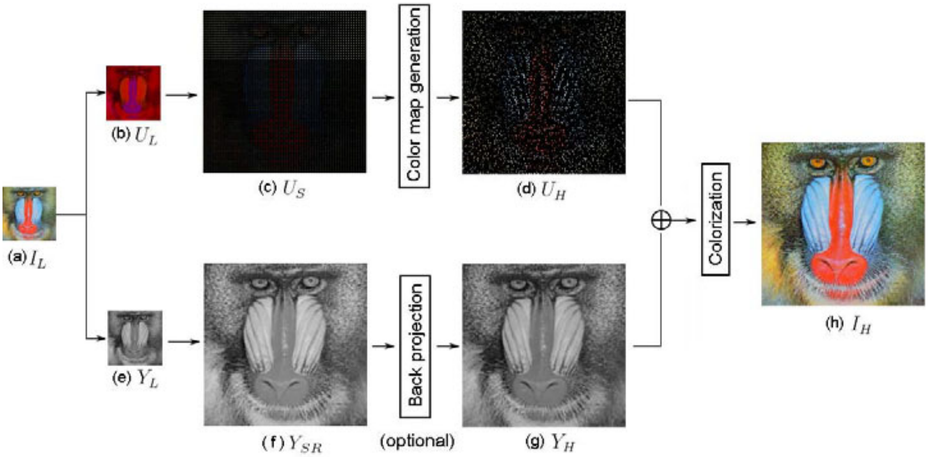


Fig. 3. The pipeline of our algorithm. (a) LR input image. (b) The chrominance component of input image. (c) Initial chrominance map produced by expanding (b) with the desired scale without any interpolation. (d) Adjusted chrominance map. (e) The luminance component of input image. (f) Upsampled image using any SR algorithm. (g) Upsampled image produced by adding the back-projection constraint (if necessary). (h) Final color SR image obtained by combining the color map (d) and the SR luminance image (g) using colorization.

parameter tuning. Moreover, our approach is generic and can be used with any existing SR technique.

The remainder of this paper discusses our SR color assignment procedure and demonstrates results on several examples using both reconstruction and learning-based techniques. The paper is concluded with a short discussion and summary.

2 Colorization Framework for Super Resolution

The pipeline of our approach is summarized in Fig. 3. Given a LR color image (Fig.3 (a)), our goal is to produce a SR color image (Fig.3 (h)). To achieve this goal, the input LR image is first decomposed into the luminance channel Y_L and the chrominance channels U_L and V_L . For simplicity, we use only the U channel to represent chrominance since the operations on the U and V channels are identical. For the luminance, the HR luminance channel Y_H is constructed from Y_L by using any preferred SR algorithm. To assign the RGB colors to the final SR image I_H, we use the colorization framework introduced by Levin et al. [29]. For the colorization, we introduce a method to generate chrominance samples which act as the seeds for propagating color to the neighboring pixels. The chrominance samples are obtained from the low resolution input, U_L, however the spatial arrangement of these chrominance values are generated automatically from the relationships between intensities in Y_L and Y_H.

Before we explain the colorization scheme, we note that we apply back-projection for computing Y_H from Y_L when the selected SR algorithm does not already include the back-projection procedure. We explain the reason for this first, before describing the colorization procedure.

2.1 Luminance Back-Projection

Enforcing the reconstruction constraint is a standard method which is used in many reconstruction based algorithms [9,10,11,12,13]. The difference among these various approaches is the prior imposed on the SR image. In our framework, the reconstruction constraint is enforced by minimizing the back-projection error of the reconstructed HR image Y_H against the LR image Y_L without introducing extra priors. This can be expressed as as:

$$Y_H = \arg\min_{Y_H} \| Y_L - (Y_H \otimes h) \downarrow \|^2, \tag{1}$$

where $\downarrow$ is the downsampling operator and $\otimes$ represents convolution with filter h with proportional to the magnification factor.

Assuming the term $Y_L - (Y_H \otimes h) \downarrow$ follows a Gaussian distribution, this objective equation can be cast as a least squares minimization problem with an optimal solution Y_H obtained by the iterative gradient descent method [5].

The reason to incorporate the reconstruction constraint is that the desired output should have the similar intensity values as the input image. As discussed in Section 1, learning-based techniques often suffer from luminance shifts due to training example mismatches. Conventional wisdom is that back-projection may remove hallucinated details, however, we found that adding this procedure had little effect on the synthesized details. Fig. 4 shows an example of the gradient histogram of the original Y_{SR} as more iterations of back-projection are applied. We can see that the gradient profiles exhibit virtually no change, while the color errors measured using the CIEDE200 metric against the ground truth are significantly reduced. This is not too surprising given that the estimated luminance image is downsampled in the back-projection process described in Eq. (1). Thus, back-projection is correcting luminance mismatches on the low-pass filtered image, allowing the fine details to remain. For SR techniques that already includes back-projection, this step can be omitted.

3 Colorization Scheme

The core of our approach lies in using image colorization to propagate the chrominance values from the LR input in order to add color to the upsampled SR luminance image. In [29], a gray-scale image is colorized by propagating chrominance values which are assigned via scribbles drawn on the image by the user. In our approach, the initial chrominance assignment comes from the LR image. The positions of these assignments are adjusted to better fit the HR luminance channel. We first review the image colorization and then describe our procedure to build the chrominance map.

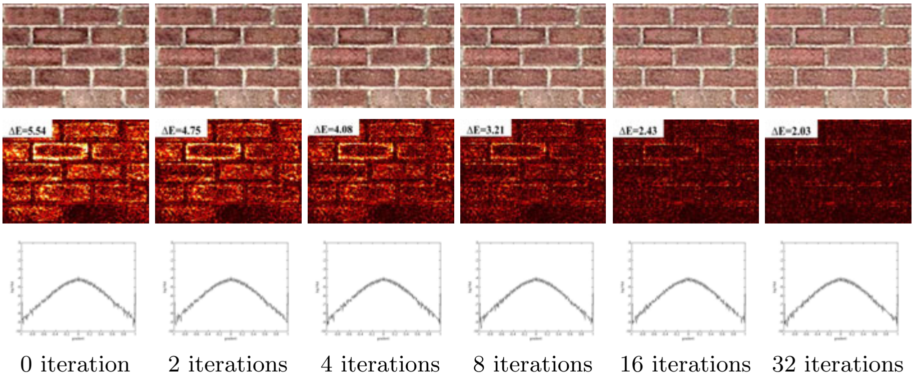


Fig. 4. Illustration of the benefits of back-projection. Estimated HR images (top), their CIEDE2000 color difference maps (middle), and gradient magnitude profiles (bottom) are shown at different iterations based on Eq. (1).

3.1 Image Colorization

Image colorization [29] computes a color image from a luminance image and a set of sparse chrominance constraints. The unassigned chrominance values are interpolated based on the assumption that neighboring pixels $\mathbf{r}$ and $\mathbf{s}$ should have similar chrominance values if their intensities are similar. Thus, the goal is to minimize the difference between the chrominance $U_H(\mathbf{r})$ at pixel $\mathbf{r}$ and the weighted average of the chrominance at neighboring pixels:

$$E = \sum_{\mathbf{r}} (U_H(\mathbf{r}) - \sum_{\mathbf{s} \in N(\mathbf{r})} w_{\mathbf{rs}} U_H(\mathbf{s})) \qquad (2)$$

where $w_{\mathbf{rs}}$ is a weighting function that sums to unity. The weight $w_{\mathbf{rs}}$ should be large when $Y_H(\mathbf{r})$ is similar to $Y_H(\mathbf{s})$, and small when the two luminance values are different. This can be achieved with the affinity function [29]:

$$w_{\mathbf{rs}} \propto e^{-(Y_H(\mathbf{r}) - Y_H(\mathbf{s}))^2 / 2\sigma_r^2} \qquad (3)$$

where σ_r is the variance of the intensities in a 3×3 window around $\mathbf{r}$. The final chrominance image is obtained by minimizing Eq. 2 based on the input luminance image and chrominance constraints. The final RGB image is computed by recombining the luminance and the estimated chrominance.

3.2 Chrominance Map Generation

To perform image colorization, chrominance values must be assigned to a set of pixels, or seed points, from which the color is propagated. In [29], scribbles from the user-input are used as the initial assignment of color. In this paper, the chrominance from the LR image is used for the initial color assignment. For

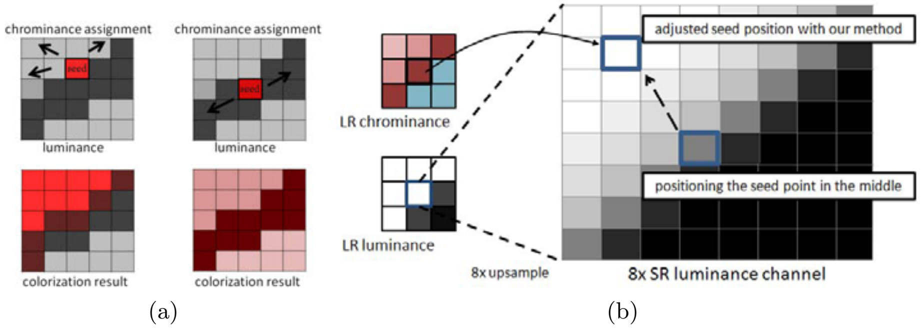


Fig. 5. (a) The effect of the chrominance seed position on the final colorization result are shown. The arrows indicate the chrominance propagation based on the intensity affinity based on the seed location. (b) Our aim is to adjust the seed point to be located at a position in the HR luminance result that is more similar the LR image luminance. This will produce a better colorization result.

example, for an 8× upsampling, a pixel in the LR image can be mapped to any of the pixels in the corresponding 8×8 block of corresponding HR pixels. The key in our colorization scheme lies in the positioning of the seed points in the upsampled image since blindly assigning the chrominance value to the middle of the patch may not produce the best result and can likely result in undesired color bleeding. This is illustrated in Fig. 5(a), where the we see that the estimated chrominance values are sensitive to the position of the seed point (i.e. hard constraint), especially on the edges.

Our strategy is to place the chrominance value in a position in the upsampled patch where the luminance value of the computed SR (Y_H) is closest to the original LR pixel's intensity (Y_L) as shown in Fig. 5(b). This approach, however, can be sensitive to noise and we therefore introduce a simple Markov Random Field (MRF) formulation to regularize the search direction for assigning the seed point. The idea is that the neighboring seed points are likely to share the same search direction in the HR image. Fig. 6 outlines the approach using an example with 8× upsampling.

The search directions are discretized into four regions (Fig. 6 (a)) which serve as the four labels of the MRF, i.e. $l_x \in \{0, 1, 2, 3\}$. Let $\mathbf{x}$ be a pixel coordinate in the LR image and $\mathbf{X}$ be the upsampled coordinate of the point $\mathbf{x}$. Let $N_i(\mathbf{X})$ be the neighborhood of $\mathbf{X}$ in the direction i, where $i \in \{0, 1, 2, 3\}$. A standard MRF formulation is derived as:

$$E = E_d + \lambda E_s, \tag{4}$$

where E_d is the data cost of assigning a label to each point $\mathbf{x}$ and E_s is the smoothness term representing the cost of assigning different labels to adjacent pixels. The term λ serves as the typical balancing weight between the data cost and the smoothness cost. Each cost is computed as follows :

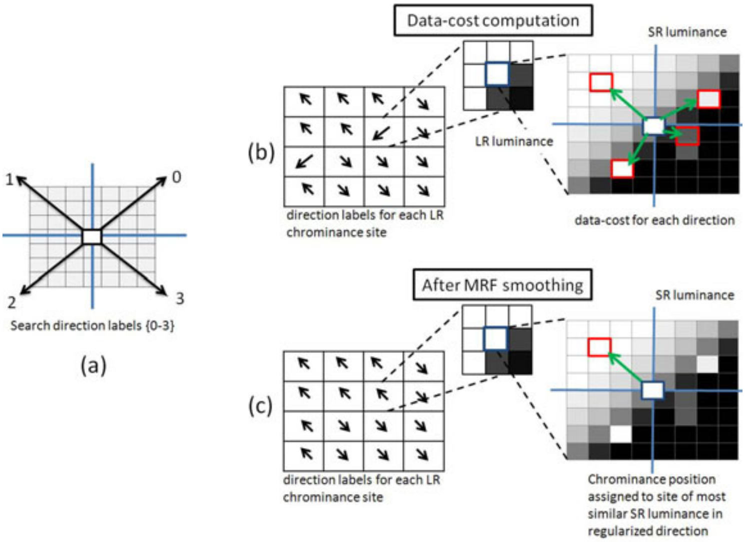

1
0
2
3
Search direction labels {0-3}
(a)
Data-cost computation
(b)
SR luminance
LR luminance
direction labels for each LR chrominance site
data-cost for each direction
After MRF smoothing
(c)
SR luminance
direction labels for each LR chrominance site
Chrominance position assigned to site of most similar SR luminance in regularized direction

Fig. 6. The MRF example: (a) Discretized search directions. (b) Data cost computation in each search direction. (c) Smoothness constraint to regularize results. The MRF smoothness prior regularizes the search direction to be similar to the search directions of neighboring LR pixels.

$$E_d(l_{\mathbf{x}} = i) = \min_{\mathbf{Z} \in N_i(\mathbf{X})} |Y_L(\mathbf{x}) - Y_H(\mathbf{Z})| , \tag{5}$$

and

$$E_s(l_p, l_q) = f(l_p, l_q) \cdot g(Y_{pq}), \tag{6}$$

where $f(l_p, l_q) = 0$ if $l_p = l_q$ and $f(l_p, l_q) = 1$ otherwise. The term $g(\xi) = \frac{1}{\xi+1}$ with $Y_{pq} = \|Y_L(p) - Y_L(q)\|^2$, where $\mathbf{p}$ and $\mathbf{q}$ are neighboring pixels. This weighting term encourages pixels with similar LR luminance intensity values to share the same directional label. The MRF labels are assigned using the belief propagation (BP) algorithm [31].

After computing the search direction using the MRF regularization, the chrominance value from the LR image is placed on the pixel with the most similar luminance value in the regularized search direction. Fig. 7 shows an example of the results obtained before and after applying the chrominance map adjustment. Bleeding is present without the adjustment, however, the results is much closer to the ground truth with the adjustment.

4 Experimental Results

Here we show results of our colorization scheme on 4 representative images shown in Fig. 8. For brevity, we only show the error maps and selected zoomed regions.

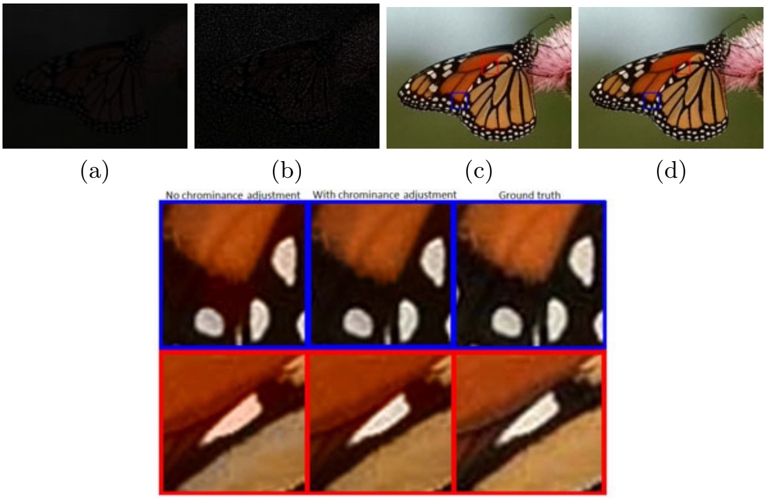


Fig. 7. (a) Initial color map U_S. (b) Color map U_H. (c) Colorization result using (a). (d) Colorization result using (b). Color map (b) produce better results without leakage at boundaries since the chrominance points are well located.

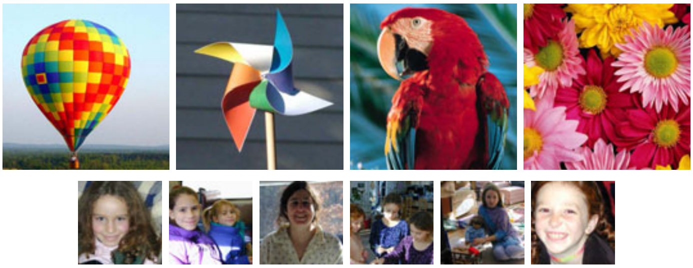


Fig. 8. (Top) Images used for our experiments. (Bottom) Images used as the training examples for the learning-based SR.

Full resolution images of our results, together with additional examples, are available online. For the color difference measure, we use the CIEDE2000 metric [26,27] together with a "hot" color-map. The mean color errors, ΔE, for all pixels as defined by the CIEDE2000 metric are provided.

The first two results are shown in Fig. 9 and Fig. 10. The images have been upsampled using 4× magnification using the recent reconstruction based SR algorithm in [13]. The results were produced with executable code available on the author's project webpage. Our colorization results are compared with the de facto UV-upsampling technique (also used in [13]). As can be seen, the overall error maps for our results are better. For the zoomed regions, we can see that artifacts about edges are less noticeable using our technique.

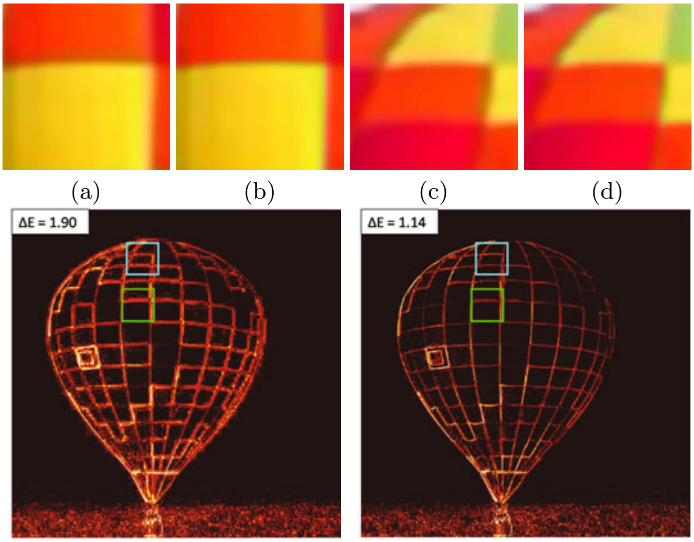


Fig. 9. Example 1 (Ballon): 4× reconstruction-based upsampling has been applied to the "ballon" image. UV-upsampling (a,c) is compared with our result (b,d).

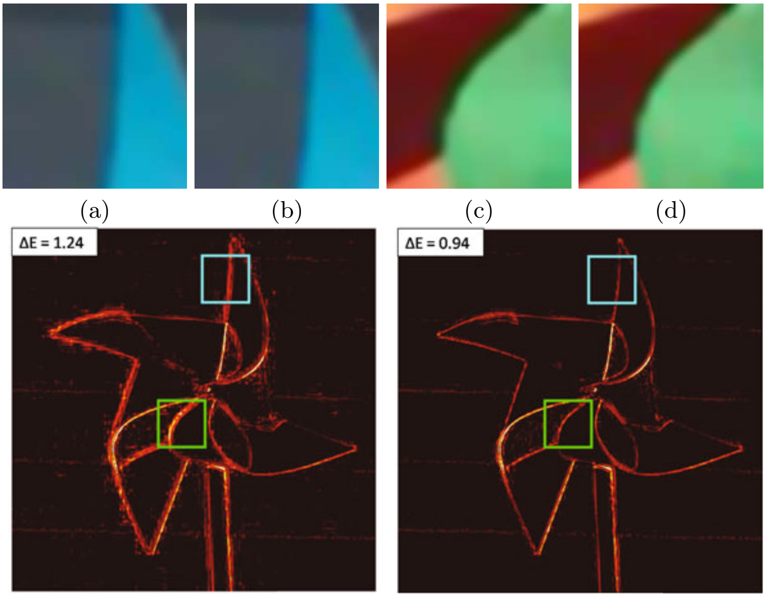


Fig. 10. Example 2 (Pinwheel): 4× reconstruction-based upsampling has been applied to the "pinwheel" image. UV-upsampling (a,c) is compared with our result (b,d).

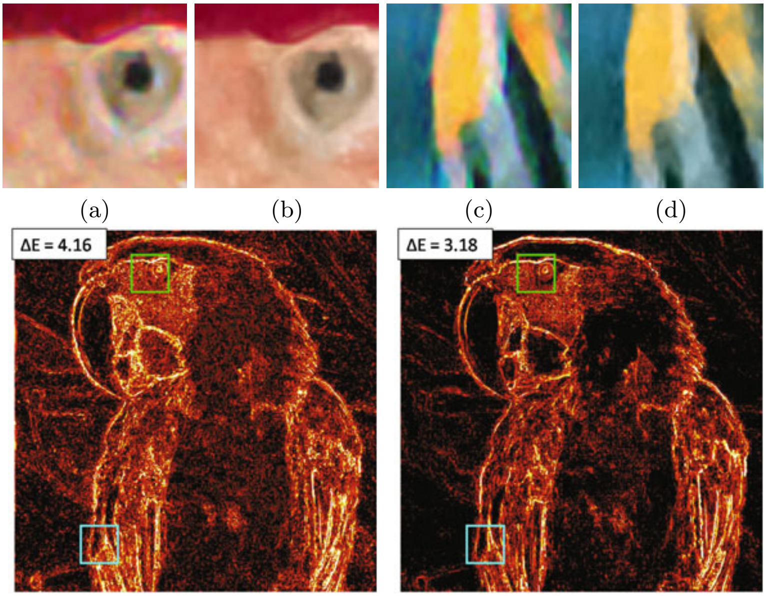


Fig. 11. Example 3 (Parrot): 4× learning-based upsampling (a,c) has been applied to the the "parrot" image. Full RGB SR is compared with our result (b,d).

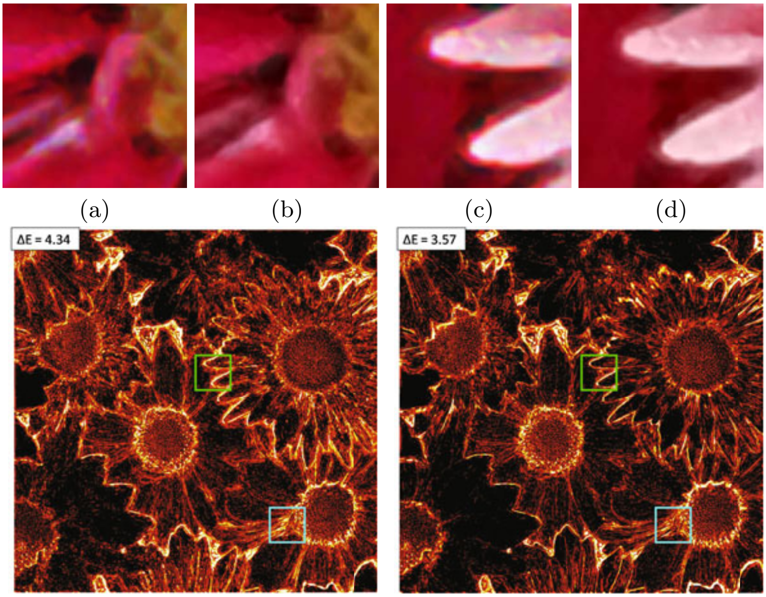


Fig. 12. Example 4 (Flowers): Example 2 (Parrot): 4× learning-based upsampling (a,c) has been applied to the the "parrot" image. Full RGB SR is compared with our result (b,d).

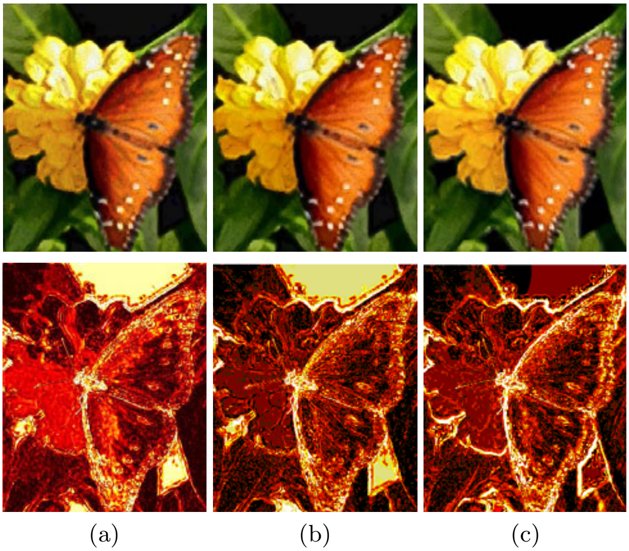


Fig. 13. Example showing the benefits of back-projection. (a) learning-based result; (b) our approach without back-projection; (c) our approach with back-projection.

The next two results are shown in Fig. 11 and Fig. 12. Fig. 8 (bottom) shows the training images used for the learning examples, which are the the same images used in the [16]. We use our own implementation of the full RGB learning method using the one-pass algorithm described in [16]. For our results, we first apply back-projection on the SR luminance channel before performing the colorization step. Learning-based techniques exhibit more random types of color artifacts, however, our approach is still able to improve the results as shown in the errors maps and zoomed regions.

The final example demonstrates the benefits of the optional back-projection procedure when the SR luminance image exhibits significant intensity shifting. In this example, only two of the training images are used to produce the SR image. Fig. 13(a) shows the result and the associated error. Fig. 13(b) shows our results obtained by only applying the colorization step and Fig. 13(c) shows the results when back-projection is used followed by our colorization method. We can see the error is significantly reduced when the back-projection procedure is incorporated.

5 Discussion and Summary

The focus of this paper is on assigning the final color values in the super resolution pipeline, and not how to perform SR itself. Therefore, our results are affected by the quality of the SR technique used, which is evident in the learning-based examples which tend to produce a higher overall error. However, even in these examples, our approach is able to offer a better final color assignment when compared with the ground truth. For reconstruction-based approaches, our overall edges appear sharper compared to basic UV-upsampling. We note that our approach inherits the limitations of image colorization. In particular, color bleeding may occur in regions with different chrominance but similar luminance values. However, the reasonably dense chrominance sampling from the LR image helps to keep such artifacts localized.

While we introduce an MRF regularization to aid in the chrominance map assignment, poor assignment of chrominance values can obviously result in undesired artifacts. Our quantitative measurements suggest our current approach is reasonable. We envision that better results could be obtained in the future with more sophisticated strategies for the chrominance placement.

In summary, we have introduced a new approach for assigning colors to SR images based on image colorization. Our approach advocates using back-projection with learning-based techniques and describes a method to adjust the chrominance values before performing image colorization. Our approach is generic and can be used with any existing SR algorithms.

References

1. Allebach, J., Wong, P.: Edge-directed interpolation. In: Proc. IEEE International Conf. on Image Processing, pp. 707–710 (1996)
2. Li, X., Orchard, M.T.: New edge-directed interpolation. In: Proc. IEEE International Conf. on Image Processing, pp. 311–314 (2000)
3. Caselles, V., Morel, J.M., Sbert, C.: An axiomatic approach to image interpolation. IEEE Trans. on Image Processing 7, 376–386 (1998)
4. Thevenaz, P., Blu, T., Unser, M.: Image Interpolation and Resampling. Academic Press, USA (2000)
5. Irani, M., Peleg, S.: Motion analysis for image enhancement: Resolution, occlusion, and transparency. Journal of Visual Communication and Image Representation 4, 324–335 (1993)
6. Morse, B., Schwartzwald, D.: Image magnification using level-set reconstruction. In: Proc. IEEE International Conf. Computer Vision, pp. 333–341 (2001)
7. Tappen, M.F., Russell, B.C., Freeman, W.T.: Exploiting the sparse derivative prior for super-resolution and image demosaicing. In: IEEE Workshop on Statistical and Computational Theories of Vision, pp. 2074–2081 (2003)
8. Lin, Z., Shum, H.: Fundamental limits of reconstruction-based superresolution algorithms under local translation. IEEE Trans. on Pattern Analysis and Machine Intelligence 26, 83–97 (2004)
9. Tai, Y.W., Tong, W.S., Tang, C.K.: Perceptually-inspired and edge directed color image super-resolution. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1948–1955 (2006)
10. Dai, S., Han, M., Xu, W., Wu, Y., Gong, Y.: Soft edge smoothness prior for alpha channel super resolution. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
11. Ben-Ezra, M., Lin, Z., Wilburn, B.: Penrose pixels: Super-resolution in the detector layout domain. In: Proc. IEEE International Conf. Computer Vision, pp. 1–8 (2007)

12. Sun, J., Sun, J., Xu, Z., Shum, H.: Image super-resolution using gradient profile prior. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
13. Shan, Q., Li, Z., Jia, J., Tang, C.K.: Fast image/video upsampling. ACM Trans. Graph. (Proc. of SIGGRAPH ASIA) 27, 1–7 (2008)
14. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. International Journal of Computer Vision 40, 25–47 (2000)
15. Liu, C., Shum, H.Y., Zhang, C.S.: Two-step approach to hallucinating faces: global parametric model and local nonparametric model. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 192–198 (2001)
16. Freeman, W.T., Jones, T., Pasztor, E.C.: Example-based super-resolution. IEEE Computer Graphics and Applications 22, 56–65 (2002)
17. Baker, S., Kanade, T.: Limits on super-resolution and how to break them. IEEE Trans. on Pattern Analysis and Machine Intelligence 24, 1167–1183 (2002)
18. Sun, J., Zheng, N.N., Tao, H., Shum, H.Y.: Image hallucination with primal sketch prior. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 729–736 (2003)
19. Yeung, D., Chang, H., Xiong, Y.: Super resolution through neighbor embedding. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 275–282 (2004)
20. Wang, Q., Tang, X., Shum, H.Y.: Patch based blind image super resolution. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 709–716 (2005)
21. Liu, C., Shum, H.Y., Freeman, W.T.: Face hallucination: Theory and practice. International Journal of Computer Vision 75, 115–134 (2007)
22. Tai, Y.W., Liu, S., Brown, M.S., Lin, S.: Super resolution using edge prior and single image detail synthesis. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2010)
23. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proc. IEEE International Conf. Computer Vision, pp. 349–356 (2009)
24. Jianchao, Y., John, W., Thomas, H., Yi, M.: Image super-resolution as sparse representation of raw image patches. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
25. Kopf, J., Cohen, M., Lischinski, D., Uyttendaele, M.: Joint bilateral upsampling. ACM Transactions on Graphics (Proc. of SIGGRAPH) 26 (2007)
26. Johnson, G.M., Fairchild, M.D.: A top down description of S-CIELAB and CIEDE2000. Color Research and Application 28, 425–435 (2002)
27. Sharma, G., Wu, W., Dalal, E.D.: The CIEDE2000 color difference formula: Implementations notes, supplementary test data and mathematical observations. Color Research and Application 30, 21–30 (2005)
28. Fattal, R.: Upsampling via imposed edges statistics. ACM Trans. Graph. (Proc. of SIGGRAPH) 26 (2007)
29. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. ACM Trans. Graph. (Proc. of SIGGRAPH) 23, 689–694 (2004)
30. Liu, X., Wan, L., Qu, Y., Wong, T.T., Lin, S., Leung, C.S., Heng, P.A.: Intrinsic colorization. ACM Transactions on Graphics (Proc. of SIGGRAPH Asia) 27, 152:1–152:9 (2008)
31. Tappen, M.F., Freeman, W.T.: Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In: Proc. IEEE International Conf. Computer Vision, pp. 900–907 (2003)

Programmable Aperture Camera Using LCoS

Hajime Nagahara[1], Changyin Zhou[2], Takuya Watanabe[3], Hiroshi Ishiguro[3], and Shree K. Nayar[2]

[1] Kyushu University
[2] Columbia University
[3] Osaka University

Abstract. Since 1960s, aperture patterns have been studied extensively and a variety of coded apertures have been proposed for various applications, including extended depth of field, defocus deblurring, depth from defocus, light field acquisition, etc. Researches have shown that optimal aperture patterns can be quite different due to different applications, imaging conditions, or scene contents. In addition, many coded aperture techniques require aperture patterns to be temporally changed during capturing. As a result, it is often necessary to have a *programmable aperture camera* whose aperture pattern can be dynamically changed as needed in order to capture more useful information.

In this paper, we propose a programmable aperture camera using a Liquid Crystal on Silicon (LCoS) device. This design affords a high brightness contrast and high resolution aperture with a relatively low light loss, and enables one change the pattern at a reasonably high frame rate. We build a prototype camera and evaluate its features and drawbacks comprehensively by experiments. We also demonstrate two coded aperture applications in light field acquisition and defocus deblurring.

1 Introduction

In the past decades, coded aperture techniques have been studied extensively in optics, computer vision and computer graphics, and a variety of coded aperture techniques have been proposed for various applications. The optimal aperture patterns can be quite different from one application to another. For defocus deblurring, coded apertures are optimized to be broad-band in the Fourier domain [1] [2]. For depth from defocus, coded apertures are optimized to have more zero-crossing frequencies [3] [4]. For multiplexing light field acquisition, an optimal set of aperture patterns are solved for the best signal-to-noise ratio (SNR) after de-multiplexing [5]. Aperture can also be coded in the temporal dimension for motion deblurring [6]. Coded aperture methods have also been used in many other applications, including lensless imaging [7] [8], natural matting [9], etc. Figure 1 shows a collection of some coded apertures that were proposed in the past.

There are many situations where the aperture pattern should be dynamically updated as needed. First, from the aspect of information capturing, ideally aperture pattern should be adaptive to scene contents. For example, the pattern should be optimized for defocus deblurring if the scene has a large depth, and

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 337–350, 2010.

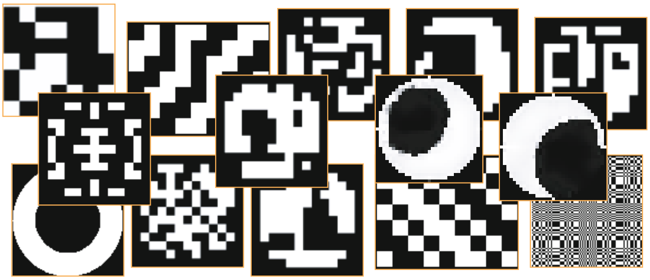
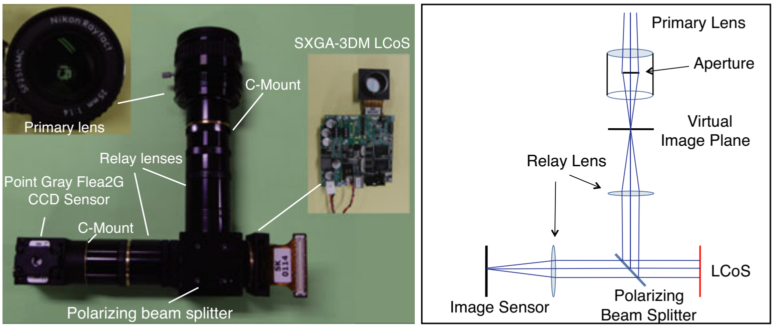


Fig. 1. A variety of coded aperture patterns proposed for various applications

it should be optimized for motion deblurring if the scene has many objects in motion. Secondly, aperture pattern should be adaptive to the specific application purpose. For example, people have shown that a coded aperture optimized for defocus deblurring is often a bad choice for depth from defocus [4]; and multiplexing light field acquisition technique requires different aperture codings for different target angular resolutions. Thirdly, the pattern should be adaptive to the imaging condition. For example, the optimal aperture pattern for defocus deblurring is different at different image noise levels as shown in [2]. In addition, some coded aperture techniques need to capture multiple images with different aperture patterns (e.g., [6] [4] and [5]). In all these situations, people need a *programmable aperture camera* whose aperture pattern can be updated at a reasonable speed.

In literature, people has used transmissive liquid crystal displays (LCD) to control aperture patterns [8] [5]. However, the LCD implementation has severe drawbacks. The electronic elements on LCD pixels occlude lights and lead to a low light efficiency. These occluders also cause strong and complicated defocus and diffraction artifacts. These artifacts can be very strong and eliminate the benefits of aperture codings. Consider the popular applications of coded aperture (e.g., defocus deblurring, depth from defocus), we argue that a good programmable aperture is necessary to have the following features:

1. Easy mount. For different applications or scenes, people may use different lenses and sensors. Therefore, it is important to build a programmable aperture that can be easily mounted to different lenses and sensors.
2. High light efficiency. The loss of light leads to decreased SNR. As shown in [2] [10], a high light efficiency is the key to achieve high performance in defocus deblurring, depth from defocus, multiplexing light field acquisition, etc.
3. Reasonable frame rate. Some coded aperture techniques capture multiple images of a scene using different aperture patterns [4] [5]. For dynamic scenes, these techniques require multiple images to be captured within a reasonable short time in order to reduce motion blur, and at the same time, the aperture pattern must also be updated at the same frame rate and be synchronized with the sensor exposure.



(a) Our prototype programmable aperture camera (b) The optical diagram of the prototype camera

Fig. 2. Programmable aperture camera using an LCoS device. (a) Our prototype LCoS programmable aperture camera. In the left-top corner is the Nikon F/1.4 25mm C-mount lens that is used in our experiments. On the right is an LCoS device. (b) The optical diagram of the proposed LCoS programmable aperture camera.

4. High brightness contrast. Most optimized aperture patterns in literature have high brightness contrast - many of them are binary patterns. We may fail to display optimized patterns without a high brightness contrast.

To meet these requirements, we propose in this paper a programmable aperture camera by using a Liquid Crystal on Silicon (LCoS) device as shown in Figure 2. LCoS is a reflective liquid crystal device that has a high fill factor (92%) and high reflectivity(60%). Compared with transmissive LCD, an LCoS device usually suffers much less from light loss and diffraction. Figure 2 shows the structure of our proposed programmable aperture camera. The use of LCoS device in our prototype camera enables us to dynamically change aperture patterns as needed at a high resolution (1280 × 1024 pixels), a high frame rate (5 kHz maximum), and a high brightness contrast. By using the relay optics, we can mount any C-Mount or Nikkon F-Mount lens to our programmable aperture camera. Remarkably, our implementation used only off-the-shelf elements and people may reproduce or even improve the design for their own applications.

A detailed description and analysis to our proposed system will be given in Section 3. The features and limitations of the present prototype camera are evaluated via experiments in Section 4. The proposed coded aperture camera can be a platform to implement many coded aperture techniques. As examples, in Section 5, we demonstrate the use of our prototype camera in two applications: multiplexing light field acquisition [5] and defocus deblurring [1] [2].

2 Related Work

Coded aperture technique was first introduced in the field of high energy astronomy in 1960s as a novel way of improving SNR for lensless imaging of x-ray and

γ-ray sources [11]. It is also in the 1960s that researchers in optics began developing unconventional apertures to capture high frequencies with less attenuation. In the following decades, many different aperture patterns were proposed (e.g., [12] [13] [14] [15] [7]).

Coded aperture research resurfaces in computer vision and graphics in recent years. People optimize coded aperture patterns to be broad-band in the Fourier domain in order that more information can be preserved during defocus for the later deblurring [1] [2]. Levin et al. [3] optimizes a single coded aperture to have more zero-crossing in the Fourier domain so that the depth information can be better encoded in a defocused image. Zhou et al. [4] show that by using the optimized coded aperture pair, they will be able to simultaneously recover a high quality focused image and a high quality depth map from a pair of defocused images. In the work [5], Liang et al. proposed to take a bunch of images using different coded aperture patterns in order to capture the light field.

Coded apertures have also been used for many other applications. Zomet and Nayar propose a lensless imaging technique by using an LCD aperture [16]. Raskar et al. uses a coded flutter shutter aperture for motion deblurring [6].

Coded aperture camera can be implemented in several ways. One popular coded aperture implementation is to disassemble the lens and insert a mask, which can be made of a printed film or even a cutted paper board [1] [3] [2]. The major disadvantages of this method are that one has to disassemble the lens, and that the pattern cannot be easily changed once the mask is inserted. Note that most commercial lenses cannot be easily disassembled without serious damages. People have also used some mechanical ways to modify apertures. Aggarwal and Ahuja propose to split the aperture by using a half mirror for high dynamic range imaging [17]. Green et al. build a complicated mechanical system and relay optics to split a circular aperture into three parts of different shapes [18].

To dynamically change aperture patterns during capturing, people has proposed to use transmissive liquid crystal display (LCD) devices as in the work [16] [5]. One problem with the LCD implementation is that the electronic elements sit in the LCD pixels not only block a significant portion of incoming light but also cause significant diffractions. Some custom LCDs are designed to have a higher light efficiency. However, these LCDs usually either have much low resolution (e.g., 5x5 pixels in [5]) or are prohibitively expensive. In this paper, we propose to use a reflective liquid crystal on silicon (LCoS) device [19], which has much higher light efficiency and suffers less from diffraction. LCoS has been used before in computer vision for high dynamic range imaging [20]. Another similar device that could be used to modulate apertures is the digital micro-mirror device (DMD). Nayar and Branzoi use a DMD device to control the irradiance to each sensor pixel for various applications, including high dynamic range and feature detection [21]. However, each DMD pixel only has two states and therefore DMD devices can only be used to implement binary patterns.

3 Optical Design and Implementation

We propose to implement a programmable aperture camera by using a liquid crystal on silicon (LCoS) device as an aperture. LCoS is a reflective micro-display technique typically used in projection televisions. An LCoS device can change the polarization direction of rays that are reflected by each pixel. Compared with the typical transmissive LCD technique, it usually produces higher brightness contrast and higher resolution. Furthermore, LCoS suffers much less from light loss and diffraction than LCD does. This is because the electronic components sitting on each pixel of LCD device block lights and cause significant diffraction, and on the contrary, an LCoS device has all the electronic components behind the reflective surface and therefore provides much higher fill factors.

One of our major design goals is to make the primary lens separable from the programmable aperture in order that people can directly attach any compatible lenses without disassembling the lens. To achieve this, we propose to integrate an LCoS device into relay optics.

As shown in Figure 2, our proposed system consists of a primary lens, two relay lenses, one polarizing beam splitter, an LCoS device, and an image sensor. Only off-the-shelf elements are used in our prototype camera implementation. We choose a Forth dimension display SXGA-3DM LCoS micro-display. Table 1 shows the specifications of this LCoS device. We use two aspherical doublet lenses of 50mm focal length (Edmund Optics, part #49665) for the relay optics, a cube polarizing beam splitter (Edmund Optics, part #49002), and a Point Grey Flea2 camera (1/3″ CCD, 1280x960 pixels at 25fps). The camera shutter is synchronized with the LCoS device by using an output trigger (25 Hz[1]) of the LCoS driver.

People have a plenty of freedom in choosing primary lenses for this system. The primary lens and the image sensor are attached to the optics via the standard C-mount. Therefore, a variety of C-mount cameras and lenses can be directly used with this prototype system. SLR lenses (e.g., Nikkon F-mount lenses) can also be used via a proper lens adopter. In our experiments, we use a Nikon 25mm F/1.4 C-mount lens.

We can see from Figure 2 (b) that an incoming light from a scene is first collected by the primary lens and focused at the virtual image plane. A cone of light from each pixel of the virtual image plane is then forwarded by the first relay lens to the polarizing beam splitter. The beam splitter separates the light into S-polarized and P-polarized (perpendicular to each other) lights by reflection and transmission, respectively. The reflected S-polarized light is further reflected by LCoS. The LCoS device can rotate the polarization direction at every pixel by arbitary degrees. For example, if the pixel on LCoS is set to 255 (8bit depth), the polarization of the light is rotated by 90 degree and becomes P-polarized, and then the light will pass through the splitter and reach to the sensor. If the pixel on LCoS is set to 0, the polarization will not be changed by LCoS and the reflected light will be blocked by the splitter.

[1] Note that the LCoS can be modulated at 5 kHz maximum. We use 25Hz in order that it can be synchronized with the sensor.

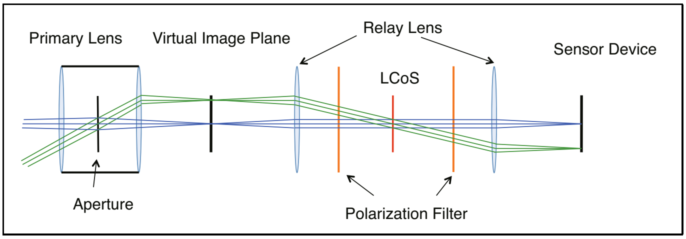


Fig. 3. An equivalent optical diagram to that in Figure 2 (b). The virtual image plane and the sensor plane are conjugated by the relay lens. The LCoS is the aperture stop of the system.

Consider the LCoS device as a mirror, the diagram in Figure 2 (b) can be easily shown equivalent to that in Figure 3. The proposed optics can be better understood from Figure 3. The sensor is located at the focal plane of the second relay lens, therefore the sensor plane is conjugate to the virtual image plane whose distance to the first relay lens is the focal length of the first relay lens. The LcoS device is relatively smaller than other stops in this optical system and works as the aperture stop.

4 Optical Analysis and Experimental Evaluation

Effective F-Number. Since the LCoS device works as the aperture stop in the proposed system, F-number ($f/\#$) of the primary lens is no longer the effective $f/\#$ of the camera. The actual $f/\#$ of the system is decided by focal length of the relay lens f_r and physical size of LCoS. For a circular aperture, $f/\#$ is usually defined as the ratio of focal length to the aperture diameter. For the rectangle nature of the LCoS, we use $2\sqrt{uv/\pi}$ as the diameter, where (u, v) is the dimension of LCoS. Therefore have:

$$f/\# = \frac{2}{f_r}\sqrt{\frac{uv}{\pi}}. \tag{1}$$

According to Equation 1, the effective $f/\#$ of the prototype can be computed as $f/2.84$, while the $f/\#$ of the primary lens is $f/1.4$.

Field of View. Figure 3 shows that the relay system copies the virtual image to sensor plane by a magnification ratio of $1 : 1$. Therefore, the field of view (FOV) of the proposed camera is the same as if the sensor were placed at the virtual image plane. The FOV can be estimated by using the sensor size and the effective focal length of the primary lens:

$$FOV \approx 2\arctan\frac{d}{2f_p}, \tag{2}$$

Table 1. Specification of LCoS device

Resolution	1280×1024 pixels
Reflective depth	8 bits
Pixel fill factor	>92%
Reflectivity	60%
Contrast ratio	400:1
Physical dimension	17.43×13.95 mm
Switching pattern	40 μs

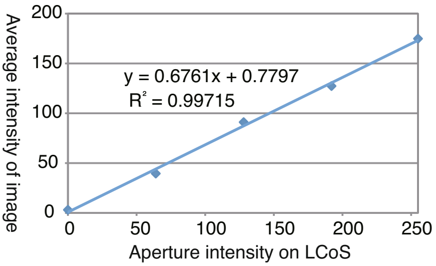


Fig. 4. The aperture transmittance is linear to the LCoS intensity

where d is a diagonal size of the sensor and f_p is the effective focal length of the primary lens.

Our prototype camera uses a 25mm lens and therefore the camera FOV can be computed as 13.69^o according to Equation 2. Of course, we can change the FOV by using a primary lens with a different focal length.

Light Efficiency. Light efficiency is one of the most important index in a coded aperture camera. Ideally, the light efficiency of our prototype camera is calculated by:

$$27.6\% = 50\%(polarization^{2}) \times 92\%(fillfactor) \times 60\%(reflectivity). \qquad (3)$$

We notice that many other optical elements in the camera (e.g., a beam splitter, two relay lenses, and an LCoS device) may also attenuate the intensity of captured images. To measure the light efficiency accurately, we captured two images of a uniformly white plane. One image was captured using our prototype camera, and another image was captured without the LCoS aperture (the same sensor and the same lens with $f/\#$ set to 2.8). The ratio of the averaged brightness of these two captured images is computed as 41.54:202.0, which indicates the light efficiency of the system. The light efficiency of our system is about 21%.

The theoretical light efficiency of a transmissive LCD[3] can also be calculated using a similar formula:

$$7.4\% = 50\%(polarization) \times 55\%(fillfactor) \times 27\%(transmittance). \qquad (4)$$

The light efficiency of our LCoS implementation is at least three times higher than that of the LCD implementation.

[2] A polarized beam splitter splits incoming lights based on their polarizations. Although the light interacts with the splitter twice, the light efficiency of beam splitter is still 50%. This is because 100% light will pass through the splitter at the second interaction when its polarization is aligned to that of the splitter.

[3] Note that the fill factor or transmittance of the LCD can be slightly different due to different implementations (e.g., physical sizes and resolutions). We assume a typical LCD with a similar physical size and resolution to the LCoS used in our implementation.

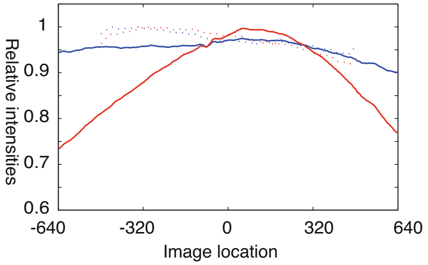


Fig. 5. Vignetting profiles. The red and blue solid lines indicate the horizontal vignetting curves of the prototype camera and a regular camera, respectively. The dashed lines indicate their vertical vignetting profiles.

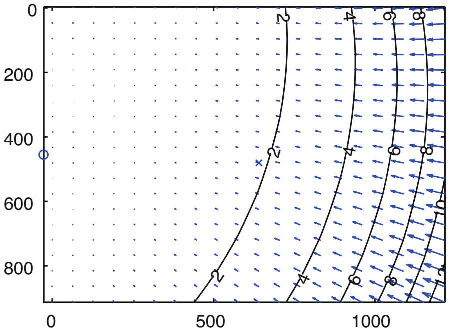


Fig. 6. Geometric distortion due to the use of doublet lenses

Vignetting. From the two images captured with and without the LCoS aperture, we can compute the vignetting curves of the prototype camera and a normal camera. The horizontal vignetting curves of our prototype camera and a normal camera are shown in Figure 5 in red and blue solid lines, respectively. The corresponding dashed lines show the vertical vignetting curves.

Transmission Fidelity. Another important quality index of a coded aperture implementation is the transmission fidelity – the consistence between the actual transmittance of coded aperture and the input intensity of the LCoS device. To evaluate the transmission fidelity, we captured images of uniformly white plane using circular apertures of different intensities. Figure 4 shows the line of the average intensity of captured images with respect to the input intensities of the circular aperture (implemented using LCoS device). This plot confirms that the aperture intensity is linear to the actual light transmittance rate. Also, by a linear regression, we can calculate the maximum contrast ratio is 221:1. Although this contrast is not as high as in Table 1, it has been high enough for most coded aperture applications.

Distortion. Another problem that has been caused by the use of doublets in the relay optics is image distortion. The geometric distortion is calibrated by using the Matlab camera calibration toolbox as shown in Figure 6. The circle indicates a center of distortion and the arrows represent displacements of the pixel introduced by the lens distortion. These calibrated camera parameters will be used to compensate the geometric distortions in the captured images.

PSF Evaluation. Lens aberration and diffraction may distort the actual PSFs. To assess the PSF quality of the prototype camera, we display a coded aperture and then calibrate the camera PSFs at 5 depths and 5 different view angles.

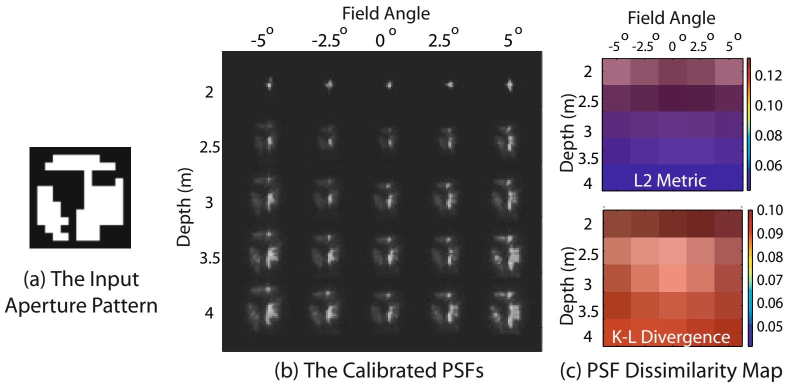


Fig. 7. Evaluating the PSFs of the prototype camera. (a) The coded aperture pattern used in the evaluation. This pattern is picked without specific intentions. (b) The calibrated PSFs at five depths ranging from 2m to 4m, and five field angles ranging from -5^o to 5^o. We can see that the scale of PSFs varies with both depth and field angle (due to field curvature), while the shape of PSFs appears similar. (c) The shape dissimilarity between the input pattern and each PSF is computed according to two metrics: L_2 distance at the top, and K-L divergence at the bottom (as used in the work [22]).

Table 2. Specification of the prototype camera

Image resolution	1280×960 pixels
Frame rate	25 fps
Minimum F-number	2.84
FOV(diagonal)	13.76^o (25 mm Nikkon C-mount)
Actual aperture contrast	221:1
Light transmittance	20.56%

Without specific intentions, we use the aperture pattern as shown in Figure 7 (a) in this evaluation. Figure (b) shows how PSFs varies with depth and field angle. We can see that the scale of PSF is related to the field angle. This is because the use of doublets in the relay optics leads to a field curvature.

We can see that the shapes of most PSFs are still very similar. To measure the similarity between these PSFs and the input aperture pattern, we normalize the scale of each PSF and compute its L_2 distance to the input pattern. A distance map is shown in the top of Figure 7 (c). We can see that according to the L_2 distance, the PSF shape deviation decreases as the blur size increases. It is known that L_2 distance is not a good metric to measure the PSF similarities in defocus deblurring. To measure the dissimilarity between two PSFs, we use the Wiener reconstruction error when an image is blurred with one PSF and then deconvolved with another PSF. This reconstruction error turns out to be a variant of K-L divergence as shown in the work [22]. We plot this dissimilarity map in the bottom of Figure 7 (c). We can see that all the dissimilarity values are small and decrease as the blur size increases.

The specifications of the prototype programmable aperture camera are shown in Table 2 as a summary.

5 Evaluation by Applications

5.1 Programmable Aperture for Light Field Acquisition

We first use our prototype programmable aperture camera to re-implement the multiplexing light field acquisition method, which is first proposed by Liang et al. [5]. A 4D light field is often represented as $l(u, v, x, y)$ [23], where (u, v) is the coordinates on the aperture plane and (x, y) is the coordinates in the image plane.

For a light field acquisition technique using coded aperture, the spatial resolution in the (x, y) space is simply determined by the sensor resolution and the angular resolution in the (u, v) space is determined the resolution of coded apertures. Bando et al. [9] use a 2x2 color coded aperture to capture light fields and then use the information to do layer estimation and matting. Liang et al. [5] propose a multiplexing technique to capture light fields up to 7×7 angular resolution. For any $m \times n$ angular resolution light field acquisition, the multiplexing method requires $m \times n$ images captured using $m \times n$ different coded apertures.

With our prototype programmable aperture camera, it is easy to capture light fields with various angular resolutions. We use S-matrix for the multiplexing coding (see [24] for a deep discussion on the multiplexing coding). Figure 8 (top) shows four of the 31 aperture patterns[4] that we generate from an S-Matrix. Since the aperture pattern of the prototype camera can be updated at a video frame rate (25 fps), it only takes 1.2 seconds to capture all of the images. If we could increase the camera frame rate further or lower the aperture resolution, the programmable aperture camera could be able to capture light fields of moving objects.

From the 31 captured images, we recover the light field of resolution $1280 \times 960 \times 31$ (7×5 (u,v) resolution excluding the four corners). Figure 9 shows the images for different viewpoints (u, v) and their close-ups. From the close-ups, we can see the disparities of the text clearly. With the recovered light field, people will be able to do further post-processing including depth estimation and refocusing as shown in [5] and [9].

5.2 Programmable Aperture for Defocus Deblurring

Another important limit of most existing coded aperture implementations is that the actual shape of the produced PSF often deviates from the input pattern due to lens aberration and diffraction. Note that the effects of lens aberration and diffraction can be quite different in different lenses. For the complexity

[4] This is because the code length of S-matrix must be $2^n - 1$.

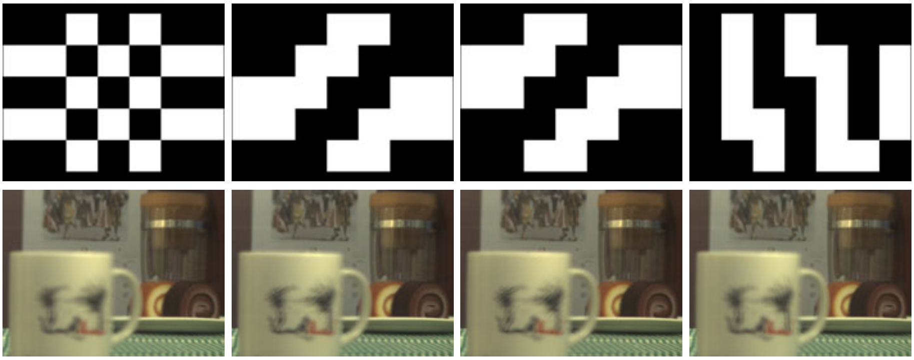


Fig. 8. Four multiplexing aperture codings and the corresponding captured images. Upper row shows four of the 31 aperture patterns that we generate from an S-Matrix. Bottom row shows the four corresponding captured images.

of the modern lenses, it is difficult to take these effects into account during pattern optimization. The effects of these imperfections on the optimality of the apertures are often overlooked in the literature.

With a programmable aperture camera, we will be able to evaluate the input aperture pattern by analyzing the captured images, and then improve the aperture patterns dynamically for a better performance. In this experiment, we apply this idea to the coded aperture technique for defocus deblurring.

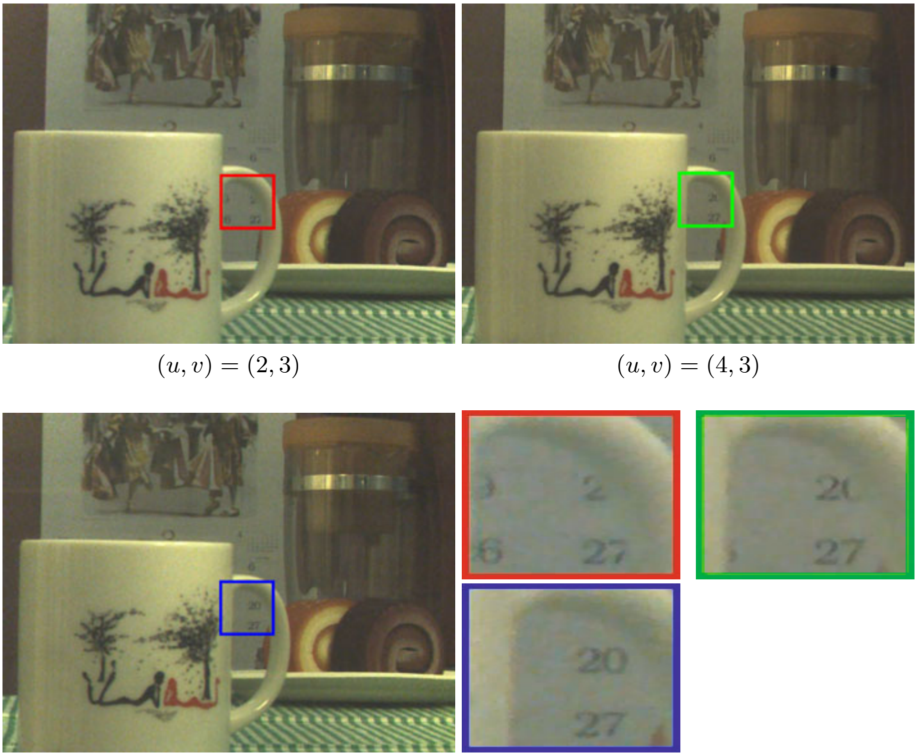
Zhou and Nayar [2] propose a comprehensive criterion of aperture evaluation for defocus deblurring, which takes image noise level, the prior structure of natural images, and deblurring algorithm into account. They have also shown that the optimality of an aperture pattern can be different at different noise levels and scene settings. For a PSF k, its score at a noise level σ is measured as:

$$R(K|\sigma) = \Sigma \frac{\sigma^2}{|K|^2 + \sigma^2/|F_0|^2}, \tag{5}$$

where K is the Fourier transform of the PSF k, and F_0 is the Fourier transform of the ground truth focused image. This definition can be re-arranged as

$$R(K|\sigma) = \Sigma \frac{\sigma^2 \cdot |F_0|^2}{|K|^2 \cdot |F_0|^2 + \sigma^2} \approx \Sigma \frac{\sigma^2 \cdot A}{|F|^2 + \sigma^2} \propto \Sigma \frac{A}{|F|^2 + \sigma^2}, \tag{6}$$

where A is the average power spectrum of natural images as defined in the work [2], and F is the Fourier transform of the captured image. Therefore, given a captured defocused image F, the equation 6 can be used to directly predict the quality of deblurring without calibrating the PSF and actually performing deblurring, while all the effects of aberrations and diffraction have been taken into account. Obviously, for the best deblurring quality, we should choose the aperture pattern which yields the lowest R value.



$(u, v) = (2, 3)$

$(u, v) = (4, 3)$

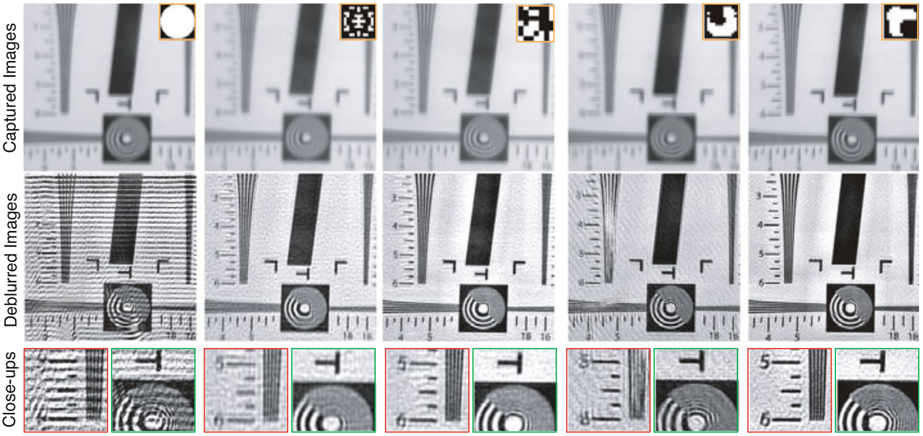
$(u, v) = (6, 3)$

Close-up images

Fig. 9. The reconstructed 4D light field. Images from three different view points (u, v) are generated from the reconstructed 4D light field, and their close-ups are shown in the right-bottom corner. From the close-up images, we can see the disparities of the text.

In our experiment, we capture a set of defocused images of an IEEE resolution chart (shown in the first row of Figure 10) by using the aperture patterns shown in Figure 1. We compute the R value from each captured image and find that the lowest R value is achieved by using the pattern shown in Figure 10 (e). This indicates that this pattern is the best among all these candidate patterns in the present imaging condition and scene settings.

Note that this prediction is made directly from the observed defocused images without PSF calibration or deblurring. The computation only involves few basic arithmetic operations and one Fourier transform, and therefore can be done at real time. For comparison, the second row of Figure 10 shows the deblurring results of several different aperture patterns. These results confirm that the pattern in (e) is the best for defocus deblurring in this particular image condition.



(a) Circular Pattern (b) Levin et al. (c) Veeraraghavan et al. (d) Zhou&Nayar σ=0.02 (e) Zhou&Nayar σ=0.03

Fig. 10. Pattern selection for defocus deblurring by using the programmable aperture camera. We capture a set of defocused images of an IEEE resolution chart using the patterns shown in Figure 1, and evaluate their qualities using Equation 6. The pattern shown in Column (e) is found to be the best according to our proposed criterion. To verify this prediction, we calibrate the PSFs in all the captured images, do deblurring, and show deblurring results (the second and third rows). We can see that the deblurring result in Column (e) is the best, which is consistent with the prediction.

6 Conclusion and Perspectives

In this paper, we propose to build a programmable aperture camera using an LCoS device which enables us to implement aperture patterns of high brightness contrast, light efficient and resolution at a video frame rate. Another important feature of this design is that any C-Mount or F-Mount lenses can be easily attached to the proposed camera without being disassembled. These features make our design applicable to a variety of coded aperture techniques. We demonstrate the use of our proposed programmable aperture camera in two applications: multiplexing light field acquisition and pattern selection for defocus deblurring.

We are aware that our prototype camera has many imperfections. For example, using two doublets to relay the lights has led to severe lens aberration, vignetting, and field curvature; and the light efficiency of the prototype camera is lower than that in design. How to optimize the optical design to minimize these imperfections will be our future work.

References

1. Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J.: Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. ACM Trans. Graphics (2007)

2. Zhou, C., Nayar, S.: What are good apertures for defocus deblurring? In: International Conference of Computational Photography, San Francisco, U.S (2009)
3. Levin, A., Fergus, R., Durand, F., Freeman, W.: Image and depth from a conventional camera with a coded aperture. Proc. ACM SIGGRAPH 26, 70 (2007)
4. Zhou, C., Lin, S., Nayar, S.: Coded Aperture Pairs for Depth from Defocus. In: Proc. International Conference on Computer Vision, Kyoto, Japan (2009)
5. Liang, C.K., Lin, T.H., Wong, B.Y., Liu, C., Chen, H.: Programmable aperture photography: Multiplexed light field acquisition. ACM Trans. Graphics 27 (2008)
6. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. ACM Trans. Graphics, 795–804 (2006)
7. Gottesman, S., Fenimore, E.: New family of binary arrays for coded aperture imaging. Applied Optics, 4344–4352 (1989)
8. Zomet, A., Nayar, S.: Lensless Imaging with a Controllable Aperture. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, p. 346. IEEE Computer Society, Los Alamitos (2006)
9. Bando, Y., Chen, B., Nishita, T.: Extracting depth and matte using a color-filtered aperture. ACM Trans. Graphics 27 (2008)
10. Hasinoff, S., Kutulakos, K., Durand, F., Freeman, W.: Time-constrained photography. In: Proc. International Conference on Computer Vision, pp. 1–8 (2009)
11. Caroli, E., Stephen, J., Cocco, G., Natalucci, L., Spizzichino, A.: Coded aperture imaging in X-and gamma-ray astronomy. Space Science Reviews, 349–403 (1987)
12. Welford, W.: Use of annular apertures to increase focal depth. Journal of the Optical Society of America A, 749–753 (1960)
13. Mino, M., Okano, Y.: Improvement in the OTF of a defocused optical system through the use of shaded apertures. Applied Optics, 2219–2225 (1971)
14. Varamit, C., Indebetouw, G.: Imaging properties of defocused partitioned pupils. Journal of the Optical Society of America A, 799–802 (1985)
15. Ojeda-Castafieda, J., Andres, P., Diaz, A.: Annular apodizers for low sensitivity to defocus and to spherical aberration. Optics Letters, 487–489 (1986)
16. Zomet, A., Nayar, S.: Lensless imaging with a controllable aperture. In: Proc. Computer Vision and Pattern Recognition, pp. 339–346 (2006)
17. Aggarwal, M., Ahuja, N.: Split Aperture Imaging for High Dynamic Range. International Journal of Computer Vision 58, 7–17 (2004)
18. Green, P., Sun, W., Matusik, W., Durand, F.: Multi-aperture photography. Proc. ACM SIGGRAPH 26 (2007)
19. Wikipedia: Liquid crystal on silicon, `http://en.wikipedia.org/wiki/Liquid_crystal_on_silicon`
20. Mannami, H., Sagawa, R., Mukaigawa, Y., Echigo, T., Yagi, Y.: High dynamic range camera using reflective liquid crystal. In: Proc. International Conference on Computer Vision, pp. 14–20 (2007)
21. Nayar, S.K., Branzoi, V., Boult, T.: Programmable imaging: Towards a flexible camera. International Journal of Computer Vision 70, 7–22 (2006)
22. Nagahara, H., Kuthirummal, S., Zhou, C., Nayar, S.: Flexible depth of field photography. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 60–73. Springer, Heidelberg (2008)
23. Levoy, M., Hanrahan, P.: Light field rendering. In: Proc. ACM SIGGRAPH, pp. 31–42 (1996)
24. Schechner, Y., Nayar, S., Belhumeur, P.: A theory of multiplexed illumination. In: Proc. International Conference on Computer Vision, vol. 2, pp. 808–815 (2003)

A New Algorithmic Approach for Contrast Enhancement

Xiaolin Wu and Yong Zhao

Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario, Canada L8S 4K1

Abstract. A novel algorithmic approach for optimal contrast enhancement is proposed. A measure of expected contrast and a sister measure of tone subtlety are defined for gray level transform functions. These definitions allow us to depart from the current practice of histogram equalization and formulate contrast enhancement as a problem of maximizing the expected contrast measure subject to a limit on tone distortion and possibly other constraints that suppress artifacts. The resulting contrast-tone optimization problem can be solved efficiently by linear programming. The proposed constrained optimization framework for contrast enhancement is general, and the user can add and fine tune the constraints to achieve desired visual effects. Experimental results demonstrate clearly superior performance of the new technique over histogram equalization.

1 Introduction

The contrast of a raw image can be far less than ideal, due to various causes such as poor illumination conditions, low quality inexpensive imaging sensors, user operation errors, media deterioration (e.g., old faded prints and films), etc. For better and easier human interpretation of images and higher perceptual quality, contrast enhancement becomes necessary and it has been an active research topic since early days of computer vision and digital image processing.

Contrast enhancement techniques can be classified into two approaches: context-sensitive or point-wise enhancers and context-free or point enhancers. In context-sensitive approach the contrast is defined in terms of the rate of change in intensity between neighboring pixels. The contrast is increased by directly altering the local waveform on a pixel by pixel basis. For instance, edge enhancement and high-boost filtering belong to the context-sensitive approach. Although intuitively appealing, the context-sensitive techniques are prone to artifacts such as ringing and magnified noises, and they cannot preserve the rank consistency of the altered intensity levels. The context-free contrast enhancement approach, on the other hand, does not adjust the local waveform on a pixel by pixel basis. Instead, the class of context-free contrast enhancement techniques adopt a statistical approach. They manipulate the histogram of the input image to separate the gray levels of higher probability further apart from the neighboring gray levels. In other words, the context-free techniques aim to increase

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 351–363, 2010.

the average difference between any two altered input gray levels. Compared with its context-sensitive counterpart, the context-free approach does not suffer from the ringing artifacts and it preserves the relative ordering of altered gray levels. This paper is mainly concerned with a rigorous problem formulation for context-free contrast enhancement, and accordingly it develops a general optimization framework to solve the problem.

Despite more than half a century of research on contrast enhancement, most published techniques are largely ad hoc. Due to the lack of a rigorous analytical approach to contrast enhancement, histogram equalization seems to be a widely accepted synonym for contrast enhancement in the literature and in textbooks of computer vision and image processing. The justification of histogram equalization as a contrast enhancement technique is heuristic, catering to an intuition. Low contrast corresponds to a biased histogram and thus can be rectified by reallocating underused dynamic range of the output device to more probable pixel values. Although this intuition is backed up by empirical observations in many cases, the relationship between histogram and contrast has not been precisely quantified.

There is no mathematical basis for the uniformity or near uniformity of the processed histogram to be an objective of contrast enhancement in general sense. On the contrary, histogram equalization can be detrimental to image interpretation if carried out mechanically without care. In lack of proper constraints histogram equalization can over shoot the gradient amplitude in some narrow intensity range(s) and flatten subtle smooth shades in other ranges. It can bring unacceptable distortions to image statistics such as average intensity, energy, and covariances, generating unnatural and incoherent 2D waveforms. To alleviate these shortcomings, a number of different techniques were proposed to modify the histogram equalization algorithm [1,2,3,4,5,6]. Very recently, Arici *et al.* proposed a histogram modification technique that first finds a histogram $\mathbf{h}$ in between the original input histogram $\mathbf{h}_i$ and the uniform histogram $\mathbf{u}$ and then performs histogram equalization of $\mathbf{h}$. The intermediate histogram h is determined by minimizing a weighted distance $\|\mathbf{h}-\mathbf{h}_i\|+\lambda\|\mathbf{h}-\mathbf{u}\|$. By choosing the Lagrangian multiplier λ the user can indirectly control undesirable side effects of histogram equalization. This latest paper also gave a good synopses of existing contrast enhancement techniques. We refer the reader to [7] for a survey of previous works, instead of reparaphrasing them here.

In our view, directly processing histograms to achieve contrast enhancement is an ill-rooted approach. The histogram is an awkward, obscure proxy for contrast. The popularity of histogram equalization as a context-free contrast enhancement technique is apparently because no mathematical definition of context-free contrast has ever been given in the literature. This paper fills the aforementioned long-standing void by defining a measure of expected context-free contrast of a transfer function, with this contrast measure being one if the input image is left unchanged. Furthermore, to account for the distortion of subtle tones caused by contrast enhancement, which is inevitable in most cases, a counter measure of tone subtlety is also introduced. The notions of expected contrast and tone subtlety give rise to a new perceptual image quality measure called contrast-tone ratio. The new measure sets an ideal objective for the enhancement of perceptual image quality, which seeks to achieve high contrast and subtle tone reproduction at the same time. But using the contrast-tone ratio as an objective function for maximization is computationally difficult because the function is highly nonlinear. Instead, we formulate contrast enhancement as a problem of maximizing the expected contrast subject to limits on tone distortion. Such a contrast-tone optimization problem can be converted to one of linear programming, and hence it can be solved efficiently in practice.

In addition, our linear programming technique offers a greater and more precise control of visual effects than existing techniques of contrast enhancement. Common side effects of contrast enhancement, such as contours, shift of average intensity, over exaggerated gradient, etc., can be effectively suppressed by imposing appropriate constraints in the linear programming framework. In the new framework, Gamma correction can be unified with contrast-tone optimization. The new technique can map L input gray levels to an arbitrary number Ł of output gray levels, allowing Ł to be equal, less or greater than L. It is therefore suited to output conventional images on high dynamic range displays or high dynamic range images on conventional displays with perceptual quality optimized for device characteristics and image contents.

Analogously to global and local histogram equalization, the new contrast enhancement framework allows the use of either global or local statistics when optimizing the contrast. However, in order to make our technical developments in what follows concrete and focused, we will only discuss the problem of contrast enhancement over an entire image instead of adapting to local statistics of different subimages. All the results and observations can be readily extended to locally adaptive contrast enhancement.

The remainder of the paper is organized as follows. In the next section we introduce some new definitions related to the intuitive notions of contrast and tone, and they lead to a new image quality measure called contrast-tone ratio. In section 3, we pose the maximization of the contrast-tone ratio as a problem of constrained optimization and develop a linear programming approach to solve it. In section 4 we discuss how to fine tune output images according to application requirements or users' preferences within the proposed contrast-tone optimization framework. Experimental results are reported in section 5, and they demonstrate the versatility and superior visual quality of the new contrast enhancement technique.

2 Contrast, Tone, and a New Perceptual Quality Measure

Contrast enhancement involves a remapping of input gray levels to output gray levels. In fact, such a remapping is required when displaying a digital image of L gray levels on a monitor of Ł gray levels, $L \neq \text{Ł}$. This remapping is carried out by an integer-to-integer transfer function

$$T : \{0, 1, \cdots, L-1\} \rightarrow \{0, 1, \cdots, \text{Ł}-1\} \tag{1}$$

The nature of the physical problem stipulates that the transfer function T be monotonically non-decreasing, because T should never reverse the order of intensities.[1] In other words, we must have $T(j) \geq T(i)$ if $j > i$. Therefore, any transfer function satisfying the monotonicity has the form

$$\begin{aligned} T(i) = \sum_{0 \leq j \leq i} s_j,\ 0 \leq i < L \\ s_j \in \{0, 1, \cdots, Ł - 1\} \\ \sum_{0 \leq j < L} s_j < Ł. \end{aligned} \tag{2}$$

The last inequality ensures the output dynamic range not exceeded by $T(i)$.

In (2), which is a general definition of the transfer function T, s_j is the increment in output intensity versus a unit step up in input level j. Therefore, s_j can be interpreted as context-free contrast at level j, which is the rate of change in output intensity without considering the pixel context. Note that a transfer function is completely determined by the vector $\mathbf{s} = (s_0, s_1, \cdots, s_{L-1})$, namely the set of contrasts at all L input gray levels.

Having associated the transfer function T with context-free contrasts s_j's at different levels, we induce from (2) a natural definition of expected (context-free) contrast of T for an image I:

$$C(\mathbf{s}) = \sum_{0 \leq j < L} p_j s_j \tag{3}$$

where p_j is the probability that a pixel in I has input gray level j.

The above defined expected contrast quantifies the colloquial meaning of contrast. To verify this let us examine some special cases.

Proposition 1. *The maximum expected contract $C(\mathbf{s})$ is achieved by $s_k = Ł - 1$ such that $p_k = \max\{p_i | 0 \leq i < L\}$, and $s_j = 0$, $j \neq k$.*

Proof: Assume for a contradiction that $s_j = n > 0$, $j \neq k$, would achieve higher expected contrast. Due to the constraint $\sum_{0 \leq j < L} s_j < Ł$, s_k equals at most $Ł - 1 - n$. But $p_j n + p_k(Ł - 1 - n) \leq p_k(Ł - 1)$, refuting the previous assumption. ■

Proposition 1 agrees with our perception that the highest contrast is achieved when the transfer function is a single step (thresholding) function that converts the input image from gray scale to binary. The binary threshold is set at level k such that $p_k = \max\{p_i | 0 \leq i < L\}$ for maximum expected contrast.

The lowest (zero) expected contrast is trivially achieved by a constant transfer function $T(i)$, namely $s_i = 0$ for all $0 \leq i < L$. Again, this agrees with our intuition of zero contrast.

In many applications it makes sense to preserve the average intensity while maximizing the expected contrast. In such cases, the average-preserving maximum expected contrast is achieved by $s_k = Ł - 1$, $s_j = 0$, $j \neq k$, such that

[1] This restriction may be relaxed in locally adaptive contrast enhancement. But in each locality the monotonicity should still be imposed.

$\sum_{0\leq j<k} p_j \approx \sum_{k\leq j<L} p_j$. Namely, $T(i)$ is the binary thresholding function at the average gray level.

If $L = \text{Ł}$ (i.e., when the input and output dynamic ranges are the same), the identity transfer function $T(i) = i$, namely, $s_i = 1$, $0 \leq i < L$, achieves expected contrast $C(\mathbf{1}) = 1$ regardless the gray level distribution of the input image. Therefore, the unit expected contrast means a neutral expected (context-free) contrast level without any enhancement. The notion of neutral contrast can be generalized to the cases when $L \neq \text{Ł}$. We call $\tau = \text{Ł}/L$ the tone scale. In general, the transfer function

$$T(i) = \left\lfloor \frac{\text{Ł}-1}{L-1} i + 0.5 \right\rfloor,\ 0 \leq i < L \tag{4}$$

or equivalently $s_i = \tau$, $0 \leq i < L$, achieves the neutral contrast $C(\tau\mathbf{1}) = \tau$. We note the following simple and yet important property of context-free contrast.

Proposition 2. *The* $\max\min\{s_0, s_1, \cdots, s_{L-1}\}$ *is achieved if and only if* $C(\tau\mathbf{1}) = \tau$, *or* $s_i = \tau$, $0 \leq i < L$.

Proposition 2 states that the simple linear transfer function, i.e., doing nothing in the traditional sense of contrast enhancement, actually maximizes the minimum of context-free contrasts s_i of different levels $0 \leq i < L$, and the neutral contrast $C(\tau\mathbf{1}) = \tau$ is largest possible when satisfying this maxmin criterion.

In terms of visual effects, smooth tone reproduction demands the transfer function to meet the maxmin criterion of proposition 1. This is because tone continuity requires small increment between adjacent gray levels to avoid contours or banding effects. Given a transfer function $T(i)$, define the tone subtlety of $T(i)$ as

$$\begin{aligned} \varPhi(\mathbf{s}) &= \max_{1\leq i\leq \text{Ł}} \left\{T^{-1}(i) - T^{-1}(i-1)\right\} \\ T^{-1}(i) &= \min\{j : T(j) = i\} \end{aligned} \tag{5}$$

In the definition we account for the fact that the transfer function $T(i)$ is not a one-to-one mapping in general. The smaller the value of $\varPhi(\mathbf{s})$ the smoother the tone reproduced by $T(i)$. It is immediate from the definition that the best achievable tone subtlety is $\tau = \min_{\mathbf{s}} \varPhi(\mathbf{s})$. But since the dynamic range Ł of the output device is finite, the two visual quality criteria of high contrast and tone continuity are in mutual conflict. Therefore, the mitigation of such an inherent conflict is a critical issue in designing contrast enhancement algorithms, which is seemingly overlooked in the existing literature on the subject.

Following the discussions above, a new perceptual image quality measure presents itself, which we call the contrast-tone ratio (CTR)

$$CTR = \frac{C(\mathbf{s})}{\varPhi(\mathbf{s})} \tag{6}$$

For the linear transfer function (4), $CTR = 1$ regardless of the intensity histogram of input image I. Also, if the input histogram is uniform then the highest possible CTR is 1, meaning that no further enhancement is possible. For a general input histogram, we are interested in finding the transfer function $T(i)$ that maximizes CTR, or achieves sharpness of high frequency details and tone subtlety of smooth shades at the same time.

3 Contrast-Tone Optimization by Linear Programming

In the proceeding section we formally defined the expected contrast $C(\mathbf{s})$ of a transfer function $T(i)$ on an image I. It also shown that the expected contrast is a good, meaningful measure of the overall contrast of an image. With the expected contrast $C(\mathbf{s})$ as a measurement of overall contrast one would attempt to perform contrast enhancement by finding the "optimal" transfer function $T(i)$, among all permissible ones, that maximizes C. But this single-minded approach would likely produce over-exaggerated, unnatural visual effects, as revealed by Proposition 1. The resulting $T(i)$ degenerates a continuous-tone image to a binary image. This maximizes the contrast of a particular gray level but completely ignores accurate tone reproduction.

In order to find a correct approach of improving visual quality it is helpful to model contrast enhancement as a problem of optimal resource allocation in competition with tone subtlety. The achievable expected contrast $C(\mathbf{s})$ and tone subtlety $\Phi(\mathbf{s})$ are physically confined by the output dynamic range Ł of the display. In (3) the optimization variables $s_0, s_1, \cdots, s_{L-1}$ represent an allocation of Ł available output intensity levels, each competing for a larger piece of dynamic range. While contrast enhancement necessarily invokes a competition for dynamic range (an insufficient resource), a highly skewed allocation of Ł output levels to L input levels can deprive some input gray levels of necessary representations. This causes unwanted side effects, such as flattened subtle shades, unnatural contour bands, shifted average intensity, and etc. Such artifacts were noticed by other researchers as drawbacks of the original histogram equalization algorithm, and they proposed a number of ad hoc. techniques to alleviate these artifacts while sticking to the baseline of histogram equalization.

As argued in the end of the proceeding section, a more principled solution of the problem is to maximize the contrast-tone ratio. Unfortunately, $C(\mathbf{s})/\Phi(\mathbf{s})$ is highly non-linear in $\mathbf{s}$. Instead of having $C(\mathbf{s})/\Phi(\mathbf{s})$ directly as the objective function, we develop a linear programming algorithm that maximizes $C(\mathbf{s})$ with linear constraints induced by $\Phi(\mathbf{s})$. Specifically, let us pose and examine the following constrained optimization problem:

$$\begin{aligned} &\max_{\mathbf{s}} \sum_{0 \le j < L} p_j s_j \\ &\text{subject to (a)} \sum_{0 \le j < L} s_j < \text{Ł}; \\ &\qquad\qquad\quad \text{(b) } s_j \ge 0,\ 0 \le j < L; \\ &\qquad\qquad\quad \text{(c)} \sum_{j \le i < j+\phi} s_i \ge 1,\ 0 \le j < L - \phi. \end{aligned} \tag{7}$$

In (7), constraint (a) is to confine the output intensity level to the available dynamic range; Constraints (b) ensure that the transfer function $T(i)$ be monotonically non-decreasing; Constraints (c) specify the coarsest level of tone subtlety $\Phi(\mathbf{s})$ allowed, where ϕ is an upper bound $\Phi(\mathbf{s}) \leq \phi$. The objective function and all the constraints are linear in $\mathbf{s}$.

Computationally, the original optimization problem of (7) is one of integer programming. This is because the transfer function $T(i)$ is an integer-to-integer mapping, i.e., all components of $\mathbf{s}$ are integers. But integer programming is NP-hard. To make the problem tractable we relax the integer constraints on $\mathbf{s}$ and convert (7) to a linear programming problem. By the relaxation any solver of linear programming can be used to solve the real version of (7). The resulting real-valued solution $\mathbf{s} = (s_0, s_1, \cdots, s_{L-1})$ can be easily converted to an integer-valued transfer function:

$$T(i) = \left\lfloor \sum_{0 \leq j \leq i} s_j + 0.5 \right\rfloor, \ 0 \leq i < L \tag{8}$$

For all practical considerations the proposed relaxation solution does not materially compromise the optimality. As a beneficial side effect, the linear programming relaxation simplifies constraint (c) in (7), and allows the contrast-tone optimization problem to be stated as

$$\begin{aligned} & \max_{\mathbf{s}} \sum_{0 \leq j < L} p_j s_j \\ \text{subject to} \quad & \sum_{0 \leq j < L} s_j < \text{Ł}; \\ & s_j \geq 1/\phi, \ 0 \leq j < L. \end{aligned} \tag{9}$$

4 Fine Tuning of Visual Effects

The proposed contrast-tone optimization framework is general and it can achieve desired visual effects by adding proper constraints to (9). We demonstrate the generality and flexibility of the proposed linear programming approach to image enhancement by some examples among many possible applications.

The first example is the integration of Gamma correction into contrast-tone optimization. The optimized transfer function $T(\mathbf{s})$ can be made close to the Gamma transfer function by adding to (9) the following constraint

$$\sum_{0 \leq i < L} \left| (L-1)^{-1} \sum_{0 \leq j \leq i} s_j - [i(L-1)^{-1}]^{\gamma} \right| \leq \mathit{\Delta} \tag{10}$$

where γ is the Gamma parameter and $\mathit{\Delta}$ is the degree of closeness between the resulting $T(\mathbf{s})$ and the Gamma mapping $[i(L-1)^{-1}]^{\gamma}$.

In applications when the enhancement process cannot change the average intensity of the input image by certain amount $\mathit{\Delta}_{\mu}$, the user can impose this restriction easily in (9) by adding another linear constraint

$$\left| \frac{L}{\text{\L}} \sum_{0\le i<L} p_i \sum_{0\le j\le i} s_j - \sum_{0\le i<L} p_i i \right| \le \mathit{\Delta}_\mu \tag{11}$$

Besides the use of constraints in the linear programming framework, we can incorporate context-based or semantics-based fidelity criteria directly into the objective function of contrast-tone optimization. The expected contrast $C(\mathbf{s}) = \sum p_j s_j$ and the CTR depend only on the point statistics of the input image. We can complement $C(\mathbf{s})$ and CTR by weighing in the semantic or perceptual importance of increasing the contrast at different gray levels by w_j, $0 \le j < L$. In general, w_j can be set up to reflect specific requirements of different applications. In medical imaging, for example, the physician can read an image of L gray levels on an Ł-level monitor, $\text{\L} < L$, with a certain range of gray levels $j \in [j_0, j_1] \subset [0, L)$ enhanced. Such a weighting function presents itself naturally if there is a preknowledge that the interested anatomy or lesion falls into the intensity range $[j_0, j_1]$ for given imaging modality. In combining point statistics and domain knowledge or/and user preference, we introduce a new objective function

$$\max_{\mathbf{s}} \left\{ \sum_{0\le j<L} p_j s_j + \lambda \sum_{0\le j<L} w_j s_j \right\} \tag{12}$$

where the Lagrangian multiplier λ regulates the relative importance of the expected contrast and a user-prioritized contrast.

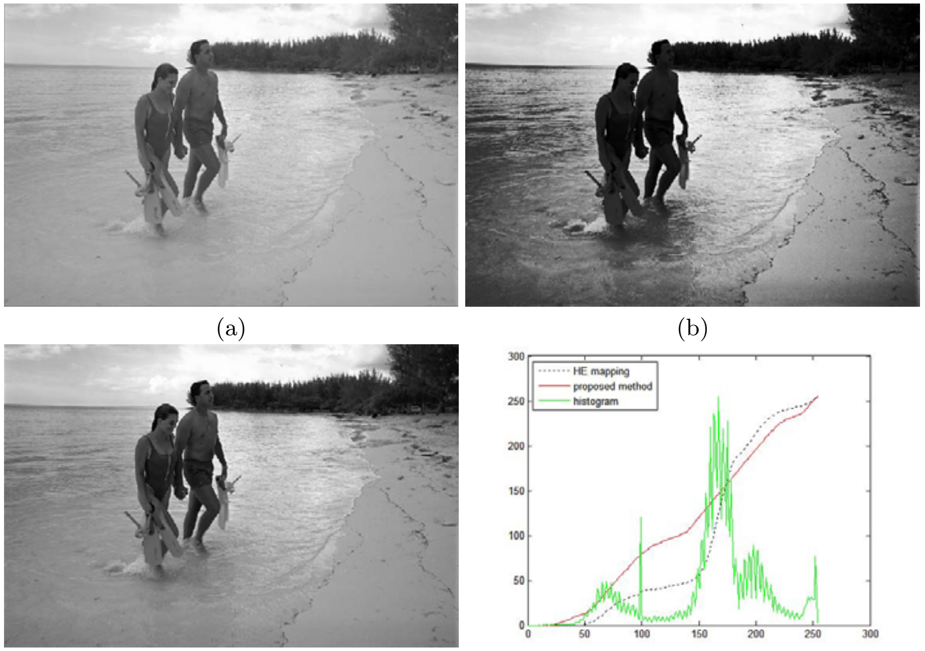
In summarizing all discussions above we finally present the following general linear programming framework for visual quality enhancement.

$$\begin{aligned}
&\max_{\mathbf{s}} \sum_{0\le j<L} (p_j + \lambda w_j) s_j \\
\text{subject to} \quad &\sum_{0\le j<L} s_j < \text{\L}; \\
&s_j \ge 1/\phi, \ 0 \le j < L; \\
&\sum_{0\le i<L} \left| (\text{\L} - 1)^{-1} \sum_{0\le j\le i} s_j - [i(L-1)^{-1}]^\gamma \right| \le \mathit{\Delta} \\
&\left| \frac{L}{\text{\L}} \sum_{0\le i<L} p_i \sum_{0\le j\le i} s_j - \sum_{0\le i<L} p_i i \right| \le \mathit{\Delta}_\mu .
\end{aligned} \tag{13}$$

5 Empirical Results

Fig. 1 through Fig. 4 present some sample images that are enhanced by the proposed contrast-tone optimization technique in comparison with those produced by histogram equalization. In addition to visual inspection we compare the two methods by the new image quality measure CTR as well in Table 1. As expected, in all cases the proposed technique achieves significantly higher CTR than histogram equalization.

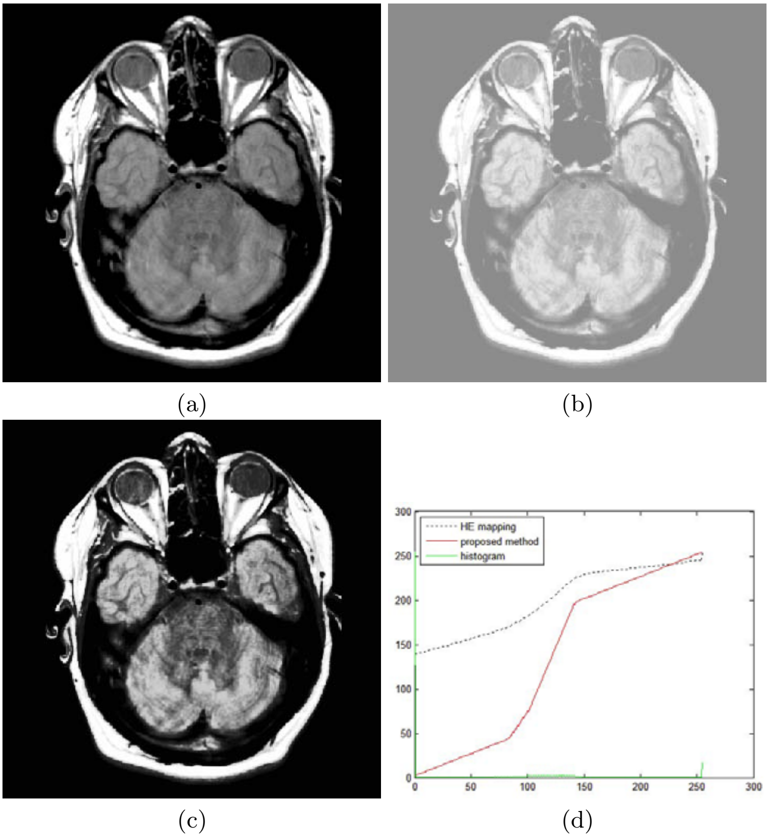
In image Beach (Fig. 1), the output of histogram equalization is too dark in overall appearance because the original histogram is skewed toward the bright range. But the proposed method enhances the original image without introducing unacceptable distortion in average intensity. This is because of the constraint that bounds the relative difference ($< 20\%$) between the average intensities of the input and output images. Fig. 2 shows an example when the user assigns higher weights w_j in (13) to gray levels j, $j \in (a, b)$, where $(a, b) = (100, 150)$ is a range of interest (brain matters in the head image). Fig. 3 compares the results of histogram equalization and the proposed method when they are applied to a typical portrait image. In this example histogram equalization overexposes the input image, causing an opposite side effect as in image Beach, whereas the proposed method obtains high contrast, tone continuity and small distortion in average intensity at the same time. In Fig. 4, the result of joint Gamma correction and contrast-tone optimization by the new technique is shown, and compared with those in difference stages of the separate Gamma correction and histogram equalization process. The image quality of the former is clearly superior to that of the latter.



(a) (b)

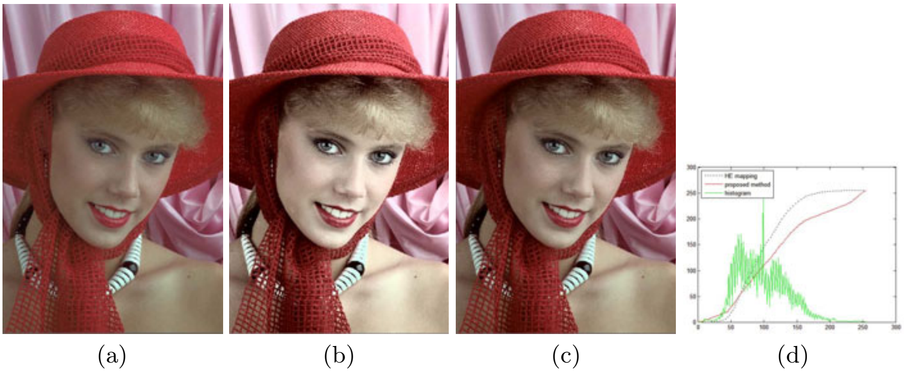
(c) (d)

Fig. 1. (a) the original, (b) the output of histogram equalization, (c) the output of the proposed method, and (d) the transfer functions and the original histogram



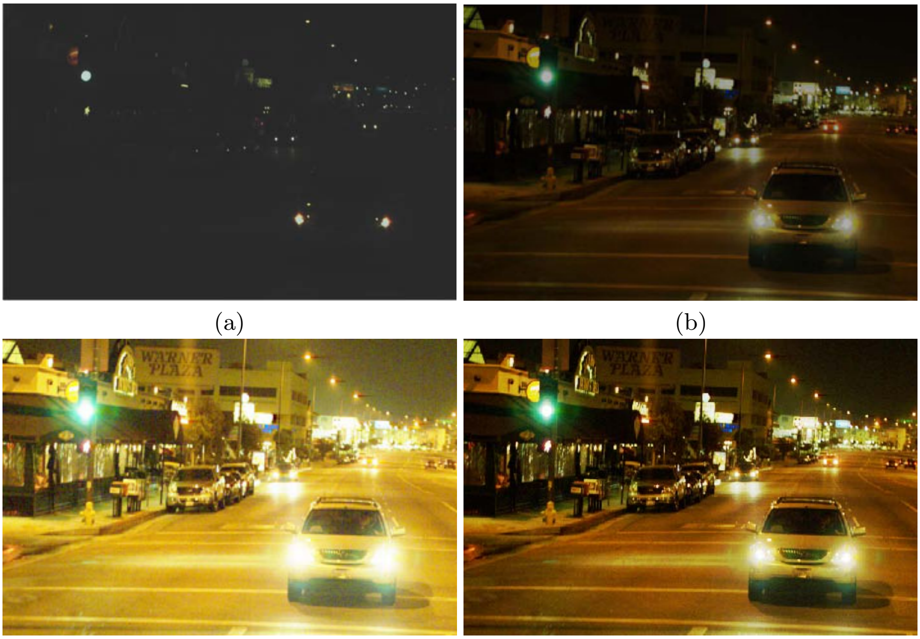
(a) (b) (c) (d)

Fig. 2. (a) the original, (b) the output of histogram equalization, (c) the output of the proposed method, and (d) the transfer functions and the original histogram



(a) (b) (c) (d)

Fig. 3. (a) the original, (b) the output of histogram equalization, (c) the output of the proposed method, and (d) the transfer functions and the original histogram



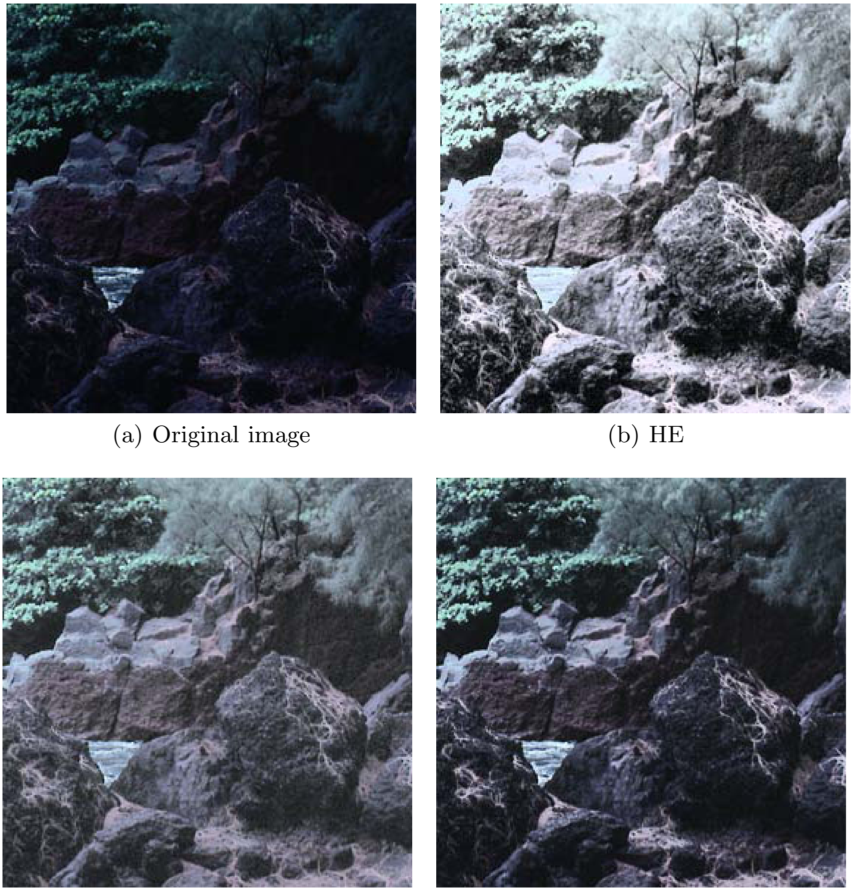
(a) (b) (c) (d)

Fig. 4. (a) the original image before Gamma correction, (b) after Gamma correction, (c) Gamma correction followed by histogram equalization, and (d) joint Gamma correction and contrast-tone optimization by the proposed method

Table 1. Comparison in CTR between histogram equalization and the proposed method

Image	Histogram equalization			Proposed method		
	Expected contrast	Tone Subtlety	CTR	Expected contrast	Tone subtlety	CTR
Beach	2.81	25	0.11	1.41	2	0.71
Head	0.58	8	0.07	0.73	2	0.36
Portrait	2.11	51	0.04	1.60	6	0.27

The proposed approach is also compared with the well-known contrast-limited adaptive histogram equalization (CLAHE) [8] in visual quality. CLAHE is considered to be one of the best contrast enhancement techniques, and it alleviates many of the problems of histogram equalization, such as over- or under-exposures, tone discontinuities, and etc. Fig. 5 is a side-by-side comparison of the proposed method, CLAHE and HE. CLAHE is clearly superior to HE in perceptual quality, as well recognized in the existing literature and among practitioners, but it is somewhat inferior to the proposed method in overall image quality, particularly in the balance of sharp details and subtle tones.



(a) Original image

(b) HE

(c) CLAHE

(d) The proposed

Fig. 5. Comparison of different methods on image Rocks

6 Conclusion

A new, general image enhancement technique of optimal contrast-tone mapping is proposed. The resulting problem can be solved efficiently by linear programming. The solution can increase image contrast while preserving tone continuity, two conflicting quality criteria that were not handled and balanced as well in the past.

References

1. Kim, Y.T.: Enhancement using brightness preserving bi-histogram equalization. IEEE Trans. Consum. Electronics 43, 1–8 (1997)
2. Wang, Y., Chen, Q., Zhang, B.: Image enhancement based on equal area dualistic sub-image histogram equalization method. IEEE Trans. Consum. Electronics 45, 68–75 (1999)

3. Gauch, J.M.: Investigations of image contrast space defined by variations on histogram equalization. In: Proc. CVGIP: Grap. Models Image Processing, pp. 269–280 (1992)
4. Stark, J.A.: Adaptive image contrast enhancement using generalizations of histogram equalization. IEEE Trans. Image Processing, 889–896 (2000)
5. Chen, Z.Y., Abidi, B.R., Page, D.L., Abidi, M.A.: Gray-level grouping(glg): An automatic method for optimized image contrast enhancement–part i: The basic method. IEEE Trans. Image Processing 15, 2290–2302 (2006)
6. Chen, Z.Y., Abidi, B.R., Page, D.L., Abidi, M.A.: Gray-level grouping(glg): An automatic method for optimized image contrast enhancement–part ii: The variations. IEEE Trans. Image Processing 15, 2303–2314 (2006)
7. Arici, T., Dikbas, S., Altunbasak, Y.: A histogram modification framework and its application for image contrast enhancement. IEEE Trans. Image Processing 18, 1921–1935 (2009)
8. Pisano, E.D., Zong, S., Hemminger, B., Deluca, M., Johnston, R.E., Muller, K., Braeuning, M.P., Pizer, S.: Contrast limited adaptive histogram image processing to improve the detection of simulated spiculations in dense mammograms. Journal of Digital Imaging 11, 193–200 (1998)

Seeing through Obscure Glass

Qi Shan, Brian Curless, and Tadayoshi Kohno

Department of Comptuer Science & Engineering, University of Washington
{shanqi,curless,yoshi}@cs.washington.edu

Abstract. Obscure glass is textured glass designed to separate spaces and "obscure" visibility between the spaces. Such glass is used to provide privacy while still allowing light to flow into a space, and is often found in homes and offices. We propose and explore the challenge of "seeing through" obscure glass, using both optical and digital techniques. In some cases – such as when the textured surface is on the side of the observer – we find that simple household substances and cameras with small apertures enable a surprising level of visibility through the obscure glass. In other cases, where optical techniques are not usable, we find that we can model the action of obscure glass as convolution of spatially varying kernels and reconstruct an image of the scene on the opposite side of the obscure glass with surprising detail.

1 Introduction

Obscure glass is a class of window glass used to separate spaces while allowing light and a limited amount of visual information to pass through. It is not uncommon to find such glass embedded, for instance, in office and conference room doors, in front doors of homes, in the exterior windows of bathrooms, and in the windows of close-proximity homes. Obscure glass is typically transparent, but has a surface texture that results in distorted and blurry images of a scene when photographing through the glass. The intent is to provide some degree of privacy, but how much privacy is actually provided?

In this paper, we explain the action of obscure glass and describe two complementary methods for seeing through it. First, we explore optical methods that include tuning the camera configuration, and show that, surprisingly, when the glass is textured only on the side facing the observer, household liquids may be used to cancel much of the obscuring effect of the glass. Indeed, it is possible to see through some visually impenetrable obscure glasses using a drop of honey and an inexpensive, compact video device, such as an iPod nano$^{\text{TM}}$. Second, we model the action of obscure glass as a spatially varying blur applied to a latent image. We then develop a calibration technique to recover this blur (assuming temporary access to both sides of the glass) which can be used to deblur a photo of a scene taken through the obscure glass at a later time, thus recovering the latent image of the scene.

Our contributions are threefold. First, to our knowledge, this is the first paper to define and address the problem of seeing through obscure glass. Second, the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 364–378, 2010.

optical methods, though building on related ideas in camera design and measurement of refractive substances, are novel in their application to minimizing optical degradation caused by obscure glass. Third, we have developed a new technique for recovering spatially varying blur kernels from small numbers of images by leveraging the sparsity of the kernel functions in this setting.

In the remainder of the paper, we review related work (Section 2), discuss the characteristics of obscure glass (Section 3), explore optical and calibrated deconvolution methods for seeing through it (Section 4), and conclude with results and discussion (Sections 5 and 6).

2 Related Work

Refractive index matching. One of our approaches to seeing through obscure glass is to reduce its distorting and blurring effect with a substance (nearly) matching its index of refraction. The idea of matching index of refraction is well-known as a tool for measuring the index of refraction of an irregularly shaped, transparent solid by immersing it in liquids whose refractive indices are known or readily measured [1]. This idea has also been applied to the problem of multi-view 3D volumetric reconstruction of refractive objects by computed tomography [2,3].

Environment matting. We also attempt to undo the effects of obscure glass by measuring the blurring and distorting properties of the glass. This measurement step is known as "environment matting" [4]. The idea is to shoot photographs through a refractive object, behind which is a changing background. In the original formulation, the background was a monitor displaying horizontal and vertical hierarchical stripe patterns, and per pixel filtering rectangles were recovered [4]. Follow-on work used different monitor patterns and filter representations: a single, smooth color wash pattern to recover refractive distortion for smooth objects [5], many images of a Gaussian stripe sweeping in different directions to recover multi-modal, oriented, Gaussian filters [5], and wavelet patterns to recover per-pixel wavelet filters [6]. An alternative approach uses a single, large background image that is moved around behind the refractive object, recovering per pixel filters, typically for smooth objects [7]. Agarwal et al. [8] recover distortion maps for refractive objects by analyzing their effect on the optical flow of a video being played in the background. These methods generally require many images or they impose restrictive assumptions on the blurring properties of the glass. Many images is problematic in that access to both sides of the glass may require a relatively quick capture process so as not to be excessively intrusive. Peers and Dutre [9] reduce the number of images using wavelet noise functions and a non-linear wavelet approximation; we employ similar patterns but explicitly encourage sparsity in the kernels during recovery. Our work builds most directly on environment matting, but it is also related to work on inverse light transport [10,11].

Also relevant is work in computer vision on recovering the shape of refractive surfaces. A nice survey can be found in [12]. Our approach is to recover blur kernels that can arise from fine (sub-pixel) texture, rather than recovering an explicit shape profile. The work of Murase [13] is particularly relevant in that an undistorted image is recovered as part of the shape estimation process, though the setting is different, as it depends on time-varying water motion.

Non-blind deconvolution. Recovering a latent image with a given image transform (blurring) matrix is known as non-blind image deconvolution. Recent research has yielded promising results using a natural image statistics prior and multi-scale techniques [14,15,16]. In this paper, we use a formulation similar to [15].

Security. The computer security community has several prior works studying "information leakage" via optical emanations. Kuhn found that the images on a CRT monitor can be reconstructed from reflections off a white wall [17]. Backes et al. extended this work to reconstruct images using reflections off diverse sets of objects, ranging from teapots to shirts [18,19]. Our work explores image reconstruction through refraction instead of reflections.

3 Characteristics of Obscure Glass

Obscure glass is plane glass with a surface texture – geometric perturbations – on at least one side of the glass. Light rays incident on the glass generally reflect and refract at the interface. The dominant visual effect when looking through obscure glass arises from refraction of viewing rays. If the glass is relatively smooth but wavy, the refraction results in a distorted image. If the glass has a finer texture, viewing rays can be significantly "scrambled" by refraction at the obscure glass surface, resulting in a blurred image of the scene.[1] Fig. 1(c) illustrates this effect and shows an image shot through an obscure glass sample; note the distortion and blur in this example.

Obscure glass varies in design and optical properties. The most noticeable variation is the geometric design of the surface texture, which can be locally smooth or rough, and exhibit various larger scale patterns for aesthetic appeal. Examples of obscure glass appear in Figs. 1 and 3-5. The surface perturbations are usually applied to one side of the glass, though glass with perturbations on both sides is not uncommon. When the texture is on one side, there is evidently no fixed rule as to which side will face outward when decorating offices and homes. Still, we note that for glass facing the outdoors, the flat side is often on the (dirtier) outside for ease of cleaning. Similarly, the flat side of a shower room door often faces toward the tub, for ease of cleaning mineral deposits, etc.

Finally, the refractive index of obscure glass, while nominally around 1.5-1.6, can vary according to the composition of the glass; manufacturers use different "recipes" for glass that results in this variation, sometimes using different recipes for different glasses in their own product lines.

[1] Taken to an extreme, the texture can be so rough as to completely scatter the rays, yielding no useful image; this kind of glass is called "frosted glass."

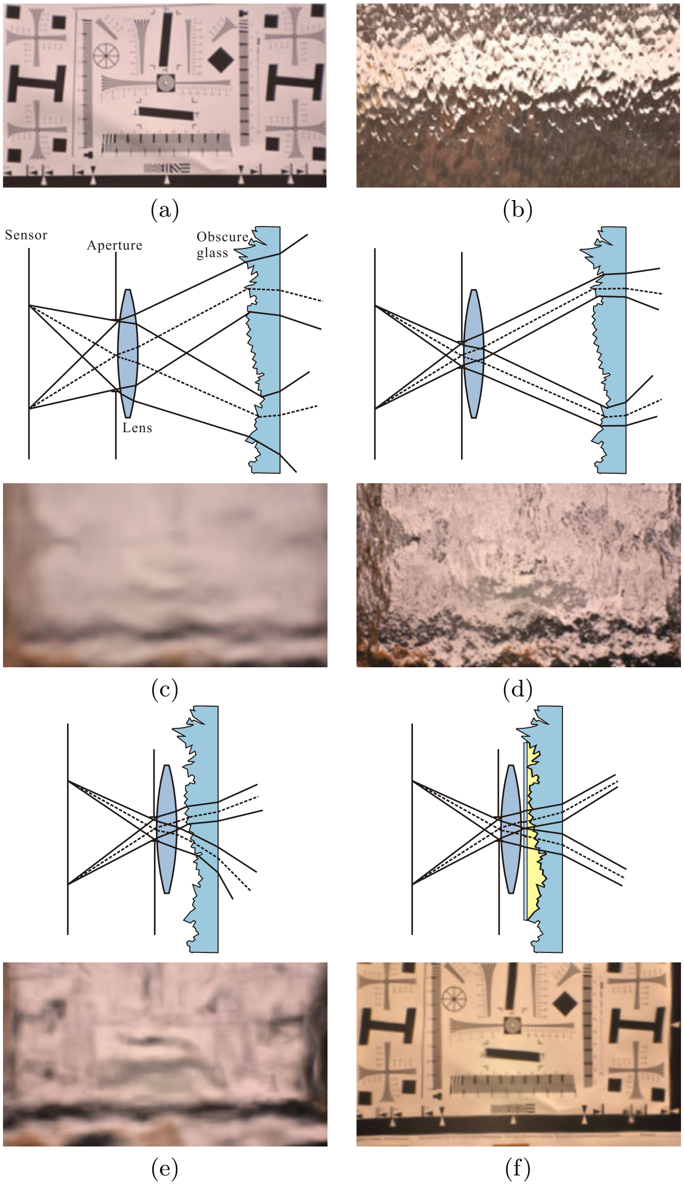


Fig. 1. A target scene (a) is photographed through a piece of obscure glass (b). A wide aperture (c) results in a blurry image, as the rays are broadly scattered. (The dotted lines represent chief rays; the solid lines are drawn to suggest the spread of refraction.) Narrowing the aperture (d) reduces the scatter, but severe distortion remains. Bringing the camera closer to the glass (e) yields a less distorted image. Interestingly, this image seems blurrier than the one in (d); this is because the distortion in (d) juxtaposes the blurred pixels at random, creating the false impression of higher frequencies. After applying a drop of liquid with refractive index close to that of the glass (pressed against the glass with a microscope slide cover slip), a nearly clear shot can be taken (f).

4 Seeing through Obscure Glass

In this section, we present two complementary approaches to the problem of trying to resolve a clear image of a scene observed through obscure glass: an optical approach to improve image clarity and a calibration approach for estimating and deconvolving blur kernels.

4.1 Optical Approach

To improve the photographic quality when shooting through obscure glass, we propose three optical strategies: small aperture, close camera placement, and applying a substance to the glass to reduce the refractive effect of the near side of the glass before imaging.

Small aperture. To reduce the blurring effect of obscure glass, the aperture can be stopped down (reduced in diameter), in effect restricting the range of scene points that will contribute to the light recorded at a point on the sensor. Fig. 1(d) illustrates the effect of reducing the aperture.

It is well-known that most imperfections (defocus, spherical aberration, etc.) in lens systems themselves can be reduced by stopping down the aperture. What remains is geometric distortion. Lens manufacturers go to great lengths to correct for most of these imperfections across a range of settings in advance, so that the aperture need not be stopped down to take a clear image of the in-focus plane. However, with obscure glass, even when we take a photo with a well-corrected lens, the image is degraded because we have introduced a new optical element – the obscure glass – that introduces potentially severe aberrations. Stopping down the aperture reduces the blurring aberrations, but distortion remains.

Note that blurring occurs as long as the cone of rays from the scene arriving at the aperture must pass through a locally “rough” region. The scale of the roughness that matters is thus really a function of the size of that cone. An undulating surface may seem smooth at millimeter scale, but if the aperture is wide open so that the cone of rays passing through the surface is at centimeter scale, then the result will be blur.

Camera placement. Stopping down the aperture will reduce image blur introduced by obscure glass, but some amount of distortion will likely remain. The reason for the distortion is that points on the sensor are recording narrow cones of light arriving from very different parts of the obscure glass, which can have very different (uncorrelated) surface orientations. To minimize the spread of cones across the glass, we can simply place the aperture (more precisely, the entrance pupil) as close to the glass as possible, as illustrated in Fig. 1(e).

It is worth noting that a variety of commonly available camera and lens solutions may be applied at this point, e.g., an SLR with a conventional lens stopped down as far as possible or with a specialty lens capable of very narrow apertures (high F-numbers) [20]. Ideally, the optics and camera placement would be tuned, if possible, to place the entrance pupil at the surface of the obscure glass. There is, however, a surprisingly simple and effective alternative. Very small cameras are becoming increasingly available in cell phones, webcams, and pocket video recorders. These cameras naturally have small apertures, and some of them can be placed very close to obscure glass, separated from the surface by a few millimeters or less. And, despite their small apertures, they have relatively low F-numbers, which means they can take (sometimes modest resolution) photos without requiring long exposures.

While placing the camera close to the glass can reduce the effect of distortion, a limitation is the fact that the camera itself becomes more noticeable from the opposite side of the glass, which makes the approach intrusive. However, this is not a concern when observing, e.g., a computer screen or documents in an office with no one inside. Further, a very small camera could be fairly unobstrusive; the camera (by itself) inside an iPod nano is already suggestively small.

Refractive index matching. Even with a closely positioned camera with a small aperture, if the surface has fine scale texture, then some amount of blurring and distortion will remain. When the texture is on the near side of the glass, then an unusual solution becomes possible. Recall Snell's law of refraction:

$$\eta_{\text{in}} \sin\theta_{\text{in}} = \eta_{\text{out}} \sin\theta_{\text{out}} \tag{1}$$

where θ_{in} and θ_{out} are the angles of the incident and refracted ray, respectively, taken with respect to the surface normal, and η_{in} and η_{out} are the indices of refraction on either side of the interface. The strength of refraction at the interface is controlled by how far the refractive index ratio $\eta_{\text{out}}/\eta_{\text{in}}$ is from unity. If we could smoothly "plaster over" the texture with a clear substance that had the exact same index of refraction as the glass, then the blurring, distorting effect of that texture would disappear. Fig. 1(f) illustrates this idea.

Precisely matching the index of refraction of obscure glass is challenging for several reasons. First, the index of refraction of glasses can vary depending on their compositions. Second, many of the standard, laboratory liquids used to cancel indices of refraction around $1.5 - 1.6$ are toxic. Finally, many of these liquids are very low viscosity; when applied to a vertical surface, they are difficult to contain, tending to run down the glass.

Instead, we propose to use non-toxic, high viscosity, household substances. For example, honey has proved to be very close in refractive index of glass and works well in experiments. The match is not exact, but combined with closely placed cameras with small apertures, visibility can improve significantly.

A critical limitation of this approach is the need for the textured surface to be on the near side of the glass. Still, for surfaces with texture on two sides, this approach can significantly reduce the degradation due to the obscure glass (cutting it "in half"). Further, if it is possible to deposit a (long lasting, ideally unnoticeable) substance on the far side of the glass in advance, then an optical portal is created.

4.2 Deconvolution Approach

In this section, we formulate the action of obscure glass as convolution of spatially varying kernels with a latent image and describe calibration methods for estimating these kernels and later recovering the latent image of an unknown scene. We note that this calibration scenario is plausible when access to both sides of the glass are available for a period of time, and then later, when only one-sided access is available, a degraded image of the scene is taken.

Image formation process. The image recorded at the sensor is a weighted sum of light rays scattering through the obscure glass and passing through the optical system of the camera. In principle, we need to recover a light field weighting function and latent light field within a small volume. To simplify the problem, we assume that the scene has constant depth and diffuse materials, or more loosely, minimal parallax and minimal view-dependent reflection with respect to the scattering of rays through a small portion of the obscure glass.

These assumptions allow us to model the scene as a latent image L and the formation of image I as a weighted sum of latent image pixels:

$$I = \mathbf{F}L + N, \tag{2}$$

where $\mathbf{F}$ is a degradation matrix whose rows correspond to spatially varying kernels, and N is sensor noise.[2] Note that $\mathbf{F}$ encodes both blur and distortion, and that L is generally larger in dimensions than I, i.e., $M_L > M_I$ where M_L and M_I are the number of pixels in L and I, respectively. Further, we assume the glass is not color tinted, and we do not model possible dispersion (wavelength dependent refraction), so that, while I and L are color-valued, $\mathbf{F}$ is not. Our goal is to recover $\mathbf{F}$ from measurements and then invert its effect to recover an image of a scene L restricted to L's overlap with I.

Recovering the degradation matrix F. We will recover $\mathbf{F}$ by recording a set of images $\mathcal{I} = [\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3 \ldots]$ through obscure glass in front of a known, changing background $\mathcal{L} = [\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3 \ldots]$. The image formation problem then becomes:

$$\mathcal{I} = \mathbf{F}\mathcal{L} + \mathcal{N}, \tag{3}$$

where $\mathcal{N}$ is sensor noise across the image set. Assuming independent and identically distributed Gaussian noise, the optimal $\mathbf{F}$ can be computed by minimizing:

$$E(\mathbf{F}) = \|\mathbf{F}\mathcal{L} - \mathcal{I}\|_2^2 \tag{4}$$

$\mathbf{F}$ is in general an $M_L \times M_I$ matrix; thus, in principle, M_L images are needed to estimate it, which could take a very long time to capture and process. By optimizing camera parameters, as described in the previous section, we have found that the filter support (and thus the number of free variables) in each row of $\mathbf{F}$ can be reduced significantly, e.g., to 100×100 for a 400×400 image. Still, an entirely brute force approach would require 10,000 images in this case.

To allow us to operate with fewer images (fewer observations than unknowns), we assume the blur kernels of the obscure glass are sparse in the spatial domain; i.e., we require most of the elements of each row of $\mathbf{F}$ to be zero. If the rows of

[2] We neglect lighting reflections from the camera side of the glass to the camera, which are minimized by placing the camera close to the glass.

F were instead dense, then the blur caused by them would be so severe as to make subsequent latent image recovery attempts essentially impossible. Thus, our assumption requires working with obscure glass and imaging setups where it actually *is* feasible to recover a useful image of the scene behind it.

We can encode a sparsity prior as a penalty on the L_1-norm of **F**, giving:

$$E'(\mathbf{F}) = \|\mathbf{F}\mathcal{L} - \mathcal{I}\|_2^2 + \gamma\|\mathbf{F}\|_1, \tag{5}$$

where γ is a weight to balance the data term and the prior term (set to 10^{-2} in all of our experiments). Thus, the problem is transformed into energy minimization with total variation regularization, solvable with existing techniques [15,21]. We note that this minimization can be performed independently for each row of **F**; i.e, the per pixel kernels can be estimated independently, and that each image pair provides three measurements (one per color channel) per pixel.

The obscure glass will not amplify or negate light, thus the elements $f_{i,j}$ of **F** must be in the range $[0, 1]$. Rather than impose these bounds directly, we take a simple (sub-optimal) approach of performing the optimization, then clamping all values $f_{i,j}$ to the range $[0, 1]$. Those values affected by the clamping step are then held constant and the optimization and clamping are repeated once more.

To reduce the number of variables and accelerate kernel estimation, we take a hierarchical approach that leverages the sparsity assumption. Specifically, we start with downsampled images and estimate kernels at the coarsest level where each kernel has at most 25 elements. After clamping, we then employ nearest neighbor upsampling to initialize the kernels at the next finer level. Any kernel variables at the finer level that are initialized to zero are held constant at zero. This process is repeated for each level. This approach reduces the required number of input images for two reasons. First, at the coarser levels, the size of the kernel is small enough to enable direct estimation with a small number of input image pairs. Second, at the finer levels, a large proportion of the pixels tend to be held constant at zero, thus reducing the number of unknowns, again requiring fewer images. See Fig. 2 for a visualization of the kernels.

We note that our solution method is tuned for speed, not optimality. Ignoring run time, we could explore slower methods that, e.g., exactly enforce constraints, and attempt to find the global optimum per pixel.

Image reconstruction process. After recovering **F**, we can now take a photograph of an unknown scene observed from the same viewpoint through the obscure glass and reconstruct the latent image L of the scene. The kernels in the rows of **F** generally perform some blurring, and it is well known that direct deconvolution of a blur filter (effectively inverting **F**) is highly sensitive to noise.

Recent deconvolution methods [14,15] have shown that regularizing priors based on natural image gradient statistics can significantly improve results. We follow the formulation of [15] which we summarize briefly here. In particular, we solve for the L and auxiliary function μ that minimizes

$$E(L, \mu) = \|I - \mathbf{F}L\|_2^2 + \lambda\Phi(\mu) + w\|\mu - \nabla L\|_2^2 \tag{6}$$

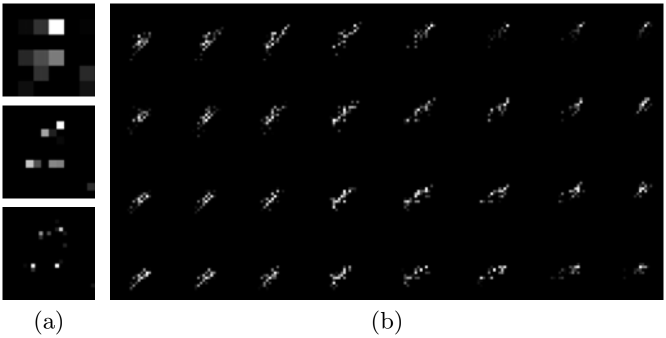


Fig. 2. Each kernel is estimated in a multi-scale fashion, shown coarsest to finest (top to bottom) in (a). In (b), we visualize a set of computed spatially varying kernels in a local neighborhood.

where $\mu = (\mu_x, \mu_y)$ is encouraged to be similar to ∇L (the gradient of L) through the third energy term, and

$$\Phi(\mu) = \sum_{i,j} \phi(\mu_x(i,j)) + \phi(\mu_y(i,j)). \tag{7}$$

The function $\phi()$ encodes the natural gradient prior. We refer the reader to [15] for the exact form of the prior. The energy is minimized following the procedure (and constant parameters) in that prior work, generally alternating between optimizing μ with fixed L and optimizing L with fixed μ. In [15], $\mathbf{F}$ corresponds to a spatially invariant blur, and solving for L can be performed with the Fast Fourier Transform. In our problem, the kernels are spatially varying, so we must modify the algorithm; we solve for L using an iterative, least squares, conjugate gradient solver, terminating when the ℓ^2-norm of the difference between consecutively computed L's is less than a threshold, set to 10^{-3} in our experiments.

5 Results

In this section, we first show how the physical imaging configurations affect the quality of the images captured through obscure glass. Then we show image reconstruction results using the calibration-based approach.

5.1 Optical Experiments

As discussed in Section 4.1, narrow apertures and close placement to the obscure glass gives the best imaging results. We experimented with four different camera configurations, listed here in order of decreasing aperture size and distance from the glass (due to physical limitations): a Nikon D90 SLR with an 18-105mm lens (set to 18mm and F/22), the same SLR with the LOREO Lens in a Cap [20] (35mm fixed focal length, F/64), an iPhone 3G camera, and an iPod nano video

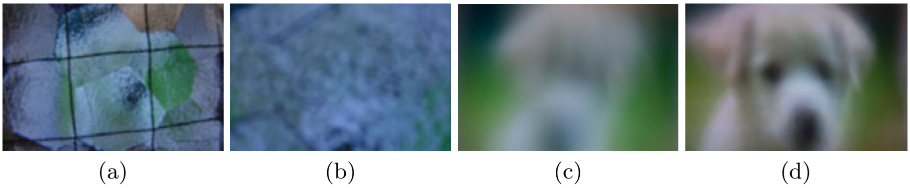


Fig. 3. Images captured using different cameras placed as close as possible to the obscure glass: (a) a Nikon SLR with 18-105mm lens, 18mm, F/22, (b) Nikon SLR with Lens in a Cap (35mm, F/64), (c) an iPhone 3G, and (d) an iPod nano video camera

camera. Fig. 3 shows images captured with these optical devices through a challenging piece of obscure glass. The tiny aperture of the iPod nano and the ability to place it closest to the glass leads to the clearest image. The nano's resolution is relatively low (640 × 480), but sufficient relative to the blurring caused by the glass in this example.

We also experimented with different substances to cancel the refractive properties of the glass as described in Section 4.1. Fig. 4 illustrates the results for one piece of obscure glass using the Nikon SLR and the Lens in a Cap. The elastomer used in [22] would have been particularly convenient; it is flexible enough to fill in the crevices of the glass and does not leave a trace. Unfortunately, its refractive index did not match the glass very well. Karo Syrup, Grade A clover honey, and wintergreen oil performed better; we applied each to a microscope slide cover slip and placed it against the obscure glass. In our tests with various glasses, wintergreen oil was often the best match, but it has very low viscosity and thus quickly runs out from under the cover slip, making it extremely difficult to use. Honey was the best compromise, as it tended to match the glasses reasonably well and has very high viscosity, so would tend to stay in place long enough to shoot photos and short videos. We refer the reader to the supplementary material for an example of combining small camera imaging (iPod nano) through obscure glass with an applied substance.

5.2 Image Deconvolution Experiments

In this section, we show experimental results for calibrating spatially varying kernels and recovering a latent image. To be as realistic as possible, we took the following steps. First, the textured side of the glass was oriented away from the viewer, thus was not susceptible to applying a refractive index matching substance. Next, images were taken of a controlled background – an LCD monitor – placed roughly 30 cm behind the glass. Then, the camera was removed and then replaced near the obscure glass, to simulate the effect of performing calibration at one time and later having to reposition the camera to take a shot of a the scene. The repositioning was done manually, and multiple shots were taken in an attempt to collect an image similar in viewpoint to the original. In the experiments, a handful of images works surprisingly well for the glass we tested.

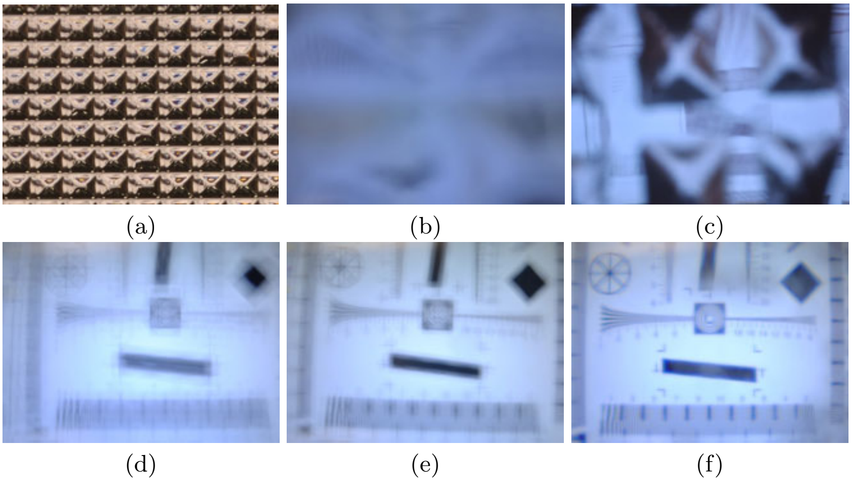


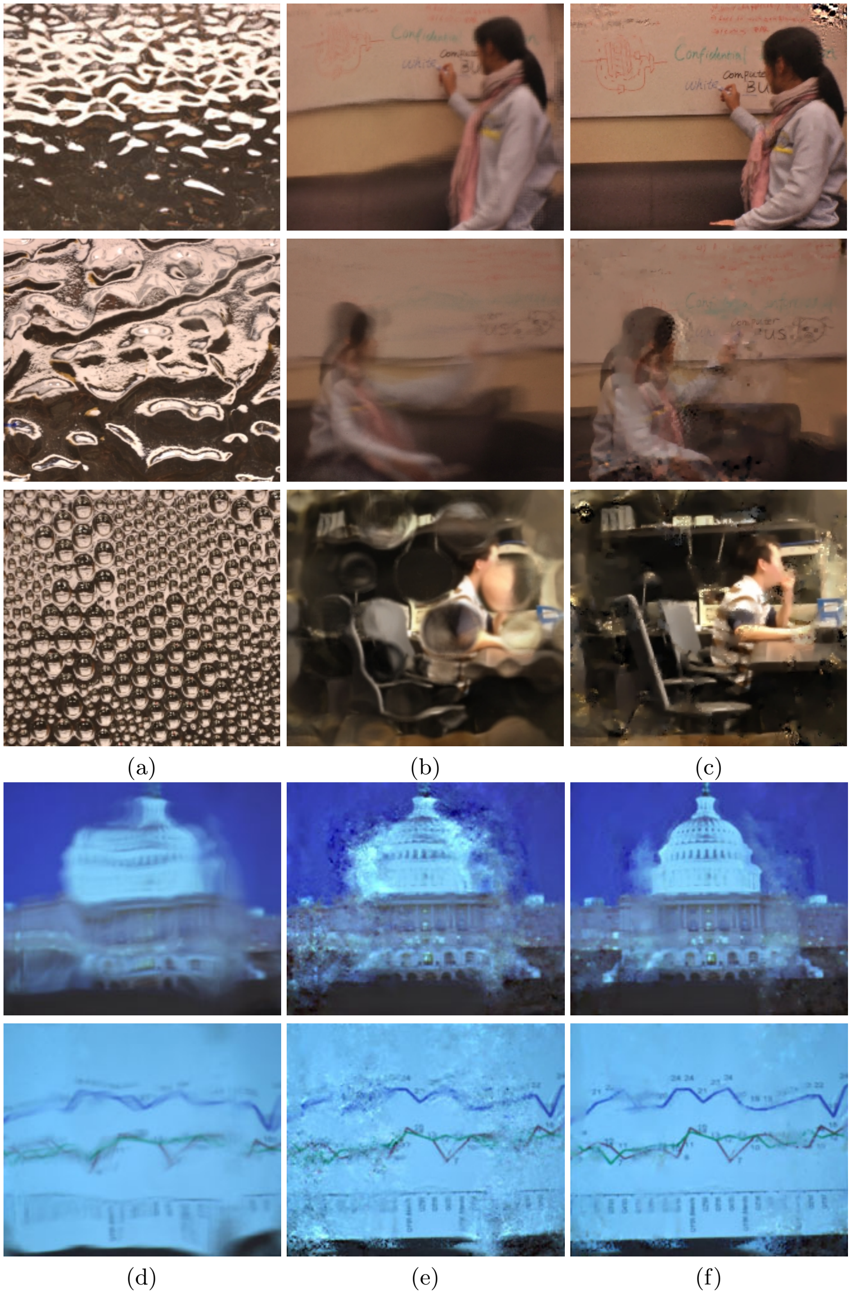
Fig. 4. Images captured with different kinds of liquid substances added to the obscure glass (a). Substances included: (b) no substance, (c) elastomer [22], (d) Karo Syrup, (e) Grade A clover honey, (f) wintergreen oil.

Finally, the calibration background was photographed without the obscure glass, and these images were aligned to the calibration images taken through the obscure glass. Precise alignment of the calibration image sets was not necessary, since the kernels being recovered can accommodate pixel offsets. The calibration pattern was 150 different instances of Perlin noise [23] (a different pattern per color channel, as well), which has both low and high frequency content. We used the Nikon SLR, as it was easier to maintain constant exposure settings for it than the iPhone 3G and iPod nano.[3] Images were taken with the 18-150mm lens at F/22 or the Lens in a Cap at F/64 using a tripod with exposures lasting several seconds. We note that, regardless of the imaging set-up, there will exist an obscure glass that introduces problematic degradations; our goal here is to demonstrate how much clearer an image can become given those degradations.

Fig. 5 shows the results for latent image recovery using several different obscure glasses. In each case, we show the best recovered image among the repositioned set, since these images were shot with the understanding that likely only one would be at the right position. The top three rows demonstrate the ability to recover latent images of non-planar scenes, with kernels restricted to 45 × 45 in size. Where distortion was large at the boundaries, we did not have sufficient calibration image coverage to recover the kernels, resulting in some artifacts.[4] The second row exhibits some artifacts due to repositioning error.

[3] The iPod nano had the added complication of recording compressed video, which resulted in severe blocking artifacts when compressing the noise patterns.

[4] We pad image boundaries of captured images using pixel replication as needed.



(a) (b) (c)

(d) (e) (f)

Fig. 5. Image reconstruction results with glass calibration. Column (a): three kinds of obscure glasses. Column (b): images captured through glasses in column (a). Column (c): reconstructed images with 45 × 45 kernels. Column (d) images captured through the obscure glass in Fig. 1 (b). Column (e): reconstructed images with 45 × 45 kernels. Column (f): reconstructed images with 95 × 95 kernels.

The bottom two rows correspond to a conference room scenario, where the photographs are taken of two consecutively projected slides on a presentation screen, as seen through obscure glass. This example illustrates the importance of using sufficiently large kernels. Severe artifacts are apparent for 45×45 kernels, with much better results obtained for 95×95 kernels.

We found that working with numbers of input images in the range of 100-200 generally worked well; not surprisingly, going much lower than this range degraded the results. Solving for larger kernel sizes was generally preferable, at the expense of increased compute time. We also note that better results are possible without camera or monitor repositioning; we intentionally made the problem more challenging for the sake of realism.

Our kernel estimation procedure is fairly slow. For a 400×400 image, compute time is 40 CPU-hours for 45×45 kernels and over 200 CPU-hours for 95×95 kernels. The method is at least trivially parallelizable; we ran all of our computations on a cluster of 150 CPUs in under 2 hours in the worst case. High performance was obviously not our goal, though a notable area for future work. Deconvolution, by contrast, requires only one minute per 400×400 image.

6 Discussion and Future Work

We have posed the problem of trying to see through obscure glass, and developed both optical and calibration-based software techniques to do so. In our experience, the most effective solution is to apply a refractive index matching substance to the glass, essentially nullifying the effect of the glass when the match is exact. The match is, however, not always exact; further, when the textured surface is facing away from the viewer, it may not be feasible or desirable to leave a substance on that surface in real-world scenarios. Thus, it becomes important to undo the distorting and blurring effects of obscure glass. Our calibration approach is one way to accomplish this, and we have found it to be fairly effective when the blur and distortion are not extreme relative to the resolving capabilities of the optical setup. When the blur kernels are dense with large support, then it becomes difficult or impossible to recover a meaningful image. Thus, for privacy purposes, obscure glasses with dense blur kernels are preferable.

Our calibration-based approach is somewhat complex, requiring fairly careful re-positioning of the camera to recover a reasonably clear image, especially important for complex glass surface geometry. This step could be improved, e.g., by bracing the camera physically against the frame of the glass in a controlled manner. Another limitation of this approach is the required access to both sides of the glass at some point in time. An area of future work is to perform this step by taking a set of images of the natural scene on the other side and blindly recovering the latent images and the blur kernels. This scenario might be plausible if, for instance, one is attempting to recover images from a monitor, television, or projection screen on which slides or video are playing, providing a set of distinct images. Another future avenue would be to consider multiple viewpoints, moving the camera across the glass, and estimating the distorting and blurring structure of the glass, e.g., a height field and index of refraction.

Acknowledgements

We thank Steve Seitz and Sameer Agarwal for early discussions and ongoing feedback, and our sponsors, the University of Washington Animation Research Labs, Microsoft, Adobe, and Pixar.

References

1. Lide, D.R.: CRC Handbook of Chemistry and Physics, 90th edn. (2009)
2. Sharpe, J., Ahlgren, U., Perry, P., Hill, B., Ross, A., Hecksher-Sørensen, J., Baldock, R., Davidson, D.: Optical projection tomography as a tool for 3d microscopy and gene expression studies. Science 296, 541–545 (2002)
3. Trifonov, B., Bradley, D., Heidrich, W.: Tomographic reconstruction of transparent objects. In: Proc. Eurographics Symp. on Rendering, pp. 51–60 (2006)
4. Zongker, D.E., Werner, D.M., Curless, B., Salesin, D.H.: Environment matting and compositing. SIGGRAPH, 205–214 (1999)
5. Chuang, Y.Y., Zongker, D.E., Hindorff, J., Curless, B., Salesin, D., Szeliski, R.: Environment matting extensions: towards higher accuracy and real-time capture. SIGGRAPH, 121–130 (2000)
6. Peers, P., Dutré, P.: Wavelet environment matting. In: Rendering Techniques, pp. 157–166 (2003)
7. Wexler, Y., Fitzgibbon, A.W., Zisserman, A.: Image-based environment matting. In: Rendering Techniques (2002)
8. Agarwal, S., Mallick, S.P., Kriegman, D.J., Belongie, S.: On refractive optical flow. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3022, pp. 483–494. Springer, Heidelberg (2004)
9. Peers, P., Dutré, P.: Inferring reflectance functions from wavelet noise. In: Rendering Techniques, pp. 173–182 (2005)
10. Seitz, S.M., Matsushita, Y., Kutulakos, K.N.: A theory of inverse light transport. In: ICCV (2005)
11. Sen, P., Chen, B., Garg, G., Marschner, S.R., Horowitz, M., Levoy, M., Lensch, H.P.A.: Dual photography. SIGGRAPH 24 (2005)
12. Ihrke, I., Kutulakos, K.N., Lensch, H.P.A., Magnor, M., Heidrich, W.: State of the art in transparent and specular object reconstruction. In: STAR Proceedings of Eurographics (2008)
13. Murase, H.: Surface shape reconstruction of a nonrigid transparent object using refraction and motion. TPAMI, 1045–1052 (1992)
14. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. SIGGRAPH (2007)
15. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. SIGGRAPH (2008)
16. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Progressive inter-scale and intra-scale non-blind image deconvolution. SIGGRAPH (2008)

17. Kuhn, M.G.: Optical time-domain eavesdropping risks of CRT displays. In: IEEE Symp. on Security and Privacy (2002)
18. Backes, M., Dürmuth, M., Unruh, D.: Compromising reflections – or – how to read LCD monitors around the corner. In: IEEE Symp. on Security and Privacy (2008)
19. Backes, M., Chen, T., Duermuth, M., Lensch, H.P.A., Welk, M.: Tempest in a teapot: Compromising reflections revisited. In: IEEE Symp. on Security and Privacy (2009)
20. LOREO (Lens in cap), http://www.loreo.com/pages/products/loreo_lenscap_spec.html
21. Kim, S.J., Koh, K., Lustig, M., Boyd, S.: An efficient method for compressed sensing. In: ICIP (2007)
22. Johnson, M.K., Adelson, E.H.: Retrographic sensing for the measurement of surface texture and shape. In: CVPR, pp. 1070–1077 (2009)
23. Perlin, K.: An image synthesizer. SIGGRAPH Comput. Graph. 19, 287–296 (1985)

A Continuous Max-Flow Approach to Potts Model

Jing Yuan[1], Egil Bae[2], Xue-Cheng Tai[2,3], and Yuri Boykov[1]

[1] Computer Science Department, University of Western Ontario, London Ontario,
Canada N6A 5B7
{cn.yuanjing,yboykov}@gmail.com
[2] Department of Mathematics, University of Bergen, Norway
{Egil.Bae,tai}@math.uib.no
[3] Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
tai@mi.uib.no

Abstract. We address the continuous problem of assigning multiple (unordered) labels with the minimum perimeter. The corresponding discrete Potts model is typically addressed with a-expansion which can generate metrication artifacts. Existing convex continuous formulations of the Potts model use TV-based functionals directly encoding perimeter costs. Such formulations are analogous to 'min-cut' problems on graphs. We propose a novel convex formulation with a continous 'max-flow' functional. This approach is dual to the standard TV-based formulations of the Potts model. Our continous max-flow approach has significant numerical advantages; it avoids extra computational load in enforcing the simplex constraints and naturally allows parallel computations over different labels. Numerical experiments show competitive performance in terms of quality and significantly reduced number of iterations compared to the previous state of the art convex methods for the continuous Potts model.

1 Introduction

The multi-partitioning problem, or multi-labeling problem, was extensively investigated in image processing and computer vision [1]. It computes the optimal labeling $l \in l_1, ..., l_n$ of each graph node or image pixel. Looking for such optimal labeling function with respect to some energy functional is an important mathematical strategy to model a wide range of applications, e.g. image segmentation [2,3], 3D reconstruction [4] etc. In this work, we focus on the Potts model that does not favor any particular order of the labels. The Potts model is also referred to as a *piecewise constant labeling* model which minimizes the total perimeter of the one-label (constant) regions.

In a discrete setting, Potts model corresponds to a practically important special case of a Markov Random Field (MRF) defined over a graph [5]. A typical MRF energy sums unary potentials defined over graph nodes and pairwise potentials defined over graph edges. When pixels can take only one of 2 labels, the

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 379–392, 2010.

resulting binary energy function can be efficiently and globally minimized by graph cuts [6], provided that the pairwise potentials are submodular [7]. However, for more than two labels typical MRF optimization problems are NP hard, so is Potts model. In particular, Potts model corresponds to a multi-terminal graph cut problem where only provably good approximate solutions are guaranteed, for example, via α-expansion or $\alpha - \beta$ swap [2] and some LP relaxations [8,9]. Another drawback of the discrete setting is that the results are often biased by the discrete grid causing metrication errors. Such visual artifacts can be largely reduced by either adding more neighbour nodes [10,11] or applying high-order cliques [12]. However, extra computation and memory load are introduced.

Parallel to these developments, variational methods have been proposed for solving the same Potts model in the spatially continuous setting where a bounded image domain is considered. In this regard, level set introduces the most direct and natural way to encode the piecewise constant labeling function and its related computation provides an efficient way to resolve the optimal partitions with a subgrid accuracy, see e.g. [13,14,15] and its variant of the piecewise constant level set method (PCLSM) [16,17]. Unfortunately, these formulations are nonconvex and computation often gets stuck in a local minima. Recently, convex relaxation approaches were proposed, e.g. [3,18,19,20,21,22]. Compared to level set methods, great advantages in numerics can be achieved, e.g. reliable algorithms can be build up by standard convex optimization theories [23]. Since a strict mathematical proof of the exactness of such convex relaxation approaches to the nonconvex Potts model is still open and argued, its approximation result can only be accepted as suboptimal. One may claim the convex relaxation method gives the solution which is closer to the exact global minimum than the local minima by the level set formulation. Our experiment results confirmed this.

In this paper, we study and solve the Potts problem in the spatially continuous setting through its convex relaxed formulation, i.e. the convex relaxed Potts model. In [18,22], such convex minimization problem is computed directly through the minimization over the labeling functions, i.e. tackle the minimal cut problem in a direct way, extra computation load is introduced to explore the pointwise simplex constraint within each iteration. Bae et. al. [21] proposed an equivalent dual model and its associated smoothing formulation based on the maximum entropy regularization, which properly avoids the extra step to handle simplex constraints and leads to a much simpler numerical scheme. To the best of our knowledge, none of previous works investigates the potential max-flow formulation which is dual to the concerning minimal cut. This is in contrast to the discrete case, where the minimal cut of a graph is often studied and computed over its dual maximal flow formulation, most efficient algorithms of graph-cuts were designed and explained in a flow maximization manner [24]. We devote this work to study the max-flow model associated to the convex relaxed Potts model. We also propose a fast max-flow based algorithm for computing continuous min-cuts. Experiments show that our max-flow algorithm is much more efficient than the state of art of computational methods [18,22].

Contributions. We summarize our main contributions in this paper as follows: first, we propose the novel max-flow formulation to the minimal cut of the given continuous image domain, i.e. the convex relaxed Potts problem. We show the studied max-flow and min-cut models are equivalent and dual to each other, hence the convex relaxed Potts problem can be solved through the proposed max-flow formulation. Analysis of the max-flow problem also leads to a new variational perspective of the corresponding minimal cut or continuous Potts problem. In addition, we build up the new multiplier-based max-flow algorithm upon the equivalent primal-dual model. It is numerically reliable and efficient. Its convergence can be proved by classical optimization theories. Our experiments show it is around 4 times faster than the previous methods [18,22]. Last but not least, such algorithm has a natural parallel framework over labeling functions and can, therefore, be easily implemented and accelerated on a parallel platform.

2 Convex Relaxed Potts Model and Previous Works

2.1 Convex Relaxed Potts Model

The Potts model originates from the statistical physics [25] and its spatially continuous version tries to partition the continuous domain Ω into n disjoint subdomains $\{\Omega_i\}_{i=1}^n$ by

$$\min_{\{\Omega_i\}_{i=1}^n} \sum_{i=1}^n \int_{\Omega_i} \rho(l_i, x)\, dx + \lambda \sum_{i=1}^n |\partial \Omega_i| \tag{1}$$

$$\text{s.t.} \quad \cup_{i=1}^n \Omega_i \, = \, \Omega \,, \quad \Omega_k \cap \Omega_l \, = \, \emptyset \,, \forall k \neq l \tag{2}$$

where $|\partial \Omega_i|$ measures the perimeter of each disjoint subdomain Ω_i, $i = 1 \ldots n$. The function $\rho(l_i, x), i = 1 \ldots n$, evaluates the performance of assigning the label l_i to the specified position x. As a special case, the piecewise constant Mumford-Shah functional can be encoded in terms of (1) with $\rho(l_i, x) = |I(x) - l_i|^p$ where $l_1 \ldots l_n$ are the given grayvalue constants. Obviously, Potts model favors the labeling with 'tight' boundaries.

Let $u_i(x), i = 1 \ldots n$, denote the indicator function of the disjoint subdomain Ω_i, i.e.

$$u_i(x) \, := \begin{cases} 1 \,, & x \in \Omega_i \\ 0 \,, & x \notin \Omega_i \end{cases} \,, \quad i = 1 \ldots n \,.$$

The perimeter of each disjoint subdomain can be computed by

$$|\partial \Omega_i| \, = \int_\Omega |\nabla u_i| \, dx \,, \quad i = 1 \ldots n \,. \tag{3}$$

The Potts model (1) can then be rewritten as

$$\min_{u_i(x) \in \{0,1\}} \sum_{i=1}^n \int_\Omega \left\{ u_i(x) \rho(l_i, x) \, + \, \lambda \, |\nabla u_i| \, \right\} dx \,, \ \text{ s.t. } \sum_{i=1}^n u_i(x) \, = \, 1 \,, \, \forall x \in \Omega \tag{4}$$

where the constraints to $u_i(x)$, $i = 1 \ldots n$, just corresponds to the condition (2) of subdomains Ω_i, $i = 1 \ldots n$.

Clearly, the Potts model (4) is nonconvex due to the binary configuration of each function $u_i(x) \in \{0, 1\}$. The *convex relaxed Potts model* [20,22,21] proposes to relax such binary constraints to the convex interval $[0, 1]$ and approximates (4) by the reduced convex optimization problem:

$$\min_{u \in S} \sum_{i=1}^{n} \int_{\Omega} u_i(x)\, \rho(l_i, x)\, dx \;+\; \alpha \sum_{i=1}^{n} \int_{\Omega} |\nabla u_i|\; dx \qquad (5)$$

where S is the convex constrained set of $u(x) := (u_1(x), \ldots, u_n(x))$:

$$S \;=\; \{u(x) \,|\, (u_1(x), \ldots, u_n(x)) \;\in\; \triangle_+ \,,\; \forall x \in \Omega\,\}\,,$$

$\triangle_+$ is the simplex set, i.e.

$$\text{for } \forall x \in \Omega\,, \quad \sum_{i=1}^{n} u_i(x) \;=\; 1\,; \quad u_i(x) \in [0, 1]\,, \quad i = 1 \ldots n\,.$$

The computation result of the convex relaxed Potts model (5) gives rise to a cut of the continuous image domain Ω with multiple terminals. (5) is, therefore, also called the *continuous min-cut model* in this paper. This is in comparison to its equivalent max-flow formulation proposed in later sections.

2.2 Previous Works

In [18], Zach et al introduced an alternating optimization approach to solve (5) in a numerically splitting way:

$$\min_{u, v \in S} \sum_{i=1}^{n} \int_{\Omega} v_i(x)\, \rho(l_i, x)\, dx \;+\; \frac{1}{2\theta} \left\| u - v \right\|^2 \;+\; \alpha \sum_{i=1}^{n} \int_{\Omega} |\nabla u_i|\; dx\,.$$

Obviously, when θ takes a value small enough, the above convex optimization problem properly approximates the convex relaxed Potts model (5). Within each iteration, two substeps are taken to tackle the total-variation term and explore the pointwise simplex constraint S respectively.

In [22], a Douglas-Rachford splitting algorithm was proposed to solve a quite similar problem as (5), where a variant of the total-variation term is considered:

$$\int_{\Omega} \sqrt{\left|\nabla u_1(x)\right|^2 + \ldots + \left|\nabla u_n(x)\right|^2}\, dx\,.$$

As in [18], the proposed splitting procedure involves an outer loop with two substeps, where the first substep solves a tv minimization problem iteratively until convergence, while the second substep projects the current solution to the convex set S. In [26] Nestorovs algorithm was applied to the problem, however this algorithm does not solve the problem exactly, only within a suboptimality bound.

In [20,27], the authors introduced another relaxation based on a multi-layered configuration, which was shown to be tighter. A more complex constraint on the dual variable p is given to avoid multiple countings. In addition, a PDE-based projection-descent scheme was applied to achieve the minimum.

In contrast to [18,22,20,27], [21] did not try to tackle the labeling function of the continuous min-cut problem (5) directly, but solved its equivalent dual formulation:

$$\max_{p_i \in C_\alpha} \int_\Omega \left\{ \min \left(\rho(l_1, x) + \operatorname{div} p_1 \; \ldots \; \rho(l_n, x) + \operatorname{div} p_n \right) \right\} dx \, . \tag{6}$$

where $\operatorname{div} p_i$, $i = 1 \ldots n$, correspond to the total-variation terms under the dual perspective and the convex set C_α is defined as

$$C_\alpha \; = \; \{ p \,|\, \|p\|_\infty \leq \alpha \, , \; p_n|_{\partial\Omega} = 0 \} \, . \tag{7}$$

Once the optimal functions $p_i^*(x)$, $i = 1 \ldots n$, were resolved, the labeling functions $u_i(x)$, $i = 1 \ldots n$, can be simply recovered by

$$u_k^*(x) \; = \; \begin{cases} 1 & \text{if } k = \arg\min_{i=1\ldots n} \; \rho(l_i, x) + \operatorname{div} p_i^*(x) \\ 0 & \text{otherwise} \end{cases} \quad . \tag{8}$$

provided the above argmin is unique. It was further shown by [21] that the nonsmooth dual formulation (6) can be properly approximated by the maximization of a smooth energy function, i.e.

$$\max_{p_i \in C_\lambda} \; -s \int_\Omega \left\{ \log \sum_{i=1}^{n} \exp\left(\frac{-f_i - \operatorname{div} p_i}{s} \right) \right\} dx \, . \tag{9}$$

Such a smooth dual model (9) approaches (6) with a maximum entropy regularizer and can be solved efficiently by a simple and reliable algorithmic scheme due to its smoothness and convexity.

In this paper, we propose a new continuous max-flow formulation which is equivalent to the continuous min-cut model (5), actually dual to each other. In theory, it provides a new variational perspective to investigate the continuous min-cut with multiple terminals or labels. In numerics, its great advantages over previous works are: it avoids pointwise projections onto the simplex constraint S within each outer loop as [18,22]; in comparison to [21,26], it exactly solves (6) without any smoothing procedure; it is globally optimized based on an efficient and reliable multiplier-based max-flow algorithm, in contrast to the PDE-descent method [20,27] whose convergence may suffer from uncareful stepsizes resulting in suboptimums; experiments show a faster convergence rate, about 4 times, than [18,22].

3 Continuous Max-Flow Model

In this section, we introduce the novel continuous max-flow formulation to the continuous min-cut problem (5) with n labels.

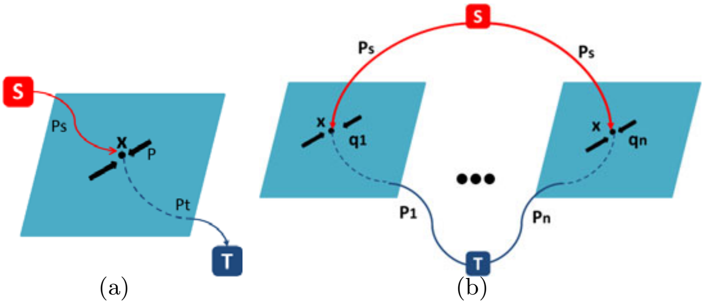


Fig. 1. (a) Continuous settings of max-flow with two labels; (b) Continuous configuration of max-flow with n labels

3.1 Continuous Max-Flow Model

Continuous Max-Flow Model with 2 Labels. Before we introduce the continuous max-flow model with n labels, we first introduce the recent study of the continuous max-flow model with 2 labels proposed by the authors [28] which is dual to the continuous s-t cut. This is directly analoguous to the graph-based max-flow and s-t cut: given the continuous image domain Ω, we assume there are two terminals, the source s and the sink t, see figure (a) of Fig. 1. We assume that for each image position $x \in \Omega$, there are three concerning flows: the source flow $p_s(x) \in \mathbb{R}$ directed from the source s to x, the sink flow $p_t(x) \in \mathbb{R}$ directed from x to the sink t and the spatial flow field $p(x) \in \mathbb{R}^2$. The three flow fields are constrained by capacities

$$p_s(x) \le C_s(x) , \quad p_t(x) \le C_t(x) , \quad |p(x)| \le C(x) ; \quad \forall x \in \Omega . \tag{10}$$

In addition, for $\forall x \in \Omega$, all flows are conserved, i.e.

$$p_t - p_s + \operatorname{div} p = 0 , \quad \forall x \in \Omega . \tag{11}$$

Therefore, we formulate the corresponding max-flow problem by maximizing the total flow from the source:

$$\max_{p_s, p_t, p} \int_\Omega p_s \, dx \tag{12}$$

subject to flow constraints (10) and (11).

Yuan et al [28] proved that such a continuous max-flow formulation (12) is equivalent to the continuous s-t min-cut problem [3,29] as follows:

$$\min_{u(x) \in [0,1]} \int_\Omega (1-u) C_s \, dx + \int_\Omega u C_t \, dx + \int_\Omega C(x) \, |\nabla u| \, dx . \tag{13}$$

Actually, (13) just gives the dual model to (12) and the labeling function $u(x)$ is the multiplier to the flow conservation condition (11). Furthermore, an efficient and reliable max-flow based algorithm can be built up through (12).

Continuous Max-Flow Model with n Labels. Motivated by the above observations, we give a continuous configuration of the max-flow model with n labels, see figure (b) of Fig. 1:

1. n copies Ω_i, $i = 1 \ldots n$, of the image domain Ω are given in parallel;
2. For each position $x \in \Omega$, the source flow $p_s(x)$ tries to stream from the source s to x at each copy Ω_i, $i = 1 \ldots n$, of Ω. The source flow field is the same for each Ω_i, $i = 1 \ldots n$, i.e. $p_s(x)$ is unique;
3. For each position $x \in \Omega$, the sink flow $p_i(x)$, $i = 1 \ldots n$, is directed from x at the i-th copy Ω_i to the sink t. The n sink flow fields $p_i(x)$, $i = 1 \ldots n$, may be different;
4. The spatial flow fields $q_i(x)$, $i = 1 \ldots n$, are defined within each copy Ω_i, $i = 1 \ldots n$. They may also be different from each other.

For such a contiuous setting, we give the constrained conditions for flows $p_i(x)$ and $q_i(x)$, at $x \in \Omega$, as follows

$$|q_i(x)| \leq C_i(x) , \quad p_i(x) \leq \rho(\ell_i, x) , \quad i = 1 \ldots n ; \tag{14}$$

$$\big(\operatorname{div} q_i - p_s + p_i\big)(x) = 0 , \quad i = 1, \ldots, n . \tag{15}$$

Note: there is no constraint for the source flow $p_s(x)$.

We, then, formulate the respective continuous max-flow model, over all the flow fields $p_s(x)$, $p(x) := (p_1(x), \ldots, p_n(x))$ and $q(x) := (q_1(x), \ldots, q_n(x))$, as

$$\max_{p_s, p, q} \; \big\{ P(p_s, p, q) := \int_\Omega p_s \, dx \big\} \tag{16}$$

subject to (14) and (15).

In the following section, we introduce the equivalent models of the continuous max-flow formulation (16). We show its equivalent dual model just gives the continuous min-cut model (5) provided $C(x) = \alpha$.

Comments. It is easy to notice that when the source flow $p_s(x)$ tries to pass the same position x at each Ω_i, $i = 1 \ldots n$, in view of the flow conservation condition (15), we have

$$p_s(x) = \operatorname{div} q_i(x) + p_i(x) , \quad i = 1 \ldots n .$$

Observe the righthand of the above formulation and the configuration shown in Fig. 1, $p_s(x)$ is constrained and should be given within a feasible set, i.e. consistent to all n flow configurations of $\operatorname{div} q_i(x) + p_i(x)$, $i = 1 \ldots n$, at x. Consider the flow capacity constraint of $p_i(x)$ (14), it is easy to conclude that

$$p_s(x) = \min(\operatorname{div} q_1(x) + \rho(l_1, x), \ldots, \operatorname{div} q_n(x) + \rho(l_n, x)) , \quad \forall x \in \Omega . \tag{17}$$

Therefore, the maximum of $\int_\Omega p_s \, dx$ suggests

$$\max_{|q_i(x)| \leq C_i(x)} \int_\Omega \big\{ \min(\rho(l_1, x) + \operatorname{div} q_1, \ldots, \rho(l_n, x) + \operatorname{div} q_n) \big\} \, dx , \tag{18}$$

which discovers the dual model (6) of [21] when $C_i(x) = \alpha$ are constant.

We can consider each image copy Ω_i, $i = 1 \dots n$, together with the constrained sink flow field $p_i(x)$ and the spatial flow field $q_i(x)$ given in (14), as a 'filter' F_i whose capacity at $x \in \Omega$ is constrained by $\operatorname{div} q_i(x) + p_i(x)$. Then one can explain the max-flow model (16) such that all the filters F_i, $i = 1, \dots, n$, are layered one by one and the source flow $p_s(x)$ tries to pass such a stack of 'filters' in one time. It is obvious that $p_s(x)$ is bottlenecked by the minimum capacity of $\operatorname{div} q_i(x) + p_i(x)$, $i = 1 \dots n$. In such a filter configuration, (16) aims to maximize the total flow passing this 'filter' set.

3.2 Equivalent Primal-Dual Formulation

We introduce the multiplier functions $u_i(x)$, $i = 1 \dots n$, to the flow balance condition (15). Therefore, we have the equivalent primal-dual model of (16)

$$\max_{p_s, p, q} \min_{u} \big\{ E(p_s, p, q; u) := \int_\Omega p_s \, dx + \sum_{i=1}^n \int_\Omega u_i (\operatorname{div} q_i - p_s + p_i) \, dx \big\} \tag{19}$$

$$\text{s.t.} \quad p_i(x) \le \rho(\ell_i, x) \,, \quad |q_i(x)| \le C_i(x) \,; \quad i = 1 \dots n$$

where $u(x) := (u_1(x), \dots, u_n(x))$.

Rearranging the energy function $E(p_s, p, q; u)$ of (19), we have

$$E(p_s, p, q; u) = \int_\Omega \big\{ (1 - \sum_{i=1}^n u_i) \, p_s + \sum_{i=1}^n u_i \, p_i + \sum_{i=1}^n u_i \operatorname{div} q_i \big\} \, dx \tag{20}$$

For the primal-dual model (19), the conditions of the minimax theorem (see e.g., [30] Chapter 6, Proposition 2.4) are all satisfied. That is, the constraints of flows are convex, and the energy function is linear in both the multiplier u and the flow functions p_s, p and q, hence convex l.s.c. for fixed u and concave u.s.c. for fixed p_s, p and q. This confirms the existence of at least one saddle point, see [30,31]. It also follows that the min and max operators of the primal-dual model (19) can be interchanged, i.e.

$$\max_{p_s, p, q} \big\{ \min_{u} E(p_s, p, q; u) \big\} = \min_{u} \big\{ \max_{p_s, p, q} E(p_s, p, q; u) \big\} \,. \tag{21}$$

3.3 Equivalent Dual Formulation

Now we investigate the optimization of (19) by the min-max order as the right-hand side of (21), i.e. first maximize $E(p_s, p, q; u)$ over the flow functions p_s, p and q then minimize over the multiplier function u. We show that this leads to the equivalent dual model of the continuous max-flow formulation (16), i.e.

$$\min_{u} \big\{ D(u) := \sum_{i=1}^n \big(\int_\Omega u_i(x) \, \rho(\ell_i, x) \, dx + \int_\Omega C_i(x) \, |\nabla u_i| \, dx \big) \big\} \tag{22}$$

$$\text{s.t.} \quad \sum_{i=1}^n u_i(x) = 1 \,, \quad u_i(x) \ge 0 \,.$$

Optimization of Flow Functions p, q and p_s. In order to optimize the flow function $p(x)$ in (20), let us consider the following maximization problem

$$f(q) \,=\, \max_{p \leq C} \, p \cdot q \,. \tag{23}$$

where p, q and C are scalars. When $q < 0$, p can be chosen to be a negative infinity value in order to maximize the value $p \cdot q$, i.e. $f(q) = +\infty$. In consequence, we must have $q \geq 0$ so as to make the function $f(q)$ meaningful. Observe now that

$$\begin{cases} \text{if } q = 0\,, \text{ then } p \leq C \text{ and } f(q) \text{ reaches the maximum } 0 \\ \text{if } q > 0\,, \text{ then } p = C \text{ and } f(q) \text{ reaches the maximum } q \cdot C \end{cases} . \tag{24}$$

By virtue of (24), we can equally express $f(q)$ by

$$f(q) \,=\, q \cdot C\,, \quad q \geq 0\,. \tag{25}$$

Apply (23) to the maximization of $E(p_s, p, q; u)$ of (20) over the sink flows $p_i(x)$, $i = 1 \ldots n$, we have

$$\max_{p_i(x) \leq \rho(l_i, x)} \int_\Omega u_i p_i \, dx \,=\, \int_\Omega u_i(x) \rho(l_i, x) \, dx\,, \quad u_i(x) \,\geq\, 0\,,\; i = 1, \ldots, n\,. \tag{26}$$

For the maximization over the spatial flow functions $q_i(x)$, $i = 1, \ldots, n$, it is well-known [32] that

$$\max_{|q_i(x)| \leq C_i(x)} \int_\Omega u_i \operatorname{div} q_i \, dx \,=\, \int_\Omega C_i(x) \left|\nabla u_i\right| \, dx\,. \tag{27}$$

Furthermore, observe the source flow function $p_s(x)$ is unconstrained, the maximization of (20) over p_s simply leads to

$$1 - \sum_{i=1}^n u_i(x) \,=\, 0\,, \quad \forall x \in \Omega\,. \tag{28}$$

By the results of (28), (26) and (27), it is easy to conclude that the maximization of the primal-dual model (20) over flow functions p_s, p and q gives its equivalent dual model (22), hence we have

Proposition 1. *The continuous max-flow model* (16)*, the primal-dual model* (19) *and the dual model* (22) *are equivalent to each other.*

In this work, we focus on the case when $C_i(x) = \alpha$, $\forall x \in \Omega$ and $i = 1, \ldots, n$. Obviously, we have

Proposition 2. *When $C_i(x) = \alpha$, $\forall x \in \Omega$ and $i = 1 \ldots n$, the dual model* (22) *equals the continuous min-cut model* (5)*.*

3.4 Variational Perspective of Flows and Cuts

Through the above analytical results, we can also give a variational perspective of flows and cuts, which recovers conceptions and terminologies used in the graph setting.

Consider the maximization problem (23), for any fixed q, let some optimal p^* maximize $q \cdot p$ over $p \leq C$. By means of variations, if such $p^* < C$ strictly, its variation directly leads to $q = 0$ since the variation δp can be both negative and positive. On the other hand, for $p^* = C$, its variation under the constraint $p \leq C$ gives $\delta p < 0$, then we must have $q > 0$. In terms of graph-cut, $p^* < C$ means p does not reach its maximum C, i.e. 'unsaturated'; then it leads to $q = 0$ which means the so-called 'cut'.

In the same manner, for the maximization of $p_i(x)$, $i = 1 \ldots n$, it is easy to see that when the flow $p_i(x) < \rho(l_i, x)$ at $x \in \Omega$, i.e. 'unsaturated', we must have $u_i(x) = 0$, i.e. $u_i(x)p_i(x) = 0$, which means that at the position x, the flow $p_i(x)$ has no contribution to the energy function and the flow $p_i(x)$, from $x \in \Omega_i$ to the sink t, can be 'cut' off from the energy function of (19). On the other hand, in view of (8), the indicator function $u_i(x) = 0$ definitely means the position x is not labeled as l_i.

4 Multiplier-Based Max-Flow Algorithm

Observe that the energy function of the primal-dual model (19) just gives the Lagrangian function of (16) where $u_i(x)$, $i = 1 \ldots n$, are the corresponding multiplier functions. We introduce our multiplier-based max-flow algorithm, which is based on the augmented lagrangian method [23]. We define the augmented Lagrangian function

$$L_c(p_s, p, q, u) = \int_\Omega p_s \, dx + \sum_{i=1}^n \langle u_i, \operatorname{div} q_i - p_s + p_i \rangle - \frac{c}{2} \sum_{i=1}^n \|\operatorname{div} q_i - p_s + p_i\|^2$$

where $c > 0$. Each iteration of the algorithm can then be generalized as follows:

- Optimize spatial flows q_i, $i = 1 \ldots n$, by fixing other variables:

$$q_i^{k+1} := \arg \max_{\|q_i\|_\infty \leq \alpha} -\frac{c}{2} \left\|\operatorname{div} q_i + p_i^k - p_s^k - u_i^k/c\right\|^2 , \tag{29}$$

  which can be solved by Chambolle's projection algorithm [33].
- Optimize sink flows p_i, $i = 1...n$, by fixing other variables

$$p_i^{k+1} := \arg \max_{p_i(x) \leq \rho(\ell_i, x)} -\frac{c}{2} \left\|p_i + \operatorname{div} q_i^{k+1} - p_s^k - u_i^k/c\right\|^2 , \tag{30}$$

  which can be computed at each $x \in \Omega$ in a closed form.

– Optimize the source flow p_s and update multipliers u_i, $i = 1 \ldots n$

$$p_s^{k+1} := \arg\max_{p_s} \int_{\Omega} p_s \, dx - \frac{c}{2} \sum_{i=1}^{n} \left\| p_s - (p_i^{k+1} + \operatorname{div} q_i^{k+1}) + u_i^k / c \right\|^2 \,, \quad (31)$$

$$u_i^{k+1} = u_i^k - c \, (\operatorname{div} q_i^{k+1} - p_s^{k+1} + p_i^{k+1}) \,. \quad (32)$$

Both can be obtained in a closed form.

Consider the above numerical steps, it is easy to see that the two flows q_i and p_i, $i = 1 \ldots n$, computed by (29) and (30) can be handled independently for each label i. Hence, (29) and (30) can be implemented in a parallel way. Once such two steps are finished, the source flow $p_s(x)$ and the labeling functions $u_i(x)$, $i = 1 \ldots n$, are updated. Obviously, such parallelism naturally originates the configuration shown in Fig. 1.

5 Experiments

In this section, we show some experiments to validate the proposed max-flow model and its resulted algorithm. The quality of the relaxation (5) has been evaluated extensively in [18,22,21] where it has been shown to be competitive to several state of the art methods from discrete optimization like alpha expansion and alpha beta-swap [2] for approximately minimizing the Pott's energy. In addition the variational model comes with the important advantage of rotational invariance, which means that metrication errors are avoided. We will therefore not elaborate too much on the quality of the solutions in this paper. Examples are given in Figure (2), where we have used the Mumford-Shah data term $\rho(\ell_i, x) = |I(x) - \ell_i|^2$, $i = 1, ..., n$. As we see, equally good solutions as alpha expansion are produced, but without the metrication artifacts.

In contrast to the minimization approach of Zach et. al. [18], the proposed algorithm can be proved to converge by classical optimization theories. The Douglas-Rachford splitting approach given in [22] can also be proved to converge (in the discrete setting), but we experienced that our approach was more efficient than both these approaches. The inner problem has the same complexity for all approaches, since it is dominated by the process of iteratively solve a tv minimization problem. However, in contrast to [18,22] our approach avoids iterative projections to the convex set S and consequently require much less outer iterations. Convergence is reached for a wide range of the outer "step size" c. To measure converge, we find a good estimate of the final energy E^* by solving the problem with 10000 outer iterations. The energy precision at iteration k is then measured by $\epsilon = \frac{E^k - E^*}{E^*}$. For the three images (see Fig. 2), different precision ϵ are taken and the total number of iterations to reach convergence is evaluated, see Tab 1: clearly, our method is about 4 times faster than the Douglas-Rachford-splitting [22], the approach in [18] is even slower and failed to reach such a low precision.

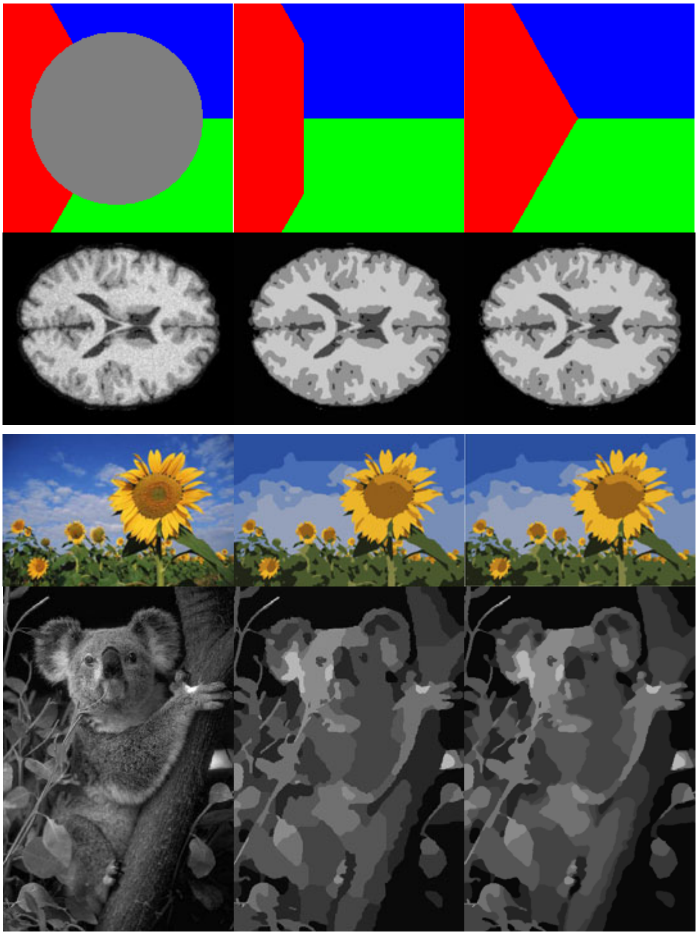


Fig. 2. Each row (from left to right): the input image, result by Alpha expansion with 8 neighbors, result by the proposed max-flow approach. For the experiment in 1st row (inpainting in gray area), $\alpha = 0.03$ and $n = 3$; 2nd row, $\alpha = 0.04$ and $n = 4$, 3rd row, $\alpha = 0.047$ and $n = 10$; 4th row, $\alpha = 0.02$ and $n = 8$.

Table 1. Comparisons between algorithms: Zach et al [18], Lellmann [22] and the proposed max-flow algorithm: for the three images (see Fig. 2), different precision ϵ are taken and the total number of iterations to reach convergence is evaluated

	Brain $\epsilon \leq 10^{-5}$	Flower $\epsilon \leq 10^{-4}$	Bear $\epsilon \leq 10^{-4}$
Zach et al [18]	fail to reach such a precision		
Lellmann et al [22]	421 iter.	580 iter.	535 iter.
Proposed algorithm	88 **iter.**	147 **iter.**	133 **iter.**

6 Conclusions

In this paper, we introduce and study the novel continuous max-flow model which is dual to the continuous min-cut problem, i.e. the convex relaxed Potts model. We also propose a variational perspective of flows and cuts in the continuous configuration, which recovers and well explains connections of flows and cuts. Moreover, in comparison to previous efforts which are trying to compute the optimal labeling functions in a direct way, we propose the new multiplier-based max-flow algorithm. Main advantages of such max-flow algorithm are: it avoids extra computation load to explore the simplex constraint, each flow is adjusted in a simple way and its numerical scheme contains a natural parallel framework, which can be easily accelarated. Numerical experiments show it outperforms state of art approaches in terms of quality and efficiency.

Acknowledgements. This research has been supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Accelerator Grant R3584A04, the Norwegian Research Council (eVita project 166075), MOE (Ministry of Education) Tier II project T207N2202) and IDM project NRF2007IDMIDM002-010.

References

1. Paragios, N., Chen, Y., Faugeras, O.: Handbook of Mathematical Models in Computer Vision. Springer, New York (2005)
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on PAMI 23, 1222–1239 (2001)
3. Nikolova, M., Esedoglu, S., Chan, T.F.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM J. App. Math. 66, 1632–1648 (2006)
4. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 82–96. Springer, Heidelberg (2002)
5. Li, S.Z.: Markov random field modeling in image analysis. Springer, New York (2001)
6. Greig, D.M., Porteous, B.T., Seheult, A.H.: Exact maximum a posteriori estimation for binary images. J. Royal Stat. Soc., Series B, 271–279 (1989)
7. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. IEEE Transactions on PAMI 26, 65–81 (2004)
8. Komodakis, N., Tziritas, G.: Approximate labeling via graph-cuts based on linear programming. In: Pattern Analysis and Machine Intelligence, pp. 1436–1453 (2007)
9. Wainwright, M., Jaakkola, T., Willsky, A.: Map estimation via agreement on (hyper)trees: Message-passing and linear programming approaches. IEEE Transactions on Information Theory 51, 3697–3717 (2002)
10. Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: ICCV 2003, pp. 26–33 (2003)
11. Kolmogorov, V., Boykov, Y.: What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In: ICCV, pp. 564–571 (2005)

12. Kohli, P., Kumar, M.P., Torr, P.H.: p^3 and beyond: Move making algorithms for solving higher order functions. IEEE Transactions on PAMI 31, 1645–1656 (2009)
13. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: algorithms based on hamilton-jacobi formulations. J. Comput. Phys. 79, 12–49 (1988)
14. Chan, T., Vese, L.: Active contours without edges. IEEE Image Proc. 10, 266–277 (2001)
15. Vese, L.A., Chan, T.F.: A new multiphase level set framework for image segmentation via the mumford and shah model. IJCV 50, 271–293 (2002)
16. Lie, J., Lysaker, M., Tai, X.: A binary level set model and some applications to Mumford-Shah image segmentation. IEEE Img. Proc. 15, 1171–1181 (2006)
17. Lie, J., Lysaker, M., Tai, X.C.: A variant of the level set method and applications to image segmentation. Math. Comp. 75, 1155–1174 (2006)
18. Zach, C., Gallup, D., Frahm, J.M., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: VMV 2008 (2008)
19. Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of continuous multi-label problems. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 792–805. Springer, Heidelberg (2008)
20. Chambolle, A., Cremers, D., Pock, T.: A convex approach for computing minimal partitions. Technical Report TR-2008-05, University of Bonn (2008)
21. Bae, E., Yuan, J., Tai, X.: Global minimization for continuous multiphase partitioning problems using a dual approach. UCLA CAM Report [09-75] (2009)
22. Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C.: Convex multi-class image labeling by simplex-constrained total variation. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 150–162. Springer, Heidelberg (2009)
23. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific (1999)
24. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)
25. Potts., R.B.: Some generalized order-disorder transformations. Proceedings of the Cambridge Philosophical Society 48, 106–109 (1952)
26. Lellmann, J., Becker, F., Schnörr, C.: Convex optimization for multi-class image labeling with a novel family of total variation based regularizers. In: IEEE International Conference on Computer Vision (ICCV), pp. 646–653 (2009)
27. Pock, T., Chambolle, A., Bischof, H., Cremers, D.: A convex relaxation approach for computing minimal partitions. In: CVPR, Miami, Florida (2009)
28. Yuan, J., Bae, E., Tai, X.: A study on continuous max-flow and min-cut approaches. In: CVPR, USA, San Francisco (2010)
29. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., Osher, S.: Fast global minimization of the active contour/snake model. Journal of Mathematical Imaging and Vision 28, 151–167 (2007)
30. Ekeland, I., Téman, R.: Convex analysis and variational problems. Society for Industrial and Applied Mathematics, Philadelphia (1999)
31. Fan, K.: Minimax theorems. Proc. Nat. Acad. Sci. U. S. A. 39, 42–47 (1953)
32. Giusti, E.: Minimal surfaces and functions of bounded variation. Australian National University, Canberra (1977)
33. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision 20, 89–97 (2004)

Hybrid Compressive Sampling via a New Total Variation TVL1[*]

Xianbiao Shu and Narendra Ahuja

University of Illinois at Urbana-Champaign, Urbana, IL61801, USA,
{xshu2,n-ahuja}@illinois.edu

Abstract. Compressive sampling (CS) is aimed at acquiring a signal or image from data which is deemed insufficient by Nyquist/Shannon sampling theorem. Its main idea is to recover a signal from limited measurements by exploring the prior knowledge that the signal is sparse or compressible in some domain. In this paper, we propose a CS approach using a new total-variation measure TVL1, or equivalently TV_{ℓ_1}, which enforces the sparsity and the directional continuity in the gradient domain. Our TV_{ℓ_1} based CS is characterized by the following attributes. First, by minimizing the ℓ_1-norm of partial gradients, it can achieve greater accuracy than the widely-used $\text{TV}_{\ell_1\ell_2}$ based CS. Second, it, named hybrid CS, combines low-resolution sampling (LRS) and random sampling (RS), which is motivated by our induction that these two sampling methods are complementary. Finally, our theoretical and experimental results demonstrate that our hybrid CS using TV_{ℓ_1} yields sharper and more accurate images.

1 Introduction

Digital images or signals are conventionally acquired by Nyquist/Shannon sampling. That requires, to incur no loss, the underlying analog signal must be sampled at Nyquist rate which is at least twice its highest analog frequency. The resulting raw digital data is too large to sense, transmit and store in many applications. One solution to this problem is the well-known image compression methodology, such as the JPEG2000 [20] compression standard, which represents a digital image by a smaller number of dominant components and relaxes the storage and transmission requirements. However, sensing a large image is still challenging.

Recently, compressive sensing [7] or particularly compressive sampling, has been introduced to address this problem more efficiently. CS exploits the redundancy present in the image at the time of sampling itself. Instead of sensing all the pixels that define the complete image, compressive sampling acquires a linear combination of randomly selected pixels and recovers the full image from these samples [16,17,3,22,8]. Instead of first sampling and then compressing, this imaging model avoids sampling of the redundant aspects of the data in the first place.

Compressive sampling assumes that an image, vectorized as $\mathbf{I}$ of size L, can be represented as $\mathbf{I} = \Psi\mathbf{u}$ in some space, where $\mathbf{u}$ has K non-zero elements (called K-sparsity). Instead of sensing $\mathbf{u}$ directly in the Ψ domain, it may be easier to efficiently

[*] The support of the National Science Foundation under grant IIS 08-12188 and the Office of Naval Research under grant N00014-09-1-0017 is gratefully acknowledged.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 393–404, 2010.

sample I in a different subspace defined by Φ. Then, sensing acquires a small number of projections of I onto this subspace such that $\mathbf{b} = \Phi\mathbf{I}$, where $\Phi \in \mathbb{C}^{M \times L}(K < M < L)$ is a sampling matrix. Given the measurements $\mathbf{b}$, CS recovers the K dominant components constituting $\mathbf{u}$. This translates into the problem of estimating the sparsest $\mathbf{u}$ satisfying the measurement vector $\mathbf{b}$:

$$\min_{\mathbf{u}} \|\mathbf{u}\|_0 \quad \text{s. t.} \quad A\mathbf{u} = \Phi\Psi\mathbf{u} = \mathbf{b} \tag{1}$$

However, ℓ_0-norm minimization is an NP-complete problem [15]. Fortunately, it has been proven that the intractable ℓ_0-problem is equivalent to the convex minimization of $\|\mathbf{u}\|_1$, if the sampling matrix $A = \Phi\Psi$ obeys uniform uncertainty principle (UUP), introduced in [2] and refined in [4]. According to the definition in [4], a measurement matrix $A \in \mathbb{R}^{M \times L}$ is said to obey UUP with an oversampling factor λ, if the inequality

$$\frac{1}{2} \cdot \frac{M}{L} \|f\|_2^2 \leq \|Af\|_2^2 \leq \frac{3}{2} \cdot \frac{M}{L} \|f\|_2^2 \tag{2}$$

holds for all $K-$sparse signals f, where $K \leq M/(\alpha\lambda)$ and $\alpha > 0$ is a sufficiently large constant. According to [2], random sampling matrix and Fourier sampling matrix both obey UUP with $\lambda = \log(L/K)$ and $\lambda = \log^6(L/K)$ respectively. They are capable of recovering $\mathbf{u}$ (with an overwhelming probability) from $\mathbf{b}$ of size $M \geq \alpha K \log(L/K)$ and $M \geq \alpha K \log^6(L/K)$ respectively.

In addition to the sparsity in the Ψ-transform domain (wavelets [3,17], curvelets [10] et al.), compressive sampling often uses Total variation (TV) [18] to exploit the sparsity in finite difference domain. In some applications [10,13,12,24], Ψ-transform sparsity and TV are enforced together to improve the recovery accuracy as follows:

$$\min_{\mathbf{u}} \text{TV}(\Psi\mathbf{u}) + \beta\|\mathbf{u}\|_1 \quad \text{s. t.} \quad \|\Phi\Psi\mathbf{u} - \mathbf{b}\|_2^2 \leq \sigma^2 \tag{3}$$

Where β trades TV with Ψ-transform sparsity and σ^2 is the noise variance.

In this paper, we concentrate on how to evaluate and improve TV based compressive sampling. The most widely-used form of TV in CS [16,17,3,13,24] including Single-Pixel Camera (SPC) [8] is $\text{TV}_{\ell_1\ell_2}$, which computes the summation of the magnitudes of gradients (SMG) across the image: $\text{TV}_{\ell_1\ell_2}(\mathbf{I}) = \sum_i \sqrt{(D_h\mathbf{I})_i^2 + (D_v\mathbf{I})_i^2}$ where D_h and D_v are horizontal and vertical gradient operators. This TV measure has the following shortcomings: (1) The field of gradient magnitudes is not as sparse as partial gradients fields; (2) $\text{TV}_{\ell_1\ell_2}$ is prone to causing blurring across sharp edges, since SMG prefers to suppress large partial gradients; (3) SMG is a nonlinear operator, which makes it difficult to minimize $\text{TV}_{\ell_1\ell_2}$ efficiently. To seek a more efficient decoding algorithm, [14] uses an invertible operator Ω, which we call TV'_{ℓ_1}, given by $\Omega\mathbf{I} = \|D_h\mathbf{I}\|_1 + \|D_v\mathbf{I}\|_1$. However, TV'_{ℓ_1} seeks the intensity continuity horizontally and vertically, but fails to enforce the intensity continuity diagonally. Thus, to overcome these shortcomings of $\text{TV}_{\ell_1\ell_2}$ and TV'_{ℓ_1}, a new TV measure is needed.

In CS, random sampling is generally assumed to be near-optimal in reducing the sampled data for unstructured images [7,2]. [16,17,3] combine low-frequency sampling and random sampling, on intuitive grounds alone, without formal justification. In this paper, we present a hybrid CS method using a new TV measure with the following two contributions:

1. We propose a new TV measure TV_{ℓ_1}, which recovers piecewise smooth images with all possible sharp edges by exploiting the sparsity and continuity in the gradient domain. In addition, the UUP condition shows our TV_{ℓ_1} achieves higher accuracy and requires fewer measurements for the same quality of reconstruction than previous $\text{TV}_{\ell_1\ell_2}$.
2. We present a theoretical analysis on hybrid sampling, which shows that low resolution sampling (LRS) and random sampling (RS) indeed complement each other for most natural images, and gives the criteria for the best combination of LRS and RS.

This paper is organized as follows. Section 2 describes our TV_{ℓ_1} based hybrid CS. Section 3 discusses implementation of our method. Section 4 presents experimental results. Section 5 gives concluding remarks.

2 Proposed TV Based Hybrid CS

Total variation $\text{TV}_{\ell_1\ell_2}$ is a widely-used measure for enforcing intensity continuity and recovering a piecewise smooth image in CS [16,17,3,13,24]. In this paper, we propose a new TV measure TV_{ℓ_1}, which exploits the continuity and sparsity in the partial gradient domain. In comparison with $\text{TV}_{\ell_1\ell_2}$, our TV_{ℓ_1} is able to recover sharper images with greater accuracy. Our TV_{ℓ_1} based CS problem can be formulated as follows.

$$\min_{\mathbf{I}} \text{TV}_{\ell_1}(\mathbf{I}) \quad \text{s. t.} \quad \Phi\mathbf{I} = \mathbf{b} \quad \text{and} \quad \Phi'\mathbf{I} = \mathbf{d} \tag{4}$$

where Φ is random sampling (RS) matrix or Fourier sampling matrix for large-scale images, and Φ' is low-resolution sampling (LRS) matrix, which acquires LR data d. To compare our TV_{ℓ_1} with $\text{TV}_{\ell_1\ell_2}$ directly, we do not combine our TV_{ℓ_1} with any Ψ-transform sparsity, even if their combination might improve the recovery accuracy.

2.1 A New TV Measure

In this section, we present a new TV measure TV_{ℓ_1}. For intensity continuity in Fig. 1(a), the pixel $I_{i,j}$ is desired to be of similar value to its four neighbors in smooth regions.

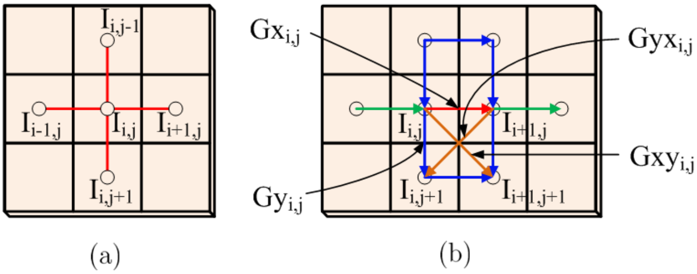


Fig. 1. (a) For intensity continuity, or gradient sparsity, we enforce each pixel, e.g. $I_{i,j}$, to be continuous with its 4 neighbors. (b) For gradient continuity, we enforce each partial gradient, e.g. $Gx_{i,j}$ marked as a red line, to be of similar value to its 6 neighbors marked as blue lines.

Similarly, partial gradients $Gx_{i,j} = I_{i+1,j} - I_{i,j}$ and $Gy_{i,j} = I_{i,j+1} - I_{i,j}$ can be continuous along all directions except their own directions, where they are desired to be discontinuous to obtain a sharp edge. Take $Gx_{i,j}$ (Fig. 1(b)) for example, our TV$_{\ell_1}$ will not enforce its continuity along the horizontal axis, but will do so along all other directions, as in Fig. 1(b)). For notational simplicity, we consider the continuity of partial gradients in a 2×2 neighborhood $(I_{i,j}, I_{i+1,j+1}, I_{i+1,j}, I_{i,j+1})$. The continuity constraints depend on the direction $\overrightarrow{D}$ associated with the edge, if it exists in the neighborhood. For different cases of $\overrightarrow{D}$, the continuity constraints are:

$$\begin{cases} \|Gxx_{i,j}\|_1 = \|Gx_{i,j} - Gx_{i,j+1}\|_1 = 0 \text{ if } \overrightarrow{D} \text{ is vertical.} \\ \|Gyy_{i,j}\|_1 = \|Gy_{i,j} - Gy_{i+1,j}\|_1 = 0 \text{ if } \overrightarrow{D} \text{ is horizontal.} \\ \|Gxy_{i,j}\|_1 = \|Gx_{i,j} + Gy_{i+1,j}\|_1 = 0 \text{ if } \overrightarrow{D} \text{ is left-lower.} \\ \|Gyx_{i,j}\|_1 = \|Gy_{i,j} - Gx_{i,j}\|_1 = 0 \quad \text{ if } \overrightarrow{D} \text{ is right-lower.} \end{cases}$$

Thus, we enforce the directional continuity of $\mathbf{Gx}$ and $\mathbf{Gy}$ by minimizing the ℓ_1-norm of $\mathbf{Gxy}, \mathbf{Gyx}, \mathbf{Gxx}$ and $\mathbf{Gyy}$. $Gxx_{i,j}$ is the derivative of $Gx_{i,j}$ along the vertical axis and $Gyy_{i,j}$ is the derivative of $Gy_{i,j}$ along the horizontal axis. Actually, $\|Gxx_{i,j}\|_1 = \|I_{i+1,j+1} + I_{i,j} - I_{i+1,j} - I_{i,j+1}\|_1 = \|Gyy_{i,j}\|_1$. By including the intensity continuity constraints in Fig. 1(a), we define our TV measure TV$_{\ell_1}$ as follows:

$$\text{TV}_{\ell_1}(\mathbf{I}) = \|\mathbf{Gx}\|_1 + \|\mathbf{Gy}\|_1 + \gamma(\|\mathbf{Gxy}\|_1 + \|\mathbf{Gyx}\|_1 + 2\|\mathbf{Gxx}\|_1) \qquad (5)$$

where γ trades the intensity continuity with the gradient continuity. $\mathbf{Gx}, \mathbf{Gy}, \mathbf{Gxy}$ and $\mathbf{Gyx}$ are respectively horizontal, vertical, and two diagonal partial gradients in Fig. 1(b). Given our goal is to recover the sparsest gradients, $\|Gxx_{i,j}\|_1 = \|Gyy_{i,j}\|_1 = 0$ implies zero partial gradients along one of four directions in the 2×2 neighborhood, or equivalently $Gx_{i,j} = 0$, $Gy_{i,j} = 0$, $Gxy_{i,j} = 0$ or $Gyx_{i,j} = 0$. In this case, minimizing $||\mathbf{Gxx}||_1$ is redundant under the condition of minimal $||\mathbf{Gx}||_1 + ||\mathbf{Gy}||_1 + ||\mathbf{Gxy}||_1 + ||\mathbf{Gyx}||_1$. Thus, our TV$_{\ell_1}$ can be simplified.

$$\text{TV}_{\ell_1}(I) = \|\mathbf{Gx}\|_1 + \|\mathbf{Gy}\|_1 + \gamma(\|\mathbf{Gxy}\|_1 + \|\mathbf{Gyx}\|_1) \qquad (6)$$

This simplified TV$_{\ell_1}$, enforces the sparsity and directional continuity in the gradient domain by seeking the γ-weighted sparsity of partial gradient fields $\mathbf{G} = [\mathbf{Gx}; \mathbf{Gy}; \mathbf{Gxy}; \mathbf{Gyx}]$.

In comparison with previous TV measures (TV$_{\ell_1\ell_2}$ and TV$'_{\ell_1}$), our TV$_{\ell_1}$ based CS can recover any piecewise smooth image with all possible sharp edges (horizontal, vertical or diagonal), where the tuning parameter γ plays a crucial role in determining its preference. In general, TV-based CS seeks the image that has the minimal TV value and is closest to the measurements. The widely-used measure TV$_{\ell_1\ell_2}$ minimizes the sum of magnitudes of gradients (SMG) and penalizes larger partial gradients. Thus TV$_{\ell_1\ell_2}$ is prone to recovering a blurred image (Fig. 2(b)). TV$'_{\ell_1}$, or equivalently a special case TV$_{\ell_1,\gamma=0}$, is prone to recovering an image of sharp horizontal and vertical edges in Fig. 2(c) by enforcing $\|\mathbf{Gx}\|_1 + \|\mathbf{Gy}\|_1$. However, these two images (Fig. 2(b)(c)) cause larger ℓ_1-norm of $\mathbf{Gyx}$. TV$_{\ell_1,\gamma=1}$ equally penalizes the ℓ_1-norm of each elements in the four partial gradient fields $\mathbf{G}$, whether large or small. So, TV$_{\ell_1,\gamma=1}$ is prone to recovering the sharp image of diagonal edges (Fig. 2(d)), since it has small TV$_{\ell_1,\gamma=1}$ and is closest to the original image (Fig. 2(a)).

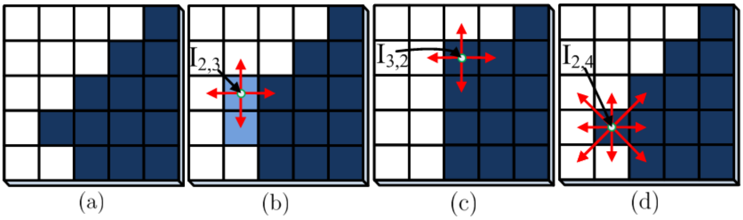


Fig. 2. Comparison of TV measures (the intensities of white, dark-blue and light-blue pixels are 1, 0 and 0.5). (a) Original sharp corner, (b) blurred image recovered by minimizing $\mathrm{TV}_{\ell_1\ell_2}$, (c) straight edge image recovered by minimizing $\mathrm{TV}_{\ell_1'}$, (d) diagonal corner image recovered by minimizing $\mathrm{TV}_{\ell_1,\gamma=1}$.

Table 1. The sparsity of a 256× 256 LENA image in the field of gradient magnitudes and the four partial gradient fields. The partial gradient fields has similar sparsity, which is much smaller than that of the gradient magnitude field.

$\sqrt{\mathbf{Gx}^2+\mathbf{Gy}^2}$	$\mathbf{Gx}$	$\mathbf{Gy}$	$\mathbf{Gxy}$	$\mathbf{Gyx}$
40652	28111	27767	28760	28633

2.2 UUP Condition for TV_{ℓ_1} Based CS

In this section, we present the UUP condition for TV based compressive sampling. According to this UUP condition, we compare our TV_{ℓ_1} and previous $\mathrm{TV}_{\ell_1\ell_2}$ in terms of the number of measurements for lossless recovery. An image $\mathbf{I}$ can be represented as a linear combination of each partial gradient field plus some constant values, i.e., $\mathbf{I} = \Psi_x\mathbf{Gx} + \mathbf{I_x} = \Psi_y\mathbf{Gy} + \mathbf{I_y} = \Psi_{xy}\mathbf{Gxy} + \mathbf{I_{xy}} = \Psi_{yx}\mathbf{Gyx} + \mathbf{I_{yx}}$, where constant vectors $\mathbf{I_x}$, $\mathbf{I_y}$, $\mathbf{I_{xy}}$ and $\mathbf{I_{yx}}$ are equal to some rearrangements of the first row pixels as well as the first and last column pixels. For instance, $\mathbf{I_x}$ is a repetition of the first column pixels. According to (4), $b = \Phi\mathbf{I} = \Phi\Psi_x\mathbf{Gx} + \Phi\mathbf{I_x} = \Phi\Psi_y\mathbf{Gy} + \Phi\mathbf{I_y} = \Phi\Psi_{xy}\mathbf{Gxy} + \Phi\mathbf{I_{xy}} = \Phi\Psi_{yx}\mathbf{Gyx} + \Phi\mathbf{I_{yx}}$. Suppose $\gamma = 1$ and no LRS for simplicity, our TV_{ℓ_1} based CS problem (4) is reformulated as:

$$\min_{\mathbf{G}} \|\mathbf{G}\|_1 \quad \text{s. t.} \quad A\mathbf{G} = [\mathbf{b_x}; \mathbf{b_y}; \mathbf{b_{xy}}; \mathbf{b_{yx}}] \tag{7}$$

where the partial gradient fields $\mathbf{G} = [\mathbf{G}_1; \mathbf{G}_2; \mathbf{G}_3; \mathbf{G}_4] = [\mathbf{Gx}; \mathbf{Gy}; \mathbf{Gxy}; \mathbf{Gyx}]$, the sampling matrix $A = diag(A_1, A_2, A_3, A_4) = diag(\Phi\Psi_x, \Phi\Psi_y, \Phi\Psi_{xy}, \Phi\Psi_{yx})$, and the sampled data $[\mathbf{b_x}; \mathbf{b_y}; \mathbf{b_{xy}}; \mathbf{b_{yx}}] = [\mathbf{b} - \Phi\mathbf{I_x}; \mathbf{b} - \Phi\mathbf{I_y}; \mathbf{b} - \Phi\mathbf{I_{xy}}; \mathbf{b} - \Phi\mathbf{I_{yx}}]$.

If replacing the objective function with $\sqrt{\mathbf{Gx}^2 + \mathbf{Gy}^2}$, we induce the $\mathrm{TV}_{\ell_1\ell_2}$ based CS problem. The major difference between these two TV is $\mathrm{TV}_{\ell_1\ell_2}$ enforces the sparsity in the gradient magnitude fields and TV_{ℓ_1} enforces that of partial gradients.

Now, we compare the sparsity (denote its maximal value as K_1) in each partial gradient field and that (denoted as K_2) in the gradient magnitude field. For most natural images, it is generally true that $K_1 \leq K_2$, as shown in Table 2. In the gradient magnitude field $\sqrt{\mathbf{Gx}^2 + \mathbf{Gy}^2}$, K_2 is equal to the size of pixels having non-zero

$\mathbf{Gx}$ or $\mathbf{Gy}$. Thus, K_2 is larger than both the sparsity of $\mathbf{Gx}$ and that of $\mathbf{Gy}$. At each pixel, the gradient magnitude is equal to $\sqrt{Gx_{i,j}^2 + Gy_{i,j}^2}$, the diagonal gradients $Gxy_{i,j} = Gx_{i,j} + Gy_{i+1,j}$ and $Gyx_{i,j} = Gy_{i,j} - Gx_{i,j}$. So, the sparsity of diagonal gradients $\mathbf{Gxy}$ or $\mathbf{Gyx}$ is smaller than K_2, which equals the size of pixels having non-zero $\mathbf{Gx}$ or $\mathbf{Gy}$. Thus, we prove that $K_1 \leq K_2$ for any image.

According to (7), each individual sampling matrix $A_i, i = 1, 2, 3, 4$, corresponds to a partial gradient field $\mathbf{G}_i$ (size $N^2 \times 1$, image size: $N \times N$). For each random sampling matrix $A_i \in \mathbb{R}^{M \times N^2}, 1 \leq i \leq 4$ to obey the UUP condition (2), the inequality

$$\frac{1}{2} \cdot \frac{M}{N^2} \|\mathbf{G}_i\|_2^2 \leq \|A_i \mathbf{G}_i\|_2^2 \leq \frac{3}{2} \cdot \frac{M}{N^2} \|\mathbf{G}_i\|_2^2 \tag{8}$$

must hold for any partial gradient $\mathbf{G}_i$ whose sparsity satisfies $K_1 \leq M/(\alpha \log(N^2/M))$. In other words, each $A_i \in \mathbb{R}^{M \times N^2}$ obeys the UUP condition, provided that $M \geq \alpha K_1 \log(N^2/K_1)$. In our TV$_{\ell_1}$ based CS (7), we need to induce the UUP condition of the big matrix A which involves all four gradient fields $\mathbf{G}$. By summing the 4 components in (8), we obtain the inequality for the matrix A:

$$\begin{aligned} \frac{1}{2} \cdot \frac{M}{N^2} \sum_i \|\mathbf{G}_i\|_2^2 \leq \textstyle\sum_i \|A_i \mathbf{G}_i\|_2^2 \leq \frac{3}{2} \cdot \frac{M}{N^2} \sum_i \|\mathbf{G}_i\|_2^2 \\ \frac{1}{2} \cdot \frac{M}{N^2} \|\mathbf{G}\|_2^2 \leq \quad \|A\mathbf{G}\|_2^2 \quad \leq \frac{3}{2} \cdot \frac{M}{N^2} \|\mathbf{G}\|_2^2 \end{aligned} \tag{9}$$

Obviously, the combined sampling matrix A obeys the UUP condition, given that each sampling matrix $A_i, i = 1, 2, 3, 4$ obeys the UUP condition, or given the condition $M \geq \alpha K_1 \log(N^2/K_1)$. Suppose the gradient magnitude is sampled randomly, we can induce that the number of measurements required by previous TV$_{\ell_1 \ell_2}$ based CS is $M \geq \alpha K_2 \log(N^2/K_2)$.

Therefore, our TV$_{\ell_1}$ based compressive sampling requires fewer samples than TV$_{\ell_1 \ell_2}$ for the same quality of reconstruction. In other words, based on the same number of measurements, our TV$_{\ell_1}$ based CS will recover an image of higher quality.

2.3 Optimal Hybrid Sampling

For most natural images, our hybrid sampling (4) consisting of low-resolution sampling (LRS) and random sampling (RS) requires fewer measurements than random sampling alone for the same quality of reconstruction. In this section, we will give a theoretical analysis on the optimal hybrid sampling and its minimal number of measurements for lossless reconstruction.

Both low resolution sampling (LRS) and random sampling (RS) aim at reducing the size of sampled data non-adaptively. The major difference is that LRS measures the low-frequency information with averaging filter (block size: $n \times n$, frequency $F = 1/n$) while RS senses the combination of randomly-selected data.

To demonstrate how RS and LRS complement each other in our TV$_{\ell_1}$ based CS, we develop a hierarchical gradient transform (HGT), similar to Wavelet transform. HGT consists of an average basis Ψ'' at the coarsest level and a series of difference bases

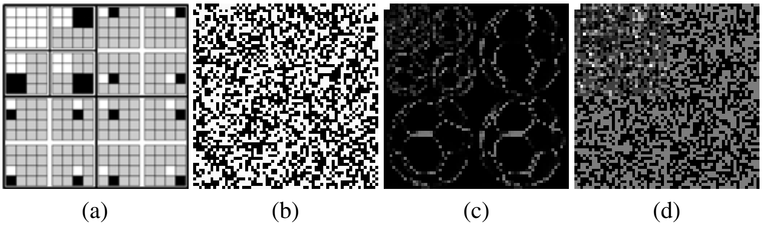


Fig. 3. (a)Hierarchical gradient transform (HGT), (b) a Bernoulli random matrix, magnitude of HGT of (c) a Ball image and (d) the Bernoulli random matrix

Ψ' at finer levels (Fig. 3(a)). Consider a 2×2 block at the finest level, all the partial gradients inside this block are highly correlated. Thus, our HGT represents these partial gradients by three partial gradients at the left-upper pixel. Similarly, we can de-correlate the partial gradients in larger scale 2×2 blocks at coarser levels, as shown in Fig. 3(a). Thus, given an image, HGT outputs a series of hierarchical gradients $\mathbf{G}'$ and some average responses.

For a piecewise smooth image, $\mathbf{G}'$ has denser non-zero elements at coarser levels (Fig. 3(c)) while Bernoulli random sampling (RS) senses $\mathbf{G}'$ almost uniformly in the HGT domain (Fig. 3(d)). Thus, sole RS is not efficient and hybrid sampling is desired. In hybrid sampling (4), given LR samples $\mathbf{d}$ at coarser levels, we measure $\mathbf{G}'$ on the rest finer levels (denoted by $\mathbf{G}'_d$) by random sampling (RS). Since $\mathbf{G}'_d$ is quite sparse, hybrid sampling sacrifices some low-resolution samples $\mathbf{d}$ for dramatically reducing the number of RS measurements $\mathbf{b}$.

An $N \times N$ image $\mathbf{I}$ can be represented as linear combinations of LR samples $\mathbf{d}$ on coarser scales and K'_d-sparsity $\mathbf{G}'_d$ associated with $\mathbf{d}$, $\mathbf{I} = \Psi'\mathbf{G}'_d + \Psi''\mathbf{d}$. For the sake of simplicity, we approximate our TV_{ℓ_1} minimization by enforcing the sparsest $\mathbf{G}'_d$, and reformulate (4) as follows:

$$\min_{\mathbf{G}'_d} \|\mathbf{G}'_d\|_1 \quad \text{s. t.} \quad A'\mathbf{G}'_d = \Phi\Psi'\mathbf{G}'_d = \mathbf{b} - \Phi\Psi''\mathbf{d} \tag{10}$$

where A' is the sampling matrix. The minimal number of measurements for the lossless reconstruction is $\alpha K'_d \log(N^2/K'_d)$, where α is a constant.

Proposition 1. *The hybrid sampling approach consisting low-resolution sampling ($F = 1/n$) and M random projections is capable of recovering the original image (size $N \times N$), if the sampling matrix A' obeys UUP [2] for the unknown K'_d-sparsity coefficients $\mathbf{G}'_d$ at the finer levels of HGT. Consequently, for lossless reconstruction, the minimal number of measurements M_{min} equals $(N/n)^2 + \alpha K'_d \log((N^2 - (N/n)^2)/K'_d)$, where α is a constant.*

The optimal hybrid sampling depends on selection of LRS, which is defined by its frequency ($F = 1/n$) and other parameters, such as d and K'_d. By varying LRS and its corresponding RS, we can seek the optimal hybrid sampling with the smallest number of measurements ($\hat{M}_{min} = (N/\hat{n})^2 + \alpha\hat{K}'_d \log((N^2 - (N/\hat{n})^2)/\hat{K}'_d)$), where $1/\hat{n}$ is the frequency of the optimal LRS.

3 Implementation Issues

3.1 Practical Hybrid Sampling

One problem with random sampling is its inefficiency for large-scale images. The notable CS application of random sampling is Single Pixel Camera (SPC)[22,8], which is advantageous over the conventional pixel-array camera in reducing sampling rate (ratio of sample size and data size, denoted as R). It sequentially acquires random linear measurements of scene brightness by a digital micro-mirror (DMD) and thus its sensing rate is limited. To date, DMD can provide at most 32000 random patterns/second. Suppose we need to capture an image of size 1024×768 at $R = 10\%$, then the sensing process takes $1024 \times 768 \times 0.1/32000 = 2.46$ seconds. Our hybrid sampling can increase the frame rate (RS) by incorporating some LR samples and even reduce the total sampled data from RS and LR, for the same quality of reconstruction.

Another problem with random sampling is its high computational cost. For instance, to recover a 1024×768 image at $R = 10\%$, we need more than 7 gigabytes of memory just to store the Bernoulli random matrix. To reduce the cost of time and memory, many efforts have been made to develop structural sampling methods (Fourier transform[12], scrambled Fourier[1], Hadarmard transform[9], Noiselet[6,3]). A typical application of structural sampling is Magnetic Resonance Imaging (MRI)[12] using Fourier sampling.

3.2 Sparsity Decoding

In this section, we present our approach to recover the image from the limited measurements by decoding the sparse gradient $\mathbf{G}$. There is a number of algorithms available for decoding, including Orthogonal Matching Pursuit (OMP)[23], Basis Pursuit(BP)[5] listed in SparseLab Toolbox[21], second-order cone programming (SOCP) implemented in ℓ_1-Magic[11], and iterative shrinkage/thresholding (IST)[24].

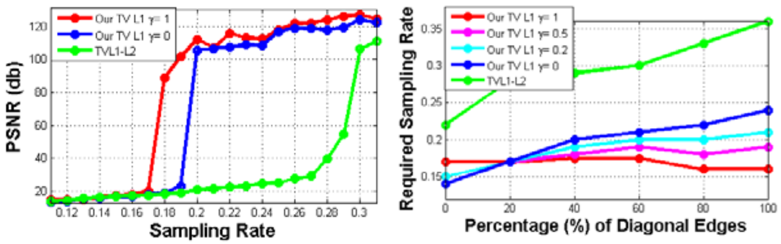
For decoding, we aim to solve (4) and recover the image $\mathbf{I}$ and its sparse partial gradients $\mathbf{G}$. We employ a primal-dual interior-point optimization routine called PDCO [19]. Since random sampling is computationally costly, we need to replace it by Fourier sampling for sensing large-scale images. Given the partial Fourier data $\mathbf{b}$, we use the IST method [24] to solve (4) to recover the image $\mathbf{I}$.

4 Experimental Results

In this section, we present some experimental results to compare our hybrid compressive sampling using TV_{ℓ_1}, with the widely-used $\text{TV}_{\ell_1\ell_2}$ based CS method. We present results for both qualitative (visual) and quantitative evaluations.

4.1 Selection of Parameter γ

As shown in (6), our TV_{ℓ_1} seeks the γ-weighted gradient sparsity and recovers images with sharp edges. For a sharp image containing 40% diagonal edges, our TV_{ℓ_1} can achieve much higher accuracy than $\text{TV}_{\ell_1\ell_2}$and its accuracy depends on selection of γ (Fig. 4(a)). As shown in Fig. 4(b), in comparison with other TV measure, our TV_{ℓ_1}



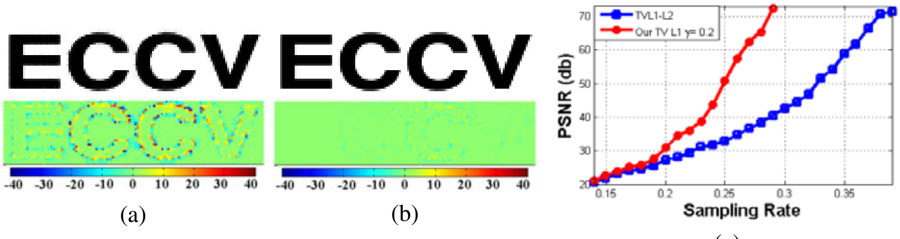
(a) (b)

Fig. 4. Comparison of TV measures. (a) The recovery accuracy of a sharp image in which 40% of edges are diagonal. (b) The required sampling rates on different images, for the recovery accuracy (PSNR) to be large than 40dB.

($\gamma = 1$) requires fewer samples at images containing many diagonal edges and more samples at images containing few diagonal edges, for the same recovery accuracy. That means, the value of the optimal γ should be proportional to the percentage of diagonal edges in the image. This result is consistent with the claim that our TV$_{\ell_1}$ can recovery all possible sharper edges (vertical, horizontal or diagonal) in Sect. 2.1. In our following experiments, the optimal γ is selected as $0.2 \leq \gamma \leq 1$.

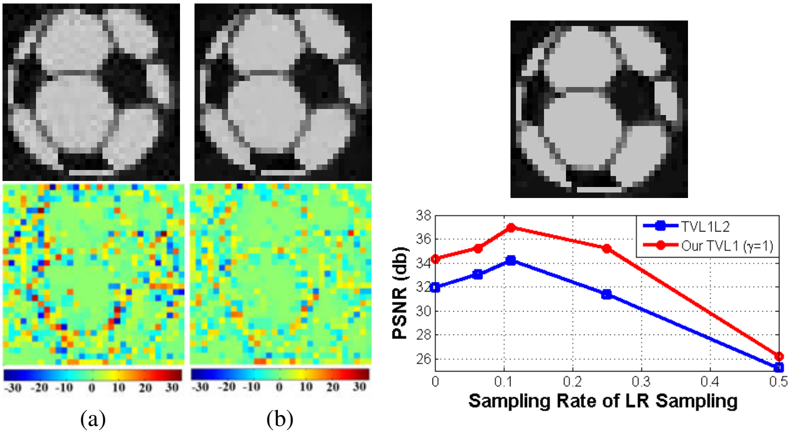
4.2 Hybrid Compressive Sampling via Our TV$_{\ell_1}$

To show the advantage of our TV$_{\ell_1}$ over TV$_{\ell_1\ell_2}$, we choose small piecewise smooth images, e.g., ECCV image in Fig. 5 and Ball image in Fig. 6, due to the expensive sparsity decoding. As shown in Fig. 5, our TV$_{\ell_1}$ based CS is able to reconstruct the sharp ECCV image almost perfectly while TV$_{\ell_1\ell_2}$ causes serious artifacts at $R = 25\%$. Our TV$_{\ell_1}$ is still advantageous over previous TV$_{\ell_1\ell_2}$ at varying sampling rates (Fig. 5(c)). As shown in Fig. 6, Ball image is almost a real image, except that we remove some noise in the gray region. Given the same sampled data, our TV$_{\ell_1}$ acquires an image (Fig. 6(b)) whose Peak-Signal-Noise-Ratio (PSNR) is 3.0dB higher than that recovered by previous TV$_{\ell_1\ell_2}$ (Fig. 6(a)).



(a) (b) (c)

Fig. 5. Recovered ECCV images (upper) and error maps (lower) by (a)TV$_{\ell_1\ell_2}$ (PSNR=32.88dB) and (b) TV$_{\ell_1}$ (PSNR=48.17dB) at the sampling rate $R = 25\%$ (LRS:6.25% and RS:18.75%). (c) Comparison of TV measures on ECCV image sensed by our hybrid sampling with LRS ($F = 1/4$).



(a) (b)

Fig. 6. Given random sampling (41%) and LR sampling ($F = 1/3$) on Ball image (upper-right), images recovered by (a) TV$_{\ell_1\ell_2}$ (PSNR=29.8dB) with its error map, and (b) TV$_{\ell_1}$ (PSNR=32.8dB) with its error map. Given the fixed total hybrid sampling rate ($R = 60\%$), we show the recovery accuracy of TV$_{\ell_1}$ and TV$_{\ell_1\ell_2}$ at varying LR sampling rates (lower-right).

Table 2. The estimated and real minimal number of required measurements on the Ball image (N=32), for each hybrid sampling methods associated with different LRS (block size: $n \times n$)

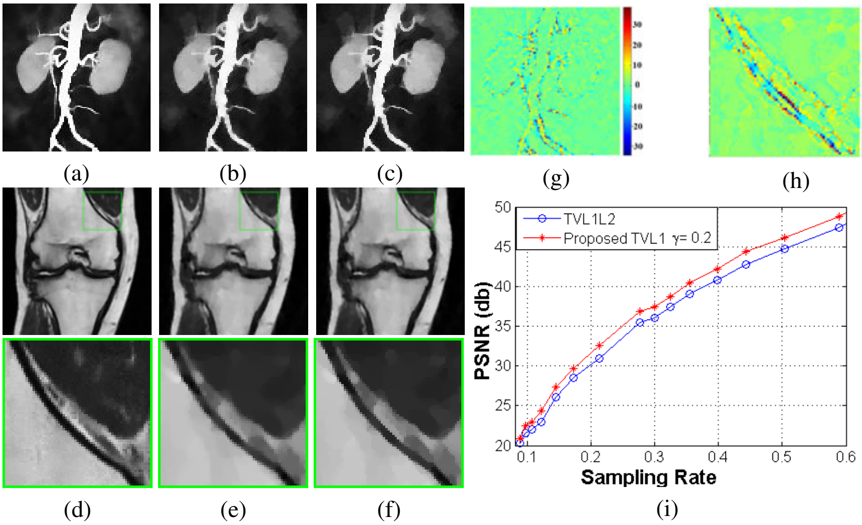
$n \times n$ LRS	K_1'	$Esti.M_{min}$ at $\alpha = 1.2$	$RealM_{min}$ for PSNR $\geq$ 40 dB
No LRS	576	$0 + 576\alpha$= 691	717
4×4	512	$64 + 512\alpha = 678$	680
3×3	440	$121 + 440\alpha = 649$	653
2×2	368	$256 + 368\alpha = 698$	665
2×1	250	$512 + 235\alpha = 794$	756

For most natural images, low-resolution sampling (LRS)and random sampling (RS) can complement each other. For instance, the recovery accuracy of our TV$_{\ell_1}$ is improved by combining RS with LRS ($F = 1/3$) on Ball image (Fig. 6). As shown in Fig. 6, both our TV$_{\ell_1}$ and previous TV$_{\ell_1\ell_2}$ achieve the optimal accuracy at the LRS ($F = 1/3$), given the total sampling rate $R = 60\%$.

According to Proposition 1, we can determine the optimal hybrid sampling that requires the fewest samples M_{min}. Now, we want to verify Proposition 1 by some experimental results. Since K_1' is comparable to $N^2 - (N/n)^2$, we approximate the estimated M_{min} by $(N/n)^2 + \alpha K_1'$. Table 4.2 shows one successful case ($\alpha = 1.2$), in which our estimated M_{min} is close to the real M_{min} required to achieve the accuracy (PSNR = 40dB). At $\alpha = 1.2$, hybrid sampling with LRS ($F = 1/3$) requires the smallest M_{min} and thus is optimal, which is consistent with the accuracy chart in Fig. 6.

4.3 Evaluation of Our TV$_{\ell_1}$ by Fourier Sampling

Now, we evaluate our TV_{ℓ_1} based CS on two real MR images (Chest and Bone) by Fourier sampling ([12]). Given 14% Fourier samples, our TV_{ℓ_1} can recover a Chest image I_1 (Fig. 7(c)), whose PSNR is 1.3dB higher than that I_2 (Fig. 7(b)) by $TV_{\ell_1\ell_2}$. Figure 7(g) shows the difference map $I_d = I_1 - I_2$, which is close to the second derivatives of Chest image (Fig. 7(a)). Similarly, the region boundary (Fig. 7(f)) in Bone image recovered from our TV_{ℓ_1} is obviously sharper than that in Fig. 7(e), which is also demostrated by their difference map (Fig. 7(h)). Thus, our TV_{ℓ_1} is prone to enforcing sparse partial gradients in piecewise smooth images. Besides, our TV_{ℓ_1} achieves higher accuracy at varying sampling rates than $TV_{\ell_1\ell_2}$ in recovering these images, as shown in Fig. 7(i).



(a) (b) (c) (g) (h)

(d) (e) (f) (i)

Fig. 7. Comparison of TV measures by Fourier sampling. (a) Original Chest image sensed at $R = 14\%$, images recovered by (b) TV$_{\ell_1\ell_2}$ (PSNR= 26.0dB) and (c) TV$_{\ell_1}$ (PSNR=27.3dB). (d) Original Bone image sensed at $R = 9.34\%$, images recovered by (e) TV$_{\ell_1\ell_2}$ (PSNR=27.1dB) and (f) TV$_{\ell_1}$ (PSNR=27.6dB). (g) Difference of (c)and (b). (h) Difference of (f)and (e). (i) Accuracy vs. sampling rate on Chest image.

5 Conclusion

In this paper, we propose a hybrid compressive sampling method using a new TV measure TV_{ℓ_1}, for recovering a piecewise smooth image containing all possible sharper edges from limited measurements. We induce a UUP condition for TV based compressive sampling, which shows that our TV_{ℓ_1} requires fewer measurements than widely used $TV_{\ell_1\ell_2}$ for the same quality reconstruction. In addition, some theoretical analysis is presented to show the advantage of hybrid sampling over random sampling for most natural images and how to seek the optimal hybrid sampling. Finally, our TV_{ℓ_1} based hybrid CS achieves better performance in experimental results.

References

1. Candes, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Comm. Pure Appl. Math. 59(8), 1208–1223 (2006)
2. Candes, E., Tao, T.: Near-optimal signal recovery from random projections and universal encoding strategies? IEEE Transactions on Information Theory 52(12), 5406–5245 (2006)
3. Candes, E., Romberg, J.: Sparsity and incoherence in compressive sampling. Inverse Prob. 23(3), 969–986 (2007)
4. Candes, E., Tao, T.: Decoding by linear programming. IEEE Transactions on Information Theory 51, 4203–4215 (2005)
5. Chen, S., Donoho, D.: Atomic decomposition by basic pursuit. SIAM J. Sci. Comp. 20, 33–61 (1998)
6. Coifman, R., Geshwind, F., Meyer, Y.: Noiselets. Appl. Comp. Harmonic Analysis 10, 27–44 (2001)
7. Donoho, D.: Compressed sensing. IEEE Trans. on Information Theory (2006)
8. Duarte, M.F., Davenport, M.A., Takhar, D., et al.: Single-pixel imaging via compressive sampling. IEEE Signal Processing Magazine 25(2), 83–91 (2008)
9. Gan, L., Do, T., Tran, T.: Fast compressive imaging using scrambled block hadamard ensemble. EUSIPCO
10. He, L., Chang, T.C., Osher, S., Fang, T., Speier, P.: Mr image reconstruction by using the iterative refinement method and nonlinear inverse scale space methods. UCLA CAM Report, pp. 06–35 (2006)
11. L1-magic: http://www.acm.caltech.edu/l1magic
12. Lustig, M., Donoho, D., Santos, J., Pauly, J.: Compressed sensing mri. IEEE Sig. Proc. Magazine (2007)
13. Ma, S., Yin, W., Zhang, Y., Chakraborty, A.: An efficient algorithm for compressed mr imaging using total variation and wavelets. CVPR (2008)
14. Maleh, R., Gilbert, A.C., Strauss, M.J.: Sparse gradient image reconstruction done faster. ICIP 2, 77–80 (2007)
15. Natarajan, B.K.: Sparse approximate solutions to linear systems. SIAM Journal on Computing 24, 227–234 (1995)
16. Romberg, J.: Variational methods for compressive sampling. Proc. SPIE 6498, 64980J–2–5 (2007)
17. Romberg, J.: Imaging via compressive sampling. Comm. Pure Appl. Math, 14–20 (2008)
18. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D Nonlinear Phenomena 60(1), 259–268 (1992)
19. Saunders, M.A.: Pdco: Primal-dual interior-point method for convex objectives. Systems Optimization Laboratory, Stanford University (2002)
20. Skodras, A., Christopoulos, C., Ebrahimi, T.: The jpeg2000 still image compression standard. IEEE Signal Processing Mag. 18, 36–58 (2001)
21. SparseLab: http://sparselab.stanford.edu
22. Takhar, D., Laska, J., Wakin, M., Duarte, M., et al.: A new compressive imaging camera architecture using optical-domain compression. Proc. of Computational Imaging IV at SPIE Electronic Imaging 6065, 43–52 (2006)
23. Tropp, J.A., Gilbert, A.C.: Signal recovery from partial information via orthogonal matching pursuit. IEEE Transactions on Information Theory 53, 4655–4666 (2007)
24. Yang, J., Zhang, Y., Yin, W.: A fast tv-l1-l2 algorithm for image reconstruction from partial fourier data. To be submitted to IEEE Trans. on Special Topics (2008)

Perspective Imaging under Structured Light

Prasanna Rangarajan, Vikrant Bhakta, Marc Christensen,
and Panos Papamichalis

Lyle School of Engineering,
Southern Methodist University, Dallas, U.S.A.
{prangara,vrbhakta,mpc,panos}@lyle.smu.edu

Abstract. Traditionally, “Structured Light” has been used to recover surface topology and estimate depth maps. A more recent development is the use of “Structured Light” in surpassing the fundamental limit on spatial resolution imposed by diffraction. But, its use in surpassing the diffraction limit remains confined to microscopy, due to issues that arise in macroscopic[1] imaging: perspective foreshortening, aliasing and need for calibration. Also, no formal attempt has been made to unify the above embodiments, despite their common reliance on “Structured Light”.

An original contribution of this work is the use of “Structured Light” in surpassing the diffraction limit of macroscopic imaging systems. Other contributions include

- unifying the “Structured Light” embodiments in a single framework
- realizing *OSR* and depth-estimation in a single un-calibrated setup

when the image planes of the imaging *&* illumination system are parallel. Potential applications include bar code scanning and surveillance.

Keywords: Structured Light, Structured Illumination, Diffraction, Optical Super-Resolution, Super-Resolution, Depth Estimation, PROCAMS.

1 Introduction

The term ”Structured Light” refers to periodic light patterns such as the light emerging from sunlit ”venetian blinds”, or the physical projection of a periodic pattern printed on a transparency. It provides a means to impose an artificial pattern of known spatial frequency, upon a scene.

The earliest attempts at using “Structured Light” in optics[1–3] were focussed on the accurate measurement of surface topology. The idea is to project a sinusoidal illumination pattern onto the target surface, at a known angle. The reflected image viewed from a different position (and or angle) reveals lateral displacements and frequency changes (*Fig.* 1(a)) that are related to topological variations. The same principle governs “Structured Light” Depth Estimation in computer vision[4–6]. Excellent overviews of the topic are presented in[7–11].

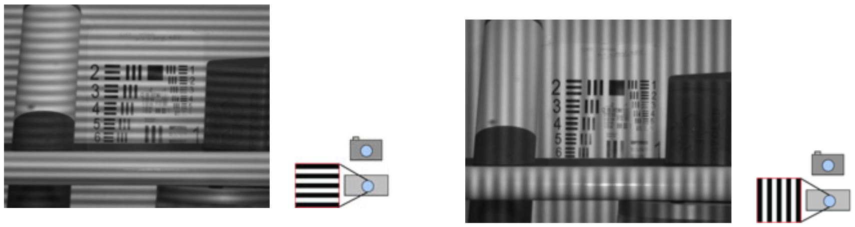
[1] We use the term macroscopic imaging to refer to imaging systems that exhibit significant perspective distortion, and demagnification (images of objects are smaller than their true size).

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 405–419, 2010.

Recently, "Structured Light" has been used in microscopy, [12–16] to recover spatial frequencies that are lost to diffraction-induced-blurring[17]. The technology is referred to as "Optical Super Resolution" (*OSR*), and relies on the ability to shift spatial frequencies outside the passband of the imaging system into the passband. It owes its success to the seminal work of Lukosz & Marchand in 1963 [18], who proposed a method for shifting spatial frequencies, by modulating the amplitude of a periodic pattern with scene information.

Despite advances in "Structured Light"-microscopy, realizing *OSR* in macroscopic imaging, remains an open problem. The challenge lies in realizing amplitude modulation, while overcoming the distortion in the illuminating pattern as viewed by the imaging system (*Fig.*[1(a)]) . The distortion arises due to the displacement between the center-of-perspective (*COP*) of the imaging & illumination systems. It has been observed that the distortion can be overcome by coinciding the *COP*'s using a beam splitter[19], or collocating them such that the periodic pattern is aligned with the epipolar lines[20] (*Fig.*[1(b)]). But, it remains to be proven that such arrangements can shift spatial frequencies outside the passband of a macroscopic imaging system, into the passband.

The above issue is addressed in *Section.*(3), with the aid of the model proposed in *Section.*(2). The model also provides a unified treatment of "Structured Light" imaging, when the image planes of the camera & projector are parallel. This allows us to explore the possibility of realizing *OSR* and depth-estimation (*Section.*(4)) in a single setup. Experiments in *Section.*(5) and the supplementary material, confirm our findings.



(a) The phase of the pattern is distorted due to bending at depth discontinuities.

(b) The pattern appears undistorted and superimposed onto image of the scene.

Fig. 1. A camera observing a 3D scene illuminated by a sinusoidal light pattern, in a canonical stereo setup with vertical epipolar lines

2 Structured Light Imaging in a Parallel Stereo Setup

In this section, we develop a mathematical model for the relationship between the intensity of a projector pixel and its corresponding camera pixel, when the optical axes of the projector and camera are parallel, as shown in *Fig.*[2(a)].

(X, Y, Z)	coordinates of a scene point in the world coordinate system (coordinate convention is shown in *Fig.* 2(a))
Bold uppercase letters	points in the world coordinate system
Roman lowercase letters	pixel coordinates in the camera *&* projector image planes, and scalars in general
Serif lowercase letters	signals *&* images
$\mathcal{CALLIGRAPHIC}$ letters	Fourier Transform of signals

Suppose

- $\overrightarrow{\mathbf{O}_c\mathbf{O}_p} \triangleq [b_x,\ b_y,\ b_z]^T$ is the baseline between the camera and the projector, whose center-of-perspective (COP) are at $\mathbf{O}_c$ and $\mathbf{O}_p$ respectively
- The projector and camera image planes are located at distances Z_p, Z_c behind the respective COP's
- s_p, s_c represent the size of a projector *&* camera pixel respectively in $\frac{\texttt{mm}}{\texttt{pixel}}$
- $M_p \times N_p$, $M_c \times N_c$ (rows × columns) represent the size of the projector *&* camera images in `pixels`
- $x_c\ y_c$, $x_p\ y_p$ represent the camera *&* projector image coordinates respectively.

Assumptions

- The camera point spread function (psf) $\mathsf{h}(x, y)$ does not change appreciably within the projector depth of field
- The images captured by the camera are strictly diffraction limited, and free of aliasing (optical cutoff frequency < detector Nyquist frequency)

At the outset, we assume the illumination pattern is a raised sine pattern with 2D spatial frequency (ξ_0, η_0) $\frac{\texttt{cycles}}{\texttt{image}}$. This assumption will be relaxed to accommodate arbitrary periodic patterns in *Section.*(2.4).

Suppose that the intensity of the $(x', y')^{th}$ projector pixel is given by

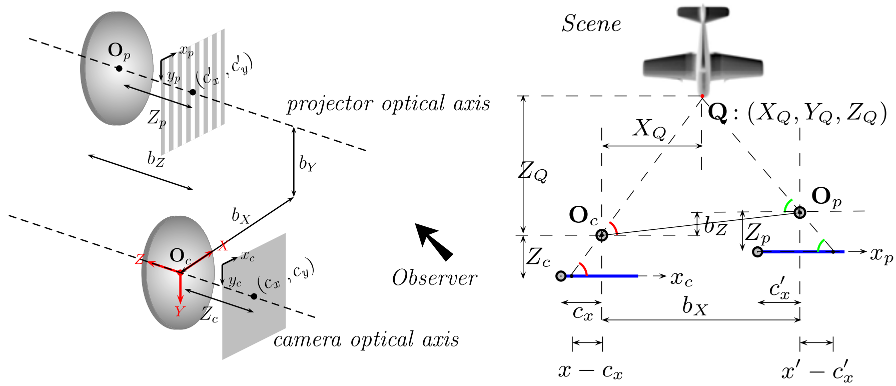
$$\mathsf{s}_\theta(x', y') = \frac{A}{2} + \frac{A}{2}\ sin\left(2\pi\left(\frac{\xi_0}{N_p}x' + \frac{\eta_0}{M_p}y'\right) + \theta\right) \qquad \begin{array}{l} 0 \le x' < N_p \\ 0 \le y' < M_p \end{array} \tag{1}$$

The term A in *Eq.* 1 represents a scalar constant, and θ represents the phase of the sinusoidal pattern. The 2D spatial frequency (ξ_0, η_0) in *Eq.* 1 is deliberately expressed in the image independent unit $\frac{\texttt{cycles}}{\texttt{image}}$, as it directly correlates with the bin index in the 2D-*DFT* (*Discrete Fourier Transform*) of $\mathsf{s}_\theta(x', y')$.

2.1 Geometric Relationship between a Scene Point and Its Corresponding Projector and Camera Pixels

Suppose (x', y') and (x, y) are the pixel projections of the scene point (X_Q, Y_Q, Z_Q) onto the projector and camera image planes, as illustrated in *Fig.* 2(b). Simple geometry reveals[2] that

[2] The expressions for x, x' follow from the geometry of the similar triangles corresponding to the *red*, *green* angles in *Fig.* 2(a). The expressions for y, y' can be derived along similar lines.



(a) *Schematic of Parallel Stereo Setup* (b) *Top view of the setup*

Fig. 2. *Parallel Stereo Camera + Projector Setup*

XYZ : World Coordinate System, $\mathbf{O}_c/\mathbf{O}_p$: lens center of Camera/Projector
$(x_c, y_c)/(x_p, y_p)$: Camera/Projector Image Planes in `pixel` units
(b_X, b_Y, b_Z) : displacement between the Camera & Projector lens centers
(c_x, c_y) / (c'_x, c'_y) : principal point of Camera/Projector

$$\boxed{(X_Q, Y_Q, Z_Q) \rightarrow (x, y)} \quad \frac{x - c_x}{Z_c} = -\frac{1}{s_c}\frac{X_Q}{Z_Q} \quad , \quad \frac{y - c_y}{Z_c} = -\frac{1}{s_c}\frac{Y_Q}{Z_Q} \tag{2}$$

$$\boxed{(X_Q, Y_Q, Z_Q) \rightarrow (x', y')} \quad \frac{x' - c'_x}{Z_p} = -\frac{1}{s_p}\left(\frac{X_Q - b_X}{Z_Q - b_Z}\right) \quad , \quad \frac{y' - c'_y}{Z_p} = -\frac{1}{s_p}\left(\frac{Y_Q - b_Y}{Z_Q - b_Z}\right) \tag{3}$$

$$\boxed{(x, y) \rightarrow (x', y')} \quad \begin{aligned} x' &= \left(\tfrac{Z_Q}{Z_Q \;\; b_Z}\right) \tfrac{Z_p}{s_p}\, \tfrac{s_c}{Z_c}(x - c_x) + \tfrac{Z_p}{s_p}\left(\tfrac{b_X}{Z_Q \;\; b_Z}\right) + c'_x \\ y' &= \left(\tfrac{Z_Q}{Z_Q \;\; b_Z}\right) \tfrac{Z_p}{s_p}\, \tfrac{s_c}{Z_c}(y - c_y) + \tfrac{Z_p}{s_p}\left(\tfrac{b_Y}{Z_Q \;\; b_Z}\right) + c'_y \end{aligned} \tag{4}$$

2.2 Effect of Projector Defocus on the Illumination Pattern

In an effort to produce bright images with large field of view, projectors are designed to have large apertures. The downside is that they have a shallow depth of field, which compels us to accommodate the depth dependent blurring (*Fig.*[3]) of the projected "Structured Light" pattern.

Using *Eq.*[4] in *Eq.*[1], we can express the incident intensity at the scene point (X_Q, Y_Q, Z_Q) due to the illumination $\mathsf{s}_\theta(x', y')$, in the camera coordinates (x, y) as follows

$$\begin{aligned} &\mathsf{p}_\theta(X_Q, Y_Q, Z_Q) = A_{0,0,Z_Q} \; + \\ &A_{\xi_0,\eta_0,Z_Q} \; sin\left(2\pi\left(\begin{matrix} \left(\tfrac{Z_Q}{Z_Q \;\; b_Z}\right)\tfrac{Z_p}{Z_c}\,\tfrac{s_c N_c}{s_p N_p}\,\tfrac{\xi_0}{N_c}(x - c_x) + \tfrac{\xi_0}{N_p}c'_x + \tfrac{Z_p}{s_p}\,\tfrac{\xi_0}{N_p}\left(\tfrac{b_X}{Z_Q \;\; b_Z}\right) \; + \\ \left(\tfrac{Z_Q}{Z_Q \;\; b_Z}\right)\tfrac{Z_p}{Z_c}\,\tfrac{s_c M_c}{s_p M_p}\,\tfrac{\eta_0}{M_c}(y - c_y) + \tfrac{\eta_0}{M_p}c'_y + \tfrac{Z_p}{s_p}\,\tfrac{\eta_0}{M_p}\left(\tfrac{b_Y}{Z_Q \;\; b_Z}\right) \end{matrix}\right) + \theta\right) \end{aligned} \tag{5}$$

The real scalars $A_{0,0,Z_Q}$, A_{ξ_0,η_0,Z_Q} represent the effect of defocus MTF on the spatial frequencies $(0, 0)$, (ξ_0, η_0) at depth Z_Q.

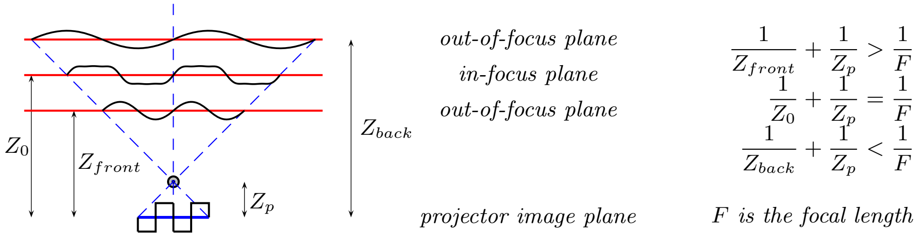

Z_0
Z_{front}
Z_p
Z_{back}
out-of-focus plane
in-focus plane
out-of-focus plane
projector image plane
$\frac{1}{Z_{front}} + \frac{1}{Z_p} > \frac{1}{F}$
$\frac{1}{Z_0} + \frac{1}{Z_p} = \frac{1}{F}$
$\frac{1}{Z_{back}} + \frac{1}{Z_p} < \frac{1}{F}$
F is the focal length

Fig. 3. The effect of projector defocus on a square pattern is a depth dependent blurring of each frequency component in the square pattern

In the ensuing discussion, we drop the reference to the subscript Q when referring to the world coordinates of the scene point (X_Q, Y_Q, Z_Q).

2.3 Imaging in a Parallel Stereo Setup, under Sinusoidal Illumination

The intensity $\mathsf{i}_\theta(x,y)$ of the $(x,y)^{th}$ camera pixel, can be expressed as the sum

$$\mathsf{i}_\theta(x,y) = \underbrace{\{\alpha_Z\ \mathsf{r}(x,y)\ \mathsf{p}_\theta(X,Y,Z)\} \otimes \mathsf{h}(x,y)}_{\textit{image under structured light}} + \underbrace{\{\mathsf{r}(x,y)\ \mathsf{a}(x,y)\} \otimes \mathsf{h}(x,y)}_{\textit{image under ambient light}}\ , \quad \begin{matrix} 0 \le x < N_c \\ 0 \le y < M_c \end{matrix} \tag{6}$$

- $\mathsf{i}_\theta(x,y)$: intensity of $(x,y)^{th}$ camera pixel
- $\mathsf{p}_\theta(X,Y,Z)$: incident intensity at (X,Y,Z) due to "Structured Light"
- $\mathsf{a}(x,y)$: ambient illumination
- $\mathsf{r}(x,y)$: detected intensity ÷ incident intensity (dependent on reflectance)
- $\mathsf{h}(x,y)$: space-invariant *psf* of the imaging system
- α_Z : normalization term that depends on the relative magnification between the camera and projector, in the horizontal and vertical directions.

Substituting *Eq.*[5] in *Eq.*[6], we arrive at the expression for the camera image, under the sinusoidal illumination $\mathsf{s}_\theta(x',y') = \frac{A}{2} + \frac{A}{2}\ sin\left(2\pi\left(\frac{\xi_0}{N_p}x' + \frac{\eta_0}{M_p}y'\right) + \theta\right)$

$$\mathsf{i}_\theta(x,y) = \left\{\mathsf{r}(x,y)\ \alpha_Z A_{Z,\xi_0,\eta_0} sin\left(\varphi(x,y) + \theta\right)\ +\ \mathsf{r}(x,y)[\alpha_Z\ A_{Z,0,0} + a(x,y)]\right\} \otimes h(x,y) \tag{7}$$

$$\varphi(x,y) \triangleq \begin{pmatrix} 2\pi\mu_Z\left(\mu_h\frac{\xi_0}{N_c}(x - c_x) + \mu_v\frac{\eta_0}{M_c}(y - c_y)\right) + 2\pi\frac{\xi_0}{N_p}c'_x + 2\pi\frac{\eta_0}{M_p}c'_y \\ +2\pi\frac{Z_p}{s_p}\left(\frac{\xi_0}{N_p}\frac{b_X}{Z\ \ b_Z} + \frac{\eta_0}{M_p}\frac{b_Y}{Z\ \ b_Z}\right) \end{pmatrix} \tag{8}$$

Please refer to *Table.*[1] for a definition of the terms in *Eq.*[7] and *Eq.*[8].

It is apparent from *Eq.*[7] that the illumination pattern experiences a depth-dependent distortion in the amplitude, frequency and phase. The methods for depth recovery discussed in this paper exploit this distortion, while the *OSR* method discussed in this paper tries to avoid the distortion.

Table 1. Definition of the parameters listed in the expression for the image captured by a camera under sinusoidal illumination (*Eq.*[7])

user defined parameters	
ξ_0, η_0	spatial frequency of illumination pattern in $\frac{\texttt{cycles}}{\texttt{image}}$
θ	phase of the illumination pattern
fixed parameters	***fixed for each camera pixel*** (x, y)
$M_c \times N_c$ (rows × columns)	size of camera image
$\mu_h = \frac{Z_p}{Z_c} \frac{s_c N_c}{s_p N_p}$	horizontal magnification
$\mu_v = \frac{Z_p}{Z_c} \frac{s_c M_c}{s_p M_p}$	vertical magnification
(c_x, c_y)	principal point of the camera
(c'_x, c'_y)	principal point of the projector
$\psi_0 = 2\pi \left(\frac{\xi_0}{N_p} c'_x + \frac{\eta_0}{M_p} c'_y \right)$	phase accumulated by the principal point of the projector
scene dependent parameters	***vary for each camera pixel*** (x, y)
$A_{Z,\xi,\eta}$	projector *MTF* at depth Z, and spatial frequency (ξ, η)
$\mu_Z = \frac{Z}{Z - b_Z}$	depth dependent magnification
$\psi_Z = 2\pi \frac{Z_p}{s_p} \left(\frac{\xi_0}{N_p} \frac{b_X}{Z - b_Z} + \frac{\eta_0}{M_p} \frac{b_Y}{Z - b_Z} \right)$	phase due to parallax
$\alpha_Z = \mu_Z^2 \; \mu_h \; \mu_v$	normalization term

2.4 Imaging in a Parallel Stereo Setup, under Periodic Illumination

The expression for $\mathsf{i}_\theta(x, y)$ derived in *Section.*(2.3) can be extended to accommodate periodic illumination patterns, by utilizing the Fourier series expansion of periodic signals. The resulting expression is used to reformulate the findings of Zhang *&* Nayar[19] in *Section.*(4.2).

For the sake of simplicity, we restrict our attention to odd-symmetric periodic illumination patterns (such as square waves[4], sawtooth waves[9]) with average value $\frac{A}{2}$, *i.e.*,

$$\mathsf{s}_\theta(x', y') = \frac{A}{2} + \frac{A}{2} \sum_{k=1}^{\infty} b_k \; sin \left(2\pi k \left(\frac{\xi_0}{N_p} x' + \frac{\eta_0}{M_p} y' \right) + k\theta \right), \quad \begin{array}{l} b_k : \textit{Fourier series coefficients} \\ b_k = \frac{4}{\pi k} mod(k, 2) \textit{ square wave} \\ b_k = -\frac{2}{\pi k} \textit{ sawtooth wave} \end{array} \tag{9}$$

Following the analysis of *Section.*(2.2), *Section.*(2.3) we find that the intensity of the $(x, y)^{th}$ camera pixel, under the illumination of *Eq.*[9], is given by

$$\mathsf{i}_\theta(x, y) = \left(\begin{array}{c} \left\{ \sum_{k=1}^{\infty} \alpha_Z \; \mathsf{r}(x, y) \; A_{Z,k\xi_0,k\eta_0} sin \left(k \; \varphi(x, y) + k \; \theta \right) \right\} \otimes h(x, y) \\ + \left\{ \mathsf{r}(x, y) [\alpha_Z \; A_{Z,0,0} + a(x, y)] \right\} \otimes h(x, y) \end{array} \right) \tag{10}$$

where $\varphi(x, y)$ is defined in *Eq.*[8], and $\{A_{Z,k\xi_0,k\eta_0}\}_{k=0}^{\infty}$ are real scalars that represent the effect of projector defocus on the spatial frequencies $\{(k\xi_0, k\eta_0)\}_{k=0}^{\infty}$, at depth Z.

With the aid of the proposed model for imaging under "Structured Light", we now examine *OSR* and depth estimation in detail.

3 Optical Super-Resolution Using Structured Light

The principle underlying "Structured Light"-*OSR* is amplitude-modulation. The idea is to shift object spatial frequencies outside the passband of the imaging system into the passband, by modulating the amplitude of a periodic pattern with scene information(*Fig.*[1(b)]). This is realized by projecting a series of phase shifted sinusoidal patterns onto the scene, as in the case of microscopy[13]. But, unlike microscopy, the difference in viewpoint between the camera *&* projector affects our ability to realize strict amplitude modulation, since it induces a depth dependent frequency *&* phase distortion in the observed sinusoidal pattern.

With the aid of the model proposed in *Section.*(2.3), we now show that the camera image represents a strictly amplitude modulated signal, when the scene dependent magnification μ_Z and phase ψ_Z, are invariant to depth.

Suppose the camera images $\{\mathsf{i}_\theta(x,y)\ :\ \theta = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$in a parallel stereo setup (*Eq.*[7]) are digitally recombined to obtain the following images

$$\underline{\mathsf{raw}(x,y):\ \text{bandlimited image of scene obtained under ambient + uniform projector illumination}}$$

$$\mathsf{raw}(x,y) \triangleq \frac{1}{4}\Big(\ \mathsf{i}_{\frac{\pi}{2}}(x,y) + \mathsf{i}_{\frac{3\pi}{2}}(x,y) + \mathsf{i}_0(x,y) + \mathsf{i}_\pi(x,y)\ \Big) = \big\{\big[\alpha_Z\ A_{Z,0,0} + a(x,y)\big]\,\mathsf{r}(x,y)\big\} \otimes \mathsf{h}(x,y)$$

$$\underline{\mathsf{m}^{\pm}(x,y):\ \text{exponentially modulated images of scene obtained under sinusoidal illumination}} \tag{11}$$

$$\begin{aligned}\mathsf{m}^{\pm}(x,y) &\triangleq \frac{1}{2}\Big(\ \mathsf{i}_{\frac{\pi}{2}}(x,y) - \mathsf{i}_{\frac{3\pi}{2}}(x,y)\ \Big) \pm \frac{\sqrt{-1}}{2}\big(\ \mathsf{i}_0(x,y) - \mathsf{i}_\pi(x,y)\ \big)\\ &= \Big\{\alpha_Z\ A_{Z,\xi_0,\eta_0}\ \mathsf{r}(x,y)\ e^{\pm j\left(2\pi\ \mu_Z\left(\mu_h \frac{\xi_0}{N_c}(x-c_x)+\mu_v \frac{\eta_0}{M_c}(y-c_y)\right)+\psi_0+\psi_Z\right)}\Big\} \otimes \mathsf{h}(x,y)\end{aligned} \tag{12}$$

In the special case that μ_Z, ψ_Z are invariant to scene depth ($\mu_Z = \mu, \psi_Z = \psi, \alpha_Z = \mu^2\mu_h\mu_v = \alpha$), we find that the expression for the modulated images $\mathsf{m}^{\pm}(x,y)$, reduces to strict amplitude modulation as shown below

$$\mathsf{m}^{\pm}(x,y) = \left(\underbrace{\alpha\ A_{Z,\xi_0,\eta_0}\ \mathsf{r}(x,y)\ e^{\pm j\varphi_0}}_{complex\ apmlitude}\ \underbrace{e^{\pm j\left(2\pi\ \mu\left(\mu_h \frac{\xi_0}{N_c}x+\mu_v \frac{\eta_0}{M_c}y\right)\right)}}_{modulating\ carrier}\right) \otimes \underbrace{\mathsf{h}(x,y)}_{low\ pass\ filter} \tag{13}$$

$$\varphi_0 = \psi_0 + \psi - 2\pi\mu\left(\mu_h \frac{\xi_0}{N_c} c_x + \mu_v \frac{\eta_0}{M_c} c_y\right) \tag{14}$$

Using the *modulation property* of the Fourier Transform[21], we identify the intermediate images shown below

$$\begin{aligned}&\overbrace{e^{\mp j\varphi_0}}^{phase\ compensation}\quad \overbrace{e^{\mp j\left(2\pi\ \mu\left(\mu_h \frac{\xi_0}{N_c}x+\mu_v \frac{\eta_0}{M_c}y\right)\right)}\ \mathsf{m}^{\pm}(x,y)}^{carrier\ demodulation}\\ &= \Big(\alpha A_{Z,\xi_0,\eta_0}\ \mathsf{r}(x,y)\Big) \otimes \underbrace{\Big\{\mathsf{h}(x,y)\ e^{\mp j\left(2\pi\ \mu\left(\mu_h \frac{\xi_0}{N_c}x+\mu_v \frac{\eta_0}{M_c}y\right)\right)}\Big\}}_{band\ pass\ filter\ centered\ at(\pm\mu_h\xi_0, \pm\mu_v\eta_0)}\end{aligned} \tag{15}$$

It is evident from *Eq.*[15] that the intermediate images contain spatial frequencies that exceed the bandwidth of the optical transfer function[3] $\mathcal{H}(\xi,\eta)$. This key insight permits us to realize "Structured Light"-*OSR* for macroscopic imaging systems.

The super-resolved result is obtained as the sum of the images

$$\mathsf{raw}(x,y) + \overbrace{e^{-j\varphi_0} e^{-j\left(2\pi\ \mu\left(\mu_h \frac{\xi_0}{N_c} x + \mu_v \frac{\eta_0}{M_c} y\right)\right)}\ \mathsf{m}^+(x,y)}^{phase\ compensation\ +\ demodulation\ in\ Fig.[4(a)]} + \overbrace{e^{j\varphi_0} e^{j\left(2\pi\ \mu\left(\mu_h \frac{\xi_0}{N_c} x + \mu_v \frac{\eta_0}{M_c} y\right)\right)}\ \mathsf{m}^-(x,y)}^{phase\ compenation\ +\ demodulation\ in\ Fig.[4(a)]}$$
$$= \mathsf{raw}(x,y) + \underbrace{\left\{\alpha A_{Z,\xi_0,\eta_0} \mathsf{r}(x,y)\right\} \otimes \left\{2\ \mathsf{h}(x,y)\ cos\left(2\pi\ \mu\left(\mu_h \frac{\xi_0}{N_c} x + \mu_v \frac{\eta_0}{M_c} y\right)\right)\right\}}_{band\ pass\ image\ in\ Fig.[4(a)]} \tag{16}$$

Fig.[5(a)] illustrates the proposed "Structured Light"-*OSR* workflow.

When are α_Z, μ_Z, ψ_Z invariant to scene geometry ?

Using the expressions for α_Z, μ_Z, ψ_Z (*Table.*[1]), we can prove that the only choice of baseline, and illumination pattern orientation that guarantees invariance with respect to 3D scene geometry, is given by

$$\begin{array}{ll} b_Z = 0 & \Rightarrow \mu_Z = 1, \alpha_Z = \mu_h \mu_v \\ \frac{\xi_0}{N_p} b_X + \frac{\eta_0}{M_p} b_Y = 0 & \Rightarrow \psi_Z = 0 \end{array} \tag{17}$$

Eq.[17] suggests that the epipoles in the camera and projector image planes are at infinity, and the illuminating pattern is aligned with the epipolar lines in each image. Examples of parallel stereo setups that satisfy *Eq.*[17] are listed below

Constraints Setup	Scene	Setup , Pattern	*OSR* direction
Horizontally Collocated Vertically Collocated Arbitrarily Collocated	no constraints no constraints no constraints	$b_Y = b_Z = 0, \eta_0 = 0$ $b_X = b_Z = 0, \xi_0 = 0$ *Eq.*[17]	$\updownarrow$ $\leftrightarrow$ $tan^{-1}\left(\frac{\xi_0}{N_p}\frac{M_p}{\eta_0}\right)$
Coincident	no constraints	$b_X = b_Y = b_Z = 0$ any (ξ_0, η_0)	$tan^{-1}\left(\frac{\xi_0}{N_p}\frac{M_p}{\eta_0}\right)$
Non-Collocated	planar facet parallel to camera *&* projector image planes	no constraints	$tan^{-1}\left(\frac{\xi_0}{N_p}\frac{M_p}{\eta_0}\right)$

Key components in proposed SI-OSR workflow of Fig.[5(a)]

1. **Identifying the demodulating frequency**
   In *Eq.*[13], the magnitude spectrum of the term $\alpha A_{Z,\xi_0,\eta_0} \mathsf{r}(x,y)$, corresponding to scene information, peaks at DC. Subsequent to modulation, the DC component of the scene information shifts to the carrier frequency $(\mu\mu_h\xi_0, \mu\mu_v\eta_0)$. We exploit this fact to identify the carrier frequency as the abscissa *&* ordinate of the largest value in the magnitude spectrum of $\mathsf{m}^+(x,y)$. For additional details, please refer to the supplementary material.

[3] Fourier Transform of the point-spread-function $\mathsf{h}(x,y)$ of the imaging system.

2. **Identifying the complex constant $e^{\pm j\varphi_0}$ for phase compensation**
   Since the DC component of the scene information is a real number, it must have zero phase before & after demodulation. We rely on this fact, to obtain an estimate of φ_0 from the phase of the DC value of the demodulated image, as $\widehat{\varphi_0} = \texttt{mean}\left[e^{-j\left(2\pi\ \mu\left(\mu_h \frac{\xi_0}{N_c}x+\mu_v \frac{\eta_0}{M_c}y\right)\right)}\mathsf{m}^{+}(x,y)\right]$
3. **Aliasing Management**
   The objective of demodulation is to restore the modulated spatial frequencies back to their rightful position. This is realized by shifting the modulated spectra, by the respective carrier frequencies $(\pm\mu\mu_h\xi_0, \pm\mu\mu_v\eta_0)$. Unfortunately, the circular nature of the frequency shift in the Discrete Fourier Transform[21], may cause some of the demodulated frequencies to wrap around the Nyquist frequency $\frac{1}{2s_c}$. The purpose of *Aliasing Management* is to avoid aliasing the demodulated spatial frequencies. Our approach to *Aliasing Management* involves an increase in the size of the modulated images $\mathsf{m}^{\pm}(x,y)$ by $(\mu\mu_h\xi_0, \mu\mu_v\eta_0)$ pixels in each direction. The increase is realized using sinc-interpolation : zero-padding the fourier spectrum followed by the Inverse Fourier Transform.

So far we have established that a collocated/coincident parallel stereo setup is sufficient to realize "Structured Light"-*OSR*. In an effort to jointly accomplish *OSR* and depth-estimation in a single setup, we now identify methods for recovering depth in a collocated/coincident parallel stereo setup.

4 Estimating Depth Using Structured Light

4.1 Un-calibrated Depth Estimation in a Collocated Setup

The following method belongs to the class of "Structured Light" methods called *Phase Measurement Profilometry* (*PMP*)[3, 22], which recover surface topology from the phase distortion induced by depth variation in a collocated stereo setup. The case of parallel stereo is of particular interest since it allows for uncalibrated depth estimation, as explained below. The workflow is summarized in *Fig.*[5(b)].

Plugging $b_Z = 0$ in *Eq.*[7], yields the following expression for the camera image under sinusoidal illumination, in a collocated parallel stereo setup,

$$\mathsf{i}_\theta(x,y) = \left\{\mathsf{r}(x,y)\ \mu_h\mu_v A_{Z,\xi_0,\eta_0} sin\left[\varphi(x,y)+\theta\right]\right\} + \left\{\mathsf{r}(x,y)\left[\mu_h\mu_v A_{Z,0,0} + a(x,y)\right]\right\} \tag{18}$$

$$\varphi(x,y) \triangleq \left(\begin{array}{l} \overbrace{2\pi\left(\mu_h\frac{\xi_0}{N_c}x+\mu_v\frac{\eta_0}{M_c}y\right)}^{linear\ phase\ due\ to\ carrier} + \overbrace{2\pi\frac{Z_p}{s_p}\left(\frac{\xi_0}{N_p}\frac{b_X}{Z}+\frac{\eta_0}{M_p}\frac{b_Y}{Z}\right)}^{\psi_Z\ :\ scene\ dependent\ phase} \\ \qquad + \underbrace{2\pi\frac{\xi_0}{N_p}c'_x + 2\pi\frac{\eta_0}{M_p}c'_y - 2\pi\mu_h\frac{\xi_0}{N_c}c_x - 2\pi\mu_v\frac{\eta_0}{M_c}c_y}_{\varphi_0\ :\ constant\ phase} \end{array}\right) \tag{19}$$

The effect of camera blur $h(x,y)$ in the expression for $\mathsf{i}_\theta(x,y)$ is disregarded, given our interest in identifying a "qualitative depth map". Careful examination of *Eq.*[18] reveals that the first term represents a phase-modulated image with carrier frequency $(\mu_h\xi_0, \mu_v\eta_0)\,\frac{\texttt{cycles}}{\texttt{image}}$, and depth-dependent phase variation (ψ_Z).

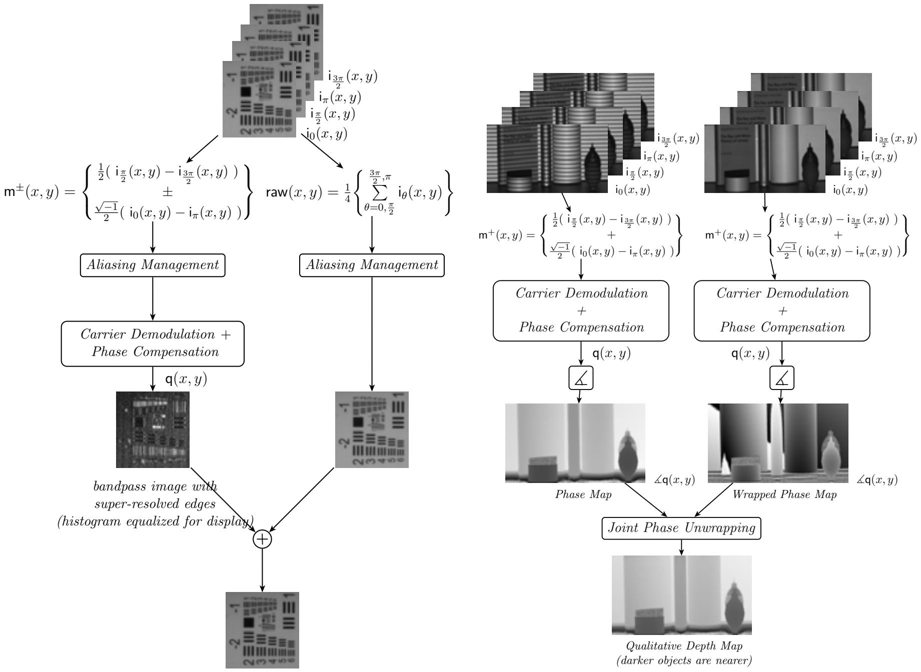
The objective of *PMP* is to recover the depth Z from the phase $\varphi(x,y)$. To this end, we recombine the camera images $\left\{\mathsf{i}_\theta(x,y)\ :\ \theta = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$ to obtain the following modulated image

$$\mathsf{m}^{+}(x,y) = \frac{1}{2}\Big(\ \mathsf{i}_{\frac{\pi}{2}}(x,y) - \mathsf{i}_{\frac{3\pi}{2}}(x,y)\ \Big) + \frac{\sqrt{-1}}{2}\big(\ \mathsf{i}_0(x,y) - \mathsf{i}_\pi(x,y)\ \big) \tag{20}$$

$$= \mu_h\mu_v A_{Z,\xi_0,\eta_0}\ \mathsf{r}(x,y)\ \underbrace{e^{j2\pi\left(\mu_h\frac{\xi_0}{N_c}x+\mu_v\frac{\eta_0}{M_c}y\right)}}_{modulating\ carrier}\quad \underbrace{e^{j\varphi_0}}_{complex\ constant}\quad \underbrace{e^{j\psi_Z}}_{scene-dependent\ phase} \tag{21}$$

$$\text{where}\quad \psi_Z = 2\pi\frac{Z_p}{s_p}\left(\frac{\xi_0}{N_p}\frac{b_X}{Z} + \frac{\eta_0}{M_p}\frac{b_Y}{Z}\right),\ \varphi_0 = 2\pi\left(-\mu_h\frac{\xi_0}{N_c}c_x - \mu_v\frac{\eta_0}{M_c}c_y + \frac{\xi_0}{N_p}c'_x + \frac{\eta_0}{M_p}c'_y\right) \tag{22}$$

The process of depth-estimation begins with the identification of the carrier frequency $(\mu_h\ \xi_0, \mu_v\eta_0)$, and the complex constant $e^{\pm j\varphi_0}$. We rely on a strategy similar to that described in *Section.*(3), for this purpose. Additional details are available in the supplementary material.


$\mathsf{i}_{\frac{3\pi}{2}}(x,y)$
$\mathsf{i}_\pi(x,y)$
$\mathsf{i}_{\frac{\pi}{2}}(x,y)$
$\mathsf{i}_0(x,y)$
$\mathsf{m}^{\pm}(x,y) = \left\{ \frac{1}{2}(\ \mathsf{i}_{\frac{\pi}{2}}(x,y) - \mathsf{i}_{\frac{3\pi}{2}}(x,y)\ ) \pm \frac{\sqrt{-1}}{2}(\ \mathsf{i}_0(x,y) - \mathsf{i}_\pi(x,y)\ ) \right\}$
$\mathsf{raw}(x,y) = \frac{1}{4}\left\{\sum_{\theta=0,\frac{\pi}{2}}^{\frac{3\pi}{2},\pi} \mathsf{i}_\theta(x,y)\right\}$
Aliasing Management
Aliasing Management
Carrier Demodulation + Phase Compensation
$\mathsf{q}(x,y)$
bandpass image with super-resolved edges (histogram equalized for display)
$\mathsf{m}^{+}(x,y) = \left\{ \frac{1}{2}(\ \mathsf{i}_{\frac{\pi}{2}}(x,y) - \mathsf{i}_{\frac{3\pi}{2}}(x,y)\ ) + \frac{\sqrt{-1}}{2}(\ \mathsf{i}_0(x,y) - \mathsf{i}_\pi(x,y)\ ) \right\}$
Carrier Demodulation + Phase Compensation
$\mathsf{q}(x,y)$
$\measuredangle$
Phase Map
$\measuredangle\mathsf{q}(x,y)$
Wrapped Phase Map
$\measuredangle\mathsf{q}(x,y)$
Joint Phase Unwrapping
Qualitative Depth Map (darker objects are nearer)

(a) Un-calibrated "Structured Light" *OSR* (images are from actual experiments)

(b) Un-calibrated "Structured Light" Depth Estimation (images are from actual experiments)

Fig. 4. Workflow for un-calibrated *OSR* and depth estimation using "Structured Light". (Zoom into the images at 1200% for a better view)

A qualitative depth map of the scene can be recovered from the instantaneous phase of the image $\mathsf{q}(x,y)$defined below

$$\mathsf{q}(x,y) \triangleq \underbrace{e^{-j\varphi_0}}_{phase\ compensation} \underbrace{e^{-j2\pi\left(\mu_h \frac{\xi_0}{N_c}x+\mu_v\frac{\eta_0}{M_c}y\right)}\ \mathsf{m}^{+}(x,y)}_{demodulation} \tag{23}$$

$$Z \propto \frac{1}{\texttt{unwrap}\left(\measuredangle\{\mathsf{q}(x,y)\}\right)} \quad , \quad \measuredangle\{\mathsf{q}(x,y)\} = tan^{-1}\left(\frac{\texttt{imag}\left(\mathsf{q}(x,y)\right)}{\texttt{real}\left(\mathsf{q}(x,y)\right)}\right) \tag{24}$$

Although our analysis suggests that four images are sufficient to recover the depth map, four or more additional images with different modulating frequency maybe required to unwrap $\measuredangle\{\mathsf{q}(x,y)\}$ unambiguously[23]. This should explain the presence of two set of images in the workflow of *Fig.*[5(b)].

A known limitation of this method is that depth information may not be available at every pixel, due to cast shadows. This can be overcome by coinciding the "center of perspective" of the camera and projector using a beam splitter ($b_X = b_Y = b_Z = 0$). But, we know from *Section.*(3) that the illumination pattern is not subject to depth dependent distortion, in a coincident camera+projector setup. Although this is desirable from the standpoint of *OSR*, it is not desirable from the standpoint of depth estimation. In [19], Zhang *&* Nayar presented a clever solution to the problem.

We now attempt to reformulate their findings using the model proposed in *Section.*(2.4), and suggest possible extensions. This effort highlights the scope of the model.

4.2 Calibrated Depth Estimation in a Coincident Setup

The following method due to Zhang[19], relies on the depth-dependent blurring induced by the limited depth of field of the projector, to recover the depth map of a scene. The setup involves a camera observing the scene under a temporally varying illumination $\mathsf{s}_\theta(x',y',t)$, produced by a projector that is coincident with the camera. The coincident geometry($b_X = b_Y = b_Z = 0$) results in zero disparity ($x = \mu_h x', y = \mu_v y'$) and guarantees that depth information is available for each camera pixel.

The temporally varying illumination pattern $\mathsf{s}_\theta(x',y',t)$ is obtained by shifting a vertical periodic pattern ($\eta_0 = 0$) with a wide range of frequencies, i.e.,

$$\mathsf{s}_\theta(x',y',t) = \frac{A}{2} + \frac{A}{2}\sum_{k=1}^{\infty} b_k\ sin\left(2\pi k\left(\frac{\xi_0}{N_p}(x'-t)\right)+k\theta\right) \tag{25}$$

Substituting $x' = x' - t$ in *Eq.*[4], and $b_X = b_Y = b_Z = 0$, $\eta_0 = 0$ into the analysis of *Section.*(2.4), yields[4] the following expression for the temporal intensity of the $(x,y)^{th}$ camera pixel

[4] $(b_X = b_Y = b_Z = 0) \Rightarrow \alpha_Z = \mu_h\mu_v, \mu_Z = 1, \psi_Z = 0.$

$$
\mathsf{i}_\theta(x, y, t) = \left(\begin{array}{l} -\sum_{k=1}^{\infty} \overbrace{\mathsf{r}(x,y)\ \mu_h \mu_v\ A_{Z,k\xi_0,k\eta_0}}^{amplitude\ along\ t} sin \left(2\pi \overbrace{\left(k \frac{\xi_0}{N_p} \right)}^{frequency} t - \overbrace{2\pi k \left(\mu_h \frac{\xi_0}{N_c}(x - c_x) + \frac{\xi_0}{N_p} c'_x \right) + k\theta}^{fixed\ phase\ offset\ along\ t} \right) \\ \quad + \quad \underbrace{\mathsf{r}(x,y)\ [\mu_h \mu_v\ A_{Z,0,0} + a(x,y)]}_{DC\ value\ along\ t} \end{array} \right) \tag{26}
$$

Careful examination of the expression for $\mathsf{i}_\theta(x, y, t)$ reveals that it is the infinite sum of blurred sinusoids, whose fourier coefficients $\{A_{Z,k\xi_0,k\eta_0}\}_{k=0}^{\infty}$ depend on depth Z. In order to ensure that the scalars $\{A_{Z,k\xi_0,k\eta_0}\}_{k=0}^{\infty}$ decrease monotonically with depth, the projector is focused on the farthest object in the scene. Zhang *&* Nayar observed that the ratio $\frac{A_{Z,2\xi_0,2\eta_0}}{A_{Z,\xi_0,\eta_0}}$ is sufficient to recover high quality depth maps. A tilted planar target with known depth variation is used to create a lookup table mapping the above ratio to the scene depth Z. This lookup table is used in conjunction with the Discrete Fourier Series coefficients of $i_\theta(x, y, t)$, to recover the depth map of a scene.

With a little effort it can be shown that *Eq.*[26] also applies to a vertically collocated setup ($b_X = 0$), and a vertical illumination pattern ($\eta_0 = 0$). This insight suggests that Zhang *&* Nayar's method also applies to collocated setups, barring cast shadows.

5 Experimental Results

The experimental setup described in *Fig.*[**??**] is used to validate the proposed model for image formation under "Structured Light". In particular, we verify the ability to realize un-calibrated *OSR &* depth-estimation in a vertically collocated camera+projector setup.

The entrance apertures of the projector and camera are visually aligned, to realize vertical collocation. Also, the projector is focused on a plane just behind the scene volume. Lastly, the aperture of the camera is stopped down to ensure that

- the imaging system is limited by diffraction and not by aliasing
- the camera *psf* does not change appreciably within the projector depth of field

The results of the experiment are shown in *Fig.*[6]. A visual comparison of *Fig.*[6(a)] *& Fig.*[6(c)] reveals clear improvement in spatial resolution due to modulation by a horizontal periodic pattern. The improvement is apparent in the bar-code pattern and the USAF Resolution target. The increase in spatial resolution is accompanied by an increase in the camera Nyquist frequency, because of aliasing-management.

A qualitative depth map of the scene (*Fig.*[6(f)]) is obtained by analyzing the phase distortion experienced by vertically periodic illumination patterns. The depth map correctly identifies the shape of the cylindrical poster tube, and the tilted cardboard carton.

(a) "Structured Light" *OSR* (images are from actual experiments)

(b) "Structured Light" Depth Estimation (images are from actual experiments)

Fig. 5. Workflow for un-calibrated *OSR* and depth estimation using "Structured Light". (Zoom into the images at 1200% for a better view)

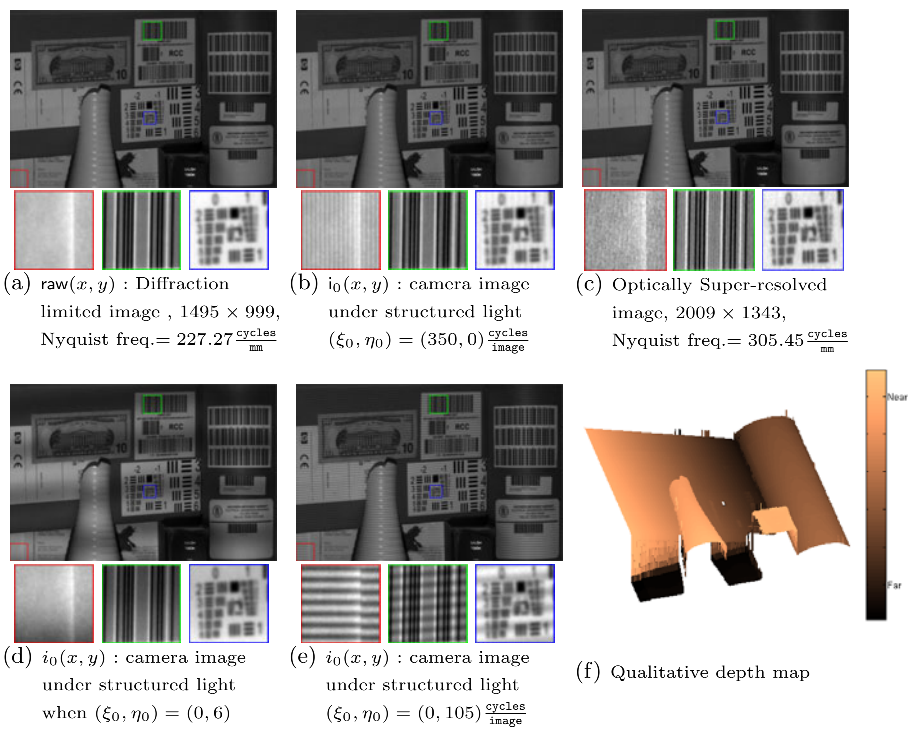
Additional experimental evidence is available in the flow diagram of *Fig.*[5], and the supplementary material.

6 Summary

Depth-estimation and Optical Super-Resolution are popular applications for "Structured Light" in computer vision and optics. Till date, they have been treated as separate problems, with no known method for surpassing the diffraction limit of a macroscopic imaging system. The mathematical framework for imaging under "Structured Light" proposed in *Section.*([2]),

- reveals a method for realizing *OSR* in macroscopic imaging systems
- unifies the two "Structured Light" embodiments, when the image planes of the imaging and illuminating systems are parallel
- reveals a variety of setups for jointly realizing *OSR* and depth-estimation. Select cases are summarized below

Setup		***OSR***	**Depth Estimation**
Horizontally Collocated	$b_Y = b_Z = 0$	$\xi_0 = 0$, un-calibrated	$\eta_0 = 0$, un-calibrated
Vertically Collocated	$b_X = b_Z = 0$	$\eta_0 = 0$, un-calibrated	$\xi_0 = 0$, un-calibrated
Coincident	$b_X = b_Y = b_Z = 0$	un-calibrated	calibrated



(a) raw(x, y) : Diffraction limited image , 1495 × 999, Nyquist freq.= $227.27 \frac{\texttt{cycles}}{\texttt{mm}}$

(b) $\mathrm{i}_0(x, y)$: camera image under structured light $(\xi_0, \eta_0) = (350, 0) \frac{\texttt{cycles}}{\texttt{image}}$

(c) Optically Super-resolved image, 2009 × 1343, Nyquist freq.= $305.45 \frac{\texttt{cycles}}{\texttt{mm}}$

(d) $i_0(x, y)$: camera image under structured light when $(\xi_0, \eta_0) = (0, 6)$

(e) $i_0(x, y)$: camera image under structured light $(\xi_0, \eta_0) = (0, 105) \frac{\texttt{cycles}}{\texttt{image}}$

(f) Qualitative depth map

Fig. 6. Experimentally realizing “Structured Light”-*OSR* and depth estimation in a vertically collocated camera+projector setup with parallel image planes. (Zoom into the images at 300% for a better view).

We are currently examining methods to reduce the number of illuminating patterns, and extend the proposed mathematical model to crossed optical axes geometry. The insight gained from *Eq.*[17] suggests that it should be possible to realize *OSR* in a collocated setup ($b_Z = 0$) with crossed optical axes, when the projected pattern appears undistorted from the camera viewpoint.

References

[1] Meadows, D.M., Johnson, W., Allen, J.B.: Generation of surface contours by moiré patterns. Appl. Opt. 9, 942–947 (1970)

[2] Takeda, M., Mutoh, K.: Fourier transform profilometry for the automatic measurement of 3-d object shapes. Appl. Opt. 22, 3977–3982 (1983)

[3] Srinivasan, V., Liu, H., Halioua, M.: Automated phase-measuring profilometry: a phase mapping approach. Applied Optics 24, 185–188 (1985)

[4] Posdamer, J.L., Altschuler, M.D.: Surface measurement by space-encoded projected beam systems. Computer Graphics and Image Processing 18, 1–17 (1982)

[5] Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In: IEEE Conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 195–202 (2003)

[6] Hermans, C., Francken, Y., Cuypers, T., Bekaert, P.: Depth from sliding projections. In: IEEE Conf. on Computer Vision & Pattern Recognition, pp. 1865–1872 (2009)

[7] Besl, P.: Active, optical range imaging sensors. Machine vision and applications 1, 127–152 (1988)

[8] Jarvis, R.: A perspective on range finding techniques for computer vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 5, 122–139 (1983)

[9] Batlle, J., Mouaddib, E., Salvi, J.: Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. Pattern Recognition 31, 977 (1998)

[10] Chen, F., Brown, G., Song, M.: Overview of three-dimensional shape measurement using optical methods. Optical Engineering 39, 10 (2000)

[11] Chen, S., Li, Y., Zhang, J., Wang, W.: Active sensor planning for multiview vision tasks. Springer, Heidelberg (2008)

[12] Heintzmann, R., Cremer, C.: Laterally modulated excitation microscopy: improvement of resolution by using a diffraction grating. In: Proceedings of SPIE, vol. 3568, p. 185 (1999)

[13] Gustafsson, M.: Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. Journal of Microscopy 198, 82–87 (2000)

[14] Zalevsky, Z., Mendlovic, D.: Optical superresolution. Springer, Heidelberg (2003)

[15] Mico, V., Zalevsky, Z., Garcia, J.: Optical Superresolution - Imaging Beyond Abbe's Diffraction Limit. Journal of Holography and Speckle 5, 110–123 (2009)

[16] Martnez-Corral, M., Saavedra, G.: The Resolution Challenge in 3D Optical Microscopy. Progress in Optics, vol. 53, pp. 1–67 (2009)

[17] Goodman, J.: Introduction to Fourier optics. Roberts & Company Publishers (2005)

[18] Lukosz, W., Marchand, M.: Optischen Abbildung unter berschreitung der beugungsbedingten Auflsungsgrenze. Opt. Acta 10, 241–255 (1963)

[19] Zhang, L., Nayar, S.: Projection defocus analysis for scene capture and image display. ACM Transactions on Graphics (TOG) 25, 915 (2006)

[20] Vaquero, D.A., Raskar, R., Feris, R.S., Turk, M.: A projector-camera setup for geometry-invariant frequency demultiplexing. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2009), Miami, Florida (2009)

[21] Bracewell, R.: The Fourier Transform and its Applications (2000)

[22] Halioua, M., Liu, H.: Optical three-dimensional sensing by phase measuring profilometry. Optics and lasers in engineering 11, 185–215 (1989)

[23] Wang, Z., Du, H., Park, S., Xie, H.: Three-dimensional shape measurement with a fast and accurate approach. Applied Optics 48, 1052–1061 (2009)

Lighting and Pose Robust Face Sketch Synthesis

Wei Zhang[1], Xiaogang Wang[2], and Xiaoou Tang[1,3]

[1] Department of Information Engineering, The Chinese University of Hong Kong
{zw009,xtang}@ie.cuhk.edu.hk

[2] Department of Electronic Engineering, The Chinese University of Hong Kong
xgwang@ee.cuhk.edu.hk

[3] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

Abstract. Automatic face sketch synthesis has important applications in law enforcement and digital entertainment. Although great progress has been made in recent years, previous methods only work under well controlled conditions and often fail when there are variations of lighting and pose. In this paper, we propose a robust algorithm for synthesizing a face sketch from a face photo taken under a different lighting condition and in a different pose than the training set. It synthesizes local sketch patches using a multiscale Markov Random Field (MRF) model. The robustness to lighting and pose variations is achieved in three steps. Firstly, shape priors specific to facial components are introduced to reduce artifacts and distortions caused by variations of lighting and pose. Secondly, new patch descriptors and metrics which are more robust to lighting variations are used to find candidates of sketch patches given a photo patch. Lastly, a smoothing term measuring both intensity compatibility and gradient compatibility is used to match neighboring sketch patches on the MRF network more effectively. The proposed approach significantly improves the performance of the state-of-the-art method. Its effectiveness is shown through experiments on the CUHK face sketch database and celebrity photos collected from the web.

1 Introduction

Automatic face sketch synthesis has drawn a great deal of attention in recent years [1][2][3][4][5] due to its applications in law enforcement and digital entertainment. For example, in law enforcement, it is useful to develop a system to search photos from police mug-shot databases using a sketch drawing when the photo of a suspect is not available. By transferring face photos to sketches, inter-modality face recognition is made possible [2]. In the movie industry, artists can save a great amount of time on drawing cartoon faces with the assistance of an automatic sketch synthesis system. Such a system also provides an easy tool for people to personalize their identities in the digital world, such as through the MSN avatar.

Computer-based face sketch synthesis is different from line drawing generation [7][8]. Line drawings without texture are less expressive than sketches with both contours and shading textures. Popular sketch synthesis methods are mostly

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 420–433, 2010.

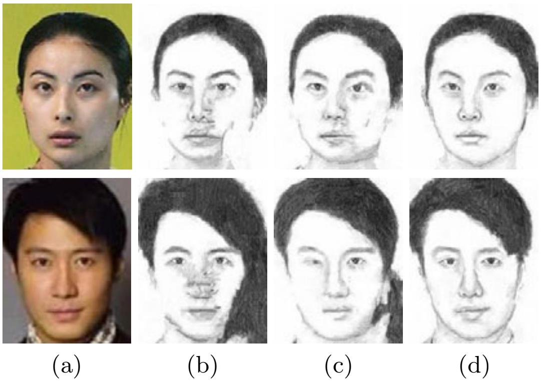


Fig. 1. Examples of synthesized sketches from web face photos. (a) Test photos; (b) Sketches synthesized by [5]; (c) Sketches synthesized by [5] with luminance remapping [6]; (d) Sketches synthesized by our method. Note that luminance remapping refers to zero-mean unit-variance normalization of the luminance channel of all photos in our implementation. This simple technique was found to be better than non-smooth mappings in image style transformation, such as histogram matching/equalization [6]. The results are best viewed on screen.

example-based, which generates a sketch with rich textures from an input face photo based on a set of training face photo-sketch pairs [1][3][4][5]. These approaches can synthesize sketches of different styles by choosing training sets of different styles. Tang and Wang [1] proposed to apply the eigentransform globally to synthesize a sketch from a photo. However, such a global linear model does not work well if the hair region is included, as the hair styles vary significantly among different people. To overcome this limitation, Liu *et al.* [3] proposed patch-based reconstruction. The drawback of this approach is that the patches are synthesized independently, ignoring their spatial relationships, such that some face structures cannot be well synthesized. In addition, face sketch synthesis through linear combinations of training sketch patches causes the blurring effect.

Following this line of work, a state-of-the-art approach using a multiscale Markov random field (MRF) model has been proposed recently [5] and achieved good performance under well controlled conditions (i.e. the testing face photo has to be taken in the frontal pose and under a similar lighting condition as the training set). This approach has some attractive features: (1) it can well synthesize complicated face structures, such as hair, which are difficult for previous methods [1]; (2) it significantly reduces artifacts, such as the blurring and aliasing effects, which commonly exist in the results of previous methods [1][3]. In spite of the great improvement compared with previous methods, this approach often fails if the testing face photo is taken in a different pose or under a different lighting condition (even if the lighting change is not dramatic) than the training set. Some examples are shown in Fig. 1. Due to the variations of lighting and pose, on the synthesized sketches by [5] some face structures are lost, some dark regions are synthesized as hair, and there are a great deal of distortions

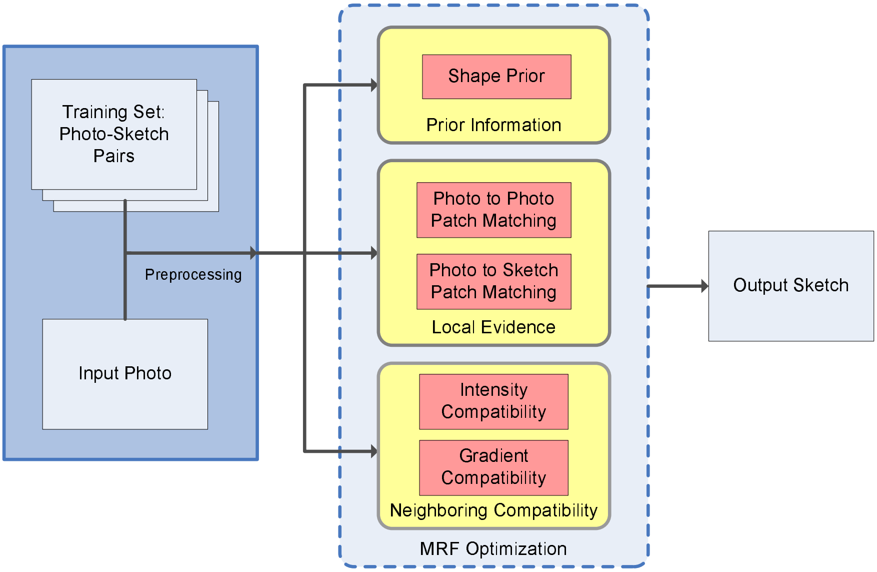


Fig. 2. Illustration of our framework

and artifacts. This is also a serious problem not addressed by other approaches [1][3][4]. It limits their applications to real-world problems.

In face recognition studies, some preprocessing techniques such as histogram equalization, and features such as Local Binary Patterns (LBP) [9], were used to effectively recognize face photos under lighting variations. In the area of nonphotorealistic rendering, luminance remapping was introduced to normalize lighting variations [6]. However, experiments show that simply borrowing these techniques is not effective in face sketch synthesis. See examples in Fig. 1.

In this paper, we address this challenge: *given a limited set of photo-sketch pairs with frontal faces and normal lighting conditions, how to synthesize face sketches for photos with faces in different poses (in the range of* $[-45^o +45^o]$*) and under different lighting conditions.* We adopt the multiscale MRF model whose effectiveness has been shown in face sketch synthesis [5] and many low-level vision problems [10]. In order to achieve the robustness to variations of lighting and pose, some important improvements are made in the design of the MRF model as summarized in Fig. 2. Firstly, a new term of shape priors specific to face components are introduced in our MRF model. It effectively reduces distortions and artifacts and restores lost structures as shown in Fig. 1. Secondly, patch descriptors and metrics which are more robust to lighting variations are used to find candidates of sketch patches given a photo patch. In addition to photo-to-photo patch matching, which was commonly used in previous approaches [3][5], our "local evidence" term also includes photo-to-sketch patch matching, which improves the matching accuracy with the existence of lighting and pose variations. Lastly, a smoothing term involving both intensity compatibility and gradient compatibility is used to match neighboring sketch patches on the MRF network more effectively.

The effectiveness of our approach is evaluated on the CUHK face sketch database which includes face photos with different lightings and poses. We also test on face photos of Chinese celebrities downloaded from the web. The experimental results show that our approach significantly improves the performance of face sketch synthesis compared with the state-of-the-art method [5] when the testing photo includes lighting or pose variations.

2 Lighting and Pose Robust Face Sketch Synthesis

In this section, we present our algorithm for face sketch synthesis. For ease of understanding, we use the single-scale MRF model in the presentation, instead of the two-scale MRF model in our implementation[1].

2.1 Overview of the Method

A graphical illustration of the MRF model is shown in Fig. 3. A test photo is divided into N overlapping patches with equal spacing. Then a MRF network is built. Each test photo patch x_i^p is a node on the network. Our goal is to estimate the status $y_i = (y_i^p, y_i^s)$, which is a pair of photo patch and sketch patch found in the training set, for each x_i^p. Photos and sketches in the training set are geometrically aligned. y_i^p is a photo patch and y_i^s is its corresponding sketch patch. If patches i and j are neighbors on the test photo, nodes y_i and y_j are connected by an edge, which enforces a compatibility constraint. The sketch of the test photo is synthesized by stitching the estimated sketch patches $\{y_i^s\}$. Based on the MRF model, our energy function is defined in the following form,

$$E(\{y_i\}_{i=1}^N) = \sum_{i=1}^{N} E_L(x_i^p, y_i) + \sum_{i=1}^{N} E_{Pi}(y_i) + \sum_{(i,j)\in \Xi} E_C(y_i^s, y_j^s), \tag{1}$$

where Ξ is the set of pairs of neighboring patches, $E_L(x_i^p, y_i)$ is the local evidence function (Subsection 2.2), $E_{Pi}(y_i)$ is the shape prior function (Subsection 2.3), and $E_C(y_i^s, y_j^s)$ is the neighboring compatibility function (Subsection 2.4). The shape prior function is specific to face components, which means that different location indicated by i has different E_{Pi}. The above MRF optimization problem can be solved by belief propagation [10] [11].

A MRF model was also used in [5], however, with several major differences with ours. It has no shape prior function which is effective in sketch synthesis. Its local evidence function only computes the sum of the squared differences (SSD) between x_i^p and y_i^p and is sensitive to lighting variations. Our local evidence function uses new patch descriptors which are more robust to lighting variations.

[1] We do find that the two-scale MRF model performs better. The details of multiscale MRF can be found in [5]. However, it is not the focus of this paper.

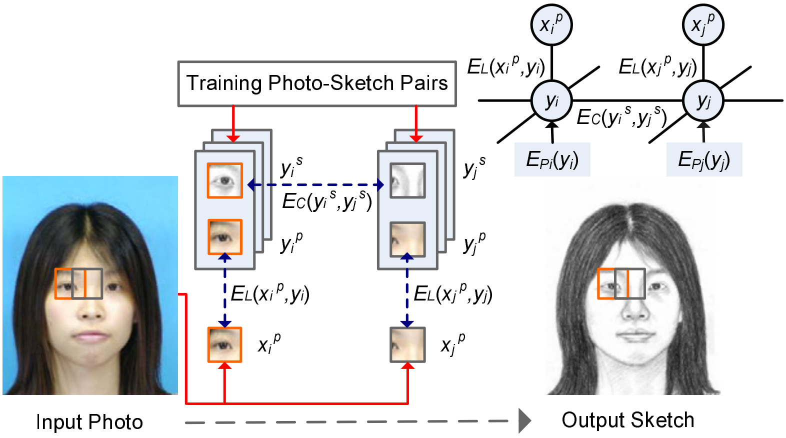


Fig. 3. Illustration of the MRF model for face sketch synthesis

Our method includes not only photo-to-photo patch matching (between x_i^p and y_i^p) but also photo-to-sketch patch matching (between x_i^p and y_i^s) to improve the robustness. The neighboring compatibility function in [5] is to minimize SSD between neighboring estimated sketch patches (y_i^s and y_j^s) in their overlapping region, while ours also minimizes the difference of gradient distributions. Details will be explained in the following subsections.

2.2 Local Evidence

The goal of the local evidence function is to find a sketch patch y_i^s in the training set best matching the photo patch x_i^p in test. However, since photos and sketches are in different modalities, it is unreliable to directly match them. So the training photo patch y_i^p corresponding to a training sketch patch y_i^s is involved. It is assumed that if y_i^p is similar to x_i^p, it is likely for y_i^s to be a good estimation of the sketch patch to be synthesized. We propose to match a testing photo patch with training photo patches and also with training sketch patches simultaneously, i.e. we define the local evidence function as the weighted sum of squared intra-modality distance d_{L1}^2 and squared inter-modality distance d_{L2}^2,

$$E_L(x_i^p, y_i) = d_{L1}^2(x_i^p, y_i^p) + \lambda_{L2} d_{L2}^2(x_i^p, y_i^s), \quad (2)$$

where λ_{L2} is the weight to balance different terms in the energy function E and it is chosen as 2 in our experiments.

Photo-to-Photo Patch Matching. A straightforward choice of E_L is the Euclidean distance between x_i^p and y_i^p as used in [5]. However, it does not perform well when the lighting condition varies. Noticing that most of the sketch contours correspond to edges in the photo, we use a difference-of-Gaussians (DoG) filter to process each photo, i.e. convolving each photo with the difference of two Gaussian

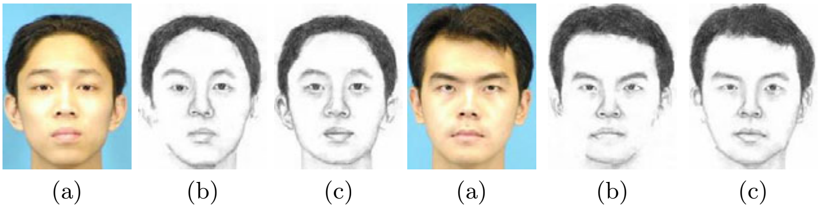


Fig. 4. Compare the results with/without DoG filtering under a normal lighting condition. (a) Test photos which are under the same lighting as the training set. (b)Synthesized sketch by the method in [5] without DoG filtering. (c) Synthesized sketches by our method with DoG filtering. To evaluate the effectiveness of DoG filtering, other parts, such as shape priors and photo-to-sketch patch matching, in our framework are not used in these examples.

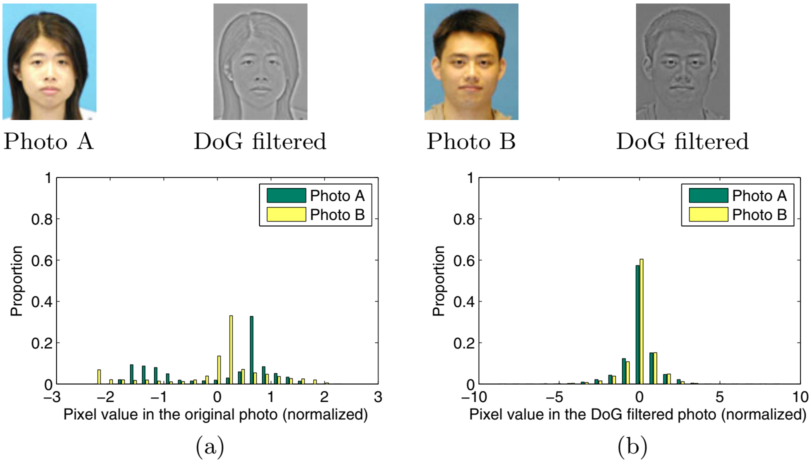


Fig. 5. Examples of DoG filtering with $(\sigma_0, \sigma_1) = (0, 4)$. Photo A is from the training set taken under the normal lighting condition, and Photo B is from the testing set taken under a different lighting condition. The pixel values of DoG filtered photos are scaled to $[0, 1]$ for visualization. (a) Histograms of pixel values of the two photos after luminance remapping. They do not match well. (b) Histograms of pixel values of the two photos after DoG filtering and normalization. They match well.

kernels with standard deviations σ_0 and σ_1, and normalize all pixel values to zero-mean and unit-variance. In our experiments, we find that $(\sigma_0, \sigma_1) = (0, 4)$ or $(1, 4)$ performs the best. DoG filtering has two advantages. First, it can detect and enhance the edges, and thus the synthesized sketch has better facial details. As shown in Fig. 4, even for normal lighting, the DoG filtering can improve facial details. Second, subtracting low-frequency component reduces the effect of lighting variations, e.g. shading effects. The example in Fig. 6 shows that DoG filtering improves synthesized facial details, especially on the nose and the

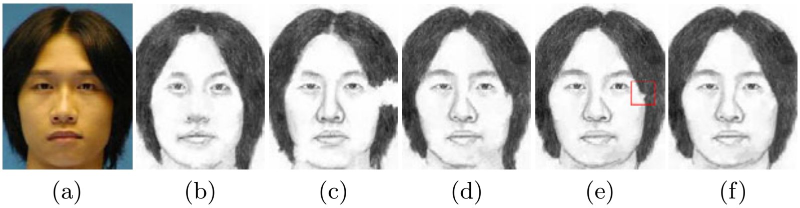


Fig. 6. Sequential illustration of the roles of each part in our framework. (a) Test photo under a different lighting condition than the training set; (b) Sketch by the method in [5] with luminance remapping as preprocessing [6]; (c) Sketch by our method with P2P+IC; (d) Sketch by our method with P2P+P2S+IC; (e) Sketch by our method with P2P+P2S+prior+IC; (f) Sketch by our method with P2P+P2S+prior+IC+GC. P2P, P2S, prior, IC and GC represent photo-to-photo patch matching, photo-to-sketch patch matching, shape priors, intensity compatibility and gradient compatibility, respectively. The results are best viewed on screen.

eyebrows, when there are lighting variations. Luminance remapping [6], which normalizes the distribution of pixel values in an image to zero-mean and unit-variance, is commonly used for lighting normalization. However, its improvement is limited in this application. An example is shown in Fig. 5. After luminance remapping, the distributions of pixel values in two photos taken under different lighting conditions still do not match. On the contrary, their distributions after DoG filtering match well. In some cases, photo-to-photo patch matching is not enough and the mismatching problem, such as the hair and profile regions shown in Fig. 6 (c), still exists. Thus, photo-to-sketch patch matching is introduced.

Photo-to-Sketch Patch Matching. The intra-modality distance between photo patches does not always work for selecting a good sketch patch. Similar photo patches under the Euclidean distance may correspond to very different sketch patches. Interestingly, people have the ability to directly match photos with sketches. Inspired by this, we propose to use inter-modality distance between testing photo patches and training sketch patches to enhance the selection ability. As the visual appearances of photo and sketch patches are different, it is difficult to directly match them. However, there exists some similarity of gradient orientations between a photo and its sketch. We choose to use the dense SIFT descriptor [12] from the family of histogram-of-orientations descriptors. Our strategy is to assign each patch a dense SIFT descriptor, and use the Euclidean distance between SIFT descriptors of photo patches and sketch patches as the inter-modality distance. To capture structures in large scales, we extract the descriptors in larger regions than patches. For each patch, we extract a region of size 36×36 centered at the center of the patch (the size of patch is 10×10), and divide it into 4×4 spatial bins of the same size. 8 orientations bins are evenly spaced over 0°-360°. The vote of a pixel to the histogram is weighted by its gradient magnitude and a Gaussian window with parameter $\sigma = 6$ centered at the center of the patch. So the descriptor is 128 dimensional. The descriptor is normalized by its $L2 - norm$, clipped by a threshold 0.2 and renormalized as reported in [12]. The synthesis result with photo-to-sketch patch matching is shown in Fig. 6 (d). It restores the hair and partial profile lost in Fig. 6 (c).

2.3 Shape Prior

Face images are a special class of images with well regularized structures. Thus shape priors on different face components can be used to effectively improve the synthesis performance. The loss of some face structures, especially the face profile, is a common problem for the patch-based sketch synthesis methods without referring to global structures. When this happens, the contours of some face components are replaced by blank regions. This problem becomes much more serious when there are variations of lighting and pose. See examples in Fig. 1. However, it can be effectively alleviated by using the prior information on different face components to guide the selection of sketch patches. In our approach, a state-of-the-art face alignment algorithm [13] is first utilized to detect some predefined landmarks on both the training sketches and the testing photo. The chosen landmarks locate in regions where loss of structures often happens, especially on the face profile. Shape priors are imposed to these regions but not in other regions. If a landmark f falls into patch i on the test photo, a prior distribution is computed via kernel density estimation,

$$E_{Pi}(y_i) = \lambda_P \ln \left[\frac{1}{\sqrt{2\pi} N_t} \sum_{k=1}^{N_t} \exp \left(- \frac{(\beta(y_i^s) - \beta_{k,f})^2}{h_f^2} \right) \right]. \tag{3}$$

N_t is the number of sketches in the training set. $\beta(y_i^s)$ is some statistic on the sketch patch y_i^s. $\beta_{k,f}$ is the statistic on a sketch patch centered at landmark f in sketch image k. h_f is the bandwidth of landmark f and is set as three times of the standard deviation of $\{\beta_{k,f}\}$. The weight $\lambda_P = 0.01$ is to normalize the metric scale of the shape prior term and the performance of our algorithm is robust to λ_P in a fairly large range.

We test several kinds of patch statistics, such as mean gradient magnitude, variance of pixel values, proportion of edge pixels, and find that mean gradient magnitude performs the best and it is chosen as $\beta(\cdot)$. It can well solve the problem of losing structures, as shown in Fig. 6 (e).

2.4 Neighboring Compatibility

The goal of the neighboring compatibility function is to make the neighboring estimated sketch patches smooth and thus to reduce the artifacts on the synthesized sketch. In our model it is defined as

$$E_C(y_i, y_j) = \lambda_{IC} d_{IC}^2(y_i^s, y_j^s) + \lambda_{GC} d_{GC}^2(y_i^s, y_j^s), \tag{4}$$

where the intensity compatibility term d_{IC}^2 is the SSD in the overlapping region between two neighboring sketch patches y_i^s and y_j^s, and the gradient compatibility term d_{GC}^2 is the squared Euclidean distance between the dense SIFT descriptors of y_i^s and y_j^s. The intensity compatibility term is for the smoothness of the output sketch. However, only using this term tends to lose some face structures since two blank regions in neighbors have high intensity compatibility. Thus, we further add the gradient compatibility constraint, which requires that the neighboring patches have similar gradient orientations. The use of gradient compatibility can further alleviate the structural loss, an example of which is given in Fig.s 6 (e) and (f) (the region in the red box). We set the weights $\lambda_{IC} = 1$ and $\lambda_{GC} = 0.1$.

2.5 Implementation Details

All the photos and sketches are translated, rotated, and scaled such that the two eye centers of all the face images are at fixed position. We crop the images to 250×200 and the two eye center positions are $(75, 125)$ and $(125, 125)$. All color images are converted to grayscale images for sketch synthesis.

- **Preprocessing on Test Photos.** Empirically, when lighting is near frontal, our algorithm can work well without the preprocessing step. However, for side light, we need to use Contrast Limited Adaptive Histogram Equalization (CLAHE) [14] for preprocessing.[2] We use the setting that the desired histogram shape is Rayleigh distribution (parameter $\alpha = 0.7$).
- **Candidate Selection.** In order to save computational cost, a step of candidate selection as suggested in [10] is used before optimizing the MRF model. For each test photo patch x_i^p, top K ($K = 20$) photo-sketch pairs with the smallest energy of $E_L(x_i^p, y_i) + E_{Pi}(y_i)$ are selected from the training set as candidates. In order to take the advantage of face structures, candidates are searched within a 25×25 local region around patch i instead of in the entire images. The final estimation y_i on node i is selected as one of the K candiates through joint optimization of all the nodes on the MRF network.
- **Two-scale MRF.** We use two-scale MRF with the same setting as in [5]. Patch sizes at the two layers are 10×10 and 20×20, respectively. MAP estimate is used in the belief propagation algorithm [10].
- **Stitching Sketch Patches.** To avoid blurring effect, we use a minimum error boundary cut between two overlapping patches on their overlapped pixels as what is usually done for texture synthesis [15].

3 Experimental Results

We conduct experiments on the CUHK database [5] commonly used in face sketch synthesis research, and a set of celebrity face photos from the web. In all the experiments, 88 persons from the CUHK database are selected for training, and each person has a face photo in a frontal pose under a normal lighting condition, and a sketch drawn by an artist while viewing this photo. In the first

[2] CLAHE improves the method in [5] little and deteriorates its performance in some cases. So we choose to report their results without the preprocessing.

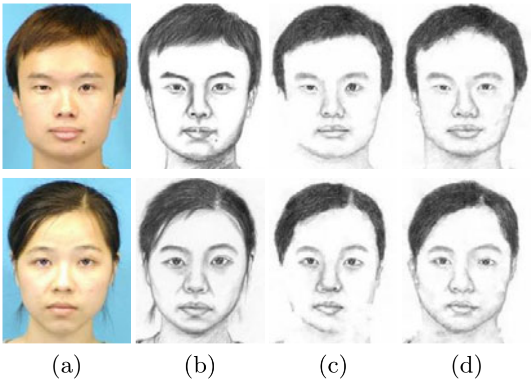


Fig. 7. Representative results on the baseline set. (a) Test photo; (b) Sketch drawn by the artist while viewing the normal lighting photo; (c) Sketch by the method in [5]; (d) Sketch by our method. The results are best viewed on screen.

experiment, 100 other persons are selected for testing. We have three data sets: the baseline set, the lighting variation set, and the pose variation set. The baseline set includes 100 face photos taken in a frontal pose under the same lighting condition as the training set. The lighting variation data set includes three photos with faces in a frontal pose with three different lightings (dark frontal/dark left/dark right) for each person. And the pose variation set includes two photos with faces in left and right poses (with 45 degrees) under a normal lighting condition for each person. In the second experiment, some face photos of Chinese celebrities with uncontrolled lighting conditions and poses are downloaded from the web.[3] All photos are with a neutral expression. Parameters are fixed throughout the experiments. It takes about 2 minutes to synthesize a sketch running our MATLAB implementation on a computer with 3.20 GHz CPU. Due to the paper length, only a limited number of examples are shown in this paper.

3.1 Lighting and Pose Variations

We first investigate the effect of lighting and pose variations separately on the CUHK database. A preliminary test is on the baseline set. Our algorithm performs as well as the method in [5]. On some photos, our algorithm can produce even better face sketches as shown in Fig. 7. To give a quantitative evaluation of the performance, we test the rank-1 and rank-10 recognition rates when a query sketch synthesized from a test photo is used to match the sketches drawn by the artist. The results are shown in Table 1.[4] Our algorithm slightly beats the previous method by 3%.

[3] The CUHK database cannot be used as a training set for photos of people from other ethnic groups, partially due to the human perception.

[4] Recognition rates cannot completely reflect the viual quality of synthesized sketches. It is used as an indirect measurement to evaluate the performance of sketch synthesis since no other proper quantitative evaluation methods are available.

Table 1. Rank-1 (Rank-10) recognition rates using whitened PCA [16]. The whitened PCA model is trained on the 100 sketches drawn by the artist while viewing the baseline set. It performs better than standard PCA without whitening on all the tasks. The reduced number of dimension is 99, and it is the best for all the tasks.

Testing set	[5]	[5] with LBP	[5] with HE	[5] with LR	Ours
Baseline	96% (100%)	-	-	-	99% (100%)
Front Light	58% (87%)	58% (87%)	70% (95%)	75% (96%)	84% (96%)
Side Lights	23.5% (56%)	25.5% (75.5%)	38% (80.5%)	41.5% (78.5%)	71% (87.5%)

Lighting. Although the previous method performs well on the normal lighting set, their performance degrades dramatically when the lighting changes. Our method performs consistently well under different lighting conditions. To make a fair comparison, we also report the results of [5] with several popular illumination normalization methods, including histogram equalization (HE) and luminance remapping (LR) [6], and with LBP [9], an illumination invariant feature.

On the recognition rate, our method beats all the others, as shown in Table 1. The method in [5] performs very poorly without any preprocessing. LR and HE improve the method in [5], but LBP improves little. LR performs better than HE and LBP. As hair and background are included in face photos, previous illumination normalization methods, such as HE, do not perform well. By converting a patch to its LBP feature, information to distinguish different components, which is important for sketch synthesis, may be lost and thus mismatching often occurs. In addition, we find that dark side lighting conditions are more difficult than dark frontal lighting, and under dark side lightings, our method beats all the others by a large amount on the rank-1 recognition rate.

On the visual quality, LR improves the method in [5], but as shown in Fig.s 8 and 9, the facial details and profile are still much worse than those given by our method. Under dark frontal lighting, their results usually have incorrect blank regions and noisy details. Under dark side lightings, the preprocessing helps only a little as it processes the photos globally. See the failed results shown in Fig. 9.

Pose. To test the robustness of our method to pose variations, we use the pose set with the similar lighting condition as the training set. As shown in Fig. 10, our method performs better than the method in [5].[5] With pose variations, the major problem of the results by [5] is to lose some structures especially on the profile. This problem can be efficiently alleviated by the shape priors, photo-to-sketch patch matching and gradient compatibility designed in our model.

3.2 Celebrity Faces from the Web

The robustness of our method is further tested on a challenging set of face photos of Chinese celebrities with uncontrolled lighting and pose variations from

[5] As we do not have the sketches drawn by the artist for different poses, the recognition rates are not tested.

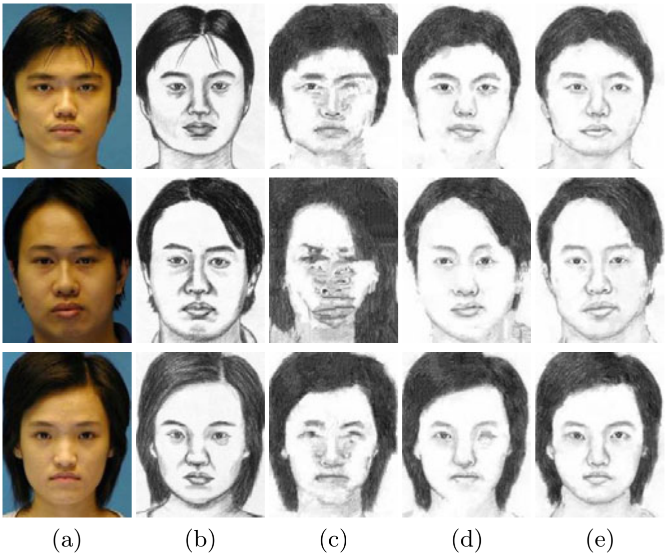


Fig. 8. Representative results on photos under the dark frontal lighting. (a) Test photo; (b) Sketch drawn by the artist while viewing a normal lighting photo; (c) Sketch by the method in [5]; (d) Sketch by the method in [5] with luminance remapping [6]; (e) Sketch by our method. The results are best viewed on screen.

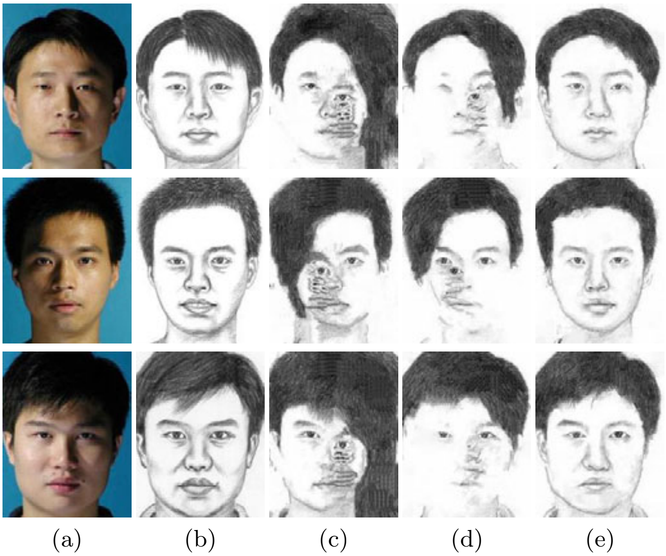


Fig. 9. Representative results of photos under dark side lightings. The notations (a)–(e) are the same as Fig. 8. The results are best viewed on screen.

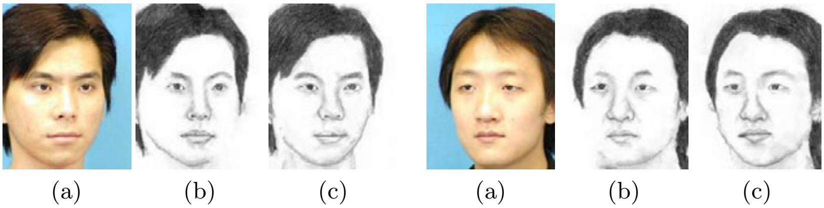


Fig. 10. Representative results of photos with pose variations. (a) Photo; (b) Sketch by the method in [5]; (c) Sketch by our method. The results are best viewed on screen.

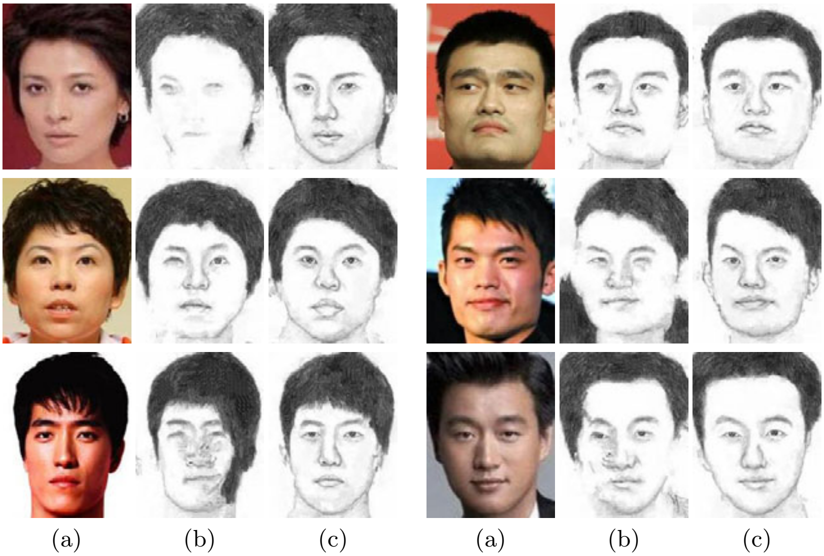


Fig. 11. Results of Chinese celebrity photos. (a) Photo; (b) Sketch by the method in [5] with luminance remapping [6]; (c) Sketch by our method. The results are best viewed on screen.

the web. They even have a variety of backgrounds. As shown in Fig. 11, the method in [5] usually produces noisy facial details and distortions, due to the uncontrolled lightings and the large variations of pose and face shape. However, our method performs reasonably well.

4 Conclusion

We proposed a robust algorithm to synthesize face sketches from photos with different lighting and poses. We introduce shape priors, robust patch matching, and new compatibility terms to improve the robustness of our method. Our method is formulated using the multiscale MRF. It significantly outperforms the state-of-the-art approach. In the future work, we would like to further investigate face sketch synthesis with expression variations.

Acknowledgement

The authors would like to thank Dr. Wei Zhang for his generous help in experiments.

References

1. Tang, X., Wang, X.: Face sketch synthesis and recognition. In: ICCV (2003)
2. Tang, X., Wang, X.: Face sketch recognition. IEEE Trans. CSVT 14, 50–57 (2004)
3. Liu, Q., Tang, X., Jin, H., Lu, H., Ma, S.: A nonlinear approach for face sketch synthesis and recognition. In: CVPR (2005)
4. Gao, X., Zhong, J., Li, J., Tian, C.: Face sketch synthesis algorithm based on E-HMM and selective ensemble. IEEE Trans. CSVT 18, 487–496 (2008)
5. Wang, X., Tang, X.: Face photo-sketch synthesis and recognition. IEEE Trans. PAMI 31, 1955–1967 (2009)
6. Hertzmann, A., Jacobs, C., Oliver, N., Curless, B., Salesin, D.: Image analogies. In: SIGGRAPH (2001)
7. Koshimizu, H., Tominaga, M., Fujiwara, T., Murakami, K.: On KANSEI facial image processing for computerized facialcaricaturing system PICASSO. In: Proc. IEEE Int'l. Conf. on Systems, Man, and Cybernetics (1999)
8. Freeman, W.T., Tenenbaum, J.B., Pasztor, E.C.: Learning style translation for the lines of a drawing. ACM Trans. Graphics 22, 33–46 (2003)
9. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE Trans. PAMI 28, 2037 (2006)
10. Freeman, W., Pasztor, E., Carmichael, O.: Learning low-level vision. IJCV 40, 25–47 (2000)
11. Yedidia, J., Freeman, W., Weiss, Y.: Understanding belief propagation and its generalizations. Exploring artificial intelligence in the new millennium 8, 236–239 (2003)
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
13. Liang, L., Xiao, R., Wen, F., Sun, J.: Face alignment via component-based discriminative search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 72–85. Springer, Heidelberg (2008)
14. Pizer, S., Amburn, E., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., Romeny, B., Zimmerman, J., Zuiderveld, K.: Adaptive histogram equalization and its variations. Computer Vision, Graphics, and Image Processing 39, 355–368 (1987)
15. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: SIGGRAPH (2001)
16. Yang, J., Zhang, D., Yang, J.: Is ICA significantly better than PCA for face recognition? In: ICCV (2005)

Predicting Facial Beauty without Landmarks

Douglas Gray[1], Kai Yu[2], Wei Xu[2], and Yihong Gong[1]

[1] Akiira Media Systems
{dgray,ygong}@akiira.com
http://www.akiira.com/
[2] NEC Labs America*
{kyu,xw}@sv.nec-labs.com
http://www.nec-labs.com/

Abstract. A fundamental task in artificial intelligence and computer vision is to build machines that can behave like a human in recognizing a broad range of visual concepts. This paper aims to investigate and develop intelligent systems for learning the concept of *female facial beauty* and producing human-like predictors. Artists and social scientists have long been fascinated by the notion of facial beauty, but study by computer scientists has only begun in the last few years. Our work is notably different from and goes beyond previous works in several aspects: 1) we focus on *fully-automatic* learning approaches that do not require costly manual annotation of landmark facial features but simply take the raw pixels as inputs; 2) our study is based on a collection of data that is an order of magnitude larger than that of any previous study; 3) we imposed no restrictions in terms of pose, lighting, background, expression, age, and ethnicity on the face images used for training and testing. These factors significantly increased the difficulty of the learning task. We show that a biologically-inspired model with multiple layers of trainable feature extractors can produce results that are much more human-like than the previously used eigenface approach. Finally, we develop a novel visualization method to interpret the learned model and revealed the existence of several beautiful features that go beyond the current averageness and symmetry hypotheses.

1 Introduction

The notion of beauty has been an ill defined abstract concept for most of human history. Serious discussion of beauty has traditionally been the purview of artists and philosophers. It was not until the latter half of the twentieth century that the concept of facial beauty was explored by social scientists [1] and not until very recently that it was studied by computer scientists [2]. In this paper we explore a method of both quantifying and predicting female facial beauty using a hierarchical feed-forward model and discuss the relationship between our method and existing methods.

* Work was performed while all authors were at NEC Labs America.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 434–447, 2010.

The social science approach to this problem can be characterized by the search for easily measurable and semantically meaningful features that are correlated with a human perception of beauty. In 1991, Alley and Cunningham showed that averaging many aligned face images together produced an attractive face, but that many attractive faces were not at all average [3]. In 1994 Grammer and Thornhill showed that facial symmetry can be related to facial attractiveness [4]. Since that time, the need for more complex feature representations has shifted research in this area to computer scientists.

Most computer science approaches to this problem can be described as geometric or landmark feature methods. A landmark feature is a *manually* selected point on a human face that usually has some semantic meaning such as *right corner of mouth* or *center of left eye*. The distances between these points and the ratios between these distances are then extracted and used for classification using some machine learning algorithm. While there are some methods of extracting this information automatically [5] most previous work relies on a very accurate set of dense manual labels, which are not currently available. Furthermore most previous methods are evaluated on relatively small datasets with different evaluation and ground truth methodologies. In 2001 Aarabi *et al.* built a classification system based on 8 landmark ratios and evaluated the method on a dataset of 80 images rated on a scale of 1-4 [2]. In 2005 Eisenthal *et al.* assembled an ensemble of features that included landmark distances and ratios, an indicator of facial symmetry, skin smoothness, hair color, and the coefficients of an eigenface decomposition [6]. Their method was evaluated on two datasets of 92 images each with ratings 1-7. Kagian *et al.* later improved upon their method using an improved feature selection method [7].

Most recently Guo and Sim have explored the related problem of automatic makeup application [8], which uses an example to transfer a style of makeup to a new face.

While all of the above methods produce respectable results for their respective data, they share a common set of flaws. Their datasets are very small and usually restricted to a very small and meticulously prepared subset of the population (*e.g.* uniform ethnicity, age, expression, pose and/or lighting conditions). The images are studio-quality photos taken by professional photographers. As another limitation, all these methods are not fully-automatic recognition systems, because they rely heavily on the accurate manual localization of landmark features and often ignore the image itself once they are collected.

We have attempted to solve the problem with fewer restrictions on the data and a ground truth rating methodology that produces an accurate ranking of the images in the data set. We have collected 2056 images of frontal female faces aged 18-40 with few restrictions on ethnicity, lighting, pose, or expression. Most of the face images are cropped from low-quality photos taken by cell-phone cameras. The data size is 20 times larger the that of any previous study. Some sorted examples can be found in figure 3, the ranking methodology is discussed in section 2. Because of the heavy cost of labeling landmark features on such a large data set, in this paper we solely focused on methodologies which do not require these features[1]. Furthermore, although landmark features and ratios appear to be correlated with facial attractiveness, it is yet unclear to what extent human brains really use these features to form their notion of facial beauty. In this paper we test the hypothesis if a biologically-inspired learning architecture can achieve a near human-level performance on this particular task using a large data set with few restrictions. The learning machine is an instance of the Hubel-Wiesel model [9] which simulates the structure and functionality of visual cortex systems, and consists of multiple layers of trainable feature extractors. In section 3 we discuss discuss the details of the approach to predict female facial attractiveness. In section 4.2 we present the experimental results. Interestingly, we develop a novel way to visualize and interpret the learned black-box model, which reveals some meaningful features highly relevant to beauty prediction and complementary to previous findings.

To summarize, we contribute to the field a method of quantifying and predicting female facial attractiveness using an automatically learned appearance model (as opposed to a manual geometric model). A more realistic dataset has been collected that is 20 times larger than any previously published work and has far fewer restrictions. To the best of our knowledge, it is the first work to test if a Hubel-Wiesel model can achieve a near human-level performance on the task of scoring female facial attractiveness. We also provide a novel method of interpreting the learned model and use it to present evidence for the existence of beautiful features that go beyond the current averageness and symmetry hypotheses. We believe that the work enriched the experiences of AI research toward building generic intelligent systems.

2 Dataset and Ground Truth

In order to make a credible attack on this problem we require a large dataset of high quality images labeled with a beauty score. As of the time of writing, no such data are publicly available. However there does exist a popular website HOTorNOT[2] with millions of images and billions of ratings. Users who submit their photo to this site waive their privacy expectations and agree to have their likeness criticized. Unfortunately the ratings that are associated with images in this dataset were collected from images of people as opposed to faces, and are not valid for the problem we are addressing. We have run face detection software on a subset of images from this website and produced a dataset of 2056 images and collected ratings of our own from 30 labelers.

2.1 Absolute vs. Pairwise Ratings

There are several kinds of ratings that can be collected for this task. The most popular are absolute ratings where a user is presented with a single image and

[1] We also note that landmark feature methods fall outside the purview of computer vision as the original images may be discarded once the features are marked and ratings are collected.

[2] http://www.hotornot.com/

asked to give a score, typically between 1 and 10. Most previous work has used some version of absolute ratings usually presented in the form of a Likert scale [10] where the user is asked about the level of agreement with a statement. This form of rating requires many users to rate each image such that a distribution of ratings can be gathered and averaged to estimate the true score. This method is less than ideal because each user will have a different system of rating images and a user's rating of one image may be affected by the rating given to the previous image, among other things.

Another method used in [11] was to ask a user to sort a collection of images according to some criteria. This method is likely to give reliable ratings but it is challenging for users to sort a large dataset since this requires considering all the data at once.

The final method is to present a user with pair of images and ask which is more attractive. This method presents a user with a binary decision which we have found can be made more quickly than an absolute rating. In section 2.3 we show how to present an informative pair of images to a user in order to speed up the process of ranking the images in a dataset. This is the method that we have chosen to label our data.

2.2 Conversion to Global Absolute Score

Pairwise ratings are easy to collect, but in order to use them for building a scoring system we need to convert the ratings into an absolute score for each image.[3] To convert the scores from pairwise to absolute, we minimize a cost function defined such that as many of the pairwise preferences as possible are enforced and the scores lie within a specified range. Let $\mathbf{s} = \{s_1, s_2, \ldots, s_N\}$ be the set of all scores assigned to images 1 to N. We formulate the problem into minimizing the cost function:

$$J(\mathbf{s}) = \sum_{i=1}^{M} \phi(s_i^+ - s_i^-) + \lambda \mathbf{s}^T \mathbf{s} \tag{1}$$

where (s_i^+/s_i^-) denotes the current scores of the i^{th} comparison and $\phi(d)$ is some cost function which penalizes images that have scores which disagree with one of M pairwise preferences and λ is a regularization constant that controls the range of final scores. We define $\phi(d)$ as an exponential cost function $\phi(d) = e^{-d}$. However this function can be any monotonically increasing cost function such as the hinge loss, which may be advisable in the presence of greater labeling noise. A gradient descent approach is then used to minimize this cost function. This iterative approach was chosen because as we receive new labels, we can quickly update the scores without resolving the entire problem. Our implementation is built on a web server which updates the scores in real time as new labels are entered.

[3] One could alternatively train a model using image pairs and a siamese architecture such as in [12]. However a random cross validation split of the images would invalidate around half of the pairwise preferences.

We note that in our study we hypothesize that in a large sense people agree on a consistent opinion on facial attractiveness, which is also the assumption by most of the previous work. Each individual's opinion can be varied due to factors like culture, race, and education. In this paper we focus on learning the common sense and leave further investigation on personal effects to future work.

2.3 Active Learning

When our system is initialized, all images have a zero score and image pairs are presented to users at random. However as many comparisons are made and the scores begin to disperse, the efficacy of this strategy decays. The reason for this is due in part to labeling noise. If two images with very different scores are compared it is likely that the image with the higher score will be selected. If this is the case, we learn almost nothing from this comparison. However if the user accidentally clicks on the wrong image, this can have a very disruptive effect on the accuracy of the ranking.

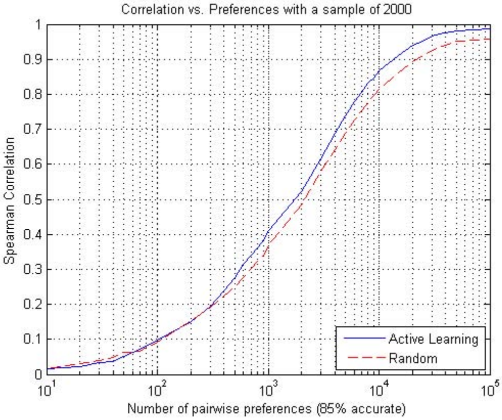


Fig. 1. Simulation results for converting pairwise preferences to an absolute score

For this reason we use a relevance feedback approach to selecting image pairs to present to the user. We first select an image at random with probability inversely proportional to the number of ratings r_i, it has received so far.

$$p(I_i) = \frac{(r_i + \epsilon)^{-1}}{\sum_{j=1}^{N}(r_j + \epsilon)^{-1}} \tag{2}$$

We then select the next image with probability that decays with the distance to first image score.

$$p(I_i|s_1) = \frac{\exp\left(-(s_1 - s_i)^2/\sigma^2\right)}{\sum_{j=1}^{N}\exp\left(-(s_1 - s_j)^2/\sigma^2\right)} \tag{3}$$

Where σ^2 is the current variance of $\mathbf{s}$. This approach is similar to the tournament sort algorithm and has significantly reduced the number of pairwise preferences needed to achieve a desired correlation of 0.9 (15k *vs.* 20k). Figure 1 shows the results of a simulation similar in size to our dataset. In this simulation 15% of the preferences were marked incorrectly to reflect the inherent noise in collecting preference data.

3 Learning Methods

Given a set of images and associated beauty scores, our task is to train a regression model that can predict those scores. We adopt a predictive function that models the relationship between an input image I and the output score s, and learn the model in the following way

$$\min_{\mathbf{w},\theta} \sum_{i=1}^{N} (s_i - y_i)^2 + \lambda \mathbf{w}^T \mathbf{w}, \qquad \text{s. t.} \qquad y_i = \mathbf{w}^\top \Phi(I_i; \theta) + b \tag{4}$$

where I_i is the raw-pixel of the i-th image represented by size 128x128 in YCbCr colorspace, $\mathbf{w}$ is a D-dimensional weight vector, b is a scalar bias term, λ is a positive scalar fixed to be 0.01 in our experiments. As a main difference from the previous work, here we use $\Phi(\cdot)$ to directly operate on raw pixels I for extracting

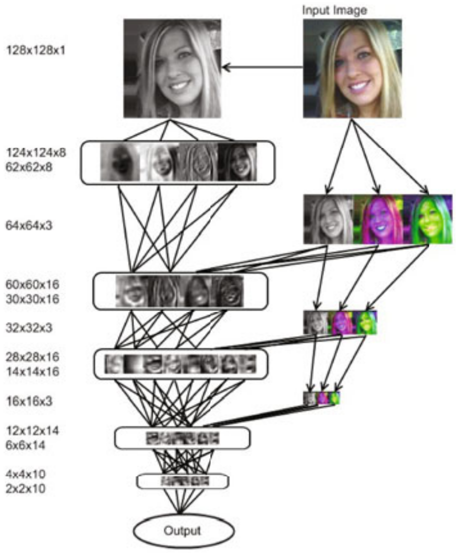


Fig. 2. An overview of the organization of our multiscale model. The first convolution is only performed on the luminance channel. Downsampled versions of the original image are fed back into the model at lower levels. Arrows represent downsampling, lines represent convolution and the boxes represent downsampling with the max operator. Feature dimensions are listed on the left (height x width x channels).

visual features, and its parameters θ are *automatically learned from data* with *no* manual efforts. In our study we investigated the following special cases of the model, whose differences are the definition of $\Phi(I;\theta)$:

- Eigenface Approach: The method has been used for facial beauty prediction by [6], perhaps the only attempt so far requiring no manual landmark features. The method is as follows. We first run singular value decomposition (SVD) on the input training data $[I_1, \ldots, I_N]$ to obtain its rank D decomposition $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, and then set $\theta = \mathbf{U}$ as a set of linear filters to operate on images so that $\Phi(I_i;\theta) = \mathbf{U}^\top I_i$. We tried various D among $\{10, 20, 50, 100, 200\}$ and found that $D = 100$ gave the best performance.
- Single Layer Model: In contrast to Eigenface that uses *global* filters of receptive field 128×128, this model consists of 48 *local* 9×9 linear filters, each followed by a non-linear logistic transformation. The filters convolute over the whole image and produce 48 feature maps, which were then down sampled by running max-operator within each non-overlapping 8×8 region and thus reduced to 48 smaller 15×15 feature maps. The results serve as the outputs of $\Phi(I_i;\theta)$.
- Two Layer Model: We further enrich the complexity of $\Phi(I_i;\theta)$ by adding one more layer of feature extraction. In more details, in the first layer the model employs separate 16 9×9 filters on the luminance channel, and 8 5×5 filters on a down-sampled chrominance channel; in the second layer, 24 5×5 filters are connected to the output of the previous layer, followed by max down-sampling by a factor of 4.
- Multiscale Model: The model is similar to the single-layer model, but with 3 additional convolution/downsampling layers. A diagram of this model can be found in figure 2. This model has 2974 tunable parameters[4].

In each of our models, every element of each filters is a learnable parameter (*e.g.* if our first layer has 8 5x5 filters, then there will be 200 tunable parameters in that layer). As we can see, these models represent a family of architectures with gradually increased complexities: *from linear to nonlinear, from single-layer to multi-layer, from global to local, and from course to fine* feature extractions. In particular, the employed *max* operator makes the architecture more local- and partially scale-invariant, which is particularly useful in our case to handle the diversity of natural facial photos. The architectures can all be seen as a form of convolutional neural network [13] [12] that realize the well-known Hubel-Wiesel model [14] inspired by the structure and functionalities of the visual cortex.

These systems were trained using stochastic gradient descent with a quadratic loss function. Optimal performance on the test set was usually found within a few hundred iterations, models with fewer parameters tend to converge faster both in iterations and computation time. We have tested many models with varying detailed configurations, and found in general that the number and size

[4] Note that this an order of magnitude less than the model trained for the task of face verification in [12].

of filters are not crucial but the number of layers are more important — $\Phi(I_i; \theta)$ containing 4 layers of feature extraction generally outperformed the counterparts with fewer layers.

4 Empirical Study

4.1 Prediction Results

A full and complete comparison with previous work would be challenging both to perform and interpret. Most of the previous methods that have been successful rely on many manually marked landmark features, the distances between them, the ratios between those distances, and other hand crafted features. Manually labeling every image in our dataset by hand would be very costly so we will only compare with methods which do not require landmark features. As of the time of publication, the only such method is the eigenface approach used in [6].

We compare the four learning methods described in Section 3 based on the 2056 female face images and the absolute scores computed from pair-wise comparisons. For each method, we investigate its performance on faces with and without face alignment. We perform alignment using the unsupervised method proposed in [15]. This approach is advantageous because it requires no manual annotation. In all the experiments, we fixed the training set to be 1028 randomly chosen images and used the remaining 1028 images for test.

Pearson's correlation coefficient is used to evaluate the alignment between the machine generated score and the human absolute score on the test data. Table 1 shows a comparison between the four methods – eigenface, single layer, two layer and multiscale models. We can see a significant improvement in the performance with alignment for the eigenface approach and a slight improvement for the hierarchal models. This discrepancy is likely due to the translation invariance that is introduced by the local filtering and down sampling with the max operator over multiple levels, as was first observed by [13]. Another observation is, with more layers being used, the performance improves. We note that eigenface produced a correlation score 0.40 in [6] on 92 studio quality photos of females with similar ages and the same ethnicity origins, but resulted very poor accuracy in our experiments. This shows that the large variability of our data significantly increased the difficulty of appearance-based approaches.

Table 1. Correlation score of different methods

Method	Correlation w/o alignment	Correlation w/ alignment
Eigenface	0.134	0.180
Single Layer Model	0.403	0.417
Two Layer Model	0.405	0.438
Multiscale Model	0.425	0.458

Though the Pearson's correlation provides a quantitative evaluation on how close the machine generated scores are to the human scores, it lacks of intuitive sense about this closeness. In figure 4 we show a scatter plot of the actual and predicted scores for the multiscale model on the aligned test images. This plot shows both the correlation found with our method and the variability in our data. One way to look at the results is that, if without knowing the labels of axes, it is quite difficult to tell which dimension is by human and which by machine. We highly suggest readers to try such a test[5] on figure 4 with an enlarged display.

Figure 3 shows the top and bottom eight images according the humans and the machine. Note that at the ground truth for our training was generated with around 10^4 pairwise preferences, which is not sufficient to rank the data with complete accuracy. However, the notion of complete accuracy is something that can only be achieved for a single user, as no two people have the same exact preferences.

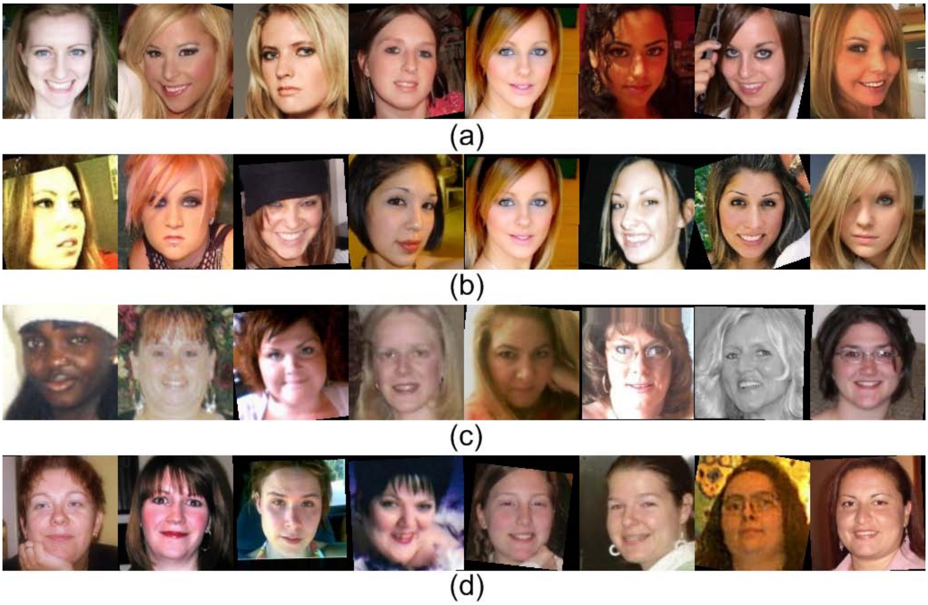


Fig. 3. The top (a/b) and bottom (c/d) eight images from our dataset according to human ratings (a/c) and machine predictions (b/d)

4.2 What Does the Model Learn?

With so much variability it is difficult to determine what features are being used for prediction. In this section we discuss a method of identifying these features to better understand the learned models. One of the classic criticisms of the hierarchical model and neural networks in general, is the *black box* problem. That

[5] Whether or not this constitutes a valid Turning test is left up to the reader.

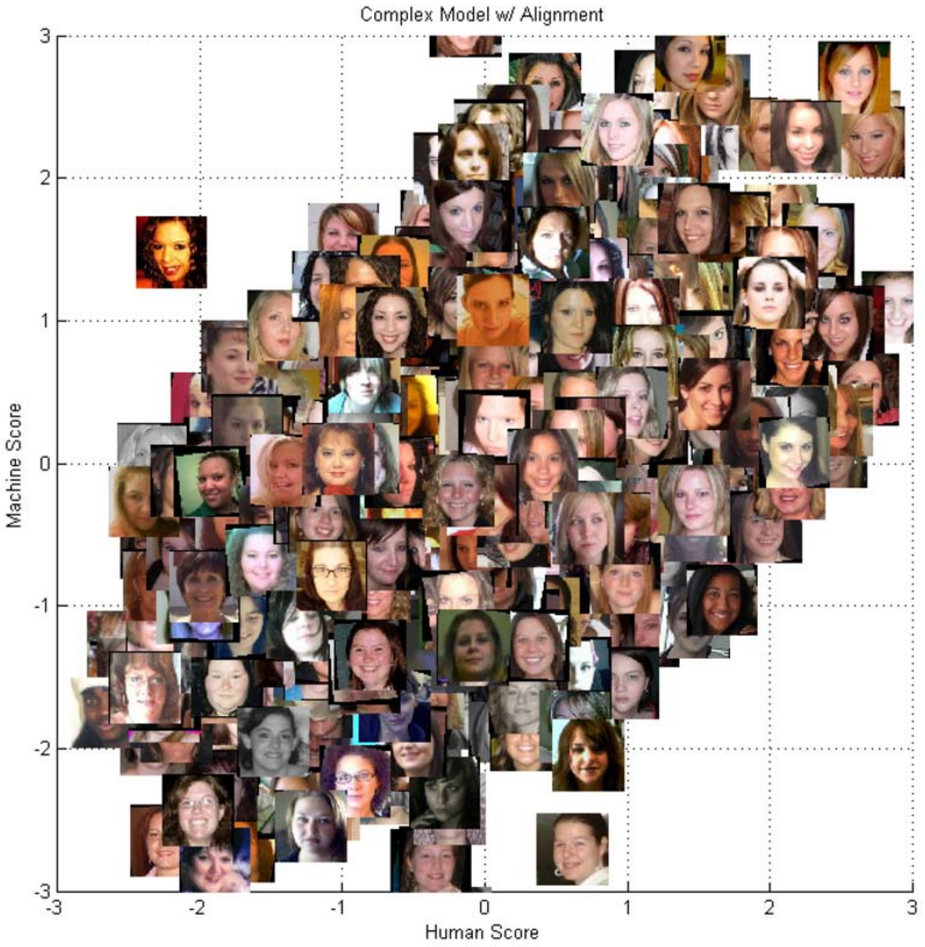


Fig. 4. A Scatter plot showing actual and predicted scores with the corresponding faces

is, what features are we using and why are they relevant? This is typically addressed by presenting the convolution filters and noting their similarity to edge detectors (*e.g.* gabor filters). This was interesting the first time it was presented, but by now everyone in the community knows that edges are important for almost every vision task. We attempt to address this issue using a logical extension to the backpropagation algorithm.

Backpropagation, the most fundamental tasks in training a neural network, is where the gradient of the final error function is propagated back through each layer in a network so that the gradient of each weight can be calculated w.r.t. the final error function. When a neural network is trained, the training input and associated labels are fixed, and the weights are iteratively optimized to reduce the error between the prediction and the true label.

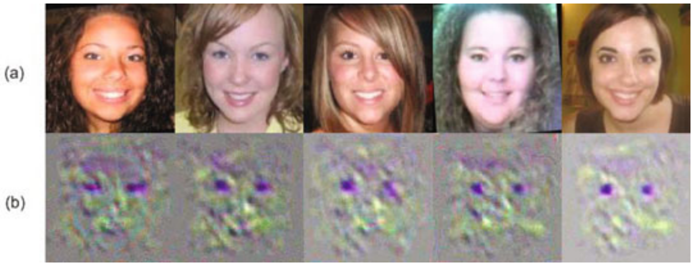


Fig. 5. Several faces (a) with their beauty derivative (b). These images are averaged over 10 gradient descent iterations and scaled in the colorspace for visibility.

We propose the *dual problem.* Given a trained neural network, fix the weights, set the gradient of the prediction to a fixed value and backpropagate the gradient all the way through the network to the input image. This gives the derivative of the image w.r.t. the concept the network was trained with. This information is useful for several reasons. Most importantly, it indicates the regions of the original image that are most relevant to the task at hand. Additionally, the sign of the gradient indicates whether increasing the value of a particular pixel will increase or decrease the network output, meaning we can perform a gradient descent optimization on the original image.

Semantic Gradient Descent. A regularized cost function w.r.t. a desired score ($s^{(d)}$) and the corresponding gradient descent update can be written as:

$$J(I_t) = \phi(s_t - s^{(d)}) + \lambda\phi(I_t - I_0) \tag{5}$$

and

$$I_{t+1} = I_t - \omega\left(\frac{\partial I_t}{\partial s} + \lambda(I_t - I_0)\right) \tag{6}$$

In our implementation we use $\phi(x) = x^2$ and use different values of λ for the luminance and chrominance color channels.

The Derivative of Beauty. The most pressing question is, *What does the derivative of beauty look like?* Figure 5 shows several example images and their respective gradients with respect to beauty for the multiscale model trained on aligned images. This clearly shows that the most important feature in this model is the darkness and color of the eyes.

The gradient descent approach can be used both to *beautify* and *beastify* the original image. If we vary the regularization parameters and change the sign of the derivative, we can visualize the image manifold induced by the optimization. Figure 6 shows how specific features are modified as the regularization is relaxed.

This shows most important features being used to predict beauty and concurs with some human observations about the data and beauty in general.

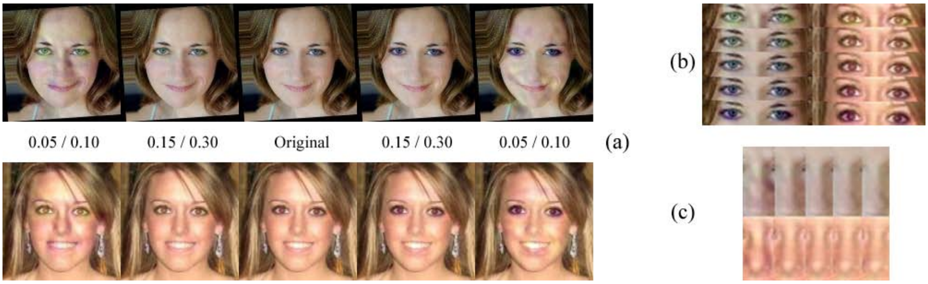


Fig. 6. The manifold of beauty for two images. (a) From left (beast) to right (beauty) we can see how the regularization term (λ_Y/λ_C) controls the amount of modification. Specific features from (a): Eyes (b) and Noses (c).

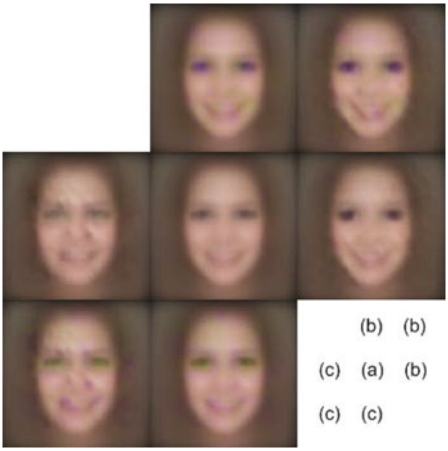


Fig. 7. The average face image (a), beautified images (b) and beastified images (c). The x axis represents changes in the luminance channel, while the y axis represents changes in the chrominance channels.

The first observation is that women often wear dark eye makeup to accentuate their eyes. This makeup often has a dark blue or purple tint. We can see this reflected on the extremes of figure 6 (c). In figure 6 (b), the eyes on the bottom are dark blue/purple tint while the eyes on the top are bright with a yellow/green tint.

The second observation is that large noses are generally not very attractive. If we again look at the extremes of figure 6 (c) we can see that the edges around the nose on the right side have been smoothed, while the same edges on the left side have been accentuated.

The final observation is that a bright smile is attractive. Unfortunately the large amount of variation in facial expressions and mouth position in our training data leads to artifacts in these regions such as in the the extremes of figure 6.

However when we apply these modifications to the average image in figure 7, we can see a change in the perceived expression.

Beautiful Features. One of the early observations in the study of facial beauty was that averaged faces are attractive [3]. This is known as the averageness hypothesis. The average face from the dataset, presented in figure 7, has a score of 0.026. The scores returned by the proposed model are all zero mean, indicating that the average face is only of average attractiveness. This would seem to contradict the averageness hypothesis, however since the dataset presented here was collected from a pool of user submitted photos, it does not represent a truly random sampling of female faces (*i.e.* it may have a positive bias).

As of the time of publication, averageness, symmetry, and face geometry are the only definable features that have been shown to be correlated with facial attractiveness. This paper presents evidence that many of the cosmetic products used by women to darken their eyes and hide lines and wrinkles are in fact attractive features.

5 Conclusion

We have presented a method of both quantifying and predicting female facial beauty using a hierarchical feed-forward model. Our method does not require landmark features which makes it complimentary to the traditional geometric approach [2] [16] [6] [7] [17] when the problem of accurately estimating landmark feature locations is solved. The system has been evaluated on a more realistic dataset that is an order of magnitude larger than any previously published results. It has been shown that in addition to achieving a statistically significant level of correlation with human ratings, the features extracted have semantic meaning. We believe that the work enriches the experience of AI research toward building generic intelligent systems. Our future work is to improve the prediction for this problem and to extend our work to cover the other half of the human population.

References

1. Cross, J., Cross, J.: Age, Sex, Race, and the Perception of Facial Beauty. Developmental Psychology 5, 433–439 (1971)
2. Aarabi, P., Hughes, D., Mohajer, K., Emami, M.: The automatic measurement of facial beauty. In: IEEE International Conference on Systems, Man, and Cybernetics, vol. 4 (2001)
3. Alley, T., Cunningham, M.: Averaged faces are attractive, but very attractive faces are not average. Psychological Science 2, 123–125 (1991)
4. Grammer, K., Thornhill, R.: Human (Homo sapiens) facial attractiveness and sexual selection: the role of symmetry and averageness. J. Comp. Psychol. 108, 233–242 (1994)
5. Zhou, Y., Gu, L., Zhang, H.: Bayesian tangent shape model: estimating shape and pose parameters via Bayesian inference. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 (2003)

6. Eisenthal, Y., Dror, G., Ruppin, E.: Facial Attractiveness: Beauty and the Machine (2005)
7. Kagian, A., Dror, G., Leyvand, T., Cohen-Or, D., Ruppin, E.: A Humanlike Predictor of Facial Attractiveness. In: Advances in Neural Information Processing Systems, pp. 649–656 (2005)
8. Guo, D., Sim, T.: Digital face makeup by example. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009)
9. Hubel, D., Wiesel, T.: Receptive fields and functional architecture of monkey striate cortex. The Journal of Physiology 195, 215–243 (1968)
10. Likert, R.: Technique for the measurement of attitudes. Arch. Psychol. 22, 55 (1932)
11. 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)
12. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 (2005)
13. Fukushima, K.: Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural Networks 1, 119–130 (1988)
14. Hubel, D., Wiesel, T.: Receptive fields, binocular interaction and functional architecture in the cats visual cortex. Journal of Physiology 160, 106–154 (1962)
15. Huang, G., Jain, V., Amherst, M., Learned-Miller, E.: Unsupervised Joint Alignment of Complex Images. In: IEEE International Conference on Computer Vision (2007)
16. Gunes, H., Piccardi, M., Jan, T.: Comparative beauty classification for pre-surgery planning. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 3 (2004)
17. Joy, K., Primeaux, D.: A Comparison of Two Contributive Analysis Methods Applied to an ANN Modeling Facial Attractiveness. In: International Conference on Software Engineering Research, Management and Applications, pp. 82–86 (2006)

Gabor Feature Based Sparse Representation for Face Recognition with Gabor Occlusion Dictionary

Meng Yang and Lei Zhang*

Biometric Research Center, Dept. of Computing,
The Hong Kong Polytechnic University, Hong Kong
{csmyang,cslzhang}@comp.polyu.edu.hk

Abstract. By coding the input testing image as a sparse linear combination of the training samples via l_1-norm minimization, sparse representation based classification (SRC) has been recently successfully used for face recognition (FR). Particularly, by introducing an identity occlusion dictionary to sparsely code the occluded portions in face images, SRC can lead to robust FR results against occlusion. However, the large amount of atoms in the occlusion dictionary makes the sparse coding computationally very expensive. In this paper, the image Gabor-features are used for SRC. The use of Gabor kernels makes the occlusion dictionary compressible, and a Gabor occlusion dictionary computing algorithm is then presented. The number of atoms is significantly reduced in the computed Gabor occlusion dictionary, which greatly reduces the computational cost in coding the occluded face images while improving greatly the SRC accuracy. Experiments on representative face databases with variations of lighting, expression, pose and occlusion demonstrated the effectiveness of the proposed Gabor-feature based SRC (GSRC) scheme.

1 Introduction

Automatic face recognition (FR) is one of the most visible and challenging research topics in computer vision, machine learning and biometrics [1], [2], [3]. Although facial images have a high dimensionality, they usually lie on a lower dimensional subspace or sub-manifold. Therefore, subspace learning and manifold learning methods have been dominantly and successfully used in appearance based FR [4], [5], [6], [7], [8], [9], [10], [11]. The classical Eigenface and Fisherface [4], [5], [6] algorithms consider only the global scatter of training samples and they fail to reveal the essential data structures nonlinearly embedded in high dimensional space. The manifold learning methods have been proposed to overcome this limitation [7], [8], and the representative manifold learning methods include locality preserving projection (LPP) [9], local discriminant embedding (LDE) [10], unsupervised discriminant projection (UDP) [11], etc.

* Corresponding author.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 448–461, 2010.

The success of manifold learning implies that the high dimensional face images can be sparsely represented or coded by the representative samples on the manifold. Very recently, an interesting work was reported by Wright *et al.* [12], where the sparse representation (SR) technique is employed for robust FR. In Wright *et al.*'s pioneer work, the training face images are used as the dictionary to code an input testing image as a sparse linear combination of them via l_1-norm minimization. The SR based classification (SRC) of face images is conducted by evaluating which class of training samples could result in the minimum reconstruction error of the input testing image with the sparse coding coefficients. To make the l_1-norm sparse coding computationally feasible, in general the dimensionality of the training and testing face images should be reduced. In other words, a set of features could be extracted from the original image for SRC. In the case of FR without occlusion, Wright *et al.* tested different types of features, including Eigenface, Randomface and Fisherface, for SRC, and they claimed that SRC is insensitive to feature types when the feature dimension is large enough. To solve the problem of FR with occlusion or corruption, an occlusion dictionary was introduced to code the occluded or corrupted components [12]. Since the occluded face image can be viewed as a summation of non-occluded face image and the occlusion error, with the sparsity constrain the non-occluded part is expected to be sparsely coded by the training face dictionary only, while the occlusion part is expected to be coded by the occlusion dictionary only. Consequently, the classification can be performed based on the reconstruction errors using the SR coefficients over the training face dictionary. Such a novel idea has shown to be very effective in overcoming the problem of face occlusion.

Although the SRC based FR scheme proposed in [12] is very creative and effective, there are two issues to be further addressed. First, the features of Eigenface, Randomface and Fisherface tested in [12] are all holistic features. Since in practice the number of training samples is often limited, such holistic features cannot effectively handle the variations of illumination, expression, pose and local deformation. The claim made in [12] that feature extraction is not so important to SRC actually holds only for holistic features. Second, the occlusion matrix proposed in [12] is an orthogonal matrix, such as the identify matrix, Fourier bases or Haar wavelet bases. However, the number of atoms required in the orthogonal occlusion matrix is very high. For example, if the dimensionality of features used in SRC is 3000, then a 3000 × 3000 occlusion matrix is needed. Such a big occlusion matrix makes the sparse coding process very computationally expensive, and even prohibitive.

In this paper, we propose to solve the above two problems by adopting Gabor local features into SRC. The Gabor filter was first introduced by David Gabor in 1946 [13], and was later shown as models of simple cell receptive fields [14]. The Gabor filters, which could effectively extract the image local directional features at multiple scales, have been successfully and prevalently used in FR [15], [16], leading to state-of-the-art results. Since the Gabor features are extracted in local regions, they are less sensitive to variations of illumination, expression and pose than the holistic features such as Eigenface and Randomface. As in other

Gabor-feature based FR works [15], [16], we will see that the Gabor-feature based SRC (GSRC) improves much the FR accuracy over original SRC. More importantly, the use of Gabor filters in feature extraction makes it possible to obtain a much more compact occlusion dictionary. A Gabor occlusion dictionary computing algorithm is then presented. Compared with the occlusion dictionary used in original SRC, the number of atoms is significantly reduced (often with a ratio 40:1 ∼ 50:1 in our experiments) in the computed Gabor occlusion dictionary. It can not only greatly reduce the computational cost in coding the occluded face images, but also greatly improve the SRC accuracy. Our experiments on benchmark face databases clearly validate the performance of the proposed GSRC method.

The rest of the paper is organized as follows. Section 2 briefly reviews SRC and Gabor filters. Section 3 presents the proposed GSRC algorithm. Section 4 conducts experiments and Section 5 concludes the paper.

2 Related Work

2.1 Sparse Representation Based Classification for Face Recognition

Denote by $A_i = [\boldsymbol{s}_{i,1}, \boldsymbol{s}_{i,2}, ..., \boldsymbol{s}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$ the set of training samples of the i^{th} object class, where $\boldsymbol{s}_{i,j}, j = 1, 2, \cdots, n_i$, is an m-dimensional vector stretched by the j^{th} sample of the i^{th} class. For a test sample $\boldsymbol{y}_0 \in \mathbb{R}^m$ from this class, intuitively, $\boldsymbol{y}_0$ could be well approximated by the linear combination of the samples within A_i, i.e. $\boldsymbol{y}_0 = \sum_{j=1}^{n_i} \alpha_{i,j} \boldsymbol{s}_{i,j} = A_i \boldsymbol{\alpha}_i$, where $\boldsymbol{\alpha}_i = [\alpha_{i,1}, \alpha_{i,2}, ..., \alpha_{i,n_i}]^T \in \mathbb{R}^{n_i}$ are the coefficients. Suppose we have K object classes, and let $A = [A_1, A_2, \cdots, A_K]$ be the concatenation of the n training samples from all the K classes, where $n = n_1 + n_2 + \cdots + n_K$, then the linear representation of $\boldsymbol{y}_0$ can be written in terms of all training samples as $\boldsymbol{y}_0 = A\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1; \cdots; \boldsymbol{\alpha_i}; \cdots; \boldsymbol{\alpha}_K] = [0, \cdots, 0, \alpha_{i,1}, \alpha_{i,2}, \cdots, \alpha_{i,n_i}, 0, \cdots, 0]^T$ [12].

In the case of occlusion or corruption, we can rewrite the test sample $\boldsymbol{y}$ as

$$\boldsymbol{y} = \boldsymbol{y}_0 + \boldsymbol{e}_0 = A\boldsymbol{\alpha} + \boldsymbol{e}_0 = [A,\ A_e] \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}_e \end{bmatrix} \doteq B\boldsymbol{\omega} \tag{1}$$

where $B = [A, A_e] \in \mathbb{R}^{m \times (n+n_e)}$, and the clean face image $\boldsymbol{y}_0$ and the corruption error $\boldsymbol{e}_0$ have sparse representations over the training sample dictionary A and occlusion dictionary $A_e \in \mathbb{R}^{m \times n_e}$, respectively. In [12], the occlusion dictionary A_e was set as an orthogonal matrix, such as identity matrix, Fourier bases, Haar wavelet bases, etc. The SRC algorithm [12] is summarized in Algorithm 1.

2.2 Gabor Filters

The Gabor filters (kernels) with orientation μ and scale ν are defined as [15]:

$$\psi_{\mu,\nu}(z) = \frac{\|k_{\mu,\nu}\|^2}{\sigma^2} e^{\left(-\|k_{\mu,\nu}\|^2 \|z\|^2 / 2\sigma^2\right)} \left[e^{ik_{\mu,\nu}z} - e^{-\sigma^2/2}\right] \tag{6}$$

Algorithm 1. The SRC algorithm in [12]

```
Normalize the columns of $A$ (in the case of non-occlusion) or $B$ (in the case of occlusion) to have unit $l_2$-norm.
Solve the $l_1$-minimization problem:
    $\hat{\boldsymbol{\alpha}}_1 = \arg\min_{\boldsymbol{\alpha}} \left\{ \|\boldsymbol{y}_0 - A\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 \right\}$   (2)
    or
    $\hat{\boldsymbol{\omega}}_1 = \arg\min_{\boldsymbol{\omega}} \left\{ \|\boldsymbol{y} - B\boldsymbol{\omega}\|_2^2 + \lambda \|\boldsymbol{\omega}\|_1 \right\}$   (3)
    where $\hat{\boldsymbol{\omega}}_1 = [\hat{\boldsymbol{\alpha}}_1; \hat{\boldsymbol{\alpha}}_{e1}]$, and $\lambda$ is a positive scalar number that balances the reconstructed error and coefficients' sparsity.
Compute the residuals:
    $r_i(\boldsymbol{y}_0) = \|\boldsymbol{y}_0 - A\delta_i(\hat{\boldsymbol{\alpha}}_1)\|_2$, for $i = 1, \cdots, k$.   (4)
    or
    $r_i(\boldsymbol{y}) = \|\boldsymbol{y} - A_e\hat{\boldsymbol{\alpha}}_{e1} - A\delta_i(\hat{\boldsymbol{\alpha}}_1)\|_2$, for $i = 1, \cdots, k$.   (5)
    where $\delta_i(\cdot): \mathbb{R}^n \to \mathbb{R}^n$ is the characteristic function which selects the coefficients associated with the $i^{th}$ class.
Output that identity$(\boldsymbol{y}_0) = \arg\min r_i(\boldsymbol{y}_0)$ or identify$(\boldsymbol{y}) = \arg\min r_i(\boldsymbol{y})$.
```

where $z = (x, y)$ denotes the pixel, and the wave vector $k_{\mu,\nu}$ is defined as $k_{\mu,\nu} = k_\nu e^{i\phi_\mu}$ with $k_v = k_{\max}/f^v$ and $\phi_\mu = \pi\mu/8$. k_{max} is the maximum frequency, and f is the spacing factor between kernels in the frequency domain. In addition, σ determines the ratio of the Gaussian window width to wavelength.

The convolution of an image Img with a Gabor kernel $\psi_{\mu,\nu}$ outputs $G_{\mu,\nu}(z) = Img(z) * \psi_{\mu,\nu}(z)$, where "$*$" denotes the convolution operator. The Gabor filtering coefficient $G_{\mu,\nu}(z)$ is a complex number, which can be rewritten as $G_{\mu,\nu}(z) = M_{\mu,\nu}(z) \cdot \exp(i\theta_{\mu,\nu}(z))$ with $M_{\mu,\nu}(z)$ being the magnitude and $\theta_{\mu,\nu}(z)$ being the phase. It is known that magnitude information contains the variation of local energy in the image. In [15], the augmented Gabor feature vector $\boldsymbol{\chi}$ is defined via uniform down-sampling, normalization and concatenation of the Gabor filtering coefficients:

$$\boldsymbol{\chi} = \left(\boldsymbol{a}_{0,0}^{(\rho)^t} \ \boldsymbol{a}_{0,1}^{(\rho)^t} \ \cdots \ \boldsymbol{a}_{4,7}^{(\rho)^t} \right)^t \tag{7}$$

where $\boldsymbol{a}_{\mu,\nu}^{(\rho)}$ is the concatenated column vector from down-sampled magnitude matrix $M_{\mu,\nu}^{(\rho)}$ by a factor of ρ, and t is the transpose operator.

3 Gabor-Feature Based SRC with Gabor Occlusion Dictionary

3.1 Gabor-Feature Based SRC (GSRC)

Images from the same face, taken at (nearly) the same pose but under varying illumination, often lie in a low-dimensional linear subspace known as the *harmonic plane* or *illumination cone* [17], [18]. This implies that if there are only variations of illumination, SRC can work very well. However, SRC with the holistic image features is less efficient when there are local deformations of face images, such as certain amount of variations of expressions and pose.

The augmented Gabor face feature vector $\boldsymbol{\chi}$, which is a local feature descriptor, can not only enhance the face feature but also tolerate to image local deformation to some extent. So we propose to use $\boldsymbol{\chi}$ to replace the holistic face features in the SRC framework, and the Gabor-feature based SR without face occlusion is

$$\boldsymbol{\chi}\left(\mathbf{y}_0\right) = X\left(A_1\right)\boldsymbol{\alpha}_1 + X\left(A_2\right)\boldsymbol{\alpha}_2 + \cdots + X\left(A_K\right)\boldsymbol{\alpha}_K = X\left(A\right)\boldsymbol{\alpha} \tag{8}$$

where $X\left(A\right) = \left[X\left(A_1\right) X\left(A_2\right) \cdots X\left(A_K\right)\right]$ and $X\left(A_i\right) = \left[\boldsymbol{\chi}\left(\boldsymbol{s}_{i,1}\right), \cdots, \boldsymbol{\chi}\left(\boldsymbol{s}_{i,n_i}\right)\right]$. With Eq. (8) and replacing $\boldsymbol{y}_0$ and A in Eq. (2) and Eq. (4) by $\boldsymbol{\chi}\left(\mathbf{y}_0\right)$ and $X\left(A\right)$ respectively, the Gabor-feature based SRC (GSRC) can be achieved.

When the query face image is occluded, similar to original SRC, an occlusion dictionary will be introduced in the GSRC to code the occlusion components, and the SR in Eq. (8) is modified to:

$$\boldsymbol{\chi}\left(\boldsymbol{y}\right) = \left[X\left(A\right),\ X\left(A_e\right)\right]\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\alpha}_e\end{bmatrix} \doteq X\left(B\right)\boldsymbol{\omega} \tag{9}$$

where $X(A_e)$ is the Gabor-feature based occlusion dictionary, and $\boldsymbol{\alpha}_e$ is the representation coefficient vector of the input Gabor feature vector $\boldsymbol{\chi}\left(\boldsymbol{y}\right)$ over $X(A_e)$. So in the case of occlusion, GSRC can be achieved by Algorithm 1 through replacing $\boldsymbol{y}$, B, A and A_e in Eq. (3) and Eq. (5) by $\boldsymbol{\chi}\left(\boldsymbol{y}\right)$, $X(B)$, $X(A)$ and $X(A_e)$ respectively. Clearly, the remaining key problem is how to process $X(A_e)$ to make the GSRC more efficient.

3.2 Discussions on Occlusion Dictionary

SRC is successful in solving the problem of face occlusion by introducing an occlusion dictionary A_e to code the occluded face components; however, one fatal drawback of SRC is that the number of atoms in the occlusion dictionary is very big. Specifically, the orthogonal occlusion dictionary, such as the identity matrix, was employed in [12] so that the number of atoms equals to the dimensionality of the image feature vector. For example, if the feature vector has a dimensionality of 3000, then the occlusion dictionary is of size 3000×3000. Such a high dimensional dictionary makes the sparse coding very expensive, and even computationally prohibitive. The empirical complexity of the commonly used l_1-regularized sparse coding methods (such as l_1_ ls [19], l_1_ magic [20], PDCO-LSQR [21] and PDCO-CHOL [21]) to solve Eq. (2) is $O\left(n^{\varepsilon}\right)$ with $\varepsilon \approx 2$ [19]. So if the number of atoms (i.e. n) in the occlusion dictionary is too big, the computational cost will be huge.

By using Gabor-feature based SR, the face image dictionary A and the occlusion dictionary A_e in Eq. (1) will be transformed into the Gabor feature dictionary $X(A)$ and the Gabor-feature based occlusion dictionary $X(A_e)$ in Eq. (9). Fortunately, $X(A_e)$ is compressible, as can be illustrated by Fig. 1.

After the band-pass Gabor filtering of the face images, a uniform down-sampling with a factor ρ is conducted to form the augmented Gabor feature vector $\boldsymbol{\chi}$, as indicated by the red pixels in Fig. 1. The spatial down-sampling is performed for all the Gabor filtering outputs along different orientations and at different scales. Therefore, the number of (spatial) pixels in the augmented Gabor feature vector $\boldsymbol{\chi}$ is $1/\rho$ times that of the original face image; meanwhile, at each position, e.g. P1 or P2 in Fig. 1, it contains a set of directional and scale features extracted by Gabor filtering in the neighborhood (e.g. the circles centered on P1 and P2). Certainly, the directional and scale features at the same spatial location are in general correlated. In addition, there are often some overlaps between the supports of Gabor filters, which makes the Gabor features at neighboring positions also have some redundancies.

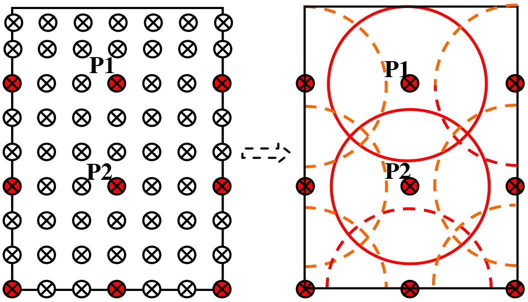


Fig. 1. The uniform down-sampling of Gabor feature extraction after Gabor filtering

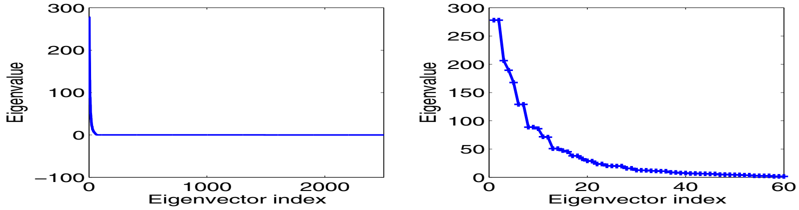


Fig. 2. The eigenvalues (left: all the eigenvalues, right:the first 60 eigenvalues) of Gabor feature-based occlusion matrx

Considering that "occlusion" is a phenomenon of spatial domain, a spatial down-sampling of the Gabor features with a factor of ρ implies that we can use approximately $1/\rho$ times the occlusion bases to code the Gabor features of the occluded face image. In other words, the Gabor-feature based occlusion dictionary $X(A_e)$ can be compressed because the Gabor features are redundant as we discussed above. To validate this conclusion, we suppose that the image size is 50×50, and in the original SRC the occlusion dictionary is an identity matrix $A_e = I \in \mathbb{R}^{2500\times 2500}$. Then the Gabor-feature based occlusion matrix $X(A_e) \in \mathbb{R}^{2560\times 2500}$, where we set ρ=36, $\mu = \{0, \cdots, 7\}$, $\nu = \{0, \cdots, 4\}$. Fig. 2 shows the eigenvalues of $X(A_e)$. Though all the basis vectors of identity matrix I (i.e. A_e) have equal importance, only a few (i.e. 60, with energy proportion of 99.67 %) eigenvectors of $X(A_e)$ have significant eigenvalues, as shown in Fig. 2. This implies that $X(A_e)$ can be much more compactly represented by using only a few atoms generated from $X(A_e)$, often with a compression ratio slightly over ρ:1. For example, in this experiment we have 2500/60=41.7 ≈ ρ=36. Next we present an algorithm to compute a compact Gabor occlusion dictionary under the framework of SRC.

3.3 Gabor Occlusion Dictionary Computing

Now that $X(A_e)$ is compressible, we propose to compute a compact occlusion dictionary from it with the sparsity constraint required by sparse coding. We call this compact occlusion dictionary the Gabor occlusion dictionary and denote it as Γ. Then we could replace $X(A_e)$ by Γ in the GSRC based FR.

For the convenience of expression, we denote by $Z = X(A_e) = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_{n_e}] \in \mathbb{R}^{m_\rho \times n_e}$ the uncompressed Gabor-feature based occlusion matrix, with each column $\boldsymbol{z}_i$ being the augmented Gabor-feature vector generated from each atom of the original occlusion dictionary A_e. The compact occlusion dictionary to be computed is denoted by $\Gamma = [\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_p] \in \mathbb{R}^{m_\rho \times p}$, where p can be set as slightly less than n_e/ρ in practice. It is required that each occlusion basis $\boldsymbol{d}_j, j = 1, 2, \cdots, p$, is a unit column vector, i.e. $\boldsymbol{d}_j^T \boldsymbol{d}_j = 1$. Since we want to replace Z by Γ, it is expected that the original dictionary Z can be well represented by Γ, while the representation being as sparse as possible. With such consideration, our objective function in determining Γ is defined as:

$$J_{\Gamma,\Lambda} = \arg\min_{\Gamma,\Lambda} \left\{ \|Z - \Gamma\Lambda\|_F^2 + \zeta \|\Lambda\|_1 \right\} \quad \text{s.t.} \quad \boldsymbol{d}_j^T \boldsymbol{d}_j = 1, \forall j \qquad (10)$$

where Λ is the representation matrix of Z over dictionary Γ, and ζ is a positive scalar that balances the F-norm term and the l_1-norm term.

Eq. (10) is a joint optimization problem of the occlusion dictionary Γ and the representation matrix Λ. Like in many multi-variable optimization problems, we solve Eq. (10) by optimizing Γ and Λ alternatively. The optimization procedures are described in the following Algorithm 2.

It is straightforward that the proposed Gabor occlusion dictionary computing algorithm converges because in each iteration $J_{\Gamma,\Lambda}$ will decrease, as illustrated in Fig. 3. Consequently, in GSRC we use Γ to replace the $X(A_e)$ in Eq. (9). Finally, the sparse coding problem in GSRC with face occlusion is

$$\boldsymbol{y}_\Gamma = B_\Gamma \boldsymbol{\omega}_\Gamma, \quad \textit{where}\ \boldsymbol{y}_\Gamma = \boldsymbol{\chi}(\boldsymbol{y}),\ B_\Gamma = [\boldsymbol{X}(A),\ \Gamma],\ \boldsymbol{\omega}_\Gamma = [\boldsymbol{\alpha}; \boldsymbol{\alpha}_\Gamma] \qquad (18)$$

Since the number of atoms in Γ is significantly reduced, the number of variables to be solved in $\boldsymbol{\omega}_\Gamma$ is much decreased, and thus the computational cost in solving Eq. (18) is greatly reduced compared with the original SRC.

Algorithm 2. Algorithm of Gabor occlusion dictionary computing

1: Initialize $\varGamma$.

We initialize each column of $\varGamma$ (i.e. each occlusion basis) as a random vector with unit l_2-norm.

2: Fix $\varGamma$ and solve $\varLambda$.

By fixing $\varGamma$, the objective function in Eq. (10) is reduced to

$$J_{\varLambda} = \arg\min_{\varLambda} \left\{ \|Z - \varGamma\varLambda\|_F^2 + \zeta \|\varLambda\|_1 \right\} \tag{11}$$

The minimization of Eq. (11) can be achieved by some standard convex optimization technique. In this paper, we use the algorithm in [19].

3: Fix $\varLambda$ and update $\varGamma$.

Now the objective function is reduced to

$$J_{\varGamma} = \arg\min_{\varGamma} \left\{ \|Z - \varGamma\varLambda\|_F^2 \right\} \quad \text{s.t.} \quad \boldsymbol{d}_j^T\boldsymbol{d}_j = 1, \forall j \tag{12}$$

We can write matrix $\varLambda$ as $\varLambda = \left[\boldsymbol{\beta}_1; \boldsymbol{\beta}_2; \cdots; \boldsymbol{\beta}_p\right]$, where $\boldsymbol{\beta}_j, j = 1, 2, \cdots, p$, is the row vector of $\varLambda$. We update $\boldsymbol{d}_j$ one by one. When updating $\boldsymbol{d}_j$, all the other columns of $\varGamma$, i.e. $\boldsymbol{d}_l, l \neq j$, are fixed. Then $J_{\varGamma}$ in Eq. (12) is converted into

$$J_{\boldsymbol{d}_j} = \arg\min_{\boldsymbol{d}_j} \left\| Z - \sum_{l \neq j} \boldsymbol{d}_l\boldsymbol{\beta}_l - \boldsymbol{d}_j\boldsymbol{\beta}_j \right\|_F^2 \quad \text{s.t.} \quad \boldsymbol{d}_j^T\boldsymbol{d}_j = 1 \tag{13}$$

Let $Y = Z - \sum\limits_{l \neq j} \boldsymbol{d}_l\boldsymbol{\beta}_l$, Eq. (13) can be written as

$$J_{\boldsymbol{d}_j} = \arg\min_{\boldsymbol{d}_j} \left\| Y - \boldsymbol{d}_j\boldsymbol{\beta}_j \right\|_F^2 \quad \text{s.t.} \quad \boldsymbol{d}_j^T\boldsymbol{d}_j = 1 \tag{14}$$

Using Langrage multiplier, $J_{\boldsymbol{d}_j}$ is equivalent to

$$J_{\boldsymbol{d}_j,\gamma} = \arg\min_{\boldsymbol{d}_j} tr\left(-Y\boldsymbol{\beta}_j^T\boldsymbol{d}_j^T - \boldsymbol{d}_j \cdot \boldsymbol{\beta}_jY^T + \boldsymbol{d}_j \cdot (\boldsymbol{\beta}_j\boldsymbol{\beta}_j^T - \gamma)\boldsymbol{d}_j^T + \gamma \right) \tag{15}$$

where γ is a scalar variable. Differentiating $J_{\boldsymbol{d}_j,\gamma}$ with respect to $\boldsymbol{d}_j$, and let it be 0, we have

$$\boldsymbol{d}_j = Y\boldsymbol{\beta}_j^T \left(\boldsymbol{\beta}_j\boldsymbol{\beta}_j^T - \gamma \right)^{-1} \tag{16}$$

Since $\left(\boldsymbol{\beta}_j\boldsymbol{\beta}_j^T - \gamma\right)$ is a scalar and γ is a variable, the solution of Eq. (16) under constrain $\boldsymbol{d}_j^T\boldsymbol{d}_j = 1$ is

$$\boldsymbol{d}_j = Y\boldsymbol{\beta}_j^T \Big/ \left\| Y\boldsymbol{\beta}_j^T \right\|_2 \tag{17}$$

Using the above procedures, we can update all the vectors $\boldsymbol{d}_j$, and hence the whole set $\varGamma$ is updated.

4: Go back to step 2 until the values of $J_{\varGamma,\varLambda}$ in adjacent iterations are close enough, or the maximum number of iterations is reached. Finally, output $\varGamma$.

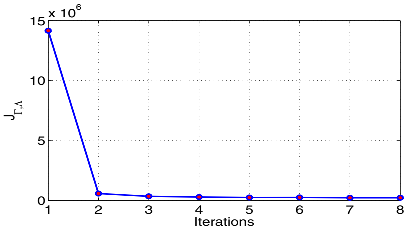


Fig. 3. Illustration of the convergence of Algorithm 2. A Gabor occlusion dictionary with 100 atoms is computed from the original Gabor-feature based occlusion matrix with 4980 columns. The compression ratio is nearly 50:1.

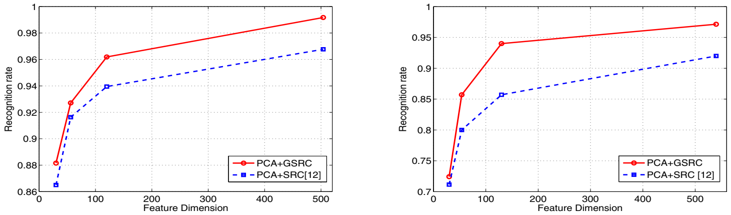
4 Experimental Results

In this section, we perform experiments on benchmark face databases to demonstrate the improvement of GSRC over SRC. To evaluate more comprehensively the performance of GSRC, in section 4.1 we first test FR without occlusion, and then in section 4.2 we demonstrate the robustness and efficiency of GSRC in FR with block occlusion. Finally in section 4.3 we test FR against disguise occlusion. In our implementation of Gabor filters, the parameters are set as $K_{max} = \pi/2$, $f = \sqrt{2}$, $\sigma = \pi$, $\mu = \{0, \cdots, 7\}$, $\nu = \{0, \cdots, 4\}$ by our experimental experiences and fixed for all the experiments below. Here we should also note that the regularization parameters in sparse coding are also tuned by experience (Actually, how to adaptively set the regularizatin parameters is still an open problem). In addition, all the face images are cropped and aligned by using the location of eyes, which is provided by the face databases. The code of our method is available at `http://www4.comp.polyu.edu.hk/~cslzhang/code.htm`.

4.1 Face Recognition without Occlusion

We evaluated the performance of the proposed algorithm on three representative facial image databases: Extended Yale B [22], [18], AR [23] and FERET [24]. In both the original SRC and the proposed GSRC, we used PCA to reduce the feature dimension. The dictionary size is set according to the image variability and the size of database. Some discussions on the dictionary size with respect to image variability are given using FERET database.

1) Extended Yale B Database: As the experiment on Extended Yale B database [22], [18] in [12], for each subject, we randomly selected half of the images for training (i.e. 32 images per subject), and used the other half for testing. The images are normalized to 192×168, and the dimension of the augmented Gabor feature vector of each image is 19760. PCA is then applied to reduce their dimensionality for classification in SRC and GSRC. In our experiments, we set λ=0.001 (refer to Eq. (2)) in GSRC. The results of SRC are from the original paper [12]. Fig. 4(a) shows the recognition rates of GSRC versus feature dimension in comparison with those of SRC. It can be seen that GSRC is much better



(a) (b)

Fig. 4. Recognition rates by SRC and GSRC versus feature dimension on (a) Extended Yale B and (b) AR database

than SRC in all the dimensions. On this database, the maximal recognition rate of GSRC is 99.17%, while that of SRC is 96.77%.

2) AR database: As [12], we chose a subset (only with illumination changes and expressions) of AR dataset [23] consisting of 50 male subjects and 50 female subjects. For each subject, the seven images from Session 1 were used for training, with other seven images from Session 2 for testing. The size of original face image is 165×120, and the Gabor-feature vector is of dimension 12000. We set λ=0.001 in GSRC. The results of SRC are from the original paper [12]. The comparison of GSRC and SRC is shown in Fig. 4(b). Again we can see that GSRC performs much better than SRC under all the dimensions. On this database, the maximal recognition rate of GSRC and SRC are 97.14% and 91.19%, respectively.

The improvement brought by GSRC on AR database is bigger than that on Extended Yale B database. This is because in Extended Yale B, mostly there are only illumination variations between training images and testing images, and dictionary size (i.e. 32 atoms per subject) is big. Thus the original SRC works very well on it. However, the training and testing samples of the AR database have much more variations of expression, time and illumination, and dictionary size (i.e. 7 atoms per subject) is much smaller. Therefore, the local feature based GSRC is much more robust than global feature based SRC in this case.

3) FERET pose database: Here we used the pose subset of the FERET database [24], which includes 1400 images from 198 subjects (about 7 each). This subset is composed of the images marked with ’*ba*’, ’*bd*’, ’*be*’, ’*bf*’, ’*bg*’, ’*bj*’, and ’*bk*’. In our experiment, each image has the size of 80×80. Some sample images of one person are shown in the Fig. 5(a).

Five tests with different pose angles were performed. In test 1 (pose angle is zero degree), images marked with ’*ba*’ and ’*bj*’ were used as training set, and images marked with ’*bk*’ were used as testing set. In all the other four tests, we used images marked with ’*ba*’, ’*bj*’ and ’*bk*’ as gallery, and used the images with ’*bg*’, ’*bf*’, ’*be*’ and ’*bd*’ as probes. Fig. 5(b) compares GSRC (λ=0.005 for best results) with SRC (λ=0.05 for best results) for different poses. The feature dimension in both methods is 350. Obviously, we can see that GSRC has much higher recognition rates than SRC. Especially, when the pose variation is moderate (0° and ±15°), GSRC's recognition rates are 98.5%, 89.5% and 96%,

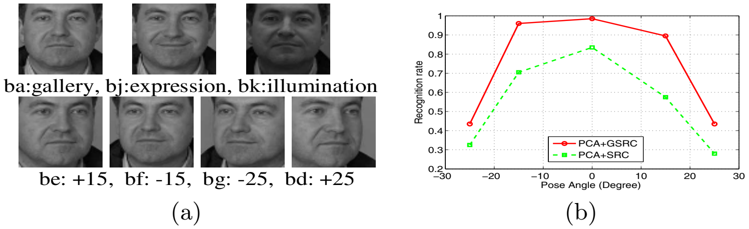


Fig. 5. Samples and results on the FERET pose database. (a). Samples of one subject. (b). Recognition rates of SRC and GSRC versus pose variation.

respectively, about 20% higher than those of the SRC algorithm (83.5%, 57.5% and 70.5%, respectively). The results also show that good performance can be achieved with a small dictionary size when image variability is small (i.e. test 1). Meanwhile, with the same dictionary size, the performance drops as image variability increases (i.e. test 2 $\sim$ 5). It is undeniable that GSRC's performance also degrades much as pose variation becomes large (e.g. $\pm 25^o$). Nevertheless, GSRC can much improve the robustness to moderate pose variation.

4.2 Recognition against Block Occlusion

In this sub-section, we test the robustness of GSRC to the block occlusion using a subset of Extended Yale B face database. We chose Subsets 1 and 2 (717 images, normal-to-moderate lighting conditions) for training, and Subset 3 (453 images, more extreme lighting conditions) for testing. In accordance to the experiments in [12], the images were resized to 96×84, and the occlusion dictionary A_e in SRC is set to an identity matrix.

With the above settings, in SRC the size of matrix B in Eq. (1) is 8064×8781. In the proposed GSRC, the dimension of augmented Gabor-feature vector is 8960 ($\rho \approx 40$). The Gabor occlusion dictionary Γ is then computed using Algorithm 2. In the experiment, we compress the number of atoms in Γ to 200 (i.e. p=200, with compression ratio about 40:1), and hence the size of dictionary B_Γ in Eq. (18) is 8960×917. Compared with the original SRC, the computational cost is reduced from about $O(\eta^2)$ with η=8781 to about $O(\kappa^2)$ with κ=917. Here the time consumption of Gabor feature extraction (about 0.26 second) could

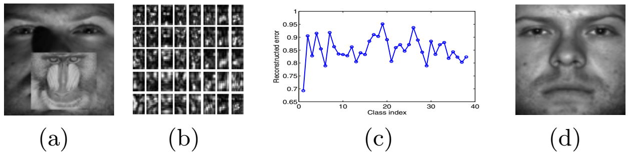


Fig. 6. An example of face recognition with block occlusion. (a). A 30% occluded test face image $\boldsymbol{y}$ from Extended Yale B. (b). Uniformly down-sampled Gabor features $\boldsymbol{\chi}(\boldsymbol{y})$ of the test image. (c). Estimated residuals $r_i(\boldsymbol{y}), i = 1, 2, \cdots, 38$. (d). One sample of the class to which the test image is classified.

be negligible, compared with that of l_1-norm minimization, which is about 90 seconds as reported in [12].

As in [12], we simulated various levels of contiguous occlusion, form 0% to 50%, by replacing a randomly located square block in each test image with an irrelevant image, whose size is determined by the occlusion percentage. The location of occlusion was randomly chosen for each test image and is unknown to the computer. We tested the performance of GSRC with λ=0.0005, and Fig. 6 illustrates the classification process by using an example. Fig. 6(a) shows a test image with 30% randomly located occlusion; Fig. 6(b) shows the argumented Gabor features of the test image. The residuals of GSRC are plotted in Fig. 6(c), and a template image of the identified subject is shown in Fig. 6(d). The detailed recognition rates of GSRC and SRC are listed in the Table 1, where the results of SRC are from the original paper [12]. We see that GSRC can correctly classify all the test images when the occlusion percentage is less than or equal to 30%. When the occlusion percentage becomes larger, the advantage of GSRC over SRC is getting higher. Especially, GSRC can still have a recognition rate of 87.4% when half of image is occluded, while SRC only achieves a rate of 65.3%.

Table 1. The recognition rates of GSRC and SRC under different levels of block occlusion

Occlusion percentage	0%	10%	20%	30%	40%	50%
Recognition rate of GSRC	**1**	**1**	**1**	**1**	**0.965**	**0.874**
Recognition rate of SRC	1	1	0.998	0.985	0.903	0.653

Table 2. Recognition rates of GSRC and SRC on the AR database with disguise occlusion ('-p': partitioned, '-sg': sunglasses, and '-sc': scarves)

Algorithms	GSRC	SRC	GSRC-p	SRC-p
Recognition rate-sg	**93.0%**	87%	**100%**	97.5%
Recognition rate-sc	**79%**	59.5%	**99%**	93.5%

4.3 Recognition Against Disguise

A subset from the AR database consists of 1399 images from 100 subjects (14 samples each class except for a corrupted image w-027-14.bmp), 50 male and 50 female. 799 images (about 8 samples per subject) of non-occluded frontal views with various facial expressions were used for training, while the others for testing. The images are resized to 83×60. So in the original SRC, the size of matrix B in Eq. (1) is 4980×5779. In the proposed GSRC, the dimension of Gabor-feature vectors is 5200 ($\rho \approx 38$), and 100 atoms (with compression ratio about 50:1) are computed to form the Gabor occlusion dictionary by Algorithm 2. Thus the size of dictionary B_Γ in Eq. (18) is 5200×899, and the computational cost is roughly reduced from about $O(\eta^2)$ with η=5779 to about $O(\kappa^2)$ with κ=899, where Gabor feature extraction consumes very little time (about 0.19 second).

We consider two separate test sets of 200 images (1 sample each session and each subject, with neutral expression). The first one contains images of the subjects wearing sunglasses, which occlude roughly 20% of the image. The second one is composed of images of the subjects wearing a scarf, which occlude roughly 40% of the images. The results by GSRC (λ=0.0005) and SRC are listed in Table 2 (where the results of SRC are from the original paper [12]). We see that on faces occluded by sunglasses, GSRC achieves a recognition rate of 93.0%, over 5% higher than that of SRC, while for occlusion by scarves, the proposed GSRC achieves a recognition rate 79%, about 20% higher than that of SRC.

In [12], the authors partitioned image into blocks for face classification by assuming the occlusion is connected. Such an SRC scheme is denoted by SRC-p. Here, after partitioning the image into several blocks, we calculate the Gabor features of each block and then use GSRC to classify each block image. The final classification result is obtained by voting. We denote the GSRC with partitioning as GSRC-p. In experiments, we partitioned the images into eight (4×2) blocks of size 20×30. The Gabor-feature vector of each block is of dimension 800, and the number of atoms in the computed Gabor occlusion dictionary Γ is set to 20. Thus the dictionary B in SRC is of size 600×1379, while the dictionary B_Γ in GSRC is of size 800×819. The recognition rates of SRC-p and GSRC-p are also listed in Table 2. We see that with partitioning, GSRC can lead to recognition rates of 100% on sunglasses and 99% on scarves, also better than SRC.

5 Conclusion

In this paper, we proposed a Gabor-feature based SRC (GSRC) scheme, which uses the image local Gabor features for SRC, and proposed an associated Gabor occlusion dictionary computing algorithm to handle the occluded face images. Apart from the improved face recognition rate, one important advantage of GSRC is its compact occlusion dictionary, which has much less atoms than that of the original SRC scheme. This greatly reduces the computational cost of sparse coding. We evaluated the proposed method on different conditions, including variations of illumination, expression and pose, as well as block occlusion and disguise. The experimental results clearly demonstrated that the proposed GSRC has much better performance than SRC, leading to much higher recognition rates while spending much less computational cost. This makes it much more practicable to use than SRC in real world face recognition.

Acknowledgements

This research is supported by Hong Kong RGC General Research Fund (PolyU 5351/08E) and Hong Kong Polytechnic University Internal Fund (A-SA08).

References

1. Zhao, W.Y., Chellppa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Survey 35, 399–459 (2003)
2. Su, Y., Shan, S.g., Chen, X.L., Gao, W.: Hierarchical ensemble of global and local classifiers for face recognition. IEEE IP 18, 1885–1896 (2009)

3. Zhang, W.C., Shan, S.g., Gao, W., Chen, X.L.: Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In: ICCV, pp. 786–791 (2005)
4. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cognitive Neuroscience 3, 71–86 (1991)
5. Belhumeur, P.N., Hespanha, J.P., Kriengman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE PAMI 19, 711–720 (1997)
6. Yang, J., Yang, J.Y.: Why can LDA be performed in PCA transformed space? Pattern Recognition 36, 563–566 (2003)
7. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
8. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2325 (2000)
9. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. IEEE PAMI 27, 328–340 (2005)
10. Chen, H.T., Chang, H.W., Liu, T.L.: Local discriminant embedding and its variants. In: CVPR, pp. 846–853 (2005)
11. Yang, J., Zhang, D., Yang, J.Y., Niu, B.: Globally maximizing, locally minimizing: Unsupervised discriminant projection with applications to face and palm biometrics. IEEE PAMI 29, 650–664 (2007)
12. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE PAMI 31, 210–227 (2009)
13. Gabor, D.: Theory of communication. J. Inst. Elect. Eng. 93, 429–457 (1946)
14. Jones, J.P., Palmer, L.A.: An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysiology 58, 1233–1258 (1987)
15. Liu, C., Wechsler, H.: Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE IP 11, 467–476 (2002)
16. Shen, L., Bai, L.: A review on gabor wavelets for face recognition. Pattern Analysis and Application 9, 273–292 (2006)
17. Basri, R., Jacobs, D.: Lambertian reflectance and linear subspaces. IEEE PAMI 25, 218–233 (2003)
18. Georghiades, A.S., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE PAMI 23, 643–660 (2001)
19. Kim, S.J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: A method for large-scale l_1-regularized least squares. IEEE Journal on Selected Topics in Signal Processing 1, 606–617 (2007)
20. Cands, E., Romberg, J.: l_1-magic: A collection of matlab routines for solving the convex optimization programs central to compressive sampling (2006), http://www.acm.caltech.edu/l1magic/
21. Saunders, M.: PDCO: Primal-dual interior method for convex objectives (2002), http://www.stanford.edu/group/SOL/software/pdco.html
22. Lee, K., Ho, J., Kriegman, D.: Acquring linear subspaces for face recognition under variable lighting. IEEE PAMI 27, 684–698 (2005)
23. Martinez, A., Benavente, R.: The AR face database (1998)
24. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.: The FERET evaluation methodology for face recognition algorithms. IEEE PAMI 22, 1090–1104 (2000)

Motion Profiles for Deception Detection Using Visual Cues

Nicholas Michael[1], Mark Dilsizian[1], Dimitris Metaxas[1], and Judee K. Burgoon[2]

[1] Computational Biomedicine Imaging & Modelling Center (CBIM),
Rutgers The State University of New Jersey,
110 Frelinghuysen Road,
Piscataway, NJ 08854-8019
{nicholam,mdil,dnm}@cs.rurgers.edu

[2] Center for the Management of Information (CMI),
The University of Arizona,
1130 East Helen Str.,
Tucson, AZ 85721-0108
{jburgoon}@cmi.arizona.edu

Abstract. We propose a data-driven, unobtrusive and covert method for automatic deception detection in interrogation interviews from visual cues only. Using skin blob analysis together with Active Shape Modeling, we continuously track and analyze the motion of the hands and head as a subject is responding to interview questions, as well as their facial micro expressions, thus extracting *motion profiles*, which we aggregate over each interview response. Our novelty lies in the representation of the motion profile distribution for each response. In particular, we use a kernel density estimator with uniform bins in log feature space. This scheme allows the representation of relatively over-controlled and relatively agitated behaviors of interviewed subjects, thus aiding in the discrimination of truthful and deceptive responses.

Keywords: face tracking, skin blob tracking, statistical shape models, deception, nearest-neighbor, support vector machine.

1 Introduction

Wherever two people communicate, deception is a reality. It is present in our everyday social and professional lives [1] and its detection can be beneficial, not only to us individually but to our society as a whole. For example, accurate deception detection can aid law enforcement officers in solving a crime. It can also help border control agents to detect potentially dangerous individuals during routine screening interviews [2].

Currently, the most successful and widespread system is the polygraph which monitors uncontrolled changes in heart rate and electro-dermal response, as a result of the subject's arousal to deceit. Unfortunately, its widespread use does not necessarily mean it is a perfect system. Firstly, in order for it to take the necessary measurements, it needs to be continuously connected to the subject's

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 462–475, 2010.

body. This means that the subject must be cooperative and in close proximity to the device. Secondly, it requires accurate calibration at the beginning of every session, so that a baseline of measurements can be established. Occasionally, it may still fail to give accurate readings, despite the calibration step, if for example, the subject's heart rate increases for reasons unrelated to deception.

Furthermore, the polygraph is an overt system, which means that the subject knows they are being monitored and also knows what measurements are being made. As a result, they may devise techniques to trick the machine, such as remaining calm, in an attempt to control their heart rate or being excited during the calibration phase, so that any excitement due to deception that the polygraph later registers, will mistakenly be regarded as a normal response.

Lastly, the polygraph requires a trained operator, whose skills and abilities control both the likelihood of human error in the interview and the length of the interview itself. Unlike computers, humans will get tired and will eventually need a break. Therefore, what is needed is an automatic and covert system, which can continuously and unobtrusively detect deception, without requiring the subject's cooperation.

In response to this need, researchers have long been trying to decode human behavior, in an attempt to discover deceptive cues. These would aid them in designing systems for automatic deception detection or for training others to detect it [3]. Some deceptive behaviors fall into one of two groups: over-control and agitation [1]. In an attempt to hide their deception, liars who are aware of possible deceptive behavioral cues, may exert extra effort in hiding any behavior [4,5] and particularly reducing movements of their hands, legs and head, while they are being deceptive [6,7,8]. At the other extreme are liars who show signs of agitated behavior triggered by nervousness and fear. As a result, their speech tends to be faster and louder [7] or they may engage in undirected fidgeting [4].

Nevertheless, it is incorrect to assume that agitated or over-controlled behavior is always a sign of deception. One should also consider the normal behavior of a person, as well as the tone and context of the communication taking place. It may be the case that some subjects have a tendency of behaving over-controlled when interrogated by strangers. Others may seem agitated during an interrogation because they had just returned from their morning jog. According to Burgoon's Expectancy Violations Theory (EVT) [9], if in a communication there is considerable deviation of the observed behavior from the expected behavior, then this is a cause for suspicion. For example, an interrogator may become suspicious of a suspect who is relaxed at the beginning of the interrogation but becomes agitated as soon as they are questioned about a crime. Furthermore, in their Interpersonal Deception Theory (IDT) [6,10], Buller and Burgoon state that deception is a dynamic process, whereby liars adjust their behavior according to how much they believe they are suspected of being deceitful. It is likely that during their interaction, liars will unintentionally reveal some behavioral cues as a result of their deception and suspicion [11].

Motivated by the importance of deception detection and the limitations of the widely used polygraph, we propose a novel, automatic and covert approach for detecting deception in interview responses, using visual cues. In every frame we track the movements of a subject's hands and head relative to their body, as well as some of their facial expressions and their 3D head pose. We aggregate these movements and expressions over each response and extract what we call *motion profiles* (see Sect. 3.3). In order to implicitly establish the *baseline* truthful responses for a subject and discriminate them from their deceptive responses we formulate this problem as a Nearest Neighbor classification problem. This formulation, together with our motion profiles, significantly outperforms the method proposed in [11] for a similar interview scenario.

The rest of our paper is organized as follows. Section 2 describes previous attempts in solving the problem of deception detection. Section 3 describes our approach. More specifically, we describe the tracking components of our approach in Sects. 3.1 and 3.2, and we describe our feature set in Sect. 3.3. We describe our experimental results in Sect. 4, we discuss some future extensions of this work in Sect. 5 and we end with some closing remarks in Sect. 6.

2 Previous Work

Having stressed the importance of automatic and covert deception detection in the previous section, we now briefly discuss a few of the research attempts to solve this problem. Some researchers look for physiological indicators which can correlate to deception, in a similar fashion to the polygraph [5]. For example, the authors of [12], build a thermodynamical model to monitor increases in blood flow around the eyes of a subject. However, this method needs a controlled environment and expensive non-standard equipment, thus hindering its broad deployment. Since the method cannot track head movements, its accuracy suffers if the subject's head is moving or at an angle to the camera. Similarly, some researchers, such as the authors of [13,14,15], use functional Magnetic Resonance Imaging (fMRI) to monitor brain activity during interviews. However, methods based on fMRI cannot be used in a covert scenario, they require specialized equipment and a cooperative subject.

Other researchers move away from physiology and attempt to analyze behavioral indicators, instead. Zhang et. al. [2] look at which facial Action Units are activated in a particular facial expression, in order to determine whether it is faked or real. Their method, however, is currently based on static images. Lu et. al. [16] track hand and head skin blobs of subjects to classify their movement signatures as over-controlled, relaxed or agitated. However, it is not convincing that the equation they used for state estimation generalizes to unseen data, given they only tested it on five subjects. One may need to learn subject specific models, since state thresholds can vary across the population. Tsechpenakis et. al. [17] extend the work of [16], translating blob features into illustrator and adaptor behaviors and combining these via a hierarchical Hidden Markov Model [18] to decide if the subject is agitated, relaxed or over-controlled. In the work of Meservy et. al. [11], the step of classifying behaviors [16,17] is bypassed and the authors attempt to directly derive deceptive cues, using blob analysis as in [16].

They segment the video data of interviews into responses and use summary data of each segment, such as the velocity variances of blobs, to make predictions but they do not achieve high accuracy.

We believe that relying on the parametric representation (mean and variance) of the summary data used in [11,16,17], causes a lot of useful information about a feature's distribution to be lost and smooths out any abrupt motions and micro-expressions that briefly occur, when a subject is being deceitful. Eckman and Friesen call this *leakage* [19], while Buller et. al. call this *non-strategic behavior* [20]. We propose to extract motion profiles, which differ from the movement signatures of [16], in that ours are nonparametric representations of the distributions of both *blob* and *facial features*. In this way, this richer representation captures any such leakage that occurs during an interview response.

3 Method

As already discussed, the main idea of our approach is to extract motion profiles of subjects within each response. These motion profiles consist of similar features used in [11,16,17] and are described in Sect. 3.3. In order to extract the features that make up the motion profiles, we use the skin blob tracker of [16] and the Active Shape Model (ASM) tracker of [21]. Sections 3.1 and 3.2 briefly review the skin blob and the ASM trackers respectively. The extracted features are then represented using log-scale histograms, which we describe in Sect. 3.3.

3.1 Head and Hand Blob Tracking

Following the method in [16], we use color analysis, eigen shape decomposition and Kalman filtering to track the position, size and orientation of the head and hand blobs. Instead of a 3D Look-Up-Table (LUT), we build a 2D LUT with Hue and Saturation color components, based on the Hue-Saturation-Value (HSV) skin color distribution of the face and hands. The Value component of the HSV representation is not used, so as to make the representation more robust to illumination than the normalized Red-Green-Blue (RGB) color representation used in [16]. The LUT is built offline from skin color samples. The system extracts face and hand like regions in each frame using the LUT and computes candidate elliptical head and hand blobs. Subsequent eigen shape decomposition and Kalman filtering, prunes the candidate blobs keeping only the most probable ones taking into account the shape of the candidates and the previous position of each of the blobs (see [16] for a more detailed description). A sample frame illustrating the detected blobs and the skin color samples used in building the color model is depicted in Fig. 1.

From the tracked positions of the blobs we compute derived features, as in [11], which are designed to capture behavioral agitation and over control by characterizing relative positioning of the hands, postural shifts and postural openness of a subject. We divide frames into quadrant regions and these are shown in Fig. 2.

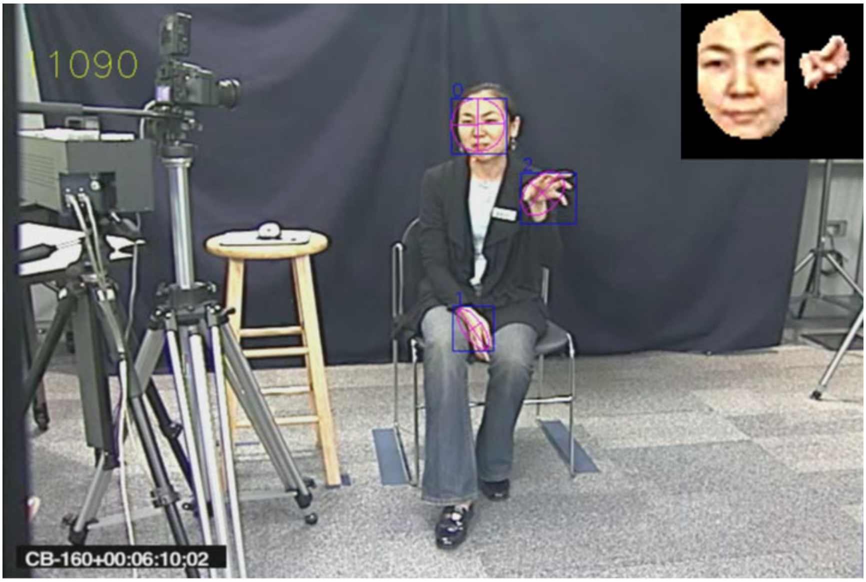


Fig. 1. Sample frame showing the tracked head (blob 0) and hands (blobs 1 and 2) of an interviewee. The tracker records the (x,y) coordinates, area and axis lengths of each detected blob. The skin color samples are shown in the upper right corner.

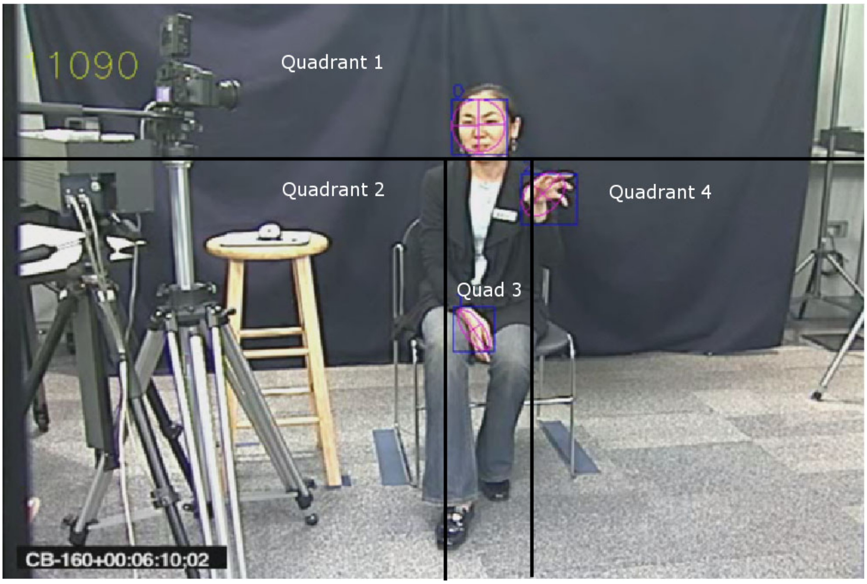


Fig. 2. Illustration of quadrant features. They are used to capture the positions of a subject's hands relative to their body.

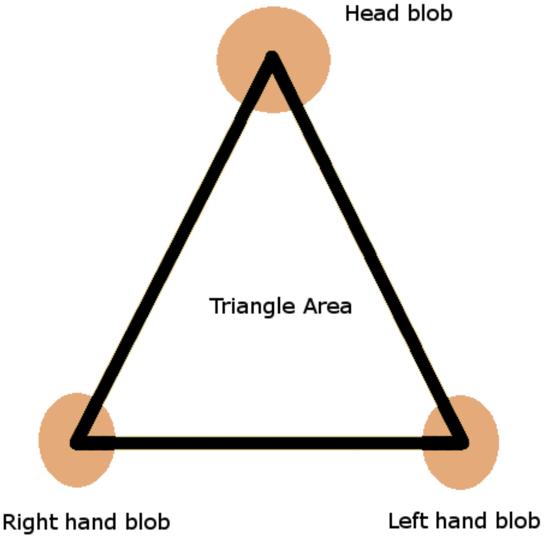


Fig. 3. Illustration of triangle area feature. It is used to quantify the degree of posture openness of a subject.

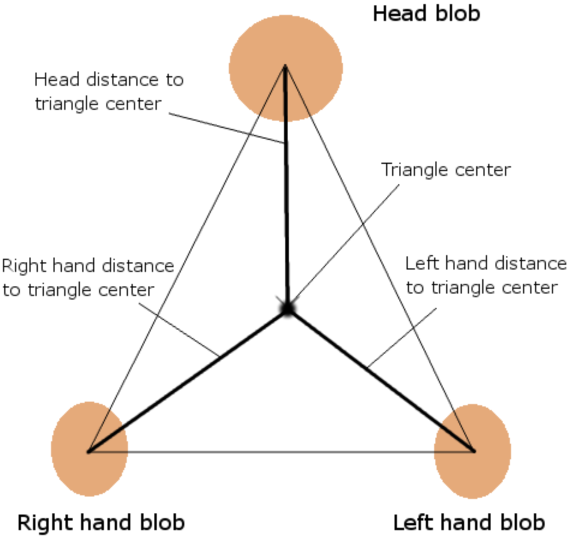

Head blob
Head distance to triangle center
Triangle center
Right hand distance to triangle center
Left hand distance to triangle center
Right hand blob
Left hand blob

Fig. 4. Illustration of distance features of each of the blobs to the triangle's center

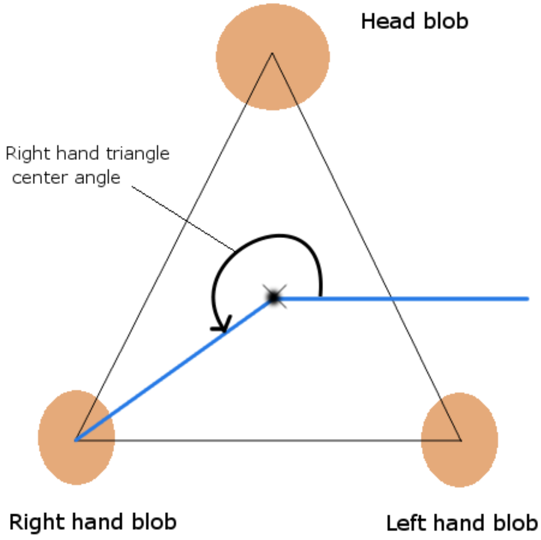


Fig. 5. Illustration of angle features of the blobs relative to the triangle's center

Imagining that the hand and head blobs form the vertices of a triangle, we can use the area and centroid of this triangle to quantify the openness of a subject's posture and any postural shifts. These features are shown in Figs. 3, 4 and 5 (refer to [11] for a more detailed explanation of these features).

In order to account for differences in subject sizes and positioning, we also look at changes in feature values. For example, we compute blob displacement $(\Delta x_{t_i}, \Delta y_{t_i})$ at time t_i, which is also proportional to velocity, using:

$$\Delta x_{t_i} = x_{t_i} - x_{t_{i\ \ 1}} \ , \tag{1}$$

$$\Delta y_{t_i} = y_{t_i} - y_{t_{i\ \ 1}} \ , \tag{2}$$

where (x_{t_i}, y_{t_i}) is its position at time t_i.

3.2 Face Tracking

Face tracking is a challenging problem because the tracker needs to generalize well to unseen faces and handle illumination changes. It should also cope with occlusions (to some degree) and pose changes, such as head rotations, which cause drastic changes in the shape of the face, causing it to lie on a non-linear manifold.

Kanaujia et. al. [21] tackle the problem with an Active Shape Model (ASM), which is a statistical model of facial shape variation. In the ASM framework, a facial shape $\boldsymbol{S}$ is represented by N landmarks, each of which is characterized by its (x, y) image coordinates, so that $\boldsymbol{S} = \{x_1, y_1, \ldots, x_N, y_N\}$. By applying

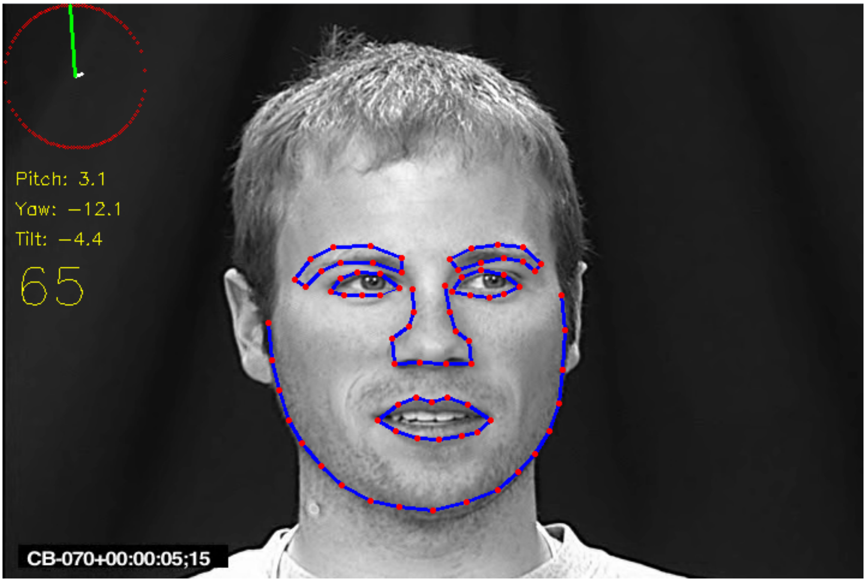


Fig. 6. Sample frame showing tracking of the 79 facial landmarks. The circle in the upper left corner depicts the estimated 3D vector of the head pose. Underneath it are the estimated values of the pitch, yaw and tilt angles of the head.

Principal Component Analysis (PCA) on an aligned training set of facial shapes, a subspace is learned which captures the major modes of shape variation by projecting shapes along the eigenvectors of the shape covariance matrix with the highest eigenvalues. In this way, an aligned shape $\boldsymbol{X} = \Phi(\boldsymbol{S})$, where Φ is the linear transformation that aligns a shape $\boldsymbol{S}$ to the mean shape $\bar{\boldsymbol{X}}$ of the subspace, can be approximated as:

$$\boldsymbol{X} \approx \bar{\boldsymbol{X}} + \boldsymbol{P}\boldsymbol{b} \ , \tag{3}$$

where $\boldsymbol{P}$ is the eigenvector matrix and $\boldsymbol{b}$ is a column vector of shape parameters (encoding).

The authors of [21] additionally propose a piecewise approximation to the non-linear shape manifold using overlapping linear subspaces. Basically this means learning separate ASM models for each subspace and dynamically switching subspaces as the pose of the tracked face changes through a rotation. Their system is made to run in real time by incorporating a Sum of Squared Intensity Differences (SSID) point tracker to track image patches across successive frames assuming small displacements. Moreover, using a Bayesian Mixture of Experts they are able to estimate the 3D pose of the head from the tracked landmarks (refer to [21] for more details). Figure 6 shows a sample frame with the 79 tracked landmarks, along with the predicted 3D head pose.

Using this method we were able to track the head pose and the (x, y) positions of the landmarks in every frame of the video sequences we analyzed. The tracked landmarks were used to compute derived features designed to capture facial micro expressions and asymmetries. Namely these are: change in angle between the mouth's corner points, change in angle between the mouth's centroid and each of its corner points, change in mouth area, displacement of inner and outer left and right eyebrows. The left/right mouth corner points are computed as the means of the three leftmost/rightmost mouth landmarks. The left/right mouth corner angle is the angle formed by the leftmost/rightmost mouth landmark and the two landmarks on either side of it. Finally, the displacement of the inner/outer eyebrow is computed using the mean displacement of the four innermost/outermost eyebrow landmarks. From this displacement we subtract the mean displacement of the six lower nose landmarks to account for head displacements, assuming that the nose is the most stable face component.

3.3 Motion Profiles

In order to summarize the tracked motions and expressions of subjects we propose to extract motion profiles. These are similar to the movement signatures of [16], however our motion profiles include facial expression information. In addition, our motion profiles are log-scaled in order to capture information important to deception detection, namely, little or no movement, and extreme movement. In each subject's response the majority of frames involve a small amount of motion. In other words, subjects rarely make extreme movements for the entire duration of their response. Therefore we change the scale of our data representation in order to properly space out the data to allow for discrimination.

All motion is histogrammed into five bins, with each bin having an exponentially increasing size. Therefore, the first bin covers a very small range (corresponding to little or no motion) and the fifth bin covers the largest range (corresponding to all extreme motions). This new representation of the data is successful at isolating the over-controlled and agitation responses that Ekman et. al. point to as being important indicators of deception [1]. In Fig. 7 we show the size of each bin for hand motion averaged over all responses for two different subjects. The graph on the left demonstrates a subject exhibiting agitated deceptive behavior: when responses are deceptive, the no motion bin shows a dip and the high motion bin shows a spike, relative to their truthful responses. The graph on the right demonstrates over-controlled deceptive behavior: when responses are deceptive, the no motion bin shows a spike and the high motion bin shows a drop, again, relative to their truthful responses.

Let $\{x_{i,j}\}_{i=1}^{F}$ be the set of F features we extract from frame j as described in Sects. 3.1 and 3.2. By grouping together features extracted from m consecutive frames, we form a feature set of the form $\{\{x_{i,j}\}_{i=1}^{F}, \ldots, \{x_{i,j+m-1}\}_{i=1}^{F}\}$, which forms the basis of the motion profile over a response r_q of m frames. For each of the F feature channels, we compute a k–bin normalized log-scale histogram of the feature values $x_{i,j}$ for $j = 1, \ldots, m$, resulting in F histograms having a total of kF bins. We call $\boldsymbol{x}_{r_q}$ a motion profile because the histograms capture

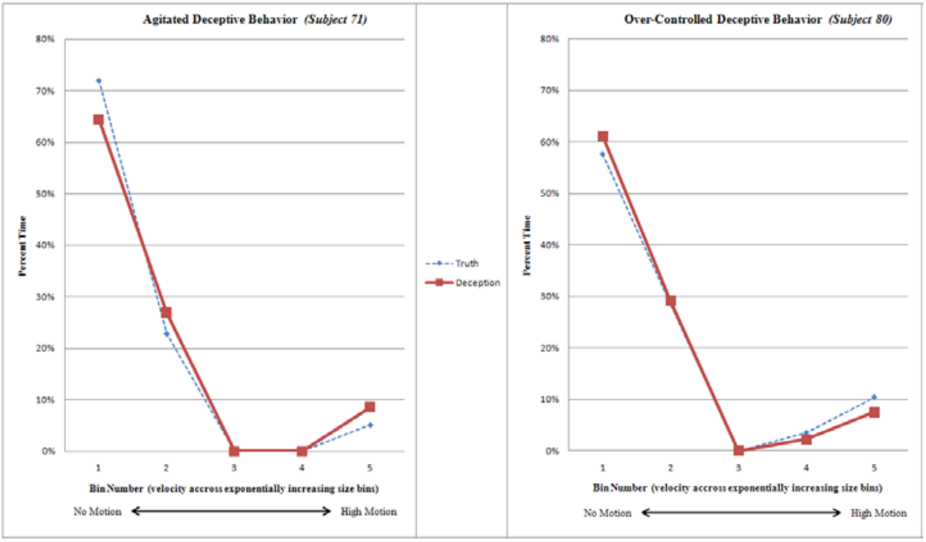


Fig. 7. Average hand motion shown for two different subjects. Graphs show 5 velocity bins from “no motion” to “high motion”

the distribution of feature values within the response. For classification we use a Nearest Neighbor classifier, whereby for a given test response, we assign it the label of its nearest training neighbor.

4 Experiments and Results

A laboratory experiment was conducted during which interviewees, who represented diverse cultural backgrounds, responded to 13 one-word answer questions and 13 longer questions posed by trained interviewers. Interviewees answered some questions truthfully and some deceptively according to a predetermined sequence. Half were randomly assigned to follow a truth-first sequence and half were randomly assigned to begin with a deception-first sequence. During the interview, three high-speed digital cameras recorded kinesic behavior: one recorded full body profile view, one recorded full body frontal view and one recorded frontal facial view only. After each block of three questions, interviewees rated their truthfulness in answering each question on a 0 (not at all) to 10 (completely truthful) scale. Interviews were typically 20 - 30 minutes long.

The recruitment efforts netted a multi-culturally diverse sample of 220 participants. Demographically, the mean age for the total sample was 28.9 years (while $\sigma = 13.34$), with 36% aged 21 and under, 48% aged 22 to 40, and 16% over 40 years of age. By gender, 55% were male and 45% were female. However, recording difficulties allowed only 147 interviews to be analyzed to date. We focused on the responses to the first 12 out of the 13 longer questions in each interview,

Table 1. Data set composition showing number of deceptive and truthful responses (six of each kind per subject) used for Leave One Out Cross Validation. Numbers based on 147 subjects.

	Deceptive	Truthful	Total
Total	882	882	1764

Table 2. Comparison of classification accuracy. Although the experimental design and classification protocol of [11] was different, it was the most similar to ours in that it dealt with classifying interview responses as deceptive or truthful.

Method	Precision	Recall	Accuracy
Mock Theft Experiment [11]	59.2%	63.6%	60.0%
SVM	68.0%	70.1%	68.5%
Nearest Neighbor	**81.7%**	**81.5%**	**81.6%**

Table 3. Mean confusion matrix of Nearest Neighbor classifiers

NN Conf. Matrix	Pred. Deceptive	Pred. Truthful
True Deceptive	81.5%	18.5%
True Truthful	18.3%	81.8%

meaning that in total we had 1764 responses (half deceptive and half truthful). The data set composition in terms of number of frames involving deceptive and truthful responses is shown in Table 1.

Each video interview was analyzed and features were extracted from each frame. The full body frontal view was analyzed by the blob tracker and the facial frontal view was analyzed by the ASM face tracker, while the profile view was not used in our current analysis. We used 5 histogram bins per feature channel with uniform log space width (specific to the current subject). In this way, the first two bins were wide enough to capture the very small feature values corresponding to over-controlled behavior, while the width of each of the remaining bins was successively increased to capture increasingly larger movements corresponding to relaxed and agitated behaviors, respectively. We built 147 separate Nearest Neighbor models (one for each of the 147 subjects), using Leave One Out Cross Validation (LOOCV), where for each of the interview responses, we hold one out to be used for testing and train the model on the rest, reporting the average LOOCV performance over all 147 NN models. We also tried an SVM classifier for each of these 147 subject-specific models with an RBF kernel (scale and complexity parameters determined by cross validation for each subject). Our motion profile NN models achieve an accuracy of **81.6%**, which is significantly better than the accuracy of the method in [11] for a similar interview scenario, and ours is over a larger dataset, too. However, note that, unlike our work, the authors of [11] attempt to build models that *generalize* over all subjects, and are, thus, doing LOOCV per subject. Instead, we build 147 *subject-specific* models, doing LOOCV per response per subject. It is, therefore, clear that our subject-specific models perform better than a general model over all subjects. We attribute this to the fact that different subjects may have different deceptive behaviors and different baseline truthful behaviors, as opposed to there being a universal deception cue or "threshold", which holds for everyone and discriminates truth from deception. All results are shown in Table 2, while Table 3 shows the mean confusion matrix from all NN models.

Our proposed system as presented has the limitation that training data must first be collected for a test subject so that the model can be trained. Acquiring such training data might not be trivial in the situations where such a system can be useful. Nevertheless, the proposed work can serve the purpose of providing the foundation for understanding exactly what constitutes the peculiarities that characterize the deceptive tactics of different individuals.

5 Future Work

In our current work we extended the feature vectors used in the previous work of [11,16,17], who focused only on blob features, by augmenting features extracted from the face, such as eyebrow displacements and mouth angle changes (see Sect. 3.2). In the future, we plan to inspect these facial features more closely and look at texture changes around key parts of the face, such as the eyes, mouth and nose. Such texture changes may be more information-rich than shape changes, possibly serving as a better indicator of behavioral state.

Additionally, we have already started looking at interview data where the subjects originate from many different cultures, since we are trying to discover culture-specific patterns in deception tactics. Once this hurdle is passed, then collecting training data will become easier because it need only be culture-specific (at least), instead of subject-specific. In this way, the applicability of the proposed method can be improved.

Moreover, our current method looks at motion profiles in a static context. Surely, utilizing temporal information of how the motion profile varies within a response could be beneficial, as shown in the proof of concept study of [17], so our future work will attempt to augment this temporal dimension to the model.

Lastly, the problem can also be posed as a Multiple Instance Learning (MIL) problem in which bags are the interview responses and their instances are all motion profiles computed within them. This intuitive learning approach may yield even more promising results.

6 Conclusion

We proposed a novel and fully automatic method for deception detection from video input. Experimental results show that this approach has great potential and contributes to understanding deception detection from visual input in general. We achieved 81.6% classification accuracy, outperforming the 60.0%, which was previously achieved by [11] on a similar but smaller dataset and under similar conditions, showing that subject-specific models work better than general models. Consistent performance over many subjects and cross-validation indicate that the model does not overfit the data. However, data from additional psychological studies of deception would help to further confirm that the behaviors discriminated by our learning algorithms are the deceptive behaviors we are attempting to isolate. Nevertheless, our results show a convincing proof of concept and suggest a promising future for the identification of deceptive behavior from video sequences.

Acknowledgments. We acknowledge the efforts of the CMI group at the University of Arizona (http://cmi.arizona.edu). Led by Dr. Judee K. Burgoon, the group was responsible for conducting the interviews and collecting all video data used in this work. We also thank Fei Yang for his assistance with processing the video data. This research has been partially supported by the following grants to Dr. Dimitri Metaxas: ONR-N000140910104 and NASA-NSBRI-NBTS01601.

References

1. Ekman, P.: Telling lies: Clues to deceit in the marketplace, politics, and marriage, vol. 2. WW Norton and Company, New York (1992)
2. Zhang, Z., Singh, V., Slowe, T.E., Tulyakov, S., Govindaraju, V.: Real-time automatic deceit detection from involuntary facial expressions. In: IEEE CVPR (2007)
3. George, J., Biros, D.P., Burgoon, J.K., Nunamaker, J.: Training professionals to detect deception. In: NSF/NIJ Symposium on Intelligence and Security Informatics, Tucson, AZ (2003)
4. DePaulo, B., Lindsay, J., Muhlenbruck, L., Charlton, K., Cooper, H.: Cues to deception. Psychological Bulletin 129, 74–118 (2003)
5. Vrij, A.: Detecting lies and deceit: The psychology of lying and its implications for professional practice. Wiley, Chichester (2000)
6. Buller, D., Burgoon, J., White, C., Ebesu, A.: Interpersonal deception: VII. Behavioral Profiles of Falsification, Equivocation and Concealment. Journal of Language and Social Psychology 13, 366–395 (1994)
7. Ekman, P.: Lying and nonverbal behavior: Theoretical issues and new findings. Journal of Nonverbal Behavior 12, 163–176 (1988)
8. Vrij, A., Edward, K., Roberts, K., Bull, R.: Detecting deceit via analysis of verbal and nonverbal behavior. Journal of Nonverbal Behavior 24, 239–263 (2000)
9. Burgoon, J.K.: A communication model of personal space violations: Explication and an initial test. Human Communication Research 4, 129–142 (1978)
10. Buller, D., Burgoon, J.: Interpersonal deception theory. Communication Theory 6, 203–242 (1996)
11. Meservy, T.O., Jensen, M.L., Kruse, J., Burgoon, J.K., Nunamaker, J.F.: Automatic Extraction of Deceptive Behavioral Cues from Video. In: Kantor, P., Muresan, G., Roberts, F., Zeng, D.D., Wang, F.-Y., Chen, H., Merkle, R.C. (eds.) ISI 2005. LNCS, vol. 3495, pp. 198–208. Springer, Heidelberg (2005)
12. Buddharaju, P., Dowdall, J., Tsiamyrtzis, P., Shastri, D., Pavlidis, I., Frank, M.G.: Automatic thermal monitoring system (ATHEMOS) for deception detection. IEEE CVPR 2, 1179 (2005)

13. Johnson, R., Barnhardt, J., Zhu, J.: The contribution of executive processes to deceptive responding. Neuropsychologia 42, 878–901 (2004)
14. Kozel, F.A., Johnson, K.A., Mu, Q., Grenesko, E.L., Laken, S.J., George, M.S.: Detecting deception using functional magnetic resonance imaging. Biological Psychiatry 58, 605–613 (2005)
15. Ganis, G., Kosslyn, S.M., Stose, S., Thompson, W.L., Yurgelun-Todd, D.A.: Neural correlates of different types of deception: An fmri investigation. Cerebral Cortex 13, 830–836 (2003)
16. Lu, S., Tsechpenakis, G., Metaxas, D., Jensen, M.L., Kruse, J.: Blob analysis of the head and hands: A method for deception detection and emotional state identification. In: Hawaii International Conference on System Sciences, Big Island, Hawaii (2005)
17. Tsechpenakis, G., Metaxas, D., Adkins, M., Kruse, J., Burgoon, J., Jensen, M., Meservy, T., Twitchell, D., Deokar, A., Nunamaker, J.: HMM-based deception recognition from visual cues. In: IEEE ICME, pp. 824–827. IEEE, Los Alamitos (2005)
18. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257–286 (1989)
19. Ekman, P., Friesen, W.V.: Nonverbal leakage and clues to deception. Psychiatry 32, 88–106 (1969)
20. Buller, D.B., Burgoon, J.K., Buslig, A., Roiger, J.: Interpersonal deception: Viii. nonverbal and verbal correlates of equivocation from the bavelas et al (1990); research. Journal of Language and Social Psychology 13, 396–417 (1994)
21. Kanaujia, A., Huang, Y., Metaxas, D.: Tracking facial features using mixture of point distribution models. In: ICVGIP (2006)

A Robust and Scalable Approach to Face Identification

William Robson Schwartz, Huimin Guo, and Larry S. Davis

University of Maryland, A.V. Williams Building, College Park, MD, 20742
{schwartz,hmguo,lsd}@cs.umd.edu

Abstract. The problem of face identification has received significant attention over the years. For a given probe face, the goal of face identification is to match this unknown face against a gallery of known people. Due to the availability of large amounts of data acquired in a variety of conditions, techniques that are both robust to uncontrolled acquisition conditions and scalable to large gallery sizes, which may need to be incrementally built, are challenges. In this work we tackle both problems. Initially, we propose a novel approach to robust face identification based on Partial Least Squares (PLS) to perform multi-channel feature weighting. Then, we extend the method to a tree-based discriminative structure aiming at reducing the time required to evaluate novel probe samples. The method is evaluated through experiments on FERET and FRGC datasets. In most of the comparisons our method outperforms state-of-art face identification techniques. Furthermore, our method presents scalability to large datasets.

Keywords: Face Identification, Feature combination, Feature selection, Partial Least Squares.

1 Introduction

The three primary face recognition tasks are *verification*, *identification*, and *watch list* [1]. In verification, the task is to accept or deny the identity claimed by a person. In identification, an image of an unknown person is matched to a gallery of known people. In the watch list task, a face recognition system must first detect if an individual is on the watch list. If the individual is on the watch list, the system must then correctly identify the individual. The method described in this paper addresses the identification task.

Previous research has shown that face recognition under well controlled acquisition conditions is relatively mature and provides high recognition rates even when a large number of subjects is in the gallery [2,3]. However, when images are collected under uncontrolled conditions, such as uncontrolled lighting and changes in facial expressions, the recognition rates decrease significantly.

Due to the large size of realistic galleries, not only the accuracy but also the scalability of a face identification system needs to be considered. The main scalability issues are the following. First, the number of subjects in the gallery can be

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 476–489, 2010.

quite large, so that common search techniques, such as brute force nearest neighbor, employed to match probe faces do not scale well. Second, in applications such as surveillance and human computer interaction, in which new subjects are added incrementally, the necessity of rebuilding the gallery models every time a new subject is added compromises the computational performance of the system.

We tackle both problems. In order to reduce the problems associated with data collected under uncontrolled conditions, we consider a combination of low-level feature descriptors based on different clues (such approaches have provided significant improvements in object detection [4,5] and recognition [6]). Then, feature weighting is performed by Partial Least Squares (PLS), which handles very high-dimensional data presenting multicollinearity and works well even when very few samples are available [5,7,8,9]. Finally, a one-against-all classification scheme is used to model the subjects in the gallery.

To make the method scalable to the gallery size, we modify the one-against-all approach to use a tree-based structure. At each internal node of the tree, a binary classifier based on PLS regression is used to guide the search for the matching subject in the gallery. The use of this structure provides substantial reduction in the number of comparisons when a probe sample is matched against the gallery and also eliminates the need for rebuilding all PLS models when new subjects are added to the gallery.

Our proposed face identification approach outperforms state-of-art techniques in most of the comparisons considering standard face recognition datasets, particularly when the data is acquired under uncontrolled conditions, such as in experiment 4 of the FRGC dataset. In addition, our approach can also handle the problem of insufficient training data – results show high performance when only a single sample per subject is available. Finally, due to the incorporation of the tree-based structure, a significant number of comparisons can be saved when compared to approaches based on brute force nearest neighbor search.

2 Related Work

Detailed discussion of face recognition and processing can be found in recent and comprehensive surveys written by Tolba et al. [2] and Zhao et al. [3].

Most approaches to face recognition can be divided into two categories: holistic matching methods and local matching methods [10]. Methods in the former category use the whole face region to perform recognition and includes techniques such as subspace discriminant analysis, SVM, and AdaBoost; these may not cope well with the generalizability problem due to the unpredictable distribution of real-world testing face images. Methods in the latter category first locate several facial features and then classify the faces according to local statistics.

Local binary patterns (LBP) and Gabor filters are descriptors widely used in face recognition. LBP is robust to illumination variations due to its invariance to monotonic gray-scale changes and Gabor filters are also robust to illumination variations since they detect amplitude-invariant spatial frequencies of pixel gray values [10]. There are several combinations or variations based on these descriptors that have been used for face recognition [6,11,12,13].

Most recently developed face recognition systems work well when images are obtained under controlled conditions or when the test image is captured under similar conditions to those for the training images. However, under varying lighting or aging effects, their performance is still not satisfactory. To perform recognition under fairly uncontrolled conditions Tan and Triggs [14] proposed a preprocessing chain for illumination normalization. They used the local ternary patterns and a Hausdorff-like distance measure. Holappa [15] used local binary pattern texture features and proposed a filter optimization procedure for illumination normalization. Aggarwal [16] presented a physical model using Lambert's Law to generalize across varying situations. Shih [17] proposed a new color space LC_1C_2 as a linear transformation of the RGB color space.

Another challenge is that most current face recognition algorithms perform well when several training images are available per subject; however they are still not adequate for scenarios where a single sample per subject is available. In real world applications, one training sample per subject presents advantages such as ease of collect galleries, low cost for storage and lower computational cost [18]. Thus, a robust face recognition system able to work with both single and several samples per subject is desirable. In [19], Liu et al. proposed representing each single (training, testing) image as a subspace spanned by synthesized shifted images and designed a new subspace distance metric.

Regarding the scalability issues discussed previously, there is also previous work focused on scaling recognition systems to large datasets. In [20] a technique for combining rejection classifiers into a cascade is proposed to speed up the nearest neighbor search for face identification. Guo and Zhang [21] proposed the use of a constrained majority voting scheme for AdaBoost to reduce the number of comparisons needed.

3 Proposed Method

In this section, we first present the feature extraction process and a brief review of partial least squares regression. Then, the proposed face identification approach is explained in two steps. Initially, we describe the one-against-all approach, then we describe the tree-based structure, which improves scalability when the gallery is large and reduces the computational cost of matching probe samples.

3.1 Feature Extraction

After cropping and resizing the faces, each sample is decomposed into overlapping blocks and a set of low-level feature descriptors is extracted from each block. The features used include information related to shape (histogram of oriented gradients (HOG) [22]), texture (captured by local binary patterns (LBP) [13]), and color information (captured by averaging the intensities of pixels in a block).

HOG captures edge or gradient structures that are characteristic of local shape [22]. Since the histograms are computed for regions of a given size, HOG is robust to some location variability of face parts. HOG is also invariant to rotations smaller than the orientation bin size.

Local binary patterns [13] have been successfully applied in texture classification. LBP's characterize the spatial structure of the local image texture and are invariant under monotonic transformations of the pixel gray values. The LBP operator labels the pixels of an image by thresholding the 3 × 3 neighborhood of each pixel using the center value. A label is obtained by multiplication of the thresholded values by the binomial factors 2^p followed by their addition. The 256-bin histogram of the resulting labels is used as a feature descriptor.

Once the feature extraction process is performed for all blocks inside a cropped face, features are concatenated creating a high-dimensional feature vector $\boldsymbol{v}$. This vector is used to describe the face.

3.2 Partial Least Squares Regression

Partial least squares is a method for modeling relations between sets of observed variables by means of latent variables. PLS estimates new predictor variables, latent variables, as linear combinations of the original variables summarized in a matrix $\boldsymbol{X}$ of predictor variables (features) and a vector $\boldsymbol{y}$ of response variables. Detailed descriptions of the PLS method can be found in [23,24].

Let $\mathcal{X} \subset \mathbb{R}^m$ denote an m-dimensional feature space and let $\mathcal{Y} \subset \mathbb{R}$ be a 1-dimensional space of responses. Let the number of samples be n. PLS decomposes matrix $\boldsymbol{X}_{n\times m} \in \mathcal{X}$ and vector $\boldsymbol{y}_{n\times 1} \in \mathcal{Y}$ into

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E}$$
$$\boldsymbol{y} = \boldsymbol{U}\boldsymbol{q}^T + \boldsymbol{f}$$

where $\boldsymbol{T}$ and $\boldsymbol{U}$ are $n \times p$ matrices containing p extracted latent vectors, the $(m \times p)$ matrix $\boldsymbol{P}$ and the $(1 \times p)$ vector $\boldsymbol{q}$ represent the loadings and the $n \times m$ matrix $\boldsymbol{E}$ and the $n \times 1$ vector $\boldsymbol{f}$ are the residuals. Using the nonlinear iterative partial least squares (NIPALS) algorithm [7], a set of weight vectors is constructed, stored in the matrix $\boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_p)$, such that

$$[cov(\boldsymbol{t}_i, \boldsymbol{u}_i)]^2 = \max_{|\boldsymbol{w}_i|=1} [cov(\boldsymbol{X}\boldsymbol{w}_i, \boldsymbol{y})]^2 \tag{1}$$

where $\boldsymbol{t}_i$ is the i-th column of matrix $\boldsymbol{T}$, $\boldsymbol{u}_i$ the i-th column of matrix $\boldsymbol{U}$ and $cov(\boldsymbol{t}_i, \boldsymbol{u}_i)$ is the sample covariance between latent vectors $\boldsymbol{t}_i$ and $\boldsymbol{u}_i$. After extracting the latent vectors $\boldsymbol{t}_i$ and $\boldsymbol{u}_i$, the matrix $\boldsymbol{X}$ and vector $\boldsymbol{y}$ are deflated by subtracting their rank-one approximations based on $\boldsymbol{t}_i$ and $\boldsymbol{u}_i$. This process is repeated until the desired number of latent vectors has been extracted.

Once the low dimensional representation of the data has been obtained by NIPALS, the regression coefficients $\boldsymbol{\beta}_{m\times 1}$ can estimated by

$$\boldsymbol{\beta} = \boldsymbol{W}(\boldsymbol{P}^T\boldsymbol{W})^{-1}\boldsymbol{T}^T\boldsymbol{y}. \tag{2}$$

The regression response, y_v, for a feature vector $\boldsymbol{v}$ is obtained by

$$y_v = \overline{y} + \boldsymbol{\beta}^T\boldsymbol{v} \tag{3}$$

where $\overline{y}$ is the sample mean of $\boldsymbol{y}$.

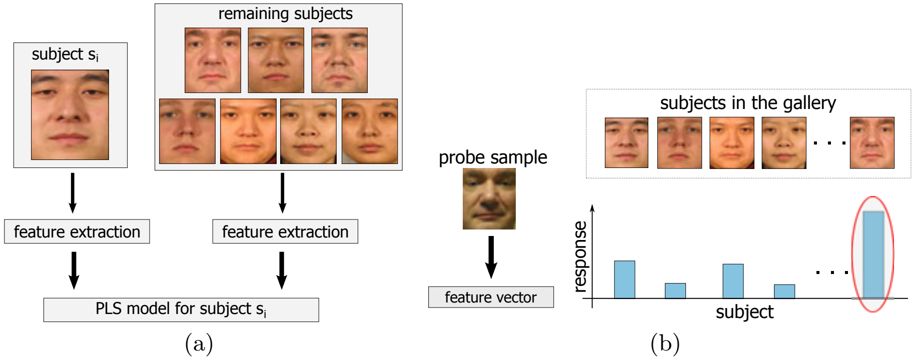


Fig. 1. One-against-all face identification approach. (a) construction of the PLS regression model for a subject in the gallery; (b) matching of a probe sample against the subjects in the gallery. The best match for a given probe sample is the one associated with the PLS model presenting the highest regression response.

Notice that even though the number of latent vectors used to create the low dimensional representation of the data matrix $\boldsymbol{X}$ is p (possibly greater than 1), Equation 3 shows that only a single dot product of a feature vector with the regression coefficients is needed to obtain the response of a PLS regression model – and it is this response that is used to rank faces in a gallery. This characteristic makes the use of PLS particularly fast for finding matches for novel probe samples, in contrast to other methods where the number of dot product evaluations depends on the number of eigenvectors considered, which is quite large in general [25].

3.3 One-Against-All Approach

The procedure to learn models for subjects in the gallery $g = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \dots, \boldsymbol{s}_n\}$, where $\boldsymbol{s}_i$ represents exemplars of each subject's face, is illustrated in Figure 1(a) and described in details as follows. Each $\boldsymbol{s}_i$ is composed of feature vectors extracted from cropped faces containing examples of the i-th subject.

We employ a one-against-all scheme to learn a PLS discriminatory model for each subject in the gallery. Therefore, when the i-th subject is considered, the remaining samples $g \setminus \boldsymbol{s}_i$ are used as counter-examples of the i-th subject. In addition, if the face dataset provides a training set we also add those samples, (excluding samples from the subject under consideration), as counter-examples of the i-th subject. Experiments show that the addition of training samples as counter-examples improves recognition rates.

When a one-against-all scheme is used with PLS, higher weights are attributed to features located in regions containing discriminatory characteristics between the subject under consideration and the remaining subjects.

Once the models have been estimated for all subjects in the gallery, the PLS regression models are stored to be later used to evaluate the responses for a probe

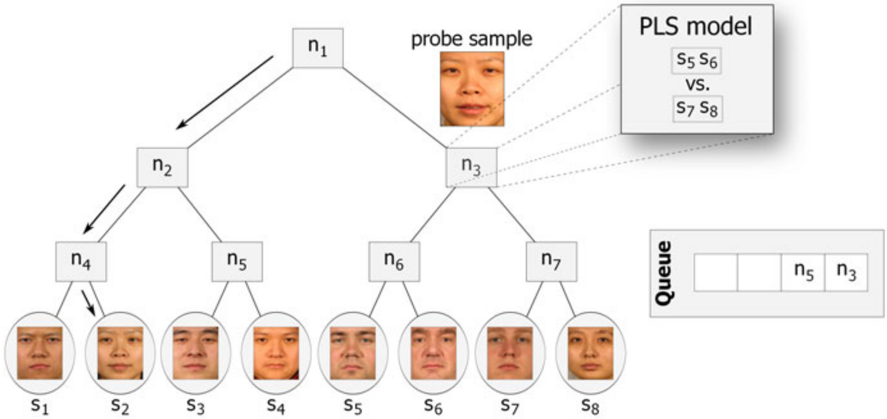


Fig. 2. Tree-based structure used to optimize the search for matches to a probe sample. Each internal node contains a PLS regression model used to guide the search, as shown in details for node n_3, which has a PLS model constructed so that the response directs the search either to node n_6 or n_7. In this example the first path to be traversed is indicated by arrows (in this case, it leads to the correct match for this particular probe sample). Alternative search paths are obtained by adding nodes that have not been visited into a priority queue (in this example nodes n_3 and n_5 will be the starting nodes for additional search paths). After pursuing a number of search paths leading to different leaf nodes, the best match is chosen to be the one presenting the highest response (in absolute value).

sample. Then, when a probe sample is presented, its feature vector is projected onto each one of the PLS models. The model presenting the highest regression response gives the best match for the probe sample, as illustrated in Figure 1(b).

3.4 Optimization Using a Tree-Based Structure

In terms of scalability, two drawbacks are present in the one-against-all scheme described in the previous section. First, when a new subject is added to the gallery, PLS models need to be rebuilt for all subjects. Second, to find the best match to a probe sample, the feature vector representing this sample needs to be projected onto all PLS models learned for the subjects in the gallery (common problem faced by methods that estimate matching scores using brute force nearest neighbor search [20]).

To reduce the need for projecting features onto all PLS models to find the best match for a probe sample, we construct a binary tree in which each node, n_j, contains a subset of the gallery subjects $t_j \subset g$, where $g = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n\}$ as defined previously. A splitting procedure is used to decide which elements of t_j will belong to the left and right children of n_j, assigning at least one sample to each child. Each internal node is associated with a PLS regression model, used afterwards to guide the search when probe samples are analyzed. In order to build the regression model for a node, the subjects assigned to the left child are defined to have response -1 and the subjects assigned to the right child are defined to have response $+1$. The splitting procedure and the building of PLS models are applied recursively in the tree until a node contains only a single subject (leaf node).

The application of the described procedure for a gallery with n subjects results in a tree containing n leaf nodes and $n-1$ PLS regression models located on the internal nodes.

We consider two approaches to split subjects between the children nodes. First, a procedure that uses PCA to create a low dimensional subspace (learned using samples from a training set) and then the K-means algorithm clusters data into two groups, each one is assigned to one child. The second approach chooses random splits and divides the subjects equally into two groups. We evaluate these splitting procedures in Section 4.3.

When a feature vector describing a probe sample is analyzed to find its best matching subject in the gallery, a search starting from the root of the tree is performed. At each internal node, the feature vector is projected onto the PLS model and according to its response, the search continues either from the left or from the right child. The search stops when a leaf node is reached. Figure 2 illustrates this procedure.

According to experimental results shown in Section 4.3, the traversal of a few search paths is enough to obtain the best match for a probe sample. Starting nodes for alternative search paths are stored in a priority queue. An internal node n_k is pushed into the priority queue when its sibling is chosen to be in the current search path. The priority associated with n_k is proportional to its response returned by the PLS regression model at its parent. Finally, since each search path leads to a leaf node, the best match for a given probe sample is chosen to be the one presenting the highest response (in absolute value) among the leaf nodes reached during the search.

The tree-based structure can also be used to avoid rebuilding all PLS models when a new subject is added into the gallery. Assuming that a tree is built for k subjects, the procedure to add a new subject $\boldsymbol{s}_{k+1}$ is described as follows. Choose a leaf node n_i, where $t_i = \{\boldsymbol{s}_j\}$; set n_i to be an internal node and create two new leaf nodes to store $\boldsymbol{s}_j$ and $\boldsymbol{s}_{k+1}$; then, build a PLS model for node n_i (now with $t_i = \{\boldsymbol{s}_j, \boldsymbol{s}_{k+1}\}$). Finally, rebuild all PLS models in nodes having n_i as a descendant. Therefore, using this procedure, the number of PLS models that needs to be rebuilt when a new subject is added no longer depends on the number of subjects in the gallery, but only on the depth of node n_i.

4 Experiments

In this section we evaluate several aspects of our proposed approach. Initially, we show that the use of the low-level feature descriptors analyzed by PLS in a one-against-all scheme, as described in Section 3.3, improves recognition rates over previous approaches, particularly when the data is acquired under uncontrolled conditions. Then, we demonstrate that the tree-based approach introduced in Section 3.4 obtains comparably high recognition rates with a significant reduction in the number of projections.

The method is evaluated on two standard datasets used for face recognition: FERET and FRGC version 1. The main characteristics of the FERET dataset are that it contains a large number of subjects in the gallery and the probe sets exploit differences in illumination, facial expression variations, and aging effects [26]. FRGC contains faces acquired under uncontrolled conditions [27].

All experiments were conducted on an Intel Core i7-860 processor, 2.8 GHz with 4GB of RAM running Windows 7 operating system using a single processor core. The method was implemented using C++ programming language.

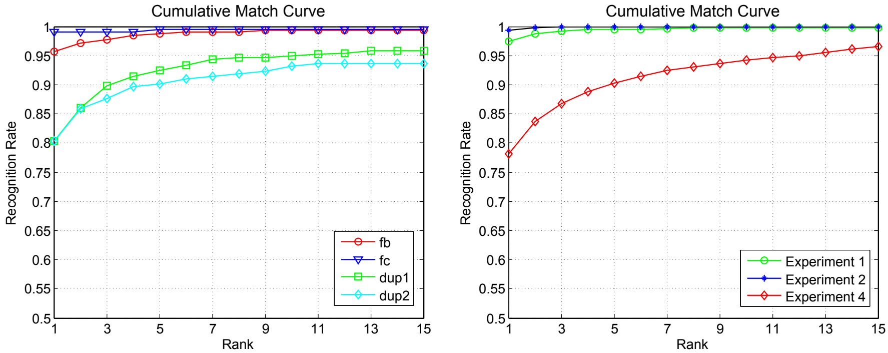
4.1 Evaluation on the FERET Dataset

The frontal faces in the FERET database are divided into five sets: fa (1196 images, used as gallery set containing one image per person), fb (1195 images, taken with different expressions), fc (194 images, taken under different lighting conditions), $dup1$ (722 images, taken at a later date), and $dup2$ (234 images, taken at least one year apart). Among these four standard probe sets, $dup1$ and $dup2$ are considered the most difficult since they are taken with time-gaps, so some facial features have changed. The images are cropped and rescaled to 110×110 pixels.

Experimental Setup. Since the FERET dataset is taken under varying illumination conditions, we preprocessed the images for illumination normalization. Among the best known illumination normalization methods are the self-quotient image (SQI) [28], total variation models, and anisotropic smoothing [15]. SQI is a retinex-based method which does not require training images and has relatively low computational complexity; we use it due to its simplicity. Once the images are normalized, we perform feature extraction. For HOG features we use block sizes of 16×16 and 32×32 with strides of 4 and 8 pixels, respectively. For LBP features we use block size of 32×32 with a stride of 16 pixels. The mean features are computed from block size of 4×4 with stride of 2 pixels. This results in feature vectors with $35,680$ dimensions.

To evaluate how the method performs using information extracted exclusively from a single image per subject, in this experiment we do not add samples from the training set as counter-examples. The training set is commonly used to build a subspace to obtain a low dimensional representation of the features before performing the match. This subspace provides additional information regarding the domain of the problem.

Results and Comparisons. Figure 3(a) shows the cumulative match curves obtained by the one-against-all approach for all FERET probe sets. We see that our method is robust to facial expressions (fb), lighting (fc) and aging effect ($dup1$, $dup2$). The computational time to learn the gallery models is 4519 s and the average time to evaluate a pair of probe-gallery samples is 0.34 ms.



(a) FERET (b) FRGC

Fig. 3. The cumulative match curve for the top 15 matches obtained by the one-against-all approach based on PLS regression for FERET and FRGC datasets

Table [1] shows the rank-1 recognition rates of previously published algorithms and ours on the FERET dataset. As shown in the table, the one-against-all approach achieves similar results on fb and fc without using the training set. Additionally, our results on the challenging $dup1$ and $dup2$ sets are over 80%.

Table 1. Recognition rates of the one-against-all proposed identification method compared to algorithms for the FERET probe sets

	Method	**fb**	**fc**	**dup1**	**dup2**
using training set	Best result of [26]	95.0	82.0	59.0	52.0
	LBP [13]	97.0	79.0	66.0	64.0
	Tan [6]	98.0	98.0	90.0	85.0
not using training set	LGBPHS [11]	98.0	97.0	74.0	71.0
	HGPP [12]	97.6	98.9	77.7	76.1
	SIS [19]	91.0	90.0	68.0	68.0
	Ours	95.7	99.0	80.3	80.3

4.2 Evaluation on the FRGC Dataset

We evaluate our method using three experiments of FRGC version 1 that consider 2D images. Experiment 1 contains a single controlled probe image and a gallery with one controlled still image per subject (183 training images, 152 gallery images, and 608 probe images). Experiment 2 considers identification of a person given a gallery with four controlled still images per subject (732 training images, 608 gallery images, and 2432 probe images). Finally, experiment 4 considers a single uncontrolled probe image and a gallery with one controlled still image per subject (366 training images, 152 gallery images, and 608 probe

Table 2. Recognition rates of the one-against-all proposed identification method compared to other algorithms for the FRGC probe sets

Method	**Exp.1**	**Exp.2**	**Exp.4**
UMD [16]	94.2	99.3	-
LC_1C_2 [17]	-	-	75.0
Tan (from [15])	-	-	58.1
Holappa [15]	-	-	63.7
Ours	97.5	99.4	78.2

images). We strictly followed the published protocols. The images are cropped and rescaled to 275 × 320 pixels.

Experimental Setup. FRGC images are larger than FERET; thus we have chosen larger block sizes and strides to avoid computing too many features. For HOG features we use block sizes of 32 × 32 with strides of 8 pixels. For LBP features we use block size of 32 × 32 with strides of 24 pixels. And the mean features are extracted from block sizes of 8 × 8 with a stride of 4 pixels. This results in feature vectors with $86,634$ dimensions.

Experiment 4 in FRGC version 1 is considered the most challenging in this dataset. Since it is hard to recognize uncontrolled faces directly from the gallery set consisting of controlled images, we attempted to make additional use of the training set to create some *uncontrolled environment information* using morphed images. Morphing can generate images with reduced resemblance to the imaged person or look-alikes of the imaged person [29]. The idea is to first compute a *mean face* from the uncontrolled images in the training set. Then, we perform triangulation-based morphing from the original gallery set to this mean face by 20%, 30%, 40%. This generates three synthesized images. Therefore, for each subject in the gallery we now have four samples.

Results and Comparisons. Figure 3(b) shows the cumulative match curves obtained by the one-against-all approach for the three probe sets of FRGC. In addition, the computational time to learn gallery models is 410.28 s for experiment 1, 1514.14 s for experiment 2, and 1114.39 s for experiment 4. The average time to evaluate a pair of probe-gallery samples is 0.61 ms.

Table 2 shows the rank-1 recognition rates of different algorithms on the FRGC probe sets. Our method outperforms others in every probe set considered, especially on the most challenging experiment 4. This is, to the best of our knowledge, the best performance reported in the literature.

4.3 Evaluation of the Tree-Based Structure

In this section we evaluate the tree-based structure described in Section 3.4. First, we evaluate procedures used to split the set of subjects belonging to a node. Second, we test heuristics used to reduce the search space. Third, we compare the

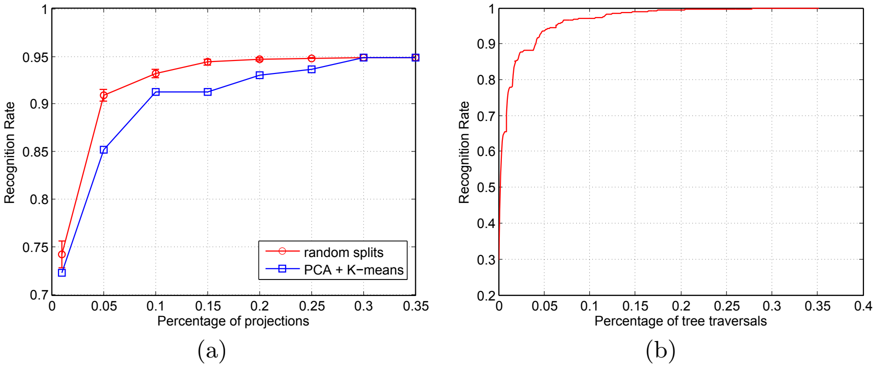


Fig. 4. Evaluation of the tree-based approach. (a) comparison of the recognition rates when random splits and PCA+K-Means approach are used; (b) evaluation of the heuristic based on stopping the search after a maximum number of tree traversals is reached.

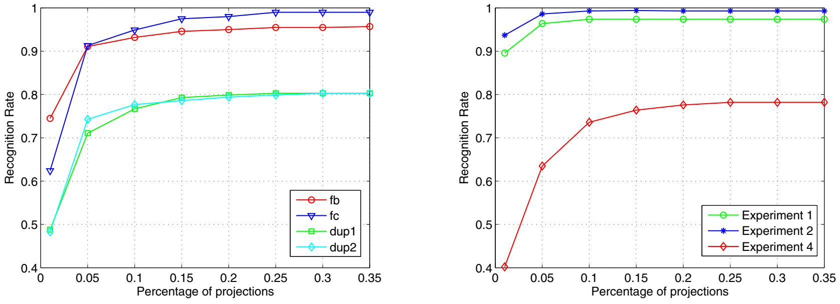
results obtained previously by the one-against-all approach to results obtained when the tree-based structure is incorporated. Finally, we compare our method to the approach proposed by Yuan et al. [20].

To evaluate the reduction in the number of comparisons, in this section the x-axis of the plots no longer displays the rank; instead it shows either the percentage of projections performed by the tree-based approach when compared to the one-against-all approach (e.g. Figure 4(a)) or the percentage of tree traversals when compared to the number of subjects in the gallery (e.g. Figure 4(b)). The y-axis displays the recognition rates for the rank-1 matches. We used probe set *fb* from the FERET dataset to perform evaluations in this section.

Procedure to Split Nodes. Figure 4(a) shows that both splitting procedures described in Section 3.4 obtain similar recognition rates when the same number of projections is performed. The error bars (in Figure 4(a)) show the standard deviation of the recognition rates obtained using random splits. They are very low and negligible when the percentage of projections increases. Due to the similarity of the results, we have chosen to split the nodes randomly. The advantages of applying random splits are the lower computational cost to build the gallery models and balanced trees are obtained. Balanced trees are important since the depth of a leaf node is proportional to $\lg n$, which is desirable to keep short search paths.

Heuristics to Reduce the Search Space. The first experiment evaluates the recognition rate as a function of the maximum number of traversals allowed to find the match subject to a probe sample; this is limited to a percentage of the gallery size. Figure 4(b) shows the maximum recognition rates achievable for a given percentage. We can see that as low as 15% of traversals are enough to obtain recognition rates comparable to the results obtained by the one-against-all approach (95.7% for the probe set considered in this experiment).

In the second experiment we consider the following heuristic. For the initial few probe samples, all search paths are evaluated and the absolute values of the



(a) FERET (b) FRGC

Fig. 5. Recognition rates as a function of the percentage of projections performed by the tree-based approach when compared to the one-against-all approach

regression responses for the best matches are stored. The median of these values is computed. Then, for the remaining probe samples, the search is stopped when the regression response for a leaf node is higher than the estimated median value. Our experiments show that this heuristic alone is able to reduce the number of projections to 63% without any degradation in the recognition rates.

Results and Comparisons. Using the results obtained from the previous experiments (random splits and adding both heuristics to reduce the search space), we now compare the recognition rates obtained when the tree-based structure is used to results obtained by the one-against-all approach. Then, we evaluate the speed-up achieved by reducing the number of projections.

Figures 5(a) and 5(b) show identification results obtained for FERET and FRGC datasets, respectively. Overall, we see that when the number of projections required by the one-against-all approach is reduced to 20% or 30%, there is a negligible drop in the recognition rate shown in the previous sections. Therefore, without decreasing the recognition rate, the use of the tree-based structure provides a clear speed-up for performing the evaluation of the probe set. According to the plots, speed-ups of 4 times are achieved for FERET, and for FRGC the speed-up is up to 10 times depending on the experiment being considered.

Finally, we compare our method to the *cascade of rejection classifiers* (CRC) approach proposed by Yuan et al. [20]. Table 3 shows the speed-ups over the

Table 3. Comparison between our tree-based approach and the CRC approach

	test set size as fraction of dataset	**10%**	**21%**	**32%**	**43%**	**65%**
CRC	speed-up	1.58	1.58	1.60	2.38	3.35
	rank-1 error rate	19.5%	22.3%	24.3%	28.7%	42.0%
Ours	speed-up	3.68	3.64	3.73	3.72	3.80
	rank-1 error rate	5.62%	5.08%	5.70%	5.54%	5.54%

brute force nearest neighbor search and rank-1 error rates obtained by both approaches. We apply the same protocol used in [20] for the FRGC dataset. Higher speed-ups are obtained by our method and, differently from CRC, no increase in the error rates is noticed when larger test set sizes are considered.

5 Conclusions

We have proposed a face identification method using a set of low-level feature descriptors analyzed by PLS which presents the advantages of being both robust and scalable. Experimental results have shown that the method works well for single image per sample, in large galleries, and under different conditions.

The use of PLS regression makes the evaluation of probe-gallery samples very fast due to the necessity of only a single dot product evaluation. Optimization is further improved by incorporating the tree-based structure, which reduces largely the number of projections when compared to the one-against-all approach, with negligible effect on recognition rates.

Acknowledgments

This research was funded by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the Army Research Laboratory (ARL). All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI, or the U.S. Government.

References

1. Phillips, P.J., Micheals, P.J., Blackburn, R.J., Tabassi, D.M., Bone, J.M.: Face Recognition vendor test 2002: Evaluation Report. Technical report, NIST (2003)
2. Tolba, A., El-Baz, A., El-Harby, A.: Face recognition: A literature review. International Journal of Signal Processing 2, 88–103 (2006)
3. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Comput. Surv. 35, 399–458 (2003)
4. Wu, B., Nevatia, R.: Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. In: CVPR, pp. 1–8 (2008)
5. Schwartz, W.R., Kembhavi, A., Harwood, D., Davis, L.S.: Human detection using partial least squares analysis. In: ICCV (2009)
6. Tan, X., Triggs, B.: Fusing Gabor and LBP feature sets for kernel-based face recognition. In: Zhou, S.K., Zhao, W., Tang, X., Gong, S. (eds.) AMFG 2007. LNCS, vol. 4778, pp. 235–249. Springer, Heidelberg (2007)
7. Wold, H.: Partial least squares. In: Kotz, S., Johnson, N. (eds.) Encyclopedia of Statistical Sciences, vol. 6, pp. 581–591. Wiley, New York (1985)
8. Schwartz, W.R., Davis, L.S.: Learning discriminative appearance-based models using partial least squares. In: SIBGRAPI (2009)
9. Dhanjal, C., Gunn, S., Shawe-Taylor, J.: Efficient sparse kernel feature extraction based on partial least squares. TPAMI 31, 1347–1361 (2009)

10. Zou, J., Ji, Q., Nagy, G.: A comparative study of local matching approach for face recognition. IEEE Transactions on Image Processing 16, 2617–2628 (2007)
11. Zhang, W., Shan, S., Gao, W., Chen, X., Zhang, H.: Local gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition. In: ICCV 2005, pp. 786–791 (2005)
12. Zhang, B., Shan, S., Chen, X., Gao, W.: Histogram of gabor phase patterns (HGPP): A novel object representation approach for face recognition. IEEE Transactions on Image Processing 16, 57–68 (2007)
13. Ahonen, T., Hadid, A., Pietikainen, M.: Face recognition with local binary patterns. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 469–481. Springer, Heidelberg (2004)
14. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. In: Zhou, S.K., Zhao, W., Tang, X., Gong, S. (eds.) AMFG 2007. LNCS, vol. 4778, pp. 168–182. Springer, Heidelberg (2007)
15. Holappa, J., Ahonen, T., Pietikinen, M.: An optimized illumination normalization method for face recognition. In: IEEE International Conference on Biometrics: Theory, Applications and Systems, pp. 6–11 (2008)
16. Aggarwal, G., Biswas, S., Chellappa, R.: UMD experiments with FRGC data. In: CVPR Workshop, pp. 172–178 (2005)
17. Shih, P., Liu, C.: Evolving effective color features for improving FRGC baseline performance. In: CVPR Workshop, pp. 156–163 (2005)
18. Tan, X., Chen, S., Zhou, Z., Zhang, F.: Face recognition from a single image per person: A survey. Pattern Recognition 39, 1725–1745 (2006)
19. Liu, J., Chen, S., Zhou, Z., Tan, X.: Single image subspace for face recognition. In: Zhou, S.K., Zhao, W., Tang, X., Gong, S. (eds.) AMFG 2007. LNCS, vol. 4778, pp. 205–219. Springer, Heidelberg (2007)
20. Yuan, Q., Thangali, A., Sclaroff, S.: Face identification by a cascade of rejection classifiers. In: CVPR Workshop, pp. 152–159 (2005)
21. Guo, G.D., Zhang, H.J.: Boosting for fast face recognition. In: ICCV Workshop. IEEE Computer Society, Los Alamitos (2001)
22. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR, pp. 886–893 (2005)
23. Elden, L.: Partial least-squares vs. Lanczos bidiagonalization–I: analysis of a projection method for multiple regression. Computational Statistics & Data Analysis 46, 11–31 (2004)
24. Rosipal, R., Kramer, N.: Overview and recent advances in partial least squares. In: Saunders, C., Grobelnik, M., Gunn, S., Shawe-Taylor, J. (eds.) SLSFS 2005. LNCS, vol. 3940, pp. 34–51. Springer, Heidelberg (2006)
25. Delac, K., Grgic, M., Grgic, S.: Independent comparative study of PCA, ICA, and LDA on the FERET data set. International Journal of Imaging Systems and Technology 15, 252–260 (2005)
26. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. TPAMI 22, 1090–1104 (2000)
27. Phillips, P.J., Flynn, P.J., Scruggs, T., Bowyer, K.W., Chang, J., Hoffman, K., Marques, J., Min, J., Worek, W.: Overview of the face recognition grand challenge. In: CVPR, pp. 947–954 (2005)
28. Wang, H., Li, S.Z., Wang, Y.: Face recognition under varying lighting conditions using self quotient image. In: IEEE International Conference on Automatic Face and Gesture Recognition, pp. 819–824 (2004)
29. Kamgar Parsi, B., Lawson, E., Baker, P.: Toward a human-like approach to face recognition. In: BTAS, pp. 1–6 (2007)

Emotion Recognition from Arbitrary View Facial Images

Wenming Zheng[1], Hao Tang[2], Zhouchen Lin[3], and Thomas S. Huang[2]

[1] Research Center for Learning Science, Southeast University, Nanjing 210096, China
wenming_zheng@seu.edu.cn

[2] Beckman Institute, University of Illinois at Urbana-Champaign, USA
haotang2@uiuc.edu, huang@ifp.uiuc.edu

[3] Visual Computing Group, Microsoft Research Asia, China
zhoulin@microsoft.com

Abstract. Emotion recognition from facial images is a very active research topic in human computer interaction (HCI). However, most of the previous approaches only focus on the frontal or nearly frontal view facial images. In contrast to the frontal/nearly-frontal view images, emotion recognition from non-frontal view or even arbitrary view facial images is much more difficult yet of more practical utility. To handle the emotion recognition problem from arbitrary view facial images, in this paper we propose a novel method based on the regional covariance matrix (RCM) representation of facial images. We also develop a new discriminant analysis theory, aiming at reducing the dimensionality of the facial feature vectors while preserving the most discriminative information, by minimizing an estimated multiclass Bayes error derived under the Gaussian mixture model (GMM). We further propose an efficient algorithm to solve the optimal discriminant vectors of the proposed discriminant analysis method. We render thousands of multi-view 2D facial images from the BU-3DFE database and conduct extensive experiments on the generated database to demonstrate the effectiveness of the proposed method. It is worth noting that our method does not require face alignment or facial landmark points localization, making it very attractive.

1 Introduction

The research on human's emotion can be traced back to the Darwin's pioneer work in [1] and since then has attracted a lot of researchers to this area. According to Ekman et al. [2], there are six basic emotions that are universal to human beings, namely, angry (AN), disgust (DI), fear (FE), happy (HA), sad (SA), and surprise (SU), and these basic emotions can be recognized from human's facial expression. Nowadays, the recognition of these six basic emotions from human's facial expressions has become a very active research topic in human computer interaction (HCI). During the past decades, various methods have been proposed for emotion recognition. One may refer to [3][4][5][6] for a survey.

Although emotion recognition has been extensively explored in the past decades, most of the previous approaches focus on the frontal or nearly frontal view facial

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 490–503, 2010.

images. But actually emotion recognition from non-frontal view or even arbitrary view facial images is of more practical utility. However, recognizing the non-frontal view emotions is very difficult. To the best of our knowledge, only a few papers address this issue [7][8][9][10][11][12][14]. In [12], Hu et al. investigated the facial expression recognition problem on a set of images with five yaw views, i.e., 0^o, 30^o, 45^o, 60^o, and 90^o, which are generated from the BU-3DFE database [13]. They used the geometric features defined on the landmark points around the eyes, eye-brow and mouth to represent the face images and then conducted the emotion recognition with various classifiers. Instead of using geometric features, Zheng et al. [14] used sparse SIFT features [15] extracted at 83 landmark points to represent the facial images. They also proposed a novel feature extraction method, based on an upper bound of the multi-class Bayes error under the Gaussian assumption, to reduce the dimensionality of the feature vectors. However, a common limitation of both methods is that the landmark points are known apriori from the original 3D face models. This may severely limit their practical applications, where no 3D face model is available. Moreover, the effectiveness of both methods is only evaluated using facial images in limited views, i.e., five yaw views. In practice, one may encounter much more different views in emotion recognition. In addition, the assumption of Gaussian distribution for each emotion category in [14] may not suffice for the true distributions of the data.

In this paper, we address the emotion recognition problem from arbitrary view facial images. To this end, we propose a novel facial image representation method, which enables us to avoid the face alignment or facial feature localization. The basic idea of the proposed image representation method is to use the region covariance matrix (RCM) [16][17] of the facial region. More specifically, we first detect the facial region from a given facial image [18], then extract a set of dense SIFT feature vectors from each facial image. The concept of dense SIFT feature vectors is illustrated in Fig.1, where the whole facial region is divided into some patches, and at the center of each patch we extract a 128-dimensional SIFT feature vector. The RCM of the facial region is then obtained by computing the covariance of the SIFT vectors. However, it should be noted that, as the dimensionality of the SIFT vectors is 128, the number of entries to be estimated in RCM may be much larger than the number of SIFT feature vectors extracted from each facial image. On the other hand, since the SIFT features are

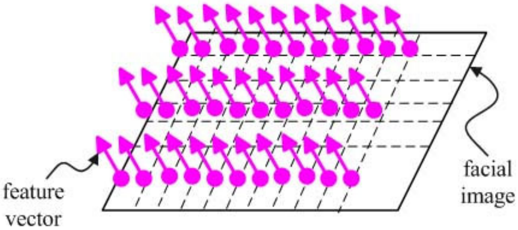


Fig. 1. The whole facial region is divided into some patches, and each patch produces a SIFT feature vector

extracted from arbitrary view facial images, they may carry much information that are irrelevant to the emotion recognition. Therefore, extracting the most discriminative features from the raw SIFT feature vectors is advantageous and necessary for improving the recognition performance.

Recall that in [14], Zheng et al. propose a discriminative feature extraction method based on an estimated Bayes error using the Gaussian distributions. However, when the samples, i.e., the SIFT feature vectors, are extracted from arbitrary view facial images, only a single Gaussian may not be enough to accurately model the distribution of the samples. To accurately model the distribution of each basic emotion class, in this paper we instead use mixtures of Gaussians, rather than a single Gaussian. The Gaussian mixture model (GMM) can be obtained via the expectation-maximization (EM) algorithm [19]. Under the GMM model, we derive a new upper bound of the multi-class Bayes error. Based on this upper bound, we develop a new discriminant analysis method, hereafter called the Bayes discriminant analysis via GMM (BDA/GMM), to reduce the dimensionality of the SIFT feature vectors while preserving the most discriminative information. Moreover, we also propose an efficient algorithm to solve for the optimal discriminant vectors of BDA/GMM.

The rest of this paper is organized as follows. In section 2, we describe the feature representation method. In section 3, we propose our BDA/GMM method. In section 4, we present an efficient algorithm for BDA/GMM. In section 5, we show the emotion classification. The experiments are presented in section 6. Finally section 7 concludes our paper.

2 Feature Representation

2.1 SIFT Feature Descriptor

In [14], Zheng et al. extracted a set of SIFT features at 83 pre-defined landmark points to describe a facial image. Then they concatenated the SIFT features to represent the image and perform classification. Their experiments demonstrated the effectiveness of SIFT features for emotion recognition. In practical applications, however, automatically locating the landmark points from arbitrary view facial image is very challenging. To overcome this problem, we use the so-called dense SIFT features description method illustrated in Fig.1 to describe the facial image, which does not need the face alignment and facial landmark points localization. More specifically, we divide the whole facial region into a set of patches. Then, we extract 128-dimensional SIFT features at the center of each patch. These features are finally used for the calculation of RCM.

2.2 RCM for Facial Image Representation

RCM was originally proposed for image representation and had been successfully applied to face detection, texture recognition, and pedestrian detection [16][17]. RCM can not only capture the statistical properties of the samples, but also be invariant to the image translation, scale and rotation changes. On the other hand, for emotion recognition we may need to integrate the SIFT feature vectors of each image to form a data point and then conduct the classification. Based on the above analysis, we use RCM to represent each facial image in this paper.

However, it should be noted that the entry number of RCM is proportional to the squared dimensionality of the SIFT feature vectors. For example, in this paper the dimensionality of the raw SIFT feature vectors is 128, resulting in (128×128+128)/2=8256 entries to be estimated in RCM. However, the number of SIFT vectors we extract from each facial image is about 450, which is much less than the number of parameters to be estimated in RCM. On the other hand, considering that the SIFT features are extracted from arbitrary view facial images, they may contain much information irrelevant to the emotion recognition. So it will be advantageous and necessary to reduce the dimensionality of the SIFT feature vectors before using the RCM representation. In the next section, we will propose a novel discriminant analysis theory aiming at reducing the dimensionality of the facial feature vectors while preserving the most discriminative information.

3 BDA/GMM: Bayes Discriminant Analysis via Gaussian Mixture Model

In this section, we propose the BDA/GMM method for dimensionality reduction. Let $\mathbf{X}_i = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \cdots, \mathbf{x}_{i,N_i}\} \in \mathbb{R}^d$ $(i = 1, 2, \cdots, c)$ denote the ith class data set, where $\mathbf{x}_{i,j}$ represents the j-th sample of the i-th class, N_i is the number of samples in the i-th class, and c denotes the number of classes.

3.1 Gaussian Mixture Model

Let $p_i(\mathbf{x}|\mathbf{x} \in \mathbf{X}_i)$ denote the class distribution function of $\mathbf{X}_i$. Then the GMM of $p_i(\mathbf{x}|\mathbf{x} \in \mathbf{X}_i)$ can be expressed as follows:

$$p_i(\mathbf{x}|\mathbf{x} \in \mathbf{X}_i) = \sum_{r=1}^{K_i} \pi_{i,r} \mathcal{N}(\mathbf{x}|\mathbf{m}_{i,r}, \mathbf{\Sigma}_{i,r}), \tag{1}$$

where each Gaussian density

$$\mathcal{N}(\mathbf{x}|\mathbf{m}_{i,r}, \mathbf{\Sigma}_{i,r}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{\Sigma}_{i,r}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m}_{i,r})^T \mathbf{\Sigma}_{i,r}^{-1} (\mathbf{x} - \mathbf{m}_{i,r}) \right\},$$

is called a Gaussian mixture component, the parameters $\pi_{i,r}$ $(0 \leq \pi_{i,r} \leq 1$ and $\sum_{r=1}^{K_i} \pi_{i,r} = 1)$ are called the mixing coefficients, and K_i is the number of Gaussian mixture components. The parameters $\pi_{i,r}$, $\mathbf{m}_{ir}$, and $\mathbf{\Sigma}_{i,r}$ of the GMM in (1) can be estimated via the EM algorithm [19].

3.2 An Upper Bound of Two-Class Bayes Error

Let $p_k(\mathbf{x}|\mathbf{x} \in \mathbf{X}_k)$ and P_k be the class distribution density function and the prior probability of the k-th class, respectively. Then the Bayes error between the i-th class and the j-th class can be expressed as [21]:

$$\varepsilon = \int \min\left\{P_i p_i(\mathbf{x}|\mathbf{x}\in\mathbf{X}_i), P_j p_j(\mathbf{x}|\mathbf{x}\in\mathbf{X}_j)\right\} d\mathbf{x}. \tag{2}$$

Let $\hat{\pi}_{k,q} = P_k \pi_{k,q}$ and $\mathcal{N}_{k,q} = \mathcal{N}(\mathbf{x}|\mathbf{m}_{k,q}, \mathbf{\Sigma}_{k,q})$. Then from (1) we have

$$\begin{aligned} &\min\left\{P_i p_i(\mathbf{x}|\mathbf{x}\in\mathbf{X}_i), P_j p_j(\mathbf{x}|\mathbf{x}\in\mathbf{X}_j)\right\} \\ =& \min\left\{\sum_{r=1}^{K_i} \hat{\pi}_{i,r}\mathcal{N}_{i,r}, \sum_{l=1}^{K_j} \hat{\pi}_{j,l}\mathcal{N}_{j,l}\right\} \le \sum_r \min\left\{\hat{\pi}_{i,r}\mathcal{N}_{i,r}, \sum_{l=1}^{K_j} \hat{\pi}_{j,l}\mathcal{N}_{j,l}\right\} \\ \le& \sum_r\sum_l \min\left\{\hat{\pi}_{i,r}\mathcal{N}_{i,r}, \hat{\pi}_{j,l}\mathcal{N}_{j,l}\right\} \le \sum_r\sum_l \sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}\mathcal{N}_{i,r}\mathcal{N}_{j,l}}, \end{aligned} \tag{3}$$

where we have used the inequality $\min(a,b) \le \sqrt{ab}$, $\forall a, b \ge 0$ in the last inequality of (3). By substituting (3) into (2), we have the following upper bound of the Bayes error [21]:

$$\varepsilon \le \varepsilon_{ij} = \sum_r\sum_l \sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}} \exp\left(-D_{i,j}^{r,l}\right), \tag{4}$$

where

$$D_{i,j}^{r,l} = \frac{1}{8}(\mathbf{m}_{i,r} - \mathbf{m}_{j,l})^T (\bar{\mathbf{\Sigma}}_{i,j}^{r,l})^{-1} (\mathbf{m}_{i,r} - \mathbf{m}_{j,l}) + \frac{1}{2}\ln\frac{|\bar{\mathbf{\Sigma}}_{i,j}^{r,l}|}{\sqrt{|\mathbf{\Sigma}_{i,r}||\mathbf{\Sigma}_{j,l}|}}, \tag{5}$$

in which $\bar{\mathbf{\Sigma}}_{i,j}^{r,l} = \frac{1}{2}(\mathbf{\Sigma}_{i,r} + \mathbf{\Sigma}_{j,l})$.

Project $\mathbf{x}$ onto a line in direction $\omega \in \mathbb{R}^d$, then the following theorem holds:

Theorem 1. Let $p_i(\mathbf{x}|\mathbf{x}\in\mathbf{X}_i)$ expressed in (1) denote the distribution function of the i-th class. Then the class distribution function $\tilde{p}_i(\omega^T\mathbf{x}|\mathbf{x}\in\mathbf{X}_i)$ of the projected samples $\omega^T\mathbf{x}$ is also a mixture of Gaussians:

$$\tilde{p}_i(\omega^T\mathbf{x}|\mathbf{x}\in\mathbf{X}_i) = \sum_{r=1}^{K_i} \pi_{i,r}\mathcal{N}(\omega^T\mathbf{x}|\omega^T\mathbf{m}_{i,r}, \omega^T\mathbf{\Sigma}_{i,r}\omega). \tag{6}$$

Proof: See supplementary materials. □

From Theorem 1, equation (5) becomes

$$\tilde{D}_{i,j}^{r,l} = \frac{1}{8}\frac{\left[\omega^T(\mathbf{m}_{i,r} - \mathbf{m}_{j,l})\right]^2}{\omega^T\bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega} + \frac{1}{2}\ln\frac{\omega^T\bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}{\sqrt{(\omega^T\mathbf{\Sigma}_{i,r}\omega)(\omega^T\mathbf{\Sigma}_{j,l}\omega)}}, \tag{7}$$

and the upper bound of the Bayes error in (4) becomes

$$\varepsilon_{ij} = \sum_r \sum_l \sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}} \left(\frac{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l} \omega}{\sqrt{(\omega^T \mathbf{\Sigma}_{i,r}\omega)(\omega^T \mathbf{\Sigma}_{j,l}\omega)}} \right)^{-\frac{1}{2}} \exp\left\{ -\frac{1}{8} \frac{[\omega^T(\mathbf{m}_{i,r} - \mathbf{m}_{j,l})]^2}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega} \right\}. \tag{8}$$

To find a useful upper bound of ε_{ij}, we introduce the following two lemmas:

Lemma 1. Let $f(x) = (1-x^2)^{\frac{1}{4}}$ $(0 \le x \le 1)$. Then $\hat{f}(x) = \left(\frac{3}{4}\right)^{\frac{1}{4}} \left(\frac{7}{6} - \frac{1}{3}x\right)$ $(0 \le x \le 1)$ is the tightest *linear* upper bound of $f(x)$ in the sense that the total gap $\int_0^1 [\hat{f}(x) - f(x)]\mathrm{d}x$ between them is minimum.

Proof: See supplementary materials. □

Lemma 2. Let $h(x) = \exp(-x)$ $(0 \le x \le a)$. Then $\hat{h}(x) = 1 - \dfrac{1-\exp(-a)}{a}x$ $(0 \le x \le a)$ is the tightest *linear* upper bound of $h(x)$.

Proof: $h(x)$ is a convex function on the interval $[0, a]$. So the linear function passing through its two ends, $(0, h(0))$ and $(a, h(a))$, is the tightest linear upper bound of $h(x)$. This function is $\hat{h}(x)$. □

From Lemmas 1 and 2, we have:

$$\left(\frac{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l} \omega}{\sqrt{(\omega^T \mathbf{\Sigma}_{i,r}\omega)(\omega^T \mathbf{\Sigma}_{j,l}\omega)}} \right)^{-\frac{1}{2}} \le A_0 - A_1 \frac{|\omega^T \varDelta\mathbf{\Sigma}_{i,j}^{r,l}\omega|}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}, \tag{9}$$

$$\exp\left\{ -\frac{1}{8} \frac{[\omega^T(\mathbf{m}_{i,r} - \mathbf{m}_{j,l})]^2}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega} \right\} \le 1 - B_{ij} \frac{[\omega^T(\mathbf{m}_{i,r} - \mathbf{m}_{j,l})]^2}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}, \tag{10}$$

where $A_0 = \left(\frac{3}{4}\right)^{\frac{1}{4}} \frac{7}{6}$, $A_1 = \left(\frac{3}{4}\right)^{\frac{1}{4}} \frac{1}{3}$, $\varDelta\mathbf{\Sigma}_{i,j}^{r,l} = \frac{\mathbf{\Sigma}_{i,r} - \mathbf{\Sigma}_{j,l}}{2}$, and $B_{ij} = \frac{1-e^{\ \lambda_{ij}}}{8\lambda_{ij}}$, in which $\lambda_{ij} = \max_\omega \frac{1}{8} \frac{\omega^T \mathbf{B}_{i,j}^{r,l}\omega}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}$ and $\mathbf{B}_{i,j}^{r,l} = (\mathbf{m}_i - \mathbf{m}_j)(\mathbf{m}_i - \mathbf{m}_j)^T$. Applying (9) and (10) to (8), we have

$$\begin{aligned} \varepsilon_{ij} \le & \sum_r \sum_l \sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}} \left\{ \left(A_0 - A_1 \frac{|\omega^T \varDelta\mathbf{\Sigma}_{i,j}^{r,l}\omega|}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega} \right) \right. \\ & \left. - B_{ij} \left[\min_\omega \left(A_0 - A_1 \frac{|\omega^T \varDelta\mathbf{\Sigma}_{i,j}^{r,l}\omega|}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega} \right) \right] \frac{[\omega^T(\mathbf{m}_{i,r} - \mathbf{m}_{j,l})]^2}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega} \right\} \\ = & \sum_r \sum_l \sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}} \left(A_0 - A_1 \frac{|\omega^T \varDelta\mathbf{\Sigma}_{i,j}^{r,l}\omega|}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega} - B_{ij}(A_0 - A_1) \frac{[\omega^T(\mathbf{m}_{i,r} - \mathbf{m}_{j,l})]^2}{\omega^T \bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega} \right), \end{aligned} \tag{11}$$

where we have used the fact that $0 \le \frac{|\omega^T \varDelta\mathbf{\Sigma}_{ij}\omega|}{\omega^T \bar{\mathbf{\Sigma}}_{ij}\omega} \le 1$.

3.3 An Upper Bound of Multiclass Bayes Error

For the c classes problem, the Bayes error can be upper bounded as $\varepsilon \leq \frac{1}{2}\sum_{i}\sum_{j\neq i}\varepsilon_{ij}$ [20]. Then, from (11) we obtain that

$$\begin{aligned}\varepsilon \leq \frac{A_0}{2}\sum_{i}\sum_{j\neq i}\sum_{r}\sum_{l}\sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}} - \frac{A_1}{2}\sum_{i}\sum_{j\neq i}\sum_{r}\sum_{l}\sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}}\frac{\left|\omega^T(\Delta\mathbf{\Sigma}_{i,j}^{r,l})\omega\right|}{\omega^T\bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}\\ -\frac{B_{\min}(A_0-A_1)}{2}\sum_{i}\sum_{j\neq i}\sum_{r}\sum_{l}\sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}}\frac{[\omega^T(\mathbf{m}_{i,r}-\mathbf{m}_{j,l})]^2}{\omega^T\bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega},\end{aligned} \tag{12}$$

where $B_{\min} = \min_{i,j}\{B_{ij}\} = \frac{1-e^{\;\lambda_{\max}}}{8\lambda_{\max}}$ and $\lambda_{\max} = \max_{i,j}\{\lambda_{ij}\}$. Recursively applying the following inequality

$$\frac{a}{b} + \frac{c}{d} \geq \frac{a+c}{b+d},\ \forall a, c \geq 0; b, d > 0 \tag{13}$$

to the error bound in (12), we have the following upper bound of the Bayes error:

$$\begin{aligned}\varepsilon \leq \frac{A_0}{2}\sum_{i}\sum_{j\neq i}\sum_{r}\sum_{l}\sqrt{\hat{\pi}_{i,r}\hat{\pi}_{j,l}} - \frac{A_1}{2}\frac{\sum_i\sum_{j\neq i}\sum_r\sum_l(\hat{\pi}_{i,r}\hat{\pi}_{j,l})^{\frac{3}{2}}|\omega^T\Delta\mathbf{\Sigma}_{i,j}^{r,l}\omega|}{\sum_i\sum_{j\neq i}\sum_r\sum_l\hat{\pi}_{i,r}\hat{\pi}_{j,l}\omega^T\bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}\\ -\frac{B_{\min}(A_0-A_1)}{2}\frac{\sum_i\sum_{j\neq i}\sum_r\sum_l(\hat{\pi}_{i,r}\hat{\pi}_{j,l})^{\frac{3}{2}}[\omega^T(\mathbf{m}_{i,r}-\mathbf{m}_{j,l})]^2}{\sum_i\sum_{j\neq i}\sum_r\sum_l\hat{\pi}_{i,r}\hat{\pi}_{j,l}\omega^T\bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}.\end{aligned} \tag{14}$$

3.4 Our BDA/GMM Method

As the exact value of the Bayes error is hard to evaluate, to minimize the Bayes error, we may minimize its upper bound instead. From (14) we may maximize the following function

$$\begin{aligned}J(\omega) = \frac{\sum_i\sum_{j\neq i}\sum_r\sum_l(\hat{\pi}_{i,r}\hat{\pi}_{j,l})^{\frac{3}{2}}[\omega^T(\mathbf{m}_{i,r}-\mathbf{m}_{j,l})]^2}{\sum_i\sum_{j\neq i}\sum_r\sum_l\hat{\pi}_{i,r}\hat{\pi}_{j,l}\omega^T\bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}\\ +\frac{A_1}{B_{\min}(A_0-A_1)}\frac{\sum_i\sum_{j\neq i}\sum_r\sum_l(\hat{\pi}_{i,r}\hat{\pi}_{j,l})^{\frac{3}{2}}|\omega^T\Delta\mathbf{\Sigma}_{i,j}^{r,l}\omega|}{\sum_i\sum_{j\neq i}\sum_r\sum_l\hat{\pi}_{i,r}\hat{\pi}_{j,l}\omega^T\bar{\mathbf{\Sigma}}_{i,j}^{r,l}\omega}.\end{aligned} \tag{15}$$

Let

$$\mathbf{B} = \sum_{i}\sum_{j\neq i}\sum_{r}\sum_{l}(\hat{\pi}_{i,r}\hat{\pi}_{j,l})^{\frac{3}{2}}(\mathbf{m}_{i,r}-\mathbf{m}_{j,l})(\mathbf{m}_{i,r}-\mathbf{m}_{j,l})^T$$

and

$$\bar{\mathbf{\Sigma}} = \sum_{i}\sum_{j\neq i}\sum_{r}\sum_{l}\hat{\pi}_{i,r}\hat{\pi}_{j,l}\bar{\mathbf{\Sigma}}_{i,j}^{r,l}.$$

Then we have the following discriminant criterion

$$J(\omega,\mu)=\frac{\omega^T\mathbf{B}\omega}{\omega^T\bar{\mathbf{\Sigma}}\omega}+\mu\frac{\sum_i\sum_{j\neq i}\sum_r\sum_l(\hat{\pi}_{i,r}\hat{\pi}_{j,l})^{\frac{3}{2}}|\omega^T(\mathbf{\Sigma}_{i,r}-\mathbf{\Sigma}_{j,l})\omega|}{\omega^T\bar{\mathbf{\Sigma}}\omega}, \quad (16)$$

where $0\le\mu\le\frac{A_1}{B_{\min}(A_0-A_1)}$ is a parameter to make the upper bound tighter, whose optimal value can be found by cross validation. Based on the above discriminant criterion $J(\omega,\mu)$, we define the optimal discriminant vectors of BDA/GMM as follows [14]:

$$\omega_1=\arg\max_{\omega}J(\omega,\mu),\quad \text{and}\quad \omega_k=\arg\max_{\substack{\omega^T\bar{\mathbf{\Sigma}}\omega_j=0,\\ j=1,\cdots,k-1}}J(\omega,\mu),\quad (k>1). \quad (17)$$

4 An Efficient Algorithm for BDA/GMM

Let $\omega=\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}\alpha$, $\hat{\mathbf{\Sigma}}_{i,r}=\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}\mathbf{\Sigma}_{i,r}\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}$, $\hat{\mathbf{\Sigma}}_{j,l}=\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}\mathbf{\Sigma}_{j,l}\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}$, and $\hat{\mathbf{B}}=\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}\mathbf{B}\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}$. Then the optimization problem (17) becomes:

$$\alpha_1=\arg\max_{\alpha}\hat{J}(\alpha,\mu),\quad \text{and}\quad \alpha_k=\arg\max_{\alpha^T\mathbf{U}_{k\ \ 1}=\mathbf{0}}\hat{J}(\alpha,\mu), \quad (18)$$

where

$$\mathbf{U}_{k-1}=[\bar{\mathbf{\Sigma}}^{-1}\alpha_1,\bar{\mathbf{\Sigma}}^{-1}\alpha_2,\cdots,\bar{\mathbf{\Sigma}}^{-1}\alpha_{k-1}]\quad\text{and}$$

$$\hat{J}(\alpha,\mu)=\frac{\alpha^T\hat{\mathbf{B}}\alpha}{\alpha^T\alpha}+\mu\frac{\sum_i\sum_{j\neq i}\sum_r\sum_l(\hat{\pi}_{i,r}\hat{\pi}_{j,l})^{\frac{3}{2}}|\alpha^T(\hat{\mathbf{\Sigma}}_{i,r}-\hat{\mathbf{\Sigma}}_{j,l})\alpha|}{\alpha^T\alpha}.$$

Let $K=\max\{K_i|i=1,2,\cdots,c\}$, $\mathbf{S}=(\mathbf{S})_{c\times c\times K\times K}$ be a $c\times c\times K\times K$ sign tensor whose elements $(\mathbf{S})_{ijrl}=s_{ijrl}\in\{+1,-1\}$, and $\mathbf{\Omega}=\{\mathbf{S}|(\mathbf{S})_{ijrl}\in\{+1,-1\}\}$ denote the set of sign tensors. Further define

$$\mathbf{T}(\mathbf{S},\mu)=\hat{\mathbf{B}}+\mu\sum_i\sum_{j\neq i}\sum_r\sum_l(\hat{\pi}_{i,r}\hat{\pi}_{j,l})^{\frac{3}{2}}s_{ijrl}(\hat{\mathbf{\Sigma}}_{i,r}-\hat{\mathbf{\Sigma}}_{j,l}).$$

Then we have

$$\hat{J}(\alpha,\mu)=\max_{\mathbf{S}\in\mathbf{\Omega}}\frac{\alpha^T\mathbf{T}(\mathbf{S},\mu)\alpha}{\alpha^T\alpha}. \quad (19)$$

From (18) and (19), the optimal vectors α_i in (18) can be expressed as

$$\begin{aligned}\alpha_1&=\arg\max_{\mathbf{S}\in\mathbf{\Omega}}\max_{\alpha}\frac{\alpha^T\mathbf{T}(\mathbf{S},\mu)\alpha}{\alpha^T\alpha},\\ \cdots&\\ \alpha_k&=\arg\max_{\mathbf{S}\in\mathbf{\Omega}}\max_{\alpha^T\mathbf{U}_{k\ \ 1}=\mathbf{0}}\frac{\alpha^T\mathbf{T}(\mathbf{S},\mu)\alpha}{\alpha^T\alpha}.\end{aligned} \quad (20)$$

Suppose that the sign tensor $\mathbf{S}$ is fixed, then the first vector α_1 in (20) is the eigenvector associated with the largest eigenvalue of $\mathbf{T}(\mathbf{S},\mu)$. The principal

Algorithm 1. Solution method for ω_i $(i = 1, 2, \cdots, k)$

```
Input:
  – GMM parameters $\mathbf{m}_{i,r}$ $(i = 1, \cdots, c)$ and $\mathbf{\Sigma}_{i,r}$, $\hat{\pi}_{i,r}$, and $K_i$. Parameter $\mu$.
Initialization:
  1. Compute matrices $\bar{\mathbf{\Sigma}}$ and $\mathbf{B}$; Perform SVD of $\bar{\mathbf{\Sigma}}$: $\bar{\mathbf{\Sigma}} = \mathbf{U}\Lambda\mathbf{U}^T$, compute $\bar{\mathbf{\Sigma}}^{-\frac{1}{2}} = \mathbf{U}\Lambda^{-\frac{1}{2}}\mathbf{U}^T$ and $\bar{\mathbf{\Sigma}}^{-1} = \mathbf{U}\Lambda^{-1}\mathbf{U}^T$, $\hat{\mathbf{\Sigma}}_{i,r} = \bar{\mathbf{\Sigma}}^{-\frac{1}{2}}\mathbf{\Sigma}_{i,r}\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}$, $\hat{\mathbf{B}} = \bar{\mathbf{\Sigma}}^{-\frac{1}{2}}\mathbf{B}\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}$;
For $i = 1, 2, \cdots, k$, Do
  1. Set $\mathbf{S} \leftarrow \text{ones}(c, c, K, K)$, where $K = \max\{K_i | i = 1, \cdots, c\}$, $\mathbf{S}_1 \leftarrow \mathbf{S}$;
  2. Solve the principal eigenvector of $\hat{\mathbf{B}}\alpha_i = \lambda\alpha_i$ via the power method;
  3. Set $(\mathbf{S}_1)_{ijlr} \leftarrow \text{sign}({\alpha_i}^T(\hat{\mathbf{\Sigma}}_{i,r} - \hat{\mathbf{\Sigma}}_{j,l})\alpha_i)$;
  4. While $\mathbf{S} \neq \mathbf{S}_1$, Do
     (a) Set $\mathbf{S} \leftarrow \mathbf{S}_1$;
     (b) Compute $\mathbf{T}(\mathbf{S}, \mu) = \hat{\mathbf{B}} + \mu \sum_i \sum_{j \neq i} \sum_r \sum_l s_{ijrl}(\pi_{i,r}\pi_{j,l})^{\frac{3}{2}}(\hat{\mathbf{\Sigma}}_{i,r} - \hat{\mathbf{\Sigma}}_{j,l})$ and solve the principal eigenvector of $\mathbf{T}(\mathbf{S}, \mu)\alpha_i = \lambda\alpha_i$ via the power method;
     (c) Set $(\mathbf{S}_1)_{ijlr} \leftarrow \text{sign}({\alpha_i}^T(\hat{\mathbf{\Sigma}}_{i,r} - \hat{\mathbf{\Sigma}}_{j,l})\alpha_i)$;
  5. If $i = 1$, $\mathbf{q}_i \leftarrow \alpha_i$, $\mathbf{q}_i \leftarrow \mathbf{q}_i / \|\mathbf{q}_i\|$, and $\mathbf{Q}_1 \leftarrow \mathbf{q}_i$;
     else $\mathbf{q}_i \leftarrow \alpha_i - \mathbf{Q}_{i-1}(\mathbf{Q}_{i-1}^T\alpha_i)$, $\mathbf{q}_i \leftarrow \mathbf{q}_i / \|\mathbf{q}_i\|$, and $\mathbf{Q}_i \leftarrow (\mathbf{Q}_{i-1} \quad \mathbf{q}_i)$;
  6. Compute $\hat{\mathbf{\Sigma}}_{p,q} \leftarrow \hat{\mathbf{\Sigma}}_{p,q} - (\hat{\mathbf{\Sigma}}_{p,q}\mathbf{q}_i)\mathbf{q}_i^T - \mathbf{q}_i(\mathbf{q}_i^T\hat{\mathbf{\Sigma}}_{p,q}) + \mathbf{q}_i(\mathbf{q}_i^T\hat{\mathbf{\Sigma}}_{p,q}\mathbf{q}_i)\mathbf{q}_i^T$ $(p = 1, \cdots, c;\ q = 1, \cdots, K_p\ )$;
  7. Compute $\hat{\mathbf{B}} \leftarrow \hat{\mathbf{B}} - \hat{\mathbf{B}}\mathbf{q}_i\mathbf{q}_i^T - \mathbf{q}_i(\mathbf{q}_i^T\hat{\mathbf{B}}) + \mathbf{q}_i(\mathbf{q}_i^T\hat{\mathbf{B}}\mathbf{q}_i)\mathbf{q}_i^T$
Output:
  – $\omega_i = \frac{1}{\sqrt{\alpha_i^T\bar{\mathbf{\Sigma}}^{-1}\alpha_i}}\bar{\mathbf{\Sigma}}^{-\frac{1}{2}}\alpha_i$, $i = 1, 2, \cdots, k$.
```

eigenvector of a matrix can be efficiently computed via the power iteration approach [22]. Suppose that we have obtained the first k vectors $\alpha_1, \cdots, \alpha_k$. Then the $(k+1)$-th vector α_{k+1} can be solved thanks to the following theorem [14]:

Theorem 2. Let $\mathbf{Q}_r\mathbf{R}_r$ be the QR decomposition of $\mathbf{U}_r$, where $\mathbf{R}$ is an $r \times r$ upper triangular matrix. Then α_{r+1} defined in (20) is the principal eigenvector corresponding to the largest eigenvalue of the following matrix $(\mathbf{I}_d - \mathbf{Q}_r\mathbf{Q}_r^T)\mathbf{T}(\mathbf{S}, \mu)(\mathbf{I}_d - \mathbf{Q}_r\mathbf{Q}_r^T)$.

In [14], Zheng et al. proposed a greedy search approach to solve the suboptimal solution to a similar optimization problem as (20), where each element of $\mathbf{S}$ should be checked at least once in each iteration of finding the suboptimal vectors. Consequently, the computation cost would increase drastically when the number of Gaussian mixture components grows. To reduce the computational cost, here we propose a much more efficient algorithm to find the suboptimal solutions to (20). To this end, we introduce the following definition:

Definition 1: Let $\mathbf{S}_1$ and $\mathbf{S}_2$ be two sign tensors and α_1 and α_2 be the principal eigenvectors of $\mathbf{T}(\mathbf{S}_1, \mu)$ and $\mathbf{T}(\mathbf{S}_2, \mu)$, respectively. If $\alpha_2^T\mathbf{T}(\mathbf{S}_2, \mu)\alpha_2 > \alpha_1^T\mathbf{T}(\mathbf{S}_1, \mu)\alpha_1$, then we say that $\mathbf{S}_2$ is better than $\mathbf{S}_1$.

According to Definition 1, solving the optimal solution in ([20]) boils down to finding the best sign tensor $\mathbf{S}$. Then we have the following theorem:

Theorem 3. Suppose that $\alpha^{(1)}$ is the principal eigenvector of $\mathbf{T}(\mathbf{S}_1, \mu)$ and $\mathbf{S}_2$ is defined as $(\mathbf{S}_2)_{ijrl} = \text{sign}({\alpha^{(1)}}^T(\hat{\mathbf{\Sigma}}_{i,r} - \hat{\mathbf{\Sigma}}_{j,l})\alpha^{(1)})$. Then $\mathbf{S}_2$ is better than $\mathbf{S}_1$.

Proof: See supplementary materials. □

Thanks to Theorem 3, we are able to improve the sign tensor step by step. We give the pseudo-code of solving k most discriminant vectors of our BDA/GMM method in Algorithm 1.

5 Classification

Suppose that $\mathbf{f}_p$ $(p \in I)$ are the raw SIFT feature vectors extracted from an image F using the method described in section [2], where I denotes the center positions of the patches in F. Let $\mathbf{W} = [\omega_1, \omega_2, \cdots, \omega_k]$ and $\mathbf{g}_p = \mathbf{W}^T\mathbf{f}_p \in \mathbb{R}^k$ be the projected feature vectors of $\mathbf{f}_p$ onto $\mathbf{W}$. Let $\mathbf{M}_{\text{COV}}$ denote the covariance matrix of the feature vectors $\{\mathbf{g}_p | p \in I\}$. Since $\mathbf{M}_{\text{COV}}$ is a symmetric matrix, we concatenate the elements in the upper triangular part of $\mathbf{M}_{\text{COV}}$ into a vector $\mathbf{v}_{\text{COV}}$. Then we have the final feature vector $\mathbf{v} = \mathbf{v}_{\text{COV}}/\|\mathbf{v}_{\text{COV}}\|$ after normalizing $\mathbf{v}_{\text{COV}}$. Now we can train a classifier, e.g., the support vector machine (SVM) [19], Adaboost [23], or simply the linear classifier [21], using all the vectors $\mathbf{v}$. For a test facial image, we use the same method to obtain the corresponding vector $\mathbf{v}_{\text{test}}$, and then classify it using the trained classifier. In this paper, we choose the linear classifier for our emotion recognition task.

6 Experiments

In this section, we conduct experiments to demonstrate the effectiveness of the proposed method. Since no facial expression database with arbitrary view facial images is available, we conduct our experiments on the facial images generated from the BU-3DFE database [13]. More specifically, by projecting the 3D facial expression models in the BU-3DFE database in various directions, we can generate a set of 2D facial images with various facial views. The BU-3DFE database consists of 3D facial expression models of 100 subjects (56 female and 44 male). For each subject, there are 6 basic emotions with 4 levels of intensities. In our experiments, we only choose the 3D models with the highest level of intensity to generate 35 facial images corresponding to 35 projection directions, i.e., seven yaw angles (-45^o, -30^o, -15^o, 0^o, $+15^o$, $+30^o$, and $+45^o$) and five pitch angles (-30^o, -15^o, 0^o, $+15^o$, and $+30^o$). Consequently, we have $100 \times 6 \times 5 \times 7 = 21000$ facial images in total for our experiments. Fig. 2 shows some examples of the generated face images.

We adopt a five-fold cross validation strategy [21] to conduct the experiments. More specifically, we randomly divide the 100 subjects into five groups, each one having 20 subjects. In each trail of the experiment, we choose one group as test set and the other ones as training set. We conduct five trials of the experiment in

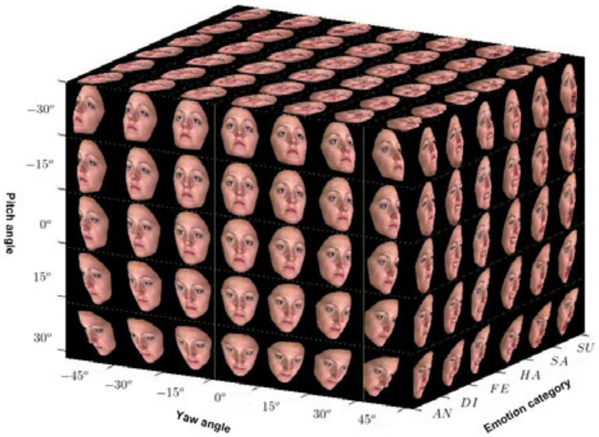


Fig. 2. Some facial images rendered from the BU-3DFE database, covering the facial images of six basic emotions, seven yaw angles, and five pitch angles

total such that each subject is used as test data once. For all the experiments, we fit the GMM with 5 different numbers, i.e., 16, 32, 64, 128, and 256, of Gaussian mixture components, and for each choice of the number of Gaussian mixture components, we apply our BDA/GMM algorithm to reduce the dimensionality of the SIFT feature vectors from 128 to 30. The parameter μ in the discriminant criterion (16) is simply fixed at $\mu = 0.5$ in all the experiments. Note that a better choice of its value may result in better performance.

Table 1 summarizes the experimental results of the overall error rates as well as the error rates of each emotion with different numbers of Gaussian mixture components. Fig.3 shows the overall confusion matrix of recognizing the six basic emotions, in which 256 Gaussian mixture components are used. From Table 1, one can see that the lowest error rate is 31.72%, achieved when 128 Gaussian mixture components are used. We can also see from Table 1 and Fig.3 that the emotions easiest to be recognized are happy and surprise, and the remaining emotions are more difficult.

Table 2 shows the overall error rates of the proposed method across various facial views when 256 Gaussian mixture components are used. In Table 2, each row of the table represents the overall error rates of different pitch angles (from -30^o to $+30^o$), while each column represents the overall error rates of different yaw angles (from -45^o to $+45^o$). From Table 2, one can clearly see that both yaw angles and pitch angles can affect the emotion recognition performance, where the best results are achieved when the facial images are frontal or near frontal.

As there are no other methods proposed for *arbitrary view* emotion recognition, we can only provide our own experimental results. Nevertheless, for comparison we also provide the results of two approaches. One is to use the linear discriminant analysis (LDA) to replace our BDA/GMM method for reducing the dimensionality of the SIFT feature vectors, and the other one is to replace the

Table 1. The overall error rates (%) of the proposed method under different numbers of Gaussian mixture components

mixture #	16	32	64	128	256
AN	43.51	43.46	42.37	43.23	42.60
DI	31.71	32.20	32.60	32.00	31.89
FE	45.06	44.60	46.49	45.17	44.89
HA	16.74	16.20	17.34	15.60	16.57
SA	44.31	43.80	41.11	41.03	42.09
SU	14.57	14.29	13.26	13.31	12.57
Ave	32.65	32.42	32.20	**31.72**	31.77

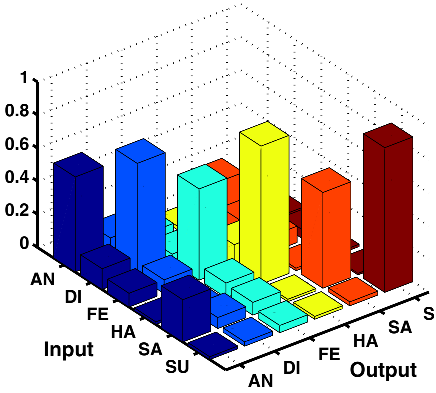


Fig. 3. The overall confusion matrix of the proposed method, where 256 Gaussian mixture components are used

Table 2. Average error rates (%) of different emotions versus different views using our method, where 256 Gaussian mixture components are used

	-30^o	-15^o	0^o	$+15^o$	$+30^o$	Ave
-45^o	39.67	35.67	31.00	33.00	43.00	36.47
-30^o	30.67	28.33	27.67	28.50	38.50	30.73
-15^o	28.33	29.17	25.83	25.83	33.17	28.47
0^o	30.83	27.83	**25.17**	25.67	31.83	**28.27**
$+15^o$	32.33	29.33	26.33	28.50	32.00	29.70
$+30^o$	32.33	29.33	29.33	32.67	35.50	31.83
$+45^o$	40.17	33.50	31.33	35.83	43.67	36.90
Ave	33.48	30.45	**28.10**	30.00	36.81	31.77

Gaussian mixtures in our BDA/GMM method with single Gaussian, denoted by BDA/Gaussian, to model each class (i.e., a view is a class) and then repeat the rest procedures in our paper, where the remaining experimental settings in

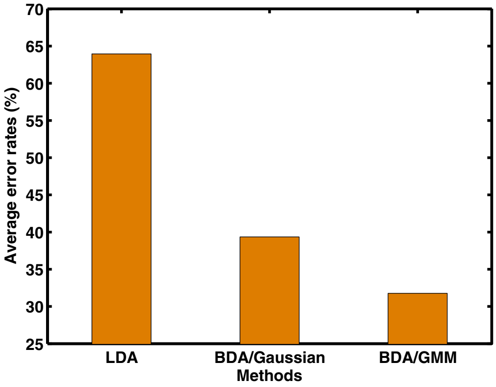


Fig. 4. Average error rate comparisons among LDA, BDA/Gaussian, and BDA/GMM

both approaches are the same as those for our BDA/GMM. Fig.4 presents the overall error rates of the three methods. From Fig.4, one can clearly see that our BDA/GMM method achieves much better results than the LDA.

7 Conclusions

In this paper we have proposed a new method to address the emotion recognition problem from arbitrary view facial images. A major advantage of this method is that it does not need face alignment or facial landmark points localization from arbitrary view facial images, both of which are very challenging. As an important part of our emotion recognition system, a novel discriminant analysis theory, called the BDA/GMM, is also developed. This new discriminant analysis theory is derived by minimizing a new upper bound of the Bayes error which is derived using the Gaussian mixture model. The proposed method is tested on a lot of facial images with various views, generated from 3D facial expression models in the BU-3DFE database. The experimental results show that our method can achieve a satisfactory recognition performance.

It is worth noting that, although having been proven to be an effective image representation method, the RCM representation may also discard some useful discriminant information, e.g., the class means of samples. Therefore, finding a better image representation method may help to improve the performance of emotion recognition. This will be one of our future work. We will also investigate whether a more advanced classifier, e.g., SVM [19] and Adaboost [23], can greatly improve the recognition performance.

Acknowledgment

This work was partly supported by NSF of China under Grants 60503023 and 60872160, and partly by the science technology foundation of Southeast University of China under the grant XJ2008320.

References

1. Darwin, C.: The expression of the emotions in man and animals. John Murray, London (1872)
2. Ekman, P., Friesen, W.V.: Pictures of facial affect. In: Human Interaction Laboratory. Univ. California Medical Center, San Francisco (1976)
3. Fasel, B., Luettin, J.: Automatic facial expression analysis: a survey. Pattern Recognition 36, 259–275 (2008)
4. Tian, Y.L., Kanade, T., Cohn, J.F.: Facial expression analysis. In: Li, S.Z., Jain, A.K. (eds.) Handbook of Facial Recognition. Springer, Heidelberg (2004)
5. Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expressions: the state of the art. IEEE Trans. on PAMI 22(12), 1424–1445 (2000)
6. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: audio, visual, and spontaneous expressions. IEEE Trans. on PAMI 31(1), 39–58 (2009)
7. Pantic, M., Patras, I.: Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. IEEE Trans. on SMC - Part B 36(2), 433–449 (2006)
8. Hu, Y., Zeng, Z., Yin, L., Wei, X., Zhou, X., Huang, T.S.: Multi-view facial expression recognition. In: Int. Conf. on Automatic Face and Gesture Recognition (2008)
9. Moore, S., Bowden, R.: The effect of pose on facial expresssion recognition. In: British Machine Vision Conference (2009)
10. Kumano, S., Otsuka, K., Yamato, J., Maeda, E., Sato, Y.: Pose-invariant facial expression recognition using variable-intensity templates. Int. J. of Comput. Vision (2009)
11. Sajama, O.A.: Supervised dimensionality reduction using mixture models. In: ICML (2005)
12. Hu, Y., Zeng, Z., Yin, L., Wei, X., Tu, J., Huang, T.S.: A study of non-frontal-view facial expressions recognition. In: Proceedings of ICPR, pp. 1–4 (2008)
13. Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.J.: A 3D facial expression database for facial behavior research. In: Proceedings of 7th Int. Conf. on Automatic Face and Gesture Recognition, pp. 211–216 (2006)
14. Zheng, W., Tang, H., Lin, Z., Huang, T.S.: A novel approach to expression recognition from non-frontal face images. In: Proceedings of IEEE ICCV, pp. 1901–1908 (2009)
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. of Comput. Vision 60(2), 91–110 (2004)
16. 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)
17. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on Riemannian manifolds. IEEE Trans. on PAMI 30(10), 1713–1727 (2008)
18. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. of Comput. Vision 57(2), 137–154 (2004)
19. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
20. Chu, J.T., Chuen, J.C.: Error probability in decision functions for character recognition. J. of the Association for Computing Machinery 14(2), 273–280 (1967)
21. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, New York (1990)
22. Golub, G., Van, C.: Matrix Computations. The Johns Hopkins University Press, Baltimore (1996)
23. R*ä*tsch G., Onoda T., M*ü*ller K.-R.: Soft margins for Adaboost. In: Machine Learning, pp. 1–35 (2000)

Face Liveness Detection from a Single Image with Sparse Low Rank Bilinear Discriminative Model

Xiaoyang Tan, Yi Li, Jun Liu, and Lin Jiang

Dept. of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, China
{x.tan,j.liu}@nuaa.edu.cn

Abstract. Spoofing with photograph or video is one of the most common manner to circumvent a face recognition system. In this paper, we present a real-time and non-intrusive method to address this based on individual images from a generic webcamera. The task is formulated as a binary classification problem, in which, however, the distribution of positive and negative are largely overlapping in the input space, and a suitable representation space is hence of importance. Using the Lambertian model, we propose two strategies to extract the essential information about different surface properties of a live human face or a photograph, in terms of latent samples. Based on these, we develop two new extensions to the sparse logistic regression model which allow quick and accurate spoof detection. Primary experiments on a large photo imposter database show that the proposed method gives preferable detection performance compared to others.

1 Introduction

Biometric techniques, which rely on the inherited biometric traits taken from the user himself for authentication, have gained wide range of applications recently [6]. Unfortunately, once such biometric data is stolen or duplicated, the advantages of biometrics become disadvantages immediately.

This situation is most commonly found in a face recognition system, where one or some photos of a valid user can be easily obtained without even physically contacting with him/her, say, through internet downloading or simply capturing them using a camera. A 2D-image based facial recognition system can be easily spoofed by these simple tricks and some poorly-designed systems have even been shown to be fooled by very crude line drawings of a human face [12]. Actually, it is a very challenging task to guard against spoofs based on a static image of a face (*c.f.*, Fig. 1), while most effort of the current face recognition research has been paid on the ”image matching” part of the system without caring whether the matched face is from a live human or not.

Current anti-spoofing methods against photograph or video of a valid user can be categorized based on different criterions, such as the kinds of biometric cues are used, whether additional devices are used, and whether human interaction is needed. A good survey of schemes against photograph spoof can be found in [8] [19]. The most commonly used facial cues include the motion of the facial images such as the blinking of eyes, and the small, involuntary movements of parts of face and head. In [19], an eyeblink-based anti-spoofing method is proposed by integrating a structured prediction method. [10] presents

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 504–517, 2010.

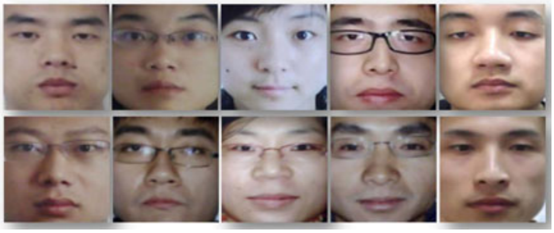


Fig. 1. Do you know which image is captured from a photo? This illustrates the difficulty of detecting photo spoof from a single static image. (Answer: all but the rightmost column are photos.)

a optimal-flow based method to capture the subtle motion of face images. Although no additional devices are required in these methods, they may encounter difficulties, for example, when a short video of the valid user is displayed or simply shaking the photograph before the camera. [7] gives an interesting example where eye-blinking and some extent of mouth movements can be well simulated using just two photographs.

Other commonly used facial cues include the surface texture of the skin and the depth information of the head. In [13], the Fourier analysis is used to capture the frequency distribution of face images of a live human. [20] lists a number of measures that could be used to characterize the optical qualities of skin from face of a live person. If specific devices are available, near infrared images or thermal images can be considered [23]. The 3D information could also be used to provide additional protection against spoof attempts with such devices as 3D cameras or multiple 2D cameras [2].

Besides facial cues, multi-modal information (*e.g.*, voices or gesture *etc.*), various challenge-response methods (*e.g.*, asking the user to blink, smile or move head) can also be considered, but these methods need either extra devices or user involvement. Another interesting research against photo spoof is to use a user-specific key to generate a random matrix to distort the face template, so that a ”stolen” face image without the key will be almost of no use [3]. This kind of method, however, mainly focus on the security of biometric templates instead of face liveness detection.

Despite the success of the above methods in some cases, non-intrusive methods without extra devices and human involvement are preferable in practice, since they could be easily integrated into an existing face recognition system, where usually only a generic webcam is equipped.

1.1 Motivations and Contributions of This Paper

Partly due to the previously mentioned drawbacks of the facial movement based methods, in this paper we focus on the methods which rely on a single static image to do spoof detection. Such methods can also be directly applied to deal with video spoof or be integrated with a video-based face liveness detection method for better performance.

The challenge here, is that the appearance of human face can change drastically due to various illumination conditions and there are also many camera-related factors that may influence the quality of images, which makes it hard to differentiate images from a live person from those from photos (*c.f.*, Fig. 1). Due to these, simply asking "what's in the image (e.g., human skin)" tends to be unreliable. Another strategy is to use various image processing techniques to extract features that highlight the difference between images from live human faces and those from photographs. In [13], a Fourier analysis based method is proposed, in which one third of components with high frequency are heuristically chosen as such features. This method works well when the photo images has low definition and small size.

In this paper, the anti-photo spoof problem is formulated as a binary classification problem, thus the statistics from the whole set of images consisting both live human faces and photographs can be fully exploited. This strategy, however, has its own difficulty in that the distributions of positive and negative are largely overlapping in the input space, and a suitable representation space is of importance. Actually, a real human face is different from a face in a photo mainly in two ways: 1) a real face is a 3D object while a photo is 2D by itself; 2) the surface roughness of a real face and a photo is different. These two factors, along with others (such as the definition of photo print and the noise introduced by the camera), usually make different image quality of a real face and a photo face under the same imaging condition (we further assume that both are properly focused). Hence exploiting such information would help to enlarge the intra-class variations between client class and imposter class.

For classification, we extended the standard sparse logistic regression classifier both nonlinearly and spatially to improve its generalization capability under the our setting (*i.e.*, high dimensionality and small size samples). It is shown that the nonlinear sparse logistic regression significantly improves the anti-photo spoof performance, while the spatial extension leads to a sparse low rank bilinear logistic regression model, which effectively control the complexity of models without manually specify the target rank beforehand (e.g., in PCA).

To evaluate our method, we collected a publicly available large photograph-imposter database containing over 50K photo images from 15 subjects. Preliminary experiments on this database show that the proposed method gives good detection performance, with advantages of realtime testing, non-intrusion and no extra hardware requirement.

The paper is organized as follows: in Section 2, we describe in detail the proposed method. Section 3 describes the photograph imposter database and gives the experimental results. Section 4 concludes this paper.

2 The Approach

We formulate the task of detecting photograph spoof as a binary classification problem. However, a simple PCA analysis indicates that there exist large overlapping between the distributions of positive and negative samples (not shown). This indicates that a suitable representation space or measure for determining whether a image arise from a live human is of importance. We do this based on the analysis of Lambertian model [18].

2.1 The Face Imaging Model

Suppose that we are given two images, $I_t(x,y)$ and $I_f(x,y)$, where $I_t(x,y)$ is taken from a live human while $I_f(x,y)$ from an imposter, say, a photograph or a frame of a video clip displayed on a laptop, and (x, y) is the position of each pixel in the image coordinate system. A useful question to ask is: what's the difference between $I_t(x, y)$ and $I_f(x, y)$ *under the same illumination condition*[1]? We examine this under the Lambertian reflectance assumption, where the face surface is modeled as an ideal diffuse reflectors, hence reflecting light according to *Lambert's cosine law* [18]. In other words, the intensity of a face image $I(x, y)$ is described as

$$I(x, y) = f_c(x, y)\rho(x, y)A_{light} \cos\theta, \tag{1}$$

where $f_c(x, y)$ term depends on the underlying camera, A_{light} is the intensity of the incoming light at a particular wavelength. The $\rho(x, y)$ term is the reflectance coefficient, which represents the diffuse reflectivity of the surface at that wavelength. The $\cos\theta = \boldsymbol{n} \cdot \boldsymbol{s}$ is the angle between the surface normal $\boldsymbol{n}$ and the incoming light ray $\boldsymbol{s}$.

First we assume that $f_c(x, y)$ is a constant. This is reasonable for many webcams. Then the client image $I_t(x, y)$ and imposter image $I_f(x, y)$ can be respectively expressed as,

$$I_t(x, y) = \rho_t(x, y)A_{light}(\boldsymbol{n_t} \cdot \boldsymbol{s}), \tag{2}$$
$$I_f(x, y) = \rho_f(x, y)A_{light}(\boldsymbol{n_f} \cdot \boldsymbol{s}). \tag{3}$$

These equations say that if under the same lighting conditions (*i.e.*, $A_{light}, \boldsymbol{s}$ terms are fixed), the differences between the two images can be made evident by comparing their surface properties, *i.e.*, the surface reflectance property $\rho_t \backslash \rho_f$, and the surface normal $\boldsymbol{n}_t \backslash \boldsymbol{n}_f$ at that point. Intuitively this is feasible since the human skin and a photograph (or the Laptop which is replaying a video clip) are made of different materials and the smoothness of their surfaces are different as well. For example, some previous work [13] uses the high frequency components of the given image to identify possible spoof. This method can be considered as a rough way to approximate the ρ value, but the information from $\boldsymbol{n}$ is lost. For the sake of robustness under various conditions (*e.g.*, against high-definition photograph spoof), exploiting full information from a given image is essential. To do this, one can write the above equations as follows,

$$I_t(x, y) = \rho_t(x, y)A(\boldsymbol{n_t} \cdot \boldsymbol{s}) \triangleq \rho_t(x, y)\mu_t(x, y) \tag{4}$$
$$I_f(x, y) = \rho_f(x, y)A(\boldsymbol{n_f} \cdot \boldsymbol{s}) \triangleq \rho_f(x, y)\mu_f(x, y) \tag{5}$$

where we denote $\mu(x, y) = A_{light}(\boldsymbol{n} \cdot \boldsymbol{s})$, which is a function of the surface normal $\boldsymbol{n}$. Hence the information we are interested in are actually encoded in the functional $\rho(x, y)$ and $\mu(x, y)$. Although we usually cannot watch the behavior of these functionals directly, we may estimate them if a series of m samples of $x_\rho^i = \rho_i(x, y)$ and $x_\mu^i = \mu_i(x, y), i = 1, 2, \ldots, m$, are available. We will call these samples "latent samples" since they are hidden by themselves, and we need a method to derive them.

2.2 Deriving Latent Samples

In this section we present two methods to derive the latent samples for our discriminative model.

[1] This assumption will be largely relaxed later.

Variational Retinex-based Method. To solve (4) or (5), one has to decompose a given face image into reflectance part $\rho(x, y)$ and illuminance part $\mu(x, y)$, which is exactly what most illumination invariant face recognition method does. Note that, however, for illumination invariant face recognition, once the albedo of a face image is obtained, the illuminance part is usually discarded. But in our case, the illuminance part is useful since it contains information about the surface normal (*c.f.* (4)). In particular, for a photograph, the surface normal is mostly constant hence the lighting factor $\boldsymbol{s}$ will actually be dominant in various $\mu_f(x, y)$ images. While for a real face, both $\boldsymbol{n}$ and $\boldsymbol{s}$ will count in $\mu_t(x, y)$. This nature of imaging variability turns out providing useful discriminative information about whether a $\mu(x, y)$ image is from a 3D object or not.

Here we prefer a type of variational Retinex approach, in which the illuminance $\mu(x, y)$ is first principally sought within the total variational framework and the albedo $\rho(x, y)$ is then estimated through Land's Retinex formula [11]. Typical methods in this line include the anisotropic smoothing method by Gross *et al.* [4] and Logarithmic Total Variation (LTV) smoothing by Chen *et al.* [1]. In this work we take the Logarithmic Total Variation (LTV) method for experiments, where a plausible illumination image $\mu(x, y)$ is estimated by minimizing a functional combining smoothness and fidelity terms:

$$\mu = \arg\min \int_{\text{image}} \|\nabla\mu\|^1 + \lambda\,|I - \mu| \tag{6}$$

where λ is the data fidelity parameter (set to 0.5 in this work). Once the μ is obtained, we estimate ρ through $\log(\rho(x, y)) = log(I(x, y) + 1) - log(\mu(x, y) + 1)$ using Land's Retinex formula [11] (*c.f.*, (4)).

Fig. 2 gives some illustration of the ρ image and μ image decomposed in this manner, from a client image and a imposter image, respectively. It can be observed that the texture of μ_f image from a photo is less rich than μ_t from a real face as expected.

Difference of Gaussian (DoG)-based Method. Another method is based on the intuitive idea that the image of a photograph taken through a webcam is essentially an image of a real face but passes through the camera system *twice* and the printing system *once*. This means that compared to an image of a real human, the imposter image tends to be more seriously distorted by the imaging system and hence has lower image quality (*i.e.*, missing more high frequency details) under the same imaging conditions. We may exploit this characteristic to distinguish an imposter image and a client image.

In particular, we do this by analyzing the 2D Fourier spectra similar to [13]. But instead of using very high frequency band which maybe too noisy, we try to exploit the difference of image variability in the high-middle band. This is done through Difference of Gaussian (DoG) filtering which is essentially a bandpass filter and has successfully been applied to remove lighting variations in face images [25]. To keep as much detail as possible without introducing noise or aliasing, we take a quite narrow inner (smaller) Gaussian ($\sigma_0 \leq 1$), while the outer one might have σ_1 of 1-2 pixels to filter out misleading low spatial frequency information. In other words, by using this preprocessing procedure, when comparing two images, we can focus more on their major part of image information without being confused by non-relevant information.

In this work, we use $\sigma_0 = 0.5$ and $\sigma_1 = 1.0$ by default. Fig. 2 gives some illustration of two images (one client image and one imposter image) and their respective Fourier

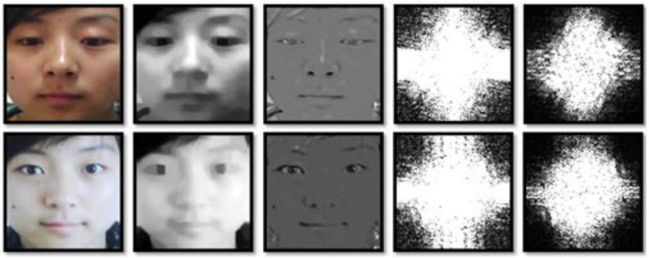


Fig. 2. Illustration of the latent samples derived for a client image (top row) and an imposter image (bottom row): from left to right, 1) the raw image; 2) the μ image estimated with LTV; 3) the ρ image estimated with LTV; 4) the centered Fourier spectra of the raw image; 4) the centered Fourier spectra of the image filtered with DoG.

spectra with/without DoG filtering. It can be observed that the DoG filter cleans the noise in the high frequency areas. In addition, the client image contains richer horizontal components in the high frequency areas than the imposter image, while the two images' respective distributions of components in various orientations are different in the middle band. Hence compared to the previous LTV-based method, the appearance variations of face images are emphasized here.

2.3 Classification

In the most simple case where the photos are taken under the same lighting condition with real faces (*i.e.*, $\boldsymbol{s_f} = \boldsymbol{s_f} = \boldsymbol{s}$), we have $\mu_f(x,y)\backslash\mu_t(x,y) = (\boldsymbol{n_f} \cdot \boldsymbol{s})\backslash(\boldsymbol{n_t} \cdot \boldsymbol{s})$. One can see that it is the surface norms $\boldsymbol{n_t}$ and $\boldsymbol{n_f}$ that dominate the ratio. This implies that one can try to first capture a real face for that specific lighting scenario then used it as reference to reject possible photo spoofing under that scenario.

In more general case where the $\boldsymbol{s_t}$ and $\boldsymbol{s_f}$ are different, one strategy is to first learn K most common lighting settings where photo spoofing may happen from training samples, using such method as Singular Value Decomposition [28]. Denote these settings as a lighting matrix $S \in \mathrm{R}^{3\times K}$. One can use this S to reconstruct any $\mu \in \mathrm{R}^D$ image such that $\| \mu - NSv \|$ is minimized, where $N \in \mathrm{R}^{D\times 3}$ is the surface normal matrix to be estimated, $v \in \mathrm{R}^K$ is the reconstruction coefficient. The object function can be optimized by coordinate descent method. After this, the reconstruction coefficient v can be used as input to a classifier.

But things become more complicated if we take the illumination distribution of each photo itself into consideration. This lighting distribution is independent with the current lighting setting, and according to Lambertian model, it should go into the albedo part but (depending on the setting of λ) the LTV decomposition (*c.f.*, (6)) often allow a significant amount of fine scene texture to leak into μ, thus making the distribution of μ become rather nonlinear. Therefore, in this work we learn a classifier directly through the obtained latent training samples without further feature extraction. In particular we adopted the sparse logistic regression model and extend it in two ways such that it may better fit the problem at hand.

Sparse Logistic Regression. Let $\mathbf{x} \in \mathbb{R}^n$ denote a sample, and $y \in \{-1, 1\}$ be the associated (binary) class label (we define imposter image as +1, client image -1). Logistic regression model is given by:

$$\text{Prob}(y|\mathbf{x}) = \frac{1}{1 + \exp(-y(\mathbf{w}^T\mathbf{x} + b))}, \tag{7}$$

where $\text{Prob}(y|\mathbf{x})$ is the conditional probability of class $y = 1$, given the sample $\mathbf{x}$, $\mathbf{w} \in \mathbb{R}^n$ is the weight vector, and $b \in \mathbb{R}$ is the intercept. Suppose that we are given a set of m training data $\{\mathbf{x}_i, y_i\}_{i=1}^m$ (*c.f.*, Sec. 2.2), where $\mathbf{x}_i \in \mathbb{R}^n$ denotes the i-th sample and $y_i \in \{-1, +1\}$ denotes the corresponding class label. The likelihood function associated with these m samples is defined as $\prod_{i=1}^m \text{Prob}(y_i|\mathbf{x}_i)$. The negative of the log-likelihood function is called the (empirical) logistic loss, and the average logistic loss is defined as:

$$\begin{aligned}\text{loss}(\mathbf{w}, b) &= -\frac{1}{m}\log\prod_{i=1}^m \text{Prob}(y_i|\mathbf{x}_i) \\ &= \frac{1}{m}\sum_{i=1}^m \log(1 + \exp(-y_i(\mathbf{w}^T\mathbf{x}_i + b))),\end{aligned} \tag{8}$$

which is a smooth and convex function. We can determine $\mathbf{w}$ and b by minimizing the average logistic loss: $\min_{\mathbf{w},b} \text{loss}(\mathbf{w}, b)$, leading to a smooth convex optimization problem. The sparse logistic regression [9,14] add a ℓ_1-norm regularization to the loss to avoid overfitting, *i.e.*, $\min_{\mathbf{w},b} \text{loss}(\mathbf{w}, b) + \lambda\|\mathbf{w}\|_1$. The major characteristic of this is that it enforces a sparse solution which is desirable for our application. In addition, there are quite a few efficient solvers for optimizing this problem, e.g., l1-log [9] and SLEP [14]. In this paper, we propose to make use of the SLEP package [16], as it enjoys the optimal convergence rate and works efficiently for large scale data.

Sparse Low Rank Bilinear Logistic Regression. To exploit the spatial property of images, we can directly operate on the two-dimensional representation of images. The goal is to learn a ”low-rank” projection matrix, or equivalently a ”low-rank” bilinear function:

$$f_{L,R}(X) = \text{tr}(L^{\mathrm{T}}XR^{\mathrm{T}}) = \text{tr}((LR)^{\mathrm{T}}X), \tag{9}$$

where $X \in \mathbb{R}^{r\times c}$, $L \in \mathbb{R}^{r\times c}$, $R \in \mathbb{R}^{c\times c}$, and r and c denote the number of rows and columns of the image X, respectively. Denote $W = LR \in \mathbb{R}^{r\times c}$, we can rewrite (9) as $f_{L,R}(X) = \text{tr}(W^{\mathrm{T}}X) = \langle\text{vec}(W), \text{vec}(X)\rangle$, leading to the traditional one-dimensional (concatenated) linear function.

However, directly learning (9) from a set of training samples can leads to overfitting, especially when m, the number of training samples is less than $p = r \times c$, the dimensionality. One standard technique is to add some penalty to control the complexity of the learned $W = LR \in \mathbb{R}^{r\times c}$. In this study, we impose the assumption that W is a “low-rank” projection matrix. Given a set of training samples $\{X_i, y_i\}_{i=1}^n$, one way to compute $W = LR$ is to optimize:

$$\min_{L,R} \text{loss}(L, R) + \lambda \times \text{rank}(LR), \tag{10}$$

where $\text{loss}(L, R)$ is a given loss function defined over the training samples, e.g., the logistic loss (8). However, $\text{rank}(LR)$ is nonconvex, and (10) is NP-hard. So instead we propose to compute L and R via

$$\min_{L,R} \text{loss}(L, R) + \lambda_1 \|L\|_{2,1} + \lambda_2 \|R\|_{2,1}. \tag{11}$$

where $\|\cdot\|_{2,1}$ is the $\ell(2, 1)$-norm of a matrix defined as the sum of the $\ell 2$ length of each column of this matrix. With appropriate parameters, (11) shall force a solution where many rows of L and R are exactly zero, so that L and R are "low-rank". As $\text{rank}(W) \leq \max(\text{rank}(L), \text{rank}(R))$, the obtained $W = LR$ is also low-rank.

To optimize (11), we apply the block coordinate descent. That is to say, we first fix R to obtain L via $\min_L \phi_R(L) + \lambda_1 \|L\|_{2,1}$, where $\phi_R(L)$ is a convex and smooth function with regard to L under given R. Similarly, we compute R under given L via $\min_R \psi_L(R) + \lambda_2 \|R\|_{2,1}$. And this process is repeated until convergence. A common practice is to terminate the program after the change of L and R (measured in the Frobenius norm) in the adjacent iterations is below a small value (1e-6 in the paper).

This model has several new features: 1) In contrasted with the existing low rank bilinear discriminative method (*e.g.*, [21]), the rank needs not to be pre-specified, but tuned via λ_1 and λ_2[2]; 2) The "low-rank" projection matrix W is obtained the computationally efficient penalty $\|\cdot\|_{2,1}$-norm without Singular Value Decomposition as needed by the trace-norm or nuclear norm; 3) Recall $f_{L,R}(X) = \text{tr}(L^{\mathrm{T}} X R^{\mathrm{T}})$, L^{T} have many columns that are exactly zero, thus being able to discarding certain columns in X_i's. Physically, this parameter matrix contains sets of learned discriminative filters for each thin strip of the face image, thus encoding spatial information.

Nonlinear Model via Empirical Mapping. In a second extension to the sparse logistic regression model, we make use of the explicit empirical mapping defined over the m training samples to transform them into the features space via the kernel mapping ϕ : $\mathbf{x} \rightarrow F$, thus we have $F = \mathbb{R}^m$. Let $\tilde{\mathbf{x}} = \phi(\mathbf{x})$, we define

$$\tilde{x}_j = \text{kernel}(\mathbf{x}, \mathbf{x}_j), j = 1, 2, \ldots, m, \tag{12}$$

where $\text{kernel}(\cdot, \cdot)$ is a given kernel function, e.g., the Gaussian kernel (*c.f.* [5] for a good account on this in the context of RBF network). With the transformed training data $\{\tilde{\mathbf{x}}_i, y_i\}_{i=1}^m$, we can apply the sparse logistic regression discussed before for constructing a sparse model.

One way to look at this model is that it can be thought of as a nonparametric probabilistic model since its number of parameters grows with the sample size while its complexity is controlled by the ℓ_1-norm prior. This characteristic is shared by many sparse nonlinear discriminative models in literatures, such as probabilistic Support Vector Machine (pSVM, [22]), Relevance Vector Machine (RVM, [26]) and Import Vector Machine (IVM, [29]). The major merit of our model, however, lies in its simplicity and its flexibility to allow a straightforward application of any efficient solver dealing with

[2] In the experiments, we follow a two-step procedure suggested in [17] to set the values of these two parameters, where a small value (1e-6) is first set for both lambda's then those coefficients with small absolute values are removed off.

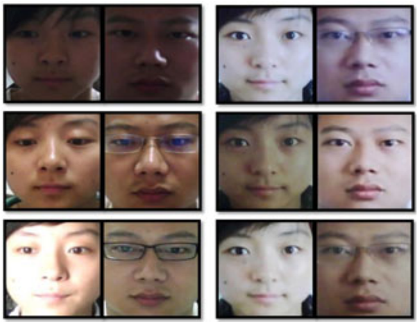


Fig. 3. Illustration of the samples from the database. In each column (from top to bottom) samples are respectively from session 1, session 2 and session 3. In each row, the left pair are from a live human and the right from a photo. Note that it contains various appearance changes commonly encountered by a face recognition system (*e.g.*, sex, illumination, with/without glasses). All original images in the database are color pictures with the same definition of 640 × 480 pixels.

usual sparse logistic regression problem, without any modification on them. Take the SLEP used here for example, its computational complexity is $O(m^2)$[15], compared to $O(m^3)$ for pSVM, $O((m+1)^3)$ for RVM and $O(m^2q^2)$ for IVM (q is the number of import points).

3 Experiments

3.1 Database

We constructed a publicly available photograph imposter database[3] using a generic cheap webcam bought from an electronic market. We collected this database in three sessions with about 2 weeks interval between two sessions, and the place and illumination conditions of each session are different as well. Altogether 15 subjects (numbered from 1 to 15)[4] were invited to attend in this work. In each session, we captured the images of both live subjects and their photographs. Some sample images from the three sessions are given in Fig. 3.

In particular, for each subject in each session, we used the webcam to capture a series of their face images (with frame rate 20fps and 500 images for each subject). During image capturing, each subject was asked to look at the webcam frontally and with neutral expression and no apparent movements such as eyeblink or head movement. In other words, we try to make a live human look like a photo as much as possible (vice versa for photograph). Some examples of the captured images are illustrated in Fig. 3 (left column).

[3] http://parnec.nuaa.edu.cn/xtan/data/NUAAImposterDB.html

[4] Since the major goal of this work is to distinguish a real face from a photograph, rather than differentiate different people as the case of usual face recognition, the requirement of large number of subjects is less demanding compared to the richness of variations contained in the datasets.

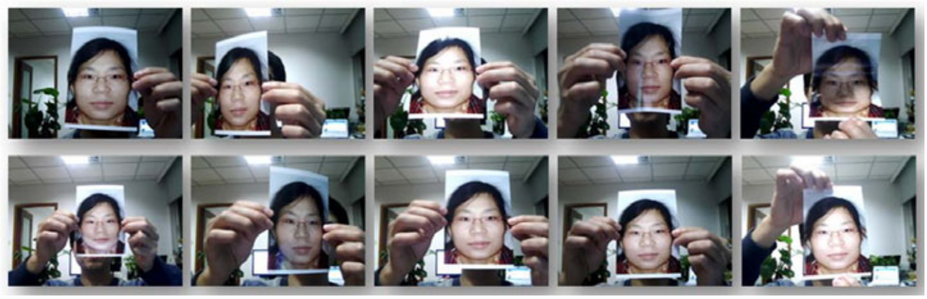


Fig. 4. Illustration of different photo-attacks (from left to right) : (1) move the photo horizontally, vertically, back and front; (2) rotate the photo in depth along the vertical axis; (3) the same as (2) but along the horizontal axis; (4) bend the photo inward and outward along the vertical axis; (5) the same as (4) but along the horizontal axis

To collect photograph samples, we first took a high definition photo for each subject using a usual Canon camera in a way that the face area should take at least 2/3 of the whole area of the photograph. We then developed the photos in two ways. The first is to use the traditional method to print them on a photographic paper with the common size of $6.8cm \times 10.2cm$ (small) and $8.9cm \times 12.7cm$ (bigger), respectively. In the other way, we print each photo on a 70g A4 paper using a usual color HP printer. Based on these, three categories of the photo-attacks are simulated before the webcam, in a way similar to [19], as shown in Fig. 4.

3.2 Settings and Performance Measure

To evaluate our methods, we first constructed a training set and a test set from the photo imposter database, both of which contain a number of client images and imposter images. The training set is constructed using the images from the first two sessions and the test set from the third session. In particular, the training set contains 889 images from the first session and 854 images from the second session and all the available subjects in the two sessions are involved.Hence we got 1743 images from 9 subjects as valid biometric trait. For the imposter images of the training set, we respectively selected 855 and 893 images from the first and the second sessions of the photograph set, hence we got 1748 imposter images in all. The test set contains 3362 images from live humans selected from session 3 and 5761 images from photos selected from session 3 as well. Table 1 gives some statistics of this. Note that there is no overlapping between the training set and the test set. In addition, some subjects in the test set are not appeared in the training set,which increases the difficulty of the problem.

All the images then undergo the same geometric normalization prior to analysis: face detected and cropped using our own Viola-Jones detector [27], rigid scaling and image rotation to place the centers of the two eyes at fixed positions, using the eye coordinates output from a eye localizer [24]; image cropping to 64 × 64 pixels and conversion to 8 bit gray-scale images.

Table 1. The number of images in the training set and test set

	Session1	session2	session3	Total
Training Set				
Client	889	854	0	1,743
Imposter	855	893	0	1,748
Total	1,744	1,747	0	3,491
Test Set				
Client	0	0	3,362	3,362
Imposter	0	0	5,761	5,761
Total	0	0	9,123	9,123

3.3 Experimental Results

Fig. 5 (Left) compares the overall performance using sparse (linear) logistic regression (SLR) with different types of input. This figure shows that the raw image (RAW) and the μ image estimated with LTV [1] (LTVu) give much worse result than the other three, *i.e.*, ρ image (LTVp), DoG filtered image (DoG) and one third of the highest frequency components in [13] (HF) (the fusion of LTVp and LTVu doesn't make big difference here and is not shown). This indicates that although the LTVu images are useful as analyzed before, their discriminative capability can not be exploited using a linear classifier.

On the other hand, the improved performance given by both LTVp and DoG shows that these two representations helps to increase the separability of the sample space. In addition, they both outperform HF in terms of AUC value[5] (respectively 0.78, 0.75 and 0.69 for the three), showing that the highest frequency components are not stable enough due to the influence of noise or aliasing in these areas. Due to the unsatisfying performance of raw gray-scale image and being rarely directly used in practice, we don't pursue this method any more in the following experiments.

To examine the effectiveness of the proposed sparse low rank bilinear logistic regression (SLRBLR), we conducted a series of experiment on the DoG images (we don't repeat the experiment on the LTV images due to their highly nonlinear distribution). We also tested a specific case of (11) named SLRBLRr1, by setting $L \in \mathbb{R}^{r \times 1}$, $R \in \mathbb{R}^{c \times 1}$, *i.e.*, $\mathrm{rank}(W) = 1$. The results are shown in Fig. 5 (Right), which shows that the sparse low rank bilinear models (with AUC value 0.92 for SLRBLR and 0.95 for SLRBLRr1) significant improve the performance upon the standard sparse logistic regression model (with AUC value 0.75).

Fig. 6 (Left) gives the results if we replace the linear classifier with a nonlinear one, *i.e.*, our sparse nonlinear logistic regression (SNLR). This shows big performance improvement upon previous. In particular, the performance of LTVu drastically improves from 0.22 to 0.92 (in terms of AUC), which verify the effectiveness of nonlinear decision boundary. Actually, by combining the LTVp and LTVu, we get further performance improvement (to 0.94). Compared to others, the ROC curve of the DoG image shows

[5] AUC: Area Under the ROC Curve, the larger the better.

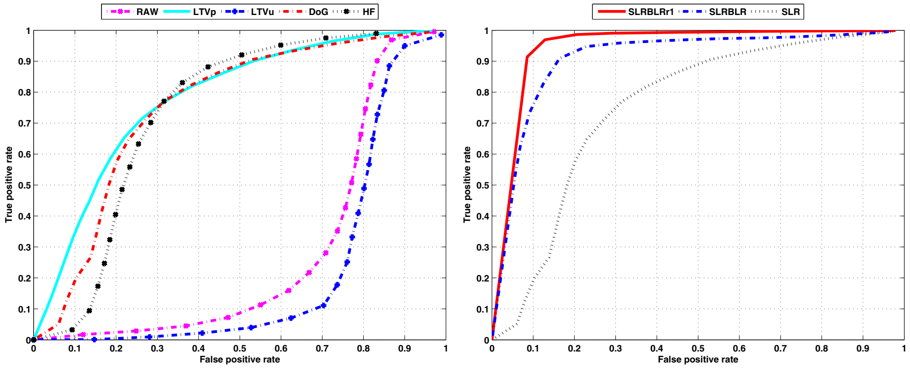


Fig. 5. (Left) Detection performance with various input features using the sparse (linear) logistic regression (SLR); (Right) Performance on the DoG images with various sparse linear discriminative model

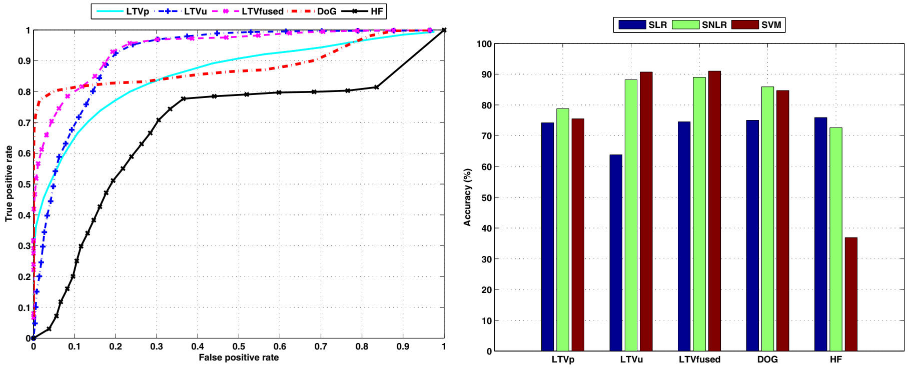


Fig. 6. (Left) Detection performance with various input features using the sparse nonlinear logistic regression. (Right) Comparison of detection rate (%) using various classification methods and input features.

a very rapid rising tendency from the very beginning of the horizontal axis. Hence it is considered the best option for use in practice among the methods compared here.

Fig. 6 (Right) shows the best overall classification accuracy of different types of input image tested respectively using sparse linear logistic regression (SLR [14]), sparse nonlinear logistic regression (SNLR) and probabilistic support vector machine (SVM [22]). We obtained this by evaluating the proportion of correctly labeled samples (either client or imposter) among the whole 9,123 test set by properly thresholding the output of each discriminative model. The figure shows that the components in the middle frequency (DoG) outperforms those in the one-third of the highest frequency (HF) by removing both the noise/alias in high frequency area and the misleading spatial information in low frequency area. In contrast, the Fourier spectra analysis method in [13] gives a classification rate of 76.7% (not shown in the figure) - about 10% lower than that of DoG. As for the image decomposition method, we see that both the albedo (LTVp) and structure (LTVu) part contribute to the discriminative capability of the system, especially when a nonlinear model is used. Combining them slightly improves the performance.

4 Conclusions

In this work, we present a novel method for liveness detection against photo spoofing in face recognition. We investigate the different nature of imaging variability from a live human or a photograph based on the analysis of Lambertian model, which leads to a new strategy to exploit the information contained in the given image. We show that some current illumination-invariant face recognition algorithm can be modified to collect the needed latent samples, which allows us to learn a sparse nonlinear/bilinear discriminative model to distinguish the inherent surface properties of a photograph and a real human face. Experiments on a large photo imposter database show that the proposed method gives promising photo spoof detection performance, with advantages of realtime testing, non-intrusion and no extra hardware requirement.

Learning the surface properties of object through samples is an classical open problem in computer vision. Although there are lots of related work in the field of texture analysis, their goal is different from ours. We believe that our work is the first one trying to use the learning technique to distinguish whether a given static image is from a live human or not. We are currently investigating the possibility to integrate various texture descriptors to further improve the performance.

Acknowledgement

Thank anonymous reviewers for their valuable comments. This research was supported by the National Science Foundation of China (60773060, 60905035), the Jiangsu Science Foundation (BK2009369) and the Project sponsored by SRF for ROCS, SEM.

References

[1] Chen, T., Yin, W., Zhou, X., Comaniciu, D., Huang, T.: Total variation models for variable lighting face recognition. IEEE TPAMI 28(9), 1519–1524 (2006)

[2] Fladsrud, T.: Face recognition in a border control environment. Tech. rep. (2005)

[3] Goh, A.: Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. IEEE TPAMI 28(12), 1892–1901 (2006)

[4] Gross, R., Brajovic, V.: An image preprocessing algorithm for illumination invariant face recognition. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 10–18. Springer, Heidelberg (2003)

[5] Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Englewood Cliffs (1999)

[6] Jain, A.K., Flynn, P., Ross, A.A.: Handbook of Biometrics. Springer, New York (2007)

[7] Joshi, T., Dey, S., Samanta, D.: Multimodal biometrics: state of the art in fusion techniques. Int. J. Biometrics 1(4), 393–417 (2009)

[8] Nixon, K., Aimale, V., Rowe, R.: Spoof detection schemes. In: Handbook of Biometrics, pp. 403–423 (2008)

[9] Koh, K., Kim, S., Boyd, S.: An interior-point method for large-scale l1-regularized logistic regression. JMLR 8, 1519–1555 (2007)

[10] Kollreider, K., Fronthaler, H., Bigun, J.: Non-intrusive liveness detection by face images. Image Vision Comput. 27(3), 233–244 (2009)

[11] Land, E.H., McCann, J.J.: Lightness and retinex theory. J. Opt. Soc. Am. 61(1), 1–11 (1971)

[12] Lewis, M., Statham, P.: CESG biometric security capablilities programme: Method, results and research challenges. In: Biometrics Consortiumn Conference (2004)

[13] Li, J., Wang, Y., Tan, T., Jain, A.K.: Live face detection based on the analysis of fourier spectra. In: SPIE, pp. 296–303 (2004)

[14] Liu, J., Chen, J., Ye, J.: Large-scale sparse logistic regression. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2009)

[15] Liu, J., Ji, S., Ye, J.: Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In: Uncertainty in Artificial Intelligence (2009)

[16] Liu, J., Ji, S., Ye, J.: SLEP: Sparse Learning with Efficient Projections. Arizona State University (2009), http://www.public.asu.edu/~jye02/Software/SLEP

[17] Meinshausen, N., Yu, B.: Lasso-type recovery of sparse representations for high-dimensional data. In: Annals of Statistics, pp. 246–270 (2009)

[18] Oren, M., Nayar, S.: Generalization of the lambertian model and implications for machine vision. IJCV 14(3), 227–251 (1995)

[19] Pan, G., Wu, Z., Sun, L.: Liveness detection for face recognition. In: Recent Advances in Face Recognition, pp. 236–252 (2008)

[20] Parziale, G., Dittmann, J., Tistarelli, M.: Analysis and evaluation of alternatives and advanced solutions for system elements. In: BioSecure (2005)

[21] Pirsiavash, H., Ramanan, D., Fowlkes, C.: Bilinear classifiers for visual recognition. In: NIPS, pp. 1482–1490 (2009)

[22] Platt, J.C.: Probabilistic outputs for support vector machines and comparisions to regularized likelihood methods. In: Advances in Large Margin Classifiers, pp. 61–74. MIT Press, Cambridge (1999)

[23] Socolinsky, D.A., Selinger, A., Neuheisel, J.D.: Face recognition with visible and thermal infrared imagery. Comput. Vis. Image Underst. 91(1-2), 72–114 (2003)

[24] Tan, X., Song, F., Zhou, Z.H., Chen, S.: Enhanced pictorial structures for precise eye localization under uncontrolled conditions. In: CVPR, pp. 1621–1628 (2009)

[25] Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Transactions on Image Processing 19(6), 1635–1650 (2010)

[26] Tipping, M.E.: The relevance vector machine. In: NIPS, vol. 12, pp. 652–658 (2000)

[27] Viola, P., Jones, M.J.: Robust real-time face detection. IJCV 57(2), 137–154 (2004)

[28] Yuille, A.L., Snow, D., Epstein, R., Belhumeur, P.N.: Determining generative models of objects under varying illumination: Shape and albedo from multiple images using svd and integrability. IJCV 35(3), 203–222 (1999)

[29] Zhu, J., Hastie, T.: Kernel logistic regression and the import vector machine. J. Compu. & Graph. Stat., 1081–1088 (2001)

Robust Head Pose Estimation Using Supervised Manifold Learning

Chiraz BenAbdelkader

New York Institute of Technology,
Abu Dhabi, United Arab Emirates
cbenabde@nyit.edu

Abstract. We address the problem of fine-grain head pose angle estimation from a single 2D face image as a continuous regression problem. Currently the state of the art, and a promising line of research, on head pose estimation seems to be that of nonlinear manifold embedding techniques, which learn an ”optimal” low-dimensional manifold that models the nonlinear and continuous variation of face appearance with pose angle. Furthermore, *supervised* manifold learning techniques attempt to achieve this robustly in the presence of latent variables in the training set (especially identity, illumination, and facial expression), by incorporating head pose angle information accompanying the training samples. Most of these techniques are designed with the classification scenario in mind, however, and are not directly applicable to the regression scenario where continuous numeric values (pose angles), rather than class labels (discrete poses), are available. In this paper, we propose to deal with the regression case in a principled way. We present a taxonomy of methods for incorporating continuous pose angle information into one or more stages of the manifold learning process, and discuss its implementation for Neighborhood Preserving Embedding (NPE) and Locality Preserving Projection (LPP). Experiments are carried out on a face dataset containing significant identity and illumination variations, and the results show that our regression-based approach far outperforms previous supervised manifold learning methods for head pose estimation.

Keywords: head pose estimation, supervised learning, manifold learning, dimensionality reduction, nonlinear regression.

1 Introduction

Head pose estimation from a single 2D image is a basic and important task of many face processing applications, viz. face recognition, face and person tracking, and human-machine interfaces [1,2,3,4]. In face recognition systems, head pose is a major source of (obviously unwanted) intra-person facial appearance variability, which can be removed by performing head pose estimation as a preprocessing step to select only face images with similar head poses for face matching. In human-computer interfaces, head pose provides a strong cue for determining a person's gaze direction and thereby inferring their focus of attention, intent,

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 518–531, 2010.

and behavior. Pose estimation can also be used as a front end processing module for face tracking to bootstrap the tracker and re-initialize it when it drifts off.

Previous work on head pose estimation from 2D images can be divided across several categories: coarse (discrete) vs. fine-grain (continuous), geometric-based vs. appearance-based methods, holistic vs. local region based. The interested reader is referred to the recent surveys for a comprehensive review [5,3].

Currently, the state of the art on head pose estimation, and a promising line of research, seems to be in manifold embedding, a special class of dimensionality reduction techniques that attempt to learn a low-dimensional manifold on which the data lies [6,7,8]. A fundamental underlying assumption of this approach is that face images with varying head pose are —geometrically speaking— points that reside on or near a low-dimensional manifold embedded in the ambient high-dimensional input space (image space), and whose intrinsic dimensionality is no more than the number of degrees of freedom of head movement [3]. This (pose) manifold models the nonlinear and continuous variations of face appearance with pose angle, and *if* learned properly, can be used to accurately predict pose angle from face images. But this manifold is highly nonlinear and complex and learning it is no easy task, particularly in the presence of distracting variation in the dataset, namely background clutter, natural variations (identity, facial expression), and imaging variations (illumination, blur, noise, etc.) in the face images.

As in any statistical learning problem, a necessary condition for accurate learning to take place is to somehow suppress extraneous variables in the training set while preserving variables of interest (the pose variable in our case). This is a recurring problem in the face processing literature, and is generally handled using one or a combination of the following two general approaches:

Image Preprocessing: preprocess the face images to extract and/or enhance certain low-level features such as histogram equalization, edge enhancement, Gabor wavelets, histogram of gradients (HOG), etc. This approach can work well for suppressing certain imaging variations, misalignment errors, and background variations.

Supervised Learning: use auxiliary information associated with the training set to bias the learning process. This approach is widely used in classification scenarios, with auxiliary information consisting of class labels of the variable of interest (e.g. subject labels in the case of face recognition).

The focus of this paper is on developing manifold learning methods of the second category in the context of head pose estimation. Previous research in this area has, for the most part, viewed the problem as a classification problem wherein the viewing sphere is (artificially) quantized into non-overlapping subintervals, and head pose is represented by a set of discrete pose labels–rather than a continuum of pose angles. This approach appears to be adequate for *coarse* pose estimation (with some reservations) [9,10,11,12,13,14,15], and other classification problems such as facial expression and face recognition [16,17,18,19,20]. It is, however, fundamentally flawed when used for fine-grain pose estimation for two main reasons: (i) pose estimation discontinuities occur at class boundaries due to the arbitrary nature of the pose classes, (ii) the numerical properties (scale, well-ordering) of the underlying pose angles are lost; for example, the difference between pose label 1 and pose label 2 is viewed no differently than between pose labels 1 and 5.

To date relatively little work exists that attempts to solve head pose estimation as a regression problem proper within a nonlinear manifold learning framework [21,22,23]. In this paper we present a principled and detailed look into this approach. Specifically, we propose a *taxonomy* of methods for using pose angles associated with the training set in the various stages of the manifold learning process. We demonstrate the proposed techniques on Neighborhood Preserving Embedding (NPE) [24] and Locality Preserving Projection (LPP) [25,17], which are linearized versions of the well-known manifold learning methods locally linear embedding (LLE) and Laplacian eigenmaps (LE), respectively. Experimental results on the FacePix database [26] show that our regression-based approach is robust to identity and illumination variations, and clearly outperforms recent similar pose estimation methods such as [11,23,14].

The remainder of the paper is organized as follows. Section 2 gives an overview of manifold embedding techniques using graph embedding as a general framework. Section 3 presents our taxonomy of supervised manifold learning methods. Section 4.4 gives experiments and results on the FacePix and AT&T datasets. Finally, Section 5 concludes with a summary and directions for future work.

2 A General Framework for Manifold Learning

Manifold learning algorithms can in general be cast in terms of a graph embedding problem based on a specific intrinsic graph that encodes certain desired statistical or geometric properties of the dataset [27]. Specifically, given a dataset of n points in p-dimensional space (denoted $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$), a manifold learning algorithm basically executes a four-stage pipeline of the following form:

1. *Neighborhood computation.*
   For each data point $\mathbf{x}_i$, determine its k closest points (called *neighborhood*), where k is a design parameter and proximity/nearness is based on some inter-point distance metric, $\mathbf{D}_{ij}$, such as Euclidean, geodesic, Mahalanobis, cosine, etc. In our case, points are face images and hence inter-point distances represent appearance dissimilarity between face images.

2. *Neighborhood graph construction.*
   A weighted graph G is constructed whose vertices are the data points, edges connect each point with its neighbors (as defined in the previous step), and the weight of an edge, $\mathbf{W}_{ij}$, represents some measure of *affinity* or similarity between two neighbor points. Intuitively, this graph encodes the intrinsic local geometry of the manifold from which the data set is sampled.

3. *Computation of a low-dimensional graph embedding.*
   This seeks a set of n d-dimensional vectors, $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n$, that preserves the properties of graph G, in other words that preserves the intrinsic geometry of the underlying manifold. This task reduces to the optimization of a quadratic form under proper regularization constraints (typically scale normalization to avoid the trivial solution), and has a closed-form solution consisting of the smallest eigenvectors of $\mathbf{B^{-1}L}$, where $\mathbf{B}$ is a diagonal matrix such that B_{ii} is equal to the sum of the ith row of $\mathbf{W}$ and $\mathbf{L} = \mathbf{B} - \mathbf{W}$ is the Laplacian of graph G.

4. *Computation of an input-to-embedding mapping.*
   This seeks a mapping that transforms new (out-of-sample) points in the p-dimensional input space to d-dimensional space, which can be solved as a non-linear regression problem on the embedding vectors obtained in the previous step, for example using GRNN's or support vector machines [23]. But clearly this step is only required where prediction (rather than visualization) is of interest.
   Because non-linear regression in a high-dimensional space is itself tricky, some techniques bypass it and instead constrain the mapping to be linear, effectively combining the third and fourth stages into one computational step. This amounts to finding the best d-dimensional linear subspace *approximation* for the nonlinear manifold. Examples of "linearized" techniques notably include Locality Preserving Projections (LPP) [25,17] which is an extension of Laplacian Eigenmaps [28], and Neighborhood Preserving Embedding (NPE) [24] and Locally Embedded Analysis (LEA) [11], both of which are linearized variants of Locally Linear Embedding [29,6].

3 Our Taxonomy of Supervised Manifold Learning

In the context of classification, supervised manifold learning generally aims to find a low-dimensional space that maximizes the separation of points from different classes while minimizing that of points within the same class (between- and within- class scatter, respectively). However in the regression scenario, the goal is rather to find directions (axes) that best *predict* the regression variable(s) associated with the dataset, in our case, the head pose angles.

With the general four-stage manifold learning framework of Section 2 in mind, we propose a *taxonomy* of methods that correspond to different ways of incorporating the pose angle information (denoted as $z_1, z_2, \cdots, z_n$) at the different stages. Our taxonomy represents a more comprehensive treatment of the regression scenario than any previous work on head pose estimation [22,23].

3.1 Overview

A summary of the taxonomy follows below, and Table 1 compares the proposed methods with previous work on supervised manifold learning, both in the context of classification and regression scenario. Clearly, the latter remains mostly wide open for contributions, which is the goal of this work.

Stage 1: Option 1.1 Construct neighborhoods using as proximity metric the similarity of z values, i.e. the neighborhood of a sample $\mathbf{x}_i$ consists of the k data samples whose z values are most similar to z_i.

Option 1.2 Construct neighborhoods using as proximity metric the inter-point distances adjusted according to the dissimilarity of respective z values.

Stage 2: Option 2.1 Adjust the graph weights (matrix $\mathbf{W}$) based on the similarity of respective z values.

Stage 3: Option 3.1 Incorporate regression information as an additional term in the objective function to be optimized.

Option 3.2 Incorporate regression information as additional constraints in the function optimization.

Table 1. Previous supervised manifold learning techniques that are related to our proposed taxonomy, in the classification (C) and regression (R) scenarios, used for different applications (face recognition (FR), pose estimation (PE), face expression recognition (FE), visualization (V), and other (O))

	[19]	[24]	[11]	[30]	[20]	[31]	[10]	[32]	[23]	[18]	[13]	[14]	[15]
Scenario	C	C	C	C	C	C	C	C	R	C	C	C	C
Application	FE	FR	V,PE	O	FR	O	O	O	PE	FR	V	PE	FR
Related to 1.1	x	x	x	x	x								
Related to 1.2						x	x	x	x				
Related to 2.1													
Related to 3.1													
Related to 3.2										x	x	x	x

3.2 Supervised Stage 1

Ideally, in order for accurate manifold learning to occur, one needs to capture the *true* neighborhood structure of the data set in the underlying manifold. However, the actual neighborhood of a data sample in the ambient input space may be contaminated with "fake" neighbors due to the presence of noise, confounding factors, and sparse sampling. The above taxonomy contains two different ways of exploiting regression information to more robustly distinguish *true* neighbors and filter out *fake* ones. Option 1.1 relies exclusively on this information while Option 1.2 attempts to reconcile information from both the inter-point distances and regression value similarities.

In principle, Option 1.2 can be implemented in infinitely many ways by using different penalty functions spanning the entire gamut between the two extremes: using only inter-point distances (the unsupervised option) and using only regression information (Option 1.1). For example in [23] Balasubramanian et al. have suggested a family of functions of the form:

$$\widetilde{D}_{ij} = f(|z_i - z_j|) \cdot D_{ij} \tag{1}$$

where $\mathbf{D}$ is the original inter-point distance matrix, $\widetilde{\mathbf{D}}$ is the adjusted distance matrix, and f is some reciprocal *increasing* positive function. We currently use the following reciprocal function: $f(u) = \alpha \cdot u/(\beta - u)$ where α and β are scalar parameters. The choice of a reciprocal function seems appropriate because it ensures that the penalty increases at a faster rate at larger values of $|z_i - z_j|$. An exponential function might also work for this same reason.

The relative merits of Option 1.1 and Option 1.2 in effect depend on the sampling density and geometric structure of the underlying manifold. Also, using the ϵ-ball approach rather than the k nearest neighbors approach may be more helpful in some cases.

Both methods are closely related to previous supervised manifold learning techniques developed for the classification scenario. Specifically, Option 1.1 is akin to techniques that limit the neighborhood to points of the *same* class [19,11,30,20]. Interestingly, Teoh et al. call this approach ”neighborhood discriminant criterion” and argue that it is equivalent to the Fisher discriminant criterion [20]. Option 1.2 is akin to methods that adjust the inter-point distances by reducing those of same-class point pairs and/or penalizing those of point pairs of different classes [31,10,32].

3.3 Supervised Stage 2

Recall that graph weights $\mathbf{W}$ represent the geometric structure of the local neighborhood based on some inter-point distance measure. Furthermore, W_{ij} essentially determines the contribution of neighbor pair $\mathbf{x}_i$ and $\mathbf{x}_j$ in the computation of the optimal embedding in Stage 3. But because neighborhoods may be contaminated with ”fake” neighbors that distort the embedding, and just as we have used regression information in Stage 1 to determine the neighborhoods more robustly (Section 3.2), we can similarly use it to penalize (i.e. reduce) the contribution of a neighbor pair by a factor proportional to the dissimilarity between their respective regression values. In other words, the new (adjusted) graph weights could of the form:

$$\widetilde{W}_{ij} = W_{ij} \cdot g(|z_i - z_j|) \tag{2}$$

or it could also be of the form:

$$\widetilde{W}_{ij} = W_{ij} + g(|z_i - z_j|) \tag{3}$$

where $g(u)$ is some positive decreasing function, such as a negative exponential (Gaussian kernel) or a reciprocal. Interestingly, using the second (additive) form is actually equivalent to adding a term to the objective function in Stage 3, hence equivalent to Option 3.1 (Section 3.4).

3.4 Supervised Stage 3

Recall that the objective function optimization represents preserving certain intrinsic geometric properties of the manifold. Hence we can incorporate regression information at this stage by extending the objective function with an additive term that represents some other geometric property to be preserved (Option 3.1). Alternatively, or simultaneously, we can incorporate this information in the form of constraints that represent some condition or property that should be avoided (Option 3.2).

Interestingly, Local Fisher Discriminant Analysis (LFDA) [12,13] and Local Discriminant Analysis (LDE) [18] both represent possible implementations of our Option 3.2 concept, *though* they are limited to the classification scenario. In LFDA, the objective function and constraints consist of "localized" versions of within- and between- class scatter, respectively. LDE uses a very similar idea. Also "kernelized" variants of both these methods were implemented using the kernel trick.

Note, however, that in order *not* to forego the convenience of solving the problem in closed-form as a generalized eigenvalue problem ($\mathbf{Ay} = \lambda\mathbf{By}$), both the additional objective function term and the constraints need to be expressed as a positive definite quadratic form ($\mathbf{y^T\Gamma y}$ where $\mathbf{\Gamma}$ is positive definite). Also, the new constraints should replace (and ideally *supersede*) the original constraints of the unsupervised method because there is no room for using both. We *could* use a non quadratic objective function and more than one set of non-quadratic constraints, but at the price of giving up the convenience of a closed-form solution for an iterative slower solution.

Below we discuss possible implementations of Option 3.1 in the context of two specific manifold learning methods: Locally Linear Embedding (LLE) and Laplacian Eigenmaps (LE). Extension to their linearized versions (NPE and LPP) is trivial. Possible implementations of Option 3.2 is work in progress, but suffice it to note here that a viable approach is to extend or generalize the LFDA concept of using between-class scatter to the regression scenario.

Implementation for LLE. We modify the usual LLE objective function by adding a second term [29,6] :

$$\Phi_{\mathrm{LLE}}(\mathbf{y}) = \sum_i |y_i - \sum_j \Omega_{ij} y_j|^2 + \lambda \, \frac{1}{2} \sum_{i,j} (y_i - y_j)^2 \Lambda_{ij} \qquad (4)$$

where λ is a scalar constant and $\mathbf{\Omega}$ is the nxn reconstruction weights matrix, and $\mathbf{\Lambda}$ is a nxn matrix that represents some measure of similarity between the $\mathbf{z}$ values of neighbor points. Clearly the intuition is to *simultaneously* (i) preserve the local neighborhood structure, and (ii) keep neighbor points with more similar pose angles closer. However note that how well this works out is closely tied to the supervision methods of Stages 1 and 2 (Sections 3.2 and 3.3), since they determine the neighborhood and the contribution weight of each neighbor pair.

It is easy to show that Equation (4) reduces to :

$$\Phi_{\mathrm{LLE}}(\mathbf{y}) = \mathbf{y}^T\mathbf{M}\mathbf{y} + \lambda\mathbf{y}^T\widetilde{\mathbf{L}}\mathbf{y} = \mathbf{y}^T(\mathbf{M} + \lambda\widetilde{\mathbf{L}})\mathbf{y} \tag{5}$$

where $\mathbf{M} = (\mathbf{I} - \mathbf{\Omega})^T(\mathbf{I} - \mathbf{\Omega})$ and $\widetilde{\mathbf{L}}$ is the Laplacian of the affinity graph induced by $\mathbf{\Lambda}$. Clearly the extension to linearized versions of LLE (such as NPE and LEA) is trivial, as we have merely replaced $\mathbf{M}$ with $\mathbf{M} + \lambda\widetilde{\mathbf{L}}$.

We currently define similarity matrix $\mathbf{\Lambda}$ based on the heat kernel function as follows, though in principle other *decreasing* functions of $|z_i - z_j|$ would do:

$$\Lambda_{ij} = \{ \begin{matrix} \exp(-\frac{1}{2}|z_i - z_j|^2/\sigma^2) & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors } \textit{and } i \neq j \\ 0 & \text{otherwise} \end{matrix} \tag{6}$$

where σ is a design parameter (the Gaussian kernel width).

Implementation for LE. We modify the usual LE objective function by adding a second term [28] :

$$\begin{aligned} \Phi_{\mathrm{LE}}(\mathbf{y}) &= \frac{1}{2}\sum_{i,j}(y_i - y_j)^2 W_{ij} + \lambda\,\frac{1}{2}\sum_{i,j}(y_i - y_j)^2 \Lambda_{ij} && (7) \\ &= \frac{1}{2}\sum_{i,j}(y_i - y_j)^2(W_{ij} + \lambda\,\Lambda_{ij}) && (8) \\ &= \mathbf{y}^T(\mathbf{L} + \lambda\,\widetilde{\mathbf{L}})\mathbf{y} && (9) \end{aligned}$$

where λ is a scalar constant and $\mathbf{\Lambda}$ and $\widetilde{\mathbf{L}}$ are as defined above in Section 3.4. Again, the extension to linearized versions of LE (such as LPP) is trivial as we have merely replaced $\mathbf{L}$ with $\mathbf{L} + \lambda\widetilde{\mathbf{L}}$.

3.5 Discussion

A common thread runs through all these methods we have proposed: to *highlight* pose variations and *suppress* variations due to other (extraneous) factors. Specifically, given the local neighborhood nature of nonlinear manifold learning, we propose to achieve this by using the regression information to: (i) determine the neighborhood (Stage 1), (ii) determine the contribution of each neighbor pair (Stage 2), and (iii) define new or additional manifold structure preservation properties (Stage 3). These methods are complementary to some extent (at least not entirely redundant) and can certainly be used in tandem. However, the inner workings of each method depends both on the dataset: how much variation it contains and how sparsely sampled the pose angles are. Also, because these methods are so closely related, it is not clear how the synergy between them will affect performance when they are used together. Further analytical and empirical work is needed to study this synergy.

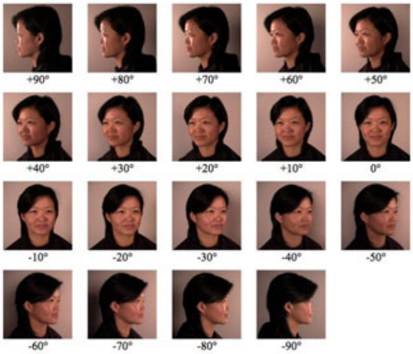


Fig. 1. Sample images from the FacePix database

4 Experiments and Results

4.1 The Data

Manifold learning and pose estimation are challenging tasks when the input face images contain significant variation, such as from illumination and identity. We test our proposed taxonomy on the FacePix database which contains face images of 30 subjects with both pose and illumination variations, namely:

Variable pose, constant illumination: 181 images for each subject captured with the yaw head pose angle varying at 1-degree increments in the range [-90,90], and with constant ambient illumination.

Constant pose, variable illumination: 181 images for each subject captured with the yaw illumination angle varying at 1-degree increments in the range [-90,90], and with constant frontal pose. Furthermore, this is done with two different illumination intensities: dark and light.

The pose and illumination angles associated with each image were annotated using a precisely calibrated mechanism. Also, the face images are scaled and aligned such that the eyes, nose, and mouth remain at fixed pixel positions in each image [26,23]. A sample of these images is shown in Figure 1. For the purpose of our experiments, we have applied some more preprocessing on these images by cropping the middle 98x98 rectangle to remove some of the background and shoulder areas, and then downsampling to a size of 25x25 pixels.

4.2 Methodology

We summarize our validation methodology in the following points:

- Use two different manifold learning algorithms: NPE [24], LPP [25].
- Use different supervision modes based on combining different implementation options for Stage 1 and Stage 3. We do not incorporate supervision into Stage 2 because it is somewhat equivalent to Stage 3 (as noted earlier).

- Use two different regression methods to estimate head pose angle from embedded face images: support vector regression (SVR) with Gaussian RBF kernel and smoothing cubic splines.
- Test on three subsets of the FacePix face images: (i) images with 1-degree pose angle increments, (ii) images with 10-degree pose angle increments, and (iii) subset (i) plus images with frontal pose and 1-degree illumination angle increments.
- Use leave-one-out cross validation to estimate pose estimation error (whereby images of 29 subjects are used for training and the images of the remaining subject are used for testing).

4.3 Visualization

Figure 2 shows the 3-dimensional embedding of the face images of 20 subjects from the FacePix dataset, based on four different manifold learning techniques and combination of supervision methods Option 1.1 and Option 3.1. In general, all methods yield a one-dimensional manifold (as expected) that is more or less ordered by pose angle, at least visually speaking. The LLE and LE manifolds are quite compact and smooth; NPE's manifold is less smooth; and LPP's manifold is the least smooth. The fact that NPE and LPP's embeddings are not as smooth as those obtained by LLE and LE is not surprising, since they only seek a linear subspace approximation of the embedding.

To get a better sense of how pose angle varies varies along these pose manifolds, we analyze the identity and pose of the (Euclidean) neighbors of each data point in the 3-dimensional embedding. Figure 3 shows that, as desired, overall each data point is surrounded by points of the same pose rather than points of the same identity. However, again, this trend is better exhibited in the LLE and LE manifolds than those of NPE and LPP.

4.4 Pose Estimation Results

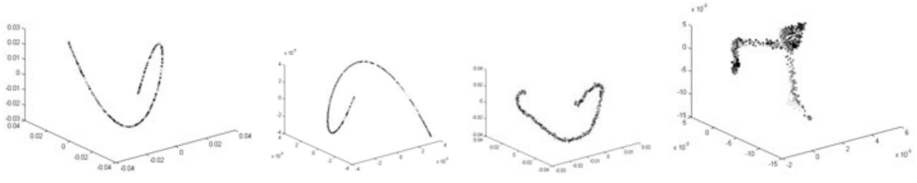
Table 2 and Table 3 show the pose angle estimation error results when using Support Vector Regression and splines, respectively, for estimating pose from the embedded face images, with $d = 20$ and $k = 50$. These results basically compare the performance of different manifold learning methods with different supervision modes. Clearly, the best performance is achieved with NPE and with the last two supervision modes, wherein supervision is incorporated both in Stages 1 and 3. The second supervision option for Stage 1 (i.e. Option 1.2) seems to perform significantly better than the first one. Overall, NPE performs better than LPP and spline regression better than support vector regression. Also, interestingly performance for the dataset containing both illumination variation and pose variation is not behind that of the dataset containing pose variation only. Hence, based on these results and on Table 1, our proposed supervision methods are superior to previous work such as [23,14].

Table 2. Mean absolute deviation of the pose angle error (in degrees), using support vector regression for pose estimation

	1-deg Pose variation		*10-deg Pose variation*		*Pose+Illum. variation*	
	NPE	LPP	NPE	LPP	NPE	LPP
unsupervised,unsupervised	8.2	9.5	11.2	14.1	9.1	15.6
Option 1.1,unsupervised	6.0	8.1	10.6	12.5	8.3	10.1
Option 1.2,unsupervised	5.5	7.9	10.8	10.3	7.7	10.0
unsupervised,Option 3.1	5.2	6.8	7.3	7.9	5.3	7.7
Option 1.1,Option 3.1	4.4	5.2	5.0	6.7	4.3	5.5
Option 1.2,Option 3.1	4.5	5.0	5.1	6.7	4.7	4.9

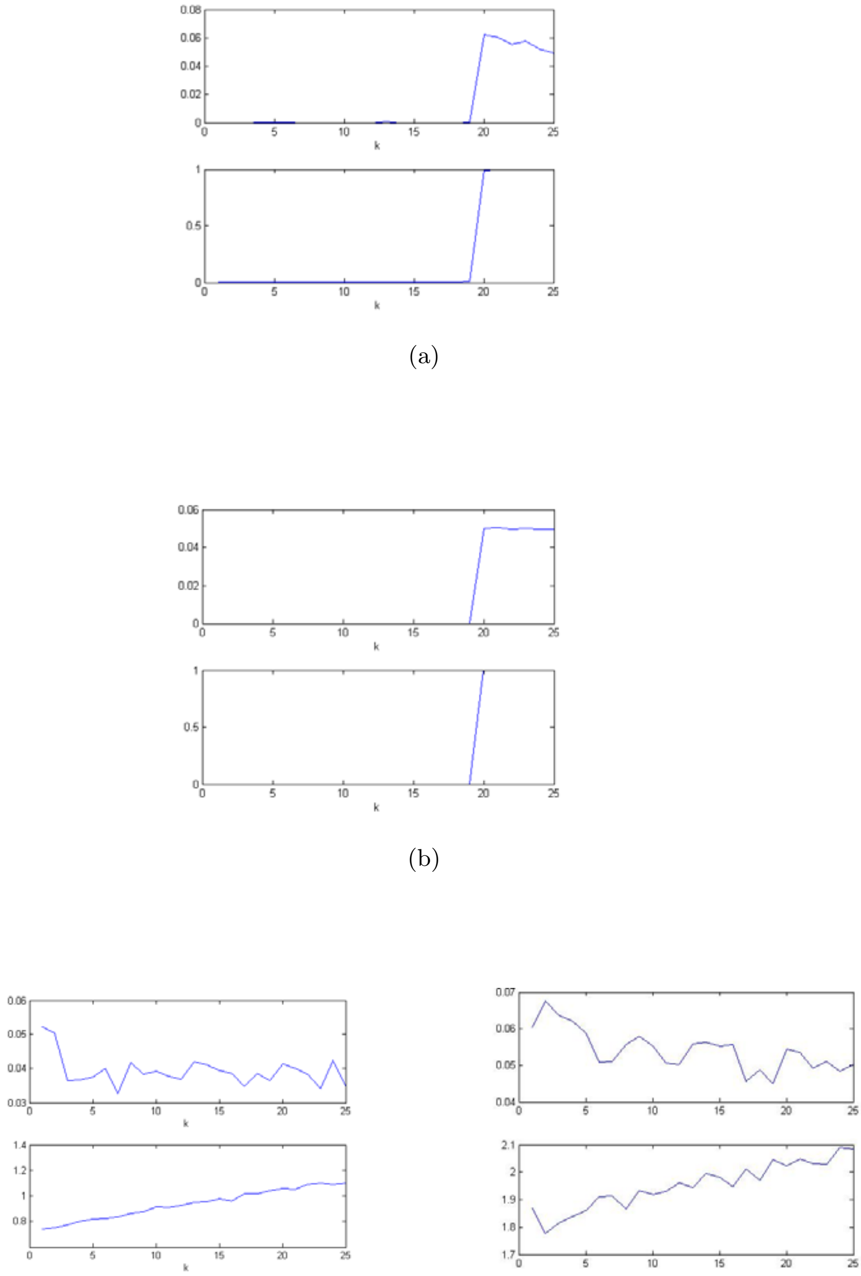
Table 3. Mean absolute deviation of the pose angle error (in degrees), using smoothing cubic splines for pose estimation

	1-deg Pose variation		*10-deg Pose variation*		*Pose+Illum. variation*	
	NPE	LPP	NPE	LPP	NPE	LPP
unsupervised,unsupervised	5.2	8.2	9.6	12.1	8.6	13.4
Option 1.1,unsupervised	4.2	7.0	9.1	10.5	7.3	9.7
Option 1.2,unsupervised	4.3	6.6	8.0	9.2	7.0	8.2
unsupervised,Option 3.1	3.5	4.6	4.6	6.1	3.9	4.3
Option 1.1,Option 3.1	2.1	3.2	4.7	5.9	3.6	4.4
Option 1.2,Option 3.1	**1.5**	3.4	3.5	5.2	2.6	3.5



(a) (b) (c) (d)

Fig. 2. 3-dimensional embedding of face images of 20 subjects of the FacePix dataset, using $k = 25$ and four different manifold learning techniques: (a) supervised LLE, (b) supervised LE, (c) supervised NPE, (d) supervised LPP. The data points are color coded differently for each subject label.



(a)

(b)

(c)

(d)

Fig. 3. Neighborhood analysis in 3-dimensional embedded space based on (a) supervised LLE, (b) supervised LE, (c) supervised NPE, (d) supervised LPP. Top figure plots probability that kth NN is of same subject versus k, and bottom figure plots absolute mean deviation of kth nearest neighbor's pose angle versus k.

5 Conclusions and Future Work

We have proposed a taxonomy of methods for solving pose estimation as a proper (continuous) regression problem within the general nonlinear manifold learning framework. The main novelty of our work lies in that, compared to previous work, we take a more comprehensive approach to the way we exploit supervision information (pose angles) into the learning process. Experiments on a face dataset containing significant identity and illumination variation have shown that our methods significantly outperform related recent work such as [11,23,14]. However, there is undoubtedly great room for improvement, most notably:

- Further analytical and empirical work to characterize the relationships between the supervision methods of the different stages, particularly in relation to pose estimation (regression) performance.
- Test on other manifold learning techniques such as Isomap [33].
- Extend to kernelized versions of NPE and LPP [27], as they only give the best linear subspace approximation of the low-dimensional embedding.
- Apply some clever feature extraction and/or preprocessing techniques on the face images (such as Gabor wavelets, histogram of gradients) to remove unwanted variation, to simplify the manifold learning process.
- Test on benchmark datasets containing more pose+illumincation variations.
- Test on benchmark datasets containing more challenging variations (facial expression, face alignment errors).

References

1. Tian, Y., Brown, L., Connell, J.: Absolute head pose estimation from overhead wide-angle cameras. In: IEEE Workshop on Analysis and Modeling of Faces and Gestures (2003)
2. Wu, J., Trivedi, M.M.: A two-stage pose estimation framework and evaluation. Pattern Recognition 41, 1138–1158 (2008)
3. Murphy-Chutorian, E., Trivedi, M.: Head pose estimation in computer vision: A survey. 31, 607–626 (2009)
4. Orozco, J., Gong, S.: Head pose classification in crowded scenes. In: British Machine Vision Conference (2009)
5. Brown, L., Tian, Y.: Comparative study of coarse head pose estimation. In: IEEE Workshop on Motion and Video Processing (2002)
6. Saul, L.K., Roweis, S.T.: Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research 4, 119–155 (2003)
7. Burges, C.J.C.: Geometric Methods for Feature Extraction and Dimensional Reduction. In: Book, Kluwer Academic Publishers, Dordrecht (2005)
8. van der Maaten, L., Postma, E., van den Herik, H.: Dimensionality reduction: A comparative review. Technical Report TiCC-TR 2009-005, Tilburg University (2009)
9. Raytchev, B., Yoda, I., Sakaue, K.: Head pose estimation by nonlinear manifold learning. In: ICPR, pp. 462–466 (2004)
10. Geng, X., Chuan Zhan, D., Hua Zhou, Z.: Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Transactions on Systems, Man, and Cybernetics: Part B 35, 1098–1107 (2005)
11. Fu, Y., Huang, T.S.: Graph embedded analysis for head pose estimation. In: FGR, pp. 3–8 (2006)

12. Sugiyama, M.: Local fisher discriminant analysis for supervised dimensionality reduction. In: International Conference on Machine learning, pp. 905–912 (2006)
13. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. Journal of Machine Learning Research 8, 1027–1061 (2007)
14. Wang, X., Huang, X., Gao, J., Yang, R.: Illumination and person-insensitive head pose estimation using distance metric learning. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 624–637. Springer, Heidelberg (2008)
15. He, X., Ji, M., Bao, H.: Graph embedding with constraints. In: IJCAI, pp. 1065–1070 (2009)
16. Yang, M.H.: Extended isomap for pattern classification. In: ICPR (2002)
17. He, X., Yan, S., Hu, Y., Niyogi, P., Zhang, H.J.: Face recognition using laplacianfaces. PAMI 27, 328–340 (2005)
18. Chen, H.T., Chang, H.W., Liu, T.L.: Local discriminant embedding and its variants. In: CVPR, pp. 846–853 (2005)
19. Zhao, Q., Zhang, D., Lu, H.: Supervised LLE in ICA space for facial expression recognition. In: International Conference on Neural Networks and Brain, pp. 1970–1975 (2005)
20. Teoh, A.B.J., Pang, Y.H.: Analysis on supervised neighborhood preserving embedding. IEICE Electronics Express 6, 1631–1637 (2009)
21. Nilsson, J., Sha, F., Jordan, M.I.: Regression on manifolds using kernel dimension reduction. In: International Conference on Machine Learning, pp. 697–704 (2007)
22. Balasubramanian, V.N., Ye, J., Panchanathan, S.: Biased manifold embedding: a framework for person-independent head pose estimation. In: CVPR (2007)
23. Balasubramanian, V.N., Krishna, S., Panchanathan, S.: Person-independent head pose estimation using biased manifold embedding. Eurasip Journal on Advances in Signal Processing 2008, 1–15 (2008)
24. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: ICCV, pp. 1208–1213 (2005)
25. He, X., Niyogi, P.: Locality preserving projections. Advances in Neural Information Processing Systems 16, 100–200 (2004)
26. Little, G., Krishna, S., Black, J., Panchanthan, S.: A methodology for evaluating robustness of face recognition algorithms with respect to variations in pose angle and illumination angle. In: International Conference on Acoustics, Speech, and Signal Processing (2005)
27. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: A general framework for dimensionality reduction. PAMI 29, 40–51 (2007)
28. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Journal of Neural Computation 15, 1373–1396 (2003)
29. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
30. Zeng, X., Luo, S.: A supervised subspace learning algorithm: Supervised neighborhood preserving embedding. In: International Conference on Advanced Data Mining and Applications (2007)
31. de Ridder, D., Kouropteva, O., Okun, O., Pietikùinen, M., Duin, R.P.: Supervised locally linear embedding. In: International Conference on Artificial Neural Networks and Neural Information Processing (2003)
32. Li, C.G., Guo, J.: Supervised isomap with explicit mapping. In: Innovative Computing, Innovation, and Control (2006)
33. Tenenbaum, J.B., Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)

Knowledge Based Activity Recognition with Dynamic Bayesian Network

Zhi Zeng and Qiang Ji

Rensselaer Polytechnic Institute,
Troy, NY, 12180, USA
{zengz,jiq}@rpi.edu

Abstract. In this paper, we propose solutions on learning dynamic Bayesian network (DBN) with domain knowledge for human activity recognition. Different types of domain knowledge, in terms of first order probabilistic logics (FOPLs), are exploited to guide the DBN learning process. The FOPLs are transformed into two types of model priors: structure prior and parameter constraints. We present a structure learning algorithm, constrained structural EM (CSEM), on learning the model structures combining the training data with these priors. Our method successfully alleviates the common problem of lack of sufficient training data in activity recognition. The experimental results demonstrate simple logic knowledge can compensate effectively for the shortage of the training data and therefore reduce our dependencies on training data.

1 Introduction

During recent years, probabilistic graphical models have received increasing attention in computer vision research, such as image segmentation, object tracking and facial expression analysis. DBNs, which are designed to model temporal events, are widely adopted for recognizing human activity. Most of the existing DBN models for activity recognition are learned purely from training data, so when the amount of training data is insufficient, the performance of these models will decrease significantly. One solution to alleviate this problem is resorting to various kinds of domain knowledge.

First order logic is an expressive language in representing the logic relations in a domain and it is widely applied in many computer vision applications. Its combination with Markov networks, the Markov logic networks (MLN), can deal with rigorous logic reasoning while maintaining the capability of handling uncertainty. However, the construction of MLN requires relatively complete knowledge of the domain. If the knowledge is limited, it may lead to a highly biased model.

In our work, we first introduce a generic DBN model integrating multiple features for activity recognition, and then present a framework to learn the DBN model combining training data with domain knowledge. The domain knowledge is represented by a set of ffigure first-order probabilistic logics, which can be further transformed to the structure prior and qualitative parameter constraints on the activity model. These prior combined with the training data are used to

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 532–546, 2010.

learn the DBN structure and parameters in a CSEM framework. With simple and generic qualitative knowledge, we obtain more representative DBN structures and accurate parameters that produce better activity recognition results.

2 Related Work

Various types of DBNs have been proposed for recognizing different activities in the literature. Standard HMM [1][2] is employed for simple activity recognition, but it is not suitable for modeling complex activities that have large state and observation spaces. Different variants of HMM try to solve this problem through factorizing the state or observation space. Parallel HMMs (PaHMMs) [3], coupled HMM (CHMM) [4] and dynamic multiply-linked HMM(DML-HMM)[5] are proposed to recognize group activities by factorizing the state spaces into several temporal processes. PaHMMs ignore the interactions between different temporal processes except a zero-order synchronization, CHMM model the interactions among multiple objects through completely coupling the temporal processes, while DML-HMM tries to discover the necessary coupling links between these processes. In comparison, layered HMM [6], switching hidden semi-Markov models [7] and Hierarchical HMM [8] try to model activity at multiple levels, with the upper layers encoding the transitions among the high-level states (such as the constituent actions) and the bottom layer encoding the transitions among the low-level states (such as action primitives). Xiang et al. [9] introduce the multiple observation HMMs that factorize the observation space into several conditional independent factors to recognize activity with large dimensional feature vectors.

As the HMM variants are still restricted by their specific model structure, more general DBNs are also employed for activity modeling. Wu et al. [10] present a DBN that combines RFID and video data to infer the activity and object labels. Their model is essentially a layered HMM with multiple observations. Besides, Laxton et al. [11] define a hierarchical DBN leveraging temporal, contextual and ordering constraints to recognize complex activities.

The model structures of the above approaches, except DML-HMM, are all manually specified. For DML-HMM, only the coupling links are learned from training data. Moreover, these approaches assume that all the activities share the same model structure and sufficient training data are available to learn the models. In contrast, we are able to learn DBN structure for each activity, even when data are insufficient, with logic knowledge exploited from activity domain.

The first-order logics (FOLs) received increasing attention in computer vision due to its expressive power on interpreting knowledge in different domain. Recent researchers [12][13] begin to investigate Markov logic networks (MLN) [14], a combination of FOLs with Markov network, in activity recognition. While MLN successfully integrates logic reasoning with data-driven inference in activity recognition, there are still several points to be considered: first, the MLN can not represent naturally causal relationships between domain elements, which are common in human activity; second, we can view the structure of the MLN as completely specified by the prior knowledge, since the potentials corresponds to the logic groundlings. In case the logic knowledge is inaccurate, the constructed MLN can not work well in activity recognition. In comparison, we choose to represent the domain knowledge with first-order probabilistic logics, and combine these knowledge with training data to learn both the DBN structures and parameters, which can incorporate approximate and partial knowledge.

In knowledge-based learning field, Tong et al. [15] have investigated qualitative constraints for BN parameter learning; however, the qualitative knowledge are expressed heuristically and not exploited for structure learning. In our work, with structure prior and parameter constraints obtained from domain knowledge, we propose a constrained structural EM algorithm to learn DBN structure combining incomplete training data these knowledge. Compared with the structural EM algorithm [16], the constrained structural EM algorithm is different at two aspects: firstly, it can estimate more reliable parameters for the candidate structures under the guide of the constraints; secondly, with the structure prior generated from the domain knowledge, we are able to employ the posterior probability rather than marginal likelihood (BIC) score for model evaluation.

3 Modeling Activity with DBN

3.1 Image Features

The feature set we used for activity recognition consists of the position, speed, shape and spatio-temporal features. For feature extraction, we first perform motion detection to detect the moving object and to extract its silhouette. Position O_Y is then measured as the distance to a reference point[1], speed O_V is evaluated as the change of the object center in pixels and the shape feature O_S includes four elements: aspect ratio of the bounding box of the moving object, filling ratio (the area of the object silhouette with respect to the area of the bounding box) and two first-order moments of the silhouette [9]. The spatio-temporal feature O_{ST} we use is the histogram of optical flow in the spatio-temporal cube.

3.2 DBN Model for Activity Recognition

As we usually observe the activity through object position, shape, speed and spatio-temporal features from the image sequence, the underlying states of these measurements provide a good representation of the activity state space. We can decompose the state X_t into a set of physical states corresponding to position state Y_t, shape state S_t, global speed state V_t and spatio-temporal state ST_t. Accordingly, the measurement O_t consists of four observations: OY_t, OS_t, OV_t and OST_t. Figure 1 shows an example of our DBN model for activity modeling. Besides nodes, there are two types of links in our model: intra-slice links and inter-slice links. While intra-slice links capture the relationships between states, and between states and their corresponding measurements. The inter-slice links

[1] We use the starting position as the reference point for Weizmann dataset and the car position as the reference point for the Parking lot dataset.

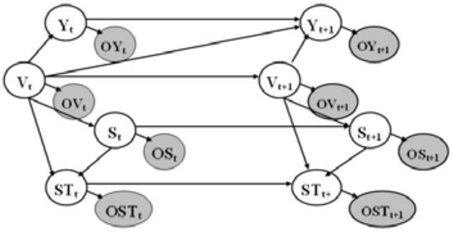


Fig. 1. Example DBN model for activity recognition

capture the dynamic relationships between states at different times. Except for the links between states and their observations, other links are learnt. Please note the links in figure 1 are just for illustration and do not always represent the true dependencies between the underlying states of different features. In next section, we will discuss how to find these dependencies through DBN structure learning. With the above modeling strategy, we can construct one DBN model for each activity and perform activity recognition through finding the model with the highest likelihood, which can be evaluated by the forward propagation of dynamic junction tree algorithm[17].

4 Knowledge Representation in Human Activity

For many computer vision applications, there often exists some approximate domain knowledge that governs the physics, kinematics, and dynamics of domain objects. Such knowledge, if exploited, can help regularize the otherwise ill-posed problems. In activity recognition, we can identify such knowledge in the form of logic rules, which can be feature-related or activity-specific. The simple feature-related low-level rules govern the formulation of most activities. Such rules are activity independent and the same rules can be applied to different types of activities. The activity-specific constraints, on the other hand, are related to the object types, interactions and dynamics for specific activities. In this paper, since the focus is single-level activity recognition, we mainly exploit the feature-related rulesin the form of first-order probabilistic logics, and then try to incorporate these knowledge in our activity model.

4.1 FOPL in Human Activity

First-order probabilistic logics is one type of knowledge representation language preserving the expressive power of first-order logic while introducing the probabilistic treatment of uncertainty. While several families of FOPLs have been proposed in the literatures [18], we keep the formal syntax and semantics defined by Halpern et al. [19].

The alphabet we used to represent the knowledge in human activity includes:

- Predicates: Is;

- Constants: POS(position), SH(shape), SP(speed), ST(spatio-temporal response), near (NR), far (FA), simple (SI), complex(CO), high(HI), low(LO);
- Function: Next;
- Connective symbols: $\vee, \wedge, \forall, \neg, |$;
- Variable: t, AS (denotes one of the three constants: POS, SH and SP), s;
- Probability operator: Pr;
- Basic numeric operator: +, *, =, >;

With the defined alphabet, we can describe the domain elements with two sorts of terms: the object term and numeric term. While the object term describes the non-numeric basic elements (i.e. "t", "shape", "position", "Next(t)") of the domain, the numeric term describes certain probabilities which are rational numbers in the interval [0 1] (i.e. Pr(Is(position, near, t))). Given these elements, we can interpret the logics of the activity domain with a set of well-formed formula, which, in our case, only consists of the relations between different probabilities. The logics we exploit in human activity include:

- Smoothness Logic

  This type of logic interprets the general knowledge about the smooth transitions between the states of the activity, and it is applicable to all states of the activities. *Logic rule*: the object is more likely to keep the previous state than transit to other states.

  $$Pr[Is(AS, s, Next(t)) \mid Is(AS, s, t)] \geq Pr[Is(AS, s, Next(t)) \mid \neg Is(AS, s, t)]$$

  Exemplar Instantiation: the speed of an object at a successive time is more likely to be low if its current speed is low than if its current speed is high.

  $$Pr[Is(SP, LO, Next(t)) \mid Is(SP, LO, t)] \geq Pr[Is(SP, LO, Next(t)) \mid Is(SP, HI, t)]$$

  This logic formula, in simplicity, can be transformed to a probabilistic constraint on the conditional probabilities of our activity model.

  $$P(V_{t+1} = L | V_t = L) \geq P(V_{t+1} = L | V_t = H)$$

  Here L denotes the low speed state and H denotes the high speed state.
- Position-motion Logic

  The position-motion logic encodes the logic relationship between the position and moving speed of the subject.

  Logic rule: The object is more likely to keep the same position state with low speed than with high speed, and meanwhile it is more likely to change position state with high speed than with low speed.

  $$\begin{aligned} &Pr[Is(POS, s, Next(t)) \mid Is(POS, s, t)] \wedge Is(SP, low, t)] \\ &\geq Pr[Is(POS, s, Next(t)) \mid Is(POS, s, t) \wedge Is(SP, high, t)]; \\ &Pr[\neg Is(POS, s, Next(t)) \mid Is(POS, s, t)] \wedge Is(SP, high, t)] \\ &\geq Pr[\neg Is(POS, s, Next(t)) \mid Is(POS, s, t) \wedge Is(SP, low, t)] \end{aligned}$$

Exemplar instantiation: With a high speed and near position in current frame, an object is more probable to be in far position in next frame than with a low speed and near position in current frame.

$$Pr[Is(POS, FR, Next(t)) \mid Is(POS, NR, t)] \wedge Is(SP, HI, t)]$$
$$\geq Pr[Is(POS, FR, Next(t)) \mid Is(POS, NR, t) \wedge Is(SP, LO, t)]$$

Similarly, we can transform this logic formula to a probabilistic constraint on conditional probabilities of the activity model.

$$P(Y_{t+1} = F|Y_t = N, V_t = H) \geq P(Y_{t+1} = F|Y_t = N, V_t = L)$$

Here N denote near position state; F: far position state.

- Shape-motion logic
There are also logic relationships between the shape and speed of the subject
Logic rule: Shape change is more likely to occur when speed is low.

$$Pr[\neg Is(SH, s, Next(t)) \mid Is(SH, s, t)] \wedge Is(SP, low, t)]$$
$$\geq Pr[\neg Is(SH, s, Next(t)) \mid Is(SH, s, t) \wedge Is(SP, high, t)]$$

Exemplar instantiation: It is more probable for an object to change from simple shape to complex shape with a low speed than with a high speed.

$$Pr[\neg Is(SH, CO, Next(t)) \mid Is(SH, SI, t)] \wedge Is(SP, LO, t)]$$
$$\geq Pr[\neg Is(SH, CO, Next(t)) \mid Is(SH, SI, t) \wedge Is(SP, HI, t)]$$

This formula can similarly be transformed to a probabilistic constraints on the activity model, which is:

$$P(S_{t+1} = 1|S_t = 0, V_{t+1} = L) \geq P(S_{t+1} = 1|S_t = 0, V_{t+1} = H)$$

Here $S_t = 1$ denotes complex shape and $S_t = 0$ denotes simple shape.

- Spatio-temporal Logic The spatio-temporal logic encodes the relationship between the spatio-temporal state and the shape change.
Logic rule: It is more probable to have high spatio-temporal response if the object undergoes shape change, than the object stays in the same shape.

$$Pr[\neg Is(ST, high, Next(t)) \mid Is(SH, s, t)] \wedge \neg Is(SP, s, Next(t))]$$
$$\geq Pr[\neg Is(ST, high, Next(t)) \mid Is(SH, s, t) \wedge Is(SP, s, Next(t))]$$

Exemplar instatiation: An object is more likely to have a high spatio-temporal response if it has a simple shape in current frame and a complex shape at next frame, than if its shape at current frame and next frame are both simple.

$$Pr[\neg Is(ST, HI, Next(t)) \mid Is(SH, SI, t)] \wedge \neg Is(SP, CO, Next(t))]$$
$$\geq Pr[\neg Is(ST, HI, Next(t)) \mid Is(SH, SI, t) \wedge Is(SP, SI, Next(t))]$$

The probabilistic constraints transformed from this logic formula is:

$$P(ST_{t+1} = 1|S_t = 0, S_{t+1} = 1) \geq P(ST_{t+1} = 1|S_t = 0, S_{t+1} = 0)$$

Here $ST_t = 1$ is the high spatio-temporal response and $ST_t = 0$ is low spatio-temporal response.

4.2 Incorporate FOPL in Activity Model

The discussion above exploits different types of domain knowledge in the form of FOPL, which can be transformed to a set of qualitative constraints on the conditional probabilities of the activity model. Now we begin to investigate how to incorporate these knowledge in our DBN model. Two types of model prior can be generated from these logic knowledge.

Parameter Constraints. First, the domain knowledge, in terms of qualitative constraints on the model conditional probabilities, can be used to regularize the parameter learning for the activity model. However, they are not necessarily the constraints on the parameters of the activity model. For example, the smoothness logic for the position state finally involves the conditional probability $P(Y_{t+1}|Y_t)$. With the example model structure in figure 1, Y_t is not the only parent of Y_{t+1}, which means $P(Y_{t+1}|Y_t)$ is not a model parameter. Thus, we still need to translate the constraints on state variables into the constraints on the model parameters. For example, if we are expected to impose the following constraint related to the conditional probability $P(A|B)$,

$$P(A = k_1|B = j_1) \geq P(A = k_2|B = j_2) \tag{1}$$

There are three possible cases according to model structure,

- B is the only parent of A: we can directly impose this constraint as $P(A|B)$ is the model parameter;
- B is not the parent of A: we do not impose this constraint as this constraint will become highly nonlinear if we represent the conditional probability $P(A|B)$ using the model parameters. In this case, the logic knowledge will be described by the structure prior, which penalize the absence of the link from B to A by the structure prior.
- B is, but not the only parent of A: Let C be the other parents of A, as $P(A|B) = \sum_C P(A|B,C)P(C|B)$, constraint in equation 1 becomes:

  $$\sum_l P(A = k_1|B = j_1, C = l) \cdot P(C = l|B = j_1) \geq \sum_l P(A = k_2|B = j_2, C = l) \cdot P(C = l|B = j_2)$$

  where l is the configuration of C. Approximating $P(C = l|B = j)$ by the expected sufficient statistics $n_{C=l,B=j}/n_{B=j}$, the above equation becomes a linear constraints on model parameter $P(A|B,C)$.

With the above strategy, the qualitative constraints can be translated to a set of linear constraints on the model parameters θ, denoted as $g_c(\theta) = a_c^T \theta - b_c \leq 0$, where a_c and b_c are the coefficients for constraint c.

Structure Prior. The existing approaches combining logic knowledge with Bayesian network often assume the existence of edge from the conditioning variable to the dependent variable, which can be viewed as hard structural constraints. In our work, we alleviate this hard constraints to a soft structure prior, which can then allow imperfect specification of the domain knowledge to certain degree. The structure prior, together with the training data, are used to learn the model structure in a Bayesian manner as we will discuss in next section.

We set the prior probabilities of the candidate structures through measuring their consistency with the logics, which is defined as:

$$P(S) = ak^{\delta_{S,C}} \tag{2}$$

where a is a normalization constant, k is a constant factor between 0 and 1 controlling the prior strength and $\delta_{S,C}$ is the total number of logic links that are absent from structure S[2]. The intuition for defining this structure prior is to penalize the model structures that are inconsistent with our domain knowledge.

5 Knowledge Based DBN Learning

In this section, we focus on incorporating the domain knowledge in the process of learning the activity model. As the dependencies among the state variables are not apparent, and different activities may have different state dependencies, discovering the DBN structure is a key step for constructing the activity model. In general, the objective of structure learning is searching for a network that fits the best with the prior knowledge and the training data. A complete structure learning scheme requires two components: a criterion to measure how well a candidate structure fits with the prior knowledge and the data, and a model searching strategy used to find the structure with the highest score by the criterion.

5.1 Criterion for Model Selection

A widely used criterion for learning the DBN structure is the BIC score. According to [20], the BIC score $BIC(S)$ can be considered as an approximation of the log marginal likelihood $\log P(D|S)$ of the structure S using Laplacian approximation.

When the prior of the candidate structures is readily available for our activity model, we can learn the model structure in a Bayesian manner, which uses the log posterior probability (LPP) as the criterion for model selection[3]:

$$\begin{aligned} Q(S) = \log P(S|D) &= \log P(D|S) + \log P(S) - \log P(D) \\ &\approx L(\hat{\theta}_S) + \log(ak^{\delta_{S,C}}) - \frac{d}{2}\log N - \log P(D) \end{aligned} \tag{3}$$

here θ_S is the parameter for structure S , $L(\hat{\theta}_S)$ is the log likelihood of $\hat{\theta}_S$, d is the number of parameters in S, N is the number of samples from all sequences.

[2] A logic link is defined as follows: if the logic constraint finally involves conditional probability $P(A|B)$, link $B \rightarrow A$ is a logic link.

[3] Since $\log P(D)$ is a constant, we can ignore it for model comparison.

5.2 Model Search

With incomplete training data, a widely adopted approach for DBN model search is the structural EM (SEM) algorithm [16]. One bottleneck of the SEM algorithm is that it requires a large amount of training sequences. Since the data is often limited, but there exists very generic logic knowledge in terms of qualitative constraints about the human activities, we propose the constrained structural EM (CSEM) algorithm to learn the model structure combining the training data with these constraints.

Before introducing the CSEM algorithm, we define the related notations as follows[4]: θ denotes the parameter of a given DBN structure, $L(\theta) = \log P(D|\theta)$ and $EL(\theta) = E_z[\log P(D, z|\theta)]$ is the log-likelihood and expected log-likelihood of θ respectively, i is the node index, k is the state of node i, n_{ijk} is the expected count of the cases in all the transition slices that node i has the state k with parent configuration j.

Given these definitions, the procedure of the CSEM algorithm is summarized in algorithm 1.

Algorithm 1. Constrained structural EM algorithm

```
For $n = 0, 1, \ldots$ until convergence
**E-step**

  1. Estimate the parameter $\theta_n$ of the current model structure $S_n$ with the qualitative
     constraints;
  2. Find all the local candidate structures of $S_n$ through adding, removing or re-
     versing one link from $S_n$ (we only change the links between the state nodes and
     do not reverse the temporal links);
  3. "Complete" the data based on $S_n$ and $\theta_n$ and compute the expected counts for
     all candidate structures
  4. For each candidate structure $S$, estimate the parameter $\theta_S$ through maximizing
     the expected log likelihood $EL(\theta_S)$ subject to the constraints;
  5. For each candidate structure $S$, compute the expected LPP score $EQ(S)$

        $EQ(S) = EL(\theta_S) + \log(ak^{\delta_{S,C}}) - \frac{d}{2} \log N - \log P(D)$

**M-step**

  1. Set $S_{n+1}$ to be the structure with the highest expected score;
```

In E-step 1, we employ the constrained EM (CEM) algorithm to estimate the parameter θ_n for model structure S_n. The E-step of the CEM algorithm is the same as the traditional EM algorithm, which first "complete" the data based on the current parameter and then compute the expected counts $\{n_{ijk}\}$. The M-step of the CEM algorithm finds the new parameters that maximizes $EL(\theta)$ subject to the set of parameter constraints $g_c(\theta) \leq 0$ that we discussed in section 4.2. We formulate this step as a constrained optimization problem:

[4] Strictly the defined terms should depend on a given structure S. we ignore S in the notation just for simplicity.

$$\begin{aligned} \max_{\theta} \quad & EL(\theta) = \sum_i \sum_j \sum_k n_{ijk} \log \theta_{ijk} \qquad (4) \\ s.t. \quad & \sum_k \theta_{ijk} = 1 \;\; \forall i, j \quad , \quad g_c(\theta) \leq 0 \;\; \forall \text{ constraint } c \end{aligned}$$

In E-step 4, we need to estimate the parameter θ through maximizing the expected log-likelihood $EL(\theta_S)$ for each candidate structure S. Since the expected counts $\{n_{ijk}\}$ are available from E-step 3, we can also estimate θ_S through solving the optimization problem in equation 4.

With the CSEM algorithm, the logic knowledge can influence the expected score $EQ(S)$ of the candidate structures in two ways: first they control the prior probabilities of the structures; secondly, they can regularize the parameter estimation for each structure and then alter the expected log-likelihood score. When the training data is limited, adding the structure prior or regularizing the parameter estimation can help improve structure learning by avoiding some local maxima caused by the noisy data in structure search process.

The CSEM algorithm is guaranteed to achieve a local optimum since it improves the model score ($Q(S)$) at each step. The proof of convergence is similar to the SEM algorithm with small difference on handling the structure prior.

5.3 Learning Activity-Dependent DBNs

People usually assume all the activities share the same model structure; the real activities, however, do not have the same dependency among the basic states. For example, the dependency between the shape and speed varies from activity to activity. People usually keep similar shape in *walking*, so the dependency between the speed and shape is weak. In comparison, this dependency is strong for *bending* as people usually undergoes large shape variation during the bending process. Thus, we learn both the model structure and parameter which capture the dependency type and strength for each activity respectively.

6 Experiments

6.1 Weizmann Dataset

The weizmann dataset contains 10 different behaviors performed by 9 people. There are total of 93 video sequences. In the experiments, we learn the DBN models with different number of training sequences (1, 3, 5, 8). The knowledge base used include 8 smoothness logic groundings, 4 position-motion logic groundings, 4 shape-motion logic groundings and 4 spatio-temporal logic groundings.

Evaluation on Activity-Dependent Structure. Figure 2 compares the activity recognition performance of activity-independent model and activity-dependent model on Weizmann dataset. We can find that the activity-dependent model outperforms activity-independent model almost in all cases (with 1 training sequences the performance is quite close). We also include the results for

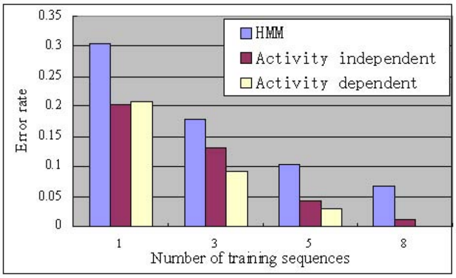


Fig. 2. Comparison of activity-independent DBN, activity-dependent DBN and HMM on Weizmann dataset

HMM with exactly the same set of features. It is easy to get from figure 2 that our DBN model outperforms the baseline HMM model significantly through explicitly modeling the dependencies among different features.

Evaluation on CSEM for Activity-Dependent Structure Learning. In table 1, we report the recognition results of the knowledge-based CSEM algorithm and the data-based SEM algorithm. With 5 and 8 training sequences, the advantage of CSEM algorithm over SEM algorithm is not significant since we have already obtained nearly perfect recognition result with SEM algorithm. However, when data becomes scarce, the CSEM algorithm gradually shows its superiority over SEM algorithm. In case of 1 training sequence, the activity-dependent model learned with CSEM algorithm outperforms the model learnt with SEM algorithm by 6.5% with same set of image features.

Comparison with Other Approaches. Since the results reported by the state-of-art approaches on Weizmann dataset are evaluated using leave-one-out cross validation, it is hard to compares our algorithm with them in the case of insufficient data. Thus, we compare our result with these approaches using 8 training sequences for each activity. Table 2 shows the comparison of our work with previous approaches. Our activity-dependent DBN models achieve the state-of-art performance on Weizmann dataset.

6.2 KTH Dataset

The KTH dataset consists of 600 video clips with 6 human activities, each of which is performed by 25 subjects in four different scenarios: outdoors, outdoors

Table 1. Recognition error of activity-dependent structures learned with CSEM and SEM

# Training sequences	1	3	5	8
SEM	0.247	0.091	0.028	0.000
CSEM	0.182	0.067	0.019	0.000

Table 2. Comparison with previous work on Weizmann dataset

Our method (SEM)	100%
Our method (CSEM)	100%
Fathi et al. [21]	100%
Jhuang et al. [22]	98.8%
Thurau et al. [23]	94.4%
Niebles et al. [24]	72.8%

with scale variation, outdoors with different clothes and indoors. The knowledge base we used in evaluating our approach on this dataset is exactly the same as on the Weizmann dataset. In the experiments, we vary the number of training sequence for each activity from 50 to 500 to study the effectiveness of the knowledge-based learning on alleviating the dependency on the data.

Table 3 compares the knowledge-based CSEM algorithm with the standard SEM algorithm in learning the activity model with different number of training subjects. We can clearly see that, when the number of training subjects is large, CSEM is only marginally better than SEM algorithm. However, when the number of training subjects becomes smaller, the knowledge we exploited gradually play more important role in activity recognition. With the complement of the logic knowledge, the CSEM algorithm can perform significantly (7.1%) better than the SEM algorithm when the number of training subjects is small.

Table 3. Comparison of CSEM and SEM

# Training Subjects	4	8	12	16	20
EM	0.760	0.828	0.862	0.880	0.892
CSEM	0.831	0.863	0.904	0.921	0.925

We also compare our approach with the state-of-art approaches on this dataset. Similar to the posted results in the literature, I use the data from 16 subjects for training. Table 4 shows that we can achieve comparable result to the state-of-art approaches.

Table 4. Comparison with previous works on KTH dataset

Our method (SEM)	88.0%
Our method (CSEM)	92.1%
Yuan et al. [25]	93.3%
Laptev et al. [26]	91.8%

6.3 Parking Lot Dataset

We also apply our algorithm to the problem of recognizing human activities in the parking lot. The dataset consists of 108 sequences for 7 activities: *walking*

(WK), *running* (RN), *leaving car* (LC), *entering car* (EC), *bending down* (BD), *throwing* (TR) and *looking around* (LA). These activities are performed by several people with scale variation, view change and shadow interference. In the experiment, we randomly split the original dataset into training set and testing set. Different algorithms are compared using training set with 10, 20, 40, 80 sequences. Each size is tested 10 times and the average recognition error is used for evaluation. We use the constraints set as those for the Weizmann dataset.

In figure 3, we compare the knowledge-based CSEM with data-based SEM algorithms in learning both activity-dependent and activity-independent model structures.

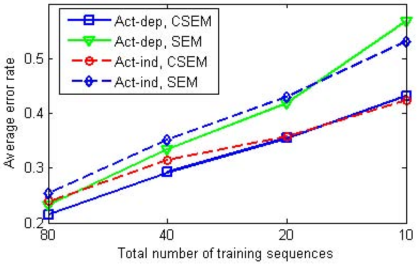


Fig. 3. Comparison of CSEM and SEM for learning activity-dependent and activity-independent models

First, we look at the performance of the activity-dependent models learnt with the CSEM algorithm and SEM algorithm. As the number of training sequences decreases, the CSEM algorithm gradually shows its advantage over SEM, which means our knowledge in terms of constraints play more and more important roles on regularizing the structure learning as data size decreases.

From figure 3, we can also find that, with 20 or 10 training sequences, the activity-dependent model obtains comparable results with activity-independent model learnt using CSEM with the same data size, while it performs worse if we learn the structure without constraints. Moreover, the activity-dependent model with CSEM learning (method 1) requires only half training data to obtain comparable result to activity-independent model with SEM learning (method 2) when the data is insufficient. Specifically, with only 10 training sequence, the recognition error of method 1 is 43.2%, while the recognition error of method 2 is 43.0% given 20 training sequence. With 20 training sequence, the recognition error of method 1 is 35.5%; in comparison, the recognition error of method 2 is 35.2% given 40 sequences. Thus, we can see that exploiting the generic logic knowledge in the activity can greatly alleviate the problem of insufficient data.

Table 5 reports the recognition result of the activity-dependent models learnt with CSEM algorithm on 80 training sequences. Our algorithm can correctly classify 78.6% of the testing sequences. The result is reasonable since the misclassifications occur between similar activities (i.e. *walking* and *looking around*), or for the activities with poor observation (i.e. *leaving car* and *entering car*)

Table 5. Confusion table of the activity recognition test on activity-dependent models learnt with constraints

	WK	RN	LC	EC	BD	TR	LA
WK	.90	.06	.00	.00	.00	.00	.04
RN	.08	.88	.00	.00	.00	.04	.00
LC	.00	.00	.65	.25	.10	.00	.00
EC	.00	.00	.35	.60	.00	.05	.00
BD	.02	.00	.04	.00	.80	.08	.06
TR	.00	.10	.00	.04	.12	.72	.02
LA	.125	.025	.025	.00	.025	.05	.75
	Overall Accuracy: 78.6%						

7 Conclusion

In this paper, we focus on exploiting prior knowledge from human activity domain and investigating a constrained structure learning method to learn activity model combining these prior knowledge with training data. Our contributions include : first, we exploit various generic while effective domain knowledge in the form of first-order probabilistic knowledge; second, after transforming the FOPLs to the structure prior and qualitative parameter constraints, we propose a constrained DBN learning approach to combine domain knowledge with training data. The experimental results demonstrate the effectiveness of our knowledge-based learning scheme in reducing the dependence on training data and alleviating the over-fitting problem when data is insufficient. It also shows promise of the activity-dependent structures in improving activity recognition. Although our learning framework is only tested on single-subject activity recognition, we are planning to apply it to multi-subject and more complex activity recognition in the future.

Acknowledgement

This project is supported in part by Defense Advanced Research Projects Agency (DARPA)'s Video and Image Retrieval and Analysis Tool (VIRAT) program under Contract No. HR0011-08-C-0135 via Kitware.

References

1. Yamato, J., Ohaya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden markov model. In: CVPR (1992)
2. Zhang, D., Perez, D., McCowan, I.: Semi-supervised adapted hmms for unusual event detection. In: CVPR (2005)
3. Vogler, C., Metaxas, D.: A framework for recognizing the simultaneous sspects of american sign language. In: CVIU (2001)

4. Oliver, N.M., Rosario, B., Pentland, A.P.: A bayesian computer vision system for modeling human interations. PAMI (2000)
5. Xiang, T., Gong, S.: Beyond tracking: Modelling activity and understanding behavior. In: IJCV (2006)
6. Oliver, N., Horvitz, E., Garg, A.: Layered representation for human activity recognition. CVIU (2004)
7. Duong, T., Bui, H., Phung, D.: Activity recognition and abnormality detection with the switching hidden semi-markov model. In: CVPR (2005)
8. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and viterbi path searching. In: CVPR (2007)
9. Xiang, T., Gong, S.: Video behavior profiling for anomly detection. PAMI (2008)
10. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Regh, J.: A scalable approach to activity recognition based on object use. In: ICCV (2007)
11. Laxton, B., Lim, J., Kriegman, D.: Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In: CVPR (2007)
12. Tran, S., Davis, L.: Event modeling and recognition using markov logic networks. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 610–623. Springer, Heidelberg (2008)
13. Biswas, R., Thrun, S., Fujimura, K.: Recognizing activities with multiple cues. In: IEEE Works. on Human Motion (2007)
14. Richardson, M., Domingos, P.: Markov logic networks. In: Machine Learning (2006)
15. Tong, Y., Ji, Q.: Learning bayesian network with qualitative constraints. In: CVPR (2008)
16. Friedman, N.: The bayesian structural em algorithm. In: UAI (1998)
17. Murphy, K.: Dynamic bayesian networks: representation, inference and learning. Ph.D. dissertation, University of California (2002)
18. Milch, B., Russell, S.: First-order probabilistic languages: Into the unknown. In: ILP (2006)
19. Halpern, J.: An analysis of first-order logics of probability. Artificial Intelligence (1990)
20. Heckerman, D.: A tutorial on learning with bayesian networks. Learning in Graphical Models (1999)
21. Fathi, A., Mori, G.: Action recognition by learning mid-level motion features. In: CVPR (2008)
22. Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologicallyinspired system for action recognition. In: ICCV (2007)
23. Thurau, C., Hlavac, V.: Pose primitive based human action recognition in videos or still image. In: CVPR (2008)
24. Niebles, J., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: CVPR (2008)
25. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR (2009)
26. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human acions from movies. In: CVPR (2008)

View and Style-Independent Action Manifolds for Human Activity Recognition

Michał Lewandowski, Dimitrios Makris, and Jean-Christophe Nebel

Digital Imaging Research Centre, Kingston University, London, United Kingdom
{m.lewandowski,d.makris,j.nebel}@kingston.ac.uk
http://dircweb.king.ac.uk/

Abstract. We introduce a novel approach to automatically learn intuitive and compact descriptors of human body motions for activity recognition. Each action descriptor is produced, first, by applying Temporal Laplacian Eigenmaps to view-dependent videos in order to produce a stylistic invariant embedded manifold for each view separately. Then, all view-dependent manifolds are automatically combined to discover a unified representation which model in a single three dimensional space an action independently from style and viewpoint. In addition, a bidirectional nonlinear mapping function is incorporated to allow projecting actions between original and embedded spaces. The proposed framework is evaluated on a real and challenging dataset (IXMAS), which is composed of a variety of actions seen from arbitrary viewpoints. Experimental results demonstrate robustness against style and view variation and match the most accurate action recognition method.

Keywords: action manifold, activity recognition.

1 Introduction

Since video recording devices have become ubiquitous, the automated analysis of human activity from a single video is now an essential area of research in computer vision. Applications for such technology include video surveillance, indexing of film archives, sports video analysis and human-computer interactions.

Variability in human shape, appearance, posture and individual style in performing some motion makes the unified description of a given action difficult. In addition, camera view, perspective and scene environment have a critical impact on the aspect of recorded data. Consequently, the task of action recognition from a single video is extremely challenging. In this paper, we propose a solution which deals with this complexity within a single powerful framework. It allows accurate action recognition from a single uncalibrated camera in a fully automatic approach which exhibits high robustness to action style and view variation.

Previous work in this field falls into two categories: view-dependent and view-independent approaches. View-dependent methods assume that all actions are recorded from a fixed viewpoint [3,7,9,1]. The standard approach uses temporal templates to represent an action by encoding the history of silhouette deformation over time [3]. Actions were also described in the space-time domain. Local

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 547–560, 2010.

space-time features were extracted from the volumetric space-time action shape derived from sequence silhouettes by solving the Poisson equation [7]. Alternatively, the structure of local 3D patches was analysed by extending interest points into the spatio-temporal domain [9]. Moreover, by taking into account dynamics, action descriptors were defined in terms of chaotic invariant features from joint tracking [1]. Although these approaches have proved very accurate, the fact they rely on videos captured from a specific view limits their practicality in real world scenarios.

As a consequence, many researchers focused on multiple camera systems to achieve view-invariant action recognition. For instance, 2D temporal templates were extended into 3D motion history volumes [27]. If point correspondences between actions are assumed to be known, then either epipolar geometry [29] or projective invariants of coplanar landmark points can be exploited [19]. The main drawback of these methods is that, since they all require multiple cameras setups, they can only be applied in a controlled environment.

More recently, research has tackled the task of action recognition from an arbitrary view, i.e. from a single video, where multi camera data are used for training. Typically, a database of exemplars from different views is created to recognise actions based on the best matching score. Although silhouettes can be used to represent an action, their intrinsic ambiguity leads to high density sampling of the view space to obtain accurate results [18]. In contrast, richer action descriptors based on 3D exemplars represented by visual hulls and hidden Markov model allow reducing significantly the size of action templates [25]. Consequently, matching between observation and exemplars has to be performed in 2D by projecting visual hulls. Since such projection from high dimensional space to low dimensional is multimodal, it impacts on the quality of the recognition rate [25]. Junejo et al. [8] proposed to represent image sequences using self-similarity based descriptors which are fairly stable under view variation and characterises well the dynamics of the scene. However, this approach relies on the rough localisation and tracking of people in the video [8]. In [28], a video is represented by a combination of 3D visual hulls with spatio-temporal volumes to build 4-dimensional action feature models. Alternatively, a video can be described as a bag of spatio-temporal features called video-words (BOW) by quantising extracted 3D points of interest [16]. Initially, a SVM was trained on BOW to recognise actions [16], but this feature was also extended with a bag of spin-images [15]. Although these schemes perform accurate action recognition, the absence of continuous action model limits their applicability.

The methods most closely related to our approach model activities by reducing dimensionality of each sequence to obtain view-invariant manifold representations [21,6,5]. [21] used R-transform as a descriptor and Isomap [23] for dimensionality reduction, whereas [5,6] chose implicit distance function representation and locally linear embedding [22]. In these approaches [21,5], generative view-independent functions are designed to interpolate between intermediate views. This generative function was also extended to handle stylistic variation of data [6,5]. However, due to the limitations of the chosen dimensionality reduction methods, none of these approaches managed to produce consistent style invariant representations, i.e. representations which are valid for a variety of individuals. Consequently, the accuracy of their systems was limited. This problem was addressed be applying non-rigid transformation [17] to artificially unify manifold representations of different people [21,6]. However, since such transformation affects manifold geometry, they may no longer reflect relationships between points in the high dimensional space. Alternatively, in [5] the topological structure of a torus was artificially constrained on the manifold to explicitly deal with stylistic variation instead of being learned from the data.

The main contribution of this paper is a new continuous view and style invariant action descriptor in a form of an Action Manifold. The proposed descriptor overcomes above limitations, since, not only, it is obtained automatically from labelled training data, but it encapsulates both style and view in a coherent torus-like two-dimensional manifold. The novel procedure used for generating torus-like descriptors takes advantage of several advanced techniques which have never been used in a view independent action recognition. They include Temporal Laplacian Eigenmaps [14] (TLE), Decomposable Generative Model [12] and Poisson Equation [7]. In addition, the method used for determining repetition neighbourhood in the TLE algorithm has been refined to handle for complex and dynamic videos of human actions. Finally, our descriptors are validated in a challenging real-life scenario of a view independent action recognition.

The structure of this paper is organised as follows. First, we describe our framework. This includes the processes of view-dependent discovery, view-independent manifold construction and mapping and a brief description of the dimensionality reduction algorithm. Secondly, the framework is validated quantitatively on a real dataset of human actions. Finally, conclusions and future work are presented.

2 View and Style-Independent Action Manifold

An action can be implicitly defined by a set of videos of a variety of people performing similar movements seen from different cameras. In our work, we aim to produce a single compact and informative model, i.e. action manifold, which represents an action independently from camera views and individuals' styles.

In our framework, the set of videos defining an action includes a variety of individuals, each of them captured on their own by a set of calibrated and synchronised cameras. Moreover, for each action, a video is labelled as a good representative; usually it is captured from a side view. We do not impose restrictions regarding video length variability for a given action and an individual may perform an action several times.

Let Y denote the set of N videos defining an action performed by different people and captured from different views. For a given view, action repetitions and variability of people define action style. Therefore, Y can be defined as $Y = \{Y^{sv}\}_{(s=1..N_s,v=1..N_v)}$, where v denotes the view class index and s is the style index. Each frame y of video is represented by D pixels: $Y^{sv} = \{y_i^{sv}\}_{(i=1..T^{sv})}$,

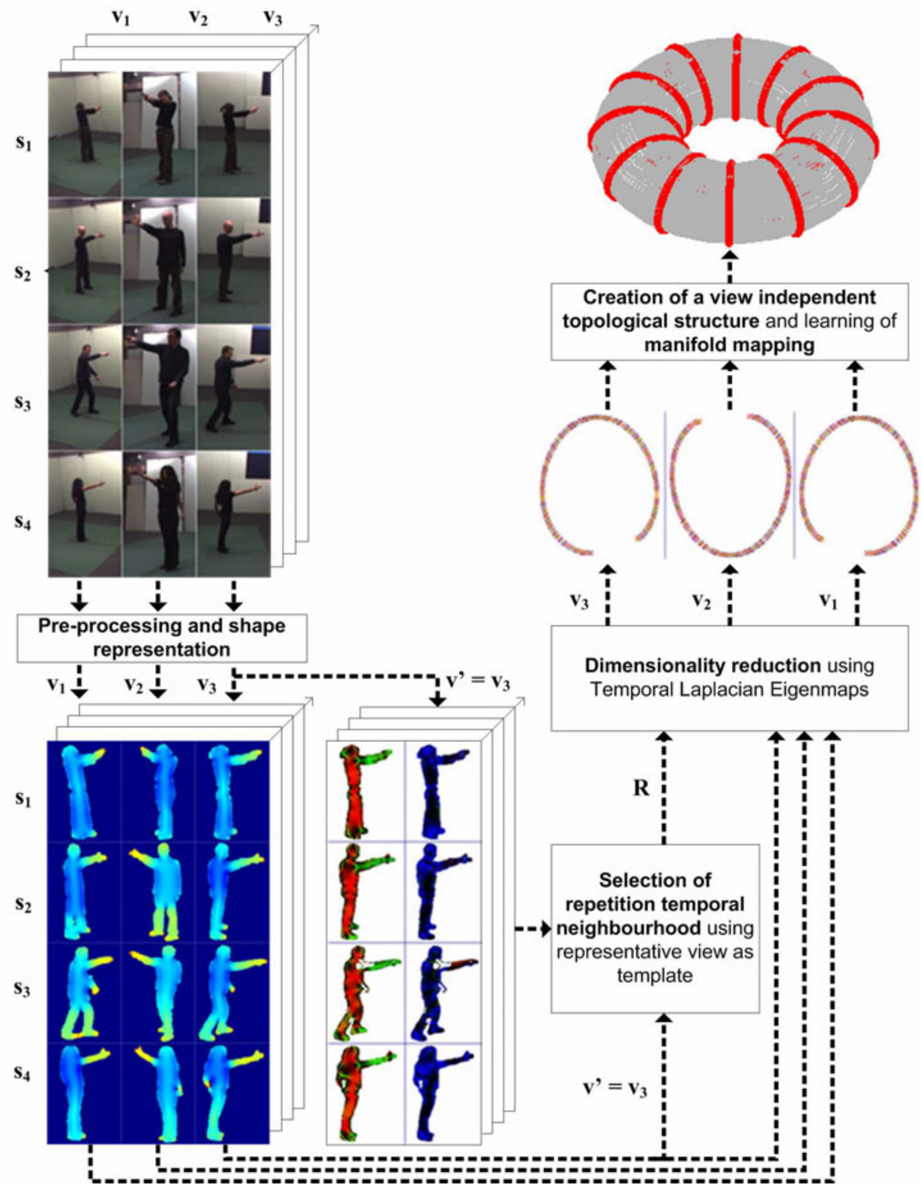

v1
v2
v3
s1
s2
s3
s4
Pre-processing and shape representation
v1
v2
v3
v' = v3
s1
s2
s3
s4
Selection of repetition temporal neighbourhood using representative view as template
v' = v3
R
Dimensionality reduction using Temporal Laplacian Eigenmaps
v3
v2
v1
Creation of a view independent topological structure and learning of manifold mapping

Fig. 1. Description of the action recognition framework for the "point" action

$y_i^{sv} \in R^D$, where T^{sv} is the number of frames in the sequence. Fig. 1 summarises the processing pipeline used to produce a unified and compact action model, X, of dimension $d \ll D$, defined by $X = \{X^{sv}\}_{(s=1..N_s, v=1..N_v)}$, where $X^{sv} = \{x_i^{sv}\}_{(i=1..T^{sv})}$ and $x_i^{sv} \in R^d$.

Our algorithm is divided into two parts. First, view-dependent analysis of action data generates a style invariant action model for each view. This is performed using Temporal Laplacian Eigenmaps, a dimension reduction algorithm with excellent generalisation properties [14]. Then, these models are combined to learn a single compact and view invariant generative model of the action using generative decomposable model [12]. Fig. 1 provides an overview of our method.

2.1 View-Dependent Manifold

Pre-processing and Shape Representation. A frame y_i^{sv} is generally defined by grey scale or colour pixel values. This very high dimensional description makes the process of learning an activity model from a frame sequence costly and inaccurate. However, many studies [18,25,5,6,12] have revealed that a binary representation of moving objects, i.e. silhouettes, are sufficient to capture the activity described by a frame sequence. Consequently, we adopt this approach in our framework.

We extract binary silhouettes y_i^{sv} from each video by a standard background subtraction technique which models each pixel as a Gaussian in RGB space [27]. When videos consist of multiple instances of a given motion, temporal segmentation is required to extract elementary motion segments Y^{sv} [26,4].

All silhouettes are normalised to deal with translation and scale variations by using the largest silhouette square bounding box available within the entire action dataset. In order to improve the quality of the normalised silhouettes, two morphological operations, i.e. bridge and open, and a median filter are applied. Lengths of all sequences Y^{sv} are also normalised to match the length of the shortest sequence T' in the set Y using the standard bicubic spline interpolation technique.

A sequence of binary silhouettes can be considered as a space-time shape surrounded by a closed surface [7]. This allows representing each silhouette by a local space-time saliency feature extracted from the solution of the Poisson equation of the corresponding volumetric surface, which takes into account the time domain [7]. This representation assigns highest gradient values within fast moving limbs which are much more informative for identifying actions, whereas torso has relatively smaller values inside (Fig. 1). As a consequence, such descriptor is significantly more powerful than binary representation [7] and essential, as it will be shown later, in the procedure allowing the selection of the TLE repetition neighbourhoods.

Dimensionality Reduction. Even with the generation of the previously described shape descriptor, the high dimension of Y remains unsuitable for analysis. Consequently, we propose to produce an informative and unified model of the action using a nonlinear dimensionality reduction method. However, most of these techniques [23,22,2,11] cannot handle large variations within a dataset such as an action performed by different people. As a result, they tend to capture the intrinsic structure of each manifold separately without generalisation. Consequently, the common embedded space shows separate and highly distorted manifolds. To deal with this fundamental issue, in this work we use the TLE algorithm which shows excellent generalisation properties [14].

TLE is an unsupervised nonlinear method for dimensionality reduction designated for time series data. It aims to preserve the temporal structure of data manifolds by introducing the concept of simultaneous exploitation of two types of neighbourhood graphs, which express implicitly temporal dependencies between data points. In our framework both graphs are constructed for the view $Y^{v\prime}$ which was labelled as a good representative. Each graph is based on a different definition of neighbour:

a. Adjacent temporal neighbours (A): the next and previous closest points in the sequential order of input.
b. Repetition temporal neighbours (R): the points similar to input but extracted from the different repetitions of activity which may vary in style. The number of R neighbours should match the number of styles N_s contained in the training set $Y^{v\prime}$.

The process of dimensionality reduction can be summarised briefly by the following steps. First, view-dependent weights W^v are assigned to the edges of graph $G' \in \{A, R\}$ to construct graphs for all views G^v using the standard LE formulation [2]. Then for each view the extended cost function is defined to combine information from both graphs:

$$argmin_{X^v}((X^v)^T L^v_A X^v + (X^v)^T L^v_R X^v) \tag{1}$$

$$subject\ to\ (X^v)^T D^v_A X^v + (X^v)^T D^v_R X^v = I \tag{2}$$

where $D^{v,G} = diag\{D^{v,G}_{11}, D^{v,G}_{22}, , D^{v,G}_{T^v T^v}\}$ is a diagonal matrix with entries $D^{v,G}_{ii} = \sum_{j=1}^{T^v} W^{v,G}_{ij}$, and $L^v_G = D^{v,G} - W^{v,G}$ is the Laplacian matrix. The minimum of the objective function can be found by applying Lagrange multipliers to Eq. 1 subject to the constraint expressed by Eq. 2 and solving the generalised eigenvalue problem:

$$(L^v_A + L^v_R)X^v = \lambda(D^v_A + D^v_R)X^v \tag{3}$$

The embedded space X^v is spanned by the eigenvectors given by the d smallest nonzero eigenvalues λ ($d = 2$). The output of this stage is a view-dependent and style-independent one-dimensional action manifold X^v (Fig. 1 and 2b).

Selection of Repetition Temporal Neighbourhood. The size of the repetition neighbourhood corresponds to the number of times an activity is repeated in the training set. Although video lengths were normalised for each action, it cannot be assumed that these videos are synchronous for two reasons. Firstly, they may start on different posture and, secondly, due to style variations, there may not be frame to frame correspondences between two action instances. Consequently, the estimation of the size and location of the repetition neighbourhood is essential. We automatically determine the optimal repetition neighbourhood by adopting the action detection procedure proposed in [7]. This schema is used to find similar motion patterns in each sequence of the training set from which R neighbours can be extracted (see lower part of Fig. 1).

First, the local space-time saliency shape descriptor defined in section 2.1 is extended with a local space-time saliency feature which is composed of 6 local space-time orientation attributes [7]. This allows indentifying regions with vertical, horizontal, and temporal "plates" and "sticks" within body and define orientation local features. Fig. 1 illustrates an example of "plate" and "stick" local features for a good representative view. Blue, red, and green colour regions correspond to temporal, horizontal, and vertical directions of local "plates" and "sticks" [7].

In the next step, a space-time cube is associated to each frame $y_i^{v\prime}$ in a sequence $Y^{v\prime}$ by sliding a warping window in time. The cube, i.e. the global space-time descriptor, combines local shape and orientations features using weighted moments of the form [7]:

$$m_{oqr} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} w(p_x, p_y, t) g(p_x, p_y, t) p_x^o p_y^q t^r dp_x dp_y dt \qquad (4)$$

where p_x,p_y are pixels coordinates, $g(p_x, p_y, t)$ denotes the characteristic function of the space-time shape, $w(p_x, p_y, t)$ is one of the seven possible weighting functions which corresponds to local features. As suggested in [7], spatial and time moments are considered up to order $o + q \leq 2$ and $r \leq 2$ respectively. Each space-time cube is centred around its space-time centroid and uniformaly scaled to preserve spatial aspect ratio.

Secondly, we calculate the matrix M $(N_v \times N_v)$ of Euclidean distances between all space-times cubes among all sequences for a particular view. To emphasise continuity and temporal coherence of the underlying action between sequentially adjacent points in time, we perform temporal windowing of matrix M by averaging distances through time within boundaries of each sequence. This implicitly leads to introducing a temporal history into each data point.

Finally, for each cube we look for the most similar motion pattern in each different repetition of activity based on M. The centre point of each most similar space-time cube becomes a repetition neighbour.

Because of possible substantial differences in speed and imperfect segmentation of action, the repetition neighbours may still not align coherently along time what may result in distortions in the embedded space. To address this problem, we incorporated a neighbourhood refinement procedure. In principle, we accept only these R neighbours for given point P which are within specific range from a corresponding point in each other sequence:

$$R' = \{P_{(i-1)*T+1} - T' \leq R_j \leq P_{iT} + T'\}, i = 2..N_s, j = 1..N_s \qquad (5)$$

where T' is defined as 10% of the normalised sequence length T. As it was mentioned earlier, the entire procedure is performed only once per action for the most discriminative view, because the temporal structure of an action is not view-dependent.

2.2 View-Independent Manifold

Generation of a View-Independent Topological Structure. Discovery of a compact representation of any human activity requires modelling both the view and body configuration jointly in a single space. Here we assume that human motion is observed from different viewpoints along a view circle at fixed camera height. Although such cylindrical setting appears limited, its robustness to view elevation variations, up to 45 degrees as shown in experimental section, makes it appropriate for many real life applications such as visual surveillance and sport analysis [5]. it is important to note that this configuration is not critical to our framework since it can easily be extended to a full view sphere-like model using training videos captured from different camera heights.

In section 2.1 style invariant body configuration manifolds were discovered for each view. Since the embedded spaces share the same topology regardless of the view, see Fig. 1 and 2b, for a given posture there is a unique correspondence on each of these manifolds. Consequently, the connection of those corresponding points in the order of view angle values creates a closed one dimensional manifold (topologically equivalent to a circle) which is the view-independent embedded space of the posture. Therefore, we define the unified representation of an activity as the combined space of the two sets of continuous one dimensional manifolds, i.e. posture and view, which are placed orthogonally to each other.

The process of producing the unified manifold comprises two steps. First, the view-dependent representations are combined: the embedded spaces X^v are aligned with respect to a good representative $X^{v\prime}$ using Procrustes analysis [24]. Since this is a rigid transformation of the spaces, the internal structure of each manifold is not changed. Secondly, each embedded representation X^v is aligned into a three-dimensional structure according to the view angle parameter $\mu^v \in [0, 2\pi]$. The outcome of this procedure reveals a torus-like structure which encapsulates both style and view (Fig. 1 and 2c). We called this structure a view and style-independent action manifold. This result is in line with previous work [5], where the usage of a torus is justified as an ideal representation for modelling

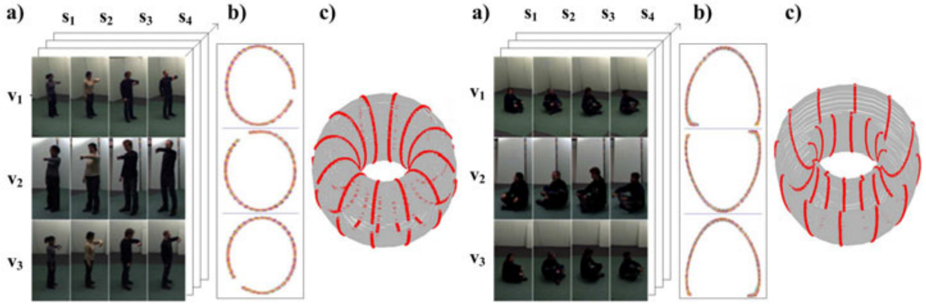


Fig. 2. Training results for quasi periodic action "check watch" (left) and non periodic action "sit down" (right): a) training videos; b) style-independent low dimensional representation for each view; c) style and view-independent manifolds

both the viewpoint and the body configuration of different activities. However, while, in that work, the topological correspondence between data points Y and an ideal torus is artificially enforced, in our approach, the torus-like representation reflects the temporal structure of the view-dependent data. Therefore, in our approach all types of motions, i.e. periodic, quasi-periodic and non-periodic, see Fig. 2, can be handled using the same framework.

2.3 Manifold Mapping

Mapping Function. In the previous section, view descriptors have been combined to form a unique view-independent action manifold. Since TLE is a spectral dimensionality reduction method, there is no mapping function between initial and embedded spaces. However, the ability to project data points from one space to the other is required for classification.

In order to provide a single projection function which allows dealing not only with stylistic variations, but also view changes, a decomposable generative model is learned [12]. This model aims at separating the intrinsic action configuration from other factors such as the motion style and view. Following [12] approach, the generative mapping function is modelled using three factors:

- Content C: a representation of the intrinsic body configuration which characterises motion as a function of time. It is invariant to either person or view.
- Style S: a time-invariant person parameter which describes the person appearance, shape and motion style.
- View point V: a time-invariant view parameter which characterises the view point from which the performed action is captured.

In our framework, content is represented by a continuous manifold while style and view are represented by the discrete classes present in the training data. For the last two factors, intermediate states can be interpolated. As a result, we are able to approximate view and style continuity. In addition, we assume that both style and view factors are time-invariant, i.e. both parameters remain constant during any instance of an action.

The procedure of fitting the decomposable generative model to the data consists of two steps. First, a set of style and view-dependent functions is trained. Then, all functions are combined into a single style and view-independent projection function.

Since mapping between the embedded manifold and the original space is highly nonlinear, generalised Radial Basis Function network [12] is applied to provide the nonlinear view-dependent mapping. It is expressed by N_s style-dependent mapping functions:

$$y^{sv} = B^{sv} * \mathit{\Psi}(x^{sv}) \tag{6}$$

where B is a $D \times E$ matrix of mapping coefficients. The kernel vector $\psi(\textbf{.})$ is defined by:

$$\mathit{\Psi}(x^{sv}) = [\mathit{\Phi}(\| x^{sv} - z_1 \|)..\mathit{\Phi}(\| x^{sv} - z_E \|)\ 1\ x^{sv}]^T \tag{7}$$

where $Z = \{z_i\}_{(i=1..E)}$ is a set of distinctive representative points in each embedded space and $\phi(.)$ is a radial basis function; here we use a thin plate spline. B^{sv} is calculated by applying the Moore-Penrose pseudo-inverse on matrix $\psi(X^{sv})$ and solving a linear system of equations: $B^{sv} = Y^{sv} * \psi(X^{sv})^{+}$ like in [12]. The set Z is obtained by calculating a mean style and view manifold, which is then transformed by a non-rigid point registration procedure, called Coherent Point Drift [17], to better fit the data.

The final view-independent decomposable generative model is obtained by multi-linear tensor analysis in the space of nonlinear mapping coefficients [12]. Each coefficient matrix B^{sv} is represented as the coefficient vector b^{sv} of dimensionality $N_e = D * E$ by column wise stacking (columns of the matrix are concatenated to form a vector). Afterwards, all coefficient vectors b^{sv} are arranged in an order three coefficient tensor B whose dimensionality is $N_s \times N_v \times N_e$. The view and style orthogonal factors are decomposed from the assembled coefficient tensor B using higher order singular value decomposition [10]:

$$B = C \times_1 S \times_2 V \times_3 F = G \times_1 S \times_2 V \tag{8}$$

where S $(N_s \times N_s)$ is the mode-1 basis of B, which represents the orthogonal basis for the style space. Similarly, V $(N_v \times N_v)$ is the mode-2 basis matrix which spans the space of viewpoint parameters and F $(N_e \times N_s * N_v)$ represents the mode-3 basis for the mapping coefficient space. C is a core tensor $(N_s \times N_v \times N_e)$ which governs the interactions between orthogonal factors represented in mode basis matrices. Coefficient eigenmodes G is a new core tensor formed by $G = C \times_3 F$ whose dimensionality is $N_s \times N_v \times N_e$. Mode-i is a tensor product as defined in [10]. As the result, view-independent and style-independent projection function is expressed by equation $y = B * \Psi(x)$.

Action Recognition. The task is performed by projecting a motion sequence into each action descriptor using the generative decomposable model presented in the previous section. Then, the dynamic time warping distance [20] is calculated to measure similarity between actions.

Given a new instance of action $\tilde{Y}^{sv}$, its length is first normalised as described in section 2.1. Then the embedded coordinates $\tilde{X}^{sv}$ of the new action are obtained by least square solution of the following nonlinear system:

$$argmin_{B\Psi} \parallel \tilde{Y}^{sv} - \tilde{B}^{sv}\Psi(\tilde{X}^{sv}) \parallel \tag{9}$$

It's minimum solution can be found by determining and optimising coefficient matrix $\tilde{B}^{sv}$ given a learned model and then projecting data by solving a linear system of equations using the Moore-Penrose pseudo-inverse :

$$\Psi(\tilde{X}^{sv}) = (\tilde{B}^{sv})^{+} * \tilde{Y}^{sv} \tag{10}$$

Coordinates of $\tilde{X}^{sv}$ are provided by the last d rows of the matrix $\Psi(\tilde{X}^{sv})$. In order to determine the optimal coefficient matrix $\tilde{B}^{sv}$, we adopt an iterative procedure [12]. First, we calculate a mean view manifold Z over all aligned mean styles manifolds Z^v to obtain a homeomorphic manifold [12]. Then, the coefficient matrix is initialised by solving the following equation:

$$\tilde{B}^{sv} = \tilde{Y}^{sv} * \Psi(Z)^{+} \tag{11}$$

Let's $\tilde{b}^{sv}$ denote a vector obtained by column wise stacking of matrix $\tilde{B}^{sv}$. Then given a mapping model as described in the previous section and any style vector, $\tilde{s}$, and any view vector $\tilde{v}$, we can define a coefficient vector $\tilde{b}^{sv}$ by the tensor product $b^{\tilde{s}\tilde{v}} = G \times_1 \tilde{s} \times_2 \tilde{v}$.

Mapping coefficients $\tilde{b}^{sv}$ can be optimised to reflect style and view of a new instance action $\tilde{Y}^{sv}$ by minimising the following error:

$$argmin_{\tilde{s}\tilde{v}} \parallel b^{\tilde{s}\tilde{v}} - G \times_1 \tilde{s} \times_2 \tilde{v} \parallel \tag{12}$$

where G is derived from learning (equation 8). Since tensor G represents the intrinsic body configuration 'content' of the considered action and manages interactions between all factors, an accurate solution for style and view can only be reach for the same action.

If the style vector, $\tilde{s}$ is known we can obtain a closed form solution for $\tilde{v}$ and vice versa. This leads to an iterative procedure for estimating $\tilde{s}$ and $\tilde{v}$ simultaneously until equation 12 converges [12]. In practice, we follow Lee's approach where $\tilde{s}$ is initialised with a mean style estimate. Since the view classes are discrete, we identify the closest view class and use it to estimate $\tilde{s}$. Finally, vector $\tilde{b}^{sv}$ is unstacked to create matrix $\tilde{B}^{sv}$; then the action $\tilde{Y}^{sv}$ is embedded into the low dimensional space using equation 10.

3 Experimental Results

3.1 Experimental Setup

The proposed framework was validated on the publicly available multi-view IXMAS dataset [27,25], which is considered as the benchmark for action recognition methods. Since the 'throw action' is not performed by all subjects, we excluded it from our experiments. As a result, the chosen dataset is comprised of 12 actions, performed 3 times by 12 different actors. Each of these 432 activity instances was recorded simultaneously by 5 calibrated cameras, and a reconstructed 3D visual hull is provided. In this dataset, actors' positions and orientations are arbitrary since no specific instruction was given during acquisition. As a consequence, the action viewpoints are arbitrary and unknown.

To obtain a dense set of action descriptors regarding viewpoints for training, we followed [21] approach where the animated visual hulls are projected onto 12 evenly spaced virtual cameras located around the vertical axis of the subject. In line with other experiments made on this dataset [16,15,28], the top view was discarded for testing.

Experiments are conducted using the leave-one-out strategy followed by [28,8,25,21]. In each run, we select one actor for testing and all remaining subjects for training. Two testing schemes were used: recognition using single view, and recognition using multiple views. In the recognition from multiple views, a simple majority voting rule was applied [16,15]. Finally, performances were compared to the other state of art methods. Unfortunately, results could not be compared with [21], because, instead of evaluating their method with original video data, they did it by using projections of the visual hulls.

3.2 Performances

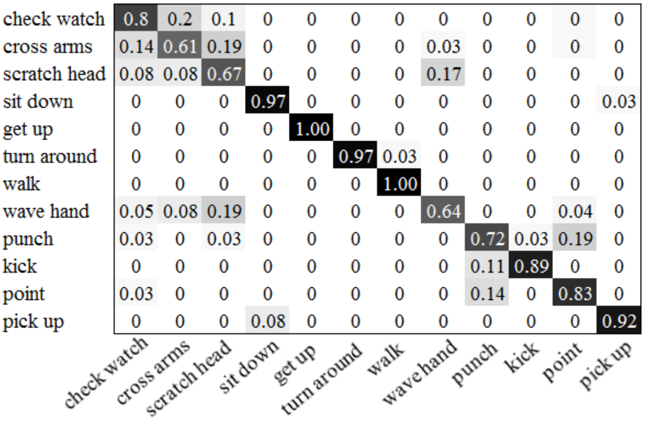
Although different approaches may use slightly different experimental settings, table 1 shows that our framework produces state of art performances. Accuracy rates obtained for an experiment aiming at only 11 actions, i.e. the 'point' action was not considered, reveals that we outperform all methods targeting this task [28,8,25] even if they considered a smaller set of subjects [8,25].

When all actions completed by all subjects are considered, i.e. 12, our framework displays results which are significantly better than Liu [15] and match those obtained by Liu [16]. Although performance alone cannot discriminate between Liu's and our method, we believe that our action models are superior. Indeed, unlike Liu's descriptors which are based on codebooks, ours consists of single integrated continuous models. Consequently, our action manifolds can be applied to many applications beyond action recognition such as synthetic action sequence generation, style recognition and camera view estimation.

Fig. 3 depicts the confusion matrix of recognition for the 'all-view' experiment. It reveals that our framework performed better when dealing with motions involving the whole body, i.e. "walk", "sit down", "get up", "turn around" and "pick up". Since temporal information is essential when dealing with highly dynamic motions and TLE aims at preserving temporal structure in each view, action manifolds of those activities are more representative. The best recognition rates 74.8%, 80.3% are obtained for camera 2 and 4 respectively. This was expected, since both views are the most similar among those used for training. Moreover, when dealing with either different, i.e. camera 1, or even significantly different views, i.e. camera 3, our framework still achieves reasonable recognition, i.e. 71.7% and 65.9% respectively. Details about average accuracy per camera can be found in supplementary material [13].

Table 1. Average recognition accuracy over all cameras (top view excluded) using either single or multiple views for testing

%	Subjects	Actions	Average Accuracy	
			Single view	All views
Weinland [25]	10	11	63.9	81.3
Yan [28]	12	11	64.0	78.0
Junejo [8]	10	11	74.1	-
Our	12	11	**75.0**	**83.1**
Liu [15]	12	13	71.7	78.5
Liu [16]	12	13	73.7	82.8
Our	12	12	**73.2**	**83.1**



	check watch	cross arms	scratch head	sit down	get up	turn around	walk	wave hand	punch	kick	point	pick up
check watch	0.8	0.2	0.1	0	0	0	0	0	0	0	0	0
cross arms	0.14	0.61	0.19	0	0	0	0	0.03	0	0	0	0
scratch head	0.08	0.08	0.67	0	0	0	0	0.17	0	0	0	0
sit down	0	0	0	0.97	0	0	0	0	0	0	0	0.03
get up	0	0	0	0	1.00	0	0	0	0	0	0	0
turn around	0	0	0	0	0	0.97	0.03	0	0	0	0	0
walk	0	0	0	0	0	0	1.00	0	0	0	0	0
wave hand	0.05	0.08	0.19	0	0	0	0	0.64	0	0	0.04	0
punch	0.03	0	0.03	0	0	0	0	0	0.72	0.03	0.19	0
kick	0	0	0	0	0	0	0	0	0.11	0.89	0	0
point	0.03	0	0	0	0	0	0	0	0.14	0	0.83	0
pick up	0	0	0	0.08	0	0	0	0	0	0	0	0.92

Fig. 3. Class-confusion matrix using multiple views. The average performance is 83.1%.

4 Conclusion

This paper introduces a novel human action recognition framework for arbitrary individuals and views. Its main contribution is a procedure for learning discriminative and unified action descriptors, which reside in a low dimensional space. These descriptors are constructed automatically by taking advantage of the TLE algorithm and a generative decomposable model. Performance of the proposed methodology has been evaluated using the IXMAS dataset and competitive results have been demonstrated. In addition, since our procedure to produce manifold based descriptor is general, it can be applied to many applications beyond action recognition such as visual surveillance or sport analysis. We plan to investigate some of these directions in future work.

Acknowledgments. The authors would like to thank Lena Gorelick from University of Western Ontario and Richard Souvenir from University of North Carolina at Charlotte for sharing their codes.

References

1. Ali, S., Basharat, A., Shah, M.: Chaotic invariants for human action recognition. In: ICCV, pp. 1–8 (2007)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. NIPS 14, 585–591 (2001)
3. Bobick, A., Davis, J.: The recognition of human movement using temporal templates. TPAMI 23(3), 257–267 (2001)
4. Cutler, R., Davis, L.: Robust real-time periodic motion detection, analysis, and applications. TPAMI 22(8), 781–796 (2000)
5. Elgammal, A., Lee, C.S.: Tracking people on a torus. TPAMI 31(3), 520–538 (2009)
6. Elgammal, A., Lee, C.S.: Separating style and content on a nonlinear manifold. In: CVPR, vol. 1 (2004)

7. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. TPAMI 29(12), 2247 (2007)
8. Junejo, I., Dexter, E., Laptev, I., Pérez, P.: Cross-view action recognition from temporal self-similarities. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 293–306. Springer, Heidelberg (2008)
9. Laptev, I.: On space-time interest points. IJCV 64(2), 107–123 (2005)
10. Lathauwer, L., Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. 21(4), 1253–1278 (2000)
11. Lawrence, N.: Gaussian process latent variable models for visualisation of high dimensional data. In: NIPS, vol. 16 (2004)
12. Lee, C., Elgammal, A.: Homeomorphic manifold analysis: Learning decomposable generative models for human motion analysis. In: Vidal, R., Heyden, A., Ma, Y. (eds.) WDV 2005/2006. LNCS, vol. 4358, pp. 100–114. Springer, Heidelberg (2007)
13. Lewandowski, M., Makris, D., Nebel, J.C.: Average recognition rates using single views. (2010); supplied as additional material, avgrecrates.tif
14. Lewandowski, M., Martinez-del-Rincon, J., Makris, D., Nebel, J.-C.: Temporal extension of laplacian eigenmaps for unsupervised dimensionality reduction of time series. In: Proc. ICPR (2010)
15. Liu, J., Ali, S., Shah, M.: Recognizing human actions using multiple features. In: CVPR (2008)
16. Liu, J., Shah, M.: Learning human actions via information maximization. In: CVPR (2008)
17. Myronenko, A., Song, X., Carreira-Perpinán, M.: Non-rigid point set registration: Coherent Point Drift. In: NIPS, vol. 19, p. 1009 (2007)
18. Ogale, A., Karapurkar, A., Aloimonos, Y.: View-invariant modeling and recognition of human actions using grammars. In: W. on Dyn. Vis. at ICCV, vol. 5 (2005)
19. Parameswaran, V., Chellappa, R.: View invariance for human action recognition. IJCV 66(1), 83–101 (2006)
20. Rabiner, L., Juang, B.H.: Fundamentals of speech recognition (1993)
21. Richard, S., Kyle, P.: Viewpoint Manifolds for Action Recognition. EURASIP J. on Img. and Vid. Proc. 2009 (2009)
22. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
23. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
24. Wang, C., Mahadevan, S.: Manifold alignment using Procrustes analysis. In: ICML, pp. 1120–1127 (2008)
25. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. In: ICCV, vol. 5(7), p. 8 (2007)
26. Weinland, D., Ronfard, R., Boyer, E.: Automatic discovery of action taxonomies from multiple views. In: CVPR, vol. 2, pp. 1639–1645 (2006)
27. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding 104(2-3), 249–257 (2006)
28. Yan, P., Khan, S., Shah, M.: Learning 4D action feature models for arbitrary view action recognition. In: CVPR, vol. 12 (2008)
29. Yilmaz, A., Shah, M.: Recognizing human actions in videos acquired by uncalibrated moving cameras. In: ICCV, vol. 1, pp. 150–157 (2005)

Figure-Ground Image Segmentation Helps Weakly-Supervised Learning of Objects

Katerina Fragkiadaki and Jianbo Shi

GRASP Laboratory, University of Pennsylvania
3330 Walnut St., Philadelphia, PA-19104
katef@seas.upenn.edu, jshi@cis.upenn.edu

Abstract. Given a collection of images containing a common object, we seek to learn a model for the object without the use of bounding boxes or segmentation masks. In linguistics, a single document provides no information about location of the topics it contains. On the contrary, an image has a lot to tell us about where foreground and background topics lie. Extensive literature on modelling bottom-up saliency and pop-out aims at predicting eye fixations and allocation of visual attention in a single image, prior to any recognition of content. Most salient image parts are likely to capture image foreground. We propose a novel probabilistic model, *shape and figure-ground aware model* (sFGmodel) that exploits bottom-up image saliency to compute an informative prior on segment topic assignments. Our model exploits both figure-ground organization in each image separately, as well as feature re-occurrence across the image collection. Since we use image dependent topic prior, during model learning we optimize a *conditional* likelihood of the image collection given the image bottom-up saliency information. Our discriminative framework can tolerate larger intraclass variability of objects with fewer training data. We iterate between bottom-up figure-ground image organization and model parameter learning by accumulating image statistics from the entire image collection. The model learned influences later image figure-ground labelling. We present results of our approach on diverse datasets showing great improvement over generative probabilistic models that do not exploit image saliency, indicating the suitability of our model for weakly-supervised visual organization.

1 Introduction

Given a collection of images containing a common object, we seek to learn a model for detection and segmentation of the object without additional supervision. The absence of figure-ground segmentation ahead of time makes this task challenging. However, learning of object models with minimum amount of supervision is necessary for scaling vision systems to large number of object categories.

Models for unsupervised learning rely on the figure consistency principle: *foreground features tend to re-occur and co-occur more consistently across images than background features.* This permits their separation from the background and incorporation into the model built for the common object. However, the task remains challenging mainly for the following reasons:

1. *Heavy clutter.* The more cluttered the images, the harder to dig out the common object.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 561–574, 2010.

2. *Persistent co-occurrence of foreground with its semantically related background.* Examples are car and road, giraffe and grass, swan and water. So, in practice, the backgrounds in the image collection are not random. Rather they are highly correlated with the common figure, making it difficult for a generative process to segment it from the background.
3. *Large intraclass variation* of many object categories due to articulation, deformation, change of view point. This violates the figure consistency principle.

We propose a novel approach that deals with the above challenges by coupling figure-ground image segmentation and learning of the common object. To our knowledge, this is the first work that exploits image saliency and figure-ground organization for weakly-supervised learning of objects.

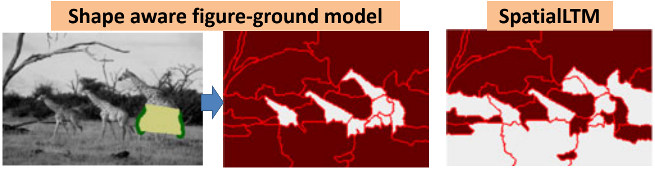


Fig. 1. The baseline model ([1]) does not discriminate between Giraffe and background due to persistent co-occurrence of Giraffe and ground in the image collection and wide variation of Giraffe shape. Wide intraclass variability is a common phenomenon in the visual domain. Our model exploits figure-ground information and effectively learns to segment the object.

We set our problem as topic discovery in the image collection: we aim at assigning image segments to visually coherent topics and learn the models for the common object (single foreground topic) and its background (possibly multiple background topics). We employ an iterative algorithm. Initially, we extract purely bottom-up figure-ground cues from each image, represented as *multiple soft figure-ground maps*. We score these maps using bottom-up image saliency. The map scores are not fixed, they change according to feature re-occurrence: figure-ground maps that propose foreground found most consistent across the image collection will iteratively get higher scores. At each iteration, we sample the highest scoring map in each image and obtain a prior on segment figure-ground labels. We perform a constrained probabilistic segment topic assignment by assigning different topics to segments that have different figure-ground labels. We accumulate image statistics and update the model parameters accordingly. Model update influences the scores of figure-ground maps and thus the figure-ground segment labels. Thus, figure-ground segmentation changes according to the model being built.

Our model has the following advantages:

– The object is naturally repulsed by its background and frequent co-occurrence of object and its semantically related background is no longer a problem. Optimizing segment topics *given* image saliency cues gives a discriminative flavor and offers robustness towards purely generative models.
– *Segment independence is not part of our assumptions.* Bottom-up saliency and figure-ground organization are operations that involve competition among segments

in each image and thus segment independence does not hold (see also fig. 2). This models the visual domain more accurately than most of the probabilistic models in previous work.
- We are not restricted to a fixed figure-ground segmentation. Our input is a set of soft figure-ground maps and is part of the learning process to choose the best one. The *loop* from feature re-occurrence back to bottom-up image saliency cues deals effectively with the presence of multiple foreground objects in each image.

The paper is organized as follows: We discuss related work in section 2. We present our model in section 3. In section 3.2 we present our representation for figure-ground image organization. Learning and inference in our model are presented in sections 4 and 5. Experimental results are in section 6. We conclude in section 7.

2 Related Work

There is extensive previous work on unsupervised or weakly-supervised learning of object categories:

Topic models. ([2], [3]) Topic models from statistical text analysis (LDA [4], p-LSA [5]) use unordered “bag of words“ representation of documents to automatically discover topics in large text corpora. In the visual domain, usually an image corresponds to a document and a local patch descriptor to a visual ’word’. Much of previous work is devoted in imposing spatial coherence between the visual words. Authors of [1] propose uniformity of topic assignments to the words belonging to the same superpixel. Work of [6] uses multiple segmentations of images and models each segment as a document. Segments well corresponding to topics are expected to have more peaked topic distributions than wrong (leaking) segments. In [7] a fixed outline of the object is used as extra input to guide learning.

Discriminative models. Part of previous work ([8], [9]) takes a discriminative approach having a negative collection of images (not containing the common object) as additional input for detection of the common object. Works of [10], [11] and [12] model weakly-supervised learning as multiple instance learning (MIL), using MILboosting for object or part detection. Recently, authors of [13] used discriminative clustering to assign figure-ground labels to image segments in the image collection such that figure and ground classes are best separated. However, their formulation, does not take into account image saliency of foreground.

In [14] a hierarchical model representation is built from a few training examples. Plausible feature groupings are discovered iteratively based on the principles of suspicious coincidence and competitive exclusion. Authors of [15] attempt to segment a pair of images containing a common object. The problem is formulated as an MRF with a global constraint about appearance histogram matching of the corresponding parts from the two images. Authors of [16] learn a generative model for segmentation of a collection of images combining appearence with object shape and pose.

Our model exploits an informative topic prior based on image figure-ground cues and maximizes a conditional likelihood of the image collection given that prior. In this way, it is more suitable for the problem of weakly-supervised learning than pure generative

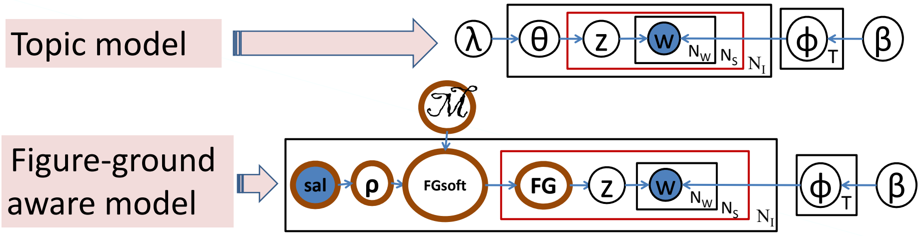


Fig. 2. Shading indicates observed variables and no shading indicates latent variables. N_i denotes number of images ($|\mathcal{I}|$), N_S number of segments and N_w number of words. *Top*: Topic model from [1]. θ^i is a multinomial distribution over topics for image $I^i$and λ is the parameter of a uniform Dirichlet prior over distributions $\theta^i, \ \ i = 1 \cdots |\mathcal{I}|$. *Bottom*: Shape and figure-ground aware model. The topic prior tightly depends on image figure-ground cues, as expressed by variable sal. Given the observed w, information flows from feature re-occurrence as expressed by ϕ back to the scores of figure-ground cues ρ, realizing the *feedback loop* from similarity across images to image figure-ground labelling. See text for the rest of notation.

models. We can tolerate larger amount of intraclass variability with smaller amount of training data. Shape is not provided as input but is recovered along the way. Discrimination is built within the image, by trying to discriminate the common object from its background. We anticipate that figure-ground information would be useful in learning most of the representations that appear in previous work, especially for objects with large intra-class variability.

3 Shape and Figure-Ground Aware Model

Adopting the terminology of topic models we claim that in images *topics are not created equal*. Extensive literature on bottom-up image saliency tells us that topics do not have uniform prior distribution given an image: Foreground topics tend to occupy salient image locations, while background topics less salient ones. Our model proposes a topic prior tightly depending on bottom-up image figure-ground cues.

Let $T = \{t_1, t_2, \cdots t_{|T|}\}$ be the topics in which to organize the image collection, t_1 denotes the single foreground topic and $t_2 \cdots t_{|T|}$ the background topics. Let s_k^i be the kth segment of image I^i and $\mathcal{S}^i$ be the set of segments of that image: $\mathcal{S}^i = \{s_1^i, s_2^i, \cdots s_{|\mathcal{S}^i|}^i\}$. Let z_k^i denote the topic of s_k^i. Let W be the word codebook and w_{kl}^i be the lth word of s_k^i (see also section 3.1). Let ϕ^z be a multinomial distribution over words given topic z and β be the parameter of a uniform Dirichlet prior over $\boldsymbol{\phi}^z, \ \ z = t_1 \cdots t_{|T|}$.

Let FG_k^i to be the figure-ground label of segment $s_k^i \ \in \ \mathcal{S}^i$ in image I^i:

$$\mathrm{FG}_k^i \in \ \{0, 1\}$$

$\mathrm{FG}_k^i \ = 1$ if the segment s_k^i belongs to the common object in $\mathcal{I}$. Note that each image may have multiple foreground objects. $\mathrm{FG} \ = 1$ refers to the presence of the common object (common figure) that is of interest to us. We abuse language and call it figure-ground label for brevity.

Let FGmap_j^i be the jth soft figure-ground map as found by bottom-up figure-ground image organization and let $\mathcal{R}^i$ be the set of these maps in image I^i (see section 3.2):

$$\mathrm{FGmap}_j^i : \ \mathcal{S}^i \longrightarrow [0,1], \quad \mathcal{R}^i = \{\mathrm{FGmap}_j^i, \ j = 1 \cdots |\mathcal{R}^i|\} \tag{1}$$

FGmap_{jk}^i represents the probability of segment s_k^i to be part of foreground *given* FGmap_j^i. According to our formulation, segments have different probabilities of foreground given different maps.

We define sal_j^i to be the saliency score of map FGmap_j^i as computed by image saliency scoring (see section 3.2).

$$\mathrm{sal}_j^i \in \ [0,1] \ \ , \qquad \textstyle\sum_{j=1}^{\mathcal{R}^i} \mathrm{sal}_j^i = 1$$

Let $\mathbf{sal^i}$ be the saliency values of maps in image I^i.

We define ρ_j^i to be the trust score of map FGmap_j^i:

$$\rho_j^i \in \ [0,1] \ \ , \qquad \textstyle\sum_{j=1}^{\mathcal{R}^i} \rho_j^i = 1$$

In contrast to the saliency score sal_j^i, the trust score of a map depends on both bottom-up saliency of each image in isolation, as well as feature re-occurrence across images. It realizes the *feedback loop* from feature re-occurrence in $\mathcal{I}$ back to image figure-ground segmentation. Intuitively, the trust score of a map is high if it maps the segments occupied by the common object to high foreground probabilities and the rest of the segments to low foreground probabilities. Let $\boldsymbol{\rho^i}$ be trust scores of maps in image I^i.Let FGsoft^i to be the map with the highest trust score in image I^i:

$$\mathrm{FGsoft}^i = \mathrm{FGmap}_\ell^i, \quad \text{where} \quad \ell = \arg\max_{j=1\cdots|\mathcal{R}^i|} \rho_j^i \tag{2}$$

Let $\mathcal{M}$ denote the shape of the common object represented by a mixture of K Gaussian distributions over vectors of real values representing shape descriptors attached to binary shape masks of the object. Let L be the dimension of our shape descriptor:

$$\mathcal{M} = \{\omega_l, \mu_l, v_l, \ 0 < \omega_l < 1, \ \textstyle\sum_{l=1}^K \omega_l = 1, \ \mu_l \in \mathbb{R}^L, v_l \in \mathbb{R}, \ l = 1 \cdots K\},$$

Naturally, the probability of a shape descriptor $\mathrm{sc} \in \mathbb{R}^L$ given shape model $\mathcal{M}$ is:

$$P(\mathrm{sc}|\mathcal{M}) = \sum_{l=1}^{K} \omega_l \cdot \exp(-\frac{||\mu_l - \mathrm{sc}||^2}{2v_l^2}) \tag{3}$$

Our input is a set of soft figure-ground maps along with a distribution of saliency scores over them. During learning we alter the initial score distribution taking into account feature re-occurrence across images. Intuitively, we assign topics to segments such that segments found to have different figure-ground labels by maps of high saliency value are mapped to different topics and segments belonging to the same topic are most similar.

Our model parameters are $\mathcal{M}$ and ϕ, w and sal are observed variables and FG, z, ρ and FGsoft are latent variables. Learning of our model amounts to optimizing the following conditional likelihood of the image collection:

$$\max_{\mathcal{M},\boldsymbol{\phi}} P(\mathbf{FG}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\rho}, \mathbf{FGsoft}|\mathbf{sal}, \beta) \tag{4}$$

$$= \max_{\mathcal{M},\boldsymbol{\phi}} \prod_{i=1}^{|\mathcal{I}|} P(\boldsymbol{\rho^i}|\mathbf{sal}^i, \mathcal{M}) \cdot P(\mathrm{FGsoft}^i|\boldsymbol{\rho^i}) \cdot \prod_{k=1}^{|S^i|} P(z_k^i|\mathrm{FG}_k^i) \cdot P(\mathrm{FG}_k^i|\mathrm{FGsoft}^i) \cdot \prod_{l=1}^{|W_k^i|} P(w_{kl}^i|z_k^i, \phi^{z_k^i})$$

We optimize a conditional likelihood of topic assignments given bottom-up saliency information of the images in $\mathcal{I}$. We call $\mathbf{sal^{\boldsymbol{i}}}, \ \ i = 1 \cdots |\mathcal{I}|$ a prior, since saliency values are computed from each image in isolation, without taking into account the image collection $\mathcal{I}$ and re-occurrence of features, that is without seeing all the data.

A byproduct of our model is the organization of backgrounds into visually coherent groups. The performance of our model in learning the common object is not sensitive to the total number of topics used, a single background topic would do. However, by increasing the number of topics, we additionaly get meaningful models for background clusters as in the topic model literature.

Our model exploits effectively the rich figure-ground information present in images to guide the topic discovery process in our weakly-supervised framework.

3.1 Image Representation

We use image segments as our basic units. Each image is described by a set of overlapping segments, obtained from multiscale segmentation. We used the multiscale normalized cut code [17] and discretized the eigenvectors using different number of segments. Within each image segment we find a number of interest points using the scale invariant saliency detector [18]. Each interest point is described by a SIFT descriptor[19]. We discretized the space of SIFT descriptors using unsupervised k-means clustering. Each segment is further described by a texture word and color word, each resulting from quantization of texton and color histograms using k-means. For ease of presentation we will refer to the description of each image segment by a bag of visual words, without discriminating among SIFT or color/texture words.

3.2 Figure-Ground Image Organization

Figure-ground labelling is a step of perceptual organization which assigns a contour to one of the two abutting regions. There is experimental evidence that rich figure-ground information is available in images much before any of their content is recognized ([20], [21]). We assume saliency and figure-ground organization are related, that is *most salient image parts tend to belong to foreground (common object) while less salient ones to background* ([22]).

We note two important properties of figure-ground organization and image saliency that violate image segment independence: 1) Competition among different image parts for visual attention allocation (illustrated in the literature through normalization of saliency scores across image locations). 2) Convexity, connectedness of foreground, center-surround figure-ground competition. We take the above into account and choose to represent bottom-up figure-ground information with a score distribution over *multiple* segment foreground probability maps (soft figure-ground maps) (see eq. 1). This suits well our unsupervised learning framework: each one of these maps proposes image figure-ground labelling and learning choses the correct one by altering their score distribution.

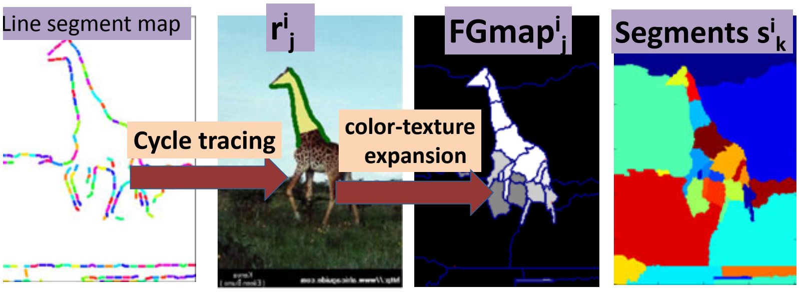

Line segment map
rⁱ_j
FGmapⁱ_j
Segments sⁱ_k
Cycle tracing
color-texture expansion

Fig. 3. *Ribbon extraction using cycle tracing*. a) Fitting straight line segments to a contour map. b) Extracting ribbons using cycle tracing, by piecing pairs of line segments in a graph partitioning framework. Yellow indicates ribbon interior. c)Figure-ground map (FGmap) obtained from color and texture expansion of the ribbon. White indicates high probability of foreground and black low probability. d) Segmentation map. Ribbons prevent over-fragmentation and achieve *scale invariance*. On the contrary, in segmentation we get very different image groupings for different numbers of segments. Here for illustration purposes we show segmentation of the finest scale (superpixels) although we used multiple segmentation scales.

Multiple Soft Figure-Ground Maps

Figure-ground organization is a mid-level process and mid-level grouping is required to provide information about figure-ground labelling. Our approach involves the following steps:

- We piece together over-fragmented segmentation boundaries to recover large (possible overlapping) foreground image structures. This can be done using multiple segmentations or greedy segment extension based on continuity of segment boundaries. We call these structures *ribbons* to distinguish them from segments and indicate that they can be obtained from different (not necessarily segment based) computational procedures. Later we present a *globally optimal* way for piecing segment boundaries for ribbon extraction using contour continuity.
- For each ribbon a segment foreground probability map is calculated: The interior of the ribbon is sent to foreground and surrounding highly contrasting segments to background. This is extended to a full segment foreground probability map by classifying each of the remaining image segments as foreground or background using color and texture features. Let r^i_j denote the jth ribbon of image I^i and $|\mathcal{R}^i|$ the number of ribbons in image I^i. The corresponding foreground probability map FGmap^i_j(see eq. 1) represents the probability of each segment s^i_k to be part of foreground *given* ribbon r^i_j. For each map $\mathrm{FGmap}^i_j, \;\; j = 1 \cdots |\mathcal{R}^i|$ we define the following sets of segments:

$$S^i_{in_j} = \{s \in \mathcal{S}^i \;\; s.t \;\; \mathrm{FGmap}^i_j(s) > l_1\}, \;\; S^i_{out_j} = \{s \in \mathcal{S}^i \;\; s.t \;\; \mathrm{FGmap}^i_j(s) < l_2\}$$

$$S^i_{dont\text{-}know_j} = \{s \in \mathcal{S}^i \;\; s.t. \;\; s \notin S^i_{in_j}, s \notin S^i_{out_j}\}$$

where $\mathcal{S}^i$ is the set of segments of image I^i and $l_2 < 0.5 < l_1$ (we chose $l_1 = 0.6$ and $l_2 = 0.4$). So, naturally, each FGmap^i_j constraints the figure-ground labelling of the segments of image I^i, sending $S^i_{in_j}$ to the foreground and $S^i_{out_j}$ to the background.

We define a shape mask $mask^i_j$: $mask^i_j(p) = \begin{cases} 1 & \text{if } \exists\, s \in S_{in^i_j} \text{ covering } p \\ 0 & \text{otherwise} \end{cases}$

describing the foreground FGmap^i_j selects.

To each mask^i_j we attach a grid shape feature sc^i_j of dimensions 6×6 and with 6 angular bins in each spatial cell to describe its shape.

- Maps are scored using saliency cues (see eq. 5) and scores are normalized to create a dictribution.We used $100 - 150$ figure-ground maps (FGmap) per image.

Cycle tracing for ribbon extraction. We present here a novel approach for ribbon extraction which we used along with the multiple segmentation approach: We piece together over-fragmented segmentation boundaries in a *globally optimal way* based on good continuity of the boundary contour, generalizing the tool for cycle tracing for contour extraction of [23]. More precisely, we threshold the output of Probability of boundary detector [24] and fit line segments in a greedy way. We build a graph $\mathbf{W}$ whose nodes correspond to *pairs of roughly parallel line segments* and edge weights e_{ij} reflect the bending energy of the side contours from pair i into pair j. We have high affinity between two pairs of line segments when the one naturally extends into the other. We discretized the complex eigenvectors of the Laplacian of $\mathbf{W}$ corresponding to complex eigenvalues with large norm. For discretization we used the shortest path algorithm to recover the cycle enclosing the largest area in the embedding space. For further details refer to [23]. Ribbons obtained this way provide scale (distance between the two parallel contours) and orientation (orientation of the symmetry axis) helping alignment and recognition of shape.

Image Saliency Scoring

Saliency is the property of some parts of the image popping out and being well separated from their surrounding. Image saliency has been extensively studied in the literature ([25], [26],[27],[28]) and is related to properties such as local contrast, global exception in the image, centrality of location.

We score our figure-ground maps using image saliency cues. In each image I^i we define the saliency value sal^i_j of each FGmap^i_j:

$$\mathrm{sal}^i_j = \frac{1}{Z} \cdot \mathrm{FGcontrast}(\mathrm{FGmap}^i_j) \cdot \mathrm{Uniqueness}(\mathrm{FGmap}^i_j) \tag{5}$$

- $\mathrm{FGContrast}(\mathrm{FGmap}^i_j)$ measures feature dissimilarity between the figure and ground that FGmap^i_j defines: $\mathrm{FGcontrast}(\mathrm{FGmap}^i) = \frac{1}{Z}\mathrm{D_{KL}}\{\mathrm{f}(p, p \in S^i_{in_j})||\mathrm{f}(p, p \in S^i_{out_j})\}$

  where $\mathrm{D_{KL}}$ denotes KL-divergence and f(pixel-set) denotes feature distribution with support in pixel-set. We used textons and quantized RGB intensity values as our features.
- $\mathrm{Uniqueness}(\mathrm{FGmap}^i_j)$ measures dissimilarity between features of the figure of FGmap^i_j and the rest of the image I^i: $\mathrm{Uniqueness}(\mathrm{FGmap}^i_j) = \frac{1}{Z} \cdot \frac{\sum_{p\ \ S^i_{in_j}} \sum_{l\ \ I^i} d_{pl}\ \mathrm{D_{KL}}(\mathrm{f}^{s(p)}_p, \mathrm{f}^i_l)}{|p\ \ S^i_{in_j}|}$

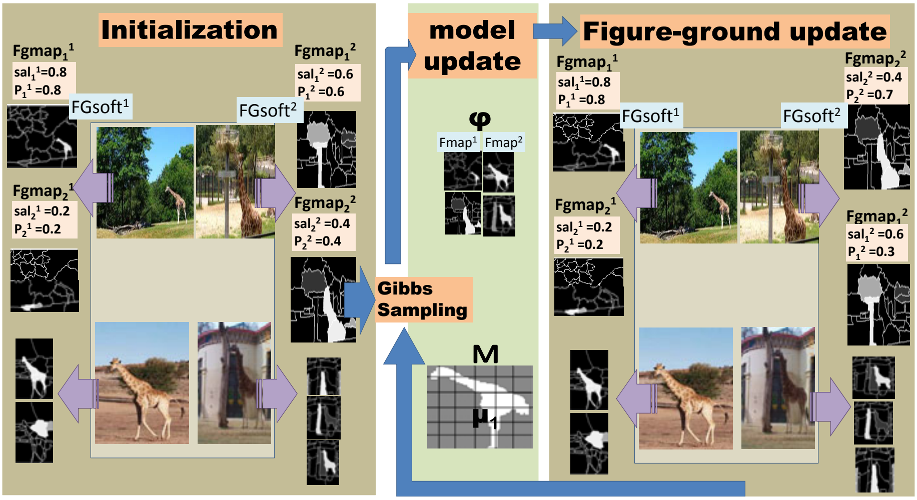


Fig. 4. *Learning a shape and figure-ground aware model.* White indicates high probability of foreground and black low probability. For each image we show the corresponding figure-ground maps ordered by their ρ scores. Notice the changing of ρ scores of the maps of the left pair of images. The presence of multiple foreground objects is not a problem in our model. A framework with fixed saliency scores would not be flexible enough to deal with multiple foreground objects present in images. For illustration purposes we use $K = 1$ for the shape model $\mathcal{M}$.

where $s(p)$ is the superpixel containing pixel p and f_l^i is the feature distribution of superpixel containing pixel l. We take into account the distances d_{pl} of pixels: high similarity found in large distances is worse that high similarity in small distances since it indicates concavity, a property of background.

In summary, in each image I^i our figure-ground representation is a set of segment foreground probability maps $\text{FGmap}_j^i, \quad j = 1 \cdots |\mathcal{R}^i|, \quad i = 1 \cdots |\mathcal{I}|$, with a distribution $\mathbf{sal}^{\boldsymbol{i}}$ of saliency scores over them.

4 Learning

We use a type of EM procedure to estimate the parameters of our model. We use Gibbs sampling to get the expected conditional likelihood of latent given the observed variables at E step. $iter_{out}$ denotes the iteration counter for the EM algorithm and $iter_{in}$ the iteration counter for Gibbs sampling. We initialize the model parameters $\mathcal{M}$ and ϕ to the uniform distributions over the corresponding domains, that is: $\phi_w^{t_i} = \frac{1}{|W|}, \quad w \in W, \quad i = 1 \cdots |T|$ and $v_l = \infty, \quad l = 1 \cdots K$.

E step: From model parameters to figure-ground constraints

Sampling of figure-ground trust scores ρ
`Initial iteration` $(iter_{out} = 1)$: Initially, since ϕ and $\mathcal{M}$ are non informative (uniform) we have: $\rho_{\mathbf{j}}^{\mathbf{i}} = \text{sal}_j^i, \quad j = 1 \cdots \mathcal{R}^i, \quad i = 1 \cdots |\mathcal{I}|, \quad iter_{out} = 1.$

$\texttt{Later iterations}$ $(iter_{out} > 1)$: Given the bag of words representation (ϕ^z, $z = t_1 \cdots t_{|T|}$) we compute for each image I^i a pixel foreground probability map Fmap^i. We assign to each image pixel p the mean foreground probability of the segments containing it:

$$\mathrm{Fmap}(p)^i = \frac{\sum_{k=1}^{|S_p|} P(z_k^i = t_1|\phi)}{|S_p|} = \frac{\sum_{k=1}^{|S_p|} \prod_{l=1}^{|W_k^i|} \phi_{w_{kl}^i}^{t_1}}{|S_p|} \tag{6}$$

where $\phi_{w_{kl}^i}^{t_1}$ is the probability of word w_{kl}^i given topic t_1, p is a pixel of image I^i and S^p is the set of segments containing it. We update the scores ρ_j^i of all the figure-ground maps FGmap_j^i in the image collection:

$$\rho_{\mathbf{j}}^{\mathbf{i}_{new}} = \frac{1}{Z} \cdot \mathrm{sal}_j^i \cdot \frac{1_{\mathrm{Fmap}^i} \cap 1_{\mathrm{mask}_j^i}}{1_{\mathrm{Fmap}^i} \cup 1_{\mathrm{mask}_j^i}} \cdot P(\mathrm{sc}_j^i|\mathcal{M}) \tag{7}$$

$j = 1 \cdots \mathcal{R}^i, \quad i = 1 \cdots |\mathcal{I}|, \quad iter_{out} > 1$
where $1_{\mathrm{Fmap}^i} = \{p, \ \mathrm{Fmap}^i(p) > \frac{1}{2}\}$, $1_{\mathrm{mask}_j^i} = \{p, \ \mathrm{mask}_j^i(p) = 1\}$ and $P(\mathrm{sc}_j^i|\mathcal{M})$ is given by equation 3.

Intuitively, figure-ground maps with high bottom-up saliency values that propose foreground agreeing with $\mathcal{M}$ and the corresponding Fmap get higher trust scores.

Determining FGsoft*:* For each image I^i we keep the highest scoring figure-ground map applying a *winner take all* strategy. Different maps may be competing with each other so averaging (marginalizing) them would not be meaningful. See equation 2.

Sampling segment figure-ground labels FG*:* $P(FG_k^i = 1|\mathrm{FGsoft}^i) = \mathrm{FGsoft}_k^i, \quad k = 1 \cdots |\mathcal{S}^i|, \ \ i = 1 \cdots |\mathcal{I}|$

Sampling segment topics z: Denote by W the word vocabulary, by $W(s_k^i)$ the words of segment s_k^i, by $n_{t_l}^w$ the number of assignments of word w to topic t_l, by n_{t_l} the total number of word assignments to topic t_l and by $n_{-s_k^i}$ the count of word assignments excluding words belonging to the segment s_k^i.

We have: $z_{s_k^i} = \begin{cases} t_1 & \text{if } \mathrm{FG}_k^i = 1 \\ \sim P^{'}(z|\mathbf{z}_{-\mathbf{s_k^i}}, \mathbf{w}) & \text{if } \mathrm{FG}_k^i = 0 \end{cases}$

with :

$$P(z_{s_k^i} = t_l|\mathbf{z}_{-\mathbf{s_k^i}}, \mathbf{w}) \propto \prod_{w \in W(s_k^i)} \left(\frac{n_{-s_k^i, t_l}^w + \beta}{n_{-s_k^i, t_l}^{(\cdot)} + |W| \cdot \beta}\right) \tag{8}$$

where $\sim$ denotes sample from distribution and $P^{'}(z|\mathbf{z}_{-\mathbf{i}}, \mathbf{w})$ is the distribution over background topics: we exclude topic t_1 from T, compute $P(z|\mathbf{z}_{-\mathbf{i}}, \mathbf{w})$ for $z = t_2 \cdots |T|$ using equation 8 and normalize. We perform 500 iterations of figure-ground segment labels updates and segment topic assignments over all segments of $\mathcal{I}$ in random order.

M step : From figure-ground constraints to model parameters

Updating multinomial distributions of words given topics $\phi^z, \ \ z = t_1 \cdots t_{|T|}$*:* We update $\phi^z, \ \ z = t_1 \cdots t_{|T|}$ be counting word assignments to topics during all the iterations of Gibbs sampling of the previous E step.

Updating shape model $\mathcal{M}$*:* Let $\mathbf{FGsoft}$ be the set of highest scoring figure-ground maps during the previous E step: $\mathbf{FG_{soft}} = \{\text{FGsoft}^i, \ \ i = 1 \cdots |\mathcal{I}|\}$. We compute all pair shape affinities between the corresponding shape features $\text{sc}_k, \ k = 1 \cdots |\mathcal{I}|$, obtaining affinity matrix $\mathbf{A}$: $\mathbf{A}_{kl} = \exp(-\frac{||\text{sc}_k - \text{sc}_l||^2}{2d^2}), \ \ k, \ l \ = 1 \cdots |\mathcal{I}|$ Since we do not expect all shape masks to be correct, we aim at extracting compact clusters in this shape feature set. We zero out pairwise affinities with values below a threshold as indicating disagreement in shape. In the remaining shape affinity set, we order our features based on the number of neighbors. Large number of neighbors indicates high probability of exhibiting the common shape. Let $sc_k^{\text{best}}, \ \ k \cdots K$ denote the K shape context features with the highest number of neighbors and n_k^{best} denote the corresponding number of neighbors. Then:

$$\mathcal{M} = \{\mu_l = \text{sc}_l^{\text{best}}, v_l = d, \ \ \omega_l = \tfrac{1}{Z} \cdot n_l^{\text{best}}, \ \ l = 1 \cdots K\} \ \ , \quad Z = \textstyle\sum_{l=1}^{K} n_l^{\text{best}}$$

That is, the weights and centers of the mixtures are updated, while the variances are kept fixed and equal to constant d (same for all datasets used).

5 Inference

We used two different kinds of inference to score the performance of our model at the end of training and at test time:

- Inference using shape and figure-ground aware model (**sFGmodel**). We compute the segmentation labelling for image I^i: $\text{label}^i(p) = \frac{(\text{Fmap}^i(p) + \text{FGmap}^i_{\text{best}}(p))}{2}, \ \ p \in I^i$ where: $\text{best} = \arg\max_{j=1 \cdots |\mathcal{R}^i|} \ \text{sal}^i_j \cdot P(\text{sc}^i_j | \mathcal{M})$, $\text{FGmap}^i_{\text{best}}(p) = \frac{\sum_{k=1}^{|\mathcal{S}_p|} \text{FGmap}^i_{\text{best}k}}{|\mathcal{S}_p|}$ where $\mathcal{S}_p$ the set of segments containing pixel p. We threshold label^i to get binary pixel labels.
- Inference using only the bag of words representation learnt from the shape and figure-ground aware model(**bagFGmodel**). We compute the segmentation labelling for image I^i: $\text{label}^i(p) = \text{Fmap}^i(p), \ \ p \in I^i$. We threshold label^i to get binary pixel labels. In **bagFGmodel** figure-ground and shape information are used for learning but only the bag of words representation $\phi^z, \ \ z = t_1 \cdots t_{|T|}$ is used to infer image labelling when scoring performance of our model.

6 Experiments

We use various datasets with different levels of difficulty to test our algorithm: Caltech 101:1) 81 images of Airplanes; MSRC: 2) 70 images of Cars, 3) 84 images of Cows; ETH: 4) 48 images of Bottles, 5) 29 images of Swans, 6) 85 images of Giraffes; WeizmannHorses:7) 80 images In the cases where the whole dataset is not included, images were picked at random.

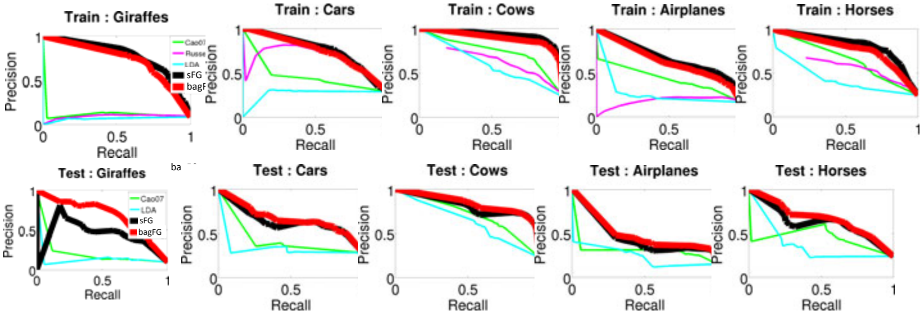


Fig. 5. *Precision-Recall curves for training and testing for 5 out of the 7 datasets.* Our models, sFG and bagFG, outperform all baseline methods.

In each dataset we randomly picked $2/3$ of images for training and $1/3$ for testing. In the datasets where ground truth segmentation is not provided we labeled it by hand by marking superpixels. We score the performance of our model using pixel precision and recall. We do not use segmentation accuracy since many times the object of interest captures a small part in the image and thus an algorithm with very low precision and high recall can get very high scores for segmentation accuracy by getting the background correctly.

We compare against 3 baseline models: 1) Standard LDA model. We used code provide in Topic Modelling Toolbox([29]). 2) Cao et al 07 ([1]) (SpatialLTM model) . In SpatialLTM words belonging to the same superpixel are assigned to the same topic. (see also sections 2). 3) Russel et al 06 ([6]). Each segment is treated as a document and segment based uniformity of words is exploited (see also section 2). We use the code provided online.

For each baseline method, we use the same features and word vocabularies as our model for a fair comparison. Since our baseline models do not discriminate between the foreground and background topics, for each topic we compute the average precision and choose the one with the highest value as the foreground topic. That is we compare against the best scoring topic found by each baseline model. For LDA and Cao et al 07 it is obvious how to get a pixel probability map from the multinomial distributions learned (see also eq. 6), which we threshold to compute our PR curves. For Russel et al 06 we sum the KL divergence scores of all segments to get a pixel score map which we threshold to obtain similar curves. The model Russel et al 07 ([6]) aims at organizing the segments of the training dataset into topics, and does not have a test component, so this method is not used at test time.

We tested both versions of our model: $\mathbf{sFGmodel}$ and $\mathbf{bagFGmodel}$. By using the $\mathbf{bagFGmodel}$ we show how figure-ground information can improve learning of even a simple representation. We believe it provides a fairer comparison with our baselines since same model representation is used to segment a new image.

The results show that using figure-ground information substantially improves the performance of even a simple bag of words represenation. We notice that in some object categories such as Giraffes or Airplanes, the best topic chosen by baseline methods,

Training	Giraffes	Cars	Cows	Airplanes	Horses	Bottles	Swans
Cao et al 07	0.124	0.460	0.738	0.436	0.651	0.274	**0.488**
Russell et al 06	0.100	0.672	0.479	0.181	0.404	0.323	0.287
LDA	0.157	0.358	0.595	0.268	0.428	0.297	0.420
sFG	**0.774**	**0.757**	**0.925**	**0.668**	**0.809**	**0.692**	0.487
bagFG	0.729	0.744	0.893	0.632	0.764	0.617	0.456
Testing							
Cao et al 07	0.208	0.423	0.706	0.315	0.448	0.244	0.593
LDA	0.144	0.331	0.627	0.241	0.368	0.229	0.538
sFG	**0.524**	0.638	0.812	0.492	0.624	0.236	0.693
bagFG	0.508	**0.702**	**0.879**	**0.544**	**0.710**	**0.239**	**0.706**

Fig. 6. *Average Precision at train and test time.* The results show that image figure-ground information is useful during training to learn the model, but at test time the representation learned is enough, using saliency in the new image does not offer more in most of the cases.

did not find similar shape across images. In these categories, the bag of feature representations is not strong enough to lead to clustering of the foreground features. In easier datasets like Cows and Horses, we see the baseline topic models to have reasonable performance. The shape and figure-ground aware model outperforms the baseline methods in all datasets.

7 Conclusion

We presented a shape and figure-ground aware model for weakly-supervised detection and segmentation of objects and their backgrounds. We show that by exploiting figure-ground information in images, we learn to segment the foreground object in challenging datasets. Our model uses a prior depending on image figure-ground cues and optimizes a discriminative cost function, which suits well our task of weakly-supervised image segmentation. We use a flexible representation of figure-ground, where figure-ground cues are influenced by feature re-occurrence in the image collection. Our model can tolerate multiple foreground objects in images and still be guided to the correct common figure, it does not make unnatural assumptions and is suitable for a wide variety of datasets. We will submit code for learning and inference for our model.

References

1. sCao, L., Fei-Fei, L.: Spatially coherent latent topic model for concurrent object segmentation and classification. In: ICCV (2007)
2. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their localization in images. In: ICCV, pp. 370–377 (2005)
3. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: ICCV, Washington, DC, USA (2005)
4. Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 2003 (2003)
5. Hofmann, T.: Probabilistic latent semantic analysis. In: Proc. of Uncertainty in Artificial Intelligence UAI 1999, pp. 289–296 (1999)

6. Russell, B.C., Efros, A.A., Sivic, J., Freeman, W.T., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
7. Gupta, A., Shi, J., Davis, L.: A 'shape aware' model for semi-supervised learning of objects and its context. In: NIPS (2008)
8. Nguyen, M., Torresani, L., de la Torre, F., Rother, C.: Weakly supervised discriminative localization and classification: A joint learning process
9. An Exemplar Model for Learning Object Classes. In: CVPR 2007 (2007)
10. 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)
11. Viola, P., Platt, J., Zhang, C.: Multiple instance boosting for object detection. In: NIPS (2006)
12. Dollár, P., Babenko, B., Belongie, S., Perona, P., Tu, Z.: Multiple component learning for object detection. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 211–224. Springer, Heidelberg (2008)
13. Joulin, A., Bach, F., Ponce, J.: Discriminative clustering for image co-segmentation. In: CVPR 2010 (2010)
14. Zhu, L.L., Lin, C., Huang, H., Chen, Y., Yuille, A.L.: Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 759–773. Springer, Heidelberg (2008)
15. Rother, C., Minka, T., Blake, A., Kolmogorov, V.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In: CVPR 2006 (2006)
16. Winn, J., Jojic, N.: Locus: Learning object classes with unsupervised segmentation. In: ICCV 2005 (2005)
17. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: CVPR 2005, Washington, DC, USA, pp. 1124–1131. IEEE Computer Society, Los Alamitos (2005)
18. Kadir, T., Brady, M.: Saliency, scale and image description. International Journal of Computer Vision V45, 83–105 (2001)
19. Lowe, D.: Distinctive image features from scale-invariant key-points. Intl. Journal of Computer Vision 60, 91–110 (2004)
20. Ren, X., Fowlkes, C.C., Malik, J.: Figure/ground assignment in natural images. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part II. LNCS, vol. 3952, pp. 614–627. Springer, Heidelberg (2006)
21. Hoiem, D., Stein, A.N., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from a single image. In: ICCV, pp. 1–8 (2007)
22. Goferman, S., Tal, A., Zelnik-Manor, L.: Puzzle-like collage. In: Computer Graphics Forum, EUROGRAPHICS (2010)
23. Zhu, Q., Song, G., Shi, J.: Untangling cycles for contour grouping. In: ICCV 2007 (2007)
24. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. PAMI 26, 530–549 (2004)
25. Itti, L.: A saliency-based search mechanism for overt and covert shifts of visual attention. Vision Research 40, 1489–1506 (2000)
26. Avraham, T., Lindenbaum, M.: Esaliency (extended saliency): Meaningful attention using stochastic image modeling. In: PAMI (2007)
27. Zhang, L., Tong, M.H., Marks, T.K., Shan, H., Cottrell, G.W.: SUN: A Bayesian framework for saliency using natural statistics. J. Vis. 8, 1–20 (2008)
28. Seo, H.J., Milanfar, P.: Static and space-time visual saliency detection by self-resemblance. J. Vis. 9, 1–27 (2009)
29. Griffiths, T.L., Steyvers, M., Tenenbaum: Finding scientific topics. In: National Academy of sciences. IEEE Computer Society, Los Alamitos (2007)

Enhancing Interactive Image Segmentation with Automatic Label Set Augmentation

Lei Ding and Alper Yilmaz

Photogrammetric Computer Vision Lab
The Ohio State University
dinglei@cse.ohio-state.edu, yilmaz.15@osu.edu

Abstract. We address the problem of having insufficient labels in an interactive image segmentation framework, for which most current methods would fail without further user interaction. To minimize user interaction, we use the appearance and boundary information synergistically. Specifically, we perform distribution propagation on the image graph constructed with color features to derive an initial estimate of the segment labels. Following that, we include automatically estimated segment distributions at "critical pixels" with uncertain labels to improve the segmentation performance. Such estimation is realized by incorporating boundary information using a non-parametric Dirichlet process for modeling diffusion signatures derived from the salient boundaries. Our main contribution is fusion of image appearance with probabilistic modeling of boundary information to segment the whole-object with a limited number of labeled pixels. Our proposed framework is extensively tested on a standard dataset, and is shown to achieve promising results both quantitatively and qualitatively.

1 Introduction

Image segmentation can be defined as the process of partitioning an image into regions corresponding to potential objects and their backgrounds. Over the course of years, image segmentation techniques without human interaction have not produced satisfactory results. In fact, fully automated segmentation is known to be an ill-posed problem due to the fact that there is (1) no clear definition of a correct segmentation; (2) no agreed-upon objective measure that defines the goodness of a segment, albeit the quality of a segment can be assessed [23] and that of a segmentation can be learned to some extent [21]. In order to do a semantically meaningful segmentation, it is essential to take *a priori* image information into account. This issue has been addressed in the literature as interactive image segmentation, which has been successfully applied in numerous articles [15,19,6,13,12,16,22]. A popular approach for user interaction is through a set of strokes or a trimap [24,5] providing known labels at certain pixels that are called seeds, from which segment labels at other pixels are to be predicted.

Although interactive image segmentation has drawn much attention, little has been done to study the problem of insufficient labels. For instance in Figure 1, in the case there is no label for the duck's beak, a typical interactive segmentation method would fail to segment it as part of the duck. A partial solution to this problem is *active label set augmentation*, which instantiates active learning [25], a framework that allows

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 575–588, 2010.

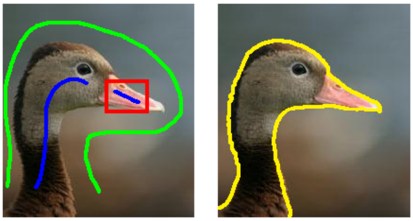


Fig. 1. When labeled pixels are insufficient, interactive image segmentation can perform badly. Our approach automatically introduces labels at critical pixels. Left: labeled object and background pixels are shown as strokes, and the automatically added labels are inside the rectangle. Right: the resulting object contour.

the learner to ask for informative labeled examples at certain costs. However, the luxury of additional labeled information is not always available. Therefore in this paper, we propose *automatic label set augmentation* to address the deficiencies in labeling. In particular, some pixels which we refer to as *critical pixels* have uncertain segment labels. The proposed scheme introduces labels at these critical pixels automatically to help make better segmentation decisions.

Specifically, our work utilizes information available at salient boundaries [18]. In other words, our method does not only merge pixels similar in appearance to the seed pixels in order to form segments; it also introduces unlabeled pixels as seeds in an automatic fashion based on nonlinear modeling of the compatibility between salient boundaries and target segments which leverages a non-parametric classification technique [26]. Our implicit assumption is that pixels within the same object, although may have distinct appearance features (e.g. colors), share similar spatial relations to salient boundaries. Intuitively, such addition of labeled information is vital to whole-object segmentation whose goal is to cut semantic objects from images with little user interaction. While the proposed method does not conclude research on whole-object segmentation, due to lack of high-level semantic knowledge, it provides a reasonable guess for previously unknown segment labels at critical pixels. Finally, the *distribution propagation* framework [8,28], which we adopt to integrate appearance and boundary cues, can be seen as an alternative to previously used graph based label propagation [31,13,12]. Because of the uncertain nature of added segment labels, the way of encoding the label or its strength as a real number at each vertex becomes inadequate, whereas the distribution propagation framework serves as a natural mechanism to integrate additional segment labels probabilistically.

Related Work. Our labeling method is most relevant to the segmentation framework introduced in [13] and recently used in [12] as label propagation, where the segment labels are encoded as a real-valued function at vertices. Label propagation for segmentation is commonly considered in a transductive setting, where known segment labels at some pixels are provided and labels at other pixels are to be estimated by minimizing a certain cost function. Besides, choosing a threshold for determining the segment labels is a difficult problem [31]. Generally, researchers adopt *ad hoc* techniques, e.g. class mass normalization [31] or adaptive window selection [14], which do not necessarily generalize well. In contrast, rigorous comparison of probabilities is made possible by using the distribution representation.

Recent years have seen much progress on estimating or utilizing salient boundaries [18,17,2,3], which are related to the main contribution of this paper. In two such papers [2,3], the authors study image segmentation from boundaries. However, their goal is either to over-segment images [3], or to extract semantically meaningful objects with sufficient labels [2]. Our work extends these methods by making the object/background inference based on the boundary information through a non-parametric learning framework and missing labels are directly handled. Among the few works addressing whole-object segmentation, an approach based on image matting is presented in [27]. In principle, our diffusion process for generating boundary related features is similar to matting. However, we only use boundary information in constructing feature vectors when appearance fails to provide sufficient evidence for object/background discrimination, whereas in [27], the authors rely on image appearance for generating mattes.

There are several papers on using priors for describing general shape or configuration of an object [29,16,9], which segment out the object in a graph-cut setting. However, the priors used limit their applicability to certain class of images to which the priors are suited. Our work can be considered complementary to theirs in terms of the underlying approach and the overall effects of segmentation, which involve automatically added labels. To better present our work, we briefly introduce the six major steps with more details to follow in their corresponding sections:

1. Distribution propagation using color features (Sec. 2).
2. Let the labeled set be $\mathcal{L}$ and the unlabeled set be $\mathcal{U}$. Identify a subset $\mathcal{A} \subset \mathcal{U}$ where the estimated distribution is ambiguous, i.e., the difference between the probabilities of being object and background is small. Also identify $\mathcal{U}^* \subset \mathcal{U} - \mathcal{A}$, which represents the set of pixels where the estimated distribution is informative, i.e., the probability difference is significant (Sec. 3);
3. Generate boundary probability map I_b indicating the probability of each pixel on a boundary (Sec. 3.1);
4. Compute diffusion signature vectors, which are the feature vectors at each pixel, from I_b (Sec. 3.1);
5. Modeling and classification using Dirichlet process multinomial logistic model, for which the training and test sets consist of feature vectors corresponding to $\mathcal{L} \cup \mathcal{U}^*$ and $\mathcal{A}$ respectively (Sec. 3.2);
6. Automatically generate segment distributions on $\mathcal{A}$. Let the labeled set be $\mathcal{L} \cup \mathcal{A}$, and the unlabeled set be $\mathcal{U} - \mathcal{A}$. Then proceed with the distributions computed by step (1) to derive new segment labels on $\mathcal{U}$, after which the algorithm stops.

2 Distribution Propagation for Labeling

In a graph representing an image, we have a set of n vertices, each of which corresponds to either a labeled or an unlabeled pixel, and a set of edges connecting neighboring pixels. Edges are denoted as $e_1, e_2, \cdots, e_m$, each of which is a binary set of

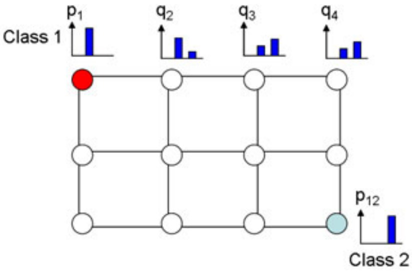


Fig. 2. Illustration of distribution propagation on an image graph. Lines connecting pixels are edges. Distributions, including known p_i's and estimated q_i's, are shown as histograms.

vertices. For representing the appearance, we consider the LUV color space due to its perceptual uniformity [11]. The LUV values of a pixel result in a feature vector $\mathcal{F}_S$ for each pixel S: $\mathcal{F}_S = (L_S, U_S, V_S)$. We define the weight of an edge as: $w(e) = \exp\left(-\sum_{i=1}^{3} \frac{(\mathcal{F}^i(v_1)-\mathcal{F}^i(v_2))^2}{2\delta_i^2}\right)$, where $v_1, v_2 \in e$, and $\mathcal{F}^i(v_j)$ is the i^{th} feature value at the pixel represented by vertex v_j. We pose image segmentation as propagating distributions at labeled vertices to unlabeled ones. Specifically, let us consider a segmentation setting where the set $\mathcal{Y}$ represents l label types which correspond to the objects and their backgrounds, $\mathcal{Y} = \{1, 2, \cdots, l\}$. A labeled vertex $i \in \mathcal{L}$ of class k is assigned a known distribution $p_i(y)$, such that $p_i(k) = 1$ and $p_i(y) = 0$ for $y \neq k$. The estimated distribution at an unlabeled vertex $i \in \mathcal{U}$ is defined as a multinomial distribution $q_i(y)$, which can be described by an l-dimensional vector, $[q_i(y=1), q_i(y=2), \cdots, q_i(y=l)]^T$ of non-negative elements summing to one. Such a distribution can be used for classification of l classes with the maximum likelihood principle, such that, the vertex is labeled as a member of class $y^* = \arg\max_y q_i(y)$.

We estimate the distributions q_i for unlabeled vertices based on the distributions p_i of labeled vertices and the neighborhood relations encoded by e_i. Our main assumption is that multinomial distributions q_i should be similar to each other inside each edge. The general setting of distribution propagation is illustrated in Figure 2, where we have two classes (marked with different colors), the labeled distributions are p_1 and p_{12}, and the estimated distributions are q_i. We describe the discrepancy between two probability distributions p and q by the Kullback-Leibler (KL) divergence, $D(p,q) = \sum_{y\in\mathcal{Y}} p(y) \log \frac{p(y)}{q(y)}$. We also define the exponential centroid distribution of an edge e_k to be:

$$q_k^E(y) = \frac{1}{Z_k} \exp(\sum_{i\in e_k} h_{i,k} \log q_i(y)), \tag{1}$$

with Z_k being a normalization constant, and $h_{i,k} = 1/|e_k|$, or 0.5 in our case. Next we can formulate the optimization on the image graph as:

$$\arg\min_{q_i} \sum_{i=1}^{n} r_i D(p_i, q_i) + \sum_{k=1}^{m} w_k \sum_{i\in e_k} h_{i,k} D(q_k^E, q_i), \tag{2}$$

where r_i is 0 for an unlabeled vertex and a positive constant for a labeled vertex. Further, we can relax (2) to the following with the optimal solution q_i^* to be the same:

$$\arg\min_{q_i,\eta_k} \sum_{i=1}^{n} r_i D(p_i, q_i) + \sum_{k=1}^{m} w_k \sum_{i \in e_k} h_{i,k} D(\eta_k, q_i), \tag{3}$$

since $q_k^E(y)$ is the minimizer η_k^* for $J(\eta_k) = \sum_{i \in e_k} h_{i,k} D(\eta_k, q_i)$ [1]. Note that the function in (3) can be decomposed vertex-wise [28]. Based on this observation, and $\sum_{y \in \mathcal{Y}} q_i(y) = 1$, the following decomposed sub-problem for vertex i

$$\arg\min_{q_i} r_i D(p_i, q_i) + \sum_{\{k: i \in e_k\}} h_{i,k} w_k D(\eta_k, q_i), \tag{4}$$

is solved in closed form by:

$$q_i(y) = \frac{1}{r_i + d_i}\left(r_i p_i(y) + \sum_{\{k: i \in e_k\}} h_{i,k} w_k \eta_k(y)\right), \tag{5}$$

where $d_i = \sum_{\{k: i \in e_k\}} h_{i,k} w_k$ is the vertex degree.

We take r_i to be a large number, and thus for a labeled vertex, $q_i(y) = p_i(y)$ meaning that our method performs interpolation respecting the user supplied labels. It can be shown that the function to optimize in (2) is convex with respect to all the $q_i(y)$ terms [28], and thus a *unique* solution is guaranteed. This is a nice property for the proposed framework, as after we add in the automatically generated distributions at the critical pixels, it is unnecessary to start over from the initial uniform distributions at other unlabeled vertices. Finally, the algorithm with guaranteed convergence is posed as iterations over the two major steps: (1) compute the centers $q_k^E(y)$ for all k, and set $\eta_k = q_k^E$; (2) solve the optimization sub-problem at each vertex i by updating $q_i(y)$ as in (5).

3 Incorporating Boundary Information

After distribution propagation on a graph corresponding to the image, we obtain reliable estimates of segment labels at pixels where there is minimal ambiguity based on appearance features. For certain regions in the image, prior segment labeling may be missing. We will refer to pixels in such regions as critical pixels. In general, critical pixels are inside segments whose corresponding vertices are disconnected from (or weakly connected to) both kinds of labeled vertices. For single object segmentation, we refer to the object and background as o^1 and o^2 respectively. Intuitively, the pixels with $t_i = |\log \frac{q_i(o^1)}{q_i(o^2)}| < \tau$, where τ is a small positive number, are selected as critical pixels, and thus $\mathcal{A} = \{i : i \in U \wedge t_i < \tau\}$. Such intuition is formalized as a decision problem using the maximum a posteriori (MAP) rule. Specifically, the two classes are likely correct (C^1) and likely wrong (C^0) predictions resulting from the distribution propagation. Class C^0 corresponds to critical pixels, which are equivalent to those pixels with $P(C^0|t) > P(C^1|t)$, where $P(C^j|t) \propto P(t|C^j)P(C^j)$. Such probabilities can be computed from a set of images with ground truth segmentation, and thus the threshold

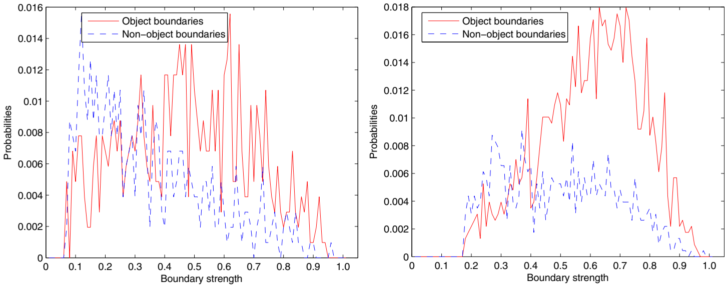


Fig. 3. Joint probabilities $P(strength, O)$ versus $P(strength, N)$, where the two classes of boundaries are object (O) and non-object (N). The relative greatness in them determines which class is more likely. It follows that higher strength implies higher chance of being an object boundary.

is set as $\tau = \max_t\{t : P(C^0|t) > P(C^1|t)\}$. Besides, in training the Dirichlet process based model, we set $\mathcal{U}^* = \{i : i \in \mathcal{U} \wedge t_i > \kappa\}$, such that $P(t > \kappa|C^0) < 5\%$.

We will next discuss a non-parametric Bayesian approach to predict segment distributions at critical pixels, which are then incorporated to the known set of labels. The distribution representation gives a principled encoding of the added segment labels with uncertainty. In regards to labeling weights, unlike original labels in which we have total confidence, we set r_i to be a small number for $i \in \mathcal{A}$, such that the effect of occasional erroneous labels can be offset by correct manual labels. After the label set augmentation, we resume, instead of restart, the distribution propagation to derive an updated segmentation result with newly added segment label distributions.

3.1 Boundary Induced Diffusion Signatures

In order to facilitate the incorporation of salient boundaries, we adopt the boundary probability described in [18]. Intuitively, the boundaries internal to semantic objects are weaker in strength (i.e., the mean value of boundary probabilities) than the external ones that outline objects. Such a claim is validated by empirical analysis on GrabCut and LHI datasets [24,30] with object-level segmentation ground truth. In this analysis, we observe that high strength in boundaries is strongly correlated with large probability of being true object boundaries, as plotted in the left (GrabCut dataset) and right (LHI dataset) panels of Figure 3 respectively.

As a result, the spatial location of a pixel with respect to the boundaries provides useful cues for determining its segment label. To get a reliable representation of such information, we first threshold and transform the probability map I_b to a set of boundary fragments $B = \{B_i\}$ by eliminating boundary pixels with small probability values, where each boundary fragment B_i is identified with the set of pixels that it intersects. An undirected boundary characteristic graph G_b is defined as: $G_b = (V, E)$, where the vertex set $V = \{v_i\}$ represents the set of pixels. The edge set $E = E^0 - E^*$, where $E^0 = \{(v_i, v_j) : v_i \text{ and } v_j \text{ are neighbors}\}$ is the full set of initial edges, and

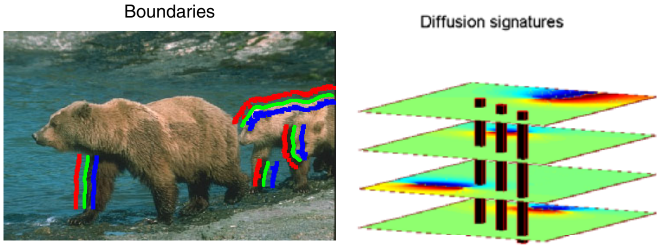


Fig. 4. Left: an image with four displayed boundary fragments together with red (+) and blue (-) heat sources. Right: the corresponding dimensions of diffusion signatures.

$E^* = \{(v_i, v_j) : \exists k \text{ s.t. } v_i \in B_k \vee v_j \in B_k\}$ is the set of edges that must be disconnected to reflect boundary constraints. By doing so, the salient boundaries are transformed to the characteristic graph G_b.

Diffusion signatures are feature vectors at vertices of G_b, and are derived using a diffusion process on the graph. Specifically, the i^{th} dimension of the diffusion signature vector corresponds to the i^{th} boundary fragment, and we place labeling "heat sources" on its both sides (left panel, Figure 4) to generate numerical descriptions. Let $f^i : V \rightarrow [-1, +1]$ be a function, $S = \{S_j\}$ be the set of vertices corresponding to labeling heat sources and U be the set of unlabeled vertices. Let λ be the mean value of boundary probabilities at pixels inside B_i. We proceed by assigning $+\lambda$ and $-\lambda$ as the values of function $f^i(S_j)$ on the two sides of the boundary fragment respectively, such that $f^i(\cdot)$ takes the opposite values on its two sides. In a vector form $\mathbf{f}^i_S = [f^i(S_1), f^i(S_2), \cdots, f^i(S_{|S|})]^T$. The stationary solution vector at unlabeled vertices is computed as: $\mathbf{f}^i_U = -\Delta^{-1}_{U,U}\Delta_{U,S}\mathbf{f}^i_S$, where $\Delta_{\cdot,\cdot}$ is a sub-Laplacian matrix of the graph G_b using block matrix notation [13]. Thus, the i^{th} dimension of diffusion signature at vertex v is defined as $f^i(v)$, which equals $\mathbf{f}^i_U(v)$ if $v \in U$, or $\mathbf{f}^i_S(v)$ if $v \in S$. In a similar vein, the diffusion signature vector at v is $[f^1(v), f^2(v), \cdots, f^{|B|}(v)]^T$. Example diffusion signatures associated with some boundary fragments are displayed on an image in the right panel of Figure 4, where red and blue refer to positive and negative values respectively. Properties of diffusion signatures in a single dimension include (1) the farther the pixels are from the boundary in the diffusion sense, the smaller the absolute values are; (2) the border at which $\{f^i(v) > 0\}$ and $\{f^i(v) < 0\}$ meets corresponds to a natural extension of the original boundary.

3.2 Non-linear Modeling and Classification

Now we present how the distribution at an unlabeled pixel $i \in \mathcal{A}$ is estimated using boundary information. The inputs to this process are the diffusion signatures derived from salient boundaries. The classification method we adopt is akin to Dirichlet process mixture models (DPMMs). Unlike Gaussian mixtures, they allow for automatic determination of the number of clusters. Our method is a semi-supervised extension of a recent DPMM based classification technique termed as Dirichlet process multinomial logistic model (DPMNL) [26]. We use semi-supervised learning to directly address the problem of insufficient labeling in interactive image segmentation.

In this model, a cluster refers to a group of salient diffusion signatures corresponding to structures in the image. A single cluster might contain both object and background pixels, which is accounted for by a multinomial logistic relation between the input diffusion signatures $\mathbf{x}$ and an output class y. Specifically, each cluster in the mixture model has parameters $\theta = (\mu, \Sigma, \alpha, \beta)$. The distribution of $\mathbf{x}$ within the cluster follows a Gaussian model $\mathcal{N}(\mu, \Sigma)$, where Σ is a diagonal matrix with elements σ_i^2. The distribution of y given $\mathbf{x}$ within the cluster follows a multinomial logistic model $P(y = j|\mathbf{x}, \alpha, \beta) = \frac{\exp(\alpha_j + \mathbf{x}^T \beta_j)}{\sum_{k=1}^{J} \exp(\alpha_k + \mathbf{x}^T \beta_k)}$, where J is the number of segments and equals 2 for single object segmentation. In the left panel of Figure 5, we illustrate the graphical model representation of DPMNL, which reveals the interdependencies among parameters and variables. Specifically, the model parameters θ are drawn from a distribution G that is drawn from a Dirichlet process $\mathcal{D}(G_0, \gamma)$, where G_0 is a base distribution over model parameters and γ is a scale parameter with a gamma prior. G_0's parameters may in turn depend on higher-level hyperparameters. However, we use fixed distributions for simplicity and they led to good performance: $\mu_i \sim \mathcal{N}(0, 1)$, $\log(\sigma_i^2) \sim \mathcal{N}(0, 1)$, $\alpha_j \sim \mathcal{N}(0, 1)$, $\beta_j \sim \mathcal{N}(\mathbf{0}, I)$, where i and j are for a feature dimension and a class respectively, and I is an identity matrix. Without loss of generality, we graphically show the effect of the model on a toy scenario in the right panel of Figure 5, where the two classes are displayed as dots and squares.

We use the Markov chain Monte Carlo (MCMC) algorithm with auxiliary parameters [20] for posterior sampling, which iterates over updating the data to cluster assignment and the cluster parameters. Our problem is relatively small in scale with several thousand training examples per image, and thus computational complexity of MCMC is affordable. In the main iteration of MCMC, we use semi-supervised learning to make use of the unlabeled pixels. Specifically,

$$E\{P(y_i|\mathbf{x}_i)\} = P(y_i = 1|\mathbf{x}_i)q_i(1) + P(y_i = 2|\mathbf{x}_i)q_i(2), \tag{6}$$

which is used in place of the conditional $P(y_i|\mathbf{x}_i)$ of an unlabeled training example (i.e., $i \in \mathcal{U}^*$). Inside this equation, $q_i(\cdot)$ values are obtained by distribution propagation. By doing so, the unlabeled examples are effectively used, instead of being discarded in the training process as in [26].

Once we obtain post-convergence parameters $\theta_i^t = (\mu_i^t, \Sigma_i^t, \alpha_i^t, \beta_i^t)$, for $t = 1, \cdots, T$ and $i = 1, \cdots, |\mathcal{L}| + |\mathcal{U}^*|$, where T is the maximum index of iteration, they are used to estimate the predictive distribution of the class label y_* for a new input diffusion signature vector $\mathbf{x}_*$:

$$P(y_* = j|\mathbf{x}_*) = \frac{\sum_{t=1}^{T} \int P(y_* = j, \mathbf{x}_*|\theta_*^t) P(\theta_*^t|\theta^t, G_0) d\theta_*^t}{\sum_{t=1}^{T} \int P(\mathbf{x}_*|\theta_*^t) P(\theta_*^t|\theta^t, G_0) d\theta_*^t}, \tag{7}$$

where the test example's parameters θ_*^t are drawn from a distribution that is drawn from a Dirichlet process $\mathcal{D}(G_0, \gamma)$: $\theta_*^t \sim \frac{1}{n+\gamma} \sum_{i=1}^{n} \delta(\theta_i^t) + \frac{\gamma}{n+\gamma} G_0$, in which $\delta(\cdot)$ is a distribution concentrated at a single point [4], and $n = |\mathcal{L}| + |\mathcal{U}^*|$. Therefore, equation (7) allows for numerical computation according to the assumed and derived distributions. Note that the predictive distribution is not based on a single parameter estimate,

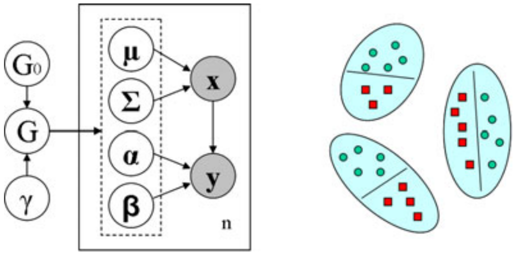


Fig. 5. Left: graphical model representation of DPMNL, where n is the number of training examples. Right: an illustration of its effect on data with 3 clusters. Inside each cluster, a linear decision boundary separates data from the 2 classes. The overall decision boundary is non-linear and not shown here.

but is an average of the predictions using all possible values of the parameters, each of which is weighted by the probability of the parameters having those values. In this way, DPMNL avoids making hard decisions on assigning diffusion signatures to clusters and thus provides flexibility in modeling the boundary information. Finally, each test diffusion signature vector is assigned to a segment class with the highest predictive probability, i.e., $\hat{y}_* = \arg\max_j P(y_* = j|\mathbf{x}_*)$ for the visualization of added labels. However, for the segmentation task, we set $p_i(y = j) = P(y_* = j|\mathbf{x}_*)$ for $i \in \mathcal{A}$ to accommodate the probabilistic nature of added labels.

4 Experiments

In order for quantitative evaluation, we perform experiments on the GrabCut dataset [24], which is one of the few datasets providing both labeling trimaps and segment ground truth. Labeled object, labeled background and unlabeled pixels are reflected in the trimaps as white, dark gray and light gray respectively. In experiments, we stick to original trimaps as provided in the dataset. We will report the performance on *set*-50 (the whole GrabCut dataset) and *set*-25 (a subset where there are a significant number of added labels). While our method helps boost the performance on the whole set, as can be seen, it enhances the performance on *set*-25 by a larger margin, which contains difficult instances due to the lack of labels at numerous critical pixels.

Details of implementation. In implementing our framework, we use superpixels [23] in place of pixels for computational efficiency, such that a superpixel, which roughly contains 15 pixels, is assumed the smallest unit for labeling. In computing diffusion signatures, boundary fragments are generated by thresholding a boundary probability map [18] at its 60^{th} percentile of non-zero values. The boundary fragments with junctions are broken into smaller ones, so that they contain simple curves for generating diffusion signatures. We also project the diffusion signatures onto the subspace learned by principal component analysis (PCA) and retain the top half dimensions to have a compact representation. Thus, the constituent dimensions of small variances resulting from boundaries with small strength are filtered out. In constructing the set $\mathcal{A}$ for automatic label set augmentation as discussed in Section 3, with an empirical analysis of the segmentation results on the GrabCut dataset using distribution propagation, τ is chosen using an MAP procedure, such that $\tau = \max_t\{t : P(C^0|t) > P(C^1|t)\} \approx 0.004$. Finally, κ is chosen as 0.845, such that $P(t > \kappa|C^0) < 5\%$, i.e., a label in $\mathcal{U}^*$ is very unlikely to be wrong.

Methods for comparison. We have used label propagation (LP) and distribution propagation (DP) as the underlying labeling methods. Together with them, we study several approaches for label set augmentation as detailed next.

- LP-original: An implementation of the random walk approach [13], where the optimal label values at unlabeled vertices are derived by solving linear equations and are thresholded to produce segment labels.
- DP-original: distribution propagation as introduced in Section 2, where in contrast to the label value representation at each vertex, a distribution is used.
- DP-GMM: DP with added labels estimated using Gaussian mixture models learned with EM [10]. A mixture model is trained for each segment class.
- DP-SVM: DP with added labels estimated using support vector machines with radial basis function kernels and probability outputs implemented in [7].
- DP-object: DP with added labels systematically being object, which produces considerably better performance than randomly guessing on the GrabCut dataset.
- DP-DPMNL: DP with added labels estimated with DPMNL, which is the proposed method.
- DP-DPMNL-AT: DP with added labels estimated with DPMNL; however, the threshold for distribution propagation is set according to adaptive thresholding [14].
- LP-DPMNL: LP with the same added labels as above.

Evaluation. For quantitative evaluation of object-background segmentation on the GrabCut dataset, we compute the error rate as the percentage of wrongly labeled pixels in the original unlabeled region. In addition, we demonstrate the error rate of automatic label set augmentation, which is the percentage of wrongly added labels at critical pixels. We note that while the added labels are not 100% accurate, distribution propagation uses these additional labels in a soft and flexible way, so the overall enhancement is not compromised by occasional erroneous labels.

The results are tabulated in Table 1, where *seg*-50/25 refers to the average error rate of segmentation and *aug*-50/25 refers to the average error rate of label set augmentation. Among all the compared methods, the proposed DP-DPMNL outperforms the others without heuristic thresholding. It can be further seen that the distribution propagation alone performs better than the label propagation, such as an error rate of 5.4% reported for a recent method detailed in [12] which uses sophisticated Laplacians. All label set augmentation methods that we use help to achieve better segmentation accuracy but at different levels; DPMNL provides better results than the other three alternatives. In particular, we attribute the less competitive performance of SVMs to their inability in modeling the cluster structures of diffusion signatures. Besides, DP-DPMNL outperforms

LP-DPMNL, which justifies the choice of distribution propagation as the underlying labeling mechanism.

A comparison with the published average error rates on the entire dataset using several recent approaches also shows that our method performs the best. In particular, our proposed method gives 3.58%, considerably better than error rates achieved in previous works [5,13,12,11]. Together with adaptive thresholding [14], our proposed method produces an error rate 3.08%, which is better than 3.3% that is generated using s-Laplacian together with adaptive thresholding. Despite this fact, we stress that the thresholding technique has heuristic nature. Thus, DP-DPMNL remains the proposed method of this paper and is used to generate all the remaining results, even though DP-DPMNL-AT can give better quantitative results given the GrabCut type of trimaps.

Table 1. Performance in terms of the average error rate in segmentation (*seg*) and that in label set augmentation (*aug*) on the whole set of 50 images and a smaller set of 25 where there are a significant number of missing labels. Compared are variations of our proposed method and recent baselines in the literature. Our proposed method (DP-DPMNL) has shown the best results among all, except for methods combined with adaptive thresholding [14].

Methods	*seg*-50	*aug*-50	*seg*-25	*aug*-25
LP-original	5.92%	–	7.03%	–
DP-original	5.22%	–	6.84%	–
DP-GMM	4.68%	25.25%	5.33%	26.72%
DP-SVM	4.49%	22.08%	5.01%	22.87%
DP-object	5.04%	26.61%	5.66%	30.61%
DP-DPMNL	**3.58%**	**11.08%**	**3.85%**	**15.35%**
DP-DPMNL-AT	**3.08%**	–	**3.14%**	–
LP-DPMNL	4.82%	–	5.63%	–
GM-MRF [5]	7.9%	–	–	–
Random walk [13]	5.4%	–	–	–
s-Laplacian [12]	5.4%	–	–	–
s-Laplacian-AT [12]	3.3%	–	–	–
Hypergraph [11]	5.3%	–	–	–

Sample segmentation results and automatically added labels by using DP-DPMNL are shown in Figure 7 for qualitative evaluation.[1] From top down, the four rows are respectively: original trimaps, results from the baseline method (DP-original), automatically added labels where red is for object and green is for background, and new results using added labels. As can be observed, the proposed method provides smooth object contours while preserving most details. Both missing object labels and background labels can be added at their appropriate places. Besides, for one of the images hard to DP-original due to insufficient labels, the evolution of the computed object contour is visualized in Figure 6, before (iterations 0 to 10) and after (11 to 20) the label set augmentation, where iteration 0 refers to the initial condition.

[1] Additional qualitative segmentation results in the same format are shown in `http://www.cse.ohio-state.edu/~dinglei/autogen.htm`

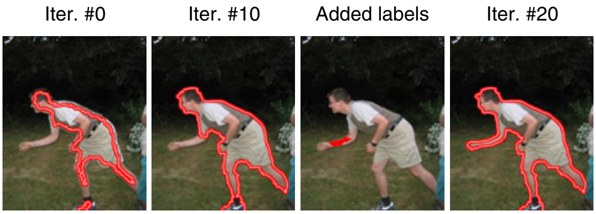


Fig. 6. Contour evolution by our approach. Please refer to Figure 7 for the provided trimap that contains insufficient labels.

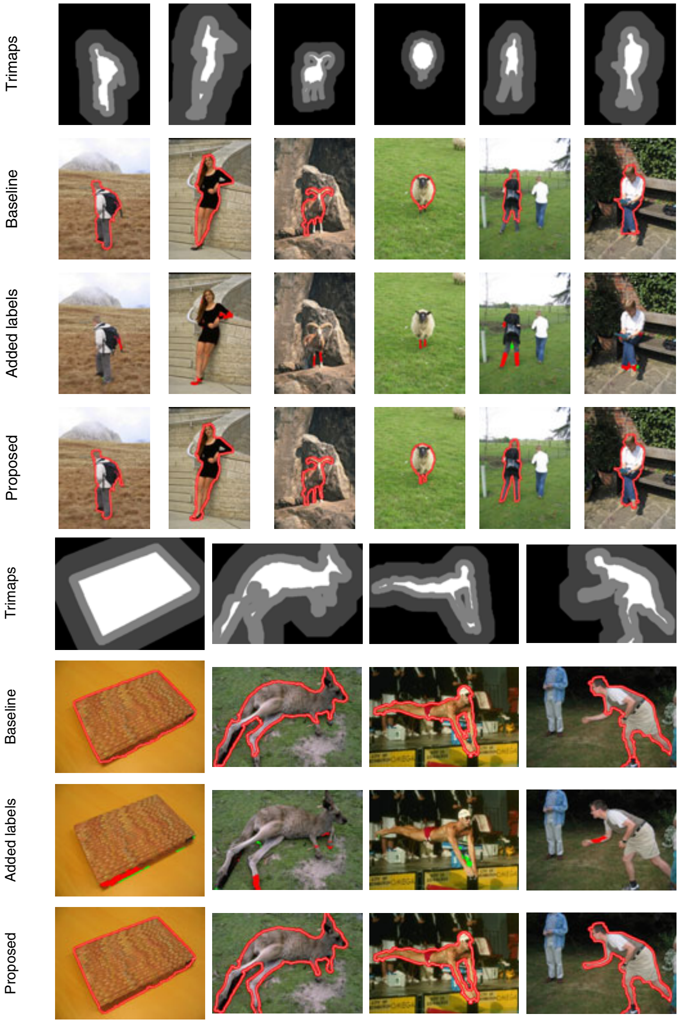


Fig. 7. Provided trimaps, baseline results, automatically added labels (red: object, green: background) and new segmentation results

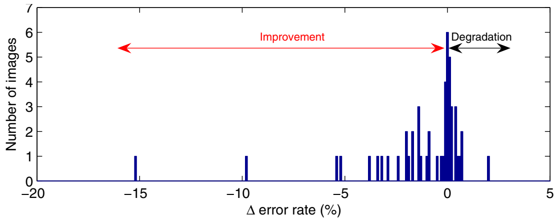


Fig. 8. Histogram of the change in error rates, or the error rate of using DP-DPMNL minus that of using DP-original, on *set*-50. Improved cases are marked red, and the degraded are in black.

5 Discussion and Conclusions

Although user interaction has been widely used for segmentation, the insufficiency in provided labeling seeds has drawn little attention. In this paper, we addressed this by exploiting the boundary information. Our experiments on the GrabCut dataset have shown that automatically added labels helped the segmentation process at different success levels. The histogram of change in segmentation error rates with label set augmentation is shown in Figure 8. We note that the proposed method has dramatically reduced the error rates in many cases. There is only one case where the added labels reduced the accuracy rate by more than 1% (which is 2.0%). This reduction in quantitative performance was due to similarity between the object and its shadow which resulted in labeling the shadow as the object. However, the result does not deteriorate qualitatively.[2] Besides, the automatically added labels can be prompted to the user to accept or reject in an interactive environment, which is a functionality not previously offered.

To summarize, we have presented a framework using distribution propagation to address interactive image segmentation. A key component of our framework is the automatic augmentation of labels at critical pixels via a Dirichlet process based non-linear model. Extensive experiments have shown that the proposed framework performs competitively in predicting critical missing labels and enhancing the overall segmentation results. In particular, using a limited number of supplied labels, we have achieved both qualitatively and quantitatively excellent results on the GrabCut dataset.

References

1. Akaho, S.: The e-PCA and m-PCA: Dimension reduction of parameters by information geometry. In: IJCNN (2004)
2. Arbelaez, P., Cohen, L.: Constrained image segmentation from hierarchical boundaries. In: CVPR (2008)
3. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR (2009)
4. Blackwell, D., MacQueen, J.B.: Ferguson distributions via polya urn scheme. Annals of Statistics 1, 353–355 (1973)
5. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: ECCV (2006)

[2] For qualitative comparison, please look at the first column, bottom group of Figure 7.

6. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In: ICCV (2001)
7. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001) (software available online)
8. Corduneanu, A., Jaakkola, T.: Distributed information regularization on graphs. In: NIPS (2005)
9. Das, P., Veksler, O., Zavadsky, V., Boykov, Y.: Semiautomatic segmentation with compact shape prior. Image and Vision Computing 27(1-2), 206–219 (2009)
10. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39(1), 1–38 (1977)
11. Ding, L., Yilmaz, A.: Interactive image segmentation using probabilistic hypergraphs. Pattern Recognition 43(5) (2010)
12. Duchenne, O., Audibert, J.Y., Keriven, R., Ponce, J., Segonne, F.: Segmentation by transduction. In: CVPR (2008)
13. Grady, L.: Random walks for image segmentation. IEEE PAMI 28(11), 1768–1783 (2006)
14. Guan, J., Qiu, G.: Interactive image segmentation using optimization with statistical priors. In: ECCV Workshops (2006)
15. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. IJCV 1(4), 321–331 (1987)
16. Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: ICCV (2009)
17. Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using contours to detect and localize junctions in natural images. In: CVPR (2008)
18. Martin, D., Fowlkes, C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE PAMI 26(5), 530–549 (2004)
19. Mortensen, E.N., Barrett, W.A.: Intelligent scissors for image composition. In: SIGGRAPH (1995)
20. Neal, R.: Markov chain sampling methods for Dirichlet process mixture models. Journal of Computational and Graphical Statistics 9(2), 249–265 (2000)
21. Peng, B., Veksler, O.: Parameter selection for graph cut based image segmentation. In: BMVC (2008)
22. Price, B.L., Morse, B., Cohen, S.: Geodesic graph cut for interactive image segmentation. In: CVPR (2010)
23. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV (2003)
24. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: SIGGRAPH (2004)
25. Settles, B.: Active learning literature survey. Tech. rep., University of Wisconsin-Madison Computer Sciences Technical Report 1648 (2009)
26. Shahbaba, B., Neal, R.: Nonlinear models using Dirichlet process mixtures. JMLR 10, 1755–1776 (2009)
27. Stein, A.N., Stepleton, T.S., Hebert, M.: Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In: CVPR (2008)
28. Tsuda, K.: Propagating distributions on a hypergraph by dual information regularization. In: ICML (2005)
29. Veksler, O.: Star shape prior for graph-cut image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 454–467. Springer, Heidelberg (2008)
30. Yao, Z., Yang, X., Zhu, S.C.: Introduction to a large scale general purpose ground truth dataset: Methodology, annotation tool, and benchmarks. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) EMMCVPR 2007. LNCS, vol. 4679, pp. 169–183. Springer, Heidelberg (2007)
31. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: ICML (2003)

Hough Transform and 3D SURF for Robust Three Dimensional Classification

Jan Knopp[1], Mukta Prasad[2], Geert Willems[1],
Radu Timofte[1], and Luc Van Gool[1,2]

[1] KU Leuven
[2] ETH Zurich

Abstract. Most methods for the recognition of shape classes from 3D datasets focus on classifying clean, often manually generated models. However, 3D shapes obtained through acquisition techniques such as Structure-from-Motion or LIDAR scanning are noisy, clutter and holes. In that case global shape features—still dominating the 3D shape class recognition literature—are less appropriate. Inspired by 2D methods, recently researchers have started to work with local features. In keeping with this strand, we propose a new robust 3D shape classification method. It contains two main contributions. First, we extend a robust 2D feature descriptor, SURF, to be used in the context of 3D shapes. Second, we show how 3D shape class recognition can be improved by probabilistic Hough transform based methods, already popular in 2D. Through our experiments on partial shape retrieval, we show the power of the proposed 3D features. Their combination with the Hough transform yields superior results for class recognition on standard datasets. The potential for the applicability of such a method in classifying 3D obtained from Structure-from-Motion methods is promising, as we show in some initial experiments.

1 Introduction

A number of methods for 3D shape class recognition have been proposed already. So far, the dominant line of work has been to use global features, *i.e.* features that need the complete, isolated shape for their extraction. Examples are Fourier or spherical harmonics [1,2], shape moments [2], shape histograms [3]. There are at

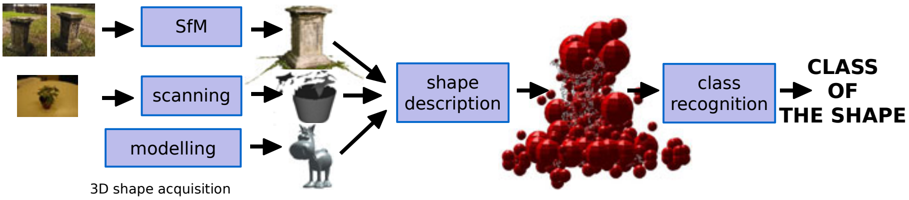


Fig. 1. Proposed approach classifies noisy 3D shapes obtained from SfM, scans etc. The method is invariant to the texture and recognizes difficult objects such as plants.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 589–602, 2010.

least three potential problems with these global approaches: (i) it is difficult to handle partial shapes. For instance, when an artifact has been damaged, even the most perfect scan will still only capture a part of what the original shape should have been, (ii) many capturing scenarios contain irrelevant, neighbouring clutter in addition to the relevant data coming from the object. Global methods mix the two, jeopardizing class recognition. Some local, skeleton-based descriptions are also known to suffer from these problems (e.g. [4]), (iii) several classes contain deformable shapes, some parts of which may be more deformable than other more rigid parts. Global methods are also less successful at handling intra-class variations while remaining sufficiently discriminative to noise, clutter, articulated deformations and inter-class variations. In many 3D application based on retrieval, classification and detection, all these three problems have to be addressed.

As work in 2D object class recognition has shown, the use of local rather than global features is advantageous. 2D class detection methods deal with occlusions and clutter quite successfully already. We therefore seek to apply these techniques to the 3D case as well. So far, relatively few 3D categorisation methods based on local features, like tensors [5], heat kernel signatures [6], integral shape descriptors [7,8], and scale dependent features [9] have been proposed.

Ovsjanikov *et al.* [10] extended the standard bag-of-features (BOF) approach of Sivic and Zisserman [19] by looking for the frequency of word pairs instead of the single word, called spatially-sensitive bags of features. Toldo *et al.* [11] described 3D shapes by splitting them into segments, which are then described on the basis of their curvature characteristics. These descriptors are quantized into a visual vocabulary. Finally, an SVM is learnt for the actual categorisation. Methods that use other information than pure shape (e.g. [12,13]) are not considered here because we are interested in the still-common case where no other information is available.

The afore-mentioned methods assume clean, pre-segmented shapes, *i.e.* without them being attached to a 3D 'background'. As such, these BOF approaches could suffer from the problem that the information can get buried under clutter, especially when the object of interest is small compared to this background. In 3D this difference is magnified. For instance, a statue of a person in front of a building may cover a large part of the 2D image scene, but will be tiny compared to the size of the building in 3D, where all objects appear with their actual, relative scales. In Hough transform based approaches, the process of recognition is tied up with hypothesis verification (through object localization). This means that it has higher discriminative power against clutter than BOF based approaches.

This paper proposes an approach to 3D shape categorisation that can perform better at the tasks described above. A 3D extension to SURF [14] serves as our local descriptor described in § 2. This feature has proved quite effective in 2D and can now be viably computed even in 3D. In contrast to a dense or random coverage with spin images [15], a 3D interest point detector picks out a repeatable and salient set of interest points. These descriptors are quantized and used in a Hough approach, like Implicit Shape Model (ISM) [16], which keeps

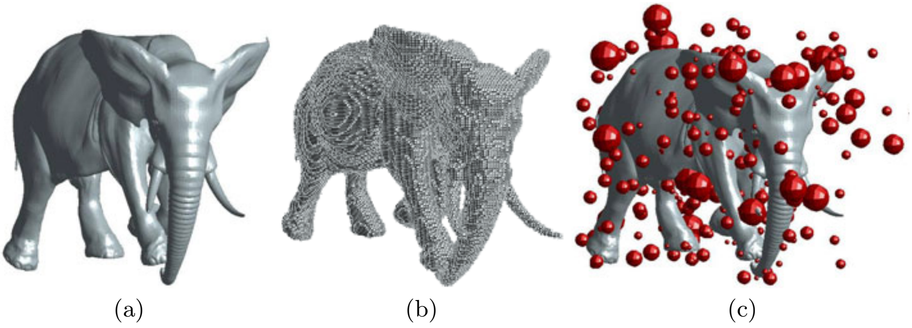


Fig. 2. Illustration of the detection of 3D SURF features. The shape (a) is voxelized into the cube grid (side of length 256) (b). 3D SURF features are detected and back-projected to the shape (c), where detected features are represented as spheres and with the radius illustrating the feature scale.

the influence of each feature better localized than in a BOF approach as seen in § 3. Our approach favorably compares to the state-of-the-art in 3D shape class recognition and retrieval as seen in § 4, § 5.

2 Shape Representation as the Set of 3D SURF Features

For our problem of class recognition, we collected a set $\mathcal{M}$ of shapes separated into two disjoint sets: (i) training data $\mathcal{M}_T$ and (ii) query data $\mathcal{M}_Q$. The m^{th} 3D shape is represented as $\{\mathtt{V_m}, \mathtt{F_m}\}$, where $\mathtt{V_m}$ is a collection of vertices and $\mathtt{F_m}$ is a collection of polygons (specifically triangles) defined on these vertices.

In order to describe each shape $m \in \mathcal{M}$ as a set of local rotation and scale-invariant interest points, we propose an extension of SURF to 3 dimensions. It is important to note at this point, that this extension can also be seen as a special case of the recently proposed Hessian-based spatio-temporal features by Willems *et al.* [17], where temporal and spatial scale are identical. As such, the theoretical results that were obtained from scale space theory still hold. Furthermore, most of the implementation details can be reused, except the fact that the search space has now shrunk from 5 to 4 dimensions (x, y, z, σ). For more in-depth information on Hessian-based localization and scale selection in 3 dimensions, we refer the reader to [17].

The extraction of the 3D features is as follows. First, we voxelize a shape in a volumetric 3D cube of size 256^3 using the intersection of faces with the grid-bins as shown in figure 2(b), after each shape is uniformly scaled to fit the cube while accounting for a boundary of 40 at each side. The cube parameters were chosen empirically. Next, we compute a saliency measure S for each grid-bin $\boldsymbol{x}$ and several scales σ (over three octaves). We define S as the absolute value of the determinant of the Hessian matrix $H(\boldsymbol{x}, \sigma)$ of Gaussian second-order derivatives $L(\boldsymbol{x}, \sigma)$ computed by box filters,

$$S(\boldsymbol{x},\sigma) = \left|H(\boldsymbol{x},\sigma)\right| = \left|\begin{pmatrix} L_{xx}(\boldsymbol{x},\sigma) & L_{xy}(\boldsymbol{x},\sigma) & L_{xz}(\boldsymbol{x},\sigma) \\ L_{yx}(\boldsymbol{x},\sigma) & L_{yy}(\boldsymbol{x},\sigma) & L_{yz}(\boldsymbol{x},\sigma) \\ L_{zx}(\boldsymbol{x},\sigma) & L_{zy}(\boldsymbol{x},\sigma) & L_{zz}(\boldsymbol{x},\sigma) \end{pmatrix}\right|, \tag{1}$$

as proposed in [17]. This has as implication that, unlike in the case of SURF [14], a positive value of S does not guarantee that all eigenvalues of H have identical signs. Consequently, not only blob-like signals are detected, but also saddle points. Finally, K_m unique features: $\mathbf{d}_{mk}, k \in \{1 \dots K_m\}$ are extracted from the volume using non-maximal suppression (see [17] for more details).

In a second stage, a rotation and scale-invariant 3D SURF descriptor is computed around each interest point. First, we compute the local frame of the feature. We therefore uniformly sample Haar-wavelet responses along all 3 axes within a distance $3 \times \sigma$ from each feature. Next, each response is weighted with a Gaussian centered at the interest point, in order to increase robustness to small changes in position. Each weighted response is plotted in the space spanned by the 3 axes. We sum the response vectors in all possible cones with an opening angle of $\pi/3$ and define the direction of the longest resulting vector as the dominant orientation. However, instead of exhaustively testing a large set of cones uniformly sampled over a sphere, we approximate this step by putting a cone around each response. After the dominant direction has been obtained, all responses are projected along this direction after which the second orientation is found using a sliding window [14]. The two obtained directions fully define the local frame. Defining a $N \times N \times N$ grid around the feature and computing the actual descriptor, is implemented as a straight-forward extension of the 2D version. At each grid cell, we store a 6-dimensional description vector of Haar wavelet responses as in [17]. In the rest of the paper, we assume $N = 3$.

For the feature k of the shape m we maintain a tuple of associated information as shown below:

$$\mathbf{d}_{mk} = \left\{ \underset{3\times 1}{\mathbf{p}_{mk}} \ , \ \sigma_{mk} \ , \ \underset{162\times 1}{\mathbf{s}_{mk}} \right\}, \tag{2}$$

where $\mathbf{p}_{mk}$ represents the relative 3D position of the feature point to the shape's centre, σ_{mk} is the scale of the feature point and $\mathbf{s}_{mk}$ is 162-dimensional 3D SURF descriptor vector[1] of the feature vector $\mathbf{d}_{mk}$.

3 Implicit Shape Model for 3D Classification

In order to correctly classify query shapes, we need to assemble a model of each class based on the local 3D SURF features, and define a ranking function to relate a shape to each class. The Implicit Shape Model converts the SURF features to a more restricted 'visual vocabulary' generated from training data. We will discuss this in § 3.1. Based on the information acquired during training, each visual word on a query shape then casts weighted votes for the location of the shape center for a particular class, which will be seen in § 3.2. Depending

[1] $3 \times 3 \times 3 \times 6 = 162$.

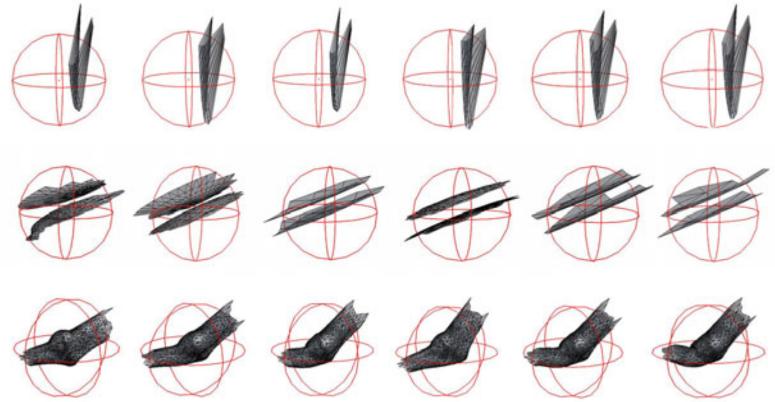


Fig. 3. Each row shows some partial 3D shapes from which features were computed that belong to the same visual word. The feature center is represented as a red dot, while the sphere represents the feature scale. Each shape is shown normalized with respect to the scale of the feature.

on whether the query shape's center is already known, the above information is used for classification in two ways as outlined in § 3.3.

3.1 Visual Vocabulary Construction

To reduce the dimensionality of feature matching and limit the effects of noise, we quantize the SURF features to a vocabulary of visual words, which we define as the cluster centers of an approximate K-means algorithm (see Muja *et al.* [18]). Following standard practice [19,20] in large-scale image searching, we set the number of visual words (clusters) to 10% of the total number of features in our training set. In practice, this yields a reasonable balance between mapping similar shapes to the same visual word (Fig. 3) while ensuring that features that are assigned the same word are indeed likely to correspond (Fig. 4).

3.2 Learning and Weighting Votes

Rather than storing a shape for each class, the ISM-based methods keep track of where a visual word v would be located on a shape of class c relative to c's center ([16,21]). This information — the collection of visual words and offsets from shape centers — is assembled from the training set, and stored along with the visual words themselves. Word v is therefore associated with a list of votes, each of those being generated from a feature (introduced in Eq. 2) and defined by the feature's class c, its vector to the shape center (x', y', z'), its scale σ', and the scale of the shape. Each word may therefore cast votes for multiple classes. Words may also cast multiple votes for the *same* class, as in Fig. 5, because there may be multiple features on a shape associated with the same visual word.

Suppose now that a query shape contains a feature at location $[x, y, z]^T$ with scale σ that is assigned to visual word v. That feature will cast a vote, λ, for a shape of class c centered at location

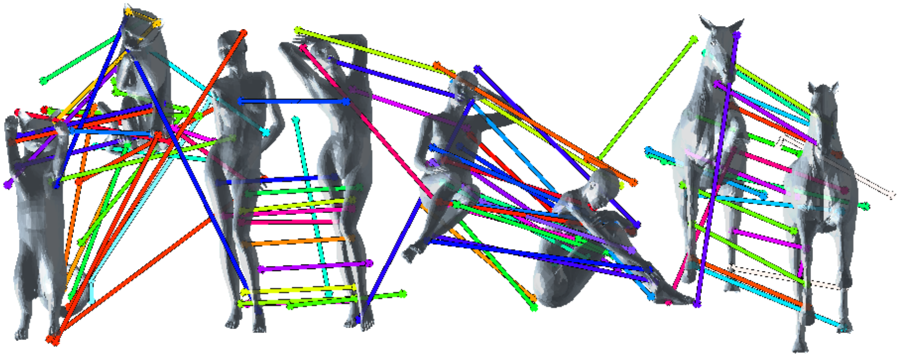


Fig. 4. Examples of visual vocabulary based correspondences between 3D shapes

$$\lambda = \left[x - x'(\sigma/\ \sigma'), y - y'(\sigma/\ \sigma'), z - z'(\sigma/\ \sigma'), \sigma/\ \sigma'\right]^T, \qquad (3)$$

with relative shape size $\sigma/\ \sigma'$. If the query shape exactly matches a training shape, the votes associated with that training shape will all be cast at the query shape's center, making a strong cluster of votes for the match. On the other hand, the votes associated with a training shape from a different class will get scattered around, because the spatial arrangement of features (and therefore visual words) will be different, see Fig. 5.

Note that although a single assignment of features to the closest visual word is natural, it is subject to noise when cluster centers are close together. Therefore, during the training phase, each feature activates the closest word and every other word within a distance τ, as in [16,20,22]. This ensures that similar visual words that are located at the same position on a shape will all vote appropriately.

An issue is that different classes may have different numbers of features, and not all features discriminate equally well between classes. We account for these next discuss factors with a pair of weights,

(i) a statistical weight W_{st} as every vote should be invariant to the number of training samples in the class,

(ii) a learned weight W_{lrn} weights every vote so it correctly votes for a class centre across training shapes.

(i) The statistical weight $W_{st}(c_i, v_j)$ weights all the votes cast by visual word v_j for class c_i by

$$W_{st}(c_i, v_j) = \frac{1}{n_{vw}(c_i)} \cdot \frac{1}{n_{vot}(v_j)} \cdot \frac{\frac{n_{vot}(c_i,v_j)}{n_{ftr}(c_i)}}{\sum\limits_{c_k \in \mathcal{C}} \frac{n_{vot}(c_k,v_j)}{n_{ftr}(c_k)}}, \qquad (4)$$

where the different numbers n are determined from the training set. For instance, $n_{vot}(v_j)$ is the total number of votes from visual word v_j, $n_{vot}(c_i, v_j)$ is the

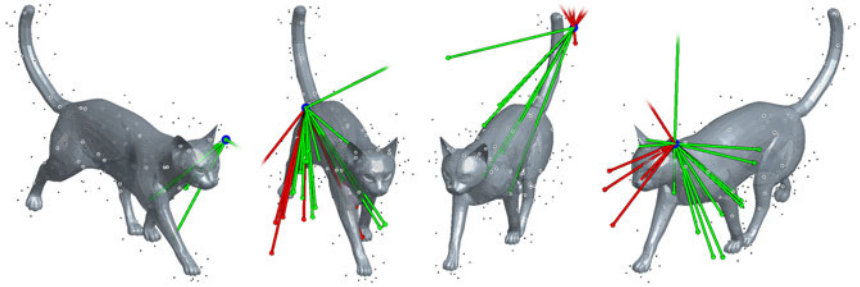


Fig. 5. Example of the votes cast from four features on a cat shape instance. All detected features are visualized as small black dots and votes are shown as lines starting from the feature (marked blue). The votes from a toy ISM model were learned from six shapes of the cat-class (visualized as green lines) and six shapes of flamingo-class (red lines).

number of votes for class c_i from v_j, $n_{vw}(c_i)$ is the number of visual words that vote for class c_i, $n_{ftr}(c_i)$ is the number of features from which c_i was learned. $\mathcal{C}$ is the set of all classes. The first term makes every class invariant to the number of visual words in its training set, while the second normalizes for the number of votes each visual word casts. The final term reflects the probability that v_j votes for class c_i as opposed to some other class.

(ii) Additionally, motivated by Maji's *et al.* [23] work, we normalize votes on the basis of how often they vote for the correct training shape centers (during training). We define λ_{ij} as the vote cast by a particular instance of visual word v_j on a particular *training* shape of class c_i; that is, λ_{ij} records the distance of the particular instance of visual word v_j to the center of the training shape on which it was found. We now apply this vote to *every* instance of visual word v_j on *every* training shape in class c_i, and compute a Gaussian function of the distance between the center position voted for and the actual center. This scheme puts more emphasis on features with voted positions close to that actual center.

For every vote λ_{ij}, our goal is to obtain one value summarizing the statistics of distances to shape centers,

$$W_{lrn}(\lambda_{ij}) = f\left(\left\{e^{-\frac{d_a(\lambda_{ij})^2}{\sigma^2}} \,\middle|\, a \in A\right\}\right), \tag{5}$$

where A is the set of all features associated with word v_j on a shape of class c_i and $d_a(\lambda_{ij})$ is the Euclidean distance as just defined. We use a standard deviation of σ taken as 10% of the shape size, which defines the accepted amount of noise. For the function f, we observed the best performance for the median.

The final weight is the combination of W_{st} and W_{lrn},

$$W(\lambda_{ij}) = W_{st}(v_j, c_i) \cdot W_{lrn}(\lambda_{ij}). \tag{6}$$

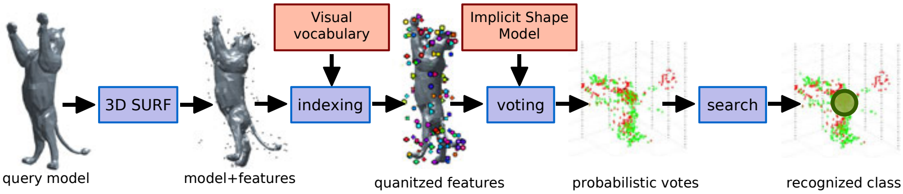


Fig. 6. Overview of our 3D ISM class recognition. On the query shape, 3D SURF features are detected, described and quantized into the visual vocabulary. Using the previously trained 3D Implicit Shape Model, each visual word then generates a set of votes for the position of the class center and the relative shape size. Finally, the recognized class is found at the location with maximum density of these votes.

3.3 Determining a Query Shape's Class

The class recognition decision for a given 3D query shape is determined by the set of 5D votes (shape center, size of the shape and class), weighted by the function W. However, we need a mechanism to cluster votes cast at nearby but distinct locations. Depending on the type of query shape, we use one of two approaches:

1. *Cube Searching (CS):* In the spirit of Leibe *et al.* [16], we discretize the 5D search space into bins; each vote contributes to all bins based on its Gaussian-weighted distance to them. The recognized class and shape center is given by the highest score. The principal advantage of this approach is that it does not require a clean query shape — noisy or partial query input is handled by explicitly searching for the optimal shape center as well as the class.
2. *Distance to Shape Center (DC):* Unlike image queries, where the shape's center within the image is usually unknown, it is quite easy to compute the centroid of a clean 3D shape, and use this as the shape center. Doing so can simplify class recognition and improve its robustness by reducing the search to the best class given this center. We do this by weighting each vote by a Gaussian of its distance to the query shape's center. Processing of such complete 3D shapes is a popular task in the 3D literature [11,10]. Obviously, the real object center coinciding with the shape center is not always valid and we cannot use it for partial shapes or for the recognition of 3D scenes (with additional clutter or noise).

4 Experiments and Applications

Our main target is to robustly classify 3D shapes. Having visually assessed the 3D SURF descriptors (§ 2) in Figs. (3,4), we evaluate it further for the difficult task of partial shape retrieval in § 4.2. Since the task is retrieval, the features are used in the BOF framework for this test. Supported by the good performance, we further use 3D SURF features in conjunction with the probabilistic Hough voting method of § 3 (ISM) to demonstrate its power for class recognition and for assessing the sensitivity to missing data on standard datasets. Our proposed method outperforms the other approaches in these clean shape datasets. Finally, we tackle classification of 3D scenes reconstructed from real life images. Such scenes are challenging due to clutter, noise and holes. We show promising results on such data in § 4.4.

4.1 Datasets

All the datasets (Fig. 9) used in our evaluations consists of clean and segmented shapes and are defined at the outset.

(i) KUL dataset: simple dataset of 94 training shapes of 8 classes from the Internet and 22 query shapes.
(ii) Princeton dataset: challenging dataset of 1.8K shapes (half training, half testing), 7 classes taken from the Princeton Benchmark [24].
(iii) Tosca+Sumner dataset: dataset for retrieval/classification [25,26] of 474 shapes, 12 classes of which 66 random ones form a test set.
(iv) SHREC'09 datasets: 40 classes, 720 training and 20 partial query shapes from the Partial Shape Retrieval Contest [27] with complete ground-truth.

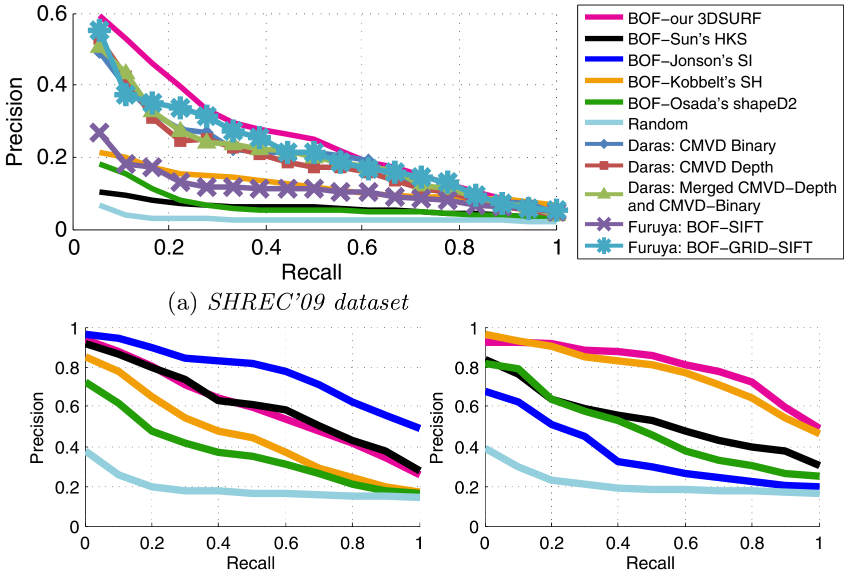
4.2 3D SURF Features for Shape Retrieval

We have presented a novel method for local features extraction and description for 3D shapes. We investigate now the performance of our approach to the state of the art descriptors.

As the task here is that of shape retrieval (as opposed to our classification based method from § 3), we use 3D SURF features in the large-scale image retrieval approach of Sivic and Zisserman [19] based on BOF. First, 3D SURF features of all shapes were quantized using the visual vocabulary as in § 3. Second, we compute the BOF vectors. Third, using the BOF, every shape model is represented as the normalized tf-idf vector [28] preferring the discriminative visual words. Finally, the similarity between shapes is measured as the L_1 distance between the normalized tf-idf vectors. L_1 measure was shown to perform better than the dot-product in image retrieval [19].

For the problem of partial shape retrieval 3D SURF is pitched against other descriptors in the SHREC'09 Contest [27] for the dataset (iv) in § 4.1. Fig. 7(a) presents our results together with results from the SHREC'09 Contest. Note that 3D SURF features outperform the rendered range-images -based SIFT descriptors [27], in similar BOF frameworks.

Fig. 7(b,c) shows the retrieval performance on two additional datasets. As the main result, we observed high sensitivity of all descriptors to to the dataset type, i.e. SI [15] outperforms all methods in Tosca dataset, while it gives the worst results on KUL dataset, but couldn't be evaluated on SHREC'09 due to computational constraints.



(a) *SHREC'09 dataset*

(b) *Tosca, small resolution shapes*

(c) *KUL dataset*

Fig. 7. Comparison of different detectors/descriptors using the video google [19] retrieval approach. The performance is measured as Precision-Recall curve. (a) SHREC'09 Partial Shape Retrieval Contest [27] provided results which were compared with our 3D SURF and other approaches. (b,c) Note that the performance highly depends on the shape's type as results very depend on dataset.

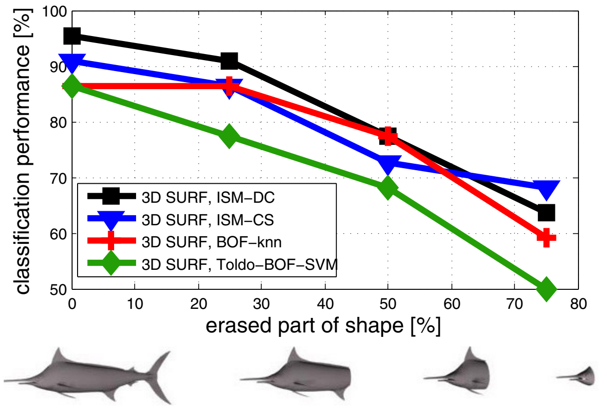


Fig. 8. Sensitivity of 3D classification to missing data. The classification performance is plotted as the shape is increasingly cropped. See the fish example on the bottom row. We found that our approach outperforms knn as well as Toldo's [11] SVM method.

We also observed (on shapes from 1.2K-65K faces and 670-33K vertices) that our method is faster than other local descriptors. In average, 3D SURF takes 20.66s, HKS [6] 111.42s and SI [15] more than 15mins. The experiment was performed on 4xQuad Core AMD Opteron, 1.25Ghz/core.

4.3 3D SURFs in the ISM Framework for 3D Classification

Here we apply our method (ISM, § 3) for shape classification in these variations:

(a) **ISM-CS:** with the cube-searching method from § 3.3 (1).
(b) **ISM-DC:** with the assumption that the shape's centre is known (see § 3.3 (2)).

The above versions of our method are compared against the following:

(i) **BOF-knn:** Encouraged by the good results of the 3D shape retrieval algorithm in § 4.2, we use this as one competitor. The test query shape is assigned to the most commonly occurring class of the best k-retrieved training shapes in a nearest-neighbor classification approach. Parameter k was learnt to optimize classification of train shapes. The shapes are represented by normalized tf-idf vectors and L_1 is used as metric.
(ii) **Toldo-BOF-SVM:** This is our implementation of Toldo *et al.* [11], where BOF vectors are computed on the training data $\mathcal{M}_T$. Then, the multi-class SVM classifier ([29]) is learned on the BOF vectors to predict the class label of the query shapes $\mathcal{M}_Q$. The kernel function is defined in terms of histogram intersection as in [11].

First, we investigate the sensitivity of classification methods with respect to the occlusions. Fig. 8 shows the performance of methods in the presence of occlusion on KUL dataset (§ 4.1 (i)). ISM-DC gives the best results for complete models and the performance of ISM-CS outperforms all methods with the more partial queries.

Table 1 summarizes all results on standard datasets of 3D shapes. Here, we measured the performance of classification methods on several datasets. Our approach using the Hough voting gave the average performance (see the last column in Table 1). The Princeton dataset (§ 4.1 (ii)) is the most challenging and although all methods gave similar results, we outperform the others. This dataset has very high variation amongst its 3D models *i.e.* the animal class contains widely varying models of 'ant' and 'fish'. For an SVM to learn a good classifier, we need a good non-linear kernel which has learnt such differences well. In such cases, non-parametric nearest-neighbor classifiers have a natural advantage.

The SHREC'09 dataset (§ 4.1 (iv)), previously used for the retrieval of partial queries, is now used for classification. ISM doesn't perform well as this method needs relatively large number of training examples [16,21] which is not satisfied in this case.

We conclude that our ISM based method beats k-nn and SVM in most cases.

4.4 3D Shape Classification of Reconstructed Real Life Scenes

As a final note, it is interesting to investigate the relative roles 2D and 3D object class detection could play in real-life. We carry out a small experiment to see whether 3D detection would really offer an added value.

Given many images taken in uncontrolled conditions around a real object, state-of-the-art methods such as the Arc3D web-service [30] can be used to

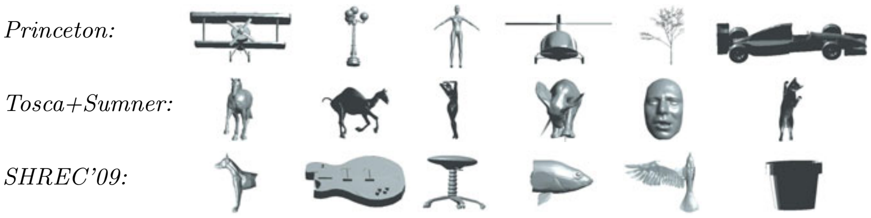


Fig. 9. Samples of query shapes from the state-of-the-art datasets

Table 1. Table summarizes all results of classification on state-of-the-art datasets. Proposed approach beats k-nn and SVM in most cases.

	Princeton			*Tosca+Sumner*			*SHREC'09*			
method	# TP	# FP	perfor.	# TP	# FP	perfor.	# TP	# FP	perfor.	avg. perf.
ISM	**529**	**378**	**58.3%**	**56**	**1**	**98%**	8	14	40%	**65.4%**
BOF-knn	491	416	54.1%	**56**	**1**	**98%**	7	13	35%	**62.4%**
BOF-SVM	472	435	52.0%	41	16	72%	**12**	**8**	**60%**	**61.3%**

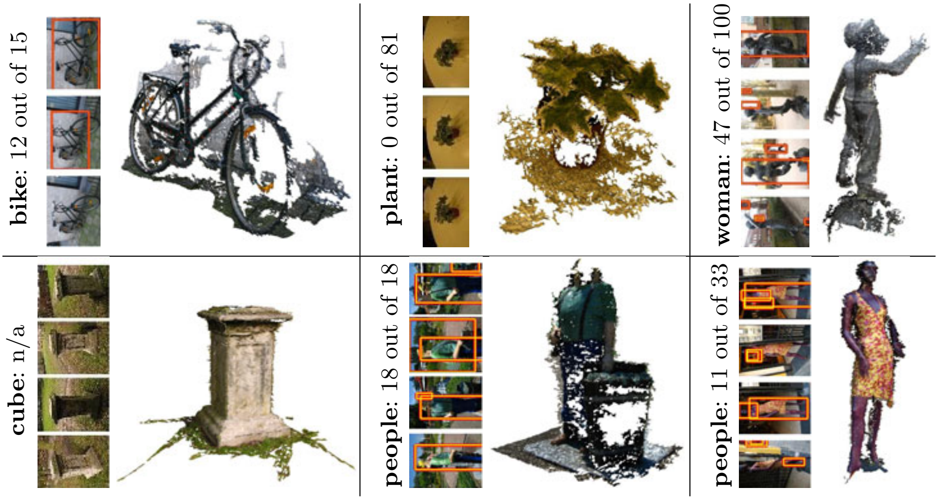


Fig. 10. 3D class recognition from the set of images. For each sample: correctly recognized class using 3D ISM, the number of correctly recognized objects in images using the method of Felzenszwalb *et al.* [31] (the best for PASCAL'08), samples of detection results are highlighted by squares, and the reconstructed shape by Arc3D [30].

extract a dense 3D model from the captured images. Such object models exhibit varying amounts of noise, holes and clutter from the surroundings, as can be seen from the examples (see Fig. 10). For each class on Fig. 10 we reuse the 3D ISM models trained on datasets of the SHREC'09 (for bike and plant classes), Tosca+Sumner (for woman) and KUL (for cube and people). We also used 2D Felzenszwalb detectors [31] trained on data from the PASCAL'08 datasets for bikes, potted plants, and pedestrians. As shown in the Fig. 10, a small test was run, where 3D reconstructions were produced from images for an instance of each of the 6 objects. In each of these cases, the classification using 3D ISM was successful, while SVM based method of Toldo *et al.* [11] failed in all cases. As to the 2D detectors, the bike was found in 12 out of the 15 images, the potted plant in none of the 81 images, and the person in 47 out of the hundred. This would indicate that given a video images input, a single 3D detection into the images could be more effective than 2D detections in separate images. But issues concerning 2D vs. 3D detection need to be explored further.

5 Conclusion

In this paper, we introduced 3D SURF features in combination with the probabilistic Hough voting framework for the purpose of 3D shape class recognition. This work reaffirms the direction taken by recent research in 2D class detection, but thereby deviates rather strongly from traditional 3D approaches, which are often based on global features, and where only recently some first investigations into local features combined with bag-of-features classification were made.

We have demonstrated through experiments, first the power of the features (§ 4.2), followed by the combined power of the features and the classification framework (§ 4.3). This method outperforms existing methods and both aspects seem to play a role in that.

Acknowledgment. We are grateful for financial support from EC Integrated Project 3D-Coform.

References

1. Kobbelt, L., Schrder, P., Kazhdan, M., Funkhouser, T., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3d shape descriptors (2003)
2. Saupe, D., Vranic, D.V.: 3d model retrieval with spherical harmonics and moments. In: Radig, B., Florczyk, S. (eds.) DAGM 2001. LNCS, vol. 2191, p. 392. Springer, Heidelberg (2001)
3. Osada, R., Funkhouser, T., Chazelle, B., Dobki, D.: Shape distributions. ACM Transactions on Graphics, 807–832 (2002)
4. Leymarie, F.F., Kimia, B.B.: The shock scaffold for representing 3d shape. In: Workshop on Visual Form (IWVF4) (2001)
5. Mian, A.S., Bennamoun, M., Owens, R.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE PAMI 28 (2006)
6. Sun, J., Ovsjanikov, M., Guibas, L.: A concise and provably informative multi-scale signature based on heat diffusion. In: SGP, pp. 1383–1392 (2009)
7. Gelfand, N., Mitra, N.J., Guibas, L.J., Pottmann, H.: Robust global registration. In: Symposium on Geometry Processing, pp. 197–206 (2005)
8. Pottmann, H., Wallner, J., Huang, Q.X., Yang, Y.L.: Integral invariants for robust geometry processing. Comput. Aided Geom. Des. 26, 37–60 (2009)

9. Novatnack, J., Nishino, K.: Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 440–453. Springer, Heidelberg (2008)
10. Ovsjanikov, M., Bronstein, A.M., Bronstein, M.M., Guibas, L.J.: Shapegoogle: a computer vision approach for invariant shape retrieval (2009)
11. Toldo, R., Castellani, U., Fusiello, A.: A bag of words approach for 3d object categorization. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2009. LNCS, vol. 5496, pp. 116–127. Springer, Heidelberg (2009)
12. Golovinskiy, A., Kim, V.G., Funkhouser, T.: Shape-based recognition of 3d point clouds in urban environments. In: ICCV (2009)
13. 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)
14. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. 110, 346–359 (2008)
15. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3d scenes. IEEE PAMI 21, 433–449 (1999)
16. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV 77, 259–289 (2008)
17. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 650–663. Springer, Heidelberg (2008)
18. Muja, M., Lowe, D.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISAPP (2009)
19. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)
20. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. IJCV (2010) (to appear)
21. Lehmann, A., Leibe, B., Gool, L.V.: Feature-centric efficient subwindow search. In: ICCV (2009)
22. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (2008)
23. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR, pp. 1038–1045 (2009)
24. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Shape Modeling International (2004)
25. Bronstein, A., Bronstein, M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Springer Publishing Company, Incorporated, Heidelberg (2008)
26. Sumner, R.W., Popovic, J.: Deformation transfer for triangle meshes. ACM Trans. Graph. 23, 399–405 (2004)
27. Dutagaci, H., Godil, A., Axenopoulos, A., Daras, P., Furuya, T., Ohbuchi, R.R.: Shrec 2009 - shape retrieval contest of partial 3d models (2009)
28. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24, 513–523 (1988)
29. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001) Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
30. Vergauwen, M., Gool, L.V.: Web-based 3d reconstruction service. Mach. Vision Appl. 17, 411–426 (2006)
31. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE PAMI 99 (2009)

Author Index