IADIS MULTI CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS

Prague, Czech Republic
22-26 July, 2013

Proceedings of the IADIS International conference Computer Graphics, Visualization, Computer Vision and Image Processing 2013

EDITED BY
Yingcai Xiao

iadis
international association for development of the information society

IADIS INTERNATIONAL CONFERENCE

COMPUTER GRAPHICS, VISUALIZATION, COMPUTER VISION AND IMAGE PROCESSING 2013

part of the

IADIS MULTI CONFERENCE ON COMPUTER SCIENCE AND

INFORMATION SYSTEMS 2013

PROCEEDINGS OF THE
IADIS INTERNATIONAL CONFERENCE
COMPUTER GRAPHICS, VISUALIZATION, COMPUTER VISION AND IMAGE PROCESSING 2013

Prague, Czech Republic
JULY 22 - 24, 2013

Organised by
IADIS
International Association for Development of the Information Society

Co-Organised by

FAKULTA PODNIKOHOSPODÁŘSKÁ
VŠE

Computer Graphics, Visualization, Computer Vision and Image Processing Volume Editor:
Yingcai Xiao

Computer Science and Information Systems Series Editors:
Piet Kommers and Pedro Isaías

Associate Editor: Luís Rodrigues

ISBN: 978-972-8939-89-2

TABLE OF CONTENTS

FULL PAPERS

SHORT PAPERS

REFLECTION PAPER

POSTERS

FOREWORD

These proceedings contain the papers of the IADIS International Conference Computer Graphics, Visualization, Computer Vision and Image Processing 2013, which was organised by the International Association for Development of the Information Society and co-organised by The University of Economics in Prague (VŠE), Czech Republic, 22 – 24 July, 2013. This conference is part of the IADIS Multi Conference on Computer Science and Information Systems 2013, 22 - 26 July, which had a total of 948 submissions.

The IADIS Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP) 2013 conference aims to address the research issues in the closely related areas of Computer Graphics, Visualization, Computer Vision and Image Processing. The conference encourages the interdisciplinary research and applications of these areas.

Submissions were accepted under the following 6 main topics:

- Computer Graphics
- Visualization
- Computer Vision
- Image Processing
- Other Related Topics

This event received 92 submissions from more than 19 countries. Each submission has been anonymously reviewed by an average of five independent reviewers, to ensure that accepted submissions were of a high standard. Consequently only 15 full papers were published. The overall acceptance rate corresponds to 16%. A few more papers were accepted as short papers, reflection paper and posters. An extended version of the best papers will be published in the IADIS International Journal on Computer Science and Information Systems (ISSN: 1646-3692) and/or in the IADIS International Journal on WWW/Internet (ISSN: 1645-7641) and also in other selected journals, including journals from Inderscience. Some of the best papers will be eligible to be extended and enhanced as book chapters for inclusion in a book to be published by IGI Global.

Besides the presentation of full papers, short papers, reflection paper and posters, the conference also included one keynote presentation from an internationally distinguished researcher. We would therefore like to express our gratitude to Professor Helwig Hauser, University of Bergen, Norway, for accepting our invitation as keynote speaker.

As we all know, organising this conference requires the effort of many individuals. We would like to thank all members of the Program Committees, for their hard work in reviewing and selecting the papers that appear in the proceedings.

This volume has taken shape as a result of the contributions from a number of individuals. We are grateful to all authors who have submitted their papers to enrich the conference proceedings. We wish to thank all members of the organizing committee, delegates, invitees and guests whose contribution and involvement are crucial for the success of the conference.

Last but not the least, we hope that everybody will have a good time in Prague, and we invite all participants for the next edition that will be held in Lisbon, Portugal.

Yingcai Xiao
The University of Akron
USA
Computer Graphics, Visualization, Computer Vision and Image Processing 2013
Program Chair

Piet Kommers, University of Twente, The Netherlands
Pedro Isaías, Universidade Aberta (Portuguese Open University), Portugal
Eva Kasparova, University of Economics, Faculty of Business Administration, Prague, Czech Republic
MCCSIS 2013 General Conference Co-Chairs

Prague, Czech Republic
July 2013

PROGRAM COMMITTEE

COMPUTER GRAPHICS, VISUALIZATION, COMPUTER VISION AND IMAGE PROCESSING 2013 PROGRAM CHAIR

Yingcai Xiao, The University of Akron, USA

MCCSIS GENERAL CONFERENCE CO-CHAIRS

Piet Kommers, University of Twente, The Netherlands
Pedro Isaías, Universidade Aberta (Portuguese Open University), Portugal
Eva Kasparova, University of Economics, Faculty of Business Administration, Prague, Czech Republic

COMPUTER GRAPHICS, VISUALIZATION, COMPUTER VISION AND IMAGE PROCESSING 2013 COMMITTEE MEMBERS

Adrian Jarabo, University Of Zaragoza,, Spain
Aiert Amundarain, Ceit, Spain
Alberto Raposo, Puc-rio, Brazil
Alessandro Artusi, University of Girona, Spain
Alessandro Rizzi, Università Degli Studi Di Milano, Italy
Alexander Pasko, Bournemouth University, United Kingdom
Anderson Maciel, UFRGS, Brazil
Andre Hinkenjann, Bonn-rhein-sieg University Of Applied Sciences, Germany
Andreas Gerndt, German Aerospace Center (DLR), Germany
Andreas Kerren, Linnaeus University, Sweden
Angelica De Antonio, Universidad Politecnica De Madrid, Spain
Antonio Diaz Estrella, Malaga University, Spain
Arcadio Reyes Lecuona, Universidad De Málaga, Spain
Arturo S. Garcia, University of Castilla-la Mancha, Spain
Beatriz Rey, Universidad Politecnica De Valencia, Spain
Belen Masia, Universidad de Zaragoza, Spain
Brian dAuriol, Kyung Hee University, Korea, Republic Of
Bruce Campbell, Rhode Island School Of Design, USA
C. C. Lu, Kent State University, Usa
Carina Gonzalez, University Of La Laguna, Spain
Carla Binucci, Università Degli Studi Di Perugia, Italy
Carlo Nati, Working Group for development of Scientific and Te, Italy
Carlos Buchart, CEIT, Spain

Carlos Gonzalez-morcillo, University Of Castilla-la Mancha, Spain
Cesar Alberto Collazos, Universidad Del Cauca, Colombia
Chang Ha Lee, Chung-ang University, Korea, Republic Of
Charalampos Georgiadis, The Aristotle University, Greece
Chih-Cheng Hung, Southern Polytechnic State University,, USA
Christos Gatzidis, Bournemouth University, United Kingdom
Christos Grecos, University Of The West Of Scotland, United Kingdom
Creto Vidal, Federal University Of Ceará, Brazil
Cristian Bonanomi, Universita' Degli Studi Di Milano, Italy
Dalton Lin, National Taipei University, Taiwan
Daniel Steffen, German Research Center For Artificial Intelligence, Germany
Daniel Thalmann, Nanyang Technological University, Singapore
Dariusz Frejlichowski, West Pomeranian University Of Technology, Poland
Davide Gadia, Università Degli Studi Di Milano, Italy
De-Yuan Huang, National Central University, Taiwan
Dongrong Xu, Columbia University, Usa
Elisabeta Marai, University Of Pittsburgh, Usa
Fadi Dornaika, Upv-ehu, Spain
Faisal Qureshi, University Of Ontario Institute Of Technology, Canada
Fatima Nunes, University of São Paulo, Brazil
Fernando Lopez, Polytechnic University Of Valencia, Spain
Flavio Prieto, Universidad Nacional de Colombia, Colombia
Fotis Liarokapis, Coventry University, United Kingdom
Francisco Gonzalez Garcia, University Of Girona, Spain
Francisco Luis Gutierrez, University Of Granada, Spain
Franck Vidal, Bangor University, United Kingdom
Frank Michel, German Research Center For Artificial Intelligence, Germany
Galina Pasko, Uformia, Norway
Georgios Sakas, TU Darmstadt, Germany
Gilles Gesquiere, Liris, France
Giovanni Farinella, University of Catania, Italy
Giovanni Gallo, Università Di Catania,, Italy
Giovanni Puglisi, University Of Catania , Italy
Giuseppe Liotta, University Of Perugia, Italy
Giuseppe Patanè, CNR-IMATI, Italy
Gustavo Patow, Universitat De Girona, Spain
Hans-Jörg Schulz, University Of Rostock, Germany
Harald Obermaier, University Of California, United States
Helmuth Trefftz, Eafit University, Colombia
Henning Barthel, Fraunhofer Institute For Experimental Software Eng, Germany
Hongchuan Yu, Bournemouth University, United Kingdom
Hugo Alvarez, Ceit, Spain
Igor Sevastianov, Nvidia Corp, USA
Ingemar Ragnemalm, Linköping University, Sweden
Isaac Rudomin, Barcelona Supercomputing Center, Barcelona
Jairo Sanchez, Vicomtech-ik4, Spain
Jian Chang, Bournemouth University, United Kingdom

Jian Zhang, Bournemouth University, United Kingdom
Jinyuan Jia, Tongji University, China
José Izkara, TECNALIA, Spain
Jose Ignacio Echevarria, Universidad de Zaragoza, Spain
José Pascual Molina Massó, Universidad de Castilla-la Mancha, Spain
Jose Remo Ferreira Brega, Sao Paulo State University, Brazil
Josip Almasi, Vrspace, Croatia
Juan Manuel Gonzalez Calleros, University Of Puebla, Mexico
Jyoti Singhai, Maulana Azad National Institute Of Technology Bhop, India
Karolj Skala, Rudjer Boskovic Institute, Croatia
Krzysztof Okarma, West Pomeranian University Of Technology, Poland
Krzysztof Walczak, Poznan University of Economics, Poland
Kurt Debattista, The University Of Warwick, United Kingdom
Kyoungsu Oh, Soongsil University, Korea, Republic Of
Laura Monroe, Los Alamos National Lab, Usa
Laura Papaleo, Province of Genova (Government Administration), Italy
Leandro Fernandes, Federal Fluminense University, Brazil
Lihua You, Bournemouth University, United Kingdom
Linda Giannini, Teacher, eTwinning Ambassador , Italy
Luca Grilli, Universita Degli Studi Di Perugia, Italy
Luciano Soares, Pontifical Catholic University Of Rio De Janeiro, Brazil
Luis Unzueta, Vicomtech-ik4, Spain
M. Emre Celebi, Louisiana State University in Shreveport, USA
Marc Daniel, Ecole Superieure D ingenieurs De Luminy, France
Marcelo Guimaraes, UNIFESP / FACCAMP, Brazil
Marcos Nieto, Vicomtech-ik4, Spain
Mario Hlawitschka, University Of Leipzig, Germany
Marla Schweppe, Rochester Institute Of Technology, USA
Marta Fairen, Universitat Politècnica De Catalunya, Spain
Matteo Cantamesse, Catholic University Of Milan, Italy
Mauricio Hess-Flores, University of California, Davis, USA
Miguel Angel, I3BH/Labhuman, Spain
Mubbasir Kapadia, University Of Pennsylvania, USA
Mu-Chun Su, National Central University, Taiwan
Nadia Ambrosetti, Universita Degli Studi Di Milano, Italy
Natalia Padilla, Etsi Informática Y Telecomunicaciones, Spain
Nathan Carr, Adobe Systems Inc., USA
Paolo Pingi, CNR - ISTI, Italy
Pascual González López, University of Castilla-La Mancha, Spain
Patricia Crossno, Sandia National Laboratories, Usa
Pawel Forczmanski, West Pomeranian University Of Technology, Poland
Pedro Cano, Universidad de Granada, Spain
Pere-Pau Vázquez, Upc, Spain
Peter Dannenmann, RheinMain University of Applied Sciences, Germany
Petros Faloutsos, York University, Canada
Pierre-frederic Villard, University Of Lorraine, France
Quan Wen, University Of Electronic Science And Technology Of, China

KEYNOTE LECTURE

INTEGRATING INTERACTIVE AND COMPUTATIONAL ANALYSIS IN VISUAL ANALYTICS

By Professor Helwig Hauser,
University of Bergen, Norway
www.ii.UiB.no/vis

ABSTRACT

In our emerging information age it becomes increasingly important that we can exploit the wealth of available data for the sake of learning, decision making, as well as for other tasks. A promising approach – not at the least targeted by the new concept of *visual analytics* in visualization research – is to cleverly integrate the strengths of computers (fast computation, efficient handling of large datasets, comparably low costs, etc.) with the strengths of the users (outstanding perceptual and cognitive capabilities, domain knowledge, etc.). In this talk, we look at one possible solution, originating in visualization research within computer science, i.e., the concept of *interactive visual analysis*, and describe it as an iterative process, enabling the integration of computational and interactive means for data exploration and analysis. Thinking of interactive visual analysis as an iterative process enables that each step is performed on the basis of a toolbox with computational and interactive visual solutions. In order to substantiate the conceptual aspects of this solution, we also look at several examples that document the successful application of interactive visual analysis.

Full Papers

HUMAN DETECTION BY USING CENTRIST FEATURES FOR THERMAL IMAGES

Irfan Raiz, Jingchun Piao and Hyunchul Shin
Hanyang University - 55 Hanyangdaehak-ro, Sangnok-gu, Ansan Kyeonggi-do, 426-791, Korea

ABSTRACT

In this paper, we present a new human detection scheme for thermal images by using CENsus TRansform hISTogram (CENTRIST) features and Support Vector Machines (SVMs). Human detection in a thermal image is a difficult task due to low image resolution, thermal noising, lack of color, and poor texture information. For thermal images, contour is one of the most useful and discriminative information, so capturing it efficiently is important. Histogram of Oriented Gradient (HOG) is still the most proven way to capture the human contour. CENTRIST is a computationally efficient technique to capture contour cues as compared to HOG, but so far no one has implemented and tested the accuracy of CENTRIST descriptor for infrared thermal images. We developed CENTRIST based human detection system for thermal images. We also made a new dataset of thermal images, since there was no realistic dataset. Experimental results show that CENTRIST carries out almost same detection rate as HOG, while reducing the training and the testing time by almost 91 %.

KEYWORDS

Human detection, vision, CENTRIST, HOG, thermal image

1. INTRODUCTION

Human detection is of fundamental importance in computer vision due to its various applications that intersect with many aspects of human life. Computer vision based systems are becoming more feasible and affordable with recent advancement of technology. In near future, vision based systems will become an essential part of our lives e.g. driver assistance systems for smart cars, video surveillance, security, and robotics. Developing a reliable and robust human detection system is one of the most challenging tasks with lots of potential applications.

Pedestrian safety is an issue of global significance. In post-industrial countries, pedestrian fatalities are inevitable. The traffic fatality rate at night is about three times higher than that in day-time. In night-time, the poor lighting condition affects the driver a lot. There is an essential need for improvement of visibility under poor lighting and weather conditions e.g. at night, bad weather, under fog, and so on. To develop a splendid human detector under different lighting conditions and various weather conditions, thermal imaging can be one of solutions for relieving the problems of visible imaging techniques. Object detection in thermal domain has attracted more interests, as the infrared cameras become more affordable.

Only limited visual information can be captured by CCD cameras under poor lighting and weather conditions. Meanwhile, the brightness intensities of thermal images are representatives of the temperatures of object surface points. Pedestrians typically emit more heat than background objects, such as trees, road, etc. Image regions containing pedestrians or other “hot” objects will be brighter than the background. Hence theoretically, infrared thermal images can be a reliable source for human detection in night-time and bad weather conditions. Thermal images, compared to visible images, lack several features, such as color and texture information, which plays a vital role for human detection and classification. The most discriminative and distinctive features of human beings from the background lie in their contours.

In this paper, we implement CENTRIST [1] feature based human detection method for thermal images. The goal is to make an appropriate feature descriptor which will be a part of our efficient human detection system. From our dataset, we extract HOG and CENTRIST features and use them to train two separate linear

Support Vector Machines (SVMs). Based on that, a comparative analysis of both methodologies is carried out.

This paper makes four major contributions.

1. We implemented CENTRIST feature extraction on pedestrian thermal image dataset, and trained a linear SVM classifier to do per window based binary classification.
2. We made our own thermal image dataset as there were no reliable thermal image datasets for on road pedestrians.
3. We compared the computational efficiency and detection accuracy of CENTRIST and HOG on our pedestrian thermal image dataset.
4. We found that thermal images need appropriate pre-processing for good performance and that the CENTRIST based method can show an accuracy similar to HOG but in a lot more efficient manner.

2. RELATED WORKS

Bertozzi [2] mentioned a human detection method as a part of the Advanced Driver Assistance System (ADAS). Various feature descriptors have been applied to human detection. However HOG is probably the most popular feature descriptor in human detection [3], [4], [5], [6], [7]. Recently the Local Binary Pattern (LBP) also shows high potential [7], [8]. A new trend in pedestrian detection is to combine multiple sources, e.g., color, local texture, motion, etc. [6], [7], [9], [10]. Wu and Nevatia [11] proposed a detection method that detects body parts by a combination of edgelet features and combines the responses of the part detectors to compute the likelihood of the presence of a person.

Kai and Arens [12] proposed a local-feature based human detector on thermal dataset. In the training phase, they used Speed Up Robust Features (SURF) [13]. Then a codebook is created by clustering these features and building Implicit Shape Model (ISM) to describe the spatial configuration of features relative to the object center. Wu et al. [1], [14] believe that contour is the most useful and discriminative information for pedestrian detection. Therefore they designed the CETRIST descriptor to detect human contour. Using the CENTRIST descriptor, they proposed the C4 algorithm that can accurately detect pedestrians. The phases of the method are suitable for parallel processing.

In terms of classifiers linear SVM is widely used for its fast testing speed. Recently SVM has been used in many application domains. It provides a supervised learning approach for object recognition such as faces [15], [16], face components [15], and pedestrians [17].

3. THERMAL IMAGE BASED HUMAN DETECTION

In this section we present our dataset and then explain CENTRIST features and how we have implemented them. For the sake of completeness, a brief introduction to HOG features is presented. Training and testing methodologies are also discussed in detail.

3.1 Thermal Pedestrian Dataset

For making our dataset, we collected approximately 5 hours of 25fps night-time video scenario along the road which contains various types of pedestrians, including along-street, across-street, and bicyclists. The ambient temperature at the time of recording was about 21^{o}C. The NEC-C200 infrared thermal camera with 320*240 pixel image resolution was used for recording. After recording, the video was split into 5 frames per second and was up-sampled to 640x480 using cubic kernel (the output pixel value is a weighted average of pixels in the nearest 4x4 neighborhood). From the extracted frames, we cropped 2400 pedestrians as positive images; number of positive images were doubled i.e. 4800, by flipping the image about vertical axis. Total 6000 negative images were also extracted, out of which 2000 were high mean, 2000 were high variance and 2000 were randomly selected. Based on careful observation of the pedestrian size and aspect ratio in the data set, we chose image window size of 54x108 for our experiment as shown in Fig 1.

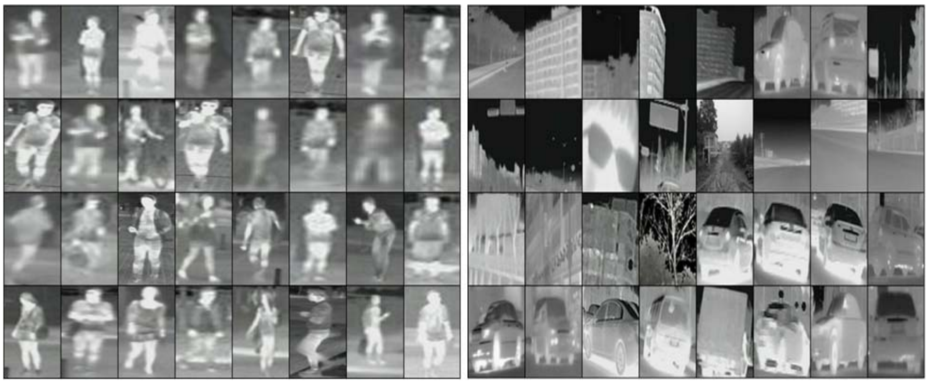


Figure 1. Pedestrian and non-pedestrian samples in the thermal image data set

3.2 CENTRIST Features

Wu et al.[1] presented a fast and efficient method of detecting humans by emphasizing on the human contour using a cascade classifier and the CENTRIST visual descriptor. The author claims that CENTRIST is particularly suitable for human detection, as it concisely encodes the sign information, and is able to capture large scale structures or contours. Also it detects humans at 20 fps speed on 640x480 resolution using only one processing thread and achieves accuracies comparable to the state-of-the-art.

To compute CENTRIST visual descriptor for an input image *I*, we perform histogram equalization, Sobel edge detection, and Census Transform to generate *I′*. In [1] authors suggested that, for visible images, preprocessing is not required. During our experiments, we have found that, for IR images, a preprocessing technique e.g. histogram equalization enhances the contour information of objects in a scene. Applying an edge detector after preprocessing the input images will result in edges/contours that are accurate and well connected.

A toy example for computing Census transform [18] is shown in Fig 2. For the pixel under consideration, in the neighboring 8 pixels, census transform finds pixel intensities that are greater than the intensity value of pixel under consideration and replace them with ‘0’; otherwise it replaces them by ‘1’. The CT value is computed by collecting bits from left to right and top to bottom and then converting them to base-10 value. CENTRIST descriptor, for the image I, is the concatenation of the histogram of CT values on the overlapping blocks composed of neighboring cells, see Fig 3.

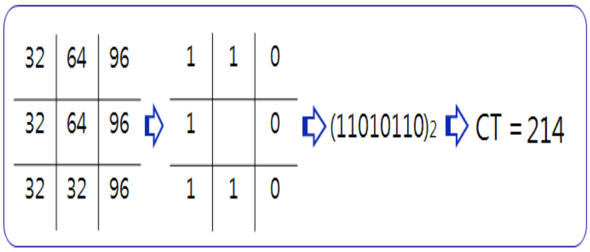


Figure 2. Census Transform

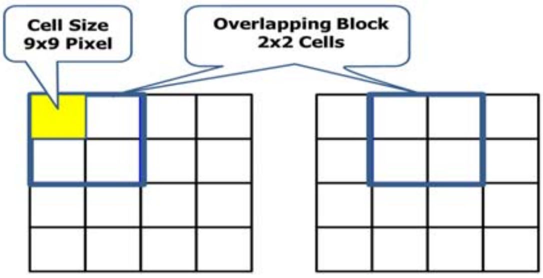


Figure 3. Cells and overlapping blocks.

The parameters that we have used for implementation of CENTRIST and HOG descriptor are presented in table 1. We currently use 54x108 pixels as the detection window size, 9x9 pixels as the cell size, and 2x2 cells as block size. We take any adjacent 2x2 cells as a block and extract a CENTRIST descriptor from each block. The overlap is kept 50% so there are 11x5 = 55 blocks, thus the feature vector for a candidate image patch has 11x5x256 = 14080 dimensions. We don’t compute Census Transform for the border pixels of the detection window as the Census Transform requires at least 3x3 region hence borders are excluded while computing the CENTRIST descriptor.

3.3 HOG Features

Histogram of Oriented Gradients (HOG) is a popular feature descriptor for object classification, especially for human detection. The working philosophy behind HOG is local object appearance and shape can often be characterized rather well by the distribution of local intensity gradients or edge directions [3].

It basically counts occurrences (histogram) of gradient orientations in localized cells of an image, see Fig 4. Gradient G_x and G_y is computed by applying [-1, 0, 1] and $[-1, 0, 1]^T$ in horizontal and vertical directions of image. Using this information gradient magnitude and orientation is calculated. Gradient information is collected from local cells into histograms using tri-linear interpolation. On the overlapping blocks composed of neighboring cells, as shown in Fig 3, normalization is performed. Extracted HOG features are robust to changes in lighting conditions and small variations in pose, thanks to the process of interpolation, local normalization, and histogram binning [3]. Table 1 shows the parameter values that we have used for feature extraction with HOG.

Table 1. Parameter Settings

Parameters	CENTRIST	HOG
Window Size	108x54 pixels	108x54 pixels
Preprocessing	Histogram Equalization	Gamma Normalization
Cell Size	9x9 pixels	9x9 pixels
Block Size	2x2 cells	2x2 cells
Number of Bins	256	9
Orientation Range	0^o-360^o	0^o-180^o
Overlap	50%	50%
Block Normalization	Nil	L2-Hys Norm
Blocks per Window	11x5	11x5
Feature Vector Length	14080	1980
Data Type	Uint8	Double
SVM	Linear	Linear

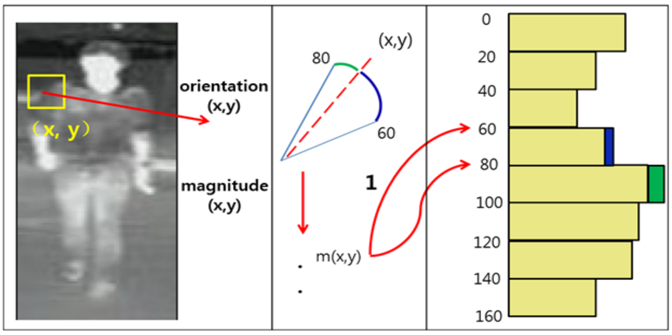


Figure 4. Histogram of Oriented Gradients.

3.4 SVM Classifier

From a set of labeled training images, we extract HOG and CENTRIST features and use them to train two separate linear SVM's. We have used the MATLAB's svmtrain and svmclassify functions with their default settings for training and binary classification of testing data respectively. Details regarding training and testing methodology are presented in the following sections.

3.5 Training Methodology

For both feature vector descriptors, we have employed a similar training methodology as used in [3], shown in Fig 5. For initial training of SVM, we have used 2800 positive and 3000 negative sample windows of resolution 54x108. Our dataset consists of total 17,000 far infrared (FIR) images of up-sampled resolution

640x480, out of which 1500 are negative training images that are resampled to create hard examples for retraining of linear SVM. Retraining of SVM with hard examples reduces of false positive rate by 10%. Table 2 shows details of our dataset used for training and testing. Training methodology consists of the following steps.

1. Take initial positive and negative window examples from training dataset and generate label vector.
2. Generate a feature vector set by encoding all positive and negative windows with the selected feature vector descriptor.
3. Generate a linear SVM model, using feature vector set and label vector.
4. Using selected the descriptor and the SVM model, search 1500 negative training images exhaustively for false positives ('hard examples').
5. Augment the initial training data with collected hard examples and retain the SVM.

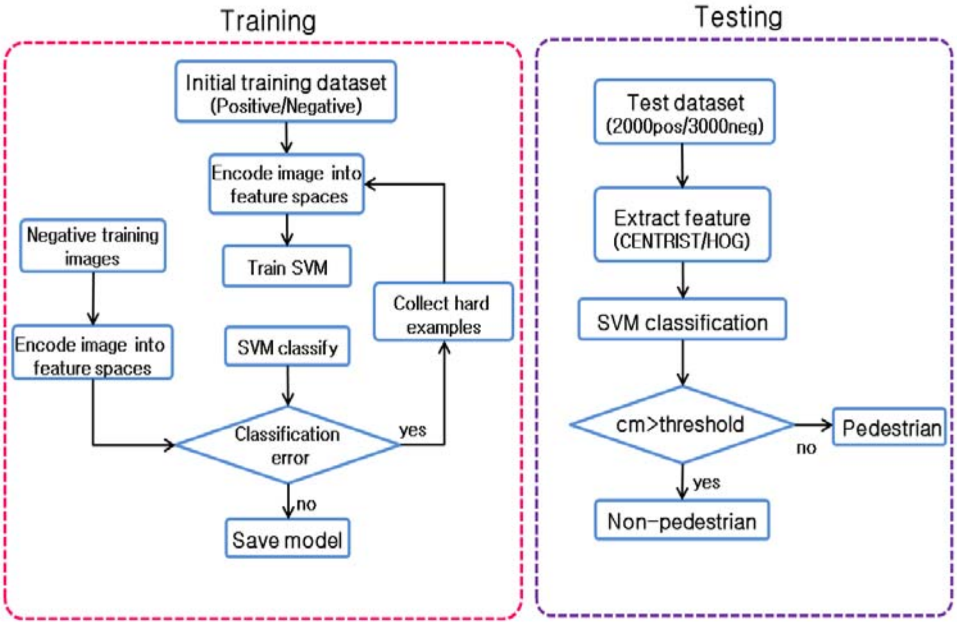


Figure 5. Training and Testing Methodology

Table 2. FIR Data Set

Parameters	
IR Dataset Resolution	640x480 pixels
Negative Training Images	1500
Total images	17000
Sample window size	54x108 pixels
Initial Training	
Positive sample windows	2800
Negative sample windows	3000
Testing	
Positive sample windows	2000
Negative sample windows	3000

3.6 Testing Methodology

A pedestrian can be detected in a scene by using brute force searching and testing of scale space in FIR camera based pedestrian detection systems. For example, all sliding window based models involve feature extraction, dense multi-scale scanning of detection windows, and binary classification, followed by non-maximum suppression [19]. The other way can be to use some simple tests to generate possible candidate locations and then verify them by using more sophisticated methods [20], [21].

For detecting pedestrians in a scene, depending upon the technique used, the number and the values of parameters (other than descriptor and classifier parameters) are quite diverse and their erroneous selection can make even a well-trained state of the art detector perform badly. Therefore, for an accurate evaluation of a descriptor, we strip it down to its basic functionality, as shown in Fig 5, for a given fixed-sized normalized window, the descriptor will extract important features which are fed to a pre-trained binary classifier. The classification accuracy is the measure of discriminative power of the descriptor under question.

4. EXPERIMENTAL RESULTS

For testing both detectors we have adopt per-window evaluation methodology. We measure the per-window (PW) performance on cropped positive and negative image windows based on equally trained binary linear SVM classifiers and then we plot the Receiver Operating Characteristic (ROC) curve.

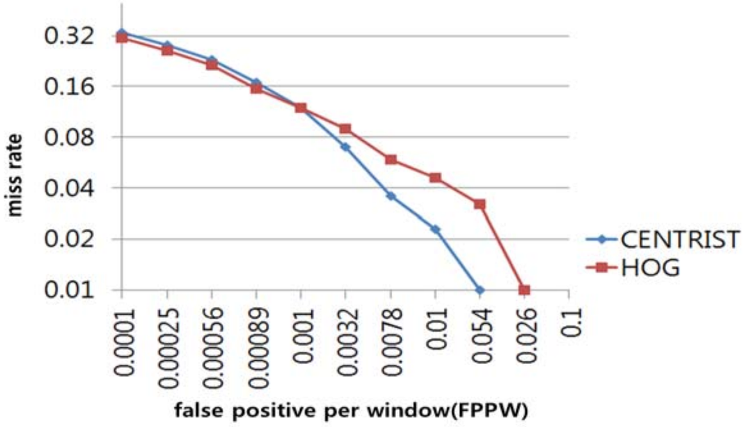


Figure 6. Comparison of Centrist and HOG pedestrian detectors on FIR dataset

Fig. 6 shows the ROC curves for HOG feature verses CENTRIST feature trained linear SVMs. For the false positive rates less than 10^{-3}, HOG seems to perform slightly better than CENTRIST. However, CENTRIST seems to perform a little better than HOG after 10^{-3}. As as far as detection accuracy is concerned, they both seem to be almost equal.

Now we present the comparison of computational complexities of both algorithms in terms of training and testing time. The reported training time only takes into account the time taken by the initial training and It does not include SVM retraining phase. From training and testing times it can be seen that CENTRIST reduces the testing and training CPU time by almost 91%, with the approximately same detection accuracy. As CENTRIST captures the contour information more efficiently than HOG, we can add other complementary features, such as self-similarity and motion, to get better detection accuracy at lower complexity when compared to HOG based method.

The experiments were carried out on a single core Intel i5-2400 3.1 GHz CPU with 8GB memory. The coding was done in MATLAB.

Table 3. Comparison of CPU time

Time	CENTRIST	HOG
Initial Training	126sec	1407sec
Testing timing	41sec	463sec

5. CONCLUSIONS

In this paper, we have implemented CENTRIST based visual descriptor for pedestrian detection for thermal images. We have created pedestrian thermal image dataset, and trained a linear SVM classifier using CENTRIST features. An experimental comparison with HOG feature descriptor was carried out regarding the human detection rate and computation time. The comparative analysis shows that CENTRIST exhibits almost equivalent detection accuracy as HOG, but it reduces the training and the testing time by 91% on average, resulting more than 10 times speed up. Hence for pedestrian detection in thermal images, CENTRIST seems to be better choice than HOG in detection accuracy and computational complexity. Furthermore, we can improve the recognition rate significantly by combining complementary information sources, e.g. motion and self-similarity with CENTRIST.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Science, ICT&Future Planning and IDEC Platform center (IPC) in Korea.

REFERENCES

[1] Wu, J. et al, 2011. Real-Time Human Detection Using Contour Cues. *In Proc. ICRA,* Shanghai, China, pp. 860-867.

[2] Bertozzi, M., 2003. Pedestrian detection in infrared images. *In Proc. IEEE Intelligent Vehicles Symp.,* Columbus, OH, pp. 662-667

[3] Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. *In CVPR,* San Diego, CA, USA, pp. 886–893.

[4] Felzenszwalb, P.F. et al, 2008. A discriminatively trained, multiscale, deformable part model. *In CVPR,* Anchorage,Alaska. USA, pp. 1-8.

[5] Maji, S. and Berg, A.C., 2009. Max-margin additive classifiers for detection. *In ICCV.* Kyoto, Japan, pp. 40-47.

[6] Schwartz, W.R. et al, 2009. Human detection using partial least squares analysis. *In ICCV,* Kyoto, Japan, pp. 24-31.

[7] Wang, X. et al, 2009, An HOG-LBP human detector with partial occlusion handling. *In ICCV*, Kyoto, Japan, pp. 32-39

[8] Mu, Y. et al, 2008. Discriminative local binary patterns for human detection in personal album. *In CVPR,* Anchorage, Alaska. USA, pp. 1-8.

[9] Doll´ar, P. et al, 2009. Integral channel features. *In BMVC,* London, England, pp. 1-11.

[10] Leibe, B. et al, 2005. Pedestrian detection in crowded scenes. *In CVPR,* San Diego, CA, USA, pp. 878–885.

[11] Wu, B. and Nevatia, R., 2007. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision,* vol. 75, No. 2, pp. 247–266.

[12] Arens, M. and Jungling, K., 2009. Feature based person detection beyond the visible spectrum. *In IEEE CVPR Workshops*, pp. 30-37.

[13] Tuytelaars, T. et al, 2006. Surf: Speeded up robust features. *In Proc. 9th European Conference on Computer Vision,* Graz, Austria, pp. 404-417.

[14] Wu, J. and Rehg, J.M., 2011. CENTRIST: A visual descriptor for scene categorization. *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 33, pp. 1489-1501.

[15] Heisele, B. et al, 2001. Face recognition with support vector machines: Global versus component-based approach. *In ICCV,* Vancouver, BC, Canada, pp. 688-694.

[16] Osuna, E. et al, 1997. Training support vector machines: An application to face detection. *In CVPR,* San Juan, Puerto Rico, pp. 130–136.

[17] Mohan, A. and Poggio, T., 2001. Example-based object detection in images by components. *IEEE Trans.Pattern Anal.Machine Intell.,* vol. 23, pp. 349–361.

[18] Zabih, R. and Woodfill, J., “Non-parametric local transforms for computing visual correspondence,” in ECCV, *Stockholm, Sweden* ,vol. 2,1994, pp. 151–158.

[19] Dollar, P. et al, 2012. Pedestrian Detection: An Evaluation of the State of the Art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol.34, no.4, pp.743-761.

[20] Walk, S. et al, 2010. New features and insights for pedestrian detection. *In CVPR,* San Francisco, USA, pp.1030-1037.

[21] Miron, A. et al, 2012. Intensity self similarity features for pedestrian detection in Far-Infrared images. *Intelligent Vehicles Symposium,* pp.1120-1125.

REFINING EYE PUPIL BORDER BY CIRCULAR SHORTEST PATH METHOD

Ivan Matveev[1] and Ivan Simonenko[2]
[1]*Computing Centre of Russian Academy of Sciences - Vavilov str., 40, Moscow, 119333, Russia*
[2]*Moscow Institute of Physics and Technology - Institutskii per., 9, Dolgoprudny, Moscow Region, 141700, Russia*

ABSTRACT

The problem of detecting precise pupil border in eye image given its initial circular approximation is addressed with circular shortest path method. Brightness gradient direction is employed to choose image pixels, which may belong to pupil boundary. Using initial approximate circles allows the method to work in a narrow ring, which contains only single pupil contour. Under these conditions the method allows to correctly handle almost all images used for iris recognition tasks and appears to be more precise than human expert in marking the pupil border. The method was tested with public domain iris databases, containing more than 80000 images totally. Experiments show that refinement of pupil border increases precision of iris recognition.

KEYWORDS

Iris segmentation, image processing, biometrics

1. INTRODUCTION

Iris recognition is one of main biometric identification technologies. Detecting iris borders in image is an important part of this method. Iris pattern in image is represented as a ring enclosed between two approximately circular and approximately concentric contours: inner border, which is an iris-pupil boundary, and outer border, which is iris-sclera boundary. Both boundaries are approximated by circles with good precision, however there are applications demanding more precise shape detection and description, as in ISO standard (2005). This particularly concerns inner (i.e. pupil) boundary. As a rule, human pupil is close to circle in shape, however in most cases it is not an ideal circle and has irregular deviations with relative magnitude around 5-10%, see Kansky (2003). Thus, a problem appears to detect a contour, which has approximately circular shape and encloses dark region (pupil) in relatively brighter background. Apparently, iris can be detected as a dark circular region in a background of sclera, in case there are no or small occlusions by eyelashes and eyelids. Problem of detecting shapes modeled by circles and ellipses (i.e. regular shapes) have attracted much attention and many methods are developed. Rich variability of methods applied to determining iris boundaries is reviewed by Bowyer, Hollingsworth and Flynn (2008, 2012). Much less attention is attracted to tracking the boundaries to their irregular (although close to circle) shape that may allow better recognition performance. Many researchers limit themselves to just an iteration of same detection algorithm like Heet al. (2009), Maenpaa (2005) or to applying same detection method in different scale under multi-scale image processing scheme, see Nabti, Ghouti and Bouridane (2008) and Pan, Xie and Ma (2008). Special methods for refinement of the boundaries, which track roundish but irregular shapes with good precision and tolerate noise were developed to much less extent. The authors could find that only one approach of active contours was specially targeted and tested with the task of iris border refinement, presented in the works of Daugman (2007), Ross and Shah (2006), Koh, Govindaraju and Chaudhary (2010). Here a method of circular shortest path proposed by Sun and Pallottino (2003) is modified and applied to the problem of refining pupil and iris borders.

2. CIRCULAR SHORTEST PATH ALGORITHM

There are plenty of methods performing detection of shortest path in images. Specific feature of the CSP approach among them is that it starts from an a priory detected point that is claimed to be an approximate contour center (or at least is lying inside the contour). In the refinement task stated here initial data for the method are even more detailed: center position and radius of approximating circle are given. Since the contour passes round a given point it is reasonable to perform a polar transformation with the pole in this point, which simplifies both representation and calculations. (Indeed, polar transformation is a very common issue in iris image processing.) Polar transformation renders a ring shape to a rectangle. This rectangle can be positioned so as to have its top side being a circle enclosing the proposed contour (enough big circle should be taken), and bottom side is to be enough small circle inside the contour. Then left side and right side both correspond to a coordinate origin line, let it be OX half-axis. The radial coordinate of polar system is transformed to abscissa of the rectangle and angle coordinate becomes ordinate. Image in OXY domain is transformed to $O\rho\varphi$ domain and is also represented as a rectangular raster. Define the size of the raster as $\mathrm{W} * \mathrm{H}$ pixels. Call it *polar representation rectangle*, see Figure 1.

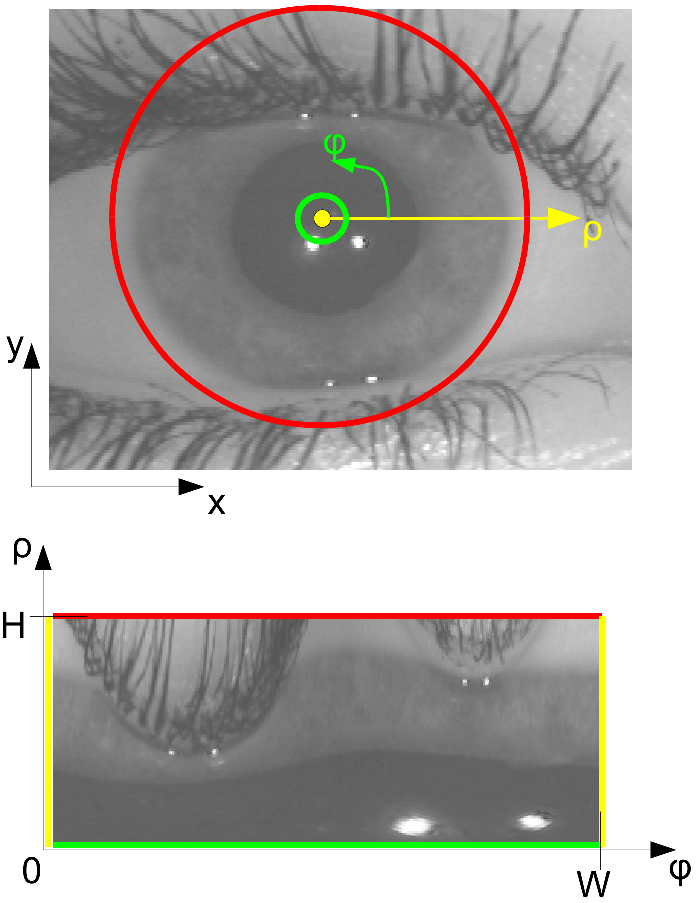


Figure 1. Sample of polar transform of iris

After the polar transformation circular path location task is rendered to a problem of detecting optimal path between left and right sides of rectangle, under the condition that terminal points at the sides have same vertical coordinate. Since contour is close to circle in shape and pole of the polar transform is inside it, the polar representation of contour is univalent, i.e. only one radius of contour corresponds to each angle value, and the contour can be expressed in terms of function $\rho(\varphi)$, $\varphi \in [0;2\pi]$, $\rho(0) = \rho(2\pi)$. Further, assuming that pole of transformation is not very close to the contour line (in other words, is near the center of the contour), one can state that derivative of radius by angle is limited: $d\rho/d\varphi \le 1$. In raster polar representation rectangle of size $\mathrm{W} * \mathrm{H}$ this function becomes a discrete sequence $\{\rho_n\}$ (or accounting for both coordinates $\{(n, \rho_n)\}$), $n \in [0; W-1]$, $\rho \in [0; H-1]$. Limitation to derivative becomes $|\rho_{n+1} - \rho_n| \le 1$. Condition to equal value at ends becomes $|\rho_{W-1} - \rho_0| \le 1$.Thus the contour is represented as a chain of points in rectangular raster, each column of the raster contains one and only one contour point, the points in adjacent columns belong to same or adjacent rows, the first and last points also belong to same or adjacent rows. Hereinafter this chain of points is called *path* $S = \{\rho_n\}_{n=0}^{W-1}$. Figure 2 illustrates the possible paths of a contour, if tracked from left to right. A path from point with coordinates $(\varphi; \rho) = (2;3)$ can go to points $(3;2)-(3;4)$, from point $(5;1)$ - into points $(6;1)$ and $(6;2)$, and if staring point is $(1;2)$ ending points can be $(8;1)-(8;3)$.

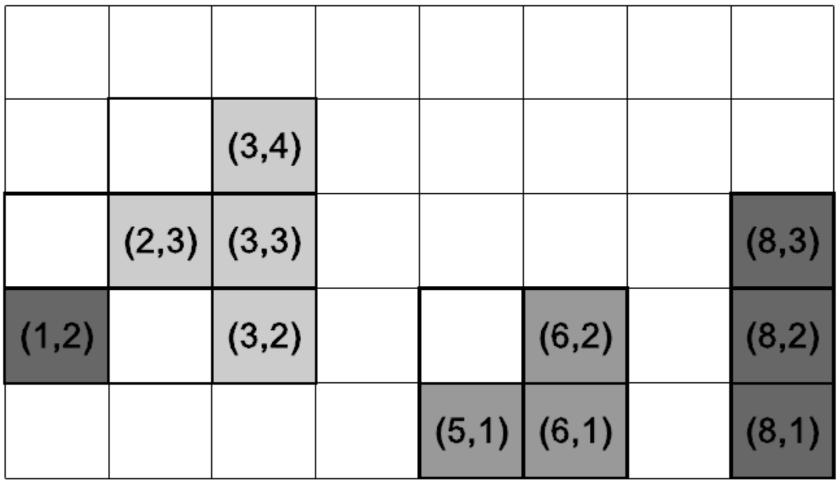


Figure 2. Allowed transitions between points in case of limited derivative

Introduce the cost of transition between points (n, ρ') and (n, ρ'') in adjacent columns in polar representation. Define it as $C((n, \rho'), (n, \rho''))$, or shorter $C_n(\rho', \rho'')$. It is composed from "inner" and "outer" parts: $C(\rho', \rho'') = C^{(O)}(\rho', \rho'') + C^{(I)}(\rho', \rho'')$. Inner part conditions the shape of the contour and favours straight horizontal lines (which are circles in original OXY domain):

$$C_n^{(I)}(\rho', \rho'') = \begin{cases} 0, \ \rho' = \rho'' \\ T_1, \ |\rho' - \rho''| = 1 \\ \infty, \ otherwise \end{cases}$$

The constant $T_1 > 0$ is the parameter which determines a "force" compelling the contour to be a straight line in polar representation (i.e. circle with center in given pole in original image). The value of T_1 depends on parameters of polar transform, namely from the scale of polar representation. While inner part depends on contour shape only, and does not depend on image, outer part is estimated from image characteristics and binds the contour to image. The outer part is the cost of passing through point (n, ρ') determined from local image characteristics in the point, and does not depend on contour shape: $C_n^{(O)}(\rho', \rho'') = w((n, \rho'))$. For a given path $S = \{\rho_n\}_{n=0}^{W-1}$ total cost is the sum of all transitions between adjacent points in the path:

$C(S)=C((0,\rho_0),(W,\rho_W))=\sum_{n=0}^{W-1}C_n(\rho_n,\rho_{n+1})$. The optimal contour is the sequence minimizing the whole cost: $S^*=\arg\min_S C(S)$. This discrete optimization problem may be solved by some known method, for instance, greedy algorithm as proposed by Sun and Pallottino (2003).

3. APPLICATION TO PUPIL BOUNDARY REFINEMENT

Application of CSP method to the problem of iris boundaries detection has specific issues. First, it is obvious that there are two circular contours in the image of an eye: pupil-iris boundary and iris-sclera boundary. Sometimes there is a contour of ophthalmic lens also. This makes it difficult to apply CSP method to initial detection of these borders, since detected contour can be any of these two and there is no perfect way of discriminating these two cases. This application was treated by Matveev (2011). So, feasible task for CSP method is refinement of already detected pupil and iris borders. Under this condition initial approximate locations of both iris boundaries is known. Thus the algorithm can run for a narrow ring containing the target boundary, rather than for the whole image. For a narrow polar representation rectangle with $\mathrm{H}<30$ a straightforward exhaustive search is faster than other elaborated algorithms.

The exhaustive search of optimal circular path is performed recursively as a set of steps. Each step involves a column of the polar representation raster (points with same value of φ). The cost of passing from a point $(0,\rho')$, in the first (left) column to the point (n,ρ''), in current column: $C((0,\rho'),(n,\rho''))\equiv C_{(n)}(\rho',\rho'')$. Since both ρ' and ρ'' range in $[1;H]$, it is necessary to calculate H^2 values $C_{(n)}$. They are calculated recursively starting from $C_{(1)}(\rho',\rho'')=1/\delta(\rho',\rho'')$. For each next column the cost of arriving to a point in it is a minimal sum of cost of arriving to some point ρ''' in the previous column and the cost of transition between adjacent columns:

$$C_{(n+1)}(\rho',\rho'')=\min_{\rho'''}\left(C_{(n)}(\rho',\rho''')+C_n(\rho''',\rho'')\right)=\min\left\{\begin{matrix}C_{(n)}(\rho',\rho'')+w(n,\rho''),\\ C_{(n)}(\rho',\rho''+1)+w(n,\rho''+1)+T_1,\\ C_{(n)}(\rho',\rho''-1)+w(n,\rho''-1)+T_1\end{matrix}\right\}$$

The incoming path for each point in the column (i.e. which of the three sums gave minimum) is recorded. At the final step (which has number $W+1$) H^2 values $C_{(W+1)}(\rho',\rho'')$ are obtained. Only values with $\rho'=\rho''$ correspond to closed contours. So the cost of optimal closed contour is $\min_\rho C_{(W+1)}(\rho,\rho)$ and it starts and ends at $\rho^*_{W+1}\equiv\rho^*_0=\arg\min_\rho C_{(W+1)}(\rho,\rho)$. From the detected radius ρ^*_{W+1} contour is tracked back easily from the recorded incoming paths.

Now consider the outer cost of transition via point $C^{(O)}(\varphi,\rho)=w(\varphi,\rho)$. From the task formulation it is clear that $w(\varphi,\rho)$ function should be constructed so as to be small in the points corresponding to the contour and big in other points. Contour points have strong brightness gradient value, thus points with small gradient should be rejected. This is done in the source image by checking the condition $\|\vec{g}\|>T_2$ in each image pixel and selecting only pixels, which satisfy this condition as possible contour points. Here $\vec{g}$ is brightness gradient vector and T_2 is a threshold. The value of T_2 is selected so as to suppress false gradients occurring due to image noise. If $3*3$ Sobel mask is used for gradient calculation the threshold can be set $T_2=6\sqrt{2}\max\{\sigma,2\}$, where σ is the brightness standard deviation caused by noise.

Next task-specific feature is that both pupil and iris are dark regions in brighter background, hence brightness gradients are directed outwards of the contour, and the angle between gradient in the point and radius-vector to this point from the center of the contour is enough small. This condition can be set as:

$$\arccos\left(\frac{\vec{x}\cdot\vec{g}}{\|\vec{x}\|\,\|\vec{g}\|}\right) < T_3 .$$

The value of threshold T_3 depends on the quality of the center detection algorithm, treated as an average (or maximum, or percentile) ratio of the distance D between detected center and true center to the radius R of the contour. It is calculated as $T_3 = \arcsin(D/R)$. Figure 3 shows the points of an image from Figure 1, satisfying both of the above conditions. The cost of transition is set to zero for these points, and is set to T_1 for all other points.

With these modifications CSP method was applied to the refinement of iris boundaries.

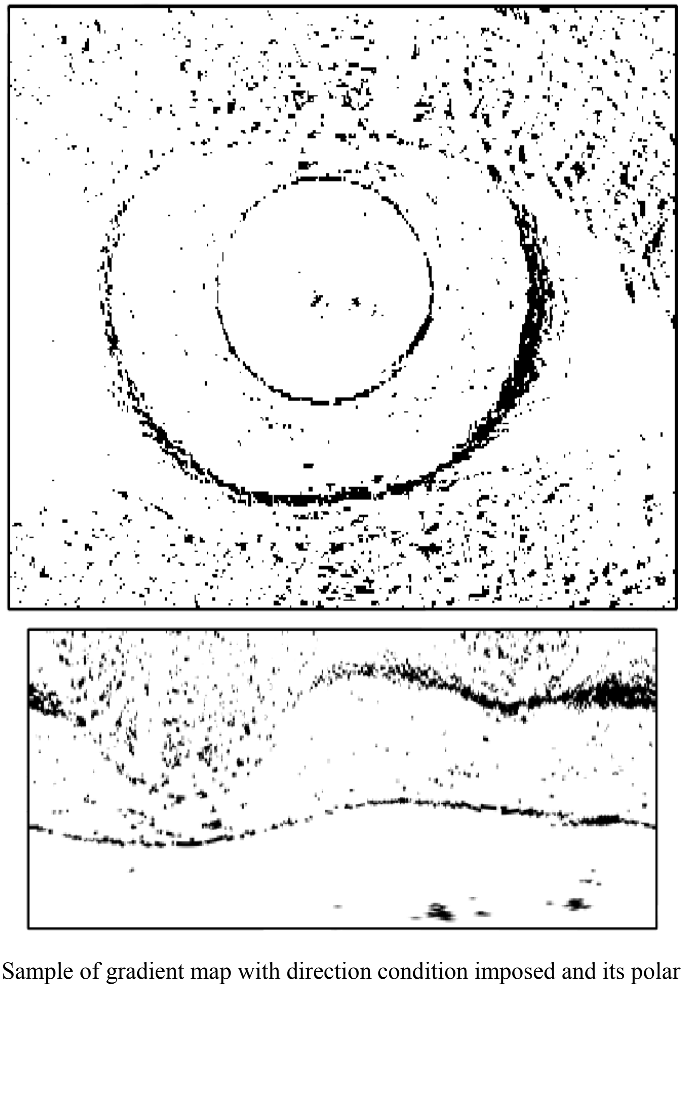


Figure 3. Sample of gradient map with direction condition imposed and its polar transform

4. EXPERIMENTS

Tests of CSP performance were done with the following iris image databases from public domain: UBIRIS.v1 (UBIRIS), CASIA-IrisV3 (CASIA), ND-IRIS (NDIRIS). Eye images were processed by human expert who indicated pupil and iris borders with most likely approximating circles. Thus each image was attributed with center positions and radii of pupil and iris circles. These data (call it *expert marking*) were then considered as "ground-truth" and were used for method verification. Unfortunately, there is no simple way to obtain refined border contours, that can be treated as "ground-truth". Human operator can manually mark quite a small share of huge image databases with such contours. This task is much more tedious and error-prone than marking circles.

So, direct tests of comparing refined borders to some "ground-truth" data and testing the quality of refinement method are not possible. Indirect methods were used instead. Refined borders were "simplified" back to circle, which has center in the point of mass center of the area, enclosed to the refined border. Radius of simplified circle was set to equate areas enclosed in this circle and in refined border. Call the circle obtained by original detection method as *original circle* and call simplified representation of refined border as *refined circle*. Although again circle, refined one does not match the original, and it can be a better approximation somehow. Figure 4 represents sample of original and refined circles. Left and middle images represent two variants of original detection which may occur with slightest changes in image brightness or algorithm parameters. Last image depicts the refined circle obtained from both variants.

Two approaches were used to compare original and refined circles. First approach is direct matching against expert marking, to estimate which kind of detection is more precise. Images from all three databases were used. Three ways were used to supply initially detected pupils and irises. First, expert marking data itself were spoiled with random noise to simulate improper detection. Second, algorithm developed by Masek (2003) was used. Third, an approximate method of detection by circular brightness gradient projections proposed by Matveev (2010) was employed. Mean square deviations of pupil center position and radius from expert marking were calculated for all three ways for original and refined circles. The scenario of this test is presented in Figure 5 and results are given in Table 1.

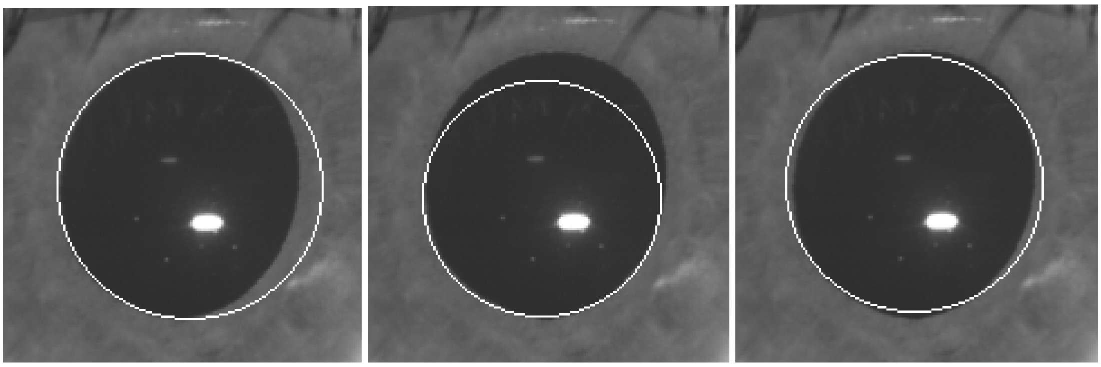


Figure 4. Variants of original pupil circle and refined circle.

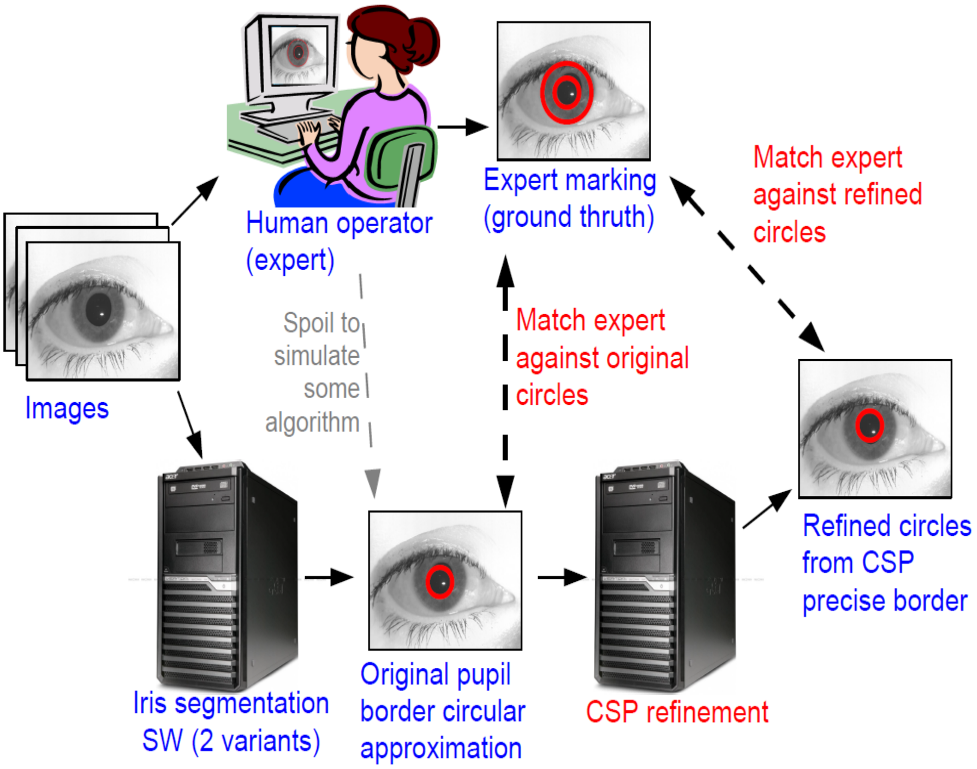

Human operator (expert)
Expert marking (ground thruth)
Match expert against refined circles
Images
Spoil to simulate some algorithm
Match expert against original circles
Refined circles from CSP precise border
Iris segmentation SW (2 variants)
Original pupil border circular approximation
CSP refinement

Figure 5. Scenario of tests of CSP refinement

Table 1. Error in pupil detection by various methods

Method	Average error of pupil radius detection in three databases, pixels			Average error of pupil center position in three databases, pixels		
	UBI	NDIRIS	CASIA	UBI	NDIRIS	CASIA
Spoiled expert Masek Matveev	8.16 4.65 7.78	8.16 7.23 6.34	8.16 5.15 5.81	5.77 3.24 5.13	5.77 5.59 4.27	5.77 3.67 4.11
Spoiled expert, refined Masek, refined Matveev, refined	5.59 4.07 4.89	2.52 2.41 2.09	1.86 1.58 1.45	3.26 2.88 3.51	1.63 2.07 1.52	1.53 1.14 1.09

CSP refinement makes improvement in precision of initial detection, although it is not effective for highly noisy images like UBIRIS.v1.

Second approach to estimate the quality and usability of refinement is judging by the "final characteristic", that is precision of iris recognition. The value of equal probability of recognition errors of first and second kind (*equal error rate, EER*) was chosen as such characteristic. CASIA Iris-Lamp database was used for tests, which contains 16213 images of 819 eyes of 411 subjects. The following steps were performed here. Templates were created from images of database by the algorithm described in the work of Daugman 2007.For its work the algorithm uses circular approximations of pupil and iris in each image. At first, pure expert marking was used for this purpose. The set of obtained templates was matched against itself and the EER value was estimated. For original expert marking its value is EER=0.752%. Then expert marking of pupils was refined by the proposed method, and same operations of template generation, matching and EER evaluation were done, with resulting EER=0.390%.

So, the refinement of pupil by circular shortest path method appears to reduce the recognition error. This can be explained by the imprecise marking of human expert.

5. CONCLUSION

Location of iris borders with high precision is an important task in automatic iris biometry. Though much attention is paid to iris border location in general, only few researchers tried developing special methods for iris border refinement after their initial detection. The authors have treated this aspect of iris border location problem with the help of circular shortest path optimization method. The CSP detection algorithm was modified to fit the peculiar properties of the task. The results of experiments show that refinement of pupil-iris boundary by CSP may be a useful addition to general scheme of iris border location.

REFERENCES

Bowyer, K., Hollingsworth, K., and Flynn, P., 2008. Image understanding for iris biometrics: A survey *Computer Vision and Image Understanding*. P.281-307.

Bowyer, K., Hollingsworth, K., and Flynn, P., 2012. *A survey of iris biometrics research: 2008-2010 Handbook of Iris Recognition*. Mark Burge and Kevin W. Bowyer, editors. Springer.

CASIA, 2005. Chinese academy of sciences institute of automation (CASIA), CASIA Iris image database http://www.cbsr.ia.ac.cn/IrisDatabase.htm.

Daugman, J., 2007.: New methods in iris recognition *IEEE Trans. on Systems, Man and Cybernetics. Part B: Cybernetics*. V.37. P.1167-1175.

He, Z., Tan, T., Sun, Z., and Qiu, X., 2009. Toward accurate and fast iris segmentation for iris biometrics. *IEEE PAMI*. V.31. P.1670-1684.

ISO, 2005. ISO/IEC 19794-6:2005 Information technology -- Biometric data interchange formats -- Part 6: Iris image data, 2005.

Kansky, J.J. 2003. *Clinical Ophthalmology: a Systematic Approach*. Elsevier, London.

Koh, J., Govindaraju, V., Chaudhary, V., 2010. A robust iris localization method using an active contour model and hough transform. *20th Int. Conf. on Pattern Recognition*. Istanbul. Turkey. P.2852-2856.

Maenpaa, T., 2005. An iterative algorithm for fast iris detection. *Int. Workshop on Biometric Recognition Systems*. Beijing. China. P.127.

Masek, L., 2003. Recognition of human iris patterns for biometric identification. http://www.csse.uwa.edu.au/pk/studentprojects/libor

Matveev, I.A., 2010. Detection of Iris in Image By Interrelated Maxima of Brightness Gradient Projections *Applied and Computational Mathematics*, V.9, No.2, pp.252-257.

Matveev, I.A., 2011. Circular Shortest Path as a Method of Detection and Refinement of Iris Borders in Eye Image. *Journal of Computer and Systems Sciences International*. V.50. N.5. P.778-784.

Nabti, M., Ghouti, L., and Bouridane, A., 2008. An effective and fast iris recognition system based on a combined multiscale feature extraction technique. *Pattern Recognition*. V.41. P.868--879.

Pan, L., Xie, M., and Ma, Z., 2008. Iris localization based on multiresolution analysis. *Proc. 19th Intern. Conf. Pattern Recognition*. Tampa, Florida, USA. P.1-4.

Phillips, P.J., Scruggs, W.T., O'Toole, A.J. et al., 2010. Frvt2006 and ice2006 large-scale experimental results *IEEE PAMI*. V.32. №.5. P.831-846.

Proenca, H., Alexandre, L.A., 2005. UBIRIS: A noisy iris image database *13th Intern. Conf. on Image Analysis and Processing*. V.3617. Cagliari, Italy: Springer, P.970-977.

Ross, A., Shah, S., 2006. Segmenting non-ideal iris using geodesic active contours *Biometrics Symposium: Special Session on Research at the Biometric Consortium*. Baltimore. USA, P.1–6.

Sun, C., Pallottino, S., 2003 Circular shortest path in images *Pattern Recognition*. V.36. N.3. P.709-719.

+ one self-citation removed

OCCLUSION HANDLING FOR PEDESTRIAN TRACKING USING PARTIAL OBJECT TEMPLATE-BASED COMPONENT PARTICLE FILTER

Daw-Tung Lin and Yen-Hsiang Chang
Department of Computer Science and Information Engineering - National Taipei University
151, University Rd., Sanshia Dist., New Taipei City, Taiwan

ABSTRACT

Pedestrian tracking plays an important role in the realm of security and intelligent video surveillance. Occlusion handling is a challenging issue in tracking multi-people. Therefore, adaptive and advanced solutions are required to fulfill the accurate tracking task and to analyze the pedestrian behavior for specific video surveillance purpose. This paper presents a novel method and address the problem of tracking and evaluating the number of people in complex scenes with occlusion conditions. In this work, collaboration of component-based human shape template and *Particle Filter* is developed and the ability of handling object occlusion is improved. The proposed system is capable of tracking specific person in real-time and handle multiple objects occlusion. The occlusion situation is predicted by using *Kalman Filter* and then each object is continuously tracked by the component shape template *Particle Filter*. Experimental results show that our algorithm is feasible and stable. The proposed tracking algorithm achieves an accuracy of up to 99.7%. The proposed approach outperforms that of the other methods for all test video datasets. Our low false negative rate reveals that the proposed tracking method is robust and superior in occlusion handling.

KEYWORDS

Pedestrian tracking, occlusion handling, video surveillance, template-based component matching, *Particle Filter*.

1. INTRODUCTION

In recent years, traditional video monitoring system has been replaced overwhelmingly by the intelligent video monitoring system. The intelligent video surveillance plays an important role in the realm of security. Among the applications of public space design, visual surveillance and intelligent environment, pedestrian detection and tracking is a key issue of computer vision system architecture (Enzweiler & Gavrila, 2009; Zhan et al., 2008). The major object tracking approaches include appearance-based (Erdem et al., 2003; Luo & Eleftheriadis, 2003; Van Beeck et al., 2011) and motion-based (Kim & Hwang, 2002; Lin & Huang, 2011) algorithms. Rosales and Sclaroff use the extended Kalman Filter to estimate 3D trajectory of an object from 2D motion (Paek et al., 2007). Ma et al. (2007) present a spatial-color model of object and develop an efficient visual tracking algorithm based on Particle Filter. Palaio and Batista (2008) represent the objects by covariance matrices and apply Particle Filter to perform object tracking. F. Li et al. (2008) combine color distribution histograms with Particle Filter and consider the target shape as a necessary factor in target model. Huang et al. (2008) perform shape analysis from foreground blobs and outputs from the foreground detection likelihood in the Particle Filter based object tracking. Shan et al. (2007) use Particle Filter and Mean Shift to track hand gesture. Reuter and Dietmayer (2011) utilize a sequential Monte Carlo multi-target Bayes filter based on random finite set theory for pedestrian tracking.

In pedestrian tracking domain, occlusion makes the detection and tracking of people a hard task to perform. Therefore, adaptive and advanced solutions are required to fulfill the tracking task and to analyze the pedestrian behavior for specific video surveillance purpose. To overcome the occlusion problem, several approaches have been developed. Enzweiler et al. (2010) propose a novel mixture-of-experts framework for pedestrian classification with partial occlusion handling. Ess et al. (2009) combine classical geometric world mapping with multi-person detection and tracking which jointly estimates camera position, stereo depth, object detections, and trajectories to improve tracking accuracy. Z. Li et al. (2008) improve the traditional mean shift tracking algorithm by using occlusion layers to represent pedestrian occlusion relation and adjusting the states of the related pedestrians to eliminate the occlusion effect during the tracking process. Ablavsky and Sclaroff (20111) formulate a layered graphic model for tracking partially occluded objects. An occluder-centric representation is defined as a first-order Markov process on activity zones with respect to the occlusion mask of the re-locatable object. Singh et al. (2008) present a two-stage multi-object tracking approach to robustly track pedestrians in occlusion scenarios by generating a high confidence partial track segments (tracklets) and then associate the tracklets in a global optimization framework. Corvee and Bremond (2010) adopt a hierarchical tree of histogram of oriented gradients (HOG) and couple with independently trained body part detectors to enhance the detection performance.

The objective of this work is to develop a robust tracking system for occlusion condition. Kalman Filter is applied to evaluate the velocity of object and predict the position of the object. When the object is moving nonlinearly, the Kalman Filter does not work well. Therefore, we further adopt Particle Filter method for nonlinear movement tracking. A robust occlusion detection algorithm is developed in this work for the situation that two or more objects are partially occluded. We further modify the Particle Filter algorithm and propose a template-based component matching method to robustly and accurately handle various occlusion situations during tracking process.

The remainder of this paper is organized as follows. Section 2 illustrates the architecture of the proposed tracking system. Section 3 elucidates the main technique of the proposed occlusion handling method. This technique contains occlusion detection and template-based component matching with particle filter. Section 4 describes the experimental dataset and presents the simulation results. Finally, conclusions are drawn in Section 5.

2. OBJECT TRACKING SYSTEM ARCHITECTURE

Occlusion problem is a challenging issue in object tracking. It is difficult to identify object with occlusion, because the foreground object cannot be completely extracted. In this paper, an efficient method has been proposed and resolves the occlusion problem. Figure 1 shows the block diagram of the proposed tracking system which contains two main sub-systems: video object correspondence and occlusion handling algorithm.

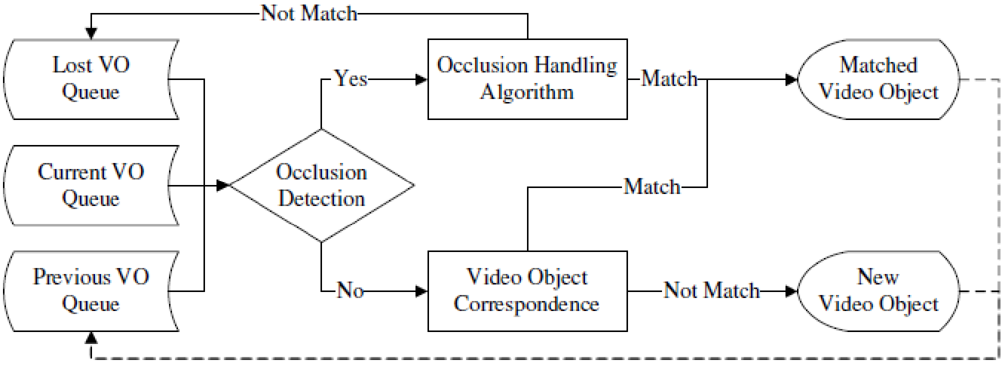


Figure 1. The overall block diagram of the proposed tracking system.

After we obtain the moving objects, we can track those moving object in the subsequent frames. Tracking is a continuously searching process for the best matched object in the current frame and the previous frame. Then the feature of object is updated so that we can predict the location of each object in the next frame for better object searching. Let VO_n^i denote a new object *i* extracted from the current frame *n*. To match VO_n^i with the object VO_{n-1}^j appears in the previous frame *n-1*, a feature matching function $FM(VO_n^i, VO_{n-1}^j)$ is computed to estimate the similarity between two objects VO_n^i and VO_{n-1}^j. Color histogram and object size are utilized as object matching features. The relation of histogram information and the physical property is shown as follows (Equation (1)).

$$FM(VO_n^i, VO_{n-1}^j) = \alpha \cdot PHist(VO_n^i, VO_{n-1}^j) + \beta \cdot PCnt(VO_n^i, VO_{n-1}^j)\,, \tag{1}$$

where α and β are the linear combinational weights of two features and α+β=1, $PHist(VO_n^i, VO_{n-1}^j)$ and $PCnt(VO_n^i, VO_{n-1}^j)$ denote the similarity probability of histogram and the similarity probability of object size between VO_n^i and VO_{n-1}^j, respectively. The similarity measure of histogram $PHist(VO_n^i, VO_{n-1}^j)$ is expressed as:

$$PHist(VO_n^i, VO_{n-1}^j) = \frac{1}{K}\sum_{z=1}^{K}\frac{C_z}{A_z + B_z - C_z}\,, \tag{2}$$

where K is the number of histogram bins, C_z is the minimum value of two bins (i.e. C_z=min(A_z;B_z)), A_z denotes the zth bin of histogram of VO_n^i and B_z denotes the zth bin of histogram of VO_{n-1}^j, and $1 \leqq z \leqq K$.

The similarity measure of the object size $PCnt(VO_n^i, VO_{n-1}^j)$ is as follows.

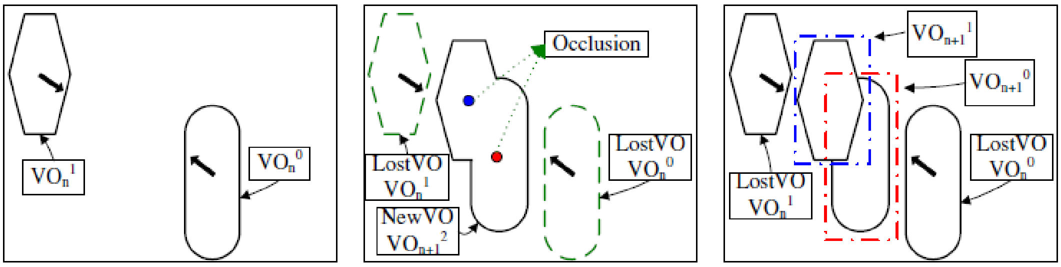
$$PCnt(VO_n^i, VO_{n-1}^j) = 1 - \frac{|u_i - u_j|}{u_i + u_j}\,, \tag{3}$$

where u_i and u_j represent the number of pixels of VO_n^i and VO_{n-1}^j, respectively. To improve the matching accuracy, the object movement features are adapted, such as velocity and position as illustrated in the next section.

3. OBJECT OCCLUSION HANDLING

3.1 Occlusion Detection

In this section, we depict the occlusion handling approach for the situation that two or more objects partially occluded and how we identify those objects. When the object is occluded, we conserve most of the features and just update the position prediction. By updating the position prediction, we can estimate the position of occluded object and determine whether the object is covered by other objects. Figure 2 illustrates the diagram of the proposed occlusion detection algorithm. Generally, the predicted position of occluded object is usually located in the nearby area of new object, and furthermore we can evaluate how many objects in the previous frame are now stay in the area of new object in the current frame.



(a) (b) (c)

Figure 2. The process of occlusion detection: (a) Frame n, (b) Frame n+1 with occlusion detection, (c) result of occlusion detection (two detected objects are drawn in red and blue line, respectively).

As illustrated in the Fig. 2(a), two objects VO_n^0 and VO_n^1 appears in frame n and are moving in different directions. Next, as shown in Fig. 2(b), VO_n^0 and VO_n^1 occlude with each other and form a new object VO_{n+1}^2 as seen in frame n+1. We can no more find objects VO_n^0 nor VO_n^1 in frame n+1. However, we can observe that the predicted centroid positions of VO_n^0 and VO_n^1 are in the area of VO_{n+1}^2. In Fig. 2(c), we show the tracking result and combine the rectangle mask and predicted position to identify the objects for occlusion situation. The proposed occlusion detection algorithm is summarized as follows.

Step 1: Predict the position of motion object VO_{n-1}^{i} in frame *n*-1 using Kalman Filter, record the object feature and update the predicted position.

Step 2: Check every object VO_{n}^{i} whether more than one predicted centroid of VO_{n-1}^{i} is located in the area of VO_{n}^{i}.

Step 3: Apply the similarity matching function (Equation (1)) and analyze the occluded video object VO_{n-1}^{i}.

3.2 Template-based Component Matching with Particle Filter

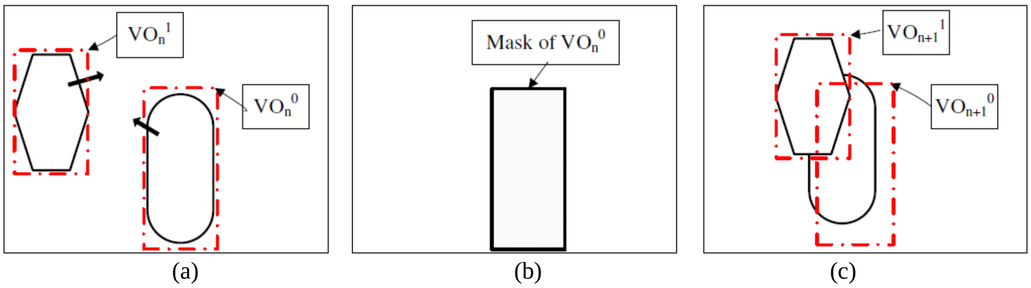


Figure 3. The rectangle mask of traditional Particle Filter: (a) the tracking result in n frame, (b) rectangle mask of sampling region, (c) the tracking result in n+1frame.

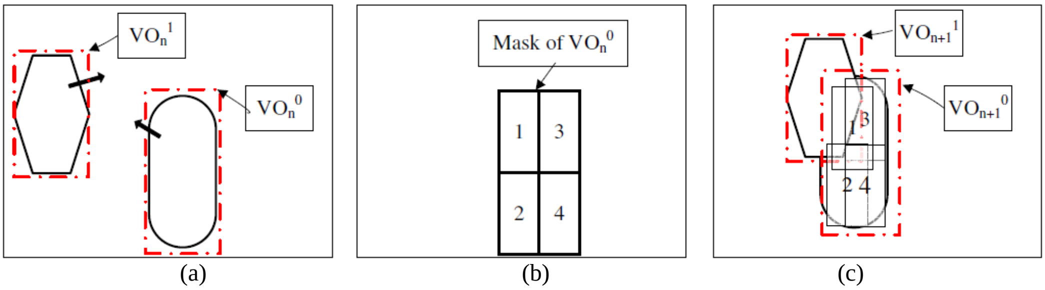


Figure 4. The rectangle mask of Particle Filter: (a) the tracking result in n frame, (b) four-part mask of sampling region, (c) the tracking result in n+1 frame.

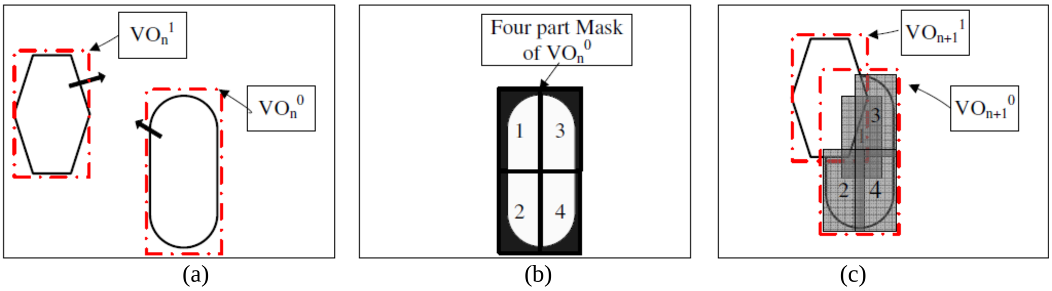


Figure 5. The template-based component masks of Particle Filter: (a) the tracking result in frame n, (b) template-based component masks of sampling region, (c) the tracking result in frame n+1.

Particle Filter is a popular object tracking method and is good for non-linear moving object prediction. We further combine color feature and *Particle Filter* to improve our tracking accuracy in occlusion situation. In the measurement update step of *Particle Filter* algorithm, the *Bhattacharyya* distance D is adopted to evaluate the similarity of hue value histogram of object *HSV* color model between each particle and object VO_{n-1}^{i}. When the value of *Bhattacharyya* parameter equals to 1, the case means two color histograms are the same. After we obtain distance distribution, we choose the measurement likelihood function as follows.

$$p(Z_t \mid X_t^i) \propto N(D;0,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp(\frac{D^2}{2\sigma^2}) \tag{4}$$

The larger the value of the above equation is, the more similar the color correlation histogram between the particle and VO_{n-1}^i is. After estimating the posterior mean state, the result presents the position of VO_{n-1}^i.

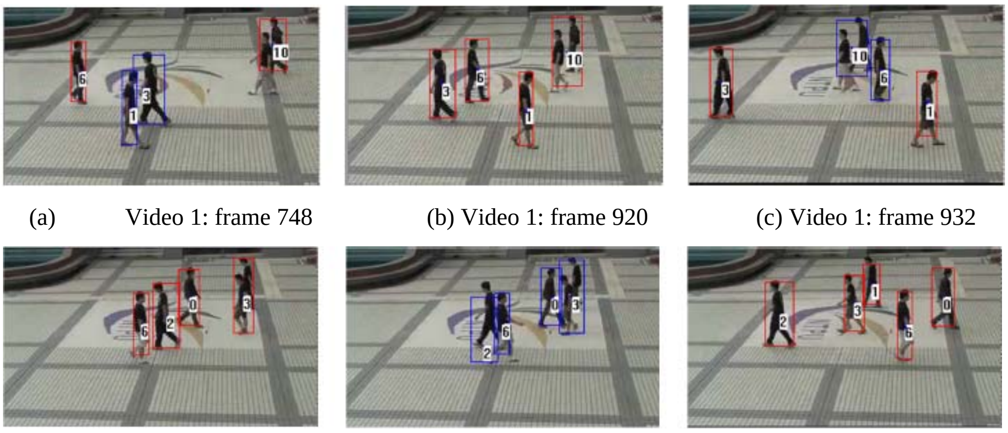
Traditional *Particle Filter* utilizes rectangle mask or ellipse mask as the sampling region for each particle. The rectangle or ellipse is in fixed shape and fixed size. We reform the sampling for better tracking accuracy. Fig. 3 illustrates the traditional Particle Filter approach. The traditional Particle Filter encounters occlusion problems. Because the Particle Filter estimates the mean movement as the tracking location. The result of tracking position will move to the false region. To resolve this problem, we have presented a novel procedure that utilizes component Particle Filter (Liu, 2008). First, we divide the rectangle mask into four components as shown in Fig. 4 . If the object is partially occluded, we can still track accurately by the other components. However, the shortage is that the human shape is irregular, the rectangle mask is not very suitable for people tracking. To further improve the tracking accuracy, we utilize the object shape to revise the component Particle Filter and propose the template-based component Particle Filter tracking approach. The template-based component masks are based on the shape of VO_n^i. Example is shown in Fig. 5 (b), the mask of VO_n^i is obtained from the shape of VO_n^i. Then, we divide this mask into four components. Each template of object VO_n^i is unique so that the result of Particle Filter matching not only obtained based on the similarity measure of color histogram and object size but also rely on the similarity of shape. As illustrated in Fig. 5(c), although one of the masks results in failure tracking due to occlusion, the other masks track successfully with the corresponding components because VO_n^i is just partially occluded and do not affect the other regions of VO_n^i.

4. EXPERIMENTAL RESULTS

The proposed template-based component particle filter tracking system has been implemented in Visual C++ 7.0 under a Windows platform. The developed system was tested in experiments on four video clips, including two test videos taken on the campus and two videos of PETS2006 evaluation dataset. The ground truth of motion pedestrians were manually generated from these four video clips. Our system performs at the speed of 10 frames per second for multiple object tracking on Intel Core 2 Duo 2.4GHz processor with 2GB Ram for 320x240 resolution video. Figures 6 and 7 demonstrate some of the cases of tracking results. The red rectangle and blue rectangle represent the tracked location of non-occlusion and occlusion conditions, respectively. The same ID in different frames indicates successful tracking of the same pedestrian in each consecutive video sequences.

This study compares the performance of pedestrian tracking with the results obtained by two well-known algorithms, namely, Mean Shift and Particle Filter, and our previous work Four-part Particle Matching (Liu, 2008). Two types of evaluation indices were adopted, i.e. frame-based statistics and object-based counting. In frame-based evaluation, three measurements were reported: correct tracking indicates the correct number of object tracking in each frame; missing tracking represents the number of objects which were lost tracked; false tracking indicates objects were identified as a new object or false positive tracking. Furthermore, we adopt the object-based indices to evaluate the tracking performance (Senior et al., 2006): false positive rate *fp* and false negative rate *fn* as ratios of numbers of object tracking.

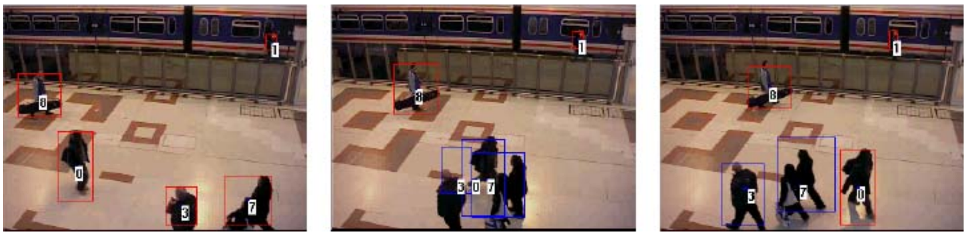
Table 1 summarizes the performance evaluation using the above mentioned indices. The proposed tracking algorithm achieves an accuracy of up to 99.7% when testing on the campus video (Video 2). Apparently, the proposed approach outperforms that of the other methods for all test video dataset. The high missing tracking in PETS2006 S-T1-G test video (Table 4) is due to some pedestrians are too small to be detected. For example, the person with ID label 8 in Fig. 7(a) walks from right to left in the scene becomes too small to be detected. Meanwhile, the person with label 0 encounters occlusion (shown in Fig. 7(b)). Then, Fig. 7(c) shows that the person still can be tracked successfully with the occlusion situation. Our low false negative rate reveals that the proposed tracking method is robust in occlusion handling.



(a) Video 1: frame 748 (b) Video 1: frame 920 (c) Video 1: frame 932

(d) Video 2: frame 947 (e) Video 2: frame 953 (f) Video 2: frame 1025

Figure 6. The results of pedestrian tracking tested on video 1 and video 2.



(a) Frame 968 (b) Frame 10000 (c) Frame 1021

Figure 7. The results of pedestrian tracking tested on PETS2006 S-T1-G.

Table 1. Performance comparison of tracking accuracies obtained by different methods measured on four video clips.

Video	Tracking Algorithm	Frame Measure				Object Measure	
		Correct Tracking	Miss Tracking	False Tracking	Correct Fraction	f_p	f_n
Video 1	Mean Shift	3571	28	4	99.1%	6/13	0/13
	Particle Filter	3587	12	4	99.6%	4/13	0/13
	Four-parts Matching	3588	11	4	99.6%	4/13	0/13
	Our Method	3592	7	4	99.7%	2/13	0/13
Video 2	Mean Shift	3705	114	6	96.9%	5/24	2/24
	Particle Filter	3759	60	4	98.3%	4/24	2/24
	Four-parts Matching	3755	64	4	98.2%	5/24	3/24
	Our Method	3785	34	4	99.0%	3/24	2/24
PETS 2006 S5-T1-G	Mean Shift	2627	1308	8	66.6%	8/25	5/25
	Particle Filter	2500	1435	8	63.4%	7/25	4/25
	Four-parts Matching	2495	1440	7	63.3%	7/25	4/25
	Our Method	2820	1115	7	71.5%	7/25	4/25
PETS 2006 S3-T7-A	Mean Shift	1809	531	10	77.0%	5/14	1/14
	Particle Filter	1781	559	8	75.9%	5/14	2/14
	Four-parts Matching	1836	504	8	78.2%	5/14	2/14
	Our Method	2192	148	8	93.4%	2/14	1/14

5. CONCLUSION

In this paper, we develop a tracking algorithm combining collaboration of component-based human shape template and Particle Filter to improve the ability of handling object occlusion. The proposed system is capable of tracking specific person in real-time and handle multiple objects occlusion. The occlusion situation is predicted by using Kalman Filter and then each object is continuously tracked by the Particle Filter. Finally, experimental results show that our algorithm is feasible and stable. The proposed tracking algorithm achieves an accuracy of up to 99.7% when testing on the campus video. The proposed approach outperforms that of the other methods for all test video datasets. Our low false negative rate reveals that the proposed tracking method is robust and superior in occlusion handling.

ACKNOWLEDGEMENT

This work is partially supported by the National Science Council under project funding contract Nos. NSC 99-2221-E-305 -012 -MY2 and NSC 101-2221-E-305-011-MY2.

REFERENCES

Ablavsky, V. and Sclaroff, S., 2011. Layered Graphical Models for Tracking Partially Occluded Objects. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 9, pp. 1758-1775.

Corvee, E. and Bremond, F., 2010. Body Parts Detection for People Tracking Using Trees of Histogram of Oriented Gradient Descriptors. *Proceedings of 2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 469-475.

Enzweiler, M. and Gavrila, D.M., 2009. Monocular Pedestrian Detection: Survey and Experiments. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 12, pp. 2179-2195.

Enzweiler, M. et al, 2010. Multi-cue Pedestrian Classification with Partial Occlusion Handling. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 990-997.

Erdem, C.E., et al, 2003. Video Object Tracking with Feedback of Performance Measures. *In IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 4, pp. 310-324.

Ess, A. et al, 2009. Improved Multi-person Tracking with Active Occlusion Handling. *Proceedings of ICRA Workshop on People Detection and Tracking*, Vol. 2.

Huang, Y. et al, 2008. A Method of Small Object Detection and Tracking Based on Particle Filters. *Proceedings of the 19th International Conference on Pattern Recognition*, pp. 1-4.

Kim, C. and Hwang, J.N., 2002. Fast and Automatic Video Object Segmentation and Tracking for Content-based Applications. *In IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, No. 2, pp. 122-129.

Li, F. et al, 2008. A Hybrid Object Tracking Method in Complex Backgrounds. *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering*., Vol. 1.

Li, Z. et al, 2008. Improved Mean Shift Algorithm for Occlusion Pedestrian Tracking. *In Electronics Letters*, Vol. 44, No. 10, pp. 622-623.

Lin, D.-T. and Huang, K.-Y, 2011. Collaborative Pedestrian Tracking and Data Fusion with Multiple Cameras. *In IEEE Transactions on Information Forensics and Security*, Vol. 6, No. 4, pp. 1432-1444.

Liu, L.-W., 2008. *Object Tracking and People Counting at Multiple Distances*. Master's Thesis, Department of Computer Science and Information Engineering, National Taipei University.

Luo, H. and Eleftheriadis, A., 2003. Model-based Segmentation and Tracking of Head-and-shoulder Video Objects for Real Time Multimedia Services. *In IEEE Transactions on Multimedia*, Vol. 5, No. 3, pp. 379-389.

Ma, J. et al, 2007. Efficient Visual Tracking Using Particle Filter. *Proceedings of the 10th International Conference on Information Fusion*, pp. 1-6.

Paek, E., et al, 2007. Mutiple-view Object Tracking Using Metadata. *Proceedings of International Conference on Wavelet Analysis and Pattern Recognition*, Vol. 1, pp. 12-17.

Palaio, H. and Batista, J., 2008. Multi-Objects Tracking Using an Adaptive Transition Model Particle Filter with Region Covariance Data Association. *Proceedings of IEEE International Conference on Pattern Recognition*, pp. 1-4.

Reuter, S. and Dietmayer, K., 2011. Pedestrian Tracking Using Random Finite Sets. *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, pp. 1-8.

Senior, A. et al, 2006. Appearance Models for Occlusion Handling. *In Image and Vision Computing*, Vol. 24, No. 11, pp. 1233-1243.

Shan, C. et al, 2007. Real-time Hand Tracking Using a Mean Shift Embedded Particle Filter. *In Pattern Recognition*, Vol. 40, No. 7, pp. 1958-1970.

Singh, V.K. et al, 2008. Pedestrian Tracking by Associating Tracklets Using Detection Residuals. *Proceedings of IEEE Workshop on Motion and video Computing*, pp. 1-8.

Van Beeck, K. et al, 2011. Towards An Automatic Blind Spot Camera: Robust Real-time Pedestrian Tracking from a Moving Camera. *Proceedings of the 12th IAPR Conference on Machine Vision Applications*, pp. 528-531.

Zhan, B., et al, 2008. Crowd Analysis: A Survey. *In Machine Vision and Applications*, Vol. 19, No. 5, pp. 345–357.

COMBINATION OF SHAPE AND VISUAL INFORMATION FOR THE REGISTRATION OF 3D POINT CLOUDS FROM TOF CAMERAS

Dominik Aufderheide[1,2], Peter Planert[1], Werner Krybus[1] and Gerard Edwards[2]
[1] South Westphalia University of Applied Sciences - Institute for Computer Science, Vision and Computational Intelligence - Luebecker Ring 2, 59494 Soest, Germany
[2] The University of Bolton - Engineering, Sports and Sciences Academic Group - Deane Road, Bolton, BL3 5AB, U.K.

ABSTRACT

Computing the correct alignment of 3D point clouds is an import task for many different applications. The vast majority of the suggested procedures realise the spatial frame-to-frame alignment of the 3D measurements by applying the iterative closest point algorithm (ICP) or variants thereof. ICP in general considers only the 3D shape for the computation of the relative pose between two given point clouds. By using actual state-of-the-art time of flight sensors (ToF) there are also visual measurements available which are completely neglected within classical ICP. This paper describes a novel framework which also employs visual information as an aiding modality for the registration process, which results in a higher accuracy of the pose estimation and lower computational costs, in comparison to the classical ICP.

KEYWORDS

Point Cloud Registration (PCR), Time-of-Flight camera (ToF), Iterative Closest Point (ICP), RGB-D sensors

1. INTRODUCTION

The use of three-dimensional data acquired from real-world objects or scenes has become a widely accepted practice in industry within the last decade. These acquired 3D models form the basis for different applications, such as reverse engineering, rapid prototyping and simulation.

The general procedure of generating a dense and complete visual model of an object, can be subdivided into different stages, as shown in Fig. 1. The process begins with the acquisition of 3D measurements (3D point clouds in a metric Euclidean coordinate system) from different viewpoints located around the object. The second step contains an optional pre-processing of the raw point clouds such as noise filtering, background subtraction or segmentation. In order to generate a complete and closed visual model of the object, the 3D point measurements need to be moved in the same coordinate system. Since all measurements are relative to the position of the acquisition device this is only possible if the motion of either the camera or the object during the acquisition process can be calculated based on the set of point clouds. This fusion of point clouds is often referred to as point cloud registration (PCR).

Many sensor units, such as laser scanners (see (Axelsson 1999)) or structured light scanners (see (Koninckx et al. 2003)) produce an immense amount of 3D point measurements per pose, which results in high computational costs to compute iteratively the frame-to-frame alignment of the captured point clouds, without using any kind of prior knowledge. In the past, this operation was considered as an offline batch processing task. Today the increasing capability of the available computing hardware, coupled with the advances in corresponding sensory techniques hint at the possibility of real-time PCR, which would be a big step towards an on-the-fly scene acquisition framework. Aufderheide and Krybus (2010) have described the numerous applications for such a modeling device.

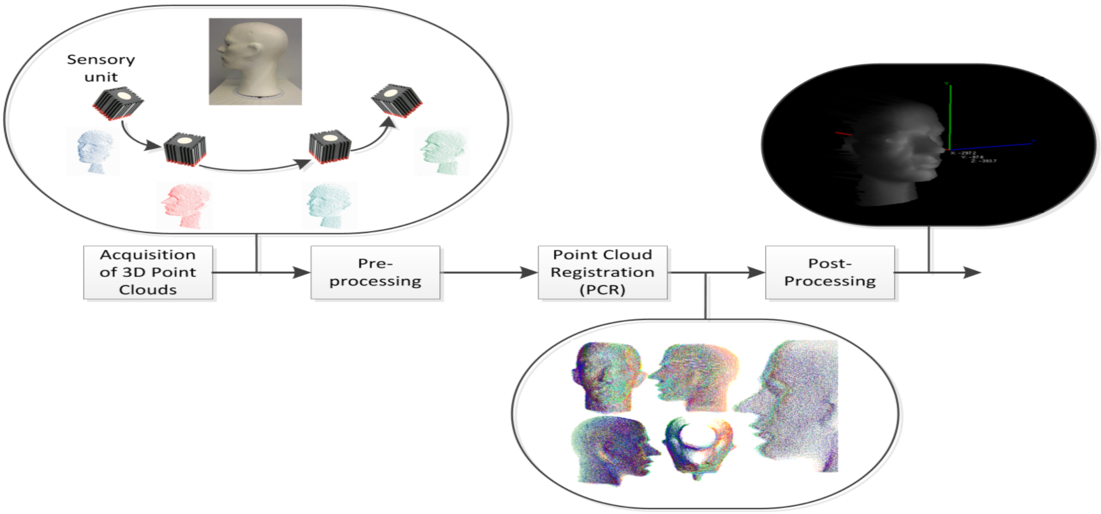

Sensory unit
Acquisition of 3D Point Clouds
Pre-processing
Point Cloud Registration (PCR)
Post-Processing

Figure 1.Generalized 3D reconstruction process based on point cloud measurements including four stages

It is possible to identify two distinctive directions of research in this context, where the most promising results in recent years were produced by using classical algorithms for PCR within a massive-parallel implementation e.g. on GPU[1] or FPGAs[2]. The most prominent example is the KinectFusion project carried out by Microsoft research based on their Kinect sensor. Newcombe et al. (2011) implemented a hierarchical ICP for real-time sensor tracking in six degrees of freedom (6 DoF) with a highly parallel architecture on GPUs. Belshaw & Greenspan (2011) presented a highly efficient implementation of a brute-force nearest neighbours based ICP for object tracking, on an FPGA platform and was able to carry out a speed of 200 frames per second. This speed performance is faster than a PCR routine based on an AK-d tree based ICP implemented in software.

Nevertheless, it is also possible to identify research directions which focus on the extension of classical ICP-based methods, in order to reduce the computational costs of PCR. In this context it is possible to (i) optimize the minimisation problem within the ICP, (ii) optimize the search of point correspondences which are the base for the ICP or (iii) estimate a robust initial guess of the rigid transformation $\boldsymbol{T}$ (includes rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$) that transforms the two point clouds.

This paper concentrates on the last issue while we suggest a crude-to-fine ICP (C2F-ICP) framework which utilizes sparse visual features to compute an initial crude estimate of the rigid transformation which is then used in a second step to compute the final refined pose based on classical ICP.

The remainder of this paper is organised as follows: section 2 gives an overview about classical ICP implementations and the typical drawbacks of that framework. Section 3 covers the broad concept of incorporating visual information into PCR and describes the estimation of a preliminary rigid transformation. Section 4 gives an overview and a detailed description of the C2F-ICP framework. Section 5 summarises the experimental evaluation of the proposed framework and provides some implementation details. Finally section 6 concludes the whole work and shows potential future work.

2. ITERATIVE CLOSEST POINT (ICP) – AN OVERVIEW

One of the fundamental weaknesses of classical ICP schemes for point cloud registration is the concentration on shape information only. Even if common sense suggests that dense 3D point clouds contain generally enough discriminative information for the computation of a precise alignment, there are numerous examples where shape-based PCR methods will fail. This can easily be clarified by analyzing the general structure of the ICP itself:

[1] GPU – Graphics processing unit

[2] FPGA – Field programmable gate array

The problem of finding the optimal rigid transformation $\boldsymbol{T}$ that aligns two given sets of points $S=\{\boldsymbol{s}_{1,}\boldsymbol{s}_2,\cdots\boldsymbol{s}_n\}$ with $\boldsymbol{p}_i\in\mathbb{R}^3$ and $D=\{\boldsymbol{d}_{1,}\ \boldsymbol{d}_2,\cdots\boldsymbol{d}_m\}$ with $\boldsymbol{d}_i\in\mathbb{R}^3$ within a unified coordinate system based on ICP can be subdivided into four different steps:

Initial transform – The first step in ICP algorithm is the initial transform of point cloud S with a given estimate of the rigid transformation $\breve{\boldsymbol{T}}$. In most cases, the initial estimate is propagated either from the former iteration of ICP for the actual frame or the final transformation from the previous frame, based on a constant velocity assumption.

Correspondence – The next step consists of the identification of homologous information between the two given point clouds S and D, which means that 3D point pairs $[\boldsymbol{s}_i,\boldsymbol{d}_j]$ are identified which describe the same physical world point. Due to the fact, that the identification of corresponding points within huge point clouds is computational complex, Rusinkiewicz and Levoy (2001) suggest a former selection of a subset of points, e.g. based on uniform or random subsampling. This leads to the generation of simplified points sets S' and D'**.** The matching of corresponding points is realized in most cases by using a simple strategy based on minimizing the Euclidian distance between a point $\boldsymbol{s}_k$ from S' and a corresponding point $\boldsymbol{d}_k$ from D'. Other strategies e.g. Chen et al. (1999) utilize the surface normal vector of a point $\boldsymbol{s}_k$ to find a corresponding point in D', which is often labeled as *normal shooting*. Neugebauer (1997) introduced a technique based on *reverse calibration*, which projects the source point onto the destination mesh from the point of view of the destination mesh's range camera. Fig. 2 illustrates the differences between those classical matching techniques.

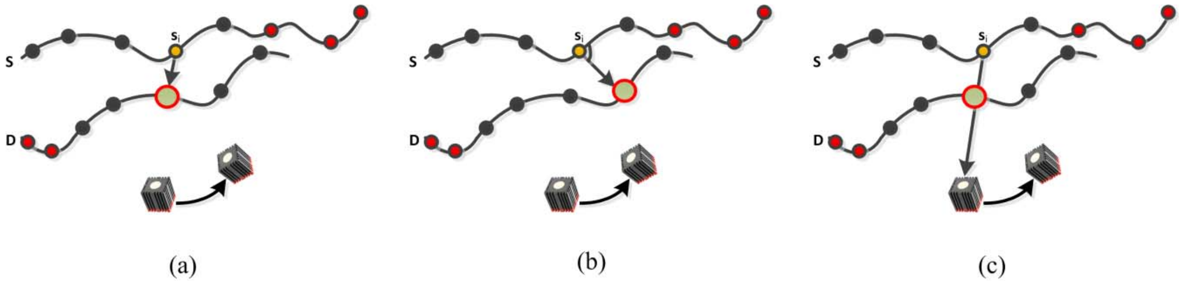


Figure 2. Comparison of methods for finding homologous points – (a) Closest point by Euclidean distance, (b) Normal shooting, (c) Reverse calibration

It was shown by Zhang (1994) that the correspondence search is the most computationally expensive step in the ICP algorithm: If the first point cloud contains *n* points and the second data set contains *m* points, the complexity of a closest point query within the complete search space is $O(nm)$. By introducing a k-dimensional binary tree (kd-tree) it is possible to reduce the computational complexity to $O[n\,log(m)]$. The use of k-d trees for closest point computation converts the problem to the search within a binary tree. At each node of the tree, a test is performed to decide which side of a hyperplane the closest point will lie on.

Pose estimation – Based on the set of correspondences (mathematically at least three point pairs are necessary) the parameters of the rigid transformation can be computed. The optimal transformation, which maps cloud S onto D, $\widetilde{D}=\boldsymbol{R}S+\boldsymbol{t}$, can be found by minimizing the error between all N point pairs as shown in the following equation:

$$\min_{\boldsymbol{R},\boldsymbol{t}}\sum_{i=1}^{N}\|\boldsymbol{d}_i-(\boldsymbol{R}\boldsymbol{s}_i+\boldsymbol{t})\|^2 \qquad (1)$$

Arun et al. (1987) have suggested that this can be realized efficiently by applying a singular value matrix decomposition (SVD) in a non-iterative way to $\boldsymbol{R}$ and $\boldsymbol{T}$.

Evaluating registration accuracy – Once $\boldsymbol{R}$ and $\boldsymbol{t}$ are computed by SVD decomposition, the quality of the result has to be evaluated in order to decide if the ICP needs to continue for another iteration. Due to the fact that the whole optimization problem is based on minimizing the least-squared distance from Equation 1, this is used as an indicator to 'abort' the algorithm. So if the least-squared distance of the computed transformation falls below a certain threshold ζ_{LS} the abort criteria is fulfilled. Due to the fact that for noisy data sets it is difficult to find an optimal threshold value, in addition also the following abort criteria are used:

I. Stop ICP, if squared distance for actual estimate of $\boldsymbol{R}$ and $\boldsymbol{t}$ lies below a certain threshold:

$$\sum_{i=1}^{N} \|\boldsymbol{d}_i - (\boldsymbol{R}\boldsymbol{s}_i + \boldsymbol{t})\|^2 < \zeta_{LS} \tag{2}$$

II. Stop ICP, if the incremental rotation and translation relative magnitudes are both less than thresholds, here $\boldsymbol{R}$ and $\boldsymbol{t}$ are the actual estimates from actual iteration and $\boldsymbol{R}^-$ and $\boldsymbol{t}^-$ the estimates from the former iteration:

$$\frac{|\boldsymbol{R}^-|}{|\boldsymbol{R}|} < \zeta_{Rr} \wedge \frac{|\boldsymbol{t}^-|}{|\boldsymbol{t}|} < \zeta_{Tr} \tag{3}$$

with $|\boldsymbol{t}| = \sqrt{\sum_{i=1}^{3}(t_i^2)}$ and $|\boldsymbol{R}| = \sqrt{\sum_{i=1}^{3}(\omega_i^2)}$. Here ω are rotations about the *x*, *y*, and *z* axes as coded within the rotation matrix $\boldsymbol{R}$.

III. Stop ICP, if the incremental rotation and translation absolute magnitudes are both less than thresholds:

$$|\boldsymbol{R}^-| < \zeta_{Ra} \wedge |\boldsymbol{t}^-| < \zeta_{Ta} \tag{4}$$

IV. Stop, if the number of iterations exceeds a certain maximum ζ_{IT}.

If one of the abort criteria is fulfilled the actual estimates $\boldsymbol{R}^-$ and $\boldsymbol{t}^-$ are used as final estimates $\boldsymbol{R}$ and $\boldsymbol{t}$, otherwise the four steps of the ICP are repeated.

The fundamental problem of the ICP algorithm is the fact that many iterations are required if the initial guess is poor. These many iterations which have a high computational cost, mean that the ICP algorithm run on a non parallel architecture is not suited for real-time operation.

Another important drawback of all classical ICP frameworks is that only structural information is used for the alignment of the shapes, which can be explained from an historical point of view, because for many decades the available sensory units (e.g. laser scanners) delivered only depth information. Nowadays time-of-flight (ToF) or other RGB-D[3] sensors are available which are able to deliver up to 60 depth images per second and also the corresponding RGB or greyscale images. So the scheme described above neglects a reasonable amount of information for PCR. The following section introduces a scheme for using the available visual information to compute an initial estimate of the rigid transformation between two point clouds.

3. VISUAL POINT CLOUD REGISTRATION

The general fact that ICP algorithms neglect all visual information delivered by a ToF or RGB-D sensor is not just a drawback in terms of efficiency or accuracy, but also in terms of robustness. This is illustrated in the following figure, where two different objects (a sphere and a pyramid) are observed by a moving depth camera. The corresponding depth images for the two different viewpoints indicate that structural depth information alone is not suitable, in such a case, to fuse different point clouds, because there are not enough distinctive points to establish a set of correspondences in 3D.

Such a scenario can lead to a non-converging behaviour of the ICP and even worse the complete wrong fusion of point clouds. In such a case the visual information can help to guide the ICP in such a way that it converges to a minimum, because it is possible to estimate an initial transformation based only on visual information.

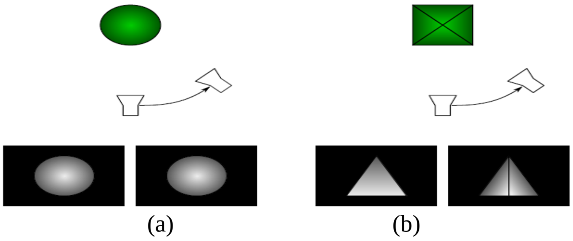


Figure 3. Depth maps delivered by a moving range sensor for (a) a spherical object and (b) a pyramidal object[4]

[3] RGB-D – RGB-Depth

[4] Figure adopted from Korth (2013).

The following figure shows a scheme for the usage of visual information in estimation of rigid transformation for PCR.

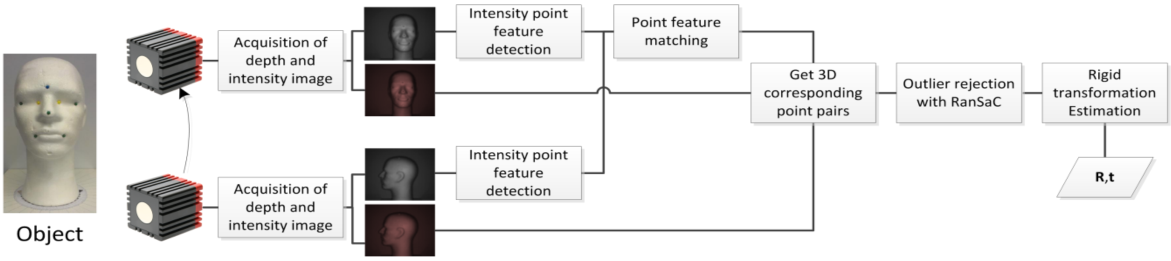

Object
Acquisition of depth and intensity image
Intensity point feature detection
Point feature matching
Get 3D corresponding point pairs
Outlier rejection with RanSaC
Rigid transformation Estimation
R,t
Acquisition of depth and intensity image
Intensity point feature detection

Figure 4. Framework for PCR based on visual information

Two subsequent intensity or RGB images ($\boldsymbol{I}_k$ and $\boldsymbol{I}_{k+1}$) and depth images ($\boldsymbol{D}_k$ and $\boldsymbol{D}_{k+1}$) are acquired by a ToF camera, where a pixel $[u, v]$ in $\boldsymbol{I}_k$ represents the same world point as the same pixel in $\boldsymbol{D}_k$. The first step consists of the detection of distinctive point features in both intensity images. So two sets of 2D point features are computed which are labelled as $X = [\boldsymbol{i}_1, \dots, \boldsymbol{i}_n]$ with $\boldsymbol{i}_i \in \mathbb{R}^2$ and $X' = [\boldsymbol{j}_1, \dots, \boldsymbol{j}_m]$ with $\boldsymbol{j}_i \in \mathbb{R}^2$. For this work different state-of-the-art methods from computer vision, such as scale-invariant feature transform (SIFT) from (Lowe (2004)), speeded-up robust features (SURF) as suggested by (Bay et al. (2008)), features from accelerated segment test (FAST) from (Rosten (2010)) and center surround extremes (CenSurE) as introduced by (Agrawal 2008)[5] are implemented and evaluated for their applicability in the given context. Fig. 5 shows examples for two given intensity images and the resulting point features by using SIFT feature detector.

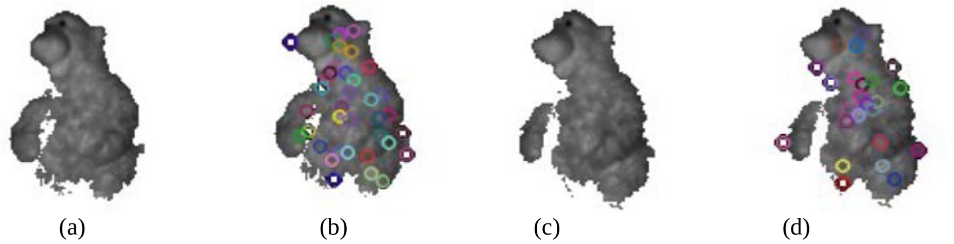


Figure 5. Examples of two intensity images (a) - $\boldsymbol{I}_k$, (c) - $\boldsymbol{I}_{k+1}$ and the detected point features (b) – X, (d) – X'

The point features build the base for finding 2D correspondences. The matching itself is realized by using the point descriptors suggested by (Lowe (2004)), which build a 16x16 pixel local neighborhood around each point. Within this neighbourhood the magnitude and orientation of the image gradients are computed. The orientations are collected within a histogram, where each point is weighted by its magnitude. The main orientation Θ is denoted as the orientation of that keypoint and the orientation histogram forms the base for a 128-dimensional feature descriptor. The matching is done in feature space by using squared absolute distances between feature descriptors.

It was shown in Aufderheide et al. (2009) that feature matching in 2D intensity images is inherently an unstable problem and even an optimal matching strategy would lead to erroneous point pairs within the set of correspondences. This is illustrated by Fig. 6, which shows typical matching results by applying SURF for different object rotations.

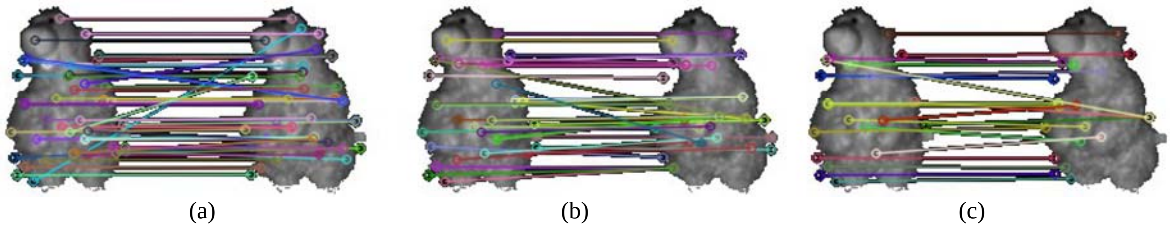


Figure 6. Examples of feature matching results with SURF for different angle differences (a) 5°, (b) 10°, (c) 15° between first and second intensity image

[5] We used the simplified STAR detector.

For that reason the following algorithm for visual PCR needs to implement a strategy for outlier rejection. Our suggestion is the implementation within a random sample consensus (RanSaC) scheme. The first step is that for each 2D point pair $[\boldsymbol{i}_k, \boldsymbol{j}_l]$ the corresponding 3D point pair $[\boldsymbol{s}_k, \boldsymbol{d}_l]$ from the depth images is chosen to build a set of 3D correspondences. These can be used to estimate the rigid transformation just as described above. To reject the outliers within the dataset the estimation is embedded within a RanSaC scheme as shown in the following figure.

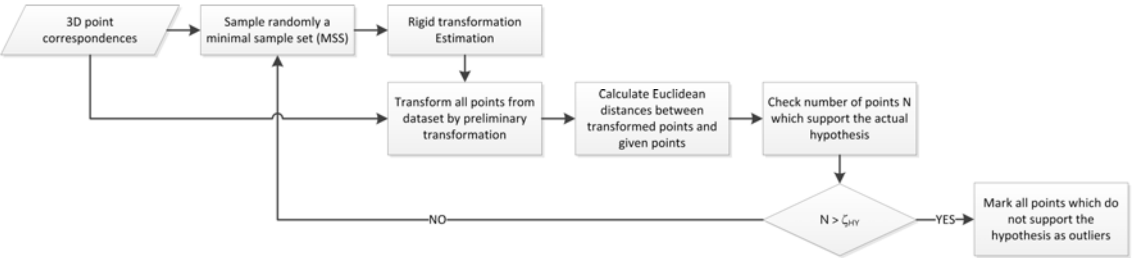

3D point correspondences
Sample randomly a minimal sample set (MSS)
Rigid transformation Estimation
Transform all points from dataset by preliminary transformation
Calculate Euclidean distances between transformed points and given points
Check number of points N which support the actual hypothesis
N > ζ_HY
NO
YES
Mark all points which do not support the hypothesis as outliers

Figure 7. RanSaC scheme for rejecting outliers during visual rigid transformation estimation

The estimation found during the last iteration of the suggested scheme can be used within a crude-to-fine ICP scheme as shown in the following section.

4. CRUDE-TO-FINE ITERATIVE CLOSEST POINT (C2F-ICP)

As mentioned above one major problem by applying ICP for PCR is to find a "good" estimate for the starting transformation $\boldsymbol{T}_0$ in order to reduce the number of iterations of the ICP. We suggest the usage of the visual-PCR (V-PCR) from section 3 within a crude-to-fine ICP scheme (C2F-ICP). This scheme uses the parameters of the rigid transformation, as computed with the above scheme, for visual point cloud registration as the initial parameters for a subsequently executed classical ICP scheme.

In cases where the visual stage fails (e.g. due to too less point correspondences) the visual stage is skipped and the motion estimate from the former frame is used as a crude initial parameter guess for ICP.

5. EXPERIMENTAL EVALUATION

The suggested C2F-ICP was tested on four different datasets as shown in Fig. 8 for four different test objects: (a) Head, (b) Bear, (c) Figure, (d) Castle. The objects were observed from a fixed camera position and rotated by fixed angles in order to generated ground truth motion data. As a sensory unit we used the Baumer TZG01 ToF camera (see Baumer (2010)) for the detailed technical specifications).

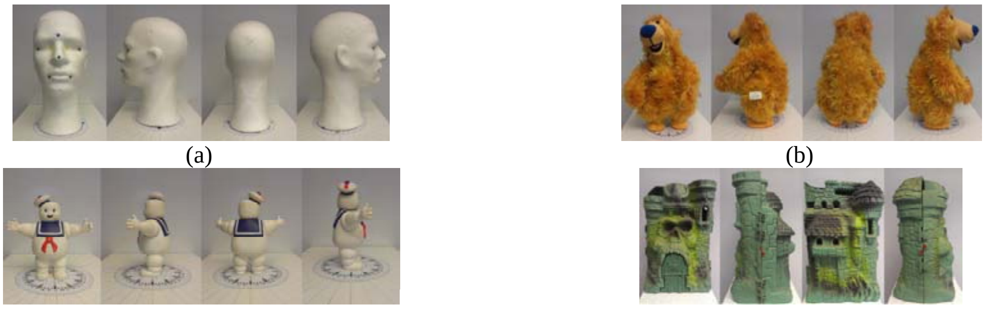

(a) (b)
(c) (d)

Figure 8. Test objects for the experimental evaluation: (a) – Head, (b) – Bear, (c) – Figure, (d) – Castle

Figure 9 gives results for the performance of our approach as the cumulated angle error versus ground truth angle, for the four test objects. We examined four different feature detector methods during the V-PCR stage of the algorithm and compared the results with the classical ICP algorithm.

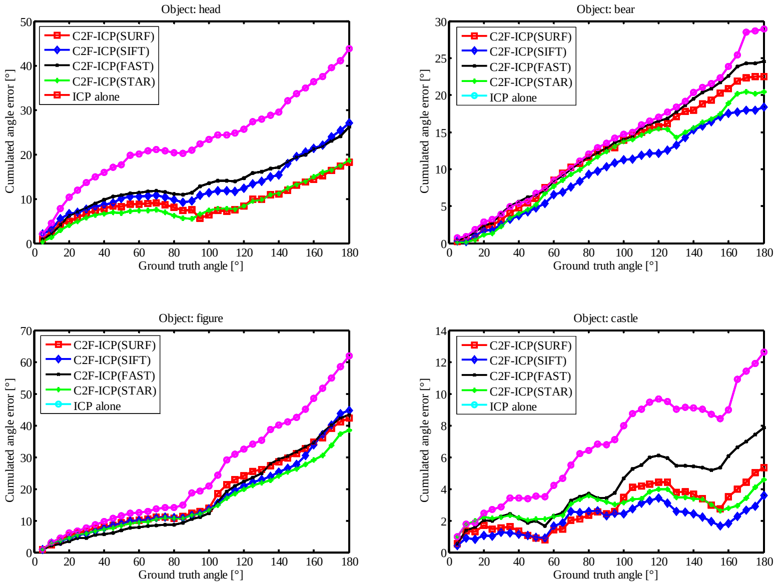


Figure 9. The PCR accuracy of C2F-ICP plotting the cumulated angle error versus ground truth angle for four different feature detector approaches, compared to classical ICP, applied to four different objects (a) Head, (b) Bear, (c) Figure, (d) Castle.

Not only can the accuracy of the rigid transformation be improved by using our technique, but also the computational efficiency can be enhanced. Figure 10 (a) , on the left hand side, shows the total processing time, on standard PC hardware, for a sequence of 36 frames with a frame-to-frame angle difference of 5°, for the castle object. while the right hand side Figure 10 (b) shows the cumulative number of ICP iterations.

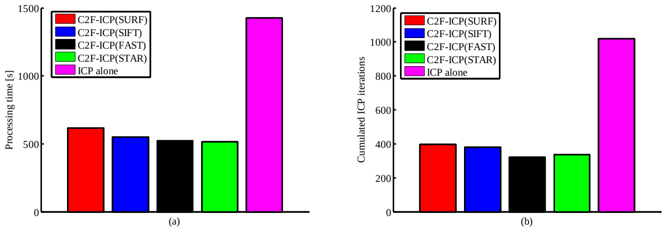


Figure 10. (a) Processing time and (b) cumulative number of iterations for four different feature detectors and classical ICP.

Figure 10 shows that the number of ICP iterations has been reduced considerably by a factor of about a third, when incorporating a reasonable initial transformation from V-PCR and despite the additional processing for the visual stage, leads to overall reduced processing time, also by a factor of about a third.

6. CONCLUSION AND FUTURE WORK

One major drawback of ICP-based PCR is the missing ability to deliver robust results for scenes without distinctive depth structures. Due to the iterative character of the ICP and the high computational costs for the correspondence search, the capability of applying ICP, under real-time conditions, is limited. The computation of reasonable initial transformation parameters for the ICP algorithm is essential for a reasonable run time.

We suggest the incorporation of visual information within a crude-to-fine ICP (C2F-ICP) scheme which utilizes 2D intensity point features, to compute an estimate for a robust guess, of the rigid transformation parameters. We proved our concept by testing it on a set of different objects and our results indicating that both estimation accuracy and computational costs are improved by using C2F-ICP.

In future work, methodologies for an initial simplification of the point clouds will be considered in order to further reduce the computational costs. Point cloud simplification (PCS) may also be a reasonable tool for the estimation of initial motion parameters, where no visual information is available e.g. in low textured scenes.

REFERENCES

Agrawal, M., Konolige, K., and Blas, M.R. 2008, CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching. *In The 10th European Conference on Computer Vision.* pp. 102-115.

Aufderheide, D., Steffens, M., Kieneke, S., Krybus, W., Kohring, C. and Morton, D. 2009. Probabilistic Scene Analysis for Robust Stereo Correspondence. *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR2009)*, pp. 697-706

Aufderheide, D., Krybus, W., 2010. A Framework for Real Time Scene Modeling based on Visual-Inertial Cues. In *Proceedings of the IADIS Computer Graphics, Visualization, Computer Vision and Image Processing Conference 2010 (CGVCVIP 2010),* pp. 385-390.

Arun, K.S., Huang, T.S., and Blodstein, S.D., 1987. Least-Squares Fitting of Two 3-D Point Sets. *In IEEE Transactions of Pattern Analysis and Machine Intelligence*, 9(5), pp. 698-700.

Axelsson, P., 1999. Processing of Laser Scanner Data – Algorithms and Applications. *In ISPRS Jorunal of Photogrammetry and Remote Sensing,* Vol. 54, Issues 2-3, pp. 138-147.

Baumer Corp. 2010. Baumer TZG01 – User’s Guide for Digital 3D Camera.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., 2008. SURF: Speeded Up Robust Features. In Computer Vision and Image Understanding, 110(3), pp. 346-359.

Chen, C.S., 1998. A Fast Automatic Method for Registration of Partially-Overlapping Range Images. *Proceedings of the Sixth International Conference on Computer Vision.* pp. 242-248

Koninckx, T.P., Griesser, A., VanGool, L., 2003. Real-Time Range Scanning of Deformable Surfaces by Adaptively Coded Structured Light. *Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling*. Alberta, Canada, pp. 293-300.

Korth, A., 2013. Real-Time 3D Surface Reconstruction Using a Moving Depth Camera. *Master Thesis*. The University of Bolton / South Westphalia University of Applied Sciences.

Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints, *In International Journal of Computer Vision*, 60(2), pp. 91.110.

Neugebauer, P., 1997. Geometrical Cloning of 3D Objects via Simultaneous Registration of Multiple Range Images.

Proceedings of 1997 International Conference on Shape Modeling and Applications, pp. 130-139.

Newcombe, R., Izadi, S., Hilliges, O., Molyneaux, D., Davison, A., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A., 2011. KinectFusion: Real-Time Dense Surface Mapping and Tracking. *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR) 2011*, Basel, Switzerland, pp. 127-136.

Rosten, E., Porter, R., and Drummond, T., 2010. FASTER and Better: A Machine Learning Approach for Corner Detection. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1), pp. 105-119.

Rusinkiewicz, S., Levoy, M., 2001. Efficient Variants of the ICP Algorithm. *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, pp. 145-152.

Zhang, Z. 1994. Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *In International Journal of Computer Vision*, 13(1), pp. 119-152.

HALLUCINATING FACES IN CURVELET

Xiang Xu, Wanquan Liu and Ling Li
Curtin University - Western Australia

ABSTRACT

In this paper, we aim to enhance the resolution of a single face image. We introduce a method which utilizes the specific features of Curvelet to select training samples and estimate local face features. Based on different characteristics of Curvelet coarse and fine coefficients, we firstly set up training sets for global face enhancement and local features estimation separately. Secondly, global faces are derived by employing sparse representation technique. Thirdly, we transfer the low-resolution local features into Curvelet frequency domain and infer the relationship of Curvelet coefficients between testing images and training images. Through the learning process, the Curvelet coefficients of the high-resolution local features can be derived. Finally, the high-resolution local features are generated through Inverse Discrete Curvelet Transformation, which are then combined with the global face to produce the final hallucinated face. Experiment demonstrates that our approach outperforms other approaches.

KEYWORDS

Face Hallucinating, Curvelet.

1. INTRODUCTION

Faces from video frames are usually too small to be recognized. In order to increase the recognition rate of those low-resolution images, many face resolution enhancement techniques have been developed. Baker and Kanade (2000) firstly presented the idea of specified face super-resolution technique. Wang and Tang (2005) introduced the Eigen-transformation algorithm at a relatively small computational cost. Liu et al. (2007) proposed a two-step face hallucination algorithm, who firstly introduced the residue concept to face hallucination. In order to reduce the computational cost, Zhuang et al. (2007) rendered the residue part by applying the Nearest Neighbor algorithm. Recently, Yang et al. (2010) proposed a method by combining Non-negative Matrix Factorization with sparse representation algorithms. Zhang and Cham (2011) proposed a face hallucinating algorithm in frequency domain instead of the conventional spatial domain. They transferred the low-resolution face images into Discrete Cosine Transformation coefficients and inferred the high-resolution coefficients through utilizing Markov Random Field. Then, the expected high-resolution face images can be acquired by adopting the inverse Discrete Cosine Transformation.

In recent years, most face hallucination techniques are proposed in spatial domain, which often require a large amount of computation. This is due to the large number of training data. In terms of frequency domain based super-resolution techniques, though they are efficient, they can hardly represent detailed facial features without the learning process Milanfar (2010). In order to synthesize the advantages of the algorithms in both spatial and frequency domains, we propose a two-step face hallucination method combined with pre-selection processes based on Curvelet features. As we know from Liu et al. (2007), in face hallucination, face images include two types of features: global features and local features. Global features describe the common human features like eyes, mouths and noses. The local features represent the specific features of an individual face image. However, in traditional two step approach, global features and local features cannot represent hallucinated global faces and local residues explicitly. For instance, hallucinated global faces often contain the detailed face features. Instead, we adopt the Curvelet frequency features to describe those two types of features in this paper:

Feature 1 *Global face features which include most of the low-frequencies of human faces;*

Feature 2 *Local face features which consist of the high-frequencies in face images.*

With these two features, we design our learning based face hallucination method in two steps. One is the low-frequency face image hallucination and the other is the high frequency based face image hallucination.

In order to reduce computational cost and reconstruction errors, we use the Curvelet features of testing images to select the training samples in both two steps. First, we decompose the low and high resolution image pairs into Curvelet features which are used to select training data. In fact, the fine Curvelet coefficients describe the high frequency components of face images, and the coarse Curvelet coefficients represent the low frequency part of face images. In order to reduce computational complexity, we only use two layers of Curvelet coefficients in this paper. Now for each image, we have both the fine and the coarse coefficients. Then we use K_{th} Nearest Neighbours algorithm to find K_1 images, which have the best matched coarse coefficients with the coarse coefficients of testing face image. Similarly, we also can find K_2 images which have the best matched fine coefficients compared with the fine coefficients of testing image. In this paper, we use the K_1 images as the training samples of first step and the K_2 images as the training set of second step.

In the first step, we estimate the high resolution global features for a low resolution testing face image using the sparse representation learning method. The examples of low resolution images are shown in the first column of Fig 4. In the second step, we produce a residue training data, and estimate the high resolution residue which compensates the missing local features for the global face in the first step. Through learning the Curvelet features of the residue training pairs, we estimate the Curvelet features of the high-resolution residue face and achieve it through the Inverse Curvelet transformation.

Our main contributions of paper have three parts.

1. We extract two types of features based on Curvelet frequency domain: low-frequency part, which represents the global features of human faces; high-frequency part, which demonstrates the local features of human faces.

2. We use the Curvelet features to select training samples for both global and local hallucination algorithms, which reduces computational cost due to the smaller training data.

3. In high frequency feature estimation, we treat each residue as an image and hallucinate this residue image through inverse Curvelet transformation in Curvelet frequency domain.

The paper is organized as follows: In Sec. 2.1 we extract image features in Curvelet frequency domain and select training samples based on global and local features respectively. Then we hallucinate the low-resolution faces to global high-resolution faces by employing the sparse representation algorithm in Sec. 2.2. In Sec. 2.3, the residual faces are derived from the Inverse Discrete Curvelet Transformation. Experimental results are illustrated in Sec. 3 with comparison with other approaches. Conclusion and future work are stated in Sec. 4.

2. PROPOSED ALGORITHM

2.1 Curvelet Based Training Sample Selection

Curvelet was first proposed by Candès and Donoho (1999), and then developed to the second generation by Candes et al. (2006), which is both fast and accurate. Previous work from Mandal et al. (2009) has proved that Curvelet can ideally extract the human face features and address the face recognition problem. For a 2D image, Curvelet transformation from Candes et al. (2006) is performed as follows:

1. Apply the 2D FFT and obtain Fourier samples $\hat{f}[n_1,n_2]$ of $f[t_1,t_2]$,where $0 \le t_1,t_2 \le n, -n/2 \le n_1,n_2 < n/2$.

2. For each scale j and angle l , compute the product $\tilde{U}_{j,l}[n_1,n_2]\hat{f}[n_1,n_2]$, where $\tilde{U}_{j,l}[n_1,n_2]$ is the discrete localizing window.

3. Wrap this product around the origin and obtain $\tilde{f}_{j,l}[n_1,n_2] = W(\tilde{U}_{j,l}\hat{f}[n_1,n_2])$, where W is the wrapping function.

4. Apply the inverse 2D FFT to each $\tilde{f}_{j,l}$ and collect the discrete Curvelet coefficients $C(j,l,k)$.

In this paper, we use Curvelet features to select training samples. First it is used to select the training samples with global features. Here we select training samples from the face data sets through Curvelet coefficients before hallucinating global faces. We transfer the low-resolution testing image and the training images into Curvelet domain, where we calculate the nearest neighbours in the low-resolution coarse coefficients of the testing image. Then, those images whose coarse coefficients are the nearest neighbours of the coefficients of the testing image are selected as the training samples.

Specifically, let $A_i^h = [A_1^h, A_2^h, \cdots, A_n^h, \cdots]$ denote the high resolution face database, and $A_i^l = [A_1^l, A_2^l, \cdots, A_n^l, \cdots]$ denote the corresponding low resolution face database. We first decompose the i_{th} low resolution face image into Curvelet features $C_i\{j\}\{t\}(k_1, k_2)(i = 1,2,\cdots,n,\cdots)$. Here j and t represent the scales and angles of Curvelet coefficients respectively. k_1, k_2 indicate the coefficient matrix positions. We simplify $C_i\{j\}\{t\}(k_1, k_2)$ to be $C_i\{j\}\{t\}$ in the left part of this paper. When a test low resolution image x comes, it is decomposed to Curvelet domain to get the coefficients $C_x\{j\}\{t\}$. In order to reduce computational cost, we set scale $j = 2$, angle $l = 8$ in this paper. Once the Curvelet coefficients $C_x\{j\}\{t\}$ are derived, the coarsest coefficient $C_x\{1\}\{t\}$ $(t = 1)$ represents the low frequency feature and the finest coefficients $C_x\{2\}\{t\}$ $(t = 1,2,\cdots,8)$ represent the high frequency feature. Figure 1 shows the Curvelet coefficients of a testing image x. For the convenience of display, all the feature images are resized with the same resolution.

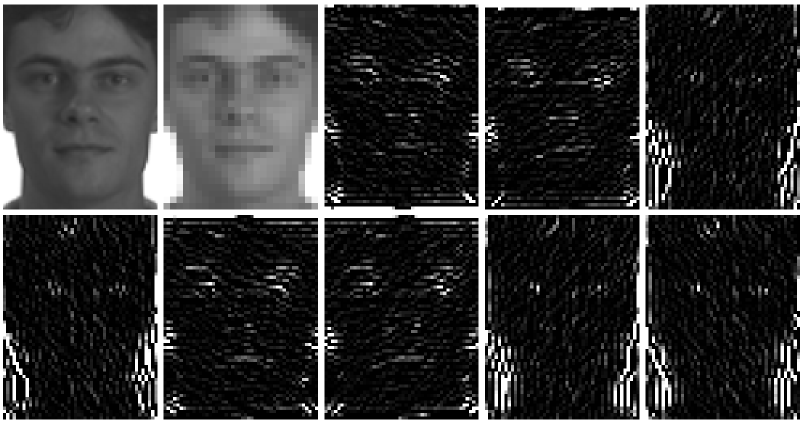


Figure 1. Curvelet Coefficients. The top left is the original face image. The left 9 images are the low-frequency image of the first layer and the 8 high-frequency images of the second layer respectively.

Now we have a set of Curvelet coefficients $C_i\{j\}\{t\}(i = 1,2,\cdots,n,\cdots)$ for the face database and $C_x\{j\}\{t\}$ for the testing face. We first utilize the low frequency feature $C_x\{1\}\{t\}$ and calculate its K_1 nearest neighbours in $C_i\{1\}\{t\}(i = 1,2,\cdots,n,\cdots)$. For computational efficiency, we first adopt the Principal Component Analysis (PCA) to reduce the dimension and then apply the nearest neighbour algorithm in K_1 elements selection. Consequently, we have K_1 selected training samples for the global face enhancement, named as $I = [I_1, I_2, \cdots, I_{K_1}]$.

Similarly, for high frequency components, we also use the local features to select another set of training samples. In order to achieve this, we treat the whole second scale of $C_x\{2\}\{t\}$ as one image and resize all the coefficients to form one column. This one column image represents the high frequency features through t different angles ($t = 8$). Similarly, we also adopt PCA for dimension reduction. We keep 20 eigenvectors corresponding to 20 largest eigenvalues, which keeps the most of the data energy. After that we select K_2 nearest neighbours in $C_i\{2\}\{t\}(i = 1,2,\cdots,n,\cdots)$ as local feature training samples, namely $F = [F_1, F_2, \cdots, F_{K_2}]$.

2.2 Hallucinating Faces via Sparse Representation

In sparse sensing theory, Donoho (2006) proved signals can be represented by basis signals through a well-constructed dictionary. If we think of face images as a kind of signal, one face image can be represented by a set of face basis when there is a large training data set. A generic image sparse representation algorithm from Yang et al. (2008) has been proposed, which estimate the high-resolution image from raw image patches. However, in reality the quality of face super-resolution depends on how well the dictionary is designed. In this paper, we design the training dictionary through the Curvelet features based selection in previous section.

Suppose the selected training set in Sec. 2.1 to be $D_h(high-resolution)$ and $D_l(low-resolution)$, a test low-resolution face image x can be reconstructed to a high-resolution global face as follows: We first solve the following equations:

$$\begin{aligned} \hat{\alpha} &= argmin \ \|\alpha\|_{l_1} \\ s.t. \quad D_l\alpha &= x \end{aligned} \qquad (1)$$

where α is the sparse representation coefficients in l_1 norm.

This sparse representation coefficient α in low-resolution face images can then be mapped to the high-resolution. The global high-resolution face can be reconstructed as:

$$y = D_h\hat{\alpha} \qquad (2)$$

In practice, the sparse representation does not perform well when we treat the whole face as one signal. Following the idea in Yang et al. (2010), we first divide face images into overlapped patches and enhance those patches respectively. Then by combining those patches, we can derive the high-resolution faces. In this paper the size of each patch is set as 4×4 in low-resolution images and 16×16 in high-resolution images. The overlapped size is set as 3 and 12 respectively. When merging the patches, the values of overlapped pixels in the high-resolution face are the average values of pixels in the same position.

2.3 Residue Face Enhancement

A global face only represents the general human features of one person, thus some detailed features might be lost. Residue face enhancement is required for this reason Liu et al. (2007). We treat the residual face to be the special features towards each individual. For each person, this residue should be unique. In this paper, we develop a learning process in frequency subspace.

More precisely, we have obtained a globally hallucinated face y in previous section. As we only have a testing image x, which is in low-resolution, we first down sample y to low-resolution, and derive its residue image s:

$$s = x - Down(y) \qquad (3)$$

where $Down$ represents the downsample function. The down sample rate is set to be 4 in this paper.

This residue image is thought of as the local features of the test image x. However, s only represents the low resolution local features and we need its high resolution local features s^h to render the globally hallucinated face y. In this section we derive this s^h through a frequency domain learning process. Since we have selected a training set $F=[F_1,F_2,\cdots,F_{K_2}]$, which are based on the local features of face images in previous section, we need to derive a high resolution residue training set $R^h=[R_1^h,R_2^h,\cdots,R_{K_2}^h]$ from F. This high resolution residue training set should represent the high frequency local features of human faces. For such purpose, we first down-sample F to low resolution, then enhance them to the high resolution image set $\widetilde{F}=[\widetilde{F}_1,\widetilde{F}_2,\cdots,\widetilde{F}_{K_2}]$. As a result, R^h can be derived as follows:

$$R_{(i)}^h = F_{(i)} - \widetilde{F}_{(i)} \qquad (4)$$

where $i=(1,2,\cdots,K_2)$.

In order to reduce the computational costs, we do not derive $\widetilde{F}$ through sparse representation enhancement for each training image. Instead, we adopt the Bicubic interpolation Hou and Andrews (1978) to obtain a smooth high resolution training set $\widetilde{F}$. R^h represent the high frequency components of the selected training images, which are then down sampled to the low-resolution version $R^l = [R_1^l, R_2^l, \cdots, R_{K_2}^l]$. Now we have a testing low resolution image s and a set of training samples R^h and R^l. We first decompose both s, R^l and R^h into Curvelet subspace and derive the corresponding Curvelet coefficients. Let $C_s^l\{j\}\{t\}$ denote the coefficients of s, $C_{r(i)}^l\{j\}\{t\})$ denote the coefficients of $R_{(i)}^l$ and $C_{r(i)}^h\{j\}\{t\}$ denote the coefficients of $R_{(i)}^h$ $(i = 1,2,\cdots,K_2)$, we formulate an optimization problem to obtain $C_s^h\{j\}\{t\}$, which is the Curvelet coefficients of s^h. For each element in $C_s^l\{j\}\{t\}$, which is a matrix, there is a corresponding matrix in $C_{r(i)}^l\{j\}\{t\})$ and $C_{r(i)}^h\{j\}\{t\}$. Here, we first formulate the following least square problem:

$$\begin{gathered} \underset{W_i}{argmin} \left\| C_s^l\{j\}\{t\} - \sum_{i=1}^{k_2} W_i C_{r(i)}^l\{j\}\{t\} \right\|_2^2 \\ s.t. \sum_{i=1}^{K_2} W_i = 1 \end{gathered} \qquad (5)$$

where $(i = 1,2,\cdots,K_2; j = 1,2;, t = 1,2,\cdots,8)$.

Then for each $C_s^l\{j\}\{t\}$, we can derive a set of W_i. The corresponding $C_s^h\{j\}\{t\}$ can be derived through:

$$C_s^h\{j\}\{t\} = \sum_{i=1}^{K_2} W_i C_{r(i)}^h\{j\}\{t\} \qquad (6)$$

We now have the complete coefficients $C_s^h\{j\}\{t\}$ of s^h, Through using the Inverse Discrete Curvelet Transformation (IDCT) in $C_s^h\{j\}\{t\}$, we can obtain the high frequency feature s^h. Finally we can derive:

$$y_f = y + s^h \qquad (7)$$

3. EXPERIMENTAL RESULTS

In this paper, we use FERET face database from Phillips et al. (2000) and CASPEARL database from Gao et al. (2008) to implement our approach. The FERET database includes 839 individuals and each individual has 2 to 10 images. We choose 239 people as testing samples and the other 600 as training samples. CASPEARL database has 1040 individuals and we choose 440 for testing and the remain 600 for training. In our experiment only one frontal image is collected for each person. Before the experiment, we first align all the images manually and crop the faces by fixing the centers of the eyes and the mouths. The size of the cropped high-resolution image is set as 128×96 and that of the low-resolution image is set as 32×24.

There are two steps in our experiment. Before each step, there is a pre-selection process, which is designed to locate a set of proper training samples to achieve a better performance compared with randomly selected training samples in the hallucinating process. Then in the first step, we try to construct a smooth, high-resolution image with global features. In other words, we want to construct a high-resolution face image with low frequency information based on the coarse coefficients in the Curvelet domain. Since we construct a high-resolution global face with low frequency information, we need to find the local features of each individual. These local features are also called residues, located in the high frequency domain. For this reason, we aim to find the high frequency information through the fine coefficients in Curvelet domain.

In summary we perform four types of experiments in our approach.

Experiment 1: Through randomly selecting K_1 training samples, we perform sparse representation super-resolution algorithm (Eq. 1).

Experiment 2: Through randomly selecting K_1 training samples, we first perform sparse representation super-resolution algorithm (Eq. 1), then perform Curvelet residual compensation (Eq. 5) based on those K_1 training samples.

Experiment 3: We first adopt our Curvelet based pre-selection algorithm to select K_1 training samples, and then perform sparse representation super-resolution algorithm (Eq. 1).

Experiment 4: We first adopt our Curvelet based pre-selection algorithm to select K_1 and K_2 training samples, then then perform sparse representation super-resolution algorithm (Eq. 1). And finally obtain the final results by combining our Curvelet residual compensation approach (Eq. 5).

Comparisons in terms of Peak Signal Noise Ratio (PSNR) and Root Mean Square Error (RMSE) of our four experiments can be found in (e), (f), (g) and (h) of Table 1 and Fig. 2, where $K_1 = 30$ and $K_2 = 20$. In Table 1, we show the PSNR values of randomly picked 6 training samples and the average results of the whole 679 testing samples in each column. It can be seen from Table 1 that when combined with sparse representation technique separately, both our Curvelet Residual compensation approach (Experiment 2) and pre-selection approach(Experiment 3) can improve the results of generic sparse representation method (Experiment 1). A more significant improvement can be seen in our final result (Experiment 4), where we adopt both our pre-selection approach and Curvelet residual compensation approach. Similarly, Fig. 2 demonstrates the average RMSE values of our 679 testing samples, where both our Curvelet Residual compensation approach and pre-selection approach can reduce the errors, no matter they are used separately (Experiment 2 and 3) or together (Experiment 4).

Table 1. PSNR Comparison of six randomly selected images and the average values of 679 Testing Samples. (a) Sparse Representation approach. Yang et al. (2010). (b) LPH super-resolution and neighbor reconstruction. Zhuang et al. (2007). (c) Eigen-Transformation hallucination. Wang and Tang (2005). (d) A two-step face hallucination. Liu et al. (2007). (e) Experiment 1. (f) Sparse Representation combined with our Curvelet residual compensation approach (Experiment 2). (g) Sparse Representation combined with our pre-selection approach (Experiment 3). (h) Our final approach (Experiment 4). Average demonstrates the average PSNR result of the whole 679 testing images.

Images	1	2	3	4	5	6	Average
(a)	28.69	29.02	30.68	27.75	29.26	31.59	29.23
(b)	24.72	22.47	26.47	21.23	24.32	25.54	24.16
(c)	21.44	18.35	23.23	15.81	24.79	20.24	21.92
(d)	25.79	27.06	28.36	24.45	27.85	29.54	26.96
(e)	28.01	28.06	29.36	27.45	28.85	31.54	28.96
(f)	30.89	30.47	31.55	29.74	30.01	32.27	30.03
(g)	30.85	30.64	31.49	29.9	29.48	32.02	29.97
(h)	31.75	31.4	32.22	30.49	31.24	32.41	30.63

In order to clarify the influence of training samples, we perform our experiment when K_1 is set as 5, 10, 20, 30, 50, 100, 200, 300, 400, 500 and 600 respectively in Experiment 4 $(K_2 = 20)$. Figure 3 describes the average PSNR values in term of different training samples (K_1) in Experiment 4. It can be seen that our approach does not depend too much on the number of training samples. It performs quite well even the number of training samples is small.

Combining pre-selection approach and Curvelet residual compensation, our approach utilizes the advantages in both spatial domain and frequency domain. Figure 4 indicates the comparison between our approach $(K_1 = 30, K_2 = 20)$ and other four typical methods when the number of training data is 200. It can be identified that our results have much smoother and clearer views even in a smaller training data. Especially when compared with those who adopt holistic face as training samples, our method has less noises around the chin area.

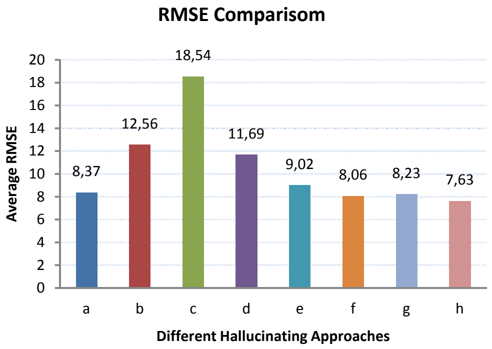


Figure 2. Average RMSE of different Hallucinating Approaches. (a) Yang et al. (2010). (b)Zhuang et al. (2007). (c)Wang and Tang (2005). (d) Liu et al. (2007). (e) Experiment 1. (f) Experiment 2. (g) Experiment 3. (h) Experiment 4.

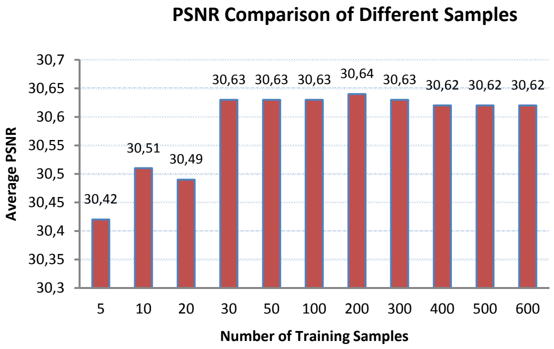


Figure 3. PSNR of Our Approach in terms of Different Training Sample $K_{1.}$

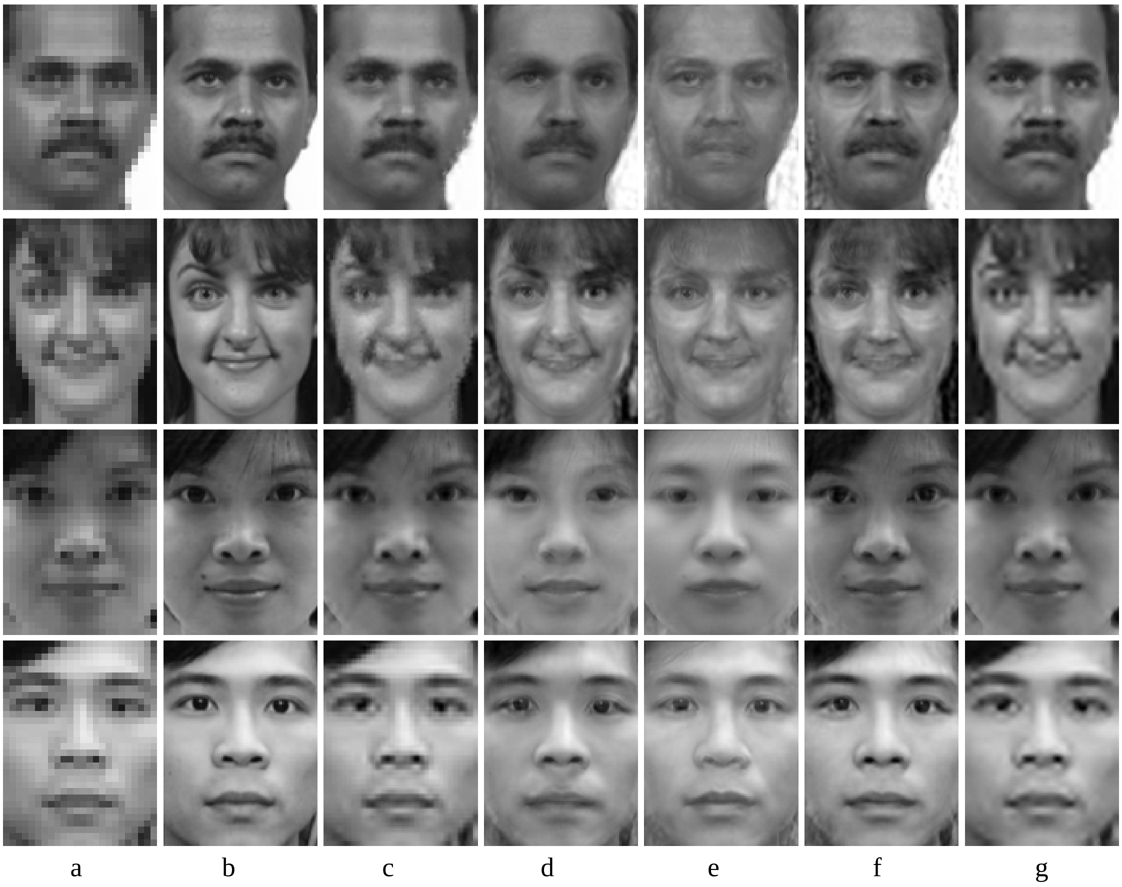


Figure 4. Comparison of different algrithms. (a) Low-resolution. (b) Original high-resolution. (c) Yang et al. (2010). (d) Zhuang et al. (2007). (e) Wang and Tang (2005). (f) Liu et al. (2007). (g) Our approach.

The comparison of different methods in terms of Peak Signal Noise Ratio (PSNR) is shown in Table 1, which includes six randomly selected people in both two databases. And the last column shows the average results of the whole experiment (239 people in FERET and 440 in CASPEAERL). Face hallucination results of Yang et al. (2010), Zhuang et al. (2007), Wang and Tang (2005) and Liu et al. (2007) are shown in (a),(b),(c) and (d) respectively. Figure 2 describes the average Root Mean Square Errors of the above methods as well. From these comparisons, one can see that our approach outperforms other existing approaches.

4. CONCLUSION

In this paper, a Curvelet feature based face hallucination approach is proposed. We first select the training samples according to the Curvelet coefficients of the testing low-resolution faces. Secondly, we use the general sparse representation idea to reconstruct the global face based on the selected training samples. Compared with general sparse representation method, this pre-selected training samples can can help improve the hallucination results. Residue compensation is then carried out. Since the residues can be thought of as the high frequency information of the face images, we select the residue training samples through locating the nearest neighbors of the high frequency coefficients in Curvelet domain. Through the learning process, the Curvelet coefficients of the high-resolution residue images are estimated. The high-resolution residue image can be derived through employing Inverse Discrete Curvelet Transformation. Finally, by combining the global face with the residue, we can derive the final high-resolution face.

Most of the face hallucination methods are based on the face recognition techniques, but not all these recognition techniques are applicable for face resolution enhancement. Therefore, face hallucination techniques should differ from the existing recognition algorithms and have its own characteristics which will be the core content of our future work.

REFERENCES

Baker, S. and Kanade, T., 2000 Hallucinating faces. *Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition.* pp. 83-88.

Candes, E., et al., 2006. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, Vol. 5**,** No. 3, pp. 861-899.

Candès, E. and Donoho, D. 1999. Curvelet: A surprising effective non-adaptive representation for objects with edges. Department of Statistics, Stanford University: Technical Report 1999-28.

Gao, W., et al., 2008. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 38**,** No. 1, pp. 149-161.

Hou, H. and Andrews, H., 1978. Cubic splines for image interpolation and digital filtering. *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 26**,** No. 6, pp. 508-517.

Liu, C., et al., 2007. Face hallucination: Theory and practice. *International Journal of Computer Vision*, Vol. 75**,** No. 1, pp. 115-134.

Mandal, T., et al., 2009. Curvelet based face recognition via dimension reduction. *Signal Processing*, Vol. 89**,** No. 12, pp. 2345-2353.

Milanfar, P., 2010. *Super-resolution imaging.* CRC Press.

Phillips, P. J., et al., 2000. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22**,** No. 10, pp. 1090-1104.

Wang, X. and Tang, X., 2005. Hallucinating face by eigentransformation. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 35**,** No. 3, pp. 425-434.

Yang, J., et al., 2008 Image super-resolution as sparse representation of raw image patches. *IEEE Conference on Computer Vision and Pattern Recognition.* pp. 1-8.

Yang, J., et al., 2010. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, Vol. 19**,** No. 11, pp. 2861-2873.

Zhang, W. and Cham, W. K., 2011. Hallucinating Face in the DCT Domain. *IEEE Transactions on Image Processing*, Vol. 20**,** No. 10, pp. 2769-2779.

Zhuang, Y., et al., 2007. Hallucinating faces: LPH super-resolution and neighbor reconstruction for residue compensation. *Pattern Recognition*, Vol. 40**,** No. 11, pp. 3178-3194.

SINEFITTING: ROBUST CURVATURE ESTIMATION ON SURFACE TRIANGULATION

Jérôme Charton, Stefka Gueorguieva and Pascal Desbarats
LaBRI (UMR 5800), Université Bordeaux 1 - 351, Cours de la Libération, F-33405 Talence cedex

ABSTRACT

A novel algorithm for robust curvature estimation based on sinusoidal curve fitting is proposed. The evaluation of this algorithm is presented on analytical surface triangulations by comparing it with other recognized fitting methods according to three criteria: convergence, precision and robustness. By experimenting on various data, we show that the *Sinefitting* algorithm is less affected by errors in vertex normal estimation.

KEYWORDS

Triangular meshes, generalized curvature estimation, surface fitting, curvature measures

1. RELATED WORK

Knowledge about the geometric shape of an object is based on the understanding of the differential structure of the object boundary surface: the principal curvatures and directions, the Gauss and the Mean curvatures of the boundary surface. We are interested in estimating this differential structure of the underlying smooth surface from a given triangulation. Since the pioneer works of [1, 21, 7], curvature estimation is a central issue in a great number of research. For surveys we refer the reader to [15, 19, 23]. Generalized curvatures, convergence and measure stability are covered by [12, 18, 3]. The approaches considered in this paper are based on the fitting paradigm: first, a quadratic form is constructed at each vertex of the surface triangulation and then the local differential structure is derived. Chen et al [4] use the normal curvature as stated by the Meusnier and Euler's theorem and locally fit a set of circles through the neighbour vertices. McIvor et al [16] use a quadratic surface fitting for determining the principal frame and the rotated principal quadric. Different variants of this method are applied depending on the type of the quadratic form (simple SQFA, extended or full). More generally, Cazals et al [2] make use of osculating jets defined as truncated Taylor expansions. For Taubin [24] the quadratic form is expressed as an integral representation that is used to obtain the curvature tensor. Langer et al [14] use integral representations of the Gauss and the Mean curvatures. Cohen-Steiner et al [5] use integrals of specific differential forms on their normal cycles. Fitting enables precise estimation of curvatures but it is very sensitive to the surface discretization and to the distribution of the edge directions in the vertex neighbourhood. Indeed, these methods make use of the surface normal approximation which strongly depends on the regularity of the triangles in this neighbourhood.

In the current paper, a novel method for curvature estimation, called *SineFitting*, is presented. At each vertex of the surface we construct a sinusoidal curve to fit the directional angles from the target vertex to its neighbours. Therefore, in contrast to Chen's method, there is no need to restrict the choice of neighbours to specific configurations (pairs of geometrically opposite vertices). Moreover, computations in under or oversampled neighbourhoods remain robust while maintaining high precision. The elaborated method has a linear convergence rate and is not acutely affected by errors in estimation of the normal.

2. OVERVIEW OF THE SINEFITTING METHOD

We refer the reader to [22] for detailed discussion on surface differential geometry. Let S be a surface, P a target vertex from S and $\vec{N}$ the unit normal vector to S in P. Let τ be the tangent plane of S in P, see Fig.1.

Given a unit vector $\vec{T}$ in τ, the osculating plane ι through $\vec{N}$ and $\vec{T}$ intersects *S* in a curve *c*. Let $\vec{n}$ be the unit normal vector of *c* in *P* and ϑ the angle between $\vec{n}$ and $\vec{N}$, $\vartheta = (\vec{n},\vec{N})$.

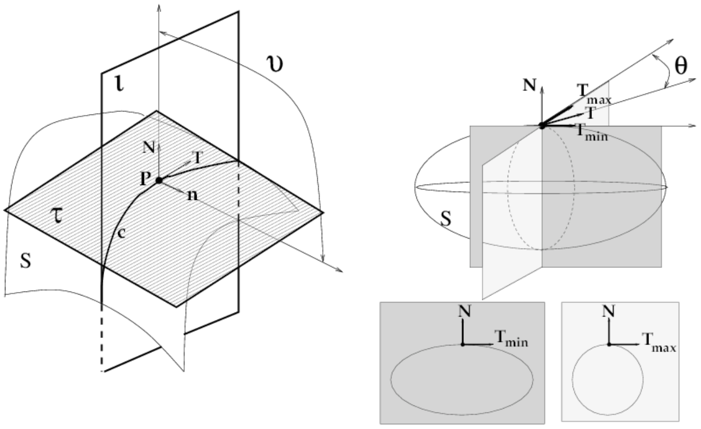


Figure 1. Local surface geometry around point *P*

According to Meusnier & Euler's theorem, the normal curvature k_T of *S* in *P* along $\vec{T}$ could be defined as

$$k_T = k.\cos(\vartheta) \quad (1)$$

where *k* denotes the curvature of *c* in *P* and $\cos(\vartheta) = \vec{n}\cdot\vec{N}$. While $\vec{T}$ is rotated around $\vec{N}$, an infinite number of normal curvatures k_T could be defined. The extreme values of k_T, k_{min} and k_{max}, are achieved along the principal directions, $\overrightarrow{T_{min}}$ and $\overrightarrow{T_{max}}$. Euler's theorem gives the relation between k_T, k_{min}, k_{max} and the angle θ, $\theta = \angle(\vec{T},\overrightarrow{T_{max}})$.

$$k_T = k_{max}{\cdot}cos^2(\theta) + k_{min}{\cdot}sin^2(\theta) \quad (2)$$

The proposed algorithm, called *SineFitting* algorithm, is based on (1) and (2) in order to evaluate the principal directions $\overrightarrow{T_{min}}$ and $\overrightarrow{T_{max}}$, the principal curvatures k_{min} and k_{max}, the Gauss and the Mean curvatures, k_G and k_H , in *P*. Our approach is based on a two steps procedure: First, the normal curvature k_T is evaluated according to (1) for curves locally fitting the normal sections of the surface. Second, a sinusoidal curve is constructed to approximate the computed values of k_T following (2). Principal curvatures and directions are calculated for specified values of the sinusoidal amplitude and frequency. Let T_S be a triangulation of *S*, *N(P)* a neighbourhood of *P*, $N=\{P_i,\ PP_i \in T_S,\ i=0,\ldots,m\text{-}1\}$ and k_i the normal curvature of *S* along PP_i. Let P_i^* be the projection of P_i on τ, M_i the middle point of PP_i and M_i the median of PP_i, shown in Fig.1. Let us construct a circle σ passing through *P* and P_i, and centred at O_i, $O_i = l_{\vec{N}} \cap m_i$, where $l_{\vec{N}}$ is the straight line supporting $\vec{N}$. Let $\vec{n}$ be the unit normal of σ, $\overrightarrow{n_\sigma} = \dfrac{\overrightarrow{PO_i}}{\left\|\overrightarrow{PO_i}\right\|}$, and k_σ be the normal curvature of σ, $k_\sigma = \dfrac{1}{\left\|\overrightarrow{PO_i}\right\|}$.

Let us now apply (1) to k_i and k_σ:

$$k_i = k_\sigma.\cos(\angle(\vec{N},\vec{n})) \quad (3)$$

The unit vectors $\vec{N}$ and $\vec{n}$ are aligned with opposite directions and thus $\cos(\angle(\vec{N},\vec{n})) = -1$. The curvature k_i can be expressed then as:

$$k_i = -k_\sigma = \frac{-1}{\left\|\overrightarrow{PO_i}\right\|} \quad (4)$$

Let ψ be the angle $\angle(M_i, P, O_i)$. From the right-angled triangle $\Delta(M_i, P, O_i)$ it follows that

$$\cos(\psi)=\frac{\left\|\overrightarrow{PM_i}\right\|}{\left\|\overrightarrow{PO_i}\right\|}=2\cdot\frac{\left\|\overrightarrow{PP_i}\right\|}{\left\|\overrightarrow{PO_i}\right\|}\quad(5)$$

The value of *cos(ψ)* can be estimated from the scalar product of $-\vec{N}$ and $\overrightarrow{PP_i}$ and by the substitution in (4) we finally compute the estimation of k_i:

$$k_i=\frac{2\cdot\vec{N}\cdot\overrightarrow{PP_i}}{\left\|\overrightarrow{PP_i}\right\|^2}\quad(6)$$

It should be noticed that the estimation (6) is similar to the estimation in [24]. The above construction gives a geometrical insight of the normal curvature estimation.

The second step of our algorithm includes the construction of a sinusoidal curve that approximates the normal curvature in *N(P)*. Our goal is to rise the constraint of using $\overrightarrow{T_{min}}$ and $\overrightarrow{T_{max}}$ in (2), and fit a linear sinusoidal expression. According to (2) we have:

$$k_i=k_{max}\cdot cos^2(\theta_i)+k_{min}\cdot sin^2(\theta_i)\quad(7)$$

where $\theta_i=\angle(\vec{T_i},\overrightarrow{T_{max}})$. We introduce the angle ϕ

$$\phi=\angle(\vec{T_0},\ \overrightarrow{T_{max}})\quad(8)$$

where $\vec{T_0}$ is any vector from $N^*(P)$, $N^*(P)=\{PP_i^*,\ i=0,\ ...,\ m-1\}$. In (7), θ_i is substituted by α_i, $\alpha_i=\theta_i-\phi$. Then (7) is rewritten as:

$$k_i=k_{max}\cdot cos^2(\alpha_i+\phi)+k_{min}\cdot sin^2(\alpha_i+\phi)\quad(9)$$

Next we substitute ϕ, k_{min} and k_{max} as:

$$\phi=-\frac{\arctan\left(\frac{b}{a}\right)}{2},a>0\quad(10)\qquad k_{min}=c-\sqrt{(a^2+b^2)}\quad(11)\qquad k_{max}=c+\sqrt{(a^2+b^2)}\quad(12)$$

The equation (9) is rewritten as:

$$k_i=a\cdot\cos(2\alpha_i)+b\cdot\sin(2\alpha_i)+c\quad(13)$$

Now we are ready to compute a sinusoidal curve to approximate in a least square way the estimated from (6) normal curvatures. The intuition of this approach is to first estimate a sine wave shape of arbitrary phase. The peaks of its deviation will correspond to the values of k_{min} and k_{max}. The principal directions $\overrightarrow{T_{min}}$ and $\overrightarrow{T_{max}}$ will correspond to the phases of the leftmost peaks in relation to the origin.

The *SineFitting* algorithm is given in Algo. 1

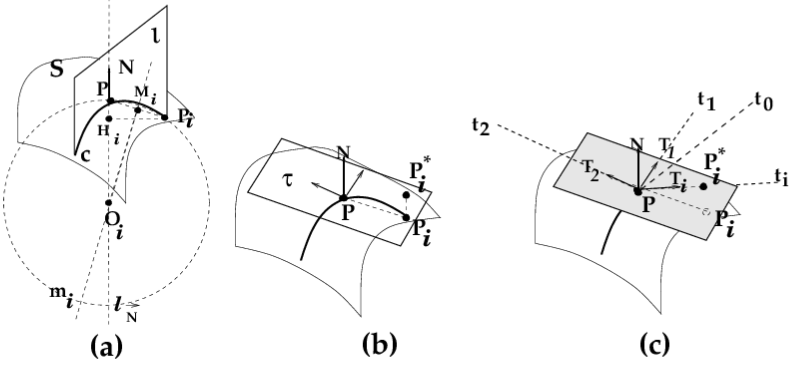


Figure 2. *SineFitting* curvature estimation over one ring neighbourhood

Algorithm 1 *SineFitting*

```
Require: P and $T_S$
Ensure: $\overrightarrow{T_{min}}$ and $\overrightarrow{T_{max}}$, $k_{min}$ and $k_{max}$, $k_G$ and $k_H$ at P
N(P) ← extract neighbour vertices $P_i$ of P
   Let m=|N(P)|, the number of neighbours
N ← the unit normal vector in P
τ ← the tangent plane in P with normal $\overrightarrow{N}$
for $i = 1$ to $m-1$ do
    $P_i^*$ ← projection of $P_i$ on τ
    $\overrightarrow{T_i} = \frac{\overrightarrow{PP_i^*}}{\left\|\overrightarrow{PP_i^*}\right\|}$
end for
for i=1 to m-1 do
    $\alpha_i = arccos\left(\frac{\overrightarrow{T_0}\cdot\overrightarrow{T_i}}{\left\|\overrightarrow{T_0}\right\|\cdot\left\|\overrightarrow{T_i}\right\|}\right)$
    $k_i^* = 2\cdot\frac{\overrightarrow{P_iP}\cdot\overrightarrow{N}}{\overrightarrow{P_iP}}$ {estimation of the normal curvature along $\overrightarrow{P_iP}(6)$}
end for
Least square sinusoidal fitting to calculate a , b and c such that
    $\begin{bmatrix} \cos(2\alpha_1) & \sin(2\alpha_1) & 1 \\ \cos(2\alpha_2) & \sin(2\alpha_2) & 1 \\ \ldots & \ldots & 1 \\ \cos(2\alpha_m) & \sin(2\alpha_m) & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} k_1^* \\ k_2^* \\ \ldots \\ k_m^* \end{bmatrix}$
Calculate $k_{min}$ (11), $k_{max}$ (12)
Calculate $\overrightarrow{T_{max}}$ (8) & (10)
Calculate $k_G = k_{min}\cdot k_{max}, k_H = \frac{k_{min}+k_{max}}{2} = c$
```

3. EXPERIMENTAL SETUP

The discussion below is detailed on three particular examples, a sphere, S_{sphere}, a right circular cylinder, $S_{cylinder}$ and a trigonometric bivariate function, $S_{trigonometric}$ defined as follows:

S_{sphere}: $x^2 + y^2 + z^2 = 2^2$ (14) $S_{cylinder}$: $x^2 + y^2 = 2^2$, $-2 \leq z \leq 2$ (15) $S_{trigonometric}$: $0.1(cos(x\pi) + cos(x\pi))$ (16)

The full experimentation data set covering all geometric configurations according to the surface classification given in [7], is accessible at http://dept-info.labri.fr/~charton/curvature_analysis/. The surface triangulation in use is the square split sampling defined in [13].

The proposed comparative analysis includes methods acknowledged as representative in both discrete and continuous approaches: the discrete approach formulates a closed form for differential geometry operators that works directly on the discrete representation as the method proposed by Desbrun, Meyer et al [6, 17] and abbreviated as SDA. The continuous approach includes a two stage procedure. First an entity fitting is processed: fitting of surfaces, as the simple quadratic fitting method of McIvor et al [16] abbreviated SQFA, or fitting of curves, as [4], or fitting of the curvature tensor, as [24] and [14]. Then fitted entities are "interrogated" in order to evaluate the principal curvatures and directions, the Gauss and the Mean curvatures. We study the performances of the *SineFitting* method with respect to three criteria: the pointwise convergence, the precision and the robustness.

Pointwise convergence

Let P be a target vertex on the surface S, $B(P, r)$ a ball centred at P with radius r and $c_{B(P, r)}$ an intersection curve, $c_{B(P, r)} = S \cap B(P, r)$. Let us consider $N(P)$ with central vertex at P and neighbour vertices $\{P_i\}_{i=0}^{5}$ on $c_{B(P, r)}$. The pointwise convergence tests for the target vertex P consist in checking if the estimated values of the curvatures converge to the exact values when $r \rightarrow 0$.

Precision

The estimated curvatures are compared with the exact ones computed from the curvature formulas for implicit surfaces given in see in [11].

For S_{sphere} curvatures are constant, $k_{min} = k_{max} = k_H = 0.5$, and $k_G = 0.25$. For $S_{cylinder}$ any point not belonging to the bottom and the up circle sides has $k_{min} = 0$, $k_{max} = 0.5$, $k_H = 0.25$ and $k_G = 0$. For $S_{trigonometric}$ the symbolic computations are performed using ***Maple17***.

Robustness

In order to test robustness, we investigate four strategies for the sampling of $\{P_i\}_{i=0}^{5}$ on $c_{B(P, r)}$ inspired from the experiments of [8], [9], [10] and [14]:

The regular neighbourhood $N_{reg}(P)$ corresponds to a regular sampling around P and enables to study convergence when no degenerate triangles occur in the vicinity of the target vertex.

In the irregular neighbourhood $N_{irreg}(P)$, pairs of vertices P_i and P_{i+2}, $i=0,1,2,$ are aligned. This geometric configuration corresponds to the "regular vertex" following [14]. Being less constrained than $N_{reg}(P)$, it focuses on the direction distribution and downplays the distances to the neighbours.

The regular neighbourhood with angle perturbation $N_{reg\delta Angle}(P)$ is constructed from $N_{reg}(P)$ by displacement of a single vertex P_j toward or away from its neighbours on the surface.

The regular neighbourhood with distance perturbation $N_{reg\delta Dist}(P)$ is constructed from $N_{reg}(P)$ by displacement of a single vertex P_j towards or away from P on the surface.

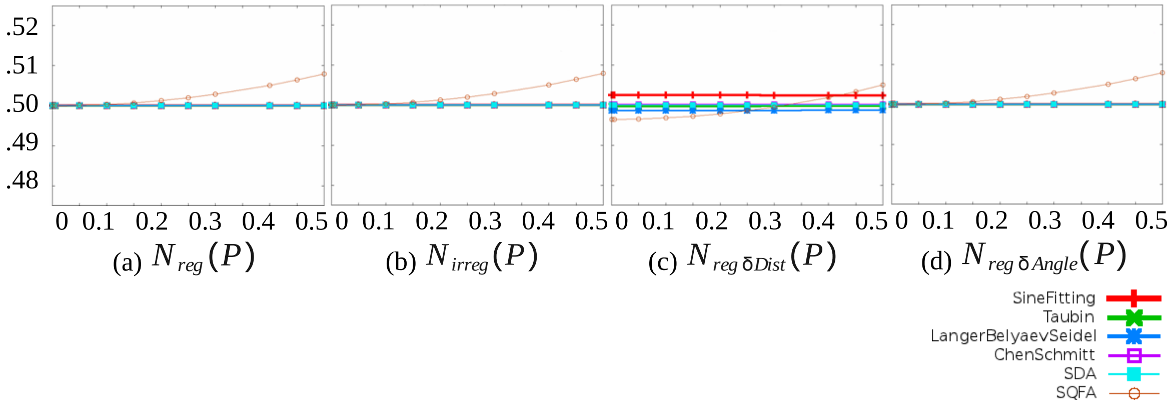


Figure 3. Sphere pointwise convergence test for mean curvature computation with area-weighted normal.

4. RESULTS

Pointwise convergence

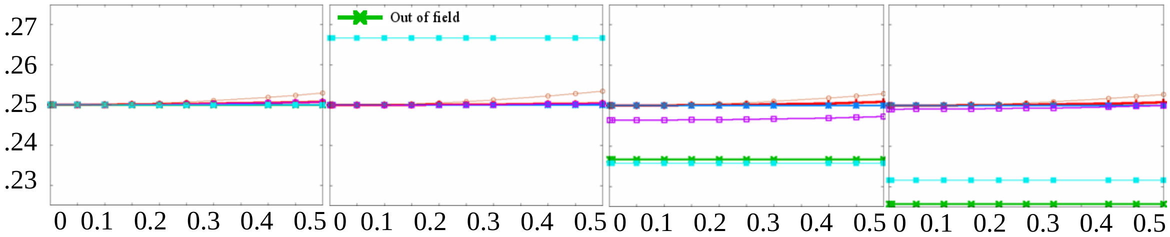
Examples of the pointwise convergence, in $P(2, 0, 0)$ on S_{sphere} are provided in Fig.3. The vertical axis indicates the variation of the mean curvature k_H. The center of the k_H variation interval is the exact value of k_H. The horizontal axis corresponds to the radius r of $B(P, r)$. According to our experimentation all methods converge to the exact curvature values when the theoretical normal is used no matter the perturbations on the neighbourhood $N(P)$. The rates of convergence are linear except for the SQFA method.

When the normal is approximated with an area weighted normal and the neighbourhood vertices are displaced for $N_{reg\delta Dist}(P)$, see Fig.3(c), only SDA converges to the exact value.

The pointwise convergence in *P(2, 0, 0)* on $S_{cylinder}$ is shown in Fig.4 and Fig.5. For this example, all methods converge to the exact values with the theoretical normal except for the case of $N_{reg}(P)$, see Fig.4(a). The methods SQFA, Langer's and the *SineFitting* converge to the exact values when the theoretical normal is used for all types of $N(P)$. The best convergence rate is achieved by the *SineFitting* method.

The results on $S_{cylinder}$ when the area weighted normal is used do not converge to the exact values when the neighbourhood is perturbed. Moreover, the type of perturbation, depending on the distance between the target vertex and the neighbours, or on the angles adjacent to the target vertex, does not have the same impact on the algorithms. As for example, Chen's and Langer's methods are precise for k_H with angle perturbation, $N_{irreg}(P)$, while approximating k_H with $N_{reg\delta Angle}(P)$ and $N_{reg\delta Dist}(P)$. The extremely erroneous value for k_H, $k_H = 0.277778$, is produced by the Taubin's method on $N_{irreg}(P)$.

From a quantitative point of view, by counting the total number of successful tests, SQFA and the *SineFitting* perform better than the others. The convergence rate makes the difference: the quadratic convergence rate of the SQFA makes it less attractive, favouring the *SineFitting* method.



(a) $N_{reg}(P)$ (b) $N_{irreg}(P)$ (c) $N_{reg\delta Dist}(P)$ (d) $N_{reg\delta Angle}(P)$

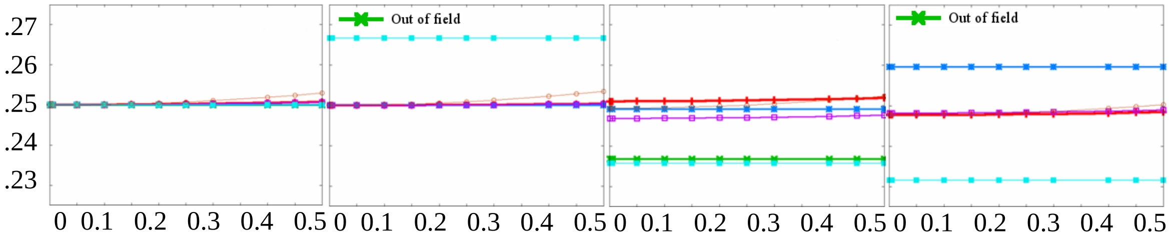
Figure 4. Cylinder pointwise convergence test for mean curvature computation with theoretical normal.

Precision

The precision is illustrated in Fig.7 with respect to the Gauss curvature evaluation on $S_{trigonometric}$. As one can see in Fig.7(b, c, f), Taubin's, Chen's and Langer's methods lack of precision in some points of inflection. The *SineFitting* method surpasses in precision the other methods: the error deviation is minimal for the different neighbourhood perturbations.

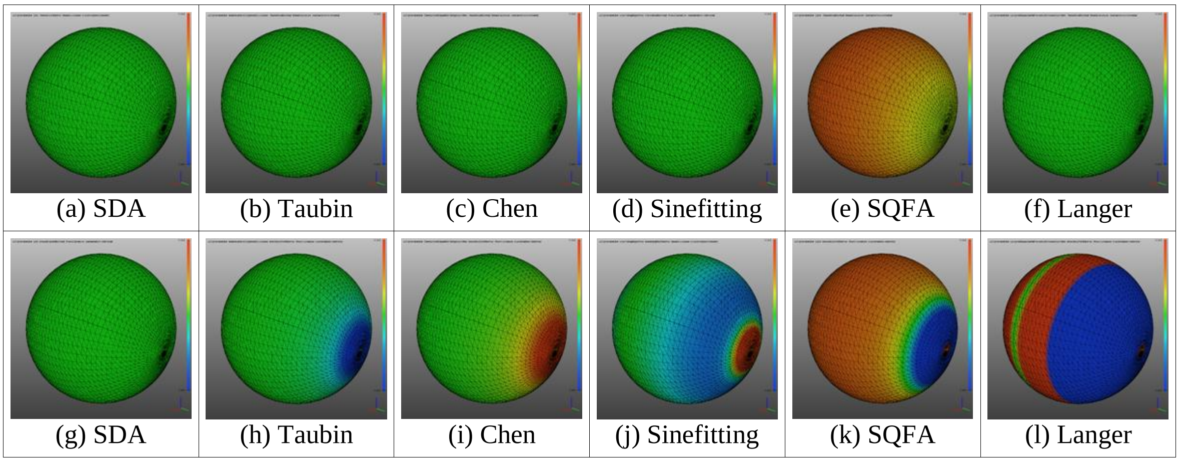
Robustness

The sensitivity of curvature estimation to the precision of the normal is illustrated in Fig.6. The regions where the estimated values are equal to the exact ones are coloured in green. With blue and red colours the regions where curvatures are respectively underestimated or overestimated are shown. All examples in Fig.6(a)-(f) use the theoretical normal while those in Fig.6(g)-(l) exploit the area-weighted normal approximation. As one can see the errors in the curvature estimation are structurally related to the quality of the underlying region triangulation, vertex valence for the poles of the sphere, and triangle quality all over the triangulation. The SDA outperforms all algorithms having similar results for both normals, the theoretical and the area-weighted ones. SQFA and Langer's methods are the more sensitive to the normal precision as seen in Fig.6(k)(l). The proposed *SineFitting* algorithm handles the irregularities in the valence and the shape of the surface triangulation when the theoretical normal is available and remains stable when the normal is approximated.



(a) $N_{reg}(P)$ (b) $N_{irreg}(P)$ (c) $N_{reg\delta Dist}(P)$ (d) $N_{reg\delta Angle}(P)$

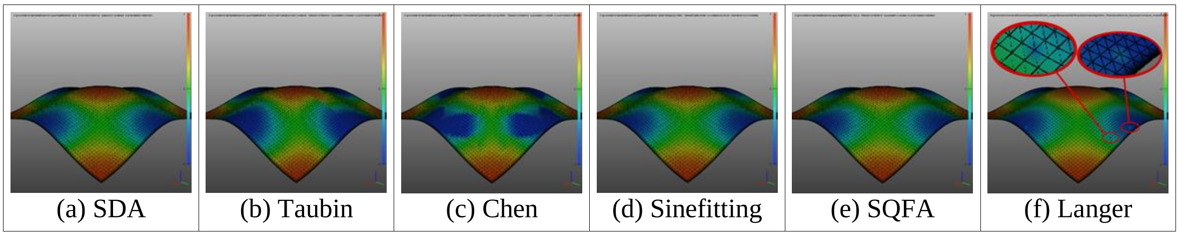
Figure 5. Cylinder pointwise convergence test for mean curvature computation with area-weighted normal.



(a) SDA (b) Taubin (c) Chen (d) Sinefitting (e) SQFA (f) Langer

(g) SDA (h) Taubin (i) Chen (j) Sinefitting (k) SQFA (l) Langer

Figure 6. Mean curvature evaluation based on theoretical and area-weighted normal



(a) SDA (b) Taubin (c) Chen (d) Sinefitting (e) SQFA (f) Langer

Figure 7. Gauss curvature evaluation based on theoretical normal

5. CONCLUSION AND PERSPECTIVES

In the present article we propose a method for the evaluation of the principal directions and curvatures, the Gauss and the Mean curvatures of surface triangulations. The elaborated method is based on a curvature *Sinefitting* algorithm and allies the advantages of the curve and surface fitting methods in processing the irregular data sampling. Three fundamental treatments are identified during the curvature evaluation at a target vertex: extraction of the neighbourhood, estimation of the normal and principal directions and curvature computation. Key attention is kept to the role of the normal estimation. A comparative analysis with other widely used methods is provided. Experimental results on chosen data set enhance the *SineFitting* performances with respect to the pointwise convergence, the precision and the robustness of the computation. One future extension of our work is to improve the normal estimation on meshes. In addition, we intend to extend our method to compute curvature on a wider scale by examining multi-ring neighbourhood. Our goal is to revisit geometric saliency and develop segmentation techniques based on precise and robust curvature estimation.

REFERENCES

[1] P.J. Besl and R. Jain. Invariant surface characteristics for 3d object recognition in range images. Computer Vision, Graphics, and Image Processing, 33(1):33-80, 1986.

[2] F. Cazals and M. Pouget. Estimating differential quantities using polynomial fitting of osculating jets. Computer Aided Geometric Design, 22:121-146, 2005.

[3] Frédéric Chazal et al. Stability of curvature measures. Comput. Graph. Forum, 28(5):1485-1496, 2009.

[4] X. Chen and F. Schmitt. Intrinsic surface properties from surface triangulation. In European Conference on Computer Vision, pages 739-743, 1992.

[5] David Cohen-Steiner and Jean-Marie Morvan. Restricted delaunay triangulations and normal cycle. In Symposium on Computational Geometry, pages 312-321, 2003.

[6] M. Desbrun et al. Discrete differential-geometry operators for triangulated 2-manifolds. In VisMath, pages 35-57. Springer-Verlag, 2002.

[7] P.J. Flynn and A.K. Jain. On reliable curvature estimation. In Computer Vision and pattern recognition, pages 110-116, 1989.

[8] R. V. Garimella and B. K. Swartz. Curvature estimation for unstructured triangulations of surfaces. Technical Report LA-UR-03-8240, Los Alamos National Laboratory, Nov 2003.

[9] T. Gatzke et al. Curvature maps for local shape comparison. In SMI, pages 246-255, 2005.

[10] T. Gatzke and C. M. Grimm. Estimating curvature on triangular meshes. International Journal of Shape Modeling, 12(1):1-28, 2006.

[11] R. Goldman. Curvature formulas for implicit curves and surfaces. Computer Aided Geometric Design, 22(7):632-658, 2005.

[12] X. Guoliang. Convergence analysis of a discretization scheme for gaussian curvature over triangular surfaces. Computer Aided Geometric Design, 23(2):193-207, 2006.

[13] B. Hamann. Curvature approximation for triangulated surfaces. In Geometric Modelling, pages 139-153, 1992.

[14] T. Langer et al. Exact and interpolatory quadratures for curvature tensor estimation. Computer Aided Geometric Design, 24(8-9):443-463, 2007.

[15] J.-L. Maltret and M. Daniel. Discrete curvatures and applications : a survey. Rapport de recherche LSIS.RR.2002.002, Laboratoire des Sciences de l'Information et des Systèmes, 2002.

[16] A.M. McIvor and R.J. Valkenburg. A comparison of local surface geometry estimation methods. Mach. Vis. Appl., 10(1):17-26, 1997.

[17] Mark Meyer. Discrete differential operators for computer graphics. PhD thesis, Pasadena, CA, USA, 2004. AAI3290476.

[18] J.-M. Morvan. Generalized Curvatures. Springer Series in Geometry and Computing, 2008.

[19] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. ACM Computing Surveys, 2:1-61, 2002.

[20] S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In 2nd Int. Symp. on 3D Data Processing, Visualization and Transmission (3DPVT 2004), 6-9 September 2004, Thessaloniki, Greece, pages 1-8. IEEE Computer Society, 2004.

[21] P.T. Sander. Generic curvature features from 3-d images. IEEE Transactions on systems, man, and cybernetics, 19(6):1623-1636, 1989.

[22] M. Spivak. A Comprehensive Introduction to Differential Geometry. Publish or Perish, inc., 1970.

[23] T. Surazhsky et al. A comparison of gaussian and mean curvatures estimation methods on triangular meshes. In ICRA, pages 1021-1026, 2003.

[24] G. Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In ICCV, pages 902-907, 1995.

[25] H. Theisel et al. Normal based estimation of the curvature tensor for triangular meshes. In Pacific Conference on Computer Graphics and Applications, pages 288-297, 2004.

HYBRID SURGERY CUTTING USING NODE SNAPPING ALGORITHM, REAL-TIME VOLUME RENDERING AND HAPTIC FEEDBACK

Jie Peng, Ling Li and Andrew Squelch
Curtin University - Kent Street, Bentley, WA 6102

ABSTRACT

We present a framework for interactive simulation of surgical cuts such as those being practiced in surgical treatment. Unlike most existing methods our framework is based on hybrid heterogeneous deformable models providing more flexibility. In order to keep the representation consistent, we apply 3-dimensional node snapping algorithm on the outer surface mesh to generate smooth cut without a large increase of element count, and employ a volumetric deformable model underneath the surface to present the internal structures and material properties of heterogeneous deformable objects. Haptic interaction is integrated into the simulation system to provide cutting tool as well as force feedback. The achieved quality and performance of the presented framework is demonstrated using human soft tissue models.

KEYWORDS

Hybrid model; Surgery simulation; Haptic interaction; Volume visualization; Mesh cutting

1. INTRODUCTION

Virtual surgical simulation is a technology dedicated to medical training and surgery planning. It is a rapidly growing field in medical imaging due on the one hand to the availability of virtual reality techniques, and on the other to the availability of detailed virtual anatomical models. Cutting simulation is the key component in surgical simulation as cutting is often a fundamental step in both conventional open surgeries and minimal invasive surgeries. However, it is a challenging task to simulate cutting operations since the object topology is changing in real-time and has to overcome conflicting requirements regarding the complexity and accuracy of the anatomical model and the speed of interaction with the model.

Most of the time in surgical simulation, it is not necessary to provide a physically-based simulation of the internal forces involved in the deformation, as a large amount of computational cost would be required. Instead, it is more important to present the three dimensional spatial relations of the complex anatomy in the cutting region as the visual clue, especially for surgical training and planning purposes (Lin *et al.* 2007). Therefore, priority should be first given to the realistic surface deformation and smooth cutting before there is a rupture. Afterward, users often focus on the exploration of the interior features.

A surface-based cutting method is capable of offering smooth and realistic cutting effect at a low computational cost, but a great deal of information about the interior structures and the material properties of the heterogeneous tissues are discarded. Meanwhile, volumetric models can incorporate appropriate internal structures and material properties for situations in which high-fidelity virtual environments are required. Direct volume rendering allows the efficient visualization of tomo-graphic 3D image data, using implicit segmentation based on transfer functions for color and opacity values, to present a great deal of information about internal structures.

We hence propose a hybrid method to deal with the cutting of heterogeneous objects, which consists of surface cutting and interior volume deformation and volume visualization. The surface mesh cutting is implemented using the node snapping algorithm and the interior volume is represented and manipulated by direct volume rendering and deformation. A haptic device is integrated into the cutting simulation system as the cutting tool so that users can observe the different materials within a deformable object as well as interact with it through touching and cutting.

The remainder of the paper is organized as follows: In Section 2, related existing work is reviewed. Our cutting simulation system is described in Section 3, which includes system overview, as well as details on surface mesh cutting, direct volume rendering, direct volume deformation and haptic interaction. Some simulation results are presented in Section 4 with the evaluation of the system performance. Section 5 concludes the paper.

2. RELATED WORK

In most of the existing simulations, cutting operation is considered only on surface models or homogeneous volumetric models. In general, surface based models are followed by some special procedures, such as the groove creation (Lin *et al.* 2007, Zhang *et al.* 2004), to generate an illusion of volumetric cuts. In (Mendoza and Laugier 2003), nonlinear elasticity of deformable objects has been modeled using nonlinear strain tensors or nonlinear spring coefficients. However, these approaches are only capable of dealing with deformable objects that consists of a single material type. More recently, adaptive hexahedral simulation meshes based on octree refinement have been widely employed (Seiler *et al.* 2011). These approaches require an explicit correspondence between the simulation elements and the embedded surface vertices. In addition, when cutting into volumetric objects, their internal volumetric structures have to be created at an extra cost. We aim at creating a hybrid model with realistic cutting effect on the outer surface and on the underneath heterogeneous volume model with detailed interior structure.

Surface mesh cutting are generally implemented by three types of topological modification approaches, e.g. element removal, mesh subdivision and mesh adaptation. Element removal technique basically removes the elements that are intersected by the cutting tool, which has been applied in (Cotin *et al.* 2000). Despite of its simplicity and computational efficiency, this method cannot present smooth cutting because of visual artifacts and the cut surfaces look unnaturally jagged. Mesh subdivision can generate smooth cutting path by subdividing tetrahedrons along the cutting planes. However, the method is computationally expensive due to the many subdivision cases and increasing element number. Another drawback is that smaller or degenerated elements may be created near the cutting site, causing instability of the simulation system. Mesh adaptation (or node snapping) is able to generate realistic cutting path without creating new elements. Smooth cut is generated by duplicating and displacing mass points that have been snapped along the cutting path.

As discussed before, surface models are not well-suited for modeling physics-based object deformation or for modeling arbitrary cutting or sculpting of objects. In fact, whenever the internal structure is important for the appearance or behavior of a graphical object, a volumetric object representation is necessary (Gibson 1999). There are three basic classes of volumetric object representations in computer graphics: 3D sampled data, such as that acquired from tomo-graphic imaging systems or computer simulations; particle system (Nakao *et al.* 2003); and geometric meshes for modeling deformable objects using mathematical techniques such as FEM (MÜller et al. 2002) or mass-spring systems (Bridson *et al.* 2002, Teschner *et al.* 2004). Another promising volumetric object representation is the linked volume, in which each element in a sampled volume is explicitly linked to its six nearest neighbors. These links are stretched, contracted, and sheared during object deformation and deleted or created when objects are cut or joined. Based on this volumetric representation, a novel approach to soft tissue deformation called 3D Chainmail is presented by S.F.F Gibson (Gibson 1997). The method was originally created for the deformation of volumetric objects as needed in surgical simulation for example. It is geometrically-based but is capable of simulating material properties to some extent.

Given the discussion above, surface model lacks the interior structure information while volumetric model could not offer smooth realistic real-time cutting, hence we propose our hybrid cutting simulation system which applies the mesh adaption as our surface cutting approach and 3D Chainmail as our volume deformation method. Volumetric model is rendered by direct volume rendering and integrated with haptic interaction.

3. METHODOLOGIES

3.1 System Overview

Our hybrid cutting system takes a two-step approach. The first step generates a smooth surface cutting, while the second step performs direct volume deformation and rendering and mixes them with surface cutting. The complete system pipeline consists of three major operations: surface cutting, volume deformation, and volume rendering and mixing. The description of each of these operations is as follows:

- Surface cutting: when the breaking strength of the material is reached, further tool motion causes the object surface to be cut and followed by a rupture. This is a progressive process and the underlying geometric model is updated using a node-snapping technique.
- Volume deformation: A cutting gutter on the underneath volumetric model is revealed along the cut opening to enhance realism. The gutter is implemented by a direct volume deformation algorithm called the Divod Chainmail (Dräger 2005) based on the Chainmail algorithm and the Enhanced Chainmail algorithm.
- Volume rendering and mixing: The last operation performs volume rendering and mixes it with surface illustration.

3.2 Surface Mesh Cutting

In this section, we will present an efficient cutting algorithm based on node-snapping that is capable of visually simulate progressive cutting with minimum increase in the number of new elements.

Node-snapping methods have been used in geographic information systems for nodal coordinate adjustments of digitized data and CAD tools for identification of features and the cleanup of data with node merging and weeding. Nienhuys and van der Stappen (Nienhuys *et al.* 2001) proposed a similar algorithm for the modification of object geometry. Later Shiyi Lin (Lin *et al.* 2007) and Yijie Lim (Lin and De 2004) applied the approach to progressive cutting.

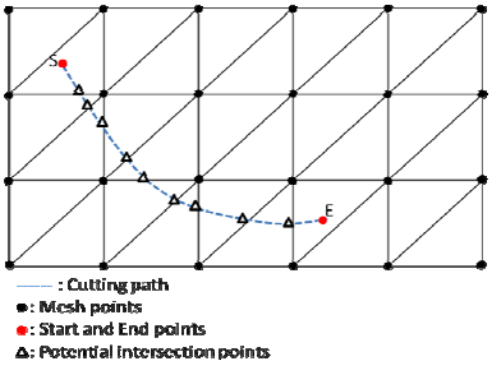


Figure 1. Cutting path intersects with mesh points

The node-snapping method starts with collision detection. After the initial collision of the haptic tool with the object surface, intersection points between the cutting path and the underlying polygon edges are calculated as shown in Figure 1. It is worth mentioning that the marked intersection points are only potential as the cutting is a progressive process. Whenever an intersection point is calculated, the local area of the mesh is updated before moving to the next intersection point. Figure 2 illustrates the details of the process. As the cut progresses, the mesh point nearest to the intersection point is snapped to the cutting path and the cut will be generated by dividing the mesh model along the cutting path and temporarily ends here. This is especially important for visual realism when the cutting tool moves very slowly and the cut should be updated in real time. The re-oriented polygon edges continuously follow the cutting path.

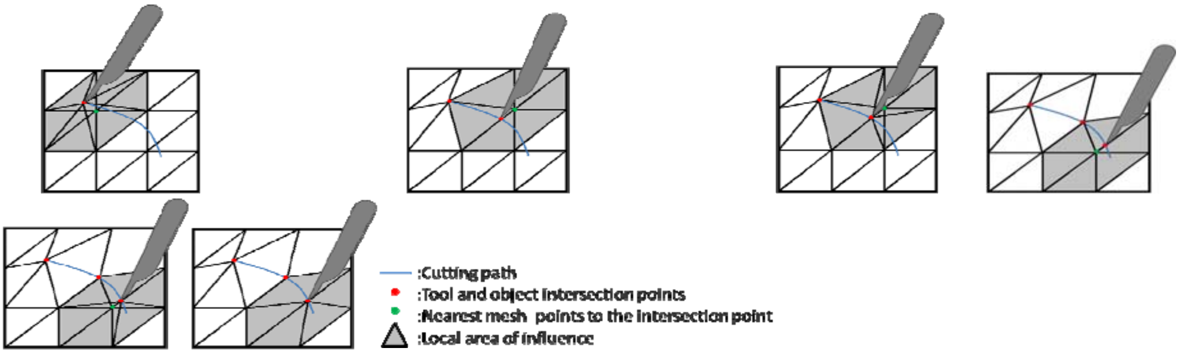

:Cutting path
:Tool and object intersection points
:Nearest mesh points to the intersection point
:Local area of influence

Figure 2. The progressive cutting

The node-snapping method starts with collision detection. After the initial collision of the haptic tool with the object surface, intersection points between the cutting path and the underlying polygon edges are calculated as shown in Figure 1. It is worth mentioning that the marked intersection points are only potential as the cutting is a progressive process. Whenever an intersection point is calculated, the local area of the mesh is updated before move to next intersection point. Figure 2 illustrates the details of the process. As the cut progresses, the mesh point nearest to the intersection point is snapped to the cutting path and the cut will be generated by dividing the mesh model along the cutting path and temporarily ends here. This is especially important for visual realism when the cutting tool moves very slowly and the cut should be updated in real time. The reoriented polygon edges continuously follow the cutting path.

For each intersection point, after the node-snapping, the next task is to open up the cut by duplicating and displacing vertexes that have been snapped along the cutting path. First, snapped points, except the starting and ending points of the cut, are duplicated twice and directly displaced at two sides of the cutting path such as vertex points S_{i1} and S_{i2} for S_i in Figure 3. The displacement direction is perpendicular to the cutting path. The original snapped points such as point S_i are then deleted. All polygon edges connected to the deleted points are reconnected to their duplicated points. As shown in Figure 3, edges originally connected to point S_i are now connected to point S_{i1} or S_{i2}, depending on which side of the cut they are located at. In particular, both points S_{i1} and S_{i2} are connected to the starting cut point S. After some points on the surface model are snapped, duplicated and deleted, surface triangles in the vicinity of the cut need to be updated. Triangles are updated according to the reconnected edges near the cut.

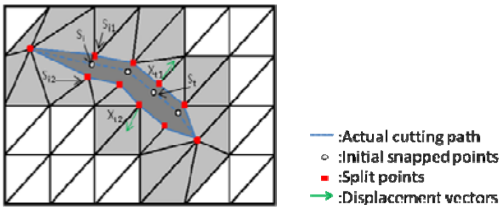


Figure 3. Cut opening generation

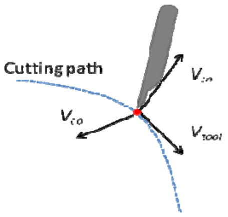


Figure 4. The definition of the cut opening vector

The displacement vector X_{t1} and X_{t2} in Figure 3 is defined on the tangent plane of the original vertex. The tangent plane normal of a vertex is calculated by the average normal of its neighboring triangles by

$$Vsn = \frac{1}{m}\sum_{j=0}^{m}\overline{Nj}$$

where m is the neighboring triangle number of point i and N_j is the normal of neighboring triangle j. Let W be the cut opening width at point i, which currently is a user-defined controlling parameter, the displacement vectors of duplicated points can be calculated by the following equations:

$$Xt1 = +\frac{W}{2}Vco\text{, } Xt2 = -\frac{W}{2}Vco$$

And the new positions of the duplicated mesh points can be calculated by:

$$St1 = St + \frac{W}{2}Vco\text{, } St2 = St - \frac{W}{2}Vco$$

where Vco is calculated according to Figure 4 as:

$$Vco = \frac{1}{|Vtool \times Vsn|} Vtool \times Vsn$$

where *Vtool* and *Vsn* are the unit vectors along the direction of the tool traveling and the tangent plane normal at node *St*, respectively.

Although this cut opening method does not follow the physical law, the generated cut is unconditionally smooth with high level of realism. This method is computational more efficient than the physically based numerical integration methods where the cut is generated by the spring forces of disconnected springs. And later we will try to build a relationship between the cut opening and the haptic feedback force, i.e., the hard the haptic press and cut, the wider the opening is.

3.3 Direct Volume Deformation

We apply the direct volume deformation algorithm (Divod Chainmail) (Dräger 2005) to simulate the gutter on the cutting site. The Divod Chainmail is based on the Chainmail algorithm and the Enhanced Chainmail algorithm.

The main advantage of the 3D Chainmail algorithm is its performance. S.F.F. Gibson has shown that each element has to be processed at most once. This allows the algorithm to work on a large data set and still produce interactive response times. A topology change can be easily done by linking or unlinking elements However; his algorithm has some major drawbacks. First of all, it is restricted to the use of rectilinear grids. It assumed at most six neighbours (left, right, top, bottom, front and back) and made assumptions about their respective position. Secondly it only works on homogeneous data. Two papers have been published which introduce methods to overcome these restrictions. The restriction to rectilinear grids is addressed by Y. Li and K. Brodlie (Li and Brodlie 2003) who introduced a Generalized Chainmail algorithm. This approach allows any number of neighbors for an element and does not make assumptions about the topology of the neighbors. The restriction of rectilinear grids is overcome by using relative rather than absolute values for the boundary constraints. M. A. Schill et al. introduce an algorithm to enable the modeling of inhomogeneous data (Schill and Gibson 1998). The basic idea is to change the chain boundaries of the elements. The movement is governed by the shape of the boundary assigned to a chain mail element. Different types of tissue are modeled with different shapes of chain regions. The problem with the introduction of inhomogeneous chain regions is that it cannot be proven that each element only has to be processed once. Hence, the speed advantage of the chain mail algorithm is lost. M. A. Schill et al. solved this problem by using sorted lists during the neighbor movement calculation. This increases the computational cost but the algorithm is still able to produce interactive frame rates.

Basically, the Divod Chainmail algorithm for local direct volume deformation consists of three parts: deformation, mapping and memory management. The first part calculates the deformation of the Chainmail object. The second part is responsible for mapping the original volume data to the Chainmail object and mapping the deformed Chainmail object back to the volume data. The third part handles the loading of the necessary portions of the Chainmail object into memory. Figure 5 illustrates the process of pulling apart the tissues along the cut through direct volume deformation.

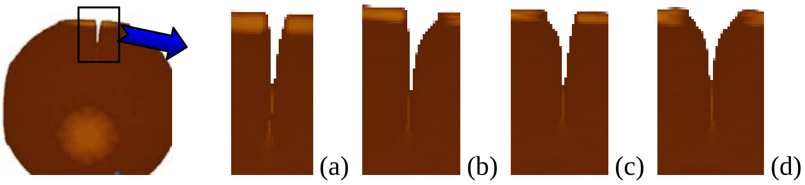


Figure 5. Direct volume cutting (a) A thin layer is cut through the sphere dataset, (b) right side is pulled away, (c) left part is pulled away and (d) left and right sides are being pulled away simultaneously and formed the cut groove.

4. IMPLEMENTATION AND RESULTS

The proposed hybrid cutting techniques are implemented on 2.4 GHz dual-CPU workstation with 2 GB memory and a high-end graphics card. The whole system is based on the open source H3D API and Volume

Haptics Toolkit (VHTK). Both are created by SenseGraphics AB. H3D is a haptic extension of the X3D scene-graph API which renders a scene graphically and haptically – a scene's objects have graphical properties (e.g., color) and haptic properties (e.g., friction). It is entirely written in C++, and it uses OpenGl and HAPI for graphics and haptics rendering. VHTK is an open-source plug-in for the H3DAPI, which was developed during a research project aiming at bringing haptics into volume data exploration interfaces and the volume data understanding process. During this project the algorithms needed for both simple and effective volume haptics were designed and the primary interface for VHTK was formed. The toolkit extends H3DAPI by introducing the scene-graph nodes necessary for loading volumetric data, handling and processing the data and for using the data to produce both visual and haptic feedback. VHTK uses the concept of 3D texture mapping for volume rendering and the traditional trial-and-error-based approach to transfer function design.

The surgical simulation system uses SensAble Technologies' PHANToM to provide force reflectance and feedback for haptic interaction. The PAHNToM provides 6 degrees of force feedback device allows the user to explore object models using the sense of touch. Eventually, the force feedback device will provide valuable sensory feedback to the surgeon during simulation of tissue deformation and cutting. All the results are performed in real-time and with haptic interaction.

By duplicating and displacing mass points that have been snapped along the cutting path, node-snapping algorithm could generate very smooth cut without adding new elements. Figure 6 shows the cut opening results on the surface on different resolution. New edges are generated and connected to their neighbor mesh points. Figure 6 (b) also shows an example of multiple cuts on the same mesh.

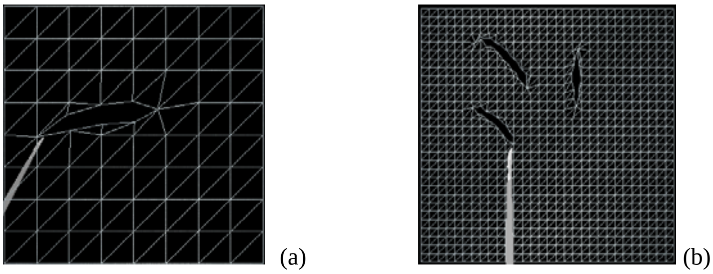

(a) (b)

Figure 6. surface cut on the mesh plane

Our cutting simulation system has been applied on datasets from CT scan of a female upper torso and a foot. The size of the torso data and the foot are 384x384x240 and 128x128x128 respectively. First we use the direct volume rendering to present the volumetric datasets as showed in Figure 7 and Figure 8.

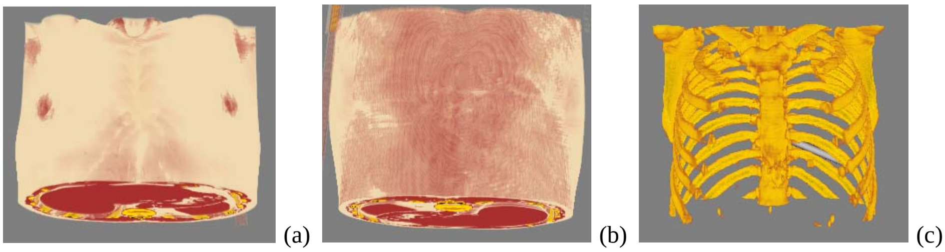

(a) (b) (c)

Figure 7. (a) Volume rendering of the female upper torso: the front, (b) volume rendering of the female upper torso: the back and (c) volume rendering of the female upper torso: the rib and spine.

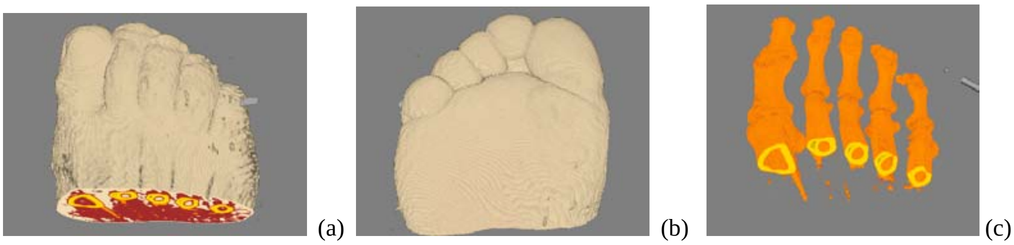

(a) (b) (c)

Figure 8. (a) Volume rendering of the foot: the front, (b) volume rendering of the foot: the back and (c) volume rendering of the foot: the bone.

In order to keep the surface and volume consistent, we use the marching-cube algorithm to extract the iso-surface from the volumetric datasets as our surface mesh. Two examples of surgical cutting simulation on the surface are shown in Figure 9. Figure 9 (a) shows the screen shot of a progressive surgical cutting on the foot surface, while Figure 9 (b) shows the surgical cutting on the female breast surface. Both of the surface models are obtained by iso-surface extraction from the volumetric data.

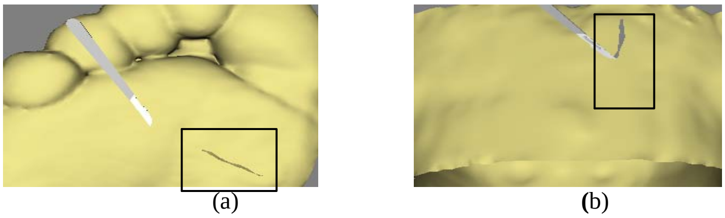


Figure 9. Surface cut on surgical mesh models.

The cut opening is widened by pulling the two sides away, to reveal the volumetric structure. Figure 10 illustrates the combination of the outer surface and underneath volumetric data. By designing different transfer functions for the volume datasets, we can choose to present only the region of interest, which is the bone in these two examples.

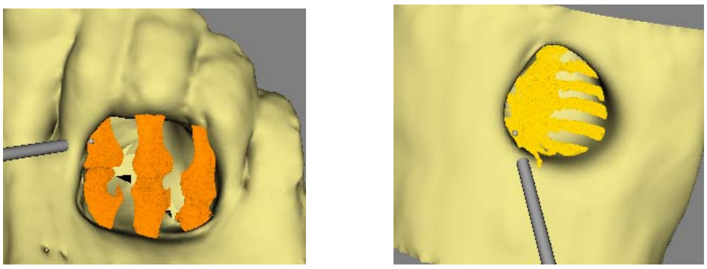


Figure 10. combination of surface and volume

Figure 11 shows the arbitrary cut with haptic device on the female chest volumetric model and Figure 12 shows the example of volume deformation by Divod algorithm with different parameters and also a snapshot of our hybrid cutting.

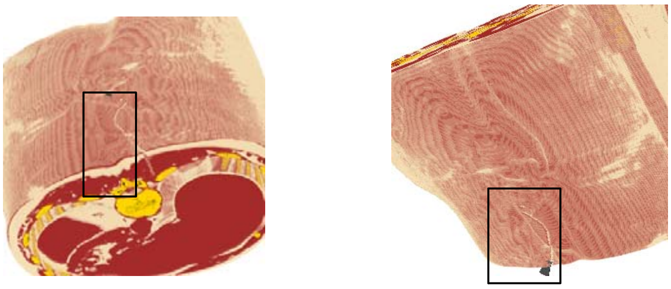


Figure 11. free cutting on volumetric model with haptic device

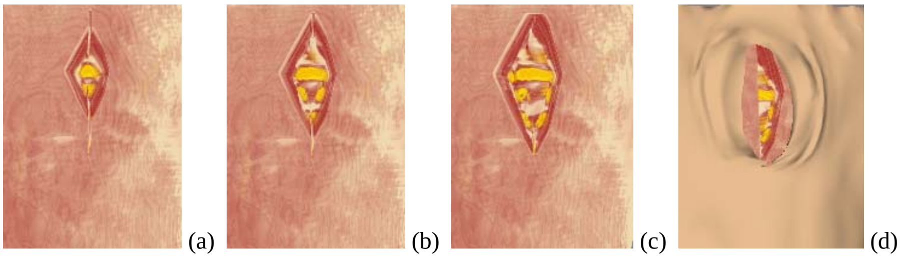


Figure 12. (a-c) volume deformation by Divod algorithm with different parameters and (d) the final result of our simulation system, including surface incision, surface deformation, volume deformation and haptic interaction.

5. CONCLUSION

In this paper, a hybrid cutting algorithm is proposed for cutting simulation on the hybrid deformable models of surface and volume. 3D node snapping and topology modification approaches are presented to generate the smooth surface on the outer surface. The cut can be manipulated and widened up to present heterogeneous interior structure and material properties of underneath volumetric deformable modes. The proposed cutting techniques can enhance the fidelity and realism of surgical simulation. They can also be used in other virtual simulation systems when topological modification is involved. Our closest goal of future work is to use two haptic devices in our system, so that the second haptic device can assists the manipulation of the object, either by "holding" the mesh or by affecting the manipulation directly.

REFERENCES

Bridson, R. et al, 2002. Robust treatment of collisions, contact and friction for cloth animation, *Proc. of SIGGRAPH' 02*, San Antonio, Texas, pp. 594-603.

Cotin, S. et al, 2000. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. The Visual Computer, 16(8):437–452.

Dräger,C., 2005. A ChainMail Algorithm for Direct Volume Deformation in Virtual Endoscopy Applications. *PhD thesis*.

Gibson, S., 1997. 3D Chainmail: A Fast Algorithm for Deforming Volumetric Objects, Proc. Symp. *Interactive 3D Graphics*, pp. 149-154

Gibson, S., 1999. Using linked volumes to model object collisions, deformation, cutting, carving, and joining. *IEEE Trans. Visualization Comput. Graph*., 5 (4) , pp. 333–348

Li Y. and Brodlie K., 2003. Soft object modelling with generalised ChainMail – extending the boundaries of web-based graphics. *Computer Graphics Forum,* 22 (4) , pp. 717–727

Lim, Y. J., and De, S., 2004. On the use of meshfree methods and a geometry based surgical cutting algorithm in multimodal medical simulations. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS'04. Proceedings. 12th International Symposium on* (pp. 295-301). IEEE.

Lin, S. et al, 2007. Snapping Algorithm and Heterogeneous Bio-Tissues Modeling for Medical Surgical Simulation and Product Prototyping, *Virtual and Physical Prototyping,* 2(2), 89-101.

Mendoza, C. and Laugier, C., 2003. Tissue cutting using finite elements and force feedback, *in Proc. IS4TM, Surgery Simulation Soft Tissue Modeling,* (Lecture Notes in Computer Science, vol. 2673), N. Ayache and H. Delingette, Eds. Berlin, Germany: Springer-Verlag, pp. 175–182.

MÜller, M. et al, 2002. Stable Real–Time Deformations, *Proc. of Symposium on Computer Animation*, San Antonio, Texas, pp. 49-54.

Nakao, M. et al, 2003. Physically-Based Fine and Interactive Soft Tissue Cutting. *IPSJ JOURNAL* 4, 44(8).

Nienhuys, H.W. and A.F. van der Stappen, 2001. A surgery simulation supporting cuts and finite element deformation. *Proceedings of the Medical Image Computing and Computer-Assisted Intervention* (MICCAI: 4), Lecture Notes in Computer Science, vol. 2208, Springer, Berlin , pp. 145–152.

Seiler, M. et al, 2011. Robust interactive cutting based on an adaptive octree simulation mesh*. The Visual Computer* (2011), 1–11. 2

Schill, M. and Gibson, S., 1998. Biomechanical Simulation of the Vitreous Humor in the Eye Using an Enhanced Chainmail Algorithm, *Proc. Medical Image Computation and Computer Integrated Surgery* (MICAI '98).

Teschner, M. et al, 2004. A versatile and robust model for geometrically complex deformable solids. *In Proceedings of Computer Graphics International CGI'04,* pp. 312–319.

Zhang, H. et al, 2004. On Cutting and Dissection of Virtual Deformable Objects, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3908-3913.

NEW RADIX-2 AND RADIX-2^2 CONSTANT GEOMETRY FAST FOURIER TRANSFORM ALGORITHMS FOR GPUS

Sreehari Ambuluri[1], Mario Garrido[1], Gabriel Caffarena[2], Jens Ogniewski[3]
and Ingemar Ragnemalm[3]

[1]Electronics Systems Division, Electrical Engineering Department, Linköping University - SE-581 83 Linköping, Sweden
[2]Bioengineering Laboratory, San Pablo CEU University - 28668 Boadilla del Monte, Madrid, Spain
[3]Information Coding Division, Electrical Engineering Department, Linköping University - SE-581 83 Linköping, Sweden

ABSTRACT

This paper presents new radix-2 and radix-2^2 constant geometry fast Fourier transform (FFT) algorithms for graphics processing units (GPUs). The algorithms combine the use of constant geometry with special scheduling of operations and distribution among the cores. Performance tests on current GPUs show a significant improvements compared to the most recent version of NVIDIA's well-known CUFFT, achieving speedups of up to 5.6x.

KEYWORDS

Fast Fourier transform (FFT), graphics processing unit (GPU), constant geometry, radix, CUDA, real-time.

1. INTRODUCTION

The Fast Fourier transform (FFT) is one of the most important algorithms for digital signal processing. Fast computation of FFTs is essential for a wide area of applications, especially those that handle large amounts of data or have to run in real time. Therefore, over the years many different projects aimed at implementing high-speed FFTs using field programmable gate arrays (FPGAs) (Garrido et. al. (2013), Garrido et. al. (2009), Duan et. al. (2011)), application-specific integrated circuits (ASICs) (Ahmed et. al. 2011) and graphics processing units (GPUs) (Volkov and Kazian (2008), Moreland and Angel (2003), Lili et. al (2010), Cui et. al. (2009), Brandon et. al. (2008), Govindaraju et. al. (2008), Duan et. al. (2011)).

FFT implementations on GPUs are especially interesting since they not only can produce a high speedup due to their massive amount of parallel cores, but also since many algorithms that depend on FFTs are executed on GPUs as well, such as (Ning et. al. 2011), (Wang et. al. 2010) or (Mazur et. al. 2011). However, when designing a GPU implementation, special care has to be taken to minimize memory transaction times and maximize the occupation of the cores.

In this paper we propose new radix-2 and radix-2^2 constant geometry FFT algorithms for GPUs. These algorithms are specially designed to optimize the use of the GPU resources. First, we use shared memory to minimize the global memory transactions, which are time consuming. Second, the algorithms make use of the concepts of constant geometry (Rabiner and Gold 1975) and processing using word groups (Baas 1999), and a special operation scheduling is used for the computations in the threads. This leads to simplification of the computations and better distribution of operations among the threads. Finally, the use of radix-2^2 provides additional improvements, as it reduces the number of multiplications in the algorithm (Garrido et. al. 2013). Although the use of radix-2^2 FFTs is successful in FPGAs (Garrido et. al. 2013), to the best of authors' knowledge this is the first time that radix-2^2 is implemented in GPUs. As a result, the improved data management and the simplifications in the operations lead to a reduction in the computation time. Experimental results show significant improvements with respect to CUFFT (Volkov and Kazian 2008).

The paper is organized as follows. Section 2 reviews the FFT and the concepts of word group and constant geometry. Sections 3 and 4 present the proposed radix-2 and radix-2^2 constant geometry FFTs, respectively. Section 5 presents the experimental results. Finally, Section 6 shows the main conclusions.

2. THE FAST FOURIER TRANSFORM

The discrete Fourier transform (DFT) of a signal in the time domain, $x[n]$, is defined as

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk}, \quad k = 0,1,\ldots, N-1$$

where $X[k]$ is the resulting signal in the frequency domain, N is the size of the transform and $W_N = e^{-j(2\pi/N)}$ is the so-called twiddle factor.

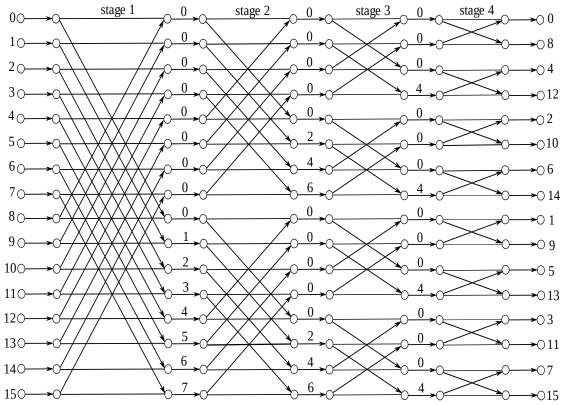


Figure 1. Flow graph of the 16-point radix-2 DIF FFT.

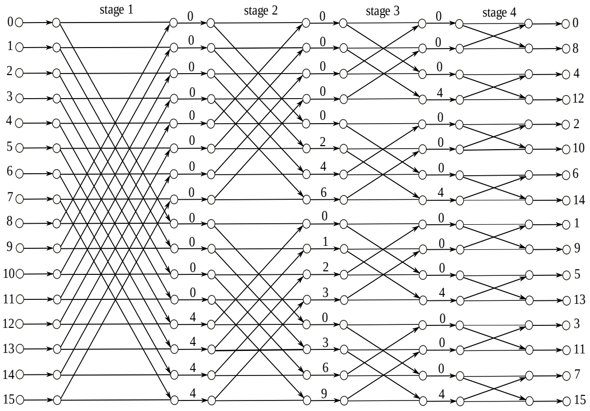


Figure 2. Flow graph of the 16-point radix-2^2 DIF FFT.

Fast Fourier Transforms (FFTs) are used to speed up the computations of DFTs. The Cooley-Tukey algorithm (Cooley and Tukey 1965) is the most common approach to calculate the FFT. It reduces the number of operations from $O(N^2)$ in the DFT to $O(N\ log(N))$. The most commonly used decomposition methods are decimation in time (DIT) and decimation in frequency (DIF) (Oppenheim and Schafer 1989).

An N-point FFT is calculated in $n=\log_r N$ stages, where r is called the radix of the FFT. The flow graph of a 16-point radix-2 DIF FFT is shown in Figure 1. Numbers at the input show the indexes of the input data, $x[n]$, whereas those at the outputs indicate the frequency, k, of the output sequence $X[k]$. Each stage of the graph consists of butterflies and rotations by the twiddle factors. Butterflies calculate additions in their upper edges and subtractions in the lower edges. Rotations are calculated according to

$$W_N^{\phi} = e^{-j\frac{2\pi}{N}\phi}$$

where ϕ are the numbers on the edges of the graph. Radices of the form 2^k are widely used in FFTs (Garrido et. al. 2013). Figure 2 shows the flow graph of a 16-point radix-2^2 DIF FFT. Radix-2^2 only differs from radix-2 FFT in the placement of the rotations (Garrido 2009). This has the advantage that rotations in odd stages are trivial, i.e., rotations by 1, j, -1 or j. These trivial rotations can be easily implemented by interchanging the real and imaginary parts of the inputs and/or changing their sign.

Further explanation of all these concepts can be found in previous literature (Oppenheim and Schafer (1989), Rabiner and Gold (1975), Garrido et. al. (2013)).

2.1 FFT Processing using Word Groups

In FFT architectures that consist of a memory and a processing element, groups of data are fetched from memory, processed and sent back to memory. This is done in an iterative fashion until all the operations of the FFT have been carried out. In this context, a word group (WG) is the number of data elements that are fetched, processed as a group and sent back to memory.

Figure 3 shows a word group of two data elements, i.e., WG = 2. The highlighted data are read from memory, processed by the butterfly, and the results are written back to memory. Then, the same is done with the next pair of data in the same stage. Once all the computations in the first stage of the FFT are carried out, the processor starts with the computations in the second stage.

Figure 4 shows the case of a word group equal to four. Now four data elements are read, processed and written back to memory as a group. In this context an epoch is defined as the number of FFT stages covered in each iteration. The use of larger word groups leads to a reduction in the total number of accesses to memory. This is beneficial in architectures with cache memory (Baas 1999), where the access time to the main memory is high. For instance, the computations highlighted in Fig 4 for WG = 4, require 4 read and 4 write operations, whereas by using WG = 2 (see Fig. 3) the memory is also accessed between stages 1 and 2, leading to 8 read and 8 write operations.

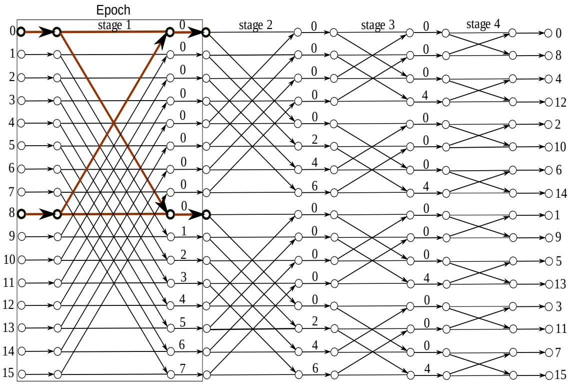


Figure 3. 16-point radix-2 DIF FFT using 2-word groups.

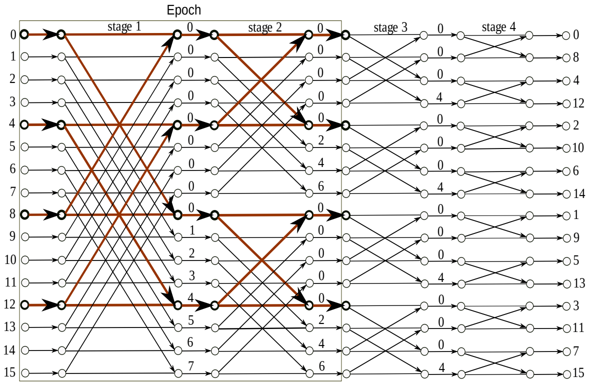


Figure 4. 16-point radix-2 DIF FFT using 4-word groups.

2.2 Constant Geometry FFT

Figure 5 shows a constant geometry (CG) radix-2 FFT (Rabiner and Gold 1975). The calculations are the same as in the conventional flow graph of the radix-2 FFT from Fig. 1. However, placement of the operations is different. The conventional FFT has the property that the outputs of any butterfly are stored in the same position as its inputs. Conversely, the constant geometry FFT has the property that all the stages follow the same pattern.

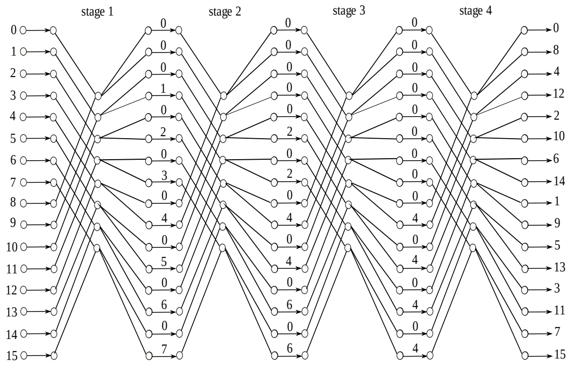


Figure 5. 16-point radix-2 DIF constant geometry FFT.

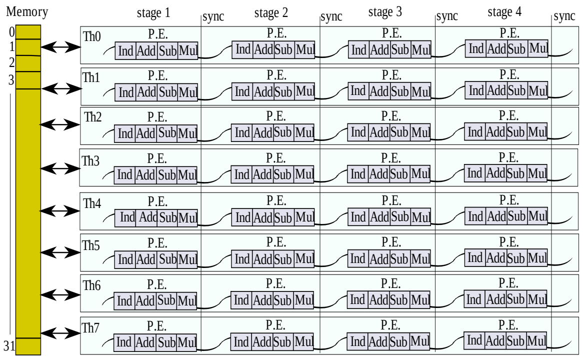


Figure 6. Parallel implementation of an FFT on GPUs.

3. PROPOSED RADIX-2 CONSTANT GEOMETRY FFT

The proposed algorithms are based on three main ideas: i) the processing using word groups; ii) the use of a constant geometry structure; and, iii) the use of an optimized schedule of the threads. This combination leads to important benefits when the FFT is calculated in GPUs. In this section the proposed radix-2 constant geometry FFT is explained. The next section will deal with the radix-2^2 implementation.

3.1 Parallelizing the FFT on GPUs

GPUs have the benefit that they can process large amounts of data in parallel. This parallelization increases the throughput and reduces the execution time. For this reason, it is important to find parallelism in the algorithms, so that the computations can be efficiently distributed among multiple parallel cores. Figure 6 shows the parallel implementation of the radix-2 FFT algorithm for the case of N = 16. As explained in Section 2, the number of stages is $\log_2 N$, each stage has N/2 butterflies, and each butterfly is followed by a multiplication by a twiddle factor. In Fig. 6 each processing element (PE) computes first the indexes of the required data and then calculates the butterflies and multiplications. After these operations, synchronization points are necessary since data must be redistributed among the threads at the end of each FFT stage.

3.2 Constant Geometry

The proposed radix-2 constant geometry FFT algorithm is based on the flow graph shown in Fig. 5. The reason why we apply the constant geometry algorithm to GPUs is that it allows for simplification of the index calculations, as will be shown next. The data elements for the FFT implementation are stored in memory. The indexing before each butterfly operation is used to determine the read and write addresses before and after the operations of each butterfly, respectively. If we consider the conventional graph in Fig. 1, the indexes of data that are processed together in a butterfly are different at each stage. The first stage processes data whose index differ in N/2 = 8. For instance, the indexes of the inputs to the first butterfly in the first stage are 0 and 8. In general, each stage $s \in \{1,...,n\}$ considers pairs of data that differ in 2^{n-s} (Garrido et.al. 2013). According to this, in a GPU the data indexes for a given thread are calculated from the thread ID (TID) as

$$\begin{aligned} I_{IN0} &= TID + floor(TID/2^{n-s}) \cdot 2^{n-s} \qquad & I_{OUT0} &= I_{IN0} \\ I_{IN1} &= I_{IN0} + 2^{n-s} & I_{OUT1} &= I_{IN1} \end{aligned}$$

where I_{IN0} and I_{IN1} are the indexes of the two input data that are processed in the same PE, and I_{OUT0} and I_{OUT1} are the output indexes. Note that the input and output indexes are the same, which allows to write the outputs in the same place where the inputs were. However, the indexes have to be recalculated at each state of the FFT, which introduces additional computations.

In the new approach using the constant geometry FFT the indexes are calculated as

$$\begin{aligned} I_{IN0} &= TID \qquad & I_{OUT0} &= 2 \cdot TID \\ I_{IN1} &= TID + N/2 = I_{IN0} + N/2 & I_{OUT1} &= 2 \cdot TID + 1 \end{aligned}$$

In this case, the indexes do not depend on the FFT stage (depicted by *s*). Therefore, they only have to be calculated once at the beginning of the computations instead of once per stage, which reduces the execution time. The fact that the input and output indexes of each stage are not the same is not an inconvenience for the GPU, because it allows for selecting the read and write addresses of the shared memory freely.

3.3 Processing using Word Groups

The second improvement of our algorithm is the use of word groups. For 2-WG radix-2, each PE in Fig. 6 calculates a butterfly and a rotation, and each stage consists of N/2 independent PE in parallel. Furthermore, each word group is processed by two threads. This provides a higher degree of parallelization than using a single thread for each PE.

This processing depends on multiple threads actually being processed in parallel on the same SM. In a GPU, threads are processed in groups called warps, which is a group of 32 threads. A warp is "woven together", and explicitly executed in parallel (Sanders and Kandrot 2011). Thus, the threads that work in the same word group are organized so that it is guaranteed that they belong to the same warp. This is basically a question of using threads with neighboring numbers. This avoids that data are unsynchronized and also avoids random errors due to race conditions.

3.4 Scheduling

Finally, an optimized scheduling has been used in order to balance the operations among the threads and, therefore, reduce the critical path in the computations. The FFT algorithm processes complex data and, thus, all the operations in the FFT are complex operations. For a 2-WG, a PE consists of a butterfly and a rotator. The butterfly requires four real additions, and the rotator needs four real multiplications and two additions. Here we assume that additions and subtractions have the same cost and count both as additions.

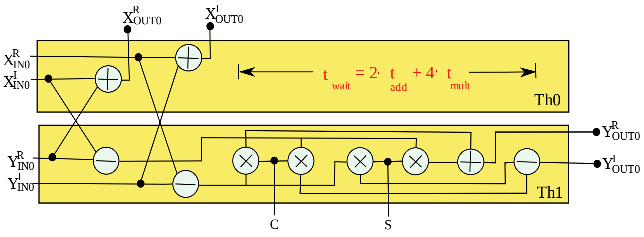


Figure 7. Unbalanced scheduling for 2-WG using 2 threads.

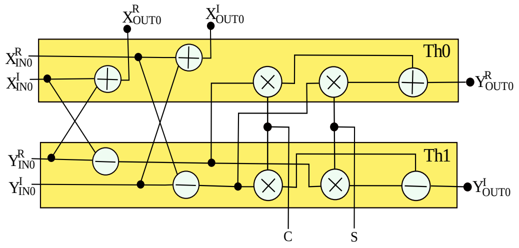


Figure 8. Balanced scheduling for 2-WG with 2 threads.

Figure 7 shows a first approach to carry out the operations using two threads for 2-WG. The operations of the upper and lower edges are carried out by the threads Th0 and Th1, respectively. The critical path (CP) is

$$CP = 4\cdot t_{add} + 4\cdot t_{mult}$$

In this scheduling, the thread Th1 has to calculate more computations than the thread Th0. Therefore, the thread Th0 has to wait for a time $t_{wait} = 2\cdot t_{add} + 4\cdot t_{mult}$ while Th1 is operating and, therefore, the workloads of the threads in Fig. 7 are unbalanced.

The critical path can be reduced by distributing the operations equally between the threads. The proposed scheduling is shown in Figure 8. This scheduling balances the operations between the threads. This reduces the critical path and, therefore, the processing time. In this case there is no waiting time and the CP is

$$CP = 3\cdot t_{add} + 2\cdot t_{mult}$$

The equalization of the computation paths between two threads in a word group also results in a better parallelization since both threads have now very similar computation paths, and more SIMD instruction can be executed. The result is a significant reduction in the computation time with respect to the unbalanced scheduling, leading to a faster processing of the FFT stages.

4. PROPOSED RADIX-2^2 CONSTANT GEOMETRY FFT ALGORITHM

This section presents the proposed radix-2^2 constant geometry algorithm for GPUs. This algorithm provides additional improvements that lead to further reductions in the execution time. First, the use of radix-2^2 reduces the number of operations of the FFT compared to radix-2. Second, the use of constant geometry guarantees a regular computation flow for all the FFT stages. Third, the proposed radix-2^2 constant geometry algorithm uses 4-WG, leading to a reduction of the synchronizations required in the FFT compared to radix-2. Finally, it uses a scheduling of the 4-WG that distributes all the operations equally among four threads.

4.1 The Radix-2^2 Constant Geometry FFT Algorithm

Figure 9 shows the proposed radix-2^2 constant geometry algorithm. The computations are the same as in the conventional radix-2^2 FFT algorithm shown in Fig. 2, yet the distribution of operations is different. With respect to the radix-2 constant geometry algorithm, the radix-2^2 constant geometry FFT algorithm only differs in the rotations that are carried out in the stages of the FFT. This can be observed by comparing Figures 5 and 9. The benefit of radix-2^2 is that the rotations in odd stages are trivial, as only multiplications by 1 and -j are needed. This simplifies the computations as those multiplications can be carried out just by changing the real and imaginary parts of the inputs and/or changing their sign. Furthermore, the pattern of the rotations in odd stages is always the same. This can be observed in Fig. 9, where the rotations at the first and at the third stages are the same.

Apart from the simplifications of the operations due to the use of radix-2^2, the use of constant geometry reduces the number of index calculations. As in the radix-2 constant geometry algorithm shown in Section 3, the indexing is the same for all the stages of the FFT. Thus, the indexes only have to be calculated once, and not at every stage.

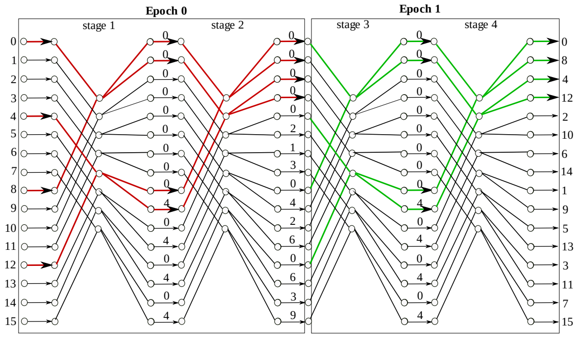


Figure 9. Proposed radix-2^2 constant geometry FFT algorithm.

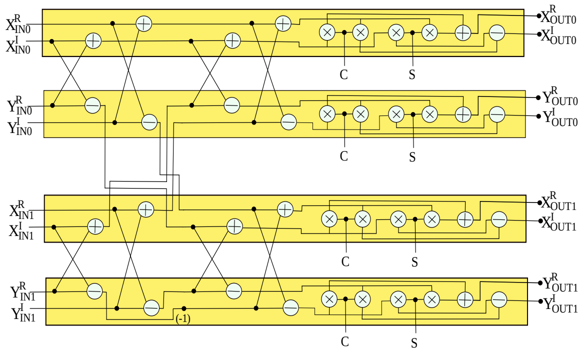


Figure 10. Scheduling for radix-2^2 4-WG using 4 threads.

4.2 Processing using Word Groups

In general, the number of stages in an epoch, S_E, is defined as

$$S_E = \log_2(WG)$$

The proposed radix-2^2 constant geometry FFT uses a word group size WG = 4 and, therefore, each epoch covers two stages of the FFT. This can be observed in Fig. 9, where the word groups are highlighted. In the GPU, the fact that S_E is 2 (due to the use of WG = 4) has the benefit that the threads only have to be synchronized every other stage. This halves the number of synchronizations compared to radix-2 and, thus, leads to a lower execution time.

4.3 Scheduling

The scheduling of the operations in the GPU threads is shown in Figure 10. The scheduling is balanced and distributes the operations of the 4-WG with radix-2^2 among 4 parallel threads. As a result, the CP is

$$CP = 6 \cdot t_{add} + 4 \cdot t_{mult}$$

In Fig. 10, it can also be observed that the trivial rotation in the 4-WG only requires a change in sign (i.e. -1). Furthermore, the synchronization after the trivial stage is not required even though several threads are used, because they are in the same warp.

4.4 Extension to 2D FFT

The proposed radix-2 and radix- 2^2 algorithms can be easily extended to 2D FFTs. Since a 2D FFT is separable into two passes of 1D FFTs (Garrido 2009), a 2D FFT breaks down to a series of 1D FFTs. For an NxN data set, N N-point FFTs are calculated for each dimension, leading to a total of 2N FFT computations.

5. EXPERIMENTAL RESULTS

This section shows the experimental results of the proposed approach. For all the experiments we have used single precision data. We have also limited the global memory access as much as possible. The twiddle factors are precomputed in advance on the CPU (and saved on hard drive between runs), copied from the host to the device and tabulated in the shared memory. The data elements are stored in the shared memory. In each stage, the data elements are read from the shared memory, processed, and written to the shared memory. This process is repeated for all the stages.

The experiments were carried out on a NVIDIA Geforce GTX 560 and CUDA toolkit v4.0.17. The device consists of 7 SMs and a total of 336 cores running at 1620-1900 MHz.

To evaluate our proposed algorithms, they have been compared with those of the NVIDIA CUFFT 4.0 library. NVIDIA CUFFT 4.0 is the most recent and fastest FFT library developed by NVIDIA. The performance of an N-point FFT is calculated in floating point operations per second (FLOPS) as

$$FLOPS = \frac{5 \cdot N \cdot \log_2 N}{t}$$

where t is the execution time. This time is measured using CUDA events. In our algorithms the execution time includes the memory transfer time of the twiddle factors and the kernel execution time. The execution time of the CUFFT 4.0 library includes the planning and the kernel execution time. We do not consider the memory transfer time of the data elements, because it is the same in all the cases.

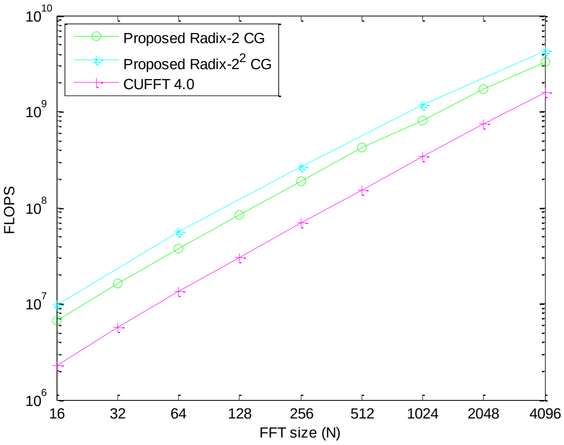


Figure 11. Comparison of the proposed 1D radix-2 and radix-2^2 CG FFTs with CUFFT 4.0.

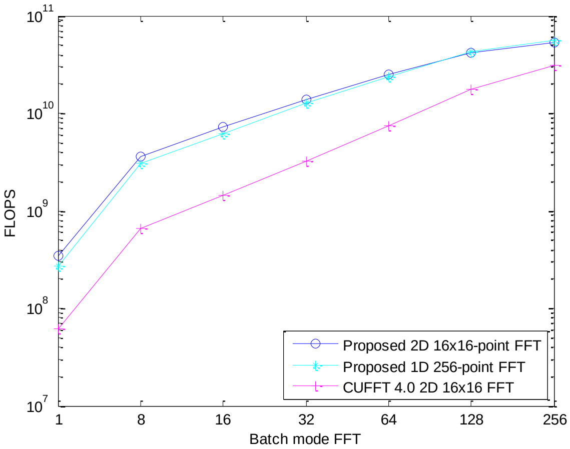


Figure 12. Batch mode comparison of the proposed FFTs with CUFFT 4.0

Figure 11 compares the execution time of a single FFT using the proposed 1D N-point radix-2 and radix-2^2 constant geometry FFTs, and CUFFT 4.0. Fig. 11 shows that the proposed radix-2 FFT algorithm improves 2 to 2.9 times the performance of CUFFT 4.0. This improvement is achieved thanks to the constant geometry and the balanced scheduling of the 2 WG using 2 threads. Furthermore, radix-2^2 achieves even higher performance than radix-2. The performance of our radix-2^2 algorithm is 2.7 to 4.3 times the performance of CUFFT 4.0. This results from the use of constant geometry, simplification of rotations, and less synchronization points due to the use of 4 WG.

Figure 12 compares multiple FFTs executed in batch mode (Sanders and Kandrot 2011). We have considered an image of 256x256 that is processed in tiles of 16x16. This consists of a stream of 256 2D 16x16-point FFTs. The 2D 16x16-point FFT is implemented by using the proposed 1D 16-point radix-2^2 CG FFT, as explained in Section 4.4. Fig. 12 shows that the performance of the proposed 2D 16x16-point FFT improves CUFFT 4.0 significantly. The speedup ranges from 1.7x to 5.6x depending on the batch mode that is chosen. Finally, in order to verify the performance of the 2D 16x16-point FFT, Fig. 12 compares it to a 1D 256-point radix-2^2 FFTs, whose complexity is comparable. As expected, both of them achieve similar results.

6. CONCLUSION

Highly efficient fast Fourier transform algorithms have been presented. The algorithms run significantly faster than CUFFT 4.0 on a modern GPU, achieving speedups up to 5.6x. This higher performance is due to the following optimizations:

1. Usage of constant geometry, which simplifies the indexing.
2. Usage of radix-2^2 constant geometry to simplify certain stages.
3. Usage of word groups with balanced scheduling, which distributes related calculations among several threads while reducing the synchronization points.
4. Usage of the shared memory for accessing both twiddle factors and the data elements.

There are several future research lines to expand and improve our work. First, we will continue to develop the new FFT algorithm in order to increase the speedup. Second, we will extend our library to larger sizes of both 1D and 2D FFTs, and also address the implementation of 3D FFTs. Third, we will also extend our library to support double precision. Finally, we plan to port the library onto other devices.

REFERENCES

Ahmed, T., Garrido, M., and Gustafsson, O., 2011. A 512-point 8-parallel pipelined feedforward FFT for WPAN. *Proceedings of Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, pp. 981–984.

Baas, B., 1999. A low-power, high-performance, 1024-point FFT processor. *In IEEE Journal of Solid-State Circuits*, Vol. 34, No. 3, pp. 380-387.

Brandon, L., Boyd, C., and Govindaraju, N., 2008. Fast computation of general Fourier transforms on GPUs. *Proceedings of IEEE International Conference on in Multimedia and Expo*, Hannover, Germany, pp. 5–8.

Cooley, J.W. and Tukey, J. W., 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation.*, Vol. 19, No. 90, pp. 297–301.

Cui, X., Chen, Y., and Mei, H., 2009. Improving performance of matrix multiplication and FFT on GPU. *Proceedings of International Conference on Parallel and Distributed Systems*, Shenzhen, China, pp. 42-48.

Duan, B., Wang, W., Li, X., Zhang, C., Zhang, P., and Sun, N., 2011. Floating-point mixed-radix FFT core generation for FPGA and comparison with GPU and CPU. *Proceedings of International Conference on Field-Programmable Technology*, New Delhi, India, pp. 1-6.

Garrido, M., 2009. *Efficient Hardware Architectures for the Computation of the FFT and Other Related Signal Processing Algorithms in Real Time*. Ph.D. dissertation, Universidad Politécnica de Madrid.

Garrido, M., Grajal, J., Sánchez, M. A., and Gustafsson, O., 2013. Pipelined Radix-2^k Feedforward FFT Architectures. *In IEEE Transactions on Very Lrge Scale Integration Systems*, Vol. 21, No. 1, pp 23-32.

Garrido, M., Parhi, K.K., and Grajal, J., 2009. A Pipelined FFT Architecture for Real-Valued Signals. *In IEEE Transactions on Circuits and Systems I*, Vol. 56, no. 12, pp. 2634–2643.

Govindaraju, N.K., Lloyd, B., Dotsenko, Y., Smith, B., and Manferdelli, J., 2008. High performance discrete Fourier transforms on graphics processors. *Proceedings of IEEE Conference on Supercomputing*, Piscataway, NJ, USA, pp. 2:1–2:12.

Lili, Z., Shengbing, Z., Meng, Z., and Yi, Z., 2010. Streaming FFT asynchronously on graphics processor units," in Information Technology and Applications (IFITA), International Forum on, vol. 1, pp. 308–312.

Mazur, R., Jungmann, J. and Mertins, A., 2011. On CUDA implementation of a multichannel room impulse response reshaping algorithm based on pnorm optimization. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 305–308.

Moreland, K. and Angel, E., 2003. The FFT on a GPU. *Proceedings of Workshop on Graphics Hardware*, pp. 112-119.

Ning, X., Yeh, C., Zhou, B., Gao, W., and Yang, J., 2011. Multiple-GPU accelerated range-doppler algorithm for synthetic aperture radar imaging. *Proceedings of IEEE Radar Conference*, Kansas City, MO, USA, pp. 698–701.

Oppenheim, A. and Schafer, R., 1989. *Discrete-Time Signal Processing*. Prentice Hall.

Rabiner, L. R. and Gold, B., 1975. *Theory and Application of Digital Signal Processing*. Prentice Hall.

Sanders, J. and Kandrot, E., 2011. *CUDA by Example*. Addison-Wesley.

Volkov, V. and Kazian, B., 2008, Fitting FFT onto the G80 architecture. [Online]. Available: http://www.cs.berkeley.edu/˜kubitron/courses/cs258- S08/projects/reports/project6_report.pdf

Wang, L., Shi, D., and Liu, D., 2010. Optimized GPU Framework for Pulsed Wave Doppler Ultrasound. *Proceedings of International Conference on Bioinformatics and Biomedical Engineering*, Chengdu, China, pp. 1–4.

TILED PROJECTION ONTO BENT SCREENS USING MULTI-PROJECTORS

Hyosun Kim[1], Christoph Schinko[1], Sven Havemann[1] and Dieter Fellner[1,2]
[1]Institute of ComputerGraphics and KnowledgeVisualization (CGV), TU Graz, Austria
[2]GRIS, TU Darmstadt & Fraunhofer IGD, Darmstadt, Germany

ABSTRACT

We provide a quick and efficient method to project a coherent image that is seamless and perspectively corrected from one particular viewpoint using an arbitrary number of projectors. The rationale is that wide-angle high-resolution cameras have become much more affordable than short-throw projectors, and only one such camera is sufficient for calibration. Our method is suitable for ad-hoc installations since no 3D reconstruction is required. We provide our method as open source solution, including a demonstrative client program for the *Processing* framework.

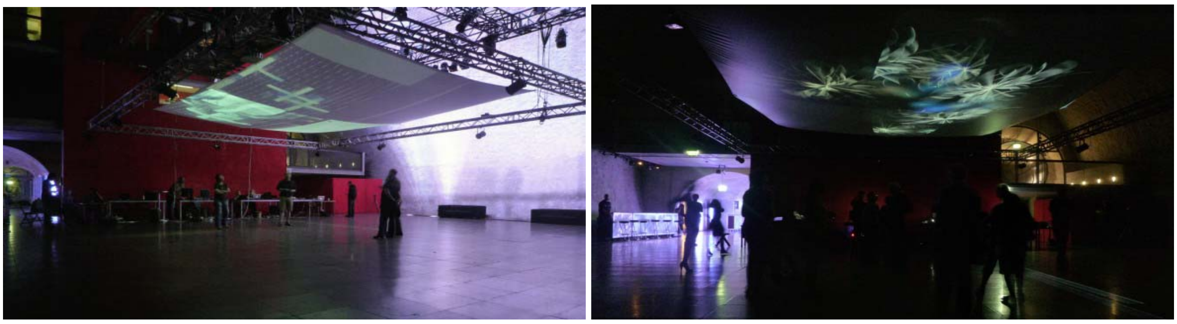


Figure 1. The Responsive Open Space installation required seamless short-throw, high-res rear projection on a bent 8 x 6 m screen. Our method solved this with 2 x 2 projectors.

KEYWORDS

Projector-camera system, multi-projector calibration

1. INTRODUCTION

A large scale display is important for high resolution imagery and visualization, such as industrial design and scientific simulation. It is becoming increasingly popular in many places such as galleries and museums, sometimes temporarily used for short term events like stage performances and exhibitions, as well. The common way to build such display systems is to tile multiple projectors and to stitch a single display surface together. However, the calibration procedure of multi-projector displays is complicated when it comes to correcting geometric misalignment within and across the different projectors. A lot of research and development has been carried out to make this process easier or more accurate. Extensive surveys are given in [Brown2005, Klose2011].

Most of the existing work is tested using well-known projectors in a permanently installed environment. An exhibition at a rented place potentially costs a lot of money, thus time is a limited resource. A lot of equipment, such as stage lighting or the sound system, is nailed to the place and hardly configurable. Implementing a projection setup at such a place imposes special demands on hardware as well as human resources. A minimal amount of hardware is desirable for quick installation and removal. Flexible cabling and light, but powerful projectors, are key components in such an effort. Apart from hardware requirements, the software needs to be robust and adjustable responding to the various configuration conditions.

The framework presented in this paper has been developed in the context of a project called Responsive Open Space initiated by ORTLOS Space Engineering Graz/London. Audiovisual artists, architects and researchers created a performative spatial environment integrating audio-visual compositions responsive to the engagement of participants among themselves. A large projection surface (8 x 6m) horizontally mounted on electronically controlled rods was installed at the Dom im Berg in Graz. The projection surface was used for interactive visualization in response to human interaction captured by a Kinect sensor. A unique sound and light environment was created. A total of four projectors created an overlapping rear-projection displaying visual compositions by creative artists. Due to size and weight of the projection surface it was heavily deformed and non-planar, which presented a main challenge.

2. RELATED WORK

Multi-projector-camera system environments can be classified into three categories according to the type of display surfaces and the number of viewer perspectives: planar (or nearly planar) tiled screens, non-planar (or arbitrary) screen with a stationary perspective, and non-planar screen for a moving viewer.

Geometric warping for planar tiled displays is generally done by finding projector-camera homographies, either using one camera [Raij2004, Oketani2005] or using multiple cameras. The overlapping projection areas are defined by placing markers on the border of the displayable area, or by using structured patterns such as checkerboards or arrays of markers [Griesser2006].

The display environment is often limited by spatial geometries, such as walls and tables in small indoor environments, which induce oblique projection angles or a discontinuity between adjacent overlapped projection areas. The most difficult problem is if the distance to the projection screen is too short, which occurs quite often. This problem can be solved by using more projectors, which also increases the resolution – the problem then is only to create a coherent image. To avoid visual distortion when images are projected on such complex geometries, a three dimensional model of the display surface needs to be obtained by conventional methods. Structured light patterns can be generated and detected by using multiple calibrated cameras, to reconstruct a 3D point cloud by epipolar geometry based on the known camera parameters [Griesser2006, Quirk2006]. With 3D fitted planes a 2-pass rendering algorithm can remap the desired image that looks perspectively correct for any viewpoint of a moving viewer [Brown2002, Raskar1999]. On the other hand, when a stationary viewpoint is given, describing piecewise planar surfaces does not really require a 3D display model. In this case, a single camera is generally placed at the location from where the viewer is supposed to observe the displayed imagery. [Tardif2003] present a function establishing the correspondence of each pixel of a projector image to a pixel of the camera image. [Yamanaka2010] generate a B-spline surface based on the small number of features estimated from the corresponding pixels between projector image and camera image. However, as the viewer moves away from the position, the imagery will begin to appear distorted.

Variation in brightness across a tiled display can split the uniformity of a single display despite perfect geometric registration. Display regions illuminated by multiple projectors look brighter, making the overlap regions very noticeable to the viewer. Color correction is necessary to get uniform illumination over the surface [Sajadi2009, Pagani2007, Kresse2003]. Ideally, luminance and chromaticity have to be measured at several input levels, for each projector and each color components. However, since any camera is sensitive to lighting and has a limited color gamut, measuring the color gamut of the display tiles in unconstrained environment is practically difficult. Intensity blending can achieve reasonable seamlessness, when projectors of the same manufacturer model are used.

3. PROJECTION SETUP AND PROBLEMS

The projection setup of the Responsive Open Space event consists of four projectors and a non-planar projection surface (see Figure 2). The projection surface is mounted on electronically controlled actuators making it possible to lift it up and down. Four overlapping projectors are mounted stationary above (behind) the projection surface in a so-called rear-projection layout. Figure 4 (left) shows a setup with parallel optical axes, whereas Figure 4 (right) shows a short-throw setup, both using two projectors.

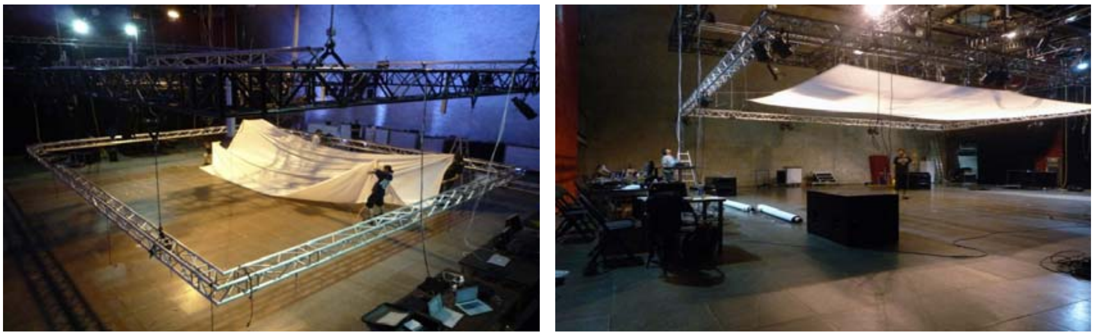


Figure 2. The projection setup of the event consists of four projectors projecting onto a movable, fabric projection surface mounted horizontally on electronically controlled actuators.

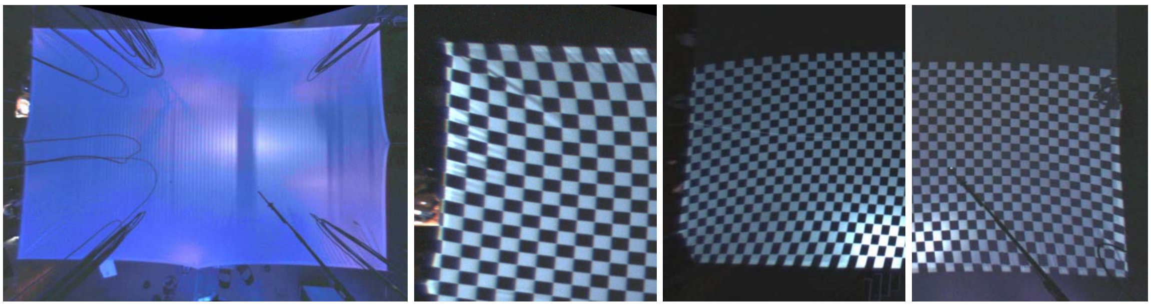


Figure 3. (a) The initial state of the display surface captured by the camera. (b) The edges of screen have irregular creases and warps as hanging from the ceiling. (c) Stage lighting can cause illumination and reflection on the surface. (d) The nailed equipment such as speakers and cables can make occlusion and shadow

To obtain the necessary information for geometric registration and brightness uniformity, a single high-resolution camera with a very wide angle lens is mounted between the four projectors that observe the whole projection area. The choice of camera and lens is attributed to the projection setup which limits the length of the USB cable of the camera. Using multiple cameras would lead to the difficulty of camera calibration on the spot. All four projectors are connected to a single computer with two dual-headed graphics cards. Using a virtual desktop, it is possible to create a single full screen that feeds all four projectors.

A fabric rear projection screen makes it possible to easily transport and install the display screen. However, its heavy but flexible properties result in easy deformation and wrinkling of the large surface, which presented additional challenges, as shown in Figure 3. Occlusion by other objects such as cables and speakers nailed to the room can cause problems when scanning the surface geometry. It has proven to be difficult to eliminate all of them near the region-of-view of the camera. Interfering light illuminated by emergency exit light or sunlight should be considered as well. Once the projectors and the display screen are lifted up to their final position, it is almost impossible to make changes to the projection setup.

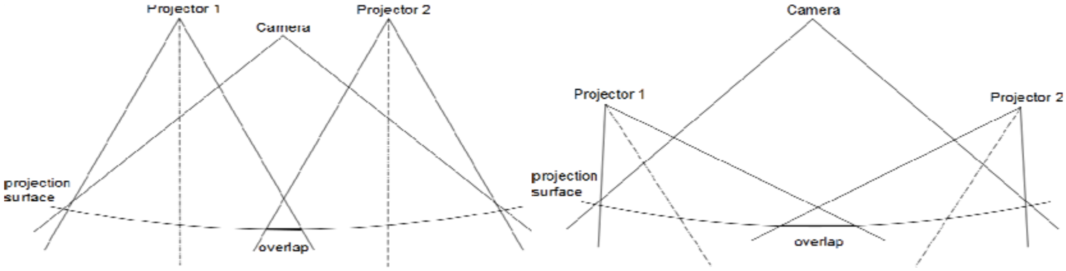


Figure 4. This figure shows two possible setups with two projectors and a camera. A parallel setup with minimal overlap is shown on the left. A very short-throw setup with larger overlap is shown on the right.

Please note that for this particular projection setup it was not feasible to mount the camera underneath the projection surface. The open space idea of the project requires multiple visitors to move freely which renders the need of a single view position redundant.

4. SOFTWARE FRAMEWORK

Tiled images on a multi-projector display must appear seamless, as if they were projected from a single projector. The main key for achieving such well-stitched imagery is to construct geometric relationship between overlapping projectors as accurately as possible. The projector calibration procedure is carried out by two software-tools dealing with the following steps:

- measuring the deformations of the projection surface and the transformation between the projectors
- compensating the deformations using the measurements of the first stage

All software created in the context of the project is open source and can be downloaded from our project website[1].

4.1 Measuring Surface Geometry for Low-Frequency Bending

The geometry of the display surface can be estimated by pixel mapping between the projected image and the camera image capturing the projected image. We can simplify the geometric registration so that the surface can be considered as an arrangement of piecewise planar surfaces. Using a checkerboard pattern can make this process easy, since feature points are exactly located at the corners of quadric patches. Depending on the surface complexity, a more or less dense checkerboard is required – which is chosen manually.

To recognize feature points accurately, we need a highly robust checkerboard recognition method. Our corner recognition method has two steps: intra-corner point detection and outermost corner point estimation (see Figure 5 left). First, intra-corner points are approximately detected by using the method proposed by Sun et al. [Sun2008]. It detects all candidate pixels that have four alternate dark and bright areas surround by extending their neighboring pixels. All the detected pixels that are adjacent to each other are merged into clusters by using connected component labeling. The center of a cluster is extracted as the initial corner position and refined through iterative process for subpixel accuracy. After all intra-corners are detected, we correlate them by the edges connecting each corner point, creating grid meshes. However these processes are for general checkerboard-like object detection. Many false meshes can be generated due to noise irrelevant to our checkerboard pattern. The pre-defined checkerboard pattern geometry is used to match the grid from the results that were found.

Detecting corner points located on the boundary of checkerboard pattern needs a different approach due to the bland brightness of their neighboring pixels. We generate the contours around the pattern boundary, and extract the convex hull of the contour as a set of candidate outermost corner points. Among the contour points that make the hull, the corner that we have to detect is selected by the closest distance from the vector consisted of the adjacent intra-corner points. In the case of the point located at the corner of the checkerboard pattern, its position can be approximated by the intersection of the two vectors passing through the neighbour corner points in case it is hardly visible (see Figure 5 right).

4.2 Projection Compensation

After obtaining the measured grid data, a *Processing* script [Reas2007] is responsible for the visualization pipeline. Processing is the platform of choice for the visual artists participating at the Responsive Open Space event. It is a programming language and development environment initially created to serve as a software sketchbook and to teach fundamentals of computer programming. It quickly developed into a tool that is used for creating visual arts. Processing is basically a Java-like interpreter, but with a new graphics and utility API together with an IDE.

[1] http:// www.cgv.tugraz.at/CGV/Research/Projects/Responsive%20Open%20Space

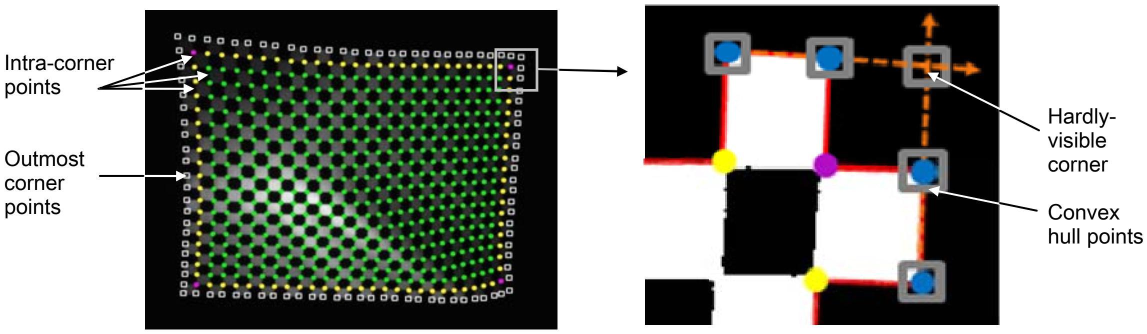


Figure 5. Once the intra-corner points are detected, graph theory is applied to generate a quad mesh. Three classes of corners with different numbers of intra-neighbors are marked by pink (2-neighbors), yellow (3-neighbors) and green (4-neighbors) circles and woven into the mesh structure (left). Two vectors passing through convex hull points (marked by blue) are intersected at a certain position where the corner of the checkerboard pattern is likely to be located (right).

The purpose of the script is to create a uniform display area relying on the measured pixel coordinates of the pre-defined grid structure. For each projector, texture coordinates of the grid are handed over to the script for displaying the client view. The client viewport is a rectangular view area that needs to be defined in pixel coordinates of the camera. A framebuffer in the size of the client viewport is used for off-screen rendering of the Processing sketch. Hooks inside the script call the sketch for rendering into the framebuffer. The display routine of the script uses the framebuffer as a texture for a pre-defined planar quad mesh using the previously calculated texture coordinates. This compensates the deformation of the projection surface from the viewpoint of the camera. The whole process is depicted schematically in Figure 6.

4.3 Intensity Blending

On the tiled projection display, the boundaries of each projection area overlap with the boundaries of the adjacent projectors. So the overlap area looks brighter than the single layer area. To render the entire surface across these regions in consistent brightness, the pixel intensities which belong to the overlapped region have to be attenuated using alpha blending.

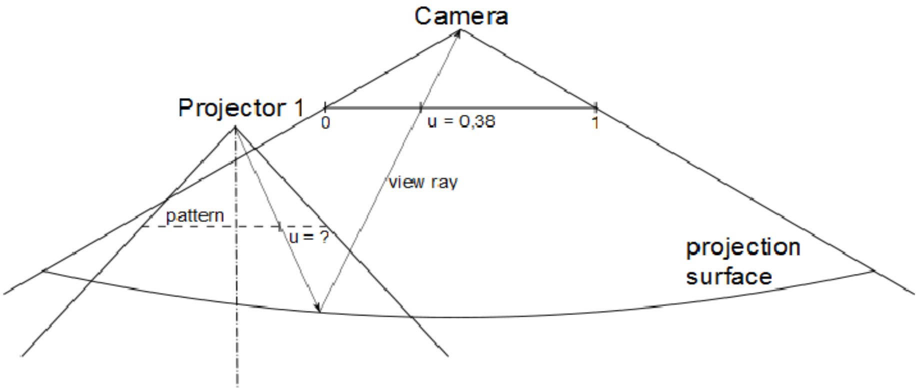


Figure 6. The projection compensation is depicted for one dimension only. A pattern is projected onto the screen and a picture is taken with the camera. For each grid point of the pattern, the u-coordinates can be measured in the camera image. These coordinates can be directly used as texture coordinates in the Processing script of the respective grid point, thus generating an undistorted projection from the viewpoint of the camera.

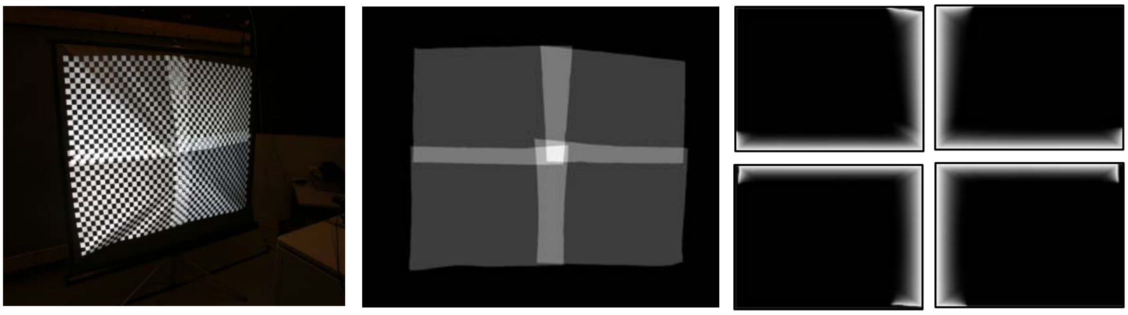


Figure 7. (a) Test bed with the deformed screen for the four projector setup. (b) The geometric relationship measured by the corner recognition. Different levels of pixel intensity represent the number of the overlapping projectors. (c) The generated alpha blending masks will be loaded as a texture map by the processing script.

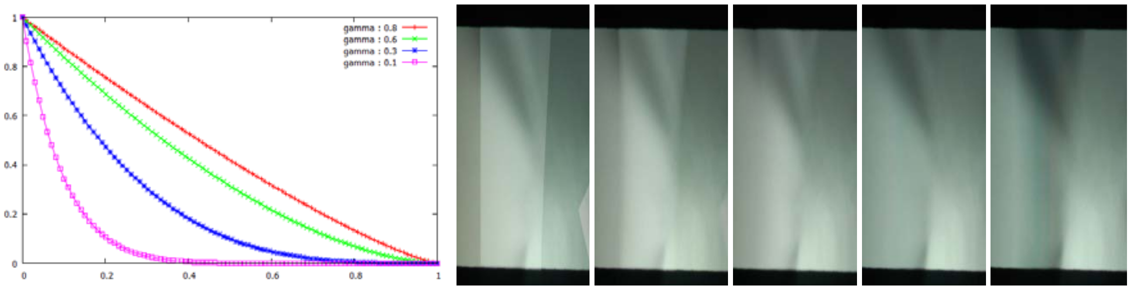


Figure 8. (a) Blending function with various *γ* values: 0.1 – 0.8. (b) Taking a form of exponentiation function with the attenuation rate as the base is useful for delicate blending adjustment. (No blending, blending with *γ:* 0.1, *γ:* 0.3, *γ:* 0.6, *γ:* 0.8 respectively)

The geometric relationship between the projectors, measured by using the camera, can exactly describe the overlapping state, as shown in Figure 7 (b). The more projector overlap, the brighter the region is. For cross-fading, we give the pixels in these areas the attenuation-rate proportional to the shortest distance from the overlapping projection boundaries. The sum of the attenuation-rate at corresponding projector pixels always adds to 1. Then the alpha weight α for intensity blending at the given pixel $p_{u,v}$ is calculated by an exponential function

$$\alpha(p_{u,v}) = (1-\beta)^{\frac{1}{\gamma}}$$

where β is the attenuation-rate and γ is the adjustment factor (see Figure 8). Through the inverse mapping to the coordinate of the projector, we can generate the alpha map for each projector (see Figure 7 (c)).

Ideally, chromaticity values of the projectors have to be measured, so that the color values in the overlapped region can be corrected for seamless colour displaying. However, for our particular projection environment interfered by lights and shadows it is not achievable to measure the color gamut of the display tiles accurately. Therefore using projectors of the same manufacturer model is required to get reasonable consistency not easily noticeable by human eyes.

5. RESULTS

We have tested our framework using various different projection scenarios. A four projector setup similar to the one used in the Responsive Open Space event was the starting point for our tests. The main difference was to not use back-projection as well as a front-facing camera. Each pattern (one per projector) was captured using a 10 megapixel camera with very wide angle lens (focal length: 2.5mm), whose intrinsic parameters of the camera were fully calibrated in advance. Recognizing the checkerboard patterns was processed in 1.5 minutes (see Table 1) leading to a total of 6 minutes for all four projectors. Generating blending masks takes about less than 3 minutes by parallel processing for all four projectors.

Table 1. Processing time (millisecond) for corner detection and blending mask generation with the different image resolution. A checkerboard pattern with 29 x 21 corner points is used

Image Resolution	Corner Detection	Blending mask
640 x 458	3042	-
864 x 600	5351	-
1280 x 916	11357	-
2592 x 1944	45256	169712
3840 x 2748	96423	172069

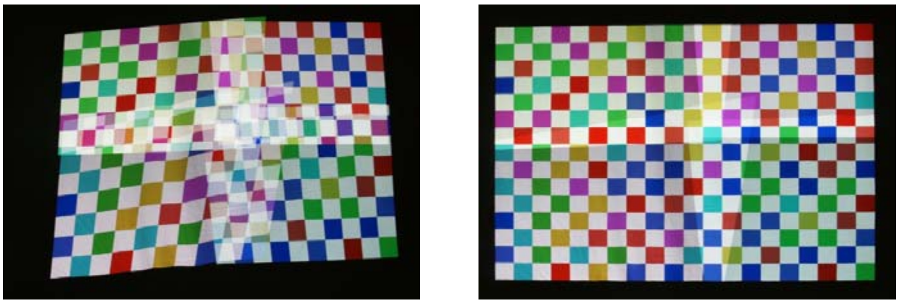


Figure 9. Displaying a checkered pattern on four overlapping projections using a non-planar projection surface leads to visual problems like nonlinear transformations and overlapping artifacts (left). The corrected result (right) shows none of these problems at all, with the exception of variations in brightness because of overlapping projections.

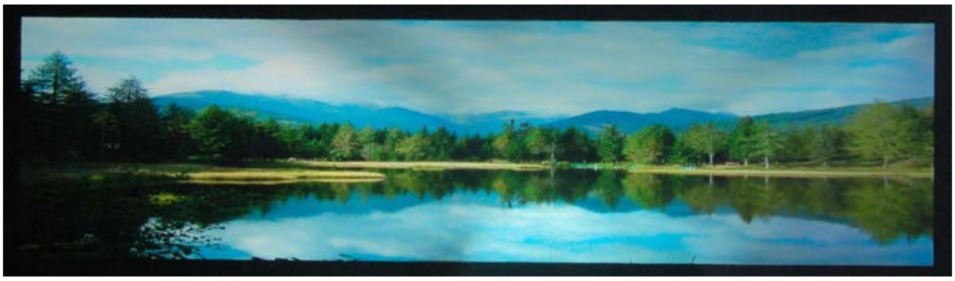


Figure 10. Imagery projected on the non-planar tiled display using three projectors. Their geometry misalignment and brightness variation across the different projectors are seamlessly corrected.

Starting from the four projector configuration, a second test setup with three projectors next to one another was accompanied by minimal configuration changes in the processing script. The results of the two test scenarios can be seen in Figure 9 and Figure 10, four projector setup and three projectors setup respectively.

6. CONCLUSION

We have presented a practical calibration method for ad-hoc installation of multi-projector displays. When a multi-projector display system is temporarily installed in a showroom with rental devices, various problems can occur, for example irregular deformation of the screen, occlusion by environmental objects and interfering lights, which are hardly configurable. While we have lack of knowledge of the used hardware, just limited time is available to revise the system according to the change of default configuration. Our approach to get the seamless projection surface in this circumstance is straightforward. The projector calibration procedure mainly consists of two steps:

- measuring the surface geometry by using a camera
- compensating the deformations using the geometric measurements

Using simple checkerboard pattern to measure the geometric relationship between the projectors is effective in many ways. It can easily filter out the minor irregularity of the screen surface such as detailed winkles, and afford manually to fill up the deficiencies of the corner points when the part of the pattern is completely occluded by other objects. The detected corner point coordinates are passed to the processing script, and then used to create a uniform display area by using the framebuffer as a texture for a predefined planar quad mesh. Our projection display system supports a quick calibration adapting to non-standard configurations dealing with arbitrary projection environments. The entire procedure will take about less than 10 minutes.

In the future, we will build an interactive (bent) projection screen using multiple cheap LED projectors, which moves automatically in response to human interaction captured by a motion sensor. Methods for recognizing the surface geometry and adapting for the quad mesh texture in realtime should be studied.

ACKNOWLEDGEMENT

The authors would like to gratefully acknowledge the generous support of the project partners especially Ivan and Andrea Redi. We would also like to thank all people involved during the event.

REFERENCES

Brown M. et al, 2005. Camera-based Calibration Techniques for Seamless Multiprojector Displays, *In IEEE Transactions on Visualization and Computer Graphics*, Vol. 11, No. 2, pp 193-206.

Brown, M.S. and Seales W.B., 2002. A practical and flexible tiled display system. *In Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, New York, NY, USA, pp. 194–203.

Chen, H. et al, 2001. Calibrating Scalable Multi-Projector Displays Using Camera Homography Trees, *In Proceedings of Conference of Computer Vision and Pattern Recognition*, pp. 9-14.

Griesser, A. and Gool, L. V., 2006. Automatic Interactive Calibration of Multi-Projector-Camera Systems, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (CVPRW'06), Los Alamitos, CA, USA, pp.8.

Klose, S. et al, 2011. Automatic Multi-projector Calibration - A Review of Systems for Non-experienced Users, *In Proceedings of the International Conference on Computer Graphics Theory and Applications*, Vilamoura, Algarve, Portugal, pp. 286-295.

Kresse, W. et al, 2003. Color consistency for digital multi-projector stereo display systems: the HEyeWall and the Digital CAVE. *In Proceedings of the workshop on Virtual Environments* (EGVE '03), Zurich, Switzerland, pp. 271-279.

Lai A. et al, 2010. Interactive Calibration of a Multi-Projector System in a Video-Wall Multi-Touch Environment. *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (UIST '10). ACM, New York, NY, USA, pp. 437-438.

Okatani, T. and Deguchi, K., 2009. Easy Calibration of a Multi-projector Display System. *International Journal of Computer Vision*, Vol. 85, No. 1, pp. 1-18.

Pagani, A. and Stricker, D., 2007. Spatially uniform colors for projectors and tiled displays, *Journal of the Society for Information Display*, Vol. 15, No. 9, pp 679–689.

Quirk, P. et al, 2006. RANSAC-Assisted Display Model Reconstruction for Projective Display, *In Proceedings of the IEEE conference on Virtual Reality*, Washington, DC, USA, pp. 318-.

Raskar, R. et al, 1999. Multi-projector Displays using Camera-based Registration. *In Proceedings of the conference on Visualization '99* (VIS '99). IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 161-168.

Reas, C. and Fry, B., 2007. *Processing: A Programming Handbook for Visual Designers and Artists*. MIT Press

Sajadi, B. et al, 2009. Color Seamlessness in Multi-Projector Displays Using Constrained Gamut Morphing, *In IEEE Transactions on Visualization and Computer Graphics*, Vol. 15, No. 6, pp 1317-1326.

Sun, W. et al, 2008. Robust Checkerboard Recognition for Efficient Nonplanar Geometry Registration in Projector-Camera Systems. *In Proceedings of the 5th ACM/IEEE International Workshop on Projector Camera Systems* (PROCAMS '08). ACM, New York, USA, Article 2 , 7 pages.

Tardif, J.-P. et al, 2003. Multi-Projectors for Arbitrary Surfaces without Explicit Calibration nor Reconstruction, *Proceedings of Fourth International Conference on 3-D Digital Imaging and Modeling* (3DIM), pp. 217 – 224.

Yamanaka, T. et al, 2010. Adaptive Image Projection onto Non-planar Screen Using Projector-Camera Systems, *In Proceedings of the International Conference on Pattern Recognition* (ICPR), pp. 307 - 310.

Yang, R. et al, 2001. PixelFlex: a Reconfigurable Multi-projector Display System. *In Proceedings of the conference on Visualization*, IEEE Computer Society, Washington, DC, USA, pp. 167-174.

VISUALIZING UNCERTAIN UNDERGROUND INFORMATION FOR URBAN MANAGEMENT

Martin Steiger, Michel Krämer, Tobias Ruppert and Jörn Kohlhammer
Fraunhofer IGD, Fraunhoferstr. 5 - 64283 Darmstadt

ABSTRACT

In this paper we present approaches for visualizing uncertainty in an application context for urban management. We describe techniques for the visualization of uncertainty and methods for the reduction of the complexity of the visualization to avoid cognitive overload. Uncertainty in both natural and man-made structures in the underground is thus communicated to the user in an appropriate, non-threatening manner. The methods were evaluated during an end-user workshop. The results of this workshop have led to various extensions to the current approach to the visualization of uncertainty in urban management.

KEYWORDS

Uncertainty visualization, urban management, underground

1. INTRODUCTION

Municipalities store and manage large amounts of data about their respective city or town. Many of them have also gathered three-dimensional models of the buildings in their town as well as vegetation, street furniture and other objects above the ground. Such information is typically very accurate as it can be measured by pure observation–aerial photography and laser scans provide a basis for precise 3D geometries. Data acquisition is usually done semi-automatically through an automatic process that is controlled and supported by human interaction. Compared to that, measuring structures that are hidden under the ground is a rather tedious task. Boreholes give information only along the drilling lines. Drilling is quite costly and–for example in densely populated regions–not always an option. The gathered information is turned into a geological shape which cannot be more than an approximation that is augmented by mathematical interpolation schemes. This naturally induces errors which lead to uncertainty in the actual geological structure. An accurate representation of reality cannot be achieved, but the amount and quality of measurements influence the overall quality of the model.

A modern city also contains not only rock but also many man-made structures under its surface. Among these are electricity lines, telephone cables and other supply lines as well as the water supply and the sewage system. Information on this kind of data is only sparsely available and highly distributed among those who are responsible–i.e. those who built it and those who modified it later on. For example, in the past, information about the depth of supply lines has rarely been documented by construction companies, because it was not required. In some cases the planned location of supply lines differs strongly from the actual location, because difficulties were encountered during the excavation–e.g. if solid rock prevents the excavators from digging. In these cases the blueprints are often not updated and thus do not reflect reality. This is also the reason why supply lines are often hit by excavators during the construction of new buildings.

In general, a combined visualization of surface and underground information can help to solve several important problems, for example in urban planning. Including information about uncertainty can help users to keep in mind that data quality is not perfect and that they have to take care when working with uncertainty. Currently, uncertainty in data is typically undefined or ignored.

There are various reasons for this. Often, uncertainty is not well documented by construction companies. In many other cases there are no means to assess the data quality appropriately (e.g. it's hard to determine the level of uncertainty in a map that is 50 years old).

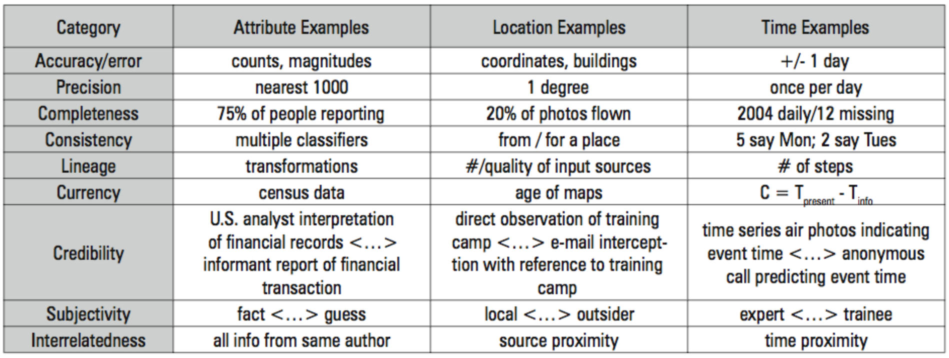
The types of uncertainty can be divided into spatial and non-spatial attributes. In order to visualize these attributes we investigate two different approaches for relevant types of uncertainty: the first approach is based on additional hull geometry for models that have uncertain location coordinates. The second approach is used for models that have non-spatial uncertainty such as currency or subjectivity.

Combining the visualization of surface and subsurface data can lead to very complex scenes where the user might have problems finding the right information. The visualization of data irrespective of the specific scenario or application might lead to cognitive information overload: the user can get lost in a 3D visualization that is cluttered with too much information. In order to avoid information overload we also present an approach to reduce the visual complexity. This can be of great help for decision makers, especially for stakeholders, who have to quickly decide based on data that contains significant amounts of uncertainty. The same technique also helps to focus on the important information and to hide objects that are not necessary for a certain problem.

In order to evaluate our approach and to benefit from already existing experience, an end-user workshop was conducted with experts from different domains. We present the results of this workshop at the end of the paper. From these we draw conclusions for further improvements to uncertainty visualization in the future.

2. RELATED WORK

Among research efforts examining the sources of data uncertainty, the first publication to mention in the context of geographical information systems (GIS) is “The Accuracy in Spatial Databases” by Goodchild and Gopal (Goodchild and Gopal, 1989). This approach considering only spatial uncertainty was extended by Unwin by including temporal uncertainty as well (Unwin, 1995). Other work describes several uncertainty typologies and representations (Crosetto, Ruiz and Crippa, 2001; Duckham et al., 2001; Gahegan and Ehlers 2000). In our approach we focus on the typology that is most commonly used today. It was proposed by MacEachren who describes sources of uncertainty in a more general way (MacEachren et al., 2005). He focuses on geographic information science (GIScience) and scientific visualization/information visualization (SciVis / InfoVis) as domains where uncertainty plays an important role. Within these two domains he identifies nine categories of uncertainty (see Figure 1). Extensions and adaptations of this can be found in several other publications (Hack et al, 2006).



Category	Attribute Examples	Location Examples	Time Examples
Accuracy/error	counts, magnitudes	coordinates, buildings	+/- 1 day
Precision	nearest 1000	1 degree	once per day
Completeness	75% of people reporting	20% of photos flown	2004 daily/12 missing
Consistency	multiple classifiers	from / for a place	5 say Mon; 2 say Tues
Lineage	transformations	#/quality of input sources	# of steps
Currency	census data	age of maps	$C = T_{present} - T_{info}$
Credibility	U.S. analyst interpretation of financial records <…> informant report of financial transaction	direct observation of training camp <…> e-mail intercept-tion with reference to training camp	time series air photos indicating event time <…> anonymous call predicting event time
Subjectivity	fact <…> guess	local <…> outsider	expert <…> trainee
Interrelatedness	all info from same author	source proximity	time proximity

Figure 1. Uncertainty categories as described by MacEachren et al., 2005

As a next step in uncertainty evaluation, Tegtmeier et al. investigate the rating within the categories in more detail, again in the context of geo-engineering (Tegtmeier, Hack, Zlatanova, 2007). The authors propose a ranking that contains different degrees of uncertainty (e.g. 1-5) and compute an overall measure of uncertainty for a given data object by weighting each uncertainty component–e.g. quality of data, quantity of data, etc.–with respect to its relevance for a specific application. Emphasizing the necessity of uncertainty consideration, a case study by Roth in the domain of floodplain mapping describes the influence of uncertainty in geographic information during decision-making (Roth, 2009). The author presents the results of a discussion with several groups of mapping experts how important it is in their daily work and which categories are the most important. They agree that uncertainty plays an important role, but does not get the required attention yet. The categories from MacEachren's table considered most important are accuracy, precision and currency. Possible methods for the visualization of uncertainty are proposed by Pang et al. who identify several methods to visualize uncertain data (Pang, Wittenbrink, Lodha, 1997). Among these are: add symbol glyphs, modify existing geometry, use animation, etc.

A comprehensive overview of the uncertainty visualization research field can be found in the work of T. Zuk (Zuk, 2008). The author states that visualizations should aim at increasing certainty by graphically exposing uncertainty. If there is additional meta-information available–e.g. the age of a dataset, its author or its lineage–information about the dataset's quality can be deduced or derived respectively. The visualization should not only contain an adequate representation of the data, but also detailed information on different quality aspects. In this way the user can decide whether and how she wants to rely on the data and make a decision based on that. Deitrick illuminates the importance of uncertainty from a very different view, namely the user-perspective (Deitrick and Edsall, 2008). Being a ubiquitous phenomenon in GIScience it requires special attention. If visualization is not adapted to user-specific needs, it just overloads the scene and thus deteriorates the overall quality of the visual representation. This, in turn, confuses the user and complicates decision making.

3. CONCEPT

In this section we first present an approach for uncertainty visualization in the geographic context. For that, a combined 3D visualization of data including surface and subsurface geometries as well as uncertainty information is realized. Uncertainty persists in many aspects of geotechnical datasets, be it in natural entities–e.g. geological bodies–or in man-made structures such as underground constructions, sewage networks or building foundations. From the visualization perspective, we divide the types of uncertainty into two major categories: spatial and non-spatial uncertainty. Based on this differentiation, uncertainty is adjusted and accordingly represented in the visualization.

3.1 Spatial Uncertainty

Uncertainty in geometric location data is induced by inaccurate and imprecise measurements. The difference between accuracy and precision has been described by Roth (Roth, 2009) and is illustrated in Figure 2. Accurate measurements are very close to the real value, but they underlie a certain variety called the “random error” (cf. Krämer, Haist, Reitz, 2007). Compared to that, in precise measurements the random error is very small, but instead a “systematical error” adds a constant bias to the measured value. The random and the systematical error induce an error range that describes an area around the measured value. The real value lies somewhere within this area.

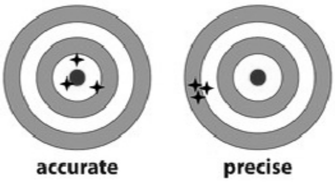


Figure 2. Accuracy versus precision in the geometric context. Adapted from Roth, 2009

Pang proposes to represent the error range of a measurement through additional geometries (Pang, Wittenbrink, Lodha, 1997). For example, an object in 2D could be surrounded by a circle with a diameter proportional to the error range. Thus, the user would know that the object’s real position is inside the circle’s boundary. We propose to implement uncertainty visualization in 3D in a similar way: a visualized geometric object with a specified spatial error–e.g. defined by an error range–is augmented with a geometric hull enclosing the 3D object. The hull emphasizes the uncertainty in the object’s location. Thereby, it does not occlude the original object, which is guaranteed by making the hull translucent. For example, we visualized the spatial uncertainty of an example water pipe with an enclosing deformed hull (Figure 3). Error ranges are typically noted in a database that contains the metadata for every object. In this case, the error ranges vary with the direction in each coordinate space. While the interpolated position in the horizontal plane has an error bound of 30cm, the vertical uncertainty has a maximal deviation of 100 cm. These ranges can be displayed via a 3D ellipsoidal hull with the respective radii.

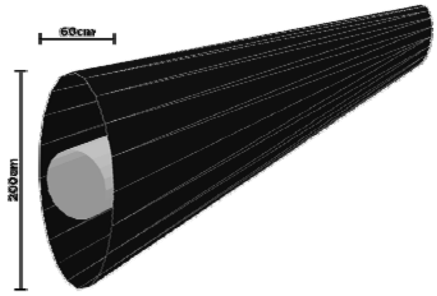


Figure 3. The inner cylinder depicts the most probable location of a water pipe. The outer hull encapsulates all possible deviations and thus serves as a measure for spatial uncertainty for the contained water pipe.

As described above, this is just one possibility to depict spatial uncertainty. To give an alternative, the visualization could modify the visual attributes of the 3D objects. For example, we could use a color gradient to represent the error range and thus reflect the inherent spatial uncertainty.

So far, we addressed spatial uncertainty regarding accuracy and precision of the measurements. In these cases error ranges that describe the deviations from the position of 3D objects are sufficient. However, integrating data from many different sources can also be a source for uncertainty. For example, the geographic location of a certain object could exist in two databases, but with different coordinates. In this case, every possible location of this object is displayed in the scene. Hence, a single object in the dataset is represented at more than one location. Such representing geometries are highlighted–e.g. via glyphs–or their visual attributes are modified depending on application-specific requirements. Showing all possible locations simultaneously could be the right choice, if the number of different locations is low and the distance between them is rather large. Otherwise several representations of the same object would overlap and the resulting object could not be identified anymore.

For those cases aggregations could provide a better solution by grouping all possible locations together. The result would be one single object representation. The lack of consistency could be displayed additionally–for example as a glyph hovering over the newly generated object.

3.2 Non-Spatial Uncertainty

Other uncertainty categories such as lineage, credibility, subjectivity, or interrelatedness cannot be described as concrete deviations of spatial correctness. Therefore, we call them non-spatial uncertainty categories. We emphasize that non-spatial uncertainty might result in spatial uncertainty. For example, the age of a measurement corresponds to the accuracy of the measurement instruments that have been used during that time period. Instead of using fixed error measurements as in the case of spatial uncertainty, we propose to partition non-spatial uncertainty variables into different categories or levels. However, it is also valid to categorize spatial uncertainty to achieve a common visualization across both types. The visualization's main purpose here is to raise the awareness of possible uncertainty in the data. Detailed information on the uncertainty attributes stored in the database is provided on demand via a glyph, a tooltip or a properties box.

In order to achieve this classification of uncertainty the heterogeneous values of each uncertainty category have to be mapped to a fixed number of uncertainty levels. This mapping is done by user-specific models or interaction. For example, buildings that exist only in cadastral maps that are older than 30 years could be mapped interactively by the user to the highest uncertainty level, and all other objects to the lowest level. A configurable mapping allows users to adjust the visualization to their needs. However, a complete model to address this task is beyond the scope of this paper.

As a major benefit of discretized uncertainty levels the number of required visualization methods can be reduced. Each method can be applied to each uncertainty category in the same manner. Treinish assumes that most visualization techniques need to be adjusted to a specific task in order to be effective (Treinish, 1999). For example, currency could be an important criterion in one case, but negligible in another. We therefore decided to enable interactive selection of these uncertainty categories that have to be depicted in the visualization.

The typical approach for visualizing the discussed uncertainty levels is realized via a diverging color map. The texture and/or the color of every object is replaced or overlaid by a single color depending on the uncertainty level of the selected uncertainty category. The user can interactively select different uncertainty categories to be visualized for each object category–e.g. geology, buildings, sewage network, etc.

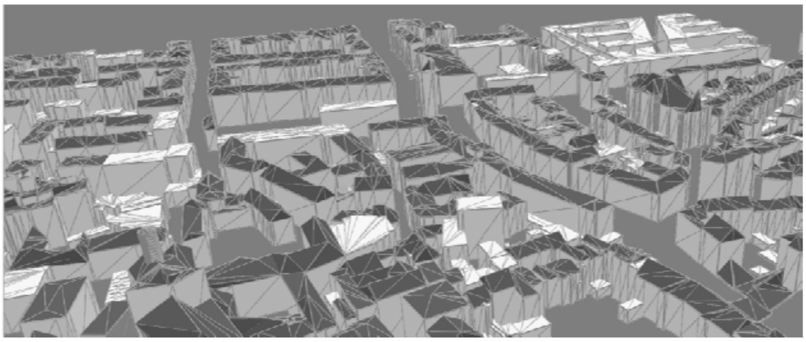


Figure 4. The wireframe outline color indicates the confidence level of individual buildings

In this case, the object representation strongly differs from the original model and information about the object's visual properties is lost. As alternatives of visualizing non-spatial uncertainty, we propose the following:

- A highlighting color is mixed in. The visual impact is weaker, but also less information is lost.
- A colored wireframe model is put on top of the original model (see Figure 4). Hardly any information is lost in this case, but the visual impact varies strongly depending on the view. The closer the camera is to the object the thinner the wireframe lines are compared to the model.
- The original model is kept unchanged, but a glyph is added close to it (usually above). The glyph gives more detailed information on the object than plain color.
- The scene is animated to focus the visual attention on objects with high uncertainty levels.

These methods inform the user that the presented data cannot be used as-is, but requires special attention.

3.3 Avoiding Information Overload

A display that is cluttered with large amounts of heterogeneous visual information quickly overstrains the user. We avoid this information overload by emphasizing relevant objects and putting irrelevant objects in the background or hiding them entirely. The visual relevance of an object depends on the specific task the user wants to address. For example, the user might want to explore the sewage network of a city. Objects above ground such as buildings or parks help to orientate in the city, but also block the user’s sight. The large amount of information that is potentially available in the scene is reduced to a few reference points. Therefore, the user is enabled to interactively hide irrelevant information in order to concentrate on things that are really of interest for his or her work. Depending on the use case, some object categories–buildings, geology, life lines, etc.–are more interesting than others. Being able to choose these categories that should be visible and others that should be hidden allows for a quick pre-selection of relevant structures.

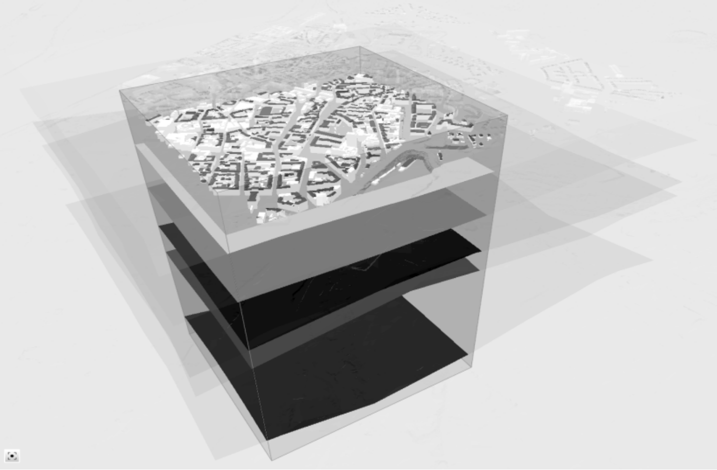


Figure 5. The visibility of the scene has been reduced to a focus cuboid on a small part of a larger city, the rest of the scene is hardly visible. The geographic extent is user-defined, whereas the depth is adjusted automatically.

In some cases, the user is just interested in a special area, consisting of a small part of a whole city. To define this area of interest, we propose a method that enables the user to mark special regions within the scene that are particularly interesting. This triggers an automatic visibility reduction of the surrounding scene in the 3D visualization by increasing its translucency up to a user-defined level. Figure 5 shows an example of such a focus cuboid.

Previously selected object categories are excluded from this procedure and keep their visibility. This method gives the possibility to specify a geographic region the user is particularly interested in. Again, irrelevant data types can be switched off. This helps to focus on a small-scale area but keep the important connections to the scene around it visible.

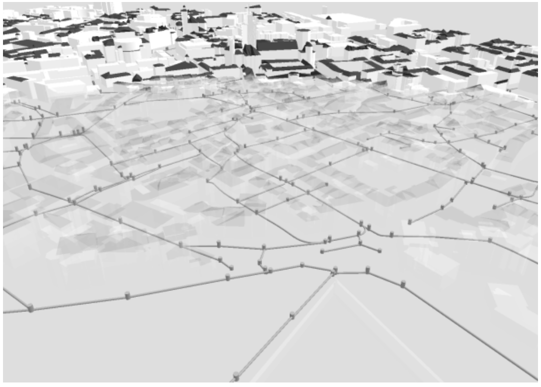


Figure 6. Bird's eye view on the city center. Irrelevant information (here: building and terrain) are stripped automatically if they are too close to the camera. The pipeline network is highlighted to increase its visibility.

In the method we just discussed, the visual properties of an area are not influenced by camera movements, which is useful if the area of interest is fixed. However, exploratory analysis of a city district would be cumbersome with this method, because interesting locations are widespread and thus the focus needs to be updated often. Therefore, we introduce a method that adapts the visual properties of objects in the scene depending on their distance to the camera. Based on the object type and its distance to the camera, the object transparency is increased. The city above ground is visible starting at a fixed distance up to infinity. Below that distance limit only a few user-defined object categories, are displayed. This “underground lens” moves along with the camera.

This provides a spatial context for orientation, but does not block sight into the layers below the terrain. This technique can be used, for example, to explore which pipelines are below the water table (see Figure 6 for a pipeline network). Again, the level of translucency can be adjusted.

4. USER WORKSHOP

In order to validate and evaluate preliminary results a workshop with end-users was conducted. Experts from different domains were invited to get diverse opinions and lively discussions. Among these were representatives from several municipalities, people from catastrophe management and geologists.

The latter group had a particular interest in uncertainty representation, because geologic formations are always approximated and their error estimations correlate strongly with the uncertainty in the data model. In contrast to many other user groups, geologists have ready-to-use data sets, and new ones created as part of their daily work. Unfortunately, uncertainty does not yet play a role in the most commonly used data structures for man-made underground objects.

This led us to the conclusion that one major topic of the workshop should be the storage of uncertainty information. As already mentioned before, a model for man-made structures needs to be designed as a first step. Then, the model has to be combined with the geological structures into a common model. The workshop showed that the expert users are aware of the uncertainty in their data even without visual representation. However, they agreed that it would be significantly easier, especially for novices or users from other domains, if discrepancies in the data would have a representation on the display. One major outcome of the workshop was the recommendation of centralized data storage and a common model for all kinds of uncertainty.

A structured database for underground structures that is similar to that of objects above ground could be used, as their nature is not fundamentally different. Tegtmeier already made a first step in that direction by comparing several existing data formats such as CityGML and GeoSciML for possible integration of underground structures (Tegtmeier, 2009).

Such an integrated system would allow software users to judge risks in urban management better than today. In order to make reasonable decisions, a visualization of uncertain information should use different degrees to highlight objects. The workshop participants agreed that three degrees are sufficient: “certain”, “uncertain” and “validated”. For some objects like geology, more degrees could be useful, but categorization then needs to be done manually. For some participants a binary indicator “certain” vs. “uncertain” was already sufficient. The geologists insisted that a categorization into different levels was crucial for certain rock structures. For them, no exact measure of uncertainty can be given, but an expert can distinguish different levels of uncertainty and thus create a categorization.

5. CONCLUSION

In this paper we discussed methods to combine surface and subsurface geotechnical data in a 3D visualization augmented with aspects from uncertainty visualization. City development is not only about surface structures but also about the underground. Current work about 3D geology modeling or uncertainty visualization only covers specific aspects. Quite a few approaches have already been undertaken in terms of integrated visualization and in uncertainty management, but often tailored to a specific problem area.

In this paper, the prevailing sources of uncertainty in the geotechnical domain have been identified and categorized. We apply uncertainty visualization to underground information in order to help city planners make decisions. Different approaches guarantee appropriate representation of additional information. Such visual representations provide the means to judge the risks as well as the opportunities of city development below the ground. Combining underground visualization with data from 3D city models supports city planners and decision makers–for example in time-critical situations. The adaptability of our visualization methods helps avoiding cognitive overload. The amount of information presented can be customized by the user to emphasize important features and reduce unneeded features to a minimum. A workshop with domain experts helped us evaluate our visualization methods. The participants agreed that an integrated visualization of uncertainty leads to a significant benefit in their daily work. They also confirmed that the methods we presented are reasonable.

ACKNOWLEDGEMENT

The research work presented here has been elaborated in the DeepCity3D project. This project is supported by the German Ministry of Education and Research (BMBF) and the French Agence Nationale de la Recherche (ANR) under the Inter Carnot-Fraunhofer program (09/2009-08/2012). Participating partners are the Fraunhofer Institute for Computer Graphics Research (IGD) in Darmstadt, Germany and the French Carnot Institute BRGM in Orléans, France.

REFERENCES

M. Crosetto, J. Moreno Ruiz, and B. Crippa. Uncertainty propagation in models driven by remotely sensed data. *Remote sensing of environment*, 76(3):373–385, 2001.

S. Deitrick and R. Edsall. Making uncertainty usable: Approaches for visualizing uncertainty information. *Geographic Visualization*, pages 277–291, 2008.

M. Duckham, K. Mason, J. Stell, and M. Worboys. A formal approach to imperfection in geographic information. *Computers, environment and urban systems*, 25(1):89–103, 2001.

M. Gahegan and M. Ehlers. A framework for the modelling of uncertainty between remote sensing and geographic information systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(3):176–188, 2000.

M. Goodchild and S. Gopal, 1989. *The accuracy of spatial databases*. Taylor & Francis, London

R. Hack, B. Orlic, S. Ozmutlu, S. Zhu, and N. Rengers. Three and more dimensional modelling in geo-engineering. *Bulletin of Engineering Geology and the Environment*, 65(2):143–153, 2006.

A. MacEachren, A. Robinson, S. Hopper, S. Gardner, R. Murray, M. Gahegan, and E. Hetzler. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science*, 32(3):139–160, 2005.

M. Krämer, J. Haist, and T. Reitz. Methods for spatial data quality of 3D city models. *Eurographics Italien Chapter Conference*, 2007.

A. Pang, C. Wittenbrink, and S. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, 1997

R. Roth. A qualitative approach to understanding the role of geographic information uncertainty during decision making. *Cartography and Geographic Information Society*, 36(4):315-330, 2009

W. Tegtmeier, R. Hack, and S. Zlatanova. The determination of interpretation uncertainties in subsurface representations. *Proceedings of the 11th Congress of the International Society for Rock Mechanics*, 105–108, Lisbon, July 2007.

L. Treinish. Task-specific visualization design. *IEEE Computer Graphics and Applications*, pages 72–77, 1999.

D. Unwin. Geographical information systems and the problem of ‘error and uncertainty’. *Progress in Human Geography*, 19(4):549, 1995

T. Zuk. 2008. *Visualizing uncertainty*. PhD thesis, University of Calgary.

UCIV 4 PLANNING: A USER-CENTERED APPROACH FOR THE DESIGN OF INTERACTIVE VISUALIZATIONS TO SUPPORT URBAN AND REGIONAL PLANNING

Diana Fernández Prieto[1], Dirk Zeckzer[2] and José Tiberio Hernández[1]
[1]*Universidad de los Andes - Bogotá, Colombia*
[2]*TechnischeUniversität Kaiserslautern - Kaiserslautern, Germany*

ABSTRACT

Decision making in the context of urban and regional planning requires communication among different stakeholders. This communication process has several barriers because of domain differences, the different nature and types of data, lack of integrated analysis tools, deficiencies in the interaction with data, and information overload. To overcome these difficulties, interactive visualizations are commonly used, and the introduction of User-Centered Design for the design of specialized interactive visualizations is an established approach to satisfy these needs. This paper presents the UCIV 4 Planning Approach to guide the design of interactive visualizations to support planning processes. This approach proposes a set of activities that help to collect relevant information about the stakeholders and their analysis tasks. Based on this information, we make suggestions of possible visualization and interaction techniques that can be applied in the design of interactive visualizations.

KEYWORDS

User-Centered Design, Analysis Task Classification, Information Visualization, Urban and Regional Planning.

1. INTRODUCTION

Interactive visualizations have been used to support urban and regional planning processes as they support effective communication and ease the comprehension and analysis of large and complex datasets (Hagen *et al.* 2009).There exist several examples of successful interactive visualizations such as Legible Cities (Chang *et al.* 2007), ESTAT (Robinson *et al.* 2005), or LIVE Singapore! (Kloeckl *et al.* 2011) that support the understanding of a region's behavior. However, barriers still exist that cause communication difficulties among stakeholders. The possible reasons, why these difficulties still persist are:

- **Domain differences:** Each stakeholder wants to know different things about a region. This could result in conflicting views on development plans (Yao *et al.* 2006).
- **Nature and types of data:** There is a large amount of new data related to each stakeholder. These data have particular attributes such as temporality, accuracy, completeness, and reliability among others that makes their integration difficult (Andrienko *et al.* 2010).
- **Lack of integrated analysis tools:** Each domain has its own analysis tools. This can produce misunderstandings when trying to communicate information about a specific field to other stakeholders (Yao *et al.* 2006).
- **Missing interaction with data:** There is a need to explore a data set through interaction techniques (Buckley and Gahegan 2000). Currently, the interaction facilities are still limited.
- **Information overload:** There is a trend to overload visualizations. Traditional visualizations of data layers allow users to display a large amount of data but at the cost of comprehensibility of information (Chang *et al.* 2007).

In order to overcome these barriers, it is necessary to provide a richer visual analysis environment. This project focuses on the design of interactive visualizations that support the stakeholders' analysis tasks. In this context, User-Centered Design (UCD) is a well-known approach to guide the development of interactive visualizations (Cartwright *et al.* 2004, MacEachren and Kraak 2001). We propose a user-centered approach based on three phases: analysis, design, and implementation. Each of these phases has a set of activities including a feedback activity whose purpose is to perform early evaluations with users. Our approach has a strong emphasis on the analysis task description, which is the basis for making recommendations about the visual representation and the interaction techniques. Our contribution consists of a structured process to guide the design of interactive visualizations to support urban and regional planning. This process takes into account the analysis tasks, a knowledge-base, and guidelines found in the literature in order to suggest and select appropriate visualization and interaction techniques. The adaptation of UCD approaches to the planning field can facilitate the understanding and communication of urban and regional phenomena among the stakeholders.

2. RELATED WORK

User-centered design (UCD) is a methodology in which users, their wants and needs, their requirements, and their tasks, are the driving force in the development of a product (Preece *et al.* 2002). The purpose of this methodology is to make products more usable, in other words, "The product should suit the user, rather than making the user suit the product" (Courage and Baxter 2005).

A large number of visualization projects that use UCD approaches are based on the definition of ISO 13407:1999 (ISO 1999). This standard called "Human-Centred Processes for Interactive Systems" provides general guidelines for introducing a UCD approach within a project. It describes an iterative process that involves at least four iterative stages before getting into the final design. Poppe and Elzakker (2006) present an adaptation of the ISO 13407:1999 cycle diagram for UCD approaches. According to ISO 13407:1999 (ISO 1999), the employment of UCD processes has several benefits for products and users: Products become easier to understand and use, products can improve user satisfaction, users can increase their productivity, and product quality is enhanced.

UCD approaches provide designers with a structured way of developing products that suit user needs. The main principles comprised in this view are: an early focus on understanding the user and the context of use, empirical testing and evaluation of the product design by representative users, and an iterative design process of four stages (Poppe and Elzakker 2006).

Applications and studies about UCD approaches in the field of geovisualization have been made in order to address the need for more useful and usable visualizations. Lloyd and Dykes (2011) mention that there is a knowledge gap between users and designers when trying to customize general visualization applications to a specific field, in this case, geovisualization. They present a long-term case study where they tested a group of UCD methods in different contexts to find out when a method is suitable for a certain purpose and when not.

Wassink *et al.* (2008) propose a three phase process for the design of interactive visualizations. These phases are: early envisioning phase, global specification phase, and detailed specification phase. Each phase can contain more than one iteration and each iteration consists of three activities: analysis, design, and evaluation where the authors suggest some methods to guide the design process.

Other projects, such as the ones presented by Carneiro (2008), Freitas *et al.* (2002), Robinson *et al.* (2005), and Roth *et al.* (2010), give insights about the advantages of the use of UCD, as, for example, the engagement of the stakeholders with the developed visualization tools, the finding of "undreamed of" requirements (Robertson 2001) associated with the displayed data, and the enhancement of understanding the problem domain.

3. UCIV 4 PLANNING APPROACH

UCIV 4 Planning is a user-centered approach for the design of interactive visualizations to support urban and regional planning processes. It is based on the detailed description of the stakeholders' analysis tasks. This includes a description of a guiding question (what do the stakeholders need to know about the region?), the data that is required to answer the guiding question, and which stakeholders of other domains have an influence on the answer.

Figure 1 illustrates the general concept of the proposed approach. It is a cyclic development model consisting of three phases: analysis, design, and implementation. Each phase uses the knowledge-base of urban and regional planning and is guided by analysis tasks. Besides, each phase includes a set of activities whose results are assessed through a feedback activity that determines if it is necessary to repeat the phase or if we can continue with the next phase. Below, we describe each of the phases and its activities.

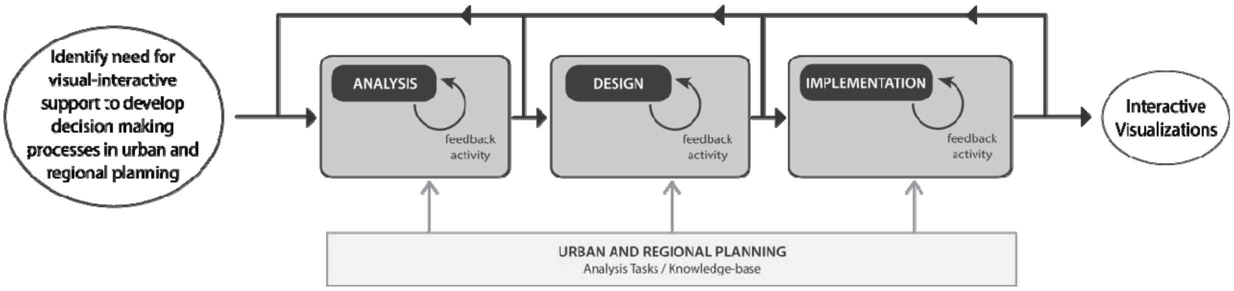

Identify need for visual-interactive support to develop decision making processes in urban and regional planning
ANALYSIS
feedback activity
DESIGN
feedback activity
IMPLEMENTATION
feedback activity
Interactive Visualizations
URBAN AND REGIONAL PLANNING
Analysis Tasks / Knowledge-base

Figure 1. UCIV 4 Planning Concept

3.1 Phase 1: Analysis

The goal of Phase 1 is to learn about the stakeholders. Two activities are suggested in order to gather and classify analysis tasks. A third activity (feedback activity) is introduced to evaluate the results of the previous activities together with the stakeholders. The output of this phase is an analysis tasks inventory.

3.1.1 Activity One

Activity One consists of three sub-activities that aim to gather essential information about the stakeholders and their analysis tasks:

1. Identification of the stakeholder profiles: to identify the stakeholder profiles, a specific questionnaire was designed to determine the domain of each stakeholder, their experience, and particular planning scenarios they have worked on.
2. Determine context of interactive visualizations: by observing a planning session, it is possible to gain knowledge about when interactive visualizations are used as well as to identify opportunities to improve planning processes through the use of visualization tools.
3. Determine analysis tasks: the stakeholders are interviewed to extract the guiding questions that will drive the development of the interactive visualizations. The core question of this interview is: what do you (as a stakeholder) need to know about the region? The answer to these questions will be used to set out specific analysis tasks. In addition to the core question, the stakeholders are asked to provide a list of the possible data needed to fulfill the analysis task.

3.1.2 Activity Two

Activity Two is focused on the classification of analysis tasks. We propose a classification method which enables to infer visualization and interaction recommendations. This classification method consists of three parts:

1. Analysis task type: Pinnel *et al.* (1999) suggest a task classification based on the cognitive processing activities required to perform each task. It provides a high level classification containing a wide range of tasks that we adapted to the urban and regional planning domain. Table 1 shows the possible categories for classifying analysis tasks.
2. Visual operations: Once the analysis task type is determined, we proceed to find the possible visual operations that can be associated with each type of analysis task. A visual operation can be defined as the visual result of applying certain transformations to a set of objects. According to Wehrend and Lewis (1990), this transformation is directly related to the user tasks. These authors present a taxonomy that includes the following visual operations: identify, locate, distinguish, categorize, cluster, distribution, rank, compare, associate, and correlate. We will use only *identify, locate, associate*, and *compare* operations as they seem to be most appropriate for analyzing geospatial data (Ogao and Kraak 2002). To link analysis task types and

visual operations we created the matrix presented in Table 1, where analysis task types are related to one or more visual operations.

Table 1.Analysis Tasks Taxonomy. Analysis Task Type and Description are adapted from Pinnel *et al.* (1999), while we combined those with the visual operations.

		Visual Operation			
Analysis Task Type	**Description**	**Identify**	**Locate**	**Associate**	**Compare**
Spatial determination	Identify the location where a specific urban phenomenon takes place.	X	X		
Comparison of values or attributes	Identify differences between attributes or values of urban elements.				X
Distinguishing between alternatives	Highlight differences between two or more urban interventions or stages in time.				X
Locating optima	Find the best location for an urban element in the urban system.		X		X
Determining trends	Discover patterns in the evolution of an urban phenomenon.			X	X
Relations between attributes	Understand and interpret relations between attributes of different urban elements.			X	X
Aggregation of information	Observe the attributes of urban elements in a higher level.			X	
Qualitative information	Establish comparative measures for qualitative attributes of urban elements.			X	X
Quantitative information	Identify patterns of change of quantitative attributes of an urban element.			X	X
Description	Observe the behavior of an urban phenomenon in a specific context.	X	X	X	X

3. Interaction operations: In addition to visual operations, we must also consider interaction operations to complement visual operations. Yi *et al.* (2007) propose seven categories to classify interaction techniques commonly used in Information Visualization. These categories are:

- Select: mark something as interesting
- Explore: show me something else
- Reconfigure: show me a different arrangement
- Encode: show me a different representation
- Abstract/Elaborate: show me more or less detail
- Filter: show me something conditionally
- Connect: show me related items

For each category, there is a set of recommended interaction techniques that can be integrated into our interactive visualization. At the end of these two activities, the descriptions of the analysis tasks, its corresponding classification, and the raw data needed should be clear. This will enable developers of the interactive visualizations to understand the interests of the different stakeholders as well as to obtain a first guide of what to do in terms of visualization and interaction operations.

3.1.3 Feedback Activity: Focus Group

The feedback activity consists of a focus group session (Preece *et al.* 2002) whose purpose is to share the results of the two previous activities with the stakeholders. The expected output of this activity is a set of analysis tasks with its corresponding classification (task type, visual operations, and interaction operations) approved by the stakeholders in order to guarantee that the intention of the original analysis tasks is preserved. In case the stakeholders do not approve the proposed analysis task description, a complete iteration of Phase 1 should be considered. However, if the stakeholders do not approve the proposed classification and related data, only a new iteration of Activity 2 and 3 should be considered.

3.2 Phase 2: Design

Phase 2 examines the criteria for selecting visual representations and interaction techniques according to the analysis task type. The first two activities focus on the search of design guidelines from involved domains and visual perception literature. The feedback activity consists of a Participatory Design session based on the evaluation of a paper prototype. The aim is to evaluate the appropriateness of the selected guidelines before proceeding with the implementation of the interactive visualizations.

3.2.1 Activity One

An important aspect to consider for the design of interactive visualizations is the stakeholders' knowledge-base. It is necessary to apply guidelines, standards, and rules for the display of spatial information in all involved domains.

For this activity, we suggest a literature review and interviews with the stakeholders so as to know, if there are specific standards that apply to a specific analysis task. The American Planning Association - APA provides general standards for urban and regional planning. Among these are those highlighted in the books “Planning and Urban Design Standards” (American Planning Association 2006) or the “Land Based Classification Standard –LBCS” document (American Planning Association 1996).

In addition to knowledge-base guidelines, several human factors aspects such as perception and cognition should also be considered (Tory and Möller 2004). In the specific context of urban and regional planning, it is important to strengthen the link between the stakeholders’ knowledge-base and the theories of perception and cognition.

From the visualization perspective, the appropriate use of visual attributes for encoding certain data types has to be considered. For example, Ware (2004) describes the operation of the visual apparatus as a function of perception. Two examples of Ware's guidelines referring to the use of color are:

- “The use of gray-scales colors is not a particularly good method for coding data”.
- “For ordinal values to be correctly and rapidly interpreted, it is important that the color sequence increases monotonically with respect to one or more of the color opponent channels”.

On the other hand, Mackinlay (1986) in his effort to automate the design of visual representations proposes a ranking of appropriateness for the use of visual attributes for encoding quantitative, ordinal, and nominal data (Mackinlay's ranking). From the interaction perspective, Yi *et al.* (2007) present an inventory of possible interaction techniques associated with each interaction operation.

3.2.2 Activity Two

Activity Two consists of designing and prototyping interactive visualizations based on the guidelines found in the previous activity. The input for this activity is the detailed description of one analysis tasks and its related knowledge-base. This description should include:

- Analysis task classification: task type, visual operations, and interaction operations
- Raw data description: data types for the required data to perform the task
- Knowledge-base related to the task: standards or guidelines for the visual representation

The prototypes developed can have different resolutions: Low-fidelity, such as sketches for presenting possible representations or paper prototypes (Snyder 2001) for modeling interaction; or hi-fidelity prototypes such as Processing (Reas and Fry 2007) prototypes with some of the visualization and interaction aspects included. For the first iteration, we recommend paper prototypes as they are fast to develop and as they allow making fast changes on the fly. Further iterations can include more advanced prototypes using larger amounts of data.

3.2.3 Feedback Activity: Participatory Design session using Paper Prototypes

In order to assess the prototypes, we propose a Participatory Design session using paper prototypes (Osman and Baharin 2009). This test aims at evaluating part of the functionality of the interactive visualizations as well as the visual representations with the stakeholders. A facilitator who knows the behavior of the proposed interactive visualization simulates the response of the system when a user (in this case a stakeholder) performs a certain action on the interface. All the responses considered for the interactive visualization should be part of the prototype so the users can see the reactions to their actions. During this session, the stakeholders are invited to comment about the visual representation of the data and the selected interaction techniques in order to improve the prototypes for a new iteration, if necessary.

3.3 Phase 3: Implementation

Phase 3 incorporates the selected visual representations and interaction techniques into interactive visualizations. The output of this activity is a set of interactive visualizations assessed with the stakeholders through a usability test. These interactive visualizations support all analysis tasks identified in Phase 1.

3.3.1 Activity One

Activity One comprises the processing of the Participatory Design session (PD session) results in terms of possible visual and interaction misunderstandings. As the stakeholders had a simulated experience during the PD session, they could point out particular aspects of the visualization and interaction that can cause troubles when performing the analysis tasks. The expected output for this activity is a summary of the misunderstandings identified by the stakeholders and the solutions they suggested.

3.3.2 Activity Two

The objective of Activity Two is to define how visual representations and interaction techniques are going to be integrated into the interactive visualization. The decision of what technology to use should be based on the results of the previous phase (analysis tasks, visual operations, and interaction operations) and a review of possible technologies that support these requirements.

3.3.3 Feedback Activity: Usability Test

We propose a usability test to evaluate the interactive visualizations that we designed regarding the following usability criteria described by Tullis and Albert (2008):

- Effectiveness: Being able to complete a task.
- Efficiency: The amount of effort required to complete the task.
- Satisfaction: The degree to which the user was happy with his or her experience while performing the task.

A series of metrics is associated with each criterion. For example, *effectiveness* can be measured using a task success metric or the error count; *efficiency* can be measured using a time-on-task metric or learnability metrics; and *satisfaction* can be measured using self-reported metrics.

After analyzing the results of the usability test, starting a new cycle should be considered:

- Starting from Phase 1 if there are misunderstandings related to the purpose of the analysis tasks, or
- Starting from Phase 2 if there are issues with the design of the visual representation or interaction, or
- Starting from Phase 3, if the implementation has to be improved.

This decision will depend largely on the evaluation results and the acceptance criteria determined by the stakeholders.

4. CONCLUSION

We have comprised and adapted relevant methods and techniques from User-Centered Design in a structured process in order to guide the design of interactive visualizations to support urban and regional planning. The use of UCIV 4 Planning Approach promotes the documentation of analysis tasks and knowledgebase guidelines that are dispersed in the literature. This not only can improve the quality of the design and production of interactive visualizations but also can help to gain a better understanding of the tasks involved in urban and regional planning.

There is potential for applying the UCIV 4 Planning approach to other fields as this approach is general enough to be extended and adjusted to other domains. In that case, the classification process presented in Phase 1 - Activity Two should be redefined based on the particular knowledge of the field.

We also found some issues that must be considered when using the approach. It is necessary to clarify the role of the stakeholders during the design process. The stakeholders are invited to participate in the feedback activities of each phase. They can comment and do suggestions about the material presented to them, but:

- How are they involved in the design process?
- Can they "change" design decisions about the interactive visualization or do they only point out their opinion?
- Do they have a role as evaluators or as co-designers?

The selection of the type of role for each stakeholder has a high impact on the acceptance of the final interactive visualizations.

By using the proposed approach it is possible to obtain visualization and interaction recommendations that support a specific analysis task. Then, depending on the analysis task type, visual and interaction operators can be used to select specific visualization and interaction techniques. The challenge is in how to produce interactive visualizations that support multiple analysis tasks avoiding the development of highly specialized tools only usable for expert trained users.

We applied the UCIV 4 Planning approach in a project. The results of this case study are presented in the paper "Using User-Centered Techniques for the Design and Evaluation of Interactive Visualizations to Support Urban and Regional Planning: Case Study Bogotá 21" (Fernández Prieto, *et al.* 2013). Future work includes the implementation of the approach in different application areas, for example, in the analysis of medical imaging or in software visualization, and lastly the formal evaluation of the interactive visualization produced by using our approach.

ACKNOWLEDGEMENT

We wish to thank Colciencias (Bogotá, Colombia) for partial funding of this project through the Young Researchers Program and the DFG (Deutsche Forschungsgemeinschaft) for funding the IRTG (International Research Training Group) 1131. We also wish to thank all members of IMAGINE Research Group specially Juan Camilo Ibarra and John Alejandro Triana, Urban and Regional Sustainability Research Group (SUR), and the Architecture School from Universidad de los Andes (Bogotá, Colombia) for their advice and participation during the course of this project.

REFERENCES

American Planning Association, 1996. *Land-Based Classification Standards* [online] Available from: http://www.planning.org/lbcs [Accessed 13 April 2013].

American Planning Association, 2006, *Planning and urban design standards*, John Wiley & Sons Inc., New Jersey, USA.

Andrienko, G., Andrienko, N., Demsar, U., Dransch, D., Dykes, J., Fabrikant, S.I., Jern, M., Kraak, M-J., Schumann, H., and Tominsky, C., 2010, 'Space, Time and Visual Analytics', *International Journal of Geographical Information Science*, vol. 24, no.10, pp. 1577–1600.

Buckley, A. and Gahegan, M., 2000, *Geographic Visualization as an Emerging Research Theme in GIScience*, Research Proposal, University Consortium for Geographic Information Science.

Carneiro, C., 2008, 'Communication and Visualization of 3-D Urban Spatial Data According to User Requirements: Case Study of Geneva', *Proceedings of the XXI ISPRS Congress*, vol. XXXVII, part B2, pp. 631–636.

Cartwright, W., Miller, S., and Pettit, C., 2004, *Geographical Visualization: Past, Present and Future Development*, Journal of Spatial Science, vol. 49, no. 1, pp. 25–36.

Chang, R., Wessel, G., Kosara, R., Suda, E., and Ribarsky, W., 2007, Legible Cities: Focus-dependent Multi-Resolution Visualization of Urban Relationships', *IEEE TVCG*, vol. 13, no. 6, pp. 1169–75.

Courage, C. and Baxter, K., 2005, *Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques*,Morgan Kaufmann Publishers, San Francisco, USA.

Fernández Prieto, D., Zeckzer, D., and Hernández, J.T., 2013, Case Study Bogota 21 – Designing Interactive Visualizations to Support Urban and Regional Planning, *EuroRV3: EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization,* Leipzig, Germany (accepted for publication).

Freitas, C. M. D. S., Luzzardi, P.R.G., Cava, R.A., Winckler,M.A.A., Pimenta, M.S., and Nedel, L.P.., 2002, Evaluating Usability of Information Visualization Techniques, *Proceedings of 5th Symposium on Human Factors in Computer Systems*, Fortaleza, Ceará, Brazil,pp. 40–51.

Hagen, H., Guhathakurta, S. and Steinebach, G., 2009, *Visualizing Sustainable Planning*, Springer and X.media.publishing, Germany.

ISO, 1999, ISO 13407:1999 Human-centered Design Processes for Interactive Systems.

Kloeckl, K., Senn, O. and Lorenzo, G. D., 2011, LIVE Singapore! An Urban Platform for Real-time Data to Program The City, *Computers in Urban Planning and Urban Management 2011*, Calgary, Canada, pp. 1-16.

Lloyd, D. and Dykes, J.,2011, Human-Centered Approaches in Geovisualization Design: Investigating Multiple Methods Through a Long-Term Case Study, *IEEE TVCG*,vol. 17, no. 12, pp.2498–2507.

MacEachren, A. M. and Kraak, M.-J., 2001, Research Challenges in Geovisualization, *Cartography and Geographic Information Science*, vol. 28, no. 1, pp. 1–11.

Mackinlay, J., 1986, Automating the Design of Graphical Presentations of Relational Information, *ACM Transactions on Graphics (TOG)*, vol.5, no. 2, pp. 110–141.

Ogao, P. and Kraak, M.-J., 2002, Defining Visualization Operations for Temporal Cartographic Animation Design, *International Journal of Applied Earth Observation and Geoinformation*, vol. 4,no. 1, pp. 23–31.

Osman, A., Baharin, H., Ismail, M.H., and Jussof, K., 2009, Paper prototyping as a rapid participatory design technique, *Computer and Information Science*, vol.2, no. 3, pp. 53–57.

Pinnel, L. D., Dockrey, M., and Borning, A., 1999, Design and Understanding of Visualizations for Urban Modeling, Technical report, University of Washington.

Poppe, E. and Elzakker, C., 2006, Towards a Method for Automated Task-Driven Generalization of Base Maps, *UDMS 2006 - 25th Urban Data Management Symposium*,Denmark, Aalborg, vol. 3, pp.3.51–3.64.

Preece, J., Rogers, Y., and Sharp, H., 2002, *Interaction Design:Beyond Human-Computer Interaction,* John Wiley & Sons Ltd., USA.

Reas, C. and Fry, B., 2007, *Processing: A Programming Handbook for Visual Designers and Artists*, The MIT Press, USA.

Robertson, S., 2001, Requirements Trawling: Techniques for Discovering Requirements, *International Journal of Human-Computer Studies*, vol.55, no. 4, pp. 405–421.

Robinson, A., Chen, J., and Lengerich, E., 2005, Combining Usability Techniques to Design Geovisualization Tools for Epidemiology, *Cartography and Geographic Information Science*, vol. 32, no. 4, pp. 243–255.

Roth, R. E., Ross, K.S., Finch, B.G., Luo, W., and MacEachren, A.M., 2010, A User-Centered Approach for Designing and Developing Spatiotemporal Crime Analysis Tools, *6th International Conference on Geographic Information Science*, Zurich, Switzerland.

Snyder, C.,2001, *Paper prototyping* [online] Available from: http://www.cim.mcgill.ca/~jer/courses/hci/ref/snyder.pdf [Accessed 20 February 2013].

Tory, M. and Möller, T., 2004, Human Factors in Visualization Research, *IEEE TVCG*, vol. 10, no. 1, pp. 72–84.

Tullis, T. and Albert, B.,2008, *Measuring the User Experience*, Morgan Kaufmann Publishers, USA.

Ware, C., 2004, *Information Visualization: Perception for Design*, Morgan Kaufmann Publishers, USA.

Wassink, I., Kulyk, O., van Dijk, E.M.A.G., van der Veer, G.C., and van der Ver, P.E., 2008, Applying a User-Centered Approach to Interactive Visualisation Design, Trends in Interactive Visualization. Advanced Information and Knowledge Processing. Springer, London. pp. 175-199.

Wehrend, S. and Lewis, C., 1990, A Problem-oriented Classification of Visualization Techniques, *Proceedings VIS'90'*, IEEE Computer Society Press, San Francisco,USA, pp. 139–143.

Yao, J., Fernando, T., and Tawfik, H., 2006, Towards a Collaborative Urban Planning Environment, *CSCWD'05 Proceedings*, Coventry, UK,pp. 554–562.

Yi, J., Kang, Y. A., and Stasko, J., 2007, Toward a Deeper Understanding of the Role of Interaction in Information Visualization, *IEEE TVCG,* vol.13, no. 6, pp. 1224–1231.

STABLE INCREMENTAL LAYOUTS FOR DYNAMIC GRAPH VISUALIZATIONS

Martin Steiger, Thorsten May and Jörn Kohlhammer
Fraunhofer IGD, Fraunhoferstr. 5 - 64283 Darmstadt

ABSTRACT

In this paper we present a set of extension techniques to stabilize interactive dynamic graph layout algorithms. It works with different existing Focus & Context methods. We first deal with the initial placement of newly inserted nodes to mitigate acting forces in the layout algorithm. Then, their influence on the existing layout is gradually increased to create a smooth transition between the old and the new layout. To complement this approach we use a look-ahead strategy that integrates additional nodes in the layout to stabilize the layout even more.

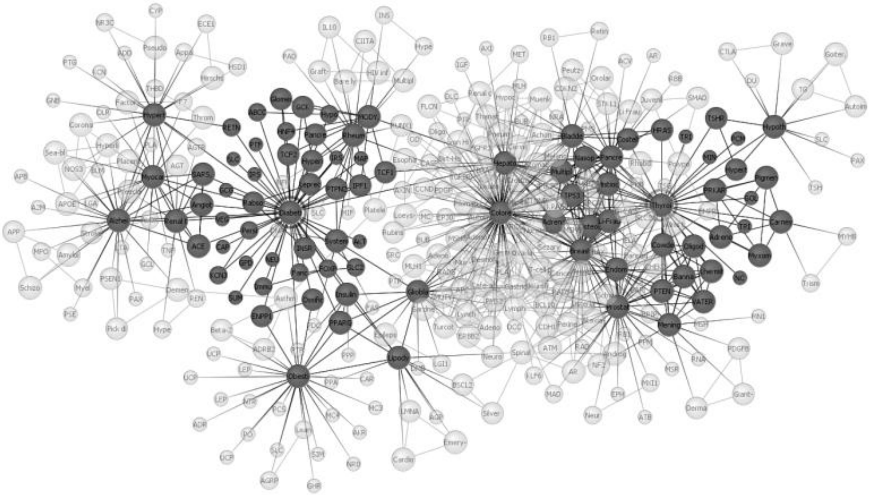


Figure 1. A screenshot from the running system showing a partial view of a graph including a set of focal nodes (marked with circles) and a set of visible context nodes. Also, a number of additional, potentially interesting ghost nodes (semi-transparent) are integrated to stabilize the layout, but not shown to the user.

KEYWORDS

Dynamic graphs, stable layout, mental map

1. INTRODUCTION

The tasks in the 2012 VAST Challenge[1] are about finding suspicious events that happened during just two days. The given data set describes states and connections of all computers in a fictive company network. The major problem here as well as in most real world scenarios is the sheer amount of data. In total, 160 million nodes are available in the database.

A visual representation of the data is vital to gain insight and decide what is suspicious and what is not. Obviously, displaying all entities and all interconnections at the same time cannot produce meaningful results. Screen space is always limited, especially compared to the ever-increasing amount of graph data, so

[1] http://www.vacommunity.org/VAST+Challenge+2012

massive overdraw is the consequence for such visualizations. But also visual perception has its limits. Even if it would be possible to draw millions of nodes, the data analyst would not be able to deal with that information. Typically, the user is interested in a small part of the graph, maybe even a single node. The challenge in the visualization is to preserve a context that is large enough for the user to orientate and to navigate towards potentially interesting nodes and to solve a given task.

Therefore, the layout of the displayed subgraph that is used in these techniques plays a key role. For small to medium-sized graph a pre-computed global layout with fixed node positions has its advantages. Above all, the visualization of a specified graph will always look the same which helps to build a mental map of the data set. In the navigation process, the nodes are simply toggled between visible and invisible state while their positions remain the same. But this also means that nodes that do not exist in the partial view influence the position of the visible nodes. This effect can be seen in Figure 2.

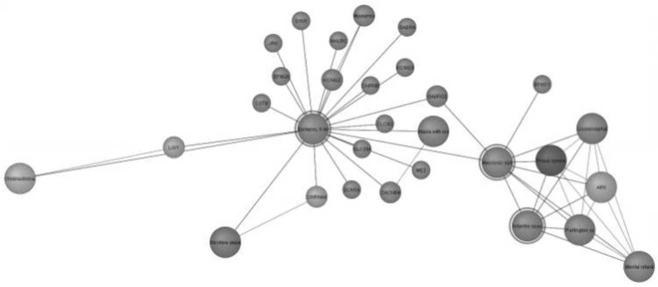


Figure 2. Some nodes are strongly pulled outwards without any visible explanation. They have to fulfill all constraints that are imposed by the full graph (adapted from May et al. with permission)

The time and memory limitations render global layout impractical for large graphs. Also, a fair amount of the computed layout is never used in the navigation process later on. One way to handle this is to perform the layout computation on-the-fly for the area that is currently visible. This process works quite well, if the displayed subgraph does not change over time. Otherwise, these topological changes can have a strong and sudden impact on the visual representation if they are not carefully dealt with. In order to communicate the performed changes it is important to create smooth transitions, for example through animation, so that the user can better understand which nodes are appearing or disappearing.

We claim to contribute a combination of features that improve the user experience with focus-based navigation concepts:

- A placement algorithm that puts new nodes close to their neighbors so that acting forces are kept minimal.
- A weight function that integrates new nodes in the existing layout in a smooth transition by gradually increasing their mass.
- An a-priori layout of all nodes that could become visible after the next focus node change.

These features can be used as separate improvements or in an integrated system. As a result, nodes are added and removed smoothly, keeping the visual changes in the existing layout to a minimum. We think that their effect on the layout reduces the cognitive effort for the user and thus support the preservation of the mental map.

The rest of the paper is organized as follows: In the next section we give a short overview on related work in the area. In Section 3 we present the data structures and algorithms we use before we go into implementation details in Section 4. Section 5 covers the test results and describes limitations. Finally, Section 6 concludes and discusses future work.

2. RELATED WORK

A survey on visual graph analysis in general has been put together by von Landesberger (Von Landesberger et al, 2011). It covers a variety of analysis techniques for node-link diagrams as well as matrix representations for partial graph views or dynamic graphs. A typical problem of some graph layout techniques is the at least partially random behavior caused by a random initialization of node positions.

Eades noted that the *visual order* of screen elements should not change for animated diagrams in order to prevent user irritation (Eades et al, 1991), but he original definition of the term "mental map" has been coined by Misue et al. (Misue et al, 1995). They discuss three concepts on mental models: Orthogonality, proximity and topology.

Based on these ideas, Purchase presented an empirical study that strengthens the assumptions on the relevance for the understanding of node-link diagrams (Purchase et al, 2007). Depending on the task, preserving the mental map seems to be more complicated than earlier works indicated (Purchase and Samra, 2008).

While most of these concepts have been developed for the layout of dynamic graphs, we conclude that they also can be applied to extracted, dynamic subgraphs of large static graphs. In both cases we can assume that a significant portion of the visible graph is topologically stable between any two changes. Different metrics on how to measure and optimize these changes has been presented earlier (Branke, 2001), but most of them compare only key frames of the full graphs. In this work, we will apply such measures to the animated transitions between key frames.

Mathematically speaking, navigation in partial graph views creates a series of subgraphs that are derived from a larger graph. Diehl and Görg presented a strategy for the transition between these subgraphs, but it requires that the full set of subgraphs is known in advance (Diehl, Görg and Kerren, 2001). Because user interaction cannot be predicted, this approach cannot be adopted “as-is” for interactive browsing. Instead, interaction is supported by morphing (Diehl and Görg, 2002). Our approach solved this problem in a different way. Instead of only reacting to the user interaction, we prepare the graph layout for different possible scenarios.

Lee et al. apply quality measurements to an optimization framework based on simulated annealing (Lee, Lin and Yen, 2006). While this approach is generic, the given performance suggests that simulated annealing might not be suitable to support interactive browsing and real-time feedback, however.

Osawa combines traditional mass-spring layout with heat models (Osawa, 2001) to create stable layouts. The user distributes virtual heat energy to one or more nodes. Thermal radiation then distributes the energy to neighbors. Depending on the amount of inherent energy, nodes become larger or smaller, edges shorter or longer. While this approach seems to be promising, it requires the user to manually tweak the visualization.

For graphs with inherent hierarchical structures, a clustering can be built up and used for the layout. This also works for dynamic graphs, as Frishman and Tal show with their work (Frishman and Tal, 2004). Visual changes in the layout are reduced by adding “spacer” vertices that reserve space for future nodes.

Some approaches make use of an evaluation function to extract interesting parts of the graph with respect to one or more selected vertices. This idea of a Degree of Interest function was brought up by Furnas who proposed to derive the set of most interesting points based on user interaction (Furnas, 1986). Heer and Card applied this concept to create a tree layout algorithm that works with multiple focus points (Heer and Card, 2004). Van Ham and Perer also showed that dynamic graph visualizations benefit from this approach (Van Ham and Perer, 2009). While their DOI function is based on a single focus point, May et al. extended the function to work with multiple focus points (May et al, 2012). However, their approach does not yet consider smooth transitions between the different extracted subgraphs.

3. DEFINITION OF GRAPH, CONTEXT AND FOCAL NODES

In mathematical terms, the data we work with has the form of a connected graph. It is defined as G(V, E) where V is a set of nodes or vertices and a set of edges E that connect some pairs of vertices. Our approach works with both directed and undirected graphs without limitations.

The technique we present is based on the browsing paradigm. This means that new nodes are reached through their neighbors. If the graph contains parts that are not connected to the rest of the graph, they cannot be reached. For the rest of the paper we will assume that the graph is connected, i.e. a path exists for every pair of vertices.

Based on the graph G we will extract the focus $F \subset G$, a subgraph that is displayed to the user as the visible part of the full graph. It is constructed from the focal nodes $Z \subset V$, a small set of vertices in G and a degree-of-interest function that defines the visible neighborhood of F. The focus nodes are initially selected by the user (e.g. through a textual search query) and thus considered to be the most interesting points for the analysis. The subgraph F surrounds the focal nodes and thus provides a context. The definition used here is analogous to the definition in the work of van Ham and Perer (Van Ham and Perer, 2009), but we decided to use the multiple focus points approach as described by May et al. (May et al, 2012) for two reasons.

First, changing a single focus point keeps most of the context as it is induced by the other nodes. Typically, the set of focal nodes is implemented as a first-in-first-out (FIFO) queue with limited size. Every time the analyst puts a new node in focus, it is added to the queue of focus points. It also adds its interesting neighbors to the visible graph. When the queue is full, the least-recently used focus node is dropped from the queue and its neighborhood is removed from the view. Using more than one focus node also keeps changes in the visual structure to a minimum which preserves the context better than the single-focus-version.

The second reason is that it also gives a sense of history which is particularly useful for browsing tasks where the path to the solution is not known beforehand or part of the solution. After the focal nodes have been picked the next step is to derive the context. To achieve that, a degree-of-interest function is used to evaluate the relevance of a node with respect to the current focus.

3.1 Initial Node Selection

The initial pick of focus nodes in a large-scale graph is a non-trivial task. Following the “Overview first, zoom and filter, then details-on-demand” paradigm by Shneiderman we notice that the overview is not available in partial graph visualizations (Shneiderman, 1996). In order to find potentially interesting regions of the graph, we cannot drill down from a high-level view to a specific region that demands our interest. We have to explicitly perform a search query that iterates over the full graph and select one or more of its results as a starting point. For example, a text search could provide all nodes whose attributes that match the expression. Alternatively, picking a keyword from a predefined list could select all vertices that are associated with that keyword.

3.2 The Degree of Interest Function

The concept of a function that evaluates the interest of a data element with the respect to the observer was first presented by Furnas (Furnas, 1986). The larger the distance from the current viewport the smaller seems to be the importance for the current task. Such a DOI function typically contains the following parts:

- An a-priori interest *(API)* that reflects the user-independent, general interest of a graph node *x*.
- The user interest *(UI)* function that represents the matching score to the user’s current task *t*.
- It is complemented by a distance function *D* that measures the distance to the focus nodes in *Z*.

Denoting the set of search parameters as *t*, the resulting function can be expressed as:

$$DOI(x, y, z) = \alpha \cdot API(x) + \beta \cdot UI(x, t) + \gamma \cdot D(x, Z)$$

Furnas presented several possible mappings for the parts of this function. For example, the API function can be generated from inherent attributes of the node or its relevance in the structure. The UI function could match a node to a user defined text search query and the distance function can be mapped directly to the minimal weighted or unweighted distance in a graph.

We will briefly sketch how the contextual subgraph can be constructed. A modified version of the Dijkstra algorithm can be used to derive the visible nodes around the focus points. Instead of using a single starting point, this variation works with multiple starting points, namely the focal nodes. The DOI of neighboring nodes is used as edge weight. In every iteration, the node with the highest interest is added to *F*. Typical stopping criteria are the number of total nodes in the set or a predefined threshold that filters out nodes with a DOI lower that a certain value. As it is, this function leads to disconnected graphs which are undesirable. A connected subgraph can be created by adding the shortest path between all focal nodes to the visible graph.

4. COMPUTING THE INCREMENTAL LAYOUT

We employ a classic force-directed layout algorithm based on the mass-spring-model that is described by Fruchterman and Reingold (Fruchterman and Reingold, 1991). Its main actors are spring tension and repulsion forces. For the initial set of nodes no placement constraints exist. Also, for subgraphs that are not connected to the rest, random placement can be used. All other nodes are at least constrained by edge forces that act between neighbors.

4.1 Placing New Nodes

The key idea here is to avoid sudden changes in the visualization through reduction of acting forces. Selecting a visible node as focus point creates a new, different set of context nodes. In order to keep the layout stable, new nodes should be placed with a distance equal to the rest length of the edge to satisfy spring constraints. However, it also desirable to place nodes as far away from all other nodes as possible to make use of free space and avoid repulsion forces.

The first aspect can be easily satisfied if the new node has only one edge. It can be placed anywhere on a circle with a radius equal to the desired edge length around the existing node. If two distance constraints exist the range of valid positions is reduced from a full circle to two points on that circle (see Figure 3a).

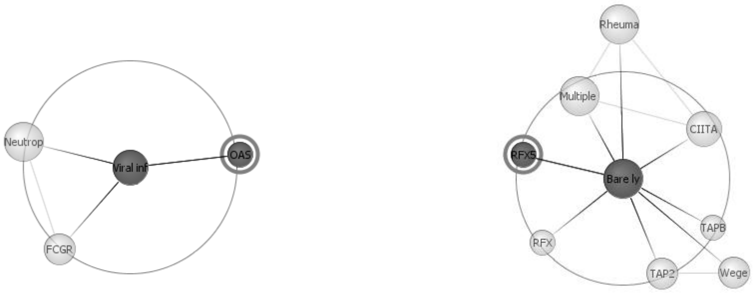


Figure 3. a) Up to two connected nodes (semi-transparent) can be placed on the circle around the existing vertex
b) Additional nodes are placed on a circle around the center of gravity of already inserted nodes

For nodes that have links to more than two neighbors, no general solutions exist that fulfills all criteria. We thus have to approximate the ideal position. First, the center of gravity of all neighbor nodes that are already in the layout is computed. Then, the final node position is a position on the circle around the center of gravity with a radius of the desired edge length (see Figure 3b). This node potentially lies outside the previously mentioned circle and does not fulfill any of the imposed constraints, but it is pushed away from the areas with high node density.

The placement on the circle is performed with respect to the number of nodes in its proximity. This area-based density measure is also required for the computation of repulsion forces of the Fruchterman-Reingold algorithm. Often-used methods to find empty space and perform distance tests quickly are spatial hashing algorithms. They define a grid-like structure of buckets that contain the graph vertices based on their position. Nodes are added and removed from the buckets as the move over the virtual grid. Given a point location the algorithm retrieves its corresponding bucket (based on hash keys). All other nodes in the bucket are considered to be close enough for further distance testing. Depending on the search distance, adjacent buckets are inspected, too.

We can make use of this data structure and query different positions on the circle. We count the number of nodes in the search area of every query location and choose the one with the lowest hit count. This ensures that new nodes are placed where most space is available.

Vertices that fall out of scope are removed from the visible graph. Typically, these nodes have been in the layout for a while and thus have “settled”. The forces that act on them are rather small. Consequently, removing such a node hardly affects acting forces on other nodes and can thus be removed without having a strong visual impact.

4.2 Adding Invisible Neighbors

When new nodes are inserted in the layout and displayed immediately, they typically move a lot in order to satisfy all repulsion and edge forces. This can be mitigated by meaningful placement of new nodes but not cured entirely.

This is why we propose a look-ahead technique that also adds nodes to the layout that could be become visible, if one of the currently visible nodes was selected as a focus point. Every time a node is added to the visible graph, the DOI function is evaluated for its neighbors as if the node was in the queue of focal nodes.

All nodes that would then become visible have already been added to the layout subgraph before. However, this subgraph can quickly become quite large, especially for highly connected graphs if the DOI function does not filter out most of the nodes.

4.3 Gradually Increasing Node Weights

When many new nodes are placed around an existing node, the existing node will jitter heavily until the newly inserted spring forces have relaxed. We thus modify the algorithm by adjusting the mass of vertices over time. Following the “separation of concerns”-pattern we first create a function that provides the number of layout iterations for every node. We track this by counting the write operations in the layout on a per node basis. Removing a node sets its counter back to zero.

We then derive the node weight from the normalized values of the counting function. It is desirable that invisible nodes reach the mass of visible nodes smoothly as the mass of a node is in direct relation with the acting forces that are used in the layout algorithm. For two vertices with mass m_1 and m_2 the distribution of forces can be computed as:

$$f_1 = \frac{m_2}{m_1+m_2} \qquad f_2 = \frac{m_1}{m_1+m_2}$$

The larger the mass of a particle is, the weaker is the influence of the force and the stronger is the influence on its counterpart. The two functions are plotted in Figure 4.

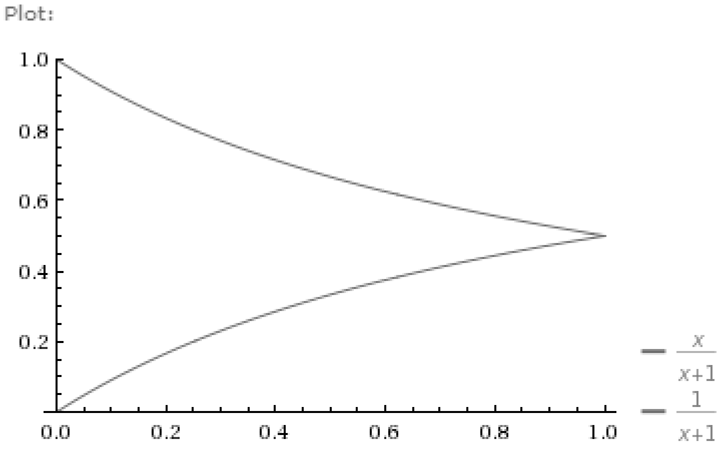


Figure 4. The weight function starts with full effect on the newly inserted node. The influence is shared as the number of layout steps increases until equilibrium is reached and both nodes are equally affected.

A similar idea has been presented by Huang et al. who introduce friction forces to stabilize dynamic layouts. The longer a node is visible the large its friction coefficient gets (Huang et al, 1998).

5. TEST RESULTS

The graph data that is used for the tests has been taken from the Diseasome[2] project. We extracted the largest connected subgraph (disconnected parts cannot be reached with exploratory search) which contained about 1500 nodes and 5500 edges. Although the edges per node ratio has an influence, our approach works on partial views only and is thus independent of the graph size per se.

Starting at a randomly chosen focus node, we explored its neighborhood over six focus changes, keeping all explored nodes visible. This created a sequence of six keyframes with an increasing number of visible nodes (from 4 to 69). Using this dataset, we run the different layouts for 100 iterations before switching to the next subgraph.

To test our methods we measured the node velocity of all visible nodes in three different settings. Few nodes with high velocities should implicate a larger penalty than many slow movements. This is why we measure the average of squared velocities.

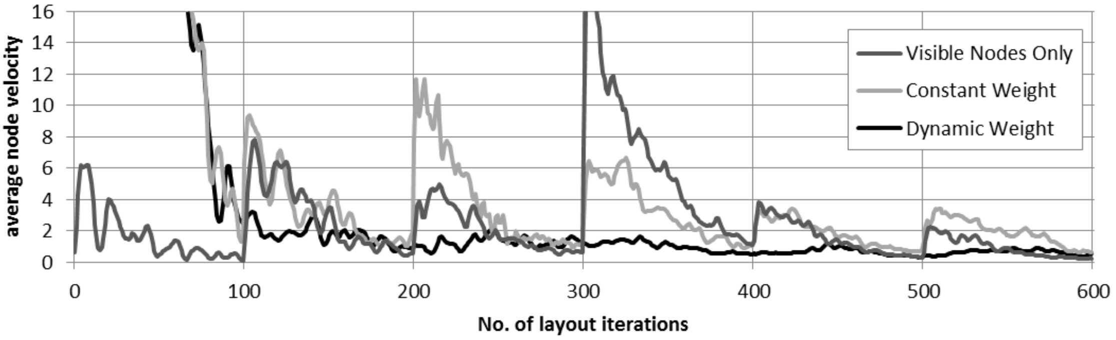


Figure 5. Node velocities for three different setups measured over six transistions with 100 layout steps each. Including invisible neighbors reduces velocity peaks (light gray). Using dynamic node weights reduces the peaks even more as the influence of invisible nodes is still very small after their insertion.

The first measurement (in medium gray) was performed with constant node weight and without the invisible neighbor layout. It performs best during the first frame as only 4 nodes need to be laid out. However, the fourth subgraph contains a star node that introduces a large number of new nodes which causes the peak at iteration 300.

The second test run (in light gray) also included the layout of the invisible neighborhood. The first subgraph contained additional 15 nodes that exert forces on the visible 4 nodes causing a rather poor performance for the first 100 iterations. At the start of frame 3 (iteration 200) a fair amount of nodes that could become visible in frame 4 has already been added. This causes a similar peak as in the first run, but less high and shifted by one keyframe. The peak at the begin of the forth frame is thus significantly smaller.

The gradual increase of node weights has also been activated for the third test run (in black). As before, the average velocity for the 4 nodes in the initial frame is very high, but once they have settled down, it outperforms the other methods, especially when a large number of nodes is added simultaneously.

6. CONCLUSION & OUTLOOK

In this paper we describe contributions that increase the quality of incremental layout algorithms. A set of nodes that will possibly become visible in the foreseeable future is already integrated in the layout computation to keep the visual representation stable when they are shown. After they have been placed close to their neighbors, their influence on the layout is gradually increased from almost none to full effect. This keeps the transition between different focal node sets smooth.

The look-ahead strategy we propose significantly improves the layout. Star nodes that are visible have many invisible neighbors that pull the node outwards. This indicates that the vertex is highly connected and

[2] http://diseasome.eu/map.html

possibly worth exploring, even if the connection cannot be seen. On the downside, a high edge to node ratio introduces a large number of nodes on the layout which can cause slowdowns.

Currently, navigating from one part to another and back does not necessarily produce the same visual structure. Whenever a part of the graph is removed from the visible context, the positions of the nodes become invalid. When the nodes are then again added to the context they are attached in a best-fit manner to the existing layout. Reusing the old positions would not help, because the position of the current visual context is different. In future research, we plan to develop a layout that produces repeatable structures and paths to these structures.

REFERENCES

J. Branke, 2001. Dynamic Graph Drawing. *Methods and Models, Drawing Graphs. Lecture Notes in Computer Science(2025)*, pp. 228-246

S. Diehl, C. Görg, A. Kerren, 2001. Preserving the Mental Map using Foresighted Graphlayout. *Proceedings of the Joint Eurographics IEEE TVGC Symposium on Visualization (VisSym)*, Wien, New York, Springer Verlag. pp. 175-184

S. Diehl, C. Görg, 2002. Graphs, They Are Changing, *Revised Papers from the 10th International Symposium on Graph Drawing*, London, UK, pp. 23-31

P. Eades, 1991. *Preserving the Mental Map of a Diagram*. International Institute for Advanced Study of Social Information Science

Y. Frishman, A. Tal, 2004, Dynamic Drawing of Clustered Graphs. *IEEE Symposium on Information Visualization, INFOVIS*, pp. 191-198

T. Fruchterman, E. Reingold, 1991. Graph Drawing by Force-directed Placement. *Software: Practice and experience,* vol. 21(11), New York, USA, pp. 1129-1164

G. W. Furnas, 1986, Generalized Fisheye Views. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* Boston, USA, pp. 16-23

F. van Ham, A. Perer, 2009. "Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest, *IEEE Transactions on Visualization and Computer Graphics*, vol.15, no.6, pp.953-960.

J. Heer, S.K. Card, 2004. DOITrees revisited: Scalable, Space-constrained Visualization of Hierarchical Data. *Proceedings of the working Conference on Advanced visual interfaces*. New York, USA, pp. 421-424

M. L. Huang, P. Eades, 1998. A Fully Animated Interactive System for Clustering and Navigating Huge Graphs, *Graph Drawing 6th International Symposium, GD' 98,* Montréal, Canada, pp. 374-383

M. L. Huang, P. Eades, J. Wang et al, 1998. Online Animated Graph Drawing using a Modified Spring Algorithm, *Journal of Visual languages and Computing*, vol. 9(6), pp.623-645

T. Kamada, S. Kawai, 1989. An Algorithm for Drawing General Undirected Graphs. *Information Processing Letters,* vol. 31(1), pp. 7-15

J.B. Kruskal, J.B. Seery, 1980. Designing Network Diagrams, *Proceedings of the First General Conference on Social Graphics*, Washington DC, USA, pp.22-50

T. May, M. Steiger, J. Davey and J. Kohlhammer, 2012. Using signposts for navigation in large graphs. *Computer Graphics Forum. v*ol.31(3/2), pp. 985-994

K. Misue, P. Eades, W. Lai, K. Sugiyama, 1995. Layout Adjustment and the Mental Map. *Journal of visual languages and computing,* 6(2) pp.183-210

T. von Landesberger, A. Kuijper, T. Schreck et al., 2011. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum* (2011), vol. 30(6), pp. 1719-1749

Y.Y. Lee, C.C. Lin, H.C. Yen, 2006. Mental Map Preserving Graph Drawing using Simulated Annealing. *Proceedings of the 2006 Asia-Pacific Symposium on Information Visualization,* vol. 60, Darlinghurst, Australia, pp. 179-188

N. Osawa, 2001. A Multiple-Focus Graph Browsing Technique using Heat Models and Force-directed Layout. *Proceedings of the Fifth International Conference on Information Visualisation*, pp. 277-283

H. C. Purchase, E. Hoggan, C. Görg, 2007. How Important is the "Mental Map"? – an Empirical Investigation of a Dynamic Graph Layout Algorithm. *Proceedings of the 14th International Conference on Graph Drawing*. Berlin, pp.184-195.

H. C. Purchase and A. Samra, 2008. Extremes are better: Investigating Mental Map Preservation in Dynamic Graphs. *Proceedings of the 5th international Conference on Diagrammatic Representation and Inference*, pp. 60–73.

B. Shneiderman, 1996. The eyes have it: A task by data type taxonomy for information visualizations. *Proceedings of the IEEE Symposium on Visual Languages*, Washington DC, USA, pp. 336-343

EFFICIENT BROWSING IN IMAGE DATABASES USING A HIERARCHY OF KERNEL PCA SUBSPACES

Marcel Spehr[1], Frank Herrlich[1], Stefan Hesse[2] and Stefan Gumhold[1]
[1]*Technical University of Dresden - Computer Graphics and Visualization Lab - TU Dresden, Fakultät Informatik,*
Nöthnitzer Straße 46, D-01187 Dresden
[2]*SAP AG - Dietmar-Hopp-Allee 16, 69190 Walldorf, Germany*

ABSTRACT

We present a novel approach for designing the search functionality in large unlabeled image databases. It combines *Relevance Feedback*, *Hierarchical Browsing* and *Kernel PCA*, uses a *Mixture-of-Gaussian* to model feature space distributions and different visualization techniques of high dimensional feature spaces. Given an image database, finding a specific single or set of pictures is achieved by assisting the user to find an as-short-as-possible browsing path through the database. Our system relies on describing each picture with an appropriate feature vector that results from applying *Kernel PCA* to image and textual based similarity matrices. We solve the page-zero-problem by presenting the centroids of a hierarchical clustering in feature space as initial suggestions. The user can then steer the search by selecting positive and negative examples which define a *Mixture-of-Gaussian* density in the parameter space. New suggestions are drawn according to this density and the user is thus directed to the desired image category. A user study proved our system to be practical and beneficial for *category search* tasks.

KEYWORDS

CBIR, image similarity, relevance feedback, Kernel PCA, image features, semantic gap

1. INTRODUCTION

With the establishment of consumer digital photography private photo collections grow larger each year. Collections exceeding 10.000 images are common. This leads to the demand for new organization and navigation schemes to access the content of these image database management systems (*IDBMS*).

Programmers implementing an *IDBMS* face different problems. How to measure the similarity of images? Given limited screen size, which visualization techniques are most adequate to support searching and presentation of search results? How can user interaction be incorporated in subsequent search steps? Here we propose a combination of well known and established techniques to build an integrated system for efficiently browsing large databases of images.

We suggest a component based approach to give the system's developer as much flexibility as possible for implementing extensions. We also keep in mind that the final assembly of parts should hide the additional complexity from the user, who is just interested in the browsing capabilities. The components are responsible for (1) measuring similarity between images, (2) visualization of feature spaces, and (3) supporting the user's interaction. Figure 1(a) outlines (1) and (3) of our approach.

Component (1) derives appropriate feature descriptors for the images. Our system shall be as flexible as possible to be applicable in diverse application scenarios. We enable the user to supply a set of arbitrary image similarity measures as suit his needs. The resulting similarity matrices are henceforth processed in a *Kernel PCA* to achieve metric feature spaces that contain image descriptors. This procedure decouples the special case of a particular image database and use case from the rest of the search pipeline.

Visualization of feature space properties (2) is central to our needs. In our application scenario (see ”Case Study” in section 4), the user is confronted with a large database of paintings and some more or less understandable abstract descriptors of them. The visualization component is responsible for conveying their meaning to the user in an understandable way. We implemented three data views: a) ordered regular grid, b) star charts, c) parallel coordinates.

There are two common ways of formulating a query to an *IDBMS* - either by text or a/several query image/s. In our scenario we want to refrain from using textual or label information to achieve a multipurpose system. Therefore we work with queries by example. The system presents suggestions from which the user chooses interesting and uninteresting images to approach his goal as R*elevance Feedback* (3).

The remainder of the paper starts with an overview of earlier work in the fields of *Relevance Feedback* and a discussion of ways to model high dimensional feature spaces as well as visualization techniques to navigate them. Then we start to describe our approach by first stating the system requirements that our solution is based on. The *Kernel PCA* algorithm that produces the image descriptors and our hierarchical browsing scheme is shortly introduced. Section 3.4 presents the way we map the user's intention to a parametrizable probability distribution in feature space using a *Mixture-of-Gaussian* model. We conclude the discussion of our work by introducing the visualization components that support the search task and shortly discuss the additional usefulness of the user's interaction data for supervised classification tasks. The final part of this paper presents a user study in which a database of 20.000 paintings and a set of 10 similarity measures is used in a *category search* scenario.

Our work's contributions are as follows. We designed a search system that distinguishes itself by being: I. Easily extensible by consistently treating arbitrary symmetric similarity measures. II. Scalable with respect to several of its components. And III. Flexibly applicable with the help of different views. To our knowledge the used techniques were never employed in this combination to solve the image retrieval task.

2. RELATED WORK

Since its early ancestor, the QBIC system (Flickner et al. 1995), content based image retrieval systems have come a long way. Due to the huge number of works in the area of navigation in high dimensional spaces for image retrieval we will only summarize the key ideas and works of those fields that inspired our work. Recent advances in content based image retrieval are thoroughly recaptured by (Datta et al. 2008).

Browsing techniques are presented in the work of (Matkovic et al. 2009) on visual analysis techniques in feature space. They provide standard tools like parallel coordinates to brush coordinate subspaces but have limited navigational capabilities. (Bartolini et al. 2007) focus on *personalization actions* as facilities to adapt the local browsing structure and thus enhancing the personal experience a user has while using the application. (Moghaddam et al. 2004) place images according to their pairwise similarity in a 2D context. They allow user adaption of the similarity values. (Ding et al. 2008) exemplify how a hierarchical browsing algorithm can speed up image retrieval significantly.

Many attempts on improving image search with more sophisticated visualization techniques can be found in the literature. (Hedman et al. 2005) compare a fish-eye view on the image space with different standard views off the data. (Combs & Bederson 1999) analyze if zooming improves image search. (Moghaddam et al. 2004) work on the integration of user models for improving visualization systems for personal photo libraries. (Brivio et al. 2010) finally is a recent approach to embed thumbnail images in a 2D map based on weighted Voronoi diagrams. For a more elaborate overview of different display, summarization and exploration techniques we refer the reader to (Camargo & González 2009).

(Rui et al. 1998) introduced *Relevance Feedback* as *Rocchio's algorithm* to the field of image retrieval. After that it became an excessively employed technique in the CBIR context. (Meilhac & Nastar 2002), (Su et al. 2003) and (D. Liu et al. 2006) worked on adapting the idea to different application scenarios and improving it. For a detailed overview of the current state of the art please refer to (Thomee & Lew 2007).

Incorporation and learning from user interaction data can be done in multiple ways. I. e. (Cox et al. 2002) demonstrated the usage of a Bayesian framework with PicHunter. (Fogarty et al. 2008) let the user re-rank search results and thereby influence future retrieval orders. (Campbell 2000) on the other hand explicitly model the uncertainty a user experiences when judging the relevance of examples images. We followed the approach of (Qian et al. 2002) and modeled the distributions in feature space by using a *Mixture-of-Gaussian* in combination with *Relevance Feedback*. We combined their scheme with the approach of (Daoudi et al. 2008) who utilize a configurable *Kernel PCA*. We further enhanced our system by supplying an appropriate visualization and interaction engine for the image search.

The basic and most difficult problem of all feature based image retrieval systems is that they must bridge the *semantic gap* between abstract features and semantic meaning. (Yang et al. 2006) can serve as exemplary for a key idea in this area - the propagation of already semantically labeled data to unlabeled data by automatic algorithms. Section 3 details our approach for its solution. For an overview of the current state of the art (Y. Liu et al. 2007) provide a good start.

3. OUR APPROACH

3.1 Design Requirements

In professional contexts like medical or engineering sciences large bodies of image data pose no exception any more. Since hardware costs continued to decrease for a long time storing large sets of images became also feasible for the average computer user. Given these constraints scalability is the first major issue a modern semi-automatic image retrieval system must address. For a holistic system we must tackle it at different stages. We do so by on the one hand employing a hierarchical browsing procedure which allows the user to first restrict his search to a certain subspace of the whole feature space. On the other hand the dimensionality of the feature space is adjustable. This is achieved by choosing different parameter values for the *Kernel PCA*.

Easy extensibility of the set of similarity measures that is used to define the final feature vector for each image is also indispensable. Here we argue that the simple inclusion of additional fixed metric feature vectors is not flexible enough for describing arbitrary distances. The usage of a *Kernel PCA* solves this constraint. Since the system simply expects a similarity matrix as input, it is very easy from a developers point of view to use all information he sees fit for a problem specific set of similarity measures between images. Hence, because of its consistent interface the employment of our technique for diverse image domains is straightforward. Online dimension reduction to a variable degree is in principle possible but not yet implemented for our system.

Finally the system's benefit is directly linked to its intuitive usability. We try to keep the interface to the system as simple as possible. In the standard settings the user only faces a canvas of images. If he is more experienced, additional visual analysis techniques of our system can be used.

3.2 From Arbitrary Similarity Measures to Image Descriptors in $\Re^n$

As mentioned before an image retrieval system's power lies in the integration of different similarity measures $s^i(I_k, I_l)$, $i \in 1...m$ between images I_k and I_l. In the remainder of this section the superscript i denotes the i th similarity measure. Here we only demand that $s^i(I_k, I_l) = s^i(I_l, I_k)$ and$s^i(I_l, I_l) \geq s^i(I_k, I_l) \forall k \neq l$. These m measures can be defined on the image signal directly, be feature based, take textual annotations into account or be based on studies in which human subjects were asked to rate perceptual distances between images.

Using these similarity functions $s^i(I_k, I_l)$ pairwise between all N images in the database produces m $N \times N$ dimensional quadratic similarity matrices S^i. Like (Bishop 2006) we interpret these values as dot products in an arbitrary feature space. An eigenvalue decomposition of each S^i delivers m sets of eigenvectors V^i (here each V^i denotes a matrix whose rows are the eigenvectors) and vectors of eigenvalues Λ^i. Compression can now be achieved by choosing just the r^i rows of V^i with the largest corresponding eigenvalue in Λ^i.

Obviously this procedure does not scale well. The size of the S^i grows quadratically with the number of images in the dataset. We address this issue by using a subset of q^i data points. Let $S'^i \in \Re^{q^i \times q^i}$ be the similarity matrices of this subset, V'^i their eigenvectors and Λ'^i their eigenvalues. Let further be $D^i \in \Re^{N \times q^i}$ the submatrix of S^i that contains in each row the similarity values from all images to the subset's images.

The new image features now result from

$$\underbrace{\begin{pmatrix} D^i_{11} & \cdots & D^i_{1q^i} \\ \vdots & \ddots & \vdots \\ D^i_{N1} & \cdots & D^i_{Nq^i} \end{pmatrix}}_{D^i} \times \underbrace{\begin{pmatrix} V'^i_{11} & \cdots & V'^i_{1r^i} \\ \vdots & \ddots & \vdots \\ V'^i_{q^i1} & \cdots & V'^i_{q^ir^i} \end{pmatrix}}_{V^i} = \underbrace{\begin{pmatrix} f^i_{11} & \cdots & f^i_{1r^i} \\ \vdots & \ddots & \vdots \\ f^i_{N1} & \cdots & f^i_{Nr^i} \end{pmatrix}}_{F^i} \tag{1}$$

This illustrates how we can adapt our procedure to the number of data values by choosing appropriate values for the r^is and q^is according to the capabilities of the machine that runs our application. The value of q^i only affects the feasibility of S'^is eigenvalue decomposition. For our case study we used a subset of $q^i = 500$ of 20.000 data points. Choosing r^i (the number of rows of V^i) depends on the storage capabilities of the system and the requirements to the quality of the final conserved data variance in feature space. A full discussion of the choice of r^i is out of scope for this paper. We refer the interested reader to the terms elbow criterion that analyzes the percentage of variance explained by the reduced data respectively the *Akaike information criterion* that weighs model complexity to model accurateness. In our implementation we suffer an additional constraint because we are holding all F^i in main memory. Though by using an out-of-core data structure that would not be necessary.

We achieve equal data ranges in each of the r^i new feature dimensions by normalizing with the inverse squares of the corresponding eigenvalues in Λ'^i. This corresponds to a data whitening process which facilitates further image browsing and feature evaluation methods.

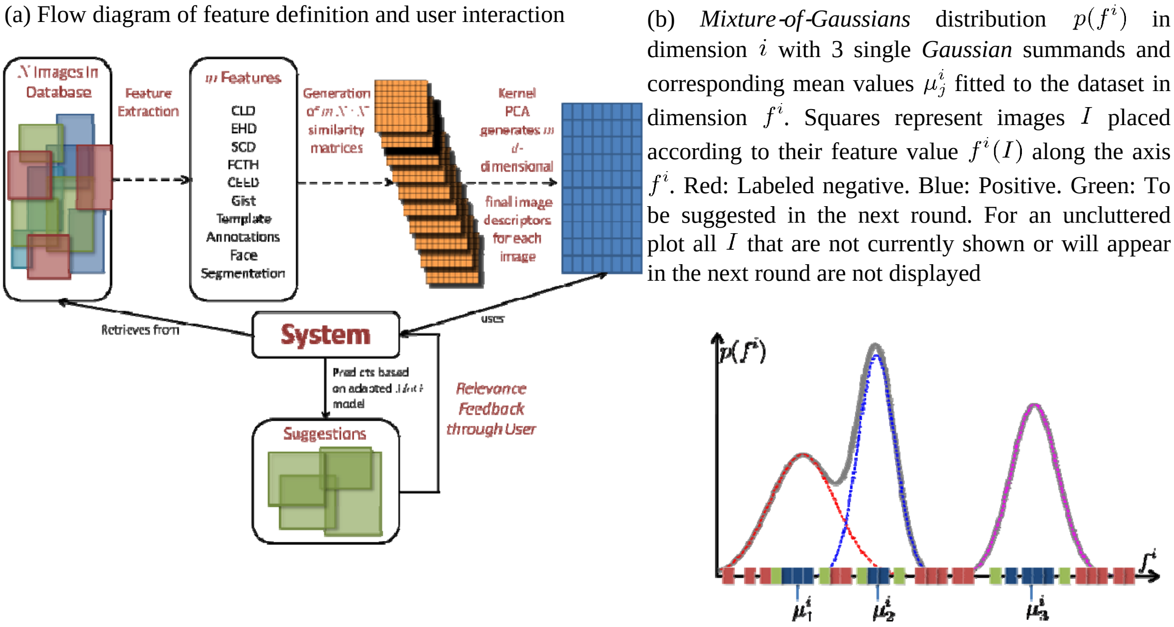
3.3 Hierarchical Browsing and the Page-Zero-Problem

At this stage of the discussion we suppose that each image I is affiliated with a set of feature vectors $f^i(I) \in \Re^{r^i}$ in metric feature spaces. The last paragraph showed how we adapt the dimension of the data points to our needs. Yet, the mere number N of images poses problems. A hierarchical procedure suggests itself. We assume the user's search pattern in feature space for our application to be from coarse (rough similarities to target category) to fine (fine tuning of result within target category). We support this browsing pattern by offering different, increasingly large browsable subsets of the image set. These subsets are constituted by the centroids of a *KMeans* clustering with an increasing number of k for each hierarchy level. At the very bottom all images are reachable. This also elegantly solves the page-zero-problem. The initial suggestions of our system are the centroids at the highest hierarchy level. This clustering has to be done only once. Adding additional images to the database can partly be compensated by making them accessible at the lowest level.

Having scalability in mind as a main system requirement, usage of a global clustering algorithm like *KMeans* usually forbids itself. We overcome this issue by running the clustering algorithm on a reduced representation of the data points. We do this by first running a standard PCA on the covariance matrix $C = F^T F$ given by the f^i. Let

$$F(I) = \begin{pmatrix} F^1 & \cdots & F^m \end{pmatrix} \in \Re^{N \times \sum_{i=1}^m r^i} \tag{2}$$

We preserve only the 3 dimensions with highest variance and thus achieve sufficient data reduction. Furthermore we partition this data in an axis parallel fashion in each dimension by creating an octree. Each octant is afterwards clustered independently. The level of the octree can be adapted to N. For our use cases a level of 3 was sufficient.


N Images in Database
Feature Extraction
m Features
CLD
EHD
SCD
FCTH
CEED
Gist
Template
Annotations
Face
Segmentation
Generation of m $N \cdot N$ similarity matrices
Kernel PCA generates m d-dimensional
final image descriptors for each image
Retrieves from
System
uses
Predicts based on adapted MoG model
Relevance Feedback through User
Suggestions
$p(f^i)$
f^i
μ_1^i
μ_2^i
μ_3^i

(a) Flow diagram of feature definition and user interaction

(b) *Mixture-of-Gaussians* distribution $p(f^i)$ in dimension i with 3 single *Gaussian* summands and corresponding mean values μ_j^i fitted to the dataset in dimension f^i. Squares represent images I placed according to their feature value $f^i(I)$ along the axis f^i. Red: Labeled negative. Blue: Positive. Green: To be suggested in the next round. For an uncluttered plot all I that are not currently shown or will appear in the next round are not displayed

Figure 1. Models of retrieval system (a) and parameterization of feature distribution (b)

3.4 Guided Browsing using Relevance Feedback and a Mixture of Gaussian Density

Usually the specific semantic context that the user has in mind for his search defines manifolds $T^i \subset \Re^{r^i}$. The goal of the concept search is to map out and deliver all images I with $f^i(I)$ within these manifolds. How can this search be visually guided by the system and reachability of all images guaranteed? We assume, that these manifolds can be approximated with *Mixture of Gaussian* probability distributions p^i. We estimate the parameters of these distributions based on the user feedback. This *Relevance feedback* enables the system to guess the user's intentions and thus leading him to the desired image - respectively image category - by sampling image suggestions from feature space according to the p^is.

The problem we are trying to solve can also be stated as defining a set of gradients in the direction where the desired image subset lies in the feature space. Therefore abstract feature dimensions originating from the dimension reduction must be made available to the user. As already mentioned our user interface mainly consists of a canvas that presents images suggested by the system. The user can decide which ones correspond to his search target and label them as positive. He can also label every image as negative that would lead him away from his target. This input is used to "learn" the T^is. Since our assumption states that at least some of the features must correspond to meaningful object categories that are of interest to the user the probability to find a desired image in some of the $\Re^{r^i}$ must not be uniform. Otherwise our system would obviously fail. We decided to describe the current estimate of the target distribution with a *Mixture of Gaussian* model (*MoG*, see figure 1(b)). Once defined it is easy to analyze.

There are multiple ways how such *MoG*s can be defined in the feature spaces $\Re^{r^i}$. Each positive sample could define one p_j^i. This would lead to a very fine granular model which is most possibly not the desired outcome. Finding the appropriate number of modes of the distribution for an aggregate approach usually involves probing for different candidate numbers of modes k^i by running first the *K-Means*-, then an *Expectation Maximization*-algorithm and finally deciding with Akaike or Bayesian information criterion which value of k^i defines the model complexity best to explain the data.

We use an alternative technique that integrates the negative samples. (Zhou & T. S. Huang 2003) state that most systems ignore them. The reason being, that in high dimensional feature spaces positive examples might well describe one single class, but negative samples usually stem from many different classes. From that it follows that the user can never label enough negatives to describe all of them.

In contrast we use the negative examples as separators between the p_j^is. We assure that each p_j^i is defined by positive samples, that are not separated by negative samples in *any* dimension. This results in different k^i for each iteration but our experiments proved this procedure to be quite effective.

Let a_j^i be the number of positive samples that define p_j^i divided by the number of all positively labeled images. Let further be μ_j^i their r^i-dimensional vector of mean values and Σ_j^i their $r^i \times r^i$-dimensional covariance matrix.

$$p^i(f^i) = \sum_{j=0}^{k^i-1} a_j^i \cdot p_j^i(f^i, \mu_j^i, \Sigma_j^i) \tag{3}$$

with $\sum_{j=0}^{k^i-1} a_j^i = 1$. The p_j^i are given by

$$p_j^i\left(f^i, \mu_j^i, \Sigma_j^i\right) = \frac{1}{(2\pi)^{\frac{r^i}{2}} |\Sigma_j^i|^{\frac{1}{2}}} \quad \exp\left(-\frac{1}{2}\left(f^i - \mu_j^i\right)^T {\Sigma_j^i}^{-1}(f^i - \mu_j^i)\right) \tag{4}$$

In each iteration of user feedback this distribution is adapted. The number of selected positive examples by the user is usually smaller or about the same size as the generally high feature dimension. According to (Hoffbeck & Landgrebe 1996) this generally renders the covariance matrices Σ_j^i meaningless. We deal with this problem by assuming independence between feature dimensions which results in diagonal matrices Σ_j^i. This would correspond to the assumption of independence between the features within the image class and we can model

$$p_j^i\left(f^i, \mu_j^i, \Sigma_j^i\right) = \prod_{h=1}^{r^i} \frac{1}{\sqrt{2\pi}\sigma_h^i} \exp\left(-\frac{\left(x - \mu_h^i\right)^2}{2\sigma_h^{i2}}\right) \tag{5}$$

Since it is very easy to fit this simple product of one dimensional *Gaussians* the performance of our system benefits.

Once the p^is are estimated one could start sampling from the set of all N images and use rejection sampling based on the probability of these images according to p^i to find new suggestions. This procedure will invariably be slow. We greatly enhance its speed by using *the Approximate Nearest* Neighbor library of (Mount & Arya 1997). It creates *a kd-tree* from the data points and allows very efficient retrieval of neighbours in a metric vector space. We make use of it by retrieving only the g nearest neighbours to the currently positively images in all m feature dimensions. Each feature dimension can make g initial suggestions. From these initial suggestion set which is usually smaller than $g \times m$ due to related similarity measures we sample our new suggestions. g varies over time and can be enlarged when there happens to be too few neighbours to present enough suggestions through the GUI. Usually we chose g=30.

This procedure also helps us in assigning importance values to the m features. These are useful when one has to decide which feature dimension is most valuable to explore further. We assign the importance values according to the frequency with which a suggested image by feature i was subsequently labeled as positive by the user.

3.5 Visualization Techniques

Visualization is one of our most essential system components. Dimension reduction techniques like the introduced *Kernel PCA* are based on the assumption that the directions in feature space that contain the majority of variance of the data values are most important because they are caused by hidden variables. In many areas this assumption totally holds true (e. g. measured 2D human height/weight data corresponds to age and sex). However, for very high dimensional image descriptors that are massively reduced to a few dimensions their inherent meaning is lost or at least hardly nameable. The visualization component's task is to convey the meaning of this abstract directions to the user and thus bridge the *semantic gap*.

Visualizing images in their feature space is a much researched topic. Our implementation offers different views on the dataset. Starcharts, parallel coordinates and grid based views are available and prove their assets and drawbacks in different scenarios according to their abilities.

The star char view mode distinguishes itself by highlighting the feature values of a selected image on the corresponding axis in red (see figure 2(a)). The user can thereby analyse which images pair according to different distances in different feature dimensions.

To avoid cluttered scenes with overlapping images we implemented an iterative procedure that first positions the images according to their respective feature value in the star chart in increasing order of their probability p^i. Then we let the more probable, hence more important images, push their overlapping counterparts away along the difference vector between the two of them. We let this procedure run until all overlaps are resolved. This ensures that the positions of the most important images are maintained as best as possible.

The second expert mode is inspired by the classical parallel coordinate view of high dimensional data (see figure 2(c)). Unlike the classical depiction the data points are not connected by lines across the axes. In fact, the images act as their own visualization of the data point. As can be seen from the screenshot, hovering the mouse cursor over one image highlights it on all the other axes. This representation has the clear advantage of separating feature dimensions. Figure 2(c) exemplifies its benefit. It shows the final context of a successful search during the case study in section 4. Even though the task is not yet introduced, one can clearly see the different distributions along the axes. It appears that feature 7 (from above, Template) is the most suited for describing the target class. Feature 10 has not been used.

The different views on the data are useful for different user groups. Expert users have the possibility of browsing each feature dimension individually. This means they can explore all $\sum_{i=1}^{m} r^i$ single dimensions separately using either the star chart view or the parallel coordinate view by simply clicking on the axes or choosing it from a drop-down menu.

Complementing the expert mode we provide an uncluttered more common presentation for the technically uneducated users. Here we show a grid layout in which the ordering of images occurs according to the probability value $p^i(f^i(I))$ (see figure 2(b)). This can be viewed as a *rank-ordered top-k returns* classifier.

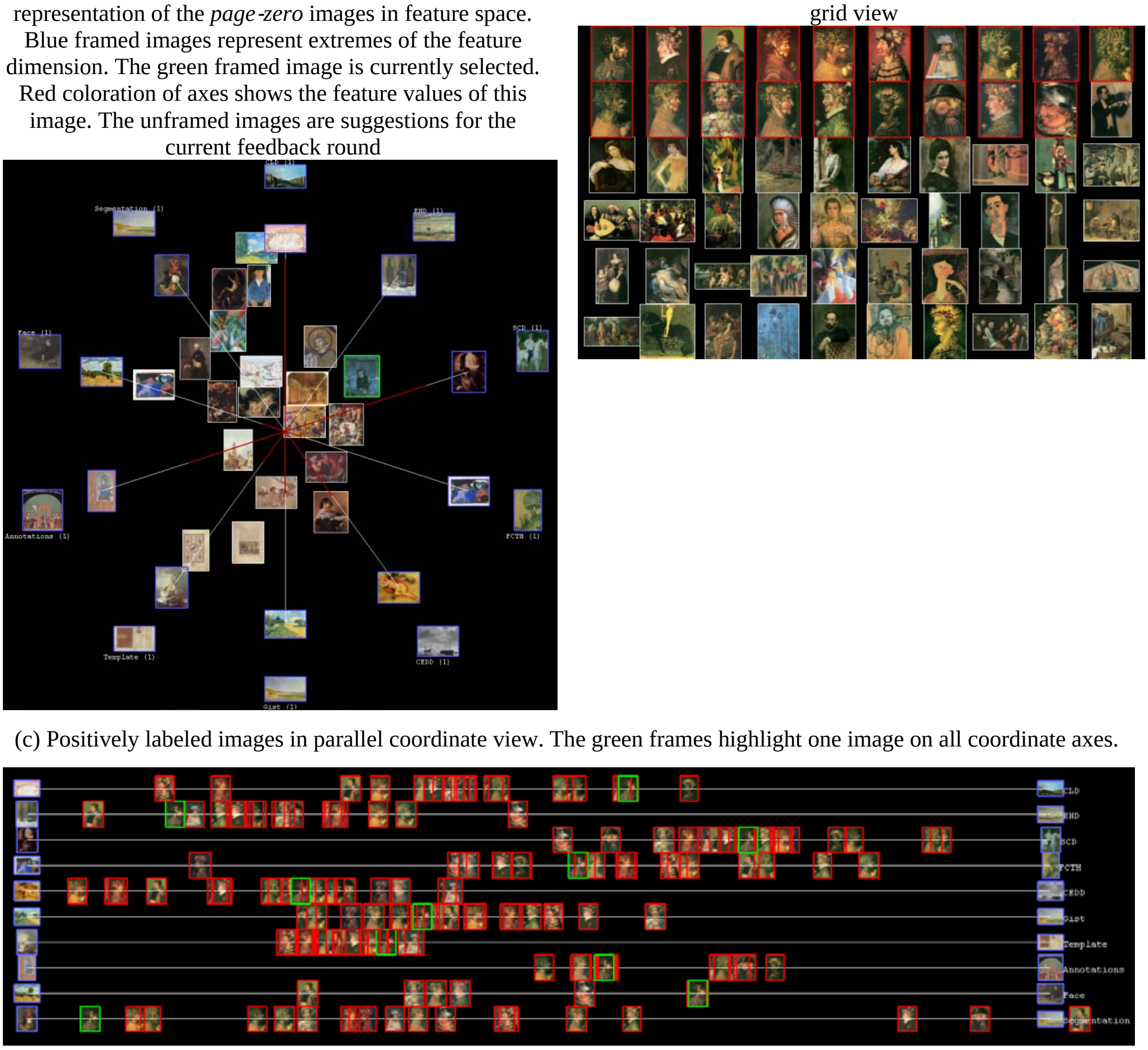
3.6 Browsing Strategies

Having explained the visual interface to our system we finish its discussion by describing how the different feature dimensions can be navigated to the user's benefit. We have m r^i dimensional feature spaces. Clearly it is not possible to show them all at once. We provide different facets to navigate them. Our initial view makes the images browsable in a space spanned by the dimensions f_0^i. Star chart and parallel coordinates view display the respective feature values along the axis. The user can either explore the combination of these most prominent directions of each feature or alternatively *jump into* one dimension i to utilize its r^i feature dimensions.

At times the user may want to take a closer look at the neighbourhood of a specific image without generating suggestions. This usually happens when the target class is roughly found. We offer the user the possibility to retrieve the nearest neighbours of a selected image in the currently enabled feature space. This allows exploration of local feature space image population and supplements our example and suggestion based browsing interface.

3.7 Analysis

The design of our system offers some additional benefits. As mentioned before it is difficult to bridge the *semantic gap* between the abstract features and the user's interpretation of the image's content. After the user is done with his search additional information can be inferred from the *MoG* model about his actual intentions. We automatically retrieve weighting factors for each feature dimension and thus infer importance values by looking at the density of feature value occurrences. If the feature values cluster along a dimension f^i in one or more groups, then f^i seems to be characteristic for the target category. If the values are uniformly distributed along feature i, it probably carries no meaning. This information could subsequently be used for fully automatic classification algorithms.



(a) Screenshot of our image browser with star chart representation of the *page-zero* images in feature space. Blue framed images represent extremes of the feature dimension. The green framed image is currently selected. Red coloration of axes shows the feature values of this image. The unframed images are suggestions for the current feedback round

(b) Images are ordered according to their relevance in our grid view

(c) Positively labeled images in parallel coordinate view. The green frames highlight one image on all coordinate axes.

Figure 2. Views on image data that our system offers

4. CASE STUDY

Evaluating complex analytic tools for visual investigation scenarios in high dimensional feature spaces is a highly debated topic. I. e. the procedure (Owen 2007) proposes is unfortunately hard to do using a small user group.

There are some plausible arguments against *target image search* scenarios. They are somehow far-fetched because of the relative rareness of the need to find one specific image that the user has in mind. Additionally the insufficient short term memory of humans complicates the task. We can only approximately remember the picture we are looking for. A database of 20.000 images, as is used in our application scenario, inevitably contains many similar images. At some point of browsing the database it becomes virtually impossible to decide which images shall be labeled positive and negative next. Due to the mentioned problems we decided to measure the system’s performance in a *target class search* scenario because here the success rate is easily measurable.

The following parts illustrate the evaluation of our search system. The evaluation's aim is to gather information about the user's effectiveness, efficiency and satisfaction concerning the quality of results provided by our system.

Dataset and Participants The evaluation was done using public domain paintings from the YORK Project DVD "10.000 Meisterwerke der Malerei" (York project 2013). Due to a number of close-ups the dataset contains nearly 20.000 unique images from various epochs. For this study we used 14 voluntary participants from the faculty of computer science. All participants had experience in using computers and common input devices. The participants were divided into two groups. The first group used our system, the second group used *Google Picasa* to accomplish the task described in the following section. Google Picasa has been chosen because of its easy to use interface and his wide capabilities for fluent browsing and handling large image sets. We also considered the usage of *imgSeek* (Cabral 2010) and *imagesorter* (Barthel 2008) but they either crashed due to memory constraints or computed too much on the fly.

Features for artwork search The quality of an image retrieval system directly depends on the features that are used to describe the images. Research from object and scene recognition respectively image and text retrieval applications resulted in a vast number of possible image descriptors and similarity measures. Since they generally originate from very different domains it is often difficult to decide how a meaningful combined similarity measure can be achieved. Concatenation is a straightforward way for doing so. However, it is generally not trivial to decide how to weigh each single feature dimension. As explained before we approach this in our work through *Relevance Feedback*.

To illustrate the system's flexibility we deliberately chose properties from very diverse feature domains. Low level features were taken from the MPEG-7 standard (Manjunath et al. 2002) or extensions based on it (Chatzichristofis & Boutalis 2011). Gist (Aude Oliva & A. B. Torralba 2001) is a mid-level feature that was shown to correlate with semantic image categories for natural images. To bridge some of the *semantic gap* contextual information were integrated by using a feature describing the distribution of faces recognized by a face detector.

User supplied semantic information can generally considered to be rare. Yet feature combinations of textual and image based information are quite promising for many different areas. Here we show how we can naturally achieve this combination of information sources with our approach. A customized text kernel using the textual annotations, which were provided with our image database, accompanies our feature set as a proof of concept.

The template feature stands for a low resolution version of the very same image it describes. Through the dimension reduction steps this feature actually reproduces the *IPC* base functions for image signals from (A. Torralba & A Oliva 2003).

The final feature we employ is based on a primitive segmentation algorithm based on the *KMeans* algorithm. It delivers a segmentation in 3 segments, is translational variant and color specific.

Table 1. Image features and similarity measures as *Kernel PCA* input for the artwork database

Image Property	Similarity Measure
Face distribution descriptor (Bradski 2010)	$exp($Weighted L_1 distance$)$
CEDD (Chatzichristofis & Boutalis 2011)	Tanimoto Coefficient
FCTH (Chatzichristofis & Boutalis 2011)	Tanimoto Coefficient
CLD (Manjunath et al. 2002)	According to MPEG-7 standard
SCD (Manjunath et al. 2002)	According to MPEG-7 standard
EHD (Manjunath et al. 2002)	According to MPEG-7 standard
Gist (Aude Oliva & A. B. Torralba 2001)	$exp(L_2$ distance$)$
Template	$exp(L_2$ distance$)$
Segmentation	Dot Product
Annotations	Normalized String co-occurrence

Task for the Participants For the evaluation we created a scenario around receiving a big amount of unsorted pictures on DVD. The dataset of 20.000 pictures represents this unsorted collection. In our scenario an uninvolved participant demands for a collection of at least eight portraits from the painter Guiseppe Arcimboldo (1526-1593). The style of this painter is unique and his portraits are composed for example with vegetables, fruits or animals (see figure 2(b)). The participants were shown six sample images of the dataset to explain the unique style of these paintings. For completing the task the participants were requested to locate pictures within the dataset and collect them to a selection.

Procedure The evaluation was divided into two parts. The first part consists of introducing the task with image examples, the setting and the program to the participants. After that, the participants had to fulfil the task. The participants were not directed or influenced during the procedure, but their interaction was observed. For the second part the participants had to complete a questionnaire with a seven point Likert scale about their experiences and the satisfaction with the programs and the fulfillment of the task.

Results and Discussion All participants completed their questionnaires and were included in the analysis. The results were twofold. On the one hand the responses showed that the task has been easier fulfilled and was less mentally challenging with our combination of techniques as through simple browsing a large dataset. This covers the rating in which our system helps to save time to find the images in comparison to Googles *Picasa*. Scrolling in large datasets with Google Picasa has been experienced as strenuous. On the other hand the analysis showed that the learning curve for using our application is steeper than with Google Picasa. The selective picking of positive and false samples in our approach has been marked as difficult to learn. This can be found in the rating of necessary comprehension for using the programs. Google Picasa needs less comprehension than our approach. We observed that the initial view (*page zero*) of our application causes some irritations. The participants were confused by images without common visible similarities to the example images. The users seemed to be unsure, which pictures had to be marked positive or negative. We could fix this problem by offering different random subsets of the centroids (see section 3.3).

Efficiency and personal satisfaction with the result has been observed higher with our approach than by using the simple browsing of Google Picasa. During the evaluation with Google Picasa, the attention of the participant dropped when scrolling and the participant overlooked some clear result images. Besides of searching for the target images, the participants had more joy of use by finding interesting similar images from other painters which had not been part of the task. Finally we asked the participants, if they would use our application for their private image collection. The users would split the use. Because of its simple usability, they would use Google Picasa for small image data sets, but they could imagine to use our approach for medium or large sets.

We noticed several characteristic outcomes of the search process. Once a target related image appeared at the beginning of the browsing process it is quite easy to mark positive examples. One tends to label many images as positive, that share a common target related attribute. As the search tends towards the end all suggested images become equally similar to the wanted image in regard to its properties first fixed. Then it becomes easier to define negative examples. At this point it is challenging to preserve the high density of the *MoG* function in the previously fixed target related feature range.

5. CONCLUSION

We presented a novel semi-automatic approach to solve the image retrieval task in presence of abstract features and unspecific target descriptions. It features an extremely simple interface which yet is powerful enough to convey meaning about the underlying distribution in feature space of the images and thus helps to bridge the *semantic gap*.

We described a modular pipeline that is easily adaptable to diverse image retrieval tasks. In this pipeline we combined well known standard techniques to solve the individual tasks. It distinguishes itself by allowing the definition of almost arbitrary similarity measures from which a metric feature space can be deduced. A multi-resolution hierarchy on the data points is employed to achieve scalability and interactivity for the browsing procedure. A new way of finding the parameters for the *MoG* model that describes the estimated distribution of the target class based on the current user interaction data was introduced. We use this model within our *Relevance Feedback* step to generate new suggestions for the user. We evaluated our work in a case study in which works of Guiseppe Arcimboldo had to be discovered among 20.000 paintings of other artists. This proved to be feasible.

Each image retrieval system faces a similar problem. What are the most potent image similarity measures that facilitate the search for a special image or an image category given a certain task and image domain? We plan to run test searches using our system for a given task with different feature subsets. Analysis of the estimated p_j^is according to their designated σ_j^is could prove beneficial. The smaller the respective variance entry for a feature dimension the better it is applicable for the investigated search task.

ACKNOWLEDGEMENT

This work was supported by the DFG Priority Program 1335: Scalable Visual Analytics.

REFERENCES

Book

Bishop, C.M., 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Chatzichristofis, S.A. & Boutalis, Y.S., 2011. *Compact Composite Descriptors for Content Based Image Retrieval: Basics, Concepts, Tools*, VDM Verlag.

Manjunath, B.S., Salembier, P. & Sikora, T., 2002. *Introduction to MPEG-7: multimedia content description interface*, John Wiley & Sons Inc.

Journal

Brivio, P., Tarini, M. & Cignoni, P., 2010. Browsing large image datasets through Voronoi diagrams. *IEEE transactions on visualization and computer graphics*, 16(6), pp.1261–70.

Campbell, I., 2000. Interactive Evaluation of the Ostensive Model Using a New Test Collection of Images with Multiple Relevance Assessments. *Information Retrieval*, 2(1), pp.89–114.

Daoudi, I., Idrissi, K. & Ouatik, S., 2008. Kernel Based Approach for High Dimensional Heterogeneous Image Features Management in CBIR Context. In *Advanced Concepts for Intelligent Vision Systems*. pp. 860–871.

Cox, I.J. et al., 2002. The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments. *Image Processing, IEEE Transactions on*, 9(1), pp.20–37.

Datta, R. et al., 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2), pp.1–60.

Flickner, M. et al., 1995. Query by Image and Video Content: The QBIC System. *Computer*, pp.23–32.

Hoffbeck, J.P. & Landgrebe, D., 1996. Covariance matrix estimation and classification with limited training data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7), pp.763–767.

Liu, Y. et al., 2007. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1), pp.262–282.

Moghaddam, B. et al., 2004. Visualization and user-modeling for browsing personal photo libraries. *International Journal of Computer Vision*, 56(1), pp.109–130.

Oliva, Aude & Torralba, A.B., 2001. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3), pp.145–175.

Owen, C.L., 2007. Evaluation of complex systems. *Design Studies*, 28(1), pp.73–101.

Rui, Y. et al., 1998. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, 8(5).

Su, Z. et al., 2003. Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. *Image Processing, IEEE Transactions on*, 12(8), pp.924–937.

Torralba, A. & Oliva, A, 2003. Statistics of natural image categories. *Network: Computation in Neural Systems*, 14(3), pp.391–412.

Zhou, X.S. & Huang, T.S., 2003. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 8(6), pp.536–544.

Conference paper or contributed volume

Barthel, K.U., 2008. Improved Image Retrieval Using Automatic Image Sorting and Semi-automatic Generation of Image Semantics. In *Proceedings of the 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*. Washington, DC, USA: IEEE Computer Society, pp. 227–230

Bartolini, I., Ciaccia, P. & Patella, M., 2007. PIBE: Manage Your Images the Way You Want! In *2007 IEEE 23rd International Conference on Data Engineering*. pp. 1519–1520.

Bradski, G., 2010. Face Detection using OpenCV.

Cabral, R.N., 2010. What is imgSeek? Available at: http://www.imgseek.net.

Camargo, J. & González, F., 2009. Visualization, Summarization and Exploration of Large Collections of Images: State Of The Art. In *Latin-American Conference On Networked and Electronic Media*.

Combs, T.T.A. & Bederson, B.B., 1999. Does zooming improve image browsing? In *Proceedings of the fourth ACM conference on Digital libraries*. pp. 130–137.

Ding, H., Liu, J. & Lu, H., 2008. Hierarchical clustering-based navigation of image search results. In *Proceeding of the 16th ACM international conference on Multimedia*. pp. 741–744.

Fogarty, J. et al., 2008. CueFlik. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*. New York, New York, USA: ACM Press, p. 29.

Hedman, A., Carr, D.A. & Nassla, H., 2005. Browsing thumbnails: a comparison of three techniques. In *Information Technology Interfaces, 2004. 26th International Conference on*. pp. 353–360.

Liu, D. et al., 2006. Efficient target search with relevance feedback for large CBIR Systems. In *Proceedings of the 2006 ACM symposium on Applied computing*. pp. 1393–1397.

Matkovic, K. et al., 2009. Large Image Collections--Comprehension and Familiarization by Interactive Visual Analysis. In *Smart Graphics: 10th International Symposium, SG 2009, Salamanca, Spain, Mai 28-30, 2009, Proceedings*. p. 15.

Meilhac, C. & Nastar, C., 2002. Relevance feedback and category search in image databases. In *Multimedia Computing and Systems, 1999. IEEE International Conference on*. pp. 512–517.

Mount, D. & Arya, S., 1997. ANN: A library for approximate nearest neighbor searching.

Qian, F. et al., 2002. Gaussian mixture model for relevance feedback in image retrieval. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*. pp. 229–232.

Thomee, B. & Lew, M.S., 2007. Relevance feedback in content-based image retrieval: promising directions. In *Proceedings of the 13th annual conference of the Advanced School for Computing and Imaging*. pp. 450–456.

Yang, J. et al., 2006. Semantic image browser: Bridging information visualization with automated intelligent image analysis. In *Visual Analytics Science And Technology, 2006 IEEE Symposium On*. pp. 191–198.

York project, 2013. Commons:10,000 paintings from Directmedia.

ACTION RECOGNITION USING COMBINED LOCAL FEATURES

Ivo Reznicek and Pavel Zemcik

Faculty of Information Technology, Brno University of Technology - Bozetechova 1/2, 602 00 Brno, Czech Republic

ABSTRACT

This paper presents a new algorithm for recognition of actions based on local space-time features. The algorithm resulted from intensive research of classification and feature extraction and it is an extension of the earlier algorithms. The most important achievement is that it is shown that carefully selected combination of space-time features leads to a greater precision of recognition on some events compared with the state-of-the-art algorithms while it is comparable on all other events. The paper describes the algorithm, its main features and improvements, demonstrates the results achieved, and draws conclusions.

KEYWORDS

Action recognition, SVM, combination of features, space-time features.

1. INTRODUCTION

Object detection, video search, and action detection have become very popular and widely used in the past decade. These tasks can be successfully used in the applications, such as video surveillance and video search and retrieval. All these tasks frequently exploit very similar processing chain consisting of local features extraction [Laptev, I. and Lindeberg, T. 2003; Dollar, P. et al., 2005; Willems, G. et al., 2008; Klaser, A. et al., 2008; Scovanner, P. et al., 2007], creation of global descriptor from the local descriptors, and classification, where the global descriptor is usually related to whole image, whole video sequence, or some portion of video sequence.

The above tasks rely on local features as they quite well describe local information about interest points in spatial domain, in case of images, and in space-time domain in case of the video sequences. Various local descriptors can be combined into global feature vector using bag-of-words [Csurka, G. et al., 2004] representation which for any image or video sequence has a nice feature of resulting vectors having the same dimensionality and thus being usable as an input for classifier. Approaches based on such representation have proven to be capable of achieving state-of-the-art results [Wang, H. et al., 2011; Wang, H. et al., 2009; Le, Q. V. et al., 2011] for action recognition tasks.

The space-time detectors were first developed and introduced by Laptev in [Laptev, I. and Lindeberg, T. 2003]; the space-time features extend the standard Harris corner detector into space-time domain. Many of the subsequently developed detectors are based on Gabor filters [Dollar, P. et al., 2005] or on the determinant of the Hessian matrix [Willems, G. et al., 2008]. Feature descriptors that are used for description of the interest point local neighbourhood range from higher order derivatives, gradient information, optical flow, and brightness information [Dollar, P. et al., 2005; Laptev, I. et al., 2008; Schuldt, C. et al., 2004] to extensions of image descriptors, such as HOG3D [Klaser, A. et al., 2008], SURF [Willems, G. et., al 2008], or 3D-SIFT [Scovanner, P. et al., 2007].

Video processing and video processing evaluation methods almost always rely on datasets. Datasets being recently and widely used for this purpose include KTH [Schuldt, C. et al., 2004], Weizmann [Gorelick, L. et al., 2005], UCF sports [Rodriguez, M. D. et al 2008], IXMAS [Weinland, D. et al., 2007], and Hollywood2 actions [Marszalek, M. et al., 2009]. The most challenging dataset is the Hollywood2 actions; it contains set of videos of a standard resolution taken from Hollywood movies with 12 real world actions annotated; the best reported results [Wang, H. et al., 2011; Wang, H. et al., 2009; Le, Q. V. et al., 2011] are currently 50%-60% (using mean average precision measure). The mean average precision metric, in this context, is defined as a mean value of all the precision recall curve surfaces for all the classes of interest.

1.1 Related Work

While in the last decade a great number of papers with various concepts for action recognition have been published, a significant part of those approaches are based on feature extraction, fixed-sized representation conversion and classifier creation. The most interesting examples of approaches are listed below.

Wang et al. evaluated in [Wang, H. et al., 2009] several combinations of feature extractors and feature descriptors, using all the important datasets available at the time. In this approach, video sequences are represented by bag-of-words and the vocabulary is created using the *k*-means algorithm. For classification purposes, the non-linear support vector machine with χ^2 kernel is used. The results are reported and measured using mean average precision.

Wang et al. [Wang, H. et al., 2011] proposed in his further work a new way of extracting the time-space interest points, called Dense trajectories. The Dense trajectories extractor is based on the assumption that search for the extrema across all three dimensions is not efficient because of the different characteristics of the space domain and the temporal domain. With this approach, the points are detected in the spatial domain and then tracked across the temporal domain. After the point trajectory has been found, the descriptor is calculated around this trajectory, while the length of all trajectories is equal. A number of descriptors were examined with this extractor. The HOG and HOF descriptors (the same as in the STIP extractor [Laptev, I. and Lindeberg, T. 2003]), trajectory descriptor, and MBH descriptor were used.

All the above feature descriptors are used separately; they are transformed into the bag-of-words [Csurka, G. et al., 2004] representation and used for training the multichannel non-linear SVM with χ^2 kernel similarly as in [Ullah, M. M. et al., 2010]. The accuracy of the algorithm is evaluated on today's datasets and it is compared with other state-of-the-art papers using the mean average precision measure.

Ullah et al. [Ullah, M. M. et al., 2010] has presented some extension of the standard bag-of-words approach where the video is segmented semantically into meaningful regions (spatially and temporally) and the bag-of-words histograms are computed separately for each region. This work also introduces a number of experiments and the results are included in our work in the comparison of results.

Le Q. V. [Le, Q. V. et al., 2011] has presented a method for the learning of features from spatio-temporal data using the independent subspace analysis. A number of experiments are included in our work in the comparison of results.

To the best of our knowledge, however, no paper has been published where all earlier known feature-like systems are combined into one solution and where the best combination is evaluated for each separate purpose.

1.2 Dataset

Marszalek et al. in [Marszalek, M. et al., 2009] proposed a dataset with twelve action classes and ten scene classes annotated, which was acquired from 69 Hollywood movies. The dataset is built from movies containing human actions and processed using script documents and subtitle files which are publicly available for those movies. The script documents contain scene captions, dialogs, and scene descriptions; however, they are usually not quite precisely synchronized with the video. The subtitles have video synchronisation so they are matched to the movie scripts and this fact can be used to improve video clip segmentation. By analysing the content of movie scripts, the twelve most frequent action classes and their video clip segments are obtained. These segments are split into test and training subsets such that the two subsets do not share segments from the same movies. Two training parts of the dataset exist; the *automatic* part, it is generated using the above-mentioned procedure while the *clean* part is manually corrected using visual information from the video. The test part is manually corrected in the same way as the clean training part of the dataset. In both cases, the correction is performed in order to eliminate "noise" from the dataset and thus to create better classifiers.

2. BASE RECOGNITION ALGORITHM

The procedure is based on the extraction of feature vectors, their transformation, and creation of classifiers; the base processing pipeline is shown in Figure 1. For the videos being processed, the local feature vectors "FV" are extracted and then transformed into the bag-of-words "BOW" representation using the visual vocabulary. The bag-of-words vectors are then combined and used as an input to the classifier creation process.

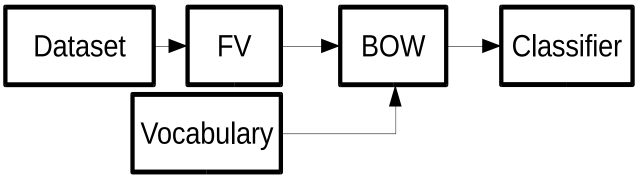


Figure 1. The base recognition pipeline: videos from dataset are converted to feature vectors, which are transformed into fixed-size representation, which in turn is used for classifier creation.

The input of the classifier engine is used for classification. The accuracy of the classification is evaluated in the processing phase using another subset of the dataset, the testing set. The outputs of the classifier are then compared with the annotations and the results are evaluated.

2.1 Feature Extraction

The purpose of the local feature extractors is to search for local extrema across the space and time domain of the input video and when the extremum is detected, the neighbourhood pixels across the space and time domain are used to obtain the feature vector describing such extrema. Alternatively, in the case of dense sampling, the extrema are not searched for and uniform sampling of space is used instead to obtain the feature vectors. In such a case, no search is required but a larger number of features need to be evaluated.

The following feature extractors were presented for action recognition: STIP, Cuboids, HesSTIP and Dense Trajectories their fundamentals will be presented below.

In the STIP extractor, the key points are searched for using the extended Harris corner detector [Harris, C. and Stephens, M. 1988]. Subsequently, for each of the detected points, the space-time patch is extracted and the HOGHOF [Laptev, I. and Lindeberg, T. 2003] descriptor is computed. The descriptor consists of the histogram of gradient descriptor and the histogram of optical flow descriptor which are simply concatenated. HOG captures the static appearance information while HOF captures the local motion information.

The Cuboids extractor is based on the 2D Gaussian smoothing kernel, which is applied spatially, and the quadrature pair of 1D Gabor filters, which is applied temporally. The non-maxima suppression and thresholding are performed and as a result of this process, the key point locations are detected. The cuboids descriptor is simply computed by concatenating the gradients obtained for each pixel in each dimension of the processed patch. Another type of cuboids extractor is also known, where the key point search procedure is replaced by the Harris corner detector.

In the HesSTIP extractor, the key points are detected using the space-time extension of the Hessian saliency measure (which is usually used for blob detection in images). The detector measures the saliency using the determinant of the 3D Hessian matrix. The descriptor vector is obtained as follows. The space-time patch is divided into cells. For each cell, the vector of weighted sums of uniformly sampled responses of the Haar-wavelets along the three axes is computed. Vectors from all cells are then concatenated.

The dense trajectories extractor is depicted in Section 1.1; to describe the detected trajectories the HOG, HOF, MBx and MBy descriptors were used. Generally, every feature extractor generates a set of feature vectors, all of which have the same dimension from a single video file.

2.2 Visual Vocabulary and Bag-of-Words

The visual vocabulary is created as a model for representation of the low-level feature space and it is formed by a set P of representatives P_i (points) in n-dimensional space. The size of the vocabulary has to be adjusted to a suitable value so that the representation of the space is compact and accurate enough at the same time. If the size is too large, nearly all low-level features become representatives of the visual vocabulary. If the size were too small, very large clusters would exist and the discriminative power of the whole solution might be adversely affected.

K-means square-error partitioning method [Duda, R. O. et al., 2000] can be used for such purpose. This algorithm iteratively processes data such that it assigns feature points to their closest cluster centres and recalculates the cluster centres. The *k*-means algorithm converges only to local optima of the squared distortion and does not determine the *k* parameter. It can be parametrized through specifying the number of iterations and the number of output clusters.

The bag-of-words [Csurka, G. et al., 2004] can represent the video sequence or its part using one feature vector with the same dimension, irrespective of the number of local space-time vectors or the video shot length; the bag-of-words representation can be (in its simple form) constructed in the following way. The input of this process is the set *S* of local feature vectors $s \in S$ and a vocabulary while the output is a histogram of the occurrences of matched input vectors. For each input vector, exactly one bin in the output histogram is incremented. This simple form of assignment is sometimes called the *hard assignment* and also has some disadvantages. The main disadvantage is that only slightly different input local feature vectors may be accumulated into totally different output histogram bins (the nearest code words are different); this may cause total dissimilarity of two similar input vectors.

The above issue is addressed in the *soft assignment* approach; the soft assignment is performed as follows. A small group of the clusters very close to the vector being processed is retrieved instead of a single cluster; all the clusters from such a group are assigned a weight corresponding to their closeness to the vector; finally each of the corresponding output histogram bins is incremented by the weight of the appropriate clusters.

The most frequently used method of weight computing is through exponential function of the distance to the cluster centre $w_i(a) = \exp(-\frac{(d(a, p_i))^2}{2\sigma^2})$, where *d* is Euclidean distance from the cluster centre to the vector while σ is a parameter and controls the width of the function. This function needs to be evaluated for each of the clusters in the group. Finally, *soft assignment* parameters correspond to the number of the very close vectors to be considered and the σ which controls the shape of soft-weighting function.

2.3 Classifier

The classifier can be described as a blackbox unit which has two modes of operation: the training phase, where the model for certain input labelled data is created, and the classification phase, where the classifier is able to decide how the tested data should be labelled. Generally, inside this box, many algorithms can be used (SVM [Zhang et al. 2007], neural networks [Kriesel D. 2007], Bayesian classifier [Friedman N. et al., 1997], etc.), the common property is that classifier creation is dependent on the set of parameters and its quality is based on these parameters. The input of the classifier is typically an input vector typical of an object, the output is a vector of class likelihood.

For action recognition and image-content recognition the most popular classifier type is the SVM (Support Vector Machine) with various kernel functions (for example, linear kernel, rbf kernel or χ^2 kernel).

3. OPTIMAL COMBINATION OF FEATURES

The above presented algorithm can be extended through a combination of different features forming the feature vector. It will be shown that a proper combination of the features can lead to an improvement of the performance beyond the state of the art when selected individually for some of the event classes.

The algorithm is depicted in Figure 2; a larger number of feature extractors are used for processing. For each feature extractor a larger number of visual vocabularies are created and used for the creation of all possible bag-of-words representations. Everything is concatenated into a multiple channels feature vector. The *selection* unit combines several input channels and it is passed to the multikernel SVM classifier creation process. This operation is repeated several times.

The classifiers created through the above mentioned processing need to be explored and the best one will be further used. The classifiers are evaluated against a chosen metric. The selected one is evaluated using the testing dataset and is measured using, for example, the average precision metric.

The whole processing needs three types of dataset: the training one, which is used for the creation of classifiers; the validation one, which is helpful in best solution selection, and the testing one, which is used for measuring the whole-system accuracy.

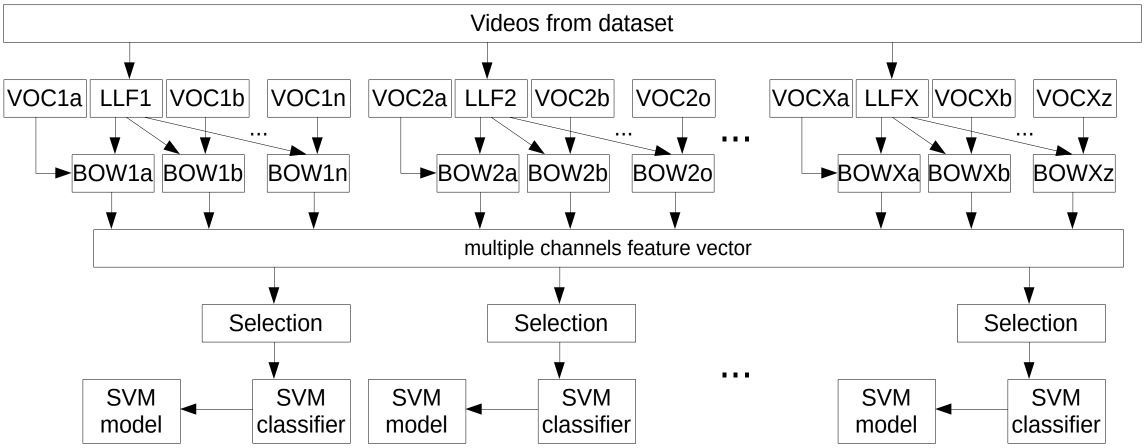


Figure 2. The algorithm block diagram; The *LLFx* boxes depict the feature extractors, the *VOCx* represent the vocabularies constructed from related feature extractors, the *BOWx* boxes depict the bag-of-words units, the multiple channels feature vector is constructed by concatenation of all vectors, but the positions of all subparts need to be kept.

The algorithm uses the non-linear support vector machine [Zhang et al., 2007] with multichannel Gaussian kernel [Zhang et al., 2007; Wang et al., 2011]. The kernel shall be defined as:

$$K(A,B) = exp(-\sum_{c\in C}\frac{1}{A_c}D_c(A,B)) \qquad (1)$$

where A_c is the scaling parameter, which is determined as a mean value of mutual distances D_c between all the training samples for the channel c, D_c(A, B) is the χ^2 distance between two bag-of-words, and A and B are the input vectors of the form:

$$A_i = (\underbrace{a_1 \dots a_{n1}}_{channel\ \langle 1,n_1\rangle},\ \underbrace{a_{n_1+1} \dots a_{n2}}_{channel\ \langle n_1+1,n_2\rangle},\dots,\ \underbrace{a_{n_i-1} \dots a_{n_i}}_{channel\ \langle n_i-1,n_i\rangle}) \qquad (2)$$

The set of channels C can be defined as:

$$C = \{\langle 1,n_1\rangle, \langle n_1+1,n_2\rangle,\dots,\langle n_i-1,n_i\rangle\} \qquad (3)$$

The bag-of-words distance D_c(A, B) may be obtained as:

$$D_c(A,B) = \frac{1}{2}\sum_{n\in c}\frac{(a_n-b_n)^2}{a_n+b_n} \qquad (4)$$

The best ratio of input channels $\{c_k, c_l, \dots, c_z\}\in C$ for a given training set is estimated using the coordinate descent method. The set of input channels needs to be specified outside of the training process. Besides this SVM, the building procedure requires the number of input parameters that affect the classifier accuracy; these parameters are automatically evaluated using the cross-validation approach [Hsu, C. W. et al., 2003]. The classifier creation process may be apprehended in the whole procedure as a black-box unit where only the set of input channels is specified, and for a given input the best performing classifier is created automatically.

The number of channels used may induce a very large space which needs to be searched. The number of possible combinations of this space can be computed as a sum of the sequence which may be defined as follows: $count=\binom{|C|}{1}+\binom{|C|}{2}+\ldots+\binom{|C|}{|C|}=2^{|C|}-1$, where |C| represents the number of channels.

Currently, we are able to achieve a good performance by an ad-hoc (manual or blind) specification of the input channel combination (it will be further shown in Chapter 4), the algorithm for automatic channels selection is now under development and was not used in this paper.

4. EXPERIMENTAL RESULTS

The main achievement of the presented work is the confirmation of the hypothesis that a suitable combination of different features for action recognition does improve the accuracy of the whole processing chain; this idea has been explored and evaluated using one of the most challenging datasets [Marszalek, M. et al., 2009] available today. The following twelve action classes were evaluated, namely: *answering the phone, driving car, eating, fighting, getting out of the car, hand shaking, hugging, kissing, running, sitting down, sitting up and standing up*.

In our experiments, the clean part of the training dataset was used for the classifier training procedure (823 samples). The automatic part of the training dataset was re-annotated and used for validation purposes (810 samples). The original testing dataset (884 samples) was used for measuring the solution using average precision for every class, the over-all classes mean average precision is reported as well.

The following feature extractors were used in the experiment, the associated list of descriptors is given in parentheses, every combination extractor and descriptor was used as a standalone features set plus all the dense trajectories descriptors were concatenated and used as well:

- Dense Trajectories (Trajectory, HOG, HOF, MBH),
- HesSTIP (ESURF)
- Cuboids (Cuboids)
- STIP (HOGHOF)

Some vocabularies were created using the *k*-means algorithm with 12 iterations; this number represents a trade-off between the processing duration and the output vocabulary achievement. To create these vocabularies, ca. 2 million local low-level features were used and were extracted from all training videos of the dataset. Vocabulary sizes were set to 1000, 6000 and 8000, all possible combinations, feature extractors and. vocabularies sizes were used.

Table 1. Results of average precision of the four best performing experiments on the validation dataset.

Action	1	2	3	4	BEST	Selected classifier
answering the phone	0.379	0.299	0.322	0.423	0.423	4
driving car	0.571	0.62	0.554	0.578	0.62	2
Eating	0.327	0.355	0.295	0.37	0.37	4
getting out of the car	0.377	0.237	0.304	0.273	0.377	1
Running	0.629	0.683	0.736	0.702	0.736	3
sitting down	0.487	0.559	0.511	0.574	0.574	4
sitting up	0.286	0.204	0.385	0.331	0.385	3
standing up	0.486	0.55	0.394	0.527	0.55	2
Fighting	0.625	0.594	0.55	0.561	0.625	1
hand shaking	0.493	0.541	0.439	0.594	0.594	4
Hugging	0.355	0.339	0.417	0.369	0.417	3
Kissing	0.531	0.630	0.594	0.609	0.630	2
Mean average precision	0.462	0.468	0.458	0.478	0.484	

The soft-assignment approach was used for the bag-of-words representation with the following parameters: $\sigma = 1$, the number of searched closest vectors was 16; these values were evaluated in [Reznicek, I. and Zemcik, P. 2011] and are suitable for bag-of-words creation from space-time low-level features.

Bag-of-words representations generated from all the possible combinations feature extractors and vocabularies become the input channels to the SVM creation process. SVMs were created as described in chapter 3. The dataset used induces the multiclass classification. The one-against-all approach was used and no relation between classes has been considered.

The number of input channels in our experiment is 24 and the total number of possibilities is then: $\binom{24}{1}+\binom{24}{2}+\ldots+\binom{24}{24}\simeq 16{,}7.10^{6}$.

We have searched about 0.1% of the desired space in a semi-automatic way and the four most interesting results (combinations) for the validation part of the dataset are presented in Table 1. The average precision is reported for each class and the mean average precision is reported for the whole validation dataset.

Table 2 represents the results *for our class-based best input channel combinations* (as shown in Table 1) achieved using the test part of the Hollywood2 dataset in the column *OUR* and they are compared to the three other authors' papers [Wang et al., 2011; Le et al., 2011; Ullah et al., 2010] which represent today's state-of-the-art for Hollywood2 dataset.

Our combination-based solution outperformed all other state-of-the-art methods in four classes, namely *driving car, running, sitting down, standing up*; in the other cases, the solution does not reach the state-of-the-art performance but it is still comparable.

As the performance of classifiers based on the combination of features is known only after the validation phase, the best solution based on the combination of features or another approach can be chosen individually for each type of action; therefore, improvement in four out of twelve actions leads to the best-known classification mechanism, also shown in Table 2.

Table 2. Results of average precision of the selected classifiers, compared with the state-of-the-art.

Action	**OUR**	[Wang et al 2011]	[Le et al. 2011]	[Ullah et al 2010]	**BEST KNOWN**
answering the phone	0.259	**0.326**	0.299	0.248	**0.326**
driving car	**0.91**	0.880	0.852	0.881	**0.91**
eating	0.491	**0.652**	0.597	0.614	**0.491**
getting out of the car	0.408	**0.527**	0.454	0.474	**0.527**
running	**0.834**	0.821	0.757	0.743	**0.834**
sitting down	**0.655**	0.625	0.594	0.613	**0.655**
sitting up	0.206	0.200	**0.257**	0.255	**0.257**
standing up	**0.663**	0.652	0.647	0.604	**0.663**
fighting	0.723	**0.814**	0.772	0.765	**0.814**
hand shaking	0.286	0.296	0.203	**0.384**	**0.384**
hugging	0.364	**0.542**	0.382	0.446	**0.542**
kissing	0.601	**0.658**	0.579	0.615	**0.658**
Mean average precision	0.533	**0.583**	0.533	0.553	**0.589**

5. CONCLUSIONS AND FUTURE WORK

The present work focuses on the recognition of gestures and actions in video sequences. The purpose of the work was to demonstrate that recognition of actions can be improved through combinations of different space-time features.

While a suitable general method for selecting of features to be combined is not yet known, our experiments demonstrate the feasibility of the idea because some of the feature combinations outperform the current state-of-the-art for four of twelve actions classes and it nearly matches the state-of-the-art for most of the remaining classes.

The implementation of the action recognition system was performed using the Hollywood2 dataset with a measurable improvement over the state of the art. The procedure of creating of a classifier based on a combination of features was also shown.

Future work includes research into algorithms for automatic selection of features, research into methods of feature fusion, and also general action recognition methods.

ACKNOWLEDGEMENTS

This work has been made possible thanks to the funding through the CRAFTERS ConstRaint and Application driven Framework for Tailoring Embedded Real-time Systems, Artemis JU, project 7H12006 and V3C - Visual Computing Competence Center, Technology Agency of the Czech Republic, TAČR, TE01020415V3C.

REFERENCES

Csurka, G. et al, 2004. Visual categorization with bags of keypoints. *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

Dalal, N. et al, 2006. Human detection using oriented histograms of flow and appearance. *In ECCV*, pages 428–441.

Dollar, P. et al, 2005. Behavior recognition via sparse spatio-temporal features. *In VS-PETS*, pages 65–72.

Duda, R. O. et al, 2000. *Pattern classification.* Wiley, New York; Chichester.

Friedman, N. et al, 1997. Bayesian network classifiers, *Machine Learning*, 29:2/3.

Gorelick, L. et al, 2005. Actions as space-time shapes. *In ICCV*, pages 1395–1402.

Harris, C. and Stephens, M., 1988. A combined corner and edge detector. *In Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Hsu, C. W. et al, 2003. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.

Klaser, A. et al, 2008. A spatio-temporal descriptor based on 3d-gradients. *In BMVC.*

Kriesel, D., 2007. *A Brief Introduction to Neural Networks*, available at http://www.dkriesel.com

Laptev, I. and Lindeberg, T., 2003. Space-time interest points. *In ICCV*, pages 432–439. IEEE Computer Society.

Laptev, I. et al, 2008. Learning realistic human actions from movies. *In CVPR*. IEEE Computer Society.

Le, Q. V. et al, 2011. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. *In CVPR*, pages 3361–3368. IEEE.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *In IJCV*, 60:91–110.

Marszalek, M. et al, 2009. Actions in context. *In CVPR*, pages 2929–2936. IEEE.

Reznicek, I. and Zemcik, P., 2011. On-line human action detection using space-time interest points. *In Zbornik prispevkov prezentovanch na konferencii ITAT*, september 2011, pages 39–45. Faculty of Math-ematics and Physics.

Rodriguez, M. D. et al, 2008. Action mach a spatio-temporal maximum average correlation height filter for action recognition. *In CVPR*. IEEE Computer Society.

Schuldt, C. et al, 2004. Recognizing human actions: A local svm approach. *In ICPR* (3), pages 32–36.

Scovanner, P. et al, 2007. A 3-dimensional sift descriptor and its application to action recognition. In ACM Multimedia, pages 357–360. ACM.

Ullah, M. M. et al, 2010. Improving bag-of-features action recognition with non-local cues. *In BMVC*, pages 1–11.

Wang, H. et al, 2011. Action recognition by dense trajectories. *In CVPR*, pages 3169–3176. IEEE.

Wang, H. et al, 2009. Evaluation of local spatio-temporal features for action recognition. *In BMVC.*

Weinland, D. et al, 2007. Action recognition from arbitrary views using 3d exemplars. In ICCV, pages 1–7. IEEE

Willems, G. et al, 2008. An efficient dense and scale-invariant spatio-temporal interest point detector. *In ECCV* (2), volume 5303 of Lecture Notes in Computer Science, pages 650–663. Springer.

Zhang, J. et al, 2007. Local features and kernels for classification of texture and object categories: a comprehensive study. *In International Journal of Computer Vision*, 73:2007.

TRAINABLE METHOD FOR PREDICTING CHARACTERISTICS OF LAND SURFACE OBJECTS

Alexander Murynin[1], Konstantin Gorokhovskiy[2] and Vladimir Ignatiev[3]
[1]Dorodnicyn Computing Centre of RAS, Vavilov st. 40, 119333 Moscow, Russia
[2]"AEROCOSMOS" Institute for Scientific Research of Aerospace Monitoring, 4, Gorokhovsky lane, 105064, Moscow, Russia, http://www.aerocosmos.info
[3]Moscow Institute of Physics and Technology (State University), 141700, Russia, Moscow Region, Dolgoprudny, Institutskiy Pereulok, 9

ABSTRACT

A new method for predicting characteristics of land surface objects has been proposed. The method is based on finding annual periodical patterns and comparison with a pattern obtained for year of observation. An example of the method application is considered. In the example authors propose, train and test a model for forecasting of crop yields based on multi-year remote observations of vegetation conditions in several regions of Russian Federation.

KEYWORDS

Image mining, remote sensing, forecasting, nonlinear regression.

1. INTRODUCTION

Currently, remote sensing techniques are the most promising ways of monitoring a condition of various objects and land areas on the earth's surface. One of the benefits of such techniques is a low cost. Many satellite imagery databases are accessible free of charge. Another benefit is immediacy of observations for the vast areas of the earth surface. A special place belongs to the satellite monitoring methods that allow getting information about a condition of objects using electromagnetic waves of a wide spectral range. The data availability and the possibility to process images with wide range of spatial resolution from few kilometers up to decimeters in long-term time series is important for many practical applications.

There is a variety of methods for objects' hidden condition prediction based on observable characteristics, obtained from a remote sensing data. They are used for diagnosis and forecasting of objects and land areas conditions on the earth's surface.

These methods are used for the prediction of expected crop yield [Murynin A. et al 2013], the evaluation of ecological environment [Bondur V. 2011], the study and prediction of natural disasters [Bondur V. et al 2010]. Most of the techniques proposed in these studies allow solving a problem only for a certain type of phenomenon. Implementation complexity and large amounts of input data makes it preferable to create a generalized method and a software tool that has certain universality, and which is applicable for a variety of forecasting problems.

Therefore, there is a need to develop an integrated approach to forecasting objects conditions on the underlying land surface using satellite images. A method for predicting this condition is proposed in the paper. The method uses an integrated approach combining a historical remote sensing data and known information about studied objects for the previous years. The availability of such reverence historical data makes it possible to train a model and verify its accuracy.

2. GENERAL METHOD OF OBJECT CONDITION FORECASTING

2.1 Basic Concepts

In order to predict the condition of an object of arbitrary nature it is required to consider not only observations that characterize its condition at the given moment, but also a history of observations prior to that moment. This approach allows identifying long-term dependencies and changes in the condition of an object. Here, the term "object" is used to describe an area on the Earth's surface (and on the captured digital image of this surface) with uniform characteristics. It can be of any shape and size (D2, Fig.1).

Let us denote the observations at the current moment as "seasonal observations" (Seasonal Observations, Fig.1). Seasonal observations include a measurement of object's parameters for the time period of interest prior to the given moment in time. Number and types of parameters calculated from a remote sensing data may vary depending on conditions such as the spatial resolution, image types, amount of data available for processing or the physical nature of an object of study.

While seasonal observations can be used to extract the information about the object of interest at the current moment, the long-term observations can help forming and verifying the predictive function. Long-term observations contain two types of information about the object. The first type is the information directly extracted from a remote sensing data, which reflect a change in observable conditions. The composition of this type of long-term observations can be as small as seasonal observations only (intra-annual), or can be expanded to all available observations for many years with multiple observations per each year. The second type of long-term historical information related to the ground-based observations. It usually comes from the sources unrelated to a remote sensing. This information used for training and verification of a model. For example such information can be taken from official government statistics (such as crop yields for various regions and years).

Cyclical seasonal observations and ability to take measurements within a time period using remote sensing data allows achieving a large number of repetitions when training a prediction model. Seasonal data have no statistical homogeneity, that is, (it) change(s) from season to season, whereas, long-term observations tend to be repeated from year to year. The fact that the seasonal observations are a part of long-term observations makes it possible to ensure uniformity of statistical data in the training model.

Thus, the requirements for the size of the training set and the uniformity of statistical data for training are taken into account. Moving further, it is required to examine various types of predictive function, as well as a possibility to take into account any log-term trends. Among other things, it is important define a way to assess the accuracy of the prediction. These tree aspects (type of model, long-term trends and assessments of the accuracy) define main concepts of the proposed method.

2.2 Workflow of Training and Predicting

The proposed method can be described as follows. The condition of an object at a given location should be fairly reliably predicted by a function whose parameters are averaged (by object's area) values of the remote sensing data during the period of each seasonal observation. The better the historical track record of the long-term observations, the more accurate the prediction of an object's condition can be made.

The schematic description of the method is shown in Fig. 1. According to the figure the forecasting workflow can be divided in three main parts. In the first stage of the Data Preparation (D-block, Fig.1) regions of interest are defined before storing the data in databases of observations. In the second stage observable informative features, which form an observable feature description of the object, are extracted from remote sensing images (D2, Fig.1).

Training Workflow (T-block, Fig.1) includes stages of creating and training a prediction model using linear or nonlinear regression on remote sensing data. The form of predictive function varies from linear, whose parameters are taken from the observable feature description of the objects, to nonlinear with factor adjustment for regions of Earth's surface and long-term temporal trend.

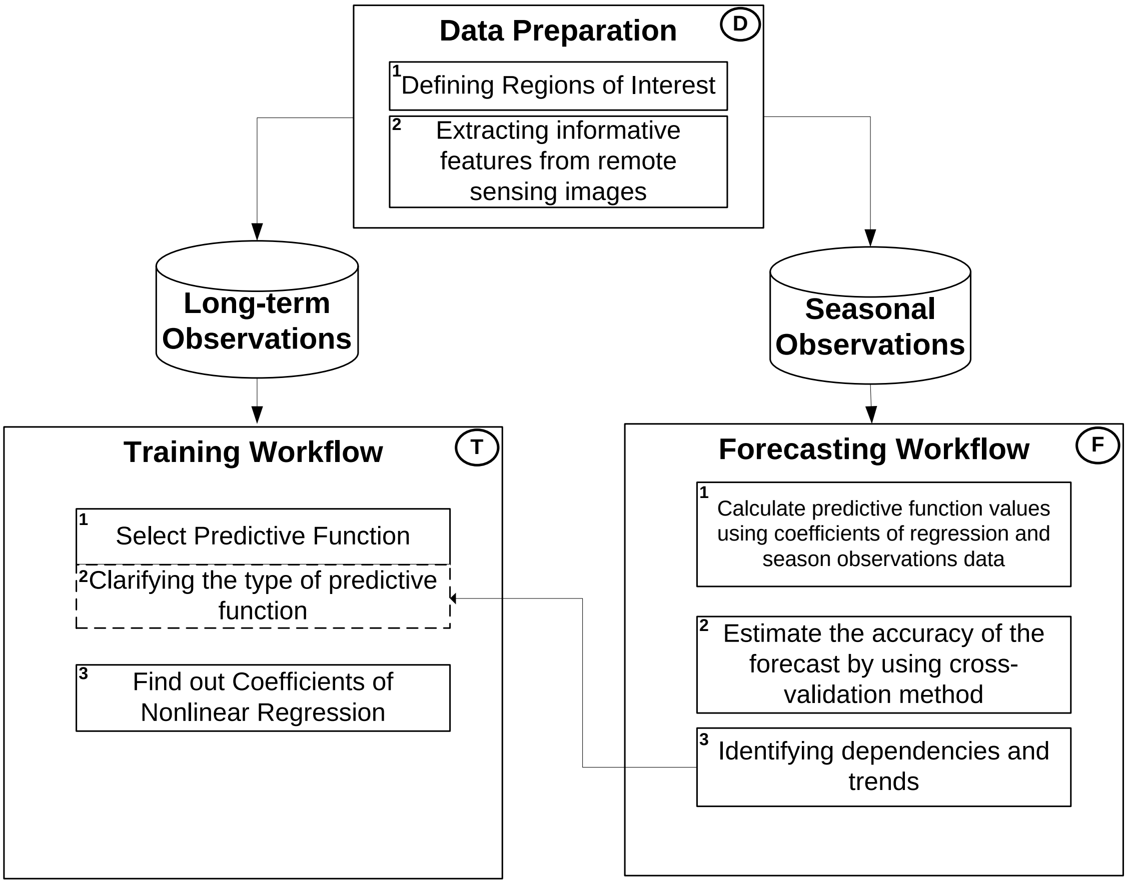

Data Preparation
D
1 Defining Regions of Interest
2 Extracting informative features from remote sensing images
Long-term Observations
Seasonal Observations
Training Workflow
T
1 Select Predictive Function
2 Clarifying the type of predictive function
3 Find out Coefficients of Nonlinear Regression
Forecasting Workflow
F
1 Calculate predictive function values using coefficients of regression and season observations data
2 Estimate the accuracy of the forecast by using cross-validation method
3 Identifying dependencies and trends

Figure 1. Workflow in forecasting the condition of objects of the underlying surface.

In the Forecasting Workflow (F, Fig.1) the condition of objects on the underlying surface is predicted using seasonal observations, which were not involved in the training of the model. Seasonal observations describe the changes of object's conditions proceeding the moment of forecasting. Thus, long-term and seasonal observations are combined. The useful information is extracted from long-term observations and accounted during training of the model. In turn, the informative features of seasonal observations influence on the result as model parameters.

Prediction results are verified using the cross-validation (F2, Fig.1). Method allows detection of a long-term trend in changes of object conditions. This, in turn, allows adjusting of the original prediction formula to take into account this trend (F3, Fig.1) for more precise prediction.

The decency of regular errors from the year of forecast is used to make a hypothesis about existence of a long-term trend (F3, Fig.1). Then, the trend hypothesis needs to be validated. In order to do that the original prediction formula is modified to take into account the assumed trend (T2, Fig.1). The resulting new formula is trained on the same data as the original one. When two formulas (with and without the trend) are obtained they are compared using confidence intervals. If the improvement form the new formula with trend is statistically significant one can make a conclusion that the trend hypothesis is valid.

2.3 Example in Yields Forecasting

The described above method can be successfully applied for the task of yields forecasting based on multiyear observations of vegetation indexes. The relationship of vegetation indices with productivity of plants is well studied [Phillips L. et al 2008].

Crop yield used in the example is the amount of agricultural yield of a certain crop harvested from the unit area.

The proposed two models can be described as follows. Crop yield of a particular culture at a given territory should be fairly reliably predicted by function whose parameters are averaged (by this region) values of vegetation indices during growth and ripening period of the crop. The better the historical track record of the indices is known, the better the forecast of crop yields can be made.

2.3.1 Basic Linear Approach in Crop Yields Forecasting

Let us assume that within the studied region the soil and climate characteristics have a small variation. The simplified model can be described in linear form as:

$$y_r = \alpha_0 + \sum_{t=1}^{T} \alpha_t \cdot \langle v_t \rangle_r \tag{1}$$

where

r -index pointing to an area (region) of the Russian Federation,

y_r -crop yield estimate for a given area (r),

$\langle v_t \rangle_r$ -average value of the vegetation condition index for a given region of the Russian Federation,

α_t - adjustable parameters of the model for individual time intervals of the vegetation period (or calendar year).

This model needs to be extended in order to be used in practical applications.

2.3.2 Model with Factor Adjustment for Regions

In the case when the amount of statistical data available for the adjustment of the individual models for each of the region is not sufficient it is required to reduce the number of adjustable parameters. Thus, in particular, one can assume that the main contributions to the difference in crop yields are made by the following factors:

- fertility of soils in a region,
- climatic differences between regions,
- amount of solar radiation, depending on the latitude of a region.

At the same time to build the model, we deliberately ignore the temporary displacement of growing season for various regions for the western part of the Russian Federation taken for this study. Using the above assumptions, the following formula can be suggested:

$$y_r = C_r \cdot \left(\alpha_0 + \sum_{t=1}^{T} \alpha_t \cdot \langle v_t \rangle_r \right) \tag{2}$$

where

r -index pointing to a region of the Russian Federation,

y_r -estimate the yield for a given region (r),

C_{rk} -coefficient of performance of the region for specific crop type,

$\langle v_t \rangle_r$ -average value of the vegetation condition index for a given region of the Russian Federation,

α_t - adjustable parameters of the model for individual time intervals of the vegetation period (or calendar year).

During the study of this model it was found that there is a noticeable correlation between the year of prediction and relative deviation of our forecast from the actual data. The relative deviations (with sign) are shown in Figure 2. The consistency of deviations brings to life the hypothesis that there is a long term trend of improvements in crop yields which does not depend on vegetation condition indexes.

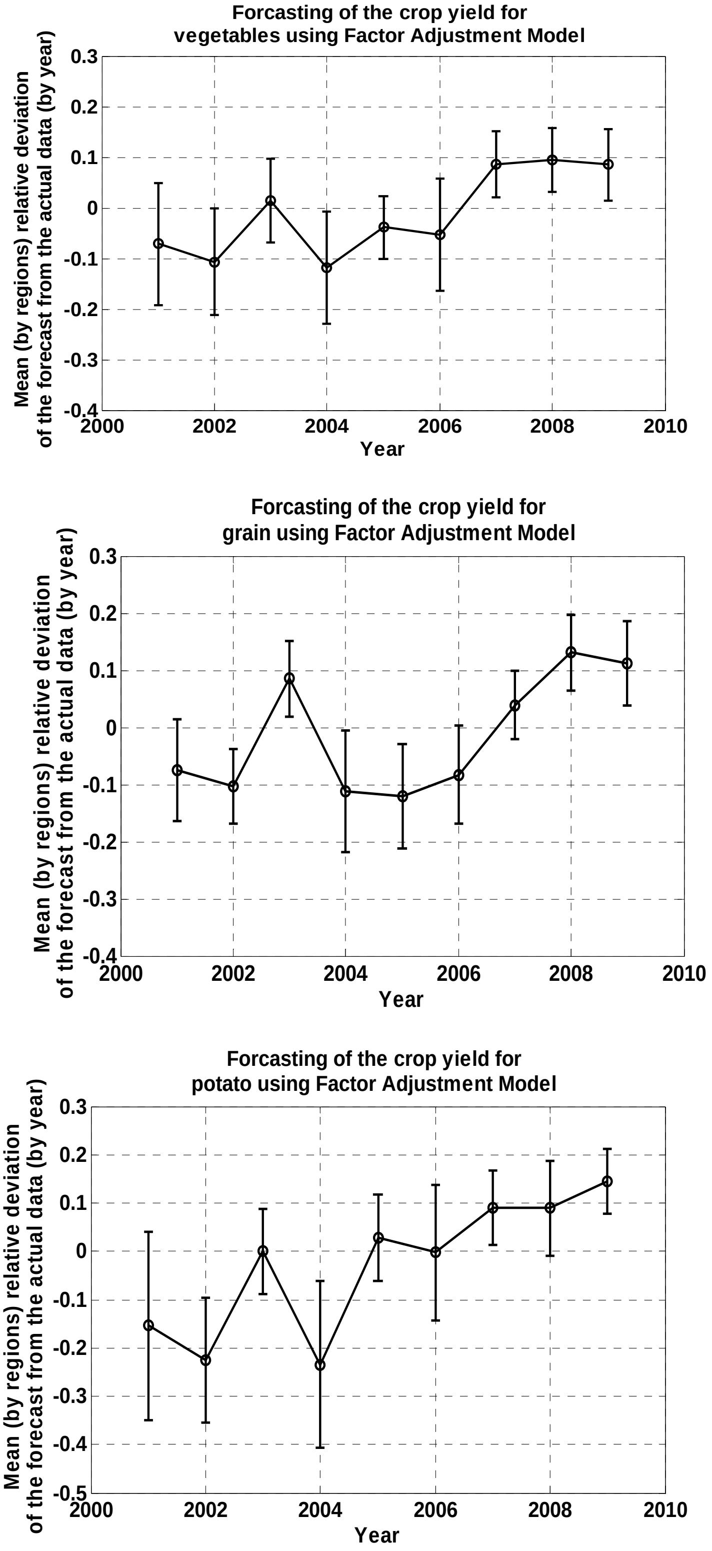


Figure 2. The appearance of the long term trend in the results from the Factor Adjustment Model for vegetables, grain and potato is shown in this figure. Negative values of the deviation indicate that the actual results were worse than predicted by model, positive values mean that the actual results are better than the model. Standard deviations across all studied regions are also shown as error bars for each year and the crop type.

2.3.3 Model with Factor Adjustment for Regions and Temporal Trend

In the past few decades, there has been a stable growth of crop yields per unit of cultivated area [Fischer R. et al 2009] all over the globe. This is due to several factors. First of all, it is worth noting the progress in genetic engineering for crops improvement. Improved seeds are more resistant to drought, temperature changes and parasites. Another factor is the more efficient use of fertilizers. Progress in the field of agricultural technology has allowed to harvest with fewer losses. Improved methods of chemical treatment resulted in better control of the pest populations.

Such improvements are referred as trend in crop yield improvements. It is required to take into account the trend in the crop yields because it is likely that similar trend will continue in the next few years.

Making the assumption that the yield changes are linearly dependent on time to the present historic interval one can improve the formula from the previous model for predicting the long-term increase in yields. Therefore the average yield for the current year can be expressed from the yield previous year by the following equation:

$$\frac{\langle y_{current}\rangle - \langle y_{start}\rangle}{\langle y_{start}\rangle} = \beta \cdot (Y_{current} - Y_{start}) \tag{3}$$

where

$\langle y_{current}\rangle$ -average crop yield for the current year $Y_{current}$,

$\langle y_{start}\rangle$ -average crop yield in year of the beginning of observations Y_{start},

β -relative annual increase in productivity due to long-term trend.

Let us express $\langle y_{current}\rangle$ in terms of the other variables:

$$\langle y_{current}\rangle = [1 + \beta \cdot (Y_{current} - Y_{start})] \cdot \langle y_{start}\rangle \tag{4}$$

We get the following formula for the refined model of crop yields:

$$y_r = [1 + \beta \cdot (Y - Y_{start})] \cdot C_{rk} \cdot \left(\alpha_0 + \sum_{t=1}^{T} \alpha_t \cdot \langle v_t\rangle_r\right) \tag{5}$$

where

r -index pointing to a region of the Russian Federation,

y_r -estimate the yield for a given region (r),

Y -current year for which the crop yields are evaluation,

Y_{start} -the year of the beginning of observations,

β -relative annual increase in productivity due to long-term trend,

C_r -coefficient of performance of the region for specific crop type,

$\langle v_t\rangle_r$ -average value of the vegetation condition index for a given region of the Russian Federation,

α_t - adjustable parameters of the model for individual time intervals of the vegetation period (or calendar year).

2.3.4 Results of Example Application

Remote sensing data for 14 regions of Russian Federation over span of 10 years (from 2000 to 2009) were used for training and validation of the models. Formation of the dataset for training and testing models has been done using the 16-day composite images captured by TERRA satellite with a spatial resolution of 500 meters. The areas for forecasting belong to the central federal district of the Russian Federation and consist oa the following districts: Vladimir, Voronezh, Ivanovo, Kursk, Lipetsk, Moscow, Nizhny Novgorod, Orel, Penza, Rostov, Ryazan, Tambov and Tula, and the Republic of Mordovia.

Satellite images were taken form the web site of U.S. Geological Survey at the following address: ftp://e4ftl01.cr.usgs.gov/MODIS_Composites/

The accuracy of the models was assessed using K-fold cross-validation method. The whole set of input data has been partitioned several times into two subsets: the training subset and the testing subset. Each time the testing subset was different. In total 10 unique testing subsets were used so that the data for each year available were used as a testing subset at least once.

The resultant accuracies of prediction for three groups of cultures and two forecasting models are shown in Table 1. Forecasting errors of crop yields is evaluated like a standard deviation of forecasting values from numbers of official statistics.

Table 1**.** Standard deviation of the forecasts crop yields for different models and cultures using cross-validation method for period 2000-2009.

	Grain	Vegetables	Potato
Factor adjustment	34.8%	16.1%	19.9%
Factor adjustment with trend	19.1%	10.5%	18.7%

The best result is generated by the model with factor adjustment and long-term trend. It shows considerable better results for all tree cultures used in the study.

3. CONCLUSION

The integrated method to forecasting conditions of objects on the underlying surface using satellite images is developed. This method was implemented as a standalone software package.

There is presented an example of applying the method in yield forecasting problem on real remote sensing data and information from official statistical data. Model with factor adjustment for regions and temporal trend allows obtaining forecasting errors from 13% to 17% depending on the culture, which is good accuracy for such kind of forecasts.

The main advantage of the suggested approach is the possibility to use free to access information, including satellite multispectral images and official statistical data. Actually, finding out the appropriate form of forecasting function on the basis of remote sensing images and data from the official statistics makes it possible to achieve fairly accurate results in forecasting of the object conditions.

We plan to continue this study with enhanced forecasting models in order to improve the accuracy and generality of the object conditions prediction method.

ACKNOWLEDGEMENT

The authors owe particular thanks to Academician Valery G. Bondur for the help with formulation of the problem as well as Professor Vladimir I. Tsurkov and Dr. Ivan A. Matveev for useful discussions.

REFERENCES

Bondur V. et al, 2010. Automated processing of time series of space images for studying the dynamics of lineaments for earthquakes prediction. *Izvestiya Vuzov. Geodeziya i Aerofotos'yomka (in Russian)*, No.4.

Bondur V. 2011. Aerospace Methods and Technologies for Monitoring of Oil and Gas Areas and Facilities *Izvestiya. Atmospheric and Oceanic Physics*, Volume 47, № 9.

Fischer, R. et al, 2009. Can Technology Deliver on the Yield Challenge to 2050? *Expert Meeting on How to Feed the World, Food and Agriculture Organization of the United Nations.* Rome, Italy, pp. 8-12.

Murynin A. et al, 2013. Analysis of Large Long-term Remote Sensing Image Sequence for Agricultural Yield Forecasting. *Image Mining. Theory and Applications. Proceedings of the 4th International Workshop on Image Mining. Barcelona, Spain, February*, pp.48-55.

Phillips, L. et al, 2008. Evaluating the species energy relationship with the newest measures of ecosystem energy: *NDVI versus MODIS primary production. Remote Sensing of Environment*, Vol. 112, Iss. 9, pp. 3538-3549.

Short Papers

MULTI-VIEW OBJECT DETECTION USING REGION-BASED RANDOM FOREST CLASSIFIERS

Ji-Hun Jung[1], Byoung-Chul Ko[1], Jae-Yeal Nam[1] and Young-Do Joo[2]

[1]Department of computer Engineering, Keimyung University Shindang-Dong Dalseo-Gu, Daegu, South Korea

[2]Department of Communication Business, SMEC Co. - Daegu Fusion R&D Center, Hosan-Dong, Dalseo-Gu, Daegu, South Korea

ABSTRACT

This paper presents a robust method for detection and classification of objects that is invariant to intrinsic and extrinsic variations in a complex background. To improve the object detection performance, we first select semantic regions and train their classifiers by considering the specific characteristics of regions. Each region-based classifier is boosted using a random forest classifier that is an ensemble of decision trees. An integration of random forest classifiers of single views is used to detect the most likely object position in the image. Then, because each view overlaps with neighbouring views, a weighted sum of the probabilities of three neighbouring views is used as the final score to determine the location and viewing angle of the object. The proposed algorithm is successfully applied to various PASCAL images, and its detection performance is better than the performances of other methods.

KEYWORDS

Object detection, region-based classifier, random forest, PASCAL, multi-view

1. INTRODUCTION

Object detection and recognition is one of the main issues of computer vision fields, because of its various applications such as image retrieval, video surveillance, object tracking, augmented reality, and human-computer interaction. Therefore, many successful object detection approaches have been proposed in recent years. The initial approaches used the colour histogram (Swan et al., 1991), histograms of oriented gradients (HOG) (Dalal et al., 2005), and shape feature (Belongie et al., 2002). However, detection of objects in complex scenes involves wide intrinsic variations (e.g., pose, colour, texture) and extrinsic variations (e.g., lighting, viewpoint, occlusion). Therefore, some recent work has aimed to overcome these intrinsic and extrinsic variations by developing new features and multi-view models.

Latev (2009) introduced an object detection method for single views by combining AdaBoost learning with local histogram features. This method selects an exhaustive set of rectangular regions in the normalized object window and uses AdaBoost to select histogram features and learn an object classifier.

For multi-view object detection, Razavi et al. (2010) proposed an extension of the Hough-based object detection to handle multiple viewpoints. This builds a shared codebook by jointly considering different viewpoints. Sharing features across views allows better use of training data and increases the efficiency of training and detection.

To improve the object detection performance for arbitrary viewpoints, this study proposes a novel approach to the detection and classification of objects. The approach is based on the oriented centre-symmetric local binary pattern (OCS-LBP) feature and region-based classifiers that consist of a random forest for each viewing angle. Moreover, the location and viewing angle of the object is determined by integration of three neighbouring views, because each view somewhat overlaps with its neighbouring views.

2. REGION SELECTION AND TRAINING OF REGION-BASED CLASSIFIERS

Inspired by Laptev (2009), we first randomly select an exhaustive set of rectangular regions in the normalized object window. In addition, we propose the region-based classifiers (models) using a random forest classifier and integral histogram based on the OCS-LBP feature. Then, the probabilities of the region-based classifiers are combined for each viewing angle. The final score of a primary view is computed from the multi-view region-based classifiers, and the location and size of an object are taken as confirmed if the final score exceeds the threshold.

2.1 OCS-LBP Histogram Construction Using Integral Histogram

Because of its low computational complexity, the centre-symmetric local binary pattern (CS-LBP) feature has recently been adopted for detection of humans and objects. However, since the original CS-LBP neglects the orientation and magnitude information, we take a different approach to the oriented CS-LBP (OCS-LBP) and thus use a new lower-dimensional feature-oriented CS-LBP.

To construct an OCS-LBP, the gradient orientation is taken as confirmed when the differences between pairs of opposite pixels in a neighbourhood are over the threshold. The gradient magnitude for an orientation is influenced by each pixel according to the closest bin in the range from 0° to 360° at 45 intervals. Once the gradient orientations of all neighbourhoods are estimated, a histogram of each *k*th orientation in a neighbourhood is binned. The OCS-LBP feature for each *k*th orientation is obtained by summing all the gradient magnitudes whose orientations belong to the *k*th bin. After that, the final set of *k* OCS-LBP features is normalized by the min-max method.

Again inspired by Laptev (2009), we subdivide each region into a 2×2 grid and compute an OCS-LBP feature separately for each part to preserve local variation using the integral histogram. The four OCS-LBP features are then concatenated into one OCS-LBP feature of dimension 32 (4×8).

2.2 Region Selection

To select proper locations and sizes for the regions and their classifiers, we first collect *N* training samples by cropping and resizing the original images to a size of 80×40, including objects. Negative samples are extracted from background regions. The training samples are divided into a learning group of *N*/2 and testing group of *N*/2 to select the proper locations and sizes for the regions.

For training, we first randomly generate *k* regions (p_1, …, p_k) with sizes ranging from 10×10 to 80×40 pixels. Then, a seed region p_1 is picked and OCS-LBP features are extracted from this region and its *n* corresponding regions (same location with the same size) in the learning group. For each selected region p_i, we restrict the training samples to the same class and learn region-based classifiers using OCS-LBP and random forests. Then, OCS-LBP features with the same location and size are extracted from the testing group. We apply trained random forests to testing data and compute the posterior probability of the positive class. This process is repeated until the iterations reach the maximum of *k* regions (p_1, …, p_k). After estimating the probabilities of all *k* regions, we select the *m* (= 100) regions with the highest probabilities. Finally, weights w_i for the region-based classifiers are computed on the basis of their relative probabilities Pr_i by using Eq. (1).

To select semantic regions, we use the random forest instead of the widely used support vector machine (SVM) and AdaBoost. A random forest is a decision tree ensemble classifier, where each tree is grown using some form of randomization. Random forests have the capacity to process vast amounts of data at high training speeds (Breman, 2001).

$$w_i = \frac{\Pr_i(object)}{\sum_{i=1}^{m} \Pr_i(object)} \tag{1}$$

3. MULTI-VIEW OBJECT DETECTION

After training all classifiers on the 12 multi-view angles, the test windows are applied to the region-based classifiers for each viewing angle θ. The total probability of each view is computed by arithmetically averaging each distribution of region-based classifiers P = (Pr1, Pr2, …, Prm) according to their importance weights:

$$\widetilde{P}^{\theta}(object) = \sum_{i=1}^{m} w_i \cdot \Pr_i^{\theta}(object) \tag{2}$$

In this study, because some regional detectors may give false results when the object is occluded, the probability of a single object view $\widetilde{P}^{\theta}(object)$ is computed from all regions. After the test window is applied to the region-based classifiers of all views, the primary view is estimated using the max operation:

$$\theta^{*} \cong \arg\max_{\theta_i}(\widetilde{\mathrm{P}}^{\theta_1}, \cdots, \widetilde{\mathrm{P}}^{\theta_{12}}) \tag{3}$$

Then, because each view somewhat overlaps with its neighbouring views, the final score used to determine the location and viewing angle of the object is computed as the following weighted sum of the probabilities of the primary view and its neighbouring views:

$$\begin{aligned} score^{\theta^{*}}(object) = sw_1 \cdot \widetilde{P}^{\theta^{*}}(object) + sw_2 \cdot \widetilde{P}^{\theta^{*}-1}(object) \\ + sw_3 \cdot \widetilde{P}^{\theta^{*}+1}(object) \end{aligned} \tag{4}$$

where sw_1, sw_2, and sw_3 are the relative weights for integration of the final score, and we set these weights to 0.6, 0.2, and 0.2, respectively, according to experiments. If the final score exceeds the threshold *T*, the primary view is accepted as showing the real object with viewing angle *θ*.

4. EXPERIMENTAL RESULTS

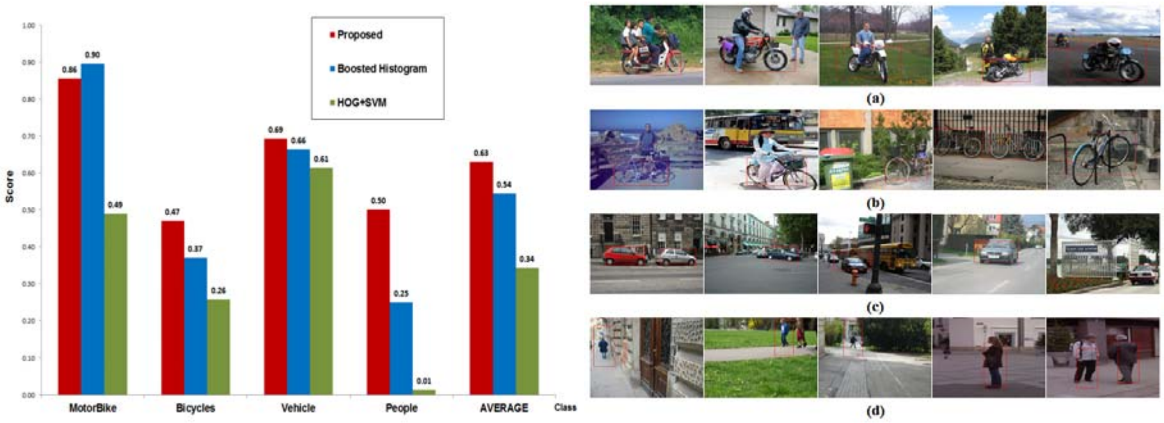
We performed experiments using the PASCAL 2005, 2006, and 2012 challenges (Web-1), which include a wide variety of objects such as people, bicycles, cars, horses, and cows. For training, we used 100 positive samples per viewing angle and randomly selected 100 negative samples from background.

For object detection, we used the standard window scanning method and applied the region-based model to the image windows with densely sampled positions and sizes. Eight times during the testing, we upsampled the 80×40 test window (40×80 for humans). The upsampling performed was by a factor of 1.2, and the maximum window size was 288×144. Precision methodologies were used to evaluate the experiments for object detection.

To evaluate the proposed algorithm, we tested its performance on four classes of VOC 2005 and VOC 2006: motorcycles, bicycles, vehicles, and people. The proposed region-based classifier with random forests was compared with the existing algorithms that yield the best performance, which are the HOG with a linear SVM (HOG+SVM) (Dalal et al., 2005) and the boosted histogram (Laptev, 2009).

As shown in Fig. 1(i), we confirmed that our proposed algorithm produced better object detection performance than the other two methods. In each of the experiments, we measured the performance using the average classification precision of the four classes. As shown in Fig. 1(i), the average score achieved by our method was 63%, which was 29% higher than that achieved by the HOG+SVM method and 9% higher than that achieved by the boosted histogram method. The best detection rate of 86% occurred for motorcycles. In contrast, the rate for bicycles was 47%. This might be due to the cluttered backgrounds of test images or the diverse views and occlusions of bicycles. The main reason for the higher detection rate of the proposed method is that the regions are selected reasonably in the salient part of the object and random forest classifiers associate the regions correctly with a specific object by the same method of ensemble of trees.

Figure 1(ii) shows some object detection results obtained with our proposed method using the PASCAL test sets.



(i) *(ii)*

Figure 1. Object detection results obtained using PASCAL images: (i) average precision comparison between the proposed method and three other methods using the same VOC 2005 and 2006 challenges; (ii) some examples of object detection with the proposed multi-view object method for (a) motorcycles, (b) bicycles, (c) vehicles, and (d) people.

5. CONCLUSION

This paper presented an object detection method for dynamic intrinsic and extrinsic variations in a complex background. To select proper locations and sizes for regions and their classifiers, we used the random forest, which is a decision tree ensemble classifier that reduces the training and testing periods. Random forests have the capacity to process vast amounts of data at high training speeds. After training region-based classifiers on 12 multi-view angles, we performed a max operation to find the primary viewing angle of the object, and we integrated neighbouring probabilities of viewing angle to combine features of submodels.

ACKNOWLEDGEMENT

This research was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation.

REFERENCES

Belongie S., Malik J., Puzicha J., 2002. Shape matching and object recognition using shape context. *IEEE Trans. on Patt. Anal. and Mach. Intell*. Vol. 24, No.4, pp. 509-522.

Breiman L., 2001. Random Forests. *Machine Learning.* Vol. 45, No. 5, pp. 5-32.

Dalal N., Triggs B., 2005. Histograms of oriented gradients for human detection. *Proc. of IEEE Conf. on Comp. Visi. and Pat. Rec*. San Diego, CA, USA, pp. 886 - 893.

Laptev I., 2009. Improving object detection with boosted histograms. *Image and Vision Computing.* Vol. 27, No. 5, pp. 535-544

Razavi N., Gall J., Gool L. V., 2010. Backprojection revisited: scalable multi-view object detection and similarity metrics for detections. *Proc. of 11th European Conf. on Computer Vision.* Heraklion, Greece, pp. 620-633.

Swain M., Ballard D., 1991. Color indexing. *Int. J. of Computer Vision*. Vol. 7, No. 1, pp. 11-32.

Web-1: http://pascallin.ecs.soton.ac.uk/challenges/VOC/

SIMULATION OF LICHEN AND MOSS GROWTH ON WOOD

Korakot Prachumrak and Janejira Chatchawal

Department of Computer Science, Faculty of Science, - King Mongkut's Institute of Technoloty, Ladkrabang, Bangkok, 10520, Thailand

ABSTRACT

This paper proposes the simulation of Cellular Automata on 3D surface to show lichen and moss growth on wood. Lichen and moss were selected since they have 2 factors that are similar to Cellular Automata: first, they all have cells, and second, the cells can be born, alive and dead. Therefore it is assumed that the Cellular Automata and the 3D Surface Cellular Automata can be used to simulate lichen and moss growth on wood. The simulation can be done as follows. First, the 3D Surface Cellular Automata is used to create the grids on the 3D planes. Secondly, the Cellular Automata is applied to define the changes on the cell level of the surface of 3D models of wood. Finally, the 3D Surface Cellular Automata is again applied for the rendering and the shading of the 3D wood models. The results of the experiments show that the Cellular Automata and the 3D Surface Cellular Automata can be used to simulate the lichen and the moss growth on wood.

KEYWORDS

Surface simulation, lichen growth, moss growth, cellular automata.

1. INTRODUCTION

Simulating changes on 3D models is one of the interesting areas in computer graphics. In the past, the basic method was basically mapping many layers of textures onto models. This method took a lot of afford and time consuming, especially when representing surface aging. Later, mathematical models were applied to simulate changes on 3D surfaces, for example, on modeling and rendering of metallic patinas [1], surface aging by impacts [2], CG representation of wood aging with distortion, cracking and erosion [3], modeling lichen growth [4] [5] and simulating moss growth [6].

This research presents a new method to simulate changes on wood. Wood in the nature is often covered by lichen and moss when it is aging and weathering. In order to show the natural phenomena as in the real scene, this research proposes a method to simulate lichen and moss growth on wood. This method applies cellular automata rules on 3D surfaces. The advantage of this method is that it can model and render the scene in a fast and realistic process.

2. RELATED WORKS

There has been earlier research on simulation of lichen and moss as follows:

Sumner [5] proposed pattern formation to simulate the lichen growth by applying the mathematical model called Diffusion Limited Aggregation (DLA). This model randomly spreading particles, when a particle collides with a cluster of particles they aggregate. The disadvantages of this model are that it results into a regular pattern and it is also time and memory consuming.

Later, Desbenoit [4] adapted the DLA model to control the lichen growth. This new method reduces the distance between a new particle and a cluster of particles, so it is less time consuming. This new model generates lichen in 3 steps, first, randomly generating spores onto the surface of the modeling object, second, spreading the lichen and last, simulating the lichen growth.

Chen [6] suggested a method to simulate weathering phenomena on models. It is called visual simulation of weathering by γ-ton tracing. γ-ton, which means old in Greek, is a method to iteratively process. In each process, a γ-ton is emitted from a γ-ton source to a 3 dimensional point source to create a γ-ton map. The γ-ton map is used to generate the textures over the model. This technique can be applied to many types of modeling including the growth of moss on a house wall.

As lichen and moss usually grow together on wood, this paper suggests a new method to simulate both lichen and moss growth together on wood models.

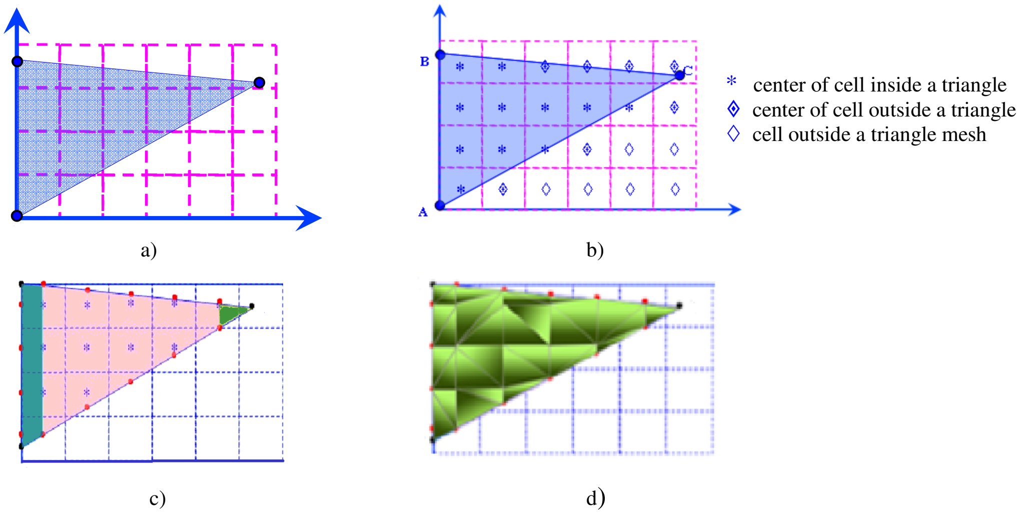
3. CELLULAR AUTOMATA TO SIMULATE LICHEN AND MOSS

3.1 Cellular Automata for 3D Surface

In 1999 Gobron and Chiba [7] were the first to introduce the application of cellular automata to simulate models. Their first work suggested applying cellular automata with voronoi diagram to generate green patina on copper. Later they [8] proposed how to apply cellular automata on cracked pattern and [9] on weathering simulation on buildings. The technique of merging cellular automata for simulating surfaces was also explained in [10]. Cellular automata was also applied on the work in 2007 to simulate Retina using cellular automata and GPU programming [11]

According to [7], cellular automata model is applied on 3D surfaces. The advantage of this method is that it can be applied onto many types of objects. The process of creating cellular automata is summarized as follows:

1) input a 3D model with Triangle mesh structure, then create grids onto each triangle face as in fig. 1.a.
2) define characteristic of each cell as in fig. 1.b
3) link the cells at the edges of each side of the triangle
4) assign the 8 neighbours to each cell
5) apply cellular automata rules as explained in the next section.
6) divide each triangle mesh into 3 parts (fig.1.c)
-part 1 🟦 has a size of half a grid in width
-part 2 🟪 dominates most of the mesh
-part 3 🟩 less than or equal to half a cell in width
7) render cells on each triangle mesh by using OpenGL functions which are Triangle Fan and Triangle Strip. Each center of the grid cell is a vertex of the Triangle (fig.1.d).


B
A
C
* center of cell inside a triangle
◈ center of cell outside a triangle
◇ cell outside a triangle mesh

a)

b)

c)

d)

Figure 1. a), b), c), d) show how to create cellular automata on a triangle mesh.

3.2 Cellular Automata Rules

Cellular automata (CA) was first introduced by Nuemann [12] and applied by Conway's "Game of life" [13]. CA can simulate 2 important rules of nature. First, each living creature consists of the smallest units called cells, similarly, cellular automata is also the smallest grid cells. Second, CA can represent dead or alive states similar to the rule of nature. From these 2 important rules, this research applied CA to simulate the lichen and moss growth on wood.

Cellular automata consists of:

1) Cell, the smallest unit in cellular automata.
2) States, each cell can represent one state at a time: dead or alive.
3) Neighbours, they are cells surrounded the middle cell we consider. Here, we applied Moore Neighbours which take 8 cells surrounded as neighbours.
4) Round, each cell changes state in every round.
5) Rule, defining cell state in a round.

In this research, S/B is used to define rules of changing states. S, survive, is the number of the surrounded neighbours to make the living cell being alive in the next round. B, born, is the number of the neighbours to make the death cell being alive.

Here is an example of applying S/B with the game of live. This case, S/B is 23/3. S = 2 3 which means that a living cell is alive in the next round if it is surrounded by 2 or 3 living neighbours. B = 3 which means that a dead cell becomes alive in the next round if it is surrounded by 3 living neighbours.

3.3 Cellular Automata Rules to Simulate Lichen and Moss Growth

The characteristic of lichen on wood is light green and flat. It smoothly covers the surface of the wood as shown in fig. 7. After many experiments, S/B = 012345678/3678 was found to be the best rule to simulate lichen growth on wood. In contrast, moss appears similar to a dark green carpet growing with different height on the wood surface as shown in fig. 7. For moss, the best rule is S/B = 012345678/1478. As moss has some thickness on wood surface, the thickness of moss can be simulated as shown in the following sections.

3.3.1 Moss Thickness Simulation

The thickness of moss starts from the middle of the cell in the same direction as that of a normal vector of the triangle mesh that we considered. The density of the moss can be randomly put on the surface arbitrary.

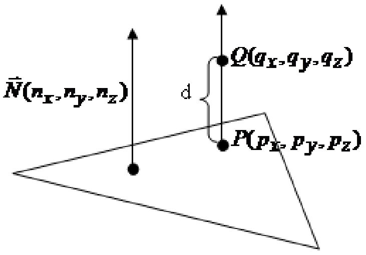


Figure 2. Creating a new point Q for moss thickness.

As shown in fig. 2, P is the middle of a cell on ABC triangle; d is the distance between Q and P; $\vec{N}$ is Normal vector of ABC plane. As $\overrightarrow{PQ}$ // $\vec{N}$, we can define Q.

3.4 Rendering Technique

This work is programmed with OpenGL, the rendering technique here applied Triangle Strips and Triangle Fans as suggested in [7].

3.5 Simulation of Lichen and moss Growth on Wood

Simulation of lichen and moss growth on wood is an iteration process as shown on fig. 3. This method can be explained as follows: first, creating grids on each triangle mesh and then calculating the middle of each cell. and defining their property whether it is inside or half inside or outside the triangle mesh. Then, link the cells at the edges of the meshes. Next, find an intersection between the line that divides half a cell and the edge of the mesh. After that, apply lichen and moss with cellular automata rules, and if moss has thickness, apply the thickness for it. Finally, render and simulate the changes on wood.

4. EXPERIMENTAL RESULTS

An experiment of lichen and moss growth was done on Stanford bunny [14], fig. 4.a, which is a standard 3D model. It consists of 4968 triangle meshes. The size of the grid is set to be 0.03, so the numbers of cells are 389,942. Fig. 4.a. Combine the cells at the edges of the meshes. Fig. 4.b. Randomly simulate the lichen growth with the rule S/B = 012345678/3678 which found to be the best rule after many experiments

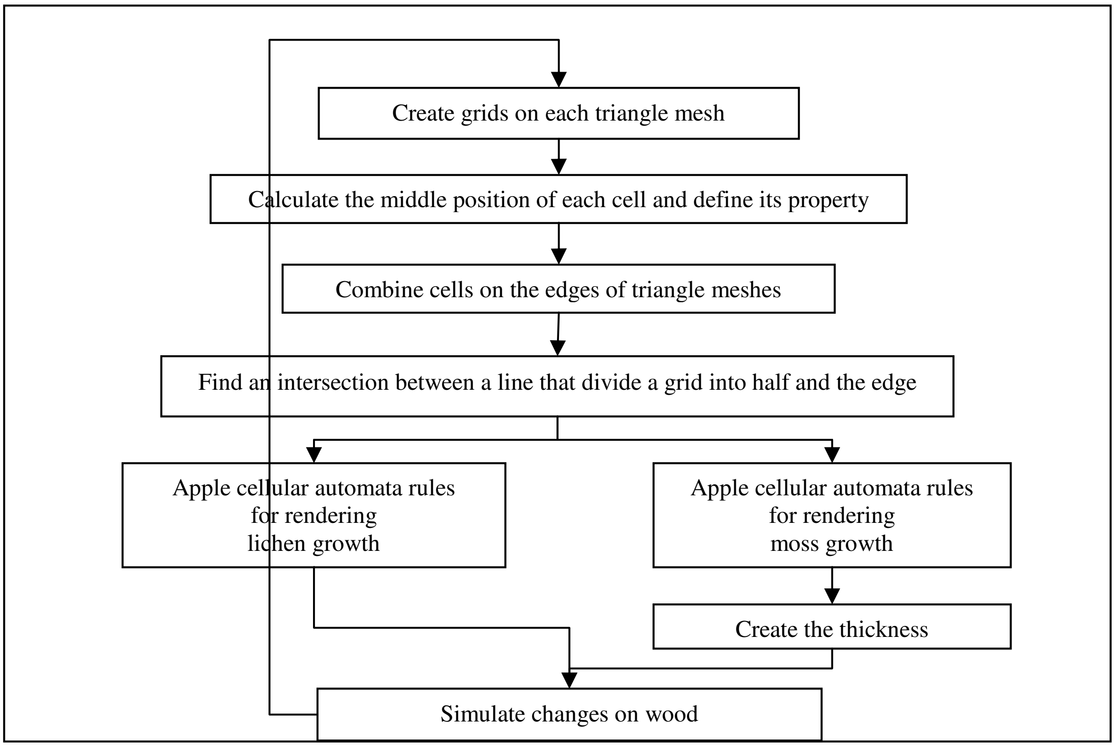

Create grids on each triangle mesh
Calculate the middle position of each cell and define its property
Combine cells on the edges of triangle meshes
Find an intersection between a line that divide a grid into half and the edge
Apple cellular automata rules for rendering lichen growth
Apple cellular automata rules for rendering moss growth
Create the thickness
Simulate changes on wood

Figure 3. method of simulating lichen and moss growth on wood.

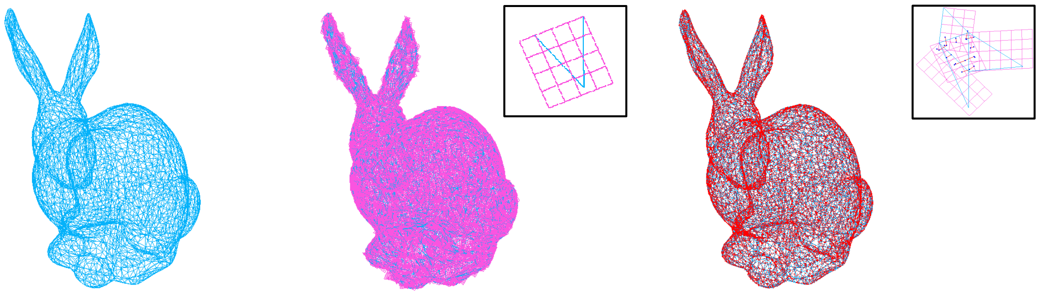


Figure 4. a) Stanford bunny, b) create grid on each mesh, c) combine cells at the edges.

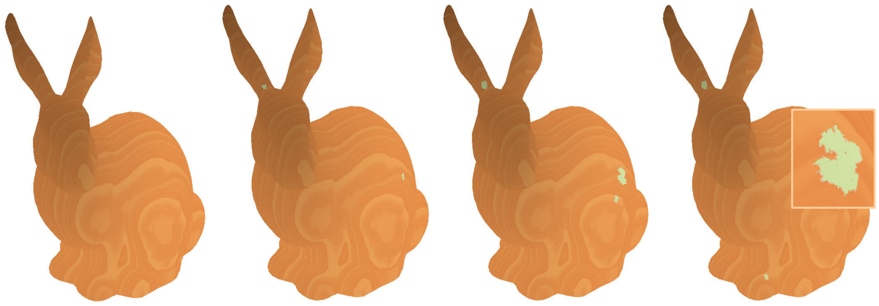


Figure 5. simulation of lichen growth on Stanford bunny at round = 0, 20, 40 and 50.

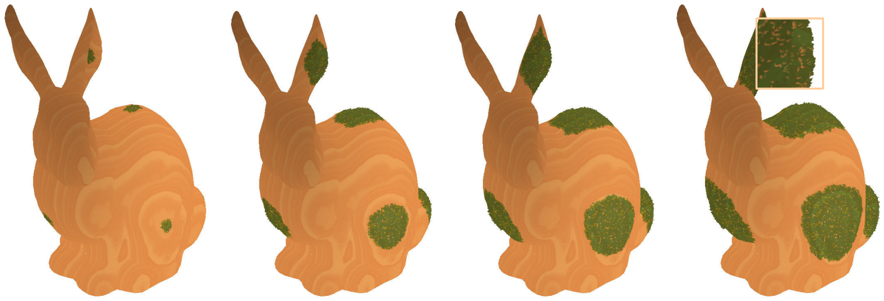


Figure 6. simulation of moss growth on Stanford bunny at round = 10, 20, 40 and 50.

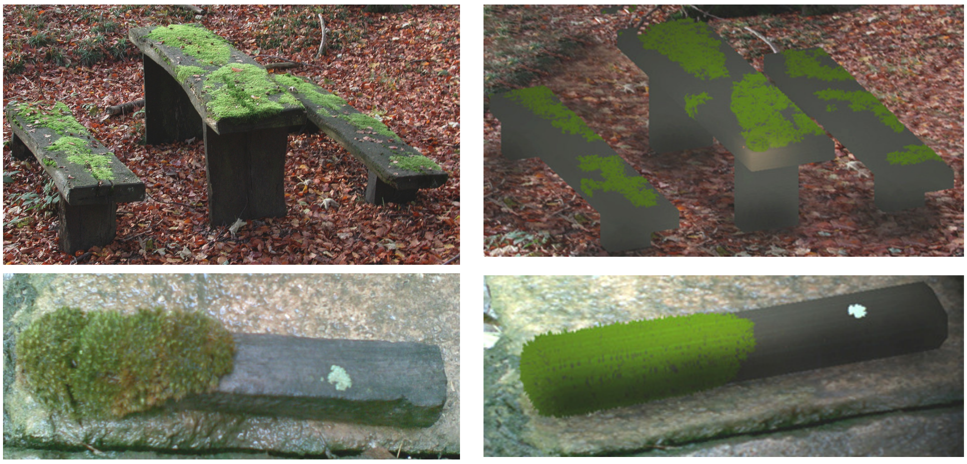


Figure 7. Simulation of lichen and moss growth of 3D models with cellular automata (right) comparing with the real scene (left).

5. CONCLUSION

Cellular automata is a theory that imitates cells of living things. Each cell has 3 states: birth, alive and death. This research applied cellular automata to model lichen and moss growth on wood simultaneously. Following numerous experiments, the best rules to simulate lichen and moss are presented here. We applied the cellular automata rules on 3D surfaces of objects including Stanford bunny and many objects to simulate the real scene. This method can render realistic results within a minute. The future work can be on the application of Cellular automata to simulate other changes on models.

REFERENCES

[1] Dorsey, J. and Hanrahan, P. Modeling and rendering of metallic patinas. *Proceedings of SIGGRAPH '96*, 1996. pp.387–396.

[2] Paquette, E., Poulin, P. and Drettakis, G. Surface aging by impacts. *Proceedings of Graphics Interface 2001*, 2001. pp.175–182.

[3] Yin, X., Fujimoto, T. and Chiba, N. CG Representation of Wood Aging with Distortion, Cracking and Erosion. *The Journal of the Society for Art and Science*, vol.3, no. 4, 2004. pp.216-223.

[4] Desbenoit, B., Galin, E. and Akkouche, S. Simulating and modeling lichen growth. *EUROGRAPHICS*, vol. 23, no. 3, 2004. pp.341-350.

[5] Sumner, R.W. "Pattern formation in lichen." Ph.D. Thesis, Départment of Electrical Engineering and Computer Science, Massachusetts institute of technology. 2001.

[6] Chen, Y., Xia, L., Wong, T. T., Tong, X., Bao, H., Guo, B. and Shum H. Y. Visual Simulation of Weathering By gamma-ton Tracing. *ACM Transactions on Graphics (SIGGRAPH 2005 issue)*, vol. 24, no. 3, Aug. 2005. pp.1127-1133.

[7] Gobron, S. and Chiba, N. 3d surface cellular automata and their applications. *Journal of Visualization and Computer Animation*, vol. 10, 1999. pp.143–158.

[8] Gobron, S. and Chiba, N. Crack pattern simulation based on 3D surface cellular automata. *The Visual Computer*, vol. 17, no. 5, 2001. pp.287-309.

[9] Even, P. and Gobron, S. Interactive three-dimensional reconstruction and weathering simulations on buildings. *CIPA XXth International Symposium: International Cooperation to Save the World's Cultural Heritage*, Torino, Italy, Oct. 2004. pp.796-801.

[10] Gobron, S, et al. "Merging cellular automata for simulating surface effects." Cellular Automata. Springer Berlin Heidelberg, 2006. pp. 94-103.

[11]Gobron, S., Devillard, F., & Heit, B. 2007. Retina simulation using cellular automata and GPU programming. *Machine Vision and Applications*, vol.18, no.6, pp 331-342.

[12] MathWorld. "Cellular Automata." [Online]. Available : ttp://mathworld.wolfram.com/topics/CellularAutomata.html.

[13] Wolfram, S. "SOME HISTORICAL NOTES." [Online]. Available : http://www.wolframscience.com/reference/notes/876b. 2008.

[14] Stanford Computer Graphics Laboratory. "The Stanford 3D Scanning Repository." [Online]. Available : http://www - graphics.stanford.edu/data/3Dscanrep/index.html. 1994

REALTIME PARTICLE SYSTEM SIMULATION AND RENDERING IN EMBEDDED SYSTEMS

Jens Ogniewski and Ingemar Ragnemalm

Information Coding Group, Linköpings University - 581 83 LINKÖPING, Sweden

ABSTRACT

The market for games for mobile phones/tablets is probably the fastest growing in the whole computer game industry. Although many of these games feature graphics which were out of reach for these systems not long ago, their quality is still far from what can be reached on modern PCs, and many algorithms used in PCs are a bad fit for these systems. For example, volumetric particle systems are very difficult to simulate and render in realtime on modern smartphones/tablets. This paper presents the first work on particle system simulation and rendering on embedded systems in realtime. This was achieved by approximating volumetric systems by 2D-systems and by using a novel, physically motivated yet simple particle motion model instead of a computational complex solver for e.g. Navier-Stokes equations.

KEYWORDS

Particle effects, Embedded Systems, Real-time, Computer Games

1. INTRODUCTION

The visualization of particle effects has been a topic of much interest since the beginnings of computer graphics, and many approaches have been presented, most of which use volumetric particle systems, e.g. (Wrenninge et al., 2010). These are based on particle movement in a so called voxel grid, which is a discretized, closed space (realized e.g. by 3D-textures). However, these grids use a very large amount of memory, and thus several approaches have been suggested for their effective compression like octrees (Laine & Karras, 2011), trading lower memory footprint for higher runtime. To render the particles, different methods are used depending on the exact effect that should be simulated, e.g. Marching Cubes (Lorensen & Cline, 1987) for liquids or light transport for smoke/clouds , e.g. (Hadwiger et al., 2009).

Most of the recent work concentrates on reducing the runtime, for example by introducing precomputing like Kun Zhou et al. (2008) or Yubo Zhang et al. (2012), or by modification of the grid, e.g. (Horvath & Geiger, 2009) or (Selleg et al., 2005). These approaches can run in realtime on contemporary PCs, however not on embedded systems, since modern smartphones/tablets have a different architecture. Here, the CPU and the GPU, but also all other integrated systems like communication, I/O etc. share the same memory and the same bus, which therefore become a bottleneck. This has however big advantages in energy savings and cost, and is thus unlikely to change. The GPUs are highly optimized towards size and energy consumption, but often have to render to high resolution screens. This means that graphic algorithms have to be carefully optimized for runtime, but especially for memory consumption. For particle effects, most designers use so called particle systems, which in these cases mean several (often animated) billboards moving in predetermined or partly-random patterns, as described in e.g. (Harris & Lastra, 2001). A few papers have presented volumetric rendering for embedded displays, like Moser & Weiskopf 82008) or Rodriguez & Alcocer (2012), omitting however the simulation part needed for animated particle effects. Furthermore, the reported frame-rate of typically only a few frames per second is too low for real-time applications like games, and these works do not include popular effects, like lightning or advection (a method to introduce small-scale details through random noise which is offset by a turbulence field, see e.g. (Neyret, 2003) or (Qizhi Yu, 2011). Finally, Krüger & Westermann (2005) and in Guay et al. (2011) suggested the use of a 2-dimensional approximation to emulate a full volumetric system. This is based on the observation that a fairly good result can be achieved solely by knowing how many particles any possible ray from the observer through the particle system would hit. Due to its low complexity, this is the approach we use here as well. We further optimize their work by using a physically motivated model with much less complexity instead of a Navier-Stokes solver as applied by them. Furthermore, Krüger & Westermann (2005) use a 2-dimensional flow field, but move the particles in 3D, and Guay et al. (2011) introduce a fake depth during the simulation, while here no depth is used at all during simulation, thus reducing simulation time. Also, Guay et al. (2011) only described fire and it is unclear if their approach can be used for other effects as well, while we present here different particle effects proving the versatility of our approach.

To the best of our knowledge this is the first work where particle effects where generated on embedded systems by the simulation of particle movements.

2. 2D PARTICLE SIMULATION

As already pointed out, memory usage should be minimized as much as possible in embedded systems. Simulating the particle movement in 2D leads to a low memory footprint since only two 2D arrays are needed to save the data, one for the particles and one for the pressure field. We suggest here to remove the pressure field as well, so that the whole system uses only one single array, which will be called a *particle-field* in the following. Each cell of the field can contain 0 to 255 particles, which was chosen so that the whole field can be saved in a single color-channel of a texture. This also means that the simulation can be done in one single step, instead of integrating the pressure field first and then moving the particles accordingly in a second step. However, since no pressure field exist the particles have to be moved in a different way, and for that we chose a force-based approach in this work.

In the real world, the movement of particles are governed by a number of different forces. The most prominent include inertia (i.e. along the current trajectory), diffusion (from places where a lot of particles reside to places where fewer particles are), and external forces (e.g. gravity). Apart from these 3 forces, we added a random force as well to emulate other and small-scale effects. We found that this approach has the additional advantage that choosing the blend-weights of these forces helps to control the simulation and thus makes it easy for the designer to create the desired effect.

Of course, each simulated particle represents a high number of real particles. Thus, the movement of each particle in the simulated system can be seen as the average movement of all particles it presents. Therefore, a more accurate simulation would be received if several particles, that travel along the same movement vector, would be allowed to travel in slightly different paths. This could be described by e.g. a gauss distribution. Here, we suggest to use a cosine function instead as an approximation, since all forces (except for diffusion) can be represented by vectors. Thus, the dot-product between the force vector and a candidate direction can be used for the force-calculation.

This has the additional advantages that it can be computed fast in GPUs and that it guarantees that particles will be moved even if the force vectors do not align exactly with any of the candidate directions.

For the diffusion, simply the difference is used between the number of particles in the starting cell of the candidate direction and the number contained in the finishing cell of the candidate direction.

To simplify calculations, all particles are allowed to move only to neighboring cells. Also, during simulation the forces between a cell and its 8 neighbors are calculated only once for the each cell, not for each particle that the cell contains. This also means that an average direction vector is used for the inertia calculation, which can be saved comfortably in 2 color-channels of the particle field texture.

Since the forces are calculated separately for each direction, particles will be moved to a number of different neighbors, based on the strength of the calculated forces. In traditional approaches however only few directions would be used, and therefore the particles spread out more quickly using our approach.

An overview of the different forces and the resulting movements is given in figure 2b-2f. The particle field used has a size of 32x32. 2a shows the initial state, the others the system after 15 simulation steps.

3. EVALUATION

To test the method it was implemented on a Nexus 10, which is a tablet running Android on a Samsung Exynos 5 Dual processor. The GPU included in this chip, a Mali T-604, is by the time of this writing a better middle class GPU for embedded systems. Comparing it to concurrent PC graphics-cards, its limitation can be clearly seen. While current PC graphics-cards can have more than 3000 cores running at more than 1 GHz, as well as up to 4 Gbyte dedicated memory, the Mali T-604 has 4 cores running at 533 Mhz, and has to share the main memory (2Gbyte in this case) with the CPU and all other circuits integrated in the chip. Thus, it is not surprising that the Nexus 10 reaches only 8006 at the Icestorm benchmark, which puts it in the middle class of mobile device (the current maximum reached by a mobile device is 11346). By comparison, the middle class graphic card NVIDIA GeForce GTX 660 reaches 137246, or more than 17 times the performance of the Nexus 10. The highest value reached on a PC is (to the knowledge of the authors) 167203, but it should be pointed out that the benchmark is slightly biased towards low performance systems since it does not take advantage of many features that high-end cards offer.

Three different systems are included in the demo: *fire*, *smoke* and *water* (see figure 1 for example pictures). The sizes of the used particle-fields were 64x64 for the water and the smoke and 32x32 for the fire.

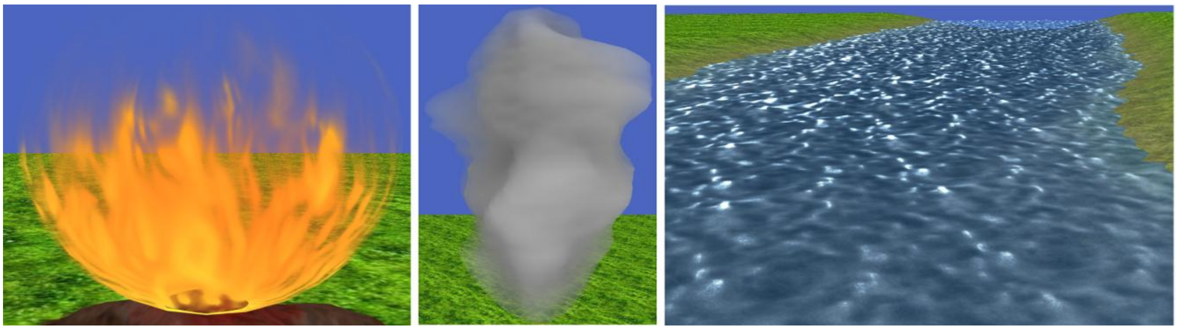


Figure 1. Example particle effects taken from the demo: 1a (left) fire, 1b (middle) smoke and 1c (right) water

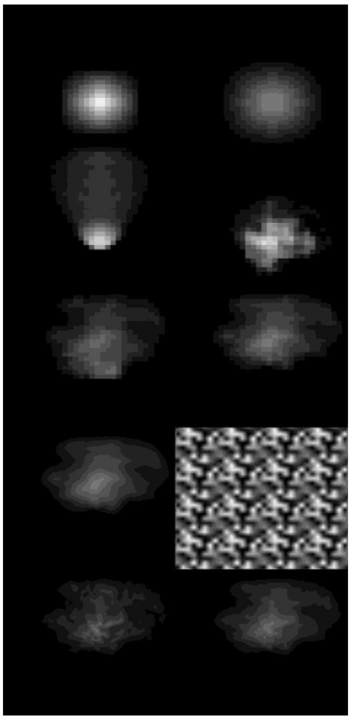


Figure 2. Example to illustrate the different steps of the suggested approach: 1st row: 2a (left) input particle field, 2b (right) movement according to entropy, 2nd row: 2c (left) movement along a common direction, 2d (right) random movement, 3rd row, 2e (left) all three forces combined, 2f (right) with additional inertia, 4th row, 2g (left) 2f drawn using linear interpolation, 2h (right) noise in a 8x8 texture repeated 4x4 times, 5th row, 2i (left) calculated advection, 2j (right) 2g and 2i blended together (1:1)

Although for some cases a 2-dimensional particle effect might be enough, e.g. a fire in a fireplace, in most cases depth needs to be introduced during rendering. This can be done e.g. by displacement mapping, which has the additional advantage that the designer can choose roughly which shape the object should have. The water and the smoke in the demo are rendered using this method, the latter one using a spheroid as basic shape. The fire was rendered purely as a texture on an otherwise unmodified spheroid. 1922 triangles were used for the fire, 7938 for the smoke, and 9660 for the water.

The particle-fields are also used to generate the textures projected onto the particle objects, and advection was added in case of the fire and the water. The average movement vectors included in the particle field were used as turbulence field for the advection. For the random noise, it was chosen to use a noise texture, which is a texture that contains random values and is a common solution for advection. This has the additional advantage that the noise can be custom tailored for the effect that should be reached, e.g. in the case of the fire it proved to be advantageous to have many high values concentrated in parts of the noise-texture and lower values in the rest, since this leads to more flame-like structures. Finally, to reduce memory consumption, noise textures of very small sizes are used, and repeated several times instead. The advection process is illustrated in figure 2g-2j.

The rendering was done in two different resolutions, 2560x1600 (which is the currently highest resolution available in tablets), as well as 1280x800, which is a high-end resolution of smaller tablets and smartphones at the time of this writing. For comparison, the iPhone 5 has 1136x640. The timing results are summarized in table 1. The time needed to render the ground is given as comparison; it uses 2386 triangles and 3 different textures depending on its height (with linear interpolation at the borders), but no other effects. These values include overhead like sending variables from the main program to the shader, and were taken from the viewpoint which (on average) lead to the worst result.

Looking at the numbers it is noticeable how close the values for the different simulations are, which is especially interesting since the particle-field of the fire is only 1/4th of the size of the other ones. Not surprisingly the smoke was rendered fastest, since its shader does not include much more than the actual geometry calculation, while the water fared worst due to its complicated advection scheme and its high number of triangles.

Table 1. Average simulation and rendering times, as well as the theoretical frame-rates of each system, which were calculated as $(1s\text{-}10{*}t_s)/t_r$, with t_s the simulation and t_r the rendering time, since the simulations run at a constant 10 fps.

	Simulation	Rendering 2560x1600	**Theo. fps 2560x1600**	Rendering 1280x800	**Theo. fps 1280x800**
Water	3.83 ms	46.1 ms	**20.9**	17.2 ms	**55.9**
Fire	3.42 ms	28.2 ms	**34.2**	10.5 ms	**92**
Smoke	4.12 ms	11.8 ms	**81.3**	6.92 ms	**139**
Ground	-	10.6 ms	**94.3**	7.14 ms	**140**

4. CONCLUSION & FUTURE WORK

A method was presented that shows how particle systems can be simulated and rendered in realtime on embedded systems, by approximating a volumetric system with a 2-dimensional one, and by using a novel highly efficient model for the particle movement. This method could prove very useful to include advanced particle effects in games for smartphones, tablets and similar systems. Performance is adequate for the target systems, but further optimizations are possible. We also aim for a unified system, where a designer would be given the possibility to control the look of the simulation and of the rendering by setting only a couple of parameters. Although this is already the case for the *smoke* and *water* simulations, we need to find further generalizations to eliminate the differences between the various cases to achieve this.

REFERENCES

Guay, M., Colin, F., Egli, R., 2011. Screen Space Animation of Fire. *Proceeding of SIGGRAPH Asia 2011 Sketches*

Hadwiger, M., Patric Ljung, P., Salama, C.R., Ropinski,T, 2009. Advanced illumination techniques for GPU-based volume raycasting. *Proceedings of ACM SIGGRAPH 2009 Courses*, Article No. 2

Harris, M.J., Lastra, A., 2001. Real-Time Cloud Rendering. *Proceedings of EUROGRAPHICS 2001*, vol. 20, no. 3

Horvath, C., Geiger, W., 2009.: Directable, high Resolution Simulation of Fire on the GPU. *ACM Transactions on Graphics 28*, 3, Article 41

Krüger, J., Westermann, R., 2005. GPU simulation and rendering of volumetric effects for computer games and virtual environments. *Proceedings of Eurographics*

Kun Zhou, Zhong Ren, Lin, S., Hujun Bao, Baining Guo, Heung-Yeung Shum, 2008. Real-Time Smoke Rendering Using Compensated Ray Marching. *ACM Transactions on Graphics 27*, 3, Article 36

Laine, S., Karras, T., 2011. Efficient Sparse Voxel Octrees. *IEEE Transactions on Visualization and Computer Graphics*, vol. 17 , iss. 8, pp. 1048-1059

Lorensen, W.E., Cline, H.E., 1987. Marching cubes: A high resolution 3D surface construction algorithm. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pp. 163-169

Moser, M., Weiskopf, D., 2008. Interactive volume rendering on mobile devices. *Vision, Modeling, and Visualization*

Neyret, F., 2003. Advected Textures. *Eurographics / Siggraph Symposium of Computer Animation*, pp. 147-153

Qizhi Yu, Neyret, F., Bruneton, E., Holzschuch, N., 2011. Lagrangian Texture Advection: Preserving both Spectrum and Velocity Field. *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 11 pp. 1612-1623

Rodriguez, M. B., Alcocer, P.P.V., 2012. Practical Volume Rendering in Mobile Devices. *Advances in Visual Computing*

Selle, A., Rasmussen, N., Fedkiw, R., 2005. A Vortex Particle Method for Smoke, Water and Explosions. *SIGGRAPH 2005, ACM TOG 24*, pp. 910-914

Wrenninge, M., Bin Zafar, N., Clifford, J., Graham, G., Penney, D., Kontkanen, J., Tessendorf, J., Clinton, A., 2010. Volumetric Methods in Visual Effects. *SIGGRAPH 2010 Course Notes*

Yubo Zhang, Zhao Dong, Kwan-Liu Ma, 2012. Realtime volume rendering using precomputed photon mapping. *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D '12)*, p.127

ISBN: 978-972-8939-89-2 © 2013 IADIS

MULTI-PERSON TRACKING BY DETECTION IN THERMAL CAMERAS

Joon-Young Kwak, June-Hyeok Hong, Byoung-Chul Ko and Jae-Yeal Nam
Dept. Computer Engineering - Keimyung University, Shindang-Dong Dalseo-Gu, Daegu, Korea

ABSTRACT

This paper presents a robust multi-person tracking algorithm by person detection using thermal images. The person tracking and detection continuously exchange information during the process each other in order to self-correct and self-update. First, a person is detected by analyzing hotspot regions in the images, and then, tracking is performed on each frame using the random forest classifier as the online learning algorithm. Second, we solve the association problem with our proposed association-check method, which considers both the spatial distance and the likelihood between target regions and detected regions. The proposed algorithm is successfully applied to a few thermal videos, including those with complex backgrounds, and the tracking results demonstrate reasonable performance.

KEYWORDS

Multi-person tracking, person detection, tracking by detection, random forest, association check.

1. INTRODUCTION

Human tracking is necessary in many real-life applications, such as video surveillance, traffic safety, human–computer interaction, and gesture recognition. In some tracking studies, the user manually defines a selection box that surrounds the person in the image by providing its location and size; then, tracking algorithms (e.g., particle filtering) overlay the image of the human over a static background. However, our research goal is to track multiple persons in a complex, moving scene without an initial person-selection box. Multi-person tracking in these situations is very challenging due to the many sources of uncertainty in terms of the locations of the subjects and the significant occlusions (Breitenstein, et al. 2011). To overcome such difficulties, tracking-by-detection algorithms (Breitenstein, et al. 2011, Kalal et al., 2010) were designed by combining detection and tracking information. Tracking-by-detection algorithms estimate a person's location in every frame independently, and detectors neither drift nor fail if the object disappears from the camera view (Kalal et al., 2010). However, the detectors deliver only a discrete set of responses, and they usually yield false locations and failed detections (Breitenstein, et al. 2011).

In addition, the person-tracking research based on a CCD camera is not effective in environments with poor illumination, whether indoors or outdoors, because of the changeable illumination, existence of shadows, and cluttered backgrounds. In contrast, thermal sensors allow for robust tracking of a human body in outdoor environments regardless of the time of the day, illumination conditions, and the subject's body posture.

In our work, we use a thermal camera instead of a CCD camera to track multiple persons, particularly at night and in outdoor environments, under the assumption that the camera was designed for mobile platform applications. Most importantly, our proposed tracking algorithm combines a tracker, based on online random forest learning, and a person detector, based on hotspot detection, in order to integrate two sources of information into one system and improve the tracking performance.

2. PERSON DETECTION AND TRACKING

2.1 Person Detection

The purpose of person detection is to initialize persons in a first frame and updates that person's location and size in subsequent frames. To detect person regions, we detect humans by analyzing hotspot regions in a thermal image, including the face and shoulder areas, because these areas emit high thermal energy that is presented our previous study (Ko et al., 2012). First, a threshold is applied to the thermal image to perform binarization for isolating the candidate hotspots that have high thermal energy using the flexible threshold. Next, morphological opening and closing is applied to the binary image to remove any small hotspots. The candidate hotspot is then enlarged in terms of both width and height to include the face and shoulder regions. Hotspots that include faces and shoulder areas tend to be very intense compared to the background and other body parts. Therefore, the boundary of hotspots are boosted by comparing the luminance contrast between the hotspots and the background. Finally, the candidate hotspots are classified as human or non-human using the extracted center-symmetric local binary pattern (CS-LBP) features and the random forest classifier.

2.2 Person Tracking

Particle filter is the most popular technique for object tracking. In this work, we used the particle filter and the random forest classifier (Ko et al., 2013) as the online learning methods for person tracking.

In order to learn the target model, training data were constructed using 15 positive samples that were detected by person detector and 30 negative samples that were randomly sampled from the background. From every frame, we retained 15 positive examples, including previously detected person regions, in order to avoid the template drift problem. Each particle is divided into six sub-blocks, and two types of random forest classifiers for the *i*th sub-block were learned using the local intensity distribution (LID) and the oriented centre-symmetric local binary pattern (OCS-LBP), which were extracted from the corresponding blocks in the 45 training examples.

3. PERSON DETECTION AND TRACKING

Data association is to assign most possible one detection to most possible one target in order to decide which detection should guide which tracker. During the process, the detection process and the tracking process exchange information in order to self-correct and self-update. In this study, we solved the association problem by modifying the greedy data association method proposed by Breitenstein et al. (2011). The main contribution of the proposed algorithm is that we use the random forest classifier in each frame as the online learning algorithm, and updated the state of the target by combining data association and tracking information.

In the first stage, when a person is detected, the target is assigned to a detection set **D**, and the initial vector of the *i*th detection is automatically set.

$$d_i = [cx_i, cy_i, w_i, h_i]^T$$

The initial tracker set **T** and the state vector of the *i*th tracker are set automatically depending on the detection result.

$$tr_i = [cx_i, cy_i, w_i, h_i, RF_i]^T$$

where (cx_i, cy_i) is the center position of the detected/tracked object, whereas w_i and h_i are the width and height of the bounding box of the same target. RF_i is the tracker classifier, determined by online learning at time *t*. If the person detection *d* is not associated with an existing tracker *tr* by the second frame, the target is regarded as a new person and the system creates a new tracker element. In contrast, if the target is not detected in several subsequent frames, it is regarded as an occlusion or a disappearance.

An association between detection and a tracker is estimated using the matching function

$$s(d_i, t_j) = \mathrm{dist}(d_i, tr_j) \cdot RF_i(di) \qquad (1)$$

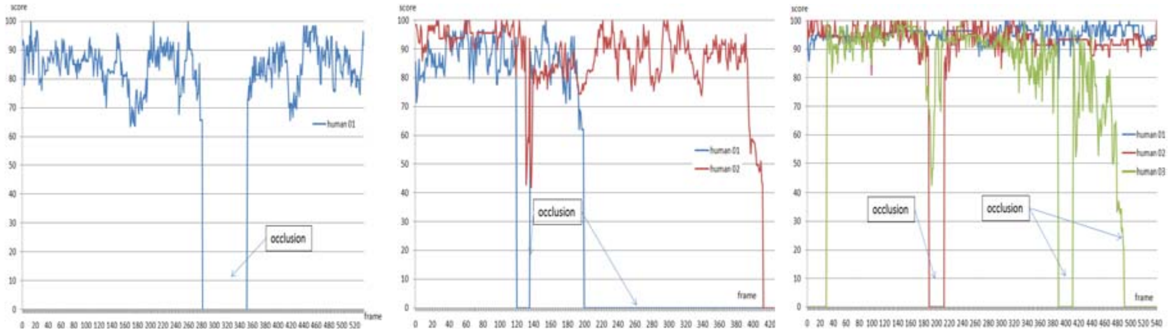
where $dist(d_i, tr_j)$ is the distance from the detected person *di* to the tracker *trj*. The function returns 1 if the result is greater than the minimum distance threshold γ. *RFi(di)* denotes the degree of association between the detected regions of *di* and the *RF* classifier of the *trj* tracker. Therefore, the higher the score, the better the match between detected person and tracking target.

Tracking update using person detection and online random forest learning involves the following procedures.

```
Algorithm : Tracking update by person detection using association check.
$Asso(d_i, t_i)$: final associations of detection di to tracker $tr_j$ in frame n
O: set of all occlusion trackers
$Occ(tr_j)$: the number of occlusion for tracker $tr_j$
1. If a person is detected first time,
   Then
     (1.1) a tracker and its state is initialized as the information of a detection
     (1.2) $Asso(d_i, tr_j) = 1$
     (1.3) Learning $RF_j$ for a tracker $tr_j$
2. Else
   2.1 For i=1 to N
         For j=1 to M
       (a) Matching score estimation between i-th detection and j-th tracker pair using matching
           function (1)
   2.2 Find maximum pair $(d_i, tr_j)$ between detection $d_i$ and tracker $tr_j$
   2.3 For all pairs
         If $s(d_i, tr_j) > T_2$
         Then
           (a) $Asso(d_i, tr_j) = 1$
           (b) State of tracker $tr_j$ is updated by combining the state of a current tracker $tr_j$ and state of
               detection $d_i$ using formula (2)
                   $tr_j^* = \alpha \cdot tr_j + (1 - \alpha) \cdot d_i$          (2)
           (c) Replace $tr_j$ to $tr_j^*$ in a set T
           (c) Learning $RF_j^*$ for a tracker $tr_j^*$
   2.4 If there is no matching for a tracker $tr_j$
         (a) For a tracker $tr_j$
             (a-1) If $(tr_j \notin O)\ O \ni tr_j$
             (a-1) Increase occlusion count $Occ(tr_j) = +1$
             (a-2) if $Occ(tr_j)$>T3 Remove $tr_j$ from O and T;
   2.3 If there is no matching for a detection $d_i$
       (a) For a detection $d_i$
           (a-1) Assign $d_i$ into a set T as a new tracker $tr_j^*$
           (a-2) The state of a new tracker $tr_j^*$ is initialized as the information of detection $d_i$
           (a-3) Learning $RF_j^*$ for a tracker $tr_j^*$
```

4. EXPERIMENTAL RESULTS

We performed experiments using three types of thermal videos containing several moving persons with background clutter, sudden shape deformation, unexpected motion change, and long-term partial or full occlusion among persons and objects. To evaluate the tracking performance of the proposed method, we used the spatial overlap metric defined by Yin et al. (2007). We define the spatial overlap and temporal overlap of tracks as the overlap between ground-truth (GT) tracks and system (ST) tracks in both space and time. Figure 1 shows the tracking performance on multiple objects found in three videos in terms of the overlap area *score(GTi, STj)*. The ground-truth of the target object was marked manually.



(a) (b) (c)

Figure 1. Results of tracking multiple objects using scores for (a) Video 1, (b) Video 2, and (c) Video 3

For all three videos, the proposed scheme had a significantly smaller error rate and more robust tracking results, regardless of the occlusion (full or partial) and the camera movement.

5. CONCLUSION

In this paper, we presented a multi-person tracking algorithm and proved that tracking errors caused by occlusion or drifting can be compensated using the detection information. First, a person is detected by analyzing hotspot regions, including the face and shoulder, and then each individual person is tracked using online learning based on the random forest classifier. To track multiple persons correctly in case of occlusion and drifting, we proposed an association-checking algorithm, where the state of the target is updated by combining data association and tracking information. In the future, we plan to design more statistical association-check methods to track overlapped persons more efficiently.

ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012-0008392)

REFERENCES

Journal

Breitenstein M. D. et al, 2011. Online multiperson tracking-by-detection from a single, uncalibrated camera. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33. Pp. 1820-1833.

Kalal Z. et al. 2010. Tracking-learning-detection. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6. Pp. 1 - 14.

Ko B.C. et al. 2012. Detecting human using luminance saliency in thermal images. *Optics Letters*, Vol. 37. pp. 4350 - 4352.

Ko B.C. et al. 2013. Human tracking in thermal images using adaptive particle filters with online random forest learning. *In IEEE Transactions on Circuits and Systems for Video Technology*, (submitted).

Conference paper or contributed volume

Yin F. et al. 2007. Performance evaluation of object tracking algorithms. *In proceedings of the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, Rio de janeiro, Brazil. pp. 1-8.

Reflection Paper

A USER INTERFACE FRAMEWORK FOR MODERN OPENGL BASED ON THE OPENGL UTILITY TOOLKIT

Ingemar Ragnemalm
Dept of Electrical Engineering - Linköping University

ABSTRACT

We discuss how the GLUT library can be modified to suit the needs of modern OpenGL, what parts can be excluded or need redesigning. Based on these results, we present a new cross-platform user interface framework for OpenGL, called MicroGlut, which covers the most vital features. We also draft how interesting features currently excluded from MicroGlut could be added.

KEYWORDS

GLUT, cross-platform, OpenGL, teaching, game programming, API.

1. INTRODUCTION

In this reflection paper we discuss cross-platform user interface libraries for modern OpenGL, suggesting an approach to re-implement and modernize of the popular GLUT library, including a possible approach to identify vital features to include, as well as ways to handle incompatible features.

OpenGL has evolved tremendously in recent years. With OpenGL 3, many old solutions were deprecated, and with 3.2, an explicit split into core functionality and compatibility mode encouraged focus on the core and phasing out the deprecated material. Thus, OpenGL 3.2 can be considered the base version for modern OpenGL, with versions up to 4.3 being largely similar.

Meanwhile, the confusion has grown on the choice of user interface libraries. OpenGL does not come with any built-in cross-platform user interface API. Instead, we have to rely on third party libraries like the OpenGL Utility Toolkit (GLUT) (Kilgard 1996, Khronos Group 2013), which is used by many OpenGL books for simplifying the user interface code (Schreiner 2009, Hearn et.al. 2011, Ragnemalm 2013). It stands out as a reference API that inspires many followers (Di Benedetto et. al. 2010). After GLUT was abandoned by its creator, FreeGLUT (2012) was created, an open source superset which re-implements as well as extends GLUT. Although FreeGLUT keeps GLUT updated, several other solutions have appeared (Khronos Group 2013). The alternatives may have their justification, but they make the situation complex, often confusing (Stack Exchange 2012) and cause incompatibilities. Also, the step to OpenGL 3.2+ have made large portions of the GLUT library obsolete.

We suggest that the GLUT API needs an update, but can be kept intact for substantial, still relevant parts. Other parts may be removed or redesigned as appropriate, staying close to the old API if possible.

As a step toward such a solution, we have created subset of GLUT called MicroGlut, implementing parts of GLUT most vital for modern OpenGL. In this paper we present the chosen subset, discuss why some parts were removed, and how replacements for these removed parts can be designed and packaged.

The paper is organized as follows: Section 2 identifies significant features of GLUT. Section 3 describes features of the new package. Section 4 discusses omitted functionality. Section 5 states conclusions.

2. CHOOSING FEATURES OF GLUT TO INCLUDE IN MICROGLUT

In order to update GLUT, we could have taken the existing FreeGLUT code and rewritten all code that depends on deprecated functions. However, the identification of obsolete code would be awkward. Thus, a re-implementation from scratch was called for, one where only truly needed functionality would be included.

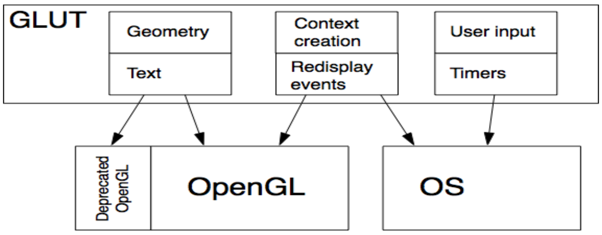


Figure 1. Discernible parts of the GLUT API.

The features of GLUT (and FreeGLUT) can be divided into a number of groups, as illustrated by Figure 1. It is notable how these parts differ in dependency on OpenGL and on the underlying operating system. Geometry and text are entirely portable and only depend on OpenGL, user input and timers only depend on the OS, while context creation and redisplay depend on both.

Since geometry and text rendering rely heavily on deprecated functionality of OpenGL, they are not easily included. The geometry calls are useful and popular, and text rendering is very valuable, so their usefulness is beyond doubt. However, not only do the current implementation of these parts rely heavily on deprecated functionality, their parts of the API also need significant updates. This is totally different from the other parts, where the existing API is perfectly relevant and where only small traces of obsolete code can be found. Context creation, as comparison, has a few obsolete options, like indexed color, but that does not disturb new code.

For the geometry support calls, the problem is that they need to be changed in order to support shaders. This was not a problem in OpenGL 2.1, where shader variables for vertices, texture coordinates and normal vectors were pre-defined, but in modern OpenGL these are user defined. Concerning the implementation, the old GLUT approach to geometry (using glMap2f and glEvalMesh2) is obsolete and therefore a completely different implementation is needed.

Text rendering is a dual problem, since GLUT has two text rendering options, bitmap and stroke fonts. Bitmap fonts are outdated because their implementation rely on glBitmap, which is obsolete. Stroke fonts are somewhat more realistic to re-implement. Our biggest gripe about them is that GLUT provides so few options for font selection that the stroke fonts are next to useless and therefore a major overhaul is called for.

We decided that geometry and text solutions either must be rewritten or omitted. We chose to omit them for the time being. We will discuss geometry and text further in section 4.

With text and geometry skipped, we made our implementation, starting with the most vital calls according to our own example programs. In order to identify additional desirable inclusions/exclusions, we decided to study how GLUT was used in the past and let that influence the implementation. Although old demos do not usually reflect the needs of new code, the user interface issues are mostly the same. Therefore even old demos can yield interesting information in this case. For this purpose, we selected the official GLUT examples from SGI, the “Examples” folder (SGI 2007). This dataset consists of 37 demos, which is a well-defined set to work from. For all demos, we replaced GLUT with our current MicroGlut implementation (in compatibility mode) and studied the result.

We chose to measure the number of usages for each GLUT function call in these demos. The calls most frequently used were glutAddMenuEntry() (181 times) and glutPostRedisplay() (108 times). Some calls occur in all or almost every program (glutMainLoop(), glutInit(), glutInitDisplayMode(), glutCreateWindow(), glutDisplayFunc). The biggest surprise was glutTimerFunc() (4 times) and glutPassiveMotionFunc() (1 time) which we consider important but were rarely used in the demos. These calls were still chosen for inclusion.

Function calls for menus, keyboard and mouse were common (12 uses and up) and were generally included. Common window management calls, dealing with window title and size, were also included. Multi-window support, however, is a debatable case. The most common multi-window call, glutSetWindow() was called 17 times, but that was only because it occurs many times in each demo were it appears. We decided to put multi-window support on hold for now.

We had already decided to exclude geometry and text, but it is still interesting to get an indication for popularity for these too. In our test, these represented the most frequent calls that were excluded. For geometry, the most popular calls were glutSolidIcosahedron() (15 times) and glutSolidTorus() (12 times) while the famous Utah teapot (glutSolidTeapot()) was called only four times. (It is, however, clearly more popular in many other sources.) Text rendering is mostly done with glutStrokeCharacter() (14 times).

The only call used over 5 times that was not included glutExtensionSupported() (6 times). Although somewhat frequently called, the usage is limited compared to its considerable complexity so it was skipped.

The number of unique GLUT calls in the dataset was 61, which is about half of the API. Many of the functions that were excluded are rarely used in the dataset.

A number of additional functions of interest were found and implemented in addition to the earlier prototype. After adding these, 22 out of the 37 demos compile and run correctly with no major changes. Considering that most demos that do not work depend on text rendering or built-in geometry, we believe that that number is pretty high, and indicates that our subset agrees fairly well with the intended usage of GLUT.

3. FEATURES OF MICROGLUT

Since MicroGlut mainly deals with platform dependent problems, the code in MicroGlut is very platform dependent indeed, so implementations for various platforms were made using separate code. The current implementations include MacOS X, Linux and Microsoft Windows. More platforms are being considered. MicroGlut has been used in two advanced computer graphics courses and was sufficient for this use.

MicroGlut is, as the name implies, intended to be small, and comes as a single source-code file. This eliminates installation problems, and encourages making any changes desired. MicroGlut is Public Domain software, so if we should abandon it, there are no legal issues in redistributing changes.

It currently includes the features shown in Table 1, and is available through the author.

Table 1. MicroGlut features

GLUT features	Implementation extent	Extensions
Window/context creation	Partial	None
Keyboard input	Complete	Key map support
Mouse input	Complete	
Timers	Complete	Repeating timer
Menus	Partial	Menu bar support
Text	Omitted/external extension	Modified API
Geometry	Omitted/external extension	Modified API
Game mode	Omitted, planned for future version	

As the table implies, apart from geometry and text, GLUT functionality is reasonably complete, even including a few extensions over FreeGLUT. It is, however, not a complete set of utility code for modern OpenGL, so it is generally complemented with other utility modules (shader loader, texture loader, math etc).

4. MISSING FEATURES AND HOW TO RE-IMPLEMENT THEM

We expect long-time GLUT users to be concerned with the omission of text and geometry. What is GLUT without the Utah Teapot? Despite their popularity, we have omitted geometry and text. The reasons were discussed in section 2. In this section, we discuss how they can be re-implemented.

We have created a prototype for a replacement to GLUT’s geometry calls, in which we implemented the Utah teapot in the OpenGL 3.2 core profile, with a modified API. A screen shot of the prototype is shown in Figure 2. As a curiosity, you may notice that the incorrect rendering in the teapot knob, as of the current GLUT implementation, has been fixed, as well as the missing bottom of the cup and plate.

Taking the teapot as example, the call for rendering it with GLUT needs a modified parameter list:

void glutSolidTeapot(GLdouble size, GLint vertex, GLint normal, GLint texCoord);

This adds three parameters from the original, where only the size was given. The parameters vertex, normal and texCoord are shader variable numbers as returned by glGetAttribLocation(). Other geometry calls could be re-implemented with a similar API change.

In our prototype, at the first call to glutSolidTeapot(), the teapot model is tesselated by the CPU and the result is uploaded to the GPU and referenced through a VAO. The approach works but initialization takes noticeable time. To improve performance, the tesselation should be performed on the GPU, and it could be desirable to tesselate the geometry in real time using tesselation shaders, but such a solution is a rather delicate balance between hardware demands and performance and therefore more work is needed to find the best solution.

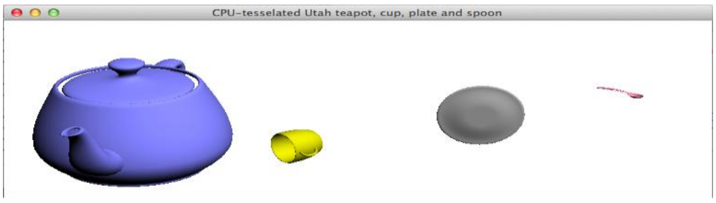


Figure 2. The Utah teapot is not part of MicroGlut, but a re-implementation in the OpenGL 3.2 core profile has been created (including the rest of the Utah set), either for future inclusion or as a separate module.

Text rendering is a different matter. Text rendering is a surprisingly complex issue, where solutions often are platform dependent, complex, or both. GLUTs font support is limited and based on deprecated functionality. Despite this, GLUT's old text functions are still referred to as "methods for displaying characters in the OpenGL package" (Hearn et. al., 2011).

A popular text rendering method is texture fonts (McReynolds and Blythe 2005, Ragnemalm 2013), which are more flexible and have good performance. We have created texture font based solutions as prototypes for replacements of GLUTs fonts and find this perfectly viable both for creating a simple plug-in solution or a more powerful approach.

Thus, we have drafted solutions for both geometry and text rendering. However, we consider it questionable if they should be an integral part of GLUT, or if it would be more suitable to put in another package and make GLUT focused entirely on platform dependent issues.

5. CONCLUSIONS

We suggest that a proper overhaul of the GLUT API would be the most beneficial path for creating an up-to-date API for portable user interfaces for OpenGL, in contrast to solutions that ignore the established API in favor of incompatible ones. As a step toward such a new GLUT version, we have developed a subset of GLUT called MicroGlut focused on the needs of modern OpenGL code. The most significant omissions in the subset were text rendering and built-in geometry, parts in need of a significant redesign. We have prototyped replacements for these parts, but the actual API for such replacements remains an open question.

REFERENCES

Di Benedetto, M., Ponchio, F., Ganovelli, F., Scopigno, R., "SpiderGL: a JavaScript 3D graphics library for next-generation WWW", 2010, Proceedings of the 15th International Conference on Web 3D Technology, pp 165-174.

FreeGLUT, 2012, http://freeglut.sourceforge.net

Hearn, D.,Baker, M.P., Carithers, W.R, Computer Graphics with OpenGL, 2011, Pearson, Boston.

Khronos Group, 2013, GLUT - The OpenGL Utility Toolkit, http://www.opengl.org/resources/libraries/glut/

Mark J. Kilgard, The OpenGL Utility Toolkit (GLUT) Programming Interface API Version 3, 1996, http://www.opengl.org/resources/libraries/glut/spec3/spec3.html

McReynolds, T., Blythe, D., Advanced Graphics Programming Using OpenGL, 2005, Morgan Kaufmann, San Fransicso

Schreiner, D., OpenGL Programming Guide, Seventh edition, 2009, Addison Wesley, Boston.

SGI, 2007, GLUT Examples, http://www.sgi.com/products/software/opengl/examples/glut/examples/

Stack Exchange, 2012, http://gamedev.stackexchange.com/questions/23652/what-alternatives-to-glut-exist

Ragnemalm, I., Polygons Feel No Pain, 2013, Linköping University.

Posters

DYNAMIC MODELING AND SIMULATION OF MOVING MARINE PLANT POPULATION

Jun Ogawa, Masahito Yamamoto and Masashi Furukawa
Hokkaido University, Japan

ABSTRACT

Cultivation of marine plants is remarkable in the field of energy development. Physical simulation on computer is convenient for the analysis of the motion of marine plant populations. This is a practical approach in terms of cost in order to design the optimal cultivation system, device and environment. Our study aims to elucidate physics phenomenon in a large cultivation system of marine plants and provide new engineering insights into cultural techniques. This paper describes how to model the physical motion of marine plant populations which is cultivated under the high volume fraction. It is possible to consider the physical interference by representing the plant with rigid bodies and using physics engine. Our proposed modeling method keeps filling plant models while avoiding overlaps between the models in an interior of three-dimensional mesh structure object in order to reduce the error of physics calculation when handling many rigid bodies. A fluid environment is constructed by Lattice Boltzmann method in the virtual space. The dynamics of marine plant is approximated by applying buoyancy and drag force. Results show that the plant populations moves with fluid velocity while causing the adhesion between individuals.

KEYWORDS

Marine plant population, physics modeling, motion simulation, initial placement, fluid analysis

1. INTRODUCTION

Seaweed is a better material for solving industrial challenges such as carbon dioxide capture and sewage treatment and so on [Demirbas, 2009] [Kheshgi, 2000] [Matsumoto, 2000]. Therefore, the cultivation of marine plants will confer many benefits to our daily life. Many studies about the marine plant cultivation have focused on the analysis of biological factors such as the amount of absorption of nutrients or light and so on. Physical and mechanical factors are also significant at the cultivation of marine plant. For instance, twist of marine plants is complex physical phenomenon which is caused by physical interference between individuals of plant population in water. This is one of factors which inhibit the growth of marine plants. Nevertheless, the study on solving physical challenges is not undertaken. Our study conducts the analysis of physical motion of marine plant based on physical simulation in order to figure out how to avoid the twining of marine plants. Physical simulation enables us to quantify physical state of marine plant populations without using real one. However, there are some challenges to realize the simulation for moving marine plant populations. One is the motion acquirement as aquatic plant. In order to simulate the aquatic plant motion precisely, it is required to calculate fluid force affecting to objects in the simulation environment. The computational costs for calculating the effects of the fluid forces to all objects are very large and thus the simulation time is increased. Therefore, the physical simulation should be executed using a method that has low calculation cost. The another one is the construction of simulation model that can prevent the slipping of plant models on the plant models collide each other. The physical interference between individuals should not be ignored if we analyze the motion of plant populations. This reason is that an individual affects into entire motion of plant populations through twist phenomenon. In this paper, we describe the dynamic modeling and simulation that a lot of plant model, which have complex morphology, are placed on the interior of 3-dimensional object. In order to realize the motion of marine plant populations on high volume fraction, we show the mechanism of determining plant models morphology as to do not always overlap plant models at the initial state. Then, the state of marine plant populations is quantified by the physical simulation.

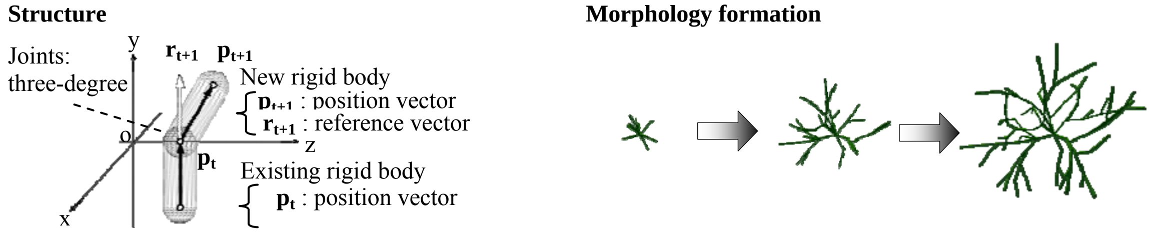
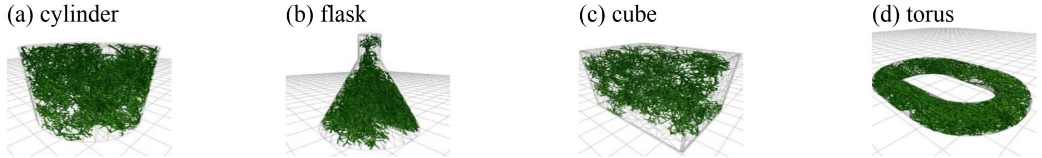


Figure 1. A image of constructing plant model


(a) cylinder

(b) flask

(c) cube

(d) torus

Figure 2. A example of tank model that has a triangle mesh structure and solid model of tank and its space

From the result, it verifies that our modeling and simulation method is efficient as the motion analysis of marine plant population.

2. PHYSICS MODELING

The physical model should be imitated a real marine plant populations. However, in the light of an actual plant, modeling of the plant morphology with physical motion is a hard work for the operator. Therefore, the modeling is necessary that it narrows down the components, which need as real marine plants, considering the trade-off between imitation and efficiency. This chapter explains about how to model marine plant populations having complex morphology and motion in fluid.

There is a technique of stochastic model as representing the growth process and morphology of plants [Kang, 2008]. This model stochastically determines bending-extension of the foliage and branches for representing the growth process. Structure update is conducted by adding previous structure information. Therefore, the morphology formation which is easy for humans to understand intuitively is performed by the stochastic model. The physical model consists of primitive of rigid body based on the structure obtained by using a probability model. In this study, we introduce the following four stochastic components:

- Primitive of rigid body: Sphere-swept volume (SSV) is adopted as a primitive of rigid body of virtual plant. Collision detection between the SSV can be calculated at high performance by figuring out the shortest distance between two line segments.
- Degree of freedom of joint: A joint between rigid bodies is set into 3 degrees of freedom. Flexibility of marine plants is represented by setting the degree of freedom in the direction of the swing direction and twist.
- Posture: Branch, which is represented at a rigid body, of the posture is represented by a position vector and rotation angle. The rotation angle denotes the direction in which extends branch, the angle is Euler angles. Rigid body has a reference vector that is the direction of its rigid body.
- Number of Branches: Branching is occurred by providing the maximum number of branching. The maximum number of branching MAX is given by integer. The number of branching for each branch is determined in the range of [0: MAX] with uniform random numbers.
- Growth Level: Growth level is an integer value that indicates the growth. A new rigid body is added into the tip of rigid body which is already placed at the previous at each time that a growth level is increased by one. Rigid body information to be added is stochastically determined as meet all of conditions of "degree of freedom" and "posture" and "branches"

We can create a growing plant population automatically and dynamically by introducing these components on the modeling process. These concepts are shown in figure 1.

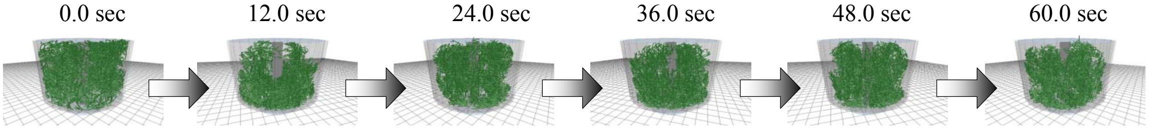


Figure 3. The result of motion transition of marine plant populations based on physical simulation

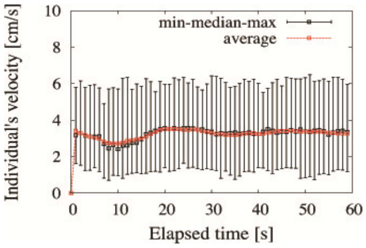


Figure 4. The distribution of individual velocities

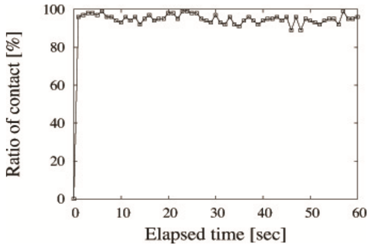


Figure 5. Time variation of proportion of adhesion clusters

In this study, physics engine (NVIDIA PhysX) is employed for simulating the motion of marine plant in fluid. A constructing of virtual fluid environment method, which has low computational costs based on the physics engine, has been proposed in the previous literature [Nakamura 2011]. Authors have proposed how to acquire the dynamics of marine plants by applying this research [Ogawa, 2013]. In the following, we show the outline of the fluid environment model for acquiring aquatic motion. In the fluid environment, the vertical upward force that is called buoyancy F_B works to objects by Archimedes' principle. The force is defined by the following equation (1):

$$F_B = \rho V g \tag{1}$$

where ρ is a fluid density, V is a volume of rigid body, and g is gravity acceleration. Drag force is also fluid force that is proportional to the square of relative velocity with a fluid. Its strength is given by equation (2):

$$F_D = \frac{1}{2}\rho A C_D v^2 \tag{2}$$

where A is a projected area, C_D is the substance specific drag coefficient, and v is relative velocity with fluid.

In order to develop realistic water flow, there is also Lattice Boltzmann method (LBM) that is one of the analysis techniques for computational fluid dynamics [Chen, 2003]. The LBM numerically simulates the fluid motion by calculating time evolution of particle velocity distribution based on the idea of cellular automata. This method can represent the high-precision flow without noise. Equation (3) and (4) are employed for calculating time evolution of particle distribution in LBM.

$$f_i(x+e_i\Delta t,\ t+\Delta t) = \frac{\lambda-1}{\lambda} f_i(x,t) + \frac{1}{\lambda} f_i^{eq}(x,t) \tag{3}$$

Where λ is the relaxation frequency that is the unique value of each fluid, fi is the particle distribution function, i is kind of particle velocity. These particles move repeating a collision. The following equation (4) represents the particle equilibrium distribution.

$$f_i^{eq}(x,t) = \omega_i \rho\left(1 - \frac{3}{2}u + 3(e_i \cdot u) + \frac{9}{2}(e_i \cdot u)^2\right) \tag{4}$$

Where f_i^{eq} is local equilibrium distribution, ωi is weight coefficient, $\boldsymbol{u}$ is fluid velocity, $\boldsymbol{ei}$ is particle velocity.

The plant populations' model is placed within a tank model of 3D object. When the number of plant individuals on simulation is large, their volume fractions become high. In the case of physics modeling, it is hard to explore vacant space since the morphology of each plant model is complex. If the physical model is allowed to overlap between the models, a fatal error occurs during the simulation run. Thus, the placement problems of individual physical model must be resolved in order to avoid the computational instability by false collision detection at the initial state. In the modeling process, the placement problem is classified to initial position determination and addition position determination of new rigid body. First step in initial position determination is to calculate the normal vector of all surfaces of 3D object (tank) from the mesh data. Second step determines a vertex P at a random number, and calculate a vertex Q which is the shortest distance with P in the vertex set of mesh. Next step is to determine the inner product between vector PQ and normal vector of all surfaces including Q. If the product of inner products is positive, P is the initial position of model. The addition position determination of new rigid body considers the collision with other rigid bodies. The rotation angle and the position of an adding rigid body are determined by random numbers. Then, the tip of the rigid body position is detected whether the internal of a 3D object. Since the rigid body is SSVs, it collides when the shortest distance between the line segments is less than the diameter of a sphere. Therefore, a new rigid body's position is defined by the tip of the rigid body and the distance of segment of rigid bodies. The results in figure 2 illustrate plant populations' model on the high volume fraction within several tank model.

3. NUMERICAL SIMULATION

In order to verify the motion of marine plant populations, we execute the numerical simulation in an environment which generates water flow. The objective of this simulation is the realization of motion of marine plant populations and the analysis of the motion for controlling physical twist of plants. In this simulation, the total number of plant models is 100. The tank model is shown in figure 2(a); this is a model which combines circular cone and cylinder. Water flow occurs from bottom to top in the tank. This flow is a stirring flow that circulates between center and verge of the tank. The total simulation time is 60 seconds.

The simulation results are shown in figure 3-5. Figure 3 illustrates the result of the motion transition of the plant populations obtained by the simulation. Figure 4 shows a diagram to box plot median, minimum, maximum and average value of individual velocities. Then, figure 5 is the result of time variation of its percentage. An adhesion cluster is defined as the number of chunks of the plant models which are connected by the contact. From figure 3 is confirmed that computer is possible to calculate the physical movement consistently stable without occurring fatal error. At least, the adhesion between individuals on initial state not happens from the result of figure 5. Therefore, the error of collision calculation can be avoided on initial state. In this simulation, the plant populations' model forms a huge mass that are contacted with each other many plant model. More than 90% of individuals have never separated from the formed cluster in the subsequent movement. Thus, this situation is that the plant populations keep the state of firmly intertwined. In addition, individual velocity is distributed almost uniform through all simulation time. This shows that plant populations are insulated from the effect of water flow due to the twist between individuals.

4. CONCLUSION

In this study, we have established a method of dynamic modeling and simulation of marine plant populations on high volume fraction in 3D object. As the outlook for future works on cultivation of marine plants, we show that our method is useful in engineering to analyze the physical state of marine plant population and the motion by the numerical analysis based on simulation.

REFERENCES

Demirbas, M.F. et al, 2009, Potential contribution of biomass to the sustainable energy development, *Energy Conversion and Management*, Vol.50, No.7, pp.1746-1760

Kheshgi, H.S. et al, 2000, The Potential of Biomass Fuels in The Context of Global Climate Change: Focus on Transportation Fuels 1, *Annual review of energy and the environment*, Vol.25, No.1, pp.199-244

Matsumoto, M. et al, 2000, Floating cultivation of marine cyanobacteria using coal fly ash, *Applied biochemistry and biotechnology*, Vol.84, No.1, pp.51-57

Kang MZ. et al, 2008, Analytical study of a stochastic plant growth model: Application to the Greenlab model, *Mathematics and Computers in Simulation*, Vol.78, No.1, pp.57-75

Nakamura K. et al, 2011, Acquisition of swimming behavior on artificial creature in virtual water environment, *Advances in Artificial Life. Darwin Meets von Neumann*, pp99-106

Ogawa J. et al, 2013, Control of water flow to avoid twining of artificial seaweed, *Journal of Artificial Life and Robotics*, Vol.17, No.3-4 , pp.383-387

Chen, S. wt al, 2003, Lattice Boltzmann method for fluid flows, *Annual review of fluid mechanics*, Vol.30, No.1, pp.329

EFFECTS OF SPACING BETWEEN ITEMS AND VIEW DIRECTION ON ERRORS IN THE PERCEIVED HEIGHT OF A ROTATED 3-D FIGURE

Kuo-Chen Huang
Product Design of Department, Ming Chuan University, Taiwan ROC

ABSTRACT

In this present study, the effects of spacing between items, view direction, and forward-rotated angle on errors in the perceived height of a rotated three-dimensional figure were investigated. Our results indicate that the spacing between items had a statistically significant effect on errors in perceived height, and that fewer errors in perceived height were made when judgments were based on a bottom-up view than based on a top-down view. Errors in perceived height were significantly smaller in response to a 15° forward-rotated angle compared with a 30 and 45° forward-rotated angle. These results have implications for graphics-based interface design, such as that used in interior design, driver navigation systems, and geological models.

KEYWORDS

Height perception, three-dimension figure, view direction

1. INTRODUCTION

Previous results have shown that humans perceive horizontal distances roughly accurately but overestimate vertical distances (Jackson & Cormack, 2008). Vertical distances are overestimated given the “looking down” condition, because the observer is not viewing the distance from the ground plane and thus lacks the proper cues for estimating distance (Stefanucci & Proffitt, 2009). Jackson and Cormack (2010) suggested that heights are overestimated more frequently from a top-down than from a bottom-up view due to risks related to falling; overestimation was minimized when the falling risks were removed.

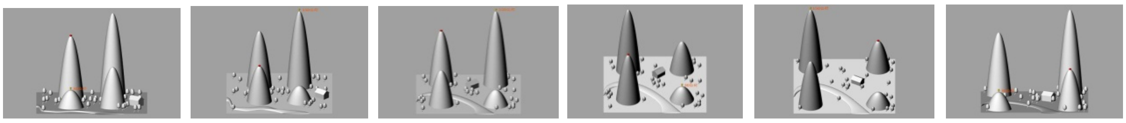
Stoper (1990) found that participants systematically underestimated the height of stimulus triangles in the upward direction and overestimated it in the downward direction. Similarly, Gottesman (2011) recently argued that when the view angle changes significantly, the cues of the position and orientation of stimulus elements are no longer useful in distance judgments; however, these cues remain useful after small changes. Westheimer and Hauske (1975) showed that vernier offset discrimination deteriorates in the presence of adjacent flanking lines; however, the effect of this interference decreases as the spacing between the flankers and the vernier increases. Huang (2009) asked participants to move the comparison stimulus to match the distance to the standard using various spacings between the comparison and standard; they found that the error increased with increased spacing.

The present research was designed to investigate possible differences in errors of perceived height when judging the height of a target against that of a standard as they relate to these three independent variables, including view direction, the forward rotated angle of the stimulus, and the spacing between items.

2. METHODS

Eighty college students were recruited as participants. All participants reported 20/20 corrected visual acuity or better.

The present experiment investigated three independent variables: spacing between items (2, 4, and 6 cm), forward-rotated angle (15, 30, and 45°), and view direction (bottom-up and top-down). These three variables were within-subject variables. The stimuli were produced by a visual simulator designed with the Rhinoceros 4.0 and Adobe Flash CS4 and were displayed in the center of a View Sonic 19-in color monitor by a Pentium-M PC equipped with Microsoft software. Some examples of these items are shown in Figure 1. The points used as the standard and comparison were located on the tops of two items that were randomly chosen from among the four items.



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

Figure 1. Stimuli used in the present study. Attributes of each stimulus were (a) (2 cm, 15°, bottom-up), (b) (2 cm, 30°, top-down), (c) (4 cm, 30°, top-down), (d) (4 cm, 45°, bottom-up), (e) (6 cm, 45°, top-down), and (f) (6 cm, 15°, bottom-up).

The exposure time for each stimulus was 5 s. Participants were asked to estimate the height of the comparison with respect to the height of the standard for their location. They were asked to respond within 5 second after the stimulus disappeared.

Errors were calculated as a ratio of the difference between the actual and estimated height to the actual height, measured as a percentage (%).These data were recorded and entered into an analysis of variance (ANOVA) using SPSS software.

3. RESULTS

A significant effect of view direction on error in perceived height ($F_{1,79}$=675.52, p<.001), indicating the error in perceived height for view direction of bottom-up (*M*=21.1, *SE*=1.1) was less than that for top-down (*M*=53.0, *SE*=0.5). The main effect of spacing between items on error in perceived height was significant ($F_{2,158}$=114.64, p<.001). Multiple comparisons using LSD method showed that the error in perceived height of 2-cm (*M*=29.5, *SE*=0.9) spacing between items was significantly smaller than those for 4- (*M*=40.7, *SE*=0.7) and 6-cm (*M*=40.9, *SE*=0.8) conditions. However, there was no difference between 4- and 6-cm conditions. Analyses identified a main effect for forward-rotated angle on error in perceived height ($F_{2,158}$=38.67, p<.001). Multiple comparisons showed that the error in perceived height of 15 degrees (*M*=28.8, *SE*=1.2) was significantly smaller than those for 30 (*M*=40.2, *SE*=0.8) and 45 (*M*=42.0, *SE*=1.4) degrees; however, there was no difference between the latter two conditions.

These findings also reflect three two-way interactive effects: between item spacing and forward-rotated angle, between view direction and item spacing, and between forward-rotated angle and view direction.

4. DISCUSSION

4.1 Spacing between Items

The data indicate that the spacing between items significantly affected errors in perceived height. As expected, the greater the spacing between items, the greater the error in perceived height. This result is similar to the results reported by Roumes, et al. (2001) and Huang (2009). One possibility for this finding is that increased spacing between items makes it more difficult to compare the height of the standard with that of the stimuli; this may be attributable to a change in the relationship between the depth cues and the visual angle, which serve as informational resources for height estimates.

4.2 Forward-Rotated Angle

Our results indicate that the greater the forward rotation of the stimulus, the greater the error in the perceived height. One possibility is that participants may judge the height of the target only with respect to the horizontal reference plane, resulting in a reduced error in perceived height when the forward-rotated stimulus angle decreases. This interpretation of errors in perceived height was also supported by Ozkan and Braunstein's (2010) basic argument that the cue of height in the visual field may be based on the presence of an implicit or explicit horizon.

4.3 View Direction

View direction had a significant effect on errors in perceived height such that the error in the perceived height from the bottom-up view was less than that from the top-down view. This result is consistent with earlier findings showing that heights are overestimated more frequently from the top than from the bottom (Stefanucci & Proffitt, 2009; Jackson & Cormack, 2008). One possibility is that the top-down view, unlike that from a true ground surface, does not encompass the optical variables that often provide depth cues. Another possibility is that participants with a higher trait-level fear of heights show increased overestimation of heights from the top-down view in both normative situations and imagery tasks (Teachman, et al., 2008).

5. CONCLUSION

Overall, our results show that errors in perceived height in a virtual environment are affected by the spacing between items, the forward-rotated angle, and the view direction. These findings are important as height perception is required for most visual systems (e.g., graphics-based interfaces). Those physical characteristics of 3-D images that affect shape and distance perception warrant additional examination in future research.

ACKNOWLEDGEMENT

The author thank the National Science Council of Taiwan for financially supporting this research under contract No. NSC 99-2221-E-130-009-MY2, and Shu-Ming Hsu for his help in conducting the experiment.

REFERENCES

Gottesman, C. V. (2011) Mental layout extrapolations prime spatial processing of scenes. *Journal of Experimental Psychology: Human Perception and Performance*, 37, 382-395.

Huang, K. C. (2009) Effects of colored lights, spacing between stimuli, and viewing distance on error in a depth-matching task. *Perceptual and Motor Skills,* 108, 636-642.

Jackson, R. E., & Cormack, L. K. (2008) Evolved navigation theory and the environmental vertical illusion. *Evolution and Human Behavior*, 29(5), 299-304.

Jackson, R. E., & Cormack, L. K. (2010) Reducing the presence of navigation risk eliminates strong environmental illusions. *Journal of Vision*, 10(5), 1-8.

Ozkan, K., & Braunstein, M. L. (2010) Background surface and horizon effects in the perception of relative size and distance. *Visual Cognition*, 18(2), 229-254.

Roumes, C., Meehan, J. W., Plantier, J., Menu, J. P. (2001) Distance estimation in a 3-D imaging display. *The International Journal of Aviation Psychology*, 11(4), 381-396.

Stefanucci, J. K., & Proffitt, D. R. (2009) The roles of altitude and fear in the perception of height. *Journal of Experimental Psychology: Human Perception and Performanc*e, 35(2), 424-438.

Stoper, A. E. (1990) *Pitched environments and apparent height.* Paper presented at the meeting of the Association for Research in Vision and Ophthalmology, Sarasota, FL.

Teachman, B. A., Stefanucci, J. K., Clerkin, E. M., Cody, M. W., & Proffitt, D. R. (2008) A new mode of fear expression: Perceptual bias in height fear. *Emotion, 8*, 296-301.

Westheimer, G., & Hauske, G. (1975) Temporal and spatial interference with vernier acuity. *Vision Research,* 15, 1137–1141.

ISBN: 978-972-8939-89-2 © 2013 IADIS

MEASURING QUALITY OF SEGMENTATIONS

Stepan Srubar
DoCS FEECS VSB-TUO - 17. listopadu 15, 708 33Ostrava-Poruba

ABSTRACT

Segmentation quality is measured by a look of a human or by evaluation methods. Many of them were proposed. Each of them have some typical problems which influences the quality. If we use these algorithms for choosing and setting of a segmentation algorithm for specific task, it is crucial to use high quality evaluation method. This article proposes method how to measure quality and shows the result of a large measurement.

KEYWORDS

Segmentation, Evaluation.

1. INTRODUCTION

Segmentation is important part of image processing. It separates objects in image using segments. Each segment can be measured for some specific properties like perimeter, area, curvature or it can process or modify image information in each segment separately. For perfect results, we need high quality segmentation which is often created by a segmentation algorithm. Therefore, quality of the algorithm should be measured by an evaluation method. Quality of the method is also crucial and its measurement is also possible. This paper compares quality of current algorithms for segmentation evaluation.

2. METHODS AND METHODOLOGY

Evaluation methods can compare two segmentations or segmentation to image. We are now interested in the first group of methods. Due to page limitation only list of methods will be provided. Well-defined methods were used as proposed by their authors: Symmetric Divergence (SYD) (Pal & Bhandari 1993), Global (GCE), Local (LCE) and Bidirectional Consistency Error (BCE) (Martin et al. 2001, 2002), Global Bidirectional Consistency Error (GBCE), normalized Hamming Distance (HD) (Huang et al 1995), Partition Distance (PD) (Cardoso & Corte-Real 2005), L (Larsen 1999), VD (Van Dongen 2000), MH (Meila & Heckerman 2001), YD (Yasnoff et al. 1977), F (Strasters & Gerbrands 1991), MC (Monteiro & Campilho 2006), H2u (Huang & Dom 1995), JC (Ben-Hur 2002), FM (Fowlkes & Mallows 1983), W (Wallace 1983), M (Mirkin 1996), PRI (Rand 1971), (Unnikrishnan 2005, 2007), Object Count Agreement (OCA) (Yasnoff & Bacus 1984), Normalized Mutual Information (NMI) (Strehl 2000), Variation of Information (VI) (Meila 2003), maximum-weight bipartite graph (BGM) (Jiang et al. 2006), Fragmentation (FRAG) (Strasters 1991). Following three methods were not defined for whole segmentation, therefore, some straightforward extension was applied: SM1, SM2 (Yasnoff et al. 1977), SFOM (Pratt 1978), Censored Hausdorff Distance (SCHD) (Paumard 1997).

All methods were implemented and practically evaluated on image data sets. They consist of sample images and their ground truth segmentations. Comparison of two segmentations results in a number. In each comparison we know if these segmentations belong to the same image or not. According to that results should split into two clusters. Typically, result of segmentations belonging to the same image are low while results from segmentations from different images are high. Therefore, we could find optimal threshold for the set of results to separate them into the two categories. Rate of results on the wrong side of the threshold will denote the rate of error for the current method. In fact there are two types of error - false acceptance (FA) and false rejections (FR). Classification depends on whether the false result is lower or higher than the threshold. Each method will have different value of the threshold but the final error rate will be the lowest possible.

Implemented methods were tested on image data set which consists of images and their ground truth segmentations. This set is created from real pictures of people, animals and landscapes. It was presented in (Martin 2001). Whole database consists of images and each image has four ground truth segmentations at least. We could compare two or more different segmentations, which belong to a single image. Images in the database are longitudinal and perpendicular. Still, they have the same resolution if they are appropriately rotated.

3. RESULTS

For final comparison, we take the threshold with the minimum sum of error rates. Total error cannot exceed 100%. Still, methods does not exceed 50% with any threshold, typically. Minimal error rate for arbitrary method can be 50% at most.

Number of evaluations for each method depends on symmetry of the method. Asymmetric methods evaluate each couple of segmentations twice but symmetric methods just once because they would give the same result. Symmetric methods evaluated 3138496 couples of segmentations, while each asymmetric method processed 6276992 couples of segmentations. This task is easily parallelizable, still, running of a single method could take tens of hours on common 4-core CPU.

Results of all implemented segmentation-segmentation methods are presented in figure 1. All methods reaching 50% are practically unusable. The lowest error was made by methods VD and HD. Since VD and HD has the same basis of the formula, the results are equal. However, the method VD is a metric. Another metric with low error rate is VI.

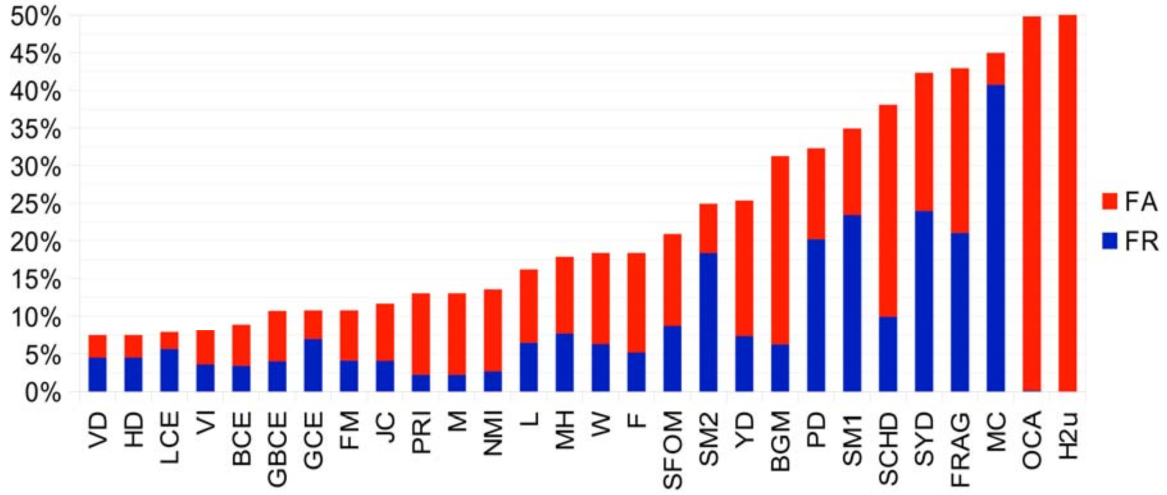


Figure 1. Results of segmentation-segmentation methods for Berkeley data set

4. CONCLUSION

Segmentation evaluation methods has very various quality. Some of them are focused on a small part of information from the whole segmentation and the quality is, therefore, poor. Even some recently proposed methods do not assure high quality. The best results were provided by method VD. This quality measurement is one of the biggest according to the size of the test set as well as the number of methods which can ensure high level of objectivity.

REFERENCES

Ben-Hur, A., Elissee, A., Guyon, I., 2002. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing*, pp. 6-17.

dos Santos Cardoso, J., Corte-Real, L., 2005. Toward a generic evaluation of image segmentation. *Image Processing*, IEEE Transactions on 14, pp. 1773-1782.

Fowlkes, E.B., Mallows, C.L., 1983. A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association 78* , pp. 553-569.

Huang, Q., Dom, B., 1995. Quantitative methods of evaluating image segmentation. *Image Processing*. Proceedings., International Conference on. Volume 3, pp. 53-56.

Jiang, X., Marti, C., Irniger, C., Bunke, H.,2006. Distance measures for image segmentation evaluation. *EURASIP J. Appl. Signal Process.*, pp. 209-209.

Larsen, B., Aone, C., 1999. Fast and efective text mining using linear-time document clustering. *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '99, New York, NY, USA, ACM, pp. 16-22.

Lee, S.U., Chung, S.Y., Park, R.H., 1990. Performance study of several global thresholding techniques for segmentation. *Comput. Vision Graph. Image Process. 52*, pp. 171-190.

Lim, Y.W., Lee, S.U., 1990. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition 23*, pp. 935-952.

Martin, D.R., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *Technical Report UCB/CSD-01-1133*. EECS Department, University of California, Berkeley.

Martin, D.R., 2002. *An empirical approach to grouping and segmentation*. Phd dissertation, University of California, Berkeley

Meila, M., Heckerman, D., 2001. An experimental comparison of model-based clustering methods. *Mach. Learn. 42, pp.* 9-29.

Meila, M. 2003. Comparing Clusterings by the Variation of Information. *Learning Theory and Kernel Machines*, pp. 173-187.

Mirkin, B.G., 1996. *Mathematical Classication and Clustering*. Kluwer Academic Publishers, Dordrecht.

Monteiro, F., Campilho, A., 2006. Performance evaluation of image segmentation. *Image Analysis and Recognition*. Volume 4141 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 248-259.

Pal, N.R., Bhandari, D., 1993. Image thresholding: some new techniques. *Signal Process. 33* pp. 139-158.

Paumard, J., 1997. Robust comparison of binary images. *Pattern Recogn. Lett. 18*, pp. 1057-1063.

Pratt, W.K., 1978. *Digital Image Processing*. John Wiley & Sons, Inc., New York, NY, USA.

Rand, W.M., 1971. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association 66*, pp. 846-850.

Strasters, K.C., Gerbrands, J.J., 1991. Three-dimensional image segmentation using a split, merge and group approach. *Pattern Recogn. Lett. 12*, pp. 307-325.

Strehl, A., Ghosh, J., Mooney, R., 2000. Impact of Similarity Measures on Web-page Clustering. Proceedings of the 17th *National Conference on Articial Intelligence: Workshop of Articial Intelligence for Web Search* (AAAI 2000), Austin, Texas, USA, AAAI (2000), pp. 58-64.

Unnikrishnan, R., Pantofaru, C., Hebert, M., 2005. A measure for objective evaluation of image segmentation algorithms. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05) - Workshops - Volume 03, Washington, DC, USA, IEEE Computer Society.

Unnikrishnan, R., Pantofaru, C., Hebert, M., 2007. Toward Objective Evaluation of Image Segmentation Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, pp. 929-944.

Van Dongen, S., 2000. *Performance criteria for graph clustering and markov cluster experiments*. Technical report, National Research Institute for Mathematics and Computer Science, Amsterdam, Netherlands.

Wallace, D.L., 1983. A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association 78*, pp. 569-576.

Yasnoff, W.A., Bacus, J.W., 1984. Scene segmentation algorithm development using error measures. *AOCH 9*, pp. 45-58.

Yasnoff, W.A., Mui, J.K., Bacus, J.W., 1977. Error measures for scene segmentation. *Pattern Recognition 9*. pp. 217-231.

APPLICATION OF BOUNDARY POLYGON MESH FOR VOXEL-BASED COMPUTATIONAL HUMAN MODELS

Tomoaki Nagaoka
National Institute of Information and Communications Technology - 4-2-1, Nukuikitamachi,Koganei,Tokyo, 184-8795, Japan

ABSTRACT

Voxel-based computational human models with anatomical structures have frequently been used in various research fields. However, deforming human voxel-models as real world human is difficult task. We propose a body-surface aware deformation method and data representation. The new method and data representation enables various sophisticated deformation techniques to the human voxel-based model deformation task.

KEYWORDS

Voxel, Polygon mesh, Volumetric Mapped Boundary, Computational human model.

1. INTRODUCTION

Recently, high-resolution voxel-based computational human models with anatomical structures have frequently been used in studies on ionizing and non-ionizing radiation protection and medical applications (Xu and Eckerman, 2009). Since those models developed on the basis of the medical imaging data (i.e., X-ray CT or MRI) were generally upright configurations, some techniques on pose change for the voxel-based models have previously been proposed (Nagaoka and Watanabe, 2008; Faraj *et al.*, 2012). In three-dimensional computer graphics (3D-CG) field, various methods and techniques to deform the human body shape are proposed and the tools and the applications implement these functionality. Unfortunately, the 3D-CG mainly focuses on surface data such as polygon meshes and not on voxel data. Therefore, the CG techniques are not directly available to the voxel data. To bridge this gap between the voxel data and the CG techniques, we propose a novel representation of the voxel data, named Volumetric Mapped Boundary polygon mesh (VMB), which is compatible with the CG techniques and tools.

In this study, we present VMB and conversion algorithm of a voxel-based model between a VMB model, and demonstrate the deformation of real voxel-based data using our algorithm.

2. VOLUMETRIC MAPPED BOUNDARY POLYGON MESH (VMB)

2.1 Definition, Algorithm and Identical Recovery

VMB is defined as a triangular mesh that can be converted to the original voxel model. For our voxel model deformation procedure, the voxel model is converted to VMB, then the VMB is deformed using mesh deformation techniques and tools, and finally the deformed VMB is reconverted to the deformed voxel model.

In order to relate the voxel model to the geometric object, we regard a voxel as a cubic object whose edges have unit length. For the voxel with indices (i, j, k), we set position coordinates of eight vertexes of the cube be (i, j, k), (i, j+1, k), (i, j, k+1), (i, j+1, k+1), (i+1, j, k), (i+1, j+1, k), (i+1, j, k+1), (i+1, j+1, k+1). Under these setting, the definition of VMB follows:

1) VMB is a triangular mesh that specifies the sub region of the volume data. Each triangle of the mesh belongs to one of the voxel.

2) Each vertex of a triangular mesh has a texture coordinates which store the voxel index.

The algorithm of converting VMB from a voxel-based model follows:

1) Generates triangles of boundary voxel face

2) Associate the index of the voxel containing the triangles to vertexes of triangles as their texture coordinates (per-face texture coordinates)

The algorithm of recovering a voxel-based model that matches the original voxel-based model follows:

1) Voxelize the VMB and associates the texture coordinate to these voxels as sampling coordinates. Voxelizing algorithm follows:

For each voxel, the minimum distance between the center of voxel and the mesh is computed. If the distance is less that 0.5 (half of the voxel size), then the voxel is marked as the boundary. If the distance is equal to 0.5, them two voxels are candidates of the boundary. In this case, use the normal vector of the polygon is used to select the one. In addition to marking, the texture coordinate of the polygon (the unique per-face 3D coordinates) is associated to the voxel as the sampling coordinate to resample the original voxel-based model.

2) Interpolate sampling coordinates of the boundary voxels to get the sampling coordinate for the interior voxels as the element-wise solutions of the boundary problem. The unique solution exists for all coordinate function (Nagaoka and Watanabe, 2009).

An important property of VMB is that if VMB is not deformed then the recovered voxel-based model matches the source voxel-based model. The proof is done in two steps. The first step is to show the boundary voxels are identical. This is clear from the recovering algorithm step 1. The second step of the proof follows from the fact that the sampling coordinates that refers the voxel itself is one of the solution of the boundary problem because these sampling coordinates suffice the boundary condition and increase linearly at interior voxels. Therefore, these coordinate functions are the one of the solution. Therefore, resampled volume is identical to the volume data of the original voxel-based model.

2.2 Experimental Results

We tested our algorithm using real voxel-based human model with resolution of 2 mm (Nagaoka *et al.*, 2004) segmented into 51 different tissue types. The voxel-based model was converted to the VMB model. The VMB model was loaded to a consumer software, Autodesk Maya 2013 application program. Two types of deformation, mesh smoothing and skeleton based pose deformation, were applied to the VMB model.

Figure 1 shows VMB models (boundary polygon meshes); (a) non deformed model, (b) smoothed model and (c) a pose deformed model. These models were re-converted to voxel-based models with the algorithm described in the previous section and compared with the original voxel-based model by difference of voxel counts per-tissue bases. It should be noted first that the original and the re-converted voxel-based model of non-deformed VMB perfectly matched. For smoothed model case, the maximum ratio of change is 3.97% with skin. For pose-deformed model case, the maximum ratio of change is 4.63% with skin.

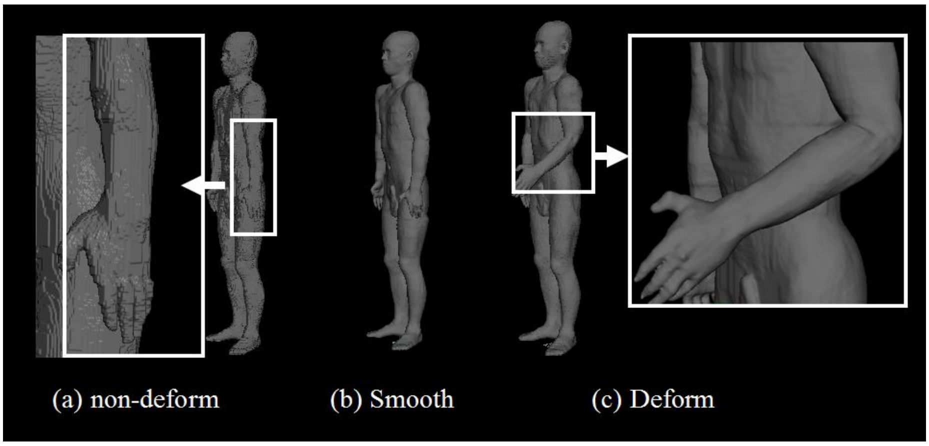


Figure 1. VMB models.

3. CONCLUSION

We proposed a novel object representation, Volumetric Mapped Boundary polygon mesh (VMB), for voxel-based models. It is shows that VMB models provide a bridge between voxel-based models and the surface model represented with polygon mesh and as a consequence various mesh editing techniques are applicable to the voxel-based model editing tasks.

ACKNOWLEDGEMENT

I am grateful to Mr. Makoto Matsumura of Adelic Research LCC. Parts of this work were supported by JSPS KAKENHI Grand-in-Aid for Young Scientists (B).

REFERENCES

Faraj, N. et al, 2012. VoxMorph: 3-scale freeform deformation of large voxel grids. *Proceedings of Computers & Graphics*. pp. 562-568.

Nagaoka, T. et al, 2004. Development of realistic high-resolution whole-body voxel models of Japanese adult males and females of average height and weight, and application of models to radio-frequency electromagnetic-field dosimetry. *Phys.Med. Biol.*, Vol. 49, No. 1, pp. 1-15.

Nagaoka, T. and Watanabe, S., 2008. Postured voxel based human models for electromagnetic dosimetry. *Phys. Med. Biol.*, Vol. 53, No. 24, pp. 7047-7061.

Nagaoka, T. and Watanabe, S., 2009.Voxel-based variable posture models of human anatomy, *Proceedings of the IEEE*, Vol. 97, No. 12, pp.2015-2025.

Xu, X.G and Eckerman, K.F., 2009. *Handbook of Anatomical Models for Radiation Dosimetry*. Taylor & Francis, New York, USA.

AUTHOR INDEX