Spring Conference on Computer Graphics SCCG 2017

in cooperation with

Computers & Graphics and ACM SIGGRAPH

Conference Materials and Posters

Mikulov, Czech Republic

May 15 - 17, 2017

Edited by

Pavel Zemčík

Vítězslav Beran

Conference Materials and Posters of
Spring Conference on Computer Graphics SCCG 2017

Conference chair:	Pavel Zemčík
Program chairs:	Ivan Viola, Jiří Bittner
Posters chair:	Vítězslav Beran
Organization chairs:	Vítězslav Beran, Andrej Ferko
Local organization team:	Michaela Studená, Elena Šikudová, Milan Ftáčnik
Reviewers:	See List of Reviewers
Cover Page:	Michal Ferko
Logo Design:	Jozef Martinka
Production Support:	Jaroslav Rozman

All papers were reviewed and selected by international program committee.

Publisher:	Brno University of Technology, Brno, Czech Republic
Webpage:	http://www.sccg.sk
ISSN:	1335-5694

Contents

Welcome to SCCG 2017

Hotel Galant, May 15 - May 17, 2017

Dear SCCG 2017 Participant,

welcome to Mikulov for the oldest annual computer graphics conference in Central Europe - Spring Conference on Computer Graphics (SCCG). These pages contain more details concerning its organization. Please, read it carefully. If anything remains unclear, do not hesitate to contact the organizers or refer to the SCCG homepage www.sccg.sk. In urgent cases, call Míša Studená (00 420 731 582 348) or Elena Šikudová (00 421 915 734 763).

Conference Highlights

The conference highlights will be, no doubts, the invited talks given by:

- **Niloy MITRA**, Virtual Environments and Computer Graphics, University College London
- **Kwan-Liu MA**, Center for Visualization, University of California at Davis
- **Daniel SÝKORA**, Department of Computer Graphics and Interaction, Czech Technical University in Prague

Conference Context and History

Spring Conference on Computer Graphics is organized by Brno University of Technology and Comenius University in cooperation with Slovak Society for Computer Science (SISp) and Mathematical Institute of the Slovak Academy of Sciences. The scientific programme and proceedings are produced in cooperation with Computers & Graphics, Eurographics and ACM/SIGGRAPH. The original name *Spring Conference* was coined by Prof. Werner PURGATHOFER from Technical University of Vienna and the tradition of the conference dates back to 1984. During the exciting 31 years, the conference was chaired by many bright minds from the whole world, such as:

Holly RUSHMEIER, USA
Carol O'SULLIVAN, Ireland
Tomoyuki NISHITA, Tokyo
Helwig HAUSER, Bergen
Reinhard KLEIN, Bonn
Karol MYSZKOWSKI, Saarbrücken
Mateu SBERT, Girona
Pavel SLAVÍK, Prague
Bert JÜTTLER, Linz
Alexander PASKO, Tokyo
Ken JOY, Davis (CA)
Alan CHALMERS, Bristol

Tosiyasu Laurence KUNII, Tokyo
Bianca FALCIDIENO, Genova
Jiří ŽÁRA, Prague
László SZIRMAY-KALOS, Budapest
Wolfgang STRASSER, Tübingen
Werner PURGATHOFER, Vienna
Diego GUTIERREZ, Zaragoza
Joaquim JORGE, Lisboa
Michela SPAGNUOLO, Italy

For the future, we know for sure that the SCCG 2022 will be chaired again by Werner PURGATHOFER, the founder of international SCCG IPC.

Conference Venue

This year the conference takes place in Mikulov, Czech Republic. Mikulov is a picturesque historical town located in southern Moravia in the very heart of a wine region. The first written record of the town dates back to 1173. The priest, cardinal and Bishop of Olomouc Franz von Dietrichstein took over the estate in 1611 to create a residence worthy of his position in the state. He was responsible for the transformation of Mikulov from a small provincial town into the most important town in Moravia at that time. The town offers perception with literally all senses. For many people the reasons for a visit include history embodied in architectural monuments, while for nature-lovers and people preferring active rest it is the unique natural conditions of the region. To those who appreciate good wine, Mikulov offers rich viticulture tradition. The geographical position, the climate and the varied soil of the Mikulov wine region with the limestone slopes of the Pálava Hills affect the unique character of the local wine. This was already discovered by Ancient Romans, who considered the Pálava region ideal for vine growing. Nowadays Mikulov and its surroundings offer a lot of vine paths with stops at local public or family wine cellars or wine tasting centres. Local vintners regularly win awards at prestigious international competitions.

Opening ceremony

The official opening of the conference will be performed by the conference chair and the Dean of the Faculty of Information Technology, Brno University of Technology - Prof. Pavel ZEMČÍK.

The scientific program will be opened by the SCCG 2017 programme chairs Ivan VIOLA and Jiří BITTNER, who invited to SCCG excellent speakers with talks on prospective and cutting-edge technology.

Special thanks to:

Ivan VIOLA and **Jiří BITTNER** who headed the International Program Committee, created the scientific program, and gave a new orientation to the conference.

We would like to thank **Niloy MITRA**, **Kwan-Liu MA**, **Daniel SÝKORA** and authors of the invited papers for accepting our invitation and sharing their thoughts with us.

We would like to thank the authors, reviewers, and IPC members for creating and selecting the most valuable recent ideas in our field. Have a good time at the Spring Conference on Computer Graphics 2017 in Czech Republic, enjoy all enjoyable and look for SCCG 2018 Call for Papers in future!

With best wishes,

SCCG 2017 Organizing Committee:

Vítězslav BERAN - the head of the organizing committee and posters chair
Pavel ZEMČÍK and Andrej FERKO - the souls of the organizing committee
Ivan VIOLA and Jiří BITTNER - the program chairs
Míša STUDENÁ, Elena ŠIKUDOVÁ and Milan FTÁČNIK - the organization team

SCCG 2017

Conference Chair

Pavel Zemčík

Pavel Zemčík (prof .Dr. Ing.) born 1965, since 2016, dean of Faculty of Information Technology, Brno University of Technology (FIT), member of Graph@FIT group, received his Ing. (M.Sc.) degree in 1989, from Faculty of Electrical Engineering, Brno University of Technology, Dr. (Ph.D.) in 1995, and doc. (tenure track) in 1998 from the same faculty. He became full professor (prof.) of FIT in 2012 and he successfully supervised 15 Ph.D. students and dozens of Bc. And M.Sc. His professional interests include computer graphics, image processing and computer vision, acceleration of algorithms of video and signal processing in programmable hardware, user interfaces, and application. In 1995, he co-founded CAMEA, spol. s r.o. (ltd.), company active in application of image and signal processing in industry and traffic. Pavel Zemčík authored/co-authored over 100 conference papers, 20 journal papers, and 5 book chapters as well as 5 patents and several industrial and traffic monitoring and enforcements applications. His hobbies, besides research, include music, skiing, and flying in ultramicro airplane and helicopter.

Publications:

- PŘIBYL Bronislav, CHALMERS Alan, ZEMČÍK Pavel, HOOBERMAN Lucy a ČADÍK Martin. Evaluation of Feature Point Detection in High Dynamic Range Imagery. Journal of Visual Communication and Image Representation. Amsterdam: Elsevier Science, 2016, vol. 38, no. 1, pp. 141-160
- BAŘINA David a ZEMČÍK Pavel. Vectorization and Parallelization of 2-D Wavelet Lifting. Journal of Real-Time Image Processing. 2015, pp. 1-13
- JURÁNEK Roman, HEROUT Adam, DUBSKÁ Markéta a ZEMČÍK Pavel. Real-Time Pose Estimation Piggybacked on Object Detection. In: Proceedings of ICCV. Santiago: IEEE Computer Society, 2015, pp. 1-9
- PŘIBYL Bronislav, ZEMČÍK Pavel a ČADÍK Martin. Camera Pose Estimation from Lines using Plücker Coordinates. In: Proceedings of the British Machine Vision Conference (BMVC 2015). Swansea: The British Machine Vision Association and Society for Pattern Recognition, 2015, pp. 1-12
- BAŘINA David a ZEMČÍK Pavel. Diagonal Vectorisation of 2-D Wavelet Lifting. In: International Conference on Image Processing (ICIP). Paris: IEEE Signal Processing Society, 2014, pp. 2978-2982

SCCG 2017

Invited Speaker

Niloy Mitra

Prof. Niloy J. Mitra leads the Smart Geometry Processing group in the Department of Computer Science at University College London. He received his PhD degree from Stanford University under the guidance of Leonidas Guibas. His research interests include shape analysis, computational design and fabrication, and geometry processing. Niloy received the ACM Siggraph Significant New Researcher Award in 2013 and the BCS Roger Needham award in 2015. His work has twice been selected and featured as research highlights in the Communication of ACM, received best paper award at ACM Symposium on Geometry Processing 2014, and Honourable Mention at Eurographics 2014. He is on the Editorial Board of ACM Transactions on Graphics and Computer Graphics Forums. Besides research, Niloy is an active DIYer and loves reading, bouldering, and cooking.

Publications:

- Lingjie Liu, Duygu Ceylan, Lin Cheng, Wenping Wang, Niloy J. Mitra. Image-based Reconstruction of Wire Art. ACM SIGGRAPH 2017
- Martin Kilian, Aron Monszpart, Niloy J. Mitra. String-Actuated Curved Folded Surfaces. ACM Transactions on Graphics 2017
- Tuanfeng Wang, Hao Su, Qixing Huang, Jingwei Huang, Leonidas Guibas, Niloy J. Mitra. Unsupervised Texture Transfer from Images to Model Collections. ACM SIGGRAPH Asia 2016
- Paul Guerrero, Gilbert Bernstein, Wilmot Li, Niloy J. Mitra. PATEX: Exploring Pattern Variations. ACM SIGGRAPH 2016
- Chi-han Peng, Yongliang Yang, Fan Bao, Daniel Fink, Dongming Yan, Peter Wonka, Niloy J. Mitra. Computational Network Design from Functional Specifications. ACM SIGGRAPH 2016

SCCG 2017

Invited Speaker

Kwan-Liu Ma

Kwan-Liu Ma is a professor of computer science and the chair of the Graduate Group in Computer Science (GGCS) at the University of California-Davis, where he directs VIDI Labs and UC Davis Center of Excellence for Visualization. His research spans the fields of visualization, computer graphics, high-performance computing, and user interface design. Professor Ma received his PhD in computer science from the University of Utah in 1993. During 1993-1999, he was with ICASE/NASA Langley Research Center as a research scientist. He joined UC Davis in 1999.

Professor Ma is presently leading a team of over 25 researchers pursuing research in scientific visualization, information visualization, visual analytics, visualization for storytelling, visualization interface design, and volume visualization. He received the NSF Presidential Early-Career Research Award (PECASE) in 2000, was elected an IEEE Fellow in 2012, and received the 2013 IEEE VGTC Visualization Technical Achievement Award for his outstanding research work. Professor Ma has been actively serving the research community by playing leading roles in several professional activities including VizSec, Ultravis, EGPGV, IEEE VIS, IEEE PacificVis, and IEEE LDAV. He has served as a papers co-chair for SciVis, InfoVis, EuroVis, and PacificVis. Professor Ma was an associate editor for the IEEE Transactions on Visualization and Computer Graphics (TVCG) during 2007-2011 and the Journal of Computational Science and Discovery during 2009-2014. He presently serves on the editorial boards of the IEEE Computer Graphics and Applications (CG&A) and the Journal of Visualization.

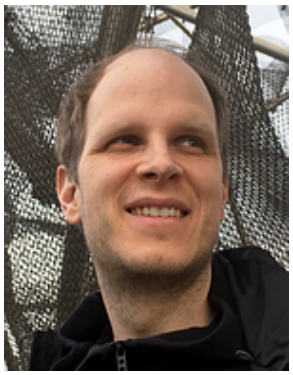
Publications:

- Chris Bryan, Kwan-Liu Ma, Jonathan Woodring. Temporal Summary Images: An Approach to Narrative Visualization via Interactive Annotation Generation and Placement. IEEE Trans. Vis. Comput. Graph. 23(1): 511-520, 2017.
- Franz Sauer, Kwan-Liu Ma. Spatio-Temporal Feature Exploration in Combined Particle/Volume Reference Frames. IEEE Trans. Vis. Comput. Graph. 23(6): 1624-1635, 2017.
- Oh-Hyun Kwon, Chris Muelder, Kyungwon Lee, Kwan-Liu Ma. A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization. IEEE Trans. Vis. Comput. Graph. 22(7): 1802-1815, 2016.
- Chris Muelder, Biao Zhu, Wei Chen, Hongxin Zhang, Kwan-Liu Ma. Visual Analysis of Cloud Computing Performance Using Behavioral Lines. IEEE Trans. Vis. Comput. Graph. 22(6): 1694-1704, 2016.
- Yuzuru Tanahashi, Chien-Hsin Hsueh, Kwan-Liu Ma. An Efficient Framework for Generating Storyline Visualizations from Streaming Data. IEEE Trans. Vis. Comput. Graph. 21(6): 730-742, 2015.
- Hongfeng Yu, Jinrong Xie, Kwan-Liu Ma, Hemanth Kolla, Jacqueline H. Chen. Scalable Parallel Distance Field Construction for Large-Scale Applications. IEEE Trans. Vis. Comput. Graph. 21(10): 1187-1200, 2015.

SCCG 2017

Invited Speaker

Daniel Sýkora



Daniel Sýkora is an Associate Professor at the Department of Computer Graphics and Interaction, Faculty of Electrical Engineering, Czech Technical University in Prague.

His primary research interest is firmly coupled with his long-standing passion for hand-drawn animation. He developed numerous techniques which allow eliminating repetitive and time-consuming tasks while still preserve full creative freedom of manual work. To turn these research ideas into practical products Daniel intensively cooperates with studio Anifilm in Prague as well as renowned industrial partners such as Disney, Adobe, or TVPaint Development.

Daniel received his MS (2003) and Ph.D. (2007) in Computer Science and Engineering from the Czech Technical University in Prague. Among Daniel's awards include e.g. the Günter Enderle Best Paper Award (Eurographics, May 2010), Best Paper Award (NPAR, August 2011) or The Neuron Award for Promising Young Scientists (Neuron, May 2017).

Publications:

- Jakub Fišer, Ondřej Jamriška, David Simons, Eli Shechtman, Jingwan Lu, Paul Asente, Michal Lukáč, and Daniel Sýkora. Example-Based Synthesis of Stylized Facial Animations. To appear in ACM Transactions on Graphics 36(4):155, 2017 (SIGGRAPH 2017, Los Angeles, USA, July 2017)
- Marek Dvorožňák, Pierre Bénard, Pascal Barla, Oliver Wang, and Daniel Sýkora. Example-Based Expressive Animation of 2D Rigid Bodies. To appear in ACM Transactions on Graphics 36(4):127, 2017 (SIGGRAPH 2017, Los Angeles, USA, July 2017).
- Michal Lukáč, Daniel Sýkora, Kalyan Sunkavalli, Eli Shechtman, Ondřej Jamriška, Nathan Carr, and Tomáš Pajdla. Nautilus: Recovering Regional Symmetry Transformations for Image Editing. To appear in ACM Transactions on Graphics 36(4):108, 2017 (SIGGRAPH 2017, Los Angeles, USA, July 2017)
- Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sýkora. StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings. In ACM Transactions on Graphics 35(4):92, 2016 (SIGGRAPH 2016, Anaheim, USA, July 2016), ACM, Patent pending
- Ondřej Jamriška, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. LazyFluids: Appearance Transfer for Fluid Animations. In ACM Transactions on Graphics 34(4):92, 2015 (SIGGRAPH 2015, Los Angeles, USA, August 2015)

SCGG 2017 Programme

Monday, May 15, 2017

11:00		registration (until 13:30)
12:00	13:30	Lunch
13:30	13:45	**Conference opening**
		Keynote 1 (chair: Jiří Bittner)
13:45	14:45	Back to the Roots: Bridging the Gap Between Hand-Drawn and Computer-Generated *Daniel Sykora*
14:45	15:15	Coffee Break
		Session 1 - Molecular Visualization (chair: Barbora Kozlíková)
15:15	15:40	QCDVis: a tool for the visualisation of Quantum Chromodynamics (QCD) Data *Dean P Thomas, Rita Borgo, Robert S. Laramee, Simon Hands* (Computers & Graphics track)
15:40	16:05	Metamorphers: Storytelling Templates For Illustrative Animated Transitions in Molecular Visualization *Johannes Sorger, Peter Mindek, Peter Rautek, Graham Johnson, Ivan Viola*
16:05	16:30	Visualization of molecular machinery using agent-based animation *Daniel Gehrer, Ivan Viola*
16:30	16:50	Coffee Break
		Session 2 - Modeling and Natural Phenomena (chair: Jiří Žára)
16:50	17:15	Computer Simulation of Furniture Layout when Moving from One House to Another *Takuya Yamakawa, Yoshinori Dobashi, makoto okabe, Kei Iwasaki, Tsuyoshi Yamamoto*
17:15	17:40	Stokes Coordinates *Yann Savoye*
17:40	18:05	Real-Time Simulation of Cumulus Clouds through SkewT/LogP Diagrams *Rui Pedro Duarte, Abel Gomes* (Computers & Graphics track)
19:00		**Conference Dinner**

Tuesday, May 16, 2017

		Session 3 - Information and Scientific Visualization (chair: Kresimir Matkovic)
09:00	09:25	Dynamic Word Clouds *Martin Seyfert, Ivan Viola*
09:25	09:50	Exploring Visual Prominence of Multi-Channel Highlighting in Visualizations *Manuela Waldner, Alexey Karimov, Eduard Gröller*
09:50	10:15	Data-Sensitive Navigation in Scientific Visualization *Peter Mindek, Gabriel Mistelbauer, Eduard Gröller, Stefan Bruckner* (Computers & Graphics track)
10:15	10:40	Transfer Function Optimization Based on a Combined Model of Visibility and Saliency *Shengzhou Luo, John Dingliana*
10:40	11:00	Coffee Break
11:00	12:00	**Keynote 2 (chair: Eduard Gröller)** Building a Factorized Scene Model: Capturing Appearance, Geometry, and Interactions *Niloy J. Mitra*

12:00	13:30	Lunch
		Session 4 - VR and Materials (chair: John Dingliana)
13:30	13:55	Design and Evaluation of a Novel Out-of-reach Selection Technique for VR using Iterative Refinement *Daniel Mendes, Daniel Medeiros, Maurício Sousa, Eduardo Cordeiro, Alfredo Ferreira, Joaquim Armando P Jorge* (Computers & Graphics track)
13:55	14:20	Automatic garment retexturing based on infrared information *Egils Avots, Morteza Daneshmand, Andres Traumann, Sergio Escalera, Gholamreza Anbarjafari* (Computers & Graphics track - invited paper)
14:20	14:45	BRDF Measurement of Highly-Specular Materials using a Goniometer *Jiri Filip, Radomir Vavra, Frank J. Maile*
14:45	15:15	Coffee Break
		Session 5 - Image processing and Vision (chair: Heikki Kälviäinen)
15:15	15:40	Combating anisotropy in histological serial sections with optical flow-based interpolation *Oleg Lobachev, Birte S. Steiniger, Michael Guthe*
15:40	16:05	True HDR camera with bilateral filter based tone mapping *Svetozár Nosko, Martin Musil, Petr Musil, Pavel Zemčík*
16:05	16:30	Effectiveness of the Bag-of-Words approach on the object search problem in 3D domain *Vladimir Privalov, Vitezslav Beran, Pavel Smrž*
16:30	16:50	Coffee Break
16:50	18:00	**Poster Session (chair: Vítězslav Beran)**
19:00		**Dinner + Wine tasting**

Wednesday, May 17, 2017

		Session 6 - Visual Analytics (chair: Elena Sikudova)
09:00	09:25	AnthroVis: Visual Analysis of 3D Mesh Ensembles for Forensic Anthropology *Katarina Furmanova, Petra Urbanova, Barbora Kozlikova*
09:25	09:50	Interactive Visual Categorization of Spinel Group Minerals *María Luján Ganuza, María Florencia Gargiulo, Gabriela Ferracutti, Silvia M. Castro, Ernesto Bjerg, Eduard Gröller, Kresimir Matkovic*
09:50	10:15	Super-Ensembler: Interactive Visual Analysis of Data Surface Sets *Lena Cibulski, Borislav Klarin, Martin Sopouch, Bernhard Preim, Holger Theisel, Kresimir Matkovic*
10:15	10:40	3DHOP: 3D Heritage Online Presenter *Marco Potenziani, Marco Callieri, Matteo Dellepiane, Massimiliano Corsini, Federico Ponchio, Roberto Scopigno* (Computers & Graphics track - Invited paper)
10:40	10:50	Coffee Break
		Keynote 3 (chair: Ivan Viola)
10:50	11:50	Audience-Targeted Exploratory and Explanatory Visualization *Kwan-Liu Ma*
11:50	12:00	**Best paper ceremony and Conference closing**
12:00	13:00	Lunch
13:00	16:00	**Excursion to Brno companies**

SHORT PAPERS & POSTERS

International Programme Committee

IPC chairmans

Ivan Viola

Vienna University of Technology, Austria

Jiří Bittner

Czech Technical University in Prague, Czech Republic

Baum, Daniel
Benes, Bedrich
Beran, Vitezslav
Bikker, Jacco
Bouatouch, Kadi
Bruckner, Stefan
Coelho, Antonio
Ďurikovič, Roman
Fuchs, Martin
Gomes, Abel
Gumhold, Stefan
Guthe, Michael
Havran, Vlastimil
Heinzl, Christoph
Höllt, Thomas
Chalmoviansky, Pavel
Chrysanthou, Yiorgos
Isenberg, Tobias
Jeschke, Stefan
Kainz, Bernhard
Kehrer, Johannes
Kolingerová, Ivana
Kozlikova, Barbora
Krone, Michael
Lawonn, Kai
Linsen, Lars
Mantiuk, Radoslaw
Matkovic, Kresimir
Mattausch, Oliver
Mindek, Peter
Musialski, Przemyslaw
Myszkowski, Karol
Patow, Gustavo
Pattanaik, Sumanta
Pereira, João Madeiras
Ritschel, Tobias
Ropinski, Timo
Sedlmair, Michael
Slavik, Pavel
Spanel, Michal
Turkay, Cagatay
Waldner, Manuela
Wang, Bei
Zara, Jiri

List of Reviewers

We thank all those (listed in alphabetical order), who helped making SCCG 2017 an interesting conference, by reviewing papers submissions within a short period of time:

Baum, Daniel
Benes, Bedrich
Beran, Vitezslav
Bikker, Jacco
Bouatouch, Kadi
Bruckner, Stefan
Coelho, Antonio
Ďurikovič, Roman
Fuchs, Martin
Gomes, Abel
Gumhold, Stefan
Guthe, Michael
Havran, Vlastimil
Heinzl, Christoph
Höllt, Thomas
Chalmoviansky, Pavel
Chrysanthou, Yiorgos
Isenberg, Tobias
Jeschke, Stefan
Kainz, Bernhard
Kehrer, Johannes
Kolingerová, Ivana
Kozlikova, Barbora
Krone, Michael
Lawonn, Kai
Linsen, Lars
Mantiuk, Radoslaw
Matkovic, Kresimir
Mattausch, Oliver
Mindek, Peter
Musialski, Przemyslaw
Myszkowski, Karol
Patow, Gustavo
Pattanaik, Sumanta
Pereira, João Madeiras
Ritschel, Tobias
Ropinski, Timo
Sedlmair, Michael
Slavik, Pavel
Spanel, Michal
Turkay, Cagatay
Waldner, Manuela
Wang, Bei
Zara, Jiri

List of Contributors

Adam Chromy

Andrew Ciambrone

Barbora Kozlikova

David Bařina

David Herman

David Kouřil

Denis Gracanin

Ivan Viola

Jaroslav Rozman

Kresimir Matkovic

Mathieu Le Muzic

Michal Spanel

Ondrej Klima

Pavel Zemcik

Petr Kleparnik

Stanislav Svoboda

Václav Stránský

A Sensor Fusion Approach to Extending Range of Environment Mapping Devices

Andrew Ciambrone, Denis Gračanin

Virginia Tech, Blacksburg, VA, USA

Krešimir Matković

VRVis Research Center, Vienna, Austria

Abstract

Currently available commercial spatial mapping devices mostly use infrared camera to obtain a depth map which is effective only for short to medium distances (3-4 meters). However, that range can be extended by using existing environment mapping devices and techniques and a combination of a camera, Inertial Measurement Unit, and Light Detection and Ranging devices supported by sensor fusion and computer vision techniques. The proposed approach consists of three steps. The first step is data collection and data fusion using embedded hardware; the second step is data processing (segmentation) and the third step is creating a geometry mesh of the environment. The developed system was evaluated by measuring the room dimension and objects within the room. This low cost system can expand the mapping range of the existing mixed reality devices such as Microsoft HoloLens device.

Keywords: mixed reality, sensor fusion, environment mapping, image processing, computer vision

1. Introduction

Spatial mapping or environment mapping is the process of exploring a real world environment and creating a digital representation of it. This process is used in various application domains, including mixed reality (MR). Current MR commercial devices, such as Microsoft HoloLens device, include support for environment mapping. However, the mapping is effective for short to medium ranges (3-4 meters). Most built environments contain large spaces that are not suitable for these devices thus limiting the applicability. The described approach focuses on augmenting the environment mapping capabilities of the current MR devices.

Real time environment mapping is an important component of many systems and is a well-researched concept. The real time approaches map the environment on the fly, usually using cloud point data. They are limited by the amount of time the system has to process the data (the amount of time it takes for the next frame of information to come in). Due to the limitations of a MR system, processing high density point cloud is difficult due to the limitations of portable Light Detection and Ranging (LIDAR) technologies and computer vision techniques. Creating a dense point cloud is often achieved by using expensive image processing techniques or LIDAR systems.

The proposed approach extends the existing environment mapping devices and techniques to map larger architectural environments using a combination of a camera, Inertial Measurement Unit (IMU), and LIDAR devices supported by sensor fusion and computer vision. The developed system was evaluated on how accurately it can estimate the dimensions of the space it is located in. The system was also compared to an existing high density LIDAR system.

2. Related Work

Mapping an environment can be done in three different ways, geometrically, radiometrically and semantically [1]. The geometric mapping spatially maps the real world, the radiometric mapping maps the color of the world, and the semantic mapping creates an understanding of the environment. Geometric mapping usually uses sensors such as LIDAR or range finder cameras while radiometric mapping typically uses cameras but could also use thermal imaging. Semantic mapping tries to create understanding in the scene such as object detection or understanding where objects are in reference to each other. Recently, the use of deep learning [1] has gained attention.

For cloud point collection there are two main types of methods, passive and active methods. Passive methods rely on reasonable lighting and typically use imagery for their methods of point collection. Active methods manipulate the scene to gather more information. Microsoft HoloLens and Kinect are good examples of mapping devices that use active methods to determine distances by using an infrared laser. An example of a passive method would be the "structure from motion" technique [1].

Feature extraction is the idea of locating point of interest from an image. Examples of the features in an 2D image are: pixel properties, textures, and shapes within an image. Texture features are the spatial placement of the intensity values in an image. For example, a checkerboard would give a different texture compared to a spherical contour. Looking at the texture features of an image is used to determine regions of an image [1]. Shape features can be detected by using contour filters such as the Canny, Sobel, Roberts, and Prewitt operators [1].

The simultaneous localization and mapping (SLAM) problem is one of incrementally building a consistent map of the en-

vironment and simultaneously determining its location within the map [2]. Work done in this area is closely related to the work done for the MR environment mapping.

Sensor fusion [3, 4] uses intelligent integration of data derived from a collection of disparate sensor so that the resulting information provide more accurate information or information that cannot be derived from individual sensors.

Image segmentation is the process of dividing a image into discrete parts [5]. The most common alfgorithms include clustering, edge detection, and machine learning techniques such as neural networks. Superpixel segmentation [6] is a method of image segmentation. Seeded Region Growing [7] is a image segmentation method that is quite often used in computer vision. The algorithm starts by picking an set of N seeds. Then for each iteration each of the seeds gain one additional pixel to its region. This is done by looking at the pixels that border the regions. The pixel that is added to the region is determined by the least difference from the region. If a pixel is approached by two regions then the algorithm considers that to be a boundary pixel [8]. The success of the algorithm relies on the selection of initial pixel seeds, e.g., the converging squares algorithm [9].

3. Problem Description

MR devices tend to be lightweight mobile devices. Their processing power, memory and other resources are constrained. Limited memory and limited CPU power mean that decisions must be made on how much data can be collected. Collecting too much data could introduce delay causing a latency between what the user sees and what the systems outputs. One solution to this problem would be to collect the sensor data on the device and process it on a more powerful machine by sending it over the network. However, this would add a bottleneck to the system due to network constraints.

Mapping large architectural (indoor) spaces is challenging due to a limited range of currently available MR devices. It can take some time for the user to navigate the whole area and complete space mapping. While user interactions do take place in the immediate surrounding of the user, having some information of the more distant parts of the environment (without the need for the user to go there) could provide better context for user interactions. For example, mapping objects further away

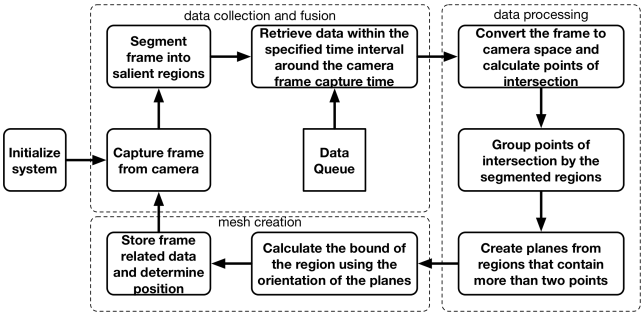


Figure 1: An outline of the system workflow.

(such as walls and high ceilings) enables construction of the model of the surrounding architectural space.

That can be achieved by augmenting the current MR devices with additional longer-range sensor technologies. However, these additions must be light and mobile and safe for the user to use. Due to these constraints main difficult design decisions must be made to produce fast but not necessarily highly accurate results. However, since this those results describe areas of space more distant form the user, even less accurate results are sufficient. When the user moves closer, the near-range mapping hardware is used to improve the accuracy.

4. Proposed Approach

Figure 1 shows the overall approach. The main processing loop is based on the camera input. Each camera frame is annotated by LIDAR and IMU data within a specified time distance from the frame capture time. The fused data is used to determine region in space and construct the surrounding geometry.

Overall, there are three main components. The first part focuses on data collection and data fusion. The second part processes the collected data and finally, the third component formulates and outputs a 3D mesh.

There are several types of data to be collected and processed by this system: images, range measurements, and the devices specific acceleration, gyro and magnetic data. This is achieved by using a combination of a camera, a LIDAR device, and an IMU device. The quality of the data and the rate of collection depends on the quality of the device. While having more data samples would create a better result, the system must fit within the constraints of a wearable device. Most LIDAR devices are big and bulky and are not portable. The most promising type of LIDAR would be a singular range finder device such as the Garmin LIDAR Lite v3.

Once the data is collected and bundled together based on the time it was captured, the system can then start processing the data. The first step is to segment the image into meaningful chunks. The purpose of the segmentation is to group pixels of the image into sections so that when the LIDAR does hit that portion of the image we can interpolate entire regions instead of just points of the image. There are various strategies for segmenting an image, such as edge detection, convolutional neural networks, or region-growing.

Given the segmented image the next step in the processing phase is to create the actual mesh from the images. Since the image is segmented into parts, the system assumes that the segments lie the same plane of orientation. To determine the region's orientation at least three LIDAR points must be captured on that particular region.

With a region's orientation it is possible to determine the size and the bounds of the region in the real world. Calculating the bounds of the region in the world space can be done by finding the intersection between a ray that coming from the camera and the calculated plane in camera space.

4.1. Data Collection and Processing

The following data types of data are being collected and used: the specific force or acceleration of the system, the angular rate, the magnetic field, image data, and distance data. To reduce the natural error of the IMU, sensor fusion is performed on the data. Since time performance is essential, a simplified version of Kalman filter, called RTQF, from the RTIMULib library is used [10]. The sampling rate for these components are 80 samples / second for the IMU sensor and 100 samples / second for the LIDAR sensor. The camera is capable of capturing up to 30 frames per second. However, due to the time it takes to process each frame the effective camera frame rate is reduced to approximately 5-6 frames per second. However, that is sufficient to map the environment given the user's speed.

We first process the data collected from the sensors before converting it into geometry meshes. This is done in three main steps, image segmentation, image annotation, and plane creation. The success of this system relies heavily on the segmentation image. A segmentation algorithm takes in image and partitions it into discrete parts. The main goal of any image segmentation is to simplify the image by grouping pixels together by corresponding surfaces or objects.

In this system the goals of the segmentation is to separate the image into surfaces and to get the bounds of the region. The system needs a segmentation algorithm that maintains consistency of the generated regions, be as close to real time as possible, and add a bias to creating larger regions. In indoor spaces there are a lot of variations in the image due to surface textures and lighting differences. Overhead lights and windows can create areas of intense light differences. Consequently, creating a good segmentation is difficult. To overcome these issues, images must be preprocessed before segmentation.

One major flaw with many image segmentation approaches is that they are often prone to noise in the image so steps must be taken to reduce the noise before the segmentation can be ran on the image. This can be done by blurring or smoothing the image. Blurring an image reduces the amount of noise and details in an image. Figure 2 left shows a segmentation without filtering. The black regions are segments below the minimum filter size. Figure 2 right shows a segmentation with filtering.

A filtered image is preferred compared to the unfiltered image because the noise and tiny details in a image prevent segmenting surfaces into larger segments. For example, when segmenting a brick wall the system would perform better if it created only one segment for the wall instead of having a segment for each brick. However, image blurring does have a cost. Blurring processes each pixel in an image which does add a computation required for each frame. It is important to realize the amount of blurring that takes place. Too much blurring and the image will lose critical details such as the edges between surfaces. Popular methods of image smoothing include Gaussian, median, and bilateral.

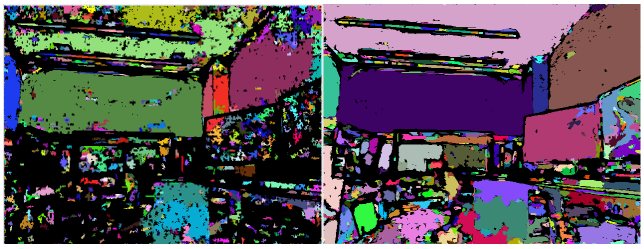


Figure 2: Segmentation without (left) and with filtering (right).

The median filter is used because it preserves the edges better then the Gaussian filter while at the same time removing noise in it. Although for this application it was found that while the bilateral filter does perform better than the median filter in retaining edges, it also takes much longer to process compared to the median filter. *RGB* is the typical format that frames from a camera come in however a color translation to a different color space such as the $L*A*B$ is rather simple. In this case the color space transformation is done using the OpenCV library.

4.2. Segmentation Algorithm

While the filtering and changing the color space does help with the differential lighting, it still holds a major influence in the image segmentation choice. Many of the image segmentation processes use a process called global thresholding. That works well for images that are consistent in lighting such as microscopic images but does not work well in an indoor space with overhead lighting. Therefore, the segmentation algorithm has to use adaptive threshold or some other method to compensate for the light differences.

Finally, the amount of time needed for the segmentation was a major decision point in choosing a optimal segmentation algorithm. Currently, a great deal of research is ongoing in using convolutional neural network (CNN) to segment an image. The selected image segmentation algorithm (Algorithm 1) is a simplified unseeded region growing algorithm.

Algorithm 1 Image Segmentation

```
procedure RegionGrowing(Img, MinRegionSize)
    Regions ← ∅
    RegionMap ← [][] (all zeros)
    RegionCounter = 1
    Seeds ← ∅
    CurrentRegion ← NULL
    while Pixels left to process do
        if Seeds.length = 0 then
            if CurrentRegion != NULL then
                Regions ← Regions ∪ {CurrentRegion}
            newSeed ∈ RegionMap ≠ 0
            CurrentSeeds ← newSeed
            CurrentRegion ← New Region based on newSeed
        repeat
            CurrentSeed ← seed ∈ Seeds
            for all Neighbor ∈ 8 or 4 neighbor near CurrentSeed do
                if Distance(CurrentSeed.Color, Neighbor.Color) < Threshold then
                    CurrentSeeds ← CurrentSeeds ∪ {Neighbor}
                    CurrentRegion ← CurrentRegion ∪ {Neighbor}
        until Seeds.length = 0
        Return Region Merging(Regions, MinRegionSize)
```

The region growing was selected because it creates consistent segments and follows the guidelines of the parameters pass such as minimum segment size. Another reason why region growing was selected over other methods was because unlike many of image segmentation algorithms region growing is a local method. It is only concerned with certain portions of the

Algorithm 2 Merge Regions

```
procedure REGION MERGING(Regions, MinRegionSize)
    for all Region ∈ Regions do
        if Region.size < MinRegionSize then
            if All Neighboring Regions are the same then
                NRegion = NRegion ∪ Region
            for all NeighRegion do
                if Distance(Region.AvgColor, NRegion.AvgColor) < Threshold then
                    Region = Region ∪ NRegion
    Return Regions
```

image during the segmentation which works well with the differences in lighting typically found in architectural spaces.

The region growing algorithm includes seed selection, region growing and mesh creation. Seed selection can be crucial to seeded region growing algorithms because those algorithms typically only have a number of regions they can grow. The algorithm above is considered unseeded because no initial seeds are set through the parameters. There is also no real hard limit to the number of regions except for the number of pixels the image. Since there is no limit to the number of seeds, the selection of initial seeds for a region has little impact and can be negated by region merging. However, a heuristic that uses a peaks method in [7] is a good method of seed selection. This method looks for the points of highest variation in the image.

After an initial seed is selected the algorithm the region growing portion begins. Popular methods for defining what a neighbor are the 8-connected or the 4-connected. In this system it was found that the 4-connected method has a slight advantage in creating an accurate segmentation. However, the advantage is not enough to justify the additional iterations the 4-connected neighborhood does create. If the color difference or distance between current region seed and the pixel neighbor isn't too large, the pixel neighbor will be added to the current region being built and added to the list of pixels to process. This process continues until there are no more seeds left. If a region runs out of seeds to process and there are still pixels left in the image left unlabeled, then another region is created on an unlabeled pixel.

Once every pixel in the image has been labeled with a region, the region merging portion of the algorithm begins. Region merging is used because the image contains artifacts from the camera sensor that were not filtered out by the median filter. For every region that is smaller then the minimum desired size, the region looks at its neighboring regions and determines if it can merge with one of those. There are two conditions to decide if a region will be merged with another. The first condition is if the difference between the average color of the region and its neighboring region is within the threshold. This condition prevents a bias from occurring with the order of the pixel processing. The second condition is when another region encompasses a region they are merged. This means that small regions that are surrounded by the same larger region will be merged with that region.

The properties of the created segments are.

1. All the pixels in the frame must belong to a region.
2. The points in the pixel must be connected at some point meaning no two isolated groups of pixels can be considered parts of the same region.
3. All regions are disjoint. There are no overlaps.

While it would be best if all of the data points are used, only those data points around the time of frame capture can be used to create independent planes. This constraint verifies that the initial position of the LIDAR data points were collected are around the same position that the frame was collected. The time range value is determined based on the assumption that the user does not move faster than one meter / second. With the frame segmented and the relevant data selected, the next step is to find where on the frame the LIDAR points lie. This is achieved by placing the frame into camera space.

Since there is little depth data, the system needs to extrapolate the depth's of other parts of the environment. This is done by assuming that the regions captured during the segmentation process lie on the same orientation plane. With the orientation of a region its bounds and size can be determined. Which allows the system to extrapolate additional depth points of the region. To achieve this the LIDAR points need to be grouped by the region which they fall into.

For each region with three or more points, a check is done to verify if the points are well suited to create the plane. If the three points are overlapping or are co-linear, creating a plane is not possible. In cases where there are more then three points a search must be performed that chooses the three points that form the largest possible triangle. By creating the largest possible triangle the amount of error in the plane creation and boundary extrapolation is reduced.

Once the system has determined that the region does contain three points that are optimal for creating a plane, the system can then proceed to calculating the position of the bounds of the region in camera space. The current implementation only calculates for eight boundary points. However, this could be easily extended to more points. In order to correctly determine the boundary point locations we need to also calculate a plane for the region. All the boundary points and LIDAR points are saved and used are used to represent the mesh of that region.

4.3. Implementation

Because the target MR devices are lightweight, the developed system also has to be lightweight. We use Raspberry PI 3, Pi camera, the Garmin LIDAR Lite 3 and the IMU based on MPU9250. The total cost of the system is approximately $200. Figure 3 shows the develop system (bottom right) deployed during the experiment.

5. Evaluation

The described system was evaluated by its ability to predict environment dimensions from the points in the created mesh and by its ability to estimate the dimensions of an object in the environment.

When evaluating the system for its ability to detect the dimensions of the room, the system's data was compared against the data from the LIDAR puck and against a ground truth which was determined from measurements taken from both a laser

Figure 3: Object dimension evaluation setup.

tape measure and a measuring tape. The evaluation was conducted in a room where Microsoft HoloLens device would be unable to detect the walls if placed in the center of the room due to the size of the room. To provide a comparison between modern state of the art versus our implementation the system was compared against the Velodyne LIDAR Puck (VPL-16). Both the implemented system and the LIDAR Puck were placed in the center of the room at approximately the same height. The implemented system was placed on top of a tripod where it was rotated around the room completing a full rotation capturing different parts of the room.

The results from the Room Dimension evaluation can be found in Table 1. The dimensions of the room were calculated by finding the maximum and minimum values points in each of the spatial dimensions. However, due to the error in the gyroscope yaw value some of the extrapolated region boundary points may had caused some of the data points to be erroneous. To account for these erroneous data points a median value was calculated at each of the directions where meshes were created. Table 1 contains the calculated dimension of both data sets.

The percent error on the x-axis of the developed system for all data points is 42.8% however when accounting for the erroneous points the percent error on the x-axis is 1.5%. For the compared system the percent error on the x-axis is 26.5%. The error in the compared system was high due to the windows in the room. The percent error found on the y-axis of the developed system for all data points 30.1%. However when accounting for the erroneous points the percent error on the y-axis is 10.3%. For the compared system the percent error on the y-axis is 0.2%. The percent error on the z-axis of the developed system for all data points is 176.02% however when accounting for the erroneous points the percent error on the z-axis is 35.5%. For the compared system the percent error on the x-axis is 6.2%.

6. Discussion

Due to the error in the gyroscope's yaw an absolute orientation of the room was not going to accurately represent the direction that the developed system was pointed at. An estimate

Table 1: Dimensions of the room.

Metric	Length x (cm)	Width y (cm)	Height z (cm)
Ground Truth	1097.28	871.22	273.05
LIDAR Puck	1388.11	873.75	289.98
System (All Points)	1567.01	1133.04	482.54
System (Median)	1080.80	960.72	176.02

of the system's orientation was later applied to the data points collected. Frames were captured at eight different orientations evenly distributed on the z-axis. This estimate may have introduced errors into the estimate of the room dimensions. Another factor that introduced error on the z-axis was that no data was collected of the floor and ceiling. That is evident in the high error in approximation of the rooms height.

Another important observation to note is of the error on the x-axis for the LIDAR puck. The LIDAR puck was calibrated and measurements were taken multiple times following the direction of the manufacturer (Figure 4). However, the error on the x-axis remained. Figure 5 shows the corresponding data captured by the developed system in each corner of the room and the center areas of the walls.

An important part of environment mapping is accurately representing the dimensions of objects in a room. In this part of the evaluation the system ability to perform this task was evaluated. The dimensions of the object that the system was trying to measure a white board hanging on a wall (Figure 3).

The distance and angle between the object and the system affected the accuracy of the system's estimate of the object's dimensions. In order to test the system's ability to accurately estimate the dimension of an object the system was placed facing that object at various distances and angles about the room.

Table 2 shows the changes in accuracy from the different distances and angles taken around the room. Figure 6 shows

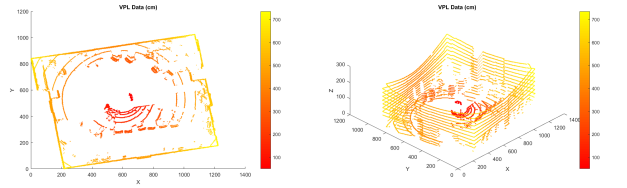


Figure 4: LIDAR data (top view) LIDAR data (perspective view).

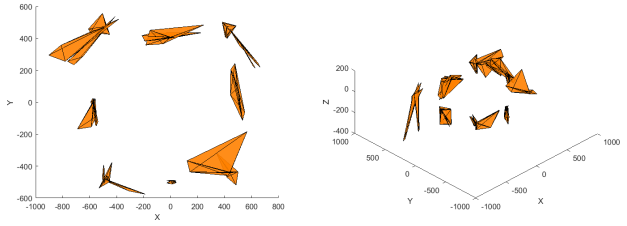


Figure 5: System data (top view) System data (perspective view).

Table 2: Dimensions of the white board (in centimeters)

Metric	Width	Height	Angle°	Distance
Actual	169.22	117.48	*N/A*	*N/A*
Reading 1	167.63	109.20	0	182.88
Reading 2	164.14	117.58	0	271.78
Reading 3	157.85	106.10	0	383.54
Reading 4	168.79	127.73	0	510.54
Reading 5	147.53	100.62	0	645.16
Reading 6	141.24	114.12	25	353.06
Reading 7	136.00	124.57	50	347.98
Reading 8	94.10	126.81	60	342.90
Reading 9	102.31	127.18	50	490.22

the percent error of the dimensions based on the distance and angle of the implemented system.

There are a few things that cause the error in calculating the error in an objects dimensions. The first is accuracy of the segmentation. If the segmentation is too large or too small the predicted boundaries of the object are going to be different then what they truly are. The effect of the segmentation error has increases as the distance increases because the further the system is from the object the more spatial impact each pixel has.

However, the largest factor in the amount of error in the calculation is the error in the gyroscope, particularly the value of the yaw. Looking at Figure 6 bottom it is apparent that the amount of error in the object's width is much higher then the error in the height. Conversely, looking at Figure 6 top when the system is parallel to the white board, the error of the width and height are roughly around the same area.

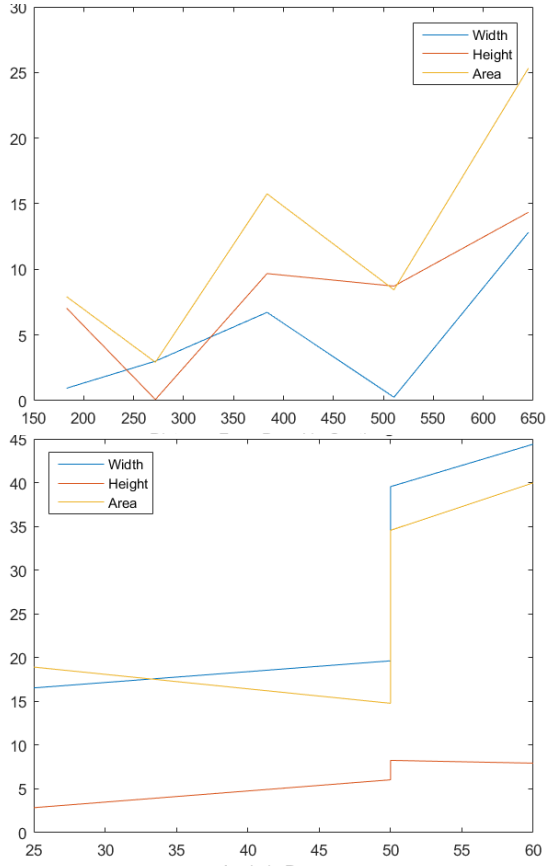


Figure 6: Top: Percentage error of object dimension estimate as a function of distance in centimeters. Bottom: Percentage error of object dimension estimate as a function of angle in degrees.

7. Conclusion

While the system based on the proposed approach has been developed, implemented and tested, many improvements are possible. Because of the limitations of the hardware and how the system is being used, the results depend on the user of the system. If the user does not provide a good amount of rotational range then many of the LIDAR points may overlap or be co-linear meaning planes can't be constructed and meshes can not be generated. The approach is limited to working with sparse point clouds. Providing a visual feedback to the user in terms of space coverage will help improve the cloud point quality and enhance environment mapping.

References

[1] Weinmann, M.. Reconstruction and Analysis of 3D Scenes: From Irregularly Distributed 3D Points to Object Classes. Springer; 2016.

[2] Durrant-Whyte, H., Bailey, T.. Simultaneous localization and mapping: Part I. IEEE Robotics Automation Magazine 2006;13(2):99–110.

[3] Das, S.. High-Level Data Fusion. Boston: Artech House; 2008.

[4] Kim, J.S., Gračanin, D., Quek, F.. Sensor-fusion walking-in-place interaction technique using mobile devices. In: Proceedings of the 2012 IEEE Virtual Reality Conference (VR 2012). 2012, p. 39–42.

[5] Gonzalez, R.C., Woods, R.E.. Digital Image Processing. Upper Saddle River, New Jersey 07458: Prentice Hall; second ed.; 2002.

[6] Ren, X., Malik, J.. Learning a classification model for segmentation. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE; 2003, p. 10–17.

[7] Adams, R., Bischof, L.. Seeded region growing. IEEE Transactions on pattern analysis and machine intelligence 1994;16(6):641–647.

[8] Kamdi, S., Krishna, R.. Image segmentation and region growing algorithm. International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2012;2.

[9] O'Gorman, L., Sanderson, A.C.. The converging squares algorithm: An efficient method for locating peaks in multidimensions. IEEE Transactions on Pattern Analysis and Machine Intelligence 1984;PAMI-6(3):280–288.

[10] Lindegaard, J.B.. RTIMULib - a versatile 9-dof IMU library for embedded Linux systems. `https://github.com/mrbichel/RTIMULib`; 2016. [last accessed 27 March 2017].

New Transforms for JPEG Format

Stanislav Svoboda, David Barina*

Faculty of Information Technology
Brno University of Technology
Bozetechova 1/2
Brno, Czech Republic

Abstract

The two-dimensional discrete cosine transform (DCT) can be found in the heart of many image compression algorithms. Specifically, the JPEG format uses a lossy form of compression based on that transform. Since the standardization of the JPEG, many other transforms become practical in lossy data compression. This article aims to analyze the use of these transforms as the DCT replacement in the JPEG compression chain. Each transform is examined for different image datasets and subsequently compared to other transforms using the peak signal-to-noise ratio (PSNR). Our experiments show that an overlapping variation of the DCT, the local cosine transform (LCT), overcame the original block-wise transform at low bitrates. At high bitrates, the discrete wavelet transform employing the Cohen–Daubechies–Feauveau 9/7 wavelet offers about the same compression performance as the DCT.

Keywords: JPEG, lossy image compression, transform coding, discrete cosine transform, discrete wavelet transform

1. Introduction

In last decades, needs for high-quality photography are growing, and so demands for efficient data storage are also growing. Therefore, it is important to compress the data as much as possible while preserving the quality of the image. For example, transferring a large number of images with high resolution across the Internet without a certain level of compression would be very time-consuming. Regarding the photography, the problem can be addressed by lossy image compression. Nowadays, the JPEG standard [1], dating back to 1991, is still the most widely used format for the lossy compression. Figure 1 shows the underlying compression chain. Each color component is transformed by blocks 8×8 using the 2-D DCT. The DCT has the property that, for a typical image, most of the visually significant information about the image in the 8×8 block is concentrated in just a few coefficients. This allows for better image compression. The transform coefficients are further quantized and fed into an entropy coder.

Since then, several other lossy image compression standards have been standardized. However, none of them became more popular with the public than the original JPEG. Particularly, the JPEG 2000 [2] decomposes large image tiles using the discrete wavelet transform (DWT). The advantage of wavelets is that wavelets are located on the small area in the image domain. Another interesting standard is JPEG XR [3], which is based on an overlapping hierarchical transform, so-called lapped biorthogonal transform (LBT). The last of the standards to be mentioned is WebP [4], based on the DCT complemented by Walsh–Hadamard transform (WHT).

Figure 1 shows that the JPEG block-wise scheme is very general. This opens the way to incorporate some other suitable transforms into the same compression chain. This is the motivation behind our research.

The rest of the paper is organized as follows. Section 2 presents the JPEG chain in the necessary level of detail. Subsequent Section 3 deals with the transforms suitable for involvement in this chain and examines their compression capabilities. Eventually, Section 4 summarizes and closes the paper.

2. JPEG Format

Part 1 of JPEG standard [1] specifies the method of lossy compression for digital images, based on the discrete cosine transform (DCT). This section describes a simplified description of JPEG image compression.

The color model to be used is $\mathrm{YC_BC_R}$. Therefore, the representation of the colors in the image is first converted from RGB to $\mathrm{YC_BC_R}$. The transformation into the $\mathrm{YC_BC_R}$ model enables the next usual step, which is to reduce the spatial resolution of the $\mathrm{C_B}$ and $\mathrm{C_R}$ components. For the rest of the compression process, Y, $\mathrm{C_B}$, and $\mathrm{C_R}$ components are processed separately.

As a next step, each component is split into blocks of 8×8 samples, $x_{k,l}$ for $(0,0) \leq (k,l) < (8,8)$. The samples are then shifted down by 128, assuming an 8-bit depth. Subsequently, each block undergoes the two-dimensional discrete cosine transform

$$X_{m,n} = \lambda_{m,n} \sum_{k,l} \cos\left(\frac{\pi(k+1/2)m}{N}\right) \cos\left(\frac{\pi(l+1/2)n}{N}\right) x_{k,l}, \quad (1)$$

*Corresponding author
Email addresses: xsvobo0b@stud.fit.vutbr.cz (Stanislav Svoboda), ibarina@fit.vutbr.cz (David Barina)

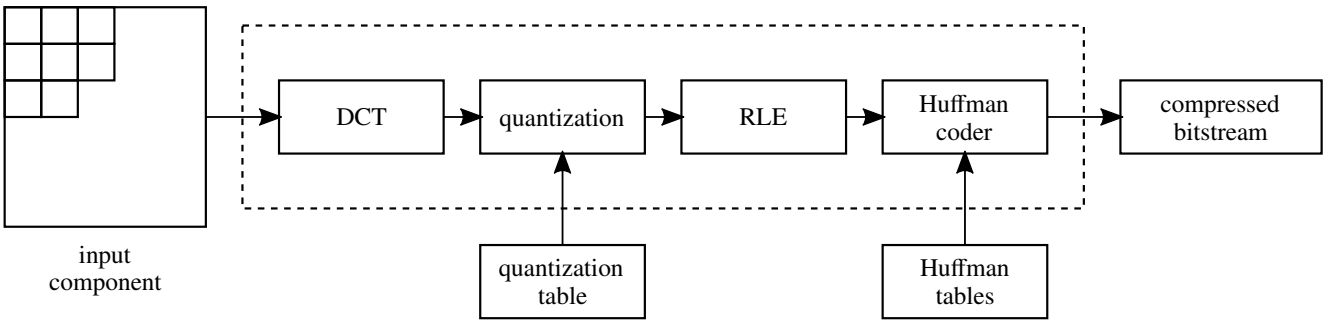


Figure 1: JPEG overview. The dashed line indicates the compression chain.

where $(0,0) \leq (m,n) < (8,8)$, $\lambda_{m,n}$ is a scaling factor, and $N = 8$. Now, the amplitudes of the coefficients are quantized. When performing a block-based transform and quantization, several types of artifacts can appear, especially blocking artifacts. The blocking artifacts are shown in Figure 2. The artifacts can be reduced by choosing a finer quantization, which corresponds to a lower level of compression.

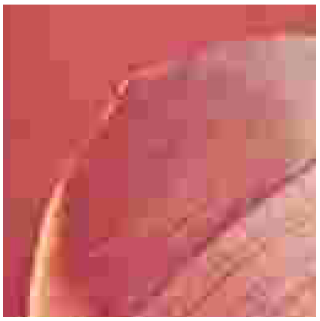


Figure 2: Blocking artifacts caused by the JPEG compression.

The DCT itself is a lossless process since the original input can be exactly reconstructed by applying an inverse transform to the coefficients $X_{m,n}$ directly. In order to achieve substantial compression ratio, quantization is applied to reduce the levels of the coefficients. The uniform quantization procedure is used to quantize the coefficients. One of up to four quantization tables $Q_{m,n}$ may be used in the quantization procedure. No default quantization table is specified in the standard. The quantization is formulated as

$$\hat{X}_{m,n} = \text{round}\left(\frac{X_{m,n}}{Q_{m,n}}\right), \tag{2}$$

where the round(a) operator rounds value a to the nearest integer. Human visual system is more immune to a loss of higher spatial frequency components than a loss of lower frequency components. This allows quantization to greatly reduce the amount of information in the high-frequency components.

After quantization, the $\hat{X}_{m,n}$ coefficients are fed into an entropy coder. Entropy coding employed in the JPEG is a special form of lossless compression. The $\hat{X}_{0,0}$ coefficient (DC coefficient) is treated differently than other coefficients (AC coefficients). The latter ones are converted into a one-dimensional "zig-zag" sequence. The rest of the process involves run-length encoding (RLE) of zeros and then using Huffman coding (arithmetic coding is possible, however rarely used).

From the above, it is clear that the scheme is almost independent of the transform used. Consequently, it would seem logical to substitute the DCT with some other similar transform. Several other papers on this topic have already been published. Some of them are briefly reviewed below. The authors of [5] examined the possibility of using the discrete Chebyshev transform (DChT) in JPEG. As reported in their paper, the DChT overcomes DCT on images with sharp edges and high predictability. In [6], the author compared the compression performance of the block-wise DCT against several lapped transforms. He concluded that lapped transforms have less blocking than the DCT and show some PSNR improvement over the DCT.

Considering the existing papers, we see that a wider comparison of the transforms in the JPEG compression chain is missing. The next section investigates the performance of some promising transforms in conjunction with the JPEG compression.

3. New Transforms for JPEG Format

This section interleaves a description of the transforms and their evaluation. The evaluation was performed on two datasets [7, 8]. At the beginning, trigonometric transforms are investigated. Subsequently, separable and non-separable wavelet, Chebyshev, and Walsh–Hadamard transforms are examined.

3.1. Discrete Sine Transform

The discrete sine transform (DST) is very similar to the DCT, except cosines are replaced with sines. Recall that the DCT has the property that, for a typical image, most of the information is concentrated in just a few coefficients $X_{m,n}$ with the lowest (m,n) indices. However, this property is not always valid for sine transforms. We found one variant for which the property holds. In the literature, this variant is referred to as the DST-VII [9] variant. Since most of the transforms investigated in this paper are separable, only the one-dimensional definitions are given from now on. The DST is defined by

$$X_m = \lambda_m \sum_k \sin\left(\frac{\pi(k+1)(m+1/2)}{N+1/2}\right) x_k, \tag{3}$$

where λ_m is a scaling factor.

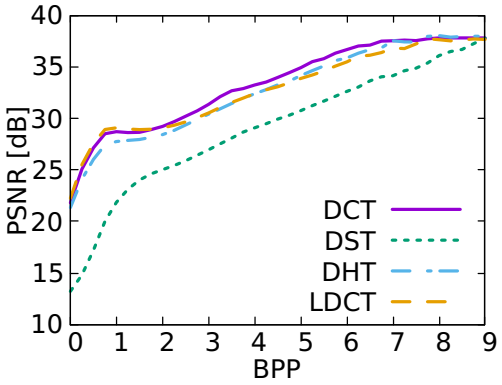


Figure 3: Comparison of the DCT, DST, DHT, and LCT. The LCT overcomes the other transforms at low bitrates.

3.2. Discrete Hartley transform

Like the previous transforms, the discrete Hartley transform (DHT) [10] is also based on trigonometric functions. In fact, its definition looks very similar to the definition of discrete Fourier transform (DFT). Unlike the DFT, the discrete Hartley transform maps real inputs onto real outputs, with no involvement of complex numbers. The transform is defined by

$$X_m = \sum_k \operatorname{cas}\left(\frac{2\pi km}{N}\right) x_k, \tag{4}$$

where $\operatorname{cas}(\alpha) = \cos(\alpha) + \sin(\alpha)$.

3.3. Local Cosine Transform

The local cosine transform (LCT) [11] reduces and smoothes the block effects. The local cosine transform is based on the standard block-based DCT. However, the local cosine transform has basis functions that overlap adjacent blocks. Prior to the DCT algorithm, a preprocessing phase in which the image is multiplied by smooth bell functions that overlap adjacent blocks is applied. This phase is implemented by folding the overlapping parts of the bells back into the original blocks. The standard block-based DCT algorithm then operates on the resulting blocks.

The folding operations are defined as

$$f_-(n) = \frac{b(n)f(-n) - b(-n)f(n)}{b(n) - b(-n)}, \tag{5}$$

$$f^+(n) = \frac{b(n)f(n) - b(-n)f(-n)}{b(n) - b(-n)}, \tag{6}$$

where the $f_-(n)$ is nth coefficient to the left (top) of the current block, the $f^+(n)$ is nth coefficient to the right (bottom), and $b(n) = \beta((2n+1)/N)$ is a bell function, where

$$\beta(n) = \begin{cases} 0 & n < -1 \\ \frac{1+\sin(\pi n/2)}{2} & \text{otherwise} \\ 1 & n > +1 \end{cases}. \tag{7}$$

The comparison of all the transforms discussed above is shown in Figure 3. The x-axis indicates bits per pixel (bpp). The discrete sine transform performs significantly worse than the reference DCT. This is caused by artifacts on block boundaries, as shown in Figure 4. Also, the discrete Hartley transform performs worse than the DCT. As we have found, this is caused by blocking artifacts at higher bitrates, where the artifacts are no longer visible with the DCT. At lowest bitrates, the local cosine transform has a better image quality than the reference DCT. Unfortunately, at higher bitrates, the image quality is slightly worse. The results on lower bitrates are caused by reduced blocking artifacts.

Figure 4: Sample image (on the left) and DST artifacts on block boundaries (on the right).

3.4. Discrete Wavelet Transform

The discrete wavelet transform (DWT) became a very popular image processing tool in last decades. For example, the JPEG 2000 standard is based on such decomposition technique. In more detail, the DWT decomposes the image into several subbands, while employing simple basis functions, the wavelets [12]. The transform is usually applied on large image tiles instead of small 8×8 blocks. Consequently, there are no blocking artifacts at all. In this paper, two well-known biorthogonal wavelets are used, the Cohen–Daubechies–Feauveau (CDF) [13] 5/3 and 9/7 wavelets. Incidentally, both of them are employed in the JPEG 2000 standard. In order to fit into JPEG compression chain, the wavelet transforms were designed to create a regular 8×8 grid of coefficients. This design corresponds to three levels of a dyadic decomposition [14]. The coordinates of the coefficients in the 8×8 blocks are computed using bit-reversal operations. In this way, the coefficients more closely correspond to the DCT coefficients. Note that both of the transforms were implemented using a lifting scheme [15, 16]. The lifting scheme can decompose the wavelet transforms into a finite sequence of simple filtering steps (lifting steps). Usually, the first step in the pair is referred to as the predict and the second one as the update.

3.5. Red-Black Wavelet Transform

The red-black wavelet transform (DWT RB) [17] is computed using a 2-D lifting scheme on a quincunx lattice [18]. The wavelets constructed in this way are inherently non-separable. Consequently, the red-black wavelets are less anisotropic than the classical tensor product wavelets (the classical DWT). In other words, the classical DWT will favor horizontal and vertical

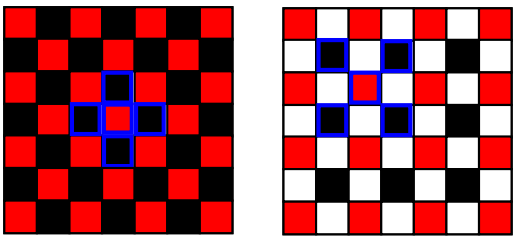


Figure 5: Lifting scheme on the quincunx lattice. Filter samples are bordered in blue. The first step (predict) on the left, the second (update) on the right.

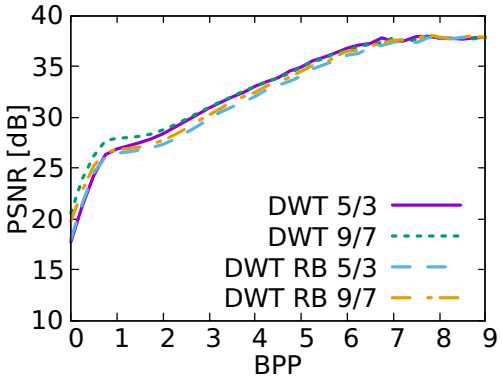


Figure 6: Comparison of the separable and non-separable (red-black) wavelet transforms. The separable CDF 9/7 transform has the best overall performance.

features of the image. This is not visible in the red-black wavelet transform.

The construction of the red-black wavelets is based on the CDF wavelets above. Therefore, the CDF 5/3 and CDF 9/7 wavelets are used also for this construction. The individual steps of the lifting scheme are illustrated in Figure 5. The details can be found in [17].

According to the results shown in Figure 6, it is clear that the separable CDF transform performs better than the red-black transforms. The worse results of the non-separable CDF 9/7 are caused by stain artifacts on the edges of objects, as illustrated in Figure 7. The CDF 9/7 transforms perform always better than the CDF 5/3. Therefore, the best combination seems to be separable the CDF 9/7 transform.

Figure 7: Comparison of the non-separable CDF 9/7 (on the left) wavelet transform and separable CDF 9/7 (on the right) in artefacts at edges.

3.6. Discrete Chebyshev Transform

The discrete Chebyshev transform (DChT) [19] is a polynomial-based transform, which employs Chebyshev polynomials of the first kind $T_n(x)$. Since the DCT is closely associated with a Chebyshev Polynomial series as $\cos(n\alpha) = T_n(\cos(\alpha))$ for some α, the discrete Chebyshev transform can be viewed as a natural modification of the DCT. The discrete Chebyshev transform is then defined using the polynomials

$$t_p(x) = (A_1 x + A_2) t_{p-1}(x) + A_3 t_{p-1}(x), \tag{8}$$

where A_1, A_2, and A_3 are constant. The transform is then defined as

$$X_m = \sum_k t_m(k)\, x_k. \tag{9}$$

3.7. Walsh–Hadamard Transform

The last of the transforms discussed in this paper is the Walsh–Hadamard transform (WHT). The computation [20] of this transform should be very fast since only additions/subtractions are involved here. The transform is defined as

$$X_m = 1/N \sum_k W(m,k)\, x_k, \tag{10}$$

where

$$W(m,k) = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & +1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix}. \tag{11}$$

Figure 8 shows the performance of the discrete Chebyshev transform. It is evident that the DChT is everywhere slightly below the DCT. Finally, Figure 9 shows an overall comparison, including the Walsh–Hadamard transform. Also, the WHT does not overcome DCT in any part of the plot. The only advantage of

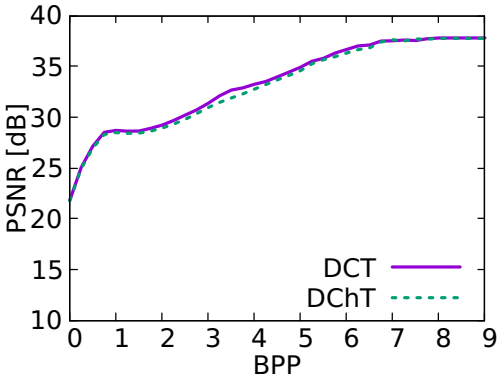


Figure 8: Comparison of the DCT and DChT. The DCT is slightly better.

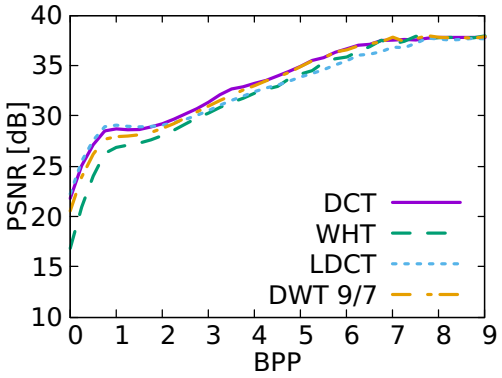


Figure 9: Overall comparison of selected transforms.

the WHT is the computation performance. In summary, only the local cosine transform (LCT) overcame the original block-wise DCT, especially at low bitrates. In addition, it removes blocking artifacts, as documented in Figures 10 and 12. Additionally, the separable discrete wavelet transform with the CDF 9/7 wavelet offers about the same compression performance as the DCT at high bitrates. To improve the mental image of evaluated transforms, bases of selected transforms are visually compared in Figure 11.

4. Conclusions

The JPEG image compression format uses a lossy form of compression based on the discrete cosine transform. This paper deals with a substitution the discrete cosine transform in the JPEG compression with some other similar transform. Several practical transforms were examined, including other trigonometric transforms, separable and non-separable wavelet transforms, a transform employing Chebyshev polynomials, and the Walsh–Hadamard transform. These transforms were evaluated on several image datasets.

The experiments show that only the local cosine transform overcomes the original block-wise DCT at low bitrates. Besides, it removes blocking artifacts. At high bitrates, the CDF 9/7 discrete wavelet transform performs similarly as the DCT.

In future work, we plan to focus on other transforms that have not been covered here, including also non-linear transforms.

Acknowledgements

This work has been supported by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science (no. LQ1602), and the Technology Agency of the Czech Republic (TA CR) Competence Centres project V3C – Visual Computing Competence Center (no. TE01020415).

References

[1] ITU-T Recommendation T.81. Information technology – Digital compression and coding of continuous-tone still images – Requirements and guidelines; 1992.

[2] ITU-T Recommendation T.800. Information technology – JPEG 2000 image coding system: Core coding system; 2000.

[3] ITU-T Recommendation T.832. Information technology – JPEG XR image coding system – Image coding specification; 2009.

[4] RFC 6386. VP8 Data Format and Decoding Guide; 2011.

[5] Mukundan R. Transform coding using discrete Tchebichef polynomials. In: Proceedings of the 6th IASTED International Conference on Visualization, Imaging, and Image Processing, VIIP. ISBN 978-088986598-3; 2006, p. 270–5.

[6] Malvar HS. Lapped biorthogonal transforms for transform coding with reduced blocking and ringing artifacts. In: IEEE International Conference on Acoustics, Speech, and Signal Processing; vol. 3. 1997, p. 2421–4. doi:10.1109/ICASSP.1997.599545.

[7] Olmos A, Kingdom FAA. A biologically inspired algorithm for the recovery of shading and reflectance images. Perception 2004;33(12):1463–73. doi:10.1068/p5321. PMID: 15729913.

[8] Franzen R. Kodak lossless true color image suit. released by the Eastman Kodak Company. URL `http://r0k.us/graphics/kodak/`.

[9] Puschel M, Moura JMF. The algebraic approach to the discrete cosine and sine transforms and their fast algorithms. SIAM Journal on Computing 2003;32(5):1280–316. doi:10.1137/S009753970139272X.

[10] Bracewell RN. Discrete hartley transform. J Opt Soc Am 1983;73(12):1832–5. doi:10.1364/JOSA.73.001832.

[11] Aharoni G, Averbuch A, Coifman R, Israeli M. Local Cosine Transform — A Method for the Reduction of the Blocking Effect in JPEG. Boston, MA: Springer US. ISBN 978-1-4615-3260-6; 1993, p. 7–38. doi:10.1007/978-1-4615-3260-6_2.

[12] Daubechies I. Ten Lectures on Wavelets; vol. 61 of *CBMS-NSF regional conference series in applied mathematics*. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics; 1992. ISBN 0898712742. doi:10.1137/1.9781611970104.

[13] Cohen A, Daubechies I, Feauveau JC. Biorthogonal bases of compactly supported wavelets. Communications on Pure and Applied Mathematics 1992;45(5):485–560. doi:10.1002/cpa.3160450502.

[14] Mallat S. A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 1989;11(7):674–93. doi:10.1109/34.192463.

[15] Daubechies I, Sweldens W. Factoring wavelet transforms into lifting steps. Journal of Fourier Analysis and Applications 1998;4(3):247–69. doi:10.1007/BF02476026.

[16] Sweldens W. The lifting scheme: A custom-design construction of biorthogonal wavelets. Applied and Computational Harmonic Analysis 1996;3(2):186–200. doi:10.1006/acha.1996.0015.

[17] Uytterhoeven G, Bultheel A. The Red-Black wavelet transform. In: Signal Processing Symposium (IEEE Benelux). IEEE Benelux Signal Processing Chapter; 1998, p. 191–4.

[18] Feilner M, Ville DVD, Unser M. An orthogonal family of quincunx wavelets with continuously adjustable order. IEEE Transactions on Image Processing 2005;14(4):499–510. doi:10.1109/TIP.2005.843754.

[19] Corr P, Stewart D, Hanna P, Ming J, Smith FJ. Discrete Chebyshev transform. a natural modification of the DCT. In: Proceedings 15th International Conference on Pattern Recognition. ICPR; vol. 3. 2000, p. 1142–5. doi:10.1109/ICPR.2000.903746.

[20] Shanks JL. Computation of the fast Walsh-Fourier transform. IEEE Transactions on Computers 1969;C-18(5):457–9. doi:10.1109/T-C.1969.222685.

Figure 10: Visual comparison of the original image (left), DCT (middle, PSNR 23.0 dB), and LCT (right, PSNR 23.3 dB).

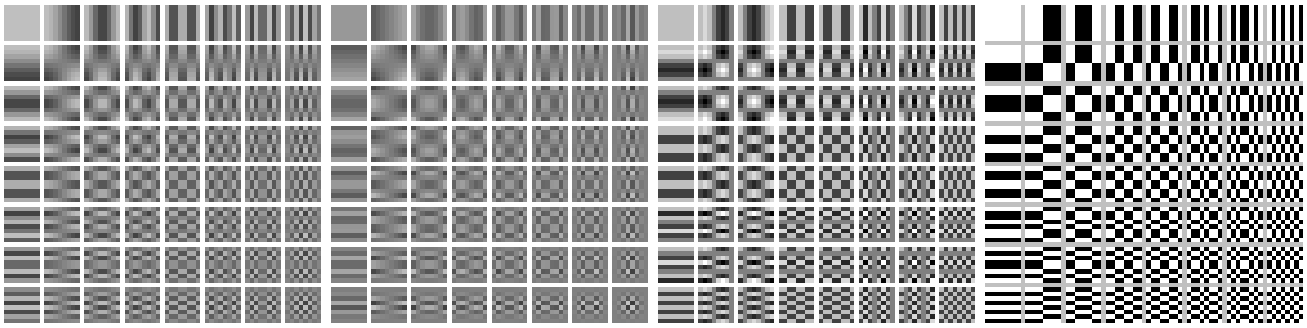


Figure 11: Basis images of selected transforms, from the left: the DCT, DChT, DHT, and WHT. DC coefficient is located in the top left corner.

Figure 12: Blocking artifacts, from the left: the original image, DCT, and LCT.

Model-based Radiostereometric Analysis Using Intensity-based 2D/3D Registration Pipeline: Feasibility Study

Ondrej Klima[a,*], Adam Chromy[b], Petr Kleparnik[a], Michal Spanel[a], Pavel Zemcik[a]

[a]*Brno University of Technology, Faculty of Information Technology, IT4Innovations Centre of Excellence, Božetěchova 1/2, 612 66 Brno, Czech Republic*
[b]*Brno University of Technology, Faculty of Electrical Engineering and Communication, Technicka 3082/12, 616 00 Brno, Czech Republic*

Abstract

Model-based radiostereometric analysis (MBRSA) methods exploit, in most cases, feature-based 2D/3D registration. In this paper we focus on a feasibilty of the intensity-based 2D/3D registration approach applied in MBRSA. To evaluate the feasibility, we created a data set containing stereo pairs of both synthetic and real radiographic images of a metallic radius bone implant. Evaluation, we performed and present, reveals sufficient accuracy of the intensity-based registration pipeline and its robustness to image artifacts. The results obtained using synthetic radiographs show comparable accuracy with the feature-based non-overlapping area (NOA) approach. The registration process using real X-Ray images did not require preprocessing of the input radiographs neither was significantly affected by the presence of the metallic bone screws. This study presents an introductory part of an ongoing research.

Keywords: Radiostereometric analysis, Implant migration, 2D/3D registration, Levenberg-Marquardt optimization

1. Introduction and Related Work

Radiostereometric Analysis (RSA), proposed by Selvik [1], is a method for monitoring the fixation of an implant within a bone [2]. The analysis allows an identification of a micro-motion between the implant and the bone. The RSA method is indicated especially in cases of total joint replacement, such as total knee (TKA) or total hip (THA) arthroplasty [3, 4, 5]. The method is based on pairs of stereo radiographs. The traditional approach exploits tantalum markers injected into the bone together with markers attached to the implant. The tantalum markers are shown in Figure 1. Markers positions in the 3D space are obtained by triangulation from the radiographic stereo pair. The patient commonly undergoes several following-up examinations during the two years after the intervention [6]. The implant migration is revealed when the relative pose between the bone markers and the implant markers differs among the examinations.

However, several issues exist related with attaching markers to the implant. The markers attached to the implant may be occluded in the radiographs by the implant itself, the marked implants are significantly more expensive, and moreover, the implants may be weakened by the markers [2]. These issues are addressed by the model-based radiostereometric analysis (MBRSA), proposed first by Valstar [7]. The method is based on a 2D/3D registration of a virtual implant model into the radiographic stereo pair instead of marking the implant, while the bone markers remain involved.

*Corresponding author. Tel.: +420 54114-1402
Email address: iklima@fit.vutbr.cz (Ondrej Klima)

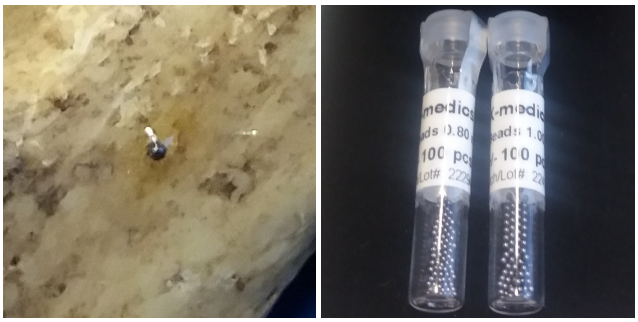


Figure 1: A tantalum bead of 0.8 mm in diameter glued to a dry cadaveric bone (left). A packing of two hundred tantalum markers (right).

Most of the model-based RSA methods depend on *feature-based* registration, exploiting the edges detected in the radiographs [7, 2, 4]. We have recently proposed an *intensity-based* method for the 2D/3D registration of a bone atlas into the X-Ray images [8]. The main goal of this preliminary study is to verify that the intensity-based registration is feasible in terms of the model-based radiostereometric analysis using the previously proposed approach.

2. Intensity-based Registration Method

The aim of the registration is to recover an accurate pose of the implant model within the 3D space of the stereo radiographic pair. A rough initial pose estimate provided by a user is required. The proposed method is designed for binary images.

The registration is performed as an iterative optimization. Digitally reconstructed radiographs (DRR) are rendered from

the implant model in current pose in each iteration. A similarity between the DRR images and the radiographs is evaluated using the mean square error (MSE) measure. The rotation and translation of the model are consequently adjusted to minimize the differences between the real and the virtual radiographs. The resulting pose of the implant model is obtained when the dissimilarities are minimal.

The registration is formulated as a non-linear least squares (NLS) problem:

$$p^* = \arg\min_p \frac{1}{N} F(p)^T F(p) \tag{1}$$

where $p = (R, T)$ is a pose vector formed by the rotation and translation of the implant model, N is a count of pixels contained in the radiographs and $F(p)$ is a vector of residuals between the original X-Ray and DRR images:

$$F(p) = \begin{pmatrix} \text{vec}(\text{DRR}_{AP}\ (x) - \text{XRay}_{AP}\ ) \\ \text{vec}(\text{DRR}_{LAT}(x) - \text{XRay}_{LAT}) \end{pmatrix} \tag{2}$$

where $|F(p)| = N$. The optimization is solved using the Levenberg-Marquardt algorithm [9], which is a highly effective method in terms of 2D/3D registration [10]. The accuracy of the method strongly benefits from the optimization on a pixel level [8]. The optimization can be interpreted as a minimization of a non-overlapping area (NOA) between the real and virtual radiographs, similarly to the approach proposed by Valstar [7].

3. Experimental Results and Discussion

3.1. Synthetic Radiographs

The accuracy of the intensity-based method was evaluated using synthetic radiographs. The radiographs were generated as binary images from a polygonal model of a radius bone implant consisting of 71,689 vertices and 143,762 faces. The virtual X-Ray images were of size 849×206 and 873×277 pixels respectively with horizontal and vertical pixel spacing equal to 0.143 mm. Consequently, the vector of residuals $F(p)$ was formed by 416,715 elements. The registration was repeated 10 times, initialized with various randomly generated pose estimates. The differences between the initial and the ground-truth poses were limited to ±9 mm in translation and ±9 ° in rotation. The registration took 55 iterations on average.

Table 1: Distributions of translational errors (n = 10).

	T_x [mm]	T_y [mm]	T_z [mm]
Mean	0.033	-0.010	-0.068
Std. Dev.	0.139	0.013	0.106

Table 2: Distributions of rotational errors (n = 10).

	R_x [°]	R_y [°]	R_z [°]
Mean	-0.107	-0.406	-0.009
Std. Dev.	0.133	0.566	0.231

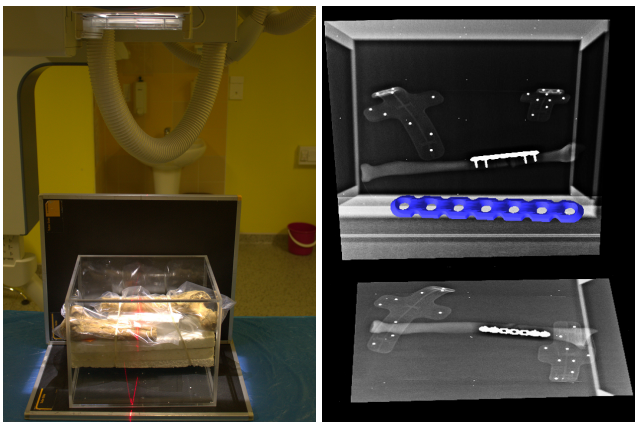


Figure 2: Illustration of the experimental setup placed in the X-Ray machine (left). The actual calibrated radiographs capturing the bone model with attached implant (right). In front of the radiographs there is a virtual model of the implant, highlighted by a blue color.

The results are shown in Table 1 and Table 2 as translational and rotational error distributions with respect to individual axis. The highest rotational and the lowest translational errors occurred in case of the y axis, which corresponds to the longitudinal axis of the implant model. This is an expected result as the virtual X-Ray images were taken approximately in xy and yz planes in the space of the implant model.

Table 3 shows comparison between the intensity-based registration pipeline and the feature-based NOA approach. The intensity-based method has slightly larger rotational error, but on the other hand is more accurate in translation. However, this comparison is rather tentative according to a different nature of evaluation data sets and a different kind of involved implants. The radius bone implant is not significantly asymmetric in comparison with implants dedicated for THA and TKA interventions.

Table 3: The largest standard deviations for translation and rotation.

	T [mm]	R [°]
NOA (Valstar [7])	0.221	**0.524**
Proposed approach	**0.139**	0.566

3.2. Real Radiographs

An illustration of the experimental setup is shown in Figure 2 left. The radiographs were taken using the X-Ray cassettes of size 35×43 cm and calibrated using a custom made biplanar RSA cage. The captured object is placed on the Styrofoam inside the Plexiglas calibration cage. For the real-world evaluation of the method we used a plastic model of a fractured radius bone with a metallic bone plate. The bone plate was attached to the plastic model using metalic screws. Radiographs of the model were taken serially from anterior-posterior and lateral views and calibrated using the DLT approach [11]. The actual radiographs are shown in Figure 2 right.

A rough initial pose of the implant model in the 3D space was set interactively. The initial pose is shown in detail in Fig-

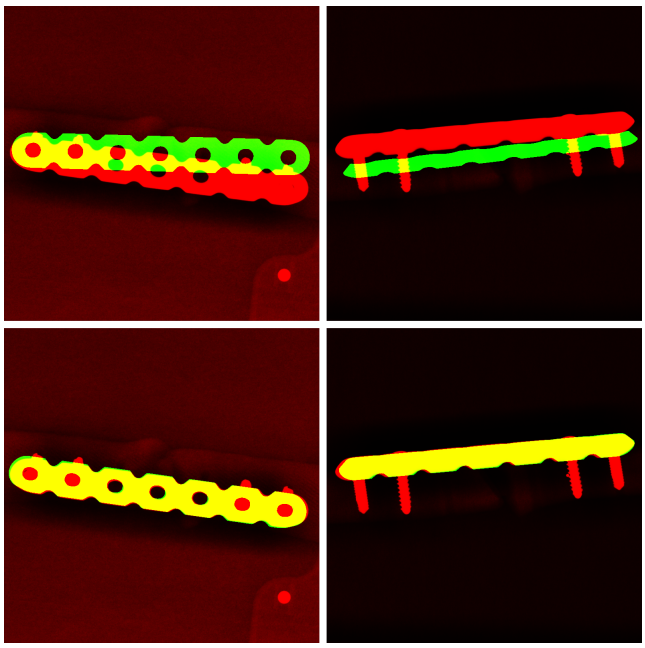


Figure 3: Initial pose of the implant provided by the user (top). The resulting pose recovered by the proposed 2D/3D registration method (bottom).

ure 3 in the top row. The original radiographs are visualized by a red color and overlayed by green shaded DRR images rendered from the implant model. The overlapping area is emphasized by a yellow color.

Although the method is dedicated for binary images, the original radiographs were used for the registration instead. This was possible due to high contrast between the metallic bone plate and the surroundings. The registration process took 72 iterations, the final result of the registration is shown in Figure 3 in the bottom row. It can be seen that the registration did not fail due to lower contrast surroundings, nor due to the high contrast bone screws which are not a part of the implant model.

4. Conclusion

We have verified that the intensity-based 2D/3D registration is clearly feasible in terms of model-based radiostereometric analysis. Moreover, the radiographs are suitable for the registration without further processing. The quantitative evaluation based on synthetic X-Ray images revealed that the intensity-based method and the feature-based non-overlapping area approach have tentatively comparable accuracy. The currently ongoing work is focused on the real-world accuracy evaluation using the implants dedicated for the THA and TKA interventions. The registration pipeline is publicly available at http://www.fit.-vutbr.cz/~iklima/prods.php?id=458 and http://www .fit.vutbr.cz/ ~iklima /prods.php?id=505.

ACKNOWLEDGEMENTS

This work has been supported by the Internal Science Fund of Brno University of Technology grant no. FEKT/FIT-J-17-4745 Intermodal Registration of 3D Data in Health-care and by the Technology Agency of the Czech Republic grant no. TA04011606 Modern Image Processing Techniques and Computer-based Planning in Trauma Treatment. The authors thank the 3Dim Laboratory s.r.o. for providing the radius bone plate and its virtual model and the University Hospital in Ostrava for capturing the experimental radiographs.

References

[1] Selvik, G.. Roentgen stereophotogrammetric analysis. Acta Radiologica 1990;31(2):113–126. URL http://dx.doi.org/10.1177/028418519003100201.

[2] Kaptein, B., Valstar, E., Stoel, B., Rozing, P., Reiber, J.. A new model-based rsa method validated using cad models and models from reversed engineering. Journal of biomechanics 2003;36(6):873–882.

[3] Madanat, R., Mäkinen, T.J., Aro, H.T., Bragdon, C., Malchau, H.. Adherence of hip and knee arthroplasty studies to rsa standardization guidelines: A systematic review. Acta Orthopaedica 2014;85(5):447–455. URL http://doi.org/10.3109/17453674.2014.934187.

[4] Nazari-Farsani, S., Finnilä, S., Moritz, N., Mattilla, K., Alm, J., Aro, H.. Is model-based radiostereometric analysis suitable for clinical trials of a cementless tapered wedge femoral stem? Clinical Orthopaedics and Related Research 2016;474:2246–2253.

[5] Shareghi, B., Johanson, P.E., Kärrholm, J.. Clinical evaluation of model-based radiostereometric analysis to measure femoral head penetration and cup migration in four different cup designs. Journal of Orthopaedic Research 2017;URL http://dx.doi.org/10.1002/jor.23177.

[6] Valstar, E.R., Gill, R., Ryd, L., Flivik, G., Börlin, N., Kärrholm, J.. Guidelines for standardization of radiostereometry (rsa) of implants. Acta Orthopaedica 2005;76(4):563–572. URL http://dx.doi.org/10.1080/17453670510041574.

[7] Valstar, E., de Jong, F., Vrooman, H., P.M., R., J.H.C., R.. Model-based roentgen stereophotogrammetry of orthopaedic implants. Journal of Biomechanics 2001;34(6):715–722.

[8] Klima, O., Kleparnik, P., Spanel, M., Zemcik, P.. Intensity-based femoral atlas 2d/3d registration using levenberg-marquardt optimisation. Proc SPIE 2016;9788:97880F–97880F–12. URL http://dx.doi.org/10.1117/12.2216529.

[9] Kelley, C.T.. Iterative methods for optimization. Frontiers in applied mathematics; Philadelphia: SIAM; 1999. ISBN 0-89871-433-8. URL http://opac.inria.fr/record=b1096699.

[10] Klima, O., Chromy, A., Zemcik, P., Spanel, M., Kleparnik, P.. A study on performace of levenberg-marquardt and cma-es optimization methods for atlas-based 2d/3d reconstruction. IFAC-PapersOnLine 2016;49(25):121 – 126. URL http://dx.doi.org/10.1016/j.ifacol.2016.12.021.

[11] Choo, A.M., Oxland, T.R.. Improved rsa accuracy with dlt and balanced calibration marker distributions with an assessment of initial-calibration. Journal of Biomechanics 2003;36(2):259 – 264. URL http://dx.doi.org/10.1016/S0021-9290(02)00361-5.

Maya2CellVIEW: Integrated Tool for Creating Large and Complex Molecular Scenes

David Kouřil

TU Wien, Vienna, Austria

Mathieu Le Muzic

TU Wien, Vienna, Austria

Barbora Kozlíková

Masaryk University, Brno, Czech Republic

Ivan Viola

TU Wien, Vienna, Austria

Abstract

Scientific illustrators communicate the cutting edge of research through their illustrations. There are numerous software tools that assist them with this job. They often use professional modeling and animation 3D programs which are primarily used in games and movie industry. Because of that, however, these tools are not suitable for scientific illustrations out of the box. This work focuses on visualization of structures and processes in biology, aiming mostly to scales of nano- to micrometers. At this scale we often do not gain much by using hyper-realistic rendering styles that the professional software aims for. Instead, we want to employ more simplified style which helps to communicate the important story without losing too much detail or scientific precision. The aim of our proposed system is to push abilities of illustrators working with large-scale molecular scenes. This is done by connecting two existing software packages, Maya and cellVIEW. This allows to combine the real-time rendering possibilities of cellVIEW, and modeling and animation tools of Maya, which results in more effective and efficient workflow.

Keywords: visualization, molecular visualization, illustration

1. Introduction

In this day and age, scientists come up with new findings almost every day. Unfortunately, not all of these are ever shown to the general public. There can be several reasons for that. New discovered facts are usually pieces of a bigger picture. Also, all the information might be already available, spread over several databases, but putting them all together would take significant effort and time. On top of that, scientists are not usually trained and encouraged to expose their results to the public audience.

This is the job of scientific illustrators. These people are, first and foremost, experts in their fields, but they are also investing a significant amount of time on acquiring and perfecting their artistic skills. They use these skills to visualize the science outcomes in their domain using easily understandable images, animations, or other forms of media.

This paper focuses on visualization of structure and function of objects in cell and molecular biology. To show examples of such visualizations, we can point to works of Drew Berry [1], Graham Johnson [2], or Janet Iwasa [3]. The importance of their work lies not only in bringing the science to the laymen. Humans are visual beings and by seeing something they can understand certain concepts more deeply or differently. This applies to other scientists as well. In practice this means that scientific illustrations can serve also as initiators of discussions. New ideas, hypotheses, and experiments might emerge just by seeing concepts differently or as a compilation of information into one cohesive artwork.

With the increasing popularity of computer-generated imagery, it has been naturally adopted by scientific illustrators as well. Tools have become more accessible and easier to use over the years. Today, software solutions like Maya, Cinema4D, or 3D Studio Max have become industry standards for any task that revolves around modeling 3D geometry and its rendering. Game and movie industry are the leading fields of industry that push computer graphics software creators forward and provide most of the revenue for them. This means that these tools, no matter how versatile they try to be, are being skewed towards the use cases in movies and games. In consequence, people who want to create scientific content might sometimes struggle to use these tools. Still, illustrators have been already able to create amazing images and movies showing audience phenomena from all science disciplines.

CellVIEW [4] is a tool with the ability to render large biological macromolecular scenes at interactive frame-rates. It has been designed and implemented with regards to this use case

and it utilizes state-of-the-art rendering techniques. It employs several modern algorithms to reduce the amount of processed geometry in macromolecular scenes to provide its users with real-time performance. As a result, cellVIEW can render scenes containing up to several billion atoms with a frame-rate above 60 FPS. CellVIEW has been implemented using the Unity game engine. The rendering style has been inspired by illustrations by David Goodsell[1] who has developed his own style of abstracting the shape of individual proteins to reduce visual noise in the picture. CellVIEW imitates this approach by incorporating a level-of-detail scheme. The farther the protein is from the camera, the less amount of its atoms is rendered and the rendered atoms are scaled up. This approach results in a multi-scale visualization—the user can zoom in to see individual atoms of a protein, or can zoom out and see the whole dataset with its distinguishable compartments. The biggest dataset that has been visualized using cellVIEW is human immunodeficiency virus (HIV). However, performance tests that have been performed indicate that larger datasets (e.g., Escherichia coli bacterium) should be possible to render using cellVIEW as well.

2. Method

The actual workflow obviously differs from illustrator to illustrator. They use different software, different plug-ins, and, most importantly, they have different data and project goals.

It is however important to define the general overview of the workflow. Our goal is to connect one of the specific domain tools with a professional software. By doing that, we want to achieve faster and therefore more effective workflow.

We consider the pipeline to be composed of two major steps: modeling and rendering. In the modeling step, all the data, requirements, hypothesis, ideas, and stories are compiled into a 3D scene or animation. Artists usually use software-specific features like particle systems or physics simulations to get there.

The next step is the generation of either a still image or an animated video from the 3D scene/animation. This is equally, maybe even more, important as the first step. By using certain rendering techniques we can underline concepts which are important to the artwork. The level of detail has to be carefully chosen not to overwhelm the audience with visual noise. Traditionally, the rendering step has been very computationally intensive. Today with the power of modern GPUs and the increasing availability of such devices, the rendering times have been reduced dramatically. Still, this represents one of the main limitations in the workflow of an illustrator or animator. Even just one minute delay can cause distraction.

This is the problem that we wanted to solve by using custom state-of-the-art molecular renderer. As it was mentioned before, at this domain, hyper-realistic results of modern rendering methods are not that beneficial. Instead, we want to employ more illustrative, simplified rendering styles. That brings us another benefit that such rendering method has better performance, allowing us to render at interactive (real-time) framerates. By using a fast renderer, we want to eliminate the time cost of the rendering step when using conventional off-line renderers. The difference between the previous and ours approach is illustrated in Figure 1.

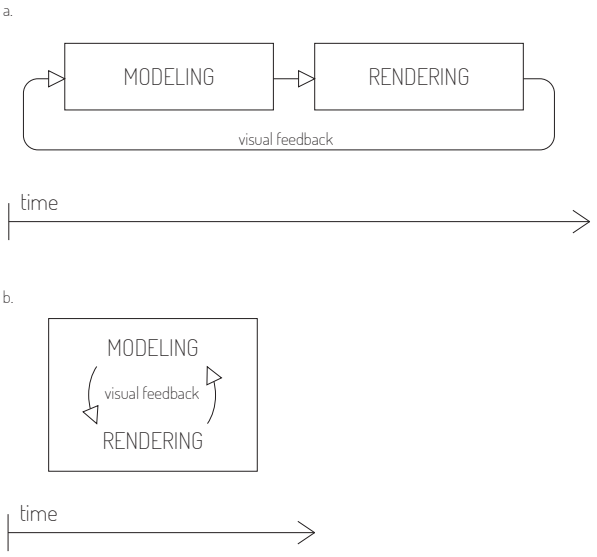


Figure 1: Workflow, a. traditional approach with off-line rendering, b. our approach with real-time rendering. By using more simple, faster rendering method, we eliminate the rendering step completely

Ultimately, we operate with two programs and we want to use some functionality from the first one and other functionality from the second one. The naive solution would be to implement our desired features into the software tool that lacks them. In our case, that would mean we could either implement a fast and visually appealing renderer into the modeling software tool, or we could do the opposite and implement the desired modeling tools into the renderer program.

The problem with the first approach is that substantial percentage of the 3D software packages that artists use the most are commercial solutions with closed source code. It is possible to extend them via API that they provide but that is not enough if we want to implement a state-of-the-art technique that requires the latest technology in terms of graphics API.

We also do not want to implement the desired 3D modeling and animation tools into a specific molecular visualization program. This would create development overhead which would not bring much benefit on its own. Besides that, every artist uses different instruments (different software, shortcuts, additional plug-ins, etc.) and pleasing all of them would be an impossible goal to achieve.

Instead, we came with an alternative solution to this problem. We do not want to re-implement from scratch something that is already available to use. Instead, we went for a different approach and tried to connect the two tools in such way that would allow us to use both sets of features at the same time.

[1] Examples of Goodsell's work can be found on his webpage `http://mgl.scripps.edu/people/goodsell/illustration/public`

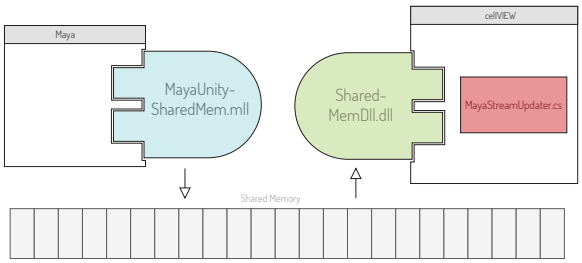


Figure 2: Overview of system components.

We do this by using both programs and establishing a communication channel between them.

We use writing and reading from shared memory to accomplish this communication. In our case the communication is one-way. We can model or animate the scene in one program and transfer the data describing this scene to have it rendered in the second program. As we will show, this can be done in a straightforward fashion as our scenes can be simplified to a set of descriptors defining each molecule.

On the side of the modeling program, it is sufficient to see only an approximation of the scene. One way would be to use a simple geometry, like cube or sphere, as a placeholder for each molecule. Such placeholder would then serve as an object that we manipulate with instead of an actual molecule. However, we came with a different approach. With Molecular Maya plug-in, we can load a molecule description in the form of a PDB file. With this plug-in it is also possible to generate a polygonal surface mesh representation with adjustable detail level. Therefore, we load a molecule and generate a very low resolution mesh which serves as our placeholder.

3. Implementation

The architecture of the system consists of three main parts—the plug-in for Maya, the plug-in for Unity, and the Unity script—as shown in Figure 2. The data flows only in one way—we write into shared memory with Maya plug-in and read from shared memory with Unity plug-in. This simplifies the situation from the implementation point of view because we do not have to design any synchronization scheme.

The plug-in for Maya is using Maya API and is written in C++ programming language. The function of this plug-in is to parse the 3D scene, look for all molecular objects, and output their positions and rotations, along with an information about the type of the instance, into the shared memory (Figure 3).

The Unity side of the system consists of two parts—a C++ plug-in and a C# script. The C++ plug-in takes care of reading the data from the shared memory, while the C# script, which is an intergral part of cellVIEW, receives this data and uses it to render the final molecular scene.

Note that there are two types of interoperability between the components: (1) shared memory functionality, which enables two processes to communicate, and (2) interoperability between C++ and C#, which allows us to pass memory addresses from the plug-in to the script on the Unity side.

3.1. Shared Memory

Shared memory is a segment of system memory which can be accessed by multiple programs. It is used as an efficient way to establish communication between separately running processes. Our implementation has been done for Windows operating system, however, the concept of shared memory can be found in other operating systems as well.

From now on, we will be talking about the implementation that has been done for Windows. There are several ways to access shared memory there. The C++ library *boost* provides a class that establishes an abstraction above shared memory functionality. Similarly, the *Qt* framework also has a class with comparable function set. We chose to not use any of these. Instead, we directly used *Windows API* function calls to operate with shared memory. This solution has been chosen because we wanted to use the lowest possible layer due to performance concerns. In consequence, this means that our implementation is tied to Windows platform only. Porting to other platforms should however be straightforward either by using operating system specific calls or by using one of the mentioned libraries. To use the shared memory between two processes on the Windows platform, the memory-mapped files system was utilized.

3.2. C++ and C# interoperability

Unity engine is written with a combination of C++ and C#/.NET. The .NET API is exposed to users which enables them to write scripts in either C# or JavaScript. These scripts are used to implement a gameplay or other behaviour.

Managed code is a code which runs under Common Language Runtime (CLR) virtual machine. *Unmanaged code* is any other code which does not need CLR but runs directly on the hardware instead.

C# and .NET framework have been designed with interoperability in mind. This means that programmers can reuse their

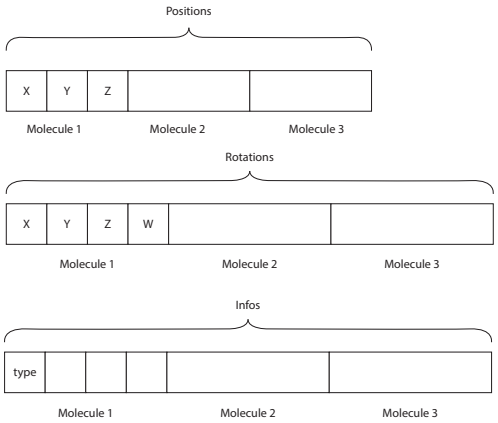


Figure 3: Shared memory data layout

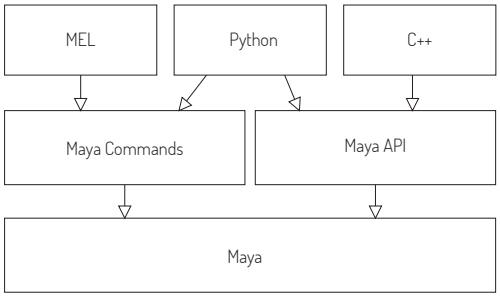


Figure 4: Maya programming ecosystem.

code written in other languages. There are several types of interoperability but we are interested in calling an unmanaged (C++) code from a Unity script written in C#.

We use a feature called Platform Invoke. Platform Invoke enables managed (C#) code to call unmanaged functions implemented in dynamic link libraries (DLLs). The process of transforming arguments from types in native code into equivalent types in managed code is called *marshalling*.

3.3. Maya API

There are two ways how one can extend Maya's functionality: scripts and plug-ins. The technologies which can be used to do so and their mutual relations are shown in Figure 4.

Scripting can be done using either MEL or Python scripting language and it basically provides an alternative to performing actions via GUI. Everything you can do by clicking in GUI, you can do by typing commands through Maya's command line. Longer scripts can be written and run through the built-in editor. Scripts are most commonly used for automation.

The second way to extend Maya is by creating plug-ins which use Maya API, either in C++ or Python. There are several types of plug-ins that the users can create but typically these are either implementation of new custom MEL command, or implementation of a custom Dependency Graph (internal scene data structure) node type.

It should be noted that both scripts and plug-ins are supposed to work together inside the Maya ecosystem. Different means and programming languages should be chosen accordingly to the project.

3.4. MEL vs. Python vs. C++ in Maya

MEL (Maya Embedded Language) is a scripting language similar to other scripting languages like Bash or Perl. It is generally the fastest way to customize Maya in any way. For any more complex tasks it is generally better to use Python scripting or even turn the functionality into a plug-in.

Maya API can be thought of as a level directly under MEL scripting. With Maya API, we can create several types of plug-ins but the two most common ones are new custom commands which can then be called with MEL, and a custom Dependency Graph nodes. Plug-ins using Maya API can be implemented either in Python or C++.

Python is a powerful and easy-to-learn scripting language. Its advantage is that it is interpreted which means that there is no need for a compilation step. This is beneficial for developers in the phase of prototyping because they can make changes more quickly. The disadvantage is also a consequence of Python being interpreted—Python is expected to be slower than most compiled languages like C++. We wanted to implement the functionality which works in real-time. For this reason we decided that working in Python was not the ideal approach in this project.

C++ Maya API was our choice for creating our custom plug-in. It has been chosen primarily because we wanted to get as much performance as we can. But there is another reason for using C++. By doing so, we can easily call the operating system API to issue function calls that work with the shared memory.

4. Demonstration

To demonstrate the solution, two artificial molecular scenes have been created. Both examples can be seen in Figure 5. The first scene contains instances of several protein types typically found in blood plasma. The instances are positioned and rotated randomly using Maya's scripting interface.

Second scene has been created by instancing proteins along a linear curve, showing the possibility to use Maya's internal tools to create strand-like structures.

5. Conclusion

From the practical point of view, the main goal of this work was to investigate, how the state-of-the-art renderer cell-VIEW could be integrated into the modern professional 3D software Maya.

This has been successfully achieved by employing the shared memory to establish communication between these two programs. Naturally, many aspects of the system could be improved upon. A closer collaboration with an illustrator would be extremely helpful at this stage. Additional design of the system should be based on an actual scientific scene. It would be interesting to see how the system fits into a working pipeline.

The pipeline does not have to end with just two programs communicating together. Some further processing might be performed after the image is rendered with cellVIEW, for example, compositing or editing with tool like Adobe After Effects.

Experts from the domain of biological visualization are always eager to tell their stories and express their ideas. It is very common that they want to integrate interactivity in this process. Hopefully, the system presented in this paper will be expanded and fulfill the ultimate goal of helping illustrators and animators with the process of communicating the science to a broad audience.

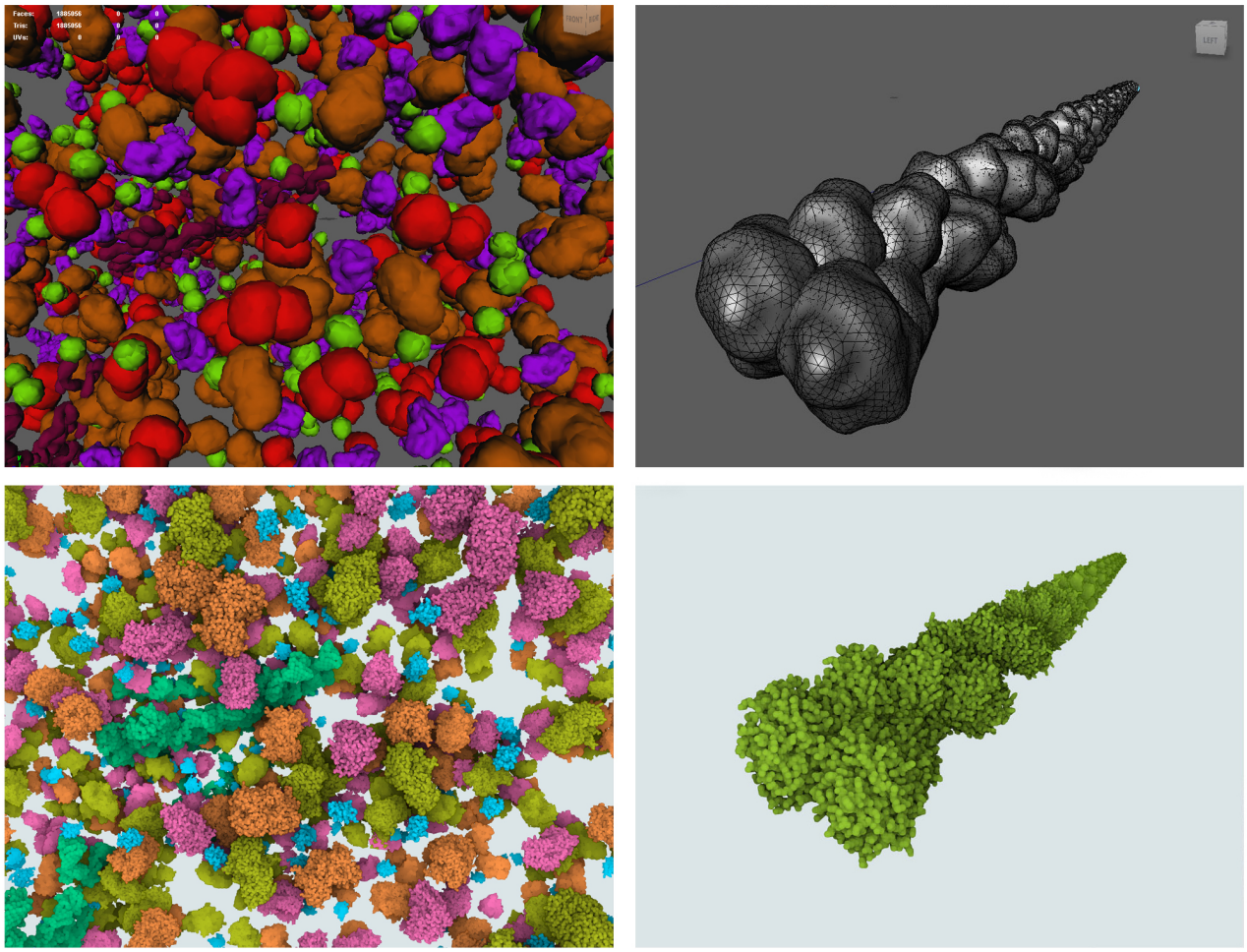


Figure 5: Demonstration on artificial scenes. Left column shows the scene with blood proteins, right column the scene with linear strand. Top row: screenshots of the scenes modeled in Maya using mesh placeholders, bottom row: final frames rendered in atomic detail with the cellVIEW.

References

[1] Berry, D.. Molecular animations. 1995-2017. URL `http://www.molecularmovies.com/movies/viewanimatorstudio/drew%20berry/`.

[2] Johnson, G., Noske, A.B., Marsh, B.J.. Rapid visual inventory and comparison of complex 3d structures. 2011. URL `https://www.youtube.com/watch?v=Dl1ufW3cj4g`.

[3] Iwasa, J.H.. Animating the model figure. Trends in cell biology 2010;20(12):699–704.

[4] Muzic, M.L., Autin, L., Parulek, J., Viola, I.. cellview: a tool for illustrative and multi-scale rendering of large biomolecular datasets. In: Bühler, K., Linsen, L., John, N.W., editors. Eurographics Workshop on Visual Computing for Biology and Medicine. EG Digital Library; The Eurographics Association. ISBN 978-3-905674-82-8; 2015, p. 61–70. URL `https://www.cg.tuwien.ac.at/research/publications/2015/cellVIEW_2015/`.

A Visual System for Detecting the Occupancy of a Car Park by Deep Neural Networks

Vaclav Stransky[a,b,1], Jaroslav Rozman[a,1], David Herman[b,1], David Hlavon[b,1], Adam Babinec[a,b,1]

[a] *Faculty of Information Technology, Brno University of Technology, Czech Republic*
[b] *RCE Systems, s.r.o., Brno, Czech Republic*

Abstract

The concept of smart cities is inherently connected with efficient parking solutions based on the knowledge of individual parking space occupancy. The subject of this paper is the design and implementation of a robust system for analysing parking space occupancy from a multi-camera system with the possibility of visual overlap between cameras. The system is designed and implemented in Robotic operating system (ROS) and its core consists of two separate classifiers. The more successful, however, a slower option is detection by a deep neural network. A quick integration is provided by a less accurate classifier of movement with a background model. The system is capable of working in real time on a graphic card as well as on a processor. The success rate of the system on a testing data set from real operation exceeds 93%. Smart parking concepts inherently include efficient parking solutions based on the knowledge of the occupancy of individual parking spaces. This paper describes just such a system that allows easy orientation in the parking lot, both for management and for the driver.

Keywords: occupancy of a car park, vehicle detection, deep neural networks, background model

1. Introduction

With a continuously growing number of cars, there are inconvenient difficulties with parking them. Although there are still vacant parking spaces in a car park, it is often complicated for the drivers to find these spaces in a large car park. A solution can be seen in applications that mark the occupancy of individual parking spaces and inform the drivers about their location on information boards or through another visualization method. The development of such an application is the subject of this paper.

The described application is developed from square one in C++ and Python programming languages using the existing libraries OpenCV [1] and QT4 [2]. The main focus is on the classification of parking space occupancy itself and on the functioning of the system as a whole. Additional functions include recording of detected changes, creating graphs from obtained data and additional information for the users. The requirement is for the final application to work non-stop and in real time using inexpensive hardware equipment. The classification must work in all weathers and in changing light conditions. The solution should be general but the neural network can be trained for a specific car park.

The used systems currently consist of sensors based on microwaves [3], on magnetic field [4][5] or on ultrasound [6][7]. Since each parking space requires its own physical sensor, the installation of the system as well as the subsequent maintenance is rather demanding. If the operator wished to change the layout of parking spaces, it would also be necessary to move the respective sensors. Therefore, applications detecting the occupancy of parking spaces using computer vision have been coming into existence lately. Compared to sensor systems, the installation is much easier, especially the layout of the spots as well as maintenance. Moreover, a camera system enables a live overview of the scene and associated extended usage (identifying the type of vehicle, colour etc.) However, their success rate is incomparable to sensor systems. Monitoring with cameras also brings a number of disadvantages, such as legislation problems and the necessity to inform the car park users. Detection gets more difficult in bad weather or at night with insufficient artificial lighting. If a foreign object gets into the view of the camera or if the camera tilts (e.g. due to wind), false detections can occur. A comparison of the existing methods is summarized in table 1.

Table 1: Comparison of existing solutions

Method	Success rate	Demands of installation	Demands of maintenance
Microwaves	95%	high	high
Magnetic field	>99%	high	medium
Ultrasound	98-99%	high	medium
Computer vision	85-98%	medium	low

The existing methods using computer vision use SVM [8], image segmentation [9], gradient histogram [10] or cascade de-

Email addresses: xstran16@stud.fit.vutbr.cz, vaclav.stransky@rcesystems.cz (Vaclav Stransky), rozmanj@fit.vutbr.cz (Jaroslav Rozman), david.herman@rcesystems.cz (David Herman), david.hlavon@rcesystems.cz (David Hlavon), ibabinec@fit.vutbr.cz, adam.babinec@rcesystems.cz (Adam Babinec)

tector [11]. Comparing the success rate of the listed methods is complicated since each author used a different testing set. The success rate of these methods given by the authors ranges from 85% to 95%, however, data sets most often do not include occlusions or low vision. Higher success rate is achieved by methods using deep neural networks [12][13], where the success rate exceeds 95%. The main shortcoming is the speed of classification, which the authors give as 15 seconds for the entire car park.

The system described in this paper is designed and implemented in a Robot operating system [14], thanks to which it is divided into several separate nodes. The main nodes are: processing images from camera, filtering the images, transforming the images, local classifiers, a global classifier and presentation of results. Images are filtered based on a comparison of histograms of correct and faulty images. To achieve a better detection, undesirable transformations, distortions[1] and perspective[2] are removed from the image. An image prepared in this way is used for parking space occupancy classification using two classifiers. The main focus is on classification using machine learning, which uses deep convolutional neural network. Detection of arrival and departure is supplemented by a classifier based on a background model. The results of both classifiers are merged and published. If the system consists of more cameras, the images from each camera are processed independently and the results of partial classifiers are merged all at once.

The introduced solution is interesting thanks to its high success rate, easier installation and the ability to classify the occupancy of hundreds of parking spaces in real time. The system is not flawless, in some cases, there is faulty classification. However, this occurs in less than 7% of cases and with the gradual development of the application and the increase of the training set, there is less and less of them. The method most often fails in low visibility (at night, when the camera is wet with dew), due to changes of weather (rainfall, snow, sharp shadows) and in cases of extreme occlusion, when a car is partially or completely blocked by a larger object (lorry).

2. Design of architecture in ROS

The application is created in the environment of Robot operating system (ROS). The key advantage of ROS within this application is the possibility to divide the program into individual nodes (processes) that communicate with each other by messages (communication is based on TCP/IP). This allows reset of only one node in case of its failure, easy creation of various versions of the application based on requirements, transparency, fine-tuning by means of intercepting messages and suchlike. See the scheme 2 for a brief overview of the structure of the entire application. A defined group of nodes can be added more times, based on the number of cameras. The results of these groups are processed together and presented.

[1] When there is a distortion of the image, the transverse magnification is not the same across the entire image field, which disrupts the geometrical similarity of the object and its image.

[2] In a perspective image, the displayed objects seem to grow smaller and come together.

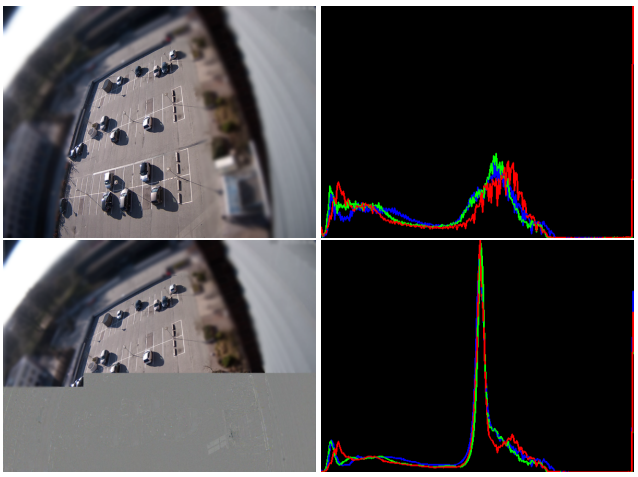


Figure 1: A correct and a faulty image with respective histograms

2.1. *Nodes of the system*

Processing images from the camera Images from camera are process by `Gscam` node already created in ROS. It receives video from IP camera transmitted by RTSP, checks the functionality of the camera and in case of its failure, renews the transmission of images. It transmits the processed images by means of an already created message, including additional information about the image and the status of the camera.

Filtering of images Another node compares the histogram of an incoming image with an average histogram of previous images (figure 1). If the histograms differ, the image is not sent and another image is selected. If the histograms are similar, the image is sent and the average histogram is updated. This way helps to remove faulty images from the camera and at the same time, to keep the images with a gradual change (e.g. the change of lighting). The node sends the images with a set frequency and discards redundant images.

Transformation of images For a more accurate classification, undesirable deformations are removed from the images. The next node removes distortion and perspective image so that all parking spaces in the image are of similar size. Samples of images without transformations and with removed deformations can be seen in figure 3.

Local classifiers A local classifier is one of the most important parts of the entire application. The design of the application enables a larger number of simpler local classifiers whose partial results are further processed and merged. During initialization, the node loads a configuration file that describes the layout of parking spaces, their shape and division into groups. The output of each local classifier is a set of soft decisions about the occupancy of the respective parking spaces. As a local classifier, practically any function deciding about the occupancy of parking spaces with a success rate of more than 50% can be used. Examples of successful linear classifiers include edge classifier, classifier based on a background model

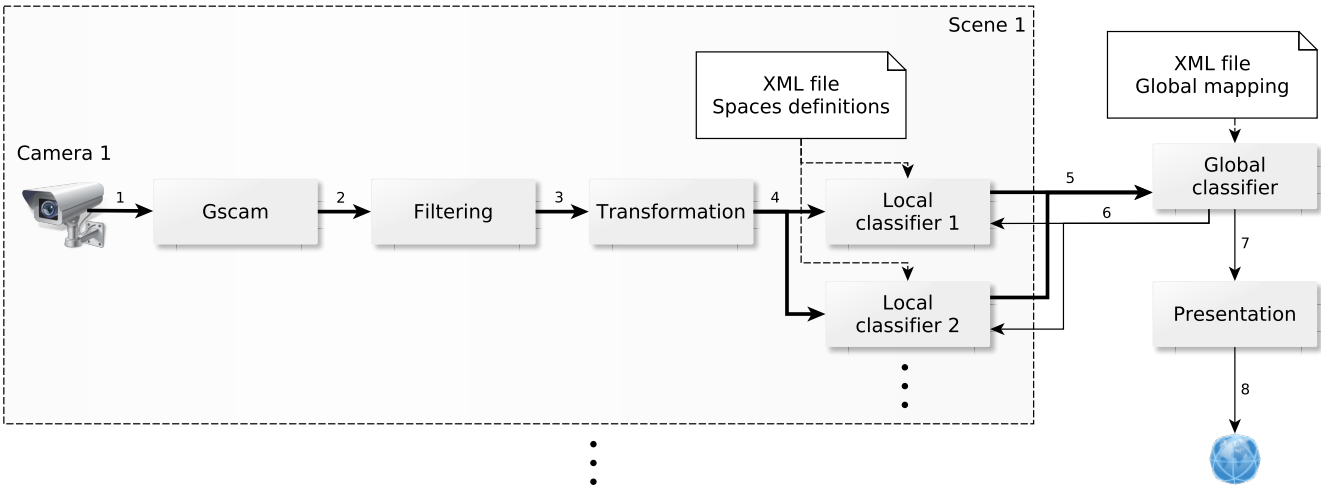


Figure 2: A brief scheme of the system design depicts the most important ROS nodes (1: rtsp, 2: raw images, 3: selected best images at a certain frequency, 4: images with removed deformations, 5: soft decisions about occupancy, 6: queries to check a specific location, 7: final decision about occupancy, 8: http)

Figure 3: Original image, image after removal of distortion and image after removal of perspective

and detection of moving vehicles or a classifier based on comparison of histograms. The current version of the described application contains two classifiers that are described in the following section.

Global classifier The results from local classifiers are merged in a global classifier that is common for the entire application. The selections include weighted average (original), neural network or SVM. The global classifier has two main functions. Firstly, it merges the soft decisions from local classifiers within one group, which means that it determines the final occupancy of parking spaces from one specific camera. Another function is the merging of decisions about occupancy of one parking space scanned by two cameras. The assignment of two or more markings to a specific parking space is done by a configuration file that maps the markings within each scene onto the global marking of a parking space.

Presentation of results The obtained results are published in several ways. For an immediate check of the correctness of the system, they can be sent via HTTP protocol and displayed on a web page. For a later analysis, they are saved into the database as a change in the occupancy of a specific space in current time.

3. Classification of parking space occupancy

For classification, two different classifiers are used and described here: classification using a deep neural network and classification using a background model.

3.1. Classification using a deep neural network

For work with neural network, we selected framework Caffe [15] that enables to carry out calculations on a processor or on a graphic card. For the design of the entire structure of neural network, Caffe uses Protobuffer serialization text format from Google. In Caffe, there are several interface varieties (command line, Python, MATLAB, C++), which allowed a simple connection to the developed system. Due to the success rate of existing and freely available networks, it would be counter-productive to design our own network; therefore, based on testing (section 5), the *GoogleNet* network was selected [16], which needs no further adjustments. In the described system, two output classes are used: a vacant or occupied space. The output of the network is the probability that the testing sample contains an occupied space. The network was trained for thirty epochs, out of which the best one was used: the fifteenth epoch. Validation accuracy starting from this epoch reached the value of 99.76%.

The design of the system enables classification of all parking spaces (using GPU) or classification by one space (using CPU). The classifier contains a planning queue with three priorities. The lowest priority is assigned to parking spaces where no change has been detected recently but it is suitable to check the evaluation. The highest priority is assigned to spaces where the movement model detected a change and it is necessary to find out the occupancy immediately. If the occupancy changes, a classification with medium priority is planned after a time defined by the user so that this decision can be checked.

3.2. Classification using a background model

The second classifier uses a background model for classification. When comparing a current image with the background model, it is possible to detect moving vehicles that are arriving at or leaving the monitored parking space. This method requires initialization assuming that some parking spaces are occupied. In this case, it is impossible to create the background model at the beginning and the principle must be reversed. Firstly, a model for the situation when a car is present is created; after the first major change, the space is classified as vacant and then it is possible to create a background model. In this case, the background model needs to be provided with information which parts of the image are to be updated (background) and which are to remain without updates (occupied spaces).

The main risks of this method lie in updating the background model. As opposed to monitoring travelling vehicles on a road, it is impossible to keep updating the background model for the entire time. During the period when the parking space is occupied, the background may change significantly (change of light conditions, snow on the road, shadow etc.). After the vehicle has left the space, the background model differs from the current image to such an extent that the space is still considered as occupied.

3.3. Merging the results of the classifiers

The application is designed so that it can run without problems on average hardware without a graphic card. On an average processor (see section 5), the classification of one parking space by a neural network takes one to two seconds, the classification of the entire car park with two hundred spaces thus takes several minutes. As opposed to that, processing the image with the second described classifier meets the condition of evaluating the image within one second. In reality, the system works in this way: the results from the background model are sent for a global decision, and in case a change has been detected, the global classifier asks the neural network about correctness. If the model does not detect any change, the neural network classifies all spaces one by one to check. The final decision is sent back to the classifiers and the background model is updated.

4. Marking of parking spaces

Since the system is developed for static cameras and car parks where the position of parking spaces does not change, it is possible to mark parking spaces before the start of the classification. Thus, detecting vehicles in an image becomes a simpler problem of classifying the occupancy of a specific space.

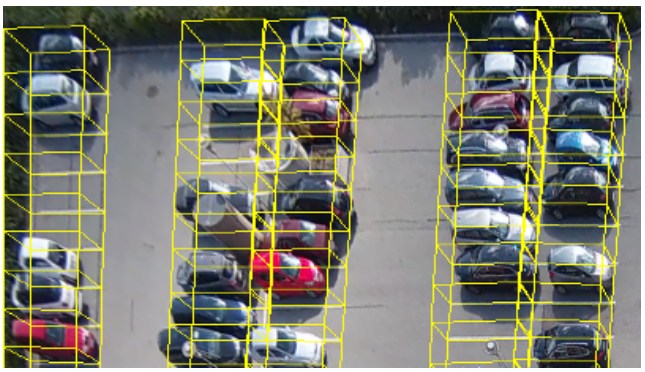


Figure 4: Marking of parking spaces

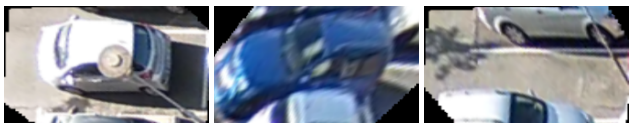


Figure 5: Examples of cut-out spaces (occupied spaces on the left and in the middle, vacant space on the right)

Since the camera is not placed vertically above the parking space, space marking in 3D is used. Firstly, the user marks the entire car park and enters the real dimensions, from which the placement of the camera can be calculated. Then, the user marks each parking space by four points and by height and the system calculates a 3D model of the parking space in the image. For an example of marking of several spaces see figure 4.

In deep neural network, the marking is used to cut out training samples (figure 5) and subsequently for the classification itself where the inputs are these segments. Moreover, for classification by a background model, a mask is applied on the cut-out parking space that eliminates the overlapping areas of adjacent parking spaces (figure 6).

5. Testing

The system was tested on data sets of total length of 32 hours from a real car park that was monitored by two cameras and consisted of 193 parking spaces. For an easy interpretation of the obtained data, also basic visualization elements were implemented, such as a graph of the occupancy history of the entire car park (see figure 7).

The success rate of the entire system depends predominantly on the success rate of the deep neural network, therefore, further tests dealt with neural networks. The first test examined the influence of removing deformation from the image on the classification success rate. For samples without removed deformations we obtained a classification success rate of 95.23% and 94.18% on two data sets, for samples with removed deformations, the success rate was 99.20% and 99.01%. For this

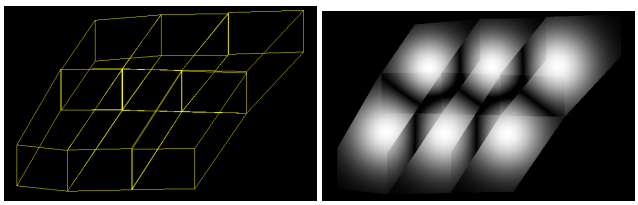


Figure 6: Composition of six masks of parking spaces

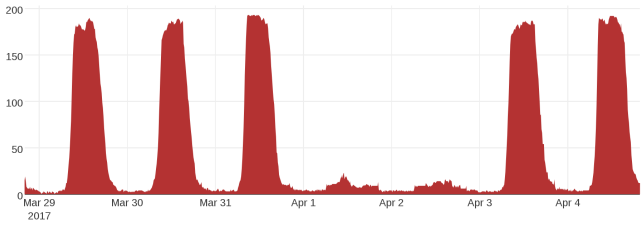


Figure 7: Graph of the occupancy history of the testing car park (from Wednesday to Tuesday)

reason, further tests dealt only with samples with removed deformations.

Two pre-trained architectures were selected whose success rate was the highest, and they were further adjusted for various input sizes. These architectures were a convolutional neural network *GoogleNet* with input sizes of 32x32, 64x64 and 224x224 pixels, and a residual neural network *Resnet* with input sizes of 256x256 pixels. (Networks such as *GoogneNet256* or *VGG16-32* were eliminated.) *GoogleNet* contains 22 layers with parameters (27 including merging layers), *ResNet* consists of 50 layers. The second test examined the success rate of the selected networks pre-trained and tested on samples with the best resolution that can be achieved in the system. We created a data set for final training that contained 7376 training samples (3791 occupied spaces and 3585 vacant spaces) and two different data sets for testing; the first one with 1881 samples (1181 occupied spaces and 700 vacant spaces) and the other one with 1802 samples (1020 occupied spaces and 782 vacant spaces). The sizes of samples ranged from 120x80 to 130x130. Data were obtained from two cameras (5mpx with 2592x1920 resolution) from the testing car park at various points in time, in changing light conditions and weather conditions on 25 different days. The training epoch that was selected was always the one with the best result on the testing set. Chart 2 summarizes the classification success rate and the speed of classification of one parking space on CPU (*Intel Core i7-4702MQ CPU @ 2.20GHz*) and on GPU (*GeForce GTX 1080*). Based on the results, the most successful network was selected and used in the application, namely *GoogleNet 224x224*; if a higher speed is needed, it would be more suitable to use *GoogleNet* with a smaller sample size.

The third test examined the influence of reducing and repeated enlargement of the size of both training and testing samples on the success rate of the two best networks from the previous test. The goal was to find out the minimum size of samples for which the system has the highest success rate. Based on the

Table 2: Comparison of deep neural networks

Name of network	Size of sample	Speed of CPU	Speed of GPU	Success rate of set 1	Success rate of set 2
GoogleNet	32x32	1.1s	3.7ms	98.83%	99.01%
GoogleNet	64x64	1.2s	5.3ms	99.10%	98.90%
GoogleNet	224x224	1.4s	11.7ms	99.64%	98.96%
Resnet	256x256	1.7s	13.3ms	97.18%	92.75%

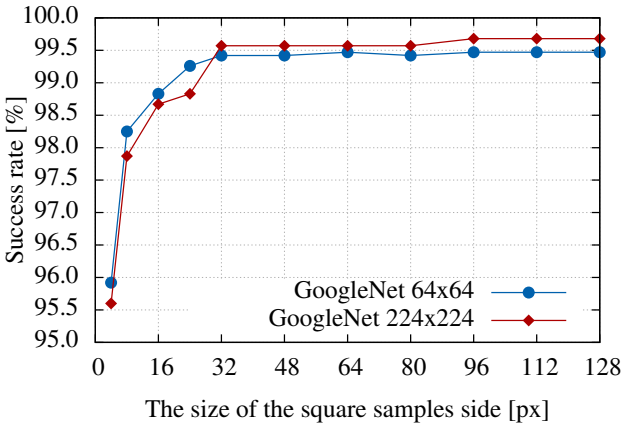


Figure 8: Dependence of success rate on samples size

obtained results, it is possible to reduce data flow from cameras to the central unit as much as possible; then the samples can be enlarged again in the central unit to the required input size of the network. From graph 8, it can be seen that it is possible to reduce the samples as much as to the size of 32x32 pixels, and the *GoogleNet 224x224* network will still have a success rate of more than 99.5%. In case of further reduction, the *GoogleNet 64x64* network becomes more successful. An interesting fact is a success rate of almost 96% for samples of the size of 4x4 pixels. Such a high success rate can be achieved primarily thanks to the fact that both the training and the testing sets were obtained from only one car park and it is therefore likely that the network is not general enough.

6. Conclusion

This paper described a system for classification of parking space occupancy and introduced two basic methods that are used: a deep neural network and a background model. The resulting application achieves a very good success rate of 99.6% on the testing set and is being tested in real operation. The integration of the solution of parking space occupancy detection into a higher whole will make it easier for the drivers to find a vacant parking space, eliminate undesirable traffic in the car park and last but not least, will also reduce the production of CO_2. It can also provide the car park managers with a lot of information about occupancy times, utilization of specific spaces and more.

In the continued development, the goal will be to increase the success rate by increasing the training set and generalizing the trained models by adding data from other car parks. Based on the requirements of the target customer, other functionalities of the application will be added, such as the expected time when the parking space will become vacant, advanced statistics about car park utilization etc. The system is currently being adjusted to work on an integrated system with limited computer sources.

7. Acknowledgements

This paper summarizes a part of a masters diploma thesis that is being created based on a company assignment under the supervision of Ing. Jaroslav Rozman, PhD.[3] on the part of FIT BUT and Ing. David Herman[4] on the part of the company RCE systems s.r.o.

References

[1] Bradski, G., Kaehler, A.. Learning OpenCV: Computer Vision in C++ with the OpenCV Library. O'Reilly Media, Inc.; 2nd ed.; 2013. ISBN 1449314651, 9781449314651.

[2] Blanchette, J., Summerfield, M.. C++ GUI Programming with Qt 4. Upper Saddle River, NJ, USA: Prentice Hall PTR; 2006. ISBN 0131872494.

[3] Dokur, O., Katkoori, S., Elmehraz, N.. Embedded system design of a real-time parking guidance system. In: 2016 Annual IEEE Systems Conference (SysCon). 2016, p. 1–8. doi:\bibinfo{doi}{10.1109/SYSCON.2016.7490653}.

[4] Zhang, Z., Tao, M., Yuan, H.. A parking occupancy detection algorithm based on amr sensor. IEEE Sensors Journal 2015;15(2):1261–1269. doi:\bibinfo{doi}{10.1109/JSEN.2014.2362122}.

[5] Åoli, P., Marasovi, I., Stefanizzi, M.L., Patrono, L., Mainetti, L.. Rfid-based efficient method for parking slot car detection. In: 2015 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM). 2015, p. 108–112. doi:\bibinfo{doi}{10.1109/SOFTCOM.2015.7314120}.

[6] Zhou, H., Li, Z.. An intelligent parking management system based on RS485 and RFID. In: 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). 2016, p. 355–359. doi:\bibinfo{doi}{10.1109/CyberC.2016.74}.

[7] Sifuentes, E., Casas, O., Pallas-Areny, R.. Wireless magnetic sensor node for vehicle detection with optical wake-up. IEEE Sensors Journal 2011;11(8):1669–1676. doi:\bibinfo{doi}{10.1109/JSEN.2010.2103937}.

[8] Prasad, A.S.G., Sharath, U., Amith, B., Supritha, B.R., Asokan, S., Hegde, G.M.. Fiber bragg grating sensor instrumentation for parking space occupancy management. In: 2012 International Conference on Optical Engineering (ICOE). 2012, p. 1–4. doi:\bibinfo{doi}{10.1109/ICOE.2012.6409571}.

[9] Yusnita, R., Norbaya, F., Basharuddin, N.. Intelligent parking space detection system based onimage processing. International Journal of Innovation, Management and Technology 2012;3(3):232.

[10] Fusek, R., Sojka, E., Mozdřeň, K., Surkala, M.. Energy based descriptors and their application for car detection. In: Computer Vision Theory and Applications (VISAPP), 2014 International Conference on; vol. 1. IEEE; 2014, p. 492–499.

[11] Maria, G., Baccaglini, E., Brevi, D., Gavelli, M., Scopigno, R.. A drone-based image processing system for car detection in a smart transport infrastructure. In: Electrotechnical Conference (MELECON), 2016 18th Mediterranean. IEEE; 2016, p. 1–5.

[12] Amato, G., Carrara, F., Falchi, F., Gennaro, C., Vairo, C.. Car parking occupancy detection using smart camera networks and deep learning. In: 2016 IEEE Symposium on Computers and Communication (ISCC). 2016, p. 1212–1217. doi:\bibinfo{doi}{10.1109/ISCC.2016.7543901}.

[13] Jermsurawong, J., Ahsan, M.U., Haidar, A., Dong, H., Mavridis, N.. Car parking vacancy detection and its application in 24-hour statistical analysis. In: 2012 10th International Conference on Frontiers of Information Technology. 2012, p. 84–90. doi:\bibinfo{doi}{10.1109/FIT.2012.24}.

[14] Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., et al. Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software. 2009,.

[15] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22Nd ACM International Conference on Multimedia. MM '14; New York, NY, USA: ACM. ISBN 978-1-4503-3063-3; 2014, p. 675–678. doi:\bibinfo{doi}{10.1145/2647868.2654889}. URL `http://doi.acm.org/10.1145/2647868.2654889`.

[16] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, p. 1–9.

[3]Faculty of Information Technology, Brno University of Technology

[4]RCE Systems, s.r.o., Brno, Czech Republic

SPONSORS

Acknowledgements to Sponsors

This conference could not exist without the generous and noble-minded support of our sponsors.