Proceedings of the

OAGM&ARW Joint Workshop

Vision, Automation and Robotics

May 10-12, 2017
Palais Eschenbach
Vienna

OAGM - Austrian Association for Pattern Recognition
ARW - Austrian Robotics Workshop

TU Graz
Graz University of Technology

OAGM

OESTERREICHISCHE COMPUTER GESELLSCHAFT
AUSTRIAN COMPUTER SOCIETY

GMAR
Die österreichische Gesellschaft für Mess-, Automatisierungs- und Robotertechnik

Peter M. Roth, Markus Vincze, Wilfried Kubinger, Andreas Müller,
Bernhard Blaschitz and Svorad Stolc (eds.)

Proceedings of the
OAGM&ARW Joint Workshop
Vision, Automation and Robotics

May 10-12, 2017
Vienna, Austria

Austrian Association of Pattern Recognition (OAGM)
GMAR Gesellschaft für Mess-, Automatisierungs-,
und Robotertechnik

Editors

Peter M. Roth, Markus Vincze, Wilfried Kubinger, Andreas Müller, Bernhard Blaschitz and Svorad Stolc

Layout

Austrian Association of Pattern Recognition
http://aapr.at/

GMAR Gesellschaft für Mess-, Automatisierungs-, und Robotertechnik
http://www.gmar.at/

Cover

Stefan W. Schleich

Sponsors

ISBN (e-book) 978-3-85125-524-9
DOI 10.3217/978-3-85125-524-9

Contents

Preface

The second OAGM and ARW Joint Workshop on "Vision, Automation and Robotics" held in Vienna, at Palais Eschenbach, from May 10 to 12, 2017, provides a platform bringing together researchers, students, professionals and practitioners from both research directions to discuss new and emerging technologies in the field of machine driven perception and automated manipulation/autonomous movement. The OAGM and ARW workshops have a long tradition since 1980 and 2011, respectively, also stimulated by the Austrian RoboCup workshops (since 2006). Due to the highly overlapping interests of both communities the first joint event was organized in 2016. This second joint workshop will further strengthen the interaction of scientists working in vision, automation and robotics.

Computer Vision tries to perceive the physical world from image or video data resulting in applications such as scene understanding, object detection and tracking and 3D reconstruction. Thus, the main problems are to find suitable representations and to design and implement efficient (learning) algorithms. In contrast, Robotics aims at dealing with moving arms, graspers, and eventually moving vehicles. There are one or more actuators which have to be controlled accordingly in a planned matter for fulfilling given jobs. Some of them consist of additional sensors, e.g., graspers get some feedback for they can correctly catch and hold object without losing or destroying it; or the mobile device stops in front of an obstacle. These examples clearly demonstrate the relations between both fields. The outer world/the actual scenery is perceived by cameras; a consistent set of knowledge is modeled for the actuator for operating successfully either in a planned or even in an unplanned – standalone – strategy. Thus, there is a considerable interest in describing approaching features and possibilities and how the combination of different technologies could be beneficial.

The aim of the joint workshop is to discuss latest academic and industrial approaches and to demonstrate the recent progress. The call for papers resulted in 43 submissions, where finally according to the reviews of an international programme committee 37 contributions (23 talks, 14 posters) have been selected for presentation at the workshop. To highlight outstanding contributions, there prizes will be awarded during the joint workshop: The *OAGM Best Paper Award* sponsored by the *Austrian Computer Society (OCG)* and the *IEEE RAS Austria Best Student Paper Award*.

The goal of the workshop, bridging the gap between the Austrian Visual Computing and Robotics communities, is also supported by inviting three internationally established researchers representing both field:,Daniel Cremers (TU Munich, Germany), Pedro Sanz (Universitat Jaume I, Spain) and Herold Artés (RobArt GmbH, Austria), representing both areas.

Markus Vincze (General chair of the workshop)
Wilfried Kubinger (Chairman ARW)
Peter M. Roth (Chairman OAGM)
Vienna, May 2017

General Chair

Markus Vincze (TU Wien)

Programme Chairs OAGM

Bernhard Blaschitz (AIT)
Peter M. Roth (TU Graz)
Svorad Štolc (AIT)

Programme Chairs ARW

Wilfried Kubinger (UAS Technikum Vienna)
Andreas Müller (Johannes Kepler University Linz)
Markus Vincze (TU Wien)

Web Chair

Friedrich Praus (UAS Technikum Vienna)

Programme Committee OAGM

Helmut Ahammer (Medical University of Graz)
Nicole Artner (TU Wien)
Csaba Beleznai (AIT)
Horst Bischof (TU Graz)
Kristian Bredies (University of Graz)
Wilhelm Burger (Upper Austria University of Applied Sciences)
Christia Eitzinger (Profactor)
Cornelia Fermüller (University of Maryland)
Friedrich Fraundorfer (TU Graz)
Harald Ganster (Joanneum Research)
Margrit Gelautz (TU Wien)
Martin Hirzer (TU Graz)
Florian Kleber (TU Wien)
Reinhold Huber-Mörk (AIT)
Walter G. Kropatsch (TU Wien)
Arjan Kuijper (Fraunhofer IGD)
Roland Kwitt (University of Salzburg)
Christoph Lampert (IST Austria)
Mathias Lux (Alpen-Adria-Universität Klagenfurt)
Hubert Mara (Heidelberg University)
Martin Humenberger (AIT)
Bernhard Moser (Software Competence Center Hagenberg)
Gerhard Paar (Joanneum Research)
Roland Perko (Joanneum Research)
Thomas Pock (TU Graz)
Horst Possegger (TU Graz)
Hayko Riemenschneider (ETH Zurich)
Josef Scharinger (Johannes Kepler University Linz)
Roberst Sablatnig (TU Wien)
Otmar Scherzer (University Vienna)
Daniel Soukup (AIT)
Darko Stern (TU Graz)
Andreas Uhl (University of Salzburg)
Martin Welk (UMIT Hall/Tyrol)
Martin Winter (Joanneum Research)
Christopher Zach (Toshiba Research Europe)

Programme Committee ARW

Mathias Brandstötter (Joanneum Research)
Alexander Hofmann (UAS Technikum Vienna)
Bernhard Dieber (Joanneum Research)
Gerald Fritz (Profactor)
Stefan Gspandl (incubed IT GmbH)
Michael Hofbaur (Joanneum Research)
Gernot Kronreif (ACMIT)
Wilfried Kubinger (UAS Technikum Vienna)
Wilfried Lepuschitz (Practical Robotics Institute Austria)
Kurt Niel (FH Upper Austria)
Andreas Müller (Johannes Kepler University Linz)
Justus Piater (University of Innsbruck)
Friedrich Praus (UAS Technikum Vienna)
Bernhard Rinner (Alpen-Adria-Universität Klagenfurt)
Lukas Silberbauer (taurob OG)
Gerald Steinbauer (TU Graz)
Markus Vincze (TU Wien)
Christian Wögerer (Profactor)
Michael Zillich (TU Wien)

Awards 2016

The

OAGM Best Paper Award 2016

was awarded to the paper

On a Fast Implementation of a 2D-Variant of Weyl's Discrepancy Measure

by

Christian Motz and Bernhard Moser.

The

IEEE RAS Austria Best Student Award 2016

was awarded to the paper

Localization of an Automated Guided Vehicle (AGV) by Stereo Based Visual Odometry and Artificial Landmark Detection

by

Daniel Klingersberger and Gerald Zauner.

Index of authors

Keynote Talks

Recent Achievements in Underwater Intervention Systems the Role of Perception & Robotic Manipulation

Pedro J. Sanz

IRS-Lab, Universitat Jaume I, Spain

Abstract

From the UJI foundation (1991), one of the research elds more active has been robotics. So, a lot of di erent activities concerning this exciting eld have been developed during these years. In addition, many projects, some of them funded by European and Spanish institutions have been successfully carried out. There are other robotic labs at UJI, but only one working in the underwater domain: IRS-Lab. Thus, after more than twenty years of research in some speci c technologies (e.g. multisensory based manipulation, telerobotics, or human-robot interaction HRI), always applied to real life scenarios, a few years ago we face the underwater intervention context. In this new scenario the dream is named the underwater autonomous vehicle for intervention (I-AUV). However, a long path is still necessary to pave the way to underwater intervention applications performed in a complete autonomous way. This presentation reviews the di cults to overcome, the solutions explored and the evolution timeline in the way towards I-AUVs, putting the enphasis on the main contributions reached through those projects coordinated by the IRS-Lab, and always considering the role played by perception and manipulation there.

Dense & Direct Methods for 3D Reconstruction & Visual SLAM

Daniel Cremers

Computer Vision Group, Department of Computer Science,
Technical University of Munich, Germany

Abstract

The reconstruction of the 3D world from images is among the central challenges in computer vision. *Starting in the 2000s, researchers have pioneered algorithms which can reconstruct camera motion and sparse feature-points in real-time. In my talk, I will introduce spatially dense methods for camera tracking and 3D reconstruction which do not require feature point estimation, which exploit all available input data and which recover dense or semi-dense geometry rather than sparse point clouds. Applications include 3D photography, 3D television, and autonomous vehicles.*

Austrian Robotics Workshop

A framework for cellular robots with tetrahedral structure

Michael Pieber and Johannes Gerstmayr

***Abstract*— An adaptive tetrahedral element (ATE) has been designed, which can attach to and detach from other ATEs along their deformable faces. The goal is to obtain any configuration or shape autonomously. The tetrahedrons edges represents six actuators and each ATE has its own micro-controller, battery and wireless transceiver module. Several connected ATEs are forming an adaptive robot with tetrahedral structure (ARTS) which is intended to represent any geometric form with a piecewise flat surface. Contrary to existing cellular and tetrahedral robots ARTS combines the advantages of self-reconfigurable modular robots and tetrahedral robots which have the ability to change their shape.**

I. INTRODUCTION

Self-reconfigurable robots with the ability to represent arbitrary shapes leads to an enormous number of real-world applications. Such applications are feasible within the self-assembly of large scaffolds, using ATEs with an overall size of one meter. Adaptive structures are needed e.g. for the growing complexity of current architectural design. In the mid-range size of ATEs, using centimeters for each actuator, the possibility to represent any 3D geometry could be used for rapid-prototyping and for the visualization of 3D structures in business and education.

II. RELATED WORK

Ahmadzadeh et al. [1] identified and cited 94 modular robots. Most of these are arrays of kinematically-constrained simple robots with few degrees of freedom [5], [3], [8], [2], [7]. These robots can attach to and detach from each other manually or automatically mostly with a mechanically [5] or magnetic [8], [2] connection mechanism.

The combination of self-reconfiguration robots with the ability to represent arbitrary shapes are presented recently in [6]. The connection mechanism along the deformable faces of the ATEs are patented [4] by the authors of the present paper.

III. ARTS – A TETRAHEDRAL ROBOT

ARTS is a modular robotic system which is based on adaptive tetrahedral elements (ATEs). The single ATEs can be understood as cells of a larger structure, similar to cells in biology. Each ATE can deform and has six degrees of freedom resp. actuators. In a continuum mechanics interpretation, an ATE can undergo any kind of stretch or shear deformation. The deformation of the single ATEs gives the robotic system are large amount of variability.

Michael Pieber and Johannes Gerstmayr are with the Institute of Mechatronic, University of Innsbruck, 6020 Innsbruck, Austria {michael.pieber, johannes.gerstmayr}@uibk.ac.at

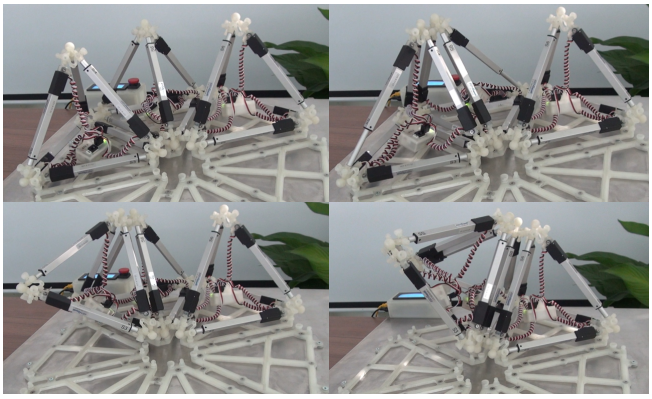


Fig. 1. Three tetrahedral elements attach along their deformable faces to an adaptive robot with tetrahedral structure. The elements standing in initial position an a plate. First the element on the left side attach to the middle ATE. In the next step both ATEs are connecting to the third ATE on the right side.

Each ATE itself is a mechatronic system, which includes the actuators, four double-spherical joints, 3 pairs of connectors at each of the four faces, a control and power unit, a wireless connection and a battery, see Fig. 2. In the current design, most parts are manufactured using a high-end 3D printer 'ProJet 3500 HD' from 3D Systems, with the material VisiJet M3-X. In comparison to conventional cubic or spheric modular robots, the tetrahedral structure leads to a light-weight design. Furthermore, the ATEs can connect and change the overall shape of the structure, see Fig. 1, and finally shall have the possibility to move ATEs along the surface by deformation of surrounding ATEs. As a challenge of the design, there are restrictions for the elongation of each actuator, which leads to severe limitations of the motion space of each cell. This also limits the angles of the edges at the spherical joints, being boundaries to the geometrical design.

The system of ATEs, from which we currently have built four fully functional elements, is used in a way, that they are always either positioned at a fixed space on a ground plate, or they are connected to one or several other ATEs, compare Fig. 1. The unique design is based on the connection at the faces, rather than the nodes. This avoids any restrictions within the connection of several tetrahedral elements, as known from other tetrahedral robots, see the references provided above. The advantage of tetrahedral robots is the convenient computation of the movement of the structure, which can be understood as a deformable mesh. The mesh – similar to a finite element mesh – can be modeled to be elastic with certain geometric limitations, which can be

implemented on a computer code similar to the computation of a space truss. The single point-to-point motions of ARTS are sent to each ATE via a wireless connection from a master, which is connected to a conventional personal computer.

The main problem, which is currently investigated, is based on the difference of the idealized tetrahedral mesh and the constructed geometry of the ATEs, which brings in restrictions in the motion space of the system. Promising ways to overcome these limitations have been worked out and will be presented.

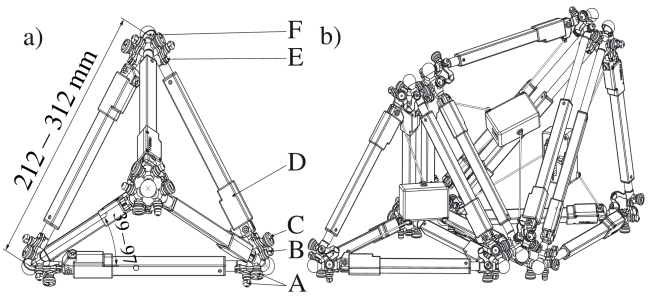


Fig. 2. Topview of a single ATE a) mechanical design of an ATE shown without cables and electronics: A-docking mechanism, B-male connector, C-female connector, D-actuator, E-orientation element, F-spherical joint; b) three connected ATEs forming an adaptive robot with tetrahedral structure

IV. RECONFIGURATION MECHANISM

Besides the mechatronic design, the control of the ATEs can be challenging, as soon as many cells are connected to each other, compare Fig. 3. In addition to the design of ARTS, we are developing several computational schemes, which define the motion of each ATE for reconfiguration from one to another shape. In order to fulfill this challenging task, the computation is split into three parts:

1) In the first part, the initial and the final mesh of the structure is computed. It is necessary that both configurations consist of a similar number of ATEs. The simplest way is depicted in Fig. 3, where the initial configuration consists of a rectangular block.
2) The rectangular block in Fig. 3a can be understood as parking positions of the ATEs. The main task of reconfiguration, is to find according parking positions to each of the ATEs of the structure, which is a hollow sphere in the present case. The shortest ways for movement of ATEs along the surface are depicted in Fig. 3a-e. This shows how a single ATE needs to be moved. In fact, the movement strategy is done such, that an ATE which has the longest distance to the base is selected in the structure, see Fig. 3e. This ATE is moved to an available parking space at the base block, which is closest to the center. While the algorithm is computing the destruction of the hollow sphere, the steps are then applied in reversed order.
3) In the final step, the movement of the ATEs needs to be performed by means of mesh deformation. This is done such that the cells can move along the surface. Currently, this is done with manual inputs only, however, an algorithm which can automatically compute this transformation is currently developed.

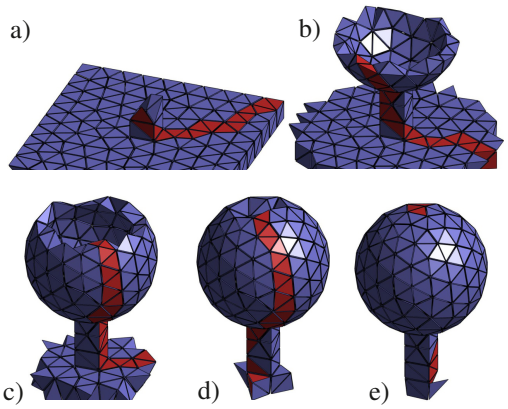


Fig. 3. Exemplary steps to reconfigure from one to another configuration. a) parking position; a-e) the red colored surfaces mark the shortest path for movement of ATEs along the surface.

Converting a complex structure (A) into another complex structure (B) can be performed such that between these configurations, the ATEs are transformed into a rectangular block. In this way, only the reconfiguration from a rectangular block to a complex structure must be computed.

V. CONCLUSIONS

The single adaptive tetrahedral elements (ATEs) follow a light-weight design principle. ARTS leads to a highly redundant superstructure and has the potential for a disruptive technology. Current limitations are within geometric restrictions of the workspace and the differences between an idealized geometric mesh and the real (constructed) geometry of ATEs.

REFERENCES

[1] H. Ahmadzadeh, E. Masehian, and M. Asadpour, "Modular Robotic Systems: Characteristics and Applications," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 81, no. 3-4, pp. 317–357, 2016.

[2] B. K. An, "EM-Cube: Cube-shaped, self-reconfigurable robots sliding on structure surfaces," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3149–3155, 2008.

[3] R. Belisle, C.-h. Yu, and R. Nagpal, "Mechanical Design and Locomotion of Modular Expanding Robots," *ICRA 2010 Workshop Modular Robots: State of the Art*, pp. 17–23, 2010.

[4] J. Gerstmayr and M. Pieber, "Modulares, selbst rekonfigurierbares Robotersystem," pCT/EP2016/073703, 2016.

[5] M. Jorgensen, E. Ostergaard, and H. Lund, "Modular ATRON: modules for a self-reconfigurable robot," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 2, pp. 2068–2073, 2004.

[6] M. Pieber and J. Gerstmayr, "An Adaptive Robot with Tetrahedral Cells," *The 4th Joint International Conference on Multibody System Dynamics, Montral, Canada*, 2016.

[7] J. W. Romanishin, K. Gilpin, and D. Rus, "M-blocks: Momentum-driven, magnetic modular robots," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4288–4295, 2013.

[8] V. Zykov, A. Chan, and H. Lipson, "Molecubes: An Open-Source Modular Robotics Kit," *IROS-2007 Self-Reconfigurable Robotics Workshop*, pp. 3–6, 2007.

Package Delivery Experiments with a Camera Drone

Jesús Pestana[1] and Michael Maurer[1] and Daniel Muschick[2] and Devesh Adlakha[1]
and Horst Bischof[1] and Friedrich Fraundorfer[1]

***Abstract*— The undergoing efforts for the integration of robotics into logistics systems is affecting the production workflow at all stages, from the transportation and the handling of parts inside storage and production facilities to the final product distribution. In this paper we address the problem of delivering a package by means of a multirotor drone. We describe a fully autonomous package delivery flight demonstration prepared in collaboration with an industrial partner. All computations are performed in real-time on-board the drone. A gimbal camera is utilized to realize the vision-based localization, by means of fiducial markers, of the delivery position and the landing platform on a pickup truck. The demonstration consists of the fully autonomous execution of the following tasks: the drone takes-off from the truck, looks for the delivery position, proceeds to land and drop the package, flies back to the distribution truck and follows it, and the flight is finished by performing the landing on the static vehicle. The experiments focus on the performance of the vision-based truck following.**

I. INTRODUCTION

In this paper we present a fully autonomous drone that using only on-board processing is able to perform coarse navigation using GPS, vision-based precise vehicle following and landing on static platforms (see Figs. 1 & 3). We used our system to perform a fully autonomous package delivery flight demonstration in collaboration with an industrial partner. The main technical challenges related to this work are the navigation control, the real-time vision-based pose estimation of the vehicle and the landing positions and their integration with the navigation control. In order to obtain the required localization precision for the vehicle following and the landing tasks we use visual fiducial markers.

Drones are a hot topic and an ongoing research area. These aerial platforms are suitable for being integrated in logistics systems, for instance, for the transportation of goods. Package delivery by means of an autonomous drone can significantly reduce the costs of distribution. A succinct feasibility analysis by D'Andrea [4] estimated its operating cost at 10 cents for a 2 kg payload and a 10 km range.

The main challenges faced by real-world drone package delivery are highlighted by the following selection of recent research works: an obstacle mapping method that encodes at cell-level the value of occupancy and its variance [1], testing modern deep-learning based object detection algorithms on-board drones [6], trajectory planning intended for navigation in cluttered environments [3] and landing on vehicles that are moving in straight roads at speeds of up to 40 km/h [2].

[1]Institute for Computer Graphics and Vision, ICG - TU Graz
{pestana,maurer,bischof,fraundorfer}@icg.tugraz.at
[2]Institute of Automation and Control, Graz University of Technology
daniel.muschick@bioenergy2020.eu

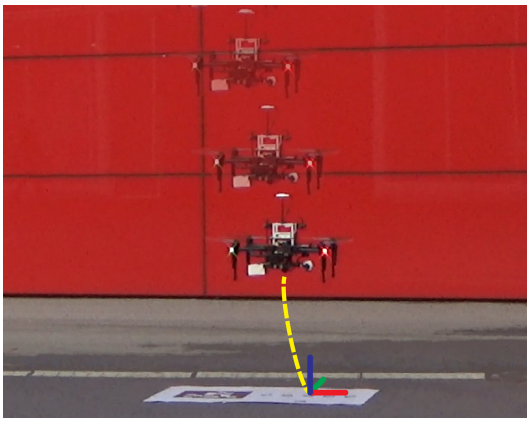


Fig. 1. Illustration of autonomous vision-based controlled drone landing on a marked delivery position in order to deliver a package.

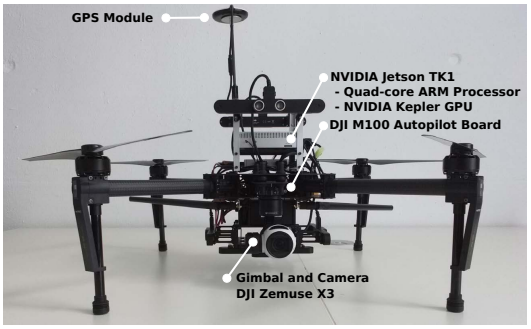


Fig. 2. DJI M100 quadrotor equipped with a Nvidia Jetson TK1 on-board computer (DJI Manifold), autopilot, GPS module, DJI Zenmuse X3 gimbal camera (1280×720 px) and an electro-magnet to carry a package of 100 g.

II. SYSTEM OVERVIEW

a) Hardware Setup

Our drone is equipped as shown in Fig. 2. For the experimental tests, the E-Mobility electric powered pickup truck "ELI" from SFL Technologies [7] was used, see Fig. 3. The delivery position and the landing platform are tagged using a 39×39 cm 36h11-family Apriltag fiducial marker [5]. Using this approach the relative pose of the gimbal camera with respect to the landing-platform at a distance of 3.5 m can be estimated with an accuracy of around 3 cm.

b) Software Setup

The inter-module communication is achieved by means of the Robot Operating System (ROS). Since our experimental results focus on the car following performance, only the main modules related to this task are explained, which are: the gimbal camera landing-platform tracking, the vehicle speed estimation and the control algorithm.

Fig. 3. Drone vision-based vehicle following, marked with a 39×39 cm Apriltag. Experiment: 3 min, mean speed 7.91 km/h and top speed 13.35 km/h.

b.1) Gimbal Camera Landing Platform Tracking

The drone's GPS measurements, the gimbal current orientation and the camera relative pose to the marker are combined to estimate the position of the markers in world coordinates. During specific tasks, these position estimates can be used to command the gimbal to point at the marker that is positioned on top of a landing platform. This approach is used during the vehicle following, package delivery and landing tasks.

b.2) Vehicle Speed Estimation

The marker relative pose estimates are calculated at around 25 fps for a resolution of 1280×720 px. These estimates are stored in a queue with a length of 20 elements. The vehicle speed is estimated for every linear coordinate using linear regression on the elements of the queue, which does not incur significant computation costs.

b.3) Navigation Control Algorithm

The flight behavior of our drone was characterized by performing speed command step-response identification tests. A rough controller parameter tunning was calculated based on the resulting model and it was later experimentally improved.

We utilize a feedback loop controller based on the PID controller architecture for the three linear coordinates and the yaw heading. In order to improve its performance, the controller utilizes both position and speed references. The utilized measurement feedback are the position and velocity provided by the autopilot telemetry, obtained through the fusion of GPS data with the IMU and magnetometer data.

III. EXPERIMENTAL RESULTS

a) Package delivery mission

We succeed in performing a fully autonomous mission where the drone takes-off from the truck, follows a GPS predefined flight trajectory, looks for the delivery position, proceeds to land and drop the package, takes-off again, flies back to the distribution truck, follows it for a while and lands on the static vehicle. This mission is summarized in our video[1].

b) Vehicle following experiment

The task of the drone is to follow the vehicle that is marked with a landing-platform at a constant distance of 2.5 m from behind and above. The vehicle speed estimate, see Sec. b.2, is used as speed reference for the controller.

The vehicle following experiment lasted 3 min during which the drone performed the task successfully all the time. Pictures of this experiment are shown in Fig. 3 and the logged trajectories and speeds of the drone and the vehicle are plotted in Fig. 4. Overall, during this experiment, the mean and top vehicle speed were 7.91 km/h and 13.35 km/h, and the root mean square error (RMSE) of the position and speed control tracking error were 0.37 m and 1.34 km/h.

[1]Package delivery demo: `https://youtu.be/bxM6dls2wuo`

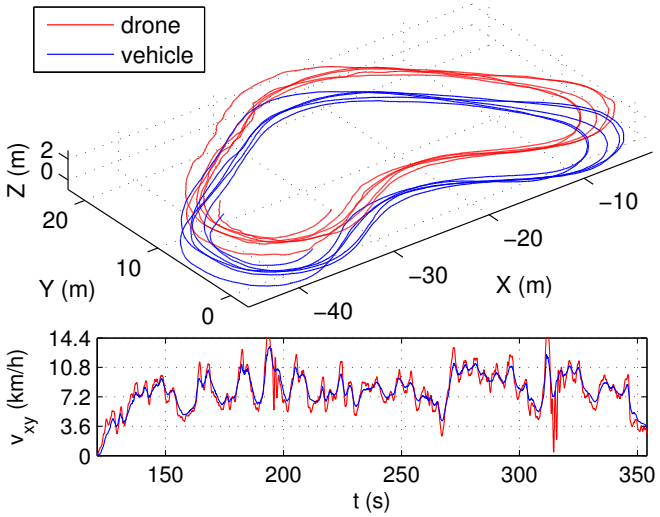


Fig. 4. Vehicle following experiment of 3 min duration. The plot shows the (red) drone and (blue) vehicle 3D positions and speeds over time.

IV. SUMMARY

In this paper we presented a fully autonomous drone that using only on-board processing is able to perform coarse navigation using GPS and vision-based precise vehicle following and landing (see Figs. 1 & 3). Our fully autonomous package delivery flight demonstration, carried out in collaboration with SFL Technologies, was reported by local newspapers[2,3]. In future work we plan to use this system as a first step towards performing autonomous landing on a moving vehicle.

ACKNOWLEDGMENTS

The authors thank SFL Technologies for providing the testing environment and the electric vehicle ELI [7].

REFERENCES

[1] A. Agha-mohammadi, "Confidence-aware occupancy grid mapping: A planning-oriented representation of environment," *IROS2016 Workshop*.

[2] A. Borowczyk, D.-T. Nguyen, A. P.-V. Nguyen, D. Q. Nguyen, D. Saussié, and J. L. Ny, "Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle," *arXiv preprint*, 2016.

[3] S. Daftry, S. Zeng, A. Khan, D. Dey, N. Melik-Barkhudarov, J. A. Bagnell, and M. Hebert, "Robust monocular flight in cluttered outdoor environments," *IROS2016 Workshop, arXiv:1604.04779*, 2016.

[4] R. D'Andrea, "Guest editorial can drones deliver?" *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, 2014.

[5] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[6] A. Sankalp, D. Geetesh, J. Sezal, M. Daniel, A. Greg, Y. Song, and S. Sebastian, "Autonomous semantic exploration using unmanned aerial vehicles," *IEEE IROS2016 Workshop*, 2016.

[7] SFL Technologies, "E-Mobility electric powered vehicle ELI," http://www.sfl-technologies.com/spektrum/e-mobility/, accessed: 2017-03-14.

[2]ELI roll-out demo - Mein Bezirk - `http://bit.ly/2hott8t`

[3]ELI roll-out demo - Kleine Zeitung - `http://bit.ly/2it6N2P`

A Model-Based Fault Detection, Diagnosis and Repair for Autonomous Robotics systems

Stefan Loigge[1] and Clemens Mühlbacher[1] and Gerald Steinbauer[1] and Stefan Gspandl[2] and Michael Reip[2]

***Abstract*— Autonomous robots comprise of several complex software and hardware components which interact with the environment to fulfill a certain task. Due to the non-determinism, inherent of the environment and complexity of the components one cannot expect that the robot will never show a fault. Instead one needs to deal with the occurrence of faults in the robotics system. As we focus on autonomous robots the robot should deal with faults in an automated fashion.**

In this paper, we present a model-based fault detection and diagnosis method with a simple but powerful method to repair faults. Using this method, the robot can detect and react to faults in a timely manner. Furthermore, no human intervention is necessary thus allowing the robot to be autonomous. As not every repair can be performed by the robot itself the system allows the robot also to inform the maintenance staff which repairs are necessary. Thus, this approach reduces the time for fault localization of the maintenance staff.

I. Introduction

Autonomous robots perform tasks in (partly) open environments. To perform such a task, the robot uses several complex software and hardware components which interact with each other. Due to the (partly) open environment and the complex components, one cannot assume that no fault will occur. Instead one needs to design the robotic system with faults in mind. Thus, one either add fault handling in each component or one uses a more general approach. One such general approach is the use of a model-based approach as outlined in [1]. The model is used to describe the system behavior and to allow the system to detect a fault.

The use of a model-based approach allows the robot to determine if a fault has occurred. Furthermore, the robot can determine which component most likely caused this fault. Using the information which component is faulty the robot can determine which action to perform to react to this fault. Besides the possibility that the robot detects and reacts to a fault a model-based approach also allows to separate the current system description from the fault detection and localization components. As the model is used to describe the system the fault detection and localization can be done on the model only. Thus, one can use the software to perform this reasoning for many different robots without changes. The only thing which needs to be changed for a robot is the model of the robot. As many robotic system reuse components of other robots, or have similar robot components one can often reuse parts of already existing models. Thus, further decreasing the effort to perform fault detection and localization.

In this paper, we present such a model-based diagnosis approach. The method uses several different observers to observe properties of the system. These properties are observed to detect a fault. With the help of the observed properties, the system can derive a diagnosis which component caused the fault. This allows pinpointing the fault without extra costs as the only information necessary for the diagnosis is already provided through the definition of the observations. To allow the robot to react to a detected fault a simple rule engine can be used. The rule engine allows the robot to react fast to a fault and to trigger more complex repair actions. Through this fast reaction, one can reduce the chance that a robot will endanger itself or pose a threat to its surrounding.

The remainder of the paper is organized as follows. In the next section, we will give an overview of the fault detection, diagnosis, and repair system. The proceeding section discusses the different observers which check system properties in more detail. Afterward, we discuss the diagnosis engine which is used to identify the faulty component. In Section V we discuss the rule engine and how it can be used to react to faults. In Section VI, we show a use case where the system was used on an industrial robotics system. Before we conclude the paper, we discuss some related research. Finally, we conclude the paper and point out some future work.

II. System Overview

To create a robotic system, the robot operating system (ROS) [2] is often used as a framework. With the help of ROS one can use several software components, which are called nodes, and interact with each other. This interaction can be performed with the help of publisher-subscriber principle which allows exchanging message between each ROS node. To define and identify for such communication channel ROS uses so-called topics. These are strings defining an n-to-n communication channel. Furthermore, one can use service calls to provide a service from one component to another. In the remainder of the paper, we will focus on messages exchanged by topics as these are used more often as services and allow an easy introspection.

Using ROS, a robotic system can be created which uses several software components interacting with each other. As we are interested in detecting and identifying faults and react to these faults we use the system depicted in Figure 1. The

[1]Stefan Loigge, Clemens Mühlbacher and Gerald Steinbauer are with the Institute for Software Technology, Graz University of Technology, Graz, Austria. {sloigge,cmuehlba,steinbauer}@ist.tugraz.at

This work is partly supported by the Austrian Research Promotion Agency (FFG) under grant 843468.

[2]Stephan Gspandl and Michael Reip are with incubedIT, Hart bei Graz, Austria. {gspandl,reip}@incubedit.com

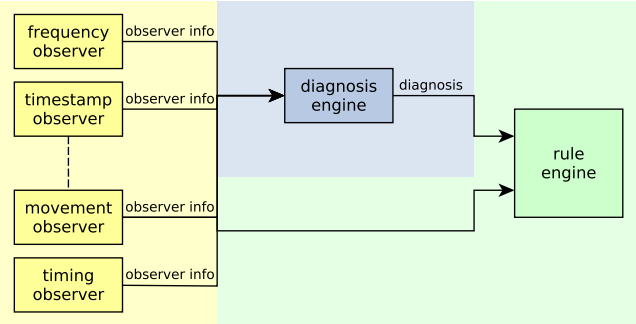


Fig. 1. Monitoring, diagnosis, and fault handling overview: observer (yellow), diagnosis engine (blue), rule engine (green), [4]

system consists of three parts. A set of observers which is used to detect a fault. A diagnosis engine which identifies the component which caused the fault. The usage of observers and a diagnosis engine for a ROS was already proposed in [3] and was extended in this paper. Finally, a rule engine is used to react to faults.

To allow the method to be applied for already existing software, it is of interest that the used software components are not needed to be altered. Thus, instead of detecting a fault in the software components directly, we use information provided by the interaction of the software components. This allows that we can detect a fault without changing existing software components. This can be simply achieved in ROS by introspection on the topics which are used for the communication. By observing properties of a topic, e.g. frequency of communication on a topic, the system can be checked if it conforms to the given model. This observation is provided using different observers where each observer is used to determine if a specific property hold. We will discuss in the next section in more detail which observers exist.

With the observations, only the robot would only be able to detect that a fault has occurred. But the robot needs also to determine which component caused the fault. This is of special interested if several malfunctions are detected at the same time. With the help of the model of the system and a reasoning process, the diagnosis engine determines which components are faulty. The reasoning performed uses a consistency-based diagnosis [5] approach which searches for a minimal set of components which are blamed for being faulty explain the observations. We discuss the diagnosis engine in more detail in Section IV.

After the robot, has determined which components might have caused the fault the robot needs to react to this fault. This is achieved with the help of a rule engine which uses the current diagnosis of the system together with the observations. By combining the diagnosis and the observations the rule engine can determine which rule should be triggered to execute a specific repair. This allows the robot to react in a timely manner. If a planning system would be used as it was described in [3] a possible high planning time may not allow such a fast reaction. Due to this reaction, the robot can bring itself into a safe state which can be used afterward to perform a more complex repair. Let's consider a simple example. The robot detects that the laser scanner used for navigation is malfunctioning. After determining this malfunction, the robot can react and stop immediately. Thus, the robot will not drive into an obstacle. After the robot, has stopped, the robot can perform a more complex reasoning which repair should be performed with another method [3]. Or it may even try to reconfigure itself to deal with the fault [4]. In this paper, we will focus only on a quick reaction to a fault and not a complex repair or reconfiguration mechanism. We will discuss the rule engine in more detail in Section V.

The complete system as it is described in this paper is public available under `http://git.ist.tugraz.at/ais/model_based_diagnosis`.

III. Observers

As outlined above we use several observers to check if a certain property of the robotic system holds. These observers are used to mediate between the concrete messages send in the robotic system and the abstract model of the system. This allows that the model of the system uses a predicate based representation of the robotic system which simplifies the diagnosis process. Furthermore, the observers can use specifically design methods to observe a certain property allowing a small computation overhead to provide the observations.

To properly supervise the system different types of observers are used. Some observers observe the behavior of a node directly where others observe the behavior or the message exchanged. To observe the behavior of the node directly two observers can be used.

- The activated observer checks if a node is present in the robotic system. Thus, allowing to check if the system is properly configured.
- the resource observer checks if a specific node in the system uses a predefined amount of system resources, e.g. CPU. This allows checking if the node neither consumes too many resources, e.g. a memory leak causing the accumulation of memory nor the consumption of too fewer resources, e.g. no CPU usage as the node has deadlocked itself.

To observe the behavior of the message exchange in the system the following six observers can be used.

- The time-out observer checks if at least one message was sent within a specified time interval. This allows checking if a topic is used for communication and performs a watchdog functionality for a topic. Thus, allowing to revival problems which cause the communication to break down, e.g. the node which should send an information can't produce an output.
- The HZ observer checks on a topic if messages are exchanged with a given frequency. This allows checking if a communication is done on a regular basis. Thus, allowing to check if the node which provides the information is overloaded.
- The time-stamp observer checks if the timestamp of a message send is not too old. This allows checking if

old data are sent in the system thus reveal problems to produce new data.
- The timing observer checks the time difference between one message send on one topic and one message sends on another topic. This can be used to check if a node produces an expected output within the expected time frame. Thus, one can detect if a processing step takes too long.
- The score observer checks if the float value of a topic is within a range. This allows checking the calculated score, specifying the performance of an algorithm outcome.
- The movement observer uses two topics which specify the movement of the robot for correlation. This correlation can be used to check if the expected movement differs significantly, e.g. the movement measured by the IMU is different to the movement measured by the odometry.

Using the different observer types different properties of the system can be checked. As the observations, may be subject to noise one cannot simply use the raw values to perform the check. Instead one can apply different filter mechanism to process the raw values before performing a check. Thus, the raw value to check, e.g. the frequency of a topic is treated as a signal which needs to be filtered as it is common in signal processing [6].

After filtering the raw values of the observation one needs to perform a check to determine if the observed values are acceptable. This can be done by simple checks which determine the correctness using comparison with a fixed value. But it is also possible to use a more complex test which uses a statistical approach. This is done by performing a student-t-test [7] on the filtered data. Through this test one can check if the hypothesis that the observation is acceptable needs to be withdrawn. Thus, allowing to perform a check considering the statistical uncertainty.

All except one observer type check the raw value observed with a nominal value of the mode, e.g. the frequency of a topic with the expected value. The movement observer is the exception, as it correlates two values with each other. The idea is to use the redundant information in the robotic system to check for consistency. This follows the idea of residuals [8] which create an error term between redundant information in the system. To do so, we first derive from each movement measurement the resulting acceleration. Thus, if the movement is given by the current velocity the movement is differentiated to get the acceleration. Afterward, the accelerations of one input are subtracted from the other input. If no fault occurs this value is zero. Due to the noise measurement, the value follows a Gauss distribution with zero mean. With the help of the filter methods, one can estimate the mean of the distribution and use this estimation to perform a check if the value is close enough to zero.

IV. Diagnosis Engine

Using the observers one can detect if one property of the system behaves not as defined. This allows to detect a fault but does not allow to isolate the faulty component directly. Instead one needs to perform a reasoning. We use the idea of consistency-based diagnosis [5] to perform this reasoning. The reasoning uses the information about the observations taken from the system as well as the topology of the system. This allows handling fault propagation properly. To specify the system, we define a system to consists of a set $\mathcal{N}$ defining the nodes of the system. These are the software components which are running and need to be diagnosed. Additionally, the system consists of a set $\mathcal{M}$ defining the topics which are used to exchange messages between the software components. To represent the input topics to a node we use the function $input : \mathcal{N} \rightarrow 2^{\mathcal{M}}$. The output which is produced by a node is defined through $output : \mathcal{N} \rightarrow 2^{\mathcal{M}}$. Using the set $\mathcal{N}$, and the functions $input$ and $output$ one can describe the information flow of the system. This information flow is of interest as a fault can be propagated along this information flow.

To define a software component n to be faulty we use the predicate $AB(n)$. Besides the software component also a topic can be observed to be faulty thus we write $AB(m)$ that on observation indicate that the message exchange m is not as expected. Please note that we are only interested in the predicates $AB(n)$ which are used to explain a faulty behavior. Thus, we will search for a minimal set of $AB(n)$ predicates which explain the observations.

To specify the fault propagation, we use the following logical formula which is defined for each $n \in \mathcal{N}$.

$$\forall m_o \in output(n) : AB(m_o) \rightarrow \left(AB(n) \bigvee_{m_i \in input(n)} AB(m_i) \right)$$

The formula states that if the output of a software component seems to be faulty either the component is faulty or one of its inputs where faulty. Thus, one can propagate the fault from input to output.

With the help of the above formula, we can define the fault propagation in the system per the structure of the system. Besides the structure of the system, we need also to define how the observations made a link to the components in the system. This link depends on the type of observation made. We use the following formulas to link the observations and the components of the system.

- If component n is observed with the help of an activated observer ($obs_{activated}(n)$) we state the following logical formula.

  $$\neg obs_{activated}(n) \rightarrow AB(n)$$

  As we directly observe the component we can detect that the component is faulty if the observation indicates a fault.
- If component n is observed with the help of a resource observer ($obs_{resource}(n)$) we state the following logical formula.

  $$\neg obs_{resource}(n) \rightarrow AB(n)$$

As we directly observe the component we can detect that the component is faulty if the observation indicates a fault.

- If a topic m is observed with the help of a time-out observer ($obs_{timeout}(m)$) we state the following logical formula.

$$\neg obs_{timeout}(m) \rightarrow AB(m)$$

As we only observe a topic we can only state that the topic is abnormal and use the structure to determine which component caused this fault.

- If a topic m is observed with the help of an HZ observer ($obs_{hz}(m)$) we state the following logical formula.

$$\neg obs_{hz}(m) \rightarrow AB(m)$$

As we only observe a topic we can only state that the topic is abnormal and use the structure to determine which component caused this fault.

- If a topic m is observed with the help of a time-stamp observer ($obs_{timestamp}(m)$) we state the following logical formula.

$$\neg obs_{timestamp}(m) \rightarrow AB(m)$$

As we only observe a topic we can only state that the topic is abnormal and use the structure to determine which component caused this fault.

- If two topics m_1 and m_2 are observed with the help of a timing observer ($obs_{timing}(m_1, m_2)$) we state the following logical formula.

$$\neg obs_{timing}(m_1, m_2) \rightarrow (AB(m_1) \vee AB(m_2)).$$

If the timing of the two topics does report an error one of the topics need to cause the fault. As we only observe that at least one of the topics need to be abnormal we need to use the structure to determine which component caused this fault.

- If a topic m is observed with the help of a score observer ($obs_{score}(m)$) we state the following logical formula.

$$\neg obs_{score}(m) \rightarrow AB(m)$$

As we only observe a topic we can only state that the topic is abnormal and use the structure to determine which component caused this fault.

- If two topics m_1 and m_2 are observed with the help of a movement observer ($obs_{movement}(m_1, m_2)$) we state the following logical formula.

$$\neg obs_{movement}(m_1, m_2) \rightarrow (AB(m_1) \vee AB(m_2) \vee AB(movement))$$

The formula states that if the movement is observed to be faulty then either one of the topics is abnormal or the movement relation is not valid. The movement relation may not be valid as we may observe the difference between the IMU and the odometry. If the robot now slips the odometry and the IMU do no longer agree but none of the components is faulty. Instead, the model of the environment imposing that these two sources of information are redundant does not longer hold.

With the logical formulas from above, the model of the system is described. Furthermore, the link between the observations and the model of the system is defined through the logical formulas from above. With the help of this logical formula, one can derive which set of $AB(n)$ predicates is consistent. This set represents the software components which need to be faulty to explain the observed faults. As we are interested in the most likely explanation we follow the idea of Occams razor and search for a minimal set of $AB(n)$ predicates which are consistent.

To find this minimal set we use a minimal hitting set algorithm. The algorithm uses a sat solver to derive if a set of $AB(n)$ predicates is consistent. If the set of predicates is consistent the algorithm has found a diagnosis. Otherwise, the algorithm uses the predicates $AB(n)$ which are part of the conflict in the checked set of $AB(n)$ predicates to choose the next $AB(n)$ to add to the set to avoid this conflict. Due to this conflict-driven search, the algorithm can derive a minimal set in an efficient manner [5]. To perform the necessary calculations of the algorithm we use the implementation of [9].

V. Rule Engine

After detecting a fault and identifying the faulty components the robot needs to react to this fault. To deal with faulty components the robot needs either to perform a repair action [3] or change the configuration of the robotic system [4] to deal with this problem. In either case, it takes some time to deal with the fault properly. This can cause the robot to operate in an unknown state in an unsafe manner. Thus, the robot needs first to react swiftly to bring the robotic system in a known a safe state. This imposes that the robotic system will not harm itself or its environment. Additionally, often such a reaction is sufficient as some faults cannot be fixed by the robot itself, e.g. a broken wheel.

To allow the robot to perform a fast reaction we propose a simple but powerful rule engine. The simplicity of the rule engine is not only due to the simple model how the robot should react but also due to the limited reasoning which is performed to choose the reaction. This restricts the possible reactions of a robot but allows to perform the reactions fast without a large computation overhead. The reaction triggered by the rule engine is a kind of reflex of the robot. Thus, only preventing it from further harm if possible.

To perform the reaction, the rule engine uses a set Obs of the observations made so far. The set is updated with each incoming observation to ensure that only one observation per component/topic for a specific type is present. This update also ensures that only the newest information is used. To trigger the rules an additional set is used, the set $PosAb$ of components which are possibly faulty. The set defines those components which are part of a minimal diagnosis. Thus, if one has two diagnoses $\{\{m_1\}, \{m_2\}\}$ the set of possibly faulty components consist of the elements of both diagnosis ($\{m_1, m_2\}$. This set simplifies reasoning as one

does not reason over different diagnosis but only over the set of components which may be faulty. The components which may be the faulty need either to be observed more closely or need to be repaired. Additionally, one cannot assume that this component works properly with the information given so far. Thus, this set is sufficient to decide which action to execute.

The rule engine consists of a set of rules $\mathcal{R}$ where each rule r is a tuple comprising the following elements.

- A set $posObs$ defining observations which should have been made
- A set $negObs$ defining observations which should not have been made
- A set $posPosAb$ which is a set of components which should have been diagnosed as possibly faulty
- A set $negPosAb$ which is a set of components which should not have been diagnosed as possibly faulty
- α an action to execute.

Due to the use of the sets, one can simply perform the reasoning by intersecting the sets to determine if the rule should be triggered. As some observations, may be missing one may face the problem that neither $obs_{resource} \in Obs$ nor $\neg obs_{resource} \in Obs$ holds, thus one cannot take a decision if the observation of the resource is true or false. If one would strictly perform the reasoning a rule may not triggered because $obs_{resource} \notin Obs$ holds although $\neg obs_{resource} \notin Obs$ holds. This is of special interest as not every observer may state regularly which observations are true but only state which observations are false. To deal with this problem we trigger a rule if no contradicting information is observed. This is achieved by the following simple procedure.

Trigger the rule if neither of the following holds.

- $\overline{posObs} \cap Obs \neq \emptyset$, where $\overline{posObs} = \{\neg po | po \in posObs\}$
- $negObs \cap Obs \neq \emptyset$
- $\overline{posPosAb} \cap PosAb \neq \emptyset$, where $\overline{posPosAb} = \{\neg posAb | posAb \in posPosAb\}$
- $negPosAb \cap PosAb \neq \emptyset$

As only set operations are performed one can perform an efficient reasoning which allows a fast reaction. Especially as one can assume that the sets $posObs$, $negObs$, $posPosAb$ and $negPosAb$ are small. Thus, one can perform this checks in $\mathcal{O}(|posObs| * log(Obs) + |negObs| * log(Obs) + |posPosAb| * log(PosAb) + |negPosAb| * log(PosAb))$ which allows a fast reaction even in case of many observations or many possible faulty components.

As rules, should only be used to allow the robot to react to faults, instead of continuously checking the rules, they are only checked if the set of observations or possible faulty components changes. This allows to save resources but also prohibits to trigger a rule multiple times without any change in the system.

After deciding that a rule should be triggered one needs to execute the action α which is defined for this rule. The actions range from printing a message to the console or to a log file over changing parameters to triggering the execution of an external script. Thus, one can trigger nearly arbitrary behavior to react to a fault.

VI. Use Case

Before we discuss related research, we will show a simple use case of the system. The use case is the simplified odometry calculation of a robot which delivery good in a warehouse, see [10] for a detailed description of the robot. The odometry is calculated using the wheel encoders and an IMU is used to improve this odometry. The IMU is fused with the calculation by using the rotation of the IMU instead of the calculated rotation. Thus, if the IMU is fault free the odometry is improved. To show the impact of the proposed system three faults is simulated. The IMU can either be stuck to zero after some time, it can overestimate the rotation by 20% or issue that there is no rotation after rotating a certain amount of time.

To evaluate the impact of the system the robot was commanded to move between six waypoints in the environment, for three minutes. During the movement, the wheel encoder the IMU measurements and the real position of the robot were determined. The real position of the robot was determined with the help of an OptiTrack system. After the movement of the robot was recorded the odometry is calculated using the wheel encoders and the IMU. Additionally, one observer is used which checks if the calculated rotation of the wheel encoder and the IMU correlate. This allows detecting a fault of the IMU. Using this fault detection, the diagnosis can calculate that the IMU is faulty. In such a case the rule engine changes a parameter to ensure that the IMU is no longer used for the odometry calculation.

The evaluation compares the error between the ground truth and the calculated odometry which always uses the IMU and the calculated odometry which only use the IMU if it is not diagnosed to be faulty. In case the IMU was stuck to zero after several seconds the mean error was reduced by 28.1% and the root mean squared error (RMS) was reduced by 39.9%. In case the IMU was overestimating the rotation by 20% the mean error was reduced by 25.6% and the root mean squared error (RMS) was reduced by 35.6%. In case the IMU was reporting zero rotation after one second of rotation the mean error was reduced by 39.3% and the root mean squared error (RMS) was reduced by 50.9%. Thus, the use of the diagnosis system could react quickly enough to improve the odometry calculation drastically. The evaluation was performed on an intel i5-2430M with 8 GB of RAM and took less than 2 % of the CPU.

VII. Related Research

We begin our discussion of related research with the method proposed in [11]. The method adds to each software module so-called software sensors. These sensors supervise the execution of a software component which is treated as a black box. Thus, the software component can be developed and tested independently from the sensors. During the execution, the software sensor checks for faults and report these faults on a diagnosis port. To ease the reuse of the

sensors these sensors uses interfaces which are specific to the type of information they are interested in, e.g. a state change in the component. The information provided by these sensors on the diagnosis port can afterward be used by a monitor. The monitor allows to view the sensing result and thus show which faults are present in the system. This contrasts with the method we propose in this paper as we use the information provided by the observer to calculate a diagnosis. Additionally, our observers allow checking for properties which need to hold between different components, e.g. the movement measured by the wheel encoder and by the IMU.

Another method to observe a robotic system was proposed in [12]. Each module in the system is accompanied with a detection module which checks if the module works as expected. This check is performed with the help of a residual calculation. If the residual is not zero a fault is detected. All the detected faults are gathered in a fault signature and used for fault identification. This identification is performed with the help of an incidence matrix. The matrix describes in a static manner which fault causes which observations. This contrasts with our approach as we do not assume that we can simply enumerate all possible observation and faults in a matrix. To react to a fault, the method presented in [12] reacts on the high-level which uses defined recovery actions, which are chosen per the severity of the fault. This is like our approach which use a simple rule engine to perform a reaction but delegates the fault handling to more complex reasoning whereas the rule engine allows a fast reaction.

A method which uses a rule system for observations was presented in [13]. The system defines safety rules which are checked during runtime. To define this rules a domain specific languages is used which allows defining conditions for the rules and which actions to trigger if a condition holds. The rules use information which is provided on different topics to define a safety rule. The actions are afterward executed on the robotic hardware and can be defined in the framework separately. The main difference to our system is that we separate the detection and the reaction to a fault. This allows us to use several observations to determine which component is faulty and afterward react depending on the faulty component.

As we have briefly outlined above our method is based on the method presented in [3]. The method presented in [3] also uses observers to detect a fault and a diagnosis engine to identify the fault component. Additionally, a planning system is used to repair if a fault is detected. Instead of using a planning system to find a proper repair we use a simple rule engine to allow the robot a fast reaction but also restricts the possible repairs which can be performed. To allow a fast reaction and a proper repair one can combine both methods and first react with the rule engine and afterward trigger a planning step for a proper repair. The other difference between the method presented in this paper and the method presented in [3] is the underlying implementation. The underlying implementation presented in this paper use plugin-based observers which are more efficient than the implementation of the observers used in [3].

VIII. Conclusion and Future Work

Autonomous robots perform tasks in a (partly) unknown environment. This is done by using several complex software and hardware components. These components need to proper function and properly interact with each other to allow the robot to achieve its task. Due to the complexity of the components and the (partly), unknown environment one cannot expect that the robot will perform its task without a fault. Instead one needs to address the problem of fault occurrence in the robotic system.

In this paper, we presented a model based approach which allows that the robot detects and identifies a fault. This is achieved by observing the communication between the components and checking this communication for specific properties. These properties are derived from the system and specify the proper function of the system. If a property indicates a fault a diagnosis engine is used to determine the minimal set of components which is faulty. Using the result of this diagnosis engine a simple rule engine can be used to allow the robot to react to a fault. This reaction can be used to repair the fault or to bring the robot in a safe state to perform a more complex repair action.

The current approach uses static properties of the system to determine if a fault has occurred. It is left for future work to extend this approach to also consider dynamic changes of the properties. This would allow to detect a malfunction in the dynamic behavior of the system as well as to determine a malfunction of a component which changes its static behavior per a defined system state.

References

[1] G. "Steinbauer and C. Mühlbacher, "Hands off - a holistic model-based approach for long-term autonomy," in *Workshop on AI for Long-Term Autonomy, 2016 IEEE International Conference on Robotics and Automation (ICRA)*.

[2] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[3] S. Zaman, G. Steinbauer, J. Maurer, P. Lepej, and S. Uran, "An integrated model-based diagnosis and repair architecture for ros-based robot systems," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 482–489.

[4] S. Loigge, "Unified and dependable robot control architecture based on ros," Master's thesis, Faculty of Computer Science and Biomedical Engineering, Graz University of Technology, 2016.

[5] R. Reiter, "A theory of diagnosis from first principles," *Artificial intelligence*, vol. 32, no. 1, pp. 57–95, 1987.

[6] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.

[7] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[8] J. Gertler, *Fault detection and diagnosis in engineering systems*. CRC press, 1998.

[9] T. Quartisch and I. Pill, "Pymbd: A library of mbd algorithms and a light-weight evaluation platform." in *25th International Workshop on Principles of Diagnosis (DX-2014)*, 2014.

[10] C. Mühlbacher, S. Gspandl, M. Reip, and G. Steinbauer, "Improving Dependability of Industrial Transport Robots Using Model-Based Techniques," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[11] A. Lotz, A. Steck, and C. Schlegel, “Runtime monitoring of robotics software components: Increasing robustness of service robotic systems,” in *Advanced Robotics (ICAR), 2011 15th International Conference on*. IEEE, 2011, pp. 285–290.

[12] D. Crestani, K. Godary-Dejean, and L. Lapierre, “Enhancing fault tolerance of autonomous mobile robots,” *Robotics and Autonomous Systems*, vol. 68, pp. 140–155, 2015.

[13] S. Adam, M. Larsen, K. Jensen, and U. P. Schultz, “Towards rule-based dynamic safety monitoring for mobile robots,” in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2014, pp. 207–218.

Visual Localization System for Agricultural Vehicles in GPS-Obstructed Environments*

Stefan Gadringer[1], Christoph Stöger[1] and Florian Hammer[2]

Abstract— **Accurate outdoor localization and orientation determination using the Global Positioning System (GPS) usually works well as long as the GPS antenna receives signals from a sufficient number of satellites. Especially in agricultural applications, the respective lines of sight are frequently obstructed due to the presence of trees. In this paper, we investigate the applicability of an alternative method for position and orientation estimation that is based on a stereo-camera system and Visual Odometry (VO). We have experimentally validated our approach in a logging road scenario. Based on the results of the position and orientation estimation, we discuss challenges of VO in such a non-trivial environment.**

I. Introduction

Localization of a vehicle is a very important task and hence a research topic for decades. In general, localization is possible with sensors like GPS, rotary encoder, IMU (Inertial Measurement Unit), laser scanner or a camera. Of course, there exist even more sensors and each one has its own pros and cons in terms of accuracy, drift, price, etc. The area of application highly depends on these properties. In this paper, we focus on outdoor localization in natural terrain. This is an important topic for precision farming [4], for example. Hereby, the question is always the same: Which sensors are suitable for the application?

As discussed in [25], a GPS antenna always needs intervisibility to several satellites to guarantee an accurate position estimation. This is sometimes impossible in areas like in a forest where trees occlude the satellites. The usage of wheel odometry via rotary encoders is not suitable as well due to problems with inaccuracies of the wheel geometry and slipping situations. In comparison, an IMU allows a good estimation of the orientation but not for the position because the double integration of the acceleration results in a high drift over time. A laser scanner has a very high position accuracy on the one hand but it is very expensive and not so well proofed for high vibrations on the other hand. Thus, just the camera remains of the sensors mentioned above. This sensor is relatively cheap but a position and orientation estimation via VO is normally linked with high computing demand and continuous growth of the drift per number of used images. Furthermore, overexposed images and other problems like branches that occlude cameras need a robust implementation of a VO to be able to get a valid pose estimation. However, this paper shall show the applicability of Visual Odometry to estimate position and orientation in different wooden environments with ambiguous natural structures.

This paper is structured as follows. Section II gives an overview of related work. Visual Odometry and all its components are explained in Section III. Finally, the experiments are shown in Section IV. Last but not least, Section V contains the conclusion as well as some remarks about future work.

II. Related work

Visual Odometry (VO) is the incremental estimation of the pose (position & orientation) via examination of the changes on images due to motion induction [24]. The research on VO already started in the early 1980s and one if its advantage is that no prior knowledge about the environment is necessary. A good example is the implementation of Cheng et al. [6], [21], which was used in the rover of the NASA Mars exploration program. Since then VO was continuously under research, which means that the literature about Visual Odometry is huge. Therefore, this section just contains an overview about relevant literature of VO for the localization of a vehicle in an outdoor environment.

Nister et al. [22] proposed one of the first real-time VO which was capable of a robust pose estimation over a long track. They use a stereo-camera system and detect Harris corner features [15] in the images. 3D points are estimated through triangulation of the corresponding features in a stereo pair. In a next step Nister et al. use these 3D points and the features of a following image to estimate the pose via a 3D-to-2D algorithm as described in [24]. RANSAC (Random Sample Consensus) [12] removes outliers in the motion estimation step. Regarding to Scaramuzza et al. [24], this VO procedure was a high improvement to previous implementations and is still used by many researcher.

Comport et al. [7] use a similar procedure but estimate the motion using 2D-to-2D instead of 3D-to-2D feature correspondences. With reference to Scaramuzza et al. this results in a more accurate pose because triangulation is not needed.

In [26], [17] or [27] bundle adjustment is applied to further reduce the drift of the Visual Odometry. Bundle adjustment optimizes the latest estimated poses using features over more than just two stereo pairs. Konolige et al. [17] show that this step reduces the final position error about a factor of two to five.

*Parts of this work have been supported by the Austrian COMET-K2 programme of the Linz Center of Mechatronics (LCM), and was funded by the Austrian federal government, and the federal state of Upper Austria.

[1]Stefan Gadringer and Christoph Stöger are with the Institute of Robotics, Johannes Kepler University, 4040 Linz, Austria {stefan.gadringer,christoph.stoeger}@jku.at

[2]Florian Hammer is with the Linz Center of Mechatronics GmbH, 4040 Linz, Austria florian.hammer@lcm.at

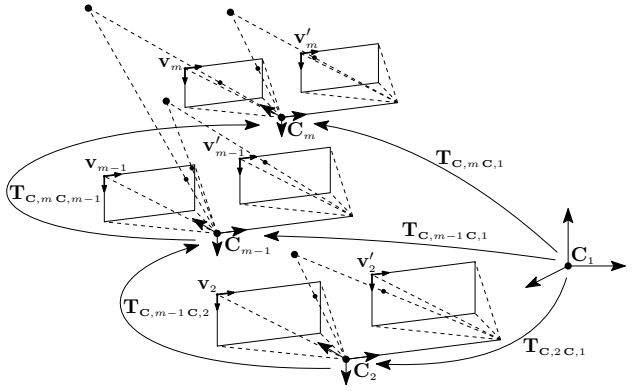


Fig. 1: Illustrated VO problem of a stereo system (relative transformations $\mathbf{T}_{\mathbf{C},m-1\,\mathbf{C},2}$, $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1}$ / absolute transformations $\mathbf{T}_{\mathbf{C},2\,\mathbf{C},1}$, $\mathbf{T}_{\mathbf{C},m-1\,\mathbf{C},1}$, $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},1}$)

Furthermore, the usage of additional sensors like GPS, laser scanner or IMU can improve the pose estimation. For example, in [1], [23], [17] or in [27] the integration of an IMU reduces the error in orientation. In [17] Konolige et al. achieve with their implemented real-time VO a maximum relative position error of just 0.1 % over a 9 km long track. Another good result is shown by Tardif et al. [27] over a 5.6 km long track. This dataset was acquired by a tractor driving next to an orange grove and on a street for the return to the garage.

III. Visual Odometry

As discussed in Section II, Visual Odometry incrementally estimates the pose. Figure 1 shows this for a typical case using a stereo-camera system. The calculation of a relative homogeneous transformation $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1} \in \mathrm{SE}(3)$ of an image pair $\{m-1, m\}$ with camera centers / camera coordinate systems $\mathbf{C}_{m-1}$ and $\mathbf{C}_m$ is done via features in the images. As shown in the figure, the coordinate system of the left camera is the reference point of a transformation $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1}$, which transforms from $\mathbf{C}_{m-1}$ to $\mathbf{C}_m$. The rigid body transformation is given by

$$\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1} = \begin{bmatrix} \mathbf{R}_{\mathbf{C},m\,\mathbf{C},m-1} & {}_{\mathbf{C},m}\mathbf{t}_{\mathbf{C},m\,\mathbf{C},m-1} \\ 0 & 1 \end{bmatrix} \tag{1}$$

where $\mathbf{R}_{\mathbf{C},m\,\mathbf{C},m-1} \in \mathrm{SO}(3)$ is the orthogonal rotation matrix and ${}_{\mathbf{C},m}\mathbf{t}_{\mathbf{C},m\,\mathbf{C},m-1} \in \mathbb{R}^3$ the translation vector, represented in the coordinate system $\mathbf{C}_m$. The concatenation of all relative transformations results in the absolute transformation $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},1} = \mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1}\mathbf{T}_{\mathbf{C},m-1\,\mathbf{C},1}$ from $\mathbf{C}_1$ to $\mathbf{C}_m$.

Therefore, the main task of a VO is to calculate the relative transformations $\mathbf{T}_{\mathbf{C},m\,\mathbf{C},m-1}$ and finally to concatenate them to get the full camera trajectory $\mathbf{T}_{\mathbf{C},m:\mathbf{C},1} = \{\mathbf{T}_{\mathbf{C},2\,\mathbf{C},1}, \ldots, \mathbf{T}_{\mathbf{C},m\,\mathbf{C},1}\}$ between the camera centers $\mathbf{C}_1$ and $\mathbf{C}_m$.

The structure of our VO approach is similar to the one of Nister et al. [22] and it starts with the feature detection and description but it uses the more distinct features A-KAZE [2] instead of Harris [15]. The next step is to match features between a stereo pair and one consecutive image, either left or right. Then, the triangulated stereo correspondences and the matched $2D$ features are used for the pose estimation. At the end, key frames are selected and windowed bundle adjustment [28] is applied to further optimize the previous calculated poses [27].

A. Feature Detection and Description

Feature detection is one of the most important steps in a feature-based Visual Odometry system. Regarding to Fraundorfer [13], important properties of features are detection repeatability, localization accuracy, robustness against noise as well as computation efficiency. In [8], Cordes et al. compare many different detection algorithms and the detector A-KAZE [2] proofs to be the best candidate in terms of localization accuracy and suitable number of detected features. This detector is implemented in OpenCV [5] and is an extension of the algorithm KAZE [3] to detect blobs. In general, these features are image patterns with different intensity, color and texture compared to its adjacent pixels and they are more distinctive than corners [13]. This is especially important in natural environment with ambiguous structures like branches or leaves. In our case, A-KAZE detects blobs in a nonlinear scale space with four octaves and the same amount of sub-levels.

In addition to the detection algorithm, A-KAZE also provides one for the description of a feature, which is implemented in OpenCV as well. It converts the area around a feature into a binary descriptor which has a length of 486 bit. Every comparison between two areas results in three bit. The description algorithm of A-KAZE is called M-LDB (Modified-Local Difference Binary) and is rotation and scale invariant. According to Alcantarilla et al., A-KAZE allows efficient and successful feature matching, which are mandatory properties of a good descriptor.

B. Feature Matching

The task of this step is to find feature correspondences among images. The easiest way to achieve matching between two images is to compare all feature descriptors of the first image with every other descriptor of the second one. This search is quadratic in the number of features. Fortunately, the usage of epipolar or motion constraints simplifies this task and reduces the computation time drastically. This is necessary to facilitate an online VO system, which could be used on a vehicle like a tractor during its operation in a field or forest.

Our stereo VO relies on rectified images, which are remapped image pairs with horizontal and aligned epipolar lines to each other (see [13]). Thus, epipolar matching just allows a match between features which lie on the same horizontal epipolar line or rather image row.

Descriptors of two consecutive left or right images can be matched via a motion constraint. As proposed in [10], we assume a constant velocity model between two frames. Using the known motion, we can project the $3D$ point of a already matched stereo correspondence into the other image. A constant window of $2 \cdot 35 \times 2 \cdot 35$ pixel around the projected position defines the allowed area of possible features and therefore reduces the computing time.

The comparison between two binary descriptors itself is done via calculating the Hamming distance [14], which is the number of different bits and a very efficient operation. Normally, the descriptor with the minimum Hamming distance is chosen as the best match. To improve the robustness of the matching, we additionally apply the distance-ratio-test as proposed in [20]. It just accepts a match if the ratio between the two closest neighbors is below a threshold $r_{max} \in \mathbb{R}$ with $0 < r_{max} < 1$. Using binary descriptors, the ratio $r_H \in \mathbb{R}$ between two descriptors is defined as

$$r_H = \frac{d_{H,1}}{d_{H,2}} < r_{max}, \tag{2}$$

where $d_{H,1} \in \mathbb{N}$ and $d_{H,2} \in \mathbb{N}$ are the Hamming distances of the two closest neighbors, respectively. In our case, we use an empirical threshold of $r_{max} = 0.71$ which helps to remove ambiguous matches that can occur at repeatable structures like branches.

C. Motion Estimation and Key Frame Selection

In this step, the calculation of the relative camera motion, i.e. the relative transformation $\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}$ between an image pair $\{m-1,m\}$, takes place. Therefore, we use calibrated stereo-cameras and two sets of corresponding features F_{m-1} and F_m of the images $m-1$ and m, respectively.

For the 3*D*-to-2*D* algorithm, the features of F_{m-1} are defined by 3*D* points in $\mathbf{C}_{m-1}$ and the one of F_m by 2*D* image points [24]. Normally, we use 2*D* features of the left image with coordinate system $\mathbf{v}_m$. Alternatively, if the motion estimation fails due to less feature matches, features of the right image with coordinate system $\mathbf{v}'_m$ can also be used to prevent a failure of the VO. The estimation of the 3*D* points is done via the linear triangulation method of Hartley and Zissermann [16], which is implemented in OpenCV [5]. Using a function $\mathrm{d_E}$ to calculate the Euclidean distance [11], the transformation $\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}$ can be found through minimizing the image reprojection error of all features

$$\min_{\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}} \sum_{i=1}^{n} \mathrm{d_E}\left({}_{\mathbf{v},m}\mathbf{t}_{\mathbf{v},m\mathbf{X},i},\, {}_{\mathbf{v},m}\hat{\mathbf{t}}_{\mathbf{v},m\mathbf{X},i}(\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1})\right)^2. \tag{3}$$

Thereby, ${}_{\mathbf{v},m}\mathbf{t}_{\mathbf{v},m\mathbf{X},i}$ is the 2*D* coordinate vector of the image point $\mathbf{x}_i$ and ${}_{\mathbf{v},m}\hat{\mathbf{t}}_{\mathbf{v},m\mathbf{X},i}$ the image coordinate vector of the 3*D* point $\mathbf{X}_i$, which is observed in $\mathbf{C}_{m-1}$ and projected through $\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}$ and the corresponding camera projection matrix [16] into image m. Equation (3) can be solved using at least three 3*D*-to-2*D* correspondences, is known as P3P (Perspective from three Points) and returns four solutions. Therefore, at least one another point is necessary to get a single and distinct solution. PnP-algorithms (Perspective from n Points) like EPnP (Efficient PnP) [18] use $n \geq 3$ correspondences to solve the problem. Normally, these methods just calculate accurate results if the used correspondences are correct. If this is not guaranteed, the well known procedure RANSAC (Random Sample Consensus) [12] should be used to remove wrong correspondences, so called outliers. In [13], such a robust motion estimation using RANSAC is explained more in detail. Our VO uses EPnP for the pose estimation and a preliminary non-minimal RANSAC with five points to acquire trustworthy results of the outlier removal as suggested by Fraundorfer et al. [13].

If the first motion estimation with the left image fails due to less feature matches, or the motion is implausible (position or orientation is unrealistic), then the estimation is retried with 2*D* features of another image as a backup. The order of these images is the following. Firstly, the right image of the actual stereo frame is used. If the motion estimation with the features of this image is also unsuccessful, then a consecutive still unused left or right image is used until the motion estimation step is successful. This procedure avoids a failure of the VO with high probability.

The selection of key frames is another important component of our VO. In general, the drift of a VO increases with every frame, i.e. every relative motion, which is used for the update of the absolute motion. Therefore, the concatenation of small motions should be avoided to keep the drift as low as possible. This means that the transformation $\mathbf{T}_{\mathbf{C},m\mathbf{C},m-1}$ should not be used to update the absolute transformation $\mathbf{T}_{\mathbf{C},m\mathbf{C},1}$ if the motion between the image pair $\{m-1,m\}$ is small or even zero. Instead, we should stay with $\mathbf{T}_{\mathbf{C},m-1\mathbf{C},1}$.

We define a stereo frame m as a key frame $\overline{m}$ if its relative transformation is used for the absolute motion update. Our defined requirement is that the relative change in position is bigger than 2 m or the relative angle of rotation [9] is bigger than 20°.

D. Bundle Adjustment

Windowed bundle adjustment [28] is the last important step in our feature-based VO system. It is used to optimize the relative transformations of the most recent $\overline{M}$ key frames. For simplicity, we assume n 3*D*-points $i \in \{1,\ldots,n\}$, which are seen in a window of $\overline{M} \leq \overline{m}$ key frames $j \in \{\underline{m},\ldots,\overline{m}\}$. Hereby, the index of the oldest stereo frame in the window is defined as $\underline{m} = (\overline{m} - \overline{M} + 1)$. To reduce the computation demand, our VO just uses a window with the most recent $\overline{M} = 2$ key frames, i.e. in total the features of four images are used for the optimization.

Bundle Adjustment is, like in (3), again the minimization of the image reprojection error and is given by

$$\min_{\mathbf{T}_{\mathbf{C},j\mathbf{C},1},\,{}_{\mathbf{C},1}\mathbf{t}_{\mathbf{C},1\mathbf{X},i}} \sum_{i=1}^{n} \sum_{j=\underline{m}}^{\overline{m}} \mathrm{d_E}\left({}_{\mathbf{v},j}\mathbf{t}_{\mathbf{v},j\mathbf{X},i},\, {}_{\mathbf{v},j}\hat{\mathbf{t}}_{\mathbf{v},j\mathbf{X},i}(\mathbf{T}_{\mathbf{C},j\mathbf{C},1},\, {}_{\mathbf{C},1}\mathbf{t}_{\mathbf{C},1\mathbf{X},i})\right)^2. \tag{4}$$

Thereby, ${}_{\mathbf{v},j}\mathbf{t}_{\mathbf{v},j\mathbf{X},i}$ and ${}_{\mathbf{v},j}\hat{\mathbf{t}}_{\mathbf{v},j\mathbf{X},i}$ are, respectively, the vectors of the observed and estimated 2*D* coordinates of point i in key frame j. Due to the projection of the point $\mathbf{X}_i$ into the image plane, the estimated coordinates are dependent on the absolute transformations $\mathbf{T}_{\mathbf{C},j\mathbf{C},1}$, the 3*D* coordinate vector ${}_{\mathbf{C},1}\mathbf{t}_{\mathbf{C},1\mathbf{X},i}$ and the corresponding camera projection matrices. The camera parameters are assumed as constant and known via a prior calibration. The minimization of (4) is done using the sparse bundle adjustment library of Lourakis et al. [19].

Fig. 2: Vehicle with measurement setup and DGPS-receiver

IV. Experimental validation

Our realistic dataset shows the performance of our VO on a track through a forest. It contains GPS data as well as images during a drive of a truck on a logging road. The vehicle used for the measurement is further discussed in Section IV-A and sample images of the road can be seen in Section IV-B.

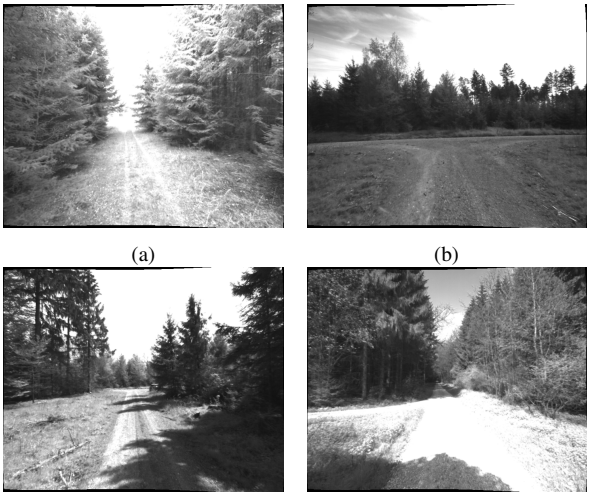
A. Setup

A small truck, equipped with a stereo-camera system, was used for the measurement. The cameras are mounted on the back of the driver's cab via aluminum profiles and magnets. This mounting position guarantees a good viewpoint backward without having unwanted objects within the field of view. In addition to the cameras, a DGPS-unit (Differential Global Positioning System) is used for ground truth although the signal strength lacks inside the dense forest.

The vehicle and its measurement setup is shown in Fig. 2. The cameras are mounted parallel on an aluminum profile at a distance of approximately one meter. The 12 V battery of the truck powers both cameras inside the wired box. Two Gigabit Ethernet cables facilitate the data transfer of the stereo-camera system, which operates at 10 Hz. A higher sample rate of the cameras is unsuitable due to the high computing time of the VO. We used the following sensors and devices:

- 2× JAI GO monochrome-cameras (JAI GO-5000M-PGE) with a maximal sampling rate of 22 Hz with the full resolution of 2560×2048 pixel
- 1× DGPS-system with open sky localization error of ca. 2 cm/0.1°
- 1× Xsens MTi-30 IMU with 400 Hz sampling rate (additional sensor for further experiments)
- Lenovo Thinkpad S540 with Intel Core i7-4510U CPU @ 2.00 GHz and 16 GB RAM
  Windows 7 Professional SP1 - 64 Bit

The JAI GO cameras allow a maximum resolution of 2560×2048 pixel. Due to lots of bumps on the logging road, the long exposure time of the cameras might blur images at darker areas of the forest. Therefore, we use 2×2 pixel binning and a resulting resolution of 1280×1024 pixel to decrease the exposure time. The resolution of the images is further decreased to 640×512 pixel by software to reduce the computing time of feature detection and description. After the decrease of the resolution, a rectification of these images is also done.



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

Fig. 3: Sample images of the driven logging rode

(a) Left stereo image (b) Right stereo image

Fig. 4: Stereo image pair with branch occlusion

B. Experiments

Our dataset contains two different drives of the presented vehicle on a logging rode and illustrates a realistic performance of our VO system. Figure 3 shows some road sections of our scenarios. Widespread areas and overexposed images may result in an inhomogeneous distribution of features, which is a big challenge for the VO.

The first scenario of our dataset is a 3×75 m long test drive on the part of the logging road, which is shown in Fig. 3a. In this dense forest area, our proposed VO demonstrates its robustness against overexposed images and occluded cameras like shown in Fig. 4, where a branch occludes the left camera entirely. The implementation is robust enough to handle such situations and still estimates a valid pose. The results of our test drive are presented in Fig. 5. The starting point is marked with a circle. Due to the low signal quality of the GPS, the reference position exhibits some inconsistencies. The plotted coordinate system is the one of the GPS with X pointing to East, Y to North and Z upwards.

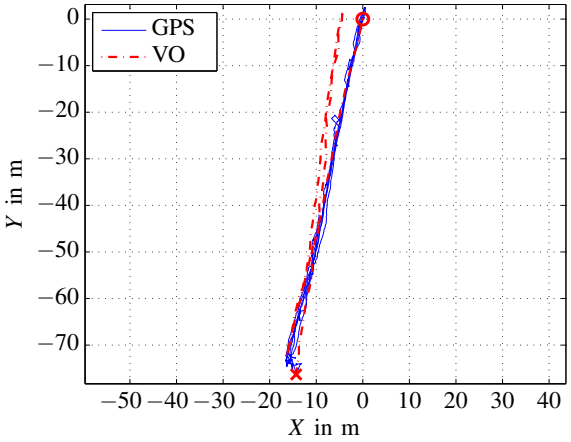


Fig. 5: Scenario 1 – Comparison of the estimated trajectory with GPS

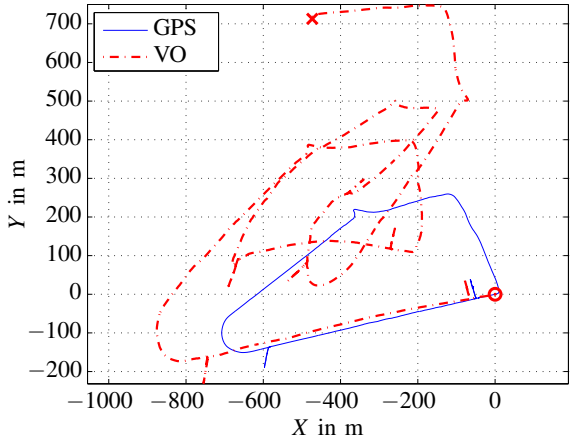


Fig. 6: Scenario 2 – Comparison of the estimated trajectory with GPS

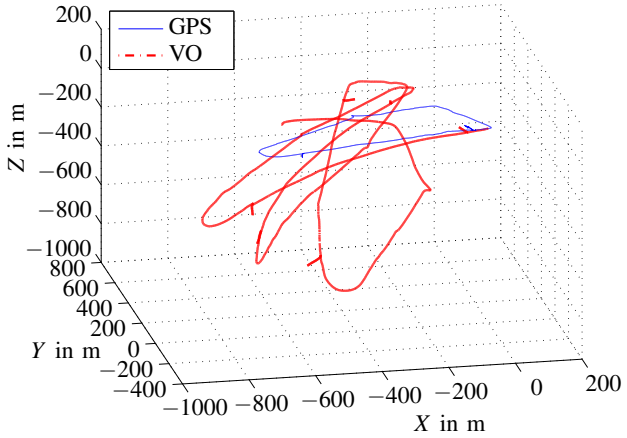


Fig. 7: Scenario 2 – Comparison of the estimated 3*D*-trajectory with GPS

As shown in Fig. 5, the estimated pose of the first 75 m fits very well with ground truth. The estimated trajectory of the return slightly deviates from the GPS reference. The inaccurate estimation of the orientation happens due to occlusions of the left camera like it is shown in Fig. 4. However, our robust VO prevents a total failure and still allows a valid but slightly inaccurate pose estimation via using images of the right camera instead. The third track of the logging road is estimated well as a straight line again.

Using the mentioned laptop, the computation time of our off-line VO of this scenario is about 0.529 s per stereo pair. This time duration is increased due to the occlusion of the left camera, which acquires the additional processing of the right image instead of just the left one. This problem especially happens at the return of the vehicle because the cameras are mounted on the back of the driver‘s cab.

The second scenario of our dataset is a 3×2169 m long drive of the presented vehicle on a logging road. This scenario should deliver an answer about the drift behavior of our implemented VO. Figure 3b represents the first image of this sequence. The results are shown in Fig. 6. The estimated trajectory is inconsistent with the ground truth and just the first 2169 m long loop can be identified somehow. Then, the trajectory continuously deviates from the driven track. If we look closely at the start of Fig. 6, it shows that distances are estimated too large in general. The whole trajectory seems to be scaled compared to the original track.

For a better understanding of the results, it is helpful to further investigate the 3*D*-trajectory illustrated in Fig. 7. Referring to the estimated VO path of this figure, from the beginning the truck starts to move downwards and also to twist sideways. This results in a distorted trajectory instead of a more or less planar movement of the truck.

The explanation of the occurrent problem can be found with a closer look at the features, which are used for the pose estimation. Figure 8 shows the detected A-KAZE features of Fig. 3b, and the sweeping area only contains a few key points. Most of them are found at the treetops in the upper half of the image. In the worst-case scenario, for example if all trees have the same height, all features are just on one line instead of being well distributed in the image. The outcome of this is an ill-conditioned pose estimation and hence an inaccurately estimated distance and pitch-angle. The problem of this scenario is that image positions hardly change by a further increase of the distance.

However, as shown in Fig. 9, the yaw angle can be estimated well because a planar rotation definitely changes the image positions of these features. The figure clearly illustrates every turn of the track and the good consensus of the yaw angle for each loop. Just some minor deviations due to different drive behavior and drift can be seen. This means that Fig. 9 shows the potential of our implemented VO for applications which mainly rely on a good estimation

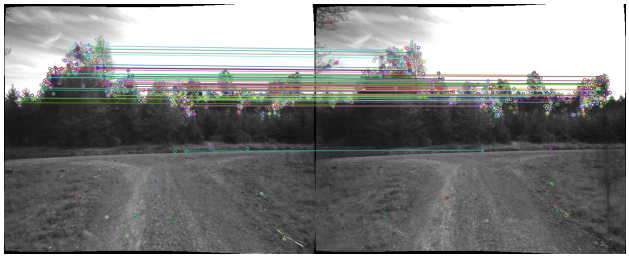


Fig. 8: Features of two left images used for pose estimation

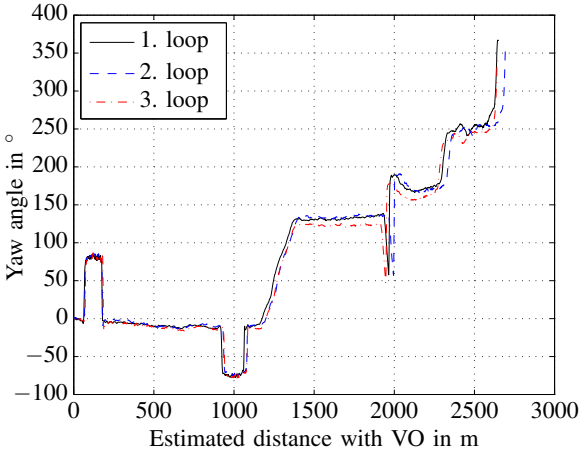


Fig. 9: Scenario 2 – Comparison of estimated yaw angles

of the yaw angle like it's necessary for agricultural vehicles.

Using the given laptop, in average the computing time of the pose estimation takes 0.349 s per stereo pair. This equates to approximately three pose estimations per second.

V. Conclusion and Future Work

We developed a Visual Odometry system that is based on a stereo-camera pair and capable of estimating the position and orientation of an agricultural vehicle in GPS-obstructed environments. We deployed our system on a small truck and carried out measurements for two different logging road scenarios. The results show that an insufficient distribution of features can lead to an ill-conditioned pose estimation, and hence to an inaccurately estimated distance and pitch angle. Due to the incremental concatenation of relative motions, this results in an increased error in position. However, our robust VO system is highly capable of estimating the orientation (yaw angle) with acceptable accuracy in unstructured environment. This is especially shown in the first scenario in the dense forest where the signal quality of GPS lacks.

Future work includes the improvement of the distribution of features and hence the pose estimation. Uniformly distributed features could be achieved via using different detector parameters for the upper and lower half of the image.

Another goal is to reduce the computing time of the VO to facilitate an online system. This can mainly be done via the parallelization of repeatable tasks like feature detection and description.

Furthermore, the next steps include the incorporation of the data of an IMU that were recorded simultaneously during our measurements. We plan to use a data fusion algorithm such as a Kalman Filter to improve the overall accuracy by combining the VO with the IMU data.

References

[1] M. Agrawal and K. Konolige, "Rough terrain visual odometry," in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, vol. 1, 2007, pp. 28–30.

[2] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *British Machine Vision Conference (BMVC)*, 2013.

[3] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," in *Computer Vision–ECCV*. Springer, 2012, pp. 214–227.

[4] S. Blackmore, "Precision farming: An introduction," *Outlook on Agriculture*, vol. 23, no. 4, pp. 275–280, 1994.

[5] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Y. Cheng, M. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, 2005, pp. 903–910.

[7] A. I. Comport, E. Malis, and P. Rives, "Accurate quadrifocal tracking for robust 3d visual odometry," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 40–45.

[8] K. Cordes, L. Grundmann, and J. Ostermann, "Feature evaluation with high-resolution images," in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2015, pp. 374–386.

[9] E. B. Dam, M. Koch, and M. Lillholm, *Quaternions, Interpolation and Animation*. Datalogisk Institut, Københavns Universitet, 1998.

[10] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, pp. 1403–1410.

[11] M. M. Deza and E. Deza, *Encyclopedia of Distances*. Springer, 2009.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] F. Fraundorfer and D. Scaramuzza, "Visual odometry, part II: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[14] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[15] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of 4th Alvey Vision Conference*, 1988, pp. 147–151.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003.

[17] K. Konolige, M. Agrawal, and J. Sola, "Large scale visual odometry for rough terrain," in *Robotics Research*. Springer, 2011, pp. 201–212.

[18] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate $o(n)$ solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[19] M. I. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software (TOMS)*, vol. 36, no. 1, p. 2, 2009.

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[22] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2004, pp. I–652.

[23] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[24] D. Scaramuzza and F. Fraundorfer, "Visual odometry, part I: The first 30 years and fundamentals," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[25] T. Strang, F. Schubert, S. Thölert, R. Oberweis, M. Angermann, B. Belabbas, A. Dammann, M. Grimm, T. Jost, S. Kaiser, *et al.*, "Lokalisierungsverfahren," Deutsches Zentrum für Luft-und Raumfahrt (DLR), Tech. Rep., 2008.

[26] N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel, "Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle," in *Autonome Mobile Systeme*. Springer, 2006, pp. 157–163.

[27] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4161–4168.

[28] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment – a modern synthesis," in *Vision Algorithms: Theory and Practice*, ser. Lecture Notes in Computer Science (LNCS), B. Triggs, A. Zisserman, and R. Szeliski, Eds. Springer-Verlag, 2000, vol. 1883, pp. 298–372. [Online]. Available: https://hal.inria.fr/inria-00590128

Development of a fully Automated tuning system for organ pipes*

Clemens Sulz[1] and Markus Trenker[2]

***Abstract*— Many pipe organs consist of thousands of pipes, divided basically into two different types: flue pipes and reed pipes. Because of the fact, that the principle of sound generation differs, reed pipes must be tuned by hand periodically, which is a time-consuming and thus expensive process. The aim of this project was to do a feasibility study, to determine if this tuning process can be automated and to build up several prototypes for extensive testing. Thereby different actuator technologies were examined and evaluated. Finally a very cheap and compact actuator solution was developed. Appropriate software for controlling the system was programmed and the required drive electronics were developed. Tests with the prototypes have shown that the system is able to perform the tuning process in much shorter time than a human being with satisfying precision.**

I. INTRODUCTION

The pipe organ, called the king of instruments, has fascinated people for hundreds of years. It is the only instrument, which is played by feet and hands simultaneously, produces a huge range of tone colors and covers the whole frequency spectrum of the human hearing. Pressing a key causes air to stream into specific pipes, whereby each pipe produces one tone with a determined tone pitch and timbre. There can be thousands of pipes in a single pipe organ, with each pipe producing a unique sound.

Basically, two types of pipes are used in pipe organs: flue pipes and reed pipes (left side of Fig. 1). The sound of the flue pipes is generated in the same way as in a real flute. The air stream strikes against the lip and begins oscillating with a specific frequency. The result is a standing wave or vibrating column of air inside the pipe body. These are the facade pipes a beholder can generally see in a church and which represent the majority of the pipe stock.

The pipes of the other type, reed pipes, work in a completely different way and are hidden inside the organ. Within the pipe foot there is a metal tongue, which begins to oscillate, if air flows through the pipe (right side of Fig. 1). The so-called tuning spring is used to adjust the pitch of the reed pipe, because it defines the oscillatory length of the tongue. The tone color of reed pipes allows imitating trumpets, clarinets, oboes or other wind instruments.

*This work was supported by Rieger Orgelbau GmbH, Schwarzach

[1]Clemens Sulz, MSc wrote his Master Thesis about this topic and got his degree as MSc in Engineering at University of Applied Sciences, FH Technikum Wien, Vienna in 2016 clemenssulz@yahoo.de

[2]DI Dr. Markus Trenker supervised this Master Thesis and lectures at the Institute for Advanced Engineering at University of Applied Sciences, FH Technikum Wien, 1200 Vienna markus.trenker@technikum-wien.at

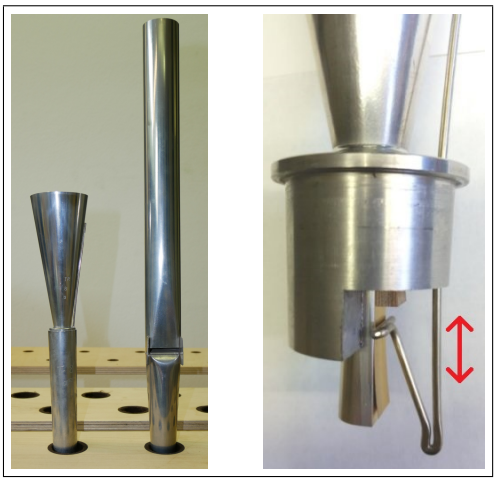


Fig. 1. Left picture: both types of organ pipes (reed pipe and flue pipe); Right picture: inner parts of reed pipe (tuning spring is moved to tune the pipe)

II. PROBLEM DESCRIPTION

The pitch of flue pipes depends directly on the velocity of sound, which in turn depends on air temperature. So the pitch is lowered, if the temperature is decreased and vice versa. Because of the fact, that in reed pipes the tongue oscillates and not the air, the pitch of these pipes stays almost constant. A temperature change of just 1-2°C causes an audible detuning of the organ. Not least because of the lower number of reed pipes and their easier tunability, the pitch of the reeds is tuned to the pitch of the flue pipes. To tune the pipes the tuning spring has to be moved up or down for each single pipe. Generally this tuning process requires two people (one sitting at the keyboard pressing down the keys and one tuning the pipes) and takes between a few hours and several days for big organs. Because of the associated expense, the reed pipes often are not in tune and are not used by the organist. The aim of this project was to develop a system, which can tune reed pipes automatically. Refined, the aim was to determine whether a technical system is basically able to tune the reed pipes with satisfying precision in an acceptable amount of time. Furthermore, because of the number of reed pipes (usually a few hundred) the solution should be very cost-effective. Especially the little reed pipes were a challenging object of research due to the high sensibility of the tuning spring. Thereby movements of less than a micrometre are required to adjust the pitch exactly enough. The final stage of the project was to build a few prototypes to test and demonstrate the abilities of the system.

III. STATE OF THE ART

At the beginning of this project an extensive market and patent review was done to find out, if any similar applications are already on the market. Thereby a few patents for automated organ tuning were found belonging to the German organ builder Voigt ([5], [7] and [6]). Furthermore, a project within the framework of a bachelor thesis by Fachhochschule Kiel [2] was found. But in contrast to the idea of automating the tuning process for reed pipes all these applications are developed to modify the pitch of flue pipes, whereby these projects are primarily concerned with conceptual studies. The company Rieger Orgelbau [1], which was the main cooperation partner for this project, has developed a system, which allows the tuning person to control the organ with a smartphone app. Specifically it is possible to play the keys of the organ via the smartphone, so the second person is no longer needed. This system represented the newest state of technology at the beginning of this project. If an actuator, which should be developed in the course of this project, would be combined with this system, a fully automated tuning application would be established.

IV. ACTUATOR RESEARCH

Following the analysis of the mentioned system a research on actuators was performed. Thereby the most important criteria were cost efficiency and space requirements. Furthermore the components and structure of reed pipes should not be modified, or if it is unavoidable, as little as possible. This would make it feasible to upgrade already existing organs with the tuning system. Before the research took place, force investigations on various tuning springs on three different pipes of different size were conducted to determine, how much force an actuator should be able to apply. The highest value which was measured was 6,0N. Including an appropriate safety surcharge for the following research a minimum guide value of 10N was defined.

A. Piezoelectric drives

Because of the required precision, piezoelectric drives were examined as a first step. One possible new type of piezo drive is the motor X15G (Fig. 2) from Elliptec [3]. If the piezo crystal inside this actuator is driven by the natural resonant frequency of the whole actuator, the rotor begins to move forward. With a second specific frequency the motor could also be moved backwards. According to the datasheet

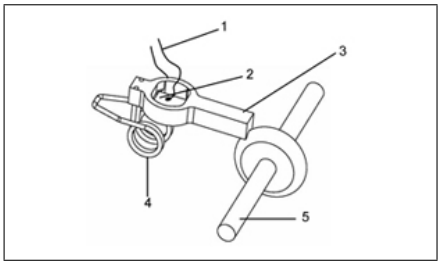


Fig. 2. Piezo motor X15G; 1...wires, 2...piezo ceramic, 3...resonator, 4...spring, 5...rotor [3]

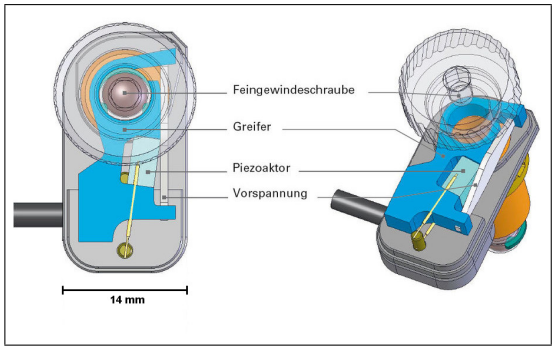


Fig. 3. Piezomike [4]

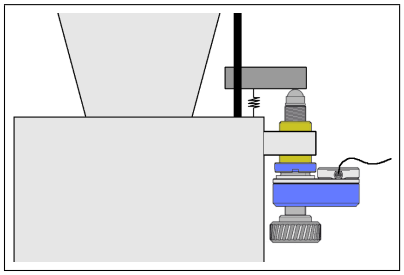


Fig. 4. Piezomike implemented on reed pipe

the drive can also be used as a linear actuator, whereby the drive could be attached directly to the tuning spring to move it up or down. Unfortunately, it became apparent that this drive can only raise 1.2N, which is much too little for this application.

A second piezoelectric drive, which was investigated, is the Piezomike (Fig. 3) from PI GmbH [4]. With 20N thrust, it would be strong enough for the tuning application. The piezo crystal inside the actuator is expanded slowly because of the controlled increase of voltage, whereby the gripper starts rotating the screw. If the final position is reached, the voltage is switched off and the gripper goes back to the starting position jerkily without moving the screw. Right side of Fig. 4 shows a schematic diagram for a possible implementation on a reed pipe with a spring, whereby the tuning spring is pulled against the Piezomike. A resulting advantage with this kind of drive would be the possibility to tune the pipe manually through rotation of the screw shaft without disassembling the tuning system. Unfortunately the high price of 500$/pcs. inhibits the application in this project.

B. Stepper motors

Because of the possibility of fine positioning of stepper motors, these drives were investigated following the piezo drives. Stepper motors with premounted threaded-spindle shafts were examined in detail. A possible application is illustrated in Fig. 5. The spindle nut in combination with a connected adapter part transforms the rotation into translational movements for the tuning spring. This solution could bring up the required forces, but because of the centric motor shaft and the frame size of a stepper motor this motor type would not fit between the pipes.

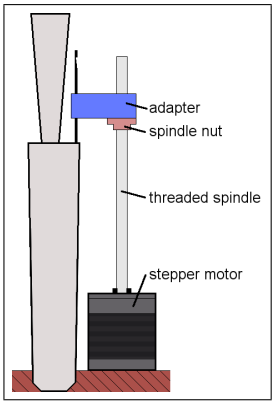


Fig. 5. Stepper motor with threaded shaft as tuning device

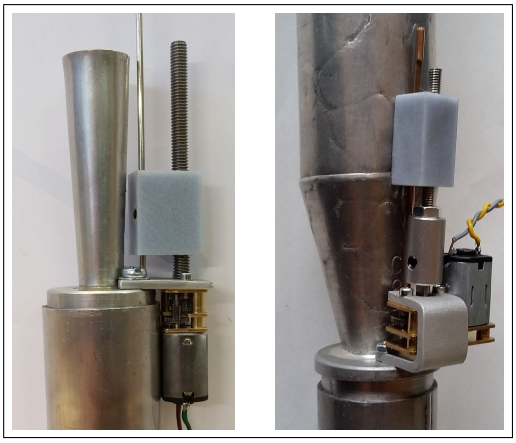


Fig. 7. Prototypes with compact gearbox motors

C. DC gearbox drives

As a smaller and cheaper alternative to stepper motors DC gearbox motors were explored. Thereby the structure with a threaded shaft, as in the last section on stepper motors, should be used. Due to the gear reduction such a motor could be significantly smaller. Also the drive electronics would be simpler to implement.

V. PRACTICAL REALIZATION

In the following section the implemented solution is described in detail.

A. Implemented drive technology

Owing to the fact, that the last described drive technology with a gearbox motor seems to be the best one, it was chosen to build a prototype and for further evaluation. An appropriate gearbox motor was available, wherefore this drive was used for a first prototype (Fig. 6). For testing the prototype, software, which will be described in section V-C, was developed in parallel. With the experimental setup, first successes in tuning the pipe were achieved. Nonetheless searching for alternative gearbox drives was continued, whereby a very compact and cheap gearbox motor was found, which is perfectly suited to the tuning application. This drive already has a threaded metric output flange. Furthermore, there is an alternative "'Flip-Type"' of this motor available. Thereby the input and output shaft are on the same side of the gearbox. This allows a more compact design and the reduction of the distance between the tuning spring and the threaded shaft. In Fig. 7 prototypes with both kinds of motors are pictured. Note that the left prototype contains the same pipe as in Fig. 6 to enable one to see the difference in size. To transform the rotating movement of the spindle into translation for the tuning spring, an adapter component of high-strength plastic was manufactured. In this component the tuning spring is fixed with a grub screw. If this single screw is loosened, the pipe can be tuned manually without disassembling the automatic tuning system. Thus an excellent, mechanical solution for the tuning system was found. Because of the low thread pitch, high precision positioning in micrometre range creates no problems for this actuator system. Additionally, the high gear reduction results in a very low drive torque needed for the motor. Performed force measurements showed, that the solution can generate about 60N, which is 10-times more than required.

To move the tuning spring and correct the pitch of the pipes, a control loop is necessary. This loop must contain the final control element or actuator, which executes the calculated move, frequency detection and a logic unit or software, which processes the frequency measurements. In Figure 8 such a control loop is depicted.

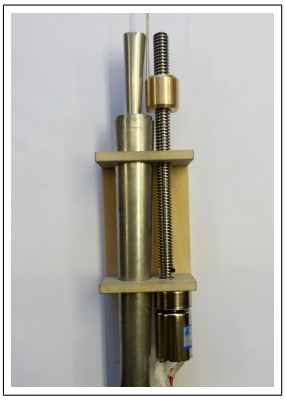


Fig. 6. First prototype with gearbox motor

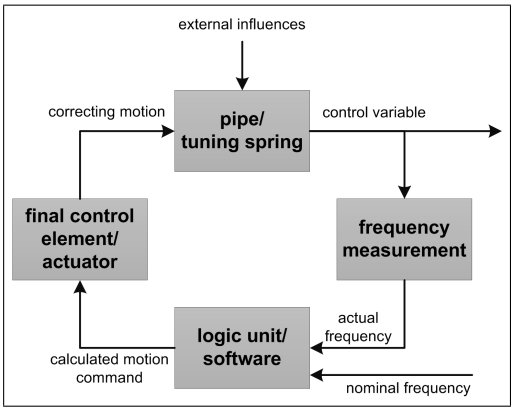


Fig. 8. Control circuit for automated pipe tuning

B. Frequency detection

At the beginning of this project the bought-in tuning device TLA CTS-32-C [8] was used for pitch detection. It communicates with the software part over an USB-Interface and was especially developed for organ builders and their needs. Because of the high price of the tuning device, an own solution for detecting the frequency was developed. Using a variable bandpass filter, it is possible to extract a sinusoidal wave with the fundamental frequency of the pipe from a complex audio signal, which is recorded by a microphone. Through detecting the zero-crossing-rate of this sinus the pitch of the pipe can be calculated directly, using an Arduino platform for this purpose in prototype stage.

C. Software

For calculating the required movements of the motors from the frequency measurement, appropriate software was developed in C#. For controlling the tuning system in the prototyping phase a graphical user interface (GUI) was designed. The tuning device and the electronics (described in the following section) are connected via USB to the computer, on which the software is executed. On the GUI the current divergence to nominal frequency is charted in real-time. The motors are not driven continuously, but stepwise. The length of the switched-on pulses depends on the divergence to nominal frequency of the pipe, followed by a stop until the next pulse length is calculated. This stepwise mode is needed because of the very high sensibility of the reed. At the smallest pipes a one micrometer motion of the tuning spring results in 0.5 cents deviation of pitch.

D. Electronics

To transform the calculated pulses from the software into voltage for the motors, drive electronics and an appropriate logic unit are needed. Therefore, an Arduino board with three motor shields (extension boards) was used. Each board can drive two motors, so six pipes can be connected simultaneously for prototyping. Furthermore, the motor shields support motor current measurement, so it can be detected without additional sensors, if the motor is stalling, e.g. if the tuning spring has reached its end position.

VI. RESULTS

After finishing the constructing phase, the prototyping setup was tested extensively. An endurance test was performed with one pipe to verify fatigue strength of the system. Thereby the motor moved the tuning spring for about 20 hours continuously (3715 tuning cycles), until one gear wheel was abraded. This number of tuning cycles would never be reached in a real organ, so the drive is applicable from this point of view.

The precision and the speed of the automated tuning process meet the requirements set for this project. A pipe can be tuned in less than ten seconds with satisfying precision ($\pm$0.5 cents), whereby the manual process takes about 30 seconds for each pipe. The system can perform tuning even more accurately, whereby the tuning time increases.

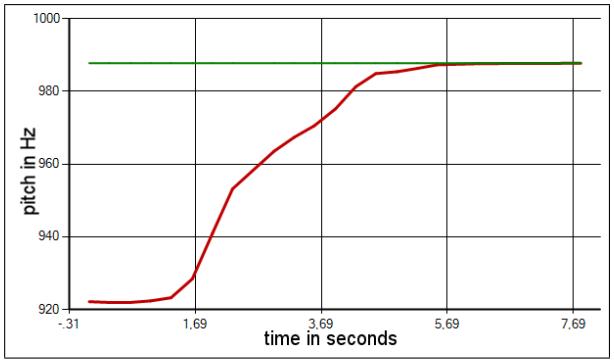


Fig. 9. Resulting tuning process of reed pipe (green...nominal value, red...actual value)

VII. CONCLUSION AND OUTLOOK

Overall, the main aim of this project, to evaluate the possibility of automated reed pipe tuning, was reached at an early stage and extensive additional developmental work was done. Because of the low price and the small size of the implemented actuator the results actually exceeded the author's own expectations by far. In future work the software should be transformed from PC to an embedded system and should be integrated into the real organ control system. Thereby the organ could be programmed to tune itself at specific dates or tuned by starting the process from a smartphone from anywhere. Using more than one bandpass filter would enable tuning several pipes simultaneously. That would be a significant advantage over to manual pipe tuning.

ACKNOWLEDGMENT

Firstly, I would like to express my sincere gratitude to my advisor Dr. Markus Trenker for the continuous support of my Master Thesis, for his patience, motivation, and immense knowledge. His experience helped me in all the time of writing this thesis.

My sincere thanks also goes to Wendelin Eberle, the CEO of Rieger Orgelbau GmbH, who provided me an opportunity to join his team for this project, and who gave access to the companies knowledge and provided specific tools and organ parts. Without this precious support it would not be possible to conduct this research.

REFERENCES

[1] Rieger Orgelbau GmbH. [Online]. Available: http://www.rieger-orgelbau.com/
[2] T. Bothe and J. Kablitz, "Selbststimmende Orgelpfeife," Kiel: FH Kiel, 2014.
[3] Elliptec GmbH, "Elliptec Motor X15G," Dortmund, 2016. [Online]. Available: http://www.elliptec.com/de/produkte/motor-x15g/
[4] Physik Instrumente (PI) GmbH, "Piezomike linearaktoren," 2016. [Online]. Available: http://www.physikinstrumente.de/technologie/piezomike-linearmotoren.html
[5] D. Voigt and M. Voigt, "Pfeifenorgel mit selbstregulierender Stimmung," German Patent DE102 011 013 444, 2012.
[6] M. Voigt, "Stimmungseinrichtung für gedackte Orgelpfeifen," German Patent DE102 013 012 821, 2015.
[7] M. Voigt, "Stimmungseinrichtung für Orgelpfeifen," German Patent DE102 012 021 644, 2014.
[8] *Tuning Set CTS-32-C*, manual, www.tuning-set.de, 2008.

RobWood - Smart Robotics for Wood Industry

Thomas Haspl[1], Claudio Capovilla[1], Alfred Rinnhofer[1], Victor J. Exposito Jimenez[1], Stefan Maier[2], Alexander Heinisch[2], Matthias Völkl[3], Manfred Zarnhofer[4], Robert A. Jöbstl[5], Erhard Pretterhofer[6], Bernhard Dieber[1] and Herwig Zeiner[1]

***Abstract*— Many branches of the manufacturing industry in general, and smaller wood processing companies in particular, are facing challenges related to producing ever smaller lot sizes under increasing time pressure. The *RobWood* project aims to increase the flexibility of such companies by providing a tool-chain to easily program robots for wood processing. In this paper we present an overview on our approach to robot programming by using models of the finished product.**

I. INTRODUCTION

Austrias wood processing industry accounts for 10 billion Euro, and ranks with a 3.9% trade balance surplus on second place just behind the tourism industry (4.2%). Each year, about 18,000 building construction permissions are issued, where prefabricated houses have a share of approx. 30-35%, with an upward tendency over the last years [2]. Ever higher demands regarding quality standards and individuality pose serious challenges to companies in the wood processing industry.

The goal of the *RobWood* project is to enable strong individualization of products at an equal or higher level of production efficiency through new technological approaches. The integration of robotics, sensor technology, and knowledge transfer with appropriate human-computer interfaces, applied in production, helps to optimize operating procedures in the wood industry. The use cases on which we work on in this project come specifically from the manufacturing of wooden prefabricated houses. Here, every house can be individualized but the parts are prefabricated in a factory instead of building them on site. In order to do this, many different steps have to be performed at each part like cutting, milling and clamping and the joining of different parts like steam brakes with the wooden elements.

Model-based programming is a powerful concept, which can lead to more natural interaction and easier programming of industry robots. Employees in smaller wood processing companies without in-depth knowledge regarding traditional robot programming will so be able to program robots themselves.

*The work reported in this article has been supported by the Austrian Research Promotion Agency under grant nr. 849896

[1] JOANNEUM RESEARCH
{firstname.lastname}@joanneum.at
[2] RIB-SAA
[3] ABB AG
[4] Zarnhofer Holzbau GmbH
[5] Haas Fertigbau Holzbauwerk GmbH & CoKG
[6] Holzcluster Steiermark GmbH

Research into intelligent technologies for accessing the data and knowledge created thereby has a strong leverage effect on its usage, already within single production enterprises and additionally across company boundaries.

The robot based production optimization pursued by the project has enormous potential regarding the creation of new jobs also in more rural areas, the efficient use of resources, and the transfer of insights to other sectors.

The rest of this paper is structured as follows: In section II we present related work, in section III we describe the challenges of automated wood processing, in sections IV and V we present the concept and tool chain of our solution and finally we conclude in section VI.

II. RELATED WORK

The trend towards computer based planning and processing methods has been finding its way into the wood manufacturing industry since a few years. In some sectors of the manufacturing industry, automated *CAD/CAM (computer aided design/manufacturing)* systems that generate machining data for use in lot size one manufacturing out of geometrical *CAD* data already exist - such as in the prefabricated concrete parts industry, and of course metal cutting on CNC machines as well as additive printing for prototype construction.

A. Model-based industrial Robot Programming

New research work is investigating new programming methods for making complex tasks easier to program for standard industrial robots [9]. Common approaches include offline programming methods with a complete 3D model [7]. The second common procedure called *teach-in*, or online programming, is very time consuming for complex task processing. Other approaches such as intuitive robot programming for *SMEs (small and medium-sized enterprises)* are described e.g. in [3], [13]. This approach is based on new types of e.g. gesture-based definition of poses, trajectories, and tasks. It is based on a visual programming concept that allows non-skilled programmer operators to create programs. For complex manipulation processes with a huge amount of *CAD* models, this approach does not significantly reduce the effort for the programming task.

B. Model-based Approaches

In short, *model-driven engineering (MDE)* [12], [14] is summarized as follows: model once, generate anywhere. This principle is particularly relevant when it comes to the building of robot applications. The modeling is done on different

abstraction levels and the (mostly) automated translation of models to machine readable codes increases efficiency for creation and maintenance of applications. Combined with the improved quality of implementation and reduced fault susceptibility, flexibility in production can be increased significantly. Another approach is the use of *domain specific languages (DSL)* [4]. These have been drawing attention especially in the area of service robotics during the last few years.

C. Complex Wood Manufacturing Processes for Robots

Typically a user of such a robot based environment has to perform different subtasks along a given manufacturing wood processing chain [6]. A technician may look at technical features, the customer indeed is mainly interested in how satisfactory they are solved. Automatically the user is scoring his satisfaction within the process execution, either in terms of time to perform, the process stability or comparing outcome quantity, efficiency and limitations. Complexity may be defined as the necessity to involve more than pure kinematic robotic control to perform a task, therefore going beyond the well-known operations.

III. PROBLEM STATEMENT

The main focus of innovative and new type of model based robot programming for wood manufacturing industries lies on the ease of use for non-experts in robotics coming from this specific domain. No knowledge about traditional off-line-programming or specifics of robot programming should be required. This requires the selection and implementation of an applicable method for creating the necessary data about manufacturing steps for the machining of solid wood elements with an articulated robot using different kinds of tools. As human labour is an integral part of manufacturing however, to make such a system available on the market today, interactive methods for the collaboration between human workers and a robotic system have to be examined and established.

Production steps where human interaction with the work piece is necessary should be kept at a minimum, where the machine operator can decide between using human labour where it might save time or material, while keeping the robot as fully engaged as possible. To ensure this, the commands for the robots need to be generated directly from the *CAD* plans, which have been enhanced with semantic information. Interfaces between *CAD* systems and robot installations need to use standardized formats as often as possible. In *CAD* there are various specific formats available, which need to be analysed, evaluated and may be adapted or enhanced for the intended use. The evaluation of existing open standards is also an important task in the *RobWood* project. However, future industrial use and uptake of the proposed technologies are subject to support by the *CAD* system providers. There are standardized interfaces in existence for use in *CNC (computerized numerical control)* production environments (DIN 66025/ISO 6983). These are mainly used for portal systems and toolsets and might need adaptation for usage with buckling arm robots.

Starting from a desired pose of an robots attached tool, one has to solve the so called inverse kinematics problem for a robot to obtain the corresponding robot axes configuration [5], [10]. This problem is difficult to solve mathematically, and typically has several solutions as shown in Figure 1 for an elbow up and elbow down configuration. A model-based approach, however, would offer this functionality for a broader spectrum of robot types and in particular, as accessible component earlier within the model-based software tool chain.

For the intended use case the number of produced unique pieces is one, although the reuse of parts of the models has to be considered. This also affects the specifications for creating enhanced user- and programming interfaces. Apart from specific variables such as technical interfaces, drivers, catalogues and configurations, the resulting user interface should strive to be as independent from the robot system as possible. System configuration should be kept to a minimum and is done with simple configuration procedures and minor manual settings. The process should not require an expert in robot programming or setup.

Special care is to be taken to ensure that the person resetting the system is not able to bypass security measures built into the system and that access to the robot for configuration is only available during idle times. Especially the manual steps required during a reset are to be built with simple visually enhanced instructions so that the wood manufacturing personnel can safely perform the necessary procedures without the help of experts in robotics.

The particular requirements of the wood manufacturing industry imply the necessity of a tool catalogue, which holds all required and possible tools as well as their corresponding procedure parameters such as speed of operation and logistics of operation. These catalogues can differ between system configurations, for example for the same procedure but requiring different tools. Bringing the various configurations into a form that is both readily comprehensible as well as comprehensive will be one of the challenges of the proposed project. To provide this system also for timber frame construction the various steps related to treatment, positioning and assembly need to be considered as part of the overall procedure, even though these tasks are not be fulfilled by the same robot system but with an assisting system, yet at the same time keeping the transparency for the user.

Special focus is granted to the usage of the envisioned systems in time sharing and collaborative environments, where different companies share one robot system or a specialised provider offers the robot system as a service. This increases the importance of using standardized interfaces to offer the simple exchange of treatment models and object data while at the same time ensuring *IPR (intellectual property)*. The vision for the final system is to integrate the whole class of production control systems and production planning systems. As various robot systems could be part of the same production not only the machine-to-human but also

the machine-to-machine communication has to be taken into account.

IV. THE ROBWOOD CONCEPT

The main task of the *RobWood* project is the development of several tools, which allow to combine each other in a very flexible and fast modifiable way.

A. A Model driven Approach

First of all, an overall controlling software, which is able to process and forward *CAD*-data, is needed. Within the *RobWood* project this software implementation has been named *Manufacturing Execution System (MES)*. A data sink from this *MES* has to be a robot cell controller, which takes care of all the robot related information such as the actual position or the life cycle status of particular tools. The cell controller should actually serve as abstraction layer between the *MES* and the robot platform. As many different robots can be applied, every robot would need a customized cell controller in order to meet the interface requirements of the *MES*. Additionally, a *Quality Inspection Unit (QIU)* has been developed. This is a vision based unit and is responsible for controlling and ensuring the lasting quality of the workpieces. In order to achieve a high grade of flexibility and reusability of data, a cloud service for exchanging *CAD*-data with other companies and users is implemented. This unit is called *Cloud Exchange Service (CES)*.

B. The Production Workflow

The combination and way of interaction of all parts involved in the *RobWood* project is visualized in figure 1, where all the gray shaded blocks indicate tools, which have been developed within the project.

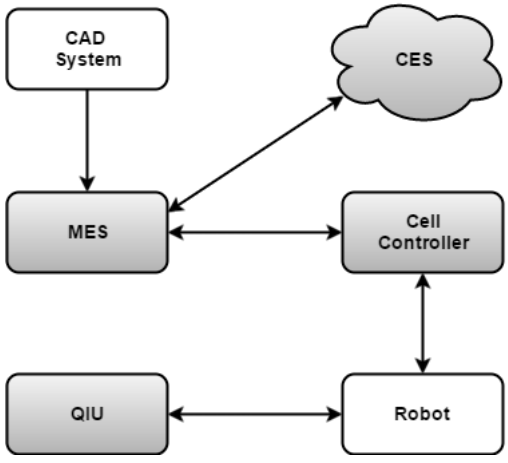


Fig. 1. Relations between the participating Units.

At the beginning, the designer designs, layouts and analyzes the house. During his/her work in the *CAD*-system this design is broken down into automatically producible elements. After handing over the elements to the *MES*, they are nested with other elements if possible and handed over to the production process. The production process is a quite broad term, starting from laser applications to plot the elements, augmented reality applications, printing plans or barcodes over to the actual manufacturing of the product. In this phase it is possible to interact with other plants or the cloud services to optimize production (e.g. by producing similar elements with specific characteristic in a plant dedicated to these). When the elements are taken over to the production process they are forwarded to certain cell controllers specified to interact with an unique robot or machine. These cell controllers are placed on site and give detailed insights in the actual work in progress of the robot. The main purpose of these controllers is to abstract the robots interface to the *MES* and give a more detailed insight for the user. The robot itself controls mechanical units needed to manufacture the product or to grant safety to the users.

C. Domain Specific Language

A *Domain Specific Language (DSL)* is a useful concept, which basically depicts a programming language that is created for a specific purpose. In other words, a *DSL* is a tool with limited focus, which we found to be an ideal opportunity for the *RobWood* project. Domain specific modeling is often used to describe concerns in robotics with concepts and notations to get closer to the respective problem domain and to raise the level of abstraction [8].

For the project we chose to implement a textual *DSL* instead of a graphical one as many frameworks are only available for the Java language runtime stack. A textual *DSL* also provides better integration to apply it in the future system which is based on the *.NET* language stack.

The most relevant issue in our selection process is the integration of the *DSL* in the *.NET* language stack. Another point that we have considered is the integration in a future platform. For these reasons we have decided to choose an internal *DSL* because we are able to implement our approach in a more affordable way. This kind of language fits very well for reactive systems, which are the systems that respond to external events, similar to robots. For all these aspects, the *F#* programming language was chosen to build our domain language.

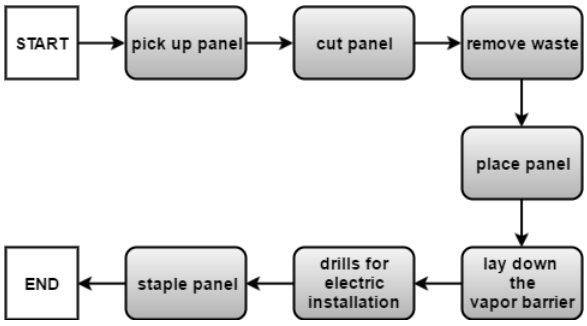


Fig. 2. Diagram of the Production.

In figure 2 the whole production process is visualized as a sequence of several tasks described by the *DSL*. These particular tasks can again be described as a composition of more fine-grained tasks. Such an example of the *pick up*

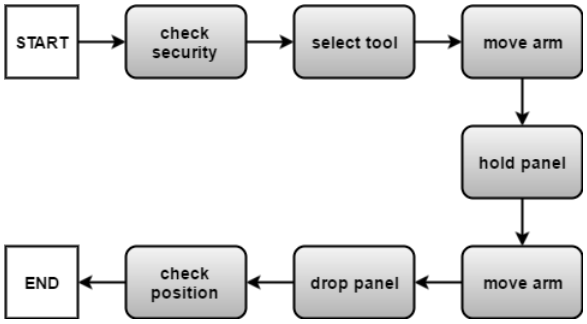


Fig. 3. Diagram of the *"pick up panel"* process as stated in figure 2.

panel task is shown in figure 3. Our *DSL* has three main commands to build each process. A ***process*** is used when a process starts. The name of the process is set as an attribute in the syntax. A ***task*** describes an automatized operation. The ***use*** command is used when the process is supposed to use a specific tool. Parameters can also be used to configure the behavior of the tool. So, the *pick up panel* process as in figure 3 can with the help of the *DSL* finally be written as follows.

```
process "pickup_panel"
task "check security"
use "arm" parametersv "5.5,6.0,8.0"
use "vacuum_griper" parameters "hold"
use "arm" parameters "3.5,3.0,8.0"
use "vacuum_griper" parameters "drop"
use "laser" parameters "0.0,0.0,0.0,1.5,1.5,1.5,1.5,0.0"
```

V. TOOL CHAIN

A. Manufacturing Execution System

The *Manufacturing Execution System (MES)* is responsible for preparing the *CAD* data for production and for performing and tracking the transformation of raw materials into finished goods.

As depicted in figure 4 the product is handed over by the *CAD* system. First, the *MES* checks, if the delivered data is correct regarding syntactic and semantic concerns such as closed contours or the considerations of the maximum producible dimensions. If these checks fail and an automated manufacturing process cannot be guaranteed, a meaningful report will be presented to the user, so that he can take further corrections.

After validating the *CAD* data, the system forwards the data to the *CAD* reader, which transforms it into an internal representation. Therefore, the beams of timber and panels of timber or gypsum are merged into elements according to their identifiers. Then, the surrounding contour of the elements is calculated. After that the beams of timber can be stored in the database. Since the beams are not part of the automatic manufacturing process, this information is only used for displaying them within the production units plan as well as to calculate the positions where the gypsum and timber panels have to be connected to the beams. The next step is to store the panels and the position (layer) of the steam brake in the database. Finally, all mounting parts for the particular processing steps are determined and written to the *CAD* file. This parts could be e.g. drillings for power outlets or heating systems.

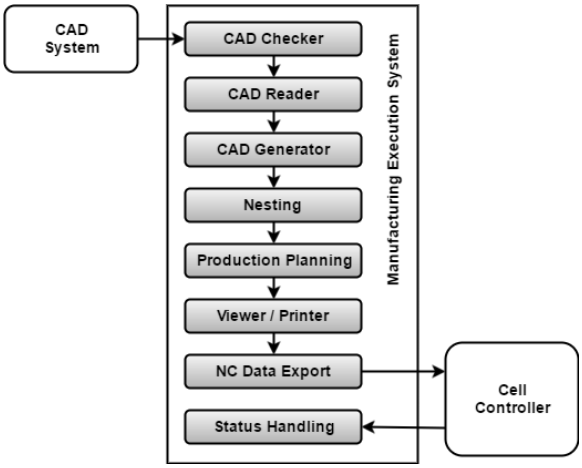


Fig. 4. Scheme of the Manufacturing Execution System.

The next unit within the *MES* is the *CAD* generator. In this step the positions for clamping are generated. Therefore, positions alongside the beams are calculated within a given distance. The dimensions of the steam brake is also calculated in this step. Since the steam brake is considered to be cut orthogonally only, and to be at least as broad as the element, this step can be reduced to calculating the length according to the elements contours surrounding rectangle plus some extra overhang.

The fourth part of the *MES* is called *Nesting* and enables the user to produce multiple elements to be processed on a single carrier. Elements are placed next to each other within certain constraints, which include dimensional restrictions, maximum number of mount parts per carrier or the minimum distance between elements. The latter is important to prevent an overlapping of the steam brake.

After the elements are nested on carriers the whole production process can be observed and planned in the production unit list. This list shows the current state of the production unit and it is also possible to view or print the plans of these production units, showing all the details of the elements contours according to their position on the carrier. Similar to this plans the carrier can be augmented by further information from a laser system.

The last unit of the *MES*, before the product is handed over to the production, is the *NC Data Export*. When the cell controller requests a new production unit, the *MES* prepares the next production unit in the production unit list to be produced.

Throughout the whole manufacturing process, the production unit can be tracked and its current status can be observed. Further, it is possible to analyze production, failures and throughput through a reporting service provided.

B. Cloud Exchange Service

The *Cloud Exchange Service (CES)* offers the opportunity for several companies to exchange different *CAD* modules,

the corresponding metadata and tool catalogues. The purpose of the *CES* is to lower the integration barrier of cloud services for *SMEs*. Its key functions are the management of *CAD* data. The *CAD*-files are of particular interest within *RobWood*, but also within non-spatial information such as processing sequences, which should be made more accessible and easily available through the cloud-exchange services. Cloud services may also be used to evaluate this data.

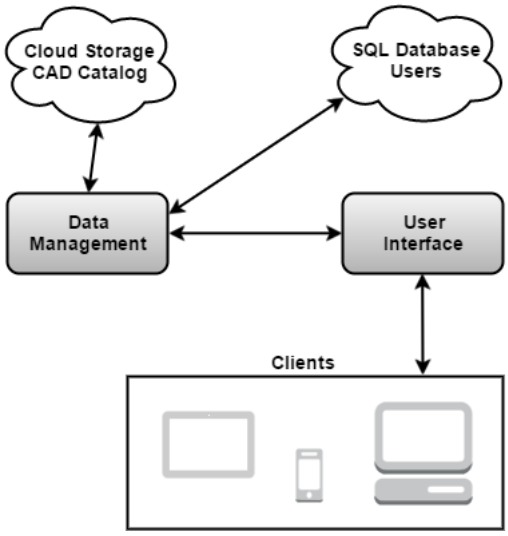


Fig. 5. Scheme of the Cloud Exchange Service Unit.

A general structure of the system is shown in figure 5. The system is developed on the basis of microservices architecture explained in [15], which provides more flexibility, resilience and scalability than a monolithic architecture. These microservices are small services that are in charge of small tasks. So, it is possible to build the whole system by the combination of them. The microservices are independent and connected through a *RESTful API* [11]. If needed, there can be added more microservices with few changes in the system architecture.

C. Cell Controller

The cell controllers main purpose is to request new data from the *MES*, optimize it for the specific robot platform and forward the prepared data to the robot. Besides that, it also serves as a human device interface, visualizing the status of the robot on a detailed level, giving the user more insight into the tasks performed by the robot. The information received from the *MES* contains all necessary data to automatically produce the elements on the carrier which is currently at the robots station. This includes all panels to be cut and placed, the steam brake and the drillings and clamping positions. Depending on the layer of the data blocks, their complexity and their position (e.g. path planning) a rearrangement of those can be done by the cell controller to increase cycle time and decrease waste. Each of these data blocks represent a single task to be performed at the robot.

Furthermore, the cell controller monitors the stock of raw materials and alerts the user if the cell runs out of timber or gypsum panels. Errors detected during production and the progress of the production unit are reported to the *MES* as well, depending on their severity and the kind of repair actions to be taken. Finally, after all tasks have been committed by the robot, the cell controller notifies the *MES* and requests the next production unit.

D. Robot

The robot is the last instance in the production line of the manufacturing process. It consists of a control unit able to receive new tasks from the cell controller. Each task consists of a single work package, e.g. contour and position of a panel. This package is handled by the robots control unit and split into single mechanical movements. The most important abilities of the robot are cutting, positioning, clamping and stapling the panels as well as placing the steam brake and drilling holes (e.g. for power outlets).

After the robot gets the task assigned to place a panel, it has to lift the panels out of the store. While doing so, it has to check if enough panels are on stock for future proceedings. If this is not the case it has to indicate the cell controller to notify the user. Since the refilling of the panel storage is not automated, the user has to fill these by hand. After lifting the panels, the robot places them on a dedicated work bench, the *carrier*, where the cutting is performed. The cutting is executed according to the contour information handed over by the cell controller. Depending on the material to be cut (e.g. timber, gypsum) the robot will change its tools automatically. Since the work bench is sloped down, the waste and dust emerging during cutting slips off the panel and has not to be removed explicitly. This allows that further layers can be positioned without an additional cleaning step. When all parts of a layer are positioned, the robot gets the instructions to nail the panels. Again, the tools are changed and staples or nails are driven through the panels into the beams to tighten them. Depending on the layering of the element the steam brake is applied after this step. To do so, the roll containing the steam brake is released and the robot pulls off the steam brake from the roll. After reaching the desired length, an orthogonal cut is made and the roll is locked again. After applying the steam brake, additional layers of panels can follow where the previous described steps have to be repeated. When all layers of the element are produced, the robot starts to drill holes through the panels and the steam brake before the production unit is released and handed over to the next production station of the plant.

E. Quality Inspection Unit

The *Quality Inspection Unit (QIU)* is used for supervising the clamping process required to combine different wooden elements. It consists of a number of components, which are shown in figure 6. Again, the units main parts are marked in gray.

A visual inspection sensor is mounted in or on the physical set-up in the production cell or even on the robot arm itself. This sensor data is processed by the *Sensor Control and*

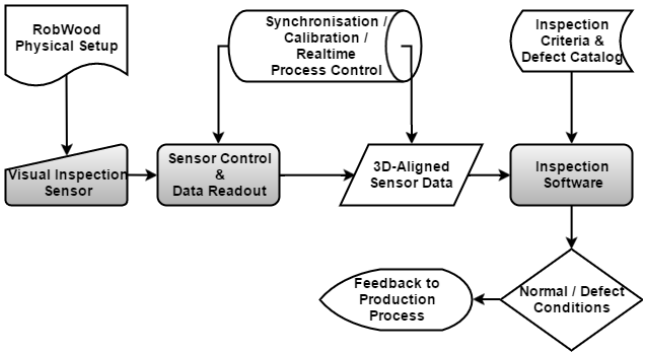


Fig. 6. Scheme of the Quality Inspection Unit

Data Readout unit that takes synchronization and calibration information from the control system to generate inspection sensor data aligned in 3D to the specimen to be produced.

The main part of the *QIU* is the *Inspection Software* that is fed with inspection criteria and quality thresholds and a defect catalog. The software decides upon defects and failures and reports those back to the production process. For fulfilling all these tasks the *QIU* provides a set of functionalities.

First of all, it captures the clamps once clamped, using a dedicated visual 2D/3D sensor. Afterwards, it generates a 3D representation, a so called *Digital Terrain Model (DTM)*, of the specimen surface. This *DTM* is then be analyzed with respect to segments that significantly exceed the ordinary surface plane of the specimen. Optionally a-priori information about the position of applied clamps can be used, to minimize the search space. In any case, the segments that indicate defects, are detected in this step. For such segments the responsible *MES* is beeing notified about the erroneous clamp, its position and optionally the amount and/or type of defect.

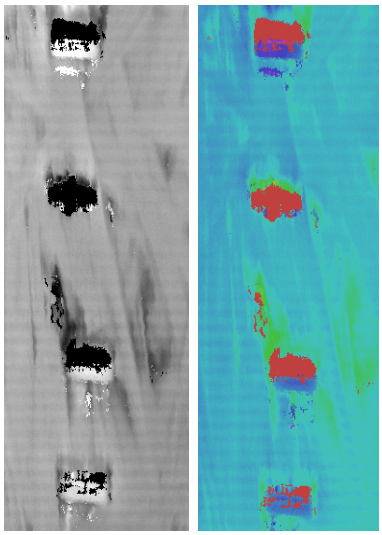


Fig. 7. Sample Images of not fully applied Clamps.

In figure 7 a sample of not fully applied clamps as detected by the *QIU* is shown. In the color coded image (right) the red segments represent undefined areas, which can be caused by an occlusion from a staple fully applied to generate a small ditch, or from an occlusion caused by a staple not fully applied. Dark blue areas are portions with higher elevation, hence not fully applied staples being detected as production errors . Light green means measured ditches. The measurement direction is from above.

VI. CONCLUSIONS

This paper contains an outline of the *RobWood* approach. We have described how we could design components which could be used for the company to program the robot with less effort. This is an important issue for workers without profound programming skills. A first, a series of tests of the particular components of the tool chain took place in a gradual manner at the *Holzinnovationszentrum* [1]. At the end, an integration test with all the components passed successfully. To sum up, we believe that our approach can be integrated in the production for the wood industry in the next three to five years.

REFERENCES

[1] Holzinnovationszentrum gmbh. [Online]. Available: http://www.hiz.at
[2] Proholz austria. [Online]. Available: http://www.proholz.at
[3] T. Dietz, U. Schneider, M. Barho, S. Oberer-Treitz, M. Drust, R. Hollmann, and M. Haegele, "Programming system for efficient use of industrial robots for deburring in sme environments," in *ROBOTIK 2012; 7th German Conference on Robotics*, May 2012, pp. 1–6.
[4] M. Fowler, *Domain-specific languages*. Pearson Education, 2010.
[5] A. Goldenberg, B. Benhabib, and R. Fenton, "A complete generalized solution to the inverse kinematics of robots," *IEEE Journal on Robotics and Automation*, vol. 1, no. 1, pp. 14–20, 1985.
[6] J. O. Huckaby and H. I. Christensen, "A taxonomic framework for task modeling and knowledge transfer in manufacturing robotics," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
[7] P. Neto, J. N. Pires, and A. P. Moreira, "Cad-based off-line robot programming," in *2010 IEEE Conference on Robotics, Automation and Mechatronics*, June 2010, pp. 516–521.
[8] A. Nordmann, N. Hochgeschwender, D. L. Wigand, and S. Wrede, "A survey on domain-specific modeling and languages in robotics," *Journal of Software Engineering in Robotics*, vol. 7, no. 1, 2016.
[9] Z. Pan, J. Polden, N. Larkin, S. V. Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87 – 94, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736584511001001
[10] R. P. Paul and B. Shimano, "Kinematic control equations for simple manipulators," in *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, Jan 1978, pp. 1398–1406.
[11] L. Richardson and S. Ruby, *RESTful web services*. " O'Reilly Media, Inc.", 2008.
[12] C. Schlegel, T. Hassler, A. Lotz, and A. Steck, "Robotic software systems: From code-driven to model-driven designs," in *2009 International Conference on Advanced Robotics*, June 2009, pp. 1–8.
[13] M. Spangenberg and D. Henrich, "Towards an intuitive interface for instructing robots handling tasks based on verbalized physical effects," in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Aug 2014, pp. 79–84.
[14] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, and A. Wortmann, "A new skill based robot programming language using uml/p statecharts," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 461–466.
[15] H. Zeiner, M. Goller, V. J. Expósito Jiménez, F. Salmhofer, and W. Haas, "Secos: Web of things platform based on a microservices architecture and support of time-awareness," *e & i Elektrotechnik und Informationstechnik*, vol. 133, no. 3, pp. 158–162, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00502-016-0404-z

Task-Dependent Configuration of Robotics Systems

Alexander Pagonis[1] and Clemens Mühlbacher[1] and Gerald Steinbauer[1] and Stefan Gspandl[2] and Micheal Reip[2]

***Abstract*— To solve a task, a robotics system uses several different hardware and software components. Each of these components solves a specific subtask to allow the overall task to be solved. Thus, the proper selection of the set of components is crucial for the success of performing a task. This selection can become complex if one needs to consider that each of these components has its own dependencies which need to be fulfilled to work properly. Due to this complexity, the proper selection of components is time-consuming and error prone. Additionally, domain knowledge is necessary to consider all dependencies correctly.**

To proper choose the components without the need of a domain expert one can follow a model based approach. In this paper, we show how such a model-based approach can be used. We present a tool that, based on a domain model, automatizes the selection of the necessary components to implement a set of given tasks. Due to this automatic selection mechanism, one can either simply check if a robotic system can perform a task or which components need to be added to allow the robot to perform the given task.

I. Introduction

A robotics system consists of several hardware and software components which interact with each other to achieve a given task. The selection of the hardware and software components is often done by a domain expert, ensuring that the task can be fulfilled with the given selection. This is a time-consuming task, as one needs to know which dependency each component has, e.g. a computer vision algorithm depends on a camera but does not specify which camera exactly. Additionally, one possible needs to consider many possibilities how a dependency can be met to find an optimal selection. Following simple scenario is used to highlight these difficulties: The task the robot must fulfill is to localize itself. One could now use a localization which is based on a laser or a localization which is based on the camera. In case there is a Kinect camera [1] available but no laser, a camera-based localization approach would probably be preferred. But one could use the depth image to simulate a laser scanner and thus use also the localization based on a laser scanner. This simple example already shows that one needs to consider several possibilities and necessary dependencies to allow a robot to solve a task.

[1]Alexander Pagonis, Clemens Mühlbacher and Gerald Steinbauer are with the Institute for Software Technology, Graz University of Technology, Graz, Austria. {apagonis,cmuehlba,steinbauer}@ist.tugraz.at
This work is partly supported by the Austrian Research Promotion Agency (FFG) under grant 843468.
[2]Stephan Gspandl and Michael Reip are with incubedIT, Hart bei Graz, Austria. {gspandl,reip}@incubedit.com

Instead of choosing the hardware and software components manually, one can follow a model-based approach for the robotic system as it was outlined in [2]. The idea is to use a model that describes the task as well as the available hardware and software components, their capabilities and dependencies. By using this model one can automatically generate a list of components that are necessary to fulfill a task. The model does not only allow to generate a list of components to fulfill a task but it also allows the robot to check if a task can be executed with the given hardware and software. Furthermore, the robot can use the model to decide which alternative software and hardware modules to use if one part of the system does not work correctly. Such a reconfiguration is of special interest if one considers complex tasks which can be achieved through several means.

In this paper, we present a tool which allows performing such a model-based configuration of a robotic system automatically. The tool can be used to derive which set of components needs to be present to allow fulfilling a task. Furthermore, the tool allows checking if a given robotic configuration can fulfill a task. Additionally, all possible component compositions that allow solving the given task can be viewed. This allows checking which alternatives are possible and which components are redundant in the system. To allow an easy configuration the tool does not only suggests possible configurations but also allows to interactively vary the given configuration. This makes the configuration process easy and allows for a quick decision on the best fitting set of components.

The remainder of the paper is organized as follows. In the next section, we discuss the design of the configuration tool. This description comprises the used knowledge base, the method which is used to derive a correct configuration, and the description of the user interface. The proceeding section discusses a simple example scenario and presents how the tool can be used. This is followed by a section discussing the limitations of the approach. Afterward, we will discuss some related research. Finally, we conclude the paper and point out some future work.

II. The Configuration Tool

As we motivate above using a model one can automate the generation of a configuration for a given task. This generation uses the model to determine the dependencies between software component and hardware component. Furthermore, the model describes the different possibilities to resolve a dependency. To ensure that the model can answer a query in a timely manner and to allow still the model to be expressive

we use an Ontology for the model. With the help of the model, the tool can derive the dependencies which need to be met to fulfill a task.

To derive which configuration fulfills the dependencies a separate reasoning process is performed. This separate reasoning process uses the data contained in the model to yield a minimal configuration. Through this separation, the model can be capped simply by avoiding the "complex" reasoning for a minimal configuration.

Using the information from the ontology and the reasoning to derive a minimal configuration the tool can present a possible configuration to the user through a graphical user interface. The interface allows selecting tasks to perform, which components are used as well as which configuration would be minimal. In the remainder of the section, we will discuss each part in more detail.

A. Ontology

To model the relationships between tasks and necessary components, an ontology describing this relationship is needed. The ontology we use for the implementation is an open-source knowledge base and can be found at [3]. In this ontology, tasks are referred as capabilities. Each capability can be comprised of other basic capabilities. The ontology also describes the relationship of capabilities to hard- and software components that are needed for their implementation. Some of these components may be compulsory and do not include alternatives while others may be chosen from a pool of similar components that may all be used to fulfill the same task.

The information, stored in the ontology can be loaded and queried with an appropriate tool. We use the framework Jena [4] to load the ontology into a model. The model can be queried using the SPARQL query language. The Jena framework allows multiple ontologies to be loaded into a single model. The base ontology we use already contains references to the sub-ontologies, including descriptions of robot components. Therefore, it is enough to load the base ontology as all sub-ontologies will be loaded into the model automatically by the framework.

B. Calculation of Configuration

With the help of the ontology mentioned above, we can define the dependencies which need to be met to perform a task. The above calculate gives as a set of capabilities Cap, which can be requested to be fulfilled directly or indirectly. We use the variables X and Y in the remaining subsections for variables with the domain of capabilities $dom(X) = dom(Y) = Cap$. Besides the capabilities, we have additionally the set of components $Comp$. These components describe a software component, e.g. a laser-based localization or a hardware component, e.g. a laser scanner. We use the variable Z in the remainder of the subsection for variables with the domain of components $dom(Z) = Comp$. As the description of the components is rather abstract one needs a concrete implementation/realization of such a component, e.g. a Sick LMS100 for a laser scanner. To describe this implementation/realization of components the ontology above yields the set $ImplComp$. In the remainder of the subsection, we use the variable W for variables with the domain of the implementations of components $dom(W) = ImplComp$.

Beside the sets of possible capabilities, components and their implementation we additionally have four different functions describing the dependencies which need to be fulfilled for a capability, component and its implementation. The function $capReqCap : Cap \rightarrow 2^{Cap}$ describes which set of capabilities needs to be fulfilled by the robot to implement a certain capability. For example, the capability $liftObject$ depends on two other capabilities $moveArm, graspObject$ which is described as follows $capReqCap(liftObject) \rightarrow \{moveArm, graspObject\}$. To describe the dependencies between capabilities and components the function $capReqComp : Cap \rightarrow 2^{Comp}$ is used. For example, the capability $liftObject$ depends on two components $arm, gripper$ which is described as follows $capReqComp(liftObject) \rightarrow \{arm, gripper\}$. Each requested component can be implemented differently to link a component and an implementation we use the predicate $implComp : Comp \times ImplComp \rightarrow \{\top, \bot\}$. Like a capability a component can depend on capabilities, we use the function $compReqCap : Comp \rightarrow 2^{Cap}$ to describe this dependency. Additionally, a component can depend on other components which define through the following function $compReqComp : Comp \rightarrow 2^{Comp}$. Using this functions and the predicate we can define the dependencies which need to be met to implement a certain capability.

As we are interested in a configuration of the system which is minimal we need a specific reasoning to derive such a configuration. This is done by first extracting all dependencies of a task together with every possibility to meet this dependency. The model does not store all dependencies in a single level. Instead, some dependencies may result from other dependencies. Therefore, a recursive extraction of dependencies must be performed.

Once all these dependencies are extracted, a constraint problem can be defined to find (all) minimal configurations which fulfill the dependencies. This is done as follows. For each capability Y which is required the predicate $reqCap(Y)$ is used to describe the capabilities and components which are needed by the robot.

$$reqCap(Y) \rightarrow \bigwedge_{X \in capReqCap(Y)} reqCap(X) \wedge \bigwedge_{Z \in capReqComp(Y)} reqComp(Z)$$

Through this equation, one can simply resolve the recursive dependencies on the capabilities. Some of these capabilities might need components. As several hard- or software instances can implement a specific component we use an equation for the required capabilities to resolve components. If a component W implements a required component Z we

define the following constraint.

$$reqComp(Z) \rightarrow implComp(Z, W) \wedge comp(W)$$

Like capabilities, components can have dependencies. Thus, to model these dependencies we use another constraint.

$$comp(W) \rightarrow \bigwedge_{X \in compReqCap(W)} reqCap(X) \wedge \bigwedge_{z \in compReqComp(W)} reqComp(z)$$

Using the ontology, we can instantiate the constraints automatically. The instantiated constraints are gathered in the set $\mathcal{C}$. As we want to derive a minimal configuration for a given task X we first need to add $reqCap(X)$ to the set $\mathcal{C}$. Afterward, we need to find a minimal set of components $\mathcal{W}$ to fulfill this requirement. This is achieved by the following minimization problem.

$$\underset{\mathcal{W}}{\operatorname{argmin}} \left(|\{W | comp(W) \wedge W \in \mathcal{W}\}|\right) s.t. \mathcal{C}$$

The solution to this minimization problem is a minimal set of components to use to guaranty that all dependencies are met.

With the above-defined constraint problem, the minimal configuration can be generated. To realize these constraints in an efficient manner the constraint solving is split into two parts. The first part is the parsing of the ontology to extract the minimal dependency for one component. The second part uses this extracted data to find a minimal configuration in a very efficient way through a constraint solver.

The first part is done by extracting the minimal set of necessary capabilities, for a chosen task, by recursively traversing the referenced capabilities of the chosen tasks. Using the resulting set of capabilities all mandatory components are extracted. Additionally, during the recursion one creates a separate set of mandatory components for each alternative realization. After this extraction, the minimal set of necessary capabilities, as well as the mandatory components, are already determined. Therefore, only the extraction of the various combinations of alternative components, such that the required tasks still can be fulfilled, must be done. We solved this problem by using the constraint solver choco [5]. To represent the constraints, we generated a matrix $\mathcal{H}$. Each row in the matrix represents one abstract component descriptions. Each row vector $\mathcal{V}$ describes a component that can implement these abstract component descriptions. With the help of this representation the data can be modeled as follows:

- For all entries i of all vectors $\mathcal{V}$ in the matrix $\mathcal{H}$, a variable $\mathcal{E}(i)$ is generated
- The domains of these Variables $\mathcal{E}(i)$ are restricted, based on the component they implement. It is limited to the domain $\{0, \mathcal{B}(\mathcal{V})^{\mathcal{K}(\mathcal{H})+1}\}$, where $\mathcal{B}(\mathcal{V})$ denotes the maximum number of entries of all vectors $\mathcal{V}$ and $\mathcal{K}(\mathcal{H})$ is the index of the vector $\mathcal{V}$ within the matrix $\mathcal{H}$.

With this model all combinations of alternative components that are necessary to fulfill the given task can be determined using the following constraint:

$$\sum_{i=0}^{\mathcal{N}(\mathcal{H})-1} \mathcal{E}(i) \overset{!}{=} \sum_{i=1}^{\mathcal{M}(\mathcal{H})} \mathcal{B}(\mathcal{V})^i$$

Here, $\mathcal{N}(\mathcal{H})$ denotes the total number of entries of all vectors $\mathcal{V}$ within $\mathcal{H}$ while $\mathcal{M}(\mathcal{H})$ denotes the number of rows within the matrix $\mathcal{H}$. This ensures that all values of $\mathcal{E}(i)$ are zero, except for a single entry between all entries $\mathcal{E}(i)$ that share the same domain. This single non-zero entry is equal to the chosen component among all component alternatives that implement the same main component. To retrieve the components represented by the values $\mathcal{E}(i)$ the index i is used as an index in an array containing the component names.

C. User Interface

In this section, the relevant components of our graphical user interface are described. The GUI itself is structured into separate tabs which all fulfill different tasks.

1) Defining Source Ontologies: The GUI features a tab that empowers the user to load any desired ontology (Figure 1). The definition of more than one ontology will result in a single model that contains all relationships. For this, the user is provided with a list that contains all ontologies added so far at the top of the tab. At the bottom, there are two buttons, one to add another ontology and another button to load the defined ontology files. Upon a click on the Load button, the ontologies will be loaded and scanned for capabilities, components and other important information. For this, the user may define keywords that identify components and capabilities, within the ontology, in the Settings tab. This generic approach should guarantee that the software can also incorporate different ontology sets.

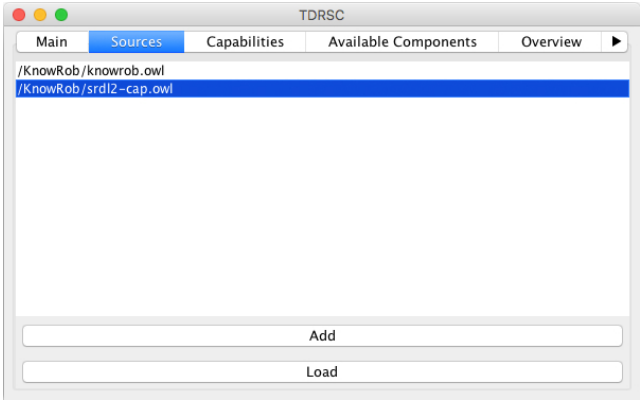


Fig. 1. The Sources tab of the GUI. Here the user may define the paths to the ontologies which are to be loaded into the model.

2) Defining Capabilities: After having loaded the desired ontologies, the user may define the desired tasks. There may be multiple of them or just one. This can be configured in the Capabilities tab as depicted in Figure 2. In this tab, desired tasks may be added using the Add button at the

bottom. This will add a combo box with all the previously extracted capabilities, of which the desired one may be chosen. Additionally, the list of chosen capabilities may be saved to disk.

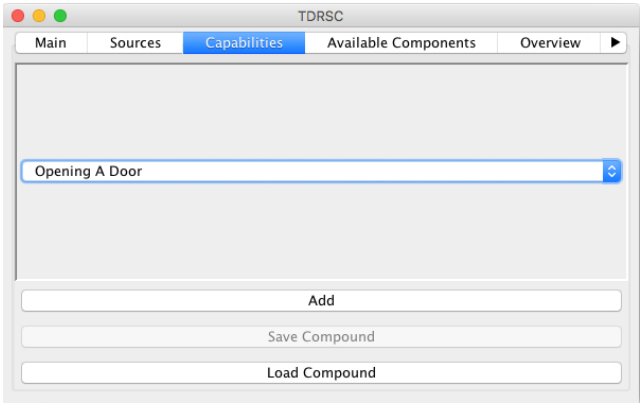


Fig. 2. The Capabilities tab of the GUI. Here the user may define tasks the robot should be capable of.

3) Defining Available Components: This tab is structured exactly like the capabilities tab. Here the available components (hardware as well as software) may be defined.

4) Feedback: Our program automatically analyses the situation anytime the user makes a change to the task requirements or available components. The result of this analysis is depicted in the Overview tab (Figure 3). It is divided into two sections including tree views. The left tree depicts the relationships between the chosen tasks and any subtask that describes parts of it. Also, it shows which components are necessary to implement these subtasks. On the right side, the user is provided with an overview of all components that need to be available to implement the desired tasks. In case the program could find several components that can be used to implement the same task, the component may be expanded and checked for the available options. In this view, missing components are depicted in red while available components (as defined within the Components tab) appear in green color (Figure 4).

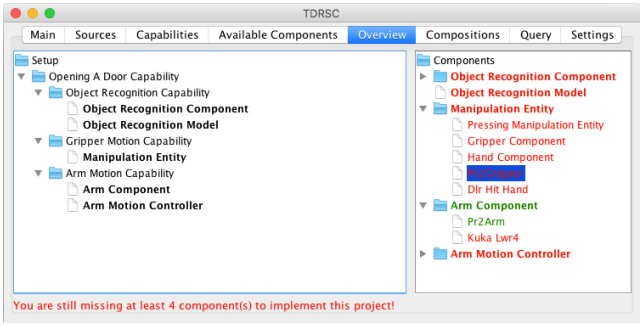


Fig. 3. The Overview tab of the GUI. Here the user gets a feedback about the given situation.

5) Configuration Proposals: As the desired tasks, may be implemented with a wide variety of different constellations

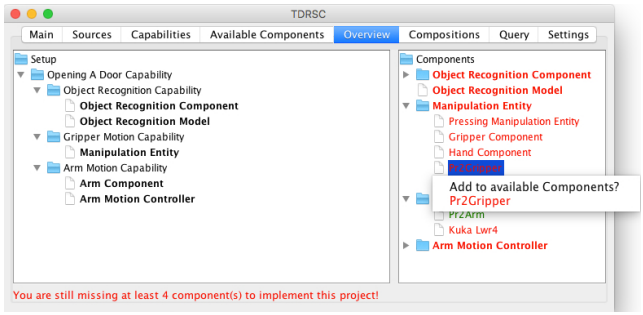


Fig. 4. Adding available components from within the Overview tab, using a pop-up menu.

of components, the GUI also features a tab that suggests possible component setups that fulfill the desired tasks. This is depicted in Figure 5.

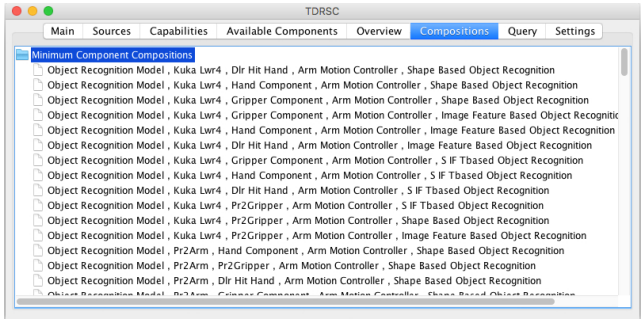


Fig. 5. The Compositions tab of the GUI. Here the user is provided with a list of all configuration that can solve the desired task.

6) Manually Query Data: To manually query the loaded data, the GUI also features a Query tab (Figure 6). In this tab, the user may define custom queries on the loaded model. For convenience, the program extracts all available prefixes that can be added with just a few buttons clicks. The result of the query will be displayed in a separate pop-up window as depicted in Figure 7. 5.

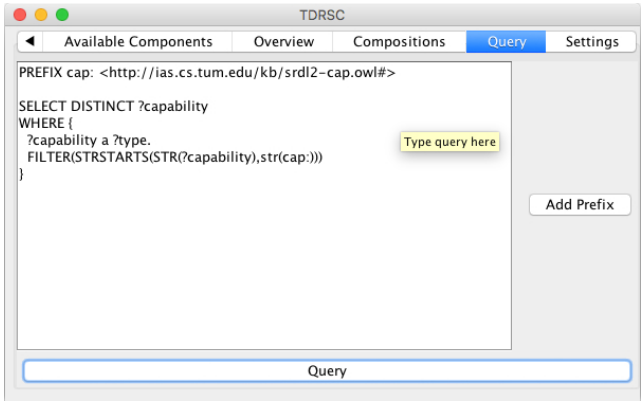


Fig. 6. The Query tab of the GUI. The user may query the data manually using this tab and the SPARQL query language.

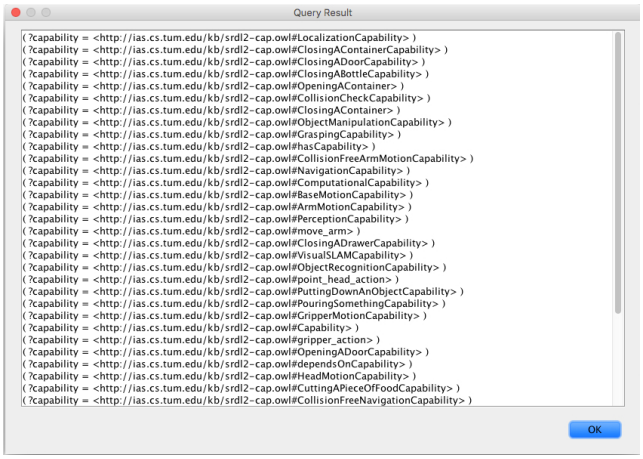


Fig. 7. The result of a user defined query is depicted in this pop-up window.

III. Example Scenario

Before we discuss the related research, we will discuss a simple running example, showing the different steps of a configuration. If one wants to know the minimal set of required components, necessary to implement a robot that can open a door one can use our program through performing the following steps:

- In order to identify the necessary components (hardware as well as software) using our program, first the ontology stated in section II-A, using the Source tab as depicted in Figure 1 must be loaded. After the loading process is done, the program has identified the capabilities and components, defined in the given ontologies. For the example, we use the ontology of [6] which contains all necessary capabilities.
- Now the capability ”Open a door“ should be available for selection in the Capabilities tab. It can be selected, by using the Add button at the bottom of the tab and selecting it in the newly created combo box as depicted in Figure 2.
- The program analyses the given situation online. Therefore, now, the user can already check for the necessary components to achieve the task in the Overview tab. If one has already components in mind which should be used, one may define them in the Available Components Tab. Assumed there is a ”Pr2Arm“ component that should be used. One can add these components before or after checking the necessary components.
- Now the user may want to check the necessary components to implement this task. If the ”Pr2Arm“ was added in advance the output of the Overview tab will be as depicted in Figure 3. There is also a possibility to add available components directly from within this overview. For this one may simply right-click the desired component and click the pop-up menu (Figure 4)
- Alternatively one may also check all possible constellations to implement this task by looking at the Compositions tab as depicted in Figure 5.

Optionally, custom SPARQL queries on the loaded model may be performed using the query tab. An example query to retrieve all available capabilities of the loaded model is depicted in figure 6. The result (all available capabilities) is shown in a pop-up as shown in Figure 7.

IV. Limitations of the Approach

The approach presented allows an easy specification of the robotics requirements and its resulting configuration. Additionally, one can generate a minimal configuration for the robot. These calculations are based on an ontology which describes the necessary dependencies to perform a task. Due to this specification, one may encounter several problems.

First, the ontology used in the example specifies the requirement for a home like an environment. The requirements may differ in a factory environment or on a planetary mission. To cope with this problem one could argument the requirements with a specification which environment the robot is operating in. Thus, one could add the information of the environment to the ontology to derive the proper set of requirements.

Another important limitation is the abstraction of the ontology. Let’s consider the example which specifies that one needs a robotic arm to fulfill the task. Thus, one can choose an arbitrary arm which may not be possible in practice as the arm does not allow to create enough force to perform the task or is too heavy to be placed on the robot. To tackle this problem one need to add additional constraints which need to be considered like the force which needs to be applied, maximum weight, ... Such constraints may not be simply integrated into the ontology reasoning. Instead one may add an additional layer of reasoning to check these constraints. Thus, one could find a configuration per the ontology and afterward check the additional constraints to rule out not applicable configurations.

V. Related Research

We start our discussion of related research with the semantic robot description language (SRDL) published in [7]. The description language allows describing the capabilities of the robot as well as the hardware and software components. Furthermore, dependencies of the capabilities and the components can be described. This description allows the robot to check if the dependencies are met for a specific capability. Additionally, the robot can enumerate all components which are missing for a specific capability. Thus, the first step for an automatic configuration of the robot is possible. To use this description in a robotic system SRDL was integrated into a general knowledge base for a robot through KnowRob [8]. This integration was used in the RoboEarth language [6] to allow an easy transfer of action recipes to perform a task. With the help of the SRDL, the robot could check if a certain action recipe to perform a task can be used. In this paper, we used SRDL as a basis for our tool to allow the derivation of a minimal configuration. Thus, instead of just checking if a robot can perform a capability our tool also allows getting

a minimal configuration such that the robot can perform a capability.

Another method which uses an ontology to describe the environment was presented in [9]. The method uses a description of the environment which is based on an ontology together with a description of skills the robot can perform. Using this description, the robot can plan a sequence of skills which need to be executed to perform a certain task. Such a task was presented in [10] where the robot had to place parts of an industrial kitting operation.

To plan which robot can perform which task in a heterogeneous group of robots the method outlined in [11] can be used. The method defines capabilities which have preconditions which need to be met to allow the execution as well as information which need to be provided by the robotic system to allow the capability to be executed. Thus, the robot can plan which capabilities need to be executed to perform a task. Furthermore, the robot can use the hardware description to check if such an execution is possible. As many capabilities, e.g. grasp an object, may only work under certain restrictions, e.g. size of the object, one can add an approximation description to each capability which defines which conditions need to hold approximately to allow the execution of the capability. This allows to define capabilities in more detail and thus allow a better distribution of tasks among the robotic group. The method outlined in [11] focus on distributing tasks in a group of robots whereas the tool we present in this paper focuses on the configuration of the robotic system during the design phase. Thus, instead of planning which capabilities to use to fulfill a task, we show which different minimal configurations of the robotic system allow the robot to execute the capability. This allows the developer of the robotic system to choose the best fitting set of capabilities.

The method presented in [12] extends AutomationML [13] to allow the modeling of a robotic system. This is done by extending the given concepts with robotic specific concepts such as actuators or sensors. Furthermore, the method allows an automatic conversion of AutomationML specifications into an ontology which can be used to check for consistency. This can be used to model a robotic system with AutomationML and afterward check through the transformation to an ontology if every dependency is met or if a component is used which does not solve any given task. Beside these checks, further checks can be applied to verify that this system can be realized with the help of ROS [14]. This allows the developer to ensure that the modeled robot can be realized. After these checks are performed one can apply a model-to-text transformation to generate code stubs which ensure proper communication and operation per the modeled system. This allows a faster creation of a robotic system following a model-based approach like the method outlined in [2]. In contrast to our approach, the method does not offer the possibility to check which components are necessary to perform a capability. Thus, the method presented in [12] is also not able to specify a minimal configuration which fulfills the need for a specific capability as our tool can.

Besides the description of tasks for configuration, such a description is also often used to assign a task, e.g. in a multi-robot system. One such example is which uses an ontology to assign tasks is presented in [15]. The method uses an ontology per robot to describe which roles can be performed and how these roles are performed. Additionally, tasks are described in the ontology and how they can be executed through a role which is assigned to a robot. The system uses this ontology to find a matching robot and assigns different roles for different robots to fulfill the specified task.

VI. Conclusion and Future Work

The proper configuration of a robotic system for a given task is a time consuming and tedious task. Especially one needs an expert to perform this task to consider all possibilities as well as all dependencies. To address this problem, in this paper we presented a tool for the automatic configuration of a robotic system for a given task. The tool uses an ontology-based knowledge base, allowing to reuse publicly available knowledge bases, to describe which dependencies, exist between a task and software and hardware components. Furthermore, we have presented a method to derive a minimal set of software and hardware components to fulfill a certain task. This allows the user to simply find a possible configuration of the robotic system, that allows the robot to fulfill its task. To allow an easy interaction the tool has a graphical user interface which allows the user to select tasks as well as used components. Thus, the user can specify the currently used components on the robot to check if a new task can be achieved by the robot, or which components need to be added to allow the robot to achieve a given task.

Currently, the tool can only be used by a human to decide which components to use to allow the execution of a specific task. It is left for future work to allow the robot itself to use the tool. This would open the possibility that the robot finds alternative solutions to a task during runtime. Thus, the robot could reconfigure itself to react to a fault or changes in its task. Furthermore, currently only a minimal number of components is searched for the configuration, neither computation costs nor investment or development costs are considered in the configuration. It is left for future work to integrate these costs to allow to find a configuration which minimizes the computation effort or to minimize the investment costs.

References

[1] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[2] G. ”Steinbauer and C. Mühlbacher, “Hands off - a holistic model-based approach for long-term autonomy,” in *Workshop on AI for Long-Term Autonomy, 2016 IEEE International Conference on Robotics and Automation (ICRA)*.

[3] Knowrob.org. Capability ontology - knowrob. [Online]. Available: http://knowrob.org/kb/srdl2-cap.owl

[4] Apache.org. Jena framework - apache. [Online]. Available: https://jena.apache.org/

[5] choco solver.org, “choco-solver.” [Online]. Available: http://www.choco-solver.org/

[6] M. Tenorth, A. C. Perzylo, R. Lafrenz, and M. Beetz, "The roboearth language: Representing and exchanging knowledge about actions, objects, and environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1284–1289.

[7] L. Kunze, T. Roehm, and M. Beetz, "Towards semantic robot description languages," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5589–5595.

[8] M. Tenorth and M. Beetz, "Knowrob: A knowledge processing infrastructure for cognition-enabled robots," *The International Journal of Robotics Research*, vol. 32, no. 5, pp. 566–590, 2013.

[9] F. Rovida and V. Krüger, "Design and development of a software architecture for autonomous mobile manipulators in industrial environments," in *Industrial Technology (ICIT), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3288–3295.

[10] A. S. Polydoros, B. Großmann, F. Rovida, L. Nalpantidis, and V. Krüger, "Accurate and versatile automation of industrial kitting operations with skiros," in *Conference Towards Autonomous Robotic Systems*. Springer, 2016, pp. 255–268.

[11] J. E. Buehler, "Capabilities in heterogeneous multi robot systems." in *Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI) Doctoral Consoritum*. AAAI, 2012, pp. 2380–2381.

[12] Y. Hua, S. Zander, M. Bordignon, and B. Hein, "From automationml to ros: A model-driven approach for software engineering of industrial robotics using ontological reasoning," in *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*. IEEE, 2016, pp. 1–8.

[13] R. Drath, A. Luder, J. Peschke, and L. Hundt, "Automationml-the glue for seamless automation engineering," in *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*. IEEE, 2008, pp. 616–623.

[14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[15] F. Amigoni and M. A. Neri, "An application of ontology technologies to robotic agents," in *Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*. IEEE, 2005, pp. 751–754.

An Autonomous Transportation Robot for Urban Environments

Konstantin Lassnig[1] and Clemens Mühlbacher[2] and Gerald Steinbauer[2] and Stefan Gspandl[3] and Michael Reip[3]

***Abstract*— The transportation of goods is a central task of today's economy. The cheap transportation of goods allows the wide spread of today's internet based sales. To perform such transportation tasks one currently relies on humans. This imposes constraints when the transportation can be performed and imposes constraints on the costs. To address this time and cost constraints an automatic transportation of goods is preferred.**

Such an automatic transportation can be performed by an autonomous robot, as the ones used in warehouse environments. Although such environments are diverse and undergo a certain amount of change they are still rather static environments. To allow robots to perform the transportation in outdoor environments several problems need to be tackled. One needs to deal with large operation areas, uneven ground, and dynamic objects. In this paper, we present a robot system which can cope with these problems and allows to perform transportation tasks in outdoor environments. The focus of this paper will be on the localization and navigation of the robotic in the outdoor environment allowing the robot to perform outdoor deliveries.

I. Introduction

The cheap transportation of goods is a central part of today's economy. Reasonable prices of goods which are sold over the internet, heavily depend on transportation costs. Today's supply chain requires a very dense distribution network and relies on the fact that sending a lot of packages on the same route is cheap. The larger number of goods for one route the cheaper it becomes. This is in contrast with the need for transporting goods to a single customer. Such a transportation is characterized by a few goods for one transportation route. To address this, robots offer a possible solution. Using a robot, the transportation can be performed in a flexible manner. Additionally, if multiple robots are used one can simply balance the load of transportation tasks on several robots.

The use of a robot fleet for transportation tasks is getting adopted for warehouse environments nowadays [1], [2], [3]. These robot systems allow transporting goods in the warehouse without the need of an adaption of the warehouse. This is achieved by using algorithms allowing a localization and navigation in an indoor environment [4]. These algorithms use a 2D map of the environment. Such a map can be stored easily in the robots memory for a warehouse but not for large outdoor environments such as a city. Furthermore, the 2D map can be easily used in a warehouse for navigation as one can assume a reasonable flat ground. Such an assumption cannot be made for an outdoor environment where the robot needs to ensure that it is not falling over road curbs.

To allow a robot system to be used for transportation tasks in a large scale outdoor environment, one needs to address the problems which are imposed by the scale of the environment as well as the uneven ground. In this paper, we show a robot system which addresses these problems. The size of the environment is addressed by splitting the environment into smaller areas allowing the robot to keep only a small map in its memory. To allow the robot to be globally localized one additionally stores how the small pieces are related to each other. To tackle the uneven ground only the area close to the robot needs to be considered. This space is represented as a 2.5D surface and interpreted to find possible holes.

The remainder of the paper is organized as follows. In the next section, we will discuss the software system used by the robot to perform transportation tasks in an outdoor environment. The proceeding section discusses how the robot localizes itself despite the size of the environment. In Section IV we discuss how the robot navigates in the environment. This section also comprises a description how the robot deals with the uneven ground. Afterward, we discuss some related research. Finally, we conclude the paper and point out some future work.

II. System overview

Fig. 1. The transport robot [5].

In this paper, we discuss a robot which can perform a transportation task on a university campus autonomously. The robot can navigate indoor as well as outdoor. Furthermore, the robot considers the uneven ground outdoors to

[1]Konstantin Lassnig is with ARTI, Graz Austria. This author was with the Institute for Software Technology when contributing, Graz University of Technology, Graz, Austria. klassnig@arti-robots.com

[2]Clemens Mühlbacher and Gerald Steinbauer are with the Institute for Software Technology, Graz University of Technology, Graz, Austria. {cmuehlba,steinbauer}@ist.tugraz.at

[3]Stephan Gspandl and Michael Reip are with incubedIT, Hart bei Graz, Austria. {gspandl,reip}@incubedit.com

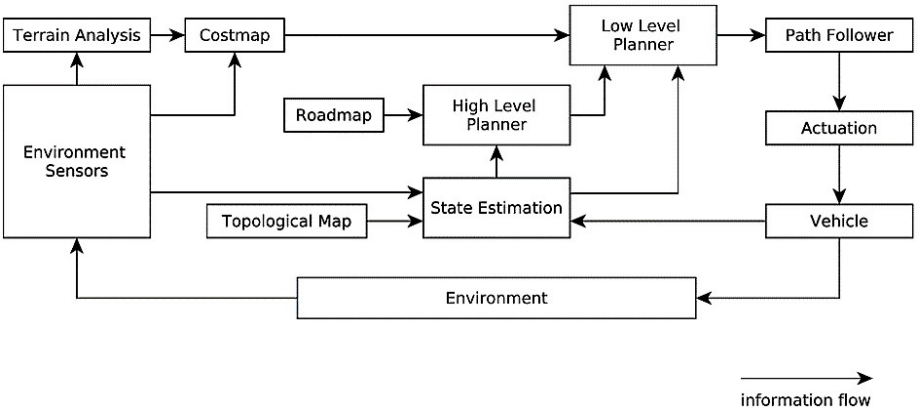


Fig. 2. System overview of the transport robot [5].

safely navigate between buildings. The robot is depicted in Figure 1 and is based on a pioneer 3-AT platform which allows the robot to navigate indoors as well as outdoor. Additionally, the robot has three laser scanners to detect obstacles. Two laser scanner are mounted horizontally to detect obstacles, like cars or people. Furthermore, these two lasers are used to localize the robot. The third laser is mounted tilted down to scan the ground in front of the robot. This laser is used to build a map of the local terrain. Besides the laser scanners, the robot has a GPS sensor for the localization and mapping. To improve the accuracy of the odometer of the robot an inertial measurement unit (IMU) is used which is mounted on the robot. To perform the transportation task, the robot uses the system architecture depicted in Figure 2.

The robot uses its sensors to estimate the current location. This is done using the robots odometer, the IMU, GPS and the horizontal laser scanners. Due to this redundancy, the estimation of the current location is stable in areas where one sensor may yield wrong results, e.g. the GPS sensor near tall buildings. To perform the estimation, the robot matches the sensor readings with the information of a topological map. The topological map consists of several small maps which are linked to each other to allow the robot to only keep small maps to be localized.

Using the estimation of the current location together with a road map, the robot plans a high-level path for navigation. The roadmap describes possible traversal routes in the environment on a higher level. Due to this abstraction, the planning can be done very efficiently even in the case of large environments. After generating a high-level plan, the plan is passed to the lower-level planner which tries to find a valid path in the environment for each path segment in the high-level plan. This is done by considering the current small local map of the environment as well as the sensor data which are used to build a cost map. If a valid path is found the robot tries to follow this path as accurate as possible.

To incorporate the information of the terrain the robot uses the tilted laser to perform a terrain analysis. Afterward, the results of this terrain analysis are used to update the cost map. Thus, holes, as well as small objects which are below the horizontal laser scans but bigger than the robots clearance, are added to the cost map as obstacles. This allows the robot to consider the terrain in the low-level planning.

In the following two sections, we will discuss the localization as well as the navigation in more detail.

III. Localization

Starting from an initial known position the robot needs to know its location during the entire delivery. This is done through one part of the robot system which is used to localize the robot. This localization should ensure that the robot has an estimation of its global position. First, the robot corrects its odometer to get a good estimation of its 2D position using dead reckoning. Afterward, it uses the created topological map to localize itself.

To correct the odometer of the robot we use an unscented Kalman filter (UKF) [6]. The Kalman filter uses the raw odometer of the robot to perform a prediction of the robot pose. This prediction is formed in a probabilistic manner with a position and a covariance matrix specifying the uncertainty. The covariance matrix is defined in such a way that the linear speed has a higher accuracy as the rotational speed, as the rotation is badly estimated through the raw odometry due to the slippage of the wheels during rotation. To correct the prediction the IMU data are used. The IMU data is used to provide an additional estimation of the robots velocity in all three axes as well as the global orientation the robot has in space. As in the case of the raw odometry, the IMU data update the estimate in a probabilistic manner with the help of a covariance matrix. The covariance matrix for the IMU data is formed in such a manner that the rotational speed is estimated more accurately than with the raw odometry. Due to the use of the Kalman filter, we have a better estimation of the robot pose instead of the very noisy raw odometer of the robot.

After the odometer is corrected the robot can perform its estimation on the topological map. The topological map is a graph with vertices which represent positions in the world and edges which represent connections between those posi-

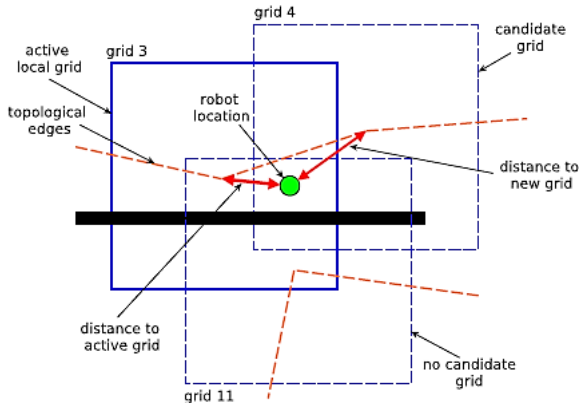


Fig. 3. Grid selection for the localization, together with the topological map [5].

tions. Each vertex is specified as a full 2D pose in the global reference frame allowing to specify the difference from the robots location to any frame in the graph. Furthermore, each vertex contains a grid map representing the local environment at this position. It is ensured that every position within the grid map can reach the center. To ensure proper connections of vertices a connection is only made if the combination of both grid maps allow the robot to reach one vertex from the other. Let's consider the simple example of a topological map as it is depicted in Figure 3. Grid 11 is close to grid 3, but due to the wall between these two grids, no connection between grid 11 and grid 3 is made. Thus, the robot knows which traversals are possible in the environment with the help of the grid map. We will exploit this knowledge to select the right grid for localization if the robot moves beyond the area of one grid.

Initially, the robot knows its starting location, this is done through the input of the user. After the robot has selected the initial position the vertex which is the closest to the current initial location is chosen. Additionally, the robot should check if it can move between the initial pose to the grid map vertex. Using this vertex, the robot can use the grid map of the vertex to localize itself. This is done with the help of a particle filter [7]. The particle filter uses the grid map to align the current laser measurements with the occupied cells in the grid map. Additionally, the robot uses the GPS signal to localize itself. This is done by anchoring each vertex with a GPS position. Thus, by using the current GPS signal the robot can estimate its position relative to the currently used vertex in the topological map. Using the grid map and the GPS the robot derives an estimation of its current location. If the robot is moving in the grid map the localization can be done with the current grid map. But as we assume a large space of the outdoor environment the robot will at some point reach the border of the grid map. In such a case the robot needs to decide which vertex in the topological map is the next one to localize itself. This is done by checking the distance to each vertex in the topological map which has a connection to the currently used vertex. The vertex with the smallest distance to the current robot pose is used for future localization. Thus, the robot will switch the vertex and the occupancy grid only if it is closer to that vertex than any other vertex which could be reached from the robot.

Let's consider the situation in Figure 3. If the robot is moving from grid 3 to grid 4. It checks the distance from the vertex of grid 3 and the distance of vertex of grid 4. But the robot does not check the distance to the vertex of grid 11 as the robot already knows that there is no possibility that it has traveled from grid 3 to grid 11. If the distance of grid 4 is larger than the distance to grid 3 the robot will use grid 3 for future localization.

Due to the use of the connections within the topological map one saves the effort to check all nearby vertexes if they should be used for localization. Furthermore, more important is that the robot will not select a vertex which cannot be reached. Let's consider the map in Figure 3. Grid 3 and 4 can be on the outside of the building whereas grid 11 is the inside of the building. Thus, if the robot is localized outside of the building it does not make sense that the robot jumps suddenly through the wall inside the building. As we don't consider grid 11 as an alternative such a jump is not possible. This also allows the robot to move close to the wall of the building without being incorrect localized.

IV. Navigation

With the help of the topological map, the robot can localize itself. Using this localization, the robot can plan its path to the destination. To do so, the robot uses a hierarchical planning approach. As we consider a large-scale environment the robot is not able to use a grid map of the complete environment. Thus, the robot uses a road map to plan the overall navigation. This allows the robot to plan for the large environment in a fast manner. After a plan is found using the road map the robot use the current grid to search for a mid-level plan within this map to move between different nodes in the roadmap. Finally, the robot uses a local planner to move along the mid-level plan and avoid obstacles which are not present in the grid map.

The road map which is used by the robot to generate a high-level plan consists of a graph of nodes which specify locations and edges which describe possible traversals between these nodes. A roadmap together with the high-level plan is depicted in Figure 4. The road map is constructed by considering the distance between the nodes and if the node is collision free. To check if a node is collision free the footprint of the robot and the local grid map of the position to check is used. As the robot, has not specified a complete description of the environment traversability, one uses the positions used during mapping as a seed for the road map calculation. This allows that the robot uses the positions and traversals which were created during mapping.

To plan a path within the road map the robot first determines the closest node of the road map to its current location. Afterward, the closest node to the destination is determined. The node close to the current position is the start node of the search and the node close to the destination is the goal node of the search. After determining these two nodes the

robot performs a graph search for the shortest path through the A^* algorithm [8].

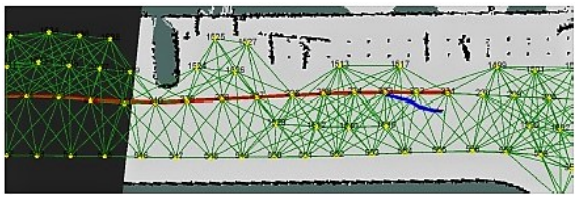


Fig. 4. Road map (green) together with the high-level plan (red) and the low-level plan (blue) [5].

After the robot has generated a high-level plan it generates a mid-level plan on the current grid map to plan to the next node in the road map which is part of the high-level plan. The node which is the next one to pass through is determined by the current location of the robot. The robot considers every node as reached which is in a certain range. To determine this node, the robot uses a queue of nodes within the high-level plan. After the head of the queue is in range the robot pops the head from the queue and uses the new head of the queue as the next goal to plan to. Additionally, if the node is the last node in the queue the robot plans to the destination as the high-level plan only ensure that the robot moves near the destination.

For the mid-level plan on the current grid map, the robot uses the information of the current grid map to determine if it can traverse a grid cell or not. Using this information, the robot uses its current location together with the next node to find a plan. This plan consists of a sequence of grid map cells to traverse. The sequence is found by using the A^* algorithm [8]. As the grid map only specifies a limited area of the world the algorithm can determine the path very fast. Additionally, the path which needs to be planned is most of the time short compared to the high-level plan.

Using the mid-level plan on the current grid map the robot has derived a path which should lead to the current node of the high-level plan considering the known static objects. As we consider a dynamic environment the robot needs to deal with these obstacles as well. This is done by creating a local plan with the help of the dynamic window approach [9]. The local plan is generated several times per second to allow to react to changes. To plan locally the robot uses a cost map which contains the static obstacles, the information from the horizontal laser scan, the information from the elevation map and information about the grass around the robot.

As we argued above one cannot assume that the robot moves on a flat surface. Thus, the robot needs to deal with the uneven ground. Through the construction of an elevation map in a local area, the robot can detect holes and barriers. The elevation map is constructed with the help of the sensor data of the tilted laser. Each of the laser measurements is transformed to specify a position in the world frame. Afterward, the measurement is projected on a grid which defines the height information of the environment. To in cooperate the sensor measurement into the grid a Bayes update per grid cell is used [10]. This allows the robot to deal with the noise of the sensor measurements. After generating the height information one detects holes and barriers by deriving the gradient for each grid cell. Using this gradient one can define a threshold which determines if this hole or barrier is traversable by the robot. If the gradient exceeds a certain limit the robot cannot traverse this grid cell and it is assumed to be a lethal obstacle for the local planner. An example of grid cells which are marked due to a large gradient is depicted in Figure 5. As the elevation map is projected through the gradient into the cost map one can use a standard 2D-planning algorithm to find a local plan.

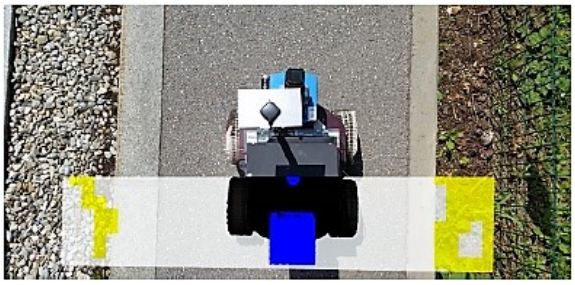


Fig. 5. Detection of edges with the help of the elevation map [5].

Besides the uneven ground, the robot needs also to consider the grass to proper navigate in an outdoor environment. During outdoor navigation, the robot should preferably stick to roads and sidewalks. Thus, the robot needs to detect the grass surrounding the robot. To perform this detection, the robot uses the tilted laser scanner. The tilted laser scanner does not only provide the information about the distance from obstacles but it also contains the information about the intensity of the reflection. Using the intensity and the distance one can identify grass in the environment. A simple linear separator is sufficient to detect grass properly. This relation between distance and reflection intensity with the linear separator is depicted in Figure 6. With the help of this classifier, the robot can detect grass in its vicinity. An example of this detection is depicted in Figure 7. Using this information, the robot adds increased costs in the cost map for the local planner on those positions which indicate grass. Thus, the robot avoids the grass if possible but will also traverse it if necessary.

By combining the grass, the information of the elevation maps, the laser scan measurements of the horizontal laser as well as the static objects in the map the robot can safely navigate locally. Thus, the robot neither hits an object nor falls down a step. We consider these data only for a local area around the robot. This has the benefit of a smaller memory footprint for each local cost map but also the drawback that this information cannot be used for localization or high-level navigation.

V. Related Research

Before we conclude the paper, we discuss some related research.

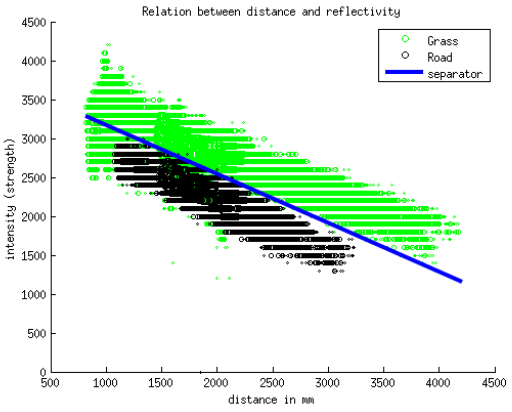


Fig. 6. Separator to detect grass with the help of the laser scanner [5].

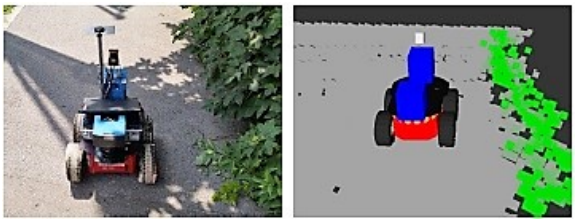


Fig. 7. Detection of the grass though the laser scanner [5].

We start our discussion of related research with the robot presented in [11]. The robot could take a long tour through Munich without a prior created map or GPS information. Instead, the robot was using its sensors to react locally in a safe manner and asked humans for information about the direction. This was done by approaching humans and the recognition of basic commands to derive the direction of the desired destination. In contrast, our robot has a prior created map which allows it to move autonomously without asking for directions. This is also desirable in the case of a transport robot which should transport goods to a customer.

In [12] the method to deal with large maps was described. The authors use a topological map to allow an efficient representation of large areas. The vertices in the topological map are spots of interests such as a square or a crossing. The edges represent paths between these places. For each edge, a traversal behavior is defined. Thus, one can use different behaviors to perform the traversal. With the help of this method, the robot could drive autonomously in a park. Our robot uses, in contrast, a topological map which contains enough information to allow the robot to be always localized not only in interesting places. Furthermore, the robot uses a denser road map allowing it to plan its route more accurate.

A very close related work to ours was presented in [13]. The robot navigated more than 3km in the city Freiburg in an autonomous fashion. To localize itself, the robot used a topological map where each vertex in the graph contains a map of one part of the environment. In contrast, our approach additionally used the GPS signal for estimating the robot pose within the particle filter. To navigate the robot, the method presented in [13] created a high-level plan using the graph of the topological map. Each vertex is connected to those vertices in the graph which allow moving between these two locations. Thus, using this graph the robot can derive a simple high-level plan for the navigation. Whereas the robot uses a planner on grid map basis to navigate between different vertices of the topological map. This contrasts with our approach as we use a finer grained road map for the high-level planning which allows us to choose the path more precisely.

VI. Conclusion and Future work

The transportation of goods is an essential part of our today's economy. The transportation often takes place in outdoor environments by delivering goods to costumers. To provide cost-efficient and flexible deliveries, robots are a promising solution.

In this paper, we presented an autonomous transport robot which is capable of navigating in large scale outdoor environments. To perform this transportation, the robot addresses the problem of a large-scale environment, uneven ground, and grass which should only be traversed if necessary. To deal with the large scale of the environment the robot uses a topological map. This map stores areas of the environment which are loaded on demand. This allows that the robot only needs to keep a small part of the environment in its memory and perform the localization on it. We furthermore showed how the robot can exploit the topological map to switch between the different parts to allow the robot to be localized during the complete delivery. To deal with the uneven ground, the robot builds an elevation map for its local environment. Afterward, the robot determines within the elevation map dangerous terrain and avoids it. To deal with the grass we have shown a simple solution with a linear classification for laser scan measurements. This detection allows the robot to detect grass precisely enough to avoid the grass if possible.

The robot presented in this paper mainly used several laser scanners to localize itself and it is left for future work to add more sensors to perform localization as well as navigation. Especially cameras would be of interest as they allow a detailed localization in many areas which don't offer features for a laser scanner. The additional use of a camera would increase the quality of the terrain classification.

References

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," in *Proceedings of the 19th national conference on Innovative applications of artificial intelligence - Volume 2*, ser. IAAI'07. AAAI Press, 2007, pp. 1752–1759. [Online]. Available: http://dl.acm.org/citation.cfm?id=1620113.1620125

[2] E. Guizzo, "Three Engineers, Hundreds of Robots, One Warehouse," *Spectrum, IEEE*, vol. 45, no. 7, pp. 26–34, 2008.

[3] C. Mühlbacher, S. Gspandl, M. Reip, and G. Steinbauer, "Improving dependability of industrial transport robots using model-based techniques," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3133–3140.

[4] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[5] K. Lassnig, "An autonomous robot for campus-wide transport tasks," Master's thesis, Faculty of Computer Science and Biomedical Engineering, Graz University of Technology, 2016.
[6] S. J. Julier and J. K. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *AeroSense'97*. International Society for Optics and Photonics, 1997, pp. 182–193.
[7] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," *AAAI/IAAI*, vol. 1999, no. 343-349, pp. 2–2, 1999.
[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
[9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
[10] A. Kleiner and C. Dornhege, "Real-time localization and elevation mapping within urban search and rescue scenarios," *Journal of Field Robotics*, vol. 24, no. 8-9, pp. 723–745, 2007.
[11] A. Bauer, K. Klasing, G. Lidoris, Q. Mühlbauer, F. Rohrmüller, S. Sosnowski, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss, "The autonomous city explorer: Towards natural human-robot interaction in urban environments," *International Journal of Social Robotics*, vol. 1, no. 2, pp. 127–140, 2009.
[12] K. Košnar, T. Krajnık, V. Vonásek, and L. Preucil, "Lama-large maps framework," in *Proceedings of Workshop on Field Robotics, Civilian-European Robot Trial*, 2009, pp. 9–16.
[13] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "Autonomous robot navigation in highly populated pedestrian zones," *Journal of Field Robotics*, vol. 32, no. 4, pp. 565–589, 2015.

User-Centered Assistive Robotics for Production
-
Human-Robot Interaction Concepts in the AssistMe project

Markus Ikeda, Gerhard Ebenhofer, Jürgen Minichberger, Gerald Fritz, Andreas Pichler
Andreas Huber, Astrid Weiss

***Abstract*—In this paper we present results of the AssistMe project which aims at enabling close human-robot cooperation in production processes. AssistMe develops and evaluates different means of interaction for programming and using a robot-based assistive system through a multistage user-centered design process. Together with two industrial companies human-robot cooperation scenarios are evaluated in two entirely different application areas. One field of application is the assembly of automotive combustion engines while the other one treats the machining (polishing) of casting moulds. In this paper we describe the overall project methodology, followed by a description of the selected use case and a detailed outline of the first two expansion stages. The paper closes with an overview on the results of the first two rounds of user trials and gives an outlook on the next expansion stage of the human-robot cooperation scenario.**

I. INTRODUCTION

Traditional robot systems are programmed mostly offline with text based programming languages or by complex CAD/CAM based simulation tools. That is suitable for traditional robot systems used in specialized situations such as optimized and fenced working environments, only applicable for high production volumes. Robots for smaller production volumes (applicable for SMEs) would require two main success factors. That's on the one hand safe applicability without expensive safety hardware like dedicated workspace or fences. On the other hand systems would benefit from applicability for smaller production volumes and lot sizes which requires frequent reprogramming – ideally without expensive software tools or robot and computer vision specialists. Robot manufacturers address both safety and ease of use and reprogramming with contemporary products. Limitation of system power and implementation of safety relevant control system structures as well as safety relevant functionality like safely limited speed or workspace are used to make systems safe enough for even collaboration, as it is defined in the DIN ISO 10218 standard. Improved user interfaces should make systems useable without special training. Main modalities implemented by the system used (a Universal Robot UR10 system) are touch based programming with graphical elements as well as manual interaction by hand-guidance during parameterization of the programs. Human workers and the robots could work as a team through more flexible human-robot interaction [1]. But how to develop a robot system that meets the needs of its users in an industry 4.0 environment? An answer has to take User Experience (UX) into account, which – according to Alben [2] – comes everywhere into play were humans interact with a system. This includes cooperation and usability but also factors such as perceived safety, stress, or emotions [3]. The work presented in this paper illustrates how a UX study helped improving a standard-software to a physical interaction interface for real-world usage. A multistage user-centric design approach was performed, involving representative factory works performing user studies in their actual working environment. Finally we want to introduce a proposal of the improved interface to be evaluated at the very end of the AssistMe project.

II. RELATED WORK

The Industry 4.0 paradigm of close human-robot cooperation makes fundamental research necessary, not only in robotics, but also in user-centered HRI. Little research has been performed so far concerning industrial robotics, associated UX, and how HRI impacts production performance. Existing research already showed potential application scenarios of physical HRI [4] and that the UX of robots changes over time [5]. A methodological approach how to evaluate the usability of teach pendants for teaching a robotic arm was demonstrated by [6]. Current research for example is the learning of motor skills by pHRI [7] and the industry-oriented application [8]. The focus of our research follows a similar interest as [6] especially on how to use UX to improve a newly introduced robotic arm without a safety fence in a factory environment.

III. ASSISTME SYSTEMS

In the AssistMe project two usecases in three expansion stages are evaluated. One of the usecases is the assembly of a combustion engine. That includes the installation of a cylinder head cover. The installation is carried out manually by stacking the cover with pre-inserted screws onto the motor block and tightening the screws with a manual power tool.

Markus Ikeda, Gerhard Ebenhofer, Jürgen Minichberger, Gerald Fritz and Andreas Pichler are with Profactor GmbH, Austria (e-mail corresponding author: markus.ikeda@profactor.at)

Andreas Huber is with the Institute of Automation and Control, Vienna University of Technology, Austria (e-mail: huber.cognition@yahoo.com).

Astrid Weiss is with the Institute of Automation and Control, Vienna University of Technology, Austria (e-mail: Astrid.Weiss@tuwien.ac.at).

The electronic screwdriver of the manual workplace is fitted with a push start mechanism, electronic control unit and a shut-off clutch and therefore starts rotating when pushed onto the screw and stops motion when retracted respectively when a predefined torque is reached. The working instruction of the workstation includes several additional process steps. An automatic screw tightening system is expected to provide assistance and to reduce the workload at the workstation for the human worker. A state-of-the-art collaborative robot system is equipped with the power tool (Fig. 1) and programmed to perform screw tightening operations in the required order and accuracy to meet a defined process quality (screw-in depth, torque,…).

Fig. 1 - Usecase Combustion Engine Assembly – screw positions to be parameterized by the user.

A. *Robot A*

Robot A is a standard Universal Robot UR10 system with its teach pendant and the integrated programming and parameterization infrastructure. A basic script for the movement contains the pre-screwing process and can be called by the teach pendant program. The teach pendant program manages position variables (that have to be parameterized by the worker) and the execution of the global program to process the screws in the correct order.

Fig. 2 - www.zacobria.com - UR10 programming

B. *Robot B*

To be able to provide smooth and precise one hand-guidability a FT-sensor was integrated in robot B. Shortkey buttons trigger alignment shortcuts (Fig. 3). Preconfigured TCP alignments can be triggered and cause the tool to rotate around the TCP to move the tool intuitively to an (e.g. perpendicular) orientation to maximize process stability and robustness towards inaccurate teach-in of process points.

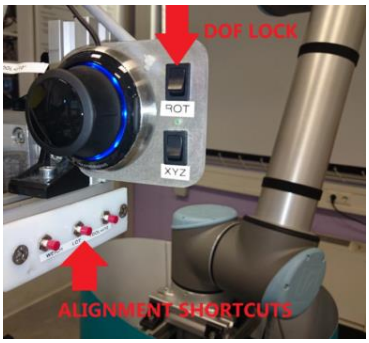


Fig. 3 - FT-sensor, shortcuts and DOF locks

The GUI of the robot controller interface was replaced by XROB, a PC-based robot programming system, that covers both robotics and sensors and algorithms to assess sensor data. Benefits are on the one hand simplification of the interplay between robotics and machine vision and on the other hand simplification of the programming experience for the robot (that was perceived as confusing with robot A).

XROB (Fig. 4) is capable to manage several sensors and evaluation algorithms. Program templates can be used to compose basic functionality to advanced and reusable subprograms. Prior to evaluation of robot B templates for a combined rough 3D position deviation compensation and a 2D position fine compensation were prepared for reuse by the workers.

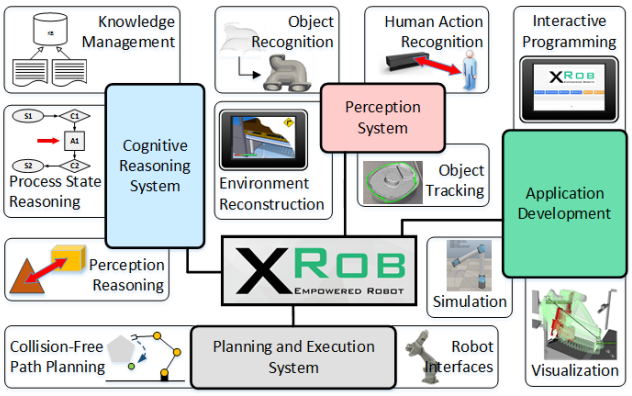


Fig. 4 - XROB framework

IV. USER EVALUATION

The goal was to explore if there is a difference in the UX between a robot with remote-HRI (robot A) and a technical revised version of this robot with physical-HRI (robot B). Both robots offered two control modes: remote control via touch-panel and direct-manual control via physical guidance. The touch-panel for remote control featured a graphical user interface consists of buttons to steer the robot and to save the taught movement trajectories. The physical-HRI mode enabled the operators to control the robotic arm directly, manually and without an additional intermediate layer. Robot A was optimized for remote control, whereas the improvement of robot B consisted of an extended physical HRI. Five participant were recruited to participate in the two studies. This small participant number can be sufficient to identify the most severe usability problems and was already discussed by [9].

The current study was conducted one year after the previous one. Within this time, robot A was upgraded to robot B, so robot B could only be examined after robot A. However, both studies had the same structure: (1) Introduction of the robot: Each participant was introduced to the robot and its control mechanisms. The participants were assigned the task to parameterize the process points in a predefined robot program. That means they had to bring the robot's tool to a precise position above the screw and that they had to adopt the position parameters to a program in the UR-teach pendant (for robot A) or to the XROB-user interface. Fig. 1 shows the screw positions as process points. For process quality precision of the parameterization is crucial. As Fig. 5 points out especially lateral or orientation deviances are critical for process effectivity while vertical positioning could be effectively observed visually during teach in process.

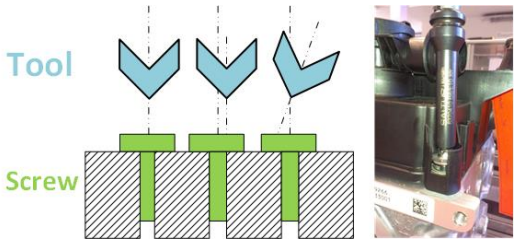


Fig. 5 - screwing process - error sources & real view

In order to relief stress and increase compliance, the participants were assured that the focus of investigation was only the robot's performance and there were no negative implications for them. (2) Conducting the user study: Each participant was audio- and videotaped with two cameras in order to generate a holistic perspective. This included a head mounted camera (first-person view - Fig. 6) and a hand camera (context oriented view). (3) Post-study questionnaires, including NASATLX, SUS, and self-developed items. The aim of the analysis was to compare the temporal demand, and the UX (including usability and performance expectancy) of the first and second version of the robot prototype. The findings are used for a the third (and last last) technical revision (design of the user interfaces of robot C, D and E) before the robot is deployed in the normal factory environment. The analysis of the video data (comments, reactions and feedbacks) consisted of (1) a rough clustering of all relevant issues, (2) a detailed description of their key features, and (3) overlapping topics were merged to categories or differentiated from each other.

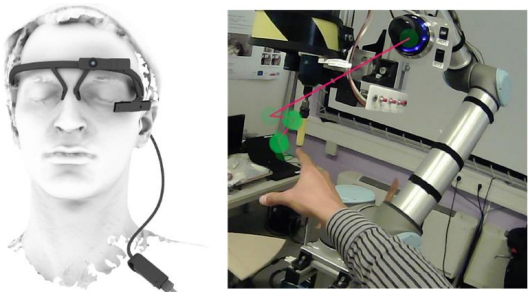


Fig. 6 – Head Mounted Device for gaze tracking - gaze tracking results

V. RESULTS

A total of five male assembly workers were recruited to participate in both studies (a representative sample for the factory with which we collaborated). The sample might be rather small but even for companies with several 1000+ employees it was difficult to find workers who work at a special part of the assembly line, predictively for the whole project duration (2 years+) who fulfill requirements (left- / right-handedness, age, robot training,…). Each participant was interviewed for 30 minutes and filled in demographic questionnaires afterwards. The mean age of the study participants was 45.4 (SD=5.7) and they had no prior experience with robotic systems. Four out of five participants had experience with computers and automated systems previous to the studies.

The teaching using robot A yielded requirements regarding robot hand guidance. Gear friction yields stacking and imprecise movement. Locking of certain degrees of freedom (e.g. rotation or translation,…) is asked for by the users as well as semiautomatic tool alignment and expected to improve both programming time and process quality.

A state of the art force torque sensor was integrated (in robot B) as well as buttons to call perpendicular realignment or locking of rotational or translational degrees of freedom. That should make the robots more effective. Additionally a RGB-D sensor as well as a 2D sensor for position deviation correction were added (see Fig. 7). Robot B was evaluated with exactly the same assignment of parameterization of the process points. The teaching duration using remote (robot A) and physical (robot B) control mode was extracted from the video recordings. Table I shows a decrease in average duration by 23.11%, and a strong shift from software- to

manually controlled usage. This shift towards the direct manual guidance of the robot was also measurable in two dimensions of UX: Usability and Performance Expectancy. The first was investigated using the System Usability Scale (SUS). The second describes one's belief that using the system will help him or her to attain gains in job performance, and was measured using two items which were derived from [4]. Table II shows the increase in the dimensions Usability, Learnability and Performance Expectancy.

TABLE I
AVERAGE DURATIONS OF THE TEACHING PROCESS IN STUDY I AND II INCLUDING THE PERCENTAGE OF BOTH CONTROL MODES

Duration (m:s)	Robot A	Robot B
Average Total [SD]	6:25 [2:27]	3:36 [1:03]
Remote Control [%]	6:25 [100.00]	1:01 [28.43]
Physical [%]	0:00 [0.00]	2:35 [71.57]

TABLE III
USER EXPERIENCE IN STUDY I AND II INCLUDING PERFORMANCE EXPECTANCY (PE), SYSTEM USABILITY SCALE (SUS), AND ITS SUBSCALES USABILITY (SUS-U) AND LEARNABILITY (SUS-L)

Duration (m:s)	Robot A	Robot B	Diff. [%]
PE [0-5]	2.40 [1.08]	3.40 [0.89]	1.0 [20.0]
SUS-U [0-4]	2.00 [0.73]	2.53 [0.27]	0.5 [12.5]
SUS-L [0-4]	1.70 [0.76]	2.60 [0.65]	0.9 [22.5]
SUS [0-100]	48.50 [13.99]	63.50 [3.79]	15.0 [15.0]

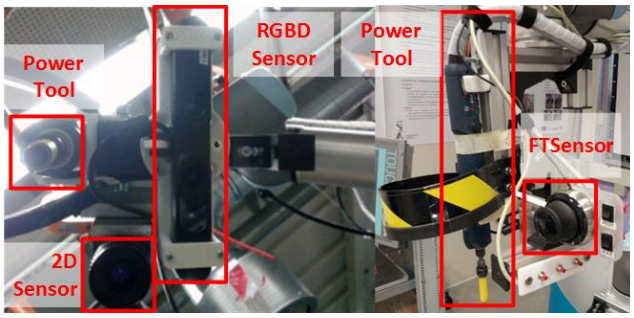


Fig. 7 - Robot B - Tool

During expansion stage 1 experiments (with robot A) the user had to use the Touch Panel 95.4% of the time while manual guidance mode was used only 4.6% of the time. This was due to cumbersome navigation in menus and submenus on the robot teach pendant during the parameterization process. As a consequence a more linear programming approach is proposed for expansion stage 2 which led to the integration of the XROB programming system.

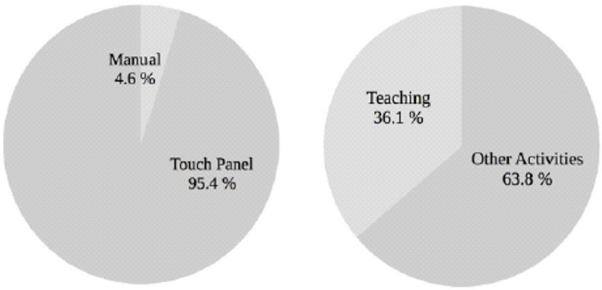


Fig. 8 - programming activities

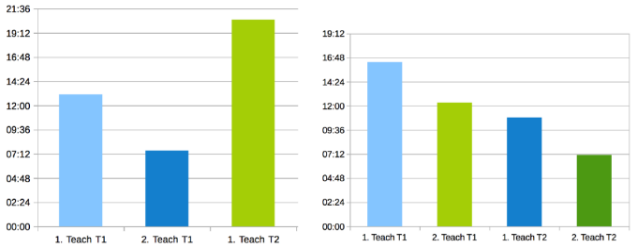


Fig. 9 - programming time with / without additional programming effort for parameterization of machine vision algorithms

Fig. **9** shows the programming time for robot A (T1) and robot B (T2). Total programming time including machine vision increased while training effect and additional input modalities (FT-sensor powered hand guidance,…) yield a net decrease of programming time.

Fig. **10** shows that the small acceptance of the manual guidance input modality in robot A can be increased dramatically if the implementation addresses user requirements and wishes.

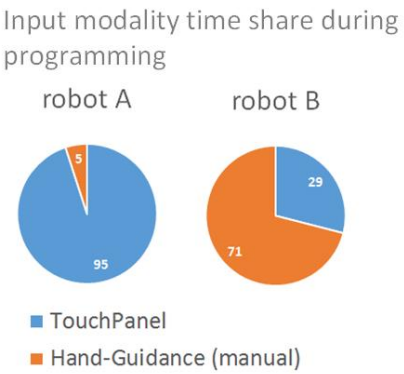


Fig. 10 - preferred input modalities for robot A and B

During the video analysis of robot B the expressed emotions and thoughts of the participants were clustered into several main categories. Qualitative feedback mainly focused on ergonomic details, such as the shape of the handholds on the robot, the positions and drag of the buttons/switches, and the fluency of the manual robot guidance. All of the volunteers pointed out that the robot should actively support them during the teaching process. Main feedback clusters

were interpreted and conclusions drawn. Visual feedback during teach in was requested. If possible, information should be projected to the work piece surface. This would require additional projection technology as proposed by AssistMe.

VI. Proposal for Final Evaluation

An additional projection technology would enable spatial augmented reality methods.

A. Robot C

Spatial augmented reality interfaces are proposed and implemented as tangible user interface. Physical interaction with the product to process only might further minimize programming effort and be an easy to perceive means of interaction. A tangible marble is used for teach in of process points and the sequence of their processing. Therefore a 3D camera is integrated with a projector to detect marbles [21] positioned on top of screws to acquire spatial process points as

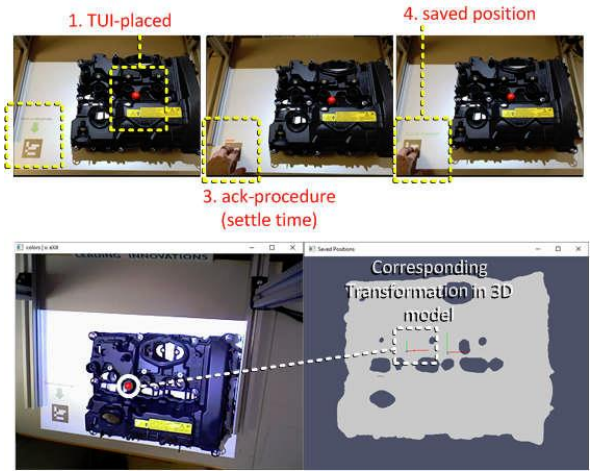


Fig. 11 - Tangible User Interface

well as taps onto projected buttons to confirm their order or other interactions with the programming system.

B. Robot D

Robot D is controlled via a 2D interface as depicted in Fig. 13. Process points are entered by tapping onto a 2D representation of the processed object. A machine vision algorithm determines the spatial region of the tapped point and therefore determines both 3D process points and the sequence of the process points from the tapping order. Fig. 14 shows the technology applied to a bin-picking process where one of several objects in the 3D sensors field of view can be selected in a 2D representation of that data. The same technology is applied to the selection of regions and process-points on a single object in the sensor's field of view. Implicitly also the order of the process points can be entered.

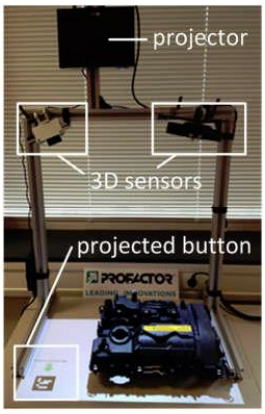


Fig. 12 - Tangible User interface system setup

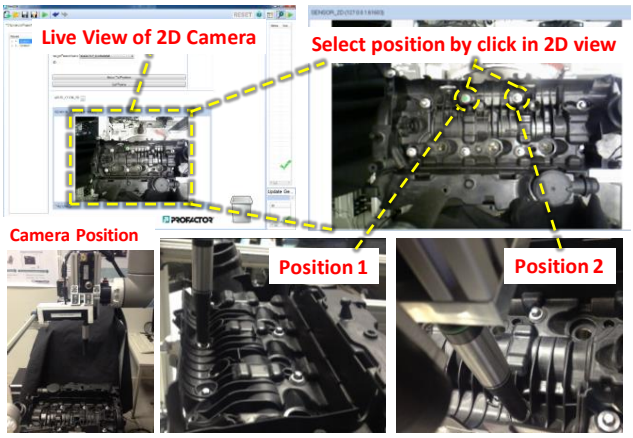


Fig. 13 – Define process points in 2D

C. Robot E

Robot E is programmed by positioning an externally tracked device (Fig. **15**) or an extension like a stick to the process point. Once calibrated a precise position of a stick's tip mounted on an externally position tracked device can be calculated in real-time. Process points and their order are programmed by ordered tipping onto screws in question.

Fig. 14 - 2D tap based process point selection (https://www.youtube.com/watch?v=nrhXEqG014o)

VII. Conclusion

The presented study demonstrated that the not-intermediated (direct manual) interaction with the robot can increase the experience of the robot's capabilities (usability,

performance expectancy).

Fig. 15 – pointer with external tracking

The outcomes of a previous user study [22] led to a technical revision of the HRI mechanisms of the first robot prototype by incorporating the worker's feedback. In the current study the same workers tested the HRI mechanisms of the revised robot and the findings were compared with the previous version. Furthermore, it seems unlikely that the results can be explained by practice effects, due to the period of one year between the studies and the completely different interaction methods. However, the findings of the current study drove the last technical revision of the system (robot C, D, E) which will feature improvements in ergonomics and be evaluated in a final evaluation in 05/06 2017.

Collaboration can be improved by adding visual feedback on the robot and the work piece during the teaching (to reduce the burden of switching attention between the robot and touch panel). [15] [16] introduce the notion Spatial Augmented Reality (SAR) and describe it as enhancement or aggregation of several Augmented Reality (AR) technologies. One formulation [17] might be a depth camera projector based system to project (correctly distorted) information on three dimensional objects instead of flat screens (Figure 3) and may be used for projection of buttons. (Applied) robotics does not make use of SAR methods extensively. [18] introduces a projection based safeguard system for robotic workspaces especially for collaboratively used workspace. [19] gives an overview on Tangible User Interfaces (TUI) which denote interfaces that can be manipulated physically, and which have an equivalent in the digital world and represent a mean for interactive control. The project proposes a combination of TUI and SAR methods. Hand-guided positioning of the robot might be uncomfortable or time consuming due to inappropriate input modalities (friction afflicted robot drives, unintuitive touch screens,…). These were motivations for the implementations of technologies integrated in robot C,D and E and will be evaluated in the final evaluation in AssistMe.

The new HRI mechanisms of robot C, D and E will be based on the paradigm of joint/shared attention, which describes the shared focus of two individuals on an object. Joint/shared attention is realized when one individual alerts another to an object by verbal or non-verbal means such as eye-gazing or pointing (gestures). The application of this paradigm will result in gesture-based HRI mechanisms for robot C. This design decision will shift human-robot interaction towards the dynamics during human-human or human-animal interactions. Therefore, we expect that this approach will help to increase perceived safety, overall acceptance and to ease the transition of working with newly introduced robots.

ACKNOWLEDGMENT

This research is funded by the project AssistMe (FFG, 848653), SIAM (FFG, 849971) and by the European Union in cooperation with the State of Upper Austria within the project "Investition in Wachstum und Beschäftigung" (IWB).

REFERENCES

[1] A. Weiss, R. Buchner, M. Tscheligi und H. Fischer, „Exploring human-robot cooperation possibilities for semiconductor manufacturing," in Collaboration Technologies and Systems (CTS), 2011 International Conference on, 2011.

[2] D. Wurhofer, T. Meneweger, V. Fuchsberger und M. Tscheligi, „Deploying Robots in a Production Environment: A Study on Temporal Transitions of Workers' Experiences," in Human-Computer Interaction--INTERACT 2015, Springer, 2015, pp. 203-220.

[3] R. Buchner, N. Mirnig, A. Weiss und M. Tscheligi, „Evaluating in real life robotic environment: Bringing together research and practice," in RO-MAN, 2012 IEEE, 2012.

[4] S. Griffiths, L. Voss und F. Rohrbein, „Industry-Academia Collaborations in Robotics: Comparing Asia, Europe and North-America," in Robotics and Automation (ICRA), 2014 IEEE International Conference on, 2014.

[5] A. Weiss, R. Bernhaupt und M. Tscheligi, „The USUS evaluation framework for user-centered HRI," New Frontiers in Human--Robot Interaction, Bd. 2, pp. 89-110, 2011.

[6] G. Biggs und B. MacDonald, „A survey of robot programming systems," in Proceedings of the Australasian conference on robotics and automation, 2003.

[7] http://www.kuka-robotics.com/en/products/industrial_robots/sensitiv/lbr_iiwa_7_r800/start.htm.

[8] http://www.mrk-systeme.de/index.html.

[9] https://en.wikipedia.org/wiki/Universal_Robots.

[10] ISO 10218-1:2011 Robots and robotic devices -- Safety requirements for industrial robots -- Part 1: Robots.

[11] ISO 10218-2:2011 Robots and robotic devices -- Safety requirements for industrial robots -- Part 2: Robot systems and integration.

[12] ISO/TS 15066:2016 Robots and robotic devices -- Collaborative robots.

[13] M. Bovenzi, „Health effects of mechanical vibration," G Ital Med Lav Ergon, Bd. 27, Nr. 1, pp. 58-64, 2005.

[14] A. Huber, A. Weiss, J. Minichberger und M. Ikeda, First Application of Robot Teaching in an Existing Industry 4.0-Environment. Does it Really Work? Societies, 2016.

[15] O. Bimber und R. Raskar, Spatial augmented reality: merging real and virtual worlds, CRC Press, 2005.

[16] R. Raskar, G. Welch und H. Fuchs, „Spatially augmented reality," in First IEEE Workshop on Augmented Reality (IWAR'98), 1998.

[17] K. Tsuboi, Y. Oyamada, M. Sugimoto und H. Saito, „3D object surface tracking using partial shape templates trained from a depth camera for spatial augmented reality environments," in Proceedings of the Fourteenth Australasian User Interface Conference-Volume 139, 2013.

[18] C. Vogel, M. Poggendorf, C. Walter und N. Elkmann, „Towards safe physical human-robot collaboration: A projection-based safety system," in Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, 2011.

[19] H. Ishii, Tangible user interfaces, CRC Press, 2007.

[20] C. Harrison, H. Benko und A. D. Wilson, „OmniTouch: wearable multitouch interaction everywhere," in Proceedings of the 24th annual ACM symposium on User interface software and technology, 2011.

[21] Bishop, Durell. "Marble answering machine." Royal College of Art, Interaction Design (1992).

[22] Ebenhofer G., Ikeda M., Huber A., Weiss A., User-centered Assistive Robotics for Production - The AssistMe Project, ÖAGM/ARW 2016, University of Applied Sciences, Wels, Austria

Design of an Autonomous Race Car for the Formula Student Driverless (FSD)

Marcel Zeilinger[1], Raphael Hauk[1], Markus Bader[2] and Alexander Hofmann[3]

***Abstract*— Formula Student Germany is a race car competition for student teams competing with self-designed, self-developed and self-built vehicles. These cars have been competing to win in various disciplines every year since 2006 at the Hockenheimring in Germany. In 2016, a new discipline, Formula Student Driverless, was announced for the following year, targeting autonomous race cars that complete tracks of 5 km over 10 laps as fast as possible without the help of human racer pilots or remote control systems. This paper will cover the framework, the sensor setup and the approaches used by the Viennese racing team TUW Racing.**

I. INTRODUCTION

Formula Student Germany (FSG) is an international construction competition for students with the aim of promoting research in multiple disciplines on a concrete application.

The competition is divided into three classes: electric, combustion and driverless vehicle. The last classification, which has existed since 2017, is called Formula Student Driverless (FSD), where either combustion or electric cars are permitted. Contrary to the other two classes, the driverless class rules permit the use of a team's *existing race car* which has already attended previous FSG race(s). Therefore, formerly manually-driven race cars equipped with appropriate sensors, actors and computing hardware can be adapted to driverless cars. Nevertheless, vehicles have to comply with the restrictions of their appropriate class; in particular, manual operation must still be possible.

Regardless of the class in which a team participates, the event itself is divided into static and dynamic disciplines, and points can be scored in each discipline. In the static discipline, a cost report and a business plan must be presented by each team. Before a car is allowed to participate in a dynamic event, a technical inspection must be passed to ensure the car is mechanically and electrically safe, in accordance with the rules [3]. Since a car's total score is comprised of both disciplines, the fastest car need not necessarily win.

In the following sections TUW Racing will describe its concept for participation in the FSD 2017. First, an overview is provided of the solutions and academic approaches available in the area of autonomous driving, with regard to similar solutions in general mobile robotics. Then an overview of the vehicle and the sensors, actuators, as well as the software components used is provided. Furthermore, we will describe our approach to the problems posed by the competition and reflect upon its effectiveness up to this point.

Fig. 1. Racecar Edge8 from the previous season (2016) which has been redesigned to drive autonomously

II. RELATED WORK

In order to describe related work, time was invested in studying competitions with a similar focus. In addition, studies were conducted on commercially available solutions to autonomous driving such as the NVIDIA DRIVE[1] products.

The Carolo Cup[2] is a competition where student teams build autonomous 1:10 sized cars, which have to cope with reality-inspired driving situations faced by passenger cars [12]. While there is a static competition in the form of a concept presentation as well, the dynamic events consist of parking and driving a free drive with and without obstacles. During the free drive, the model car needs to master challenges like intersections, speed limits or dynamic obstacles such as other vehicles or passengers crossing the track, which is marked with white lines on a dark ground similar to real road surface markings. In the paper [12], *Technical Evaluation of the Carolo-Cup 2014*, the authors state under *Lessons Learned* the importance of high quality hardware products and the need for a robust computer vision algorithm in detecting road markings. Similar to this competition are the NXP Cup student competition, also known as the Freescale

[1]Marcel Zeilinger and Raphael Hauk are with TUW-Racing, Adolf-Blamauergasse 1-3, 1030 Wien, Austria firstname.lastname@racing.tuwien.ac.at

[2]Markus Bader is with the Automation Systems Group at Institute of Computer Aided Automation, TU Wien, Karlsplatz 13, 1040 Vienna, Austria firstname.lastname@tuwien.ac.at

[3]Alexander Hofmann is with the Institute of Computer Science at the University of Applied Sciences FH Technikum Wien, Höchstädtplatz 6, 1200 Vienna, Austria firstname.lastname@technikum-wien.at

[1]NVIDIA DRIVE http://www.nvidia.com/object/drive-px.html

[2]Carolo Cup: https://wiki.ifr.ing.tu-bs.de/carolocup/

Cup[3], and the Crazy Car[4] race, which has been hosted by FH Joanneum in Austria since 2008, where students from schools and universities are eligible for participation.

Other relevant competitions comparable to the FSD are the DARPA Challanges of 2004 to 2007. The car Stanley [9] won this competition in 2005, but since that time many products in the field of autonomous driving have entered the market.

Autonomous control is an important new subfield in the automotive sector as commercial autopilots and driver assistance systems become more and more popular.

To classify the different levels of system functionality and compare actual driverless cars, the Society of Automotive Engineers has introduced a classification scheme[5].

Many commercially available software products fall into classes 0-2, thus only observing driver environment and taking on only limited driving tasks. The rare cases of actual automated driving systems in classes 3-5 can be found in Google Autonomous Cars or Tesla Autopilots. Google makes heavy use of a multi-beam 3D-laser scanner to understand the entire environment. It also solves problems related to the traffic behaviour of human drivers and it is integrated into external maps and weather services. Tesla uses radar and sonar, in some cases mounted behind the vehicle's outer structure, to detect individual classes of obstacles, e.g. with radar: moving obstacles; or, with ultrasonic sensors: other cars beside the vehicle. [1] [5]

Nvidia presented an autonomous control solution where the system was taught to drive exclusively by RGB camera input [2]. Machine Learning was used to match steering input with camera images, so the vehicle steers based on scenes recognised in the camera view. Nevertheless, the driver has to control the throttle.

An example of a framework for autonomous driving is provided by Nvidia DriveWorks. It provides interface layers that allow easy incorporation of new sensors, execute machine vision algorithms and even perform trajectory planning. Advantages are its range of out-of-the-box detection algorithms and the heavy use of graphics-accelerating hardware with perception computation times of within a few milliseconds. An apparent disadvantage is that only Nvidia hardware can be used. Another drawback is that the architecture lacks an actuator control component.

The approaches and frameworks in Google's, Tesla's and Nvidia's applications solve a variety of problems associated with driverless cars in traffic that are not relevant to the FSD competition. Accidents, recognising people, lane markings, sidewalks or traffic signs do not have to be handled at all in the FSD. There are no intersections where maneuvers have to be negotiated with other cars; for that fact, there no other cars on the track at all. This year, TUW Racing's main goal is to design a solid base system as an ideal start for improvements and qualitative increments in upcoming years.

[3]Freescale Cup: https://community.nxp.com/docs/DOC-1011

[4]Crazy Car: https://fh-joanneum.at/projekt/crazycar/

[5] SAEReport:https://www.sae.org/misc/pdfs/automated_driving.pdf

Therefore our solution is based on open-source components. Due to the fact that TUW Racing initiated the project with experts in mobile robotics, the commonly used Robot Operating System ROS[7] is used for modularisation and communication.

ROS enables TUW Racing to interface the required components of the system, from camera to motor control. Therefore, for speedy development, the team selected hardware devices with drivers already available.

In the next section, the hardware and the software frameworks used will be described.

III. The Race Car

In this section, the competition, the race car's hardware, as well as the software implemented are described in detail.

A. Competition

The dynamic component of the race has three challenges: an acceleration race, a skid pad and a track drive. The acceleration race is a 75m-long straight track followed by a 100m straight exit lane. The skid pad track consists of two congruent circles, touching externally, with a diameter of 18.25m. The vehicle enters on the tangent of the circles at their contact point, drives the right circle twice followed by the left twice before leaving via the tangent again. The track drive is a closed loop circuit with an unknown layout consisting of up to 80 m straights, up to 50 m diameter constant turns, and hairpin turns with a 9 m diameter, among other miscellaneous features such as chicanes, or multiple turns. The vehicle must recognise and drive exactly ten laps with a maximum distance of 500 m each. The track is marked by blue and white striped traffic cones on the car's left edge and yellow and black striped cones on its right edge. The cones are connected via a high-contrast colored line sprayed onto the road; however, the line may extend out to the side. Additionally, the stop zones on the acceleration and skid pads are marked using orange and white striped cones. The distance from one cone to another along either edge of the track is up to 5 m, and the minimum track width is 5 m, except on the skid pad, where the width is 3 m. The traffic cone layout is predefined in the rules.

B. Hardware

The total power allowed for formula student cars with any kind of powertrain is 80kW, and for electric powertrains a maximum voltage of 600V at any time is allowed. The other major limitation is the wheelbase of at least 1525mm. Aside from those stipulations, students are free to design their cars' characteristics as they wish.

Both rear wheel hub motors of the TUW Racing car have a maximum torque of 30 Nm at a weight of 3.6kg and a gear transmission ratio of 12:1. The total weight of the car is 160.5kg with a top speed of 30.5m/s, accelerating in 3.1 seconds to 27.8m/s. Fig. 2 shows a rough sketch of the vehicle with sensors.

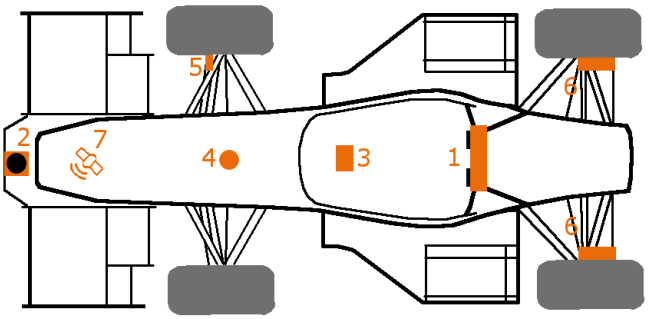


Fig. 2. Sensor placement on the vehicle: 1 stereo camera, 2 laser scanner, 3 IMU, 4 steering angle encoder, 5 wheel speed encoder, 6 rotor position encoder, 7 GPS

1) Actuators: TUW Racing had to adapt its vehicle with actuators that still allow for human operation of the vehicle. The brake pedal may not be blocked and the steering must be easily steerable by hand despite the gearboxes and motors that are to be added.

- **Brake System:** Due to the already-optimised brake balance, the brake system is mechanically operated via the pedal. Details in the mounting allow the driver to still press the pedal. An alternative way to decelerate is through reverse operation of the motors, although brake response is faster and conserves energy.
  The brake system must always be able to stop the car within a maximum of 10 m, even in the face of a single failure in the system, including power loss or any mechanical failure.
- **Steering:** The additional steering motor is mounted to the existing steering strut at the top of the monocoque. With an average-size driver in the vehicle, the car weighs about 240 kg and requires 25 Nm to steer while the car is not moving. This is the force TUW Racing designed for, since it would enable testing with a driver.

The competition requires students to design emergency systems in a detailed Failure Mode and Effects Analysis (FMEA). For instance, power or mechanical failure to the brakes or steering must be accounted for with fallback systems. When emergency braking is initiated by remote or failure detection in another subsystem, the vehicle must enter a safe state that simultaneously relies on the actuator's operation. For instance, a vehicle steering 60 degrees to the left while a full brake is initiated should first steer to the center position to optimise friction on the wheels.

2) Sensors: Sensor placement on the vehicle is shown in Fig. 2.

- **Camera:** TUW Racing selected a ZED[6] stereo camera for its visual sensors. It features an opening angle of 90deg and a base line of 120mm, connected via USB3, which enables generation of depth images at a range of 20m. The camera is mounted on the topmost point of the roll bar (1). In order to automate the calibration process and to estimate the extrinsic camera matrix, the team used visual markers attached to the vehicle's chassis on specific locations, as shown in Fig. 3

[6]ZED Stereo Camera: https://www.stereolabs.com/

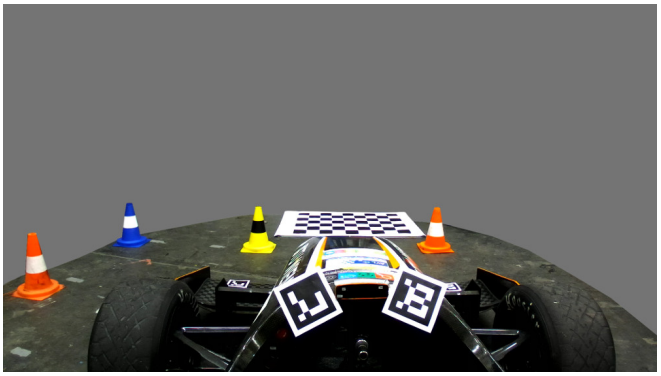


Fig. 3. Image of left camera with visual markers attached to vehicle to estimate extrinsic camera matrix

- **Laser Scanner:** A laser scanner was placed inside the front wing, at the lowest point, as it is a planar scanner and the cones are relatively short. It is tilted downwards, so that it points at the bottom of a cone from the maximum distance. This counteracts damping influence, which could lead to the rising of laser orientation during accelerations, and the lowering thereof during braking. With the use of (2) a Hokuyo 20LX planar laser scanner, the autonomous system is able to detect cones within the field of view using circle detection and heuristics to filter non-cone circular obstacles, e.g. wall detection.
- **GPS:** For accurate absolute position measurement, a dGPS, provided by a Piksi Multi GNSS module (7), is used along with two beacons placed outside the race-track. The beacons allow for more precise positioning than a common GPS system does.
- **IMU:** The relative movement of the vehicle is measured by a motorsport-grade IMU (3), which measures rotation in the yaw axis as well as acceleration in the x and y directions.
- **Odometry:** To accurately determine the front wheels' speed and distance travelled, TUW Racing uses an inductive wheel spin sensor (5) at each front wheel, and for the rear wheels a rotor position encoder is used for the car's TUW-Racing-developed motors(6).
  The steering angle is measured by a rotary position encoder (4) connected to the steering shaft. All measurements acquired from these devices are used within the software framework which is described in the following section.

C. Software

In order to integrate and process sensor measurements, ROS is used as the base framework and nodes were implemented for the following tasks.

- **SLAM (Simulations Localization and Mapping):** The vehicle's pose must be estimated and the race track must be reconstructed while driving[8].
- **Machine vision:** The traffic cones, their positions, and colors, must be detected using multiple cameras and a laser range sensor.

- **Path extraction:** A path must be computed which guides the vehicle between the traffic cones to the destination and marks the drivable area.
- **Motion control:** The motion controller uses the drivable area detected between the cones, computes an optimal trajectory, and dispatches the actual motion commands for steering and accelerating the car.

In Fig. 4 one can see an overview of the messaging system among the ROS components. It shows the connection of the actuators to the sensors, as well as the processing steps needed to achieve desired control sequences. Some of them are implemented as nodelets to minimise messaging delays.

In the following section techniques used are described in more detail.

IV. Approach

The EKF-SLAM implementation used is based on Macsek's [6] Master's thesis. The filter uses perception of cone measurements with uncertainties and generates a map of the cones in 2D. It also determines the car's own position on the map, computes its position relative to previous laps, and performs motion control. It is necessary to predict the vehicle's pose ahead of time in order to send proper motion commands, since actuators have a considerable reaction delay.

For sensing, it is vital to detect traffic cones in the environment and estimate their location relative to the sensors. The known location of the sensors on the vehicle enables one to create a global map of landmarks for the trajectory-planning algorithm.

In the first development phase, TUW Racing used Pioneer 3AT in a simulated and real world scenario in order to generate measurements, try out sensors and plan algorithms, as shown in Fig. 5.

A. Traffic cone detection

Because of the variety of sensors used (lasers and cameras) various techniques have to be implemented accordingly. Two approaches are used to perform cone detection and position estimation on the stereo camera images. In the first approach, a depth image based on a block-matching algorithm is used. This allows for extraction and removal of planes in a 3D data set. The track floor that is geometrically known can be pre-segmented, and cone candidates appear as isolated objects. These candidates are then further evaluated using classical image processing steps on the estimated location in the image plane, similar to the approach of Yong and Jianru [11]. However, a block matching algorithm cannot be easily adapted to use the known structure of the environment for improved perception of target objects. Moreover, the computation effort and thus hardware requirements would be unnecessarily high, since one would have to operate on an entire image.

In the second approach, cones can be detected in both images separately. The algorithm computes the disparity, the z-depth and consecutively the 3D position of the cone, based only on the UV center points of the same object determined in the left and right images.

In both approaches, the program determines the colors heuristically with RGB color thresholds. If the estimated center point is within a white or a black stripe, the color sample point is moved downwards until another color match is determined.

In addition, the planar laser scan is searched for point clusters of distances that match the radius of the cone at the height scanned. If the total distance spanned by coherent scan points is above this radius but below the total radius of the cone base, the algorithm considers the center point between the cluster to be the center point of a cone. In order to avoid detection of obstacles other than cones, the detector filters long connected components such as walls.

The detection results of both algorithms are then fused with a feature-based EKF-SLAM [6]. Thus the team benefits from the advantages of both of the detection approaches, because the fusion algorithm respects the reliability of each detection result and takes the more reliable detection result into greater account. For example, distance estimation at high distances is less accurate at image-based detection, due to the fact that pixel disparity is smaller. On the other hand, laser-based detection cannot reliably detect cones that are behind other cones.

B. Mapping

For TUW Racing, it is essential that the mapping components be able handle failures in the detection component. The camera or the laser could have short periods of blindness due to sunlight or uneven road conditions. In these cases a cone might be missed. Since cones are guaranteed to be at most 5 m apart, missing ones can be filled in. Because all cones on the left side have a different color than those on the right, heuristics can fill in cones missing from one side based on the cones observed on the other side.

Since the traffic cones themselves are the best reference for position and orientation on the track, the team uses an EKF-SLAM that maps them and corrects the vehicle's position based on their perceived change of position. An additional challenge here is that the same track sections are driven ten times throughout the competition. Thus, the mapper must recognize previously mapped cones by ID, even though they are of the same color and shape. An index counter from the starting line is insufficient due to potential perception failure. Nor does the vehicle have a full 360deg line of sight. In a narrow left turn, due to the vehicle's pose in an optimal trajectory, it is possible that only the outer cones might be seen. If the mapping algorithm performs corrections to positions on the map on those right lane cones, the left lane cones must also be updated automatically.

C. Trajectory Planning

From the mapping one can derive a path which lies in the center of the travel area; path-planning is therefore obsolete. To start, a normal vector is projected from the center point between the first left and right cones forward. The first cones

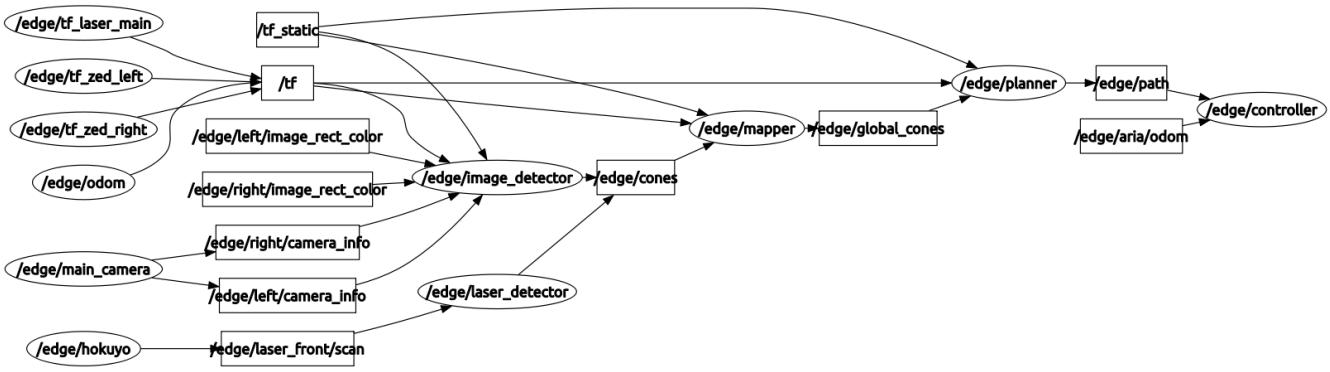


Fig. 4. ROS nodes and message subscriptions in rqt_graph

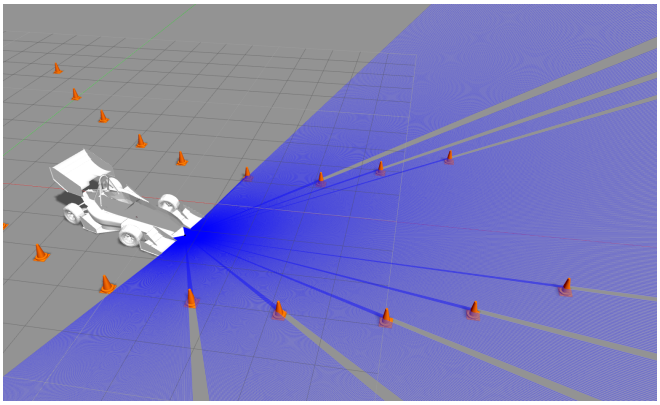


Fig. 5. Edge8 in a simulated environment using GazeboSim.

Fig. 6. Cone detection on still of last year's race recorded using a GoPro camera

to the left and right of this vector are identified, then the center point identified, and a vector projected forward again. Since the left and right sides can be distinguished, a situation where the cut to the left of the vector which contains a right-side cone, or vice-versa, is manageable. The vector is then rotated accordingly so that the sides are split correctly.

But following the center path would not be very efficient, so the model predictive [4] motion controller computes its own optimal local trajectory. The MPC [10] used optimises the trajectory using the latest perception state and incorporates the mechanical constraints of the vehicle's motion model, such as maximum linear acceleration and deceleration, actuator delays and disturbances resulting from vehicle dynamics, e. g . sideslip angle.

V. Results

In order to build a foundation for simulation of the race car in software that integrates well into ROS, TUW Racing extended its approach to the Pioneer 3AT and used gazebo simulations to test marker mapping, odometry corrections based on simulated IMU data, trajectory planning, as well as a basic self-developed motion controller. Fig. 5 shows the robot following a narrow turn in the simulation, with obstacles of width and height comparable to those of cones on the track. It successfully navigates a 180deg turn, detects shapes based on simulated laser scans and determines the center path accurately.

In Fig. 6 results with a cone detection algorithm on a still from a video of last year's autocross race at the FSG are shown. The cones detected in the image are enclosed by yellow rectangles. The algorithm is based on a trained cascade classifier using a histogram by an oriented gradients feature descriptor. Photos of the specific cones seen in the image, from angles of 0deg horizontal to about 40deg turned downwards, served as training data. The algorithm turned out to be computationally expensive and susceptible to variations in lighting and weather conditions. With the trained algorithm on last year's video we achieved an unstable detection rate of 30% to 90% depending on the track section. The algorithms explained above, evaluated on recordings with the stereo camera, detected 75% of the traffic cones in the image regardless of lighting. Most of the missed cones were affected by very strong shadows cast by large nearby obstacles, wear of the cone itself or cones that partially covered each other. The location, color and shape of the cones, the point of view, background, the vehicle itself and lighting are very similar to the scene in the image.

Evaluation of detection on a recording of last year's race has shown that different light conditions can be a problem for the car's visual sensor. Extreme light exposure to the camera or even driving from shade into sunlight can be disturbing, as the camera needs some time to adjust to the brightness. Similar issues emerged in evaluation of the stereo camera mounted on the car driving a short sample track. In Fig. 7, the camera adjusts to increasing brightness using integrated exposure control when the car enters a sunlight-drenched part of the track. As the more distant cones become visible by the aperture's adjustment, the cone marked with the orange

Fig. 7. Images of the ZED stereo camera showing how light conditions affect the visual cone appearance.

circle gets darker and less visible.

The Lidar suffers from similar issues, as can be seen in Fig. 8. The cone detection algorithm was tested in an outdoor environment during rain. Since dealing with environmental conditions is part of the challenge of Formula Student, the vehicle will have to run regardless of the current weather. The small axis markers show laser scan points that hit a cone in the vicinity of the robot. The small red dots represent points relative to the robot where an obstacle was detected. The white visual markers on the field show the cones the robot itself is currently detecting; the numbered axis markers show the cones mapped. Usually the field would be empty at the sensor's maximum range, but during rain, reflections are measured at some point, resulting in a noisy outer cloud structure. Near the robot some of the rays are instantly reflected and reduce the amount of rays that hit a cone, thus making accurate detection more difficult. Fig. 8 shows that the algorithm used is able to map cones with decreasing accuracy but at higher distances. Since scan data has to be compared over time until a cone can be classified correctly, the speed of the detection algorithm is also adversely affected.

One can expect the effect of small rocks on the track to have similar effects on detection, as at high speeds they are usually sprayed across the track by the vehicle.

VI. Conclusion

In this paper the TUW Racing team's approach to the FSD 2017 competition was presented. Details were provided on its hardware and the approach to its control software.

The team will begin testing within the next month on a test site with road conditions that resemble the event location and a setup with obstacles corresponding to those detailed in the rules. Initially the team will validate the accuracy of the motion model and perception system in separate road sections. Then it will complete entire tracks at low speeds before gradually increasing speeds while improving controller parameters.

While TUW Racing's main goal this year was successful participation in the first Formula Student Driverless event, its goal for further seasons will extend to improving the efficiency of the vision algorithms and accuracy of the underlying motion model. The choice of the sensors, actuators and computers as well as the software framework will be influenced by performance in this year's competition.

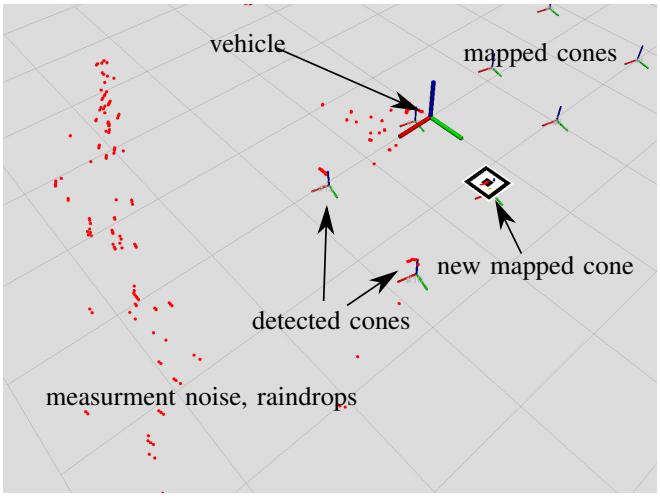


Fig. 8. ROS Rviz view of an outdoor test with the pioneer during rain. The laser scan (red) is disturbed by the raindrops.

Acknowledgment

The project is supported by TTTech Automotive GmbH.

References

[1] "How Google's Self-Driving car works," Accessed: 14-March-2017. [Online]. Available: http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works

[2] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016. [Online]. Available: http://arxiv.org/abs/1604.07316

[3] Formula Student Germany e.V. Formula Student Rules. Accessed: 14-March-2017. [Online]. Available: https://www.formulastudent.de/fsg/rules/

[4] T. Howard, M. Pivtoraiko, R. Knepper, and A. Kelly, "Model-Predictive Motion Planning: Several Key Developments for Autonomous Mobile Robots," *Robotics Automation Magazine, IEEE*, vol. 21, no. 1, pp. 64–73, March 2014.

[5] S. Ingle and M. Phute, "Tesla Autopilot : Semi Autonomous Driving, an Uptick for Future Autonomy," vol. 3, no. 9, Sep 2016.

[6] M. Macsek, "Mobile Robotics: EKF-SLAM using Visual Markers for Vehicle Pose Estimation," Master's thesis, Vienna University of Technology, 2016.

[7] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[9] S. Thrun and et al., "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006. [Online]. Available: http://dx.doi.org/10.1002/rob.20147

[10] G. Todoran and M. Bader, "Expressive navigation and Local Path-Planning of Independent Steering Autonomous Systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4742–4749.

[11] H. Yong and X. Jianru, "Real-time traffic cone detection for autonomous vehicle," in *2015 34th Chinese Control Conference (CCC)*, July 2015, pp. 3718–3722.

[12] S. Zug, C. Steup, J. B. Scholle, C. Berger, O. Landsiedel, F. Schuldt, J. Rieken, R. Matthaei, and T. Form, "Technical evaluation of the Carolo-Cup 2014 - A competition for self-driving miniature cars," in *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*, Oct 2014, pp. 100–105.

Concept and Implementation of a Tele-operated Robot for ELROB 2016*

Florian Fuchslocher[1,2], Martin Rambausek[1], Wilfried Kubinger[1] and Bernhard Peschak[2]

***Abstract*— The current use of mobile robots in search and rescue scenarios like natural or man-made disasters is often required to protect emergency response personnel from dangerous situations and support them in their work. At the same time the localisation and rescue of victims has to be achieved fast and reliable. However, a fully autonomously controlled robot is highly sophisticated and needs many sensor components whose data has to be combined. Due to this fact, mostly a combination of a tele-operating system and a fully autonomous system is implemented.**

This paper focuses on developing a concept for a taurob robot for participating in the European Land Robot Trials 2016 (ELROB). The aim is to integrate suitable sensors into the robot system and to implement a tele-operating mode. The selection of the sensors is based on criteria's of the competition's scenarios. For the implementation of the tele-operating mode, the Robot Operating System (ROS) is used. Two variants, a keyboard and an Xbox Controller, are tested to steer the robot.

The obtained results show that operating the robot by the Xbox controller is easier and more precise than by the keyboard. Combined with the sensors, the system shows an overall solid performance and provides a good basis for further development.

I. INTRODUCTION

Disaster control and its dangers are a big topic, that can be covered by robots to protect rescuers from hazardous environments [5]. The European Land Robot Trial is a convention for showing the abilities of different unmanned systems in realistic scenarios [2]. The aims are headed towards the greatest possible autonomy and strong performance. To participate at ELROB 2016 the servicerobot Robbie was designed to fulfill the requirements of three different scenarios [3]. This robotic system is based on an Austrian robot platform from the company taurob [1] and was developed to compete in the challenges of three scenarios. Reconnoitring of structures (e.g., mapping), search and rescue (find and drag a dummy body) and Reconnaissance and disposal of bombs and explosive devices (EOD/IED). To achieve results in these challenges, several sensors and a controlling system were integrated to the robot. In the following, these systems are specified and their performance is discussed.

*This project has been partly funded by MA23 - City of Vienna within the Project Call 16-02 "Photonics: Foundations and industrial applications".

[1]Florian Fuchslocher, Martin Rambausek and Wilfried Kubinger are with University of Applied Sciences Technikum Wien, Hoechstaedtplatz 6, 1200 Vienna, Austria, email: wilfried.kubinger@technikum-wien.at

[2]Florian Fuchslocher and Bernhard Peschak are with the Austrian Armed Forces, Rossauer Laende 1, 1090 Vienna, Austria, email: bernhard.peschak@bmlvs.gv.at

II. SYSTEM OVERVIEW

The system consists of the robot vehicle and its selected sensor components.

A. Robot Vehicle

The basis of this project is a Taurob robot [1]. It is a rugged mobile service robot, which is driven differentially and has capabilities to handle rough terrain. The robot structures itself into the base, the wheels and the wheeled driven rubber tracks. For a better climbing performance, it is also possible to adjust the latter ones in height by the robot's driving motors. This function makes it able to climb slopes and stairs up to 45 degrees and obstacles up to 35cm in height. The robot is waterproof and designed for harsh environments, too. Interaction with its environment is achieved by the robot arm, which has the strength to pull a body with 20kg in weight. Additionally, the robot is equipped with Ethernet ports to allow easy integration of various hardware components like the robot arm and sensors.

Moreover, the upper side of the robot base includes a voltage supply socket. It provides two voltage levels, one stabilized 12V (max. 4A) and a 24V (max. 5A) battery voltage. These two voltage levels make it possible to power all used hardware components. The robot's integrated WLAN router achieves a wireless communication between the robot and a laptop. Fig. 1 shows an example of the Taurob robot with all integrated sensor components.

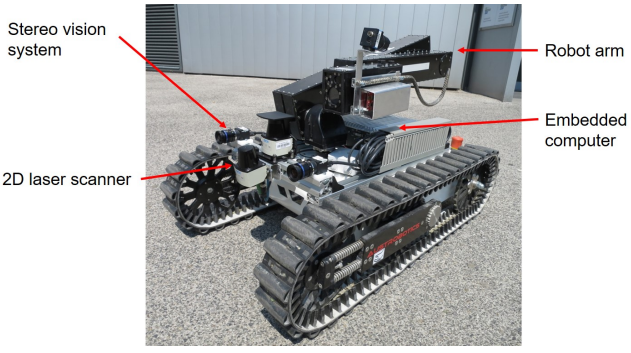


Fig. 1. Sensor setup of Robbie

B. Robot Periphery

For participating in all ELROB scenarios, a stereo vision system, several Cameras, two laser scanners, a robot arm including a nuclear sensor and a hook, a GPS module and an embedded computer are integrated to the robot's periphery. All hardware components are essential for the tele-operator

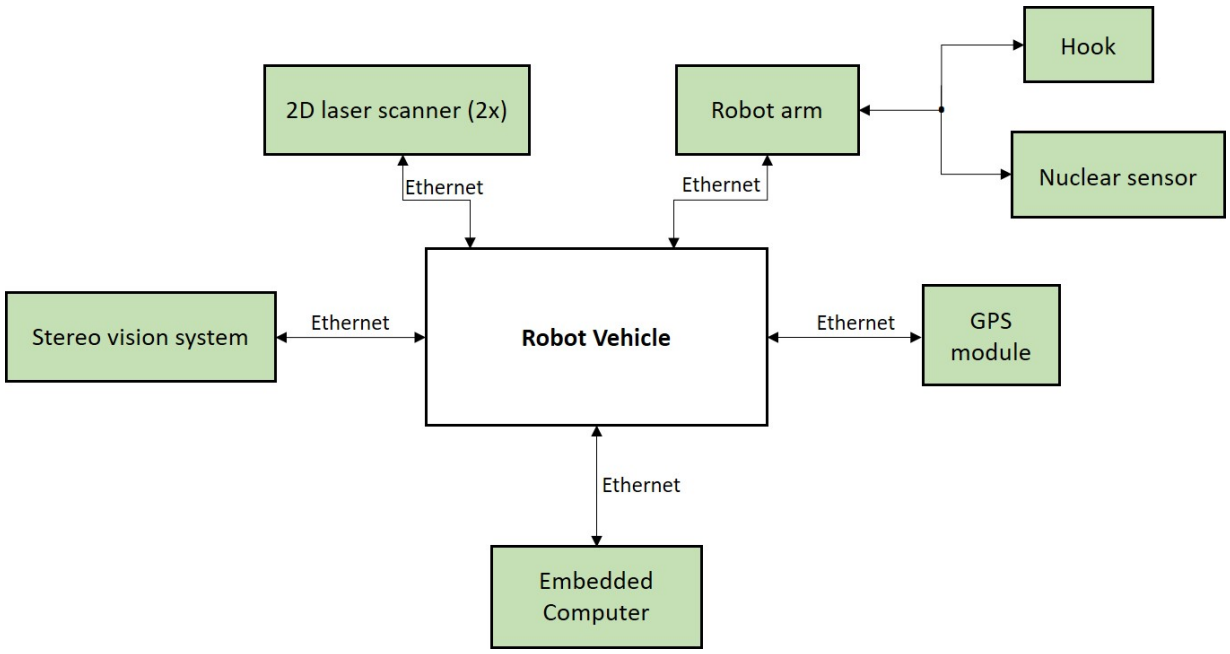


Fig. 2. Connections between robot and hardware components

to operate the robot through rough terrain, detect objects of interests and interact with them. The following block diagram (Fig. 2) shows connections between sensor components and the robot.

The single cameras of the Robot are located on different points, giving the operator the possibility to obtain a broad view around the robot. Two cameras on the front and rear are placed in the body between the tracks, each giving a view in the needed driving direction. Another camera is positioned either on the left or right side of the front, giving the possibility to reconfigure and choose the needed side view before operating manually. A last camera is mounted to the last joint of the robot's arm providing a view from the hooks position (e.g. view top-bottom, to see the tracks and the driveway in overview).

Two laser scanners placed in the front are needed for generating maps and position in the operation via SLAM-algorithms. The nuclear sensor and the GPS module are applied for positioning and obtaining radiation heat maps.

The stereo vision system is installed on the front of the robot. Its purpose is dedicated to future development of 3D map generation and autonomous operation.

III. TELE-OPERATING MODE

To control the robot manually, a tele-operating mode is implemented using ROS. One criteria is to provide two variants for the operator, which can be chosen later. These two steering variants are:

- Keyboard controlled steering
- Steering by Xbox controller

For realisation of both steering variants various ROS packages were integrated in the software environment.

TABLE I
KEYBOARD BUTTON CONFIGURATION

Button	Function
u	Left rubber track forwards (right turn)
i	Go forward
o	Right rubber track forward (left turn)
j	Left rotation
k	Current command stop
l	Right rotation
m	Left rubber track backwards (right turn)
,	Go backward
.	Right rubber track backwards (left turn)

A. Keyboard control

First approach is to control the robot by a keyboard. To realise this variant, the teleop_twist_keyboard package [4] was integrated into the ROS environment. This allows the operator to control the vehicle with the input buttons shown in Table I.

To quit operating the robot by the keyboard, CTRL-C has to be entered. After that the robot stays in safety state and cannot be controlled by keyboard as long as the teleop_twist_keyboard package is restarted.

B. Xbox controller

Alternatively, ROS also provides controlling the robot by Xbox controller [7]. Therefore, the joy package [6] is used for implementing the controller into the ROS environment. After implementation, every steering command of the robot can be used to configure it to one of the Xbox controllers or joysticks buttons. In Fig. 4 the wired Xbox controller and its buttons are demonstrated.

Since there are only a few of control commands, not all

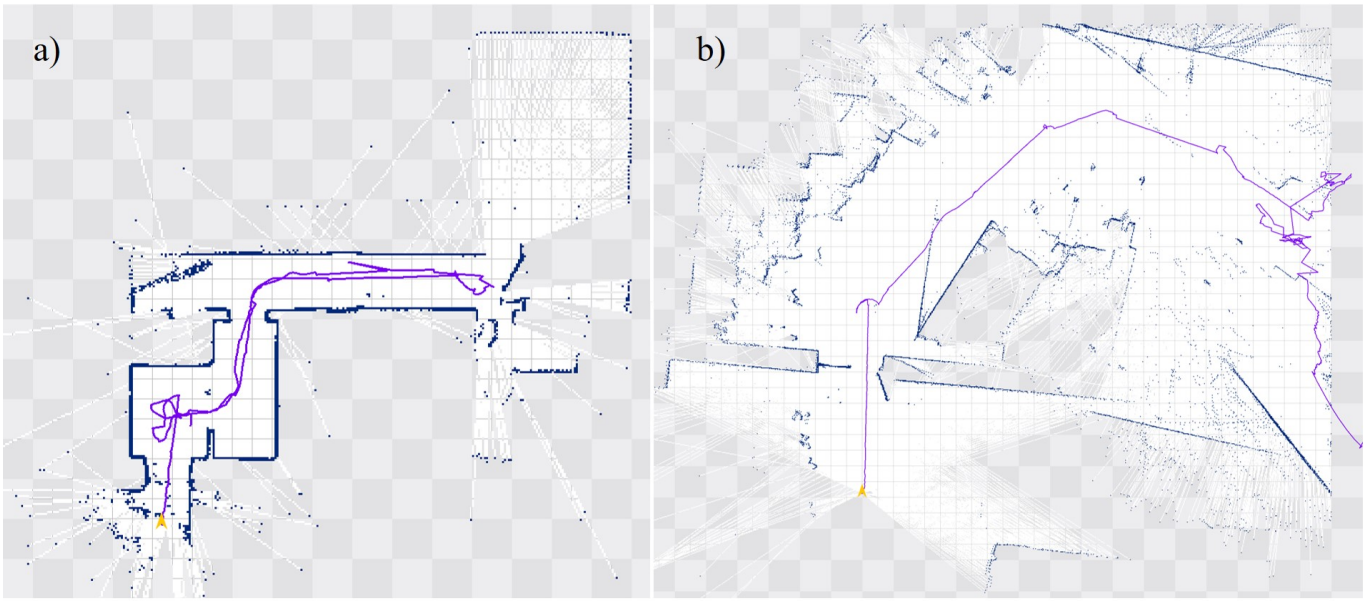


Fig. 3. Mapping results a) small indoor rooms b) disrupted map in a big open area at ELROB 2016

Fig. 4. Xbox Controller for operating the robot

buttons are used for steering the robot. The following table II shows which of them are implemented for which function.

TABLE II

XBOX BUTTON CONFIGURATION

Button	Function
Left joystick (up/down)	Drive (forward/backward)
Left joystick (left/right)	Rotate (left/right)
Right joystick (up/down)	Height-adjustment rubber tracks (up/down)
Right bumper	Release button
Left bumper	Turbo button

Due to safety reasons the enable button have to be pressed all time for steering the robot by the joystick. Speed can be controlled depending on how strong the stick gets pushed. That means if the joystick gets only half pushed, the robot will also drive with half the speed. Alternatively, the left bumper can be used for enable the robot's full speed mode. In this mode, Robbie drives with around $5km/h$ full speed. As mentioned before, it is possible to overcome obstacles by adjusting the robot's rubber tracks up or down. This is done by moving the right joystick up or down respectively. Safety state is achieved when all buttons are not pressed.

IV. RESULTS AND DISCUSSION

Compared to the keyboard variant, the Xbox controller is the better decision for operating the robot in all ELROB challenges. The results of both variants compared are shown in Table III. While the Xbox controller obtains mostly excellent results, the keyboard is improvable in most criteria. Since all buttons of the keyboard are closely packed together, steering the robot becomes more complex. Moreover delay times of the sent commands were detected during tests with the keyboard, which also effects sensitivity and accuracy of steering the robot. Additionally, increasing speed by the keyboard needs to push and hold one "drive-button" and push the "speed-up-button" at the same time. In contrast, the Xbox controller varies speed by inclining the joystick, which results to more sensitivity and accuracy.

TABLE III

EVALUATION OF BOTH STEERING VARIANTS (+ ... EXCELLENT, ∼ ... AVERAGE, • ... BAD)

Evaluation Criteria	Keyboard	Xbox Controller
Handling	∼	+
Accuracy	•	+
Sensitivity	•	+
Flexibility	∼	∼
Feasability	+	+
Safety functions	+	+

Mainly, there are several aspects coming out of this development. The tele-operated operation was a great success at the ELROB event. While the video stream is slightly delayed due to transfer limitation, the analogue joysticks are

able to compensate that with nearly stepless motion, giving the option not to lose time by having to stop and wait for movement transmission on the screen. The map presented in Fig. 3 was generated in two different areas. The left part (a) shows a testrun inside a building with small rooms. The right part of the picture (b) shows the generated map of the reconnoitring of building structures challenge in big rooms during the ELROB competition. In (b) it can be seen that the system tends to lose orientation and fails by generating a solid map of the area.

V. SUMMARY AND OUTLOOK

The aim of this paper was to develop a concept for the taurob robot to participate at the ELROB 2016. One part was to integrate two Pointgrey cameras for a stereo vision system, two Sick 2D laser scanners, one Garmin GPS module and one embedded computer into the robot system. The selection of these sensors was based on the criteria of the ELROB scenarios. Additionally, a tele-operating mode was implemented by the open source program ROS for steering the robot by a keyboard or an Xbox controller. To check reliability of the robot system for the ELROB, trials in specific test scenarios were carried out. The obtained results of the test scenarios show that operating the robot by the Xbox controller is easier and more precise than by the keyboard. Furthermore, it is possible to build a 2D map of indoor areas by the 2D laser scanner. Moreover, a dummy body with a length of 1,80m and a weight of 20kg can be dragged by the robot's arm. Although there is still some potential for improvement in different fields, the system achieved a 3rd rank in the Reconnoitring of structures part of competition.

The next step will be the upgrade to a more autonomous state of operation including navigation and full support of visual data processing as well as 3D Mapping including radiation and visual integration of Points of interests (POIs). Also, an advanced controlling system for the arm is planned. The project is in further development and will be featured on schedule for EnRicH 2017 in Austria and the next ELROB 2018 in Riga.

REFERENCES

[1] austrobotics.com. (2016) austrobotics: robots for research and education. austrobotics. [Online]. Available: http://austrobotics.com

[2] ELROB2016. (2016) Elrob - the european land robot trial. ELROB 2016. [Online]. Available: http://www.elrob.org/elrob-2016

[3] ELROB2016b. (2016) Concept and rules, elrob 2016, 9th european land robot trials. ELROB 2016. [Online]. Available: http://www.elrob.org/elrob-2016-rules

[4] T. Graylin. (2016) Teleop-twist-keyboard. Robot Operating System ROS. [Online]. Available: http://wiki.ros.org/teleop-twist-keyboard

[5] T. Luksch and K. Berns, *Autonome mobile Systeme 2007*, 20th ed. Kaiserslautern, DE: Springer, 2007.

[6] M. Quigley, B. Gerkey, K. Watts, and B. Gassend. (2016) Joy. Robot Operating System ROS. [Online]. Available: http://wiki.ros.org/joy

[7] Videogameconsolelibrary. (2016) Xbox 360 controller. Videogameconsolelibrary.com. [Online]. Available: http://www.videogameconsolelibrary.com

A Robust and Flexible Software Architecture for Autonomous Robots in the Context of Industrie 4.0

Marco Wallner[1], Clemens Mühlbacher[1], Gerald Steinbauer[1], Sarah Haas[2], Thomas Ulz[2] and Jakob Chrysant Ludwiger[3]

Abstract— **The next industrial revolution should allow the production of individually configured items at the cost of a currently mass-produced commodity. To make this possible, autonomous robots play an essential role. These robots use their knowledge about the world and the task as well as sensor information to derive the next action to achieve a common goal. To give research in this area the possibility to test novel methods and algorithms for the next industrial revolution, the RoboCup Logistic League was established. The league uses a small shop floor environment wherein a group of robots has to produce customized goods within a given time-frame.**

In this paper, we present a software framework for a group of autonomous robots which deal with the problems of the RoboCup Logistic League. The software is separated into three distinct layers allowing modularity as well as maintainability. The framework provides all the needed functionality starting from creating plans for a fleet of robots, to the hardware skills to detect machines, to move between locations and interact with the physical world. To show the use of the software framework a use-case is presented. This use-case is the exploration of an unknown factory hall with a fleet of autonomous robots. All the presented solutions in this paper were tested in the RoboCup world championship 2016 in Leipzig. There the system showed its robustness and its capability to solve issues arising with the next industrial revolution.

I. Introduction

To increase customer satisfaction and sales, the industry tries to fulfill the desire of the customer for an individual product to the price of a product created by mass-production. As current methods in industrial production cannot provide these low costs for highly customized products, new approaches need to be developed. This is done through an ongoing automation in the industry which leads to the so-called Industrie 4.0 [1]. This term, originated by the German government, describes the abstract shape of the next generation in industry. One of the manifestations of this vision is the idea of a fully autonomous fabrication with smart machines.

To create such a smart factory two parts are essential. On the one hand, there are configurable smart machines necessary which are capable of performing the manufacturing steps. On the other hand, an intelligent delivery system between these devices is needed to allow the transportation of intermediate products to produce compound goods. As the usage of the machines and the scheduling can no longer be statically determined for a production line but need to adapt to the current requests, new algorithms need to be developed. To test these algorithms, the RoboCup Logistic League [2] was initiated as part of the annual RoboCup world championship [3]. The idea is to create a simplified version of a smart factory which serves as a testbed and a standardized benchmark for novel algorithms and approaches. With this, different aspects can be tested and evaluated regarding distinct components as well as the complete framework.

In this paper, we present a software framework to solve the challenges of the RoboCup Logistic League. The framework allows to schedule the entire fleet of robots and to react to changes in the environment in a reactive manner. Furthermore, the system is capable of reacting to faults during the execution of a task assigned to a robot or even a full drop-out of a robot in the fleet. To allow an efficient, modular and maintainable implementation, the framework uses a layered architecture. This approach allows to schedule the fleet on the highest level while considering the reactive interaction between the robot and its environment on the lowest level. To show how this software framework is used we present in this paper how the robotic fleet can explore an unknown shop floor. With the help of this example, we will show how the fleet is scheduled to cover the shop floor efficiently. Additionally, we will demonstrate how the machines are detected and identified.

The remainder of the paper is organized as follows. In the next section, we will discuss the RoboCup Logistic League in more detail. The proceeding section discusses the layered architecture. Afterward, we will discuss the software components which are used for the exploration of the shop floor environment. Before we conclude the paper we will discuss some related research. Finally, we conclude the paper and point out some future work.

II. RoboCup Logistic League

RoboCup, proposed and founded in 1997, is an annual international robotics competition. There, teams from all over the world compete in different disciplines, such as humanoid robots, soccer robots, rescue robots and the mentioned logistics competition.

The logistics league simulates the problems arising in a smart factory. Primarily it provides a standardized testbed for test new algorithms and approaches for smart factories.

[1]Marco Wallner, Clemens Mühlbacher and Gerald Steinbauer are with the Institute for Software Technology, Graz University of Technology, Graz, Austria. {mwallner,cmuehlba,steinbauer}@ist.tugraz.at.

[2]Sarah Haas and Thomas Ulz are with the Institute of Technical Informatics, Graz University of Technology, Graz, Austria. {thomas.ulz,sarah.haas}@tugraz.at.

[3]Jakob Chrysant Ludwiger are with the Institute of Automation and Control, Graz University of Technology, Graz, Austria. jakob.ludwiger@tugraz.at.

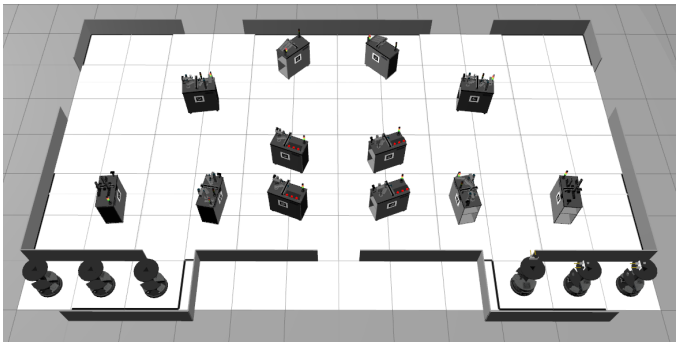


Fig. 1. RoboCup Logistics game-field in the simulator. Two attending teams with 3 robots and 6 machines each.

This is achieved by a controlled environment which contains a modular production system and a fleet of robots which need to be controlled. To allow fair conditions, a standardized robot platform (Robotino by Festo [4], three per team) is used for the mobile robots as well as standardized modular production systems (MPS by Festo, six per side) for the fabrication steps.

To emphasize the idea of a smart factory, the rules require that the fleet performs its task completely autonomously. Thus no intervention from humans is allowed. The idea is to put a robot in a workshop and let it explore the environment on its own, find machines to work with and produce products according to arriving orders. For this, the whole scenario is split into two phases, the exploration, and the production phase.

A. Exploration Phase

In this first phase, the robots have no knowledge about their environment. They have to explore the game-field (see Figure 1, a screenshot of the RoboCup Logistic League simulation [5]) and find the machines located there. To award points for the detection of such a machine, the robots have to report their observations to a central referee box. Each report contains the type of the machine, the shown status light as well as it is position in the field. If all the machines have been found, or after some deadline has passed, the next phase is invoked.

B. Production Phase

In this phase the actual production takes place. Random orders are placed by the central referee box, and both teams try to produce these as fast as possible.

1) Products: The products are mocked up as cups (base) with a defined number of rings pressed on it and a cap. The color of each part of the product is defined in the order.

2) Order: An order consists of the demanded product (e.g. a red base cup with two rings, the first ring blue, the second one yellow and a black cap) and its earliest delivery time as well as the deadline for the delivery of this product.

3) Modular Production System: To produce the ordered product, the mobile robots can use the six production systems of their team. There are four types of these workstations:

- *1x Base Station:* Providing bases in the demanded color.
- *2x Ring Station:* Mounting a ring in requested color on the provided base.
- *2x Cap Station:* Mounting a cap in required color on the provided base.
- *1x Delivery Station:* Point to deliver a product in the given time window.

As the mounting of a ring represents the addition of some feature to a product, some ring colors require additional bases as ”raw“ material. Thus also the need of deliveries for supply material is modeled in this scenario.

III. Software Architecture

To solve the tasks of the Logistics League, we propose the following software architecture. The software is split into three distinct layers, namely high-level, mid-level and low-level. Each layer is independent of the other layers within this concept. The lower layers provide functionality to the upper one [6]. Furthermore, higher layers command the actions of the lower layers.

The highest level of our software architecture is responsible for the connection of the different parts. It connects to the central referee box as well as an arbitrary number of connected robots as it can be seen in Figure 2.

To allow independent development and testing of each layer defined interfaces are necessary. Additionally, to feature different programming languages for each layer, Google’s protocol buffers are used for these interfaces. This independence is used as the high-level is written in Java, the mid-level using a belief-desire-intention [7] engine (openPRS [8], C) and the low-level is written in C++ using the ROS (Robot Operating System [9]) framework. The communication scheme for one robot can be seen in Figure 3.

For each interface dedicated protocol buffer (protobuf) messages are defined. With this structure, an increasing abstraction of the physical world can be achieved from the bottom up to the top. The message used between the high-level to the midlevel can be seen as an example in Listing 1.

Listing 1. Protobuf message to communicate between the layers.

```
message PrsTask {
  required Team teamColor = 1;
  required uint32 taskId = 2;
  required uint32 robotId = 3;

  optional ExecutionResult result = 4;

  optional ReportMachinesTask reportTask = 5;
  optional ExploreMachineTask explTask =  6;
  optional GetWorkPieceTask getWPTask = 7;
  optional PrepareCapTask prepCapTask = 8;
  optional DisposeProdTask dispProdTask = 9;
  optional DeliverProdTask deliProdTask = 10;
}
```

The lowest layer is responsible for small tasks close to the hardware, e.g. to move to a waypoint, grab an object, detect an AR-tag or analyze the status light of a machine (see Section IV-A.2 and Section IV-A.1 for further details). We call the execution of these small tasks skills in the remainder

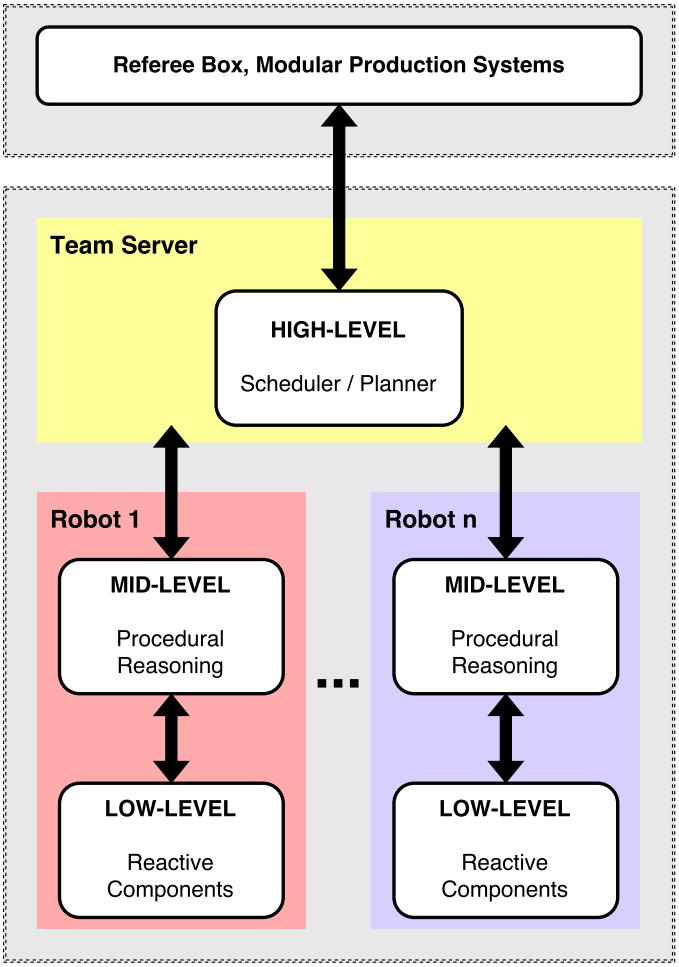


Fig. 2. Overall software architecture with links to the central refbox, the team server and the connected n robots.

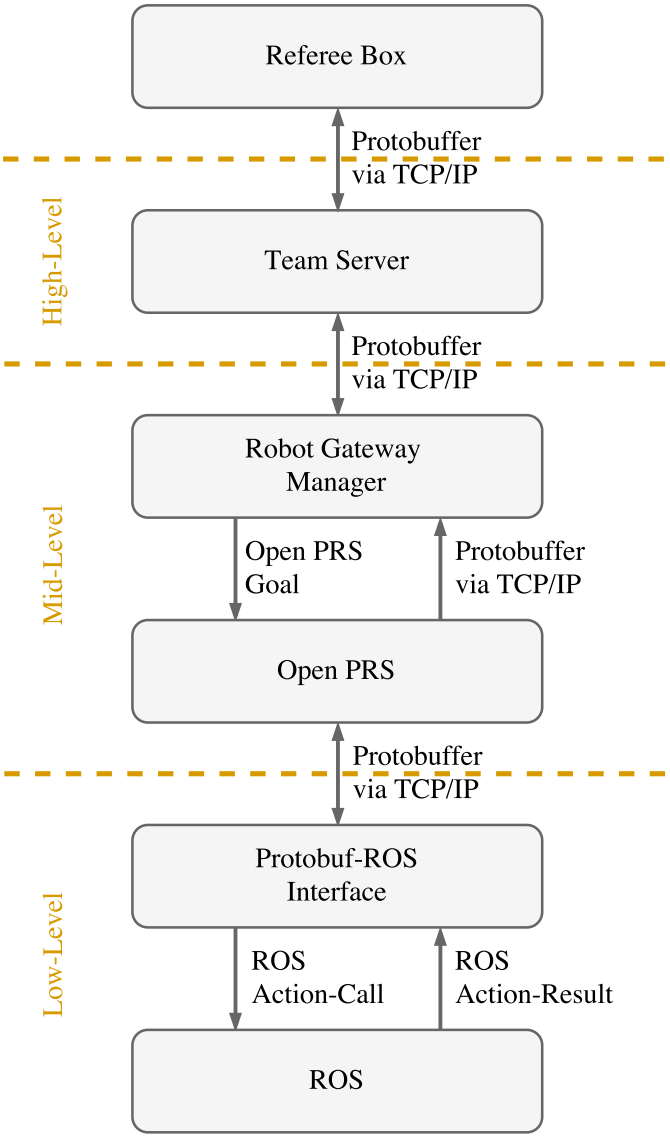


Fig. 3. Communication between the different layers for one robot using Google's protocol buffers.

of the paper. These skills are provided to the next higher layer, the mid-layer, via defined messages.

The mid-layer, therefore, can use these skills to perform more complex tasks such as exploring a zone of the game-field, get a base from the base station or deliver the product holding in its gripper. Additionally, a first error detection and recovery behavior are implemented here, e.g. the system checks if there is a product in the gripper after the low-level has successfully grabbed something. These complex tasks are again provided via defined protobuf messages to our highest layer, the team server.

Here a central knowledge-base is held and a game strategy is derived (see Section IV-B for further details). This central point enables the system to conclude a global optimal game strategy for the complete robotic fleet. The global strategy is derived using a simple planning system which uses a hierarchical task network [10] to properly create the products. Due to the centralized knowledge base one does not need to deal with synchronization of knowledge bases of the robot or distributed planning. Instead a "simpler" approach for planning can be applied.

IV. Selected Software Components

To get an idea of the functional interaction of our robot system some selected components are presented. First of all, two low-level modules necessary to detect and identify a machine are presented in SectionIV-A. Additionally, the scheduling algorithm (located at the high-level component of our system) which manages the discovery of the unknown game-field is presented in Section IV-B.

A. Machine Detection and Identification

To be able to gather information about the unknown environment it is necessary for the robot to recognize elements surrounding it. One important type of these elements is a modular production system, i.e. the machines capable of producing the ordered products. To identify the machines AR-tags are used which are placed at two sides of these machines.

Fig. 4. View of the robot in front of machine through the light detection camera at the RoboCup 2016 in Leipzig, Germany.

1) AR Detection: To localize objects in a defined frame of reference, it is first necessary to localize the robot itself. For this, a laser scanner and the knowledge of the fixed outer boundaries of the factory are used to infer the position of the robot using an adaptive Monte Carlo localization approach [11]. Using the particle filter also the confidence of the current location can be inferred.

With the known location of the robot and the known position of a camera mounted on the robot, it is possible to infer the position and orientation of seen augmented reality (AR) tags. For this, the open source AR-Tag tracking library Alvar is used. These tags are of defined size (allows derivation of distance to the tag) and are mounted at the input and the output of each machine. Each machine has a defined tag id for the input as well as the output. Using this knowledge, the position of the machine can be calculated having at least one of the tags seen. The accuracy and reliability of this measurements are further improved using a moving average filter. The filter is used to correct the estimate of the machine position with the help of several measurements. This raw data of the location of the machines is used by the higher layers as described in Section IV-B to determine which zone the machine is in. The information about the occupied zone is then reported to the referee box to earn points during the exploration.

2) Light Detection: To fully identify a machine, additionally to the AR-tag, the position and orientation of it, as well as the shown light pattern, needs to be reported. The light pattern is used to uniquely identify the machine. For this, the robot moves to a point in front of the machine. Afterwards, the robot captures an image of the machine. The captured image can be seen in Figure 4. These views have random backgrounds with arbitrary components, colors, and structures in it. Therefore, a detection of the light with the help of a blob detection is difficult to configure and is unreliable. Instead, one can exploit the fixed structure of the traffic lights. All of them have the same geometry regardless

Fig. 5. Cropped traffic light by the histogram of oriented gradients detector.

of the shown light pattern. They have a defined ratio between length and height, are sectored in three parts and are always upright.

This knowledge could be exploited by applying different manually generated and adjusted rules to determine the position of the traffic light in the image. Instead of these manually created rules, our approach uses a machine learning approach allowing the method to be more reliable, easy to configure and adapt to new environments with no effort. We use the static feature of the structure to train a histogram of oriented gradients (HOG) detector as described by [12]. This detector exploits that the mentioned static features manifest in a static gradient pattern.

Using the results of the HOG detector, a region of interest (ROI) can be extracted. The result of this cropping can be seen in Figure 5. Here the cropped traffic light is shown for all possible light combinations. The HOG detector has the advantage of almost no false-positive detections, i.e. if a ROI is found, there is a traffic light in it with a high probability.

To report the type of shown light pattern, a mapping from the traffic light image (which light is on and which is off) to a representing number is needed. For this, the lighting condition is encoded in a binary fashion, i.e. the representing state is calculated as:

$$\text{state} = s(\text{green})^0 + s(\text{yellow})^1 + s(\text{red})^2 \tag{1}$$

with

$$s(x) = \begin{cases} 2, & \text{if } x \text{ is on} \\ 0, & \text{else.} \end{cases} \tag{2}$$

With this mapping, a feed forward artificial neural network can be trained. We used the scaled conjugate gradient descent algorithm described in [13] to train the network. With this trained network, it is possible to map a newly seen image to a vector of probabilities describing the likelihood of each class as described in [14].

This gathered information can then be used by the higher layers to build up a knowledge base about the environment as described in Section IV-B.

The chain of a HOG-detector and a neural network was chosen as none of these approaches need a lot of computing power during the execution (only once at training time) to avoid the tuning of several parameters. The used neural network further increases the reliability as it can be trained to be resilient to different lighting situations.

B. Scheduling Algorithm

The robots have no information about their environment at the start of the game. Therefore, they have to use sensors

to observe the environment and to gather information. This is achieved by using a laser scanner for localization and cameras for machine detection.

To explore the game-field in an efficient manner with multiple robots, a scheduling algorithm has to be implemented. For this, our software architecture described in SectionIII comes into play. With the centralized team server, it is possible to generate a global exploration strategy and to combine the information delivered by all the robots into one reliable and consistent database.

During the exploration phase, all robots have the non-blocking task to report all seen machines, i.e. the zone, orientation (in discrete steps) and light pattern as well as the corresponding confidence. These updates are sent to the team server if parts of the information changes (e.g. orientation is corrected), new information is added (e.g. a light pattern is detected) or the confidence of a property rises. This information is then collected at the team server as an `observations` database.

To start the exploration with no observed date (i.e. at the beginning of the exploration phase) the default task for the first robot is to explore the top most left zone of the game field if the team starts at the right start box or the top right zone of the game field if the team starts at the left start box (see Figure 1). Using this simple strategy, the probability is very high that on the way to the destination zone the robot observed other machines and reported them to the team server. As soon as another robot is ready for a task or the first robot has finished its navigation, the robot gets the task assigned to visited a zone. During the visiting of a zone, the robot detects if a machine is within the zone. If a machine is present, the robot performs a light detection of the machine. If no machine position is reported so far, the robot gets a backup task to visit a randomly chosen zone which was not visited before. Otherwise, the robot is sent to a zone with a high probability that a machine is in this zone (one robot has reported that there should be a machine) but was not visited before. If all zones are visited, the zone with the lowest confidence is chosen as the next task. This allows maximizing the confidence of the machine information. The simplified algorithm can be seen in Algorithm 1.

With the start position in the team boxes (as it can be seen in Figure 1) it is very likely that at least one machine is seen already in the start position. Thus the usual procedure is that the first robot directly reports at least one machine at start-up. The team server creates a task for this robot and sends it to discover the light state and the correct orientation. On its way, the robot reports other machines, and so the other robots can be sent to zones with machines too. Thus the backup solution to drive to some randomly chosen zone is rarely used.

This dynamic scheduling allows a very efficient and fast exploration of the whole game field. This is necessary as the game field is rather large ($12m \times 6m$) for the low speed these robots are able to move.

Another advantage of the global view of the team server can be used here too. The machines are distributed at the game-field in a symmetric fashion to allow fair conditions for both teams. This constraint can be used for a sanity check of the reports, i.e. before the final result is sent to the referee box, it is checked if it makes sense and the most probable consistent set of observations is reported.

Algorithm 1: Exploration Algorithm

Input: observations, notVisitedZones, #MPS, thresh
Output: task

```
1:  if observations = ∅ then
2:    if oppositeZone ∈ notVisitedZones then
3:      return exploreZone(oppositeZone)
4:    else
5:      zone = chooseRandom(notVisitedZones)
6:      return exploreZone(zone)
7:    end if
8:  else
9:    if numZonesNotVisited(observations) > 0 then
10:     zones = zonesNotVisited(observations)
11:     zone = getZoneWithLowestConfidence(zones)
12:     return exploreZone(zone)
13:   end if
14:   if mFound(observations, thresh) < #MPS then
15:     zone = chooseRandom(notVisitedZones)
16:     return exploreZone(zone)
17:   else
18:     zones = zonesNotVisited(observations)
19:     zone = getZoneWithLowestConfidence(zones)
20:     return exploreZone(zone)
21:   end if
22: end if
```

After the exploration phase, the set of reliable machine positions and orientations is then broadcasted to the connected robots to allow them to work during the production phase with the gathered information. Also if one robot has to be restarted during the production phase, the information about the position of the machines is provided as a new (or in this case restarted) robot connects to the team server.

V. Related Research

In the previous section we have discussed our software architecture how to solve the challenges in the RoboCup logistic league. Within this section we will discuss another approach to solve the problems in RoboCup Logistic League. We will compare our approach to the Carologistics Team which won the world championships several times. As the Carologistics Team describes in its team description paper [15], they also use a three-layer architecture.

A. Carologistics

The main difference is that no central coordinator is used. Instead a distributed, local-scope and incremental reasoning approach [16] is chosen. This has the advantage of no single point of failure but also the disadvantage that no optimal global strategy can be derived. To keep a consistent view of

the physical world, a permanent synchronization of the robots is needed. For this purpose, one of the agents is chosen to act as a leader responsible for collecting and distributing a view of the world and manages reservation of resources.

1) Software Architecture: The function of each robot is separated into the three distinct layers responsible for deliberation (high level), i.e. decision making and planning, a reactive skill engine (mid-level) and low-level components for e.g. motion and vision.

The reasoning and planning component is implemented using a CLIPS rule engine [17]. This allows an incremental reasoning to derive at any time-point for each of robot a local optimal decision. The mid-level is designed as a Lua-based behavior engine [18]. With this, simple and complex skills can be modeled as a hybrid state machine. This modularity allows tuning and optimization of skills for specific tasks. The underlying robot framework used is Fawkes [19]. This framework is an alternative to ROS and provides several low-level functionalities as e.g. AMCL, hardware interfaces to the Robotino base and navigation plugins.

2) Light Detection: The light pattern detection (described in Section IV-A.2) is solved by the Carologistics Team using a more complex and more configuration-intense way. The region of interest (ROI) is cropped using the fusion of the camera and the laser scanner. The robot is aligned with the use of the mounted AR-tag. As this tag can be mounted arbitrarily on the machine, this only allows a course alignment. With the use of the laser scanner and the knowledge of the type of machine (via the AR-tag), the relative position of the mounted traffic light can be calculated. For this, the exact location of the light for each side of the machine is necessary with respect to the machine base. After this, the region of interest can be restricted a first time. With the knowledge of the position of the laser scanner as well as the camera, it is possible to calculate the position of this ROI in the image frame. Here several heuristics are used to find the shown traffic light, e.g. the fixed width to height ratio, that there have to be three distinct lights stacked in a vertical manner and much more. Having this, the state of the traffic light is determined using the color of the ROIs for the red, orange and green image section. This is done using a defined space for off and on in the YUV color space.

Our approach avoids the need for all the configuration by using the HOG-detector and the neural network. The detector eliminates the need for geometric heuristics and knowledge about the machine, and the neural network generalizes enough (trained with several lighting conditions) to detect the state of the traffic light without the need of a tuned color model. This allows more robustness as e.g. different lighting, or a displacement of the mounted camera or the laser scanner would lead to wrong classifications with the solution presented by the Carologistics team.

VI. Conclusion and future work

With the help of the next industrial revolution, it should be possible to produce individual configured products to the price of current mass-production. This ambitious scheme requires smart factories with modular machinery and an intelligent and flexible transportation system. Such transport can be provided by a fleet of autonomous robots. To offer a standardized testbed for different aspects of such smart factories the RoboCup Logistic League was established.

In this paper, we presented a software architecture which can be used to solve various problems appearing in the context of the RoboCup Logistic League. The software architecture consists of three layers which interact with clearly defined interfaces. The top layer manages the entire robotic fleet, generates an optimal global schedule, and is responsible for error detection and correction. For this, it uses the mid-layer which provides complex tasks (e.g. explore a zone, deliver a product). Here these skills are decomposed, and the mid-layer commands simple skills (move to a waypoint, open the gripper) to the lowest layer.

The software was successfully tested at the RoboCup world championship 2016 and allowed us to rank among the top three teams [20].

Besides the general software architecture, we described in this paper several components in more detail. These components allow the robot to explore an unknown factory. The presented components range from a scheduling mechanism to distribute the work onto the entire fleet down to mechanisms to detect the type of the machine defined by a signal light pattern.

The system is designed in such a way that faults are detected. Thus the system can react to faults properly. This allows the system to reliably to execute its task. To improve the reliability of our system even further the next step is to implement an online diagnosis system as described in [21]. This system can use different measures (e.g. publishing frequency of particular topics, time to respond to actions) to detect abnormal system behavior and furthermore calculate a diagnosis.

References

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, p. 239, 2014.

[2] T. Niemueller, D. Ewert, S. Reuter, A. Ferrein, S. Jeschke, and G. Lakemeyer, "Robocup logistics league sponsored by festo: A competitive factory automation testbed," in *Automation, Communication and Cybernetics in Science and Engineering 2015/2016*. Springer, 2016, pp. 605–618.

[3] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," in *Proceedings of the first international conference on Autonomous agents*. ACM, 1997, pp. 340–347.

[4] U. Karras, D. Pensky, and O. Rojas, "Mobile robotics in education and research of logistics," in *IROS 2011–Workshop on Metrics and Methodologies for Autonomous Robot Teams in Logistics*, vol. 72, 2011.

[5] F. Zwilling, T. Niemueller, and G. Lakemeyer, "Simulation for the robocup logistics league with real-world environment agency and multi-level abstraction," in *Robot Soccer World Cup*. Springer, 2014, pp. 220–232.

[6] E. Gat *et al.*, "On three-layer architectures," *Artificial Intelligence and Mobile Robots*, vol. 195, p. 210, 1998.

[7] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge, "The belief-desire-intention model of agency," in *International Workshop on Agent Theories, Architectures, and Languages*. Springer, 1998, pp. 1–10.

[8] F. F. Ingrand, R. Chatila, R. Alami, and F. Robert, "Prs: A high level supervision and control language for autonomous mobile robots," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 43–49.

[9] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[10] K. Erol, J. A. Hendler, and D. S. Nau, "Umcp: A sound and complete procedure for hierarchical task-network planning." in *AIPS*, vol. 94, 1994, pp. 249–254.

[11] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," *AAAI/IAAI*, vol. 1999, no. 343-349, pp. 2–2, 1999.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[13] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.

[14] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Springer, 1990, pp. 227–236.

[15] T. Niemueller, T. Neumann, C. Henke, S. Schönitz, S. Reuter, A. Ferrein, S. Jeschke, and G. Lakemeyer, "Improvements for a robust production in the robocup logistics league 2016."

[16] T. Niemueller, G. Lakemeyer, and A. Ferrein, "The robocup logistics league as a benchmark for planning in robotics," in *WS on planning and robotics (PlanRob) at Int. Conf. on Aut. planning and scheduling (ICAPS)*, 2015.

[17] R. M. Wygant, "Clipsa powerful development and delivery expert system tool," *Computers & Industrial Engineering*, vol. 17, no. 1-4, pp. 546–549, 1989.

[18] T. Niemüller, A. Ferrein, and G. Lakemeyer, "A lua-based behavior engine for controlling the humanoid robot nao," in *Robot Soccer World Cup*. Springer, 2009, pp. 240–251.

[19] T. Niemueller, A. Ferrein, D. Beck, and G. Lakemeyer, "Design principles of the component-based robot software framework fawkes," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2010, pp. 300–311.

[20] S. Haas, D. Keskic, C. Mühlbacher, G. Steinbauer, T. Ulz, and M. Wallner, "Robocup logistics league tdp graz robust and intelligent production system grips," 2016.

[21] S. Zaman, G. Steinbauer, J. Maurer, P. Lepej, and S. Uran, "An integrated model-based diagnosis and repair architecture for ros-based robot systems," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 482–489.

3D Vision Guided Robotic Charging Station for Electric and Plug-in Hybrid Vehicles

Justinas Mišeikis[1], Matthias Rüther[2], Bernhard Walzel[3], Mario Hirz[3] and Helmut Brunner[3]

***Abstract*— Electric vehicles (EVs) and plug-in hybrid vehicles (PHEVs) are rapidly gaining popularity on our roads. Besides a comparatively high purchasing price, the main two problems limiting their use are the short driving range and inconvenient charging process. In this paper we address the latter by presenting an automatic robot-based charging station with 3D vision guidance for plugging and unplugging the charger. First of all, the whole system concept consisting of a 3D vision system, an UR10 robot and a charging station is presented. Then we show the shape-based matching methods used to successfully identify and get the exact pose of the charging port. The same approach is used to calibrate the camera-robot system by using just known structure of the connector plug and no additional markers. Finally, a three-step robot motion planning procedure for plug-in is presented and functionality is demonstrated in a series of successful experiments.**

I. INTRODUCTION

Nowadays it is common to see electric vehicles and plug-in hybrids on our roads. Worldwide plug-in vehicle sales in 2016 were 773600 units, 42% higher compared to 2015 [1]. For example Norway plans to rule out sales of any combustion engine cars by 2025 [4]. However, a new problem being faced by EV and PHEV drivers is having an accessible, fast and convenient battery charging, especially when traveling longer distances. It is a common problem of fast chargers being idly occupied after the car is fully charged if the owner does not return to the vehicle. For example, Tesla has added an additional idle fee to discourage drivers leaving their cars at the chargers for longer than necessary [7]. A solution to avoid this problem and to enable a comfortable fast charging would be an automated robot-based charging system combined with automated car parking.

A. Charging Ports and Cables

Worldwide, there are many types of EV and PHEV charging ports, as well as different charging port placement locations on the vehicle. Each one of them has benefits and detriments, and car manufacturers have not decided on a common standard yet. This introduces an additional inconvenience of finding the correct type of charger, or having to carry a number of bulky adapters. As long as there is no standard, it would be more convenient to let the charging station detect the correct port type and adapt accordingly.

Another issue is the current weight and stiffness of a quick charging cable. For example, the weight of a CCS-Type 2 charging cable rated for the power up to 200 kW is 2.26 kg/m and outer diameter of 32 mm. With longer cable lengths, this becomes difficult for people to handle, but would not be an issue for a robot [6]. Cooled charging cables can help to solve this problem without increasing the cable diameter, but these are not yet standard [17].

B. Existing Automated EV Charging Methods

Automatic charging solutions have been researched both in academic and industrial environments. Volkswagen has presented an e-smartConnect system, where a Kuka LBR-iiwa robot automatically plugs in the vehicle after it autonomously parks in a specific target area (allowing for less than 20 cm by 20 cm error). It is also limited to one charging port type [8].

Tesla has demonstrated a concept of a snake-like robot automatically plugging in their EV, however, no technical details on the charging port localisation or robot operation were revealed [9].

The Dortmund Technical University has presented a prototype of the automatic charging system called ALanE. It is based on a robot arm capable of automatically plugging and unplugging a standard energy supply to an electric vehicle. The system is controlled via smartphone. However, full capabilities and flexibility of this concept system are not clear [3].

The NRG-X concept presents itself as a fully automatic charging solution. It can be adapted to any EV or PHEV and is capable of fast charging. Furthermore, it has a tolerance for inaccurate parking positions. The NRG-X system is based on combination of conductive and inductive charging on the under-body of the vehicle, thus an adapter for the vehicle is necessary. Furthermore, in the current concept configuration the charging power is limited to 22 kW [5], which results in over 7 times longer charging compared to 170 kW charging [22] and perspective 350kW [11].

[1]Justinas Mišeikis is with Department of Informatics, University of Oslo, Oslo, Norway justinm@ifi.uio.no

[2]Matthias Rüther is with Graz University of Technology, Institute for Computer Graphics and Vision, Graz, Austria ruether@icg.tugraz.at

[3]Bernhard Walzel, Mario Hirz and Helmut Brunner are with Graz University of Technology, Institute for Automotive Engineering, Graz, Austria {bernhard.walzel, mario.hirz, helmut.brunner}@tugraz.at

Comparisons of the time taken to charge a vehicle using different charging systems is shown in Figure 1.

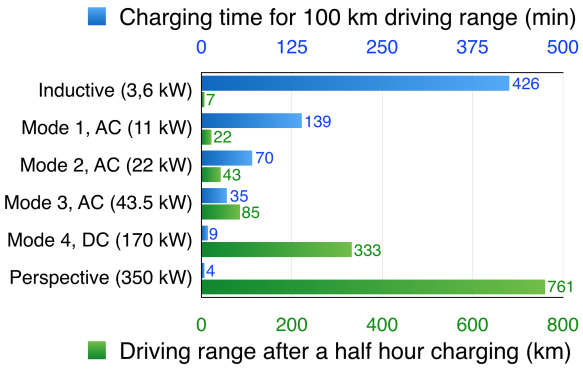


Fig. 1. Driving distance and charging time comparison of different charging systems [22].

C. Related Research

Automated charging has been well researched, especially for mobile robots. Typically, there is a custom made charging station, which is localized by the robot either using a direct communication or using computer vision based methods. These methods are normally based on having special markers on the charging station, which are localised in order for the robot to correctly align itself and approach the station. Removing markers would impede the operation [12] [19] [18] [14].

Another concept developed specifically for the detection of charging ports on EVs was based on adding an array of RFID tags on the car. Reading RFID signals allows to find the exact position and orientation of the charging port and plug it in automatically [16]. However, this still requires modification to the vehicle and would not support non-adapted cars.

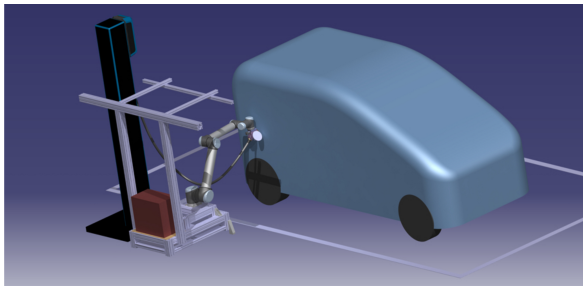


Fig. 2. CAD model of the robotic charging station concept.

D. Method Presented in This Work

We present a conductive robot-based automated charging method for EVs and PHEVs, which does not require any modifications to existing vehicles. First of all, we present a quick eye-to-hand calibration procedure to calibrate the vision sensor and the robot to work in the same coordinate system. It estimates both, the placement of the vision sensor in relation to the robot base as well as between the end-effector and the plug. Then we use shape-based matching and triangulation to locate and identify the charging port of the car and guide the robot, holding a charging cable, to precisely plug in the charger. Once the car is fully charged, the robot will automatically unplug from the vehicle, which will be ready to be driven away. The visualisation of the concept robotic charging station is shown in Figure 2.

This paper is organized as follows. We explain the proposed method in Section II. Then we provide our test setup, experiments and results in Section III, followed by conclusions and future work in Section IV.

II. METHOD

A. Detection of the Charging Port

A majority of the car charging ports are manufactured from texture-less black plastic material, making it difficult to obtain good features in the camera image. Similarly, the measurements made using time-of-flight cameras, which use the projection of infrared (IR) light, are noisy and inaccurate due to IR absorption by the material. As an alternative solution, a stereo-camera setup was used as the vision sensor.

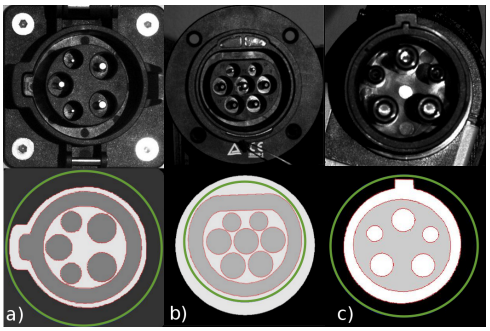


Fig. 3. Input images, simplified template models and automatically created shape-based templates for matching. Type 2 socket is shown in column a), type 1 socket in b) and type 2 connector plug is shown in c). Green circles define the area of interest for the model creation and the red outline line defines the created shape model.

The first step in the detection procedure is to find the location of the charging port in stereo images using shape-based template matching. Models were created for two types of the charging ports as well as the power plug connector, later to be used for eye-to-hand calibration. Figure 3 shows the camera images and simplified model images, which are used to automatically generate shape-based templates later to be used for matching. Template matching was performed using a *Halcon Machine Vision* software, which has proven to perform well in given conditions of low-contrast input images [2]. Matching results in a 2D Affine transformation matrix defining the template location in the image.

By taking x and y coordinates of the corresponding object points in images from each of the stereo

cameras, the depth information defined by z-axis can be calculated. The vision sensor in our setup has both stereo cameras fixed in relation to each other looking slightly inwards, with rotation around Y (vertical) axis. Solving Eq. 1 provides the real-world coordinates X, Y and Z of a point seen by the stereo cameras. Inputs (x_1, y_1) and (x_2, y_2) are the point coordinates in camera 1 and camera 2 respectively. Variable f is the focal length of the camera and b defines a baseline (distance) between the stereo cameras. Rotation between the cameras around Y-axis is defined by θ.

$$\begin{aligned} Z_0 &= \frac{b}{tan(\theta)} \\ Z &= \frac{b*f}{x_1 - x_2 + \frac{f*b}{Z_0}} \\ X &= \frac{x_1 * Z}{f} \\ Y &= \frac{y_1 * Z}{f} \end{aligned} \tag{1}$$

After the charging port is found in the input images, stereo triangulation is used to obtain 3D real-world coordinates of the port position, providing 5 to 7 reference points depending on the charging port type. Using the points, a perspective transformation is calculated using the least squares fit method to obtain the exact position and orientation of the charging port in relation to the vision sensor. Least squares fit for finding the orientation optimises for 3 unknowns (A, B and C), which later are mapped to roll, pitch and yaw angles. The least square error function is defined in Eq. 2, where x, y and z are coordinates of the reference points.

$$e(A,B,C) = \sum (Ax + By + C - z)^2 \tag{2}$$

Then, the error function is differentiated and set to zero, as shown in Eq. 3.

$$\begin{aligned} \frac{\partial e}{\partial A} &= \sum 2(Ax + By + C - z)x = 0 \\ \frac{\partial e}{\partial B} &= \sum 2(Ax + By + C - z)y = 0 \\ \frac{\partial e}{\partial C} &= \sum 2(Ax + By + C - z) = 0 \end{aligned} \tag{3}$$

The resulting linear equations with 3 unknowns are solved to get the orientation of the object. This can also be seen as 3D plane fitting to the given points.

B. Marker-less Eye-to-Hand Calibration

In order to operate the vision sensor and the robot in the same coordinate system, eye-to-hand calibration is necessary. The eye-to-hand calibration estimates the transformation between the vision sensor and the robot base. Using this transformation, the position of any object detected by the vision sensor can be recalculated into the coordinate system of the robot, allowing the robot to move to, or avoid that location.

Normally, a well structured object, like a checkerboard of known size and structure is used in the calibration process. However, it requires mounting it on the end-effector of the robot and can still result in additional offsets. We use the known structure of the connector plug and previously presented shape-based template matching with orientation estimation to obtain the precise pose. Eye-to-hand calibration is based on an automatic calibration procedure for 3D camera-robot systems, which uses the calibration method proposed by Tsai et al [15] [21].

The result of the eye-to-hand calibration are two transformation matrices. The first one defines the position of the vision sensor in relation to the robot base and the second one defines the position of the end point of the connector plug in relation to the end-effector of the robot.

The marker-less eye-to-hand calibration can be beneficial if the robot is placed on a moving platform, so the relative position between the vision sensor and the robot can change. Furthermore, it would benefit in cases when the robot has interchangeable end-effector attachments with different connector plugs. In both of these cases, recalibration procedure could be done automatically without any reconfiguration.

C. Robot Motion Planning

Given the limited workspace and all the movements being defined by camera measurements, robot control in Cartesian coordinates was used. The *MoveIt!* framework, containing multiple motion planning algorithms, was used for the initial testing [20]. The best performance in the defined case was demonstrated by the RRT-connect algorithm, which is based on the rapidly exploring random trees [13].

In order to get smoother motion execution and more human-like motions, a velocity based controller was used instead of the standard one provided in ROS. Better performance is achieved by calculating and directly sending speed commands to each of the robot joints, thus reducing the execution start time to $50-70$ ms compared to around 170 ms using the official ROS UR10 drivers [10].

D. Plugging-In Procedure

After the pose of the charging port is calculated, the coordinate system is assigned with the origin placed at the center of the plug and Z-axis looking outwards. Similarly, the coordinate system is assigned to the connector plug, which is held by the robot. The goal of the plug-in procedure is to perfectly align connector plug with the charging port, so the last movement is simply along one axis. In order to achieve that, a

three-step procedure was used, visualised in Figure 4. Firstly, the robot moves the plug at high velocity to the approach position, which is within a 0.1 meter radius from the charging port. The second step is to reduce the velocity to 10% of the maximum robot joint speed and move to the final alignment position. In this pose, the connector plug and the charging port are fully aligned by their Z-axis and just a few millimeters away from the contact point. The last step is to move at just 2% of the maximum speed along Z-axis and perform the plug-in motion. During this move, the forces and torques exerted on the end effector of the robot are monitored. In case the forces exceed a given threshold, the system is halted to prevent any damage.

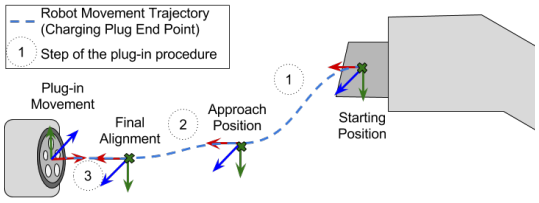


Fig. 4. Three step plug-in procedure plan. Firstly, the robot moves the connector plug to the *Approach Position*, which lies approximately 0.1 meter away from the charging port. The second move aligns the Z-axes of the charging port and the plug, and gets the plug just a few millimeters away from the port. The final plug-in movement performs the plugging in motion along Z-axis.

E. Unplugging

After the vehicle is charged fully or to the desired battery level, the robot has to disconnect the charger. Under the assumption that there were no position changes during the charging process, the unplugging procedure was simplified to follow the recorded way-points of the plug-in procedure in the inverse order. First, the robot gets back to the approach position and then returns to the stand-by position, where it is docked while waiting for the next task. The stand-by position ensures an unobstructed view of the parked vehicle for the vision sensor.

III. EXPERIMENTS AND RESULTS

A. Experiment Setup

At the current stage, the testing was limited to the lab environment. The experimental setup consists of an UR10 robot arm, a vision sensor containing stereo cameras and a charging port holder with inter-changeable charging ports. The charging port holder has variable height, position and angle to simulate various imperfect parking positions and differences in charging port locations on the vehicle. Two types of the charging ports, Type 1 and 2, have been used, as previously seen in Figure 3.

The connector plug is attached to the end-effector of the robot using a custom 3D printed attachment, shown in Figure 5. The charging cable is also attached

Fig. 5. Custom 3D printed connector plug holder attached to the end-effector of the UR10 robot.

to simulate realistic weight exerted on the robot during the operation. The whole experimental setup is shown in Figure 6.

The final goal was to locate the charging port using the vision sensor and estimate its pose. Then, the pose is transformed into the coordinate system of the robot and the end point of the connector plug is aligned and plugged in to the charging port. After a brief pause to simulate the charging process, the unplugging movement is performed and the robot moves back to the stand-by position.

Results of each part of the process are discussed separately and followed by the final evaluation of the whole system.

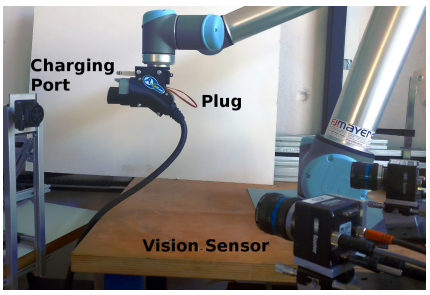


Fig. 6. The whole experiment setup. On the left the charging port holder can be seen. The robot is holding the connector plug, and the vision sensor made up of two stereo cameras is seen on the right hand side.

B. Template Matching

Template matching for Type 1 and Type 2 charging ports as well as the connector plug (Type 2) has worked well for various illumination and angles up to 45° relative to the viewing angle of the camera. The matching confidence score for good alignment was over 95%. The recognition speed on the full camera image was varying between 300*ms* and 800*ms*. By narrowing down the search area, for example by identifying the darker than average regions in the image, the recognition speed can be reduced to under 150*ms*. The results can be seen in Figure 7.

The limit for the successful recognition under low illumination or overexposure was when the edges of the socket or plug structure are still visible. The

connector plug was made out of more reflective plastic, resulting in a few cases when reflections caused the accuracy issues regarding the rotation. However, these issues were observed very rarely under specific viewing angles, and matching accuracy dropped below 90%, so these cases could be easily identified.

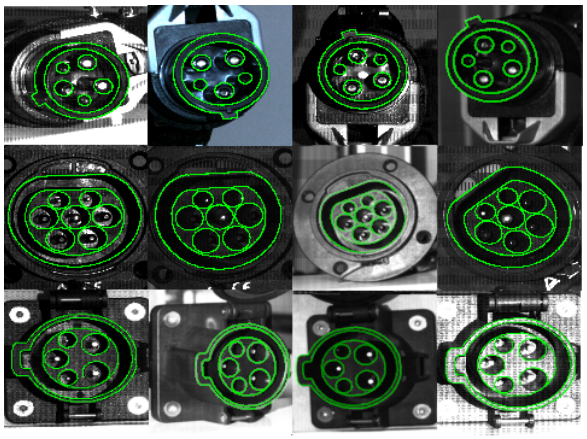


Fig. 7. Results of the template matching. A high variety of angles and lighting conditions were tested. Viewing angles up to 45° resulted in successful detection with accuracy dropping beyond that. Row 1: Type 2 connector plug. Row 2: Type 1 socket. Row 3: Type 2 socket.

C. Eye-to-Hand Calibration

In the given configuration, the structure of the connector plug was used as a marker for eye-to-hand calibration. During the calibration process it was turned to face the vision sensor, while during the normal operation it faces away from the camera. Furthermore, the outer ring of the plug is angled, so the pins of the plug had to be used as reference points to get the accurate calibration.

The end point of the connector plug was rotated around each of the axis as well as moved to different locations within the field-of-view of the vision sensor. In total, 26 poses were recorded and used until the calibration converged. Additionally, 3 instances were discarded because of the incorrect template matching result. The average translation error within the working space was reduced to 1.5*mm*, which was sufficient for our application at this stage. Possibly, having more poses would reduce the positional error even further. With the eye-to-hand calibration completed, coordinate frames for the camera position and the end point of the connector plug can be added to the model, as shown in Figure 8.

D. Finding Charging Port Pose and Robot Movements

As the final evaluation, we used the whole process pipeline and analysed whether the plug-in motion was successful or not.

There were 10 runs executed in total using Type 2 connectors. For the first 5 runs the charging port was

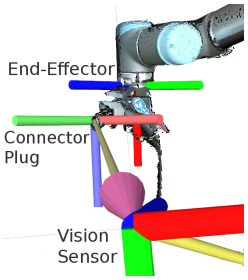


Fig. 8. Eye-to-hand calibration results. Visualisation of the assigned coordinate frames to the vision sensor, the end-effector of the robot and the end point of the connector plug. Resulting point cloud is overlayed onto the visualisation of the robot model.

angled at 10° in relation to the vision sensor, and for the remaining 5 runs, the angle was increased to 30°.

The robot successfully connected the plug 8 out of 10 times. Both failures occurred by missing the rotation of the plug, which were determined by the misalignment of the guidance slot on the charging port. However, the safety stop automatically initialised in both of the cases ensuring that the robot stopped before causing any damage.

TABLE I

SUMMARY OF THE PLUG-IN MOTION EXPERIMENTS WITH CHARGING PORT PLACED AT TWO DIFFERENT ANGLES

Exp	**Charging Port Angle** 10°	**Charging Port Angle** 30°
1	Success	Success: Misalignment
2	Success: Misalignment	*Failed: Missed rotation*
3	Success	Success
4	*Failed: Missed rotation*	Success: Misalignment
5	Success: Misalignment	Success: Misalignment

However, even when the plug was successfully inserted in the charging port, there were some alignment issues. In 5 out of 8 successful runs, the plug was not fully inserted into the charging port. It was caused by a small angular offset varying between 2° and 5°. The contact was still made, so the charging process would be successful, however, there was additional strain due to imperfect alignment. The misalignment occurred more frequently during the experiments, where the charging port was placed at 30° angle. The results are summarised in Table I.

As expected, the unplugging process was successful during all the runs. It simply follows already executed trajectory in the inverse order, meaning that as long as the position of the charging port did not change during the time it was plugged in, there should be no issues with the unplugging process.

IV. CONCLUSIONS AND FUTURE WORK

We have presented a vision-guided and robot-based automatic EV and PHEV charging station. The goal is to allow automated conductive fast charging of electric and hybrid vehicles and avoid the issue of a charged car taking up the space when it is not necessary.

The presented approach is a combination of multiple methods. First of all, the shape-based template matching is used to identify the charging port type and use the information from stereo cameras to precisely estimate its position and orientation. The same method is used in the marker-less eye-to-hand calibration, which results in the transformation matrices to be used to convert the position of the charging port from the coordinate system of the vision sensor to the robot. Then, the robot, holding a connector plug, is used to approach and finally plug in the charger cable into the EV or PHEV. Having a precisely estimated orientation is a big challenge and observation of the forces exerted on the end-effector of the robot are necessary to identify any possible misalignment, and stop or readjust if needed. Our approach has proven to work in the lab conditions under indoor illumination and using a custom made charging port holder.

Adding a force sensor to the robot would allow the robot to operate using the impedance controller based on force measurements and adjust it during the plug-in procedure according to the strains observed on the end effector. This would likely to be a solution for the observed cases with misalignment issues.

The project will be continued by improving the connector plug detection accuracy and automating the marker-less calibration procedure, where the robot would perform calibration movements automatically.

Furthermore, current tests were performed under the assumption that the charging port lid or cap was already opened. A linear actuator is already included in the setup, however, it was not used in current experiments. Future work includes finding the charger lid, identifying it's opening mechanism and using the robot to open and close it for the charging process. This would also require identification of the vehicle model to indicate the correct parking position and localise the approximate position of the charging port.

With the test electric vehicle to be delivered in the near future for testing purposes, the system will be evaluated on the real EV in the garage setup and outdoor tests. Communication between the vehicle and the charging station is also under development and this will enable the combination of the robot-based charging system with autonomous parking functions.

REFERENCES

[1] "EV-Volumes - The Electric Vehicle World Sales Database," http://www.ev-volumes.com/country/total-world-plug-in-vehicle-volumes/, (Accessed on 03/08/2017).

[2] "HALCON The power of machine vision - MVTec Software GmbH," http://www.mvtec.com/products/halcon/, (Accessed on 03/07/2017).

[3] "Ladesystem der TU Dortmund betankt Elektroautos automatisch - Fakultt fr Elektrotechnik und Informationstechnik - TU Dortmund," http://www.e-technik.tu-dortmund.de/cms1/de/Service_Termine/Weitere_Meldungen/Archiv/Ladesystem_Elektroautos/index.html, (Accessed on 03/03/2017).

[4] "Norway to completely ban petrol powered cars by 2025," http://www.independent.co.uk/environment/climate-change/norway-to-ban-the-sale-of-all-fossil-fuel-based-cars-by-2025-and-replace-with-electric-vehicles-a7065616.html, (Accessed on 03/08/2017).

[5] "NRG-X - Automatic Charging for E-Mobility," http://www.nrg-x.com/, (Accessed on 03/06/2017).

[6] "PHOENIX CONTACT — Homepage Corporate Website," https://www.phoenixcontact.com/, (Accessed on 03/07/2017).

[7] "Tesla owners who leave cars at Superchargers after charging will pay $0.40/minute," http://www.theverge.com/2016/12/16/13990854/tesla-supercharger-electric-fee-model-s-parking, (Accessed on 03/08/2017).

[8] "e-smartconnect: Volkswagen is conducting research on an automated quick-charging system for the next generation of electric vehicles," https://www.volkswagen-media-services.com/en/detailpage/-/detail/e-smartConnect-Volkswagen-is-conducting-research-on-an-automated-quick-charging-system-for-the-next-generation-of-electric-vehicles/view/2448500/7a5bbec13158edd433c6630f5ac445da, July 2015, (Accessed on 03/03/2017).

[9] "Tesla Unveils Snakelike Robot Charger for Electric Cars," http://www.livescience.com/51791-tesla-electric-car-robot-charger.html, 2015 August, (Accessed on 03/03/2017).

[10] T. T. Andersen, "Optimizing the Universal Robots ROS driver." Technical University of Denmark, Department of Electrical Engineering, Tech. Rep., 2015.

[11] C. Bracklo. (2016, Mar.) CharIN e.V.: The road to the success of a global charging standard - technology, standardization, organization. 2016. [Online]. Available: http://charinev.org/media/association-infos/

[12] U. Kartoun, H. Stern, Y. Edan, C. Feied, J. Handler, M. Smith, and M. Gillam, "Vision-based autonomous robot self-docking and recharging," in *Automation Congress, 2006. WAC'06. World*. IEEE, 2006, pp. 1–8.

[13] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.

[14] R. C. Luo, C. T. Liao, and K. C. Lin, "Vision-based docking for automatic security robot power recharging," in *Advanced Robotics and its Social Impacts, 2005. IEEE Workshop on*. IEEE, 2005, pp. 214–219.

[15] J. Miseikis, K. Glette, O. J. Elle, and J. Torresen, "Automatic calibration of a robot manipulator and multi 3D camera system," in *2016 IEEE/SICE International Symposium on System Integration (SII)*, Dec 2016, pp. 735–741.

[16] H. Oh, B. An, A. L. Smith, M. Raghavan, and F. C. Park, "An RFID localization algorithm for a plug-in electric vehicle recharging robot," in *Consumer Electronics (ICCE), 2015 IEEE International Conference on*. IEEE, 2015, pp. 176–177.

[17] PHOENIX CONTACT, "E-Mobility DC-Quickcharging with up to 350 A, online document," 2015. [Online]. Available: https://www.phoenixcontact.com/assets/downloads_ed/global/web_dwl_promotion/52007525_DE_INT_E-Mobility_LoRes.pdf

[18] M. Silverman, B. Jung, D. Nies, and G. Sukhatme, "Staying alive longer: Autonomous robot recharging put to the test," *Center for Robotics and Embedded Systems (CRES) Technical Report CRES*, vol. 3, p. 015, 2003.

[19] M. C. Silverman, D. Nies, B. Jung, and G. S. Sukhatme, "Staying alive: A docking station for autonomous robot recharging," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 1050–1055.

[20] I. A. Sucan and S. Chitta, "Moveit!" *Online at http://moveit.ros.org*, 2013.

[21] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 345–358, 1989.

[22] B. Walzel, H. Brunner, and M. Hirz, "Requirements on Petrol Stations in Year 2025," in *14. Symposium Energyinnovation, Graz, Austria*, Feb 2016.

A Visual Servoing Approach for a Six Degrees-of-Freedom Industrial Robot by RGB-D Sensing

Thomas Varhegyi, Martin Melik-Merkumians, Michael Steinegger, Georg Halmetschlager-Funek, and Georg Schitter

***Abstract*— A visual servoing approach is presented that uses depth images for robot-pose estimation utilizing a marker-less solution. By matching a predefined robot model to a captured depth image for each robot link, utilizing an appropriate approximation method like the Iterative Closest Point (ICP) algorithm, the robot's joint pose can be estimated. The a-priori knowledge of the robot configuration, alignment, and its environment enables a joint pose manipulation by a visual servoed system with potential to collision detection and avoidance. By the use of two RGB-D cameras a more accurate matching of the robot's links is feasible while avoiding occlusions. The modeled links are coupled as a kinematic chain by the Denavit-Hartenberg convention, and are prevented from divergence during the matching phase by the implementation of an algorithm for joint pose dependency. The required joint orientation of the robot is calculated by the ICP algorithm to perform a pose correction until its point cloud align with the model again. First tests with two structured light cameras indicated that the recognition of the robot's joint positions brings good results but currently only for slow motion tasks.**

I. INTRODUCTION

The fourth industrial revolution involves the use of new robotic technologies for smart and efficient work-flows in an innovative way. Humans will work together with robots side-by-side and integrate them in their every day work life as a collaborative device. Therefore, a collision detection with humans and the environment has to be established, for instance, with pressure sensitive skins [1, 2] or abnormal force recognition [3, 4] which are two approaches for a collaborative aspect. Another idea is the integration of visual perception [5, 6]. Robots should see where they are, know and see the environment they move in and know how they can grab and move without disturbing the work-flow. The focus of this paper lies on the application of computer/-machine vision methods for image processing and robot actuation. Vision-based motion control of robots is called visual servoing, where the robot manipulator is operated by the evaluation of visual information from an eye-to-hand (camera fix to workspace position) or an eye-in-hand (camera attached to robot) composition [7]. Figure 1 shows the recording of a robot in an eye-to-hand composition, that is used for the visual servoing approach in this paper. The advantage of visual servoing is that the teach-in procedure of a robot can be omitted since tool-tip-pose errors caused by low accuracy between the tool-tip-pose and the joint angle can be corrected in addition. These visual information can be exploited as position- or image-based information [8–10]. Position-based detection uses interest-points in the image to detect the object position, while image-based detection uses a template image of the designed object to predict how the camera should be aligned to the object.

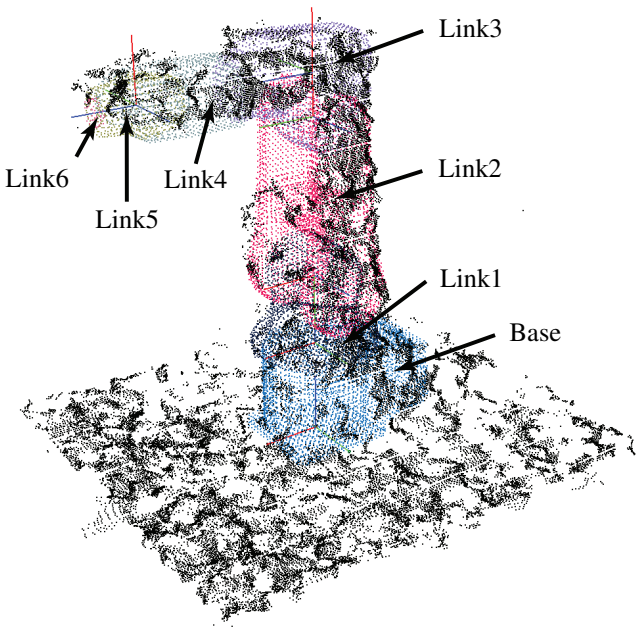
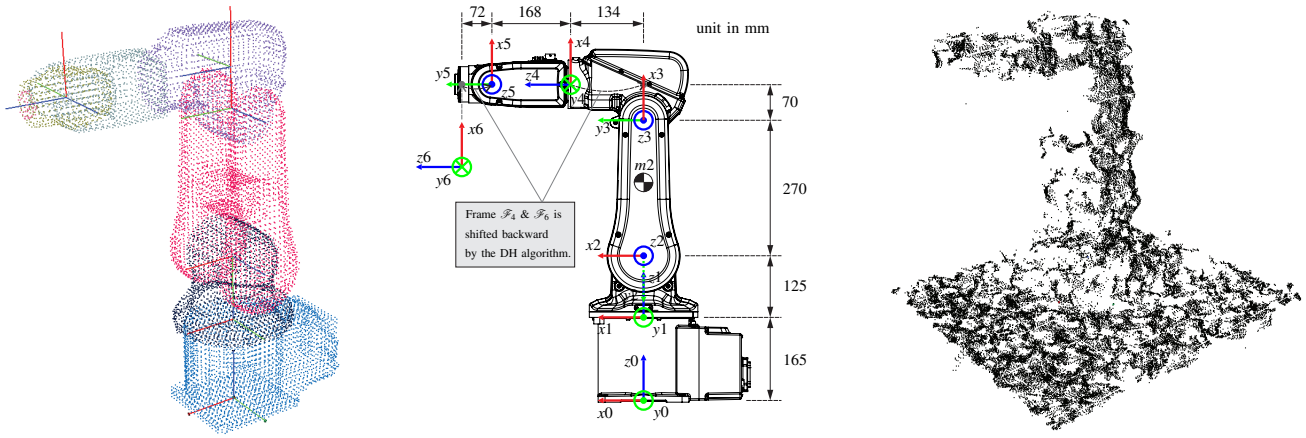


Fig. 1: ABB IRB 120 point cloud model overlaid by the captured point cloud from the Intel® RealSense R200.

So far, mainly 2D cameras have been applied for visual servoing applications [11–13]. The accuracy of the interest point estimation in the image as edges or corners determines how precisely the robot can be positioned by 2D cameras. For objects without distinctive characteristics as curved shapes without edges, these kinds of camera systems do not suite perfectly. In this case depth sensing cameras is the better choice.

RGB-D imaging systems can be separated into three main groups. First, stereo vision systems [14] which are based on two cameras and feature disparity where the depth information is obtained by the use of triangulation. Second, structured light cameras [15, 16] with the same basic principles as stereo vision cameras but instead of the second camera a projector is used. It emits a patterned light (usually infra-red light) and measures the disparity of the

All authors are with the Automation and Control Institute (ACIN), Vienna University of Technology, A-1040 Vienna, Austria. Contact: melik-merkumians@acin.tuwien.ac.at (corresponding author)



(a) (b) (c)

Fig. 2: (a) ABB IRB 120 point cloud model - the (joint) frames are set-up by DH convention. Each link is implemented as an independent model but coupled to its neighbor over the respective DH transformation; (b) ABB IRB 120 kinematic structure; (c) Robot depth image captured by two Intel® RealSense R200 in a distance of 1.5 m.

captured pattern in comparison to the original one to get the depth information by triangulation. Third, Time-of-Flight (ToF) cameras [17] where the depth information is measured over the elapsed time of pixel-wise emitted modulated light signals reflected by the detectable object.

RGB-D cameras provide point clouds (with position information in $\mathbb{R}^3$) generated from depth data. Thus, it is not necessary to seek for interest points for orientation estimation since the detected objects are already available as 2.5D objects in the workspace. Similar to an image-based approach a 3D model of an object can be matched to the point cloud via the Iterative Closest Point (ICP) algorithm [18–20] to find its alignment. It minimizes the distance between two point clouds with the requirement that the two point clouds are roughly close to each other (the initial guess), until they are aligned. The ICP algorithm consists of the following phases:

- **Selection** of point pairs,
- **Matching** of these point pairs,
- **Rejection** of point pairs due to individual consideration,
- **Error metric** assignment,
- **Minimizing** the error metric.

With the ICP algorithm, an alignment can be achieved within a few iterations.

Now, the idea is, instead of matching the whole robot as a rigid body, to split the robot into its links and match them separately (cf. Figure 1) in an eye-to-hand composition, such that the orientation of its joints can be estimated. In this case the use of markers can be omitted since the joint orientation can be calculated from the alignment of the links to each other, which makes this approach a versatile applicable method for industrial applications. The knowledge of the robot's kinematic chain gives the possibility of robot pose variation by well-defined joint orientations as well as the variation of the joint orientation during motion to correct the trajectory in case of work-flow disturbance. The goal of this approach is a visual servoing concept by depth sensing with a potential to collision protection and avoidance in a collaborative applicable manner.

This paper is organized as follows. In Section II, the applied method is described. The implementation of the robot's link point cloud models is described in Section III. The description of the setup and the camera alignment is described in Section IV. In Section V, the presented work is summarized and Section VI concludes the paper.

II. METHODS & APPROACH

The goal of the presented approach is to track a manipulator with six Degrees-of-Freedom (DoF) by two RGB-D imaging systems for joint position perception and visual servoing. For the measurement of the robot's joint alignment, the cameras are placed in an eye-to-hand composition. This allows to capture the whole manipulator from a wider view and avoid occlusions. The depth sensing technology with the highest accuracy for positioning and object matching is derived by comparing two different camera technologies. Therefore, a structured light camera and a ToF camera is applied and tested. Before the pose of the robot can be estimated, the position of both cameras have to be extrinsically calibrated, to get a perfect aligned point cloud from both cameras. The camera calibration is carried out as a transformation of the camera coordinate system by its physical position relative to the robot's base coordinate system.

For a matching process of point clouds by an appropriate approximation method like ICP to receive the robot's joint positions as mentioned in Section I, the models of its links, generated from Computer-Aided Design (CAD) files, have to be prepared. This is done by aligning the link models in the CAD files in their initial position as shown in Figure 2a and

TABLE I: Denavit-Hartenberg parameter

Name	Symbol	Description
joint angle	θ_i	angle between x_{i-1} and x_i about z_{i-1}
link offset	d_i	distance between the origin of frame $\mathscr{F}_{i-1}$ and $\mathscr{F}_i$ along z_{i-1}
link length	a_i	offset between frame $\mathscr{F}_{i-1}$ and $\mathscr{F}_i$ along x_i
link twist	α_i	angle between z_{i-1} and z_i about x_i

subsequently the generation of point cloud representation. The dependence of each links pose to each other in the model will be set-up by applying the Denavit-Hartenberg (DH) convention to achieve the kinematic chain as shown in Figure 2b. The DH convention describes the transformation between two frames of a manipulator by a homogeneous transformation matrix ${}^{i-1}\boldsymbol{T}_i \in \mathbb{R}^{4\times4}$ with four parameters by placing the joint coordinate frames in a predefined way. These transformations are represented by four basic transformations between the joints as a chain of two rotations and two translations

$$ {}^{i-1}\boldsymbol{T}_i = \mathrm{Rot}_{z_{i-1},\theta_i}\mathrm{Trans}_{z_{i-1},d_i}\mathrm{Trans}_{x_i,a_i}\mathrm{Rot}_{x_i,\alpha_i} \ , \qquad (1) $$

with the DH parameters listed in Table I.
This convention will simplify the calculation effort for matching via the ICP algorithm to only one DoF per joint and keep the links dependent from each other. The deviation from the robot's point cloud to the model is used for the calculation of the joint velocities to align both point clouds again. The whole implementation is realized with the free Point Cloud Library (PCL) [21], which includes numerous algorithms for handling of n-dimensional point clouds and three-dimensional geometries, in the framework of the Robot Operating System (ROS) [22]. ROS is a collection of libraries, tools and conventions for writing robot operating software.

III. MODEL IMPLEMENTATION

In an initial step point clouds from the CAD models of the robot's links have to be generated. It is important that, before the point clouds can be generated, the alignment of the CAD modeled links are prepared correctly as mentioned in Section II. First, they have to be aligned in their initial direction (cf. Figure 2b), second, their coordinate system must be set to the center of their rotation axis, and third, the link coordinate systems have to be translated such that they match with the DH convention as it is done for link four and six (translation in x direction) as shown in Figure 2b. The point clouds are generated by the tool *pcl_mesh2pcd* (based on *take views and fuse them together*) from the PCL to achieve an envelope point cloud of the CAD models.

Every link is implemented as an own object with the properties summarized in Table II, with the first four parameters as constants and the transformation matrix and joint angle as variables. The robot's links will not separate from each other during the ICP algorithm performs the matching, since they

TABLE II: Robot link properties

Parameter	Class
Name	std::string
Point cloud	pcl::PointCloud<PointXYZRGBA>*
Color	pcl::visualization::PointCloudColorHandlerCustom<PointXYZRGBA>*
DH-parameter	std::vector<double>
DH-transformation matrix	Eigen::Matrix4f
Joint angle	std::double

are coupled by the transformation of DH with the parameters of Table III and the dependencies given by

$$ {}^{0}\boldsymbol{T}_n = \prod_{i=1}^{n} {}^{i-1}\boldsymbol{T}_i \ , \qquad (2) $$

$$ \boldsymbol{T}_{n,\alpha} = {}^{0}\boldsymbol{T}_n \boldsymbol{T}_\alpha {}^{n}\boldsymbol{T}_0 \ , \qquad (3) $$

$$ {}^{n-1}\boldsymbol{T}_n = {}^{n-1}\boldsymbol{T}_0 \boldsymbol{T}_\alpha {}^{0}\boldsymbol{T}_n \ . \qquad (4) $$

In Equation (2), ${}^{0}\boldsymbol{T}_n \in \mathbb{R}^{4\times4}$ is the transformation of joint n between the base coordinate system and the coordinate system of joint n as the product of the DH transformations. By the use of the short notation $\mathrm{c}(\cdot) = \cos(\cdot)$ and $\mathrm{s}(\cdot) = \sin(\cdot)$, the DH transformation matrix ${}^{i-1}\boldsymbol{T}_i$ from Equation (1) can be written as

$$ \begin{aligned} {}^{i-1}\boldsymbol{T}_i &= \begin{bmatrix} {}^{i-1}\boldsymbol{R}_i & {}^{i-1}\boldsymbol{p}_i \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{c}_{\theta_i} & -\mathrm{s}_{\theta_i}\mathrm{c}_{\alpha_i} & \mathrm{s}_{\theta_i}\mathrm{s}_{\alpha_i} & a_i\mathrm{c}_{\theta_i} \\ \mathrm{s}_{\theta_i} & \mathrm{c}_{\theta_i}\mathrm{c}_{\alpha_i} & -\mathrm{c}_{\theta_i}\mathrm{s}_{\alpha_i} & a_i\mathrm{s}_{\theta_i} \\ 0 & \mathrm{s}_{\alpha_i} & \mathrm{c}_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \ , \end{aligned} \qquad (5) $$

with ${}^{i-1}\boldsymbol{R}_i \in \mathbb{R}^{3\times3}$ the rotation between the frame $\mathscr{F}_{i-1}$ and $\mathscr{F}_i$, the translation ${}^{i-1}\boldsymbol{p}_i \in \mathbb{R}^{3\times1}$ from the origin $\mathscr{O}_{i-1}$ to $\mathscr{O}_i$ and the vector of zeros $\boldsymbol{0} \in \mathbb{R}^{3\times1}$. $\boldsymbol{T}_{n,\alpha} \in \mathbb{R}^{4\times4}$ in Equation (3) is the transformation of joint n in the base coordinate system by the transformation matrix

$$ \boldsymbol{T}_\alpha = \begin{bmatrix} \boldsymbol{R}_z(\alpha) & \boldsymbol{0} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \ , \qquad (6) $$

with the rotation matrix $\boldsymbol{R}_z(\alpha) \in \mathbb{R}^{3\times3}$ along the joint rotation axis which is obtained from the Euler angles by the ICP

TABLE III: Denavit-Hartenberg parameters of the industrial robot ABB IRB 120

JointNr.	θ_i [°]	d_i [mm]	a_i [mm]	α_i [°]
1	q_1	165	0	0
2	q_2	125	0	$-\pi/2$
3	$q_3 - \pi/2$	0	270	0
4	q_4	0	70	$-\pi/2$
5	q_5	302	0	$\pi/2$
6	q_6	0	0	$-\pi/2$

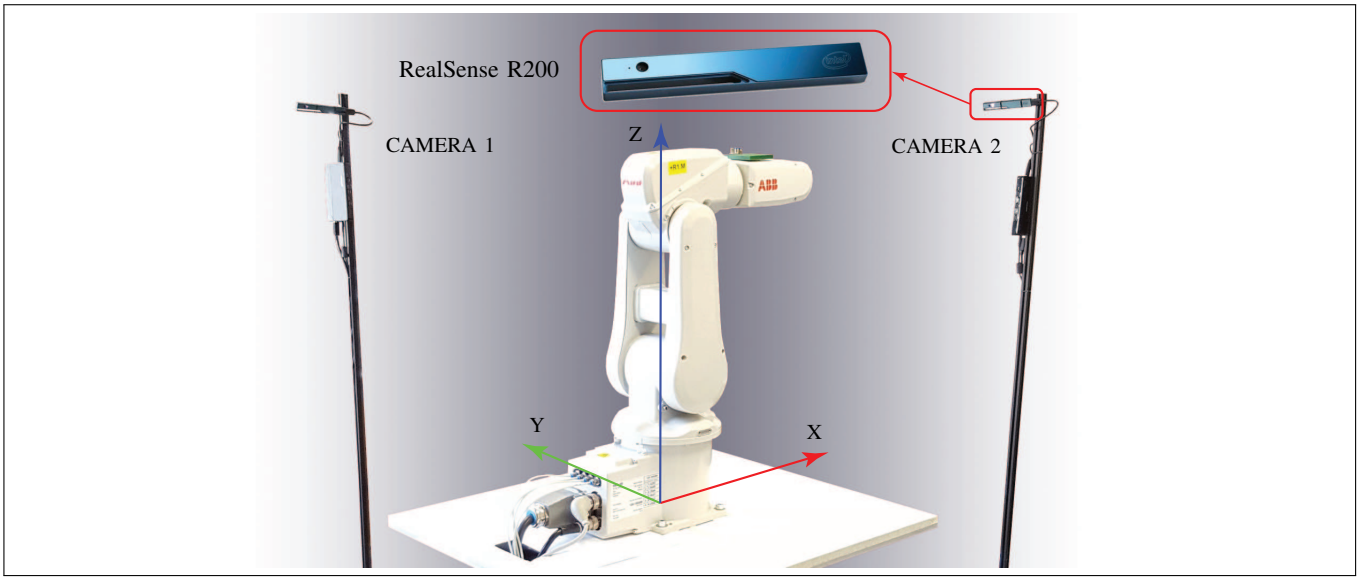


Fig. 3: Experimental setup with two Intel® RealSense R200 structured light cameras in 90° alignment to an ABB IRB 120.

algorithm. $\boldsymbol{R}_z(\alpha)$ performs a roll, pitch, or yaw rotation about the angle α according to the joint rotation axis. The new DH-transformation matrix ${}^{n-1}\boldsymbol{T}_n \in \mathbb{R}^{4\times4}$ of Equation (4) will be saved after the rotations have been performed. By iteration of these equations every rotation $\boldsymbol{R}_z(\alpha)$ of a joint n will be passed to the following joints. This guarantees that every joint keeps coupled to each other.

IV. SETUP & CAMERA ALIGNMENT

Two identical structured light cameras (Intel® RealSense R200) are used in the setup (cf. Figure 3) to avoid occlusions and to get a denser point cloud representation of the robot as shown in Figure 2c. The cameras are positioned in 90° to each other. This angle has been chosen since the influence of the illumination disturbance by the projected structured lights is minimized. Each camera is placed 65 cm above the robot with a pitch angle of 30° down to have a wider view. The extrinsic camera calibration will be performed through a plane calibration. Therefore, the table where the robot is placed on has been detected by the outliers' detection method Random Sample Consensus (RANSAC) to receive the model coefficients of the plane $\mathscr{A}_{xy}$. With the model coefficients, the dihedral angle between the plane normal and camera image normal can be derived by the equation

$$cos(\varphi) = \frac{\vec{n_1} \cdot \vec{n_2}}{|\vec{n_1}| \cdot |\vec{n_2}|} \tag{7}$$

where $\vec{n_1} = (a_1, b_1, c_1)$ is the normal vector of the plane $\mathscr{A}_{xy}$ in z direction and $\vec{n_2} = (a_2, b_2, c_2)$, the normal vector of the camera image plane $\mathscr{A}_{yz}$ along the x direction with the plane coefficients a_i, b_i, c_i for $i = 1, 2$. The cameras are aligned by the rotation with φ from Equation (7) (plus the camera pitch angle) and the known translation from the robot's base.

V. EVALUATION & RESULTS

The test system, which is used to evaluate the proposed approach, consists of a personal computer with an Intel® Core™ i5-3470 @ 3.20 GHz, 4096 MB RAM, and a GeForce GT 630 with the operation system Linux Ubuntu 16.04 @ 64 bit.

TABLE IV: The parameters used for the Iterative-Closest-Point algorithm

Max. Correspondence Distance	Max. Iterations	Transformation Epsilon	Euclidean Fitness Epsilon
0.003 m	100	1e-8 m	5e-4 m

So far, the structured light cameras and the robot motion communication are implemented successfully in ROS. The cameras and the robot are launched as ROS nodes such that they can communicate with each other. The Point cloud models of the robot's links are generated from CAD files and coupled together via the DH convention such that they depend on each other and that a rotation of joint one, for instance, has an effect to the other joints (cf. Equations (2) to (4)). A visualization is implemented to visualize the model together with the captured depth image as shown in Figure 1. The joint positions and alignments from the implemented model are observable and controllable. Since the ICP algorithm needs an initial guess where it should start the matching, an initial robot position for program start has been chosen as shown in Figure 2a, otherwise a correct estimation of the position would be hardly possible. In the first experiment the built-in ICP algorithm from the PCL has been tested with the parameters from Table IV and structured light cameras with moderate results. While for the initial pose (start pose) reasonably accurate joint angles with $\pm0.5°$ have been measured, the deviation increased up to $\pm5°$ during motion. These evaluation results were obtained for slow motion tasks ($\leq 1\,°/s$). For faster movements the ICP algorithm is not able to finish the required number of

iterations on the test system and the point cloud model can not be matched. The low accuracy of the ICP algorithm in the experiments for low speeds may occur to the very bumpy surface images from the cameras (Figure 1), which makes it difficult to calculate an accurate match. A smoothing of the robot's point cloud by the moving least squares method from the PCL also does not significantly improve the results, since the outliers' in the robot's point cloud surface are too large (cf. Figure 2c) to achieve good results.

VI. CONCLUSION & OUTLOOK

A robot point cloud model generated from CAD data for each robot link have been adopted and linked via the DH convention. A linked motion algorithm is integrated so that each link depends from each other. The first tests with structured light cameras and the ICP algorithm from the PCL showed moderate results. For the next tests with structured light cameras, the results should be improved by the implementation of a Levenberg-Marquardt Optimizer [23, 24] for an optimized registration. The change of the camera system to ToF cameras will also bring better results with the general ICP algorithm. So far the operation area is limited by only two cameras, because the robot's tool center point is not detectable overall by reason of occlusions in negative y-direction. A remedy would be to place a third camera right from the robot. This is feasible with a ToF camera but challenging with a structured light camera due to illumination disturbance from the counterpart. An alignment of 60 degrees for three structured light cameras would be better, since all the three cameras would receive the same disturbance which is less than if two of three fully receive it. A faster and more general model implementation would bring the implementation of an automatic model generation from COLLAborative Design Activity (COLLADA) [25] data which can be generated easily by CAD programs. With the COLLADA data (version 1.5.0) not only the geometry parameter would be loaded, the mechanical parameter as mass, inertia and center of mass could be loaded too, which is interesting for the robot dynamic. This would remove the model preparation as mentioned in Section III for a more user-friendly application.

REFERENCES

[1] J. O'Neill, J. Lu, R. Dockter, and T. Kowalewski, "Practical, stretchable smart skin sensors for contact-aware robots in safe and collaborative interactions", in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 624–629.

[2] C. Liu, Y. Huang, P. Liu, Y. Zhang, H. Yuan, L. Li, and Y. Ge, "A flexible tension-pressure tactile sensitive sensor array for the robot skin", in *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, IEEE, 2014, pp. 2691–2696.

[3] A. De Luca and R. Mattone, "Sensorless robot collision detection and hybrid force/motion control", in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, IEEE, 2005, pp. 999–1004.

[4] K. Kosuge and T. Matsumoto, "Collision detection of manipulator based on adaptive control law", in *Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, 2001, pp. 117–122.

[5] C. Morato, K. N. Kaipa, B. Zhao, and S. K. Gupta, "Toward safe human robot collaboration by using multiple kinects based real-time human tracking", *Journal of Computing and Information Science in Engineering*, vol. 14, no. 1, p. 011 006, 2014.

[6] B. Schmidt and L. Wang, "Depth camera based collision avoidance via active robot control", *Journal of Manufacturing Systems*, vol. 33, no. 4, pp. 711–718, 2014.

[7] A. Muis and K. Ohnishi, "Eye-to-hand approach on eye-in-hand configuration within real-time visual servoing", *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 4, pp. 404–410, 2005.

[8] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of basic visual servoing methods", *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, 2011.

[9] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: Keeping the object in the field of vision", in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, IEEE, vol. 2, 2002, pp. 1624–1629.

[10] T. Koenig, Y. Dong, and G. N. DeSouza, "Image-based visual servoing of a real robot using a quaternion formulation", in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, IEEE, 2008, pp. 216–221.

[11] E. Marchand and F. Chaumette, "Visual servoing through mirror reflection", in *IEEE Int. Conf. on Robotics and Automation, ICRA'17*, 2017.

[12] D. Tsai, D. G. Dansereau, T. Peynot, and P. Corke, "Image-based visual servoing with light field cameras", *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 912–919, 2017.

[13] N. Shahriari, S. Fantasia, F. Flacco, and G. Oriolo, "Robotic visual servoing of moving targets", in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 77–82.

[14] H. Liu, S. Huang, N. Gao, and Z. Zhang, "Binocular stereo vision system based on phase matching", in *SPIE/COS Photonics Asia*, International Society for Optics and Photonics, 2016, 100230S–100230S.

[15] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review", *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[16] S. K. Nayar and M. Gupta, "Diffuse structured light", in *Computational Photography (ICCP), 2012 IEEE International Conference on*, IEEE, 2012, pp. 1–11.

[17] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (ToF) cameras: A survey", *IEEE Sensors Journal*, vol. 11, no. 9, pp. 1917–1926, 2011.

[18] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes”, in *Robotics-DL tentative*, International Society for Optics and Photonics, 1992, pp. 586–606.

[19] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm”, in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, IEEE, 2001, pp. 145–152.

[20] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, “CAD-model recognition and 6DOF pose estimation using 3D cues”, in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, IEEE, 2011, pp. 585–592.

[21] R. B. Rusu and S. Cousins, “3D is here: Point cloud library (PCL)”, in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011.

[22] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source robot operating system”, in *ICRA workshop on open source software*, Kobe, vol. 3, 2009, p. 5.

[23] S. Roweis, “Levenberg-Marquardt Optimization”, *Notes, University Of Toronto*, 1996.

[24] Y. Wu, W. Wang, K. Lu, Y. Wei, and Z. Chen, “A new method for registration of 3D point sets with low overlapping ratios”, *Procedia CIRP*, vol. 27, pp. 202–206, 2015.

[25] M. Barnes and E. L. Finch, “COLLADA - digital asset schema release 1.5.0”, *Sony Computer Entertainment Inc. Atazadeh, Sam Amirebrahimi, Alireza Jamshidi (7048) 3D-Cadastre, a Multifaceted Challenge*, 2008.

Toward Safe Perception in Human-Robot Interaction

Inka Brijacak, Saeed Yahyanejad, Bernhard Reiterer and Michael Hofbaur[1]

Abstract— **Perception is a major component of a system when it comes to the concept of safety in human-robot interaction. Although designing a mechanically safe robot may reduce lots of potential hazards, it is still beneficiary or even required to have detailed knowledge of the current status of the robot, human, and other environmental entities. We refer to this knowledge as perceptional awareness, or simply perception, that subsumes: (i) what our system perceives from robot state and its environment, (ii) what our system perceives from human state, and (iii) what a human perceives from the robot state. In this paper we provide requirements for a holistic architecture to construct safe perception using multiple heterogeneous and independent sensors and processing units in any environment that includes both robots and humans. We also illustrate our concepts on the basis of particular instances of this scheme realized in the robotic lab.**

I. INTRODUCTION

Nowadays, robots are being used widely in different fields due to their precision, accuracy, reliability, and easy deployment. In many initial applications of robots, they are functioning separated from humans in isolated areas. With advances of technology and the necessity for coexistence of robots and humans (e.g., medical application, service robots, collaborative production lines), the new era of human-robot interaction (HRI) has emerged. HRI studies and describes the types and characteristics of the possible interactions that can exist between a robot and a human.

When a human is working in a close distance with robots, the safety of the human becomes an important issue. Initially, safety requirements for many industrial robotic applications were achieved just by a physical separation of humans from any robot (e.g., using barriers or fences). This simple and effective way to impose safety, however, prevents direct interaction between humans and robots to work collaboratively. The relevant international standard for safety in industrial robots [10], [11], which specifies accepted means to impose safety, however, allows also human-robot collaboration in four clearly defined scenarios. The new technical specification ISO TS 15066 "Robots and robotic devices – Collaborative robots" [12] provides even more details on these operational settings and specifies comprehensive force, pressure, and speed limits for unintended human-robot interactions (collisions).

Risk reduction during human-robot interaction has three main approaches: (i) redesigning the system and the task realization, (ii) using functional or physical safeguards, and (iii) raising the awareness of the operator/user, either using active warnings during operation and/or by specific training. Taken into account the industrial experience, redesigning the system is the most effective risk reduction strategy and should always be applied first. However, when operating adaptively in less structured environments, redesign alone is often insufficient, and additional functional safety measures are mandatory [13].

It is possible to combine these three approaches to achieve higher levels of safety. In spite of that, no matter how accurate a system is designed, the continuous monitoring (the second approach mentioned above) is an important factor for a safe system. To be able to understand the status of the environment or a system, the concept of perception plays an important role. Similar to human perception, the concept of the perception for a system can be twofold:

- External: What a system sees, perceives, or understands from the environment, i.e., what types of object are around me? What are their positions, speed, shape, size? What are the states of other systems around me?
- Internal: What a system sees, perceives, or understands about itself, i.e., where should I be? Where am I? What is my current state?

For both of these perception types, we need dedicated sensors to obtain relevant data upon for perception. This demanding task requires to deal with the following issues:

- sensory data acquisition and storing the data
- data mining, enhancement, and filtering
- sensor fusion
- time synchronization
- dependable, safety-enabled operation.

The complexity highly increases with the larger number of heterogeneous sensors such as, safety-enabled laser scanners (LIDARs), RGB cameras, thermal cameras, time-of-flight (ToF) cameras, haptic sensors, proximity sensors, ultrasonic sensors, robot internal sensors (e.g., torque sensors), pressure sensors, etc. Redundancy achieved by using diverse sensor types highly improves the reliability of the overall perception unit. Dealing with diverse sensors requires one to carefully consider the different interfaces, data types, sampling rates and, of course, a potentially large amount of data. In order to deploy such an inclusive perception scheme in real-world robotic systems, however, it is also important to consider the requirements set by the relevant standards that include the entire life cycle of the system starting with the development process itself, hard- and software-requirements and functional issues for all forms of the system's operation. This goes far beyond the requirements necessary to realize a laboratory demonstrator. As a consequence, it is helpful to

[1] All authors are with JOANNEUM RESEARCH ROBOTICS - Institute for Robotics and Mechatronics - Cognitive Robotics Group, Austria <firstname.lastname>@joanneum.at

consider these aspects early in the R&D efforts in order to qualify for real-world applications.

II. Related Work

With the growing applications of robotics in human life and considering the high importance of human safety in HRI, more and more research is being dedicated to assure a safe perception in collaborative environments. Kulić [13] was one of the first who has provided an extended and detailed overview regarding the safety in HRI. She managed to define many important terms in this scope and formulate a metric for danger measurement. [20] also provides a quick overview of safety issues in HRI. The work done in [16] categorizes the safety strategies in three categories: I) crash safety (e.g., limiting the power/force), II) active safety (e.g., by using proximity sensors, vision systems and force/contact sensors), and III) adaptive safety for intervening in the robot operation and applying corrective actions that lead to collision avoidance without stopping the robot operation. In their work and also the work from [6] the focus is more on the design principles. The latter also considers robustness, fast reaction time, and context awareness as the main parameters of a safe design. One interesting and genuine idea mentioned there and originally in [7] is the recommendation to design the robots such that they are predictable for a human. For instance, by using special sounds or human-like movements for a robot, the human can expect and foresee the robot's moves and accordingly avoid unintended collisions with the robot.

Some other researchers just focus on detecting and localizing the human and accordingly prevent the robot from colliding with it. Depending on the type of the detecting sensor, their performance is evaluated. Active or marker-based sensors may be more challenging to implement and less convenient to apply in real scenarios, but they can provide a quite accurate and reliable collision avoidance. Their proximity distances can reach up to a few centimeters between human and robot [14]. On the other hand, other types of sensors such as cameras and laser scanners may have higher error ranges, but by combining multiple sensors together we can minimize the risk. In this direction, [18] fuses data from multiple heterogeneous 3D sensors to detect any moving object approaching the robot. Similar work has been done by [19], which constructs point clouds and 3D models of the moving objects and the robots in order to avoid collisions.

Safety of a human is not always achieved by immediate protection from danger or collision. Sometimes hazards can be results of long-term inappropriate actions in HRI. For instance, [4] looks at the human safety from an ergonomic aspect, which is a complementary point of view. They consider a work environment which ensures the occupational safety and describe the requirements for a workplace where human and robot can jointly perform an assembly process without separation between their workspaces. They also consider some human factors such as the age of the working person.

In our work we are going to look at the safe perception in HRI with a holistic approach. We are going to explain what kinds of criteria are necessary to be obeyed in order to have a safe perception architecture and why a single safety precaution will fail.

III. Risk analysis in safe perception

In the design of a robot system, risk assessment is a main measure for achieving standards-compliant safety. The general process, consisting of risk analysis and risk evaluation, is described in [9], with extensions specific to robots given in [11] and [12]. First, the potential hazards of the robot system during all phases of its life cycle are to be identified. The hazards given in the annex of [11] may be seen as a list of examples, which must always be considered and carefully examined with the specific robot system and its application/task in mind. All identified hazards are then to be evaluated in terms of their risks. From the obtained results it may become clear that the risks have to be reduced by certain measures, leading also to updates in the results of risk analysis and evaluation, and thus a new iteration of the process steps. Eventually the residual risks of any remaining hazards are sufficiently low to allow for the designed robot system to be realized and considered safe.

Risk assessment and safe perception influence each other in several ways. Already during the risk analysis, the capabilities gained from the perception infrastructure can serve as measures that counter certain hazards and reduce risks. But the risk analysis must also consider any potential hazards arising from system components, including also those that are specifically employed for safety reasons. If the used components do not provide a sufficiently high integrity / performance level or they are placed or configured in suboptimal ways, their total effect may be deteriorating. However, such choices will typically be identified and mitigated in the further course of the iterative risk assessment and risk reduction, so that the final solution is able to achieve the required safety properties. When modifying the system design to achieve risk reduction, the integration of the safety perception infrastructure or the modification of its integration can be a central measure. Thus, as one of its results, the iterative process of risk assessment and risk reduction gives constraints on effective sensor placement that enables comprehensive sensor fusion later on. An example of such an improvement can be seen in the step from the arrangement depicted in Fig. 3 to the one in Fig. 5. In the running system, any residual risks are permanently relevant. The setup of the safe perception must be designed in a way that potential hazards that could not be eliminated by system design can be prevented or dealt with accordingly based upon perception.

Finally, the operation of the system continuously gives opportunities to gather new knowledge that can be used in a refined risk assessment to further improve the system's safety. This could be done any time, but is necessary in particular when the system is going to be modified. Possible inputs may come from user feedback, other persons observing the operation, or also the system itself. For the latter, we envision a component that is able to identify events that

may need further offline analysis later using different kinds of potentially related data available to the system.

IV. Safe perception architecture requirements

An architecture for a safe perception system typically includes components of the types machine, sensor, human, and processing unit. To construct a suitable architecture, we need a good understanding of these components in terms of their functionality and reliability as well as their relations and interfaces. Here we are going to propose a generic architecture by pointing out the requirements which enable the realization of a safe perception system for a typical collaborative robot system. This architecture should be independent from the robot type, size of the workspace, and environmental factors as much as possible and also easy to deploy. In order to achieve such a goal, we have to consider the possibilities of failure of individual components in a system as discussed in Section III. Accordingly, an ideal safe architecture considers/includes the following requirements:

- Embed safety inside different building blocks: consider safety not just as an add-on but embed it in each system component, robot, planning, and programming decisions. However, always keep the distinction between operational functionality and safety functionality in mind.
- A modular architecture: makes it easy to add/remove various hardware and software components. For instance, Robot Operating System (ROS) [17] has been used for our modular software architecture to provide a simple message passing and hardware abstraction.
- Adding parallel redundancy: use multiple sensors in parallel over independent platforms to make sure that the failure of one is not causing the whole system to fail.
- Heterogeneous system: using different types of sensors (e.g., laser scanner, time-of-flight camera, thermal camera, speech recognition, light curtain, etc.) to make sure that the system is robust against changing environmental variables. For example, if there are poor visibility conditions at the workplace, conventional cameras may fail to obtain a picture but a thermal or time-of-flight (ToF) camera can help and even provide images through fog or smoke.
- Reproducibility: which makes it easy to re-implement in different scenarios and setup the perception system in other new workspaces.
- Mapping the Environment: modeling the 3D environment in order to further simulate, localize and position the sensors and objects in the environment. This helps to decide how and where to mount the sensors to achieve the maximum coverage (high spatial distribution helps the robustness in case of local failures).
- Context aware: takes the context of the ongoing scenario into account either by receiving it from operator or by analyzing the scene. Accordingly the system adapts the parameters and decision-making to that specific scenario.
- Intelligent: learn from the previous situations (from both false-positives and false-negatives) and hence provide feedback data and parameter correction for future improvement. Using machine learning in robot perception is an example to achieve this goal.
- Exploiting human perception: warn the human about the potential hazards. Unlike the conventional sensory perception, we do not only inform the human in close-to-danger scenarios. Instead, we additionally count on human perception by constantly giving a feedback regarding the state of the robot to the human, for example by producing a sound according to the movements of the robot. This way the human herself/himself can make a decision if she/he feels something is out of the order.

As mentioned above, redundancy is a major design paradigm to realize safety through perception. Relevant standards such as the previously mentioned ISO 10218 and ISO 13849 enforce redundancy throughout the system for achieving a required performance level for a safety function, i.e. redundancy in sensors, computational units and actuators as indicated in Figure 1.

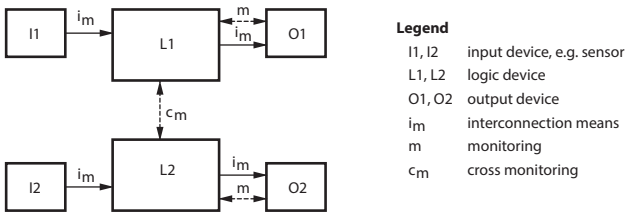


Fig. 1. Redundant Safety Architecture (cat. 3, ISO 13849-1 cl. 6.2.6)

This classic layout for achieving a high integrity / performance level has to be incorporated carefully as not to tamper with the safety of the overall system. This is important in particular as our complex robot system will involve both safety functionality at high integrity level as well as functional components with lower integrity level that should also contribute valuable information to improve the overall safety. In industry, one typically talks about *yellow* and *gray* components, referring to high integrity safety and general functional components, respectively. A clear structure, both in terms of hardware and software, is required in order to obtain the safety functionality at the desired performance levels.

V. Architecture realization

In our lab we have various types of serial robotic manipulators in workspaces where safe human-robot interaction or collaboration is compulsory. Therefore, we utilize sensors for highly dependable perception using safety LIDARS (yellow hardware – OMRON OS32c) at performance level D (PLd) [8]. On the other hand, we intend to use functionally powerful time-of-flight (ToF) cameras (gray hardware - PMD Pico Flexx) for environmental perception. Similarly, the control of the robots involve the low-level safety-enabled robot controllers (yellow hardware/software) in combination with a high-level control system that is implemented in ROS (gray

hardware/software). The overall system should not just act as a ROS system with add-on safety, but integrate safety inclusively.

We propose a safety-enabled system architecture that solves the safe robot perception and control task through 3 levels of hardware abstraction. Basis for this architecture that is given in Figure 2 is a safety-rated robot controller (in our case the KUKA Sunrise controller for the sensitive iiwa robot). High level control is implemented in ROS running on separate (Linux-based) controllers. In between those two control layers, we introduce a safety-rated controller (e.g. a safety PLC) that connects to both, safety-rated sensors (safety LIDARS in our case) and the safety-rated input of the low-level controller. This allows us to implement dependable safety functionality that goes beyond the simple safety-logic of the low-level controller. However, it might also be implemented directly on the low level controller if the device offers to implement high integrity safety functions. This layered model clearly defines a priority structure where the safety-enabled control system takes control whenever a critical safety issue is detected. Thus, there is no direct connection that allows the ROS System to issue control actions for the low level controller except the authorized connection through this safety control layer.

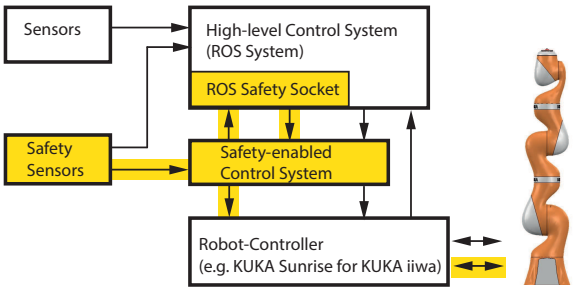


Fig. 2. Safety-enabled Architecture

Up to now, this structure resembles the classic add-on safety architecture. However, we intend to go beyond this architecture that will enable more inclusive perception and control schemes. As a consequence, we propose to provide a highly dependable ROS safety socket that connects the safety-controller to the ROS environment. Furthermore safety sensors could be connected to the ROS environment as well. For example our safety LIDARs provide safety-enabled outputs that define region interceptions through (safe) binary signals, whereas the more informative LIDAR scan is provided through standard interfaces to the ROS system. With our safety socket, we intend to enable ROS functionality not just at different levels of priority, but also at different levels of dependability. This safety-socket is only one pre-requisite. We also have to provide dependable and in particular trustworthy ROS nodes and communication between them and the socket. The standard ROS system does not address IT security adequately [15]. To compensate for this security flaw, our institute colleagues recently proposed a scheme for application-level security and safe communication [3], [1] for ROS that is now under consideration by the Open Source Robotics Foundation (OSRF) to be included in the SROS project for future public release.

Alongside of this implementation effort that will provide the necessary building blocks for a safety-rated perception and control functionality, we evaluated possibilities for functionally rich and safe multi-sensory perception using the standard ROS environment as an experimental testbed. We have set up a heterogeneous perception system comprising of two safety-rated OMRON OS32C laser scanners with data fusion running on two different computers and one or two ToF cameras for acquiring 3D data from the environment (the aforementioned PMD Pico Flexx camera and the single-beam ToF sensors Terraranger). We consider the proper combination of different technologies of parallel and independent sensors and the resulting high redundancy as a prerequisite for fulfilling safety requirements. Additionally, to achieve robustness in case of local failures, it is necessary to mount the sensors in a distributed way. As a basis for making safety-related decisions in the running system, we are going to define a distinction of three danger zones that are reported by our sensor fusion: *Danger*, *Warning*, and *Safe*. Their origin is in the origin of a robot, and they are surrounding the robot in a circular way. The border between *danger* and *warning* zones is defined using safety separation distance defined in ISO/TS 15066 [12]. Using distance of a moving object from a depth sensor, it will be decided in which danger zone the movement is detected.

The example setup of sensors which is shown in Figure 3 results the sensor fusion shown in Figure 4. Sensors are mounted close to each other, which leads to a higher chance for all sensors to fail together when a local hazard happens (e.g. physical damages). Knowing that, and also for a specific collaborative use case, sensors are mounted as shown in Figure 5. Regarding modular architecture and reproducibility, it is also very easy to change the mounting for other use-cases and workspaces. However, more automatized setup of sensors for maximum coverage of the workspace and their calibration is planned for the future work.

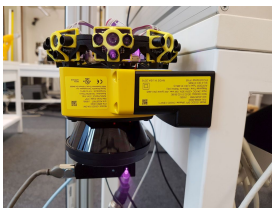


Fig. 3. Example of a problematic setup where 3 different types of sensors are mounted just next to each other. This setup increases the chance of perception failure due shadowing effects and local hazards such as physical damages.

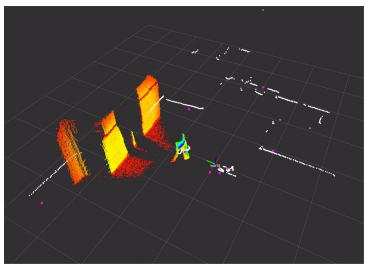


Fig. 4. Visualization of the 3D position data in RViz obtained from Teraranger Tower (8 pink points), Pico Flexx Camboard ToF camera (colored points), and laser scanner (white points).

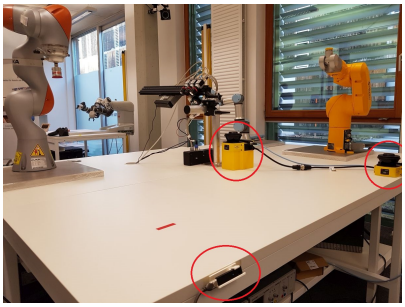


Fig. 5. Distributed setup of sensors including 2 laser scanners and one Pico Flexx ToF camera around the workplace.

The sensors we have chosen are not supposed to be used for object/human detection and localization per se, but mainly for distance measurement. Therefore, with both types of sensors, we need to perform some post-processing in order to be able to detect and perceive an approaching object. In order to have a safety-eligible sensory data analysis and decision making, we need to reduce the chance of false positive and false negative in our perception system. Safety-wise, perception scenarios with false negative (i.e., a human approaching the robot is not detected) are far more dangerous compared to scenarios with false positive. In case of false positive, on the other hand, we may observe instances of unwanted robot speed reduction or even a complete stop, which is affecting the system performance but not the safety property. For instance, in case of ToF camera the following steps are being performed to robustly detect a moving object:

- **Filtering** the depth image: it is performed by using various filtering method (e.g., median filter in both spatial and temporal domain) which mitigates the false detection. Filtering steps are shown in Figures 6 and 7, and resulting filtered depth image is shown in Figure 8.
- **Background image**: recording filtered depth image at startup. The background image is refreshed if there is no movement detected for a specific period of time (Figure 8).
- **Difference image**: subtraction of background and current depth image (Figure 8 – Figure 10 = Figure 12).
- **Blob Detection** in binary difference image (Figure 13). To avoid detecting changes produced by noise and also using the prior-knowledge of the size of an approaching object (e.g., human) we adjust the parameters of our blob detector (such as expected shape and size) in a way to detect only the intended moving targets. When at least one blob is detected, it means that there is a movement in the workspace, and therefore we can proceed with the next two steps.
- **Masking** the original depth image: binary difference image is used as a mask in order to have real depth data of each pixel of the blob that is assumed to be a moving object (Figure 11).
- **Final depth information** of detected moving object: is a result of using median value of depth info from the masked image. Higher importance is given to closer distances that still have a smaller covering area in the depth image, such as an intruding arm of a human.

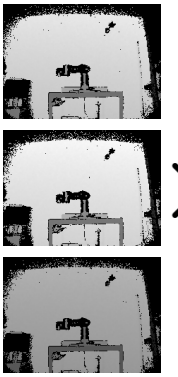


Fig. 6. Original depth images

Fig. 7. Filtered depth images

Fig. 8. Final background image

Fig. 9. Original images with human

Fig. 10. Final filtered depth image

Fig. 11. Masked original depth image

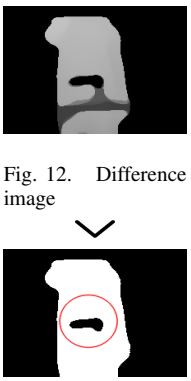


Fig. 12. Difference image

Fig. 13. Blob detection in BW diff. image

For the laser scanners, which provide 2D data (scan a plane or a cross-section out of the 3D space), the process of extracting distance of a detected object and its coordinates is aligned with the one of the ToF camera:

- **Background** data: resulting background data is median filter applied on temporal domain of data collected in initialization step.
- **Difference** data is calculated as subtraction of background data and current data every time stamp.
- **Movement** in the workspace is detected if the percentage of not moving points is less than 98.5%.
- **Transformation** of depth data from the laser coordinate system to the robot's coordinate system is done using Euclidean distance, taken into account the fixed position of the laser scanner relative to the robot.

Every time stamp we have the result of our sensor fusion as the final danger zone. From each sensor, regarding distance of a human, or any other moving object in robot's workspace, it is decided in which danger zone the detection happened, and the final danger zone is the worst case of all three. Measuring the separation distance between the object/human and the robot, in constant speed setting situations with worst-case value taken into account, it is ensured that the robot system never gets closer to the operator than the protective separation distance [12].

While it is essential to have a direct link from the safety-oriented sensor fusion to the robot control for adapting speed limitations or triggering an emergency stop, the combined information from the sensors also serves as a valuable input for generating task-level plans for the robot system. We use ROSPlan [2] as our infrastructure for task planning, which allows us to formulate the planning domain in the quasi-standard Planning Domain Definition Language (PDDL) [5]. The planner, given abstract logical models of the system and relevant entities in its surroundings on the one hand and goals to be achieved on the other, would typically generate sequences of actions such as picking up a certain object, placing it in a certain pose into the product that is being built, and fixing it there in a certain manner, using a certain path of motion trajectories from a set of possible ones. There could also be actions representing interaction with humans via user interface components or invoking arbitrary meaningful functionalities of connected devices.

The currently obtained safety zone information and other results of sensor fusion can be mapped to logical facts in the planning domain, and they in turn can be used in the conditions of PDDL actions in order to tie their applicability to the current safety situation. Examples for such conditional safety limitations include forbidding certain actions as a whole, forbidding trajectories in which parts of the robot would intrude certain zones or exceed a certain speed limit, forbidding interacting with potentially hazardous objects, or forcing the robot to assume a predefined home pose between any two other poses. The planning system takes care that such restrictions are not only considered when a new plan is generated but also that the current plan's execution is halted when an assumed precondition, safety-related or other, for a robot action is found to be not actually fulfilled, or when an action's execution was not successful. Then, starting from the updated current state, a new plan is generated and goes into effect.

VI. Conclusion

In this paper, we have emphasized the importance of a safe perception system in HRI scenarios where both human and robot coexist in a shared environment and collaborate toward their goals. We have taken into account a holistic approach toward safe perception and managed to introduce the requirements for a general architecture that integrates safety in any robotic environment independent of scenario, scale, shape, and the number of robots and humans. This architecture is modular, reproducible, context aware, intelligent and also has parallel redundancy, heterogeneous sensors, and embedded safety.

Furthermore we have presented how our safe perception is set up for a collaboration scenario in our lab to demonstrate the simplicity and reusability of our approach in real-world applications. In this demonstration multiple safety standards have been considered and included in order to have a correct risk analysis and safety-zone calculation.

References

[1] B. Breiling, B. Dieber, and P. Schartner, "Secure communication for the robot operating systems," in *Proceedings of the 11th Annual IEEE International Systems Conference*, 2017, to appear.

[2] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, A. Carrera, N. Palomeras, N. Hurtós, and M. Carreras, "Rosplan: Planning in the robot operating system," in *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015.*, 2015, pp. 333–341. [Online]. Available: http://www.aaai.org/ocs/index.php/ICAPS/ICAPS15/paper/view/10619

[3] B. Dieber, S. Kacianka, S. Rass, and P. Schartner, "Application-level security for ros-based applications," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4477–4482.

[4] M. Faber, J. Bützler, and C. M. Schlick, "Human-robot cooperation in future production systems: Analysis of requirements for designing an ergonomic work system," *Procedia Manufacturing*, vol. 3, pp. 510 – 517, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2351978915002164

[5] M. Fox and D. Long, "PDDL2.1: an extension to PDDL for expressing temporal planning domains," *J. Artif. Intell. Res. (JAIR)*, vol. 20, pp. 61–124, 2003. [Online]. Available: http://dx.doi.org/10.1613/jair.1129

[6] M. Giuliani, C. Lenz, T. Müller, M. Rickert, and A. Knoll, "Design principles for safety in human-robot interaction," *International Journal of Social Robotics*, vol. 2, no. 3, pp. 253–274, 2010. [Online]. Available: http://dx.doi.org/10.1007/s12369-010-0052-0

[7] M. Huber, H. Radrich, C. Wendt, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer, "Evaluation of a novel biologically inspired trajectory generator in human-robot interaction," in *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, Toyama, Japan, 2009, pp. 639–644.

[8] ISO, *ISO 13849-1:2006: Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*. Geneva, Switzerland: International Organization for Standardization, Nov. 2006.

[9] ISO, *ISO 12100:2010: Safety of machinery – General principles for design – Risk assessment and risk reduction*. Geneva, Switzerland: International Organization for Standardization, 2010.

[10] ISO, *ISO 10218-1:2011: Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots*. Geneva, Switzerland: International Organization for Standardization, July 2011.

[11] ISO, *ISO 10218-2:2011: Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration*. Geneva, Switzerland: International Organization for Standardization, July 2011.

[12] ISO, *ISO/TS 15066:2016: Robots and robotic devices – Collaborative robots*. Geneva, Switzerland: International Organization for Standardization, Feb. 2016.

[13] D. Kulić, "Safety for human-robot interaction," Ph.D. dissertation, University of British Columbia, 2006.

[14] P. A. Lasota, G. F. Rossano, and J. A. Shah, "Toward safe close-proximity human-robot interaction with standard industrial robots." in *CASE*. IEEE, 2014, pp. 339–344.

[15] J. McClean, C. Stull, C. Farrar, and D. Mascareas, "A preliminary cyber-physical security assessment of the robot operating system (ros)," in *Proc. SPIE*, vol. 8741, 2013, pp. 874 110–874 110–8. [Online]. Available: http://dx.doi.org/10.1117/12.2016189

[16] G. Michalos, S. Makris, P. Tsarouchi, T. Guasch, D. Kontovrakis, and G. Chryssolouris, "Design considerations for safe human-robot collaborative workplaces," *Procedia {CIRP}*, vol. 37, pp. 248 – 253, 2015.

[17] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, 2009.

[18] P. E. Rybski, P. Anderson-Sprecher, D. Huber, C. Niessl, and R. G. Simmons, "Sensor fusion for human safety in industrial workcells," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, 2012, pp. 3612–3619. [Online]. Available: http://dx.doi.org/10.1109/IROS.2012.6386034

[19] B. Schmidt and L. Wang, "Contact-less and programming-less human-robot collaboration," *Procedia {CIRP}*, vol. 7, pp. 545 – 550, 2013.

[20] M. Vasic and A. Billard, "Safety issues in human-robot interactions." in *ICRA*. IEEE, 2013, pp. 197–204.

OAGM Workshop

Pose Estimation of Similar Shape Objects using Convolutional Neural Network trained by Synthetic data

Kiru Park, Johann Prankl, Michael Zillich and Markus Vincze

***Abstract*— The objective of this paper is accurate 6D pose estimation from 2.5D point clouds for object classes with a high shape variation, such as vegetables and fruit. General pose estimation methods usually focus on calculating rigid transformations between known models and the target scene, and do not explicitly consider shape variations. We employ deep convolutional neural networks (CNN), which show robust and state of the art performance for the 2D image domain. In contrast, normally the performance of pose estimation from point clouds is weak, because it is hard to prepare large enough annotated training data. To overcome this issue, we propose an autonomous generation process of synthetic 2.5D point clouds covering different shape variations of the objects. The synthetic data is used to train the deep CNN model in order to estimate the object poses. We propose a novel loss function to guide the estimator to have larger feature distances for different poses, and to directly estimate the correct object pose. We performed an evaluation using real objects, where the training was conducted with artificial CAD models downloaded from a public web resource. The results indicate that our approach is suitable for real world robotic applications.**

I. INTRODUCTION

Pose estimation of objects in color and depth images is essential for bin-picking tasks to determine grasping points for robotic grippers. Man-made objects are usually manufactured using 3D CAD models having exactly the same shapes with negligible errors. The well-constrained environment enables the robot to identify each pose by comparing features of the pre-created template and an input image [14]. However, it is not possible to provide 3D CAD models for natural objects, such as vegetables or fish, where each object has a slightly different shape. Object pose estimation with template based approaches would need a huge number of templates in order to cover each individual pose and the different shape variants. Hence, these approaches would lead to large databases and a high processing time for matching of the templates.

Recently, CNN based approaches provide reasonable results for most computer vision tasks including image classification and object detection in 2D images [13] [15]. This achievement is accomplished with a large number of training examples, e.g., [4] [7]. The 2D image datasets are usually collected from web resource and annotated by non-expert persons with tools using a user-friendly interface. For RGB-D images or 2.5D point clouds it is difficult to collect a large number of examples from public web services and it is also hard to annotate the exact poses by non-expert persons. This results in a lack of training data and causes an additional complexity to train a CNN for estimating 6D poses in the 3D space. Therefore, pre-trained CNNs are used for extracting features from color or depth images, and the extracted features are used to train linear regressors to estimate the poses [16]. Although there are several datasets which have 6D pose information for more than 15K images [9], [10], it is still not enough to train a deep CNN and none of them consider object classes with large shape variations.

All authors are with the Vision4Robotics group, Automation and Control Institute, Vienna University of Technology, Austria {park, prankl, zillich, vincze}@acin.tuwien.ac.at

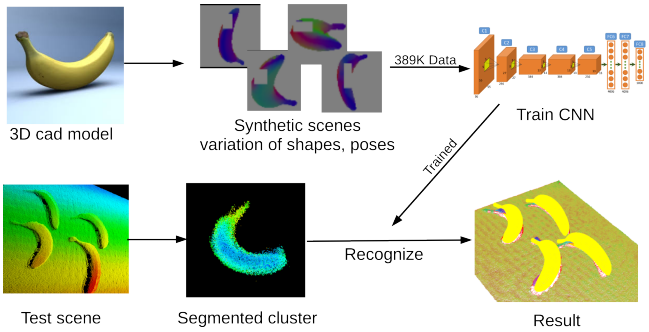


Fig. 1: Overview of the proposed framework. An artificial 3D CAD model is used to generate synthetic scenes with varied shapes and poses in order to train the deep CNN. The trained network can compute poses of each of segmented clusters.

In this paper, we propose a simple pose estimator that can be used to estimate poses of objects with shape variations, such as vegetables or fruit, using a CNN and a single depth image as input. Synthetic depth images containing various poses and shapes of a CAD model are generated to train the proposed CNN. No more template information is required after training. This simplicity is one of the advantages of the proposed model for the robust estimation of object poses with different shape variants. The experiments show that our concept is suitable for real world robotic applications.

As a summary, our paper provides the following contributions:

- We propose a framework that is able to generate synthetic training images and consists of a deep CNN pose estimator for the estimation of poses of natural object classes such as vegetables and fruit.
- Pairwise training is applied to train the deep CNN with a loss function that minimizes the errors between the estimated poses and exact ground truth poses and low-level feature distances between similar poses.
- We show that our estimator successfully estimates poses of real fruit using more than two hundred test images,

which are collected with a stereo camera widely used in industrial applications.

The remainder of the paper is organized as follows. In Section II we provide an overview of related work. Our proposed approach for Deep CNN based pose estimation is introduced in Section III. In Section IV, we present experiments with our trained pose estimator with test images containing real bananas. We conclude the paper with final remarks and plans for further work in Section V.

II. RELATED WORKS

Object detection and its pose estimation is an essential task for robots and industrial applications, especially for picking and placing tasks. The exact 6D pose information of an object is required to decide about grasp points for picking and to define proper locations for placing. Therefore, pose estimation in 3D space has received a lot of attention with various approaches which dominantly include feature matching based methods and recently convolutional neural network based methods. State of the art methods are able to perform classification of objects and pose estimation at the same time [1], [18]. In the brief review below we focus on feature based approaches with a local or global descriptor and CNN based approaches.

A. Feature based approaches

Extracting features from training and test data, matching correspondence and calculating single transformation from a trained model to target scenes are typical processes of feature based approaches. Features for the 3D domain are designed to provide a generalized representation of the object shape using local attributes. One popular example is SHOT developed by Tombari et al. [17]. In [1] Aldoma et al. developed an approach which uses various features to generate possible hypotheses and select hypotheses which minimizes a cost function in order to remove false-positives. These feature based pose estimation approaches generally compute rigid transformations, which implicitly assumes that training models and target objects have the same shape. Wohlkinger et al. [19] uses CAD model to train global features to recognize real objects. This method shows robustness to shape variations, but it needs a large number of template images.

B. CNN based approaches

To employ recent convolutional neural networks, successfully used in the 2D image domain, to the 3D domain, which does not have enough training data, researchers tried to use pre-trained CNNs as a feature descriptor and trained additional classifiers for recognition and linear regression for pose estimation [16]. But [16] constrains object poses to in-plane rotation on the table, with one single degree of freedom. Generation of synthetic data is an option for training a CNN with depth images as input. [3] uses a 3D CAD models in order to train the typical CNN structure and finally gains a descriptor for a single channel depth images. This model was used for object classification tasks. Also, [3] considers object classification tasks, but this approach generates depth images from CAD models containing both, varied view points and randomly morphed shapes. CNN based 6D pose estimation is also described in [18], [5]. Both use pair-wise training to guide intermediate features to have larger distances for larger pose deviations. They design a small CNN network, which has only two convolution layers in order to train the CNN using a small number of training examples. In contrast to these approaches, we use a deep CNN which has five convolutional layers and pre-trained weights computed by a large number of 2D images. However, we refer to their pairwise training approaches to get a robust pose estimation performance.

III. METHOD

In the following paragraph we provide a detailed description of the proposed pose estimation approach, which consists of a deep CNN, generation of synthetic images and a pose refinement step for the final result, shown in Fig.1.

To be able to exploit the structure and pre-trained weights of well-established and tested CNNs taking three channels of a 2D color image as input, we transform single-channel depth images to three-channel color images. Finally, the pose estimation procedure at test time is described, including the refinement step to minimize the translational error.

A. Deep CNN for pose estimation with depth images

We employ Alexnet, which has proven results for 2D image classification tasks. The only different part is the last fully connected layer, which in our case has only four output channels for estimating the rotational transformation in quaternions, instead of a thousand channels for classification. Also, the final output is filtered by *tanh* function to provide normalized results between -1 and 1. The reason why we use a quaternion representation instead of Euler angles with three parameter is, the non-linearity and periodicity of Euler angles. For example, the numerical difference between 0 and 359 degrees is large, although the difference of the angles is small. However, the quaternion representation allows to calculate the pose difference as distance of each component of the quaternion values [12]. Most of the state of the art CNN models including Alexnet uses a 2D color image as input. State of the art for CNNs applied to depth images is to convert the depth image in the one channel to a color coded image in the three channels [6]. Among the possible color coding methods, directly matching each axis component of a surface normal to separate image channels has shown a superior performance [6]. Optionally, we use the depth value to scale the values of each pixel as described in (1) and (2).

$$I_D = 1.0 - \frac{P_z - min_z + \delta}{max_z - min_z + 2\delta} \tag{1}$$

$$P_{data} = I_D[N_x\, N_y\, N_z] \tag{2}$$

where P_{data} describes a single data point represented in the three channels. I_D is the scaled depth value and the remaining three values N_x, N_y and N_z are the individual axis of the

surface normal. The depth value I_D is normalized using the maximum and the minimum value of the point cloud with a margin δ to avoid zero values. Furthermore, the normalized depth value is subtracted from one to get higher values for closer points. Finally, every point is projected to a pixel in the 2D image.

B. Generation of synthetic training data

To train a CNN a large number of training examples is required, which cover each possible viewpoint of the object. We developed a fully autonomous data generation framework, which is able to cover all possible poses and shape variations. A 3D CAD model, e.g. from a public web resource or a reconstructed 3D scanned model, can be used as a reference model for this framework. The first step is to convert the CAD model to a point cloud format and to transform the reference coordinate system to the centroid of the model. After that, rotations for each axis are defined with 5 degree increments, which results in about 373K possible poses. In addition to the pose transformation, the shape transformation, i.e., scaling and shear is also defined for each pose. Scale and shear factors for each axis is randomly selected between a specified range in order to cover possible variations of the object. The reference model is transformed with the defined transformation matrix. Then it is placed to a location with a proper distance – usually found in the pose estimation scenario – to the camera. Self-occluded points are removed using a standard ray tracing of a camera view. Additionally, a randomly placed 2D rectangle is used to remove small parts of the object, in order to simulate partial occlusions and segmentation errors. Finally, the remaining points are used to render a depth image and non-object points or background points are filled with mean values (e.g. $P_{data} = [0.5\ 0.5\ 0.5]$ in case the normalized values are within $[0..1]$). The finally generated image is stored including the pose transformation using quaternions, i.e. in the same format the deep CNN provides.

C. Pairwise training for robust pose estimation

As proposed in [18], [5] our network is trained with input pairs to minimize feature distances of similar poses and maximize feature distances of different poses. The pose difference of a training pair is defined as the Euclidean distance between each quaternion component. Hence, a pair of training examples with a pose distance less than ρ_s is regarded as as positive pair and if the distance is larger than ρ_d it is regarded as a negative example (cf. 3).

$$\omega = \begin{cases} 1, & \text{if } ||q_{anchor} - q_{pair}||_2 < \rho_s\,, \\ 0, & \text{if } ||q_{anchor} - q_{pair}||_2 > \rho_d\,. \end{cases} \tag{3}$$

ω is given to the loss function to determine whether the current pair of images is positive or not, as described in (6). q_{anchor}, q_{pair} denote four-dimensional vectors of each pose transformation serialized from quaternion representation.

The whole input batch for each iteration is filled with positive and negative pairs. As described in Fig. 2, a data

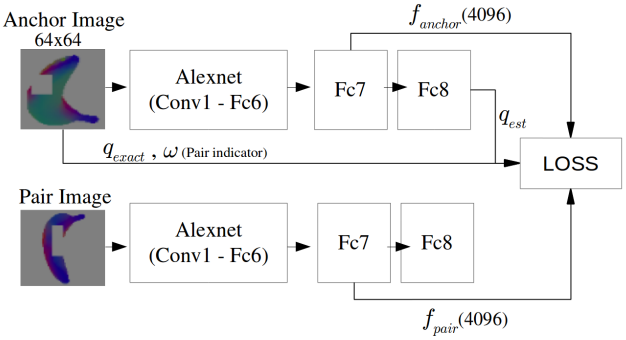


Fig. 2: Streamlines for pairwise training using shared weights for the CNNs. Output from both streamlines, i.e. the 7th layers and the last layers are used to compute the loss for the annotated training pairs.

pair is fed into the CNNs with the same weights and computed separately. To calculate the loss in each iteration, we use the output of the seventh fully connected layer with 4096 dimensions and the last fully connected layer with 4 dimensions, which is furthermore used to predict the rotation information in quaternion.

The loss function L for training can be separated into two part as described in (4).

$$L = l_r + l_f \tag{4}$$

For N batch images per each iteration, l_r represents a regression error between the annotated pose and the estimated pose which is defined as Euclidean distance (cf. 5), while l_f of 6 represents contrastive loss to guide features to have a smaller distance for similar poses and a larger distance for different poses.

$$l_r = \frac{1}{2N}\sum_{n=1}^{N} ||q_{est} - q_{exact}||_2^2 \tag{5}$$

$$l_f = \frac{1}{2N}\sum_{n=1}^{N} (\omega)d^2 + 2(1-\omega)max(1-d,0)^2 \tag{6}$$

$d = ||f_{achor} - f_{pair}||_2$ denotes the Euclidean distance between features computed from the seventh fully connected layer. ω, the parameter to classify training pairs as positive or negative examples, with similar or different poses is set in the data generation process. This contrastive loss has generally been used to train Siamese networks, which compare pairs of images [8]. In each iteration weights of the CNNs are updated to minimize the loss function using a stochastic gradient descent (SGD) solver. For this l_r is used to update all weights of the CNN, while l_f effects all weights except those of the last fully connected layer.

D. Estimation procedure

In contrast to the training, for pose estimation only a single stream line with one deep CNN is used. The last fully connected layer directly predicts the pose represented in quaternion. Given a depth image or a point cloud we classify

TABLE I: Pose estimation results with the proposed CNN

	Proposed CNN with ICP	Proposed CNN without ICP	ICP from Random Pose
Precision	**0.956**	0.822	0.265
Time (ms)	140±32	129±32	155±33

segmented objects. For the sake of simplicity in this paper we use a simple dominant plane segmentation and a nearest neighbour clustering of 3D points. The pre-processing to provide the input to the CNN is identical as for training (cf. III-B). The trained CNN directly estimates the rotation for the input segment. The corresponding tentative translation is computed from the centroid of the reference model and the segmented point cloud. Finally, a pose refinement step is performed. Basically, the translational error is dominantly caused by the difference between centroids of the reference model and the test image. This is because the centroid of the reference model is derived by the whole object shape, while the test image lack of occluded parts. To minimize this error, self-occluded parts of the reference model are removed after initial alignment, and the centroid of the reference model is recalculated. As a final step, we apply an iterative closest point (ICP) algorithm.

IV. EXPERIMENTS

We perform experiments to prove our concept with real bananas. An artificial 3D CAD model of a banana is selected and converted into a point cloud, further used to generate training images and store the ground truth pose. Scaling and shear transformations are randomly varied from 0.8 to 1.2 for each of three directions of views generated every 5 degree along each axis. The margin δ to calculate the depth to color conversion is set to 0.5. The CNN is implemented with the Caffe framework [11]. We set the initial weights using the pre-trained network, trained with Imagenet data [4]. To decide about positive and negative examples for pairs training examples, we set the threshold $\rho_s = 0.2$ for positive and to $\rho_d = 1.0$ for negative examples. Positive and negative pairs are randomly selected during the first epoch of cycles. The set of pairs is then fixed for further iterations to reduce training time. Every input image is re-sized to 64x64 pixel, while keeping the ratio between heights and widths of the rendered view. Test images are captured with an Ensenso N35, an industrial stereo sensor that provides only depth information with a resolution of 640x512. We assume robust segmentation results for the test scenes. Therefore, we placed the bananas on the table with enough distance to each other, in order to robustly extract segments, after detecting the dominant plane. We prepare five test scenes consisting of multiple bananas and approximately 278 scenes containing single banana per image using four different bananas. Estimated poses are evaluated manually. The criterion for the evaluation is based on the graspablity of the detected object, i.e. if the estimated pose is accurate enough to successfully grasp the object it is counted as positive. All experiments are performed with an Intel i7-6700K and a NVIDIA GTX1080 train the CNN.

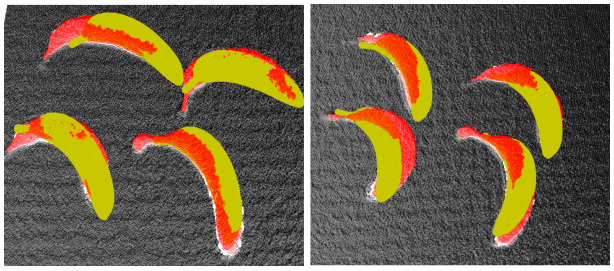
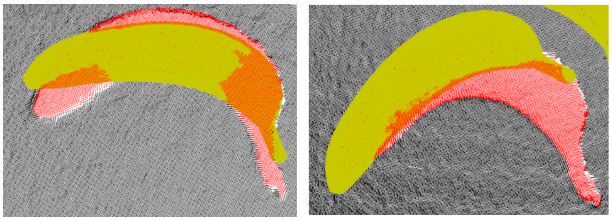


Fig. 3: Visualization of the estimated poses of multiple bananas. Red: real bananas in the test scene, yellow: estimation results



(a) Initial alignment before ICP (b) final alignment after 10 iteration

Fig. 4: Example of a bad alignment after ICP. This example is converged to match with an edge part of the banana

A. Results for bananas

Fig. 3 briefly shows the results for the test scenes containing multiple bananas. As shown in Table 1, the overall accuracy after pose refinement is about 95.6% and the computational time for each segment is about 0.14 second for each object, which is highly acceptable for robot grasping tasks.

B. Side effect of refinement steps using ICP

ICP generally improves the results. However, it sometimes causes worse alignment as shown in Fig. 4. This is because of the shape difference between the reference model and target scenes. The general ICP, which we use assumes a rigid transformation between the reference model and target model. Hence, depending on the inlier threshold ICP converges to partially fit to the scene, while the remaining point cloud does not contribute.

V. CONCLUSIONS

In this paper, we proved the concept of estimating poses of objects with a high shape variance using a deep CNN estimator. Furthermore, the proposed framework is able to use any kind of artificial or real scanned 3D model in order to generate enough data for training the deep CNN. This on going research will further be improved with the following ideas:

- The general rigid transformation ICP is not enough to refine the pose because the shape difference between the reference model and the individual objects. We refer to

non-rigid ICP [2] as an option to further improve the pose estimation.

- The preparation of an extensive annotated dataset will lead to an objective evaluation of our approach with various parameters and settings and a comparison to state of the art methods.
- Here, we assumed a correct segmentation result. In future we need to investigate optimal segmentation methods for real world experiments.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement No. 600623, STRANDS and industrial funding from OMRON Corporation in Japan

REFERENCES

[1] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypothesis verification framework for 3d object recognition in clutter," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1383–1396, 2016.

[2] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid icp algorithms for surface registration," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[3] F. M. Carlucci, P. Russo, and B. Caputo, "A deep representation for depth images from synthetic data," *arXiv preprint arXiv:1609.09713*, 2016.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[5] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim, "Siamese regression networks with efficient mid-level feature extraction for 3d object pose estimation," *arXiv preprint arXiv:1607.02257*, 2016.

[6] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 681–687.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[8] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 1735–1742.

[9] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[10] T. Hodan, P. Haluza, S. Obdrzalek, J. Matas, M. Lourakis, and X. Zabulis, "T-less: An rgb-d dataset for 6d pose estimation of texture-less objects," *arXiv preprint arXiv:1701.05498*, 2017.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[12] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[14] E. Muñoz, Y. Konishi, V. Murino, and A. Del Bue, "Fast 6d pose estimation for texture-less objects from a single rgb image," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5623–5630.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[16] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1329–1335.

[17] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision*. Springer, 2010, pp. 356–369.

[18] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3d pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3109–3118.

[19] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *2011 IEEE International Conference on Robotics and Biomimetics*, Dec 2011, pp. 2987–2992.

Confusing Similarity between Visual Trademarks A Dataset Based on USTTAB Examinations*

Lukas Knoch[1] and Mathias Lux[2]

***Abstract*— Trademarks are an important visual clue for customers to identify brands, products and companies, and can influence the buying decision significantly. One major problem with visual trademarks is, that newly registered trademarks are required by law not to be visually similar to existing ones. Therefore, automatic detection of visually similar trademarks is an important use case for content based image retrieval. Confusing similarity between trademarks is defined by law, and numerous cases of the *United States Trademark Trial and Appeal Board* (USTTAB) handling trademark similarity are available. In this paper we present a novel and freely available data set for evaluation of trademark similarity algorithms based on real life data, ie. all registered trademarks in the USA as well as USTTAB decisions and expert opinions. The data set should serve as a basis for further investigations, ie. extension of the data set by crowd sourcing and consideration of the intuitive concept of visually confusing similarity.**

I. Introduction

Visual trademarks, or *logos*, often influence our buying decisions and are therefore valuable goods for the companies owning the visual trademark. A common and well known example is the Apple company logo (compare Figure I) present on iPhones, iPads and Apple computers. Apple Computers invests time and money to find out if other companies worldwide use similar logos on similar products. The same approach is also taken by many companies who define themselves through their brands, like Nike, Adidas, or Red Bull.

Fig. 1. Examples of well known logos and protected trademarks in many countries including the Apple logo, the Github logo and the Nike swoosh.

To avoid confusion between different trademarks, they must be *dissimilar enough* to each other. Some companies even try to trick customers by deliberately using trademarks that are similar to well known signs. To avoid fraud, trademarks can be protected by law. There are several offices in charge of managing trademark registrations for different regions including the *European Union Intellectual Property Office* (formerly Office for Harmonization in the Internal Market, short OHIM) or the *United States Patent and Trademark Office* (short USPTO). If a new trademark is registered, it has to be ensured that there is no confusing similarity to any other previously registered marks. This difficult job is executed by professional *trademark examiners* who compare the different trademarks to each other and decide about the similarity. While there are systems in place like the textual *Vienna Classification* [21], taxonomies which are intended to help the examiner, these systems are tedious and error prone as they rely on manual annotation.

Another way of assisting the examiners are visual trademark retrieval systems. These systems can take a specific trademark as an input and deliver a set of trademarks ranked by similarity to the query image, which is commonly referred to as query by example in content based image retrieval. While several systems have been proposed [28], [9], [15], their retrieval performance leaves a lot of room for improvement [25]. There are several papers suggesting new algorithms for visual trademark retrieval, but their evaluations are based on trademark datasets downloaded from the internet [22], [23], pure shape datasets like MPEG-7 [13], [1] or hand picked ground truth [27], [20], [5], [26]. Unfortunately, objective evaluation of these systems is currently hardly possible as there are no datasets available that (i) represent real world data, ie. the actual visual trademarks registered at the trademark offices, and (ii) that are based on expert opinions and court decisions.

To aid with the development of content based visual trademark retrieval systems, this paper introduces a realistic novel dataset based on real world trademark trials. Our dataset can provide the base for research on content based visual information retrieval systems. The dataset contains $1{,}859{,}218$ visual trademarks registered at the United States Patent Office (USPTO) as well as three different sets of ground truths based on trials at the United States Trademark Trial and Appeal Board (USTTAB). The raw visual trademarks and trial data is provided by Google[1][2], the extracted meta data is available at a public website[3].

* This article is based on the master's thesis of Lukas Knoch and has been done with the support of the World Intellectual Property Organization, WIPO, partially as the result of an internship at the UN Headquarters in Geneva, CH

[1] Lukas Knoch was student at Alpen-Adria Universität Klagenfurt, Austria lukas.knoch@aau.at

[2] Mathias Lux is Associate Professor at the Institute for Information Technology at Alpen-Adria Universität Klagenfurt mathias.lux@aau.at

[1] https://www.google.com/googlebooks/uspto-trademarks-usamark.html, last visited 2016-01-19

[2] https://www.google.com/googlebooks/uspto-trademarks-ttab.html, last visited 2016-01-19

[3] www.rumpelcoders.at/usttabdataset

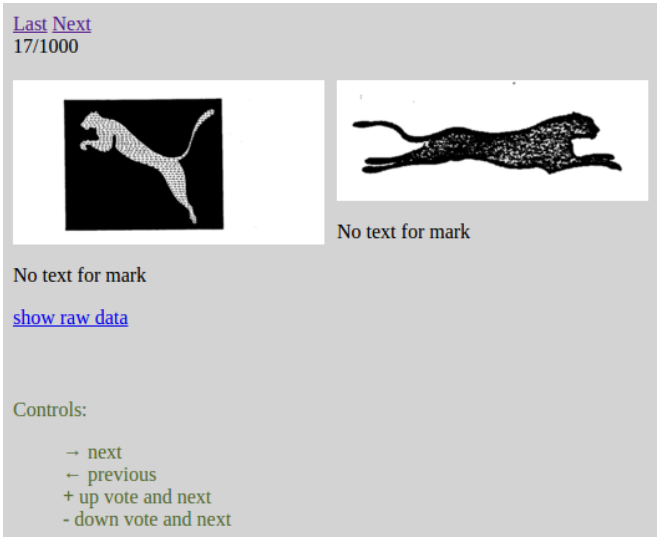


Fig. 2. The view on a pair of logos in the visual similarity evaluation application.

II. Dataset

As already mentioned Google offers several trademark collections as free download[4] in cooperation with the USPTO. Note at this point that these downloads offer the actual USPTO data, ie. the actual image files filed for registration as well as the resulting metadata. On the Google site, daily trademark applications, images and the USTTAB trials data from 1955 until today are available. All of these can be downloaded in chronologically ordered ZIP-archives containing an XML file with describing all trials in the specific period of time.

A. Selection Criteria

For the creation of our new ground truth, the trials from 1955 until end of August 2015 were chosen, being all trials available at the time of extraction. Each trial entry in the retrieved data contains the *party-information*, a section that includes information about all parties involved in the trial. Each party has zero or more properties, which correspond to the trademarks associated with it. The properties are identified by a unique identification and a serial number.

A first filtering step was taken by selecting only those trials that do regard an opposition. For the dataset only oppositions are interesting, as those contain cases of confusing similarities, in contrast to obvious ones. In the next step, all entries with exactly two parties and exactly one associated property per party were selected. In all other cases it is not possible to distinguish the trademarks relevant for this claim. By joining this data with all available US trademark images, trials regarding non-visual trademarks could be removed.

As the presence of trademark images does not guarantee that the trial was filed because of visual similarity, the next step was to detect the *type of similarity*. Unfortunately, there is no formal classification contained in the data. To overcome this problem, a web-based application was developed, which allows experts to decide whether the trial was based on visual similarity or not. The experts were chosen from three different areas of expertise: One from the field of visual information retrieval at the University of Klagenfurt, one from the field of trademark retrieval at the World Intellectual Property Organisation and one with appropriate knowledge in both fields. To be able to create a sufficiently big ground truth in reasonable time, 1000 trials were randomly chosen from the previously selected. The application showed two trademark images next to each other and asked the expert to decide whether the claim was due to visual similarity or not. To assist the experts in their decision, the trademark name was presented beyond the image if one was present (compare Fig. I).

0 1

Fig. 3. One of the logo pairs in the USTTAB strict ground truth

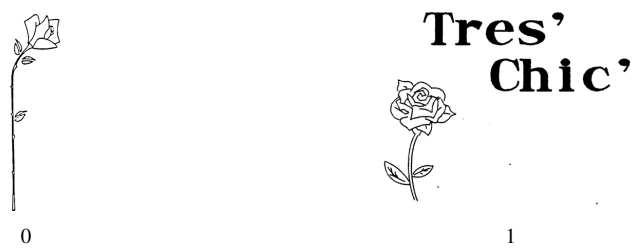
For the 1000 logo pairs, all experts agreed on visual similarity in 160 cases. At least two of the three experts agreed on visual similarity in 384 cases while there are 451 trials in which only one expert judged that the trial was due to visual similarity. The 1000 pairs included nine control pairs of obvious visual similarity, which were correctly answered by all experts.

B. Properties

The resulting dataset consists of 1.8 million visual trademarks. Those trademarks are either registered, pending or canceled in the USPTO registration data base. The set is composed by 1,587,248 verbal signs, 533,910 non-verbal signs and 4,867,626 combined trademarks. The signs are of varying image quality with different resolution, in color, gray scale or binary black & white format. As this data is directly form the USPTO's registration data base, its composition is realistic and, therefore, well suited for objective evaluations.

From the USTTAB trials and the expert's decisions, three blends of the data set were created. The first blend includes only logos on which all experts agreed. It is therefore referred to as *strict ground truth*. An example for this set can be seen in Fig. 3. The second blend consists of the logo pairs a majority of experts agreed on, the *majority ground truth* (cp. Fig. 4). Finally, the *minority ground truth* consists of all pairs with at least one expert voting for visual similarity (cp. Fig. 5).

[4] https://www.google.com/googlebooks/uspto-trademarks.html, last visited 2016-01-19



0 1

Fig. 4. One of the logo pairs in the USTTAB majority ground truth not being part of the strict set.

0 1

Fig. 5. One of the logo pairs in the USTTAB minority ground truth not being part of the strict set or the majority set.

C. Data Format

The dataset is defined in multiple text files. The first file, *data_full.txt*, contains the registration number of all trademarks used as diversifiers as well as all trademarks from the ground truth. Each line contains one number. The files *data_10.txt* and *data_1.txt* contains a 10% and the 1% random sample in the same format for test on smaller data sets, while still providing comparability. The ground truth is available in the folder *groundtruth*. This folder contains the files *gt_strict.txt*, *gt_majority.txt* and *gt_minority.txt*, which hold a comma separated list of trademark registration numbers identifying the visually similar logo pairs.

III. Retrieval Baseline

To provide a baseline for comparison, several state of the art algorithms were tested on the new dataset. The tests were executed with a benchmark software based on LIRE [19], which was presented in [16]. Note at this point that all descriptors used in the test as well as the benchmarking suite have been contributed to the LIRE open source project[5].

A. Tested Features

The following features were chosen to be tested on the new dataset because they not only cover a wide diversity of features types like color, shape, texture and combinations of them, but also because some of them were proposed as well suited in the trademark retrieval domain [2]. *Local Binary Patterns* (LBP) [11] represent the local texture of an image by encoding the threshold of each pixel's neighborhood in a binary number. A rotation invariant version can be achieved by restricting the observed patterns the so-called *uniform patterns*. For *Binary Patterns Pyramid (BPP)* a spatial pyramid was applied on the LBP. The *Shapeme Histogram Descriptor* (Shapeme) captures the global shape of an image by extracting the shape context and clustering with K-nearest neighbors. In this experiment, the shape contexts were calculated for 256 points chosen by Jitandta's algorithm with three time oversampling and 512 bins for the descriptor [3]. *Centrist* is a feature similar to LBP and also captures local texture. *Joint Composite Descriptor* (JCD) [29] combines the two fuzzy histogram features Color and Edge Directivity Descriptor [8] and Fuzzy Color and Texture Histogram [6]. *Adaptive Contours and Color Integration Descriptor (ACCID)* [12] captures visually salient shapes and combines them with a fuzzy color histogram. *Pyramid Histogram of Oriented Gradients* (PHOG) [4] extracts information about the local shape and the layout of the shape with a with a Spatial Pyramid Kernel. In this experiment, 15 orientation bins were used as that has been found effective in the context of trademark retrieval ($PHOG_{15}$, cp. [16]).

For the evaluation, the logos were resized to a maximum width and height of 512 pixel retaining aspect ratio. In an additional preprocessing step, a despeckle filter was applied and the white pixels were trimmed. Table III-A shows the result of the outlined features on the full USTTAB dataset utilizing the strict ground truth. As can be seen easily from Table III-A, $PHOG_{15}$ outperforms the other descriptors regarding recall and mean average precision. In terms of average and normalized rank, the Shapeme feature performs better than PHOG.

Fig. 6 shows the comparison of the mean average precision (MAP) for $PHOG_{15}$, Shapeme, ACCID, JCD, BBP, and Centrist on the three different ground truths. For Shapeme and $PHOG_{15}$, the MAP correlates to the agreement of the experts. The less agreement in the ground truth, the lower the MAP.

IV. Conclusion and Challenges

The data set as presented provides a hard challenge to researchers in visual information retrieval. While the data from the USTAB trials provides pairs of trademarks with confusing similarity, for both of the pairs it is very likely to find numerous visually similar other logos, which were not part of a trial. Moreover, companies often file trademarks in different version, re-register them or have multiple data records in the USTAB registration data base. Fig. 7 shows an example result list from searching for a visual trademark from the ground truth. At position 0 the query is shown and only on position 49 of the list the offending trademark is found. However, it can be easily seen that the logos in between are visually similar to the trial's logo pair.

While this is definitely a problem for a common use case like digital photo retrieval, in the visual trademark domain the experts doing inquiries certainly go beyond the first few results and finding the offending logo in the first 100 or even 500 results helps them with their work. Note also at this point that the data set is especially about confusing similarity, not near duplicate search, as the latter one has been subject to a lot of research already. Therefore, for future work we

[5] `http://www.lire-project.net/`, last visited 2016-01-19

Feature	Rank	$\widetilde{Rank}$	Recall@100	Recall@500	MAP
LBP	230,784.8	0.124	0.267	0.323	0.178
LBP (RotInv)	250,123.1	0.135	0.305	0.389	0.164
Shapeme	201,853.2	**0.10856828071845802**	0.488	0.513	0.378
Centrist	307,558.3	0.165	0.500	0.502	0.496
BPP	327,727.0	0.176	0.500	0.503	0.496
JCD	267,515.1	0.144	0.503	0.512	0.492
ACCID	227,305.3	0.122	0.505	0.510	0.499
$PHOG_{15}$	220,036.4	0.122	**0.5248344370860927**	**0.5364238410596026**	**0.5157031013278772**

TABLE I

THE RESULTS OF THE STRICT GROUND TRUTH (302 QUERIES) EVALUATED ON THE FULL USTTAB COLLECTION IN TERMS OF AVERAGE RANK, NORMALIZED RANK, RECALL AT 100, RECALL AT 500, AND MEAN AVERAGE PRECISION.

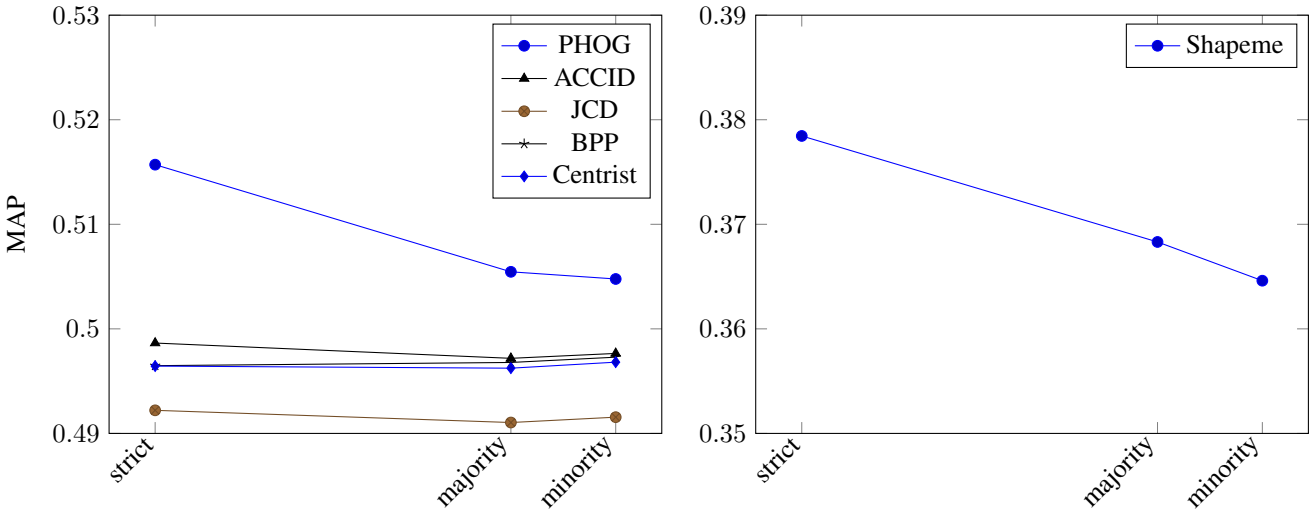


Fig. 6. Comparison of MAP results of different algorithms on the three USTTAB ground truths strict, majority and minority for the full collection. The x axis is scaled to represent the number of queries in each ground truth (302 for strict, 750 for majority and 882 for minority). While BPP, ACCID, JCD and CENTRIST hardly show any change in value, PHOG and Shapeme seem to mirror the human perception.

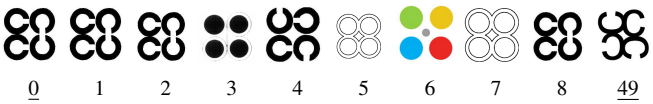


Fig. 7. Examples of retrieval results for a logo pair from the ground truth. At rank 0 the image shows the query, then the first eight results and only at rank 49 the logo from the corresponding USTAB trial.

aim to take a close look at the evaluation procedure, ie. by investigating the possibility of taking into account similar images that have not been in trials, as has been done for the pooling method in text information retrieval [18].

The data set has already been employed for testing different parameters of the PHOG and Shapeme features as well as extensive evaluations using other local and global features alike. The findings have already been integrated in the trademark search engine of the World Intellectual Property Organization (WIPO) of the United Nations[6].

However, there is a long way to go and there are several tasks, for which we propose crowd workers to be employed:

Identification of multiple instances. As noted before logos are submitted and re-submitted by the same company all around the world. These duplicate entries, which are often near duplicates in the visual domain, are visually similar, but should be considered separately. Crowd workers could identify and label the (near) duplicate entries.

Offending logos not investigated by the appeal board. As it is a lengthy and complicated process to file an appeal, there are a lot of visually confusing similarities that have no been investigated by the appeal board. In the current version of the data set these offending logos might show up as false positives in benchmarking. Crowd workers could label the offending logos to be treated separately.

Judging visually confusing similarity. While we had experts judge the offending logos upon visual vs. conceptual confusion, we think that the intuitive concept of visually confusing logos in the head of actual consumers is different to the concept adopted by legal experts. With the help of crowd workers we could paint a picture of how consumers see visual trademarks as well as the relevance and impact of offending logos and provide feedback to the legal experts.

[6] `http://www.wipo.int/branddb`, last visited 2016-08-30

ACKNOWLEDGEMENTS

We'd like to thank the Glenn MacStravic from the World Intellectual Property Organization for his ongoing support and critical reflection and discussion of our work.

REFERENCES

[1] S. Agarwal, N. Chaturvedi, and P. K. Johari, "Content based trademark retrieval by integrating shape with colour and texture information," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 4, pp. 295–302, 2014.

[2] S. Belongie and J. Malik, "Matching with shape contexts," in *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, 2000, pp. 20–26.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 509–522, Apr 2002.

[4] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, ser. CIVR '07. New York, NY, USA: ACM, 2007, pp. 401–408. [Online]. Available: http://doi.acm.org/10.1145/1282280.1282340

[5] A. Cerri, M. Ferri, and D. Giorgi, "A new framework for trademark retrieval based on size functions," *Vision, Video, and Graphics*, 2005.

[6] S. Chatzichristofis and Y. Boutalis, "Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval," in *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS '08. Ninth International Workshop on*, May 2008, pp. 191–196.

[7] B. D. Cullity, *Introduction to Magnetic Materials*. Reading, MA: Addison-Wesley, 1972.

[8] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Comput. Surv.*, vol. 40, no. 2, pp. 5:1–5:60, May 2008. [Online]. Available: http://doi.acm.org/10.1145/1348246.1348248

[9] J. Eakins, J. Boardman, and K. Shields, "Retrieval of trade mark images by shape feature-the artisan project," in *Intelligent Image Databases, IEE Colloquium on*, May 1996, pp. 9/1–9/6.

[10] R. K. Gupta and S. D. Senturia, "Pull-in time dynamics as a measure of absolute pressure," in *Proc. IEEE International Workshop on Microelectromechanical Systems (MEMS'97)*, Nagoya, Japan, Jan. 1997, pp. 290–294.

[11] D.-C. He and L. Wang, "Texture unit, texture spectrum, and texture analysis," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 28, no. 4, pp. 509–512, Jul 1990.

[12] C. Iakovidou, "Development, implementation and evaluation of methods for the description and the retrieval of multimedia visual content using intelligent techniques," pp. 84–100.

[13] T. Iwanaga, H. Hama, T. Toriu, and T. T. Zin, "A modified histogram approach to trademark image retrieval," *International Journal of Computer Science and Network Security*, vol. 11, no. 4, April 2011.

[14] R. Jain, K. K. Ramakrishnan, and D. M. Chiu, "Congestion avoidance in computer networks with a connectionless network layer," Digital Equipment Corporation, MA, Tech. Rep. DEC-TR-506, Aug. 1987.

[15] T. Kato, "Database architecture for content-based image retrieval," pp. 112–123, 1992. [Online]. Available: http://dx.doi.org/10.1117/12.58497

[16] L. Knoch, "Content based search and retrieval in visual trademarks and logos," 2016.

[17] Q. Li, "Delay characterization and performance control of wide-area networks," Ph.D. dissertation, Univ. of Delaware, Newark, May 2000. [Online]. Available: http://www.ece.udel.edu/~qli

[18] A. Lipani, M. Lupu, and A. Hanbury, "Splitting water: Precision and anti-precision to reduce pool bias," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 103–112.

[19] M. Lux and S. A. Chatzichristofis, "Lire: Lucene image retrieval: An extensible java cbir library," in *Proceedings of the 16th ACM International Conference on Multimedia*, ser. MM '08. New York, NY, USA: ACM, 2008, pp. 1085–1088. [Online]. Available: http://doi.acm.org/10.1145/1459359.1459577

[20] A. Nigam, A. K. Garg, and R. Tripathi, "Content based trademark retrieval by integrating shape with colour and texture information," *International Journal of Computer Applications*, vol. 22, no. 7, May 2011.

[21] W. I. P. Organization, *International Classification of the Figurative Elements of Marks: Vienna Classification*, ser. WIPO publication. World Intellectual Property Organization, 1997. [Online]. Available: https://books.google.at/books?id=nw3QAAAACAAJ

[22] M. Rusiñol, D. Aldavert, D. Karatzas, R. Toledo, and J. Lladós, "Interactive trademark image retrieval by fusing semantic and visual content," in *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings*, 2011, pp. 314–325. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20161-5_32

[23] Z. Shaaban, "Trademark image retrieval system using neural networks," *International Journal of Computer Science and Network*, vol. 3, no. 1, February 2014.

[24] W. V. Sorin, "Optical reflectometry for component characterization," in *Fiber Optic Test and Measurement*, D. Derickson, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[25] TM5, "Report on the TM5 image search project," 2015, Project Report.

[26] R. H. van Leuken, M. F. Demirci, V. J. Hodge, J. Austin, and R. C. Veltkamp, "Layout indexing of trademark images," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, ser. CIVR '07. New York, NY, USA: ACM, 2007, pp. 525–532. [Online]. Available: http://doi.acm.org/10.1145/1282280.1282356

[27] C.-H. Wei, Y. Li, W. Y. Chau, and C.-T. Li, "Trademark image retrieval using synthetic features for describing global shape and interior structure," *Pattern Recognition*, vol. 42, no. 3, pp. 386–394, 2009.

[28] J. Wu, C. Lam, B. Mehtre, Y. Gao, and A. Narasimhalu, "Content-based retrieval for trademark registration," *Multimedia Tools and Applications*, vol. 3, no. 3, pp. 245–267, 1996. [Online]. Available: http://dx.doi.org/10.1007/BF00393940

[29] K. Zagoris, S. Chatzichristofis, N. Papamarkos, and Y. Boutalis, "Automatic image annotation and retrieval using the joint composite descriptor," in *Informatics (PCI), 2010 14th Panhellenic Conference on*, Sept 2010, pp. 143–147.

[30] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.

Feedback Loop and Accurate Training Data for 3D Hand Pose Estimation[†]

Markus Oberweger[1], Gernot Riegler[1], Paul Wohlhart[1] and Vincent Lepetit[1,2]

***Abstract*— In this work, we present an entirely data-driven approach to estimating the 3D pose of a hand given a depth image. We show that we can correct the mistakes made by a Convolutional Neural Network (CNN) trained to predict an estimate of the 3D pose by using a feedback loop of Deep Networks, also utilizing a CNN architecture.**

Since this approach critically relies on a training set of labeled frames, we further present a method for creating the required training data. We propose a semi-automated method for efficiently and accurately labeling each frame of a depth video of a hand with the 3D locations of the joints.

I. INTRODUCTION

Accurate hand pose estimation is an important requirement for many Human Computer Interaction or Augmented Reality tasks. Due to the emergence of 3D sensors, there has been an increased research interest in hand pose estimation in the past few years [3], [6], [7]. Despite the additionally available information from 3D sensors, it is still a very challenging problem, because of the large number of degrees of freedom, and because images of hands exhibit self-similarity and self-occlusions.

A popular approach to predict the position of the joints is to use a discriminative method [3], [7], which are now robust and fast. To further refine the pose, such methods are often used to initialize a complex optimization where a 3D model of the hand is fit to the input depth data [5].

In this paper, we build upon recent work that learns to generate images from training data [1] in order to remove the requirement of a 3D hand model. We introduce a method that learns to provide updates for improving the current estimate of the pose, given the input image and the image generated for this pose estimate. Running these steps iteratively, we can correct the mistakes of an initial estimate provided by a simple discriminative method.

However, this approach, amongst other recent work (*e.g.* [6], [7]), has shown that a large amount of accurate training data is required for reliable and precise pose estimation. Although having accurate training data is very important, there was only limited scientific interest in the creation of such, and authors have had to rely on *ad hoc* ways that are prone to errors [6]. These errors result in noisy training and test data, and make training and evaluating uncertain. Therefore, we developed a semi-automated approach that makes it easy to annotate sequences of articulated poses in 3D from a single depth sensor only.

In the next two sections, we first describe our proposed feedback loop, and then we present our method for efficiently creating training data.

II. TRAINING A FEEDBACK LOOP

We aim at estimating the pose of a hand in the form of the 3D locations of its joints from a single depth image. We assume that a training set of depth images labeled with the corresponding 3D joint locations is available. An overview of our method is shown in Fig. 1.

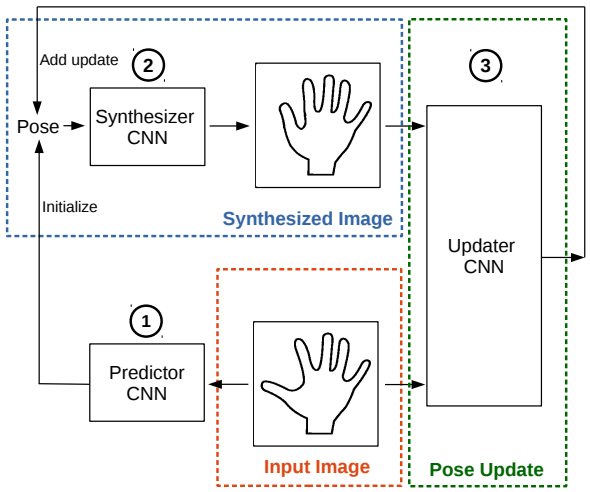


Fig. 1. Overview of our method: We use a CNN (1) to predict an initial estimate of the 3D pose given an input depth image of the hand. The pose is then used to synthesize an image (2), which is used together with the input depth image to derive a pose update (3). The update is applied to the pose and the process is iterated.

We first train a *predictor* to predict an initial pose estimate in a discriminative manner given an input depth image. We use a Convolutional Neural Network to implement this predictor with a very simple architecture [3].

In practice, the initial pose is never perfect, and following the motivation provided in the introduction, we introduce a hand model learned from the training data. This CNN-based model, referred to as *synthesizer*, can synthesize the depth image corresponding to a given pose. The network architecture is strongly inspired by [1]. It predicts an initial latent representation of feature maps, followed by subsequent unpooling and convolution layers to generate a depth image.

Further, we introduce a third function that we call the *updater*. It learns to predict updates to improve the pose estimate, given the input image and the image produced

[†]This work is based on published work in ICCV'15 [4] and CVPR'16 [2].

[1]The authors are with the Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria {oberweger,riegler,wohlhart,lepetit}@icg.tugraz.at

[2]The author is with the Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux, Bordeaux, France

by the synthesizer. We iterate this update several times to improve the initial pose estimate. Again, the updater function is implemented as a CNN. The architecture is inspired by a Siamese network with two identical paths. One path is fed with the observed depth image and the second path is fed with the image from the synthesizer.

Ideally, the output of the updater should bring the pose estimate to the correct pose in a single step, which is a very difficult problem in practice. However, our only requirement from the updater is to predict an update resulting in a pose closer to the ground truth. The introduction of the synthesizer allows us to virtually augment the training data and to add arbitrary poses to the set of poses, which the updater might perceive during testing and be asked to correct. We refer to our paper [4] for more details.

III. CREATING TRAINING DATA EFFICIENTLY

Since the presented hand pose estimation method critically relies on labeled training frames, we present a method for the creation of such frames. Given a sequence of depth maps capturing a hand in motion, we want to estimate the 3D joint locations for each depth map with minimal effort.

We start by automatically selecting a subset of depth frames, we will refer to as *reference frames*, for which a user is asked to provide annotations. Our method selects these reference frames based on the appearances of the frames over the whole sequence. For this, we train an autoencoder that learns an unsupervised representation that is sensitive to image nuances due to hand articulation. We use this representation to formalize the frame selection as a submodular optimization. A user is then asked to provide the 2D reprojections of the joints with visibility information in these reference frames, and whether these joints are closer or farther from the camera than the parent joint in the hand skeleton tree. We use this information to automatically recover the 3D locations of the joints by solving a least-squares problem. Next, we iteratively propagate these 3D locations from the reference frames to the remaining frames. We initialize the pose of the frame with the pose of the visually closest reference frame and optimize the local appearance together with spatial constraints. This gives us an initialization for the joint locations in all the frames. However, each frame is processed independently. We can improve the estimates further by introducing temporal constraints on the 3D locations and perform a global optimization, enforcing appearance, temporal, and spatial constraints over all 3D locations for all frames. If this inference fails for some frames, the user can still provide additional 2D reprojections; by running the global inference again, a single additional annotation typically fixes many frames. See our paper [2] for more details.

IV. EVALUATION

We evaluate our hand pose estimation method on the NYU Hand Pose Dataset [7], a challenging real-world benchmark for hand pose estimation. This dataset is publicly available, is backed up by a huge quantity of annotated samples, and also shows a high variability of poses.

We show the benefit of using our proposed feedback loop to increase the accuracy of the 3D joint localization in Fig. 2. While [7] and [3] have an average 3D joint error of 21 mm and 20 mm respectively, our proposed method reaches an error reduction to 16.5 mm. The initialization with the simple and efficient proposed predictor has an error of 27 mm. When we use a more complex initialization [3] with an error of 23 mm, we can decrease the average error to 16 mm.

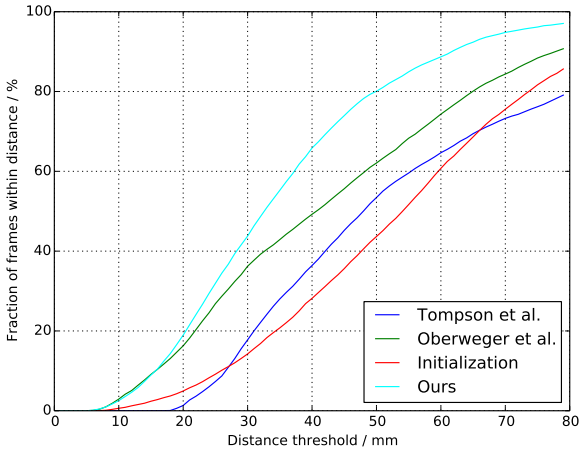


Fig. 2. Quantitative evaluation of hand pose estimation. The figure shows the fraction of frames where all joints are within a maximum distance. A higher area under the curve denotes better results. We compare our method to the baseline of Tompson *et al.* [7] and Oberweger *et al.* [3]. Although our initialization is worse than both baselines, we can boost the accuracy of the joint locations using our proposed feedback loop.

To demonstrate our training data creation approach, we evaluate it on a synthetic dataset, which is the only way to have depth maps with ground truth 3D locations of the joints. On this dataset we evaluate the accuracy of the automatically inferred 3D locations for the reference frames. We obtain an average 3D joint error of 3.6 mm only from 2D reprojections with visibility and depth order. Our method is also robust to annotation noise. We can propagate the 3D joint locations to the remaining frames, for which we achieve an average 3D joint error of 5.5 mm over the full sequence by only requiring manual 2D annotations for 10% of all frames.

REFERENCES

[1] A. Dosovitskiy, J. T. Springenberg, and T. Brox, "Learning to Generate Chairs with Convolutional Neural Networks," in *CVPR*, 2015.

[2] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently Creating 3D Training Data for Fine Hand Pose Estimation," in *CVPR*, 2016.

[3] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands Deep in Deep Learning for Hand Pose Estimation," in *Proc. of CVWW*, 2015.

[4] ——, "Training a Feedback Loop for Hand Pose Estimation," in *ICCV*, 2015.

[5] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and Robust Hand Tracking from Depth," in *CVPR*, 2014.

[6] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture," in *CVPR*, 2014.

[7] J. Tompson, M. Stein, Y. LeCun, and K. Perlin, "Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks," *ACM Transactions on Graphics*, vol. 33, no. 5, pp. 169–179, 2014.

Active contour models for individual keratin filament tracking

Dmytro Kotsur[1], Rudolf E. Leube[2], Reinhard Windoffer[2] and Julian Mattes[3]

***Abstract*— As a major component of the cytoskeleton, keratin filaments form a branched network, which plays a significant role in the mechanical response, motion and dynamics of the cell. They undergo a complex dynamic lifecycle, which we aim to investigate by tracking individual filaments. In this paper we introduce an active contour-based tracking algorithm to analyze the motion of individual keratin filaments in sequences of confocal images. The algorithm combines parametric active contours (snakes) with Lukas-Kanade's algorithm for optical flow calculation. We define an image preprocessing workflow to compute robustly the external energy of the snake and we impose an additional structural constraint for controlling the length of the contour.**

I. INTRODUCTION

The cytoskeleton plays a main role in cellular motility and dynamics, which in turn is of high relevance for vital and also for pathological processes, such as wound healing and tumor metastasis [5]. As a major component of the cytoskeleton, keratin filaments form a branched network and are essential for the mechanical response to external forces. Biophysical investigation and analysis of different types of keratin filaments requires their localization and the extraction of their motion in the time-sequences of consecutive confocal images. As it was shown previously [7], [3], [4], this problem can be successfully approached for separated individual actin filaments. However, applying this approach to tracking of keratin filaments within a branched network may lead to additional complications and errors, as for example, uncontrolled growth of the snake. In this paper we introduce a tracking algorithm based on stretching open active contours [3] to analyze the global motion features of individual keratin filaments within their network. We define an image preprocessing workflow to calculate robustly the "external energy" of the snake and impose an additional structural constraint for controlling the length of the contour.

II. TRACKING ALGORITHM

In this section, we first define our active contour model as a minimization problem. Then, we introduce an "external energy" based on the image and impose a contour length constraint to control snake growth. Finally, we combine all steps together and present an overall tracking procedure.

[1]Dmytro Kotsur is with Software Competence Center Hagenberg GmbH (SCCH), Austria, Dmytro.Kotsur@scch.at

[2]Rudolf E. Leube and Reinhard Windoffer are with MOCA, Institute of Molecular and Cellular Anatomy, RWTH Aachen University, Aachen, Germany, {rleube, rwindoffer}@ukaachen.de

[3]Julian Mattes is with MATTES Medical Imaging GmbH, Hagenberg, Austria, Julian.Mattes@mattesmedical.at

A. Parametric snakes: active contour models

We define a filament as a parametric curve $\mathbf{x}(s) = [x(s), y(s)], s \in [0,1]$. According to [2], the position of the filament within a frame in a time-sequence is obtained by minimizing the following so-called "energy" functional:

$$E = \int_0^1 \frac{1}{2}\left(\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2\right) + E_{ext}\left(\mathbf{x}(s)\right)ds \quad (1)$$

where α and β are parameters which control the stretching and bending resistance of the curve, correspondingly. This problem is solved by reducing (1) to a differential equation and applying an iterative scheme with an artificial time variable t:

$$\mathbf{x}_t(s,t) = \alpha\mathbf{x}_{ss}(s,t) + \beta\mathbf{x}_{ssss}(s,t) - \nabla E_{ext}(\mathbf{x}(s,t)) \quad (2)$$

The impact of the "external energy" E_{ext} or the gradient of "external energy" ∇E_{ext} is crucial in this problem, because the convergence of a snake considerably depends on this term.

B. External energy and structural constraints

In Xu et al. [6] the gradient of the "external energy" ∇E_{ext} is replaced by the vector field $\mathbf{v}(x,y) = [u(x,y), v(x,y)]$, which minimizes the functional:

$$\mathscr{E} = \int\int \mu\left(u_x^2 + u_y^2 + v_x^2 + v_y^2\right) + |\nabla f|^2|\mathbf{v} - \nabla f|^2 dxdy \quad (3)$$

where $f(x,y)$ is the intensity of the pixel at the position (x,y), $|\bullet|$ is the Euclidean norm and μ is the regularization (smoothness) parameter. The vector field $\mathbf{v}(x,y)$ is called gradient vector flow (GVF). In this case the evolution of the snake on a single frame is defined as follows:

$$\mathbf{x}_t(s,t) = \alpha\mathbf{x}_{ss}(s,t) + \beta\mathbf{x}_{ssss}(s,t) - \mathbf{v}(\mathbf{x}(s,t)) \quad (4)$$

It is shown in [6] that GVF has a larger capture range, compared to the vector field given by ∇E_{ext} defined in [2]. It also improves the snake convergence in case of high concavities. However, the intensity variation along a filament may be high, which leads to additional errors during snake convergence. Therefore, we preprocess images applying the following pipeline of filters: Gaussian smoothing; Hessian ridge enhancement; gamma contrast correction.

The drawback of the snake algorithm itself as defined in [2] is that the open-ended contour (Fig. 1C) tends to shrink over time (Fig. 1D). To overcome this, we use a stretching term for open ends as defined in [7]. However, it may lead to overgrowth of the contour (Fig. 1E). We it this by processing endpoints separately. We define an additional distance-based "energy" potential for the branching and end points of the

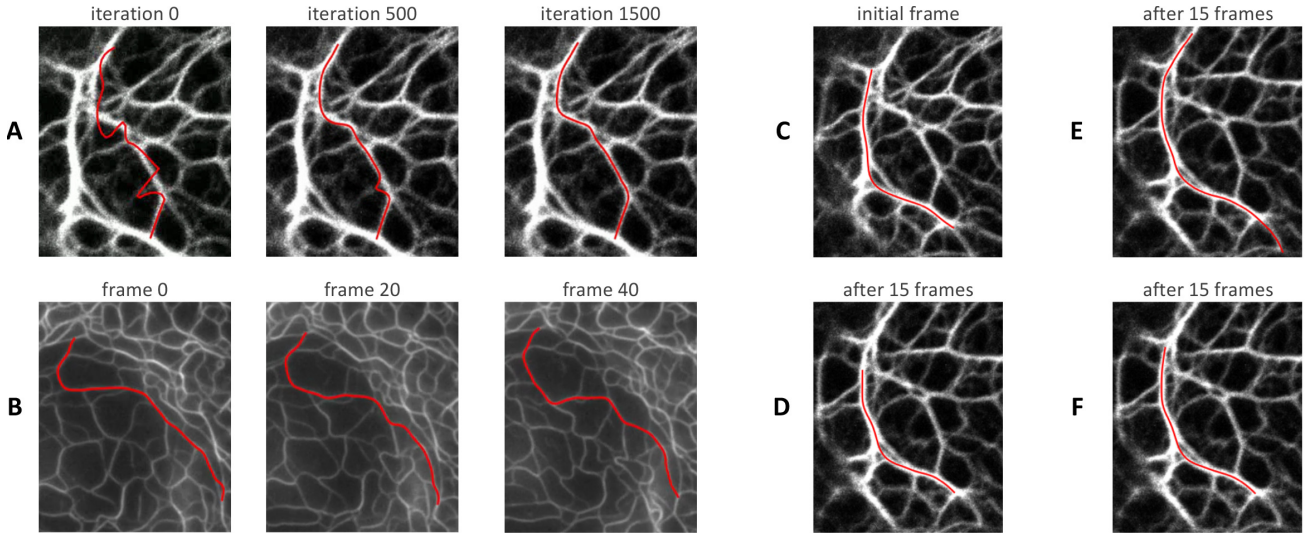


Fig. 1. Life cell imaging of SW13 cells expressing fluorescent HK8-CFP and HK18-YFP proteins (frames were recorded every 30 sec). (A) Snake evolution on a single frame; (B) Tracking result for an individual filament on a time-sequence of 40 frames; (C) Initial position of the snake on the first frame and (D-F) after 15 frames: (D) without stretching term and length constraint; (E) with stretching term only; (F) with stretching term and distance-based potential;

network and allow snake endpoints to be captured by the force field induced by the potential (Fig. 1F).

C. Overall tracking procedure

In our setting, the tracking of individual filaments consists of two main routines: refinement of the position of the filament on the current frame and transition of the filament from the current to the next frame in the sequence. For the second step, we apply pyramidal Lucas-Kanade optical flow computation [1]. It allows to obtain a reasonable fit in case of large deformations of the filament. Incorrect mappings obtained by the optical flow algorithm require the repetition of the refinement step using active contours. Thus, we propose the following tracking procedure (see Fig. 2):

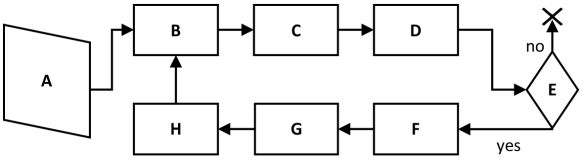


Fig. 2. Block-diagram of the overall tracking algorithm

(A) *Initialization*: The filament is initialized on the first analyzed frame. This can be done manually by user or additional (semi-)automatic segmentation procedures.
(B) *Image preprocessing*: Gaussian smoothing; Hessian ridge detector; gamma contrast correction.
(C) *Calculate the GVF* on the preprocessed image.
(D) *Optimize the position of the snake* on the current image based on the GVF obtained in (C) and take into account a stretching term for open ends [3] and potential for the endpoints.
(E) If the current image isn't the last one in the analyzed sequence, go to the next step. Otherwise, exit the procedure here.
(F) *Calculate the pyramidal optical flow* of the current image with respect to the next image in the time-sequence as described in [1].
(G) Transfer the snake to the next image in the sequence based on the calculated optical flow field.
(H) Select the next image and repeat starting from step (B).

A result obtained by this procedure is depicted in Fig 1. Fig. 1A shows the convergence of the snake on a single frame with an "external energy" as defined above. Fig. 1B shows a filament being tracked in an image sequence of 40 frames.

ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 642866, and the DFG (LE 566/22-1).

REFERENCES

[1] J.-Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the Algorithm," Intel Corporation Microprocessor Research Labs, Tech. Rep., 2000.
[2] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
[3] H. Li, T. Shen, D. Vavylonis, and X. Huang, "Actin filament tracking based on particle filters and stretching open active contour models," *Medical Image Computing and Computer-Assisted Intervention*, vol. 12, no. 2, pp. 673–681, 2009.
[4] H. Li, T. Shen, M. B. Smith, I. Fujiwara, D. Vavylonis, and X. Huang, "Automated actin filament segmentation, tracking and tip elongation measurements based on open active contour models," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009.
[5] D. M. Toivola, P. Boor, C. Alam, and P. Strnad, "Keratins in health and disease," *Current Opinion in Cell Biology*, vol. 32, pp. 73–81, 2015.
[6] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snakes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
[7] T. Xu, H. Li, T. Shen, N. Ojkic, D. Vavylonis, and X. Huang, "Extraction and analysis of actin networks based on open active contour models," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 1334–1340.

Reading of an Analog Liquid Level Gauge on an Oil Platform with a Mobile Robot using 2-D Images

Peter Henöckl[1]

***Abstract*— An approach to automatically read oil platforms' liquid level gauges, originally designed to be read by human operators is presented in this paper. Grayscale image data is acquired from different heights to enhance reliability and minimize deviations due to outdoor influences like reflections and translucence. The position of the level gauge in the scene image is determined, the liquid column is extracted and the level of the liquid is returned using image processing methods.**

I. INTRODUCTION

Measuring the level of a liquid in a container is seen as a solved task. However, as the gauge may not be altered in any way, conventional methods of detection using ultrasonic, magnetic, mechanical, pneumatic, conductive, microwave or capacitive sensors cannot be applied. New optical methods as reciprocally placed photo-LEDs and transistors described in [1] look promising, but as there is no possibility to reliably get behind the gauge, the sensor chosen here is a 2-D camera. To acquire the liquid level of the level gauge a mobile robot (fig. 1) approaches the level and captures the gauge taking images. Although camera based level detection is already greatly described e.g. in [2], not having a closed environment with a correctly positioned bottle on a conveyor belt, brings a big increase in complexity similarly found in [3], [4] and [5]. Coping with different lighting, reflections, backgrounds, objects shining through or alternating weather conditions and locating the level gauge in the scene image brings new additional challenges.

The acquisition of the level is done in three consecutive steps. First the position of the level gauge in the image is determined. To improve the reliability of the following level reading and reduce influences as reflections or translucence of objects in the background, that can be seen in fig. 2, this step is performed repeatedly using different images.

Compared to tests with polarized filters and usage of a flash combined with a very short aperture time of the camera's shutter, using multiple scene images makes the biggest enhancement in readability and reproducibility of the same quality. Images of the same level gauge are taken from different angles. As the level is a horizontal feature, horizontal disturbances have a much higher influence than vertical ones and are to be compensated. To achieve this the height of the camera is altered. Secondly warped images of the liquid column are created giving optimal conditions for level detection. Based on these images an estimation of the level follows accepting multiple level hypotheses for each column.

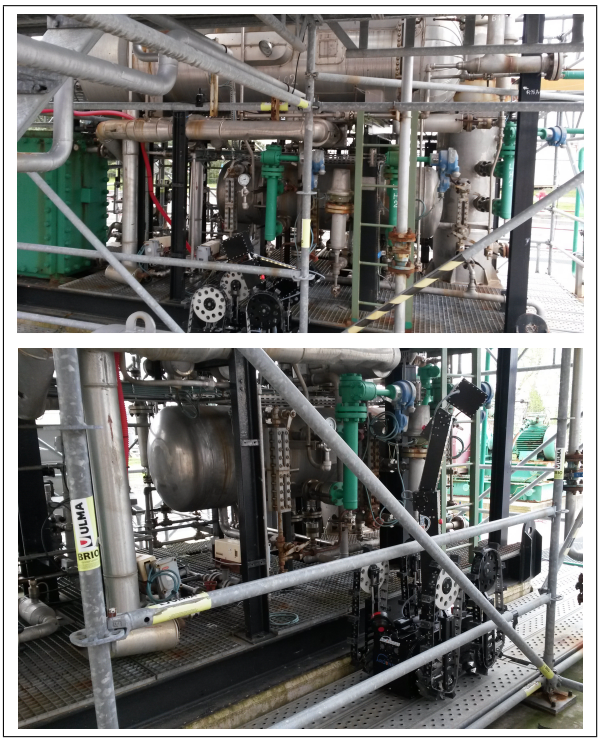


Fig. 1. Mobile robot approaching the liquid level gauge and positioning its arm to take a picture for level detection using image processing.

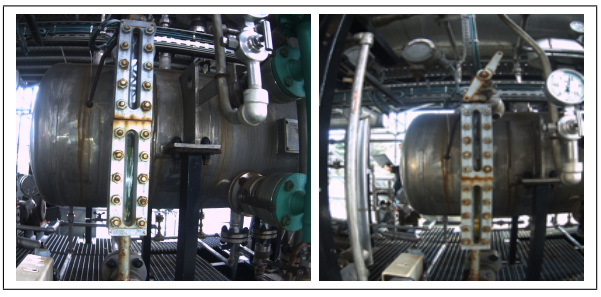


Fig. 2. Reflections (left) and translucence of a pipe behind the liquid level gauge (right) are examples for challenges for a correct reading.

II. METHODS

Fig. 3 shows the structure of the proposed method. Due to preprocessing steps in other parts of the overall code, grayscale images are the basis for the detection. Changing colors of the liquid in the gauge make RGB images of minor importance. The position of the level gauge in the 3D space is known and the pose of the mobile robot and its arm can be

[1]Peter Henöckl is with Faculty of Electrical Engineering, Automation and Control Institute, TU Vienna, 1040 Vienna, Austria peter.henoeckl@gmx.at

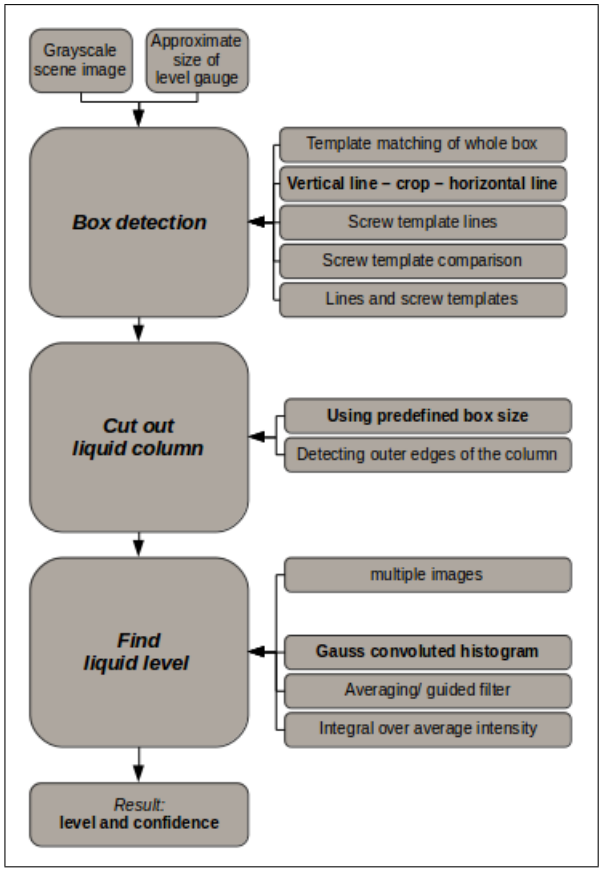


Fig. 3. Structure of level detection

approximated quite well using sensor data. Therefore the size of the box can be estimated and used for further processing.

A. Box Detection

As the outer box of the level gauge has hardly any unique features, tests applying feature-based algorithms like SIFT did not bring the results intended. To robustly find the correct location of the box a combination of five approaches is used. For implementation on the robot all five are combined comparing their returned edge points of the box as well as their confidence values. Thereby maximum knowledge of the accuracy of the detected box position in the image can be achieved, that is crucial for further processing and for a final output of a confidence value of the reading.

The straight forward approach is template matching, using an image of the whole outer box as a template. The box template is resized using the height of the box in the scene image. Fig. 4 shows, that this works nicely for scenes, where the expected height has just minor deviations from the real one and the image is taken frontal or is made looking like a frontal image by warping in detection preprocessing. This keeps the influences of distortion and rotation low. Furthermore lighting conditions should be similar.

The second approach focuses on detecting the outer lines of the box based on Canny edge detection [6] and Hough line detection [7]. Varying the parameters of edge detection and just taking the hough lines that are roughly vertical,

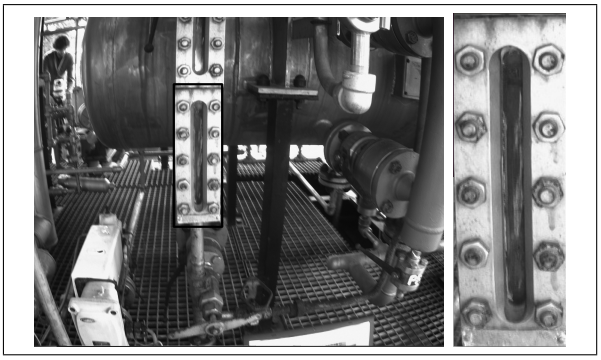


Fig. 4. Detecting the outer box with basic template matching works for ideal conditions.

i.e. within a certain angle threshold, gives possible lines for the left and right border of the outer box. Using further knowledge about the structure of the level gauge the final vertical border lines are found. The upper and lower line of the box are not as present in the image. The horizontal lines containing the most points in the Canny image rarely concur with those vertical box borders. To overcome this, the scene image is cropped on the left and right side using the found vertical borders. As the grayscale image recieved from prepocessing steps is often warped, there sometimes occurs a black part at the top and bottom of the scene image. If that is the case, the horizontal lines standing out most are the transitions between the real image and the black parts. To solve this, the black parts at the top are filled with the same intensity as the uppermost pixels of the real scene, that can be seen in fig. 6. The black parts at the bottom are filled with the same intensity as the pixels at the bottom of the real scene image. In the new image the protruding horizontal lines are the upper and lower box borders. To make sure to correctly distinguish the horizontal borderlines from other remaining horizontal lines within the cropped image, again knowledge about the structure of the liquid level gauge is used. The four intersections of the vertical and horizontal border lines are returned as box edge points. If the box dimensions are given, they provide a further constraint to reliably detect the vertical as well as the horizontal lines and find the correct borders.

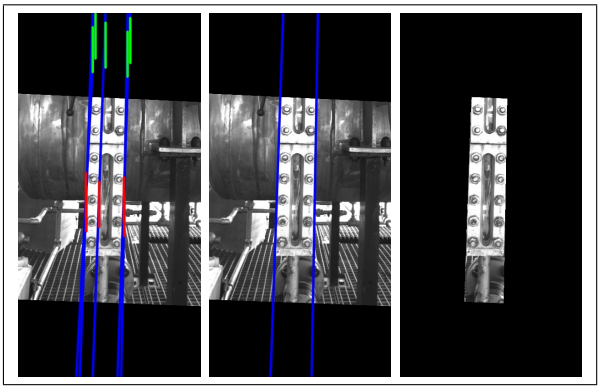


Fig. 5. Attain vertical borders of the box

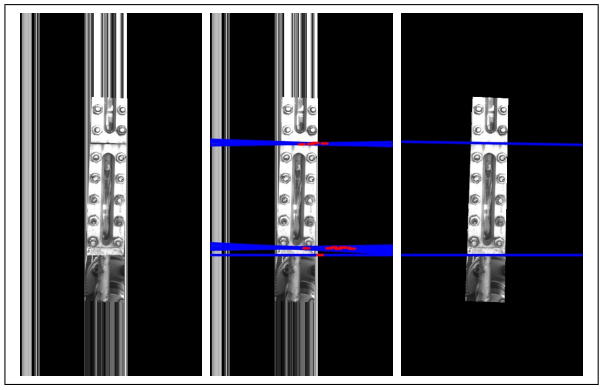


Fig. 6. Attain horizontal borders of the box

In the third approach that can be seen in fig. 7 advantage is taken of the texture of the level gauge. Besides the metal box and the liquid column in the middle it consists of ten big screws arranged in two vertical lines. A set of screw images is used for template matching and detecting possible screws in the scene. The concept is to deliberately look for more than ten screws and then classify them into so called good screws, that do belong to the gauge, and bad screws, that do not. This is done by creating a new black image, where the center of every found screw is marked as a white pixel. If a pixel is already white, the one below is made white instead to make it count. Afterwards Hough line detection is applied in this binary image to find lines of screws. The two lines with most participating pixels are used to finally determine the position of the box. Knowledge about the maximum amount of screws or about their similar vertical distance can be used to optimize the result.

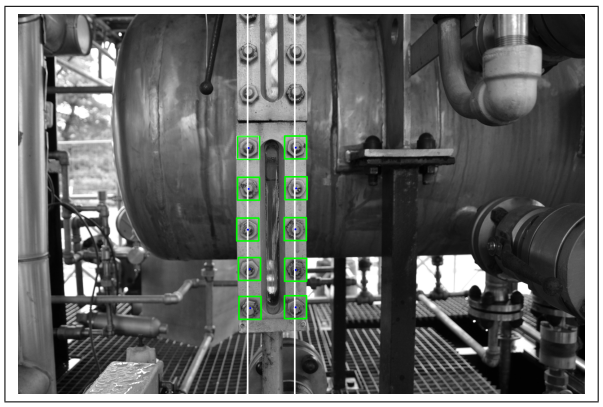


Fig. 7. Attain the position of the outer box by using template matching to find screws of the box. More than the existing 10 screws are to be found to then form lines of screws.

If the height of the box is given, the fourth approach can be applied. Similar to the third approach white dots are created in a black image for found screw templates. However, the white dots for found screws are made bigger and compared to an image created in the algorithm, consisting out of ten big white filled dots. Those are placed exactly on the spots, where a level gauge of the given size has located its screws. The comparison is done by sliding the artificial ten dot image over the scene image and adding one to the correlation variable for each pixel that is white in both images. The point with the highest correlation marks the estimated position of the level gauge's outer box.

To combine the strengths of the algorithms mentioned above, the fifth option is based on line detection of box borders and template matching with screws. Instead of getting just the two best vertical lines, more of them are to be found implementing a Canny edge detector and Hough transform. Next screw templates are found in the scene. Lines, as well as screws are then graded identifying their relative horizontal distances. There have to be a certain number of screws in the vicinity of a line, to mark both of them as good. Fig. 7 shows, how finally good screws and lines are marked in green, other discarded ones in blue and bad ones in red.

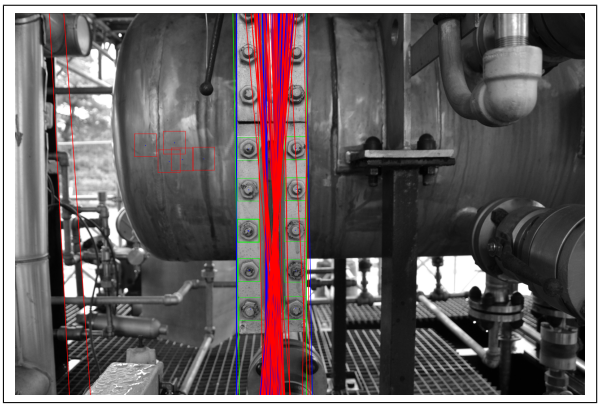


Fig. 8. Combine the use of template screws and line detection to optimize the result.

B. Cut-out of Liquid Column

The combination of the box detection methods above lays the foundation for localizing the inner liquid column and cutting it out. As the result image of the box detection contains just the box, the position of the inner liquid column is acquired using height and width of the column in respect to the box size. As the appearance of the box is known, the outer edges of the column are searched for in a specific area, to get detailed borders.

C. Level Detection

Having acquired and cut out the liquid column, the level is obtained. Allthough there are many different kinds of disturbances when detecting the liquid level, reflections and translucence are the ones affecting the reading the most, as described in fig. 2. Horizontal reflections of the sun or nearby objects create horizontal lines, that often are even more prominent than the real water level. Pipes or other objects behind the liquid level gauge also create horizontal gradients in the intensity image of the liquid column that is used to obtain the correct level. To overcome these disturbances, images of the gauge are taken from different heights.

As the mobile robot's arm is restricted to five degrees of freedom, the normal pose of the camera mounted on the arm has to be altered. To achieve readings from different heights,

the camera has to turn around its lateral axis. Beside this pitching movement it needs to move along the vertical axis as illustrated in fig. 9. These movements result in images taken from different heights, where the reflections and objects behind the liquid level gauge move vertically in respect to the liquid column itself. However, the level of the liquid remains at the same height within the column (see fig. 10).

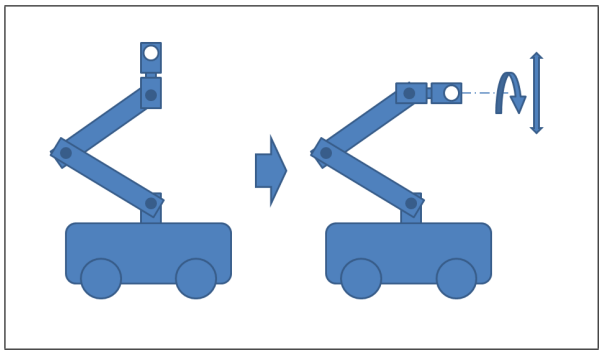


Fig. 9. Robot pose to achieve readings of the liquid level gauge from different heights (Camera has to turn around the lateral axis, i.e. pitch and has to move along the vertical axis.)

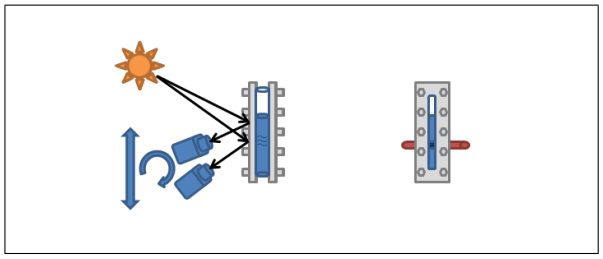


Fig. 10. Taking images from different heights to make reflections and translucence of objects behind the gauge move vertically while the actual liquid level remains at the same vertical position

As for box detection, reliability of the reading is of higher importance than speed. Hence three algorithms for level detection are performed and combined resulting in a final value and confidence.

The obvious method is the detection of the most outstanding horizontal line. Canny edge detection is used, Hough transform is applied and only lines within a certain angle threshold are considered. However, this first part of level detection is not restricted to find just one line, but multiple ones. The reason for accepting multiple level hypotheses for one column is that there can be found horizontal lines within the column that have nothing to do with the real liquid level.

The y-positions of the detected level hypotheses are then normalized between 0 and 100 and subtracted from 100 to get the liquid levels in percent. The whole range from zero to a hundred percent is divided in equally sized intervals and a histogram is created. Each level hypothesis in the histogram is then convoluted with a Gauss function. This takes into accout that there might be slight deviations of the real level position in the liquid column images that are cut out in the box image, that is detected and cut out of the original scene image. Fig. 11 shows that the histograms with Gauss filtering are created out of every column image and summed up. To optain the real liquid level, the position of the maximum of the resulting function is found.

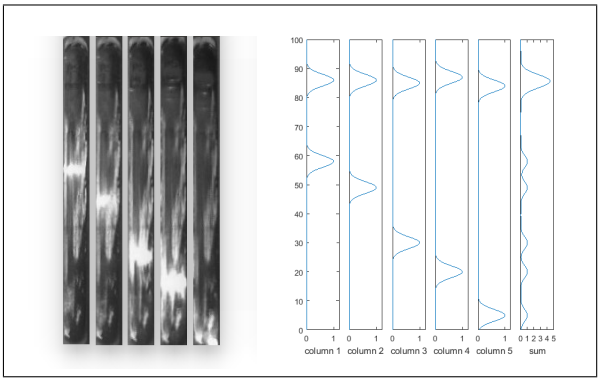


Fig. 11. Liquid columns with reflections at different heights. The strongest gradients are represented with Gauss functions and added. The final level is obtained by finding the maximum of the sum of the functions describing the columns.

The second option gets the average intensity for every horizontal row of pixels in the column image. This array of intensity values having the size of the number of rows in the column image is smoothed with a filter and the following level detection algorithm is performed. All intensity values are normalized to have a maximum value of 1. Starting from the top, a separator divides the intensity values in two parts. Then two integrals are obtained. On one side the integral above the curve is used, on the other side of the separator the integral below is used. As the lower part of the liquid column contains the liquid, it is expected to have the higher intensity. Nevertheless, it is also done the other way round to achieve safe results. The sum of the two integrals is stored in a new array for each position of the separator. The position, where the sum becomes a maximum is selected as the result for the level detection.

III. Experiments

A. Box Detection

Images are taken by the robot outdoors under very different weather conditions. Evaluating the template matching approach, it can be shown, that the more the illumination and weather resemble the conditions on the template image, the better it works.

When performing approaches three to five, it becomes obvious, that according to different sized boxes in the images, the template screws have to be adapted, thus resized to perfectly fit the screws in the scene image. To cover not just frontal images of the level gauge but also those taken from slightly above and slightly below the height of the box center, also screw templates have to be chosen accordingly. The set of templates has to contain screws photographed from different angles.

When running the different algorithms for box detection, it showed that every single one of them has its strengths and weaknesses. Different approaches perform best depending on illumination, weather condition, distance of the camera to the

level gauge or resolution of the image. As the correctness of the reading and the declaration of the confidence of the final value are of particular importance, performing different approaches and a subsequent comparison are worth the additional time needed.

Running tests of the algorithm on the real robot on an oil platform training site it became apparent, that the underlaying algorithm that takes images of the level gauge returns images with low deviations of the box position from the image center. On average the probability of the box being close to the center of the scene image is much higher than it being close to the edge of the image. Taking this into account, the confidence of the detected box being correct is multiplied with an additional function

$$1 - \frac{c}{100}\sqrt{\frac{(w/2-x)^2+(h/2-y)^2)}{(w/2)^2+(h/2)^2}}$$

where w is the width and h is the height of the scene. x and y define the center-point of the found box. c is a constant giving the percentage of how much the confidence is lowered if the box center is in one of the corners of the image, i.e. the box center-point with the biggest distance to the scene center that is still in the image.

Using screw templates worked best for high resolution images and only slight differences in size and illumination. The approach for box detection based on finding the correct border lines outperformed this method when the image had a low resolution. Fig. 12 shows the results of box detection for six different images, that are used to get the correct waterlevel. Fig. 13 shows the cropped and warped boxes, for later getting the column images.

B. Cut-out of Liquid Column

Tests have shown that the precision and correctness of cutting out the liquid column of the level gauge highly depends on the preceding box detection. Having detected the outer box with an error less than ten percent of its width, ensures a high probability of getting a correctly cut out liquid column as a basis for the following level detection.

C. Level Detection

After testing with a halogen work light serving as an artificial sun and a model of the level gauge, images taken in the real environment with sunlight show the same results. As images of the level gauge have been taken from five or six different heights, they now have to be arranged in a way they can be compared to each other. Using one column image as reference the other column images are fitted by subtracting intensities while slightly shifting the column image to fit. Multiple checks with small changes in size improve the result.

Applying the Histogram-Gauss-Adding method on images that are taken from different heights delivers good results. To verify the algorithm also sets of images with many scenes with high intensity gradients at the same height are tested. Fig. 14 shows the liquid columns cut out of the boxes in

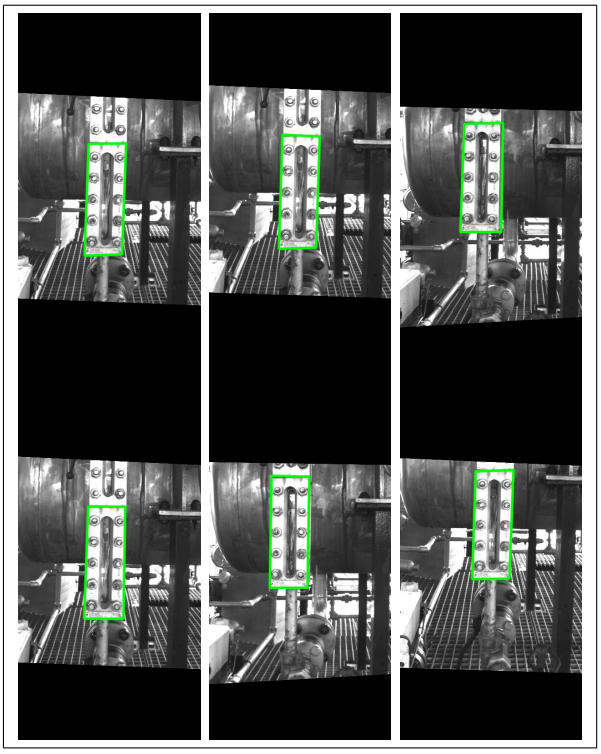


Fig. 12. Box detection in multiple images of the scene that are used to obtain the liquid level

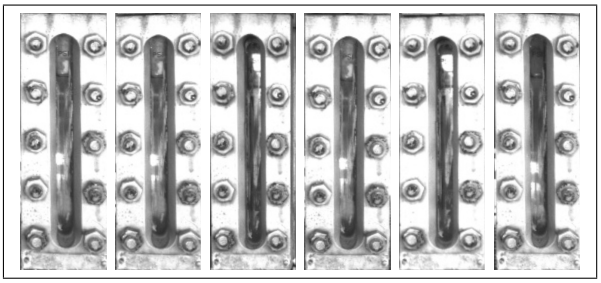


Fig. 13. Resulting box images for liquid column extraction

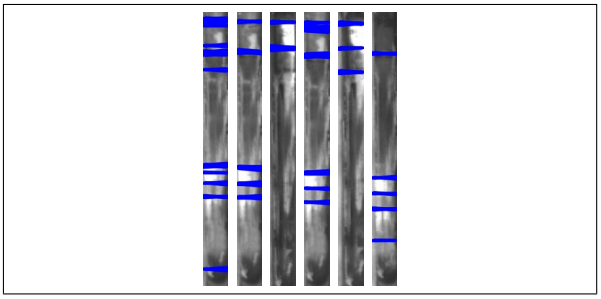


Fig. 14. Extracted liquid columns with multiple level hypotheses (Despite the fact that four out of six images have similar positions of the reflections the correct level is still found.)

fig. 13, that originally belong to the scenes in fig. 12. It can be seen that there are a lot of wrong level hypotheses at approximately the same height. However, as more column images contain correct level hypotheses, the correct ones predominate and the real level is returned.

Test have shown that the Histogram-Gauss-Adding method described above mostly outperforms approaches like creating an average image as in fig. 15 using

$$i_x = \frac{1}{n}(i_{x_1} + i_{x_2} + ... + i_{x_n}).$$

On the right side of the original image, images show the same situation photographed from different heights with the column already identified and cropped. In the following images the original columns are added equally weighted with linear blending. First the first two columns are added, then the first three, then the first four and in finally all five of them are put into one single image. The found level is marked in red. This clearly shows, that reflections can be suppressed using multiple images taken from different heights. Further improvements are reached using a guided filter. The middle one of the column images is used as a guidance image for this edge preserving filtering method. However, this approach only works for perfectly alligned images. Therefore it is just used to raise the confidence of the reading, if it delivers similar results as the method using Gauss functions.

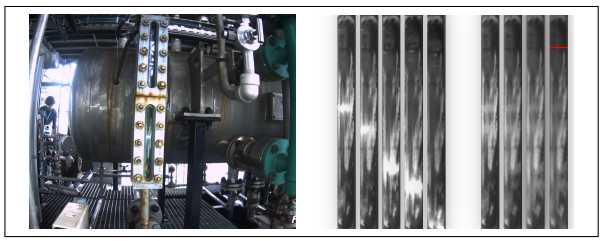


Fig. 15. Scene image with reflections in the liquid column, extracted liquid columns from images taken from different heights and addition of 2,3,4 and 5 column images

Doing tests to compare the different level detection algorithms it can be seen that everyone of them has its advantages that make it reasonably useful for a reliable detection. Line detecting algorithms have their strengths in transparent liquids similar to water. The integral over the intensity array performs best for liquids that give a big intensity difference compared to the empty part of the liquid column.

IV. CONCLUSIONS

The initial task of detecting the liquid level of an analog gauge was reached using an algorithm for locating the outer box in the image, based on canny edge detection, hough line detection and template matching. The level was then obtained identifying the horizontal gradients standing out most. The crucial enhancement of the reliability of the process was achieved using multiple images and creating a sum of Gauss functions, each at the position of a level hypothesis. Disruptive effects of sunlight, rain and even objects like pipes shining through can be handled. Despite being cheaper and easier to implement than solutions with flashlight, polar filters or spectral filters, the reached confidence value of the reading can be increased drastically by a small additional arm movement of the robot, taking multiple images. Future detection algorithms may base on this approach to detect other kinds of reflective objects in outdoor conditions.

REFERENCES

[1] E. Musayev, S. E. Karlik, "A novel liquid level detection method and its implementation" *Sensors and Actuators A: Physical*, vol. 109, pp. 21–24, Dez. 2003.

[2] K. J. Pithadiya, C. K. Modi, J. D. Chauhan, "Machine Vision Based Liquid Level Inspection System using ISEF Edge detection Technique" in *Proc. International Conference and Workshop on Emerging Trends in Technology ((ICWET'10)*, Mumbai, India, Feb. 2010, pp. 601–605.

[3] S. Park; N. Lee; Y. Han; H. Hahn, "The Water Level Detection Algorithm using the Accumulated Histogram with Band Pass Filter" *World Academy of Science, Engineering & Technology*, issue 32, p. 193, Aug. 2009.

[4] T. Hies, P. S. Babu, Y. Wang, R. Duester, H. S. Eikaas, T. K. Meng, "Enhanced water-level detection by image processing" in *Proc. 10th International Conference on Hydroinformatics*, Hamburg, Germany, Jan. 2012.

[5] M. Iwahashi, S. Udomsiri, "Water Level Detection from Video with Fir Filtering" in *Proc. IEEE 16th International Conference on Computer Communications and Networks (ICCCN'07)*, Honolulu, Hawaii, USA, Aug. 2007, pp. 826–831.

[6] L. Ding, A. Goshtasby, "On the Canny edge detector" *Pattern Recognition*, vol. 34, pp. 721–725, Mar. 2001.

[7] R. O. Duda, P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures" *Communications of the ACM*, vol. 15, pp. 11–15, New York, USA, Jan. 1972.

Novel Human Machine Interaction with Sticky Notes for Industrial Production

Gernot Stuebl, Thomas Poenitz, Harald Bauer, and Andreas Pichler[1]

***Abstract*— In this paper we present a 3D documentation system which utilizes new human machine interaction concepts on the example of virtual sticky notes. Using different tracking techniques the virtual notes can be attached to a physical object and are displayed on a tablet in an Augmented Reality way. The main intention is to strengthen the interplay between construction and production of industrial machines as the virtual notes are synchronized with a production lifecycle management system.**

I. PROBLEM DESCRIPTION

An essential part of machine manufacturing is the interplay between construction and production. Often this connection leaks information in both ways: the construction team changes details in the last minute, while in production things are mounted in a different order or way as it was intended. Since in the end both sides have to synchronize their knowledge this results mainly in a mass of notes often stuck on the machine or even worse written on the machine itself. This industrial spotlight paper presents a novel development integrating latest technologies to manage position based notes digitally in an Augmented Reality based way.

II. STATE-OF-THE-ART

Although in media one can see photos of tablets showing Augmented Reality (AR) overlays of shop floors, industrial grade AR documentation systems are rare. To our knowledge the most similar system available to our proposition is Docufy [1]. It is an AR interface to a dedicated content management system and used to display technical content like manuals in a read-only way. The mapping of the data to 3D is managed via a manual registration step of interest points originating from a priori known CAD data. When the tablet is moving, the interest points are tracked and the 3D data adapts to the new viewpoint.

III. PROPOSED SYSTEM

In this paper we propose a system which enables a user to view, edit and add virtual sticky notes to a machine during assembling by using Augmented Reality techniques. The notes are position dependent and synchronized with the 3D database of a production lifecycle management (PLM) system. Since the construction team mainly works with the PLM system, they immediately have access to the information and may modify existing or add new notes directly.

[1]PROFACTOR GmbH, 4407 Steyr-Gleink, Im Stadtgut A2, Austria {Forename.Surname}@profactor.at

A. Tracking System

For the tracking we pursue a multi-modality strategy, which utilizes a combination of

- a commercially available infrastructure-based tracking system,
- fiducial Augmented Reality markers,
- and a visual real-time tracking system.

The infrastructure-based tracking system is the main source of 6 degree-of-freedom (DOF) data for the system. See Figure 1 for a tablet enhanced with a tracking system receiver. The two major extensions to the state-of-the-art

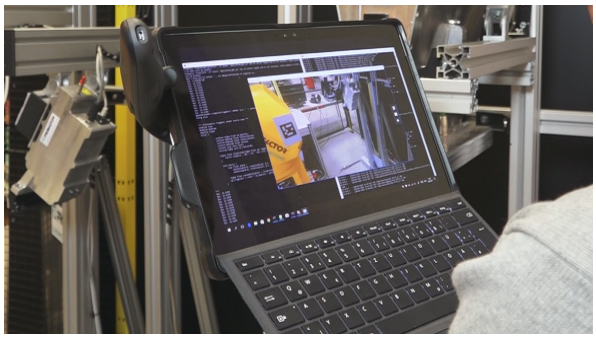


Fig. 1. Tablet with attached tracking system receiver on the left side. This enables a 6 degree-of-freedom positioning in space.

concern the tracking part as well as the way a user can interact with the system.

The initial registration is done with an AR marker system based on the work of Garrido-Jurado et al. [2]. Fiducial markers are preferred to feature based approaches as the presence of stable features on industrial objects cannot be guaranteed. After that the tracking data is transformed to the coordinate system of the machine's CAD model which is provided by the PLM software. This allows the attachment of notes to positions in the CAD data.

A special approach is required for movable parts of the machine, like panels. When mounted to the machine, their relative positions to the base CAD data can be determined using attached AR markers.

An additional feature was built in to handle unmounted sub-assemblies. They can be annotated like any other machine parts, however they have to be pre-identified in a manual step. This is the input of a real-time tracker which is an extension of Akkaladevi et al. [3].

When a user likes to add a new note to the machine he/she has two possibilities to define the position: either the desired position is touched with the measuring tip mounted on the

backside of the tablet or an additional AR marker is put on the desired position, see Figure 2. The additional marker

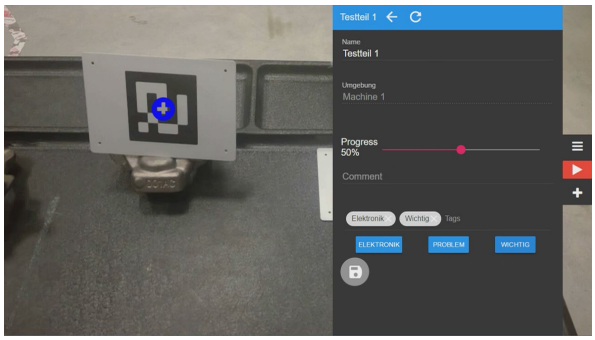


Fig. 2. Screenshot of position input via an additional marker in the image. The position of the marker defines the position of the new sticky note. The pane on the right shows pre-defined tags connected with the marker.

can be removed after the system took over its position. While the measuring tip might be the more intuitive way to define a position, AR markers have an advantage: they can be combined with note-templates, e.g. different markers for different users or different types of notes.

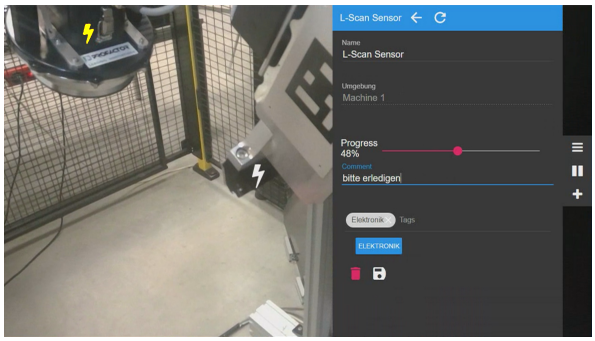


Fig. 3. Screenshot with two sticky notes in the image (flash icons on the left). The top left note is selected and the pane on the right displays the according information.

B. Human Machine Interaction

The base of the human machine interaction is a live view of the tablet's camera in which the sticky notes are overlaid in an Augmented Reality fashion. When touching a note, a form with additional information pops up, see Figure 3. From a technologically viewpoint this is a HTML overlay, reflecting the client-server based architecture of the software.

Beside the standard information for a sticky note like name, type, and description an additional tagging system has been implemented for a flat hierarchy of the notes. To each note one ore more pre-defined tags can be assigned, which allow an easy to use filtering, e.g.: showing only electronic related notes in the view. Examples of the tags can also be seen in Figure 2 and 3 on the bottom right.

Furthermore the system supports different user roles, which differ in the amount of information they can edit and/or which sticky notes they can see.

IV. TECHNICAL EVALUATION

The system is currently in an evaluation phase. Tests regarding positioning showed up accuracies of ± 0.5cm in a volume of $5\text{m} \times 5\text{m} \times 5\text{m}$.

The used hardware is a standard Microsoft surface tablet with the internal camera set to a resolution of 1280×720px and a maximum achievable frame rate of 25fps of the overlay. However the limiting factor for the frame rate is the tablet camera itself. With an external industrial camera frame rates up to 100fps have been achieved.

For the registration step and the moveable parts also the marker size and camera-marker-distance have been evaluated: for a reasonable marker size of 7cm the largest distance to the marker is 2m. With higher distances the detection and therefore the visualization becomes in-stable.

V. CONCLUSION AND FUTURE WORK

In this paper we presented an industrial grade AR based documentation system which extends the current state-of-the-art by integrating additional tracking modalities and furthermore allows the user to edit the information in an intuitive way. The proposed system is currently on a technology readiness level of 7. The next steps are extensive field tests. Although the system can be used stand-alone, the ideal synergy is together with a PLM system. Currently the import/export functions support only one such system, however this will be extended in future.

Furthermore of vital interest is the evaluation of time savings gained by using our system. For this a detailed study will be set up together with industrial partners.

From an application point of view a possible extension could be the replacement of the PLM input with a Virtual Reality (VR) system. This would allow one user to add information on a model in VR which is then immediately shown to a different user on the real object. This could be the base for numerous multi-user scenarios e.g. the usage as remote maintenance system. The advantage over a traditional 2D camera assisted remote maintenance system would be the exact 3D positioning of the information on the object of interest.

ACKNOWLEDGMENT

This research is funded by the project SIAM (FFG, 849971) and by the European Union in cooperation with the State of Upper Austria within the project Investition in Wachstum und Beschäftigung (IWB).

REFERENCES

[1] RE'FLEKT. Kothes / Docufy - Virtually enhanced handbooks with AR intergrated into your editing system, 2016. https://www.re-flekt.com/portfolio-view/augmented-reality-technical-documentation/.

[2] S. Garrido-Jurado, R. Muñoz Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recogn.*, 47(6):2280–2292, June 2014.

[3] S. Akkaladevi, M. Ankerl, C. Heindl, and A. Pichler. Tracking multiple rigid symmetric and non-symmetric objects in real-time using depth data. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5644–5649, May 2016.

Image Registration and Object Detection for Assessing Unexploded Ordnance Risks - A Status Report of the DeVisOR Project*

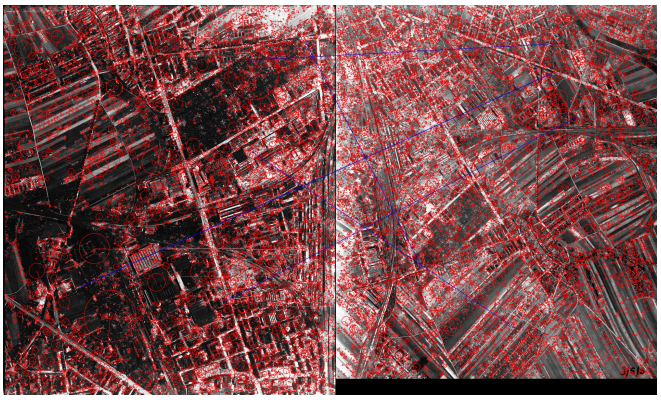
Simon Brenner[1], Sebastian Zambanini[1] and Robert Sablatnig[1]

I. INTRODUCTION

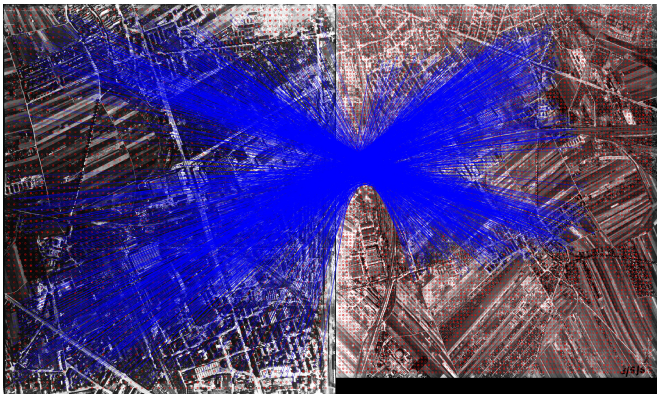
Although the last acts of war in Central Europe date back to the times of World War II, Unexploded Ordnance (UXO) from that period still poses a serious hazard for population and construction projects [3]. For a preliminary estimation of UXO risks, specialized companies retrieve and interpret aerial images from WWII surveillance flights over the area of interest. This process includes the registration of historic aerial images to modern satellite images, and the detection and mapping of certain objects that indicate increased combat activity in the surveyed area. Currently, these tasks are performed in time-consuming manual work. The DeVisOR project, which was started in 2016 as a cooperation between the Computer Vision Lab and the Information Engineering Group (TU Wien), as well as the Luftbilddatenbank Dr. Carls GmbH as an industrial project partner, aims at supporting the above named tasks with computer vision and visualization techniques. This paper gives a half-time status update of the project achievements as well as an outlook for the final year.

II. IMAGE REGISTRATION

The registration of WWII aerial images to modern satellite images is particularly challenging because the landscape has changed drastically in the course of seventy years. Not only buildings and roads, but also vegetation, agricultural use and the courses of rivers may have changed, so that it becomes difficult to find reliable common features [4], [5]. Additionally, the available images are partly in suboptimal condition. We therefore propose a semi-automatic framework for the registration process, in which first the easier task of registering the historical images among each other is performed automatically. Due to the varying conditions even among the historical images (seasonal changes, weather, destruction, image noise) and the absence of *a priori* information about their relative rotation and translation, only feature-based registration methods, such as SIFT [2], are applicable. We found that automatic scale space feature detection is too unstable for the given image data; however, for each image the approximate aircraft altitude and the focal length of the camera is known. We can therefore normalize the scales of the images and perform a dense sampling of features at a fixed scale, which significantly improves the matching stability. Figure 1 shows an example. To refine the resulting registration and account for parallax effects resulting from uneven terrain and different capturing angles, we successfully applied a deformable fine registration approach, that was originally designed for the registration of multi-modal medical data [1].



(a) Scale space extrema



(b) Densely sampled features

Fig. 1: Comparison of feature matching stability

Guided by an interactive visualization of the registration results, the user can then select the most suitable historical image and manually georeference it; all the other images are then registered transitively.

We are also working on a novel registration algorithm that is currently able to register about a third of the WWII images in our test data set directly to modern satellite images and thus supplement the above named framework.

*This work is supported by Austrian Research Promotion Agency (FFG) under project grant 850695

[1]Simon Brenner, Sebastian Zambanini and Robert Sablatnig are with Faculty of Informatics, Institute of Computer Aided Automation, Computer Vision Lab, TU Wien, 1040 Vienna, Austria sbrenner@caa.tuwien.ac.at, zamba@caa.tuwien.ac.at, sab@caa.tuwien.ac.at

III. OBJECT DETECTION

An UXO risk for a region of interest is derived from various indicators of combat activities on historical aerial images. These could be destroyed buildings, anti-aircraft artillery positions, trenches or bomb craters; the latter ones are by far the most numerous and simultaneously the most difficult to reliably identify on aerial images, as they can easily be confused with other small round objects such as trees [3].

The development of strategies for automatic detection of such combat indicators is scheduled for the current year. We are planning to adapt state of the art machine learning approaches to the problem; we hope to be able to exploit the fact that typically a time series of registered aerial images is available for the region of interest. As the task at hand is a critical one, a human expert will always be required to validate and refine the results. We will thus, just as for the registration problem, aid the user with an interactive visualization component for parameter exploration.

IV. IMPLEMENTATION

In order to maximize both the benefit to our industrial project partner and the usage and testing of our methods, we have been developing software tools that blend in to their daily workflow, namely in the form of plug-ins for their preferred GIS software. The first working prototype of the registration component was delivered in February 2017 and tested in both the German and Austrian branch of the Luftbilddatenbank GmbH. Apart from minor bugs and usability issues, the overall feedback was positive and encouraging.

REFERENCES

[1] M. P. Heinrich, M. Jenkinson, M. Bhushan, T. Matin, F. V. Gleeson, S. M. Brady, and J. A. Schnabel, "Mind: Modality independent neighbourhood descriptor for multi-modal deformable registration," *Medical Image Analysis*, vol. 16, no. 7, pp. 1423–1435, 2012.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[3] S. Merler, C. Furlanello, and G. Jurman, "Machine learning on historic air photographs for mapping risk of unexploded bombs," in *Proceedings of the 13th International Conference on Image Analysis and Processing*, ser. ICIAP'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 735–742.

[4] V. Murino, U. Castellani, a. Etrari, and a. Fusiello, "Registration of very time-distant aerial images," *Proceedings. International Conference on Image Processing*, vol. 3, pp. 989–992, 2002.

[5] S. Nagarajan and T. Schenk, "Feature-based registration of historical aerial images by area minimization," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 116, pp. 15–23, 2016.

FORMS – Forensic Marks Search*

Manuel Keglevic and Robert Sablatnig[1]

Abstract— **The goal of the project FORMS is to support the search and comparison of toolmarks by forensic experts with a semi-automatic system in order to identify and solve connected criminal cases. The proposed methodology uses a neural network with triplet architecture to compute similarities between toolmark images. Further, to allow an accurate evaluation under real-world conditions a dataset consisting of more than 3000 images of cylinder locks with toolmarks from real criminal cases is created as part of the project.**

I. INTRODUCTION

Lock snapping is a common way for forced entry in Europe. The unique imprints of the pliers used for these break-ins significantly support the investigation of such offenses and are crucial as evidence in the following court cases. However, manual examination of these toolmarks in order to find multiple uses of the same tool is a time consuming task due to the amount of samples. Therefore, the goal of the project FORMS (Forensic Marks Search) is a two-fold solution for this problem: firstly, an application which allows for search and comparison of toolmark images stored in a centralized database. Secondly, a methodology based on state-of-the-art machine learning techniques for an automatic by similarity in order to reduce the amount of images requiring manual examination.

The project started in Fall 2015 and is funded by the Austrian Security Research Programme KIRAS. The project partners are the Computer Vision Lab of the TU Wien, the Bundeskriminalamt (Criminal Intelligence Service Austria), the CogVis GmbH, and VICESSE.

II. TOOLMARK DATASET

Since the validity of comparative forensic examination of toolmarks has been challenged in court, various papers have been published on the comparison of toolmark images [5]. This lead to the development of methodologies for the automatic comparison of striated toolmarks and datasets like the NFI Toolmark Dataset published by Baiker et al. [1].

However, in contrast to forensic images of toolmarks from real criminal cases, these toolmarks were created in constrained environments. Therefore, to allow an evaluation of the real-world performance of toolmark comparison methods, a new dataset was created as part of the FORMS project. This dataset, created by photographing cylinder locks seized during criminal investigations using a microscope, consists of approximately 3000 toolmark images from about 50 different crime series. In order to investigate the influence of lighting each of the 154 cylinder was photographed on both sides under 11 different lighting conditions. In Figure 1 a broken lock cylinder with toolmarks is shown on the left side. The appearance of the toolmarks can vary heavily due to material differences in the material, the force applied or the lighting conditions. For example in Figure 1 on the right the appearance difference due to varying depth of the toolmarks is illustrated.

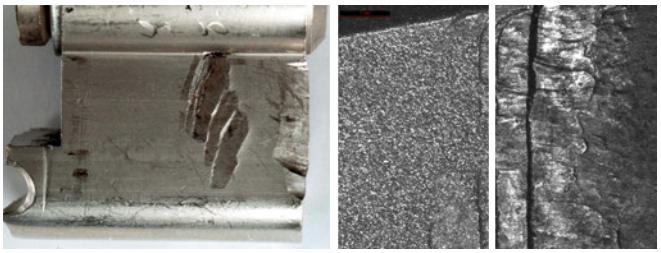


Fig. 1: Image of a broken lock cylinder with toolmarks created by a locking-plier (left). Matching toolmarks on two lock cylinders photographed using a comparison microscope with a magnification factor of 20 (right).

III. METHODOLOGY

As shown in Figure 1 on the right side, extracting foreground (the toolmark) from background (lock cylinder) is challenging due to varying background structure depth of the toolmark. Therefore, the region of interest is marked by the forensic expert by hand in a first step. Local image patches extracted in these regions of interests are then compared using a neural network. The network architecture used is based on triplet learning which has for instance been applied to face detection [4] and local image patches [2]. Further, Keglevic and Sablatnig showed [3] that it can be used to compute similarity measures for striated toolmarks. To capture the unique properties of this problem like varying lighting conditions and background the neural network is trained from scratch. In order to create the necessary training data a ground-truth tool was created as a plugin for the image viewer nomacs[2]. This tool allows the definition and pixel perfect alignment of matching polygons in toolmark images. Using these annotations matching patches for the training and evaluation process can be created along these matching polygons. First results show promising result, however for an in-depth assessment of the performance an evaluation has to be performed as soon as the whole dataset is annotated.

*This work has been funded by the Austrian security research programme KIRAS of the Federal Ministry for Transport, Innovation and Technology (bmvit) under Grant 850193.

[1]Manuel Keglevic and Robert Sablatnig are with the Computer Vision Lab, TU Wien, mkeglevic@caa.tuwien.ac.at

[2]https://github.com/nomacs/nomacs-plugins

ACKNOWLEDGEMENTS

This work has been funded by the Austrian security research programme KIRAS of the Federal Ministry for Transport, Innovation and Technology (bmvit) under Grant 850193. We would like to thank the forensic experts of the Criminal Intelligence Service Austria and the LKA Wien (AB08 KPU) for their help. The Titan X used for this research was donated by the NVIDIA Corporation.

REFERENCES

[1] M. Baiker, I. Keereweer, R. Pieterman, E. Vermeij, J. van der Weerd, and P. Zoon, "Quantitative comparison of striated toolmarks," *Forensic Science International*, vol. 242, pp. 186–199, 2014.

[2] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, "PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors," *ArXiv*, 2016.

[3] M. Keglevic and R. Sablatnig, "Learning a Similarity Measure for Striated Toolmarks using Convolutional Neural Networks," in *Proceedings of the 7th IET International Conference on Imaging for Crime Detection and Prevention (ICDP)*, 2016.

[4] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[5] R. Spotts, L. S. Chumbley, L. Ekstrand, S. Zhang, and J. Kreiser, "Optimization of a Statistical Algorithm for Objective Comparison of Toolmarks," *Journal of Forensic Sciences*, vol. 60, no. 2, pp. 303–314, 2015.

Riemannian Manifold Approach to Scheimpflug Camera Calibration for Embedded Laser-Camera Application

Xiaoying Tan[1], Volkmar Wieser[1], Stefan Lustig[2] and Bernhard A. Moser[1]

***Abstract*— This industrial spotlight paper outlines a Riemannian geometry inspired approach to measure geometric quantities in the plane of focus of a Scheimpflug camera in the presence of nonlinear distortions caused by the Scheimpflug model and non-linear lens distortion.**

I. Introduction

For the standard pinhole camera model, the image sensor is parallel to the lens plane and perpendicular to the optical axis. For this type of camera, the points on a plane surface parallel to the lens can be focused sharply on the sensor plane. However, for some specific application scenarios, the surface of interest is oblique to the lens plane. For example, to capture most parts of a tall building facade into the camera view, the camera needs to be tilted upwards with respect to the building facade. In this case, the standard camera is only able to project a narrow line region of the building facade on sharp focus.

It is interesting that the Gaussian focus equation remains valid under the condition that the sensor plane, the lens plane and the object plane intersect in a common line [5].

The Scheimpflug model is encountered in various fields of applications, e.g., architectural photography [6] or in ophthalmology for measuring the thickness of the cornea [3].

In this industrial spotlight paper we address the problem of accurately measuring geometric quantities in the Scheimpflug plane in the presence of non-linear lens distortion effects by following a Riemannian geometry approach [1]. In contrast to state-of-the-art approaches the outlined approach is feasible on embedded platforms and gets along without guessing initial values and iterative optimization steps. Rather, it models the image formation mapping from the Scheimpflug plane to the image plane directly by exploiting point-to-point correspondences and interpolation.

In section II we recall the Scheimpflug model and calibration approaches from literature. Section III-A outlines our parameter-free approach together with experimental results.

II. Scheimpflug Camera

In contrast to the standard pinhole camera, in the Scheimpflug camera model the sensor plane and the lens plane are no longer parallel. See Fig. 1 of a schematic view of the Scheimpflug model. The mathematical model of its image formation mapping can be derived from decomposing the mapping from world coordinates (X,Y,Z) to image pixel coordinates $(\widetilde{x}_t,\widetilde{z}_t)$ into a concatenation of mappings as

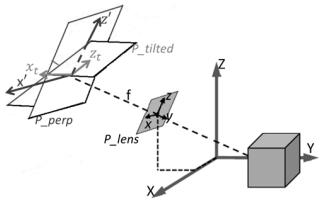


Fig. 1. Scheimpflug camera model: the sensor plane P_{tilt} and lens plane P_{lens} are no longer parallel. The image formation mapping is modeled by means of the virtual parallel plane P_{perp}.

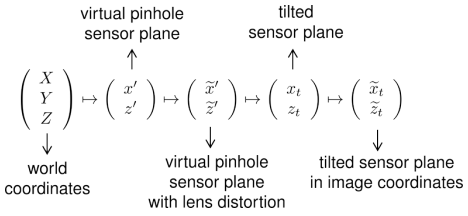


Fig. 2. The process of image formation of the Scheimpflug model according to (1), (2) and (3)

indicated in Fig. 2. First of all, the mapping from (X,Y,Z) to a virtual parallel sensor plane (x',z') models the familiar pinhole camera. By taking non-linear radial and tangential lens distortion effects into account, due to suboptimal shape and mounting of lens, and modeling these effects by means of polynomial functions we obtain

$$\begin{aligned}\begin{pmatrix}\widetilde{x}'\\ \widetilde{z}'\end{pmatrix} := \begin{pmatrix}x'\\ z'\end{pmatrix} + \begin{pmatrix}\Delta x\,(k_1 r^2 + k_2 r^4 + k_3 r^6)\\ \Delta z\,(k_1 r^2 + k_2 r^4 + k_3 r^6)\end{pmatrix}\\ + \begin{pmatrix}2t_1 x' z' + t_2\,(r^2 + 2x'^2)\\ 2t_2 x' z' + t_1\,(r^2 + 2z'^2)\end{pmatrix},\end{aligned} \tag{1}$$

where $r^2 := \Delta x^2 + \Delta z^2$, $\Delta x := x' - x_0$, $\Delta z := z' - z_0$, (x_0, z_0) are the coordinates of the optical axis on P_{perp}, k_1, k_2, k_3 are radial and t_1, t_2 are tangential distortion parameters. The mapping from $(\widetilde{x}',\widetilde{z}')$ to (x_t, z_t) models the proper Scheimpflug effect by taking the tilt of the sensor plane into account. Let us denote by α the angle between $\widetilde{z}'$ and z_t and by β the angle between $\widetilde{x}'$ and x_t, then due to [4] we obtain

$$\begin{pmatrix}x_t\\ z_t\end{pmatrix} := \lambda \cdot \begin{pmatrix}\widetilde{x}'/\cos\beta + \widetilde{z}'\tan\alpha\tan\beta\\ \widetilde{z}'/\cos\alpha\end{pmatrix} \tag{2}$$

where $\lambda := f/(f - \widetilde{x}'\tan\beta - \widetilde{z}'\frac{\tan\alpha}{\cos\beta})$ and f is the focal length. Finally, we obtain the image pixel coordinates

$$\begin{pmatrix}\widetilde{x}_t\\ \widetilde{z}_t\end{pmatrix} := \begin{pmatrix}S_x & -S_z\cot\theta\\ 0 & S_z/\sin\theta\end{pmatrix} \cdot \begin{pmatrix}x_t\\ z_t\end{pmatrix} + \begin{pmatrix}v_0\\ w_0\end{pmatrix} \tag{3}$$

[1] X. Tan, V. Wieser and B. Moser are with the Software Competence Center Hagenberg (SCCH), xiaoying.tan@scch.at

[2] S. Lustig is associated with SCCH stefan.lustig@scch.at

where (w,h) denotes the image size in number of pixels, (S_w,S_h) the sensor size in millimeter, (v_0,w_0) the coordinates of the principle point, θ the shearing angle in the sensor coordinate system and $S_x := w/S_w$, $S_z := h/S_h$. To this end we obtain a mapping

$$\Theta : (X,Y,Z) \mapsto (\widetilde{x}_t,\widetilde{z}_t) \tag{4}$$

which depends in total on 17 parameters (6 extrinsic, 2 Scheimpflug angles, 4 intrinsic, 5 distortions coefficients).

III. Scheimpflug Camera Calibration

A standard way for camera calibration in computer vision is the approach of minimizing a functional that measures to which extent the model (4) fits a given set of point-to-point correspondences resulting from a marker positions of a calibration plate. A familiar choice for the functional is the sum of squared projection errors. In particular, the estimate of the extrinsic parameters is not that easily performed. Therefore, usually simplified approximations are used as initial guess. For example, [2] starts from a distortion-free model and derives a first guess of the pinhole camera parameters as an approximation. It is then used as an initialization of a nonlinear bundle adjustment optimization that accounts for distortion and the 2-tilt Scheimpflug angels. In a similar way [4] starts with Zhang's method [7] for estimating the Scheimpflug angels α, β. In a further step, α and β are kept fix and the remaining parameters are estimated, again by using Zhang's method. This procedure is iterated until convergence.

A. Approach for Embedded Laser-Camera Application

The application scenario is about real-time affine reconstruction of geometric quantities by means of an embedded laser-camera system based on a DSP (TMS320DM6435, 700 MHz, 5600MIPS) and a hard-real time requirement of processing a measurement below 10ms. On such a platform the computational effort of trigonometric functions is about 20–40 times higher than standard vector operations. In our approach we exploit the fact that the laser projection plane and the plane of focus of the Scheimpflug camera are congruent. This setting allows a simplification of the general calibration procedure and gets along without the use of computational expensive functions.

Since the mapping (4) reduces to $\widetilde{\Theta} : (X,Z) \mapsto (\widetilde{x}_t,\widetilde{z}_t)$. Instead of solving the inverse problem of identifying the 17 parameters of the Scheimpflug camera model and tackling the problem from a global perspective, we consider the resulting geometric deformation as representation of a Riemannian manifold and exploit its local notions of angle and length of curves for accomplishing measurement tasks. In this view the measurement problem is solved by the following steps: (a) register point-to-point correspondences by means of a sufficiently dense grid of point markers on the plane of focus resulting from straight lines (geodesics in Euclidean geometry) and extraction of the point locations in the image by image processing; (b) determine the neighboring deformed grid points to the sample point; (c) apply 3-spline interpolation for approximate recovery of the corresponding geodesics in the resulting Riemannian manifold; (d) determine the Riemannian coordinates in the local coordinate system given by the geodesics; (e) compute the local inverse in order to obtain the Euclidean coordinates. In contrast to computing the full camera model which involves trigonometric functions and fractions, the outlined approach is also feasible on an embedded system as only polynomials of maximal degree 3 have to be evaluated. Fig. 3 shows an example of a deformed regular grid of calibration points by a Scheimpflug camera and the result of angle measurement based on this approach. The result shows that the systematic angle reconstruction error resulting from nonlinear Scheimpflug and lens distortion effects can be reduced substantially which meets the industrial requirements of the specific application.

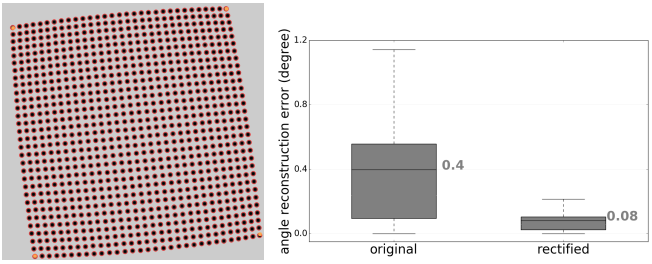


Fig. 3. Left: deformed regular grid of points by Scheimpflug camera and radial and tangential lens distortion: $\alpha = \beta = 5°$, $k_1 = -4.5e^{-3}mm^{-2}$; right: angle reconstruction errors with 249 pairs orthogonal calibration lines and 286 pairs test lines with different inclined angles (left box: original lines with the same distortion as the grid, mean = 0.397°, std = 0.431°; right box: distortion rectified lines, mean = 0.082°, std = 0.084°.)

ACKNOWLEDGMENT

This work has been partly funded by the Austrian COMET Program.

References

[1] I. Chavel, *Riemannian geometry, a modern introduction*. Cambridge University Press, 1993.

[2] P. Fasogbon, L. Duvieubourg, P.-A. Lacaze, and L. Macaire, "Intrinsic camera calibration equipped with Scheimpflug optical device," in *Proc. SPIE, 12th Int. Conference on Quality Control by Artificial Vision 2015*, vol. 9534, 2015, pp. 953 416–953 416–7.

[3] M. Ibáñez-Ruiz, P. Beneyto-Martin, and M. Pérez-Martínez, "Lens density measurement with scheimpflug camera in vitrectomised eyes," *Archivos de la Sociedad Española de Oftalmología (English Edition)*, vol. 91, no. 8, pp. 385–390, 2016.

[4] A. Legarda, A. Izaguirre, N. Arana, and A. Iturrospe, "A new method for Scheimpflug camera calibration," in *10th Int. Workshop on Electronics, Control, Measurement and Signals*, June 2011, pp. 1–5.

[5] T. Scheimpflug, "Improved method and apparatus for the systematic alteration or distortion of plane pictures and images by means of lenses and mirrors for photography and for other purposes," May 1904, GB 1196/1904.

[6] R. Sidney, "The manual of photography: Photographic and digital imaging," R. E. Jacobson, S. F. Ray, G. G. Atteridge, and N. R. Axford, Eds. Great Britain: Oxford: Focal Press, 2000, ch. The geometry of image formation, pp. 39–60.

[7] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

On Quality Assurance of 3D Bust Reconstructions

Gernot Stuebl, Christoph Heindl, Harald Bauer, and Andreas Pichler[1]

***Abstract*— In this paper a non-reference method for quality assurance in 3D bust reconstruction is presented. The proposed approach is part of an automatic parametrization concept for 3D reconstruction applications with no ground-truth data available. It is based on a novel concept of pair-wise view comparisons, which is new in this field. Evaluation on a dataset of human bust scans shows perfect prediction of human votes.**

I. INTRODUCTION

Exact reconstruction of the human body especially the bust is an application field which got boosted by the raise of low-cost 3D printers and online 3D printing services. Nevertheless creating a high fidelity 3D reconstruction often involves manual post processing.

Recent publications present systems which are able to do reconstructions on a quality level which makes post processing unnecessary, see Heindl et al. [1]. However for these the quality strongly relies on a correct parametrization of the system. Unfortunately parametrization is dependent on the scan data. So no golden standard for a parameter setting exists and the parameter values have to be adopted for each reconstruction individually. In principle human interaction has been shifted from direct manipulation/correction of 3D data to the selection of correct parameter values. Having this in mind, an (semi-)automatic configuration of the parameter values is desirable.

The paper is outlined as followed: first Section II gives an overview of traditional quality assurance methods for 2D and follows with related work in the field of 3D quality assurance. The main approach is described in Section III, whereas Section IV presents the results on a dataset of 3D bust reconstructions. This is followed by a discussion on the applicability of the approach in Section V as well as a conclusion and outlook to future research in the last section.

II. RELATED WORK

A vital part of an automatic parametrization system is a component for assessing the reconstruction quality. The following subsections covers related work in this domain with an introduction of traditional 2D measures and the main emphasis on 3D quality assurance.

A. 2D Quality Assurance

In 2D there are traditional (dis-)similarity measures which are used for quality assurance. Some of these can also be adopted to 3D. A simple one is the Root-Mean-Squared Error (RMSE) [5] of two images I, K which is defined as

$$\mathrm{RMSE}(I,K) := \sqrt{\frac{1}{mn}\sum_{p=0}^{m-1}\sum_{q=0}^{n-1}(I(p,q)-K(p,q))^2} \quad (1)$$

and measures the deviation in each pixel. Based on this the Peak Signal to Noise Ratio (PSNR) [5] is defined as

$$\mathrm{PSNR}(I,K) := 20\cdot\log\frac{I_{\max}}{\mathrm{RMSE}(I,K)} \quad (2)$$

with $I_{\max}$ the maximum possible value in the image (e.g. 255 for monochromatic 8 bit images). PSNR measures the signal fidelity between an original and a disturbed image. A more complex measure is Structural Similarity index (SSIM) [2] which is designed to judge signal fidelity in the way the human vision system does. It is sensitive to structural distortions such as noise contamination, blurring, and insensitive to non-structural distortions such as luminance and contrast change. The mathematical definition is

$$\mathrm{SSIM}(\vec{x},\vec{y}) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (3)$$

with $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ as stabilization constants for the division with weak denominators, where $L = 2^b - 1$ denotes the dynamic range of pixel-values with b as the number of bits per pixel and $k_1 = 0.01$ and $k_2 = 0.03$.

B. 3D Quality Assurance

Generally, quality assurance algorithms are divided into full-reference (FR), reduced-reference (RR) and no-reference (NR) algorithms. This distinction is based on the amount of information that is available.

Full-reference algorithms rely on a ground-truth data, e.g. early attempts to judge quality through texture and geometric resolutions belong to this category, see Pan et al.[3]. Also a broad range of algorithms which measure the quality of 3D codecs or stereoscopic 3D are full-reference based, see Mekuria et al. [4]. You et al. [5] give a good overview on how traditional 2D measures can be used for FR 3D quality assurance.

For reduced-reference algorithms the ground-truth is not fully available. Instead of this, selected features are calculated from the ground-truth and used as input of the quality assurance system, see Wang et al. [6] or Rehman and Wang [7].

A recent example for a no-reference algorithm is presented by Alexiadis et al. [8]. In this work the 2D key frames which are needed to build the 3D reconstruction are compared to

[1]PROFACTOR GmbH, 4407 Steyr-Gleink, Im Stadtgut A2, Austria {Forename.Surname}@profactor.at

synthesized versions of it. The authors utilize a SSIM based measure to adjust the reconstruction settings. This is close to the proposed approach in this paper. The main difference is that we do not need to process the available key frames but instead work only on synthetic views.

III. AUTOMATIC PARAMETRIZATION

The aim of automatic parametrization is to determine optimal values for different reconstruction parameter-types. In this case optimality means that the parameter value is near or equal the value a human operator would have chosen for the given data. Figure 1 depicts examples for the influences of different parameter-types.

(a) *ColourCorrection* value 0. (b) *ColourCorrection* value 7. (c) Difference image.

(d) *PoseEstimation* value 0. (e) *PoseEstimation* value 7. (f) Difference image.

(g) *SmoothnessTerm* value 0. (h) *SmoothnessTerm* value 4. (i) Difference image.

(j) *MaxIterations* value 0. (k) *MaxIterations* value 7. (l) Difference image.

Fig. 1. Reconstruction effects of different parameter-types shown on Model 0001 in Subfigures (a) to (c), Model 0019 in Subfigures (d) to (f), Model 0010 in Subfigures (g) to (i), and Model 0033 in Subfigures (j) to (l). The images in the last column highlight the differences. The parameter values are set to the extremes, to better demonstrate the effects.

The following procedure illustrates how a non-professional human operator could select a good parameter setting:

1) The operator sets or alters a parameter value.
2) The operator lets the reconstruction run.
3) The operator inspects the result from different views if it is better or worse than before.
4) The operator repeats the steps until some level of reconstruction quality is reached.

Based on this we propose an approach using pair-wise view comparison of different reconstructions. For a parameter-type α the accumulator matrix M_α is a symmetric matrix defined as

$$M_\alpha(k,l) := \sum_{i:=0}^{n} \mathrm{SIM}(V_i(R_{\alpha,k}), V_i(R_{\alpha,l})) \tag{4}$$

where k,l are elements of the ordered parameter value set P_α and $R_{\alpha,k}$ is the reconstruction. Elements in P_α are chosen such there is an increasing influence of the parameter to the observed visual effect. n is the number of equally spaced views around $R_{\alpha,k}$, whereas each view V_i is a 2D projection of the 3D object. For comparison of two images as similarity SIM the Peak Signal to Noise Ratio (PSNR) is used.

We assume that for humans the skin area is very important for quality judgement. Therefore the images are converted into the Hue, Saturation, Value (HSV) colour space before comparison. This should make the comparison more sensitive to skin parts, see Sedlacek [9]. A detailed evaluation and discussion of this step follows in Subsection IV-C and Section V.

Pixel-wise comparison is performed only on the bust itself, since the background is masked out during comparison. Given this framework we propose the optimal parameter value $o_\alpha \in P_\alpha$ to be defined as

$$o_\alpha := \arg\max_{k \in P_\alpha} \sum_{l \in P_\alpha} M_\alpha(k,l). \tag{5}$$

Literally speaking the parameter value o_α creates 2D views which are most similar to the views created with all other values. The hypothesis is that this is also a good parameter value which a human would choose.

IV. EVALUATION

Due to the lack of free datasets for bust reconstruction, an own dataset has been built up during an open house presentation in the company.

A. Dataset

The dataset contains 32 3D human bust scans showing different people, further called models. The data is acquired with a turntable and an off-the-shelf RGB-D sensor. Each individual is scanned in eight key poses. For the detailed set-up of the scan process see Heindl et al. [1].

For the reconstruction four different parameter-types are inspected: colourcorrection level, number of steps for pose estimation, surface smoothness term and maximal iteration of the bundle adjustment. These types form the parameter set $S = \{$ *ColourCorrection, PoseEstimation, SmoothnessTerm, MaxIterations* $\}$. For a detailed explanation of the reconstruction software and the parameter semantics see again Heindl et al. [1].

Eight models are assigned to each parameter-type $\alpha \in S$. A model is reconstructed with the full range of parameter values, which are 8 values for *ColourCorrection, PoseEstimation, MaxIterations*, and 5 values for *SmoothnessTerm*. The parameter space is discrete and the values form the individual parameter value sets P_α.

In a questionnaire 32 people (16 male, 16 female) between 19 and 55 years old, were asked to choose the most aesthetic reconstruction for each model. Since every reconstruction is mapped to a parameter value, they implicitly chose the parameter value which led to the best reconstruction quality.

The best parameter choices according to the human votes have been counter-checked to produce reasonable reconstructions. During this result preparation, one model which was assigned to *SmoothnessTerm*, had to be omitted because of inconsistencies in the data. In detail the parameter value with the most human votes for this model leads to a failed reconstruction similar to the bottom right picture of Figure 4. Therefore the final dataset consists of 31 human judged model reconstructions.[1]

B. Evaluation Criteria

Figure 2(a) and Figure 2(b) show example distributions of human decisions for specific models. One can see the variances in the votes. To cover these variances we define the following correctness criterion:

Definition 1. *A parameter value estimation is correct if it is inside $\mu \pm \sigma$ of the human decisions.*

To test this criterion, human judgements have been simulated with random values. In detail for each decision distribution (e.g. Subfigure 2(b)) a uniformly distributed random value in the same discrete parameter range was generated. If the random value fulfilled the correctness criterion for the decision distribution, it was counted as correct, otherwise as incorrect. With 1000 trials this lead to a mean accuracy of 0.5095 and $\sigma = 0.0841$ which can be seen as baseline for the following tests.

C. Results

In an evaluation which is run on each decision distribution in the dataset, the best parameter value for the reconstruction of a model is estimated using Equation 5 with PSNR as similarity measure. After that the parameter value is checked against the decision distribution with Definition 1. Therefore if the parameter value is inside $\mu \pm \sigma$ of the human decisions, the parameter value estimation is counted as correct and false otherwise. This procedure lead to an estimation accuracy of 1 on the dataset of 31 judged reconstructions.

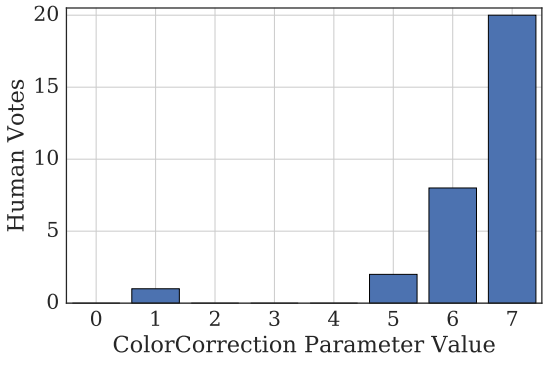
The evaluation has also been run with two other (dis-) similarity measures: Root-Mean-Squared Error (RMSE) and Structural Similarity index (SSIM), see Table I.

The first is a standard measure for deviations. Applying it the accuracy drops to 0.9032. This is further interesting since the RMSE is also the denominator in Equation 2. One can see that the logarithm in the equation is important in this context.

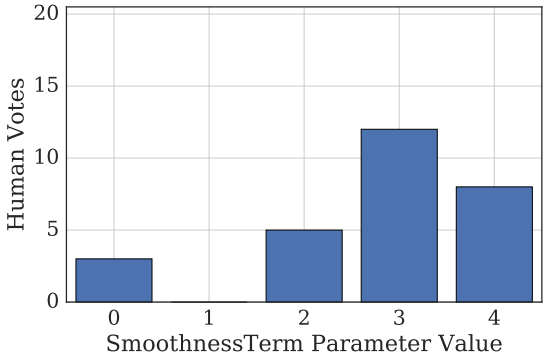
When applying SSIM, which should reflect human perception, the accuracy drops to 0.9032. A detailed look on the results reveals that RMSE as well as SSIM fail on the models assigned to *ColourCorrection*.

A similar comparison has been performed with different colour spaces, see Table II. Beside the HSV colour space Red, Green, Blue (RGB), YCbCr, Grayscale and CIE-Lab colour spaces have been evaluated. RGB is a standard in image representation. When using it the accuracy drops to 0.7742. Recent publications indicate that YCbCr colour space shows advantages in skin detection, see Shaik et

[1]The full dataset can be requested by emailing the main author.



(a) Parametertype *ColourCorrection* on Model 0001.



(b) Parametertype *SmoothnessTerm* on Model 0093.

Fig. 2. Examples of human decision distributions for parameter-types *ColourCorrection* on Model 0001 in Subfigure (a) and *SmoothnessTerm* on Model 0093 in Subfigure (b). One can see the variance in the data.

(Dis-)similarity	Accuracy
PSNR	1
RMSE	0.9032
SSIM	0.9032

TABLE I

ACCURACY ON THE DATASET EVALUATED WITH DIFFERENT IMAGE (DIS-)SIMILARITIES FOR FORMULA 4. EVALUATED MEASURES ARE PEAK SIGNAL TO NOISE RATIO (PSNR), STRUCTURAL SIMILARITY INDEX (SSIM) AND ROOT-MEAN-SQUARED ERROR (RMSE). PSNR PERFORMS BEST.

Colour space	Accuracy
HSV	1
RGB	0.7742
YCbCr	0.7419
Grayscale	0.7419
CIE-Lab	0.7188

TABLE II

ACCURACY ON THE DATASET EVALUATED USING DIFFERENT IMAGE COLOUR SPACES. PEAK SIGNAL TO NOISE RATIO (PSNR) IS USED AS SIMILARITY MEASURE. EVALUATED COLOUR SPACES ARE HUE, SATURATION, VALUE (HSV), RED, GREEN, BLUE (RGB), YCBCR, GRAYSCALE AND CIE-LAB. USING HSV SHOWS THE HIGHEST ACCURACY.

al. [10]. Nevertheless by using this colourspace the accuracy drops to 0.7419. On the other hand with Grayscale colourspace the accuracy drops also to 0.7419. This is of further interest since the applied grayscale conversion algorithm simply takes the Y component of YCbCr and omits the colour channels. This procedure is common usage in photo editing software like Photoshop [2] or GIMP [3]. A further look on the results uncovers that YCbCr and Grayscale have their wrong estimations on the same models. Therefore CbCr colour encoding adds no benefit to using the Y channel alone in this application. CIE-Lab colour space was also evaluated since it approximates human vision, unfortunately in this application the accuracy dropped to 0.7188.

D. Comparison with state-of-the-art

A comparison with state-of-the-art is difficult, since the algorithms are usually embedded into a certain application scenario which is not always exchangeable.

Nevertheless the *Evaluation of the appearance quality* part in the publication of Alexiadis et al. [8] has been adopted to our set-up: The parameter value of which the reconstructed views are most similar to the ground-truth key-frames is chosen as best value. Like in Alexiadis et al. the similarity measure is SSIM and the colour space HSV.

When run on the dataset the accuracy is at 0.4062. This is not a fair comparison since the appearance evaluation is only a part of the whole framework of Alexiadis et al. and only confirms that the set-ups of both approaches cannot be intermixed.

V. DISCUSSION

This section contains a discussion about the applicability of the approach as well as considerations on the runtime.

A. Applicability

The proposed approach relies on an interesting property of the reconstruction principle: changes in the parameter value lead to mainly distinct local deviations in the model.

PSNR as the chosen similarity measure has to be sensitive to this deviations. To visualize this a metric Multi Dimensional Scaling (MDS) [11] algorithm is utilized. An MDS algorithm tries to position each object in multi-dimensional space such that the between-object distances are kept as well as possible. This gives more insight into the working principle of the proposed approach since it illustrates which images are similar from the view of PSNR.

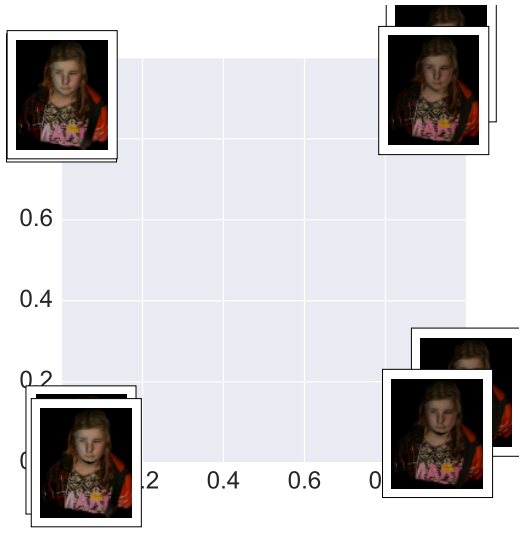


Fig. 3. Multidimensional scaling layout of all frontal views for parameter-type *ColourCorrection* on Model 0093 using Peak Signal to Noise Ratio (PSNR) as similarity measure and Hue, Saturation, Value colour space. Top right is the best choice, bottom left and right show deviations on the pine, top left on the right cheek. The farther the images are away from each other the more they are different in the meaning of PSNR. The images form clusters according to local deviations in the reconstruction.

In Figure 3 all frontal view reconstructions in the whole parameter value range for *ColourCorrection* of a specific model (0093 in the dataset) are laid out with an MDS algorithm. To create the necessary distance matrix for the algorithm, the similarities in M_α were converted to distances. On the top right is the optimal reconstruction. Bottom left and bottom right show deviations on the pine, whereas top left deviates on the left cheek. It can be seen that images with similar deviations are clustered together.

However the increasing visual effect of the parameter values, mentioned in Section III, is not visible in the layout, on the one side because MDS is a form of non-linear dimensionality reduction and on the other side PSNR as underlying measure does not fully reflect the human visual perception.

Figure 4 depicts also a MDS layout for a whole parameter value range (parameter-type SmoothnessTerm on Model 0098 in the dataset). On bottom right is the rare case of a complete failed reconstruction, which has a high distance to the other images. One can see that the case of a global deviation is treated well, as long as it is not in the majority of the images.

The dependency on distinct local deviations can be a loss of generality of the approach. However especially in the area of human 3D reconstruction there should be a wide range of possible applications. Furthermore our approach is not dependent on a certain reconstruction principle.

[2]http://www.adobe.com/at/products/photoshop.html
[3]https://www.gimp.org

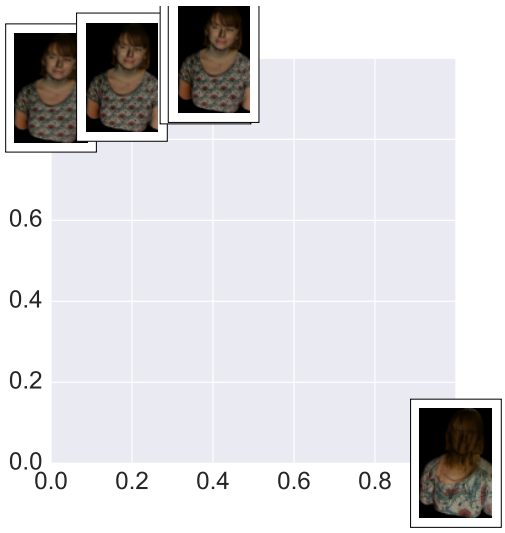


Fig. 4. Multidimensional scaling layout of all frontal views for parameter-type *SmoothnessTerm* on Model 0098 using Peak Signal to Noise Ratio (PSNR) as similarity measure and Hue, Saturation, Value colour space. The farther the images are away from each other the more they are different in the meaning of PSNR. Therefore the failed reconstruction on bottom right has a high distance to the other images.

A further eventual loss of generality is the coupling to a specific colour space (HSV) together with the assumption that human decisions are dependent on skin deviations. All models in the dataset are Central Europeans with white skincolour. It is not sure that the proposed approach in this configuration works also with models having other skincolours. Nevertheless the approach is a good starting point for future work, see Section VI.

A final point regarding applicability is that the proposed approach inspects all parameter-types isolated, see Section VI on future work to this issue.

B. Runtime Considerations

The proposed method utilizes a brute force evaluation of all parameter values. While the final comparison of the views is computationally cheap, the reconstruction itself is time consuming: On an Intel Core i5-200 CPU with a NVIDIA Geforece GTX 560 and 16GB RAM it takes in the mean 145 s to do a reconstruction. To overcome this issue, the reconstruction has been implemented as web service in the Amazon Cloud.

Since the reconstructions are independent of each other, they could be run fully in parallel, benefiting from the virtually infinite computational power in the cloud. However in practice we run the parallelization in a way such that one parameter-type can be fully evaluated at once.

VI. Conclusion and Future Work

In this work an approach utilizing pairwise comparison of 2D views from different 3D model reconstructions has been demonstrated, which simulates human quality choices. The approach shows perfect prediction on the given dataset.

The essential part of the approach is to select the reconstruction which is most similar to all others. The effect is that reconstructions with local deviations are sorted out. This idea is new and might inspire other scientific work.

From the technical side there are two main possibilities of improvement, which are caused by the nature of the used dataset. First the dataset only covers white-skinned Central Europeans and the approach is coupled to a specific colour space. So there could be a loss in generality when inspecting models with other skincolours. To overcome this a future work could use a face detector as pre-step and parametrize the comparison to the actual skincolour. For this new models have to be added to the dataset.

Another future work may approach the issue of isolated parameter-type evaluation. Unfortunately with the available questionnaire, combinations of parameters cannot be evaluated since they are not in the data. However for future work this would be very interesting, since it could provide further insights to the generality of the approach. In case that there will be significant dependencies between parameter-types a future version may include some kind of genetic algorithm to find the best combination.

Acknowledgment

This research is carried out within the ”FTI-Project ProTechLab” project funded by the State of Upper Austria through the Strategic Economic and Research Program ”Innovatives OÖ 2020”.

References

[1] C. Heindl, S.C. Akkaladevi, and H. Bauer. *Capturing Photorealistic and Printable 3D Models Using Low-Cost Hardware*, pages 507–518. Springer International Publishing, Cham, 2016.

[2] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, April 2004.

[3] Y. Pan, I. Cheng, and A. Basu. Quality metric for approximating subjective evaluation of 3D objects. *IEEE Transactions on Multimedia*, 7(2):269–279, April 2005.

[4] R. Mekuria, P. Cesar, I. Doumanis, and A. Frisiello. Objective and subjective quality assessment of geometry compression of reconstructed 3D humans in a 3D virtual room, 2015.

[5] J. You, G. Jiang, L. Xing, and A. Perkis. *Quality of Visual Experience for 3D Presentation - Stereoscopic Image*, pages 51–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[6] X. Wang, Q. Liu, R. Wang, and Z. Chen. Natural image statistics based 3D reduced reference image quality assessment in contourlet domain. *Neurocomputing*, 151, Part 2:683 – 691, 2015.

[7] A. Rehman and Z. Wang. Reduced-reference image quality assessment by structural similarity estimation. *IEEE Transactions on Image Processing*, 21(8):3378–3389, Aug 2012.

[8] D.S. Alexiadis, A. Chatzitofis, N. Zioulis, O. Zoidi, G. Louizis, D. Zarpalas, and P. Daras. An integrated platform for live 3D human reconstruction and motion capturing. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–1, 2016.

[9] M. Sedlacek. Evaluation of RGB and HSV models in human faces detection. Central European seminar on computer graphics, Budmerice. In *IIIA.1-5 - Conference on Computer Systems and Technologies - CompSysTech2004*, page 125131, 2004.

[10] K.B. Shaik, P. Ganesan, V. Kalist, B.S. Sathish, and J.M.M. Jenitha. Comparative study of skin color detection and segmentation in HSV and YCbCr color space. *Procedia Computer Science*, 57:41 – 48, 2015. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

[11] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics)*. Springer, 2nd edition, August 2005.

An Image Analysis System for Selective Recovery of Non-ferrous Metal

Malte Jaschik[1], Alfred Rinnhofer[2], Martina Uray[3] and Gerhard Jakob[4]

***Abstract*— To increase the recycling rate for non-ferrous metal, a high speed sorting line with a high throughput rate of up to 1ton per hour was built. The system comprises an Image Analysis System to detect shredder particles and calculate their position on the belt as well as several 2D and 3D shape features. ElectroMagnetic Tensor Spectroscopy (EMTS) or Laser-Induced Breakdown Spectroscopy (LIBS) characterize each particle based on its metal components. Tests were conducted under hard conditions in an industrial environment. For a full covered 400mm x 100mm belt area the Image Analysis System needs less than 24.5ms at a feature calculation accuracy up to 95%. The developed system can easily be adapted to other scenarios.**

I. INTRODUCTION

The requirements of the sorting line described in this work are to detect and classify non-ferrous metal particles. The load speed of the sorting line is given by 1ton/hour. A vibrating feeding system is used to load the belt (width of 400mm) and for fragment separation. Due to the limitation of the vibrating feeding system, about 28g/s of particles can be loaded with a belt speed of 2m/s. The subsystems need an accuracy of 0.5mm/px in length and width. Therefore, a line rate of 4kHz is required. Fig. 1 shows a schematic of the components and Fig. 2 a sample image of such a particle. The system is developed to work even under heavy industrial conditions (like dust, vibrations of machines, etc.).

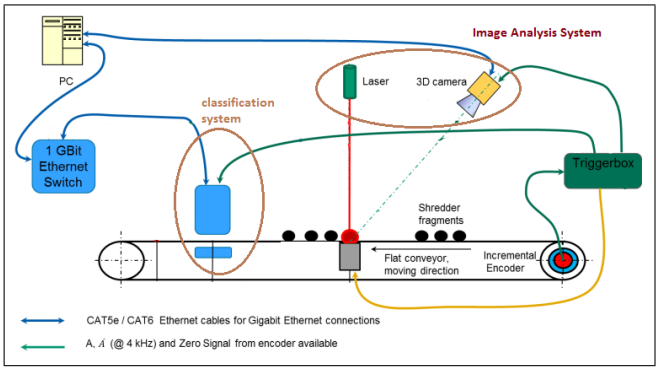


Fig. 1. Schematic of the developed sorting line.

The developed sorting line consists of two independent subsystems (detector and classifier). An Image Analysis System identifies every single particle on the conveyor and calculates its exact position on the belt as well as 2D and 3D shape features. The analysis and classification of the material of the particles is provided by EMTS or LIBS. The EMTS measures the electrical conductivity while the LIBS characterizes the chemical composition. To achieve optimum results of the classification systems, it is essential that the position and specified shape features of every single particle is derived to utmost precision by the Image Analysis System. Therefore, all subsystems are synchronised by an incremental encoder.

Joanneum Research Forschungsgesellschaft mbH
DIGITAL - Institute for Information and Communication Technologies
[1]malte.jaschik@joanneum.at
[2]alfred.rinnhofer@joanneum.at
[3]martina.uray@joanneum.at
[4]gerhard.jakob@joanneum.at

II. DISCRIPTION OF SUB SYSTEMS

A. *Image Analysis System for position and shape calculation*

The Image Analysis System is based on laser triangulation and comprises a line laser, an Automation Technology C4-1280-GigE 3D camera and a computer for calculation. Using subpixel algorithms a height resolution, defined by the optical resolution and the angle between laser plane and camera, of 0.15mm can be achieved. Due to the belt movement the laser line migrates along the surface of the fragments. The camera acquires a 2D image of each laser line and calculates a 3D profile that is sent to the computer via GigE. The analysis algorithm stores the 3D profile and builds an image, called subframe (currently consisting of 200 3D profiles), a small part is illustrated in Fig. 3. The camera can acquire a grayscale image (Fig. 2) as well, but it is not used in the current application.

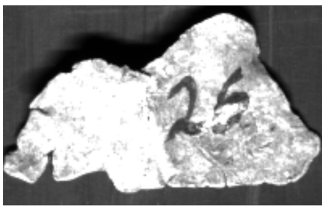


Fig. 2. Grey-value image of one metal particle

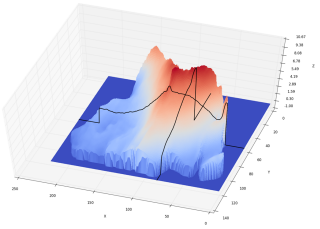


Fig. 3. 3D image of one metal particle (same as Fig. 2)

To remove noise due to non-flatness of the conveyor a background model is calculated and subtracted from the image (for each 3D profile). By binarizing the subframe with an experimentally determined height threshold areas of interest are selected. If the size of an area is big enough, it defines a particle hypothesis and features are calculated. Features reach from simple positions to more complex ones, like feret diameters or maximum cross-sectional area (overall 25 different features in 2D and 3D respectively).

B. Classification systems

The current application employs two different classification subsystems. Both systems provide a satisfactory solution for the non-ferrous metal classification. Currently linear classifiers with several features are used, but in the future neuronal networks will be trained.
The LIBS system measures the chemical composition of the particles and separates them into cast and wrought aluminium categories and into selected aluminium and magnesium alloys.[1]
The EMTS system on the other hand measures the electrical conductivity of the particles to separate the fragments into aluminium, copper and brass categories.[2]

III. RESULTS

A. Timing Performance

Due to the described settings, the maximum computation time is 50ms for one subframe. Two types of evaluations were realized with separated aluminium particles on a 1000mm x 400mm belt area. The first test (small covering, SC) simulated the real coverage with 139g particles. In the second test the belt was fully covered (full covering, FC). Three test sets of particles were used. The particles were placed on the belt and processed three times in the same arrangement. The arrangement itself was varied three times, such that nine timing tests for every particle size were done. Table I shows the test conditions and the calculation time for one subframe. As can be seen, the maximum computation time is 24.5ms for a full covered belt with 55 large samples.

TABLE I
MAXIMUM COMPUTATION TIME FOR FEATURE CALCULATION ON ONE SUBFRAME (200 3D PROFILES)

Test set	Small		Medium		Large	
Sample size [mm]	9x9		20x20		30x30	
Test size	SC	FC	SC	FC	SC	FC
Sample count	71	355	29	145	11	55
Max. time [ms]	20.91	23.33	21.26	24.1	11.78	**24.5**

B. Accuracy

To verify the accuracy of the Image Analysis System objects with defined dimensions were used (e.g. eurocents and washers) as well as real particles. Due to the complex real particle shapes no ground truth for heights, areas and diameters were available. Therefore, just the positions and recognition rates of real particles were tested.

The feature calculation accuracy is higher than 95% and almost every single particle can be detected (see Table II). Nearly all coins were detected correctly. Only one misdetection was observed since two 2 eurocents were not separated on the belt. The small deviation of the area can be explained by the fact that reflections on the edge lead to overestimate the real object size. Thus, the height is measured also on edge regions with height values produced by reflections. The height is furthermore influenced by the shape of the coins. Only the edge has full height, whereas the rest of the surface

TABLE II
ACCURACY OF THE IMAGE ANALYSIS SYSTEM IN %

Sample	Height	Area	Diameter	Found
1 eurocent	96.74	98.13	99.89	100
2 eurocent	98.91	96.68	99.87	99
5 eurocent	99.14	99.29	99.29	100
10 eurocent	98.25	99.10	99.10	100
20 eurocent	99.67	98.33	99.27	100
50 eurocent	98.97	98.97	99.58	100
Washer 16	98.94	95.18	97.12	100
Washer 20	95.38	98.50	98.84	100
Washer 22.5	97.38	98.49	95.66	100
Shredder	-	-	-	100

is below this level caused by different motives. As can be seen in Fig. 4 the height and area could be used to derived a simple image based classifier for coins.

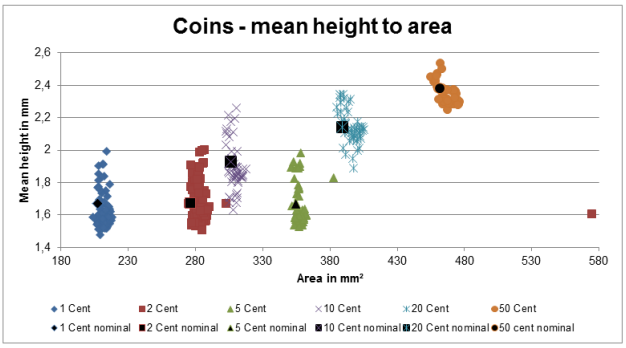


Fig. 4. Simple coin classification and comparison of the calculated values for the mean height and area with the nominal values.

All washers were detected correctly. The accuracy of the washer analysis is similar to the coins, only the area is a little less accurate, due to the hole in the middle of the washers.
The real particles were all detected by the system and the position on the belt were calculated correctly.

IV. CONCLUSIONS

In this work we have shown that the Image Analysis System is capable of detecting particles and calculating all required features at very high accuracy over 95%. This can be done in less than 24.5ms at a full covered belt with subframes of 400mm width and 100mm length. The system exceeds all requirements and has enough processing capabilities for several extensions. Its simplicity and independency of other systems enable its usage for other applications as well.

ACKNOWLEDGMENT

This research was funded by the European Commission under FP7, project *ShredderSort*, Grant Agreement Nr. 603676.

REFERENCES

[1] E. Grifoni, S. Legnaioli, G. Lorenzetti, S. Pagnotta, and V. Palleschi, "Applying libs to metals processing," *Spectroscopy*, pp. 20–31, 2015.
[2] Y. Tao, W. Yin, W. Zhang, Y. Zhao, C. Ktistis, and A. Peyton, "A very-low-frequency electromagnetic inductive sensor system for work-piece recognition using the magnetic polarizability tensor," *IEEE Sensors Journal*, 2017.

Automated Quality Assessment of Remelted Steel Ingots

Daniel Gruber[1], Harald Ganster[1] and Robert Tanzer[2]

***Abstract*— For high quality steel products it is essential to have specific understanding of the underlying steel production process such as the electric slag remelting process (ESR). To assist the currently manual assessment there is a high need for objective quality measures and standardized evaluation methods. A set of relevant parameters can be derived from the so-called pool profiles that give insight to the remelting process. Based on texture segmentation and ridge detection a computer-vision based automated evaluation of the pool profiles is achieved. A comparison with manually extracted pool profiles from expert metallurgists shows the feasibility of the approach and the good performance of the automated analysis. Further evaluation on different types of steel blocks will yield valuable insight to and improve the overall steel production process.**

I. Introduction and Motivation

The field of quality management and improvement in high quality steel production is one of the deciding reasons whether a steel producer remains competitive or not. In the production of high quality steel products for demanding applications it is essential to remelt conventional produced ingots. In order to yield specific understanding of the remelting process as well as to improve the process, there is a high need for an objective and standardized evaluation of remelted blocks.

The advantage through technology is to be able to substitute pure manual quality control and, thus, very time-consuming work flows. Furthermore, it is possible to provide repeatable calculations of quantitative measurements. This paper presents a vision-based solution to be able to automate those processes.

Currently, most of the structure evaluation is done manually and the information is stored in different analog and digital files. In order to be able to store all information in one place, a software was developed where various different kinds of meta data can be directly mapped to the analyzed steel block.

After a short introduction of the data material (Section II) and a brief overview of related work (Section III), Section IV gives insight into the quality assessment of steel ingots. Section V presents the automated segmentation, Section VI an objective method to derive steel quality parameters and Section VII gives some final conclusions and an outlook on future work.

[1]JOANNEUM RESEARCH Forschungsgesellschaft mbH, DIGITAL - Institute for Information and Communication Technologies, Austria daniel.gruber@joanneum.at, harald.ganster@joanneum.at

[2]BÖHLER Edelstahl GmbH & Co KG, Austria robert.tanzer@bohler-edelstahl.at

II. Data Material

Figure 1 shows the scheme of an ESR. Those remelted high quality steel blocks have a weight up to 20 tons. To analyze the inner solidification of such blocks, it is necessary to saw out longitudinal slices from the center of the block. Furthermore, these plates are cut into pieces to be able to handle size and weight as illustrated in Figure 2.

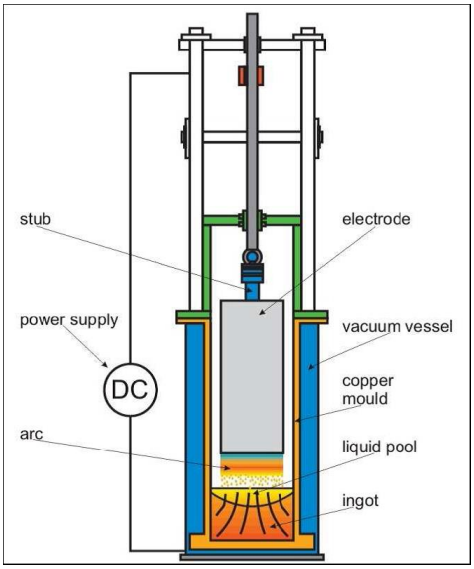


Fig. 1: Scheme of electric slag remelting process.

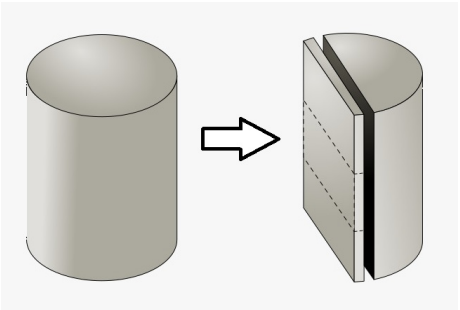


Fig. 2: Preparation of steel blocks for evaluation.

In order to gain deeper knowledge of the remelting process those plates are ground, polished and etched to reveal the inner crystalline solidification structure. Those structures provide information directly linked to the remelting parameters and as a consequence are essential for optimizing these parameters. Changes within the remelting process are directly related to the solidification structure [11], [8]. As a

last step, the prepared steel specimen are scanned by a 4k line scan camera. The problem with conventional assessment approaches is that the preparation of the data material is very costly and time consuming. Thus, the available data material for this work consisted of only three blocks with manually annotated ground truth.

III. Related Work

Vision-based approaches are already well established in assessment of material surface characteristics. As well there are also several approaches related to steel quality assessment.

A computer vision based microstructure analysis and classification approach is introduced in [3]. The strategy is to set up a complex histogram representing a 'fingerprint' of a microstructure. With the aid of those histograms it is possible to classify similar texture patterns by calculating the χ^2 distance.

Characterization of steel specimen surfaces are also presented in [2]. Signatures of surface profiles are extracted with multiresolution wavelet decomposition. Furthermore, surface roughness parameters are derived from those signatures.

Another feature extraction from micrographs is elaborated in [7]. The focus within this paper lies on extracting features like grain size, anisotropy of grains and the amount of δ phase.

Further research on vision-based steel surface inspection mainly focuses on the detection of defects. A summary of detectable surface defects and approaches to identify them can be found in [5].

Nevertheless, the proposed methods focus on the analysis of microscopic scale specimens (few mm^2) with their specific microscopic structures or the detection of defects. In contrast, the approach presented in this paper aims at the inspection and analysis of a full steel block with its macroscopic features. Those features exhibit completely different appearances than the microscopic structures.

IV. Quality Assessment of Steel Ingots

Significant parameters for the quality of steel can be derived from so-called pool profiles, which can be derived from inspection of the remelted steel blocks. With the aid of those pool profiles it is possible to determine certain quality attributes within the whole steel block. Therefore the equality of the individual pool profile lines with their surroundings are taken into account. Figure 3 shows manually derived pool profiles of an example steel block plate. These are generated by human experts (metallurgists) who try to identify the growth direction of the dendrites[1] in the image. Based on those direction vectors, lines in predefined distances are estimated perpendicular to the vectors. This process is very time consuming and prone to human error. Furthermore, the results are influenced by subjective interpretation and, thus, experts easily end up with diverse results.

[1]Dendrites are complex three-dimensional tree-like structures. Dendritic morphology is the most commonly observed solidification structure [9], p. 78.

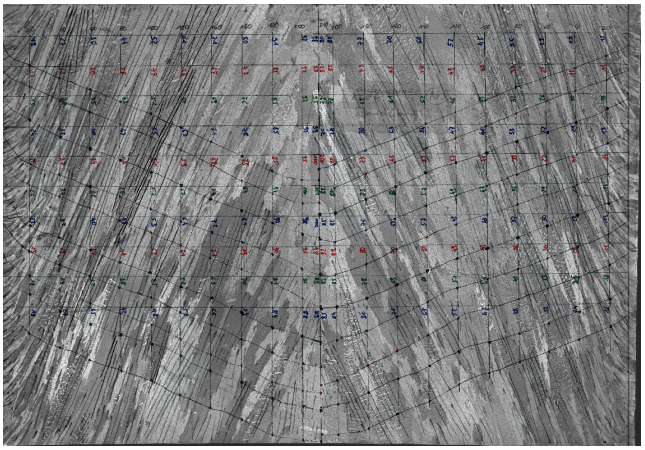


Fig. 3: Manually derived pool profiles.

Further ground truth data analysis revealed that some blocks show much more irregularities on top, bottom and in the middle due to the globular solidification in those areas. To be still able to extract meaningful pool profiles, metallurgists disregard those areas and simply classify pool profiles in regions with trans-crystalline solidification only. This basically means that trans-crystalline solidification areas provide representative information, whereas globular areas are basically unstructured and as a result do not provide meaningful information for the pool profiles. Thus, for an objective evaluation it is essential to automatically distinguish between globular and trans-crystalline solidification areas.

V. Steel Specimen Segmentation

The consequential first step of the automated quality assessment is the segmentation of globular and trans-crystalline solidification areas. The main idea for automated segmentation is based on the different textural appearance (regular and irregular patterns) of the different solidification regions. Therefore, various algorithms for the description of the surfaces were selected. The resulting classification gives information about where the actual extraction of information used for pool profile generation/calculation can be retrieved from.

Due to the lack of extensive ground truth data, it was necessary to find suitable texture features and to implement customized classification methods rather than to train already existing classifiers. The following sections give an overview about the selected algorithms and the respective evaluation results.

A. Gabor Filter

The basic idea of using Gabor filters was to analyze spatial frequencies and their orientations within image patches. Trans-crystalline solidification areas represent areas with clearly visible frequencies and orientations whereas globular solidification areas do not. 2D Gabor filters are sinusoid functions combined with a Gaussian (see Figure 4) [6].

Two classes of training patches were created for globular and trans-crystalline solidification areas. These patches

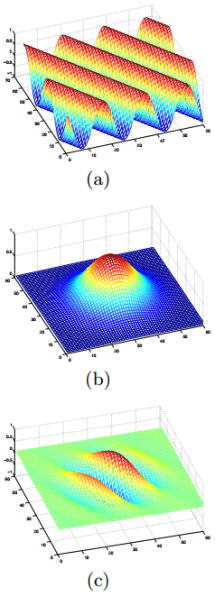


Fig. 4: Gabor filter composition: (a) 2D sinusoid oriented at 30° with the x-axis, (b) a Gaussian kernel, (c) the corresponding Gabor filter [6].

were used to generate covariance descriptors of Gabor filter outputs with one frequency and six orientations. Nearest neighbor classification was used for evaluation.

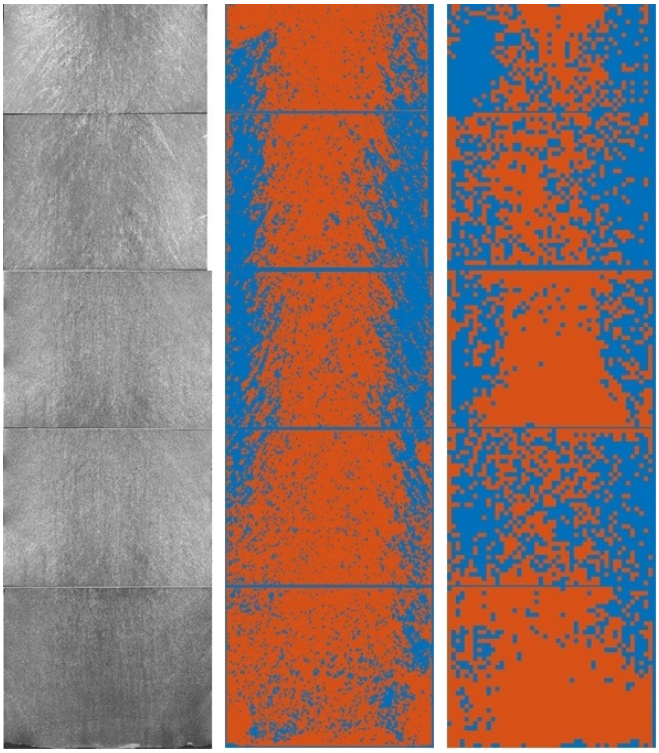
Figure 5 shows a whole steel block and the segmentation results. From Figure 5b it is obvious that the classification output delivers plausible results for the trained type of steel, although the trans-crystalline areas are not perfectly classified if the orientation of the solidification structure does not perfectly match the trained ground truth data.

B. Spatial Filter Bank

The paper presented by Ahmadvand and Daliri [1] introduces a way to perform invariant texture classification by using a spatial filter bank in multi-resolution analysis. The generated features comprise l_1-norm, standard deviation and entropy calculated from the spatial filter bank results of the original patch and the discrete wavelet transformed patch. Proposed filters are Gaussian, Laplacian of Gaussian and local standard deviation.

Same as for Gabor filters, two different patch classes are used to set up two feature matrices. For classification simple Mahalanobis distances between the feature vector and the matrices are calculated to determine class affiliation.

Although certain regions (middle and bottom in Figure 5c) are extracted more homogeneously than in the Gabor filter approach, the classification output does not yield a satisfactory result as it is too dependent on selection of training patches. The filter bank matches good within direct surroundings of training patch areas, whereas other areas cannot clearly be separated.



(a) Original image. (b) Gabor filter output with nearest neighbor classification. (c) Spatial filter bank output with Mahalanobis distance classification.

Fig. 5: Original image and segmentation output.

C. Local Binary Patterns (LBP)

LBP [10] are used to describe the surrounding of a pixel. This is done by comparing a pixel to each of its neighbors (which [10] defines by radius and number of points on the consequential circle). Given eight neighbors LBP result in an eight digit binary number where each digit gives information about whether the center point value is greater/equal or smaller than its neighbor. To retrieve information about a larger area LBP for each pixel in that area are summed up in a histogram illustrated in Figure 6.

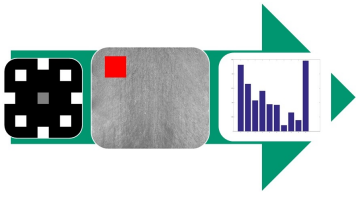


Fig. 6: LBP histogram generation.

To be able to determine certain edge and line information of an area's histogram, we decided to summarize inverted patterns, same orientation patterns or patterns that just describe noise. Overall dominating bins like noise and white/black dots are deleted from the histogram. Following those steps, it is possible to determine features (histogram bins) that correlate with the desired regions.

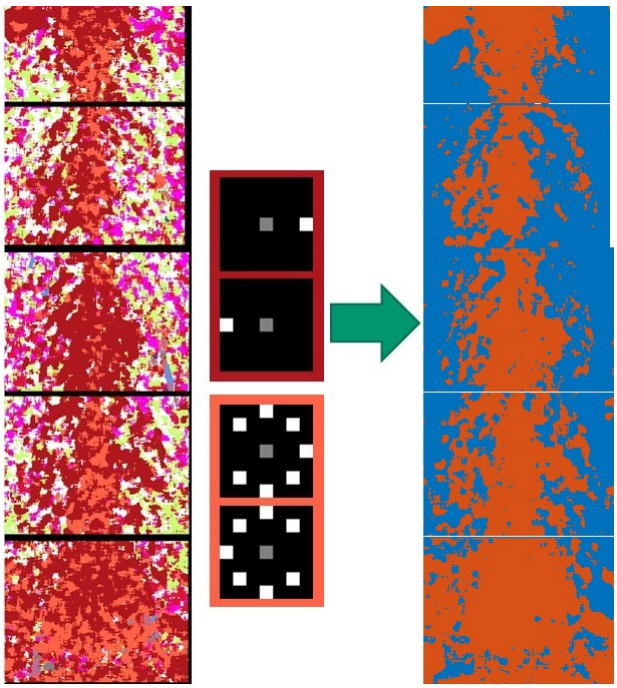


Fig. 7: LBP dominating feature/bin output.

Figure 7 shows a color coded image on the left hand side where each color matches a bin from the summarized histogram. The output image on the right hand side was generated by using a majority filter calculating the dominating feature for a specified area around a center pixel and then plotting its assigned color in the output image. The background colors of the illustrated patterns (in the middle) correlate with the colors in the left image. Together they represent the orange area within the final binary output image on the right. It is clearly evident that horizontal lines (line endings) smoothly correlate with globular solidification areas whereas trans-crystalline areas are dominated by other orientations.

D. Feature Comparison

Experiments with different steel compositions have shown that the Gabor, as well as the spatial filter bank approach, do not deliver generic solutions. Even for equal types of steel with other block dimensions, those algorithms do not deliver satisfying results.

Interestingly, the discovery that dominating horizontal orientations correlate with the globular solidification area, was also proven for further steel blocks. The validation of the segmentation output was performed by metallurgists visually. Thus, the segmentation based on LBP (Figure 7) is used as basis for the quality parameter extraction.

VI. Pool Profiles

As previously mentioned, pool profiles are used to determine quality parameters. Therefore, a fast and reliable process that can produce repeatable results with a minimum need of human interaction is required.

The best performing method during analysis of different steel types is based on a combination of scale-space and ridge detection. The ridge detection is similar to a biometric fingerprint recognition approach [4] with the difference that in this application regions with constant directions are important, whereas in fingerprint recognition characteristics like crossing points or ridge ends are relevant.

A. Ridges and Orientations

The algorithm for ridge preparation, extraction and orientation calculation is based on a paper presented by Hong, Wan and Jain [4]. They show a way to identify and normalize ridge regions within an image and to calculate their orientations. Figure 8 illustrates the process of pool profile derivation on a small sample sector of a steel specimen.

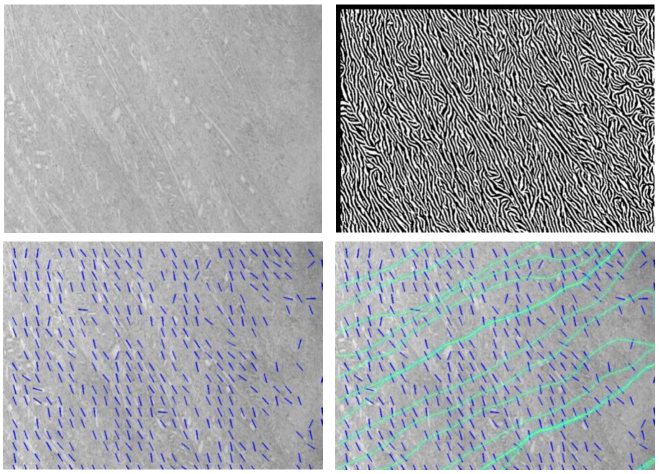


Fig. 8: Pool profile detection by ridge analysis. Top: Sector of steel specimen and derived ridges. Bottom: Derivation of orientations and final pool profile.

B. Orientation Filtering

The cutting or etching process in preparation of the steel sample or the imaging/scanning process itself can lead to artifacts. In order to handle those problematic areas, it is necessary to implement a filtering algorithm for the ridge orientations.

The first step of optimization takes place during preprocessing and delivers a mask of non-valid areas through a gray scale segmentation process performed on a smoothed and re-sampled image of the steel specimen.

The second step is the filtering of derived orientations. This filtering relies on homogeneity properties of the orientations in image areas. If an orientation vector is not in conformity with its surrounding/neighboring orientations it is treated as outlier and, thus, filtered before deriving the final pool profile.

Additionally, as a third step orientations are calculated on different scales of the image to pre-filter orientations deviating from smaller scales.

Figure 9 shows a sample steel plate and the calculated orientations (short blue lines) with and without filtering. It

is clearly evident that areas with little or even no information content were masked out. The resulting orientations are smaller in number, but more expressive.

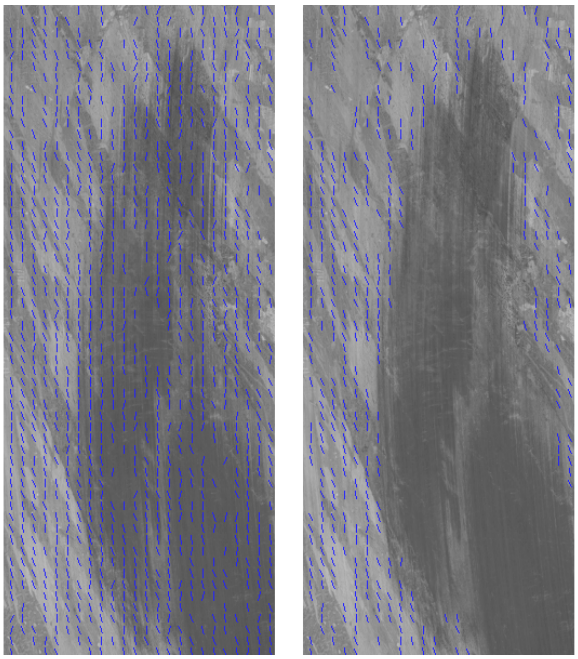


Fig. 9: Optimization of orientation detection. Left: Result without filtering. Right: Result with filtering.

C. Pool Profile Results

The pool profile itself comprises of trace lines derived from ridge orientations. Each trace line is calculated from a given individual starting point by calculating a normal on the underlying orientation to the consequential next one and so forth. The calculation begins either from the outer borders (left and right) to the middle or vice versa. Figure 10 shows an example of automatic generated pool profiles overlaid on automatically detected orientations. The two colors of the trace lines represent the different starting orientations.

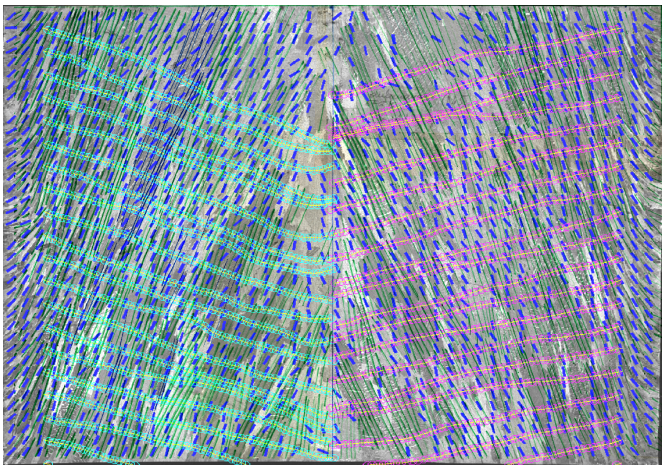


Fig. 10: Automatic generated pool profiles from the sample steel block displayed in Figure 3.

Metallurgists verified the quality of this approach by comparing the manually derived ground truth (Figure 3) with the achieved results (Figure 10). The comparison shows the good correspondence of manually generated ground truth with automated derived pool profiles.

VII. Conclusions and Outlook

This paper presented algorithms to perform steel specimen segmentation for classification of globular and transcrystalline solidification areas and algorithms to automate pool profile generation. Figure 11 displays a whole steel block with segmentation and pool profiles. The automated quality assessment is currently under evaluation by metallurgists on additional steel blocks.

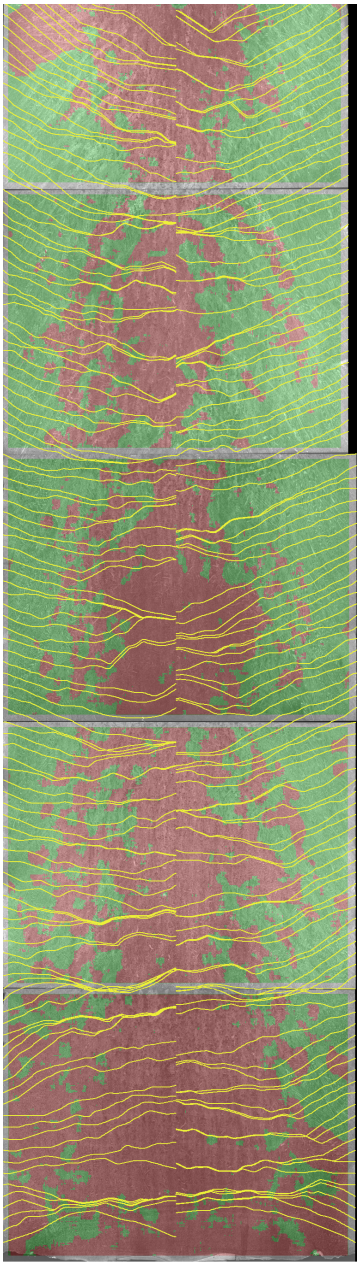


Fig. 11: Final result with segmentation and pool profiles.

First feedback indicates that the method for segmentation and pool profile generation is applicable for a wide range of steel products. This might require further implementations and/or parametrization for segmentation and pool profile generation. In the future, as image acquisition will take place regularly and, thus, more data will be available, we intend to investigate approaches based on deep learning, that will enhance automated segmentation and quality assessment even further.

ACKNOWLEDGMENT

This research was partly funded by BMVIT/BMWFJ under COMET programme, project nr. 836630, by "Land Steiermark" trough SFG under project nr. 1000033937, and by the 'Vienna Business Agency'.

REFERENCES

[1] A. Ahmadvand and M. R. Daliri, "Invariant texture classification using a spatial filter bank in multi-resolution analysis," *Image and Vision Computing*, vol. 45, pp. 1 – 10, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885615001328

[2] S. I. Chang and J. S. Ravathur, "Computer vision based non-contact surface roughness assessment using wavelet transform and response surface methodology," *Quality Engineering*, vol. 17, no. 3, pp. 435–451, 2005. [Online]. Available: http://dx.doi.org/10.1081/QEN-200059881

[3] B. L. DeCost and E. A. Holm, "A computer vision approach for automated analysis and classification of microstructural image data," *Computational Materials Science*, vol. 110, pp. 126 – 133, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927025615005066

[4] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, Aug 1998.

[5] N. Neogi, D. K. Mohanta, and P. K. Dutta, "Review of vision-based steel surface inspection systems," *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 50, 2014. [Online]. Available: http://dx.doi.org/10.1186/1687-5281-2014-50

[6] V. S. N. Prasad and J. Domke, "Gabor filter visualization," in *Technical Report*. Maryland: University of Maryland, 2005.

[7] A. Rinnhofer, W. Benesova, G. Jakob, and M. Stockinger, "Feature extraction from micrographs of forged nickel based alloy," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 2, 2006, pp. 391–394.

[8] W. Schützenhöfer, G. Reiter, R. Tanzer, H. Scholz, R. Sorci, F. Arcobello-Varlese, and A. Carosi, "Experimental investigations for the validation of a numerical pesr model," in *International Symposium on Liquid Metal Processing and Casting*. Nancy, France: SF2M, 2007, pp. 49–55.

[9] D. M. Stefanescu and R. Ruxanda, *ASM Handbook Volume 9: Metallography and Microstructures*. ASM International, 2004, ch. Fundamentals of Solidification, pp. 71–92.

[10] M. T., "The local binary pattern approach to texture analysis - extensions and applications." Ph.D. dissertation, Infotech Oulu and Department of Electrical and Information Engineering, University of Oulu, 2003, dissertation. Acta Univ Oul C 187, 78 p + App. [Online]. Available: http://herkules.oulu.fi/isbn9514270762/

[11] R. Tanzer, A. Graf, W. Schützenhöfer, and G. Reiter, "Description and validation of a var-model for a high strength maraging steel," in *2nd International Conference on Modelling of Metallurgical Processes*, Graz, Austria, 2007.

Fusion of Point Clouds derived from Aerial Images

Andreas Schönfelder[1,2], Roland Perko[1], Karlheinz Gutjahr[1], and Mathias Schardt[2]

Abstract— **State of the art dense image matching in combination with advances in camera technology enables the reconstruction of scenes in a novel high spatial resolution and offers new mapping potential. This work presents a strategy for fusing highly redundant disparity maps by applying a local filtering method to a set of classified and oriented 3D point clouds. The information obtained from stereo matching is enhanced by computing a set of normal maps and by classifying the disparity maps in quality classes based on total variation. With this information given, a filtering method is applied that fuses the oriented point clouds along the surface normals of the 3D geometry. The proposed fusion strategy aims at the reduction of point cloud artifacts while generating a non-redundant surface representation, which prioritize high quality disparities. The potential of the fusion method is evaluated based on airborne imagery (oblique and nadir) by using reference data from terrestrial laser scanners.**

I. INTRODUCTION

While the processing of aerial and satellite imagery for the generation of 2.5D Digital Elevation Models (DEM) from Multi-View Stereo (MVS) systems is a standard procedure in the field of photogrammetry and remote sensing, the reconstruction of complex 3D scenes poses several new challenges. Therefore, this work focuses on a 3D fusion of point clouds, in contrast to classical mapping approaches that only produce and fuse 2.5D DEMs or elevation maps (cf. [14]). In order to process large frame airborne and satellite imagery, it is necessary to ensure that the MVS system is capable of processing data of arbitrary size in adequate runtime at highest possible geometric accuracy.

The main contribution of this work is an easy to implement, scalable 3D point cloud fusion strategy which builds on classic multi-view stereo pipelines. By restricting, respectively weighting, disparities based on their quality it is possible to generate surface representations of large-scale datasets in adequate runtime, simultaneously reducing the redundancy in the point cloud and increasing the geometric accuracy.

II. STATE OF THE ART

Typically, the processing of multiple stereo images yields one depth map or disparity map per stereo pair. To generate one consistent, non-redundant representation of the mapped scene, the depth maps have to be fused. Some MVS systems tackle this problem by linking surface points directly in the process of image matching. In contrast, MVS systems like PMVS [4], use multi-photo consistency measures to optimize position and normals of surface patches and iteratively grow the surface starting from a set of feature points. In many MVS systems, depth maps are generated via Semi-Global Matching (SGM) [6] and spatial point intersection yielding one depth map per stereo pair. SGM is one of the most common stereo matching algorithms used in mapping applications offering robust and dense reconstruction while preserving disparities discontinues.

Depth map fusion or integration is one of the main challenges in MVS and different approaches have been developed over the last decades. Authors of [17] propose an excellent benchmark dataset for the evaluation of MVS surface reconstruction methods. As mentioned in [12], the Middlebury MVS benchmark test demonstrates that global methods tend to produce the best results regarding completeness and accuracy, while local methods like [3] offer good scalability at smaller computational costs. Moreover MVS methods can be categorized based on their representation which can differ from voxels, level-sets, polygon meshes up to depth maps [17]. Authors like [5] and [15] focus on the fusion of depth maps to generate oriented 3D point clouds. The surface reconstruction in terms of fitting a surface to the reconstructed and fused points is defined as a post-processing step which can be solved using algorithms like the generic Poisson surface reconstruction method proposed by Kazhdan *et al.* [8].

Regarding the processing of aerial imagery scalability is an important factor. As mentioned in [12], a number of scalable fusion methods have been presented in the last years, e.g. [3], [11], [18], yet they are still not able to process billions of 3D points in a single day or less [18]. Kuhn *et al.* [9] propose a fast fusion method via occupancy grids for semantic classification. The fusion method complements state-of-the-art depth map fusion as it is much faster. However, it is only suitable for applications that have no need for dense point clouds. All of the mentioned scalable fusion methods have in common, that octrees are used as underlying data structures. Kuhn *et al.* [10] introduce an algorithm for division of very large point clouds. They discuss different data structures and their capability for the decomposition of reconstruction space. In addition, Kuhn *et al.* [12] show that the 3D reconstruction of fused disparity maps can be improved by modeling the uncertainties of disparity maps. These uncertainties are modeled by introducing a feature based on Total Variation (TV) which allows pixel-wise classification of disparities into different error classes. Total variation in context with MVS was first introduced by Zach *et al.* [19]. They propose a novel range integration method using a global energy functional containing a TV regularization force and an L^1 data fidelity term for increased robustness to outliers.

[1]Joanneum Research Forschungsgesellschaft mbH, Steyrergasse 17, 8010 Graz, Austria {firstname.lastname}@joanneum.at

[2]Graz University of Technology, Steyrergasse 30, 8010 Graz, Austria {firstname.lastname}@tugraz.at

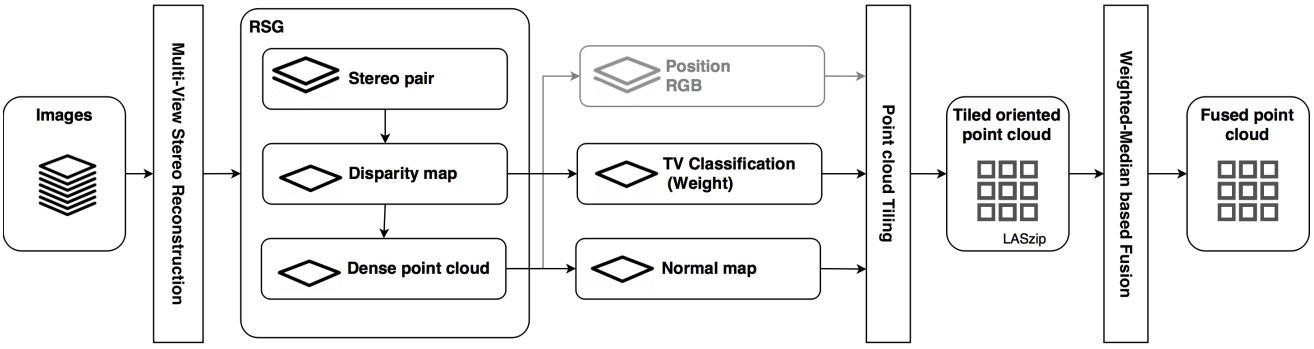


Fig. 1. Workflow of the processing pipeline for point cloud fusion.

As mentioned before, Rothermel *et al.* [15] fuse depth maps in terms of oriented 3D point cloud generation. They introduce a local median-based fusion scheme which is robust to outliers and produces surfaces comparable to the results of the Middlebury MVS. Similar to Fuhrmann and Goesele [3] points are subsampled using a multi-level octree. Favoring points with the smallest pixel footprint, an initial point set is created utilizing nearest neighbor queries optimized for cylindrical neighborhoods, points are then iteratively filtered along line of sight or surface normals. The capability of the fusion strategy for large scale city reconstruction and the straight forward manner for implementation make it particularly interesting for this work. In our work we adopt the concept of the fusion strategy using a weighted median approach favoring high quality disparities assessed by a total variation based classification.

III. METHODOLOGY

The proposed framework builds upon the Remote Sensing Software Graz (RSG)[1]. The photogrammetric processing (i.e. image registration, stereo matching) leads to different intermediate results which are utilized in the processing pipeline (see Fig. 1). Disparity maps are derived from a set of epipolar rectified images using a matching algorithm based on SGM [6]. Forward and backward matching are employed to derive two point clouds via spatial point intersection per stereo pair whose coordinates are stored in East-North-Height (ENH) raster files (i.e. a three band raster file holding the coordinates in geometry of the disparity map). The advantage of this approach is that coordinates can be accessed directly while preserving the spatial organization, i.e. the structure, of the point cloud.

In the next step, surface normals and weights are computed and stored into a compressed LAS file (i.e. a lossless compressed data format for point cloud data) [7]. Subsequently, the point clouds are assigned to tiles in order to enable a tile-wise fusion of the data. Fig. 1 depicts the complete workflow of the presented processing pipeline.

[1] `http://www.remotesensing.at/en/remote-sensing-software.html`

A. Oriented Point Cloud Generation

While in Rothermel *et al.* [15] normals are derived based on a restricted quadtree triangulation [13], we estimate surface normals in a least squares manner. A moving window operation is applied on the ENH raster files. Normals are derived by locally fitting a plane to the extracted point neighborhood. The normal estimation fails in areas with less than three reconstructed disparities. By introducing a threshold defining a minimum number of successfully reconstructed points, we are able to control the robustness of the normal calculation. In our experiments we set the pixel neighborhood to 5 pixels and used a threshold of 3 points for all datasets.

B. Disparity Quality Assessment

The quality of disparities is affected by many factors like variation of texture strength and surface slant. We assess the quality of disparities in order to derive weights for every single observed point. These weights are later used in the fusion procedure using a weighted-median approach. Kuhn *et al.* [12] introduced a TV-L^2 based classification of the disparities uncertainty. In contrast to many TV-L^1 based MVS methods, the L^2 norm takes noise and outliers into consideration which is required to measure the quality of the disparities. The TV is calculated over square windows with increasing radius m resulting in $n \in [1, 20] \subset \mathbb{N}$ discrete classes. Starting from a neighborhood containing 8 connected pixels at a radius of $m = 1$ it increases by the factor of $8m$. The discretization is achieved by introducing a regularization term τ which limits the TV to stay below a certain value. These TV classes describe the degree of the local oscillation of the disparities. The outlier probability can be obtained by learning error distributions from this classification using ground truth disparities. In our case we evaluate the quality of the disparities based on the work of Kuhn *et al.* [12] using a regularization term of $\tau = 2$.

Due to the lack of ground truth disparities, we are not able to learn error distributions directly. Therefore, we analyze the quality of the classified disparities in 3D space. Reference data from Terrestrial Laser Scanners (TLS) is used to assess the quality of the raw dense point cloud for every single TV class independently. According to Cavegn *et al.* [2], vertical Digital Surface Models (DSM) are computed for

facade patches where reference data is available. Analysing the DSM derived from the classified pointcloud and the reference data enables us to compute the weights in form of a weighting function. The weighting function is derived by calulating the standard deviation of the flatness error and fitting an exponential function in a least squares manner. The flatness error is defined as the point cloud deviations to a best fitting plane and is also an indicator for the noise of the 3D geometry [1].
Later on, we evaluate the fused pointcloud in a similiar way, to gain insight on the potential and quality of the entire fusion method. Specific information regarding the evaluation routine, selected test areas and datasets are given in Section IV.

C. Weighted-Median Based Fusion

The concept of median-based fusion originates from fusion algorithms for the generation of 2.5D DSMs. Rothermel *et al.* [15] adapted the idea by fusing point clouds in 3D space along a defined filtering direction. While for close range datasets the line of sight is suitable as filtering direction, point-wise normals are used for the fusion of aerial datasets. We adapt this fusion strategy using a weighted-median based approach.
In a first step, an initial pointset P is created from the input point cloud by storing the input point cloud in an octree data structure. The pointset P is derived by subsampling the point cloud with the centroid of the points located in a leaf node. In our work the entire fusion process was realized with the aid of the Point Cloud Library (PCL ver. 1.8.0) [16] which also provides a custom tailored octree implementation.
As a result of the disparity quality assessment every point possesses a weight representing the quality of the point. We add up the weights of all points located in the same leaf node. Thus, the weight of the initial point $p \in P$ is an indicator for the density and quality of the reconstructed scene.
Subsequently, the point cloud is fused using nearest neighbor queries optimized for cylindrical neighborhoods. For every point in the initial pointset P a set of candidate points Q, located in a cylinder with its central axis given by the initial point and its normal, is derived. Points with surface normals diverging more than 60° are discarded for further processing. After the candidate pointset Q is detected, the point p is filtered by projecting all candidate points onto the surface normal of the initial point p. Taking the weighted-median of all deviations to the point p yields the new point coordinates. Especially for noisy data further iterations can be inevitable to generate a consistent surface representation. Between every iteration, duplicate points are united to avoid redundant computations. A detailed description of the original fusion routine including the parameters and employed neighborhood queries is given in [15].
In a first iteration, Rothermel *et al.* [15] includes all points of the input point clouds for the identification of the candidate pointset Q. To speed up further iterations the filtering is restricted to the initial pointset $p \in P$ solely. In our case, we restrict the filtering of the point cloud to the initial pointset P from the beginning on. We compensate the loss of detail of the input point cloud by approximating the density of the captured 3D scene with the accumulated weight. The final surface representation is derived by discarding points with weights smaller than a defined threshold α. The influence of the threshold is analyzed in Section IV-A. In this way large and highly redundant 3D point clouds can be fused in moderate time (e.g. processing 2.5 billion points on a computer with 16 cores within a single day, resulting in a fused point cloud whose density fits the spatial resolution of the input imagery).

IV. RESULTS

In this section we discuss results obtained with the proposed fusion pipeline. The datasets used for the evaluation are provided by the ISPRS/EuroSDR project on "Benchmark on High Density Aerial Image Matching"[2] and consist of one nadir and one oblique dataset.

A. Oblique Aerial Imagery

The oblique imagery dataset was acquired over the city of Zürich with a Leica RCD30 Oblique Penta camera consisting of one nadir and four oblique 80 megapixel camera heads. While the nadir camera head is pointing downwards, directly towards the earth, the four oblique camera heads are tilted at an angle of 35 degrees, each pointing in a different cardinal direction. The entire datasets comprises 135 images, captured from 15 unique camera positions. While the nadir imagery leads to a Ground Sample Distance (GSD) (i.e. the spatial resolution) of 6 cm the GSD of the oblique views vary between 6 and 13 cm. Reference data captured with terrestrial laser scans provide accurate and reliable information for the evaluation of the datasets. The evaluation was carried out by computing DSM's of different facade patches distributed over the test area. More information on the image acquisition, benchmark and reference data can be found in [2].

Photogrammetric Processing and Pre-processing. In a first step, the image registration was carried out using the interior and exterior orientation parameters provided along with the image data. Subsequently images are matched in flight direction with an overlap of 70%, resulting in a total of 314 stereopairs, containing approximately 10.6 billion points. After the generation of disparity maps TV classes and normal maps are computed. As mentioned in Section IV the weighting function assigns a weight to every TV class which is then used in the fusion process.

The derived weighting function is depicted in Fig. 3 and shows that a correlation between TV classes and the geometric precision (i.e. level of noise) can be verified.

[2] http://www.ifp.uni-stuttgart.de/ISPRS-EuroSDR/ImageMatching/

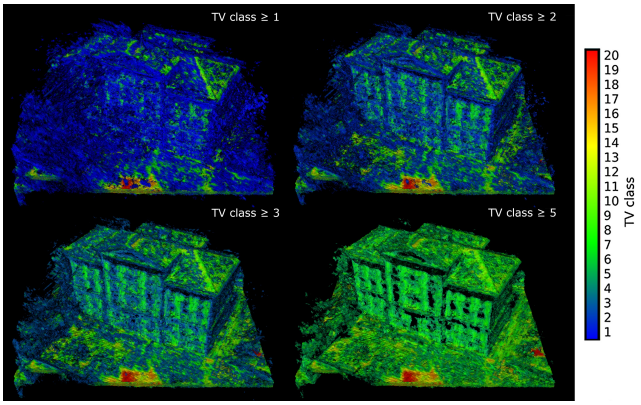


Fig. 2. Raw dense point cloud restricted to different TV classes.

While higher TV-classes show smaller standard deviations and deliver better overall accuracy, lower TV-classes are more likely to contain outliers (also cf. Fig. 2). TV classes greater than 8 are only present in flat areas facing the camera position. Since we focus on the reconstruction of vertical surfaces (i.e. facades) the information obtained by the test areas is extrapolated for all TV classes. The weighting function is derived by inverting the estimated function and defining the minimum weight with 1.0.

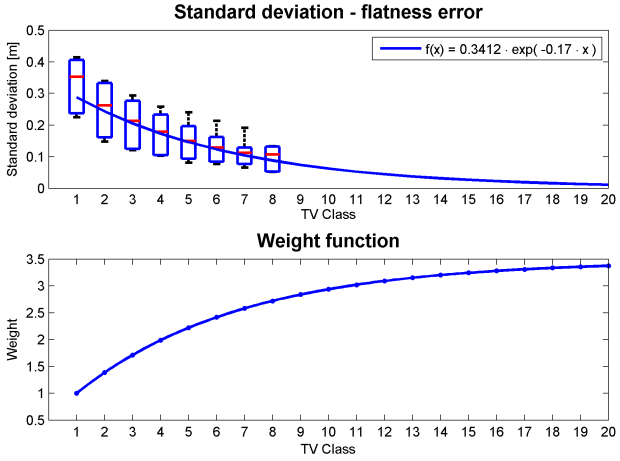


Fig. 3. Box plots representing the standard deviation of the flatness error derived from different test areas for all available TV classes *(top)*. Estimated weight function *(bottom)*.

Point Cloud Fusion. The fusion of the point cloud was carried out in three iterations with a cylinder radius of 15 cm (i.e. two times the GSD) and a height of 1.5 m. It is worth mentioning that, in some cases, during the image acquisition parts of the helicopter skids protruded into the camera angle, which leads to strong distortions in the matching procedure. The size of the octrees leaf node, which is used for the generation of the initial pointset, controls the approximate output density of the fused point cloud. Therefore, faster runtimes can be achieved producing point clouds with lower density. The resolution used for the oblique imagery is set to 10 cm, to match the GSD of the input data. Within the point cloud fusion process, the points are filtered along the surface normal and weights are accumulated. The final surface representation is derived by discarding low weights, which are more likely to contain outliers. As depicted in Fig. 4, increasing the minimum weight threshold α leads to more accurate, however less dense point clouds.

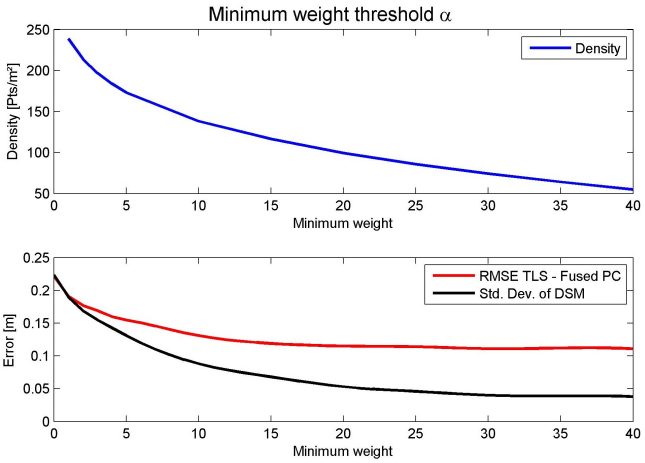


Fig. 4. Impact of rejecting low weighted points after the fusion procedure on density *(top)*, accuracy and precision *(bottom)*.

Since the fusion method produces oriented point clouds, a mesh representation can be computed using Poisson surface reconstruction [8]. The complete workflow is depicted in Fig. 5. The runtime of the fusion process can be improved by discarding low level TV classes in a pre-processing step. However, the rejection of low level TV classes causes a loss in detail in areas with bad coverage.

Evaluation. In order to measure the capability of the fusion routine different statistical measures are analyzed. The RMSE of the deviations between the reference point cloud and fused point cloud, give information about the accuracy of the 3D geometry. The standard deviation of the vertical digital surface model indicates the noise level of the point cloud, respectively the distribution of points perpendicular to the facade. As mentioned before, the density can be controlled by setting the octree resolution and by regulating the threshold for the minimum weight α. In Table I the raw point cloud is compared to the fused point cloud considering the influence of TV weights. The minimum weight threshold α is set to generate point clouds with comparable densities. Test areas include the school building located in the northern part of the mapped scene and the tower building located in the south.

TABLE I

COMPARISON OF THE FUSION ROUTINE REGARDING WEIGHTS.

	min. weight α	Density [pnts/m^2]	RMSE Fused PC-TLS [m]	Mean Fused PC-TLS [m]	Std. Dev. of DSM [m]
Raw (unfused)	-	4398.00	0.199	0.108	0.296
Fused (no weights)	20	75.15	0.122	0.067	0.052
Fused (weighted)	30	74.25	0.111	0.063	0.040
Fused (weighted pre-filter TV >1)	18	75.23	0.102	0.049	0.032

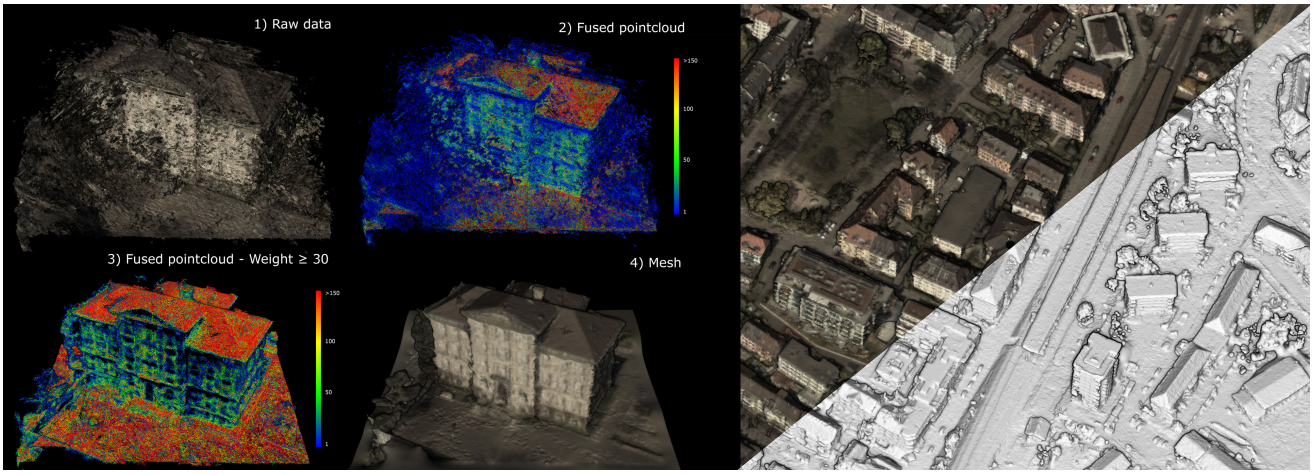


Fig. 5. Processing pipeline of the point cloud fusion: *(1)* Raw data from dense image matching (50.64 M points), *(2)* fused point cloud (1.73 M points), *(3)* discarding weights smaller than $\alpha = 30$ (0.47 M points), *(4)* mesh generation, and *(right side)* merged surface tiles.

Regarding the oblique dataset, best results can be achieved by neglecting points with TV class 1. By doing so, execution time is speed up by a factor of 2.2. Compared to the raw point cloud the fusion procedure reduces noise while improving the accuracy of the point cloud (see Fig. 6). A visual assessment shows that the fused point cloud including all TV classes and applying weights produces the best results regarding completeness and outliers (see Fig. 7). As expected, roof structures and other nadir oriented faces are reconstructed with the highest precision. Table II shows that in all cases the precision of the point cloud can be improved while decreasing redundant information.

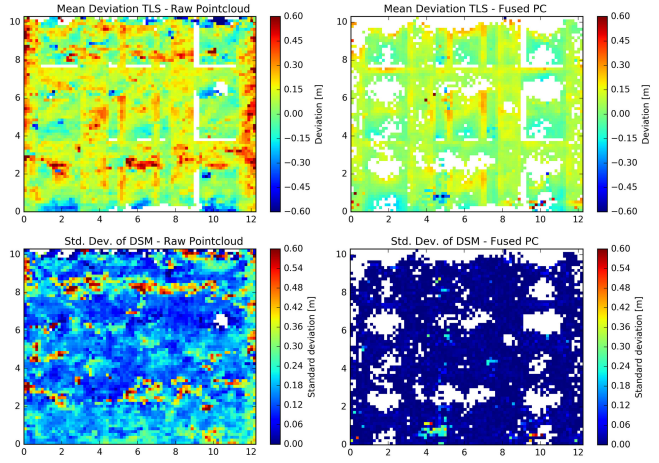


Fig. 6. Comparison of the main school facade before and after fusion procedure (cf. Fig. 5): Mean deviation between DSM derived from terrestrial laser scanner data and point cloud *(top)*, and standard deviation of the point clouds DSM representing the level of noise *(bottom)*.

Fig. 7. Taking all TV classes into account produces point clouds containing less outliers *(left)*, in contrast to point clouds restricted to TV classes > 1 *(right)*.

TABLE II

COMPARISON OF TEST AREAS BEFORE AND AFTER THE POINT CLOUD FUSION.

	Density [pnts/m^2]	RMSE Fused PC-TLS [m]	Mean Fused PC-TLS [m]	Std. Dev. of DSM [m]
Tower South (raw)	2345.9	0.378	0.051	0.538
Tower South (fused)	49.4	0.204	0.003	0.087
Tower North (raw)	1781.4	0.427	-0.222	0.447
Tower North (fused)	45.3	0.195	-0.052	0.071
Tower West (raw)	3570.8	0.350	0.237	0.499
Tower West (fused)	62.7	0.256	0.152	0.155
Roof (raw)	13864.2	0.150	-0.023	0.218
Roof (fused)	178.7	0.122	0.028	0.105

B. Nadir Aerial Imagery

The nadir image dataset covers an area of approximate 1.5 $\times$ 1.7 km^2 in the city of Munich. The dataset was acquired by a DMC II 230 megapixel aerial image camera with a spatial resolution of 10 cm and consists of 15 panchromatic images. As depicted in Fig. 8, facade information can be reconstructed by utilizing the proposed fusion routine. Due to the wide angle of the aerial camera, enough information is captured to produce 3D city models from nadir aerial imagery.

V. CONCLUSION

A novel method for fusing 3D point clouds was presented. The underlying point clouds originate from stereo matching of aerial images and were enriched by the calculation of surface normals and a classification of the disparity maps into quality classes. The proposed filtering method then fused the point cloud in direction of the surface normals and used a weighting based on the classification. Evaluation to ground truth data showed the increased quality of the fused point cloud while reducing the redundancy. Overall,

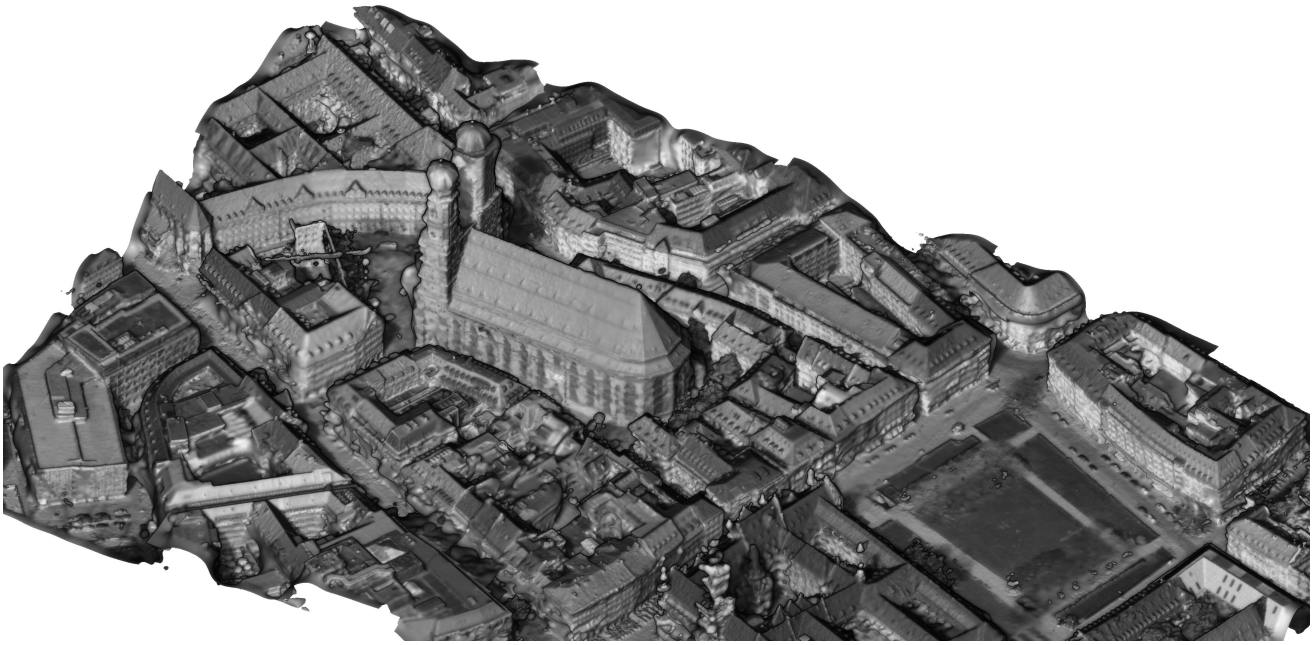


Fig. 8. Reconstructed surface from nadir aerial imagery. The depicted surface shows the Frauenkirche in Munich, located in the west part of the test area. Therefore, west-facing facades cannot be reconstructed.

this fusion concept can be easily put into state-of-the-art mapping pipelines, is able to handle large point clouds due to the tiling concept and can be applied for terrestrial, aerial or satellite based mapping application.

ACKNOWLEDGMENT

This research was partly funded by BMVIT/BMWFW under COMET programme, project nr. 836630, by Land Steiermark through SFG under project nr. 1000033937, and by the Vienna Business Agency.
The authors would like to thank Stefan Cavegn and Norbert Haala for providing the terrestrial laser scanner reference data.

REFERENCES

[1] A. H. Ahmadabadian, S. Robson, J. Boehm, M. Shortis, K. Wenzel, and D. Fritsch, "A comparison of dense matching algorithms for scaled surface reconstruction using stereo camera rigs," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 78, pp. 157 – 167, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924271613000452

[2] S. Cavegn, N. Haala, S. Nebiker, M. Rothermel, and P. Tutzauer, "Benchmarking High Density Image Matching for Oblique Airborne Imagery," *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 45–52, Aug. 2014.

[3] S. Fuhrmann and M. Goesele, "Fusion of depth maps with multiple scales," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 148:1–148:8, Dec. 2011. [Online]. Available: http://doi.acm.org/10.1145/2070781.2024182

[4] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, Aug 2010.

[5] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *IEEE International Conference on Computer Vision (ICCV)*, June 2015.

[6] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb 2008.

[7] M. Isenburg, "Laszip," *Photogrammetric Engineering and Remote Sensing*, vol. 79, no. 2, pp. 209–217, 2013.

[8] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 29:1–29:13, July 2013. [Online]. Available: http://doi.acm.org/10.1145/2487228.2487237

[9] A. Kuhn, H. Huang, M. Drauschke, and H. Mayer, "Fast probabilistic fusion of 3d point clouds via occupancy grids for scene classification," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. III-3, pp. 325–332, 2016. [Online]. Available: http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/III-3/325/2016/

[10] A. Kuhn and H. Mayer, "Incremental division of very large point clouds for scalable 3d surface reconstruction," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 157–165.

[11] A. Kuhn, H. Hirschmüller, and H. Mayer, *Multi-Resolution Range Data Fusion for Multi-View Stereo Reconstruction.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 41–50. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40602-7_5

[12] A. Kuhn, H. Hirschmüller, D. Scharstein, and H. Mayer, "A TV prior for high-quality scalable multi-view stereo reconstruction," *International Journal of Computer Vision*, pp. 1–16, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11263-016-0946-x

[13] R. Pajarola, "Large scale terrain visualization using the restricted quadtree triangulation," in *Visualization*, Oct 1998, pp. 19–26.

[14] R. Perko and C. Zach, "Globally optimal robust DSM fusion," *European Journal of Remote Sensing*, vol. 49, pp. 489–511, Sept. 2016.

[15] M. Rothermel, N. Haala, and D. Fritsch, "A Median-Based Depthmap Fusion Strategy for the Generation of Oriented Points," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 115–122, June 2016.

[16] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[17] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, June 2006, pp. 519–528.

[18] B. Ummenhofer and T. Brox, "Global, dense multiscale reconstruction for a billion points," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015. [Online]. Available: http://lmb.informatik.uni-freiburg.de//Publications/2015/UB15

[19] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust tv-l1 range image integration," in *IEEE International Conference on Computer Vision (ICCV)*, Oct 2007, pp. 1–8.

Superresolution Alignment with Innocence Assumption: Towards a Fair Quality Measurement for Blind Deconvolution

Martin Welk[1]

Abstract— **Quantitative measurements of restoration quality in blind deconvolution are complicated by the necessity to compensate for opposite shifts of reconstructed image and point-spread function. Alignment procedures mentioned for this purpose in the literature are sometimes not exactly enough specified; alignment-free approaches sometimes do not take into account the full variability of possible shifts. We investigate by experiments on a simple test case the errors induced by interpolation-based alignment procedures. We propose a new method for MSE/PSNR measurement of image pairs involving non-integer displacements that is based on a superresolution approach. We introduce an innocence assumption in order to keep deviations that can be explained by shifted sampling grids out of the error measurement. In our test case, the new measurement procedure reduces the variations in MSE/PSNR measurements substantially, creating the hope that it can be used for valid comparisons of blind deconvolution methods.**

I. INTRODUCTION

The removal of blur in images by blind image deconvolution has been studied for many years [2], [3], [4], [5], [6], [10], [16], [21], and received increasing interest during the last years [1], [7], [8], [9], [11], [12], [14]. A frequently used simplifying assumption is that the blur is spatially invariant, i.e. the redistribution of intensity is described by the same point-spread function (PSF) h at each image location. Blur is then described by a convolution between the unobserved sharp image g and the PSF h; incorporating additive noise n, the observed image f is given by the blur model

$$f = g * h + n \,. \tag{1}$$

Whereas for non-blind deconvolution one assumes that f and h are known, and aims at an estimate u for the sharp image g, the knowledge of h is often not available in practice, thus necessitating blind deconvolution where the estimate u of the sharp image is to be obtained along with the PSF h, using only f as input image.

A variety of approaches to solve this task have been developed, creating the need for quality comparisons. Besides visual assessment, one is interested in quantitative measurements of reconstruction quality versus a known ground truth.

Frequently used standard measures for image reconstruction methods include the mean-square error (MSE) as well as the signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR) both of which are closely related to the MSE; furthermore, sometimes the average absolute error (AAE) is advocated. Another measure that puts some more emphasis on important structural details of images such as contrast edges is the structural similarity index (SSIM), see [17]. Let us shortly recall the first three measures.

For a reference (ground-truth) image g and degraded (or reconstructed) image u, both of size $n \times m$ pixels, their MSE is given by

$$\mathrm{MSE}(u,g) = \frac{1}{nm} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (u_{i,j} - g_{i,j})^2 \,. \tag{2}$$

Provided that u and g have equal mean intensity μ (which we will assume in the following), this is the variance $\mathrm{var}(u-g)$ of $u-g$. Using the variance of g given by

$$\mathrm{var}(g) = \frac{1}{nm} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (g_{i,j} - \mu)^2 \,, \tag{3}$$

and the range $R(g) := \max_{i,j} g_{i,j} - \min_{i,j} g_{i,j}$ (255 for saturated 8-bit images), one can compute the SNR

$$\mathrm{SNR}(u,g) = 10 \log \frac{\mathrm{var}(g)}{\mathrm{var}(u-g)} \,\mathrm{dB} \tag{4}$$

and PSNR

$$\mathrm{PSNR}(u,g) = 10 \log \frac{R(g)^2}{\mathrm{var}(u-g)} \,\mathrm{dB} \,. \tag{5}$$

For non-blind deconvolution, both MSE/(P)SNR and SSIM are frequently used to assess reconstruction quality. Although these quantitative measures are not always in good agreement with visual assessments by humans, they are generally accepted as simple and objective measures. For a recent study on measures that approximate better the human perception of restoration quality see [13].

In blind deconvolution, however, their application meets a difficulty: If the reconstructed image u is translated by an arbitrary, often non-integer, displacement d, and the point-spread function h is translated by $-d$, these translations cancel in the convolution $u * h$. Blind deconvolution results that differ just by such opposite translations of u and h must therefore be considered equally valid reconstructions. An example of such shifts that indeed occur in blind deconvolution results is shown in Fig. 1. This precludes a straightforward (P)SNR or SSIM comparison of blind deconvolution results with ground truth. Obviously, some kind of alignment – rigid registration restricted to translations as transformations – has to be applied.

Nevertheless, blind deconvolution results are compared by PSNR and other quantitative measures in a number of works, e.g. [6], [7], [8], [9], [10], [14]. In many of these,

*This work was not supported by any organization

[1]Martin Welk is with Department of Biomedical Informatics and Mechatronics, Private University for Health Sciences, Medical Informatics and Technology (UMIT), 6060 Hall/Tyrol, Austria martin.welk@umit.at

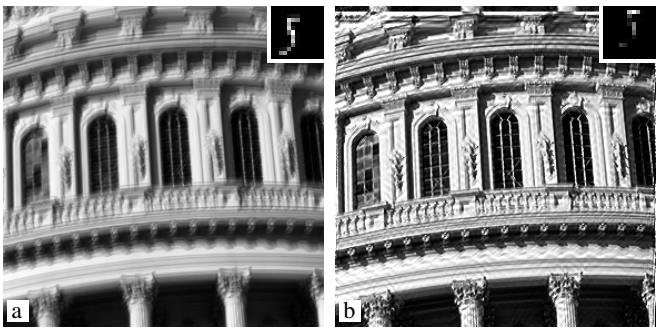


Fig. 1. **(a)** Synthetically blurred image with ground-truth PSF, from [11]. – **(b)** Blind deconvolution result with PSF, from [12]. Note the opposite shifts of image and PSF.

no alignment whatsoever is mentioned [6], [8], [9], [10]. Such an evaluation relies implicitly on the assumption that estimated PSFs are aligned with the ground truth PSF; probably this is approximately achieved by some test cases with small PSF support. Efforts to compensate shifts, either for images or for PSFs, are found in [7], [14], [20]. A benchmark established in [7] is based on simulating camera shake by generated trajectories. Multiple ground truth images are acquired directly along those trajectories, and the best match is used for error measurement. On one hand, a computational alignment step is avoided in this way. On the other hand, the procedure constrains shifts to the ground-truth trajectory which may be insufficient since blind deconvolution methods can well yield translations in which the coordinate origin of the PSF does not happen to be on the (unknown) trajectory that was used to generate the ground truth. The benchmark from [7] is also used in [20] and part of the evaluation in [14]. Further tests in [14] are based on data from [9]. Here, absolute errors of PSFs are measured, namely for "(aligned) blur estimates" with respect to ground truth PSFs. This allows indeed to handle unconstrained displacements. Details of the alignment procedure are not given, however.

In the following we discuss how to make precise such an alignment procedure. We focus on a scenario where a ground-truth image and PSF are available, and restoration quality is to be estimated by measuring the error between the ground truth and reconstructed images. In specifying the alignment procedure, some choices have to be made: first, should one register the reconstructed image to the ground truth image, or vice versa, or should perhaps both be transformed? Which interpolation procedure is to be used in the registration process? It is not a far-fetched guess that these details will influence the subsequent error measurements. In fact, we will demonstrate by a simple experiment in Section II that, dependent on details of the registration, the PSNR measures vary by 1.5 dB and more.

Given the fact that relative improvements of one blind deconvolution method over the other as reported in e.g. [7], [14] often amount to as little as 0.5 dB or even less, such a difference is significant.

This might be mitigated by using multiple test images and performing statistics on the errors measured for these. However, questions remain: Since errors introduced by interpolation can be expected to differ substantially between test cases where the displacement is approximately integer, and test cases where the displacement is near a half-pixel position, results may be strongly biased towards blind deconvolution methods that, for whatever reason, tend to reconstruct PSFs in similar pixel alignment as the ground-truth. Given the complexity of procedures both for constructing apparently realistic test cases, and of the blind deconvolution procedures themselves, it is such favourable alignments occur more often for some methods under investigation than for others. In such a case, the bias won't necessarily average out for larger sample sizes.

For this reason, we pursue in this paper the goal to establish an alignment procedure for blind deconvolution results that avoids these pitfalls. We focus here on the MSE, from which (P)SNR can be derived via (4), (5).

Structure of the paper. In Section II we evaluate the errors introduced by interpolation-based alignment procedures using a simple test case. Section III establishes the fundamentals of an alignment procedure by superresolution in order to avoid these errors. The details of the procedure are discussed in Section IV, followed by experiments on the previously introduced test case in Section V. A short summary and outlook in Section VI concludes the paper.

II. ALIGNMENT BY INTERPOLATION

To assess the errors introduced by alignment with interpolation, we set up a simple test case based on a ground truth grey-value image shown in Fig. 2 (a). We blur this image by 16 different PSFs shown in Fig. 2 (b); all these PSFs are downsampled versions of the same high-resolution PSF with horizontal and vertical shifts in $1/4$ pixel steps. One blurred image is shown in Fig. 2 (c). Each of the blurred images is deconvolved with each of the 16 PSFs using the non-blind deconvolution method from [18] with the same parameters ($\alpha = 0.01$, 300 iterations). This yields 256 deblurred images with effective shifts w.r.t. the ground truth images from -0.75 to $+0.75$ pixels in x and y direction; one exemplary deblurred image is shown in Fig. 2 (d).

We can now measure the MSE (and resulting PSNR) for each deblurred image w.r.t. the ground truth image. In the following we report PSNR values as this is the most familiar measure in deconvolution literature. To reduce the impact of boundary artifacts, a 20 pixel wide margin is excluded from the measurement, thus using a 88×88 central patch of the reference image.

We notice first that in the 16 translation-free cases (where the same PSF was used for blurring and deblurring) the PSNR varies between 29.74 and 30.41 dB, with an average of 30.07 dB and a standard deviation of 0.21 dB.

Next, we measure PSNR values for the entire set of 256 deblurred images. Here, the ground-truth and reconstructed images are aligned using either bilinear and bicubic interpolation with the ground-truth shift values. For the direction of alignment we consider three settings: (a) warping the

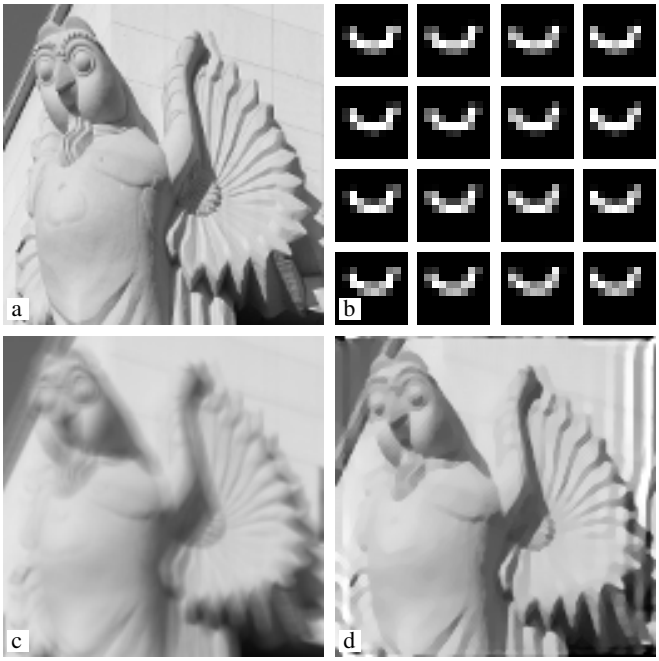


Fig. 2. **(a)** Ground truth image, 128×128 pixels. (Clipped, downscaled and converted to greyscale from a photograph of the building of TU Vienna. Source of original image: https://upload.wikimedia.org/wikipedia/commons/e/e9/TU_Bibl_01_DSC1099w.jpg, Author: Peter Haas. Available under licence CC BY-SA 3.0.) – **(b)** 16 PSFs, 10×10 pixels each, subsampled from the same high-resolution input. The shift from row to row and from column to column is 0.25 pixels. – **(c)** Image (a) blurred by convolution with PSF from (b), first row, second column. – **Bottom right:** Image (c) deblurred with PSF from fourth row, third column, resulting in a shift relative to ground truth of $(0.25, 0.75)$ pixels.

reconstructed image to match the ground-truth image; (b) warping the ground-truth image to match the reconstructed image; (c) applying half the shift to each of the ground-truth and reconstructed image. Statistics of the resulting PSNR values are presented in Table I.

To bring the previous procedure closer to a true blind deconvolution setting, we now switch to determining also the displacement from a minimisation of the MSE (or maximisation of the PSNR). To avoid analysing possible multiple optima, we employ here a brute-force optimisation varying the displacements in x and y direction in 0.01 steps from -1 to 1; note that the exact displacements occur in the sequence of displacements sampled thereby. Table II contains statistics of the misestimations δx, δy of the x and y displacements, and the resulting PSNR. The latter values are slightly higher than in Table I but not seriously so.

As can be expected, warping the reconstructed image to match the ground truth (see columns marked (a) in Tables I and II) leads to lower PSNR values for image pairs with non-integer displacements. The variation is about 3 dB with bilinear interpolation; bicubic interpolation reduces it to about 2 dB which is still likely to warp comparisons substantially. When aligning instead the ground truth to the reconstructed images (columns (b) in Tables I and II) PSNR values are surprisingly higher for non-integer displacements, which means by comparison to the no-shift cases a clear overestimation of reconstruction quality. Apparently the warping of the ground truth image introduces some blur which matches well the remaining blur in the deconvolution results.

Inspection of the detail results corroborates that for the same image pair the choice which image is aligned to which one leads to discrepancies in PSNR of 4 dB and more with bilinear, and still about 3 dB with bicubic interpolation. Distributing the shift to both images (columns (c) in Table I) yields similar results as shifting the ground truth. As this proceeding does not offer an advantage, we do not pursue it further in the computationally more expensive scenario of Table II where also the displacements are optimised.

III. ALIGNMENT BY SUPERRESOLUTION

We turn now to designing a procedure for image reconstruction error measurement with alignment. We give preference to the MSE as basis of our considerations because unlike the (P)SNR it treats the two images being compared in a completely symmetric way. We want to keep this symmetry also in the alignment procedure, thereby removing one of the arbitrarities of interpolation-based alignment procedures. For easier comparison to usual PSNR figures we will nevertheless report in the experiments later PSNR values computed from our MSE measurements.

An obvious requirement is that for perfectly aligned images the standard MSE measure has to be reproduced. Whereas the procedure will be described for prescribed

TABLE I

PSNR STATISTICS FOR 256 RECONSTRUCTED IMAGES WITH ALIGNMENT BY THE KNOWN (GROUND-TRUTH) SHIFT USING BILINEAR OR BICUBIC INTERPOLATION; (A) WARPING THE RECONSTRUCTED IMAGE, (B) WARPING THE GROUND TRUTH, (C) HALF-WAY WARPING GROUND TRUTH AND RECONSTRUCTED IMAGE.

Interpolation	bilinear			bicubic		
Alignment	(a)	(b)	(c)	(a)	(b)	(c)
min	27.47	29.74	29.74	28.35	29.74	29.39
max	30.41	33.54	33.55	30.41	31.84	31.55
(max−min)	2.94	3.80	3.81	2.06	2.10	2.16
mean	28.57	32.18	31.25	29.23	30.85	30.05
standard dev.	0.711	0.970	0.805	0.474	0.489	0.459

TABLE II

STATISTICS OF DISPLACEMENT MISESTIMATIONS δx, δy AND PSNR FOR 256 RECONSTRUCTED IMAGES WITH ALIGNMENT ESTIMATED BY MSE MINIMISATION.

Interpolation	bilinear		bicubic	
Alignment	(a)	(b)	(a)	(b)
$\max\lvert\delta x\rvert$	0.18	0.17	0.07	0.16
std. dev. δx	0.079	0.081	0.028	0.080
$\max\lvert\delta y\rvert$	0.17	0.15	0.06	0.14
std. dev. δy	0.064	0.067	0.024	0.072
min PSNR	27.47	31.42	28.37	29.89
max PSNR	30.41	33.61	30.41	31.86
(max−min) PSNR	2.94	2.19	2.04	1.97
mean PSNR	28.67	32.67	29.25	31.10
std. dev. PSNR	0.697	0.527	0.471	0.465

displacement values, minimisation of the MSE measure is an obvious way to estimate also unknown displacements.

For the following, let us consider two images u and g, which are sampled representations of continuous-scale images. To specify the sampling process more precise, we assume that each pixel of g is the integral of the underlying continuous-scale image G over a rectangled region such that all pixels together tesselate (a rectangle of) the image plane:

$$g_{i,j} = \int_i^{i+1} \int_j^{j+1} G(x,y)\,\mathrm{d}y\,\mathrm{d}x\,, \tag{6}$$

and similarly for u whose grid is of equal resolution but shifted by $d = (\alpha,\beta) \in \mathbb{R}^2$,

$$u_{i,j} = \int_{i+\alpha}^{i+1+\alpha} \int_{j+\beta}^{j+1+\beta} U(x,y)\,\mathrm{d}y\,\mathrm{d}x\,. \tag{7}$$

Without loss of generality, we assume $0 \le \alpha,\beta < 1$.

Whereas in the special case of band-limited images sampled with at least their double limiting frequency, Shannon's sampling theorem guarantees that u and g contain full information on their continuous counterparts, this can usually not be expected to hold true for natural images; thus the continuous images U and G are in fact unknown.

A good measure for the discrepancy between u and g should essentially measure the discrepancy between their continuous versions U and G. In other words, we do not want to punish reconstructions for badly aligned grids, and formulate therefore an "innocence assumption" *(in dubio pro reo – in case of doubt for the defendant)*: Whatever discrepancy between two images can plausibly be attributed to different sampling, shall not enter the discrepancy measure. In particular, if a sufficiently plausible continuous-scale image $V \equiv U \equiv G$ exists from which both u and g can be obtained by sampling, their discrepancy should be measured as zero. Notice that the exact meaning of the word "plausible" remains to be specified later.

Our considerations can be boiled down to a discrete image v of size $(2n+1)\times(2m+1)$ whose pixels are the intersections of pixels of u and g:

$$v_{i,j} = \int_{\xi_i}^{\xi_{i+1}} \int_{\eta_j}^{\eta_{j+1}} V(x,y)\,\mathrm{d}y\,\mathrm{d}x\,, \tag{8}$$

$i = 0,\dots,2n,\ j = 0,\dots,2m$, where $\xi_i = i/2$ for even i and $\xi_i = i/2+\alpha$ for odd i, $\eta_j = j/2$ for even j and $\eta_j = j/2+\beta$ for odd j. Note that pixel (i,j) of g covers the four pixels $(2i,2j)$, $(2i,2j+1)$, $(2i+1,2j)$ and $(2i+1,2j+1)$ of v whereas pixel (i,j) of u covers the four pixels $(2i+1,2j+1)$, $(2i+1,2j+2)$, $(2i+2,2j+1)$ and $(2i+2,2j+2)$ of v. The image v is therefore a superresolution image [15] of g and u, albeit with pixels of different sizes. In x direction grid cells of size α alternate with such of size $1-\alpha$, whereas in y direction the same is true with β and $1-\beta$.

In the general situation when U and G cannot be chosen as equal, we want to retain this idea and construct a superresolution image v that tries to reconciliate the information of u and g as good as possible. The discrepancy of u and g will then be measured by combining discrepancies between u and v, and between v and g.

In the perfectly aligned case, $\alpha = \beta = 0$, the MSE (2) of images g and u can be combined from the MSEs between each of g and u and their average $v := \frac{1}{2}(g+u)$ via

$$\mathrm{MSE}(u,g) = 2(\mathrm{MSE}(u,v) + \mathrm{MSE}(v,g))\,. \tag{9}$$

Moreover, using the parallelogram identity (or by an easy combination of Cauchy-Schwarz' inequality with the arithmetic-geometric mean inequality) we see that for any other image v the right-hand side of (9) will be greater than $\mathrm{MSE}(u,g)$. This motivates the following definition.

Definition. *Let images u and g of size $n\times m$ sampled as in* (6), (7) *be given. Let a class X of $(2n+1)\times(2m+1)$-images v sampled as in* (8) *be given. For each image $v \in X$, define v_u, v_g as the downsamplings of v to the grids of u and g, respectively. The alignment-MSE MSE_X between u and g with respect to X is defined as*

$$\mathrm{MSE}_X(u,g) = \min_{v\in X} 2(\mathrm{MSE}(u,v_u) + \mathrm{MSE}(v_g,g))\,. \tag{10}$$

Application of this definition requires, first, the specification of the class X for given images u, g. The class X essentially defines what are plausible superresolution images. Second, a minimisation method will be needed to find the minimiser. We will turn to these issues in the next section.

IV. SPECIFYING CONSTRAINTS

Given u and g, a superresolution image v as specified in the previous section must satisfy the equations

$$\begin{aligned} &\alpha\beta v_{2i,2j} + \alpha\bar\beta v_{2i,2j+1} \\ &\quad + \bar\alpha\beta v_{2i+1,2j} + \bar\alpha\bar\beta v_{2i+1,2j+1} = g_{i,j}\,, \end{aligned} \tag{11}$$

$$\begin{aligned} &\bar\alpha\bar\beta v_{2i+1,2j+1} + \bar\alpha\beta v_{2i+1,2j+2} \\ &\quad + \alpha\bar\beta v_{2i+2,2j+1} + \alpha\beta v_{2i+2,2j+2} = u_{i,j} \end{aligned} \tag{12}$$

for $i = 1,\dots,n,\ j = 1,\dots,m$, where we have used the abbreviations $\bar\alpha := 1-\alpha$, $\bar\beta := 1-\beta$.

On one hand, these are just $2nm$ equations for $4nm + 2n + 2m + 1$ pixels of v (from which two corner pixels could be eliminated as they are neither covered by g nor by u); additional conditions will therefore be necessary to remove this underdetermination. On the other hand, for images u and g that do not match perfectly, we expect that the equations should be satisfied only approximately. which favours smoothness. Thus, we are led to reformulate our equation system into the minimisation of an energy function

$$E(v) = S_g(v) + S_u(v) \tag{13}$$

under suitable constraints, where S_g and S_u are quadratic

error terms for the equations above,

$$S_g(v) := \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} \big(g_{i,j} - \alpha\beta v_{2i,2j} - \alpha\bar\beta v_{2i,2j+1} - \bar\alpha\beta v_{2i+1,2j} - \bar\alpha\bar\beta v_{2i+1,2j+1}\big)^2 , \quad (14)$$

$$S_u(v) := \sum_{i=0}^{n-1}\sum_{j=0}^{m-1} \big(u_{i,j} - \bar\alpha\bar\beta v_{2i+1,2j+1} - \bar\alpha\beta v_{2i+1,2j+2} - \alpha\bar\beta v_{2i+2,2j+1} - \alpha\beta v_{2i+2,2j+2}\big)^2 . \quad (15)$$

Up to constant factors, S_g and S_u are just the $\mathrm{MSE}(g, v_g)$ and $\mathrm{MSE}(u, v_u)$ from the alignment-MSE definition.

Let us therefore now discuss possible constraints for this minimisation problem. These constraints will constitute the class X of images to minimise over that appeared in the definition of the alignment-MSE.

Note first that in the equations (11), (12) for subsequent indices i or j the two input images u and g alternate. This suggests that for images u and g that do not perfectly match, solutions of (11), (12) are likely to develop oscillating patterns like stripes of alternating intensity or checkerboard structures, so the discrepancy between u and g can be translated to the image boundary where the first and last row and column of v are linked only to one of the input images and therefore provide degrees of freedom that can absorb the discrepancy. In extreme, this could mean that even for completely mismatching u and g highly oscillatory images v might exist that fulfil (11), (12) without any error. Such solutions should be rejected by a suitable class X.

In order to prevent v from developing strong high-frequency structures, a natural requirement could be that v should be essentially interpolating; thus each pixel intensity $v_{i,j}$ should be in the interval bounded by the intensities $g_{\lfloor i/2\rfloor,\lfloor j/2\rfloor}$, $u_{\lfloor (i-1)/2\rfloor,\lfloor (j-1)/2\rfloor}$ of the two input pixels it is linked to by (11), (12). Whilst conceptually elegant and free of additional parameters, this constraint turns the minimisation of (13) into a quadratic minimisation problem on a highly nonconvex domain. We aim therefore at relaxing this constraint to a convex regularisation that warrants a unique solution as well as a practical minimisation procedure.

We extend therefore the energy function (13) to

$$E(v) = S_g(v) + S_u(v) + \gamma T(\nabla v) \quad (16)$$

where T is a regulariser that depends on the derivatives $\nabla v = (v_x, v_y)$ of v approximated by finite differences, and $\gamma > 0$ is a regularisation weight.

With regard to the quadratic nature of the mean square error to be measured, a Whittaker-Tikhonov regularisation

$$T(\nabla v) := \sum_{i,j} |\nabla v|^2 \quad (17)$$

lends itself as a natural candidate, which yields a convex quadratic minimisation problem, also removing completely the non-uniqueness of the original equations. Minimisers can efficiently be computed using standard iterative solution methods for the linear system of minimality conditions.

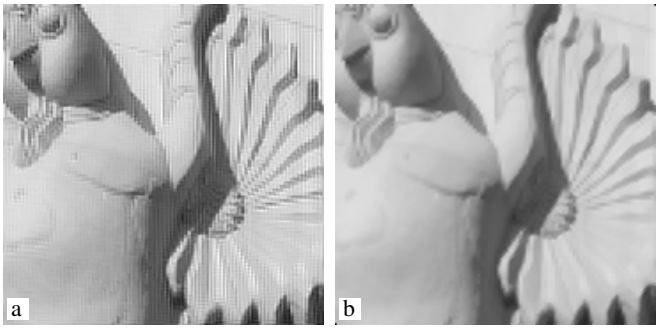


Fig. 3. **(a)** Superresolution image created in aligning the images from Fig. 2(a) and (d) with Whittaker-Tikhonov regularisation, $\gamma = 0.003$. Alignment-MSE measurement with this superresolution image yields a PSNR of 46.05 dB. – **(b)** Same with TV regularisation, $\gamma = 0.03$, yielding a PSNR of 29.92 dB.

A further candidate is total variation

$$T(\nabla v) := \sum_{i,j} |\nabla v| . \quad (18)$$

To find minimisers with this regularisation, one can use, e.g., a gradient descent approach where the regularisation is realised via a locally analytic scheme related to single-scale Haar wavelet shrinkage; we use here a variant of the scheme from [19] adapted to the unequal pixel sizes of v.

As a general rule, in order to just remove the underdeterminedness of the equation system (11), (12), it is desirable to keep the regularisation weight γ rather small.

V. EXPERIMENTS

We evaluate the regularised superresolution alignment procedure from the preceding two sections by the test case from Section II. Starting with Whittaker-Tikhonov regularisation, we observe that for large regularisation weight such as $\gamma = 0.3$ fairly precise estimates for the displacement can be obtained. However, the superresolution images in this case are severely blurred, leading to overestimation of alignment-MSE and low PSNR. For example, the resulting PSNR for the images from Fig. 2(a) and (d) is 28.61 dB. On the other hand, reducing the regularisation parameter to $\gamma = 0.003$ yields extremely high PSNR estimates, e.g. 46.05 dB for the same two images. The reason is that the superresolution images are far away from interpolating between u and g, showing unnatural oscillations, see Fig. 3(a). In contrast, TV regularisation yields plausible results over a wide range of regularisation parameters, see the exemplary superresolution image in Fig. 3(b). For a more detailed evaluation we focus therefore on TV regularisation.

We measure first reconstruction errors for the known exact displacements, see column (a) of Table III. Next we estimate the displacements using the TV-regularised error measure itself, see column (b). Once more the minimisation is done by a grid search with x and y displacements varying from -1 to $+1$ in 0.01 steps. The TV regularisation weight γ is set to 0.03. As the application of the superresolution alignment in this brute-force minimisation is computationally expensive, we add a third scenario, column (c), in which a faster

variant of the superresolution alignment with Whittaker-Tikhonov regularisation and large regularisation parameter $\gamma = 0.3$ is used for the displacement estimation, followed by the actual MSE/PSNR computation with TV regularisation and $\gamma = 0.03$. The latter method gives in a few cases a slightly lower PSNR than the ground-truth displacement, but otherwise approximates the previous scenario well.

It is evident that the variation of PSNR measures is reduced to about half with respect to the measurements with bicubic interpolation, both in terms of the amplitude between maximal and minimal PSNR and the standard deviation. With an amplitude of 1.2 dB it is close to the variation of the shift-free subset of 0.7 dB as reported in Section II.

VI. SUMMARY AND OUTLOOK

In this paper we have studied the reliability of MSE/PSNR measurements for the quality assessment of blind deconvolution results, where the necessity arises to compare images that may be shifted relative to each other by non-integer displacements. An experimental study of simple alignment procedures with bilinear and bicubic interpolation showed that it introduces substantial deviations into the discrepancy measures in question. Comparisons of blind deconvolution methods should therefore not be based on such procedures. As an attempt to overcome this difficulty, we have designed a superresolution-based error measurement procedure that can substantially reduce the variations in MSE/PSNR estimates induced by the alignment step, leaving error margins that are closer to the uncertainty in shift-free cases.

In future work, these tests will have to be extended to more test cases. The applicability of the proposed procedure to other error measures such as MSSIM [17] or perceptual similarity measures [13] will be studied. Further analysis will be devoted to the observed variation of error measures among the shift-free reconstructed images. It will also be of interest to include the PSF into the displacement estimation.

Furthermore, the proposed approach will be used for comparisons between blind deconvolution methods. Taking into account the results from the present contribution and the envisioned more extensive studies will help to assess the significance of method differences in such work.

TABLE III

STATISTICS OF DISPLACEMENT MISESTIMATIONS δx, δy AND PSNR FOR 256 RECONSTRUCTED IMAGES WITH SUPERRESOLUTION-BASED ALIGNMENT WITH TV REGULARISATION, (A) USING EXACT DISPLACEMENTS, (B) ESTIMATING DISPLACEMENTS BY MSE MINIMISATION WITH TV REGULARISATION, (C) ESTIMATING DISPLACEMENTS BY MSE MINIMISATION WITH WHITTAKER-TIKHONOV REGULARISATION.

Setting	(a)	(b)	(c)
$\max\|\delta x\|$		0.09	0.09
std. dev. δx		0.037	0.033
$\max\|\delta y\|$		0.08	0.08
std. dev. δy		0.031	0.036
min PSNR	29.38	29.40	29.40
max PSNR	30.47	30.63	30.46
(max−min) PSNR	1.09	1.23	1.06
mean PSNR	29.93	29.98	29.92
std. dev. PSNR	0.240	0.263	0.236

REFERENCES

[1] M. S. C. Almeida and L. B. Almeida, "Blind and semi-blind deblurring of natural images," *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 36–52, 2010.

[2] L. Bar, N. Sochen, and N. Kiryati, "Variational pairing of image segmentation and blind restoration," in *Computer Vision – ECCV 2004, Part II*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Berlin: Springer, 2004, vol. 3022, pp. 166–177.

[3] T. F. Chan and C. K. Wong, "Total variation blind deconvolution," *IEEE Transactions on Image Processing*, vol. 7, pp. 370–375, 1998.

[4] T. F. Chan, A. M. Yip, and F. E. Park, "Simultaneous total variation image inpainting and blind deconvolution," *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, pp. 92–102, 2005.

[5] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *Proc. SIGGRAPH 2006*, New York, NY, July 2006, pp. 787–794.

[6] V. Katkovnik, D. Paliy, K. Egiazarian, and J. Astola, "Frequency domain blind deconvolution in multiframe imaging using anisotropic spatially-adaptive denoising," in *14th European Signal Processing Conference (EUSIPCO 2006)*. Florence, Italy: EURASIP, 2006.

[7] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *Computer Vision – ECCV 2012, Part VII*, ser. Lecture Notes in Computer Science, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin: Springer, 2012, vol. 7578, pp. 27–40.

[8] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1964–1971.

[9] ——, "Efficient marginal likelihood optimization in blind deconvolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2657–2664.

[10] D. Li, R. M. Mersereau, and S. Simske, "Blind image deconvolution through support vector regression," *IEEE Transactions on Neural Networks*, vol. 18, no. 3, pp. 931–935, 2007.

[11] G. Liu, S. Chang, and Y. Ma, "Blind image deblurring using spectral properties of convolution operators," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5047–5056, 2014.

[12] P. Moser and M. Welk, "Robust blind deconvolution using convolution spectra of images," in *1st OAGM-ARW Joint Workshop: Vision Meets Robotics*, K. Niel, P. M. Roth, and M. Vincze, Eds. Wels, Austria: Österreichische Computer-Gesellschaft, 2016, pp. 69–78.

[13] R. Reisenhofer, S. Bosse, G. Kutyniok, and T. Wiegand, "A Haar wavelet-based perceptual similarity index," arXiv.org, Tech. Rep. cs:1607.06140, 2016.

[14] K. Schelten, S. Nowozin, J. Jancsary, C. Rother, and S. Roth, "Interleaved regression tree field cascades for blind image deconvolution," in *IEEE Winter Conference on Applications of Computer Vision*, 2015, pp. 494–501.

[15] J. Tian and K.-K. Ma, "A survey on super-resolution imaging," *Signal, Image and Video Processing*, vol. 5, pp. 329–342, 2011.

[16] C. R. Vogel and M. E. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Transactions on Image Processing*, vol. 7, pp. 813–824, 1998.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[18] M. Welk, "A robust variational model for positive image deconvolution," *Signal, Image and Video Processing*, vol. 10, no. 2, pp. 369–378, 2016.

[19] M. Welk, J. Weickert, and G. Steidl, "A four-pixel scheme for singular differential equations," in *Scale Space and PDE Methods in Computer Vision*, ser. Lecture Notes in Computer Science, R. Kimmel, N. Sochen, and J. Weickert, Eds. Berlin: Springer, 2005, vol. 3459, pp. 585–597.

[20] L. Xu, S. Zheng, and J. Jia, "Unnatural L_0 sparse representation for natural image deblurring," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1107–1114.

[21] Y.-L. You and M. Kaveh, "Anisotropic blind image restoration," in *Proc. 1996 IEEE International Conference on Image Processing*, vol. 2, Lausanne, Switzerland, Sept. 1996, pp. 461–464.

Generative Adversarial Network based Synthesis for Supervised Medical Image Segmentation*

Thomas Neff[1], Christian Payer[1], Darko Štern[2], Martin Urschler[2]

Abstract— **Modern deep learning methods achieve state-of-the-art results in many computer vision tasks. While these methods perform well when trained on large datasets, deep learning methods suffer from overfitting and lack of generalization given smaller datasets. Especially in medical image analysis, acquisition of both imaging data and corresponding ground-truth annotations (e.g. pixel-wise segmentation masks) as required for supervised tasks, is time consuming and costly, since experts are needed to manually annotate data. In this work we study this problem by proposing a new variant of Generative Adversarial Networks (GANs), which, in addition to synthesized medical images, also generates segmentation masks for the use in supervised medical image analysis applications. We evaluate our approach on a lung segmentation task involving thorax X-ray images, and show that GANs have the potential to be used for synthesizing training data in this specific application.**

I. INTRODUCTION

Modern machine learning methods based on deep neural network architectures require large amounts of training data to achieve the best possible results. For standard computer vision problems, large datasets, such as MNIST [12], CIFAR10 [10], or ImageNet [23], containing millions of images, are publicly available. In the medical field, datasets are typically smaller by several orders of magnitude, as the acquisition process of medical images is costly and time consuming. Furthermore, ethical concerns make it harder to publicly release and share datasets.

Finding methods to improve performance when training deep learning methods on small datasets is an area of active research. Recent work in the medical imaging domain has shown that it is possible to improve performance with small datasets by putting application specific prior knowledge into a deep neural network [17]. Another approach has been made popular by the U-Net [21] architecture for biomedical image segmentation, which demonstrated how strong data augmentation can be used to deal with low amounts of training data in deep network architectures. Even though data augmentation is simple to implement and achieves good results, it is only able to produce fixed variations of any given dataset, requiring the augmentation to fit the given dataset.

Transfer learning approaches such as [19] show that training on large datasets (e.g. ImageNet) followed by fine-tuning on a small dataset achieves state-of-the-art results for datasets consisting of natural images. For medical imaging, the learned features from large natural image datasets may not be suitable, as the image features are very different compared to natural images. Furthermore, there is no straightforward way of transferring 2D features to 3D features, which poses a limitation when working with 3D medical images. Due to the difference in features between medical and natural images, another approach is to use unsupervised feature extractors (e.g. Autoencoders [27]) which are trained on medical images only. Nevertheless, transferring weights learned by these unsupervised methods requires the target network architecture to be close to the source architecture, which is rarely the case.

The requirement for large amounts of training data also popularized image generation methods in deep learning contexts. Recently, research has shown that Generative Adversarial Networks (GANs) [4] can be used for a large variety of applications such as image-to-image translation [6] or unsupervised representation learning [18]. GANs have also been successfully used for unsupervised domain adaptation [8] of multi-modal medical imaging data, demonstrating their potential for use with small medical imaging datasets.

Our goal was to use GANs in a completely different way, by using the high quality of the generated images to augment our small set of training data. We propose a novel modification to GANs, which generates new, synthetic images as well as the corresponding segmentation masks from random noise. This allows us to use the synthetic data as training data for a supervised segmentation task. We show that this architecture manages to produce convincing segmentation masks for the generated images. We evaluate the generated images in two different scenarios on an image segmentation task and show that training on purely generated images achieves results comparable to training on real images for very small datasets.

II. RELATED WORK

A. Training Data Augmentation

Training data augmentation is a commonly used method to reduce the effects of overfitting with small training datasets as well as improve the generalization of the trained network. Most machine learning frameworks allow for simple augmentation such as rotation, translation or intensity shifts of training data. AlexNet [11] was one of the first convolutional neural network (CNN) architectures to implement online data augmentation with successful results. However, data augmentation only achieves good results if the augmentation can actually occur in the data, and is relevant to the required application. For medical imaging, elastic deformations [21]

*This work was supported by the Austrian Science Fund (FWF): P 28078-N33.

[1]Thomas Neff and Christian Payer are with the Institute for Computer Graphics and Vision, Graz University of Technology, Austria thomas.neff@student.tugraz.at

[2]Darko Štern and Martin Urschler are with Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria martin.urschler@cfi.lbg.ac.at

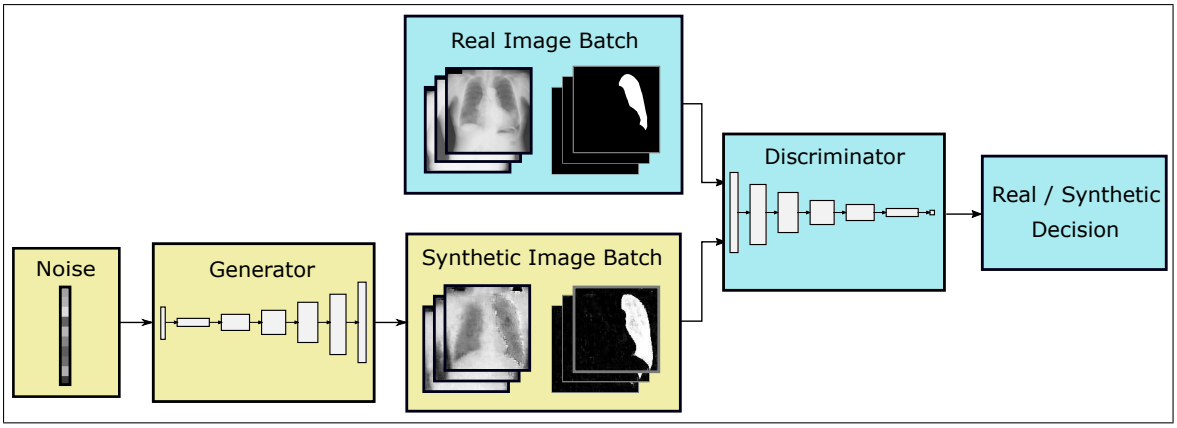


Fig. 1. Proposed GAN architecture incorporating the segmentation mask in the real and synthetic image batches

are especially useful for biomedical segmentation, as they can provide realistic variations of the input data, similar to natural variations.

B. Transfer Learning

Transfer learning aims to improve the learning of a target task in a target domain, given the learned knowledge of a source task in a source domain [16]. Applied to neural networks, it describes the process of training a source network on a source dataset, followed by transferring the learned features to train a different target network on a target dataset [28]. In the context of small datasets, this can be applied in different ways. It is possible to train on a large dataset, e.g. ImageNet, remove the final layer of the network architecture and fine-tune to a smaller target dataset [19]. A different approach is taken by using Autoencoders, which compress a given image to a vector representation and reconstruct the image from this compressed representation. As an example, denoising Autoencoders [27] have been used to extract robust features with great success. However, transferring Autoencoder features typically requires a target network architecture very similar to the source architecture, which is rarely the case.

C. Image Generation

A novel approach to tackle the issue of small datasets for training deep learning methods is to synthesize new training data via image generation methods. Recent research has shown that it is possible to render realistic images using 3D models to alleviate the problem of small datasets [22]. This has the advantage of being able to create an unlimited amount of training data of various scenarios, as long as the images are realistic enough. Rendered images have also recently been used to improve the performance of anatomical landmark detection in medical applications by learning on a dataset of rendered 3D models and fine-tuning on medical data [20]. The disadvantage of using rendered images is that the virtual model and scene parameters need to be explicitly defined and tuned towards the application, which is time consuming.

Generative Adversarial Networks [4] represent a different approach to image generation. A generator and a discriminator network are trained to compete against each other. The goal of the discriminator is to decide if any given image is real or synthetic. The generator generates synthetic images in the hope of fooling the discriminator. Since the generator never directly sees the training data and only receives its gradients from the discriminator decision, GANs are also resistant to overfitting [3]. However, the training process of GANs is very sensitive to changes in hyperparameters. The problem of finding the Nash Equilibrium between the generator and the discriminator generally leads to an unstable training process, but recent architectures such as DCGAN [18] and WassersteinGAN [2] improved on this substantially.

III. METHOD AND ARCHITECTURE

Standard GANs either exclusively learn to generate images [4], or learn to perform image transformations [6]. However, in order to use the generated images for other supervised deep learning tasks, like image segmentation, it is also necessary to have a ground-truth solution for any given input image.

We propose a modification to the standard GAN architecture, which forces the generator to create segmentation masks in addition to the generated images. The discriminator then has to decide whether an observed image-segmentation-pair is real or synthetic. This forces both the discriminator and generator to implicitly learn about the structure of the ground-truth, making the resulting generated data useful for training in a supervised setup. While it is known that using ground-truth labels in the discriminator improves the image quality [24], this is the first time, to our knowledge, that the ground-truth is used to directly generate new image-segmentation-pairs. Fig. 1 illustrates this architecture.

As the foundation for our proposed architecture, we use the DCGAN [18] architecture, which has shown to achieve good results while having increased training stability in many different applications, compared to the previous GAN architectures. DCGAN uses a convolutional generator and discriminator, makes use of batch normalization, and replaces

all pooling layers with convolutions. The generator takes a noise vector z as input and feeds it through multiple fractionally strided convolutions in a fully convolutional manner to generate synthetic images $G(z)$. The discriminator receives both real images x and synthetic images $G(z)$, feeds them through a fully convolutional classification network which classifies any given image as either real, i.e. $D = 1$, or synthetic, i.e. $D = 0$. The discriminator uses the cross entropy loss function

$$l_D = \frac{1}{m}\sum_{i=1}^{m}\left[log\left(D\left(G\left(z^{(i)}\right)\right)\right) + log\left(1 - D\left(x^{(i)}\right)\right)\right], \quad (1)$$

where the mini-batch size m describes the number of training inputs for stochastic gradient descent [15], i denotes the current index in the mini-batch, $x^{(i)}$ is the real image, $z^{(i)}$ is the noise vector sample, D is the discriminator output and G is the generator output. The generator loss

$$l_G = \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \quad (2)$$

only takes the discriminator output of the generated images $D(G(z))$ into account.

By minimizing l_G, the generator is trained to generate images $G(z)$ which look real, i.e. $D(G(z)) \approx 1$, while by minimizing l_D, the discriminator is trained to correctly classify real and synthetic images, i.e. $D(x) \approx 1$ and $D(G(z)) \approx 0$. Therefore, generator and discriminator play against each other, as the generator creates synthetic images which fool the discriminator into believing they are real, while the discriminator attempts to classify real and synthetic images correctly every time.

In order to implement the additional segmentation mask generation, the DCGAN architecture was modified to use 2-channel images, where the first channel corresponds to the image, and the second channel corresponds to the segmentation mask. The discriminator network then simply classifies image-segmentation-pairs instead of images only. The GAN therefore creates synthetic image-segmentation-pairs, which we then further use for the supervised training of a segmentation task. For most GAN setups, this change is simple to implement, as no change in the training process is necessary, making this adaptation very flexible.

IV. EVALUATION

A. Materials

We evaluate our proposed method using a 3-fold cross-validation setup on the SCR Lung Database [26], which is composed of the JSRT Lung Database [25] with corresponding ground-truth segmentation masks. The cross-validation splits are set up so that all 247 images are tested once, using 82 test images, and randomly picking 20 validation images and 145 training images from the remaining images. The images are downscaled to a resolution of 128x128, on which all evaluations are performed. In order to demonstrate possible strengths and limitations of the GAN for even smaller datasets, we evaluate different scenarios on the full dataset, as well as on a reduced dataset. For the reduced dataset, the cross-validation setup for test and validation data is the same as for the full dataset, only the amount of training data is reduced to 30 images by randomly picking them from the training images of the full dataset. For the quantitative evaluation, we chose to perform image segmentation using the U-Net [21] fully convolutional network architecture.

B. Experimental Setup

For our proposed GAN architecture, we adapted the DCGAN [18] TensorFlow [1] implementation tf-dcgan[1]. We modified the architecture to include support for the generation of segmentation masks and increased the image resolution to 128x128. The higher resolution made it necessary to increase the number of generator and discriminator feature maps. We also used a random noise vector z of higher dimension as the generator input. The noise vector dimension was fixed at 400, using uniform noise in the range of $[-1,1]$. Generator feature map sizes were set to $[512,256,128,128,128]$, discriminator feature map sizes were set to $[128,128,256,512,512]$. As suggested in [18], the convolutional kernel sizes were kept at 5. The weights of all convolutional layers were initialized randomly using a normal distribution with zero mean and a standard deviation of 0.05. The input data was scaled to be in the range of $[-1,1]$. The used optimizer was Adam [9] with a learning rate of 0.0004 and an exponential decay rate for the first and second moment estimates of $\beta_1 = 0.5$, $\beta_2 = 0.999$. The training was done using a mini-batch size of 128. The network was trained for 12000 mini-batches in total, as after 12000 mini-batches the overall quality of the generated images $G(z)$ was high for all cross-validation folds. Samples were generated every 200 mini-batches of training. To slightly reduce the impact of Mode Collapse [3], where the generator learns to map several different noise vector inputs z to the same output image $G(z)$, the resulting GAN images were checked for similarity by using a perceptual image hash, which removes images that are almost identical in a batch of samples. Training the GAN took approximately 24 hours per cross-validation fold on an Intel i7-6700HQ CPU @ 2.60 GHz and an NVidia GTX980M GPU with 8 GB of GPU memory.

For the quantitative segmentation results, we used a U-Net architecture of depth 4, replacing max pooling with average pooling for downsampling. This U-Net was implemented using Caffe [7]. Although data augmentation is used to great effect and is also described as a strength of the U-Net [21], we decided not to use it in any of our experiments, in order to specifically evaluate the impact the synthetic GAN samples have on the training process and the resulting segmentation masks. All convolution kernel sizes were set to 3, with feature map sizes of 64 and weights initialized using the MSRA [5] method. We used the Nesterov [14] optimizer at a learning rate of 0.00001 for the segmentation task, with a momentum of 0.99 and a weight decay of 0.0005. The

[1] https://github.com/sugyan/tf-dcgan

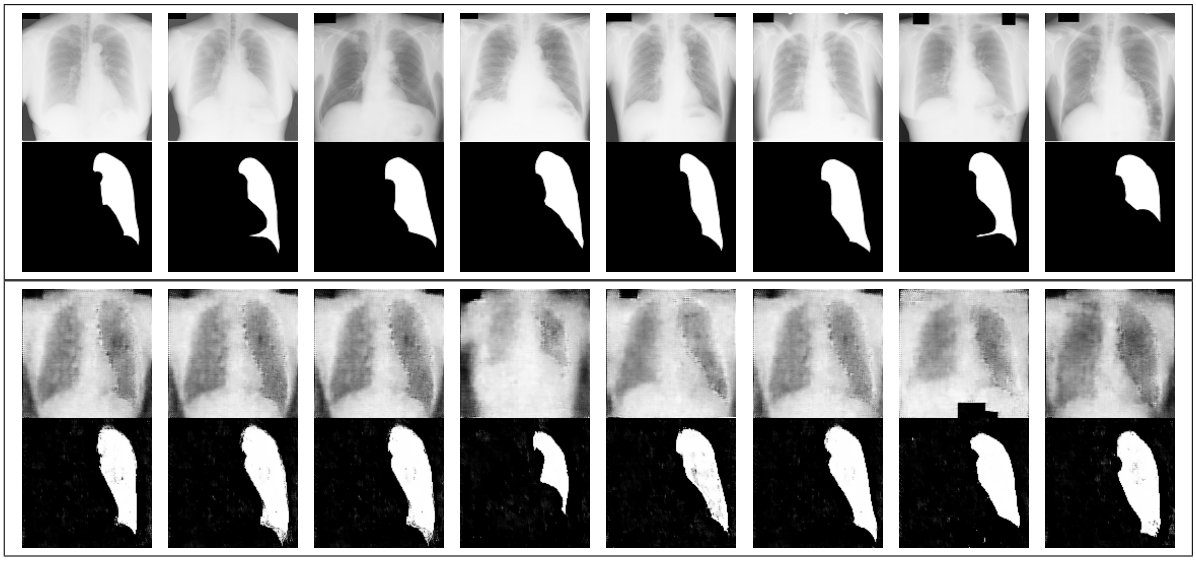


Fig. 2. Sample images and segmentation masks from the real training data (top) compared to synthetic data created by the GAN trained on the full training set (bottom)

mini-batch size was set to 16. The network was trained until the average of the validation error over the last 10 epochs started to increase. The input data was scaled to be in the range of $[-1, 1]$. Since the generated GAN images and segmentation masks are in the value range of $[-1, 1]$, the resulting segmentation image needs to be post-processed to arrive at a binary segmentation mask, which can then be used as an input for the U-Net. To achieve this post-processing, a threshold, largest component and hole filling filter were applied to the generated GAN segmentation masks before they were fed into the U-Net. The threshold was set at the pixel value of 150, and the hole-filling algorithm used is based on geodesic morphology as described in Chapter 6 of [13]. We tested the segmentation performance when using only real training data, a mix of real and synthetic data, as well as only synthetic data. For the synthetic data, we generated a batch of 120 images and segmentation masks from the fully trained GAN. For evaluating the segmentation results, we used the Dice coefficient and Hausdorff distance metrics. Training the U-Net took approximately 3 hours per experiment on the same machine as described above.

C. Results

For the full dataset, Fig. 2 illustrates generated images and segmentation masks from the fully trained GAN, compared to real images and segmentation masks. The quantitative evaluation results for the full dataset can be seen in Table I.

For the reduced dataset, the quantitative evaluation results are shown in Table II.

V. DISCUSSION AND CONCLUSION

Small datasets pose large issues for deep learning methods, leading to overfitting and lack of generalization. We propose an adaptation of Generative Adversarial Networks, where the generator network is trained to generate artificial images in addition to their corresponding segmentation masks. While the qualitative results shown look very promising, they also heavily depend on the amount of training the GAN receives. Fig. 2 shows that using a fully trained GAN to create segmentation data in addition to image data still leads to high quality images. The segmentation also matches the generated images very well, suggesting that both the generator and discriminator are forced to learn the structure of the segmentation as well. However, it can also be seen that small noise artefacts appear in the region of the left lung of the image. These artefacts do not appear if the GAN

TABLE I
QUANTITATIVE RESULTS OF SEGMENTATION USING THE FULL TRAINING SET

U-Net training data		Evaluation metrics			
# Real	# Synthetic	Dice (mean)	Dice (stddev)	Hausdorff (mean)	Hausdorff (stddev)
145	0	**0.9608**	0.0101	**6.1229**	5.0183
145	120	0.9537	0.0121	6.3147	4.8708
0	120	0.9172	0.0283	9.3564	6.0651

TABLE II
QUANTITATIVE RESULTS OF SEGMENTATION USING THE REDUCED TRAINING SET

U-Net training data		Evaluation metrics			
# Real	# Synthetic	Dice (mean)	Dice (stddev)	Hausdorff (mean)	Hausdorff (stddev)
30	0	**0.9464**	0.0158	7.6384	6.0395
30	120	0.9394	0.0133	**7.2885**	5.1007
0	120	0.9312	0.0199	7.6091	5.5654

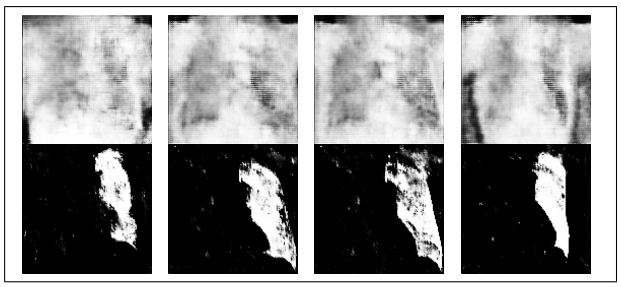


Fig. 3. Sample images and segmentation masks generated by the GAN trained on the full dataset if the training is stopped too early

is trained without generating segmentation masks. We also experience a mild form of Mode Collapse [3], as some of the generated images look very similar. While the images obtained by the fully trained GAN shown in Fig. 2 have a high quality, Fig. 3 illustrates that, if the training time for the GAN is too short, generated images are unusable for later supervised training, as the image quality is too low. Finding a suitable stopping point for GAN training is still a hot topic of current research, as a lower GAN loss during training typically does not indicate higher image quality of the generated images. However, recent modifications to the GAN learning process show that it is possible to correlate the GAN loss with image quality [2], which enables the possibility of stopping the GAN training once the loss is under a certain threshold.

The results of the quantitative evaluation on the full dataset shown in Table I indicate that the GAN images are not sufficient to replace the real images in this case. Using a combination of real and synthetic images to train our segmentation network, the Dice score and Hausdorff distance results are comparable to the results obtained by training on real images only. When only synthetic images obtained by the GAN are used to train the segmentation network, the performance is worse. For the reduced dataset evaluation, the results shown in Table II are not as conclusive. The network with the best Dice score was trained exclusively on real images, while the network with the lowest Hausdorff distance was trained on a combination of real and synthetic images. A very interesting point, however, is that for the reduced dataset, the network trained exclusively on generated GAN images performed almost as well as the network trained on real images, showing significant potential of GANs for training data generation. It is also worth mentioning that the U-Net trained exclusively on generated GAN images from the reduced dataset performed better than the U-Net trained exclusively on generated GAN images from the full dataset. We suspect that this is because the GAN has an easier time to converge to generating high quality images for the reduced dataset compared to the full dataset, leading to better image quality of the generated images.

The quantitative results still have room for improvement. As a further outlook, it would be interesting to incorporate data augmentation in the GAN by using elastic deformations to induce variance in the GAN's training data, which may potentially lead to a greater variety of generated GAN images. Overall, we demonstrated that GANs have significant potential for synthesis of medical training data for supervised tasks by learning to generate segmentation masks in addition to artificial image data.

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A System for Large-scale Machine Learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. Berkeley, CA, USA: USENIX Association, 2016, pp. 265–283. [Online]. Available: http://dl.acm.org/citation.cfm?id=3026877.3026899

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *ArXiv e-prints*, Jan. 2017.

[3] I. Goodfellow, "NIPS 2016 Tutorial: Generative Adversarial Networks," *ArXiv e-prints*, Dec. 2016.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 1026–1034. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.123

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *ArXiv e-prints*, Nov. 2016.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: ACM, 2014, pp. 675–678. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654889

[8] K. Kamnitsas, C. Baumgartner, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, A. Nori, A. Criminisi, D. Rueckert, and B. Glocker, "Unsupervised domain adaptation in brain lesion segmentation with adversarial networks," in *Information Processing in Medical Imaging (IPMI)*, June 2017. [Online]. Available: https://www.microsoft.com/en-us/research/publication/unsupervised-domain-adaptation-brain-lesion-segmentation-adversarial-networks/

[9] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, vol. abs/1412.6980, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[10] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[12] Y. LeCun, C. Cortes, and C. J. Burges, "The MNIST database of handwritten digits," 1998.

[13] A. M. Mharib, A. R. Ramli, S. Mashohor, and R. B. Mahmood, "Survey on liver ct image segmentation methods," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 83–95, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10462-011-9220-3

[14] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O (1/k2)," in *Soviet Mathematics Doklady*, vol. 27, 1983, pp. 372–376.

[15] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015, http://neuralnetworksanddeeplearning.com.

[16] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[17] C. Payer, D. Štern, H. Bischof, and M. Urschler, "Regressing Heatmaps for Multiple Landmark Localization Using CNNs," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 230–238.

[18] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in *International Conference on Learning Representations*, 2016.

[19] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 512–519. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2014.131

[20] G. Riegler, M. Urschler, M. Rüther, H. Bischof, and D. Štern, "Anatomical Landmark Detection in Medical Applications Driven by Synthetic Data," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 85–89.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: http://lmb.informatik.uni-freiburg.de//Publications/2015/RFB15a

[22] A. Rozantsev, V. Lepetit, and P. Fua, "On Rendering Synthetic Images for Training an Object Detector," *Computer Vision and Image Understanding*, vol. 137, pp. 24 – 37, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314214002446

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[24] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf

[25] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K.-i. Komatsu, M. Matsui, H. Fujita, Y. Kodera, and K. Doi, "Development of a Digital Image Database for Chest Radiographs With and Without a Lung Nodule," *American Journal of Roentgenology*, vol. 174, no. 1, pp. 71–74, Jan. 2000. [Online]. Available: http://dx.doi.org/10.2214/ajr.174.1.1740071

[26] B. van Ginneken, M. Stegmann, and M. Loog, "Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database," *Medical Image Analysis*, vol. 10, no. 1, pp. 19–40, 2006.

[27] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390294

[28] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3320–3328. [Online]. Available: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf

Using a U-Shaped Neural Network for minutiae extraction trained from refined, synthetic fingerprints

Thomas Pinetz[1], Daniel Soukup [1], Reinhold Huber-Mörk [1] and Robert Sablatnig[2]

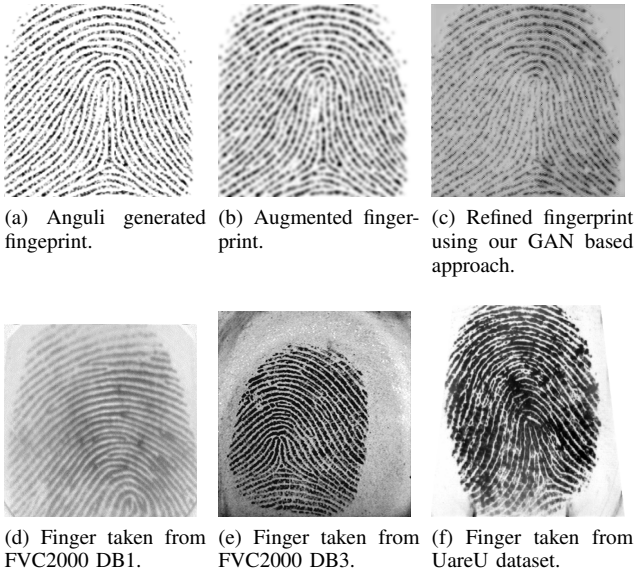
***Abstract*— Minutiae extraction is an important step for robust fingerprint identification. However, existing minutia extraction algorithms rely on time consuming and fragile image enhancement steps in order to work robustly. We propose a new approach, combining enhancement and extraction into a Convolutional Neural Network (CNN). This network is trained from scratch using synthetic fingerprints. To bridge the gap between synthetic and real fingerprints, refinements are used. Here, an approach based on Generative Adversarial Networks (GANs) is used to generate fingerprints suited for training such a network and improving its matching score on real fingerprints.**

I. Introduction

Because of their uniqueness and their temporal stability [10], fingerprint minutiae are a reliable way to determine the identity of an individual. Minutiae points are irregularities in ridge patterns, described using coordinates and orientation [17]. Over 150 different irregularities in fingerprints have been identified [18]. While the amount of minutiae on a single fingerprint varies from finger to finger, there are approximately one hundred of such points comprising a regular fingerprint [17]. It was reported that only 10 - 15 minutiae are required to reliably identify an individual [17].

Currently fingerprint matchers like BOZORTH [25] work using minutiae landmarks. Extraction of minutiae is a hard problem though, which heavily relies on good quality fingerprint images [10]. To combat this, image enhancement algorithms are used [4]. Still, reliable minutiae extraction on arbitrary fingerprint images is an open problem as existing feature extractors largely rely on image quality (focus, resolution, skin condition, etc.) [23].

With the rise of deep learning in similar fields [7], [14], [19] and the availability of synthetic fingerprint generators [2], [5], it looks promising to use such methods for minutiae extraction. This paper contributes a new network for minutiae extraction following the idea to solve an equivalent segmentation problem. In this work the synthetic fingerprint generator Anguli [2] is used because of its availability. Anguli generates the training data needed as is shown in Fig. 1a. Because of the difference to real data as visualized in Fig. 1(d-f), augmentations are used (Fig. 1b) as described in Section IV. Here we contribute a novel technique to refine fingerprints based on the GANs [8] paradigm. An example output can be seen in Fig. 1c. Regularization is used to force the refinement network to retain the annotation data while outputting a refined representation of the simulated fingerprint.



(a) Anguli generated fingeprint. (b) Augmented fingerprint. (c) Refined fingerprint using our GAN based approach.

(d) Finger taken from FVC2000 DB1. (e) Finger taken from FVC2000 DB3. (f) Finger taken from UareU dataset.

Fig. 1. Illustration of the fingerprint data used in this work. (a-c are synthetic fingerprints, while d-f are real fingerprints.)

The rest of the paper is organized as follows. Section II reviews related work. In Section III and IV the minutiae extraction algorithm and the refinement method are described in detail. In Section V the results obtained with our method are presented. Finally in Section VI we draw our conclusions.

II. Related Work

Minutiae detection for a sufficiently enhanced image is done by binarization of the grayscale image [10]. Currently fingerprint minutiae extractors use image enhancement routines to achieve the desired quality [10], [4], [25], [24].

Recently there has been a similar approach to the minutiae extraction problem using a pre-trained Convolutional Neural Network, in a forensic setting [23]. However the CNN in [23] is used as a pre-processing step to find large regions containing a minutiae point. Then logistic regression and region pooling are used to extract the actual minutia position.

In our approach the minutia extraction problem is redefined as a binary segmentation task, which the CNN solves directly. With our method there is no need for any time consuming pre- or post-processing. Additionally, synthetic fingerprint generators are used to train the network from scratch and make it suitable for the minutiae extraction problem.

[1]Austrian Institute of Technology, Donau-City-Straße 1, 1220 Wien {thomas.pinetz.fl, daniel.soukup, reinhold.huber-moerk}@ait.ac.at

[2]TU Wien, Karlsplatz 1, 1040 Wien sab@caa.tuwien.ac.at

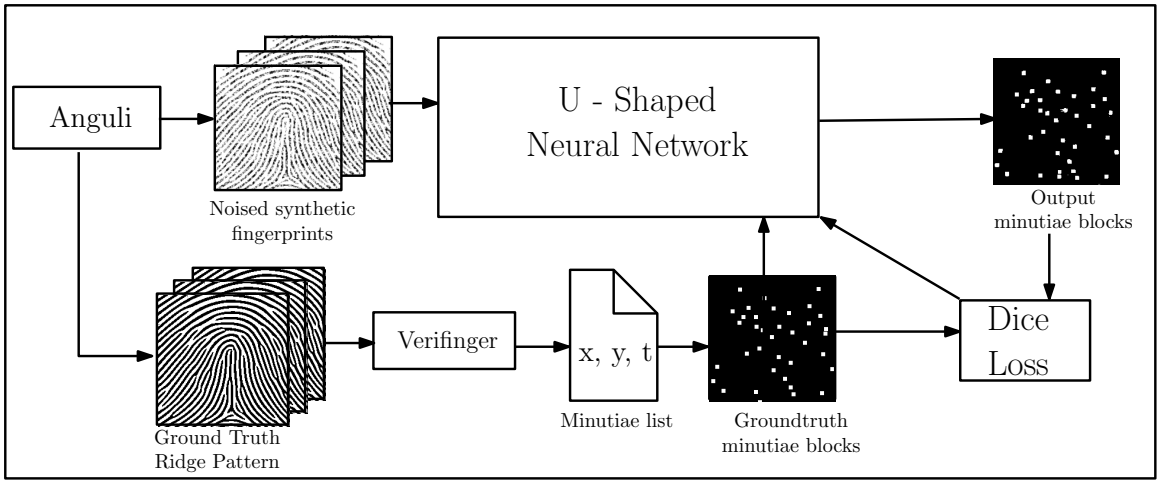


Fig. 2. The whole processing pipeline used for training the minutiae extraction network.

The U-shaped network architecture applied here is used for medical segmentation applications [19], [7]. Training deep neural networks is also the focus of [9], where residual connections are used to allow training of very deep neural networks. Research into making residual connection better is reported in [12], [27], [7].

The GAN framework is first introduced in [8]. Improvements to the stability of adversarial training are proposed in [20], [3]. Based on the results in GANs a refinement network is introduced in [21] for the gaze direction of eye images. In our work a similar approach is used to refine fingerprints.

III. Minutiae Extraction using CNN

The ground truth minutiae list is turned into a binary image by creating an image with the same shape as the corresponding fingerprint and setting every pixel to zero. Then every point in a minutiae region is set to one. A minutia region is defined as a 7×7 pixel square encapsulating the minutia landmark as its centroid. Our deep neural network is used to find a mapping from the input fingerprint to this binary image. This procedure turns the task into a binary segmentation problem.

A. Training Pipeline

The synthetic fingerprint generator Anguli [2] is used to generate a training set. As can be seen in Fig. 2 Verifinger is used to extract the ground truth of the original ridge pattern. For the purpose of this algorithm it is assumed that the minutiae extractor works perfectly on the ridge pattern. Therefore the estimated bifurcations and terminations of the ridge image in Fig. 2 are input to the learning stage as well as ground truth for evaluation. The deviation of the minutiae map and the network output is calculated using dice loss (1), where α is a smoothing factor. Dice loss is reported to produce almost binary outputs [7].

$$loss = -\frac{2y_{pred}y_{true} + \alpha}{\sum y_{pred} + \sum y_{true} + \alpha} \tag{1}$$

B. Network Architecture

The base architecture of the models used in this work can be seen in Fig. 3 and builds on the U-Shaped Network pioneered in [19]. The key differences are:

1) Strided convolution instead of pooling to learn downsampling filters.
2) 224×224 crop to preserve the aspect ratio of the fingerprints.
3) Layer blocks on intermediate levels of the U-Shaped Network instead of pure convolutions.
4) Batch Normalization [11] before every convolution.
5) Dropout with a probability of 0.5 before the final Convolution Layer.
6) Upsampling is done by repeating the pixel in a 2×2 window. Then the upsampling feature maps are concatenated with the output feature maps of the layer block on the same level in the downsampling path. Finally batch normalization, a Regularized Linear Unit (ReLU) activation function and a 3×3 convolution are applied to all the feature maps, before they are passed on to the next layer block.

To preserve information flow, the amount of filters is doubled, when the size of the input data is reduced, as observed in [22]. The layer blocks on specific levels vary in the number of filters used. A model is build with only Wide Residual Blocks [27] (WRN), one with only Densely Connected Blocks [12] (DenseNet) and one with only Bottleneck Residual Blocks [7] (ResUnet). In total, each model used in this work has approximately 8 million parameters.

C. Extracting a Minutiae List

The output of the neural network is a binary minutia regions map. For biometric authentication, a list of minutia points with quality and orientation is needed. For the final position of the minutiae the connected components of the binary map are used. The centroid of each component represents one minutia position. The area a of the connected

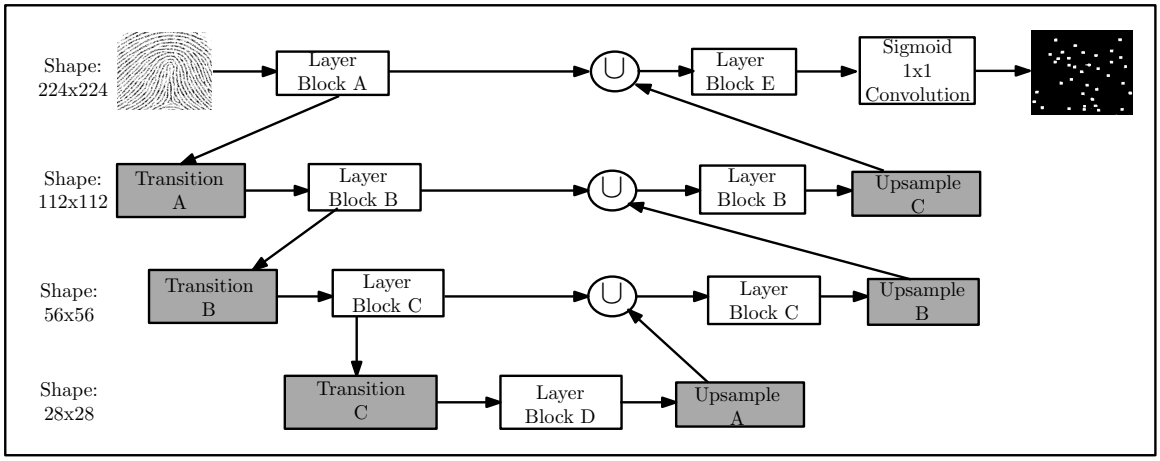
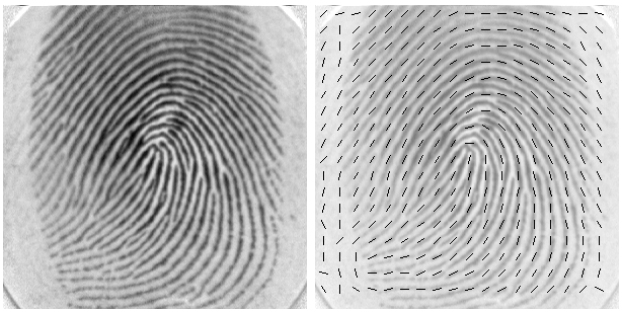


Fig. 3. U-Shaped network architecture used for minutiae extraction.



(a) (b)

Fig. 4. Estimation of the orientation field for a sample fingerprint taken from the FVC2000 DB 1.

(a) (b) (c)

Fig. 5. Anguli [2] generated ridge pattern with two different impressions and the minutiae extracted using Verifinger [24] .

components is used to determine the quality of the minutiae between $0-100$ using $quality = min(a*2,100)$.

The orientation of the minutiae is extracted using an orientation field as described in [10]. The orientation is estimated for every 16×16 region as visualized in Fig. 4b.

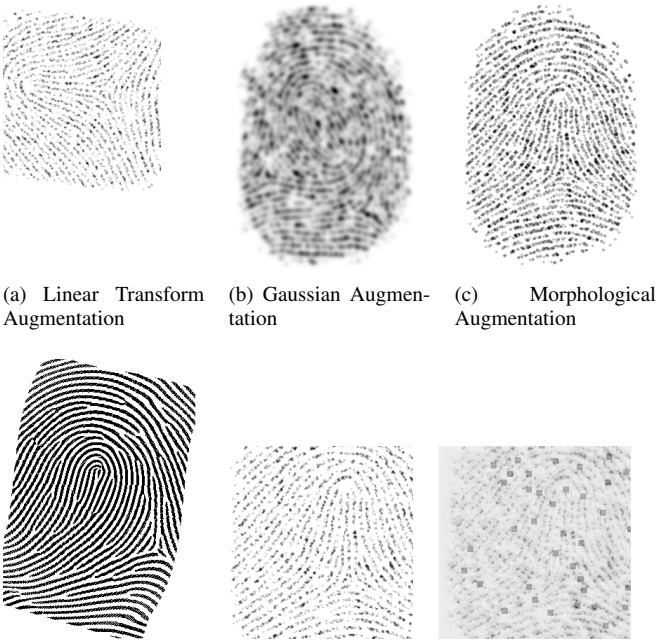
IV. Fingerprint Refinement

The synthetic fingerprint generator Anguli [2] is used to generate random ridge patterns (Fig. 5a). Then multiple variations of every ridge pattern are generated by using different noise models as can be seen in Fig. 5(b,c). Each variation is called an impression of that particular ridge pattern. Because Anguli does not output the minutiae information, a commercial minutiae extractor, Verifinger [24], is used to extract the minutiae data out of the ridge pattern. For the purpose of this paper it is assumed that Verifinger works perfectly on the binary ridge pattern. Those minutiae landmarks are then used for all the impressions (Fig. 5(a-c)).

A. Augmentation on Synthetic Fingerprints

By comparing Fig. 1(d-f) with Fig. 5(a-c) the differences between real and synthetic fingerprints are easily spotted. To bridge this gap the following augmentations are used:

1) **Non linear distortions**: To model the contact region of a fingerprint, random non-linear distortions are used. This also introduces changes in local ridge frequency to synthetic fingerprints as can be seen in Fig. 6d. The distorted ridge pattern is used by Anguli to generate new impressions.
2) **Morphological operations**: Grayscale Dilation and Erosion are used to model wet and dry fingerprint images [5]. An example of this can be seen in Fig. 6c.
3) **Random rotation, translation and shearing**: Fingerprint images are randomly translated, rotated and sheared to gain invariance to linear transformations. An example of this can be seen in Fig. 6a.
4) **Random Blurs**: The images are randomly blurred with a Gaussian kernel, where the variance varies to simulate noisy fingerprints as can be seen in Fig. 6b.
5) **Random Mirroring**: Fingerprint images are randomly mirrored either horizontally or vertically with a 0.5 probability for each direction.
6) **Refinement Network**: A Refinement Neural Network, based on GANs is used to refine images to look more like real world fingerprints. The input size to the network is 224×224. Therefore synthetic fingerprints are resized by a random factor between zero and the difference in image dimension, while keeping the aspect ratio. Then a random 224×224 crop of the



(a) Linear Transform Augmentation (b) Gaussian Augmentation (c) Morphological Augmentation

(d) Distorted Ground Truth Ridge Pattern (e) Refinement Network Input (f) Refinement Network Output with minutiae regions.

Fig. 6. Illustration of the refined data.

resized image is used as input to the network. An example input and output image can be seen in Fig. 6e and Fig. 6f.

B. Refinement Network

The Refinement Network used in this work is based on the GAN paradigm, where a dual optimization problem is solved. A refiner and a discriminator network are simultaneously trained against each other. The refiner network tries to fool the discriminator by applying refinements to a synthetic fingerprint, while the discriminator is used to discriminate between fake and real data. The purpose of such a network is to find a Nash Equilibrium [20] where both networks are optimal.

The only application of a refinement network to our knowledge is in [21]. In our work the approach therein is extended by using noise on the input data to improve the stability of training such a network [3].

One key observation is that using the input image itself for regularization is limiting the amount of possible refinements for fingerprints. Here we propose to use the Hessian of the image instead of the image itself for regularization. The Hessian represents the actual ridge pattern of the fingerprint independent of the pixel intensity values. Mean Squared Error is used to penalize deviation from the Hessian, while the refiner network still needs to fool the discriminator network.

The refiner network uses the same architecture as the minutiae extraction network (Fig. 3), only smaller in the number of layer blocks and filters. Wide residual blocks [27] are used for every layer block starting with 32 filters and doubled on its way down and halved on their way up. Fingerprints like in Fig. 6f are produced by this method. Here, the problem observed by current synthetic fingerprint refiners of modeling noise is addressed by using such a network [5].

V. EXPERIMENTS

This section showcases the results obtained with our method. All our models were programmed using the python framework Keras [6] and trained on a Nvidia Geforce Titan X. For training, the Adam [13] optimizer is used with an initial learning rate of 0.001. The learning rate is cut in half, if the validation error has not decreased for three consecutive epochs. For other minutiae extraction algorithms, an Intel Xeon - W3550 CPU was used.

A. Experimental Setup

For training 28.000 fingerprints with five impressions per fingerprint were generated using Anguli. In total 140.000 fingerprints were used for training, which included a validation set of 10.000 fingerprints. The different impressions can be seen in Fig. 1. Out of the impressions three contain medium noise and the other two use little and heavy noise respectively.

Non linear distortions are used on 3.000 of those fingerprints and on all of their impressions. All the other augmentations, as described in Section IV are applied on the fly.

An annotated real dataset of 300 fingerprints constructed from 220 samples of the sd04 [26] and the 80 images of the fvc2000 DB4_B [15] dataset are used additionally to increase the effectiveness of the classifier. The real dataset used for the refinement network is the UareU [1] dataset.

B. Deep Learning Experiments

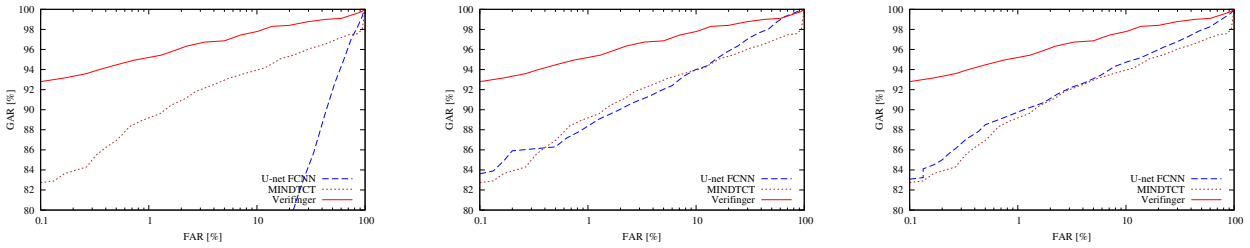
In Fig. 9 the difference in performance for the various layer blocks defined in Section III can be seen. In contrast to the findings in [12] using densely connected blocks did not work as well for the minutiae detection problem. Bottleneck residual blocks performed similarly to wide residual blocks, which is similar to the findings in the original paper [27].

C. Experiments on FVC2000 databases

Here, the performance of our method is compared to other minutiae extraction algorithms on the FVC2000 [15] dataset consisting of real world fingerprints. To match the minutiae against each other, the minutiae matcher BOZORTH [25] was used. The results of this experiment can be seen in Fig. 7 and Fig. 8, where GAR and FAR denote the Genuine Acceptance Rate and False Acceptance Rate accordingly. Using those metrics the Equal Error Rate (EER) can be calculated by finding the rate where (2) holds.

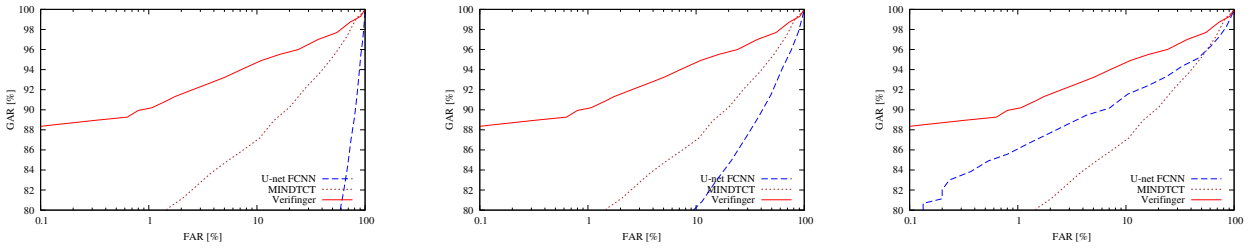
$$GAR = 1 - FAR \tag{2}$$

The extracted EER of the evaluated minutiae extractors is shown in Table I. Our algorithm performs better than



(a) Using only Anguli generated data. (b) Using Augmentations (c) Using Augmentations and Refinements.

Fig. 7. Equal Error Rate Comparison on FVC2000 [15] DB 1 using synthetic, augmentated or refined data.



(a) Using only Anguli generated data. (b) Using Augmentations (c) Using Augmentations and Refinements.

Fig. 8. Equal Error Rate Comparison on FVC2000 [15] DB 3 using synthetic, augmentated or refined data.

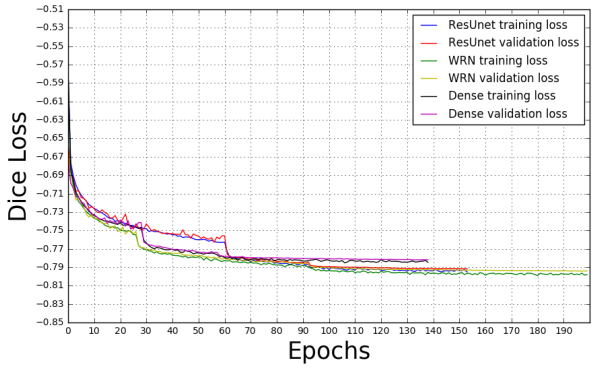


Fig. 9. Model comparision between Dense Blocks, Wide Residual Blocks and Bottleneck Residual Blocks.

MINDTCT on real datasets. Also we report clear performance improvements by using a refinement network. Additionally it is also the fastest method, when run on a GPU.

D. Sample Results for Refinement Network

The only quality metric to our knowledge for GANs is the inception score [20], which is not applicable for our use case. Therefore, this section shows the visual result of the refinement network. In Fig. 10 we can see a comparison of using self regularized MSE versus the Hessian regularized version of the network. In the Hessian regularized examples the ridge pattern is better preserved and less artifacts are introduced into the refined fingerprint.

E. Sample Results for Various Fingers

An illustration on which minutiae are found using different training data is given in Fig. 11. Here, by training solely on

TABLE I

EQUAL ERROR RATE AND ENROLLMENT SPEED FOR FVC2000 [15] DATABASES

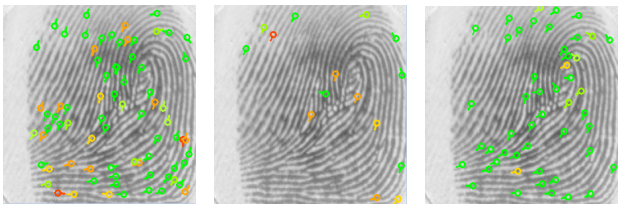
Algorithm	DB 1	DB 3	Time in sec.
Synth. Unet FCNN	21.80%	32.75%	0.12 on gpu
Augm. Unet FCNN	7.01%	16.63%	0.12 on gpu
Ref. Unet FCNN	5.99%	9.42%	0.12 on gpu
MINDTCT [25]	6.63%	12.11%	0.14 on cpu
Verifinger [24]	3.28%	6.31%	1.08 on cpu

(a) Self regularized GAN Output (b) Hessian regularized GAN Output. (c) Hessian regularized GAN Output.

Fig. 10. Sample refiner network output images for self regularized and Hessian regularized training based on the GAN approach.

synthetic fingerprints the minutiae map is clearly wrong as shown in Fig. 11. The network trained on augmented data outputs a subset of the correct minutiae. In contrast, the network trained on GAN data outputs a reasonable minutiae map for this example.

An example of a clear mismatch between two images of the same fingerprint can be seen in Fig. 12. Even though the matching score is 0%, overlapping minutiae are found. However, the orientation does not match because of the noise in the fingerprint.



(a) FCNN trained with synthetic data

(b) FCNN trained with augmented data

(c) FCNN trained on augmented + GAN data.

Fig. 11. Comparison of the output of the same network trained only on synthetic, on augmented and on refined data.

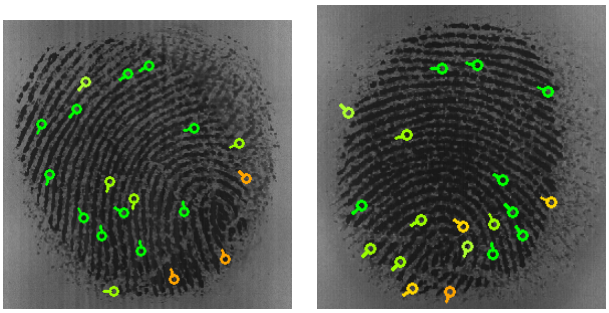


Fig. 12. 0% match of two impressions of the same finger taken from FVC 2002 [16] database with minutiae extracted using the FCNN algorithm.

VI. Conclusion

In this work the possibility of reformulating the fingerprint minutiae extraction problem as a binary segmentation task is shown. Deep learning is used to address this problem. Even with synthetic data as a substitute to annotated real data, the algorithm is able to detect reasonable minutiae with better results than MINDTCT on the FVC2000 dataset without fine tuning of any parameters. Additionally, the performance gain of using our refinement approach was clearly illustrated and advances in training GANs are likely to bring better performance for this minutiae extraction algorithm. A first step is made by using the Hessian instead of the image itself for regularization. However, this performance gain illustrates the dependence on good training data.

Currently, the angle of the minutiae points are calculated using an orientation field. In a future network, we want to learn the orientation of the minutiae by using the orientation field of the ground truth ridge pattern. We believe that better than state-of-the-art performance can be reached using deep learning given sufficiently diverse training data.

ACKNOWLEDGMENT

We thank Peter Wild and Thomas Pock for their helpful insights.

References

[1] "UareU Database," http://www.neurotechnology.com/download.html, 2007, [Online; accessed 01-March-2017].

[2] A. H. Ansari, "Generation and storage of large synthetic fingerprint database," Ph.D. dissertation, Indian Institute of Science Bangalore, 2011.

[3] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, vol. 2016, 2017.

[4] K. Cao, E. Liu, and A. K. Jain, "Segmentation and enhancement of latent fingerprints: A coarse to fine ridgestructure dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 9, pp. 1847–1859, 2014.

[5] R. Cappelli, D. Maio, and D. Maltoni, "Sfinge: an approach to synthetic fingerprint generation," in *International Workshop on Biometric Technologies (BT2004)*, 2004, pp. 147–154.

[6] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[7] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *International Workshop on Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*. Springer, 2016, pp. 179–187.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[10] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777–789, 1998.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[12] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," *arXiv preprint arXiv:1611.09326*, 2016.

[13] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[15] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "Fvc2000: Fingerprint verification competition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 402–412, 2002.

[16] ——, "Fvc2002: Second fingerprint verification competition," in *16th international conference on Pattern Recognition. Proceedings.*, vol. 3. IEEE, 2002, pp. 811–814.

[17] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[18] A. A. Moenssens, *Fingerprint techniques*. Chilton Book Company London, 1971.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[20] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2226–2234.

[21] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," *arXiv preprint arXiv:1612.07828*, 2016.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[23] Y. Tang, F. Gao, and J. Feng, "Latent fingerprint minutia extraction using fully convolutional network," *arXiv preprint arXiv:1609.09850*, 2016.

[24] S. VeriFinger, "Neuro technology (2010)," 2010.

[25] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. Mccabe, S. Janet, and K. Ko, "User's guide to nist biometric image software (nbis)," 2007.

[26] C. I. Watson and C. Wilson, "Nist special database 4," *Fingerprint Database, National Institute of Standards and Technology*, vol. 17, p. 77, 1992.

[27] S. Zagoruyko and N. Komodakis, "Wide residual networks," *CoRR*, vol. abs/1605.07146, 2016. [Online]. Available: http://arxiv.org/abs/1605.07146

Photometric Stereo in Multi-Line Scan Framework under Complex Illumination via Simulation and Learning

Dominik Hirner[1,2], Svorad Štolc[1], Thomas Pock[2]

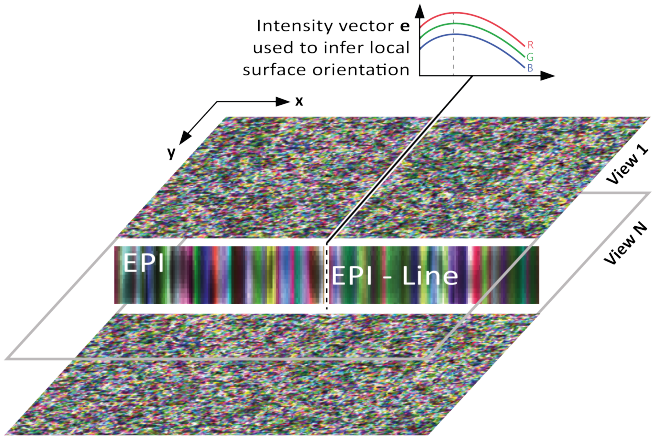


Fig. 1: Visualization of the image stack created by the multi-line scan acquisition. The middle part shows the EPI-lines (here a slice through the image stack). The dashed line represents the read out of one such EPI-line with the respective RGB intensity vector **e**, which is used in order to infer (by training the network) the surface gradient in transport direction ∇x.

***Abstract*— This paper presents a neural network implementation of photometric stereo formulated as a regression task. Photometric stereo estimates the surface normals by measuring the irradiance of any visible given point under different lighting angles. Instead of the traditional setup, where the object has a fixed position and the illumination angles changes around the object, we use two constant light sources. In order to produce different illumination geometries, the object is moved under a multi-line scan camera. In this paper we show an approach where we present a multi-layer perceptron with a number of intensity vectors (i.e. points with constant albedo under different illumination angles) from randomly chosen pixels of six materials with different reflectance properties. We train it to estimate the gradient of the surface normal along the transport direction of the given point. This completely eliminates the need of knowing the light source configuration while still remaining a competitive accuracy even when presented with materials which have non-Lambertian surface properties. Due to the random pooling of the pixels our implementation is also independent from spatial information.**

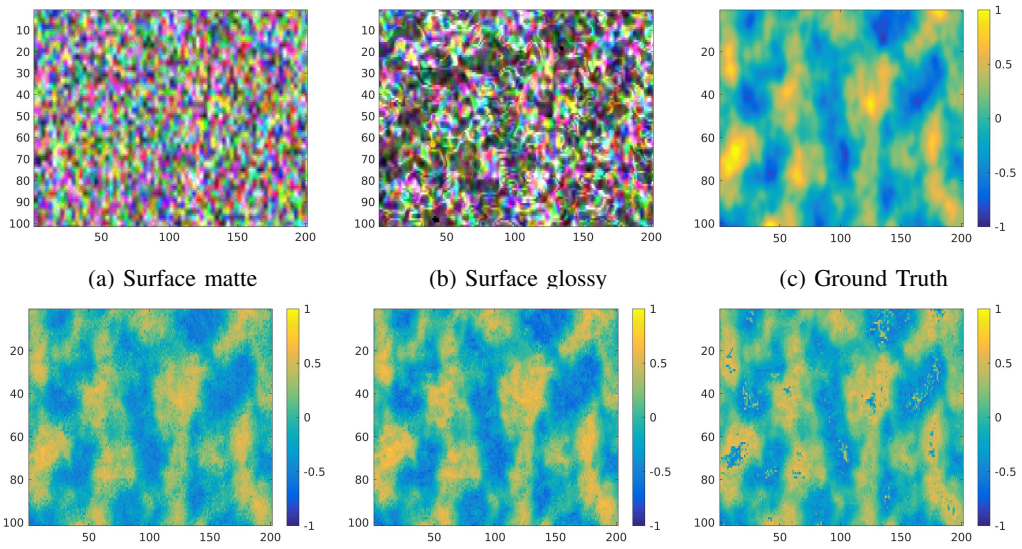
I. INTRODUCTION

The goal of photometric stereo is to estimate the surface normals (and therefore 3D information) of an object using 2D images. This is done by exploiting Lambert's cosine law [1], which states that the intensity of the light at a point is directly proportional to the cosine of its surface normal and the angle of the incident light (see Eq. (1)). By measuring the light intensity of each point under different known and fixed illumination angles the surface normal of each point can be calculated. This approach was first introduced by Woodham in 1980 [2]. However, this equation only holds with the assumption of a Lambertian surface, i.e. a surface that scatters the light in all directions equally. In case of specular reflections the observed intensity of a point also dependents on the position of the observer and therefore the basic approach of photometric stereo does not hold. In the standard photometric stereo approach the orientation and position of the observer (i.e. camera) is known and fixed. Light-field processing via light-field cameras can be seen as an add-on to the general photometric stereo idea. A light-field is a 4-D radiance function written as $L(u,v,s,t)$, where (u,v) denotes the angle, and (s,t) denotes the position of each light ray respectively. To capture a light-field with a camera, a number of different approaches exist, for instance commercially avaliable plenoptic cameras such as the Lytro [3] or by using an array of cameras (multi-camera array) [4]. Using multi-line scan acquisition with a light-field in order to create 2.5/3D surface structure was first introduced in [5]. The same multi-line scan light field camera was used in this approach, which acquires multiple single lines (in our implementation 13) with different viewing angles at one time. Between the active lines on the sensor there are a number of predefined inactive lines (in our implementation 40), so that different viewing angles are produced within one acquisition step without the need of placing several cameras (as e.g. in a multi-camera array).

In our setup an object is placed underneath the camera and is transported in a defined direction over time with two constant light strips placed orthogonal to the transport direction. Between two acquisition steps s_i and s_{i+1} the object has to move the distance equivalent by exactly one pixel. After the acquisition process, the single lines acquired by one such step of each active line on the sensor are concatenated and thus all possible lighting angles and a number of different views are created. This produces a 3D light field structure (two spatial and one directional dimension), instead of the usual 4D structure. This 3D light field can be represented as an image stack that can be seen in Fig. 1. This allows for a fast in-line acquisition suitable for industrial inspection. However, since different lighting responses are dependent on the movement of the object, only inference in the transport

[1]AIT Austrian Institute of Technology GmbH, Vision, Automation & Control, Vienna, Austria {dominik.hirner, svorad.stolc}@ait.ac.at

[2]Graz University of Technology, Institute for Computer Graphics and Vision, Graz, Austria pock@icg.tugraz.at



(a) Surface matte (b) Surface glossy (c) Ground Truth

(d) Prediction Lambertian (e) Prediction Semi-glossy (f) Prediction glossy

Fig. 2: Yellow pixels show positive and blue pixels show negative gradients. Predictions of the surface gradient in transport direction are shown as follows: (a) Surface of the Lambertian material (first view of the 3D light field data), (b) Surface of the glossy material (first view of the 3D light field data), (c) ground truth surface normal gradient in transport direction ∇x used as labels for the regression network, (d) surface normal gradient in transport direction of a Lambertian material learned by the network, (e) surface normal gradient in transport direction of a semi-glossy (gloss = 0.25, roughness = 0.75, see Fig. 4) material learned by the network, (f) surface normal gradient in transport direction of a very glossy material learned by the network. The properties of the different materials can be seen in Fig. 4. One can see that the peaks of the specular lobes (i.e. areas of the biggest negative or positive gradients in (c)) can produce wrong gradient signs.

direction is possible.

The basic method of photometric stereo uses the fact that the observed intensity (or light response) of a given point is dependent on the surface normal orientation as well as the direction of the light, under the assumption of viewing a Lambertian material and a constant albedo. This can be formulated as follows:

$$\mathbf{e} = \mathbf{L} \cdot \mathbf{n} \cdot a \tag{1}$$

where $\mathbf{e} = [e_1 \dots e_n]^T$ is a vector of observed intensities, $\mathbf{L}$ is a matrix describing the light directions and n denotes the surface normal $\mathbf{n} = [n_x, n_y, n_z]^T$. a denotes the albedo which is a scalar value in range $a \in [0,1]$. Inverting this linear equation system yields:

$$\mathbf{n} \cdot a = \mathbf{L}^+ \cdot \mathbf{e} \tag{2}$$

Solving this over-determined least squares problem produces an estimation of the surface normal (Note: L^+ is the Pseudo-Inverse of the light direction matrix using, e.g. the Moore-Penrose method [6]). Instead of solving Eq. (1) directly we use a multi-layer perceptron in order to learn a mapping between the intensity responses ($e_R = [e_{R1} \dots e_{R13}]^T, e_G = [e_{G1} \dots e_{G13}]^T, e_B = [e_{B1} \dots e_{B13}]^T$) in each pixel to the gradient of the surface normal in transport direction $\nabla x = {}^{an_x}/_{an_y}$.

Some results of the learned mapping are visualized in Fig. 2. The figure depicts the same small area in all six images. The first two images are examples of how the surfaces that where used for inference from two different material types (matte and glossy) looks like. For both images the first view (i.e. the first illumination angle) of the 3D light field structure was taken. The remaining images show a color-coded visualization of the surface normal gradient ∇x.

The intensity vectors e_R, e_G and e_B correspond to the observed intensity values of the different illumination angles (here referred to as views) for each channel of the RGB pixel value respectively. These three vectors (e_R, e_G and e_B) are then stacked vertically for each pixel in order to create the data samples for the network, which then has the form $E = [e_{R1} \dots e_{R13}, e_{G1} \dots e_{G13}, e_{B1} \dots e_{B13}]^T$, where $E \in \mathbb{R}^{39}$ (three color-channels a 13 illumination angles). These data-points of all six datasets are then randomly shuffled in order to avoid a spatial bias due to, e.g. non-constant lighting before presenting it to the network. Since the cumulative number of all points from all datasets is very large (around 3 million samples), a batch based training approach with a batch size of 1000 was used rather then an online learning approach. We used the TensorFlow library [7] for the implementation of the network as well as for the cost function and optimizer.

II. Related Work

3D reconstruction using 2D images has been a well studied problem in the field of computer vision. Over the years many different methods to solve this problem arose. In [8] range scanning with stripe patterns were combined with

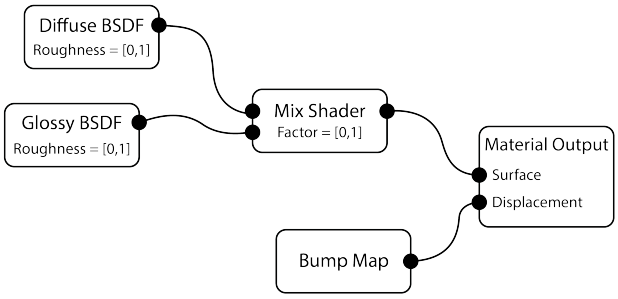


Fig. 3: Schematic illustration of the Blender Node Setup for creating different materials.

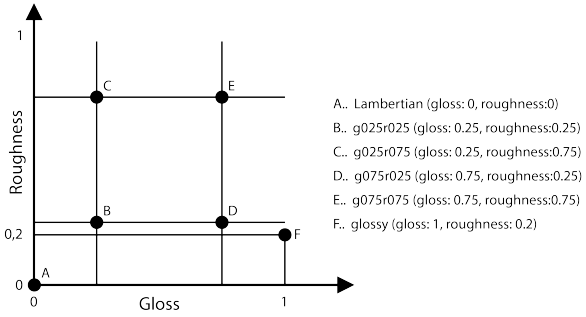


Fig. 4: Visualization of the six different datasets created by changing the roughness and the percentage of which the glossy or diffuse node is taken.

photometric stereo with five light sources in order to recover the 3D surface of an object. Using epipolar plane image (EPI) structures from motion analysis for depth reconstruction was introduced in [9]. The paper by Tao deals with incorporating a shading term to depth from defocus with correspondence cues in order to refine the shape estimation [10]. In [11] Hayakawa used a singular-value decomposition (SVD) of a formulated matrix in order to get a surface normal estimation without the need of a-priori knowledge of the light source direction under the Lambertian assumption. Some machine learning approaches have been explored, such as [12] where a multi-layered neural network was used in order to learn the mapping between image intensities and the surface normal orientation, using a Gaussian sphere with average reflectance as the training data. In [13] Cheng used a symmetrical 6-layer neural network to train a mapping between the vectorized image and a reflectance value for each pixel. Another machine learning approach has been investigated in [14], where a neural network was used in order to solve the shape from shading problem, previously introduced by [15].

III. Experiments

A. Generating Ground Truth Data

Blender 2.78 [16] Cycles Renderer was used to generate the ground truth data. This artificial ground truth data has some advantages over real-world acquisition, such as the ease of modification of the setup, feasibility of generating many images quickly as well as being less prone to errors. However, in order to make the resulting images more realistic, some artefacts, such as jitter or salt&pepper noise, can be taken into consideration. The goal while creating the ground truth data was to cover as much ground as possible with the synthetic data regarding the task. The network should learn a mapping between the RGB intensity vectors of the different views and surface properties, to the surface normal gradient. As it is infeasible to cover all possible mappings between color, light reflectance and surface normals, a random approach was chosen. A uniformly distributed, 8-bit random color pattern was created (each RGB color channel uniformly distributed between 0-255) and used as a texture. The blender-intern noise texture and displacement map node was used in order to create a random surface normal structure on a flat surface. With this approach the possible mapping space is sparsely covered. Furthermore we created six different material datasets with different gloss values using a mixture of the Diffuse BSDF and Glossy BSDF node shaders. In this model two parameters can be changed, namely the gloss factor (controlled by the mix node) and the roughness of the two BSDF nodes. For the sake of simplicity, the roughness is the same for both, the Diffuse and the Glossy BSDF node. A schematic illustration of this setup can be seen in Fig. 4. This model is based on a presentation from Gastaldo [17], where he states:

$$\mathscr{R} + \mathscr{T} + \mathscr{A} = 1 \tag{3}$$

where $\mathscr{R}$ denotes reflectivity, $\mathscr{T}$ denotes transparency and $\mathscr{A}$ denotes absorption. Furthermore he states that reflectivity can be divided into diffuse reflectivity ($\mathscr{R}_d$) and specular reflectivity ($\mathscr{R}_s$). With this he derives:

$$\mathscr{R}_d + \mathscr{R}_s + \mathscr{T} \leq 1 \tag{4}$$

In our setup $\mathscr{R}_d$ correlates to the Diffuse BSDF node and $\mathscr{R}_s$ to the Glossy BSDF node. Transparency was not taken into consideration (i.e. is always 0) as we exclude glass like materials from our data. The roughness parameter of the Diffuse BSDF node corresponds with the roughness of the Oren-Nayar reflectance model [18]. The model used for the glossy factor of the material was GGX [19]. The roughness parameter of the GGX model simulates microscopic bumps in the surface, so that the reflections of the material look blurrier the higher the roughness parameter is.

We excluded a glossy dataset with a roughness value of 0, which would imitate a mirror like behavior. However,

a material with a roughness of 0.2 already shows highly specular behavior.

The multi-line scan camera setup as described in Sec. I was recreated in Blender, where a plane with a random color texture and a bump map (see Fig. 3) was moved underneath the camera. During each animation step the plane was moved by exactly one pixel. The resulting images were concatenated and reshaped in order to create a 3D image stack representation of the light field. Each image plane is then shifted to the left in the following manner:

$$\forall x,y,i : I'_i(x,y) = I_i(x-40i,y) \tag{5}$$

where $i \in [0\dots12]$ denotes the index of the image in the 3D light field structure, $I_i \in \{width \times height \times 3\}$ is the spatial image domain of the i'th view and I'_i denotes the new translated image. Since the disparity (i.e. the gap between active lines on the camera sensor) is 40 pixels it was used as the shifting constant. The resulting overlap (at most 12×40 in the last view) is then cropped. This is done so that the EPI-lines are vertical with no slope, as they would be with an object with true 3D geometries.

B. Network Parameters Evaluation

For the optimal performance of a neural network some parameter evaluation and tuning, such as changing the number of hidden neurons, using different activation functions or cost functions, is needed. In our evaluation, we looked at 3 different activation functions, namely linear, Sigmoid and rectified linear unit (RELU). The input layer, which consists of 39 neurons is fully connected with the hidden layer. We tried different numbers of neurons in the hidden layer for each evaluated activation function respectively. The results of these experiments can be seen in Table I. For the read-out of the output layer, which consists of one neuron since we only regress the gradient in the transport direction, a linear activation function was used.

Given the problem we want to solve and our material properties, one can expect that a low number of hidden neurons will suffice and already give a good performance, as the Lambertian reflectance function is low dimensional. The low dimensionality of a Lambertian reflectance has been proven and explored in [20]. Having a less complex network architecture can be beneficial for both the runtime as well as the generalization of the network. Here 1, 3, 10 and 20 neurons of one hidden layer were used.

The cost function used to measure the quality of the regressed prediction (therefore also the value used for the optimization of the network) was the mean square error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{6}$$

For optimization the batch based gradient descent algorithm with a learning rate of $\eta = 0.001$ was used. The dataset was split into 80% training set and 20% testing set, as proposed by the Pareto Principle by J.M. Juran [21]. The network was trained for 100 epochs.

Table I shows that the Sigmoid has both the best overall, as well as the best performance in a single run with 20 neurons in the hidden layer. As the aforementioned experiments were performed only to show the overall tendency and convergence of the network structure, a small learning rate η was used for all the experiments. However, [22] shows that exploring this parameter further is important for the overall network accuracy. For this task we found that a learning rate of $\eta = 0.2$ works best which improved the overall accuracy of the network to $MSE_{train} = 0.020464$ and $MSE_{test} = 0.02052$ when trained for 100 epochs.

TABLE I: Training and testing MSE with different numbers of neurons and activation functions.

Training set MSE					
# hidden neurons	1	3	10	20	avg
act. fct.					
linear	0.05903	0.05988	0.05760	0.05857	0.05877
Sigmoid	0.05429	0.05285	0.05263	**0.04792**	**0.05192**
RELU	0.05605	0.05543	0.05283	0.05150	0.05395
Testing set MSE					
linear	0.05902	0.05972	0.05777	0.05855	0.05877
Sigmoid	0.05402	0.05276	0.05266	**0.04768**	**0.05178**
RELU	0.05608	0.05571	0.05312	0.05147	0.054095

C. Network Performance

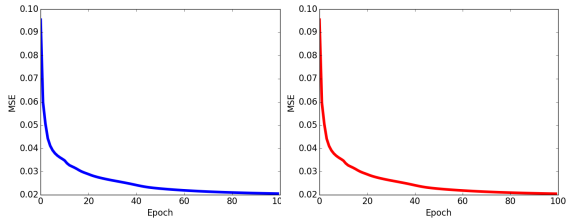


Fig. 5: Evolution of the mean square error over epochs with a learning rate of $\eta = 0.2$. Left: Training set (80% randomly chosen from all sets), right: Testing set (20% randomly chosen from all sets).

Fig. 5 shows the convergence of the overall accuracy on the training and test set, combining and shuffling all six created datasets. This was done in order to generalize the network as much as possible regarding the material type (matte, semi-glossy or glossy). Once the network was learned it was applied to each material type individually and the accuracy of the prediction on the whole set was reported. For simplicity we use acronyms for each created dataset, as shown in Fig. 3. For the sake of simplicity we took the liberty of reporting the error on the whole dataset (data points used for training and testing combined). As the errors on the training and on the testing set are very close together and there is no sign of overfitting the network, this liberty can be taken without distorting the results. The best performance was achieved on the semi-glossy datasets. The larger error on the glossy dataset is due to the fact that the sign of the surface normal is sometimes predicted wrong if the specular lobe is narrow and outside of the observed range. This can

TABLE II: MSE of each individual whole dataset applied to the network. On the left we report the accuracy of our neural network, then the accuracy of the Lambertian model when 80% of the Lambertian dataset was used to estimate the illumination matrix **L** from Eq. (1) as an analogy of learning (L.m.L stands for Lambertian model Lambertian datasets). In the last column 80% of all datasets were used (Lambertian model all dataset).

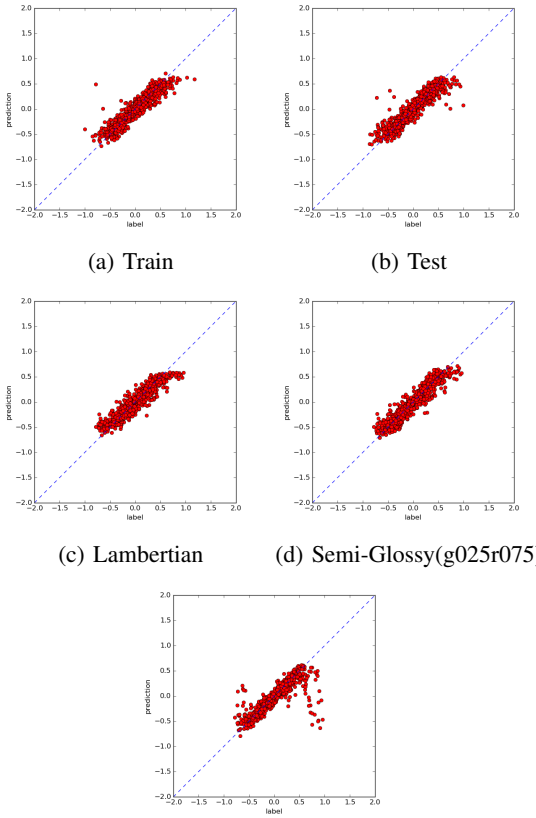
Dataset	$MSE_{network}$	$MSE_{L.m.L}$	$MSE_{L.m.a}$
Lambertian	0.01637	0.02435	0.02804
g025r025	0.01537	0.05550	0.03268
g025r075	0.01835	0.02063	0.02741
g075r025	0.01760	0.24619	0.10237
g075r075	0.01795	0.03233	0.02930
glossy	0.03722	0.89302	0.37912
avg.	0.02047	0.21200	0.09982

also be seen in the correlation plots in Fig. (6) where some of the outliers from the glossy dataset also show up in the correlation plot for the whole train and test dataset.

We compare our results with the model-based Lambertian approach by solving Eq. (1) for **L** as an analogy of learning with the same dataset training/testing split as for our machine learning approach. For this the assumption of an constant albedo with a value of 1 was taken. Despite it can be argued that the Lambertian model only works for Lambertian materials. The quantitative results are reported in Table II. It can be seen that the L.m.L. approach completely failed for the glossier material datasets. On the other hand the L.m.a. approach proved to perform in average about twice as good improving significantly especially on the glossy cases. Last but not least, we show that our neural network approach outperforms the traditional photometric stereo by far for the given task, especially for glossier material.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we showed a neural network based machine learning approach in order to learn a mapping between intensity vectors (i.e different illumination angles) of points with different reflectance properties to a surface normal gradient. We showed that in our approach we do not need to know the position and direction of the light source as well as no spatial information and were still able to produce competitive accuracy. The proposed machine learning approach outperformed the standard photometric stereo based on the Lambertian model by 5-10 times. We tested the network on synthetically generated data and showed that our implementation works well even for very glossy surface properties. In our simulations the train error converges very fast which suggests that we did not yet reach the absolute best accuracy possible and increasing the number of features as well as training the network for longer may still increase the overall prediction of the multilayer perceptron. The mean absolute error (MAE) can be advantageous as it is more robust against outliers [23], however since we excluded strong outliers manually in our datasets beforehand we did not need to use MAE. Nevertheless, exploring this cost function in the future should be done.



(a) Train (b) Test

(c) Lambertian (d) Semi-Glossy(g025r075)

(e) Glossy

Fig. 6: (a-e) Show the correlation plot between label and prediction of ∇x for the respective datasets of 100 samples uniformly taken from the set. (a) combines 80% of all datasets (which were randomly chosen). (b) combines 20% of all datasets (which were randomly chosen). (e) shows some outliers where the sign of the gradient was wrongly predicted due to the high specular response. The stronger outliers on (a) and (b) also come from this set.

For future work we intend to extend this approach to perform material classification (e.g. classify matte, glossy, semi-glossy material etc.) as well as learning the albedo of the created datasets. In this paper we only used synthetic data in order to prove the correctness of the method, however an evaluation on real-world data for the trained networks would be the next step. Additionally, we want to investigate the possibilities of inference on the surface normal gradient orthogonal to the transport direction.

REFERENCES

[1] J.H. Lambert. *Photometria sive De mensura et gradibus luminis, colorum et umbrae*. Sumptibus viduae Eberhardi Klett, typis Christophori Petri Detleffsen, 1760.

[2] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):191139–191139–, 1980.

[3] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light Field Photography with a Hand-Held Plenoptic Camera. Technical report, April 2005.

[4] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3):765–776, July 2005.

[5] Svorad Štolc, Reinhold Huber-Mörk, Branislav Holländer, and Daniel Soukup. Depth and all-in-focus images obtained by multi-line-scan light-field approach. In *IS&T/SPIE Electronic Imaging*, pages 902407–902407. International Society for Optics and Photonics, 2014.

[6] Jonathan S Golan. Moore–penrose pseudoinverses. In *The Linear Algebra a Beginning Graduate Student Ought to Know*, pages 441–452. Springer, 2012.

[7] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[8] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. In *ACM transactions on graphics (TOG)*, volume 24, pages 536–543. ACM, 2005.

[9] Robert C Bolles, H Harlyn Baker, and David H Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[10] Michael W Tao, Pratul P Srinivasan, Jitendra Malik, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1940–1948, 2015.

[11] Hideki Hayakawa. Photometric stereo under a light source with arbitrary motion. *JOSA A*, 11(11):3079–3089, 1994.

[12] KV Rajaram, Guturu Parthasarathy, and MA Faruqi. A neural network approach to photometric stereo inversion of real-world reflectance maps for extracting 3-d shapes of objects. *IEEE transactions on systems, man, and cybernetics*, 25(9):1289–1300, 1995.

[13] Wen-Chang Cheng. Neural-network-based photometric stereo for 3d surface reconstruction. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 404–410. IEEE, 2006.

[14] Siu-Yeung Cho and Tommy WS Chow. Shape recovery from shading by a new neural-based reflectance model. *IEEE Transactions on Neural Networks*, 10(6):1536–1541, 1999.

[15] Berthold KP Horn and Michael J Brooks. *Shape from shading*. MIT press, 1989.

[16] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam,

[17] Franois Gastaldo. How to make your own physically correct shading. Blender Converence, 2012.

[18] Michael Oren and Shree K Nayar. Generalization of lambert's reflectance model. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246. ACM, 1994.

[19] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 195–206, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[20] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, Feb 2003.

[21] F John Reh. Pareto's principle-the 80-20 rule. *BUSINESS CREDIT-NEW YORK THEN COLUMBIA MD-*, 107(7):76, 2005.

[22] Igiri Chinwe Peace, Anyama Oscar Uzoma, and Silas Abasiama Ita. Effect of learning rate on artificial neural network in machine learning. In *International Journal of Engineering Research and Technology*, volume 4. IJERT, 2015.

[23] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

3D Localization in Urban Environments from Single Images*

Anil Armagan[1], Martin Hirzer[1], Peter M. Roth[1] and Vincent Lepetit[1,2] .

***Abstract*— In this paper, we tackle the problem of geo-localization in urban environments overcoming the limitations in terms of accuracy of sensors like GPS, compass and accelerometer. For that purpose, we adopt recent findings in image segmentation and machine learning and combine them with the valuable information given by 2.5D maps of buildings. In particular, we first extract the façades of buildings and their edges and use this information to estimate the orientation and location that best align an input image to a 3D rendering of the given 2.5D map. As this step builds on a learned semantic segmentation procedure, rich training data is required. Thus, we also discuss how the required training data can be efficiently generated via a 3D tracking system.**

I. INTRODUCTION

Accurate geo-localization of images is a very active area in Computer Vision, as it can potentially be used for applications such as autonomous driving and Augmented Reality. As the typically available GPS and compass information are often not accurate enough for such applications, we recently proposed a method that builds only on untextured 2.5D maps [3]. In general, 2.5D maps hold the 2D information about the environment, more precisely the buildings' outlines and their heights. However, this approach is limited in practice, as it heavily relies on the often unreliable and error prone extraction of straight line segments to find the re-projections of the corners of the buildings.

To overcome this limitation, as shown in Fig. 1, we replace this step by semantic segmentation (i.e., [4] and [5]) to extract the visible façades and their edges, which is described in more detail in Sec. II. Since learning the necessary model requires a large amount of training data, as detailed in Sec. III, we use a 3D tracking algorithm to semi-automatically label the huge amount of required training images. In order to estimate the correct pose, we introduce two strategies. The first strategy samples random poses around the initial pose given by the sensors and selects the best one. The second strategy builds on a more advanced search algorithm by using CNNs to iteratively update the pose. Both approaches are discussed in Sec. IV.

II. SEMANTIC SEGMENTATION

Given a color input image I, we train a fully convolutional network (FCN) [5] to perform a semantic segmentation. FCN applies a series of convolutional and pooling layers to the input image, followed by deconvolution layers to produce a segmentation map of the whole image at the original resolution. In our case, we aim at segmenting the façades and the edges at building corners or between different façades. Everything else is referred to as "background". We therefore consider four classes: façade, vertical edges, horizontal edges and background. We use a stage-wise training procedure, where we start with a coarse network (FCN-32s) initialized from VGG-16 [6], fine-tune it on our data, and then use the thus generated model to initialize the weights of a more fine-grained network (FCN-16s). This process is repeated in order to compute the final segmentation network having an 8 pixels prediction stride (FCN-8s).

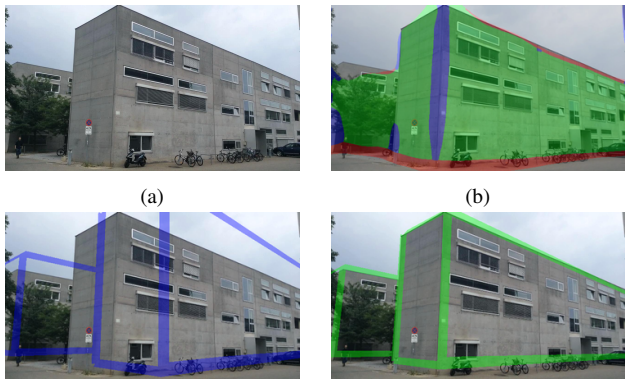


Fig. 1. Overview of our approach: Given an input image (a), we segment the façades and their edges (b). We can either sample poses around the pose provided by the sensors or use CNNs to move the camera starting from the sensor pose (c), and keep the pose that aligns the 2.5D map and the segmentation best (d).

III. ACQUISITION OF TRAINING DATA

Deep-learning segmentation methods require a large number of training images to generalize well, however, manual annotation is costly. We therefore use a 3D tracking system [3] to easily annotate frames of video sequences. First, we create simple 3D models from the 2.5D maps. Then, for each sequence, we initialize the pose for the first frame manually, and the tracker estimates the poses for the remaining frames. This allows us to label façades and their edges very efficiently. More precisely, we recorded 95 short video sequences using a mobile device. In order to ensure an accurate labeling, in particular for the edges, we only keep frames in which the re-projection of the 3D model is well aligned with the real image, and remove those frames that suffer from tracking errors or drift.

* This work was funded by the Christian Doppler Laboratory for Semantic 3D Computer Vision.

[1] Institute of Computer Graphics and Vision, Graz University of Technology, Graz, Austria {armagan, hirzer, pmroth, lepetit}@icg.tugraz.at

[2] Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux, Bordeaux, France

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

Fig. 2. Iteratively pose refinement from an initial sensor estimate: (a) Test image with overlaid ground truth pose, (b) initial noisy sensor pose, (c) segmented image, (d) finally pose obtained with our method.

IV. 3D Localization

Building on the same segmentation approach trained using the training data as described in Secs. II and III, we proposed two different approaches for pose estimation.

A. Direct Pose Selection [1]

Given a coarse initial estimate $\tilde{\mathbf{p}}$ of the pose provided by the sensors and a 2.5D map of its surrounding, the goal is to finally estimate the correct pose $\hat{\mathbf{p}}$. Therefore, we sample poses in a regular grid around $\tilde{\mathbf{p}}$ and estimate

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} \mathscr{L}(\mathbf{p}), \tag{1}$$

where $\mathscr{L}(\mathbf{p})$ is the log-likelihood

$$\mathscr{L}(\mathbf{p}) = \sum_{\mathbf{x}} \log P_{c(\mathbf{p},\mathbf{x})}(\mathbf{x}). \tag{2}$$

The sum runs over all image locations $\mathbf{x}$, where $c(\mathbf{p},\mathbf{x})$ is the class at location $\mathbf{x}$ when rendering the model under pose $\mathbf{p}$, and $P_c(\mathbf{x})$ is the probability for class c at location $\mathbf{x}$ where P_c is one of the probability maps predicted by the semantic segmentation.

B. CNN-based Refinement [2]

As this brute-force strategy is not very efficient, we additionally proposed a CNN-based approach for iterative pose refinement. To refine the location, we discretize the directions along the ground plane into 8 possible directions and train a network to predict the best direction to refine the currently estimated location. We also add a class that indicates that the estimated location is already correct and should not be changed. Thus, given the semantic segmentation of the current input image and a rendering of the 2.5D map from the current pose estimate, the network, denoted by CNN_t, yields a 9-dimensional output vector:

$$\mathbf{d}_t = \mathrm{CNN}_t(R_\mathrm{F}, R_\mathrm{HE}, R_\mathrm{VE}, R_\mathrm{BG}, S_\mathrm{F}, S_\mathrm{HE}, S_\mathrm{VE}, S_\mathrm{BG}), \tag{3}$$

Here, S_F, S_HE, S_VE, and S_BG denote the probability maps computed by the semantic segmentation for the classes façade, horizontal edge, vertical edge and background, respectively; R_F, R_HE, R_VE, R_BG are binary maps for the same classes, created by rendering the 2.5D map for the current pose estimate.

In addition, we train a second network to refine the orientations:

$$\mathbf{d}_o = \mathrm{CNN}_o(R_\mathrm{F}, R_\mathrm{HE}, R_\mathrm{VE}, R_\mathrm{BG}, S_\mathrm{F}, S_\mathrm{HE}, S_\mathrm{VE}, S_\mathrm{BG}), \tag{4}$$

where $\mathbf{d}_o$ is a 3-dimensional vector, covering the probabilities to rotate the camera to the right, to the left or not rotate it at all.

Starting from the initial estimate $\tilde{\mathbf{p}}$, we iteratively apply CNN_t and CNN_o and update the current pose. These steps are iterated until both networks are converged and predict not to move. In particular, there are two main advantages of having two networks: (a) As the networks for translation and orientation are treated separately, we do not need to balance between them. (b) The two detached problems are much easier to solve, reducing both, the training and the inference effort.

V. Results and Summary

Two illustrative results obtained by the approach described in Sec.IV-B are shown in Fig. 2. It clearly can be seen that the initial sensor poses (Fig. 2(c)) does not cover the groundtruth (Fig. 2(a)) very well, whereas the finally estimated poses (Fig. 2(c)) using the segmentation results (Fig. 2(b)) perfectly fit the buildings. Overall, this demonstrates that adopting ideas from semantic segmentation in combination with convolutional neural networks and the information provided by 2.5D maps can successfully be used for estimating the poses of buildings and thus their exact location. For more details, we would like to refer to [1] and [2].

References

[1] A. Armagan, M. Hirzer, and V. Lepetit, "Semantic Segmentation for 3D Localization in Urban Environments," in *JURSE*, 2017, best Paper Award.

[2] A. Armagan, M. Hirzer, P. M. Roth, and V. Lepetit, "Learning to Align Semantic Segmentation and 2.5D Maps for Geolocalization," in *CVPR*, 2017.

[3] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit, "Instant Outdoor Localization and SLAM Initialization from 2.5D Maps," in *ISMAR*, 2015, best Paper Award.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *CoRR*, 2015.

[5] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *CVPR*, 2015.

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, 2014.

Depth-guided Disocclusion Inpainting for Novel View Synthesis*

Thomas Rittler[1,2], Matej Nezveda[1,2], Florian Seitner[2], and Margrit Gelautz[1]

Abstract— **The generation of novel views is a crucial processing step in 3D content generation, since it gives control over the amount of depth impression on (auto-)stereoscopic devices and enables free-viewpoint video viewing. A critical problem in novel view generation is the occurrence of disocclusions caused by a change in the viewing direction. Thus, areas in the novel views may become visible that were either covered by foreground objects or were located outside the borders in the original views. In this paper, we propose a depth-guided inpainting approach which relies on efficient patch matching to complete disocclusions along foreground objects and close to the image borders. Our method adapts its patch sizes depending on the disocclusion sizes and incorporates the depth information by focusing on the background scene content for patch selection. A subjective evaluation based on a user study demonstrates the effectiveness of the proposed approach in terms of quality of the 3D viewing experience.**

I. INTRODUCTION

The generation of novel views from an existing single view and its corresponding depth map is a crucial processing step for 3D content generation and processing. Such newly generated views enable the users to watch 3D content on different types of 3D displays, including multi-user autostereoscopic devices with a comfortable range of viewing perspectives, and navigate in 3D space for free-viewpoint video applications. The 2D input image and its associated depth map – known as 2D-plus-depth [11] – can be delivered by a variety of sources such as depth sensors based on time-of-flight or structured light (e.g., Microsoft's Kinect), stereo cameras, or 2D-to-3D conversion techniques.

A principal problem in novel view generation is the occurrence of disocclusions due to a change in the viewing direction. Some areas in the original views that were either covered by a foreground object or were located outside the image borders may become visible in the novel views. To deal with these disocclusions, one common approach is to pre-process the depth maps. In particular, filtering techniques are applied to the associated depth maps prior to the novel view generation [16]. Although this approach can reduce the appearance of disocclusions, it can also lead to spatial distortions in the scene geometry of the novel views.

Another approach is to use image inpainting techniques to fill in the disoccluded areas in the novel views with suitable estimates derived from the visible scene content. However, traditional inpainting algorithms (e.g., [5]) do not take into account additional knowledge provided by the depth data. For that reason, several inpainting strategies have been proposed that incorporate depth information during disocclusion filling [6], [8], [10], [13], [1], [15], [14]. While most related work aims at rendering photorealistic views, suitable inpainting approaches may also be required in the context of non-photorealistic rendering [9]. A few depth-induced inpainting strategies build upon PatchMatch (PM) [2], which is a randomized search algorithm that quickly finds correspondences between disjoint image patches. For example, He et al. [10] add the depth information to the PM algorithm by restricting the validity of patches used for inpainting. However, as their method was initially proposed for foreground object removal, the authors rely on a-priori depth information in the region to be filled which is not available when considering disocclusions. Morse et al. [13] extend PM from single image completion to stereo image pairs by not only incorporating depth information extracted from the stereo pairs but also allowing the matching of patches across the stereo pairs. However, the additional original view of a stereo image pair is not available in a 2D-plus-depth setup as considered in this work. Additionally, none of the aforementioned depth-guided inpainting approaches considers subjective quality assessment in the evaluation of their results. However, the results of Bosc et al. [3] indicate the need of subjective quality assessment in terms of novel views evaluation, as commonly used 2D quality metrics do not reflect the subjective quality of novel views. A very recent publication [4] gives an in-depth evaluation using the Middlebury ground truth data set, but does not incorporate user studies.

In this paper, we propose a depth-guided inpainting approach for disocclusion filling in novel views based on PM. Our approach incorporates the supplementary depth information to favor background patches during the disocclusion inpainting and uses adaptive patch sizes for efficient hole filling. We perform a paired comparison user study to evaluate our inpainting results in the context of stereoscopic viewing and present experimental results that show that our depth-guided inpainting approach yields better subjective quality compared to several earlier approaches.

The rest of the paper is organized as follows: Section 2 describes the proposed inpainting method. Section 3 provides details on our experimental setup. Section 4 presents the results of the user study along with some inpainting examples, and Section 5 concludes the paper.

*This work was supported by the Technology Agency of the City of Vienna (ZIT) under the project PAINT3D and finalized under the project Precise3D, funded by the Austrian Research Promotion Agency (FFG) and the Austrian Ministry BMVIT under the program ICT of the Future.

[2] emotion3D GmbH, Gartengasse 21/3, 1050 Vienna, Austria; {nezveda, seitner}@emotion3d.tv

[1] Institute of Software Technology and Interactive Systems, Vienna University of Technology, Favoritenstrasse 9-11/188-2, 1040 Vienna, Austria; margrit.gelautz@tuwien.ac.at

II. Proposed Approach

We suggest an inpainting technique that builds upon PM as an efficient strategy for finding patch correspondences based on color differences. The proposed approach incorporates adaptive patch sizes and search space restrictions based on depth information, as explained in the following subsections. First, the formalism of the general inpainting problem is recapped [5]: Let I be an input image and $\Omega \subseteq I$ a "hole" to be filled, called the *target* region. That is, Ω denotes all the missing pixels within I. Additionally, the *source* region Φ provides samples used in the infilling process. The goal is now to complete the missing region Ω with data from Φ so that the resulting image will be visually coherent. While conventionally $\Phi = I \setminus \Omega$, we restrict Φ to candidates from the image background as part of our approach.

A. Adaptive patch sizes

As opposed to iterative inpainting approaches that shrink the holes by successively copying patches of constant size, we perform the inpainting step only once at the end of the image completion chain, with the goal to avoid propagating erroneous inpainting results from one iteration step to the next. Our non-iterative approach is enabled by the usage of adaptive patch sizes. If fixed-size patches are used and the patch size is smaller than the size of Ω, there are some target patches containing no valid image information (see blue rectangle in Fig. 1a) that is required to compute the patch similarities.

For that purpose, a threshold τ_1 is specified to ensure a minimum percentage of valid pixels in each target patch. The corresponding patch size for each target pixel is determined by successively incrementing the patch dimensions until the percentage of the valid source pixels exceeds τ_1. Hence, the selected patches are smaller near the borders and are growing as the patch's central pixel is moving towards the hole's centroid, as illustrated in Fig. 1b. As a side effect, fewer patches are involved in the color synthesis of an individual pixel (based on weighted color averaging of overlapping patches) near the boundaries of Ω, which helps avoid blurring artifacts in these regions.

By introducing adaptive patch sizes it is guaranteed that the majority of the target patches contain a certain percentage of valid pixels. However, there may arise situations where the combination of target and source patches becomes impractical, as schematically illustrated in Fig. 1c. Hence, a second threshold τ_2 (equal to or smaller than τ_1) is specified to maintain the majority of valid pixels in the matching step and to ensure a minimal overlap between valid pixels of the target patch and the corresponding source patch.

B. Depth

There are two major reasons for disocclusions that cause blank areas in novel views: (a) areas that had been covered by a foreground object in the original view, and (b) areas along the image borders that had been outside the field of view in the original image. While scene depth is not taken into account when dealing with case (b), it is reasonable to fill

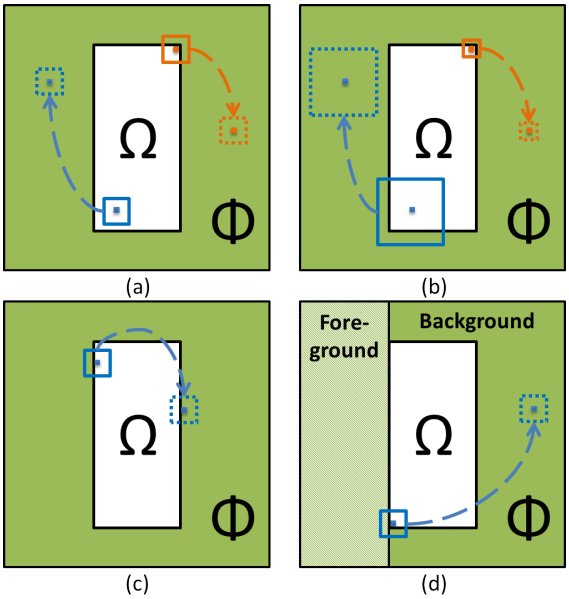


Fig. 1. Schematical overview of the basic concepts of our inpainting approach: (a) constant versus (b) adaptive patch size; (c) problem of non-overlapping valid pixels between target and source patch; (d) target patch comprising foreground and background pixels. Further details are given in the text.

occlusions of group (a) with image data obtained from background regions. As these holes emerge due to sharp depth transitions (i.e., depth discontinuities) at object boundaries, a target patch may comprise pixels that belong to foreground objects as well as pixels that are part of the background, as illustrated in Fig. 1d. Consequently, inpainting artifacts occur – hereinafter also referred to as *foreground color blur* – which are caused by color bleeding from the foreground. Therefore, depth information is incorporated in the matching stage to find appropriate patch correspondences and prevent foreground regions from being used for filling disoccluded regions.

Since depth information is not available in the target region, the depth values have to be synthesized first from the warped depth values in the surrounding. For every hole in Ω, each scanline is first filled by a constant value determined as the maximum depth value of the left and right pixel located at the hole boundary. Then, the minimum of the newly filled in depth values is selected as a lower bound of permissible depth levels in the nearest-neighbor search for target patches of the respective hole. An additional outlier removal based on the statistics of the depth histogram is applied to make the procedure more robust to depth map inaccuracies.

III. Experimental Setup

In order to investigate the effectiveness of our proposed inpainting algorithm on the perceived quality of stereoscopic images, a pair-wise comparison study was conducted. The stereo pairs used for evaluation were formed by the original left views and novel right views, i.e., synthesized views derived from the left views and the corresponding depth maps with disocclusions filled by inpainting. This section

TABLE I

THIS TABLE LISTS THE NAME, NUMBER (PERCENTAGE) OF DISOCCLUDED PIXELS AND THE CHARACTERISTICS OF THE IMAGES USED IN THE SUBJECTIVE STUDY.

Name	Disocclusions	Characteristics
Arm	54050 (2.6%)	low-textured background
Bird	29790 (1.4%)	moderately textured background
Crowd	57711 (2.7%)	cluttered repetitive background
Edge	51173 (2.5%)	highly textured background
Flower	50483 (2.4%)	repetitive background

describes the test material, the inpainting techniques used for comparison and the selected subjective methodology including a description of the test environment and subjects.

A. Dataset

All inpainting methods are evaluated on footage from a movie sequence. Five still images – termed as *Arm*, *Bird*, *Crowd*, *Edge* and *Flower* – have been chosen as test images, with a resolution of 1920×1080 pixels. The selected images cover different image characteristics including varying densities of background texture and diverse amounts of disoccluded pixels, as summarized in Table I.

B. Algorithms

We compare our depth-guided PM inpainting approach (DPM), which was described in Section 2, against our implementation of PM [2] with constant patch sizes of 51×51 pixels, the image completion function content-aware fill (CAF) of Adobe's Photoshop CS5[3], which does not use depth information, and horizontal background replication (HBR) [7]. We use the following, same constant parameter settings to generate the results: $\{\tau_1, \tau_2\} = \{10\%, 10\%\}$. The thresholds have been chosen to provide a small but reasonable amount of valid pixels to be used for patch matching while preventing target patches from becoming too large, which would lead to blurrier inpainting results and increase the overall runtime of the algorithm.

C. Subjective assessment procedure

The Pair Comparison (PC) method has been chosen to quantify the subjective ratings [12]. In the PC method, a pair of stimuli is compared and the subjects are asked to rate the quality of the stimuli in terms of preferences using a ternary scale (i.e., stimulus A is preferred, stimulus B is preferred, or stimuli A and B are equally preferred).

Particularly, using 4 inpainting approaches and 5 images, a total number of 30 pair comparisons had to be performed by each subject. Each pair was presented successively in random order. The subjects were allowed to switch interactively between the two stimuli of a pair. Moreover, each subject performed a trial run in which the test methodology was introduced.

We compute the quality score for each method by increasing its respective counter by 1 in case of a preference and 0.5 in case of an equal valuation. The accumulated value is then divided by the number of comparisons per method and by the total number of participants. Hence, the final score shows the percentage of comparisons "won", e.g., a value of 100 indicates that this method has always been preferred over any other approach.

[3]http://www.adobe.com/technology/projects/content-aware-fill.html

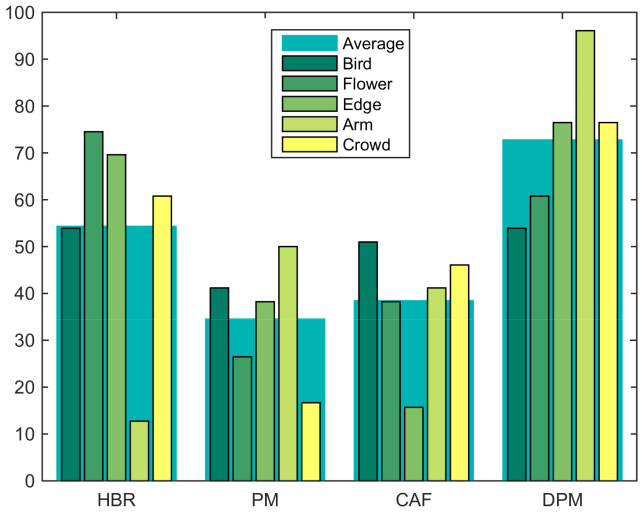


Fig. 2. Pair comparison scores of the subjective study.

The test sequences were displayed on a 23.6″ stereoscopic display (i.e., Acer GD245HQ) with a native resolution of 1920×1080 pixels and the NVIDIA 3D vision controller. To provide an ideal test setup, the room was darkened to avoid external visual disturbances and the viewing distance was set to one and a half times the screen size.

Seventeen non-expert observers (six female and eleven male observers aged between 17 and 49) participated in the study. All of the subjects were screened for visual acuity, color vision and stereo vision according to ITU-R BT.1438 recommendation [12].

IV. Results and Discussion

In Fig. 2, the PC scores obtained for the five test images are presented, grouped by the evaluated inpainting methods. Our proposed approach DPM performs best and is preferred on average in 72.75% of all comparisons. In contrast, the other PatchMatch-based inpainting methods PM and CAF attain significantly lower average PC scores of 34.51% and 38.43%, respectively.

Fig. 3 offers a closer look at some examples of inpainted regions. Regarding our approach, the study participants remarked a clear delineation of the foreground objects. A possible explanation is the reduction of artifacts caused by foreground color blur, which are mainly perceived as unnatural shadows of the objects (cf. DPM and PM in the second and third row of Fig. 3). Additionally, it can be seen that for holes at the image border, it is possible to inpaint coherent information by using adaptive instead of fixed-size patches.

The lower score of our approach (60.78%) compared to HBR (74.51%) for the image *Flower* may be caused by significant inaccuracies of the corresponding depth map. In

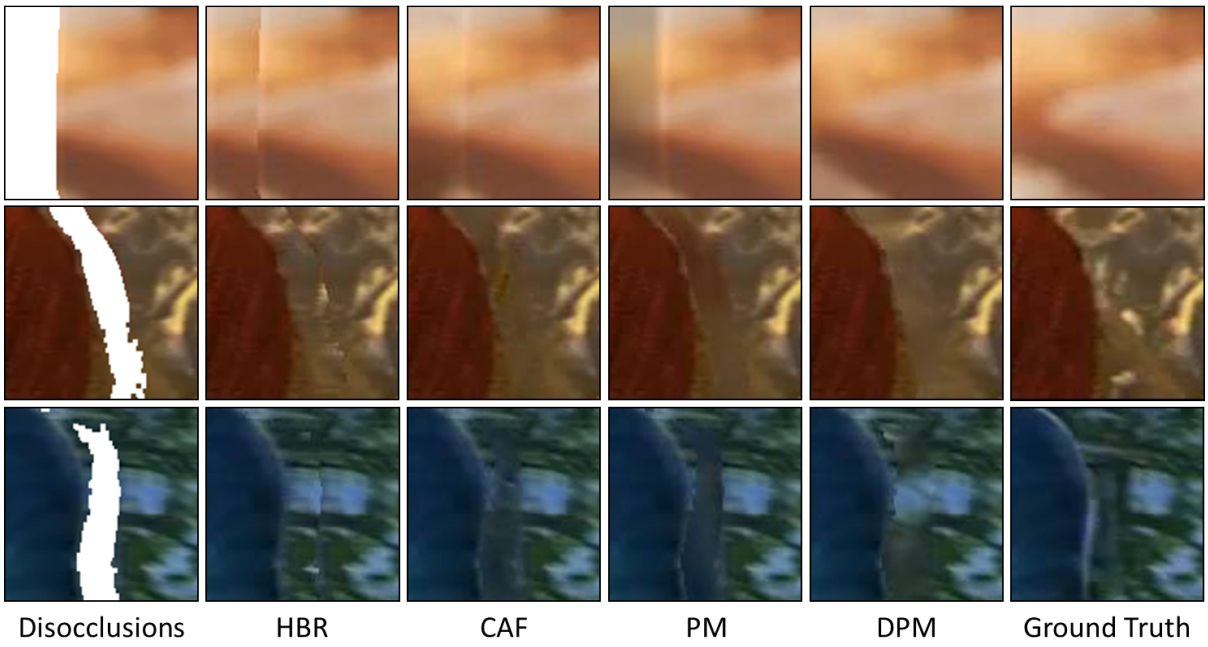


Fig. 3. Visual comparison of inpainting results. The first row shows a snippet including a hole at the border of image *Flower*. The second and third row show snippets including holes caused by depth discontinuities for images *Crowd* and *Edge*, respectively. Best viewed in color.

particular, parts of the background area have been erroneously labeled as foreground and thus are not taken into account in the patch matching step according to the predefined depth constraints. Consequently, artifacts are present in the inpainted region, which however could be avoided by adjusting the depth-based outlier removal.

Another interesting finding is the approximately uniform distribution of PC scores among the investigated inpainting methods for the image *Bird*. The observers declared that they found it hard to detect any differences, which might be due to the fact that *Bird* exhibits the smallest number of disoccluded pixels (see Table I). Additionally, these disoccluded pixels are located in primarily low textured areas outside the main focus of the observer's attention. Similarly, the better result of the relatively straightforward inpainting method HBR (54.31% on average) compared to PM (34.51% on average) and CAF (38.43% on average) may lie in the fact that in our test images the inconsistencies caused by HBR inpainting become mainly noticeable in highly textured background regions near the image margin, whereas observers tend to pay more attention to the central image area covered by the foreground object.

V. Conclusion

We have introduced a depth-guided inpainting approach that addresses the filling of disocclusions in novel views. Our method is based on efficient patch matching and produces visually very satisfying results for both disocclusions at image borders and disocclusions along the boundaries of foreground objects. Our method adapts its patch sizes to the disocclusion sizes. For disocclusions along objects, we additionally incorporate the depth information by focusing on the background scene content for patch selection. A subjective evaluation of the stereoscopically perceived quality of the synthesized novel views showed the effectiveness of our proposed approach. For future work, we plan to extend our technique to disocclusion inpainting of video sources.

References

[1] I. Ahn and C. Kim, "Depth-based disocclusion filling for virtual view synthesis," in *IEEE International Conference on Multimedia and Expo*, 7 2012, pp. 109–114.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, pp. 24:1–24:11, 2009.

[3] E. Bosc, R. Pepion, P. L. Callet, M. Pressigout, and L. Morin, "Reliability of 2D quality assessment methods for synthesized views evaluation in stereoscopic viewing conditions," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video*, 2012, pp. 1–4.

[4] P. Buyssens, O. Le Meur, M. Daisy, D. Tschumperlé, and O. Lézoray, "Depth-guided disocclusion inpainting of synthesized RGB-D images," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 525–538, 2017.

[5] A. Criminisi, P. Perez, and K. Kentaro, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[6] I. Daribo and B. Pesquet-Popescu, "Depth-aided image inpainting for novel view synthesis," in *IEEE International Workshop on Multimedia Signal Processing*, 2010, pp. 167–170.

[7] M. Eisenbarth, F. Seitner, and M. Gelautz, "Quality analysis of virtual views on stereoscopic video content," in *19th International Conference on Systems, Signals and Image Processing*, 2012, pp. 333–336.

[8] J. Gautier, L. M. Josselin, and C. Guillemot, "Depth-based image completion for view synthesis," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*, 2011, pp. 1–4.

[9] M. Gelautz, E. Stavrakis, and M. Bleyer, "Stereo-based image and video analysis for multimedia applications," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Systems (XXth ISPRS Congress)*, 2004, pp. 998–1003.

[10] L. He, M. Bleyer, and M. Gelautz, "Object removal by depth-guided inpainting," in *Austrian Association for Pattern Recognition Workshop*, vol. 2, 2011, pp. 1–8.

[11] ISO/IEC 23002-3, "Information technology – MPEG video technologies – Part 3: Representation of auxiliary video and supplemental information," 2007.
[12] ITU-R Recommendation BT.1438, "Subjective assessment of stereoscopic television pictures," 2000.
[13] B. Morse, J. Howard, S. Cohen, and B. Price, "Patchmatch-based content completion of stereo image pairs," in *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2012, pp. 555–562.
[14] S. M. Muddala, R. Olsson, and M. Sjöström, "Spatio-temporal consistent depth-image-based rendering using layered depth image and inpainting," *EURASIP Journal on Image and Video Processing*, vol. 2016, no. 1, pp. 1–19, 2016.
[15] S. M. Muddala, M. Sjostrom, and R. Olsson, "Depth-based inpainting for disocclusion filling," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video*, 2014, pp. 1–4.
[16] M. Nezveda, N. Brosch, F. Seitner, and M. Gelautz, "Depth map post-processing for depth-image-based rendering: A user study," in *IS&T/SPIE Electronic Imaging*, 2014, pp. 90 110K–90 110K.

Line Processes for Highly Accurate Geometric Camera Calibration

Manfred Klopschitz, Niko Benjamin Huber, Gerald Lodron and Gerhard Paar

***Abstract*— The availability of highly accurate geometric camera calibration is an implicit assumption for many 3D computer vision algorithms. Single-camera applications like structure from motion or rigid multi-camera systems that use stereo matching algorithms depend on calibration accuracy. We present an approach that has proven to deliver accurate geometric information in a reliable, repeatable manner for many industrial applications. The major limitation in typical camera calibration methods is the printing accuracy of the used target. We address this problem by modeling the calibration target uncertainty as a line process and incorporate a lifted cost function into a bundle adjustment formulation. The regularized target deformation is incorporated directly into the non-linear least-squares estimation and is solved in a non-iterative, principled framework.**

I. INTRODUCTION

Geometric camera calibration defines the mapping between points in world coordinates and their corresponding image locations. These parameters model imperfections of the camera optics, i.e. lens distortion, intrinsic parameters of the idealized pinhole camera and extrinsic parameters like absolute camera orientation and relative orientation for multi-camera setups. Most calibration methods assume known $3D$ world points and minimize a reprojection error of the known $3D$ structure into detected image correspondences. The resulting error is a result of model imperfections, target imperfections and feature point localization inaccuracies.

Impressive reprojection errors have been shown in [5] by estimating feature points and $3D$ structure in an iterative procedure. We argue, like [2], [4], that the most important aspect for many applications is printing accuracy, but present a non-iterative calibration formulation that estimates and corrects for target uncertainty within a single bundle adjustment minimization.

The geometric camera calibration process estimates the mapping between points in world coordinates and their corresponding image locations. We define the image projection using standard notation, for the pinhole model

$$\mathbf{x}_p = KR[I|-\tilde{\mathbf{C}}]\mathbf{X} = P\mathbf{X} \quad \Big| \quad K = \begin{bmatrix} f & & c_x \\ & f & c_y \\ & & 1 \end{bmatrix}$$

R and $\tilde{\mathbf{C}}$ model the location of the camera in space and K defines the intrinsics. Lens distortion is added to the pinhole projection, for example using this popular model:

$$\mathbf{x}_d = \mathbf{x}_p + \mathscr{F}_D(\mathbf{x}_p, \delta)$$

$$\mathscr{F}_D(\mathbf{x}_p, \delta) = \begin{bmatrix} x_{1p}(k_1 r_p^2 + k_2 r_p^4) + 2p_1 x_{1p} x_{2p} + p_2(r_p^2 + 2x_{1p}^2) \\ x_{2p}(k_1 r_p^2 + k_2 r_p^4) + p_1(r_p^2 + 2x_{2p}^2) + 2p_2 x_{1p} x_{2p} \end{bmatrix}$$

with $\mathbf{x}_p = (x_{1p}, x_{2p})^T$, $r_p = \sqrt{x_{1p}^2 + x_{2p}^2}$ and $\delta = (k_1, k_2, p_1, p_2)^T$. k_1, k_2 are the radial distortion coefficients and p_1, p_2 the tangential distortion coefficients.

II. A LIFTED STRUCTURE ADJUSTMENT FORMULATION

Bundle adjustment (BA) minimizes the sum of the geometric distances of all image measurements $\mathbf{x}_{ij}$ and their corresponding projected 3D points $P_i\mathbf{X}_j$ in image space:

$$\min_{P_i, \delta, \mathbf{X}_j} \sum C(\mathbf{x}_{ij}, \mathscr{F}_D(P_i\mathbf{X}_j, \delta))$$

where P_i is the pinhole camera model, δ the distortion parameters and C is the reprojection error, for example with a quadratic error $C_s(\mathbf{x}, \mathbf{x}_p) = \left\|\mathbf{x} - \mathbf{x}_p\right\|^2$ for classical BA. Optimizing all BA parameters with all pinhole terms, distortion terms and the structure $\mathbf{X}_j$ simultaneously is ill-conditioned. Therefore, related work that also adjusts the calibration target updates the structure $\mathbf{X}_j$ in an iterative way by using heuristics of multiple BA runs [2] or use minimal structure constraints [4] and suffer from convergence issues and limitations in possible distortion models.

We want to limit the adjustment of the calibration target as far as possible and only adjust the structure if the observed error cannot be explained by other parameters of our model. Suppose we have a scalar error e and rewrite the error as a robust kernel $\psi(e)$ by introducing an additional variable w, *i.e.* a line process [3]

$$\psi(e) = \min_w \left(2w^2 e^2 + (1-w^2)^2\right) | w \in [0,1].$$

For small errors $w \to 1$ and for large errors w vanishes and $\psi(e)$ becomes constant, see [7] for an intuitive explanation in the context of outlier estimation (the same kernel is used here for simplicity) and [6] for a recent application to robust BA. We apply this concept to camera calibration and introduce variables to represent the correctness of the calibration target and therefore $3D$ structure. Adding the lifted cost function to represent structure imperfections leads to this extended calibration formulation:

$$\min_{P_i, \delta, \mathbf{X}_j, w_j} \left\{ \sum C(\mathbf{x}_{ij}, \mathscr{F}_D(P_i\mathbf{X}_j, \delta)) + \alpha \sum_j \psi(\left\|\mathbf{X}_j - \mathbf{X}_{jc}\right\|) \right\}$$

$$= \min_{P_i, \delta, \mathbf{X}_j, w_j} \left\{ \sum_{ij} C(\mathbf{x}_{ij}, \mathscr{F}_D(P_i\mathbf{X}_j, \delta)) \right.$$

$$\left. + 2\alpha \sum_j w_j^2 \left\|\mathbf{X}_j - \mathbf{X}_{jc}\right\| + \alpha \sum_j (1 - w_j^2)^2 \right\}$$

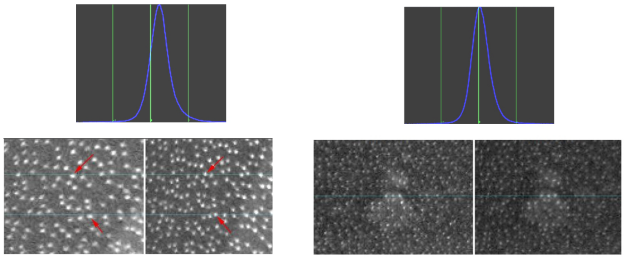
Joanneum Research Forschungsgesellschaft mbH, Steyrergasse 17, 8010 Graz, Austria firstname.lastname@joanneum.at

This work was supported by the K-Project Vision+ which is funded in the context of COMET - Competence Centers for Excellent Technologies by BMVIT, BMWFJ, Styrian Business Promotion Agency (SFG), Vienna Business Agency, Province of Styria Government of Styria and FFG under the contract 838299 HiTES3D. The programme COMET is conducted by the FFG.

where $\mathbf{X}_{jc}$ is the original reference $3D$ point and $\left\|\mathbf{X}_j - \mathbf{X}_{jc}\right\|$ corresponds to the deviation from this reference during calibration and α is a free parameter. Note that here each structure point has its own lifting variable w_j, it is also possible to represent the target accuracy with just one global scalar w. The system is solved using a standard non-linear least squares solver [1].

III. INDUSTRIAL APPLICATIONS

The presented calibration formulation has been used in different industrial applications for single- and multi-camera calibration and long term calibration maintenance using commercially printed (low cost) targets that are affected by printing inaccuracies. A handheld stereo system calibration has been kept by non-expert users under 0.06 pixel RMS reprojection error for over a year. Because non-expert users are involved, strong and robust convergence properties are essential. Figure 1 shows rectified images of this device with and without the proposed structure adjustment. The whole system performs volumetric simultaneous localization and mapping (SLAM) without opportunities for loop closing. A $3D$ model of the volumetric fusion can be seen in Figure 2. For the accuracy evaluation ground truth data of the floor plan of the scene is available. Rectification errors are accumulated through the volumetric fusion, leading to a detectable influence of slight rectification errors. A rectification error like in Figure 1a leads to drift in height of about 5cm, the shown scene is 4 meters long.



(a) Weak calibration, $0.15px$ rectification deviation from zero mean.

(b) Rectification with proposed method, nearly perfectly centered optical flow check.

Fig. 1: The top row shows a histogram of rectification deviations. They are obtained by computing a histogram of the vertical component of unconstrained optical flow initialized with the stereo result. The histogram range is ± 2 pixel. The bottom row shows the image pairs with example epipolar lines.

Figure 3 shows a stereo based inspection application for corrosion monitoring in hot steel components, ladles and process chambers that can cope with up to $1.600°C$. The main goal of the system is the detection of thinning of material i.e. volumetric changes in registered consecutively measured models. The typical distance to the target lies between 60 and $200cm$. To cope with the varying distance range focusable liquid lenses were used (Varioptic Caspian). The lenses are focusable from $7cm$ to infinity and are newly calibrated after focus change and prior to each measurement campaign. The calibration of the liquid lenses together with the high temperature environment poses the greatest challenge in this application.

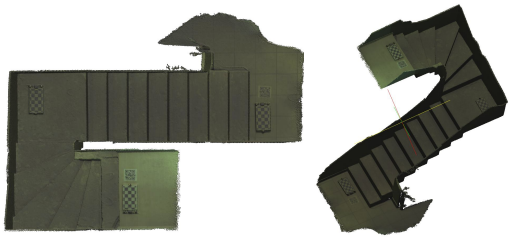


Fig. 2: A resulting 3D model obtained with a SLAM system calibrated by the presented method. Rectification errors of 0.2px are clearly noticeable in this application and lead to insufficient model accuracy.

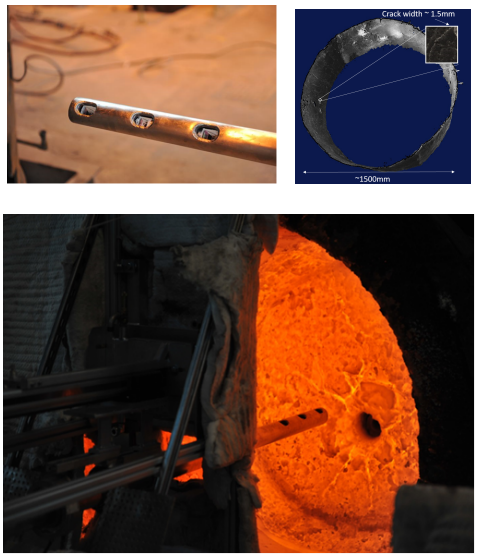


Fig. 3: Stereo system with active speckle projection for the inspection of red hot steel components, ladles and chambers with up to $1.600°C$. ©Materials Processing Institute supported by Dr BG Crutchley of i3D robotics Ltd.

REFERENCES

[1] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[2] A. Albarelli, E. Rodolà, and A. Torsello, "Robust camera calibration using inaccurate targets," *Trans. Pattern Anal. Mach. Intell*, vol. 31, no. 2, pp. 376–383, 2009.

[3] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.

[4] K. H. Strobl and G. Hirzinger, "More accurate pinhole camera calibration with imperfect planar target," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1068–1075.

[5] M. Vo, Z. Wang, L. Luu, and J. Ma, "Advanced geometric camera calibration for machine vision," *Optical Engineering*, vol. 50, no. 11, pp. 110 503–110 503, 2011.

[6] C. Zach, "Robust bundle adjustment revisited," in *European Conference on Computer Vision*. Springer, 2014, pp. 772–787.

[7] M. Zollhöfer, M. Nießner, S. Izadi, *et al.*, "Real-time non-rigid reconstruction using an rgb-d camera," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 156, 2014.

Bilateral Filters for quick 2.5 D Plane Segmentation

Simon Schreiberhuber[1], Thomas Mörwald[2] and Markus Vincze[1]

Abstract— **We present a simple and practicable approach to segment organized point clouds gathered with RGBD sensors into planar elements. The algorithm proves to execute extremely fast while delivering all the dominant planes of a scene. As an integral part of our segmentation algorithm we examined two off the shelf and one heavily modified filtering algorithms to increase the quality of the point cloud before the actual segmentation process. The results of two of these algorithms were promising. One provides a favorable tradeoff between speed and quality while the other delivers superior quality at high computational cost.**

I. Introduction

In mobile robotics many tasks have to be fulfilled in indoor environments. More specifically one task could e.g. include the search or classification of objects lying on the floor. Instead of processing all the points captured by the RGBD sensor it would be beneficial to early on discard some of the points that can not be part of the task. Removing the dominant planes from the scene is one common measure to achieve this. This becomes obvious when we observe that indoor environments are dominated by planar surfaces.

While other plane segmentation algorithms operate on unfiltered depth data, our algorithm utilizes a filtering step. Data as it is captured by an RGBD sensor tends to have multiple sources of noise, all of which tend to make the fitting of planes difficult. Reducing the noise upfront therefore is a prerequisite to a fast and simple plane segmentation approach.

To create ideal conditions for our plane segmentation algorithm we discuss three filter approaches. With these filters we aim to refine planar regions while keeping the geometric details where they are needed. We show the results generated by the standard Bilateral Filter [7], the Sigma Adaptive Bilateral Filter [2] and the adapted Bilateral Mesh Denoising algorithm [4]. A discussion shows how these filters relate to each other and how they behave in specific situations. We describe the modifications necessary to apply the Bilateral Mesh Denoising algorithm to depth data and demonstrate its effectiveness.

Regarding the core of our plane segmentation, we offer a comparison to two other algorithms: The comparably slow approach shown by Holz [5] which uses RANSAC to refine a rough normal based plane segmentation and an approach shown by Wang [8] where a rough segmentation is improved on a point-wise basis. Both algorithms start with clustering the points into a 3 D voxel grid. By doing this they are replacing the inherent neighborhood information with a costly spacial relation. Finding the nearest neighbors to a specific point no longer is a simple access to the neighboring depth pixels but a search of all points in the adjacent voxel blocks. For our segmentation we follow a similar two-step approach as in [8] but make use of the neighborhood information contained in the organized point cloud.

[1]Simon Schreiberhuber and Markus Vincze are with the Vision4Robotics group (ACIN - TU Wien), Austria {schreiberhuber, vincze}@acin.tuwien.ac.at

[2]Thomas Mörwald was a member of the Vision4Robotics group.

This work is supported by the European Comission through the Horizon 2020 Programme (H2020-ICT-2014-1, Grant agreement no: 645376), FLOBOT.

II. Related Work

Most plane segmentation approaches can be assigned to two categories. A direct approach, where planes are directly matched with the existing points, and indirect approaches where the scene is transformed into another representation.

RANSAC [3] is a direct approach that iteratively tests randomly generated plane hypothesis against a point cloud and is often used to find the ground plane of a scene. To extract multiple planes from a scene RANSAC has to be used repeatedly to assign points to different planes. The outcome of this approach is highly dependent on the order in which the RANSAC algorithm finds the planes. Thus the affiliation of points to planes is ambiguous.

The approach shown in [5] therefore does not use RANSAC for the segmentation itself but uses it to refine already existing plane hypothesis. These hypothesis are generated by clustering normal vectors in normal space or spherical coordinates. This delivers clusters of points, each of which is assembled by multiple planes facing the same direction. Averaging the normals within each of these clusters leads to a plane hypothesis which allows to separate the points into their according planes. Calculating the distance of the points to these plane hypothesis directly allows to cluster these points into their according planes.

A more direct approach was chosen in [8] is based on roughly clustering plane patches within a 3 D voxel grid. Some of these blocks within the voxel grid are containing enough points to approximate planes. In the following step it is possible to connect neighboring grid blocks to bigger surfaces wherever these planes are facing in roughly the same direction. The approach chosen by Zhang [10] is to find lines along the horizontal scan-lines which are cuts trough planes. In a second step the normals get estimated along these line segments to find corresponding segments between scan-lines. Fitting line segments can then be connected to a planar region.

The V-disparity algorithm [11] transforms the 3 D data into a V-disparity map and therefore reduces the 3 D plane fit to

a 2 D line fit which greatly reduces the computational effort. While not being as straight forward as any of the presented direct algorithms it is capable of finding planes where noise is dominant or in rough outdoor environments [9].
To improve results of certain plane segmentation algorithms it is vital to reduce noise of the input data by proper filtering. While a Gaussian Blur might be sufficient to remove noise from some intensity images it is not fit to be applied to depth maps. Besides not being able to handle areas where the sensor was unable to capture data this filter would destroy any information on discontinuities. Bilateral filtering [7] therefore is more selective and reduces over-smoothing along discontinuities. It is therefore a possible candidate for point clouds but has serious issues regarding our task since this filter introduces a bending along edges of tilted planes. This so called ski effect can be tackled by restraining the filter from working on edges [2], or by applying a bilateral filter, that is specifically designed for 3D geometry [4].
Approaches as the Total Variation [6] and the Total Generalized Variation [1] based algorithms do not inherit their principle from convolution. Instead they minimize a cost function to fulfill a tradeoff of being close to the input and minimizing a smoothness measure. The Total Variation image denoising algorithms work well for intensity images but do have downsides as a tendency to frontoparalell planes when applied to depth-maps. This tendency in particular can be countered by using the Total Generalized Variation algorithm which allows for more refined regularization with higher order derivatives but at the cost of increased computational complexity.

III. Filter

The quality of depth images obtained from the Kinect is moderate, especially in distances bigger than 3 meter (see Figure 2). To reduce the noise and other artifacts like quantization, the raw data has to be filtered.

A. Bilateral Filter

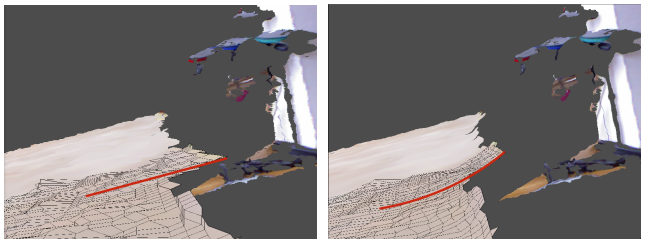
The bilateral filter is a suitable candidate. It preserves discontinuities and smooths out noise.

$$D_p^* = \frac{1}{W_p} \sum_{q \in S_p} G_{\sigma_s}(\|p-q\|) G_{\sigma_c}(|D_p - D_q|) D_q \tag{1}$$

$$W_p = \sum_{q \in S_p} G_{\sigma_s}(\|p-q\|) G_{\sigma_c}(|D_p - D_q|) \tag{2}$$

p: the coordinate of the resulting pixel.
q: the coordinate of a surrounding pixel.
G_σ: Gauss function.
$D_{p,q}$: depth values of p or q.
S_p: the neighborhood of p where $|p-q| < r_{Sth}$.
W_p: a normalization term.
σ_s: standard deviation for difference in depth.
σ_c: standard deviation for pixel distance.

This filter unfortunately introduces the unpleasant ski effect as shown in Figure 1.



(a) Unfiltered kinect image of a desk. (b) After filtering the edge of the plane is bent upwards.

Fig. 1: The ski effect (red) is introduced by Bilateral Filtering.

B. Sigma Adaptive Bilateral Filter

Andreas Deutschmann [2] introduced the Sigma Adaptive Bilateral Filter which got rid of the ski effect and is containing edges, by reducing sigma around corners and edges.

$$D_p^* = \frac{1}{W_p} \sum_{q \in S_p} G_{\sigma_{s,p}}(\|p-q\|) G_{\sigma_{c,p}}(|D_p - D_q|) D_q \tag{3}$$

$$W_p = \sum_{q \in S_p} G_{\sigma_{s,p}}(\|p-q\|) G_{\sigma_{c,p}}(|D_p - D_q|) \tag{4}$$

Where

$$\sigma_{s,c,p} = \sigma_{s,c,max} + m_{sat,p} * (\sigma_{s,c,min} - \sigma_{s,c,max}) \tag{5}$$

is depending on the depth-maps curvature

$$m_{sat,p} = \begin{cases} 1 & \text{if } m > (1-k_{th}) \\ 0 & \text{if } m < k_{th} \\ m & \text{else} \end{cases} . \tag{6}$$

With

$$m_p = \frac{\tilde{m}_p - \tilde{m}_{min}}{\tilde{m}_{max} - \tilde{m}_{min}} \tag{7}$$

and

$$\tilde{m}_p = \left\| \frac{1}{|R_p|} \sum_{q \in_p R} (P_p - P_q) \right\| . \tag{8}$$

The terms are described the following:

$\tilde{m}$: is the raw curvature.
m: is the normalized curvature of the surface see Figure 5a.
m_{sat}: is a curvature that is saturated by k_{th} and $(1-k_{th})$.
$\sigma_{s,p}$: standard deviation of the Gauss filter. Weighing depending on the difference in depth.
$\sigma_{c,p}$: standard deviation of the Gauss filter. Weighing depending on the pixel distance.
$P_{p,q}$: the point at p or q, given as vector $P_{p,q} = \begin{bmatrix} x_p & y_p & z_p \end{bmatrix}^T$
R_p: like S a neighborhood around p where $\|P_p - P_q\| < r_{Rth}$.

This filter essentially is a Bilateral Filter which is suppressed in critical regions like edges (see Figure 3).

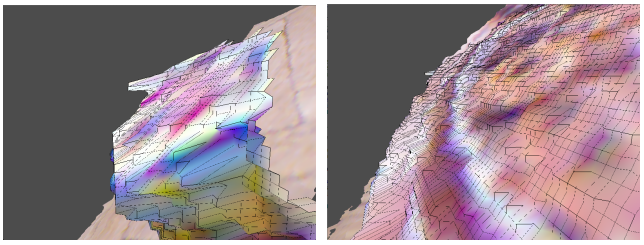
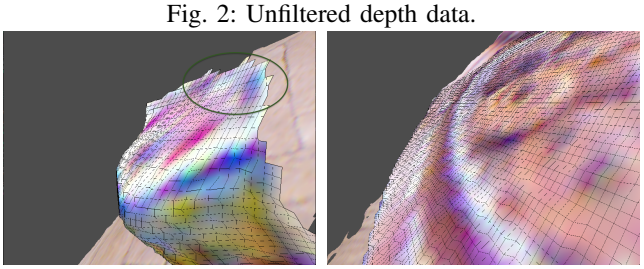
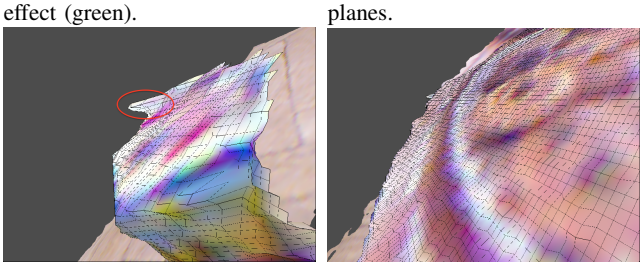


Fig. 2: Unfiltered depth data.



(a) Bilateral Filter. Introduces ski effect (green).

(b) But shows good results at planes.



(c) Sigma Adaptive Filter. Preserves details like edges, but can't filter noise at discontinuities (red).

(d) Delivers good results when applied to planar regions.

Fig. 3: Filtering results of Bilateral Filter and Sigma Adaptive Filter.

C. Bilateral Tangential Filter

The promoted filter is based on the Bilateral Mesh Denoising algorithm [4] which is used for meshes but not raw depth data. The idea behind this filter is to correct each point along its normal by a value composed by the deviation of surrounding points to its tangent plane. We adapt this principle to depth data by not correcting the points along their normal as in [4] but along the camera view rays. The filter is written as

$$C_p = \frac{1}{W_p} \sum_{q \in S_p} G_{\sigma_s}(\|p-q\|) G_{\sigma_c}(d_{p,q}) d_{p,q} \tag{9}$$

$$W_p = \sum_{q \in S_p} G_{\sigma_s}(\|p-q\|) G_{\sigma_c}(d_{p,q}) \tag{10}$$

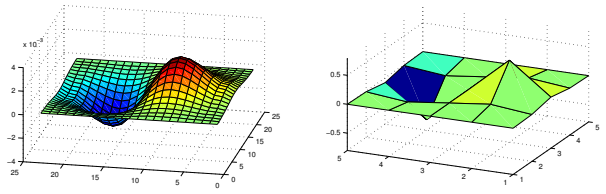
where the correction term $C_{p,k}$ is used to correct the depth values

$$D_p^* = D_p + C_p. \tag{11}$$

$d_{p,q}$ is the distance of the point q to the tangent plane of p

$$d_{p,q} = n_p \cdot (P_p - P_q). \tag{12}$$

It is not implied in this equation, but this filter is meant to be used iteratively.



(a) $K_{q,big}$.

(b) $K_{q,small}$.

Fig. 4: Filtering kernels, to calculate horizontal and vertical derivation of x, y and z. Sizes for these kernels are 23x23 and 5x5.

The quality of this filter strongly depends on the normal vectors n_p which tend to be difficult to obtain, especially along discontinuities and in noisy data. Incorrect normal values can make the algorithm locally unstable. Figure 6 shows a good example for how normal vectors affect the result. The normal vector is calculated by the vertical and horizontal derivation of x, y and z coordinates by the image coordinates u and v.

$$n_p = \frac{\tilde{n}_p}{\|\tilde{n}_p\|} \qquad \tilde{n}_p = \begin{bmatrix} \frac{dx_p}{du} \\ 0 \\ \frac{dz_p}{du} \end{bmatrix} \times \begin{bmatrix} 0 \\ \frac{dy_p}{dv} \\ \frac{dz_p}{dv} \end{bmatrix} \tag{13}$$

To obtain the needed derivatives we can not rely on a Canny Edge detection like approach since this would lead to wrong normals along discontinuities. We therefore have to mix the Canny Edge detection with the idea of the Bilateral Filter. To reduce the impact of discontinuities on the normals, points which are further away from the center point contribute less or not at all. This is achieved by an other Gaussian term G_{σ_n}.

$$\frac{d(x,y,z)_p}{du,v} = \sum_{q \in S_p} K_{q,p} G_{\sigma_n}(D_p - D_q)((x,y,z)_p - (x,y,z)_q) \tag{14}$$

Since the depth data along edges of objects is often distorted, it is necessary to compensate for that by locally extending the kernel:

$$K_{q,p} = \begin{cases} K_{q,big} & \text{if } c_p > c_{th} \\ K_{q,small} & \text{else} \end{cases} \tag{15}$$

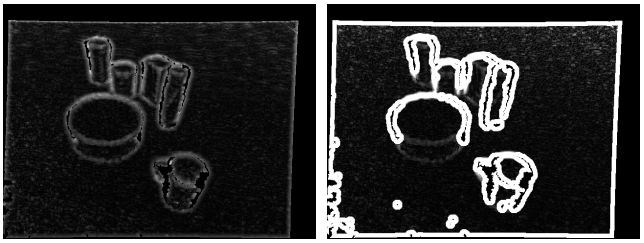
The filter kernels itself are shown in Figure 4. As basis to decide we are using a measure for how erratic the image is (Figure 5b).

$$c_p = \sum_{q \in R'_p} |D_p - D_q| \tag{16}$$

One example for proper filtering kernels are shown in Figure 4. Note that R'_p is in this case the neighborhood of p where $|p-q| < r_{R'th}$.

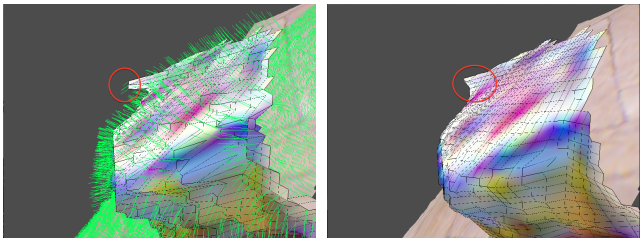
IV. Results of Filtering

The standard Bilateral Filter introduces the unpleasant ski effect [2] and therefore does not preserve information on edges (see Figure 3). On said edges the ski effect refers to a

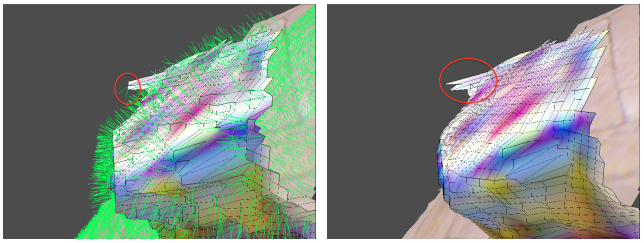


(a) m_p. (b) c_p

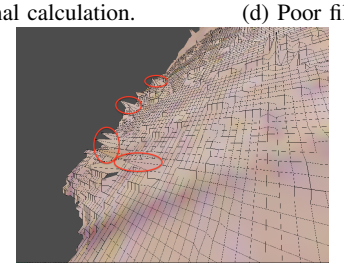
Fig. 5: Curvature m_p and unsteadyness c_p. These measures are used to guide σ in the adaptive filter and the normal estimation in the tangential filter.



(a) Correct normals. (b) Reasonable result.



(c) Poor normal calculation. (d) Poor filtering.



(e) Even along plane surfaces, poorly calculated normals are leading to artifacts or instabilities.

Fig. 6: Filtering results of the proposed Bilateral Tangential filter.

slight bending towards fronto-parallelity. Note that this effect gets stronger as the planes get tilted.
The Sigma Adaptive Bilateral Filter gets rid of this effect by not filtering in these critical regions. As seen in Figures 3c and 3d the results are comparable to the standard Bilateral Filter but without creating the ski defects. Spikes, as they often appear at sharp edges, will unfortunately not undergo any smoothing. Since we selected this filter to support our segmentation we created an GPU (AMD Radeon HD 6750M) implementation that computes within 25 ms.
For the Bilateral Mesh Denoising algorithm the results are different (see Figure 6). While being equally as suitable for the planar regions as the Bilateral and Sigma Adaptive Bilateral Filter, this algorithm shows the best results along discontinuities. In terms of computational complexity this algorithm unfortunately is way more demanding than the other two. This is mainly due to the complex normal estimation but also because it needs two to three iterations the other algorithms compute within one.

V. Segmentation

Two examples for state of the art algorithms coming close to a $30\,Hz$ segmentation rate are [5] and [8]. The algorithm shown in [5] utilizes a segmentation in normal space but is slower than the other. We therefore follow the approach shown in [8] where the points are split into a 3D voxel grid. A coarse pre-segmentationin on these then segments a majority of points with a relatively small amount of computations. Although the this approach is the faster one, it still is overly complicated for our needs. Seperating the organized pointcloud into equally sized cubes (voxels) only creates the need to compare these voxels to their 26 neighbouring voxels.

A. Hierarchical Plane Segmentation

The proposed algorithm follows the idea of pre-segmentation and splits the depth image into smaller patches similar to [8] but does it in image space. This reduces the number of neighbors for each patch to 8 and therefore saves computation time. The main steps of the algorithms are:

1) Patch generation: The depth data is grouped into equally sized section with sizes like e.g. $10 \times 10\,pixel$. It is then tried to fit a plane into these points. If there are enough points within a threshold of this plane the patch is retained and the points will be assigned to this patch. When this criteria is not met the patch is discarded and the according points stay unassigned.
2) Patch Segmentation: The initially unassigned patches get grouped together to assemble planes. This happens according to their normal vector and position.
3) Post filtering: During this phase, no new patches will be added, but every pixel, which is bordering onto a plane and meets certain conditions, will be assigned to this plane.

1) Patches: As already mentioned. Patches are small equally sized fragments of the depth image and described by their plane equation:

$$ax + by + cz - 1 = 0 \tag{17}$$

The parameters can be acquired by principal component analysis of all points. After getting the parameters, it is necessary to test if they provide a good description of the plane. To ensure this, at least a certain percentage (e.g. 90 %) of the points considered for this patch should be inside the

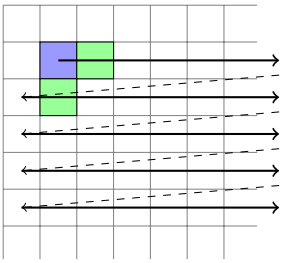


Fig. 7: Segmentation strategy for patches. Every valid patch (blue) is a cluster of points e.g. $10 \times 10\,pixel$ and will be connected to an neighboring (green) existing collection of patches if it fits to one of the existing plane hypothesis. If it can not be added to an existing hypothesis it will become the starting point for a new hypothesis.

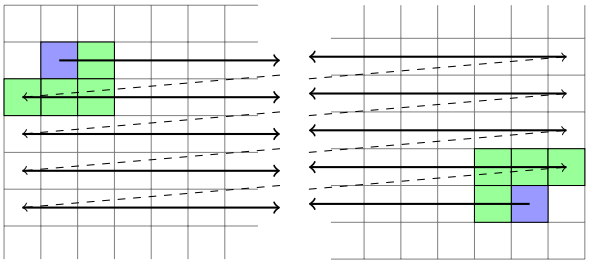


Fig. 8: The bottom up and top down processing steps following the same pattern: The center point (blue) is traversing the image pixelwise in the directions top-down (left) or bottom-up (right). When one of the center points neighboring pixels (green) is a suitable candidate for the center points plane hypothesis, it will get added to this plane.

approximated plane. For this the distance

$$d = \frac{|ax+by+cz-1|}{\sqrt{a^2+b^2+c^2}} < d_{th} \tag{18}$$

has to be below a threshold (e.g. 1 cm).

2) Segmentation: These patches can easily be grouped by any clustering algorithm that supports 4 or 8 connectivity. Neighboring planes or patches can be combined by meeting the criteria of pointing roughly in the same direction e.g. +-15 °.

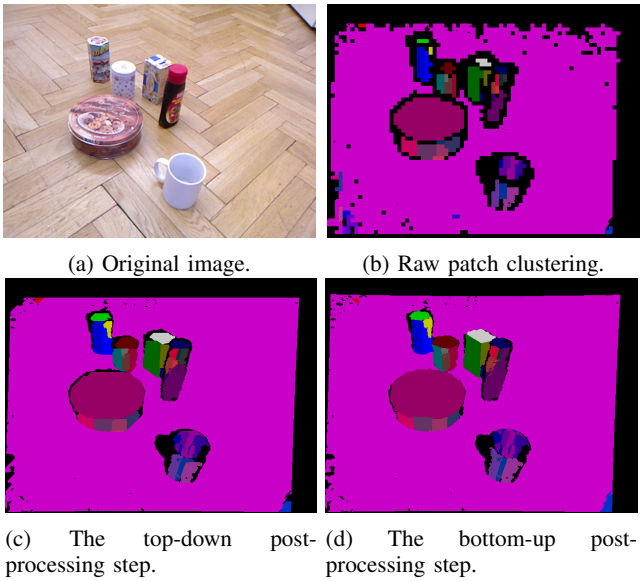
In this implementation it was sufficient to run one pass with the following strategy (see Figure 7):

1) If the current patch (blue) is not already assigned to a plane, create a new plane with this patch as first member.
2) If the neighboring (green) patch to the right has the same normal direction as the plane of the current patch, add the patch (green) to this plane. If the patch to the right is already assigned to a plane, and both plane normals are similar, merge the planes.
3) Merge the patch to the bottom with the current plane if the normal direction is similar.

3) Post-processing: The segmentation of the bigger patches are by far not satisfying because they leave a lot of pixel unassigned. In the last step the filter is running from top left to bottom right and vice versa (see Figure 8) to assign pixel to the most fitting plane. To assign a pixel to a plane it must meet one of the following criteria, otherwise it stays unassigned or assigned to its current plane.

- The considered point is unassigned and fits inside the neighboring plane.
- If the point is already assigned to a plane, which size is a lot smaller (e.g. factor of 10) than the new plane, the point simply has to be close enough ($d < d_{th}$ to get reassigned.
- If the point is already assigned to a plane, which is of similar size ($|P_{new}|f > |P_{current}| > |P_{new}|\frac{1}{f}$) the point has to be closer to the new plane, than to the old plane ($d_{P_{new}} < d_{P_{current}}$).

Note that small planes can't take away points from bigger planes but bigger planes sure can do this to smaller ones.



(a) Original image. (b) Raw patch clustering.

(c) The top-down post-processing step. (d) The bottom-up post-processing step.

Fig. 9: Synopsis of the segmentation process.

This is a strategy to eliminate smaller planes, that might be created in the first step due to oversegmentation. One might replace this strategy by a more sophisticated one. An other parameter that could additionally be taken into account is the normal vector of each point, which should show into the same direction as the plane it is added to.

VI. Results of Segmentation

The simple plane segmentation algorithm provides useable results for indoor scenarios as seen in Figure 9. It is notable that the depthmap quality degrades in the image corners. As a result the algorithm wrongly creates another plane in this region (bottom right corner). Besides this, the algorithm shows the desired behavior. The cylindrical regions around the cans and boxes are approximated by smaller planes, while smaller planar surface patches of boxes get detected as such. In terms of frame rate our algorithm is competitive as it runs at 22 Hz while processing a 640×480 pixel depth map. The algorithms described by Holz [5] (7 Hz) and Wang

[8] (25 Hz) additionally implement some kind of obstacle detection but do not utilize pre-filtering. Apart from this, the conditions are reasonably similar. On the hardware side all results where achieved on an Intel Core i7 with around 2 GHz while utilizing only one CPU core and the GPU for pre filtering.

VII. Conclusion

We introduced a new plane segmentation approach for 2.5 D data. It shows competitive results for both, quality and speed. Our algorithm relies on a filtering step that improves the quality of the input data. Hence, we conducted an analysis of three filters to find a fitting candidate.

We selected the Sigma Adaptive Bilateral Filter wich balances speed and quality. Our GPU implementation of the filter algorithm runs within 25 ms on an AMD Radeon HD 6750M. The mesh denoising algorithm [4], together with our modifications showed promising results. To utilize this algorithm in real-time, GPUs with higher performance could be a possible solution. Apart from this, both filters could be improved by adding a noise model that handles the increased noise levels at higher distances.

The proposed segmentation algorithm shows competitive results that were achieved with a hierarchical strategy. Splitting up the segmentation into a coarse pre-segmentation and a fine grained post-processing step holds the run-time competitive. Future work could extend this algorithm with a sensor model that leads to additional rules such as e.g. depth dependent thresholds.

References

[1] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM J. Img. Sci.*, vol. 3, no. 3, pp. 492–526, Sep. 2010.

[2] A. Deutschmann, *Kantenselektiver Filter für Punktwolken*. TU Wien, 2013.

[3] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[4] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. New York, NY, USA: ACM, 2003, pp. 950–953.

[5] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, *Real-Time Plane Segmentation Using RGB-D Cameras*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 306–317.

[6] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259 – 268, 1992.

[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Jan 1998, pp. 839–846.

[8] Z. Wang, H. Liu, Y. Qian, and T. Xu, *Real-Time Plane Segmentation and Obstacle Detection of 3D Point Clouds for Indoor Scenes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 22–31.

[9] D. Yiruo, W. Wenjia, and K. Yukihiro, "Complex ground plane detection based on v-disparity map in off-road environment," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 1137–1142.

[10] L. Zhang, D. Chen, and W. Liu, "Fast plane segmentation with line primitives for rgb-d sensor," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, p. 8, 2016.

[11] J. Zhao, J. Katupitiya, and J. Ward, "Global correlation based ground plane estimation using v-disparity image," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, April 2007, pp. 529–534.