Proceedings of the

OAGM Workshop 2018

Medical Image Analysis

May 15-16, 2018
Private University for Health Sciences,
Medical Informatics and Technology (UMIT)
Hall in Tirol

OAGM - Austrian Association for Pattern Recognition

UMIT the health & life sciences university
TU Graz Graz University of Technology
OAGM
OESTERREICHISCHE COMPUTER GESELLSCHAFT AUSTRIAN COMPUTER SOCIETY

Martin Welk, Martin Urschler, and Peter M. Roth (eds.)

Proceedings of the

OAGM Workshop 2018

Medical Image Analysis

May 15-16, 2018
Hall/Tyrol, Austria

Austrian Association of Pattern Recognition (OAGM)

OESTERREICHISCHE COMPUTER GESELLSCHAFT
AUSTRIAN COMPUTER SOCIETY

OAGM

Editors

Martin Welk, Martin Urschler, and Peter M. Roth

Layout

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

Cover

Stefan W. Schleich

ISBN (e-book) 978-3-85125-603-1
DOI 10.3217/978-3-85125-603-1

Contents

Preface

The Private University for Health Sciences, Medical Informatics and Technology (UMIT) and the Austrian Association for Pattern Recognition (AAPR/OAGM) welcome you at Hall/Tyrol for the 42nd Annual Workshop of the AAPR that takes place on May 15/16 at the UMIT campus.

The workshop provides a platform for presentation and discussion of research progress as well as current projects within the AAPR community. In this year's edition of the workshop, OAGM2018, we additionally focus on the theme of medical image analysis and applications of computer vision, image processing and pattern recognition in the medical context, with the aim to bring together Austrian and nearby located groups working on this topic for discussion and establishing potential collaborations.

From the vivid Austrian and international community in the field, a total of 24 full papers and application spotlight papers were submitted to the workshop. Prior to the workshop, the program committee has carefully reviewed all submissions. From the submitted papers, 19 papers were finally included in the conference program as oral or poster presentations. Two invited speakers, Prof. Marleen de Bruijne (Rotterdam/Copenhagen) and Prof. Bjoern Menze (Munich), will present keynote lectures on their research in Medical Image Analysis. The conference program is complemented by 8 featured presentations in which scientists from the AAPR community will showcase outstanding recent contributions accepted by leading international conferences and journals.

Combining all these, the final program represents an impressive cross-section of current research in the medical image analysis, pattern recognition and vision field in and around Austria. We look forward to lively discussions and scientific exchange during the conference.

Martin Welk, Martin Urschler, Peter M. Roth
Hall/Tyrol, May 2018

Workshop Chair

Martin Welk, UMIT Hall/Tyrol

Workshop Co-Chairs

Martin Urschler, Ludwig Boltzmann Institute for Clinical Forensic Imaging
Peter M. Roth, Graz University of Technology

Program Committee

Helmut Ahammer, Medical University of Graz
Reinhard Beichel, University of Iowa
Csaba Beleznai, Austrian Institute of Technology
Horst Bischof, Graz University of Technology
Kristian Bredies, Karl Franzens University of Graz
Katja Bühler, VRVis Vienna
Wilhelm Burger, FH Hagenberg
Gernot Stübl, Profactor GmbH and JKU Linz
Cornelia Fermüller, University of Maryland
Friedrich Fraundorfer, Graz University of Technology
Karl Fritscher, UMIT Hall/Tyrol
Harald Ganster, Joanneum Research Graz
Margrit Gelautz, Vienna University of Technology
Sasa Grbic, Siemens Corporate Res. Princeton
Martin Hirzer, Graz University of Technology
Bernhard Kainz, Imperial College London
Walter G. Kropatsch, Vienna Univ. of Technology
Roland Kwitt, University of Salzburg
Christoph Lampert, IST Austria
Georg Langs, Medical University of Vienna
Mathias Lux, Alpen-Adria University Klagenfurt
Klaus Maier-Hein, DKFZ Heidelberg
Hubert Mara, Heidelberg University
Bernhard Moser, SCC Hagenberg
Thomas Pock, Graz University of Technology
Hayko Riemenschneider, ETH Zürich
Robert Sablatnig, Vienna University of Technology
Josef Scharinger, JKU Linz
Konrad Schindler, ETH Zürich
Veronika Schöpf, Karl Franzens University of Graz
Rainer Schubert, UMIT Hall/Tyrol
Andreas Uhl, University of Salzburg
Markus Vincze, Vienna University of Technology
Tomaz Vrtovec, University of Ljubljana
Christian Wachinger, LMU Munich
Christopher Zach, Toshiba Research Cambridge

Awards 2017

The

OAGM Best Paper Award 2017

was awarded to the papers

Generative Adversarial Network based Synthesis for Supervised Medical Image Segmentation

by

Thomas Neff, Christian Payer, Darko Štern, and Martin Urschler

and

Using a U-Shaped Neural Network for minutiae extraction trained from refined, synthetic fingerprints

by

Thomas Pinetz, Daniel Soukup, Reinhold Huber-Mörk, and Robert Sablatnig.

The

IEEE RAS Austria Best Student Award 2017

was awarded to the paper

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

by

Stefan Loigge, Clemens Mühlbacher, Gerald Steinbauer, Stephan Gspandl, and Michael Reip.

Index of authors

Keynote Talks

Machine learning imaging biomarkers

Marleen de Bruijne[1]
marleen.debruijne@erasmusmc.nl

Abstract

Quantitative analysis of medical imaging data is increasingly important in clinical studies as well as in the diagnosis, monitoring, and prognosis of disease in individual patients. Traditional techniques measure factors that are well-known to indicate disease, such as for instance the density of lung tissue, which relates to lung function, or the size of certain brain structures, which may help to predict the development of dementia. Advances in machine learning together with increased computational power now allow a new, more data-driven approach: image characteristics related to disease outcome can be learned directly from databases that combine medical imaging data with other patient data. This talk will cover different approaches to learning disease-specific models from imaging data, including techniques to address common issues in (medical) image analysis: varying scan protocols, weakly annotated data, and missing data.

[1]University of Copenhagen, Denmark

Extracting and modeling information from medical images

Bjoern Menze[1]
bjoern.menze@tum.de

Abstract

The computer based extraction of biomarkers that support the evaluation of clinical image data is an established field in diagnostic radiology. Recently, approaches and ideas that are described by terms, such as 'image phenotyping', 'imaging genetics', or 'radiomics', gained significant interest in the field. They all share a similar technical approach and aim at the direct inference of properties of the underlying disease grade and process using image information, replacing or complementing genetic and clinical descriptors in diagnostic decisions. Of particular relevance in their design and application is the identification of patient subgroups that may be susceptible to new targeted treatments. It is widely believed that this new generation of computational decision support tools has the potential to transform the quantitative analysis of clinical imaging data and the implementation of empirical diagnostic rules in the clinical workflow. Following the pipeline for a 'radiomics'-like information extraction, I will present recent work on medical image quantification, benchmarking of algorithms, and data-driven as well as physical-inspired modeling of the underlying disease process. A focus will be on applications from the field of oncological imaging.

[1]TUM Computer Science

Contributed Session 1

Image Retrieval with BIER: Boosting Independent Embeddings Robustly

Michael Opitz, Georg Waltner, Horst Possegger and Horst Bischof

***Abstract*— Deep metric learning methods embed an image into a high dimensional feature space in which similar images are close to each other and dissimilar images are far apart from each other. However, state-of-the-art deep metric learning approaches typically yield highly correlated embeddings. To address this issue, we propose a method called Boosting Independent Embeddings Robustly (BIER) which divides the last embedding layer of a metric CNN into several smaller embeddings. We train these embeddings with online gradient boosting to make the learners more diverse from each other. During training, each learner receives a reweighted training sample from the previous learner. Additionally, we use an auxiliary loss function to increase the diversity between learners. In our experiments we show that BIER significantly reduces correlation in the embedding layer and consequently improves accuracy. We evaluate BIER on several image retrieval datasets and show that it significantly outperforms the state-of-the-art.**

I. INTRODUCTION

Deep Convolutional Neural Network (CNN) based metric learning approaches learn a distance function between images. This function maps semantically similar images close to each other and dissimilar images far apart from each other.

State-of-the-art approaches in metric learning typically saturate or decline due to over-fitting, especially when large embeddings are used [4]. To address this issue, we proposed a learning approach, called Boosting Independent Embeddings Robustly (BIER) [5], [6], which leverages large embedding sizes more effectively. Rather than using a single large embedding, BIER divides the last embedding layer of a CNN into multiple non-overlapping groups (see Fig. 1). Each group is a separate metric learning network on top of a shared feature extractor. To make learners diverse from each other we train our learners with online gradient boosting [6], and use auxiliary loss functions between pairs of learners [5].

We demonstrate the effectiveness of our metric on several image retrieval datasets [4], [7] and show that we can significantly outperform state-of-the-art approaches.

II. BIER

To train our network, we adapt an online gradient boosting algorithm [1]. During forward propagation we sample an mini-batch and compute the loss function for the first learner. The learner then reweights the training set according to the negative gradient of the loss function for the successive learner. After the last learner computes the loss, the gradients are backpropagated to the hidden layers of the CNN, as illustrated in Fig. 1.

*This work was supported by the Austrian Research Promotion Agency (FFG) Project MANGO (836488) and DARKNET (85891).

Graz, University of Technology, michael.opitz@icg.tugraz.at

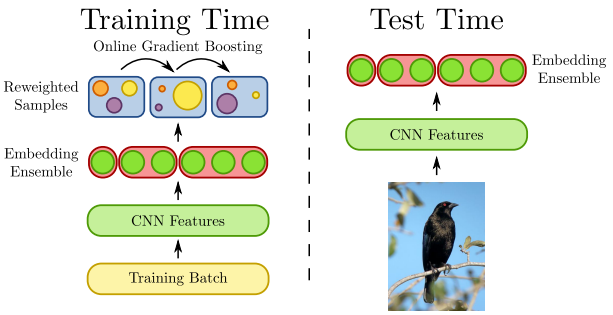


Fig. 1. During training time BIER uses online gradient boosting to train the individual learners. During test time we simply concatenate the predictions of all our learners to a single feature vector.

To further increase diversity in our method, we propose a novel auxiliary loss function [5]. We add adversarial regressors on pairs of learners. These regressors try to map one embedding to an other embedding, maximizing their similarity. Since we are using a gradient reversal layer [2], our hidden layers minimize the similarity w.r.t. to these regressors, making the embedding more diverse.

III. RESULTS

In our experiments we observe that BIER significantly reduces correlation of the embedding on the CUB dataset [7] by about 47.8%. We also compare our method and baseline to the state-of-the-art in Table I. BIER significantly improves performance and outperforms state-of-the-art methods.

TABLE I

EVALUATION OF BIER ON CUB [7] AND STFD. ONLINE PRODUCTS [4].

	CUB (R@1)	Stanford Online Products (R@1)
Proxy NCA [3]	49.2	73.7
Baseline	51.8	66.2
BIER	**57.5**	**74.2**

REFERENCES

[1] A. Beygelzimer, S. Kale, and H. Luo, "Optimal and Adaptive Algorithms for Online Boosting." in *Proc. ICML*, 2015.

[2] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-Adversarial Training of Neural Networks," *JMLR*, vol. 17, no. 59, pp. 1–35, 2016.

[3] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, "No Fuss Distance Metric Learning Using Proxies," in *Proc. ICCV*, 2017.

[4] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep Metric Learning via Lifted Structured Feature Embedding," in *Proc. CVPR*, 2016.

[5] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, "Deep Metric Learning with BIER: Boosting Independent Embeddings Robustly," *arXiv:cs/1801.04815, submitted to TPAMI*, 2018.

[6] ——, "BIER: Boosting Independent Embeddings Robustly," in *Proc. ICCV*, 2017.

[7] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

Multi-Label Whole Heart Segmentation Using Anatomical Label Configurations and CNNs

Christian Payer[1], Darko Štern[2], Horst Bischof[1], Martin Urschler[2]

I. Extended Abstract

The accurate analysis of the whole heart substructures, i.e., left and right ventricle, left and right atrium, myocardium, pulmonary artery and the aorta, is highly relevant for cardiovascular applications. Therefore, automatic segmentation of these substructures from CT or MRI volumes is an important topic in medical image analysis [4]. Challenges for segmenting the heart substructures are their large anatomical variability in shape among subjects, the potential indistinctive boundaries between substructures and, especially for MRI data, artifacts and intensity inhomogeneities resulting from the acquisition process. To objectively compare and analyze whole heart substructure segmentation approaches, efforts like the MICCAI 2017 Multi-Modality Whole Heart Segmentation (MM-WHS) challenge are necessary and important for potential future application of semi-automated and fully automatic methods in clinical practice. We participated in the MM-WHS challenge, where we proposed a deep learning framework for fully automatic multi-label segmentation [2]. Evaluated on the MM-WHS challenge test data, we rank first for CT and second for MRI with a whole heart segmentation Dice score of 90.8% and 87%, respectively, leading to an overall first ranking among all participants.

Our proposed method [2] performs fully automatic multi-label whole heart segmentation with CNNs using volumetric kernels. Due to the extensive memory and runtime requirements of volumetric CNNs, we use a pipeline of two CNNs that first *localizes* the heart in lower resolution volumes to crop a standardized region around the heart, followed by obtaining the final *segmentation* in higher resolution within this region (see Fig. 1).

The *localization* CNN based on the U-Net [3] performs landmark localization using heatmap regression [1] to predict the approximate center of the bounding box around all heart substructures. Then, we crop a region with fixed physical size around the predicted center, ensuring that the region encloses all segmentation labels. Within the cropped region, the multi-label *segmentation* CNN predicts the heart substructure labels of each voxel. For this task, we use an adaptation of the fully convolutional end-to-end trained SpatialConfiguration-Net (SCN) from [1]. The main idea of the three-component SCN is to learn from relative positions among structures to focus on anatomically feasible configurations as seen in the training data.

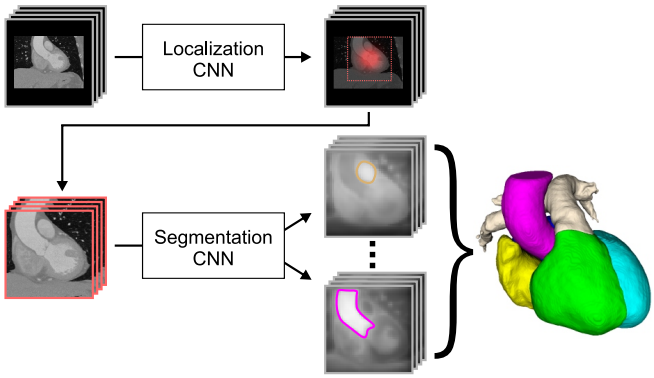


Fig. 1. Overview of the fully automatic two-step multi-label segmentation pipeline. The first CNN uses a low resolution volume as input to localize the center of the bounding box around all heart substructures. The second CNN crops a region around this center and performs the multi-label segmentation. The figure is adapted from [2].

In the first component of the SCN, a U-Net-like architecture [3] generates the first intermediate label predictions, corresponding to a voxel-wise probability of all labels. Then, the second component models the spatial configuration among labels, by using consecutive convolution layers to transform these probabilities to positions of other labels, generating the second intermediate label predictions. Finally, the third component multiplies both intermediate predictions, which results in the combined label predictions. Note that only when trained in an end-to-end manner, this last multiplication ensures that the first and second network component perform as expected. Without any further postprocessing, choosing the maximum value among the label predictions for each voxel leads to the final multi-label segmentation.

*This work was supported by the Austrian Science Fund (FWF): P 28078-N33.

[1]Christian Payer and Horst Bischof are with the Institute of Computer Graphics and Vision, Graz University of Technology, Austria christian.payer@icg.tugraz.at

[2]Darko Štern and Martin Urschler are with the Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria martin.urschler@cfi.lbg.ac.at

References

[1] C. Payer, D. Štern, H. Bischof, and M. Urschler, "Regressing Heatmaps for Multiple Landmark Localization Using CNNs," in *Proc. Med. Image Comput. Comput. Interv.* Springer, 2016, pp. 230–238.

[2] C. Payer, D. Štern, H. Bischof, and M. Urschler, "Multi-label Whole Heart Segmentation Using CNNs and Anatomical Label Configurations," in *Stat. Atlases Comput. Model. Hear. ACDC MMWHS Challenges. STACOM 2017.* Springer, 2018, pp. 190–198.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proc. Med. Image Comput. Comput. Interv.* Springer, 2015, pp. 234–241.

[4] X. Zhuang and J. Shen, "Multi-scale patch and multi-modality atlases for whole heart segmentation of MRI," *Med. Image Anal.*, vol. 31, pp. 77–87, 2016.

Contributed Session 2

Multivarite Manifold Modeling of Functional Connectivity in Developing Language Networks

Ernst Schwartz[1,2], Karl-Heinz Nenning[1], Gregor Kasprian[1], Anna-Lisa Schuller[2], Lisa Bartha-Doering[2] and Georg Langs[1]

***Abstract*— In a recent paper [2], we presented a method for the modelling of brain networks in the space of symmetric positive definite matrices (Sym^+). We showed that this mathematical framework enables an accurate representation of the effects of factors such as age, sex or mental state on the Functional Connectivity (FC) between brain regions.**

I. METHOD

FC of two brain regions is determined from the covariance $\mathrm{Cov}(p_1,p_2)$ of the BOLD fMRI signal time courses p_1 and p_2 observed at distinct locations in the brain. Matrices $P, P_{\alpha\beta} = P_{\beta\alpha} = \mathrm{Cov}(p_\alpha, p_\beta), i,j \in 1\ldots n$ representing networks of FC between n observed regions are elements of the Riemannian Manifold $\mathscr{M}$ of Symmetric Positive Definite (SPD) matrices Sym_n^+. Positive-definiteness implies $\mathbf{v}^\top P\mathbf{v} > 0 \quad \forall \mathbf{v} \in \mathbb{R}^{+n}, P \in \mathrm{Sym}_n^+$, which renders elements of P interrelated. Euclidean operations do not accurately reflect this underlying geometry of the SPD manifold and can therefore lead to distorted results.

We are interested in describing the effects of known extrinsic information such as patient age, sex or current mental activity on the measured FC matrices. In the Euclidean setting, linear models of the effects of such covariates x_{ij} are fitted to observations P_i obtained from i sources by simple least squares. However, this type of modelling makes the assumption that individual entries $P_{\alpha\beta}$ are mutually independent. By solving the regression model directly in Sym_n^+ [1] as

$$\min_{\tilde{M}\in\mathscr{M},\mathbf{V}_j\in T_P\mathscr{M}} \sum_{i=1}^{N} \left|\left|\mathrm{Log}_{\tilde{P}_i}(P_i)\right|\right|^2_{T_{\tilde{Y}_i}\mathscr{M}}, \quad \tilde{P}_i = \mathrm{Exp}_{\tilde{M}}\left(\sum_{j=1}^{K} \mathbf{V}_j x_{ij}\right) \tag{1}$$

, the resulting intercept $\tilde{M}$ and factors $\mathbf{V}_j$ are by definition elements of Sym_n^+ themselves and therefore capture the interdependence of the entries $P_{\alpha\beta}$.

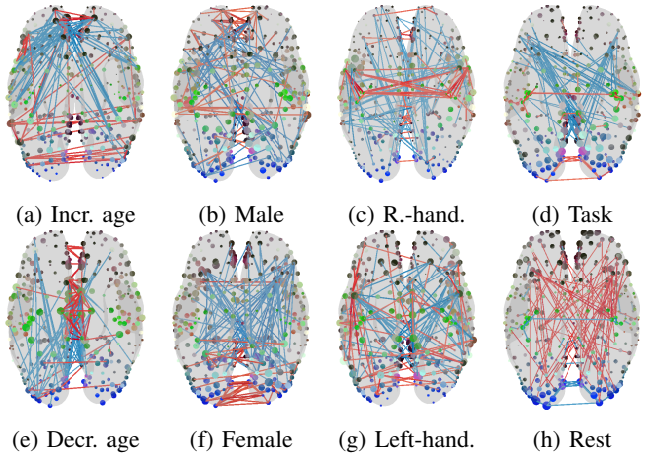
II. RESULTS

We computed both Euclidean and Riemannian (Eq. 1) models of FC measured in 20 children aged 6 to 13 in relationship to their age, sex, handedness and mental state (at rest vs. performing a language task).

*This project was supported by FWF (KLI 544-B27, I 2714-B31) and OeNB (15356, 15929).

[1]CIR Lab and Division of Neuroradiology and Musculoskeletal Radiology, Dept. of Biomedical Imaging and Image-guided Therapy, Medical University Vienna ernst.schwartz@meduniwien.ac.at

[2]Dept. of Pediatrics and Adolescent Medicine, Medical University Vienna



(a) Incr. age (b) Male (c) R.-hand. (d) Task

(e) Decr. age (f) Female (g) Left-hand. (h) Rest

Fig. 1: Effect of varying individual covariates $\mathbf{V}_j$ (from [2])

We were able to show that the Riemannian model (Fig. 1) more accurately reflects the observed population in numerous ways. For example, the expected value of the distribution of the values of the intercept $\tilde{M}$ ($\mathbb{E}[\tilde{M}] = -0.0169$) more closely matches that of the overall population ($\mathbb{E}[M] = -0.0173$) , whereas the Euclidean mean $\hat{M}$ introduces a bias towards anti-correlations ($\hat{M}$ ($\mathbb{E}[\hat{M}] = -0.0418$).

Using both the Euclidean and Riemannian models, we simulated the FC of an average subject and vary the simulated mental state by adjusting the corresponding covariate. We compute the correlation between the simulated FC of a language-specific brain area, the Peri-Sylvian Language area (PSL) and the average FC of the same region obtained from a large reference cohort [3]. The maximum observed correlation between the refrence profile and those obtained from the simulations is higher for the Riemannan model ($R^2 = 0.61$, $\rho <$ 1e-37 compared to $R^2 = 0.58$, $\rho <$ 1e-35), indicating a higher predicitive performance of the Riemannian model.

REFERENCES

[1] H. J. Kim, N. Adluru, M. D. Collins, M. K. Chung, B. B. Bendlin, S. C. Johnson, R. J. Davidson, and V. Singh, "Multivariate general linear models (mglm) on riemannian manifolds with applications to statistical analysis of diffusion weighted images," in *CVPR 2014*, pp. 2705–2712.

[2] E. Schwartz, K. Nenning, G. Kasprian, A. Schuller, L. Bartha-Doering, and G. Langs, "Multivariate manifold modelling of functional connectivity in developing language networks," in *IMPI 2017*, 2017, pp. 311–322.

[3] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium, *et al.*, "The wu-minn human connectome project: an overview," *Neuroimage*, vol. 80, pp. 62–79, 2013.

Early Predictors of Bone Infiltration in Multiple Myeloma Patients from T2 weighted MRI Images

Roxane Licandro[1,2], Johannes Hofmanninger[2], Marc-André Weber[3], Bjoern Menze[4], Georg Langs[2]
roxane.licandro@meduniwien.ac.at

***Abstract*— The analysis of bone infiltration patterns is a key issue in assessing the progression state of Multiple Myeloma (MM) and corresponding treatment response. MM is a blood affecting disease, that leads to an uncontrolled proliferation and malignant transformation of plasma cells and B-lymphocytes and ultimately can lead to osteolytic lesions first visible in Magnetic Resonance Imaging (MRI). It is particularly important to reliably assess lesions as early as possible, since they are a prime marker of disease advance and a trigger for treatment. However, their detection is difficult. Here, we present first results for the prediction of lesion progression based on longitudinal T2 weighted MRI imaging data. We evaluate a predictor for the identification of early signatures of emerging lesions, before they reach report thresholds. The algorithm is trained on longitudinal data, and visualizes high-risk locations in the skeleton.**

I. Introduction

Multiple Myeloma (MM) is a blood affecting malignancy of the bone marrow, that disturbes the generation pathway of plasma cells and B-lymphocytes and results in their uncontrolled proliferation and malignant transformation. Consequently, it leads to the alteration of bone remodelling mechanisms, by promoting bone resorption and inhibiting bone formation [9] and thus triggers the formation of focal or diffuse bone marrow infiltration. The gold standard for observing these initial infiltration patterns is MRI [1][7][4]. Subsequently, the progression of the disease leads to the building of osseous destructions, which are observable using low-dose Computer Tomography (CT) [5]. Figure 1 illustrates the infiltration pattern of a focal lesion evolving over four examination time points of a single patient. MM evolves over a precursor state of *Monoclonal Gammopathy of Undertmined Significance* (MGUS) and developes to an asymptomatic form of the disease called *smoldering Multiple Myeloma* (sMM), which progresses to the symptomatic form of MM [4]. Thus, it is particularly important to identify sMM patients of high risk of developing MM to enable early treatment [6]. Early detection includes the tracking of image positions over time to identify early signatures of their forming and to predict infiltration patterns of future disease states. The challenges here lie first in the accurate alignment of subject whole body images, second in imaging artefacts, and subtle non rigid deformations, and third in capturing the heterogeneity of diffuse infiltration patterns and their imaging signatures. Further variability is caused by different treatment strategies and patient specific treatment responses, and progression speed.

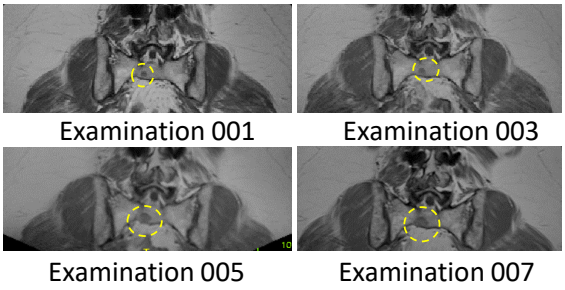


Fig. 1. Focal bone infiltration patterns are visible in MRI scans over multiple examination time points of one patient's sacrum.

A. Contribution

In this work we show and evaluate a predictor for future bone infiltration patterns oberserved in longitudinal T2 weighted MRI data. We propose a learning routine for a local predictor of lesion emergence and change, and show first results for prediction on T2 weighted MRI data. For providing predictive signatures of bone lesions, longitudinal relationships between subsequent stages of bone lesions and corresponding infiltration patterns of MM patients are assessed. The contribution of this work is three fold: (1) the longitudinal alignment of multiple bodyparts in whole body MRIs, (2) a classifier incorporating data from different disease stages in MM and (3) a probability prediction to identify bone regions evolving to diffuse or osteolytic lesions. An overview of the methodology proposed is given in Section II. The dataset and results of the evaluation are presented in Section III and this paper concludes in Section IV with an overview of possibilities for future work.

II. Methodology

In this section we summarize the processing steps for longitudinal alignment of T2 weighted MR images and the training for estimating a local lesion risk for future lesion emergence. In Figure 2 the computation pipeline proposed is visualised, which consists of a data acquisition component, data preprocessing component, a predictor training routine and a lesion risk score computation component.

[1] Department of Visual Computing and Human-centered Technologies - Computer Vision Lab, TU Wien; [2] Department of Biomedical Imaging and Image-guided Therapy - Computational Imaging Research Lab, Medical University of Vienna; [3] Department of Interventional and Diagnostic Radiology - Section Musco-Skelatal Imaging, Heidelberg University,; [4] Institute of Biomedical Engineering - Image-based biomedical modelling, Technische Universität München

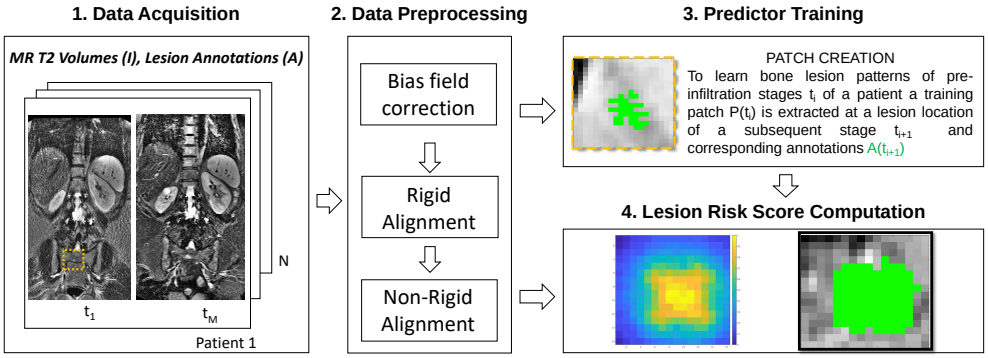


Fig. 2. Lesion Evolving Risk Computation Pipeline

A. Alignment of Longitudinal Acquisitions

The longitudinal analysis of subsequent lesion states of a subject requires precise registration of a patient's data $I_{t_i} = \{I_{t_1}, ...I_{t_M}\}$ over several examination time points t_i. In this a work a patient's image at a timepoint t_i is aligned to all subsequent timepoints $x = t_{i+1}, ..., t_M$, depending on the number of available data. We applied bias field correction before alignment using the FAST toolbox[1] [3] integrated in the FMRIB Software Library (FSL)[2]. The registration procedure is two fold: (1) Affine alignment is performed using a block matching approach for global registration (*reg_aladin*). (2) Non-rigid registration (*reg_f3d*) is performed to further transform the image of the affine registration step locally to the target at time point x. Both methodologies used are integrated in the NiftyReg toolbox[3] [8]. The performed registration offers accurate correspondences between follow-up images, which serves as basis for the extraction of imaging data depicting the development of bone infiltration and for the local lesion risk score calculation. The registered images were manually inspected if the lesions' position are in correspondence between examination time point. Figure 3 visualises a source scan of Patient 24 at examination time point 002 (left), the transformed source scan after affine registration (2nd column), the transformed source scan after affine + non-rigid registration (3rd column) and the target scan at examination time point 004 (middle).

B. Predictor Training Routine

In this study we used acquisitions from the body region of thorax, abdomen and pelvis (please cf. Section III for details regarding the dataset used). This area is considered, since most lesions occur there. For the application of a risk predictor we differentiate between the prediction of two lesion types: on the one hand lesions which *emerge* over time, i.e. which are not reported in the first scan, but in the subsequent scan, and on the other hand *growing* lesions, which are annotated in both observed examination time points. Image patches are extracted for every patient around a lesion's region longitudinally over following states. Two different patch sizes are evaluated within this work ($8 \times 4 \times 8$, $16 \times 4 \times 16$ voxels with a voxelspacing of 1.302 mm $\times$ 6 mm $\times$ 1.302 mm). Data augmentation is performed by rotating every patch in steps of 20 degrees and randomly alternating the lesion location within the patch to obtain a higher number of training data and a more variable dataset. This results in *72 different patches per lesion*. To summarize, for 53 emerging lesions we obtain 3816 patches and for 44 growing lesions 3168 patches are created. For the generation of a test and training set, lesion wise leave one out cross validation is performed in such a way, that a testset consists of the 72 patches created from one single lesion and the training set of the remaining ones, i.e. in case of emerging lesions the testset would consist of 72 patches of one lesion and the train set of 3744 patches of the 52 remaining lesions.

C. Prediction of Local Lesion Evolving Risk

In this work we demonstrate the application of predicting future lesions and mark corresponding high risk locations, by incorporating knowledge from early signatures of emerging bone lesions to train a predictor. After the registration process the aligned T2 weighted MR images $I_{t_i}(I_{t_j})$ with corresponding subsequent time points t_j of the same patient lie in the same space of the target image I_{t_j} and corresponding annotation S_j of the lesion . In a next step these obtained pairs $(I_{t_i}(I_{t_j}), S_j)$ serve as basis for patch creation and subsequent training of a random forest classifier predicting future lesion labels from the present image data. This results in a score for each voxel position V expressing the probability determined by the trained random forest for a new input image.

III. RESULTS

In this section the dataset used for evaluation is presented and qualitative and quantitative results for the evaluation of the application of machine learning using random forests to predict lesion location is discussed.

A. Study Details

Within this study 220 longitudinal whole body (wb) MRIs from 63 patients with smoldering multiple myeloma were

[1] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FAST [accessed 19th of February 2018]

[2] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSL [accessed 19th of February 2018]

[3] http://cmictig.cs.ucl.ac.uk/research/software/software-nifty/niftyreg [accessed 19th of February 2018]

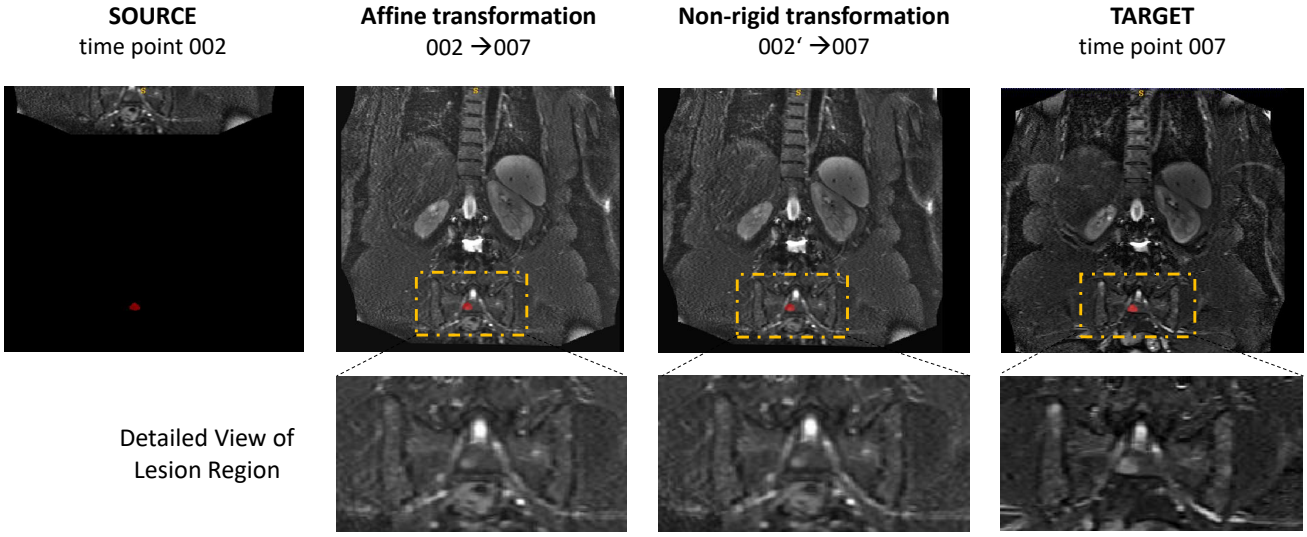


Fig. 3. Visualisation of the registration procedure of follow-up images of patient 24 from time point 002 to time point 007. The annotation of one lesion at time point 007 is visualised in red in all images in the first row. In the second row details of the lesion regions are illustrated.

TABLE I
DEMOGRAPHICS OF PARTICIPANTS

Patients	63 (39 male)
Age range (yrs)	29 -76
Median age at initial MRI (yrs)	55
Therapy	Radiation or resection
Median interval between MRIs	13 months
Median observation time	46 months

acquired between 2004 and 2011, following the 2003 guidelines [2]. At least one wbMRI was performed per patient. According to the IMWG consensus statement patients are considered to have symptomatic myeloma with the requirement of treatment, if more than one focal lesion with a diameter greater than 5 mm is present [1]. Thus, the focal lesions' annotation start at a lesion size bigger than 5 mm and is performed manually by medical experts. In Table I the demographics of the study participants is summarized. The protocol of this study was approved by the institutional ethics committee and all subjects gave their informed consent prior to inclusion. The scanning was performed on a 1.5 Tesla Magnetom Avanto (Siemens Healthineers, Erlangen, Germany) scanner. For the T2 weighted turbo-spin echo sequence (repetition time (TR):3340 ms milliseconds (ms), echo time (TE): 109 ms, section thickness (ST): 5 mm, acquisition time (TA): 2:30 min was performed of the head, thorax, abdomen, pelvis and legs using a coronal orientation. The duration of a scan was approximately 40 minutes long, no contrast medium was given.

B. Quantitative Evaluation Result of Lesion Risk Prediction

For the quantitative evaluation and for obtaining comparability between the different tested setups, the Area Under the ROC Curve (AUC) is computed, based on the probability estimates of the local lesion risk predictor for the test patch using scikit learn[4]. In Table II the mean AUC for emerging and growing lesion types are summarized. For every lesion type two different patch sizes are evaluated.

TABLE II
SUMMARY RESULTS

Lesion Type	Patch Size	Mean AUC
Emerging	8 x 4 x 8	0.904146
	16 x 4 x 16	0.8887
Growing	8 x 4 x 8	0.72949
	16 x 4 x 16	0.89803

C. Qualitative Evaluation Result of Lesion Risk Prediction

Figure 4 illustrates a prediction result for an emerging lesion. The test image (left) is a transformed image from examination time point 001 to 007 using the warping information obtain by the registration procedure introduced in Section 2. The extracted patch of this image in the region of the lesion visible in the target image I_{007} (right) is visualised in the first row in the center, with the predicted label in green and the annotation of the future lesion position extracted from image I_{007} in blue. In the second row the predicted probability map of the local lesion risk score is visualised, where orange shows regions of high probability and blue of low probability.

D. Discussion

For emerging lesions the mean AUC decreases with an increasing patch size. In contrast to this the mean AUC is increasing with increasing patch size for growing lesions. It is observable that emerging lesions achieve an approximately 0.20 higher mean AUC for the smaller patch size compared to similar mean AUC values for patches of size 16 x 4 x 16.

[4]http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html[accessed 2nd of March 2018]

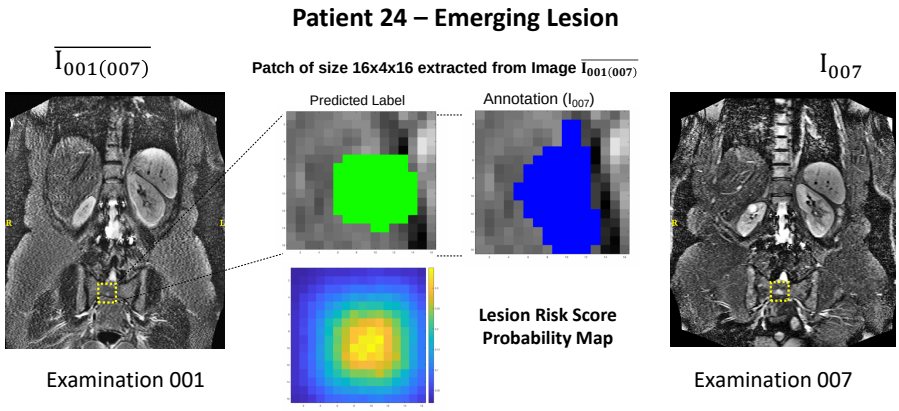


Fig. 4. Prediction of an emerging lesion from examination time point 001 to time point 007. The predicted label is visualised in green, below the underlying Local Lesion Risk Score probability map is shown and the manual annotation is visualised in blue.

IV. CONCLUSION

We presented an application of a classifier to predict a local lesion emergence risk for the analysis and visualisation of regions of high risk for bone lesions to emerge. A random forest predictor is trained using lesion image patches and annotations of subsequent lesions states of the longitudinal MR T2 weighted dataset. A challenge of this application is the accurate longitudinal alignment between images of subsequent examination time points of one patient. This is the first attempt to train a classifier to predict bone infiltration patterns in multiple myelome, while recent approaches are focusing on the detection and tracking (e.g. [10] for PET-CT) with deep learning techniques. So far the predictor is limited to image patches already located at approximate lesion locations focusing on the delineation of the lesions. We aim to adapt the proposed method to predict probability maps for entire images and different modalities. This will enable the longitudinal analysis of bone infiltration patterns caused by the progress of multiple myeloma.

ACKNOWLEDGMENTS

This work was supported by the DFG and the Austrian Science Fund (FWF) project number I2714-B31.

REFERENCES

[1] M. A. Dimopoulos, J. Hillengass, S. Usmani, E. Zamagni, S. Lentzsch, F. E. Davies, N. Raje, O. Sezer, S. Zweegman, J. Shah, A. Badros, K. Shimizu, P. Moreau, C.-S. Chim, J. J. Lahuerta, J. Hou, A. Jurczyszyn, H. Goldschmidt, P. Sonneveld, A. Palumbo, H. Ludwig, M. Cavo, B. Barlogie, K. Anderson, G. D. Roodman, S. V. Rajkumar, B. G. Durie, and E. Terpos, "Role of Magnetic Resonance Imaging in the Management of Patients With Multiple Myeloma: A Consensus Statement," *Journal of Clinical Oncology*, vol. 33, no. 6, pp. 657–664, feb 2015.

[2] B. G. M. Durie, R. A. Kyle, A. Belch, W. Bensinger, J. Blade, M. Boccadoro, J. Anthony Child, R. Comenzo, B. Djulbegovic, D. Fantl, G. Gahrton, J. Luc Harousseau, V. Hungria, D. Joshua, H. Ludwig, J. Mehta, A. Rodrique Morales, G. Morgan, A. Nouel, M. Oken, R. Powles, D. Roodman, J. San Miguel, K. Shimizu, S. Singhal, B. Sirohi, P. Sonneveld, G. Tricot, B. Van Ness, and Scientific Advisors of the International Myeloma Foundation, "Myeloma management guidelines: a consensus report from the Scientific Advisors of the International Myeloma Foundation," *The Hematology Journal*, vol. 4, no. 6, pp. 379–398, 2003.

[3] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich, and S. M. Smith, "FSL." *NeuroImage*, vol. 62, no. 2, pp. 782–90, aug 2012.

[4] J. K. Kloth, J. Hillengass, K. Listl, K. Kilk, T. Hielscher, O. Landgren, S. Delorme, H. Goldschmidt, H.-U. Kauczor, and M.-A. Weber, "Appearance of monoclonal plasma cell diseases in whole-body magnetic resonance imaging and correlation with parameters of disease activity," *International Journal of Cancer*, vol. 135, no. 10, pp. 2380–2386, nov 2014.

[5] L. Lambert, P. Ourednicek, Z. Meckova, G. Gavelli, J. Straub, and I. Spicka, "Whole-body low-dose computed tomography in multiple myeloma staging: Superior diagnostic performance in the detection of bone lesions, vertebral compression fractures, rib fractures and extraskeletal findings compared to radiography with similar radiation," *Oncology letters*, vol. 13, no. 4, pp. 2490–2494, apr 2017.

[6] M.-V. Mateos, M.-T. Hernández, P. Giraldo, J. de la Rubia, F. de Arriba, L. L. Corral, L. Rosiñol, B. Paiva, L. Palomera, J. Bargay, A. Oriol, F. Prosper, J. López, J.-M. Arguiñano, N. Quintana, J.-L. García, J. Bladé, J.-J. Lahuerta, and J.-F. S. Miguel, "Lenalidomide plus dexamethasone versus observation in patients with high-risk smouldering multiple myeloma (QuiRedex): long-term follow-up of a randomised, controlled, phase 3 trial." *The Lancet. Oncology*, vol. 17, no. 8, pp. 1127–1136, aug 2016.

[7] M. Merz, T. Hielscher, B. Wagner, S. Sauer, S. Shah, M. S. Raab, A. Jauch, K. Neben, D. Hose, G. Egerer, M.-A. Weber, S. Delorme, H. Goldschmidt, and J. Hillengass, "Predictive value of longitudinal whole-body magnetic resonance imaging in patients with smoldering multiple myeloma," *Leukemia*, vol. 28, no. 9, pp. 1902–1908, sep 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24535407http://www.nature.com/doifinder/10.1038/leu.2014.75

[8] M. Modat, D. M. Cash, P. Daga, G. P. Winston, J. S. Duncan, and S. Ourselin, "Global image registration using a symmetric block-matching approach." *Journal of medical imaging (Bellingham, Wash.)*, vol. 1, no. 2, p. 024003, jul 2014.

[9] P. Tosi, "Diagnosis and treatment of bone disease in multiple myeloma: spotlight on spinal involvement." *Scientifica*, vol. 2013, p. 104546, dec 2013.

[10] L. Xu, G. Tetteh, J. Lipkova, Y. Zhao, H. Li, P. Christ, M. Piraud, A. Buck, K. Shi, and B. H. Menze, "Automated Whole-Body Bone Lesion Detection for Multiple Myeloma on 68 Ga-Pentixafor PET/CT Imaging Using Deep Learning Methods," *Contrast Media & Molecular Imaging*, vol. 2018, pp. 1–11, jan 2018.

Volumetric Reconstruction from a Limited Number of Digitally Reconstructed Radiographs Using CNNs

Franz Thaler[1,*], Christian Payer[1] and Darko Štern[2]

***Abstract*— We propose a method for 3D computed tomography (CT) image reconstruction from 3D digitally reconstructed radiographs (DRR). The 3D DRR images are generated from 2D projection images of the 3D CT image from different angles and used to train a convolutional neural network (CNN). Evaluating with a different number of input DRR images, we compare our resulting 3D CT reconstruction to those of the filtered backprojection (FBP), which represents the standard method for CT image reconstruction. The evaluation shows that our CNN based method is able to decrease the number of projection images necessary to reconstruct the original image without a significant reduction in image quality. This indicates the potential for accurate 3D reconstruction from a lower number of projection images leading to a reduced amount of ionizing radiation exposure during CT image acquisition.**

I. INTRODUCTION

Aiming to visualize the interior body structure, computed tomography (CT) is not an invasive medical imaging technique, although it utilizes X-rays and as such, exposes the patient to radiation. Nevertheless, CT remains the dominant technique in three dimensional (3D) medical imaging due to fast acquisition and good quality of results. To visualize the interior structure of a subject, a 3D CT image is generated from a set of two dimensional (2D) X-ray images taken from different axial angles around the subject. A widely used method for 3D CT reconstruction from a set of 2D X-ray images is the filtered backprojection (FBP). By taking into account the angle from which the 2D X-ray images were acquired, FBP accumulates the backprojections of the filtered 2D X-ray images onto a 3D volume. A downside of the FBP method is that it requires a relatively high number of projections to give a reliable reconstruction, which directly correlates to the amount of radiation. Exposure to radiation increases the probability of cancer [11], which is especially problematic for applications dependent on frequent or repeated X-ray based imaging techniques.

Reducing the amount of radiation when generating 2D X-ray images lowers their quality and consequently also decreases the quality of the reconstructed 3D CT image. Different methods have been proposed to improve the quality of the reconstruction of a 3D CT image that is generated with low-dose radiation. In the approach [9] for improving the reconstruction quality, the low-dose CT sinogram data restoration is combined with an advanced edge-preserving filter in the image domain. Due to latest achievements in machine learning approaches especially with convolutional neural networks (CNN) that outperformed humans in a classification task [2], deep neural networks became also attractive for reconstruction applications. In the low-dose CNN based reconstruction method proposed in [16], the quality of the X-ray image is increased by learning to improve each low-dose ray of the image. The method [5] applies a CNN to the wavelet transform coefficients to suppress noise that is specific to low-dose CT image acquisition.

Another group of methods that reduces radiation exposure are based on beam blockers, which partially block X-rays allowing only a subset of them to reach the subject's body. The works in [1] and [8] make use of a stationary blocker for scatter suppression using a compressed sensing technique. An evaluation in respect to the number of slits and the reciprocation frequency using moving beam blockers was done in [7]. In [12] low-resolution detectors are combined with high-resolution coded apertures to achieve super-resolution. The work of [17] applied a single-slice and a multi-slice super-resolution method on low-dose CTs to improve the image quality by utilizing a CNN. Differently from the blocker based methods, where the same number of X-ray projections is used, the same amount of radiation can be reduced without blocking the X-ray bins but decreasing the number of X-ray images used in the reconstruction of the CT image. However, these few-view CT images are heavily burdened by artifacts when FBP is used for reconstruction. The work of [3] proposed a gradient-based dictionary learning algorithm for CT reconstruction from a reduced number of views, using the vertical and horizontal gradient images as input. In the approach [18] a CNN is used to improve the quality of a few-view reconstructed images by learning it's mapping to a full-view reconstruction.

An extreme case of CT reconstruction from a low number of X-ray images can also be found in 3D/2D image registration. As explained in [10] one approach for 3D/2D reconstruction is the intensity-based approach that aims to reconstruct the inter-operative 3D CT image from as few as possible X-ray images from different views. Namely, during minimally invasive surgeries it is required to exactly locate the instruments in use within the patient's body. To accomplish this, a high quality 3D CT image of a patient is acquired pre-operatively and registered with a single or multiple X-ray images from different directions that are generated inter-operatively. This is done repeatedly during surgery, exposing not only the patient but also the medical

*This work was supported by the Austrian Science Fund (FWF): P28078-N33.

[1]Franz Thaler and Christian Payer are with the Institute of Computer Graphics and Vision, Graz University of Technology, Austria f.thaler@student.tugraz.at

[2]Darko Štern is with the Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria darko.stern@cfi.lbg.ac.at

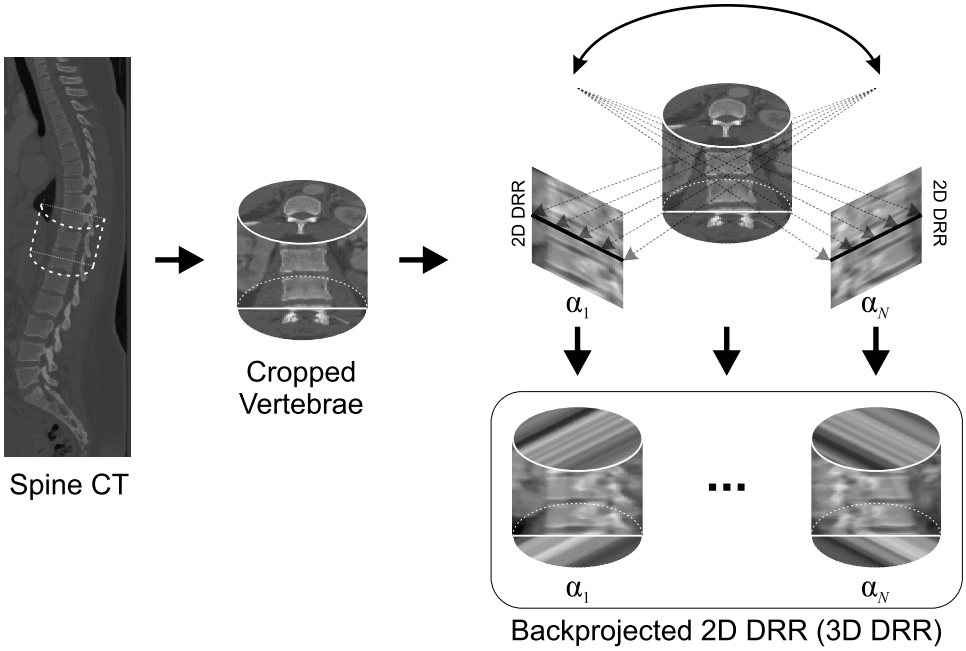


Fig. 1: Generation of digitally reconstructed radiographs (DRR) used in CNN based 3D reconstruction.

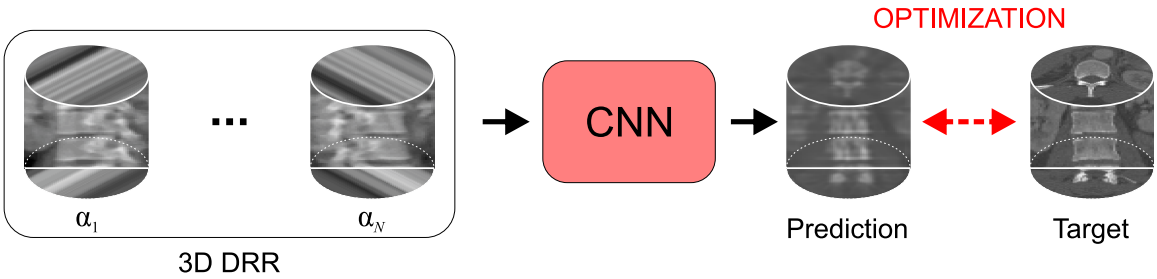


Fig. 2: Reconstruction of a 3D CT vertebra image from a limited number of backprojected 2D DRR images (3D DRR) using a U-Net based CNN.

staff to radiation.

A similar problem of reconstructing a 3D image given one or multiple 2D images is also seen in the computer vision community. The method in [14] utilizes a supervised learning approach to reconstruct depth information from a single RGB image. Resulting in a volumetric binary image, the method [15] based on a CNN uses a RGB image and depth information to recover a 3D shape of the scene. Multi-view reconstruction methods utilize multiple 2D input images from different views to reconstruct the surface of the scene in a 3D volumetric representation. For example the SurfaceNet introduced in [4] using a 3D CNN does not only use multiple images, but similar to CT reconstruction also the corresponding information of the angle from which each image is taken.

In this paper we propose a method based on a CNN for 3D CT image reconstruction from a limited number of 2D projection images. Our approach utilizes a framework that generates 2D digitally reconstructed radiographs (DRR) from an arbitrary direction and uses them to train a CNN for reconstructing the original 3D image. We conducted the experiments with a different number of DRR images and compare the reconstruction results with the FBP method. We show that by using a machine learning based approach the number of images required for the reconstruction can be reduced without significant decrease in performance. The results indicate that our approach has a potential to be used for accurate 3D reconstruction from a lower number of views, thus reducing the amount of ionizing radiation.

II. METHOD

In our method we generate 2D DRR images from different angles (Fig. 1) to be further used for training a CNN to reconstruct the original 3D CT image (Fig. 2). Due to their complex shape, which is challenging for reconstruction, we used spine images including several vertebrae for 3D CT reconstruction. These images are cropped from whole spine CT images and brought to the canonical position. DRR images are generated as a sum projection from the volumetric images in different angles. When generating DRR images the volumetric images are augmented by translation, rotation and scaling. Each generated DRR image is backprojected to a volume and used as input for training the U-Net based CNN

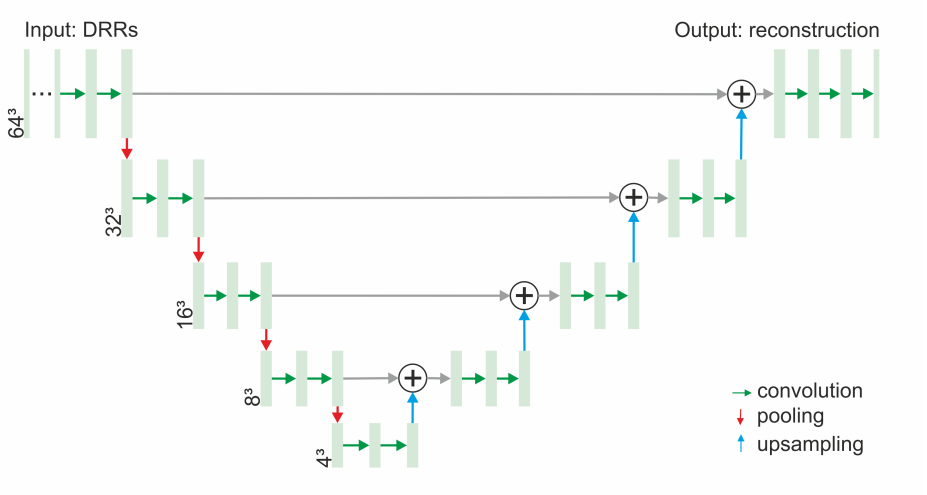


Fig. 3: Network architecture.

[13]. The reconstruction results of the CNN are quantitatively compared to the FBP approach.

The generation of the cropped 3D CT images of the vertebrae is explained in subsection II-A and the generation of DRR images in subsection II-B. Our network architecture is described in subsection II-C. A short overview of the FBP approach is presented in subsection II-D.

A. 3D CT Vertebra Image Generation

In our approach for 3D reconstruction from a reduced number of views, we used 3D CT images of vertebrae and their surrounding structure cropped from a whole spine CT as visualized in Fig. 1. Since each vertebra has a different orientation in the 3D spine image and to simplify the reconstruction task, we bring all vertebrae in the cropped 3D CT image to a canonical position, which is centered at the vertebral body's center, and defined by three orthogonal vectors representing the three dimensions. To find the position and orientation of the vertebra in the original spine image, we used two predefined points, i.e. a point in the center of the vertebra and at the tip of the spinous process. The first vector of the vertebra's orientation corresponds to the tangent vector of the polynomial curve connecting the centers of the vertebral bodies. The second vector corresponds to the direction of the spinous process' tip and the third vector is defined by the right hand rule. Based on the position and the orientation of each vertebra in the spine image, we cropped a cube that captures the vertebra and it's surrounding structures.

B. 2D DRR Image Generation

In our approach we used 2D DRR images generated for reconstructing the cropped 3D CT vertebra image as explained in the previous subsection. The DRR images are generated as a sum projection of the cropped 3D CT vertebra image from an angle lying on the mid-axial plane of the volumetric image as shown in Fig. 1. We experimented with a different number N of 2D DRR images uniformly distributed with fixed angles around the axial plane. Since each projection image for angle α_n, $n = 1,\ldots,N$ is identical to the projection image for angle $\alpha_n + 180°$, we only generate DRR images from angles in the range of $0°$ to $180°$.

C. Network Architecture

Our network architecture is based on the U-Net introduced in [13] and visualized in Fig. 3. Our CNN with volumetric kernels has a set of N 3D DRR images as an input (Fig. 2). Each 3D DRR image corresponds to a single 2D DRR image backprojected to a volume of the same size as the original 3D vertebra image (Fig. 1). A 3D DRR image is created by repeating the 2D DRR image in the volume shifted by the angle α_n that corresponds to the direction the 2D DRR image was acquired from.

Our CNN architecture is defined as follows: for each level in the contracting path, two consecutive convolution layers are used while a subsequent average pooling layer creates the input for the next lower level. When the maximum number of levels is reached, the two convolutions are followed by an upsampling layer. This upsampling layer is then merged with the convolution layer output of the contracting path of the same level by utilizing an add layer. In the expanding path, every upsampling and add layer is followed by two convolution layers until the original size of the image is reached. After that, a final convolution with output size one generates the prediction corresponding in size to the 3D vertebra image. To obtain the CNN parameters ω we used L_1 loss between all voxels $m \in M$ of the predicted image $\hat{p}$ and the 3D vertebra image p:

$$\hat{\omega} = \arg\min_{\omega} \frac{1}{m} \sum_{m \in M} |\hat{p}_m(\omega) - p_m|. \quad (1)$$

D. Filtered Backprojection

We compare our results with the standard approach used in CT reconstruction, represented by the FBP. Since simply summing up the backprojected 2D DRR images, i.e. the 3D DRR images, gives a blurry reconstruction of the original

3D image, the FBP utilizes a ramp filter R to reduce the contribution of low frequencies in Fourier space $\mathscr{F}$ before summation. Thus, before doing backprojection, each ray of the 2D DRR images $I_j \in I$ is transformed to the frequency domain by utilizing the Fourier transform and is multiplied with a ramp filter. After applying the inverse Fourier transform $\mathscr{F}^{-1}$, the filtered rays $\hat{I}_j$ are returned to their original position in the 2D DRR image $\hat{I}$:

$$\hat{I}_j = \mathscr{F}^{-1}(R \cdot \mathscr{F}(I_j)). \tag{2}$$

III. EVALUATION

A. Material

The data used in this work encompasses CT scans of the spine of 10 different patients from which we extracted all present vertebrae. The spine CTs vary in size, spacing and vertebrae contained within them. The volumes have a size of $512 \times 512 \times K$, where K ranges from 507 to 625. The number of vertebrae included in the volumes ranges from 17 to 19, giving us a total of 176 vertebrae. We separated the CT images into a training and a test set in the ratio of 80 to 20, i.e. we used eight spines for training and two spines for testing. As a result, the training set encompasses 141 vertebrae and the test set 35.

As a first step, all 3D CT spine images are transformed to have an equal size. Since the spacing parameter takes care of the real world mapping, which keeps the image ratios intact, rescaling the 3D CT images does not lead to any distortion. Also, since the CT spine images have a high resolution for being 3D, we also downsample the spine images to $256 \times 256 \times 256$, which is approximately half the size in each dimension. The CT images are downsampled with tricubic interpolation and then stored to the hard drive and used for any further processing. By utilizing this downsampling, image loading is accelerated and the memory consumption is reduced.

B. Augmentation

For the CNN to be successfully trained, a training data set that consists of target 3D vertebra images and corresponding single- or multi-view input 3D DRR images has to be increased. Therefore, we utilized online augmentation, which performs a translation, rotation and scaling when cropping the 3D vertebra images from the original CT spine images. We performed a translation by dislocating the center of the vertebral body in all three dimensions and a rotation by adding an offset to the angle of each of the orthogonal vectors defining the canonical position of the vertebra. In contrast to translation and rotation, scaling is performed uniformly in all three dimension to prevent a distortion of anatomical structures. We did not argument the angle α_n of the projection 2D DDR images.

C. Experimental Setup

Our hardware setup consisted of a CPU Intel Core i7-930 @ 2.80GHz, 24 GB RAM and a GeForce GTX TITAN X with 12 GB. We implemented our network in Keras[1] using TensorFlow[2] as it's backend, volumetric image processing was done utilizing the ITK framework[3].

The target 3D vertebra images we used as ground truth have a size of $64 \times 64 \times 64$ voxels and their size in physical space is set to 120 mm per dimension. For online 3D vertebra image augmentation we set the translation range to 15 mm, the rotation range to 30° and the scaling percentage to 15%. We utilized a uniform distribution to generate the augmentation parameters and the order of execution is rotation, translation and scaling.

Due to the different amount of image information projected from different angles when generating the 2D DRR images from the 3D vertebra image cropped as a cube, we introduced the same cylindric mask to all the 3D cropped vertebra images before generating 2D DRR images. Additionally, when calculating the loss function in Eq. (1), only the pixels M inside the cylinder are taken into account.

For the implementation of the FBP we utilized the open source library scikit-image[4] on our data.

We trained our network with a mini-batch size of one for 200 epochs, each of them used 200 iterations, resulting in a total of 40.000 samples used for training. As a loss function we utilized L_1 given in Eq. (1), a weight regularization was done with L_2 and a factor of 0.0005. As an optimizer we used Adam [6], the learning rate was set to 0.0002, the first and second moment estimates are defined as $\beta_1 = 0.9$ and $\beta_2 = 0.999$. All convolution layers are defined as volumetric convolutions, we used zero padding and as kernel initializer we utilized He normal [2]. For all convolutions except the final one, we used a kernel size of $3 \times 3 \times 3$, 64 filters and ReLU as activation function. The final convolution's kernel size was set to $1 \times 1 \times 1$, we used just one filter and no activation function. Furthermore, we used 3D average pooling and 3D nearest neighbor upsampling layers with a kernel size of $2 \times 2 \times 2$.

IV. RESULTS

We train individual networks for different numbers N of 3D DRR input images and compare the predicted 3D reconstruction quantitatively and qualitatively to the results of the FBP. Fig. 4 and Table I show the mean absolute error to the target 3D vertebra images used as ground truth for both methods respectively. The center slice of the ground truth of one 3D vertebra image as well as our qualitative results and those from the FBP are presented in Fig. 5 and 6. All images represent the reconstruction of the same 3D vertebra image using a different number of views, $N \in \{1, 2, 3, \ldots, 120, 180\}$. All vertebra image slices included in our qualitative results correspond to one another by having the exact same center voxel. Furthermore, the brightness setting is identical for all vertebra image slices, however, for better contrast some values are truncated. This is especially true for the FBP

[1] https://keras.io/
[2] https://www.tensorflow.org/
[3] https://itk.org/
[4] http://scikit-image.org/

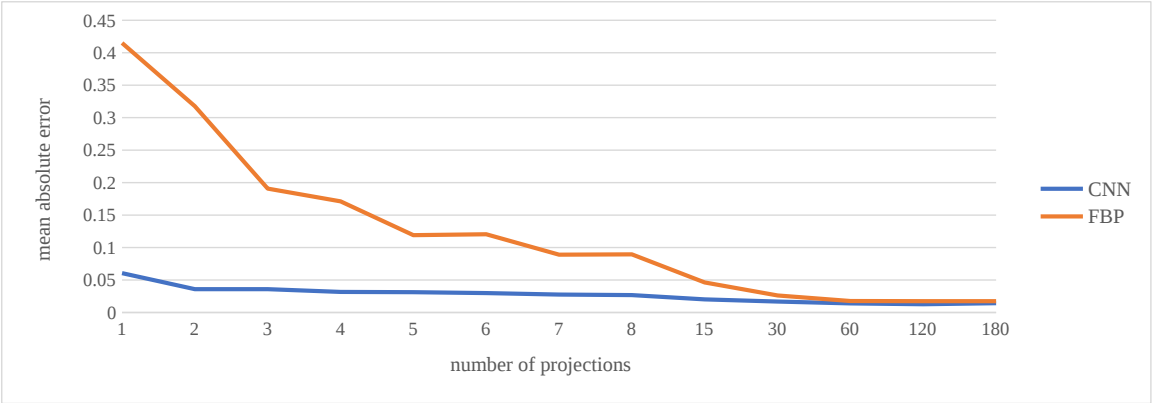


Fig. 4: Mean absolute error of our CNN and FBP to ground truth for a different number of projection images (N).

results using only very few views, since they show a large deviation in color value.

TABLE I: Mean absolute error ± standard deviation of our CNN and FBP to ground truth.

Projections (N)	CNN (10^{-2})	FBP (10^{-2})
1	6.06 ± 2.24	41.50 ± 7.39
2	3.59 ± 0.80	31.77 ± 5.43
3	3.59 ± 0.72	19.07 ± 3.32
4	3.17 ± 0.58	17.11 ± 2.87
5	3.12 ± 0.58	11.90 ± 1.88
6	2.99 ± 0.63	12.04 ± 2.07
7	2.76 ± 0.51	8.90 ± 1.43
8	2.68 ± 0.53	8.95 ± 1.46
15	2.02 ± 0.38	4.63 ± 0.73
30	1.69 ± 0.33	2.62 ± 0.37
60	1.40 ± 0.26	1.76 ± 0.23
120	1.27 ± 0.21	1.74 ± 0.23
180	1.43 ± 0.30	1.74 ± 0.23

V. DISCUSSION

To visualize the interior body structure CT imaging utilizes a number of X-ray images captured from different axial angles. Involving a large number of X-ray images increases the ionizing radiation not only to the patient but also to the medical staff involved in the image acquisition. Reducing the number of views from which X-ray images are generated leads to a significant decrease in the quality of the 3D CT image, when reconstructing with the standard FBP method. In this work we investigate the potential of a machine learning based approach to improve the quality of the reconstructed images when the number of views is limited. Inspired by the previous work in [4] that comes from the computer vision community, we constructed a framework that reconstructs the volumetric image based on multi-view 2D images. Compared to [4] that use camera images with a higher number of views to reconstruct the surface of an outdoor scene, in our approach the 3D CT image is reconstructed from a sparse number of 2D projection images. Differently from [16], where high quality single axial CT reconstruction was done from the low quality 2D image obtained by accumulating few-view backprojections, our CNN based method reconstructs the 3D CT image directly from the backprojected DRR images.

When utilizing deep CNNs, a significant number of training data is required, which in our scenario of CT reconstruction would require access to a large set of X-ray projection images used for CT reconstruction. Therefore, in this paper we investigate the possibility of using DRR images as a substitution for the real X-ray images, thus, showing that CNNs combined with 2D DRR images can be used for reconstructing 3D CT images. We followed the standard backprojection procedure used in CT reconstruction but replaced the backprojection of the X-ray images with the backprojected DRR images, i.e. 3D DRR images. CNNs are then trained to compensate the missing information coming from omitted backprojected X-ray images.

The quantitative results in Fig. 4 and Table I show that our method performs better than the FBP for a small number of views and almost the same when a large number of DRR images is used. Thus, the quality of the 3D reconstruction becomes almost the same for both methods when using 60 3D DRR images as input. However, as seen in Fig. 5 and 6, when using 30 3D DRR images from different angles our method already provides good qualitative results with only a small amount of artifacts present in the 3D CT reconstruction, whereas the FBP still suffers from a lot of artifacts. Moreover, our method is able to visualize all important structures of the vertebra in the reconstruction of the 3D image by utilizing only 15 different views. Using only two 3D DRR images, the silhouette of the vertebra reconstructed by our method can be recognized especially in sagittal view, while the FBP method requires eight views for a similar quality of the reconstructed images. For a single 3D DRR input image, neither our nor the FBP method managed to produce useful results.

By showing a better quality than the FBP method when using a small number of DRR images to reconstruct a 3D CT image, the results indicate that our method can be used to reduce the number of X-ray images to reconstruct 3D CT images in real world scenarios. Thus, our method shows that by utilizing a CNN it is possible to reduce the overall amount of radiation during 3D reconstruction. Reducing the exposure

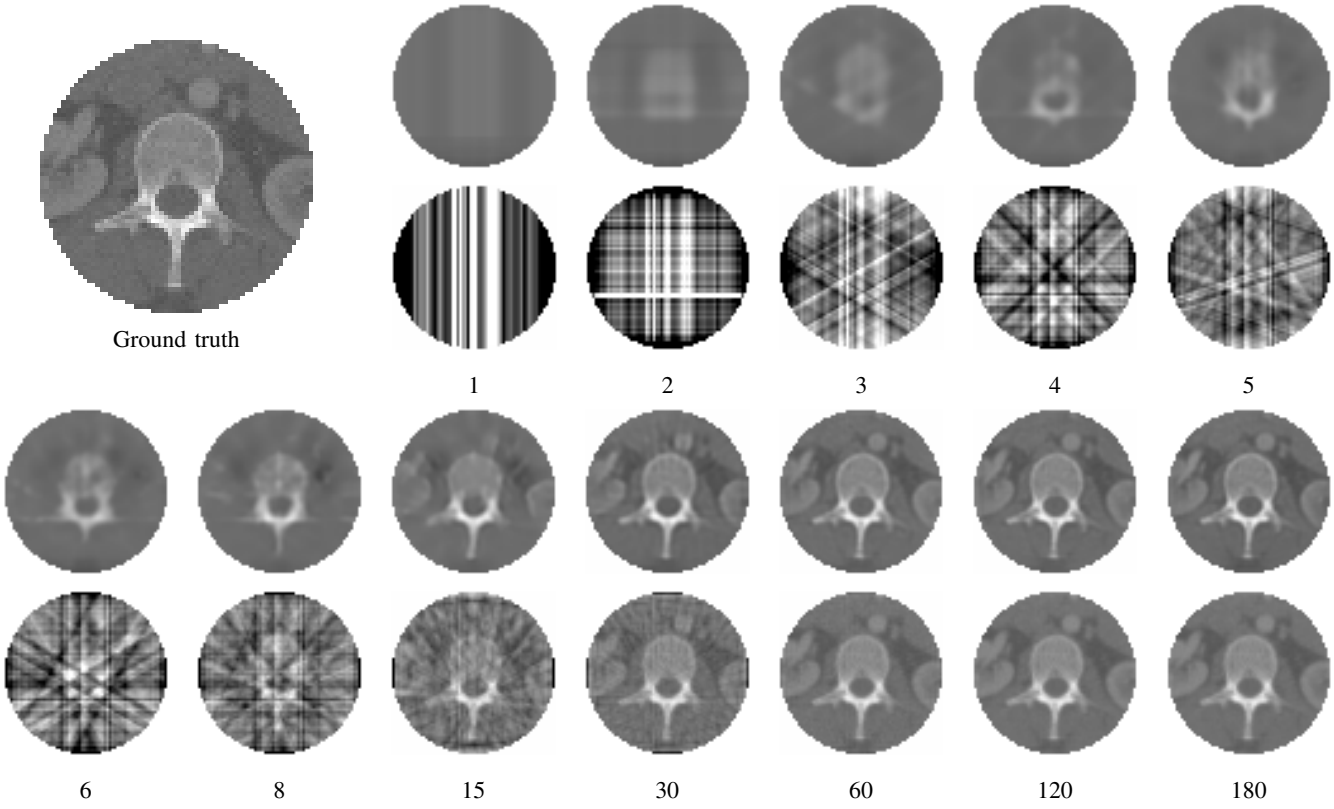


Fig. 5: Qualitative results of axial slices for different number of projection views comparing our method (top images) with FBP (bottom images). Ground truth is shown on top left.

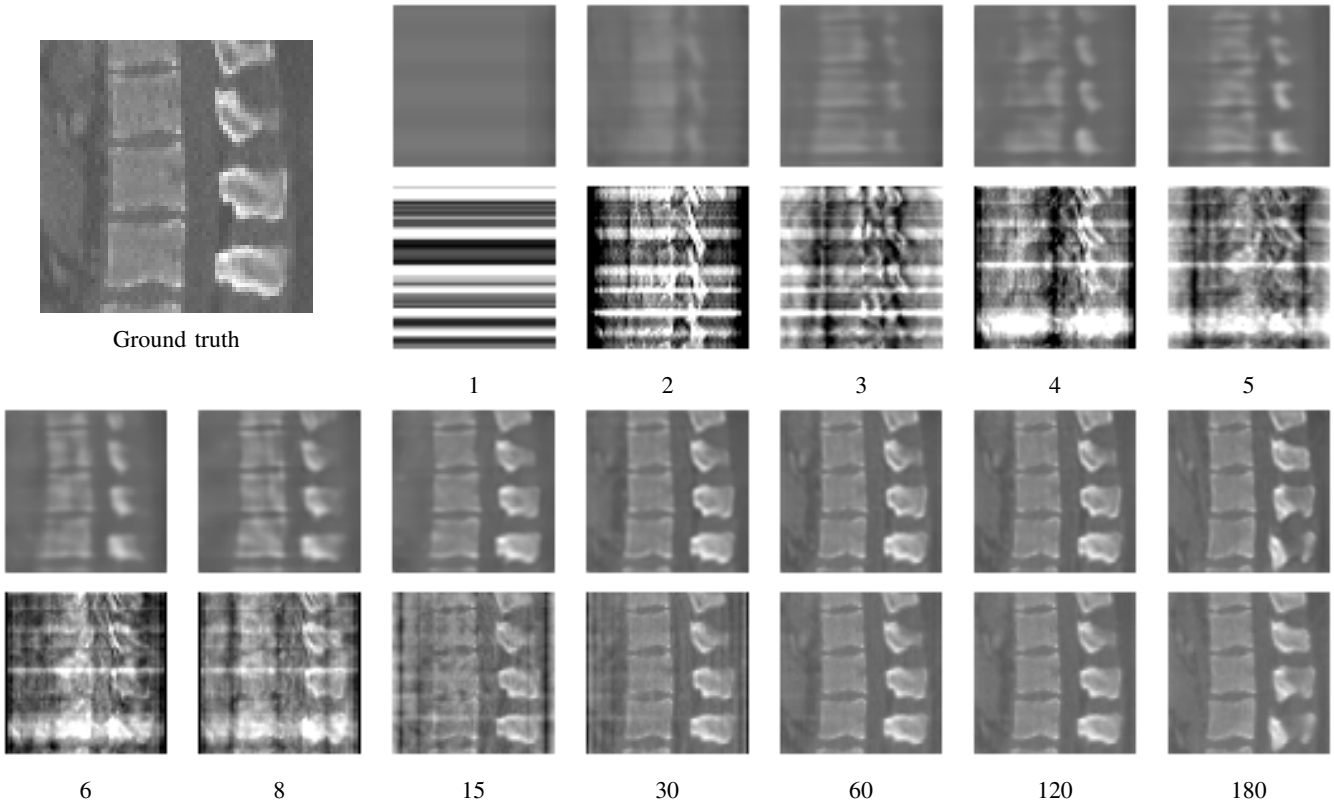


Fig. 6: Qualitative results of sagittal slices for different number of projection views comparing our method (top images) with FBP (bottom images). Ground truth is shown on top left.

to ionizing radiation for patients during CT image acquisition is especially beneficial when patients are subject to frequent examinations. Another beneficial application of our method could be for a scenario, when it is unfavorable or not feasible to acquire a large amount of views. An example of such a case is 3D/2D registration during minimally invasive and image guided surgeries, where 3D reconstruction from a limited number of X-ray images decreases the exposure of both patients and medical staff to ionizing radiation.

VI. CONCLUSION

In this paper we proposed a method for multi-view 3D CT image reconstruction from 3D DRR images using machine learning. Our method improves the quality of the reconstructed 3D CT image compared to the standard FBP by utilizing a CNN for a small number of views and gives similar results for a high number of views. By reducing the number of 3D DRR images required for a 3D CT reconstruction, our method indicates the possibility to decrease the amount of necessary X-ray images in real world scenarios. Thus, it is possible to reduce the amount of ionizing radiation exposed to the patient and the medical staff during image acquisition for examination and surgery. In our future work, we aim to further improve the quality of the results as well as to evaluate the performance of our approach on different datasets.

REFERENCES

[1] X. Dong, M. Petrongolo, T. Niu, and L. Zhu, "Low-dose and scatter-free cone-beam CT imaging using a stationary beam blocker in a single scan: phantom studies," *Computational and mathematical methods in medicine*, vol. 2013, 2013.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[3] Z. Hu, Q. Liu, N. Zhang, Y. Zhang, X. Peng, P. Z. Wu, H. Zheng, and D. Liang, "Image reconstruction from few-view CT data by gradient-domain dictionary learning," *Journal of X-ray science and technology*, vol. 24, no. 4, pp. 627–638, 2016.

[4] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "Surfacenet: An end-to-end 3D neural network for multiview stereopsis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2307–2315.

[5] E. Kang, J. Min, and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction," *Medical physics*, vol. 44, no. 10, 2017.

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations 2015*, pp. 1–15, 2015.

[7] T. Lee, C. Lee, J. Baek, and S. Cho, "Moving beam-blocker-based low-dose cone-beam CT," *IEEE Transactions on Nuclear Science*, vol. 63, no. 5, pp. 2540–2549, 2016.

[8] W. Liu, J. Rong, P. Gao, Q. Liao, and H. Lu, "Algorithm for X-ray beam hardening and scatter correction in low-dose cone-beam CT: Phantom studies," in *Medical Imaging 2016: Physics of Medical Imaging*, vol. 9783. International Society for Optics and Photonics, 2016, p. 978332.

[9] J. Ma, J. Huang, Z. Liang, H. Zhang, Y. Fan, Q. Feng, and W. Chen, "Image fusion for low-dose computed tomography reconstruction," in *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2011 IEEE*. IEEE, 2011, pp. 4239–4243.

[10] P. Markelj, D. Tomaževič, B. Likar, and F. Pernuš, "A review of 3D/2D registration methods for image-guided interventions," *Medical image analysis*, vol. 16, no. 3, pp. 642–661, 2012.

[11] D. L. Miglioretti, E. Johnson, A. Williams, R. T. Greenlee, S. Weinmann, L. I. Solberg, H. S. Feigelson, D. Roblin, M. J. Flynn, N. Vanneman, *et al.*, "The use of computed tomography in pediatrics and the associated radiation exposure and estimated cancer risk," *JAMA pediatrics*, vol. 167, no. 8, pp. 700–707, 2013.

[12] E. Mojica, S. Pertuz, and H. Arguello, "High-resolution coded-aperture design for compressive X-ray tomography using low resolution detectors," *Optics Communications*, vol. 404, pp. 103–109, 2017.

[13] 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.

[14] A. Saxena, S. H. Chung, and A. Y. Ng, "3-D depth reconstruction from a single still image," *International journal of computer vision*, vol. 76, no. 1, pp. 53–69, 2008.

[15] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[16] X. Yang, V. De Andrade, W. Scullin, E. L. Dyer, N. Kasthuri, F. De Carlo, and D. Gürsoy, "Low-dose X-ray tomography through a deep convolutional neural network," *Scientific reports*, vol. 8, no. 1, p. 2575, 2018.

[17] H. Yu, D. Liu, H. Shi, H. Yu, Z. Wang, X. Wang, B. Cross, M. Bramler, and T. S. Huang, "Computed tomography super-resolution using convolutional neural networks," in *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3944–3948.

[18] J. Zhao, Z. Chen, L. Zhang, and X. Jin, "Few-view CT reconstruction method based on deep learning," in *Nuclear Science Symposium, Medical Imaging Conference and Room-Temperature Semiconductor Detector Workshop (NSS/MIC/RTSD), 2016*. IEEE, 2016, pp. 1–4.

Unsupervised Identification of Clinically Relevant Clusters in Routine Imaging Data

Johannes Hofmanninger[1] Markus Krenn[1] Markus Holzer[1] Thomas Schlegl[1] Helmut Prosch[1] Georg Langs[1*]

***Abstract*— This abstract is a summary of [3]. Currently, computational image analysis typically relies on well annotated and curated training data. While these kind of data sets enable the creation of accurate and sensitive detectors for specific findings, they are limited, since annotation is only feasible on a relatively small number of cases. We propose unsupervised learning to group patients based on non-annotated clinical routine imaging data. We show that based on learned visual features, we identify population clusters with homogeneous (within clusters) but distinct (across clusters) clinical findings. To evaluate the link between visual clusters and clinical findings, we compare clusters with corresponding radiology report information extracted with natural language processing algorithms.**

I. IDENTIFICATION OF CLUSTERS

Spatial Normlization We perform spatial normalization to establish spatial correspondences of voxels across the population. For this purpose, we employ a multi-template spatial normalization algorithm that is able to deal with the high variability present in routine imaging data [4].

Feature Extraction We extract features that capture complementary visual characteristics in order to map an image to a visual descriptor representation. We densely sample Haralick [2] features to encode rotation invariant texture and *Shape Features* (3D-SIFT [5]) to encode rotation variant gradient changes. Subsequently we quantize these features to *Bag of visual Words* to summarize local features to global volume descriptors. In advance, we augment the features with their spatial position in the reference space. Finally, we learn a set of 20 latent topics of co-occurring feature settings by using *Latent Dirichlet Allocation* (LDA) [1]. This allows to interpret an image as a mixture of topics represented by its 20 dimensional topic assignment vector.

Clustering We perform clustering of the population to retrieve groups of subjects with (visually) similar properties. Here we interpret the Euclidean distance between two volume descriptors as a measure of visual similarity.

II. EVALUATION

Clustering is performed on the full set of images, while for evaluation only records with a report are considered. Reports are processed by a natural language processing pipeline mapping free text to a set of pathology terms. Aim of the evaluation is to test the hypothesis, that the clustering reflects pathological subgroups in the population. In order to do so we test whether volume label assignments (pathology terms) are associated with cluster assignments. A cell-χ^2-test is performed for each term and each cluster to test whether its cluster frequency is significantly different from its population frequency.

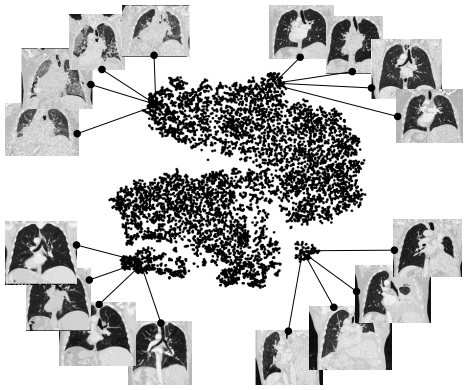


Fig. 1. T-SNE embedding illustrating the routine imaging landscape

III. RESULTS

We discovered more than 250 (positive and negative) associations between clusters and terms. We find that combining complementary features improves clustering compared to individual feature sets and show that learning latent topics of commonly occuring feature classes furthermore improve results. We demonstrate that visual features extracted from the lungs have prognostic power for numerous pathological findings.

[1]Department of Biomedical imaging and Image-guided Therapy Computational Imaging Research Lab, Medical University of Vienna, Austria www.cir.meduniwien.ac.at

*This research was supported by teamplay which is a Digital Health Service of Siemens Healthineers, by the Austrian Science Fund, FWF I2714-B31, and WWTF S14-069.

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[2] R. M. Haralick, K. Shanmugam, *et al.*, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[3] J. Hofmanninger, M. Krenn, M. Holzer, T. Schlegl, H. Prosch, and G. Langs, "Unsupervised identification of clinically relevant clusters in routine imaging data," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 192–200.

[4] J. Hofmanninger, B. Menze, M.-A. Weber, and G. Langs, "Mapping multi-modal routine imaging data to a single reference via multiple templates," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham: Springer International Publishing, 2017, pp. 341–348.

[5] M. Toews and W. M. Wells, "Efficient and robust model-to-image alignment using 3d scale-invariant features," *Medical image analysis*, vol. 17, no. 3, pp. 271–282, 2013.

Poster Session

Generative Adversarial Networks to Synthetically Augment Data for Deep Learning based Image Segmentation*

Thomas Neff[1], Christian Payer[1], Darko Štern[2,1], Martin Urschler[2,1]

Abstract— **In recent years, deep learning based methods achieved state-of-the-art performance in many computer vision tasks. However, these methods are typically supervised, and require large amounts of annotated data to train. Acquisition of annotated data can be a costly endeavor, especially for methods requiring pixel-wise annotations such as image segmentation. To circumvent these costs and train on smaller datasets, data augmentation is commonly used to synthetically generate additional training data. A major downside of standard data augmentation methods is that they require knowledge of the underlying task in order to perform well, and introduce additional hyperparameters into the deep learning setup. With the goal to alleviate these issues, we evaluate a data augmentation strategy utilizing Generative Adversarial Networks (GANs). While GANs have shown potential for image synthesis when trained on large datasets, their potential given small, annotated datasets (as is common in e.g. medical image analysis) has not been analyzed in much detail yet. We want to evaluate if GAN-based data augmentation using state-of-the-art methods, such as the Wasserstein GAN with gradient penalty, is a viable strategy for small datasets. We extensively evaluate our method on two image segmentation tasks: medical image segmentation of the left lung of the SCR Lung Database and semantic segmentation of the Cityscapes dataset. For the medical segmentation task, we show that our GAN-based augmentation performs as well as standard data augmentation, and training on purely synthetic data outperforms previously reported results. For the more challenging Cityscapes evaluation, we report that our GAN-based augmentation scheme is competitive with standard data augmentation methods.**

I. INTRODUCTION

Modern machine learning methods currently revolutionize our daily life. Especially *deep learning* [9] based methods consistently show improvements in the state-of-the-art every year, and for many *computer vision* tasks they even surpass human performance [6]. However, what most of these methods have in common is that they are *supervised*, therefore requiring *annotated* data for training. Furthermore, most deep learning methods benefit from a large amount of data to train, often in the range of hundreds of thousands of images. To circumvent the issue of insufficient annotated training data, common approaches such as *transfer learning* [3], *domain adaptation* [3] and *data augmentation* [13] can be followed. While transfer learning and domain adaptation are very popular, they are not as easily applicable for tasks where no large public datasets or pre-trained network parameters of a close domain are available, e.g. in medical image analysis. For this reason, we focus on data augmentation to deal with small amounts of data, in particular data augmentation using images synthesized from a generative model.

While data augmentation is most commonly done by using simple transformations, more sophisticated approaches for synthesizing additional training data have been proposed as well. For example, it has been shown that by rendering photorealistic, synthetic images and performing a set of transformations on those rendered images, they can be used to train an object detector with good performance [14]. Similarly, in the medical domain, it has been shown that by training a deep neural network on high-quality rendered 3D images from other computer vision tasks and fine-tuning it towards medical data, the general network performance can be improved when data is scarce [12]. This shows that data augmentation by using a generative model can improve the training of deep learning methods.

Recently introduced by Goodfellow et al., Generative Adversarial Networks (GANs) provide an attractive method of learning a generative model by training a deep neural network [4]. GANs have demonstrated potential in tasks such as state-of-the-art image generation [7], or synthetic data generation [16], [10]. The idea of using GANs in the context of data augmentation also saw some advancements in research, for example with the *SimGAN* [16] architecture. The main idea of SimGAN is to render synthetic images with corresponding labels (e.g. images of human eyes with their corresponding gaze direction) and refine those synthetic images with a *refiner*-GAN. This GAN uses the information from real, unlabeled images of the same domain while preserving the label information of the rendered images to generate realistic, refined images, which can further be used as training data for a supervised deep network. However, while GANs show impressive results when trained on large datasets, it is still a topic of active research how GANs behave when trained on a small amount of data, as most GAN-related research focuses on large datasets.

For this work, we will focus on data augmentation methods in the context of *image segmentation* tasks. As segmentation is a pixel-wise problem, the acquisition of annotated segmentation masks is even more time consuming, compared to e.g. classification tasks, as a human annotator has to label every pixel manually, which makes automated annotation methods highly desirable. Building upon the architecture we proposed in 2017 [10], we present a GAN-based data augmentation strategy incorporating state-of-the-art methods

*This work was supported by the Austrian Science Fund (FWF): P 28078-N33.

[1]Thomas Neff, Christian Payer, Darko Štern and Martin Urschler are with the Institute of Computer Graphics and Vision, Graz University of Technology, Austria thomas.neff@student.tugraz.at

[2]Darko Štern and Martin Urschler are with the Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria martin.urschler@cfi.lbg.ac.at

of GAN optimization, such as the Wasserstein GAN with gradient penalty (WGAN-GP) [5]. Our goal is to evaluate the segmentation performance of GAN-based data augmentation compared to standard data augmentation methods, especially in the case of small datasets. We perform experiments on two segmentation tasks, one from medical imaging, i.e. X-ray lung segmentation of the *SCR Lung Database* [17], and another, more challenging one, from computer vision, i.e. urban scene understanding of the *Cityscapes* [2] dataset. Additionally, we compare the segmentation performance when training with different ratios of real and generated data, to further evaluate the impact of GAN samples on the training process.

II. Related Work

A. Data Augmentation

Data augmentation is the process of generating additional training data from the available existing data [3]. Typically, this is done by using annotation-preserving transformations on the input data, such as randomly rotating, translating or deforming the image. Through the random nature of data augmentation, it can be used to potentially generate an 'infinite' amount of training data by augmenting the already existing data. For medical image analysis, data augmentation such as elastic deformation has been used with much success in combination with convolutional neural networks, as demonstrated by the *U-Net* [13] architecture for medical image segmentation. Although data augmentation is an effective way of dealing with the issue of small amounts of training data, it is not universally applicable, as prior knowledge of target domain and task is required to find a good data augmentation. Furthermore, the parametrization of data augmentation methods introduces another set of important hyperparameters, which can have a significant impact on the error made by the deep learning method.

B. Generative Adversarial Networks

Due to their end-to-end nature, good generated image quality and compatibility with modern deep learning techniques, GANs are the current state-of-the-art for generative models. A GAN consists of two subnetworks: the *generator* G, and the *discriminator* D, which play against each other in a two-player minimax game [4]. The generator synthesizes data from an input noise vector $\mathbf{z} \sim p_z$. The discriminator is a standard classification network, which receives real data $\mathbf{x} \sim p_R$, as well as data from the generator $G(\mathbf{z})$ as input. The goal of the discriminator is to perfectly classify each input image as either *real* or *synthetic*, while the goal of the generator is to synthesize images as close as possible to the real data, which leads to the discriminator misclassifying real and generated data [4].

As GAN optimization requires finding a Nash Equilibrium [4] between the generator and discriminator, GANs have historically been very unstable to train, which led to a lot of research focused on improving their training stability and generated image quality. Especially WGAN-GP [5] enjoys large popularity, due to being stable across different datasets and network architectures, as well as producing high quality images. The main idea behind WGAN is to use the Wasserstein-1 distance as its optimization criterion, which intuitively computes the cost of the optimal transport plan to transform the real data distribution p_R to the generator distribution p_G. Further improving on WGAN, WGAN-GP approximates the Wasserstein-1 by using a soft penalty on the gradient norm of the discriminator/critic. For this gradient norm, Gulrajani et al. [5] sample uniformly from a distribution $p_{\hat{\mathbf{x}}}$ that is defined along the lines between pairs of samples from the real data distribution p_R and the model distribution p_G, leading to the following optimization function:

$$L = \underbrace{\mathbb{E}_{\mathbf{x}\sim p_R}\left[D(\mathbf{x})\right] - \mathbb{E}_{\mathbf{z}\sim p_z}\left[D\left(G(\mathbf{z})\right)\right]}_{\text{WGAN critic loss}} + \underbrace{\lambda\, \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{\mathbf{x}}}}\left[\left(||\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})||_2 - 1\right)^2\right]}_{\text{WGAN-GP gradient penalty}}, \tag{1}$$

where λ describes the gradient penalty coefficient and $\mathbb{E}$ denotes the expectation operator.

III. Method and Implementation

The standard GAN definition only allows for the generation of images, without respective labels. Therefore, for the generated data to be used for data augmentation, the conventional GAN formulation needs to be modified to also generate corresponding labels. For this modification, we build upon our previously proposed GAN architecture, which jointly generates images and their corresponding segmentation masks, for direct use of training data augmentation [10]. Compared to the standard GAN formulation, this architectural adaptation is a simple change in the network architecture, and can therefore be used with any GAN training scheme, such as WGAN-GP.

The main idea of this architecture is to fuse the image and segmentation mask to create an *image-segmentation pair*. This is done by concatenating both images along the channel axis. When training the GAN, the generator is now modified to generate image-segmentation pairs, instead of just images. The discriminator follows a similar principle, and now takes image-segmentation pairs as input, and its goal is to correctly decide if any given image-segmentation pair is real or synthetic. Therefore, the first convolutional layer of the discriminator needs to be modified to accept inputs, where the number of channels is equal to the number of channels of the image-segmentation pair.

All GANs in our experiments are trained using the WGAN-GP training scheme and loss function, as we found this to be the most robust method for training GANs, even across multiple datasets. For the gradient penalty hyperparameter, we used the default value suggested by Gulrajani et al., setting $\lambda = 10$. Additionally, compared to [10], we increase the image resolution in order to test how well the GAN is able to handle higher resolutions. Our code is based

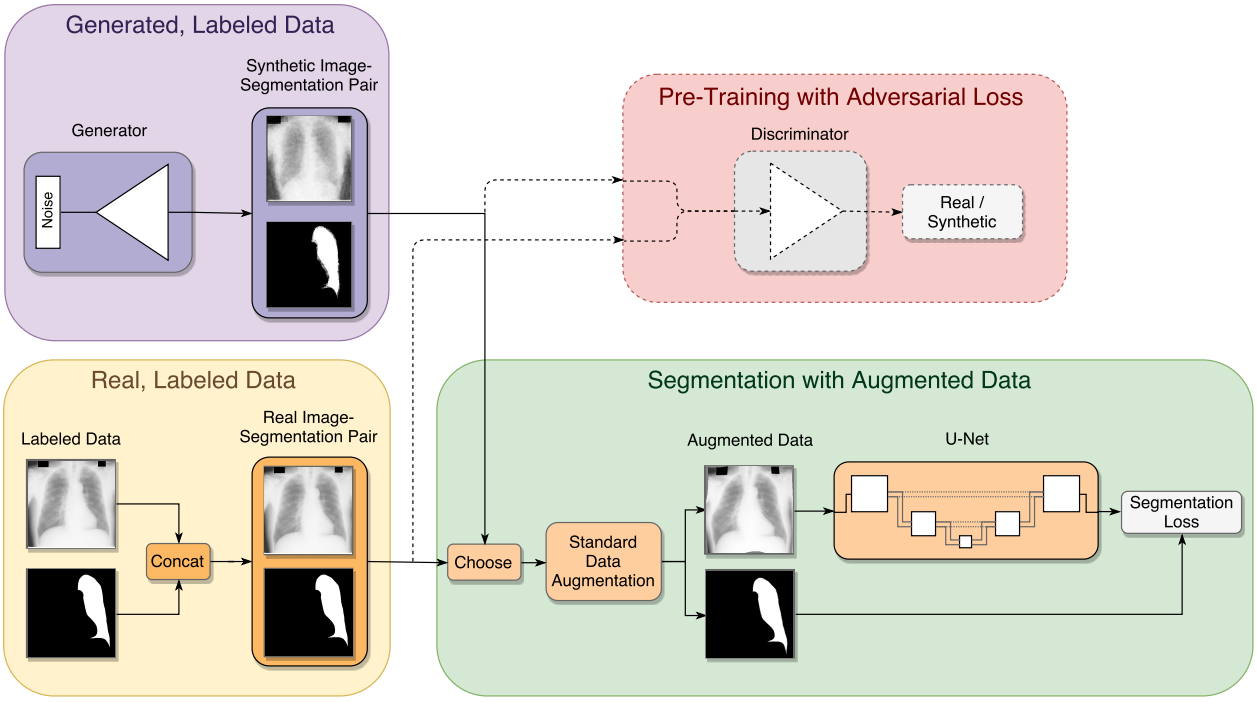


Fig. 1: Evaluation setup containing a pre-trained GAN generator producing image-segmentation pairs and a U-Net style segmentation network. The dashed lines and boxes illustrate the pre-training part of our setup.

on the code provided by the authors of WGAN-GP [5][1], using the TensorFlow [1] deep learning framework. For our evaluations, we use the following evaluation setup. First, we split the data into one or multiple training, validation, and test sets. Then, we train GANs for every training set of this dataset, until the generated image quality does not further improve. Finally, we take the fully-trained generator network, and use it directly as an input to a U-Net based [13] segmentation network. Compared to our previous evaluation [10], where a fixed amount of image-segmentation pairs was sampled, this on-the-fly generation allows for a much larger range of images to be sampled from the generator, better capturing the variation that the generator has learned from the data. For training the segmentation network, we use different ratios of real and generated data, and apply either no additional standard data augmentation, or a combination of standard data augmentation methods composed of *intensity shifts*, *intensity scaling*, *random translations*, *horizontal flipping* and *elastic deformations*. When mixing real and generated data, we exclusively use the specific GAN that was trained on the same real training data set, to keep all training sets separate. Our evaluation setup is illustrated in Figure 1.

IV. Implementation Details

Our GAN network architecture is based on DCGAN [11], only modified to generate image-segmentation pairs instead of just images. For the SCR Lung Database, our U-Net based segmentation network consists of 3 levels and uses a constant number of 64 filters for every convolutional layer. For the Cityscapes dataset, we use a U-Net based network with 4 levels and a constant number of 256 filters for every convolutional layer. We use Adam [8] as our optimizer for all networks, using a learning rate of $\eta = 0.0001$ and decay rates of $\beta_1 = 0.5$ and $\beta_2 = 0.9$ for the first and second moments, respectively. All network weights were initialized using the *He* initializer [6]. We use ReLU activations in all layers, except for the final generator layers, which use tanh activations. In contrast to our GANs, we do not use Batch Normalization in our segmentation networks.

For every segmentation network, we train for at least 3000 iterations. During training, we keep track of the minimal validation loss and its iteration number, as well as the network parameters at the validation loss minimum. Every time a new validation loss minimum is found, we train for at least 3000 more iterations. In practice, this resulted in a good compromise of network performance and training time. The final metric for our evaluation is the segmentation performance of the segmentation network on the unseen test data. For the evaluation on the test set, and therefore the final segmentation performance, we upsample the output of our segmentation networks using bicubic upsampling to the target resolution. Afterwards, we compute the Dice coefficient and the Hausdorff distance (for the SCR Lung Database), as well as the mean Intersection-over-Union (mIoU) (for the Cityscapes dataset) as our evaluation metrics. All evaluations were done on an NVIDIA Tesla K80 with

[1]GitHub: Improved Training of Wasserstein GANs, `https://github.com/igul222/improved_wgan_training`, Accessed: 14.03.2018

12GB of GPU memory, although all networks were designed for an NVIDIA GTX980M with 8GB of GPU memory. For both evaluations, all input images were intensity-normalized to a range of $[-1, 1]$.

V. Evaluation

A. Lung Segmentation of the SCR Lung Database

1) Dataset Description: The *SCR Lung Database* [17] is a dataset consisting of 247 chest X-ray images, taken from the JSRT database [15]. Its image resolution is $[2048 \times 2048]$ at a physical resolution of 0.175 mm per pixel in each dimension, and it contains groundtruth segmentation masks for 5 objects: both lungs, the heart and both clavicles. For our evaluation, we chose the task of segmenting the left lung from the image.

2) Evaluation Setup: All images are downsampled to a resolution of $[256 \times 256]$ before we use them for training in order to fit all our networks into GPU memory while still being able to use a large enough minibatch size for stable training. We shuffle the dataset randomly and split it into 3 folds, each containing 135 training images, 30 validation images and 82/83 test images, chosen such that all images are contained exactly once in the set of test images. For the final evaluation of the segmentation performance, we report performance as the average Dice score and Hausdorff distance over all folds.

As the first step of our evaluation, we train our modified GAN for each of the 3 folds of training data, resulting in 3 fully-trained GANs. As it is difficult to determine a quantifiable stopping criterion for the training of GANs, every GAN was trained for a fixed number of 10000 iterations, which took approximately 24 hours per GAN. The raw image-segmentation pairs from the generator are in the intensity range of $[-1, 1]$. Therefore, when training our segmentation network using generated images, we threshold all segmentation masks at 0 when computing the segmentation loss.

For the main part of our evaluation of the SCR Lung Database, we train multiple segmentation networks for every fold, using an exhaustive set of combinations of real and generated data as well as with and without standard data augmentation. We evaluated different combinations of standard data augmentation on one fold of the cross-validation set to find suitable augmentation parameters for the final comparison. To speed up this parameter search, we fixed elastic deformation at 10 pixels for each control point. The results for different augmentation methods are shown in Table I, and the combination of parameters listed in bold are used as our standard data augmentation method for the final evaluation. Important to note is that this parameter search was done on only a single fold of the validation set, therefore those results are not comparable to our final quantitative results which are averaged over all folds. Before computing our final segmentation performance metrics, we extract only the largest connected component. As the left lung is only a single connected component in all our images, this reduces false predictions of the resulting segmentation masks. Training each segmentation network took approximately 8 hours on our setup.

TABLE I: Comparison of augmentation parameters for the SCR Lung Database. For further evaluation, we use the augmentation parameters listed in **bold** as our standard data augmentation.

Augmentation Parameters				Validation Performance
Intensity shift around zero (stddev)	Intensity scaling around one (stddev)	Random translation around zero (stddev)	Elastic deformation around zero (stddev)	Dice (mean)
-	-	-	-	96.98%
-	-	-	10 px	97.06%
-	-	10 px	10 px	96.85%
-	0.05	-	10 px	97.03%
-	0.05	10 px	10 px	96.77%
0.05	-	-	10 px	96.40%
0.05	-	10 px	10 px	96.91%
0.05	0.05	-	10 px	96.63%
0.05	**0.05**	**10 px**	**10 px**	**97.12%**

3) Results: Example images from our GANs trained on the SCR Lung Database can be seen in Figure 2. The final segmentation performance, averaged across all folds, is shown in Table II. In order to better compare GAN-based data augmentation and standard data augmentation, we also present examples of resulting segmentation masks for two of our networks: the network trained on a mix of real and generated data without data augmentation (GANs-based augmentation), and the network trained solely on real data with standard data augmentation. Some of the best resulting examples, as well as the example showing the worst performance are shown in Figure 3.

B. Semantic Segmentation of the Cityscapes Dataset

1) Dataset Description: For our second evaluation, we chose the task of semantic segmentation using the *Cityscapes* [2] dataset. Cityscapes is a challenging dataset for semantic urban scene understanding, which aims to capture the complexity of real-world urban scenes. For 30 object classes divided into 8 groups, pixel-level and instance-level segmentation masks are provided for every image. The base resolution of all images is $[2048 \times 1024 \times 3]$. This dataset consists of 2975 training images and 500 validation images with finely annotated segmentation masks, with an online submission system used to evaluate performance on the test set, for which the groundtruth segmentation masks are not known. Since this segmentation problem is much more challenging compared to the lung segmentation problem of the SCR Lung Database, we decided to only do segmentation of the 8 object groups (*'categories'*) defined in the Cityscapes dataset, and not on the individual classes.

2) Evaluation Setup: We created our own data split from the given training and validation sets. We used all 500 images from the Cityscapes validation set as our test set. For our internal validation set, we randomly selected 400 images from the Cityscapes training set. Finally, our training set consisted of the remaining 2575 images from the Cityscapes training set. Due to the much larger amount of data compared to the SCR Lung Database and the time consuming nature

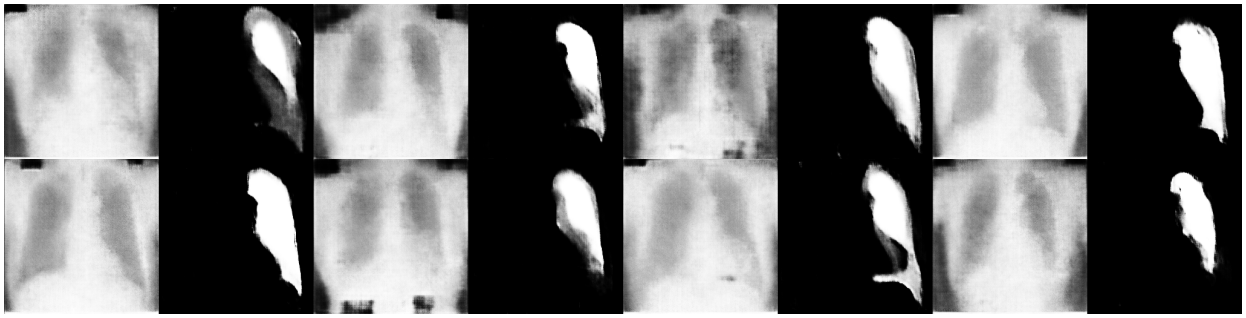


Fig. 2: Example images from our GAN trained on the SCR Lung Database. Odd columns show generated images, while even columns show the respective generated segmentation masks.

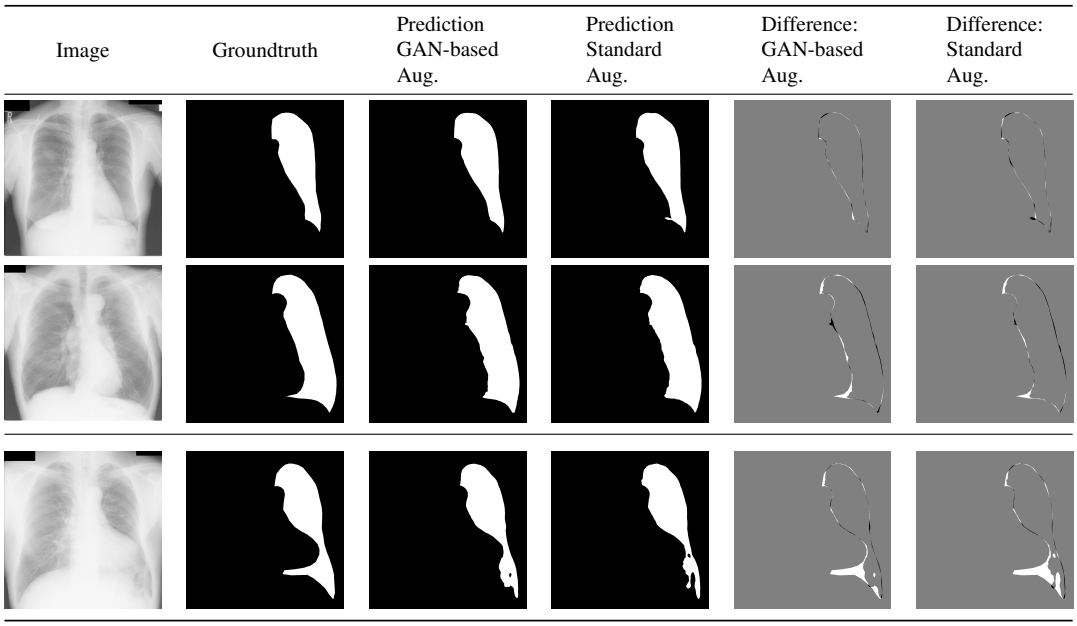


Fig. 3: Comparison of segmentation masks from fully trained segmentation networks between standard data augmentation and GAN-based data augmentation for the SCR Lung Database. Rows 1 and 2 show good results, while row 3 shows the worst performing test example. Columns 3 and 5 show results from our segmentation network trained using GAN-based augmentation with a mix of real and generated data, while Columns 4 and 6 show the results of our segmentation network trained using standard data augmentation.

TABLE II: Segmentation performance comparison between training on real data, generated data, and mixed data, using either no additional data augmentation, or standard data augmentation (see Table I), evaluated on our test set of the SCR Lung Database. Since the previous work of Neff et al. [10] was not tested on full resolution, the Hausdorff distance was omitted from these results, as it is not an accurate comparison.

Network ID	# real pairs in minibatch	# generated pairs in minibatch	Aug.?	Dice (mean)	Dice (stddev)	Hausdorff (mean)	Hausdorff (stddev)
16-0-Aug	16	0	yes	97.65%	1.65%	1.2057 mm	0.3131 mm
16-0-NoAug	16	0	no	97.42%	1.66%	1.2626 mm	0.3440 mm
8-8-Aug	8	8	yes	97.65%	2.28%	**1.1722 mm**	0.3313 mm
8-8-NoAug	8	8	no	**97.68%**	1.47%	1.2106 mm	0.3154 mm
0-16-Aug	0	16	yes	96.55%	1.63%	1.2651 mm	0.3067 mm
0-16-NoAug	0	16	no	96.32%	2.02%	1.3273 mm	0.3434 mm
Neff et al. [10]	16	0	no	96.08%	1.01%	-	-
	≈ 8	≈ 8	no	95.37%	1.21%	-	-
	0	16	no	91.72%	2.83%	-	-

TABLE III: Comparison of augmentation parameters for the Cityscapes dataset. For further evaluation, we use the augmentation parameters listed in **bold** as our standard data augmentation.

Augmentation Parameters				Validation Performance
Intensity shift around zero (stddev)	Intensity scaling around one (stddev)	Random translation around zero (stddev)	Horizontal flipping	mIoU
0.10	0.10	10 px	no	69.89%
0.05	0.05	5 px	no	74.21%
-	-	-	no	78.04%
0.05	0.05	-	yes	79.50%
-	**-**	**-**	**yes**	**80.42**%

of our evaluation, we only evaluate on this single fold of data. For all networks in this evaluation, we downscaled the resolution of all input images to $[256 \times 128 \times 3]$ to be able to fit our generator network and our segmentation network into memory at the same time, while still keeping a sufficiently large minibatch size for training stability. The Cityscapes dataset contains a lot of small, thin structures (e.g. objects such as street lights and traffic signs), that get reduced to just a few pixels in size when downsampling to such an extent. This is especially apparent for the *'Human'* and *'Object'* categories, which contain the smallest objects in the dataset, such as pedestrians and street lights.

Before training our segmentation networks, we train our modified GAN on the Cityscapes dataset for 10000 iterations, as the image quality did not improve further after that. For the standard data augmentation, we experimented using a set of multiple different augmentation methods, and chose the best one based on the performance on the validation set. The validation results for different augmentation methods are shown in Table III, and the combination of parameters listed in bold are used as our standard data augmentation method for further training.

We threshold the output segmentation images of the generator to get discrete segmentation masks.

For our final evaluation of the Cityscapes dataset, we train our segmentation network on different ratios of real and generated data, similar to the previous experiment, using either no augmentation or standard data augmentation. Each segmentation network took approximately 24 hours to train until convergence.

3) Results: Our final segmentation performance for all different evaluation setups of the Cityscapes dataset is shown in Table IV. Additionally, we show the resulting mIoU for every category for the four best performing networks in Figure 4. Since the significant amount of downsampling of the input images results in small objects vanishing or being reduced to single-pixel size, and therefore not being useful for training, we also present mIoU results excluding the 'Human' and 'Object' categories in Table IV. Similar to our evaluation of the SCR Lung Database, in Figure 5 we show an example of a resulting segmentation mask to better compare GAN-based augmentation to standard augmentation.

TABLE IV: Segmentation performance comparison between training on real data, generated data, and mixed data, using either no additional data augmentation, or standard data augmentation, evaluated on our test set of Cityscapes.

Network ID	# real pairs in minibatch	# generated pairs in minibatch	Aug.?	mIoU	mIoU excluding 'Human' and 'Object'
8-0-Aug	8	0	yes	**78.59**%	**88.94**%
8-0-NoAug	8	0	no	76.16%	87.67%
4-4-Aug	4	4	yes	75.48%	87.32%
4-4-NoAug	4	4	no	76.30%	87.86%
0-8-Aug	0	8	yes	47.05%	65.35%
0-8-NoAug	0	8	no	46.32%	64.26%

VI. Discussion and Conclusion

Our main focus of this work was to perform a comparison between standard data augmentation and GAN-based data augmentation. For the first comparison, we chose to perform medical image segmentation of the SCR Lung Database, as this allows us to directly compare to previously reported results. Figure 2 shows that for the SCR Lung Database, our GAN manages to generate high-quality images with corresponding segmentation masks that fit the generated image well. Compared to our previously published GAN examples of this dataset shown in [10], the generated samples are of much higher quality and more closely resemble the training data. We also do not experience mode collapse of our generated samples compared to these previous results, as the resulting samples show similar variety to the training set the generator was trained on. Looking at the segmentation performance shown in Table II, we can see that the Dice scores and Hausdorff distances are very close between all networks, only showing a significant gap for the networks trained on strictly synthetic data and for our previous results in [10]. Augmenting with synthetic images from a trained GAN does not decrease the segmentation performance, and networks trained with a mix of synthetic and real images stay competitive with networks trained on strictly real data, using standard data augmentation. Even though the difference is small, the best result (Dice score, standard deviation of Dice) of our evaluation was achieved using our GAN-based augmentation, i.e. using a network trained on mixed real and synthetic data. This suggests that GAN-based augmentation might be a viable augmentation strategy in the future, especially if GAN research further improves on the quality and variety of generated images. Furthermore, it is very interesting to see that our network trained on purely synthetic data achieves better results compared to the network trained on real data of our previous work [10]. This is mostly due to the higher quality of the synthetic images sampled from our GAN on-the-fly. This shows that our GAN has managed to learn enough about the underlying training data distribution to produce valuable images for training segmentation networks.

Looking at the samples produced from our GAN-

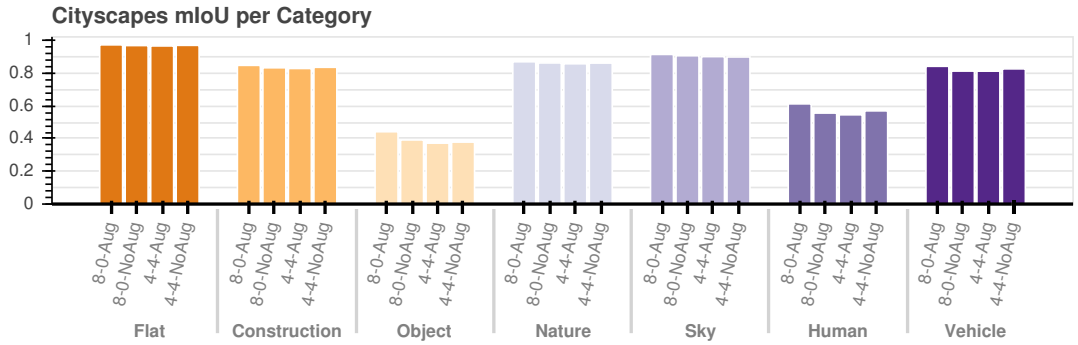


Fig. 4: mIoU for every category of the Cityscapes dataset. For visual clarity, we omit the worst performing networks and only report results for the four best networks, identified by their network ID shown in Table IV.

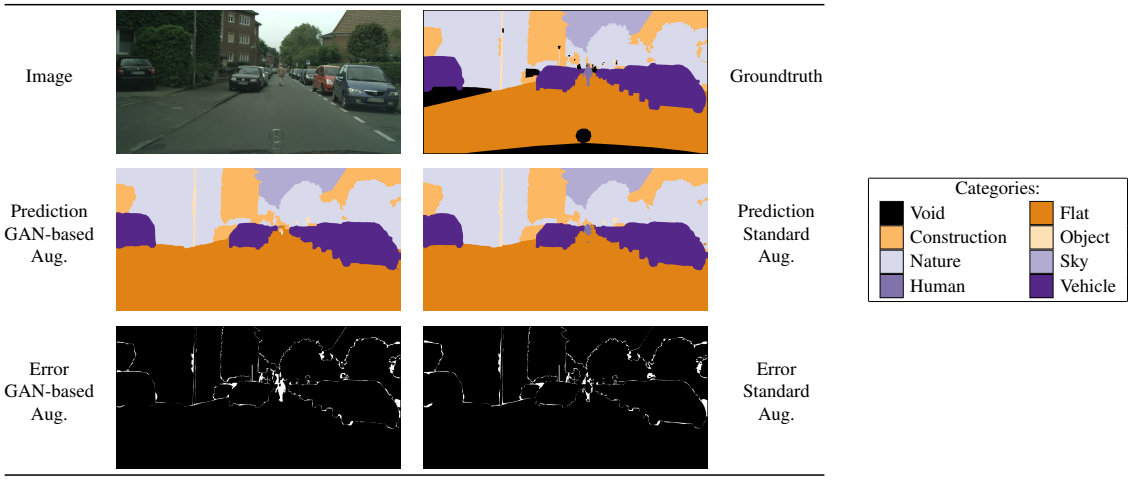


Fig. 5: Comparison of segmentation masks from fully trained segmentation networks between standard data augmentation and GAN-based data augmentation for the Cityscapes dataset for the best performing test image.

augmented network and our network trained with standard augmentation in Figure 3, we can see that the segmentation quality is also equally good. For some test images, the network trained with GAN-based data augmentation produces better segmentation masks, while for others, the network trained with standard data augmentation achieves higher quality results. Since the Dice scores and Hausdorff distances are almost identical, and we cannot determine significant differences in image quality, it seems that the lung segmentation problem for this dataset is already very well modeled by the U-Net. Additional augmentation does not provide any more benefits, but also does not have a negative impact on the results either. However, GAN-based augmentation also does not lead to worse performance in this case, which was not the case in our evaluation presented in [10], suggesting that the higher quality GAN images from WGAN-GP improved the overall augmentation method, and our GAN managed to better capture the distribution of our training data.

For the evaluation on the Cityscapes dataset, results are interesting as well. While our GAN managed to generate images of reasonable variety, the image quality is not as high, as the Cityscapes dataset is more complex, and therefore more difficult to learn for a generative model. Looking at the quantitative evaluation of the Cityscapes dataset, we found that the best standard data augmentation for this dataset and our segmentation network architecture was to just use horizontal flipping (see Table III). Using other combinations of intensity shift, intensity scaling, or random translation led to worse segmentation performance. For some settings, the segmentation performance was even worse than not using data augmentation at all. Similarly, we can observe that using horizontal flipping in combination with our GAN-based approach (*4-4-Aug*) leads to worse performance compared to just using GAN-based augmentation (*4-4-NoAug*). This illustrates an important point of data augmentation - the augmentation parameters require careful tuning to fit the dataset, as unsuitable data augmentation can have a negative effect by drastically reducing the segmentation performance. From our final segmentation performance shown in Table IV, we can see that the network trained on real data, using horizontal flipping for data augmentation (see Table III), achieved the best performance compared to all other networks. However, we can again observe that the network trained using GAN-

based augmentation without additional standard data augmentation achieves similar performance to the network that was trained on real data without augmentation. Especially interesting is that when using GAN-based augmentation without standard augmentation, the results are better than some of the results when using standard augmentation shown in Table III. This illustrates that our GAN has learned a reasonable representation of our training data, even though the generated samples are not of high quality.

Compared to the highscore database of the Cityscapes dataset[2], our baseline performance for the category mIoU is in line with the weaker results on the online database. This is mostly due to the large amount of downsampling we perform on the input images, as well as that we do not use pre-trained networks as all other competing methods do. Because one of our main goals was to evaluate how GAN-based data augmentation affected the results of training segmentation networks, we did not want to additionally pre-train our networks, as that would introduce another variable that significantly impacts training behavior of deep networks.

Performing large amounts of downsampling leads to a lower segmentation performance for small or thin structures, and borders between regions, as those fine details vanish when downsampling is applied. This effect can be seen by comparing the resulting segmentation masks of our networks, shown in Figure 5. Most of the errors of our results are in the border regions between classes, as the fine detail necessary to determine exact borders is lost during downsampling. We also observe the consequence of downsampling in Figure 4, where we show results for every category. While our networks consistently show weaker performance on the 'Object' and 'Human' categories, the other categories show good results, given that we only used a standard U-Net segmentation network architecture with additional data augmentation methods. Computing the mIoU over all categories but those two, we achieve much better scores, as can be seen in Table IV.

To conclude, we performed an extensive evaluation of the possibilities of using GANs for training data augmentation in image segmentation tasks. From our current results, we cannot conclude GAN-based augmentation has a positive or negative impact, but we believe that if a GAN was able to fully learn the training data distribution, the additional synthetic data could be highly useful as a regularizer for deep networks. Compared to standard data augmentation, GAN-based augmentation does not require extensive data analysis to find out optimal augmentation parameters. Especially in the Cityscapes evaluation, we saw how certain data augmentation parameters can lead to much worse performance, therefore an augmentation method that is learned from data would save a lot of effort in fine-tuning deep networks. The most straight-forward future improvement would be to increase the resolution and representation power of our GAN, leading to higher quality synthetic images. We are certain that such an improved generative model could be used as a data augmentation method to improve performance for supervised deep learning tasks.

[2]Cityscapes Pixel-Level Semantic Labeling Task Results, {https://www.cityscapes-dataset.com/benchmarks/\#pixel-level-results}, Accessed: 14.03.2018

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 (OSDI)*. USENIX Association, 2016, pp. 265–283.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[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 27 (NIPS)*. Curran Associates, Inc., 2014, pp. 2672–2680.

[5] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems 30 (NIPS)*. Curran Associates, Inc., 2017, pp. 5769–5779.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2015, pp. 1026–1034.

[7] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," *Proceedings of the Sixth International Conference on Learning Representations (ICLR)*, 2018.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the Third International Conference on Learning Representations (ICLR)*, 2015.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[10] T. Neff, C. Payer, D. Štern, and M. Urschler, "Generative Adversarial Network based synthesis for supervised medical image segmentation," in *Proceedings of the OAGM&ARW Joint Workshop*, 05 2017, pp. 140–145.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with Deep Convolutional Generative Adversarial Networks," in *Proceedings of the Fourth International Conference of Learning Representations (ICLR)*, 2016.

[12] 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.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer, Cham, 2015, pp. 234–241.

[14] A. Rozantsev, V. Lepetit, and P. Fua, "On rendering synthetic images for training an object detector," *Computer Vision and Image Understanding (CVIU)*, vol. 137, pp. 24 – 37, 2015.

[15] 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 (AJR)*, vol. 174, no. 1, pp. 71–74, Jan. 2000.

[16] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2242–2251.

[17] 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.

Multi-camera Array Calibration for Light Field Depth Estimation

Bernhard Blaschitz[1], Doris Antensteiner[1] and Svorad Štolc[1]

***Abstract*— At the core of stereo methods for depth estimation and 3D reconstruction lies geometric calibration, i.e. the determination of intrinsic and extrinsic camera parameters and consecutive image rectification, such that the epipolar constraints are met in all views. In this spotlight paper, we present a multi-camera array calibration that fulfills the requirements for 3D reconstruction. The method is based on an optimization procedure that minimizes the reprojection error. We used it to calibrate the Xapt Eye-sect XA camera array with 4x4 camera modules equipped with identical wide-angle lenses. For this particular setup, we analyzed the algorithm's precision step by step, from initial pairwise multi-view stereo calibration to final bundle adjustment, to assess influence of each individual step. The conducted quantitative analysis based on the reprojection error revealed superiority of the bundle adjustment over all other considered intermediate steps yielding accuracy as much as 33x higher than the initial pairwise method. In order to demonstrate real-world performance of the calibrated camera array, we present a number of acquisitions of different physical objects along with estimated disparity maps and corresponding texture images generated by a light field multi-view stereo algorithm.**

I. Introduction

In recent years, there has been a boom of commercially available stereo and multi-camera systems for both consumer as well as industrial applications. Geometries of existing multi-camera systems are very diverse: from matrix cameras such as Xapt Eye-sect XA used in this paper, through plenoptic cameras such as Lytro or Raytrix, to unstructured multi-camera systems that make use of multiple free camera modules. Capturing multiple views of a scene by multi-camera systems is often interpreted as *light field*, which is the 4D radiance function of 2D position and 2D direction of each light ray propagating thorough space in regions free from occluders [1].

The steady improvement of stereo matching algorithms creates a high need for automated tools for calibrating such light field systems, consecutively allowing highly accurate depth estimation and 3D reconstruction.

The basis of multi view calibration is the geometric calibration, i.e. the determination of intrinsic and extrinsic camera parameters [2], as well as multi-view stereo and bundle adjustment [3] and image rectification.

In this case study, we present a multi-camera array calibration pipeline that fulfills the requirements for 3D reconstruction, such as epipolar constraints. The method is based on an optimization procedure which minimizes the reprojection error.

[1]all three are with AIT Austrian Institute of Technology, Giefingasse 4, 1210 Vienna, Austria firstname.lastname@ait.ac.at

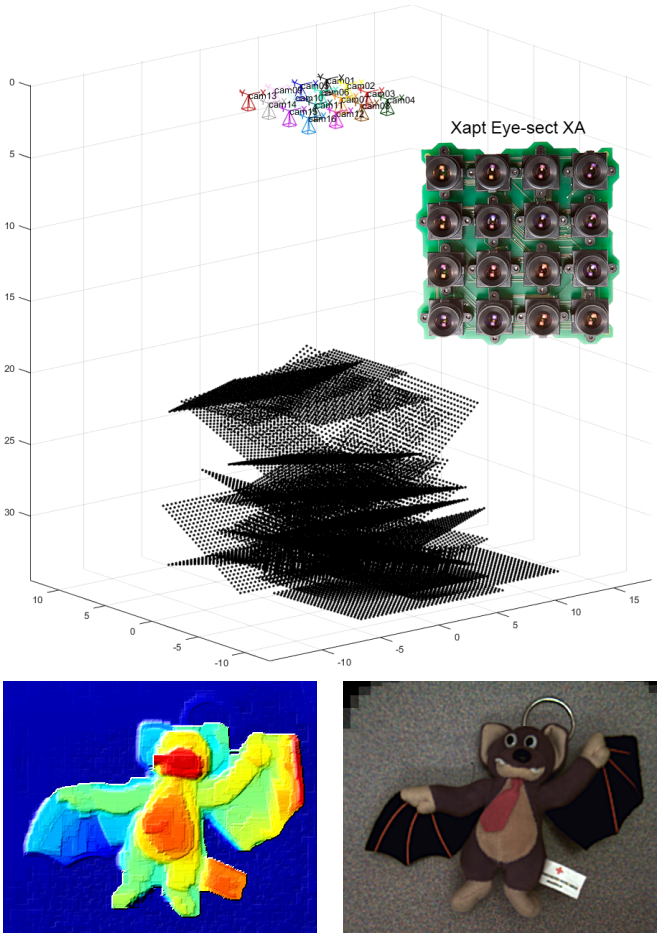


Fig. 1. *Top:* Estimated positions of the 16 camera modules of Xapt Eye-sect XA as well as estimated poses of the presented calibration patterns as a result of the proposed optimization procedure minimizing reprojection errors. Note that each camera module has a resolution of just 480×480 pixels. *Bottom:* Example of a depth model (left: disparity map; right: texture image) obtained by a light field multi-view stereo algorithm making use of the calibrated system. See also Figs. 4 and 5 for further examples of reconstruction from the same setup.

For a multi-view system, which is positioned in an unstructured manner and thus results in a irregular light field, it is favorable to implement a generic multi-view matching scheme. In this paper we considered an algorithm inspired by [4] and [5], extended by a real-time discrete-continuous optimizer [6] for a globally consistent depth map under the generalized first-order *total variation* (TV) prior.

In Sec. II we describe our multi-view calibration pipeline, which improves existing methods. In this case study, the calibration of the Xapt Eye-sect XA camera array, its accuracy as well as a number of depth reconstructions generated by a

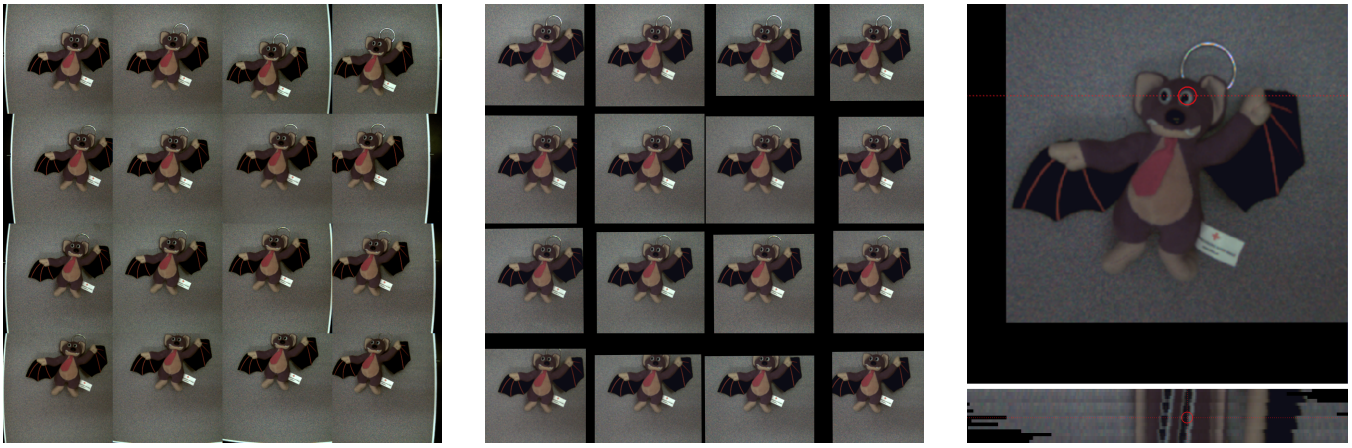


Fig. 2. *Left:* The input image taken by the Xapt Eye-sect XA camera. *Middle:* The undistorted rectified image. Note that all edges of the chessboard are bent in the left image (with distortion) and straight unbent (without distortion) in the middle image. *Right:* The rectified view of Camera 9 and the associated horizontal EPI image; the corresponding 3d reconstruction is in Fig. 1.

light field multi-view stereo algorithm are shown in Sec. III. Finally, in Sec. IV we conclude this study.

II. Calibration Method

We present a methodology, which was implemented in Matlab and relies on the *Complete Camera Calibration Toolbox for Matlab* [8], mainly for the intrinsic calibration. Our contribution improves over the prior art in the following points:

- It makes use of a high precision calibration target and accompanying algorithms [7], which improve the accuracy of automated pattern detection (see Fig. 3) and is stable to defocusing and sensitive to mirroring.
- It computes a true multi-view calibration instead of a pairwise stereo calibration. For this we use *bundle adjustment* [3], which optimizes intrinsic and extrinsic camera parameters by minimizing the overall reprojection error (see Sec. II-A). It can also cope with patterns that are not visible in all cameras.
- It allows image rectification suitable for light field processing, such as depth measurement and 3D reconstruction (see Sec. II-B).

A. Optimizing camera parameters

The notation complies with the camera model introduced in *OpenCV Toolbox* [9] and builds on the toolbox from [8].

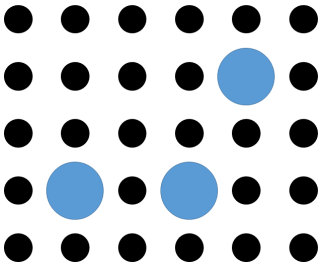


Fig. 3. We use an improved central element [7] for the calibration target, which has the advantage that only three dots in the center have to be visible in order to recognize the pattern, with a high robustness w.r.t. defocusing.

The intrinsic camera model has 10 degrees of freedom: two focal lengths f_x, f_y, two principal point coordinates c_x, c_y, camera skew α, three radial distortion parameters k_1, k_2, k_3 and two tangential distortion coefficients p_1, p_2, which comprise the distortion parameters $d = (k_1, k_2, p_1, p_2, k_3)$. Furthermore, there are 6 degrees of freedom for extrinsic camera parameters T, which comprise the position and rotation of the camera in a global coordinate system.

The bundle adjustment [3] is a non-linear method for refining extrinsic and intrinsic camera parameters, as well as the structure of the scene. It is characterized by minimizing the reprojection error by a standard least-squares approach

$$E(\mathbf{C}, \mathbf{X}) = \sum_{i=1}^{n} \sum_{j=1}^{m} dist(x_{ij}, C_i(X_j)))^2, \qquad (1)$$

where $C_i(X_j) = C(H_i, T_i, d_i, X_j)$ is the *reprojected point*, i.e. the image of a point $X_j \in \mathbb{R}^3$ as observed by the i-th camera. Furthermore, x_{ij} is the corresponding detected point of the calibration pattern and $dist(x_{ij}, C_i)$ is the point's reprojection error.

We initialize the minimization with a single view calibration, choose one camera as the central view and initialize the other cameras' extrinsic parameters T by factoring out the average difference in pose of the detected calibration pattern. This is inspired by the initialization of the stereo calibration in [8], hence the assigned designation *pairwise* in Tab. I.

The quadratic objective function of Eq. 1 is minimized with a standard least squares solver. To avoid getting stuck in a local minimum, due to many degrees of freedom, an outer iteration for different optimization phases keeps certain parameters fixed.

For the phase *patterns* in Tab. I, only the positions and rotations of calibration patterns are optimized and all camera specific parameters remain unchanged. For the next phase, *extrinsics*, poses of calibration patterns as well as intrinsics are fixed and only the positions and rotations of all cameras are optimized. Finally, in the phase *bundle adjustment* all parameters are allowed to change.

B. Image rectification for light field processing

In order to facilitate light field / multi-view correspondence analysis methods, images captured by the system need to be rectified making use of the obtained calibration model. Since all cameras usually point to different directions and their locations are rarely coplanar, the standard stereo image rectification [10], which is defined for two cameras, cannot easily be generalized.

As described in [5], all camera views need to be reprojected to a common regression plane ε, which turns the costly warping necessary for cross-comparison between multiple images to simpler translation and scaling operations. If all camera centers are coplanar and ε is chosen parallel to the camera plane, the entire image manipulation needed for the correspondence analysis between multiple cameras reduces just to a translation, which poses a significant computational and algorithmic advantage over the standard stereo approach.

The rectified images have been computed as follows: the regression plane ε has been chosen parallel to the plane fitted through the camera centers and minimizing the squared distance to all calibration patterns. Then, all camera images have been projected onto ε and resampled with the same regular pixel grid. The obtained images form the rectified light field which is required to perform depth estimation.

III. Calibrating Xapt Eye-sect XA

With regard to demonstrating real-world performance of the proposed calibration model, we have taken an example of the Xapt Eye-sect XA camera array with 4x4 camera modules equipped with identical wide-angle lenses. Initially a number of images of AIT's calibration target [7] were acquired for estimating the camera's calibration model, see Fig. 1 (top) for a visualization.

A. Comparison of the reprojection errors

In order to compare the results of our method comprising bundle adjustment with the original method of Bouguet [8], we have conducted a quantitative accuracy analysis based on the reprojection error. The results of this analysis are shown in Tab. I.

For a typical set of calibration images, the reprojection errors resulting from Eq. (1), which are computed per camera after a pairwise optimization with respect to Camera 6, are shown in row *pairwise*. The reprojection errors after further optimization of the calibration pattern poses are given in row *patterns*. The row *extrinsics* shows errors after additional optimization of extrinsic parameters for all cameras. Finally, the reprojection errors after the full bundle adjustment, which also includes optimizing the intrinsic parameters of all cameras, are provided in row *bundle adjustment*.

This exemplary application shows that the biggest drop in the reprojection error occurs in the first step after the initialization, which is when the optimization of the calibration pattern poses took place. Nevertheless, the bundle adjustment showed superior results over all other considered intermediate steps yielding an accuracy which is 33x higher than the initial pairwise method.

B. Depth estimation using light fields

Light field data is captured with the Xapt Eye-sect XA camera by measuring the irradiance values from different viewpoints on the object. For each observation we thereby obtain 4×4 images with different viewpoints, each of which has a resolution of 480×480 pixels. For the multi-view correspondence analysis we considered 32 random camera pairs out of all 120 (i.e. $\binom{16}{2}$) possible pairs.

Since the Xapt Eye-sect XA shows irregularities in the system geometry, namely camera positions were off the grid by as much as 3% of the baseline, the implementation of a robust matching algorithm for 3D reconstruction is essential. Therefore we implemented a robust multi-view matching algorithm for a qualitative evaluation of the camera calibration as described below.

We generate *normalized gradient* features for comparing local image structures, which we compare using the *sum of absolute differences* (SAD). This approach proved more performant compared to the traditional *Census transform / Hamming distance* [11] tandem, especially when coupled with a subsequent regularizer.

Using the resulting features of the rectified light field images (which we obtained with the calibration model as described in Sec. II-B), we perform a correspondence analysis inspired by [4] in each spatial location $(x,y) \in X \times Y$ of a chosen reference view of the camera matrix. The analysis is conducted separately for pairs of cameras. Each hypothesis for a defined camera pair contributes to one global cost function. The resulting cost volume describes the matching quality of visual structures for defined disparity hypotheses within the light field views.

A globally consistent depth solution was obtained under the *total variation* (TV) prior, using a real-time discrete-continuous optimizer proposed in [6]. This algorithm shows exceptional performance, both concerning the speed as well as the solution quality. Further depth refinement methods can be implemented as described in [12].

Figs. 4 and 5 show qualitative reconstruction examples.

IV. Conclusions

We presented a pipeline for geometric multi-view calibration, which includes bundle adjustment. With our routines we have calibrated a multi-view camera and used it for capturing depth information.

The conducted quantitative analysis based on the reprojection error revealed superiority of the bundle adjustment over all other considered intermediate steps yielding an accuracy as much as 33x higher than the initial pairwise method. The largest refinement of the reprojection error was observed in the first step after the initialization during the optimization of the poses of the calibration pattern.

For a qualitative assessment of the calibration model we implemented a multi-view stereo matching algorithm which includes a real-time discrete-continuous optimizer which allows a globally consistent depth map under the generalized first-order *total variation* (TV) prior.

TABLE I

Comparison of the mean reprojection errors (in pixel) per camera of Xapt Eye-sect XA array for different phases of the proposed algorithm. Note that the algorithm was initialized with the *pairwise* method and reference camera 6, subsequently different parameters were optimized: in *patterns* poses of the calibration patterns, in *extrinsics* only the 16 cameras' poses and in *bundle adj.* all parameters, including camera intrinsics.

Phase/Camera	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	Avg.
pairwise	2.28	8.75	9.32	10.29	2.27	0.09	8.49	16.64	6.21	1.91	13.47	12.73	4.06	8.18	9.01	8.44	7.63
patterns	0.37	0.44	0.57	0.41	0.44	0.54	0.62	0.42	0.38	0.41	0.51	0.77	0.46	0.34	0.49	0.93	0.51
extrinsics	0.16	0.24	0.33	0.22	0.24	0.54	0.58	0.25	0.22	0.15	0.29	0.28	0.29	0.24	0.37	0.29	0.29
bundle adj.	0.12	0.19	0.30	0.18	0.23	0.15	0.55	0.19	0.19	0.13	0.26	0.21	0.27	0.21	0.33	0.22	0.23

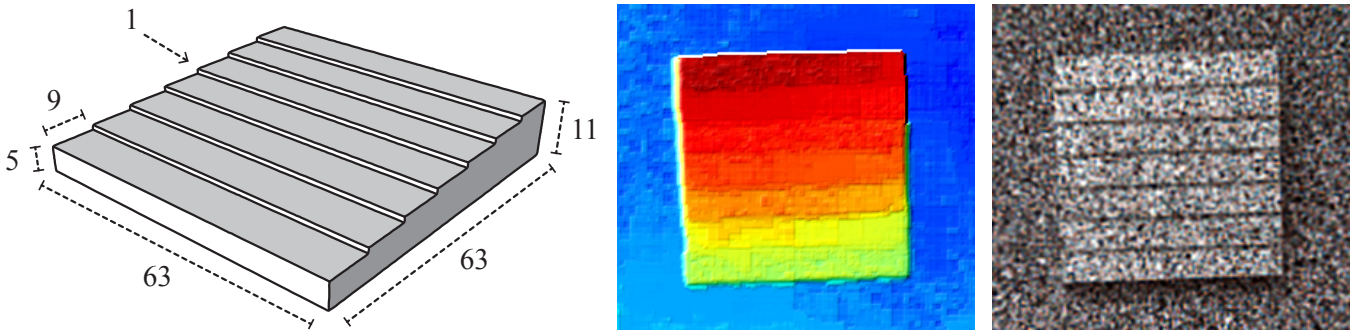


Fig. 4. The performance of the presented multi-camera array calibration was tested by means of a 3D printed staircase object with seven 1 mm steps. The object was acquired with the Xapt Eye-sect XA camera at the working distance of approx. 340 mm. The estimated system's baseline and the average focal length was approx. 90 mm and 710 mm, respectively. The corresponding depth resolution was $\Delta z \approx 1$ mm. Despite a low camera resolution and limited baseline, the obtained disparity map generated by the calibrated camera array accurately reproduced each individual step of the staircase, hence we consistently operate at or above the predicted depth resolution of this system. That is additional evidence of the calibration model's high accuracy additionally to low reprojection errors.

REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. of Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH, New York, NY, USA, 1996, pp. 31–42.

[2] F. Ciurea, D. Lelescu, P. Chatterjee, and K. Venkataraman, "Adaptive geometric calibration correction for camera array," *Electronic Imaging*, vol. 2016, no. 13, pp. 1–6, 2016.

[3] Y. Furukawa and J. Ponce, "Accurate camera calibration from multi-view stereo and bundle adjustment," *International Journal of Computer Vision*, vol. 84, no. 3, pp. 257–268, 2009.

[4] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 15, no. 4, pp. 353–363, 1993.

[5] R. T. Collins, "A space-sweep approach to true multi-image matching," in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*. IEEE, 1996, pp. 358–363.

[6] A. Shekhovtsov, C. Reinbacher, G. Graber, and T. Pock, "Solving dense image matching in real-time using discrete-continuous optimization," in *21st Computer Vision Winter Workshop*, 2016.

[7] B. Blaschitz, S. Štolc, and D. Antensteiner, "Geometric calibration and image rectification of a multi-line scan camera for accurate 3d reconstruction," in *IS&T Electronic Imaging*, 2018.

[8] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.

[9] G. Bradski, "Opencv toolbox," *Dr. Dobb's Journal of Software Tools*, 2000.

[10] R. Klette, *Concise computer vision*. Springer, 2014.

[11] C. Zinner, M. Humenberger, K. Ambrosch, and W. Kubinger, "An optimized software-based implementation of a census-based stereo matching algorithm," in *International Symposium on Visual Computing*. Springer, 2008, pp. 216–227.

[12] D. Antensteiner, S. Štolc, and T. Pock, "A review of depth and normal fusion algorithms," *Sensors*, vol. 18, no. 2, 2018.

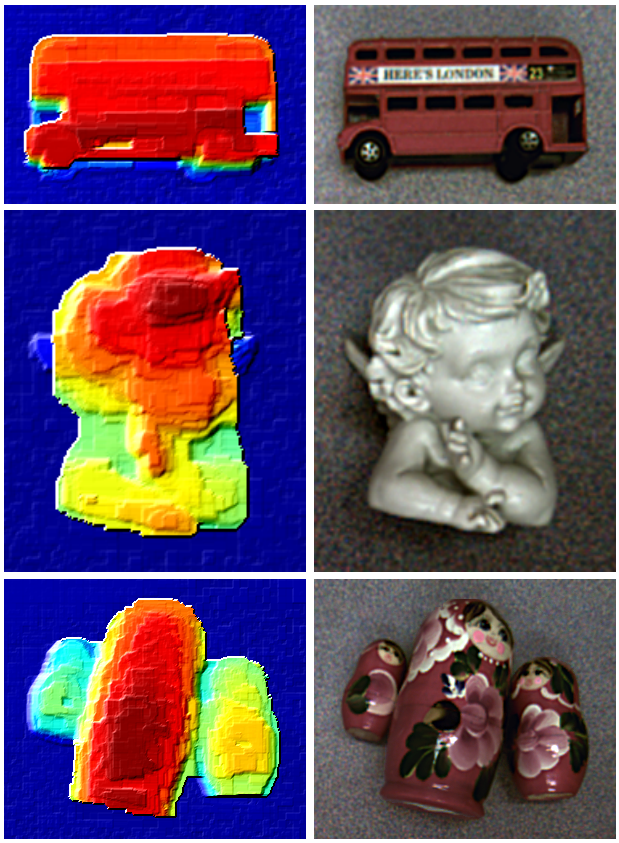


Fig. 5. Examples of the estimated disparity maps (left) and corresponding texture images (right) for several real-world objects. The disparity maps are displayed in pseudo colors, where blue stands for areas further away and red for areas closer to the camera. In order to increase readability of surface details, slight shading was applied to disparity maps.

CNN training using additionally training data extracted from frames of endoscopic videos

Georg Wimmer[1] and Michael Haefner[2] and Andreas Uhl[1]

Abstract— **Insufficient amounts of labeled training data poses a big problem in machine learning, especially for medical applications where medical image data sets are usually quite small. In this work we propose a method to increase the amount of labeled endoscopic image data in order to improve the classification accuracy of automated diagnosis systems for the classification of colonic polyps. Starting from a small colonic polyp endoscopic image database, we increase the number of images by tracking the content shown in the images through the endoscopic videos and by extracting patches from frames of the videos that show the same content as in the images of the database, but under different viewing conditions. By means of our proposed method we are able to increase the amount of labeled image data by factor 40, without adding images of insufficient image quality or images without clearly visible features for the differentiation of colonic polyps. We will show that this increased amount of training images can drastically improve the performance of CNNs, which are state-of-the-art in the automated classification of colonic polyps.**

I. Introduction

Modern endoscopy devices are able to take images and videos from inside the colon which facilitates computer-assisted analysis of the acquired material with the goal of detecting and diagnosing abnormalities.

Usually, endoscopic image databases consist of image patches that are manually extracted from images routinely captured during endoscopy or from manually chosen frames of endoscopic videos. The image patches of the databases show regions of interest with clearly visible mucosa structures and/or geometric features that enable a differentiation between healthy and affected mucosa (in our case we differentiate between different classes of colonic polyps). The labels for those extracted image patches are provided by medical experts.

Although videos are routinely recorded during endoscopy, the video material cannot be used for the training of automated diagnosis systems since there are no labels given for the mucosal regions shown in the videos, except for the ones where images of frames were manually extracted and labeled. Furthermore, for large parts of the video the image quality is insufficient to enable a classification of the shown mucosal regions.

In this work we propose a method that generates additional labeled image data with sufficient image quality by tracking the regions shown in the manually extracted patches (with given label information) throughout the video. By automatically extracting image patches of those regions from endoscopic video frames we generate additional image data with given label information. Those new image patches show the same regions as shown in the original, manually extracted image patches, but under different viewing conditions (different scales and viewpoints), with different image qualities and potentially also with different imaging modalities (image enhancement technologies like e.g. i-Scan modes can be switched on and off during endoscopy). The final step of our proposed method filters out all image patches with insufficient image quality. Contrary to previous approaches assessing the informativeness of frames in colonoscopic videos [1], [2], we do not only focus on image blur as quality measures but also on the visibility of mucosal texture structures and discard all images without clearly visible texture structures.

To test if our approach to increase the number of training images is suited for automated diagnosis systems, we apply them to the old (manually extracted) and to the new, enlarged database and compare the classification results of the two databases. More specifically, we train convolutional neural networks (CNNs) using both databases and compare their classification accuracy.

Convolutional neural networks are state-of-the-art in the automated diagnosis of colonic polyps and outperform handcrafted image representations as shown in [3], [4]. Generally, thousands or millions of images are used and required as data corpus to achieve well generalizing deep architectures. In endoscopic image classification however, the available amount of data usable as training corpus is often much more limited to a few hundreds or thousands of images or even less. By means of our proposed method to increase the amount of labeled training data we aim to overcome this issue and train nets that perform better and are less overfitted to the training data.

This work presents two contributions:

- We propose a method that fully automatically generates labeled endoscopic image data. The additional image data is extracted from frames of endoscopic images and those images with insufficient image quality are discarded. To the best of our knowledge, this has not been done before in literature.
- We train CNNs on the old, original database and the new, enlarged database and compare their results.

This work was supported by the Austrian Science Fund, TRP Project 206.

[1] G. Wimmer and A. Uhl are with the University of Salzburg, Department of Computer Sciences, Jakob Haringerstrasse 2, 5020 Salzburg, Austria {gwimmer, uhl}@cosy.sbg.ac.at

[2] M. Haefner is with the St. Elisabeth Hospital, Landstraßer Hauptstraße 4a, A-1030 Vienna, Austria

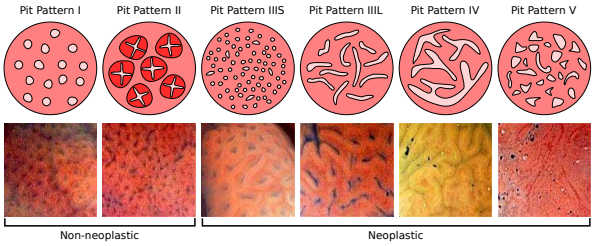


Fig. 1. The 6 pit pattern types along with exemplar images and their assigned classes in case of a two class (non-neoplastic vs neoplastic) differentiation

II. Colonic Polyps

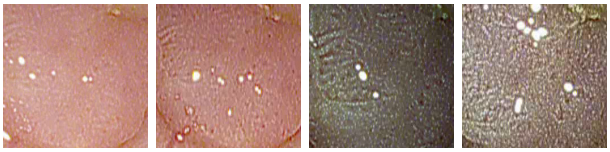
Colonic polyps are a rather frequent finding and are known to either develop into cancer or to be precursors of colon cancer. Colonic polyps are usually divided into hyperplastic, adenomatous and malignant polyps. In order to determine a diagnosis based on the visual appearance of colonic polyps, the pit pattern classification scheme was proposed by [5]. A pit pattern refers to the shape of a pit, the opening of a colorectal crypt. The various pit pattern types and exemplar (HM-endoscopic) images of the classes are presented in Fig 1. The pit pattern classification scheme differentiates among six types. Type I (normal mucosa) and II (hyperplastic polyps) are characteristics of non-neoplastic lesions, type III-S, III-L and IV are typical for adenomatous polyps and type V is strongly suggestive to malignant cancer.

In this work we use the two-classes classification scheme differentiating between non-neoplastic and neoplastic lesions. This classification scheme is quite relevant in clinical practice as indicated in [6].

Our original colonic polyp image database consists of manually extracted patches from frames of HD colonoscopic videos with high image quality. The patches are recorded using either white light (WL) endoscopy or the i-Scan technology. The i-Scan (Pentax) image processing technology [7] is a digital contrast enhancement method which consists of combinations of surface enhancement, contrast enhancement and tone enhancement. The three i-Scan modes operate as follows:

1) i-Scan1 augments pit pattern and surface details, providing assistance to the detection of dysplastic areas. This mode enhances light-to-dark contrast by obtaining luminance intensity data for each pixel and adjusting it to accentuate mucosal surfaces.
2) i-Scan2 expands on i-Scan1 by adjusting the surface and contrast enhancement settings and adding tone enhancement attributes to the image. i-Scan2 assists by intensifying boundaries, margins, surface architecture and difficult-to-discern polyps.
3) i-Scan3 is similar to i-Scan2, with increased illumination and emphasis on the visualization of vascular features. This mode accentuates pattern and vascular architecture.

In Fig. 2 we see an image showing an adenomatous polyp with WL endoscopy (a) and i-Scan (b,c,d)



(a) WL (b) i-Scan 1 (c) i-Scan 2 (d) i-Scan 3

Fig. 2. Images of a polyp using WL endoscopy and different i-Scan modes.

The i-Scan modes and WL can be switched on and off by the endoscopist during colonoscopy. Most of the videos contain sequences with all 4 imaging modalities (WL i-Scan 1,2,3).

The high definition (HD) colonic polyp image database was acquired by extracting patches of size $256 \times 256 \times 3$ from frames of HD-endoscopic (Pentax HiLINE HD+ 90i Colonoscope) videos. The database consists of patches gathered with 4 different imaging modalities (three different i-Scan modes (modes 1,2,3) and WL endoscopy). The database consists of 478 image patches (144 images showing non-neoplastic mucosa and 301 images showing neoplastic polyps) from 84 patients.

III. Frames from Endoscopic Videos

In this section we propose our unsupervised method to extract high quality image patches with label information from frames of endoscopic videos.

A. Movement Estimation

In endoscopic videos, the video capture device (the endoscope) is moving through the colon. So contrary to movies, its alone the video capture device that is moving and not the objects that are shown in the video. Since the movements are often very fast, it is quite hard to track the position of the regions shown in the endoscopic video. The rapid movements of the endoscope cause movement blur and fast changes of the distances from the camera to the mucosal wall (which causes sharp transitions from well focused sections of the video to sections of the video that are completely out of focus and hence quite blurry). Furthermore, there are no hard edges in the mucosal images and big parts of the videos are recorded out of focus because the distance from the endoscope to the mucosal wall is very often too high and sometimes also too low to be in the optimal focus range of the camera. That means big parts of the videos appear blurry. Additionally, the imaging modalities (WL, i-Scan 1,2,3) can change from one frame to the other in the videos.

This all makes it quite difficult to reliably track objects in the videos. Furthermore, major parts of the video do not exhibit enough image quality to be used to effectively differentiate between different types of polyps.

For the patches of our endoscopic image database, label information were provided by medical experts. As already mentioned in the introduction, we want to automatically generate additional image patches with label information. By tracking the region shown in the original image patch we are able to take more images of the considered region of interest.

So, our first task is to track the content shown in a 256 x 256 sized patch throughout the video (forward and backward through the video).

Since we are facing highly complex motion (e.g. position-variant transformations and parallax effects) in endoscopic videos, simple motion models are not sufficient to describe the motions between successive HD endoscopy video frames.

In this work we use the optical flow estimation by Black and Anandan [8], which is part of the implementation available for the work in [9]. This method is quite versatile when it comes to the estimation of arbitrary complex motions between images. This is mainly due to the fact that optical flow methods allow to estimate local motion, while simpler methods usually work well only with global motion.

We use the following notations to describe our proposed method: f_0 denotes the frame from which an image patch from our endoscopic image database was manually extracted, $p_0(\vec{x}_0)$ denotes the manually extracted patch from frame f_0 and $\vec{x}_0 = (x_0, y_0)$ denotes its position inside frame f_0 (the coordinate of the middle point of the patch). f_i denotes the i-th frame starting from f_0 (either forwards or backwards through the video) and $p_i(\vec{x}_i)$ denotes the patch extracted from f_i, where x_i is the tracked position of the region shown in $p_0(\vec{x}_0)$. The optical flow estimation is applied from frame (f_{i-1}) to frame (f_i) and not from f_0 to f_i because of the distinctly more accurate movement estimations in our experiments for estimating the movement from frame to frame. Movement estimation is applied to gray scale versions of the frames and image patches.

Although optical flow estimation usually works quite fine to track the content shown in a patch from frame to frame, it can fail in case of extreme motions, extreme image blur, and changing imaging modalities. Furthermore, errors in the movement estimation would add up the longer we track a region through the video.

To avoid any errors of the optical flow estimations, we apply a correlation based movement estimation as backup if the image patch $p_{i-1}(x_{i-1})$ is too different to the subsequent patch $p_i(x_i)$. More specifically, if the correlation coefficient between the patch $p_{i-1}(\vec{x}_{i-1})$ and patch $p_i(\vec{x}_i)$ (where $\vec{x}_i$ is the position tracked from $p_{i-1}(\vec{x}_{i-1})$ by means of the optical flow estimation) is beneath a threshold $C_t = 0.8$ (the correlation coefficient can range from +1 (for two identical patches) to -1 (for a patch and its inverse version)), then the position $\vec{x}_i$ of the patch p_i is re-evaluated by selecting the position that leads to the highest correlation coefficient with patch $p_{i-1}(\vec{x}_{i-1})$:

$$\vec{x}_i = (x_i, y_i) = \max_{(x,y)}(corr(p_{i-1}(x_{i-1}, y_{i-1}), p_i(x,y))), \quad \text{with}$$

$$corr(p_{i-1}(x_{i-1}, y_{i-1}), p_i(x,y)) = corr(q^{i-1}, q^i) = \frac{\sum_{m=1}^{256}\sum_{m=1}^{256}(q_{mn}^{i-1} - \overline{q}^{i-1})(q_{mn}^{i} - \overline{q}^{i})}{\sqrt{(\sum_{m=1}^{256}\sum_{m=1}^{256}(q_{mn}^{i-1} - \overline{q}^{i-1})^2)(\sum_{m=1}^{256}\sum_{m=1}^{256}(q_{mn}^{i} - \overline{q}^{i})^2)}},$$

where q_{mn}^i denotes the gray value of the pixel in the image patch q^i with position (m,n) and $\overline{q}^i$ denotes the medium gray value over all pixels in q^i.

This way of tracking is more time consuming than the optical flow estimation but it is also more accurate in the presence of strong blur and/or high camera movement.

Furthermore, we have four stop conditions to avoid errors in tracking the content of the patches. If one of the following stop conditions applies, then we stop tracking the position of the content in the patches any further since the risk of incorrect movement estimations becomes too high:

1) We stop if the estimated movement from one frame to the next one exceeds 50 pixels (Euclidean distance $d(x_{i-1}, x_i) > 50$).
2) We stop if two successive patches p_{i-1} and p_i differ too strongly. For this, we resize both patches to size 32×32. If the difference between the gray values of the downsized patches exceeds 10 in average, then the patches are considered as too different. This stop condition is applied to detect cuts in the video or to stop tracking if the content shown in the video changes too fast to enable a reliable tracking.
3) If the tracked position x_i of the patch p_i is so far outside of the frame f_i that the patch exceeds the border of the frame.
4) We stop at latest at the 300th iteration (maximal 300 iterations forward through the video and maximal 300 iterations backward through the video). Since the frame rate of the videos is 25 frames per second, this corresponds to tracking the content shown in the original patch p_0 for at most 12 seconds backwards and forwards through the video starting from frame f_0. We set that maximum number of iterations ($i \leq 300$) to avoid that small individual errors in the movement estimation sum up to a more significant error in the movement estimation.

We ensured by manual inspection of the automatically extracted patches, that all additionally extracted patches show the same content as shown in the original patch (p_0) (but with different viewpoints and scales). Hence, the label information of each original patch also applies to all patches p_i originated from the original patch p_0.

B. Image Quality Control

In order to differentiate between different types of polyps, the polyps and their pit-pattern structure have to be clearly visible. As already mentioned before, major parts of the video does not exhibit enough image quality to enable a differentiation between different types of polyps.

To ensure that only those automatically extracted image patches are further used to train automated diagnosis systems that enable a correct diagnosis, some criteria were determined to differentiate between images with high enough image quality and those images that are discarded because of limited image quality. The image quality tests are applied to grayscale versions of the (originally RGB) image patches. The following threshold values to differentiate between informative and non-informative patches were set so that the qual-

ity assessment of the automatically extracted image patches widely corresponds with the authors subjective opinion.

1) Reflections: If the number of overexposed pixels (gray value > 240) in a newly extracted patch exceeds 3500 (that is about one of 19 pixels), then the image patch is classified as uninformative.
2) Darkness: If the number of underexposed pixels (gray value < 45) in a newly extracted patch exceeds 4000 (that is about one of 15 pixels), then the patch is classified as uninformative.
3) Blur and visibility of texture structures: The image patch is subdivided into 10×10 pixel regions and standard deviations are computed for each of those regions. The highest (the top 20%) and the lowest (the bottom 20%) standard deviations are omitted and the mean value of the remaining standard deviations ($\overline{std}$) is computed as quality measure. The highest standard deviations are omitted because those outliers would heavily influence the mean value and since reflections can cause high standard deviations in the 10x10 pixel regions. The lowest standard deviations are omitted since it is not necessary that texture structures are clearly visible everywhere in the patch. It is sufficient if most parts of a patch are informative but it does not pose a problem if small parts of a patch are recorded out of focus. If $\overline{std}(p_i) < 5$, then the patch is classified as uninformative.
4) Blur and pit pattern structure: Our quality measure to rate the visibility of texture structures like the pit pattern structure is based on difference of Gaussians (DoG). We apply DoG by subtracting a Gaussian blurred image ($\sigma = 1$, filter size 5×5) from a stronger Gaussian blurred image ($\sigma = 3$ and filter size 9×9). Then each pixel value of the DoG image is replaced by its absolute value. The resulting non-negative DoG image of a grayscale endoscopic image patch highlights mucosal structures like the pit-pattern structure. Similar to the standard deviations and because of the same reasons, the highest 10% and the lowest 10% of the DoG values are omitted and the mean value ($\overline{DoG}$) is computed of the remaining DoG values. If $\overline{DoG}(p_i) < 2$, then the patch is classified as uninformative.
5) Comparison to the reference image patch p_0: Frames recorded with the i-Scan imaging modality show clearly more contrast and a better visability of mucosal structures than those frames that were recorded with traditional white light endoscopy. Furthermore, images showing healthy mucosa usually show less contrast than those images showing adenomatous polyps. So the quality measurements do not only respond to the quality of the frames but also to the used imaging modality and the shown content. There are even some original image patches that do not fulfill all of the before mentioned criteria (most of them show healthy mucosa and were captured using WL endoscopy). So we introduce an additional quality measure that balances the image patch quality with reference to the quality of the original image patch p_0. If an image patch p_i does not fulfill one of the before mentioned quality thresholds, but if the considered quality measure of the image patch p_i is at most 5% worse than the quality measure of the original patch p_0, then the image patch p_i is still classified as informative. We do not want to throw away patches that are only very slightly worse in terms of image quality than the reference patch p_0. On the other hand, if an image patch p_i is clearly more blurry and if the mucosal texture structures are clearly less visible than for the original image patch ($\overline{DoG}(p_i) < 0.6 \times \overline{DoG}(p_0)$ or $\overline{std}(p_i) < 0.6 \times \overline{std}(p_0)$), then the image patch p_i is classified as non-informative, even if it fulfills all before mentioned criteria.
6) Movement: If the estimated movement $d(x_{i-1}, x_i) > 15$, then the patch p_i is classified as uninformative. Image patches with higher movement almost always suffer from movement blur.
7) Duplicity: If $corr(p_i, p_{i-1}) > 0.95$, then p_i is classified as uninformative. If there is hardly any difference between two patches than we only use one of them. This step will be later motivated in Section IV.

In Figure 3 we show some examples of informative and non-informative patches. On the left side we see examples that were accepted as informative image patches and on the right side we see examples that originate from the same original patches as the image patches to their left and that were discarded because of insufficient image quality.

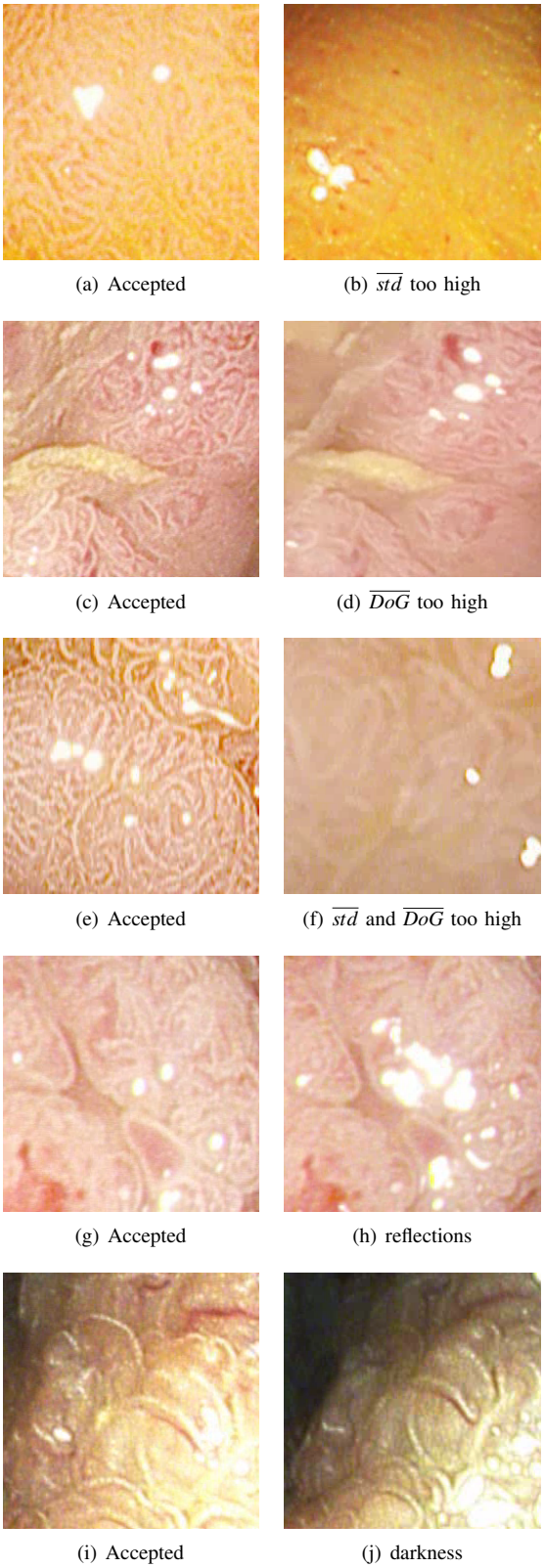
The original HD colonic polyp image data base consists of 478 image patches. The enlarged version of the database using our proposed method includes 18969 image patches. So in average, about 39 image patches were generated from one patch of the original database. In case of 40 original patches, no additionally patches with sufficient image quality could be generated. The maximum number of generated image patches from one original patch is 270. The standard deviation over the generated patches per original frame is 43, so there are huge variations in the number of additionally generated image patches per original patch.

IV. CNN TRAINING

This section gives the implementation details for CNN training and the description of the employed nets.

We employ two nets in this work, the VGG-f net [10] and the VGG-16 network [11]. The VGG-f net consists of five convolutional layers and three fully connected layers with a final SoftMax classifier. The VGG-16 net consists of 13 convolutional layers subdivided in 5 convolutional blocks (where each of the 2-3 convolutional layers inside of a block have the same number and sizes of filters) and three fully connected layers with a final SoftMax classifier.

The two nets are trained from scratch and we randomly initialize the coefficients of the layers based on [12]. The last fully connected layer is acting as soft-max classifier and



(a) Accepted (b) $\overline{std}$ too high

(c) Accepted (d) $\overline{DoG}$ too high

(e) Accepted (f) $\overline{std}$ and $\overline{DoG}$ too high

(g) Accepted (h) reflections

(i) Accepted (j) darkness

Fig. 3. Examples of informative (left) and uninformative (right) patches originating from the same original patch.

computes the training loss (log-loss). Stochastic gradient descent (SGD) with weight decay ($\lambda = 0.0005$) and momentum ($\mu = 0.9$) is used for the training of the models.

As already mentioned in the introduction, CNN training is applied to two different image databases. We use the original colonic polyp image database and the new enlarged database for training the CNNs.

Training is performed on batches of 128 images each, which are for each iteration randomly chosen from the training data and subsequently augmented (see Section V).

In case of the original database, the 128 images are for each iteration randomly chosen from the training data.

In case of the enlarged database, the image patches that originate from one original image patch are quite similar and we need to consider that fact for the selection of training images per batch. If we would randomly choose images for training like for the original database, then the images of some patients would be used very often for training (those with a lot of automatically extracted image patches), whereas images of other patients (those with a low number of automatically extracted image patches) would be used much less for training. On the other hand, if we would randomly choose the patients and then randomly choose one image per patient for training, then we would not really profit from the high amount of additional image data since the chance that a specific image is chosen for training from a patient with a high number of automatically extracted patches is very low (compared to an image of a patient with a low number of automatically extracted patches). So we decided to first randomly choose one of the original patches, whereat the probability of those original patches varies with the amount of images that originate from them. More specifically, the probability to choose one original patch is multiplied by factor f_n with $f_n = \sqrt[3]{n}$, where n is the number of patches originating from one original patch (including the original patch). After one original patch is chosen, we randomly choose an image patch that is originating from the considered original patch (including the original patch itself), where each image patch has the same chance to be chosen. So for example, if the number of image patches originating from original patch A is 100 ($n(A) = 100$) and 1 for patch B ($n(B) = 1$, only the original patch itself), then the probability of choosing any image belonging to patch A for training is $\sqrt[3]{100} = 4.64$ times higher than choosing the image belonging to patch B. On the other hand, the probability that the one image of patch B is chosen for training is higher by factor $100/4.64 = 21.54$ than the probability that one specific image of patch A is chosen. This approach to select the training images was the reason to discard image patches that are very similar to other image patches (the duplicity criteria in Section 3).

V. Experimental Setup

Our employed nets require input image sizes of $224 \times 224 \times 3$. The image data is normalized by subtracting the mean image of the training portion. We then linearly scale each image within $[-1, 1]$.

We use data augmentation to increase the number of images for training and validation. Augmentation is applied

CNN architecture	Training Database	
	Original Database	Enlarged Database
VGG-16	76.2 (6.6)	86.9(7.2)
VGG-f	83.9(5.7)	84.2(3.8)

TABLE I

MEAN ACCURACIES OVER THE 10 FOLDS AND THE STANDARD DEVIATIONS (IN BRACKETS) FOR THE TWO NETS ON BOTH DATABASES

to the batches of images extracted for training. The augmentation is based on cropping one sub-image (224×224 pixels) from each image patch with randomly chosen position. Subsequently, the sub-image is randomly rotated (0°, 90°, 180° or 270°) and randomly either flipped or not flipped around the horizontal axis. Validation is performed using a majority voting over five crops from the validation image using the upper left, upper right, lower left, lower right and center part.

We perform 10-fold cross-validation to achieve a stable estimation of the generalization error, where each of the 10 subsamples of an image database consists of the images from about 10% of the patients of a database. All images of one patient are in one subsample and each subsample consists of about 10% of the total images of a database (at least in case of the original database there are about 10% of the images in one subsample, in case of the enlarged database the number of images per subsample can vary depending on the number of additionally extracted patches per patient). All nets are trained using the training portion of our data corpus (9 of the 10 subsamples). The final validation is performed on the left-out part. That means for each database and for each of the two network architectures, ten different nets are trained, one for each of the 10 folds. To ensure the highest possible comparability between the results of the two databases, we used the same folds for both databases. That means the training data corpus of a fold consists of image data from the same patients for both databases (whereat the patients contain distinctly more images in case of the enlarged database). For both databases, validation is performed on the validation data corpus of the original database for each fold. That means validation is always performed on the same images for both databases (to have comparable results), whereas training is always performed on a much bigger data corpus in case of the enlarged database, but from images of the same patients as for the original database.

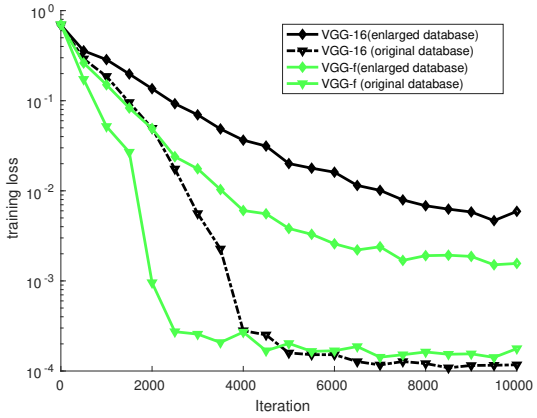
In our experiments, we compute the overall classification rate (OCR) for each fold and report the mean OCR over all 10 folds with the respective standard deviation.

A. Results and Discussion

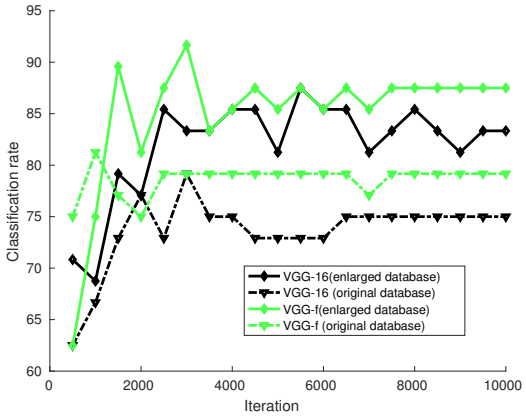
The results of the experiments using our two nets trained on the original colonic polyp database as well as trained on the enlarged version of the database are presented in Table I.

As we can see in Table I, the VGG-16 net clearly profits from additional training data (86.9% vs 76.2%). The VGG-f net on the other hand did not really profit from the additional training data. The results only increased by 0.3% using the enlarged database. A possible reasons for the different outcomes of the two nets is that the VGG-16 net has much more layers and parameters to be learned. The original database was not large enough to properly train this big net. The much smaller VGG-f net on the other side was not able to profit that much from the additional training data. The overall quality of the automatically extracted image patches is slightly worse compared to the original patches. So we guess that the difference in the quality of the training images (from the enlarged database) compared to the evaluation images (from the original database) is the reason for the only very small improvement of the results for the VGG-f net using the enlarged image database for training.

In Figure 4 we see the the training losses and the validation accuracies during training (fold 1 of 10) for the two nets on both databases. We can observe that for both net



(a) loss



(b) accuracy

Fig. 4. Comparison of the training losses and the validation accuracies during training on the two databases.

architectures, the training loss decreases much faster on the smaller original database and reaches much lower levels at the end of training as for training on the enlarged database. The validation accuracies stagnate earlier for training on the original database as for training on the enlarged database.

This all indicates that the nets are more overfitted to the training data corpus in case of the smaller original database.

VI. Conclusion

In this work we presented an approach to increase the amount of labeled image data by a fully automated system that extracts image patches of endoscopic video frames from mucosal regions where label information is available. In that way we were able to increase the number of images by factor 40 and hence distinctly increase the amount of training data for automated diagnosis systems. Care was taken that only images with sufficient image quality and clearly visible mucosal texture structures were extracted. We showed that the increased number of training images can drastically improve the performance of CNNs. The mean accuracy of the VGG-16 net increased from about 76 % using the original database to nearly 87% using the enlarged image database for training. The results of the much smaller VGG-f net did only slightly improve using the additional training data. We furthermore showed that the increased number of training images reduces the overfitting to the training data corpus.

References

[1] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, May 2016.

[2] J. Oh, S. Hwang, J. Lee, W. Tavanapong, J. Wong, and P. C. de Groen, "Informative frame classification for endoscopy video," *Medical Image Analysis*, vol. 11, no. 2, pp. 110 – 127, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S136184150600079X

[3] M. H. Eduardo Ribeiro, Andreas Uhl, "Colonic polyp classification with convolutional neural networks," in *Proceedings of the 29th IEEE International Symposium on Computer-Based Medical Systems (CBMS'16)*, June 2016, pp. 253–258.

[4] E. Ribeiro, A. Uhl, G. Wimmer, and M. Häfner, "Exploring deep learning and transfer learning for colonic polyp classification," *Computational and Mathematical Methods in Medicine*, vol. 2016, p. Article ID 6584725, 2016.

[5] S.-E. Kudo, S. Hirota, T. Nakajima, S. Hosobe, H. Kusaka, T. Kobayashi, M. Himori, and A. Yagyuu, "Colorectal tumours and pit pattern," *Journal of Clinical Pathology*, vol. 47, pp. 880–885, 1994.

[6] S. Kato, K.-I. Fu, Y. Sano, T. Fujii, Y. Saito, T. Matsuda, I. Koba, S. Yoshida, and T. Fujimori, "Magnifying colonoscopy as a non-biopsy technique for differential diagnosis of non-neoplastic and neoplastic lesions," *World Journal of Gastroenterology*, vol. 12, no. 9, pp. 1416–1420, March 2006.

[7] S. Kodashima and M. Fujishiro, "Novel image-enhanced endoscopy with i-scan technology," *World Journal of Gastroenterology*, vol. 16, no. 9, pp. 1043–1049, 2010.

[8] M. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, pp. 75–104, 1996.

[9] S. Deqing, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*, 2010, pp. 2432–2439.

[10] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5*, 2014.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[12] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10)*, vol. 9, 2010, pp. 249–256.

Bringing classical Robot Vision descriptors to Deep Learning*

Jean-Baptiste Weibel, Timothy Patten, Michael Zillich and Markus Vincze

Abstract— **Robot vision still relies heavily on classical hand-crafted features because of their demonstrated robustness. Recent advances in Deep Learning have been drastically outperforming such classical approaches on images, however, transferring this success to 2.5D data in a robust manner is still an open question, both because of the challenges introduced by an additional dimension and the lack of non-artificial large 3D dataset. In this work, we demonstrate the benefits of using the PointNet[8] architecture to improve upon any histogram-based descriptor. In particular, we introduce a network using a global shape descriptor and a local descriptor that is directly applicable to a point cloud, and that keeps the robustness of the original descriptors (rotation invariance, robustness to variable point density and occlusion). We obtain 83 % accuracy on the ModelNet 40 dataset, using 10-100x less parameters than some competing methods.**

I. INTRODUCTION

Since the emergence of cheap and reasonably precise 2.5D sensors, such as the Kinect, 3D object classification has been intensively studied. It is an essential task for any indoor robot. Many descriptors have been created for this purpose specifically for such sensor. They all use point clouds, which became the de facto data representation for robotic tasks.

The state-of-the-art deep learning methods used in general computer vision have not been widely adopted by the field, partly because they do not mix well with the unstructured representation that is a point cloud, but also because large datasets acquired with such sensor are still quite rare, which limits the direct applicability of such methods.

This problem can be avoided altogether by using artificial computer generated models to train a neural network, but compared to 3D artificial models, data acquired by robots tends to be noisier in a few characteristic ways:

- unaligned objects
- occlusion
- outliers points
- sensor noise

It is impossible to make any assumption about the pose of objects in the scene. As such, the only two options are to use rotation-invariant features, a path chosen by most classical descriptors, or to align the data and compute features on aligned models. If a large body of work exist regarding the computation of robust local reference frame, aligning full objects with noisy data is a challenging problem, and even though they can deal with sensor noise to some extent, most of these methods are quite sensitive to outliers or occlusion. Occlusions can be caused either by other objects in front of our object of interest, or by viewpoint being unaccessible to the robot. Outliers, on the other hand, are often remnants from a previous segmentation step. Finally, sensor noise is inevitable, and tends to increase with the distance to the sensor. Robustness to variable point density and variable noise is therefore important.

Most deep learning architectures developed so far do not consider all type of noise. In particular, most of the recent and best performing works are assuming aligned models, which is, as discussed before, non-trivial to obtain in a robotic context. These noise sources have all been studied with more classical approaches which led to robust descriptors.

We present in this paper a method to bridge the gap between those two worlds. Our proposed architecture makes use of the PointNet[8] architecture to replace the histogram with a learned probabilistic version, making it possible to learn end-to-end architectures that work on the original feature space. Any histogram-based descriptor can be improved this way, keeping the original features advantages, such as rotation invariance (no alignment necessary), robustness to occlusions and variable point density.

Our contribution is the introduction of a general method to improve the descriptiveness of any histogram-based descriptor through learning while keeping their robustness. We also showcase this method on an ESF-like global shape descriptor [18] coined L-ESF and a SHOT-like local descriptor [16] coined L-SHOT.

II. RELATED WORK

A. 3D Classification

Unsurprisingly, deep learning techniques dominate the field of 3D classification. Due to the multiplicity of data representation available when dealing with 3D data, the approaches can differ drastically. We present here a non-exhaustive list focusing on the most common and most promising architecture.

The most straightforward way to apply convolutional neural networks (CNN) to 3D data is to start by extracting 2D views from the full model. These depth maps can then be fed to any available CNN. Many mappings from single-channel to three-channel representation have also been developed[3][2], thus avoiding the issue of the lack of large dataset for training through the re-use of images-learned features. Once a representation for each view is computed, different schemes for pooling them have been developed, from view pooling [15] to more complex view-set reasoning

*This research has received funding from the European Community's Seventh Framework Programme under grant agreement no. 610532, SQUIRREL.

All authors are with the Vision4Robotics group (ACIN - TU Wien), Austria {weibel, patten, zillich,prankl, vincze}@acin.tuwien.ac.at

[17]. While performing very well in practice, thanks to the advances of 2D CNN, these approaches generally lose all information between the views, and raise the problem of view selection. In an indoor robotic context, not all viewpoints are accessible. They tend to use a lot more parameters than other methods.

It is also possible to extend the design of 2D CNN to 3D CNN, and to learn features over voxel grids. This direction has been explored in [19] but also [7]. While this direction is a meaningful extension of CNN, they bring with them two main problem inherent to the design of the network. First, the additional dimension is an optimization burden, and because of the explosion of the number of parameters, such approaches have been forced to work on relatively coarse voxel grids, making the data much less informative. Partitioning the space as in [11] is a way to tackle this issue. Second, because of the design of the convolution, they are not rotation invariant. They would thus need either to separately learn each orientation of the object, making learning harder, or align the model beforehand, which is in itself hard to perform robustly.

The last direction is to work directly on point clouds. This data representation loses any explicit neighbourhood information. One way to get around it is to rely on the creation of a KD-Tree over the set of points as in [6]. However, the KD-Tree itself depends on the orientation of the model. Moreover, a reasonably large noise can also affect the structure of the KD-Tree. Another option is to rely only on the implicit information of the coordinates, as in [8] or [10]. The architecture from [8] will be referred to as the PointNet architecture. By drastically expanding the feature dimension of the coordinates (from 3 to 1024 in multiple steps), the network can then learn up to 1024 function expressing a probability of presence in a certain area of the space. This approach also requires aligned data. The author relies on a spatial transformer network[5], to learn a data dependent alignment. Learning such an alignment over a whole object, however, is vulnerable to outliers and occlusion in two ways: the alignment itself will be affected and the representation will then be computed on a not only incomplete but also potentially misaligned set of points. The same author also demonstrated the possibility of hierarchical feature learning with the PointNet architecture[9], using a second PointNet to learn over a set of local descriptors. In [10], the author follows a similar path, except that it assumes that the models are already aligned, and improve on the layers of the network by subtracting a weighted version of the maximum activation value over the whole set for each filter.

B. Robotic classification

As in classical vision, handcrafted features were carefully developed to capture either local, regional or global information.

To capture local information, a whole family of descriptors were created following variants of the scheme introduced by the Point Feature Histogram (PFH) and Fast Point Feature Histogram (FPFH) [13]. They both rely on a set of angle between the normal of a point of interest and its neighborhood.

The best performing local handcrafted descriptor is probably the Signature of Histograms of OrienTations (SHOT) descriptor [16]. It first aligns the neighborhood along a local reference frame. This local reference frame is computed as a repeatable representation of the statistical distribution of points. Once aligned, the sphere around the point of interest is divided in 32 spatial bins, and a histogram is computed over each one of them.

The idea behind PFH/FPFH was extended to capture viewpoint specific global information, by considering angles with the vector going from the viewpoint to the centroid of the considered object. This approach was coined Viewpoint Feature Histogram (VFH)[12]. On the other hand, the Ensemble of Shape Functions (ESF) [18] relies on simple randomly sampled geometrical construct, such as pairs and triangles, instead of relying on the direct neighborhood of a point of interest.

Through their use of randomization, histogram as a pooling strategy, or other methods, all these descriptor have proven their robustness to most of the type of noise encountered in data acquired by a robotic platform. However, their descriptiveness is no match for more modern deep learning approaches, and as such, it is typically challenging to reach state-of-the-art results using such descriptors.

III. Learned Descriptors for Robotic Classification

Considering both the desire for robustness and the need for good accuracy and generalization, we propose to learn descriptors using the PointNet architecture, as a histogram-like pooling solution thus providing robust yet descriptive features.

In this section, we will first detail how we propose to learn such a representation, then provide two concrete applications, one on a global shape representation coined Learned-ESF (L-ESF), and one on a local shape representation coined Learned-SHOT (L-SHOT).

A. Learning histogram-like features

As illustrated in the previous section, most of the descriptors used in robotic rely on histograms, which is a crucial part of their robustness. Approaching noisy data as a set is a good trade-off between the amount of information discarded and the robustness of the description, and a histogram is the most straightforward way to approach set pooling. The PointNet architecture, by working on one data point at a time but optimizing over the whole set, provides a structure to learn functions that activate when a data point is present nearby. In [8], those functions behave like probabilistic bins over the Euclidean space. The global optimization ensures that those functions are optimally spread. An ensemble of such functions can therefore be seen as a probabilistic histogram that is trainable end-to-end. This idea can be extended to any other space.

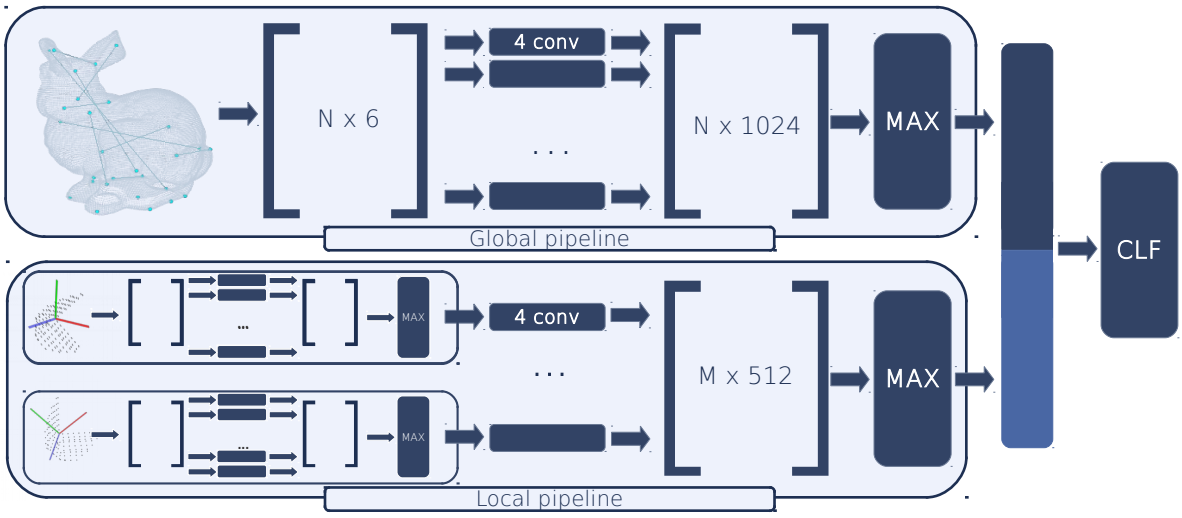


Fig. 1. System overview of our proposed method. The classifier layer is composed of two fully connected layers. N is the number of pairs sampled, M is the number of local descriptors sampled.

In this setup, the spatial transformer network of the original PointNet architecture is not necessary as we are not working on Euclidean coordinates. Instead, we use 4 one-dimensional convolutional layers, with a kernel size of 1, and a progressively increasing number of filters, and then a max-pooling layer to implement our histogram-like feature learning scheme. Each of the layer include a batch normalization step [4]. We also subtract a weighted version of the maximum value of a given filter over the whole set to each output, as described in [10]. We choose to use ReLU as an activation function.

B. Global Pipeline - Learned ESF

In a first step, we want to extract robust shape informations globally. This pipeline is strongly inspired by the ESF descriptor [18], which is to our knowledge the best performing global handcrafted shape descriptor that does not depend on the viewpoint choice.

The ESF descriptor computation first samples randomly pairs and triplets of points and extracts handcrafted features for each one. It then creates various histograms over them. For a pair of points, the features chosen are the distance between them, and the percentage of the line from one point to the other that is filled with surface points. This is done by tracing the line with the help of the 3D Bresenham algorithm in a voxel grid of size 64x64x64. For triplets of points, the angles of the triangle and the area covered by the triangle is computed. The robustness of this descriptor comes both from the use of histogram and its randomness.

In our pipeline, we decided to focus on pairs of points. On top of the features extracted for the ESF descriptor, we also draw inspiration from the Point Pair Features[1], and Fast Point Feature Histogram[13] descriptors. All the features we extract from each pair are rotation invariant, thus making the whole pipeline rotation invariant as well. That allows us to not have to rely on any form of alignment.

Based on these considerations, after scaling our point cloud to the unit sphere, our global shape descriptor computation starts by randomly sampling pairs of points. For $\overrightarrow{p}_1$ and $\overrightarrow{p}_2$ the two sampled points, and $\overrightarrow{n}_1$ and $\overrightarrow{n}_2$ their respective normal vectors (which are assumed to be normalized), we first extract the distance d between the points as in (1).

$$d = \left\| \overrightarrow{p}_1 - \overrightarrow{p}_2 \right\| \tag{1}$$

We also consider the cosine similarity between the normals (2), and the absolute value of the cosine similarity between the vector $\overrightarrow{p}_1 - \overrightarrow{p}_2$ and each of the normals (3)

$$cos(\angle(\overrightarrow{n}_1, \overrightarrow{n}_2)) = \overrightarrow{n}_1 . \overrightarrow{n}_2 \tag{2}$$

$$\left| cos(\angle(\overrightarrow{p}_1 - \overrightarrow{p}_2, \overrightarrow{n}_{\{1,2\}})) \right| = \left| \frac{(\overrightarrow{p}_1 - \overrightarrow{p}_2) . \overrightarrow{n}_{\{1,2\}}}{\left\| \overrightarrow{p}_1 - \overrightarrow{p}_2 \right\|} \right| \tag{3}$$

Finally, as in the ESF descriptor, we consider the percentage of $\overrightarrow{p}_1 - \overrightarrow{p}_2$ that is on the surface of the object. We follow the PointNet[8] architecture for the classification of the global pipeline: the set of features is first fed to a convolutional neural network, always operating on a single pair at a time. After enlarging the feature dimension, the set is then max-pooled.

Due to the intractable number of potential pairs, we tend to lose any information relative to finer structures during the sampling step. For this reason, we introduce a local descriptor pipeline.

C. Local Pipeline - Learned SHOT

As mentioned in the previous section, we need a way to capture finer structure to improve the descriptiveness of our approach. However, computing local descriptor densely

would be wasteful. As such, we need a way to direct our sampling of local patches.

1) Attention Model: Given the nature of our global shape, we choose to base our attention model on the statistical consistency of the normal orientations. Indeed, finer structures are characterized by higher angle between neighboring normals. However, we cannot solely rely on the local variations of the angle between normals, otherwise, any rounded surface would be considered salient. Consider the example of a flower pot: local descriptors on the leaf are probably more informative than redundant local descriptors on the pot itself.

We therefore chose to first create a histogram of angles between normals and their neighbors. We then normalize the histogram such that its sum is equal to one. With $\|P\|_0$ the number of non-zeros elements in our histogram and P_k the k-th entry in the histogram, we apply the following transformation:

$$\tilde{P}_k = \begin{cases} \|P\|_0 - P_k \text{ if } P_k \leq \|P\|_0 \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

That allows us to only capture the statistically significant angles and remove all highly recurrent angles. We can then reproject the saliency value for each point by simply looking up the saliency value corresponding to the angle, and then summing over the whole neighborhood.

The actual sampling of salient points is done using a Poisson sampling. For each point sampled, we set a neighborhood of twice the number of points used for the descriptor computation to be invalid for sampling. However, we use the mean distribution between the distribution described before, and a uniform sampling to better capture the model specificities.

2) Local Descriptor: To design our local descriptor, we followed the design of the SHOT descriptors[16]. The first step for its computation is to compute a robust and repeatable local reference frame (LRF) to align our point against. Then, the sphere of all neighbors is divided in 32 bins, a histogram of normal angles being computed for each of them.

Considering the success of the PointNet architecture [8] over raw point coordinates, we chose a similar path when designing our local descriptor. We first transform all the neighboring points in the LRF computed as in the SHOT descriptor, instead of relying on a learned alignment, and use our sampled salient point as the origin. We can then directly feed the coordinates of our points to a smaller scale PointNet architecture. Indeed, by not having to include a spatial transformer network, we can drastically reduce our number of parameters.

IV. EXPERIMENTS

A. Invariance and Robustness

We designed our network around robust features. To demonstrate their intrinsic power, we devised a set of transformations applied to the input point cloud, and report the corresponding accuracy. **It is important to note that those experiments are performed without re-training the network, which is only trained on clean data.** The following experiments demonstrate that the network carry over the robustness of the chosen features. We report the results of our proposed architecture in comparison to the original descriptors, whose robustness has already been proven. No experiments are required regarding rotation invariance, as all features fed to the network are rotation invariant.

1) Point Density robustness: We want to evaluate how well our network behave depending on the point density of the point cloud. We chose to randomly remove points from the original instead of a more structured form of downsampling. The results of the experiments are reported in the figure IV-A.1

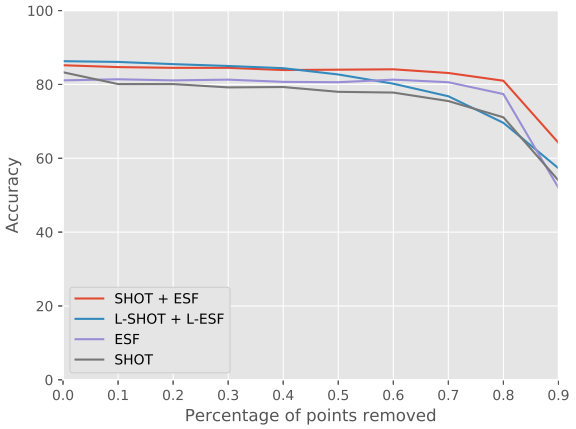


Fig. 2. Influence of the point density on the accuracy

2) Occlusion robustness: To simulate occlusion, we chose to remove a neighborhood around a randomly sampled point. The size of the neighborhood corresponds to the percentage of the points being removed. The results are reported in the figure IV-A.2

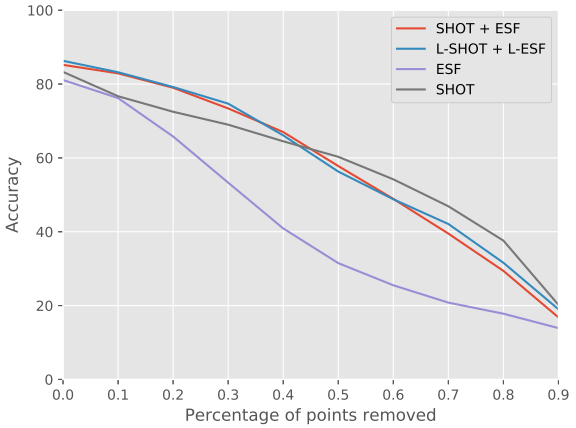


Fig. 3. Influence of occlusion on the accuracy

3) Sensor noise: Working with full models prevents us from using a realistic sensor noise model. However, we can still model a generic noise by adding Gaussian noise. We choose the standard deviation of our Gaussian noise as a proportion of the longest distance between points in the point cloud. The results are reported in the figure IV-A.3

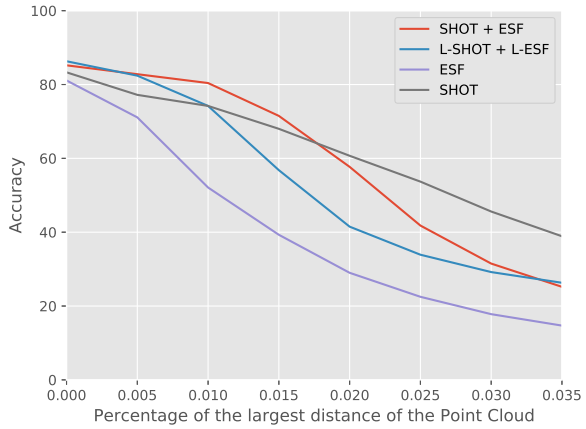


Fig. 4. Influence of sensor noise on the accuracy

B. ModelNet

We perform our evaluation on the ModelNet dataset[19], a CAD model dataset which has two variants, ModelNet40 with 40 classes, and ModelNet10 which is a 10 class subset of the ModelNet40.

Our method was designed to be used on Point Clouds, so as a pre-processing step, we extract Point cloud from the original CAD models by first down-sizing the CAD model to the unit sphere, and then sampling a point every 1mm of the surface of the point cloud for ModelNet40, and every 2mm for ModelNet10. Due to the random nature of our algorithm, we average the accuracy over 5 run on the test set.

For the global pipeline, we sample 5000 pairs on ModelNet40, and 3200 on ModelNet10. For the local pipeline, we sample 50 salient points drawn using Poisson sampling from a distribution which is the average of a uniform distribution and our attention model described above.

For the SHOT, ESF and SHOT+ESF results, we simply replace our learned version of the descriptor with the original descriptor, as implemented in [14]. We keep the same classification layers, and still merge the set of SHOT descriptors with a PointNet architecture for a fair comparison.

The accuracy results can be found in table I for ModelNet

TABLE I
ACCURACY ON THE MODELNET DATASET [19]

	MN10	MN40	Input
3DShapeNets[19]	83.5	77	Voxel grid
VoxNet[7]	92	83	Voxel grid
PointNet[8]		89.2	Point Cloud
Kd-Networks[6]	94	91.8	KD-Tree
MVCNN[15]		90.1	Views
SHOT	83.3	73.9	Point Cloud
ESF	81.1	70.4	Point Cloud
SHOT+ESF	85.2	76.9	Point Cloud
Ours	**86.3**	**83.0**	Point cloud

C. Discussion

As described above, we achieve results that are better or on par with the methods based on voxel grids and classical descriptors, but worst than the view-based methods. It should however be noted that view-based methods uses architecture with a lot more parameters (10-100 millions compared to around 1 million for ours). Compared to point cloud based methods, Kd-Networks performs best but requires aligned models, so it is not as robust, and PointNet learns an alignment which is itself sensitive to occlusion and outliers.

In the case of ModelNet10, most of the misclassification are caused by confusion between `night_stand` and `dresser`, and between `table` and `desk`. In the case of ModelNet40, most of the misclassification are caused by confusion between `flower_pot` and `plant`, on top of the confusion made over the ModelNet10 dataset. A deeper observation of the CAD models in each of these class show that those mistakes are quite reasonable and shows promising potential application of this architecture as a robust generic shape feature.

In our study of the robustness of our model, we can see that most of the desirable properties of the classical descriptors are kept. Only the robustness to Gaussian noise seems lower. This is due to the setup of our experiment: by not retraining our network with any kind of data augmentation, this study focuses on the intrinsic properties of the features used, rather than on the already demonstrated learning capabilities of neural network. However, in a classical descriptor, the robustness to Gaussian noise mostly comes from the use of a histogram, and if our learned equivalent of a histogram has not faced noisy data during training, it is unlikely to cope with it during testing.

The key benefit of our method is its robustness: it carries over the benefits of classical handcrafted features, and it does not require any alignment of the models, as the feature extracted are themselves rotation-invariant. This allows us to use a more compact network, as we do not require parameters for a spatial transformer network, or additional parameters that would be necessary to cover the representation over many different orientations. Through the use of randomization, smaller parameter number and batch normalization, our model is less likely to overfit as every representation of a given instance is slightly different, both during training and testing.

V. CONCLUSIONS AND FUTURE WORK

We have shown in this paper a novel architecture that provide robust yet descriptive shape features. Moreover, the scheme devised in this paper can be used to adapt any local or global histogram-based handcrafted feature into a learned descriptor, thus providing better task specific performance and end-to-end learning.

The flexible nature of the architecture also allows its use in both a single and multi-view scenario. In the case of multiple views, it can perform efficiently, as information already computed over previous sets can easily be included to the next step through the max-pooling layer over the whole set, all that while still preserving inter-views information.

Finally, by keeping track of the indices used during the max pooling step, we can gather interpretable information regarding the contribution of local structures. As an extension, such local descriptors could be use for correspondence problems, such as pose estimation which can be done from sampling pairs, as demonstrated in [1].

REFERENCES

[1] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model Globally, Match Locally: Efficient and Robust 3d Object Recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[2] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal Deep Learning for Robust RGB-D Object Recognition," *arXiv:1507.06821 [cs]*, July 2015, arXiv: 1507.06821. [Online]. Available: http://arxiv.org/abs/1507.06821

[3] S. Gupta, R. Girshick, P. Arbelez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *Proceedings of the European Conference on Computer Vision*, ser. ECCV'14, 2014.

[4] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *International Conference on Machine Learning*, June 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[5] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial Transformer Networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2017–2025. [Online]. Available: http://papers.nips.cc/paper/5854-spatial-transformer-networks.pdf

[6] R. Klokov and V. Lempitsky, "Escape from cells: Deep kd-networks for the recognition of 3d point cloud models," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[7] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 922–928.

[8] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3d Classification and Segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5099–5108. [Online]. Available: http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space.pdf

[10] S. Ravanbakhsh, H. Su, J. Schneider, and B. Poczos, "Deep Learning with Sets and Point Clouds," *International Conference on Learning Representations (ICLR)*, 2017.

[11] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 2155–2162.

[13] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, ser. ICRA'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1848–1853. [Online]. Available: http://dl.acm.org/citation.cfm?id=1703435.1703733

[14] 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.

[15] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proc. ICCV*, 2015.

[16] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proceedings of the 11th European Conference on Computer Vision Conference on Computer Vision: Part III*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 356–369. [Online]. Available: http://dl.acm.org/citation.cfm?id=1927006.1927035

[17] C. Wang, M. Pelillo, and K. Siddiqi, "Dominant set clustering and pooling for multi-view 3d object recognition." in *Proceedings of British Machine Vision Conference (BMVC). 2017*, 2017.

[18] 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.

[19] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1912–1920.

Towards ScalableFusion: Feasibility Analysis of a Mesh Based 3D Reconstruction

Simon Schreiberhuber[1], Johann Prankl[1] and Markus Vincze[1]

***Abstract*— This work describes a novel real time approach for creating, storing and maintaining a 3D reconstruction. Previous approaches for reconstruction attach one uniform color to every geometric primitive. This one-to-one relationship implies that even when geometrical complexity is low, a high resolution colorization can only be achieved by a high geometrical resolution. Our contribution is an approach to overcome this limitation by decoupling the mentioned relationship. In fact newer, higher resolution color information can replace old one at any time without expensively modifying any of the geometrical primitives. We furthermore promise scalability by enabling capture of fine grained detail as well as large scale environments.**

I. Introduction

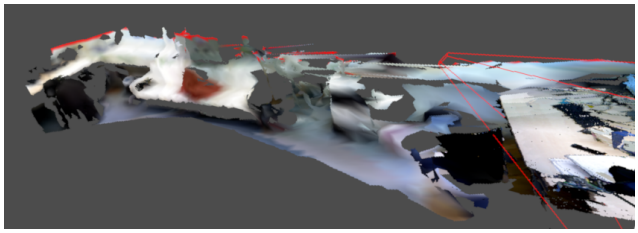
When mapping 3D environments based on the input of an RGBD sensor two steps are usually executed simultaneously.Localization, in which the camera position is tracked relative to the reconstruction or keyframes and the reconstruction itself. This process of Simultaneous Localization And Reconstruction (SLAM) aims to produce a dense representation of reality which finds adaption in augmented reality, robotics and other fields.

One of the first reconstruction algorithms introduced by Izadi et al. was KinectFusion [5], which maintains a volume in form of a 3D grid. In this approach, the grid is populated with values of a Truncated Signed Distance Function (TSDF) indicating where the closest surface resides. The initial implementation is only able to map small volumes of fixed position, size and resolution. By dynamically changing the position of the active reconstruction volume, Kintinuous [10] extends the basis algorithm and enables the reconstruction of bigger scenes. The use of voxel hashing [8] allows higher resolution reconstructions by reducing the memory footprint required for the reconstruction volume. KinectFusion spawned further notable expansions like DynamicFusion [7], which introduces a warp-able volume to reconstruct non rigid objects.

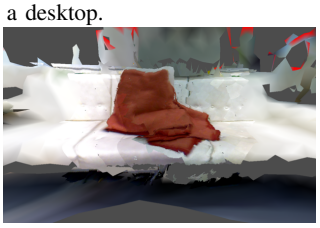
Another thoroughly researched approach is made popular by ElasticFusion [11] where the captured points are stored as surfels, small discs with diameter, orientation, position and color. This allows to store surfaces with varying spatial resolution depending on which distance was perceived by the sensor.

[1]All authors are with the Vision4Robotics group, Automation and Control Institute (ACIN), TU Wien, Austria {schreiberhuber, prankl, vincze}@acin.tuwien.ac.at

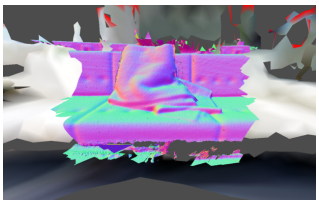
This work is supported by the European Comission through the Horizon 2020 Programme (H2020-ICT-2014-1, Grant agreement no: 645376), FLOBOT.



(a) Ongoing reconstruction of an office. To the left all the geometry is presented in low detail. To the right the sensor (red) is capturing a desktop.



(b) Multiple different frames with varying exposure times contributed texture to the different segments on the couch.



(c) Color coded surface normals are only shown where the high quality version of the surface is loaded.

Fig. 1: The Level of Detail (LOD) system is apparent when looking at a large scene (a). Most of what is displayed consists out of a few triangles with colored corners. More details only appear when zooming in or following the capturing sensor (a). This becomes apparent when looking at the normals (c) which are only shown for fully loaded geometry.

Our approach is inspired by the systems introduced by [11] and [5] but has some essential differences/additions:

- Directly working on a triangle mesh enables us to use textures which are spanned over the used triangles. This strategy inherently propagates the neighborhood information given by the depth map into the reconstruction. This is contrary to the disk shaped surfels used by ElasticFusion [11] which are unconnected and have to overlap to appear like uniform surfaces.
- Storing the texture separate from the geometry allows the meshed surface to be spanned with color information of arbitrary resolution. The density of color information is no longer bound to the geometrical resolution as in ElasticFusion and KinectFusion.
- A Level of Detail (LOD) system which offloads the data residing on the GPU memory to the more plentiful system memory is required when capturing bigger scenes. For user interaction purposes this offloaded data is conserved on the GPU in a lower quality version.

This is not taken care of by the mentioned systems, but absolutely necessary for bigger scenes. Results of this are shown in Fig. 1.

- The segmentation of a captured frame into smaller surfaces is the logical result of the LOD system and the chosen texturing approach. The goal is to split the scene into small manageable chunks which need to be of sufficient size to make texture allocation rational.

II. Scalable Fusion

While the mentioned approaches [5] and [11] work on an intermediate data format, which has to be transformed into a triangle mesh for rendering, our system directly creates and maintains a triangle mesh.

Even more severe, the preceding algorithms attribute only one color to each point of the reconstruction, which then gets interpolated across the triangle surfaces. We, on the other hand, are spanning textures over the surface creating a more detailed reconstruction without increasing the number of triangles.

The consecutive steps performed to incorporate a new camera frame into the reconstruction are presented in the following subsections. These sections are listed in the general order in which they are applied to a frame. To improve performance, this order is later broken up where possible by a threading system described in III.

A. Camera Tracking

Tracking is directly taken from ElasticFusion [11]. But instead of also using the photometric odometry our adaption is limited to the projective ICP approach publicly released by Whelan et al. [11]. Tracking is mostly done relative to the current state of reconstruction. During initialization of the map, an intermediate representation is used based on one keyframe.

B. Geometry Refinement Update

The noise impairing the depth values delivered by RGBD sensors is neither independent nor Gaussian. As a simple example, we imagine a static sensor facing an object at a distance of 4 meters. At distances of about 4 meters, the quantization noise is in the range of centimeters. Usually, the value would appear at one of the closest quantized values, even when observing these values over multiple frames.Following the assumption that this noise behaves Gaussian we would only have to calculate the mean of enough samples to end up with a low standard deviation. From our experiments, we know that we cannot eliminate quantization errors this way, as quantization effects would still be visible this procedure.

In ElasticFusion [11] this is implicitly handled by introducing a "weight" property for surfels. This weight increases the longer a surfel gets observed. During these observations, position, color and normal vectors get updated with the sensor values. With increasing weight of a surfel, these updates become weakened further and further. In the end, the surfels become static, and if the camera does not move, the quantization effects become prominent even with this method. What eventually mitigates this effect is the mechanism which increases the spatial resolution of the reconstruction.

If the sensor approaches a surface in ElasticFusion, surfels become split up into multiple smaller surfels appropriate for the newly gathered data. When doing this, the weight of the new surfels gets reset and a new process of refinement begins cleared of the formerly quantization polluted geometry.

Our approach is inspired by these weights. Instead of spawning new geometry every time the sensor approaches a surface, we only do this when it is beneficial for the reconstructions quality.

For each surface patch, our algorithm stores an additional texture containing values for every sampled surface point p. The values contained for each of these texture pixel (texel) are:

μ_k The average deviation of the k measurements from the actual surface. This is used to indicate where the meshed surface deviates from the sensor's perception.

σ_k An estimate of the noise level. It decreases with every additional measurement. The smaller it is, the less influence new measurements have on the geometry. We also define $\sigma_{s,k}$ as the estimated noise level of the sensor projected onto the surface point p.

$\sigma_{m,k}$ A value which stores estimated minimal noise level that was achievable with the current measurements until step k. The estimate is assuming the quantization effects as the only limiting factor. Similar to before the subscript s refers to the projected value $\sigma_{m,s,k}$ of the sensor.

Each pixel of the texture is updated with the following set of equations: The estimated minimal noise level σ_m is updated by

$$\sigma_{m,k+1} = \min(\sigma_{m,k}, \sigma_{m,s,k}). \tag{1}$$

Updating σ itself is done by

$$\sigma'_{k+1} = \frac{\sigma'_k \sigma'_{s,k}}{\sigma'_k + \sigma'_{s,k}} \tag{2}$$

with

$$\sigma'_{s,k} = \sigma_{s,k} - \sigma_{m,k+1}, \tag{3}$$

$$\sigma'_k = \sigma_k - \sigma_{m,k+1} \tag{4}$$

and therefore

$$\sigma_{k+1} = \sigma'_{k+1} + \sigma_{m,k+1}. \tag{5}$$

It shall be noted that σ'_{k+1} will always be smaller than σ'_k and $\sigma'_{s,k}$ which implies the assumption that every further measurement improves the result. This system also guarantees that σ_k is only approaching $\sigma_{m,k}$ with increasing iteration count k but never falls below it. The resulting values $\sigma'_{s,k}$ and σ'_{k+1} are used to update μ by

$$\mu_{k+1} = \left(\frac{\mu_k}{\sigma'_k} + \frac{d_{s,k} - d_k}{\sigma'_{s,k}} \right) \sigma'_{k+1} \tag{6}$$

with $d_{s,k}$ being the distance of this surface point perceived by the sensor and d_k being the distance of the reconstructed point/texel to the sensor.

Transcribing these texture bound updates to the vertices is done by shifting the vertex positions along the view rays such that μ_{k+1} ends up being 0 wherever possible.

It shall be noted that these updates do not necessarily have to occur every time new sensor values are available for a certain surface pixel. When the estimated noise level of the sensor data σ_k is higher than on the surface $\sigma_{s,k}$, no update needs to be made. The same applies when the perceived depth values are too far off of what has been mapped. This would indicate either unmapped geometry of an already reconstructed surface, or the surface being invalid.

C. Expand Update

As soon as the sensor generates a new frame from a new position, the formerly mapped surface elements are used to render a depth map in the current sensor position. This artificial depth map is then compared to the depth values currently perceived by the sensor. If depth values of the sensor are in proximity to what is mapped, the already existing surfaces will receive an update as described in the previous section. If the captured surface leaves this proximity towards the camera, it will be added (meshed) to the current reconstruction. The thresholds used for these operations as well as $\sigma_{s,k}$ are dependent on depth, pixel position and also on the sensor itself. The sensor characteristics used for the Asus Xtion Pro are derived by Halmetschlaeger-Funek et al. [4] and approximated with a polynomial.

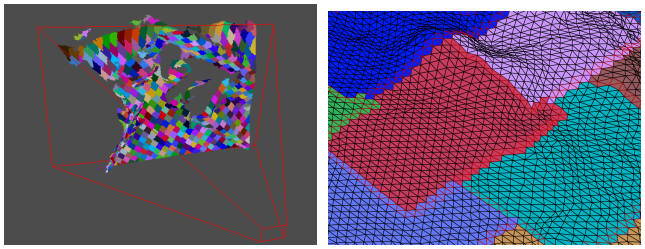
D. Meshing

After identifying the novel parts of the captured depth map, 3D points are created by applying the pinhole model to project the depth pixel. These points are then segmented into smaller blocks depending on their distance to each other and estimated normal vector. The neighborhood information derived from the organized point cloud is directly used in this and also for spanning triangles between each neighboring set of 3 points. When doing so, it again is taken care that no triangles get created where neighboring depth values are within thresholds mentioned in II-C. The results of this segmentation and the meshing process can be seen in Fig. 2.

E. Stitching

Generating a mesh on a single organized depth map is computationally undemanding due to the neighborhood information always being present on the 2D image plane. The situation changes as soon as we seek to integrate new sensor data into an existing reconstruction.

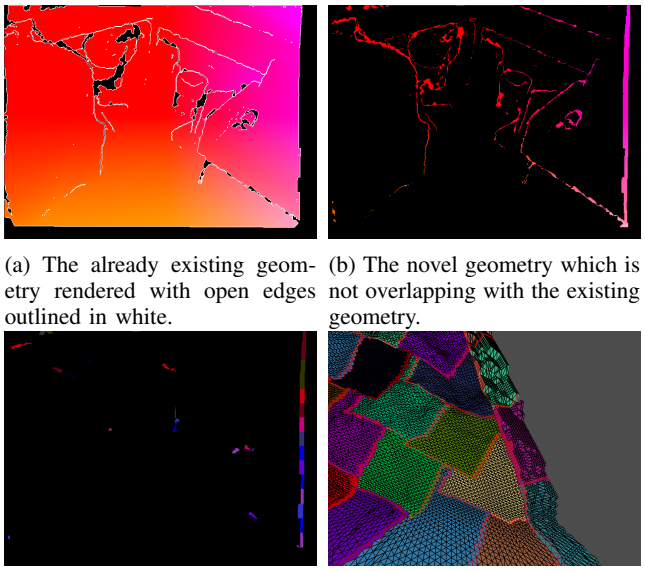
To tackle this problem, we search all the visible triangles captured prior to the current frame for open edges. This refers to every edge where a triangle does not border to another. These edges then get projected in the pixel space of the current frames depth map. When doing so, finding a potential neighbor for a reconstructed triangle within the set of novel triangles is a simple lookup in the current image plane. The whole process of expand the reconstruction is shown in Fig. 3.



(a) The coarse segmentation of the pointcloud into smaller patches

(b) These patches get meshed into a regular pattern of triangles.

Fig. 2: The triangle creation process applied to a single frame.



(a) The already existing geometry rendered with open edges outlined in white.

(b) The novel geometry which is not overlapping with the existing geometry.

(c) Coarse segmentation of the novel geometry. Note how smaller regions do not get mapped.

(d) Finished stitch. The novel geometry appears rough since it was not improved by additional observations.

Fig. 3: The steps required to connect novel data of a sensor frame to existing geometry. The open edges of the geometry (a) outlined in white are connected to the coarse segmentation (c). The result (d) shows a blatant line in the segmentation pattern.

III. Implementation

Modern desktop hardware still distinguishes between memory bound to the GPU and system memory which is bound to the CPU. Access can not be done across memory spaces without doing expensive data transfer over the PCI-E bus. Therefore, our data structures are designed to mirror the information between CPU and GPU and only synchronized when absolutely necessary.

Modifications on the geometry occur on either the GPU or the CPU depending on which processor is more fit for

the performed task. This has implications on the data structure. While data stored CPU space is vastly interconnected, the structures on the GPU only store very few references between elements.

We furthermore use a threading system to simultaneously process tasks which are not fully interdependent. While e.g. the geometry of one frame is used to update the mesh, data which is not needed can be transferred from GPU memory to system memory (download). The odometry is likewise not explicitly reliant on the most current version of geometry, tracking of a new frame can therefore occur concurrently with the integration of the last frame. A example of this system is shown in a timeline in Fig. 4.

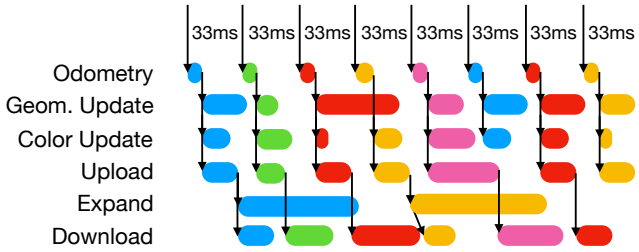


Fig. 4: Each of the rows depicts one thread specialized for its task. The lines show the data flow, beginning with the capture of images at 30 Hz. It is shown how e.g. the geometry refinement update of one frame prevents the geometry refinement update of the following. For the download task, this strategy is not an option. To ensure all the updates made to geometry will be secured, a download step can only be postponed but never dropped.

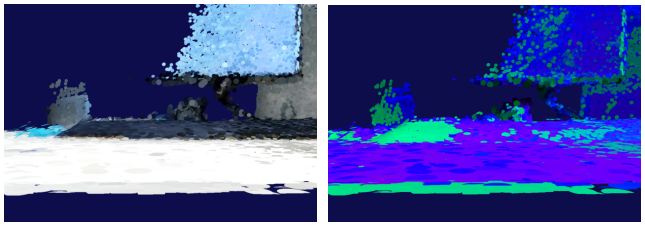
IV. Evaluation

Whelan et al. evaluated the performance of ElasticFusion by comparing the trajectory of the camera odometry to the ground truth captured by Sturm et al. [9]. Further comparisons included the distance of the resulting surfels to the ground truth geometry used to artificially render a dataset. Since our odometry only resembles a part of what is being used by ElasticFusion, we renounce to run these tests at this early state of the pipeline. It should be noted though, that we do not see any technical limitation that opposes the integration of the remaining mechanisms to match ElasticFusions performance.

A. Qualitative Comparison

When looking at surfaces reconstructed by ElasticFusion it is noticeable (Fig. 5) that some of the surfaces are mapped multiple times. This is due to discrepancies in camera tracking and sensor values which we are partially overcoming with a thresholding system that takes standard deviations of the sensor into account.

We are also utilizing a stitching mechanic for connecting geometry that has been created in consecutive frames. Due to imperfection in our stitching algorithm some of these stitches are not complete as shown in Fig. 7. A situation which



(a) Colorized ElasticFusion reconstruction of a desktop. This view is cutting trough the (double) surface and facing a monitor.

(b) ElasticFusion creates multiple layers of the same surface made distinguishable by the green and blue colors (colorized by number of observations).

Fig. 5: ElasticFusion has the tendency of doubling surfaces by creating a secondary layer of surfels.

is worsened by oversegmentation and sequentially clustered surfaces.

In case the sensor is approaching an already mapped surface we replace the old textures of surfaces with the newer higher resolution versions. In its current form this happens without taking care of exposure time and other effects, thus segment borders become visible by abrupt changes in intensity. Fig. 8 shows the increased color density as well as the mentioned discontinuities and offers a comparison to ElasticFusion.

B. Memory Consumption

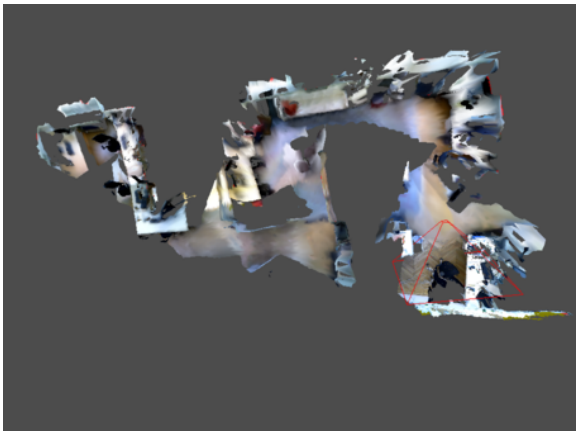
Our current implementation shows its advantage when the sensor keeps exploring new surfaces. In these situations, ElasticFusion will eventually run out of GPU memory while ScalableFusion offloads finished chunks to system memory. This is shown in Figure 9, where the memory consumption of ElasticFusion is steadily increasing while our implementation adjusts its use of GPU memory on the demand. Consecutively this also means that the memory consumption increases when over-viewing big but also detailed structures. In our tests this never posed a problem though.

C. Computational Performance

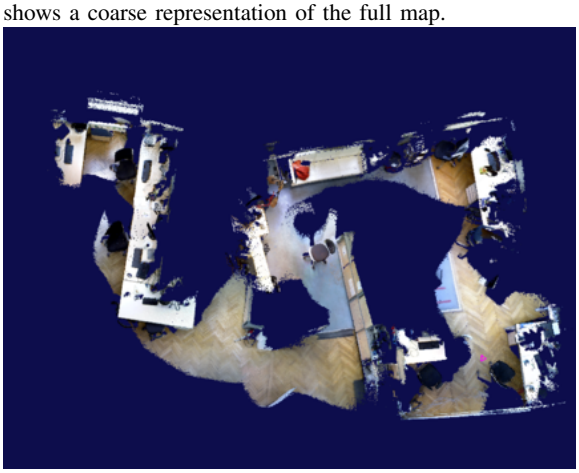
As depicted in Fig. 4 almost all of the tasks are run at the designed 30 Hz. The only task massively deviating from this design goal is the Expand (Section II-C) task. Instead of the targeted 30 ms, it takes 150 to 800 ms to complete. As long as novel geometry is not introduced at a high rate, these durations will not pose a serious limitation.

For our experiments we used a desktop Intel Core i7-7700K CPU in combination with a Nvidia Geforce GTX 1070 with 8 GB VRAM. This combination easily ran the tracking and update steps at the full frame rate (30 Hz) while expanding the geometry at approximately 5 Hz. Most of the CPU cores are utilized to some extent, but mainly waiting for GPU tasks to finish. The GPU was taxed to about 70% of its capacity, which implies some headroom for future features.

Running the same software on a notebook resulted in skipped frames for tracking ($\sim$ 9 Hz), update steps ($\sim$ 8 Hz) and hiccups in the user interface. The expand step ran at an



(a) When zoomed out like this, our scalable system only shows a coarse representation of the full map.



(b) ElasticFusion on the other hand always renders all the surfels.

Fig. 6: The reconstruction of scenes like a whole office triggers the LOD system. When the user interface view is zoomed out, our system (a) only renders a low detail version of the reconstruction while ElasticFusion (b) still renders every mapped surfel.

even lower frequency of ~ 1.4 Hz. The resulting reconstruction nevertheless yields similar quality of what the desktop fabricated. The notebook features an Intel Core i7-3740QM CPU with a Nvidia Quadro K2000M and 2 GB VRAM.

V. Conclusions

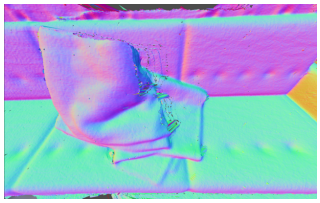
It is shown that directly working on triangles and vertices is feasible in terms of computational effort and even beneficial when maintaining bigger reconstructions.

Conducting all of the meshing in pixel space presents itself as efficient approach for a potentially CPU-intensive problem.

When comparing the resulting normals rendered by ElasticFusion and our approach, it becomes evident that we achieve a similar level of detail (Fig. 7).

Textures appear superior in many instances due to being captured in higher resolution. This representation is less

(a) Textured model generated by our system.

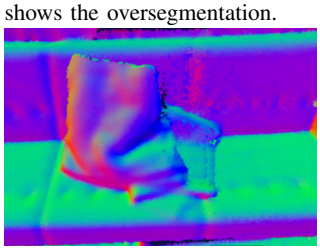


(b) Color coded surface normals.

(c) Stitching artifacts appear between segments.

(d) Color coding the segments shows the oversegmentation.

(e) Same scene captured by ElasticFusion.



(f) Surfels color coded by their normal vector (ElasticFusion).

Fig. 7: The couch scene captured by our system (a-d) and by ElasticFusion (e, f). While the results of our system are comparable on the geometry side, a closer look (c, d) to where the (red) blanket initially shadows the couch from the sensor reveals stitching issues. Geometry needs to be connected between frames which is negatively influenced by noise of geometry data. We hope to fix this issue with a post processing step.

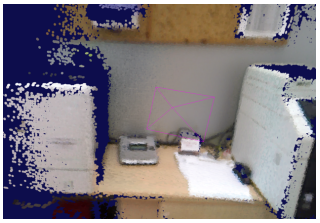
forgiving for rolling shutter sensors, changes in exposure times and tracking errors. As a result, borders between textured patches manifest themselves as sudden changes in intensity as seen in Fig. 8.

VI. Outlook

This paper describes a reconstruction pipeline in an immature state and therefore leaves some problems untreated.

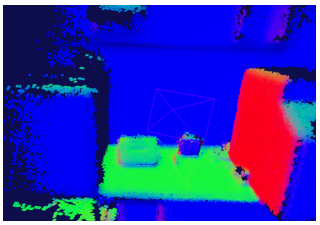
As already indicated in Section II-A the odometry is limited to the use of ICP instead of also exploiting photometric alignment as in [11]. Besides adding these missing parts we are also considering the usage of feature based approaches like ORB SLAM [6] or different featureless ones as Direct Sparse Odometry (DSO) [3].

Texture gets captured in varying lighting conditions, exposure and angles. This leads to reconstructions with very fragmented, non-uniformal texturing. A first step to counter this would be a system to estimate and spare out specular highlights as introduced for ElasticFusion [12]. To further improve quality, the integration of vignetting compensation

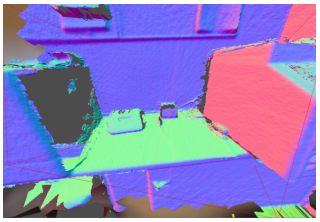


(a) Surface model captured by the surfel based ElasticFusion.

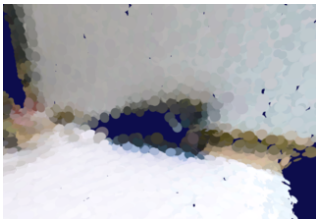
(b) The same input data fed to our system.



(c) Surfels colorized by their normals.

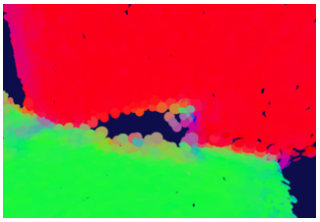


(d) Surfaces colorized by their normals.

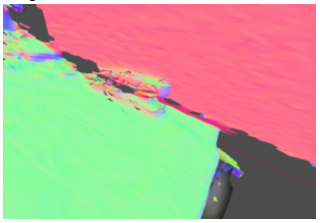


(e) Details like the stapler are barely captured by ElasticFusion.

(f) While our approach has a higher resolution texture for the stapler.



(g) The stapler geometry is barely mapped by ElasticFusion.



(h) Our approach introduces artifacts for the fine geometry of the stapler.

Fig. 8: In a scenario where the sensor is slowly approaching surfaces ElasticFusion (left), as well as our approach (right), improve the geometrical surface quality with a similar principle yielding similar results. When zooming in (e-h) the improvements due to our texturing approach become apparent.

[1] as well as high dynamic range and exposure control presented by Alexandrov et al. [2] is planned.

Other unmentioned tasks are the removal, simplification and tessellation of geometry which are planned for implementation.

It remains to be seen, how much impact these additional features and further optimization will have on the system performance. We are confident though, that further development of this software will improve its utility while keeping the moderate hardware requirements.

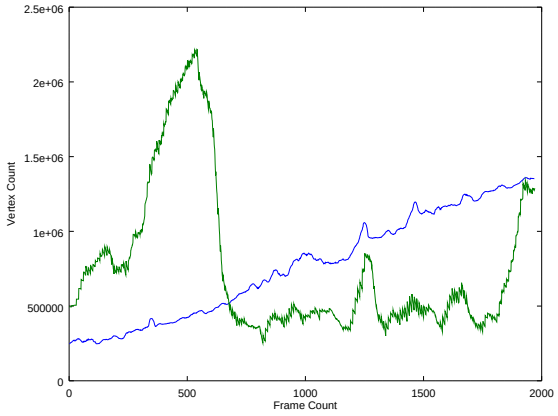


Fig. 9: While the vertex count of ElasticFusion (blue) keeps increasing steadily, the count of ScalableFusion (green) only depends on what is visible momentarily. This also implies that when the sensor overviews a large area full of highly detailed surfaces, the memory consumption spikes (Frame 500).

References

[1] S. V. Alexandrov, J. Prankl, M. Zillich, and M. Vincze, "Calibration and correction of vignetting effects with an application to 3d mapping," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4217–4223.

[2] ——, "Towards dense slam with high dynamic range colors," in *2017 Compute Vision Winter Workshop (CVWW)*, Feb 2017.

[3] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *CoRR*, vol. abs/1607.02565, 2016.

[4] G. Halmetschlager-Funek, M. Suchi, M. Kampel, and M. Vincze, "Xtion's gone! What's next? An evaluation of ten different depth sensors for robotic systems," *Under Review for IEEE Robotics Automation Magazine*, 2018.

[5] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11, 2011, pp. 559–568.

[6] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017. [Online]. Available: https://doi.org/10.1109/TRO.2017.2705103

[7] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 169:1–169:11, Nov. 2013.

[9] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[10] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul 2012.

[11] T. Whelan, S. Leutenegger, R. S. Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[12] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense SLAM and light source estimation," *I. J. Robotics Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.

Page Segmentation and Region Classification Based on Region Bounding Boxes

Thomas Lang[1], Markus Diem[1], Florian Kleber[1] and Robert Sablatnig[1]

***Abstract*— We present an approach for the segmentation and classification of digital document images of newspapers or similar types of documents which are described by a set of partially correct text and image bounding boxes. These flawed region descriptions are used to locate the bounding boxes of the true page component boundaries in the documents. The resulting regions are classified as text, images, charts or tables. In addition to the individual evaluation of the segmentation and classification steps, the combined physical layout analysis system is evaluated and compared to the page segmentation results of an open-source document analysis software.**

I. INTRODUCTION

Page segmentation is a crucial step in document analysis, as it is a requirement for other tasks like OCR to have isolated document regions containing only one type of content and for determining the structure of a document. Layout analysis is an active research field and methods continue to improve, as is evident from the biannual competitions organized by PRImA [2]. The general aim is to locate page components and to classify them according to their content type and meaning. We present a method for the more constrained task of page segmentation and component classification for digital document images with existing partially correct annotations of text and image regions. Such region descriptions may be the result of layout analysis of PDF documents, which was studied by Chao and Fan [1]. Even though the text and image components can be accurately extracted from PDF documents, the embedded raster images are often broken up into smaller parts when the file is created by publishing software. As a result, the appearance of the images in the document is unchanged, but when the attempt is made to extract image boundaries from the PDF file, instead of the actual image borders, only smaller image segments are obtained. In addition, if clipping masks are not properly extracted along with the images, only rectangular bounding boxes are obtained, even if the image portions displayed in the document have more complex shapes.

In this paper, we use a dataset containing newspaper pages in the form of raster images and corresponding sets of rectangular image and text region descriptions, which are known to have been extracted from PDF files by the provider of the dataset. Unfortunately, this dataset is not publicly available. The image region boundaries show all the described problems resulting from PDF extraction. An example of such problematic image region descriptions can be seen in Fig. 1. Additionally, sets of manually annotated ground-truth chart and table regions are available to us (see section III-A). Since these annotations also contain only rectangular region boundaries (bounding boxes), the proposed segmentation method produces rectangular regions as well, even if the underlying document region has a more complex shape. However, the method could easily be adapted to detect the exact boundaries of document components, as will be explained in section II-A. The page segmentation and classification method has previously been published, along with the evaluation results of the classification step [5]. For the sake of completeness, we first provide a short summary of both steps and of the classification results. We then present measures for the evaluation of the segmentation step. Results are shown for the segmentation step, for the full system and for a comparison to a different page segmentation system.

Fig. 1: The image region boundaries included in the dataset describe only segments of the actual image, overlap each other and falsely include surrounding text segments. The goal is to find the bounding box containing only the image.

II. METHODOLOGY

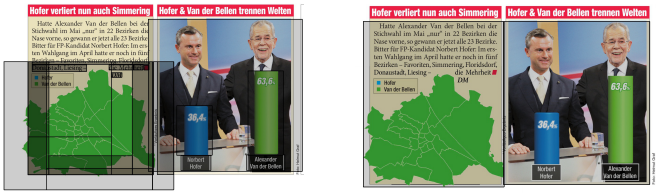
We first segment the partially incorrect image regions to obtain region rectangles that fit visible document components. These regions are then classified as being of either the chart or image class, and the text regions are classified as text or tables. It is assumed that all document images are not skewed or warped in any way.

A. Image Region Segmentation

The segmentation of image regions begins by removing the text from the gray-scale document image using a simple rule-based approach. For each of the text region rectangles, the

*This work was supported by APA-IT

[1]Thomas Lang, Markus Diem, Florian Kleber and Robert Sablatnig are with Computer Vision Lab, Institute of Visual Computing & Human-Centered Technology, Faculty of Informatics, TU Wien, Vienna, Austria {tlang,diem,kleber,sab}@cvl.tuwien.ac.at



(a) Before segmentation (b) After segmentation

Fig. 2: Example of rectangular image regions before and after segmentation.

respective document image segment binarized and the more frequent value is treated as the background. Pixels deviating from this background value by more than a specified threshold are considered foreground pixels. Connected component analysis is performed on the binary foreground mask. Two criteria are used to eliminate non-text pixels from this mask. First, connected components (CCs) are removed if their pixel area deviates from the area of all other components by more than a certain relative threshold. Second, CCs are removed if the color of most of their pixels differs from the dominant text color, which is found using k-means clustering on the a* and b* channels of the image in L*a*b* color space. As a last step, to account for smooth text borders, the image region mask is dilated. Afterwards, the mask is used to replace all text pixels in the gray-scale image with the background value.

What remains after the text removal procedure is a document image containing vector graphics, separator lines and other layout elements, which can be used in combination with the available image region rectangles to locate the boundaries of the image components. First, adjacent and overlapping (clustered) image region rectangles like the ones shown in Fig. 1 are grouped together. As in the text removal step, a global threshold is computed to find the background value of the page. Afterwards, a foreground mask is created containing all pixels which are darker than the background value (for pages with a bright background). For each group of image region rectangles, a "cluster mask" is created, containing all its pixels. The logical AND operation between a cluster mask and the document foreground mask defines a cluster foreground mask. As explained previously, we only use the bounding boxes of these components. An example of the result of the segmentation can be seen in Fig. 2.

If the goal was to obtain exact regions, their boundaries are defined by the cluster foreground masks. The text regions would have to be reduced to the borders of the text masks used to remove the text from the document image.

B. Region Classification

The available text and image regions (not segmented) are used along with manually annotated ground-truth rectangles to generate feature descriptors for each region class (text, image, chart, table). First, each region is sampled down to the highest image pyramid level (smallest size) which is still larger or equal to 64×64. Afterwards, one or more HOG features [4] are computed by sliding a 64×64 window over the region rectangle with a step size of 32 in both directions. Therefore, in each region dimension with the size dim, $\frac{dim-64}{32}$ shifts are performed. Each feature is associated with the class of the region it was computed from. Finally, the collected features from the training set are used to train a random forest classifier.

For the classification, the features are computed in the same way on the test set regions, after which they are classified by the random forest. However, since we are interested in the classes of the regions and not the individual feature samples, we add the votes of all bagged trees of the random forest for each window position inside the same region in order to find a decision for the whole region.

III. EVALUATION

The segmentation and the classification steps are first evaluated separately. We then present results for the complete system. Additionally, these results are compared to a different page segmentation method, which is part of the Tesseract OCR engine[1] developed by Google [8]. All evaluation steps are performed on a dataset of raster images of contemporary newspaper pages containing text region bounding boxes along with partially correct image region rectangles.

A. Classification

For the evaluation of the classification, the chart and table regions in 6211 newspaper pages have been manually annotated. As a result, 891 pages contain at least one chart or table. We use 70 % of this dataset for training and the remaining pages as test set (624 training pages, 267 test pages). The training set is balanced by reducing the number of feature descriptors of each class to the minimum class size. The test sets are also balanced, by classifying the same number of regions for each class. The random forest predictor included in OpenCV 3.3[2] is used with a tree depth limit of 25 and a maximum number of trees of 150. Text regions are classified as text or tables and image regions are classified as images or charts. Therefore, we train and evaluate two random forest classifiers: one for the distinction between text and tables and one for the distinction between images and charts. For the computation of text and image feature descriptors, the available (partly incorrect) region rectangles are used. We rely on the assumption that the image rectangles contain image regions for the most part, even if some of them overlap and don't always fit the actual image parts in the document. The confusion matrices in Tables I and II show the classification results. The rows represent the actual classes; the columns are the predictions. For the classification of text regions as text or tables, 99 regions are wrongly classified, which is equal to an overall error rate of 0.05 (2168 regions in total). For the image/chart classification, the overall error rate is 0.1, with 73 wrongly classified regions out of 702 in total.

[1] https://github.com/tesseract-ocr/tesseract/
[2] https://opencv.org/

	Text	Table	Re.
Text	1048	36	0.97
Table	63	1021	0.94
Pr.	0.94	0.97	

TABLE I: Text/Table confusion matrix

	Image	Chart	Re.
Image	328	23	0.93
Chart	50	301	0.96
Pr.	0.87	0.93	

TABLE II: Image/Chart confusion matrix

B. Segmentation

For the segmentation evaluation, all image regions contained in 70 of the 267 pages of the test set have been manually annotated. Since the image rectangles resulting from the segmentation are classified either as images or charts, their correctness is measured by comparing the segmented rectangles to all ground-truth image and chart regions. The ground truth set contains a total of 389 image and chart region rectangles, compared to 461 which are produced by the segmentation. Two kinds of evaluation are performed.

First, the amount of area overlap between computed and ground-truth region rectangles is computed. The ratio of total intersection area to total area of computed or ground-truth regions is the precision or recall respectively. The overall precision of the segmented regions is 0.94 and the recall is 0.77. If the same measurements are taken without the chart regions in the ground-truth set, the precision decreases to 0.9, while the recall increases to 0.88. This already shows that large parts of the ground-truth charts are not or only partly matched by segmented image regions.

The second evaluation aims to find not only the region overlap, but the number of region rectangles which are segmented correctly. For each page, there are two sets of regions S_C and S_{GT} containing the segmentation results and the ground truth. We define two regions r_1 and r_2 as *fitting* if their Jaccard index

$$J(r_1, r_2) = \frac{\text{area}(r_1 \cap r_2)}{\text{area}(r_1 \cup r_2)}, \tag{1}$$

is greater than $1-T$, where T is a tolerance value. Both region sets S_C and S_{GT} contain subsets of regions for which a fitting region exists in the other set:

$$F_C = \{r_C \in S_C \mid \exists r_{GT} \in S_{GT} : \text{fits}(r_C, r_{GT})\}, \tag{2}$$

$$F_{GT} = \{r_{GT} \in S_{GT} \mid \exists r_C \in S_C : \text{fits}(r_C, r_{GT})\}. \tag{3}$$

Furthermore, a region r of one set $S_C \setminus F_C$ or $S_{GT} \setminus F_{GT}$ is *covered* by regions $r_1^*, r_2^*, ..., r_n^*$ of the other set S_{GT} or S_C if it fits their union area, meaning that $J(r, r_1^* \cup r_2^* \cup ... \cup r_n^*) > 1-T$. This concept of covered regions is similar to merges and splits in the evaluation measures described by Clausner et al. [3]. The sets of covered computed and of covered ground truth regions are called C_C and C_{GT}. A region is *matched* if it fits one or more regions in the other set: $M_C = F_C \cup C_C$, $M_{GT} = F_{GT} \cup C_{GT}$. The ratio of matched regions in the computed or ground-truth set can be interpreted as the region matching precision Pr_M and recall Re_M respectively. Table III shows the numbers of regions in each set summed over all documents in the dataset. The F_1 score is defined as the harmonic mean of the recall and precision values.

T	$\sum\lvert F_{GT}\rvert$	$\sum\lvert F_C\rvert$	$\sum\lvert C_{GT}\rvert$	$\sum\lvert C_C\rvert$	$\textbf{Re}_M$	$\textbf{Pr}_M$	F_1
0.05	156	156	1	0	0.4	0.34	0.37
0.1	169	169	1	0	0.44	0.37	0.4
0.15	172	172	1	0	0.44	0.37	0.41
0.2	181	181	1	0	0.47	0.39	0.43
0.25	187	186	1	2	0.48	0.41	0.44
0.3	188	187	3	2	0.49	0.41	0.45

TABLE III: Segmentation region matches (images and charts)

T	$\sum\lvert F_{GT}\rvert$	$\sum\lvert F_C\rvert$	$\sum\lvert C_{GT}\rvert$	$\sum\lvert C_C\rvert$	$\textbf{Re}_M$	$\textbf{Pr}_M$	F_1
0.05	154	154	1	0	0.58	0.33	0.42
0.1	166	166	1	0	0.62	0.36	0.46
0.15	169	169	1	0	0.63	0.37	0.46
0.2	177	177	1	0	0.66	0.38	0.49
0.25	181	181	1	1	0.68	0.39	0.5
0.3	182	182	2	1	0.68	0.4	0.5

TABLE IV: Region matches of image regions only

It can be seen that depending on the tolerance value T, the recall Re_M varies between 0.4 and 0.49, and the Pr_M lies in the range 0.34 to 0.41. Compared to the area-based evaluation results with a recall of 0.77 and a precision of 0.94, the values are significantly lower. This shows that some of the region bounding boxes in each of the sets (computed or ground truth) only partly intersect the region rectangles in the other set, but do not match them exactly. The amount of regions covered by a set of other regions (C_{GT} and C_C) is rather low (1.6 % and 1.1 % for $T = 0.3$), but it does occur.

We again take a second measurement without the chart regions in the ground truth set, which reduces its size from 389 to 269. The results are shown in Table IV. With the ground truth set containing only image regions, the recall increases significantly (by 0.19 on average). However, the number of matched regions is almost unchanged. The increased recall value is mainly explained by the smaller size of the ground truth set (30.8 % decrease). This means that most ground-truth chart bounding boxes are not matched by segmented image regions. The almost unchanged precision values Pr_M show that many of the segmented image regions lie outside the ground-truth image regions. This can be explained by them being parts of chart regions, which often contain, but do not fully consist of images.

C. Complete System

For the evaluation of the complete layout analysis system, both methods from the segmentation evaluation (measuring intersecting areas and counting exact matches) are reused, but for each class individually. The results represent the performance of the full segmentation and classification procedure. The results of the area-based evaluation in Table V again show that the method generally fails to produce correct chart regions. For the text regions, we reuse the available region bounding boxes. Therefore, the errors result only from text regions being falsely classified as tables. Table VI

	Text	Image	Chart	Table
Re	0.98	0.8	0.15	0.81
Pr	0.99	0.92	0.37	0.79
F_1	0.98	0.86	0.21	0.8

TABLE V: Results of the area-based evaluation of the complete system.

T		0.05	0.1	0.15	0.2	0.25	0.3
Text	Re_M	0.98	0.98	0.98	0.98	0.98	0.98
	Pr_M	0.99	0.99	0.99	0.99	0.99	0.99
	F_1	0.99	0.99	0.99	0.99	0.99	0.99
Image	Re_M	0.53	0.57	0.58	0.61	0.62	0.63
	Pr_M	0.33	0.36	0.36	0.38	0.4	0.4
	F_1	0.41	0.44	0.45	0.47	0.48	0.49
Chart	Re_M	0.01	0.02	0.02	0.03	0.03	0.03
	Pr_M	0.03	0.05	0.05	0.08	0.1	0.1
	F_1	0.01	0.03	0.03	0.04	0.05	0.05
Table	Re_M	0.63	0.63	0.64	0.65	0.67	0.67
	Pr_M	0.6	0.6	0.6	0.6	0.61	0.61
	F_1	0.61	0.61	0.62	0.62	0.64	0.64

TABLE VI: Results of the match-based evaluation of the complete system.

again shows that almost no chart regions are matched. Since the text regions are not segmented, but only classified, the results are about as high as in the area-based evaluation. The image results have decreased because of segmented region rectangles not matching the ground-truth ones. The F_1 scores of table regions lie between 0.61 and 0.64.

D. Comparison to Tesseract

We use a converter developed by PRImA Research[3] to store the page segmentation output of the Tesseract engine (version 3.04) in the PAGE format [7]. Since Tesseract produces non-rectangular region polygons, in order to compare them to our method, it is necessary to use just their bounding boxes. An example of region polygons produced by Tesseract can be seen in Fig. 3. Directly adjoining region polygons are merged to a single region. Table VII shows the area-based evaluation results of the Tesseract regions compared to our method. The "proposed method" columns show the same numbers as in section III-C. Since Tesseract does not detect chart regions, this class is left out. For the "Image*" row, all chart regions are treated as image regions. Tesseract produces a large amount of table regions, which causes its recall value to be very high (0.81), but the precision to be low (0.4), resulting in an F_1 score of only 0.54 compared to 0.8 with our method.

[3] http://www.primaresearch.org/tools/TesseractOCRToPAGE

Fig. 3: Example of image-class polygons produced by Tesseract.

Type	Precision		Recall		F_1	
	Tess.	Prop.	Tess.	Prop.	**Tess.**	**Prop.**
Text	0.92	**0.99**	0.86	**0.98**	0.89	**0.98**
Image	0.8	**0.92**	0.78	**0.8**	0.79	**0.86**
Image*	0.87	**0.94**	0.73	**0.77**	0.79	**0.85**
Table	0.4	**0.79**	0.81	0.81	0.54	**0.8**

TABLE VII: Comparison between Tesseract and the proposed method.

IV. CONCLUSION

We have seen that the proposed page segmentation and classification method based on region bounding boxes provides satisfactory results in some respects. The classification based on HOG and random forests achieved low error rates and the overall F_1 results are higher than those of Tesseract. For a more general use case, it would be interesting to extend the method to generate more exact (non-rectangular) region descriptions, which would in turn also require more exact (e.g. polygonal) ground-truth information for evaluation.

The evaluation results show that even with simple methods, data extracted from PDF files can be used to obtain a viable page layout description. However, it has also become clear that the method fails at detecting chart regions, because they consist of multiple image and text elements. Such complex objects would have to be addressed in a different way, like considering combinations of page elements as possible regions.

REFERENCES

[1] H. Chao and J. Fan, "Layout and content extraction for pdf documents," in *International Workshop on Document Analysis Systems*. Springer, 2004, pp. 213–224.

[2] C. Clausner, A. Antonacopoulos, and S. Pletschacher, "ICDAR2017 competition on recognition of documents with complex layouts - RDCL2017," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, nov 2017.

[3] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Scenario driven in-depth performance evaluation of document layout analysis methods," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 1404–1408.

[4] 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.

[5] T. Lang, M. Diem, and F. Kleber, "Physical layout analysis of partly annotated newspaper images," in *Proceedings of the 23rd Computer Vision Winter Workshop*, 2018, pp. 63–70.

[6] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[7] S. Pletschacher and A. Antonacopoulos, "The page (page analysis and ground-truth elements) format framework," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 257–260.

[8] R. Smith, "An overview of the tesseract ocr engine," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE, 2007, pp. 629–633.

The Convex-Concave Ambiguity in Perspective Shape from Shading

Michael Breuß[1], Ashkan Mansouri Yarahmadi[1] and Douglas Cunningham[2]

***Abstract*— Shape from Shading (SFS) is a classic problem in computer vision. In recent years many perspective SFS models have been studied that yield useful SFS approaches when a photographed object is close to the camera. However, while the ambiguities inherent to the classical, orthographic SFS models are well-understood, there has been no discussion of possible ambiguities in perspective SFS models. In this paper we deal with the latter issue. Therefore we adopt a typical perspective SFS setting. We show how to transform the corresponding image irradiance equation into the format of the classical orthographic setting by employing spherical coordinates. In the latter setting we construct a convex-concave ambiguity for perspective SFS. It is to our knowledge the first time in the literature that this type of ambiguity is constructed and verified for a perspective SFS model.**

I. INTRODUCTION

Shape from Shading (SFS) is a fundamental problem in computer vision [6]. Given a single greyscale input image, the SFS process makes use of the variation of grey values appearing by light reflectance at an objects' surface to reconstruct the 3D shape. In order to avoid ill-posedness in the reconstruction process as much as possible, especially modeling assumptions on illumination and light reflectance in a scene are employed in SFS.

The classic SFS model assumes the camera to perform an orthographic projection, that light falls on the scene of interest in parallel rays from infinity and that photographed objects have a Lambertian surface yielding the light reflectance, cf. [6], [7]. Within the last years the setting of perspective camera projection has received much attention in SFS, which we consider here as the *perspective shape-from-shading (PSFS)* models. For some milestones in the development of PSFS, let us mention here Lee and Kuo [10] who formulate an image irradiance equation with Lambertian surfaces and a nearby light source, including yet simplifications such as image formation over triangular surface patches and a linear approximation of the reflectance map. There are several groups who worked on more general models formulated explicitly by *partial differential equations (PDEs)* [3], [12], [16]. These models also feature Lambertian surface reflectance and parallel lighting from infinity. Furthermore, Prados and Faugeras considered a point light source in finite range of the photographed scene – more specifically, putting it at the center of projection, being roughly equivalent to modeling a camera with flashlight – and introduced a light attenuation factor. The latter enables some degree of well-posedness [1], [13], [14] but also makes the model itself considerably more complicated.

The arguably most studied SFS model in the SFS literature is the classic SFS described by Horn [5], see also [17]. As a particular feature, it involves the *convex-concave ambiguity* [7]. This means, given an input image of an object, the classic SFS model itself cannot distinguish between convex and concave versions of the objects' surface, e.g. a photographed mound (convex) seen from above and with lighting from above could as well be a photographed cavity (concave) of the analogous form. Let us note that the convex-concave ambiguity is of interest not only in computer vision, but also in the context of understanding models of human perception; for example, if the silhouette edge information specifies a convex shape, cf. [9], then human perceivers will have a strong bias towards seeing the surface as convex regardless of other shading factors such as specular highlights [11].

Our Contribution. In this paper, we show for the first time in the literature that also PSFS models may exhibit the convex-concave ambiguity. The model set-up we explore incorporates in addition to the perspective camera projection and Lambertian surface reflection that the light source is at the projection center. This choice coincides with the light source position employed by Prados and Faugeras. Similarly to the proceeding in [4], [8], where different models as here were considered, we propose to reformulate the arising PDE in a *spherical coordinate system*. We show that the reformulated PDE is in spherical coordinates of *exactly the same form* as the classic PDE of orthographic SFS from Horn. By exploring then the known mechanisms of the convex-concave ambiguity for classic SFS, we construct the corresponding ambiguity for PSFS. We show this construction here in detail for input signals, but it is evident that the proceeding can easily be extended in a straightforward way to images.

II. SHAPE FROM SHADING SET-UP

In the next paragraphs we briefly elaborate on the orthographic as well as the perspective camera projections and SFS models.

A. Orthographic Shape from Shading

For the classic orthographic SFS model, in addition to the projection itself the position of light source and the surface normal vectors used in the Lambertian reflectance formula are the main construction elements. Let us review them one by one, recalling thereby the setting described in more detail in [6], [7].

[1]Brandenburg Technical University, Institute of Mathematics, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany {breuss,ashkan.mansouriyarahmadi}@b-tu.de

[2]Brandenburg Technical University, Institute of Computer Science, Konrad-Wachsmann-Allee 5, 03046 Cottbus, Germany douglas.cunningham@b-tu.de

Coordinate system. Our choice is identical to the classic setting of a right handed coordinate system shown in Figure 1. The motivation behind this is to have the outward normals to the surface $z = f(X,Y)$ pointing always to the positive direction of the z-axis.

Lighting. The light source is assumed to be far away from the scene and placed at the positive side of the z axis, uniformly illuminating the surface as parallel beams coming from infinity. In this way, the outward normals of the surface always point to the light source. To model this, we employ the light vector $\omega = (\omega_1, \omega_2, \omega_3)^\top$ as the representative of the existing light in scene and consider its length to be unit. About the direction of ω, it is convenient to direct it from the surface to the light source, since in this case the outward normal to the surface and the light vector both point to the positive side of the z-axis and the incident angle θ among them is of interest.

Surface normal vectors. The normal vector $\mathbf{n} = (-p,-q,1)^\top$ describes the surface geometry. The surface normals always point to the positive side of the z-axis. Here, p and q are the rate of change of surface f in x and y directions, respectively.

Lambertian reflectance. The irradiance of a Lambertian surface depends only on the angle between the normal $\mathbf{n}$ at a surface point with the light vector ω reaching to that point. Thus a Lambertian surface looks identical from any point it is observed. This is formalized by Lambert's cosine law

$$I(x,y) = \rho(x,y)\,(\omega \cdot \mathbf{n}(x,y)) \tag{1}$$

where ρ represents the albedo of the Lambertian surface, which we set for simplicity always equal to one in this paper.

This setting allows to formulate an *eikonal equation* as the constituting equation of classical, orthographic SFS:

$$\|\nabla u(x,y)\| = \sqrt{\frac{1}{I(x,y)^2} - 1} \tag{2}$$

Here, $I(x,y)$ is the image irradiance found at the pixel with coordinates (x,y) in the input image.

Let us note that in the constituting equation (2), the depth of the unknown surface $u(x,y)$ is described in terms of first-order derivatives within the nabla operator. Therefore one may add an arbitrary constant α to the depth, obtaining $u(x,y)+\alpha$, and obtain the same PDE as in (2). Consequently, the PDE (2) describes the *shape* of an object up to a free *additive* parameter, but not the actual depth in a scene.

B. Perspective Shape from Shading

In simplest form the perspective projection could be performed by a *pinhole camera*, see Figure 2. Let us note that we specify the world coordinate system by capital letters, while we write small letters for coordinates in the image coordinate system, given in the image plane. This distinction is of some importance in the perspective setting, and in below paragraphs we give more details.

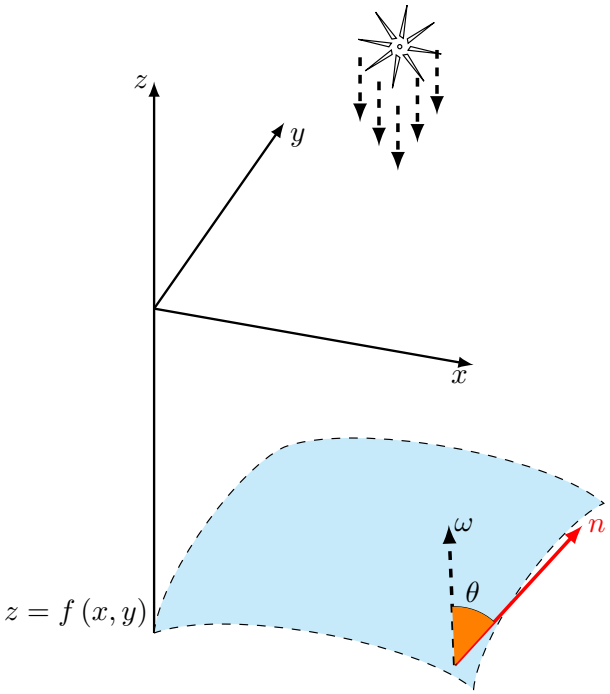


Fig. 1: An orthographic setup with a surface $z = f(x,y)$ to be located at the negative side of the z axis. The source of light ω is assumed to be located at infinity illuminating the surface as parallel beams in direction of z-axis, as seen from the surface. In such a setup, every object point (x,y,z) is projected to the image plane (x,y). Let us note that world coordinates (X,Y) are here identical to image coordinates (x,y), so we do not distinguish these coordinate systems explicitly in the orthographic setting.

World coordinate system. To describe any point in a scene, three axis of X, Y and Z are used. They form a right-handed coordinate system as shown in Figure 2.

Image plane. It is spanned by two axis, featuring x and y as coordinates, as depicted in Figure 2 and its origin c is called principal point. The image plane is located at the distance $Z = f$ from the origin C of the world coordinate system. Here f is the focal length of the pinhole camera.

It is important to note that one may describe the perspective projection via

$$\Delta\colon \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \in \mathbb{R}^3 \mapsto \underbrace{\begin{pmatrix} fX/Z \\ fY/Z \end{pmatrix}}_{(x,y)^T} \in \mathbb{R}^2 \tag{3}$$

where f is again the focal length.

Let us note here that all points along the lines of projection – i.e. with the same ratio X/Z and Y/Z, respectively, as exemplified in dashed red in Figure 2 – are mapped to the same point in the image plane. Therefore, even without making it explicit, it is evident that in our setting of perspective SFS

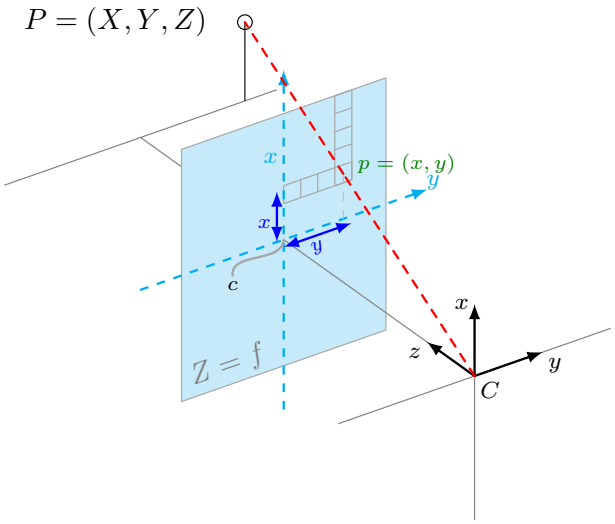


Fig. 2: A schematic view of perspective projection with an adopted pinhole camera. Here we distinguish between world coordinates and image coordinates.

there is a free *multiplicative* parameter that allows to scale the world coordinates. It is not an additive one as in the orthographic setting.

III. PERSPECTIVE SFS AS EIKONAL EQUATION

We proceed by parametrizing a surface of interest within the spherical coordinate system. In the next paragraphs we derive in detail the needed components, namely the parametrized surface, the gradient, the basis vectors, the normal vectors and the light vector all in the spherical coordinate system. Having these components in hand, we could further derive the constituting equation of PSFS in terms of an eikonal equation.

A. Surface parametrization

We consider the surface point (X,Y,Z) in world coordinate system and present it as a vector $\mathbf{u}$:

$$\mathbf{u} := \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{4}$$

or further as

$$\underbrace{r\mathbf{e}_r}_{\mathbf{u}} := \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{5}$$

Here,

$$r = \sqrt{X^2+Y^2+Z^2} \tag{6}$$

is the radial distance between a surface point of interest $(X,Y,Z)^\top$ and the source of light located at the center of the spherical coordinate system. Moreover, $\mathbf{e}_r$ represents the element of the orthonormal basis vectors in a spherical coordinate system and in direction of the radial depth coordinate r. One could further write $\mathbf{e}_r$ as

$$\mathbf{e}_r = \frac{(X,Y,Z)^T}{\sqrt{X^2+Y^2+Z^2}} \tag{7}$$

Let us now make explicit how to convert the Cartesian coordinates shown in (4) using trigonometric expressions to spherical ones, i.e.

$$\mathbf{u} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} r\sin\theta\cos\phi \\ r\sin\theta\sin\phi \\ r\cos\theta \end{pmatrix} \tag{8}$$

Here, one may obtain

$$\theta = \arccos \frac{Z}{\sqrt{X^2+Y^2+Z^2}} \tag{9}$$

and

$$\phi = \begin{cases} \arctan\left(y/x\right), & x>0 \\ \arctan\left(y/x\right)+\pi, & x<0 \wedge y \geq 0 \\ \arctan\left(y/x\right)-\pi, & x<0 \wedge y<0 \\ +\pi/2, & x=0 \wedge y>0 \\ -\pi/2, & x=0 \wedge y<0 \\ \text{undefined}, & x=0 \wedge y=0 \end{cases} \tag{10}$$

B. Gradient in Spherical Coordinates

We observe that (6), (9) and (10) display the dependency of r, θ and ϕ on the Cartesian coordinates X, Y and Z. This allows us to employ the chain rule and write the differential operators $\partial/\partial r$, $\partial/\partial\theta$ and $\partial/\partial\phi$ based on the differential operators $\partial/\partial X$, $\partial/\partial Y$ and $\partial/\partial Z$:

$$\frac{\partial}{\partial r} = \frac{\partial}{\partial X}\frac{\partial X}{\partial r} + \frac{\partial}{\partial Y}\frac{\partial Y}{\partial r} + \frac{\partial}{\partial Z}\frac{\partial Z}{\partial r} \tag{11}$$

$$\frac{\partial}{\partial \theta} = \frac{\partial}{\partial X}\frac{\partial X}{\partial \theta} + \frac{\partial}{\partial Y}\frac{\partial Y}{\partial \theta} + \frac{\partial}{\partial Z}\frac{\partial Z}{\partial \theta} \tag{12}$$

$$\frac{\partial}{\partial \phi} = \frac{\partial}{\partial X}\frac{\partial X}{\partial \phi} + \frac{\partial}{\partial Y}\frac{\partial Y}{\partial \phi} + \frac{\partial}{\partial Z}\frac{\partial Z}{\partial \phi} \tag{13}$$

However, a more compact form of (11), (12) and (13) could be written using matrix multiplication as

$$\begin{pmatrix} \partial/\partial r \\ \partial/\partial\theta \\ \partial/\partial\phi \end{pmatrix} = \begin{pmatrix} \partial X/\partial r & \partial Y/\partial r & \partial Z/\partial r \\ \partial X/\partial\theta & \partial Y/\partial\theta & \partial Z/\partial\theta \\ \partial X/\partial\phi & \partial Y/\partial\phi & \partial Z/\partial\phi \end{pmatrix} \begin{pmatrix} \partial/\partial X \\ \partial/\partial Y \\ \partial/\partial Z \end{pmatrix} \tag{14}$$

Expanding expressions using (8) one may rewrite the right hand side of the latter equation after a few computations as

$$\begin{pmatrix} \sin\theta\cos\phi & \sin\theta\sin\phi & \cos\theta \\ r\cos\theta\cos\phi & r\cos\theta\sin\phi & -r\sin\theta \\ -r\sin\theta\sin\phi & r\sin\theta\cos\phi & 0 \end{pmatrix} \begin{pmatrix} \partial/\partial X \\ \partial/\partial Y \\ \partial/\partial Z \end{pmatrix} \tag{15}$$

To derive the basis vectors $\mathbf{e}_r$, $\mathbf{e}_\theta$ and $\mathbf{e}_\phi$ and consequently the normal vector to the surface point $\mathbf{u}$, we need to invert the appearing matrix, such that we receive an expression for the gradient operator in terms of the spherical coordinates. After some computations one may obtain

$$\begin{pmatrix} \partial/\partial X \\ \partial/\partial Y \\ \partial/\partial Z \end{pmatrix} = \begin{pmatrix} \sin\theta\cos\phi & \frac{\cos\theta\cos\phi}{r} & \frac{-\sin\phi}{r\sin\theta} \\ \sin\theta\sin\phi & \frac{\sin\phi\cos\theta}{r} & \frac{\cos\phi}{r\sin\theta} \\ \cos\theta & \frac{-\sin\theta}{r} & 0 \end{pmatrix} \begin{pmatrix} \partial/\partial r \\ \partial/\partial\theta \\ \partial/\partial\phi \end{pmatrix} \tag{16}$$

as the result.

C. Basis Vectors in Spherical Coordinates

We proceed by rewriting (16) in a more compact way as

$$\begin{pmatrix} \partial/\partial X \\ \partial/\partial Y \\ \partial/\partial Z \end{pmatrix} = \mathbf{e}_r \left(\partial/\partial r\right) + \frac{1}{r}\mathbf{e}_\theta \left(\partial/\partial \theta\right) + \frac{1}{r\sin\theta}\mathbf{e}_\phi \left(\partial/\partial \phi\right) \quad (17)$$

Here $\mathbf{e}_r$, $\mathbf{e}_\theta$ and $\mathbf{e}_\phi$ are the basis vectors of the spherical coordinate system, and at the same time represents the orthonormal basis of $\mathbb{R}^3$, that are derived by normalizing the columns of the matrix in (16):

$$\mathbf{e}_r = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}, \mathbf{e}_\theta = \begin{pmatrix} \cos\theta\cos\phi \\ \cos\theta\sin\phi \\ -\sin\theta \end{pmatrix}, \mathbf{e}_\phi = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix} \quad (18)$$

By considering the derivatives in (17), we compile the components related to changes observable over the unit sphere as

$$\nabla_{(\theta,\phi)} \coloneqq \begin{pmatrix} \frac{1}{r\sin\theta}\frac{\partial}{\partial\phi} \\ \frac{1}{r}\frac{\partial}{\partial\theta} \end{pmatrix} \quad (19)$$

With the same approach, we may derive just the θ-part of the normal vector as given in a *polar coordinate system* as

$$\nabla_{(\theta)} \coloneqq \frac{1}{r}\frac{\partial}{\partial\theta} \quad (20)$$

Note that one may compute $\frac{\partial}{\partial\theta}\mathbf{e}_r$ and $\frac{\partial}{\partial\phi}\mathbf{e}_r$ as

$$\frac{\partial}{\partial\theta}\mathbf{e}_r = \frac{\partial}{\partial\theta}\begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix} = \mathbf{e}_\theta \quad (21)$$

and

$$\frac{\partial}{\partial\phi}\mathbf{e}_r = \frac{\partial}{\partial\phi}\begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix} = \sin\theta\cdot\mathbf{e}_\phi \quad (22)$$

respectively. These identities will be used for deriving the normal vector to the surface in next subsection.

D. Surface Normals in Spherical Setting

By knowing, $\mathbf{e}_r$, $\mathbf{e}_\theta$ and $\mathbf{e}_\phi$ to establish a right-handed coordinate system, the cross product $\frac{\partial\mathbf{u}}{\partial\phi}\times\frac{\partial\mathbf{u}}{\partial\theta}$ will give us a normal vector. Below we sketch the computation:

$$\begin{aligned}
\mathbf{n} &\overset{(4)}{=} \frac{\partial\mathbf{u}}{\partial\phi}\times\frac{\partial\mathbf{u}}{\partial\theta} = \frac{\partial\left(r\mathbf{e}_r\right)}{\partial\phi}\times\frac{\partial\left(r\mathbf{e}_r\right)}{\partial\theta} \\
&= r\frac{\partial r}{\partial\phi}\left(\mathbf{e}_r\times\mathbf{e}_\theta\right) + r\frac{\partial r}{\partial\theta}\left(\sin\theta\mathbf{e}_\phi\times\mathbf{e}_r\right) + r^2\left(\sin\theta\mathbf{e}_\phi\times\mathbf{e}_\theta\right) \\
&= r\frac{\partial r}{\partial\phi}\left(\mathbf{e}_\phi\right) + r\sin\theta\frac{\partial r}{\partial\theta}\left(\mathbf{e}_\phi\times\mathbf{e}_r\right) + r^2\sin\theta\left(\mathbf{e}_\phi\times\mathbf{e}_\theta\right) \\
&= r\frac{\partial r}{\partial\phi}\mathbf{e}_\phi + r\sin\theta\frac{\partial r}{\partial\theta}\mathbf{e}_\theta - r^2\sin\theta\mathbf{e}_r
\end{aligned}$$

Thus one could write the normal vector $\mathbf{n}$ as

$$\mathbf{n} = \begin{pmatrix} r\frac{\partial r}{\partial\phi} \\ r\sin\theta\frac{\partial r}{\partial\theta} \\ -r^2\sin\theta \end{pmatrix} \quad (23)$$

that realises itself with respect to the basis vectors $\left(\mathbf{e}_r,\mathbf{e}_\theta,\mathbf{e}_\phi\right)^\top$. The Euclidean norm of $\mathbf{n}$ can be computed as

$$\|\mathbf{n}\| = r\sqrt{\left(\frac{\partial r}{\partial\phi}\right)^2 + \sin^2\theta\left(\frac{\partial r}{\partial\theta}\right)^2 + r^2\sin^2\theta} \quad (24)$$

E. Illumination

As we desire to put the source of light located at the center of the spherical coordinate system, we are in a spherical coordinate system in the position to write it in a very simple format, namely

$$\omega = (0,0,-1)^T \quad (25)$$

Such a light vector has two major properties, *(i)* it always points to the center of the spherical coordinate system, and *(ii)* the inward normal vectors to any surface parametrized in the spherical coordinate system has the same direction with ω.

F. The PDE of spherical PSFS

We now put the developments together to formulate the brightness equation (26) of PSFS with no light attenuation term in spherical coordinates. We have by Lambert's law

$$I = \rho\left(\omega\cdot\frac{\mathbf{n}}{\|\mathbf{n}\|}\right) \overset{\rho=1}{\Longleftrightarrow} I\|\mathbf{n}\| = \omega\cdot\mathbf{n} \quad (26)$$

We recall that I and ω are the image irradiance of the input image and the light vector, respectively. Substituting the computed expressions (23) and (25) in (26), we obtain

$$\omega\cdot\mathbf{n} = (0,0,-1)^\top\cdot\begin{pmatrix} r\frac{\partial r}{\partial\phi} \\ r\sin\theta\frac{\partial r}{\partial\theta} \\ -r^2\sin\theta \end{pmatrix} = r^2\sin\theta \quad (27)$$

By substituting (19), (24) and (27) in (26), we come after a few more steps of computation to the *eikonal PDE in spherical coordinates*

$$\left\|\nabla_{(\theta,\phi)}u(\phi,\theta)\right\| = \sqrt{\frac{1}{I(\phi,\theta)^2}-1} \quad (28)$$

In coming sections, we prefer to work with one dimension lower than possible, i.e. inside a polar coordinate system which can easily be extended to the spherical one. We believe that this will help us to explain the way the in which the fast marching [15] is adopted and also to visualise our results more effectively. In polar system our eikonal equation (28) shows as

$$\left\|\nabla_{(\theta)}u(\theta)\right\| = \sqrt{\frac{1}{I(\theta)^2}-1} \quad (29)$$

with $\nabla_{(\theta)}$ defined as (20).

Let us briefly comment on the role of free parameters concerning the depth. As we head for drawing a parallel to the classic orthographic setting in orthographic SFS, one may think that it should be possible to have a free additive parameter w.r.t. the depth as in the orthographic SFS setting.

This may be realised by adding here a constant value η to the depth r in a polar system as

$$\hat{r} = r + \eta \tag{30}$$

However, as (20) shows, the depth parameter is part of the differential operators in a polar or spherical system. Writing

$$\nabla_{(\theta)} := \frac{\beta}{r+\eta}\frac{\partial}{\partial\theta} \tag{31}$$

setting

$$\beta := {}^{(r+\eta)}/_{r} \tag{32}$$

we see that a multiplicative parameter β can be chosen to compensate a shift η occurred for a curve as the result of its r coordinate change inside a polar system. A simple, additive shift in the r coordinate is not an invariant in this formulation.

IV. SOLVING SPHERICAL EIKONAL PDE

To solve the eikonal equation (29) we adopt a solution vector U such that

$$U_j := j \cdot h_\theta \tag{33}$$

having

- $j \in \mathbb{N}$ acting as an index,
- h_θ is a fixed step size, representing the distance among any pair of adjacent cells in our vector U.

We start by taking the operator in (20) and replace its ${}^{1}/_{r}$ term by ${}^{1}/_{u}$, so that

$$\nabla_{(\theta)} u(\theta) := \frac{1}{u(\theta)}\frac{\partial u(\theta)}{\partial\theta} \tag{34}$$

Now, we replace $\nabla_{(\theta)} u$ appeared in (29) with its discretised backward/forward approximations as

$$\sqrt{\left(\frac{1}{U_{j-1}}\cdot\frac{U_{j-1}-U_j}{h_\theta}\right)^2} = \sqrt{\frac{1}{I^2(\theta_j)}-1} \tag{35}$$

respectively

$$\sqrt{\left(\frac{1}{U_{j+1}}\cdot\frac{U_{j+1}-U_j}{h_\theta}\right)^2} = \sqrt{\frac{1}{I^2(\theta_j)}-1} \tag{36}$$

We will make use of these discrete forms in the fast marching method.

Here, h_θ, $I(\theta_j)$ and U_j are all known, whereas both U_{j-1} and U_{j+1} are unknowns. Here onward, we take steps towards rewriting (35) and (36) so that they are numerically solveable in form of a *fixed point iteration* approach. We briefly demonstrate the procedure at hand of (35). To start, we introduce the *new* and the *old* instances of U_{j-1}

$$\frac{1}{U^{old}_{j-1}}\cdot\frac{U^{new}_{j-1}-U_j}{h_\theta} = \sqrt{\frac{1}{I^2(\theta_j)}-1} \tag{37}$$

One may notice that in the context of the fast marching method we employ, in (37) we always have

$$\frac{U^{new}_{j-1}-U_j}{h_\theta} \geq 0 \tag{38}$$

and the data U^{old}_{j-1} is known.

For initialization of the above iteration, we set the unknown term U^{old}_{j-1} inside (37) using its already known neighbor U_j.

Now, we take the only unknowns in (37), namely U^{new}_{j-1} to one side that leads to

$$U^{new}_{j-1} = U_j + h_\theta \cdot U^{old}_{j-1}\sqrt{\frac{1}{I^2(\theta_j)}-1} \tag{39}$$

To solve the eikonal PDE (29), we find the fixed point at each point j iteratively. More precisely, we start by taking (39) and keep on recursively updating U^{new}_{j-1} based on U_j and U^{old}_{j-1} until $\left|U^{new}_{j-1} - U^{old}_{j-1}\right| < \varepsilon$. At that point we adopt $U_{j-1} := U^{new}_{j-1}$ and proceed at node $j-2$.

The analogous procedure can be performed for the forward discretization in the other direction.

V. EXAMPLE FOR CONVEX-CONCAVE AMBIGUITY

We start the discussion by taking the irradiance signal shown in Figure 3 which is produced based on the black curve shown in Figure 4, which we denote here as *concave curve* in analogy to the orthographic setting. Let us stress that the concave curve takes on the role of the given geometry. Note that, the irradiance signal at the point $(1, {}^{3\pi}/_{2})$ is drawn as a bullet. The reason behind is our original concave curve visualised as black inside the Figure 4 does not have a well defined gradient at the point $(6.5, {}^{3\pi}/_{2})$. Such bullet is observed in all coming irradiance signals created based on the concave curve.

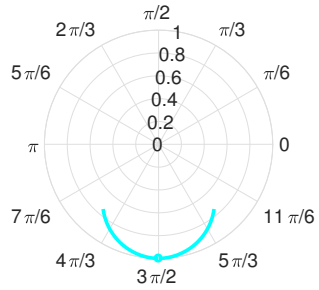


Fig. 3: The input irradiance signal corresponding to the concave curve – with the shape of a hat function – shown in Figure 4, depicted there in black. The bullet mentions that our original concave curve visualised as black inside the Figure 4 does not have a well defined gradient at the point $(6.5, {}^{3\pi}/_{2})$.

We now solve the eikonal equation (29) by providing it the given irradiance signal. This produces a *convex curve* depicted in red in Figure 4.

Let us note that we obtain here computationally a convex curve as we enforce this by the construction of the fast

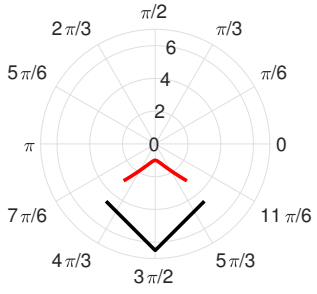


Fig. 4: A pair of concave and convex curves shown as black and red colors, respectively. The former one is the original curve and the latter one is produced by adopting the irradiance signal, shown as Figure 3, and the eikonal equation shown as (29).

marching method, as explained in the previous paragraph. The point in the numerical construction is, that in setting the sign within the square in (35) of the backward/forward differences, we force here the computed solution to have a higher depth value than of one point that must be given to the algorithm for starting. Here, we gave the depth of the top point of the convex curve to the method, leaving the construction of the actual shape to the method.

In order to see that our proceeding has indeed given the sought ambiguity, we now employ another means to verify this by asking if both curves shown in Figure 4 have the same irradiance signals. To perform this test, we adopt (26) in order to compute the irradiance signals taking into account the normal vectors that can be computed for both curves, the result of which is displayed in Figure 5. We observe the desired result, namely that both convex/concave curves shown inside the Figure 4 have effectively the same irradiance signals.

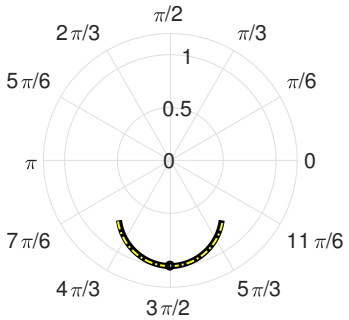


Fig. 5: A nearly indistinguishable pair of irradiance signals shown in solid black and dashed, respectively. The former belongs to the concave curve and the latter corresponds to the convex curve shown in Figure 4.

Finally, let us clarify the impact of the possible free parameters in the new model. We explore this by showing that the convex curve shown in red in Figure 6 can be transformed by keeping the same irradiance signal.

The originally computed solution itself is shown as the red curve in Figure 4, and we see that the top point has the depth $r = 1$. Thus, adding $\eta = -1$ uniformly to the r coordinates *plus* using the multiplicative change as we elaborated, we produce the transformed convex curve shown in Figure (6).

To make clear that the elaborated combination of additive and multiplicative changes enables to produce an invariant irradiance signal, we perform now exactly this clarification. Once again we produce the irradiance signals of both curves under question and show them as Figure 7. As one observes, both irradiance signals overlap each other with an inaccuracy represented with a hallow bullet around the point $(I = 1, \theta = {}^{3\pi}/_{2})$. The reason behind the hollow bullet is again that the gradient in polar system (20) is not defined if $r = 0$, that makes the irradiance signal corresponding to the transformed curve undefined. Moreover, the irradiance signal of the concave curve is not also well defined at the mentioned point.

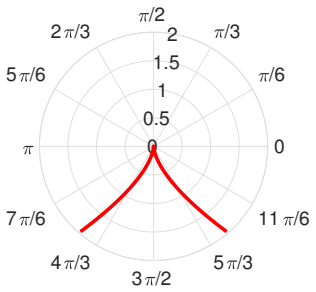


Fig. 6: Our transformed convex curve, produced by adding the value of $\eta = -1$ to the r coordinate of the computed convex curve shown in Figure 4, and adopting the corresponding multiplicative parameter to compensate the effect of the transformation. One could observe that the irradiance signal of the transformed convex curve coincides with the one corresponding to the computed convex curve shown inside the Figure 7.

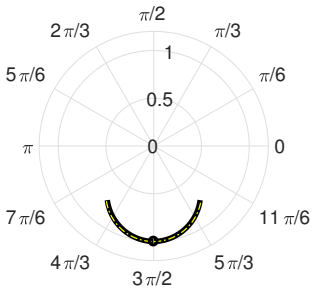


Fig. 7: Comparison of irradiance signals shown as solid black and dashed yellow. The former belongs to the *convex curve* shown in Figure 4 and the latter corresponds to the transformed curve shown inside the Figure 6. The hollow bullet represents the fact that the irradiance values of the curve shown in Figure 6 around the point $(I = 1, \theta = {}^{3\pi}/_{2})$ is not defined. The curves coincide to a large degree.

VI. CONCLUSION

We have verified the existence of the convex/concave ambiguity in a perspective SFS model. In our presented example

we have assured that every step is validated with respect to this goal. Our assumption about the light position as well as the decision not to take into account an attenuation term allowed our constituting equation of PSFS to be represented as an eikonal equation of the identical form as in classic, orthographic SFS. The main point in our proceeding was thereby the use of the spherical coordinate system. It is evident that the described proceeding for the construction of the ambiguity can easily be performed in the full spherical system i.e. for images instead of signals. In future work we will extend the considerations about ambiguities and work on ways to resolve them.

REFERENCES

[1] M. Breuß, E. Cristiani, J.-D. Durou, M. Falcone, and O. Vogel, Perspective Shape from Shading: Ambiguity Analysis and Numerical Approximations. SIAM Journal on Imaging Sciences, 5 (2012), 1, 311–342.

[2] A.R. Bruss, The eikonal equation: some results applicable to computer vision. Journal of Mathematical Physics, 23 (1982), 890–896.

[3] F. Courteille, A. Crouzil, J.-D Durou, and P. Gurdjos: Towards shape from shading under realistic photographic conditions. In Proc. ICPR, 2004, 277–280.

[4] S. Galliani, Y.C. Ju, M. Breuß and A. Bruhn: Generalised Perspective Shape from Shading in Spherical Coordinates. In Proc. SSVM, 2013, 222–233.

[5] B.K.P. Horn, Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View. PhD thesis, Department of Electrical Engineering, MIT, Cambridge, Massachusetts, USA, 1970.

[6] B.K.P. Horn, Robot Vision. MIT Press, 1986.

[7] B.K.P. Horn and M.J. Brooks, Shape from Shading. Artificial Intelligence Series, MIT Press, 1989.

[8] Y.C. Ju, S. Tozza, M. Breuß, A. Bruhn and A. Kleefeld, Generalised Perspective Shape from Shading with Oren-Nayar Reflectance. In Proc. BMVC, 2013, Article 42.

[9] J.J. Koenderink, What does the occluding contour tell us about solid shape? Perception, 13(3):321330, 1984.

[10] K.M. Lee and C.C.J. Kuo, Shape from Shading with Perspective Projection. Computer Vision, Graphics, Image Processing: Image Understanding 59, No. 2 (1994), 202–211.

[11] B. Liu and J.T. Todd, Perceptual biases in the interpretation of 3d shape from shading. Vision research, 44(18):21352145, 2004.

[12] E. Prados and O. Faugeras, Perspective Shape from Shading and Viscosity Solutions. In Proc. ICCV, 2003, vol. II, 826–831.

[13] E. Prados, F. Camilli, and O. Faugeras, A unifying and rigorous shape from shading method adapted to realistic data and applications. J. Math. Imag. and Vis. 25 (2006), 3, 307–328.

[14] E. Prados, F. Camilli, and O. Faugeras: A viscosity solution method for shape-from-shading without image boundary data. M2AN Math. Model. Numer. Anal. 40 (2006), 2, 393–412.

[15] J.A. Sethian, Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press, 1999.

[16] A. Tankus, N. Sochen, and Y. Yeshurun, A New Perspective [on] Shape-from-Shading. In: Proc. 9th IEEE Int. Conf. Comp. Vis. (vol. II), Nice, France, October 2003, pp. 862–869.

[17] R. Zhang, P.S. Tsai, J.E. Cryer, and M. Shah, Shape-from-shading: a survey. IEEE Trans. Pattern Anal. Mach. Intell., 21 (1999), 690–706.

Contributed Session 3

Fast Solvers for Solving Shape Matching by Time Integration

Martin Bähr, Robert Dachsel and Michael Breuß

Abstract— **The main task in three-dimensional non-rigid shape correspondence is to retrieve similarities between two or more similar three-dimensional objects. An important building block of many methods constructed to achieve this goal is a simplified shape representation called feature descriptor, which is invariant under almost isometric transformations. A recent feature descriptor relies on the full numerical integration of the geometric heat equation. This approach involves to solve a system of linear equations with multiple right-hand sides. To this end, it is necessary to find a fast and accurate numerical scheme in conjunction with the solution of a sparse linear system and many different right sides. In this paper we evaluate direct, iterative and model order reduction (MOR) methods and their influence to shape correspondence applications which will be validated on standard shape data sets with different resolutions.**

I. INTRODUCTION

The examination of shape correspondence is a fundamental task in computer vision, pattern recognition and geometry processing. Performing the shape correspondence process is a key component for problems such as 3D scan alignment or space-time reconstruction and is essential in various applications including shape interpolation and statistical modelling, see [21]. The fundamental task of shape correspondence is to identify an explicit relation between the elements of two or more given shapes, whereby the shape of a three-dimensional geometric object can be described by its bounding surface. In this context, a challenging setting is that of non-rigid shape correspondence, where the shapes are almost isometric. Almost isometric shapes lead to a large variety of possible deformations such as poses of human or animal bodies.

An important solution strategy is to achieve a pointwise non-rigid shape correspondence using so called descriptor based methods. For this, a feature descriptor has to be developed which characterizes each element on the shape regarding its geometric relation. An interesting type of descriptors is based on the spectral decomposition of the Laplace-Beltrami operator, see e.g. [17], [20]. However, these methods rely on the expansion of eigenfunctions of the Laplace-Beltrami operator, which is for instance used to approximate the solution of the geometric heat equation, cf. [2]. A recent alternative compared to eigenfunction expansion methods is based on the full numerical integration of the underlying partial differential equations (PDEs), cf. [3]. Experiments based on time integration methods confirm a substantially higher accuracy compared to spectral methods in many cases.

Institute for Mathematics, Brandenburg Technical University, Platz der Deutschen Einheit 1, 03046, Cottbus, Germany {baehr,dachsel,breuss}@b-tu.de

Application of time integration methods leads to a new non-negligible challenge – solving a system of linear equations with multiple right-hand sides. Dealing with large sparse systems implicates two main issues, the accuracy of the solution and the computational efficiency of the numerical solver. Generally, direct and iterative methods are the most common solvers to compute the solution. However, almost always it remains an open question, which of both is the best choice to solve the underlying problem. Direct methods for solving the same system for different right sides, are fast and offer an extremely high accuracy. However, this type of solvers may use substantial memory and appears to be rather impractical for shapes with many thousands of points. In contrast, iterative methods are naturally not tweaked for extremely high accuracy but are very fast in computing approximate solutions. They require less memory space and are thus inherently more attractive candidates for this application. Nevertheless, the runtime of iterative methods depends on the data, size, sparsity and required accuracy and makes a tool that is not straightforward to use.

Let us mention, that the number of the right-hand sides of the underlying system are directly related to the number of points of the regarded shapes. Therefore, the increase of the size of the point cloud defining a shape leads to an extreme rise of the computational effort. Due to this fact, an alternative approach is the use of model order reduction. Such techniques can be used to approximate the full system by a significantly reduced model, that is much faster to solve than the original system. In this work, we consider the modal coordinate reduction (MCR), which involves the use of projection matrices to approximate the geometric heat equation. The accuracy of MCR for a given problem depends on the number and structure of equations. In case of shape matching we compare the correspondence of several shapes. Therefore, a correct matching depends on a good numerical quality of the physical process on each of the regarded shapes. For this reason, the application of the MCR method could lead to a more sensitive result with respect to the quality of the matching.

Our Contributions. In this work, we address the mentioned challenges by investigating numerical methods for computing feature descriptors based on time integration methods. To this end, we will compare direct, iterative and MCR methods in terms of accuracy and computational efficiency for shape matching by application of the classic feature descriptor defined via the geometric heat equation.

II. THEORETICAL BACKGROUND

In this section, we introduce the basic facts that are necessary to define the shape matching framework.

A. Almost Isometric Shapes

The shape of a three-dimensional geometric object can be described by its bounding surface. This is a two-dimensional curved object, embedded into a three-dimensional Euclidean space. In this paper, we model shapes as compact two-dimensional Riemannian manifolds $\mathscr{M}$, equipped with metric tensor $g \in \mathbb{R}^{2\times 2}$.

In this setting, two shapes $\mathscr{M}$ and $\tilde{\mathscr{M}}$ may be considered as isometric if there is a map T that unfolds one surface onto the other by preserving the intrinsic distance. From the mathematical point of view there exists a smooth homeomorphism $T : \mathscr{M} \to \tilde{\mathscr{M}}$ with

$$d_{\mathscr{M}}(x_1,x_2) = d_{\tilde{\mathscr{M}}}(T(x_1),T(x_2)) \quad \forall x_1,x_2 \in \mathscr{M} \tag{1}$$

where $d_{\mathscr{M}}(x_1,x_2)$ is the intrinsic distance. The intrinsic distance can be thought as the shortest curve along the surface $\mathscr{M}$ connecting x_1 and x_2.

The notion of purely isometric shapes appears too restrictive. Many shapes include an additional small elastic deformation. These occur by either the transformation itself, noisy datasets or by transferring the continuous shape into a numerical framework. This acts as a "small" distortion for the intrinsic distance. We call two shapes $\mathscr{M}$ and $\tilde{\mathscr{M}}$ almost isometric, if there exists a transformation $S : \mathscr{M} \to \tilde{\mathscr{M}}$ with

$$d_{\mathscr{M}}(x_1,x_2) \approx d_{\tilde{\mathscr{M}}}(S(x_1),S(x_2)) \quad \forall x_1,x_2 \in \mathscr{M} \tag{2}$$

B. The Heat Equation on Shapes

The heat equation that yields a useful descriptor involves the Laplace operator when considering the Euclidean plane. In order to respect the curvature of a manifold in 3D, techniques from differential geometry are employed [5]. The resulting Laplace-Beltrami operator is defined on a smooth manifold $\mathscr{M}$. In this context, let us recall that for a given parameterisation of a two-dimensional manifold, the Laplace-Beltrami operator applied to a scalar function $u : \mathscr{M} \to \mathbb{R}$ can be expressed in local coordinates:

$$\Delta_{\mathscr{M}} u = \frac{1}{\sqrt{|g|}} \sum_{i,j=1}^{2} \partial_i \left(\sqrt{|g|} g^{ij} \partial_j u \right) \tag{3}$$

where g^{ij} are the components of the inverse of the metric tensor and $|g|$ is its determinant.

The *geometric heat equation* describes how heat would propagate along a surface $\mathscr{M}$ and can be formulated as

$$\begin{cases} \partial_t u(x,t) = \Delta_{\mathscr{M}} u(x,t) & x \in \mathscr{M}, t \in I \\ u(x,0) = \exp\left(-\frac{d_{\mathscr{M}}(x-x_i)^2}{2\sigma^2}\right) & \\ \langle \nabla_{\mathscr{M}} u, \mathbf{n} \rangle = 0 & x \in \partial\mathscr{M} \end{cases} \tag{4}$$

where the initial condition $u(x,0)$ is a given by a highly peaked Gaussian heat distribution. In this context σ^2 is the variance parameter and $x_i \in \mathscr{M}$ is the centre of the Gaussian distribution. Many shapes appear as a closed manifold with $\partial\mathscr{M} = \emptyset$, where boundary conditions do not appear. For the case $\mathscr{M}$ has boundaries, we additionally require u to satisfy homogeneous Neumann boundary conditions, where $\mathbf{n}$ is the normal vector to the boundary.

C. The Feature Descriptor and Shape Correspondence

a) Feature Descriptor: Considering the surface itself is unsuitable for many shape analysis tasks. A simplified representation is needed which is often called a feature descriptor. In this context, the feature descriptor stores the geometry of the surface at a certain local region. We restrict the spatial component of $u(x,t)$ to

$$f_{x_i}(t) = u(x,t)_{|x=x_i} \text{ with } u(x,0) = \exp\left(-\frac{d_{\mathscr{M}}(x-x_i)^2}{2\sigma^2}\right) \tag{5}$$

and call the f_{x_i} the feature descriptors at the location $x_i \in \mathscr{M}$. Let us note that there exists a physical interpretation of the feature descriptors. The heat based feature descriptor describes the rate of heat transferred away from the considered point x_i. Since we used an intrinsic approach, let us note, the feature descriptor can not distinguish between intrinsic symmetry groups.

b) Shape Correspondence: To compare the feature descriptors for different locations $x_i \in \mathscr{M}$ and $\tilde{x}_j \in \tilde{\mathscr{M}}$ on respective shapes $\mathscr{M}$ and $\tilde{\mathscr{M}}$, we simply define a distance $d_f(x_i,\tilde{x}_j)$ using the L_1 norm

$$d_f(x_i,\tilde{x}_j) = \int_I |f_{x_i} - f_{\tilde{x}_j}| \, \mathrm{d}t \tag{6}$$

The tuple of locations $(x_i,\tilde{x}_j)$ with the smallest feature distance should belong together. This condition can be written as a minimisation problem for all locations:

$$(x_i,\tilde{x}_j) = \underset{\tilde{x}_k \in \tilde{\mathscr{M}}}{\arg\min}\, d_f(x_i,\tilde{x}_k) \tag{7}$$

By using $\tilde{x}_j = S(x_i) = x_i$, the map S can pointwise be restored for all x_i. Let us mention, that without further alignment the restored map S is neither injective nor surjective because the minimisation condition is not unique.

III. DISCRETISATION ASPECTS

As indicated we will construct a feature descriptor by direct discretisation of the geometric heat equation. In order to approximate the equation on a shape we have to take care of three aspects. First, a discrete approximation of our continuous and closed surface as well as of the time domain is needed. Second, a suitable discrete Laplace-Beltrami operator has to be defined. Third, quadrature formulas need to be used to approximate the time integration.

We start with the integrated form of the geometric heat equation in (4) over time and space

$$\int_I \int_{\mathscr{M}} \partial_t u(x,t) \, \mathrm{d}x \, \mathrm{d}t = \int_I \int_{\mathscr{M}} \Delta_{\mathscr{M}} u(x,t) \, \mathrm{d}x \, \mathrm{d}t \tag{8}$$

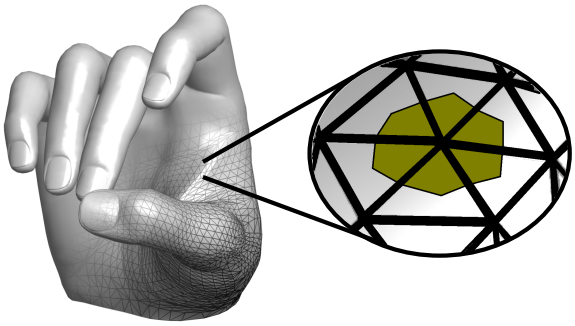


Fig. 1. Continuous and discrete representation of a shape. The discrete approximation of a shape is given by non-uniform and linear triangles. The construction of volume cells is done by using the barycentric area around a vertex.

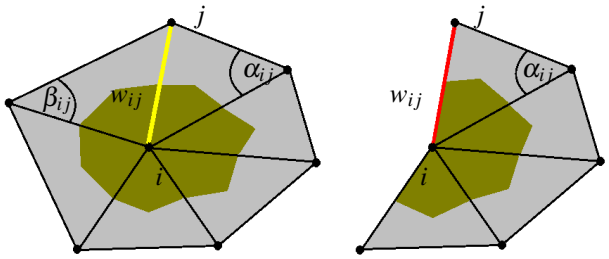


Fig. 2. The cotangent weight scheme as discretisation of the Laplace-Beltrami operator. Left: interior edge Right: boundary edge

A. Discrete Space and Time Domain

A suitable surface representation is given by a triangular mesh, cf. Figure 1. In more detail, a triangulated surface is given by the tuple $\mathscr{M}_d = (P,T)$. The point cloud $P := \{x_1,\dots,x_N\}$ contains the finite number of vertices (given as coordinate points) a shape consists of. The entire mesh can be formed by connecting the vertices x_i so that one obtains two-dimensional triangular cells. Therefore, the set of linear triangles T contains the neighborhood relations between vertices forming a triangle.

Further, we sub-divide the meshed surface and the time axis as follows:

$$\mathscr{M}_d = \bigcup_{i=1}^{N} \Omega_i \quad \text{and} \quad I = \bigcup_{i=1}^{M} I_k \tag{9}$$

where Ω_i is the barycentric cell volume surrounding the i-th vertex. For time, let $I_k = [t_{k-1}, t_k]$ and $t_0 = 0$, where the time increment $\tau = t_k - t_{k-1}$ for all $k \in \{1,\dots,M\}$ is uniformly chosen.

B. Finite Volume Approach

Now we restrict the integrated geometric heat equation to Ω_i and I_k. For values $x \in \Omega_i$ and $t \in I_k$ we have

$$\int_{I_k}\int_{\Omega_i} \partial_t u(x,t)\,\mathrm{d}x\,\mathrm{d}t = \int_{I_k}\int_{\Omega_i} \Delta_{\mathscr{M}}\, u(x,t)\,\mathrm{d}x\,\mathrm{d}t \tag{10}$$

Further we use the definition of the cell average

$$u_i(t) = u(\bar{x}_i,t) = \frac{1}{|\Omega_i|}\int_{\Omega_i} u(x,t)\,\mathrm{d}x \tag{11}$$

where $|\Omega_i|$ is the surface area of the i-th cell. Now, we apply the divergence theorem to substitute the volume integral on the right-hand side into a line integral over the boundary of the volume cell to define the averaged Laplacian of the i-th cell

$$Lu_i(t) = \frac{1}{|\Omega_i|}\int_{\Omega_i} \Delta_{\mathscr{M}}\, u(x,t)\,\mathrm{d}x \tag{12}$$

A function defined on all cells is represented by an N-dimensional vector $\mathbf{u}(t) = (u_1(t),\dots,u_N(t))^\top$. Using the mean value expression for equation (10), we obtain a system of integrated ODEs:

$$\int_{I_k} \partial_t \mathbf{u}(t)\,\mathrm{d}t = \int_{I_k} L\mathbf{u}(t)\,\mathrm{d}t \tag{13}$$

C. Discrete Spatial Operator

Many schemes have been proposed to estimate the Laplace-Beltrami operator for a triangular meshed surface [13], [15]. A commonly used method is the cotangent weight scheme introduced in [11]. As a result, for a function $\mathbf{u}$ defined on a triangular mesh the discrete Laplace-Beltrami operator $L \in \mathbb{R}^{N\times N}$ reduces to the following sparse matrix representation

$$L = D^{-1}W \tag{14}$$

The symmetric weight matrix W reads component-wise

$$W_{ij} = \begin{cases} -\sum\limits_{j\in N_i} w_{ij} & \text{if } i = j \\ w_{ij} & \text{if } i \neq j \text{ and } j \in N_i \\ 0 & \text{else} \end{cases} \tag{15}$$

where N_i denotes the set of points adjacent to x_i. The weights w_{ij} of the edge (i,j) can be classified into interior E_i and boundary edges E_b respectively

$$w_{ij} = \begin{cases} \frac{\cot\alpha_{ij}+\cot\beta_{ij}}{2} & \text{if } (i,j) \in E_i \\ \frac{\cot\alpha_{ij}}{2} & \text{if } (i,j) \in E_b \end{cases} \tag{16}$$

as shown in Figure 2. Furthermore, α_{ij} and β_{ij} denote the two angles opposite to the edge (i,j), for details we refer to the mentioned source. The Matrix

$$D = \mathrm{diag}\,(|\Omega_1|,\dots,|\Omega_i|,\dots,|\Omega_N|) \tag{17}$$

contains the local cell areas. Let us note that L is not symmetric after computing the matrix product.

D. Discrete Time Integration

Discrete time integration of ODEs can be done by using standard numerical methods through time step methods. Common time integration schemes are the explicit Euler method, the implicit Euler method and the trapezoidal rule known as Crank-Nicolson method. The Crank-Nicolson method is also an implicit method, however it is second-order in time and will only produce slightly more computational

cost than the implicit Euler method. For this reason, we consider only the explicit Euler method and the Crank-Nicolson method.

a) Explicit Euler Method: As a first step we apply the fundamental lemma of calculus for the left-hand-side of (13)

$$\int_{I_k} \partial_t \mathbf{u}(t)\,\mathrm{d}t = \int_{t_{k-1}}^{t_k} \partial_t \mathbf{u}(t)\,\mathrm{d}t = \mathbf{u}(t_k) - \mathbf{u}(t_{k-1}) \tag{18}$$

Finally we approximate the integral on the right-hand side by using the rectangle method

$$\int_{t_{k-1}}^{t_k} L\mathbf{u}(t)\,\mathrm{d}t \approx \tau L\mathbf{u}(t_{k-1}) \tag{19}$$

and by using the notation $\mathbf{u}(t_k) = \mathbf{u}^k$ we obtain

$$\mathbf{u}^k = (I + \tau L)\,\mathbf{u}^{k-1} \tag{20}$$

where $k \in \{1,\dots,M\}$ and $\mathbf{u}^0 = u_0$. Due to the fact that the values of u^{k-1} are known, we can easily compute the corresponding values u^k at time k by simple matrix-vector multiplication. This explicit scheme is known to be just conditionally stable, see [19]. The stability requirement yields a limitation on the size of the time step τ.

b) Crank-Nicolson Method: If we apply the trapezoidal rule at the integral of the right-hand-side of (13)

$$\int_{t_{k-1}}^{t_k} L\mathbf{u}(t)\,\mathrm{d}t \approx \frac{\tau}{2}\left(L\mathbf{u}(t_k) + L\mathbf{u}(t_{k-1})\right) \tag{21}$$

we obtain finally

$$\left(I - \tfrac{\tau}{2}L\right)\mathbf{u}^k = \left(I + \tfrac{\tau}{2}L\right)\mathbf{u}^{k-1} \tag{22}$$

To compute the values u^k at time k it requires solving a system of linear equations as well as a matrix-vector multiplication in each time step. Therefore, it is numerically more intensive than the explicit Euler method, however it has second-order accuracy in time. The considerable advantage of an implicit scheme is the numerical stability independently of the time step size τ, cf. [19]. However, the Crank-Nicolson method is sensitive for problems with discontinuous initial conditions.

IV. Numerical Solvers

An essential key requirement for a correct shape matching is a sufficient accuracy of the computed numerical solution. However, the geometric heat equation has to be solved for each point and on each shape for a fixed time interval $t \in (0, t_M]$. Consequently, the computational costs are directly related to the number of points of the regarded shapes. This fact suggests that one may forego high accuracy in exchange for a faster computational time. Therefore, a qualitative analysis of numerical methods for the geometric heat equation in context to shape matching is absolutely essential.

As seen in the last section, the temporal integration can either be done explicitly or implicitly. For both approaches there exist several numerical solvers, which have different advantages in terms of computational effort and accuracy of the computed solution. In the following, we give a short overview.

A. Explicit Methods

Explicit schemes are simple iterative schemes of the form $u^k = (I + \tau L)u^{k-1}$ such as (20). The typical time step restriction has a rather small upper bound and makes these methods unsuitable for shape matching. An alternative is the usage of the Fast Explicit Diffusion (FED) or Fast Semi-Iterative (FSI) scheme, which is well-known in image processing. For a detailed presentation of FED or FSI we refer to [6], [7]. The core idea behind FSI is to consider an explicit scheme and interleave time steps that significantly violate the upper stability bound with small stabilising steps. To decrease numerical rounding errors and simultaneously increase the approximation quality, FSI uses cycles of varying time steps. The cyclic FSI scheme, which accelerates the explicit diffusion scheme (20), for the m-th cycle with cycle length n is given by

$$u^{m,k} = \alpha_k \cdot (I + \tau L)u^{m,k-1} + (1 - \alpha_k) \cdot u^{m,k-2} \tag{23}$$

$$n = \left\lceil \sqrt{\frac{3t_M}{\tau_{\max} \cdot C} + \frac{1}{4}} - \frac{1}{2} \right\rceil, \quad \tau = \frac{3t_M}{C \cdot n(n+1)} \tag{24}$$

$$\alpha_k = \frac{4k+2}{2k+3}, \quad u^{m,-1} = u^{m,0}, \quad k = 1,\dots,n \tag{25}$$

where C is the number of outer FSI cycles, t_M the diffusion time and $\tau_{\max}$ the theoretical upper bound for a stable explicit finite difference scheme. The FSI scheme can be applied whenever the matrix L in (20) is negative semidefinite and symmetric. The underlying matrix L is not symmetric, however by multiplication of D to equation (20) we have

$$D\mathbf{u}^k = (D + \tau W)\mathbf{u}^{k-1} \tag{26}$$

where W is symmetric and negative semidefinite.

B. Implicit Methods

Implicit schemes result in a linear system of equations (compare (22)) and lead theoretically to an unconditionally stable scheme without a time step restriction. However, solving linear equations requires significant computational effort and therefore a fast solver for large sparse linear systems of equations is necessary.

Standard methods for solving linear systems are direct and iterative solvers. Direct methods compute highly accurate solutions and are predestined for solving a system with multiple right-hand sides. In that case, the underlying matrix will be factorised one-time, and subsequently each right hand side is solved by forward and backward substitution. Due to the fact that the underlying system matrix is sparse the computational costs for the substitution step will be at most $\mathcal{O}(n^2)$, where n is number of equations. In contrast, iterative solvers can compute approximate solutions in a very fast way. A particular class of iterative solvers designed for use with large sparse linear systems is the class of *Krylov*

subspace solvers; for a detailed exposition see [18]. The main idea behind the Krylov approach is to search for an approximative solution of $A\mathbf{x} = \mathbf{b}$, with $A \in \mathbb{R}^{n\times n}$ a large regular sparse matrix and $\mathbf{b} \in \mathbb{R}^n$, in a suitable low-dimensional subspace $\mathbb{R}^l$ of $\mathbb{R}^n$ that is constructed iteratively with l being the number of iterates. The aim in the construction is thereby to have a good representation of the solution after a few iterates. Let us note that this construction is often not directly visible in the formulation of a Krylov subspace method.

We propose to employ the well-known conjugate gradient (CG) scheme of Hestenes and Stiefel [8], which is still an adequate iterative solver for problems involving sparse symmetric and positive definite matrices. Let us note, if we multiply the matrix D to equation (22), we obtain

$$\left(D - \tfrac{\tau}{2}W\right)\mathbf{u}^k = \left(D + \tfrac{\tau}{2}W\right)\mathbf{u}^{k-1} \tag{27}$$

such that $\left(D - \frac{\tau}{2}W\right)$ is symmetric positive definite. With $A = \left(D - \frac{\tau}{2}W\right)$, $b = \left(D + \frac{\tau}{2}W\right)u^{k-1}$ and $x = u^k$ the latter equation corresponds to solving a system $Ax = b$ at time k. For the CG method, one can show that the approximate solutions $\mathbf{x}_l$ (in the l-th iteration) are optimal in the sense that they minimise the so-called energy norm of the error vector. In other words, the CG method gives in the l-th iteration the best solution available in the generated subspace [10]. Since the dimension of the Krylov subspace is increased in each iteration, theoretical convergence is achieved at latest after the n-th step of the method if the sought solution is in $\mathbb{R}^n$. The CG algorithm requires in each iteration a sparse matrix-vector-multiplication. One main advantage is that a practical solution can be reached quickly after a small number l of iterations, which yields to a quick termination of the CG method and total costs of at most $\mathcal{O}(ln^2)$. However, in practice numerical rounding errors appear and one may suffer from convergence problems for very large systems. Thus, a *preconditioning* is recommended to enforce all the beneficial properties of the algorithm, along with fast convergence [1]. Moreover, it may require fine-tuning of parameters in the preconditioned conjugated gradient method (PCG) and in addition increase potentially the computational cost.

C. Model Order Reduction

The introduced explicit and implicit methods have to handle large sparse systems, whereby the computational costs depends on the point cloud size. Model Order Reduction (MOR) techniques can be used to approximate the full ODE system by a very low dimensional system, while preserving the main characteristic of the original ODE system. Existing MOR techniques [12], [14] are polynomial, projection and non-projection based methods. In this work, we apply the widely used modal coordinate reduction (MCR) method, which is a projection based approach. The concept of MCR is to transform the full model from physical coordinates in physical space to modal coordinates in modal space by using the eigenvector matrix of this system. Subsequently, it removes those modes that have less important contributions to the system responses. Generally, only a few modes have a significant effect on the system dynamics within the frequency range of interest.

After discretisation of the PDE (4) in space, we obtain a system of ODEs (compare (13)) in format

$$u'(t) = Au(t) \tag{28}$$

with a system matrix $A \in \mathbb{R}^{n\times n}$. For A being diagonalisable, there exists a matrix $S \in \mathbb{R}^{n\times n}$ with eigenvectors of A and a diagonal matrix $\Lambda \in \mathbb{R}^{n\times n}$ with the corresponding eigenvalues λ_i such that

$$A = S\Lambda S^{-1} \tag{29}$$

Inserting of (29) in (28) and the subsequent multiplication of S^{-1} leads to

$$S^{-1}u'(t) = \Lambda S^{-1}u(t) \tag{30}$$

The latter equation is the starting point for a strategic selection of eigenvalues and eigenvectors (modes). It is well-known, that the low frequencies (corresponding to small eigenvalues) dominate the dynamics of the underlying physical system. Suppose $m \ll n$ ordered eigenvalues $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_m$ are deemed of interest. Consequently, we obtain with $\widetilde{\Lambda} \in \mathbb{R}^{m\times m}$ of Λ and $\widetilde{S} \in \mathbb{R}^{n\times m}$ the reduced model of order m

$$w'(t) = \widetilde{\Lambda}w(t) \tag{31}$$

where $w = \widetilde{S^{-1}u}$. This low dimensional system is much faster to solve than the original one. Applying the Crank-Nicolson method to (31) based solely on diagonal matrices, such that implicit method can solved by only matrix-vector-multiplications.

V. EXPERIMENTAL RESULTS

In general, we perform a dense point-to-point correspondence, involving all vertices the shapes are made off. In detail, the experiments are evaluated as follows:

a) Hit Rate: The percentage Hit Rate is defined as $TP/(TP+FP)$, where TP and FP are the number of true positives and false positives, respectively.

b) Geodesic Error: For the evaluation of the correspondence quality, we followed the Princeton benchmark protocol [9]. This procedure evaluates the precision of the computed matchings x_i by determining how far are those away from the actual ground-truth correspondence x^*. Therefore, a normalised intrinsic distance $d_{\mathscr{M}}(x_i, x^*)/\sqrt{A_{\mathscr{M}}}$ on the transformed shape is introduced. Finally, we accept a matching to be true if the normalised intrinsic distance is smaller than the threshold 0.25.

c) Dataset: For experimental evaluation, we compare datasets at three different resolutions, namely small, middle and large. For the small ($N = 4344$) and medium ($N = 27894$) shapes, examples of the wolf and cat class are used, taken from the TOSCA data set [2]. The Fat Baby shapes have a large resolution ($N = 59727$) and are taken from the KIDS dataset [16]. The datasets are available in the public domain as shown in Figure 3. All shapes provide ground-truth and degenerated triangles were removed.

d) General Parameters: We set the stopping time to $t_M = 25$ and the variance parameter $\sigma = 1$ for the still free parameters of the geometric diffusion process in (4). For the implicit methods the time increment $\tau = 1$ was fixed for all experiments. All three parameters are chosen without a fine-tuning, since we are interested to figure out the differences of the numerical methods compared to accuracy and computational costs.

All experiments were done in MATLAB R2017b on a recent Desktop Computer with an Intel Xeon(R) CPU E5-2609 v3 CPU running at 6x 1.9GHz and XGB of 15.6 GB RAM. The sparse linear system for the direct method was prefactorised with the SuiteSparse package [4]. The computed eigenvalues and eigenvectors for MCR are computed by the Matlab internal function *eigs*.

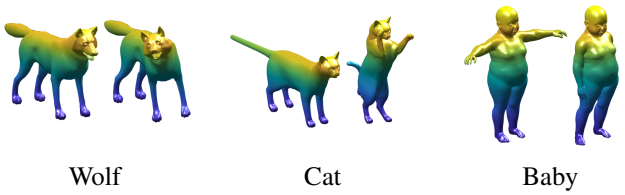


Fig. 3. For experimental evaluation, we compare shapes at three different resolutions, namely small, middle and large. These are represented by the "wolf", "cat" and "baby" , taken from the TOSCA dataset.

A. Numerical Evaluation

a) Experiment - Wolf: First of all we analyse the wolf shape with a point cloud size of only $N = 4344$ points. In case of an explicit method (20) we get the time step restriction $\tau_{\max} \approx 0.0085$s, which corresponds to around 2900 iterations. The CPU time (in seconds) of the explicit method with around 360s offers unacceptable running costs and is consequently quite inefficient. In contrast, the FSI scheme for (26) takes control over the individual time steps. Due to the fact, that the final stopping time is fix we can only specify the number of cycles C. Increasing C, whereby the cycle length n becomes smaller, improves the accuracy of FSI compared to the geodesic error of the standard explicit method, see Figure 4. Already for $C = 2$ we can achieve respectable results with an additional dramatic speed up of the explicit

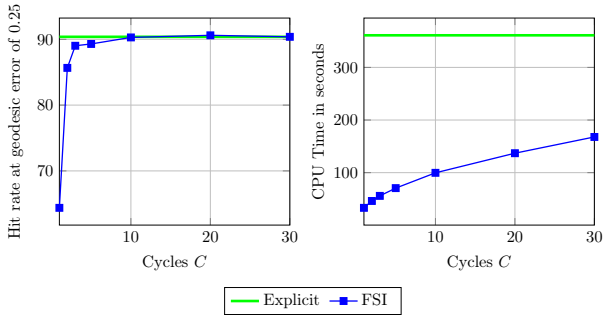


Fig. 4. Results on the dataset wolf. We compare the geodesic error up to 0.25 (left) and the performance time (right) between the explicit method and the FSI scheme for different number of cycles C.

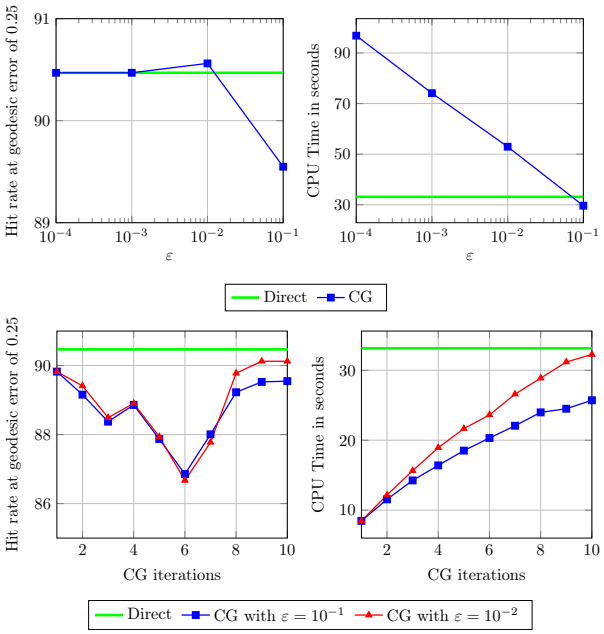


Fig. 5. Results on the dataset wolf. We compare the geodesic error up to 0.25 (left) and the performance time (right) between the direct method and the CG method for different ε (top) and different number of CG iterations l (bottom) for $\varepsilon = 10^{-1}$ and $\varepsilon = 10^{-2}$. We observe a dint-like effect (first down, then up) w.r.t. the hit rate when increasing CG iterations.

method. However, we will see that the computational costs of FSI of around 50s are still not too efficient.

The CPU time can be reduced to around 33s by using the direct method for (22) and generates the same geodesic error accuracy as the explicit method. Solving the linear system (27) with the CG method requires a stopping criterion. In general the *relative residual* $\frac{\|\mathbf{b}-A\mathbf{x}\|_2}{\|\mathbf{b}\|_2} \leq \varepsilon$ will be used. Increasing ε leads to a faster CG approximation, however the accuracy remained almost unchanged also for the relatively large value $\varepsilon = 10^{-1}$ cf. Figure 5. For this reason we tested CG for $\varepsilon = 10^{-1}, 10^{-2}$, and different number l of CG iterations, see also Figure 5. In this case, the reduction of l compared to the geodesic error accuracy leads to slight oscillations, which are still acceptable, yet with a fast CPU time of around 10s.

Finally, we explore the MCR technique. The whole MCR process includes the computation of eigenvalues, eigenvectors and the subsequent solving of the resulting reduced system. For this experiment we increase the number of used ordered modes, starting from $N_{max} = 1$ and end up to $N_{max} = 3000$. The evaluation in Figure 6 shows that the accuracy of the geodesic error depends on the number of used modes. For a larger amount of modes the accuracy increases significantly. However, it is remarkable that the results for the small spectrum $N_{max} \approx 10$ are similar to the large spectrum $N_{max} \approx 1000$. Only from a certain size ($N_{max} \approx 2000$) upwards we receive an acceptable matching result, however in unacceptable running time. Nevertheless, the CPU time by using a smaller number of modes is unbeatable. For $N_{max} = 100$ modes the approximative solution is computed

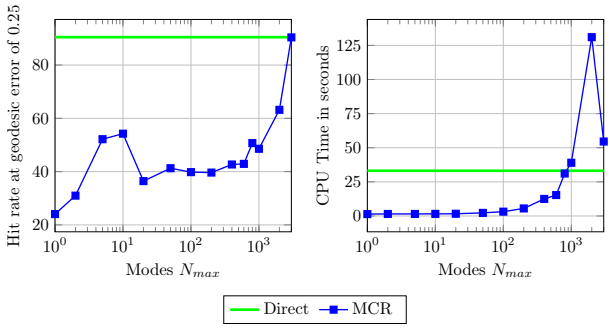


Fig. 6. Results on the dataset wolf. We compare the geodesic error up to 0.25 (left) and the performance time (right) between the direct method and the MCR technique for different number of modes $N_{\max}$.

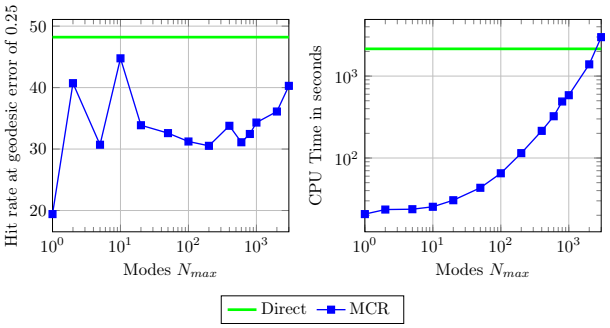


Fig. 8. Results on the dataset cat. We compare the geodesic error up to 0.25 (left) and the performance time (right) between the direct method and the MCR technique for different number of modes $N_{\max}$.

in 3s under consideration of a much less geodesic error accuracy.

b) Experiment - Cat: In the following, we consider the medium dataset cat with a cloud point size of $N = 27894$ points. Although the FIS scheme outperforms the explicit method, it is quite inefficient in consequence of the large spectrum of eigenvalues, which depends on the mesh size of the discretised shape. The mesh size is here very small, as is often the case in shape matching applications, and therefore the allowed time step $0 < \tau_{\max} \ll 1$ is also extremely small-sized. Consequently, the dramatic rise of the computational effort can not intercept the low costs of matrix-vector-multiplication.

The direct method solves the linear system in around

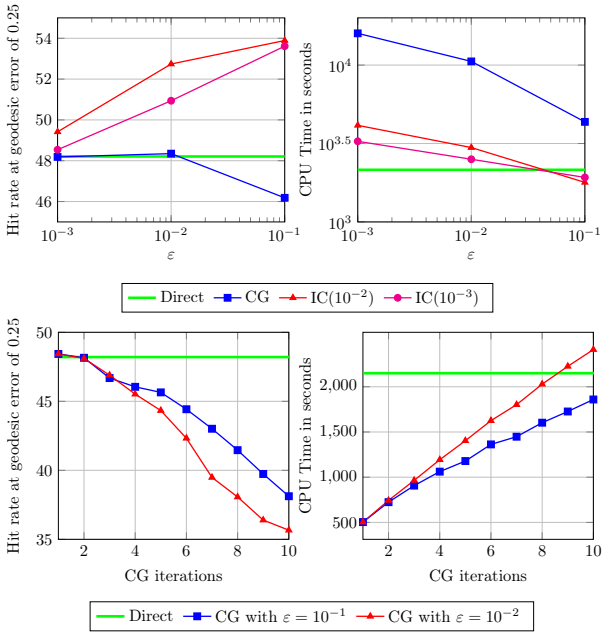


Fig. 7. Results on the dataset cat. We compare the geodesic error up to 0.25 (left) and the performance time (right) between the direct method and CG respectively IC(10^{-2}), IC(10^{-2}) for different ε (top). In addition, we present a comparison between the direct method and the CG method for the first few CG iterations l (bottom) for $\varepsilon = 10^{-1}$ and $\varepsilon = 10^{-2}$. Let us comment, that we observe here only the first part of the dint-effect, compare Figure 5, the hit rate increase of the dint will start with iteration $l = 14$.

2150s. Due to the large matrix size, we apply CG, and PCG with the incomplete Cholesky (IC) decomposition, as shown in Figure 7. For the latter decomposition often a numerical fill-in strategy IC(γ) is used, called *drop tolerance*, where the parameter $\gamma > 0$ describes a dropping criterion, cf. [18]. We have found by tests that $\gamma \in [10^{-2}, 10^{-3}]$ gives the best trade-off between accuracy and CPU time. Increasing ε leads naturally to faster computations but slightly worse results. Compared to the time performance of the direct method we achieve only a minor improvement. However, if we take a closer examination of the required iterations l, for CG and PCG with $\varepsilon = 10^{-1}$, it is conspicuous that PCG needed just about 1 iteration and CG on the other hand 20 iterations. The latter observation again inspires the idea to perform the CG method for a smaller number of iterations $l \leq 10$, which accordingly should be sufficient to gain acceptable results in fast CPU time. The results of CG for $\varepsilon = 10^{-1}, 10^{-2}$, and a number l of CG iterations is shown in Figure 7. Decreasing l reduces the time costs a lot, for instance one can save around 1500s for $l = 1$ compared to the direct method.

Application of MCR achieves the same solution behaviour as the wolf dataset, see Figure 8. Increasing the amount of used ordered modes leads to a significantly higher accuracy of the geodesic error and to more stable performance. Nevertheless, the computational costs of MCR grow exponentially (by increasing N_{max}) and a practicable value N_{max} is accordingly small. It is observable that the geodesic error for the range $N_{max} \in [20, 1000]$ is almost equal and oscillates just weakly. Even if the MCR technique for $N_{max} = 200$ looses around 35% accuracy (from 48 to 30) the simultaneous extremely short running time of around 100s is remarkable. Therefore, one may save around 95% of the computational time in relation to the direct method.

c) Experiment - Baby: Finally, we investigate the large dataset baby with a cloud point size of $N = 59727$ points. As before, we will study the shape matching correspondence in relation to the accuracy of the geodesic error and the computational effort between the direct method, the CG method (for $\varepsilon = 10^{-1}$ and $l \leq 10$) and the MCR technique.

The direct method requires around 10300s ($\approx$ 2 hours and 50 minutes) to solve 59727 linear systems on each shape for each time step. At this point it is recognizable, that large

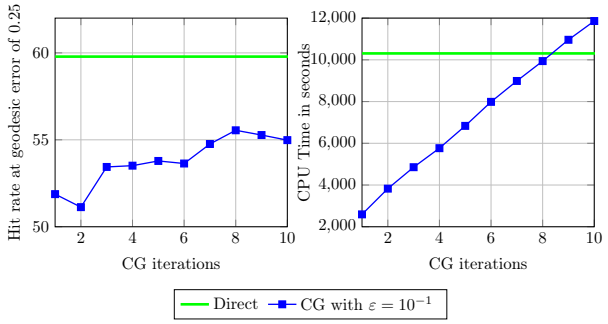


Fig. 9. Results on the dataset baby. We compare the geodesic error up to 0.25 (left) and the performance time (right) between the direct method and CG method for different number of iterations l and $\varepsilon = 10^{-1}$.

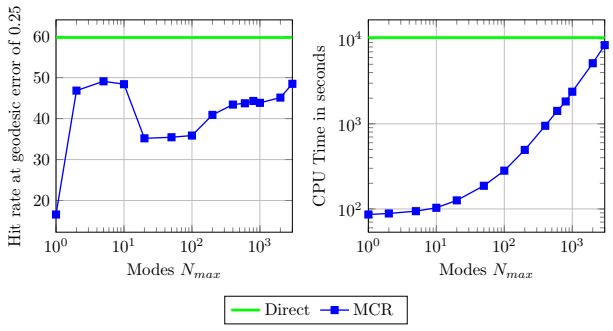


Fig. 10. Results on the dataset baby. We compare the geodesic error up to 0.25 (left) and the performance time (right) between the direct method and the MCR technique for different number of modes $N_{\max}$.

datasets produce extreme highly computational costs.

The evaluation of CG for $\varepsilon = 10^{-1}$ and $l \leq 10$ is shown in Figure 9. The percentage deviation of the accuracy of CG in relation to the direct method is approximately around 10%. In contrast, for $l = 1$, we can reduce the computational effort significantly to around 2600s, which can save more than 75% of the computational costs.

The MCR technique yields identical results as in the other two cases, compare Figure 10. The geodesic error oscillates in the range of $N_{max} = [1,20]$ and from $N_{max} = 20$ up to $N_{max} = 3000$ it converges against the solution of the direct method. Unfortunately, the convergence is very slow and a huge number of modes N_{max} is required. As before, for $N_{max} = 200$ the deviation of accuracy is around 35% (from 60 to 40), yet MCR needs around 500s. This means MCR reduce extremely the computational effort to 5%.

VI. Conclusion and Further Work

Experimental results confirm that the direct method, the CG method and the MCR technique are predestined for solving shape matching by time integration. Each of these methods has its own main advantage. The direct method is very accurate but can be inefficient. The CG method may reduce the computational costs to around 70%, whereby the percentage deviation of the accuracy in relation to the direct method is still less than 10%. The MCR technique is extremely fast and can save around 95% CPU time of the direct method, however it looses around 35% accuracy. Let us mention, that the underlying datasets are noise-free, therefore a further issue to examine is the influence of noisy data to the robustness of the numerical solvers.

The experiments show another interesting point of the MCR technique. It is remarkable that the results for a small spectrum are similar to the ones for a significantly larger spectrum. Unfortunately, the small spectrum suffers by a performance collapse at a low range of modes, roughly $N_{max} \in [1,20]$. Therefore, an interesting aspect would be to tune the small spectrum so that it becomes more stable. One may also consider other, more elaborated MOR techniques to possibly obtain a better trade-off between quality and computational effort.

References

[1] M. Bähr, M. Breuß, Y. Quéau, A. S. Boroujerdi, and J.-D. Durou, "Fast and accurate surface normal integration on non-rectangular domains," *CVM*, vol. 3, no. 2, pp. 107–129, 2017.

[2] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*. Springer, 2009.

[3] R. Dachsel, M. Breuß, and L. Hoeltgen, "Shape matching by time integration of partial differential equations," in *Proc. of SSVM*, 2017, pp. 669–680.

[4] T. A. Davis, "Algorithm 930: Factorize: An object-oriented linear system solver for matlab," *ACM Trans. Math. Softw.*, vol. 39, no. 4, pp. 28:1–28:18, July 2013.

[5] M. P. do Carmo, *Differential geometry of curves and surfaces*, 2nd ed. Dover Publications, 2016.

[6] S. Grewenig, J. Weickert, and A. Bruhn, "From box filtering to fast explicit diffusion," in *Proc. of DAGM*, 2010, pp. 533–542.

[7] D. Hafner, P. Ochs, J. Weickert, M. Reißel, and S. Grewenig, "Fsi schemes: Fast semi-iterative solvers for pdes and optimisation methods," in *Proc. of GCPR*, 2016, pp. 91–102.

[8] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *NIST*, vol. 6, no. 49, pp. 46–70, 1952.

[9] V. G. Kim, Y. Lipman, and T. A. Funkhouser, "Blended intrinsic maps," in *ACM (TOG)*, vol. 30, no. 4, 2011, pp. 1–12.

[10] G. Meurant, *Computer Solution of Large Linear Systems*. Elsevier Science, First Edition, 1999.

[11] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *In Visualization and Mathematics III*. Springer, 2002, pp. 35–57.

[12] S.-B. Nouri, "Advanced model-order reduction techniques for large-scale dynamical systems," Ph.D. dissertation, Department of Electronics, Carleton University, Canada, 2014.

[13] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental Mathematics*, vol. 2, no. 1, pp. 15–36, 1993.

[14] Z.-Q. Qu, *Model Order Reduction Techniques with Applications in Finite Element Analysis*. Springer, 2004.

[15] M. Reuter, F. E. Wolter, and N. Peinecke, "Laplace-Beltrami spectra as shape-DNA of surfaces and solids," *Computer-Aided Design*, vol. 38, no. 4, pp. 342–366, 2006.

[16] E. Rodolà, S. Rota Bulo, T. Windheuser, M. Vestner, and D. Cremers, "Dense non-rigid shape correspondence using random forests," in *Proc. of CVPR*, 2014, pp. 4177–4184.

[17] R. Rustamov, "Laplace-Beltrami eigenfunctions for deformation invariant shape representation," in *Symp. Geometry Processing*, 2007, pp. 225–233.

[18] Y. Saad, *Iterative Methods For Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2nd Edition, 2003.

[19] G. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Clarendon Press, Oxford Applied Mathematics and Computing Science Series, 1985.

[20] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

[21] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," in *Euro STAR*, 2010, pp. 61–82.

A Study of Spectral Expansion for Shape Correspondence

Robert Dachsel, Michael Breuß and Laurent Hoeltgen

Abstract— **The main task in three dimensional non-rigid shape correspondence is to retrieve similarities between two or more similar three dimensional objects. A useful way to tackle this problem is to construct a simplified shape representation, called feature descriptor, which is invariant under deformable transformations. A successful class of such feature descriptors is based on physical phenomena, concretely by the heat equation for the heat kernel signature and the Schrödinger equation for the wave kernel signature. Both approaches employ the spectral decomposition of the Laplace-Beltrami operator, meaning that solutions of the corresponding equations are expressed by a series expansion in terms of eigenfunctions. The feature descriptor is then computed at hand of those solutions. In this paper we explore the influence of the amount of used eigenfunctions on shape correspondence applications, as this is a crucial point with respect to accuracy and overall computational efficiency of the method. Our experimental study will be performed at hand of a standard shape data set.**

I. INTRODUCTION

In many tasks, it is useful to describe a three dimensional geometric object by its bounding surface, often referred as shape. Thereby, the investigation of shape correspondence is a fundamental operation in visual computing, with many potential fields of applications including medical imaging, geometric modeling and digital heritage [8], [10], [12]. For a general shape correspondence scenario there are two or more shapes given, and it is the task to find a reasonable relation/pairing between them. In the context of rigid shape correspondence, shapes may be considered similar if there exists a rigid transformation between them. Since those transformations can be represented compactly as a rotation, translation and reflection, many solution techniques are well established such as Iterative Closest Point methods, cf. [3], [6] among others. A more challenging yet oftentimes more realistic setting is that of non-rigid shape correspondence, where the shapes are able to undergo almost isometric transformations, leading to a large variety of possible deformations such as poses of human or animal bodies.

An important solution strategy to obtain pointwise shape correspondence is to employ a feature descriptor computed over each shape, and to attempt to match the values of the feature descriptor. The task of the feature descriptor is to characterize each element on the shape regarding its geometric properties. Ideally, this feature descriptor is invariant under deformable transformations, which is challenging to achieve in its construction.

Related Work. In this paper we consider a modern class of feature descriptors that can handle almost isometric transformations, namely the so-called *spectral methods* that are based on the spectral decomposition of the Laplace-Beltrami operator. In the framework of shape analysis these spectral methods were first proposed in [11]. Based on developments in [16], the *heat kernel signature (HKS)* has been introduced [17]. It assigns each point on an object surface a unique signature based on the fundamental solution of the geometric heat equation. The latter is a partial differential equation (PDE) that contains in its spatial part the Laplace-Beltrami operator and describes the time evolution of heat on an objects' surface. Later, a scale invariant extension of this approach was developed in [5]. In [1] another feature descriptor inspired by the Schrödinger equation was proposed. This feature descriptor is called the *wave kernel signature (WKS)* and represents the average probability of measuring a quantum mechanical particle at a specific location. For both descriptors, the spectral decomposition of the incorporated Laplace-Beltrami operator leads to a series expression of its eigenfunctions and eigenvalues. The contributions in such a series are ordered in the sense that especially the first terms contain the low frequency components describing the macroscopic (global) properties of a shape. Thus, taking into account a corresponding part of the spectral components yields a feature descriptor robust to local errors such as (high frequent) noise but vulnerable to global distortions such as e.g. changes in shape topology.

Our Contributions. In this paper we report on our ongoing investigation of the number of eigenfunctions employed for constructing the HKS and WKS, evaluated with respect to the shape correspondence task. We are not aware of a thorough study of this aspect in the previous literature. In many publications the number of eigenfunctions is set to a fixed value (e.g. first 300 eigenfunctions) without further explanations, representing a very defensive, heuristic choice by our computational experience. With our paper we make an attempt to fill this gap in the current literature. Furthermore, we especially evaluate the HKS and WKS using a much smaller amount of eigenfunctions than usually employed, leading to some interesting conclusions and potential new challenges.

II. THEORETICAL BACKGROUND

In this section we introduce the basic facts that are necessary to define the shape correspondence framework.

A. Almost Isometric Shapes

The bounding surface of a three dimensional geometric object is a two dimensional curved object, embedded into the three dimensional Euclidean space. It is natural to model

Institute of Mathematics, Brandenburg Technical University, Platz der Deutschen Einheit 1, 03046, Cottbus, Germany {dachsel,breuss,hoeltgen}@b-tu.de

shapes as compact two dimensional Riemannian manifolds $\mathscr{M}$, equipped with metric tensor $g \in \mathbb{R}^{2\times 2}$. In this setting, two shapes $\mathscr{M}$ and $\tilde{\mathscr{M}}$ may be considered as *isometric* if there is a mapping T that unfolds one surface onto the other, thereby preserving the intrinsic distances of the unfolded surface. From the mathematical point of view there exists a smooth homeomorphism $T : \mathscr{M} \to \tilde{\mathscr{M}}$ with

$$d_{\mathscr{M}}(x_i,x_j) = d_{\tilde{\mathscr{M}}}(T(x_i),T(x_j)) \quad \forall x_i,x_j \in \mathscr{M}, \tag{1}$$

where $d_{\mathscr{M}}(x_i,x_j)$ is the intrinsic distance taken on manifold $\mathscr{M}$. The intrinsic distance can be thought as the shortest curve along the surface $\mathscr{M}$ connecting x_i and x_j.

As indicated, the notation of isometric shapes appears too restrictive for many applications [4]. Many shapes surrounding us in the world appear with additional small elastic deformations. These occur by either the transformation itself (e.g. as in elastic bending), by geometric noise in datasets, or by transferring a continuously described shape into a numerical framework which may act as a small distortion on the intrinsic distance. We call two shapes $\mathscr{M}$ and $\tilde{\mathscr{M}}$ *almost isometric*, if there exists a transformation $S : \mathscr{M} \to \tilde{\mathscr{M}}$ with

$$d_{\mathscr{M}}(x_i,x_j) \approx d_{\tilde{\mathscr{M}}}(S(x_i),S(x_j)) \quad \forall x_i,x_j \in \mathscr{M}. \tag{2}$$

B. Heat and Schrödinger Equation on Shapes

The common property of the considered PDEs is that the incorporated spatial derivatives have the format of the Laplace-Beltrami operator. Note that the latter is identical to the standard Laplace operator when considering the PDEs in the Euclidean plane, the Laplace-Beltrami operator just takes additionally into account the geometric properties of curvature of a manifold in 3D by making use of tools from differential geometry [7]. Consequently, the Laplace-Beltrami operator is defined on a smooth manifold $\mathscr{M}$.

Turning to the mathematical formulation of the *Laplace-Beltrami operator*, let us recall that for a given parameterisation of a two dimensional manifold, we can express it in local coordinates:

$$\Delta_{\mathscr{M}} u = \frac{1}{\sqrt{|g|}} \sum_{i,j=1}^{2} \partial_i \left(\sqrt{|g|} g^{ij} \partial_j u \right), \tag{3}$$

where $u : \mathscr{M} \to \mathbb{R}$ is a scalar function, g^{ij} are the components of the inverse of the metric tensor and $|g|$ is its determinant.

The *geometric heat equation* describes how heat would propagate along a surface $\mathscr{M}$. The corresponding initial-boundary value problem can be formulated as

$$\begin{cases} \partial_t u(x,t) = \Delta_{\mathscr{M}} u(x,t) & x \in \mathscr{M}, t \in \mathbb{R}^+ \\ u(x,0) = u_0 & \\ \langle \nabla_{\mathscr{M}} u, n \rangle = 0 & x \in \partial\mathscr{M} \end{cases} \tag{4}$$

where u_0 is a given heat distribution. Many shapes appear as closed Riemannian manifold with $\partial\mathscr{M} = \emptyset$, where boundary conditions do not appear. For the case $\mathscr{M}$ has boundaries, we additionally require u to satisfy the homogeneous Neumann boundary conditions, where n is the normal vector to the boundary. In this context the inner product $\langle .,. \rangle$ lives in the tangent space. This choice conserves the amount of heat $\|u\|^2_{L_2(\mathscr{M})} = \text{const}\ \forall t \in \mathbb{R}^+$.

The free, time-dependent *Schrödinger equation*

$$\begin{cases} i\partial_t u(x,t) = \Delta_{\mathscr{M}} u(x,t) & x \in \mathscr{M}, t \in \mathbb{R}_+ \\ u(x,0) = u_0 & \\ \langle \nabla_{\mathscr{M}} u, n \rangle = 0 & x \in \partial\mathscr{M} \end{cases}, \tag{5}$$

where i is the imaginary unit, allows to study how a free and massive quantum particle would move on the surface $\mathscr{M}$. In quantum mechanics, the dynamics of a particle is described by its complex wave function $u(x,t)$ and its probability amplitude, whose square norm $\|u\|^2_{L_2(\mathscr{M})}$ is equal to the probability density for finding the particle at a specific position for a fixed t. In this context u_0 has the interpretation of an initial wave package.

Separation of Variables. First, we assume that the solution will take the form $u(x,t) = \phi(x)T(t)$ due to the fact that we are working with linear and homogeneous partial differential equations. This approach works because if the product of two functions ϕ and T of independent variables x and t is a constant, each function must separately be a constant. At the end, we are able to separate the equations to get a function of only t on one side and a function of only x on the other side

$$\kappa \frac{\partial_t T(t)}{T(t)} = \frac{\Delta_{\mathscr{M}} \phi(x)}{\phi(x)} = \text{const} = -\lambda, \tag{6}$$

where κ summarizes both equations ($\kappa = 1$ for heat equation, $\kappa = i$ for Schrödinger equation), and $-\lambda$ is called the separation constant which is arbitrary for the moment. This leaves us with two new equations, namely an ordinary differential equation for the temporal component

$$\partial_t T(t) = -\kappa\lambda T(t) \quad t \in \mathbb{R}_+ \tag{7}$$

and the spatial part takes the form of the Helmholtz equation

$$\begin{cases} \Delta_{\mathscr{M}} \phi(x) = -\lambda \phi(x) & x \in \mathscr{M} \\ \langle \nabla_{\mathscr{M}} \phi, n \rangle = 0 & x \in \partial\mathscr{M} \end{cases}, \tag{8}$$

where the value of the constant λ has the meaning of the operator's eigenvalue.

The Spatial Part. The Laplace-Beltrami operator is a self-adjoint operator on the space $L_2(\mathscr{M})$ (since we assumed the shapes to be compact). This implies that the *Helmholtz equation*

$$\begin{cases} \Delta_{\mathscr{M}} \phi_k(x) = -\lambda_k \phi_k(x) & x \in \mathscr{M} \\ \langle \nabla_{\mathscr{M}} \phi_k, n \rangle = 0 & x \in \partial\mathscr{M} \end{cases} \quad k \in \{1,\dots\infty\} \tag{9}$$

has infinite non-trivial solutions for certain (discrete) values called eigenvalues, and corresponding eigenfunctions, which is a result of the spectral theorem. The ordered spectrum of eigenvalues $0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_\infty$ and corresponding eigenfunctions $\phi_1, \phi_2, \ldots, \phi_\infty$ form an orthonormal basis of $L_2(\mathscr{M})$, and constant functions are solutions of the Helmholtz equation with eigenvalue 1 (only for Neumann boundary conditions or no shape boundaries).

It is well known that the eigenfunctions are a natural generalization of the classical Fourier basis to functions on shapes. Let us note that the physical interpretation of the Helmholtz equation is the following. The shape of a 3D object can be thought of as a vibrating membrane, the ϕ_k can be interpreted as its vibration modes whereas the λ_k have the meaning of the corresponding vibration frequencies, sorted from low to high frequencies, as shown in Figure 4

The Temporal Part. For each $k \in \{0, \dots \infty\}$ there is an ordinary differential equation (7) left, corresponding to $\lambda \equiv \lambda_k$, which can be solved by integrating both sides using the indefinite integral. This leads to

$$\int \frac{1}{T(t)} \mathrm{d}T(t) = -\kappa\lambda_k \int \mathrm{d}t \quad \Rightarrow \quad T(t) = \alpha_k e^{-\kappa\lambda_k t}, \quad (10)$$

where the integration constant α_k should satisfy the initial condition of the k^{th} eigenfunction. The final product solution then reads as

$$u_k(x,t) = \alpha_k e^{-\kappa\lambda_k t} \phi_k(x). \quad (11)$$

The principle of superposition says that if we have several solutions to a linear homogeneous differential equation then their sum is also a solution. Therefore, a closed-form solution of the heat equation in terms of a series experession can be written as

$$u(x,t) = \sum_{k=1}^{\infty} \alpha_k e^{-\lambda_k t} \phi_k(x), \quad (12)$$

and the solution of the Schrödinger equation reads as

$$u(x,t) = \sum_{k=1}^{\infty} \alpha_k e^{-i\lambda_k t} \phi_k(x), \quad (13)$$

where the coefficient α_k fulfill the initial condition.

Heat Kernel Signature. The coefficients α_k in our expansion can be computed by using the L^2 inner product

$$\alpha_k = \langle u_0, \phi_k \rangle_{L^2(\mathscr{M})} = \int_{\mathscr{M}} u_0(y)\phi_k(y)\,\mathrm{d}y \quad (14)$$

such that we have

$$u(x,t) = \sum_{k=1}^{\infty} \left(\int_{\mathscr{M}} u_0(y)\phi_k(y)\,\mathrm{d}y \right) e^{-\lambda_k t}\phi_k(x) \quad (15)$$

$$= \int_{\mathscr{M}} u_0(y) \left(\sum_{k=1}^{\infty} e^{-\lambda_k t}\phi_k(y)\phi_k(x) \right) \mathrm{d}y. \quad (16)$$

The term inside the brackets is called the *heat kernel* $K(x,y,t)$, and it describes the amount of heat transmitted from x to y after time t. By setting the initial condition to be a delta heat distribution at the position y with $u_0(y) = \delta_x(y)$, we thus obtain after [17] the *heat kernel signature*

$$\mathrm{HKS}(x,t) = \sum_{k=1}^{\infty} e^{-\lambda_k t} \left|\phi_k(x)\right|^2, \quad (17)$$

where the shifting property of the delta distribution $f(x) = \int_{\mathscr{M}} f(y)\delta_x(y)\,\mathrm{d}y$ was used. The quantity $\mathrm{HKS}(x,t)$ describes the amount of heat present at point x at time t.

Wave Kernel Signature. The *wave kernel signature* [1] is defined to be the time-averaged probability of detecting a particle of a certain energy distribution at the point x, formulated as

$$\mathrm{WKS}(x,t) = \lim_{T\to\infty} \frac{1}{T} \int_0^T |u|^2 \mathrm{d}t = \sum_{k=1}^{\infty} \left|\alpha_k\right|^2 \left|\phi_k(x)\right|^2. \quad (18)$$

Furthermore, $\alpha_k = \alpha(e_k)$ becomes a function of the energy distribution e_k of the quantum mechanical particle and can be chosen as a log-normal distribution i.e.

$$\left|\alpha_k\right|^2 = \exp\left(\frac{-(e - \log\lambda_k)^2}{2\sigma^2} \right), \quad (19)$$

where the variance of the energy distribution is denoted by σ, see again [1] for more details.

C. Discretization Aspects

While we have described now the analytical setting, we have to translate the analytical set-up into a discrete format allowing to deal with shape data.

Discrete Surfaces. A suitable surface representation is given by a triangular mesh. In more detail, a triangulated surface is given by the tuple $\mathscr{M}_d = (P,T)$. The point cloud $P := \{x_1, \dots, x_N\}$ contains the finite number of vertices (given as coordinate points in $\mathbb{R}^3$) a shape consists of. The entire mesh can be formed by connecting the vertices x_i so that one obtains two-dimensional linear triangles. Therefore, the set of linear triangles T contains the neighborhood relations between vertices forming a triangle. Further, we sub-divide the meshed surface as follows:

$$\mathscr{M}_d = \bigcup_{i=1}^{N} \Omega_i \quad \text{and} \quad \Omega = \mathrm{diag}\,(\Omega_1, \dots, \Omega_N) \in \mathbb{R}^{N\times N} \quad (20)$$

where Ω_i is the barycentric cell volume, surrounding the vertex x_i, as shown in Figure 1.

Discrete Laplace-Beltrami Operator. Many schemes have been proposed to estimate the Laplace-Beltrami operator for a triangular meshed surface [2], [14], [15]. A commonly used method is the cotangent weight scheme introduced in [13]. As a result, for a function defined on a triangular mesh the discrete Laplace-Beltrami operator $L \in \mathbb{R}^{N\times N}$ reduces to the following simple sparse matrix representation

$$L_{ij} = \begin{cases} -\sum\limits_{j\in N_i} w_{ij} & \text{if } x_i = x_j \\ w_{ij} & \text{if } x_i \neq x_j \text{ and } x_j \in N_i \\ 0 & \text{else} \end{cases}, \quad (21)$$

where N_i denotes the set of points adjacent to x_i. As shown in Figure 1, the weights w_{ij} of the edge (x_i, x_j) can be classified into interior E_i and boundary edges E_b, respectively,

$$w_{ij} = \begin{cases} \frac{\cot\alpha_{ij} + \cot\beta_{ij}}{2} & \text{if } (x_i,x_j) \in E_i \\ \frac{\cot\alpha_{ij}}{2} & \text{if } (x_i,x_j) \in E_b \end{cases}. \quad (22)$$

as also shown in Figure 1. Furthermore, α_{ij} and β_{ij} denote the two angles opposite to the edge (x_i, x_j), for details we

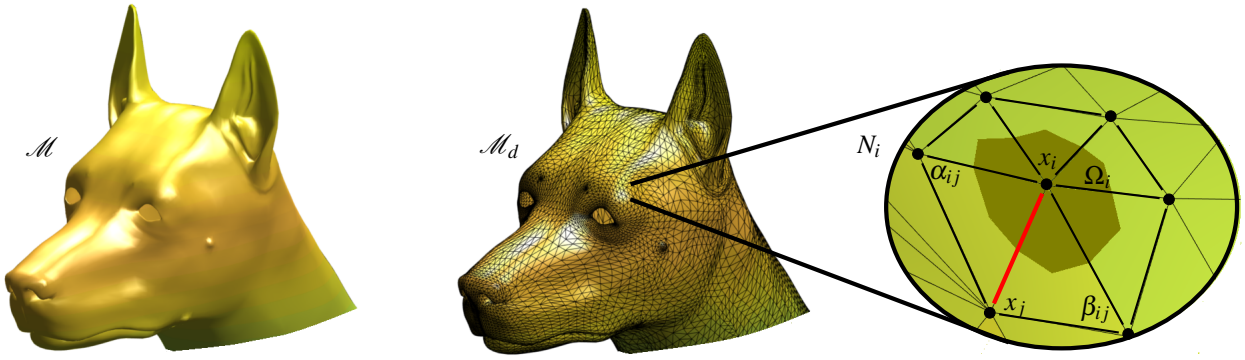


Fig. 1. Continuous and discrete representation of a shape. **Left:** In the continuous setting a shape is modeled by a two dimensional Riemannian manifold $\mathscr{M}$. **Middle:** Its discrete approximation $\mathscr{M}_d$ is given by a point cloud P, where the points are connected by non-uniform and linear triangles T. **Right:** For the construction of the discrete Laplace-Beltrami operator at the point x_i, it requires the set of adjacent points N_i, the surrounding cell volume Ω_i and the two angles α_{ij} and β_{ij} opposite to the edge (x_i, x_j).

refer to the mentioned source. The eigenfunctions and eigenvalues of the discrete Laplacian are computed by performing the generalized eigen-decomposition

$$L\phi_k = -\lambda_k \Omega \phi_k\,, \quad k \in \{1,\ldots,N\}\,, \tag{23}$$

However, an important practical aspect of the eigen-decomposition is that computing a full spectrum is very time and memory consuming (e.g. all eigenfunctions have to be stored in a dense $\mathbb{R}^{N\times N}$ matrix). Therefore, only the first $\tilde{N} \ll N$ of the eigenvalues in increasing order and corresponding eigenfunctions are computed. The eigenvalue $\lambda_1 = 0$ belongs to the constant eigenfunction ϕ_1, containing no information useful for shape correspondence.

The discrete HKS and WKS read for a given point x_i finally

$$\mathrm{HKS}(x_i,t) = \sum_{k=2}^{\tilde{N}} e^{-\lambda_k t} \left|\phi_k(x_i)\right|^2 \quad \text{and} \tag{24}$$

$$\mathrm{WKS}(x_i,t) = \sum_{k=2}^{\tilde{N}} \left|\alpha_k\right|^2 \left|\phi_k(x_i)\right|^2 . \tag{25}$$

Discrete Time and Energy Scale. For the HKS, the time axis is sampled at 100 samples, $t_1,\ldots,t_{100}$, where the time is logarithmically scaled over the time interval with $t_1 = 4\ln 10/\lambda_{\tilde{N}}$ and $t_{100} = 4\ln 10/\lambda_2$. The energy scale of the WKS becomes $e_1 = \log(\lambda_2) + 2\sigma$ and $e_{100} = \log(\lambda_{\tilde{N}}) - 2\sigma$, and the parameters were set as described in [1].

III. THE CORRESPONDENCE PROBLEM

For two points $x_i \in \mathscr{M}_d$ and $\tilde{x}_i \in \tilde{\mathscr{M}}_d$ the condition for a point-wise correspondence can be written as a minimisation problem:

$$(x_i,\tilde{x}_j) \;\Leftrightarrow f(\tilde{x}_j) = \min_{k=1,\ldots,\tilde{N}} \{d_{FD}(x_i,\tilde{x}_k)\}\,. \tag{26}$$

where $d_{\mathrm{FD}}(x_i,\tilde{x}_j)$ is the *feature distance*.

For the HKS the squared distance is measured by computing a normalised L_2-norm

$$d_{\mathrm{HKS}}(x_i,\tilde{x}_j) = \left\{ \int_{t_1}^{t_{100}} \left(\frac{|\mathrm{HKS}(x_i,t) - \mathrm{HKS}(\tilde{x}_j,t)|}{\int_{\mathscr{M}} \mathrm{HKS}(x,t)\mathrm{d}x} \right)^2 \mathrm{d}\log t \right\}^{\frac{1}{2}} \tag{27}$$

and the WKS uses a distance based on the L_1-norm of the normalised signature difference

$$d_{\mathrm{WKS}}(x_i,\tilde{x}_j) = \int_{e_1}^{e_{100}} \left| \frac{\mathrm{WKS}(x_i,e) - \mathrm{WKS}(\tilde{x}_j,e)}{\mathrm{WKS}(x_i,e) + \mathrm{WKS}(\tilde{x}_j,e)} \right| \mathrm{d}e. \tag{28}$$

IV. EXPERIMENTAL RESULTS

A. Evaluation Measure and Dataset

In general, we perform a dense point-to-point correspondence, involving all vertices the shapes are made off. We investigate the quality of the established correspondences at hand of several quality measures that are standard in the abovementioned literature.

The tests we discuss here are chosen so that they are in some sense generic, as they are not tuned to a specific class of shapes with some special property. Our aim is to study a typical setting for the shape correspondence problem in detail and to get an insight into typical phenomena occuring when approaching the task. The dog shapes appear to be suitable for this proceeding, as they show a reasonable range of mesh widths and transformations, but not topological changes (representing in shape matching a hard problem to resolve by itself which is beyond the scope of this paper) that may especially hinder to find correspondences by the first few eigenfunctions.

We concentrate in this paper on discussing our current findings in analyzing mutual influences of eigenfunctions, eigenvalues (i.e. ordering of eigenfunctions), and matching accuracy of spectral methods for shape correspondence. In detail, the experiments are evaluated as follows.

Hit Rate. The percentual *hit rate* is defined as $TP/(TP+FP)$, where TP and FP are the number of true positives and false positives, respectively.

Fig. 2. The dog class of the TOSCA data set. The transformed shapes are almost isometric modifications of the reference shape (left).

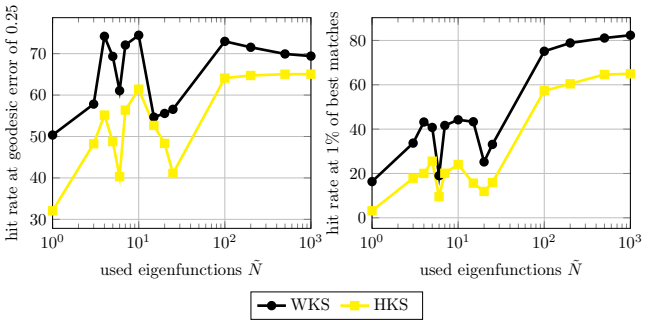


Fig. 3. Here we compared both the HKS and the WKS for finding correspondences within a geodesic error of 0.25 (left) and correct correspondences at 1% of best matches (right) as a function of the used eigenfunctions. For large $\tilde{N}$, the performance goes into saturation. Small $\tilde{N}$ leads roughly to a similar quality in terms of an acceptable geodesic errors but unstable performance.

Cumulative Match Characteristic (CMC). The CMC curve evaluates the hit rate for finding true corresponding pairs within the first 1% of best matches. Thereby the best matches are those with the smallest feature distance, arranged in increasing order.

The Geodesic Error. For the evaluation of the correspondence quality, we followed the Princeton benchmark protocol [9]. This procedure evaluates the precision of the computed matching x_i by determining how far are those away from the actual ground-truth correspondence x^*. Therefore a normalised intrinsic distance $d_{\mathscr{M}}(x_i,x^*)/\sqrt{A_{\mathscr{M}}}$ on the transformed shape is introduced. Finally, we accept a matching to be true if the normalised intrinsic distances are smaller than the threshold 0.25.

Dataset. For the experiment nine dog shapes are used, taken from the TOSCA data set [4], available in the public domain as shown in Figure 2.

B. Influence of the Amount of Eigenfunctions to Shape Correspondence

For the first experiment we increase the number of used ordered eigenfunctions, starting from $\tilde{N} = 3$ and end up to $\tilde{N} = 1000$, and study the quality of finding correspondences. Note that we average in this experiment over all correspondence computations performed over the given data set, see Figure 2.

The evaluation in Figure 3 shows how the matching precision of the HKS and WKS depends on the number of used eigenfunctions. After taking into account about hundred eigenfunctions the performance for both descriptors goes into saturation, i.e. it does not increase significantly anymore. Considering especially the comparison of correct correspondences within the geodesic threshold, displayed in Figure 3 (left), it is remarkable that the results for a very small spectrum $\tilde{N} \approx 10$ are similar to the ones for the large spectrum $\tilde{N} \approx 1000$. Especially the WKS descriptor gains already reasonable results in the regime of the small spectrum.

We also see, however, that within the small spectrum the correspondence quality suffers by large variations including a performance collapse of $15\% - 20\%$ at a specific range of eigenfunctions, $\tilde{N} \approx 3$ and $\tilde{N} \approx 20$ for both descriptors. On the other hand, the performance is stabilized and stays stable for a larger amount of eigenfunctions.

It would surely be interesting to attempt to tune the small spectrum in such a way that it becomes more stable, e.g. by removing eigenfunctions that result in a performance loss. In this paper, such considerations motivate us to explore the phenomena potentially encountered when using a small spectrum.

C. Shape Correspondence Using a Small Spectrum

The test just discussed above shows that it may be worth the effort to inspect in more detail the situation of the small spectrum of eigenfunctions. To this end we will also employ a finer sampling of the error behaviour for the small range of eigenfunctions as in the first test.

First of all we study the eigenfunctions itself for the reference dog shape and three arbitrarily selected almost isometric counterparts from our test data set. We pay attention to the small spectrum where $\tilde{N}$ is ranging from 3 to 25. By comparing the eigenfunctions on the reference shape and transformed shape, they should be similar since eigenfunctions belonging to low frequencies are stable under almost isometric transformations. However, as visualized in Figure 4 for selected examples of eigenfunctions, not all appear to be similar. In order to make this impression quantitative, we define an averaged error with respect to the reference shape (dog0),

$$e(\phi_i,\tilde{\phi}_i) = \frac{1}{N}\sum_{k=1}^{N} \left| |\phi_i(x_k)| - |\tilde{\phi}_i(x_k)| \right| , \qquad (29)$$

where ϕ_i and $\tilde{\phi}_i$ are the i^{th} eigenfunctions on the reference and transformed shape, respectively. Then, we compare the matching performance of the HKS and WKS for the selected shapes as a function of eigenfunctions, considered at the small spectrum, as shown in Figure 5.

The results show that there is a surprisingly large error when comparing the first few eigenfunctions of the Laplace-Beltrami operator, as seen in Figure 5. For interpretation one may consider the analogy to approximate a given signal by the first few terms of a Fourier series; it seems that the dog shape incorporates in the low frequency range a few frequencies that are not highly significant and therefore not captured equally well by shape variations. We conjecture that

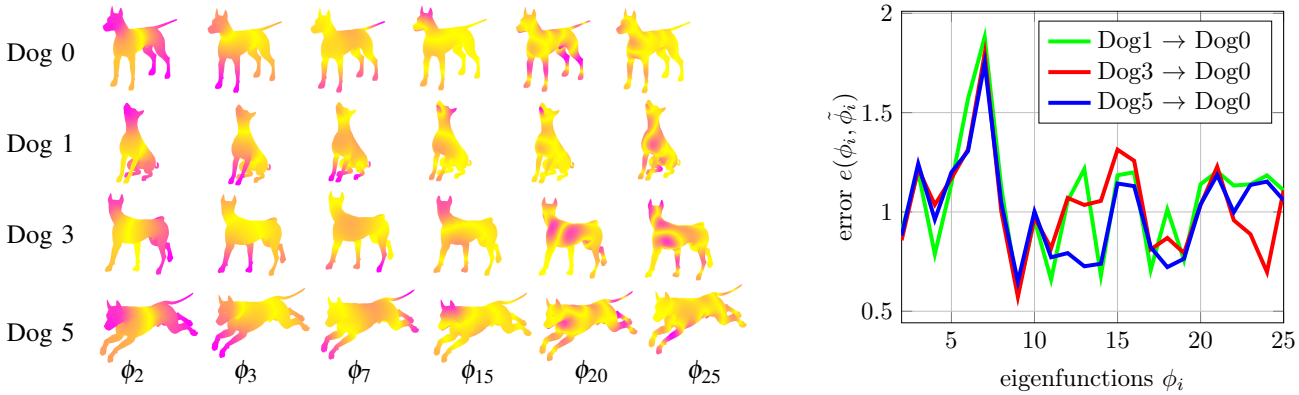


Fig. 4. Comparing selected eigenfunctions on the dog and their error under almost isometric transformations. **Left:** The absolute value of the eigenfunction of the Laplace-Beltrami operator computed on examples of the dog dataset. The colors represent the values of the eigenfunctions, pink being the most positive and yellow are almost zero values. **Right:** Error of the eigenfunctions of the reference shape (dog 0) and the almost isometric counterparts (dog 1, dog 3, dog 5). The smaller the error, the more stable are the eigenfunctions under almost isometric transformations.

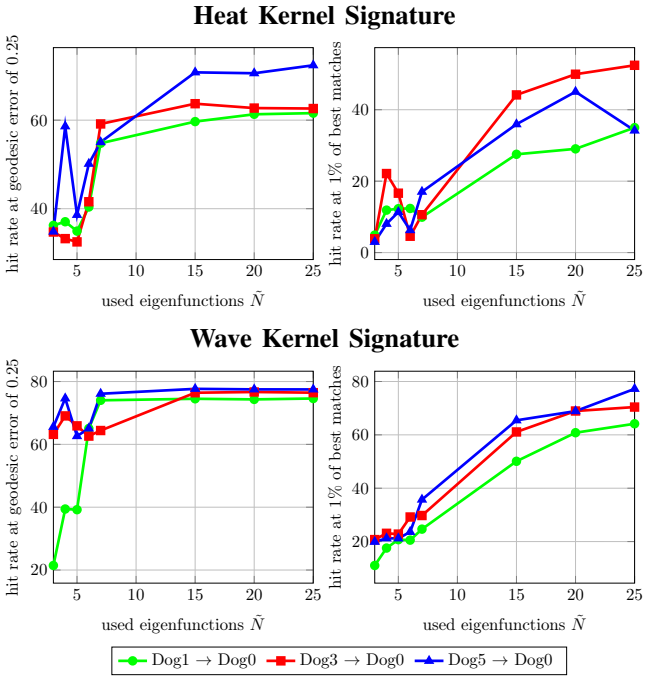


Fig. 5. The evaluation of shape correspondence for the selected dog shapes using the HKS and WKS. The performance is plotted as a function of the used small spectrum.

this is a phenomenon that can be found in a similar way for some low frequencies in other shapes. The phenomenon gets stabilized after taking into account a few more eigenfunctions here. Let us stress that this interesting aspect is not visible in tests as performed usually in the literature where error averages over large data sets with many different shapes are computed.

Secondly, we also observe a correlation between the error in matching eigenfunctions of the Laplace-Beltrami operator and the shape matching performance with HKS and WKS when comparing Figure 5 to results given in Figure 3. After the discussion above, it is evident that this is mainly observable in the first few eigenfunctions, yet the WKS may still achieve in some cases high accuracy in terms of admissible geodesic error. One may also infer that the WKS is often more robust against differences in eigenfunctions at low frequencies than the HKS. This appears to be physically intuitive since solutions of the Schrödinger equation bear a more complex wave interaction pattern (arguably the main point leading to high accuracy) than the smoothly varying solutions of the heat equation (where consequently the low frequencies carry most information).

One may also conjecture that the occurence of a very low error as for $i = 9$ seems to have a significant stabilizing effect, especially concerning best matches. This effect may also be present in the Dog3→Dog0 experiment in the range of 20 to 25 eigenfunctions. Comparing the red lines in Figure 5 and HKS best matches in Figure 3 (top right), we see that the hit rates in the other two experiments deteriorate as they do not benefit from the conjectured mechanism.

V. CONCLUSION AND FURTHER WORK

The computation of shape correspondences is a computationally intensive task. Therefore it would be highly desirable to develop a spectral method relying only on the first few terms of the corresponding series expansion. We have pointed out some aspects of approximations using such small spectra at hand of a typical shape correspondence experiment. We think that our discussion illuminates some points that can be important for the construction of such a method.

In the future we plan to pursue this open issue. For this it will be imperative in a first step to perform more experiments with other shapes, and to validate the aspects we found in this paper also at hand of simple, specifically constructed shapes.

REFERENCES

[1] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: a quantum mechanical approach to shape analysis," *IEEE Computer Vision Workshops (ICCV Workshops)*, pp. 1626–1633, 2011.
[2] M. Belkin, J. Sun, and Y. Wang, "Discrete Laplace operator on meshed surfaces," in *Proceedings of SOCG*, 2008, pp. 278–287.
[3] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," in *IEEE Trans. Pattern Anal. Mach. Intell. 14*, 1992, pp. 239–256.

[4] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*. Springer, 2009.
[5] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. International Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1704–1711.
[6] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," in *Image Vision Comput. 10*, 1992, pp. 145–155.
[7] M. P. Do Carmo, *Differential geometry of curves and surfaces: revised and updated second edition*. Dover Puplications, 2016.
[8] M. Kilian, N. J. Mitra, and H. Pottmann, "Geometric modeling in shape space," in *In Proceedings of SIGGRAPH*, vol. 26, no. 3, 2007.
[9] V. G. Kim, Y. Lipman, and T. A. Funkhouser, "Blended intrinsic maps," in *ACM Transactionson Graphics*, vol. 30, no. 4, 2011.
[10] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, and J. F. D. Shade, "The digital michelangelo project: 3d scanning of large statues." in *In Proceedings of SIGGRAPH*, 2000, pp. 131–144.
[11] B. Lévy, "Laplace-Beltrami eigenfunctions towards an algorithm that 'understands' geometry." in *Int. Conf. on Shape Modeling and Applications*, 2006.
[12] J. Maintz and V. M., "A survey of medical image registration," in *Medical Image Analysis 2*, 1998, pp. 1–36.
[13] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *In Visualization and Mathematics III*. Springer, 2002, pp. 35–57.
[14] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Experimental Mathematics*, vol. 2, no. 1, pp. 15–36, 1993.
[15] M. Reuter, F. E. Wolter, and N. Peinecke, "Laplace-Beltrami spectra as shape-DNA of surfaces and solids," *Computer-Aided Design*, vol. 38, no. 4, pp. 342–366, 2006.
[16] R. Rustamov, "Laplace-Beltrami eigenfunctions for deformation invariant shape representation," in *Symp. Geometry Processing*, 2007, pp. 225–233.
[17] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383–1392, 2009.

Image texture classification with morphological amoeba descriptors

Franz Schwanninger[1] and Martin Welk[2]

Abstract— **We investigate the applicability of quantitative texture descriptors based on morphological amoebas in the context of a machine learning approach to texture classification. Morphological amoebas are a type of contrast-adaptive structuring elements originally designed for adaptive morphological image filters, and they stand in a close relation to local edge-weighted pixel graphs of an image. A recently introduced class of texture descriptors is obtained by applying graph indices from quantitative graph theory to those pixel graphs. Additionally we consider descriptors that refer to the geometric shape of the amoebas. In our approach, these descriptors are histogram encoded and fed into a linear support vector machine (SVM). We demonstrate our approach using a small number of texture samples from the VisTex database as training data. In further experiments, we study how selected parameters of the amoeba construction influence the classification performance.**

I. INTRODUCTION

In this paper we consider texture-based image classification. Textures are intrinsic structures of image regions and can be classified into different categories. Machine learning as well as other approaches that aim at classifying images depending on their textural content often make use of quantitative texture descriptors [10]. In this paper, we focus on descriptors that are constructed from *morphological amoebas.* Originally introduced as structuring elements in adaptive mathematical morphology [15], morphological amoebas are of interest for texture analysis as they encode local image structure [22]. Their construction is inherently related with subgraphs of the edge-weighted pixel graph of an image in which edge weights are computed from image contrast. Using these subgraphs as input for the computation of *graph indices* gives rise to a class of graph-based texture descriptors [22] from which part of the descriptors in this paper are chosen. To these, we add descriptors obtained by evaluating geometric information of the amoebas. The descriptors are then encoded and forwarded to Support Vector Machines (SVMs) [3] where they are aggregated in order to enhance the classification performance. This scalable approach is demonstrated by two graph-index based texture descriptors and one texture descriptor encoding geometric amoeba properties.

*This work was not supported by any organization

[1]The work presented here is part of the master thesis of Franz Schwanninger at the Department of Biomedical Informatics and Mechatronics, Private University for Health Sciences, Medical Informatics and Technology (UMIT), 6060 Hall/Tyrol, Austria, franz.schwanninger@edu.umit.at

[2]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

II. RELATED WORK

Image texture analysis has been investigated for a long time. Haralick [9] shows an early overview of structural and statistical approaches; others show frequency-based models [14], filter banks [16], or fractals [20].

Morphological amoebas as spatially adaptive neighborhoods have been introduced by Lerallut et al. [15]. Our short recap on the amoeba construction follows the presentation in [23] which is already adapted to the combination with graph indices in order to construct texture descriptors.

Graph indices have their roots in the analysis of molecular graphs [25], [1], [19] but have meanwhile developed into an important tool for a broad range of network analysis tasks [5]. Although graph models have been widely used in image analysis, see [17], methods originating from quantitative graph theory – like graph indices – have not played a significant role in texture analysis so far. By the construction of texture descriptors from graph indices and amoebas in [22], a first step into this direction has been made. The applicability of these descriptors to image texture segmentation with geodesic active contours has been investigated in [24]. In these works, the texture descriptors are evaluated by simple local statistics and thresholds.

On the other hand, machine learning approaches play an important role in modern image processing [12], [4], [13]. During the past two decades, SVMs [3] have been used for virtually all kinds of classification tasks in image processing.

III. AMOEBA CONSTRUCTION

In this section we describe in more detail the construction of amoebas. We assume that images are represented by a regular grid of pixels. The pixels can be interpreted as the vertices of an edge-weighted graph, the *pixel graph*, the edges of which connect adjacent pixels. Regarding what pixels are considered as adjacent, the most common choices are 4-neighborhoods, where adjacency is restricted to vertical and horizontal neighbors, and 8-neighborhoods that include also diagonal neighbors. Figure 1 shows two pixel graphs for the local environment of the seed pixel v_0 highlighted in the center according to the methods described later on in this section: one resulting from a 4-neighborhood and one from a 8-neighborhood.

Following [22], amoebas are constructed using Dijkstra's shortest path algorithm [7]. Similar to a region growing-based approach, local neighborhoods are established by using the neighborhood methods stated before. Starting from the seed pixel, region growing iteratively includes new neighbors to the graph until their distance from the seeding pixel

(a) 4-neighborhood (b) 8-neighborhood

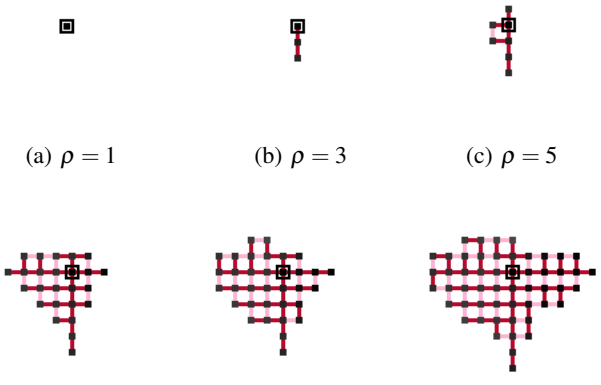
Fig. 1: Pixel graphs based on 4-neighborhoods and 8-neighborhoods for the same originating pixel. The Dijkstra tree is shown in dark red, all remaining edges in light red.

exceeds ρ. The edge weights, which enforce locally adaptive amoeba shapes, are defined according to [22] as

$$w_{p,q} := \left(\|p-q\|^2 + \beta^2 |u_p - u_q|^2 \right)^{1/2}, \qquad (1)$$

where $\|p-q\|$ is the Euclidian distance between the vertices p and q; $|u_p - u_q|$ denotes the gray value difference of the corresponding pixels. The *contrast scale* β allows to weight between both values and will be subject to closer investigation in Section VIII.

Figure 2 shows how amoebas evolve for varying ρ together with their Dijkstra trees.



(a) $\rho = 1$ (b) $\rho = 3$ (c) $\rho = 5$

(d) $\rho = 7$ (e) $\rho = 9$ (f) $\rho = 11$

Fig. 2: Amoebas and Dijkstra shortest path trees for one pixel at different ρ. The seeding pixel is in the center.

Amoebas and their respective pixel graphs can be computed for each pixel in an image. However, for the texture classification presented in this paper it might also suffice to compute these for a subset of pixels.

In the following section, we will review some texture descriptors obtained from amoebas and their respective Dijkstra trees.

IV. AMOEBA DESCRIPTORS

Suitable descriptors for texture include graph indices as well as other features encoding amoeba properties. As shown in a large-scale evaluation [6], there exist hundreds of possible candidates which are derived from vertex distances, information-theoretic methods, or edge connectivity. For our current investigation we only use a few selected features, based on previous work [22]. The presented descriptors are invariant to the pose of a texture or a respective object in an image. This is an advantage over some other approaches, like neural networks.

A. GRAPH INDICES

Quantitative graph descriptors can be computed for a graph G. Here, graph indices are obtained from the Dijkstra trees that have been extracted from amoebas before. They can further be divided into distance-based indices, such as the Wiener index and the Harary index, and information-theoretic concepts, such as the Bonchev-Trinajstić information indices or Dehmer entropies. Some of the previously investigated graph indices in [22] are similar to each other, such as the Wiener and Harary indices. We would like to avoid such redundancies because we aim for encoding distinct amoeba properties in order to achieve good classification results.

The **Wiener index** is defined as

$$W(G) := \sum_{1 \le i < j \le n} d(i,j). \qquad (2)$$

with $d(i,j)$ representing the distance between vertices v_i and v_j. We complement the distance-based approach of the Wiener index by adding a functional based on a **Dehmer entropy** [5] and derived in [22]. It reads

$$f^V(v_0) = \mathrm{e}^{M \Sigma_{j=1}^{n} q^{d(0,j)}} \qquad (3)$$

where v_0 is the seeding vertex as used in the region growing approach. For our current investigation, we follow [22] in fixing the parameters to $M = 1$ and $q = \mathrm{e}^{-0.1}$. There are candidates for additional descriptors, including the methods shown in [22]. In the context of a machine learning based approach, the texture classification does not benefit from a high number of descriptors if they do not encode additional amoeba properties. Depending on the pixel graph they are based on, some graph indices, like the Bonchev-Trinajstić information indices, generate values that are extremely spread between high or low values. That makes them harder to handle within a machine learning framework, where a proper scaling or normalisation of features is required. Preprocessing by suitable transformations of the value range may be necessary for these.

B. GEOMETRIC AMOEBA DESCRIPTORS

The second class of amoeba descriptors to be considered does not rely on graph-based concepts like the graph indices shown in the section before. Instead, geometric features of the amoeba shapes are used. In order to compute the eccentricity of an amoeba A given by the set V of its pixels, we start by introducing the moments

$$m_{l,k} = \sum_{V} \left(u(x,y) \cdot (x - \bar{x})^j \cdot (y - \bar{y})^i \right) \qquad (4)$$

where $(\bar{x},\bar{y})$ is the mass center of V,

$$\bar{x} = \frac{m_{10}}{m_{00}}\,,\ \bar{y} = \frac{m_{01}}{m_{00}}\,, \tag{5}$$

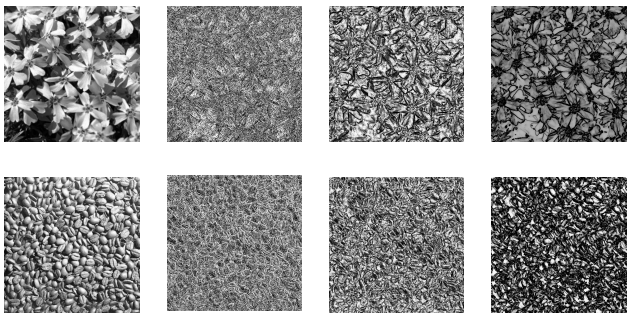
and $u(x,y)$ represents the grayvalue in the respective location. We write down the matrix of second order moments for the respective combinations of l and k as

$$M = \begin{bmatrix} m_{02} & m_{11} \\ m_{11} & m_{20} \end{bmatrix}. \tag{6}$$

From the spectral decomposition of M, we obtain the major moment $m_{\max}$ as the larger eigenvalue, and the minor moment $m_{\min}$ as the smaller one. The **eccentricity** of the amoeba is then given by

$$\varepsilon(A) = \sqrt{1 - \frac{m_{\min}}{m_{\max}}}\,, \tag{7}$$

and serves as a geometric amoeba descriptor. Further candidates for descriptors of this class include other measures derived from moments, like Hu-moments [11], but are not included in the current investigation. Figure 3 shows the amoeba descriptors that have been presented so far for two exemplary gray-value images.



(a) u_g (b) $f^V(v_i)$ (c) $\varepsilon(A)$ (d) $W(G)$

Fig. 3: Examples for texture classes *Flowers2* and *Food1*: Grayscale image, Dehmer entropy, Eccentricity and Wiener index, $\beta = 0.5, \rho = 11$. The input images are converted to grayscale, downscaled and clipped from the *VisTex* database [18], see Fig. 4.

V. FEATURE ENCODING

The texture descriptors obtained in the last section do not yield good results when directly applied to machine learning on a pixel-based approach due to their spatial variation. Therefore the descriptors are subjected to an encoding step before forwarding them to SVMs.

Prior to feature encoding, we rearrange the computed values of the N_D descriptors in a matrix structure with one column per descriptor, and $M \times N$ rows each corresponding to one pixel of a training image. Here, M is the number of rows, and N the number of columns of the training images. For training, square and randomly sampled subimages with $M = 64$ and $N = 64$ are used. Using the three amoeba-based descriptors presented before, this results in a 4096×3 matrix. For the following training tasks, the columns of this matrix are interpreted as feature vectors

$$\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{M\times N} \in \mathbb{R}^{N_D}\,. \tag{8}$$

The subsequent encoding is based on **histograms** obtained from cluster information. Clustering is here applied to the full training set that includes several training images for each class. In this way, we obtain a *textural vocabulary* comparable to the *visual vocabulary* as stated in [2]. It describes all possible textural variants present in the training set, and encodes each training sample as a histogram, composed of the number of occurencies of the respective texture in an image.

First of all, feature vectors are obtained for all classes, all graph indices and all images in the training set. The training set is encoded as

$$\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{M\cdot N\cdot N_C\cdot N_I} \in \mathbb{R}^{N_D}\,. \tag{9}$$

Note that for the given subimage size as well as for the number of classes $N_C = 16$ and for the training images per class N_I, the structure can potentially be very large. However, this is feasible for the given small dataset with a low number of training samples; for larger databases a Monte-Carlo approach can be considered.

Clustering through K-means is applied to the feature vectors from (9). The resulting cluster centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K \in \mathbb{R}^{N_D}$ represent the textural vocabulary on which the following classification is based. Each of the training image samples $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{M\cdot N} \in \mathbb{R}^{N_G}$ is assigned to its closest cluster, with the assignments given as $q_1, \ldots, q_{M\cdot N} \in \{1, \ldots, K\}$. The histogram $\boldsymbol{f}_{\text{hist}} \in \mathbb{R}^K$ is then given by $\left[\boldsymbol{f}_{\text{hist}}\right]_k = |\{i : q_i = k\}|$.

The training data for the entire data set then results in

$$\boldsymbol{f}_{\text{hist},1}, \ldots, \boldsymbol{f}_{\text{hist},N_C\cdot N_I} \in \mathbb{R}^K\,, \tag{10}$$

describing a K-dimensional feature space, where K is the number of clusters chosen for the K-means clustering, that contains training data for all N_C classes and N_I training samples per class. The data set for classification testing is obtained in the same way.

Table I shows the textural vocabulary, using the three graph indices shown in the columns.

VI. TRAINING DATA

As our focus is on image data with rich texture information, we choose the VisTex database [18] to test the discriminative power of texture descriptors. This database is well-suited for such a task as it contains many images with pure texture information without additional objects, variation in lighting or occlusions, which might distract from the desired goal. A subset of the original color images from the VisTex database, each with a resolution of 512×512 pixels, is manually grouped by visual textural similarity prior to classification, resulting in 16 classes. Some classes of the original database contain visually highly dissimilar images and are therefore split into subclasses. For instance, the class *Fabric* is divided into six subclasses that can easily

TABLE I: Textural vocabulary, represented by cluster centers, for the selected texture descriptors, $\rho = 7, \beta = 0.29$ and $K = 16$

k	$f^V(v_0)$	$\varepsilon(A)$	$W(G)$
1	116.4	16.2	110.8
2	166.9	22.3	147.2
3	136.4	60.0	124.1
4	120.7	10.6	150.9
5	178.7	72.7	148.9
6	224.4	91.2	177.2
7	210.2	42.0	170.5
8	112.0	53.9	82.6
9	108.0	12.4	65.3
10	106.0	46.9	15.3
11	165.9	127.6	140.3
12	151.8	0.0	6.3
13	229.5	151.6	180.5
14	117.8	116.0	87.0
15	254.7	254.8	251.0
16	110.2	246.1	71.7

be distinguished visually. Our final set of training samples is composed of 56 images. All images are converted to gray scale according to ITU Rec. 601,

$$Y \leftarrow 0.299\,R + 0.587\,G + 0.114\,B, \tag{11}$$

where Y, R, G, B denote the grayvalue and the red, green, and blue intensities, respectively.

Note that the class assignment has been made without attention to the strengths or weaknesses of the methods investigated in this paper. There are classes that may be discriminated easily as well as classes that may be very hard to discriminate from others, or have large intra-class variation, which poses a challenge to texture classification.

Table II displays the assignment of images from the original VisTex database to classes in building the training set. The unique identifier used as an alternative to the subset name is found in the column *idx*, while the column *indices* indicates the original index of the image, for instance, 7 in the first row is the index of Bark1.0007.png. Finally, *imagecount* represents the total number of images in each respective class.

TABLE II: VisTex subclass labels

Subsetname	*idx*	*indices*	*imagecount*
Bark1	a	7,8,9,10,11,12	6
Fabric1	b	0,1,2,3	4
Fabric2	c	8,9,10	3
Fabric3	d	11,12	2
Fabric4	e	13,14	2
Fabric5	f	15,16	2
Fabric6	g	18,19	2
Flowers1	h	0,1	2
Flowers2	i	2,3	2
Flowers3	j	4,5,6,7	4
Food1	k	2,3,4	3
Grass1	l	1,2	2
Metal1	m	1,2,3,4,5	5
Sand1	n	0,1,2,3,4,5,6	7
Stone1	o	4,5	2
Water1	p	0,1,2,3,4,5,6,7	8

We intend that the classification methods that are investigated in the following, achieve a good performance using small training data sizes. However, many approaches to machine learning would require large amounts of training data.

The training samples originate from these classes as 64×64 subimages that have been obtained from the 512×512 images. To compensate for the small size of the data set and the fact that some of the subclasses consist of as few as two images, we generate a larger data set by extracting subimages of size 64×64 as stated in the previous section. The subimages are taken from random locations in the large images.

Furthermore, to avoid boundary effects, we compute amoebas only for seed pixels which have a distance of at least ρ from the image boundary, where ρ is the maximum amoeba radius.

Images containing strong color information like the flower classes could easily be distinguished from others by comparing their colors. As we aim for classification through texture information only, we perform a grayvalue conversion on all images before applying further methods. The average grayvalue or brightness of an image will not directly influence texture descriptors presented above. Figure 4 shows one example image in color for each class.

We employ two distinct data sets for the training and evaluation of the machine learning framework. Thereby, weaknesses like overfitting would be well detected: An algorithm suffering from overfitting would perform well on the training set but would fail to generalize, and thus achieve bad performance on unseen test data. In generating our training data set, the subimage extraction step as described above takes a similar role as common alternative methods for enlarging small data sets such as the data augmentation techniques that are popular in the context of neural networks. As in the shown approach both data sets are based on the same images, they may contain the same image regions and thus are not fully independent from each other. Another strategy to training set organization that could be considered in order to overcome the limitations of small data sets like the VisTex database would be n-fold cross-validation.

VII. TRAINING ARCHITECTURE

Applying a classic machine learning approach, we start by extracting a number of descriptors from an image. The resulting features are encoded using histogram encoding as described above, and then handed over to linear SVMs for training. We refrain from using higher-order kernels, as they suffer from overfitting in this task. To assess how well the methods perform on the testing data set, we measure the accuracy of the classification. Higher accuracy measures indicate better performance. It is well-known that the accuracy score does not assess classification properly if the data set is unbalanced [21]. In our setting, this should not constitute a problem since we use equal data set size for all classes during training. Accuracy is therefore considered an appropriate tool for basic performance evaluation. We refrain thus from

(a) Bark1 (b) Fabric1 (c) Fabric2 (d) Fabric3
(e) Fabric4 (f) Fabric5 (g) Fabric6 (h) Flowers1
(i) Flowers2 (j) Flowers3 (k) Food1 (l) Grass1
(m) Metal1 (n) Sand1 (o) Stone1 (p) Water1

Fig. 4: Example images for each VisTex class obtained from the VisTex database [18]. ©1995 Massachusetts Institute of Technology. Developed by Rosalind Picard, Chris Graczyk, Steve Mann, Josh Wachman, Len Picard, and Lee Campbell at the Media Laboratory, MIT, Cambridge, Massachusetts. Under general permission for scholarly use.

using additional values like recall, precision or mean average precision (mAP), which are popular performance indicators in recent research [8].

SVMs, as introduced by Cortes and Vapnik [3], are a standard method for binary classification, still, they can be applied to a given multiclass classification task. For the discrimination of two classes, one SVM is sufficient. For multiclass classification, as shown here for 16 classes, multiple SVM classifiers are necessary. For each SVM, the current class is interpreted as positive, whereas all other classes are treated as negative. Effectively, one must train one SVM per class, which comes down to 16 SVMs in our setting. Figure 5 summarizes all steps taken to classify texture within this classic machine learning framework.

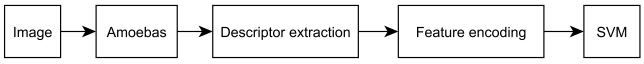


Fig. 5: Classic machine learning model for images applied to texture classification with amoeba descriptors

For the following trainings we use 8 training samples per class and histogram encoding, as well as 32 samples in a testing set for accuracy estimation.

VIII. RESULTS

In this section we evaluate how amoeba-based texture descriptors perform within the previously described architecture with histogram feature encoding. Some of the parameters involved have a large impact on the performance of this approach. We will therefore subject these to a closer investigation.

The size of the textural vocabulary obtained by the training stage depends on the number of clusters defined by the clustering method. Figure 6 shows the classification accuracy for different choices of K between 8 and 64. The values for K are chosen based on the number of classes and different textures used in this data set. As the given data set contains 16 classes, the values of K in our tests are chosen around 16.

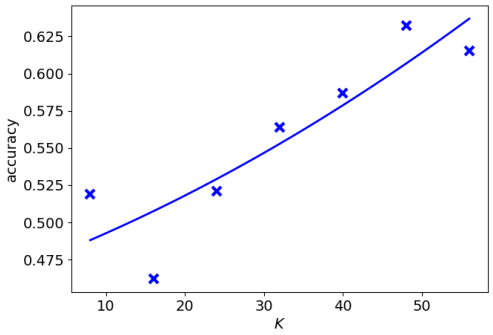


Fig. 6: Accuracy for varying K with $\beta = 0.29$ and $\rho = 7$ and 4-neighborhoods

As can be seen, in general, larger K leads to better performance, with the accuracy reaching a plateau for $K = 48$ and higher. For our further tests, we retain $K = 48$, which warrants faster computation than larger values. Furthermore, we avoid the problems that come with high-dimensional feature spaces and are known as "the curse of dimensionality". As expected, choosing $K < N_C$ results in a performance drop, as different textures have specific cluster centers which tend to be ignored in this case.

The construction of amoebas includes many possible design choices than can be made, while some restrictions are given by the current implementation. Table III gives an overview of the choices.

TABLE III: Amoeba parameters for closer investigation

Parameter	choices
Neighborhood	4-nbhd, 8-nbhd
Norm type	L^1, L^2, L^∞
Patch type	Euclidian patch, amoeba
Edge-weight type	weighted, unweighted
Graph type	fully connected, Dijkstra tree
β	0.10…0.43
ρ	3…9

As described in the introduction, possible choices for the local neighborhood include 4-neighborhoods and 8-

neighborhoods. Alternatives to the L^2 norm in the computation of the distance metric are L^1 or L^∞ norms. Instead of locally adaptive amoebas, fixed Euclidian patch may be considered as mentioned already in [22], [23]. Graph indices can be computed either from fully connected pixel graphs, from Dijkstra trees or even from unweighted Dijkstra trees where the edge weights have been stripped off.

For closer investigation we stick with the L^2 norm in amoeba computation as already mentioned in earlier sections, and restrict ourselves to the Dijkstra tree with edge weights. The amoeba parameters β and ρ have a large potential impact on the classification. Therefore, we vary their values within the boundaries stated in Table III. Classification accuracies for sampled values of ρ with 4-neighborhoods as well as 8-neighborhoods are compiled in Table IV.

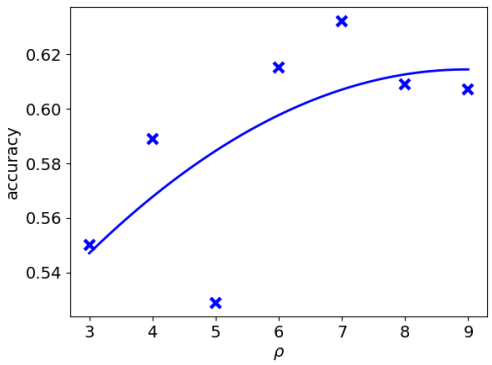
TABLE IV: Classification accuracies for different neighborhoods, and ρ, $\beta = 0.29$

ρ	4-nbhd	8-nbhd
3	0.533	0.548
4	0.589	0.544
5	0.529	0.595

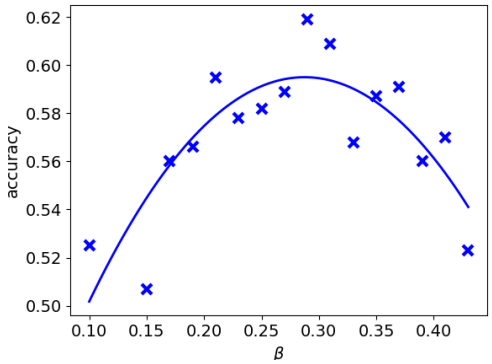
The average values for each parameter from Table IV are shown in Table V.

TABLE V: Average accuracies for parameter values from Table IV

4-nbhd	8-nbhd
0.550	0.562



(a) Accuracy for varying ρ, $\beta = 0.29$ and 4-neighborhoods



(b) Accuracy for varying β, $\rho = 6$ and 4-neighborhoods

Fig. 7: Accuracy for varying ρ and β and their second order polynomial regression

A closer look at the parameters ρ and β is given in Figure 7. The accuracy as a function of β shows a maximum at 0.29, while values in a wider range $\beta \in \{0.20 \ldots 0.40\}$ may still achieve good results. Each data point represents the result of one test set. We expect that using multiple test sets per point would improve the smoothness of the graphs. No class discrimination is possible for $\beta = 0$, as in this case each amoeba has a regular round shape and encodes no texture information whatsoever.

Based on the results, we can summarize our findings regarding the parameters as follows:

- Single values for accuracy vary when using 4-neighborhoods or 8-neighborhoods, there is no clear trend which performes better. Since 8-neighborhoods come with higher computational demand, further investigation will be based on 4-neighborhoods.
- β is best set to lower values. In the test runs shown, best outcomes were observed for $\beta = 0.29$. A more detailed analysis showed that smaller or larger values may still lead to good results. The performance almost continuously decreased for $\beta > 0.3$.
- In most test cases, large values for ρ performed better than smaller ones, with $\rho = 7$ ranging best among the investigated cases. Larger values may perform better, but their use is currently precluded by excessive computational expense.
- Computational expense also increases when β decreases. Note that amoeba shapes get more roundish and approach Euclidian patches for $\beta \to 0$, thus for smaller β amoebas contain more pixels.
- Optimal values of β and ρ show no obvious dependency from each other for $\beta \in \{0.10, \ldots, 0.43\}$ and $\rho \in \{3, \ldots, 9\}$.

Finally, we investigate how classification performs on a smaller number of classes. To this end, we train SVMs on subsets of the 16 classes. Due to the high number of possible combinations only a small selection will be displayed here. Table VI shows the results from choosing random subsets from the training set. Throughout these experiments, 8 training samples per class were used, and 32 samples per class were used for testing.

As expected, the training sets consisting of only two classes are discriminated easily, and the accuracy generally decreases as more classes are involved. For some combinations of two classes, the accuracy may even reach 100%.

However, not all sets containing a certain number of classes result in the same accuracy, as some classes are discriminated more easily from each other, while others are

TABLE VI: Classification accuracies for 30 random class subsets for 2, 4, and 8 different classes. Classes are named according to Figure 4. All tests used $\rho = 7$, $\beta = 0.29$, and $4 - neighborhoods$

classes	accuracy	classes	accuracy
b l	1.0	c e j n	0.843
n o	1.0	a d i p	0.820
m p	1.0	a e h j k l o p	0.800
c e	1.0	a g j k	0.796
b c e l	0.992	a g j m	0.796
d p	0.968	b c d e i l m o	0.785
h n	0.937	j m	0.781
d l	0.937	a b d g h k n p	0.777
i m	0.921	a b c e f j l o	0.765
d e f o	0.875	c g h j l n o p	0.765
a o	0.875	a g m p	0.765
b c e g h i k o	0.867	b f g i j l m o	0.757
f j l p	0.859	a b d g i j k n	0.738
a h n o	0.851	e h j p	0.734
a b e f g k l o	0.847	a d g j l m n p	0.683

harder. The strategy used to create this table may also be used to measure how good individual pairs of classes can be distinguished. A thorough investigation would require to train $\binom{16}{2} = 120$ class pairs.

IX. CONCLUSION

In this paper we have demonstrated that texture descriptors derived from morphological amoebas can be used within a classic machine learning approach to texture classification, and achieve a reliable discrimination of textures. A small number of selected features based on graph indices and geometric properties of amoebas was combined in order to encode texture. Feature values were histogram encoded and fed into a SVM.

By additional experiment series, we have investigated the influence of important parameters of the feature computation on the classification performance. Finally the relation between the number of classes and the classification performance was tested. As expected, fewer classes can be distinguished more accurately.

This work represents a first step into the combination of amoeba-based texture descriptors with machine learning techniques. Future work will be needed to better assess the capabilities of this approach in comparison with other texture descriptors and other machine learning techniques. Alternative clustering methods as well as strategies for optimal parameter choice could be studied in more detail. Moreover, it will be interesting to investigate the applicability of the proposed technique for more advanced texture analysis tasks such as texture-based image segmentation.

REFERENCES

[1] D. Bonchev and N. Trinajstić, "Information theory, distance matrix, and molecular branching," *Journal of Chemical Physics*, vol. 67, no. 10, pp. 4517–4533, 1977.

[2] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Proceedings of the 22nd British Machine Vision Conference*. BMVA Press, 2011, pp. 76.1–76.12, http://dx.doi.org/10.5244/C.25.76.

[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[4] 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.

[5] M. Dehmer, "Information processing in complex networks: Graph entropy and information functionals," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 82–94, 2008.

[6] M. Dehmer, F. Emmert-Streib, and S. Tripathi, "Large-scale evaluation of molecular descriptors by means of clustering," *PLOS ONE*, vol. 8, no. 12, pp. 1–10, 12 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0083956

[7] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[9] R. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[10] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[11] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[12] C. Huang, L. Davis, and J. Townshend, "An assessment of support vector machines for land cover classification," *International Journal of remote sensing*, vol. 23, no. 4, pp. 725–749, 2002.

[13] 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

[14] G. G. Lendaris and G. L. Stanley, "Diffraction-pattern sampling for automatic pattern recognition," *Proceedings of the IEEE*, vol. 58, no. 2, pp. 198–216, Feb 1970.

[15] R. Lerallut, É. Decencière, and F. Meyer, "Image filtering using morphological amoebas," *Image and Vision Computing*, vol. 25, no. 4, pp. 395–404, 2007.

[16] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons," *International Journal of Computer Vision*, vol. 43, no. 1, pp. 29–44, 2001.

[17] O. Lezoray and L. Grady, Eds., *Image Processing and Analysis with Graphs: Theory and Practice*. Boca Raton: CRC Press, 2012.

[18] R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard, and L. Campbell, "VisTex database," Online ressource, http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html, 1995, retrieved 2013-11-20.

[19] D. Plavšić, S. Nikolić, and N. Trinajstić, "On the Harary index for the characterization of chemical graphs," *Journal of Mathematical Chemistry*, vol. 12, no. 1, pp. 235–250, 1993.

[20] P. Soille and J.-F. Rivest, "On the validity of fractal dimension measurements in image analysis," *Journal of Visual Communication and Image Representation*, vol. 7, no. 3, pp. 217–229, 1996.

[21] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[22] M. Welk, "Discrimination of image textures using graph indices," in *Quantitative Graph Theory: Mathematical Foundations and Applications*, M. Dehmer and F. Emmert-Streib, Eds. CRC Press, 2014, ch. 12, pp. 355–386.

[23] ——, "Amoeba techniques for shape and texture analysis," in *Perspectives in Shape Analysis*, M. Breuß, A. Bruckstein, P. Maragos, and S. Wuhrer, Eds. Cham: Springer, 2016, ch. 4, pp. 73–116.

[24] ——, "Graph entropies in texture segmentation of images," in *Mathematical Foundations and Applications of Graph Theory*, M. Dehmer, F. Emmert-Streib, Z. Chen, X. Li, and Y. Shi, Eds. Weinheim: Wiley-VCH, 2016, ch. 7, pp. 203–231.

[25] H. Wiener, "Structural determination of paraffin boiling points," *Journal of the American Chemical Society*, vol. 69, no. 1, pp. 17–20, 1947.

Depreciating Motivation and Empirical Security Analysis of Chaos-Based Image and Video Encryption

Mario Prieshuber[1], Thomas Hütter[1], Stefan Katzenbeisser[2], and Andreas Uhl[1]

Abstract

Over the past years an enormous variety of different chaos-based image and video encryption algorithms have been proposed and published. While any algorithm published undergoes some more or less strict experimental security analysis, many of those schemes are being broken in subsequent publications. In this work we show that two main motivations for preferring chaos-based image encryption over classical strong cryptographic encryption, namely computational effort and security benefits, are highly questionable. We demonstrate that several statistical tests, commonly used to assess the security of chaos-based encryption schemes, are insufficient metrics for security analysis. We do this experimentally by constructing obviously insecure encryption schemes and demonstrating that they perform well and/or pass several of these tests. In conclusion, these tests can only give a necessary, but by no means a sufficient condition for security. As a consequence of this work, several security analyses in related work are questionable; further, methodologies for the security assessment for chaos-based encryption schemes need to be entirely reconsidered. For more details, we would like to refer to the original work [1].

REFERENCES

[1] M. Preishuber, T. Hütter, S. Katzenbeisser, and A. Uhl, "Depreciating motivation and empirical security analysis of chaos-based image and video encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2137–2150, 2018. [Online]. Available: 10.1109/TIFS.2018.2812080

[1]University of Salzburg, Department of Computer Sciences {mpreis,thuetter,uhl}@cosy.sbg.ac.at
[2]University of Darmstadt katzenbeisser@seceng.informatik.tu-darmstadt.de

Contributed Session 4

A Network Traffic and Player Movement Model to Improve Networking for Competitive Online Games

Philipp Moll, Mathias Lux, Sebastian Theuermann, Hermann Hellwagner[1]

***Abstract*— The popularity of computer games and e-sports is enormously high and still growing every year. Despite the popularity computer games often rely on old technologies, especially in the field of networking. Research in networking for games is challenging due to the low availability of up-to-date datasets and network traces. In order to achieve a high user satisfaction while keeping the network activity as low as possible, modern networking solutions of computer games take players' activities as well as closeness of players in the game world into account. In this paper, we analyze the Battle Royale game mode of the online multiplayer game *Fortnite*, where 100 players challenge each other in a king-of-the-hill like game within a constantly contracting game world, as an example for a popular online game with demanding technical requirements. We extrapolate player movement patterns by finding player positions automatically from videos, uploaded by Fortnite players on popular streaming platforms and show, how they influence network traffic from the client to the server and vice versa. This extended abstract features the highlights of [1], which has been accepted at the NetGames 2018 event.**

Regarding networking aspects, most games rely on decade old techniques. They still transmit their time critical information, such as player positions and game world updates, via UDP, and use TCP for matchmaking and information presented in context of the game itself. Neither of these protocols was designed for online games. That is noticeable at every big game launch, where game servers crash under the load of hundreds of thousands of players and network nodes struggle to keep up.

Novel information-centric networking (ICN) architectures may be better suited for multiplayer online gaming as ICN approaches have a strong focus on the information itself. In current research, the characteristics of Named Data Networking (NDN) [2], which is one implementation of an ICN, are used beneficially in order to reduce redundancy and network latency. However, the lack of extensive data sets and proprietary software makes research on alternative networking architectures for online games challenging.

The goal of our work is to provide means to test new networking approaches in the context of online gaming. We focus on the popular game *Fortnite* and analyze it regarding its network traffic, game mechanics that influence networking, i.e. player encounters and players' positions in the game world, and the average player behavior. Based on our observations we extrapolate game network traffic in combination with simulated user behavior that can be used to simulate the progress of a game round and the network traffic generated.

[1]Institute of Information Technology, Alpen-Adria-Universität Klagenfurt
{firstname}.{lastname}@aau.at

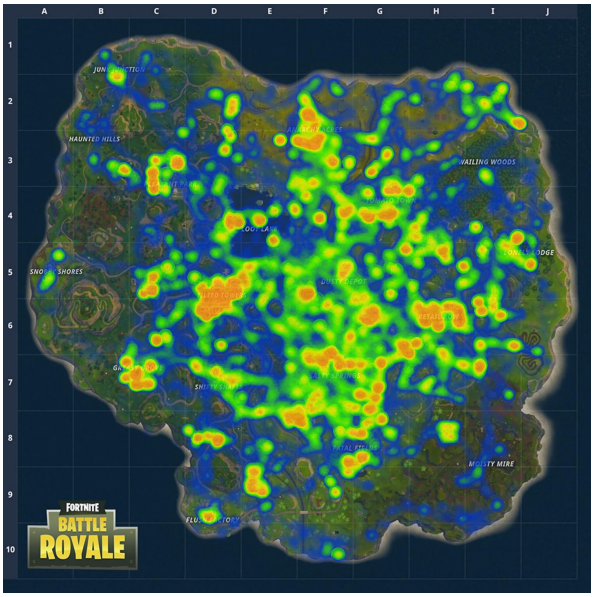


Fig. 1. Heat map from player movements in Fortnite Battle Royale generated automatically from more than 36 hours of game streams.

We first obtained more than 36 hours of streamed in-game video footage from YouTube and utilized OpenCV's template matching algorithms to track player positions on the game map. Based on the players' movement patterns, we obtained a heat map of hot spots, where players moved frequently throughout the analyzed games, as shown in Fig. 1. We then analyzed the basic structure of a game round, which is very well defined for the Battle Royale game genre, and modeled random player movements and encounters throughout the game world. In combination with network traces obtained by playing the game, we modeled network traffic for each client and inferred the network traffic on the server. The resulting tools to simulate game rounds are provided as open source software and are available with sample simulation output for further research on GitHub: `https://github.com/phylib/FortniteTraces`.

References

[1] P. Moll, M. Lux, S. Theuermann, and H. Hellwagner, "A network traffic and player movement model to improve networking for competitive online games," in *Proceedings of the 16th Annual Workshop on Network and Systems Support for Games (NetGames 2018)*. IEEE, 2018, p. to appear.

[2] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 66–73, July 2014.

Reliably Decoding Autoencoders' Latent Spaces for One-Class Learning Image Inspection Scenarios

Daniel Soukup[1] daniel.soukup@ait.ac.at Thomas Pinetz[1] thomas.pinetz@ait.ac.at

***Abstract*— In industrial quality inspection, it is often the case that a lot of data of desired product appearance can be provided at training time, while very little erroneous examples are available. Thus, in order to train an inspection system, the target appearance has to be learned independently from the availability of defect samples. Defects have to be identified as anomalies w.r.t. the trained data distributions in the online inspection phase. In deep learning, autoencoders are a well known choice to realize anomaly detection scenarios, where significantly larger reconstruction errors of objects' images indicate defects. However, as the latent code contains enough information to reliably reconstruct good example images, the question arises if a decision about the validity of an input image can already be drawn in that latent space during online inspection. This would speed up the system by more than a factor of 2 by sparing the processing of the autoencoder's decoder part. Variational Autoencoders (VAE) are a modern variant of the classical autoencoder architecture, which could facilitate this purpose, because of its imposed regularization term, that forces the latent codes to be standard normally distributed.**

I. INTRODUCTION

In quality inspection of highly optimized industrial production processes, e.g. textile industries, a low rate of flaws is usually observed. This results in small and unrepresentative samples of defects. In such cases, inspection systems have to learn the valid product appearances only by means of valid product samples. Only later during actual inspection, some defects occur occasionally, which then have to be identified as anomalies or novelties w.r.t. the trained data distributions. Such a setting is referred to as *one-class learning*, *anomaly detection*, or *novelty detection* [8]. For image processing tasks, this kind of inspection is difficult for patterns that are on the one hand regular and contain repetitive structures on different scales, while on the other hand local variations and distortions are possible, which let each object region appear slightly different than its neighboring regions. An example of such a product type are textiles (Fig.1).

The majority of object regions are valid but vary slightly w.r.t. a trained area. An appropriate representation of the object structure under inspection should reflect a distinct deviation of defective areas, but at the same time it should be robust w.r.t. allowed distortions occurring due to inherent perturbations in the production processes.

A number of algorithms have been proposed to handle one-class learning problems. Very popular are one-class SVM [11], which separate the training data from the origin of the feature space using a hyperplane with maximum margin. SVMs can implicitly be applied to nonlinear and high-dimensional feature spaces. Sparse coding or convolutional sparse coding (CSC), respectively, was proposed to tackle the problem of novelty detection in images of nanofibrous material production [3] [4] by learning dictionaries to yield accurate and sparse representations.

We aim to process images of product parts, where the relevant, representative features have to be implicitly determined by the method itself in the course of the training procedure on multiple scales. Thus nowadays, Convolutional Neural Networks (CNN) are a reasonable choice. In deep learning, autoencoders [9] are a well known tool to perform unsupervised learning of object representations. They were extensively investigated and used for unsupervised pretraining, representation learning, data compression, etc. (e.g. [13]). Autoencoders are trained to reconstruct the input data as exactly as possible through a bottleneck layer of neurons, spanning the so called *latent space*. As a consequence, the autoencoder has to come up with internal representations of the trained patterns (e.g. images) that allow it to reconstruct input data only from those internal compressed vector codes, the so called *latent variables* or *latent codes*. They can be seen as a non-linear version of Principle Component Analysis (PCA) [2], because data are projected to an appropriate subspace in a non-linear manner, e.g. CNN layers, whereas the re-projection error is minimized. Alain and Bengio showed that autoencoders are capable of implicit recovery of the data generating density [1].

Consequently, autoencoders are a well-suited means to handle the image one-class learning task at hand. Cascades of convolutional layers (encoder) enable the identification of relevant pattern features on multiple scales, so that a characteristic, compressed latent representation can be obtained for trained good example images, that enables another cascade of transposed convolutional layers to decode that latent code into a reconstructed image. Good examples, similar to the trained patterns can be significantly better reconstructed than images comprising a deviation w.r.t. to the trained images, i.e. a defect. The autoencoder's decoders are capable of reconstructing input reliably only from the latent codes. Thus the entire structural information about an input image must already be coded in that latent variable. This raises the question, if the decoder part could be fully omitted in the online inspection phase after training is fully accomplished. The final decision about input validity could be solely drawn by evaluating the latent codes, which would spare more than half of the processing effort.

We investigate the opportunities of one-class learning with

[1]AIT Austrian Institute of Technology GmbH, Center for Vision, Automation & Control, Vienna, Austria

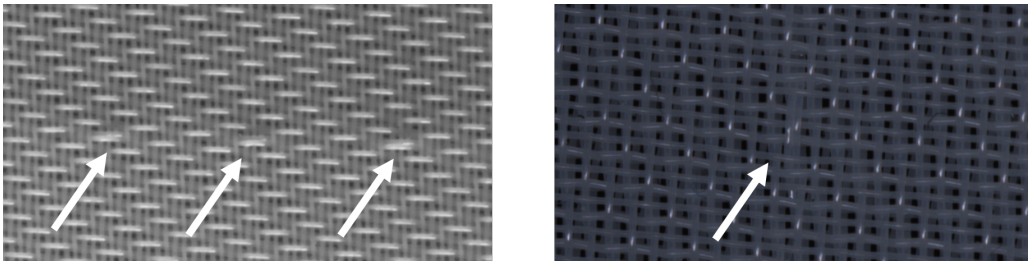


Fig. 1. Two cutouts from the investigated example web patterns with examples of weaving flaws indicated by white arrows. "Pattern 1" (left) and "Pattern 2" (right).

autoencoders, more specifically the reliability of drawing decisions already in the latent space for online applications of autoencoders in one-class learning for image anomaly detection tasks. In Section II, we describe autoencoders and the usage of the latent space in more detail. Results of experiments on two textile examples comprising weaving flaws are presented in Section III. We summarize and conclude in Section IV.

II. One-class learning with (Variational) Autoencoders

Autoencoders are an age-old concept in the area of neural networks (e.g. Rumelhart et al [9]). An autoencoder is a function $f : \mathbb{R}^n \to \mathbb{R}^n$, implemented as a neural network, that is trained to optimally reconstruct input data - images of good product appearance in our case - by minimizing the so-called *reconstruction loss function*

$$L_{rec} = \sum_i ||x_i - f(x_i)||_2^2. \tag{1}$$

f consists of an *encoder* part $c : \mathbb{R}^n \to \mathbb{R}^l$, where $l < n$, and a *decoder* part $d : \mathbb{R}^l \to \mathbb{R}^n$, such that $f = c \circ d$. As $l < n$, the autoencoder's weights have to adjust in such a way during the optimization process, that the l-dimensional latent codes contain all the information of the trained input patterns in order that the decoder d can reconstruct input images x_i of the training data distribution. Additionally, we make use of convolutional layers' expressive capabilities to extract representative image features. Thus we implement the encoder and the decoder as deep CNNs or tranposed CNNs, respectively.

In one-class learning, training data only contain examples of valid object appearance. When an image comprising a defect, e.g. weaving flaw, is presented to the fully trained autoencoder, the corresponding reconstruction error will be significantly larger than reconstruction errors of trained valid examples, by which the defect is detectable in the online phase.

However, as the total information about the training patterns and naturally deviations of which must already be mapped in the corresponding latent codes, there should be a way to already draw the decision from those latent codes. Such an early decision would speed up inspection processing in the online phase. Schlegl et al [10] managed to exactly implement this idea by means of Generative Adversarial Networks (GAN), which had been trained to generate artificial samples of the valid image data. Defects could be detected by the discriminator network, which was trained to detect anomalies w.r.t. to the training distribution. Makhazani et al [7] presented the Adversarial Autoencoder, a variant of a more classical autoencoder, where an adversarial network was trained to match the distribution of the latent codes with a predefined appropriate data distribution, e.g. standard normal distribution, by means of an added regularizing loss term to the reconstruction loss. Such a simple-shaped latent distribution would simplify the evaluation of latent codes in the one-class learning setting, because deviations from it can easily be detected.

The so-called Variational AutoEncoder (VAE) was introduced by Kingma et al [6]. Similarly to the Adversarial Autoencoder, the VAE is realized by adding a regularizing loss term to the reconstruction loss. This *latent loss* measures the dissimilarity of the latent codes' distribution to a predefined well-shaped target distribution, i.e. standard normal distribution, by means of the *Kullback-Leibler* divergence (KL):

$$\begin{aligned} L &= L_{rec} + \lambda \cdot L_{lat}, \quad with \\ L_{lat} &= KL(Q(z|X), \mathcal{N}(0,I)), \end{aligned} \tag{2}$$

where $Q(z|X)$ is the PDF of the distribution of latent codes $z_i = c(x_i) \in Z \subset \mathbb{R}^l$ given training examples $x_i \in X$. Usually, $Q(z|X) := \mathcal{N}(\mu(X;\theta), \Sigma(X;\theta))$, where μ and Σ are estimated by a neural network, in our case the encoder c. Although a regularization parameter λ is generally not required for VAE, in our application it was necessary in order to decrease the influence of the latent loss L_{lat}. Otherwise, L_{lat} dominates L_{rec} in the training process and the latent codes collapse to zero mean. That violates the main objective of optimal reconstruction, as those flattened latent codes do not have the expressive power to code pattern structures for reconstruction anymore. Doersch [5] provides a good tutorial over VAE, where he also discusses the requirement of regularization parameters for VAE.

In the online phase of the inspection with VAE, the decision, if an image contains a defect or is similar to the trained valid image distribution, could be made by means of the latent loss L_{lat}, which is at least in the average significantly larger for anomalies than valid data, just like for the reconstruction loss with classical autoencoders. We

analyze the applicability and reliability of decision making in one-class learning on the basis of latent VAE loss rather than the reconstruction loss in the next section on the basis of two textile examples.

III. Experiments

We present experiment results for two different web patterns (Fig. 1). The webs comprise regular structures, but also swirling local variations, which are typical for textiles. The autoencoders have to capture the distribution of allowed pattern variation from a set of sampled patches, where no weaving defects occur (*training set*). From another region of valid product appearance, patches were sampled which are not used in training, but only for validation (*validation set*). Around defect regions, i.e. weaving flaws, we extracted randomly distributed patches containing those defect patterns (*defect set*). In the setting of one-class learning, those were naturally also not used for training, as they are assumed to be not available in sufficient amounts for training in real scenarios. The size of all patches was fixed to 64×64 pixels, a field of view (FOV) where regular structures and disruptions of which are apparent.

According to the size of image patches, the input size of the autoencoder has a FOV of 64×64 as well. The autoencoder architecture is a U-shaped CNN with a bottleneck in the middle, yielding the latent codes $c(X)$, which is inspired by architectural elements from VGG[12] architectures, where only small filters are used, e.g. 3×3 to stick with the VGG scheme. In the encoder part c, the resolution of feature maps is decreased at every convolutional layer by strides of 2, while the number of feature maps is increased by a factor of 2. The decoder d, which generates the reconstruction of the input images from the latent codes, is structured analogously, only in a transposed manner, i.e. the number of feature maps is reduced and resolution of feature maps is increased. For all convolutional layers and all except the last transposed convolutional layers, the ReLU non-linearity was chosen. The last transposed convolutional layer is complemented by a *tanh* non-linearity.

The search for an optimal regularization parameter λ, balancing the influences of reconstruction and latent losses L_{rec} and L_{lat}, respectively, was conducted by repeating the training process with different values of λ and choosing the one, for which the reconstruction loss and the latent loss deviate most significantly between valid and the few available defective patches. For both experiment patterns, $\lambda = 10^{-5}$ was optimal. For a real scenario, where absolutely no defective examples are available in the training phase, this λ search is not applicable.

The appropriate learning rate was 10^{-3} and all autoenoders were trained for 10000 iterations. The training curves appeared to be very smooth and precisely reproducible between different runs with varying random training sets.

In Figs. 2 and 3, we visualize the distributions of reconstruction losses and latent losses of individual training (blue), validation (green), and defect (red) patches after the corresponding autoencoders were fully trained. We have augmented those histograms with Gaussian approximations in order to emphasize the gross distribution structures. Moreover, we computed the cross-entropy distances $H(p,q)$ between the training, validation, and defect distributions, respectively, according to

$$H(p,q) = -\sum_i p_i \cdot log_2(q_i), \tag{3}$$

which measures the dissimilarity between two distributions p and q (Tab. I).

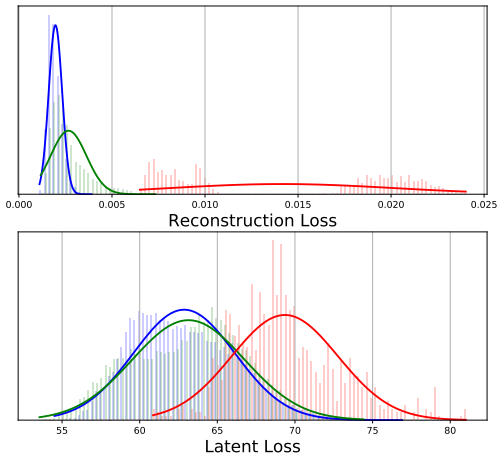


Fig. 2. Distributions of reconstruction loss (top) and latent loss (bottom) of individual patches sampled from Pattern 1. Training patches (blue), validation patches (green), defect patches (red). Histogram distributions of loss values augmented with Gaussian approximations.

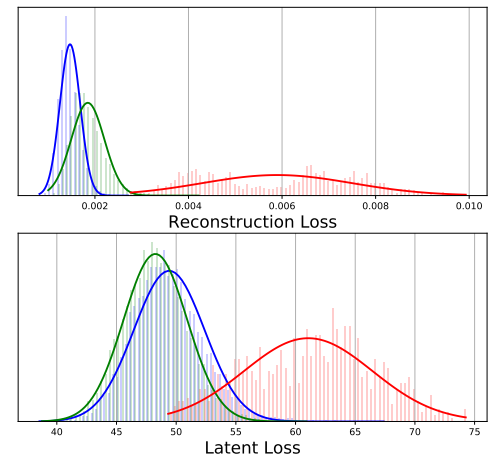


Fig. 3. Distributions of reconstruction loss (top) and latent loss (bottom) of individual patches sampled from Pattern 2. Training patches (blue), validation patches (green), defect patches (red). Histogram distributions of loss values augmented with Gaussian approximations.

For both patterns, from Figs. 2 and 3 as well as from Tab. I, it is apparent that the training and validation histograms are

TABLE I

CROSS-ENTROPY DISTANCES (EQU. 3) BETWEEN DEPICTED LOSS DISTRIBUTIONS OF TRAINING (T), VALIDATION (V), AND DEFECT (D) DATA SETS (FIGS. 2, 3) INDICATING THE DISSIMILARITY OF LOSS VALUE DISTRIBUTIONS BETWEEN DIFFERENT DATA SETS T, V, AND D, RESPECTIVELY.

Cross entropy $H(p,q)$	H(T,V)	H(T,D)	H(V,D)
Pattern 1 - Reconstruction Loss	0.67	20.35	20.30
Pattern 1 - Latent Loss	0.10	7.64	6.95
Pattern 2 - Reconstruction Loss	1.15	19.70	19.27
Pattern 2 - Latent Loss	0.17	6.98	9.38

more similar for the latent loss. Reconstruction losses of validation patches tend to be slightly larger than those of the training patches. However, that is not so disconcerting, as the autoencoders are explicitly optimized to tightly fit the training distributions, mainly by optimizing the reconstruction error. Thus a minor increase of reconstruction errors for not trained valid samples is to be expected and acceptable. More important is that the defect distributions are distinctly deviating from both the training and validation distributions in order to make defect detection feasible at all. While the defect distributions according to the latent loss are overlapping with both valid distributions, defects obviously comprise consistently, significantly larger reconstruction losses than the valid examples. This makes defect detection on the basis of reconstruction loss more reliable. Both observations are also confirmed by the cross-entropy distances in Tab. I.

IV. CONCLUSIONS

In industrial image inspection tasks, one-class learning is a common scenario, where target product appearances have to be learned solely on the basis of valid product examples, because examples of defects are not available in sufficient amounts for training. Autoencoders are a well investigated means in deep learning for learning data distributions in an unsupervised manner. Thus they are appropriate methods for one-class learning, where they are trained to reconstruct input training images through a bottleneck layer of neurons as precisely as possible. In online inspection, input defects result in measurably larger reconstruction errors than valid examples by which they are identifiable. We investigated the opportunities to speed up that process by drawing the inspection decision already from the outputs of that bottleneck layer, the latent codes. In order to simplify the structure of distributions of valid latent codes and therefore the decision making procedure, we applied VAE regularization, where the latent codes are forced to possibly follow a standard normal distribution by means of an added regularization term.

Our experiments with two textile examples show, that in the average, defective images actually comprise larger latent errors by which they could be identified over valid patches. However, an analysis of the distributions of reconstruction errors and latent errors over individual patches, the results indicated that defect images are not as reliably distinguishable form valid images on the basis of latent VAE codes than on the basis of reconstruction errors. In a real inspection task, it would be difficult to set a threshold, which serves as decision boundary between valid and defect images on the basis of latent codes. While drawing the decision from latent codes would speed up computations in the online phase by more than a factor of 2, it seems to be insufficiently reliable. Either more false positives or overseen defects are the consequence. Probably because the reconstruction loss is the main workhorse of autoencoder training and it is explicitly optimized to extract latent codes to images, the decoder is the distinctly smarter tool for analyzing latent codes in one-class learning. If reliability counts, then it is better to invest the computational effort and go for the reconstruction loss as the decision measure. In addition, the training procedure becomes simpler, because the search for an appropriate regularization parameter steering the influence of the latent codes' distribution can be omitted.

REFERENCES

[1] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3563–3593, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2750359

[2] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Netw.*, vol. 2, no. 1, pp. 53–58, Jan. 1989.

[3] G. Boracchi, D. Carrera, and B. Wohlberg, "Novelty detection in images by sparse representations," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (IEEE SSCI)*, Orlando, FL, USA, Dec. 2014, pp. 47–54.

[4] D. Carrera, G. Boracchi, A. Foi, and B. Wohlberg, "Detecting anomalous structures by convolutional sparse models." in *IJCNN*. IEEE, 2015, pp. 1–8.

[5] C. Doersch, "Tutorial on variational autoencoders," 2016, cite arxiv:1606.05908.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013. [Online]. Available: http://arxiv.org/abs/1312.6114

[7] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow, "Adversarial autoencoders," in *International Conference on Learning Representations*, 2016. [Online]. Available: http://arxiv.org/abs/1511.05644

[8] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "Review: A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, June 2014. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2013.12.026

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362.

[10] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Information Processing in Medical Imaging - 25th International Conference, IPMI 2017, Boone, NC, USA, June 25-30, 2017, Proceedings*, 2017, pp. 146–157.

[11] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 1999, pp. 582–588.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[13] 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

Detection of Bomb Craters in WWII Aerial Images*

Simon Brenner[1], Sebastian Zambanini[1] and Robert Sablatnig[1]

Abstract— **The analysis of aerial images from World War II surveillance flights allows a preliminary estimation of unexploded ordnance risk for large scale construction projects. To support this task, which is currently carried out manually, an automatic approach for the detection of bomb craters in such historical images was developed and evaluated.**

I. INTRODUCTION

Unexploded Ordnance (UXO) from World War II still poses a hazard for construction projects in Central Europe. Specialized companies provide a preliminary risk estimation by retrieving and interpreting aerial images from WWII surveillance flights over areas of interest. For this, such historical aerial images have to be georeferenced and searched for certain objects that indicate increased combat activity in the surveyed area, such as bomb craters, trenches or artillery stations. Currently, both the georeferencing and the search for warfare evidence are performed manually by specialists. Within the FFG-Bridge project DeVisOR (**De**tection and **Vis**ualization of unexploded **O**rdnance **R**isks), which was conducted from 2015 to 2017 in cooperation with the Luftbilddatenbank Dr. Carls GmbH (LBDB) as an industrial project partner, methods for the automation of the aforenamed tasks were developed.

This paper is concerned with the detection of warfare evidence in WWII aerial images, where we focused on the detection of bomb craters. First, they are the most frequent type of warfare-related objects found in the aerial images; second, they are the most direct evidence for the presence of UXO, as at least 10% of all bombs that were dropped in WWII are assumed to have not exploded [6]. We developed a machine learning approach based on Convolutional Neural Networks (CNNs) for the automatic detection of bomb craters. Furtheremore, the integration of the detector into the working environment of our industrial partner in the form of a plugin for the GIS *ArcMap* will be elaborated.

II. RELATED WORK

Merler et al. [6] developed an approach to automatically generate UXO risk maps via bomb crater detection in WWII aerial images. They created a set of *eigen-craters* from a principal component analysis of example craters and trained a classifier based on a variant of AdaBoost. The results are promising, but they are aiming at detecting clusters of craters. The dataset they were using for evaluation was not published.

More work has been done on the automatic detection of asteroid impact craters on extraterrestrial surfaces [3][7][5], lately also using CNNs [2]. Although the problem seems to be similar to ours, there are some major differences. First, asteroid impact craters exhibit a high variation in size; second, the source images are of better quality than the historical aerial images; and third, on other planets there are no trees or man made objects that can easily be confused with craters.

III. BOMB CRATER DATASET

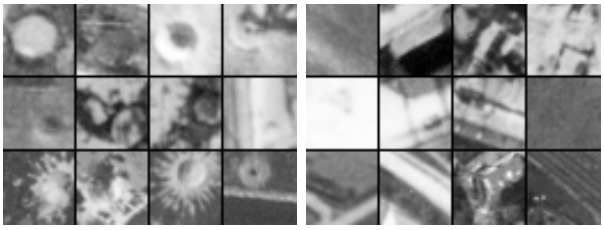
To our knowledge, no labelled dataset for crater detection in historical aerial images is currently publicly available. LBDB as an industrial partner provided a selection of their finished projects, covering both urban and rural areas. In these projects, the historical aerial images have been georeferenced and the bomb craters mapped by experts. The analysis is only performed within a defined region of interest (ROI). Usually, several overlapping historical images from different dates have been georeferenced in the ROI, to ease the determination of warfare evidence by the expert.

In total the provided projects contain about 10000 craters in world coordinates. After semi-automatically assigning the craters to the images in which they are visible, we ended up with roughly 20000 craters in image coordinates, along with their diameters. Equally, the ROI is mapped to the individual images; this is crucial for evaluation, as no ground truth data are available outside the ROI.

The crater positions exported in this format are very flexible and can be used in different ways to train and evaluate machine learning algorithms. For our approach, a CNN was trained for a binary classification problem on image patches (see Section IV). We therefore prepared our training data by extracting image patches of positive and negative examples. As the ground resolutions of the individual aerial images are approximately known, images patches with a fixed absolute size are extracted. A statistical assessment of crater sizes resulted in a mean crater diameter of 7.8m with a standard deviation of 2.0. Assuming a normal distribution, 95% of the crater diameters can be assumed to lie between 3.8m and 11.8m. In order to provide the classifier with some context, we set the patch size to 20x20m. The patches are extracted with a sliding window over the ROIs, with a stride of 1/4 the patch size. Patches containing a crater position are stored as positive examples, patches not containing a crater are randomly selected as negative samples or discarded, in order to match the number of positive samples. This approach was

*This work was 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@cvl.tuwien.ac.at, zamba@cvl.tuwien.ac.at, sab@cvl.tuwien.ac.at



(a) positive (b) negative

Fig. 1: Examples of training patches

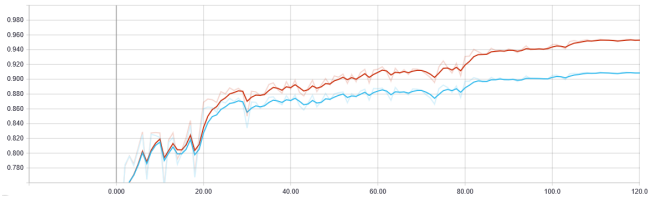


Fig. 2: Training of the CNN. Mean accuracy on the training set (red) and test set (blue) over training epochs.

preferred to the direct extraction of patches centered at the crater positions, as it already mimics the planned detection procedure and implicitly introduces translational variation within the examples. With this approach about 85000 patches per class were generated. Figure 1 shows some examples of extracted patches; note that both the positive and the negative examples are very heterogenous. The dataset was randomly divided into training and test set with a ratio of 4:1 and scaled to fit the CNNs input.

IV. THE CNN CLASSIFIER

First experiments with local binary patterns, Haar features and simple neural networks could not deliver satisfying results. We therefore employed the DenseNet CNN architecture [4], which emerged as state of the art in image classification in 2017. Using the training data described in Section III, we trained a 40 layer DenseNet with an input size of 32x32 pixels on a binary classification problem. The network was optimized using Nesterov Momentum. After experiments with learning rates we ended up at the learning curve depicted in Figure 2, plateauing at a mean accuracy of 0.91 on the test set after approximately 110 epochs.

Detection is performed by moving a sliding window of fixed ground size of 20x20m over the ROI, with a stride of 1/4 the window size, similarly to the generation of the training data. The sub-images covered by the sliding window are then scaled to the correct input size and processed by the CNN, which returns a confidence for that window containing a crater.

As craters only make up an average of 0.4% of the observed areas, classification by simply applying a threshold to the confidences leads to an unfeasible number of false positives; the precision of the raw approach on realistically distributed data was at 0.04. Raising the threshold does not solve the problem; it only lowers the recall without significantly improving the precision. The following section describes a post-processing approach that exploits spatial information to tackle the problem.

V. POSTPROCESSING

To alleviate the precision problems described in the previous section, spatial information and a-priori assumptions about bomb crater distributions are exploited to filter the CNN output. Specifically, the following ideas are employed:

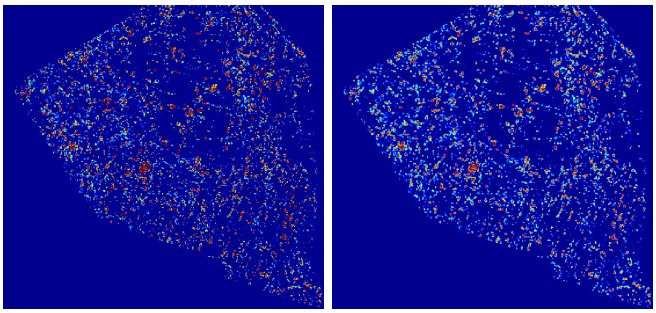
1) **Spatial proximity prior:** Bombs are typically dropped in clusters; 'lonely' detections are therefore more likely to be false positives.
2) **Non-cluster suppression:** As every individual crater is hit by the sliding window more than once, it is unlikely that a real crater is only detected in one of those window positions; therefore detections that are not part of a cluster are considered outliers.
3) **Non-maximum suppression:** This is a standard operation in object detection that reduces multiple detections of the same object to the one with the maximum confidence.

To practically apply the above named ideas, first the confidences computed for sliding window positions are stored in a 2-dimensional *confidence map*. Figure 3a visualizes an example of such a map. To introduce the spatial proximity prior, the map is convoluted with a gaussian kernel with $\sigma = 0.5$ (Figure 3b). This penalizes isolated areas of high confidences. The resulting filtered confidence map is thereafter thresholded to produce a binary *detection map* (Figure 3c). The threshold was set to 0.88, which maximizes the F1-score of the detector. For the non-cluster suppression, 8-connected components with less than 6 elements are removed from the detection image (Figure 3d). Finally, all detections which are not local maxima in the filtered confidence map are removed from the detection map, resulting in the final set of detections (Figure 3e). Figure 3f shows those detections superimposed on the original image.

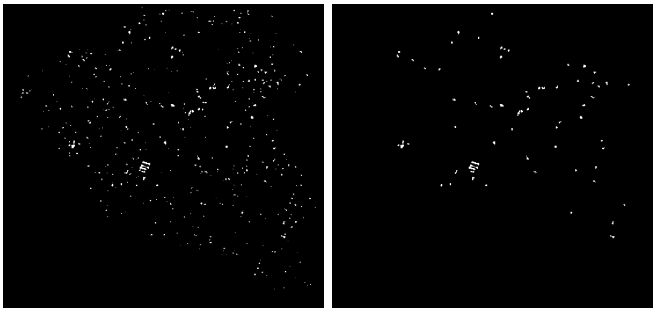
The described procedure converts the CNN outputs to individual crater positions for single images. However, as described in Section III, typically several overlapping aerial images are available for a given ROI. Therefore, detections from different images have to be merged to receive the complete set of craters for a ROI. In this stage, double detections from different images are eliminated. For that purpose, all craters are transformed to world coordinates and a neighborhood-based clustering is applied, where neighborhood is defined by a maximum euclidean distance, which was determined empirically. The clusters are then replaced by their centroids.

VI. RESULTS

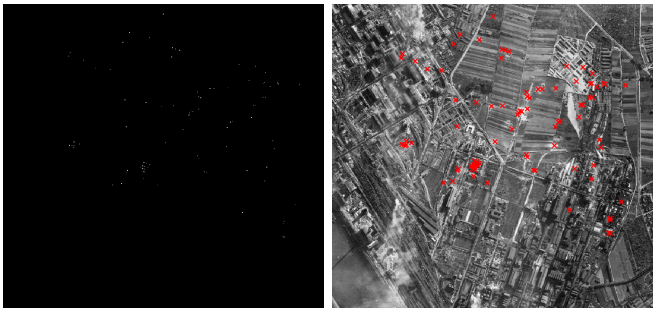
The CNN was evaluated on the test set of image patches as described in Section III. At the maximum accuracy of 0.91 at a decision threshold of 0.5 and a 1:1 ratio of positive and negative examples, patches with craters were detected



(a) confidence map (b) gauss filtered confidence map



(c) detection map (d) non cluster suppression



(e) non maximum suppression (f) final detections

Fig. 3: Visualization of postprocessing steps (example)

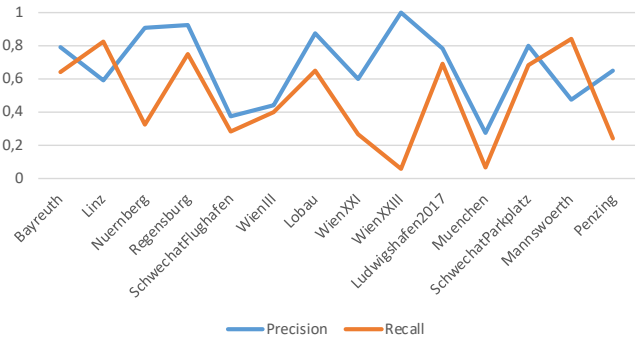


Fig. 4: Precision and recall for the different test projects

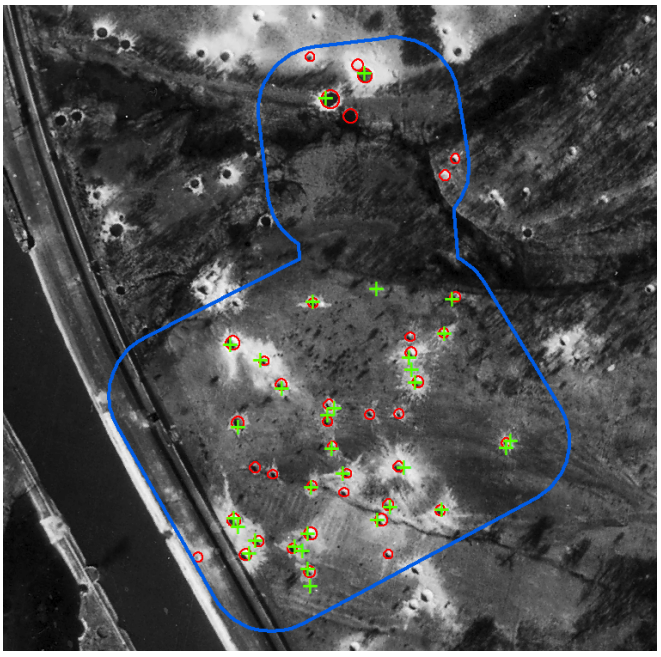


Fig. 5: A detetion result created by the ArcMap plugin. Green: our detected craters. Red: ground truth craters. Blue: ROI.

with precision of 0.907 and a recall of 0.913. In a realistic scenario with a ratio of approximately 250 negative examples on 1 positive example the precision drops to 0.04.

The end-to-end detection solution including the postprocessing steps described in Section V was evaluated on 14 full example projects provided by LBDB with approximately 2500 ground truth craters in total. A detection was regarded a true positive if its euclidean distance to the closest ground truth crater is smaller than the diameter of that crater. Figure 4 shows the precision and recall of the detector for the individual test projects. As shown in this figure, the results vary significantly from project to project. The average results, weighted by the number of ground truth craters present in a project, are a precision of 0.74 and a recall of 0.6.

VII. IMPLEMENTATION

As this work was developed in the course of an industry-oriented project, the described detector was implemented in a form that could easily be adapted into the worlflow of the company partner. To this end we extended the *ArcMap* plugin for the registration of historical aerial images, which was developed earlier in the project [1], with our crater detection approach. The plugin enables the user to automatically detect craters in the source images linked to an *ArcMap* project. These images are first processed by the dense CNN using the TensorFlow framework. The resulting confidences are then post-processed in the plugin code and the resulting detections transformed to the world coordinate system of the project. Finally, the craters from different images are merged. The detected craters are then present as features in the *ArcMap* project and can be refined and edited by the user. Figure 5 shows an example of our detection results.

VIII. CONCLUSIONS

In this project, an approach for the automatic detection of bomb craters in WWII aerial images was developed and implemented. While the performance of our solution does

not allow an application as a fully automated system, it can assist the human specialist in a semi-automated fashion, thus saving time and effort. Additionally, the automated detection can provide a second instance of inspection, in the sense that it may draw the specialist's attention to ambiguous points, thus reducing the chance of overlooking important evidence in large regions of interest.

As a further use case, our approach would allow to automatically generate risk estimations for large areas, using the intermediate confidence maps. This would allow the company to quickly provide potential customers with a first rough estimate of the UXO risk of a certain area, before performing a more detailed analysis.

REFERENCES

[1] S. Brenner, S. Zambanini, and R. Sablatnig, "Image registration and object detection for assessing unexploded ordnance risks a status report of the devisor project," in *Proceedings of the OAGM&ARW Joint Workshop*. Verlag der Technischen Universität Graz, 2017, pp. 109–110.

[2] J. P. Cohen, H. Z. Lo, T. Lu, and W. Ding, "Crater detection via convolutional neural networks," *CoRR*, vol. abs/1601.00978, 2016.

[3] W. Ding, T. F. Stepinski, Y. Mu, L. Bandeira, R. Ricardo, Y. Wu, Z. Lu, T. Cao, and X. Wu, "Subkilometer crater discovery with boosting and transfer learning," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 4, pp. 39:1–39:22, July 2011.

[4] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016.

[5] J. R. Kim, J.-P. Muller, S. van Gasselt, J. G. Morley, and G. Neukum, "Automated crater detection, a new tool for mars cartography and chronology," *Photogrammetric Engineering & Remote Sensing*, vol. 71, no. 10, pp. 1205–1217, 2005.

[6] 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.

[7] P. G. Wetzler, R. Honda, B. L. Enke, W. J. Merline, C. R. Chapman, and M. C. Burl, "Learning to detect small impact craters," in *WACV/MOTION*. IEEE Computer Society, 2005, pp. 178–184.

Semi-Automatic Retrieval of Toolmark Images

Manuel Keglevic[1] and Robert Sablatnig[1]

***Abstract*— In order to identify and solve connected cases forensic experts are currently comparing toolmarks from crime scenes manually. However, especially for frequently occurring crimes like burglaries this task is cumbersome. In order to support the work of the forensic experts, we propose a semi-automated system for finding similarities between toolmark images in large databases. Our methodology uses convolutional neural networks to compute local image similarities in these toolmark images. This work presents the proposed approach and the evaluation conducted on a dataset of more than 3,000 toolmark images collected from real criminal cases.**

I. Introduction

A technique commonly used for break-ins by criminals all over Europe is the so-called *lock snapping*. Using a tool, like for instance a locking plier, door locks can be quietly broken (*snapped*) in a short amount of time using force and leverage. This, however, leaves unique imprints, i.e. toolmarks, of the pliers used on the cylinder locks. As an example, Figure 1 shows multiple toolmarks of the same tool on a broken lock cylinder. By comparing two different toolmarks using a comparison microscope forensic experts can assess if the marks were made by the same tool. This can either be used to confirm that a seized tool was used to commit a crime, or to link multiple cases together and thereby significantly support the investigation of such offenses. Furthermore, the toolmarks found on these locks are crucial as evidence in the following court cases.

Yet, the manual examination and comparison of the toolmarks found is a cumbersome task due to the number of burglaries occurring every year. Therefore, within the project FORMS we developed a semi-automatic system in order to assist the forensic experts. The proposed system consists of an application, which enables the forensic experts to catalog and search toolmark images in a central database, and a methodology based on machine learning. This methodology computes similarities between toolmark images automatically in regions manually annotated by the forensic experts. This way, the forensic experts are presented with a list of toolmarks sorted by similarity in order to reduce the amount of images requiring manual examination. The project FORMS was conducted in cooperation with the Bundeskriminalamt (Criminal Intelligence Service Austria), the CogVis GmbH and VICESSE. It started in Fall 2015 and concluded in February 2018 and was funded by the Austrian Security Research Programme KIRAS.

This paper is structured as follows: firstly, the state of the art in automatic toolmark comparison is presented. Secondly, the dataset, which was created in cooperation with the austrian police, is describe. Thirdly, the methodology based on learning local images similarities using Convolutional Neueral Networks (CNNs) and its evaluation is presented. Finally, we conclude with the advantages and disadvantages of our proposed system and an outlook for future work beyond the FORMS project.

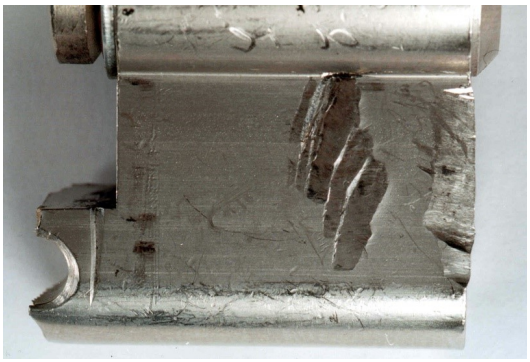


Fig. 1: Snapped lock with multiple toolmarks.

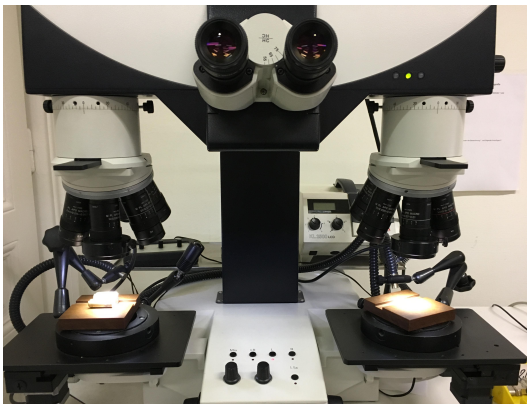


Fig. 2: Leica comparision microscope used by the forensic experts in Austria.

II. State of The Art

Since the validity of comparative forensic examination of toolmarks has been challenged in court, the development of automatic tools for the comparative examination of toolmarks has been in focus of the forensic community to obtain statistical support for the notion of the *uniqueness* of toolmark patterns [14], i.e. the existence of 'measurable feature with high degree of individuality" [1]. For the comparison of striated toolmarks this led to a variety of methodologies [1], [2], [4], [3], [7], [8], [12] which operate on 1D profiles extracted from either 2D images or 3D surface

[1]Computer Vision Lab, TU Wien, 1040 Vienna, Austria keglevic@cvl.tuwien.ac.at

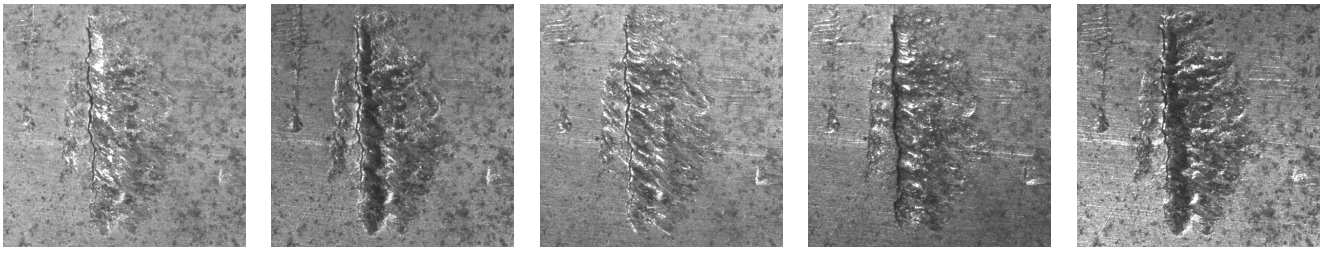


Fig. 3: Toolmark crop under different lighting conditions.

	Tools	Locks	Sides	Images
2015	25	115	230	1,782
2016	23	82	164	1,263
total	48	197	394	3,046

TABLE I: Statistics of the captured toolmark images divided by year.

scans. Similarity scores are commonly computed using either global [2], [4], [3] or local [8] Cross-Correlation (CC). Bachrach et al. [1] propose the use of locally normalized squared distances as similarity measure. In contrast to these approaches, Petraco et al. [12] propose an approach based on machine learning by using dimensionality reduction and Support Vector Machines (SVM) for the classification of the tool. More recently it was shown that CNNs outperform other methods by learning a similarity measure for striated toolmarks [11], [13].

However, all those experiments were carried out on striated toolmarks created under laboratory conditions. This includes for instance fixed angles of attack, constrained lighting conditions, high resolution 3D surface scans and hand-selected tools and surface materials as shown for instance by the only publicly available dataset; the NFI Toolmark dataset created by Baiker et al. [2].

III. Dataset

In order to allow an evaluation of the performance under real-world conditions, we created a dataset using lock cylinders. These lock cylinders were seized by the austrian Police in the course of break-in investigations in Vienna during the years 2015 and 2016. All images were captured using the Leica comparison microscope used by the forensic experts, which is shown in Figure 2. In order to allow an evaluation of the influence of different lighting conditions, a variable light ring was utilized to capture the toolmarks under 11 different lighting conditions. In Figure 3 the effect of different lighting conditions is shown on an example. In total, 197 lock cylinders from 48 linked cases were photographed on both sides. The resulting 3,046 images were divided into training set and test set by year, i.e. 2015 for training and 2016 for testing. In Table I the number of images and locks for each set is listed.

To provide matching local image similarities, matching patches in the images were annotated using a plugin developed for the image viewer nomacs. In this tool polylines are

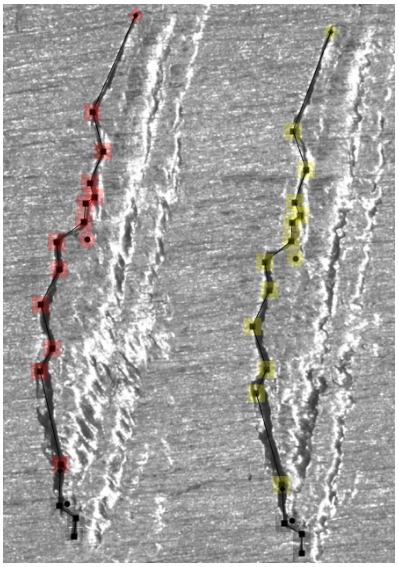


Fig. 4: Matching toolmarks annotated using a polyline and fitting with a transformation matrix.

used to describe the toolmark edges and matching toolmarks are fitted using transformation matrices. By utilizing the fact that the cylinder locks are an approximately flat surface, the capturing angle is orthogonal to this surface and the distance of the camera is always the same, the possible transformations can be restricted to translations and rotations. In this way, matching patches can simply be extracted by moving a window along the polylines and their matching (transformed) counterparts. Matching toolmarks can either be found on the same lock cylinder, as shown in Figure 4, or on different lock cylinders from the same linked case. Since all cylinder locks photographed originate from linked cases, it is guaranteed that for each tool multiple toolmarks exist.

To allow for training and evaluation of different local image similarity approaches, 41,030 and 25,014 images patches were extracted from the training set and the test set, respectively. Additionally, 50,000 matching and 50,000 non-matching image pairs were created to enable comparable evaluations.

IV. Methodology

In this section, both parts of the proposed methodology are decribed. Firstly, the neural network used to compute local

images similarities is shown. Secondly, our two approaches to combine the local image similarities for the retrieval of similar toolmark images are presented.

A. Local Image Similarities

Our proposed neural network with triplet loss is based on the work of Balntas et al. [5]. Similar to siamese networks [6] the network architecture consists of multiple branches with shared weights as shown in Figure 5.

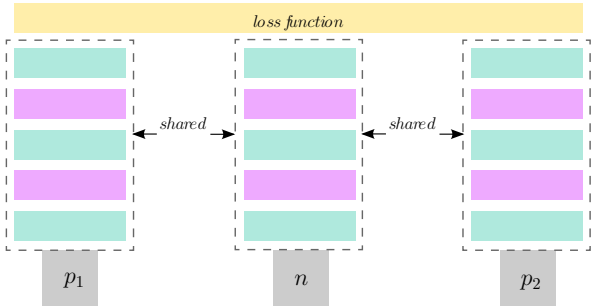


Fig. 5: Triplet architecture

The training is performed by forwarding three input samples, i.e. a triplet, through these equal CNN branches. Each triplet consists of an anchor x_{p_1}, a positive (matching) sample x_{p_2} and a negative (non-matching) sample x_n. The loss function then combines the three outputs $f(x_i)$ and the error is back-propagated. In contrast to other triplet loss functions, like the SoftMax Ratio proposed by Hoffer et al. [9] which only takes one negative distance into account, all three distances between the samples are used:

$$\begin{aligned} \Delta^+ &= \left\| f(x_{p_1}) - f(x_{p_2}) \right\|_2 \\ \Delta_1^- &= \left\| f(x_{p_1}) - f(x_n) \right\|_2 \\ \Delta_2^- &= \left\| f(x_{p_2}) - f(x_n) \right\|_2 \end{aligned} \tag{1}$$

Instead of forcing the distance Δ^+ to be just smaller than Δ_1^-, it is forced to be smaller than $\Delta^* = min(\Delta_1^-, \Delta_2^-)$. In this way negative mining is performed implicitly as illustrated in Figure 6.

The loss function is then defined as:

$$\ell(T) = \left(\frac{e^{\Delta^+}}{e^{\Delta^+} + e^{\Delta^*}} \right)^2 + \left(1 - \frac{e^{\Delta^*}}{e^{\Delta^+} + e^{\Delta^*}} \right)^2 \tag{2}$$

In contrast to the network proposed by Balntas et al., we employ a DenseNet CNN architecture for each of the branches [10].

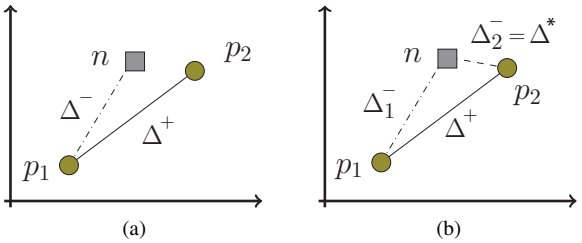


Fig. 6: SoftMax Ratio (a) compared to SoftPN (b) [5].

B. Toolmark Retrieval

In order to retrieve toolmark images, the local image similarities have to be combined to form a similarity measure. For this we use two different approaches.

Firstly, local patches are extracted along the annotated toolmark edges in fixed steps and their features are computed using the neural network described in the previous section. The features are then pairwisely compared using the euclidean distance for each step. The resulting distance between two toolmarks is then computed by summing up these distances and normalizing by length. For toolmarks with different length, the alignment is shifted until a minimal distance is found. The advantage of this approach is, that it is simple and computationally inexpensive. However, it requires an exact annotations since otherwise patches on different parts of the toolmark may be compared against each other. Small variations can lead to accumulated length differences which cannot be compensated by this approach as shown in Figure 7.

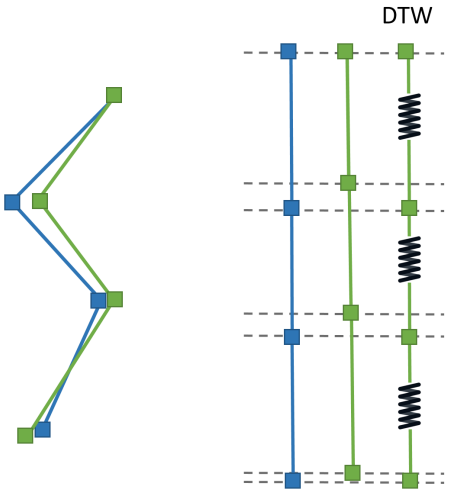


Fig. 7: Matching in fixed step sizes compared to a distance computation using dynamic time warping (DTW).

Secondly, to relax the requirement of a fixed step size, dynamic time warping (DTW) is proposed to allow for a more flexible matching of the local images patches. This way small inaccuracies in the annoation process and the resulting changes in the length of the toolmark segments can be compensated. In Figures 7 on the right the advantage of the DTW approach is visualized.

V. Evaluation

For computing the local image similarities the neural network is trained using the extracted image patches of the training set and evaluated on the test set. Two different approaches were evaluated for the selection of the positive samples.

The first strategy was to define positive samples as matching patches from different locks and different lighting conditions in order to train the network to the high variability of materials and lighting conditions. However, using this

strategy a false positive rate at 95% recall (FPR95) of only about 80% percent is achieved on 100,000 evenly distributed matching and non-matching pairs of patches. This can be explained by the high variability of the presented image patches and human errors in the annoation process.

In order remove the influence of human errors in the annoation process and restrict the variability of the matching patches, as a second strategy, positive samples were defined as patches from just different lighting directions on exactly the same position. This way, an FPR of under 30% is achieved on 100,000 matching and non-matching images pairs which were selected using the same strategy. One interpretation for the remaining false positives is that many patches, mainly from locks made out of shiny materials, are indistinguishable due to the limited dynamic range of the images.

Using this trained network to compute the local image similarities, a cumulative match score of about 70% at a retrieval rate of 20% can be achieved for the toolmark images. In Figure 8 the cumulative match characteristic is depicted for both the approach using a fixed step size and the DTW method. It can be seen, that the fixed approach performs slightly better since in this case the annotations were done precisely. Yet, the DTW approach provides comparable results with the advantage of an added flexibility in the annotation process.

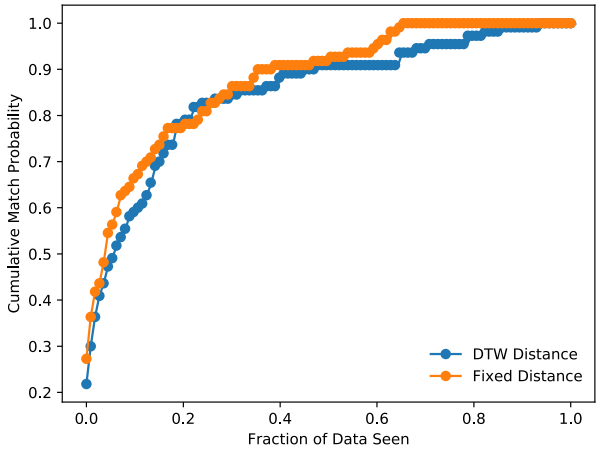


Fig. 8: Cumulative match characteristic on the test set using either a fixed step size or dynamic time warping. The data is retrieved ranked by similarity.

VI. Conclusion

In this paper a two step approach for computing toolmark similarities was presented. Firstly, a neural network using a triplet architecture was proposed to compute local image similarities. Secondly, two approaches for combining local image similarities to form distance scores for toolmark images were shown. The proposed system was evaluated on a toolmark dataset created by photographing cylinder locks from real criminal cases. It was shown that with a probability of more than 70% a matching toolmark is found in case 20% of the images in a database are retrieved. Even though this leaves room for improvement, these results are promising and show that an automated retrieval systems can valuably support the work of forensic experts. For future work, the proposed approaches will be extended to other areas of forensic images; as for instance footwear impressions.

Acknowledgment

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] B. Bachrach, A. Jain, S. Jung, and R. D. Koons, "A Statistical Validation of the Individuality and Repeatability of Striated Tool Marks: Screwdrivers and Tongue and Groove Pliers," *Journal of Forensic Sciences*, vol. 55, no. 2, pp. 348–357, 2010.

[2] 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.

[3] M. Baiker, N. D. Petraco, C. Gambino, R. Pieterman, P. Shenkin, and P. Zoon, "Virtual and simulated striated toolmarks for forensic applications," *Forensic Science International*, vol. 261, pp. 43–52, 2016.

[4] M. Baiker, R. Pieterman, and P. Zoon, "Toolmark variability and quality depending on the fundamental parameters: Angle of attack, toolmark depth and substrate material," *Forensic Science International*, vol. 251, pp. 40–49, 2015.

[5] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, "PN-Net: Conjoined Triple Deep Network for Learning Local Image Descriptors," *ArXiv*, 2016.

[6] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 539–546.

[7] W. Chu, R. M. Thompson, J. Song, and T. V. Vorburger, "Automatic identification of bullet signatures based on consecutive matching striae (CMS) criteria," *Forensic Science International*, vol. 231, no. 1, pp. 137–141, 2013.

[8] L. S. Chumbley, M. D. Morris, M. J. Kreiser, C. Fisher, J. Craft, L. J. Genalo, S. Davis, D. Faden, and J. Kidd, "Validation of tool mark comparisons obtained using a quantitative, comparative, statistical algorithm." *Journal of Forensic Sciences*, vol. 55, no. 4, pp. 953–61, 2010.

[9] E. Hoffer and N. Ailon, "Deep metric learning using Triplet network," *ArXiv*, 2014.

[10] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[11] 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.

[12] N. D. K. Petraco, H. Chan, P. R. D. Forest, P. Diaczuk, C. Gambino, J. Hamby, F. L. Kammerman, W. Brooke, T. A. Kubic, L. Kuo, G. Petillo, E. W. Phelps, A. Pizzola, and D. K. Purcell, "Application of Machine Learning to Toolmarks - Statistically Based Methods for Impression Pattern Comparisons," NCJRS (239048), Tech. Rep., 2012.

[13] R. Sablatnig, "Retrieval of striated toolmarks using convolutional neural networks," *IET Computer Vision*, 2017. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-cvi.2017.0161

[14] 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.

Contributed Session 5

Large Area 3D Human Pose Detection Via Stereo Reconstruction in Panoramic Cameras

Christoph Heindl[1] ,Thomas Pönitz[1], Andreas Pichler[1] and Josef Scharinger[2]

***Abstract*— We propose a novel 3D human pose detector using two panoramic cameras. We show that transforming fisheye perspectives to rectilinear views allows a direct application of two-dimensional deep-learning pose estimation methods, without the explicit need for a costly re-training step to compensate for fisheye image distortions. By utilizing panoramic cameras, our method is capable of accurately estimating human poses over a large field of view. This renders our method suitable for ergonomic analyses and other pose based assessments.**

I. INTRODUCTION AND RELATED WORK

Human pose estimation, characterized as the problem of localizing specific anatomic keypoints, has enjoyed substantial attention in recent years due to the large number of potential applications. It has been shown that keypoint based pose descriptions provide important cues for a variety of tasks such as activity recognition [1] and biomechanical analysis [2].

Inferring pose from a highly articulated, potentially self-occluding, non-rigid body is, in general, a hard and ill-posed problem. Non-optical approaches encompass electromechanical [3] or inertial sensor [4] based suits. Optical methods traditionally applied intrusive active or passive markers [5], [6] for keypoint detection. Early marker-free methods detected body parts in single images [7], [8], [9], [10]. 3D stereo imaging was used to infer human poses from sparse depth-maps [11], [12]. Real-time dense depth cameras greatly simplified the reconstruction task [13], [14], [15], [16] by providing additional metric constraints.

Recently, single and multi-person pose estimation in monocular images made significant progress [17], [18], [19]. Especially the existence of large-scale human annotated datasets [20], [21] accelerated deep learning based approaches [22], [23], [24].

Fisheye lenses have, despite their large field of view, received little attention mostly due to their inherent image distortions which significantly alter the appearance of objects as they move through its line of sight. Among the methods published, researchers have considered single person [25] detection, safe human-robot interactions [26] and head pose tracking [27]. As most of the optical solutions mentioned above assume a pinhole lens model, their results cannot be directly applied to fisheye images.

In this work we propose a 3D pose detector using two fisheye cameras in general position. We apply a deep convolutional network based 2D pose estimator to the input images and reconstruct the corresponding 3D joint coordinates via stereoscopic constraints. We show that proper rectilinear view generation from raw fisheye input images allows us to avoid tedious dataset generation and network training steps. We demonstrate the usefulness of our approach in a challenging 6×6 meter working area, and consider the applicability to ergonomic analysis with respect to accuracy and robustness. To our knowledge we are the first to propose a practical large-scale 3D human pose estimation system based on fisheye lenses.

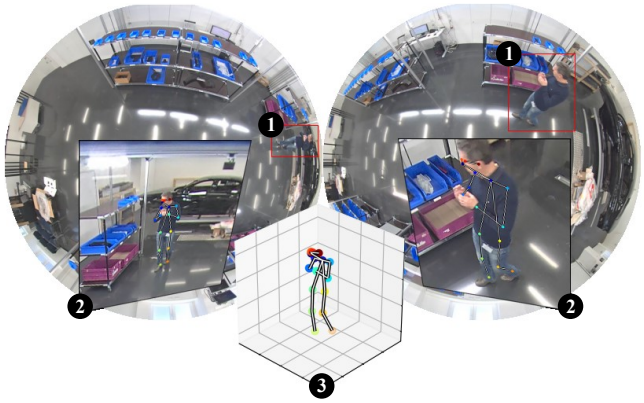


Fig. 1: Overview. From two highly distorted 180° fisheye images coarse human location cues are inferred (1). Regions of interest are transformed into rectilinear views and articulated 2D human poses are then predicted via deeply learned architectures (2). The corresponding 2D joints are then triangulated via stereoscopic constraints to yield accurate 3D body part locations (3) even in the outer edges of a fisheye lens.

II. NOTATION

Throughout this work we use lower-case non-bold characters x to denote scalars, bold-faced lower-case characters $\mathbf{x}$ represent column-vectors and upper-case bold characters $\mathbf{A}$ for matrices. $\mathbf{x}_i$ denotes the i-th element of x, $\mathbf{A}_{ij}$ the i-th row and j-th column of $\mathbf{A}$. For low dimensional vectors we also write $\mathbf{q}_x$ instead of $\mathbf{q}_0$ when it seems practical.

III. LENS MODELS AND PROJECTION FUNCTIONS

A majority of pose estimation methods mentioned in Section I assume that the camera model is sufficiently described by the pinhole camera model. Significant attention has therefore been paid in describing and correcting distortions

[1]Profactor GmbH, Im Stadtgut A2, 4407 Steyr-Gleink, Austria
[2]JKU Department of Computational Perception, Altenbergerstr. 69, 4040 Linz, Austria

that usually appear in ordinary lenses with moderate radial distortion [28], [29]. However, these models are incompatible with wide angle lenses as their projective properties are not well captured.

We provide an brief overview of common lens models and projection functions in the next subsections. For an in depth discussion see [30], [31], [32]. We consider only rotational symmetric lenses and assume that the principal point and the focal length is known. Both parameters can be determined by a number of methods [33], [34], [35].

We consider the characteristics of a lens to be captured in functional relationship between distorted image points on the focal plane and corresponding object points. As illustrated in Figure 2, the radial distance r_d of the optical center to the distorted image point is a function of the object's inclination angle with the z-axis θ, the plane-polar angle around the optical axis ϕ and the focal length f. We consider radially symmetric lenses and therefore assume that the angle ϕ, in contrast to θ, remains unchanged by the lens (see Figure 2).

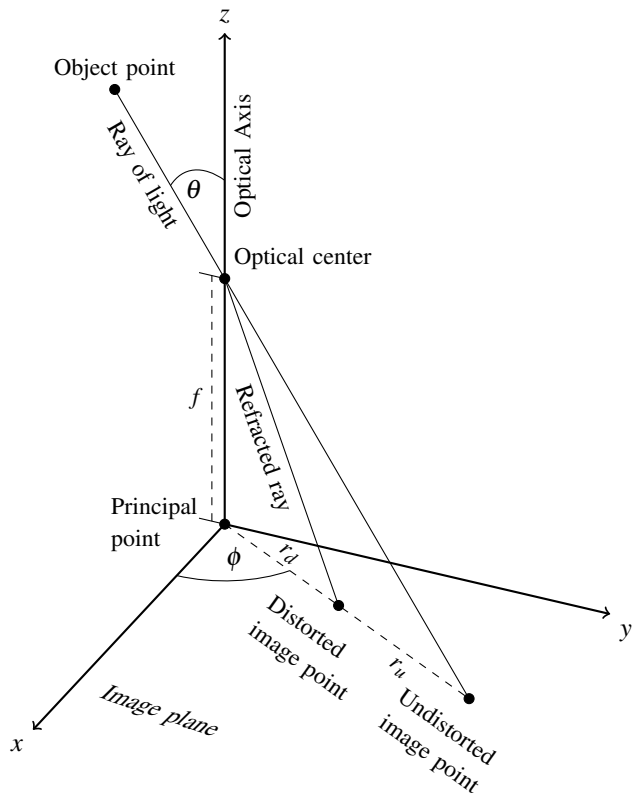


Fig. 2: Illustration of the general projection model and its related parameters. A ray of light emitted from a 3D object passes through the optical center and is potentially refracted due to lens characteristics. The measures r_u and r_d correspond to the ideal (rectilinear) and distorted (actual) radial distances from the principal point. The inclination angle θ measures the angular difference between the ray of light and the optical axis. The angle ϕ denotes the plane-polar angle around the optical axis. The focal length f represents the distance between the optical center and the image plane.

A. Rectilinear projection

The most frequently found projection in computer vision is the pinhole projection. Due to the property that this projection preserves straight lines it is also termed the rectilinear projection. The projection function is given by

$$r_d = f\tan(\theta). \tag{1}$$

For this particular model $r_d = r_u$ and for large field of views the projected image becomes increasingly large and finally infinite when the field of view reaches 180° degrees.

B. Fisheye projections

Similar to rectilinear lenses, fisheye lenses have been manufactured to adhere to optical-engineered projection behavior. The projection functions governing these designs are also known as the classic projection functions and are listed below

$$\text{Equidistant: } r_d = f\theta \tag{2}$$

$$\text{Stereographic: } r_d = 2f\tan(\tfrac{\theta}{2}) \tag{3}$$

$$\text{Equisolid: } r_d = 2f\sin(\tfrac{\theta}{2}) \tag{4}$$

$$\text{Orthographic: } r_d = f\sin(\theta). \tag{5}$$

In Figure 3 we compare radial distorted distances, r_d, as a function of the inclination angle, θ, for all projection models. Note, the monotonicity of the functions that ensures that all angles are mapped to different radial distances.

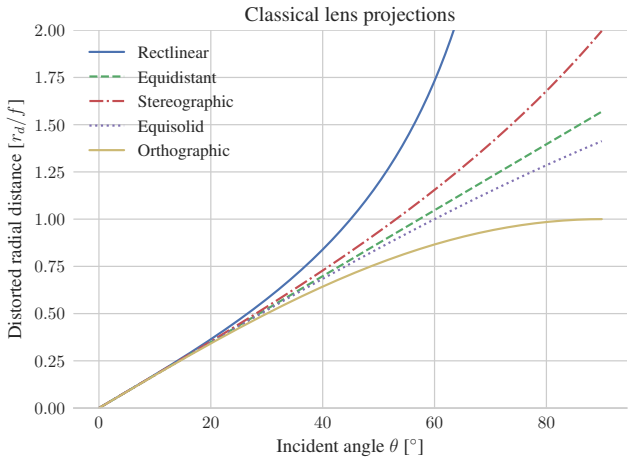


Fig. 3: Plots of classical fisheye projection equations showing normalized radial distorted distances r_d/f as a function of the inclination angle θ. Note the monotonicity of the functions that ensures that all angles are mapped to different radial distances.

Besides the classic projection functions, various other models have been proposed. Most notably are polynomial models [30], a summation of sine terms model [36] and a universal model [35]. Unlike the classical optical-engineered models, these models try to capture a variety of different lenses in a single formula. While the classic projection formulas can be inverted algebraically (i.e determining the θ from r_d) this may not be as easily true for the alternatives.

IV. RECTILINEAR VIEW GENERATION

Generating rectilinear views from fisheye images, as shown in Figure 4, is an important technique in our approach, as our 2D pose detectors assume upright images taken by a pinhole camera model.

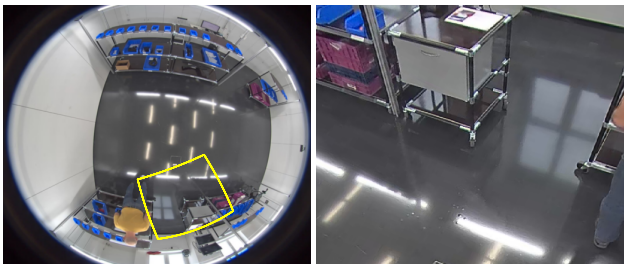


Fig. 4: Upright rectilinear view generation. Fisheye input image on the left, rectilinear view on the right. The bounds of the rectilinear view are shown in yellow in the left image.

Our approach to generate rectilinear views is based on virtual pinhole cameras that share the same origin as the fisheye cameras but are arbitrarily rotated with respect to their physical counterpart. In order to map image points between the artificial pinhole and physical fisheye view the following steps are applied in order.

1) *Un-project* - computes object points from distorted image points using the destination camera model.
2) *Rotate* - rotates object points into the source camera space.
3) *Project* - computes distorted image points from object points using the source camera model.

When applied to all pixels of a destination view, this method leads to efficient lookup maps of corresponding locations with sub-pixel accuracy. The destination image is then formed by interpolating pixels via lookup coordinates. While we are usually interested in mapping fisheye image and rotated pinhole image coordinates, our method works for any pair of camera models.

A. Projection from object space

To compute distorted homogeneous image coordinates $\mathbf{i} = [\mathbf{i}_x, \mathbf{i}_y, 1]^T$ for a Cartesian point $\mathbf{o} = [\mathbf{o}_x, \mathbf{o}_y, \mathbf{o}_z]^T$ in object space, we first compute its spherical coordinates with respect to the camera intrinsic frame

$$\begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \|\mathbf{o}\| \\ \arccos\left(\mathbf{o}_z / \|\mathbf{o}\|\right) \\ \arctan 2(\mathbf{o}_y, \mathbf{o}_x) \end{bmatrix}. \tag{6}$$

Next, we compute r_d from θ according to the lens projection model (see III-B). The vector $[r_d \ \phi]^T$ then denotes the polar coordinates of the distorted image point. The Cartesian coordinates are given by

$$\begin{bmatrix} \mathbf{i}_x \\ \mathbf{i}_y \\ 1 \end{bmatrix} = \begin{bmatrix} r_d \cos\phi \\ r_d \sin\phi \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{c}_x \\ \mathbf{c}_y \\ 1 \end{bmatrix} \tag{7}$$

where $[\mathbf{c}_x \ \mathbf{c}_y \ 1]^T$ denotes the camera principal point. Henceforth, we denote the projection operation $\text{project}(\mathbf{o}; M)\colon \mathbb{R}^3 \to \mathbb{R}^3$ as a functional mapping that takes three-dimensional object points, $\mathbf{o}$, to homogeneous two-dimensional image points $[\mathbf{i}_x, \mathbf{i}_y, 1]^T$ using the lens model M.

B. Reverse projection from image space

Reversing the process outlined in Section IV-A is ambiguous, as depth is lost during projection and all locations along a ray project to the same image coordinates. For our purposes it suffices to un-project image coordinates to points on the unit sphere, as our consideration mainly involves purely rotated cameras. First, we compute the polar coordinate representation for the image point i

$$\mathbf{n} = \mathbf{i} - \mathbf{c} \tag{8}$$

$$\begin{bmatrix} r_d \\ \phi \end{bmatrix} = \begin{bmatrix} \|\mathbf{n}\| \\ \arctan 2(\mathbf{n}_y, \mathbf{n}_x) \end{bmatrix}. \tag{9}$$

We then apply the reverse projection function to obtain θ from r_d according to the lens model. The vector $[r \ \theta \ \phi]^T$ describes a ray from the origin into object space through i in spherical coordinates. Setting $r = 1$, constrains the point to the unit sphere. Converting back to Cartesian coordinates gives

$$\begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \\ \mathbf{o}_z \end{bmatrix} = \begin{bmatrix} r\sin(\theta)\cos\phi \\ r\sin(\theta)\sin\phi \\ r\cos(\theta) \end{bmatrix}. \tag{10}$$

We define the reverse projection operation $\text{unproject}(\mathbf{o}; M)\colon \mathbb{R}^3 \to \mathbb{R}^3$ to be a functional mapping from homogeneous image points $[\mathbf{i}_x, \mathbf{i}_y, 1]^T$, to three-dimensional object points $\mathbf{o}$ using the lens model M.

The general mapping of image points between two purely rotated cameras can now be written as

$$\mathbf{o} = \text{unproject}([\mathbf{i}_x \ \mathbf{i}_y \ 1]^T; M) \tag{11}$$

$$\mathbf{o}' = \mathbf{R}'^T \mathbf{R}\mathbf{o} \tag{12}$$

$$\mathbf{i}' = \text{project}(\mathbf{o}'; M') \tag{13}$$

where M, M' are the respective camera lens models and $\mathbf{R}$, $\mathbf{R}'$ are camera orientations. The mapping is simplified when both lens models are rectilinear, in which case the mapping can be conveniently described by a homography of the following form

$$\mathbf{i}' = \mathbf{K}'\mathbf{R}'^T\mathbf{R}\mathbf{K}^{-1}[\mathbf{i}_x, \mathbf{i}_y, 1]^T \tag{14}$$

where $\mathbf{K}$ and $\mathbf{K}'$ contain camera intrinsics.

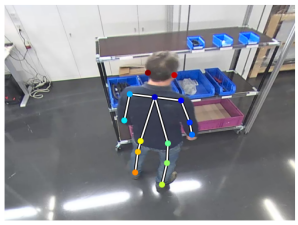
V. HUMAN POSE ESTIMATION

Human body part estimation consists of two steps. First, two-dimensional human poses are estimated in rectilinear views formed from both fisheye cameras. Then, a three-dimensional reconstruction is computed based on stereographic constraints. Both steps are detailed below.

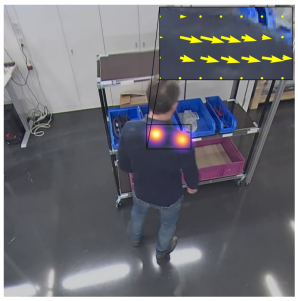
A. 2D Human Pose Estimation

Our 2D pose detection is based on the method of Cao et al. [24], which takes as input a rectilinear image view and outputs anatomic keypoint locations. A neural network is used to predict body joint confidence maps and a set of two dimensional vector fields that encode so called body limb affinities. The basic building block of neural network is a set of convolutional filters that are iteratively applied to previous results in order to refine confidence and affinity maps. In the predicted confidence maps, peak localization is performed to identify potential joint candidates. Then, a graph algorithm guided by affinity maps is used to determine the connectivity. Figure 5 illustrates various stages of the algorithm.

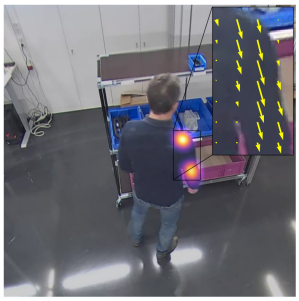
(a) Rectilinear input image.



(b) Computed body joints and limbs.



(c) Joint confidences and part affinities between neck and right shoulder.



(d) Joint confidences and part affinities between right shoulder and right elbow.

Fig. 5: Various stages of the 2D pose estimation algorithm. From a rectilinear view (top-left) a convolutional neural network predicts for every joint and limb a confidence and affinity vector field (bottom-right and-bottom left). A graph based algorithm the constructs a skeleton body model based on these inputs (top-right).

As the detector is directly applied to upright rectilinear views we can use pre-trained network weights and avoid time consuming manual annotation of fisheye images. In order to bootstrap the rectilinear view generation an initial guess of people positions in fisheye images is needed. This can be solved in a number of ways, such as using a people detector [37] or performing foreground segmentation [38]. Once an initial view orientation is known, subsequent rectilinear views can be computed automatically by re-focusing the view on detected 2D human poses.

B. 3D Human Pose Reconstruction

Computing 3D joint coordinates requires at least 2 fisheye images with projections of the same world space joint. Given a rigid transformation between two capture devices, the 3D location can be obtained via triangulation. For static camera setups the rigid transformation can be estimated [39] in a preprocessing step. For non-rigid setups, camera position and scene geometry needs to be inferred simultaneously. This is considered a bundle adjustment problem for which numerous iterative non-linear solutions have been proposed [40], [41].

In either case, the usual linear epipolar constraints for stereo setups do not hold, because fisheye cameras exhibit non-linear projection functions. Therefore, we perform triangulation directly in rectilinear views instead of fisheye images, for which the constraints hold. Without loss of generality, let $\mathbf{P} \in \mathbb{R}^{3\times4}$ be the camera projection matrix of the first rectilinear camera given by

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{R}^T & \mathbf{0} \\ 0 & 1\end{bmatrix}\mathbf{W}^{-1} \tag{15}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the rectilinear projection matrix, $\mathbf{R} \in \mathbb{R}^{3\times3}$ the orientation of rectilinear view with respect to the parental fisheye camera, $\mathbf{W} \in \mathbb{R}^{4\times4}$ is the position and orientation of the fisheye camera in world space and $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the identity matrix. Similarily we define $\mathbf{P}'$ for the second rectilinear view. The simultaneous projection of a homogeneous world point $\mathbf{x}$ in either camera focal plane is given by

$$w\left[\mathbf{i}_x, \mathbf{i}_y, 1\right]^T = \mathbf{P}\mathbf{x} \tag{16}$$

$$w'\left[\mathbf{i}'_x, \mathbf{i}'_y, 1\right]^T = \mathbf{P}'\mathbf{x}. \tag{17}$$

Rewriting Equation 16 line by line and denoting by $\mathbf{p}_i$ the i-th row of $\mathbf{P}$ we get

$$w\mathbf{i}_x = \mathbf{p}_0\mathbf{x} \tag{18}$$

$$w\mathbf{i}_y = \mathbf{p}_1\mathbf{x} \tag{19}$$

$$w = \mathbf{p}_2\mathbf{x}. \tag{20}$$

Then, the Direct Linear Transform (DLT)[42] algorithm is given by eliminating w from Equations 18, 19 via substitution using Equation 20. Rewriting leads to two linear equations in four unknowns of $\mathbf{x} = \begin{bmatrix}x & y & z & w\end{bmatrix}^T$

$$(x'\mathbf{p}_2 - \mathbf{p}_0)\mathbf{x} = 0 \tag{21}$$

$$(y'\mathbf{p}_2 - \mathbf{p}_1)\mathbf{x} = 0. \tag{22}$$

Two views then yield the four equations. The system of equations may be written in matrix form as $\mathbf{A}\mathbf{x} = \mathbf{0}$ and solved for $\mathbf{x}$ in multiple ways [43]. The DLT algorithm allows us to compute 3D point coordinates for corresponding image coordinates in two rectilinear views. Applying it to two-dimensional joint correspondences leads to three-dimensional reconstruction of body parts. Figure 6 shows several successful reconstructions.

VI. EVALUATION AND RESULTS

The setup we use for evaluation covers a 6×6 meter working area with two fisheye cameras of type Axis M3007-PV mounted to the ceiling at height of 3 meters. The baseline between the cameras is roughly 1.5 meters. The simulated working area consists of several shelves that often

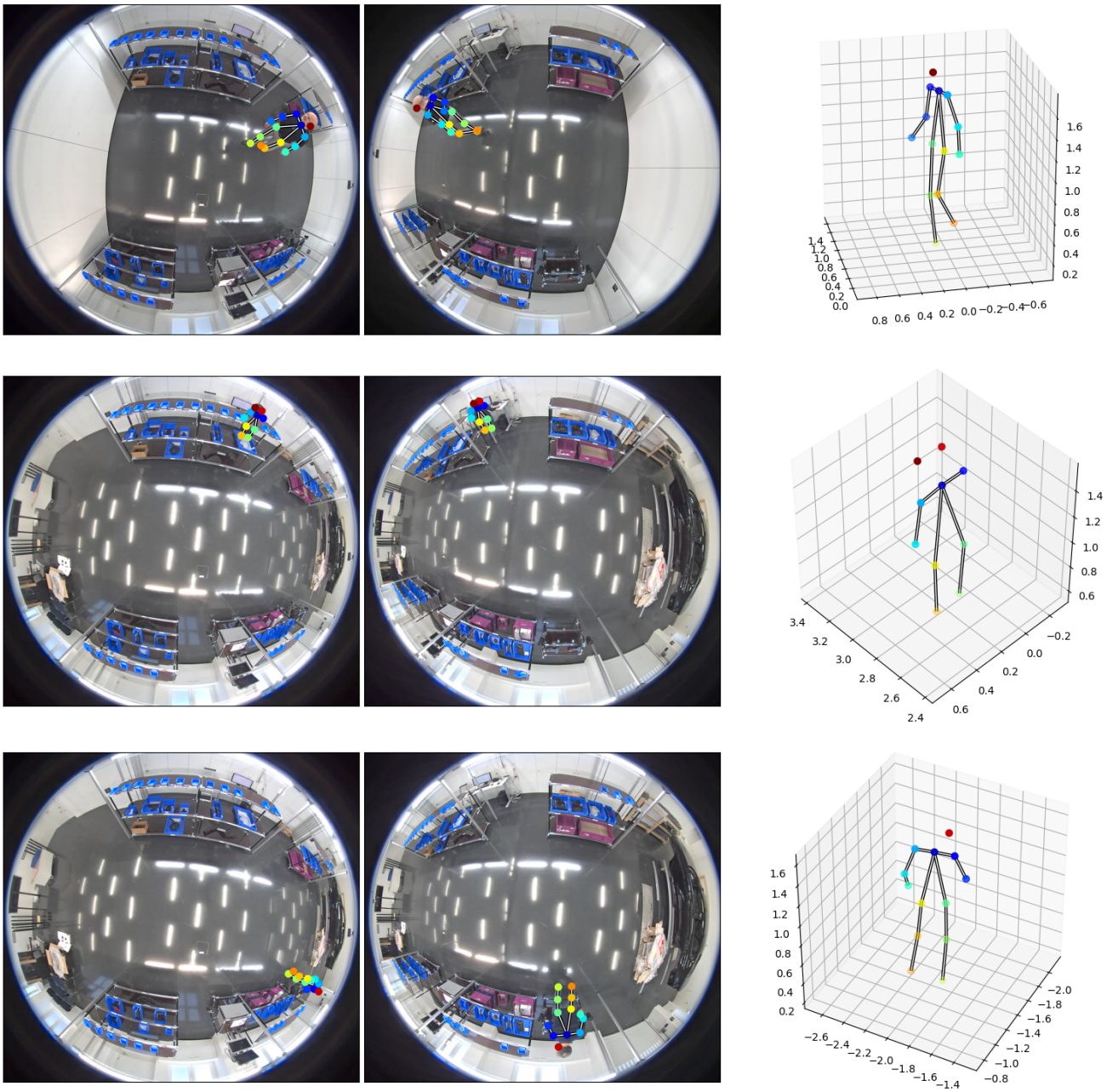


Fig. 6: Examples of 3D stereo reconstruction of human body joints in fisheye images. Left/middle: input images and superimposed detected two dimensional body joints. Right: Three dimensional metric body model.

cause partial body occlusions. We captured raw RGB video from both cameras at rate of 12 FPS at a resolution of 2592×1944 over a period 4 weeks producing a total of 20 hours material. The video data contains 4 different people performing common assembly tasks.

The fisheye cameras have been intrinsically calibrated using the method described in [33]. We obtained the extrinsic calibration for each camera separately by using an external tracking device[1], whose tracking targets can be automatically detected in images. By capturing a set of 3D and corresponding distorted 2D image correspondences in rectilinear views, we solve for the unknown pose $\mathbf{W}$ using a iterative scheme [44].

We assess the accuracy of the calibration and rectilinear view generation by measuring lengths of known objects in reconstructed stereo scenes. For better readability we split the field of view of the fisheye camera into disjoint rings corresponding to increasing radial distortions. The results are shown in Table I.

We trained the 2D pose estimation algorithm on the

[1]HTC Vive `https://www.vive.com/eu/`

	Target length (m)	Measured length (m)
Central area	1.5	1.49 ±0.01
Outer area	1.5	1.52 ±0.018
Lens edge area	1.5	1.56 ±0.035

TABLE I: Measurement errors incurred by the stereo setup inaccuracies. For better comprehensibility we split the fish-eye field of view into three concentric rings that mark central (low-error), outer area (mid-error) and edge area (high-error). Shown are target lengths as well as upper/lower limits over multiple measurements.

COCO[21] dataset, which defines 18 body joints and 17 limbs (see Figure 7). Since the accuracy of the 2D pose estimation has already been studied elsewhere[24], [45], we concentrate on evaluating the quality of the 3D reconstruction. We validate the 3D reconstruction by accumulating 3D limb lengths over video segments grouped by individual persons. Ideally, limb lengths are stationary. However, due to stereo setup imprecision and fluctuations in 2D detection we observe varying limb lengths as shown in Figure 8. Bear in mind that the errors are mostly introduced when people move around in the lens edge area. Figure 9 shows the relative reconstruction frequencies of each limb.

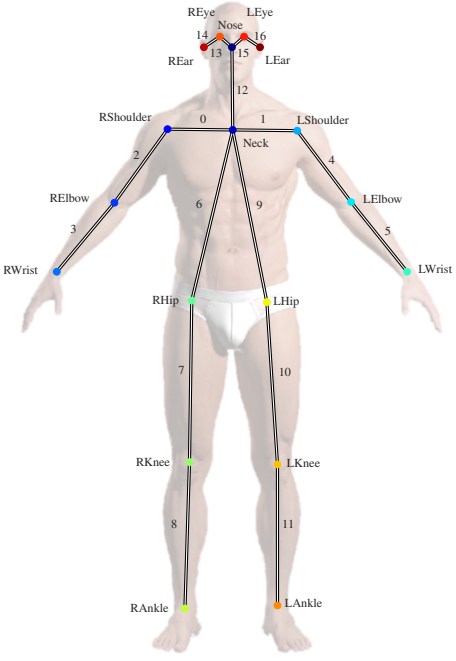


Fig. 7: Joint names and limb indices for the COCO[21] model.

We ran the evaluation on a workstation with an Intel i7-7700 3.6 GHz, 16 GB RAM, and a NVIDIA GeForce GTX1060 graphics card with 6 GB memory. Relevant performance metrics for key stages in our algorithm are given in Table II.

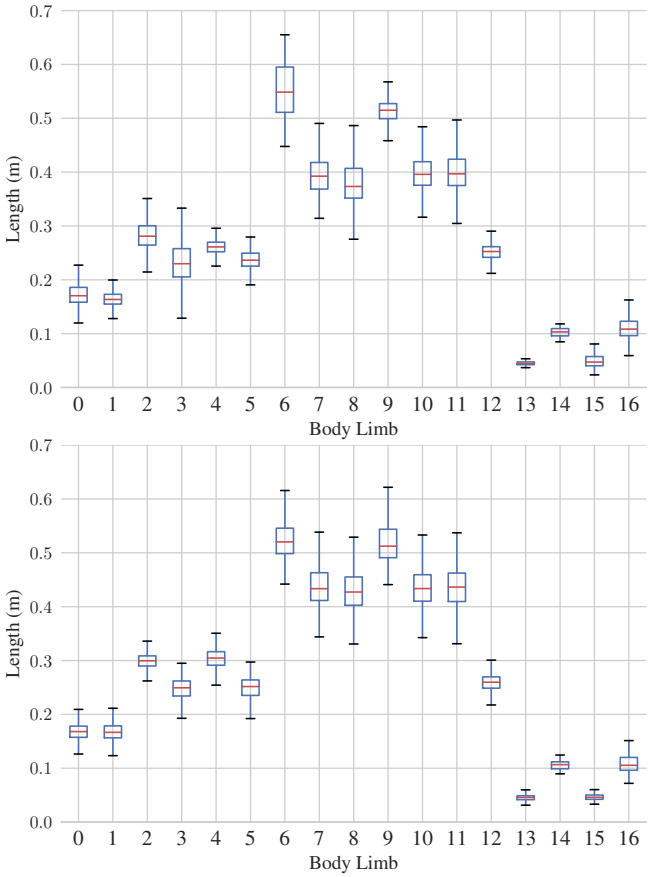


Fig. 8: Metric limb length statistics for two different people over sequence of 10800 frames (15 minutes). Note, the second person has longer legs - he is roughly 5-8 cm taller. The area covered is 6×6 meter and includes serval obstacles that lead to occlusions. Refer to Figure 7 for limb number lookup.

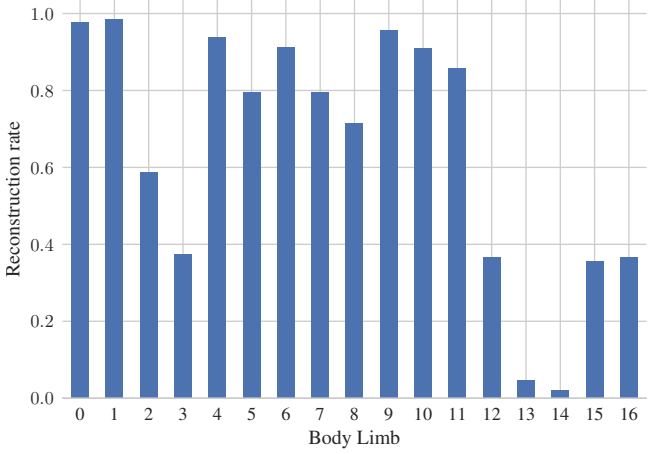


Fig. 9: Reconstruction frequencies of individual limbs over the entire video testing set. Refer to Figure 7 for limb number lookup. Facial features are reconstructed less frequently compared to larger body parts due to visibility constraints.

View Resolution	View Generation (s)	2D Detection (s)	3D Reconstruction (s)
320×320	0.02 ±0.001	0.60 ±0.02	0.01 ±0.001
640×640	0.10 ±0.01	1.80 ±0.05	0.01 ±0.001

TABLE II: Performance timings of key stages in our algorithm. We compare two different resolutions of rectilinear views and note how they affect each stage of the reconstruction pipeline.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel human pose detector that predicts three-dimensional body part locations from two highly distorted fisheye cameras in general position. We demonstrated that the highly enlarged field of view of a fisheye lens is a compelling advantage in reducing hardware complexity. Especially the number of cameras needed to capture the scene can ne reduced and thus many related calibration efforts can be avoided. With regard to pose evaluations, we find that analyses are possible with an accuracy of 2-3 cm over a range of 6x6 meters.

We utilized recent deep-learning based approaches to 2D pose estimation in images and showed that generating artificial rectilinear views avoids the re-training of the neural network. To our knowledge we are the first to consider deep-learning based human pose reconstruction using stereo fisheye lenses. As a matter of fact we observe increasing inaccuracies in 3D reconstruction in the limit of the lens.

In future work we will therefore reconsider the current triangulation method and verify whether additional smoothness constraints can help to reduce the errors. Another point of interest is reduction of runtime complexity, by improving the runtime of the 2D pose detector, in order to achieve real-time performance.

ACKNOWLEDGMENT

This research was supported in part by Lern4MRK (Austrian Ministry for Transport, Innovation and Technology), "FTI Struktur Land Oberoesterreich (2017-2020)", the European Union in cooperation with the State of Upper Austria within the project Investition in Wachstum und Beschäftigung (IWB), as well as AutoScan (FFG, 853416).

REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[2] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[3] M. Motion, "Gypsy motion capture system," 2004.

[4] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: full 6dof human motion tracking using miniature inertial sensors," *Xsens Motion Technologies BV, Tech. Rep*, vol. 1, 2009.

[5] I. Vicon Motion Systems, "Vicon Motion Systems." http://www.vicon.com.

[6] I. Qualisys, "Qualisys.The Swedish motion capture company." http://www.qualisys.com.

[7] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.

[8] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1014–1021, IEEE, 2009.

[9] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 588–595, IEEE, 2013.

[10] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013.

[11] H.-D. Yang and S.-W. Lee, "Reconstruction of 3d human body pose from stereo image sequences based on top-down learning," *Pattern Recognition*, vol. 40, no. 11, pp. 3120–3131, 2007.

[12] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 499–504, IEEE, 2000.

[13] S. Knoop, S. Vacek, and R. Dillmann, "Sensor fusion for 3d human body tracking with an articulated 3d body model," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1686–1691, IEEE, 2006.

[14] Y. Zhu and K. Fujimura, "Constrained optimization for human pose estimation from depth sequences," in *Asian Conference on Computer Vision*, pp. 408–418, Springer, 2007.

[15] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 3108–3113, IEEE, 2010.

[16] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1297–1304, Ieee, 2011.

[17] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1913–1921, 2015.

[18] U. Iqbal, M. Garbade, and J. Gall, "Pose for action-action for pose," *arXiv preprint arXiv:1603.04037*, 2016.

[19] U. Iqbal, A. Milan, and J. Gall, "Pose-track: Joint multi-person pose estimation and tracking," *CoRR*, vol. abs/1611.07727, 2016.

[20] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pp. 3686–3693, 2014.

[21] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[23] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," *CoRR*, vol. abs/1312.4659, 2013.

[24] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, vol. 1, p. 7, 2017.

[25] K. Otsuka, S. Araki, K. Ishizuka, M. Fujimoto, M. Heinrich, and J. Yamato, "A realtime multimodal system for analyzing group meetings by combining face pose tracking and speaker diarization," in *Proceedings of the 10th international conference on Multimodal interfaces*, pp. 257–264, ACM, 2008.

[26] E. Cervera, N. Garcia-Aracil, E. Martinez, L. Nomdedeu, and A. P. Del Pobil, "Safety for a robot arm moving amidst humans by using panoramic vision," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 2183–2188, IEEE, 2008.

[27] R. Stiefelhagen, J. Yang, and A. Waibel, "Simultaneous tracking of head poses in a panoramic view," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 3, pp. 722–725, IEEE, 2000.

[28] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[29] D. C. Brown, "Decentering distortion of lenses," *Photogrammetric Engineering and Remote Sensing*, 1966.

[30] A. Basu and S. Licardie, "Alternative models for fish-eye lenses," *Pattern recognition letters*, vol. 16, no. 4, pp. 433–441, 1995.
[31] D. Schneider, E. Schwalbe, and H.-G. Maas, "Validation of geometric models for fisheye lenses," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 3, pp. 259–266, 2009.
[32] C. Hughes, P. Denny, E. Jones, and M. Glavin, "Accuracy of fish-eye lens models," *Applied optics*, vol. 49, no. 17, pp. 3338–3347, 2010.
[33] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.
[34] S. Shah and J. Aggarwal, "Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation," *Pattern Recognition*, vol. 29, no. 11, pp. 1775–1788, 1996.
[35] D. B. Gennery, "Generalized camera calibration including fish-eye lenses," *International Journal of Computer Vision*, vol. 68, no. 3, pp. 239–266, 2006.
[36] T. J. Herbert, "Calibration of fisheye lenses by inversion of area projections," *Applied optics*, vol. 25, no. 12, pp. 1875–1876, 1986.
[37] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *arXiv preprint arXiv:1703.07402*, 2017.
[38] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.
[39] S. Abraham and W. Förstner, "Fish-eye-stereo calibration and epipolar rectification," *ISPRS Journal of photogrammetry and remote sensing*, vol. 59, no. 5, pp. 278–288, 2005.
[40] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
[41] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-d vision: from images to geometric models*, vol. 26. Springer Science & Business Media, 2012.
[42] R. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pp. 761–764, IEEE, 1992.
[43] R. I. Hartley and P. Sturm, "Triangulation," *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.
[44] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
[45] G. Ning and Z. He, "Dual path networks for multi-person human pose estimation," *arXiv preprint arXiv:1710.10192*, 2017.

Vision-based Autonomous Feeding Robot

Matthias Schörghuber[1,4], Marco Wallner[1], Roland Jung[2], Martin Humenberger[3], Margrit Gelautz[4]

***Abstract*— This paper tackles the problem of vision-based indoor navigation for robotic platforms. Contrary to methods using adaptions of the infrastructure (e.g. magnets, rails), vision-based methods try to use natural landmarks for localization. However, this imposes the challenge of robustly establishing correspondences between query images and the natural environment which can further be used for pose estimation. We propose a monocular and stereo VSLAM algorithm which is able to, first, generate a map of the target environment and, second, use this map to robustly localize a robot. Our hybrid VSLAM approach is able to utilize map points from the previously generated map to (i) increase robustness of its local mapping against challenging situations such as rapid movements, dominant rotations, motion blur or inappropriate exposure time, and to (ii) continuously assess the quality of the local map. We evaluated our approach in a real-world environment as well as using public benchmark datasets. The results show that our hybrid approach improves the performance in comparison to VSLAM without an offline map.**

I. INTRODUCTION

In order to perform autonomous navigation, robot platforms need to be able to robustly and reliably localize themselves and track their position within their operation area. Good examples are commercially available robot platforms fulfilling logistic tasks, e.g. within hospitals or warehouses. For localization, these systems rely on magnetic markers, rails or other infrastructure-based guidance systems. To continuously track the pose between such markers, many robot platforms perform dead reckoning approaches such as wheel odometry. The target robot platform in this paper, namely Wasserbauer's "Butler Gold" feeding robot (shown in Fig. 1), currently uses a very similar approach for autonomous navigation. Even if this and related approaches perform well in target environments, they require costly adaptions of the infrastructure and thus only work within a very well-defined area or even only along well-defined paths.

Since, on the one hand, the application fields of robots are not limited to industrial environments where necessary adaptions can be made, and on the other hand, the costs and efforts of these installations should be reduced, alternative approaches are investigated. The robotics community suggests to use cameras (mounted on the robot) for navigation since they provide rich information about the environment and are easy to install in comparison to other technologies (a survey can be found in [10]). Following this idea, in this paper, we present a visual navigation system, especially designed for autonomous robot indoor navigation. The goal is to robustly localize and track the robot's position within a certain area using passive cameras only. We excluded active light emitting technologies to not interfere with the environment and to enable a possible extension for outdoor usage. In the nature of the application, the vision system has to operate in a challenging environment as dynamic objects (moving cows) are present and the structure (feed, tools, constructions) changes from mission to mission. Furthermore, the system has to be robust against a variety of environmental conditions such as dirt, dust, moisture, lighting or occlusions.

Fig. 1: Target robot platform for feeding with front-facing stereo camera system. Right image: © Wasserbauer

II. RELATED WORK

Similar to many other navigation tasks, for visual localization and pose estimation, certain landmarks are needed. We roughly differentiate between artificial and natural landmarks. Artificial landmarks, such as QR-Codes, are well defined and need to be placed manually. Natural landmarks, such as significant corners or well-textured areas in the image, need to be identified automatically using proper feature extraction methods. In this work, we focus on visual navigation using natural landmarks, since our target is a general approach where the environment does not need to be adapted. The problem of visual localization using natural landmarks is addressed in several ways. Structure-from-Motion (SfM, e.g. [8]) uses multiple images to estimate their positions and to reconstruct the captured environment. Image-based localization (e.g. [11]) uses the resulting maps to estimate the pose of query images. While applications

The research leading to these results has received funding from the Austrian Ministry for Transport, Innovation and Technology (BMVIT) within the ICT of the Future Programme of the Austrian Research Promotion Agency (FFG) under grant agreement no. 849909 (FarmDrive) and Industrienahe Dissertation Programme under grant agreement no. 848518 (AVIS).

[1]Matthias Schörghuber and Marco Wallner are with the Austrian Institute of Technology {matthias.schoerghuber,marco.wallner}@ait.ac.at

[2]Roland Jung is with the Alpen-Adria Universität Klagenfurt roland.jung@aau.at

[3]Martin Humenberger is with Naver Labs Europe martin.humenberger@naverlabs.com

[4]Matthias Schörghuber and Margrit Gelautz are with the Vienna University of Technology margrit.gelautz@tuwien.ac.at

of the mentioned approaches are often found in large-scale and offline localization tasks where memory consumption and processing time play minor roles, in robotics these two issues are critical. Addressing these challenges, important and relevant methods for visual pose estimation such as visual odometry (VO, [6], [4], [15]), visual inertial odometry (VIO, [12], [1]), and visual simultaneous localization and mapping (VSLAM, [9], [5]) were introduced. We differentiate VO from VSLAM by the property that VO does not implement global map optimization or loop-closure, i.e., the process of recognizing that a place was visited before to reduce drift. Therefore, VO algorithms typically maintain only a local map and "forget" about the past. While VO may exhibit more drift than VSLAM, it is computationally more efficient. VIO additionally uses data from inertial measurement units (IMUs) to combine measures from inertial sensors (gyroscope, accelerometer) with visual information. A recent in-depth overview of SLAM discussing common architectures, history, the present, and future is presented in [3] and [14].

Image-based localization has its strength in absolute localization and VO/VSLAM in relative pose estimation, yet an approach which robustly combines them is still missing. In this paper, we propose an approach to overcome this problem. Recently maplab [12] was published where this problem is addressed as well. The authors perform VIO locally and simultaneously fuse the estimated relative pose with absolute localizations within the existing map. In contrast, our approach tracks and uses the existing map points directly.

We present a VSLAM algorithm which is able to combine robust geo-referenced global offline maps, previously generated using VSLAM (in this paper we use the presented algorithm) or SfM, with local online maps which are generated during run-time.

III. Visual Navigation

We propose a visual navigation system which allows to localize and track the pose of the feeding robot within the operational area. To be robust against the changing environment, we perform an initial mapping process where a map is first generated using our VSLAM algorithm and in a second step, registered onto a floor plan (to enable absolute localization). This map is then used in subsequent missions as reference and referred to as offline map. However, in the nature of the application, the existing map is outdated due to the dynamic environment as camera occlusion, structural changes or moving objects can occur. Leveraging the combination of using the initial offline map and performing online mapping, we aim to (i) operate in a global coordinate system defined by the offline map, (ii) allow flexible movements and robustness against structural changes by performing online mapping and (iii) suppress map degeneration on challenging scenes by simultaneous validation of the online mapping with the existing offline map.

The vision pipeline for pose estimation, map generation and localization is inspired by modern SLAM systems (see Section II). It uses a feature based approach with the map consisting of keyframes, 3D points and their 2D observations within the keyframes. The most similar approach to our implementation is ORB_SLAM [9]. It presents a VSLAM system with a localization only mode (i.e. without mapping), however, neither loading or storing nor usage of pre-existing maps is possible.

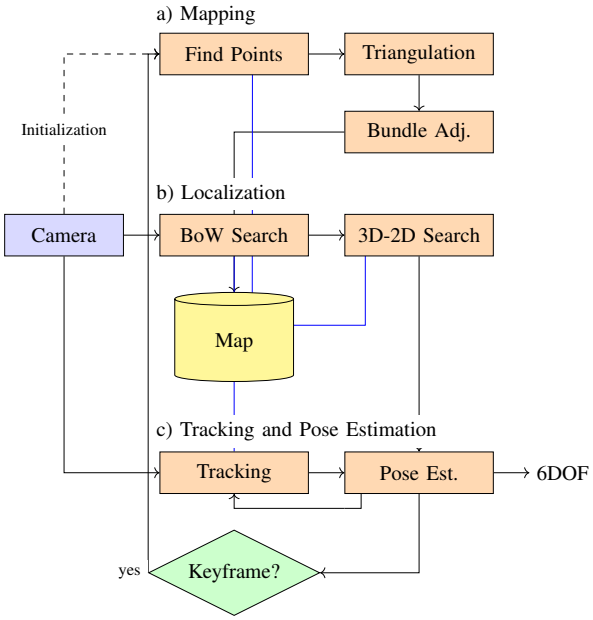


Fig. 2: Algorithmic overview of the main operational modes of the VSLAM implementation.

The core blocks of our implementation are a) Mapping, b) Localization and c) Tracking and Pose Estimation as shown in Fig. 2 and described in the following.

a) Mapping: The mapping process aims to find and triangulate new reliable 3D points between the existing map and new input images. In the *Find Points* procedure, a correspondence search of image features not associated with 3D points is performed with features of neighboring keyframes. If no map is present during initialization of the algorithm, the correspondence search is performed either with the stereo image pair or with two consecutive images with sufficient translation in between in case of monocular input. The new 3D point candidates are triangulated and validated using geometric properties. Finally, *Bundle Adjustment* is performed with keyframes and 3D points affected by the new measurements. We distinguish between 3D points originating from the online and offline map within this optimization step. The 3D points of the existing global map are assumed to be fixed and reliable, thus they are higher prioritized. This prioritization prevents a drift or degeneration between the online and the offline map, as new measurements are aligned to the coordinate system defined by the offline map even in cases where online map points are dominant. To prevent map degeneration and undefined behaviour in subsequent missions, the offline map is not updated with new measurements in our current application.

b) Localization: Localization aims to find the pose of a query image within a given map. Our implementation performs a two-step process. First, the most similar keyframes are retrieved with a visual vocabulary based approach. In such a visual vocabulary the image descriptors are clustered into words of a pre-defined vocabulary (bag of words [7]). This methodology allows to efficiently compare images on this reduced set of words. Image similarity is determined by the presence of shared words of the vocabulary between images.

The second step performs a correspondence search between 3D points visible in the retrieved keyframe and 2D image points of the query image. Finally, the position and orientation of the query image can be estimated using multiple correspondences with non-linear optimization by minimizing the reprojection error.

c) Tracking and Pose Estimation: This module aims to estimate the pose of the current image by tracking an existing pose (from last tracks or successful localization) within the map. First, using the extrapolated input pose, 3D points within the camera frustum are projected onto the image plane. Second, using these estimates, guided matching is performed to establish 2D-3D correspondences between the 3D points and the 2D keypoints of the current image. The new pose is estimated by minimization of the reprojection error between the 3D point projections and corresponding 2D image feature observations.

Similar to the mapping process, we distinguish between 3D points originating from the online and offline map and prioritize the measurements within the pose estimation step accordingly. Furthermore, the tracking of 3D points from the offline map allows to determine map consistency and prevents degeneration in challenging scenes. If no 3D points of the offline map can be found, the robot either left the operational area (which can be validated using previous poses) or the camera provides no valid information for global localization (e.g. occlusions or close-up scenes). As a consequence, the algorithm tries to relocalize within the offline map. If this is not successful, a navigation error occurred and a recovery mode has to be triggered. Based on the number of visible 3D points originating from the offline map and total number of 3D points, it is decided when new keyframes shall be generated and added to the online map.

The VSLAM performance strongly depends on the robot movement. If fast movements are present, the algorithm has to deal with motion blur and large unobserved gaps between frames. This especially occurs when motions with a mainly rotational component are present. Furthermore, with monocular input, the triangulation requires a translational component, which defines the theoretical accuracy of the 3D reconstruction. The proposed approach, using an offline map as basis, increases robustness to such challenging robot movements because even if triangulation of new 3D points fails, 3D points from the offline map can be tracked. In the same way, if tracking is lost, the robot can relocalize itself within the offline map. Furthermore, the number of visible offline points can be used as quality indicator of the online map.

In order to provide the poses within the application specific world coordinate frame, the map is registered onto a geo-referenced 2D floor plan. This rigid transformation is estimated using manually selected correspondences between map points and salient features on the floor plan such as wall corners.

IV. Tests and Evaluation

We performed two experiments to evaluate our VSLAM system. First, in a real-world environment with recordings acquired in a cowshed and second, with a benchmark dataset from the community.

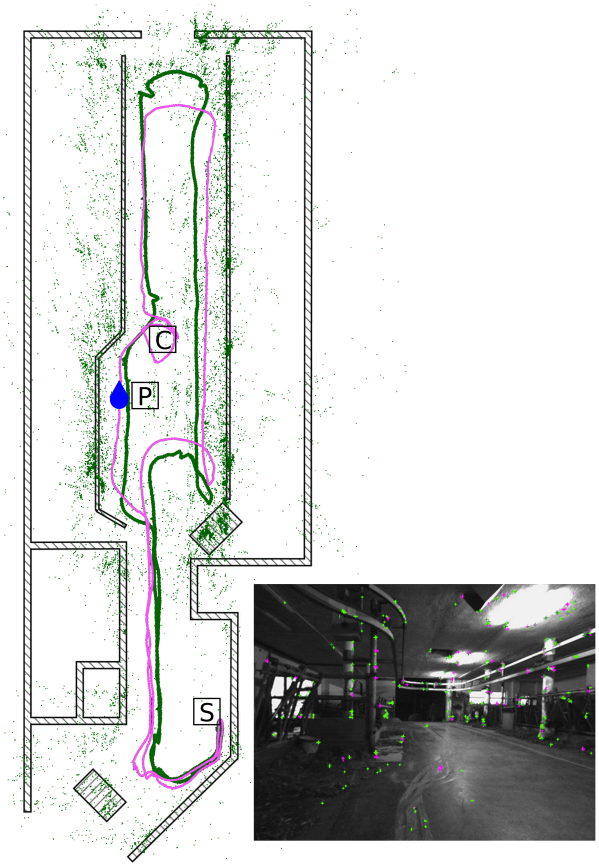
For the real-world test, we used one recording[1] to generate an offline map and a second one for combining offline maps with online mapping. The trajectories are slightly different, as can be seen in the purple and green trajectory in Fig. 3 (a). The green trajectory represents the trajectory of the initial mapping run after registration onto the floor plan. The green dots indicate the locations of the 3D points generated during this run. The purple line represents the trajectory of the second recording using the 3D map from the green mission as the offline map. Both missions have a common starting and end point indicated at the position *S* within Fig. 3 (a), hence the double line in the entry path of the cowshed.

The camera image in Fig. 3 (b) shows the 2D projections of the observed 3D points as seen from the position *P* marked by a blue circle and an arrow indicating the viewing direction in Fig. 3 (a). The green points visible within the camera image in (b) correspond to 3D points originating from the offline map (and correspond to the green dots in 3 (a)). The purple points represent newly generated 3D points in the online map.

The consistency of the purple trajectory with the floor plan confirms that the algorithm was able to perform its mission within the application specific world coordinate frame. Furthermore, with the online mapping, the system was able to estimate the pose even when the robot moved differently in comparison to the data available from the offline map; this is especially visible at the circular movement at position C in Fig. 3 (a).

Since no ground truth was available for quantitative analysis of the cowshed data, we used the *EuRoC* [2] dataset to evaluate our assumptions of increased robustness and accuracy that can be achieved by additionally including 3D points of an offline map into the core computations of the algorithm. This dataset was created with a multi-rotor unmanned aerial vehicle (UAV) equipped with a stereo camera sensor providing 20 frames per second. The dataset consists of sequences captured in a machine hall (*mh*) scenario with positional ground truth and two laboratories (*v1*, *v2*) equipped with a motion capture system providing 6 DOF ground truth. The sequences were recorded with varying difficulties. A higher difficulty implies faster translational

[1]The actual robot control was performed using its navigation system.



(a) (b)

Fig. 3: (a) 3D map from the VSLAM system registered onto the cowshed floor plan; (green) 3D points and trajectory of initial map generation; (purple) trajectory of a subsequent mission. (b) 3D points of map projected onto camera image captured from the blue position *P* in (a). Green dots represent observed 3D points from the initially generated offline map, and purple dots the online generated 3D points.

and rotational movements of the UAV, operation in a low-textured environment or exposure differences due to auto shutter effects. We processed the sequences with our VSLAM algorithm, which is able to provide pose information at camera frame rate, and estimated the accuracy of each sequence by comparing it with the ground truth using the root mean square errors of absolute trajectory error (ATE) and relative pose error (RPE). The latter consists of a translational (RPE_t) and a rotational (RPE_r) component. These metrics are defined in [13].

The results are shown in TABLE I. It can be seen, as expected, that the trajectory could be estimated more precisely on less complex sequences *mh_01*, *mh_02*, *v1_01* and *v2_01* than on the more difficult *mh_04*, *mh_05*, *v1_02* and *v2_02* sequences.

In a second run, marked with an asterisk (*), the VSLAM uses as offline map the map generated from the corresponding underlined sequence (e.g. *mh_02** used the map from *mh_01* as offline map as indicated in TABLE I). In all cases, the VSLAM was able to improve the pose estimation compared to runs without an offline map, as can be seen from the bold figures in TABLE I.

Sequence	$ATE[m]$	Sequence	$ATE[m]$	$RPE_t[m]$	$RPE_r[deg]$
mh_01	0.177	v1_01	0.138	0.056	0.984
mh_02	0.126	v1_02	0.187	0.182	2.488
mh_02*	**0.121**	v1_02*	**0.124**	**0.145**	**1.97**
mh_04	0.484	v2_01	0.103	0.080	4.501
mh_05	0.389	v2_02	0.273	0.243	9.385
mh_05*	**0.290**	v2_02*	**0.184**	**0.224**	**8.77**

TABLE I: Comparison of estimated VSLAM trajectories to a ground truth with the ATE and RPE metric [13] on the *EuRoC* [2] dataset. Sequences marked with Asterisk (*) use an offline map generated from the corresponding underlined sequence. All values represent the average over multiple executions.

V. CONCLUSIONS

In this paper, we presented a novel VSLAM approach for navigation of an autonomous feeding robot. We applied our algorithm to recordings from a cowshed and showed the successful operation within this application domain. For quantitative evaluation of the approach, we used a community established dataset where ground truth data is available. The results show that our approach of combining an offline map with online mapping successfully improves the accuracy of the pose estimation.

The increased accuracy is achieved by incorporating reliable 3D points from the offline map in order to robustly triangulate new accurate 3D points. Consequently, the new 3D points are inherently aligned to the mission specific world coordinate frame. Furthermore, in the case of lost pose tracking, the robot can relocalize itself in the offline map and continue its mission.

Possible future work is the analysis of the robustness towards application specific environmental conditions such as dust along with further improvements to this aspect and methods for updating the offline map.

REFERENCES

[1] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 298–304.

[2] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[4] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[5] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM Large-scale direct monocular SLAM," in *European Conference on Computer Vision*, 2014, pp. 834–849.

[6] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.
[7] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
[8] P. Moulon, P. Monasse, and R. Marlet, "OpenMVG. an open multiple view geometry library." https://github.com/openMVG/openMVG.
[9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
[10] W. Sakpere, M. Adeyeye-Oshin, and N. B. Mlitwa, "A state-of-the-art survey of indoor positioning and navigation systems and technologies," *South African Computer Journal*, vol. 29, no. 3, pp. 145–197, 2017.
[11] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient effective prioritized matching for large-scale image-based localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1744–1756, 2017.
[12] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *arXiv preprint arXiv:1711.10250*, 2018.
[13] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.
[14] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017.
[15] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *IEEE International Conference on Computer Vision*, 2017, pp. 3923–3931.

A workflow for 3D model reconstruction from multi-view depth acquisitions of dynamic scenes*

Christian Kapeller[1], Braulio Sespede[1], Matej Nezveda[2], Matthias Labschütz[3], Simon Flöry[3], Florian Seitner[2] and Margrit Gelautz[1]

***Abstract*— We propose a workflow for generating high-quality 3D models of dynamic scenes for the film and entertainment industry. Our 3D scanning system comprises multiple synchronized 3D Measurement Units that incorporate stereo analysis. We give an overview of the involved algorithms for stereo matching, point cloud registration, semi-automatic post-processing and mesh generation, and demonstrate selected steps of their implementation. The computed 3D models will provide content for 360 degree video production and 3D augmented reality applications.**

I. Introduction

The extensive usage of computer graphics in the film and advertisement industries has drastically raised the demand for high-quality 3D model generation from real film content. Related applications include the creation of realistic special effects and upcoming trends towards immersive 3D augmented reality, virtual reality and 360 degree video content production. While several passive and active depth measurement sensors have become available over the last decade, currently available multi-view 3D scanning solutions have various shortcomings. For example, ReCap [3] from Autodesk processes multiple images of an object from various viewing positions for 3D object reconstruction, however, it appears not to be capable of coping with dynamic scenes. Active devices based on structured light or time-of-flight techniques often place severe restrictions on the scene environment (for example, indoor locations only) or have significant limitations on the size of the scanned area or the number of viewpoints. Furthermore, the scanning information is usually only available as point cloud data, while professional film post-production workflows require accurate 3D mesh models.

We propose to overcome some of these limitations by the development of a scalable and cost-efficient workflow for accurate 3D scanning of film sets and the creation of corresponding high-quality 3D models. The system comprises two stages: online content acquisition (i.e., production side) and offline quality enhancement and mesh generation (i.e., post-production side). On the production side several stereo-based 3D Measurement Units (3*DMU*) are used to acquire a scene from multiple viewpoints in a highly synchronized way. On the post-production side the captured views of all 3*DMU*s enable the conversion of the recorded 2D footage into high-quality mesh models. The suggested workflow is explained in more detail in Section III.

*This work has been funded by the Austrian Research Promotion Agency (FFG) and the Austrian Ministry BMVIT under the program ICT of the Future (project "Precise3D", grant no. 6905496)

[1] Institute of Visual Computing and Human-Centered Technology, Vienna University of Technology, Favoritenstrasse 9-11/193-06, 1040 Vienna, Austria; {braulio.sespede, christian.kapeller, margrit.gelautz}@tuwien.ac.at

[2] emotion3D GmbH, Gartengasse 21/3, 1050 Vienna, Austria; {nezveda, seitner}@emotion3d.tv

[3] Rechenraum e.U., Stutterheimstraße 16-18/2/3/20a, 1150 Vienna, Austria; {matthias.labschuetz, simon.floery}@rechenraum.com

II. Related Work

Single camera calibration relates to the process of estimating the intrinsic camera parameters such as focal length, principal point offset and skew factor. When calibrating multiple cameras, the extrinsic parameters describing the geometric relationship between these cameras are essential. Popular approaches for single camera calibration either rely on 2D objects such as planes [26] or 1D objects such as a wand with multiple collinear points [27]. For multiple camera calibration, a pairwise stereo calibration procedure can be applied [11]. In contrast, the authors of [21] present a fully automatic multiple camera calibration procedure.

Depth reconstruction from stereo image pairs has been studied extensively in the literature, with algorithms traditionally being grouped into local, global and semi-global approaches. Local methods proposed during the last years often rely on adaptive support weight techniques [12] and cost-volume filtering [13]. Very recently, deep convolutional neural networks have been successfully used for depth estimation. For example, Kendall et al. [15] propose an end-to-end deep learning system for computing disparity maps. The 4Dviews [10] system uses an iterative patch sweep method.

Rough alignment of two or more 3D measurements of a scene by so-called global registration is typically based on geometric or topological features. In recent work, the Super4PCS algorithm [17], which matches approximately planar four-point configurations, has emerged as a fast yet robust method. In local registration (fine-tuning the results of global registration) the iterative closest point (*ICP*) algorithm [4] and its variants [18] are well established techniques formulated as optimization problems in a least-squares sense.

Sumner et al. [20] propose a method to interactively and non-rigidly deform meshes. They reduce the complexity of the problem of deforming a mesh by deforming a graph structure, termed the embedded deformation (*ED*) graph. Non-rigid deformation of a mesh can be used to perform non-rigid alignment of two meshes, as shown in the Fusion4D [9] pipeline, which reconstructs a non-rigid scene in real-time. In

Fusion4D, aligned data is stored and fused in a 3D voxel grid. The popular marching cubes [16] and dual contouring [14] algorithms provide the means to transform volumetric to mesh data.

Some further literature will be addressed in the context of the method description in the next section.

III. Proposed Approach

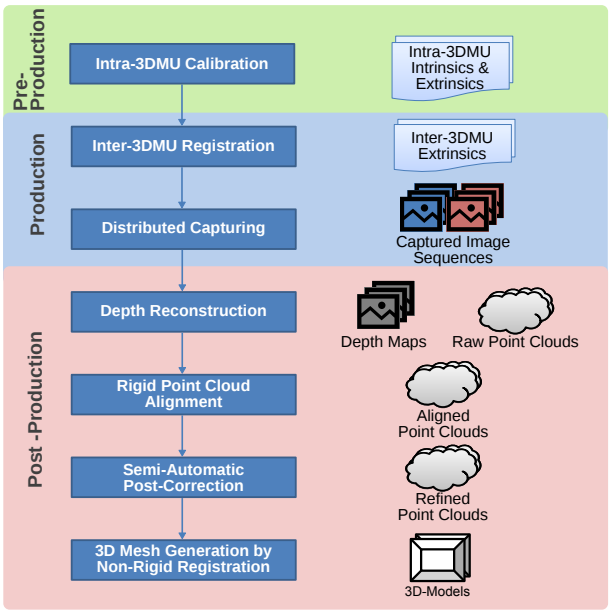


Fig. 1. Processing pipeline of the proposed workflow.

Our reconstruction framework, as shown in Figure 1, is designed to capture scenes with multiple 3*DMU*s. From the acquired data we reconstruct depth information individually per 3*DMU*. In a subsequent step, registration errors among the individual units are minimized using rigid point cloud alignment. After that, the results are refined with our semi-automatic post-correction tool. In the last step we generate mesh models by means of non-rigid alignment.

In the following, we give a detailed overview of the individual steps involved in our processing pipeline.

A. 3*DMU* Setup and Calibration

We illustrate our sensor setup comprising two 3*DMU*s as depicted in Figure 2. Each 3*DMU* is composed of two industrial-grade XIMEA cameras [24]. Each camera contains a 2/3 inch RGB sensor and is capable to record at a 2464×2056 (5 MPix) resolution in RAW format at a maximum of 60 frames per second. Thus, in total we use four cameras c_i to observe the scene. Here, $i \in \{0,1,2,3\}$ is the camera index, where 0 represents the left camera of $3DMU_1$, 1 the right camera of $3DMU_1$, 2 the left camera of $3DMU_2$ and 3 the right camera of $3DMU_2$. Captured image data is recorded by a controlling computer with USB3 interface. We achieve accurate synchronization of the cameras with a hardware interface operating in master/slave mode.

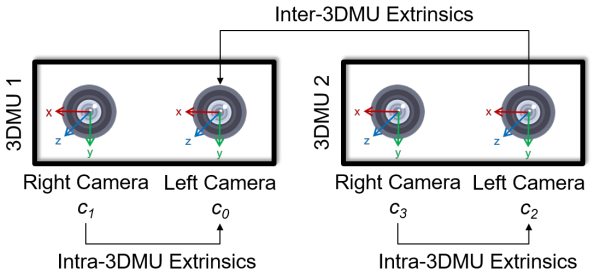


Fig. 2. Illustration of our sensor setup. Please note that cameras are facing the scene, thus left and right are flipped. A detailed description can be found in the text.

For calibration, we perform (i) intra-calibration to obtain the intrinsic and extrinsic parameters for each 3*DMU* individually and (ii) inter-calibration to obtain extrinsic parameters between multiple 3*DMU*s. For both we use the approach of Zhang [26] to compute calibration parameters based on a circle grid calibration pattern. For the former calibration parameters are obtained in a pre-production step as these parameters stay fixed for a 3*DMU*. In particular, we compute calibration matrices and distortion coefficients for each camera individually, and rotations and translations for c_1 to c_0 and c_3 to c_2. For the latter, calibration parameters are obtained during production as soon as the position of each 3*DMU* is fixed. In particular, the inter-3*DMU* extrinsics encode the rotation and translation from c_2 to c_0.

B. Depth Reconstruction

We reconstruct dense disparity maps from each individual 3*DMU* by means of stereo matching. Our method is based on the cost-volume filtering algorithm employed in [19]. For each pixel in the left image of a stereo pair with size $w \times h$, the costs of matching a corresponding pixel in the right image are computed using the Census dissimilarity metric [25] in a given disparity search range d. This gives rise to a cost-volume of dimensions $w \times h \times d$. Subsequently, the cost-volume is aggregated (i.e., filtered) using the fast edge preserving permeability filter [8]. The selected disparity values are those with minimum costs in the cost-volume. This step yields a raw disparity map. Finally, unreliable and occluded pixels are eliminated by means of a consistency check. Disparity values are compared with those of corresponding pixels in a second disparity map that was computed in the same fashion but with the right image as reference. Pixels that disagree by more than 1 disparity count are dropped. In order to improve the quality of the results and run-time, the disparity map computation is embedded into a hierarchical matching scheme. For each stereo image pair a Gaussian image pyramid with $k = 3$ layers is built. Stereo matching is performed first on the coarsest layer l_2. Based on this initial disparity map, an offset map is computed that guides disparity estimation on the next finer layer l_1. The process is then repeated for the finest layer l_0 of the Gaussian pyramid. In image sequences slightest changes of capturing conditions introduce temporal noise in the computed disparity maps. We

address this issue by temporal filtering in the cost-volume following the approach in [7]. Using the intra- and inter-3*DMU* calibration information we project the disparity maps into point clouds in a common coordinate system.

C. Semi-automatic Post-Correction

Depth estimation errors, e.g. due to untextured regions, make additional steps of correction necessary to improve the quality of the multi-view point cloud reconstruction for high-quality 3D film content generation. With this goal in mind, we develop a semi-automatic multi-view 2D-plus-depth visualization and correction tool. To effectively reduce noise and outliers from the resulting point cloud, we implement the multi-view consistency filter of [23] and outlier removal algorithm of [6]. The tool also allows the user to make spatio-temporally coherent local corrections on the disparity maps in a joint-view manner. When it comes to local corrections, we perform binary segmentation operations on 2D video [5] to extract areas we are interested in correcting through user-assisted scribbles in key-frames.

D. Point Cloud Registration and Mesh Generation

In order to fine-tune global 3*DMU* registration, we perform local rigid pairwise alignment of the raw point clouds for a specific frame (that differs by rigid-body transforms only). We employ the method of Pottmann et al. [18], representing an unknown rigid-body transform by its linear velocity vector field and minimizing approximations of a point cloud's squared distance function. In case extrinsic calibration information of the individual 3*DMU*s is not available, a global alignment step based on 4PCS [1] precedes the local alignment step.

We improve a reference frame by fusing each successive frame with the reference. The reference frame is stored as a signed distance field, a volume data-set. For alignment we first extract the reference mesh (the zero-crossing surface in the signed distance volume) via marching cubes. Non-rigid alignment is performed to align the next frame as a point cloud to the reference frame. We follow the approach of Sumner et al. [20] by generating an ED graph for the point cloud to be aligned. Skinning the vertices to the ED graph uses the weighting presented in Fusion4D [9], as this results in a smoother deformation. A least-squares problem that minimizes the distance of the point clouds while maintaining approximative local rigidity through regularization is formulated based on the linear velocity vector field. To integrate the aligned point cloud into the reference volume, we update the signed distance of each voxel grid point by projection onto an implicit point set surface [2]. In a final step, we extract a mesh from the merged signed-distance field via marching cubes.

IV. Experiments

We have conducted a capturing session with two 3*DMU* prototypes and acquired multiple data-sets with real world calibration and image sequence data. Results of the initial processing steps are shown in Figure 3. For these test scenes we created spherical objects of known size in order to assess the accuracy of the 3D reconstruction. Using the intra-3*DMU* calibration information the image sequences were rectified (Figure 3 (a,b) and (e,f)). The computed stereo-derived disparity maps (Figure 3 (c,g)) were afterwards projected into point clouds (Figure 3 (d,h)). We currently capture with a frame rate of 25 fps. By measuring the sizes of the captured spheres in the reconstructed point clouds, we have determined that the diameters of the two reconstructed spheres deviate from their true sizes by less than 6 percent for both 3*DMU*s. A screenshot of the implemented post-correction tool is shown in Figure 4. The displayed scribbles are projected to 3D space using calibration and disparity information and then back-projected to other views, reducing user interactions and aiding the user in the 3D segmentation process.

V. Summary and future work

We have proposed a multi-view depth reconstruction system in the context of a film and media production workflow. The approach aims at the generation of high-quality and temporally coherent 3D meshes from dynamic real world scenes. The implemented processing chain includes algorithms for stereo-based depth reconstruction and geometric 3D data processing, in conjunction with semi-automatic post-processing techniques for further quality enhancement.

The next step of the ongoing project will be to evaluate the results of the individual components and the system as a whole in terms of accuracy and efficiency. Besides quantitative measurements on scene targets of pre-defined size and shape, we plan to evaluate the system by rendering novel views into the viewpoint of an additional reference 3DMU and computing error maps [22]. A complementary qualitative user study will connect objective assessment and subjective judgement.

References

[1] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 85:1–85:10, 2008.

[2] M. Alexa and A. Adamson, "On normals and projection operators for surfaces defined by point sets," in *SPBG'04 Symposium on Point - Based Graphics 2004*, 2004, pp. 149–155.

[3] Autodesk, "Recap," accessed: 2018-04-30. [Online]. Available: https://www.autodesk.com/products/recap/overview

[4] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–607.

[5] N. Brosch, A. Hosni, C. Rhemann, and M. Gelautz, "Spatio-temporally coherent interactive video object segmentation via efficient filtering," in *Pattern Recognition*, vol. 7476, 2012, pp. 418–427.

[6] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5556–5565.

[7] C. Çığla and A. Aydın Alatan, "An improved stereo matching algorithm with ground plane and temporal smoothness constraints," in *European Conference on Computer Vision. Workshops and Demonstrations*, 2012, pp. 134–147.

[8] C. Çığla and A. Aydın Alatan, "Information permeability for stereo matching," *Signal Processing: Image Communication*, vol. 28, no. 9, pp. 1072–1088, 2013.

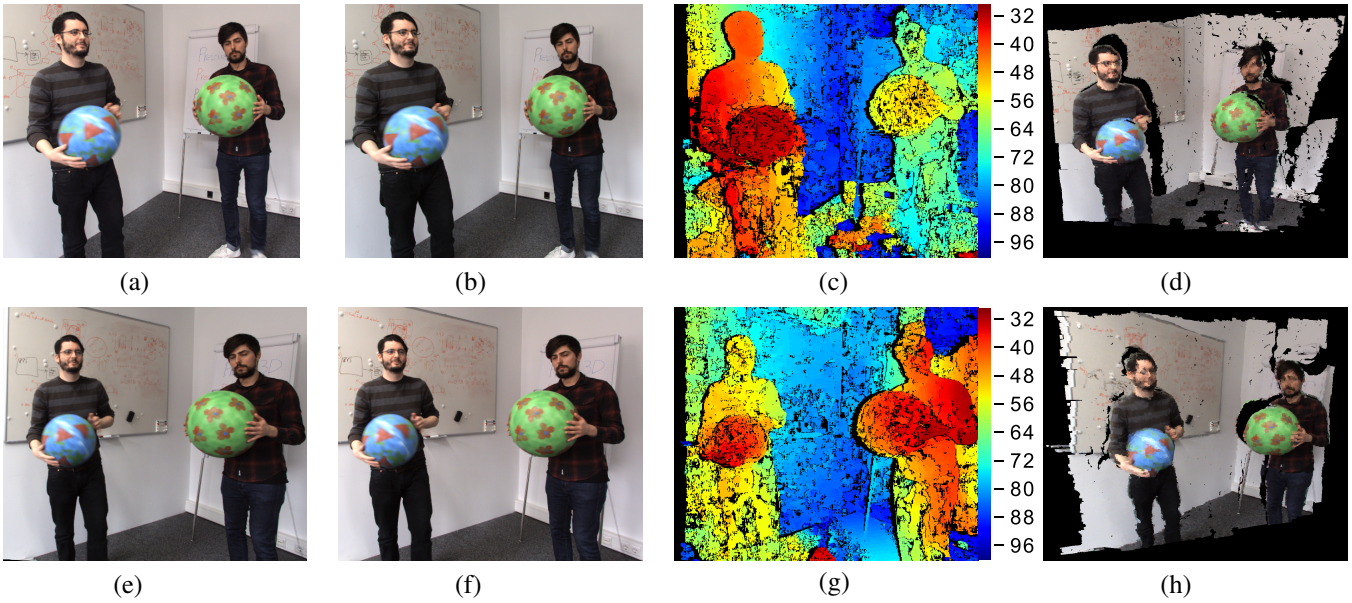


Fig. 3. Data acquired with our experimental setup. The top row shows results acquired by $3DMU_1$, the bottom row those of $3DMU_2$. (a)-(b) and (e)-(f): rectified RGB stereo image pairs; (c) and (g): color-encoded disparity maps, numbers indicate disparity values; (d) and (h) point clouds derived from depth maps.

Fig. 4. User interface of the post-correction tool with scribble placed by the user on a key-frame. The histogram displayed on the left shows the disparities covered by the scribble and supports the segmentation of the selected object and projection of the scribble to other views.

[9] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, *et al.*, "Fusion4D: Real-time performance capture of challenging scenes," *ACM Transactions on Graphics*, vol. 35, no. 4, p. 114, 2016.

[10] T. Ebner, I. Feldmann, S. Renault, O. Schreer, and P. Eisert, "Multi-view reconstruction of dynamic real-world objects and their integration in augmented and virtual reality applications," *Journal of the Society for Information Display*, vol. 25, no. 3, pp. 151–157, 2017.

[11] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[12] A. Hosni, M. Bleyer, and M. Gelautz, "Near real-time stereo with adaptive support weight approaches," in *International Symposium 3D Data Processing, Visualization and Transmission*, 2010, pp. 1–8.

[13] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2013.

[14] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 339–346, 2002.

[15] A. Kendall, H. Martirosyan, S. Dasgupta, and P. Henry, "End-to-End Learning of Geometry and Context for Deep Stereo Regression," in *IEEE International Conference on Computer Vision*, 2017, pp. 66–75.

[16] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[17] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," *Computer Graphics Forum*, vol. 33, no. 5, pp. 205–215, 2014.

[18] H. Pottmann, S. Leopoldseder, and M. Hofer, "Simultaneous registration of multiple views of a 3D object," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, no. 3/A, pp. 265–270, 2002.

[19] F. Seitner, M. Nezveda, M. Gelautz, G. Braun, C. Kapeller, W. Zellinger, and B. Moser, "Trifocal system for high-quality inter-camera mapping and virtual view synthesis," in *International Conference on 3D Imaging*, 2015, pp. 1–8.

[20] R. W. Sumner, J. Schmid, and M. Pauly, "Embedded deformation for shape manipulation," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 80, 2007.

[21] T. Svoboda, D. Martinec, and T. Pajdla, "A convenient multicamera self-calibration for virtual environments," *Presence: Teleoperators & Virtual Environments*, vol. 14, no. 4, pp. 407–422, 2005.

[22] M. Waechter, M. Beljan, S. Fuhrmann, N. Moehrle, J. Kopf, and M. Goesele, "Virtual rephotography," *ACM Transactions on Graphics*, vol. 36, no. 1, pp. 1–11, 2017.

[23] K. Wolff, K. Changil, H. Zimmer, C. Schroers, M. Botsch, O. Sorkine-Hornung, and A. Sorkine-Hornung, "Point cloud noise and outlier removal for image-based 3D reconstruction," in *International Conference on 3D Vision*, 2016, pp. 118–127.

[24] XIMEA GmbH, "Ximea MC050CG-SY product specification brochure," accessed: 2018-02-18. [Online]. Available: https://www.ximea.com/files/brochures/xiC-USB3_1-Sony-CMOS-Pregius-cameras-brochure-HQ.pdf

[25] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European Conference on Computer Vision*, 1994, pp. 151–158.

[26] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[27] Z. Zhang, "Camera calibration with one-dimensional objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 892–899, 2004.

Globally Consistent Dense Real-Time 3D Reconstruction from RGBD Data

Rafael Weilharter[1,2], Fabian Schenk[1], Friedrich Fraundorfer[1]

***Abstract*— In this work, we present a dense 3D reconstruction framework for RGBD data that can handle loop closure and pose updates online. Handling updates online is essential to get a globally consistent 3D reconstruction in real-time. We also introduce fused depth maps for each keyframe that contain the fused depths of all associated frames to greatly increase the speed for model updates. Furthermore, we show how we can use integration and de-integration in a volumetric fusion system to adjust our model to online updated camera poses. We build our system on top of the InfiniTAM framework to generate a model from the semi-dense, keyframe based ORB SLAM2. We extensively evaluate our system on real world and synthetic generated RGBD data regarding tracking accuracy and surface reconstruction.**

I. INTRODUCTION

In recent years, the ubiquity of inexpensive RGBD cameras, pioneered by the Microsoft Kinect, has led to a series of applications for 3D scene reconstruction in areas such as augmented/virtual reality, robotics, gaming and general 3D model estimation. Modern 3D reconstruction systems have to provide a globally consistent model on large scale in real-time. This does not only require a robust camera pose estimation algorithm but also on-the-fly model updates that incorporate loop closures and pose refinements. The robust camera motion estimation process is the main difference between current systems and divides them roughly into three categories.: (i) Iterative closest point (ICP) methods aim to align 3D points but require sufficient 3D structure and a correspondence matching step [7], [10]. Instead of point clouds, (ii) direct methods estimate motion by processing image information directly. Dense [8] and semi-dense [3], [5] variants based on the photo-consistency assumption exist. This makes these approaches especially susceptible to illumination changes and direct methods are typically restricted to small inter-frame motion. (iii) Feature-based methods extract features, match correspondences and estimate motion by minimizing the reprojection error [4], [9]. The extracted features are more robust to illumination changes than the direct methods based on the photo-consistency assumption and are also suitable for larger inter-frame motions.

Regardless of the applied method, many systems rely solely on frame-to-model tracking to estimate the camera movement in real-time and sequentially build a dense 3D model [10], [7]. While such systems are real-time capable they accumulate significant drift over time and usually cannot correct this drift by revisiting the same place (see Fig. 1). To tackle this problem, more advanced Simultaneous Localization and Mapping (SLAM) systems perform loop closure and pose graph optimization to reduce the drift. However, such an approach is computationally very expensive and often only acquires a semi-dense [5] or sparse map [9]. If in addition a dense 3D model is desired, the SLAM system usually relies on expensive hardware, e.g. the Bundle Fusion system [3] only runs on a combination of an NVIDIA GeForce GTX Titan X and a GTX Titan Black.

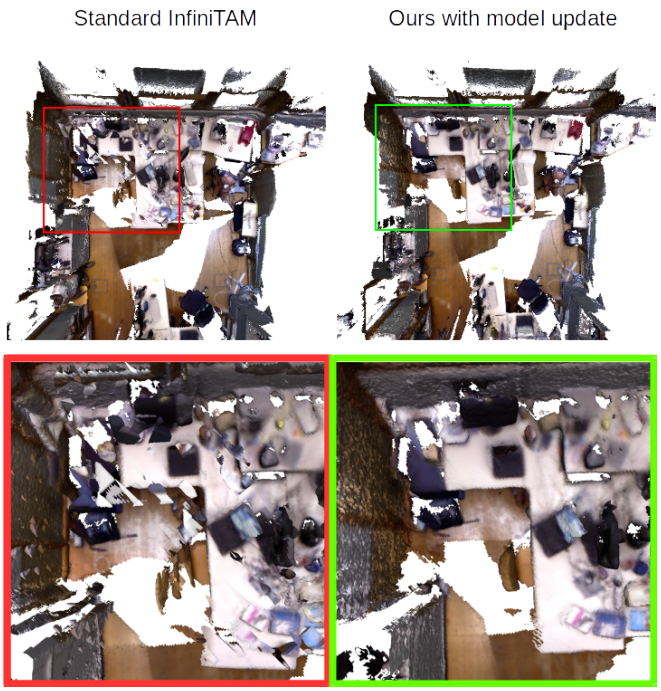


Fig. 1: Reconstructed dense 3D models: No update vs our system with keyframe based depth map fusion and global model update. We can see the effects especially in the upper left corners where a loop closure occurs.

In this paper, we propose a real-time dense 3D reconstruction method that successfully combines the state-of-the-art ORB SLAM2 system [9] with the dense volumetric fusion framework InfiniTAM [7]. To validate our method, we compare the trajectory estimations and surface reconstruction accuracy of several methods on the standard TUM RGBD benchmark dataset and the synthetic ICL NUIM dataset. The

*This work was financed by the KIRAS program (no 850183, CSIS-martScan3D) under supervision of the Austrian Research Promotion Agency (FFG).

[1]Institute of Computer Graphics and Vision (ICG), Graz University of Technology, Styria, Austria

[2]Ludwig Boltzmann Institute for Clinical Forensic Imaging (LBI CFI), Graz, Styria, Austria

rafael.weilharter@student.tugraz.at

{schenk, fraundorfer}@icg.tugraz.at

key contributions presented in this work can be summarized as:

- Implementation of a de-integration method which allows to refine and alter the 3D model online when large changes in the estimated trajectory occur, e.g. in the case of a loop closure detection.
- A global model update which can delete and merge keyframes in retrospect.
- A keyframe based depth fusion, where we fuse information of frames into their respective keyframes instead of integrating them directly into the model. This significantly speeds up the re-integration process required for a global model update.
- An extensive evaluation of both, the trajectory error and the surface reconstruction error on several benchmark data sets

II. RELATED WORK

In the past decades, 3D reconstruction has been a very active research field when we focus on work that utilizes RGBD data. Kerl at al. [8] proposed a combination of photometric and geometric error minimization in their visual SLAM system. To reduce the acquired drift, they use keyframes: Every new frame is at first matched to the latest keyframe and as long as there is not too much difference, i.e. the camera has not moved to far, no drift is accumulated. Furthermore, when revisiting a previously seen region old keyframes can enforce additional constraints on the pose graph, also known as loop closures. Although this system is capable of real-time performance on a CPU, it can not do so at a full resolution of 640×480.

ORB SLAM 2 is a state-of-the-art feature-based SLAM system for monocular, stereo and RGBD cameras that runs in real-time on a single CPU. It estimates the camera motion by minimizing the reprojection error and implements loop closure, relocalization, map reuse and bundle adjustment for pose refinement. All these methods show promising results regarding runtime and tracking accuracy, but either generate no [8] or only a sparse global 3D model [9], [4].

One of the first systems to achieve a dense 3D reconstruction in real-time, by exploiting the massively parallel processors on the GPU, was the KinectFusion [10]. The KinectFusion system estimates the current sensor pose with an ICP algorithm and integrates the aquired data via truncated signed distance function (TSDF). In order to achieve real-time performance, the algorithms for both tracking and mapping are fully parallelized. However, the KinectFusion system, which spawned several re-implementations and further works, lacks the scalability for larger scenes due to memory issues (addressing and/or lack thereof).

In order to tackle the problem of a large memory footprint, research on sparse volumetric representations [11], [12] has sprouted. These works successfully use either octrees or hash tables to refer to allocated memory blocks efficiently. One of the works that is able to achieve a very high framerate while reconstructing a dense 3D model is InfiniTAM [7]. It models the 3D world as voxel blocks using a TSDF representation. In order to reduce memory usage, only the scene parts inside the truncation band, i.e. the voxels close to a surface, are usually represented densely in an $8 \times 8 \times 8$ block. A hash table manages these voxel blocks to guarantee a constant lookup time (in case of no collisions).

ElasticFusion [14] reconstructs the 3D world as a number of circular surfels that correspond to the surfaces in the real world. Managing and processing these surfels requires a strong graphics card and is not capable of large-scale reconstructions. Dai et al. proposed BundleFusion [3] that models the world with a voxel block hashing framework [11]. They always optimize over all previously seen frames, generating a globally consistent model. To handle this large amount of data, they utilize two strong graphics cards.

In contrast to [14], [3], we present a framework that generates a globally consistent, dense model in real-time on a consumer graphics card. While pose estimation runs on CPU, the 3D model is generated and processed on the GPU. Most volumetric fusion frameworks do not allow to correct the model [7], [11], [10], e.g. after loop closure (see Fig. 1). We propose several extensions to [7] that enable on-the-fly corrections to generate a globally consistent model in real-time.

III. GLOBALLY CONSISTENT DENSE 3D RECONSTRUCTION

In this paper, we present a real-time 3D reconstruction framework that works with RGBD data. We combine the accurate trajectory estimation of ORB SLAM2 [9] with the volumetric fusion implementation from InfiniTAM v2 [7]. We add novel techniques to InfiniTAM in order to support global model updates that arise from e.g. loop closure. This includes a de-integration method, a global model update step and fusion into the depth maps of keyframes to enable fast real-time processing.

A. Camera Model

We receive an RGBD frame at each time step t that consists of an RGB image I_t and a depth map D_t. We expect I_t and D_t to be aligned and synchronized, such that at a certain pixel position $\vec{x} = (x,y)^\top$ the RGB values are given as $I_t(\vec{x})$ and the corresponding depth as $D_t(\vec{x})$. Then the homogeneous 3D point $\vec{X} = (X,Y,Z,1)^\top$ in the respective camera coordinate system can be computed from $\vec{x}$ and the corresponding depth $Z = D_t(\vec{x})$ with the inverse projection π^{-1}:

$$\pi^{-1}(\vec{x}) = \vec{X} = \left(\frac{x - c_x}{f_x} Z, \frac{y - c_y}{f_y} Z, Z, 1\right)^\top, \tag{1}$$

where f_x, f_y are the focal lengths and c_x, c_y are the principal points. Similarly, the pixel position $\vec{x}$ can be recovered from $\vec{X}$ as:

$$\pi(\vec{X}) = \vec{x} = \left(\frac{X \cdot f_x}{Z} + c_x, \frac{Y \cdot f_y}{Z} + c_y\right)^\top. \tag{2}$$

B. Rigid Body Motion and Warping Function

We restrict the motion between frames to a rigid body motion $g \in SE(3)$. A common representation for g is as

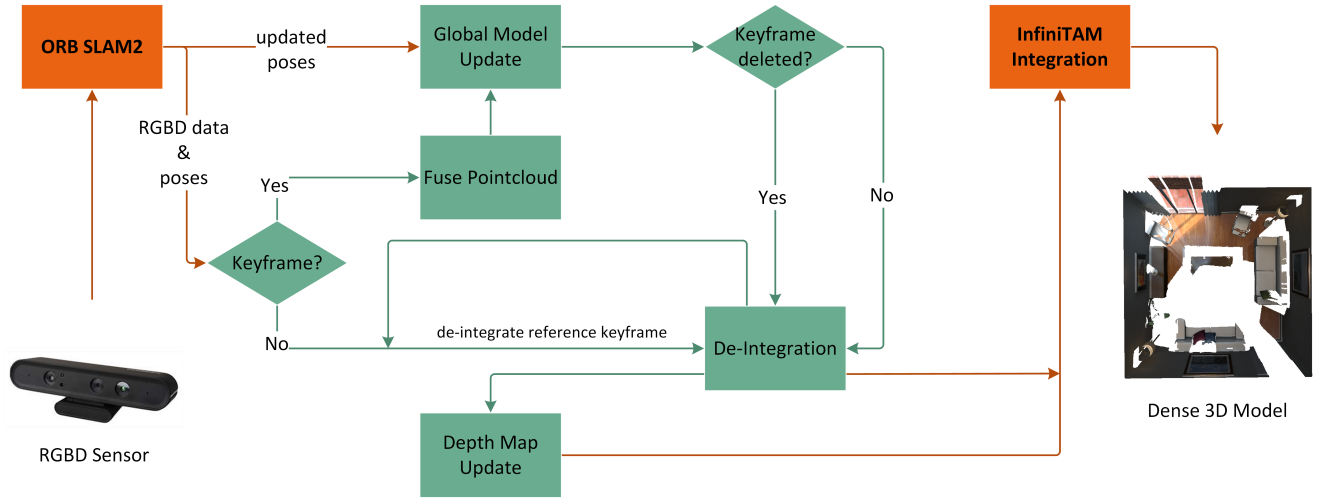


Fig. 2: Our system takes as input the estimated poses from ORB SLAM2 and the RGBD data from the sensor. If the current frame is not a keyframe, we update the corresponding depth map. Otherwise we fuse its depth image with the pointcloud and update the global model. In case that a keyframe has been deleted, we de-integrate it and fuse its information into the next best (closest) keyframe. If not, we de-integrate the frame with its old pose and re-integrate it with its new pose.

transformation matrix T comprising a 3×3 rotation matrix $R\in SO(3)$ and a 3×1 translation vector t:

$$T_{4\times4}=\begin{bmatrix}R_{3\times3} & t_{3\times1}\\ 0 & 1\end{bmatrix}. \tag{3}$$

The transformation of a point $\vec{X}$ under motion g can be written as:

$$g(\vec{X})=\vec{X}'=T_{4\times4}\vec{X}. \tag{4}$$

The rigid motion g only has 6 degrees of freedom, thus T with its 12 parameters is over-parametrized. We use a minimal representation as twist coordinates ξ defined by the Lie algebra $se(3)$ associated with the group $SE(3)$. From the 6-vector ξ the transformation matrix T can be recovered by the matrix exponential $T=exp(\xi)$.

We define the full warping function τ that re-projects $\vec{x}$ from frame j with depth $D_j(\vec{x})$ to frame i under the transformation matrix T_{ij} as:

$$\vec{x}'=\tau(\xi_{ij},\vec{x},D_j(\vec{x}))=\pi(T_{ij}\pi^{-1}(\vec{x},D_j(\vec{x}))). \tag{5}$$

C. Combining ORB SLAM2 and InfiniTAM

Fig. 2 shows the interaction between ORB SLAM2 and InfiniTAM. Note that our contributions are depicted in green. We feed the RGBD data (either acquired live via sensor, or from a dataset) into the ORB SLAM2 system and receive the estimated poses for every frame. Then we update our depth maps and adjust the global model if necessary before we integrate the new information into the volumetric representation of InfiniTAM. We explain our depth map and global model updates in detail in the following sections.

D. Model Updates by Re-Integration

We reconstruct our scene geometry by sequentially fusing RGBD data into the TSDF representation. The TSDF is defined as:

$$T(\vec{V})=max\left(-1,min\left(1,\frac{D_t(\pi(\vec{V}))-Z}{\mu}\right)\right) \quad , \tag{6}$$

where $\vec{V}=(X,Y,Z)^\top$ is a voxel given by its center coordinates, $\pi(\vec{V})$ computes the projection of the voxel onto the depth image, while $D_t(\pi(\vec{V}))$ is the measured depth at the calculated image location. Since $\pi(\vec{V})$ maps the voxel into the camera frame, Z is the distance between the camera and voxel along the optical axis. There are only values between -1 and 1 allowed, corresponding to the distances $-\mu$ and μ respectively (thus truncated). As a result, positive values are assigned to voxels that reside in free space: The closer the voxel is to the surface, the smaller its value. If the voxel lies directly on the surface, its value is set to zero and behind the surface, increasing negative values are assigned.

We adapt the strategy presented by Curless and Levoy [2] and update the TSDF for every new observation i as:

$$\begin{aligned}F_i(\vec{V})&=\frac{W_{i-1}(\vec{V})F_{i-1}(\vec{V})+w_i(\vec{V})T(\vec{V})}{W_{i-1}(\vec{V})+w_i(\vec{V})} \quad ,\\ W_i(\vec{V})&=W_{i-1}(\vec{V})+w_i(\vec{V}) \quad ,\end{aligned} \tag{7}$$

where $W_0(\vec{V})=0$ and the uncertainty weight $w_i(\vec{V})$ is usually set to 1, which results in an averaging of the measured TSDF observations.

For a globally consistent reconstruction we need to be able to alter our model with updated camera poses, e.g. when a loop closure is detected. In order to do so, we need to delete old information from the 3D model. We can de-integrate an observation by reversing the operation of (7):

$$\begin{aligned}F_i(\vec{V})&=\frac{W_{i-1}(\vec{V})F_{i-1}(\vec{V})-w_i(\vec{V})T(\vec{V})}{W_{i-1}(\vec{V})-w_i(\vec{V})} \quad ,\\ W_i(\vec{V})&=W_{i-1}(\vec{V})-w_i(\vec{V}) \quad .\end{aligned} \tag{8}$$

The operations of integrating and de-integrating are symmetric, i.e. one inverts the other. Thus, an observation, if it becomes invalid or updated, can be deleted by de-integrating it from its original pose and re-integrating it with a new pose if necessary.

E. Depth Map Fusion in Keyframes

The idea behind fusing the depth maps of frames into their reference keyframe is to create a system that is able to adapt to global changes within real-time. Without the depth map fusion each frame would have to be re-integrated separately when a model update is induced, while our technique re-integrates only the fused depth maps of keyframes. Since on average only every 10th frame is selected as keyframe, this reduces the amount of operations by a factor of 10. ORB-SLAM2 inserts a keyframe if all of the following conditions are met: (i) more than 20 frames have passed sinice the last global relocalization, (ii) more than 20 frames have passed since the last keyframe insertion or local mapping is idle, (iii) at least 50 keypoints are tracked in the current frame, (iv) more than 10% of keypoints in the current frame are not seen by its reference keyframe. Additionally (for RGBD data), a keyframe is added whenever the number of close keypoints drops below a certain threshold $\tau_t = 100$ and the frame could at least create $\tau_c = 70$ new close keypoints. Fig. 2 depicts our process flow: If a frame is not chosen as keyframe by ORB SLAM2, we fuse its depth map D_c into the depth map of its reference keyframe D_{KF} (see Fig. 3). This update step is closely related to the volumetric fusion integration step presented in (7) :

$$\begin{aligned} D_{KF,i}(\vec{x}') &= \frac{W_{i-1}(\vec{x}')D_{KF,i-1}(\vec{x}') + w_i(\vec{x})Z'}{W_{i-1}(\vec{x}') + w_i} \quad , \\ W_i(\vec{x}') &= W_{i-1}(\vec{x}') + w_i(\vec{x}) \quad , \end{aligned} \tag{9}$$

where $\vec{x}' = \tau(\xi_{ij}, \vec{x}, D_j(\vec{x}))$ is the reprojected pixel position, $Z' = [T_{KF,c}\pi^{-1}(\vec{x}, D_c\vec{x})]_z$ is the z-coordinate of the transformed point, D_c the depth map of the current frame, $T_{KF,c}$ the transformation from current frame to keyframe and the weight $w_i(\vec{x})$ is set equal to 1, which leads to an averaging of the depth values. Please note that we truncate $\vec{x}'$ to always work on integer pixel positions. The difference to the volumetric fusion (7) step is that we update the depth map of the keyframe $D_{KF,i}$ instead of the TSDF values in the model.

In order to not lose any information, we store unfused points in a pointcloud. The pointcloud is represented as a vector, where each entry corresponds to a 3D point, which is transformed into the keyframe coordinates but could not be added to the depth map. Points are not fused into the depth map and added to the pointcloud when either of two conditions arise: (i) The point is transformed out of boundaries variables, i.e. the x and/or y coordinate are negative or larger than the image size or (ii) the depth difference is too large, which can be described as:

$$\left| \frac{1}{D(\vec{x}')} - \frac{1}{Z'} \right| < \Theta_\tau \quad , \tag{10}$$

where Θ_τ is a threshold. This is especially needed on edges in the scene, where it might occur that a point far behind the edge in the new frame would transform onto the edge in the keyframe, e.g. due to rounding. We choose to not update the RGB data which might yield better coloring results but would also increase runtime. Furthermore, unlike depth where invalid measurements can occur, color information is available for every pixel and it is therefore sufficient to color the whole 3D model by just using the RGB image of the keyframe.

Since after every new frame the 3D world model is updated, we need to de-integrated the depth map with the reference keyframe first, then update it with the new depth map of the frame and finally re-integrate it. On the other side, if the current frame is a keyframe, we try to fuse the pointclouds into the new keyframes depth map. In this case every homogeneous 3D point $\tilde{\vec{X}}$ of the pointcloud is transformed into the current keyframe by using the transformation matrix T_{cn} (transforming a point of frame n into the current frame c):

$$\tilde{\vec{X}}' = T_{cn} \cdot \tilde{\vec{X}} \quad , \quad T_{cn} = T_{cw} \cdot T_{wn} \quad , \tag{11}$$

where T_{cw} is the transformation matrix from world coordinates into the current keyframe, and $T_{wn} = T_{nw}^{-1}$ the inverse of the transformation matrix from world coordinates into the keyframe n. We now apply the mapping $\pi(X')$ (2) to get the 2D image coordinates of the current keyframe the 3D point maps to. Finally we can again calculate the update step (9) if the mapped point lies within the image boundaries and satisfies (10). Every point we are able to map in this manner is removed from its pointcloud and if the number of points within a pointcloud falls below a certain threshold Θ_{pc} we delete the whole pointcloud. After this process, we use the depth map as a complemented and smoothed depth image (see Fig. 3), which we integrate into the InfiniTAM model.

F. Global Model Update

The ORB SLAM2 system continuously refines the estimated poses and whenever a new keyframe is selected, we verify the integrated poses from the model with the updated poses. If a significant change occurs, we update our 3D model in real-time. We achieve this model update by de-integrating the depth map with the old pose from the model and re-integrating it with the new pose. In cases, where ORB SLAM2 deletes a keyframe KF_{delete}, we search for the closest keyframe $KF_{closest}$ and de-integrate both from the 3D model. Then we fuse KF_{delete} into $KF_{closest}$ with (9) and re-integrate $KF_{closest}$ into the model.

G. Implementation

Since this process of constant de-integrating and re-integrating can be computationally intensive, we parallelized the update step via CUDA specific code. Note that in a few cases we run into the problem of collision (two or more points in the frame correspond to the same coordinates in the keyframe). In this case, only 1 point will be integrated and the other points are lost. However, this loss of information

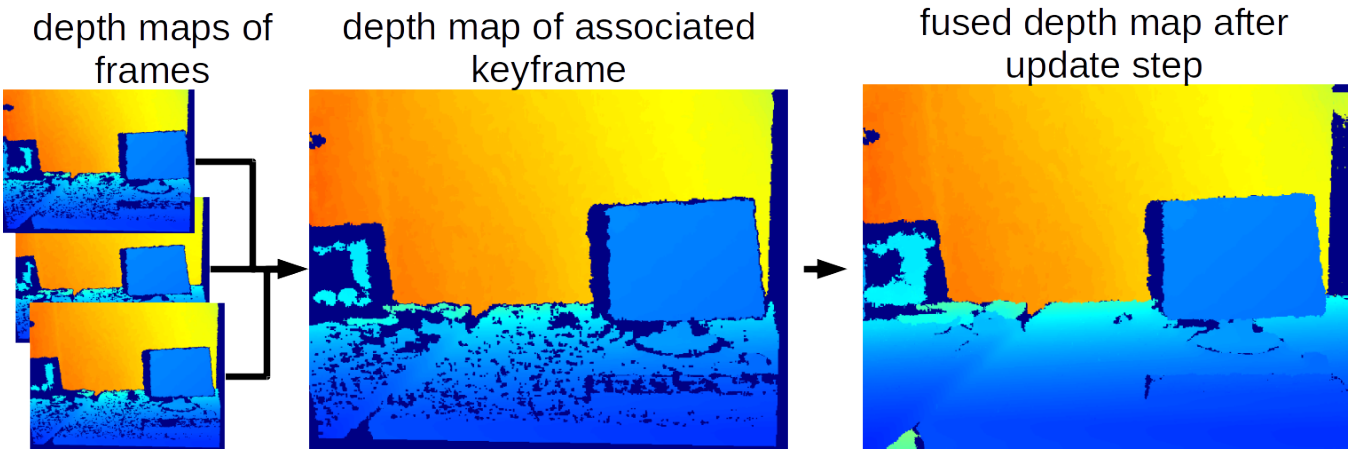


Fig. 3: Our depth map update complements and smooths the depth map of the keyframe.

can be tolerated for the sake of speed (and not needing any atomic operations). Furthermore, we introduce a "fast mode", where frames will only be integrated into the model if a new keyframe is processed, i.e. new frames will only update the depth map of their keyframe but are not directly integrated into the 3D model. A downside to this is that visual feedback provided to the viewer is not immediate but always one keyframe behind. In this manner our system is able to process an average of 15-20 frames per second, where the largest limiting factor is the tracking of ORB-SLAM2 running exclusively on the CPU.

IV. RESULTS AND DISCUSSION

To demonstrate our capabilities we test our system on several real-world image sequences from the TUM RGBD dataset [13] and on the synthetic ICL-NUIM dataset [6]. We evaluate standard InfiniTAM (ITM) [7], ICPCUDA [14], DVO SLAM [8], RGBD SLAM [4] and our method based on ORB SLAM 2 [9]. ICPCUDA is a very fast implementation of ICP with online available code [1]. We run all systems in their standard settings from using the code available online at maximum resolution of 640×480. For RGBD SLAM, we set the feature detector and descriptor type to ORB and extract a maximum of 600 keypoints per frame. In ORB SLAM2, we extract 1000 features per frame with a minimum of 7 per cell and 8 scale pyramid levels. Finally, we run DVO SLAM with its standard 3 scale pyramid levels. We test all the systems on an Intel Core 2 Quad CPU Q9550 desktop computer with 8GB RAM and an NVIDIA GeForce GTX 480. For all models we chose a voxel size of $2cm$ and a truncation band μ of $8cm$ and limited the depth measurements from $0.2m$ to $5.0m$. We empirically found the parameters $\Theta_\tau = 0.005$ and $\Theta_{pc} = 1000$. A high value choice of Θ_τ can result in depth inconsistencies at edges (as stated in III-E), while $\Theta_\tau = 0$ would reject any depth map update. The purpose of Θ_{pc} is to save memory by deleting the whole point cloud if it falls below this threshold. Therefore, a high threshold leads to a deletion of more pointcloud entries and consequently trades a loss of information for memory capacity.

A. Trajectory and Drift Estimation

TABLE I: ATE RMSE on the TUM RGB-D dataset and the synthetic ICL-NUIM dataset *[m]*

	ITM	ICP CUDA	DVO SLAM	RGBD SLAM	ORB SLAM2
fr1/desk	0.291	0.144	0.169	0.027	**0.022**
fr1/desk2	0.483	0.273	0.148	0.041	**0.023**
fr1/room	0.523	0.484	0.219	0.104	**0.069**
fr1/xyz	0.032	0.042	0.031	0.017	**0.010**
fr2/desk	0.114	1.575	0.125	0.092	**0.079**
fr2/xyz	0.042	0.223	0.021	0.016	**0.013**
fr3/office	1.258	1.161	0.120	0.034	**0.011**
fr3/nstn	1.979	1.666	0.039	0.051	**0.018**
lr/kt0	0.045	0.697	**0.006**	0.011	0.008
lr/kt1	0.009	0.045	**0.005**	0.013	0.162
of/kt0	0.054	0.205	**0.007**	0.029	0.027
of/kt1	0.025	0.275	**0.004**	0.724	0.051

We use the evaluation tools provided by [13] to calculate the absolute trajectory error (ATE) and the relative pose error (RPE). As suggested in [13], we compare the root mean squared error (RMSE) of the ATE and RPE. The ATE directly compares the absolute distances of the trajectory in the ground truth file and the output trajectory of the various systems. This is a good measurement for global consistency in SLAM systems. Let $P_{1:n}$ be the estimated trajectory and $Q_{1:n}$ the ground truth trajectory. Then we can find a least-squares solution for the rigid-body transformation S which maps $P_{1:n}$ onto $Q_{1:n}$ and compute the absolute trajectory error at time step i:

$$F_i := Q_i^{-1} S P_i \quad . \tag{12}$$

Table I shows the results for the ATE RMSE where ORB-SLAM2 outperforms all other systems on the TUM RGBD sequences. On the ICL-NUIM datasets, DVO-SLAM outshines ORB-SLAM2. This is due to the synthetic nature of the datasets, where perfect depth values allow a very accurate tracking for DVO-SLAM, whilst ORB-SLAM2 still needs to rely on the extracted ORB features. The high error value

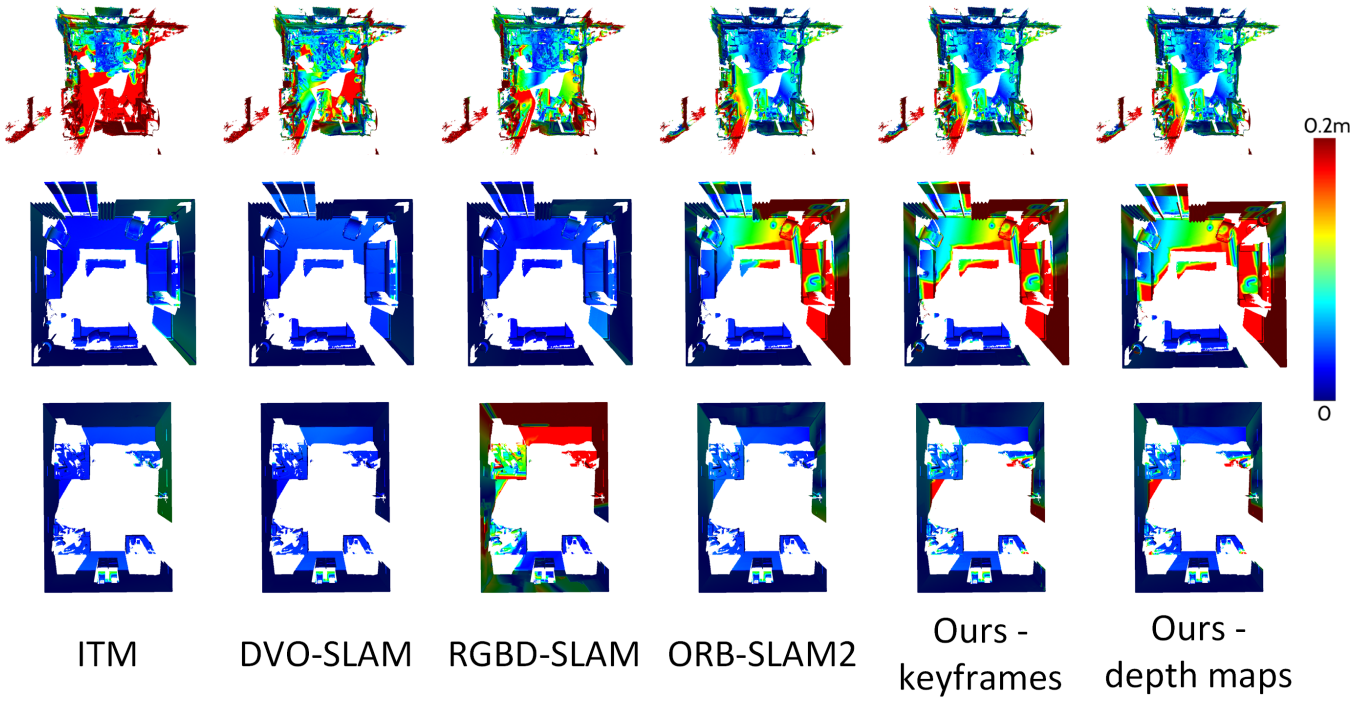


Fig. 4: Surface Reconstruction: Heat maps depicting the error from the ground truth model to the estimated model. Datasets from top to bottom: fr1/room, lr/kt1, of/kt1.

(a) lr/kt0 (b) fr1/xyz

Fig. 5: Sample reconstruction models of our approach from the TUM RGBD and ICL-NUIM datasets.

for the lr/kt1 sequence with ORB-SLAM2 is a result of not revisiting any structure and therefore being unable to perform a loop closure. The RPE computes the relative difference of the trajectory over a fixed time interval Δ. In visual odometry systems it evaluates the drift between frames and in SLAM systems it can measure the accuracy at loop closures. The RPE at time step i is defined as:

$$E_i := (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta}) \quad . \tag{13}$$

We choose to evaluate the RPE in Table II over the time interval of 1 second ($\Delta = 1s$). Here again, the synthetic ICL-NUIM datasets show slightly better results for the other systems compared to ORB-SLAM2. In order to be able to use the TUM tools, we converted all datasets into the TUM format, i.e. we changed the image and ground truth formats and added the associate files which can also be generated with the provided tools. In cases where the algorithm is non-deterministic, i.e. the result trajectories differ for every run, we execute the algorithm 10 times and take the mean value. Algorithms which belong to this category are ORB SLAM2 and RGBD SLAM.

B. *Surface Reconstruction Accuracy*

To measure the surface reconstruction accuracy, we calculate the one-sided Hausdorff distance from the groundtruth

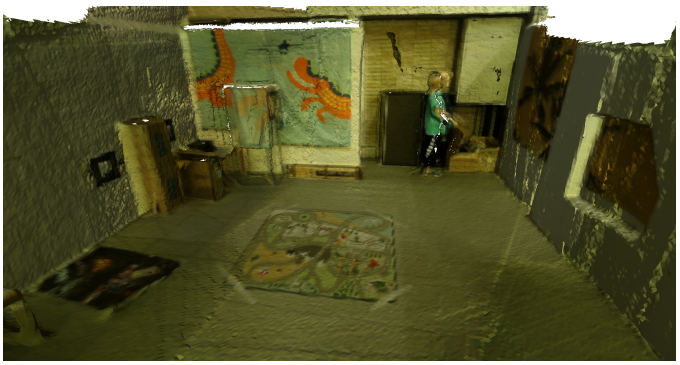
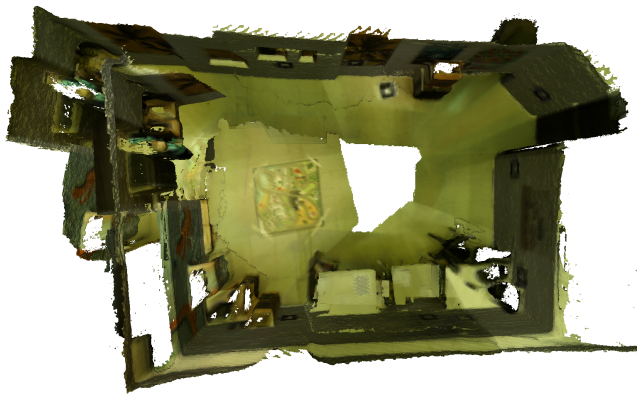


Fig. 6: Sample reconstruction models of our approach from our own Orbbec Astra Pro recordings.



(a) Original InfiniTAM

(b) Our approach

Fig. 7: Sample reconstruction of a room recorded and reconstructed in real-time with our Orbbec Astra Pro. (a) shows the original InfiniTAM reconstruction, which is unable to adapt the model to loop closure (see top left corner). (b) depicts our approach with a globally consistent model.

TABLE II: Translational RPE RMSE on the TUM RGB-D dataset and the synthetic ICL-NUIM dataset with $\Delta = 1s\ [\frac{m}{s}]$

	ITM	ICP CUDA	DVO SLAM	RGBD SLAM	ORB SLAM2
fr1/desk	0.207	0.100	0.052	0.036	**0.026**
fr1/desk2	0.327	0.164	0.061	0.045	**0.033**
fr1/room	0.259	0.129	0.056	0.053	**0.048**
fr1/xyz	0.047	0.031	0.024	0.027	**0.016**
fr2/desk	0.024	0.109	0.016	0.018	**0.012**
fr2/xyz	0.007	0.027	0.005	0.006	**0.004**
fr3/office	0.052	0.131	0.017	0.016	**0.009**
fr3/nstn	0.242	0.263	0.017	0.019	**0.015**
lr/kt0	0.005	0.140	**0.002**	0.003	0.008
lr/kt1	**0.001**	0.017	0.002	0.002	0.074
of/kt0	**0.003**	0.061	**0.003**	0.005	0.016
of/kt1	**0.002**	0.152	**0.002**	0.007	0.034

TABLE III: Evaluation of the surface reconstruction accuracy: Hausdorff distances from the ground truth surface to the reconstructed surfaces *(m)*.

	ITM	DVO SLAM	RGBD SLAM	Ours		
				all frames	keyframes	depth maps
fr1/desk	0.067	0.071	0.037	**0.033**	0.037	0.034
fr1/desk2	0.091	0.088	0.078	**0.043**	0.051	0.048
fr1/room	0.228	0.152	0.164	**0.084**	0.091	0.087
fr1/xyz	0.033	0.046	0.019	**0.012**	0.017	0.015
lr/kt0	**0.004**	0.005	0.006	0.008	0.016	0.016
lr/kt1	0.015	**0.007**	0.008	0.097	0.114	0.113
of/kt1	0.014	**0.006**	0.095	0.017	0.027	0.025

3D model to the reconstructed 3D model:

$$d_H(X,Y) = \sup_{x \in X} \inf_{y \in Y} \mathrm{d}(x,y) \quad , \tag{14}$$

where X is the set of groundtruth vertices, Y the set of the reconstructed vertices and $\mathrm{d}(x,y)$ is the Euclidian distance between the two vertices x and y. We sample each vertex in X, find the distance to the closest point in Y and take the average. Table III lists the result of this process for different

datasets and methods. ORB SLAM2 outperforms all other systems on the freiburg1 datasets when integrating the model frame by frame without using de-integration (all frames). However, note that we used already optimized trajectories for this test and thus no pose updates had to be incorporated. When we only integrate keyframes into the model, i.e. all non keyframes will not be processed by the system, the reconstruction error increases slightly. We counter this effect by using our fused depth maps. On the ICL-NUIM datasets, InfiniTAM and DVO SLAM outshine ORB SLAM2. This is due to the synthetic nature of the datasets, where perfect depth values allow a very accurate tracking for the former two, whilst ORB SLAM2 still needs to rely on the extracted ORB features. Furthermore, we can see in Fig. 4 that no loop closure could be performed in the lr/kt1 dataset (due to not revisiting any structure), which leads to a larger error. Note that in the of/kt1 dataset our method shows some areas with an increased error. The reason for this is that no keyframe was detected there and consequently no values exist.

For further qualitative evaluation we tested our system on several well known datasets (see Fig. 5) and also on datasets recorded with our own Orbbec Astra Pro (see Fig. 6). The whole extend of our method is illustrated in Figure 7: The original InfiniTAM is unable to adapt the model to global updates and therefore structures can appear at the wrong places, e.g. the reconstruction of 2 walls on the left and the tables at the bottom. With our approach we obtain a globally consistent model.

V. CONCLUSIONS

In this paper we presented a real-time capable method to combine the tracking accuracy of a state-of-the-art SLAM system [9] with the dense model generation of a volumetric fusion system [7]. We utilize the depth maps of all frames but fuse them into the depth map of their corresponding keyframes. The fused depth map is then integrated into the 3D model instead of every single frame, resulting in a speedup of about a factor of 10. Using fewer keyframes can increase the speedup even further, but will also impact the quality of the model, especially if translation and rotation between keyframes becomes very large. In this manner our system is able to adapt the model online, when updated poses are available, e.g. after loop closure or bundle adjustment. For real world data we have shown that our method yields excelling results, especially when compared to the original InfiniTAM ICP approach. Note that our system is not limited to ORB SLAM2, but can in theory work with any keyframe based tracking method. Therefore, it could enable means for a globally consistent dense real-time 3D reconstruction for many different SLAM and VO systems often lacking this feature.

REFERENCES

[1] "Icpcuda," https://github.com/mp3guy/ICPCUDA, 2018, [Accessed 20-March-2018].

[2] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1996, pp. 303–312.

[3] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 3, p. 24, 2017.

[4] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with an rgb-d camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[5] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[6] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014.

[7] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1241–1250, 2015.

[8] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 2100–2106.

[9] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[10] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2011, pp. 127–136.

[11] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, p. 169, 2013.

[12] F. Steinbrücker, J. Sturm, and D. Cremers, "Volumetric 3d mapping in real-time on a cpu," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 2021–2028.

[13] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 573–580.

[14] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research (IJRR)*, vol. 35, no. 14, pp. 1697–1716, 2016.

Contributed Session 6

Efficient 3D Pose Estimation and 3D Model Retrieval

Alexander Grabner[1], Peter M. Roth[1], and Vincent Lepetit[2,1]

I. Problem Statement and Motivation

Retrieving 3D models for objects in 2D images is an increasingly important problem, driven by the recent emergence of large databases of 3D models such as ShapeNet [1]. However, this task is challenging for two main reasons: (1) 2D images and 3D models have considerably different representations and characteristics, making it hard to compare them. (2) The appearance of objects can significantly vary with the pose, but it is in general unknown and, thus, multiple poses have to be considered, which is very inefficient. To overcome these problems, in [2] we proposed first to predict the object's pose and then to use the estimated pose as a prior to retrieve 3D models from a database. In the following, we give a short summary of the approach in Sec. II and a sketch of results in Sec. III. For more details, we refer to [2].

II. Overview of the Approach

Pose Estimation: To robustly compute the 3D pose of the objects of interest, similar to [3], we predict the 2D image locations of virtual control points using a CNN. In particular, we compute the 2D image locations of the projections of the object's eight 3D bounding box corners. The actual 3D pose is then estimated by solving a perspective-n-point (PnP) problem. As this requires the 3D coordinates of the virtual control points to be known, we predict the spatial dimensions of the object's 3D bounding box and use these to scale a unit cube, which approximates the ground truth 3D coordinates. For this purpose, we introduce a CNN architecture which jointly predicts the 2D image locations of the projections of the eight 3D bounding box corners (16 values) as well as the 3D bounding box dimensions (3 values).

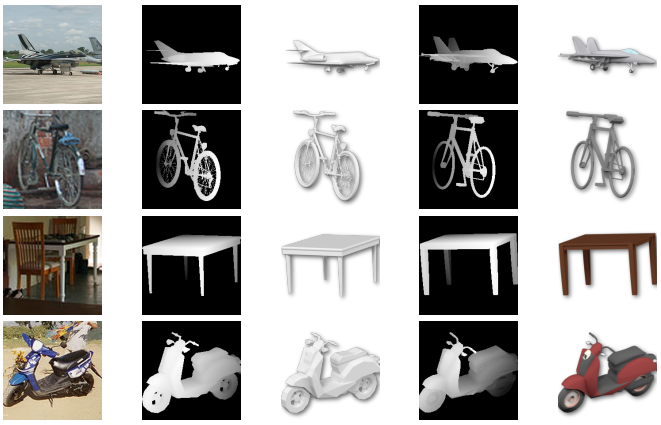
Model Retrieval: The actual 3D model retrieval is realized via descriptor matching between RGB images and depth images rendered under the estimated objects' pose. Using this prior significantly reduces the computational complexity compared to methods which need to process multiple renderings per 3D model. In addition, using depth instead of RGB images avoids problems with texture, different material properties, and illumination. However, RGB images and depth images have considerably different characteristics. Thus, we introduce a multi-view metric learning approach based on triplet loss optimization [4], which maps images from both domains to a common representation.

[1]Institute of Computer Graphics and Vision, Graz University of Technology, Austria {alexander.grabner}@icg.tugraz.at

[2]Laboratoire Bordelais de Recherche en Informatique, University of Bordeaux, France

III. Discussion and Illustrative Results

In this way, we are the first to report quantitative results for 3D model retrieval on Pascal3D+ [5] and show that our method, which was trained purely on Pascal3D+, retrieves rich and accurate 3D models from ShapeNet given RGB images of objects in the wild. In addition, we significantly outperform the state-of-the-art in 3D viewpoint estimation on Pascal3D+. A few illustrative 3D model retrieval results are sketched in Fig. 1. For more results, we refer to [2].



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

Fig. 1: 3D pose estimation and 3D model retrieval from ShapeNet given unseen images from Pascal3D+: (a) query RGB image, (b) depth image and (c) RGB rendering illustrating the ground truth pose and 3D model from Pascal3D+, (d) depth image and (e) RGB rendering illustrating our predicted pose and retrieved 3D model from ShapeNet.

Acknowledgment

This work was funded by the Christian Doppler Laboratory for Semantic 3D Computer Vision.

References

[1] A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep., 2015.

[2] A. Grabner, P. M. Roth, and V. Lepetit, "3D Pose Estimation and 3D Model Retrieval for Objects in the Wild," in *Proc. CVPR*, 2018.

[3] M. Rad and V. Lepetit, "BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth," in *Proc. ICCV*, 2017.

[4] K. Weinberger and L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[5] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond Pascal: A Benchmark for 3D Object Detection in the Wild," in *Proc. WACV*, 2014.

Being lazy at labelling for pose estimation

Georg Poier[1] David Schinagl[1] Horst Bischof[1]

Arguably, the best performing methods for real-time pose estimation heavily rely on models learned from data [1], [2], [3], [5]. Such data-driven models need to be trained on a large corpus of labeled data to deliver the expected results. Labeled data, however, is difficult to provide in the desired quantity, realism and accuracy. This is the case, in particular, for the task of 3D pose estimation of articulated objects. In this work we show that using a simple observation, which we exploit in a self-supervised training procedure, we are able to substantially reduce the amount of required labels. Using the proposed training procedure we are able to reach the same performance, even with one to two orders of magnitude less labeled samples.

The featured presentation covers the work published in [4]. Implementation source code and additional material can be found at `https://poier.github.io/PreView`.

I. Introduction to the task

In this work we focus on the exemplary task of 3D hand pose estimation from depth data. We want to learn a model, which – given a single depth image capturing a human hand – estimates the hand's pose. For this task the articulated structure and specific natural movements of the hand frequently cause strong self-occlusions. This not only makes the task more difficult, it also makes the currently necessary annotation procedure a huge effort for human annotators.

II. A simple observation

A largely unexplored direction to reduce the annotation effort is to exploit unlabeled data. This direction bears the advantage that unlabeled data for this task is easy to obtain in large quantities. Hence, we present a method that exploits unlabeled data by making use of a specific property of the pose estimation task. The method is based on the observation that pose parameters are predictive for the object appearance of a known object from any viewpoint. That is, given the pose parameters of a hand, the hand's appearance from any viewpoint can be estimated. This observation might not seem helpful upfront, since it assumes the pose – which we want to estimate – to be known. However, the observation becomes helpful if we capture the scene simultaneously from different viewpoints.

III. Exploitation of the observation

With a different camera view, we can guide the training of the pose estimation model. More specifically, by capturing another view, this additional view can be used as a target for training a model, which itself guides the training of the underlying pose representation. That is, by training a model which estimates a small number of latent parameters from the first camera view, and subsequently predicts a different view solely from these few parameters, these parameters become very predictive for the object pose. Setting the task up this way, a pose representation can be learned by simply capturing the hand simultaneously from different viewpoints and learning to predict one view given the other.

Using the low-dimensional pose representation learned from unlabeled data, a rather simple mapping to a specific target (*e.g.*, joint positions) can be learned from a much smaller number of training samples than required to learn the full mapping from input to target. Moreover, the model can easily be trained end-to-end – jointly with labeled and unlabeled data – in a semi-supervised fashion.

IV. Experimental results

Through an experimental evaluation we show that using the semi-supervised training procedure the proposed method consistently outperforms its fully supervised counterpart, as well as the state-of-the-art in hand pose estimation – even if all available samples are labeled.

In a more practical experiment we investigate the case where the number of unlabeled samples is larger than the number of labeled samples and find that the proposed method performs on par with the baseline, even with one order of magnitude less labeled samples. This indicates that the joint training regularizes the model to ensure that the learned pose representation can be mapped to the target pose space using the specified mapping.

In additional qualitative and quantitative experiments, we investigate the representations learned without any labeled data. In this way, we find that the proposed training procedure vastly improves the specificity of the learned representation and its predictiveness for the pose compared to related approaches towards learning without labels.

References

[1] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proc. CVPR*, 2017.

[2] H. Guo, G. Wang, X. Chen, and C. Zhang, "Towards good practices for deep 3d hand pose estimation," *ArXiv e-prints*, vol. abs/1707.07248, 2017.

[3] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. ICCV*, 2015.

[4] G. Poier, D. Schinagl, and H. Bischof, "Learning pose specific representations by predicting different views," in *Proc. CVPR*, 2018.

[5] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: Data, methods, and challenges," in *Proc. ICCV*, 2015.

[1]Institute for Computer Graphics and Vision, Graz University of Technology, Austria `find.us@the.web`