PROCEEDINGS OF THE AUSTRIAN ROBOTICS WORKSHOP 2022

Robotics for Assistance and in Healthcare

June 14-15, 2022
Congress Center Villach, Europaplatz 1, 9500 Villach

Lisa-Marie Faller, Justus Piater,
Gerald Steinbauer-Wagner, Mathias Brandstötter (Eds.)

ISBN 978-3-99076-109-0

June 2022, Villach

OVE Österreichischer Verband für Elektrotechnik
Bundesministerium Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie
KÄRNTEN University of Applied Sciences
FFG Forschung wirkt.
L3aD
ADMiRE
IEEE Austria Section
GMAR

Preface

The Austrian Robotics Workshop (ARW) emerged as a series of annual events by the Austrian robotics institutions, during which there is an exchange of thoughts and ideas. This workshop has been taking place since the beginning of the 21st century and is organized alternately by the individual robotics institutions in Austria, each time focusing on different topics.

This year's Austrian Robotics Workshop (ARW2022) is focusing on the topic of "robotics for assistance and in healthcare". While being open to many facets of robotics, the ARW 2022 is including, but not limited to topics such as mobile manipulation, multi robot coordination, artificial intelligence for sustainability. The contributions for the 2022 workshop covered a wide range of topics, ranging from industrial robots, service robotics to sensor-based applications. A student session was dedicated to ongoing or early work to encourage Master- and PhD-students to present and discuss their research topics. Students and young researchers from individual institutions and robotic companies presented their scientific work and thus entered into a lively exchange of ideas with the other researchers from other robotics institutions.

We would like to thank all authors, reviewers, presenters and speakers for their contributions to the workshop. Furthermore, we would like to thank Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation, and Technology, OVE - Österreichischer Verband für Elektrotechnik, GMAR – Gesellschaft für Mess-, Automatisierungs- und Robotertechnik, IEEE Robotics and Automation Society Austrian Chapter and Carinthia University of Applied Sciences – FH Kärnten for their support and contributions to the workshop.

Lisa-Marie Faller, Justus Piater, Gerald Steinbauer, Mathias Brandstötter

Villach, June 2022

Sponsors

FH Kärnten - gemeinnützige Gesellschaft mbH, A-9524 Villach, Europastraße 4 T: +43 (0)5 / 90 500 - 0 F: - 9910 www.fh-kaernten.at

FH Kärnten - gemeinnützige Gesellschaft mbH, A-9524 Villach, Europastraße 4 T: +43 (0)5 / 90 500 - 0 F: - 9910 www.fh-kaernten.at

Program Committee

- Mohamed Aburaia, University of Applied Sciences Technikum Wien
- Mathias Brandstötter, JOANNEUM RESEARCH Forschungsgesellschaft mbH
- Raimung Edlinger, FH Oberösterreich
- Lisa-Marie Faller, Carinthia University of Applied Sciences
- Daniela Krainer, Carinthia University of Applied Sciences
- Wilfried Kubinger, University of Applied Sciences Technikum Wien
- Andreas Müller, Johannes Kepler University Linz
- Kurt Niel, FH Oberösterreich
- Andrea Ojdanic, University of Applied Sciences Technikum Wien
- Justus Piater, University of Innsbruck
- Michael Rathmair, JOANNEUM RESEARCH Forschungsgesellschaft mbH
- Christian Schöffmann, Alpen-Adria-Universität Klagenfurt
- Lukas Silberbauer, taurob OG
- Gerald Steinbauer, Graz University of Technology
- Darijo Stojicic, ABB
- Markus Vincze, Vienna University of Technology
- Wolfgang Werth, Carinthia University of Applied Sciences
- Wilfried Wöber, University of Applied Sciences Technikum Wien
- Christian Wögerer, Profactor GmbH
- Hubert Zangl, Alpen-Adria-Universität Klagenfurt

Keynote Speakers

- Margit Gföhler, TU Wien
- Alexander Kollreider, Tyromotion GmbH
- Bernhard Rinner, Alpen-Adria-Universität Klagenfurt
- Ales Hribar, Kinestica d.o.o.

Awards

- Best Paper Award: **Offroad Terrain Classification for Mobile Robots,** *W. Walch, M. Eder, K. Mautner-Lassnig, G. Steinbauer-Wagner*
- Best Poster Award: **Trajectory Adaptation from Demonstrations with Constrained Optimization,** *H. Pérez-Villeda, M. Severiano and J. Piater*

Table of Contents

FH Kärnten - gemeinnützige Gesellschaft mbH, A-9524 Villach, Europastraße 4 T: +43 (0)5 / 90 500 - 0 F: - 9910 www.fh-kaernten.at

FH Kärnten - gemeinnützige Gesellschaft mbH, A-9524 Villach, Europastraße 4 T: +43 (0)5 / 90 500 - 0 F: - 9910 www.fh-kaernten.at

Offroad Terrain Classification for Mobile Robots*

Wendelin Walch[1], Matthias Eder[1], Konstantin Mautner-Lassnig[2] and Gerald Steinbauer-Wagner[1]

***Abstract*— In recent years, the capabilities of mobile robots have increased significantly, opening up new potential applications in a variety of settings. One area where mobile robots show great promise is in offroad terrain classification, where the ability to accurately identify and navigate different types of terrain is critical. In this paper we present a new pipeline for terrain classification in offroad environments. The main contributions of this pipeline are a combined offroad dataset utilizing publicly available datasets and minimizing manual labeling, a validated network architecture for optimized generalization to new environments, and a post-processing step to improve the reliability of the classification in the context of offroad navigation. The proposed approach was evaluated using publicly available data as well as newly collected data from offroad environments.**

I. INTRODUCTION

Terrain classification is a highly active research field in the computer vision and mobile robot domains. Providing semantic information about the environment around mobile robots from camera images opens new application areas. Performing tasks in offroad terrain is still a struggle for mobile robots, due to the lack of understanding of the environment around the system. The use of terrain classification aims to improve the semantic understanding of the robot's environment. While terrain classification in structured environments such as in automated driving is already performed with good results [20], transferring the research results to unstructured environments such as offroad areas is still a challenge [11]. Terrain classification overlaps strongly with the problem of semantic segmentation. Thus, many approaches use Convolutional Neural Networks (CNN) trained on annotated camera images [13], [28]. Such methods heavily depend on a large amount of training data.

Due to the huge diversity in unstructured environments, the camera images used for training need to cover a wide range of information about scenes and vegetation as well as seasons, light, and weather conditions. While the former is a challenge for the (automated) annotation of the images, the latter is a challenge for the acquisition of a large enough dataset. Another issue is, that CNNs trained for terrain classification typically work well only in the specific area in which the training data was recorded, reducing the applicability of the trained network to other scenarios.

*This work was supported by ARTI Robots and partially funded by the Austrian Research Promotion Agency (FFG) with the project Palona.

[1]Wendelin Walch, Matthias Eder and Gerald Steinbauer-Wagner are with the Institute of Software Technology, Graz University of Technology, Graz, Austria. {wendelin.walch, matthias.eder, steinbauer}@ist.tugraz.at

[2]Konstantin Mautner-Lassnig is with ARTI Robots, Gössendorf, Austria. k.ml@arti-robots.com

To overcome these challenges, we propose an adapted pipeline for general reliable offroad terrain classification using CNNs. First, to reduce the needed time for training data collection we reuse already publicly available offroad datasets. We combine and rearrange these datasets to a single one (CombSet), which represents a far richer dataset covering a wide range of environments under various conditions and providing a large amount of annotated images for training the terrain classification. Second, a broad evaluation of state-of-the-art CNNs for image segmentation was conducted to find a CNN architecture that is well-suited for a general offroad terrain segmentation. Based on the best performing network architecture a new terrain classification pipeline was developed that comprises a training step with optimized labels and a post-processing step based on confidence levels and spatial relations to improve the classification results to be useful for automated navigation. To evaluate the proposed pipeline in particular concerning generalization to new environments, new offroad data was collected in three different areas within Austria.

The remainder of the paper is organized as follows. In the next section we will discuss related research. In Section III we will present the proposed pipeline for terrain classification in more detail. In the following section the results of the experimental evaluation are presented. In Section V we draw some conclusions and discuss future work.

II. RELATED RESEARCH

In this section we discuss related work on various approaches for terrain classification, publicly available datasets in this context, and methods for image annotation.

A. Terrain classification

The authors of [17] implement a path prediction in structured environments using three different network structures to predict a driveable path out of a single image where tracks or roads are visible. The output of the trained CNN are segmented images containing the framings of the detected paths. As offroad environments often do not contain a path or track to follow, other approaches are required to find free space to navigate on. Therefore, the authors of [13] introduced a self-supervised ground segmentation network which uses stereo pair images, disparity, and intensity of the image to identify the ground. The authors of [4] focus on terrain traversability of slopes and obstacles using a stereo camera. Therefore, 3D points are sampled form the stereo camera into a 2,5D grid and fitted by B-Splines to estimate the slope. Based on this information, the traversability of the environment can be derived. The presented approach is computationally

expensive and thus relies on external infrastructure to achieve the ground segmentation in driveable and non-driveable terrain. Using full segmented images, the authors of [22] introduce importance-aware semantic segmentation (pixel-wise annotation) for urban environments. This is used to focus on possible risks for autonomous driving during the training of the network and to evaluate their importance. This way, it is possible to assign e.g. higher importance to pedestrians than to other static obstacles. In [16], transfer learning is performed to transfer scenes between on-road and offroad scenarios to construct annotations of the offroad scene. This works well for coarse annotation and shows that with a pretrained network, training of a new network is faster than starting from random initial values. This approach is also used in [28] where a pretrained network called DeconvNet is introduced. This network is trained with data from the Pascal Visual Object Classes (Pascal VOC) dataset [9], showing that it is also possible to improve performance with synthetic data in an offroad environment.

B. Datasets

There are multiple publicly available datasets for self-driving cars in urban environments (e.g. KITTI [1] and CityScapes [6]) which consist of different modalities to tackle different research topics. Most of the datasets provide annotated images using different classes for urban environments such as road, building, or traffic lights. For unstructured offroad environments fewer image datasets with semantic annotations are available. The Yamaha-CMU Offroad Dataset (YCOR) [23] is the only dataset containing images recorded in winter. However, the used vehicle is often seen on the bottom of the recorded images making it not applicable for the work on terrain classification. Other datasets (e.g. [7], [25]) are also not applicable for terrain classification due to the provided data types or missing annotations. Offroad datasets with the correct modality and a pixel-wise annotation of the image are Freiburg Forest [30], Robot Unstructured Ground Driving (RUGD) [33], and RELLIS-3D [18]. Each dataset has slight differences in camera setting, lighting, weather conditions, and annotation labels.

C. Annotation

The crucial part of datasets for terrain classification using a supervised learning process are the annotations or sometimes called labels. The annotations consist dense pixel-wise semantic classes. Often, the pixels hold a class identification number which refers to a specific semantic class such as vegetation or road. The standard approach to generate annotations is to manually annotate image by image (e.g. in [33]) which is time-consuming and error-prone. As assistance for the manual process, different tools are available providing features, such as edge detection or tracking the annotations over the following video frames [2], [27]. Post-processing techniques using a dense Conditional Random Field (CRF) can be applied to densify gained annotations retrieved by the annotation tools [23].

To motivate humans for manual annotation and to improve the annotation quality, gamification (interactive consensus agreement games) was used to generate image annotations which worked well for classification tasks [29]. The idea to use video games to generate annotations was also used in [26] which proved that the technique produces usable annotations and improves performance for real world applications. Using synthesized scenes to generate data is performed in different fields such as object detection [10], and indoor scene understanding [14], but not yet for offroad scenes.

III. Methodology

This section describes the proposed terrain classification pipeline for offroad environments in detail. First, the creation of a new combined dataset using already existing offroad datasets is presented. Second, it is described how existing CNN architectures which were previously already used for terrain classification were evaluated to find one that generalizes best. Third, a post-processing step based on confidence and spatial relations is presented which is used to optimize the segmentation result for navigation applications.

A. Combined Offroad Dataset (CombSet)

For the terrain classification we chose to combine multiple datasets containing RGB images, which provide dense information about the surrounding scene. The goal is to create an offroad terrain classification dataset that is large and diverse enough to train a neural network that generalizes well for different environments. The main goals of combining existing datasets are to avoid manual data collection and labeling as well as to represent different conditions and scenarios well. This way, datasets having different scenes, weather conditions, camera settings, or even seasons are combined into a comprehensive data collection that can be used to train a reliable image segmentation.

To consider as many situations as possible, several available datasets were combined into a new dataset called CombSet. For the CombSet dataset we combine three publicly available datasets: (1) Freiburg Forest [30], (2) RUGD [33], and (3) RELLIS-3D [18]. These datasets combined hold 14036 images with different ground-truth annotations. Each of them brings in its own features, helping to address the variety in offroad terrains. RUGD represents a unique view close to the ground which results in irregular boundaries and no noticeable structures like geometric shapes, vanishing points, or horizontal lines in the images. RUGD contributes a lot of different terrains as well as some semi-urban images from a village with man-made obstacles. Freiburg Forest contributes different forest-like locations recorded multiple times on different days. This leads to different light conditions in the same location and lens flares, which greatly impact the camera image. Rellis-3D provides a lot of open field scenes with man-made obstacles and different vegetation as well as bodies of water. In addition, in this dataset, trees are not represented in the majority of the image backgrounds in contrast to the other datasets. These three datasets together

represent a diverse dataset concerning used cameras, represented vegetation, light conditions, camera position/view, and observed structures. This will help to represent the diversity of the offroad scenes well for the semantic segmentation.

A crucial part of CombSet generation is the clever merging of the original semantic classes from the datasets into a combined annotation for CombSet. The challenge is to combine the semantic classes with the best match in terms of semantics from each dataset. When analyzing the defined semantic classes of the used datasets, different annotations for similar classes could be observed, making them incompatible with each other. To solve this issue, a closer evaluation of the semantic classes was performed to identify common class labels for CombSet.

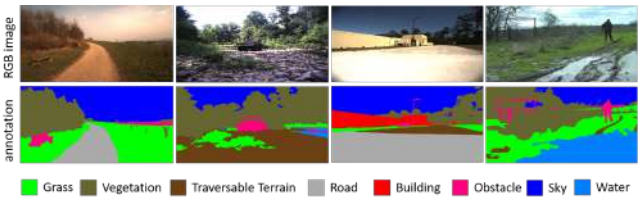


Fig. 1. CombSet sample images with the merged semantic classes.

For this, eight general semantic classes were identified: (1) road/path, (2) grass, (3) building, (4) water, (5) obstacle, (6) vegetation, (7) traversable terrain, and (8) sky. Road/Path represents all classes representing man-made tracks. The building class represents objects which do not change their position and mark reference points in the scene. The obstacle class combines all objects which can move such as vehicles and persons or objects which are too small for generating a reference point like a pole or sign. All pixels which do not fit into a defined class are represented in the CombSet as an obstacle due to the missing information and their potential risk in navigation. For the class vegetation, non-traversable natural objects such as trees, bushes, and other defined vegetation are combined. Traversable terrain combines all defined classes which are traversable but do not appear frequently enough to justify an own class. This class also represents different characteristics like a rock-bed which is traversable in contrast to a single rock. Although a puddle is a water body that is not very deep and is no lethal threat to the mobile robot, it is merged into the water class because of its similarity. The classes grass and sky are unchanged from the original datasets. No other classes were merged into these classes. This has an impact on the training of the neural network when considering the pixel-class distribution. Sample images with the new semantic classes can be seen in Figure 1.

B. CNN Comparison

To identify potential CNN architecture candidates for terrain classification, various network structures were trained and evaluated using CombSet. For deployment on a mobile robot, we assume limited computational resources. This is a crucial factor in choosing the architecture for the semantic segmentation used in offroad terrain classification. The evaluated CNN architectures in this work are: (1) Global Convolutional Network (GCN) [24], (2) AdapNet [32], (3) AdapNet++ [31], (4) RefineNet [21], (5) MobileUNet [19], and (6) DeepLabV3+ [5]. These architectures can be trained and run on single GPU units. Moreover, the selected architectures employ various measures to keep used resources low. DeepLabV3+, RefineNet, and GCN are using a Residual Network (ResNet) [15] framework as front end. This was pretrained with the ImageNet [8] dataset. AdapNet and AdapNet++ are initialized with the pretrained weights of ResNet. The number of layers and their sizes in ResNet influences the used resources as well. In the original implementation of most networks, the ResNet101 structure was used. In consideration of the resources later for the deployment on a mobile robot the ResNet50 was used which led to lower resource usage. All architectures were trained with the same parameter, optimization function, and training data (see Section IV-A for details).

C. Post-Processing

The output layer of the CNN provides an image containing for each pixel an array with the length of the semantic classes holding their probabilities. The class with the highest probability is usually seen as the predicted class (winner takes it all), even if the probabilities are distributed rather uniform.To increase the reliability of the segmentation result, the output is post-processed. For this, we propose the use of a Confidence Score (*CS*), an indicator for the reliability of the class prediction of each pixel in a given image. *CS* is a threshold where each prediction probability p lower than CS ($p < CS$) is considered a weak prediction and is set to be void which is treated as an obstacle. Pixels with a high probability ($p \geq CS$) will be assigned their corresponding classes. The resulting predictions can then be used for tasks such as autonomous robot navigation by further combining the results into a binary class (traversable, non-traversable). Due to the application of CS and blurred boundaries between larger structures such as streets and grass the resulting binary traversability image shows several tiny groups of pixels without classification which are with high probability part of its surrounding class. Thus, we added another post-processing step which conducts a morphological closing operation with a kernel of $9x9$, to smoothen the final traversability analysis. The derivation of an optimal CS threshold is discussed in the following section.

IV. EVALUATION & RESULTS

In this section we presents the detailed results on the evaluation of the performance and generalization of common CNN architectures for offroad terrain classification using CombSet. Moreover, we present results for the selection of the optimal Confidence Score. Finally, we show an evaluation of the best scoring network using newly recorded and segmented data, which is not part of CombSet.

A. Evaluation of CNN Architectures

The training of the different networks was conducted for 200 epochs on a Nvidia Quadro RTC 8000 GPU with

CUDA 11.3 and Tensorflow 2.6.0. RMSprop was used as the optimization function with an initial learning rate of $\alpha = 0,0001$ and decay of $\beta = 0,995$. As loss function, cross entropy loss with softmax [3] was used. The training occurs in batches of 10 images. After each epoch, a validation of the actual performance of the CNN is performed. As metrics, the mean accuracy of the prediction and the mean Intersect over Union (mIoU) [9] are used.

The results of the CNNs trained with CombSet can be seen in Table I. The RefineNet architecture performs best, slightly better than the GCN architecture. However, a direct comparison of the segmentation results shows more precise boundaries for the GCN architecture. As a good separation of semantic classes is an important part for the offroad terrain classification we choose the GCN architecture as the best performing architecture which will be further analyzed.

Network	mIoU	Accuracy
AdapNet	0,6411	0,8812
AdapNet++	0,6575	0,8655
DeepLabV3+	0,0235	0,0376
GCN	0,6839	0,9020
RefineNet	0,6985	0,9032
MobileUNet	0,6094	0,8384

TABLE I

TRAINING RESULTS OF DIFFERENT CNN ARCHITECTURES USING COMSET. GCN WITH LOWER PERFORMANCE BUT BETTER SEPARATION OF REGIONS WAS SELECTED AS BEST NETWORK CANDIDATE.

B. Network Generalization

To further evaluate the generalization performance of the GCN network architecture trained with ComSet, new data was recorded for validation. Therefore, three recording sessions were performed with two different mobile robots in three different locations in Austria: (1) the military training ground Seetaler Alpe, (2) the University Campus Inffeldgasse, and (3) the military training ground Allentsteig. As robot platform, Husky from Clearpath Robotics and Mercator [12], a universal offroad platform, were used.

Figure 2 shows an example image and the corresponding segmentation from unseen data recorded at the military training ground Seetaler Alpe. The segmentation shows various wrong classifications which renders the trained network not applicable for this new dataset. As a root cause for this issue, a different camera mounting was identified. While the camera in the validation set was tilted by 15 degrees, the camera setting in CombSet was always set to be horizontal.

To confirm this assumption, a new sensor setup was tested on the university campus Inffeldgasse using two identical cameras, mounted horizontally as well as tilted by 15 degrees. Figure 3 shows the segmentation results for both camera setups, with the horizontally aligned camera in the top row and the 15° tilted camera in the bottom row. A direct comparison of the two segmentation results confirms that the tilted camera leads to misclassifications by the trained GCN.

However, as these recordings do not represent a full offroad environment, further data at the military training

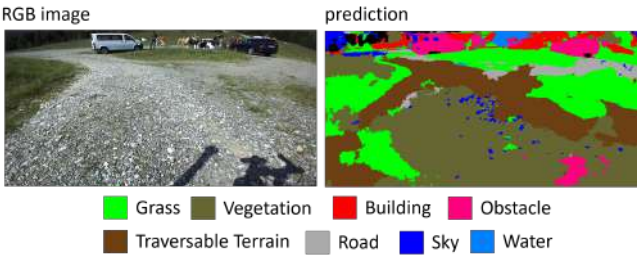


Fig. 2. Data recorded at the military training ground Seetaler Alpe. Input image to the GCN recorded with ZED2 camera (left). Output image of the GCN (right).

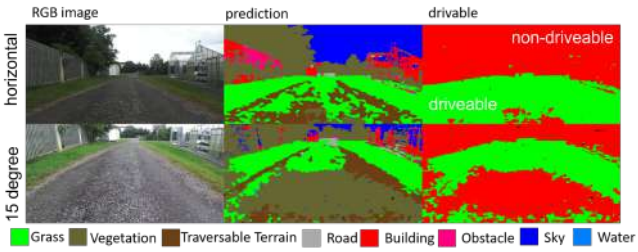


Fig. 3. Data recorded at the Campus Inffeldgasse at Graz UNiversity of Technology. Horizontally aligned camera (top) and 15° tilted camera (bottom). Input image (left), GCN segmentation (middle), and mapping of the segmentation to the category driveable and non-driveable (right).

ground Allentsteig was recorded. There, the recorded data was manually annotated to provide a quantitative evaluation as well. Figure 4 shows recordings from Allentsteig with the manually added annotations. Having a look at the segmentation on the right side one can see that it validates the consideration of the camera setup influences the segmentation.

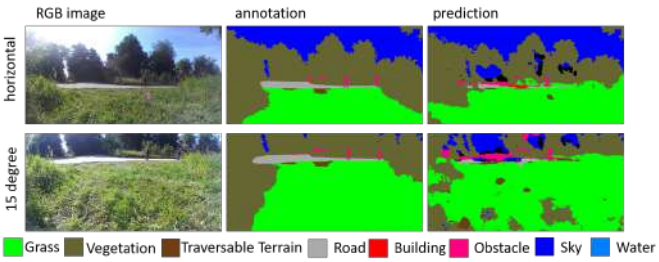


Fig. 4. Data recorded on the military training ground Allentsteig. Horizontally aligned camera (top) and 15° tilted camera (bottom). Input image (left), manually added annotation (middle), and GCN prediction (right).

A quantitative evaluation of the network results on the validation data can be seen in Table II. As baseline, 100 images from the CombSet testset were selected for a direct comparison using mIoU and accuracy as metrics. It clearly shows the influence of the proper camera setting.

C. CS Threshold

An investigation of different CS thresholds using the CombSet data showed the best trade-off between removed true negative and true positive pixels at 0.76. Using a

Dataset	accuracy	mIoU
CombSet Baseline	0,8735	0,6484
Allentsteig Horizontal Camera	0,7814	0,4234
Allentsteig 15 degree Camera	0,5544	0,2562

TABLE II

QUANTITATIVE EVALUATION OF THE INFLUENCE OF THE CAMERA POSE USING 100 IMAGES FROM THE TEST SET OF COMBSET DATASET AS BASELINE AND ALLENTSTEIG DATA WITH BOTH CAMERA POSES.

threshold of 0.99 would remove more than 19% of the image information where 13,24% are true positives. With a CS threshold of $CS = 0.76$ the numbers of removed wrongly and correctly segmented pixels are equal. The whole pipeline and the steps between can be seen in Figure 5, where an image from the CombSet test set is segmented and CS is applied as well as the post-processing. There we can see that the resulting post-processed image still contains some void pixels which refers to bigger areas without information and should be considered as obstacles. Anyhow, we achieved an image segmentation with dense information about the scene without much interference for the navigation.

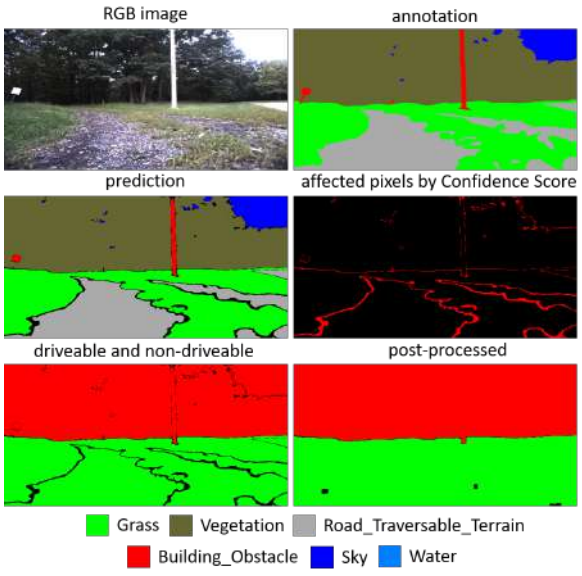


Fig. 5. Confidence Score approach using a test image from CombSet. Input image (top left), Annotation (top right), GCN segmentation (middle left), affected pixels in red by CS of 0.76 (middle right), segmentation mapped on driveable and non-driveable with the affected pixels in black (bottom left), and post-processed image (bottom right).

To validate the performance of GCN on unseen data, images from the Allensteig recording session were processed as well. Figure 6 shows a scene in the forest with challenging light conditions. Figure 7 shows an open scene with a car and the mobile robot oriented directly to the sun. The predictions show a decent performance for scene understanding, as well as a reliable segmentation that can be used for navigation.

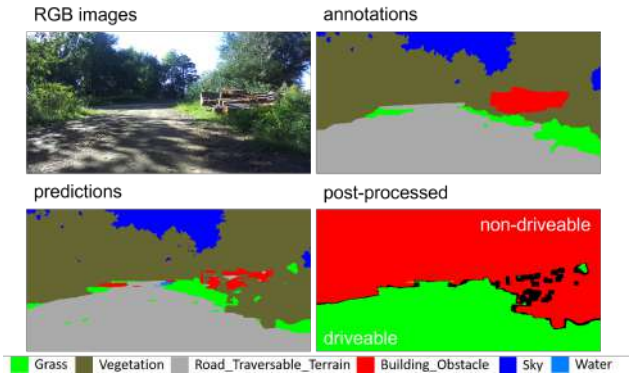


Fig. 6. Results of the GCN and CS for new unseen data from the Allentsteig recording session. Scene show challenging light conditions. Input image (top left), annotation (top right), GCN segmentation (bottom left), prediction with CS and post-processing step (bottom right).

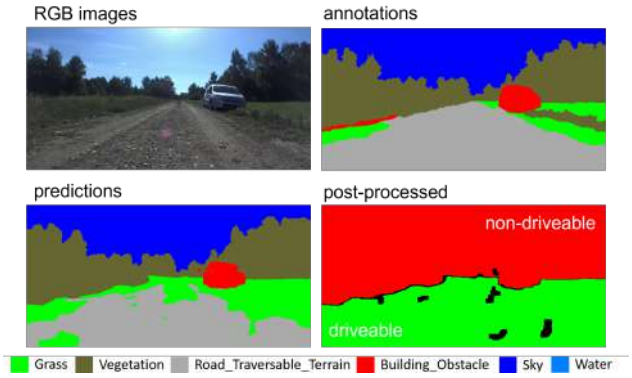


Fig. 7. Results of the GCN and CS of new unseen data from the Allentsteig recording session. The scene is facing direct into the sun and provides a car as obstacle. Input image (top left), annotation (top right), GCN prediction (bottom left), segmentation with CS and post-processing step (bottom right).

V. CONCLUSION

In this work, a pipeline for offroad terrain classification was presented. It can segment images from an unstructured offroad environment into eight semantic classes, relevant for navigation. As basis, three publicly available offroad datasets were combined into the dataset CombSet which can be used to train offroad terrain classification. CombSet is realized to provide a larger dataset than actually available datasets and represents a larger diversity in the images of offroad situations. It is larger than every other offroad dataset published to date with image pixel-wise annotation. A selection of state-of-the-art CNNs for semantic segmentation was trained and evaluated. The best performing network was GCN which showed the best performance on the test set. During the evaluation, influences of the camera setup (position and orientation) occurred and were solved by adjusting the camera setup.

The resulting segmentation is further processed to reach a more reliable prediction for applications such as navigation. The Confidence Score *CS* threshold is introduced which gives the possibility to remove pixel segmentation which have low confidence. The loss of information due to thresholding

using the CS is partly compensated by post-processing morphing operations. The resulting segmentation provides dense information of the covered scene concerning driveable and non-driveable segments. Overall, we showed that neural network models trained for semantic segmentation on offroad environments are generalizing well on new unseen data when following the proposed approach. For future work, the application of data augmentation will be investigated to improve the generalization. Moreover, the transfer of the segmentation to probabilistic gridmaps will be forced to improve the reliability of the proposed pipeline.

REFERENCES

[1] H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision (IJCV)*, 2018.

[2] Boris Sekachev, Nikita Manovich, Maxim Zhiltsov, Andrey Zhavoronkov, Dmitry Kalinin, Ben Hoff, TOsmanov, Dmitry Kruchinin, Artyom Zankevich, DmitriySidnev, Maksim Markelov, Johannes222, Mathis Chenuet, a andre, telenachos, Aleksandr Melnikov, Jijoong Kim, Liron Ilouz, Nikita Glazov, Priya, Rush Tehrani, Seungwon Jeong, Vladimir Skubriev, Sebastian Yonekura, vugia truong, zliang7, lizhming, and Tritin Truong, "opencv/cvat: v1.1.0," 2020.

[3] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Proceedings of the 2nd International Conference on Neural Information Processing Systems*, ser. NIPS'89. Cambridge, MA, USA: MIT Press, 1989, p. 211–217.

[4] A. Broggi, E. Cardarelli, S. Cattani, and M. Sabbatelli, "Terrain mapping for off-road autonomous ground vehicles using rational b-spline surfaces and stereo vision," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 648–653.

[5] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *CoRR*, vol. abs/1802.02611, 2018. [Online]. Available: http://arxiv.org/abs/1802.02611

[6] 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," *CoRR*, vol. abs/1604.01685, 2016. [Online]. Available: http://arxiv.org/abs/1604.01685

[7] L. Dabbiru, C. Goodin, N. Scherrer, and D. Carruth, "Lidar data segmentation in off-road environment using convolutional neural networks (cnn)," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 2, no. 6, pp. 3288–3292, 2020.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[9] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, 2014.

[10] S. R. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S. B. Kang, and T. Paek, "Learning to be a depth camera for close-range human capture and interaction," *ACM Trans. Graph.*, vol. 33, no. 4, jul 2014. [Online]. Available: https://doi.org/10.1145/2601097.2601223

[11] B. Gao, S. Hu, X. Zhao, and H. Zhao, "Fine-grained off-road semantic segmentation and mapping via contrastive learning," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5950–5957, 2021.

[12] R. Halatschek, K. Ramanna, W. Url, and G. Steinbauer-Wagner, "Universal offroad robot platform for disaster response," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 11 2020.

[13] M. Hamandi, D. Asmar, and E. Shammas, "Ground segmentation and free space estimation in off-road terrain," *Pattern Recognition Letters*, vol. 108, pp. 1–7, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865518300631

[14] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Scenenet: Understanding real world indoor scenes with synthetic data," *CoRR*, vol. abs/1511.07041, 2015. [Online]. Available: http://arxiv.org/abs/1511.07041

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[16] C. J. Holder, T. Breckon, and W. Xiong, "From on-road to off: Transfer learning within a deep convolutional neural network for segmentation and classification of off-road scenes," in *ECCV Workshops*, 2016.

[17] C. Holder and T. Breckon, "Learning to drive: End-to-end off-road path prediction," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 2, pp. 217–221, 2021. [Online]. Available: http://dro.dur.ac.uk/34081/

[18] P. Jiang, P. R. Osteen, M. Wigness, and S. Saripalli, "RELLIS-3D dataset: Data, benchmarks and analysis," *CoRR*, vol. abs/2011.12954, 2020. [Online]. Available: https://arxiv.org/abs/2011.12954

[19] J. Jing, Z. Wang, M. Rätsch, and H. Zhang, "Mobile-unet: An efficient convolutional neural network for fabric defect detection," *Textile Research Journal*, vol. 92, no. 1-2, pp. 30–42, 2022. [Online]. Available: https://doi.org/10.1177/0040517520928604

[20] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9396–9405, 2019.

[21] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," *CoRR*, vol. abs/1611.06612, 2016. [Online]. Available: http://arxiv.org/abs/1611.06612

[22] X. Liu, Y. Han, S. Bai, Y. Ge, T. Wang, X. Han, S. Li, J. You, and J. Lu, "Importance-aware semantic segmentation in self-driving with discrete wasserstein training," *CoRR*, vol. abs/2010.12440, 2020. [Online]. Available: https://arxiv.org/abs/2010.12440

[23] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*. Springer, 2018, pp. 335–350.

[24] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters - improve semantic segmentation by global convolutional network," *CoRR*, vol. abs/1703.02719, 2017. [Online]. Available: http://arxiv.org/abs/1703.02719

[25] Z. Pezzementi, T. Tabor, P. Hu, J. K. Chang, D. Ramanan, C. Wellington, B. P. W. Babu, and H. Herman, "Comparing apples and oranges: Off-road pedestrian detection on the NREC agricultural person-detection dataset," *CoRR*, vol. abs/1707.07169, 2017. [Online]. Available: http://arxiv.org/abs/1707.07169

[26] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision (ECCV)*, ser. LNCS, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906. Springer International Publishing, 2016, pp. 102–118.

[27] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, 2008. [Online]. Available: http://dx.doi.org/10.1007/s11263-007-0090-8

[28] S. Sharma, J. E. Ball, B. Tang, D. W. Carruth, M. Doude, and M. A. Islam, "Semantic segmentation with transfer learning for off-road autonomous driving," *Sensors (Basel, Switzerland)*, vol. 19, 2019.

[29] P. Upchurch, D. Sedra, A. Mullen, H. Hirsh, and K. Bala, "Interactive consensus agreement games for labeling images," in *HCOMP*, 2016.

[30] A. Valada, G. Oliveira, T.Brox, and W. Burgard, "Deep multispectral semantic scene understanding of forested environments using multimodal fusion," in *International Symposium on Experimental Robotics (ISER 2016)*, 2016. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2016/OB16c

[31] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *International Journal of Computer Vision*, vol. 128, pp. 1239–1285, 2019.

[32] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," 05 2017, pp. 4644–4651.

[33] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

An investigation of accuracy, repeatability and reliability of Spot® from Boston Dynamics

Robert Fina[1], Hubert Gattringer[1], and Andreas Müller[1]

Abstract— **The constantly increasing usage of quadruped robots in science as well as in industry shows that the last decade has experienced an immense leap in this technology. Several commercial products are already available and have been deployed in various applications. One of them is Spot® from Boston Dynamics, a ready-to-use, highly agile, robust mobile platform which is able to handle diverse situations. Despite these qualities, the robot comes with one drawback, the unknown system that is basically a black box. The goal of this work is to establish a starting point to gain insights about the accuracy, repeatability and reliability of the system. Accuracy and repeatability are investigated with the help of a highly accurate optical measuring system. The reliability is examined in various field tests. It is confirmed that Spot® convinces with high reliability and can undoubtedly be used in diverse applications where positioning is involved, such as inspections.**

I. INTRODUCTION

With the steadily increasing requirements and demands in the field of mobile robotics in general, specifically research regarding legged robots has increased considerable in recent years. Wheeled or tracked robots have significant limitations in comparison to legged or, in particular, quadruped robots. The latter two are able to move smoothly in different terrains, on different grounds, and are even able to work in complex or hazardous environments, meaning that obstacles such as fallen trees, stairs or else can be handled by these kinds of mobile robots. In order to enhance a robot's stability in difficult environments, advanced control methods, such as MPC (Model Predictive Control)[16] or Reinforcement Learning [12] can be applied. Despite legged robots' multiple advantages, only a limited number is currently commercially available.

One of the outstanding examples is the quadruped robot Spot®, the latest release from Boston Dynamics after BigDog and LS3. Spot® can be classified as an agile robot, being even able to counteract disturbances, such as being pushed, and maintain its balance[7], [8]. Thus, Spot® can be considered one of the most advanced and intelligent quadruped robots [3]. All joints are electrically actuated. Spot® is able to move or trod in all directions, is equipped with SLAM (simultaneous localization and mapping), providing the robot with all necessary information about its surroundings, and thus enabling it to perform in a diversity of complex terrains.

[1] Institute for Robotics, Faculty of Mechatronics, Johannes Kepler University, 4040 Linz, Austria

Besides the legged robots of the Boston Dynamic dynasty, also the quadruped robot ANYmal from ANYbotics [2], and Aliengo from Unitree [20] need to be mentioned, as they can be considered similar to some extent. ANYmal's legs provide outstanding mobility, making this robot a reliable instrument for difficult environments. The modular structure of the software, based on ROS, makes it extremely useful as a research platform. Aliengo is equipped with a depth-perception vision system, and can thus also be operated in adaptive complex terrains.

So far, quadruped robots are widely used in different fields, thus tackling a variety of research. In [21], a virtual reality interface is developed in order to enable an operator to control multiple robots both synchronously and asynchronously, making operation in dynamic construction environments possible. Also focusing on the dynamic construction environment, [1] enables construction monitoring through recording with a legged robot. Due to an increase in complexity when it comes to a simultaneous operation of multiple robots, an approach to design the operational governance of a heterogeneous multi-robot fleet is proposed in [14] and [19].

Using mobile robots as substitutes for humans for certain tasks can be an important aspect for increasing human safety. Quadruped robots can be used instead of humans in mines, for tasks such as installation, operation and maintenance of machines and other equipment [17], as well as for mapping [11]. They also found their way into the health industry, e.g. as a substitute for guide dogs [15]. Legged robots can also be of help when it comes to dealing with infectious diseases, as the ongoing Covid-19 pandemic clearly shows. Tasks, such as a contactless monitoring of vital signs [9], or even robots in 'caring roles' [18] are of significant importance. Nevertheless, specifically when it comes to human interaction, social acceptance may not be neglected [5]. Also, services and tasks related to a pandemic, such as contactless last-mile delivery, disinfection, patrolling and else can be carried out by mobile robots [10].

A. *Spot® Use Cases*

Boston Dynamics provides with Spot® a ready-to-use, balanced platform in terms of costs, benefits and features, with continuous development of the core software. However, using Spot® also has its disadvantages. It is a black box, and there is no access to the core software, wherefore its behaviour is mostly not predictable. Spot®'s basic features

and its Python API are a foundation which enables rapid implementation of use cases. The basic features include the access to the actual state of the robot, the sensor data, and the control of the robot. The standard software allows to command Spot® per waypoints or trajectories, and enables autonomous mission operations. It is also simple to add customised payload, which can communicate with the Spot® core software. For clarification, the following examples are discussed.

Zimmermann et al. [23] equipped Spot® with a light-weight robot arm, similar to a tutorial by Boston Dynamics[6]. The difference is that Zimmermann did not use a Spot® arm but one from Kinova Robotics. The movements of the attached arm are disturbances to the quadruped robot, and have to be considered. Zimmermann et al. show that the task of fetching an object can be successful with the following steps. First, an identification of model parameters is carried out. A simple model of the mobile manipulator with a position correction term, which compensates the influence of the arm to the body, is introduced. As a final step, a trajectory optimisation is performed to calculate the velocities for Spot and the arm separately. This work shows clearly that the black box behaviour can be compensated to solve the given task, without any learning techniques.

2020, a team using Spot® won the DARPA Subterranean Challenge, Urban Circuit. Bouman et al. [4] showed how Spot® can be used for large-scale and long-duration missions. The advantage over wheeled robots is that the agility of an quadruped robot enables it to cope with uneven surfaces, stairs, obstacles on the ground, etc. in urban environments. The developed system NeBula (Networked Belief-aware Perceptual Autonomy) combines a belief manager, perception, and planning to enable autonomy in extreme environments. NeBula builds a trust schematics of the environment, which helps Spot® to be able to move around safely. The reasons for that will be discussed in detail in Section III. Therefore, the combination of the two novel technologies is an outstanding example.

As it is shown, a quadruped robot can be used in a vast range of disciplines. Spot® from Boston Dynamics was already used in health care, construction, public service, and in many other challenging environments. An analysis of the black box Spot® is not done yet. Therefore, a closer look at the system, with the architecture described in Section II, is carried out in this work. The accuracy, and repeatability are analysed and the behaviour in different environments is observed. Results are discussed in Section III.

II. System Architecture

To observe the behavior under different circumstances and environments, only Spot® itself is necessary. For the analysis of the accuracy and the repeatability a high-resolution measurement system is required, as is an application to repeatable command the robot.

A. Hardware Setup

The motion capturing system from Qualisys (MOCAP) includes 10 cameras (6 MP, image resolution of 3072×1984, max. frame rate of 450 fps) working at a rate of 100 Hz. The accuracy of the system is sub-millimeter. The connection between the MOCAP and the control station of the robot is hard wired, whereas Spot® and the control station are connected wireless, as can be seen in Fig. 2. The measurements took place indoor in an industrial environment, see Fig. 1.

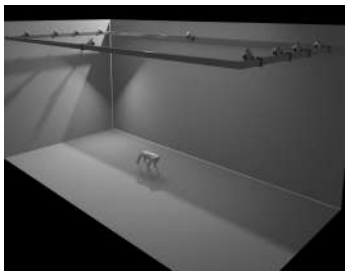


Fig. 1: The motion caption lab, where the measurements for the accuracy and repeatability took place.

B. Software Setup

The Python API provides the robot's operator with the possibility to command the robot through velocity, pose, and mission commands. In this work, the Python API is used to command Spot® to different positions to analyses the accuracy and repeatability. It is not used to investigate the behaviour of the robot in different situations. The position data is extracted from Spot® with 70 Hz with the Python API. The architecture is depicted in Fig. 2.

III. System Analysis and Results

The functionality of Spot® strongly depends on the perception of its environment. Therefore, the system was analysed regarding its behaviour under different conditions, which just comes done to reducing the ability to perceive the environment by covering cameras. Spot® is also tested in different environments, e.g. soft surfaces.

The accuracy and repeatability analysis is performed without any visual limitations.

A. Reliability

Spot® can be controlled by a remote controller or by using the Python API. An operator can use the remote controller to give direct commands to the robot, which can

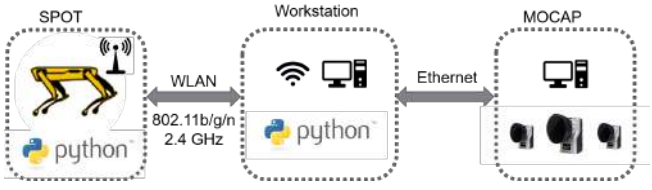


Fig. 2: Software and hardware architecture.

be interpreted as velocity control, or to arrange a mission, which Spot® autonomously executes. The remote controller is used to command the robot with various visual limitations in different environments to analyses its behaviour in this work. The test set includes outdoor - different grass heights, slopes, floor grid, etc. - and indoor tests - soft surfaces, such as a mattress of height 15 cm.

Five cameras are mounted on Spot®, one on each side, except at the front where there are two. Some tests were run where a varying number of cameras were occluded. It is observable that by reducing the robots visibility the gait's stability decreases. The lower limit that the robot can still handle well is three cameras. A working camera in walking direction is always recommended.

Tests on non-solid surfaces showed two effects. When the surface, which Spot® perceives, does not correspond to the ground, such as high grass, it causes problems. The stand ends in a different place than calculated which leads to strong instabilities and stress for the hardware. The system starts to jiggle when the robot is moved on a soft surface - a mattress of height 15cm - and its body height is changed to the maximal limit.

Surfaces like floor grids that cover ducts or similar are not a major obstacle for Spot®. Because of the camera angles, the floor grid is not perceived as uniform ground, but neither is it considered a gap, which has to be avoided. It has been observed, during the outdoor testing, that it is not a challenge to move the robot over terrain or in unstructured environments. A major advantage of quadruped robots is that uneven surfaces are not a concern.

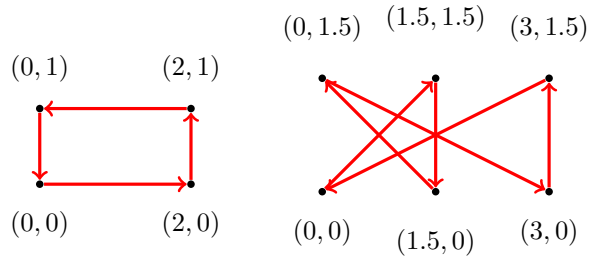
B. Accuracy and Repeatability

The system defines two inertial frames, the *vision frame* and the *odom frame*. The orientation and origin of both are set, when the robot boots up. The pose in the *odom frame* is evaluated over time through the kinematics of Spot®. The *vision frame* represents the pose over time by incorporating the *odom frame* information and visual information of the robot's environment. The *vision frame* and *odom frame* data are compared for the analysis of Spot®'s accuracy and repeatability, as a ground truth, data from a motion capturing system is shown.

A total of four different trajectories were driven 50 times each. The data present in this work is recorded with the default walking height of 61 cm. The trajectory set includes

- the *line walk*, where the goal is to walk two meter forward and backward,
- the *sideways walk*, where the goal is to walk two meter left or right,
- the *rectangle walk*, where the goal is to walk a rectangle-shaped trajectory with a side length of two meters, and
- the *points walk*, where the goal is to walk to different poses.

The *rectangle walk* and *points walk* trajectories are depicted in Figure 3.



(a) *rectangle walk* (b) *points walk*

Fig. 3: The trajectories *rectangle walk* and *points walk* used for the accuracy and repeatability measurements. (x, y) represent the positions in the *odom/vision frame*, which are commanded.

In Table I the results are listed and the end positions of the robot, for each test scenario, are depicted in a 2D map in Figures 5 to 8. The *odom frame* data is not included, due to the low accuracy and repeatability, as can be seen in Table I. The repeatability and accuracy is high when the *vision frame* is used as a reference, as is confirmed with the measurements by a motion capturing system. A comparison of both measurement methods - odometry and motion capturing - by a Bold-Altmann diagram shows that the odometry can be used as an accurate measurement alternative, as can be seen in Figure 4.

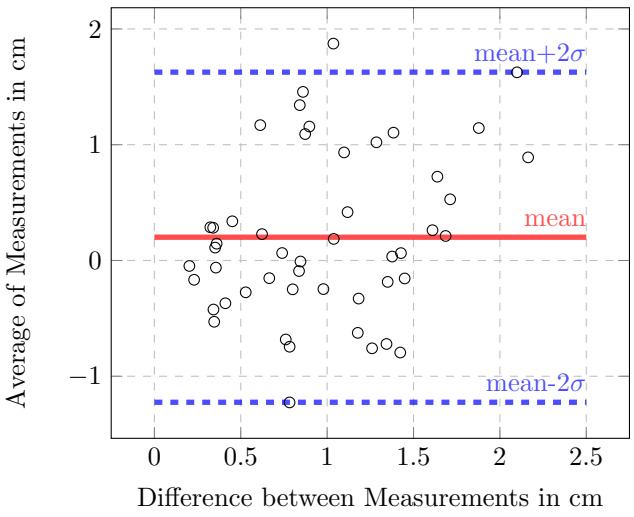


Fig. 4: A Bold-Altmann diagram comparing the motion capturing system and the Spot®'s vision odometry measurement from the *rectangle walk* scenario.

The start position of each trajectory differs from $(0, 0)$, because at the start the robot is sitting and has to stand up to

walk, which changes the body position. The results show that Spot® can be used in industrial and scientific matter without concerns about accuracy and repeatability.

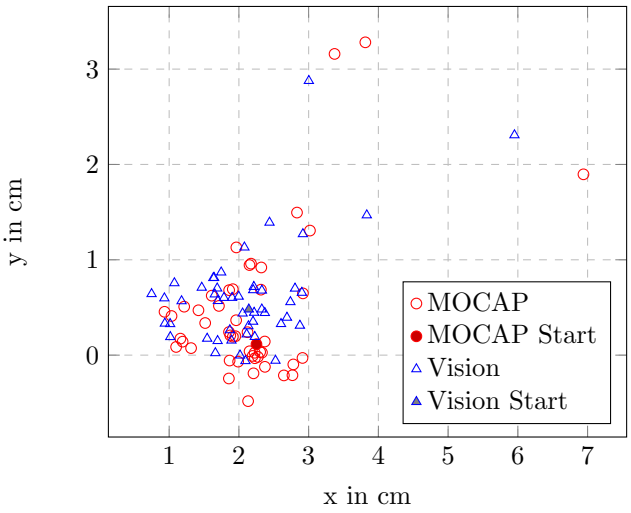


Fig. 5: Endpositions of Spot® in a 2D map of the *forward walk* scenarios.

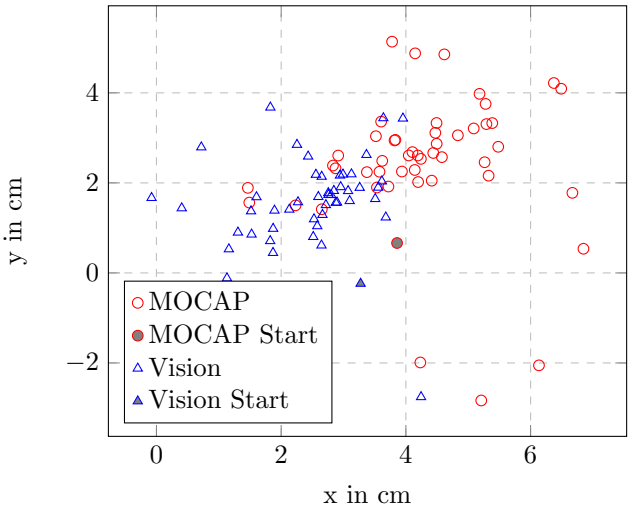


Fig. 6: Endpositions of Spot® in a 2D map of the *sideways walk* scenarios.

C. Synthesis

The analysis showed that Spot® struggles when the ground is incorrectly assessed. The surface, the robot observes, has to be the way it seems to the robot, meaning that if it looks solid it should be, otherwise Spot® has a hard time walking. The ground should not be too yielding in order to be able to stand and walk on it in a stable manner. Three cameras, at three different sides, should work correctly for a stable and reliable gait.

Some effects can be counteracted with different settings of Spot®. A ground friction coefficient can be set to cope

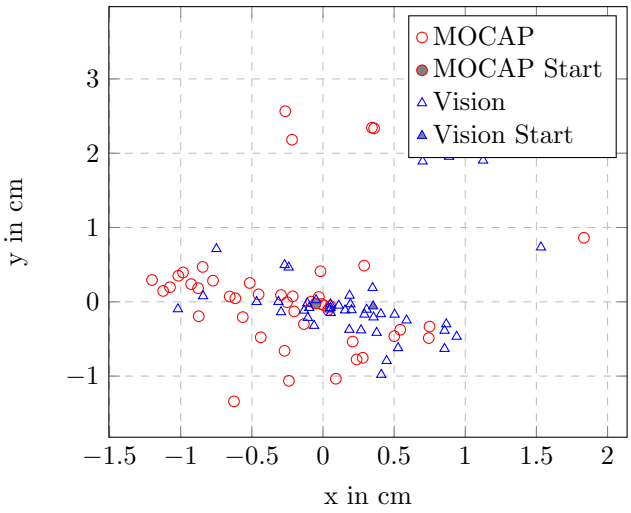


Fig. 7: Endpositions of Spot® in a 2D map of the *rectangle walk* scenarios.

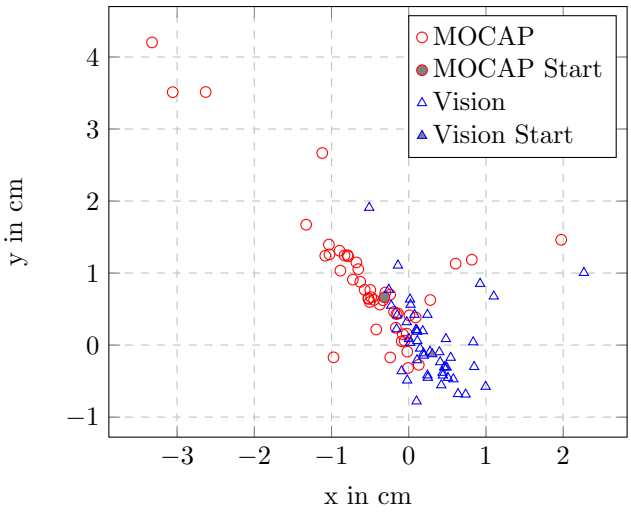


Fig. 8: Endpositions of Spot® in a 2D map of the *points walk* scenarios.

with different ground properties - the default value is 0.6. A smaller value decreases the step size and the step frequency to be able to walk on slippery surfaces. It is recommended to stay in the interval of $[0.4, 0.8]$[1] The step height can be set to three different levels, *Low, Medium and High*, where *High* is recommended to improve the ability to navigate[2]. The gait can be changed from *walk* to *crawl*, where three feet are always in contact with the ground which enhances the stability of the robot.

It is shown that Spot®'s accuracy and repeatability are high if the *vision frame* is used as the inertial frame. The *odom frame* should be avoided as a reference.

[1]https://dev.bostondynamics.com/protos/bosdyn/api/proto_reference.html #bosdyn.api.spot.TerrainParams

[2]https://support.bostondynamics.com/s/article/Tunable-platform-settings

TABLE I: List of standard deviation, variance and mean distance of the endpositions measured in different frames and scenarios: *forward walk* (f), *sideways walk* (s), *rectangle walk* (r), and *points walk* (p).

	Odom Frame				Vision Frame				MOCAP System			
	f	s	r	p	f	s	r	p	f	s	r	p
Standard Deviation in cm	8.56	28.41	29.31	81.48	1.03	1.23	0.53	0.56	1.15	1.96	0.70	1.78
Variance in cm^2	73.33	807.18	859.12	6638.86	1.07	1.52	0.28	0.32	1.32	3.86	0.49	3.16
Mean Value in cm	17.52	56.61	45.26	149.38	2.66	2.45	0.88	0.81	2.58	4.16	1.08	1.83

IV. OUTLOOK

The analysis in this work shows that Spot® is a reliable and accurate quadruped robot. Quadruped robots have advantages in unstructured environments making them well suited for operations in disaster areas. Further outdoor testing to verify the accuracy in terrain and unstructured environments using a Real Time Kinematic (RTK) as ground truth is an interesting next investigation.

The increasing usage rate of quadruped robots in the public and the increased market opportunities raise new questions in the field of human robot interaction (HRI) as well as ethical ones, like Moses et al. and Yunus et al. analysed in [22], [13]. HRI is mainly studied in industrial environment with mobile/static manipulators. Robots like Spot® open up new opportunities for HRI.

The cooperative handling of tasks by a mobile robot and a robot arm - meaning the mobile manipulator - still raises many questions. Due to the history of the institute of robotics with serial robot arms, a special light-weight serial robot arm is developed, which will be mounted on top of Spot® to investigate methods to handle the disturbance to the system and the impact on accuracy and repeatability.

ACKNOWLEDGMENT

This work has been supported by the "LCM - K2 Center for Symbiotic Mechatronics" within the framework of the Austrian COMET-K2 program.

REFERENCES

[1] K. Afsari, S. Halder, R. King, W. Thabet, J. Serdakowski, S. DeVito, M. Ensafi, and J. Lopez, "Identification of indicators for effectiveness evaluation of four-legged robots in automated construction progress monitoring," in *Construction Research Congress 2022*, 2022, pp. 610–620.

[2] ANYbotics. ANYmal - autonomous legged robot. [Online]. Available: https://www.anybotics.com/anymal-autonomous-legged-robot/

[3] P. Biswal and P. K. Mohanty, "Development of quadruped walking robots: A review," *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 2017–2031, 2021.

[4] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, *et al.*, "Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2518–2525.

[5] P. R. Chai, F. Z. Dadabhoy, H.-w. Huang, J. N. Chu, A. Feng, H. M. Le, J. Collins, M. da Silva, M. Raibert, C. Hur, *et al.*, "Mobile robotic systems in patient-facing functions: national acceptability survey, single site feasibility study and cost-effectiveness analysis," *medRxiv*, 2020.

[6] B. Dynamics, "Tutorial: Playing fetch with spot." [Online]. Available: https://dev.bostondynamics.com/docs/python/fetch_tutorial/fetch1

[7] E. Guizzo, "By leaps and bounds: An exclusive look at how boston dynamics is redefining robot agility," *IEEE Spectrum*, vol. 56, no. 12, pp. 34–39, 2019.

[8] E. Guizzo and E. Ackerman, "$74,500 will fetch you a spot: For the price of a luxury car, you can now buy a very smart, very capable, very yellow robot dog," *IEEE Spectrum*, vol. 57, no. 8, pp. 11–11, 2020.

[9] H. Huang, P. Chai, C. Ehmke, G. Merewether, F. Dadabhoy, A. Feng, C. Li, A. Thomas, M. da Silva, M. Raibert, *et al.*, "Agile mobile robotic platform for contactless vital signs monitoring," *TechRxiv, to be published*, vol. 10, 2020.

[10] A. Khamis, J. Meng, J. Wang, A. T. Azar, E. Prestes, Á. Takács, I. J. Rudas, and T. Haidegger, "Robotics and intelligent systems against a pandemic," *Acta Polytechnica Hungarica*, vol. 18, no. 5, pp. 13–35, 2021.

[11] A. Koval, S. Karlsson, and G. Nikolakopoulos, "Experimental evaluation of autonomous map-based spot navigation in confined environments," *Biomimetic Intelligence and Robotics*, p. 100035, 2022.

[12] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[13] J. Moses and G. Ford, "See spot save lives: fear, humanitarianism, and war in the development of robot quadrupeds," *Digital War*, vol. 2, no. 1, pp. 64–76, 2021.

[14] S. C. Nandakumar, S. Harper, D. Mitchell, J. Blanche, T. Lim, I. Yamamoto, and D. Flynn, "Bio-inspired multi-robot autonomy," *arXiv preprint arXiv:2203.07718*, 2022.

[15] V. Neubauer, "A prototype for robot assisted navigation for blindand visually impaired people," in *Proceedings of the 1st MCI Medical Technologies Master's Conference*, 2019.

[16] P. Saraf, A. Sarkar, and A. Javed, "Terrain adaptive gait transitioning for a quadruped robot using model predictive control," in *2021 26th International Conference on Automation and Computing (ICAC)*. IEEE, 2021, pp. 1–6.

[17] N. A. Sifferlinger, "Roboter im Bergbau–wo liegt der Bedarf?" *BHM Berg-und Hüttenmännische Monatshefte*, vol. 166, no. 2, pp. 53–58, 2021.

[18] S. Sumartojo and D. Lugli, "Lively robots: Robotic technologies in covid-19," *Social & Cultural Geography*, pp. 1–18, 2021.

[19] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth, *et al.*, "Cerberus: Autonomous legged and aerial robotic exploration in the tunnel and urban circuits of the darpa subterranean challenge," 2022.

[20] UnitreeRobotics. UnitreeRobotics Aliengo. [Online]. Available: https://www.unitree.com/products/aliengo/

[21] D. Wallace, Y. H. He, J. C. Vaz, L. Georgescu, and P. Y. Oh, "Multimodal teleoperation of heterogeneous robots within a construction environment," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2698–2705.

[22] A. Yunus and S. A. Doore, "Responsible use of agile robots in public spaces," in *2021 IEEE International Symposium on Ethics in Engineering, Science and Technology (ETHICS)*. IEEE, 2021, pp. 1–5.

[23] S. Zimmermann, R. Poranne, and S. Coros, "Go fetch!-dynamic grasps using boston dynamics spot with external robotic arm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4488–4494.

A low-drift LiDAR-based odometry for subterranean areas

Hamid Didari[1] , Eva Reitbauer[2] , Christoph Schmied[2] , Gerald Steinbauer-Wagner[1]

***Abstract*— This paper proposes a low-drift LiDAR-based odometry estimation for navigating robots in a subterranean environment. Due to the limitations of subterranean areas, sensors like Global Navigation Satellite System (GNSS) and cameras are limited in use. An alternative data source that can be used in subterranean areas is LiDAR. Due to lack of global information and measurement noise, pose estimation based on LiDAR is subject to drift. In this work, we introduce a solution to reduce LiDAR odometry drift. LiDAR odometry estimates the robot's pose by matching subsequent point clouds. Due to the lack of an absolute measurement system, the pose estimation has a cumulative error. A slight drift in the attitude estimation causes a massive error in the overall pose estimation. The proposed method tackles this problem with a filter for redundant information caused by flat ground and utilizes an Inertial Measurement Unit (IMU) to provide reliable information on the robot's attitude and fuses it with an Extended Kalman filter (EKF). The proposed method was tested on a dataset recorded in a highway tunnel and compared to state-of-the-art approaches. The final results outperform the state-of-the-art solutions in pose estimation in a subterranean environment.**

I. INTRODUCTION

Robotic systems are rapidly developing and employed in various fields to assist humans. In particular, robots are being utilized to work in complex environments for applications ranging from mining [13], search-and-rescue [8], industrial monitoring [3] to planetary exploration [6]. Pose estimation plays a critical role in making robots able to perform tasks and is one of the fundamental problems for autonomous robots. It provides the knowledge of robot's pose and instant feedback to the pose controllers. The pose of a robot can be estimated in various ways depending on what type of sensors are used. Each sensor type (e.g., LiDAR, radar, camera, GNSS) has advantages and disadvantages. For example, a global localization technique based on GNSS provides high accuracy on a highway, but it typically fails to work correctly inside a building.

One of the main criteria for selecting a sensor for estimating the pose are the conditions inside the working environment. This work focuses on subterranean areas, which typically have low light and bad signal reception, making both camera and GNSS less valuable options. LiDAR, which provides distance measurements from surrounding environments, may bypass the two mentioned issues and it has been widely used in robotic systems [16],[15]. Besides advantages compared to GNSS, LiDAR works in a local frame, and it suffers from drift over time. One way to reduce the drift of LiDAR odometry is loop closure. Loop closure is a fundamental part of many Simultaneous Localization and Mapping (SLAM) algorithms that helps correct the estimated pose of the robot by finding a loop in the map. One way of having a loop closure is to utilize a bag-of-words [20], which detects similarities between two scenes by feature clustering to construct a dictionary. However, a robot might never return to a previous location. Thus it is essential that LiDAR odometry is able to work well alone.

SLAM based on LiDARs has been the cornerstone of mobile robot mapping and navigation research. Leingartner provides a comparison between different sensors and mapping algorithms in a tunnel environment [11]. According to their results, Gmapping [5] is one of the best performing algorithms in a 2D setting, but it requires reliable odometry data. Moreover, the results confirm the environmental challenges for vision-based approaches due to the lack of light and features.

3D SLAM algorithms based on LiDARs typically utilize a scan registration approach, estimating the relative transformation between two scans. Iterative Closest Point (ICP) [1] is a well-known scan registration algorithm that iterates between finding corresponding points from two consequent scans, estimating the transformation between them, and minimizing a cost function. Other variations of ICP with different cost functions and correspondence methods have been developed [12],[19].

This paper introduces a LiDAR odometry system to estimate the pose of a robot in a subterranean area. With each new scan, the proposed system first uses filters to reduce the size of the point cloud and remove stationary points. Furthermore, it estimates the transformation matrix between the current and last point cloud based on Generalized Iterative Closest Point (GICP)[19] and it calculates the robot's pose. Furthermore, the cumulative errors are compensated by utilizing sensor fusion techniques using IMU data.

Due to the cumulative error in pose estimation, the robot's attitude drifts over time. Previous works utilize the Flat-Ground Assumption (FGA) and argue that the robot should not have a pitch/roll axis attitude, thus resetting the attitude frequently to compensate for the error.

The main difference between the proposed method and previous works is the use of an EKF with IMU data to correct the drifting errors on the attitude instead of using the FGA. While the IMU-based EKF produces similar results in most cases, it can outperform the alternatives in an uneven ground scenario. Furthermore, a Z-filter is introduced to

[1]Hamid Didari and Gerald Steinbauer-Wagner are with the Institute of Software Technology, Graz University of Technology, Graz, Austria. {hamid.didari, steinbauer}@ist.tugraz.at

[2]Eva Reitbauer and Christoph Schmied are with the Institute of Geodesy, Graz University of Technology, Graz, Austria. {eva.reitbauer, schmied}@tugraz.at

remove stationary/pseudo-stationary points in the output of the LiDAR. Moreover, we utilize the EKF, which is presented in [18] to improve the estimation of the pose.

The remainder of the paper is structured as follows. Section II covers related works. Section III explains the proposed method in detail. After that, Section IV describes the experiments and the analysis of results to show the effectiveness of the proposed method. Finally, Section V concludes the work.

II. RELATED WORKS

In this section, we review the related work on LiDAR-based odometry estimation. Cartographer [7] is a Graph-based SLAM, and it uses IMU and LiDAR data. Cartographer utilizes two steps. First, it estimates an initial transformation using IMU data. Then it uses LiDAR data to improve the estimation through scan matching between the current point cloud and a local map that is created with recent point clouds. Furthermore, it creates a global map from local maps and re-arranges local maps between each other so that they form a coherent global map.

HDL-Slam [10], like Cartographer, is a Graph-based SLAM approach. It uses GICP to match the two-point cloud. Since the odometry that comes from the matching point cloud drifts significantly on the z-axis, it assumes that the robot moves on flat ground. Whenever the LiDAR contains enough points on a flat area, it corrects the estimated attitude and position. This assumption works well in the area where the flat ground assumption is true, but it will not work on uneven ground.

In our previous work (Reitbauer et al., [18]), we developed an Extended Kalman Filter (EKF) to fuse information from LiDAR, wheel odometery, and IMU. We used GICP to estimate the pose from LiDAR and used both wheel odometry and the estimated pose from GICP as observations in the measurement update of the filter. The IMU was used as a reference navigation sensor to propagate the filter state in the time update. In this work, we enhace the pose estimation from LiDAR data by using keyframes and an attitude correction, and insert the enhanced pose into the EKF presented in [18].

III. METHOD

This section presents the details of the proposed method. A block diagram of the proposed method is shown in Figure 1. First, we explain how we estimate the robot's pose by utilizing the transformation matrix between two point clouds. Then we go into the details of recent algorithms for point cloud matching. After that, we explain the filters that are used on the point clouds. Furthermore, we present the EKF that is used for fusing the acceleration vector of IMU with LiDAR data to prevent the estimated attitude from drifting. Finally, we utilized the presented EKF in [18] to improve results and feed the scan matching algorithm with an initialized seed.

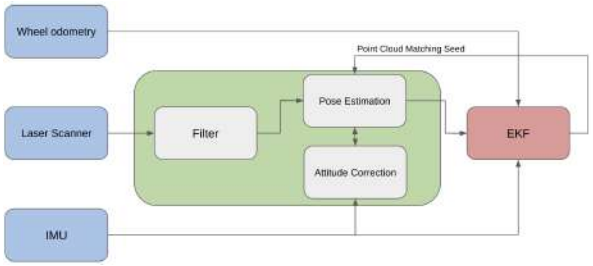


Fig. 1. The block diagram of the developed package for pose estimation. The blue boxes are sensors, the green box shows the parts developed in this work, and the red box is the EKF that is presented in [18] for sensor fusion.

A. Pose Estimation

One way to calculate the robot's pose X is to multiply the estimated transformation matrices between two point clouds to each other over time, as shown in Equation 1:

$$X = \prod_{i=2}^{t} T_{i,i-1} \tag{1}$$

where $T_{i,i-1}$ is the transformation matrix between the point cloud i and $i-1$ and t is the number of current point cloud.

Due to the limited accuracy of the LiDAR, there is inherent noise in the point clouds. Using the mentioned method will result in a cumulative error over time. Utilizing keyframes is an approach to reduce the cumulative error [9]. By selecting a point cloud as a keyframe and comparing all subsequent frames to it, the error remains constant for a short period. The keyframe is updated when the robot moves/rotates more than a certain threshold. The first point cloud is selected as the first keyframe in the proposed method. The robot's pose can be estimated by multiplying the transformation matrices between all keyframes, as shown in Equation 2:

$$X = \prod_{j=2}^{k} T_{j,j-1} T_{k,t} \tag{2}$$

where $T_{j,j-1}$ is the transformation matrix between the point cloud j and $j-1$, which are chosen as keyframes, $T_{k,t}$ is the transformation between the current point cloud and the last point cloud, which was chosen as a keyframe and k is the number of keyframes.

The registration of 3D point clouds is an ongoing research issue, and there are various different methods. One commonly used method is Iterative Closest Point (ICP). ICP is a matching method that works with a source point cloud (P) and a target point cloud (Q) [1]. The algorithm searches for corresponding points in these point clouds (p_i and q_i) and estimates the optimal transformation T_i by minimizing the Euclidean distance between corresponding points. The algorithm will iterate the process until a termination criterion is satisfied.

The state-of-the-art implementation of ICP utilizes SVD [14] to minimize the cost function given in Equation 3.

$$f = \sum_{i=1}^{N} ||p_i - T q_i||^2 \tag{3}$$

A significant drawback of the ICP algorithm is the inherent weakness against noisy data. Since the algorithm works directly on points from the point cloud and tries to minimize the distance between two points, the result is directly affected by sensor noise.

Normal Distribution Transformation (NDT) [2] employs a voxel-based association. By dividing the space into voxels and fitting a normal distribution to the points in each voxel, the problem of noisy data can be mitigated. The transformation matrix is estimated by maximizing the likelihood of two-point clouds under the distributions of voxels. NDT also requires less calculation due to voxelizing the space, So it is faster to compute.

Another way of handling the noise issue is to utilize the Generalized ICP (GICP) [19]. Instead of matching corresponding points, GICP utilizes the covariance matrices of neighborhood points to calculate the transformation matrix:

$$\begin{aligned} \tilde{p}_i &= \mathcal{N}(\hat{p}_i, C_i^p) \\ \tilde{q}_i &= \mathcal{N}(\hat{q}_i, C_i^q) \end{aligned} \tag{4}$$

where $\tilde{p}_i$ and $\tilde{q}_i$ are distributed according a normal distribution($\mathcal{N}$) with the mean values $\hat{q}$, $\hat{p}$ and for the neighborhood covariance matrix of points C_i^p and C_i^q. The transformation error can be defined as:

$$d_i = \tilde{p}_i - T\tilde{q}_i. \tag{5}$$

Moreover, the distribution of d_i can be calculated as:

$$\begin{aligned} d_i &\sim \mathcal{N}(\hat{q}_i - T\hat{p}_i,\, C_i^q + TC_i^pT^t) \\ &= \mathcal{N}(0,\, C_i^q + TC_i^pT^t). \end{aligned} \tag{6}$$

Finally, the cost function for GICP is given in Equation 7:

$$f_{\text{GICP}} = \sum_{1} d_i^T (C_i^q + TC_i^pT^T)^{-1} d_i. \tag{7}$$

B. Point Cloud Filtering

The large output size of the LiDAR point cloud combined with the presence of noise in the data leads to the inevitable use of filtering techniques [17]. The first essential filter is downsampling points. By reducing the amount of data, the cost/time of computation decreases while maintaining important information about the environment. Also, it helps to remove some outlier points. We utilize voxelizing as a downsampling filter. Moreover, we utilize an outlier rejection filter to remove any remaining outliers.

Due to the nature of the matching algorithms, stationary/pseudo-stationary points in the output of the LiDAR (e.g., the ground in front of the LiDAR always produces similar points in the point cloud, which looks stationary to the algorithm) will result in inaccurate pose estimation. Diaz et al developed a ground filtering algorithm for autonomous driving [4] and achieved promising results for detecting ground points. Since the main purpose of our study is localization, not navigation, we do not need to remove the ground points accurately. Ground points usually have the same height in the point cloud, and they can be removed by utilizing a filter on the z-axis (Z-filter). This filter has far less accuracy for detecting ground points than Diaz's work, but it is good enough for localization, and it has far less computational cost. This work assumes the points which have less height than 20 cm belongs to the ground.

C. Attitude Correction

As mentioned before, attitude drift is one of the main reasons for drifting LiDAR odometry over time. For instance, a 1-degree Drift over pitch angle leads to a 1.7 m error in 100 m. The attitude's error can come from drifting over time or an error in scan matching. HDL-SLAM algorithm uses FGA to tackle this problem, but the assumption does not hold for all cases. These errors can be compensated by fusing the gravity vector from IMU using EKF. The EKF state vector can be defined as:

$$x_k = \{\phi, \theta, \psi\}, \tag{8}$$

where ϕ, θ, ψ are the Euler angles and these angles measures by pose estimation. So, The transition matrix (F) matrix in EKF equations is equal to the identity matrix:

$$F = I. \tag{9}$$

The acceleration measurements from IMU are a combination of gravity and the dynamic acceleration of the robot:

$$a_{\text{IMU}} = a_g + a_d, \tag{10}$$

where a_{IMU} is the output acceleration vector, a_g is the gravity vector, and a_d is the dynamic acceleration vector. The dynamic acceleration is equal to zero when the robot does not move or has a constant speed and the IMU measures the gravity vector. This information can be used to compensate for the error in the angles. For detecting the impact of dynamic acceleration on the IMU measurement, g' is defined as:

$$g' = |1 - \frac{||a_{\text{IMU}}||}{||g||}|, \tag{11}$$

where g is the normalized gravity vector in the world frame. The value of g' is equal to zero if the IMU output is equal to the gravity vector. The dynamic acceleration would result in a bigger or smaller size of the IMU vector, which increases the value of g'. When g' is less than a threshold ($\varepsilon_{g'}$), we can assume that we don't have a dynamic acceleration and use IMU acceleration as a gravity vector. But if this is bigger than $\varepsilon_{g'}$, the IMU vector contains dynamic acceleration, and we couldn't use it for angles correction. In this work, we set $\varepsilon_{g'}$ to 0.05. Bigger value for $\varepsilon_{g'}$ causes errors in angles.

The expected measurement ($\hat{z}$) is the normalized gravity vector. It can be estimated by transforming the gravity vector from the world frame to the robot frame:

$$\hat{z} = h(x) = R(x) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \sin(\theta) \\ -\sin(\phi)\cos(\theta) \\ \cos(\phi)\cos(\theta) \end{pmatrix} \tag{12}$$

where $R(x)$ is the rotation matrix from the states. The measurement residual is computed as:

$$\tilde{y} = a_{IMU} - \hat{z} \tag{13}$$

The linearised observation functions can be written as:

$$H_k = \begin{pmatrix} 0 & \cos(\theta) & 0 \\ -\cos(\phi)\cos(\theta) & \sin(\phi)\sin(\theta) & 0 \\ -\sin(\phi)\cos(\theta) & -\cos(\phi)\sin(\theta) & 0 \end{pmatrix} \tag{14}$$

The EKF procedure can be summarized as :

- Prediction:
  - The states are updated by pose estimation
  - Predicted covariance estimate $\hat{P}_k = P_{k-1} + Q$
- Correction:
  - Measurement residual $\tilde{y}$ = a_{IMU} - $\hat{z}$
  - Innovation covariance $S_k = H_k \hat{P}_k H_k^T + R$
  - Kalman gain $K_k = \hat{P}_k H_k^T S_k^{-1}$
  - Updated state estimate $x_k = \hat{x}_k + K_k \tilde{y}$
  - Updated covariance estimate $P_k = (I - K_k H_k)\hat{P}_k$

where R and Q are noise covariance respectively for measurement and prediction.

D. Sensor Fusion

We utilize the IMU acceleration vector for correcting the attitude of LiDAR pose estimation. Besides that, IMU can be used to improve the accuracy of pose. Another sensor that can also be used in subterranean areas is wheel odometry. In our previous work [18], we developed an EKF which uses a pose from LiDAR as well as wheel odometry data in the measurement update and an IMU with a strapdown algorithm in the time update. The proposed method of this paper takes the enhanced pose estimation with keyframes and the EKF using an acceleration vector and inserts the estimated pose into the EKF presented in [18].

IV. Results

This section presents the experimental results of the proposed method and compares it to state-of-the-art LiDAR-based odometry solutions. We first introduce the dataset that is used for evaluation. Then, since the point cloud matching algorithms play a vital role in pose estimation, we compare the mentioned algorithms for point cloud matching in section III to each other. Furthermore, we present the impact of each part of the proposed method on pose estimation. Finally, we compare our method to other approaches. We used the Euclidean distance between the estimated pose and ground truth in the XY (2D) and the XYZ axis (3D).

A. Tunnel dataset

The tunnel dataset was recorded at Zentrum am Berg in Eisenerz, Austria. The robot was equipped with a Velodyne LiDAR (model: VLP-16) and an IMU (model: XSens MTi-G-710). The robot was navigated in the tunnel manually, and the ground truth was obtained by tracking the robot with two Leica total stations. The total travel distance for the robot is 135 meters. Figure 2 shows the robot and tunnel environment.

Fig. 2. Picture of the robot and tunnel environment.

B. Scan Matching

Scan matching is the central part of our pose estimation, and its performance directly affects the quality of the pose estimation. Thus, we compared the mentioned algorithms for point cloud matching in section III to choose the best scan matching method. For comparison, the transformation matrix between two point clouds is estimated by them and then by utilizing Equation 2, the pose is estimated. In Table II, the Euclidean distance between the final estimated position and ground truth in 2D and 3D for different scan matching methods is presented. The error of the vanilla ICP is 127.94 m which is expected due to the inherent weakness of its cost function against noisy data. Although NDT has more robust against noise, it did not show good performance because of the high similarity in the tunnel environment.

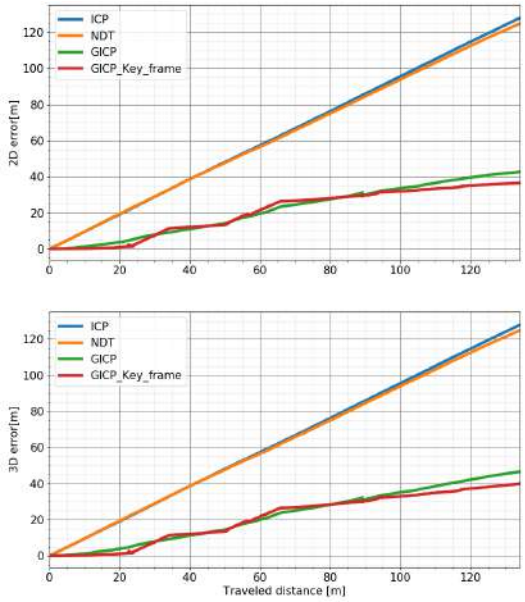


Fig. 3. Error of estimated pose vs. traveled distance in 2D and 3D for different point cloud matching algorithms for the tunnel dataset.

GICP does not have the inherent weakness of ICP. Thus, it had the best performance, which is 42.5 m in 2D and 46 m in 3D. Moreover, this performance can be improved by utilizing keyframes. Figure 3 shows the errors of scan matching algorithms on traveled distance. ICP and NDT couldn't estimate the transformation matrix well, resulting in a significant cumulative error. GICP also suffers cumulative error but far less than ICP and NDT. Since the GICP outperforms the alternatives, it is chosen as the scan matching method.

TABLE I

ERROR OF ICP, NDT,GICP, AND GICP WITH KEY FRAME.

Method	2D-error (m)	3D-error (m)
ICP	127.94	128
NDT	124.8	125.2
GICP	42.5	46
GICP-Key-frame	39.89	40.45

C. LiDAR-BASED POSE ESTIMATION

In order to evaluate different variants of the proposed pose estimation pipeline, we determined the error between the estimated pose and ground truth in the X, Y, and Z axis. The result can be seen in Figure 4. Z-filter indicates the error of using Z-filter together with GICP. Despite improvement it still shows a significant error. As well the Z-filter-key-frame illustrates the effect of utilizing keyframes. There is a significant improvement in 2D, but it still shows a big error in 3D. To reduce the error in 3D, an IMU is employed. The Z-filter-key-frame-IMU shows the effect of using IMU together with other parts. As shown in the figure, the error in the axis was reduced significantly. The pose estimation is further enhanced by utilizing the work of [18] for sensor fusion.

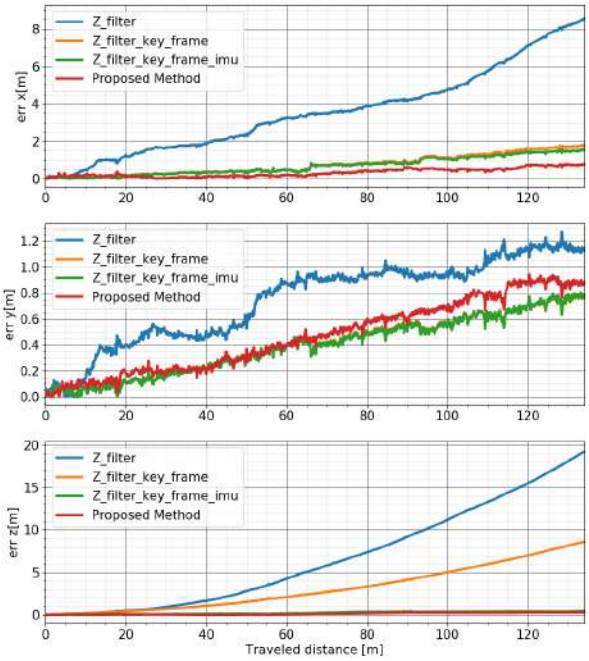


Fig. 4. Comparison between the effect of different variants of the proposed pose estimation pipeline.

D. Evaluation Against the State-of-the-Art

For comparison with other works, HDL-SLAM [10] and our previous work [18] was chosen. HDL-SLAM is a well-known LiDAR-based SLAM, and it uses GICP for point cloud matching. Thus it is a good candidate to compare the result of this work to it. Finally, we compare our results to work presented by our previous work because we employed the same sensor fusion setup. Figure 5 shows the performance of HDL-SLAM, our previous work and the proposed method. HDL-SLAM suffers from cumulative error more than the proposed method and our previous work. The proposed method performed better than our previous work In 2D and 3D.

TABLE II

SUMMARY OF ACCURACY RESULTS ON TUNNEL DATASET

Method	2D-error(m)	3D-error(m)
HDL-SLAM	41.25	42.74
HDL-SLAM-FGA	39.34	39.65
GICP-Z-filter	8.28	20.72
GICP-Z-filter-key-frame	1.95	8.72
GICP-Z-filter-key-frame-IMU	1.73	1.77
Reitbauer et al. [18]	1.50	2.40
Proposed-method	**1.17**	**1.20**

Table II shows the final error (Euclidean distance) in 2D and 3D for the different methods. HDL-slam tries to optimize the trajectory over time and due to stationary points in the point cloud, it failed to improve the estimation. By using

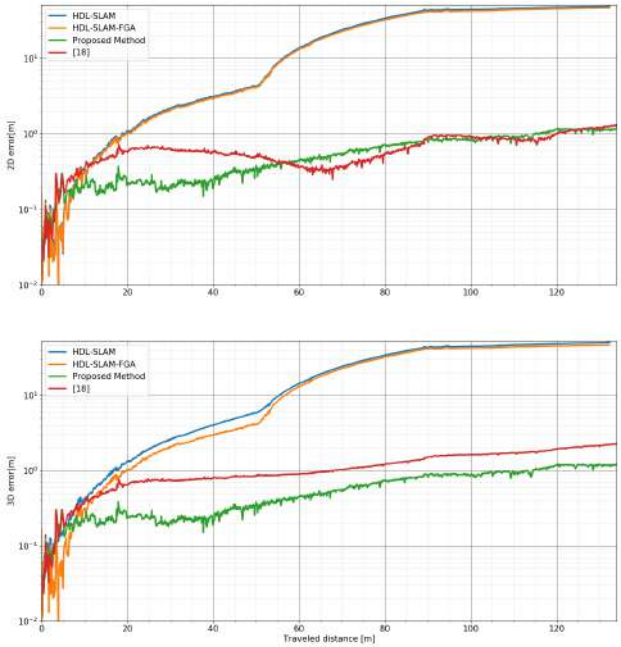


Fig. 5. Comparison between proposed method against the State-of-the-Art.

our approach and removing stationary points with a Z-filter, the final error drops to 8.29 m in 2D and 20.72 m in 3D. The drift over the z-axis is mainly caused by drifting in roll and pitch angle. This drift can be reduced by utilizing IMU. Furthermore, using the keyframes, the final error can be reduced to 1.95 m in 2D and 8.72 m in 3D. It can be seen in the table by using IMU, the error in the 3D is reduced to 1.77 m which is less than for the approach from [18], however, the 2D error is 23 cm larger than in [18]. Finally, by employing EKF that is presented [18] to improve the estimation, we reach 1.17 m and 1.20 m error in 2D and 3D which is less than 0.9 percent of the traveled distance.

V. CONCLUSIONS

This paper proposes a LiDAR-based odometry system to estimate a robot's pose in a subterranean area. This study shows that by removing stationary/pseudo-stationary points in point clouds, the accuracy of GICP can be improved. Furthermore, by using IMU data for attitude correction, pose estimation accuracy can be improved. The effectiveness of the proposed method is shown with an empirical experiment using a real-world dataset acquired in a tunnel setting. The proposed method reaches 1.17 m and 1.20 m error in 2D and 3D which is less than 0.9 percent of traveled distance, outperforming the state-of-the-art solutions in this experiment. Moreover, the proposed method benefits usability in an uneven ground scenario. In the future, a better filter for removing stationary/pseudo-stationary points will be designed.

VI. ACKNOWLEDGMENT

This research was funded by the Austrian Federal Ministry for Agriculture, Regions and Tourism via the Österreichische Forschungsförderungsgesellschaft (Austrian Research Promotion Agency) (879693).

REFERENCES

[1] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[2] P. Biber and W. Straßer, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2743 – 2748 vol.3, 11 2003.

[3] R. Bogue, "Robots for monitoring the environment," *Industrial Robot: An International Journal*, vol. 38, pp. 560–566, 10 2011.

[4] N. Diaz, O. Gallo, J. Caceres, and H. Porras, "Real-time ground filtering algorithm of cloud points acquired using Terrestrial Laser Scanner (TLS)," *International Journal of Applied Earth Observation and Geoinformation*, vol. 105, p. 102629, 2021.

[5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[6] J. Haruyama, T. Morota, S. Kobayashi, S. Sawai, P. G. Lucey, M. Shirao, and M. N. Nishino, "Lunar holes and lava tubes as resources for lunar science and exploration," in *Moon: Prospective Energy and Material Resources*, V. Badescu, Ed. Springer Berlin Heidelberg, 2012, pp. 139–163.

[7] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.

[8] J. Jennings, G. Whelan, and W. Evans, "Cooperative search and rescue with a team of mobile robots," in *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97*, 1997, pp. 193–200.

[9] G. Klein and D. Murray, in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, title=Parallel Tracking and Mapping for Small AR Workspaces*, 2007, pp. 225–234.

[10] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, 02 2019.

[11] M. Leingartner, J. Maurer, G. Steinbauer, and A. Ferrein, "Evaluation of sensors and mapping approaches for disasters in tunnels," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2013, pp. 1–7.

[12] K.-L. Low, "Linear least-squares optimization for point-to-plane ICP surface registration," *Chapel Hill, University of North Carolina*, vol. 4, no. 10, pp. 1–3, 2004.

[13] R. Lösch, S. Grehl, M. Donner, C. Buhl, and B. Jung, "Design of an Autonomous Robot for Mapping, Navigation, and Manipulation in Underground Mines," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1407–1412.

[14] S. Marden and J. Guivant, "Improving the Performance of ICP for Real-Time Applications using an Approximate Nearest Neighbour Search," *Australasian Conference on Robotics and Automation*, 2012.

[15] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, "6D SLAM with an application in autonomous mine mapping," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 2, 2004, pp. 1998–2003 Vol.2.

[16] A. Nuchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM - 3D mapping outdoor environments," *Fraunhofer IAIS*, vol. 24, 11 2006.

[17] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP variants on real-world data sets," *Autonomous Robots*, 04 2013.

[18] E. Reitbauer, S. Christoph, and H. Didari, "Subterranean positioning for a semi-autonomous robot supporting emergency task forces," *International Conference on Localization and GNSS*, 2022.

[19] A. Segal, D. Hähnel, and S. Thrun, "Generalized-ICP," *Proc. of Robotics: Science and Systems*, 06 2009.

[20] J. Sivic and A. Zisserman, "Efficient Visual Search of Videos Cast as Text Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 591–606, 2009.

Screen-printed Stretchable Capacitive Pad for Tactile and Proximity Sensing

Mohammad Alshawabkeh*, Hosam Alagi†, Stefan Escaida Navarro‡,
Christian Duriez‡, Björn Hein§ and Lisa-Marie Faller*
*Carinthia University of Applied Sciences - ADMiRE Lab, Villach, Austria
Email: m.alshawabkeh@fh-kaernten.at, l.faller@fh-kaernten.at
†Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
Email: hosam.alagi@kit.edu
‡Inria - Lille Nord Europe and University of Lille - CRIStAL, Villeneuve d'Ascq, France
Email: stefan.escaida-navarro@inria.fr, christian.duriez@inria.fr
§Karlsruhe Institute of Technology (KIT) and Karlsruhe University of Applied Sciences, Karlsruhe, Germany
Email: bjoern.hein@kit.edu

***Abstract*—Using the soft robotics in the development of the collaborative robots will be a new innovative method. The flexibility of the soft robots makes them safe and more cooperative. In this work, an additive manufacturing approach is used to develop a soft and stretchable capacitive pad, this approach makes the design process more convenient and easier to provide the sensor with new potential. This paper outlines the evaluation of the soft capacitive pad in terms of tactile and proximity sensing. In future work, this sensing pad will play an important role in the development of a soft robotic gripper. In previous work, the operational principle of similar sensors is discussed, in this work we focus more on the design and the sensor performance, during stretching, for both tactile and proximity aspects. During the evaluation process, a normal tactile force (2 N to 16 N) is applied on the pad, which led to a change in capacitance (3 pF to 7 pF). Also, the sensing pad is evaluated under a strain test by 5 cm. As a result, the capacitance of the tactile sensing layers is increased (0.6 pF to 1.5 pF), and decreasing values for the proximity sensors (1.4 pF and 1.5 pF).**

***Index Terms*—soft robotic grippers, soft sensors, stretchable sensors, capacitive sensor, additive fabrication, 3D-printing.**

I. Introduction

Different robotic applications are contributing into our life including grasping robots [1], [2] and haptics [3]. In addition to human-robot interaction [4], where the safety aspect in this interaction is essential [5].

Due to the fact that the soft robots are flexible and more adaptable than the rigid robots, that need limited and complicated control, this makes the soft robot better suited for human-robot interaction and for dealing with soft and delicate objects.

In order to correctly utilize the soft robotics into various applications, it is important to embed sensors into these soft systems for sensing their environment and precise control. These sensors are used to deliver internal information of the system itself [6], also to give information about the external environment. The capacitive sensing principle is chosen among the others in the soft sensors fabrication due to its robustness and easy manufacturing.

The fabrication of the soft systems is more possible and feasible using new fabrication techniques like multi-material additive manufacturing. For instance, using Galinstan, or silver nanowires (AgNWs) with Polydimethylsiloxane (PDMS)-based layers for flexible and stretchable conductive traces widen the abilities of new high-tech sensors and their utilization.

Furthermore, different complicated tasks can be performed using the integrated sensors including grasping objects [7] and objects identification [8]. In addition to dealing with objects in terms of classification and grasping, these sensors can provide information about the material [9], and detect how rough the object surface is and its temperature. Next section discusses the related literature to manufacturing and capacitive sensors potentials using various fabrication techniques, also how these sensors are utilized in robotic applications. Section III provides overview of the electronic unit used in measuring the changes in capacitance, also it describes sensor design and its fabrication process, in addition to the setup used for conducting the intended experiments. The fourth section presents the experiments steps and the results. Last section views the conclusions and the future steps.

II. Related Work

In the simplest form of a capacitive sensor, two conductive plates/conductors form the capacitor, however, one conductor is enough to provide change in capacitance. These sensors can be manufactured easily and they are ideally fitting to other systems, similar sensors are fabricated using PyzoFlex technology [10]. Furthermore, these sensors can be operated in coupling and proximity modes. For applications, they can be applied in object shape detection [11], tactile and proximity sensing. An important aspect is that the sensor can have different conductive layers in one fabrication process to form a sensor which is capable of sensing both tactile and proximity, simultaneously.

In [12], a carbon nanotube-based sensor is developed with range of $r = 5\,\mathrm{N}$. In [13], a 3D force sensing setup is

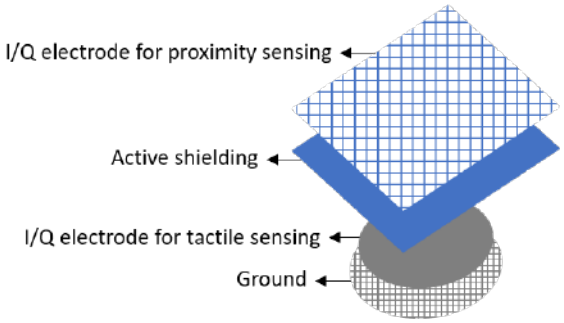


Fig. 1. The main electrodes of the capacitive pad. Where the upper two layers form the proximity sensor, and the lower layers construct the tactile sensor.

developed utilizing capacitive sensors. Also, in [14], a system for directed force measurement is done using an array of capacitive sensors. A high sensitivity capacitive sensors ($s = 3\,\mathrm{mN}$) is developed in [15], where nanoparticles are used in doping silicone. In [16], a capacitive sensor integration with soft robotic finger is shown.

More detailed version of the paper is presented in the IEEE I2MTC (The International Instrumentation and Measurement Technology Conference) 2022, held in Ottawa, Canada.

III. System Description

A. Capacitive Measurement Unit and Sensing Principle

The used electronic measurement unit is designed at the Karlsruhe Institute of Technology (KIT). This unit is the new modified version on the one in [17]. The unit has the following specs: data rate with maximum value of $d_r = 400\,\mathrm{Hz}$, resolution with maximum value of $r = 50\,\mathrm{fF}$, supporting 8 sensors, with both differential and single-ended modes, active shielding, and in-phase-quadrature (IQ) measurement. An exciter frequency of $f_{\mathrm{exc}} = 81\,\mathrm{kHz}$ $f_{exc} = 81\,\mathrm{kHz}$ is applied with single-ended mode in this paper. Where the tactile and proximity sensing are conducted by the IQ-channels, Fig. 1. As described in the previous figure, the active shielding wire provides protection of the proximity sensing layer, and significantly decreases its effect on the tactile sensing layers.

The simplified formula of the parallel-plate capacitor is represented as follows:

$$C = \varepsilon_r \varepsilon_0 \frac{A}{d} \tag{1}$$

Where A refers to the plates' surface area, and d refers to the distance between the plates. This formula provides easier understanding of the relation between the change in capacitance and the physical deformation of the sensor.

The applied normal force on the sensor decreases the distance between the plates, thus the capacitance increases. When stain force is applied to the sensing pad, the area increases while the distance between the plates decreases.

B. Capacitive Sensors Design and Fabrication

The sensing concept behind the sensor is related to the interfering with the sensor's electric fields by human touch or other object. Here, the tactile effect will decrease the distance between the plates where the electric fields are formed. On the other hand, when an object or human body crosses the electric field, this is interpreted as a disturbance in the capacitance. Two modes can be used for measurement; in the single-ended mode the capacitive measurement is achieved between the transmitter and ground. On the other hand, the measurement can be done using differential mode, [18], [19]. The transmitter is connected to an excitation signal with a specific frequency. The cables/electrodes used in the measurements are relatively easy to manufacture using a variety of materials, [20], where they will be durable, tiny, and flexible, also when stretchable material is applied in the fabrication process like Galinstan, these electrodes can be elastic and compressible, [16]. In this paper, the sensor is fabricated layer by layer, such that the tactile sensing part has upper and lower layers/plates, same thing goes for the proximity function, where proximity sensors are placed at the top of the tactile sensors. In Fig. 2, the tactile/force sensors are design from two circular plates with upper and lower diameters of $6\,\mathrm{mm}$ and $8\,\mathrm{mm}$, respectively. On the top of the pad, each proximity sensor has two identical upper and lower layers, where the bottom and top widths are $13\,\mathrm{mm}$ and $6.5\,\mathrm{mm}$, respectively. With main and tapering lengths of $23.5\,\mathrm{mm}$ and $11\,\mathrm{mm}$, respectively. The dielectric layers of the pad are fabricated from EcoFlex 10-fast silicone, while the conductive buses are done from liquid metal, Galinstan. These buses are formed by applying the Galinstan into the cavity of the required printed pattern. where the dielectric material is applied at the top of each Galinstan layer. Finally, coaxial electrodes are sued to make the connection with the end of each Galinstan bus for external read-outs.

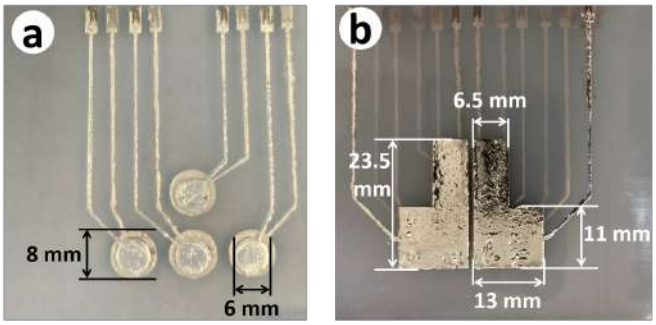


Fig. 2. The capacitive pad sensors design. a) The lower layers of the pad form the tactile sensors. b) The upper layers of the pad form the proximity sensors and covering the overall pad.

C. Evaluation Setup

In this work, the sensing part of the experiment includes the capacitive pad and its sensing unit. While the deformation part consists of the UR3e robot (provides repeatability of +/- $0.03\,\mathrm{mm}$) with custom-made end-effector to conduct each experiment, Fig. 3, this figure also shows the stretching end-effector of the experiment. Where two main experiments are applied to the sensing pad: strain/stretching the pad horizontally; and compressing the pad by a normal force vertically

using the tactile end-effector, Fig. 4. These end-effectors are 3D-printed and are connected to a 3D-force sensor from SCHUNK.

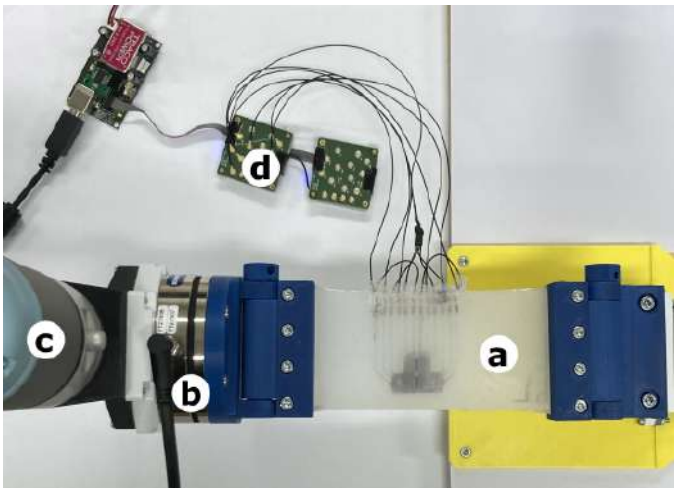


Fig. 3. The experiment setup. The figure shows the used setup in this work, where the same setup shows the end-effector of the strain experiment. a) shows the capacitive pad. b) shows the Schunk-Axia80 force sensor. c) UR3e robot's end-effector. d) the capacitive measurement unit.

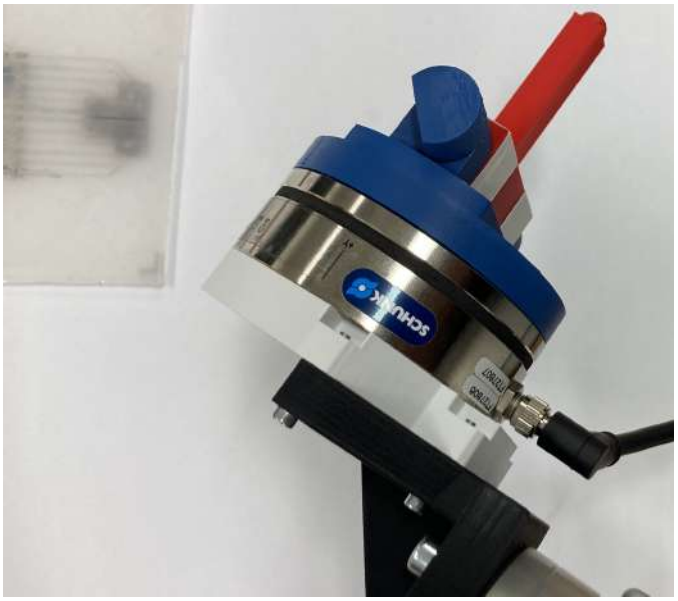


Fig. 4. The tactile end-effector used to apply a normal force on the pad using the 3D-printed red pointer.

IV. Results and Discussion

A. Strain test

This test concludes stretching the pad for a distance of $50\,\mathrm{mm}$ horizontally using the robot, Fig. 3a. While the robot is applying the strain test over a 10 steps, the strain force and the change in capacitance (for tactile and proximity) are logged simultaneously.

After the first strain distance of $5\,\mathrm{mm}$, both sensors show a counter behaviour is semi-linear way. This behaviour can be interpreted by the deformation of the sensor's thickness, where the applied strain increases the pad's length and decreases its thickness; the distance between the capacitors plates. As a result, the capacitance increases.

The capacitance change of the proximity sensors behaves in opposite manner comparing with the tactile sensors. This can be explained by the deformation of the proximity plates, where the strain action lengthens the pad while reducing the area of these plates, as a result, this will decrease the capacitance.

B. Normal tactile force

Different end-effector is 3D-printed with suitable pointing area to apply a normal force on each tactile sensor, Fig. 4. The normal force is applied into the pad for $0.5\,\mathrm{mm}$ per step. This leads to decreasing the distance between the tactile capacitive plates, thus the capacitance will increase, Fig. 5. All tactile sensors share the same behaviour, despite the rate of change for each; this is resultant from a mismatch in the thicknesses of the dielectric and conductive layers during manufacturing.

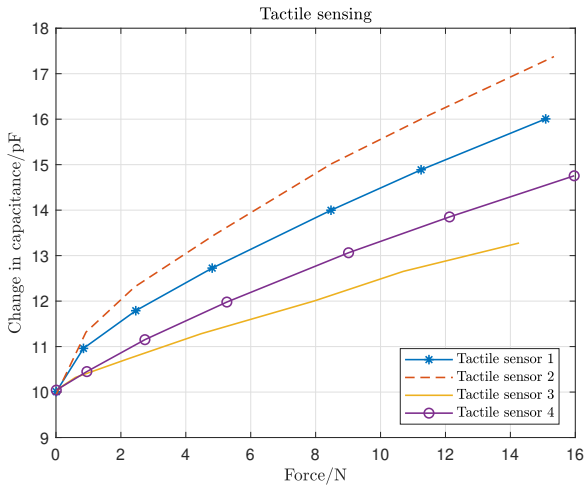


Fig. 5. The change in capacitance value of the tactile sensors under applied normal force.

V. Conclusions

This paper shows the manufacturing process of a stretchable pad with capacitive sensing principle. The pad sensing capabilities; tactile and proximity, are discussed in terms of design, working principle, and evaluation. The use of Galinstan as the conductive material, and the EcoFlex as dielectric and main substrate, demonstrates practical and easy fabrication. The tactile part of the pad consists of four circular sensors, these sensor show similar behaviour against the applied pressure; due to a nonuniform thickness of the fabricated dielectric and Galinstan layers, the sensors vary in terms of sensing magnitudes. The strain test led to decrease the proximity sensors capacitive measurements, as the deformation reduces electrodes areas, while the tactile sensors values increased

as their thickness decreases. The future outlook focuses on evaluating the proximity capability by sensing different materials. Also, utilizing the overall pad, with modified design, into robotics-related application.

ACKNOWLEDGMENTS

The research leading to these results has partially received funding from the Federal Ministry for Digital and Economic Affairs (BMDW) within the framework of COIN "Aufbau", 8th call of the Austrian Research Promotion Agency (FFG) - project number 884136 (iLEAD) and from the "European Regional Development Fund" (EFRE) and "REACT-EU" (as reaction of the EU to the COVID-19 pandemic) by the "Kärntner Wirtschaftsförderungs Fonds" (KWF) in the project Smarter Leichtbau 4.1 (16048 | 34261 | 49704). Also this work was partially funded by the Carl-Zeiss-Foundation within the JuBot Project.

REFERENCES

[1] J. Zhou, S. Chen, and Z. Wang, "A soft-robotic gripper with enhanced object adaptation and grasping reliability," *IEEE Robotics and Automation Letters*, vol. 2, pp. 2287–2293, 2017.

[2] C.-H. Liu, T.-L. Chen, C.-H. Chiu, M.-C. Hsu, Y. Chen, T.-Y. Pai, W.-G. Peng, and Y.-P. Chiang, "Optimal design of a soft robotic gripper for grasping unknown objects," *Soft Robotics*, vol. 5, no. 4, pp. 452–465, Aug. 2018. [Online]. Available: https://doi.org/10.1089/soro.2017.0121

[3] H. Culbertson, S. B. Schorr, and A. M. Okamura, "Haptics: The present and future of artificial touch sensation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 385–409, 2018. [Online]. Available: https://doi.org/10.1146/annurev-control-060117-105043

[4] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Springer International Publishing, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1

[5] D. P. Nguyen, F. Bottarel, M. Hoffmann, U. Pattacini, L. Natale, and G. Metta, "Merging physical and social interaction for effective human-robot collaboration." in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9. [Online]. Available: https://www2.informatik.uni-hamburg.de/wtm/publications/2018/NBHPNM18/08625030.pdf

[6] J. Tapia, E. Knoop, M. Mutný, M. A. Otaduy, and M. Bächer, "MakeSense: Automated sensor design for proprioceptive soft robots," *Soft Robotics*, vol. 7, no. 3, pp. 332–345, Jun. 2020. [Online]. Available: https://doi.org/10.1089/soro.2018.0162

[7] E. W. Hawkes, H. Jiang, D. L. Christensen, A. K. Han, and M. R. Cutkosky, "Grasping without squeezing: Design and modeling of shear-activated grippers," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 303–316, 2018.

[8] L. Chin, M. C. Yuen, J. Lipton, L. H. Trueba, R. Kramer-Bottiglio, and D. Rus, "A simple electric soft robotic gripper with high-deformation haptic feedback," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 2765–2771.

[9] Y. Ding, H. Zhang, and U. Thomas, "Capacitive proximity sensor skin for contactless material detection," 10 2018.

[10] C. Rendl, P. Greindl, M. Haller, M. Zirkl, B. Stadlober, and P. Hartmann, *PyzoFlex: Printed Piezoelectric Pressure Sensing Foil*. New York, NY, USA: Association for Computing Machinery, 2012, p. 509–518. [Online]. Available: https://doi.org/10.1145/2380116.2380180

[11] S. Escaida Navarro, S. Koch, and B. Hein, "3d contour following for a cylindrical end-effector using capacitive proximity sensors," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 82–89.

[12] L. Chen, J. Liu, X. Wang, B. Ji, X. Chen, and B. Yang, "Flexible capacitive hydrogel tactile sensor with adjustable measurement range using liquid crystal and carbon nanotubes composites," *IEEE Transactions on Electron Devices*, vol. 64, pp. 1968–1972, 2017.

[13] S. Somlor, R. S. Hartanto, A. Schmitz, and S. Sugano, "A novel tri-axial capacitive-type skin sensor," *Advanced Robotics*, vol. 29, no. 21, pp. 1375–1391, 2015. [Online]. Available: https://doi.org/10.1080/01691864.2015.1092394

[14] J.-H. Ho, A. Perez, and W.-C. Wang, "Development of a polymeric capacitive 3-d tactile sensor," in *Health Monitoring of Structural and Biological Systems 2010*, T. Kundu, Ed. SPIE, Mar. 2010. [Online]. Available: https://doi.org/10.1117/12.847721

[15] A. Rana, J.-P. Roberge, and V. Duchaine, "An improved soft dielectric for a highly sensitive capacitive tactile sensor," *IEEE Sensors Journal*, vol. 16, no. 22, pp. 7853–7863, 2016.

[16] S. Escaida Navarro, S. Nagels, H. Alagi, L.-M. Faller, O. Goury, T. Morales-Bieze, H. Zangl, B. Hein, R. Ramakers, W. Deferme *et al.*, "A model-based sensor fusion approach for force and shape estimation in soft robotics," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5621–5628, 2020.

[17] H. Alagi, S. Escaida Navarro, M. Mende, and B. Hein, "A versatile and modular capacitive tactile proximity sensor," in *2016 IEEE Haptics Symposium (HAPTICS)*, Apr. 2016, pp. 290–296.

[18] C. Luo, M. A. Borkar, A. J. Redfern, and J. H. McClellan, "Compressive sensing for sparse touch detection on capacitive touch screens," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 639–648, 2012.

[19] L. K. Baxter, Ed., *Capacitive sensors design and applications*. IEEE Press, 1997. [Online]. Available: https://ieeexplore.ieee.org/servlet/opac?bknumber=5264832

[20] L.-M. Faller, M. Krivec, A. Abram, and H. Zangl, "Am metal substrates for inkjet-printing of smart devices," *Materials Characterization*, vol. 143, pp. 211–220, 2018, metal Additive Manufacturing: Microstructures and Properties. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1044580317316273

Comparing Supervised and Unsupervised Machine Learning Methods for Traffic Sign Classification

Peter Hönig[1], Lucas Muster[1], Mohamed Aburaia[1] and Wilfried Wöber[1,2]

***Abstract*— Traffic sign classification is typically implemented using supervised machine learning methods, primarily CNNs. But these methods are prone to overfitting and lack explainability due to their "black box" characteristic. Compared to supervised methods, unsupervised machine learning methods, such as autoencoder and GAN, are great at dimensionality reduction and feature extraction and have already been successfully applied for image classification. To investigate if unsupervised learning methods perform significantly different than supervised learning methods, this study compares performances of a supervised VGG16 classifier, an autoencoder-based classifier and a GAN-based classifier, using the GTSRB dataset. Results show accuracy scores of 98.4% for the VGG16 classifier, 84.7% for the autoencoder-based classifier and 11.0% for the GAN-based classifier.**

I. INTRODUCTION

As stated in the SAE International standard J3016C [15], cars with autonomy level 3 or higher are in need of traffic sign recognition, a method composed of object localization and image classification. State of the art traffic sign classification is performed primarily with convolutional neural networks (CNNs) [18], [16]. Lately, unsupervised learning methods, originally designed for dimensionality reduction and feature extraction [5], have been successfully applied for classification of handwritten digits in [8] and [1], as well as image classification in medical imaging [19]. Based on these successful applications of unsupervised learning methods for classification, this study aims to investigate the effect of supervised and unsupervised training procedures for traffic sign classification, regarding accuracy. In this study we contribute to the state of the art by conducting a comparative study of three classification models. A VGG16 [17] model is compared with two classifiers, based on generative adversarial networks (GANs) [12] and autoencoders [2]. The GAN-based classifier and the autoencoder-based classifier are trained in an unsupervised manner. All three models are trained and tested using the German-traffic-sign-recognition-benchmark (GTSRB) dataset. Accuracy scores are calculated and McNemar tests [14] are conducted to compare the models' performances.

This paper proceeds with an overview of the state of the art in chapter II. Selected methods are described in chapter III. Results are presented and discussed in chapter IV and a final summary and outlook is given in chapter V.

*This work was not supported by any organization

[1]Department of Industrial Engineering, University of Applied Sciences Technikum Wien, 1200 Vienna, Austria; {mr20m009, muster, aburaia, woeber}@technikum-wien.at

[2]Institute for Integrative Nature Conservation Research, University of Natural Resources and Life Sciences, Vienna

II. STATE OF THE ART

In a prior study from 2011 [18], the GTSRB dataset was composed and a classification challenge was held. The highest accuracy score of 98.98% was reached by a supervised CNN classifier with a multilayer perceptron [7]. In 2018 a European dataset [16] was composed and the highest accuracy score of 98.62% was reached with an 8-layer CNN based on the VGG16 model [17]. [10] reviews findings in the field of representation learning, a domain which investigates unsupervised learning and its usage for classification and transfer learning. Unsupervised learning methods can be used as classifiers or as preliminary stages for classifiers. [13] discusses the benefits of unsupervised learning methods and their capabilities of learning representative features and reviews multiple studies where the use of unsupervised learning for classification improved speed and accuracy of subsequent supervised learning tasks. More recent implementations of representation learning include the usage of autoencoders [2] and principal component analysis (PCA) [5], which have already been successfully implemented for classification tasks [8], [1]. As discussed in [4], GANs [9] have been used for representation learning as well [19].

III. MATERIALS AND METHODS

A modified VGG16 [17] classifier, a custom autoencoder [2] and 43 deep convolutional GANs [12] are implemented and trained using an input image size of 32x32, Python 3 and the Keras API [6]. Training parameters are listed in table I. The models are tested with the same test images. Accuracy scores are calculated to compare the models' performances. McNemar tests [14] are conducted to evaluate if the models' performances differ significantly.

A. VGG16 classifier

A VGG16 [17] model is chosen for the supervised CNN classifier. Input image size is defined with 32x32x3. Since VGG16 is built for an input image size of 224x224x3, only three out of five convolution blocks are used, resulting in a latent space size of 8x8x256. Pre-trained weights by ImageNet [11] are used for initialisation.

B. Autoencoder-based classifier

The autoencoder [2] consists of an encoder and a decoder. To build the encoder, the feature extraction layers of the pre-trained modified VGG16 model are used. The decoder is designed as the inverse model of the encoder. After the autoencoder training, the encoder is detached from the decoder and the layers of the encoder are frozen. The frozen weights

of the encoder layers are used for the feature extraction part of a new VGG16 model with randomly initialized flatten and dense layers. This pre-trained classifier with frozen feature extraction is trained again.

C. GAN-based classifier

The GAN-based classifier is a combination of 43 unsupervised deep convolutional GANs [12], using an input image size of 32x32x1. Each of the 43 GANs is trained on a single traffic sign class. After the 43 GANs are trained with their individual traffic sign class images, the discriminators are saved. The GAN-based classifier is represented by the discriminators which act as binary classifiers to distinguish between the images referring to the class they have been trained on and images which do not refer to the class they have been trained on.

TABLE I
MODEL TRAINING PROPERTIES

Model	Input shape	Epochs	Batch size
VGG16 modified	32x32x3	30	878
Autoencoder	32x32x3	50	878
Autoencoder-based classifier	32x32x3	30	878
Each GAN	32x32x1	20 000	32

IV. RESULTS

The unsupervised learning methods did not outperform the supervised learning method in regards to accuracy scores. With an accuracy score of 98.4%, the VGG16 classifier performed as expected, compared to prior traffic sign classification studies, using supervised CNNs [18], [16]. However, it is not guaranteed that the VGG16 model learned representative features. The autoencoder-based classifier reached an accuracy score of 84.7%. The unsupervised feature extraction did not improve the models accuracy score. The GAN-based classifier reached an accuracy score of 11.0%. This indicates that GAN discriminators might not be suited for multiclass classification of traffic sign images, based on the dataset and GAN architecture used. One possible explanation is, that GAN discriminators are trained to distinguish real from fake images. The trained probability distribution of the discriminator seems not to be transferable to distinguish between trained images and novel images from foreign classes. The null-hypothesis of the McNemar tests is that performances of two observed models do not differ significantly [14]. For all McNemar tests, the null-hypothesis was rejected since p-values were below $\alpha = 0.05$, indicating that all models perform significantly differently.

V. SUMMARY AND OUTLOOK

A comparative study was conducted, comparing model performances of a VGG16 classifier with an autoencoder-based classifier as well as a GAN-based classifier. Results show accuracy scores of 98.4% for the VGG16 classifier, 84.7% for the autoencoder-based classifier and 11.0% for the GAN-based classifier. McNemar tests confirmed that the models' performances differ significantly. The supervised VGG16 model performed as expected while the accuracy score of the GAN-based classifier indicates that multiple GAN discriminators, trained on individual classes, are not suited for multiclass classification, based on the training set and GAN architecture used. Future research could apply post-hoc evaluation techniques [3] to investigate if representative features have been learned. Only by understanding if representative features have been learned, CNNs might be applied in fields with strict safety requirements.

REFERENCES

[1] J. Almotiri, K. Elleithy, and A. Elleithy, "Comparison of autoencoder and principal component analysis followed by neural network for e-learning using handwritten recognition," in *2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, USA, May 2017, pp. 1–5.

[2] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, vol. 27, Bellevue, Washington, USA, July 2012, pp. 37–49.

[3] A. Barredo Arrieta *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, June 2020.

[4] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[6] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[7] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 1918–1921.

[8] M. Gogoi and S. A. Begum, "Image classification using deep autoencoders," in *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Coimbatore, India, Dec. 2017, pp. 1–5.

[9] I. Goodfellow *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, p. 139–144, Oct. 2020.

[10] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2656–2666.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84 – 90, 2012.

[12] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2016.

[13] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[14] L. Sachs and J. Hedderich, *Angewandte Statistik. Methodensammlung mit R*, 12nd ed. Berlin: Springer, 2006.

[15] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE International Std. J3016 202 104, 2021.

[16] C. G. Serna and Y. Ruichek, "Classification of traffic signs: The european dataset," *IEEE Access*, vol. 6, pp. 78 136–78 148, Dec. 2018.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," presented at the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 2015.

[18] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *The 2011 International Joint Conference on Neural Networks*, San Jose, CA, USA, Aug. 2011, pp. 1453–1460.

[19] D. R. Yerukalareddy and E. Pavlovskiy, "Brain tumor classification based on mr images using gan as a pre-trained model," in *2021 IEEE Ural-Siberian Conference on Computational Technologies in Cognitive Science, Genomics and Biomedicine (CSGB)*, Novosibirsk - Yekaterinburg, Russia, May 2021, pp. 380–384.

Evaluation of Reinforcement Learning Algorithms for the online 3D Bin Packing Problem*

Johanna Gottwald[1], Patrick Streck[1], Maximilian Beinhofer[1], Markus Hofmarcher[2], Michael Widrich[2]

***Abstract*— Optimizing the filling degree of load carriers packed by robots is typically done by using heuristics. Missing information about future item sizes and the possibility of rotating items increases the intricacy of the optimization problem. As Reinforcement Learning (RL) has been shown to be efficient in solving combinatorial optimization problems (e.g. Alpha GO), this paper looks into the performance of RL algorithms on the online 3D bin packing problem with rotation. For this, three RL algorithms were tested on various settings and their performance was compared to known heuristics. The experiments showed that depending on the setting, RL is able to achieve a similar or better performance than the heuristics.**

I. INTRODUCTION

In warehouse logistics, maximizing the occupied volume per load carriers (LC) has a high commercial value, as e.g. costs per shipment and packing can be reduced. The resulting combinatorial optimization problem is known as bin packing problem (BPP) [6]. While in *offline* bin packing (BP), the sizes of all pending items are given, in *online* BP only the size of the current item is known. Thus, in online BP no preliminary item ordering can be performed, which makes it more interesting for robotics, as less complex warehouse solutions are needed for supplying the individual items. As generalizing to varying situations can be problematic for heuristics, Reinforcement Learning (RL) could be an alternative for solving the BPP [8].

There exists a broad range of research dealing with the application of RL on the 3D BPP. While some papers propose novel algorithms [11], others focus on the combination of RL with other methods like attention networks [12][7]. This paper aims at getting an insight into the possible performance of different state-of-the-art RL algorithms on the *online 3D BPP with rotation*. Therefore, a virtual environment of a 3D BPP was developed in which the three RL algorithms *Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC)* and *RUDDER* were trained and their performance on various settings was evaluated and compared to the one of heuristics.

II. PROBLEM STATEMENT

In this paper, an online version of the 3D BPP with rotation is considered. As each object can be represented by its bounding box, all items sampled over a finite time interval are assumed to be cuboid. After sampling each item has to be placed inside an LC, making use of a 90° rotation of the item wrt. its z-axis if desired. An item assignment is considered valid if the entire item is placed inside an LC without exceeding its borders. Once an item is put into an LC, its position cannot be changed. Moreover, items can only be placed on top of an LCs' content and cannot be shifted below other items. Furthermore, at any time step, only one LC is available for item assignment. If no valid position can be found for the current item, the LC is closed and a new one is used to place the item. Overall, the goal is to minimize the total number of LCs needed to pack all sampled items.

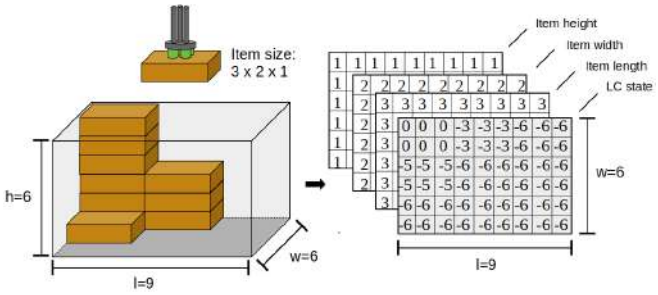


Fig. 1. Example of a 4D tensor encoding a LC state and the size of the current item as feature maps. The stated measurements are given in voxels.

III. ENVIRONMENT SETTING

RL is a goal-directed computational approach, where an agent (a learner) is trained by interacting with its environment [10]. Based on the action performed in a certain state, it receives a numerical reward signal and the next state. Overall, the agent aims to maximize the total reward over time.

A. State Space

The state space is used to encode the discretized LC state and size of the sampled item. For this, a 4D tensor is used (example given in figure 1), where the first dimension contains a map of the negative of the remaining heights at the individual LC positions in voxels. The remaining tensor dimensions comprise mappings containing the length, width and height of the current item in voxels in each matrix entry. This encoding was chosen to enable convolutional neural networks (CNNs) to relate the LC state with the item size.

B. Action Space

For item assignment, the agent has to decide on a target position in the LC and if the item shall be rotated. Therefore, a discrete tuple action space is used, where the first entry contains a boolean value, which decides the rotation action (0: no rotation, 1: rotation by 90°). The second entry consists of a vector encoding the LC positions and an action for

*This work was partly supported by the Austrian Research Promotion Agency (FFG) under grant 873923.

[1]J. Gottwald, P. Streck and M. Beinhofer are with TGW Logistics Group GmbH, Marchtrenk, Austria forename.surname@tgw-group.com

[2]M. Hofmarcher and M. Widrich are with the Institue of Machine Learning, Johannes Kepler University, Linz, Austria surname@ml.jku.at

TABLE I
EXPERIMENTAL SETUP AND RESULTS

	LC Setting $l \times w \times h$	Item Settings $l \times w \times h$	Items per Episode	First Fit	Floor Building	Column Building	Random Agent	PPO	SAC	RUDDER
Setting 1	$13 \times 13 \times 12$	$3 \times 3 \times 3$	128	**85.207%**	**85.207%**	**85.207%**	32.647%	**85.207%**	**85.207%**	56.805%
Setting 2	$12 \times 29 \times 16$	$4 \times 7 \times 4$	132	53.103%	53.103%	53.103%	29.829%	66.379%	**88.506%**	66.379%
Setting 3	$13 \times 25 \times 16$	$4 \times 6 \times 4$, $8 \times 4 \times 4$	108	53.048%	77.079%	48.965%	29.096%	**77.434%**	76.578%	56.883%
Setting 4	$13 \times 21 \times 18$	$4 \times 5 \times 6$, $10 \times 4 \times 3$	108	38.113%	52.199%	40.115%	27.295%	42.063%	49.871%	**54.089%**
Setting 5	$13 \times 20 \times 16$	l, w, h in rage $[2, 6]$	100	36.449%	**47.731%**	33.551%	26.234%	38.305%	43.127%	39.469%

The table shows the experimental setup as well as the average obtained filling degrees per episode in percentage. Each setting was tested for 500 episodes with 5 different seeds, with the episode length defined by the number of items to be packed.

closing the current LC and opening a new one. In the latter case, the item is assigned to the top-left LC corner, utilizing the chosen rotation action. The selected position encodes the xy-coordinate, to which the top-left item corner is assigned. For simplicity, it is assumed, that an item is placed with its edges parallel to the ones of the LC. To avoid invalid item assignments (e.g. collisions with the LC or over-height) an action mask is used to filter those actions beforehand. Further, as a perfectly centred position of the item on the gripper of the robot is unlikely in practice, a tolerance of one voxel is considered in the mask wrt. the item length and width. Moreover, opening a new LC is only enabled by the mask, if no valid assignment for the given item exists in the current LC.

C. Reward

If an LC is closed, the negative percentage of the empty LC space is given as reward. Otherwise, the reward is zero.

IV. APPROACHES

We tested the three RL algorithms PPO, SAC and RUDDER on the described problem setting and compared their results to the performance of the *First Fit (FF)*, the *Floor Building (FB)* and the *Column Building (CB)* heuristic as described by [11]. Additionally, the performance of an agent which randomly interacts with its environments, under consideration of the action mask, was used as a lower bound.

A. Heuristics

To select a target position, each of the heuristics iterates over all LC positions row-by-row, starting at the top-left corner. While FF assigns items to the first feasible position in the LC found during iterating, FB chooses the position that leads to the lowest feasible item location in the LC. In contrast, CB selects the highest possible location in the LC. If no position is found, the iteration is repeated with the rotated item. If still no feasible position is found, the item is assigned to the top-left corner of a new LC.

B. Proximal Policy Optimization

PPO [9] is an *on-policy* method, which learns a policy by using a clipped objective function, which ensures that the current policy is not destroyed by too large update steps. In each iteration, several parallel actors collect data by interacting with the environment for a certain amount of steps. Afterwards, the collected experience is used to update the networks via gradient ascent on the objective function.

C. Soft Actor-Critic

SAC [4] is an *off-policy* method, where an agent should succeed at a task by interacting as randomly as possible with its environment [4]. Hence, in addition to the expected reward, the entropy is maximized. The obtained experience is collected in a buffer, which is used to train a policy. We used the discrete version of SAC as described in [2].

D. RUDDER

Assigning credit to actions that are responsible for rewards in the future is a challenging task in RL [1]. RUDDER can be used on top of other RL methods to tackle this problem. By using a Long Short-Term Memory (LSTM)[5], the rewards of an episode are redistributed and assigned to actions causing delayed rewards. We use RUDDER in combination with PPO, as described in [1].

V. EXPERIMENTS

We have evaluated the approaches described in section IV on five settings of item- and LC-sizes. The respective settings and results are listed in Table I. The item dimensions were selected to cover different configurations, ranging from a single item of cubic shape to several cuboid items of random sizes in a given range. While in some settings the RL algorithms gave better results than the heuristics (eg. Setting 2), in others at most similar performance was achieved (eg. Setting 5). Comparing the results of the RL algorithms, the off-policy method SAC on average achieved the highest filling degrees. Further, RUDDER too showed the capability of achieving a good performance (eg. Setting 4).

VI. CONCLUSION AND FUTURE WORK

In this paper, we tested three RL algorithms on various settings of the 3D BPP. We compared their performance with known heuristics and showed that RL has the capability of matching or improving the performance of the selected heuristics. Due to the computational time needed for training (especially with RUDDER), only a small range of hyper-parameters was tested. By performing further hyper-parameter experiments, the performance of the RL agents could be improved (e.g. for RUDDER on Setting 1 and 3). Moreover, runs with additional seeds should be performed and statistical significance tests are needed to consolidate the results. Furthermore, the algorithms were only tested in a virtual setting until now. Hence, a realization utilizing a pick-robot needs to be evaluated in a further step.

REFERENCES

[1] J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter, "Rudder: Return decomposition for delayed rewards," 2019.

[2] P. Christodoulou, "Soft actor-critic for discrete action settings," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1910.07207

[3] J. Csirik and G. J. Woeginger, *On-line packing and covering problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 147–177. [Online]. Available: https://doi.org/10.1007/BFb0029568

[4] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *CoRR*, vol. abs/1801.01290, 2018. [Online]. Available: http://arxiv.org/abs/1801.01290

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[6] M. Hofri, *Probabilistic Analysis of Algorithms: On Computing Methodologies for Computer Algorithms Performance Evaluation*, ser. Texts and Monographs in Computer Science. Springer, 1987. [Online]. Available: http://gen.lib.rus.ec/book/index.php?md5=ae7fcb22b3a2600c49cdef010adf021a

[7] Y. Jiang, Z. Cao, and J. Zhang, *Solving 3D Bin Packing Problem via Multimodal Deep Reinforcement Learning*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2021, p. 1548–1550.

[8] O. Kundu, S. Dutta, and S. Kumar, "Deep-pack: A vision-based 2d online bin packing algorithm with deep reinforcement learning," pp. 1–7, 2019.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv.org*, Aug 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[10] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[11] R. Verma, A. Singhal, H. Khadilkar, A. Basumatary, S. Nayak, H. V. Singh, S. Kumar, and R. Sinha, "A generalized reinforcement learning algorithm for online 3d bin-packing," 2020.

[12] J. Zhang, B. Zi, and X. Ge, "Attend2pack: Bin packing through deep reinforcement learning with attention," 2021.

Time-Optimal Swing-Up Trajectory Optimization for a Spherical Inverted Pendulum Mounted on a Linear Robot

Simon Schmidt[1] and Hubert Gattringer[1] and Andreas Müller[1]

***Abstract*— This paper presents the time-optimal swing-up of a spherical pendulum (SP) mounted at the end-effector (EE) of an elastic linear robot. For the optimization, the pendulum is separated and represented as a dynamical model with moving base point. The corresponding Optimal Control Problem (OCP) comprises the cost function (minimizing time and jerk) and the physical constraints accounting for the constraints on position, velocity, and acceleration limits on the base point (i.e. the EE). The solution is calculated with a multiple shooting method leading to the desired swing-up trajectories. The stabilization of the pendulum is ensured by means of a time-variant Linear Quadratic Regulator (LQR) that is designed along the desired trajectories. Experimental results of the swing-up and the stabilization of the pendulum at its upper equilibrium are presented.**

I. INTRODUCTION

Inverted pendulums are often studied benchmark examples in the scientific community although they do not really have a direct industrial use. However, their importance for education and engineering curricula is undoubted since sophisticated methods for dynamical modeling, parameter identification, control and optimization can be studied. In this context, different setups like single, double, and triple pendulums on carts moving on a linear axis, rotary units or spherical units exist. An overview and history of inverted pendulums can be found in [6]. In [4], a spherical pendulum (SP) on a 7-axis LWA Kuka robot is considered. They use a special tailored swing-up optimization and a time-variant LQR controller. As extension a fast re-planning method using QP optimization is discussed in [9]. [8] present a method for swing-up of an SP using energy based methods. Therein only simulation results are available and the performance is less in contrast to optimization based swing-up trajectories. The LQR stabilization (without swing-up) of an SP mounted on the end-effector (EE) of a redundant robot is shown in [7]. The nullspace of the redundant robot is used for e.g. obstacle avoidance. An inverted pendulum with 1 degree of freedom (DOF) mounted on an industrial robot is also considered in [11] where a special algorithm for the erecting is demonstrated. An impressive swing up and stabilization of a triple pendulum mounted on a linear moving cart is reported in [3]. They designed special non-linear controllers for this purpose. Concepts from discrete mechanics and optimal control (DMOC) for the control of a Furuta (rotary) double pendulum can be found in [5]. Therein simulation results are presented only.

In this contribution, we discuss the time-optimal swing-up and stabilization of an SP mounted on the EE of a large scale linear robot. In contrast to the papers mentioned above, the robot used for erecting the pendulum is inherently elastic due to its design. We use the multiple shooting method to calculate the time-optimal trajectories, and a time-variant LQR controller for the stabilization. A system overview of the setup is shown in Fig. 1. To be able to better differentiate between the robot and the SP, they are separated in this figure. The modeling and optimization is done on a standard computer where a code generation is used to generate the real-time code for an *Automation PC* (*APC*) running at a sampling rate of 800 µs. Hardware (*APC*, torque controller - *ACOPOS*, input Modules -*X*20) from *B&R Industrial Automation* is used as overall control device. The torque controller communicate with the *APC* via *Powerlink* bus while the encoders of the SP are read in on an *X*20 device that communicates via *X*2*X* bus system. The SP is a special design with a mass of 0.4 kg and a length of $l_2 = 0.66$ m. *Scancon* encoders of type *SCH*16*F* resolve one revolution with 5000 counts, see Fig. 2 for a photo of the buildup. Light weight cables are used for the connection of the encoders to the *X*20 modules not to disturb the swing-up. The workspace of the linear robot is a cube with dimensions of about $4 \times 2 \times 2$ m. The linear robot has 5 DOF (3 linear, 2 rotatory at the wrist). In this study, only the 3 linear DOF are used. The hardware limits of the robot can be found in Tab. I, which are to be respected in the optimal control.

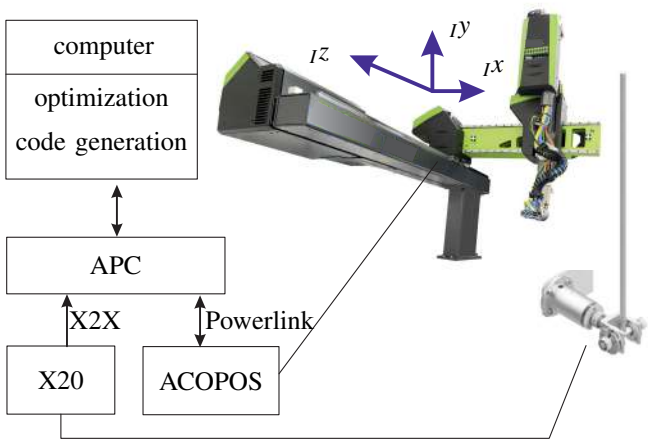


Fig. 1. System Overview

The paper is organized as follows. Section II presents

[1]Simon Schmidt, Hubert Gattringer, Andereas Müller are with Institute of Robotics, Johannes Kepler University Linz, 4040 Linz, Austria {simon.schmidt,hubert.gattringer,a.mueller}@jku.at

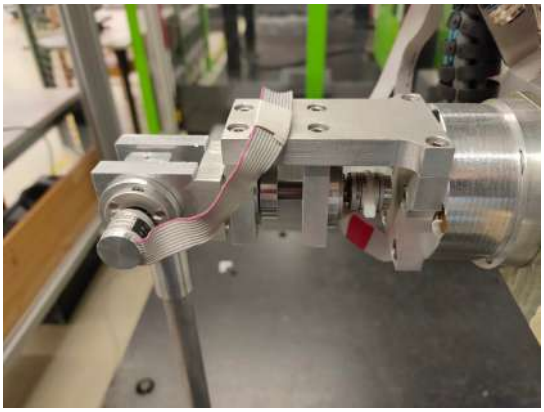


Fig. 2. Photo of spherical joint

the dynamical modeling of the SP. The formulation of the optimal control problem (OCP) can be found in Sect. III. Details about the used control concept gives Sect. IV. Experimental results conclude the paper in Sect. V.

II. Setup and Modeling

For the optimization and control, the inverted pendulum is separated from the robot, see Fig. 3. It can be described by two minimal coordinates, namely the two pendulum angles,

$$\mathbf{q}_P^T = (q_{P1}\ q_{P2}). \tag{1}$$

The motion is prescribed by the EE position $\mathbf{r}_E(t)$ of the elastic robot. The Lagrange equations serve as dynamic equations of motion (EOM) written as

$$\mathbf{M}(\mathbf{q}_P)\ddot{\mathbf{q}}_P + \mathbf{g}(\dot{\mathbf{q}}_P, \mathbf{q}_p) = \mathbf{B}(\mathbf{q}_P)\mathbf{u}, \tag{2}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{g}$ contains the Coriolis, centrifugal and gravitation effects. Also Coulomb and viscous friction are included in $\mathbf{g}$. The acceleration of the EE $\mathbf{u} = \ddot{\mathbf{r}}_E$ serves as control input, which enters the EOM via the control input transformation matrix $\mathbf{B}(\mathbf{q}_P)$ that depends on the pendulums configuration $\mathbf{q}_P$. For control and simulation purposes, the system is transformed to state space form. With

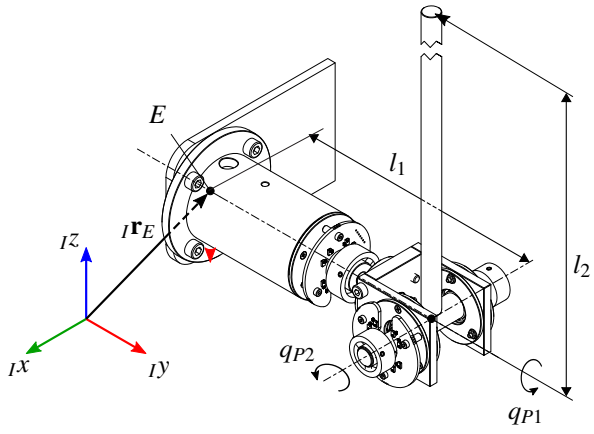


Fig. 3. Sketch of spherical pendulum

the state vector, containing also the position and velocity of the EE,

$$\mathbf{x}^T = (\mathbf{q}_P^T\ \dot{\mathbf{q}}_P^T\ \mathbf{r}_E^T\ \dot{\mathbf{r}}_E^T) \tag{3}$$

the dynamical model is

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{q}}_P \\ \mathbf{M}^{-1}(\mathbf{B}(\mathbf{q}_P)\mathbf{u} - \mathbf{g}(\mathbf{q}_P, \dot{\mathbf{q}}_P)) \\ \dot{\mathbf{r}}_E \\ \mathbf{u} \end{pmatrix} := \mathbf{f}(\mathbf{x}, \mathbf{u}). \tag{4}$$

III. Optimal Control Problem

The goal of the optimization is to find a feasible solution for the motion of the pendulum from the lower to the upper equilibrium in minimal time. Since the optimized trajectories will be executed by the physical robot, which is elastic, it is desirable to ensure smoothness of the trajectories. In order to obtain such a smooth trajectory, the jerk of the EE is used as input for the OPC

$$\mathbf{u}_{opt} = \dddot{\mathbf{r}}_E. \tag{5}$$

Therefore, the overall state used in the optimization problem is set to

$$\mathbf{x}_{opt}^T = \begin{pmatrix} \mathbf{x}^T\ \dot{\mathbf{r}}_E^T \end{pmatrix}. \tag{6}$$

The corresponding dynamical system is

$$\dot{\mathbf{x}}_{opt} = \begin{pmatrix} \dot{\mathbf{x}} \\ \mathbf{u}_{opt} \end{pmatrix} := \mathbf{f}_{opt}(\mathbf{x}_{opt}, \mathbf{u}_{opt}). \tag{7}$$

The variables to be optimized are the final swing-up time t_e and the jerk $\mathbf{u}_{opt}(t)$ of the EE. The overall OPC can be formulated as

$$\min_{t_e, \mathbf{u}_{opt}} \int_0^{t_e} \left(1 + k\mathbf{u}_{opt}^T\mathbf{u}_{opt}\right) dt \tag{8}$$

subject to

$$\mathbf{x}_{opt,min} \le \mathbf{x}_{opt} \le \mathbf{x}_{opt,max} \tag{9}$$

$$\mathbf{u}_{opt,min} \le \mathbf{u}_{opt} \le \mathbf{u}_{opt,max} \tag{10}$$

$$\dot{\mathbf{x}}_{opt} = \mathbf{f}_{opt}(\mathbf{x}_{opt}, \mathbf{u}_{opt}) \tag{11}$$

$$\mathbf{x}_{opt}(0) = \mathbf{x}_{opt,0} \tag{12}$$

$$\mathbf{x}_{opt}(t_e) = \mathbf{x}_{opt,e} \tag{13}$$

The cost function (8) is a trade-off between time-optimal solution and the minimum overall jerk. This trade-off can be controlled by the weight k. Since the focus is on fast trajectories, a small k is used. Inequality constraints (9) account for physical constraints on the pendulum (position and velocity) as well as on the robot (EE position, velocity and acceleration). For a specific example, considered in the next section the used values can be found in Tab. I. Equation (10) represent jerk input constraints, while equality constraints (11) account for the system dynamics (7).

Initial and terminal constraints in (12) and (13) for the pendulum are

$$\mathbf{q}_P^T(0) = (-\pi\ 0), \quad \mathbf{q}_P^T(t_e) = (0\ 0), \tag{14}$$

$$\dot{\mathbf{q}}_P(0) = 0, \quad \dot{\mathbf{q}}_P(t_e) = 0, \tag{15}$$

TABLE I
LIMITS ON THE ROBOT MOTION

	min	max	unit
q_{P1}	-2π	2π	rad
q_{P2}	-0.1	0.1	rad
$\dot{\mathbf{q}}_P$	-20π	20π	rad/s
r_{Ex}	$x_{E0}-0.1$	$x_{E0}+0.1$	m
r_{Ey}	$y_{E0}-0.25$	$y_{E0}+0.1$	m
r_{Ez}	$z_{E0}-0.75$	$z_{E0}+0.75$	m
$\dot{r}_{Ex}$	-1	1	m/s
$\dot{r}_{Ey}, \dot{r}_{Ez}$	-1.5	1.5	m/s
$\ddot{r}_{Ex}$	-2.5	2.5	m/s^2
$\ddot{r}_{Ey}, \ddot{r}_{Ez}$	-3.5	3.5	m/s^2
$\mathbf{u}$	-100	100	m/s^3

and for the robot they are

$$\mathbf{r}_E(0) = \mathbf{r}_{E0}, \quad \mathbf{r}_E(t_e) = \mathbf{r}_{Ee}, \tag{16}$$

$$\dot{\mathbf{r}}_E(0) = 0, \quad \dot{\mathbf{r}}_E(t_e) = 0, \tag{17}$$

$$\ddot{\mathbf{r}}_E(0) = 0, \quad \ddot{\mathbf{r}}_E(t_e) = 0. \tag{18}$$

The OCP is solved with the direct Multiple Shooting method [2] using N shooting intervals. The ODE (11) is integrated numerically with a Runge-Kutta scheme of $4th$ order. This shooting method is implemented within the CASADI framework (an open-source tool for nonlinear optimization) [1] along with the IPopt solver (a software package for large-scale nonlinear optimization using an interior point method) [10].

IV. CONTROL

For the stabilization of the pendulum on the desired trajectory and the upper equilibrium position a time-variant Linear Quadratic Regulator (LQR) is utilized. The desired trajectory $(\mathbf{x}_d, \mathbf{u}_d)$ is the result of the OCP substituted into the state vector (3) ($\mathbf{x}_d$ equals $\mathbf{x}_{opt}$ reduced by $\ddot{\mathbf{r}}_{E,opt}$; $\mathbf{u}_d = \ddot{\mathbf{r}}_{E,opt}$). By introducing the deviations $\mathbf{x} = \mathbf{x}_d + \Delta\mathbf{x}$ and $\mathbf{u} = \mathbf{u}_d + \Delta\mathbf{u}$ for the state and the input, respectively, the linearized system reads

$$\Delta\dot{\mathbf{x}}_i = \mathbf{A}_i\Delta\mathbf{x}_i + \mathbf{B}_i\Delta\mathbf{u}_i \tag{19}$$

where $i = 1,\ldots,N$ accounts for the discretization of the desired trajectory from the shooting method. The system matrices $\mathbf{A}_i, \mathbf{B}_i$ can be calculated from (4) by

$$\mathbf{A}_i = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\Big|_{\mathbf{x}_{d,i},\mathbf{u}_{d,i}}, \quad \mathbf{B}_i = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right)\Big|_{\mathbf{x}_{d,i},\mathbf{u}_{d,i}}. \tag{20}$$

The LQR minimizes the cost function (weighting between squared state values by $\mathbf{Q}$ and squared input values by $\mathbf{R}$)

$$\min J_i = \min \frac{1}{2}\int_0^t \Delta\mathbf{x}_i^T\mathbf{Q}\Delta\mathbf{x}_i + \Delta\mathbf{u}_i^T\mathbf{R}\Delta\mathbf{u}_i dt \tag{21}$$

resulting in an optimal (time-variant) control law

$$\Delta\mathbf{u}_i = \mathbf{K}_i\Delta\mathbf{x}_i \tag{22}$$

along the desired trajectory. Fig. 4 shows the overall control scheme. The LQR ($\Delta\mathbf{u}$) delivers in combination with the desired accelerations from the optimization $\ddot{\mathbf{r}}_{E,d}$ the required control accelerations $\ddot{\mathbf{r}}_{E,c}$. To use them as input variables for the position controlled robot, these values have to be integrated twice. An additional notch filter accounts for the vibrations due to the elastic robot. It is designed to cancel out the first eigenfrequency of the robot. The filtered signal $\mathbf{r}_{E,c}$ is the desired position for the robot. A standard PD-controller calculates the motor torques $\mathbf{M}$ of the robot drives. The state $\mathbf{x}$ containing the EE-position/velocity of the robot $\mathbf{r}_E, \dot{\mathbf{r}}_E$ as well as the pendulum angles/velocities are measured and fed back.

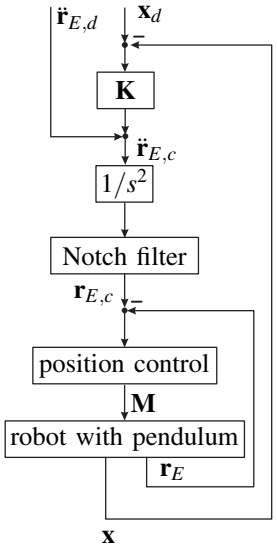


Fig. 4. Control Scheme

V. EXPERIMENTAL RESULTS

The experimental implementation of the previous sections shows details for the swing-up and stabilization. A discretization of the OCP of $N = 100$ intervals leads to a swing-up time of the SP of $t_e = 2.1$s. The optimization took $t_{cpu} = 8$s on a standard computer. The weighting between time- and jerk-optimal input is chosen to $k = 10^{-5}$. Start- and endpoint of the robot swing-up positions are defined equal as

$$\mathbf{r}_{E0} = \mathbf{r}_{Ee} = \begin{pmatrix} 0.2 \\ -0.6 \\ 1 \end{pmatrix} m. \tag{23}$$

Alternatively, the endpoint of the robot can be set free (within the constraints), leading to a slightly lower swing-up time. The weighting matrices for the LQR design are chosen constant as

$$\mathbf{Q} = diag[1,1,10^{-4},10^{-4},10,10,10, 10^{-2},10^{-2},10^{-2}] \tag{24}$$

$$\mathbf{R} = diag[1,1,1] \tag{25}$$

for all discretization steps. Fig. 5 shows desired and measured angles of the pendulum. It can be seen that the swing-up is mainly done by the first pendulum angle q_{P1} that changes from $-\pi$ rad to 0 rad. Angle q_{P2} has only small deviations around 0 rad. The measured and the desired values

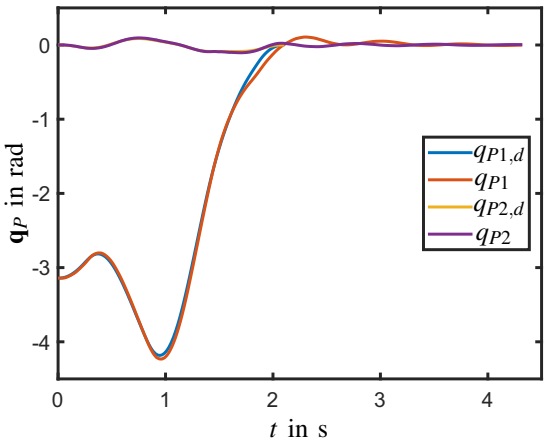


Fig. 5. Pendulum angles $\mathbf{q}_P$

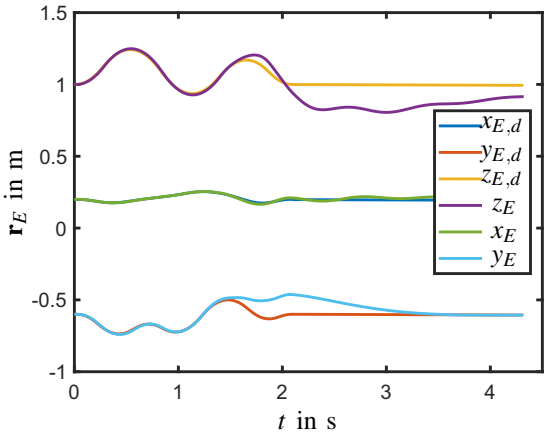


Fig. 6. EE-Positions of the robot $\mathbf{r}_E$

coincide very well. A comparison between the desired- and the measured EE-positions can be seen in Fig. 6. Again the signals coincide very well, especially in the first 1.5 s. There are small deviations in the last phase of the swing-up in the time range between 1.5 and 2.1 s since this critical for a successful experiment. It seems that this deviations start approximately at a horizontal angle q_{P1} of $-\pi/2$, which is a singular position for this problem. Nonetheless, the overall experiment works very well. The critical phase in the swing-up starting at 1.5 s can also be seen in the desired controller accelerations $\ddot{\mathbf{r}}_{E,c}$ in Fig. 7 as increased control action.

A youtube video of the swing-up experiment as well as some changes of stabilized set-points can be found at `https://www.youtube.com/watch?v=_naHO1ECrf8` .

VI. SUMMARY AND OUTLOOK

This paper presents the successful swing-up of a custom built spherical pendulum mounted at the EE of an elastic linear robot. Future work will focus on calculation of optimal trajectories for different (unstable) set-points. In a further step the whole robot dynamics including the wrist degrees of freedom will be used. Up to now, the parameters of the pendulum used in the model and optimization are taken

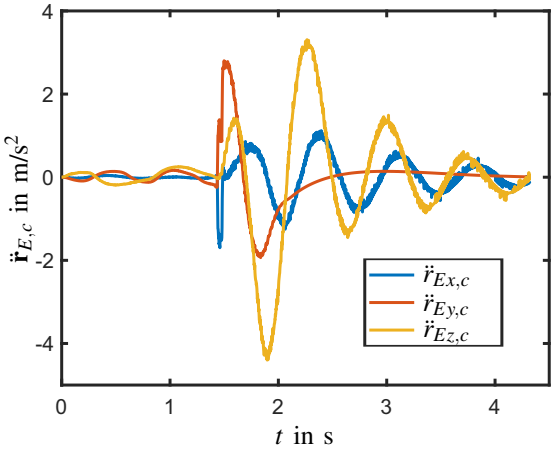


Fig. 7. Control signal $\ddot{\mathbf{r}}_{E,c}$

from CAD data. The accuracy of these parameters can be increased by an identification process.

ACKNOWLEDGMENT

This work has been supported by the "LCM – K2 Center for Symbiotic Mechatronics" within the framework of the Austrian COMET-K2 program.

REFERENCES

[1] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[2] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," in *Fast motions in biomechanics and robotics*. Springer, 2006, pp. 65–93.

[3] T. Glück, A. Eder, and A. Kugi, "Swing-up control of a triple pendulum on a cart with experimental validation," *Automatica*, vol. 49, no. 3, pp. 801–808, 2013.

[4] C. Hartl-Nesic, J. Kretschmer, M. Schwegel, T. Glück, and A. Kugi, "Swing-up of a spherical pendulum on a 7-axis industrial robot," *IFAC-PapersOnLine*, vol. 52, no. 15, pp. 346–351, 2019.

[5] J. Ismail and S. Liu, "Efficient planning of optimal trajectory for a furuta double pendulum using discrete mechanics and optimal control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 456–10 461, 2017.

[6] K. H. Lundberg and T. W. Barton, "History of inverted-pendulum systems," *IFAC Proceedings Volumes*, vol. 42, no. 24, pp. 131–135, 2010.

[7] G. Schreiber, C. Ott, and G. Hirzinger, "Interactive redundant robotics: Control of the inverted pendulum with nullspace motion," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 158–164.

[8] A. Shiriaev, H. Ludvigsen, and O. Egeland, "Swinging up of the spherical pendulum," *IFAC Proceedings Volumes*, vol. 32, no. 2, pp. 2193–2198, 1999.

[9] M. N. Vu, C. Hartl-Nesic, and A. Kugi, "Fast swing-up trajectory optimization for a spherical pendulum on a 7-dof collaborative robot," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 114–10 120.

[10] A. Wächter and L. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[11] A. Winkler and J. Suchỳ, "Erecting and balancing of the inverted pendulum by an industrial robot," *IFAC Proceedings Volumes*, vol. 42, no. 16, pp. 323–328, 2009.

Door Pose Estimation and Robot Positioning for Autonomous Door Opening

Ulrich Mitterhuber[1], Raimund Edlinger[1], Roman Froschauer[1] and Andreas Nüchter[2]

***Abstract*— For autonomous robots to deliver value in human centered environments, they must be able to autonomously open doors. For doing so, they have to overcome multiple challenges, one of which is, to estimate the desired door's orientation and position. The information if the door handle is on the left or right side of the door must also be obtained. In this work, a novel method, solving the stated problems is proposed. It's perception is based on a sensor fusion of a monocular camera based state-of-the-art deep learning object detection algorithm with a 2D laser scan and subsequent line estimation. Additionally, a differential drive controller, using the advancement of continuous goal pose updating, is proposed. During real-world experimentation with a differential drive robot, the implemented system was able to position the robot in front of a door every time with sufficient accuracy and is thus found to solve the stated problem successfully.**

I. Introduction

Autonomous mobile robots can be used in many cases such as retail, industrial site monitoring or rescue robotics just to name a few. To enable them operating in human environments they must be capable to detect doors and door handles, position themselves in front of them, manipulate them and subsequently traverse through them. A door traversing behaviour could be utilised in a fully autonomous application such as in industrial site monitoring or as a semi autonomous assistance function in for example search and rescue applications.

The aim of this work is to tackle the above stated problem by solving the first tasks required for autonomous door opening. More specifically, a robot behaviour for detecting a previously unknown door, estimating it's pose, determining the side at which the door handle is located and then positioning the robot in front of the door at a specific distance, is implemented. The implemented methods aim to be executed in real time on embedded computing platforms commonly used on autonomous robots.

To fulfill the stated objective, multiple algorithms were developed which require the sensor input from a calibrated monocular RGB camera with a minimum resolution of 320x240 pixels, a 2D laser scanner covering the horizontal field of view (FOV) of the camera and an odometry source for robot state estimation. The robots also must provide a velocity controller which accepts a linear and angular velocity. For door pose estimation, a state-of-the-art deep learning based object detection algorithm, more specifically *YOLOV5*[1], is used to detect doors within camera images. This information is then fused with a laser scan to determine the pose of the door by using line estimation. For that, three different line estimation algorithms are implemented and compared experimentally. The side of the door handle in the door is estimated using the output of the object detection. Lastly, a differential drive controller using the contribution of continuous goal pose updating is implemented.

The method relies on the conditions that before starting, the robot is less than laser scanner's maximum measurement distance away from the door, there are no obstacles between the robot and the door and the door is within the FOV of the camera.

In the following pages first, in sec. II the current state-of-the-art in door opening is examined. After that, sec. III and sec. IV show the implemented methods with their postulated hypothesis and their experimental validation respectively. Sec. V sums up the findings and provides an outlook on possible fields of applications and further improvements of this work.

II. Related Work

The field of research in opening and detecting doors and estimating their pose has been very active for many years. However, especially since the rise of deep learning, many methods based on convolutional neural networks were proposed.

An example of an older approach for opening doors is the assistance system proposed in [9]. The human user of the system has to use green laser pointer to indicate where the door handle is. This point is then detected by the robots vision system and the door opening procedure is started. It is the only work discussed here, which needs a user to actively interact with the robot.

The works proposed in [17], [10] and [14] are not focused on door detection but rather include methods for door handle detection. [17] proposed a deep learning based algorithm while [14] used a template matching algorithm.

The authors of [8] proposed a mobile robot employing a manipulator for autonomous disinfection operations. They use a 3D model of a building and the Iterative Closest Point algorithm [5] for detecting door handles and then refining their pose after positioning.

*This work was not supported by any organization

[1]Ulrich Mitterhuber, Raimund Edlinger and Roman Froschauer are with University of Applied Sciences Upper Austria, 4600 Wels, Austria ulrich.mitterhuber@fh-wels.at, raimund.edlinger@fh-wels.at

[2]Andreas Nüchter is with Informatics VII - Robotics and Telematics, Julius-Maximilians University Würzburg andreas.nuechter@uni-wuerzburg.de

[1]https://github.com/ultralytics/yolov5

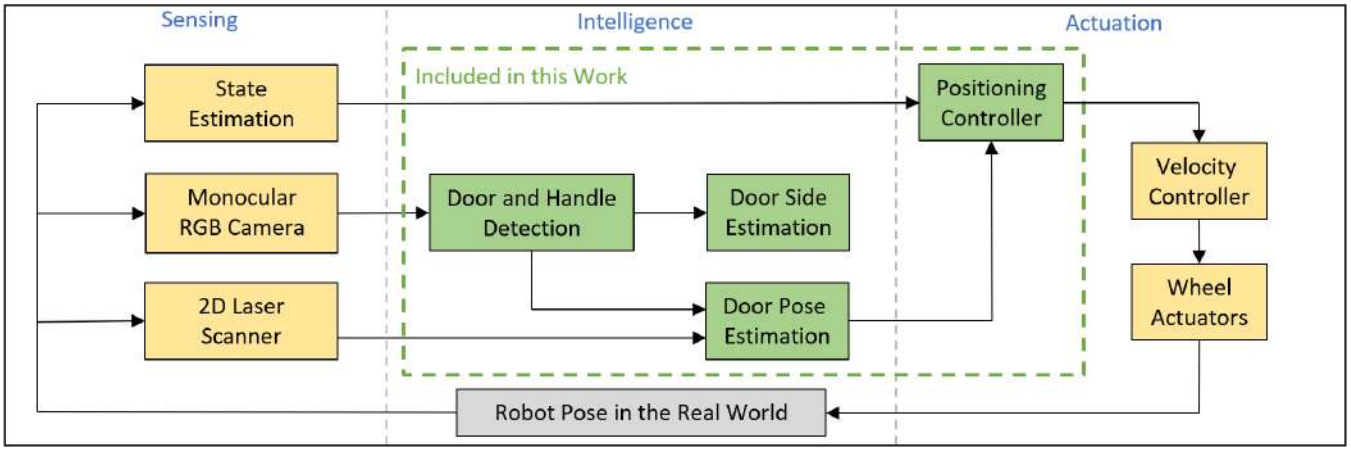


Fig. 1. System overview: The elements in green are developed in this work.

There were also some methods published in which the robot's objective only is to manipulate a door. Thereby the door poses are already coarsely marked in a map or included through prior information such as the robot's pose. [15], [23] and [12] for example each propose different methods of opening or even traversing doors, however do not detect them themselves. They all use the information provided through the map but also employ point cloud processing algorithms to refine the door pose. [3] on the other hand, proposes a semi-autonomous system, for which the robot must be pre-positioned well in front of the door. When the operator starts the opening procedure, the robot uses a 2D laser scanner, and the assumptions where it expects the door, to estimate it more accurately.

The available methods which are detecting doors without prior information can be further divided into two parts. First, methods relying not on deep learning and second, deep learning based ones.

Three examples for non deep learning algorithms are [21], [16] and [4]. The method proposed in [21] uses uncoloured 3D point clouds while the one proposed in [16] uses coloured 3D point clouds to detect doors and estimate their poses in 3D space. [4] uses classical machine vision techniques and the assumption that a door has many vertical and horizontal lines to detect doors within camera images.

The works proposed by [2] and [20] use the same deep learning based object detection algorithms to both detect doors and door handles. They use the obtained bounding box to choose a ROI in a depth map they obtain with a RGB-D camera. A plane is then fitted into the depth map. With that, the door pose becomes available. They estimate the pose of the door handle in a similar fashion as well. The authors of [11] also proposed a convolutional neural network to first estimate a ROI representing the door. They then use the point cloud and visual data from within the ROI to estimate the door plane and find the door handle with non deep learning methods. Lastly, [18] compares various deep learning based algorithms to estimate if a door is open, semi-open or closed. They do however not estimate the doors pose or detect door handles.

The door pose estimation algorithm developed in this work is similar to the one proposed in [2] and [20]. However, it is based on a simpler sensor design which enables it to employ model estimation methods with lower computational requirements.

III. Method

Fig. 1 shows the methods developed in this work embedded in their context of an autonomous mobile robot. The elements in yellow and gray are required for the methods to work and are therefore not part of this study while the ones in green represent the methods developed in this work.

A. Door and Door Handle Detection

The developed door and door handle detection algorithm is based on a state-of-the-art object detection method, more specific *YOLOV5* which is an improvement over the original *YOLO* architecture proposed in [19]. It was chosen due to it's very good implementation, vast community support and the ability to be executed in real-time on an embedded platform.

In order to enable the model to recognise doors and door handles, a data set was created which consists of images and labels of these objects. For its creation, own images were taken and labeled. However, to increase its size and diversity, images of publicly available datasets provided by the authors of [18] and [2] were added as well. They had to be relabeled partly but added a lot of diversity in terms of geographical bias and camera parameters. In total, the dataset consists of 1022 images and labels with a class instance count 1031 for doors and 1103 for door handles. A sample of the dataset is shown in Fig. 2.

The exact dataset split for training and evaluation, the specific hyper-parameter choices, the training results and the model deployment are further explained in section IV.

B. Door Pose Estimation

As already noted in sec. I, this method relies on sensor fusion. More specifically it uses a cooperative sensor fusion

Fig. 2. Object detection dataset sample: Strong red indicates door and light red indicates door handle.

approach whereby the data of the exteroceptive monocular camera and the 2D laser scanner are combined. [1]

The motivation for fusing those sensors is that this enables the method to leverage the recent developments in deep learning machine vision algorithms which provide great performance-efficiency trade-offs such as shown in works as [19] and combine that with the efficiency of 2D laser scan processing. This combination makes the method perfectly suited for autonomous robotics. Furthermore, assumptions which simplify the problem drastically are made. First, a door is always seen as a plane perpendicular to the ground plane and second, the often in robotics made simplification that the world in which the robot's base navigates is represented as a 2D plane, where an object only has the three degrees of freedom $x, y, theta$, is used. A visual representation of the method, which is often referred to in the following algorithm explanations, is shown in Fig. 3.

The input of the algorithm is comprised of door and door handle bounding boxes delivered by the object detection algorithm, the camera's calibration parameters and a laser scan. Since the different input artefacts are not delivered at exactly the same time, they are approximately synchronised with each other. It is important to note that the camera's and laser scanner's optical centers must not be at the same position. For all calculations, the points obtained by the laser

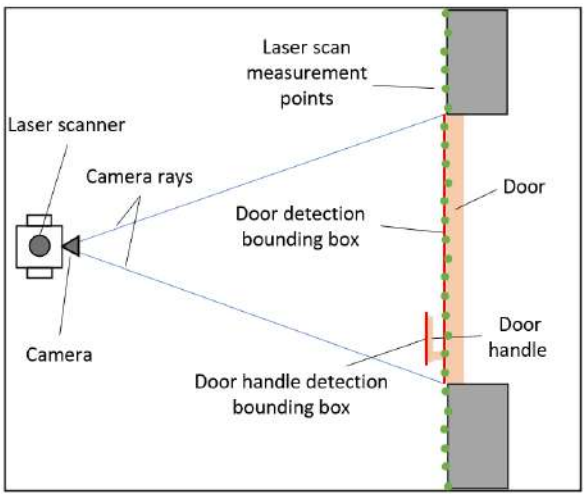


Fig. 3. Door pose estimation method overview.

scanner are transformed into the camera's frame.

The algorithm's first step is to evaluate if a valid door was detected and chose one if multiple were detected. Choosing one of multiple is done by finding the one with the highest corresponding confidence score. The selected door must however obey to a geometric rule which is that the box is not allowed to go all the way out to the left or right image end. This is done to ensure that the door is fully covered in the image and therefore a bounding box representing only the half of a door is not chosen.

With the door bounding box being chosen, in the next step, it's width and the camera's parameters are used to calculate vectors representing camera rays from image pixels. For the left ray's image pixel x coordinate, the x value of the left vertical edge of the bounding box is taken and vice versa for the right ray. The y coordinates for the ray calculation are set to a constant value representing the height at which the camera is mounted on the robot. The calculated rays are then projected onto the 2D x, y plane. Fig. 3 shows the door and door handle bounding boxes in red and the camera rays in blue.

The 2D ray vectors are then used to calculate angles representing the left and right edge of the bounding box. To find the laser scan points which are within the ray angles, first, a x, y point of each laser scan range point is calculated which is then transformed from the scan frame into the camera frame. After that, each point's angle from the camera center is calculated. Every angle that is between the range of the left and right ray angle is assumed to be located on the door. Laser scan range points with invalid values are discarded. Fig. 3 shows the laser scan measurement points in green.

The following step in the algorithm is to fit a 2D line, representing the door pose, into the selected laser scan points. For that, three methods to choose from were implemented. The first one, which is further referred to as naive, connects the leftmost to the rightmost point to construct the line. It ignores all other points within the range and is therefore expected to be very unstable when outliers are occurring at the door edges. The second one uses the commonly known ordinary least squares estimation to fit a line into all available door-points. If there are no outliers at all, this method is expected to deliver optimal results. Due to the fact that it is expected that the algorithm has to cope with many outliers because of already expected bounding box inaccuracies and sensor errors, the third method chosen is the *Random Sample Consensus* (RANSAC) proposed in [7]. It is a probabilistic method commonly known in robotics for its ability to very efficiently find models in noisy data. It is expected that the RANSAC method yields the most accurate results in real-world scenarios. To verify this hypothesis and thus determine the best method for the task at hand, thorough experiments, shown in sec. IV, were conducted.

Once a door line is estimated, the left and right edge points of the door represented in 2D space are obtained. This is done by calculating an intersection each for the left and right camera rays with the estimated door line. Using those

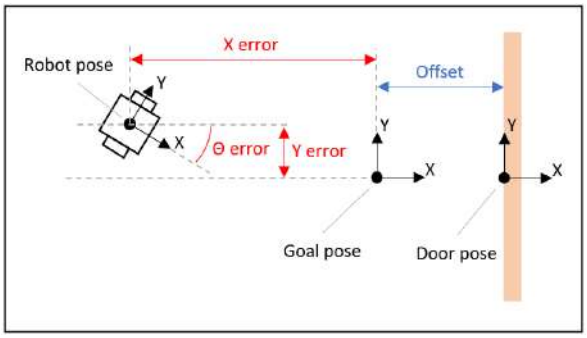


Fig. 4. Robot positioning control visualisation.

two points, the doors center coordinate x, y and its rotation *theta* are calculated. Lastly, to obtain the goal pose to which should be navigated in front of the door, a variable offset is added normal to the door. It's value needs to be chosen while contemplating the workspace size of the manipulator used on the mobile robot. Fig. 4 shows an estimated door and goal pose.

Parallel to the steps above, the estimation of the door handle side is executed. If there is a door handle bounding box found to be within the selected door bounding box it is determined if the center of the handle is in the left or right half of the door. The information of left or right is provided for further manipulation methods. With it, it is clear on which side the door is to be opened and in which direction the handle needs to be turned.

C. Differential Drive Positioning Controller

The controller implemented to move the robot from its starting pose to the goal pose is based on the well known differential drive controller proposed in [22]. Fig. 4 shows the robot in its starting pose, a goal pose and the error which is minimised by the controller.

Another hypothesis made here is that the final positioning accuracy can be improved, if during the robot is moving towards a goal pose, the goal pose estimation goes on and the old estimation is updated continuously, instead of only setting it once at the beginning. This is based on the assumptions that first, the door pose estimation is more reliable than the pose estimation of the robot and second, that if the door is at the beginning of the positioning partly occluded, the view of the door will get better during positioning and with that the goal pose estimation also gets more accurate. It is expected that the door detection is only reliable if the robot is far away from the door. For that, the euclidean distance between the robot's current pose estimation and the goal pose is used to determine when to stop updating. The validity of this hypothesis is evaluated during experimentation shown in section IV.

IV. Experimental Results

To validate the hypothesis made above and verify the functionality of the developed methods, real-world experiments were conducted. The differential drive robot platform used for testing provided an odometry solely based on wheel velocities, a wheel velocity controller, a 360 degree 2D laser scanner with 1 degree angular resolution and a calibrated RGB camera with a horizontal FOV of 60 degrees delivering images with a size of 640x480.

A. Door and Door Handle Detection

For training the object detection model, the dataset was split randomly into 715 train, 153 validation and 154 test images. Training was executed on a PC comprising of an AMD Ryzen 9 3950X CPU, 128 GB DDR 4 RAM and two NVIDIA RTX 2080 Ti graphics cards running on Ubuntu 18.04. The batch size was set to 90 images with a size of 640x640 each. For data augmentation random horizontal flipping with a probability of 0.5, linear translation with a factor of 0.1 and scaling with a factor of 0.5 were used. *YOLOV5* provides a few different model architecture types from which the lightweight *YOLOV5S* was chosen. Training was done from scratch and stopped after 154 epochs because no improvements were made over the previous 30 epochs. For model evaluation the mean average precision metric (mAP), introduced for the PASCAL VOC challenge in [6], was employed. When evaluating the trained model on the test dataset, it achieved a mAP 0.5 of 90.1% and mAP 0.5:0.95 of 59.9%. Therefore it is concluded, that the model is able to reliably detect previously unseen doors and door handles. Some qualitative inference results are shown in Fig. 5. Both images were acquired in the real world and are not included in the training dataset. It can be observed that all doors and door handles present in the images are detected.

For inference, the model was deployed within a ROS node implemented in plain Python and PyTorch on the *NVIDIA Jetson AGX Xavier*. It must be noted that no optimization frameworks such as *NVIDIA TensorRT* were employed. Nonetheless, the model is able to perform inference with an input image size of 640x640 with a speed of above 30 frames per second.

B. Door Pose Estimation

To validate the hypothesis that the RANSAC based door line estimation method is the most accurate one for this task, a measurement series of 20 samples comparing each method to a ground truth was taken. For each sample, the robot was placed in a new scenario in which the door or

Fig. 5. Object detection: Qualitative inference examples.

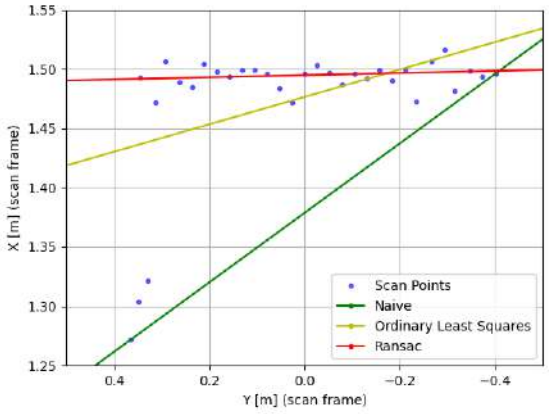


Fig. 6. Door line estimation methods comparison with an obstacle at the left side.

an obstacle placement was changed. During experimentation with the RANSAC method, the max trials parameter was set to 1000 and the residual threshold to 0.01. Figure 6 shows one sample measurement with strong outliers at the left side resulting from an obstacle. It can be observed that the RANSAC method is the most and the naive method the least accurate. Each estimated door line is subsequently used to calculate the door's estimated pose. The estimated and ground truth door poses were then used to calculate an absolute error for every dimension, e.g. $x, y, theta$, of each estimated pose, for each sample. The three pose dimension errors are then summed up resulting in a single error metric for a given sample. With that, for each method, 20 samples of total absolute pose errors representing their accuracy were obtained. Tab. I shows the mean and standard deviations calculated for each methods error data. To determine if the collected data supports the hypothesis that the method based on RANSAC is the most accurate, i.e. yields the lowest error, two Z-Tests with known variances such as described in [13] were executed. The significance threshold was set to 5% . In both tests, the null-hypothesis was that both methods are equally good while the alternative-hypothesis was that the error of the naive and least squares method is higher than the RANSAC method's error. In both tests, the null-hypothesis was reject and thus the data significantly supports the hypothesis that the RANSAC method is the most accurate one.

TABLE I

COMPARISON OF THE IMPLEMENTED LINE ESTIMATION ALGORITHMS WITH REAL WORLD DOORS.

Method	Mean [1]	Standard Deviation [1]
Naive	0.3212	0.1774
Least Squares	0.1167	0.0667
Ransac	0.0311	0.0062

TABLE II

COMPARISON OF INITIAL AND CONTINUOUS GOAL POSE UPDATING IN THE CONTROLLER.

Goal Updating	Mean [1]	Standard Deviation [1]
Initial	0.2492	0.0948
Continuous	0.0650	0.0361

C. Differential Drive Positioning Controller

To validate the hypothesis that continuously updating the goal pose during moving the robot to the goal yields a more accurate final placement than updating only once in the beginning, an experiment was conducted. The robot was placed at 18 different start poses from which it was already able to estimate the door pose, i.e. the goal-pose. From each start pose, positioning was once done with and without continuous updating. The euclidean distance parameter for stopping continuous updating was set to 0.2m. For estimating the goal pose, the RANSAC method with the maximum trials parameter set to 1000 and the residual threshold set to 0.01, was used. Lastly, derived from the employed robot's manipulator workspace size, the goal offset was set to 0.5m.

After each positioning procedure, the error between the reached and the ground truth pose was measured. To represent the positioning error within one metric, again the absolute $x, y, theta$ errors were summed up to form a total absolute pose error. Tab. II shows the mean and standard deviation of both method's errors. To determine if the collected data supports the hypothesis that continuous goal pose updating yields a more accurate final pose, i.e. a lower positioning error, than only estimating the goal once, a Z-test, such as used before, comparing both method's measurements, was executed. The significance threshold was again set to 5%. The null-hypothesis was that both methods perform equally good while the alternative-hypothesis was that non continuous updating yields a higher error. When executing the test, the null-hypothesis was clearly rejected and with that the data significantly supports the hypothesis that continuous goal pose updating improves the positioning accuracy.

V. SUMMARY AND OUTLOOK

This work aimed at solving the problem of detecting doors, estimating their pose and controlling a differential drive robot to a specified goal in front of the door. The objective was successfully fulfilled by developing a novel door pose estimation algorithm based on deep learning object detection and 2D laser scan processing. It was also shown that the implemented controller achieves a higher positioning accuracy when the goal pose is estimated and updated continuously while the robot is moving towards the goal pose. The fields of application of the proposed method do not only lie at autonomous door opening, but also in areas such as environment exploration where the method could be used to detect various objects autonomously, estimate their pose and integrate them into an environment map. Future work will include the creation of a more robust and diversified

dataset for the training of the neural network and deploying the algorithm on other types of mobile robots.

REFERENCES

[1] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39 830–39 846, 2020.

[2] M. Arduengo, C. Torras, and L. Sentis, "Robust and adaptive door operation with a mobile robot," *Intelligent Service Robotics*, vol. 14, no. 3, pp. 409–425, 2021.

[3] B. Axelrod and W. H. Huang, "Autonomous door opening and traversal," in *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*. IEEE, 2015, pp. 1–6.

[4] N. Banerjee, X. Long, R. Du, F. Polido, S. Feng, C. G. Atkeson, M. Gennert, and T. Padir, "Human-supervised control of the atlas humanoid robot for traversing doors," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 722–729.

[5] A. Censi, "An icp variant using a point-to-line metric," in *2008 IEEE International Conference on Robotics and Automation*. Ieee, 2008, pp. 19–25.

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

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

[8] A. Giusti, V. Magnago, D. Siegele, M. Terzer, C. Follini, S. Garbin, C. Marcher, D. Steiner, A. Schweigkofler, and M. Riedl, "Balto: A bim-integrated mobile robot manipulator for precise and autonomous disinfection in buildings against covid-19," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 1730–1737.

[9] A. Jain and C. C. Kemp, "Behaviors for robust door opening and doorway traversal with a force-sensing mobile manipulator." Georgia Institute of Technology, 2008.

[10] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to open new doors," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 2751–2757.

[11] A. Llopart, O. Ravn, and N. A. Andersen, "Door and cabinet recognition using convolutional neural nets and real-time method fusion for handle detection and grasping," in *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*. IEEE, 2017, pp. 144–149.

[12] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, *et al.*, "Autonomous door opening and plugging in with a personal robot," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 729–736.

[13] D. C. Montgomery and G. C. Runger, *Applied statistics and probability for engineers*. John Wiley & Sons, 2010.

[14] C. Ott, B. Bäuml, C. Borst, and G. Hirzinger, "Employing cartesian impedance control for the opening of a door: A case study in mobile manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on mobile manipulators: Basic techniques, new trends & applications*, 2005.

[15] S. A. Prieto, A. Adán, A. S. Vázquez, and B. Quintana, "Passing through open/closed doors: A solution for 3d scanning robots," *Sensors*, vol. 19, no. 21, p. 4740, 2019.

[16] B. Quintana, S. A. Prieto, A. Adán, and F. Bosché, "Door detection in 3d colored laser scans for autonomous indoor navigation," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–8.

[17] B. Ramalingam, J. Yin, M. Rajesh Elara, Y. K. Tamilselvam, M. Mohan Rayguru, M. Muthugala, and B. Félix Gómez, "A human support robot for the cleaning and maintenance of door handles using a deep-learning framework," *Sensors*, vol. 20, no. 12, p. 3543, 2020.

[18] J. G. Ramôa, V. Lopes, L. A. Alexandre, and S. Mogo, "Real-time 2d–3d door detection and state classification on a low-power device," *SN Applied Sciences*, vol. 3, no. 5, pp. 1–13, 2021.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[20] S. Roelofs, B. Landry, M. K. Jalil, A. Martin, S. Koppaka, S. K. Tang, and M. Pavone, "Vision-based autonomous disinfection of high-touch surfaces in indoor environments," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2021, pp. 263–270.

[21] R. B. Rusu, W. Meeussen, S. Chitta, and M. Beetz, "Laser-based perception for door and handle identification," in *2009 International Conference on Advanced Robotics*. IEEE, 2009, pp. 1–8.

[22] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[23] M. Stuede, K. Nuelle, S. Tappe, and T. Ortmaier, "Door opening and traversal with an industrial cartesian impedance controlled mobile robot," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 966–972.

Estimation of Robot-Specific Parameters for Robot Motion Models

Lea Zinkanell[1], Matthias Eder[1] and Gerald Steinbauer-Wagner[1]

***Abstract*— Localization is a well-studied problem in the field of mobile robotics and is a challenging task as observations like motion estimates or sensor readings are subject to errors. To accurately estimate a robot's pose, such errors need to be considered and thus modeled. In this work, we focus on estimating a robot's pose after motion commands were executed on it. Therefore, an approach to automatically estimate the parameters of the classical Velocity Motion Model using least squares optimization is proposed. It is assumed that the commanded velocities differ from the actual velocities, as noise is distorting the robot's motion. The proposed approach was tested on artificially generated measurements, samples acquired using a simulated robot, and data acquired by conducting experiments with a real robot. The results show that the approach performs better for the measurements acquired with the real robot than with the samples generated in a simulated environment.**

I. INTRODUCTION

Localization is an important problem in the field of mobile robots. It denotes the estimation of the robot's pose with respect to the environment using sensor measurements. Accurate knowledge of the robot's location is needed to control the robot and perform complex tasks, such as navigation [12]. Because all aspects of mobile robot systems such as sensing and acting are subject to noise, uncertainty, and non-determinism localization becomes a state estimation problem where the most likely state of the system is estimated based on uncertain observations. Robot localization can be stated as the estimation of the following conditional probability $p(x_t|z_{1,..t}, u_{1,..t})$ where the unknown state x_t at a time t is the 2D (x, y; θ (yaw)) or 3D (x, y, z; α, β, γ (roll, pitch, yaw)) pose and the observations are the collected sensor measurements $z_{1,..,t}$ and control commands to the robot $u_{1,..,t}$ until time t. Standard approaches for online state estimation in Robotics are different variants of Bayes Filter such as Extended Kalman Filter [11] or Particle Filter [6]. All these methods employ a two-stage process where first a prediction of the last estimated state towards the new time step ($\hat{x_t}$) using the control inputs is performed which is then corrected using the actual measurements. For the correction step a sensor model is used that estimates how likely a measurement is in a given state $p(z_t|\hat{x_t})$. A good prediction or prior for the correction is important for the performance of the state estimation. The prediction in the context of localization is usually performed using a probabilistic motion model that represents the conditional probability $p(x_t|x_{t-1}, u_t)$ for a transition between the poses x_{t-1} and x_t. Various motion models were proposed in the literature that are based on inputs from sensors such as odometry or IMU [1], online estimations of robot motions [10] or direct control inputs [8]. In this paper, we focus on the so-called *Velocity Motion Model* where the motion of the robot is controlled by a translational (v) and a rotational (ω) velocity (differential drive) [12]. The conditional probability of the pose transition is represented by zero-mean normal distributed noise that is added to the control inputs. The parameter of this additive noise depends on the environment (e.g. floor) and the robot's locomotion system. Thus, this parameter needs to be determined or estimated beforehand to obtain a good representation of the uncertainty in the robot's motion. Often these parameters are guessed or optimized manually.

In this paper, we propose an approach to automatically estimate the parameters of the classical *Velocity Motion Model* using least squares optimization. In the first stage, short random motion samples of the robot in the actual environment are executed and the robot displacement is recorded. In a second step, the parameters of the motion model are optimized to represent the probability distribution of the commanded motion best.

The remainder of the paper is structured as follows. First, related work on the identification of uncertainties in robot motion is discussed. In Section III the method for the probabilistic estimation of robot-specific motion parameters is presented. An experimental evaluation of the proposed approach using simulated and real-world experiments is presented in the next section. In Section V we conclude the paper and discuss some future work.

II. RELATED RESEARCH

In this section we discuss relevant research which is mainly related to estimating the robot's motion (often referred to as odometry) and the reduction of systematic errors and the estimation of non-systematic errors.

In [13], the author assessed an autonomous mobile vehicle equipped with a motion controller. Since the motion controller received localization information from the optical wheel encoders, an analysis of the impact of errors from such measurements by deriving the location estimator and its covariance matrix was conducted.

In [2], Borenstein and Feng introduced practical methods to measure and reduce common errors in differential drive systems. Such errors include uncertainty about the effective wheelbase and unequal wheel diameters and belong to the class of systematic errors. To isolate and measure these systematic errors, the authors discussed two test sequences:

[1]Lea Zinkanell, Matthias Eder and Gerald Steinbauer-Wagner are with the Institute of Software Technology, Graz University of Technology, Graz, Austria. {lea.zinkanell, matthias.eder, steinbauer}@ist.tugraz.at

the Unidirectional Square Path and the Bidirectional Square Path, which is known as the UMBmark test. While the UMBmark test is only able to identify systematic errors, the approach discussed in this work is not constrained to a systematic error type and represents thus a more generic approach.

Chong and Kleeman presented an accurate odometry system, which includes modeling systematic and non-systematic errors [5]. To calibrate the systematic errors, the authors applied the UMBmark test [3]. To model the non-systematic errors, the authors developed a novel error model, which also used a normal distribution and expressed the errors using a covariance matrix. In contrast to other models, complex paths were divided into smaller segments and approximated by either a circular arc motion, a straight line, or a rotation around the center of the robot.

Martinelli presented an error model for an odometry system for a synchro-drive and a method to evaluate it [9]. Four parameters were introduced, representing the systematic error components and the non-systematic error components. The non-systematic errors were modeled using a normal distribution. The resulting covariance matrix depended on the path the robot followed. The error parameters were estimated by measuring the change in rotation and the position between the initial and the final configuration of the robot for a given path. The advantage of this method is that the actual path of the robot did not need to be known.

Roy and Thrun proposed a statistical method for calibrating the odometry of mobile robots in [10]. Instead of basing their method on acquired measurements, the authors proposed an algorithm that uses the robot's sensors to automatically calibrate the robot during operation.

In [4] the authors proposed an approach for the simultaneous calibration of odometry parameters (e.g. wheel diameter and base) and sensor placements (extrinsic parameters). The approach uses measurements from the robot's odometry and a Lidar sensor and formulates the calibration as a joint optimization problem. To collect data for the optimization step meaningful motions need to be executed. Moreover, the paper presents an analysis of the observability of parameters and the choice of useful trajectories. A similar approach in estimating the robot motion model is followed in this work.

III. METHODOLOGY

In this section, we present in detail the proposed probabilistic approach to estimate the robot-specific parameters of the *Velocity Motion Model*. We are interested in this automated parameter estimation as this motion model is standard in several realizations of localization and mapping approaches such as *Adaptive Monte Carlo Localization* [12] and its implementation within the ROS Navigation Stack [7]. In the next subsections, we will introduce the motion model and its parameter more formally and describe the estimation approach in more detail.

A. Velocity Motion Model

This motion model represents the state transition probability $p(x_t|u_t, x_{t-1})$ for a differential drive robot on a 2D plane controlled by providing a translational (v) and a rotational (ω) velocity. In the proposed model the velocities are assumed to be constant for a short time window. Thus, each movement of the robot forms an arc. The fraction of the velocities describes the radius r of the arc, the mobile robot is traveling on.

$$r = \left|\frac{v}{\omega}\right| \tag{1}$$

In order to apply the motion model, the following is assumed:

- the initial position and orientation of the robot $(x, y, \theta)^T$,
- the commanded velocities v and ω, and
- the duration Δt of the movement are known

The applied velocities are assumed to be constant during the short period of time Δt to allow an accurate approximation for true motion of the robot. Based on these assumptions, the successor pose $(x', y', \theta')^T$ can be computed:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{\omega}\sin\theta + \frac{v}{\omega}\sin(\theta + \omega\Delta t) \\ \frac{v}{\omega}\cos\theta - \frac{v}{\omega}\cos(\theta + \omega\Delta t) \\ \omega\Delta t \end{pmatrix} \tag{2}$$

Equation 2 represents the motion of the robot with constant velocities on an ideal arc.

Due to non-systematic motion errors, such an ideal motion can be hardly achieved in reality. Such errors disturb the robot's movement and negatively affect the estimation of the successor pose. These motion errors need to be considered in the probabilistic motion model. This is achieved by assuming that the commanded velocities differ from the actual velocities due to motion errors. The errors which distort the commanded velocities are modeled as additive random error variables ε_b, which have a zero mean and are normally distributed with the standard deviation b. Using these assumptions, the actual uncertain velocities $\hat{v}$ and $\hat{\omega}$ can be modeled as:

$$\begin{pmatrix} \hat{v} \\ \hat{\omega} \end{pmatrix} = \begin{pmatrix} v \\ \omega \end{pmatrix} + \begin{pmatrix} \varepsilon_{\sigma_v} \\ \varepsilon_{\sigma_\omega} \end{pmatrix} \tag{3}$$

The standard deviations of the translational and rotational motion error are modelled separately. They represent the magnitude of the uncertainty in the successor pose. According to the *Velocity Motion Model* the standard deviation for the translational and rotational velocity error can be computed as:

$$\sigma_v = \alpha_1|v| + \alpha_2|\omega| \tag{4}$$

$$\sigma_\omega = \alpha_3|v| + \alpha_4|\omega| \tag{5}$$

σ_v represents the standard deviation for the translational error, while σ_ω represents the standard deviation for the rotational error. The standard deviations of the motion errors are proportional to the velocities, consequently, the uncertainty increases linearly with increasing velocities. This implies that for higher velocities, the stronger successor positions will deviate.

The different α_i represent robot-specific error parameters. α_1 determines the influence the translational velocity v has on

the standard deviation of the translational motion error, while α_2 determines the influence the rotational velocity ω has on the standard deviation of the translational error. Likewise, α_3 determines the influence v has on the standard deviation of the rotational error, while α_4 determines the influence ω has on the standard deviation of the rotational error. The error parameters α_i model the robot's accuracy w.r.t. commanded motion, which implies that the larger the parameters are, the less accurate the robot's motion is. Usually, the error parameters are assumed to stay constant, which makes it possible to estimate them beforehand. The parameters are assumed to be zero or non-negative.

By replacing the commanded velocities $\langle v, \omega \rangle$ with the noisy velocities $\langle \hat{v}, \hat{\omega} \rangle$ in Equation 3, the noisy successor pose can be computed as follows:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{\hat{v}}{\hat{\omega}} sin\theta + \frac{\hat{v}}{\hat{\omega}} sin(\theta + \hat{\omega}\Delta t) \\ \frac{\hat{v}}{\hat{\omega}} cos\theta - \frac{\hat{v}}{\hat{\omega}} cos(\theta + \hat{\omega}\Delta t) \\ \hat{\omega}\Delta t \end{pmatrix} \quad (6)$$

Equation 6 can be used to predict the robot's pose within a state estimation approach.

B. Parameter Estimation

To be able to apply the motion model for a particular robot system and environment, we need to identify the robot-specific error parameters α_i. An accurate estimation of the error parameters is vital to provide precise predictions of the robot's successor pose. It has to be noted, that in the full original motion model two more error parameters α_5 and α_6 are used to represent the error of the final rotation, which were not considered in this work.

Algorithm 1: EstimateVMM

Input: Sampling Parameter: v_{min}, v_{max}, ω_{min},ω_{max}, Δt, N_v, N_ω, N_{move}
Output: Model Parameter: $\alpha_1, ..., \alpha_4$

1 $\Delta v = (v_{max} - v_{min})/N_v$.
2 $\Delta\omega = (\omega_{max} - \omega_{min})/N_\omega$.
3 $\langle x, y, \theta \rangle = \langle 0, 0, 0 \rangle$
4 **for** $i = 1, ..., N_v$ **do**
5 **for** $j = 1, ..., N_\omega$ **do**
6 $u = \langle v_{min} + i \times \Delta v, \omega_{min} + j \times \Delta\omega \rangle$
7 $\sigma_{u,v} = 0$
8 $\sigma_{u,\omega} = 0$
9 **for** $k = 1, ..., N_{move}$ **do**
10 $\langle x', y', \theta' \rangle = \text{executeMotion}(\langle x, y, \theta \rangle, u, \Delta t)$
11 $\langle \hat{v}, \hat{\omega} \rangle = \text{EstimateVel}(\langle x, y, \theta \rangle, \langle x', y', \theta' \rangle)$
12 $\langle x, y, \theta \rangle = \langle x', y', \theta' \rangle$
13 $\sigma_{u,v} += (\hat{v} - v)^2$
14 $\sigma_{u,\omega} += (\hat{\omega} - \omega)^2$
15 $\sigma_{u,v} = \sqrt{\frac{1}{N_{move}-1} \sigma_{u,v}}$
16 $\sigma_{u,\omega} = \sqrt{\frac{1}{N_{move}-1} \sigma_{u,\omega}}$
17 $\langle \alpha_1, \alpha_2 \rangle = \text{solveOLS}(\langle \sigma_{u,v} \rangle)$
18 $\langle \alpha_3, \alpha_4 \rangle = \text{solveOLS}(\langle \sigma_{u,\omega} \rangle)$

For estimating the parameters $\alpha_1, .., \alpha_4$ we use the Algorithm 1. The algorithm comprises three major parts: (1) motion sampling, (2) determination of velocity-depending uncertainty, and (3) parameter estimation.

In a first step for each sampled velocity combination $\langle v, \omega \rangle$ N_{move} motion samples are generated (line 1 to 6). In the current implementation the space of possible velocity combination is uniform sampled using the velocity limits of the robot platform (v_{min}, v_{max}, ω_{min},ω_{max}) and a given numbers (N_v, N_ω) for the subdivision of the two velocity ranges.

The sampled velocity combinations are executed on the robot system N_{move} times for a time window Δt and the initial and end robot pose of that motion are tracked (lines 9 to 10).

By applying Equations 7 to 13 (inverse motion model) [12] , one can retro-guess the actual executed velocities during the motion sample and thus the additive motion errors (line 11). The purpose of the original formalization is to compute the probability of a potential successor pose, based on velocity information and the starting position. For the proposed approach we only use estimation of the actual uncertain velocity pair $\langle \hat{v}, \hat{\omega} \rangle$. Please note that this set of estimations represents the probability distribution of the executed velocities for each velocity pair.

$$\mu = \frac{1}{2} \frac{(x - x')\cos\theta + (y - y')\sin\theta}{(y - y')\cos\theta - (x - x')\sin\theta} \quad (7)$$

$$x^* = \frac{x + x'}{2} + \mu(y - y') \quad (8)$$

$$y^* = \frac{y + y'}{2} + \mu(x' - x) \quad (9)$$

$$r^* = \sqrt{(x - x^*)^2 + (y - y^*)^2} \quad (10)$$

$$\Delta\theta = \text{atan2}(y' - y^*, x' - x^*) - \text{atan2}(y - y^*, x - x^*) \quad (11)$$

$$\hat{v} = \frac{\Delta\theta}{\Delta t} r^* \quad (12)$$

$$\hat{\omega} = \frac{\Delta\theta}{\Delta t} \quad (13)$$

Having acquired these set of measurements for each combination of a translational and rotational velocity (u), in a second step maximum likelihood estimation is applied to estimate the standard deviation of the normally distributed motion errors ($\sigma_{u,v}$, $\sigma_{u,\omega}$), which are offsets to the commanded velocities. The standard estimator for the standard deviation of a normal distribution is applied in an iterative way (lines 13 to 14). Please note that we assume an error that has a zero mean.

Knowing the estimated standard deviations ($\langle \sigma_{u,v}, \sigma_{u,\omega} \rangle$) for velocity pairs ($u = \langle v, \omega \rangle$), the error parameters α_i can be estimated in a third step (line 17 to 18). According to Equation 4 and 5 two linear equations for each sampled velocity combination exist. Each equation comprises two unknown variables, the error parameters α_1 and α_2 for the standard deviation of the translational error, and α_3 and α_4 for the standard deviation of the rotational error. This leads to two ordinary least square optimization problems for the

parameters α_i with an equation of the following form for each sampled velocity combination u:

$$(|v|, |\omega|)(\alpha_1, \alpha_2)^T = \sigma_{u,v} \tag{14}$$
$$(|v|, |\omega|)(\alpha_3, \alpha_4)^T = \sigma_{u,\omega} \tag{15}$$

The solving of the two optimization problems is done by applying the ordinary least squares method (OLS) for the proposed model. This method provides an optimal estimation for the unknown variables based on the velocity combinations and estimated standard deviations. The performance of the ordinary least squares method depends on the quality of the measured samples. It is also to be noted, that OLS is sensitive towards outliers, which may distort the estimations. Due to this sensitivity, it may be necessary to pre-process the data before estimating the parameters.

IV. EVALUATION & RESULTS

To evaluate the proposed approach for estimating the robot-specific parameter of the *Velocity Motion Model* three experiments were conducted. In the first experiment artificially generated motion data with known parameters were used to validate the basic algorithm. In a second experiment motion data from a simulation of the Husky robot were used to evaluate the algorithm on more realistic data. Although the motion models in simulations are usually simplified, collecting data including ground truth in larger quantities is easier in simulation. Finally, the approach was evaluated using data from a real Husky robot. The results of the three conducted experiments will be compared and discussed in detail in the following sections.

A. Experimental setup

For all experiments, the operating system Ubuntu 18.04 was used. Scripting was done in Python utilizing Python libraries such as Matplotlib, Numpy, Math, Scipy, and Rospy. The simulation of the robot was performed with the Gazebo simulator and the standard Husky simulation package. The simulated, as well as the real robot, were controlled using ROS Melodic, which was also used to obtain information about the robot's current state. Simulation experiments were conducted in an empty world environment using the standard friction coefficients. Motion samples are obtained using Gazebo's internal ground truth. The real robot is equipped with two 2D laser scanners which were used to measure the robot's displacement using a standard scan matching library. Please note that the estimation of the robot's displacement only needs to be locally consistent. Thus, no global localization approach is needed.

Initial tests with the real robot were conducted in a small indoor space with wooden flooring. Further experiments were conducted outdoors on asphalt.

See Figure 1 for the simulated and real robot setup.

Fig. 1: Simulated (left) and real robot (right) setup used in the evaluation.

B. Method Validation on Generated Data

In this experiment the sampling algorithm for the *Velocity Motion Model* as presented in [12] was used to generate artificial motion samples. Please note that the distribution of these samples represents the uncertainty encoded in the probabilistic motion model. For the sake of validating the approach true error parameters α_i were set to:

$$\alpha_1 = 0.4753, \alpha_2 = 0.7482, \alpha_3 = 0.1001, \alpha_4 = 0.3720$$

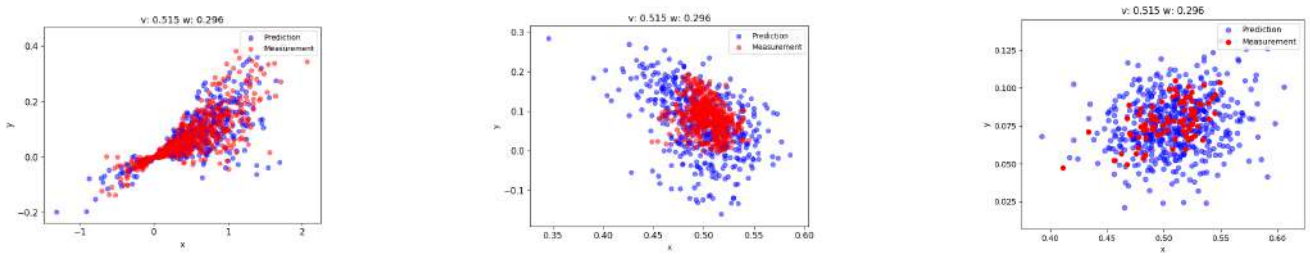
Since the true error parameters α_i were known, the estimations were directly evaluated by computing the difference between the estimated and the true error parameters. Following that approach, the proposed method could be validated, presuming the robot behaved perfectly according to the velocity motion model. It was observed, that the accuracy of the estimations strongly depended on the number of samples.

Table I shows the estimated error parameters obtained from the artificial data. In the experiment 100 velocity combinations with 100 sample movements each were obtained. The estimations of the error parameters α_i are quite accurate, as the difference between the true error parameters and the estimated error parameters is around 1 to 3 %. Figure 2a depicts the distribution of the generated motion samples (robot's displacement) for a single velocity combination using the ground truth motion parameter as well as using the estimated parameter.

	true	estimated	Δ	$\Delta/\%$
α_1	0.4753	0.4792	0.0039	0.82
α_2	0.7482	0.7557	0.0074	0.99
α_3	0.1001	0.1039	0.0038	3.80
α_4	0.3720	0.3633	0.0087	2.34

TABLE I: Actual and estimated error parameters derived from artificial data. 100 velocity combinations and 100 samples each were used for the estimation.

Table II shows the performance of the approach using only 10 velocity combinations and 10 motion samples each. One can clearly so the degration in the performance of the parameter evaluation.



(a) (b) (c)

Fig. 2: Measured movement samples for a given velocity combination ($v = 0.515\ m/s, \omega = 0.296\ rad/s$) versus a prediction using the estimated motion parameters with the Velocity Motion Model (robot's end position): a) using artificial data generation, b) within the simulation environment, and c) using the real robot.

	true	estimated	Δ	$\Delta/\%$
α_1	0.4753	0.5296	0.00543	11.14
α_2	0.7482	0.6525	0.0957	12.79
α_3	0.1001	0.1121	0.0120	11.99
α_4	0.3720	0.3930	0.0209	5.61

TABLE II: Actual and estimated error parameters derived from artificial data. 10 velocity combinations and 10 samples each were used for the estimation.

As a second metric, the normalized Kullback–Leibler divergence D_{KL} between the distribution of the measured and the predicted end positions is used. This metric estimates the similarity of two distributions and fits well for evaluating a probabilistic motion model. In the calculation binning and a 2-dimensional histogram is used. The divergence computed for 100 velocity combinations resulted in a distance of 0.006. This indicates that the predictions of the successor positions are quite accurate. Consequently, it can be said, that the proposed approach works well, assuming that the robot behaves according to the *Velocity Motion Model*.

C. Error Estimation in Simulation

Experiments in the Gazebo simulator were conducted using the standard simulation model of a Husky robot. To obtain the motion samples, the simulated robot is commanded with constant velocities for a small time window. To eliminate high uncertainties of the simulation if the robot is commanded from a full stop, a controller is implemented that first accelerates the robot to the desired velocities and starts recording the initial and end pose after that. The data obtained in the simulation this way was used for the parameter estimation.

Evaluating data acquired in the Gazebo simulator was more difficult, as the true error parameters of the simulated robot model were not known. For this reason, it was important to compare predictions and measurements visually, as well as objectively by computing the KL-divergence for the data set.

It has to be noted that for certain velocity combinations, which form a small turning radius, the measurements deviate much stronger in their successor positions, than for velocity combinations, which form larger radii. The Husky robot is 0.9 m long, 0.67 m wide, and has four non-steerable wheels, thus tight turns result in larger motion errors due to issues like the clattering of the wheels over the floor. Due to that issue, a minimum turning radius constraint of 1 m was enforced.

Using the data recorded in the simulation the error parameters α_i were estimated with the following values:

$$\alpha_1 = 0.007, \alpha_2 = 0.058, \alpha_3 = 0.0, \alpha_4 = 0.655$$

Since α_4 was estimated to be relatively large compared to the other error parameters, it appears that the rotational velocity was subject to a larger error. Which is reasonable for a skid-steered robot.

For each velocity combination, two 2D histograms were computed, one for the predicted data set and one for the measured data set. An example of a velocity pair can be seen in Figure 2b. The KL-divergence was calculated, by applying a bin-to-bin comparison. The results for each velocity combination were summed up, resulting in the KL-divergence for the complete model. The average KL-divergence was calculated to be 0.61 when considering 100 velocity combinations with 100 movement samples each. Compared to the KL-divergence for the generated data, this is a rather high value. These results show that the proposed approach together with the rather simple motion model does not work as accurately as expected for the simulation data.

D. Error Estimation on a real robot

Following the evaluation of simulated data, experiments with a real Husky robot were conducted. These experiments were the most interesting because noise originating from the surface properties and other environmental properties additionally influenced the robot's motion. Since over-determined equation systems produce more accurate results when considering more equations, 100 velocity combinations and 100 samples per velocity combination were measured for the generated and the simulated data. As the real-world experiments with the robot were more time-consuming, only 30 velocity combinations and 50 samples per velocity combination were collected at the end. The velocities for the measurements

were chosen according to a uniform distribution. The translational velocity v took values between 0.2 m/s and 1.0 m/s, while the rotational velocity took values between -0.5 rad/s and 0.5 rad/s.

For the data collected with the real robot, the error parameters α_i were estimated with the following values:

$$\alpha_1 = 0.044, \alpha_2 = 0.032, \alpha_3 = 0.051, \alpha_4 = 0.146$$

Based on these values, it can be concluded that the translational velocity has a bigger influence on the translational motion than the rotational velocity. Similarly, it appears that the rotational velocity has a bigger influence on the rotational motion than on the translational error. The distribution of the samples for an exemplary velocity pair can be seen in Figure 2c.

The normalized KL-divergence was estimated at 0.42. As this value is lower than in the simulated case, it indicates that the proposed approach works more accurately for the real Husky robot than for the simulated one. It can be concluded, that the error parameters α_i were accurately estimated for the data recorded with the real Husky robot. The better match of the real and predicted position are also visible in Figure 2c.

V. CONCLUSION

Motion prediction models are an important part of several localization and mapping approaches. Estimating the robot-specific parameters of such models is cumbersome. In this paper we proposed an automated approach to estimate these parameters which samples motion executions for various combinations of translational and rotational velocities and uses optimization-based parameter estimation to obtain the model parameter.

The approach has been evaluated with artificial data, data from a robot simulation, and data obtained with a real robot. When using generated measurements to validate the approach, it was observed, that the accuracy of the estimated α_i parameters heavily depended on the number of motion samples. The more measurements per velocity combination were considered, the more accurate the estimations of the standard deviations of the motion errors were. In general it cloud be seen, that if a robot behaves according to the probabilistic motion model, the error parameters α_i can be accurately estimated.

By conducting experiments in the Gazebo simulator, it was found, that the predictions appeared to be quite inaccurate, compared to the predictions for the artificially generated samples. A large rotational error was observed, causing stronger deviations in the predicted successor positions, than in the measured successor positions. In general it was evident, that the simulated Husky robot did not behave as expected. From an investigation of the obtained data it can be assumed that the measured successor positions were simply normally distributed. This lead to the conclusion, that either the chosen motion model was not suitable for the simulated Husky robot or that the error model used in the Gazebo simulator need to be adapted to reflect real robot motions. In general estimating the error parameters for simulated data works, but that motion errors in the predicted samples are largely overestimated, which implies that a different motion model may be better suited.

By conducting experiments on a real robot it was observed that the sample measurements were in general more accurate than the measurements of the simulation. During these experiments, the robot was also subject to environmental influences, such as the ground texture. As the experiments with the real robot were time-consuming, less measurements were recorded, which may have affected the parameter estimations.

By comparing the results of the simulation and the real robot, it was concluded, that the proposed approach works more accurately when used on a real robot. All in all it can be said, that the proposed approach works sufficiently accurate for real robots and can be applied for relative position estimation and prediction.

For future work the influence of different environment factors such as friction coefficients and surfaces need to be considered. Moreover, an adaptation of the approach to different motion models and different locomotion systems would be interesting.

REFERENCES

[1] M. B. Alatise and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter," *Sensors*, vol. 17, no. 10, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/10/2164

[2] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 869–880, 1996.

[3] ——, "Umbmark: A method for measuring, comparing, and correcting dead-reckoning errors in mobile robots," University of Michigan, Tech. Rep., 1994.

[4] A. Censi, A. Franchi, L. Marchionni, and G. Oriolo, "Simultaneous calibration of odometry and sensor parameters for mobile robots," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 475–492, 2013.

[5] K. S. Chong and L. Kleeman, "Accurate odometry and error modelling for a mobile robot," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, 1997, pp. 2783–2788 vol.4.

[6] L. Lang, W. shiang Chen, B. R. Bakshi, P. K. Goel, and S. Ungarala, "Bayesian estimation via sequential monte carlo sampling—constrained dynamic systems," *Automatica*, vol. 43, no. 9, pp. 1615–1622, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109807001653

[7] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The Office Marathon: Robust Navigation in an Indoor Office Environment," in *IEEE International Conference on Robotics and Automation*, 2010.

[8] P. Martin, E. Johnson, T. Murphey, and M. Egerstedt, "Constructing and implementing motion programs for robotic marionettes," *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 902–907, 2011.

[9] A. Martinelli, "The odometry error of a mobile robot with a synchronous drive system," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 399–405, 2002.

[10] N. Roy and S. Thrun, "Online self-calibration for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 3, 1999, pp. 2292–2297 vol.3.

[11] L. Teslić, I. Škrjanc, and G. Klančar, "Ekf-based localization of a wheeled mobile robot in structured environments," *Journal of Intelligent & Robotic Systems*, vol. 62, no. 2, pp. 187–203, 2011.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Mass.: MIT Press, 2005.

[13] C. Wang, "Location estimation and uncertainty analysis for mobile robots," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 1988, pp. 1231–1235.

SVM Classification of EMG Signals for Control of a Robotic Hand

Jakob Ziegler[1], Maximilian Kallinger[1], Hubert Gattringer[1] and Andreas Müller[1]

Abstract— **Over the last decades, research on robotic devices for the replacement of lost limbs, such as actively actuated prostheses, has been intensified. In this paper we present the development of a low-cost robotic arm together with a sensor bracelet for electromyographic (EMG) control. With a total cost of about $300 the introduced system aims at contributing to the development of inexpensive alternatives for commercially available products. The muscular activity measured on two positions on the forearm is used to recognize pre-defined hand motions, or more precisely gestures and grasps. To this end, the machine learning method support vector machines (SVM) is used as classification algorithm, achieving a mean classification accuracy of 89.45 %, with a minimum accuracy of 85.63 % and a maximum accuracy of 98.02 % for the individual motions. The recognized gestures and grasps are then executed on the robotic device, whereby all five fingers of the hand segment are actuated with servo motors and cables.**

I. INTRODUCTION

In 2008, one in 190 adults in the United States was living with lower or upper limb loss. This number is projected to double until 2050 [1]. Although upper limb amputations are about 20 times less frequent than lower limb amputations, their impact on the quality of life is more detrimental [2]. The upper limbs are particularly important for most activities of daily living, especially feeding and personal hygiene, as well as for communication.

Advances in mechanical systems more and more offer the possibilities to restore the lost functionality and dexterity of impaired or absent upper extremities. Accordingly, sophisticated prostheses are commercially available nowadays. They are however very costly with prices up to $75000 [3], which hinders a broader usage [4]. Latest advances in additive manufacturing techniques, like 3D printing, might help to overcome this issue. A recent study shows that nearly every second adult with traumatic major upper limb amputations abandons the prosthetic device. Next to comfort and weight, unsatisfactory function has been identified as one of the main reasons for rejection. This further fosters the research interest in intelligent control of hand and arm orthoses and prostheses, which already gained increased attention over the past years. As control input, biological signals like electroencephalography (EEG) or especially electromyography (EMG) are thereby preferred over mechanical signals, e.g. delivered by inertial measurement units or force sensors [5]. In contrast to mechanical signals, biological signals offer the advantage to be accessible prior to the movement, as they are the trigger for muscle contraction, and are therefore directly related to the movement intention. Exploiting this so-called electro-mechanical delay has the potential to minimize the lag between the motion intention of the user and the reaction of the robotic system.

This paper aims at contributing to the development of a low-cost robotic arm with electromyographic control of the hand segment. Two EMG sensors mounted on a specially designed bracelet measure muscle activities on the forearm. Based on these EMG signals pre-defined hand motions are recognized utilizing the well-known machine learning technique support vector machines (SVM) [6].

II. ROBOTIC ARM

In 2012 the french sculptor and designer Gaël Langevin started to create *InMoov* [7], an open source 3D printed humanoid robot. 3D printing templates and construction recommendations of this robot served as a basis for the present work. The current version of the robotic arm consists of a hand and a forearm segment, see Fig. 1. Except for screws, cables and springs all structural parts of the robotic arm are 3D printed. The five fingers of the hand segment can be actuated individually via cables (Fig. 2), which are attached to servo motors with up to 196 N pulling force at 7.4 V supply voltage. Pretension of the cables is ensured with springs. The original joints connecting the individual finger segments are replaced by 3D printed parts made of elastic filament, which results in less jerky finger motions. As central control unit a micro controller (Arduino Due) was used. Together with the bracelet measuring the muscular activity of the human forearm, discussed in the next section, the total cost of the system is about $300. In addition to the *rest* position of the hand, four hand gestures or grasps are implemented: *stretched*, *fist*, *rock* and *tripod*. Fig. 3 shows a comparison of the pre-defined human hand motions and their implemented robotic counterparts.

III. ELECTROMYOGRAPHIC CONTROL

A. EMG Bracelet

In addition to conventional control of the robotic hand, the device shall be controllable by electromyographic signals. To this end, low-cost surface EMG sensors (*MyoWare*) are embedded in a specially designed forearm bracelet shown in Fig. 4. The bracelet is made of elastic 3D printable filament and consists of sensor mountings, which can be assembled in a modular manner. This allows for an easy integration of additional EMG sensors. In the current version two EMG sensors are utilized, the bracelet therefore consists of two interlocked parts and is fixed to the user's forearm with a fastener similar to a wristwatch. Each EMG sensor

[1]Jakob Ziegler, Maximilian Kallinger, Hubert Gattringer and Andreas Müller are with Institute of Robotics, Johannes Kepler University of Linz, 4040 Linz, Austria jakob.ziegler@jku.at

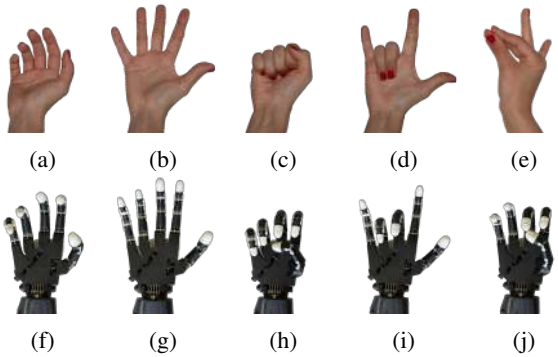
Fig. 1: Fully assembled robotic arm.

(a) (b)

Fig. 2: Flexion (a) and extension (b) of a single finger.

consists of two electrodes measuring the muscle activity and one reference electrode. The current geometry of the bracelet is meant to allow a positioning of the two sensors according to forearm muscle groups responsible for finger flexion and finger extension, respectively. As every hand and finger motion imposes specific requirements on the musculoskeletal system, there also have to be motion-specific electromyographic activity patterns that are anticipated to be measurable. Fig. 5 exemplarily shows the EMG signals recorded for different hand motions. As can be seen there are noticeable differences in the EMG signals of the individual gestures or grasps, which shall be used for hand gesture recognition using support vector machines (SVM).

In general, EMG signals are affected by many factors that can be classified e.g. on their origin or their impact on the signal [8]. Extrinsic noise sources, like cable motion artifact



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

(f) (g) (h) (i) (j)

Fig. 3: Pre-defined hand motions: rest position (a,f), stretched (b,g), fist (c,h), rock (d,i) and tripod (e,j).

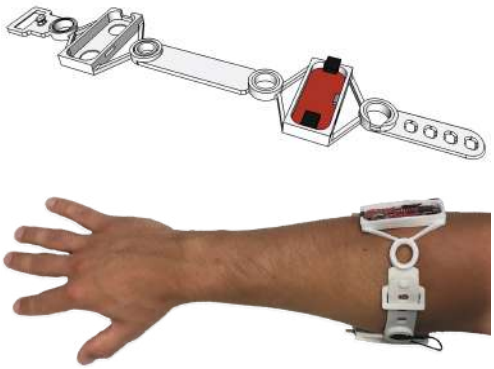


Fig. 4: The designed EMG bracelet for the usage of two EMG sensors (top) and positioning of the bracelet on the forearm of the user (bottom).

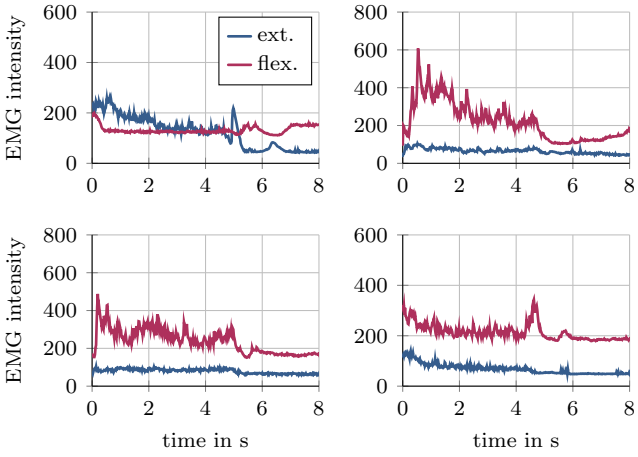


Fig. 5: EMG signals of the hand motions stretched (top left), fist (top right), rock (bottom left) and tripod (bottom right). The motions are maintained in the first 5 s of the graphs.

or power line noise, can be effectively handled by up-to-date electronics. On the contrary, intrinsic noise sources like thermal noise and noise originating from the electrode-skin interface as well as from the relative movement between muscle and skin can be difficult to deal with [9]. The pattern of the measured EMG signal during a specific movement is additionally dependent on e.g. electrode placement, on the number of recruited muscle fibers and fatigue [10], varies from user to user and from day to day [11], and can reach peak-to-peak values ranging from 100 µV to 10 mV [12]. To increase the interpretability of muscle activities, the according measured raw signals have to be transformed into features. There exist three main categories that EMG features can be assigned to, which are time domain, frequency domain and time-frequency domain [13]. An extensive overview of various EMG features can be found in [14], [15]. The features used in this paper are chosen out of a larger feature set, as they provided the best results in terms of computational effort and classification accuracy. Mean value, standard deviation and mean frequency of the EMG signals within a moving window of fixed length were used as features.

B. Support Vector Machines

Support vector machines are frequently used in the field of machine learning. Basically a SVM classifier separates data with a so-called *hyperplane* in a way that the gap, or margin, between the defined classes is maximized.

With the aid of the simple example illustrated in Fig. 6a) an outline of the SVM classification algorithm is described as follows: let $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1,...,m$ be data points of a training data set in an n-dimensional feature space that are classified either as □ or ○. Define a vector $\mathbf{w}$, which is perpendicular to the separating hyperplane (assuming such exists), a scalar b and a vector of labels $\mathbf{y} \in \mathbb{R}^m$, $y_i \in \{1,-1\}$ with $y_i = 1$ for □-samples and $y_i = -1$ for ○-samples, such that

$$y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) - 1 \geqslant 0. \tag{1}$$

Given two samples $\mathbf{x}_\square$ and $\mathbf{x}_\bigcirc$ that lie directly on the respective class borders, the margin between the classes is defined as $d = |\mathbf{x}_\square - \mathbf{x}_\bigcirc|^\mathsf{T} \frac{\mathbf{w}}{\|\mathbf{w}\|}$. With the definition of $y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) - 1 = 0$ for samples directly on the class borders, d can be written as $d = 2/\|\mathbf{w}\|$. The corresponding optimization problem is then

$$\min \frac{1}{2}\|\mathbf{w}\|^2, \tag{2}$$

subject to (1), which can be formulated by denoting

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_i \alpha_i [y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) - 1], \tag{3}$$

where α_i are Lagrange multipliers. In order to find $\mathbf{w}$ and b that minimize (3), the function

$$\Theta(\alpha) = \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha) \tag{4}$$

is introduced. Since (4) is an unconstrained minimization problem, we know at the minimum point that $\frac{\partial L}{\partial \mathbf{w}} = \frac{\partial L}{\partial b} = 0$, which delivers

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{5}$$

$$\sum_i \alpha_i y_i = 0, \tag{6}$$

thus (4) may be re-expressed as

$$\Theta(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\mathsf{T}\mathbf{x}_j. \tag{7}$$

Solving (7) for α_i, s.t. $\alpha_i \geq 0$ and $\sum_i \alpha_i y_i = 0$ finally gives the solution of the optimization problem. Only data samples with a nonzero α_i have influence on the definition of the class borders and are therefore called *support vectors*. The class of a new data sample $\mathbf{u}$ can now be determined with the decision function $h(\mathbf{u})$, where a positive sign means $\mathbf{u}$ is of class □ and a negative sign means $\mathbf{u}$ is of class ○

$$h(\mathbf{u}) = \operatorname{sgn}(\mathbf{w}^\mathsf{T}\mathbf{u} + b). \tag{8}$$

In the presence of outliers in the data set a slack variable $\xi_i \geq 0$ and a weight factor $C > 0$ can be introduced in order to increase the robustness of the classification and to avoid overfitting. This means outliers are allowed but penalized, the corresponding optimization problem is then

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \tag{9}$$

$$\text{s.t.} \quad y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) \geqslant 1 - \xi_i \tag{10}$$

which again results in (7), but with the additional constraint of $0 \leqslant \alpha_i \leqslant C$. As (7) shows, the classification problem mainly depends on the dot product $\mathbf{x}_i^\mathsf{T}\mathbf{x}_j$ which can be seen as some kind of similarity measure. SVM also provides a comfortable method to classify data points that are not linearly separable in feature space (as illustrated in Fig. 6b) by mapping the data into a higher dimensional space using a function $\Phi : \mathbb{R}^n \to \mathbb{R}^l$, $l > n$, $\mathbf{x}_i^\mathsf{T}\mathbf{x}_j \mapsto \Phi(\mathbf{x}_i)^\mathsf{T}\Phi(\mathbf{x}_j)$. If the dimension of this new feature space is high enough, a linear separation eventually becomes possible, now depending on the dot product of the transformed data. The idea is now to use a kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^\mathsf{T}\Phi(\mathbf{x}_j)$ that delivers the dot product of a higher dimensional space without actually having to calculate the transformation. This renders the optimization problem and the resulting decision function to

$$\Theta(\alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{11}$$

$$h(\mathbf{u}) = \operatorname{sgn}(\sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{u}) + b) \tag{12}$$

where $k(\mathbf{x}_i, \mathbf{u})$ again serves as measure of similarity. Two frequently used kernel functions are the radial basis function (RBF) kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma(\|\mathbf{x}_i - \mathbf{x}_j\|)^2} \tag{13}$$

and the polynomial kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \left(\gamma \mathbf{x}_i^\mathsf{T}\mathbf{x}_j + r\right)^d, \tag{14}$$

which is applied in this paper.

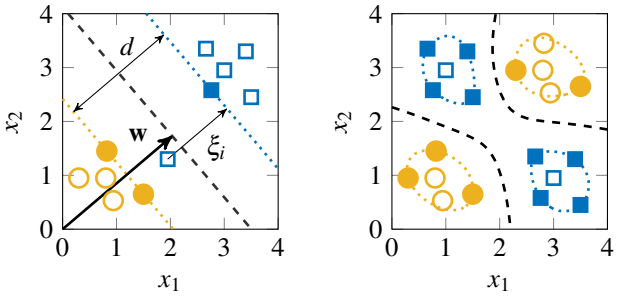


Fig. 6: a) Basic idea of SVM with two-dimensional data samples linearly separated into the two classes ○ and □ by a hyperplane (dashed line), where the dotted lines symbolize the class borders, and b) separation of data, which are not linearly separable in feature space, using an RBF kernel. The support vectors are illustrated by filled markers.

C. Classification Model

To generate a training dataset, EMG signals were measured at a frequency of 500 Hz while each hand gesture was maintained 30 times for approximately 5 s, followed by another 5 s of rest. Accordingly, a classification of the five classes *rest*, *stretched*, *fist*, *rock* and *tripod* is considered. Since SVM classification belongs to the group of supervised learning methods, a classification model has to be generated in advance, i.e. calculating the optimal values for α and b according to some training data, to be able to solve (12). Each training data sample has to be labeled with respect to one of the defined classes. A push button was used during the measurements to manually label each feature vector $\mathbf{x}_i \in \mathbb{R}^n$, containing the $n = 6$ features (3 per EMG sensor), of the $i = 1,...,m$ samples per window. As stated above the mean frequency (MNF) [14] is calculated next to mean value und standard deviation of the EMG signal:

$$MNF = \frac{\sum_{j=1}^{m} f_j P_j}{\sum_{j=1}^{m} P_j}, \tag{15}$$

where f_j is the frequency of the EMG power spectrum P_j at frequency bin j calculated via fast Fourier tranform (FFT). All features are calculated within non-overlapping time windows of 300 ms.

Due to the used polynomial kernel function, the classification algorithm is dependent on the parameters C, γ, r and d, which can be chosen arbitrarily. The overall accuracy and robustness of the classification can be increased by fine tuning those parameters, which was done empirically with parameter values set to $C = 1$, $\gamma = 1$, $r = 1$ and $d = 2$. A 5-fold cross validation was employed to assess the classification performance.

IV. EXPERIMENTAL RESULTS

A summary of the classification results presented as a confusion matrix is shown in Tab. I. As can be seen, the highest classification accuracy is achieved for the stretched hand. This seems reasonable, as the muscle activities differ visibly from the other gestures (Fig. 5), which show somewhat similar EMG patterns. Especially the EMG signals of the gestures *fist* and *rock*, which essentially is a fist with two stretched fingers, show high resemblance. This explains the quite frequent misinterpretation of the gesture *fist* as *rock* and vice versa, as it is shown in Tab. I. Overall the SVM classification of the hand gestures shows a mean accuracy of 89.45 %. This complies with conducted live experiments, where it could be observed that roughly 17-18 out of 20 hand gestures are recognized correctly.

Generally, the introduced setup and methodology showed good robustness considering experiments with the same person but on different days. Nevertheless, the EMG measurement and therefore also the classification accuracy is sensitive to the placement of the EMG sensors, which should be carefully attached to the respective positions on the forearm. In this regard, the developed bracelet effectively supported the repeatability of the sensor placement. When another person is using the bracelet for hand gesture recognition we recommend to carry out a training phase to re-calibrate the classification model, as otherwise the rate of correct identifications might drop drastically.

TABLE I: Classification accuracy

		predicted class				
		rest	stretched	fist	rock	tripod
true class	rest	**88.56 %**	2.03 %	1.66 %	3.87 %	3.87 %
	stretched	1.78 %	**98.02 %**	-	-	0.20 %
	fist	1.97 %	-	**86.00 %**	11.44 %	0.59 %
	rock	4.11 %	-	4.72 %	**85.63 %**	5.54 %
	tripod	4.49 %	0.41 %	0.82 %	5.31 %	**88.98 %**

V. SUMMARY AND OUTLOOK

In this paper, a low-cost robotic arm is presented, where the fingers of the hand segment are controllable by EMG signals. Thereby, slightly adapted open-source templates are used to 3D print the structural parts. The actuation is realized with servo motors and cables and allows for moving the fingers individually. EMG sensors measuring the muscle activity on the forearm, as well as the supervised learning method SVM are employed to recognize defined hand gestures, which are executed by the robotic system. With a specially designed bracelet, made of 3D printed modules of elastic material, currently two EMG sensors can easily be placed at the appropriate positions.

Note that the training and the validation of the SVM classification model was done based on the data of one single person. As one of the next steps, EMG data from additional persons will be incorporated, which allows for assessing the classification accuracy and overall system performance in a more general way. With the current setup and empirically tuned parameters a decent classification accuracy of the hand gestures of 89.45 % on average can be achieved. Nevertheless, the system would benefit from additional EMG sensors. Some of the defined hand gestures have significant similarities with respect to the measured muscle activities and are prone to confusion. Thus, further information of muscle activities on additional forearm regions will presumably improve the overall classification accuracy. With the modular conception of the EMG bracelet such an upgrade can be realized without big effort. In order to further raise the classification accuracy, it might be necessary to put some extra effort into the parameter optimization and a deeper analysis of the effect of different kernel functions. Also, the incorporation of additional features and the selection of a final set of features based on their significance might increase the overall accuracy. Different degrees of overlapping concerning the time windows for feature calculation and the according effect on the classification result will as well be part of future investigations. The Goal thereby is to minimize the lag between the motion intention of the user and the reaction of the robotic system, which should not exceed approximately 125 ms [16].

ACKNOWLEDGMENT

This work has been supported by the COMET-K2 Center for Symbiotic Mechatronics of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria.

REFERENCES

[1] K. Ziegler-Graham, E. J. MacKenzie, P. L. Ephraim, T. G. Travison, and R. Brookmeyer, "Estimating the prevalence of limb loss in the united states: 2005 to 2050," *Archives of physical medicine and rehabilitation*, vol. 89, no. 3, pp. 422–429, 2008.

[2] R. F. Baumgartner, "Guest editorial: Upper extremity amputation and prosthetics," *Journal of Rehabilitation Research and Development*, vol. 38, no. 4, pp. vii–x, 2001.

[3] J. Ten Kate, G. Smit, and P. Breedveld, "3d-printed upper limb prostheses: a review," *Disability and Rehabilitation: Assistive Technology*, vol. 12, no. 3, pp. 300–314, 2017.

[4] L. A. Wheaton, "Neurorehabilitation in upper limb amputation: understanding how neurophysiological changes can affect functional rehabilitation," *Journal of neuroengineering and rehabilitation*, vol. 14, no. 1, pp. 1–12, 2017.

[5] J. L. Pons, "Rehabilitation exoskeletal robotics," *IEEE Engineering in Medicine and Biology Magazine*, vol. 29, no. 3, pp. 57–63, 2010.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] The InMoov website. [Online]. Available: http://www.inmoov.fr

[8] C. J. De Luca, "The use of surface electromyography in biomechanics," *Journal of applied biomechanics*, vol. 13, no. 2, pp. 135–163, 1997.

[9] C. J. De Luca, L. D. Gilmore, M. Kuznetsov, and S. H. Roy, "Filtering the surface emg signal: Movement artifact and baseline noise contamination," *Journal of biomechanics*, vol. 43, no. 8, pp. 1573–1579, 2010.

[10] D. Tkach, H. Huang, and T. A. Kuiken, "Study of stability of time-domain features for electromyographic pattern recognition," *Journal of neuroengineering and rehabilitation*, vol. 7, no. 1, p. 21, 2010.

[11] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Critical Reviews™ in Biomedical Engineering*, vol. 30, no. 4-6, 2002.

[12] C. D. Joshi, U. Lahiri, and N. V. Thakor, "Classification of gait phases from lower limb emg: Application to exoskeleton orthosis," in *Point-of-Care Healthcare Technologies (PHT), 2013 IEEE*. IEEE, 2013, pp. 228–231.

[13] M. A. Oskoei and H. Hu, "Myoelectric control systems—a survey," *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.

[14] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for emg signal classification," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7420–7431, 2012.

[15] N. Nazmi, M. A. Abdul Rahman, S.-I. Yamamoto, S. A. Ahmad, H. Zamzuri, and S. A. Mazlan, "A review of classification techniques of emg signals during isotonic and isometric contractions," *Sensors*, vol. 16, no. 8, p. 1304, 2016.

[16] T. R. Farrell and R. F. Weir, "The optimal controller delay for myoelectric prostheses," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 15, no. 1, pp. 111–118, 2007.

An Evaluation of Mask Estimation Methods for Transparent Objects*

Veronika Rettner[1], Jean-Baptiste Weibel[1] and Markus Vincze[1]

***Abstract*— The ageing population in many countries and a shortage of skilled health care workers are becoming a central issue in many western societies. Service robots can be a part of the solution by providing assistance for the elderly and extending their autonomy. This task requires a good understanding of their environment. The perception and manipulation of transparent objects is, however, extremely challenging despite the pervasiveness of such objects in the daily life of people needing medical assistance. Understanding the strengths, limitations and necessary improvements of state-of-the-art methods is therefore essential to get closer to this objective. In this work, a new real-world dataset of 640 images of various transparent objects of different shape, material and complexity is introduced to evaluate selected algorithms for transparent object detection. This includes traditional depth- and IR-based approaches as well as pre-trained CNN pipelines, namely TOM-Net, ClearGrasp, TransLab, Trans2Seg. TransLab shows the overall best results with an IoU as high as 55.85%, followed by Cleargrasp and Trans2Seg, highlighting the overall better performance of CNN based approaches. With metrics well below 4%, TOM-Net on the other hand shows the lowest scores on our dataset. Thin plastic objects and complex shapes including fine details such as tubes are identified as challenges for transparent object detection. Finally, object poses are also annotated in the dataset created enabling further works in the area.**

I. INTRODUCTION

With the current progress in robotics and in perception specifically, a service robot assisting the elderly or supporting the staff in hospitals is becoming more realistic than ever. Such assistance, however, requires the ability to handle transparent glass or plastic containers, widely used in the medical field as they enable users to see the content and facilitate sterilization. Transparent objects are hard to work with due to their non-Lambertian reflection properties. RGB representations are dependent on the background of the container and widely used RGB-D sensors like the Intel RealSense or the Microsoft Kinect v1 fail to predict the depth of such objects, resulting in zero or invalid depth values. Such sensors project a known pattern on the scene in the near infrared spectrum, but do not account for its view-dependant reflections and refractions. Figure 1 shows an example RGB-D image of transparent objects captured with a RealSense D435 camera. While there are sensors that are more suitable to handle transparency, such as light-field cameras, regular RGB-D cameras remain attractive due to their ubiquity and lower cost. Until now only few methods exploit the available depth data to obtain more reasonable shape information. Most approaches proposed in the literature predict silhouettes of transparent objects solely from RGB data. To provide a more precise picture, the performance of several state-of-the-art methods for transparent mask prediction is compared. For this purpose, a new dataset containing various transparent household goods made from different materials and with different geometric complexity is collected. RGB, depth and infrared images are captured with camera poses covering the entire half-sphere around the objects. The different methods evaluated show big differences in performance, with RGB-based CNN approaches giving the best overall results. Details like transparent tubes or handles are challenging for all methods, as well as thin-walled plastic objects with simpler geometries.

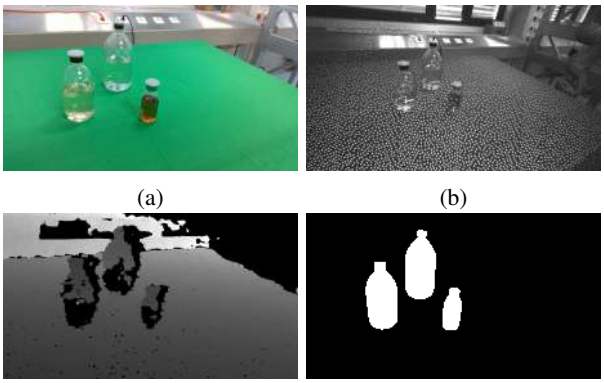


Fig. 1: Different input data, (a) RGB image, (b) IR image and (c) depth image, of transparent objects captured with a RealSense D435 camera and the corresponding groundtruth mask (d). The depth image shows that the depth pixels corresponding to transparent objects are mostly invalid.

In this paper, Section II reviews the relevant literature, Section III describes the process used to efficiently collect the dataset presented, and Section IV describes the evaluation procedure and the results obtained. Finally, Section V concludes and discusses further works.

II. RELATED WORKS

In this section, we first detail the existing approaches using the projected pattern or the predicted depth from dot pattern-based RGB-D sensors to estimate transparent objects' masks, and then review the existing approaches using RGB to estimate transparent objects' masks.

*The research leading to these results has received funding from EC Horizon 2020 for Research and Innovation under grant agreement No. 101017089, TraceBot, and the Austrian Science Foundation (FWF) under grant agreement No. I3968-N30 HEAP

[1]All authors are with the Vision for Robotics Laboratory, Automation and Control Institute, TU Wien, Austria. weibel, vincze@acin.tuwien.ac.at

Object detection is challenging for RGB-D sensors in the case of transparent, highly reflective materials or generally any non-Lambertian materials. These materials affect the appearance of the projected pattern in the scene in view specific manners, affecting both block matching and stereo matching depth prediction methods. Block matching methods relying on the knowledge of the dot pattern fail more consistently as the pattern is non-linearly transformed. This invalid depth information was used in combination with the GrabCut segmentation algorithm to extract silhouettes [7], [4]. Alt et al. [1] used depth maps from several views and looked for depth inconsistencies caused by transparent objects. Boundary label predictions from appearance and depth features were integrated into a Markov Random Field (MRF) model for glass object segmentation in [19]. The projection of IR light on the scene and its distortion by transparent objects is also very informative. Hagg et al. [5] exploited the distortion of the IR pattern of an active RGB-D camera to recognise reflective and transparent objects. Ruppel et al. [14] directly used the raw infrared images and generate a transparency candidate map from the scattered projection pattern. The object detection was performed by a standard blob detection algorithm. Works based on reflection and distortion of light can be pushed even further with specialized sensors. The unique properties of light-field images [11], [24], [25] and polarization images [8] were used to allow the segmentation of transparent objects. RGB images also contain information about transparent objects. McHenry et al. [12] used cues for transparency like specular highlights or distortion of the background texture to train an SVM classifier for transparent edge and region detection. Such classical approaches have since largely been outperformed by deep learning models and object detection architecture like SSDs are used to predict bounding boxes of transparent objects from RGB images in [9]. Stets et al. [17] used a large-scale synthetic dataset to train a CNN with a VGG16-Net backbone, showing good generalisation to real-world scenes. This was further improved by TransLab, a segmentation network trained on large-scale real-world data which was introduced by Xie et al. [21]. Furthermore, a Transformer-based segmentation pipeline [22] was later trained on the same dataset and outperformed purely CNN-based methods. Very recently, a cascade network architecture was proposed by [6], introducing modules with residual learning and point-based graph convolution to enhance boundary prediction. More information can also be inferred about transparent objects, such as how they affect the path of light. In TOM-Net [2] a transparent object mask, an attenuation mask, and a refractive flow is predicted using an adapted mirror-link CNN [16]. While other image matting methods for transparent objects depend on specific backgrounds or patterns [26], [20], TOM-Net was trained on an entirely synthetic dataset of transparent objects rendered in front of different scenes and patterns. Finally, ClearGrasp [15] uses both RGB and the noisy depth as input for transparent object depth prediction. The transparent object mask and the scene normals are predicted from the RGB image and the noisy depth is completed by optimization according to the predicted surface normals.

Fig. 2: Selected scenes from our dataset showing the wide range of objects of different complexity and materials.

Fig. 3: Setup consisting of a robot manipulator and an RGB-D camera.

III. DATASET COLLECTION METHOD FOR TRANSPARENT OBJECTS

For evaluation of different approaches, a new dataset with RGB-D images of two different types of transparent objects is introduced: It includes common transparent household items as well as containers used for various medical applications. Objects of different complexity are selected, ranging from simple symmetric objects and flat objects to objects with more and finer details as well as vessels filled with contents. An overview over some scenes of our dataset is given in Figure 2.

In addition to RGB and depth images, the dataset also contains infrared images and camera poses. Overall, it consists of 640 images in 10 scenes. Since 64 views are used to capture each scene, the dataset also is suitable for evaluation of multi-view reconstruction methods. Indeed, the sampled views cover the entire half-sphere above the scenes as illustrated in Figure 4.

An overview of published datasets featuring transparent objects is given in Table I. ClearGrasp is listed twice as both a synthetic and a real-world dataset are reported. Our dataset includes all information about objects in the scene, including their pose, is only made of real scenes, and has a better view coverage than most datasets.

A. Acquisition

The setup used for capturing the dataset consists of a 6 DoF robotic manipulator with an RGB-D camera attached to its end effector (see Figure 3). A RealSense D435 camera

Dataset	Type	# Images	# Objects	Clutter	Mask	Depth	Pose
Stets et al. [17]	synthetic	80,000	600		x	x	
ClearGrasp [15]	synthetic	50,000	6	x	x	x	x
ClearGrasp [15]	real-world	286	10	x	x	x	x
KeyPose [10]	real-world	48,000	15		x	x	x
Trans10K [21]	real-world	10,428	10k+	x	x		
TODD [23]	real-world	15,000	6	x	x	x	
Our dataset	real-world	640	15	x	x	x	x

TABLE I: Overview of datasets published online at the time of this work, which feature transparent objects.

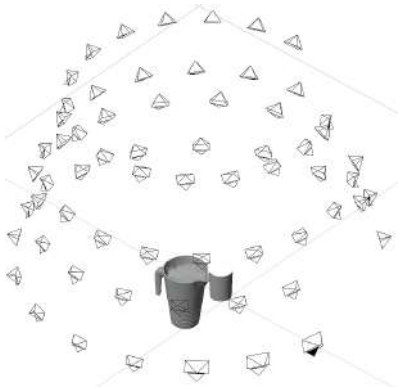


Fig. 4: Sketch of the camera poses used to capture the data of one scene.

(a) (b)

Fig. 5: Outlines of the 3D models (a) are projected onto RGB images (b), allowing the alignment with the object contours.

is selected since it allows both the collection of RGB-D images and infrared images. For calibration of the eye-in-hand setup fiducial markers are used. For each scene, one or more objects are placed on a table and the camera is moved along a predefined trajectory, capturing 64 frames and storing the camera poses obtained using the well-calibrated arm kinematics. A sketch of camera poses corresponding to images taken during one scene is depicted in Figure 4. The trajectory runs circularly around the scene and the angle of the camera relative to the table plane is changed every 16 frames resulting in 4 different angles. The distance between the camera and the objects remains in the range of 60-100cm throughout the capture of the entire dataset. The light source is fixed above the setup, therefore the camera angle also changes in regard to the light source.

B. Annotation

The transparent objects of our dataset are annotated using the annotation toolkit introduced by [18]. Here, 3D models of the transparent objects are projected to the RGB images considering intrinsic and extrinsic parameters. The object pose is obtained via Blender [3] by aligning the outlines of the projected models with the RGB images (see Figure 5).

The derived object poses are then used to render ground truth masks for the RGB images using the respective camera poses. The 3D models of the transparent objects were either created using CAD software or obtained by spray-painting the objects and then applying object reconstruction methods for opaque objects.

IV. EXPERIMENTS

A. Experimental Setup:

We compared different approaches for mask estimation of transparent objects, including one based on invalid depth data in combination with GrabCut, another one based on raw infrared images and several CNN-based pipelines using purely RGB information.

Invalid or zero-depth values can be used to extract silhouettes of transparent objects. However, not all transparent pixels cause zero depth values, resulting in an incomplete silhouette either with holes or incomplete parts. Therefore, morphological closing with a circular kernel and contour detection are applied on the inverted image to extract object candidates. Furthermore, the available RGB information is incorporated to improve the mask using the GrabCut segmentation algorithm [13]. As initialization for GrabCut, the zero depth masks are processed and labelled to create rough trimaps: The mask is dilated and the true pixels are labelled as possible foreground, while the rest is labelled as background. Then a copy of the mask is eroded and the true pixels are labelled as definite foreground and added to the trimap. Finally, the segmentation algorithm is applied on each pair of RGB image and depth-based trimap.

In addition, the IR pattern of the RealSense D435 is hardly visible on the surface of transparent objects, resulting in failure of the intern depth prediction of the camera. In this work, the algorithm proposed by Ruppel et al. [14] is used to exploit the lack of IR speckles on transparent surfaces to obtain a transparency candidate map. The algorithm is adapted to work with the pattern of the RealSense D435 camera, which predicts depth by Active IR Stereo Vision, in comparison to the Structure Light camera used in the original

work. A high-pass filter is applied to the normalised IR image and subsequently blurred with a median filter. After repeated dilation with a circular kernel, a threshold is applied and the resulting mask is filtered with a blob detection algorithm by size.

For evaluation of CNN-based methods, four different deep learning pipelines are tested on the proposed dataset for mask prediction of transparent objects: TOM-Net [2], ClearGrasp [15], TransLab [21], and Trans2Seg [22]. Here, we use the pre-trained models, which are already available online. From TransLab and Trans2Seg only the binary masks are considered for evaluation, excluding the prediction of the object class. In addition, the "tiny" configuration of transformer is used for Trans2Seg training and inference.

B. Results:

Table II shows the recall, precision, F1 score and IoU achieved on our dataset for selected approaches. Masks from invalid depth are also included as a baseline. TransLab shows the highest precision, F1 score and IoU, and even the recall is rather comparable to ClearGrasp. ClearGrasp has the highest recall of around 76%, but scores significantly lower than TransLab for all other metrics. The F1 score and IoU of Trans2Seg are similar to the results of ClearGrasp, but the recall is much lower. The precision exceeds ClearGrasp by around 15% in absolute value. A better performance of Trans2Seg was expected, since it is the follow-up work of TransLab. However, for inference on our dataset, the "tiny" configuration of the transformer was used, which can explain the results. The GrabCut-based approach using invalid depth data clearly shows an improvement in comparison to the baseline masks from raw invalid depth. It achieves a recall of above 58%, but has otherwise lower scores than the CNN pipelines discussed above. The IR-based approach performs worse on average than invalid depth-guided GrabCut, even yielding a recall lower than the baseline masks. The reasons may lie in the high angular and material dependency of this method as well as the relatively sparse pattern used by the RealSense D435 (e.g. when compared to the Kinect). Although it is not optimised for exact segmentation, the results still indicate the potential of IR-based approaches. In contrast, TOM-Net performs much worse than the invalid depth masks and every other algorithm investigated in this study. TOM-Net achieves average scores of just a few percent for each metric on the dataset. A reason might be that it is trained just on synthetic data, whereas the other networks are trained on real-world images. Please note that pre-trained models are used in this evaluation, i.e differences in the results might be attributed to both the different training datasets as well as the algorithms themselves. In addition, an angular dependency of the metrics is found for all approaches, shown in Figure 6 for F1 score, IoU, recall and precision between the angles of 14° and 55°. For TransLab and Trans2Seg, overall high values with low angular dependency are found, with a decrease in the metrics only at higher angles (between 44° and 55°). Similar results are obtained for ClearGrasp, but here also a slight decrease for low angles is visible,

Method	Recall [%]	Precision [%]	F1 [%]	IoU [%]
Invalid Depth*	43.29	27.62	30.77	19.92
Depth+GC [13]	58.84	39.25	43.06	30.92
IR-based [14]	37.03	41.56	33.13	25.05
TOM-Net [2]	3.57	3.45	2.96	1.84
ClearGrasp [15]	**75.86**	49.99	56.24	42.72
TransLab [21]	73.50	**71.67**	**67.54**	**55.85**
Trans2Seg [22]	54.02	65.62	52.86	41.86

TABLE II: Overall evaluation results of selected transparent object segmentation techniques on our dataset. The highest score for each metric is highlighted in bold.

leading to a maximum for F1 score, IoU and precision at intermediate angles. Only the recall increases slightly with the angle. The depth based methods, i.e. invalid depth and the combination with GrabCut, also show a more or less pronounced maximum at intermediate angles for all metrics except for recall, were a severe decrease is found for higher angles. The IR-based approach shows a drastic increase from very low values for all metrics at low angles up to the high values also achieved by TransLab and Trans2Seg. In contrast, TOM-Net overall yields very low metrics and no meaningful dependency on the angle is found.

The results also differ vastly for the individual objects and materials. Some examples of frames from different scenes are shown in Figure 7, featuring thick-walled glass objects in the first row, thin-walled plastic bottles in the second row, a pipe and a dustpan in third row and a medical object with plastic containers and flexible tubes in the last row. In Table III the averaged results of all frames for four scenes are shown in more detail. The best results are obtained for thick-walled glass objects (see Table III(a)), whereas thin-walled plastic objects are more challenging (Table III(b)). Also, flat lying objects with a lot of contact with the ground (Table III(c)) prove to be difficult for some approaches. As the most complex object, the medical kit is also the most difficult for mask prediction due to fine details and the cluttered scene (Table III(d)).

Based on these results, future work should take more complex cases like transparent objects containing liquids, very thin plastic objects, semi-transparent objects and fine structures into account, while also delivering reliable results for opaque objects. Also, the results suggest that the output of the CNNs could be improved by using more and diverse training data.

V. CONCLUSIONS

A comparison between different state-of-the-art approaches for mask prediction of transparent objects was given and a new real-world dataset of transparent objects featuring RGB-D images, infrared images and camera poses and manually annotated groundtruth masks was introduced. The evaluation on the dataset showed great variation in the performance for different mask prediction methods. The best overall performance was achieved by TransLab in regards of precision, F1 score and IoU, yielding values up to 71.67%, 67.54% and 55.85%, respectively. The average highest re-

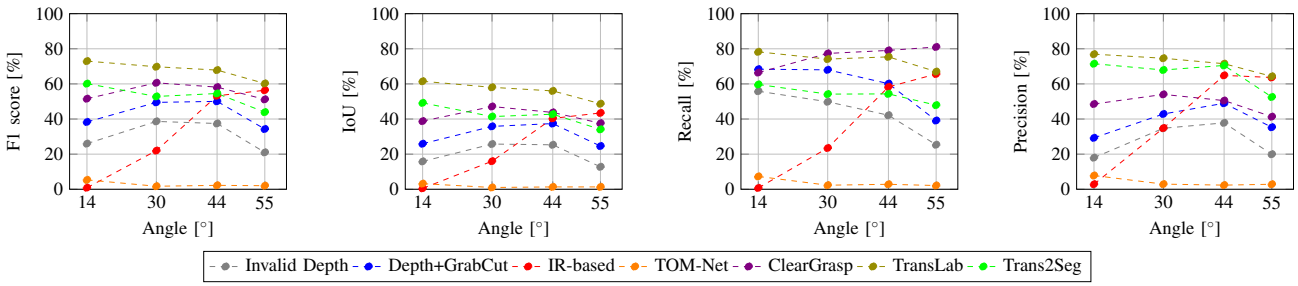
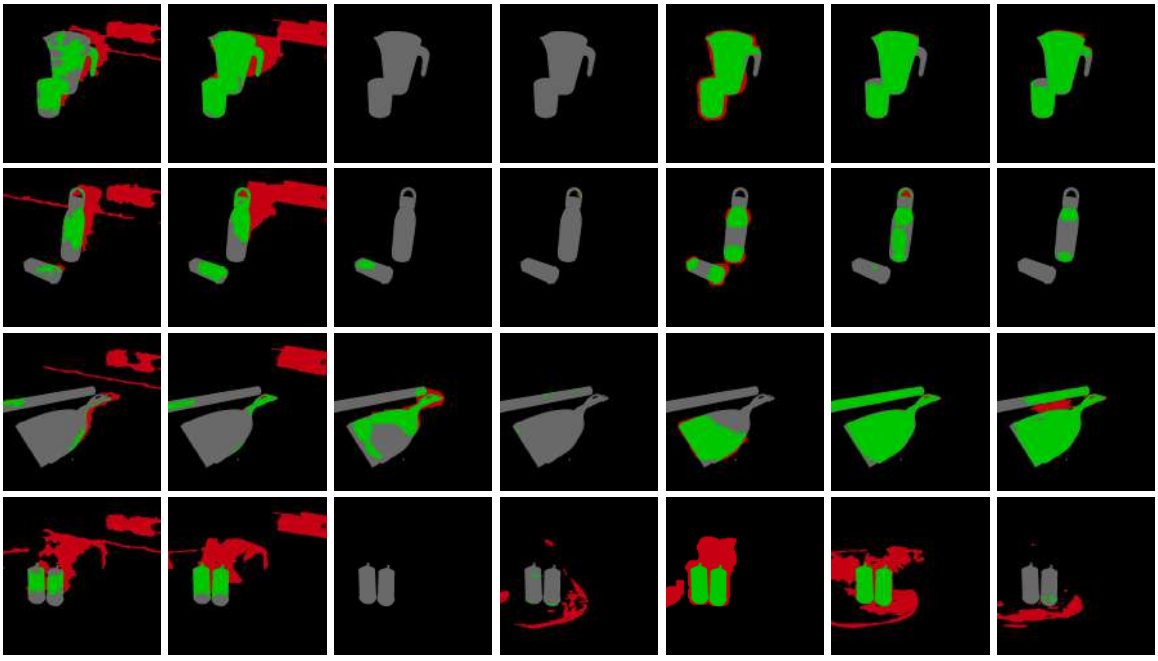


Fig. 6: Effect of different camera poses on the metrics of the selected segmentation methods. Please note that an angle of 0° corresponds to image plane and table in perpendicular position.



(a) (b) (c) (d) (e) (f) (g)

Fig. 7: Evaluation on four frames of different scenes from our dataset: (a) invalid depth mask, (b) depth + GrabCut [13], (c) IR-based [14], (d) TOM-Net [2], (e) ClearGrasp [15], (f) TransLab [21] and (g) Trans2Seg [22]. The colours indicate true positive (green), true negative (black), false positive (red) and false negative (grey) pixels.

Method	Recall [%]	Precision [%]	F1 [%]	IoU [%]
Invalid Depth	61.64	45.79	51.22	36.22
Depth+GC [13]	79.59	60.62	67.5	53.14
IR-based [14]	46.41	51.08	46.79	39.75
TOM-Net [2]	9.64	12.54	9.74	6.74
ClearGrasp [15]	93.04	66.09	76.09	62.61
TransLab [21]	92.78	**93.92**	**93.13**	**87.39**
Trans2Seg [22]	**93.32**	90.98	91.41	85.21

(a) Scene containing glass objects with thick walls.

Method	Recall [%]	Precision [%]	F1 [%]	IoU [%]
Invalid Depth	39.84	24.30	28.09	16.71
Depth+GC [13]	51.89	32.36	36.85	23.70
IR-based [14]	16.15	48.42	22.05	14.05
TOM-Net [2]	4.66	4.53	3.32	1.83
ClearGrasp [15]	**72.59**	57.73	61.46	45.76
TransLab [21]	55.81	**88.77**	**66.25**	**51.44**
Trans2Seg [22]	26.01	69.11	33.86	22.53

(b) Scene containing plastic objects with thin walls.

Method	Recall [%]	Precision [%]	F1 [%]	IoU [%]
Invalid Depth	16.99	22.70	16.32	9.45
Depth+GC [13]	24.26	36.06	25.23	16.46
IR-based [14]	20.16	55.39	27.71	17.93
TOM-Net [2]	7.40	7.63	6.17	3.50
ClearGrasp [15]	36.75	44.78	37.78	25.59
TransLab [21]	**69.23**	**90.89**	**76.56**	**64.40**
Trans2Seg [22]	39.37	71.67	46.36	35.09

(c) Scene containing a pipe and a dustpan.

Method	Recall [%]	Precision [%]	F1 [%]	IoU [%]
Invalid Depth	47.46	8.39	13.78	7.52
Depth+GC [13]	65.13	12.79	20.62	11.70
IR-based [14]	36.55	6.95	11.25	6.92
TOM-Net [2]	14.14	4.69	5.74	3.18
ClearGrasp [15]	**87.91**	**20.02**	**31.39**	**19.52**
TransLab [21]	71.47	17.59	27.45	16.70
Trans2Seg [22]	54.08	15.45	22.58	13.87

(d) Scene containing a medical object.

TABLE III: Evaluation results of transparent object segmentation for different scenes.

call was observed for ClearGrasp with 75.86%. However, the recall achieved by TransLab is quite comparable with 73.50%. This clearly suggests that TransLab is the most effective approach in this comparison. However, only pre-trained models were used and TransLab was trained on the most extensive and varied dataset in this study. On the whole, simple and thick-walled objects were the easiest for mask prediction, while plastic objects with thin walls were more challenging. All approaches struggled with complex objects like the medical kit due to the cluttered appearance and the finer details.

REFERENCES

[1] N. Alt, P. Rives, and E. Steinbach, "Reconstruction of transparent objects in unstructured scenes with a depth camera," in *2013 IEEE International Conference on Image Processing*. IEEE, sep 2013.

[2] G. Chen, K. Han, and K.-Y. K. Wong, "Tom-net: Learning transparent object matting from a single image," in *CVPR*, 2018.

[3] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[4] C. Guo-Hua, W. Jun-Yi, and Z. Ai-Jun, "Transparent object detection and location based on RGB-d camera," *Journal of Physics: Conference Series*, vol. 1183, p. 012011, mar 2019.

[5] A. Hagg, F. Hegger, and P. G. Plöger, "On recognizing transparent objects in domestic environments using fusion of multiple sensor modalities," in *RoboCup 2016: Robot World Cup XX*. Springer International Publishing, 2017, pp. 3–15.

[6] H. He, X. Li, G. Cheng, J. Shi, Y. Tong, G. Meng, V. Prinet, and L. Weng, "Enhanced boundary learning for glass-like object segmentation," *arXiv preprint arXiv:2103.15734*, 2021.

[7] Y. Ji, Q. Xia, and Z. Zhang, "Fusing depth and silhouette for scanning transparent object with RGB-d sensor," *International Journal of Optics*, vol. 2017, pp. 1–11, 2017.

[8] A. Kalra, V. Taamazyan, S. K. Rao, K. Venkataraman, R. Raskar, and A. Kadambi, "Deep polarization cues for transparent object segmentation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8599–8608.

[9] M. P. Khaing and M. Masayuki, "Transparent object detection using convolutional neural network," in *Big Data Analysis and Deep Learning Applications*. Springer Singapore, jun 2018, pp. 86–93.

[10] X. Liu, R. Jonschkowski, A. Angelova, and K. Konolige, "KeyPose: Multi-view 3d labeling and keypoint estimation for transparent objects," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.

[11] K. Maeno, H. Nagahara, A. Shimada, and R.-I. Taniguchi, "Light field distortion feature for transparent object recognition," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2013.

[12] K. McHenry, J. Ponce, and D. A. Forsyth, "Finding glass," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 973–979 vol. 2, 2005.

[13] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM SIGGRAPH 2004 Papers*, 2004.

[14] P. Ruppel, M. Görner, N. Hendrich, and J. Zhang, "Detection and reconstruction of transparent objects with infrared projection-based rgb-d cameras," in *International Conference on Cognitive Systems and Information Processing (ICCSIP)*, 2020.

[15] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3634–3642.

[16] J. Shi, Y. Dong, H. Su, and S. X. Yu, "Learning non-lambertian object intrinsics across ShapeNet categories," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017.

[17] J. Stets, Z. Li, J. R. Frisvad, and M. Chandraker, "Single-shot analysis of refractive shape using convolutional neural networks," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, jan 2019.

[18] M. Suchi, B. Neuberger, T. Patten, and M. Vincze, "3d-sadt: Simple annotation & dataset toolkit for robotic vision," submitted to IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2022.

[19] T. Wang, X. He, and N. Barnes, "Glass object segmentation by label transfer on joint depth and appearance manifolds," in *2013 IEEE International Conference on Image Processing*. IEEE, sep 2013.

[20] B. Wu, Y. Zhou, Y. Qian, M. Cong, and H. Huang, "Full 3d reconstruction of transparent objects," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–11, aug 2018.

[21] E. Xie, W. Wang, W. Wang, M. Ding, C. Shen, and P. Luo, "Segmenting transparent objects in the wild," *arXiv preprint arXiv:2003.13948*, 2020.

[22] E. Xie, W. Wang, W. Wang, P. Sun, H. Xu, D. Liang, and P. Luo, "Segmenting transparent objects in the wild with transformer," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, aug 2021.

[23] H. Xu, Y. R. Wang, S. Eppel, A. Aspuru-Guzik, F. Shkurti, and A. Garg, "Seeing glass: Joint point cloud and depth completion for transparent objects," in *CoRL*, 2021.

[24] Y. Xu, H. Nagahara, A. Shimada, and R. ichiro Taniguchi, "Transcut: Transparent object segmentation from a light-field image," in *International Conference on Computer Vision (ICCV)*, 2015.

[25] Y. Xu, H. Nagahara, A. Shimada, and R.-i. Taniguchi, "Transcut2: Transparent object segmentation from a light-field image," *IEEE Transactions on Computational Imaging*, vol. 5, no. 3, pp. 465–477, 2019.

[26] S.-K. Yeung, T.-P. Wu, C.-K. Tang, T. F. Chan, and S. Osher, "Adequate reconstruction of transparent objects on a shoestring budget," in *CVPR 2011*. IEEE, jun 2011.

Trajectory Adaptation from Demonstrations with Constrained Optimization

Héctor Pérez-Villeda[1] Matteo Saveriano[1,3] Justus Piater[1,2]

***Abstract*— This paper proposes an approach for the adaptation of robot trajectories taken from a set of demonstrations. The problem is formulated as a constrained optimization problem where the set of demonstrations are used as target values to build a Quadratic Program (QP). The constraints constitute the adaptation's conditions of the new trajectory, e.g. new initial or final points or keep the trajectory within a specific range. The performance of our approach is verified in the adaptation of a set of demonstrations taken from a Panda robot for new conditions.**

I. INTRODUCTION

Imitation learning approaches aim to generalize tasks to novel situations. Most approaches are designed under a learning framework, where a given criteria is minimized. A variety of approaches exist where a set of task demonstrations are used to train a given model to provide generalization for new different conditions. Some representative approaches within this field are i.g. Task-Parameterized Gaussian Mixture Model (TP-GMM) [2] that considers as task parameters, the homogeneous transformations between arbitrary reference frames. By observing human demonstrations from each of these frames the robot is able to learn the spatial relationship between start, goal, and via points in the trajectory. Conditional Neural Movement Primitives (CNMPs) Seker et al. [3] generate motion trajectories by sampling observations from the training data and predicting a conditional distribution over target points, e.g. robot position, forces, and any task parameters. However, CNMPs have limited extrapolation capabilities. A possibility to improve the extrapolation performance is to combine imitation and reinforcement learning [1]. In order to maximize the generalization to new conditions, these models require a training process aimed to maximize the adaptation capabilities by minimizing a given loss function. However, there is no way to guarantee the conditions will be fully reached for the adaptation. Besides, the nature of some manipulation tasks requires reaching a certain level of precision for the new conditions to be adapted to. In this paper, we tackled the adaptation problem by a constrained optimization approach that uses a set of demonstrations as target points to build a linear regression model using a set of Basis Functions (BF). The conditions to meet by the adaptation are defined as constraints of the QP. In this way, it is possible to satisfy the new conditions that requires the adaptation. Our approach allows defining equality and inequality constraints at both position and velocity levels. Our approach is validated in the adaptation of a set of demonstrations taken from a Panda robot, where the adaptations involve different equality and inequality constraints at the same time.

[1] Department of Computer Science, University of Innsbruck, Technikerstrasse 21a, Innsbruck, Austria.

[2] Digital Science Center (DiSC), University of Innsbruck, Austria.

[3] Department of Industrial Engineering, University of Trento, Italy.

II. METHOD

Given a set of N observations $D = [\{t_1, \mathbf{y}_1\}, \cdots, \{t_N, \mathbf{y}_N\}]$ where $t \in \mathbb{R}$ defines the independent variable and $\mathbf{y} \in \mathbb{R}^d$ the target values of dimension d. The goal is to find a set of parameters $\mathbf{w} \in \mathbb{R}^M$ that minimize the sum of squared errors:

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{y}(t_n, \mathbf{w}) - \mathbf{y}_n)^2 \tag{1}$$

The model can be defined as a linear combination of fixed, nonlinear BF $\phi(t)$, i.e., $\mathbf{y}(t, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(t) = \mathbf{w}^{\mathrm{T}}\phi(t)$, where $M-1$ is the number of BF.

The regression problem can be rewritten as a QP that allows to impose constraints, in the form

$$\begin{aligned} \underset{\mathbf{w}^*}{\text{minimize}} \quad & \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{P}\mathbf{w} + \mathbf{q}^{\mathrm{T}}\mathbf{w} \\ \text{s.t.} \quad & \begin{cases} \mathbf{y}_l \leq \mathbf{G}\mathbf{w} \leq \mathbf{y}_u \\ \mathbf{A}\mathbf{w} = \mathbf{y}_A \\ \dot{\mathbf{y}}_l \leq \dot{\mathbf{G}}\mathbf{w} \leq \dot{\mathbf{y}}_u \end{cases} \end{aligned} \tag{2}$$

where $\mathbf{w}^*$ is the optimal vector that minimize the Sum of squared errors (SSE) given in Eq. 1; $\mathbf{P} = 2\mathbf{M}^{\mathrm{T}}\mathbf{M}$, and $\mathbf{q} = -2\mathbf{M}^{\mathrm{T}}\mathbf{c}$ defines the standard form expressions, $\mathbf{M} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}$ and $\mathbf{c} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{y}$; where $\mathbf{y} \in \mathbb{R}^{dN}$ is the stacked vector of target values and $\mathbf{\Phi} \in \mathbb{R}^{N \times M}$ is know as the *design matrix*

$$\mathbf{\Phi} = \begin{bmatrix} 1 & \phi_1(t_1) & \dots & \phi_{M-1}(t_1) \\ 1 & \phi_1(t_2) & \dots & \phi_{M-1}(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \phi_1(t_N) & \dots & \phi_{M-1}(t_N) \end{bmatrix}. \tag{3}$$

$\mathbf{A}\mathbf{w} = \mathbf{y}_A$ defines the equality constraints at the position level constructed from a predefined set of P data points $D^A = \{\mathbf{t}_A, \mathbf{y}_A\}$, where $\mathbf{y}_A \in \mathbb{R}^{P \times d}$ defines the desired values of the regression evaluated at $\mathbf{t}_A \in \mathbb{R}^P$. The matrix $\mathbf{A} \in \mathbb{R}^{P \times M}$ is calculated as $\mathbf{A} = \mathbf{\Phi}(\mathbf{t}_A)$. On the other hand, $\mathbf{y}_l \leq \mathbf{G}\mathbf{w} \leq \mathbf{y}_u$ represent the inequality constraints at the position level and is constructed from a set of Q data points $D^G = \{\mathbf{t}_G, \mathbf{y}_l, \mathbf{y}_u\}$, where $\mathbf{y}_l, \mathbf{y}_u \in \mathbb{R}^{Q \times d}$ are the lower and upper boundaries data points respectively and the matrix $\mathbf{G} \in \mathbb{R}^{Q \times M}$ is calculated as $\mathbf{G} = \mathbf{\Phi}(\mathbf{t}_G)$. The inequality constraints are used to keep the regression values evaluated at $\mathbf{t}_G \in \mathbb{R}^Q$ within the range $[\mathbf{y}_l, \mathbf{y}_u]$. Finally, $\dot{\mathbf{y}}_l \leq \dot{\mathbf{G}}\mathbf{w} \leq \dot{\mathbf{y}}_u$ represents the inequality constraints at velocity level that are defined from a set of

V datapoints $\dot{D}^G = \{\mathbf{t}_v, \dot{\mathbf{y}}_l, \dot{\mathbf{y}}_u\}$, where $\dot{\mathbf{y}}_l, \dot{\mathbf{y}}_u \in \mathbb{R}^{V \times d}$ are the lower and upper velocity boundaries datapoints respectively and the matrix $\dot{\mathbf{G}}$ is calculated as $\dot{\mathbf{G}} = \dot{\mathbf{\Phi}}(\mathbf{t}_v)$, where $\dot{\mathbf{\Phi}} \in \mathbb{R}^{V \times M}$ defines the partial derivatives of the BF $\dot{\mathbf{\Phi}} = \frac{\partial \phi(x)}{\partial x}$. The velocity inequality constraints are used to keep the velocity of the regression evaluated at $\mathbf{t}_v$ within the range $[\dot{\mathbf{y}}_l, \dot{\mathbf{y}}_u]$. This is specially useful to generate smooth trajectory motions in the reproduction of the adapted task. By solving the QP in Eq. 2, it is possible to find the optimal vector $\mathbf{w}^*$ that minimizes the sum of squares errors (1) and satisfies at the same time the equality and inequality constraints (2).

III. Results

This section presents the results obtained from applying our method to robot trajectory adaptation. For this experiment, we have used a dataset of 9 different $2D$ trajectories with 700 data points each, forming a dataset $D = \{\mathbf{t}, \mathbf{y}\}$ where $\mathbf{t} \in \mathbb{R}^{2700}$, with values within the range $[0,1]$; whereas the target values are $\mathbf{y} \in \mathbb{R}^{2700 \times 2}$. The used Basis Functions is conformed by a set of 18 functions $\phi(t) = [1, t, \sin(\alpha_0 t), \cos(\alpha_0 t), \cdots, sin(\alpha_7 t), \cos(\alpha_7 t)]$ with $\alpha_i \in \{0.1, 1, 5, 10, 20, 30, 40, 50\}$. The set of BF and their parameters were selected empirically motivated from the Fourier BF.

The first experiment is shown in the Fig. 1 Case I. The adaptation includes new initial and final points of the trajectory which are defined as two equality constraints for $t = 0$ and $t = 1$, indicated for purple markers. We have also defined a set of inequality constraints in Y axis as $\{\mathbf{t}, \mathbf{y}_u = 0.055\}$, (blue light area), used to keep the Y axis trajectory values lower than $\mathbf{y}_u$. The plot presents the results for three different tuples of initial and final points. In these results, the adapted trajectory fully satisfy the new initial and final conditions as well as the imposed inequality constraint in the Y axis, and most importantly, keeping the shape of the trajectory, which means, the new obtained trajectory has a similar shape that the demonstrations.

In Fig. 1 Case II a second adaptation case is shown using the same data set. Here, we present a comparison between two adapted trajectories, the orange one is adapted only in position and the blue one is adapted in position and velocity. The conditions of adapted position for both trajectories are the same, indicated by the purple markers. For the blue adapted trajectory, the velocity constraint is defined within the range $[-0.55, 0.55]$. The velocity for both trajectories is shown in Fig. 1c). The orange trajectory moves freely due to the lack of constraint, whereas the blue trajectory remains within the imposed velocity range defined in the inequality constraint. In Fig. 1b) the respective position trajectories are shown. Both trajectories satisfied the initial and final adaptation conditions and both keep the shape of the trajectory overall. However, the blue trajectory will be the one that produces smoother motions in the reproductions due to the velocity constraints.

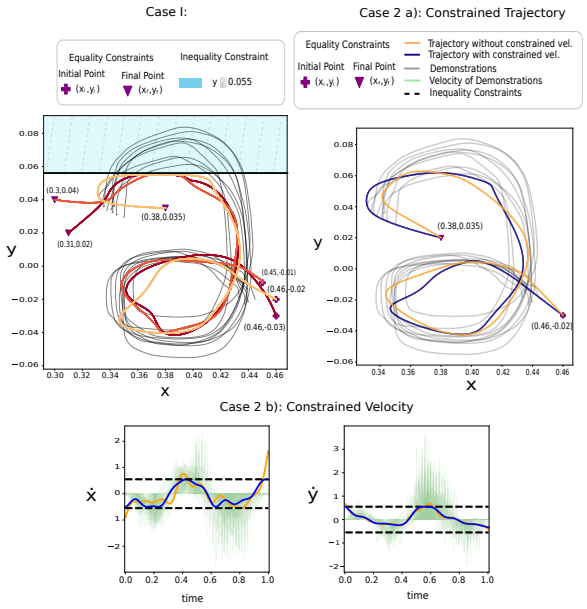


Fig. 1. Case I: Adaptation for new initial and final positions with a constraint in the Y axis. Case II: Adaptation for a new initial and final position with velocity constraints.

IV. Conclusions

The proposed approach addresses the adaptation problem for new conditions at both position and velocity level for a set of demonstrations. The approach is defined as a regression problem and handled as Constrained Quadratic Optimization, where the criteria to be minimized is defined by a sum-of-square errors of data points of the demonstrations, and the constraints represent the new adaptation conditions. The approach is validated in a set of trajectories taken from a Panda Robot. The adaptation involves new initial and final points as well as velocity constraints. The results show our proposed approach can fully satisfy the new imposed adaptation conditions while keeping the shape of the trajectory overall. The approach has important relevance 1) to scenarios with continuous changes that demand continuous adaptations of the trajectory, 2) to adaptations that require the shape of the trajectory to be preserved, and 3) to trajectories that demands high level of accuracy for the new adapted conditions. Our approach considers the following future work: I) Introduction of slack variables in the optimal solution vector, which are essential to relax the constraints and guarantee feasible solutions of the QP. II) Extend the adaptation to 3D trajectories and reproduce them in real scenarios. III) Comparison with similar methods e.g. CNMP, TP-GMM. IV) Explore some methods for better selection of the BF.

References

[1] M. T. Akbulut, E. Oztop, M. Y. Seker, H. Xue, A. E. Tekden, and E. Ugur, "Acnmp: Skill transfer and task extrapolation through learning from demonstration and reinforcement learning via representation sharing," in *Conference on Robot Learning*, 2020.

[2] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[3] M. Y. Seker, M. Imre, J. H. Piater, and E. Ugur, "Conditional neural movement primitives," in *Robotics: Science and Systems*, 2019.

Grasping the Inconspicuous

Hrishikesh Gupta*, Stefan Thalhammer*, Markus Leitner, Markus Vincze

***Abstract*— Transparent objects are common in day-to-day life and hence find many applications that require robot grasping. Many solutions toward object grasping exist for non-transparent objects. However, due to the unique visual properties of transparent objects, standard 3D sensors produce noisy or distorted measurements. Modern approaches tackle this problem by either refining the noisy depth measurements or using some intermediate representation of the depth. Towards this, we study deep learning 6D pose estimation from RGB images only for transparent object grasping. To train and test the suitability of RGB-based object pose estimation, we construct a dataset of RGB-only images with 6D pose annotations. The experiments demonstrate the effectiveness of RGB image space for grasping transparent objects.**

I. INTRODUCTION

Object detection and pose estimation are two of the most fundamental problems in the field of robot vision, crucial for robotic object grasping and manipulation. Although robot object manipulation by means of pose estimation itself is a quite challenging problem, it still offers a good and wide range of solutions for manipulating opaque objects. Towards this considerable research has been devoted to robotic manipulation of objects using 3D data (e.g. RGB-D images, point clouds) [21] [30]. However, many of these algorithms cannot be immediately applied to transparent objects [19]. This is because, existing commercial depth sensors, such as projected light or time-of-flight sensors, assume that objects have Lambertian surfaces that can support diffuse reflection from the sensor. Depth sensing fails when these conditions do not hold, e.g., for transparent or shiny metallic objects. Transparent objects are a common part of everyday life, from reading glasses to plastic bottles – yet these unique visual and material properties make them difficult for machines to perceive and manipulate, especially with the mentioned sensors.

Our main premise for the experimental setup in this manuscript is that RGB images provide enough information for object pose estimation for transparent objects. Towards this, we proposed an experimental setup using a canister as a transparent object. Which is a sterile medical object often used in the medical field for temporary storing and processing of fluids. Hence, has a strong use-case for robot object grasping.

For evaluation of our assumption, we propose an experimental setup: 1) First, collect an RGB dataset for our transparent object. 2) Annotate it using a state-of-art 6D pose estimation annotation tool [14]. 3) Use the state-of-art pose estimation method [23] for pose estimation for our transparent object. 4) Grasping of the pose-estimated transparent object, to evaluate our assumption and the effectiveness of the RGB-only image space. An overview of the grasping process is shown in Figure 1. We further qualitatively and quantitatively evaluate our experiments which prove the premise of our experimental setup.

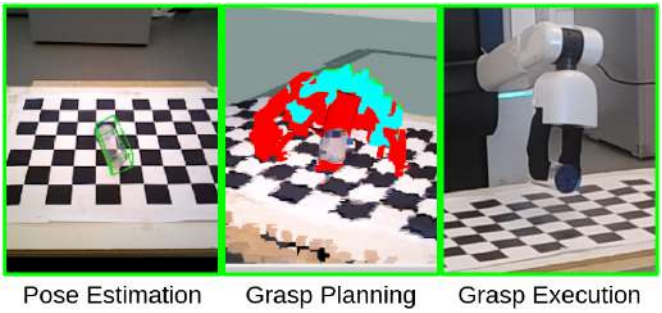


Fig. 1. **Overview** Our method leverages RGB images for transparent object grasping. We first perform pose estimation, grasp annotation, and planning, and as a final step execution of the grasp.

The following sections are organized as, section 3 describes our experimental setup. Where the pose estimation pipeline is discussed, along with the collection and annotation of the training data. In section 3 we also describe the grasping pipeline we use to evaluate our assumption for the usage of RGB only images. In Section 4 we present the experimental results of the transparent object grasping and discuss the results. Finally section 5 we talk in brief about our findings and possible future works.

II. RELATED WORK

Recent work tackling transparent object grasping and manipulation lies at the intersection of object detection, segmentation, geometric reasoning, depth reconstruction, and boundary detection. Applying these challenging problems to transparent objects received increased attention lately.

Classical methods mostly rely on peculiarities of such objects, such as specular reflections and local characteristics of edges due to refraction [15]. [4] used an additive model of latent representations to learn the appearance of transparent objects and remove the influence of background. These methods were made to perform localization of the objects and showed promising results in small experiments. Methods for transparent object segmentation started with the focus on formulating an energy function based on Light-Field linearity (LF-Linearity) [27] and occlusion detection from the

*Equal contribution, All authors are with the Automation and Control Institute (ACIN), TU Wien, 1040 Vienna, Austria {gupta, thalhammer, leitner, vincze}@acin.tuwien.ac.at

4D light-field image were optimized to generate the segmentation images. Recently, [25] introduce the Translab model for transparent object segmentation. They also introduced the first large-scale real-world transparent object segmentation dataset, termed Trans10K. It has 10K+ images. One of the most recent methods [7] combine polarization with deep learning and propose a polarized CNN for transparent object segmentation. Compared with previous methods, this still requires additional input data(Polarizing light-field) apart from the RGB only.

Methods such as TOM-Net [3] addresses the problem of transparent object matting. And formulating the problem as a refractive flow estimation problem. They propose a multi-scale encoder-decoder network to generate a coarse input, and then a residual network refining it to a detailed matte.

When it comes to transparent object pose estimation, initial methods either leverage failure modes of depth sensors like Microsoft Kinect[1] and estimate object pose with a known 3D shape model [24] [12], or also structured light sensors [17]. Because of the requirement for prior object models or specific sensors, those approaches do not allow a simple scenario where the only sensor is a stereo camera, or a commercial camera taking a couple of pictures of a scene "in the wild".

Many of the methods for transparent objects pose estimation rely on depth information. Hence many recent methods try to complete the missing depth information for transparent objects. The most recent and relevant methods closest to our work provide depth completion from an RGB image with inaccurate depth information [19] [31].

Transparent objects have been previously studied in various computer vision applications, including object pose estimation [9] [13] [12]. Works on estimating transparent object pose and geometry might assume knowing the object 3D model [12] [17]. In [13] [12], the pose of a rigid transparent object is estimated by 2D edge feature analysis. In [5], SIFT features are used to recognize the transparent object. However, low-level traditional features are not as discriminative as high-level deep features. The most recent method for Pose Estimation of the transparent objects introduced a keypoint-based feature [10] for pose estimation, trained on stereo images. But this requires manually choosing keypoints that should best describe the object pose with the addition of stereo images instead of RGB only.

All these methods heavily rely on additional input information besides RGB, for transparent object pose estimation. Hence requiring more complicated annotation processes such as in [10] and models. Hence we put forward our method requiring RGB only information for transparent object pose estimation and grasping. We show through our experiments, both qualitatively and quantitatively, that RGB images provide enough information for transparent object manipulation.

III. Experimental Setup

To demonstrate the potential of RGB for transparent object grasping, an experimental setup for pose estimation and manipulation of a transparent canister is created. The canister, shown in Figure 3 is a medical sterile object often used in the medical industry. In recent times due to growth in the automation of the medical sector, grasping such medical transparent objects canister is often one of the encountered hurdles in robotics.

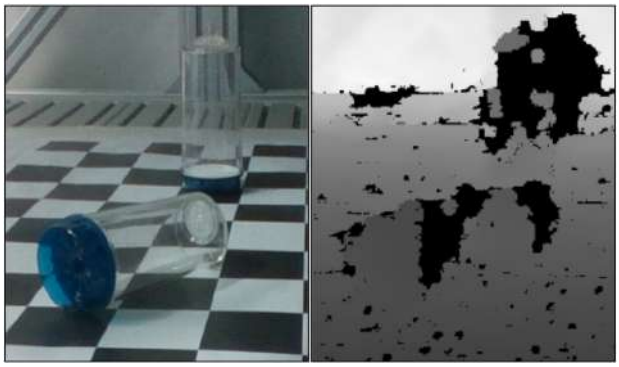


Fig. 2. **Point cloud of the Canister** Missing depth information of the transparent canister due to its non-lambertian nature.

Towards our experimental setup, firstly we collected a dataset of the transparent object and annotate it. A state-of-art pose estimation method [23] is trained on the collected dataset. The trained estimator is deployed in a robotic setup, to estimate the canister's pose on a tabletop. Based on this estimate the object is grasped.

A. Pose Estimation

Most robot grasping methods require object orientation and localization information in the scene. Hence, pose estimation is often a predecessor step for robot object grasping. Most transparent object pose estimation methods use some form of depth information. Since depth information for transparent objects produces degenerate solutions (see Figure 2) [19] [31], most of these methods add an extra step of either depth refinement [19] [31], extracting depth information of the background for refinement [26], or use some other form of the intermediate representation of the depth like disparity maps [10] for pose estimation.

Since the basic assumption of this work is that RGB images provide suitable information for transparent object grasping, we use a recent RGB-based object pose estimator [23]. The method only requires RGB images and an object model and does direct pose regression and detection in an end-to-end fashion.

[23] differs from most traditional methods dor pose estimation as it does not requires a preliminary detection stage and instead couple together the process of finding object classes and corresponding geometric correspondences, similar to [6]. Thus the method is also agnostic to the number of instances of the object in the scene.

The method takes an RGB image and a 3D model as input. The initial model is a multi-scale feature pyramid network, which takes input RGB image and generates as object hypothesis, object class, geometric correspondences and 6D pose

[1] https://en.wikipedia.org/wiki/Kinect

Fig. 3. **Dataset** Example samples from our training dataset showing the full variation of the provided backgrounds.

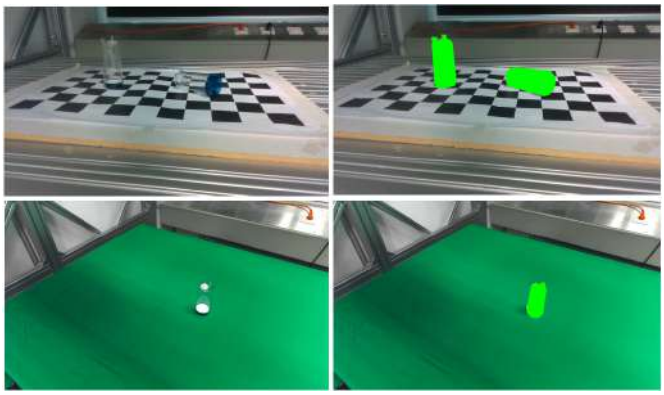


Fig. 4. **Dataset annotation examples** Example annotated images from the dataset. On the left column we see the original RGB images and on the right the corresponding annotations.

1) Training Data: For our data capturing process, we use the Realsense D435[2] and the ZED from stereolabs[3]. The camera is attached to the end-effector of the KUKA arm robot[4] and moved around the object in a sequence, where for each sequence around the object 104 images are taken uniformly sampled from various heights, angles, and distances around the object in a particular pose. Since the sequences are defined manually, the pose of the camera relative to the origin of the robot is known for each captured image.

For estimating the pose of the object i.e, relative to the camera attached to the robot end-effector we leverage multi-view geometry. Information of which is provided by multiple images taken in a sequence around the object, this is used by our pose annotation tool [14].

In total, we record 15 sequences each containing 104 images, of which 6 sequences were captured with only one transparent object instance and 9 with two instances of the object. This results in 1352 training images in total. Since we are dealing with transparent objects, it is even more vital for us to make our method robust to illuminations and backgrounds even more so as compared with non-transparent objects. Hence, we also use varying amounts of environment lighting and background patterns while data capturing. Particularly we use various dotted patterns, checkerboard patterns, metallic surfaces, etc for making our method robust to the background variation. For robustness to illumination, we make use of varying environment lighting and natural lights. For each sequence being captured, we vary object pose, instances, background, and illumination. Examples of our dataset can be seen in the Figure 3.

For 6D pose annotation of the object i.e, the transformation from the camera to the object in the given scene we use 3D-SAT [14]. Which is a state-of-art method for object pose annotation of RGB/RGBD sequences. One of the strongest relevance for our work is that the depth data is not necessary unlike other methods [14] and thus enabling the annotation of objects that are unsuitable for depth-based methods.

The annotation tool [14] requires a 3D model of the object being pose annotated along with the recorded sequence of images and camera intrinsics. After recording a sequence the available data is imported to the Blender annotation GUI. This enables the alignment of 3D object models with the imported RGB images to retrieve the 6D pose of objects. Pose annotation is done by aligning the 3D model to the object in each image of the sequence. Figure 4 shows a few examples of our 6D pose annotations, where the 3D model is aligned with the object. Multiple grasp poses were annotated by hand for our transparent object as shown in Figure 5.

2) Training: The weights of the backbone are pre-trained on ImageNet [18] and fine-tuned for 100 epochs using the Adam [8] optimizer with a learning rate of 1^{-5} and a batch size of 8. To benefit more from the pre-trained feature extractor we do not update the parameters of batch normalization and the convolution layers of the first two stages of the backbone during training.

Since the training dataset is quite limited in terms of pose variations we apply translation and zoom augmentation with up to 5% each. In order to not overfit to the training data, standard image augmentations such as brightness, contrast, blur and color changes are used. Similar to [16], [22], [20].

B. Grasping Pipeline

Performance of object detection and pose estimation methods often deteriorate when deployed on real-world robots [11] [1] [2]. Thus in order to evaluate our proposed method and experimental setup, we evaluated its performance in a grasping experiment using a Toyota HSR robot [28], [29].

Multiple grasp poses are annotated by hand for the object as shown in Figure 5. These annotated grasp poses are then transformed to the robot base frame using the estimated object pose of our method [23]. Based on the potential grasp poses, multiple trajectories are calculated and the first collision-free trajectory found is executed. A grasp is successful if the object is lifted and remains stable in the gripper.

[2]https://www.intelrealsense.com/depth-camera-d435/

[3]https://www.stereolabs.com/assets/datasheets/zed2-camera-datasheet.pdf

[4]https://www.kuka.com/de-at/produkte-leistungen/robotersysteme/industrieroboter/lbr-iiwa

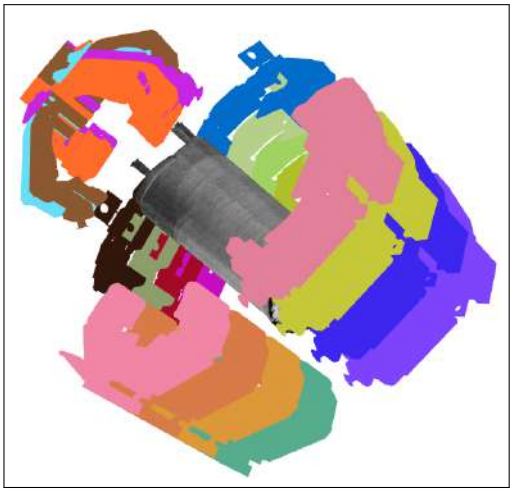


Fig. 5. **Grasp Annotation** 20 possible grasp configurations are annotated. The randomly colored grippers are scaled to 50% for visibility.

IV. Experiments

In this section, we present the results of our experimental setup. As mentioned previously the premise of our experimental setup is that the RGB images provide sufficient information for transparent object grasping. To evaluate the performance of our pose estimation based on our experimental setup, we evaluate it in a real-world robotic grasping experiment.

Our grasping setup contains a Toyota HSR robot [29] used for grasping our transparent canister. We place our canister on a wooden table, where the robot is looking at the canister at approximately and 45° angle. We place the canister on the table in three different ways: upright position, recumbent position, and attached to the base plate. We also use two different backgrounds, in-particular we use the checkerboard and the original wooden background of our table. As the first is already part of our training dataset and the other is to see how well our method generalizes to the unseen backgrounds.

A. Quantitative Results

In this section, we describe the results of our grasping experiments. We perform in total 5 grasps with randomized placement on the table, for the each of the four scenarios upright and recumbent with seen and unseen tabletop. Evaluation is based on three distinct cases the grasping attempt can result in:

- *Full Grasp*: The object is grasped and remains stable in the gripper
- *Reached Grasp*: A suitable grasp position is reached, but the grasp is unsuccessful due to the gripper moving the object previous to grasping.
- *Failed Grasp*: Neither the object is grasped nor a suitable grasping position is reached.

We assign a score of 1, 0.5 and 0 for *Full Grasp*, *Reached Grasp* and *Failed Grasp*, respectively. Reported scores in Table I are normalized by the number of grasp attempts.

As we see in Table I, the grasping experiments are slightly more successful in the case of the seen background as compared to the unseen background. Yet, the trained model generalizes to unseen object appearances. Showing RGB provides an informative modality for transparent object handling. The reason for the failed grasp attempts for upright case is mainly caused by the grasping method being agnostic to object geometry. The grasping attempts have significantly deteriorated for the case of the canister being in the recumbent position, for both the seen and unseen. Although the scores remained similar for both. The reason for the significant drop in grasping performance for recumbent cases is grasp-point sampling usually picking the grasp points that are protruding the table plane, hence leading to a collision of the gripper.

TABLE I

Grasping Experiments Canister grasping from a tabletop with known and unknown surface.

Tabletop	Seen		Unseen	
	upright	recumbent	upright	recumbent
Full Grasp	0.6	0.2	0.4	0.2
Reached Grasp	0.1	0.1	0.1	0.0
overall	0.7	0.3	0.5	0.2

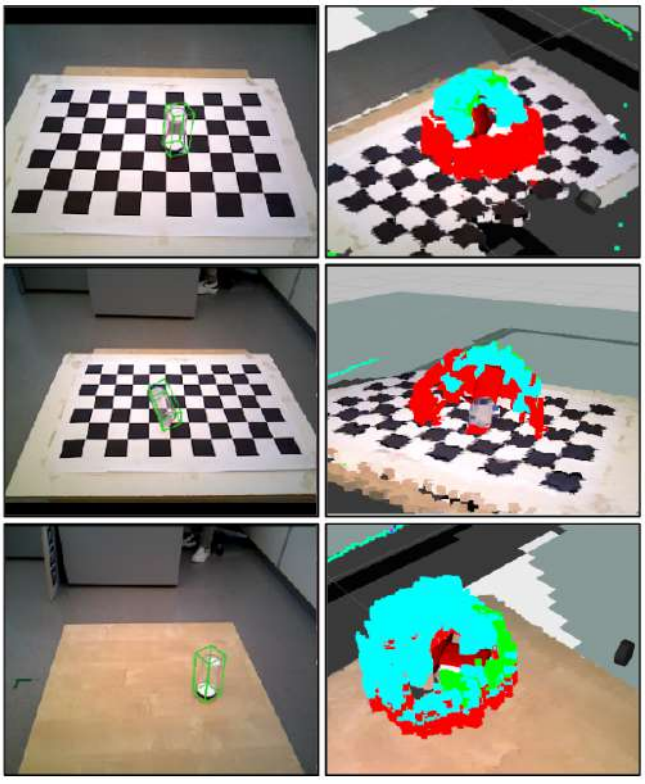


Fig. 6. **Pose Estimation and Grasp Point Sampling** The left column of images indicates estimated poses with a green bounding box. Right shows all grasps, grasps protruding the table (red), grasps not protruding the table plane (blue and green), and chosen grasp (green).

B. Qualitative Results

In this section, we introduce and describe the qualitative results of our grasping experiments. We first discuss the

visual results of the estimated poses and the choice of grasp points. Then we discuss the results of the grasping experiments, the cases where it succeeds, and the failure cases and give the reasoning behind them.

1) Pose Estimation and Grasp Sampling: The left column of Figure 6 shows the estimated 6D poses for our transparent object. We observe a small offset in the rotation of the estimated pose. The rotation is difficult to estimate since only the top part of the canister provides cues to disambiguate the rotational symmetry along the longitudinal axis of the object.

The right column of Figure 6 shows the possible grasp points around the pose estimated canister. The red points in the figure show the grasp points that are protruding the table plane, while the blue and green are the valid graspable points not protruding the table's surface. The top row shows that for the upright position of the canister it is easy to plan executable grasp trajectories. While the middle row of Figure 6 shows the graspable points for the canister in the recumbent position.

In the case of the canister being in a recumbent position (Figure 6 middle row), we have seen the performance drop in grasping successes, Table I. This is mainly because of the main axis of the canister lying along the table surface, resulting in the grasp point being chosen close to the table surface (red grasp points). This leads to the collision of the gripper with the surface, leading to failed grasps. Poses are slightly worse for the scenario featuring tabletops unseen during training time, bottom row of Figure 6.

2) Grasping the Canister: The top two rows of Figure 6 show some of the examples of successful grasps. We also introduced a distractor, i.e the base plate of the canister in our grasping experiments as seen in Figure 7, which is not part of our dataset for training the pose estimator. The trained model generalizes well, with successful grasps to cases with added distractor (canister base-plate) and unseen tabletops.

3) Failure Cases: The last row of Figure 7 shows one of the failure cases for the canister in the upright position. As mentioned above, grasping sometimes fails since grasp planning does not account for the object geometry. Significant improvements can already be achieved by improving grasp point sampling and grasp trajectory planning. Additionally, providing a richer and more diverse dataset, in terms of table top texture and pose variations will improve pose estimation and thus the grasping success.

V. SUMMARY AND OUTLOOK

In this work, we conduct a study to evaluate the effectiveness of the RGB-only image space for transparent object pose estimation for robot object grasping. The experiments and the results prove our assumption about the usability of RGB We conduct several successful grasps on a transparent object, even in an completely unseen setting, saying scene and background. Future work will investigate and provide improvements by training the proposed setup on a much larger scale of data including more instances and variations in the scene. As well as improvements for grasp sampling, grasp filtering and grasp planning.

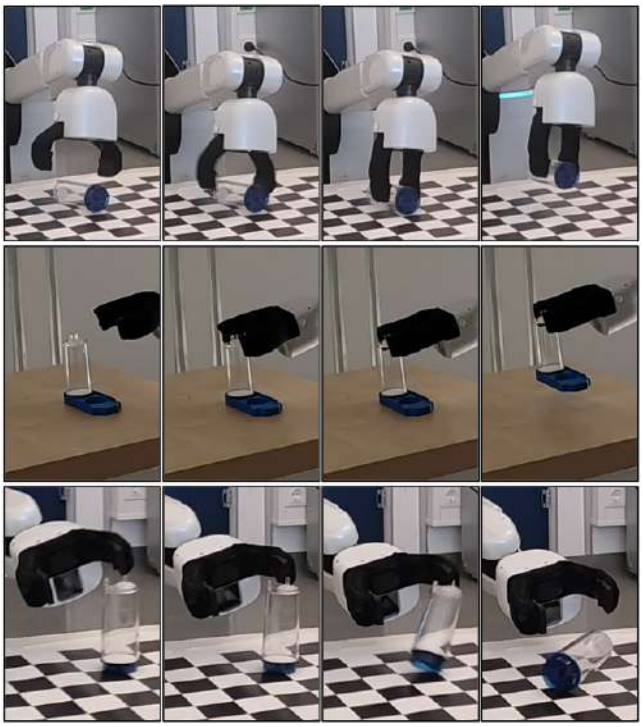


Fig. 7. **Grasping Sequence with a Distractor** The Canister placed in its base plate, as such unseen during training, is picked from the table.

VI. ACKNOWLEDGEMENT

We gratefully acknowledge the support of the EU-program EC Horizon 2020 for Research and Innovation under grant agreement No. 101017089, project TraceBot, the support by the Austrian Research Promotion Agency (FFG) under grant agreement No. 879878, project K4R and the NVIDIA Corporation for the donation of the GPU used for this research. Furthermore we would like to thank Bernhard Neuberger for supporting this work with his knowledge and expertise.

REFERENCES

[1] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1378–1385.

[2] D. Bauer, T. Patten, and M. Vincze, "Verefine: Integrating object pose verification with physics-guided iterative refinement," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4289–4296, 2020.

[3] G. Chen, K. Han, and K.-Y. K. Wong, "Tom-net: Learning transparent object matting from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9233–9241.

[4] M. Fritz, G. Bradski, S. Karayev, T. Darrell, and M. Black, "An additive latent feature model for transparent object recognition," *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[5] C. Guo-Hua, W. Jun-Yi, and Z. Ai-Jun, "Transparent object detection and location based on rgb-d camera," in *Journal of Physics: Conference Series*, vol. 1183, no. 1. IOP Publishing, 2019, p. 012011.

[6] T. Hodan, D. Barath, and J. Matas, "Epos: Estimating 6d pose of objects with symmetries," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 703–11 712.

[7] A. Kalra, V. Taamazyan, S. K. Rao, K. Venkataraman, R. Raskar, and A. Kadambi, "Deep polarization cues for transparent object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8602–8611.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[9] U. Klank, D. Carton, and M. Beetz, "Transparent object detection and reconstruction on a mobile platform," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 5971–5978.

[10] X. Liu, R. Jonschkowski, A. Angelova, and K. Konolige, "Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 602–11 610.

[11] M. R. Loghmani, B. Caputo, and M. Vincze, "Recognizing objects in-the-wild: Where do we stand?" in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2170–2177.

[12] I. Lysenkov, V. Eruhimov, and G. Bradski, "Recognition and pose estimation of rigid transparent objects with a kinect sensor," *Robotics*, vol. 273, no. 273-280, p. 2, 2013.

[13] I. Lysenkov and V. Rabaud, "Pose estimation of rigid transparent objects in transparent clutter," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 162–169.

[14] T. P. Markus Suchi, Bernhard Neuberger and M. Vincze, "3d-sadt: Simple annotation dataset toolkit for robotic vision," in *Submitted to IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[15] K. McHenry, J. Ponce, and D. Forsyth, "Finding glass," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 973–979.

[16] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7668–7677.

[17] C. J. Phillips, M. Lecce, and K. Daniilidis, "Seeing glassware: from edge detection to pose estimation and shape recovery." in *Robotics: Science and Systems*, vol. 3, 2016, p. 3.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[19] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3634–3642.

[20] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," in *Proceedings of the european conference on computer vision*, 2018, pp. 699–715.

[21] A. Ten Pas and R. Platt, "Using geometry to detect grasp poses in 3d point clouds," in *Robotics Research*. Springer, 2018, pp. 307–324.

[22] S. Thalhammer, M. Leitner, T. Patten, and M. Vincze, "Pyrapose: Feature pyramids for fast and accurate object pose estimation under domain shift," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 909–13 915.

[23] S. Thalhammer and M. Vincze, "Cope: End-to-end trainable constant runtime object pose estimation," in *submission to the European Conference on Computer Vision*, 2022.

[24] T. Wang, X. He, and N. Barnes, "Glass object localization by joint inference of boundary and depth," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 2012, pp. 3783–3786.

[25] E. Xie, W. Wang, W. Wang, M. Ding, C. Shen, and P. Luo, "Segmenting transparent objects in the wild," in *European conference on computer vision*. Springer, 2020, pp. 696–711.

[26] C. Xu, J. Chen, M. Yao, J. Zhou, L. Zhang, and Y. Liu, "6dof pose estimation of transparent object from a single rgb-d image," *Sensors*, vol. 20, no. 23, p. 6790, 2020.

[27] Y. Xu, H. Nagahara, A. Shimada, and R.-i. Taniguchi, "Transcut: Transparent object segmentation from a light-field image," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3442–3450.

[28] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of the research platform of a domestic mobile manipulator utilized for international competition and field test," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 7675–7682.

[29] T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of human support robot as the research platform of a domestic mobile manipulator," *ROBOMECH journal*, vol. 6, no. 1, pp. 1–15, 2019.

[30] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3750–3757.

[31] L. Zhu, A. Mousavian, Y. Xiang, H. Mazhar, J. van Eenbergen, S. Debnath, and D. Fox, "Rgb-d local implicit function for depth completion of transparent objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4649–4658.

Gaussian Process Regression for Inverse Kinematics

Lucas Muster[1], Mohamed Aburaia[2] and Wilfried Wöber[3]

***Abstract*— For the calculation of the inverse kinematics in robotics, approaches such as the analytic or numerical method are used, which provide unambiguous and robust results, but have a disadvantage such as high calculation and modelling effort. Due to recent successes in the field of computer hardware, new techniques are emerging. These technologies overcome several limitations of the aforementioned methods for the calculation of inverse kinematics for robotics. In this context, this study introduces Gaussian process regression for estimating the inverse kinematics for various robots. The results show that a regression accuracy of over 99% for each axis has been achieved for five given robot kinematics.**

I. INTRODUCTION

A robotic system and its pose of the corresponding tool center point (TCP) is represented through the angular position of the individual axes [3]. For the counterpart of that, namely the calculation of the inverse kinematics, the position and orientation of the TCP is given, represented by $\mathbf{x}$. From this pose, the axes $\mathbf{q}$ of the robotic system are then calculated as shown in equation 1, where f is a nonlinear function due to nonlinear relationships between the existing joints [8].

$$\mathbf{q} = f^{-1}(\mathbf{x}), \text{ where } \mathbf{q} = \begin{bmatrix}\theta_1 & \theta_2 \dots & \theta_n\end{bmatrix}^T \tag{1}$$

The computation of robot kinematics represented in equation 1 is challenging and requires the kinematic structure, the robots configuration and limitations [1]. Furthermore, due to the number of axes, a linear mapping of the kinematic structure is not possible [9].

To overcome the aforementioned problems different methods are proposed, which can be divided into traditional methods, such as algebraic, geometric and numerical (iterative), as well as novel methods such as data-driven and hybrid methods [3], [9]. Recently, data-driven approaches, are emerging in the field of robotics, which use data and flexible models. These models and their parameters are optimized using pose examples [3].

One of the most well-known probabilistic models in the field of machine learning is the Gaussian process regression (GPR) [5], [14]. GPR is a non-parametric regression model, therefore it is not bound to a specific function and uses the Bayesian approach to derives a probability distribution over all possible values [5]. Relying on the theoretical benefits of Gaussian processes this study tackles the calculation of a GPR model, trained on real data points of the given robot to obtain a inverse kinematic model.

This study is organized as follows: Chapter 2 describes the state of the art for machine learning-based inverse kinematic estimation. Chapter 3 presents the methods used before discussing the experimental results in Chapter 4. Subsequently, the results are presented in Chapter 5.

II. STATE OF THE ART

Machine learning is a part of data-driven methods and an alternative way to calculate and control a robot [2], [12], [15]. Methods such as neural networks [5], random forest [6], support vector machine [10] and k nearest neighbor regression [4] are used to learn the corresponding model and to subsequently predict new data points [3]. The goal of these models is the prediction of the individual axis angle positions, which leads to the positioning of the TCP depending on the kinematic model. GPR is widely used for nonlinear computations and, as described above, offers advantages over other machine learning methods [5], [14]. An approach on how to use GPR in robotics, is shown in [16], using a GPR to compute the inverse kinematics of a seven degree of freedom robotic arm. Each observation of the data consists of a total of twenty-one inputs, the seven axis positions, the seven axis velocities, and the seven axis accelerations. This type of regression also lends itself to humanoid robots, as [7] shows. An extended version of the GPR is the Gaussian Process Latent Variable Model (GPLVM), used for kinematics with a high number of robot axis [11].

III. METHODS

For the calculation of the GPR, a test dataset X and the training dataset X_* are measured on the physical robot, which consists of different position values of the end effector. Additionally, the corresponding angular values of the joints are stored in the design matrix Y. For data processing, random positions of the robots were generated. These data matrices are the basis for training. To compute this, a prior is defined, which is determined by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x},\mathbf{x}')$, shown in equation 2 [14].

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}')) \tag{2}$$

It is assumed that the observed data points differ from the function value with some noise ε, based on a Gaussian distribution (see equation 3).

$$y = f(x) + \varepsilon \text{ with } \varepsilon \sim \mathcal{N}(0, \sigma_n^2) \tag{3}$$

[1]Lucas Muster is with the Department of Industrial Engineering, University of Applied Sciences Technikum Wien, 1200 Vienna, Austria muster@technikum-wien.at

[2]Mohamed Aburaia is with the Department of Industrial Engineering, University of Applied Sciences Technikum Wien, 1200 Vienna, Austria aburaia@technikum-wien.at

[3]Wilfried Wöber is with the Department of Industrial Engineering, University of Applied Sciences Technikum Wien, 1200 Vienna, Institute for Integrative Nature Conservation Research, University of Natural Resources and Life Sciences, Vienna Austria woeber@technikum-wien.at

From this we can declare the joint distribution of the observed data points and the function values related to the prior as shown in the following equation 4 [5], [14].

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix} \right) \quad (4)$$

Note the noise factor σ_n^2, which is incorporated into the covariance as shown in equation 4. By conditioning the joint Gaussian prior distribution on the observations, the predictions of the mean function $\bar{\mathbf{f}}_*$ and the corresponding covariance matrix $cov(\mathbf{f}_*)$ can be determined [5], [14].

IV. Implementation

The inverse kinematics calculation with the GPR was implemented for five different robot kinematics, which are shown in figure 1, using the sklearn library [13]. Due to the different kinematic properties, it can be tested how well the regression can be determined with the GPR. A separate GPR was trained and evaluated for each of these kinematic models. The optimization of the hyperparameters is computed by maximizing the log-marginal likelihood.

V. Results

The results show the GPR performance in the table I and II. The regression resulting in a small mean squared error (MSE) and a high R^2 score for each data point. Every robot kinematic regression has an R^2 score over 99% for each axis.

VI. Summary and Outlook

This work has dealt with machine learning in the field of robotics. The results show, that the GPR can be used for calculating the inverse kinematics for different types of robot models. Future research could deal with the implementation of a GPLVM, whereby a dimension reduction is performed beforehand to possibly achieve better results [11].

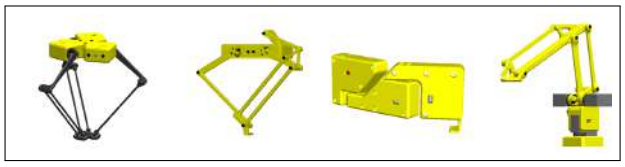


Fig. 1. Representation of the robot kinematics, for the calculation of the inverse kinematics with GPR. From left to right: Delta robot, 2D robot, scara robot (RRP) and palletizing robot. The kinematics of the scara robot (RRR) differs from the kinematics of the Scara robot (RRP) only by a rotatory, instead of the last translatory axis.

TABLE I

Results of the R^2 Score with Gaussian Process Regression.

R^2			
Robot	Axis 1	Axis 2	Axis 3
Delta robot	99.99%	99.99%	99.99%
Scara robot (RRR)	99.99%	99.99%	-
Scara robot (RRP)	99.97%	99.08%	99.99%
Palletizing robot	99.99%	99.99%	99.99%
2D robot	99.99%	99.99%	-

TABLE II

Results of the MSE with Gaussian Process Regression.

MSE			
Robot	Axis 1	Axis 2	Axis 3
Delta robot	2.05e-06	2.79e-06	2.34e-06
Scara robot (RRR)	2.21e-08	2.09e-05	-
Scara robot (RRP)	3.44e-04	9.47e-03	2.15e-10
Palletizing robot	9.38e-07	5.67e-06	2.58e-06
2D robot	3.89e-07	1.01e-06	-

References

[1] L. Aggarwal, K. Aggarwal, and R. J. Urbanic, "Use of artificial neural networks for the development of an inverse kinematic solution and visual identification of singularity zone(s)," *Procedia CIRP*, vol. 17, pp. 812–817, 2014.

[2] A. R. Almusawi, L. C. Dülger, and S. Kapucu, "A New Artificial Neural Network Approach in Solving Inverse Kinematics of Robotic Arm (Denso VP6242)," *Computational Intelligence and Neuroscience*, vol. 2016, 2016.

[3] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse Kinematics Techniques in Computer Graphics: A Survey," *Computer Graphics Forum*, vol. 37, no. 6, pp. 35–58, 2017.

[4] M. Arunadevi, D. J. Praveen, M. A. Ruchik, P. Kishore Gandhi, K. V. Siddu, and S. Vishnu Prasad, "Machine learning approach for inverse kinematics in trajectory planning of pioneer 2 manipulator with cubic spline interpolation," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 807–813, 2021.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[7] D. Forte, A. Ude, and A. Kos, "Robot learning by gaussian process regression," in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*, 2010, pp. 303–308.

[8] A. T. Hasan, N. Ismail, A. Hamouda, I. Aris, M. Marhaban, and H. Al-Assadi, "Artificial neural network-based kinematics jacobian solution for serial manipulator passing through singular configurations," *Advances in Engineering Software*, vol. 41, no. 2, pp. 359–367, 2010.

[9] P. Jha and B. B. Biswal, "A Neural Network Approach for Inverse Kinematic of a SCARA Manipulator," *IAES International Journal of Robotics and Automation (IJRA)*, vol. 3, no. 1, pp. 52–61, 2014.

[10] R. Khanna and M. Awad, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, 04 2015.

[11] N. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Advances in Neural Information Processing Systems*, S. Thrun, L. Saul, and B. Schölkopf, Eds., vol. 16. MIT Press, 2004.

[12] N. A. Mohamed, A. T. Azar, N. E. Abbas, M. A. Ezzeldin, and H. H. Ammar, "Experimental kinematic modeling of 6-dof serial manipulator using hybrid deep learning," in *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)*. Cham: Springer International Publishing, 2020, pp. 283–295.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] C. E. Rasmussen, *Gaussian Processes in Machine Learning*, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds. Berlin, Heidelberg: Springer, 2004.

[15] M. M. Sanjeev, M. J. Thomas, T. K. S. Kumar, A. P. Sudheer, and M. L. Joy, "Determination of inverse kinematic solutions for a 3 degree of freedom parallel manipulator using machine learning," in *2020 IEEE Students Conference on Engineering Systems (SCES)*, 2020, pp. 1–6.

[16] D.-Y. Yeung and Y. Zhang, "Learning inverse dynamics by gaussian process regression under the multi-task learning framework," *The Path to Autonomous Robots, G.S. Sukhatme (ed.)*, pp. 1–12, 10 2009.

Towards Deep-Learning-Based Local Features for Visual SLAM Algorithms: A Comparison

Stefan Dimnik[1,2], Matthias Schörghuber[1], and Marco Wallner[1]

Abstract— **Reliable localization and mapping is the basis for a wide range of applications in autonomous robot operation, self-driving cars, augmented reality, and many more areas. Using imaging sensors, sparse feature-based visual simultaneous localization and mapping (VSLAM) approaches are widely used. The selection of keypoints and the invariance of their descriptors to changes in viewpoint and lighting is crucial for a robust long-term operation. Most modern VSLAM systems rely on classic feature extractors and descriptors such as ORB or SIFT. In contrast, ML-based methods such as SuperPoint, HF-Net, or R2D2 dominate the benchmarks in the domain of visual place recognition already. In this paper, we extend openVSLAM to compare different keypoint detection and description methods (ORB, HF-Net, SuperPoint) and evaluate typical localization and run-time metrics on the OpenLoris and EuRoC dataset. Experimental results show comparable results in terms of localization accuracy but smaller and possibly more stable maps for the machine-learning-based approaches that can be directly used for visual place recognition tasks. Shifting the feature calculation to the GPU also releases resources on the CPU for other tasks.**

I. INTRODUCTION

Resilient long-term localization in various environments is a key enabling technology for a variety of applications. E.g., in the domain of autonomous acting mobile robots, actions and operations can be performed only with a decent knowledge of its 6-DoF pose in some reference coordinate frame ([1], [2], [3]). LiDAR-, or in general, depth-sensor-based approaches can provide a reasonable long-time performance [4] as long as the geometric properties of the environment do not change too much, i.e., walls and dominant objects are static. Using such sensors is often not possible in many applications for various reasons, e.g., costs or the mechanical setup. Enabling true long-term operation with camera-based, visual SLAM systems is still an open issue as the environment's appearance changes drastically over time (e.g., different lighting conditions, the course of the seasons). Most recent feature-based VSLAM systems rely on classic features like ORB or SIFT and perform well in short-term benchmarks only. The research field of visual place recognition (VPR) addresses these problems and has already brought up solutions ([5], [6]) that can handle significant variations in appearance.

*The research leading to these results has received funding by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) in the frame of the FFG "ICT of the Future" program as part of the "openSCHEMA" project (grant no. 887533).

[1] S. Dimnik, M. Schörghuber, and M. Wallner are with the AIT Austrian Institute of Technology GmbH, Vienna, Austria {firstname.surname}@ait.ac.at.

[2]S. Dimnik is with the University of Applied Sciences Wiener Neustadt stefan.dimnik@fhwn.ac.at

In this work, we evaluate how the achievements in the VPR domain can be used for resilient visual SLAM. Our main contributions are (i) the extension of the openVSLAM framework to work with a variety of different feature types and (ii) an extensive evaluation of localization- and run-time-metrics on the OpenLoris and EuRoC dataset.

II. RELATED WORK

Local feature detection and matching are fundamental problems in computer vision with a long history. We differentiate between classic and machine learning-based methods. Most classic approaches consist of two stages: First, keypoints (distinct and salient image regions) are extracted. Second, the surrounding image region of the keypoint is summarized by a descriptor. Prominent and vastly used classic methods are SIFT [7], ORB [8], and AKAZE [9]. They differ in the type of detector, complexity and consequently, the matching quality and computational requirements. ORB is a prominent choice for real-time applications like VSLAM [10]. More recently, machine learning is used to find relevant feature regions and extract more expressive and stable local descriptors. A recent survey about local features is presented in [11]. An overview of the evolution of local features to deep-features and how deep-features are trained is given in [12]. Research on deep-features is very active, especially in the domain of visual place recognition (VPR) resp. visual localization [13], [14]. Well performing methods are SuperPoint [15], R2D2 [16], and HF-Net [17].

SuperPoint pre-trains on synthetic data and uses a self-supervised approach to extract local features in a single forward pass. HF-Net is a hierarchical two-stage visual localization approach. As a first step, coarse localization is performed using image-retrieval with a global image-level descriptor followed by local feature matching. They use the SuperPoint architecture for local features but jointly train local and global features. This joint-training method can further improve loop closing capabilities of VSLAM algorithms.

Bojanic et al. [18] compares the performance of several classic and deep local feature methods. They found that deep models do not outperform classical models in general but provide better invariance for specific cases (e.g., large variation in illumination). In addition, the SuperPoint model was shown to have faster extraction times than ORB when using a GPU, which argues for its use in real-time applications such as VSLAM.

VSLAM has been a prominent research topic for several decades too. For a brief overview, we refer to the survey

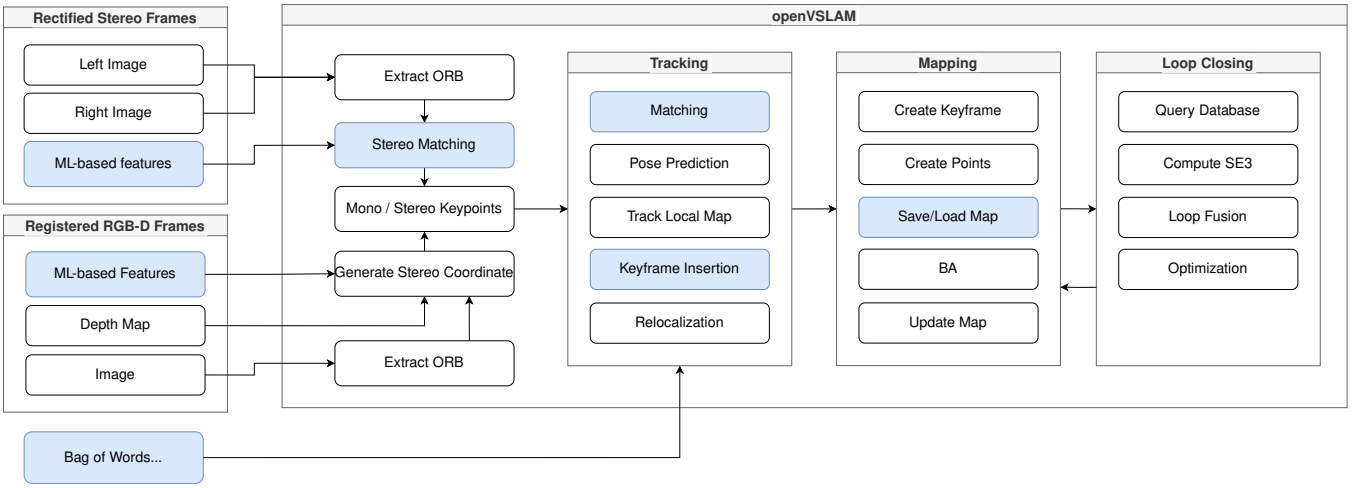


Fig. 1: Overview of the used evaluation setup. Changes to the openVSLAM framework are marked in blue. The framework was extended to allow the input of pre-computed features and descriptors for RGBD and stereo images as input. Retrained FBoW vocabularies are used for each descriptor type. The stereo matching was adapted to support floating-point valued descriptors too. Changes in matching and keyframe insertion strategies were needed as described.

paper of Cadena et al. [19]. Recent VSLAM systems can be classified into feature-based [20], [10], [21], and photometric or direct approaches [22], [23]. Feature-based approaches use local feature methods to find correspondences between images. Direct approaches avoid the costly extraction of features by directly finding correspondences based on pixel intensities. Recently, advances were made by integrating machine learning techniques into VSLAM algorithms [24], [25], [26]. For a more complete overview, we refer to surveys on VSLAM in general [27], and specific for feature-based approaches in [28].

This work focuses on using deep local features for VSLAM applications. In [29], ORB-SLAM [30] was modified to support SuperPoint and GCNv2 [31] as local features. They evaluated the performance on the KITTI [32] outdoor dataset. SuperPoint failed on several sequences. GCNv2, trained on an indoor dataset, could not generalize to the autonomous driving outdoor setting of KITTI. Similarly, DXSLAM uses ORB-SLAM2 with HF-Net as feature backbone. DXSLAM focuses in their evaluation on the visual-localization part of the system by using the OpenLoris [33] dataset. We use a similar VSLAM approach with SuperPoint and HF-Net as local features in this work. In contrast to DXSLAM, we make a more detailed evaluation of the effects of using deep local features and evaluate with the established VSLAM stereo benchmark dataset EuRoC [34].

III. Implementation

We start with DXSLAM as our machine-learning-based VSLAM baseline as it uses HF-Net to extract local features and builds upon ORB-SLAM2. To evaluate the impact of different deep-learning-based local features, we added modular feature extractors for HF-Net and SuperPoint and extended the interface of openVSLAM to accept pre-computed keypoints and descriptors. The choice of openVSLAM as the basis for the evaluation is motivated by the comparison in [35] and as it is a modern, widely used, modular VSLAM framework supporting monocular, stereo, and RGBD data as input. The applied changes are depicted in Fig. 1. As openVSLAM supports binary ORB features only, the matching had to be modified to support the floating-point-based descriptors of the machine-learning-based approaches ($L1$ distance as measure). Similarly, the FBoW vocabulary was retrained with SuperPoint and HF-Net features using every 20^{th} frame from the OpenLoris and TUM RGBD dataset [36]. The insertion criterion for new keyframes is finetuned for ORB characteristics in the original openVSLAM. Using SuperPoint and HF-Net, fewer but more salient keypoints are extracted. Therefore we added an additional insertion criterion to insert a new keyframe and adapted the thresholds accordingly. openVSLAM distinguishes "reliable" and "tracked" landmarks, the first being observed by at least 3 keyframes. With this information, we add new keyframes if the number of tracked becomes 20% more than the number of reliable landmarks. This empirical motivated additional insertion criterion reduces the tracking losses significantly (especially in areas with low textures and fast movements).

IV. Evaluation

In this section, we present the results of our experiments with different feature extraction models and compare them to the baseline implementation of openVSLAM using ORB. In a fist step, we compare our system to the existing machine-learning enabled VSLAM implementation DXSLAM. As there is only an evaluation of DXSLAM for the OpenLoris dataset available, we perform the first experiments comparing DXSLAM and our extended version of openVSLAM on this RGBD dataset in TABLE I. To gain more insights into the possible improvements introduced by the machine-learning-based features on longer and more complex sequences, we continue the evaluation on the EuRoC stereo dataset in TABLE II. The evaluations were performed on a workstation

(a) Office 1 (b) Office 3 (c) Office 5

Fig. 2: Sample images from the OpenLoris office dataset.

(a) Machine Hall (b) Vicon Room 1 (c) Vicon Room 2

Fig. 3: Sample images from the EuRoC dataset.

(Intel® Xeon® E5-1620v3, GeForce® GTX 1080Ti) running Ubuntu 20.04. Therefore, the time measurements are not to be taken as absolute values but can represent a trend or a comparison between the methods.

A. Evaluation Metrics

For a quantitative evaluation of our results, we compare the estimated trajectory with a ground truth trajectory using the absolute pose error *APE* and relative pose error *RPE* metric as proposed by Sturm et al. [36].

The estimated trajectory $\boldsymbol{P} = \{\boldsymbol{P}_1, \ldots, \boldsymbol{P}_n\}$, $\boldsymbol{P}_i \in \mathrm{SE}(3)$ and the ground truth trajectory $\boldsymbol{Q} = \{\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_n\}$, $\boldsymbol{Q}_i \in \mathrm{SE}(3)$ are aligned to each other by finding a transformation $\boldsymbol{S} \in \mathrm{SE}(3)$ between both trajectories using the Umeyama method [37]. For each correspondence between ground truth and estimation the error $\boldsymbol{F}$ is defined as

$$\boldsymbol{F}_i = \boldsymbol{Q}_i^{-1} \boldsymbol{S} \boldsymbol{P}_i. \tag{1}$$

For the $\mathrm{APE}_{\mathrm{RMSE}}$ metric value on the full trajectory, we compute the root mean squared error (RMSE) of the translational component of all error values

$$\mathrm{APE}_{\mathrm{RMSE}} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left\| \mathrm{trans}(\boldsymbol{F}_i) \right\|^2}. \tag{2}$$

The relative pose error (RPE) estimates the local performance, i.e., drift of the algorithm. The relative motion between frame pairs from the estimated trajectory is compared with the relative motion from the ground truth trajectory as

$$\boldsymbol{E}_k^{\Delta} = (\boldsymbol{Q}_i^{-1} \boldsymbol{Q}_{i+\Delta})^{-1} (\boldsymbol{P}_i^{-1} \boldsymbol{P}_{i+\Delta}). \tag{3}$$

For our experiments we choose the smallest $\Delta \in \mathbb{N}$ s.t. the traveled path, i.e., the sum of euclidean distances d, between two ground-truth poses $\sum_{j=1}^{\Delta} d\left(\boldsymbol{Q}_i, \boldsymbol{Q}_{i+j}\right) \geq 1\mathrm{m}$. For the *RPE* metric value on the full trajectory, we compute the RMSE of the $K = \left|\boldsymbol{E}^{\Delta}\right|$ translational components of all error values

$$\mathrm{RPE}_{\mathrm{RMSE}} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \left\| \mathrm{trans}(\boldsymbol{E}_k^{\Delta}) \right\|^2}. \tag{4}$$

Considering the translational parts is sufficient to evaluate the overall performance as errors in rotation lead directly to deviations in translation.

B. OpenLoris RGBD Office Dataset

As we have integrated deep features into openVSLAM similar to DXSLAM, we first compare our implementation to the results of DXSLAM. DXSLAM provides published evaluation results for the OpenLoris dataset only. Therefore, the evaluation on the OpenLoris dataset was performed to allow a direct comparison with the existing system. Examples from this dataset can be seen in Fig. 2.

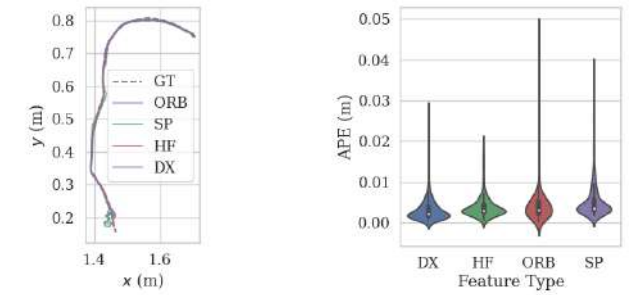
TABLE I: Impact of different keyframe insertion strategies on the APE for the OpenLoris dataset: DXSLAM (i), openVSLAM (ii), our optimized and OpenLoris overfitted version (iii), and our generalized implementation described in section III (iv).

	$\mathrm{APE}_{\mathrm{RMSE}}$ /*mm*					
Sequence	DX (i)	ORB (ii)	HF (iii)	SP (iii)	HF (iv)	SP (iv)
office1	79.3	76.8	62.5	**57.7**	85.0	81.3
office2	**59.9**	64.7	64.2	61.5	66.6	65.6
office3	**4.3**	7.0	5.7	5.1	4.7	5.4
office4	**55.8**	60.5	73.0	57.3	65.5	60.6
office5	162.0	135.5	154.1	**104.4**	159.8	108.8
office6	71.3	76.8	61.3	**58.1**	84.2	84.6
office7	87.8	**85.8**	91.6	93.2	89.2	87.5

TABLE I shows the results on the OpenLoris office dataset for the APE only as the sequences are too short (about 1m) to get further meaningful insights using RPE. Our experiments showed that the condition for inserting new keyframes is crucial for the overall performance and avoiding tracking loss. The conditions to insert a new keyframe are optimized for ORB in openVSLAM. HF-Net and SuperPoint have a significant other behavior in the extraction pattern - specially in hardly textured areas, very few keypoints are extracted. This leads to tracking loss as the existing conditions do not cover these cases. In a first step, we ported the conditions used in DXSLAM for keyframe insertion for our openVSLAM setup with machine-learning-based features. With this setup, we achieve similar performance to DXSLAM on the OpenLoris dataset but suboptimal performance on the EuRoC data. This leads us to conclude, that the insertion strategy used by DXSLAM is highly (over-)fitted to the OpenLoris data. We have implemented our own condition set that works for both OpenLoris and EuRoC. We show the results of the different keyframe insertion strategies in TABLE I. Due to the more general version, we perform worse for OpenLoris than DXSLAM, but we can see from the results that the APE metric is strongly affected by overfitting. Selected trajectories and their APE distributions (plots are generated using the evo framework [38]) are shown in Fig. 4. No significant difference stands out in the distributions, but some advantage of the machine-learning based approaches and especially SuperPoint is noticeable.

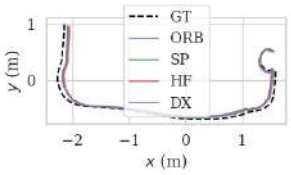
C. EuRoC Stereo Dataset

We choose the well-known EuRoC VSLAM dataset for a more sound assessment and further evaluation. The EuRoC

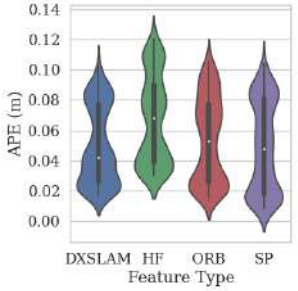


(a) Trajectory office3

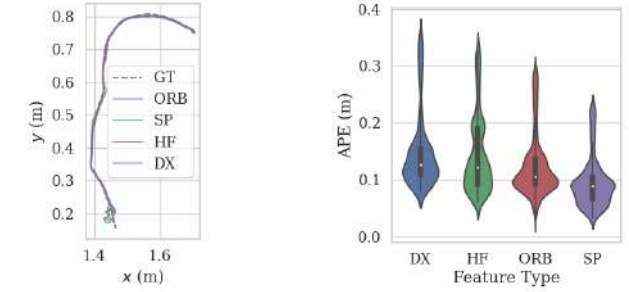
(b) APE office3



(c) Trajectory office4



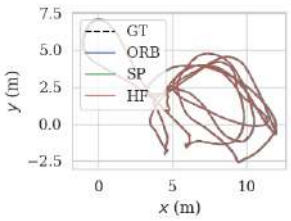
(d) APE office4



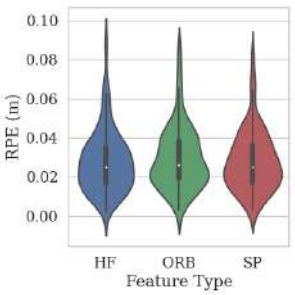
(e) Trajectory office3

(f) APE office5

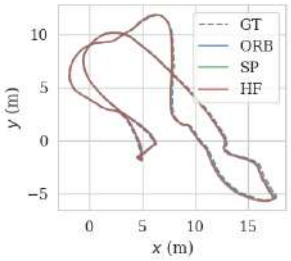
Fig. 4: Selected OpenLoris evaluation results. Trajectory and APE for DXSLAM (DX), openVSLAM with ORB (ORB), openVSLAM with HF-Net (HF), and openVSLAM with SuperPoint (SP). HF and SP with generalized keyframe insertion strategy, i.e., referencing to (iv) in TABLE I.



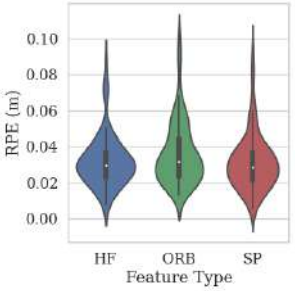
(a) Trajectory mh_03



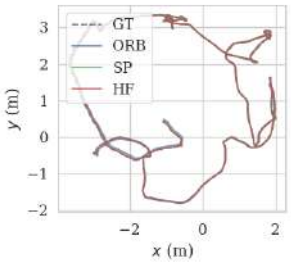
(b) RPE mh_03



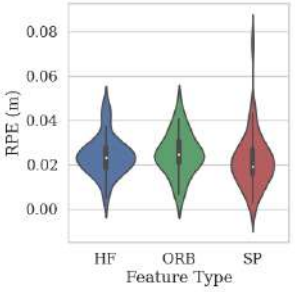
(c) Trajectory mh_04



(d) RPE mh_04



(e) Trajectory v2_01



(f) RPE v2_01

Fig. 5: Selected EuRoC evaluation results for machine hall sequence 3 (medium), 4 (hard), and vicon room 2, sequence 1 (easy). Trajectory and RPE distribution as violin plot for ORB, HF-Net, and SuperPoint based VSLAM.

dataset contains 11 sequences of a UAV operating in a machine hall (mh) and two laboratories (vicon room) (v1,v2) with different degrees of difficulty, tagged as easy (e), medium (m) to hard (h), as annotated in TABLE II. Samples of the dataset can be seen in Fig. 3. Medium and hard sequences contain rapid motion, fast rotations, motion blur, and severe brightness changes.

Overall, there is no statistically significant difference in APE and RPE over all EuRoC sequences identifiable. Especially for the RPE, the distribution of the error is quite similar, see Fig. 5 for the mh_03, mh_04 and v2_01 sequences. This might be explained by the fact, that even EuRoC scenes are not long enough to introduce sufficient changes in the appearance of the environment (e.g., seasonal changes, lighting) to play off the descriptive strengths of HF-Net and SuperPoint over ORB. This coequal local and global tracking performance comes with a significant reduction in map size, as can be seen in the number of landmarks used in TABLE II. This leads directly to a lower tracking and mapping time needed for the machine-learning-based solutions. The $\bar{t}_{\text{tracking}}$ column shows the time to track landmarks without extraction time. The tracking using HF-Net and SuperPoint is faster or equal to ORB, although the matching of the binary descriptor is faster than the float 256 descriptors of the machine-learning-based features.

V. SUMMARY AND OUTLOOK

In this paper, we evaluated the possible improvements using machine-learning-based local features like SuperPoint and HF-Net for visual SLAM applications. We showed similar performance in APE and RPE but significantly smaller maps with fewer landmarks. Using SuperPoint or HF-Net as local features, the created VSLAM maps may be

TABLE II: Results on the EuRoC sequences. **Best** and second best results are marked bold and underlined respectively. Tracking- and mapping-times are given as the mean and variance over all frames of the sequence. No valid results for v2_03 (h) are given as the tracking is lost for all setups.

	APE_{RMSE} /mm			RPE_{RMSE} /mm/m			#Keyframes /1			#Landmarks $/10^3$			$\bar{t}_{tracking} \pm \sigma$ /ms			$\bar{t}_{mapping} \pm \sigma$ /ms		
Sequence	ORB	HF	SP	ORB	HF	SP	ORB	HF	SP	ORB	HF	SP	ORB	HF	SP	ORB	HF	SP
mh_01 (e)	**27.3**	35.3	32.7	20.8	**20.7**	21.0	275	276	**265**	12.50	10.92	**10.19**	**75**±27	86±28	78±25	720±611	539±322	**505**±310
mh_02 (e)	26.8	23.5	**21.6**	18.3	**14.7**	14.8	**240**	255	**240**	10.29	10.14	**9.19**	**67**±25	85±28	79±27	616±462	444±289	**409**±289
mh_03 (m)	37.1	32.7	**28.5**	33.6	**32.2**	**32.2**	**236**	276	267	10.06	10.48	**9.09**	**74**±26	91±30	83±29	627±399	**504**±294	506±322
mh_04 (h)	**109.6**	200.8	184.4	39.8	**35.0**	35.3	276	264	**251**	13.19	10.90	**9.20**	74±27	84±30	**37**±17	439±241	334±146	**202**±119
mh_05 (h)	56.5	59.3	**51.0**	28.9	**28.1**	28.3	303	282	**247**	13.61	11.04	**9.20**	**72**±26	86±31	77±30	476±259	400±209	**339**±193
v1_01 (e)	**36.1**	39.7	42.6	91.9	**89.7**	91.1	**204**	281	238	12.25	12.04	**9.87**	92±29	92±30	**78**±36	944±601	572±433	**505**±401
v1_02 (m)	**18.6**	42.8	22.9	**43.8**	44.5	46.4	**182**	352	320	10.47	14.33	**10.24**	73±31	70±33	**58**±36	532±333	256±167	**240**±198
v1_03 (h)	**36.7**	163.1	58.3	**58.9**	61.2	61.2	**246**	510	345	12.22	17.92	**7.86**	66±31	68±35	**29**±21	329±212	199±116	**99**±68
v2_01 (e)	36.6	22.1	**17.2**	26.4	**25.5**	25.7	**257**	302	284	14.66	10.22	**8.68**	85±32	79±31	**66**±30	531±246	320±159	**274**±148
v2_02 (m)	**26.3**	84.5	92.0	37.4	37.9	**36.1**	**261**	522	500	**12.54**	14.80	12.79	82±29	70±29	**60**±30	513±357	228±131	**202**±145
v2_03 (h)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

used directly for VPR tasks. In a next step, the matching and validation process could be improved using machine learning. SuperGlue [39], a neural network that matches two sets of local features by jointly finding correspondences and rejecting non-matching points could be one option for SuperPoint. Further computational resources are transferred to the GPU, allowing for more complex CPU tasks.

REFERENCES

[1] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," *Intelligent Industrial Systems 2015 1:4*, vol. 1, no. 4, pp. 289–311, 2015.

[2] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajnik, "Artificial Intelligence for Long-Term Robot Autonomy: A Survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4023–4030, 2018.

[3] M. Wallner, C. Mühlbacher, G. Steinbauer, S. Haas, T. Ulz, and J. Ludwiger, "A robust and flexible software architecture for autonomous robots in the context of industrie 4.0," in *Joint Workshop on Vision, Automation & Robotics, OAGM/AAPR ARW*, 2017.

[4] G. Kim, B. Park, and A. Kim, "1-Day Learning, 1-Year Localization: Long-Term LiDAR Localization Using Scan Context Image," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1948–1955, 2019.

[5] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[6] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, vol. 113, p. 107760, 2021.

[7] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[9] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281–1298, 2011.

[10] 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.

[11] J. Ma, X. Jiang, A. Fan, J. Jiang, and J. Yan, "Image matching from handcrafted to deep features: A survey," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 23–79, 2021.

[12] G. Csurka, C. R. Dance, and M. Humenberger, "From handcrafted to deep local features," *arXiv preprint arXiv:1807.10254*, 2018.

[13] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, and Others, "Benchmarking 6dof outdoor visual localization in changing conditions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[14] N. V. Keetha, M. Milford, and S. Garg, "A Hierarchical Dual Model of Environment- And Place-Specific Utility for Visual Place Recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6969–6976, 2021.

[15] D. Detone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 337–349, 2018.

[16] J. Revaud, P. Weinzaepfel, C. de Souza, and M. Humenberger, "R2D2: Repeatable and reliable detector and descriptor," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[17] P. E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 12 708–12 717, 2019.

[18] D. Bojanic, K. Bartol, T. Pribanic, T. Petkovic, Y. D. Donoso, and J. S. Mas, "On the comparison of classic and deep keypoint detector and descriptor methods," *International Symposium on Image and Signal Processing and Analysis, ISPA*, vol. 2019-September, pp. 64–69, 2019.

[19] 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.

[20] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*. IEEE, 2007, pp. 1–10.

[21] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, and G. Le Besnerais, "OV^{2}SLAM : A fully online and versatile visual SLAM for real-time applications," *IEEE Robotics and Automation Letters*, 2021.

[22] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and MachineI Intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[23] 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.

[24] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 1281–1292.

[25] H. Zhan, C. S. Weerasekera, J.-W. Bian, and I. Reid, "Visual Odometry Revisited: What Should Be Learnt?" in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4203–4210.

[26] S.-H. K. Ue-Hwan Kim and J.-H. Kim, "Simvodis: Simultaneous visual odometry, object detection, and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[27] M. Servières, V. Renaudin, A. Dupuis, and N. Antigny, "Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking," *Journal of Sensors*, vol. 2021, p. e2054828, 2021.
[28] R. Azzam, T. Taha, S. Huang, and Y. Zweiri, "Feature-based visual simultaneous localization and mapping: A survey," *SN Applied Sciences*, vol. 2, no. 2, pp. 1–24, 2020.
[29] C. Deng, K. Qiu, R. Xiong, and C. Zhou, "Comparative study of deep learning based features in slam," in *2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. IEEE, 2019, pp. 250–254.
[30] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *CoRR*, vol. abs/1610.06475, 2016.
[31] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, "GCNv2: Efficient Correspondence Prediction for Real-Time SLAM," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3505–3510, 2019.
[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, 2013, zSCC: 0003034.
[33] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, Y. Guo, Z. Wang, Y. Zhang, B. Qin, W. Yang, F. Wang, R. H. M. Chan, and Q. She, "Are we ready for service robots? the OpenLORIS-Scene datasets for lifelong SLAM," in *2020 International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3139–3145.
[34] 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*, 2016.
[35] A. Merzlyakov and S. Macenski, "A comparison of modern general-purpose visual SLAM approaches," *CoRR*, vol. abs/2107.07589, 2021.
[36] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.
[37] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991, zSCC: 0001980 Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
[38] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.
[39] P. E. Sarlin, D. Detone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching with Graph Neural Networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4937–4946, 2020.

3D-ToF vision-based detection and dynamic work space visualization for an ABB GoFa robot

Daniel Schillhammer[1], Clemens Ambros[2], Ali Aburaia[2] and Mohamed Aburaia[2]

***Abstract*— This paper presents a safeguarding and visualization approach for human protection to further enhance flexible and efficient industrial processes. By combining a customary projector with a depth camera, the robot's work space can be visualized and any intrusion can be detected reliably. Kinematic parameters of a robotic manipulator are extracted and combined into 3-dimensional danger and warning zones. A 2D-projection of these zones is visualized dynamically on work surface level. The measures have been integrated within a collaborative industrial assembly process featuring a lightweight robot to enhance visibility and hazard perception. An intrusion detection was realized using safety-certified equipment, which leads to a high potential for safety certification of the application.**

I. Introduction

Starting from 2015, an annual average growth rate in industrial robot installations resulted in an operational stock over 3 million units in 2020 [1]. Robots are able to reduce the workload of employees and enable complex automation steps with increased productivity. As new technologies emerge, traditional robot-based automation processes and corresponding safeguarding increasingly do not meet emerging market requirements. As a subcategory to Human-Robot-Interaction (HRI), Human-Robot-Collaboration (HRC) combines human advantages with the efficiency and precision of a machine. Suitable lightweight robots such as models from ABB [2], Universal Robots [3] and KUKA [4] are already widely available. These robots are able to be used in the "highest" collaboration mode: power and force limiting (PFL) [5]. While the maximum force and torque values depend on the specific application, most collaborative processes are somewhat limited in their productivity due to lower payloads and speed. Furthermore, most applications require additional external protective devices to further reduce the residual risk, even if lightweight robots are used. This often leads to another collaboration mode - speed and separation monitoring (SSM) - which generally allows the robot to move with higher speeds and payloads and therefore to be more efficient. In the context of future flexible production paradigms, new electro-sensitive protective equipment is being developed and increasingly used to secure robotic systems and to tackle productivity issues.

Currently, used protection methods in collaborative robotics do not provide visualization of the robot's work space. Therefore, awareness of potential hazards caused by the manipulator is limited for human workers. This can lead to unfavorable or even hazardous scenarios, resulting in unwanted collisions. Primary safety measures alone are not sufficient to further enhance overarching production flexibility within industrial production facilities and human-robot workplaces. This paper presents a novel and safety-certifiable approach to safeguard an industrial collaborative assembly process by developing a concept for intrusion detection using a 3D depth camera and a supportive visualization of the robot's work space.

[1]tofmotion GmbH, Am Europlatz 2, Building G, 1120 Vienna, Austria
daniel.schillhammer@tofmotion.com
[2]Faculty of Industrial Engineering, University of Applied Sciences Technikum Wien, Höchstädtplatz 6, Vienna, Austria
<firstname>.<lastname>@technikum-wien.at

II. Related work

Industrial robot applications must provide certain safety functions with a corresponding Performance Level [6]. The required Performance Level is regulated by the standard EN ISO 13849-1:2015 and provides information about the reliability of the safety system used [7]. According to EN ISO 12100:2010, protective devices can be divided into separating and non-separating [8]. One advantage of non-separating protective devices is that they can be parameterized and individually reconfigured. To guarantee a safe work environment for humans without separating devices (e.g. fences), all objects must be detected within the safeguarding space. The concept is non-separating approach.

Because of the visualization aspect of this paper, optical systems are discussed in more detail. Vogel et al. already described an approach for a safe human-robot-collaboration based on a projector camera system [9]. In this paper, an image is projected on the floor by a projector and an acquired pattern is evaluated by two cameras. In the further development of this approach, a robot is integrated [10] [11]. In this case, the robot provides information about the joint positions and the trajectory. Based on these mechanisms, a dynamic visualization of the robot's working area can be realized and an intrusion in the safeguarding zone can be detected by using image comparison methods. Hietanen et al. presented a similar approach where the system consists of a projector, a consumer depth camera (Kinect V2) and a Universal Robot UR10 [12]. The projector is used for visualization of the dynamically calculated zones. The Kinect camera provides a point cloud which allows the position of the human in relation to the robot to be determined. If the human is detected in a danger zone, the system triggers a stop of the robot via a ROS interface. The algorithm for calculating the zones is based on the joint positions of the robot. Within our approach and in order to implement safe monitoring of dynamic working zones of robots, which can be useful in real-

world industrial production scenarios, safety-certified sensor systems are required. There are only two safety-certified systems currently commercially available, ”spotguard” [13] and FreeMove system [14]. Both systems are based on the Time-of-Flight principle. The concept presented here is a non-separative approach.

III. Materials and methods

The project is implemented within the scope of a human-robot-collaboration station at the UAS Technikum Wien. The robot GoFa (internally CRB-15000) from ABB is used as a manipulator capable of collaborative tasks. The parameters of the danger and warning zones are extracted and displayed dynamically. The project is divided into two parts in order to establish a clear separation between visualization and intrusion into any zone. The synchronization of both parts is realized by a PC running Ubuntu 20.04 and ROS Noetic.

A. ABB Externally Guided Motion (EGM)

The ABB robot uses a controller called Omnicore. For the calculation of the danger and warning zones, information about the position of the robot in the work space is required. This information must be obtained from the robot controller itself. Since there is no standard interface for transmitting the joint data, an extended interface from ABB is used. With the help of the EGM module, the joint data can be transmitted to a PC via Ethernet every 4ms. This module is programmed in the ABB controller and on the PC side. The data is published cyclically on the PC by the ”Joint State Publisher” of ROS.

B. Projector

Due to environmental conditions and for better luminosity, a laser-based projector (Optoma ZH406ST Full-HD) is used. In this case, a high lumen value of 4,200 ANSI lumens is necessary to ensure good visibility. Because of the low projection distance, the short distance method of the projector is used. In order to visualize the entire working area of the robot including the protective zones, the image size must be taken into account for the mounting height. With the selected projector, an image width of 3.4m is achieved and an image height of 1.9m with a projection distance of 1.7m is set.

C. Safety camera

To detect any intrusion into a danger or warning zone, the 3D-Time-of-Flight camera ”spotguard” from tofmotion including the corresponding SDK is used in this setup. The sensor has a detection capability of 50mm at a maximum distance of 4.5m. The camera is suitable for safety functions up to PL d Kat. 2 (EN ISO 13849-1:2015) and SIL2 HFT1 (EN 62061:2016) and thus would be able to satisfy safety requirements for this specific application. In order to repeatedly upload new parameters for the danger and warning zones, the tofmotion SDK is adapted and therefore no longer compliant with safety standards. The system is configured by using the ROS Framework on an Ubuntu OS.

D. Communication

A standard PC running Ubuntu 20.04 LTS is used for communication between robot controller, projector and camera. To communicate with the controller and to access joint data of the robot during motion a UDP connection via Ethernet is established and the library abb_libegm is installed. Additionally, a ROS C++ wrapper is written to publish robot joint data to the framework. The representation of the convex hull curve of the zones is displayed through a HDMI connection to the projector. The connection between PC and camera uses UDP for image transfer and TCP for writing or reading camera specific data. By using the supplied tfm library, a 3D point cloud of the current real scene is displayed in RViz. In figure 1 is shown an overview of the system.

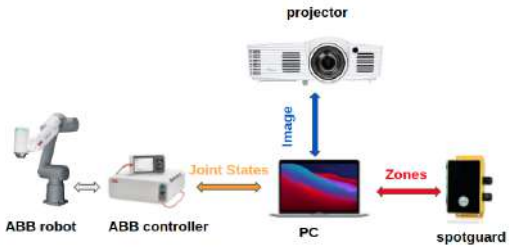


Fig. 1. Schematic overview of entire system

E. Coordinate transformation

The system consists of the projector, robot (with all axes) and the camera which refer to a coordinate system in the base of the robot and each component is calibrated to it. The projector provides a 2D image with the corresponding shapes. This 2D image can be transferred to the world coordinate system through a transformation matrix. The information from the CAD model of the workplace is used to determine the transformation matrix.

Each component of the system is calibrated to the reference coordinate system in the base of the robot. The 2D image of the projector is transferred to the world coordinate system through a transformation matrix. A CAD model of the HRC station is used to determine the transformation matrix. The reference plane is located at the base link of the robot.

The translation vector and the rotation matrix are determined by a camera calibration. Classic methods using checkerboard patterns are only possible to a limited extent, because of the low image resolution. The camera is using low power ToF technology and thus has lower measurement accuracy. Therefore, a calibration method using reflective points with high distance accuracy has been chosen. In this approach, three retro-reflective spherical markers are mounted on the end effector of the robot and recorded. By detecting the three retro-reflective markers within the point cloud and measuring the Cartesian positions P_1, P_2 and P_3 in the camera coordinate system, a transformation matrix between marker and camera coordinate system for a selected pose of the robot is obtained.

$$T_C^M = \begin{pmatrix} R & \vec{t} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \vec{x_M} & \vec{y_M} & \vec{z_M} & \vec{t_{C,M}} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

By positioning and defining the calibration markers and the marker coordinate system K_M, the columns of the transformation matrix can be calculated [15]. All vectors shown are normalised.

$$\vec{x}_M = \frac{\vec{p}_3 - \vec{p}_1}{|\vec{p}_3 - \vec{p}_1|} \quad (2)$$

$$\vec{y}_M = \frac{\vec{p}_2 - \vec{p}_1}{|\vec{p}_2 - \vec{p}_1|} \quad (3)$$

The vector of the z-axis is calculated by the cross product of the vectors $\vec{x}_M$ and $\vec{y}_M$. The z-axis is defined as positive upwards.

$$\vec{z}_M = \vec{x}_M \times \vec{y}_M \quad (4)$$

The translation vector is based on the vector $\vec{p}_1$ and determines the transformation matrix from the camera to the marker.

$$\vec{t}_{C,M} = \vec{p}_1 \quad (5)$$

The next step is to look for the transformation from the marker to the world coordinate system. This can be solved by a known transformation matrix of the marker to the TCP and the forward kinematics of the robot. To solve the transformation matrix between robot and marker, the current joint states values are read from the robot controller and solved by using of the forward kinematics. This results in the desired transformation from the world coordinate system to the marker.

$$\vec{\Theta} = (\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6)^T \quad (6)$$

$$T_W^M = T_W^6(\vec{\Theta}) \cdot T_6^M \quad (7)$$

The required transformation for a given robot pose from the camera to the world coordinate system can be determined by the following equation.

$$T_W^C = T_M^C \cdot T_W^M = T_M^C \cdot T_M^{W-1} \quad (8)$$

F. Convex hull visualization

The approach to create the convex hull differs for the projector and the camera, as the projector provides a 2D image and the ToF camera provides a depth image and/or point cloud. Information of the robot's axis positions is needed to configure and visualize the safety zones. This information is transmitted cyclically every 4ms by the robot controller via the EGM server. The data is made available to the ROS framework by using a ROS C++ wrapper as a rostopic *joint_states* with the help of the *joint_state_publisher*. The information of the current joint position of each joint (1-6) is needed to calculate the convex hull. Two convex hulls are calculated and displayed, a danger zone and a warning zone. A zone represents a convex hull of the current robot pose. The algorithm for creating the convex hulls includes the following steps:

1 Get the current joint position from joint 1-6
2 Calculate a circle in each joint with a given radius using point sampling (eq. 9)
3 Transform all circles to a reference plane (base_link)
4 Run convex hull algorithm of Graham et al. (1983) [16] which calculates the danger zone as well as the warning zone from circles

$$points\_in\_circum = \cos(\frac{2\pi}{4}) \cdot r + \sin(\frac{2\pi}{4}) \cdot r \quad (9)$$

The calculated hull curves are coloured according to their zone type (danger zone: red, warning zone: yellow) and projected into the world coordinate system (base_link). The image of the convex hull of the current robot position is cyclically updated and transmitted to the projector. This projects both zones on the plane of the HRC work space. A black background image is used for better contrast and visibility.

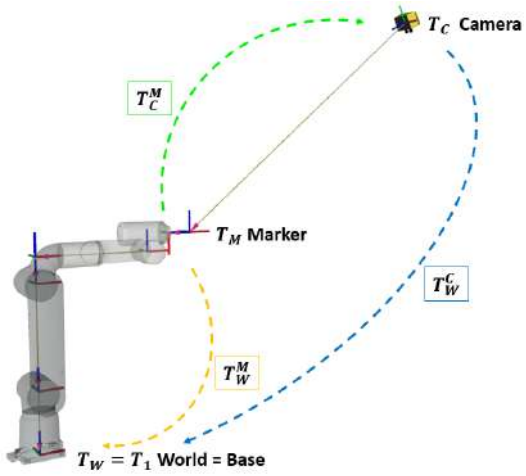


Fig. 2. Schematic representation of the relation of the coordinate systems

G. Safety zones

Within the given setup, the movement of the manipulator (hazardous machine function) can be defined by three spheres with their centers in the joints one, three and five of the robot. The SDK can be used to define these spheres as danger and warning zones with differing radii, and has been modified to upload the zones continuously. The detection of any violation of the zones triggers an OSSD signal to stop the robot in category 1 (danger zone) or 2 (warning zone) according to EN 60204-1:2019. The manipulator can therefore be safeguarded based on the three spheres. As the robot is located within the zones and thus would lead to a triggering of the safety function, additional spherical blanking zones according to EN IEC 62046:2018 are defined.

Figure 3 shows an exemplary robot pose with all zones simulated in RViz which are uploaded to the camera.

Fig. 3. Blanking, danger and warning zones in joints one, three and five of the ABB GoFa robot. The combination of all zones results in a protective shell around the robot.

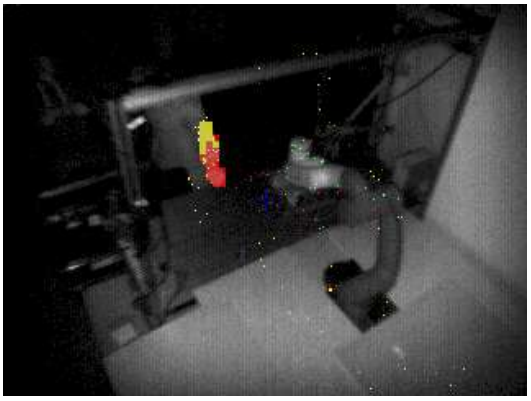


Fig. 4. Human arm falls below PSD and violates danger zone which leads to a safe stop of the robot

IV. Conclusion

A dynamic projection of the hull curve of a work space of the robot ABB GoFa has been realized. The calculation of the envelope curve is based on the current angle values of the robot control via the EGM interface of the robot controller. The projection visualizes the envelope curve on a 2D plane in the reference coordinate system of the robot's base. The projection of the hull curve is not used for any safety function, but only to support the visual perception of hazards for humans.

The violation of a warning or danger zone by an object is realized using a safety-certified 3D ToF camera. An extrinsic camera calibration using retro-reflexive markers has been conducted. The camera software was modified accordingly in order to continuously upload zone configurations. In this implementation, humans can efficiently share the work space with robots. This means that if the robot is in a safe pose or moving along a path that does not interfere with the human's task, the worker can proceed with the assembly application. If the protective separation distance (PSD) is violated, a safe signal is triggered by the camera leading to a safe stop of the manipulator (Figure 4). The safety-certified algorithm for detecting a violation has not been changed, thus no revalidation of the detection probability is necessary. The safe integration of the OSSDs of the spotguard was not part of this work, as a safety PLC and a Profisafe interface are required here. At the time of this work, no safety interface was available for the robot.

A. Outlook

By using the projector to visualize additional elements such as virtual buttons within the work space, new process input methods can be realized. To detect an activation of virtual buttons, an addition zone can be defined within the camera software and linked with a specific robot reaction through the controller. Additionally, live process data can be projected onto the work surface to further augment the assembly process.

To establish a safe communication between the signals of the camera and the robot, a safety PLC has to be implemented. A Profisafe communication could be used to read safe camera signals and transmit them to the Omnicore controller. To realize a safety certification, a safety standard-compliant reading of the robot joint data is necessary. In this work, a non-secure protocol was used as the manufacturer does not offer a secure protocol for providing the joint data.

References

[1] International Federation of Robotics (IFR). Executive Summary World Robotics 2021 Industrial Robots.

[2] ABB Asea Brown Boveri Ltd. GoFa™ CRB 15000, 2022, https://new.abb.com/products/robotics/collaborative-robots/crb-15000.

[3] Universal Robots GmbH. Collaborative Robots from Universal Robots, 2022, https://www.universal-robots.com/products/.

[4] KUKA AG. Industrial robots from KUKA, 2022, https://www.kuka.com/en-at/products/robotics-systems/industrial-robots.

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

[6] *Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots*, EN ISO 10218-1:2011.

[7] *Safety of machinery - Safety-related parts of control systems - Part 1: General principles for design*, EN ISO 13849-1:2015.

[8] *Safety of machinery - General principles for design - Risk assessment and risk reduction*, EN ISO 12100:2010.

[9] Christian Vogel, Maik Poggendorf, Christoph Walter, and Norbert Elkmann. *Towards safe physical human-robot collaboration: A projection-based safety system*. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3355–3360, 2011.

[10] Christian Vogel, Christoph Walter, and Norbert Elkmann. *Safeguarding and Supporting Future Human-robot Cooperative Manufacturing Processes by a Projection- and Camera-based Technology*. *Procedia Manufacturing*, 11:39–46, 2017. 27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy.

[11] Christian Vogel, Christoph Walter, and Norbert Elkmann. *A projection-based sensor system for safe physical human-robot collaboration*. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5359–5364, 2013.

[12] Antti Hietanen, Roni-Jussi Halme, Jyrki Latokartano, Roel Pieters, Minna Lanz, and J. Kämäräinen. *Depth-sensor – projector safety model for human-robot collaboration*. 2018.

[13] tofmotion GmbH. spotguard® 2022, https://www.tofmotion.com/products.

[14] Veo Robotics, Inc. Veo FreeMove, 2022, https://www.veobot.com/.

[15] Christina Ramer. *Arbeitsraumüberwachung und autonome Bahnplanung für ein sicheres und flexibles Roboter-Assistenzsystem in der Fertigung*. Doctoral thesis, FAU University Press, 2018.

[16] Ronald L Graham and F Frances Yao. *Finding the convex hull of a simple polygon*. *Journal of Algorithms*, 4(4):324–331, 1983.

Continuous Target-free Extrinsic Calibration of a Multi-Sensor System from a Sequence of Static Viewpoints

Philipp Glira[1], Christoph Weidinger[1], Johann Weichselbaum[1]

***Abstract*— Mobile robotic applications need precise information about the geometric position of the individual sensors on the platform. This information is given by the extrinsic calibration parameters which define how the sensor is rotated and translated with respect to a fixed reference coordinate system. Erroneous calibration parameters have a negative impact on typical robotic estimation tasks, e.g. SLAM. In this work we propose a new method for a continuous estimation of the calibration parameters during operation of the robot. The parameter estimation is based on the matching of point clouds which are acquired by the sensors from multiple static viewpoints. Consequently, our method does not need any special calibration targets and is applicable to any sensor whose measurements can be converted to point clouds. We demonstrate the suitability of our method by calibrating a multi-sensor system composed by 2 lidar sensors, 3 cameras, and an imaging radar sensor.**

I. INTRODUCTION

Robots are typically equipped with several sensors to continuously observe their surroundings. For this purpose, various sensor modalities are used due to the specific strengths and weaknesses of each modality. The most commonly used sensors for environment perception are cameras, lidar sensors, radar sensors, ultrasound sensors, and infrared sensor. The data of these sensors is combined by means of *sensor fusion* to get a more complete, accurate, and reliable description of the environment. However, in order to properly fuse the different sensor data streams, they need to be correctly aligned. A misalignment exists if *systematic* discrepancies between the data of different sensors are observed. One way to minimize such discrepancies is a proper *extrinsic calibration*[1] of the sensors (Fig. 1).

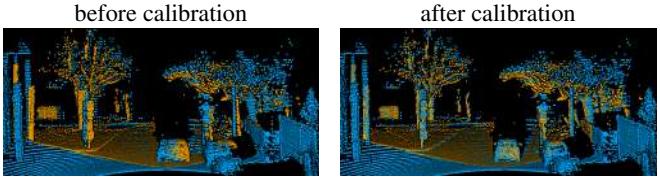


Fig. 1. Effect of an extrinsic sensor calibration on the relative orientation of two lidar point clouds.

*The research leading to these results has received funding from the Mobility of the Future programme. Mobility of the Future is a research, technology and innovation funding programme of the Republic of Austria, Ministry of Climate Action. The Austrian Research Promotion Agency (FFG) has been authorised for the programme management.

[1]AIT Austrian Institute of Technology, AAS Assistive and Autonomous Systems, 1210 Vienna, Austria philipp.glira@ait.ac.at, christoph.weidinger@ait.ac.at, johann.weichselbaum@ait.ac.at

[1]sometimes also denoted as *mounting calibration*

The extrinsic calibration defines the rotational and translational offset w.r.t. to a fixed reference coordinate system. It has 6 degrees of freedom (DoF) and is typically defined by 3 Euler angles[2] (α_x, α_y, α_z) and 3 translation components (t_x, t_y, t_z) [15].

We propose in this work a new online target-free calibration method. The main advantages in comparison to previous works are that:

- the calibration is widely sensor agnostic, as it is applicable to any sensor whose measurements can be converted to point clouds
- the calibration parameter estimates are continuously improved during the operation of the robot until a user-defined threshold is reached
- multiple sensors can be calibrated at the same time.

The rest of the paper is organized as follows. Related work is reviewed in the subsequent section II. The calibration design, procedure, and estimation are described in sections III-A to III-C. The mathematical description of the optimization task is given in section III-D. A few details about the implementation are reported in section III-E. Finally, experimental calibration results from an 8 minute drive are summarized in section IV.

II. RELATED WORK

A variety of extrinsic calibration methods have been published over the last few decades – an overview was recently published by [13]. Major differences exist w.r.t. the following properties:

- **target-based vs. target-free**: Most calibration methods use special calibration targets (or environments). In most cases, textured objects with a relatively simple geometry are used, e.g. checkerboard patterns, spheres, or cubes. The design of these calibration targets is optimized w.r.t. the perception properties of the sensors. A planar target for the simultaneous calibration of cameras, lidar sensors, and radar sensors was developed by [2]. In contrast, target-free methods use the unstructured environment of the robot to estimate the calibration parameters. The suitability of these environments for parameter estimation must be ensured. A method for the calibration of radar sensors w.r.t. lidar sensors was published by [9].
- **online vs. offline**: Calibration can be performed during (online) or before/after (offline) the operation of a robot. In mobile robotics, target-based methods are mostly

[2]or by an equivalent quaternion

performed offline in the course of a dedicated calibration procedure. Depending on the temporal stability of the calibration, the estimated parameters can differ from the ones during operation of the robot. Online methods, however, estimate (and possibly continuously refine) the calibration parameters. This is especially useful in case of a continuous miscalibration of the sensors, e.g. due to thermal influences or mechanical stress. An online calibration method can also be convenient if the relative position of the sensors is often changed, e.g. due to a frequent re-arrangement of the sensors (as it is the case at the AIT). An online calibration based on the matching of planes and edges which continuously also tracks the parameter uncertainties was published by [11].

In general, the choice of the optimal calibration method strongly depends on the specifics of a multi-sensor system. Thus, it is emphasized, that no definitive statement about the best properties of a calibration can be made.

III. METHOD

A. Calibration design

The method proposed herein aims to estimate the *extrinsic calibration* of an extereoceptive sensor. The main properties of the proposed calibration are:

- The calibration is applicable to any kind of sensor which provides 2D or 3D point clouds of the environment, either directly or indirectly. Lidar sensors, for instance, directly provide point clouds as opposed to stereo cameras or imaging radars which only indirectly provide point clouds through stereo matching and radar target extraction, respectively.
- The extrinsic calibration of a sensor is estimated w.r.t. the coordinate system of another sensor (*reference sensor*). The point clouds observed by these two sensors must overlap in object space.
- The calibration is continuously running during operation of the robot (*on-site calibration*[3]), i.e. no dedicated calibration procedure or post-processing of data is required. Consequently, we also do not use any special calibration patterns or objects. Instead, the immediate surroundings of the robot, observed as point clouds, are used for calibration.
- The extrinsic calibration is estimated from point clouds which are acquired while the robot is static, i.e. not moving. This has two main advantages:
  1) The path of the robot is not part of the optimization problem. Thus, a possibly erroneous path has no negative influence on the calibration process. Otherwise, the estimated calibration parameters might compensate for these errors due to the well-known correlation of the calibration parameters to the robot's path [6].
  2) Time stamps also do not have to be considered in the optimization problem. Thus, erroneous time stamps, e.g. due to a slightly incorrect time synchronization of the sensors, have no negative influence on the calibration.
- Depending on the observed scene, a single static position might not be sufficient to estimate the extrinsic calibration with acceptable precision. Thus, the calibration is refined iteratively when the robot reaches a new static position. Thereby, the previous parameter estimates, together with its precision estimates, are used as a priori observations (see next point). The whole calibration process stops once the precision of the calibration parameters are below a user-defined threshold, or in other words, once the calibration is sufficiently accurate.
- If a priori observations (measurements) of the 6 calibration parameters (or a subset thereof) exist, they are considered in the estimation process. Such observations can stem e.g. from 3D models of the multi-sensor system, from manual measurements (e.g. by using a measuring tape), or from a previously performed calibration. These observations are weighted in the least squares optimization according to their observation precision (uncertainty).
- It is possible to omit the estimation of individual calibration parameters. This is useful if some of the parameters are known in advance with very high precision and the calibration procedure is not expected to improve these estimates. In practice, this often applies to the translation vector which can be measured with a precision in the sub-millimeter range by other means, e.g. by using a total station. In contrast, it is typically rather difficult to directly measure the rotational components of the extrinsic calibration. In this context it should be noted, that incorrect angles can lead to very large displacements in object space, as the effect of angular errors is directly proportional to the range of the observed objects.

As prerequisites for the our method we assume that:

- approximate values for the 6 calibration parameters are known in advance. This prerequisite stems from the ICP (iterative closest point) algorithm [1] which is used to estimate the 6 DoF transformation between the two overlapping sensor point clouds. In practice, the rotation angles need to be known typically with a precision of a few degrees and the translation vector with a precision of a few centimeters.
- the sensors capture rather dense 3D point clouds of

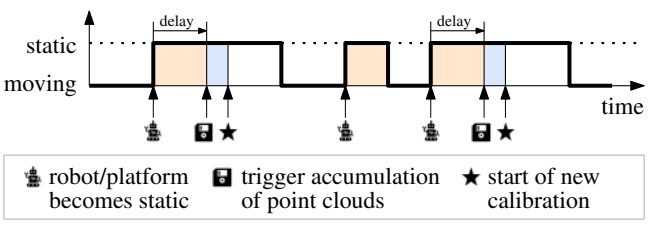


Fig. 2. Temporal sequence of the calibration.

[3]sometimes also denoted as *on-the-job calibration* or *self calibration*

the environment. However, if a sensor provides a 2D point cloud only, i.e. a single profile of the environment, the calibration is still applicable, but depending on the observed scene it might be difficult to estimate all 6 parameters of its extrinsic calibration. Consequently, only a subset of the 6 parameters should be estimated in such cases.

Finally it should be noted, that the calibration method is *not* applicable to sensors which do *not* observe the object space, e.g. navigation sensors like GNSS or IMUs, c.f. Fig. 5.

B. Calibration procedure

The temporal sequence of the calibration is depicted in Fig. 2. We distinguish between a static and a moving state of the robot, e.g. derived by the current angular and linear velocity of the robot (twist). Each time the robot becomes static, the accumulation of point clouds is triggered after a certain time delay (e.g. 2 seconds). The delay should ensure that the robot comes completely at rest before the data acquisition starts. As soon as the accumulation of point clouds is completed, a new calibration is started. This sequence is repeated until the estimated calibration parameters are sufficiently accurate.

C. Calibration estimation

The extrinsic calibration of a sensor is estimated indirectly through *point cloud matching*[4]. More specifically, the point cloud of the sensor to calibrate is matched with the overlapping point cloud of another sensor, denoted as the *reference sensor* throughout this paper. Consequently, the estimated calibration parameters describe the transformation between the two sensor coordinate systems.

The whole processing workflow of a single calibration is depicted in Fig. 4. The two accumulated sensor point clouds are used as data input, c.f. Fig. 2. First, the normal vector and a corresponding planarity value are estimated for each point. The planarity values range from 0 to 1, where 1 corresponds to a perfect plane [16]. Then, some basic point cloud filtering is carried out. This typically includes a minimum range filter, a maximum range filter, an intensity-based filter, a voxel-based thin out, and a minimum planarity filter. The latter is used to keep only planar areas (e.g. roofs, streets, walls) of the point cloud, whereas the non-planar areas (e.g. vegetation, edges, corners) are filtered out (Fig. 3). Additional filters strongly depend on the specifics of a sensor.

[4]sometimes also denoted as *point cloud registration*

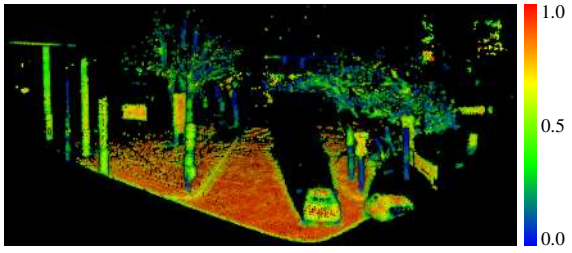


Fig. 3. Lidar point cloud colored by planarity. Points with low planarity are filtered out before matching the point clouds.

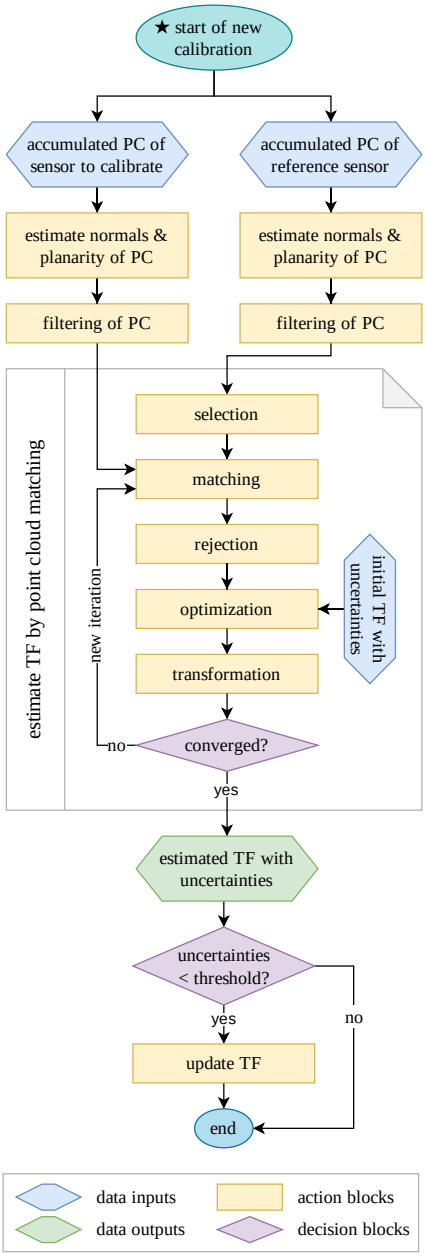


Fig. 4. Processing workflow of a single calibration (PC = point cloud, TF = transformation).

We use an ICP-like algorithm for point cloud matching with extended features. It optimizes the alignment of the two sensor point clouds by iteratively minimizing the distances (discrepancies) within the overlap area of these point clouds. This is accomplished by transforming in each iteration the point cloud of the sensor to calibrate, whereas the point cloud of the reference sensor remains fixed. As pointed out in section III-A, an approximate initial estimate of the relative alignment of the point clouds is needed. The algorithm can be broken down into five main steps [5], c.f. Fig. 4:

1) **selection**: A subset of points is selected in the point

cloud of the reference sensor. These points are selected within the overlap area of the two point clouds. We apply the *uniform sampling* selection strategy [5] which leads to a homogeneous distribution of the selected points in object space.

2) **matching**: The corresponding points of the selected subset are searched in the point cloud of the sensor to calibrate. As corresponding point the nearest neighbor is used. The nearest neighbor search is efficiently solved using a k-d tree.
3) **rejection**: False correspondences (outliers) are detected and rejected by checking the compatibility of corresponding points. Specifically, we reject correspondences on the basis of their distance and on the basis of the angle between their normals.
4) **optimization**: The transformation parameters are estimated for the sensor point cloud by minimizing the distances between corresponding points. A detailed description of this step follows in the next section.
5) **transformation**: The sensor point cloud is transformed using the estimated parameters.

Finally, a suitable convergence criterion is tested. If it is not met, the process restarts from the matching step.

Once the ICP algorithm converged, the estimated precision (uncertainty) of the transformation parameters is compared with a user-defined threshold. Consequently, the transformation between the sensors is updated only if the observed scene was well-suited for the parameter estimation.

D. Optimization

The 6 unknown parameters of the extrinsic calibration, specifically the rotation angles α_x, α_y, α_z and the translation components t_x, t_y, t_z, are estimated in a non-linear weighted least squares adjustment with conditions only (a.k.a. *Gauß-Markov* adjustment model); the solution formulas can be found e.g. in [12] (chapter 6) or [3] (chapter 4.4). The objective of the adjustment is to minimize the weighted sum of squared residuals:

$$\Omega = \underset{\alpha_x,\alpha_y,\alpha_z,t_x,t_y,t_z}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} p_i r_i^2 \right\} \tag{1}$$

where r_i is the residual of the i-th observation, p_i the corresponding weight, and n the total number of observations.

We minimize two types of residuals:

1) **Point-to-plane distances between corresponding points**. This observation type is the main element of most contemporary ICP implementations.
The residual is defined for the k-th correspondence as

$$r_k = ((R\mathbf{p}_k + \mathbf{t}) - \mathbf{q}_k)^T \mathbf{n}_k \tag{2}$$

where $\mathbf{p}_k$ and $\mathbf{q}_k$ are the corresponding points of the sensor to calibrate (movable) and the reference sensor (fixed), respectively, R is the rotation matrix composed by the Euler angles α_x, α_y, α_z, $\mathbf{t}$ is the translation vector with its components t_x, t_y, t_z, and $\mathbf{n}_k$ is the normal vector of $\mathbf{q}_k$. We prefer the signed point-to-plane error metric over alternative error metrics (e.g. point-to-point) due to its high convergence speed [14], its straight-forward mathematical formulation, and the fact that corresponding points only need to belong to the same plane[5] [5].
The weight of these residuals is defined according to the theory of least squares [12] (chapter 3.3) as

$$p_k = 1/\sigma_d^2 \tag{3}$$

where we propose to estimate σ_d from all a priori point-to-plane distances as follows:

$$\sigma_d = 1.4826 \cdot \text{mad} \tag{4}$$

Thereby, mad denotes the *median of absolute differences* (w.r.t. the median) [8] of the point-to-plane distances. It is commonly used as a robust estimator for the standard deviation of a set of random variables which is generally assumed to be normally distributed, but still contaminated by a small number of outliers.
2) **Differences to initial values of transformation parameters**. This observation type is crucial for our method as it allows (from the second calibration site onwards) to appropriately transfer the estimates from a previoulsy performed calibration. Additionally, these observations can be used (at the first calibration site) to consider any other a priori estimates of the parameters, e.g. estimates from a 3D model or from manual measurements (c.f. section III-A).
The residuals are defined for each transformation parameter as follows:

$$\begin{aligned} v_{\alpha_x} &= \overline{\alpha}_x - \alpha_x \\ v_{\alpha_y} &= \overline{\alpha}_y - \alpha_y \\ v_{\alpha_z} &= \overline{\alpha}_z - \alpha_z \\ v_{t_x} &= \overline{t}_x - t_x \\ v_{t_y} &= \overline{t}_y - t_y \\ v_{t_z} &= \overline{t}_z - t_z \end{aligned} \tag{5}$$

where $\overline{\alpha}_x$, $\overline{\alpha}_y$, $\overline{\alpha}_z$, $\overline{t}_x$, $\overline{t}_y$, $\overline{t}_z$ are the observed initial values and α_x, α_y, α_z, t_x, t_y, t_z are the estimated unknown parameters.
These residuals are weighted by considering their uncertainty estimates, e.g. for α_x:

$$p_{\alpha_x} = 1/\sigma_{\alpha_x}^2 \tag{6}$$

where σ_{α_x} is the precision of the observed value. If the initial values stem from a previous calibration, their squared precision (i.e. their variance) is given by the diagonal of the a posteriori covariance matrix of the unknown parameters, i.e. $\text{diag}(C_{\hat{x}\hat{x}})$.

[5] In contrast, when using the point-to-point error metric, corresponding points need to be exactly identical. However, such correspondences are typically very unlikely, as point clouds randomly sample the object space.

E. Implementation

The calibration is implemented as package with a single node for ROS2 (robot operating system). It is written mainly in python. The Point Data Abstraction Library (PDAL[6]) is used for pre-processing of the accumulated point clouds. The ICP-like point cloud matching algorithm is mainly based upon the following libraries: numpy, pandas, scipy, and lmfit[7]. We named this algorithm "simpleICP" and published it as open source on Github and PyPi under the MIT license [4]. Development tests have been made mainly with the simulation environment Webots[8]. When calibrating multiple sensors at the same time, an independent node is used for each sensor pair, e.g. sensor1-to-sensor2, sensor1-to-sensor3, sensor1-to-sensor4, etc.. Currently, the full processing of the pipeline as depicted in Fig. 4 takes approximately 3–5 seconds.

IV. EXPERIMENTAL RESULTS

The proposed calibration method was applied to estimate the extrinsic calibration of the sensors depicted in Fig. 5. The sensor rig was mounted on a car and is composed by an *Ouster OS1-64* (lidar1), a *Blickfeld Cube 1* (lidar2), 3 *ptgrey* cameras with 1/3" CMOS sensor and a resolution of 1.3 Mpx (camera1/2/3), and the *Indurad iSDR-p* (imaging radar).

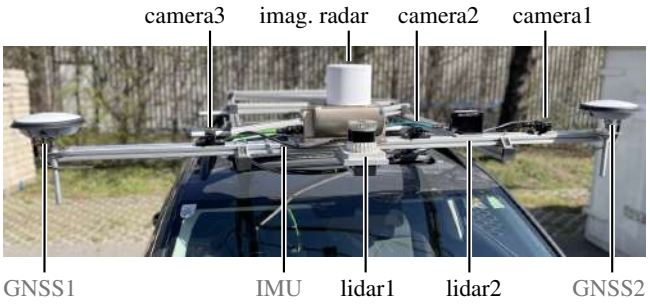


Fig. 5. The proposed method was applied to estimate the extrinsic calibration of 3 cameras, 2 lidars, and an imaging radar sensor. (The calibration of the GNSS and IMU navigation sensors, however, is not within the scope of this method.)

As pointed out in III-A, each sensor must provide a point cloud in order to apply our calibration method. Both lidar sensors directly provide point clouds. The stereo matching point cloud is derived from the camera images using the algorithm described in [10]. The imaging radar provides so-called range-amplitude maps; a point cloud is generated by extracting radar targets from these maps [7]. In Fig. 6 the point clouds provided by these sensors are visualized for a single calibration site.

The sensors have been calibrated during an 8 minute drive near the AIT headquarter (Fig. 7). In this time period, the car stopped 12 times. Accordingly, the calibration was triggered at 12 different calibration sites. In order to distinguish between the static and moving state of the car, GNSS

[6]pdal.io

[7]numpy.org, pandas.pydata.org, scipy.org, lmfit.github.io

[8]cyberbotics.com

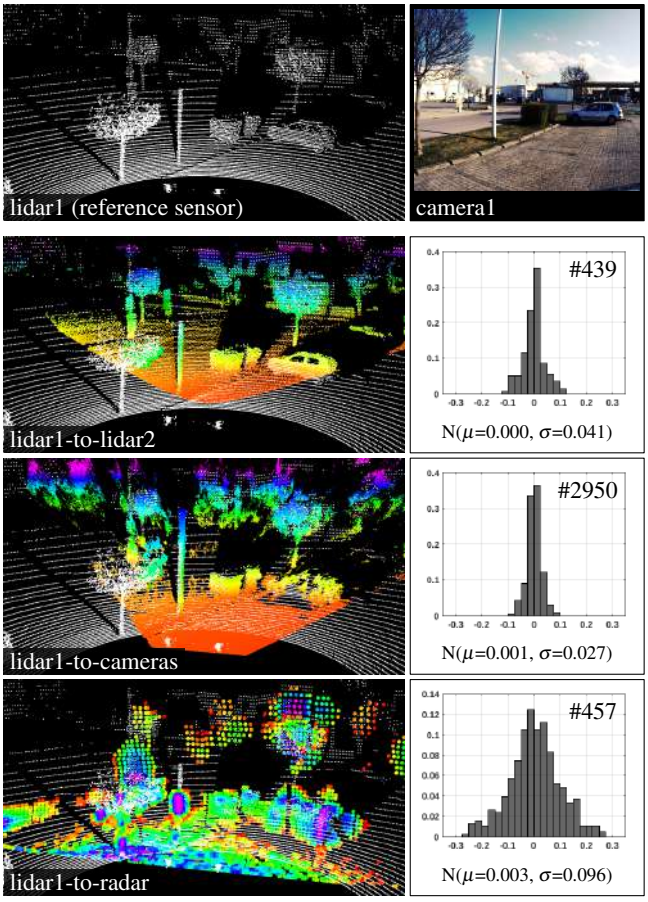


Fig. 6. Data belonging to calibration site 7, c.f. Fig. 7. First row: point cloud of the reference sensor lidar1 and a corresponding image of the site. Rows 2–4: point clouds of the sensors to calibrate (left) and histograms of the residual point-to-plane distances (right). (The point clouds of lidar2 and of the cameras are colored by height, whereas the radar point cloud is colored by measured intensity.)

measurements have been processed in a relatively simple Kalman filter. Fig. 7 shows the position of the calibration sites, as well as the corresponding point clouds as collected by the sensors lidar1 and lidar2.

It is recommended to use the sensor with the highest measurement accuracy and a large field of view (FOV) as *reference sensor*. Ideally, the point cloud of this sensor shares a large overlap area with the point clouds of the other sensors. Consequently, we have chosen the sensor lidar1 as reference for our multi-sensor system. It has the highest measurement accuracy and the largest FOV due to its 360° rotating antenna. The extrinsic calibration of the other sensors was estimated in parallel by 3 independent calibration procedures: lidar1-to-lidar2, lidar1-to-cameras, lidar1-to-radar. Fig. 6 shows the point clouds and the histograms of the residual point-to-plane distances for a single calibration site. As can be seen, the mean of the residuals is very close to zero in all 3 cases. The standard deviation, however, differs for each sensor combination as it mainly depends on the measurement accuracy of the sensors and on the selection of correspondences, specifically the minimum planarity value

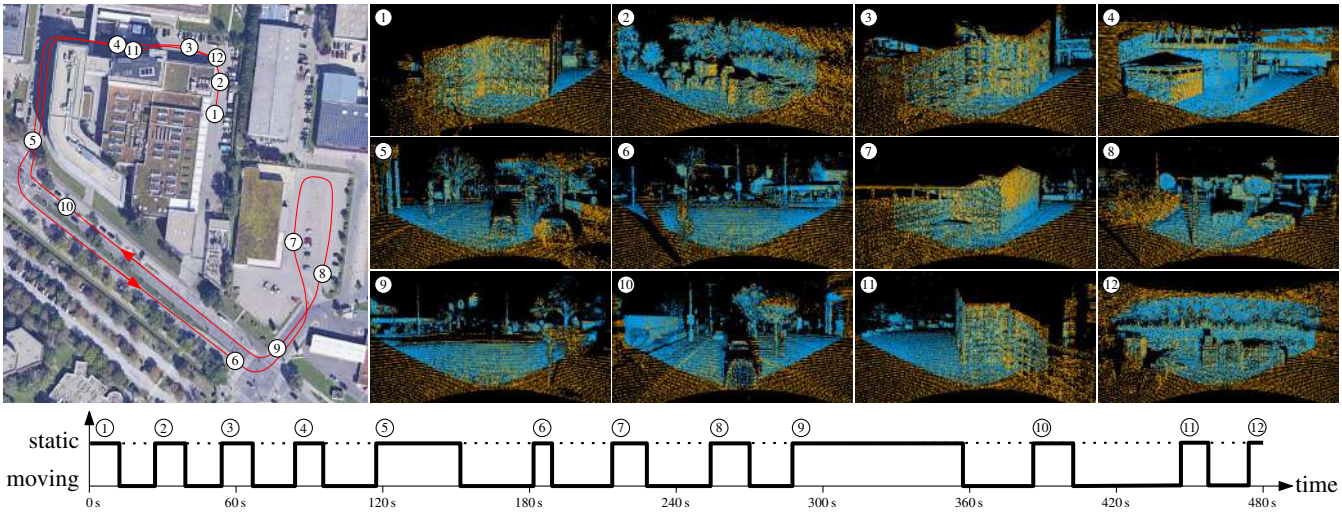


Fig. 7. The multi-sensor system depicted in Fig. 5 was calibrated during an 8 minute drive at 12 different calibration sites.

used for point cloud filtering.

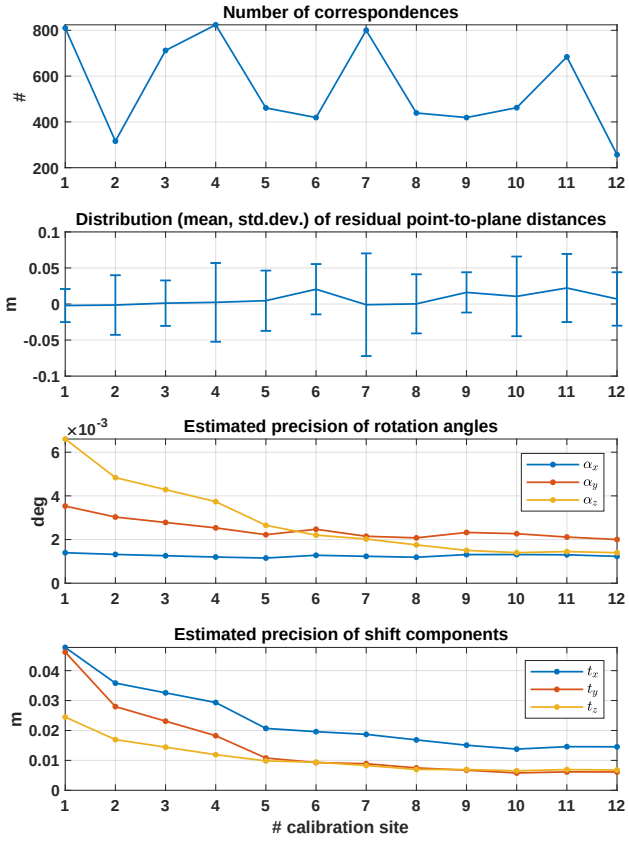


Fig. 8. Temporal development of some calibration quality indicators for the sensor combination lidar1-to-lidar2.

Fig.8 shows, specifically for the sensor combination lidar1-to-lidar2, the temporal development of some calibration quality indicators. The first graph shows the number of correspondences used in the optimization. This number mainly depends on the amount of planar areas in the scene, c.f. Fig. 7. The second graph shows the distribution of the residual point-to-plane distances. The standard deviation, again, strongly depends on the characteristics of the observed scene. On the contrary, the mean value shows a systematic behavior: it is very close to zero in the first few iterations but gets slightly larger in magnitude later. This can be explained through the iterative refinement character of our calibration method: relatively seen, the influence of the point-to-plane residuals (residual type 1, c.f. section III-D) on the parameter estimation is highest at the first calibration site. Subsequently, however, the relative influence (i.e. their weights) of the second type of residuals, the direct observation of transformation parameters from previous calibrations, continuously grows. Finally, the third and fourth graphs show the estimated precision of the transformation parameters as given by the a posteriori covariance matrix of the unknown parameters $C_{\hat{x}\hat{x}}$. Here, one can observe that their precision is improving over time which can be regarded as one of the main benefits of our method.

V. SUMMARY AND OUTLOOK

We proposed in this work a new method for the extrinsic calibration which:

- does not use any special calibration targets (target-free)
- is widely sensor agnostic
- can be applied simultaneously to multiple sensors
- continuously improves the calibration parameter estimates over time

Our future work will concentrate on:

- the automatic removal of dynamic objects (like cars, persons, etc.) from the collected point clouds
- the calibration of sensors which provide 2D point clouds only, i.e. a profile of the environment
- better understand the long-time behavior of the estimated precision of the calibration parameters, e.g. for an operation time of several hours

In addition, we plan to publish the code as open source package for ROS2.

REFERENCES

[1] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[2] J. Domhof, J. F. Kooij, and D. M. Gavrila, "An extrinsic calibration tool for radar, camera and lidar," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8107–8113.

[3] W. Förstner and B. Wrobel, *Photogrammetric Computer Vision – Statistics, Geometry, Orientation and Reconstruction*. Springer, 2016.

[4] P. Glira, "simpleICP: Implementations of a rather simple version of the iterative closest point algorithm in various languages." https://github.com/pglira/simpleICP, Apr. 2022.

[5] P. Glira, N. Pfeifer, C. Briese, and C. Ressl, "A correspondence framework for ALS strip adjustments based on variants of the ICP algorithm," *PFG Photogrammetrie, Fernerkundung, Geoinformation*, vol. 2015, no. 4, pp. 275–289, 08 2015. [Online]. Available: http://dx.doi.org/10.1127/pfg/2015/0270

[6] P. Glira, N. Pfeifer, and G. Mandlburger, "Hybrid orientation of airborne lidar point clouds and aerial images," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2019. [Online]. Available: https://doi.org/10.5194/isprs-annals-IV-2-W5-567-2019

[7] P. Glira, C. Weidinger, T. Kadiofsky, W. Pointner, K. Ölsböck, C. Zinner, and M. Doostdar, "3d mobile mapping of the environment using imaging radar sensors," *Proceedings of RadaConf2022*, 2022.

[8] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, 1974.

[9] L. Heng, "Automatic targetless extrinsic calibration of multiple 3d lidars and radars," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 669–10 675.

[10] M. Humenberger, C. Zinner, M. Weber, W. Kubinger, and M. Vincze, "A fast stereo matching algorithm suitable for embedded real-time systems," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1180–1202, 2010.

[11] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-lidar systems with online extrinsic calibration," *IEEE Transactions on Robotics*, 2021.

[12] E. Mikhail and F. Ackermann, *Observations and least squares*. University Press of America, 1976.

[13] J. Nie, F. Pan, D. Xue, and L. Luo, "A survey of extrinsic parameters calibration techniques for autonomous devices," in *2021 33rd Chinese Control and Decision Conference (CCDC)*, 2021, pp. 3543–3548.

[14] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. Quebec City, Canada: IEEE, 2001, pp. 145–152.

[15] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[16] M. Weinmann, B. Jutzi, and C. Mallet, "Semantic 3d scene interpretation: A framework combining optimal neighborhood size selection with relevant features," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 3, p. 181, 2014.

Continual Learning Benchmarks for Antipodal Grasping

Sayantan Auddy[1] Jakob Hollenstein[1] Matteo Saveriano[1,3] Antonio Rodríguez-Sánchez[1] Justus Piater[1,2]

***Abstract*— A continual learning robot that can repeatedly learn from new data without forgetting past knowledge is surely preferable to a robot that cannot learn incrementally. As most continual learning methods focus on image classification tasks, it is not clear if or how they scale to more complicated vision problems such as robotic grasp prediction. To fill this gap, we propose a set of benchmarks that can be used to evaluate continual learning methods on the problem of antipodal grasping. We adapt a state-of-the-art grasp prediction model for continual learning and evaluate multiple baselines using our benchmarks. Our preliminary findings indicate that replay-based methods may be suitable for the grasp prediction task.**

I. INTRODUCTION

Antipodal robot grasping is arguably a challenge that has already been met, considering the performance of state-of-the-art methods [8], [11], [14]. All such methods either rely on large scale grasp datasets [3], or use domain randomization [16] to diversify the training data [3], [10]. However, it may not be possible to completely anticipate every possible situation in advance. A better strategy is to augment the grasp learning methods with *continual learning* (CL) [13] to make them capable of learning from a sequence of multiple small, disjoint, and non-IID datasets that are encountered over time.

Barring a few works which address continual learning for robotics [2], [5], most of the current CL research [7], [9], [15], [18] focuses on image classification problems using relatively simple network architectures. It is not apparent how well these methods scale to robotics applications, which typically involve the use of more complex network components and architectures. If properly applied to robot learning, CL can help achieve open-ended robot learning, and this would be especially useful for a ubiquitous robotics problem such as vision-based grasp prediction. With this motivation, we present preliminary work on a set of continual learning benchmarks that can be used to evaluate the effectiveness of continual learning methods on the problem of grasp learning. We adapt a state-of-the-art grasp learning method [8] for continual learning and evaluate multiple baselines on our benchmarks. Our initial results indicate that replay-based CL strategies outperform regularization-based CL.

II. CONTINUAL LEARNING FRAMEWORK

A. Benchmarks

The Cornell dataset [14] is a widely used dataset for grasp learning [1]. It consists of 885 RGB images of 280 different household objects, where each image is annotated with multiple top-down grasp rectangles. We create the following 4 benchmarks by partitioning the images of this dataset into multiple *tasks* (sub-datasets), where each benchmark is learned independently by learning its tasks sequentially:

Shape: We manually partition the 885 images of the Cornell dataset into 5 tasks, where each task contains images of objects with one of these shapes: *rectangular* (e.g. boxes, TV remotes), *rim* (objects with a graspable rim, e.g. bowls, frisbees), *long* (elongated objects, e.g. stick), *round* (objects with a circular symmetry, e.g. apple, potato), and *handle* (objects with a handle, e.g. spatula, toothbrush).

Width 5: For each image we compute the average of the 5 largest grasp widths. Then, 5 tasks are defined by partitioning the images according to their average grasp width such that each task has roughly the same number of images.

Width 10: We follow the same process as Width 5, but here we partition the 885 images into 10 tasks.

Object: We compute the number of images for each distinct object and then choose 10 objects with the most number of images. Each of these 10 objects corresponds to a task.

Shape has highly disbalanced tasks, whereas the *Width* benchmarks have balanced tasks. *Object* has very few images per task. For each task, we create training and validation sets in the ratio 75:25 and use image augmentation (random rotations, translations, crops) to learn from such small datasets.

B. Baselines

As our base architecture, we choose the fully convolutional network proposed in [8]. This network has 1.8×10^6 parameters and produces heatmaps for the grasp center, orientation and width. Using this, we implement the following baselines:

Finetuning (FT): For each benchmark, a network is initialized at the beginning and is then successively finetuned on each task. This forms the lower performance baseline, as here we would expect that only the last task is remembered.

Replay 20% (RE20): This setting is similar to FT, but for each task 20% of the training data is randomly cached and then combined with the training data of the next task. For example, while training for task 2, we use all the data of task 2 and 20% of the data each from tasks 1 and 0.

Replay 100% (RE100): This is the same as RE20, except that here all the data from past tasks is cached and replayed.

Synaptic Intelligence (SI): This setup is similar to FT, but the grasp prediction network's parameters are protected from *catastrophic forgetting* [13] using a regularization term according to the formulation of Synaptic Intelligence [18].

Joint Training (JT): This forms the upper performance baseline. For learning each task, we use a freshly initialized network and the data for all tasks up to that point. Note that

[1] Department of Computer Science, University of Innsbruck, Technikerstrasse 21a, Innsbruck, Austria. {name.surname}@uibk.ac.at

[2] Digital Science Center (DiSC), University of Innsbruck, Austria.

[3] Department of Industrial Engineering, University of Trento, Italy.

all the data of previous tasks is also used for RE100 but it does not reinitialize the network for each task.

III. Results

We train and evaluate the baselines on each benchmark independently, where the tasks of a benchmark are learned in sequence. Each training run is repeated with 3 independent seeds. To compute the accuracy of the predicted grasp, we compare the intersection over union (IoU) of the prediction with the grasp annotations, and if any of these comparisons have an IoU of more than 25%, the prediction is considered to be accurate [8]. After training on each task, each baseline is evaluated on the validation sets of the current and past tasks. For example, after training on task 3, the network is evaluated on tasks 0, 1 and 2. The average of these validation accuracies is depicted in Fig. 1. The upper baseline JT maintains an average accuracy close to 90% after all tasks for all benchmarks. On the other hand, the performance of the lower baseline FT drops as more tasks are learned, clearly exhibiting *catastrophic forgetting* [13]. The regularization-based CL method (SI [18]) is not able to avoid forgetting past tasks and performs similar to FT. The replay-based baselines perform better than FT and SI but worse than JT.

In Fig. 2, we show how the validation accuracy for each of the 10 tasks of the *Width 10* benchmark changes as newer tasks are learned. The drop in accuracy for the oldest tasks can be seen for all the baselines except JT. This drop is severe for FT and SI, for which the accuracy for task 0 (which corresponds to objects needing the smallest and most precise grasps) drops to around 20% after all tasks are learned. It can also be seen that for FT and SI, whenever a new task is learned, its accuracy starts around 90%, but starts dropping sharply as newer tasks are learned.

Using the validation accuracies we also compute continual learning metrics [4] in Tab. I (for *Width 10*). In terms of accuracy (ACC) over all tasks, and remembering (REM) past tasks, JT is the best, followed by the two replay-based baselines. Since joint training and replay involve the storage of data from past tasks, they achieve low scores on the storage size efficiency metric (SSS). Interestingly, FT has the highest forward transfer (FWT) score, indicating that it is the best at using past knowledge to learn newer tasks.

IV. Summary and Outlook

The preliminary results presented in this short paper indicate that (i) parameter regularization may not be as effective as replay-based CL for continual grasp learning, and (ii) it is possible to learn grasps using small datasets. To further expand these findings, we will evaluate more continual learning methods [6], [7], [15], [17] in the future. We also plan to perform similar evaluations for other robot vision applications such as affordance detection [12]. Our goal for this future work will be to identify areas where current continual learning methods can be improved to make them more suitable for robotics tasks.

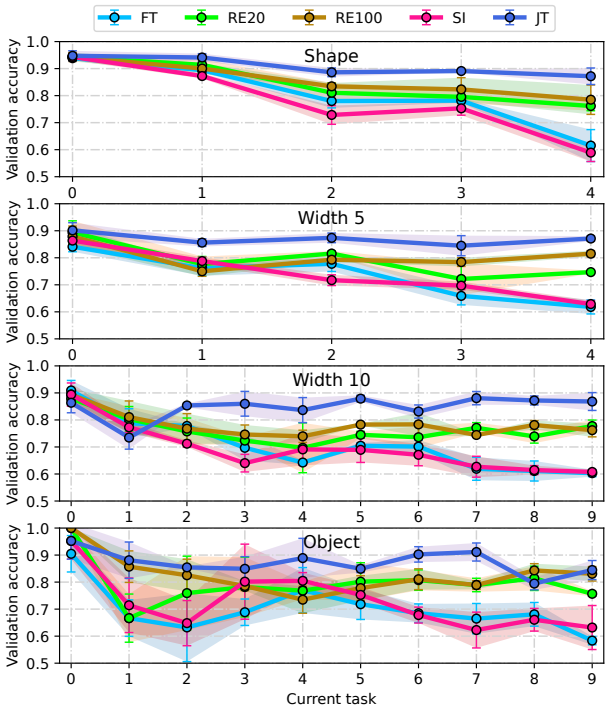


Fig. 1. Cumulative validation accuracies for current and past tasks. Each data point shows the mean accuracy of all tasks till that point on the $x-$axis.

TABLE I
CL metrics for Width 10 (1-best, 0-worst).

Benchmark	Baseline	ACC	REM	FWT	SSS
Width 10	FT	0.663	0.733	**0.732**	**1.000**
	RE20	0.752	0.827	0.711	0.912
	RE100	0.769	0.839	0.681	0.550
	SI	0.655	0.717	0.698	**1.000**
	JT	**0.858**	**0.943**	0.702	0.550

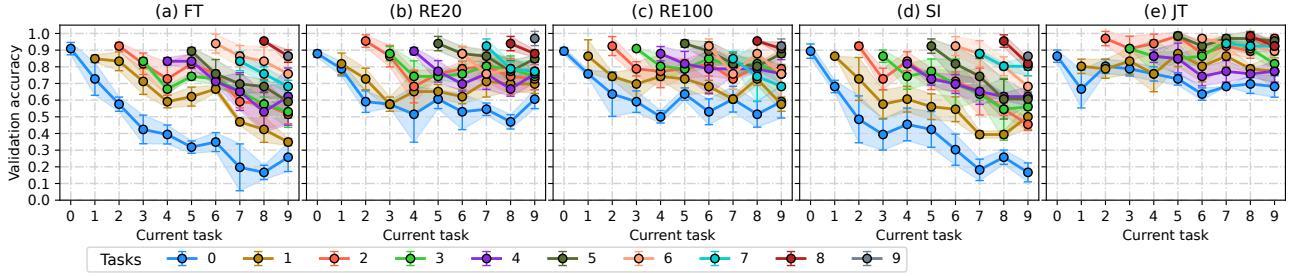


Fig. 2. Per-task validation accuracies for *Width 10*, showing how the accuracy of each task changes as newer tasks are learned.

REFERENCES

[1] "Robotic Grasping on Cornell Grasp Dataset," https://paperswithcode.com/sota/robotic-grasping-on-cornell-grasp-dataset-1, accessed: 2022-04-22.

[2] S. Auddy, J. Hollenstein, M. Saveriano, A. Rodríguez-Sánchez, and J. Piater, "Continual learning from demonstration of robotic skills," *arXiv preprint arXiv:2202.06843*, 2022.

[3] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3511–3516.

[4] N. Díaz-Rodríguez, V. Lomonaco, D. Filliat, and D. Maltoni, "Don't forget, there is more than forgetting: new metrics for continual learning." [Online]. Available: http://arxiv.org/abs/1810.13166

[5] C. Gao, H. Gao, S. Guo, T. Zhang, and F. Chen, "Cril: Continual robot imitation learning via generative and prediction model," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 6747–5754.

[6] T. L. Hayes, K. Kafle, R. Shrestha, M. Acharya, and C. Kanan, "REMIND your neural network to prevent catastrophic forgetting." [Online]. Available: http://arxiv.org/abs/1910.02509

[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks." [Online]. Available: http://arxiv.org/abs/1612.00796

[8] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," version: 3. [Online]. Available: http://arxiv.org/abs/1909.04810

[9] T. Lesort, "Continual learning: Tackling catastrophic forgetting in deep neural networks with replay processes." [Online]. Available: http://arxiv.org/abs/2007.00487

[10] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[11] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach." [Online]. Available: http://arxiv.org/abs/1804.05172

[12] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1374–1381.

[13] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," vol. 113, pp. 54–71. [Online]. Available: http://arxiv.org/abs/1802.07569

[14] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 1316–1322.

[15] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," p. 10.

[16] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, *et al.*, "Domain randomization and generative models for robotic grasping," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3482–3489.

[17] J. von Oswald, C. Henning, B. F. Grewe, and J. Sacramento, "Continual learning with hypernetworks," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://arxiv.org/abs/1906.00695

[18] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence." [Online]. Available: http://arxiv.org/abs/1703.04200

Towards Self-Awareness in Multi-Robot Systems

Mohammad Rahmani[1] and Bernhard Rinner[2]

I. Introduction

Self-awareness (SA) is a broad concept borrowed from cognitive science and psychology that describes the property of a system, which has knowledge of "itself," based on its own senses and internal models. This knowledge may take different forms, is based on perceptions of both internal and external phenomena, and is essential for being able to anticipate and adapt to unknown situations [2]. Deploying this concept on robots poses some fundamental challenges and requires some key capabilities of autonomous robots: (1) learn inference models from sensor inputs, (2) infer its state and the environment's state based on the models, and (3) detect abnormalities between observed and inferred behaviors. An abnormality detection may indicate a new phenomena observed by the robot and trigger the creation of a new model. Over time the robot acquires a set of models representing different phenomena.

This work proposes a framework towards self-awareness in multi-robot systems and presents preliminary results from a simulation study. In particular, we adopt hierarchical dynamic Bayesian networks (DBN) for modelling the observed internal (via proprioceptive sensors) and external (via exteroceptive sensors) phenomena [5]. Hierarchical DBNs allow to perform inferences and contextualize proprioceptive and exteroceptive sensory data at different abstraction levels. These inferences serve then as input for abnormality detection. We further extend modelling to multi-robot systems by coupling hierarchical DBNs.

II. Framework

Even though current studies [4], [3] suggest detecting abnormality from an inference model trained by evenly ordered exteroceptive and proprioceptive sensory data, we suggest deriving independent inference models for each of these types of sensors and pair only exteroceptive models with a following proprioceptive model. As such, the robots can (1) choose the most appropriate proprioceptive inference model according to the best predicting exteroceptive model, (2) deduce preferred next states from proprioceptive most probable states derived from its inference model to be used for control decisions, and (3) ignore exteroceptive observations for which no control decision should be made internally.

[1]Mohammad Rahmani is with the Digital Age Research Center (DECIDE doctoral school) at the University of Klagenfurt, Austria. mohammad.rahmani@aau.at

[2]Bernhard Rinner is with the Institute of Networked and Embedded Systems at the University of Klagenfurt, Austria. bernhard.rinner@aau.at

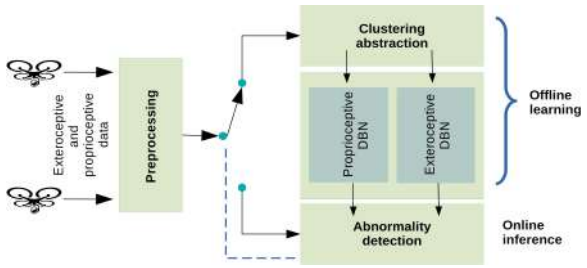


Fig. 1. The proposed SA framework learns offline proprioceptive and exteroceptive models which are used for online abnormality detection.

Figure 1 depicts our proposed SA framework for a multi-robot system. Proprioceptive and exteroceptive sensor data is preprocessed and either forwarded to offline learning or online abnormality detection. In the offline phase, two models based on coupled hierarchical DBNs are learned from the observed behavior of the robots. In the online phase, inferences from the learned models are compared with the current observations. An abnormality indicates a deviation between learned and observed behavior and may trigger the learning of new models.

A. Coupled hierarchical DBNs

Causal-temporal behaviors can be modeled by DBNs, which also support a hierarchical representation using various well-known approaches. For example, Kalman filters can be used for the continuous level, whereas particle filters can be used for the discrete level. We thus adopt DBNs for our modelling approach (cp. Figure 2). We cluster the observed sensor data Z into a sequence of quasi-stationary segments where the continuous state X represents the behavior within a segment and each segment corresponds to an abstracted state S. Coupled DBNs introduce an additional coupling layer D between multiple DBNs and causal relationships among the abstract state variables to model the interaction between multiple robots.

B. State transition and Abnormality Detection

For collective behavior, we couple the contributing individual behaviors based on their abstract states and refer to the possible combinations as coupled state D. State transitions indicate changes of quasi-stationary behaviors and are modelled by a matrix of state transition probabilities computed by the occurrence of successive coupled states according to closest observations.

Abnormality detection is founded on measuring the distance between the most probable coupled state prediction and the current observations of multiple robots. In particular, we

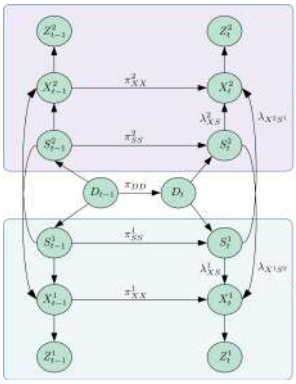


Fig. 2. Hierarchical coupled DBN. Horizontal lines present temporal relationship between random variables at two consequent time steps t and $t-1$. Vertical lines present the causal relationship between them. Z represents the observation, X the continuous state, S the abstract level state, and D the coupled state. π represents the transition probabilities at different levels and λ the occurrence likelihood of states according to lower level parameters.

use the Kullback–Leibler divergence between the center of the forming abstract states of D and the robots' observation as abnormality value. If this value exceeds a threshold the current abstract state is inconsistent with the observed behavior triggering the creation and training of a new coupled DBN model.

III. Preliminary Results

We conducted a simulation study with the multi unmanned aerial vehicle (UAV) simulator CTU-MRS [1] which is built upon the Robot Operating System (ROS). We use four scenarios where two UAVs fly along rectangular trajectories and capture GPS position as proprioceptive data and LIDAR measurements as exteroceptive data for our study. These multi-UAV scenarios are implemented in a leader-follower architecture using model-predictive control for the Pixhawk 4 autopilots. One scenario serves as reference scenario[1] for learning the initial models (cp. Figure 3 left), whereas the others serve as test scenarios that include some blockage along the planned trajectories resulting in some evasive manoeuvres of the UAVs[2].

We trained the coupled DBNs with the captured GPS and LIDAR data from 10 simulation runs. The sensor data was clustered into 75 abstract states (cp. Figure 3 right). Figure 4 (top) shows the abnormality values for the reference and one test scenario. The abnormality values (Kullback–Leibler divergence) for the reference scenario remains below 10^3 units while for the test scenario, they reach approximately to 5.0×10^5 units in the regions where the blockage happens.

Figure 4 (bottom) shows the abnormality values for the LIDAR. For feature extraction of the LIDAR data, we used an artificial neural auto-encoder with 5 layers each for the encoder and the decoder and reduced the LIDAR scans to 2 dimensions. The abnormality value increases before the UAVs enter the blockage area. This early detection is

[1] youtube.com/watch?v=GHD4VmcIHFo
[2] youtube.com/watch?v=1YGSk7YKcpI

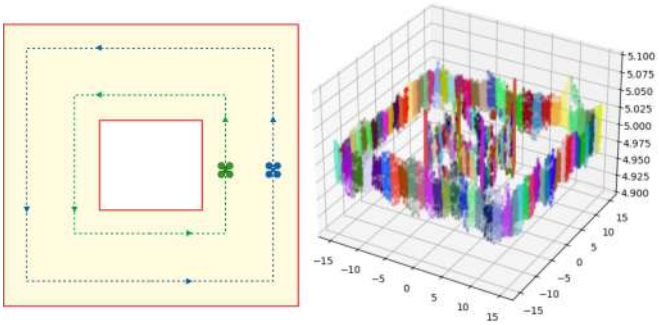


Fig. 3. Two UAV reference scenario (left) and clustering of the GPS data to form coupled states (right).

expected since the LIDAR can scan the environment of some distance.

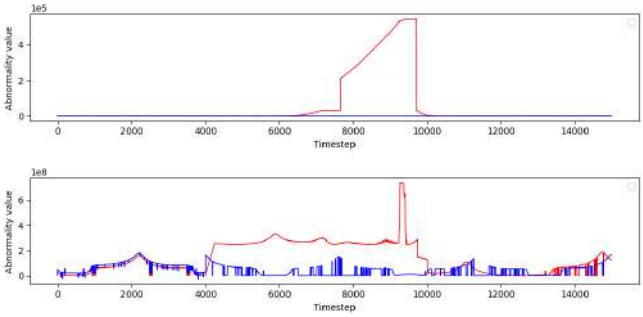


Fig. 4. Abnormality values of the reference (blue) and test (red) scenario using the proprioceptive model and GPS data (top) and the exteroceptive model and the LIDAR data (bottom). The evasive UAV behavior in the test scenario results in a significant increase of the abnormality value.

IV. Conclusion

We introduced a framework towards self-awareness in multi-robot systems capable of learning offline proprioceptive and exteroceptive models which can be used for online abnormality detection. As future work we plan to expand our simulation study to more complex scenarios including different sensors, to investigate alternative distance metrics for abnormality detection, and to efficiently rank the validity of multiple models.

References

[1] T. Báča *et al.*, "The mrs uav system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, 2021.

[2] N. Dutt, C. S. Regazzoni, B. Rinner, and X. Yao, "Self-awareness for autonomous systems," *Proceedings of the IEEE*, vol. 108, no. 7, pp. 971–975, 2020.

[3] K. Friston, B. Sengupta, and G. Auletta, "Cognitive dynamics: From attractors to active inference," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 427–445, 2014.

[4] S. Haykin, *Cognitive Dynamic Systems: Perception-action Cycle, Radar and Radio*. Cambridge, 2012.

[5] C. S. Regazzoni, L. Marcenaro, D. Campo, and B. Rinner, "Multisensorial generative and descriptive self-awareness models for autonomous systems," *Proceedings of the IEEE*, vol. 108, no. 7, pp. 987–1010, 2020.

Sensor fusion of IMU and GPS for geofencing on an industrial control system for safe mowing in road areas

Stefan Dumberger[1], Raimund Edlinger[1], Philipp Bauer[1], Michael Zauner[1], Andreas Nüchter[2]

***Abstract*— The paper presents the implementation of geofencing on a mobile industrial controller for the automation of a mobile carrier vehicle for agricultural use. This system is used to prevent collision with well-known obstacles as well as avoid the accidental run-away of the vehicle under autonomous operation. It describes the mathematical principle used, as well as the additional features needed for a practical implementation on an industrial controller. As it is necessary to ensure the operation under real-world conditions, the paper also handles the data acquisition, pre-filtering and sensor fusion for system. Finally the correct behavior of the system is evaluated with multiple test-cases as well as experimental field tests.**

I. INTRODUCTION

This publication is part of a larger research project, where the goal is autonomous mulching of highway embankments using the agricultural tool carrier platform Reform Metron P48RC, depicted in Fig. 1. The focus of this paper in particular is the usage of geofencing by defining a limited area of operation and by extend preventing the robot to work in sections not intended to be mulched, collide with well known obstacles (e.g.: trees, pipes, trees, ditches, ...) or in the worst case drive onto the motorway.

The term geofencing, referring to the definition taken from Koch [2, p 11], describes a service for monitoring of a virtual boundary related to a physical area and raising a signal once a relevant object ether enters or leaves this area. While most geofencing systems use a geographical border defined via WGS84 coordinates, the concept itself can be used in any coordinate system where both the perimeter as well as the current position itself can be expressed relative to a common point of reference (e.g. a local map of a building).

The main complexity in our case is the reliability on this system to prevent potentially dangerous situations while being aware that the area of operation will contain a variety of objects, like trees or bridges, which will temporary interfere or completely block the reception of GNSS signals. Therefore an additional sensor fusion to bridge temporary outages was implemented.

II. RELATED WORK

Geofencing has a wide array of application on different scales in modern technologies. Fundamentally it can be divided into two main categories: The monitoring of entering a specific area and the check that a object or mobile platform is not leaving a specific area.

Fig. 1: Carrier platform with attachment and automation kit

The first use case is mainly used in logistics, where a region is formed around the destination of a delivery. Once the truck enters this region, a signal is sent to the logistics center, which can better plan the immanent arrival [7]. Another application is the collection of tolls. Once a vehicle gets into the vicinity of a tollbooth (approx. 200m), the system automatically requests the corresponding data from the provider. This results in the transaction already being handled once the vehicle reaches the tollbooth and the gate can open without any delay. This approach provides a high density of up to 300 vehicles per hour while providing individual billing conditions for different companies [3].

For the second category a variety of use cases can be found in various fields. For example [1] describes a method to use geofencing in agriculture for the monitoring of animal herds. In an logistical context the check for exiting a specific area can be used to examine if a vehicle is on its correct route or help in the case of theft of a vehicle [5].

When combining both the entering and leaving of a specific area, new applications arise for example when handling dangerous areas, emergencies or natural disasters. in these scenarios the systems described in [8] and [6] help in distributing information quickly, gathering information about the situation as well as self-organization for affected persons.

However in most of these examples the systems expect the equipment to have access to a GNSS signal all the time. When this assumption cannot be guaranteed the system can only provide convenience features, which may not work

[1]Authors are with University of Applied Sciences Upper Austria, 4600 Wels, Austria *forename.sirname*@fh-wels.at

[2]Andreas Nüchter is with Faculty Informatics VII - Robotics and Telematics, Julius-Maximilians-University Würzburg, 97070 Würzburg, Germany andreas.nuechter@uni-wuerzburg.de

all the time. Whenever a geofencing application must work under all circumstances, the system cannot rely on GNSS alone and in most cases additional infrastructure sensors to guard the perimeter are required (See [3] and [1]).

III. ALGORITHMIC APPROACH

The core principle of geofencing is to check whether or not a POI (point of interest) is within a predefined region. In our case the region is given via a list of boundary points creating a closed polygon.

However the simple check for the point to be inside the polygon is not enough in a practical use-case considering measurement uncertainty and the dynamics of the robotic systems. The border polygon needs to be scaled inwards to generate a buffer zone. In addition the current distance to the border can be interesting for adapting the maximum speed of the vehicle.

A. point-in-polygon algorithm

To decide whether or not a POI is inside the polygon, the Jordan curve theorem for polygons [4] is used. It generates a ray beginning at the POI and counts the intersections against all polygon edges $\overrightarrow{P_nP_{n+1}}$. Whenever the total count of intersections is odd, the POI is inside the polygon. To check for intersection the following equation can be used, using $C = P_n$, $D = P_{n+1}$, $A = POI$ and the arbitrary point $B = \begin{pmatrix} POI_x \\ POI_y + 1000 \end{pmatrix}$.

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{(B_y - A_y)(D_x - C_x) - (B_x - A_x)(D_y - C_y)} \cdot \begin{pmatrix} -(D_y - C_y) & (D_x - C_x) \\ -(B_y - A_y) & (B_x - A_x) \end{pmatrix} \begin{pmatrix} C_x - A_x \\ C_y - A_y \end{pmatrix} \tag{1}$$

An intersection can be found whenever $\alpha, \beta \in [0,1]$.

B. polygon scaling

The generation of the scaled down polygon needs to be handled with care, as this procedure can generate overlaps, as can be seen in Fig. 2a. If not removed, these overlaps will break the point-in-polygon algorithm for the newly created polygon. Therefore the following steps need to be taken:

1) transforming points inwards: In the first step four temporary points need to be generated for each polygon corner B using the previous corner A and next corner C. When ordering the corners in a counterclockwise manner, these points can be calculated using

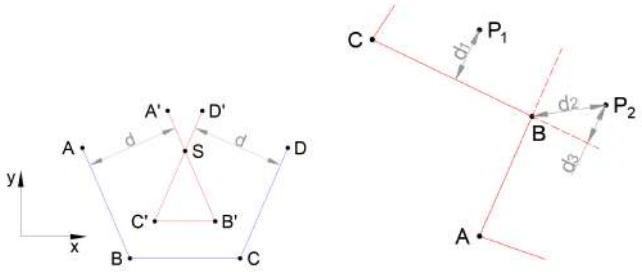
$$\begin{pmatrix} P_x \\ P_y \end{pmatrix} = \begin{pmatrix} X_x \\ X_y \end{pmatrix} + d \cdot \begin{pmatrix} \overrightarrow{e}^{\perp}_{XY\,y} \\ -\overrightarrow{e}^{\perp}_{XY\,x} \end{pmatrix} \text{ with } \begin{cases} X = A, Y = B \text{ for } P_1 \\ X = B, Y = C \text{ for } P_3 \end{cases}$$
$$\begin{pmatrix} P_x \\ P_y \end{pmatrix} = \begin{pmatrix} Y_x \\ Y_y \end{pmatrix} + d \cdot \begin{pmatrix} \overrightarrow{e}^{\perp}_{XY\,y} \\ -\overrightarrow{e}^{\perp}_{XY\,x} \end{pmatrix} \text{ with } \begin{cases} X = A, Y = B \text{ for } P_2 \\ X = B, Y = C \text{ for } P_4 \end{cases} \tag{2}$$

Afterwards a new corner B' can be calculated using the intersection between the two straight lines g_1 and g_2, defined as

$$g_1(x) = \frac{P_{2y} - P_{1y}}{P_{2x} - P_{1x} \cdot x + \frac{P_{2x} \cdot P_{1y} - P_{1x} \cdot P_{2y}}{P_{2x} - P_{1x}}}$$
$$g_2(x) = \frac{P_{4y} - P_{3y}}{P_{4x} - P_{3x} \cdot x + \frac{P_{4x} \cdot P_{3y} - P_{3x} \cdot P_{4y}}{P_{4x} - P_{3x}}} \tag{3}$$

and Cramer's rule to calculate the x- and y- component.

2) removing overlaps: As stated beforehand, this reduction of the polygon can lead to overlaps. To remove these phenomenons a section of four neighbouring corners A'. B', C' and D' is evaluated for each corner. Whenever the two vectors $\overrightarrow{A'B'}$ and $\overrightarrow{C'D'}$ intersect, using equation (1), the two inner points B' and C' are removed and replaced by the intersection point S.



(a) scaled polygon overlap (b) distance problem

Fig. 2: visualization of mathematical problems encountered in the algorithm

C. shortest distance to polygon

When determining the distance between the POI and the polygon, two scenarios can occur: The nearest polygon feature is ether a edge or a corner. Our approach assumes the nearest feature to be an edge in the first place and checks if this assumption was correct afterwards.

1) distance to edge: To calculate the minimal distance between the POI and an polygon edge, the edge is interpreted as a straight line defined by the two neighbouring corners. Then the normal distance between this line and the POI is calculated. However, as can be seen on the example of P_2 in Fig. 2b, this calculated distance can lay outside the confined section between the two polygon corners. To filter these outliers a rotated bounding box is used.

2) rotated bounding box: We define a rotated bounding box as a square around two points A and B. The length is equal to the distance between the two points, the orientation is the same as the vector $\overrightarrow{AB}$ and the width is defined two times the arbitrary distance d. With these constraints the four corners of the box can be calculated as follows:

$$\begin{pmatrix} p_{0x} \\ p_{0y} \end{pmatrix} = \begin{pmatrix} A_x \\ A_y \end{pmatrix} + d \cdot \begin{pmatrix} \overrightarrow{e}^{\perp}_{AB\,y} \\ -\overrightarrow{e}^{\perp}_{AB\,x} \end{pmatrix} \tag{4}$$

$$\begin{pmatrix} p_{1x} \\ p_{1y} \end{pmatrix} = \begin{pmatrix} A_x \\ A_y \end{pmatrix} + d \cdot \begin{pmatrix} -\overrightarrow{e}^{\perp}_{AB\,y} \\ \overrightarrow{e}^{\perp}_{AB\,x} \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} p_{2x} \\ p_{2y} \end{pmatrix} = \begin{pmatrix} B_x \\ B_y \end{pmatrix} + d \cdot \begin{pmatrix} \overrightarrow{e}^{\perp}_{ABy} \\ -\overrightarrow{e}^{\perp}_{ABx} \end{pmatrix} \quad (6)$$

$$\begin{pmatrix} p_{3x} \\ p_{3y} \end{pmatrix} = \begin{pmatrix} B_x \\ B_y \end{pmatrix} + d \cdot \begin{pmatrix} -\overrightarrow{e}^{\perp}_{ABy} \\ \overrightarrow{e}^{\perp}_{ABx} \end{pmatrix} \quad (7)$$

Using the normal distance calculated in the last step as d, this value is valid minimal distance as long as the POI is inside the polygon defined by P_1. P_2, P_3 and P_4.

3) distance to corner: If no suitable candidate is found using this method on all corners of the polygon, the nearest feature to the POI is by definition a corner. This distance can now be calculated using the pythagorean theorem between the POI and every polygon corner. The lowest result is automatically the globally smallest distance.

IV. Implementation

The POI algorithm was implemented on an industrial controller communicating with an external GNSS module. For the analysis of accuracy and repeatability, the measurement system has been prototyped for the first outdoor tests.

A. Hardware Architecture

The hardware implementation for this project is shown in Fig. 4 and split into two dedicated hardware modules to maximize flexibility.

1) Sensor board: Raw sensor data acquisition is done on its own PCB to allow the independent data access over network from multiple computation units. In addition to the sensors, the board contains multiple voltage regulators and an ATMEL ATxmega32A4 microcontroller, which reads out all sensor values in an fixed frequency and provides the measurements over the network.

The inertial measurement unit (IMU) on the board is a TDK InvenSense MPU9250. It can measure linear accelerations up to $\pm 16g$, angular velovity up to ± 2.000 degrees per second, and a magnetic field strength up to $\pm 4900 \mu T$.

Finally the board uses a u-blox NEO-M8N module for global positioning. This module is technically able to use the free services provided by GPS, GLONASS, Galileo and BEIDOU. However in the scope of this project only GPS and GLONASS were enabled to test whether the system could also work on low-cost hardware.

2) Industrial controller: All further data processing and computation is done on a X90CP174.24-00[1]. This industrial controller is based on an 650 MHz ARM processor with 256 MB SDRAM and was chosen due to its IP 67 rating and optional support for a dedicated and certified safety CPU module.

All code is written in ANSI C and uses the B&R Automation real-time operating system to communicate between the individual software modules and also interact with other software running on the controller at the same time (e.g. robot controller, data logging, ...).

[1]https://www.br- automation.com/de/produkte/steuerungssysteme/x90-mobile- steuerungssystem/x90-mobile-steuerung/x90cp17424-00/

3) Communication: To transmit data between the sensor board and the X90 controller can be achieved via two different modes: The first method uses a TCP connection between both partners. This allows the detection of a new connection, synchronization and automatic retransmission of lost or corrupted packets. On the other the second method uses UDP broadcasts. While this mode cannot correct any transmission errors, it is possible to provide the data to multiple participants simultaneously and not affect the overall network load.

B. Software Architecture

While the algorithm described in section III describes the core approach of geofencing, it cannot be used directly and additional preprocessing of the data needs to happen beforehand. The overall software architecture is split into multiple modules and their interaction depicted in Fig. 3.

C. IMU parser

The IMU parser module converts the incoming data stream from the sensor board into individual data packets. In Addition this module is responsible to filter corrupted packets and obvious measurement errors. Handling these errors and outliers in the very front of the processing pipeline helps to produce a overall smoother position estimation.

D. AVD module

The AVD (Acceleration - Velocity - Distance) module calculates speed and distance travelled based on the IMU data. As this process depends heavily on integration of measurements, of all modules it is the most susceptible to sporadic sensor errors. A special consideration has to be taken during the initialization of this module. As the whole software uses geo-referenced coordinates, the default value of (0,0) would let the system start somewhere in the Atlantic Ocean and would result in a gradual adjustment over approx. half an hour until the position is usable. To prevent this behavior, the AVD module is reset whenever a positional error of more then 500m is detected and re-initialized with the current measured GPS position.

E. Kalman filter

This module combines the measurement from the GPS sensor with the output from the AVD module to determine the current position of the robot and by extension to generate a POI for the geofencing point-in-polygon algorithm. The combination of both independent data sources enables the filtering of the random walk inherently present in every GNSS system, the compensation of temporary drift near building or large objects and the bridging of short events without GNSS information like driving under a bridge.

F. CoordTransform

The CoordTransform module translates all GPS positions from Cartesian coordinates into the UTM reference frame. This transformation is necessary as working in the nonlienear cartesian space would violate some mathematical assumptions of independence and by extension would break or at least impair the results of both AVD and kalman filter.

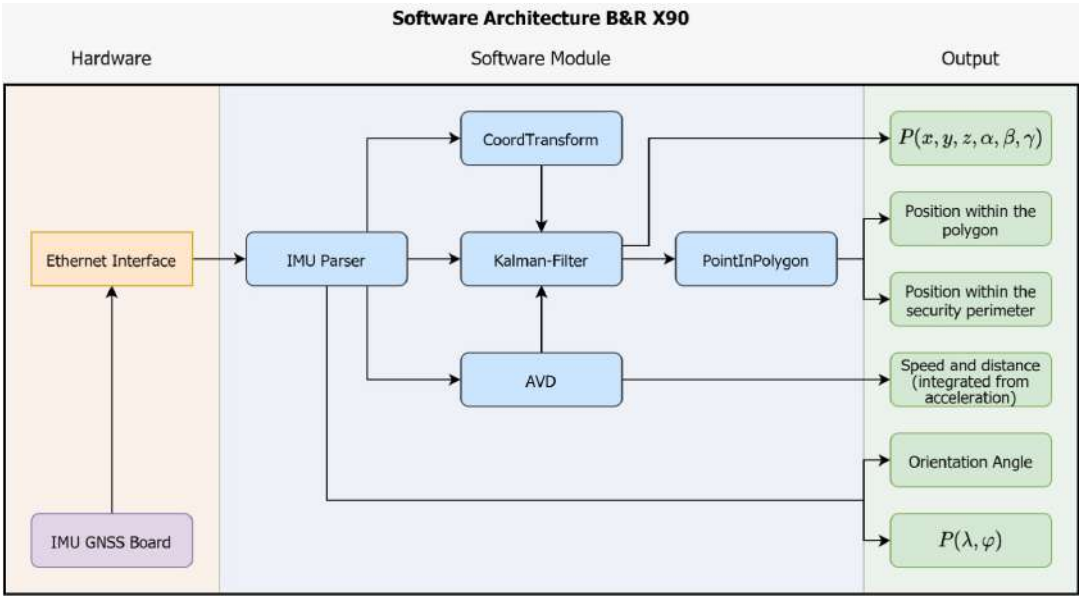


Fig. 3: Software architecture on the X90 controller

V. EXPERIMENTAL RESULTS

In this section, both lab and real-world test results are discussed. The detailed evaluation of all hardware and software components would exceed the scope of this paper, therefore we focus mainly on the newly developed point-in-polygon module.

A. Evaluation of algorithm and implementation

For the evaluation of the algorithm a black box testing approach was chosen. The X90 Controller is fed with synthetic data from a Matlab script and the results are checked automatically against the Matlab implementation of the algorithm and manually by visualizing the results as a plot, where the polygon is drawn in black, the POI in red, an the scaled polygon in green. In addition the nearest polygon feature in each plot is highlighted in purple.

For debugging purposes the values chosen for both polygon points and the POI do not use reasonable UTM coordinates, but an arbitrary reference frame around the point (250,250). However relative distances between points correspond to the scale expected in real-world scenarios. Examples of relevant test cases are listed below.

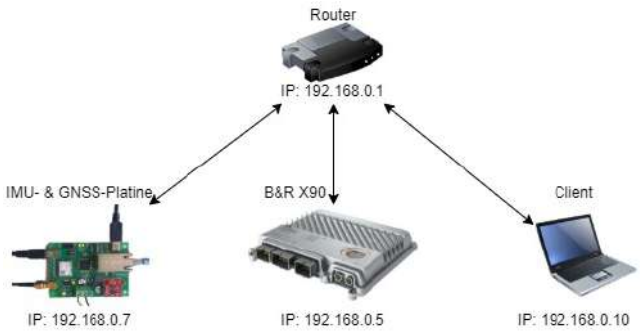


Fig. 4: Hardware and network configuration

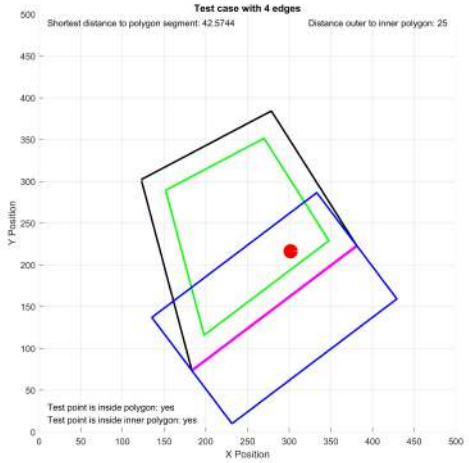


Fig. 5: Test against a polygon with four edges.

1) Simple polygon: The first test case checks the basic operation of the algorithm. In this case the number of polygon edges was limited to four. Fig. 5 shows the correct behaviour: The POI was evaluated both inside the original as well as the scaled polygon and the correct nearest feature was recognized, marked by the blue rotated bounding box.

2) Polygon scaling: In the next test case the main focus was, whether the implementation could handle a variable number of polygon edges and if the calculation of the scaled down polygon worked as expected. Therefore polygons with both convex and concave corners were generated. The result in Fig. 6 shows the correct scaling and expected behaviour.

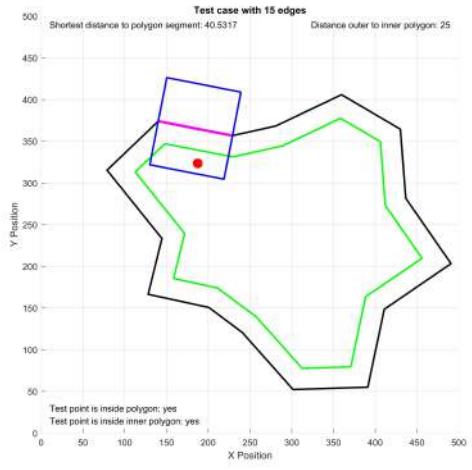


Fig. 6: Example of a scaling test case.

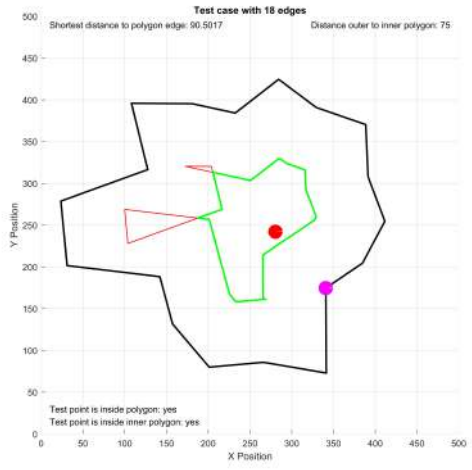


Fig. 7: Overlap test case

3) Overlaps and distance to corner: The final use case depicted in Fig. 7 depicts the evaluation of two functions. On the one hand the plot shows the nearest detected polygon feature to be a corner which is correct. On the other hand the distance for the scaled polygon was increased to force overlaps to occur. The algorithm filtered everything as expected, indicated by the sections marked in red being removed.

B. Field tests

To test the system under real-world conditions, all components were mounted on the mobile cattier platform depicted in Fig. 1 as part of the automation kit. A local and freely accessible meadow was chosen as a test location due to the combination of open field and some trees which may interfere with the GNSS signal. The perimeter polygon was created using coordinated taken from Google Maps and consists of the eight points listed in Table I.

TABLE I: Coordinates of the test area

Point	Coordinates in degrees		Coordinates in UTM (U33)	
	North	*East*	*North*	*East*
P0	48.22060	14.10073	5341210.375	433205.289
P1	48.22060	14.10053	5341210.549	433190.534
P2	48.22097	14.10053	5341251.674	443190.915
P3	48.22097	14.09995	5341252.178	443147.836
P4	48.22158	14.099450	5341320.414	433111.493
P5	48.22177	14.09972	5341327.472	433173.232
P6	48.22165	14.10028	5341312.353	433230.254
P7	48.22152	14.10105	5341312.353	433230.254

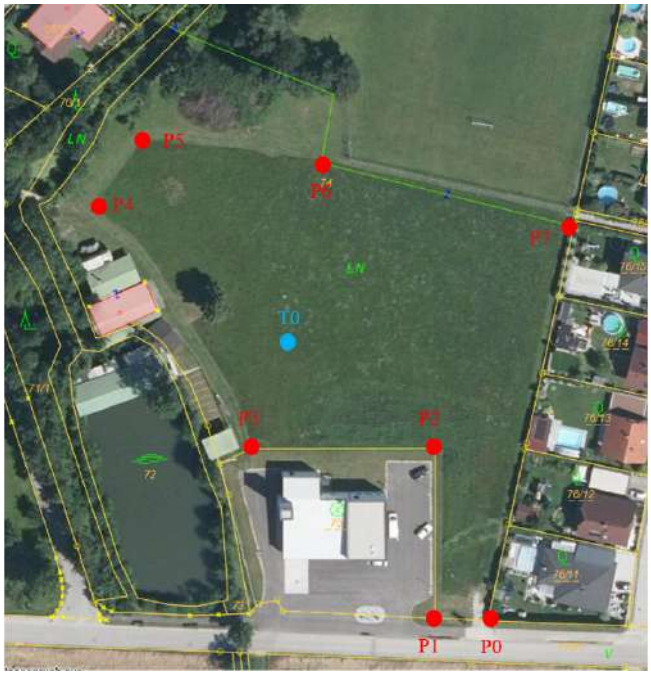


Fig. 8: Representation of the test area

The vehicle was placed on the start position T0 (see Fig. 8) and the autonomous operation was activated. However this resulted in an emergency stop due to the current position being outside the perimeter. The error could be traced back to the GNSS module measuring a position off by about 30 meter in reference to the actual location and therefore physically outside the perimeter. Forcing the correct GPS data by hand using a debugger, showed the geofencing to work as expected. However for a reliable and long term solution, the change to a DGPS system as positional reference will probably be necessary.

VI. SUMMARY AND OUTLOOK

In summary this work shows a software system for reliable geofencing running on an industrial grade controller provided the sensor measurements being correct. While not pursued at the moment, due to current lack of practical tests, the approach also shows potential for being moved to the safety CPU of the X90 controller to provide a functionally safe and certifiable module in the future.

ACKNOWLEDGMENT

The research of these results has been accomplished within the SMARTER - Slope Maintenance Automation using Real-Time Telecommunication and advanced Environment Recognition project. This work has been funded by the Austrian Research Promotion Agency (FFG) within the program "Mobility of the future" nr. 879646.

REFERENCES

[1] Q. M. Ilyas and M. Ahmad, "Smart Farming: An Enhanced Pursuit of Sustainable Remote Livestock Tracking and Geofencing Using IoT and GPRS," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–12, Dec. 2020. [Online]. Available: https://www.hindawi.com/journals/wcmc/2020/6660733/

[2] F. Koch, K. Lakkaraju, and F. Meneguzzi, Eds., *Agent Technology for Intelligent Mobile Services and Smart Societies: Workshop on Collaborative Agents, Research and Development, CARE 2014, and Workshop on Agents, Virtual Societies and Analytics, AVSA 2014, Held as Part of AAMAS 2014, Paris, France, May 5-9, 2014. Revised Selected Papers*, 1st ed., ser. Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg : Imprint: Springer, 2015, no. 498.

[3] S. K. Nagothu, "Automated toll collection system using gps and gprs," in *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 0651–0653.

[4] L. Narens, "A nonstandard proof of the Jordan curve theorem," *Pacific Journal of Mathematics*, vol. 36, no. 1, pp. 219–229, Jan. 1971. [Online]. Available: http://msp.org/pjm/1971/36-1/p20.xhtml

[5] R. R. Oliveira, I. M. Cardoso, J. L. Barbosa, C. A. da Costa, and M. P. Prado, "An intelligent model for logistics management based on geofencing algorithms and RFID technology," *Expert Systems with Applications*, vol. 42, no. 15-16, pp. 6082–6097, Sept. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417415002316

[6] R. Passarella, S. P. Raflesia, D. Lestarini, Taufiqurrahman, R. F. Malik, Sutarno, H. Ubaya, and A. Rifai, "Disaster mitigation management using geofencing in indonesia," in *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, 2017, pp. 1–4.

[7] F. Reclus and K. Drouard, "Geofencing for fleet amp; freight management," in *2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST)*, 2009, pp. 353–356.

[8] A. Suyama and U. Inoue, "Using geofencing for a disaster information system," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 2016, pp. 1–5.

Development of a Graph-Based Translation From BPMN to Executable Sequences for Industrial Robotic Systems

Christine Zeh[1], Katrin Clauss[1], Maximilian Papa[1,2] and Sebastian Schlund[1,2]

***Abstract*— The demands regarding high mix, low volume manufacturing and faster product life cycles make flexible production indispensable. Collaborative robots are widely regarded as an enabler for this flexible production. Further, they also achieve the goal of human-centered production envisioned by Industry 5.0. However, its installation requires significant efforts by skilled specialists and robotics experts for robot programming. In order to improve accessibility for shop-floor workers, the focus in previous works lies on the combination of graphical/function-based and declarative programming that supports fast reconfiguration. The business process model notation (BPMN) was used for the user input of hardware-independent robot skills. Further, the so-called SAMY-Core was developed to generate control commands for the respective hardware. Based on these two components, this work focuses on the concluding translation of BPMNs to machine executable processes as the final component within the SAMY framework to finalize its entire pipeline from the user input to the hardware-specific code execution. For the translation, the SAMY-BPMN is processed to a graph, which contains all skills and can return the correct following actions by means of states of the robot system. As a result, it is shown that this translation and thus the entire pipeline is feasible, allowing non-expert users to change the system both quickly and easily.**

I. Introduction

Increasing global competition, shorter product life cycles and individual customer requirements demand a high degree of flexibility in production [1]. Conventional production systems are proving to be too inflexible in this case, and thus, the digitalization of the components and intelligent automation (Industry 4.0) is seen as an enabler. While in the beginning complete automation was sought to achieve this flexibility, it quickly became clear that replacing the human is not considered viable [2]. Thus, human-centered production is established as one of the three pillars of the so-called Industry 5.0 by the European Commission [3].

Collaborative robots (cobots) are widely regarded as one of the enabler for these flexible production requirements [4], [5]. Unfortunately, the reconfiguration of a human-cobot workspace still bears various challenges. Robot experts are usually still needed for programming, and safety would also have to be determined again by safety experts after each adaptation [6]. However, today's shortage of skilled workers represents a significant challenge in this area [7].

The research project "SAMY" aims to address this simplification of accessibility by automating the modification process in control programs of industrial collaborative robotic systems. Preliminary work in this project found that the combination of graphical/function-based (e.g., moving and connecting blocks containing tasks like "pick&place" in a two-dimensional working space) and declarative descriptions (where the work systems' process chain is described with the mentioned blocks and not directly programmed in machine code) form the most effective and robust method for reconfiguring the work system from a user-centric perspective [8]. From these graphical descriptions, the specific robot code must be generated subsequently. As a result of this insight, the SAMY-Editor and the SAMY-Framework (SAMY-Core) [9] were created. In terms of user-centric programming and controlling robotic systems, the SAMY-Editor operates as the frontend using BPMN as the graphical user interface. Whereas the SAMY-Core acts as the backend to the robot for generating the machine code and the final control commands (see figure 1). However, both approaches, using BPMN as well as states and actions for processing, cannot be merged trivially. A middleware between the BPMN and the core is therefore needed. In addition, the translation must somehow process the BPMN, to provide the right work system sequence for generating the corresponding robotic control commands.

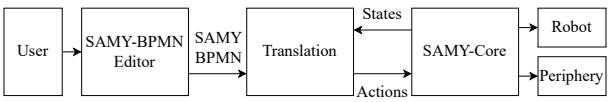


Fig. 1. Simplified framework for declarative programming and its translation to control commands for interaction with periphery

For this reason, this paper describes the process of getting from a simple graphical user notation to the machine-readable representation of it. This middleware is needed to ensure the whole SAMY pipeline (shown in figure 1) from the user input to the actual signal- and movement-conversation on the robotic system. For that, the processing of the graphically described states is needed to ensure a sequence of actions for controlling a robotic system. The development necessary for this, to make BPMNs executable, shows a new approach independent of SAMY.

*This project was funded by the Federal Ministry for Climate Protection, Environment and Energy, Innovation and Technology (BMK), and carried out within the framework of the programme "Production of Future" under the grant agreement number "877362" within the project "SAMY – Semi-Automated Modification in Control Programmes of Industrial Collaborative Robotic Systems".

[1] Fraunhofer Austria Research GmbH, Theresianumgasse 7, 1040 Vienna, Austria

[2] TU Wien, Institute for Management Science, Theresianumgasse 27, 1040 Vienna, Austria

II. State of the Art

Simplifying programming and configuration of industrial robots is a field of ongoing research, in particular by adding an abstraction layer [10]. The main idea of this abstraction is, that many tasks can be represented by a very small number of predefined skills (e.g., "move to <position>" and "pick up <work piece>"). Thus, these skills represent a hardware independent solution of programming a robot, where the available skills are defined by the robot's hardware and its sensors. Based on that, the challenge is combining standardized abstractions and getting from simple user notation to the machine-readable representation of it.

A. Abstraction Layers and Skills-based programming

Hoyos et al. [11] introduce a definition and management of skills, which can be accessed through some user interface. The ability to start one skill at a time, allows for a simple parsing of the abstraction to the machine readable code. The usage of Google Blockly [1] to simplify the programming by adding an abstraction layer is reflected upon in [12] and [13]. In this instance, the parsing of the robot application created through the abstraction layer is bypassed by directly mapping robot specific source code to the Blockly blocks. Thomas et al. [14] uses the Unified Modeling Language suitable for programming (UML/P) to combine the robot code with a graphical modelling language.

B. BPMN as abstraction layer

The use of BPMN as an abstraction layer to the implementation of skill-based industrial programming is a novel approach. Therefore, no existing research can be used as basis for the conversion and processing of the skills plan. In addition, the cited papers of the last section use their development on a specified hardware. Consequently, the translation to machine readable code is more straightforward.

Dijkman and Van Gorp [15] define rules to rewrite BPMN 2.0 to graphs in the tool GrGen. Raedts et al. [16] developed a translation between BPMN and Petri nets to verify and validate models. Further, a conversion between resilient BPMN and directed graphs was developed by Nordemann et al. [17], to use graph-based search algorithms on the BPMN.

A different approach is to directly simulate BPMNs. Pereira and Freitas [18] describe various tools to simulate BPMN. Pufahl et al. [19] introduce further an extensible BPMN discrete event simulator. In addition, the development of a BPMN extension to enable better discrete event simulation is described by Onggo et al. [20].

The conversion of BPMN to machine readable data structure is part of extended research. However, the transcriptions in the cited papers often use sub-types of BPMNs or specialized tools, which are not applicable in the case of robotic programming. Furthermore, the research in the cited papers is focused on economic problems.

III. Methodology

The targeted translation, from graphical descriptions to machine readable plans applies to the area of human-centered reconfiguration and modification, which is hardly tested in the field of robotic programming. Therefore, an iterative software process model with the focus on prototyping and reuse of software, like Boehm's [21] spiral model as methodology to design, build and maintain the concerning interface is applied. According to Alshamrani [22], the spiral model is suitable for developing highly customizable software due to iterative loops, the high amount of risk analysis, and the ability to react to rapid changes. Exemplary iterative loops for the development of the translation software are summarized in table I and table II.

TABLE I
Development of translation software: Iteration 1 (cf. [21])

Objectives	Translation concept for graphical description to machine readable plan
Constraints	BPMN, XML, Python, directed data structure
Alternatives	Petri net/graph, C++/Python
Risks	False description of data structure, states, or transition condition/high implementation effort
Risk resolution	Literature research
Risk resolution results	Development-, translation concept
Plan for next phase	Implementation of translation concept

Table I shows the translation concept for the graphical description to a machine-readable plan and includes the following standards: The BPMN (Business Process Model and Notation)[2] is a graphical notation for processes and represents the user interface for programming the robotic system. The BPMN acts as an input for the transformation to machine-readable plans. The BPMN standard is overlaid with its own SAMY-BPMN [23]. The transformation requires a directed data structure, for which a standard graph structure[3] is used as a connection component between the BPMN as user input and the SAMY-Core [9].

TABLE II
Development of translation software: Iteration 2 (cf. [21])

Objectives	Lossless representation of BPMN to graph (graph builder)
Constraints	BPMN, XML, graph, Python
Alternatives	Standard library BPMN to graph/develop particular translation script + NetworkX
Risks	High implementation effort, loss of information
Risk resolution	Internal discussion, literature research
Risk resolution results	Translation class from BPMN to graph with processing of states & actions
Plan for next phase	BPMN processing (graph planner)

[1] https://developers.google.com/blockly
[2] https://www.iso.org/standard/62652.html
[3] https://www.maths.ed.ac.uk/ v1ranick/papers/wilsongraph.pdf

Table II breaks down the first implementation of the translation concept involves the development of a graph builder, which includes the following standards: Storing the BPMN to XML (Extensible Markup Langage)[4], formatting it to DOM specification (Document Object Model)[5] via the library XML.DOM[6] and then processing the document via Python script to generate the graph is proceeded instead of using a standard library for building a graph, due to the non conventional overlay of the BPMN. For the management of the created graph the library NetworkX[7] is used. The result is a translation class from the BPMN to a graph with the processing of states and actions.

IV. Development

The translation and processing of BPMNs are parts of the connection component between the user interface and the backend core of SAMY.

Figure 2 shows the architecture of the referred connection component defined as controller, and the already implemented SAMY interfaces (SAMY-BPMN Editor, SAMY-Core). The implementation aspect of this work focuses on the controller, specifically on the development and integration of the so-called graph planner, as well as the mapper. A further implementation step is the incorporation of the already implemented interface to the SAMY-Core. As figure 2 shows, the graph planner and the mapping components are direct parts of the controller, which is invoked through the specified interface. The interface and arrows illustrate definitions for the transition of the already developed SAMY parts to the controller. These definitions, namely the use of the SAMY-BPMN on the one hand and the use of states and actions on the other hand, are the constraints for this development. The task can therefore be described as follows: A BPMN is loaded and for each discrete event step, the program is aware of the current action to be executed and whether the present state meets the transition conditions for moving on to the next action. In the development phase only the graph planner is of interest, since the aim of the mapping, described in greater detail in the next section, is the simple parsing of the SAMY specific state and action definitions.

[4]https://www.w3.org/standards/xml/core
[5]https://www.w3.org/TR/WD-DOM/introduction.html
[6]https://docs.python.org/3/library/xml.dom.html
[7]https://pypi.org/project/networkx/

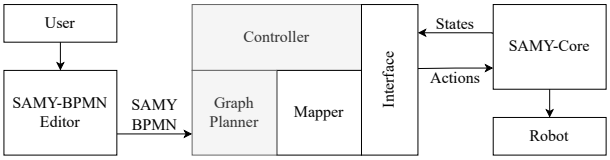


Fig. 2. Architecture of the controller and the surrounding SAMY interfaces

A. Approach for developing the middleware

The approach for the implementation of the two main parts, namely the BPMN transition and the BPMN processing, is dependent on two questions:

- BPMN Translation: How can the BPMN be stored without loss of information?
- BPMN Processing: How can the translation be simulated?

SAMY-BPMN only uses flow objects and connecting objects, which are used to describe a flow of connected activities. Since a flow can contain junctions and loops, a graph-based data structure is necessary to gain a lossless representation. The decision was made for a directed graph, which is processed further. Gateways and variable manipulation tables (VMT) are removed as nodes, to obtain a representation containing only the action-based nodes. The removed information is integrated as internal functionality of the nodes and edges. The nodes fulfil the task of managing the internal variables and returning its action. The edges are opened after checking the received state and the internal variable container. An exemplary depiction of the conversion can be seen in figure 3.

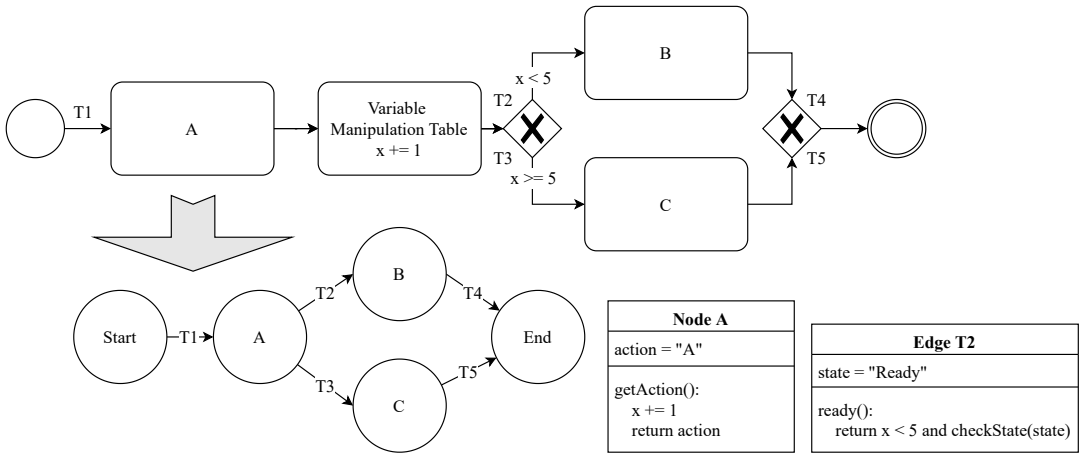


Fig. 3. An exemplary depiction of the conversion between BPMN and directed Graphs with its internal class functionality. The internal functionality of the graph objects is described in pseudo code.

The logic of the process is stored in the objects of the graph, which leads to easy processing of the graph. The initial current node is the start process. Then, at each event, every originating edge of the current node is provided with the current state list. After processing the current state list, the referring edge returns whether the requirements are met for moving on to the next node. Only if the corresponding edge is open, can the next node be reached. If the current node moves forward to the next node, the action request is invoked for this node, which also updates the variable container. As a result, a corresponding skill (e.g., pick&place) is executed on the robotic system.

B. BPMN Translation

A BPMN model is stored as an XML file, which can be processed using a DOM. This is used to store each SAMY-BPMN block type as node and each transition as edge. The result is a function-less graph with the correct connections of the BPMN. The nodes and edges both contain an object which will handle the additional functionality added by gateways and VMTs. Furthermore, it stores the action type of the nodes and the transition state of the edges. As a next step, each gateway and the VMT has to be reworked to move the functionality into the process nodes. After each transformation, the redundant gateway nodes are removed from the graph. This is realized on the basis of the following rules:

- The VMT is stored in a class and the variables are updated when the predecessor is visited within the processing step.
- The conditions for the exclusive gateway are stored in a list of tuples which get checked to open the edges.
- The combination of edges closing the parallel gateway is stored for each involved edge, to ensure each transition is opened, only if all parallel edges are open.
- The loopback gateway does not need extra functionality, therefore solely the predecessor and successor are linked.

C. BPMN Processing

The BPMN contains three blocks which are of special importance for the processing. The start block defines the initialization point of the processing, the end block defines when the processing is finished and the variable container includes all variables needed for the processing with their initial values.

With the start node set as the current node, the process checks on each state event, if the edges are open. If this is the case, the current node moves forward, updates the variable container and returns the action name. Due to parallel gateways, multiple current nodes are possible, which have to be managed collectively. To close a parallel gateway, the list of the parallel edges involved is compared to all linked open edges of current nodes. If they are a subset, the associated current nodes can move forward and merge.

D. Integration in SAMY

The core is limited regarding the user friendliness of describing the actions and states. Therefore, a separate definition for the creation of the BPMNs can be chosen, to facilitate the description of the actions and states. This definition must then be linked to the core actions by a mapping file. The mapping is stored as a YAML file, which assigns a SAMY value to each controller specific value. The mapper can access this file and then map the BPMN actions to the SAMY actions ”on the fly”.

Since there are no transition states within the BPMN, own states must be specified based on the resources of the previous node. For example, a resource could be the robot itself or integrated sensors and actuators. Each resource is defined with its state ”Resource:Ready”, to clarify whether a resource is idle. The state description ”Ready” is arbitrary and could be anything as long as it is used within the mappings. These mappings are content of the YAML file, which stores the correct translation to the SAMY states.

During a triggered state event the mapper is executed twice (see figure 4). Firstly, it is executed at the beginning to parse

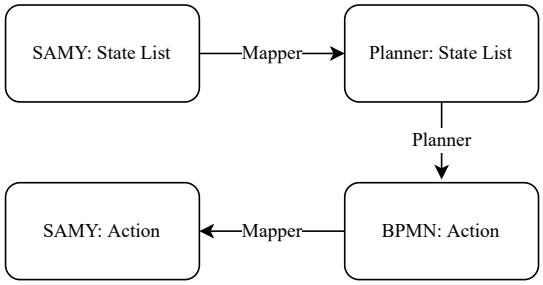


Fig. 4. Mapping pipeline from the SAMY state list to resulting SAMY action.

the SAMY state list to the defined states of the graph planner (planner state list). The graph planner can then work with the mapped states and the actions defined through the BPMN. Secondly, the mapper is executed at the end to converts the resulting BPMN actions to the SAMY actions.

V. Results and Discussion

In order to evaluate the developed translation and mapping from graphical descriptions to machine readable plans, an experimental simulation-based use case was generated for testing the developed controller and the holistic SAMY pipeline. Executing the graphically described process within an exemplary use-case shows the translation from declarative programming and reconfiguration to skill-based processing of work tasks, for generating robotic applications.

The BPMN of the realized use-case is shown in figure 5. The robotic task is to sort five objects into two boxes, with the use of a camera to detect the objects. Overall, ten objects should be moved, after this, the program stops.

The individual blocks in the BPMN are the robotic tasks (e.g., Robot:move pick pose) to be implemented. Those tasks are described as skills within the SAMY context. Loops and branches defined in the BPMN describe the logic of the entire

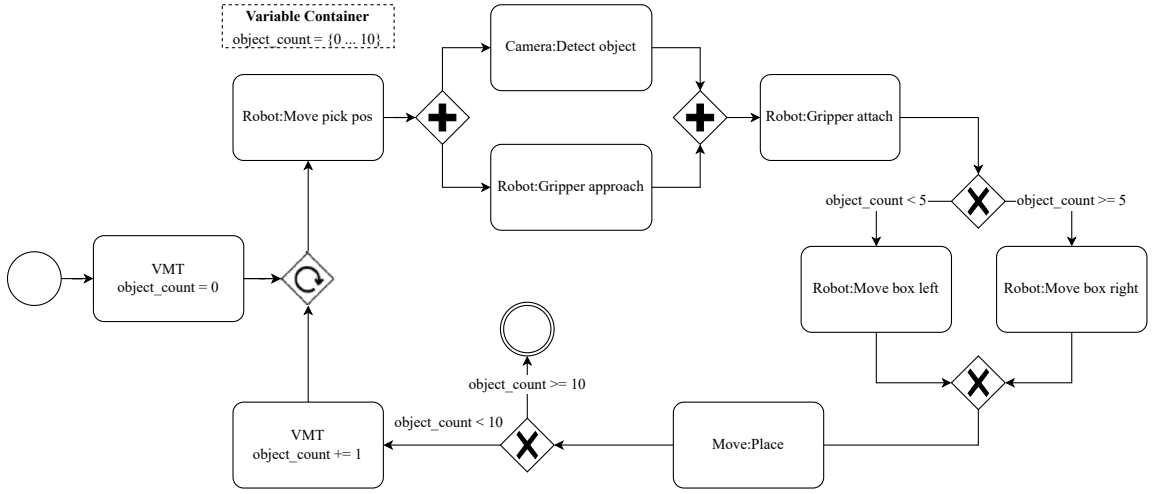


Fig. 5. SAMY-BPMN of exemplary use-case for picking and sorting objects

robotic process. For a logical flow, additional variables are needed, which are defined in the variable container with their initial values. In this use-case such an additional variable is utilized to manage the number of objects already sorted (e.g., object_count < 5). The increasing of the variable is executed by the VMT.

A direct comparison of the BPMN (Figure 5) and the graph (Figure 6) shows that on the one hand the skills still exist as nodes, but on the other hand loop gateways are no longer depicted in the graph. The visualization of exclusive gateways is also omitted and instead represented as logical conditions within the edges. Parallel gateways only appear as conditions for the parallel edges that are converging to the same node. An exemplary progression in the graph from the ”Start” node to the ”Robot:Move pick pos” node is depicted in Figure 7, which corresponds to a controller call triggered by a state change. The first block ”[Robot_RobotUR5_CRCLStatus=1, InformationSource_Camera_Status=0]” describes the physical robotic system, based on a SAMY state list. In this case, unlike the camera, the robot is idle and is waiting for the next instruction. The mapper translates the status of the hardware to a format readable by the planner. The planner checks the received state against the existing graph and returns the action of the next node ”Robot:Move pick pos” if the states match. Lastly the returned action is mapped back as SAMY action ”Move-UR5-Pick” to be processed by the SAMY-Core.

The result of the experimental simulation-based use case is the correct traversing of the graph from start to finish. All states and transition constraints are processed and a SAMY skill is invoked at each node transition, leading to a sequence of skills that controls the hardware and executes the corresponding robotic process. Thus, the feasibility of the simplified reconfiguration of a work system using the SAMY pipeline is demonstrated, where the work system can now be easily modified by drag & drop of the individual BPMN blocks of the process chain (seen in figure 5).

VI. Conclusion and Future Work

Market demands like high mix, low volume manufacturing and faster product life cycles will require a more flexible production system (e.g., human-cobot work system).

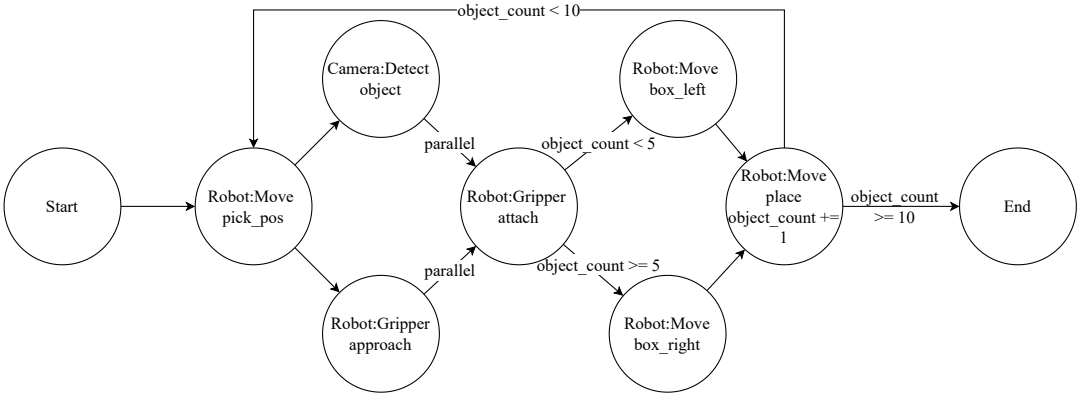


Fig. 6. SAMY-BPMN translation into directed graph of exemplary use-case for picking and sorting objects. For easier readability, the class functionality is written to the corresponding edge or node.

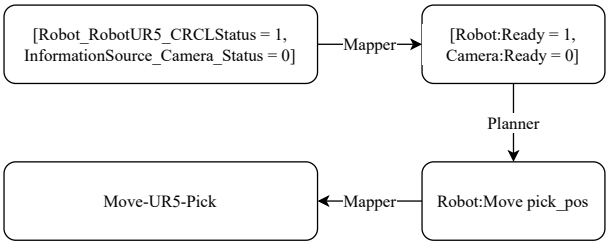


Fig. 7. The skills in the BPMN are mapped to SAMY skills in the mapper.

However, the reconfiguration of such systems is still too complicated and robotics experts are needed. Therefore, SAMY focuses on the simple programming and reconfiguration of these work systems. A (non-) expert creates a simple graphical/function-based SAMY-BPMN as declarative programming and the SAMY-Core creates machine-readable code from it. As a contribution to SAMY, this paper shows the development of the missing connection component between the SAMY-BPMN and the SAMY-Core. The developed graph planner generates a graph containing all skills from the SAMY-BPMN and its sequence. Further, the controller maps the correct skills from the BPMN to the SAMY specific skills, knowing the used hardware. Finally, the interface to the SAMY-Core is done by giving it the actions in the right sequence according to the given states of the robot system. The entire SAMY pipeline was evaluated by an experimental simulation-based use case, which showed that a translation from hardware-independent SAMY-BPMN to specific robot skills was done, meaning that a non-expert could easily reconfigure the work system using the graphical SAMY-BPMN.

Since the use case is solely simulation-based, for further in depth evaluation the experimental use case is implemented in an industrial cobot-application, testing the translation as part of the holistic SAMY pipeline and the benefiting ease of reconfiguration in real-world conditions. In addition, the translation of SAMY-BPMN could be generalized to standard BPMN to use the presented approach in a variety of different applications. Thus, BPMNs that are not cobot-specific could be translated into executable graphs using this method by adapting the SAMY-specific classes. In addition, the presented translator can be made more intelligent (e.g., states as exclusive gateways or swimlanes), so that also more complex logics can be built.

REFERENCES

[1] I. Hanschke, *Digitalisierung und Industrie 4.0 - einfach und effektiv: Systematisch und lean die Digitale Transformation meistern*. Carl Hanser Verlag GmbH & Co. KG, 2019.

[2] A. Kolbeinsson, E. Lagerstedt, and J. Lindblom, "Foundation for a classification of collaboration levels for human-robot cooperation in manufacturing," *Prod. Manuf. Res.*, vol. 7, no. 1, pp. 448–471, 2019.

[3] European Commission, "Industry 5.0." `https://ec.europa.eu/info/research-and-innovation/research-area/industrial-research-and-innovation/industry-50_en` (Accessed: 02.02.2022), 2020.

[4] A. Weiss, A.-K. Wortmeier, and B. Kubicek, "Cobots in industry 4.0: A roadmap for future practice studies on human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 4, pp. 335–345, 2021.

[5] C. Schmidbauer, S. Schlund, T. B. Ionescu, and B. Hader, "Adaptive task sharing in human-robot interaction in assembly," in *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 546–550, 2020.

[6] C. Fischer, M. Steiner, M. Neuhold, M. Papa, A. Markis, and S. Schlund, "An investigation of the measurement of transient contacts in human-robot interaction," in *Advances in Service and Industrial Robotics*, pp. 547–555, Springer International Publishing, 2022.

[7] International Federation of Robotics, "Next generation skills-enabling today's and tomorrow's workforce to benefit from automation," *International Federation of Robotics (Positioning Paper)*, November 2020.

[8] T. Komenda, J. B. Garcia, M. Schelle, F. Leber, and M. Brandstötter, "Sustainable utilization of industrial robotic systems by facilitating programming through a human and process centred declarative approach," *International Conference on Competitive Manufacturing (COMA)*, 2022.

[9] J. B. Gracia, F. Leber, M. Aburaia, and W. Wöber, "A configurable skill oriented architecture based on opc ua (in submission)," *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[10] S. Bøgh, O. S. Nielsen, M. R. Pedersen, V. Krüger, and O. Madsen, "Does your robot have skills," in *Proceedings of the 43rd International Symposium on Robotics*, 2012.

[11] J. Hoyos, A. B. Junaid, M. Raheel Afzal, A. Tirmizi, and P. Leconte, "Skill-based easy programming interface for industrial applications," in *2022 IEEE/SICE International Symposium on System Integration (SII)*, pp. 210–217, IEEE, 2022.

[12] M. Winterer, C. Salomon, J. Koberle, R. Ramler, and M. Schittengruber, "An expert review on the applicability of blockly for industrial robot programming," in *25th International Conference on Emerging Technologies and Factory Automation*, pp. 1231–1234, 2020.

[13] D. Weintrop, D. C. Shepherd, P. Francis, and D. Franklin, "Blockly goes to work: Block-based programming for industrial robots," in *2017 IEEE Blocks and Beyond Workshop*, pp. 29–36, 2017.

[14] U. Thomas, G. Hirzinger, B. Rumpe, C. Schulze, and A. Wortmann, "A new skill based robot programming language using uml/p statecharts," in *2013 IEEE International Conference on Robotics and Automation*, pp. 461–466, IEEE, 2013.

[15] R. Dijkman and P. Van Gorp, "BPMN 2.0 execution semantics formalized as graph rewrite rules," *Lecture Notes in Business Information Processing*, vol. 67 LNBIP, pp. 16–30, 2010.

[16] I. Raedts, M. Petkovic, Y. S. Usenko, J. M. E. Van der Werf, and J. F. Groote, "Transformation of BPMN Models for Behaviour Analysis," in *Proceedings of the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems*, pp. 126–137, Science and and Technology Publications, 2007.

[17] F. Nordemann, R. Tönjes, E. Pulvermüller, and H. Tapken, "Resilient process modeling and execution using process graphs," vol. 1375 of *Communications in Computer and Information Science*, pp. 3–23, Cham: Springer International Publishing, 2021.

[18] J. Pereira, A. Freitas, M. Teixeira, A. Correia, H. Adeli, A. Rocha, and L. Reis, "Simulation of BPMN process models: Current BPM tools capabilities," *Advances in Intelligent Systems and Computing*, vol. 444, pp. 557–566, 2016.

[19] L. Pufahl, T. Wong, M. Weske, E. Teniente, and M. Weidlich, "Design of an extensible BPMN process simulator," *Lecture Notes in Business Information Processing*, vol. 308, pp. 782–795, 2018.

[20] B. Onggo, N. Proudlove, S. D'Ambrogio, A. Calabrese, S. Bisogno, and N. Levialdi Ghiron, "A BPMN extension to support discrete-event simulation for healthcare applications: An explicit representation of queues, attributes and data-driven decision points," *Journal of the Operational Research Society*, vol. 69, no. 5, pp. 788–802, 2018.

[21] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, pp. 61–72, May 1988.

[22] A. Alshamrani and A. Bahattab, "A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model," *IJCSI International Journal of Computer Science Issues*, vol. 12, pp. 106–111, 2015.

[23] T. Haspl, M. Rathmair, M. Papa, M. Hofbaur, and A. M. Tonello, "Software toolchain for modeling and transforming robotic workflows into formally verifiable model representations," in *Austrian Robotics Workshop 2022*, 2022. in press.

Deep Reinforcement Learning for Continuous Robot Trajectory Control*

Simon Schwaiger[1], Mohamed Aburaia[1], Lucas Muster[1], Moritz Abdank[1] and Wilfried Wöber[2]

Abstract— **Trajectory control is fundamental to any robot application. Probabilistic path planning aims to solve trajectory control without focusing on a specific robot type. Similarly, reinforcement learning has been applied to robot control tasks with the goal of having problems of different nature be solved by the same agent. However, tested reinforcement learning models directly generate actuator control signals from sensor input. Due to the end-to-end nature of proposed implementations, trajectory and closed loop control are performed by the same model. We want to increase modularity in reinforcement learning-based robot control pipelines by solving trajectory and closed loop control separately from each other. Therefore, we formulate a continuous robot trajectory control problem as a reinforcement learning environment and evaluate agent performance for multiple environment configurations. Using this problem formulation, an agent is able to learn robot kinematics in simulation and determine joint trajectories. We deploy multi-goal reinforcement learning in order to allow agents to plan movement from an arbitrary start pose to an arbitrary goal pose without requiring a dedicated training procedure. The problem formulation is evaluated by training recent for continuous control and documenting agent performance. The results show, that continuous trajectory control can be achieved by the presented methods, allowing an agent to learn the kinematics of a six degree of freedom robot.**

I. INTRODUCTION

Trajectory planning is required for any robot in order to be able to perform tasks. However, the variety of possible robot deployments creates the need for flexible, modular and fast trajectory planning algorithms [13]. While reinforcement learning (RL), a sub-domain of artificial intelligence [9], has successfully been used to automate these tasks [1] [11] [10], implementations are mostly done in an end-to-end manner, resulting in systems that directly control actuator torque based on sensor input. The resulting RL model performs trajectory as well as closed loop control, creating solutions whose internal functionality is not observable. In order to increase modularity of RL-based solutions in robotics, we aim to solve trajectory control and closed loop actuator control independently from each other.

For this paper, a novel formulation of robot trajectory control is presented and evaluated using state of the art RL agent configurations. The problem is exposed as a RL environment with continuous observation and action-spaces, deploying multi-goal RL [5], in order to enable an agent to be trained to plan robot movement from an arbitrary start to an arbitrary goal point.

*This work was supported by the city of Vienna (MA23 – Economic Affairs, Labour and Statistics) through the research project AIAV (MA23 project 26-04)

[1]Simon Schwaiger, Mohamed Aburaia, Lucas Muster and Moritz Abdank are with the Department of Industrial Engineering, University of Applied Sciences Technikum Wien, 1200 Vienna, Austria. Correspondence E-Mail: schwaige@technikum-wien.at

[2]Wilfried Wöber is with the Department of Industrial Engineering, University of Applied Sciences Technikum Wien, 1200 Vienna, Austria and the Institute for Integrative Nature Conservation Research, University of Natural Resources and Life Sciences, 1180 Vienna, Austria

II. STATE OF THE ART

The main goal of RL is for an agent to learn from interaction with its environment [9]. The agent receives an observation depicting the current state of the environment, called *state* and determines an *action* to perform in the state. This action is taken and the environment returns a *new state*, resulting from the previous state and action. Alongside the new state, a numerical reward is provided, that indicates agent performance in achieving a task. A positive reward indicates success, while a negative reward indicates failure [12].

[11] implemented trajectory control of a three degree of freedom (DOF) robot using a deep Q-network (DQN) [15] based agent, deploying hindsight experience replay (HER) [1] and reward shaping [6]. To avoid a distinct training sequence for each movement goal, multi-goal RL [5] was implemented by including information about the current goal as part of the state. However, due to using DQN, the resulting system from [11] is only capable of discrete control.

This study contributes to the state of art by formulating the action-space in a continuous manner and expanding the system with compatibility for six DOF robots.

III. MATERIALS AND METHODS

In order to achieve continuous control, deep deterministic policy gradients (DDPG) [4] as well as asynchronous advantage actor-critic (A3C) [16] based agents were deployed due to their compatibility with continuous action spaces. DDPG and A3C are both actor-critic agents that approximate the action-value function using an artificial neural network [9], while still allowing for a continuous and high-dimensional action-space due to having a separate actor network mapping states to actions. While DDPG performs updates based on the sampled policy gradient from encountered transitions, A3C performs gradient updates in the forward view after episode termination or a set number of steps. Since a linear increase in robot DOF results in an exponential increase of observation-space size, HER is applied to compatible agents in order to ensure that the agent will encounter positive reward. HER generates alternative goals for episodes producing no reward, that are used to perform extra training episodes apart from the regular ones [1].

The trajectory control problem is implemented as a custom environment for OpenAI Gym [2], using the robot operating system (ROS) [7] for simulation of robot kinematics. The DDPG, A3C, DQN and HER implementations from [8] provide the underlying algorithms of tested agents, with tests being conducted using a six DOF UR5 robot arm [14]. We formulate the state S as a list containing the six current joint states $\vec{J}$, a normalised vector pointing from robot tool centre point (TCP) to the goal $\hat{P}_{TCP-Goal}$ as well as distance to the goal in metres $|\vec{P}_{TCP-Goal}|$, in a similar manner to [11]. Additionally, we also include a normalised vector containing euler angles $(\alpha,\beta,\gamma)_{TCP-Goal})^T$, denoting the change in rotation between TCP pose and goal pose as well as the total angle $\varepsilon = \alpha+\beta+\gamma$ between the two poses in radians. Actions A are provided in form of a list containing joint position change for each joint in radians and reward R is given based on the robot reaching a goal or colliding. State, action and reward formulations are depicted (1) to (3).

$$S = (\vec{J}, \hat{P}_{TCP-Goal}, |\vec{P}_{TCP-Goal}|, (\alpha,\beta,\gamma)_{TCP-Goal}, \varepsilon)^T \quad (1)$$

$$A = \Delta\vec{J} \quad (2)$$

$$R = \begin{cases} 1 & \text{if } |\vec{P}_{TCP-Goal}| <= 0.001 \text{ and } \varepsilon <= 0.01 \\ -1 & \text{if collision} \\ 0 & \text{else} \end{cases} \quad (3)$$

IV. EXPERIMENTAL RESULTS

The system was tested on a simulated six DOF industrial robot, comparing performance between DDPG, A3C and DQN-based agents, while having HER applied to DQN and DDPG. The DQN agent operates in a with 0.1 radians discretised action-space. All tests have been conducted five times with randomised seeds as suggested in [3], plotting the mean return per episode with a 95% confidence interval. Each agent was trained for 200000 steps, with episodes being terminated when a goal was reached, a collision occurred or when 70 steps were taken (see Fig. 1). Additionally, ten testing episodes have been conducted after each training sequence, resulting in a mean reward of -0.08 for DDPG, 0 for A3C and -0.52 for DQN.

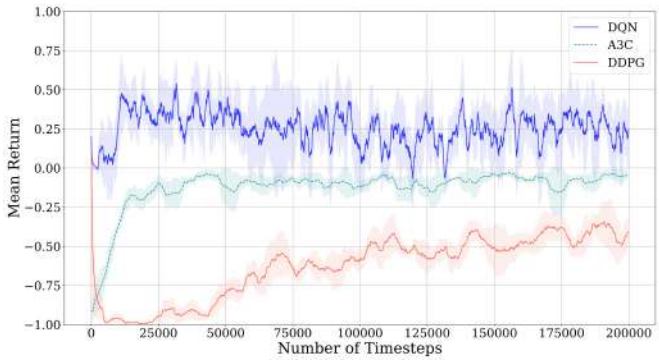


Fig. 1. Agents have been trained for 200000 time steps in the presented environment. The action-space was discretised in order to enable compatibility with the DQN agent.

V. SUMMARY AND OUTLOOK

In this paper, the formulation of continuous robot trajectory control as a RL problem was evaluated using recent RL agents. The results show, that while tested agents are capable of performing continuous robot trajectory control, success of the presented solution is highly dependent on agent's internal randomised initial parameters, due to agents encountering divergence during training. Further work must be done to increase stability of the presented formulation in order to enable deployment in a production environment.

REFERENCES

[1] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 5055–5065.

[2] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym. *arXiv preprint arXiv:1606.01540*," 2016. [Online]. Available: https://arxiv.org/pdf/1606.01540.pdf

[3] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11694

[4] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*," 2019. [Online]. Available: https://arxiv.org/pdf/1509.02971.pdf

[5] M. Plappert et al., "Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*," 2018. [Online]. Available: https://arxiv.org/pdf/1802.09464.pdf

[6] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proceedings of the Sixteenth International Conference on Machine Learning*, vol. 99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 278–287.

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

[8] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[9] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Englewood, NJ, USA: Pearson, 1995.

[10] S. Schwaiger, A. Aburaia, M. Aburaia, and W. Wöber, "Explainable artificial intelligence for robot arm control," in *Proceedings of the 32nd DAAAM International Symposium*, B. Katalinic, Ed. Vienna, Austria: DAAAM International, 2021, pp. 0640–0647.

[11] ——, "Open loop robot control using deep Q-learning," in *Proceedings of the Austrian Robotics Workshop 2021*, W. Kubinger, M. Brandstötter, C. Schöffmann, and M. Vincze, Eds., Vienna, Austria, 2021, pp. 12–17.

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: Bradford Books, 2018.

[13] P. Švestka and M. H. Overmars, "Probabilistic path planning," in *Robot Motion Planning and Control*, J. P. Laumond, Ed. Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 1998, pp. 255–304.

[14] Universal Robots A/S. (2022) Ur5 collaborative robot arm. [Online]. Available: https://www.universal-robots.com/products/ur5-robot/

[15] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[16] ——, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. PMLR, 2016, pp. 1928–1937.

Collision Avoidance using Capacitive Proximity Sensors on Series Elastic Robots for Human-Robot Collaboration

Serkan Ergun[1], Vinayak Hande[2], Johannes Sturm[2] and Hubert Zangl[1]

***Abstract*— In order to allow human beings to co-operate with robots, strict measures are required to avoid the risk of injuries. Capacitive Proximity Sensors embedded in robotic skins offer a cost-effective solution for the close-range perception of the robots' surroundings, filling the open gap between tactile sensing and mid-range perception in collaborative work cells. Recently, series elastic robots have seen increased interest for such applications, as elastic components can act as a passive mechanical energy storage in case of contact, reducing the potential impact on the human co-worker. In this work we investigate the applicability of Capacitive Proximity Sensors in a collision-avoidance scenario using a series elastic robotic arm. The robo's motion is assessed together with the capacitive output of the Capacitive Proximity Sensor. Our sensor is driven by a novel closed-loop class D amplifier fully on-chip driver circuit with small space requirements, allowing it to be directly integrated into robotic skins.**

I. INTRODUCTION

Human-Robot Collaboration (HRC) aims to eliminate physical barriers between robots and human workers to allow collaboration between humans and robots for better productivity [1]. Precautions need to be undertaken in order to avoid undesired contacts, which may lead to injuries of the human co-worker. The currently valid ISO/TS 15066:2016 [2] provides guidelines for the design of such collaborative workspaces. The specification supplements the standards for industrial robots ISO 10218-1:2011 [3] and ISO 10218-2:2011[4] with respect to HRC. ISO/TS 15066 introduces four operation modes for HRC, whereas only two operation modes, namely Speed and Separation Monitoring (SSM) and Power and force limiting (PFL) allow the human being to work in a fence-less environment alongside an active robot. SSM requires the robot to maintain a safe distance to the human being while operating, allowing it to stop ahead of potential contact. Power and force limiting (PFL), on the other hand, allows actual physical interaction if the forces exerted by the robot on the human being do not violate critical values. Values for permissible contact pressures and forces are given in e.g., Annex A of ISO/TS 15066 [2].

Most commonly, collaborative robots (cobots) rely on design elements, such as reducing sharp edges and corners to increase the contact surface. Another prominent solution is the usage of integrated force and torque sensing. Recently, an air-filled pillow-like soft robotic skin with integrated pressure sensing has been developed [5] and can be commonly found in collaborative workspaces.

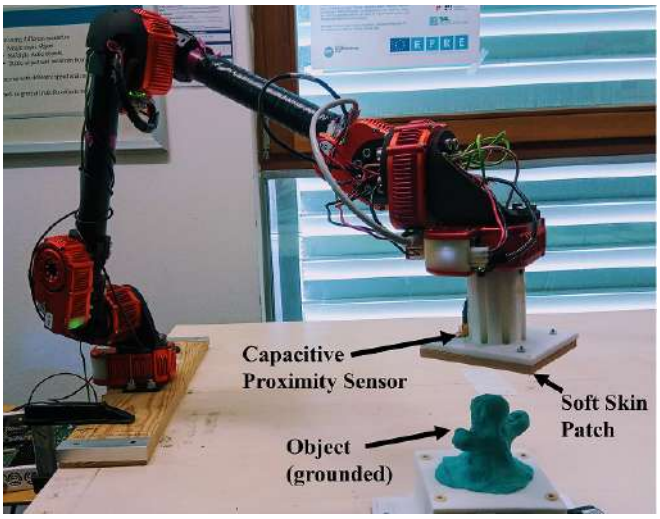


Fig. 1. Potential use case for Capacitive Proximity Sensors (CPSs) in robotics: A (CPS) is embedded in a soft skin patch at the end-effector of a series elastic robotic arm. The CPS triggers the motion stop mode of the robot prior to a potential collision. The object is grounded to yield a signal level close to human beings.

Alternatively, series elastic robots are also a considerable option as they allow to adjust their stiffness [6]. As opposed to traditional (rigid) robots, series elastic robots can store some portion of the collision energy due to their elastic nature. These aforementioned methods have in common that information (such as a change in force, torque, or electrical current) is only provided after contact. Nevertheless, such contacts should be avoided at all times to ensure workplace safety and the well-being of human co-workers. Possible solutions to overcome these tackles are external time-of-flight based sensors, such as acoustic [7][8], visual [9], or radar [10] based sensors, which may incorporate the opportunity for human detection, tracking, and classification of human co-workers in the vicinity of cobots [11]. These aforementioned sensing technologies perform well for sensing ranges of typically up to 5m which is suitable for HRC. However, typical minimal sensing ranges start from 10cm. This gap in perception can be filled with the aid of Capacitive Proximity Sensors (CPSs), allowing the robot to operate at a speed above the limit set for pure PFL. This work showcases applying CPS for collision avoidance scenarios using a series elastic robotic arm. The CPS, which can be embedded in a soft skin covering the robots' surface, initiates a motion

[1]The authors are with the University of Klagenfurt (AAU) - Institute for Smart Systems Technologies, Klagenfurt am Wörthersee, Austria, {serkan.ergun,hubert.zangl}@aau.at

[2]The authors are with the Carinthian University for Applied Sciences (CUAS) - Integrated Systems and Circuits Design, Villach, Austria, {v.hande,j.sturm}@fh-kaernten.at

stop procedure once an object or human is within a critical distance to the robot, as seen in Fig. 1. Such a rapid motion stop procedure results in brief oscillations. The robots' pose and the sensor output are recorded for evaluation and can be assessed for the potential use of range to capacitance mapping. The CPS is driven by a novel closed-loop class D amplifier fully on-chip driver circuit [12].
The following part of this paper is structured as follows: Section II lists relevant work, and section III shows the theoretical benefit of CPSs. The experimental setup for the sensor kit and collision avoidance scenario are elaborated in section IV. The results are showcased in section V and elaborated in section VI. A summary and brief outlook are given in section VII.

II. Related Work

In the last years, impact attenuating actions to prevent injuries in HRC has seen a boost in global research interest. Zanchettin et al. [13] demonstrated the use of 3D cameras and reactive motion generation to maximize task velocity while still maintaining braking distance to obstacles. Svarny et al. [14] has emphasized a scheme which combines SSM for larger distances and PFL at close distances to increase the performance of collaborative work cells, which are otherwise limited due to safety limits. Short-range perception achieved with CPSs enhances collaboration in close proximity between humans and robots in the PFL regime. The programmed maximal robot speed for an application can then be increased since the robot can slow down before contact and thus reduce the collision impact, as shown in [15]. Capacitive skins have been proposed as a means for provoking safety-rated stops by [16]. The authors of[17] developed the concept of CPS-based virtual whiskers aiming to mimic the physical whiskers of mammals. The usage of CPS is not limited to serial manipulators but was also used as a complementary sensor system on a mobile manipulator by [18].
The CPS on-chip driver circuit is commonly developed by a linear amplifier, such as Class A or Class AB topologies, as discussed by [19] and [20]. However, the on-chip driver must be able to drive a huge off-chip parasitic capacitance from the active guard to the ground i.e., approximately 300pF, with the optimal linear sinusoidal excitation signal and energy efficiency. Therefore to resolve the issues of optimum linearity and power efficiency, the concept of buck converters is employed. This benefits in high efficiency, low power consumption, and tunable frequency of a generated signal.

III. Theory

According to [15], the maximum permissible operating speed of a robot at a given spot on its surface $v_{\mathrm{R,max}}$ can be calculated by

$$v_{\mathrm{R,max}} = -\ddot{x}(t_{\mathrm{S}}+t_{\mathrm{R}}) + \sqrt{\ddot{x}^2(t_{\mathrm{S}}+t_{\mathrm{R}})^2 + v_{\mathrm{rel,max}}^2 + 2s_{\mathrm{S}}\ddot{x}}, \quad (1)$$

where $\ddot{x}$ is the smallest Cartesian deceleration for a given point on the robot within a given trajectory. It can be determined theoretically by applying robot dynamics [21] or empirically using a large set of experiments. The sensor latency t_{S} and robot latency t_{R} make up the system latency $T = t_{\mathrm{S}} + t_{\mathrm{R}}$. T covers the time period for signal acquisition, processing and transmission, and triggering the motion stop procedure. s_{S} is the safe detection distance for a given object (or body part) as defined in [15], and $v_{\mathrm{rel,max}}$ is the maximum permissible operating speed at contact as defined in (A.6) of ISO/TS 15066 [2]. It is given by,

$$v_{\mathrm{rel,max}} = \min(F_{\mathrm{T}}, p_{\mathrm{T}}A)/\sqrt{\mu K}, \quad (2)$$

where, F_{T} is the transient contact force, p_{T} is the transient contact pressure, A is the contact surface, μ is the reduced mass of the two-body system human-robot and K is the spring stiffness of the human body region in collision. For this set of equations, a fully inelastic collision is assumed, in which the full kinetic energy of the robot is transferred as inner energy to the human body. To properly model collisions between series elastic robotic arms and humans, the coefficient of restitution k needs to be determined. Nevertheless, the technical specification ISO/TS 15066 gives a more conservative approach. For collision avoidance, which is investigated in this research, set $v_{\mathrm{rel,max}} = 0\mathrm{m/s}$. An additional safety margin can be set by reducing s_{S} by that margin. (1) then simplifies to

$$v_{\mathrm{R,max}} = -\ddot{x}(t_{\mathrm{S}}+t_{\mathrm{R}}) + \sqrt{\ddot{x}^2(t_{\mathrm{S}}+t_{\mathrm{R}})^2 + 2s_{\mathrm{S}}\ddot{x}}. \quad (3)$$

IV. Experimental Setup

A. Measurement Setup

The CPS is driven by a novel closed-loop class D amplifier fully on-chip driver circuit. This proposed driver circuit is designed and fabricated in the TSMC 65nm CMOS process technology. A photo of the experimental setup is shown in Fig. 2.

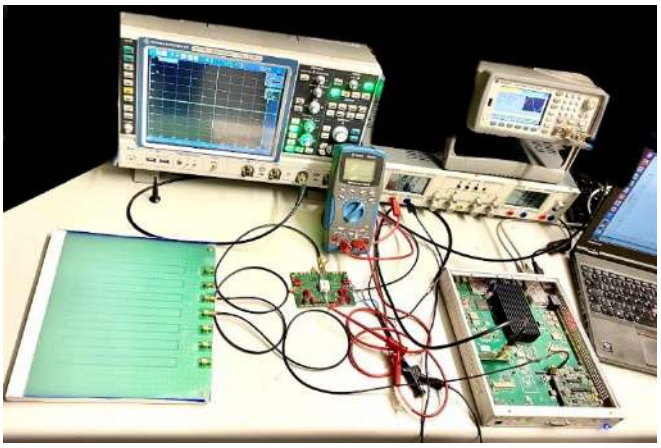


Fig. 2. Driver circuit in a robot environment test setup. The wire bonded test chip is mounted on PCB. Our setup is equipped with a signal generator, power supply, USRP X310, and an oscilloscope.

The driver circuit has been connected with wire bonding. The characterization of the driver circuit in a robot environment is performed on the fabricated prototype ASIC which is

mounted on a PCB. It is tested in a realistic environment, including a capacitive proximity sensor, a USRP X310 (which is a high-performance, scalable software-defined radio (SDR) platform), oscilloscope, and signal generator to deliver the reference voltage to the driving circuit. Moreover, a current meter is deployed to monitor the current consumption of the test chip during various setups.

The output voltage of the driver circuit is measured over capacitor load, which is varied from no-load to 1nF range. The reference voltage applied to the Test chip is 1V peak-peak sinusoidal signal at 500kHz frequency. The output shows a 500kHz frequency with second and third harmonics. The effect of harmonics increases with increase in the load capacitor value. Harmonic effects on the circuit's performance can be suppressed through post-digital filtering. Reliability of the circuit is also verified with varying frequency of f_{ref} from 10kHz to 10MHz.

Moreover, the linearity of the test chip is verified by ramping the input. As an outcome, with varying load capacitance and a wide range of frequencies, output voltage could well be achieved with a linearity coefficient (Output voltage/ Reference voltage) of nearly 94%. The lowest power consumption is 1.92mA in the setup of 1nF load capacitance and 1MHz excitation frequency.

B. Collision Avoidance Setup

Figure 3 shows the collision avoidance test setup. The setup is based on previous work [15] and is again configured in reverse. In a real-life scenario, the CPS is mounted on the robots' surface, and test objects resembling human body parts are placed at points of interest.

In our scenario, the test object is set at a distance well above the detection threshold of the sensor in the beginning of the experiment. The robots' tool center point (TCP) is then descending towards the CPS with a defined velocity. A motion stop procedure is triggered once the capacitance threshold is reached at the activation distance.

Our copper cladded test object *Sphere* is designed to resemble a human fist but can also be used for similarly shaped body regions, such as shoulders, elbows, or knees. The adequacy of this test object for representing a likewise shaped human body part has been elaborated and shown in [15].

We use a series elastic modular robotic arm kit *A-2085-06* by Hebi Robotics [22]. The motors of the Hebi robot provide settings to adjust individual limits for joint positions, velocities, and effort. The modules do not possess mechanical brakes. Therefore, a traditional emergency shut-off procedure would cause the robot to collapse and thus violate safety goals. Instead, a motion stop procedure is initiated by maintaining its position with active power (motor hold mode). This mode is activated by disconnecting the power wires (Power over Ethernet (PoE)) of its ethernet connection. In this mode, the robot is unresponsive to incoming motion commands. The series elastic nature of the robot will cause brief oscillations during rapid changes in motion. A similar test setup has also been used in [23].

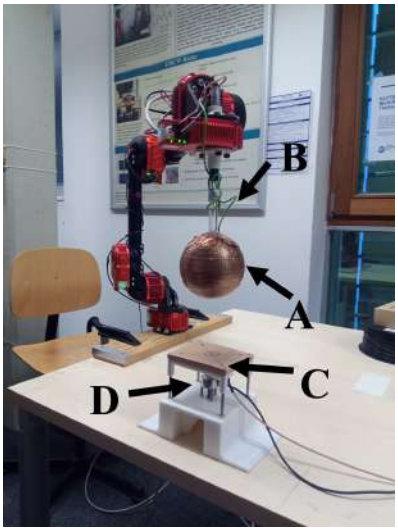


Fig. 3. Collision avoidance test setup. Both sensor and test object (**A**) are configured in reverse. The test object is grounded (**B**) as defined by [15]. The CPS is embedded in a soft skin patch (**C**). The setup can also be equipped with a force sensor and a spring (**D**) to measure collision forces, as in [23] .

V. Results

In the following, we present the results of the collision avoidance experiment showcased in Fig. 4. The Hebi robot moves with a speed of $300\text{mm}/\text{s}$ towards the sensor. The activation distance of the sensor is set as $d = 50\text{mm}$ to the *Position of Contact* (POC). The motion stop procedure consists of three phases. The first phase is the latent phase. The robot will not react until the capacitance signal is acquired, processed, and transmitted to the robot. The system latency during our configuration is approximately 65ms. This latency was determined using an external visual-based motion capture system. In the second phase, the motor hold mode triggers. The robot tries to maintain its position. Due to its kinetic energy and elastic nature, the robot will oscillate in relation to its prior speed. Here, the highest oscillation amplitude observed is 16mm. In the last phase, the robot will halt in its rest position with active power until the motor hold mode is disabled. As shown in Fig 4, the capacitance to range mapping is not linear, as the signal increases logarithmically with decreasing distance until reaching a potential saturation. Saturation can be avoided in the desired sensing range by proper frequency and gain settings. This logarithmic behavior has also been observed in [15] and corresponds to the theoretical behavior of such sensors.

VI. Discussion

ISO/TS 15066 permits undesired physical contact for all human body regions below the neck. Nevertheless, collisions with these regions should still be avoided in general, as such collisions negatively impact the personal well-being of a human co-worker, even if the robot only exerts small forces to the human. CPSs, on the other hand, allow the robot to react ahead of potential contacts, allowing reasonable operating speeds even for small detection distances below

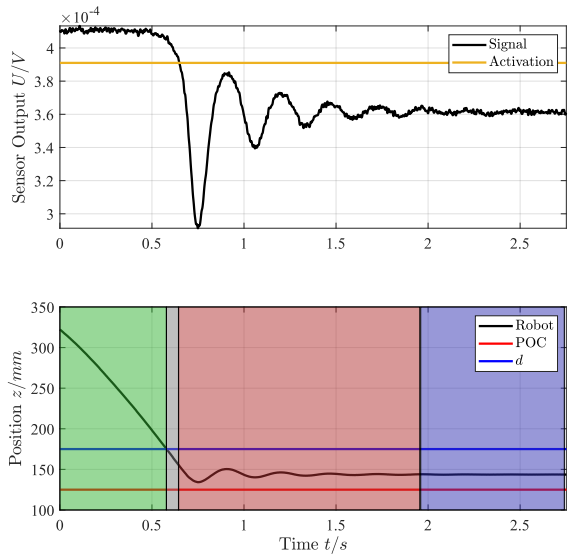


Fig. 4. Results - **a)**: Output of the CPS over time is shown in black. When the signal is below the activation threshold (orange), the motion stop procedure is activated. **b)**: Corresponding robot position over time is shown in black. The *Position of Contact* (POC) is indicated in red. The phases of the experiment are expressed by their background colors. Upon detection, the robot is operating normally (green phase). During the latent phase (grey) the robot is maintaining speed. The braking and oscillation phase is shown in red. The robot will remain in its idle position (blue phase). The sensor reacts at an activation distance of $d = 50$mm to the object (shown in blue). The robots' speed upon detection is 300mm/s.

10cm, while still avoiding contact. The actual performance of CPSs rely on multiple factors such as latencies (both sensor and robot), safe detection distance of the sensor, robot type, payload and pose. Therefore, the performance of each CPS needs to be evaluated for each robot task separately alongside its planned trajectory. Furthermore, for full safety compliance in a HRC application, the sensor needs to fulfill Performance Level d of ISO 13849-1 [24].

VII. SUMMARY AND OUTLOOK

In this work, we showcased the applicability of CPSs for collision avoidance scenarios using a series elastic robotic arm kit. The CPS is driven by a novel closed-loop class D amplifier fully on chip driver circuit. By using CPS a robot may still be able to operate if a human or object is close to the robot - below the minimal detection distance of mid-range ToF based sensors. The procedure described in this work allows fellow researchers and engineers to replicate the test and evaluate their CPSs for collision avoidance scenarios.

ACKNOWLEDGMENT

This work has received funding from the "European Regional Development Fund" (EFRE) and "REACT-EU" (as reaction of the EU to the COVID-19 pandemic) by the "Kärntner Wirtschaftsförderungs Fonds" (KWF) within the project Pattern-Skin 3520/34263749706. It also received funding from the "European Regional Development Fund" (EFRE) by the "Kärntner Wirtschaftsförderungs Fonds" (KWF) within the project CapSize 26616/30969/44253.

REFERENCES

[1] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pérez-Oria, "Working together: A review on safe human-robot collaboration in industrial environments," *IEEE Access*, vol. 5, pp. 26 754–26 773, 2017.

[2] *Robots and robotic devices- Collaborative robots (ISO/TS 15066:2016)*, International Organization for Standardization Std.

[3] *Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots*, International Organization for Standardization Std.

[4] *Robots and robotic devices — Safety requirements for industrial robots — Part 2: Robot systems and integration*, International Organization for Standardization Std.

[5] Blue Danube Robotics GmbH, *Airskin*, accessed April 12, 2022. [Online]. Available: bluedanuberobotics.com/

[6] H. Bo, D. Menoth Mohan, M. Azhar, K. Sreekanth, and D. Campolo, "Human-robot collaboration for tooling path guidance," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2016, pp. 1340–1345.

[7] Mayser GmbH & Co. KG, *Ultrasonicsafety*, accessed April 12, 2022. [Online]. Available: mayser.com/en/ultrasonic-safety-makes-lightweight-robots-safe

[8] Pepperl & Fuchs SE, *Ultrasonicsafety*, accessed April 12, 2022. [Online]. Available: https://www.pepperl-fuchs.com/global/en/ultrasonic_safety_sensor.htm

[9] tofmotion GmbH, *Tofcam*, accessed April 12, 2022. [Online]. Available: tofmotion.com/

[10] Inxpect S.p.A., *LBKSystem*, accessed April 12, 2022. [Online]. Available: inxpect.com/en/products/lbk-system/

[11] B. Ubezio, C. Schöffmann, L. Wohlhart, S. Mühlbacher-Karrer, H. Zangl, and M. Hofbaur, "Radar based target tracking and classification for efficient robot speed control in fenceless environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 799–806.

[12] M. Moradian, J. Sturm, A. Roshanghias, and H. Zangl, "Power efficient signal generation for capacitive sensors," 2022, submitted to IEEE Transactions on Instrumentation and Measurement.

[13] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 882–893, 2016.

[14] P. Svarny, M. Tesar, J. K. Behrens, and M. Hoffmann, "Safe physical hri: Toward a unified treatment of speed and separation monitoring together with power and force limiting," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7580–7587.

[15] S. Ergun, Y. Ding, H. Alagi, C. Schöffmann, B. Ubezio, G. Soti, M. Rathmair, S. Mühlbacher-Karrer, U. Thomas, B. Hein, M. Hofbaur, and H. Zangl, "A unified perception benchmark for capacitive proximity sensing towards safe human-robot collaboration (hrc)," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3634–3640.

[16] P. Chemweno, L. Pintelon, and W. Decre, "Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the iso 15066 standard for collaborative robot systems," *Safety Science*, vol. 129, p. 104832, 2020. [Online]. Available: sciencedirect.com/science/article/pii/S0925753520302290

[17] T. Schlegl, T. Kröger, A. Gaschler, O. Khatib, and H. Zangl, "Virtual whiskers — highly responsive robot collision avoidance," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5373–5379.

[18] S. Mühlbacher-Karrer, L.-M. Faller, H. Zangl, T. Schlegl, and M. Moser, "Short range capacitive proximity sensing," in *2015 IROS Workshop on Alternative Sensing for Robot Perception (WASRoP)*, 2015.

[19] X. Jiang, "Fundamentals of audio class d amplifier design: A review of schemes and architectures," *IEEE Solid-State Circuits Magazine*, vol. 9, no. 3, pp. 14–25, 2017.

[20] K. Jha, S. Mishra, and A. Joshi, "High-quality sine wave generation using a differential boost inverter at higher operating frequency," *IEEE Transactions on Industry Applications*, vol. 51, no. 1, pp. 373–384, 2014.

[21] O. Khatib, "Inertial Properties in Robotic Manipulation: An Object-Level Framework," *International Journal of Robotics Research*, pp. 19–34, 1995.

[22] Hebi Robotics Inc., “Hebi A-2085-6 Data Sheet,” April 2022. [Online]. Available: http://docs.hebi.us/resources/kits/datasheets/x-series/A-2085-06_Datasheet.pdf
[23] H. Alagi, S. Ergun, Y. Ding, T. P. Huck, U. Thomas, H. Zangl, and B. Hein, “Evaluation of on-robot capacitive proximity sensors with collision experiments for human-robot collaboration,” 2022, submitted to 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022).
[24] *Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design*, International Organization for Standardization Std.

Analysis of the Trends Towards Domain-Crossing Safety Standardization in Close Human-Robot Interaction

Michael Rathmair and Michael Hofbaur

Abstract— **The paper shows the current ongoing trends and activities in robot safety standards development. Special attention is paid to the field of close human-robot Interaction. This is increasingly important since in nearly all sectors of robot applications collaborative operation without physical protective devices between the user and the robot is targeted. Physical robot safety is thus a challenge arising in a large variety of robotic domains (e.g. industrial, service and assistance, mobile, medical, intralogistics, household, etc.). Nevertheless, in general, standardization committees are specifically implemented to develop, revise and maintain standards for specific application sectors. This is also the case in robot safety, and committee members are reflecting a group of experts in the associated application field. However, there is a gratifying trend that domain experts start to collaborate with each other, share their experiences, and harmonize standards for safe human-robot collaboration applicable to various robot application domains. This work elaborates this trend by highlighting selected standards synergies on ISO, CEN, and the national level, analyzing future trends towards domain-crossing robot safety standards, and drawing implications for system integrators, and end-users of collaborative robot applications as well as researchers driving future robot-system technologies.**

I. INTRODUCTION TO STANDARDIZATION

Standardization is an essential process for the economy in terms of value chain creation and optimization, technology development and progressive innovation actions. Forefront scientific and technological aspects and developments are continuously included to standardization processes by standard revisions or the initiation of new documents. In the case of a revision each proposal is assigned to a standards development track which typically has a timeframe of 36 month [38]. In the ISO (International Standardization Organization) the standards development process has to strictly follow a the ISO/IEC Directives, Part1 - Procedures for the technical work Consolidated ISO Supplement [31].

One of the most significant fact is that compliance with standards is not mandatory [38]. The definitions in standards reflects requirements developed by groups of domain experts called a technical committee (TC). These persons that represent a country in the ISO committee (the countries delegation for standardization) are nominated by an associated national standardization mirror-committee and persons are expected to actively contribute the national interests and know-how to the international standardization process [38].

Both authors are with JOANNEUM RESEARCH Forschungsgesellschaft mbH – Institute for Robotics and Mechatronics, 9020 Klagenfurt am Wörthersee, Austria {first name}.{surname}@joanneum.at

The general structure of ISO machinery standards is illustrated as a pyramid in Figure 1 [36]. On the top of the pyramid are laws and guidelines that form the fundamental directives mandatory to follow, and are the basis for legislative processes. CE-Marking which illustrates the conformity with the EU-Machinery directive [11] is fundamental for bringing a product to the market or putting machinery into operation. Strictly speaking, in the event of an accident, the system/machine manufacturer's work will be assessed against current legislation and not against requirements given in a standardization document. On the bottom layers of the standardization pyramid the ISO divides machinery standards into thee types:

- Type-A Standards (Basic Safety Standards): These type of standardization documents give basic concepts, principles for design, and general aspects that are applicable to machinery (i.e. ISO 12100 [16])
- Type-B standards (Generic Safety Standards): These deal define requirements for a specific safety aspect or one type of safeguard, which can be used across a wide range of machinery. (i.e. ISO 13855 [17])
- Type-C standards (Machine Safety Standards): These documents handle detailed safety guidelines for a particular machine or group of machines. (i.e. ISO 10218 [18])

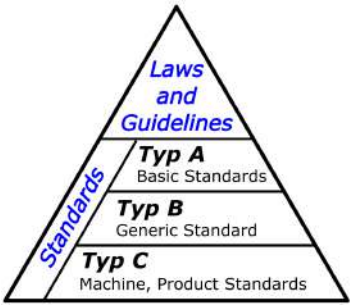


Fig. 1. Document structure of ISO machinery standards.

The rest of the paper focuses to the context of safety for robot applications and even more specifically to interaction between robot systems and humans. Since robots and machines operate collaboratively in a shared human-machine working space requirements and evaluation of safe operation receives an increased priority. Regardless of a specific application domain i.e. robotics for production, service robotics in the home, or assistance systems in medicine and rehabilitation, the primary goal is safe operation and handling of the

full robot system. The experts in standardization committees trend to combine and unify the domains just mentioned and to formulate general and abstracted requirements for human-robot collaboration and interaction.

II. ROBOTIC SAFETY STANDARDS, ASSOCIATED COMMITTEES AND STANDARDIZATION TRENDS

In this section we report about ongoing standardization activities relevant for robotics in general and in particular for human-robot collaboration. We divide into international, European and national (in this paper Austrian) level of activities. Nevertheless for selected standards as illustrated in Figure 2 we analyze the trends and give a baseline for further discussion and analysis.

A. Standards on the international ISO Level

In the international level the ISO (international standardization organization - *https://www.iso.org/*) is responsible for the main standardization activities in robotics. The corresponding committee is the ISO/TC 299 which is partitioned into 8 working groups [33]. In the context of robot safety and in particular safe human robot interaction in TC 299 the working groups WG 2 (Service robot safety), WG 3 (Industrial safety) and the new working Group WG 8 (Validation methods for collaborative applications) are the most active ones.

In the industrial robotics sector, which is under the responsibility of the TC 299/WG 3 at the time of writing the paper (April 2022) an ongoing task is the revision of the ISO 10218 (Robots and robotic devices — Safety requirements for industrial robots) [18] standards from the year 2011. Besides of this standard in particular for collaborative robotic applications since 2016 the technical specification ISO/TS 15066 [37] is in place. This ISO/TS will be integrated and revised to the new ISO 10218 part 2 version by adding the following annexes for industrial robots operated in collaborative applications:

- Annex L (normative), which is focusing to separation distance requirements that shall be fulfilled in collaborative robot applications implementing the collaborative operation mode speed and separation monitoring (SSM).
- Annex M (informative), holds quasi-static and transient force and pressure limits for potential human-robot contact situations in applications where a collaborative operation of power and force limiting (PFL) is realized.
- Annex N (informative), is about procedures and corresponding requirements for the validation of PFL collaborative applications by pressure and force measurements.

ISO/TS 15066 will then be withdrawn when the new editions of ISO 10218-1 and ISO 10218-2 are published.

In part 1 of the 10218 standard which specifies the requirements on industrial robots per se has in its revised version a new table which classifies manipulators into class I and class II robots [22]. Class I robots has to satisfy a maximum achievable speed of the TCP (tool center point), the total mass of moving parts of the manipulator has to be below of a defined limit and the maximum achievable force of the manipulator has to be below a certain specified extent. Robots satisfying these aspects have less requirements to safety functions and are more suitable into the direction of inherent safe operation in close human-robot interaction applications [22].

In April 2022 at the time of writing the paper the revised versions of ISO 10218 part 1 and part 2 are in the FDIS (final draft international standard) approval stage [31] and publication is planned for the end of 2022 [22], [23].

In addition to the activities that define a measurement method for the validation of human-robot contact situations in the ISO/FDIS 10218-2/2022 [23], a new working group WG 8 (Validation methods for collaborative applications) was established [33]. The goal of the group is to define measurement processes across different domains of robotics and to establish a common understanding for the validation of human-robot interaction scenarios by applied force and pressure measurements. For this objective, the committee's experts are developing the ISO/AWI PAS 5672 (Publicly Available Specification) [27] standard to meet the goals of an international domain-crossing consensus for validation. The defined validation processes should be applicable for a variety of robot types (stationary robots, mobile manipulators, mobile platforms, etc.) as well as for different application domains (manufacturing, logistics, service, healthcare, etc.).

For non industrial-applications (in general, robots that are applied in non-manufacturing environments) the TC 299 is working on several new but also revising standardization activities. Selected standards that are in particular relevant for discussion in the next section are described in the following itemization[33].

- ISO/AWI 13482 (Robotics — Safety requirements for service robots) [26]: According to the ISO standards development track the ISO/AWI 13482 (approved working item) is in the preparatory stage. The document defines safety requirements for physical human-robot contact situation for robots used in personal and professional/commercial service applications. In this way similar to the new version of the ISO 10218-2 [23] the ISO/AWI 13482 defines requirements for close human robot interaction but in a service robot application context. In conjunction with the standard, a technical report ISO/TR 23482-1 (Application of ISO 13482 — Part 1: Safety-related test methods) [21] was published in 2020. This describes test methods for safety requirements in the field of personal care robots as described in ISO 13482. However, the technical report also emphasizes that the test methods listed are not universally applicable to all types of service robots. In these cases, reference is made to other more specific standards.

- IEC 80601-2-77/78:2019/DAMD 1 (Particular requirements for the basic safety and essential performance of robotically assisted surgical equipment) and (Particular requirements for basic safety and essential performance of medical robots for rehabilitation, assessment, compensation or alleviation) [24], [25]: The ISO/AWI 13482 excludes medical applications. Nevertheless, TC 299 is contributing also to this section by actively working on amendment documents for IEC 80601-2-77/78. The goal is to transfer the expert's know how in the field of safety for manufacturing applications regarding technical changes to most of the medical standards within the IEC60601-1 series.
- ISO/CD 5363 Robotics (Test methods for Exoskeleton-type Walking RACA Robot) [29]: This committee draft (CD) of a standard is an ongoing new development in the committee phase [31]. Focus of the content are exoskeleton-types that assist humans to move from one location to another, by making reciprocating motion having intermittent contact with the corresponding travel surface. The standard will include test methods as well as requirements for test equipment for conformity assessment of these types of robot equipment.

Another technical committee working significantly on the normative requirements for the safe cooperation of humans and machines is ISO/TC 199 (Safety of machinery) [32]. The work program of the expert group includes, for example, the new revision of the standard ISO 13855 - ISO/DIS 13855 (Safety of machinery — Positioning of safeguards with respect to the approach of the human body) [30] which defines safety distances between moving machine parts and the machine operator or the new development of an abstracted standard for integration of machines into a smart machine system ISO/CD 11161.2 Safety of machinery — Integration of machinery into a system — Basic requirements [28].

B. Standards on the European CEN level

Robot safety initiatives in Europe are highly driven by activities of H2020 Projects such as COVR [6], RobMoSys [7], ODIN [8], ROSSINI [9] etc. In particular for testing methods for close human-robot interaction the project COVR is a main contributor. COVR has the mission to increase safety for all robots sharing space with humans by applying skill-based testing across robot domains and to create common standards for how to test and validate safety for collaborative robot applications. Results of COVR are so called safety protocols that are available on the COVR toolkit web platform (https://www.safearoundrobots.com/). As an executive summary of the developed project results a technical committee incorporated the protocols to a new European CEN/CWA standard. The development of the CEN/CWA 17835 (CEN Workshop Agerrment) (Guidelines for the development and use of safety testing procedures in human-robot collaboration tool) [10] took one year and the CWA got published in early 2022.

Since the COVR project and the developed protocols cover several application areas of robots, the CEN/CWA 17835 also applies to a wider range of robot applications besides industrial applications. The document serves as a guideline with respect to different robot categories but with the common property that they realize an application including close human-robot interaction. However, the CEN/CWA 17835 is an informative document which mainly covers the following objectives:

- Definition the concept of HRC skill in order to specify a uniform framework for skill based multi-domain robot testing procedures. In the context of the CEN/CWA 17835 such skill are exclusively addressing mechanical hazards that are relevant for human-robot commaboration (HRC) interaction properties. As already emphasized a HRC skill is an abstract description that is technically neutral and defined a functional specification of the system independent of the applied robot category or application domain.
- Specification of System-Level validation protocols (SLV protocols) that can be used as a step-by-step instruction for the execution of these measurements. Goal of SLV protocols in general is to provide a certain level of evidence in order to prove that safety measures are implemented properly with reference to corresponding application domain specific standards.
- CEN/CWA 17835 gives an overview and defines robot categories such as industrial, service, healthcare and medical robots.
- The document holds a table of test methods that include HRC skills, the robot application domain, the type of the test and a corresponding reference to standards closer specifying the requirements for test definition and execution.
- In the appendix a template for an SLV protocol as well as a examples for a SLV protocols (Test mobile platform to maintain a separation distance, Test manipulator in shared human-robot control to prevent spatial overreaching for the utilizer) are given.

C. Standards on the national level

For the robotics domain the national mirror committee reflecting the activities of the ISO TC 299 in Austria is the Committee 28 [3] hosted by the Austrian Standarisation Institute (ASI) also known as Austrian Standards.

The experts from the committee 28 are actively reacting to the described trends in the industrial robotics domain by setting up a new committee for "Smart Manufacturing" [4]. The group will support the international technical committees ISO/TC 184 - Automation systems and integration [34] and IEC/JWG 21 - Smart Manufacturing Reference Model(s) [15]. A large number of well-known Austrian companies and research centers have already expressed their interest in participating. The official launch of the smart manufacturing committee was planned for 2020 but still delayed [2]. Nevertheless, the Austrian community for robot safety actively exchanges information at regular

intervals through national industry and scientifically driven events and workshops as well as at the meetings of the ASI Committee 28.

A relatively new standard that was developed on a national level in America and worth to mention in the context of this paper is the ANSI/RIA R15.08 [1]. The RIA (Robotic Industries Association) is working on standards for mobile robot systems also covering the application domain of mobile manipulators, which are a combination of a mobile platform and a serial manipulator on top. The first part specifying the safety requirements for manufactures got published in end 2020, Part 2 that will specify the requirements to designers and integrators, and Part 3 that will define the safety requirements for the end user of mobile industrial robots are still under development. The application scenario of applied mobile manipulators is increasingly interesting for the industrial domain and since robots navigate dynamically through manufacturing sites the human-robot shared space is not longer limited to workcells. Mobile manipulation as a robot system has still significant challenges that go beyond a segmented application of applying the ISO 10218 [18] for the serial manipulator and applying the ISO 3691-4 [20] for the mobile platform. Nevertheless, the ANSI/RIA R15.08 is a first initiative regulating the requirements for mobile manipulation in terms of a close human-robot approach that is not spatially limited to workcells in the industrial context.

III. IMPLICATIONS AND DISCUSSION

In the following subsections we discuss application specific but also general implications of the described ongoing standardization activities and trends for robot implementation, testing and verification but also research and development activities. Special attention will be paid to an analysis of the cross-domain standardization trends and which advantages and disadvantages arise for applications implementing close human-robot interaction.

Type A vs. Type C Standards: In the introduction section we described the ISO structure of standards in the field of machinery (see Figure 1). Nevertheless, robotic systems are technical devices that are used even in a broader context than that of machinery. An ISO type A standard is therefore formulated in an abstract way so that it can be applied to a wide range of machines in various machinery application domains (e.g. metalworking, woodworking, assembly machines, etc.). Type A standards, however, do not describe synergies with other areas of application that may lie outside of machines, even if the defined safety requirements can be transferred in a meaningful way. For example, the ISO 12100 [16] standard gives requirements for risk assessment, but is by definition limited to the machinery domain, although the principles and a methodologies described in this standard can also be usefully applied to robotic systems in other domains, such as medical or personal healthcare robots.

For for industrial robot systems as a specific machine or a robot system as subpart of a machine, the type C standard ISO 10218 [18] is in place. On the one hand, it is very helpful for system integrators that specific type C standard has been developed for the development and integration of robot systems, but on the other hand, due to this specialization, especially when the topic of human-robot interaction plays a role, hardly any synergies to other domains are mentioned and referenced in the safety requirements.

The new ISO/TC 299/WG 8: As described in SectionII, the newly established working group 8 of ISO /TC299 is working on an ISO/PAS document which has the goal to standardize the way we validate robots for biomechanically safe human-robot interaction through measurement. Similar to the CEN/CWA 17835, the developed ISO/PAS 5672 standard should also take into account various types of collaborative robot systems beyond the manufacturing sector. The goal of a cross-domain standard requires in any case a targeted collaboration and communication of experts from the robotics application domains to be considered (logistics, healthcare, agriculture, rehabilitation, etc.). However the trends which are taken up by WG 8 is a sustainable initiative that enhances interaction between communities, technical committees and address the challenge of developing common verification and validation solutions for safe human-machine interaction in general and in particular for domain -crossing collaborative robot application.

Update of biomechanical limits for human-robot contact situations: ISO/FDIS 10218-2 [23] contains a table with biomechanical limits for force and pressure loads in human-robot contact situations. These limits are the result of a scientific study and were integrated into ISO/TS 15066 [37] in 2016. The limit values represent a conservative barrier and often prevent the implementation of collaborative robot operation in an industrial context due to the requirements of reduced handling speed. In this context, the scientific community is making important contributions to better assess contact situations with models and, in combination with digital risk assessment methods, to make safety evaluations more applicable [5], [13], [35], etc. These approaches are also significantly integrated in the working roadmap of ISO TC 299/WG 8. Special attention in the modeling and validation of contact situations is given to transient impact events (i.e. unconstrained impact in free space). The scientific work in this field provides an important basis for integrating new findings from studies as well as mathematical models [12], [14]. The trend towards virtualization in the digital world (digital twin) for the analysis of close human machine interaction scenarios is clearly emerging across domains in the technical committees and is seen as a valuable tool for safety assessment.

Abstract safety properties for runtime verification: Cross-domain definitions for safety properties can not

	Committee	Currently Published	Trend or ongoing Initiative
ISO	ISO TC299/WG 3	• ISO 10218 about industrial robot safety in the version of 2011 • ISO/TS 15066:2016 safety for collaborative robot applications	• New ISO/FDIS versions of part 1 and part 2 of the 10218 • ISO/TS 15066 will be integrated in the new version of ISO 10218-2
	ISO TC299/WG 8	• No cross-domain standard for testing methods available	• New Workgroup 8 established working on a new ISO/AWI PAS 5672 on test methods for measuring forces and pressures
	ISO TC299/WG 2	• ISO 13482 in the version of 2014 • ISO/TR 23482-1 safety related test methods for ISO 13482 of 2020	• Revision ISO/AWI 13482 under development
	ISO TC299 Amendments	• IEC 80601 - Several parts with various publication dates	• Amendments for robotically assisted surgical equipment and medical robots for rehabilitation
	ISO TC299	• No test procedure for exoskeletons available	• Committee draft of a new standard ISO/CD 5363 for test methods for exoskeletons
	ISO TC 199	• ISO 13855 about machinery safeguarding in the version of 2010	• Revised ISO/DIS of the 13855 standard • ISO/CD 11161.2 New safety standard for integration of machinery to a system
CEN	H2020 COVR project consortium	• Collection of safety protocols in the COVR toolkit • CEN/CWA 17835 about safety testing procedures in human-robot collaboration published in 2022	• COVR Hubs established for working on safety initiatives beyond the COVR project
NATIONAL	Austria ASI	• Committee 28 as mirror committee of the ISO TC 299	• Setup of a new committee for smart manufacturing • Support of the ISO TC 184 and the IEC/JWG 21
	USA	• New standard ANSI/RIA R15.08 from 2020 covering safety requirements for mobile manipulators	• Development of part 2 that is focused to integrators and part 3 that is focused to end-users

Fig. 2. Overview of the selected standards discussed in this paper

only influence the development of robot systems but can also be seen as a verification tool for the operating time. If requirements from standards are abstracted in an appropriate form, they can be continuously verified in a formalized form in terms of safety properties during the operation of a robot system. The more abstractly these properties are formulated, the easier it is to implement software features that can be implemented as safety features independently of the domain. In the sense of cyclic self-verification and thus also self-awareness of the machine, various concepts for this approach are currently being investigated in the RunRoc research project. Due to the complexity of the machines, verification and validation after adaptation and modification is correspondingly time-consuming and potentially expensive. Main expected benefit is that automatic self-verification against a formalized specification in parallel to operation would significantly reduce costs, time-to-market and in particular increases operation flexibility. From self verification features, we will not only expect a well-founded validation of the production process, it will also provide the basis for the application of modern AI methods in the reactive control context, collaborative, trustworthy and safe robot-based machinery.

Speed and Separation Monitoring: A large part of the standardization activities presented and analyzed revolve around the verification and validation of human-robot contact situations. Another mode of operation that can be seen as very important for future robot systems is speed and separation monitoring SSM according to ISO 10218, where the interaction between man and machine is implemented in a non-contact manner with collision avoidance by constantly maintaining a safety distance between man and machine. The calculation of this safety distance is specified in the new edition of ISO/FDIS 10218-2 in Annex L. The speed and separation monitoring mode is strongly linked to the challenges for sensor components. Sensors for environmental perception must be designed with defined reliability parameters and structure categories depending on the hazard potential of the application (see ISO 13849 [19] and ISO 10218). Nevertheless, high-performance environment perception is a property that must be fulfilled in close human-robot interaction in various implementation domains. A trend that is also emerging from research projects is a combined mode of operation, i.e. the best possible observance of a safety distance and, if collisions occur, no serious physical danger to humans.

Testing bodies for robot safety: Cross-domain requirements also expand the business field for institutions that are professionally (accredited) involved in the testing of robot systems. Uniform procedures and requirements for test equipment enable test processes that are applicable to a wide range of stakeholders, including robot manufacturers, system integrators and end users. In the end, the knowledge gained from the tests benefits all domain experts and improvements can be actively introduced into the improvement of the specified standard procedures. In particular, the creation of an ISO/PAS document as implemented by TC 299/WG 8 provides a good framework for agile and update-driven work.

IV. CONCLUSION AND NEXT STEPS

Standards and finally requirements for safety may significantly benefit if experts from different robotic domains collaborate bundle their expertise and develop standardization documents applicable for human-robot interaction (HRI) that is not bounded to specific scenarios. However, this paper gives a survey/overview over ongoing activities, trends and specific effects for research and system integration and participating in standards development. The state of the art in standardization processes where a committee of experts maintains and develops one set of domain specific

standards is already enhanced by ongoing activities that try to abstract and formalize collective safety requirements for various robotics applications including HRI.

As next steps, also in the sense of this publication, it should be emphasized that cross-domain considerations and developments will be essential in the definition of safety requirements, test procedures, conformity assessments and scientific work in the field of robot safety. All stakeholders are therefore encouraged to be communicative in solving challenges together and to build a multi-domain knowledge base to make future implementations of HRI applications safe and efficient.

ACKNOWLEDGMENT

This work has been supported by the Austrian Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology within the project "RunRoc"

REFERENCES

[1] ANSI - American National Standards Institute, RIA - Robotics Industries Association, *ANSI/RIA R15.08-1-2020 Industrial Mobile Robots - Safety Requirements - Part 1: Requirements For The Industrial Mobile Robot*, 2020.

[2] *Internal Austrian Standards committee document - N0467*, Austrian Standards, 2019-03-08.

[3] Austrian Standards, *Webpage Komitee 28, Lagerung / Tibotechnik und Verzahnung*, accessed April-2022. [Online]. Available: https://www.austrian-standards.at/en/standardization/committees-working-groups/national-committees/committees/124/details

[4] *Standards sind entscheidend, damit Industrie 4.0 funktioniert*, Austrian Standards, accessed November-2019. [Online]. Available: https://www.austrian-standards.at/infopedia-themencenter/specials/2-fachkongress-industrie-40/standards-sind-entscheidend-damit-industrie-40-funktioniert/

[5] M. R. da Silva, F. Marques, M. T. da Silva, and P. Flores, "A compendium of contact force models inspired by hunt and crossley's cornerstone work," *Mechanism and Machine Theory*, vol. 167, p. 104501, 2022.

[6] European Comission, CORDIS, *H2020 Webpage: Being safe around collaborative and versatile robots in shared spaces - COVR*, accessed April-2022. [Online]. Available: https://cordis.europa.eu/project/id/779966

[7] ——, *H2020 Webpage: Composable Models and Software for Robotics Systems - RobMoSys*, accessed April-2022. [Online]. Available: https://cordis.europa.eu/project/id/732410

[8] ——, *H2020 Webpage: Open-Digital-Industrial and Networking pilot lines using modular components for scalable production - ODIN*, accessed April-2022. [Online]. Available: https://cordis.europa.eu/project/id/101017141

[9] ——, *H2020 Webpage: RObot enhanced SenSing, INtelligence and actuation to Improve job quality in manufacturing - ROSSINI*, accessed April-2022. [Online]. Available: https://cordis.europa.eu/project/id/818087

[10] European Committee for Standardization (CEN), *CEN/CWA 17835 - Guidelines for the development and use of safety testing procedures in human-robot collaboration*, 2022. [Online]. Available: https://www.cencenelec.eu/media/CEN-CENELEC/CWAs/RI/cwa17835_2022.pdf

[11] *Directive 2006/42/EC Machinery Directive of the European Parliament and of the Council of the European Union*, European Parliament and of the Council of the European Union, 2006. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32006L0042

[12] C. Fischer, M. Steiner, M. Neuhold, M. Papa, A. Markis, and S. Schlund, "An investigation of the measurement of transient contacts in human-robot interaction," in *International Conference on Robotics in Alpe-Adria Danube Region*. Springer, 2022, pp. 547–555.

[13] L. Gualtieri, E. Rauch, and R. Vidoni, "Development and validation of guidelines for safety in human-robot collaborative assembly systems," *Computers & Industrial Engineering*, vol. 163, p. 107801, 2022.

[14] D. Han, M. Y. Park, J. Choi, H. Shin, and S. Rhim, "Analysis of human-robot physical interaction at collision," in *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*. IEEE, 2021, pp. 153–156.

[15] *IEC/TC 65 - Industrial-process measurement, control and automation*, IEC, accessed November-2019. [Online]. Available: https://tinyurl.com/r3psfg4

[16] International Organization for Standardization (ISO), *ISO 12100:2010-11 Safety of machinery - General principles for design - Risk assessment and risk reduction*, 2010.

[17] ——, *ISO 13855:2010 Positioning of safeguards with respect to the approach speeds of parts of the human body*, 2010.

[18] ——, *ISO 10218:2011 Robots and robotic devices*, 2011.

[19] ——, *ISO 13849-1:2015 Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design*, 2015.

[20] ——, *ISO 3691-4:2020 Industrial trucks — Safety requirements and verification — Part 4: Driverless industrial trucks and their systems*, 2020.

[21] ——, *ISO/TR 23482-1:2020 Robotics — Application of ISO 13482 — Part 1: Safety-related test methods*, 2020.

[22] ——, *ISO/FDIS 10218-1:2021 Robotics — Safety requirements — Part 1: Industrial robots*, 2021.

[23] ——, *ISO/FDIS 10218-2:2022Robotics — Safety requirements — Part 2: Industrial robot systems, robot applications and robot cells*, 2022.

[24] ——, *IEC 80601-2-77:2019/DAMD 1 Medical electrical equipment — Part 2-77: Particular requirements for the basic safety and essential performance of robotically assisted surgical equipment — Amendment 1*, accessed April-2022. [Online]. Available: https://www.iso.org/standard/83340.html

[25] ——, *IEC 80601-2-78:2019/DAMD 1 Medical electrical equipment — Part 2-78: Particular requirements for basic safety and essential performance of medical robots for rehabilitation, assessment, compensation or alleviation — Amendment 1*, accessed April-2022. [Online]. Available: https://www.iso.org/standard/83341.html

[26] ——, *ISO/AWI 13482 Robotics — Safety requirements for service robots*, accessed April-2022. [Online]. Available: https://www.iso.org/standard/83498.html

[27] ——, *ISO/AWI PAS 5672 Robotics — Collaborative applications — Test methods for measuring forces and pressures in quasi-static and transient contacts between robots and human*, accessed April-2022. [Online]. Available: https://www.iso.org/standard/82488.html

[28] ——, *ISO/CD 11161.2 Safety of machinery — Integration of machinery into a system — Basic requirements*, accessed April-2022. [Online]. Available: https://www.iso.org/standard/79368.html

[29] ——, *ISO/CD 5363 Robotics — Test methods for Exoskeleton-type Walking RACA Robot*, accessed April-2022. [Online]. Available: https://www.iso.org/standard/81161.html

[30] ——, *ISO/DIS 13855 Safety of machinery — Positioning of safeguards with respect to the approach of the human body*, accessed April-2022. [Online]. Available: https://www.iso.org/standard/80590.html

[31] *Consolidated ISO Supplement-Procedures specific to ISO*, ISO, 2021. [Online]. Available: https://www.iso.org/sites/directives/current/consolidated/index.xhtml

[32] *ISO/TC 199 Safety of machinery - Webpage*, ISO, accessed April-2022. [Online]. Available: https://www.iso.org/committee/54604.html

[33] *ISO/TC 299 Robotics - Webpage*, ISO, accessed April-2022. [Online]. Available: https://www.iso.org/committee/5915511.html

[34] *Technical committee ISO/TC 184 - Automation systems and integration*, ISO, accessed November-2019. [Online]. Available: https://www.iso.org/committee/54110.html

[35] J. Li, Y. Guan, H. Chen, B. Wang, T. Zhang, J. Hong, and D. Wang, "Real-time normal contact force control for robotic surface processing of workpieces without a priori geometric model," *The International Journal of Advanced Manufacturing Technology*, pp. 1–15, 2022.

[36] A. Schneider, *Zertifizierung im Rahmen der CE-Kennzeichnung, Konformitätsbewertung und Risikobeurteilung nach der Maschinenrichtlinie 2006/42/EG und anderen europäischen Richtlinien*. VDE Verlag, 2018, no. 5.

[37] Technical Specification, "ISO/TS 15066:2016, Robots and robotic devices – Collaborative robots," International Organization for Standardization, Tech. Rep., 2016.

[38] J. Winkler, *Standards einfach mitgestalten*. Austrian Standards, 2018, no. 1.

Software Toolchain for Modeling and Transforming Robotic Workflows into Formally Verifiable Model Representations

Thomas Haspl[1], Michael Rathmair[1], Maximilian Papa[2], Michael Hofbaur[1] and Andrea M. Tonello[3]

***Abstract*— Formal verification represents an essential concept of mathematically proving or disproving the correctness of a system based on previously defined specifications. Applied to robotic workflows it can be used to prove their functional correctness, where it gains particular importance with the introduction of robot programming types for non-experts. In this paper, a software toolchain for modeling and transforming robotic workflows into formally verifiable model representations is presented. A graphical way of modeling robotic workflows in a distinctive way with a subsequent automatic transformation into verifiable code form the core of the presented toolchain. A software for generating formal specifications based on a modelled robotic workflow completes the toolchain presented in this work. The output artifacts of the particular software parts eventually allow a formal verification of robotic workflows against a desired behavior represented by the generated specifications.**

I. INTRODUCTION

Formal verification by model checking [1] is a way to formally prove the conformity of an abstract system model with a formal specification. The basic approach of formal verification by model checking depends on two major input artifacts, the formal system model and the formal verification specifications. The formal system model is the formal representation of the system under test. The formal verification specifications describe the desired behavior of the system. Verification specifications are often also called verification properties. In this work, these two terms are used synonymously.

The quality and validity of the verification results strongly depend on the quality of the formal system model and the formal specifications. One crucial step in this process is therefore the construction of the abstract system model. Different model checking tools often require a good knowledge and experience with a provided tool or syntax. Consequently, creating a formal system model in a special format can not only be difficult, but also error-prone. This is the main problem our presented work focuses on.

Although, formal verification originated from the electronics and hardware development domain, the concept is more and more used in the domain of software development as well. A typical testing phase in software development consists of creating a number of test cases for critical scenarios. The drawback of such an approach is that the tests can only cover the scenarios they represent. With a formal verification approach this problem is solved, as model checking is characterized by automatically verifying the entire state space of the modelled system.

The step towards formally verifying the structural correctness of workflow descriptions has rarely been done so far. With our work, we want to put a focus on enabling the formal verification of workflow descriptions including industrial robots. With industrial robots we typically mean serial manipulators or stationary arm robots. This kind of robots will also appear in all examples throughout this work and has been used in the workflows for evaluating our toolchain. Nevertheless, our toolchain is theoretically also able to handle robotic platforms of all kinds.

A well-engineered formal verification system for robotic workflows can not only help to detect errors, that might remain unnoticed for a long time, much earlier, but also to exactly identify a system state that might cause a problem.

A. Related Work

The work in [8] describes the theoretical predecessor to this work. A lot of thoughts have been spent there on creating an abstract model of an robotic workflow and on creating a good formal verification structure. For the proposed concept in this work, the most promising aspects in [8] have been picked up and implemented in a number of software applications.

The work in [12] also declares an approach in formal verification of robotic systems, but puts a focus on verifying safety aspects. A methodology for robotic applications, that relies on formal verification techniques to automate traditional risk analysis methods, is presented.

The presented work in [9] also presents an approach for formally verifying safety aspects in robotic applications. The presented system is built by extensive manual modeling of all application aspects such as the workflow, the robot's environment and the verification properties.

B. Contribution

With this work we want to take a step further into formal verification of the structural correctness of robotic workflows and the automatic creation of the therefore needed formal workflow models. Most existing approaches often rely on a manual modeling step, which can be error-prone, especially in large systems. Furthermore, the structural correctness

[1]Authors are with JOANNEUM RESEARCH Forschungsgesellschaft mbH – Institute for Robotics and Mechatronics, 9020 Klagenfurt am Wörthersee, Austria {first name}.{surname}@joanneum.at

[2]Authors are with Fraunhofer Austria Research GmbH, Theresianumgasse 7, 1040 Vienna, Austria {first name}.{surname}@fraunhofer.at

[3]Authors are with Alpen-Adria-Universität Klagenfurt, Universitätsstraße 65-67, 9020 Klagenfurt am Wörthersee, Austria {first name}.{surname}@aau.at

of robotic workflows, which is a precondition for a safe application, often seems to be a little bit out of focus. With the toolchain presented in this work we want to provide a coherent toolchain for formal verification customized to robotic workflows.

II. METHODOLOGY

The requirements specification represents a collection of all requirements that are expected for the design and verification of a – in our case – robotics-based production system. As there is an ongoing trend into human-robot-collaboration in such production systems, the verification of the structural behavior of such systems should be comprehensive and reliable. The technique of symbolic model checking has the ability to meet these requirements.

NuSMV[3] is a symbolic model checking program based on binary decision diagrams, a data structure for representing Boolean functions. This command line tool is used as the main verification tool in the presented toolchain. Thus, simply speaking, the goal is to use the NuSMV model checker to formally verify robotic workflows. Manually implementing robotic workflows in NuSMV would be the most direct approach, but is tedious and error-prone, especially when the robotic applications grow in complexity. Therefore, we decided to introduce a prior step of graphically modeling the robotic application and to automatically transform this model into veryfiable NuSMV code afterwards. The basic structure of our toolchain is shown in Fig. 1.

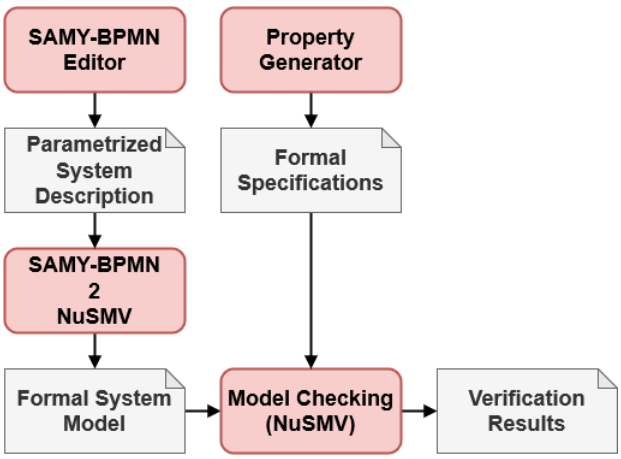


Fig. 1. Block Diagram of our proposed Toolchain

As a basis for the graphical modeling step in our approach we decided to use the *Business Process Model and Notation (BPMN)*[4], as it offers a wide range of functionality for describing complex workflows or applications. Although, BPMN has been designed for business processes, it is not exclusive to that kind of applications and is therefore also well applicable to describe technical applications including robotics.

For the actual transformation of the model into a verifiable format, we decided to use the *Atlas Transformation Language (ATL)*[6], which is a rule-based programming language for automatic model transformation.

The generation of verification properties is done with a simple command line tool, that is described in section V. Although, verification properties are necessary for model checking, the focus of this work was on model transformation. Therefore, the creation and handling of verification properties in this work might seem a bit untended.

In the following, all the just mentioned software tools and technologies, that form the underlying basis of our toolchain, are explained in more detail.

A. Business Process Model and Notation (BPMN)

BPMN is an informal specification language originally designed for modeling and documenting business processes and business workflows, where a token represents a state in the BPMN concept. It offers a number of symbols to graphically represent a workflow. In Fig. 2 you can see a simple pick and place workflow performed by a robot represented in BPMN.

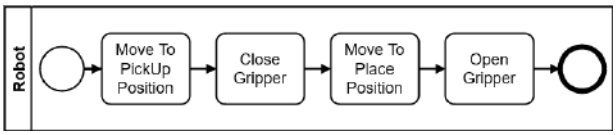


Fig. 2. Pick and Place Task represented with BPMN

The notation range of BPMN basically comprises objects of four categories, which are described in the following. Due to the very comprehensive functional extent of BPMN, only a few often used examples will be mentioned for each category.

- **Flow objects** cover activities within the workflow. An activity could be a task, which describes a concrete functional step in the diagram. The use case in Fig. 2 is displayed as a sequence of tasks. Another activities are special events such as a start node or an end node. These nodes indicate the beginning and the finalization of a workflow and are also illustrated in Fig. 2. Eventually, there are gateways in this category, that are used to interconnect other objects in a particular semantic way.
- **Connecting objects** are used to create flows of information within a BPMN diagram. This could be a directed connection between two tasks to indicate a sequence flow or an undirected association connection that shows a relationship between two connected objects.
- **Pools and swimlanes** are constructs for organizing and grouping activities. A collection of activities can visually be encapsulated within a swimlane in order to indicate a specific function or to assign them to a resource. In Fig. 2 there is shown one swimlane named *Robot*. Like in reality, a pool encompasses all swimlanes. The use of connecting objects between several swimlanes enables an appropriate information exchange between them. A pool is often used to indicate a higher-order entity such as a company, whereas swimlanes typically represent departments or particular processes.
- **Artifacts** are any kind of additional information that increases the understandability of the workflow diagram.

Examples for artifacts are data objects or textual annotations to add information to any object within a diagram.

B. Atlas Transformation Language (ATL)

The main goal of ATL is to translate an input model with a certain structure into an output model with another structure. For this, a translation file defines a set of rules that specify how particular elements from the input model are mapped into the structure of the output model. The involved components in an ATL transformation are arranged on specific semantic layers, which derive from each other from the top to the bottom. Even though, Fig. 6 already shows the transformation structure of our proposed concept, it also reveals the basic structure of ATL.

The basic or top layer determines the used model description architecture that all other components on lower levels have to conform to. Often used technologies on this layer are the *Meta Object Facility (MOF)*, the *Unified Modeling Language (UML)* or *Ecore*, which is part of the *Eclipse Modeling Framework (EMF)*.

The second layer contains the metamodels for the input and output models. A metamodel defines the structural and semantic composition of a concrete model. Compared to software development, a metamodel can be seen as a class description.

The third or bottom layer defines the actual model files for the input to and the output from the transformation. All model descriptions on this layer are concrete instantiations of the corresponding metamodels from the second layer.

The crucial step for creating an ATL transformation is the implementation of a set of rules in ATL. A rule in ATL basically maps one or more elements from the input model to the output model. With the possibility of creating helper methods and other programmatic support structures, the collection of defined rules constitutes an ATL transformation file. As the transformation file is a concrete ATL implementation, it is also located on the bottom layer. The actual translation is performed exclusively on the bottom layer, where the input model file is fed to the ATL transformation file, which then produces an output model file.

C. Symbolic Model Checking with NuSMV

For formal verification of formal system models there are basically two main approaches, the fully automatic and the interactive one. Symbolic model checking is an automatic technique for verifying formal system descriptions. A model checking tool builds a complete state space, a tree structure with all possible system states and verifies each of these states against defined specifications. As both, the system model and the verification properties, are provided in formal logic, the verification results can be considered as a mathematical proof of conformity or non-conformity of the system model against the verification properties.

NuSMV is an entirely text-based model checking tool and thus, provides its own syntax for writing verifiable model files. The verification properties can be provided in the form of *Linear Temporal Logic (LTL)*[10] or *Computation Tree Logic (CTL)*[5].

After creating a NuSMV model and defining the corresponding verification properties, the model checking process can be executed via a command line interface. In case one of the defined properties is violated in one of the system states, the model checker yields a counterexample, which basically lists the system state that does not meet the violated property.

III. MODEL CREATION

As stated in section II, we use BPMN as a basis for model creation. The main disadvantage of using BPMN to eventually create a formal system model is that BPMN can be used ambiguously. More precisely, a particular part of a process may be defined in multiple ways in BPMN. This is severely hindering for generating a formalized system model, because for formal verification a distinctive model representation is crucial in order to obtain reliable verification results. For this reason we defined a subset of BPMN which we called *SAMY-BPMN*.

A. SAMY-BPMN

In combination with strict design rules SAMY-BPMN not only supports modeling robotic workflows in an unique way, but is also able to represent a certain amount of dynamic behavior within the workflow model. SAMY-BPMN consists of nine elements, which are displayed in Fig. 3.

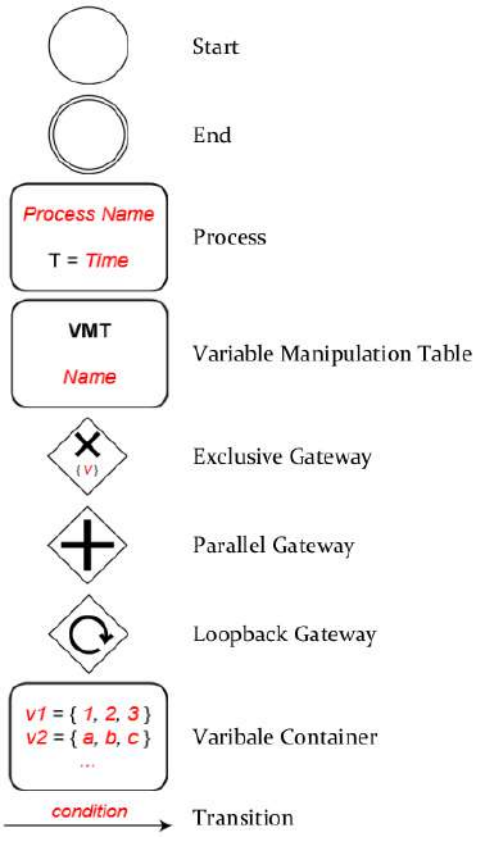


Fig. 3. Elements of SAMY-BPMN

- The **Start** element indicates the initial point of the whole workflow. By definition, a start element must appear exactly once within a workflow description.
- The **End** element is obviously the counterpart of the start element. An end element eventually concludes all possible paths through the workflow and is also meant to be used exactly once.
- A **Process** is the generic element type for any time consuming action within the workflow. Necessary parameters for a process are a name and a time delay that indicates the needed processing time of that particular task.
- A **Variable Manipulation Table (VMT)** is a special element in SAMY-BPMN, thus, it does not appear in the functional range of the conventional BPMN. As its name reveals, a VMT has the task to manipulate variables. For this purpose, the user is supposed to define a mapping scheme between two variables for a VMT.
- With an **Exclusive Gateway** a conditional routing can be represented in the workflow. The exclusive gateway takes a condition variable, which is defined in the variable container. The outgoing transitions can then be chosen according to the value of the condition variable. When an exclusive gateway is used for merging, no variable needs to be specified and the connected output becomes active if at least one input is active.
- A **Parallel Gateway** can be used to open up parallel branches in the diagram. Thus, all outgoing connections become active simultaneously in case the parallel gateway is used for splitting. Consequently, when used for merging, connected outputs only become active if all connected inputs are active.
- A **Loopback Gateway** can be used to implement a feedback loop in a workflow. All processes within the loop are executed a specified number of times or until a termination condition is met.
- The **Variable Container** is the reservoir for all variables appearing in the workflow description. Similar to the start and the end element, the variable container must appear exactly once within a workflow description, even though no variables need to be used. In that case the variable container simply remains empty. The definition of a variable requires a unique variable name and an exact range for the variable values.
- A **Transition** is the main connecting element within SAMY-BPMN. In a workflow diagram, the transitions define the sequence flow. In a special case a transition can be assigned a value from one of the variables from the variable container. In this case the transition is an outgoing from an exclusive gateway with a specified condition variable.

B. SAMY-BPMN Editor

By the customization of BPMN and the resulting subset SAMY-BPMN we are theoretically able to create workflow diagrams in a semantically distinctive way. But due to the unique and custom character of SAMY-BPMN, an appropriate editor did not exist before. So, the first software application in our verification toolchain is the *SAMY-BPMN editor*.

For developing our editor we decided to use *Graphiti*, an *Eclipse*-based framework for fast and easy development of model-based graphical diagram editors[2]. The only requirement to start with *Graphiti* is a domain model, which forms the knowledge base for storing the created diagrams semantically correct. *Graphiti* accepts domain models of various formats, we decided to create our input domain model in the *Ecore* format.

A visual representation of our SAMY-BPMN metamodel in the *Ecore* format can be seen in Fig. 4. All the elements defined by SAMY-BPMN are represented as classes with the required parameters as class attributes.

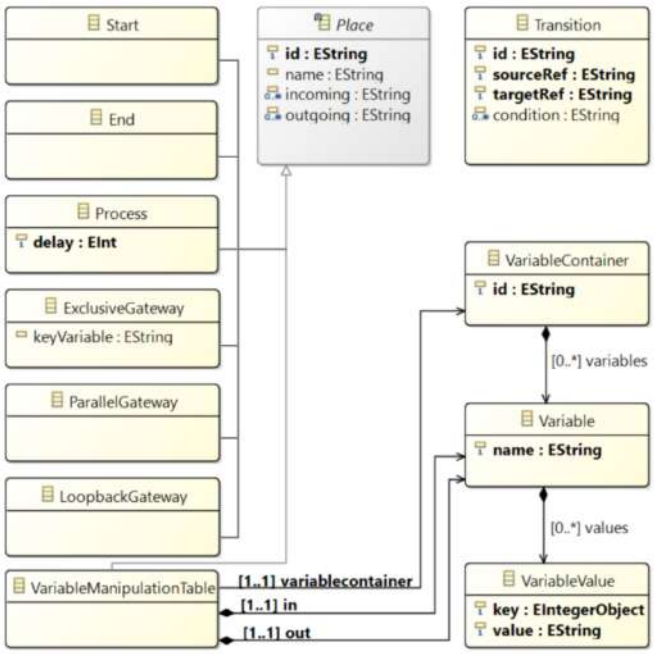


Fig. 4. Metamodel of SAMY-BPMN

Any diagram editor developed with *Graphiti* basically operates with two separate models, the domain model and the pictogram model. The domain model holds all the semantic information of a created diagram. In our editor this are parameters like process names or delays, variable definitions, element IDs or the references to transitions. The pictogram model on the other hand is solely responsible to keep track of visual information of the diagram. This could be element sizes, geometric locations of elements within the diagram or the shape of transitions. As the visual diagram information is not relevant for the further translation of a created diagram into verifiable NuSMV code, the pictogram model is obsolete for the later translation.

An editor in *Graphiti* is eventually built up by implementing a number of features for each diagram element. The two major features required by the framework are the *add*- and the *create*-feature, which are responsible for adding a diagram element to the pictogram model and the

domain model, respectively. The remaining features offered by *Graphiti* handle issues like deleting, resizing or updating elements in the diagram. After implementing the desired features for the domain elements, the editor can be built and is ready to use.

An example workflow created with the SAMY-BPMN editor is shown in Fig. 5. It shows two parallel tasks for providing and processing an object simultaneously. After performing that parallel action five time (as indicated by the counter variable *C*), a decision variable *D* is set to *TRUE* and the workflow is completed. Note, that in this example it is not specified if the particular subtasks are done by robots or human operators. The issue of resource allocation is planned to be handled in a future version of the SAMY-BPMN editor.

IV. MODEL TRANSFORMATION

Our model transformation module is based on the *Atlas Transformation Language (ATL)*. The rule-based structure of an ATL transformation provides a high amount of flexibility and thus, can be easily adapted to changes in the input or output metamodels. The transformation structure of our concept based on ATL is shown in Fig. 6.

The *Ecore*-format from the *Eclipse Modeling Framework*[11] is the basic modeling structure, to which all the components in our transformation scheme conform to. It is easy understandable, can be used graphically or text-based and is supported by a lot of frameworks and tools.

The next artifacts necessary were the metamodels for SAMY-BPMN as input and NuSMV as output. As it was already needed to create a metamodel in the *Ecore*-format of SAMY-BPMN during the implementation of the SAMY-BPMN editor, we were in the comfortable situation to use the exact same metamodel implementation in the transformation step. The metamodel for NuSMV we obtained from an third-party open-source project[7], that also coped with NuSMV-based transformation.

At the bottom layer the concrete files, that are involved in the transformation, are located. The core file here is our ATL transformation module ***SAMY-BPMN2NuSMV***. It defines the rules for mapping the elements from an SAMY-BPMN input file to an NuSMV output file. The actual file format for input and output files is *Extensible Markup Language (XML)*, which is well suitable for storing structured data in a text file. The actual file format from a diagram created in the SAMY-BPMN editor is also XML, so no intermediate conversion was needed before transforming the model.

A. Code Generation

The last step necessary before having a verifiable workflow model is to generate NuSMV code. As the output file format after transformation is XML, it can not be forwarded to the model checker directly. For this reason, we used the simple code generation framework *Acceleo*, that basically requires a template file, which forms the code skeleton of a NuSMV file and iteratively adds the necessary information from the transformed workflow model. In that way, a verifiable NuSMV file is finally generated.

V. PROPERTY GENERATION

Well defined verification properties are of high importance in order to receive reliable verification results at the end. Although, the focus of this work was on model creation and transformation, we created a simple command line tool for generating verification properties.

Our property generator takes an SAMY-BPMN diagram file as input and scans it for the contained elements. It then provides a list of found candidates for states and variables. As all processes, start and end elements in the SAMY-BPMN diagram are also transformed into NuSMV states during model transformation, the property generator interprets these elements as candidates for states as well. The variables are simply extracted from the variable container, as this is the central collecting point of all variables in a SAMY-BPMN diagram.

The next step is that the property generator provides a number of cloze sentences the user is supposed to complete. The sentences offered by the property generator represent often occurring statements and assumptions for formal verification. One example of such a sentence is as follows:

- *On all possible paths through the BPMN model there is no scenario, in which the signal **SIG** has the value **VAL** and the process **PROC** is active simultaneously.*

The bold printed placeholders form the missing information the user is supposed to provide by choosing it from the previously provided list of states and variables.

A formal specification in CTL is then created from all the fulfilled sentences by the property generator. By running in a loop, the user is able to generate an arbitrary number of properties until a termination signal is commanded. The output of the property generator is a list of formal verification properties, which can be used to finally verify the modelled and transformed workflow model with the NuSMV model checker.

VI. EVALUATION AND DISCUSSION

For the evaluation of the proposed toolchain we were grateful to receive a number of real-world workflow descriptions including industrial robots from the industrial partners of the SAMY project consortium. The evaluation approach was to conduct a comparison between the verification results achieved by our toolchain and those obtained by manual creation of the formal NuSMV models. The initial evaluation was very satisfying, but due to the low number of provided workflow descriptions, we can not consider the evaluation phase of our toolchain as completed yet. Furthermore, the provided workflows mainly consist of often used tasks such as pick and place applications. Thus, another criteria for an ongoing evaluation should be to have more diversity in the functional range of the workflow descriptions.

Although, we are highly encouraged to perform further evaluation for a better proof of quality, we were definitely able to proof the basic functioning of our toolchain. With SAMY-BPMN and the SAMY-BPMN editor we have shown that it is possible to create a description framework to

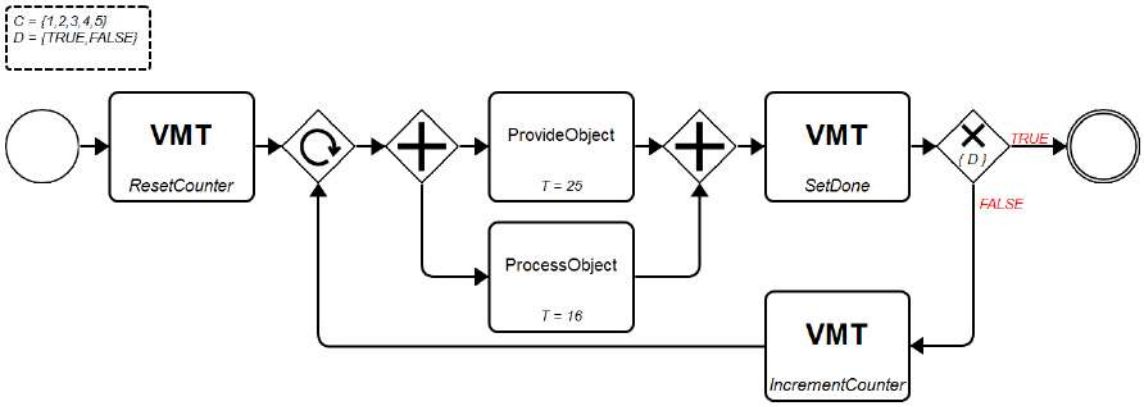


Fig. 5. Example Use Case in SAMY-BPMN

distinctively create annotated workflow descriptions for the robotics domain. With the ATL transformation unit we were able to create an automatic transformation from our custom description format SAMY-BPMN into verifiable code in NuSMV.

All together, we consider our proposed work as a lightweight software toolchain for modeling industrial workflows including robotics and transforming them into verifiable model representations.

VII. OUTLOOK

In future work, we definitely want to put a higher focus on the creation of the verification properties. The reliability and quality of the verification results depend on the formal properties in the same amount as it does on the formal model. In other words, the expressiveness of the verification results directly correlate with the quality of the formal model and the formal properties.

Another idea for future work is to bypass the need of manual workflow description completely. Despite the amount of automation in our toolchain, the user still has to describe the workflow in the SAMY-BPMN editor by himself. This, at the end, can still be a source of errors. Therefore, it would be useful to directly be able to transform a robot program into verifiable code. This would extend our toolchain of automatic model transformation by an automatic model creation.

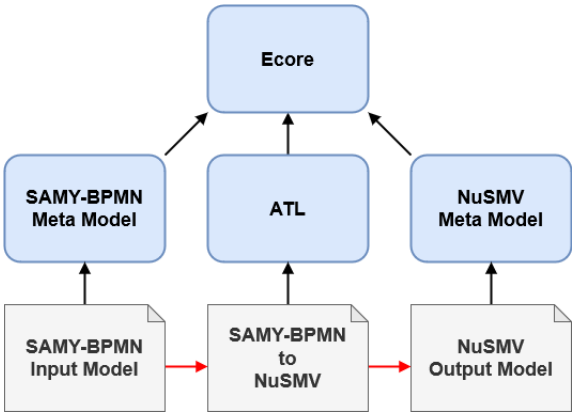


Fig. 6. Proposed Translation Scheme

ACKNOWLEDGMENT

The work of this paper was funded by the Federal Ministry for Climate Protection, Environment and Energy, Innovation and Technology (BMK), and carried out within the framework of the programme ”Production of Future” under the grant project number ”877362” within the project ”SAMY – Semi-Automated Modification in Control Programmes of Industrial Collaborative Robotic Systems”.

REFERENCES

[1] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[2] C. Brand, M. Gorning, T. Kaiser, J. Pasch, and M. Wenz, “Development of high-quality graphical model editors,” *Eclipse Magazine*, vol. 1, 2011.

[3] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri, “Nusmv: A new symbolic model verifier,” in *International conference on computer aided verification*. Springer, 1999, pp. 495–499.

[4] R. Dijkman, J. Hofstetter, and J. Koehler, *Business Process Model and Notation*. Springer, 2011, vol. 89.

[5] K. Heljanko *et al.*, *Model checking the branching time temporal logic CTL*. Citeseer, 1997.

[6] F. Jouault, F. Allilaire, J. Bézivin, I. Kurtev, and P. Valduriez, “Atl: a qvt-like transformation language,” in *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, 2006, pp. 719–720.

[7] kiyo07, “Umlstatemachine2nusmv,” https://github.com/kiyo07/ UML-StateMachine2NuSMV, 2015.

[8] M. Rathmair, T. Haspl, T. Komenda, B. Reiterer, and M. Hofbaur, “A formal verification approach for robotic workflows,” in *2021 20th International Conference on Advanced Robotics (ICAR)*. IEEE, 2021, pp. 670–675.

[9] M. Rathmair, C. Luckeneder, T. Haspl, B. Reiterer, R. Hoch, M. Hofbaur, and H. Kaindl, “Formal verification of safety properties of collaborative robotic applications including variability,” in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*. IEEE, 2021, pp. 1283–1288.

[10] K. Y. Rozier, “Linear temporal logic symbolic model checking,” *Computer Science Review*, vol. 5, no. 2, pp. 163–203, 2011.

[11] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro, *EMF: eclipse modeling framework*. Pearson Education, 2008.

[12] F. Vicentini, M. Askarpour, M. G. Rossi, and D. Mandrioli, “Safety assessment of collaborative robotics through automated formal verification,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 42–61, 2019.