2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561670

Optimizing Keypoint-based Single-Shot Camera-to-Robot Pose Estimation through Shape Segmentation

Jens Lambrecht[1], Philipp Grosenick[1] and Marvin Meusel[1]

***Abstract*— We introduce an optimization method for recent approaches on keypoint-based pose estimation of robotic manipulators utilizing monocular images. The method takes into account the segmented shape of the robot using Convolutional Neural Networks and a keypoint refinement through a set of score values. To this end, the primal 2D keypoint detection is exploited as an initial guess for further shape-based keypoint adjustments. Afterwards, the overall methods incorporates a perspective-n-point algorithm using 3D point correspondences that are derived by forward kinematics. We hereby complement an existing public dataset with annotated segmentations of a Universal Robot UR5 manipulator. The evaluation of the optimization approach shows clearly that noise on the initial keypoint detection can be suppressed and minimized. Furthermore, the overall success rate of the perspective transformation can be enhanced towards more than 90%. Thus, the overall methods is applicable for single-shot pose estimation. The evaluation results also show a significant reduction of the standard deviation of the resulting pose estimation. Consequently, the proposed optimization positively affects applicability and precision.**

I. INTRODUCTION

Camera-to-robot pose estimation based on single monocular camera images is a common challenge in robotics.On the basis of recent advancements in computer vision and machine learning, in particular Deep Learning, marker-less object pose estimation through monocular camera images becomes a viable utilization in robotics, e.g. in terms of obstacle avoidance through predicted depth images [1], six degrees of freedom (DoF) self-localization [2] as well as pose estimation for grasping of known [3] and unknown [4] objects. Nevertheless, pose estimation of robotic manipulators is a major challenge because of the non-rigid, i.e. articulated, object structure. Camera-to-robot pose estimation targets the camera extrinsics, i.e. a static or dynamic transformation between camera and robot base, e.g. in order to transform coordinates from the camera coordinate system towards the robot coordinate system. The cameras intrinsics are mostly assumed to be known based on standard camera calibration, but can also be subject to the overall machine learning approach [3], [5]. Within the general application range towards robotic manipulators, one distinguishes applications in regard to the positioning and movement of the camera. Classic hand-eye-calibration (HEC) techniques cover robot-guided cameras following the eye-in-hand principle [6] as well as cameras statically placed in the workspace [7] or at the ceiling [8]. Furthermore, one distinguished in process-integrated, i.e. online [9], and separate offline [10] calibration methods.

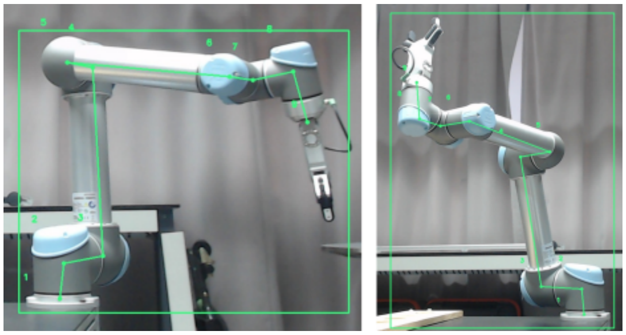

Fig. 1. 2D keypoint detection along the kinematic structure establishing a skeleton model of the robot

Recently. we introduced a novel method for robot pose estimation from monocular images through a deep neural network for keypoint detection (s. Fig. 1) and knowledge about the 3D coordinates of the keypoints as a result of forward kinematics [11]. Both, 3D keypoint information with respect to the robot base and the corresponding 2D image coordinates of the keypoints were utilized to perform a perspective transformation, i.e. perspective-n-point (PnP). As a result, we noticed that the method, when utilized as a single-shot method, still suffers from mirror effects and moderate success rates related to imprecisions of the 2D keypoint detection and general issues of PnP algorithms. Hence, the approach works better as few-shot method using extrinsic guess extension of PnP. In [12], we determined that the incorporation of synthetic image data improves both success and precision, due to automatically and precise annotation of keypoints compared manual labels. Nevertheless, the method is still rather used as a few-shot method and is performing slightly worse compared to classic offline-calibrated marker-based robot pose estimation. In the context of this work, we contributed a huge dataset[1] of synthetic and real images of a Universal Robot UR5 including annotations for keypoints as well as semantic segmentations of the robot shape. Lee et al. [13] recently proofed that our method is transferable towards other industrial manipulators, can be trained by exclusively synthetic data incorporating extensive domain randomization and outperforms common marker-based approaches for HEC. However, precision and robustness as a single-shot method are still improvable.

[1] Jens Lambrecht, Philipp Grosenick and Marvin Meusel are with the Chair Industry Grade Networks and Clouds, Faculty of Electrical Engineering and Computer Science, Technische Universität Berlin, 10587 Berlin, Germany lambrecht@tu-berlin.de

[1] https://tubcloud.tu-berlin.de/s/pFiz229pD25JKBm

Within this paper, we face the challenge of enhancing the precision and robustness of the 2D keypoint detection in order to improve accuracies and success of our approach as a single-shot method. We follow the hypothesis that a solely keypoint-specific detection neglects exploitable information about the overall geometric proportions of the keypoint distribution. According to our assumption, a more precise and robust keypoint detection can be achieved taking into account an optimization that exploits the fixed proportions of the keypoint structure supported by an adjustment based on the robot shape which is provided through semantic segmentation.

II. Related Work

The aforementioned Lee et al. [13] introduce a so-called Deep Robot-to-camera Extrinsics for Articulated manipulators (DREAM) framework that basically follows our approach in regards to 2D/3D keypoint correspondence and the usage of PnP. Whereas the general approach is the same, Lee and colleagues are training solely on synthetic images involving extensive domain randomization to bridge the reality gap. Their robot-specific datasets consist each of 100k synthetic images for the three considered robots. As an addition, they assign the keypoint configuration automatically from given joint configurations in the URDF (Unified Robot description format) of the Robot Operating System (ROS). In contrast to our work, they do not rely on standard object detection Convolutional Neural Networks (CNNs), but utilize an auto-encoder network based on the VGG-19 architecture [14] that outputs belief maps for each keypoint representing the likelihood that the keypoint is projected onto that pixel. In terms of evaluation, they also experiment with a ResNet based encoder [15]. Basically, their work proofs the transfer of the overall method towards other robot types, i.e. Franka Emika Panda and KUKA LBR. The additional two-armed Rethink Baxter robot bares general issues because of the symmetry. Experiments and results are presented for the Panda robot introducing 2D and 3D metrics. The 2D metric is the percentage of correct points (PCK) [16], which indicates whether the re-projected keypoints are within a pixel threshold around the ground truth keypoints. The 3D evaluation is taking into account the final camera-to-robot pose calculating the average distance (ADD) [17], which is the Euclidean distance of all resulting 3D keypoints to their transformed correspondences combining rotation and translation errors. The criticism about these metrics is that mainly the robustness and the consistency of the keypoints is measured, rather than the resulting accuracy of the pose estimation. Mirror effects and other issues of utilized EPnP algorithm in regards to an imprecise 2D keypoint could not be covered. Furthermore, these metrics basically only allow evaluation on synthetic data. However, a 17k real image data set is presented incorporating a depth-image-based ground truth for pose estimation of the robot using Dense Articulated Real-Time Tracking (DART) [18]. The depth data acquisition and evaluation involve different cameras. Best results are achieved for the RealSense D415 (80% for an ADD threshold of 20 mm), whereas the XBox 360 Kinect performs significantly worse (approx. 25% for the same threshold). Different neural network architectures tend to have a minor impact. Finally, evaluation is done in a static camera scenario comparing the results for 18 different joint configurations with the respective ROS package for HEC incorporating five ArUco fiducial markers [19]. The median ADD error for the DREAM method is below 20 mm, but the max range of the error is still about 50 mm. However, this is still more precise than the standard marker-based HEC method which relies on a minimum of three input images. The size of the markers and distance to the robot remain unknown and the comparison is only considering a static camera setting. As a final statement, the authors recommend to utilize more than one input image for DREAM as precision and scattering is significantly minimized.

Hu et al. [20] introduce a segmentation-driven approach for six DoF pose estimation of rigid objects. Every visible part of the objects is considered as patch which contributes a local pose prediction in the form of 2D keypoint locations. The introduced method is mainly intended to overcome occlusion-based pose estimation degradation and relies on a 3D model of the respective objects. Miseikis et. al [21] are presenting the extension of a multi object CNN towards robot base and joint pose estimation. They integrate a high-precise marker-based tracking system in order to generate training data and also derive 3D shapes of the current robot configuration through projecting 3D shape models on the images. Hereby, the segmentation is a side-effect of using the external marker-based reference system. Although, the resulting precisions for robot joints and base pose estimation seem quite accurate, the approach is poorly transferable towards different cameras as the perspective transformation as well as camera's intrinsics are fixed and inherently involved in the learning.

In comparison to former keypoint-related approaches, we intend to incorporate segmentation data and optimize the 2D keypoint estimations. This motivation is based on the fact that PnP solvers are quite sensitive towards particular 2D keypoint deviations and a more robust and precise single-shot method is aspired. Consequently, the main contribution of the paper is an improvement of markerless keypoint-based pose estimation methods for robotic manipulators through an optimization based on the shape of the manipulator. In this context, we provide an extension on our public dataset and introduce a specific implementation and evaluation of the optimization towards pose estimation of a UR5 manipulator.

III. Overall Approach

The keypoint configuration for the UR5 comprises nine keypoints along the kinematic chain defined by intersection points of axes and joints using the URDF description. The axis values are derived through the robot's control interface and forwarded to forward kinematics in order to calculate the 3D keypoint positions in respect to the robot base, s. [11] and [12]. Within our novel optimization approach, we preserve the information about the 3D skeleton model and its

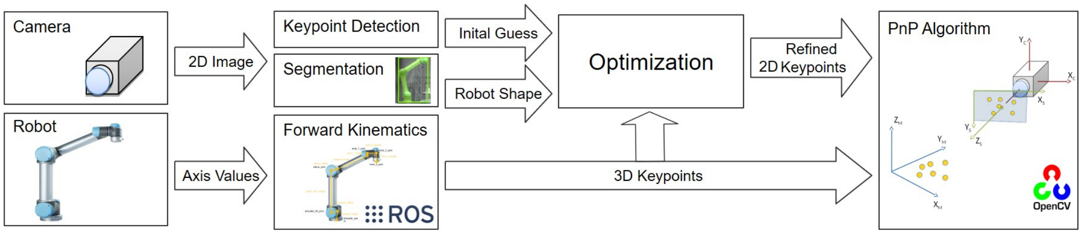

Fig. 2. Pose estimation pipeline based on segmentation-refined 2D keypoints as well as 3D keypoint correspondences calculated through forward kinematics

proportions transferring and adjusting within the 2D image plane instead of solely detecting keypoints separately (s. Fig. 2). For this purpose, the 2D segmentation of the robot serves to align the projected skeleton model within the inside of the shape.

A. 2D - 3D Perspective Transformation

Pose estimation between camera and robot is about estimating a matrix $\boldsymbol{T}$ transforming 3D point vectors in the robot coordinate system $\boldsymbol{b}$ towards point vectors in the 3D camera coordinate system $\boldsymbol{y}$ following (1).

$$\boldsymbol{y} = \boldsymbol{T} \cdot \boldsymbol{b} \tag{1}$$

Extending this problem formulation by incorporating the 2D keypoints in the image plane $\boldsymbol{a}$ and the intrinsic camera matrix $\boldsymbol{P}$ leads to (2) and (3) involving the scaling factor h. As we rely on a calibrated camera, the intrinsics are assumed to be known.

$$h \cdot \boldsymbol{a} = \boldsymbol{P} \cdot \boldsymbol{T} \cdot \boldsymbol{b} \tag{2}$$

$$h \cdot \boldsymbol{a} = \begin{bmatrix} f_X & 0 & c_X \\ 0 & f_Y & c_Y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_X \\ r_{21} & r_{22} & r_{23} & t_Y \\ r_{31} & r_{32} & r_{33} & t_Z \end{bmatrix} \cdot \boldsymbol{b} \tag{3}$$

The class of PnP algorithms solves this problem statement for n given point correspondences providing $\boldsymbol{T}$ which includes a rotatory matrix $\boldsymbol{R}$ and a translatory vector $\boldsymbol{t}$. Based on a performance comparisons in our former work, we are using the EPnP [22] algorithm.

B. Optimization Problem

From a global perspective, the optimization problem on the 2D image plane involves varying translation and rotation of the skeleton model in Cartesian space in order to determine an optimized fit of the skeleton model into the robot shape.

1) Initialization: As we aim to keep the optimization problem efficient, it makes sense to avoid a global brute-force-based optimization. Moreover, we intend to investigate if the existing keypoint baseline could serve as initial guess that is refined through the segmentation-based optimization.

2) Measure: In order to assess the quality of the optimization, we need to define a shape-based measure, i.e. the pose score that indicates the quality of the overall keypoint detection. Consequently, the local optimization can be targeted and stopped when reaching a specific threshold or a noticeable local maximum. In order to assess the keypoint distribution, we aspire to consider the whole skeleton model instead of using only particular keypoints. Furthermore, we rely on type-specific assumptions regarding the symmetry of the robot. On the one hand that may make it hard to transfer the approach to other kinematic structures, but on the other hand the exact localization of the keypoints is inherently based on joint positions that incorporate the symmetries of the kinematic structures. Thus, we aspire to define a set of different rules that form a quantitative measure, but can also be adapted towards different kinematic structures.

IV. Segmentation Dataset & Training

The aforementioned public dataset was enhanced by about 2,800 manual annotations for semantic segmentations of the UR5 robot. Thereof, 95% of the images were used for training and 5% were separated as test-split. As network topology we choose the Mask RCNN Inception ResNet detector implementation [23] with Atrous convolution [24] from TensorFlow model zoo. We utilized a pre-trained model on the COCO dataset [25] that was trained on our dataset for 25k steps with a batch size of 1 and a constant learning rate of $3 \cdot 10^{-5}$. For data augmentation, we used random horizontal flips. Fig. 3 shows an exemplary result of the shape segmentation. The network provides the following main performance values according to the COCO metrics [25]: mean average precision ($mAP = 0.913$) and mean average recall ($AR@1 = 0.940$).

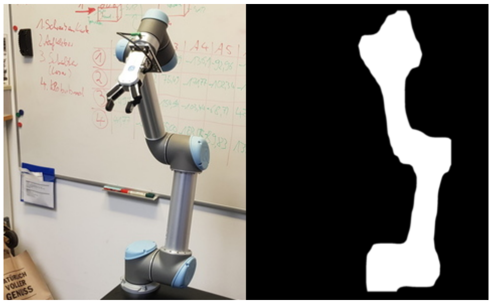

Fig. 3. Segmentation: original image (left) and binary robot mask (right)

V. Optimization Implementation

Objective of the optimization is to provide a refined pose estimation based on the binary mask and the corresponding robot skeletons. In terms of the implementation of the optimization approach, we consider three subtopics: the optimization algorithm, pose scoring and the choice of initialization values. For the implementation of the optimization algorithm, we used the SciPy library, which contains different algorithms for local and global optimization. Whereas global optimization is evaded, the following local optimizers are examined: Nelder-Mead, Powell, CG, L-BFGS-B, BFGS and SLSQP.

A. Pose Scoring

For the overall optimization it is inevitable to assess variations towards transforming the skeleton model within the shape through a cost or scoring function, i.e. the measure. To this end, we assign a score based on the projected 2D skeleton model. That score is subsequently utilized in a local search by the optimization algorithm. Initially, we experimented with a unified score function that assesses solely the keypoints based on the distance to the border of the shape. This approach was quite intuitive because of the symmetry of the UR5 robot, but was neglected because it performed significantly worse in comparison to a combination of different score values that are derived individually for keypoints and connecting lines along the skeleton model. Consequently, individual score methods are combined and accumulated in order to calculate a total score. All of these scores have different weights that define their influence on the final score value. These parameters can be optimized manually based on heuristics, but rather be refined using automatic tuning (s. Sec. V-B). In the following, the particular scores that are proposed for the UR5 are explained.

1) Point Score: Every robot keypoint is scored by its absolute distance in pixels to the edge of the mask. That distance is calculated by using ray tracing in horizontal, vertical and diagonal directions. The score is set to -1 if the point is outside the mask. Consequently, this score will result in higher values if the respective keypoint is in the middle of the robot's shape and gets penalized if the points are outside of the mask.

2) Line Score: Additional points between the robot keypoints are generated and scored like a point score. This will result in good scores if also the connecting lines between the keypoints are inside the robot shape. This is an essential refinement of the point score. Otherwise, the optimization algorithm would often put the robot in wrong configurations positioning only the keypoints inside the shape, but positioning connecting lines of the skeleton model outside.

3) Bounding Box Score: The bounding box of the overall robot mask is used to asses the general scale of the skeleton model. The less of the bounding box is filled by the keypoints, the higher the penalty on the score. This prevents mainly a very small scale that allows to position all points in the middle of the mask.

4) Keypoint Zero Score: The first keypoint of the robot represents the center of the robot's base. The specific Keypoint Zero Score is its distance from the lowest point in the mask. It penalizes the skeleton pose, if keypoint zero is far from the lowest point of the robot's shape. This underlies the assumption that the robot is mounted on a plane parallel to the ground and the image is not rotated. Although this is a limitation of the application context, this score has proven to be highly effective when starting from bad initialization values, as it makes it easier for the optimization algorithm to determine adequate translation values. This score can be adjusted respectively if the robot is mounted on the wall or ceiling.

5) Initialization Distance Score: The distance of all keypoints from their initialization position is compared. This results in a penalty the larger the distance is. This score ensures that the optimization step does not make changes to the good initialization values, unless it results in a significant score increase. It will also prevent extreme changes on already good initialization values. This score is beneficial when working with good initial values, but may be omitted when starting with random poses.

6) Point Deviation Score: The maximum deviation of the particular point scores represents a penalty for the overall score. The objective of this additional score is to take overlapping mask areas into account, where the center of the mask is not necessarily the best location for the keypoints. Assuming that all points have a similar distance to the border of the robot leads to the conclusion that the points should have a similar point score as well. If the maximum deviation of the points scores is high, all the points will not have a similar distance to the border.

B. Automatic Parameter Tuning

The respective weight parameters of the particular scores influence the overall score value and thereby the pose estimation quality significantly. Even small changes might result in major differences towards the predicted pose. As the weight parameters partially rely on each other and are interconnected, we aspire an automatic tuning of the parameter values instead of manual tuning. Hence, we define a comparison score S following (4) that calculates the ratio of the accumulated normalized keypoints deviation in comparison to the ground truth, i.e. manual annotations, taking account an initial solely keypoint detection d_{key} and the optimized or respectively adjusted keypoint values d_{opt}. As a result, S denotes the quality of the a keypoint refinement comparing it to the standard keypoint approach and using the ground truth as an overall reference. Hence, this comparison score can also be exploited to evaluate the general success of the optimization within the 2D plane.

$$S = \frac{\sum d_{key}}{\sum d_{opt}} \tag{4}$$

The implementation of the automatic parameter tuning is done through the SciPy library using the optimization algorithm L-BFGS-B. In terms of parameter optimization ground truth values from a small subset of the overall dataset, i.e. 30 images, have been utilized.

C. Initialization Values

As we intend to use a local optimization algorithm, initial values have a significant impact on the optimization. In order to find out an appropriate method for defining adequate initial values for the pose estimation, we implement and compare three different approaches that are explained in the following.

1) Constant Values: In this approach, the initial values are constant, which means that we start with a constant translation and a rotation vector projecting the skeleton model to the center of the image with a bounding box coverage of about 20% assuming a side view for orientation setting.

2) Rough Brute Force: Since it is apparent that a complete brute force approach takes far too much time, we implement a rough brute force approach using large step sizes in order to find an initial guess. To save more time, we also use a simplified scoring for guessing the initial values. This is based on a comparison of ratios of sides between the outer points of the mask and the outer points of the projected skeleton model.

3) Keypoint Approach: The keypoint approach uses the re-projected coordinates of the baseline keypoint detection algorithm using the EPnP. This approach can create initial values which have a rather good shape of the skeleton, but assumes that initial PnP is successful. If this is not the case, we fall back on the initial 2D keypoints before being re-projected.

VI. Experiments & Evaluation

In terms of evaluation, we initially assess the different initialization approaches. Afterwards, we compare the optimization approach to the former keyoint approach regarding 2D image coordinates and Cartesian pose estimation. As a result of our former work, the keypoint detection model trained on 10k synthetic images turned out to be the most precise model [12]. Consequently, this model acts as keypoint baseline. In regards to the image plane, manual labels are exploited as ground truth as there is no adequate synthetic ground truth for real evaluation images. Fig. 4 shows exemplary results of the keypoint refinement. As 2D metrics, we consider the already introduced normalized point deviation d and the related comparison score S, that were already introduced in terms of auto tuning the optimization parameters (s. Sec. V-B). An initial comparison of the considered optimization methods (s. Sec. V) does not show a significant performance difference. Nevertheless, the Nelder-Mead algorithm has the lowest deviation and the lowest median value as well as no noticeable outliers. Thus, we chose the Nelder-Mead algorithm for all further evaluations.

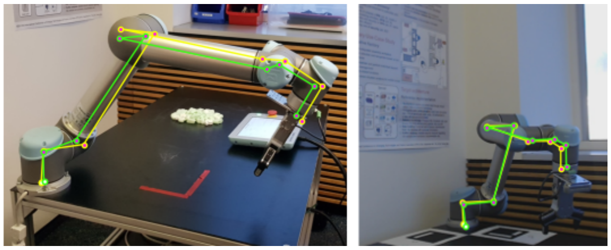

Fig. 4. Skeleton model and keypoints for the keypoint baseline (yellow) and optimization approach (green)

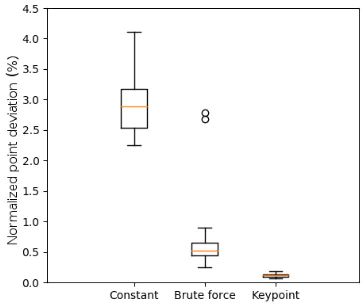

Fig. 5. Comparison of initialization approaches showing the distribution of the accumulated, normalized keypoint deviation to ground truth data

A. Initialization Methods

We performed a comparison (s. Fig. 5) between the particular initialization methods (s. Sec. V-C) on a reduced test-split of 50 images. In order to provide an adequate comparison, each initialization method and the respective optimization comes with individually tweaked weights for the scoring parameters (s. Sec. V-B). The results emphasize that the keypoint initialization is vastly superior to the other initialization methods. Results using the constant or the brute force initialization tend to run into local minima, as they often start far from the best possible fit. Utilizing a global optimization approach instead was neglected, as the computing time inhibits a practical usage within most industrial applications of the overall method. Consequently, all further results incorporate the keypoint initialization method.

B. 2D Keypoint Comparison

1) Keypoint Deviation: We utilized the same evaluation dataset to compute the accumulated, normalized position deviations comparing the optimization with the keypoint detection baseline (s. Fig. 6, left). The corresponding statistical characteristics for the optimization method are depicted in Tab. I. The results show that a major effect of the optimization is a reduction of the standard deviation, which denotes an increase of the robustness of the overall methods. This effect can be confirmed by adding artificial noise to the 2D keypoint detection. Fig. 6 shows the results for different noise values showing that the optimization can suppress im-

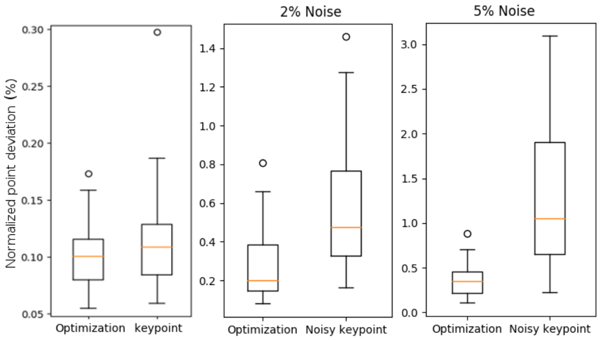

Fig. 6. Total pixel deviation for optimization and keypoint baseline (left) and with additional Gaussian noise of 2% (middle) and 5% (right)

TABLE I

STATISTICAL CHARACTERISTICS FOR THE DISTRIBUTIONS OF THE KEYPOINT DEVIATION IN REGARDS TO GROUND TRUTH DATA

	Optimization	Keypoint Baseline
Average Point Deviation	0.1022%	0.1130%
Median Point Deviation	0.1011%	0.1091%
Standard Deviation	0.0274%	0.0393%

precisions of the initial keypoint detection and Tab. II shows the comparison score S for the noise levels. However, as the ground truth comprises manual annotations, conclusions about an increase of the Cartesian pose estimation are not directly given.

TABLE II

COMPARISON SCORE IN REGARDS TO GROUND TRUTH DATA WITH ADDITIONAL GAUSSIAN NOISE OF 2% AND 5%

	Comparison Score
Ground Truth (0% noise)	1.0905
2% Artificial Noise	2.127
5% Artificial Noise	3.838

C. 3D Cartesian Comparison

We evaluate the Cartesian performance of the optimization approach taking into account the EPnP algorithm and the synthetic keypoint detection model as baseline. We consider two setups in order to assess the robustness against random robot poses and translation deviation compared to a marker-based pose estimation.

1) Robustness Against Varying Robot Poses: In this experiment, we consider a static camera at a fixed distance and a randomly moving robot. The camera is installed in approximately two meter distance to the robot while the robot is moving randomly every joint. We repurpose an 100 images test-dataset with a image resolution of 1280 x 720 pixels and respective correspondences of 2D and 3D keypoint positions. As the distance between camera and robot is static for this setting, we expect small deviations for the robot pose estimation. By assuming to have a normal distribution, the standard deviation σ of the translation vector is derived as an indicator for the overall pose dispersion. As we are using an additional RANSAC algorithm in order to neglect keypoint outliers, we also consider the success rate of the PnP depending on the maximum projection error (MPE). The keypoint baseline provides a success rate between 22.03% and 77.97% for MPEs between 3 and 11. For the same MPE range, the optimization approach gives success rates between 58% and 93.3%. Nevertheless, the resulting pose values does not show a significant difference and σ is also slightly lower for the keypoint baseline in this setting. However, it can be shown that the general applicability of the overall method toward a single-shot pose estimation could enhanced due to the optimization.

2) Fiducial Marker Comparison: As we do not have access to a high-precision position measuring system, we compare the absolute accuracy of our method to a fiducial marker pose estimation. To increase complexity, robot joints and camera are non-static in this setting. The camera is manually positioned randomly in a distance range between one and three meters. ArUco markers with edge length of 150 mm are placed next to the robot (s. Fig. 7). The markers have been calibrated to the robot base coordinate systems in order to enable direct pose comparison. We reuse another 100 images evaluation dataset and corresponding joint angles. The average position deviations and the corresponding standard deviations for optimization and keypoint baseline are summarized in Tab. III. Whereas the absolute position deviation is slightly better for the optimization in this setting, this value may be still biased due to inaccuracies of the marker calibration. Nevertheless, it is remarkable that the optimization halves the standard deviation of the position deviation. These results indicate that the optimization approach contributes to enhancements of the overall pose estimation especially regarding robustness even though a validation through a high accuracy measuring system is still pending.

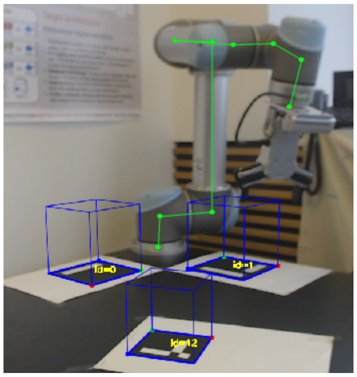

Fig. 7. Pose comparison showing marker and optimized skeleton detection

VII. CONCLUSION

We presented an optimization approach for keypoint-based marker-less pose estimation of robotic manipulators . We could show that the proposed extension improved the precision as well as the determination of the Cartesian robot pose. The overall method relies on precise segmentation. Consequently, the impact of the mask quality has to be investigated more in detail. Moreover, we consider the automatic adjustments of the score values as a potential issues for the precision of the overall method as it relies on manual annotation that is in general subject to a limited and fluctuating quality. Future work takes into account adjusting the score values based on highly precise synthetic data as well as a more detailed assessment of the absolute accuracy.

TABLE III

STATISTICAL CHARACTERISTICS FOR THE POSITION COMPARISON IN REGARDS TO A MULTI-MARKER POSE DETECTION

	Optimization	Keypoint Baseline
Average Position Deviation (m)	0.063064	0.078154
Standard Deviation (m)	0.040307	0.088089

References

[1] L. Xie, S. Wang, A. Markham and N. Trigoni, "Towards Monocular Vision based Obstacle Avoidance through Deep Reinforcement Learning," Robotics: Science and Systems (RSS) - workshop on New Frontiers for Deep Learning in Robotics, 2017.

[2] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-DoF global localization in outdoor environments," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, 2017, pp. 1525-1530, doi: 10.1109/IROS.2017.8205957.

[3] Y. Xiang, T. Schmidt, V. Narayanan and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," Robotics: Science and Systems (RSS), 2017.

[4] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley and K. Goldberg, "Learning ambidextrous robot grasping policies," Science Robotics, vol. 4, no. 26, 2019, doi: 10.1126/scirobotics.aau4984.

[5] J. Rambach, C. Deng, A. Pagani and D. Stricker, "Learning 6DoF Object Poses from Synthetic Single Channel Images," IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Munich, 2018, pp. 164-169, doi: 10.1109/ISMAR-Adjunct.2018.00058.

[6] D. Morrison, P. Corke and J. Leitner, "Closing the loop for robotic-grasping: A real-time, generative grasp synthesis approach," Robotics: Science and Systems (RSS), 2018.

[7] J. Ilonen and V. Kyrki, "Robust robot-camera calibration," 15th International Conference on Advanced Robotics (ICAR), Tallinn, 2011, pp. 67-74, doi: 10.1109/ICAR.2011.6088553.

[8] A. Feniello, H. Dang and S. Birchfield, "Program synthesis by examples for object repositioning tasks," IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 4428-4435, doi: 10.1109/IROS.2014.6943189.

[9] K. Pauwels and D. Kragic, "Integrated on-line robot-camera calibration and object pose estimation," IEEE International Conference on Robotics and Automation (ICRA), Stockholm, 2016, pp. 2332-2339, doi: 10.1109/ICRA.2016.7487383.

[10] D. Park, Y. Seo, and S. Y. Chun, "Real-time, highly accurate robotic-grasp detection using fully convolutional neural networks with high-resolution images," arXiv:1809.05828, 2018.

[11] J. Lambrecht, "Robust Few-Shot Pose Estimation of Articulated Robots using Monocular Cameras and Deep-Learning-based Keypoint Detection," IEEE International Conference on Robot Intelligence Technology and Applications (RiTA), Daejeon, 2019, pp. 136-141, doi: 10.1109/RITAPP.2019.8932886.

[12] J. Lambrecht and L. Kästner, "Towards the Usage of Synthetic Data for Marker-Less Pose Estimation of Articulated Robots in RGB Images," IEEE International Conference on Advanced Robotics (ICAR), Belo Horizonte, 2019, pp. 240-247, doi: 10.1109/ICAR46387.2019.8981600.

[13] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox and S. Birchfield, "Camera-to-Robot Pose Estimation from a Single Image," IEEE International Conference on Robotics and Automation (ICRA), Paris, 2020, pp. 9426-9432, doi: 10.1109/ICRA40945.2020.9196596.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," International Conference on Learning Representations (ICLR), San Diego, 2015.

[15] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," Computer Vision - ECCV 2018, Lecture Notes in Computer Science, 2018, vol. 11210, Springer, doi: 10.1007/978-3-030-01231-1_29.

[16] J. Tremblay, T. To, A. Molchanov, S. Tyree, J. Kautz and S. Birchfield, "Synthetically Trained Neural Networks for Learning Human-Readable Plans from Real-World Demonstrations," IEEE International Conference on Robotics and Automation (ICRA), Brisbane, 2018, pp. 5659-5666, doi: 10.1109/ICRA.2018.8460642.

[17] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," Computer Vision - ACCV, Lecture Notes in Computer Science, pp. 548-562, Springer, 2012.

[18] T. Schmidt, R. Newcombe and D. Fox, "DART: dense articulated real-time tracking with consumer depth cameras," Autonomous Robots, vol. 39, no. 3, pp. 239-258, Springer, 2015.

[19] S. Garrido-Jurado, R. M. noz Salinas, F. J. Madrid-Cuevas and M. J. Marn-Jimenez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognition, vol. 47, no. 6, pp. 2280-2292, 2014.

[20] Y. Hu, J. Hugonot, P. Fua and M. Salzmann, "Segmentation-Driven 6D Object Pose Estimation," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, 2019, pp. 3380-3389, doi: 10.1109/CVPR.2019.00350.

[21] J. Mišeikis, I. Brijačak, S. Yahyanejad, K. Glette, O. J. Elle and J. Torresen, "Two-Stage Transfer Learning for Heterogeneous Robot Detection and 3D Joint Position Estimation in a 2D Camera Image Using CNN," International Conference on Robotics and Automation (ICRA), Montreal, 2019, pp. 8883-8889, doi: 10.1109/ICRA.2019.8794077.

[22] V. Lepetit, F. Moreno-Noguer and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," International Journal of Computer Vision, vol. 81, pp. 155-166, 2008.

[23] HK. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[24] L.C. Chen, G. Papandreou, F. Schroff and H. Adam, "Rethinking atrous convolution for semantic image segmentation," arXiv:1706.05587, 2017.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, Springer, pp. 740-755, 2014.