6DoF Pose-Estimation Pipeline for Texture-less Industrial Components in Bin Picking Applications

Andreas Blank[1], Markus Hiller[2], Siyi Zhang[1], Alexander Leser[1],
Maximilian Metzner[1], Markus Lieret[1], Jörn Thielecke[2], and Jörg Franke[1]

[1] Institute for Factory Automation and Production Systems (FAPS), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
email: {andreas.blank,joerg.franke}@faps.fau.de

[2] Institute for Information Technologies (LIKE), Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
email: {markus.hiller, joern.thielecke}@fau.de

***Abstract*—Over the next few years, autonomous robots and functionalities are expected to gain increased importance for the shop floor. Perception and the derivation of autonomous behavior is of crucial importance in this context. We present a combined object recognition and pose estimation pipeline to generate pose estimates with six degrees of freedom (6DoF) for bin picking, specifically targeting the suitability for challenging scenarios with texture-less, metallic parts in industrial environments. The pipeline is based on open source algorithms, combining Convolutional Neural Networks (CNNs) and feature-matching methods to create an effective 6DoF pose estimate. We evaluate our approach on several industrial components using a articulated arm robot to guarantee a high level of comparability during the different measurement runs. We further quantify the results using known error metrics for pose estimation, compare the results to established approaches and provide statistical insight into the achieved outcomes to assess the robustness and reliability.**

I. Introduction

Short product life cycles, volatile markets, an increasing amount of product variants and complex products pose challenges to the manufacturing industry [1]. Therefore, the use of versatile robots bears potential for increasing the flexibility of production as well as for cost savings.

In production, supplied or self-manufactured components are often stored randomly stacked in small load carriers (SLC). Therefore, bin picking has a high relevance for commissioning, for packaging and for providing parts to subsequent processes. Whereas conventional handling tasks can be solved by engineering, bin picking is still a field of research. Especially when metallic non-everyday objects have to be handled.

While classical image processing techniques still have a predominant role for industrial purposes, neural networks and 3D-CAD-based template matching methods for object recognition and pose estimation have become increasingly important in several areas of robotics research, ranging from general 3D object recognition [2], [3] to environment modelling [4]. A wide range of algorithms has been proposed, with most of them, however, being pre-trained and optimized for everyday objects [2], [3], [5]. In particular, the recognition and pose estimation of texture-less metallic industrial components is still an active area of research.

978-1-7281-3605-9/19/$31.00

This paper summarizes results conducted within the research project FORobotics on mobile robots for flexible manufacturing. The addressed use case features a mobile robot, consisting of a Grenzebach L1200S AGV with integrated YASKAWA Motoman manipulator, picking parts from carried SLCs during transportation and place these parts into a pre-assembly fixture in a defined manner.

Our paper focuses within the described use case on the suitability of modern open-source object recognition and pose estimation pipelines for texture-less, metallic parts in industrial applications. An approach for a combined and optimized pipeline for pose estimation is presented, compared to existing methods and evaluated by established assessment procedures. The pipeline includes *You only look once* (YOLO) [6] for recognition and localization as well as for pre-segmentation. The pose estimation is subsequently performed through the template matching algorithm LINEMOD [2]. A significant objective of the proposed combined approach is the reduction of false-positive detections in pose estimation to provide reliable grasping and commissioning. Evaluation is performed on two real-world industrial components in single and multiple object settings of varying object and camera poses. An articulated arm robot is thereby used to guarantee a high level of comparability during measurement runs. The retrieved results show both the potential and challenges for future work to integrate the pipeline with a subsequent grasping method for an successful and robust commissioning process.

II. Related Work: Object Recognition

In computer vision, *Object Recognition* is a generic term for different tasks which can be distinguished by the type and amount of information obtained. The task solved in *Object Classification* is to specify whether an image contains objects of a certain class. *Object Localization* further enriches the classification output with positioning information in the image plane, mostly by specifying bounding boxes. For determination of the exact object contour, *Object Segmentation* assigns information to each pixel in the image as described in [7]. For bin picking, the most relevant task is the field of six *Degrees of Freedom (6DoF) Pose Estimation*, which generates information about the translation (*3DoF*) and rotation (*3DoF*) of the detected object not only in the image plane but in the three-dimensional real-world space. There are also approaches

for grasping pre-known and unknown objects without determination of 6DoF pose [8], [9] and [10]. Those present promising results on grasping using either features or deep learning. However, these researches focus mainly on everyday objects or on high, instantaneous grasp success for arbitrary objects. For industrial use cases, in contrast, the defined and highly accurate grasping of specific components for subsequent pose-defined supply or assembly processes is often mandatory.

A. Object Classification, Localization and Segmentation

Object Localization is conventionally performed using feature-based algorithms such as Scale-Invariant Feature Transform (SIFT) [11], Speeded Up Robust Features (SURF) [12] and Histogram of Oriented Gradients (HOG) [13]. Classifiers like Support Vector Machines (SVM) estimate object classes based on these features. A typical example for a pipeline using features is *Find-Object* [14]. These algorithms are robust for daily life objects with plenty of texture on the surface. We have shown the use of SIFT for bin picking of high-textured objects in [15]. However, for texture-less objects often present in manufacturing scenarios, the extraction of such robust features poses challenges [16].

Recently, deep learning (DL) has dominated this field of computer vision. YOLO [6] is an example of a powerful one-stage approach for real-time object classification and localization. It outputs bounding boxes which can then be used as pre-segmentation for succeeding steps. Other pipelines suitable for bin picking, e.g. [17], use YOLO for comparison purposes. Mask R-CNN [18] provides a high-quality instance segmentation based on Faster R-CNN [19]. It predicts an object mask in parallel with the bounding box. SSD [20] and R-FCN [21] are further algorithms for 2D object localization.

Among the introduced DL-based algorithms, single shot detectors like YOLO and SSD provide a significant advantage regarding short processing times, while region based detectors like Mask R-CNN and R-FCN only offer small accuracy benefits at the cost of slower processing [22]. While neither of them is suitable for our bin picking scenario, since they only predict 2D bounding boxes, these DL-based approaches can be used as a basis to classify, localize and segment the objects for succeeding 6DoF pose estimation. Considering further effort on pose estimation, we determine YOLO as the most suited algorithm with its short cycle times and acceptable accuracy.

B. 6DoF Pose Estimation

For pose estimation both RGB and RGB-D (additional depth channel) data can be used as input. With the availability of depth information, algorithms specifically utilizing this source of information can be used for pose estimation, e.g. point-pair feature matching [3] or template matching like LINEMOD [2]. In bin-picking where objects are often stacked closely together, these approaches show difficulties in correctly estimating the pose and produce an increased number of false positives. This has already been discovered by [16] and coincides with our own investigations.

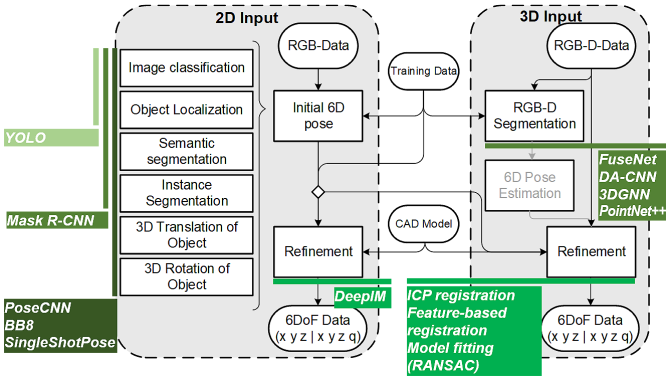


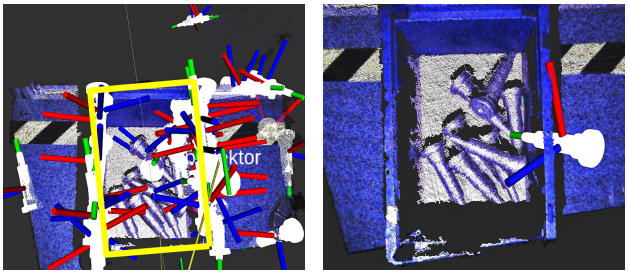
Fig. 1. General workflow of DL-based pose estimation. Available input data, methods and popular algorithms utilizing these are correspondingly annotated.

Even though many of these open-source algorithms for object recognition yield challenges if directly applied, they can be used as basis and in combination to achieve superior results. In Fig. 1 we provide an overview showing possible steps to retrieve 6DoF estimates either from 2D or 3D data, and assign available state-of-the-art algorithms to offer possible solutions for the corresponding steps. The basis for pose estimation with RGB data is the generation of an initial 6DoF pose estimate. Algorithms like PoseCNN [23] or SingleShotPose [5] can create initial poses using RGB data. Subsequently a refinement step is performed, which can e.g. be realized with the DeepIM algorithm [24]. With RGB-D data available, the refinement can be performed using 3D-data-based algorithms like Iterative Closest Point (ICP) registration [25]. RGB-D data can further be used to train neural networks for semantic segmentation like FuseNet [26], 3DGNN [27] or PointNet++ [28].

An interesting work in the area of bin picking, a pose estimation approach to detect polymer components [29] is based on the algorithm point-pair feature matching [3]. A voxel grid filter is used to reduce the size of point cloud data. The pick-success rate is at 89.7%. As the point cloud quality of our metallic components is not sufficient, we cannot use a voxel grid filter to reduce point cloud data. Thus, there is need of alternative segmentation strategies, which can be performed by DL-based algorithms. We have analyzed over a dozen state-of-the-art classical and DL-based algorithms for 6DoF pose estimation of texture-less and metallic components. According to our findings, only very few of them like SingleShotPose by Tekin *et al.* [5] and LINEMOD by Hinterstoisser *et al.* [2] seem to be partially suitable for estimating poses of industrial metallic components in cluttered environments. Based on the retrieved results, we have developed a combined pipeline for 6DoF pose estimation of texture-less metallic components, which is described and evaluated in the following chapters.

III. Pipeline for Combined Object Localization and 6DoF Pose Estimation

One approach showing promising results in our identical object use cases is the LINEMOD template matching algorithm [2]. The LINEMOD template combines image gradients and surface normals as image cues. It is applicable to objects



(a) (b)

Fig. 2. Pose estimation results using LINEMOD [2] for simple contour objects (item 3 shifting rod, cf. Fig. 4). (a) Retrieved pose estimates (white models), with the SLC marked in yellow as target detection area. (b) Pose estimate with highest confidence shown in white.

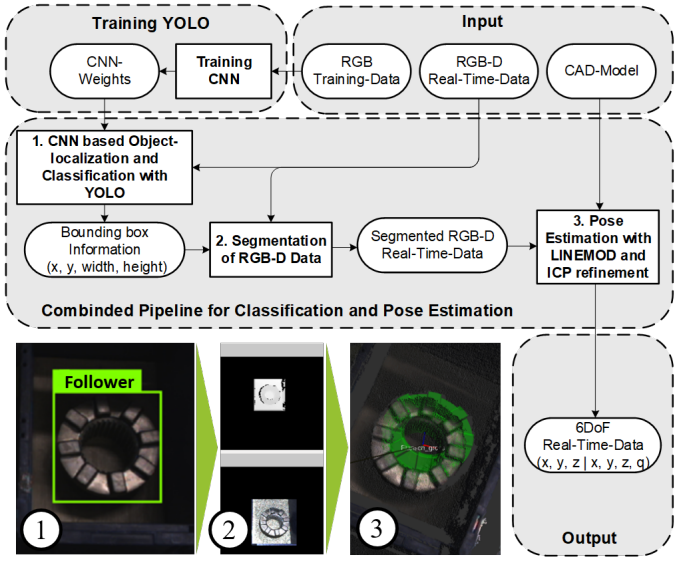


Fig. 3. Proposed pipeline for pose estimation of texture-less metallic industrial parts based on RGB-D input data.

with few or no texture but leads to an unacceptable high number of false positives in our cluttered scene as shown in Fig. 2 (a) when trying to detect objects with simple contours (also mentioned in [16]). The naïve approach of filtering these misdetections based on their confidence scores does not solve the problem since these errors frequently occur with high confidence score assigned, as visualized in Fig. 2 (b). A major task for the reliable use of this algorithm in our use case thus is the significant reduction of false positives, which constitutes one of the main objectives of our developed combined method.

Our approach combines DL-based object localization with 6DoF pose estimation based on template matching. To increase the reliability by reducing the number of false positives, we implement a pre-segmentation of the RGB-D data using the previously introduced CNN YOLO [6]. The expected increase in the poses' precision rate is supposed to enable a stable and accurate pose estimation in the extracted regions of interest using LINEMOD. This is essential for manipulation processes during subsequent picking tasks. Fig. 3 visualizes the workflow of our multi-step approach.

A. Localization of Objects

As input, our pipeline uses an RGB data stream gathered from the camera Roboception *rc_visard 65* to classify objects and generate bounding box information (cf. Fig. 3, step 1). For classification and localization, we use a modified version of YOLO with the last layers adjusted to our investigated two-class problem. For training purposes, we built upon the weights pre-trained on the COCO data set by Lin et al. [30] and extended the training of the last layers to incorporate 250 labeled images of our handling parts with different poses and backgrounds to create a task specific CNN weights file, using batch normalization for regularization. We use 100 images for the follower (item 1 in Fig. 4) and 150 images for the shifting rod (item 3 in Fig. 4). Using such supervised learning methods, a manual annotation process is required for every training image. As described in [31], data augmentation can artificially extend existing data used to train the neural network. In industrial applications this is of great importance since it helps to decrease the effort for generating training data. The deployed YOLO algorithm, adapted to the Robot Operating System (ROS) [32], already integrates data augmentation techniques. The exposure, saturation and angle of input training images are varied automatically during the training process of YOLO to artificially extend the existing training data set.

B. Segmentation of Real-time RGB-D Data

The bounding boxes generated by YOLO define the region of interest for our pose estimates, and provide the basis to segment the RGB-D data stream into fore- and background (Fig. 3, step 2). Only data within the boxes is regarded as important for further processing while the rest, having a high probability to stem from clutter or background, is filtered out by setting the corresponding values of the depth data to zero.

C. Pose Estimation within Segmented Data Streams

In the final processing step of this pose estimation approach, the filtered RGB-D data stream gets published within ROS to the LINEMOD version from Willow Garage [33]. In advance, the corresponding CAD models of the items of interest were provided to LINEMOD. The algorithm has then calculated object templates from multiple points of view. These templates are stored in a database for processing. Through reducing the RGB-D data stream during the previous segmentation step, LINEMOD has less data to process. It only performs the template-matching algorithm to the pre-segmented regions to create initial pose estimates. Finally, to further increase the accuracy of the pose estimate, our combined approach uses ICP for pose refinement (cf. Fig. 3, step 3).

IV. Setup and Procedure of Experiments

In this chapter, our demonstrator system setup, the investigated components, design of experiments as well as the evaluation metrics are described.

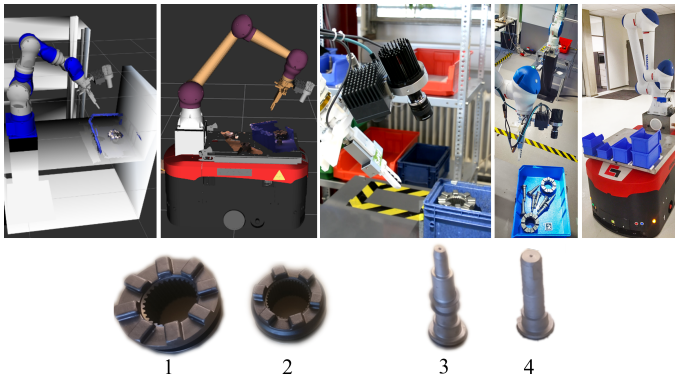


Fig. 4. Demonstrator for use case replication and investigation; Texture-less industrial components. (1) and (2): follower, (3) and (4): shifting rod

A. Demonstrator System Setup

The texture-less components under investigation are metallic parts of a rear axle differential lock. In preparation for a subsequent assembly step, the parts have to be pre-commissioned. Within the scope of FORobotics, a robot-based automation approach is investigated. Therefore, our bin picking application shall separate unsorted and randomly oriented parts provided in SLCs as shown in Fig. 4. The handling parts are two different components, each represented by two variants for examination. (1) and (2) are variants of so-called followers, while (3) and (4) represent different variants of shifting rods.

For investigations, we use a YASKAWA SIA10F and a YASKAWA HC10 articulated arm robot, each equipped with a two finger parallel gripper. Visual perception is performed on each robot by a Roboception rc_visard 65 stereo camera, enhanced by a random-dot projector for improved perception of scenes with little or no natural texture. Both, stereo camera and projector are mounted close to the manipulator's tool center, its mounting design allows easy post adjustments. ROS Kinetic and Ubuntu 16.04 LTS is used on a personal computer (PC), equipped with an Intel Xeon E-2136 (6 Cores / 3.3 GHz), 32 GB RAM and a Nvidia Quadro P2000 GPU with 5 GB RAM with CUDA Compute Capability 6.1 running CUDA Toolkit version 10.0. For deep learning training tasks, we use another more performant PC equipped with an Intel Xeon W-2133 (6 Cores / 3.9 GHz), 32 GB RAM and Nvidia GeForce RTX 2080 GPU with 8 GB RAM with CUDA Compute Capability 7.5 running CUDA Toolkit version 10.1.

To achieve a modular architecture, we have chosen a skill based approach using ROS actions (e.g. for motion control, object recognition, etc.) described in [34]. The triggering and composition of skills as orchestration to a bin picking task is performed by a SMACH [35] state machine. The resulting 6DoF pose from our pipeline gets transmitted to a tf coordinate transformation as described in [36]. The MoveIt! motion planning framework is used for trajectory planning.

B. Procedure of Experiments

To evaluate the accuracy of the poses estimated by the tested pipelines, we first place AprilTags [37] on the objects and use the thereby retrieved accurate poses together with the known object geometry as ground truth. We create two different real-world data sets with varying object and camera poses. *Dataset-A* is used to achieve a first insight into the performance of the estimation algorithms and constitutes of measurements of one single follower (item 1, Fig. 4) in fifteen different poses placed in an SLC. Considering the bin picking task, the robot moves the camera above the SLC to ten different specified scan positions, perceiving the object from different angles and distances. This results in a total $15 \times 10 = 150$ measurements for *Dataset-A*. To provide both statistical insight and robustness regarding influences caused by the environment like changes in lighting, we further create an additional *Dataset-B*, containing measurements of both follower and shifting rod (items 1 and 3, Fig. 4) in single and multiple object scenarios. Here, we specify nine different poses for each object combined with 4 different camera poses for perception, and the robot moves alternately to each scan position 25 times. This results in a number of $9 \times 4 \times 25 = 900$ poses for each object for single and multiple object scenario, summing up to in total 3600 measurements for *Dataset-B*.

C. Evaluation Metrics

Throughout the literature, several different metrics for evaluating 6DoF pose estimates have been proposed. One of the most simple ways to obtain measures for comparison is by determining the translational and rotational error. However, while the translational error constitutes a possible metric for evaluating estimated poses in our scenario, quantifying the rotational error is only partly suited due to the rotational symmetry of our industrial items.

A more representative metric used, is the *Average Distance* (AD) [2], describing the distance between ground truth and estimated pose using the CAD model. For objects with *indistinguishable* views, there exists a formulation that is abbreviated ADI (comparable to ADD-S [23]) and defined as

$$e_{\text{ADI}}\left(\check{P}, \bar{P}; M\right) = \underset{x_1 \in M}{\text{avg}} \min_{x_2 \in M} \left\| \bar{P}x_1 - \check{P}x_2 \right\|_2 . \tag{1}$$

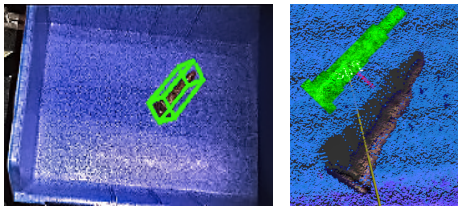
The estimated pose is here denoted by $\check{P}$, the ground truth pose by $\bar{P}$ and the object model by M. A pose estimate is considered correct if the error $e_{\text{ADI}} \leq k \cdot d$, with k being a constant to be chosen and d the largest distance between any pair of model vertices, i.e., the diameter or length.

Another metric we use is the *Visible Surface Discrepancy* (VSD) proposed by [38], specifically targeting object symmetries and occlusions and defined as

$$e_{\text{VSD}}\left(\check{P}, \bar{P}; M, I, \delta, \tau\right) = \underset{p \in \check{V} \cup \bar{V}}{\text{avg}} c\left(p, \check{D}, \bar{D}, \tau\right) . \tag{2}$$

It renders the object's model M into two 2D masks of the visible surface $\check{V}$ and $\bar{V}$ with estimated pose $\check{P}$ and ground truth pose $\bar{P}$. The tolerance δ is defined to determine the visibility. With distance images $\check{D}$ and $\bar{D}$ rendered from model M at the estimated and ground truth pose respectively, the matching cost c is calculated as

$$c\left(p, \check{D}, \bar{D}, \tau\right) = \begin{cases} d/\tau & \text{if } p \in \check{V} \cap \bar{V} \ \wedge \ d < \tau \\ 1 & \text{otherwise,} \end{cases} \tag{3}$$



(a) (b)

Fig. 5. Pose estimate retrieved with SingleShotPose ([5]). (a) shows the 2D projection of the 3D bounding box while (b) shows the estimated 6D pose visualized by the green CAD-model from a different perspective.

with d as distance between the surfaces of estimated pose and ground truth at pixel p. Thereby, τ denotes the misalignment tolerance limiting the allowed range of d. With a threshold θ, the correctness of the estimated pose is determined.

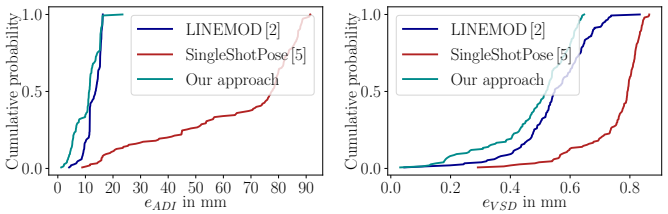
V. Evaluation and Discussion

To provide a basis for comparison of our method with the state of the art, we first evaluate SingleShotPose [5], LINEMOD [2] and our approach on *Dataset-A* (cf. IV-B). Then both the original LINEMOD approach and our combined approach are evaluated for single and multiple same-kind object scenarios (*Dataset-B*, cf. IV-B). The experimental setup corresponds to the description in Chapter IV.

A. Single Object Pose Estimation – Dataset-A

SingleShotPose extends YOLO to predict the 2D projections of the corners of the object 3D bounding box. A 6DoF pose estimate is then retrieved using the Perspective-n-Point (PnP) algorithm [39]. Using AprilTag we annotate about 1000 pictures with ground truth poses to create a dataset for training the approach on our parts and further provide the parameters of our calibrated camera to the algorithm. After training, we are able to obtain similar results as in [5] with 92.42% accuracy using a 5 pixel 2D projection of the 3D bounding box.

Although these 2D bounding box projections seem to be promising (cf. Fig. 5 (a)), qualitative analysis already shows that the results of the retrieved 6DoF pose estimates show significant errors. The pose calculated by the PnP algorithm is visualized from a different perspective in Fig. 5 (b) with the object's CAD model. The error between the estimated and the real pose is clearly visible, being specifically critical in z-direction (depth). These initial qualitative results are backed by the quantitative results retrieved from *Dataset-A* shown in Fig. 6, with the majority of errors in terms of both ADI and VSD metric located in the upper quadrant. It is to be noted that this algorithm solely relies on RGB input data and thus is a valuable approach if no depth information is available. However, using this algorithm with our limited number of available training samples and solely relying on the RGB input, even minimal errors in the 2D projection can lead to errors in the 3D pose that cannot be tolerated in the context of our bin picking use case. In contrast, both approaches utilizing depth information show superior results in terms of both metrics and are thus analyzed in more detail in the following section.



(a) (b)

Fig. 6. Pose estimation errors for follower (item 1 in Fig. 4) in *Dataset-A*, visualized as the empirical cumulative distribution function in terms of (a) the ADI and (b) VSD metric.

B. Single Object Pose Estimation – Dataset-B

Due to the availability of depth information, we decide to focus on the detailed evaluation of our combined approach based on LINEMOD in comparison with the original LINEMOD [2] on *Dataset-B*. We quantify the errors of the retrieved pose estimates based on the metrics introduced in Chapter IV.

6D accuracy in terms of ADI metric. The errors quantified by the ADI metric are displayed in the top row of Fig. 7. For applying the LINEMOD algorithm to the data of the shifting rod (cf. (a)), the majority of errors is located above $15\,\text{mm}$, with a high number of errors concentrated at around $30\,\text{mm}$, constituting a pose estimation offset of about one quarter of the size of the object diameter ($120\,\text{mm}$). The pose estimates retrieved for the follower(cf. (b)) are slightly better, but still with the majority of errors greater than $10\,\text{mm}$. In contrast, by using our method combining LINEMOD with specific pre- and post-processing techniques, we are able to reduce these pose estimation errors, resulting in errors smaller than $8\,\text{mm}$ and $6.5\,\text{mm}$ for the majority of estimates for shifting rod and follower, respectively. We are however not able to completely stabilize the estimation process in this challenging scenario of texture-less objects, leaving some outliers that still lead to erroneous pose estimates.

6D accuracy in terms of VSD metric. Unlike the ADI metric evaluating the average 3D distance of the whole object, the VSD metric calculates the error based on the visible part of the model's surface. The errors for both the direct application of LINEMOD [2] and our combined approach are shown in the bottom row of Fig. 7. Similar to the previous case, although we are not able to achieve outlier-free estimates, the robustness and accuracy is significantly improved.

Recall Rate. As stated in Chapter IV, both error metrics provide a measure to determine if a pose can be treated as correct, based on specific threshold parameters. We set these parameters to $k = 0.1$ for the ADI metric and to $\tau = 30\,\text{mm}$, $\delta = 30\,\text{mm}$ and $\theta = 0.5$ for VSD. The recall rates, defined as the percentage of estimated poses recognized as correct, are shown in Table I.

In addition to the reduction of pose estimation errors, we are able to increase the availability of pose estimates for both metrics. This constitutes an essential property for short bin picking cycle times. For single shifting rods, we were able to more than double the number of estimates. For multiple objects present in the scene, the original LINEMOD method

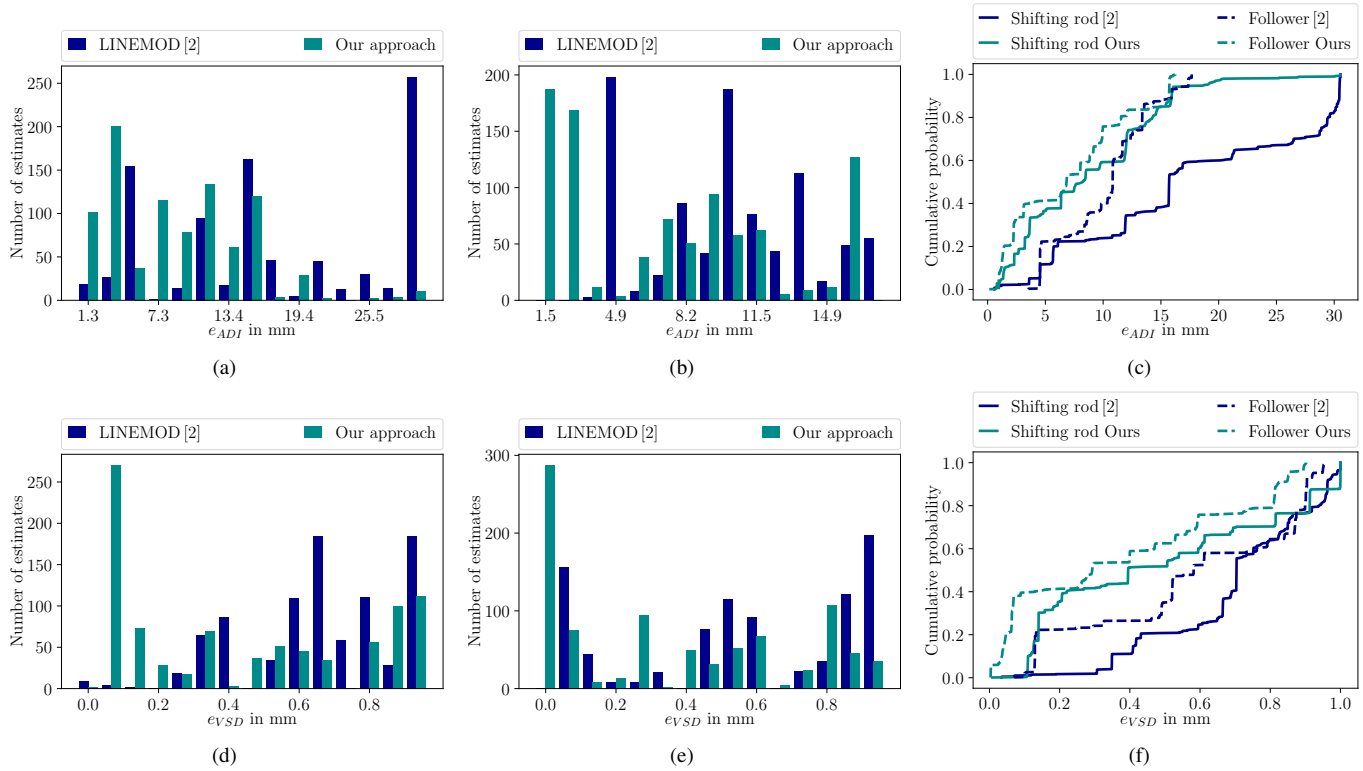


Fig. 7. Resulting pose estimation errors acquired with LINEMOD [2] and our approach for shifting rod and follower (item 3 and 1 in Fig. 4, respectively) on *Dataset-B*. Results visualized as histograms in terms of the ADI metric for (a) the shifting rod and (b) the follower, as well as in terms of the VSD metric for (d) the shifting rod and (e) the follower. (c) and (f) show the corresponding empirical cumulative distribution functions.

TABLE I
RECALL RATES ORIGINAL LINEMOD [2] AND OUR PROPOSED PIPELINE

Evaluated Item (Method, #Objects)	$\text{recall}_{\text{ADI}}$	$\text{recall}_{\text{VSD}}$
Shifting Rod (LINEMOD [2], Single)	34.4%	20.8%
Shifting Rod (Our approach, Single)	**72.1%**	**51.7%**
Shifting Rod (LINEMOD [2], Multiple)	0.0%	0.0%
Shifting Rod (Our approach, Multiple)	**66.6%**	**48.6%**
Follower (LINEMOD [2], Single)	69.1%	35.0%
Follower (Our approach, Single)	**81.0%**	**62.4%**
Follower (LINEMOD [2], Multiple)	0.0%	0.0%
Follower (Our approach, Multiple)	**72.2%**	**57.9%**

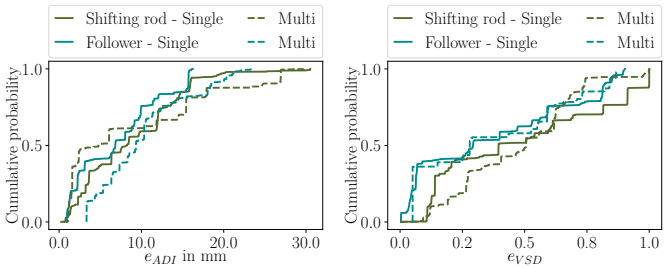


Fig. 8. Pose estimation errors retrieved by our approach for scenarios with single and multiple items in one SLC, visualized as the empirical cumulative distribution function in terms of (a) the ADI and (b) the VSD metric on *Dataset-B*.

was not able to yield any pose estimate classified as correct, while our method is significantly more robust to clutter during processing, enabling it to provide at least one reliable estimate for more than half of all conducted experiments.

C. Pose Estimation in Multiple-Object Scenarios

Since in bin picking applications SLCs often contain multiple industrial components of the same kind closely stacked together or randomly assorted, we further evaluate our approach on such a scenario with the objective of determining an accurate pose estimate of at least one object (that could be grasped). The quantitative results for an SLC containing eight shifting rods and one containing 4 followers are presented in Fig. 8. Compared to the evaluation performed on a single object also shown in this figure, the performance of our approach only slightly decreases for the multiple objects case, resulting in a wider spread of errors for both the ADI metric (cf. Fig 8 (a)) and the VSD metric (cf. Fig 8 (b)). However, the majority of errors is still located in an acceptable error margin, and combined with the results achieved regarding recall (cf. Table I), this indicates robustness of our approach in both single and multiple object scenarios.

VI. SUMMARY, CONCLUSION AND OUTLOOK

In this work, different approaches for 6DoF pose estimation in industrial applications have been described. We have shown the system setup, the design of experiments and metrics to evaluate our pose estimation pipeline with focus on textureless metallic industrial components. This work provides an

example of how existing bin picking applications can be improved by deep learning-based algorithms. The results enable to derive robust pose estimation strategies for a more reliable and time-efficient bin picking. In particular, the pose estimates retrieved with our combined pipeline show a noticeable reduction of both errors and false positive detections compared to available approaches. However, several outliers are still identifiable which have to be investigated in more detail in following work. As the recall rates show, we are able to enhance the availability of correct pose estimates, leading to shorter bin picking cycle times.

Regarding our learning approach, we performed a manual annotation process for labeling the training samples. This effort could be reduced by domain adaptation to generate synthetic data. Using a data set synthesizer (e.g. [40]), only a CAD model is required for gathering annotation data like bounding boxes and object poses. Furthermore, DL-based 6DoF pose estimation algorithms like [5], [41] and [42] are expected to improve their results allowing further evaluations in industrial bin picking scenarios.

ACKNOWLEDGMENT

This research was conducted within the project *FORobotics* (AZ-1225-16) on mobile robots in shop-floor environments, funded by the *Bavarian Research Foundation (BFS), Germany*.

REFERENCES

[1] M. M. Mabkhot *et al.*, "Requirements of the smart factory system: A survey and perspective," *Machines*, vol. 6, no. 2, 2018.

[2] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Computer Vision – ACCV 2012*, pp. 548–562, Springer, 2013.

[3] B. Drost *et al.*, "Model globally, match locally: Efficient and robust 3d object recognition," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 998–1005, June 2010.

[4] M. Hiller *et al.*, "World modeling for mobile platforms using a contextual object-based representation of the environment," in *IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 187–191, July 2018.

[5] B. Tekin *et al.*, "Real-time seamless single shot 6d object pose prediction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 292–301, June 2018.

[6] J. Redmon *et al.*, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[7] A. Garcia-Garcia *et al.*, "A review on deep learning techniques applied to semantic segmentation," *CoRR*, vol. abs/1704.06857, 2017.

[8] D. Fischinger and M. Vincze, "Empty the basket - a shape based learning approach for grasping piles of unknown objects," in *International Conference on Intelligent Robots and Systems*, pp. 2051–2057, 2012.

[9] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Rob. Res.*, vol. 34, pp. 705–724, Apr. 2015.

[10] A. Zeng *et al.*, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018.

[11] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157 vol.2, Sep. 1999.

[12] H. Bay *et al.*, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006*, pp. 404–417, Springer, 2006.

[13] N. Dalal *et al.*, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, June 2005.

[14] Labbé, M., "Find-Object." http://introlab.github.io/find-object, 2011.

[15] A. Blank *et al.*, "Bag Bin-Picking Based on an Adjustable, Sensor-Integrated Suction Gripper," in *Proceedings of the 3rd Congress Assembly, Handling and Industrial Robots*, Wiesbaden: Springer, 2018.

[16] R. He *et al.*, "A 3d object detection and pose estimation pipeline using rgb-d images," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1527–1532, Dec 2017.

[17] M. Brucker *et al.*, "6dof pose estimation for industrial manipulation based on synthetic data," in *International Symposium on Experimental Robotics (ISER)*, IFRR, 2018. accepted for publication.

[18] K. He *et al.*, "Mask r-cnn," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, Oct 2017.

[19] S. Ren *et al.*, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, June 2017.

[20] W. Liu *et al.*, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.

[21] J. Dai *et al.*, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems 29*, pp. 379–387, Curran Associates, Inc., 2016.

[22] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3296–3297, July 2017.

[23] Y. Xiang *et al.*, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *CoRR*, vol. abs/1711.00199, 2017.

[24] Y. Li *et al.*, "Deepim: Deep iterative matching for 6d pose estimation," in *Computer Vision - ECCV 2018, Munich*, pp. 695–711, 2018.

[25] S. Rusinkiewicz *et al.*, "Efficient variants of the icp algorithm," in *Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, 2001.

[26] C. Hazirbas *et al.*, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Asian Conference on Computer Vision (ACCV)*, November 2016.

[27] X. Qi *et al.*, "3d graph neural networks for rgbd semantic segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 5209–5218, Oct 2017.

[28] C. R. Qi *et al.*, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems 30*, pp. 5099–5108, Curran Associates, Inc., 2017.

[29] C. Wu *et al.*, "Cad-based pose estimation for random bin-picking of multiple objects using a rgb-d camera," in *15th International Conference on Control, Automation and Systems*, pp. 1645–1649, Oct 2015.

[30] T. Lin *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV*, (Cham), pp. 740–755, Springer International, 2014.

[31] H. Inoue, "Data augmentation by pairing samples for images classification," *CoRR*, vol. abs/1801.02929, 2018.

[32] M. Bjelonic, "YOLO ROS: Real-time object detection for ROS." https://github.com/leggedrobotics/darknet_ros, 2016–2018.

[33] R. c. Willow Garage, "ORK: Object Recognition Kitchen." https://github.com/wg-perception/object_recognition_core, 2017.

[34] L. Heuss, A. Blank, *et al.*, "Modular robot software framework for the intelligent and flexible composition of its skills." Manuscript submitted for publication in Advances in Production Management Systems, 2019.

[35] J. Bohren and S. Cousins, "The smach high-level executive [ros news]," *IEEE Robotics Automation Magazine*, vol. 17, pp. 18–20, Dec 2010.

[36] T. Foote, "tf: The transform library," in *2013 IEEE Conference on Technologies for Practical Robot Applications*, pp. 1–6, April 2013.

[37] J. Wang *et al.*, "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4193–4198, Oct 2016.

[38] T. Hodan *et al.*, "BOP: benchmark for 6d object pose estimation," in *ECCV (10)*, vol. 11214 of *LNCS*, pp. 19–35, Springer, 2018.

[39] V. Lepetit *et al.*, "Epnp: An accurate o(n) solution to the pnp problem.," *International Journal of Computer Vision*, vol. 81, no. 2, 2009.

[40] T. To *et al.*, "NDDS: NVIDIA deep learning dataset synthesizer," 2018. https://github.com/NVIDIA/Dataset_Synthesizer.

[41] C. Qi *et al.*, "Frustum pointnets for 3d object detection from rgb-d data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 918–927, June 2018.

[42] C. Wang *et al.*, "Densefusion: 6d object pose estimation by iterative dense fusion," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.