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

Andreas Blank[1], Markus Hiller[2], Alexander Leser[1], Siyi Zhang[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)
Egerlandstr. 7-9, Erlangen, D-91058 Germany
email: {andreas.blank,joerg.franke}@faps.fau.de

[2]Institute for Information Technologies (LIKE)
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Am Wolfsmantel 33, Erlangen, D-91058 Germany
email: {markus.hiller,joern.thielecke}@fau.de

***Abstract*—Over the next few years, autonomous robots and functionalities are gaining importance for the shop floor. The environment 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 6DoF pose estimates for bin picking applications, specifically targeting the suitability for challenging scenarios with texture-less, metallic parts in industrial environments. The pipeline is based on open source algorithms. Our approach combines Convolutional Neural Network (CNN)-based and feature-matching-based methods to create an effective 6DoF pose estimate. We evaluate our approach on several industrial components using a seven-axis articulated arm robot to guarantee a high level of comparability during the different measurement runs. We further quantify the results using well known error metrics for pose estimation. Also we provide statistical insight into achieved outcomes to assess the robustness and compare the results to established approaches.**

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. The wide range of available and proven industrial robots means that a suitable solution can be provided for specific applications.

In production, supplied or self-manufactured components are often provided randomly stacked in small load carriers (SLC). Therefore, bin picking has a high relevance for feeding parts to subsequent processes, for packaging and for assembly. 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. Established automation solutions such as vibratory screw feeders can only partially fulfill these tasks and are often not flexible enough. Evidence for the relevance of bin picking can also be seen in recently launched research projects in this field [2].

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

This paper summarizes research results of the project FORobotics on mobile robots for flexible manufacturing regarding bin picking of machine elements from SLCs. The paper focuses in particular on the suitability of modern open-source object recognition and pose estimation pipelines for texture-less metallic parts. An approach regarding a combined and optimized pipeline is presented, compared to existing approaches and evaluated by established assessment procedures.

II. Related Work: Object Recognition

In the area of computer vision, *Object Recognition* is a generic topic 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 of the object 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 [6]. For bin picking, the most relevant task is the further six *Degrees of Freedom (DoF) 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.

A. Object Classification, Localization and Segmentation

Object Localization is conventionally performed by using features such as Scale-Invariant Feature Transform (SIFT) [7], Speeded Up Robust Features (SURF) [8] and Histogram of Oriented Gradients (HOG) [9]. Classifiers like Support Vector Machines (SVM) estimate object classes based on these features. A typical pipeline using such features is *Find-Object* introduced in [10]. These features are robust for common objects in our daily life with plenty of texture on the surface. We have shown the successful use of SIFT in bin picking scenarios of high-textured handling objects in [11]. However, for texture-less objects, the extraction of such robust features poses challenges, leading to failures of these approaches [12].

Recently, deep learning (DL) has dominated this field of computer vision. *You only look once* (YOLO) by Redmon et al. [13] is an example of a powerful one-stage approach that can be used efficiently for real-time object classification and localization. It outputs bounding boxes which can then be used

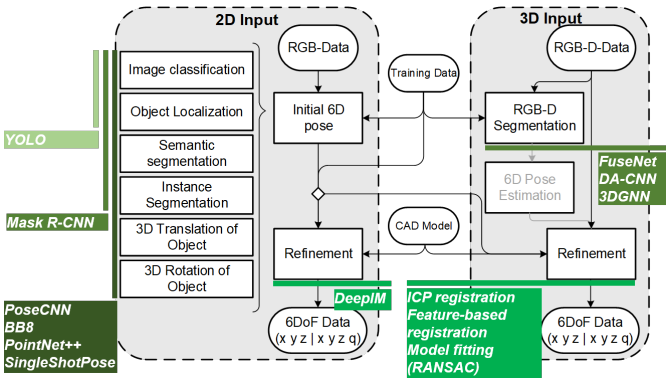


Fig. 1. Workflow of DL-based pose estimation

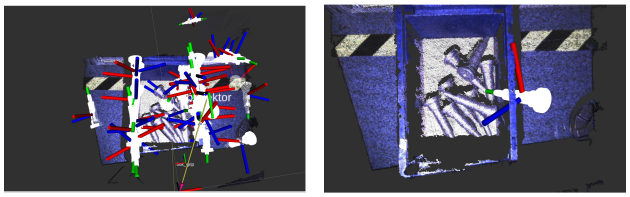
as pre-segmentation for succeeding steps. Other pipelines also suitable for bin picking like the approach by [14] use YOLO for comparison purposes. Mask R-CNN by [15] provides a high-quality instance segmentation based on Faster R-CNN featured in [16]. It predicts an object mask in parallel with the bounding box. SSD in [17] and R-FCN in [18] are further popular algorithms for 2D object localization.

Among the introduced DL-based algorithms, single shot detectors like YOLO and SSD have an impressive advantage in processing times, while region based detectors like Mask R-CNN and R-FCN only offer small accuracy advantages at slower processing times, like shown in [19]. While neither of them is suitable for our bin picking, since it predicts 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, YOLO is the best algorithm with its short cycle times and acceptable accuracy.

B. 6DoF Pose Estimation

For pose estimation, both RGB and RGB-D data can be used, while the latter brings additional depth information to the input image. With the availability of depth information, specific algorithms utilizing this source of information, like point-pair feature matching by Drost et al. [4] or template matching like LINEMOD by Hinterstoisser et al. [3], can be used for pose estimation. In bin picking applications 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 [12] and coincides with our own investigations.

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. We provide an overview showing possible steps to retrieve 6DoF pose estimates either from 2D or 3D input data in Fig. 1, and assign available state-of-the-art algorithms which we identified 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 [20], PointNet++ [21] or SingleShotPose [5] can



(a) (b)

Fig. 2. Pose results using LINEMOD for simple contour object (item 3 in Fig. 4). (a) False positives in detection area. (b) False positive result with highest confidence.

create such initial poses using RGB Data. Subsequently a refinement step is performed, which can be realized with the DeepIM algorithm [22]. With RGB-D data available, the refinement can be performed using 3D-data-based algorithms like Iterative Closest Point (ICP) registration [23]. RGB-D data can further be used to directly train neural networks for semantic segmentation like FuseNet [24] or 3DGNN [25].

An interesting work in the area of bin picking has been published by Wu et al. [26], whose pose estimation approach to detect polymer components is based on the algorithm point-pair feature matching [4]. 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.

During our research we have tested different algorithms for pose estimation of texture-less and metallic components. We analyzed over a dozen state-of-the-art classical and DL-based algorithms for 6DoF pose estimation in such scenarios. Only very few of them, like SingleShotPose [5] and the approach of Konishi et al. [27], seem to be suitable for estimating poses of industrial metallic components according to our findings. Based on the retrieved results, we have developed a combined pipeline for 6DoF pose estimation of texture-less metallic components, which is described in the following chapter.

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

One algorithm showing promising results in our use case is the mentioned LINEMOD template matching algorithm proposed by Hinterstoisser et al. [3]. The LINEMOD template combines image gradients and surface normals as image cues. It is applicable to objects 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 [12]). 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 elimination of false-positives, which is part of our investigated method.

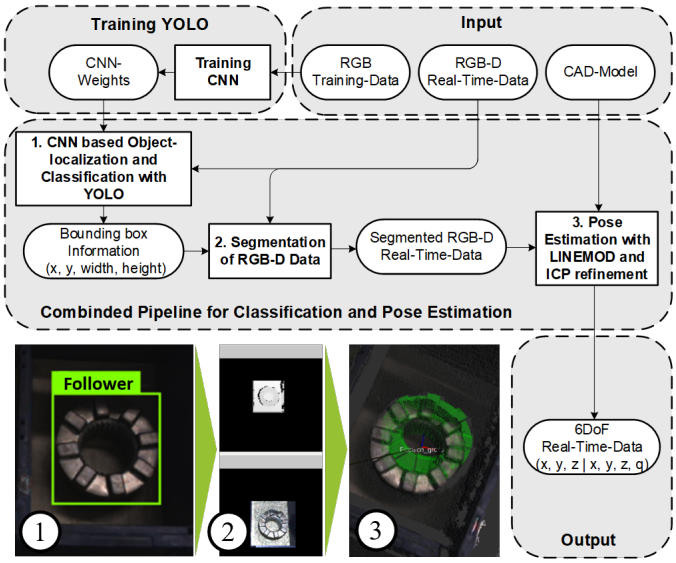


Fig. 3. Pipeline for Classification and 6DoF pose estimation for texture-less metallic industrial parts

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-based YOLO. The expected increase in the poses' precision rate is supposed to enable a stable and accurate pose estimation using LINEMOD. This is necessary for manipulation processes during subsequent picking tasks. Fig. 3 visualizes the workflow of our approach.

A. Localization of Objects

As input, our pipeline uses an RGB datastream gathered from the 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. We built upon the weights pre-trained on the COCO data set by Lin et al. [28] and extended the training of the last layers to incorporate 250 labeled images of our handling parts with different possible backgrounds to create a task specific CNN-weights file. 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 [29], data augmentation can artificially extend existing data used to train the neural network. In industrial applications this is of great importance. Data augmentation helps decreasing the effort for generating training data. The deployed YOLO algorithm, adapted to the Robot Operating System (ROS) [30], 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.

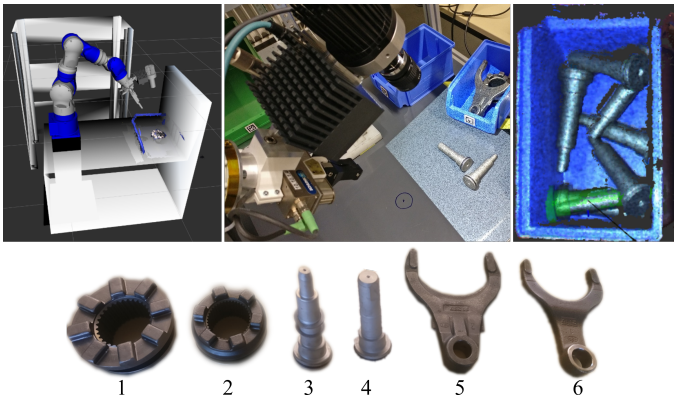


Fig. 4. Demonstrator for use case replication and investigation; Texture-less industrial use case components. (1) and (2): Follower, (3) and (4): Shifting Rod, (5) and (6): Shifting Fork

B. Segmentation of Real-time RGB-D Data

Subsequently the generated bounding boxes from YOLO serves to segment the RGB-D datastream (Fig. 3, step 2). Only data within boxes remains for further processing. Data outside, including the cluttered environment, has no importance for further processing. The filtering is performed by setting irrelevant depth data values to zero.

C. Pose Estimation within Segmented Datastreams

In the final processing step of this combined 6DoF pose estimation approach, the filtered RGB-D datastream gets published within ROS to the LINEMOD version used from Willow Garage [31]. In advance, the corresponding CAD models of the items of interest were provided to the LINEMOD algorithm. The algorithm has then calculated thereon-based object templates from multiple points of view. These templates are stored in a database for processing. Through reducing the RGB-D datastream during the previous segmentation step, LINEMOD has less data to calculate. 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 the following chapter, our demonstrator system setup, the investigated components, the design of experiments as well as the evaluation metrics are described.

A. Demonstrator System Setup

The texture-less industrial components are metallic parts of a truck's rear axle differential lock, partly nitrided. In preparation for a subsequent assembly step at the assembly-line, the parts have to be pre-commissioned into a fixture. Currently, this process step is carried out manually. Within the scope of the FORobotics use case, a robot-based automation approach is investigated. Therefore our bin picking application shall separate unsorted and randomly oriented parts, provided

in SLCs shown in Fig. 4. The handling parts in our investigated use case are three different components of the differential lock, each represented by two variants for examination. (1) and (2) are variants of so-called Followers, (3) and (4) present Shifting Rods and (5) as well as (6) different variants of Shifting Forks.

For investigations, we use a YASKAWA SIA10F seven-axis articulated arm robot with a SG150 two finger parallel gripper from PTM-Mechatronics (stroke of 100 mm). Visual perception is performed by a Roboception rc_visard 65 stereo camera, enhanced by a random-dot projector for improved percepting scenes with less 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 is used on a personal computer (PC), equipped with an Intel Core i7-4770 (4 Cores / 3.4 GHz), 16 GB RAM and a Nvidia GeForce GTX 1050Ti GPU with 4 GB RAM with CUDA Compute Capability 6.1 running CUDA Toolkit version 9.2. 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 [32]. The triggering and composition of skills as orchestration to a bin picking task is performed by a SMACH state machine. The resulting 6DoF pose from our pipeline gets transmitted to a tf coordinate transformation as described in [33]. Results are stored in a database for subsequent grasping point determination. The MoveIt! motion planning framework with integrated Open Motion Planning Library (OMPL) is used for trajectory planning. ROS rviz serves for process and environment visualization.

B. Procedure of Experiments

To evaluate the accuracy of the poses estimated by the tested pipelines, we first placed AprilTags [34] on the objects' centers and use the thereby retrieved accurate poses as ground truth. For each object we define nine different poses for evaluation in the real scene. Considering the bin picking task, the robot moves above the SLC with four separate specified scan positions. For getting different hypotheses, the robot moves alternately to each scan position 25 times. This results to a number of $9 \times 4 \times 25 = 900$ poses evaluated per object.

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 many industrial items.

A more representative metric used in this work is the *Average Distance* (AD) [3], describing the distance between ground truth and estimated pose using the object CAD model. For objects with indistinguishable views, there exists a specific formulation that is abbreviated ADI and defined as

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

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 is a constant to be chosen and d is 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 [35], specifically targeting object symmetries and occlusions and defined as

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

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 $\bar{P}$. δ is a defined tolerance 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

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

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.

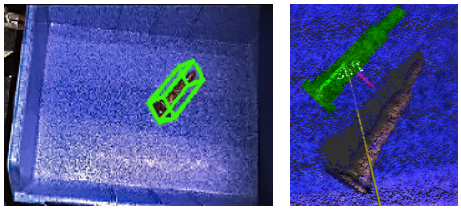
V. Evaluation and Discussion

To provide a basis for comparison of our method with the state-of-the-art, we first evaluate *SingleShotPose* [5] on a single object, namely the shifting rod displayed as item 3 in Fig. 4. Then both the original LINEMOD approach and our combined approach are evaluated for this single object and for multiple objects of the same kind present in the scene. The experimental setup corresponds to the description in Chapter IV.

A. Single Object Pose Estimation with SingleShotPose

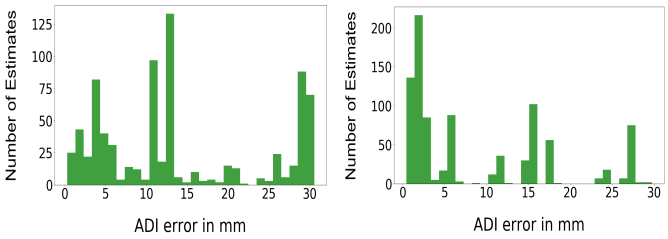
SingleShotPose extends YOLO to predict the 2D projections of the corners of the object 3D bounding box. A 6D pose estimate is then retrieved using the Perspective-n-Point (PnP) algorithm [36]. Using AprilTag we annotate about 1000 pictures with ground truth poses to create a dataset for training the approach on our parts. After training, we are able to obtain similar results as described in [5] with 92.42% accuracy using a 5 pixel 2D projection of the 3D bounding box.

Although these 2D bounding box results seem to be promising (cf. Fig. 5 (a)), qualitative analysis shows that the results of the actual 6DoF pose estimates are not acceptable for our scenario. The pose calculated by the PnP algorithm is visualized 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 the z-direction (depth). Using this algorithm with our limited number of available training



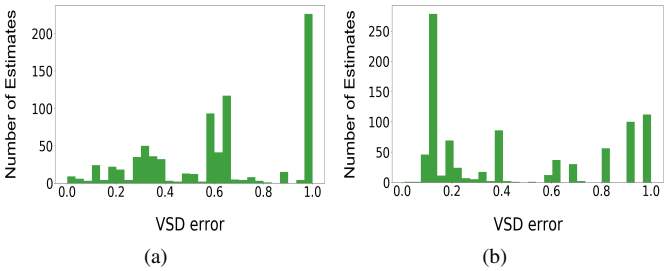
(a) (b)

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.



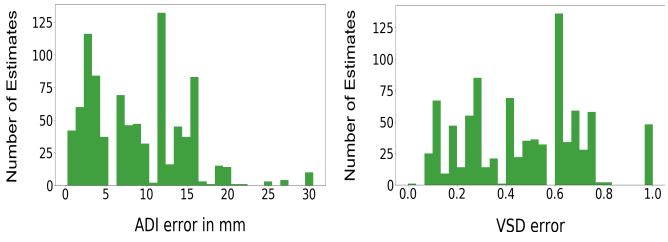
(a) (b)

Fig. 6. Pose estimation error for Shifting Rod in terms of ADI metric acquired with (a) the original LINEMOD algorithm and (b) our own approach



(a) (b)

Fig. 7. Pose estimation error for Shifting Rod in terms of VSD metric acquired with (a) the original LINEMOD algorithm and (b) our own approach



(a) (b)

Fig. 8. 6D Accuracy of Shifting Rod in terms of ADI and VSD metrics of multiple objects

samples and solely relying on RGB input, even minimal errors in the 2D projection can lead to large errors in the 3D pose, disqualifying it for application in our bin picking use case.

B. Single Object Pose Estimation with LINEMOD and our Combined Approach

To show the performance of our method, we evaluate both the original LINEMOD and our combined approach on the same test data set. We quantify the errors of the pose estimation 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 Fig. 6. For applying the LINEMOD algorithm to the data (cf. (a)), the majority of errors is located above 15 mm, with a high number of errors concentrated at around 30 mm, constituting a pose estimation offset of about one quarter of the size of the object diameter (120 mm). In contrast, by using our method combining LINEMOD with specific pre- and post-processing techniques, we are able to reduce the pose estimation errors, resulting in an error smaller than 5 mm for the majority of estimates (cf. (b)). Similar results are obtained for the other items listed in Fig. 4 (results not shown in this paper). 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 and our combined approach are shown in Fig. 7. Similar to the previous case, although we are not able to achieve outlier-free estimates, the robustness and accuracy is significantly improved by reducing the majority of errors.

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 rate, defined as the percentage of estimated poses recognized as correct, are shown in below Table I. After 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 objects, we were able to more than double the number of estimates. For multiple objects present, the original LINEMOD method was not able to yield any pose estimates classified as correct, while our method is able to significantly reduce the clutter during processing, enabling it to provide reliable estimates for more than half of all conducted experiments.

C. Multi Object Pose Estimation with Combined Approach

Since in bin picking applications, SLCs often contain multiple industrial components of the same kind closely stacked

TABLE I
RECALL RATE FOR ORIGINAL LINEMOD AND OUR COMBINED PIPELINE

Evaluated Item (Method, #Objects)	$\text{recall}_{\text{ADI}}$	$\text{recall}_{\text{VSD}}$
Shifting Rod (LINEMOD, Single)	34.4%	20.7%
Shifting Rod (Our approach, Single)	**72.1%**	**51.6%**
Shifting Rod (LINEMOD, Multiple)	0.0%	0.0%
Shifting Rod (Our approach, Multiple)	**66.5%**	**48.5%**

together or randomly assorted, we also evaluate our approach on such a scenario. The quantitative results for a SLC containing eight shifting rods are presented in Fig. 8. Compared to the evaluation performed on a single object shown in Fig. 6 for ADI and in Fig. 7 for VSD, the performance of our approach only slightly decreases, resulting in a wider spread of errors for both the ADI metric (cf. 8 (a)) and the VSD metric (cf. 8 (b)). However, especially regarding the ADI metric, the majority of errors is still located in an acceptable error margin (cf. Table I).

VI. Summary, Conclusion and Outlook

Within this paper, different approaches for 6DoF pose estimation in industrial applications are described. We show the necessary system setup, the design of experiments and evaluation metrics to test our pose estimation pipeline. Thereby the focus of our pipeline is on texture-less, metallic industrial components. The work provides an example of how existing bin picking applications can be improved by deep learning-based algorithms. The results enable to derive fast and robust pose estimation strategies for a more reliable and time-efficient bin picking. The results of our combined pipeline regarding the pose estimation error show a noticeable reduction of errors and thus prove the feasibility of the approach. However, there are also outliers identifiable which have to been investigated in more depth in following works. As the recall rates show, we are able to noticeable enhance the availability of correct pose estimates, leading to shorter bin picking cycle times.

In this work, the evaluation of the shifting rod (item 3 in Fig.4) has been presented. Therefor, we performed a manual annotation for labeling the training samples. This effort could be reduced by a Domain Adaptation to generate synthetic data. Using a data set synthesizer (e.g. [37]), only a CAD model is required gathering annotation data like bounding boxes and object poses. Thereby our approach can be extended and adapted to detect arbitrary objects. Future studies will also focus on the quality of our training on loss evaluation. Furthermore, DL-based 6DoF pose estimation algorithms like by Tekin et al. [5] and Qi et al. [38] are expected to improve their results allowing further evaluations in bin picking scenarios.

Acknowledgment

This research work belongs to the project *FORobotics* (AZ-1225-16) on mobile robots in shopfloor 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] A. Zell, "ibinpick - development of an intelligent bin picking robot system," 2018. University of Tübingen, project horizon 2018-2021.

[3] 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.

[4] 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.

[5] B. Tekin *et al.*, "Real-time seamless single shot 6d object pose prediction," *CoRR*, vol. abs/1711.08848, 2017.

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

[7] 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.

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

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

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

[11] 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.

[12] 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.

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

[14] 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.

[15] K. He *et al.*, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[16] S. Ren *et al.*, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

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

[18] J. Dai *et al.*, "R-FCN: object detection via region-based fully convolutional networks," *CoRR*, vol. abs/1605.06409, 2016.

[19] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," *CoRR*, vol. abs/1611.10012, 2016.

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

[21] C. R. Qi *et al.*, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *CoRR*, vol. abs/1706.02413, 2017.

[22] Y. Li *et al.*, "Deepim: Deep iterative matching for 6d pose estimation," *CoRR*, vol. abs/1804.00175, 2018.

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

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

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

[26] 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.

[27] Y. Konishi *et al.*, "Real-time 6d object pose estimation on CPU," *CoRR*, vol. abs/1811.08588, 2018.

[28] T. Lin *et al.*, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

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

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

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

[32] 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.

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

[34] 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.

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

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

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

[38] C. Qi *et al.*, "Frustum pointnets for 3d object detection from RGB-D data," *CoRR*, vol. abs/1711.08488, 2017.