Industrial Application of 6D Pose Estimation for Robotic Manipulation in Automotive Internal Logistics

Philipp Quentin[1,2], Dino Knoll[1] and Daniel Goehring[2]

***Abstract*— Despite the advances in robotics a large proportion of the of parts handling tasks in the automotive industry's internal logistics are not automated but still performed by humans. A key component to competitively automate these processes is a 6D pose estimation that can handle a large number of different parts, is adaptable to new parts with little manual effort, and is sufficiently accurate and robust with respect to industry requirements. In this context, the question arises as to the current status quo with respect to these measures. To address this we built a representative 6D pose estimation pipeline with state-of-the-art components from economically scalable real to synthetic data generation to pose estimators and evaluated it on automotive parts with regards to a realistic sequencing process. We found that using the data generation approaches, the performance of the trained 6D pose estimators are promising, but do not meet industry requirements. We reveal that the reason for this is the inability of the estimators to provide reliable uncertainties for their poses, rather than the ability of to provide sufficiently accurate poses. In this context we further analyzed how RGB- and RGB-D-based approaches compare against this background and show that they are differently vulnerable to the domain gap induced by synthetic data.**

I. INTRODUCTION

Despite the current advances in robotics and artificial intelligence a large proportion of processes in the internal logistics of the automotive industry are still carried out by humans. This especially concerns, but is not limited to, the process of parts sequencing for the assembly line. The essential cognitive task which is performed by humans to accomplish this process is the visual recognition that enables the picking and placing of automotive parts. The requirements for a 6D pose estimation system to competitively enable a robot to perform these tasks instead, are exceptionally high. On the one hand, this is due to the high number of different parts combined with -in the robotic sense- highly unstructured environments. On the other hand, this results from the need for short cycle times combined with near-optimal availability that leaves no room for robot errors, such as robotic crashes. This requires from the 6D pose estimation system to not only be fast, effortless maintainable and scaleable to this large number of different parts, but also to be robust against occlusions and erroneous sensor data.

Against this background the recent advances of neural networks in the field of 6D pose estimation, are promising to solve the above challenges. Current leading approaches are nowadays mainly based on neural networks and outperform traditional approaches [1]. Together with the developed frameworks to easily generate real-world [2] and theoretically unlimited synthetic [3] training data, this may have the potential to competitively enable the automation of the internal logistics. Therefore the question to current state of a 6D pose estimation pipeline applied the internal logistics in industry arises. Despite the various recent works in the fields of data generation and 6D pose estimation, there is to the best of our knowledge, no work which applies a full integrative pipeline of these blocks with such a focus.

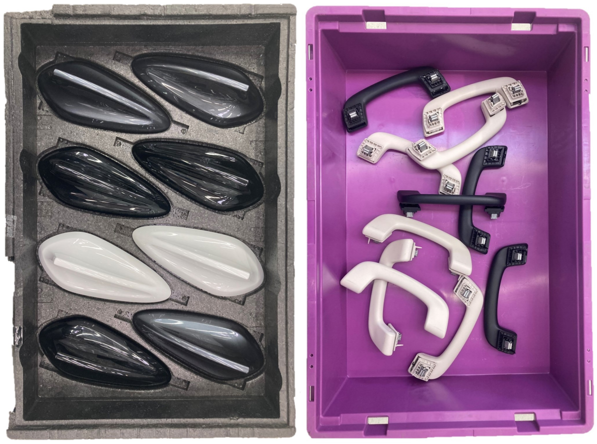

Fig. 1. Automotive antenna covers (left) and interior handles (right) in storage containers, that enforce structured and unstructured positions.

Therefore we aim to deepen the insights on this topic with building blocks from data generation to 6D pose estimation on exemplary automotive parts. To this end, we build a pipeline consisting out of three main state-of-the-art building blocks. One block for real-world data generation which implements LabelFusion [2], one block for synthetic data generation which applies NVISII [3] and one block for the 6D pose estimation itself. The 6D pose estimation block implements the state-of-the-art RGB-based approach GDR-Net [4] and the RGB-D-based approach DenseFusion [5]. We applied the pipeline to two exemplary automotive parts shown in Fig. 1, namely: antenna covers (antennas), which are supposed to represent rather difficult parts due to their their geometric and visual properties, and interior handles (handles), which are supposed to represent rather simple parts. To evaluate the pipeline with respect to industry requirements, we conducted experiments on the generated data sets together with robotic grasp and place tests on a representative sequencing process.

We found that using the data generation approaches, the performance of the trained 6D pose estimators is promising

[1] BMW Group, Munich, Germany

[2] Freie Universitaet Berlin, Dahlem Center for Machine Learning and Robotics, Berlin, Germany

in terms of scalability, but does not meet industry requirements in terms of robustness. We show that the reason for this is the inability of the estimators to provide a reliable uncertainty measure, rather than the ability to provide sufficiently accurate poses. Furthermore, we observe that in the case of the RGB-D-based estimator, despite its reliance on neural networks, it is not robust to slightly erroneous depth data, and we infer that the circumstance of not having depth data in the case of the RGB-based estimator significantly amplifies the domain gap on the distance estimate.

II. RELATED WORK

To the best of our knowledge, there is no direct comparable work, which evaluates a pipeline from data generation to 6D pose estimation on automotive parts in terms of feasibility and scalability with respect to industry requirements. Therefore, we discuss the works which are directly relevant for the building blocks of our pipeline.

A. *Real-World Data Generation*

The majority of the work towards methods for real-world data generation is contained in the publications of the several established data sets [6]–[10], which were created to evaluate and benchmark 6D pose estimators. A core approach among these works is to arrange the target objects in a static scene. Then either the sensor setup [6], [8], [9] or the static scene [7] is moved to generate samples from different viewpoints. Subsequently, the 6D poses of the target objects are either annotated in one sample [7], [8], [10] or in a 3D reconstruction [6], [9] of the scene and transferred to all samples via estimation, geometric relations or by a combination of both. This strategy to transfer one annotation to all samples in a static scene, enables the annotation of large datasets with only little human effort.

However, these works focus on the properties of the data sets and the evaluation of 6D pose estimators and less on their data generating methods. As a consequence they often do not contain detailed descriptions of these methods and the source code is not published or not completely open source as in case of [9]. In contrast to that LabelFusion [2] is open source and targets the data generation for 6D pose estimation in terms of practicability and scalability. It also follows the approach of static scenes in combination with a 3D reconstruction. As an improvement to [6], [9] it further does not require markers in the scene, which enables a high freedom for scene generation. Therefore, we decided for LabelFusion as our pipeline building block for real-world data generation.

B. *Synthetic Data Generation*

The main challenge of synthetic data generation is to effectively bridge the domain gap so that 6D pose estimators trained on synthetic data perform equally well on real-world data. To address this challenge two complementary approaches have emerged. One is domain randomization [11], which aims to strongly randomize synthetic data, such as textures, backgrounds, lighting conditions, etc., so that real-world data simply represents another of these variations to the trained models. The other is to generate photorealistic data, so that the deviation of the synthetic from the real domain becomes minimal.

In this context, [12]–[14] pursued the so-called ”render & paste” approach [1], which renders objects in combination with domain randomization techniques in front of random real photographs, achieving promising results. The main difference between these works is that [12], [13] used a rasterization-based renderer (RBR) and [14] used a physically-based renderer (PBR). The difference is that a PBR can accurately simulate the path of light through ray tracing, while a RBR approximates the path of light through rasterization. In consequence, a PBR enables physically-plausible domain randomization techniques, taking into account material properties and light interactions between objects, whereas a RBR is limited to approximated light effects and image-based domain randomization as for example randomized image contrasts [15]. Against this background the BOP challenge of 2020 [1] showed that domain randomized PBR images of objects simulated in an 3D cube, representing lightweight photorealistic scenes, improved the results on 6D pose estimation compared to RBR ”paste & render” images by large margins. Related to that DOPE [16], which was entirely trained on PBR ”paste & render” images and fully PBR photorealistic indoor and outdoor scenes, achieved on par results on the established real-world YCB-V data set [8] with 6D pose estimators trained on real data. It was shown that both of these synthetic data generating approaches were essential to reach these results.

With regard to this, the domain randomized PBR approach of the BOP challenge to generate lightweight photorealistic scenes is a good compromise in terms of scalability, as it improves on the ”paste & render” approach but does not go along with the high effort to generate full photorealistic scenes. Therefore, we explore further on this approach.

C. *6D Pose Estimation*

6D pose estimation is nowadays mostly lead by deep learning based estimators. Among these, the current leading ones can be subdivided into those rely solely on RGB images, or additionally use depth information from an RGB-D camera. [17]

1) RGB-based: Deep learning based RGB 6D pose estimators can be divided into direct, correspondence, or implicit representation based estimators. Direct estimators [8], [18] take the RGB image as an input and directly estimate a fixed 6D pose representation via a CNN architecture. A disadvantage of these methods is that they tend to suffer from difficulties during the learning process in case of existent pose ambiguities [13]. Correspondence based methods estimate the corresponding coordinates of either sparse [16] or dense keypoints [19], [20] of the target object and the object model to then retrieve the 6D pose via a perspective-n-point algorithm. This decouples the learning process from the fixed 6D pose representation and additionally allows the possibility to compute multiple pose hypotheses in case of uncertainty.

Implicit representations based methods use the object model to render it at different poses to generate a comprehensive pose-view mapping. Later at runtime the actual pose is then determined by a match of the present view to the pose-view mapping via a latent space representation generated by an autoencoder [13]. The advantage of this approach is, that it does not need ground truth pose annotations for training and is not suffering from pose ambiguities.

In this context, the leading RGB-based estimator on the BOP challenge 2022 [17] is GDR-Net [4], which combines and unifies the correspondence with direct approach. It even surpasses RGB-D-based estimators on the challenge, and therefore, we chose it as our RGB-based estimator.

2) RGB-D-based: Leading RGB-D based approaches can be divided into deep learning approaches that are extended by classical approaches, pure classical approaches or pure deep learning approaches. The former of these make use of the additional depth information by taking an initial 6D pose of an RGB-based estimator and refine this pose by e.g. an iterative closest point (ICP) algoritm [8], [13] or by comparing distances as in the fast version of GDRNPP [21]. A pure classical approach, which dominated the BOP challenge until recently, is based on the matching of the observed and model point cloud via point pair features [22]. However, a strong drawback of using depth with classical approaches is that they rely on high quality depth information and suffer from erroneous depth values, which are particularly prevalent on reflective surfaces as on the antenna. Against this background pure deep learning approaches process RGB and depth information jointly via a deep neural network to either estimate the 6D pose directly [5] or via correspondences [23], [24]. The joint learning-based approach promises to be capable to leverage the information of both modalities, while handling low-quality depth information, but still making use of it.

For this reason, and for the reason that [23], [24] are not straightforwardly multi-object multi-instance-per-class capable, we decided for the well established direct approach DenseFusion [5] of this class, which additionally fulfills the requirement of a fast inference time.

III. 6D POSE ESTIMATION PIPELINE

Following section II our pipeline consists out of three state-of-the-art building blocks for 6D pose estimation for rigid and known objects. One block for real-world data generation which implements LabelFusion [2] that serves as an highly automated generator for training, evaluation and test data. One block for synthetic data generation which applies NVISII [3] and serves as a generator for unlimited synthetic training and validation data and one block for the 6D pose estimation itself. For the 6D pose estimation block we selected GDR-Net [4] and DenseFusion (DF) [5] as they reflect leading concepts of RGB- and RGB-D-based 6D pose estimators. To acquire the RGB and RGB-D images, we have selected the low-budget Framos D435e camera, an industrialized version of the Intel Realsense D435. In doing so, we cover representative and well established state-of-the-art data generation and RGB- and RGB-D-based 6D pose estimation approaches.

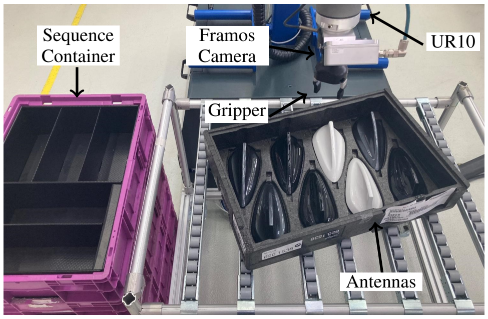

Fig. 2. Representative robotic setup for a automotive sequencing process showing antennas on a conveyor belt in front of a UR10 with a two finger gripper and a Framos camera mounted on the end effector. Left to the conveyor belt is the target sequence container.

A. Real-World Data Generation

To generate annotated ground truth real-world data we use LabelFusion in combination with a partial automated scene recording. In order to create representative data we generate scenes, which are similar to sequencing for an automotive assembly line.

In detail, we mount the camera on the end effector of a robotic arm as shown in Fig. 2. Then for each scene we randomly place a storage container, containing randomly arranged objects of one type, on a conveyor belt in front of the robotic-camera setup. To annotate a scene we let the camera move along a predefined trajectory to collect RGB-D samples from different viewpoints. Given these samples, LabelFusion creates a 3D reconstruction of the scene while simultaneously estimating the camera pose for each sample. In the next step the target objects in the 3D reconstruction are aligned with their corresponding models meshes through a human assisted ICP-fitting. Given the alignment, for each sample bounding boxes, masks and 6D poses are automatically retrieved, resulting in the annotation of the whole scene.

This procedure enables us to quickly annotate large amounts of data. Furthermore, the usage of predefined instead of hand-held trajectories ensures reproducible results and enables cycles for iterative improvement.

B. Synthetic Data Generation

To generate the lightweight photorealistic scenes as described in II-B, we use NVISII which is an open-source PBR renderer based on Nvidia's OptiX ray tracing engine [3].

Specifically, we create two types of such scenes. In the first, we spawn the part's meshes in a random position on a physical floor, so that they have the same face up as in the real sequencing process. This is meant to represent simple scenes to get the model's learning process going in the right direction and to force a strong prior on this likely pose. In the second, we spawn the part's meshes together with distractor objects in a random position above the physical floor and let them fall onto it. This way we get a scene with all kinds of poses and occlusions to further improve the model's generalization capability.

For both scenes the parts color, texture and material properties are varied, so that we obtain a mixture of parts with a strong randomized appearance and parts with an appearance similar to that of the real world. In addition, we randomize the floors appearance and the main scenes illumination texture with HDRI images and images from ImageNet [25]. Besides the main illumination, we also add a random amount of spotlights at random positions. Given these scenes, RGB-D images are rendered from the perspective of a virtual camera, whose position varies randomly between different frames. Exemplary rendered images of these scenes are displayed in Fig. 3. Finally, annotations in the form of ground truth poses, masks and bounding boxes are retrieved for each camera. Given this pipeline, unlimited domain randomized lightweight photorealistic data can be generated, while no additional personnel effort is needed.

C. 6D Pose Estimation

For both, DF and GDR-Net we adhere to the architectures and specifications used in the original publications [4], [5]. For the required detection block of both estimators, we utilize Mask R-CNN [26] following [14], [18].

IV. EXPERIMENTS

In the experiment section, we evaluate our pipeline on representative parts of the sequencing process of an automotive assembly line. To do so, we have chosen two common parts of this process, namely antennas and handles, displayed in Fig. 1. In this context the antenna, represents a difficult part as it is highly reflective and has no distinctive textures. The handles on the other hand, represent a rather simple part. Given these parts we applied our pipeline and conducted tests on collected data as well as on a robotic setup to examine the following questions: (1) What is the accuracy and robustness that we can reach and how does this compare to the requirements for an industrial deployment? (2) How high is the human effort, is real data still needed, or is our lightweight approach for generating synthetic data already sufficient? (3) How do the selected RGB- and RGB-D-based pose estimation approaches compare in this context?

A. Metrics

For the evaluation of the pose estimators on the data sets, we want a practical pose error function that indicates with a high likelihood the chance of a successful grasp to enable a detailed analysis with respect to process requirements.

As like the antennas and handles, the vast majority of automotive parts in the sequencing process are symmetry-ambiguity-invariant. Therefore, we follow [7] to use a pose error function that computes the maximum distance error d^P (MDE) between corresponding points in homogeneous coordinates $\mathbf{x}$ of the model $\mathcal{M}$ in annotated ground truth pose $\bar{\mathbf{P}}$ and estimated pose $\hat{\mathbf{P}}$ as

$$d^P(\bar{\mathbf{P}}, \hat{\mathbf{P}}; \mathcal{M}) = \max_{\mathbf{x} \in \mathcal{M}} \|\bar{\mathbf{P}}\mathbf{x} - \hat{\mathbf{P}}\mathbf{x}\|_2 \tag{1}$$

where a pose is defined as a homogeneous transformation matrix $\mathbf{P} = [\mathbf{R} \mid \mathbf{t}] \in \mathrm{SE}(3)$, consisting of a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. In doing so, we enable a practical but conservative determination of a single threshold towards a successful grasp for a wide range of grippers.

Based on this, we can calculate performance scores for industrial requirements. In our case we must avoid false positives (FP), as they can lead to robotic crashes and damaged parts. Therefore, we are interested in the precision score. Furthermore, we need also take measure on false negatives (FNs) as they lead to longer process times and can make a pose estimator impractical. Thus, we are also interested in the recall score. To this end we calculate the average precision (AP) and average recall (AR) for N samples as

$$AP = \frac{1}{N} \sum_{n=1}^{N} \frac{TP_n}{TP_n + FP_n} \tag{2}$$

$$AR = \frac{1}{N} \sum_{n=1}^{N} \frac{TP_n}{TP_n + FN_n} \tag{3}$$

where TP_n, FP_n and FN_n are the number of the true positives (TPs), FPs and FNs in a sample n, respectively.

To compute these scores we match all pose estimations of a sample n to the ground truth (GT) annotations in increasing order of the corresponding maximum distance error d^P, while every estimation can only be matched once but every GT multiple times. All matched estimations whose d^P error is smaller or equal to an error threshold θ_p are counted as a TP and a FP otherwise. From all GTs, only GTs that do not have a TP match and have a visibility ratio v of a corresponding object in a sample greater or equal to a visibility threshold θ_v, are counted as FP. The reasoning behind this is that in our case we do not have partial visibility due to occlusion but only due to process dependent field of view (FoV) restrictions. Therefore, we do not want to penalize missing estimations for objects, for which it is not intended to predict poses for, as the visible part of such objects may not even contain enough information for an optimal 6D pose estimator. Nevertheless, if the estimator provides a pose for such an object, its error has to be smaller or equal to the error threshold otherwise it is counted as a FP. In this context we set the $\theta_v = 0.85$, as we require from the process to enable such a visibility and the error threshold $\theta_p = 1.5$ cm, as this should be the upper limit for a successful grasp for a wide variety of part-gripper combinations.

B. Datasets

1) Real-World Data: Using our real-world data pipeline we generated a data set for each part. We collected the RGB-D images with a resolution of 848x480 pixels, which is the resolution of the Framos to achieve the highest depth quality. For each part we decided to collect 30 scenes at the heights around of 30 cm, 10 scenes 50 cm and 10 scenes 80 cm above the storage container. The height of 30 cm is intended to serve as a distance from which the 6D pose for the grasping will be estimated and therefore makes up the majority of the collected scenes. This height is the minimal distance at which the complete storage container is still in

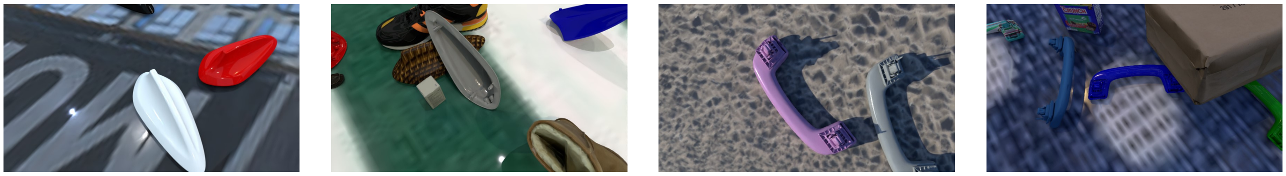

Fig. 3. Exemplary rendered images showing from the left to right right: First scene and second scene type antennas, first and second scene type handles.

TABLE I
DATASET VARIANTS PER PART

Name	Train	Validation	Test
R	80k real img.	25k real img.	25k real img.
S	50k synth. img	10k synth. img.	
R-S	R Train + S Train	R Val. + S Val.	

the FoV and provides a sufficient margin for successful depth images acquisition. The heights of 50 and 80 cm are intended to make the estimators robust against height variations. We divided the total of 50 scenes for each part into 30 training scenes with about 80k images, 10 validation and 10 test scenes with about 25k images each. Each dataset could be generated by one person in about 4 days.

2) Synthetic Data: Using our synthetic data pipeline, we generated 25,000 annotated training and 5,000 validation images for each scene type and part. This results in a total of 50,000 training and 10,000 validation images for each part, which is very similar to the seizes in [1], [16]. The distance of the camera to the target parts was chosen to be similar to the real data, with a distance between 30 and 80 cm. The scenes themselves contain randomly up to 10 distractors, spotlights and target parts each. Example images of the data sets are shown in Fig. 3.

3) Dataset Variants: In order to evaluate our research questions, we created a real world (R), a synthetic (S) and a combined data set (R-S) for each part as specified in Tab. I.

C. Implementation and Training Details

To train Mask R-CNN, we use a pre-trained ResNet-50. For the training of DF, we used the same training hyperparameters as in the original publication with 2 refinement iterations and a refinement threshold of 8 mm, but with a pretrained instead of an untrained ResNet-18. For GDR-Net, we also used the same hyperparameters as in the original publication, but excluded the built-in data augmentation, as we want to compare both estimators on the same data set.

Furthermore, for both 6D pose estimators, we excluded annotations in the training and validation parts of the data sets of objects that do not exceed a visibility of $\theta_v = 0.3$, as we observed that lower thresholds worsen the results. Moreover, we trained and validated on the GT instead on inferred detections, as we observed better results when testing such models with inferred detections on the test set.

D. Results on Dataset Variants

1) Instance Segmentation: The results for the trained Mask R-CNNs evaluated on the test set are shown in Table II. The results are given in percentage for $AP_{50(\%)}$ (the average precision at IoU 50 %), $AP_{50:95(\%)}$ and AR^{100} (the average recall for at maximum 100 detections per image). It can be concluded that both the models trained on real data and the model trained on real and synthetic data combined reach state of the art scores. The model trained on synthetic data only, do not, especially the version for the antenna.

TABLE II
MASK R-CNN RESULTS

Object	Dataset	$\mathbf{AP_{50:95(\%)}}$	$\mathbf{AP_{50(\%)}}$	$\mathbf{AR^{100}}$
Antenna	R	85.2	99.0	89.0
	S	39.5	55.8	44.1
	R-S	85.7	99.0	89.8
Handle	R	73.5	99.0	78.5
	S	63.8	96.1	68.5
	R-S	75.3	98.9	78.9

TABLE III
AVG. PRECISION & AVG. RECALL DATASET RESULTS

Object	Method	R		S		R-S	
		AP	AR	AP	AR	AP	AR
Antenna	DF	95.2	89.3	70.8	28.7	96.6	92.7
	GDR	77.4	96.3	73.0	40.9	78.6	97.2
Handle	DF	97.7	91.6	96.3	41.5	99.5	92.8
	GDR	82.6	95.3	29.4	37.3	86.6	98.4

2) 6D Pose Estimation: For the evaluation of the trained 6D pose estimation models, we used the inferred detections from the Mask R-CNNs of the same data variant. Furthermore, we used the possibility that DF provides confidences $c \in [0, 1]$ [5] as an uncertainty measure for its estimation, whereas GDR-Net does not. Unless otherwise specified, we have chosen the confidences for DF so that the AP (2) is maximized, but at maximum to $c = 0.95$, as we want to prevent robotic crashes. The confidences for the DF models for the Antenna/Handle models were set to R: 0.95/0.8, S: 0.5/0.7, R-S: 0.95/0.95.

In Tab. III we tabulate the AP (2) and AR (3) scores in % for the different DF and GDR-Net models. The result show that especially the DF models that included real data in the training are close to an optimal AP of 1 and perform in that term better than the GDR-Net counterpart. Nevertheless, despite that DF provides confidences as an uncertainty measure to adapt the AP there is no model that reaches an optimal AP. Additionally, it can be seen that the GDR-Net predominantly achieves higher AR values, which was to be expected since the poses were not filtered by an uncertainty measure. But again, despite the visibility constraint, optimal AR values are not achieved. Remarkable is the near optimal

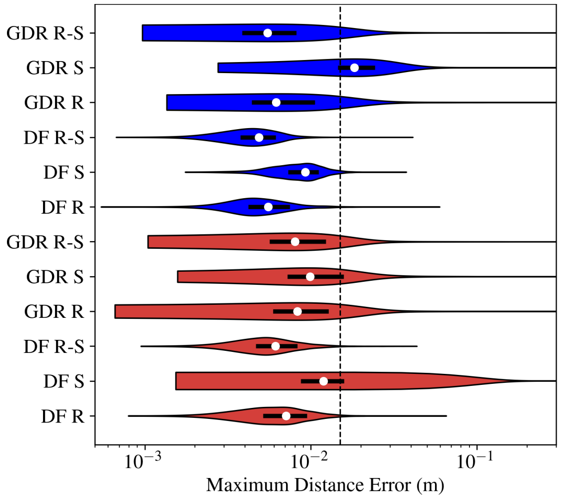

Fig. 4. Violine plots of the MDE on the test set for the different models on a logarithmic axis. The results for the antenna are shown in red and for the handle in blue. The white dots represent the location of the median and the black bars indicate the location of the lower and upper quartiles, enclosing 75 % of the data. The vertical dotted line shows the error threshold θ_p of 0.015 m. The x-axis is limited from 0.5 mm to 0.3 m.

TABLE IV
ADD Error Components

		Antenna				Handle			
		DF		GDR		DF		GDR	
		$\overline{x}$	s	$\overline{x}$	s	$\overline{x}$	s	$\overline{x}$	s
R	x	2.41	1.73	7.57	27.4	2.41	2.40	7.70	23.5
	y	2.81	2.21	6.03	19.3	2.12	1.95	6.04	21.2
	z	3.10	2.43	16.4	64.1	2.04	1.23	14.2	51.9
S	x	9.45	134	5.02	12.9	1.84	1.41	7.20	21.7
	y	7.21	64.2	4.93	11.3	1.74	1.10	7.22	25.5
	z	16.3	385	14.0	41.5	4.75	1.82	24.9	77.7

AP of the DF model trained only with synthetic data on the handle and the generally high scores of the synthetic models with the exception of the GDR-Net on the handle.

To additionally analyze the distribution of the models MDE's we complement the AP and AR scores with a violin plot in Fig. 4. The plot shows that most models have the majority of their MDEs below the critical threshold of $\theta_p = 0.015$ m. Furthermore, it is noteworthy that for the antenna the synthetic model of GDR-Net and for the handle the synthetic model of DF, have similar MDE distributions compared to their real counterparts. However, it is noticeable that all models have quite high MDE outliers and even the best performing model has outliers up to 4 cm.

Furthermore, we tabulate the mean and standard deviation of the x-,y- and z-error components of the ADD error in mm [27] in Tab. IV to enable an analysis of the error sources. Note that in this table a comparison between the real and synthetic data models need to be done with caution, as they had a different detection input and are therefore not evaluated on the same set of instances.

To additionally enable a more in depth analysis of the confidence variable of DF, we display the AP and AR curves over a confidence interval $c \in [0, 1)$ for each DF model in Fig. 5. The curves show that the models have a low sensitivity of the AR and AP scores with respect to the confidence. In particular, the AP course has mostly constant or occasionally even lower scores for higher confidence values and no optimal AP can be set. This problem is particularly evident for the models trained on synthetic data.

TABLE V
Avg. Precision & Avg. Recall Robotic Experiment Results

Object	**Method**	**R**		**S**		**R-S**	
		AP	AR	AP	AR	AP	AR
Antenna	DF	100	87.5	60.8	52.5	100	100
	GDR	100	100	87.1	85.0	97.5	97.5
Handle	DF	100	97.5	97.5	92.5	100	95.0
	GDR	97.5	87.5	0.00	0.00	80.0	60.0

E. Experiments on Robotic Grasping Setup

To evaluate how the models would perform in a real sequencing process, we performed tests on the robotic setup of Fig. 2 with a two-finger gripper that is capable of grasping both parts, and is relatively robust to pose inaccuracies. To conduct the experiments, we placed a randomly oriented storage container containing randomly arranged parts of one type on a conveyor belt in front of the robotic setup. The goal was then to sequentially pick all parts in the storage container and place them in a nearby sequence container, where for each attempt a new image of the scene was acquired. To do this, the robot performed a search movement so that the entire area of the conveyor belt was successively within the FoV of the camera. To have equal conditions, each model was tested on the same set of 5 randomly prepared scenes per part, where each scene contained 8 parts.

As the experiments offer the opportunity to make the AP and AR scores more meaningful for evaluating a deployment, we deviated from the definition of TP, FP and FN of the data set evaluation. Specifically, we counted an attempt as a TP if the robot managed to place a part in the sequence container. The slot position in this container was hard-coded, and to account for acceptable inaccuracies, the slot was chosen to be 20 mm larger in x and y directions than the maximum x and y dimensions of the respective parts in the intended placement position. An attempt was counted as a FP, if it lead to a crash or to an unsuccessful placement. We removed each part after it caused a crash so a FP could only be counted once per part. A FN was counted for each part that remained in the container after 160 seconds, which equals 20 seconds per part and is a realistic assumption. The parts that were removed because of a crash were counted as a FN.

We display the results of the experiment in Table V. The results show that optimal AR and AP values could be obtained for both parts with DF and GDR-Net. It should be emphasized that the purely synthetically trained DF model has nearly optimal values for the handles. In contrast, it performs relatively poorly for the antennas. We observed that this was especially the case when the depth images were erroneous. Fig. 6 shows such an example from the experiment. In contrast, GDR-Net trained on synthetic data was much more robust on the antennas. However, the poor

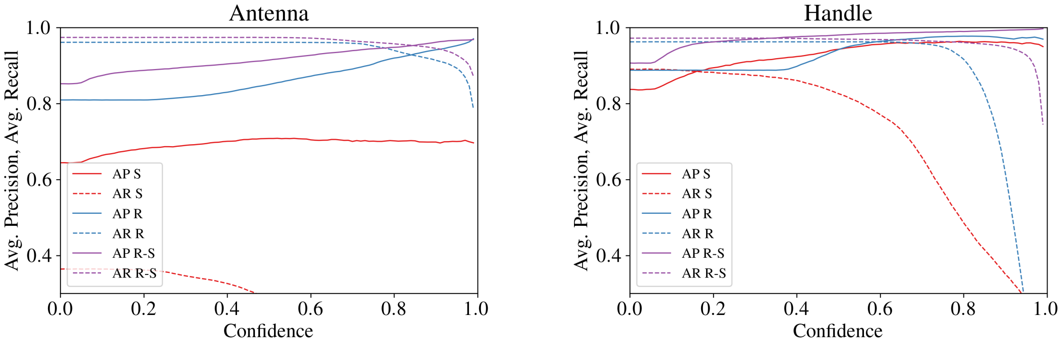

Fig. 5. Average Precision and Average Recall of DF over a confidence intervall $c \in [0, 1)$ for the Antenna on the left and the Handle on the right.

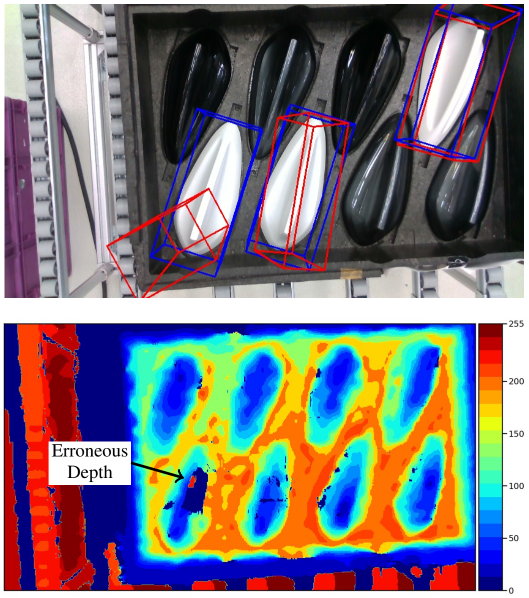

Fig. 6. RGB image with projected pose estimations of DF (red) and GDR (blue) in the form of 3D bounding boxes with a scaled heatmap of the corresponding depth image below. The wrongly estimated poses of DF on the bottom left correspond to depth areas which have false depth values.

values of the synthetically trained GDR-Net for the handle are particularly striking. Here, we observed that the poses were generally estimated too high and the grab was reaching into the void. In this context Tab. IV shows that the ADD error on the z-coordinate for this synthetic model is 24.9 mm compared to 14.2 mm for the real counterpart, while the errors on the x- and y-coordinate are in the same range.

V. DISCUSSION

Towards research question (1) we state that despite the existing nearly optimal results in the robotic experiments for both parts, it is questionable that these numbers would hold for continuous industry operation, as no model in the bigger data set achieved such results and Fig. 4 shows high outliers. In particular, we conclude that industry requirements are not met, especially in terms of robustness rather than accuracy. The biggest issue in this context is the guarantee of an optimal AP, as a measure of robustness, which, if not adhered to, leads to robot crashes with damaged parts and downtime. In our view, a major reason for not achieving the optimal AP, is the inability of the models to provide a reliable uncertainty measure rather than the ability to provide sufficiently accurate poses. Fig. 4 in particular shows that the majority of the estimated poses have a sufficiently small error. This is also true for most models that have been trained with synthetic data only. In this connection, DF provides a confidence as an uncertainty measure, but the results show that its properties are not sufficient to reliably meet an AP of 1. GDR-Net, although leading the BOP challenge 2022 [17], is even worse, as it does not provide any uncertainty at all. Without such, we conclude that a deployment is not robust to unavoidable occlusions or erroneous sensor data.

However, independent of this issue we can state towards research question (2) that the data set generation effort-benefit relationship, is very promising in terms of economic scalability. By using our pipeline, the real data sets could be generated by one person in about 4 days each and with our lightweight synthetic data approach, no further personnel effort is needed. The results show that there is reasonable evidence to assume that given an reliable uncertainty estimation, the results achievable with the real data are sufficient for an industrial deployment. Regarding the question if this would also hold for the lightweight synthetic data approach, we found that, despite the promising results, it depends on the used estimator-data-modality combination (3).

In this context the example of the reflective antenna shows that erroneous depth data leads to a large domain gap with wrong poses, for the DF as the representative of an RGB-D based approach. The GDR-Net, as a representative of an RGB-based approach, that does not use such depth data, performed significantly better in this case. On the other hand, the missing depth makes the GDR-Net generally more inaccurate on the distance estimate, which is likely to lead to insufficient results, if the gripping point is intolerant against such errors, which was the case for the handle in combination with synthetic data. We assume that the domain gap amplifies the disadvantage of an RGB-based estimator of not using direct depth information. Against this background, we suggest that the lightweight approach can be sufficient with either an RGB-D or RGB-based approach for parts that are not reflective, or if they are, a grab point must be available that is tolerant of z-errors.

Note that the generalizability of the conclusion drawn on our research question may be limited by the carefully selected parts and estimators. Nevertheless, we are convinced that the results provide valid insights into these questions.

VI. Conclusions

In this work, we evaluated a representative 6D pose estimation pipeline to investigate the status quo of whether automotive internal logistics can be competitively automated by robots. In this context, we show that a major problem to enable such a competitive automation is that the estimators do not provide reliable uncertainties for their estimates. Therefore, we suggest a strong focus of further research on this property. Furthermore, we show that the selected representative RGB approach is more robust than the selected representative RGB-D approach when a domain gap is caused by erroneous depth data. Since RGB is inherent to the RGB-D approach, this implies that RGB-D approaches can be significantly improved. This could be achieved by improving the network's awareness of erroneous depth data, or by incorporating such errors into the synthetic data generation and training of the networks. Finally, this work demonstrates the difficulties in transferring the promising scientific results of 6D pose estimation to industrial use cases.

Acknowledgment

The authors would like to thank Matthias Brucker and Luca Della Libera for their help on refactoring DF and LabelFusion, Jan-Oliver Seidenfuss and Daniel Derkacz-Bogner for their help on the synthetic data pipeline.

References

[1] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas, "BOP challenge 2020 on 6D object localization," in *European Conference on Computer Vision Workshops (ECCVW)*, 2020, pp. 577–594.

[2] P. Marion, P. R. Florence, L. Manuelli, and R. Tedrake, "LabelFusion: A pipeline for generating ground truth labels for real RGBD data of cluttered scenes," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[3] N. Morrical, J. Tremblay, S. Birchfield, and I. Wald, "NVISII: Nvidia scene imaging interface," 2020, https://github.com/owl-project/NVISII/.

[4] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 611–16 621.

[5] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3338–3347.

[6] T. Hodaň, P. Haluza, S. Obdrzálek, J. Matas, M. I. A. Lourakis, and X. Zabulis, "T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects," in *Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 880–888.

[7] B. Drost, M. Ulrich, P. Bergmann, P. Härtinger, and C. Steger, "Introducing MVTec ITODD - A dataset for 3D object recognition in industry," in *International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 2200–2208.

[8] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

[9] R. Kaskman, S. Zakharov, I. S. Shugurov, and S. Ilic, "HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects," in *International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2767–2776.

[10] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. Glent Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.-K. Kim, J. Matas, and C. Rother, "BOP: Benchmark for 6D object pose estimation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 19–34.

[11] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[12] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 1530–1538.

[13] M. Sundermeyer, Z.-C. Márton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 712–729.

[14] T. Hofer, F. Shamsafar, N. Benbarka, and A. Zell, "Object detection and autoencoder-based 6D pose estimation for highly cluttered bin picking," in *International Conference on Image Processing (ICIP)*, 2021, pp. 704–708.

[15] M. Denninger, M. Sundermeyer, D. Winkelbauer, D. Olefir, T. Hodan, Y. Zidan, M. Elbadrawy, M. Knauer, H. Katam, and A. Lodhi, "BlenderProc: Reducing the reality gap with photorealistic rendering," in *Robotics: Sciene and Systems (RSS) Workshops*, 2020.

[16] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning (CoRL)*, 2018.

[17] M. Sundermeyer, T. Hodaň, Y. Labbe, G. Wang, E. Brachmann, B. Drost, C. Rother, and J. Matas, "BOP challenge 2022 on detection, segmentation and pose estimation of specific rigid objects," in *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023.

[18] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 574–591.

[19] K. Park, T. Patten, and M. Vincze, "Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 7667–7676.

[20] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 7677–7686.

[21] X. Liu, R. Zhang, C. Zhang, B. Fu, J. Tang, X. Liang, J. Tang, X. Cheng, Y. Zhang, G. Wang, and X. Ji, "GDRNPP," https://github.com/shanice-l/gdrnpp_bop2022, 2022.

[22] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 998–1005.

[23] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation," *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 629–11 638, 2020.

[24] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "FFB6D: A full flow bidirectional fusion network for 6D pose estimation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3003–3013.

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

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[27] 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," in *Asian Conference on Computer Vision (ACCV)*, 2012, pp. 548–562.