Effects of Markers in Training Datasets on the Accuracy of 6D Pose Estimation

Janis Rosskamp, Rene Weller, and Gabriel Zachmann
Computer Graphics and Virtual Reality, University of Bremen
{j.rosskamp,weller,zach}@cs.uni-bremen.de

Abstract

Collecting training data for pose estimation methods on images is a time-consuming task and usually involves some kind of manual labeling of the 6D pose of objects. This time could be reduced considerably by using marker-based tracking that would allow for automatic labeling of training images. However, images containing markers may reduce the accuracy of pose estimation due to a bias introduced by the markers. In this paper, we analyze the influence of markers in training images on pose estimation accuracy. We investigate the accuracy of estimated poses for three different cases: i) training on images with markers, ii) removing markers by inpainting, and iii) augmenting the dataset with randomly generated markers to reduce spatial learning of marker features. Our results demonstrate that utilizing marker-based techniques is an effective strategy for collecting large amounts of ground truth data for pose prediction. Moreover, our findings suggest that the usage of inpainting techniques do not reduce prediction accuracy. Additionally, we investigate the effect of inaccuracies of labeling in training data on prediction accuracy. We show that the precise ground truth data obtained through marker tracking proves to be superior compared to markerless datasets if labeling errors of 6D ground truth exist. Our data generation tools are available online: `https://github.com/JHRosskamp/6DPoseDataGenTools`

1. Introduction

The estimation of object 6D poses in images is important for many applications. In robotics, the prediction of poses is a necessary requirement to allow robots to grasp objects and manipulate their environment [3, 36]. In augmented reality, knowing the 6D pose can be used to provide detailed instructions during assembly tasks [20, 23]. Pose estimation is also crucial in medicine, i.e., during surgeries to track tools for guided navigation [6, 19, 22].

While marker and sensor-based systems such as *Polaris* offer a precise pose estimation, they are, compared to image-based camera methods, less flexible in their applications and require specialized hardware. For this reason, deep learning-based pose estimation on images is an active area of research, with many improvements in precision and generalizability over the last years [1, 18, 25, 35].

(a) Image without markers (b) Image with markers

(c) Image with marker inpainting (d) Image with marker augmentation

Figure 1. An example of a rendered image for our teabox dataset (red object in the middle). We use the same objects, lighting and camera positions for all cases a)-d) to fairly compare the prediction performance. In c) small artifacts from inpainting are still visible.

An important requirement for pose estimation with deep learning is the pre-existing high-quality training data. For different pose estimation applications, task-specific training data, depending on the objects and scene, must be collected. The creation of datasets for image-based methods is time-consuming and requires a carefully selected recording environment for accurate ground truth estimation of 6D poses [14]. And while domain randomization or photorealistic images can be used for training, it is usually best to fine-tune on real data [28, 29].

Using marker-based object tracking makes it easy to accurately collect the ground truth of 6D poses for large numbers of images. However, a training set collected in this way will have objects with markers occurring in each image. Naturally, these markers do not exist during prediction

for general applications. It is unclear, how the model performs if markers were present during training but are missing at test time. To the best of our knowledge, the influence of marker-based images on pose estimation with deep learning has not been investigated yet.

In this paper, we analyze the influence of a training set, collected with a marker-based approach to generate ground truth poses, on the performance of the pose prediction when no markers are present at runtime. We do this by creating synthetic datasets using physically-based rendering. It consists of pairs of images, one with markers and one without. In both cases, the ground truth 6D pose is given by the renderer. This offers the advantage of having training data with identical scenes and 6D poses. Hence, a fair comparison of both sets without introducing any discrepancies due to labeling or different scenes are possible.

We quantify the error introduced by the marker-based training set for three different cases: i) unedited images with markers for tracking, ii) markers removed using inpainting, and iii) rendering markers in the synthetic training data at random positions on the object, for which the pose estimator is learned. An example of these three cases and the corresponding markerless image are shown in Figs. 1a to 1d. We show that the influence of marker-based training data on pose estimation is small and can be reduced even further with marker removal using inpainting. Additionally, we demonstrate that marker removal using inpainting is also applicable in scenarios where real images are used for training. This means that if marker-based tracking of real objects, to generate ground truth poses, is used, creating training data will be much easier since manual labeling (i.e., pose reconstruction) is no longer necessary.

Furthermore, we analyze the influence of incorrectly labeled training data on pose prediction. This error in the training set might be introduced by not carefully managing the recording environment or imprecise manual annotation. We investigate this error by creating datasets with mislabeled poses and comparing them to the baseline dataset without any errors. We show that the accuracy of pose prediction using incorrectly labeled data is worse, even for small labeling errors, compared to using unedited marker-based data. In summary, we show in this paper that

- From our proposed methods, marker removal using inpainting performs best. In fact, it has the same accuracy of 6D pose prediction as a network trained on markerless images with precise ground truth.

- If the 6D pose ground truth is labeled imprecisely for markerless images, the precise ground truth of marker-based training data results in better accuracy of pose predictions. This is even the case when unedited images with visible markers are used as training data.

2. Related Work

Several classes of methods for estimating the 6D pose of objects exist. We briefly review two significant classes for pose prediction: image-based and marker-based methods, both of which are of interest within the context of this paper. Subsequently, we discuss widely-used training sets and their respective data collection methods.

Image-based tracking In recent years, pose estimation of objects from RGB images using deep learning has achieved impressive quality, often comparable to depth-based methods [16]. One substantial advantage is the cost-effective setup for tracking. Typically, a simple RGB camera suffices, with no additional sensors required. Compared to marker-based tracking, the object's 6D pose can also be estimated in environments where a motion-capturing system hasn't been installed, such as in-home applications, or where objects cannot be prepared for motion tracking because it would render them ungraspable by robots.

Marker-based tracking Compared to marker or sensor-based tracking, image-based tracking, whether based on RGB or RGB-D, is less accurate, and metrics often measure if a pose is within 10% of the object's diameter [34]. On the other hand, marker-based tracking systems such as *Optitrack* can achieve sub-millimeter accuracy, and even though magnetic sensor-based tracking systems, such as *Polaris* or [2] are slightly less precise, sub-millimeter accuracies can still be achieved. Image-based tracking requires large volumes of object-specific training data, while marker-based methods do not require specific training and can be utilized in unfamiliar scenarios.

Datasets Numerous datasets for pose estimation tasks exist and are commonly used to benchmark the performance of network architectures. The YCB-V dataset [34], comprising approximately 130,000 frames, is a popular choice for benchmarking [11, 17, 21, 31] and will thus be used in this paper. The authors annotated the first frame using depth data and refined it using signed distance fields. The camera trajectory is then estimated by fixing the object poses, and finally, a global refinement step is applied to each scene. The T-LESS dataset [14] has 38,000 training images and 10,000 test images for each of its three sensors. It uses textureless, industry-relevant objects and collects ground truth data using a turntable with a pattern field. The LineMOD [13] and Occluded LineMOD dataset contain 1,200 manually annotated images. In the ITODD dataset [5], the ground truth was manually estimated and improved with several iterations of ICP for 3,500 images. The DoPose-6D dataset [8] collected 3,300 images using a robotic arm. Data is annotated using depth and color images to create a point cloud and then manually positioning an object in the scene to subsequently refine the pose using ICP methods. In the HOPE dataset [30], the point correspondences between RGB-D depth maps and 3D objects were

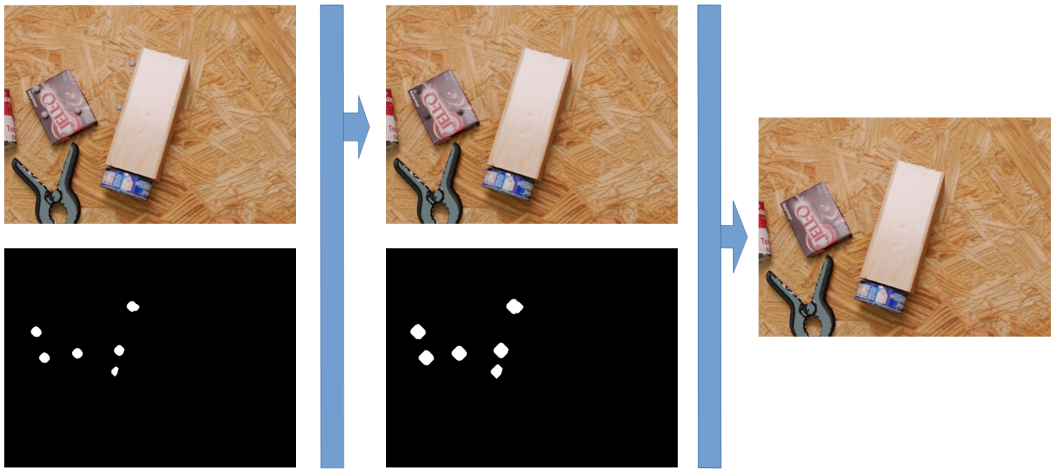

Figure 2. The marker removal pipeline is shown for an image from the YCBV dataset. An image with markers (top left) and the 4x diluted mask (bottom left) is used as input for inpainting. In the resulting intermediate image (top middle) markers are removed, but artifacts are still visible. We use the 8x diluted mask (bottom middle) and the intermediate image as new sources for inpainting. The final image is marker-free and most artifacts are removed.

identified with Procrustes for 238 images. For one-third of SegICP's [33] 7,500 training images, active markers were placed on objects such as an engine for tracking purposes. For the remaining frames, annotation was performed manually. The influence of active markers on pose estimation was not investigated. A commonality among all these datasets is the need for carefully created recording environments and a degree of manual interaction.

Many datasets utilize tabletop scenarios, as mentioned by Gouda *et al*. [8], where, for instance, turntables can be used for data generation. In general, generating new domain-specific training data in arbitrary scenes is a time-consuming task. However, by employing motion-capturing systems, one can quickly and accurately create ground truth poses, allowing even for data collection of objects in motion, such as thrown objects or items on assembly lines.

In the context of hand-object pose estimation, Hein *et al*. [12] and Hampali *et al*. [9] argue that marker-based annotation introduces bias and should be avoided. This might be the case for hand pose annotation, which requires complex sensor setups [7, 10], but object tracking necessitates far fewer sensors. To the best of our knowledge, the influence of markers in the dataset has not yet been systematically investigated.

3. Methods

In this section, we present our methodology, which includes the specifics of the datasets in Sec. 3.1 that are crucial to our investigation. We then address two main aspects: how to quantify the impact of marker-based training data on pose estimation in Sec. 3.2, and measuring the magnitude of annotation errors through manual labeling of images in Sec. 3.3. Additionally, we provide details on the training parameters and network architecture used for our investigation in Sec. 3.4.

3.1. Datasets

We decided to conduct our experiments with synthetic images over real ones. This enables us to generate two sets with exactly the same scenes and camera positions. The only difference lies in the presence or absence of markers; otherwise, the two sets are identical. This approach allows for a fair comparison of results without introducing discrepancies or errors due to slightly different poses or lighting conditions, which are extremely difficult to prevent when recording real scenes. Moreover, synthetic data, with its known 6D pose, facilitates the study of the influence of imprecise labeling on estimation. We generated the synthetic datasets using BlenderProc2, a physically-based rendering software [4], to create RGB-D images. Each scene was created with varying backgrounds and distractor objects, and a physics simulation was employed to generate plausible poses for all objects. For each scene, we rendered 25 images using randomly positioned cameras. We created two distinct sets. The larger set contains 80k images and five different 3D models from the YCBV dataset to compare results on a larger dataset and evaluate on real images. The smaller set comprises 20k images with 800 unique scenes for a single box-shaped object (see Fig. 1), which will be used to examine labeling errors. This dataset also contains a marker-based and a markerless set of 200 real images each and 59 markerless images in 3 scenes for testing.

3.2. Marker-based Training Data

Marker-based systems can automatically generate accurate ground truth of 6D object poses. As mentioned in Sec. 2, images captured in such a scenario might result in reduced performance during evaluation. To quantify the effects of markers, we trained the same neural network under four different conditions of training data:

I Training on images without markers to generate baseline data

II Training on images with markers to ascertain their influence

III Removing markers from images using inpainting methods

IV Augmenting the training set by randomly generating 1-5 markers on objects to learn pose estimations invariant to markers

Figure 1 illustrates the same image modified for all four methods. In the following, we will explain our inpainting III and augmentation method IV in detail.

Figure 3. Real images of the tea box and a texture-less tool with markers (top) and their inpainted counterparts where markers were removed (bottom). This example suggests that our marker removal process does not differentiate between rendered and real images.

Marker Removal

Instead of training on images with markers, modern inpainting methods can be used to eliminate markers from images. We use the pre-trained model of Suvorov *et al*. [27] for inpainting, without any adaptation to our objects and scenes. The removal of image features can be automated using image masks. To create these masks, we use the known marker positions, project them onto the image space, and verify their visibility using the depth image of the scene. While these masks accurately describe the marker positions, they create too many artifacts during inpainting, making them unsuitable. Instead, we create two new masks using binary dilation for the nearest neighbors and dilate four or eight times, respectively. We first apply inpainting using the initial mask and then inpaint again using the second mask. This double masking process reduces the number of artifacts, as shown in Fig. 2, and consistently removes markers. Problems only occur on a few images (see Fig. 1c) where artifacts remain. These could be manually removed, but for the purpose of automatic marker removal, we do not utilize any manual post-processing. Additionally, elongated shadows of markers are not entirely eliminated. However, these occurrences are rare and are typically caused by specific lighting conditions. To ensure the applicability of this process to real images, inpainting is illustrated in Fig. 3. In both scenarios, the markers where successfully removed.

Marker Augmentation

If we train on marker-based images, the neural network may learn markers as important patterns to predict poses. The markers are absent for pose estimation applications, leading to poor results. However, if these markers have random positions on the object and their quantity varies, mitigating this feature by learning of markers might be possible. To explore this hypothesis, we create a dataset in which several markers (between 1 and 5) are sampled onto random positions on the surface of the object (see Fig. 4). If they overlap with each other or the object, we sample a different location. We call this process marker augmentation. This approach is not feasible when real images are captured because we cannot change marker positions for every image without effort. Instead, synthetic images with randomly placed markers can extend the real training set with its fixed marker positions. To simulate this capturing scenario, we split the dataset, where half of the training images have fixed markers, and the other half has augmented markers.

3.3. Annotation Error

As mentioned in Sec. 2, the ground truth for the 6D pose of objects is often estimated using manual annotations. These annotations introduce an error in the dataset with their imprecise poses. We estimate the magnitude of these errors caused by the annotation process. To do that, we manually label the object poses in our synthetic dataset, where the precise ground truth is known. This allows the calculation of the labeling error for the rotation and translation by comparing the manual pose with the exact pose. We label the training images by using the annotation tool

Figure 4. Examples of marker augmentation for the teabox. In these images, between 1-5 markers with random positions are visible.

from [8], provided in the bop toolkit [15]. This tool uses color and depth images to create a colored point cloud, where the user estimates the 6D pose of the objects by placing their respective 3D models as close as possible to the objects point cloud. The pose is then refined using ICP.

Usually, researchers annotating datasets try to minimize the residual errors in the dataset by using sophisticated recording strategies like robotic arms or multi-camera approaches. We choose to annotate object poses manually image by image to provide an upper bound to the error inflicted by human labeling. While annotation is performed more carefully in existing datasets, they might contain depth sensor errors mentioned in Sec. 2, which ultimately change the annotated pose due to their influence on the point cloud used for labeling. We do not take this error into account.

In total, 37 frames with multiple objects were annotated. The resulting errors are shown in Tab. 1 for rotation ΔR and translation Δt errors. Δt can be described using a normal distribution with mean $\mu = 0.1$ and variance $\sigma = 2.2$. The rotation could not be fitted using the most common distributions, which might be due to the small sample size.

To generate a perturbed dataset, we assume that ΔR can also be modeled using a Gaussian distribution. While this may not be entirely accurate, it suffices for estimating the influence of annotation errors. In the perturbed dataset, the color images remain the same, but annotations such as object pose are altered.

3.4. Network and Training

We evaluate our datasets using the GDR-Net framework by Wang *et al*. [32], which uses RGB images to estimate poses. This framework unifies direct regression of the 6D pose with geometry-based indirect methods and performs well on common datasets [26]. During the training process, we retain all parameters and augmentations as proposed by the original authors. We train our small dataset, which contains 20,000 images, with a batch size of 24 for 10 epochs. For the larger YCBV dataset we use a batch size of 24 for 80 epochs. To exclude any differences in the recall due to varying weight initialization, all networks are trained using the same seed.

4. Evaluation

We evaluate our methods on two test sets, the teabox and YCBV dataset. We assess the influence of markers and errors from the ground truth using a synthetic test set of 1000 images for the teabox. Each scene is rendered from two camera positions, resulting in 500 unique scenes.

We conduct tests on synthetic and real data to evaluate the network trained on the subset of YCBV objects. The synthetic set contains 1000 images, with each object appearing around 300 times. For real test images, we use those images from the BOP challenge [15]. In total, we have 975 instances of objects.

Pose accuracy is evaluated using the ADD-S, ADD-(S), and their respective area under the curve (AUC) metrics, with a maximum threshold of 10 cm as described in [34]. The ADD metric measures the average distance of 3D model points between the ground truth pose and the predicted pose. A pose is considered correct if the average distance is smaller than 10% of its diameter. The ADD-S metric averages the distance from the predicted pose to the closest points of the ground truth. For symmetric objects, the ADD-(S) metric utilizes the ADD-S metric; otherwise, it uses the ADD metric.

Δt	0-2 mm	2-4 mm	4-6 mm	6+ mm
N	79	20	9	2
$\Delta\sigma$	0-2 °	2-4 °	4-6 °	6+ °
N	76	12	14	10

Table 1. The manual annotations are compared with the exact 6D ground truth. The error is shown as a distribution of translation errors Δt in intervals of 2 mm and rotation errors ΔR in intervals of 2 °.

	Dataset	AUC of ADD-S	AUC of ADD-(S)	ADD-(S)
I	w/o marker	97.1	90.0	84.8
II	w marker	96.2	87.8	76.6
III	inpainting	97.0	90.0	84.8
IV	marker aug.	97.0	89.8	84.2
V	aug. + marker	96.7	88.8	80.0

Table 2. Evaluation on synthetic test data for the teabox dataset. We evaluate on test data without markers.

	Dataset	AUC of ADD-S	AUC of ADD-(S)	ADD-(S)
VI	w/o marker	96.4	88.3	77.7
VII	w marker	97.5	91.0	87.1

Table 3. Evaluation on synthetic data for the teabox dataset. We evaluate on images with markers.

	Dataset	AUC of ADD-S	AUC of ADD-(S)	ADD-(S)
I	w/o marker	93.6	79.0	45.8
III	inpainting	93.2	79.1	47.5
VIII	w/o finetuning	88.8	71.9	20.3

Table 4. Evaluation on real data for the teabox dataset. We finetuned the baseline model trained on synthetic data w/o markers with real training images and compare them to the synthetic case without finetuning.

4.1. Marker-based Training Data

To accelerate training, we evaluate the four cases discussed in Sec. 3.2 using a teabox dataset of 20,000 images. The most promising method among these cases will be subsequently applied to the YCBV dataset.

Evaluation on the teabox dataset

We assess the predictions on the test set for the four specified methods mentioned in Sec. 3.2, and present the results in Tab. 2.

The results are quite similar across all methods, with a maximum deviation of 1 percent point observed in the AUC of ADD-S. Hence, we choose to employ the ADD-(S) metric for comparison, as it provides a clearer distinction in recall differences for our case.

In terms of recall, the results of both inpainting III and marker augmentation IV are comparable with the baseline I, trained on markerless images. Inpainted images exhibit some remaining artifacts and shadows compared to markerless images, which do not reduce performance. Given the results, the marker augmentation method successfully prevents the network from learning marker-specific patterns. However, in a real capture scenario, markers cannot be randomly changed between images, but rather are fixed in position. To simulate this, we use half of the training images with fixed markers, and the other half with randomly placed markers V. While the results are better than those of II, the outcomes of I and III still yield superior performance. Training on non-edited marker-based images II leads to a performance reduction of 5.6 percent points compared to I. This outcome is expected since the neural network learns the markers as additional features for pose estimation. As a sanity check, we included the evaluation using a test set with markers in Tab. 3. Notably, the best overall result VII is obtained when training and evaluating on marker-based images. This outcome aligns with expectations since the markers introduce distinct patterns that can be easily identified. On the other hand, evaluation for a network trained with the baseline dataset VI yields the second-worst results after II. This implies that evaluating objects with small marker-sized occlusions also reduces recall.

We further examine whether the findings from our experiments apply to real training images, in addition to synthetic data. Typically, real training images are used in conjunction with a large set of synthetic images to enhance model generalization. In this study, we fine-tune a model initially trained on the synthetic markerless dataset with 200 real images by using transfer learning. We train for six epochs and evaluate with the real markerless test set with 59 images. The results are shown in Tab. 4. Using markerless real images I or inpainted real images III results in a more than 25 percent point improvement in the ADD-(S) metric compared to the baseline model VIII, which was not fine-tuned. This improvement demonstrates that our real training data has an influence on the performance of the networks. This allows the investigation of the influence of inpainted images in the training set. The results for I and III are comparable, indicating that inpainted real images do not reduce recall of pose estimations.

Evaluation on the YCBV dataset

The effect of images with markers on pose prediction for the YCBV dataset was evaluated using both synthetic data (Tab. 5) and real test data (Tab. 6). We compared the baseline model I with the model trained on inpainted images III, which was the best model from the previous section and is practical to use due to its automatic removal via inpainting.

The results exhibit some variation depending on the object, but on average, both methods are comparable, with an average recall difference of approximately 1% for all metrics. So, on the synthetic test set, our marker removal method yields comparable results to the baseline.

Object	I		III	
	AUC of ADD-S	ADD-S	AUC of ADD-S	ADD-S
002_master_chef_can	81.1	79.9	80.6	77.3
004_sugar_box	79.6	77.4	80.0	77.4
008_pudding_box	79.2	75.8	80.0	76.2
025_mug	78.4	81.0	84.0	82.7
036_wood_block*	81.4	76.1	81.8	73.9
Average	79.9	**78.0**	**81.3**	77.5

Table 5. Evaluation of method I (w/o marker) and III (inpainting) on a synthetic test set. The * denotes a symmetric object.

Object	I		III	
	AUC of ADD-S	ADD-S	AUC of ADD-S	ADD-S
002_master_chef_can	92.1	79.7	94.3	86.7
004_sugar_box	96.8	98.9	96.1	95.7
008_pudding_box	89.1	76.0	89.1	78.7
025_mug	65.3	18.7	52.1	4.7
036_wood_block*	17.7	0.0	4.3	0.0
Average	92.6	84.9	**93.2**	**87.0**

Table 6. Evaluation of method I (w/o marker) and III (inpainting) on a real test set. The average includes only objects {002,004,008}. The * denotes a symmetric object.

However, for the real test data, the results for *object 025_mug* and the symmetric *036_wood_block* are poor for both methods. This discrepancy could be attributed to the gap between synthetic and real data. Therefore, we exclude these objects from further evaluations. Once again, the results demonstrate that marker removal does not decrease recall. When comparing our results with those reported by Wang *et al*. [32], we observe that our results for objects *002_master_chef_can, 004_sugar_box*, and *008_pudding_box* are of the same order of magnitude, despite training being performed on a mixture of real and synthetic data by Wang *et al*. This lends credibility to our comparison between models I and III on real data, as shown in Tab. 7. However, considering the issues with objects *025_mug* and *036_wood_block*, it is important to exercise caution when drawing conclusions from the BOP challenge's test set.

To eliminate potential model-dependent biases from GDR-Net, we verify our results on the YCBV dataset with ZebraPose [24]. In ZebraPose we trained a separate network for each object, limiting our dataset to approximately 10000 images per object. The results for networks trained on markerless images I and inpainted images III are shown in Tab. 8, evaluated on a synthetic and real test set. The average AUC of ADD-S is 0.6 percent points lower for III in comparison to I on synthetic data but is 0.9 percent points higher on real data. Overall, the ZebraPose results are consistent with those obtained from GDR-Net.

Pose Estimator	GDR-Net [32]		Ours	
	1	N	I	III
002_master_chef_can	96.6	96.3	92.1	94.3
004_sugar_box	98.3	98.9	96.8	96.1
008_pudding_box	94.8	64.6	89.1	89.1
025_mug	96.9	99.6	65.3	52.1
036_wood_block*	77.3	82.5	17.7	4.3

Table 7. Comparison of our cases I and III with GDR-Net using AUC of ADD-S on real test data. They trained either one pose estimator for the whole dataset or one pose estimator per object. Our uses one pose estimator for five objects. The * denotes a symmetric object.

Evaluation	I		III	
	syn	real	syn	real
002_master_chef_can	94.8	90.0	94.5	87.4
004_sugar_box	94.1	88.6	93.0	88.6
008_pudding_box	93.4	41.1	92.8	36.9
025_mug	93.7	73.3	93.8	61.5
036_wood_block*	88.4	11.9	87.5	13.6
Average	**92.9**	60.9	92.3	**61.8**

Table 8. Comparison of our cases I and III with AUC of ADD-S on synthetic and real test data using ZebraPose [24]. The * denotes a symmetric object.

4.2. Annotation Error

To evaluate the annotation error, we use the teabox dataset and evaluate on synthetic data only, because here the precise ground truth for the test set is known. We conduct a parameter study to systematically investigate the impact of labeling errors. As mentioned in Sec. 3.3, we describe labeling errors using a Gaussian distribution with variance σ_t and σ_R for the translation and rotation respectively. We choose to set the mean to zero. For the parameter study we decided on $\sigma_\mathrm{R} = \{0, 2.5, 5\}^\circ$ and $\sigma_\mathrm{t} = \{0, 2, 4, 8\}$ mm, where values of $\sigma_\mathrm{R} = 5^\circ$ and $\sigma_\mathrm{t} = 8$ mm are on the higher end and should not occur in labeled data.

The results are depicted in Fig. 5 using the AUC of ADD-(S) metric and in Fig. 6 for the ADD-(S) metric. On the x-axis the variance of translation is shown and the three curves represent the rotational error.

Without any rotational or translational errors, we achieve

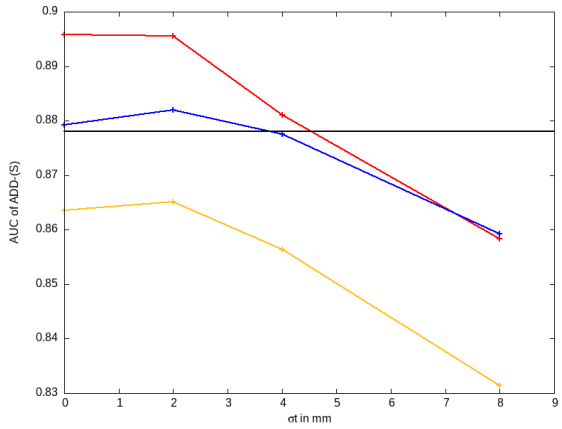

Figure 5. AUC of ADD-(S) dependent on σ_t for $\sigma_R = 0\,^\circ$ (—), $\sigma_R = 2.5\,^\circ$ (—), and $\sigma_R = 5.0\,^\circ$ (—). The horizontal black line represents the result for marker-based training from method II.

the best results, which corresponds to case I from Sec. 4.1. We can further see that a small error $\sigma_\mathrm{t} = 2\,\mathrm{mm}$ does not impact the results much. For translational errors above $\sigma_\mathrm{t} = 4\,\mathrm{mm}$, we can see a decrease of metrics. A rotational error of $\sigma_\mathrm{R} = 2.5\,^\circ$, on the other hand, decreases recall by 1.5-7 percent points, depending on the metric. Interestingly, in the case of $\sigma_\mathrm{t} = 8\,\mathrm{mm}$ and $\sigma_\mathrm{R} = 2.5\,^\circ$ results are slightly better compared to $\sigma_\mathrm{t} = 8\,\mathrm{mm}$ and $\sigma_\mathrm{R} = 0\,^\circ$

The horizontal line shows the corresponding result of the worst marker-based method; case II. We can see, that labeling errors exceeding $\sigma_\mathrm{R} = 2.5\,^\circ$ and $\sigma_\mathrm{t} = 2\,\mathrm{mm}$ or $\sigma_\mathrm{R} = 0\,^\circ$ and $\sigma_\mathrm{t} = 4\,\mathrm{mm}$ will reduce the accuracy of pose prediction to case II, the unedited images with markers. The results with synthetic markers are better if training data is not correctly labeled.

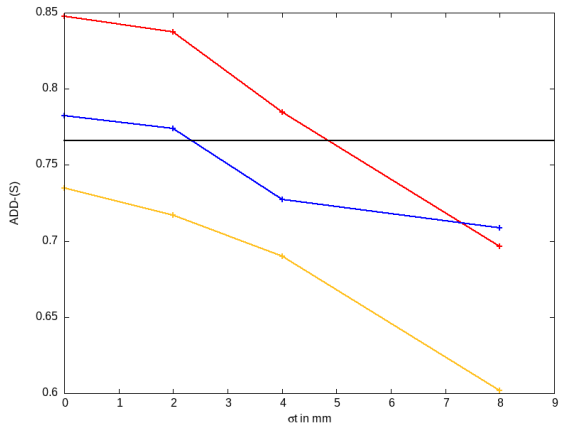

Figure 6. ADD-(S) dependent on σ_t for $\sigma_R = 0\,^\circ$ (—), $\sigma_R = 2.5\,^\circ$ (—), and $\sigma_R = 5.0\,^\circ$ (—). The horizontal black line represents the result for marker-based training from method II.

5. Conclusion

In order to achieve maximal accuracy of pose estimation, it is important to acquire high-precision training sets w.r.t. labeling accuracy. To investigate the magnitude of this interdependence, we explored a novel approach using marker-based optical tracking to generate training images with high precision. We designed our experimental setup with great care, using the same images to maintain consistency across all conditions. This approach ensures that any observed differences in prediction performance are likely attributable to the variables we were manipulating, rather than extraneous factors.

We investigated the effects of training images with and without markers on the accuracy of pose estimation. We analyzed three methods to handle images with markers during training. Using images with unedited markers as training data reduced the accuracy of predictions by 8 percent points w.r.t. ADD-(S) compared to the baseline, where markerless images were used for training. Our marker augmentation method was able to reduce learning of marker-specific patterns by randomly generating markers on the object. This increased accuracy by 3 percent points w.r.t. ADD-(S) compared to unedited images with markers. Our third method, the removal of markers in images using inpainting, achieved the best results. It has the same accuracy of 6D pose prediction as a network trained on markerless images. Additionally, we verified that our synthetic results are also valid for real training images with markers.

Furthermore, we investigated the influence of inaccuracies of the 6D pose labeling on the accuracy of pose prediction. Our findings show that, even for small labeling errors, the precise ground truth of marker-based training data results in better accuracy of pose predictions if inpainting is used to remove the markers from the images. Hence, marker-based training data is superior to a markerless dataset, if pose labeling is not done very carefully.

Our results show that, marker-based training data enables the capturing of large quantities of highly precise training data across different domains, while still yielding very good prediction accuracies. This approach notably benefits pose estimation applications that currently suffer from a lack of labeled data. High precision ground truth data obtained by marker-based methods lead to more accurate 6D pose estimation. Future work should investigate if 6D pose estimation is also more precise for real marker-based training data or if influences such as imprecise marker-object registration changes pose estimation.

Acknowledgements

The research in this paper was supported by the U Bremen Research Alliance/AI Center for Health Care, which is financially supported by the Federal State of Bremen.

References

[1] Pedro Castro and Tae-Kyun Kim. Crt-6d: Fast 6d object pose estimation with cascaded refinement transformers. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5735–5744, 2023.

[2] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. Finexus: Tracking precise motions of multiple fingertips using magnetic sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1504–1514, 2016.

[3] Alvaro Collet, Manuel Martinez, and Siddhartha S Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *International Journal of Robotics Research*, 30(10):1284–1306, 2011.

[4] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023.

[5] Bertram Drost, Markus Ulrich, Paul Bergmann, Philipp Hartinger, and Carsten Steger. Introducing MVTec ITODD — a dataset for 3d object recognition in industry. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2200–2208, 2017.

[6] Robert Elfring, Matías de la Fuente, and Klaus Radermacher. Assessment of optical localizer accuracy for computer aided surgery systems. *Comput Aided Surg*, 15(1):1–12, 2010.

[7] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with RGB-d videos and 3d hand pose annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 409–419, 2018.

[8] Anas Gouda, Abraham Ghanem, and Christopher Reining. DoPose-6d dataset for object segmentation and 6d pose estimation. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 477–483, 2022.

[9] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3d annotation of hand and object poses. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3193–3203, 2020.

[10] Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D. Twigg, and Kenrick Kin. Online optical marker-based hand tracking with deep labels. *ACM Trans. Graph.*, 37(4):1–10, 2018.

[11] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. PVN3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11629–11638, 2020.

[12] Jonas Hein, Matthias Seibold, Federica Bogo, Mazda Farshad, Marc Pollefeys, Philipp Fürnstahl, and Nassir Navab. Towards markerless surgical tool and hand pose estimation. *Int J CARS*, 16(5):799–808, 2021.

[13] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. *Proc. Asian Conf. Computer Vision*, 7724, 2012.

[14] Tomáš Hodaň, Pavel Haluza, Štepán Obdržálek, Jirí Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D Pose Estimation of Texture-Less Objects. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 880–888, 2017.

[15] Tomáš Hodaň, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiří Matas, and Carsten Rother. BOP: Benchmark for 6d object pose estimation. In *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pages 19–35. Springer International Publishing, 2018.

[16] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6d: Making RGB-based 3d detection and 6d pose estimation great again. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1530–1538, 2017.

[17] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 574–591. Springer International Publishing, 2020.

[18] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6d pose estimation. *Int J Comput Vis*, 128(3):657–678, 2020.

[19] Florentin Liebmann, Simon Roner, Marco von Atzigen, Davide Scaramuzza, Reto Sutter, Jess Snedeker, Mazda Farshad, and Philipp Fürnstahl. Pedicle screw navigation using surface digitization on the microsoft HoloLens. *Int J Comput Assist Radiol Surg*, 14(7):1157–1165, 2019.

[20] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: A hands-on survey. *IEEE Trans. Visual. Comput. Graphics*, 22(12):2633–2651, 2016.

[21] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise voting network for 6dof pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4556–4565, 2019.

[22] Long Qian, Anton Deguet, and Peter Kazanzides. ARssist: augmented reality on a head-mounted display for the first assistant in robotic surgery. *Healthc Technol Lett*, 5(5):194–200, 2018.

[23] Yongzhi Su, Jason Rambach, Nareg Minaskan, Paul Lesur, Alain Pagani, and Didier Stricker. Deep multi-state object pose estimation for augmented reality assembly. In *IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 222–227, 2019.

[24] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. ZebraPose: Coarse to fine surface encoding for 6dof object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6728–6738.

[25] Jiaming Sun, Zihao Wang, Siyu Zhang, Xingyi He, Hongcheng Zhao, Guofeng Zhang, and Xiaowei Zhou.

OnePose: One-shot object pose estimation without CAD models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6815–6824, 2022.

[26] Martin Sundermeyer, Tomáš Hodaň, Yann Labbé, Gu Wang, Eric Brachmann, Bertram Drost, Carsten Rother, and Jiří Matas. Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2784–2793, 2023.

[27] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3172–3182, 2022.

[28] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, 2017.

[29] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1082–10828, 2018.

[30] Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[31] Gu Wang, Fabian Manhardt, Xingyu Liu, Xiangyang Ji, and Federico Tombari. Occlusion-aware self-supervised monocular 6d object pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[32] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16606–16616, 2021.

[33] Jay M. Wong, Vincent Kee, Tiffany Le, Syler Wagner, Gian-Luca Mariottini, Abraham Schneider, Lei Hamilton, Rahul Chipalkatty, Mitchell Hebert, David M. S. Johnson, Jimmy Wu, Bolei Zhou, and Antonio Torralba. SegICP: Integrated deep semantic segmentation and pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5784–5789, 2017.

[34] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Proceedings of Robotics: Science and Systems*, 2018.

[35] Y. Xu, K. Lin, G. Zhang, X. Wang, and H. Li. RNNpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14860–14870, 2022.

[36] Menglong Zhu, Konstantinos G. Derpanis, Yinfei Yang, Samarth Brahmbhatt, Mabel Zhang, Cody Phillips, Matthieu Lecce, and Kostas Daniilidis. Single image 3d object detection and pose estimation for grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3936–3943, 2014.