# HomebrewedDB: RGB-D Dataset for 6D Pose Estimation of 3D Objects

Roman Kaskman[✣,†], Sergey Zakharov[✣,†], Ivan Shugurov[✣,†], and Slobodan Ilic[✣,†]

[✣] Technical University of Munich, Germany [†] Siemens Corporate Technology, Germany

{roman.kaskman, sergey.zakharov, ivan.shugurov}@tum.de, slobodan.ilic@siemens.com

## Abstract

*Among the most important prerequisites for creating and evaluating 6D object pose detectors are datasets with labeled 6D poses. With the advent of deep learning, demand for such datasets is growing continuously. Despite the fact that some of exist, they are scarce and typically have restricted setups, such as a single object per sequence, or they focus on specific object types, such as textureless industrial parts. Besides, two significant components are often ignored: training using only available 3D models instead of real data and scalability, i.e. training one method to detect all objects rather than training one detector per object. Other challenges, such as occlusions, changing light conditions and changes in object appearance, as well precisely defined benchmarks are either not present or are scattered among different datasets.*

*In this paper we present a dataset for 6D pose estimation that covers the above-mentioned challenges, mainly targeting training from 3D models (both textured and textureless), scalability, occlusions, and changes in light conditions and object appearance. The dataset features 33 objects (17 toy, 8 household and 8 industry-relevant objects) over 13 scenes of various difficulty. We also present a set of benchmarks to test various desired detector properties, particularly focusing on scalability with respect to the number of objects and resistance to changing light conditions, occlusions and clutter. We also set a baseline for the presented benchmarks using a state-of-the-art DPOD detector. Considering the difficulty of making such datasets, we plan to release the code allowing other researchers to extend this dataset or make their own datasets in the future.*

## 1. Introduction

Detection of 3D objects in images and recovery of their 6D poses is crucial for a wide range of computer vision tasks. In robotics it is essential to determine 6DoF of the object for the tasks of object grasping, manipulation and automatic assembly. Also, precise 6D poses in RGB images are of utmost importance in augmented reality (AR) applications, where overlaying 3D models on top of the real objects is critical for AR-driven assembly or repair tasks. With the rise of deep learning methods, demonstrating better performance than traditional template matching approaches or methods based on handcrafted features, it is essential to have appropriate datasets, which would encourage development and thorough evaluation of the new approaches.

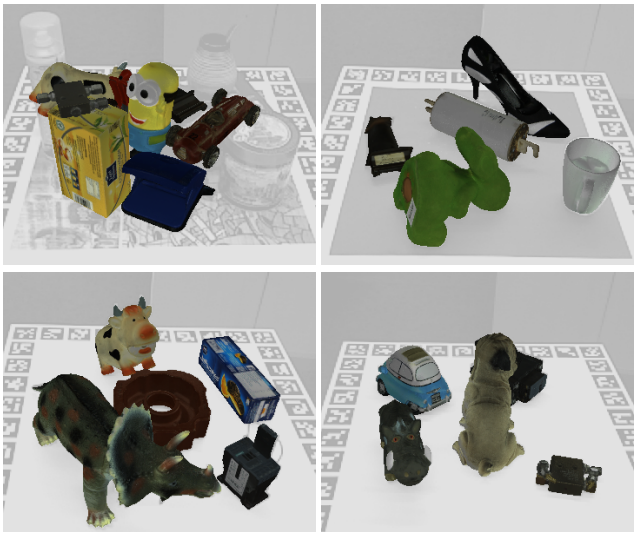

Figure 1: **HomebrewedDB scene examples:** Our dataset features 13 RGB-D annotated scenes of various difficulty. The reconstructed 3D models of the objects are rendered on top of RGB images with obtained ground truth poses.

A number of 6D pose object datasets exist, each focusing on one of the aspects of this challenging task. For example, datasets like T-LESS [18] and LineMOD [17] cover textureless and target-specific object types in particular scenarios. LineMOD contains mainly toy and household objects of distinct geometry, concentrating on object detection in cluttered environments with nonexistent or minor occlu-

sions. It includes a total of 15 separate image sequences, however each of them features 6D pose annotations only for a single object. T-LESS is a much bigger dataset, focusing exclusively on textureless industrial objects, which exhibit strong inter-object similarities and symmetries. In T-LESS, separate training and test images are explicitly provided, while it remains unclear how to sample or produce training data from LineMOD. This ambiguity has led to wide inconsistencies in presented results, making it hard to conduct a comprehensive assessment of the developed methods. Striving for the best possible results, researchers tend to train their detectors on subsets of provided real images with ground truth poses very similar to those in a test split. This strategy inevitably leads to overfitting to a particular dataset, restricts applicability of the detectors and undermines fair comparison of various approaches.

One crucial aspect that has often been neglected by 6D pose detectors is scalability. The majority of the datasets contain a rather small number of objects. This is natural since producing large number of 3D models and scenes with annotated poses is a tedious task, especially when poses of all the objects in all frames must be available. When it comes to deep learning methods, training detectors on real data yields the best results. However, the fact that 3D models of the objects are available and training data can be synthesized by rendering them has been used in only a few studies, most notably using such detectors as SSD6D [20], AAE [29] and DPOD [34]. It is remarkable that all deep learning 6DoF object detectors trained either on real or synthetic data use a single neural network per object, in contrast to 2D object detectors, such as YOLO [24], SSD [21] or R-CNNs [14, 13, 25, 15], which use one network for all object classes. A major reason for this issue is the unavailability of a proper dataset with a variety of sequences and well-defined benchmarks, causing a fallback to a well-studied simplistic dataset such as LineMOD. Nevertheless, training one network per object defeats the scalability aspect that is naturally a characteristic of deep neural networks. Therefore, central aspects of our proposed dataset is its ability to test methods' scalability by introducing corresponding benchmarks and to encourage training of detectors on renderings of 3D models instead of using real data.

Our dataset may be considered to be aligned with the OCCLUSION dataset [4], which contains poses of all the objects present in each frame. However, instead of just a single sequence with 1214 frames, our dataset contains 13 full-circle scenes filmed with both PrimeSense Carmine 1.09 (structured light) and Microsoft Kinect 2 (time-of-light) RGB-D cameras resulting in a total of 34,830 fully annotated frames with poses for all objects in all frames. The complexity of the scenes increases from simple (several separated objects per scene) to heavily-cluttered and occluded (objects close together or on top of each other and also mixed with other objects not present as 3D models). Another aspect that is not addressed in the other datasets is strong variation of the illumination, including not only changes in light intensity, but also in light color. Finally, driven by the fact that in industry objects can undergo severe appearance changes, we created a benchmark where the object appearance is altered compared to the one in available 3D models.

In the following sections, besides describing the steps of the dataset creation pipeline, we also present detection and 6D pose estimation results for all the newly-defined benchmarks of one of a recently introduced methods - Dense Pose Object Detector (DPOD) [34]. This method is trained on all the objects at once or on all the objects present in the test scene on strictly synthetic renderings of provided 3D models. The aim of our experiment was not only to test scalability of the current methods but also to demonstrate that even the best-performing methods trained on real images with one network per object obtain mediocre results when trained on multiple objects. This establishes a baseline and opens a new challenge to the community related mainly to scalability and synthetic training data. Light and appearance changes are another new aspect of this challenging dataset.

## 2. Related Datasets

Given the fact that determining ground truth 6D poses from RGB images is an ambiguous task requiring manual interventions, it is not surprising that the majority of 6D pose datasets are made with the use of RGB-D cameras. Because these cameras provide depth images aligned with color images, the task of 6D pose estimation becomes simpler and more automated. The most popular datasets for 6D pose estimation are discussed in this section.

**LineMOD Dataset.** One of the most widely used 6D pose datasets is LineMOD by Hinterstoisser *et al*. [17]. Essentially, it contains objects embedded in cluttered scenes. It was acquired with PrimeSense Carmine RGB-D sensor and in total comprises 15 objects, two of them being symmetric. For each sequence, poses of only one object are annotated. The target objects with the existing ground truth poses are either not occluded, or are subject to very slight occlusions. The original template matching method [17], which was published alongside the dataset, performed poorly on RGB and depth images separately, while the results obtained on RGB-D images were extremely good and still remain hard to achieve by many modern deep-learning-based methods. However, this method suffers from a couple of drawbacks: it scales poorly and cannot handle occlusions. OCCLUSION dataset was created by Brachmann *et al*. [4] in order to address the lack of occluded test data. In this extension of the original LineMOD additional manual annotations of 6D

poses for all the objects in each frame were included. Due to the necessity of intensive manual labor, it was done for only a limited number of frames. Availability of RGB and depth images as well as 3D CAD models with original object colors resulted in probably the widest use of this dataset of all those targeting 6D pose estimation. Deep learning methods use RGB images from this dataset for object detection and 6D pose estimation. State-of-the-art results for real and synthetic data were achieved by the recently introduced DPOD method [34].

**T-LESS Dataset.** T-LESS [18] is a recent dataset that is gaining popularity. It contains 30 textureless industrial objects and 20 RGB-D scenes captured with three synchronized cameras (PrimeSense Carmine 1.09 and Kinect 2 RGB-D cameras and Canon RGB camera). The objects in this dataset have strong inter-object similarity. The acquired scenes vary from simple (less clutter and several objects per scene) to complex (heavily cluttered, with piles of objects, mimicking a typical robotic bin-picking scenario). Both hand-designed CAD models and reconstructed 3D models are included in T-LESS. Training images contain isolated objects on black backgrounds, while test images capture entire scenes with labeled 6D poses for each object in each frame. The structure of this dataset is exceptional, but due to symmetry, low texture and the industrial nature of the objects it is very challenging, which might be the reason why it has not gained in popularity at the pace it deserves. Truly inspired by T-LESS [18], we prepared our dataset similar to this one in terms of structure. However, our dataset covers a wider range of object types, spanning toys, household objects as well as industrial objects. Also, we aimed to build a dataset with occlusion challenges, and that goes well beyond OCCLUSION, thereby providing many more frames and scenes for this task.

**YCB-Video Dataset.** This dataset is very recent and came out with the PoseCNN detector [32]. Unlike T-LESS, which contains images of the scenes from all viewpoints, this one resembles LineMOD, containing short videos depicting several household objects in the scene. However, the large number of video sequences (92) as well as the presence of occlusions in the test scenes make this dataset quite attractive.

**Other Datasets.** The following datasets have been sporadically used in some publications, but their systematic use has not been achieved. Tejani *et al*. [30] contains 2 textureless and 4 textured objects with 700 frames of test images. A characteristic of this dataset is that it includes multiple instances of the same object. Clutter and occlusions are moderate, which makes it not particularly challenging. Doumanoglou *et al*. [9] presented a bin-picking dataset with 183 test images and only two objects from the Tejani dataset, but multiple instances of them. The Challenge and Willow datasets [33] contain a larger number of objects (176), but a relatively small number of test images (353). The TUW dataset [2] is similar with 17 objects in 224 test images. Datasets that are suited for robotic tasks are the Rutgers [26], the Amazon Picking Challenge [7], and the BIGBird [27] datasets. Datasets addressing the light change challenge are TUD-Light and Toyota Light, both used and referenced in the benchmark paper by Hodan *et al*. [19].

Our dataset could be considered to be between OCCLUSION and T-LESS. It contains high-quality annotations, a multitude of objects and a large number of scenes and test images. In contrast to the recent work of Hodan *et al*. [19], who have tried to unify 8 different datasets and have concentrated on evaluation of RGB-D methods, we are more interested in RGB methods, even though full RGB-D images are available in our dataset. With scalability at its core, we design benchmarks in which one network is trained either for all the available objects or only for the objects present in a particular scene. Additionally, we introduce scenes with severe environment changes, including changes of light colors and intensities as well as changes of object appearance. We believe that this dataset will push forward research on scalable object detection and domain adaptation.

## 3. HomebrewedDB Dataset Creation

Within our HomebrewedDB dataset we introduce the following:

1. **33 highly accurately reconstructed 3D models** of toys, household objects and low-textured industrial objects of sizes varying from 10.1 to 47.7 cm in diameter.
2. **13 sequences**, each containing 1340 frames filmed using 2 different RGB-D sensors. Scenes span a range of complexity from simple (3 objects on a plain background) to complex (highly occluded with 8 objects and extensive clutter) (Fig. 3). Also, two sequences feature drastic light changes or contain objects with altered textures (Fig. 4).
3. **Precise 6D pose annotations** for dataset objects in the scenes, which were obtained using an automated pipeline (see Section 3.5).
4. **A set of benchmarks** to facilitate comprehensive evaluation of object detection and 6D pose estimation methods.

The following sections detail the dataset creation, including calibration of RGB-D sensors, reconstruction of 3D models, depth correction, acquisition of image sequences and creation of ground truth annotations. We believe that the described steps would serve as a sufficient guide on how to extend an existing or create a new dataset for 6D pose estimation.


Figure 2: Rendered reconstructed 3D models of HomebrewedDB.

### 3.1. Calibration of RGB-D Sensors

For the footage of validation and test sequences we used two RGB-D sensors: the structured-light PrimeSense Carmine 1.09 and the time-of-flight Microsoft Kinect 2. Intrinsic and distortion parameters of both sensors were estimated during the calibration procedure. We used ArUco board [12], which has yielded better calibration results in comparison with the classical checkerboard pattern and the corresponding intrinsic calibration module in OpenCV [5]. As a result of calibration, the root-mean squared reprojection error calculated at the corners of the ArUco markers is $\leq 0.5$ and $\leq 0.3$ px for Carmine and Kinect 2 respectively. Intrinsic and distortion parameters for both sensors are provided with the dataset. Because depth and color images were obtained from two different cameras, depth-to-color registration was performed using OpenNI 2.2 Driver for Carmine and Windows SDK 2.0 for Kinect 2. Given that the scenes were recorded independently with each of the sensors, there was no need in extrinsic calibration of the cameras.

### 3.2. Sequence Acquisition

In total we acquired 1 handheld and 2 turntable sequences for each of the scenes with each RGB-D sensor. Turntable sequences capture a full 360°rotation of a markerboard with objects on it using a camera mounted on a tripod. Each turntable sequence has 170 RGB and depth images filmed with elevation angles of 30°and 45°. Together with the ground truth 6D pose labels they form a validation dataset. In contrast, the test sequences were recorded in handheld mode. There were two major reasons for the handheld recording instead of using a controlled setup similar to those in T-LESS [18] or BigBIRD [27]. The first clear advantage is the close resemblance to regular camera use, while the second one is the ability to introduce more variation to camera poses in terms of considerable scale changes as well as in-plane rotation. For the test sequences, a total of 1000 RGB and corresponding depth images of each scene were captured with each sensor. While shooting each of those, a full pass around the markerboard was made. Within the test sequences, the distance from the camera to an object was varied from 0.42 to 1.43 m, while the elevations were within 11°-87°. Both color and depth images in the sequences that were recorded with the Carmine sensor have a default resolution of 640×480 px, while Kinect 2 RGB images are of size 1920×1080 px and depth images of 512×424 px. The depth images were resized to the dimension of RGB images during registration.

For each of the scenes (Figs. 3 and 4) target objects were placed on the markerboard with ArUco markers facilitating camera pose estimation. For the simpler scenarios, we placed the objects on a monochrome (white or dark-blue) Lambertian surface, and for the more complicated ones the objects resided on a multicolor reflective surface, being in some cases on top of each other. Also, more complicated sequences featured severe occlusions as well as objects not present in the dataset to make the scene more cluttered.

One new feature introduced in HomebrewedDB is a sequence aiming to test the robustness of detection and pose estimation methods with respect to significant changes in lighting conditions. To create it, we used a spotlight to repeatedly project series of light patterns with different colors and intensities onto the scene. We also created a domain adaptation sequence to evaluate robustness to considerable texture changes. For that we altered the textures of the objects by selectively painting some part of them with chalks of various colors.

### 3.3. 3D Model Reconstruction

To obtain the 3D models presented in HomebrewedDB (Fig. 2), we first scanned each object from multiple viewpoints using Artec Eva [1], a structured light 3D scanner.

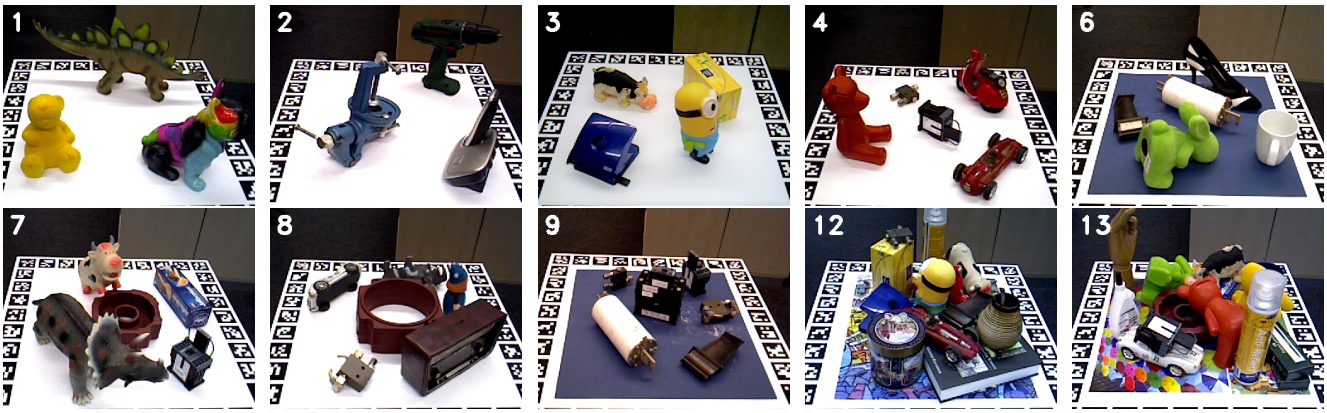


Figure 3: Sample RGB images from the sequences presented in the HomebrewedDB dataset. Complexity of the scenes varies in terms of number and size of objects, levels of occlusion and clutter.


Figure 4: Sample RGB images from sequences belonging to the domain adaptation benchmark.

We opted for Artec Eva since it provides precise depth measurements and high resolution texture maps, which are crucial for reconstructing high-quality 3D models. The following pipeline for converting the scans to completely reconstructed 3D models proceeded using Artec Studio software. First, raw meshes were reconstructed for each of the scanned viewpoints. Secondly, we manually removed unnecessary parts of the meshes and then aligned them. Then we removed outliers and minor artifacts from the models. After that we proceeded with global optimization of the mesh structure, including inpainting minor holes in the model and as inducing smoothness of the mesh. Next, we back-projected high-resolution textures onto the resulting 3D model. Finally, we used MeshLab [6] to center and and axis-align the models and computed surface normals as weighted sum of normals of the incident facets [23].

### 3.4. Depth Correction

As have others [28, 18], we observed that depth measurements by both Carmine and Kinect 2 have systematic errors: we found that the measured depth values were always slightly different from those calculated from the markers in images captured with calibrated RGB cameras. Although it has been reported [28] that a single correction multiplier is sufficient to address the depth measurement error, we confirmed in our setup what Hodan *et al*. [18] had found - that first degree polynomial works better as a correction factor for the depth measurements in our setup. Using regularized least squares to account for noise in the measurements we derived the following linear depth-correction models: $d_c = 1.0391 \cdot d - 15.8$ for Carmine and $d_c = 1.0186 \cdot d - 13.1$ for Kinect 2 (measured in millimeters). After applying the corrections, we found that the mean absolute difference from expected depth had been reduced from 14.7 mm to 2.03 mm and from 5.81 mm to 2.66 mm for Carmine and Kinect 2 respectively. We applied the corrections models to the entire dataset, so that no further user action is required.

### 3.5. Creation of Ground Truth Annotations

The estimation of 6D ground truth objects poses for each frame in the sequence proceeded as follows. First, the markerboard pose was estimated, providing us with the camera trajectory around the scene. Then we obtained a dense 3D reconstruction by signed distance field fusion of depth maps of a scene with a method of Curless and

Levoy [8], using all the images in a sequence.

The next step was to estimate a rigid body transformation of a 3D model from its own coordinate system to the coordinate system of the scene (i.e., markerboard). Given that the locations of the objects with respect to the markerboard did not change when imaged with different sensors, it was sufficient to get the object pose in the markerboard coordinate system and then transfer it to a new sensor or camera coordinate system, thereby avoiding performing reconstruction for each new sensor. For the purpose of estimating 3D model poses in the reconstructed scene we used a method by Drost *et al*. [10], which is based on point-pair feature representation of a target model used for local matching via an efficient voting scheme on a reduced 2D search space. Having a camera pose in each image estimated from the markerboard as well as having object poses in the markerboard coordinate system, 6D object poses for each of the frames can easily be computed. However, rendering the 3D models on top of RGB images revealed that even though the method by Drost *et al*. [10] gives a good initial estimate of a pose, in many cases there remain visible discrepancies, which must be mitigated in the process of further refinement.

To improve the poses, we opted for 2D edge-based ICP [11] refinement. For each object in the sequence we automatically selected RGB images where the object was not occluded. This was done by rendering all the objects in the scene with the estimated initial poses and calculating the fraction of visible pixels for the target object. Multiview consistency was enforced such that all the camera poses stayed fixed, and optimization was only done for the object pose in the scene coordinate system. Edge-based refinement was performed on RGB images because depth measurements were relatively noisy and we observed minor misregistrations between depth and RGB images, particularly for those captured with Kinect 2.

### 3.6. Accuracy of Ground Truth Poses

To evaluate the accuracy of the computed ground truth poses, we followed the same procedure as introduced by Hodan *et al*. [18]. We rendered the 3D objects using computed ground truth poses and for each pixel pair with valid depths in both rendered and captured images we computed the difference $\delta = d_c - d_r$, where $d_c$ and $d_r$ are captured and rendered depth values, respectively. The statistics obtained over the whole test set are presented in Tab. 1. As in T-LESS [18], differences exceeding 5 cm were omitted from the statistics as outliers. In HomebrewedDB, such measurements amounted to $3.7\%$, mostly caused by the clutter objects occluding the target objects, sensor measurement noise or minor discrepancies between reconstructed 3D models and real-world objects.

From the presented results it can be seen that rendered depth maps align well with the depth maps obtained with Carmine, resulting in mean depth difference close to zero and absolute mean of differences $\leq$ 2 mm. For Kinect 2 depth differences mean and absolute mean values also stay very close to zero: the absolute median value is notably higher than the absolute mean, signifying that the distribution of the depth differences is left-skewed. As noted in T-LESS [18], this might be caused by slight misregistration of RGB and depth images captured by Kinect 2, as well as higher magnitude of noise in the measurements of this depth sensor based on the time-of-flight principle.

| Sensor | $\mu_\delta$ | $\sigma_\delta$ | $\mu_{\lvert\delta\rvert}$ | $med_{\lvert\delta\rvert}$ |
|---|---|---|---|---|
| Carmine | 0.11 | 6.25 | 1.71 | 2.56 |
| Kinect 2 | 0.22 | 7.38 | 0.87 | 9.12 |

Table 1: Differences between the depth of object rendered models at the ground truth poses and the captured depth (in mm). $\mu_\delta$ and $\sigma_\delta$ is the mean and the standard deviation of the differences, $\mu_{|\delta|}$ and $med_{|\delta|}$ is the mean and the median of the absolute differences.

## 4. Benchmarks and Experiments

In this section, we present a set of benchmarks assessing the performance of a detector with respect to a variety of different conditions. In particular, such aspects as scalability and resistance to occlusions, different illumination conditions and changes in object texture are tested.

### 4.1. Evaluation Metrics

We use standard metrics for evaluating performance in 2D object detection: precision, recall and mean average precision (mAP). Conventionally, we consider an object to be correctly detected if the intersection over union (IoU) between the ground-truth and predicted bounding boxes is $\geq 0.5$. To evaluate the correctness of the estimated 6D poses, as others have done [16, 20, 34], we use the ADD score, which is defined as the average Euclidean distance between the model vertices transformed with ground truth and predicted poses:

$$m = \underset{\mathbf{x} \in \mathcal{M}}{\text{avg}} \left\| (\mathbf{R}\mathbf{x} + \mathbf{t}) - (\hat{\mathbf{R}}\mathbf{x} + \hat{\mathbf{t}}) \right\|_2, \qquad (1)$$

where $\mathcal{M}$ is a set of vertices of a 3D model, $(\mathbf{R}, \mathbf{t})$ and $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ are ground truth and predicted rotation and translation, respectively. As mentioned in [16], a predicted pose is considered to be correct if ADD calculated with this pose is less than 10% of a model diameter. However, in case of more complicated scenes, there is only a small fraction of poses falling into this category. Therefore, we also report ADD for the thresholds of 30% and 50% to give a broader overview of pose quality as well as to give an estimate of

| | | Per Scene | | | | | | | | | | | Texture / Illumination | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Scene ID** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 12 | 13 | 5 | 10 | 11 |
| **Pose** | **ADD 10%** | 50.85 | 45.08 | 33.88 | 27.25 | 25.40 | 19.19 | 25.85 | 11.08 | 7.60 | 8.68 | 13.36 | 25.40 | 16.73 | 16.77 |
| | **ADD 30%** | 81.34 | 78.46 | 64.53 | 53.10 | 54.00 | 45.39 | 39.99 | 29.78 | 20.12 | 25.68 | 33.41 | 54.00 | 40.32 | 40.83 |
| | **ADD 50%** | 88.71 | 85.69 | 75.46 | 66.98 | 61.95 | 55.26 | 46.76 | 39.64 | 29.24 | 34.49 | 45.33 | 61.95 | 51.66 | 49.72 |
| **Detect.** | **Precision** | 0.79 | 0.62 | 0.76 | 0.76 | 0.65 | 0.46 | 0.68 | 0.51 | 0.26 | 0.33 | 0.13 | 0.65 | 0.57 | 0.43 |
| | **Recall** | 0.95 | 0.64 | 0.82 | 0.92 | 0.80 | 0.68 | 0.84 | 0.62 | 0.33 | 0.40 | 0.20 | 0.80 | 0.63 | 0.53 |
| | **mAP** | 0.82 | 0.48 | 0.72 | 0.78 | 0.64 | 0.36 | 0.64 | 0.41 | 0.14 | 0.20 | 0.04 | 0.64 | 0.42 | 0.32 |

Table 2: Result of object detection and pose estimation presented on two benchmarks: (1) per scene benchmark spanning over 11 scenes of the dataset and the (2) domain adaptation benchmark evaluating the detector's generalization capabilities.

proportion of the poses which could be still a subject for further refinement.

### 4.2. Dense Object Pose Detector

For the performance evaluation we opted for the recently introduced Dense Pose Object Detector (DPOD) [34], owing to its excellent performance on LineMOD and OCCLUSION datasets and its ability to be trained on both real and synthetic data. Other detectors also could be used, and we highly encourage prospective researchers to perform evaluation on HomebrewedDB in order to facilitate progress in the direction of scalability and training from CAD models.

The DPOD detector is based on DensePose [3] applied to human pose detection and formalizes the object detection problem as a dense correspondence estimation problem. From the input image DPOD outputs the multi-labeled object segmentation mask and UV correspondence map. Obtained correspondences are further used as input to PnP and RANSAC for pose estimation. The more correspondences obtained, the less prone the detector is to wrong matches, and the easier and more accurate pose estimation becomes.

### 4.3. Scalability Benchmark

The first benchmark was introduced with a goal to evaluate the method's scalability with respect to the number of objects. The main requirement is to train a single network for all the objects available in the dataset and test it on a set of sequences containing all the objects. Specifically, for this purpose we jointly evaluate a method on the sequences from 1 to 8 (inclusive), which form a minimal subset of sequences with all 33 objects present. For this benchmark object detection and pose estimation results are reported separately for each object.

From the results presented in Tab. 3 it can be seen that the best detection and pose estimation performance is achieved for bigger objects with distinct textures and geometric features (e.g., 28, 30), while detection of smaller low-textured or glossy industrial objects (e.g., 12, 13, 14) is a considerable challenge for DPOD. Besides, pose estimation results demonstrate that the DPOD detector does not scale particularly well for this task: for 17 out of 33 objects, ADD with 10% threshold is under 10%, and there are no instances for which the achieved score was higher than 50%.

### 4.4. Scene Benchmarks

Our dataset presents a collection of 13 scenes with a varying degree of difficulty, and each of these scenes represents a separate benchmark, where all the objects it contains are used for training. Except for the scenes with altered illumination conditions or texture changes (i.e., 10 and 11), we include all the scenes in the per-scene benchmarks. All the object detection and pose estimation scores are averaged over all the objects present in the scene. In Tab. 2 the objects detection and pose estimation performance per scene is presented. As expected, DPOD demonstrates significantly better performance if both tasks in the sequences with the smaller number of objects, as well as less significant occlusions and no clutter. Also, low scores in both detection and pose estimation are reported for scene 8, which is composed exclusively of industrial objects.

### 4.5. Domain Adaptation Benchmark

The main goal of introducing the domain adaptation benchmark is to test the robustness of a method to significant changes in lighting conditions and objects textures. It is composed of three scenes (5, 10 and 11) with the same set of objects, but differing in terms of illumination and the color of object surfaces. Scene 5 was captured in ambient lighting conditions with no alteration of the objects' appearance. In contrast, in scene 10 we used a spotlight to project light of different colors and intensity onto the imaged objects to introduce considerable variations in illumination, whereas in scene 11, we applied paint on the objects' surfaces to alter their texture. For this benchmark, object detection and 6D pose estimation scores are presented per scene.

As can be seen from results in Tab. 2, performance in both detection and pose estimation is notably better in the cases of no added illumination or texture changes. Under normal conditions the resulting poses are 37% more accurate based on ADD with a 10% threshold when compared to those under altered conditions. Also, object detection performance falls far behind under altered conditions compared to normal conditions, resulting in 46% and 59% lower

| Obj. ID | ADD 10% | ADD 30% | ADD 50% | Precision | Recall | mAP |
|---|---|---|---|---|---|---|
| 1 | 23.04 | 68.30 | 81.04 | 0.86 | 0.98 | 0.85 |
| 2 | 17.85 | c 58.02 | 75.87 | 0.62 | 0.89 | 0.56 |
| 3 | 14.18 | 55.41 | 72.40 | 0.93 | 0.92 | 0.88 |
| 4 | 9.09 | 36.74 | 55.43 | 0.66 | 0.79 | 0.53 |
| 5 | 11.70 | 43.23 | 58.60 | 0.93 | 0.98 | 0.91 |
| 6 | 0.00 | 3.51 | 12.13 | 0.67 | 0.68 | 0.46 |
| 7 | 15.15 | 44.12 | 60.74 | 0.66 | 0.68 | 0.45 |
| 8 | 18.67 | 42.74 | 52.28 | 0.33 | 0.24 | 0.08 |
| 9 | 2.26 | 13.71 | 27.58 | 0.60 | 0.62 | 0.37 |
| 10 | 3.15 | 19.63 | 30.96 | 0.89 | 0.86 | 0.76 |
| 11 | 0.71 | 4.07 | 9.87 | 0.82 | 0.98 | 0.80 |
| 12 | 0.50 | 3.74 | 7.48 | 0.47 | 0.40 | 0.19 |
| 13 | 2.25 | 20.22 | 35.63 | 0.58 | 0.62 | 0.36 |
| 14 | 2.88 | 24.04 | 40.19 | 0.41 | 0.52 | 0.21 |
| 15 | 0.00 | 1.88 | 5.52 | 0.33 | 0.40 | 0.14 |
| 16 | 0.00 | 4.70 | 9.94 | 0.48 | 0.36 | 0.17 |
| 17 | 0.72 | 3.37 | 17.35 | 0.63 | 0.42 | 0.26 |
| 18 | 0.16 | 2.24 | 4.32 | 0.49 | 0.63 | 0.32 |
| 19 | 7.70 | 28.63 | 45.88 | 0.94 | 0.92 | 0.89 |
| 20 | 23.22 | 63.60 | 75.73 | 0.92 | 0.96 | 0.91 |
| 21 | 8.37 | 36.12 | 50.66 | 0.39 | 0.23 | 0.09 |
| 22 | 10.04 | 35.97 | 55.02 | 0.76 | 0.78 | 0.59 |
| 23 | 16.18 | 62.49 | 82.20 | 0.95 | 0.99 | 0.95 |
| 24 | 4.08 | 29.25 | 55.78 | 0.19 | 0.15 | 0.03 |
| 25 | 9.76 | 39.50 | 50.62 | 0.75 | 0.88 | 0.66 |
| 26 | 15.01 | 51.83 | 68.98 | 0.78 | 0.79 | 0.63 |
| 27 | 13.47 | 51.78 | 71.47 | 0.68 | 0.76 | 0.52 |
| 28 | 26.17 | 63.30 | 78.07 | 0.80 | 0.92 | 0.73 |
| 29 | 13.97 | 38.59 | 57.68 | 0.93 | 0.94 | 0.89 |
| 30 | 48.63 | 88.71 | 96.34 | 0.80 | 0.98 | 0.79 |
| 31 | 6.43 | 21.85 | 36.68 | 0.83 | 0.86 | 0.74 |
| 32 | 0.00 | 5.95 | 10.71 | 0.23 | 0.17 | 0.04 |
| 33 | 11.40 | 45.69 | 64.99 | 0.93 | 0.97 | 0.91 |

Table 3: Results of object detection and pose estimation on scalability benchmark.

mAP scores for the scenes with variations in light and texture, respectively. These results suggest that there is a lot of room for further adaptation of the DPOD detector to changing environments.

### 4.6. Drawbacks of Training on Real Data

The main reason we exclusively use synthetic data for training in our benchmarks comes from the inability of the detectors trained on a small-scale set of real data to generalize to different environments, which may differ in background, illumination, texture and other scene characteristics.

To support our claim, we selected 2 recent state of the art detectors, YOLO6D [31] and DPOD [34], trained on real LineMOD data and tested them on our new sequence containing 3 LineMOD objects - a benchvise, a drill and a phone (Scene 2 in Fig. 3). This sequence contains a minimal number of occlusions and no clutter; it may be regarded as one of the simplest scenes in the dataset. Besides, this sequence was captured with the same camera as LineMOD sequences (PrimeSense Carmine 1.09), making it similar in terms of color scheme and noise. Performance comparison of both these detectors on LineMOD sequence and our sequence can be seen in the "Real" section of Tab. 4. In spite of demonstrating very good performance when tested

| | Dataset | Method | Benchvise | Phone | Driller |
|---|---|---|---|---|---|
| Real | LM | YOLO6D [31] | 81.80 | 47.74 | 63.51 |
| | | DPOD [34] | 95.34 | 74.24 | 97.72 |
| | HB | YOLO6D [31] | 15.30 | 6.50 | 0.10 |
| | | DPOD [34] | 57.24 | 33.09 | 62.82 |
| Synthetic | LM | SSD6D + Ref. [22] | 44.30 | 26.20 | 26.90 |
| | | DPOD [34] | 66.76 | 29.08 | 66.60 |
| | HB | SSD6D + Ref. [22] | 59.40 | 29.30 | 25.10 |
| | | DPOD [34] | 70.89 | 35.56 | 66.42 |

Table 4: Pose estimation results in terms of ADD 10% metric on LineMOD sequences (LM) and HomebrewedDB (HB) sequence with the same objects.

on sequences from LineMOD, when run on our sequence with LineMOD objects, both detectors experience a significant drop in pose estimation accuracy in terms of the ADD 10% metric for all the objects. Specifically, the pose estimation accuracy of DPOD [34] dropped more than 2 times for the phone, while for YOLO6D [34] there were nearly no correctly predicted poses for the drill.

As the second part of the experiment, we evaluated 2 detectors designed for training on synthetic data, DPOD [34] and SSD6D with model-based refinement [22], on the same new sequence with LineMOD objects. From the results presented in the "Synthetic" section of Tab. 4 one can see that there is no significant difference between the results obtained on LineMOD and HomebrewedDB sequences, meaning that there is no overfitting to a particular dataset. In 5 out of 6 cases, the detectors demonstrated better performance on a simpler HomebrewedDB sequence, thus showing more predictable behavior than in the case of training on real data. Moreover, when trained on synthetic data, DPOD [34] can boast higher ADD 10% scores for all the 3 objects in the HomebrewedDB test sequence. This fact once again supports the claim that training detectors on synthetic data leads to better generalization, in contrast to training on real data, when even a seemingly insignificant changes in the environment turn out to be a decisive factor leading to a significant decrease in pose estimation accuracy.

## 5. Conclusion

In this work, we have presented a new challenging dataset for 6D object detection, covering the most important properties a solid object detector should have, namely scalability with respect to the number of objects and robustness to occlusions, illumination and appearance changes. This new dataset contains 33 objects spanning 13 scenes of various difficulty. To be able to compare the detectors to each other, we defined a set of benchmarks that test all of the above mentioned properties. Finally, we developed and presented a comparably simple, yet robust and fully automated pipeline that we used to build our dataset. We hope it will allow other researchers to be able to create their own datasets and thus promote research in 6D pose estimation.

## References


[1] Artec 3D. https://www.artec3d.com/. Accessed: 2019-03-23.

[2] Aitor Aldoma, Thomas Faulhammer, and Markus Vincze. Automation of "ground truth" annotation for multi-view RGB-D object instance recognition datasets. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 5016–5023, 2014.

[3] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018.

[4] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D Object Pose Estimation Using 3D Object Coordinates. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision ECCV 2014*, volume 8690, pages 536–551. Springer International Publishing.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.

[7] Nikolaus Correll, Kostas E. Bekris, Dmitry Berenson, Oliver Brock, Albert Causo, Kris Hauser, Kei Okada, Alberto Rodriguez, Joseph M. Romano, and Peter R. Wurman. Lessons from the amazon picking challenge. *CoRR*, abs/1601.05484, 2016.

[8] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM.

[9] Andreas Doumanoglou, Rigas Kouskouridas, Sotiris Malassiotis, and Tae-Kyun Kim. 6d object detection and next-best-view prediction in the crowd. *CoRR*, abs/1512.07506, 2015.

[10] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[11] Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):932–946, July 2002.

[12] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[13] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448.

[14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587.

[15] K. He, G. Gkioxari, P. Dollr, and R. Girshick. Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988.

[16] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 International Conference on Computer Vision*, pages 858–865. IEEE.

[17] 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. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision ACCV 2012*, volume 7724, pages 548–562. Springer Berlin Heidelberg.

[18] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[19] Tomas Hodan, 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, Jiri Matas, and Carsten Rother. BOP: benchmark for 6d object pose estimation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, pages 19–35, 2018.

[20] 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 *Proceedings of the International Conference on Computer Vision (ICCV 2017), Venice, Italy*, pages 22–29.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. 9905:21–37.

[22] Fabian Manhardt, Wadim Kehl, Nassir Navab, and Federico Tombari. Deep Model-Based 6D Pose Refinement in RGB. page 16.

[23] Nelson Max. Weights for computing vertex normals from facet normals. *J. Graph. Tools*, 4(2):1–6, Mar. 1999.

[24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc.

[26] Colin Rennie, Rahul Shome, Kostas E. Bekris, and Alberto F. De Souza. A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place. *CoRR*, abs/1509.01277, 2015.

[27] Arjun Singh, James Sha, Karthik S. Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 509–516, 2014.

[28] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[29] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D Orientation Learning for 6D Object Detection from RGB Images. pages 699–715.

[30] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3d object detection and pose estimation. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 462–477, Cham, 2014. Springer International Publishing.

[31] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction.

[32] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *CoRR*, abs/1711.00199, 2017.

[33] Ziang Xie, Arjun Singh, Justin Uang, Karthik S. Narayan, and Pieter Abbeel. Multimodal blending for high-accuracy instance recognition. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3-7, 2013*, pages 2214–2221, 2013.

[34] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: Dense 6d pose object detector in rgb images. *arXiv preprint arXiv:1902.11020*, 2019.