2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and Benchmark

Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield
{styree, jtremblay, thangt, jicheng, tmosier, jeffreys, sbirchfield}@nvidia.com
NVIDIA

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981838

***Abstract*— We present a new dataset for 6-DoF pose estimation of known objects, with a focus on robotic manipulation research. We propose a set of toy grocery objects, whose physical instantiations are readily available for purchase and are appropriately sized for robotic grasping and manipulation. We provide 3D scanned textured models of these objects, suitable for generating synthetic training data, as well as RGBD images of the objects in challenging, cluttered scenes exhibiting partial occlusion, extreme lighting variations, multiple instances per image, and a large variety of poses. Using semi-automated RGBD-to-model texture correspondences, the images are annotated with ground truth poses accurate within a few millimeters. We also propose a new pose evaluation metric called ADD-H based on the Hungarian assignment algorithm that is robust to symmetries in object geometry without requiring their explicit enumeration. We share pre-trained pose estimators for all the toy grocery objects, along with their baseline performance on both validation and test sets. We offer this dataset to the community to help connect the efforts of computer vision researchers with the needs of roboticists.**[1]

I. Introduction

Estimating the poses of objects in a scene is important for robotic grasping and manipulation in a variety of domains, such as manufacturing, healthcare, commercial, and households. In pick-and-place tasks, for example, it is necessary for the robot to know the identities and locations of objects in order to grasp them, to avoid collision with other objects during movement, and to place them relative to other objects. Yet, even today, it is not uncommon for objects in robotics labs to be modified with fiducial markers such as ARTags [1] or AprilTags [2] to support detection and pose estimation. Such modifications are needed because no widely used, off-the-shelf, general-purpose techniques currently exist.

To bridge the gap between research labs and real-world scenarios, many researchers have developed methods to detect objects and estimate their poses from RGB or RGBD images. Recent progress on this 6-DoF ("degrees of freedom") pose estimation problem has been significant [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Although there remain many open research problems, the basics are in place.

Nevertheless, from a robotics point of view, another important limitation remains: Existing pose estimators are trained for objects that are not available to most researchers. This limitation is easily observed from the comprehensive list of datasets in the BOP Benchmark for 6D Object Pose Estimation [14].[2] Because there is no easy way to acquire most of the physical objects, robotics researchers are often unable to conduct real robotic experiments leveraging pose networks already trained using these datasets.

Fig. 1: Set of 28 toy grocery objects used in the dataset.

Fig. 2: Synthetic rendering of 3D textured object models.

Of these existing datasets, arguably the most accessible are YCB [15] and RU-APC [16]. However, over time it is becoming increasingly difficult to find matching objects from these datasets in stores. For example, even though we live near the researchers who assembled the YCB dataset, we have been unsuccessful in locating a sugar box (YCB item 004) or gelatin box (YCB item 009) whose appearance matches that of the original. One of the reasons for this failure is the changing appearance of such products. In particular, food manufacturers frequently change the design on their products based on seasonal or promotional considerations or updates to their branding. As a result, networks trained on the YCB dataset exhibit degraded performance when applied to actual objects acquired from a store. A further problem is that some of the items are not of the appropriate size, shape, and weight for common robotic grippers.

To address these problems, we release a dataset and benchmark for 6-DoF pose estimation for robotic manipulation research. The dataset is designed to be:

[1] https://github.com/swtyree/hope-dataset

[2] https://bop.felk.cvut.cz/datasets/

- **Accessible.** The physical objects should ideally be accessible to anyone in the world. As a result, we selected toy grocery objects that can be purchased online. We also scanned these objects and share the resulting 3D textured models to aid in generating training images.
- **Challenging.** We captured real images of these objects with the following characteristics: 1) significant occlusions and clutter, 2) varying number of instances of each object, 3) a wide variety of poses, and 4) extreme and challenging lighting conditions.
- **Accurate.** We carefully labeled the images with ground truth poses, and errors in the labeled poses have been quantitatively assessed to be on the order of a few millimeters in world coordinates.

The dataset of test and validation images, with annotations for the latter, is available for download. Additionally, metrics against the test set annotations are computed by the BOP Benchmark evaluation server. We include initial results using both DOPE [4] and CosyPose [3], whose pre-trained network weights are released as an off-the-shelf system for use in robotics labs as well as a baseline method for further research in this area. We also provide 3D object meshes for generating synthetic data for training. We call our dataset HOPE, for Household Objects for Pose Estimation.

II. Method

In this section we describe the approach behind the dataset and benchmark.

A. Set of Objects

Because our goal is to support robotic manipulation research, it is imperative that the set of objects be: 1) somewhat realistic, 2) of the proper size and shape for grasping by a variety of robotic end effectors, and 3) accessible to researchers throughout the world. An emerging area in robotic manipulation is that of household robots for automating daily chores in a laundry room or kitchen. Such applications are crucial for facilitating aging-in-place, assisted living, and similar healthcare-related challenges. While we initially intended to scan real items from a grocery store, we quickly realized that such an approach comes with several fundamental limitations: 1) such objects are not widely available to researchers in other parts of the world, 2) the reflective metallic properties of many of these objects make for difficult scanning and rendering, and 3) the surface textures of real-world objects frequently change due to seasonal marketing campaigns. Again, it is not enough for us to provide a dataset for *perception researchers* to design and test their algorithms—we also want to support *robotics researchers* who need to leverage the latest perception research in their own labs. As a result, the trained networks from the former should be immediately usable by the latter.

To this end, we chose a set of 28 toy grocery objects, depicted in Fig. 1. These toys are widely available online and can be purchased for less than 60 USD total. Because they are not real grocery items, there are no issues with perishability or transporting of the contents. Moreover, these objects are of the proper size and shape for typical robotic grippers, with all objects having at least one dimension between 2.4 and 7.2 cm.

B. 3D Textured Object Models

Each object was scanned with a low-cost EinScan-SE desktop 3D scanner to generate textured 3D object meshes. The scanning process introduced another limitation on both the size and materials of the objects, due to the inability of the scanner to handle objects larger than about 20 cm on any side. The software tends to produce fragmented texture maps, so we used Maya to further refine the textures. Fig. 2 shows a synthetic rendering of the 3D textured object models, rendered using NViSII, a Python-interfaced ray tracing library [17]. Textured object models are useful for generating training data, as in DOPE [4] or BlenderProc [18].

C. Capturing Real Images

To acquire images for the dataset, we placed the real objects in ten different environments, with five object arrangements / camera poses per environment. The environments are shown in Fig. 3, and the object arrangements for one environment are depicted in Fig. 4. These 50 different scenes exhibit a wide variety of backgrounds, clutter, poses, and lighting. With some arrangements, objects are placed inside other containers (*e.g.,* boxes, bags, or drawers) to provide additional clutter and partial occlusion. Both RGB and depth images were collected for each scene using an Intel RealSense D415 RGBD camera at full HD resolution (1920×1080). Images were captured from distances of 0.5 to 1.0 m, which are typical of robotic grasping.

Once the objects were arranged in an environment and the camera was set, we acquired multiple images with various lighting conditions by turning on/off the lights, opening window blinds, and so forth. Since the scene was static, the annotations (described below) do not change and thus required no additional work. In this manner we were able to collect a large variety of lighting conditions, see Fig. 5.

In total, the dataset contains 50 unique scenes, 238 images (an average of 4.8 lighting variations per scene), and 914 object poses. Images from two environments ("chair") and ("window 1") are made available with ground truth for a validation set, while the remaining scenes are reserved as the test set. Minimum and maximum object dimensions are in the ranges 2.4–7.2 cm and 6.8–25.0 cm, respectively. Average object visibility is 83%, with only 39% of objects in unoccluded poses (>95% visible). Objects are set in a wide variety of orientations, and object distances range from 39.0–141.0 cm with an average distance of 73.8 cm.

D. Annotating Images with Ground Truth

We annotated the dataset with ground truth object poses by manually identifying point correspondences between images and 3D textured object models. In the PnP version of our annotation tool, the annotator selects corresponding points in the 2D texture map of the object model and the image in which the object appears, after which the 3D model is

Fig. 3: Sample images from the ten environments. "Chair" and "window 1" are included in the validation set.

Fig. 4: Five object arrangements, shown for the "break room" environment.

Fig. 5: Four lighting variations of the same scene. Insets show zoomed crops.

aligned to the image by PnP with RANSAC. Alternatively, in the RGBD version, correspondences are made between the 3D textured model and the 2.5D RGBD depth map, with alignment made by Procrustes. The RGBD annotation tool is typically faster to operate but may suffer from noise or bias in depth measurements, while PnP is more error-prone along the projection ray from the camera to the object. Most annotations were performed with the RGBD tool, resorting to PnP only when the depth-based annotation was not satisfactory. All annotations were automatically refined by SimTrack [19]. Then, if necessary, the alignment was refined manually until the reprojected model was visually aligned with both the image and the depth map.

E. Depth Calibration

Following Hodaň et al. [20], we checked the depth calibration of the RealSense camera and found a small systematic error. We captured over 50 images of a checkerboard at distances between 0.5 and 2 m using the RealSense left-infrared camera. Checkerboard corners were automatically detected, then unprojected to 3D using PnP and the camera intrinsics. The resulting coordinates were compared with the corresponding 3D coordinates obtained from the 2.5D depth map. Fitting a scale factor of 0.9804 reduced the mean absolute difference between measurements from 19.3 to 7.6 mm. The depth images included in the dataset were scaled by this factor before registering to RGB.

F. Symmetry-Aware Metrics

Perhaps the most common and easily interpretable metric for evaluating pose estimates is the average distance (ADD) metric [21], which computes the mean pairwise distance between corresponding vertices in the 3D object model under the ground truth ($\bar{\mathbf{P}}$) and predicted ($\hat{\mathbf{P}}$) poses:

$$e_{\text{ADD}} = \underset{\mathbf{x} \in O}{\text{mean}} \|\bar{\mathbf{P}}\mathbf{x} - \hat{\mathbf{P}}\mathbf{x}\|, \tag{1}$$

where $\mathbf{x} \in O$ are the positions of model vertices in the object coordinate frame, and $\|\cdot\|$ is the ℓ_2-norm.

Because ADD makes no allowances for object symmetries, it provides only limited insight into the *graspability* of detected objects based on the predicted pose, as translation errors are penalized similarly to rotation errors. Moreover, in the case of symmetric objects, ADD penalizes predictions even when the true pose cannot be determined from the input image. To overcome these limitations, some researchers adopt a mean closest point distance (ADD-S) [22]:

$$e_{\text{ADD-S}} = \underset{\mathbf{x}_1 \in O}{\text{mean}} \; \underset{\mathbf{x}_2 \in O}{\min} \|\bar{\mathbf{P}}\mathbf{x}_1 - \hat{\mathbf{P}}\mathbf{x}_2\|, \tag{2}$$

where each vertex in the ground truth model is paired with the *nearest* vertex in the prediction model, irrespective of consistency with other assignments. As shown later in this section, this metric can significantly underestimate pose error due to unrealistic pairings.

Another way to handle symmetries is to directly model the set $\mathbf{S}_O$ of symmetry transformations corresponding to

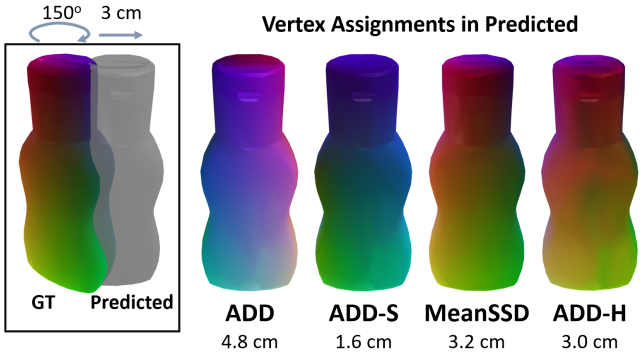


Fig. 6: Pairwise vertex assignments (shown by vertex color) between ground truth (GT) and predicted model poses using four different metrics.

the object. Starting from ADD in Eq. (1), this leads to the Mean Symmetry-Aware Surface Distance (MeanSSD):

$$e_{\text{MeanSSD}} = \min_{\mathbf{S}\in\mathbf{S}_O} \operatorname*{mean}_{\mathbf{x}\in O} \|\bar{\mathbf{P}}\mathbf{S}\mathbf{x} - \hat{\mathbf{P}}\mathbf{x}\|. \tag{3}$$

MeanSSD requires calculating or manually identifying all valid (discrete and/or continuous) symmetries for a given object, a potentially burdensome task for a large and varied set of objects. If the mean operator in Eq. (3) is replaced with max, MeanSSD becomes the Maximum Symmetry-Aware Surface Distance (MSSD) proposed by Hodaň et al. [23]:

$$e_{\text{MSSD}} = \min_{\mathbf{S}\in\mathbf{S}_O} \max_{\mathbf{x}\in O} \|\bar{\mathbf{P}}\mathbf{S}\mathbf{x} - \hat{\mathbf{P}}\mathbf{x}\|. \tag{4}$$

As a compromise between the inaccurate ADD-S metric and the labor-intensive MeanSSD/MSSD metrics, we propose a variant of the average distance metric in which the vertex correspondences between ground truth and predicted models are made by solving a linear sum assignment problem. The most famous solution to this assignment problem is the Hungarian algorithm [24], hence we denote this metric ADD-H. The assignment $f_A : \mathbf{x}_2 \mapsto \mathbf{x}_1$ produces a bijective mapping from the vertices of the model in the predicted pose to those in the ground truth pose. If we let $A = \{(f_A(\mathbf{x}_2), \mathbf{x}_2)\}$ be the set of such correspondences, then the set that minimizes the sum of distances between paired vertices is calculated as:

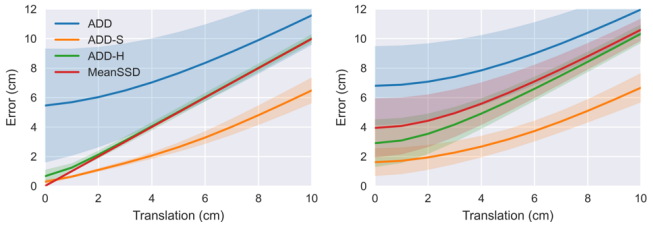
$$\tilde{A} = \arg\min_{A} \sum_{(\mathbf{x}_1,\mathbf{x}_2)\in A} \|\bar{\mathbf{P}}\mathbf{x}_1 - \hat{\mathbf{P}}\mathbf{x}_2\|. \tag{5}$$

We compute Eq. (5) using a modern variation of the Hungarian algorithm.[3] The average distance is then computed between assigned pairs $(\mathbf{x}_1, \mathbf{x}_2) \in \tilde{A}$:

$$e_{\text{ADD-H}} = \operatorname*{mean}_{(\mathbf{x}_1,\mathbf{x}_2)\in\tilde{A}} \|\bar{\mathbf{P}}\mathbf{x}_1 - \hat{\mathbf{P}}\mathbf{x}_2\|. \tag{6}$$

Fig. 6 visualizes the vertex correspondences used by each metric when computing the error between the ground truth and predicted pose for a bottle. The inset rendering on the left depicts the scene, where the predicted pose (in translucent gray) is offset from the ground truth pose (with colored vertices) by a 150° rotation about the vertical axis and a horizontal translation of about half the object width (3 cm).

[3]Implemented as `scipy.optimize.linear_sum_assignment`. Note that versions of SciPy before 0.17.0 use a much slower algorithm.



(a) Symmetry-preserving (b) Non-symmetry-preserving

Fig. 7: Error metrics (±1 standard deviation) between object poses with random rotations and increasing translation. ADD-H closely approximates MeanSSD.

The right side of the figure depicts the vertex assignments for each metric: vertices in each model ($\mathbf{x}_2 \in O$) are colored according to the paired vertices in the ground truth model ($\mathbf{x}_1 \in O$). Since ADD uses a fixed, bijective assignment based on the original vertex ordering, the error is large. In contrast, ADD-S assigns the nearest ground truth vertex to each target, resulting in a non-injective mapping, in which only points from the right side of the ground truth are used, as reflected in the green and purple color across the entire target mesh. MeanSSD uses the same vertex assignment as ADD but explicitly selects the symmetry-preserving rotation that minimizes error, resulting in a rotational alignment that is more representative of the grasp-relevant geometry of the prediction. Finally, ADD-H optimizes a bijective mapping between poses to minimize error—achieving a vertex assignment that reflects the pose symmetry, similar to MeanSSD.

To further compare the metrics, Fig. 7 shows the error, averaged over 5 trials for each of the 28 HOPE objects, as each object is translated from its initial position in increments of 1 cm. In the left plot, each object is first randomly rotated by one of the symmetry-preserving transformations, whereas in the right plot, each object is first randomly rotated by an arbitrary transformation. ADD-H closely matches the MeanSSD error without any explicit enumeration of object symmetries, especially when the rotation is symmetry-preserving. In contrast, ADD and ADD-S significantly over- and under-estimate the error, respectively.

Although ADD-H is computationally more expensive than related methods, by using an efficient algorithm to solve the assignment problem it can be applied to meshes with several hundred vertices. In our experiments, we use 500 vertices, which yields consistent results with sufficiently fast computation.

III. Experiments

We first present a validation experiment that examines the accuracy of our annotations. Then we provide a baseline experiment where we train several pose detectors and report accuracy metrics for the dataset.

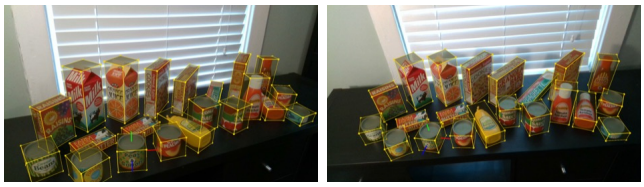


Fig. 8: Two views for computing accuracy via consistency.

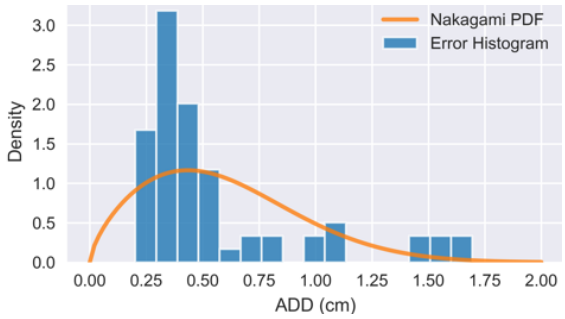


Fig. 9: Histogram of ADD errors in annotation validation experiment with best fit Nakagami distribution ($m = 0.831, \Omega = 0.830$, mean $= 0.597$, mode $= 0.435$).[4]

A. Annotation Validation Experiment

To estimate the error in our ground truth pose annotations, we captured orthogonal views ($\sim$90 degrees between camera axes) of three static scenes. Objects were annotated independently in both views, as shown in Fig. 8. Using a robust Procrustes alignment of all but one annotated object pose, we estimated the extrinsics between the two camera views. The pose of the held-out object was projected from the first view to the second, and ADD error was computed between the transferred pose and the annotation made directly in the second view. By repeating the process while holding out each object in turn, we estimated the annotation error for all objects in the scene. Fig. 9 shows a histogram of ADD for the 64 object instances contained in the three scenes. The mean and median ADD are 5.7 and 4.3 mm, respectively. We conclude, therefore, that ground truth is accurate to several millimeters.

B. Pose Prediction Baselines

We trained baseline pose prediction models for each object class using the Deep Object Pose Estimation (DOPE) [4] and CosyPose [3] methods for detecting and predicting object poses in RGB images. For DOPE, models were trained from synthetic images using domain randomization [25]. Unlike the original DOPE paper, we did not train using any photorealistic rendered images, which no doubt degrades performance. Accordingly, we denote the baseline as DOPE-DR. Two architecture variants are considered with 50×50 and 400×400 output maps, denoted DOPE-DR-50 and DOPE-DR-400, respectively. For CosyPose, we trained using photorealistic images rendered by the BlenderProc tool [18].

As both DOPE and CosyPose are RGB-only methods, both can suffer from errors along the projection ray from the camera to the object. We consider a simple method for refinement using the RGB-D point clouds: given a predicted pose, visible model vertices are aligned with the depth map by adjusting the predicted translation by a scalar factor determined by a line search. We denote this method with "-LS". CosyPose has a distinct advantage in that the object mask from the Mask R-CNN detector can be used to filter both the object mesh and RGB-D point cloud, thereby limiting the effect of occluding objects.

[4]If the vertices in two random poses have Gaussian errors, then the sum of their squared differences is given by a gamma distribution. If z follows a gamma distribution with parameters α and β, then $\sqrt{z}$ follows a Nakagami distribution with parameters $m = \alpha$ and $\Omega = \alpha\beta$. Thus the mean Euclidean distance in Eq. (1) is expected to follow a Nakagami distribution.

TABLE I: Average recall under three BOP Challenge metrics for the HOPE test set.

Method	Modality	AR_{MSSD}	AR_{VSD}	AR_{MSPD}
DOPE-DR-50	RGB	0.232	0.215	0.457
DOPE-DR-50-LS	RGB-D	0.298	0.258	0.439
DOPE-DR-400	RGB	0.297	0.277	0.498
DOPE-DR-400-LS	RGB-D	0.317	0.285	0.470
CosyPose	RGB	0.476	0.550	**0.637**
CosyPose-LS	RGB-D	**0.594**	**0.691**	0.629

C. BOP Challenge

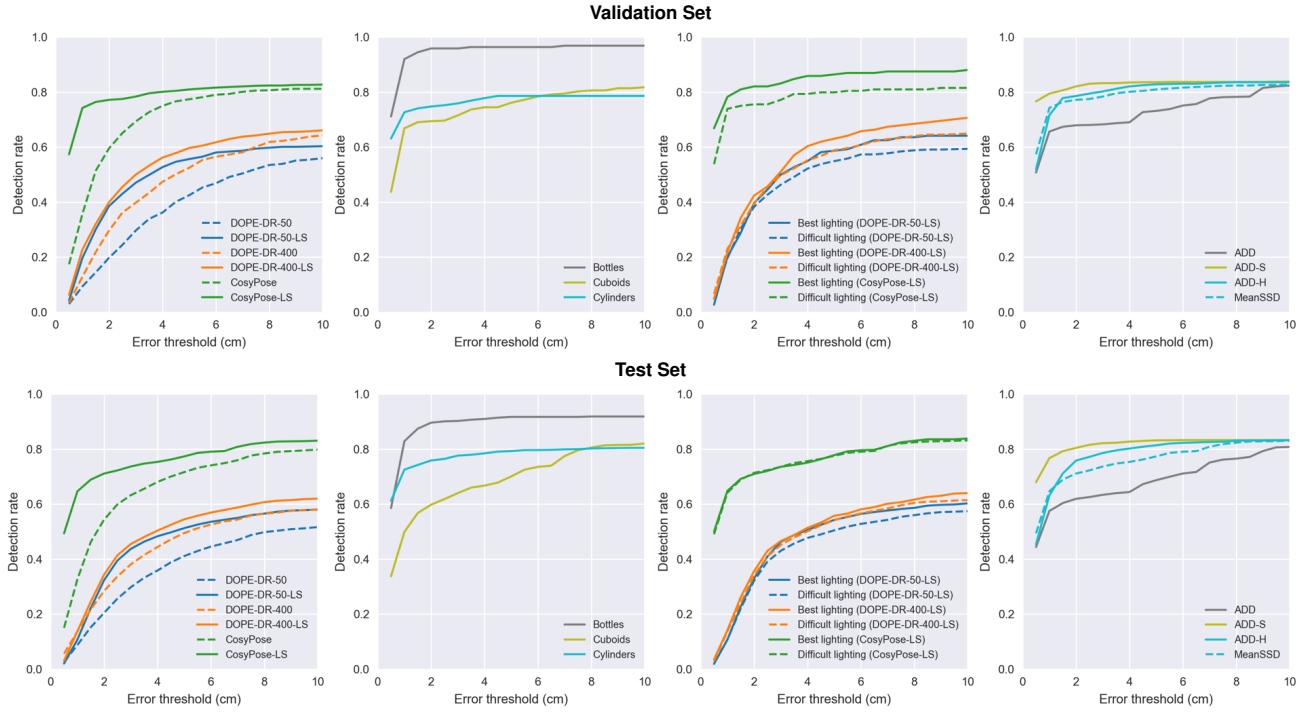
Table I shows the average recall (AR) under three metrics *as computed by the BOP benchmark* [14]: Maximum Symmetry-Aware Surface Distance (MSSD), Visible Surface Discrepancy (VSD), and Maximum Symmetry-Aware Projection Distance (MSPD). MSSD, which measures 3D alignment of model vertices, is the most relevant to robotics applications. VSD measures the discrepancy between 2D distance maps, rendered for GT and predicted poses, respectively. MSPD measures the discrepancy between 2D projections of model vertices, thereby capturing visual alignment with an eye toward augmented reality applications.

CosyPose-LS yields a nearly 2x improvement in AR_{MSSD} over DOPE-400-LS. Furthermore, mistakes made along the projection ray to the object are shown to be a significant source of error, as evidenced by the dramatic improvement made by our simple depth refinement (-LS). The refinement improves MSSD and VSD scores dramatically for all three methods, including a 25% improvement in CosyPose. Because MSPD is sensitive to subtle changes in visual alignment, it suffers slightly under depth refinement despite overall better 3D alignment.

Two aspects of the BOP metrics are worth noting: 1) as mentioned previously, the BOP definition of symmetries requires both geometric and (nearly) exact visual similarity, meaning that none of the HOPE objects are considered symmetric in the BOP challenge; and 2) the recall error threshold is defined *relative* to object size, varying from 5% to 50% of the object diameter [23].

D. Detailed Experiments

Because of our focus on graspability, we present additional results using *absolute* detection thresholds, varying from



(a) Methods (MeanSSD) (b) Object categories (CosyPose-LS, MeanSSD) (c) Lighting (MeanSSD) (d) Metrics (CosyPose-LS)

Fig. 10: Detection rate versus maximum detection threshold, computed on the HOPE validation set (top row) and test set (bottom row).

0.5 to 10 cm. We also consider all of our objects to be geometrically symmetric, ignoring small details like tabs on can lids. For defining geometric symmetries, meshes were manually aligned to a consistent set of coordinate axes and grouped into three categories: *cylinders* (360° rotation around vertical axis in 1° increments, and 180° flip top to bottom), *cuboids* (180° flip along any axis), and *bottles* (180° flip front to back).

Fig. 10 presents the results of four experiments, depicting the detection rate of ground truth objects (recall) on the y-axis while varying the maximum error threshold for a positive detection on the x-axis. We show results on both validation and test sets. To facilitate interpretability by those familiar with existing metrics, we use the MeanSSD metric.[5]

In the analysis that follows, we specifically highlight performance at two particular error thresholds: 2 cm and 10 cm. The smaller threshold (2 cm) approximately indicates errors sufficient for grasping, although this will of course vary by gripper. The larger threshold (10 cm) indicates errors sufficient for a system that is able to gather additional views of a detected object for refined results, *e.g.,* when the camera is in the robot hand—although this threshold also is an approximation.

Fig. 10 (a) compares CosyPose and DOPE using the MeanSSD metric. CosyPose-LS, trained with photorealistic images, incorporating a visual refinement step, and augmented with our simple depth refinement, predicts over 70% of objects in the HOPE test set within 2 cm. Considering DOPE, the figure shows the orthogonal benefits of a larger output map (DOPE-DR-400-*)—which improves the precision of keypoint prediction in the DOPE method—and line search depth refinement (DOPE-DR-*-LS). With line search and depth refinement, more than 30% of objects in the HOPE test set are predicted within 2 cm.

Fig. 10 (b) shows the performance of CosyPose-LS using the MeanSSD metric across three object categories. Cuboids present a particular challenge since they are less resistant to small rotational errors. In this dataset, cuboids also tend to be larger objects that are placed further from the camera and behind other objects in some scene arrangements. Highest success is obtained with bottles, with almost 90% predicted within 2 cm.

Fig. 10 (c) depicts the effect of lighting variations by comparing detection rate (again using MeanSSD) between the most favorable image and most difficult image in each scene (determined subjectively). Particularly on the test set, which is much larger than the validation set, the difference in accuracy is small, indicating a promising level of robustness in these synthetically trained methods.

Finally, Fig. 10 (d) compares the various metrics using CosyPose-LS predictions, confirming previous observations that ADD-H is a reasonable compromise that matches the results of MeanSSD without requiring explicit enumeration of object symmetries. As mentioned earlier, ADD overestimates, while ADD-S under-estimates, the error.

Finally, Table II presents detailed statistics of CosyPose-

[5]For versions of Fig. 10 (a-c) and Table II using the ADD-H metric, see the Appendix of the arXiv version of this paper.

LS predictions for each object class and category, using the MeanSSD metric, along with some statistics of the ground truth (GT) object poses. The table includes the median of the MeanSSD error among objects detected within a 10 cm threshold, as well as precision and recall at both 2 cm and 10 cm thresholds. At the most generous 10 cm detection threshold, 83% of ground truth objects (in the test set) are detected by CosyPose-LS, and at the tighter 2 cm threshold, 72% are detected, meaning that grasping may be feasible. CosyPose-LS could benefit from an improved object detection step, as nearly 20% of objects are not detected. False positives appear to be less of a problem, with >98% precision at the 10 cm threshold. These results offer room for improvement, while showing that an existing method like CosyPose can be reasonably successful as an off-the-shelf predictor for these objects.

IV. Relationship to Previous Work

Many existing datasets focus on the task of 6-DoF pose estimation of known objects [22], [26], [21], [20], [14], [27], [28], [29]. The original LineMOD dataset [21], for example, offers manual object annotations for approximately 1000 images for each of the 15 objects in the dataset. The LineMOD-Occluded dataset consists of additional annotations of the original dataset [30]. T-LESS consists of real images of untextured manufacturing parts [20]. The falling things (FAT) dataset [26] consists of synthetically generated images of random YCB [15] objects which fall under the influence of simulated gravity in 3D scenes. The YCB-Video dataset provides a large number of images from video sequences, with high correlation between images and an average of five objects visible per image [22], [31]. The YCBInEOAT ("in the end of arm tooling") dataset [12] consists of videos of 5 YCB objects, one at a time, being held and moved. The DexYCB dataset [32] contains multi-camera videos of humans grasping 20 YCB objects, one at a time. Our HOPE dataset contains 28 objects in 50 cluttered scenes, nearly 5 lighting variations per scene (for a total of 238 images), and an average of more than 18 objects per scene. The HOPE objects also appear in our HOPE-Video dataset [33], a collection of 10 short RGBD video sequences captured by a robot arm-mounted camera, each depicting 5-20 objects on a tabletop workspace.

Recently the BOP (Benchmark for 6D Object Pose Estimation) challenge [14] proposed assembling multiple 6-DoF pose estimation datasets as a central resource for evaluating algorithms. It is composed of the following datasets: LineMOD [34], LineMOD-Occluded [30], T-LESS [20], ITODD [35], HomebrewedDB [36], YCB-Video [22], Rutgers APC [16], IC-BIN [37], IC-MI [38], TUD Light [14], Toyota Light [14], and now HOPE. With the addition of these HOPE objects, we believe that this benchmark is more relevant to robotics researchers.

V. Conclusion

In this work we offer a pose estimation benchmark that is immediately applicable to robotic manipulation research. Researchers can render synthetic images using the 3D textured meshes accompanying our dataset to train pose estimation models for 28 different objects. These models can be evaluated using our challenging, accurately annotated real-world images. Finally, pose predictions can be used directly in robotics labs for detecting and manipulating physical copies of the same objects, which are readily available from online retailers. We hope that this dataset will be a useful and practical bridge between researchers in computer vision and robotics.

Acknowledgements

The authors would like to thank Tomáš Hodaň for his feedback and assistance throughout this work and for integrating the HOPE dataset into the BOP benchmark.

References

[1] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *CVPR*, 2005. 1
[2] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *ICRA*, 2011. 1
[3] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *ECCV*, 2020. 1, 2, 5
[4] 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 *CoRL*, 2018. 1, 2, 5
[5] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in *CVPR*, 2019. 1
[6] S. Peng, Y. Liu, Q. Huang, H. Bao, and X. Zhou, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *CVPR*, 2019. 1
[7] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *ICCV*, 2019. 1
[8] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *ECCV*, 2018. 1
[9] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *ICCV*, 2017. 1
[10] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu, "6-PACK: Category-level 6D pose tracker with anchor-based keypoints," in *ICRA*, 2020. 1
[11] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *CVPR*, 2019. 1
[12] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se(3)-TrackNet: Data-driven 6D pose tracking by calibrating image residuals in synthetic domains," in *IROS*, 2020. 1, 7
[13] B. Wen and K. Bekris, "BundleTrack: 6D pose tracking for novel objects without instance or category-level 3D models," in *IROS*, 2021. 1
[14] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A. G. 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 *ECCV*, 2018. 1, 5, 7
[15] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set," *IEEE Robotics and Automation Magazine*, vol. 22, no. 3, Sep. 2015. 1, 7
[16] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place," *Robotics and Automation Letters (RAL)*, vol. 1, no. 2, 2016. 1, 7
[17] N. Morrical, J. Tremblay, Y. Lin, S. Tyree, S. Birchfield, V. Pascucci, and I. Wald, "NViSII: A scriptable tool for photorealistic image generation," in *ICLR Workshop on Synthetic Data Generation (SDG)*, 2021. 2
[18] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, "BlenderProc," *arXiv preprint arXiv:1911.01911*, 2019. 2, 5

TABLE II: Results on the validation and test sets using CosyPose with the MeanSSD metric. (The median error was computed among all "true positive" predictions, determined at a 10 cm error threshold.)

			Pose Statistics			Validation Set					Test Set				
							Precision (%)		Recall (%)			Precision (%)		Recall (%)	
	Object	Diameter (cm)	Visibility (%)	Pixels ($\times 10^3$)	Distance (cm)	Median Error (cm)	@2cm	@10cm	@2cm	@10cm	Median Error (cm)	@2cm	@10cm	@2cm	@10cm
CYLINDERS	Mushrooms	7.6	77.7	13.5	68.5	0.3	87.9	100.0	82.9	94.3	0.3	87.6	100.0	61.4	70.1
	Tuna	7.6	82.0	12.3	73.0	0.4	100.0	100.0	88.6	88.6	0.3	90.9	98.7	63.1	68.5
	Yogurt	8.3	88.1	18.6	72.3	0.3	94.1	100.0	80.0	85.0	0.3	98.8	100.0	83.2	84.2
	Peaches	8.9	86.4	21.9	74.4	0.2	100.0	100.0	85.0	85.0	0.3	80.8	100.0	64.6	80.0
	Pineapple	8.9	82.5	18.5	70.5	0.3	97.3	100.0	65.5	67.3	0.3	96.4	100.0	79.0	82.0
	Cherries	9.0	87.5	22.5	67.5	0.4	73.3	100.0	55.0	75.0	0.3	87.4	93.7	79.9	85.6
	Corn	9.0	81.1	20.6	69.4	0.3	100.0	100.0	57.5	57.5	0.3	91.1	100.0	67.8	74.4
	Green Beans	9.0	86.3	22.5	69.9	0.2	93.1	100.0	77.1	82.9	0.3	92.5	97.2	76.7	80.6
	Peas & Carrots	9.0	88.1	21.7	70.3	0.2	100.0	100.0	73.3	73.3	0.3	99.0	100.0	81.6	82.4
	Tomato Sauce	10.7	83.6	17.9	79.9	0.2	100.0	100.0	80.0	80.0	0.3	100.0	100.0	76.4	76.4
	Alphabet Soup	10.8	85.0	27.1	70.8	0.2	96.9	100.0	68.9	71.1	0.3	93.5	98.9	78.9	83.5
	Parmesan	12.3	85.5	26.1	77.4	0.3	100.0	100.0	83.3	83.3	0.4	100.0	100.0	97.8	97.8
	Mean	**9.2**	**84.5**	**20.3**	**72.0**	**0.3**	**95.2**	**100.0**	**74.8**	**78.6**	**0.3**	**93.2**	**99.0**	**75.9**	**80.4**
CUBOIDS	Chocolate Pudding	9.8	88.4	17.4	70.9	0.3	96.2	100.0	83.3	86.7	0.3	77.8	100.0	69.5	89.4
	Butter	11.5	82.7	16.7	72.5	0.2	97.4	100.0	84.4	86.7	0.3	83.2	100.0	52.3	62.9
	Cream Cheese	11.6	85.0	17.9	76.6	0.2	100.0	100.0	100.0	100.0	0.3	78.3	100.0	66.4	84.8
	Raisins	15.1	76.9	31.0	73.8	0.4	92.9	100.0	65.0	70.0	0.4	96.5	98.8	69.7	71.4
	Popcorn	15.2	78.3	32.7	74.8	2.7	44.4	100.0	20.0	45.0	1.3	45.7	83.5	42.6	77.9
	Milk	20.5	82.5	45.1	80.5	0.7	75.0	100.0	72.0	96.0	4.7	45.0	99.2	40.9	90.2
	Orange Juice	20.6	78.1	50.4	75.3	0.8	60.0	84.0	60.0	84.0	4.9	44.1	100.0	43.8	99.3
	Granola Bars	20.6	77.3	47.8	78.2	0.8	43.5	78.3	40.0	72.0	0.6	79.1	91.8	66.9	77.7
	Mac & Cheese	20.7	75.3	53.2	75.1	0.3	92.7	95.1	76.0	78.0	0.8	84.4	100.0	69.9	82.8
	Cookies	20.9	77.1	44.4	81.6	0.6	90.9	95.5	80.0	84.0	0.4	87.1	98.8	71.8	81.6
	Spaghetti	25.3	78.9	35.2	74.5	0.6	85.3	100.0	82.9	97.1	1.1	74.3	98.2	63.6	84.1
	Mean	**17.4**	**80.0**	**35.6**	**75.8**	**0.7**	**79.8**	**95.7**	**69.4**	**81.8**	**1.4**	**72.3**	**97.3**	**59.8**	**82.0**
BOTTLES	Salad Dressing	15.1	89.0	27.7	72.9	0.3	100.0	100.0	100.0	100.0	0.4	96.2	100.0	90.0	93.6
	Mayo	15.3	87.3	28.5	77.0	0.3	100.0	100.0	100.0	100.0	0.4	97.4	100.0	88.9	91.3
	BBQ Sauce	15.4	87.1	27.1	73.7	0.4	100.0	100.0	93.3	93.3	0.5	98.0	99.0	93.2	94.2
	Ketchup	15.4	89.8	29.5	72.8	0.3	94.7	100.0	90.0	95.0	0.3	98.4	98.4	83.8	83.8
	Mustard	16.1	89.2	31.5	71.8	0.5	100.0	100.0	96.0	96.0	0.5	95.6	100.0	92.1	96.4
	Mean	**15.5**	**88.5**	**28.9**	**73.6**	**0.4**	**98.9**	**100.0**	**95.9**	**96.9**	**0.4**	**97.1**	**99.5**	**89.6**	**91.8**
	Overall Mean	**13.6**	**83.4**	**27.8**	**73.8**	**0.5**	**89.8**	**98.3**	**76.4**	**83.1**	**0.7**	**85.7**	**98.4**	**72.0**	**83.1**

[19] K. Pauwels and D. Kragic, "SimTrack: A simulation-based framework for scalable real-time object pose detection and tracking," in *IROS*, 2015. 3

[20] T. Hodaň, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects," in *WACV*, 2017. 3, 7

[21] 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 *ACCV*, 2012. 3, 7

[22] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *RSS*, 2018. 3, 7

[23] 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 *ECCV Workshop*, 2020. 4, 5

[24] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016. 4

[25] 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 *IROS*, 2017. 5

[26] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3D object detection and pose estimation," in *CVPR Workshop on Real World Challenges and New Benchmarks for Deep Learning in Robotic Vision*, Jun. 2018. 7

[27] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. Anastasiu, and J.-N. Hwang, "CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *CVPR*, 2019. 7

[28] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese, "ObjectNet3D: A large scale database for 3D object recognition," in *ECCV*, 2016. 7

[29] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *WACV*, 2014. 7

[30] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," in *ECCV*, 2014. 7

[31] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Intl. Conf. on Advanced Robotics (ICAR)*, 2015. 7

[32] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. V. Wyk, U. Iqbal, S. Birchfield, J. Kautz, and D. Fox, "DexYCB: A benchmark for capturing hand grasping of objects," in *CVPR*, 2021. 7

[33] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Multi-view fusion for multi-level robotic scene understanding," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6817–6824. 7

[34] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *PAMI*, vol. 34, no. 5, pp. 876–888, 2012. 7

[35] B. Drost, M. Ulrich, P. Bergmann, P. Härtinger, and C. Steger, "Introducing MVTec ITODD–a dataset for 3D object recognition in industry," in *ICCV Workshop on Recovering 6D Object Pose*, 2017. 7

[36] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic, "HomebrewedDB: RGB-D dataset for 6D pose estimation of 3D objects," in *arXiv:1904.03167*, 2019. 7

[37] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim, "Recovering 6D object pose and predicting next-best-view in the crowd," in *CVPR*, 2016. 7

[38] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim, "Latent-class Hough forests for 3D object detection and pose estimation," in *ECCV*, 2014. 7