SAR-Net: Shape Alignment and Recovery Network
for Category-level 6D Object Pose and Size Estimation

Haitao Lin[1§]   Zichang Liu[1]   Chilam Cheang[1]   Yanwei Fu[1†‡]   Guodong Guo[2]   Xiangyang Xue[1†]

[1]Fudan University   [2]IDL, Baidu Research

Abstract

Given a single scene image, this paper proposes a method of Category-level 6D Object Pose and Size Estimation (COPSE) from the point cloud of the target object, without external real pose-annotated training data. Specifically, beyond the visual cues in RGB images, we rely on the shape information predominately from the depth (D) channel. The key idea is to explore the shape alignment of each instance against its corresponding category-level template shape, and the symmetric correspondence of each object category for estimating a coarse 3D object shape. Our framework deforms the point cloud of the category-level template shape to align the observed instance point cloud for implicitly representing its 3D rotation. Then we model the symmetric correspondence by predicting symmetric point cloud from the partially observed point cloud. The concatenation of the observed point cloud and symmetric one reconstructs a coarse object shape, thus facilitating object center (3D translation) and 3D size estimation. Extensive experiments on the category-level NOCS benchmark demonstrate that our lightweight model still competes with state-of-the-art approaches that require labeled real-world images. We also deploy our approach to a physical Baxter robot to perform grasping tasks on unseen but category-known instances, and the results further validate the efficacy of our proposed model. Code and pre-trained models are available on the project webpage [1].

1. Introduction

Estimating accurate 6D poses of objects plays a pivotal role in the tasks of augmented reality [34], scene under-

---

[1]Project webpage. https://hetolin.github.io/SAR-Net

Figure 1. (a) The visual difference between synthetic and real images. They may have different textures and colors, but the shape and geometry maintain the same. (b) Illustration of shape alignment. Objects within a category have consistent 3D rotation if their shapes are visually aligned. (c) Illustration of symmetric correspondence. Most objects are manufactured with (near) symmetric shapes with reflectional symmetry or rotational symmetry.

standing [49], and robotic manipulation [7, 9, 33, 51, 53]. However, most 6D pose estimation works [8,9,15,19,28,37, 39,55,57] assume exact 3D CAD object models at *instance-level*, which unfortunately greatly limits their practical applicability in real-world applications. To this end, this paper studies the task of Category-level 6D Object Pose and Size Estimation (COPSE). Thus the model is trained only by category-level supervision, reducing reliance on the exact CAD model for each instance.

Generally, the key challenge of COPSE task lies in the huge color and shape variations of instances from the same category [43–45]. To handle *intra-class variations*, previous works [5,24,52,58] learn the RGB(-D) features of each instance to help map these instances into a unified space and minimize the intra-class variations. On the other hand, as the COPSE task relies on supervised learning from large amounts of well-labeled data, recent works [5, 30, 52, 58] utilize synthetic data to train the COPSE model. Unfortunately, as illustrated in Fig. 1(a), the *domain gap* between synthetic and real images potentially hinders the perfor-

mance of COPSE model in the real-world deployment.

While most previous works exploit texture and color cues in RGB images, the shape information has been less touched, with some recent exceptions of reconstructing the observed point cloud [6], and analyzing geometric stability of object surface patches [47]. For example, cups of similar or identical shapes have very diverse colors in Fig. 1(a). This motivates us to systematically explore shape information predominately from the depth (D) channel. Thus to alleviate challenges of intra-class variation and synthetic-real image domain gap, we propose encoding the shape by *shape alignment* and *symmetric correspondence*. Particularly, our method encourages insightful shape analysis about geometrical similarity and symmetric correspondence.

Shape alignment. Assuming the instances of the same category are well aligned by shapes, they should have the consistent 3D rotation, as cups are shown in Fig. 1(b). Thereby, the idea of shape alignment can be implemented as object 3D rotation consistency. In particular, given a category-level template shape in the form of a point cloud, it is deformed to align against the observed instance point cloud. We denote such the deformed template point cloud as an implicit representation for object 3D rotation, as shown in Fig. 2(a). Mathematically, the object rotation is thus recovered by solving the classical orthogonal Procrustes problem [46], which calculates the approximation of alignment matrix between point clouds of the category-level template and deformed one. The shape alignment learns to be robust to intra-class variations of instances.

Symmetric correspondence. Given the fact that many man-made object categories have the design principle with a symmetric structure [62], symmetry is an important geometric cue to help our COPSE task. As in Fig. 1(c), the underlying symmetry allows for reasoning the reflectional and rotational symmetry of 3D shape from occluded 2D images. Note that specific object instances are practically never perfectly symmetric due to various shape variations of instances. To this end, we exploit the underlying symmetry by point clouds of objects, as our COPSE task does not demand the exact 3D shape recovery. Furthermore, we model the point cloud of symmetric objects by an encoder-decoder structure learned end-to-end with the other components of our framework. Thus, this actually facilitates the whole framework being robust to those objects which have some parts that are less symmetric as in Fig. 2(b).

Formally, this paper proposes a novel Shape Alignment and Recovery Network (SAR-Net) to exploit the underlying object shapes for the COPSE tasks. Specifically, the RGB-D scene image is utilized as the input. We firstly employ Mask-RCNN [14] to pre-process the RGB image, and infer the segmentation mask and category label of each object instance. The points from depth channel are filtered by the predicted mask and further fed into the 3D segmentation
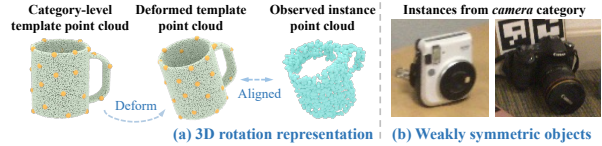

Figure 2. (a) Illustration of implicit 3D object rotation. The deformed template point cloud posses same 3D rotation with observed instance point cloud. (b) Illustration of weakly symmetric objects. Such objects usually have global symmetric shapes but asymmetric local parts.

network 3D-GCN [31] to generate observed point cloud of the object. Furthermore, taking as inputs the point clouds of both observed object instance and category-level template, our SAR-Net predicts the implicit representation of deformed template point cloud, and infers symmetric point cloud. The 3D object rotation is further computed from the category-level and deformed template point clouds by Umeyama algorithm [54]. Finally, we concatenate the observed and symmetric point clouds for a coarse object shape obtainment, which reduces the estimation uncertainty of object center (3D translation) and 3D size. Extensive experiments conducted on the category-level NOCS dataset [58] demonstrate that our synthetic-only approach outperforms the state-of-the-art methods.

Contributions. Our main contribution is to propose a novel learning paradigm that efficiently encodes the shape information by the shape alignment and symmetric correspondence for the COPSE. We present a novel framework – SAR-Net to implement this idea. In particular,

1) Based on shape similarities, our SAR-Net has the novel sub-net component that efficiently infers the implicit rotation representation by shape alignment between point clouds of the category-level template shape and instance.

2) A novel sub-net component for symmetric correspondence is proposed in this paper. It can predict symmetric point cloud from partially observed point cloud to obtain a coarse shape. The coarse shape helps to estimate the object center and size accurately.

3) Practically, our SAR-Net is a very lightweight model with only 6.3M parameters. Such a single model is capable of doing the COPSE of multiple categories, and performs better than previous approaches of more model parameters.

4) Critically, our SAR-Net is entirely trained on synthetic data and performs very well generalization on real-world scenarios. Remarkably, our synthetic-only approach still outperforms other competitors which typically require both synthetic and real-world data.

2. Related Work

Instance-Level 6D Object Pose Estimation. Most previous works [3, 17, 27, 32] estimate object pose by matching image features. Unfortunately, these methods are less effi-
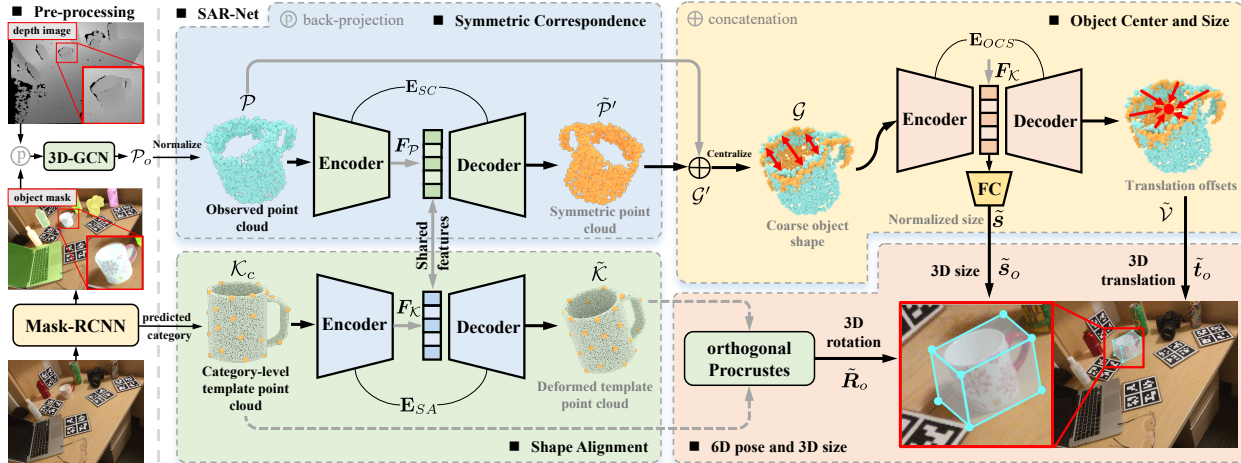
Figure 3. Architecture overview. The pre-processing stage (left) predicts the category and 2D segmentation mask of the target instance(*e.g.*, *mug*). The points back-projected from depth channel are filtered by instance mask and processed by 3D-GCN to obtain object points $\mathcal{P}_o$ which are further normalized as $\mathcal{P}$. Our SAR-Net (right) takes both normalized point cloud $\mathcal{P}$ and category-level template point cloud $\mathcal{K}_c$ as inputs to generate deformed template point cloud $\tilde{\mathcal{K}}$ and symmetric point cloud $\tilde{\mathcal{P}}'$, implemented by network $\mathbf{E}_{SA}$ and $\mathbf{E}_{SC}$, respectively. We get points of a coarse shape $\mathcal{G}$ by concatenating and centralizing the $\mathcal{P}$ and $\tilde{\mathcal{P}}'$. From $\mathcal{G}$, the network $\mathbf{E}_{OCS}$ predicts translation offsets $\tilde{\mathcal{V}}$ and normalized size $\tilde{s}$. The final 6D object pose $\{\tilde{R}_o, \tilde{t}_o\}$ and size $\tilde{s}_o$ are recovered by the post-processing stage in Sec. 3.2 and Sec. 3.4.

cient to infer poses of texture-less objects. Recent efforts are made on directly regressing 6D object pose from RGB images by CNN-based architectures, *e.g.*, PoseNet [22] and PoseCNN [63]. DenseFusion [57] introduces a cross-modal feature fusion manner for better aggregating color and depth information from RGB-D images, which infers more accurate objects pose than RGB-only methods. Such a fusion manner is also used in recent COPSE tasks [5, 52]. Another line of works [29,36,38,39,42,50,65] first regress object coordinates or keypoints in 2D images and then recover poses by Perspective-n-Point algorithm [25], *e.g.*, PVNet [39]. Recent approaches like [15, 16] resort to 3D keypoint voting for precise pose estimation. In contrast to these keypoint voting methods [15,16,39], our approach focuses on a more practical setting without relying on exact object 3D models.

Category-Level 6D Object Pose Estimation. Recent COPSE approaches [5, 6, 24, 45, 47, 52, 58] vitally alleviate the limitation of previous instance-level tasks. To handle the intra-class variations, most previous RGB-D methods [5, 24, 52, 58] represent instances of a category into a unified space. Due to significant variations in object appearance, recent methods [6, 30, 47] put more focus on geometric information of the object. StablePose [47] is a depth-based method that analyzes geometric stability of object surface patches for 6D object pose inference. Lin *et al.* [30] skillfully enforce the predicted pose consistency between an implicit pose encoder and an explicit one to supervise the training of the pose encoders and refine the pose prediction during testing. FS-Net [6] extracts shape-based features from point cloud of the target object for pose and size recovery. FS-Net estimates two perpendicular vectors for

rotation decoupling. Compared to FS-Net, the representation of shape alignment transfers the rotation estimation problem into a reconstruction one. This representation has a more intuitive geometric meaning than FS-Net, as it provides visualization of aligned shape. Recent 6D pose trackers [56,60,61] achieve real-time tracking for category-level or novel objects with very good performance. Crucially, these methods have to rely on the good initial object pose and temporal information for the tracking. In contrast, the COPSE task addressed in this paper does not assume such a good initialization existed, and conducts the 6D object pose and size estimation from the single scene image.

Symmetric Correspondence. Symmetric correspondence has been widely adopted in recent works [11, 35, 59]. The reconstruction of symmetric objects has been investigated in [21,64]. Wu *et al.* [62] use latent symmetric properties to disentangle components obtained from a single image. In the field of 6D pose estimation, HybridPose [48] is the first work to take the dense symmetric correspondences of an individual object as the intermediate representation to help the pose estimation. Differently, we fully utilize the symmetric correspondences in the same object category and extend 2D symmetric correspondences onto 3D ones, significantly improving COPSE inference performance.

3. Methodology

Problem Formulation. Given a depth image, segmented mask, and category of the target object, our goal is to estimate the 6D pose and 3D size of the object from its partially observed point cloud. We represent the 6D object pose as a rigid-body homogeneous transformation matrix $\{R_o, t_o\} \in$

SE(3), where 3D rotation $\boldsymbol{R}_o \in$ SO(3) and 3D translation $\boldsymbol{t}_o \in \mathbb{R}^3$. SE(3) and SO(3) indicate the Lie group of 3D rigid transformations and 3D rotation, individually. Finally, the 3D size of the object is formalized as $\boldsymbol{s}_o \in \mathbb{R}^3$.

Overview. We give an overview of our SAR-Net, as in Fig. 3. Our method takes as input an RGB-D image. While RGB images are utilized by Mask-RCNN [14] in pre-processing stage to infer the segmentation mask and category of each instance, our SAR-Net only processes points from depth channel to address the COPSE task. Specifically, the points back-projected from depth channel are filtered by instance mask and processed by 3D segmentation network 3D-GCN [31] to obtain observed point cloud which is further normalized. (Sec. 3.1). The network $\mathbf{E}_{SA}$ is learned to deform the category-level template point cloud to align against the observed point cloud for 3D rotation representation (Sec. 3.2). The symmetric correspondence is encouraged by the network $\mathbf{E}_{SC}$ to help predict the symmetric point cloud and complete the object shape (Sec. 3.3). Finally, the object center and size are learned from the coarse shape by using the network $\mathbf{E}_{OCS}$ (Sec. 3.4).

3.1. Pre-processing of Point Cloud

Processing observed point cloud. Given predicting segmented mask, we obtain the point cloud by back-projecting the masked depth. However, such a point cloud may still contain object and background points given by the imperfect segmentation. Thus, we further send this point cloud into the 3D-GCN [31] to purify the object points $\mathcal{P}_o \in \mathbb{R}^{3 \times N_o}$, where N_o is the number of points in $\mathcal{P}_o$. The 3D segmentation step makes our synthetically-trained model robust against the error from the 2D segmentation pipeline. Furthermore, we have to normalize the original observed point cloud $\mathcal{P}_o$. Particularly, we first calculate the centroid $\boldsymbol{r}_o = \sum \mathcal{P}_o / N_o$ of point cloud and maximum Euclidean distance $d_o = \max\{\|\mathcal{P}_o - \boldsymbol{r}_o\|_2\}$ (scalar factor) relative to its centroid. We then normalize the $\mathcal{P}_o$ to obtain the point cloud $\mathcal{P}$ by $\mathcal{P} = (\mathcal{P}_o - \boldsymbol{r}_o)/d_o$.

Processing category-level template point cloud. Given the 3D template dataset – ShapeNetCore [4], we randomly[2] select one template per category as the *category-level template shape*, which is normalized by scale, translation, and rotation as in [58]. Intuitively, instances of the same category should, at least in principle, have similar shapes as their category-level template shapes [23]. We further sample the category-level template shape into a sparse 3D point cloud $\mathcal{K}_c \in \mathbb{R}^{3 \times N_k}$ by using Farthest Point Sampling (FPS) algorithm [39], where N_k is the number of points.

3.2. Shape Alignment

Given the normalized point cloud $\mathcal{P}$, our model learns the shape similarities among instances of the same cate-

---

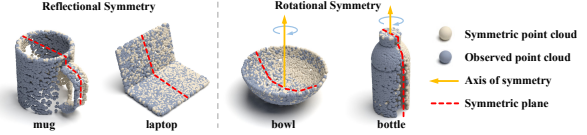[2]SAR-Net is robust to random selection as in Appendix.

Figure 4. Illustration of symmetric point cloud. We generate the ground-truth symmetric point cloud from observed point cloud of objects which have reflectional symmetry or rotational symmetry.

gory, deforming the category-level template point cloud $\mathcal{K}_c$ to align with the observed point cloud $\mathcal{P}$ visually, as demonstrated in Fig. 2 (a). This module always reconstructs the 3D points in space of template point cloud but does not generate 3D points on the surface of observed point cloud, *i.e.*, only transferring the rotational state of the observed point cloud $\mathcal{P}$ to deform the category-level template point cloud $\mathcal{K}_c$. Overall, it requires establishing a parametric encoder-decoder $\mathbf{E}_{SA}$ such that $\tilde{\mathcal{K}} = \mathbf{E}_{SA}(\mathcal{K}_c, \boldsymbol{F}_{\mathcal{P}})$. Then the task of rotation recovery $\tilde{\boldsymbol{R}}_o$ is formulated by the well-known orthogonal Procrustes problem [46] of alignment for two ordered sets of point clouds $\mathcal{K}_c$ and $\tilde{\mathcal{K}}$.

Concretely, our network uses a PointNet-like structure [20, 41] as illustrated in Fig. 3. The normalized point cloud $\mathcal{P}$ and the category-level template point cloud $\mathcal{K}_c$ are fed into $\mathbf{E}_{SA}$ to extract shape-dependent features $\boldsymbol{F}_{\mathcal{P}}$ and $\boldsymbol{F}_{\mathcal{K}}$, respectively. We then concatenate $\boldsymbol{F}_{\mathcal{P}}$ and $\boldsymbol{F}_{\mathcal{K}}$ with each point in $\mathcal{K}_c$ to generate per-point feature embedding, thus performing shape-guided reconstruction of $\tilde{\mathcal{K}}$ under the clues of geometric properties from observed point cloud $\mathcal{P}$. The reconstructed shape $\tilde{\mathcal{K}}$ implicitly encodes the 3D rotation of the observed point cloud $\mathcal{P}$, as $\tilde{\mathcal{K}}$ and $\mathcal{P}$ are enforced aligned by the network. We practially obtain the ground-truth deformed template point cloud $\mathcal{K}$ by applying actual object rotation to the category-level template point cloud $\mathcal{K}_c$. Finally, the object 3D rotation is derived from Umeyama algorithm [54] by solving registration of point clouds $\tilde{\mathcal{K}}$ and $\mathcal{K}_c$.

3.3. Symmetric Correspondence

As most manufactured object categories have a symmetric structure, we thus employ the reflectional and rotational symmetry as an essential geometric cue for the COPSE task. Such the underlying symmetry allows for reasoning the correspondence of potential symmetric point cloud from the observed point cloud. We learn an encoder-decoder structure $\mathbf{E}_{SC}$ as the mapping function, to predict symmetric point cloud $\tilde{\mathcal{P}}' = \mathbf{E}_{SC}(\mathcal{P}, \boldsymbol{F}_{\mathcal{K}})$ from observed point cloud $\mathcal{P}$. $\tilde{\mathcal{P}}' \in \mathbb{R}^{3 \times N_o}$ and $\mathcal{P}$ have the same number of points. Concretely, We concatenate $\boldsymbol{F}_{\mathcal{P}}$ and $\boldsymbol{F}_{\mathcal{K}}$ with each point in $\mathcal{P}$ to generate point-wise feature embedding, and $\mathbf{E}_{SC}$ thus predicts corresponding symmetric point cloud $\tilde{\mathcal{P}}'$ from $\mathcal{P}$.

Reflectional symmetry. As for object categories of reflectional symmetry like *mug* and *laptop*, they are usually symmetric around a fixed plane as shown in Fig. 4. We treat this

symmetry as a constraint of the prior symmetric plane to help complete object shape modeling. Thus, given observed points $\mathcal{P}$, we generate ground-truth symmetric points $\mathcal{P}'$ by flipping $\mathcal{P}$ along the symmetric plane. Thus, we present an encoder-decoder structure $\mathbf{E}_{SC}$ and learn to infer the corresponding symmetric points $\tilde{\mathcal{P}}'$ to be symmetric with $\mathcal{P}$.

Rotational symmetry. Categories of rotational symmetry like *bottle* and *bowl* poss infinite symmetric planes around the axis of symmetry as in Fig. 4, which hinders the network to get converged. One solution is to rotate observed points $\mathcal{P}$ by $180°$ around its axis of symmetry in the object frame for generating ground-truth symmetric points $\mathcal{P}'$, in which case recovers the relatively complete object shape. Thus, the rotational symmetry is simplified as reflectional symmetry. It also enables our network $\mathbf{E}_{SC}$ to infer the occluded part from the observed point cloud to obtain a coarse shape for object center and size estimation. More examples of ground-truth symmetric points refer to Appendix.

Remark. It is noteworthy that specific object instances are never fully symmetric due to shape variations. Thus, exploiting the underlying symmetry by point clouds of objects is applicable to objects which have the global symmetric shapes but asymmetric local parts, as our framework does not demand the exact 3D shape recovery.

3.4. Calculation of Object Center and Size

Furthermore, we concatenate the predicted symmetric point cloud $\tilde{\mathcal{P}}'$ and observed point cloud $\mathcal{P}$ as in Fig 3. This concatenation step generates a coarse 3D object shape $\mathcal{G}' \in \mathcal{R}^{3 \times 2N_o}$ for object center and size estimation. We then centralize points $\mathcal{G}'$ by using the calculated centroid $\boldsymbol{r}$ to get points $\mathcal{G}$, where $\boldsymbol{r} = \sum \mathcal{G}'/2N_o$. We use an encoder-decoder $\mathbf{E}_{OCS}$ to infer the translation offsets $\tilde{\mathcal{V}}$ and normalized size $\tilde{\boldsymbol{s}}$ from points in $\mathcal{G}$, *i.e.*, $(\tilde{\mathcal{V}}, \tilde{\boldsymbol{s}}) = \mathbf{E}_{OCS}(\mathcal{G}, \mathbf{F}_{\mathcal{K}})$. Notably, we incorporate ground-truth symmetric point cloud $\mathcal{P}'$ and the partial points $\mathcal{P}$ to prevent the unstable gradient propagation in the early training stage.

Translation offset learning. Inspired by previous 2D [39, 63] and 3D [15, 16, 40] keypoint voting methods, we treat the object center as a specific keypoint. The encoder-decoder $\mathbf{E}_{OCS}$ infers 3D translation offsets $\tilde{\mathcal{V}} = \{\tilde{\boldsymbol{v}}_i\}_{i=1}^{2N_o}$, where $\tilde{\boldsymbol{v}}_i$ denotes predicted translation offset from each point of $\mathcal{G}$ to the object center. The point cloud of coarse shape $\mathcal{G}$, together with predicted translation offsets $\tilde{\mathcal{V}}$ votes for potential object center $\tilde{\boldsymbol{t}}$. Finally, the voted object center $\tilde{\boldsymbol{t}}_o$ of the observed point cloud $\mathcal{P}_o$ is given as below,

$$\begin{cases} \tilde{\boldsymbol{t}} = \sum(\mathcal{G} + \tilde{\mathcal{V}})/2N_o \\ \tilde{\boldsymbol{t}}_o = (\tilde{\boldsymbol{t}} + \boldsymbol{r}) \cdot d_o + \boldsymbol{r}_o \end{cases} \tag{1}$$

where d_o and $\boldsymbol{r}_o$ are scalar factor and centroid of observed point cloud $\mathcal{P}_o$ as computed in Sec. 3.1.

Size estimation. Obtained concatenated points $\mathcal{G}$, the network $\mathbf{E}_{OCS}$ regresses the normalized size $\tilde{\boldsymbol{s}}$. Then the ac-

tual size $\tilde{\boldsymbol{s}}_o$ of the original point cloud $\mathcal{P}_o$ is recovered by the calculated scalar factor d_o (Sec. 3.1), *i.e.*, $\tilde{\boldsymbol{s}}_o = d_o \cdot \tilde{\boldsymbol{s}}$. Compared to regressing size from partially observed point cloud $\mathcal{P}$, the concatenated point cloud $\mathcal{G}$ provides a coarse shape for more accurate size estimation as discussed in Sec. 4.2.

3.5. Loss Function

We define the loss function $\mathcal{L}$ as follows,

$$\mathcal{L} = \mathcal{L}_{def} + \mathcal{L}_{sym} + \mathcal{L}_{cen} + \mathcal{L}_{size} \tag{2}$$

Deformed point reconstruction loss. Our SAR-Net performs shape-guided reconstruction from observed points. Given ground-truth deformed template points $\mathcal{K} = \{\boldsymbol{k}_i\}_{i=1}^{N_k}$ with N_k points, SAR-Net reconstructs $\tilde{\mathcal{K}} = \{\tilde{\boldsymbol{k}}_i\}_{i=1}^{N_k}$ as:

$$\mathcal{L}_{def} = \frac{1}{N_k} \sum_{i=1}^{N_k} \left\| \boldsymbol{k}_i - \tilde{\boldsymbol{k}}_i \right\|_1 \tag{3}$$

For object with rotational symmetry, we adopt the strategy as in [58]. Refer to Appendix for details.

Symmetric point reconstruction loss. The symmetric correspondence component predicts point-wise symmetric points $\tilde{\mathcal{P}}' = \{\tilde{\boldsymbol{p}}_i'\}_{i=1}^{N_o}$ based on the input observed points $\mathcal{P}$. We optimize the objective as:

$$\mathcal{L}_{sym} = \frac{1}{N_o} \sum_{i=1}^{N_o} \|\boldsymbol{p}_i' - \tilde{\boldsymbol{p}}_i'\|_1 \tag{4}$$

where N_o is the number of points in $\mathcal{P}$; $\boldsymbol{p}_i'$ and $\tilde{\boldsymbol{p}}_i'$ are the ground-truth and predicted symmetric points, respectively. As we simplify the rotational symmetry as a particular case of the reflectional symmetry (Sec. 3.3). Thus, this loss function is also helpful to tackle the case of rotational symmetry.

Translation offset loss. The network learns translation offsets $\{\tilde{\boldsymbol{v}}_i\}_{i=1}^{2N_o}$ from concatenated points to object center. The learning of $\boldsymbol{v}_i$ is supervised by minimizing the loss as:

$$\mathcal{L}_{cen} = \frac{1}{2N_o} \sum_{i=1}^{2N_o} \|\boldsymbol{v}_i - \tilde{\boldsymbol{v}}_i\|_1 \tag{5}$$

where $2N_o$ is the number of concatenated points. The $\boldsymbol{v}_i$ and $\tilde{\boldsymbol{v}}_i$ are the ground-truth and predicted translation offsets. **Size loss.** For better size recovery, we regress the size from the point cloud of a coarse shape $\mathcal{G}$ as discussed in Sec. 3.4. We supervise the size regression as:

$$\mathcal{L}_{size} = \|\tilde{\boldsymbol{s}} - \boldsymbol{s}\|_1 \tag{6}$$

where $\boldsymbol{s}$ and $\tilde{\boldsymbol{s}}$ represent the ground-truth and predicted size of the normalized point cloud $\mathcal{P}$, respectively.

4. Experiments

Datasets. (1) *NOCS Dataset* [58]. It contains six object categories including *bottle*, *bowl*, *camera*, *can*, *laptop*, and *mug*. The NOCS has two parts, *i.e.*, the synthetic part and the real-world one. For the synthetic part, there

are 300K composite images, where 25K are set aside for evaluation(CAMERA25). For the real-world part, it contains 2.75K real-scene images for evaluation(REAL275). (2) *LINEMOD Dataset* [17]. It is a widely used dataset for instance-level object pose estimation. It provides a scanned CAD model for each object. (3) *Additional Real-world Scenes*. Our model is tested on additional 6 different real scenes with 25 unseen instances from categories including *bowl*, *mug*, *bottle*, and *laptop*. The images are captured by a RealSense D435 camera and not manually pose-annotated.

Evaluation Metrics. *(1) Category-level pose and size estimation.* As [52], we compute average precision of 3D Intersection-Over-Union(IoU) at threshold values of 25%, 50% and 75% for 3D object detection. The average precision at $m°ncm$ is calculated for evaluating 6D pose recovery, *i.e.*, the percentage of poses where the translational error is below ncm and the angular error is below $m°$. Here we choose threshold values of $5°2cm$, $5°5cm$, $10°2cm$, and $10°5cm$, respectively. *(2) Instance-level pose estimation.* We use average distance metric ADD [18] for non-symmetric objects and ADD-S [63] for symmetric objects (*e.g.*, *eggbox* and *glue*). The accuracy of average distance less than 10% of the object diameters is reported.

Implementation Details. The architecture of SAR-Net and training details of 3D-GCN are presented in Appendix. We pick object models of six categories from ShapNetCore [4] and utilize the Blender software [1] to render depth images to train our model, denoted as SAR-Net(small). Additionally, we use 275K images from CAMERA dataset to train our model, denoted as SAR-Net. The training data rendered by Blender (∼60K instances) is 10 times less than that of CAMERA dataset (∼600K instances). The back-projected points of instances from synthetic depth images are disorderly sampled into 1024 points. Our SAR-Net is trained for 100 epochs with a batch size of 32 on a single RTX 2080Ti GPU. We initially set the learning rate as 0.0004 and multiply it by a factor of 0.75 every four epoch. We use the segmentation results of Mask-RCNN provided by [30] for fair comparisons. In robotic experiments, our SAR-Net is implemented on the desktop with an NVIDIA RTX2070 GPU, and pose and size estimation takes about 100ms. The model will be released on the repository of Baidu PaddlePaddle.

4.1. Main Results

Category-level NOCS Dataset. We compare our SAR-Net with NOCS [58], CASS [5], SPD [52], FS-Net [6], StablePose [47] and DualPose [30] on the CAMERA25 and REAL275 datasets in Tab. 1. For the synthetic CAMERA25 dataset, our SAR-Net achieves comparable performance to the state-of-the-art method DualPose and shows better performance under the more strict metric $5°2cm$. For the real-world REAL275 dataset, NOCS, CASS, SPD and DualPose use both synthetic data (CAMERA) and real-world data for

training; FS-Net and StablePose only use real-world data. In contrast, our method only uses synthetic data. Surprisingly, even in such a comparison, our SAR-Net outperforms all other baseline methods at all but IoU metric, as FS-Net uses the pre-calculated mean size per category. The results validate the good generalization of SAR-Net in real-world applications. Although SAR-Net(small) is trained by using 10 times less synthetic training data than CAMERA, and it already outperforms all other methods but the DualPose on the REAL275 dataset. It could save memory footprint and reduce the training time in practice. Moreover, for the pose and size estimation part of the model, our SAR-Net has less parameters than other methods. We also qualitatively show some results of our SAR-Net and DualPose [30] in Fig. 5. Our method generates more accurate rotation estimation than DualPose, especially for the *camera* category. More comparison results are shown in Appendix.

Instance-level LINEMOD Dataset. Our COPSE model could be easily employed in the instance-level task by using the exact object model as the template shape. We then generate 30K synthetic depth images for each instance for the model training. Compared with RGB(-D) methods [15, 39] or depth-only method [12, 13], our SAR-Net achieves comparable results in terms of ADD(-S) metric as in Tab. 2. Taking weakly symmetric objects (*e.g.*, *cat*) as symmetric ones, our model still gains desirable performance, which provides evidence to support the utilization of symmetric correspondence to handle those objects which have some parts that are less symmetric.

Additional Real-world Scenarios. For *additional real scenes* with multiple objects, visualization results are shown in Fig. 6 (top row). Our model generates accurate estimation, as the objects are tightly located within predicted bounding boxes. The results indicate the generalization capability of our SAR-Net in real-world applications, in terms of different depth sensors (*i.e.*, Structure Sensor [2] used by NOCS dataset, and RealSense D435 of ours) and different novel instances (18 novel instances from REAL275 and 25 novel ones of ours). See Appendix for more results.

4.2. Ablation Studies

We verify the efficacy of the key components of our SAR-Net on the REAL275 dataset in Tab. 3.

Symmetric Correspondence. We first check the importance of utilizing symmetric correspondence. We start from a basic network, which directly outputs the deformed template point cloud from shape alignment component (SA), translation offsets, and normalized size based on the *partially observed point cloud*, as in row 1. The symmetric correspondence component (SC) is then added as shown in row 2. The comparison results between **rows 1** and **2** illustrate that exploring underlying symmetric correspondence is a vital part of producing overall great results. With symmetry

Table 1. Results on REAL275 and CAMERA25 [58]: comparisons with other COPSE methods.(↑): higher better, (↓): lower better.

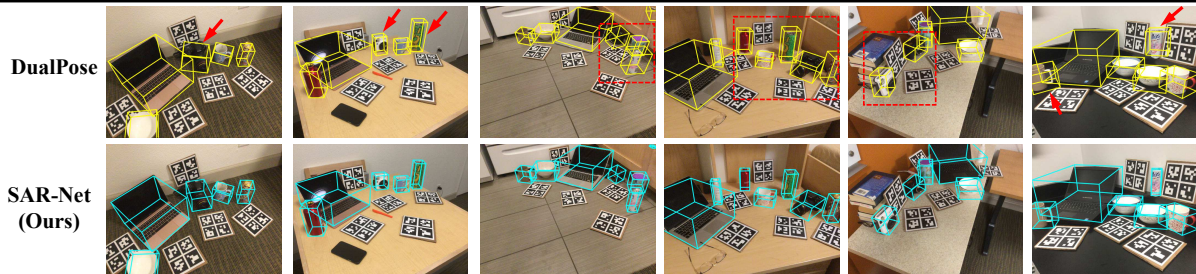| Dataset	Method	mAP (↑)						Accuracy (↑) $5°5cm$	Parameters (↓) (M)
		IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$		
REAL275	NOCS [58]	78.0	30.1	7.2	10.0	13.8	25.2	18.2	-
	CASS [5]	77.7	-	-	23.5	-	58.0	-	47.2
	SPD [52]	77.3	53.2	19.3	21.4	43.2	54.1	30.4	18.3
	FS-Net [6]	**92.2**	63.5	-	28.2	-	60.8	-	41.2
	StablePose [47]	-	-	-	-	-	-	38.8	-
	DualPose [30]	79.8	62.2	29.3	35.9	50.0	66.8	50.1	67.9
	SAR-Net(small)	80.4	**63.7**	24.1	34.8	45.3	67.4	49.1	**6.3**
	SAR-Net	79.3	62.4	**31.6**	**42.3**	**50.3**	**68.3**	**54.9**	**6.3**
CAMERA25	NOCS [58]	83.9	69.5	32.3	40.9	48.2	64.6	49.4	-
	SPD [52]	**93.2**	83.1	54.3	59.0	73.3	81.5	71.5	18.3
	DualPose [30]	92.4	**86.4**	64.7	70.7	**77.2**	**84.7**	79.9	67.9
	SAR-Net(small)	88.1	71.1	44.0	49.4	56.1	65.6	61.7	**6.3**
	SAR-Net	86.8	79.0	**66.7**	**70.9**	75.3	80.3	**81.4**	**6.3**

Figure 5. Qualitative comparisons between our SAR-Net and DualPose [30] on REAL275 dataset [58]. We visualize the estimated 6D pose and size as the tight-oriented bounding box around the target instances.

Table 2. Results on LINEMOD [17]: comparisons with other instance-level methods. 'S' is synthetic data and 'R' is real data.

Training data	Methods	ape	can	cat	driller	eggbox	glue
RGB(S+R)	PVNet [39]	43.6	95.5	79.3	96.4	99.1	95.7
RGBD(S+R)	FFB6D [15]	**98.4**	**99.8**	**99.9**	**100.0**	**100.0**	**100.0**
D(S)	CP(ICP) [13]	58.3	84.7	84.6	43.2	99.5	98.8
D(S)	CAAE [12]	74.5	90.2	90.7	97.3	99.7	93.5
D(S)	SAR-Net	64.5	83.6	91.4	84.0	99.4	**100.0**

inference, the network learns to captures more useful shape characteristics for symmetric points reconstruction and also enhances the performance of other components.

Partial or coarse shape. We then study the performance difference between using point cloud of a coarse shape and the partially observed point cloud. We analyze it from two aspects: (1) We concatenate the observed point cloud and symmetric one to obtain a coarse shape (Concat) for object center and size estimation, as in row 3. Comparing results in **rows 2** and **3**, estimation based on a coarse shape gains overall improved performance versus that relying on partially observed point cloud. (2) We then explore the importance of centralization operation (Centralize) as in row 4, *i.e.*, the concatenated point cloud is further centralized. The comparison results of **rows 3** and **4** indicate the necessity of the centralizing operation, yielding the further improvement of 1.1% at IoU_{75} and 3.6% at $5°2cm$, respectively.

Voting or regression for 3D center estimation. We replace the translation offset learning (Sec. 3.4) by regressing the object center (RegressT) as in row 5. The performance

in **row 5** consistently drop under all metrics, comparing the results in **row 4**. Therefore, the translation offset learning and voting for object center help to locate a more accurate object center than RegressT.

Symmetry-based or direct shape completion. Also, we replace the SC component with direct object shape completion by using the same network (DSComp), supervised by the Chamfer loss [10]. Compared to results in **row 4**, performance of all metrics in **row 6** drop consistently. This is probably because the shape completion focuses on detailed reconstruction, which relies on a more complicated network, but inferring a symmetric point cloud is easier. Poor reconstruction results from a direct object completion network further degrade the performance of the object center and size estimation.

Point number of template point cloud. In addition, We also explore the impact of the varied number of the category-level template point cloud $\mathcal{K}_c$ by using the full COPSE model. It is observed from **rows 4, 7, and 8** that 36 points are a good trade-off for our network to learn. The choice of 128 points degrades the performance due to the larger output space, while the 16 points are too sparse to represent the geometric structure of object, which negatively influences the final performance.

We also conduct ablation studies of various rotation representation on the REAL275 dataset in Tab. 4.

Shape alignment for 3D rotation. We replace the shape alignment (SA) of SAR-Net by other 3D rotation repre-

Table 3. Ablation studies of key components tested on REAL275. 'SA' means shape alignment (Sec 3.2). 'SC' means symmetric correspondence (Sec 3.3). 'Concat' indicates concatenating observed point cloud and symmetric one, and 'Centralize' is the centralization operation (Sec. 3.4). 'RegressT' means directly regressing object center versus predicting translation offsets in Sec. 3.4. 'DSComp' is direct shape completion. 'Point Number' is the number of points in category-level template point cloud. (Sec. 3.1). (↑): higher better.

Row	SA	SC	Concat	Centralize	RegressT	DSComp	Point Number	IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$	Accuracy (↑) $5°5cm$
1	✓						36	81.1	55.3	15.9	23.6	34.9	59.1	39.4
2	✓	✓					36	**81.2**	60.1	17.8	27.7	38.8	63.3	44.1
3	✓	✓	✓				36	80.6	62.6	20.5	31.7	39.8	65.3	46.4
4	✓	✓	✓	✓			36	80.4	**63.7**	**24.1**	**34.8**	45.3	67.4	**49.1**
5	✓	✓	✓	✓	✓		36	81.0	63.5	21.1	30.4	44.9	67.2	46.7
6	✓			✓		✓	36	80.6	59.5	19.2	28.9	41.6	65.2	45.0
7	✓	✓	✓	✓			16	79.6	62.9	22.8	33.0	**46.1**	**67.6**	47.9
8	✓	✓	✓	✓			128	79.5	59.5	21.5	32.1	43.7	66.3	47.5

Table 4. Ablation studies of using different 3D rotation representations tested on the REAL275 dataset. (↑): higher better.

Method	IoU_{50}	IoU_{75}	$5°2cm$	$5°5cm$	$10°2cm$	$10°5cm$	Accuracy (↑) $5°5cm$
quaternion	80.6	62.9	20.8	29.7	43.6	64.6	46.3
SVD [26]	**82.2**	61.8	17.8	24.3	39.6	58.6	40.0
R_{6d} [66]	81.6	**64.1**	21.7	30.5	42.6	64.2	46.6
Vector [6]	81.2	62.5	21.1	31.5	45.1	67.1	47.6
SAR-Net	80.4	63.7	**24.1**	**34.8**	**45.3**	**67.4**	**49.1**

sentation in the form of quaternion, SVD [26], continuity 6D [66] (R_{6d}), and Vector [6], respectively. The comparison results are summarized in Tab. 4. Compared to quaternion, SVD, and R_{6d}, enforced shape alignment enables better generalization, as the point reconstruction search space is smaller than the rotation space, which is easier for the network to learn. The representation of Vector and SA both have geometric meaning, but our SA performs better than Vector, especially under strict metrics of $5°2cm$ and $5°5cm$.

4.3. Robotic Experiments

Physic Baxter Robot. The robotic experiments compare the real-world performance of deploying COPSE models on a real Baxter robot executing different tasks, including object grasping, handover, and pouring as in Fig. 6 (bottom row). Baxter is a dual-arm robot mounted with a RealSense D435 Camera on the base. More configurations of the robotic experiment are detailed in the Appendix.

Grasping Task. Particularly, we use 12 unseen instances from 3 classes, i.e., 4 mugs, 4 bottles, and 4 bowls. Deployed the COPSE models, the robot is programmed to attempt 10 grasps for each object. In this experiment, our SAR-Net is compared against DualPose [30] and SPD [52] with success rates of 88.3%, 80.8% and 65.8%, respectively. The baseline methods often fail due to imprecise rotation estimation or bigger estimated bounding boxes than the exact ones of target instances. See video demo for details.

Object Handover Task. The robot interacts with the actor in this task, trying to grasp the objects in human hands. We choose the testing instance *bottle*. The Baxter successes on
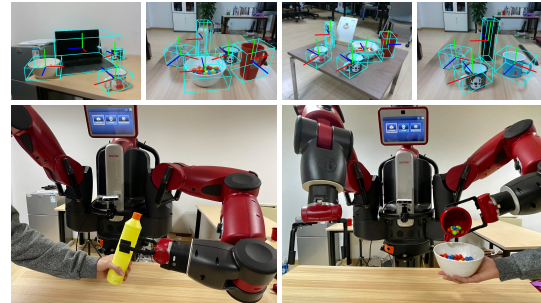

Figure 6. Estimated results given by our SAR-Net in various real clutter environments (top row). We perform tasks on a physical Baxter robot integrated with our SAR-Net (bottom row).

80% using our SAR-Net in 15 trials of the handover task, compared against that of 73.3% of DualPose and 66.7% of SPD, validating the accurate estimation of our SAR-Net.

Pouring Task. Using our COPSE model, we conduct the task of actor moving the bowl, while the robot follows the actor and executes pouring action. We choose each testing instance from *bowl* and *mug*, respectively. The robot is programmed to attempt 15 trials. Our SAR-Net is compared against DualPose and SPD with success rates of 73.3%, 60.0%, and 53.3%, respectively. The results show the efficacy of our COPSE model in robotic experiments.

5. Conclusion

We propose a lightweight geometry-based model for the COPSE task. Our network uses shape alignment to facilitate 3D rotation calculation. The symmetry correspondence of objects is utilized to complete its shape for better object center and 3D size estimation. Our method achieves state-of-the-art performance without real-world training data. Furthermore, a physical Baxter robot integrated with our framework validates the utility in practical robotic applications. However, under the inherent limitation of the depth-based method, the sensor noise and lacked discriminative details may result in ambiguities in pose recovery. Future work will consider fusing additional color information from RGB channels for more accurate pose and size recovery.

References

[1] Blender software. https://www.blender.org/. 6

[2] Structure sensor. https://structure.io/. 6

[3] Zhe Cao, Yaser Sheikh, and Natasha Kholgade Banerjee. Real-time scalable 6dof pose estimation for textureless objects. In *2016 IEEE International conference on Robotics and Automation (ICRA)*, pages 2441–2448. IEEE, 2016. 2

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4, 6

[5] Dengsheng Chen, Jun Li, Zheng Wang, and Kai Xu. Learning canonical shape space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11973–11982, 2020. 1, 3, 6, 7

[6] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, Linlin Shen, and Ales Leonardis. Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1581–1590, 2021. 2, 3, 6, 7, 8

[7] Alvaro Collet, Dmitry Berenson, Siddhartha S Srinivasa, and Dave Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *2009 IEEE International Conference on Robotics and Automation*, pages 48–55. IEEE, 2009. 1

[8] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking. *IEEE Transactions on Robotics*, 2021. 1

[9] Xinke Deng, Yu Xiang, Arsalan Mousavian, Clemens Eppner, Timothy Bretl, and Dieter Fox. Self-supervised 6d object pose estimation for robot manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3665–3671. IEEE, 2020. 1

[10] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 7

[11] Christopher Funk, Seungkyu Lee, Martin R Oswald, Stavros Tsogkas, Wei Shen, Andrea Cohen, Sven Dickinson, and Yanxi Liu. 2017 iccv challenge: Detecting symmetry in the wild. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1692–1701, 2017. 3

[12] Ge Gao, Mikko Lauri, Xiaolin Hu, Jianwei Zhang, and Simone Frintrop. Cloudaae: Learning 6d object pose regression with on-line data synthesis on point clouds. In *ICRA*, 2021. 6, 7

[13] Ge Gao, Mikko Lauri, Yulong Wang, Xiaolin Hu, Jianwei Zhang, and Simone Frintrop. 6d object pose regression via supervised learning on point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3643–3649. IEEE, 2020. 6, 7

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 4

[15] Yisheng He, Haibin Huang, Haoqiang Fan, Qifeng Chen, and Jian Sun. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3003–3013, 2021. 1, 3, 5, 6, 7

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11632–11641, 2020. 3, 5

[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 *Asian conference on computer vision*, pages 548–562. Springer, 2012. 2, 6, 7

[18] 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 *Asian conference on computer vision*, pages 548–562. Springer, 2012. 6

[19] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2930–2939, 2020. 1

[20] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 4

[21] Jarmo Ilonen, Jeannette Bohg, and Ville Kyrki. Three-dimensional object reconstruction of symmetric objects by fusing visual and tactile sensing. *The International Journal of Robotics Research*, 33(2):321–341, 2014. 3

[22] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 3

[23] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2202–2211, 2019. 4

[24] Taeyeop Lee, Byeong-Uk Lee, Myungchul Kim, and In So Kweon. Category-level metric scale object shape and pose estimation. *IEEE Robotics and Automation Letters*, 6(4):8575–8582, 2021. 1, 3

[25] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. 3

[26] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. *arXiv preprint arXiv:2006.14616*, 2020. 8

[27] Yan Li, Leon Gu, and Takeo Kanade. Robustly aligning a shape model and its application to car alignment of unknown

pose. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1860–1876, 2011. 2

[28] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018. 1

[29] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7678–7687, 2019. 3

[30] Jiehong Lin, Zewei Wei, Zhihao Li, Songcen Xu, Kui Jia, and Yuanqing Li. Dualposenet: Category-level 6d object pose and size estimation using dual pose network with refined learning of pose consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3560–3569, October 2021. 1, 3, 6, 7, 8

[31] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1800–1809, 2020. 2, 4

[32] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999. 2

[33] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019. 1

[34] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015. 1

[35] Rajendra Nagar and Shanmuganathan Raman. Reflection symmetry axes detection using multiple model fitting. *IEEE Signal Processing Letters*, 24(10):1438–1442, 2017. 3

[36] Markus Oberweger, Mahdi Rad, and Vincent Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018. 3

[37] Kiru Park, Timothy Patten, and Markus Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7668–7677, 2019. 1

[38] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2011–2018. IEEE, 2017. 3

[39] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019. 1, 3, 4, 5, 6, 7

[40] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 5

[41] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 4

[42] Mahdi Rad and Vincent Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3828–3836, 2017. 3

[43] Caner Sahin, Guillermo Garcia-Hernando, Juil Sock, and Tae-Kyun Kim. Instance-and category-level 6d object pose estimation. In *RGB-D Image Analysis and Processing*, pages 243–265. Springer, 2019. 1

[44] Caner Sahin, Guillermo Garcia-Hernando, Juil Sock, and Tae-Kyun Kim. A review on object pose recovery: from 3d bounding box detectors to full 6d pose estimators. *Image and Vision Computing*, page 103898, 2020. 1

[45] Caner Sahin and Tae-Kyun Kim. Category-level 6d object pose recovery in depth images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 1, 3

[46] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 2, 4

[47] Yifei Shi, Junwen Huang, Xin Xu, Yifan Zhang, and Kai Xu. Stablepose: Learning 6d object poses from geometrically stable patches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15222–15231, 2021. 2, 3, 6, 7

[48] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 431–440, 2020. 3

[49] Zhiqiang Sui, Zheming Zhou, Zhen Zeng, and Odest Chadwicke Jenkins. Sum: Sequential scene understanding and manipulation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3281–3288. IEEE, 2017. 1

[50] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018. 3

[51] Zhou Teng and Jing Xiao. Surface-based detection and 6-dof pose estimation of 3-d objects in cluttered scenes. *IEEE Transactions on Robotics*, 32(6):1347–1361, 2016. 1

[52] Meng Tian, Marcelo H Ang Jr, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. *arXiv preprint arXiv:2007.08454*, 2020. 1, 3, 6, 7, 8

[53] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018. 1

[54] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. 2, 4

[55] Kentaro Wada, Edgar Sucar, Stephen James, Daniel Lenton, and Andrew J Davison. Morefusion: Multi-object reasoning for 6d pose estimation from volumetric fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14540–14549, 2020. 1

[56] Chen Wang, Roberto Martín-Martín, Danfei Xu, Jun Lv, Cewu Lu, Li Fei-Fei, Silvio Savarese, and Yuke Zhu. 6-pack: Category-level 6d pose tracker with anchor-based keypoints. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10059–10066. IEEE, 2020. 3

[57] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. 1, 3

[58] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 1, 2, 3, 4, 5, 6, 7

[59] Zhaozhong Wang, Zesheng Tang, and Xiao Zhang. Reflection symmetry detection using locally affine invariant edge correspondence. *IEEE Transactions on Image Processing*, 24(4):1297–1301, 2015. 3

[60] B Wen and Kostas E Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021. 3

[61] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J. Guibas. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 13209–13218, October 2021. 3

[62] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020. 2, 3

[63] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 3, 5, 6

[64] Tianfan Xue, Jianzhuang Liu, and Xiaoou Tang. Symmetric piecewise planar object reconstruction from a single image. In *CVPR 2011*, pages 2577–2584. IEEE, 2011. 3

[65] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1941–1950, 2019. 3

[66] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 8