ASM-Net: Category-level Pose and Shape Estimation Using Parametric Deformation

Shuichi Akizuki
s-akizuki@sist.chukyo-u.ac.jp
Manabu Hashimoto
mana@isl.sist.chukyo-u.ac.jp

Graduate School of Engineering
Chukyo University
Aichi, Japan

Abstract

We propose a novel deep neural network that estimates the six degrees of freedom pose and complete shape of unseen objects from point cloud data. Our concept is to train the network that can perform well on real images captured by a consumer RGBD camera using only 3D models of the target category. To do so, we have employed two ideas. The first is modeling intra-category shape variations with active shape models that can deform the shape with a few dimensional parameters. The second is applying effective filtering processes to the training data to convert the 3D object model into a point cloud that simulates the sensor measurements. We evaluated our method on NOCS REAL275, a widely used benchmark dataset for category-level pose estimation, and confirmed its superiority over conventional methods in terms of both shape recovery and pose estimation. Our code is available at https://github.com/sakizuki/asm-net.

1 Introduction

Estimating the six degrees of freedom (6DoF) pose of objects in a scene is a fundamental technique in robotic manipulation [21, 30], scene understanding [24], and augmented reality [25]. If a 3D model of the object represented as a mesh or point cloud data is available, it can be solved in an approach called instance-level pose estimation [10, 11, 14, 19, 32].

However, even if the name of the target object is known, there are cases in which the 3D model cannot be accessed. Examples are objects whose 3D models are not publicly available, objects manufactured without 3D modeling design, and objects with individual differences in shape, such as food. Instance-level pose estimation methods cannot deal with intra-category shape variations because they assume that rigid body transformations are the only variations that occur in the object.

Pose estimation that also considers intra-category shape variations is called category-level pose estimation. To our knowledge, the first paper on this subject was that of [18]. In this method, the shape of the object is composed of a deformable shape model of semantic keypoints, and the alignment error with the keypoints detected in the image is minimized by iteratively optimizing the pose and shape deformation.

Recently, from the viewpoint of inference speed, the mainstream approach to estimation has been the forward processing of neural networks. Wang et al. [31] presented a shape

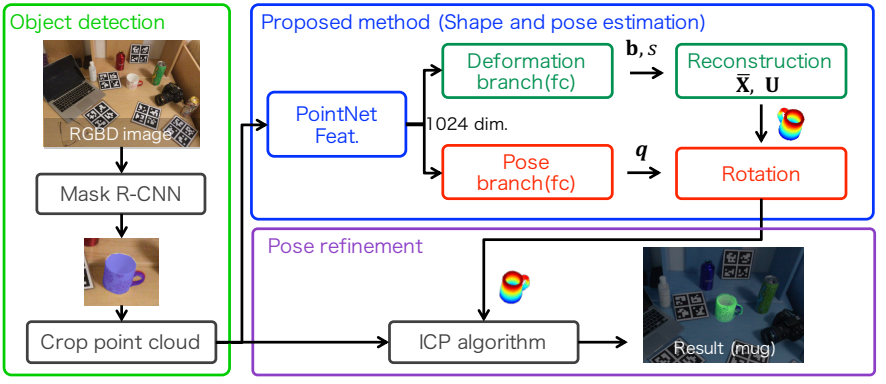


Figure 1: Overview of our method. We first detect the object mask for each instance. This figure shows an example of a mug as an instance. Next, from the point cloud of the object mask, we estimate the parameters for recovering the full 3D shape and the pose. Finally, the final output is obtained by refining the pose.

representation (normalized object coordinate space [NOCS]) that shows the object shape on a canonical coordinate system prepared for each category, and then proposed a pose estimation method using NOCS. By estimating the NOCS map, a 2D projection of NOCS, from the input scene, the method obtains a dense correspondence between the NOCS space and the input scene. Then, the pose and size are computed using the Umeyama algorithm [29], which is a closed-form solution for pose and scale estimation. Tian et al. [26] proposed a method for recovering not only the pose and size of an object but also its full 3D shape by representing the shape of the object as the sum of a shape prior and a deformation field that represents the amount of deformation at each point.

However, to train the previously proposed category-level pose estimation methods, a large number of annotations of NOCS maps and poses on real images are required. In this study, we propose a method to reduce the learning cost of category-level pose estimation using a point cloud-based networks that can be learned from synthetically generated data. Our network estimates the complete 3D shape and 6DoF pose of target objects. It only requires 3D object models from the target category composed of point clouds; real images are not required for training. The idea of the proposed method is expressed in the following two points:

1. We model intra-category shape variation with the modified Active Shape Model (ASM), which is a deformable shape representation with a small number of coefficients. This allows us to learn shape recovery easily.

2. During the training phase, we generate annotated 3D object models online by controlling both pose and deformation parameters. We also propose a filtering process that transforms the 3D object model into a point cloud that simulates observations using 3D sensors.

The performance of the proposed method was evaluated on NOCS REAL275, a widely used benchmark dataset for category-level pose estimation. In comparison with other methods, our method has advantage in terms of both shape recovery and pose estimation.

2 Related Works

2.1 Instance-level pose estimation

This approach uses a known 3D object model for pose estimation. There are two approaches—to calculate the pose based on the correspondence of local features, and the other is to learn the relationship between appearance and pose.

In the former approach, features are extracted from the local regions of the 3D object model or RGBD images. Some local features are described as the surface normal and shape distribution [22, 28]. The correspondence points obtained with feature matching are used for hough voting [27] a process to estimate pose while eliminating the failure correspondences. Recently, this process has been replaced by deep learning [10, 11, 19]. The 3D–2D correspondence is calculated by regressing the keypoint positions in the image, and the pose is estimated using a PnP algorithm.

The latter approach is based on learning the relationship between deep features extracted from the appearance of objects in RGB(D) images and their poses [14, 32].

However, they assume that the shape of the objects in the scene and that of the 3D models are the same, so they cannot deal with intra-category shape variations. Our method can handle unseen objects because it estimates not only the pose of the object but also the parameters for shape recovery.

2.2 Category-level pose estimation

The challenge of category-level pose estimation is that it must deal with differences in the shape and texture of objects in the same category. Chen et al. [6] proposed a method to learn texture variations using neural image synthesis. Other methods focus on learning shape variations. Pavlakos et al. [18] proposed a method to estimate the object shape by representing it as a parameter and optimizing it alternately with the pose. Wang et al. [31] proposed a method to represent object shape variations as a unit cube coordinate system (NOCS). 2D projection of the NOCS is estimated from RGB images. CASS [4] and Shape Prior [26] are methods used to learn deformations from the canonical shape. An algorithm was also proposed to learn feature representations unaffected by texture by using only the shape as input to the network [5].

However, all these methods require the annotation of input images, such as the 6DoF pose and NOCS maps, during training. The cost of preparing training data is also considerable. Although self-supervised method for learning pose estimation from unlabeled RGBD images has recently been proposed [16], training on real images is still required to achieve better performance. The advantage of our method is that it can train category-level pose estimation only by preparing 3D models of the target category. Our method can achieve equivalent or better performance than other methods without using real images for the training phase.

2.3 Shape deformation

We discuss deformable shape models in terms of the number of parameters required for shape deformation. A method for representing the shape of a target category by a weighted sum of N 3D object models was proposed [23]. The number of deformation parameters must be proportional to the number of objects N. The model proposed by Tian et al. [26] composed of the sum of the shape prior, which is the mean shape of the category, and the deformation

field, which represents the amount of deformation at each point. If the number of vertices of an object is M, there are $3M$ parameters.

An approach to approximate the shape using multiple 3D Gaussian mixture models has been proposed [33] to represent the shape with fewer parameters than the original number of vertices. In FS-Net [5], a recently proposed categorical-level pose estimation, a method of deforming the object shape (box-cage deformation) is proposed as data augmentation during model training. This method deforms shapes along the three coordinate axes. However, it only controls the stretching and shrinking of each axis, so it cannot change the design of the object. In addition, an conditional Generative Adversarial Nets approach [13] was proposed to generate images from two conditions: pose and shape.

The ASM [7] is a well-known method for representing variations in the shape of an object. This method compresses the semantically equivalent points of the same category objects using Principal Component Analysis (PCA), and it represents the shape variations as a weighted sum of the average shape and K eigenvectors. Methods for the modeling of textures and 3D shapes [7, 8] and for application to robotic manipulation have been proposed [21].

As the upper eigenvectors represent the deformations that characterize the category, intra-category variations can be represented with K coefficients, which is sufficiently small compared with N or M. In this study, we model shape variations with $K+1$ parameters by adding a parameter that represents scaling to the ASM.

3 Proposed method

Our method estimates the full 3D shape and 6DoF pose of the target category from a monocular RGBD image. The proposed method consists of object detection, shape and pose estimation, and pose refinement. Fig.1 shows the proposed pipeline.

For object detection, the module detects the mask of the object from the input RGB image. We use an off-the-shelf network (e.g., Mask R-CNN [9]) for mask detection. For this task, a large RGB image dataset [15], which allows us to learn a reliable detector, is available. We assume that RGB information has less relevance to object shape, so depth information is more important for estimating shape and pose. Therefore, we recover the point cloud data from the object mask and the depth image. Because of errors in the camera's intrinsic parameters, the points on the background surface may be recovered at the same time. These point clouds are far away from the object, to remove them the outlier removal by statistical analysis is applied to the depth image.

For shape and pose estimation, we input the point cloud of the object and estimate the deformation parameters and pose required to recover the full 3D shape of the object. First, the point cloud is inputted to the feature extractor of PointNet [20], and 1,024-dimensional global feature vectors are extracted. Deformation and pose (unit-quaternion) parameters are estimated by inputting these features into the deformation branch and pose branch, which are composed of fully connected layers. The deformation parameters, the mean shape, and the eigenvectors of the target category are used to recover the shape, which is then rotated according to the estimated pose. The recovered point cloud is translated to the position where the object is cropped.

For pose refinement, the final 6DoF pose is obtained by applying the ICP algorithm [1] as post-processing. The lower-right area of Fig.1 shows an example of the recognition result of the mug category. The shape of the recovered point cloud is similar to that of the mug in

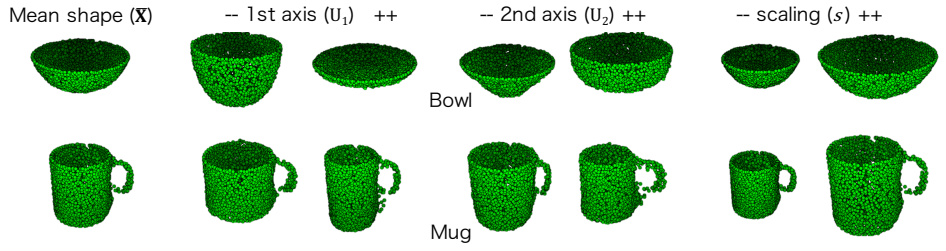


Figure 2: Shape deformation using the our ASM. From left to right are the mean shape, the change in the direction of the first and second eigenvectors, and scaling. The bowl's height and roundness were changed, and in the mug, the height of the cylindrical part and the shape of the handle were deformed.

the image.

3.1 Modeling of intra-category shape variation

To model the intra-category shape variation, we leverage ASMs, which are a method of representing an object's shape, described by the sum of the mean shape and weighted eigenvectors. The shape can be deformed by adjusting the weighting coefficients. Unlike the facial recognition task, for which ASM has been used in the past, the object recognition task involves a large variation in the size of the object. Therefore, we propose an extended version of ASM that can directly represent the scaling using an additional coefficient. We first represent the point cloud as a $3M$-dimensional vector. Then, we perform PCA to obtain the eigenvectors $\mathbf{U}$. Our ASM is denoted as a function of $\mathbf{b} = (b_1, \ldots, b_K)$ and s, as follows:

$$S(\mathbf{b}, s) = s(\bar{\mathbf{X}} + \sum_{i=1}^{K} b_i \mathbf{U}_i) \tag{1}$$

By adjusting $\mathbf{b}$ and s, we can deform the shape while keeping the characteristics of the category. Fig.2 shows the deformation of the category bowl and mug by our ASM.

To construct our ASM, we need (1) 3D point clouds with the same number of points, (2) a normalized pose, and (3) semantically identical point correspondences. For (1), M points are sampled from all object models. For (2), since the object models registered in ShapeNetCore [3] are pose normalized, so they can be used without modification. For (3), we use the non-rigid registration method [17] to obtain a semantically sorted point cloud. We specify one object as the source and apply non-rigid registration to all remaining objects (targets). The order is sorted by considering the nearest neighbor points of the deformed source and target as corresponding points.

3.2 Learning of deformation and pose from synthetic data

In this section, we describe a procedure to train shape deformation and pose estimation using synthetically generated point clouds. We only need 3D object models of the target category to train the proposed method.

While generating the training data online, we optimize the network parameters of the PointNet feature extractor, the deformation branch, and the pose branch, as described in

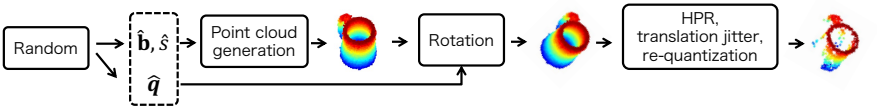


Figure 3: Procedure of training data generation.

Fig.1. To do so, we randomly generate the deformation parameters $\hat{\mathbf{b}}$ and $\hat{s}$, as well as the unit quaternion of the pose $\hat{\mathbf{q}}$, and the point cloud is also generated from deformation parameters and eigenvectors. Fig. 3 illustrates this procedure. Considering the illumination variation is necessary when dealing with RGB images, simulating it accurately is not easy because of the influence of the position of the light source and the reflections from surrounding objects. However, shape data have the advantage of being less sensitive to illumination variation. As our method only uses shape data for inference, simulating real-world data variations in a low-cost manner is possible.

Specifically, three filtering processes are applied to the generated point cloud. First, we use Hidden Point Removal (HPR) [12] for back surface removal to simulate observation by a 3D sensor. Second, we add translation jitter and Gaussian noise as data augmentation. Third, we perform re-quantization in the depth direction. Thus, we ensure that our point clouds are consistent with those recovered from RGBD images, which have a low resolution in the depth direction compared with their resolution in the x and y directions.

3.3 Loss function

The network parameters are optimized by minimizing loss, defined in Eq. (2). $\mathcal{L}_{deform}$ and $\mathcal{L}_{pose}$ are the loss to shape deformation and pose, respectively.

$$\mathcal{L} = \mathcal{L}_{deform} + \mathcal{L}_{pose} + r \tag{2}$$

$$\mathcal{L}_{deform} = D_{CD}(S(\hat{\mathbf{b}}, \hat{s}), S(\mathbf{b}, s)), \qquad \mathcal{L}_{pose} = D_{CD}(R(M, \hat{\mathbf{q}}), R(M, \mathbf{q})) \tag{3}$$

where $R(M, \mathbf{q})$ is a function that rotates the object model M according to the pose $\mathbf{q}$. By evaluating the consistency between the rotated shapes rather than the difference in the estimated rotation parameters themselves, we can calculate the loss without the ambiguity of the symmetric object pose. The regularization term r brings the estimated scale closer to a value near 1.0. The function D_{CD} is the Chamfer distance (CD) defined by

$$D_{CD}(X, Y) = \sum_{x \in X} \min_{y \in Y} ||x - y||_2^2 + \sum_{y \in Y} \min_{x \in X} ||y - x||_2^2. \tag{4}$$

4 Experiments

4.1 Datasets

To benchmark the pose estimation performance, we used NOCS-REAL275, a dataset for category-level pose estimation proposed in [31]. This dataset consists of 2,750 RGBD images with multiple objects in six categories to be recognized: bottle, bowl, can, camera,

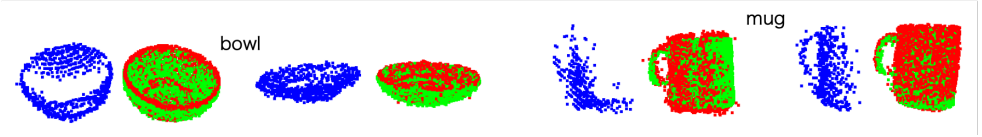


Figure 4: Examples of shape recovery. The input point cloud, recovered result, and ground truth are shown in blue, green, and red, respectively.

laptop, and mug. The 6DoF pose and the 3D bounding box size of each object are provided as the ground truth.

Only the object model is required to train the proposed method. We trained the six categories provided in ShapeNetCore [3]. As some object models in ShapeNetCore are not suitable to be considered as the same category, we removed them. For example, we excluded models in the mug class, whereas saucers and spoons were modeled together. We used 80% of the object models for training and 10% for testing and validation, respectively.

4.2 Evaluation metrics

The proposed method performs shape recovery and pose estimation. Each was evaluated using the following metrics.

Shape reconstruction: We use the CD described in Eq. 4 as a measure to evaluate the accuracy of shape recovery. The smaller the value, the more accurate the reconstruction.

Pose estimation: We use the average precision considering following terms:

- IoU_x is the intersection-over-Union (IoU) accuracy of the 3D bounding box surrounding the object. x is the overlap threshold, and the larger the threshold, the higher the accuracy.
- n°, m cm represent the error of pose estimation. Rotation and translation error are less than n° and m cm are acceptable.

4.3 Shape reconstruction

The shape recovery performance of the proposed method is evaluated from both qualitative and quantitative perspectives. The experiment was conducted on a test set selected from ShapeNetCore and NOCS-REAL275.

Qualitative evaluation: Fig. 4 shows the results of shape reconstruction for bowl and mug of the ShapeNetCore dataset. On the left is the input point cloud, and on the right is the ground truth (red) and the estimation result (green).

For the bowl, the instances with different heights are the ground truth, and the proposed method recovers the point cloud that overlaps with them with good accuracy. For the mug, the instances with different heights of handles and cylinders are the ground truth, and the proposed method recovers the point cloud that almost matches them.

Quantitative evaluation: We first evaluated the similarity between the recovered point cloud and the ground truth on the ShapeNetCore dataset. We compared our method to the mean shape of the training set shown in Fig. 5(a). The evaluation metric is the CD, and the results are shown in Table 1. As we used a size-normalized 3D model, the recovery error does not have a unit of length; the improved amount of the proposed method compared to

Table 1: Chamfer distance of shape reconstruction on ShapeNetCore

	bottle	bowl	camera	can	laptop	mug	mean
Mean shape	4.42	2.44	8.07	2.77	1.31	1.38	3.40
Ours	**1.48**	**1.14**	**2.97**	**1.32**	**1.05**	**1.25**	**1.54**

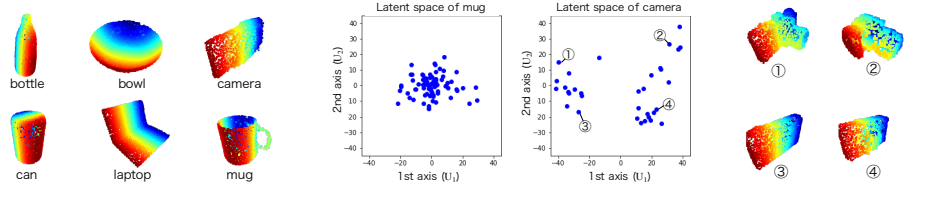
Table 2: Chamfer distance of shape reconstruction($\times 10^{-3}$) on NOCS-REAL275 dataset. "pts." means the number of reconstructed points.

	pts.	bottle	bowl	camera	can	laptop	mug	mean
Shape Prior [26]	1024	3.44	1.21	8.89	1.56	2.91	1.02	3.17
CASS [4]	500	0.75	0.38	0.77	0.42	3.37	0.32	1.06
Mean shape	2893	0.18	0.23	1.19	0.59	2.95	0.14	0.88
Ours	2893	0.23	0.06	0.61	0.15	0.60	0.10	0.29

the mean shape is our contribution. In all categories, it was confirmed that the deformation parameters estimated by the proposed method were closer to the ground truth.

We also evaluated reconstruction accuracy on the NOCS-REAL275 dataset. The results are shown in Table 2. Compared to the mean shape, the proposed method shows improvements, which indicates the effect of our deformations. The proposed method also has a smaller CD metric than all other methods. However, it should be noted that this CD metric tends to be smaller as more points are recovered. The table also shows the number of points recovered by each method.

The error of category camera is larger than that of the other categories. Fig.5(b) shows the latent space consisting of the top two axes of two categories, mug and camera, in which one point corresponds to one instance. In the example of the mug category, the plots are distributed around the origin, and the data spread out from there. However, in the camera category, there is no plot near the origin, and the plots are separated. The mean shape at the origin in latent space is a thin rectangular shape, as shown in Fig.5(a), which is different from the shape of a real camera. Four typical instances are shown in Fig.5(c). The training data included voluminous shapes, such as a single-lens reflex camera, and simple shapes, such as a compact digital camera. Thus, if the dataset contains shapes with different characteristics, the latent space will contain shapes that are different from the actual ones. This is the reason why the accuracy of shape recovery is low in this category.



(a) (b) (c)

Figure 5: Relationship of shapes and latent spaces. (a) Mean shape of each category. (b) Latent space of the mug and camera. (c) Typical instances of camera.

Table 3: Relationship between the dimension of the deformation vector and recognition performance. When $K = 0$, the recognition result is obtained by the mean shapes.

K	0	1	3	5	10
IoU_{50}	57.3	61.1	65.5	60.2	66.0
IoU_{75}	24.0	33.3	39.2	35.5	33.8
5°, 5cm	15.9	23.4	21.3	23.0	16.0
10°, 5cm	36.2	43.8	42.1	43.9	30.5

Table 4: Effectiveness of three filters used for training. H: Hidden point removal, Z:depth re-quantization, T: translation jitter.

Filters	H	H,Z	H,Z,T
IoU_{50}	56.1	68.3	65.5
IoU_{75}	33.8	37.4	39.2
5°, 5cm	24.0	25.6	21.3
10°, 5cm	38.2	43.7	42.1

4.4 Number of dimensions for shape deformation

The performance of the proposed method depends on the number of dimensions K used for shape deformation in Eq.1. To investigate the appropriate number of dimensions, we prepared a network trained with a different number of dimensions and examined its recognition performance on NOCS-REAL275. The results are shown in Table 3.

In the case of $K = 0$, no deformation is applied, and the mean shape is used for pose estimation. The performance was lower than that in the other conditions. For K values around $1-5$, the performance was relatively high because only the top eigenvectors that represent the intra-category shape variations are used for deformation. For $K = 10$, this is the case in which the lower eigenvectors are also used for deformation, as the lower eigenvectors are deformed with low commonality in the category. In this case, the improvement in recognition performance was less, and the performance was closer to that of the $K = 0$ case.

The mean cumulative contribution ratio over the categories are 0.31, 0.53, 0.61, and 0.72, for $K = 0, 1, 3, 5, 10$. The lower eigenvectors do not represent a common deformation for multiple instances, so there is no significant advantage to learning it. In the following experiments, we used $K = 3$.

4.5 Ablation study

We conducted an ablation study to confirm the effectiveness of the three filtering processes described in 3.2 to simulate the observations by 3D sensors. The relationship between the three processes and the recognition performance is shown in the Table 4. Each column indicates the filter process used: H for HPR, Z for depth re-quantization, and T for translation jitter and Gaussian noise. We confirmed that the recognition performance was improved by applying the HPR and depth re-quantization. As the proposed method inputs a point cloud that is offset to its center, the translation jitter may have caused the performance decrease.

4.6 Category-level pose estimation

We evaluated the performance of category-level pose estimation. The comparison methods are NOCS [31], CASS [4], Shape-Prior [26], FS-Net [6], and CPS++ [16]. The first four methods require annotations on real images during training, whereas CPS++ is a self-supervised learning method. The results are shown in Table 5, where the results of CPS++ and the proposed method are shown before and after the pose refinement by ICP. For the IoU metric, the comparison method was superior, but for the IoU_{75} metric, the proposed method was superior to the NOCS. The $n°$, m cm metric is more important because it can directly

Table 5: Pose estimation result of the NOCS-REAL275 dataset with different metrics. $*$ indicates the result of the 10°, 10cm metrics.

Method	Real Data	IoU_{50}	IoU_{75}	5°, 5cm	10°, 5cm
NOCS [31]	✓	80.5	30.1	9.5	26.7
CASS [4]	✓	77.7	–	23.5	58.0
Shape Prior [26]	✓	77.3	53.2	21.4	54.1
FS-Net [5]	✓	92.2	63.5	28.2	60.8
CPS++ w/o ICP [16]	✓	17.7	–	–	22.3*
CPS++ w/ ICP [16]	✓	72.8	–	–	58.6*
Ours w/o ICP	–	64.4	31.7	12.4	37.7
Ours w/ ICP	–	68.3	37.4	25.6	43.7

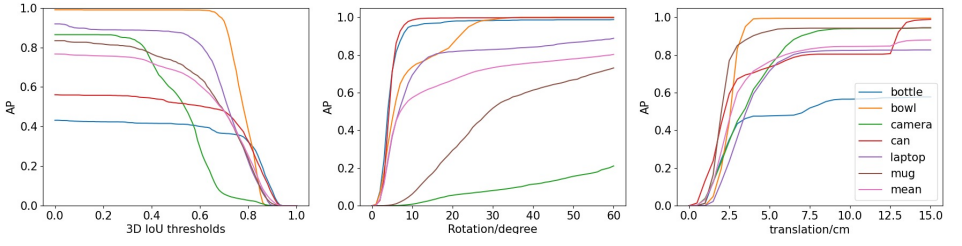


Figure 6: Result on NOCS-REAL275

measure pose accuracy. The proposed method outperformed NOCS, CASS, and Shape-Prior in the most challenging indices, the 5° ,5cm metric.

It has been reported that the performance of CPS++ can be significantly improved by applying ICP [16], but it has a problem that the results depend on the parameter settings of ICP. It is more important to compare the pose accuracy before ICP refinement. The proposed method is superior for both IoU_{50} and 10°, 5cm metrics. The details of the IoU, rotation, and translation error are shown in the Fig.6.

5 Conclusion

We proposed a category-level object pose estimation that learns from synthetically generated point clouds. Our idea is twofold—representing intra-category shape variation using a deformable shape model with a few parameters and transforming the 3D object model using filtering processes that reproduce the characteristics of the depth sensor's measurement. This allows us to generate point cloud data with annotations of deformation and pose parameters online. The advantage of the proposed method is that it does not require any manual annotation; only the 3D object models of the target category are needed for training. In future work, we will tackle the robotic manipulation of unseen objects using our method.

Acknowledgement This work was supported by JSPS KAKENHI Grant Number 21K17834.

References

[1] Paul J Besl and Neil D McKay. Method for Registration of 3-D Shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[2] Volker Blanz and Thomas Vetter. Face Recognition based on Fitting a 3D Morphable Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

[3] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical report, 2015.

[4] Dengsheng Chen, Jun Li, and Kai Xu. Learning Canonical Shape Space for Category-Level 6D Object Pose and Size Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] Wei Chen, Xi Jia, Hyung Jin Chang, Jinming Duan, Shen Linlin, and Ales Leonardis. FS-Net: Fast Shape-based Network for Category-Level 6D Object Pose Estimation with Decoupled Rotation Mechanism. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1581–1590, June 2021.

[6] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category Level Object Pose Estimation via Neural Analysis-by-Synthesis. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 139–156, 2020.

[7] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, 1995.

[8] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[10] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11632–11641, 2020.

[11] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6D Object Pose Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3385–3394, 2019.

[12] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct Visibility of Point Sets. *ACM Transactions on Graphics*, 26(3):24–es, July 2007. ISSN 0730-0301.

[13] Yasutomo Kawanishi, Daisuke Deguchi, Ichiro Ide, and Hiroshi Murase. Ω-GAN: Object Manifold Embedding GAN for Image Generation by Disentangling Parameters into Pose and Shape Manifolds. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 7945–7952, 2021.

[14] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1530–1538, 2017.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.

[16] Fabian Manhardt, Gu Wang, Benjamin Busam, Manuel Nickel, Sven Meier, Luca Minciullo, Xiangyang Ji, and Nassir Navab. CPS++: Improving Class-level 6D Pose and Shape Estimation From Monocular Images With Self-Supervised Learning, 2020.

[17] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[18] Georgios Pavlakos, Xiaowei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-DoF Object Pose from Semantic Keypoints. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[19] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4561–4570, 2019.

[20] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 652–660, 2017.

[21] Diego Rodriguez, Corbin Cogswell, Seongyong Koo, and Sven Behnke. Transferring Grasping Skills to Novel Instances by Latent Space Non-rigid Registration. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 4229–4236, 2018.

[22] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3212–3217, 2009.

[23] Jingnan Shi, Heng Yang, and Luca Carlone. Optimal Pose and Shape Estimation for Category-level 3D Object Perception, 2021.

[24] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015.

[25] David Joseph Tan, Nassir Navab, and Federico Tombari. Looking Beyond the Simple Scenarios: Combining Learners and Optimizers in 3D Temporal Tracking. *IEEE Transactions on Visualization and Computer Graphics*, 23(11):2399–2409, 2017.

[26] Meng Tian, Marcelo H Ang, and Gim Hee Lee. Shape Prior Deformation for Categorical 6D Object Pose and Size Estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 530–546, 2020.

[27] Federico Tombari and Luigi Di Stefano. Object Recognition in 3D Scenes with Occlusions and Clutter by Hough Voting. In *Pacific-Rim Symposium on Image and Video Technology*, pages 349–355, 2010.

[28] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique Signatures of Histograms for Local Surface Description. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 356–369, 2010.

[29] 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.

[30] 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 *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 10059–10066, 2020.

[31] 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 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2642–2651, 2019.

[32] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics Science and Systems (RSS)*, 2018.

[33] Kohei Yamashita, Shohei Nobuhara, and Ko Nishino. 3D-GMNet: Single-View 3D Shape Recovery as A Gaussian Mixture. In *Proceedings of British Machine Vision Conference (BMVC)*, 2020.