2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Median-shape Representation Learning for Category-level Object Pose Estimation in Cluttered Environments

Hiroki Tatemichi*, Yasutomo Kawanishi*, Daisuke Deguchi*, Ichiro Ide*, Ayako Amma† and Hiroshi Murase*
* Nagoya University, Nagoya, Japan. Email: tatemichih@murase.is.i.nagoya-u.ac.jp
† Toyota Motor Corporation, Toyota, Japan. Email: ayako_amma@mail.toyota.co.jp

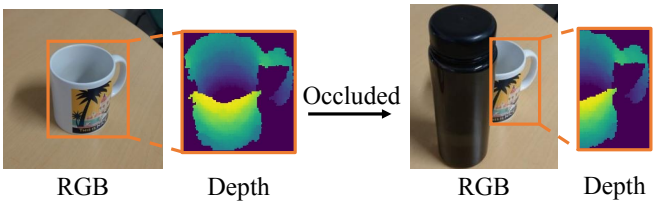
***Abstract*—In this paper, we propose an occlusion-robust pose estimation method of an unknown object instance in an object category from a depth image. In a cluttered environment, objects are often occluded mutually. For estimating the pose of an object in such a situation, a method that de-occludes the unobservable area of the object would be effective. However, there are two difficulties; occlusion causes the offset between the center of the actual object and its observable area, and different instances in a category may have different shapes. To cope with these difficulties, we propose a two-stage Encoder-Decoder model to extract features with objects whose centers are aligned to the image center. In the model, we also propose the Median-shape Reconstructor as the second stage to absorb shape variations in a category. By evaluating the method with both a large-scale virtual dataset and a real dataset, we confirmed the proposed method achieves good performance on pose estimation of an occluded object from a depth image.**

I. Introduction

In recent years, robots for daily life support are actively developed. A life support robot should be equipped with a fundamental function of holding and carrying an object which is directed by a person, such as a mug. For realizing the function, object grasping is one of the important sub-tasks in the robotics field. For object grasping, it is necessary to not only detect the target object but also estimate its pose accurately. However, there is a situation where objects are densely located, such as on a cluttered table. Since objects occlude each others in such situations, pose estimation of each of them becomes difficult. Considering this problem, in this paper, we tackle the pose estimation problem of an occluded object.

Robots are generally equipped with an RGB image sensor or a depth image sensor to observe the surroundings. In particular, depth image sensors are robust to variations in color and lighting conditions and can be used to easily detect object regions. Thus, in this paper, we deal with depth images captured by a depth image sensor.

For estimating the pose of an object from an image, it is common to extract an object region from the entire image (Fig. 1(a)). In pose estimation methods using an image [1], [5], features are extracted from images of different instances in a category. However, these methods do not consider that the observed object is severely occluded. In case the object in an input image is occluded (Fig. 1(b)), the extracted feature will be affected by the occlusion pattern.



(a) Unoccluded object (b) Occluded object

Fig. 1. Change of the observable area caused by occlusion.

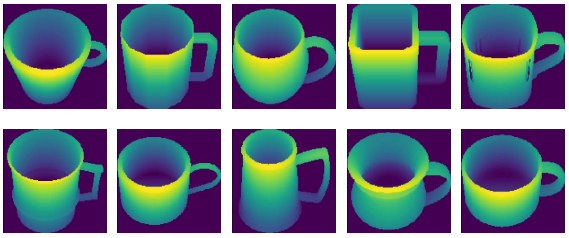


Fig. 2. Shape variations within a category.

On the other hand, in an occlusion-robust pose estimation method proposed by Sundermeyer et al. [2], features are extracted by an Augmented Autoencoder. The Augmented Autoencoder is trained to output the complete object in the image center from a perturbated image containing the object such as occlusions, different kinds of background clutter, and dynamic changes of the environment. However, this method is trained only on synthetic views of a known 3D model and is not compatible with unknown objects in a category. We call this kind of pose estimation task as *instance-level*, which is defined as follows:

- An object having specific color and shape is only considered.
- Its exact CAD model and its size are available beforehand.

In contrast, in this paper, we consider multiple objects in a category. We call this kind of pose estimation task as *category-level*, which is defined as follows:

- Objects having various colors and shapes in a category are considered. (Fig. 2)
- Unknown objects without known CAD models are targeted.

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

The goal is to realize occlusion-robust pose estimation in the category-level.

When we consider extending existing methods to category-level pose estimation, there are two difficulties. One is that occlusion causes an offset between the center of the actual object and its observable area. The other is that different instances in a category may have different shapes (Fig. 2). To tackle the first difficulty, we propose the two-stage Encoder-Decoder model that separates de-occlusion and feature extraction. De-occluding the complete region of the target object at the first stage makes it easy to align the object center to the image center accurately. Meanwhile, to tackle the second difficulty, as the second stage, we propose the Median-shape Reconstructor that is trained to decode the depth image containing a representative shaped object in the category in the same pose. Here, the median-shaped object in the category is selected as the representative shaped object. The object can be easily reconstructed from objects of various shapes. The Median-shape Reconstructor can absorb shape variations in the category.

The main contributions of this work are summarized as follows:

- We propose the two-stage Encoder-Decoder model to extract features of a de-occluded object whose center is aligned to the image center.
- We propose the Median-shape Reconstructor as the second stage to absorb shape variations in a category.
- We demonstrate the performance of the proposed method by evaluating it on a large-scale virtual dataset and a real dataset.

In this work, assuming objects are placed on a table, the input is a depth image of a target object whose upper, lower, left, or right side is occluded by another object, and the ratio of the occluded region is at most half of the target object.

The rest of this paper is organized as follows: In Section II, a brief survey is provided. In Section III, details of the proposed method are introduced. Experimental results are reported in Section IV, and a detailed discussion is provided in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

A. Object Pose Estimation from Sensory Data

Object pose estimation methods for a robot are classified into RGB image-based approaches [1], [2], [3], [4], depth image-based approaches [1], [5], [6], point cloud-based approaches [7], and voxel-based approaches [8] according to data representation. Among them, RGB and depth image representations are practical, since such sensors are mounted on life support robots commonly.

In the RGB image-based approaches, an object pose is estimated from the brightness values of the image. On the other hand, in the depth image-based approaches, an object pose is estimated from the 3D shape represented by depth values to the target object. The depth image-based approaches have the following advantages against the RGB image-based approaches:

- Robust to variations in object texture, background disturbance, and changes in lighting conditions.
- Easy to extract the target object region.

Therefore, in this paper, we take the depth image-based approach.

B. Image-based Object Pose Estimation

For image-based pose estimation, estimating the pose from features extracted from an input image is a common approach [9]. The template matching method is one of the earliest image-based methods [10]. This method utilizes many templates of the target object captured from various viewpoints beforehand, and the pose estimation result is selected from the best-matched template.

To reduce the number of templates, Murase and Nayer proposed the Parametric Eigenspace method [4]. This method represents the continuous pose change of an object by templates on a manifold in a low-dimensional subspace obtained by Principal Component Analysis (PCA). By interpolating the template of the target object on the manifold, the method realizes accurate object pose estimation even with few templates. However, since the method is based on PCA, which is an unsupervised learning method, it does not fully utilize the pose information for estimating the object's poses.

Recently, Ninomiya et al. [5] proposed a category-level pose estimation method from a depth image. They focus on Deep Convolutional Neural Networks (DCNNs) [11], which is one of the deep-learning models, as a supervised learning method for manifold embedding. They modify DCNNs for object pose estimation, named Pose-CyclicR-Net, which can accurately handle object rotation in the same category by describing the rotation angle using trigonometric functions. However, this method does not deal with the pose estimation of an occluded object.

In the case of object pose estimation by a DCNN-based model, feature vectors must be extracted from only the pixel values of the observable part of the object. Thus, when the object is occluded, the feature vector is affected by the occlusion. If a feature vector of the original unoccluded object could be extracted, a pose of the occluded object can be estimated by the same method for an unoccluded object. To extract such feature vectors, Sundermeyer et al. [2] proposed the Augmented Autoencoder, which is a generalized version of the Denoising Autoencoder [12]. They realize feature extraction from the entire object region de-occluded from an observed image with some defects in the object, background disturbance, or different lighting conditions. Although this method is effective when a small part of a specific object is occluded, it is not suitable for category-level pose estimation, and in case a large part of the object is occluded.

III. Two-stage Encoder-Decoder Model

The goal of this paper is the category-level occlusion-robust object pose estimation in cluttered environments. To achieve this goal, we propose a two-stage Encoder-Decoder model to extract features of a de-occluded object whose center is aligned

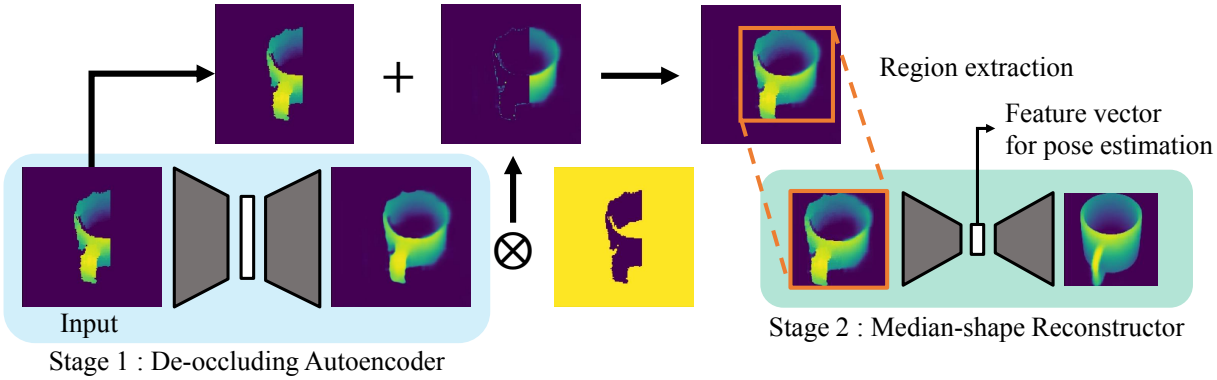


Fig. 3. Proposed two-stage Encoder-Decoder model. The input is a depth image containing the occluded target object. The De-occluding Autoencoder de-occludes the occluded part of the object. Then, the complete region of the object is extracted from the de-occluded image. Next, the Median-shape Reconstructor extracts feature vectors to decode the depth image containing the same pose of the median-shaped object in the category from the extracted region.

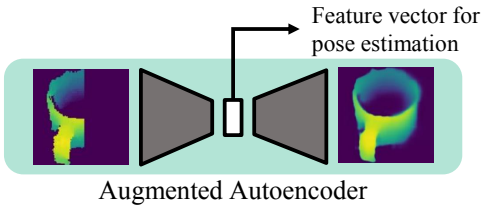


Fig. 4. Model based on Augmented Autoencoder [2].

to the image center. Fig. 3 illustrates the proposed two-stage Encoder-Decoder model. The previous model based on Augmented Autoencoder [2] (Fig. 4) does not consider the offset between the center of the actual object and its observable area. This leads to a problem that extracted features are different due to the shift of object center inside the observable region. On the other hand, the first stage Encoder-Decoder model named De-occluding Autoencoder in the proposed method de-occludes the occluded part of the object considering the offset. This allows us to explicitly specify which side of the object is being occluded. Besides, the second stage Encoder-Decoder model extracts feature vectors from the complete region of the object.

To absorb shape variations within a category, we propose the Median-shape Reconstructor as the second stage. The Median-shape Reconstructor is trained to reconstruct the same pose of the median-shaped object, which is the representative shaped object in the category.

A. Pose Representation Learning

The two models in the two-stage Encoder-Decoder model are trained sequentially. We first train the De-occluding Autoencoder to de-occlude the occluded part of the object, considering the offset between the object center and the image center. We then train the Median-shape Reconstructor to decode the depth image containing the same pose of the median-shaped object in the category. As input, a cropped image from the output of the De-occluding Autoencoder is taken.

1) Pseudo Occluding Process: In this work, we assume that the input is a depth image of a target object in which either the upper, lower, left, or right part of the object is occluded by another object. Here, the object region is extracted from the input image, and pixel values of the background are set to 0. For training the Encoder-Decoder model with such images, we need pairs of variously occluded images and corresponding target images as input. Therefore, we propose a data augmentation method by pseudoly occluding the object. To generate images whose target objects are variously occluded, we virtually occlude objects in $H \times W$ [pixels] depth images. The occluded part is randomly selected. The occlusion ratio to the object is r $(0 < r < 0.5)$. After the object is occluded, the image size will be $H \times W(1-r)$ [pixels] if it is occluded horizontally or $H(1-r) \times W$ [pixels] if it is occluded vertically. Then the image is expanded to $H \times W$ [pixels] by padding the background pixels while keeping the center of the unoccluded part of the object at the image center. Finally, the background in the image is padded to $\frac{3}{2}H \times \frac{3}{2}W$ [pixels] to be able to de-occlude most part of the object in the depth image. We call this procedure the *pseudo occluding process*.

2) De-occlusion Learning: The previous work [2] does not consider the offset between the center of the actual object and its observable area. However, it is difficult to de-occlude the occluded part of the object and correct the position shift simultaneously. To tackle the difficulty, in this work, the De-occluding Autoencoder is trained to keep the position of the unoccluded region of the object. It can de-occlude the object accurately regardless of the offset.

Fig. 5 illustrates the training of the De-occluding Autoencoder. First, the input image I_{in1} is prepared according to the pseudo occluding process (Fig. 5 (1)). The target image I_{tar1} is also prepared by expanding the background of the original image I_{org} considering the offset between the object center and the image center (Fig. 5 (2)). The De-occluding Autoencoder outputs the de-occluded image I_{out1} of the input image I_{in1} (Fig. 5 (3)). For minimizing the difference between I_{out1} and I_{tar1} (Fig. 5 (4)), the parameters are optimized.

3) De-occlusion and Alignment: After the training, the De-occluding Autoencoder will output a de-occluded image

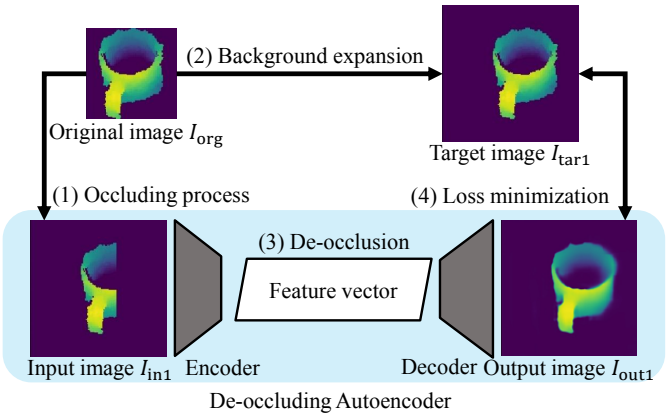


Fig. 5. Training the De-occluding AutoEncoder.

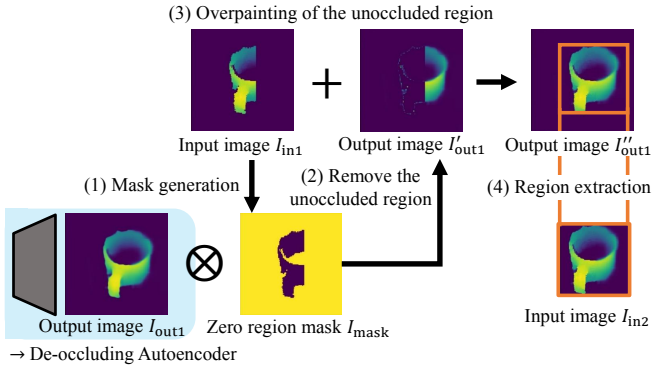


Fig. 6. De-occlusion and alignment.

similar to the target. The goal of this procedure is to generate the input image for the second stage. The main point of this procedure is as follows:

- The occluded region of the object in the input image is overpainted by the same region of the de-occluded image.
- The de-occlusion and alignment procedure aligns the object center to the image center.

Fig. 6 illustrates the de-occlusion and alignment procedure. We introduce the zero region masking to keep the observed region for feature extraction. First, a zero region mask I_{mask} is generated from I_{in1} (Fig. 6 (1)) as:

$$I_{\mathrm{mask}}(i,j) = \begin{cases} 1 & \text{if } I_{\mathrm{in1}}(i,j) = 0 \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where i and j are pixel indices of the image, and 0 is stored to zero region in I_{in1}. Second, the output image I'_{out1} is generated from I_{out1} and I_{mask} (Fig. 6 (2)) as:

$$I'_{\mathrm{out1}}(i,j) = I_{\mathrm{out1}}(i,j) \times I_{\mathrm{mask}}(i,j). \tag{2}$$

Then, we generate the output image I''_{out1} with I_{in1} and I'_{out1} (Fig. 6 (3)) as:

$$I''_{\mathrm{out1}}(i,j) = I_{\mathrm{in1}}(i,j) + I'_{\mathrm{out1}}(i,j). \tag{3}$$

The minimum rectangular region of the object in I''_{out1} is cropped, and the cropped image is squared, aligned with the

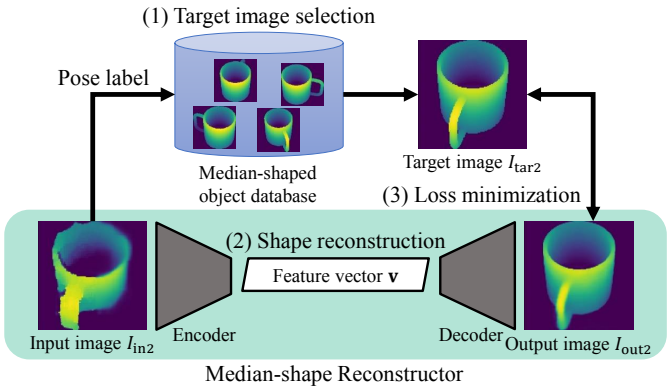


Fig. 7. Training the Median-shape Reconstructor.

longer one of the width or the height. The input image I_{in2} is generated by padding its background by $H \times W$ [pixels] (Fig. 6 (4)).

4) Median-shape Representation Learning: The Median-shape Reconstructor is an Encoder-Decoder model that decodes the image containing the same pose of the median-shaped object in a category from the input image I_{in2}. The encoder part of the Encoder-Decoder model is trained to extract feature vectors which absorb shape variations in a category.

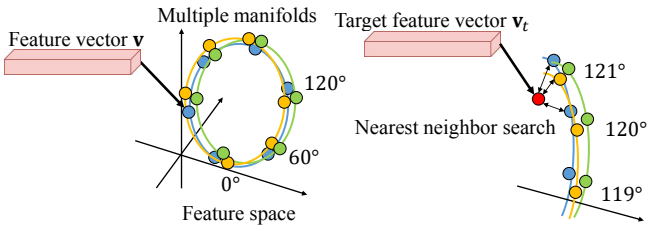
Fig. 7 illustrates the training of the Median-shape Reconstructor. First, the input image I_{in2} is generated based on Section III-A3. The target image I_{tar2} containing the same pose of the median-shaped object as the object in I_{in2} is selected from the median-shaped object database (Fig. 7 (1)). Then, the Median-shape Reconstructor encodes I_{in2} into feature vector $\mathbf{v}$ and decodes $\mathbf{v}$ into a reconstructed image I_{out2} (Fig. 7 (2)). For minimizing the difference between I_{out2} and I_{tar2} (Fig. 7 (3)), the parameters are optimized.

For selecting the median-shaped object, a DCNN model with various images containing each occluded object based on Pose-CyclicR-Net [5] is trained. The input is the de-occluded images in a category, and the output is the pose of the object represented by trigonometric. Then, intermediate activations are extracted as feature vectors, and the distances of the feature vectors are summed over each object instance. Here, we consider an object instance corresponding to the minimum sum as the median-shaped object, and construct a set of various poses of the median-shaped object as a median-shaped object database. As the median-shaped object can be easily reconstructed from objects with various shapes, we use it as the target.

After training the Median-shape Reconstructor, $\mathbf{v}$ will be used for the pose estimation as described in Section III-B.

B. Pose Estimation

The pose estimator in the proposed method is based on the Nearest Neighbor classifier the same as the related work [5]. For the training data accumulation, feature vectors $\mathcal{V} = \{\mathbf{v}\}$ are extracted from the training depth images of an object. The vectors in $\mathcal{V}$ continuously vary in the feature space



(a) Training data accumulation (b) Pose estimation

Fig. 8. Pose estimation based on the pose manifold.

(Fig. 8(a)) according to the pose of the object. By applying this procedure on multiple object instances, a set of feature vectors $\mu = \{\mathcal{V}_1, \mathcal{V}_2, \cdots\}$ is obtained. In Fig. 8(a), the different colors of the feature vectors indicate the vectors extracted from the different object instances. As the Median-shape Reconstructor absorbs the shape variation in a category, feature vectors corresponding to the same pose of different objects are expected to be the same.

In the pose estimation phase, a feature vector $\mathbf{v}_t$ is extracted from a target depth image. The pose is estimated by the Nearest Neighbor algorithm with the feature vectors $\mathcal{V}_1, \mathcal{V}_2, \cdots$ with pose labels. The pose estimator search for the nearest neighbor vector $\widehat{\mathbf{v}}$ from the set of vectors μ as:

$$\widehat{\mathbf{v}} = \underset{\mathbf{v} \in \mathcal{V}_x, \mathcal{V}_x \in \mu}{\arg\min} \|\mathbf{v} - \mathbf{v}_t\| \tag{4}$$

Finally, the pose estimator outputs the pose corresponding to the nearest neighbor vector $\widehat{\mathbf{v}}$.

IV. Evaluation

In order to evaluate the proposed method, we prepared datasets and performed experiments.

A. Dataset

We prepared a large-scale virtual dataset and a real dataset for the evaluation.

1) Large-scale Virtual Dataset: Depth images were rendered from 3D models of 105 mugs included in ShapeNet [13] as the virtual data set. For the 3D model of each mug, virtual depth images were generated by fixing the elevation angle of the camera and rotating the camera around the vertical axis with an interval of 1°. They were then scaled to 128×128 pixels. Assuming the camera is mounted on a robot, the elevation angle was set to the following three conditions: 10°, 30°, and 50°. Images from which we could not determine the pose of the object were excluded.

2) Real Dataset: Five different kinds of mugs were placed on a turntable one by one. Both color images and depth images were captured with an RGB-D image sensor Xtion PRO LIVE while rotating the turntable around the vertical axis with an interval of 1°. The distance between the depth image sensor and the target object was set to 65 cm, and the elevation angle was set to the following three conditions: 10°, 30°, and 50°.

As preprocessing, a mug was first detected from each color image using YOLOv3 [14] and a bounding box was obtained. To remove the background, the background plane was estimated by the least-squares method using the depth values sampled from the region of the turntable. The region of the object in the depth image was cropped and scaled to 128×128 pixels.

B. Experimental Setting

The Encoder and the Decoder were constructed with five Convolution layers in each Encoder-Decoder model. The mean square error was used as the loss function and Adam as the optimizer in training the models. The proposed method was evaluated in two settings: Evaluation on the large-scale virtual dataset, and evaluation on the real dataset.

1) Setting 1: Our model was trained and the methods were evaluated on the large-scale virtual dataset. For the De-occluding Autoencoder and Median-shape Reconstructor training, the depth images from the 3D models of 100 mugs were used. A total of 25,100 / 27,100 / 36,000 images for elevation angles of 10° / 30° / 50°, respectively, were used. The models were trained with the pseudo occluding process with random occlusion ratios and parts.

For the pose estimator training, depth images from the 3D models of the 30 mugs among 100 mugs were used for the models. Five images were augmented by running the pseudo occluding process five times for each image with random occlusion ratios and parts. A total of 37,650 / 40,650 / 54,000 images for elevation angles of 10° / 30° / 50°, respectively, were used.

For evaluation, depth images from the 3D models of the five mugs with an interval of 10° out of 100 mugs were used for training the models. The pseudo occluding process was run once for each image. A total of 117 / 124 / 152 images for elevation angles of 10° / 30° / 50°, respectively, were used.

2) Setting 2: Our model was trained on all of the large-scale virtual dataset and four mugs in the real dataset. The methods were evaluated on the rest of the mugs in the real dataset and repeated the validation while changing the combination of mugs in the real dataset (five-fold cross-validation). A total of 1,540 / 1,760 / 1,727 real images for elevation angles of 10° / 30° / 50°, respectively, were used.

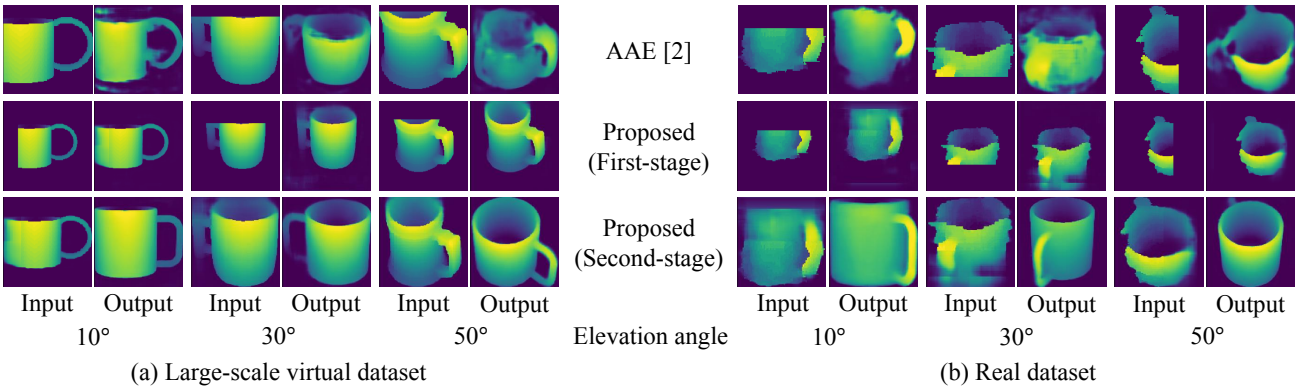
For the De-occluding Autoencoder and Median-shape Reconstructor training, depth images from the 3D models of the 100 mugs and four real mugs with an interval of 1° were used. The models were trained while running the pseudo occluding process with random occlusion ratios and parts.

For the pose estimator training, depth images from the 3D models of 30 mugs among 100 mugs for the models training and four real mugs with an interval of 1° were used. Five images were augmented by running the pseudo occluding process five times for each image with random occlusion ratios and parts.

For evaluation, depth images were used from the mug out of the four real mugs for training the models with an interval of 10°. The pseudo occluding process was run once for each image.

TABLE I
POSE ESTIMATION RESULTS IN SETTING 1

	MAAE [°] ↓			95% MAAE [°] ↓			w/in 5°[%] ↑			w/in 10°[%] ↑		
Elevation angle [°]	10	30	50	10	30	50	10	30	50	10	30	50
PCR-Net [5]	3.56	4.14	13.91	3.07	3.73	7.06	74.4	71.8	50.7	96.6	**97.6**	76.3
AAE [2]	15.55	11.44	9.29	8.63	7.10	5.72	53.8	50.0	52.6	74.4	72.6	79.6
Proposed	**3.02**	**3.50**	**2.04**	**2.67**	**2.91**	**1.78**	**88.6**	**83.1**	**94.7**	**98.3**	96.0	**100.0**



(a) Large-scale virtual dataset (b) Real dataset

Fig. 9. Visualization results.

C. Metrics

To evaluate the estimated rotation angle around the vertical axis quantitatively, we calculated the absolute angular errors between the pose estimation results and the true poses. The Mean Absolute Angle Error (MAAE) was used as the metric. As the MAAE is often affected by outliers (results with the estimated poses far apart from the true pose), we also used the 95% MAAE as a metric. The metric is calculated while excluding the upper 5% outliers. By considering an application that a robot grasps an object, we also evaluated the ratio of absolute angle error within $5°$ (*w/in* $5°$) and $10°$ (*w/in* $10°$). Besides, we visualized the ratio of absolute angle error within $n°$ $(0 \leq n \leq 20, n \in \mathbb{Z})$ in graphs.

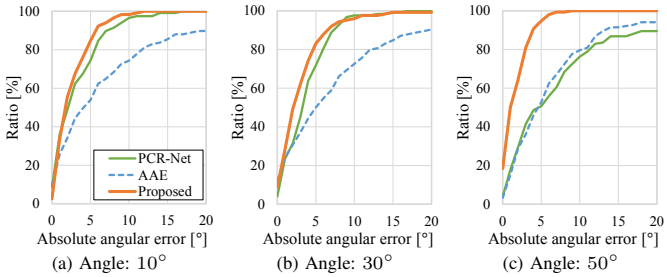
D. Comparative Methods

We evaluated two comparative methods A1 and A2, together with the proposed method.

- Comparative method A1 is based on Pose-CyclicR-Net [5], where features are extracted by a regression model based on a DCNN trained with our dataset. This method is abbreviated as PCR-Net.
- Comparative method A2 is based on Augmented Autoencoder [2], where features are extracted by the model based on Augmented Autoencoder (Fig. 4) trained with our dataset. This method is abbreviated as AAE.

E. Results

1) Setting 1: Fig. 9 shows visualization results using the Encoder-Decoder model. By comparative method A2 (AAE), the target object cannot be de-occluded accurately because such occlusions in the input image are not originally assumed.



(a) Angle: 10° (b) Angle: 30° (c) Angle: 50°

Fig. 10. Absolute angular error in setting 1.

On the other hand, in the first stage of the proposed method, the object can be de-occluded accurately by expanding the background and keeping the position of the unoccluded region of the object. As the second stage of the proposed method, the same pose of the median-shaped object can be reconstructed.

Table I shows the pose estimation results in setting 1. The proposed method achieves the best performance on most items. Fig. 10 shows the ratio of absolute angle error within $n°$ $(0 \leq n \leq 20, n \in \mathbb{Z})$ in setting 1. The proposed method is more effective than the comparative method for elevation angle $50°$.

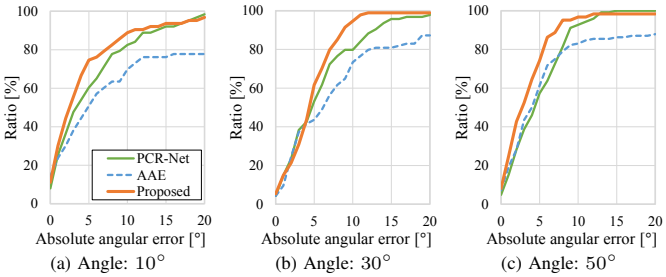
2) Setting 2: In Fig. 9, the de-occlusion of images in the real dataset is more difficult than that of images in a virtual dataset caused by the noisy observation values unique to the real dataset. Actually, by comparative method A2 (AAE), the occluded region is not well de-occluded in the case of the real dataset than that of the virtual dataset. On the other hand, by the proposed method, the Median-shape Reconstructor outputs the median-shaped object in the virtual dataset from the target object in the real dataset. The Encoder-Decoder model can

TABLE II
POSE ESTIMATION RESULTS IN SETTING 2

	MAAE [°] ↓			95% MAAE [°] ↓			w/in 5°[%] ↑			w/in 10°[%] ↑		
Elevation angle [°]	10	30	50	10	30	50	10	30	50	10	30	50
PCR-Net [5]	**5.81**	6.21	5.19	5.08	5.46	4.87	60.3	53.2	57.3	82.5	79.8	92.7
AAE [2]	19.42	17.85	14.18	14.27	10.61	8.03	50.8	43.6	61.3	69.8	73.4	83.1
Proposed	7.61	**5.15**	**4.27**	**3.98**	**4.67**	**3.19**	**74.6**	**61.7**	**74.2**	**88.9**	**94.7**	**96.8**

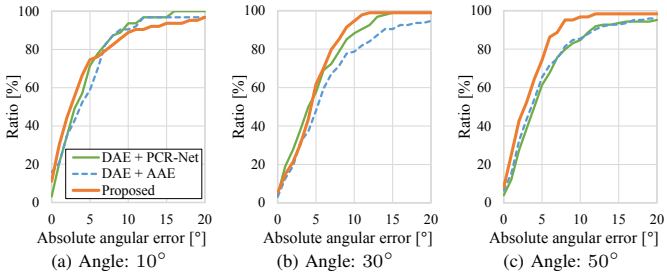
TABLE III
POSE ESTIMATION RESULTS FOR THE SECOND STAGE

	MAAE [°] ↓			95% MAAE [°] ↓			w/in 5°[%] ↑			w/in 10°[%] ↑		
Elevation angle [°]	10	30	50	10	30	50	10	30	50	10	30	50
DAE + PCR-Net	**4.60**	5.39	8.85	4.08	4.83	5.15	71.4	57.4	61.3	**93.7**	88.3	84.7
DAE + AAE	5.47	7.26	7.17	4.32	6.13	4.84	58.7	47.9	65.3	90.5	85.5	83.1
Proposed	7.61	**5.15**	**4.27**	**3.98**	**4.67**	**3.41**	**74.6**	**61.7**	**74.2**	88.9	**94.7**	**96.8**



(a) Angle: 10° (b) Angle: 30° (c) Angle: 50°

Fig. 11. Absolute angular error in setting 2.



(a) Angle: 10° (b) Angle: 30° (c) Angle: 50°

Fig. 12. Absolute angular error for the second stage in setting 2.

reconstruct the object accurately since the variation of the output images is smaller.

Table II shows the pose estimation results in setting 2. The proposed method achieves the best performance on most items also with the real dataset. Fig. 11 shows the ratio of absolute angle error within $n°$ $(0 \leq n \leq 20, n \in \mathbb{Z})$ in setting 2. The proposed method is also effective for the real dataset.

Through this evaluation, we demonstrated the effectiveness of the proposed two-stage Encoder-Decoder model.

V. DISCUSSION

A. *Effectiveness of the Median-shape Reconstructor*

To demonstrate the effectiveness of the Median-shape Reconstructor, we evaluated the second stage of the two-stage Encoder-Decoder model in setting 2. We evaluated two comparative methods B1 and B2, together with the proposed method. For each of the methods, the first stage of the two-stage model was fixed as the Deoccluding Autoencoder. The detail of the comparative methods is as follows:

- Comparative method B1 is based on Pose-CyclicR-Net [5], where the features are extracted by a regression model based on a DCNN trained with the dataset for the Median-shape Reconstructor. This method is abbreviated as DAE + PCRNet.
- Comparative method B2 is based on Augmented Autoencoder [5], where the features are extracted by the Augmented Autoencoder trained with the dataset for the Median-shape Reconstructor. The second stage model outputs the same pose of the target object instead of the same pose of the median-shape object. This method is abbreviated as DAE + AAE.

Table III shows the pose estimation results in setting 2. Fig. 12 shows the ratio of absolute angle error within $n°$ $(0 \leq n \leq 20, n \in \mathbb{Z})$. The Median-shape Reconstructor achieves the best performance except for the case of elevation angle $10°$.

Firstly, we compare the proposed method with comparative method B1 (DAE + PCR-Net). The rotation angle represented by trigonometric functions is given when the PCR-Net is trained. On the other hand, the rotation angle and the median-shape is given as images when the Median-shape Reconstructor is trained. The Median-shape Reconstructor is more effective than the PCR-Net since the encoder part of the model can extract the features which are as close as possible to features extracted from an object instance, a median-shaped object.

Secondly, we compare the proposed method with comparative method B2 (DAE + AAE). The only difference between them is that the AAE outputs the same pose of the target object, but the Median-shape Reconstructor outputs the same pose of the median-shaped object.

TABLE IV
MAAE [°] IN MULTIPLE CATEGORIES

Category	Mug	Car	Bike	Chair
PCR-Net [5]	13.91	2.09	14.36	4.52
AAE [2]	9.29	13.48	61.93	12.99
Proposed	**2.04**	**0.60**	**10.37**	**2.44**

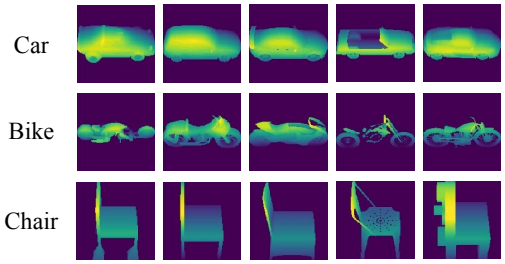


Fig. 13. Dataset with multiple categories.

Through this evaluation, we demonstrated the effectiveness of the proposed approach in which the proposed Encoder-Decoder model outputs an object instance.

B. Evaluation in Multiple Categories

We also evaluated the proposed method in multiple categories other than Mug; Car, Bike, and Chair in setting 1. The elevation angle was set to $50°$. Fig. 13 shows the examples of the dataset.

Table IV shows the Mean Absolute Angular Error (MAAE) in multiple categories. The proposed method achieves the best performance for all categories in the dataset. We confirmed that the proposed method is also effective in category-level.

VI. Conclusion

In this paper, we proposed an occlusion-robust pose estimation method of an unknown object instance in an object category from a depth image. To tackle the difficulty that occlusion causes an offset between the center of the actual object and its observable area, we proposed a two-stage Encoder-Decoder model that separates de-occlusion and feature extraction. To tackle the difficulty that different instances in a category may have different shapes, as the second stage, we proposed a Median-shape Reconstructor that is trained to decode the depth image containing the same pose of a representative shaped object in the category. By evaluating the method with both a large-scale virtual dataset and a real dataset, we confirmed the proposed method achieves good performance on pose estimation of an occluded object from a depth image. The part of the dataset used in the evaluation will be released.

As future work, we need to extend the method so that it can handle 3D axes rotations. For 3D pose estimation, our dataset must include an enormous number of depth images. We also need to improve the method to estimate the pose accurately with a smaller number of training images.

Acknowledgement

Parts of this research were supported by Grant-in-Aid for Scientific Research (JP17H00745).

References

[1] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proceedings of the 13th International Conference on Computer Vision*, Nov. 2011, pp. 858–865.

[2] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3D orientation learning for 6D object detection from RGB images," in *Proceedings of the 15th European Conference on Computer Vision*, Part 6, Sept. 2018, pp. 699–715.

[3] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 3364–3372.

[4] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance," *International Journal of Computer Vision*, vol. 14, no. 1, pp. 5–24, Jan. 1995.

[5] H. Ninomiya, Y. Kawanishi, D. Deguchi, I. Ide, H. Murase, N. Kobori, and Y. Nakano, "Deep manifold embedding for 3D object pose estimation," in *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 5, March 2017, pp. 173–178.

[6] M. Li and K. Hashimoto, "Accurate object pose estimation using depth only," *Sensors*, vol. 18, no. 4, pp. 1045:1–17, April 2018.

[7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 77–85.

[8] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 1912–1920.

[9] A. V. Patil and P. Rabha, "A survey on joint object detection and pose estimation using monocular vision," Computing Research Repository, arXiv preprint arXiv:1811.10216, Nov. 2018.

[10] R. T. Chin and C. Dyer, "Model-based recognition in robot vision," *ACM Computing Surveys*, vol. 18, pp. 67–108, March 1986.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol. 1, Dec. 2012, pp. 1097–1105.

[12] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," in *Journal of Machine Learning Research*, vol. 11, Dec. 2010, pp. 3371–3408.

[13] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," Computing Research Repository, arXiv preprint, arXiv:1512.03012, Dec. 2015.

[14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Computing Research Repository, arXiv preprint, arXiv:1804.02767, April 2018.