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

Ω-GAN: Object Manifold Embedding GAN for Image Generation by Disentangling Parameters into Pose and Shape Manifolds

Yasutomo Kawanishi, Daisuke Deguchi, Ichiro Ide, and Hiroshi Murase
Nagoya University
Aichi, Japan
Email: kawanishi@i.nagoya-u.ac.jp

***Abstract*—In this paper, we propose Object Manifold Embedding GAN (Ω-GAN) to generate images of variously shaped and arbitrarily posed objects from a noise variable sampled from a distribution defined over the pose and the shape manifolds in a vector space. We introduce Parametric Manifold Sampling to sample noise variables from a distribution over the pose manifold to conditionally generate object images in arbitrary poses by tuning the pose parameter. We also introduce Object Identity Loss for clearly disentangling the pose and shape parameters, which allows us to maintain the shape of the object instance when only the pose parameter is changed. Through evaluation, we confirmed that the proposed Ω-GAN could generate variously shaped object images in arbitrary poses by changing the pose and shape parameters independently. We also introduce an application of the proposed method for object pose estimation, through which we confirmed that the object poses in the generated images are accurate.**

I. Introduction

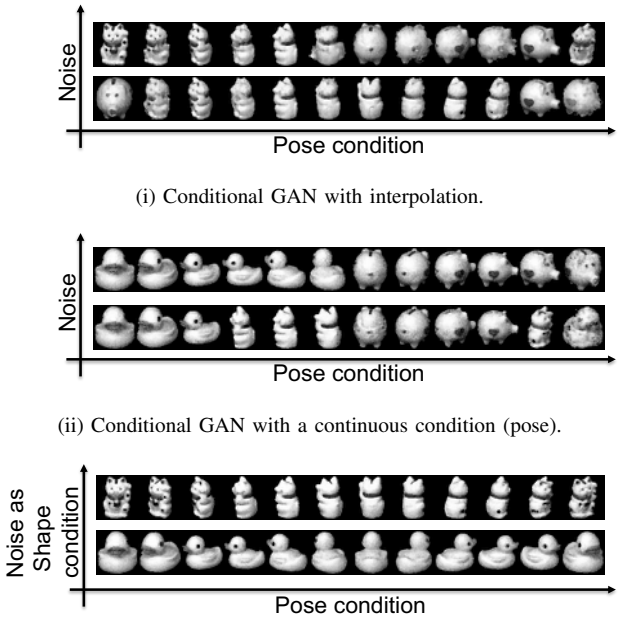
In the computer vision field, Generative Adversarial Nets (GAN) [1], which can generate unseen realistic data from noise variables sampled from a low-dimensional distribution by using Deep Neural Networks, is currently receiving considerable attention. In this paper, we focus on the image generation of variously shaped rigid objects in arbitrary poses.

Generally, a GAN generates images from noise variables sampled from a distribution such as the Gaussian. As all the data variations are embedded into a single distribution, it is difficult to generate data by controlling individual parameters.

By introducing condition variables, Conditional GAN [2] enables to generate targeted images by controlling the condition variables. The "one-hot vector" representation is usually used for the condition variables, and continuous values are also used for the condition variables to control the image variations in a regression manner. However, as the object pose variation is cyclic, it is difficult to handle the variations as is.

Another issue is about the guarantee to maintain the object instance's shape when the pose parameter is changed while the other parameters are fixed. There is no guarantee of it in the previous methods because these parameters are concatenated in training.

Therefore, if we generated images of rotating objects by changing the pose condition while fixing the shape condition, the shape of the object may also change (Fig. 1 (i) and (ii)), in case these parameters are entangled.



(i) Conditional GAN with interpolation.

(ii) Conditional GAN with a continuous condition (pose).

(iii) Proposed Ω-GAN.

Fig. 1. Image generation while fixing other parameters besides the pose parameter. Each pair of two rows is generated from different shape parameters.

In this research, we focus on the following two assumptions: 1) The pose variations of a rigid object can be parametrized by a pose parameter mapped onto a manifold in a feature space (pose manifold), which can be explicitly described by the rotation angle based on the concept of the Parametric Eigenspace method [3]. 2) The shape variations of the objects can also be parametrized by a shape parameter mapped onto a manifold in a feature space (shape manifold) independently from the pose parameter. Accordingly, we propose a novel GAN framework that disentangles the pose and shape parameters into the pose and shape manifolds. The proposed GAN samples a noise vector from the product manifold of the pose and shape manifolds and generates images corresponding to

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

the parameters (Fig. 1 (iii)). We name this GAN as *Object Manifold Embedding GAN ($\mathbf{\Omega}$-GAN)*.

By sampling a sequence of the pose parameters along with the pose manifold, we can generate images of an object in various poses while maintaining its shape, and by sampling the shape parameters from the shape manifold, we can generate variously shaped objects while maintaining their pose.

Our contributions in this study are summarized as follows:

- Object Manifold Embedding GAN ($\mathbf{\Omega}$-GAN): It can generate images of variously shaped and arbitrarily posed objects by controlling the pose and shape parameters independently.
  - Parametric Manifold Sampling: It samples the noise variables from a distribution over the pose and shape manifolds to allow controlling the cyclic pose and continuous shape variations.
  - Object Identity Loss: It makes the GAN maintain the object instance's shape when only the pose parameter is changed.
- Application to object pose estimation from a depth image for quantitative evaluation of generated images: It uses the images generated by the proposed $\mathbf{\Omega}$-GAN for data augmentation.

The rest of this paper is organized as follows: In Section II, a brief survey on GAN is provided. In Section III, we discuss the variations among the images capturing variously shaped objects in arbitrary poses. In Section IV, details of the proposed GAN framework are introduced. Experimental results are reported in Section V. Then, in Section VI, we introduce an application of the proposed GAN for object pose estimation to demonstrate its effectiveness. Finally, the paper is concluded in Section VII.

II. Related Work

Generative Adversarial Nets (GAN) [1] consists of a pair of two networks; a generator G that is trained to capture the data distribution and a discriminator D which is trained to estimate the probability of whether a sample is taken from the training data or the data generated by the generator. The generator and the discriminator are trained simultaneously, and they play the two-player minimax game as,

$$\begin{aligned}&\min_G \max_D V(D,G)\\ &= \min_G \max_D \Big(\mathrm{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})]\\ &\quad + \mathrm{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1-D(G(\mathbf{z})))]\Big)\,,\end{aligned} \tag{1}$$

where p_{data} is the distribution of the training data, and $p_{\mathbf{z}}$ is a noise distribution. After the training, the generator is able to generate a realistic fake sample $G(\mathbf{z})$ from a noise vector $\mathbf{z}$ sampled from $p_{\mathbf{z}}$.

Among various extensions of GAN [4], [5], [6], Deep Convolutional Generative Adversarial Networks (DCGAN) [7] is known to be suitable for generating high-quality images. For generating condition-specific realistic data, several approaches have been proposed. By modifying the discriminator and the generator to control by condition parameters $\mathbf{c}$, Conditional GAN (CGAN) [2] can generate data corresponding to given conditions as

$$\begin{aligned}&\min_G \max_D V(D,G)\\ &= \min_G \max_D \Big(\mathrm{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x}|\mathbf{c})]\\ &\quad + \mathrm{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1-D(G(\mathbf{z}|\mathbf{c})|\mathbf{c}))]\Big)\,.\end{aligned} \tag{2}$$

The generator generates an output from a noise variable $\mathbf{z} \sim p_z$, which is sampled from a distribution p_z, and a condition variable $\mathbf{c}$, and the discriminator judges whether the pair of the generated sample and the condition variable $(\mathbf{x}, \mathbf{c})$ is real or fake. By training them simultaneously, finally, the generator will generate $\mathbf{x}_{\text{fake}} = G(\mathbf{z}|\mathbf{c})$ that the discriminator may fail to judge. Several applications of CGAN, such as Pix2pix [8] and CycleGAN [9], have been proposed as implementations of CGAN. As a variant of CGAN, Auxiliary Classifier GAN (AC-GAN) [10] modifies the discriminator to train with an auxiliary classifier to generate condition-dependent data. InfoGAN [11] is also an extension of GAN that can learn disentangled representation in an unsupervised manner. StyleGAN [12] uses a style-based generator that focuses on separating the high-level attributes and small stochastic variations. Fader Networks [13], which is a combination of an Encoder-Decoder model and a GAN, can also control multiple attributes using sliding knobs.

Thanks to the recent advances of neural networks, 3D objects can be implicitly modeled using CNNs. Maxim *et al.* proposed an Encoder-Decoder model to rotate an object in an image with a given rotation angle [14]. HoloGAN [15] introduces a rotation model of a rigid model into a GAN framework to realize object rotation from an image. FATTEN [16] is also proposed for rotating an object in an image with a given rotation angle while preserving the object category using an Encoder-Decoder model. The object category is evaluated as a multi-class classification problem based on an Encoder-Decoder model in the method, while the proposed method handles the object identity as an instance identification problem based on the GAN-based latent variable modeling. Additionally, due to shape variations existing within a category, the method [16] cannot maintain the object shape when the pose parameter was changed.

When we assume an inappropriate distribution, the mode collapse phenomenon usually occurs while training a GAN. Therefore, how to choose the noise distribution and how to sample the noise variables from the distribution are also the issues among GAN research. Unrolled GAN [17] employs unrolling in generator training to avoid the issues above. BourGAN [18] samples the noise variables from a Gaussian mixture model generated by Bourgain Embedding. To generate images with much diversity, Mode Seeking GAN [19] explicitly maximizes the ratio of the distance between generated images for the corresponding latent codes. Recently, there are several existing GANs [20], [21] which disentangle the object pose and their appearances.

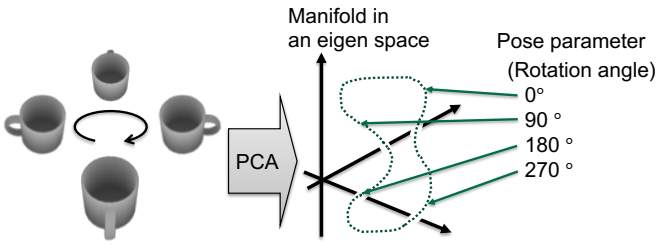


Fig. 2. By the Parametric Eigenspace method [3], images of a rotating object are mapped onto a manifold in a low-dimensional eigenspace associated with rotation parameters.

In this research, we aim to generate variously shaped and arbitrarily posed object images by explicitly controlling shape and pose parameters. However, even by using these existing GANs, it is difficult to generate such images. The pose parameter has circularity that could not be represented by Gaussian distributions, and there is no guarantee that the shape of the object in the images is maintained when only the pose parameter is changed.

III. Appearance Variations of Rigid Objects

Among the images that observe variously shaped objects in arbitrary poses, there are appearance variations caused by the following two factors; pose variation and shape variation. Here, we introduce the parametric representation of each of them, followed by the representation when they are combined.

A. Pose Variation of an Object

Based on the key concept of the Parametric Eigenspace [3], when a camera observes a rigid object from a fixed distance, the appearance variations of the object only depend on the rotation angles of the object in relation to the camera (Fig. 2). In a low-dimensional eigenspace, images of an object in various poses can be mapped onto a manifold. Here, assuming the rotation is restricted around a single axis, the appearance variation depends only on the rotation angle around the axis. Therefore, the pose parameters, namely the rotation angles around the axis, can be mapped onto a one-dimensional manifold, as shown in Fig. 2. Similarly, if the rigid object's rotation has three degrees of freedom, the pose parameters, namely the three-dimensional rotation angles, can also be mapped onto a three-dimensional manifold.

Therefore, an arbitrary pose can be represented by a pose parameter on the pose manifold $\mathcal{M}_p$.

B. Shape Variation of Objects

Each object instance has its shape. However, they share a common structure if they are similar objects. Images of variously shaped objects can also be mapped onto a manifold in a low-dimensional space where similar images correspond to similar shape parameters. Therefore, by considering the vector on the shape manifold $\mathcal{M}_s$, as a shape parameter, each shape can also be represented by the shape parameter.

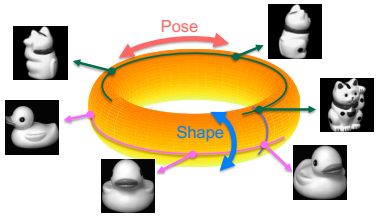


Fig. 3. Example of the product manifold.

C. Product Manifold of the Pose and Shape Manifolds

In conclusion, the appearance variations of the images of variously shaped objects in arbitrary poses can be expressed in the product manifold of the two different parameter manifolds as

$$\mathcal{M} = \mathcal{M}_p \otimes \mathcal{M}_s. \tag{3}$$

Fig. 3 shows an example of the product manifold whose pose manifold is one dimension. Several images corresponding to the pose and shape parameters are shown in the figure. Along the manifold's circumferential axis, the pose varies continuously, while the shape varies along the orthogonal direction to the circumferential axis.

IV. Object Manifold Embedding GAN

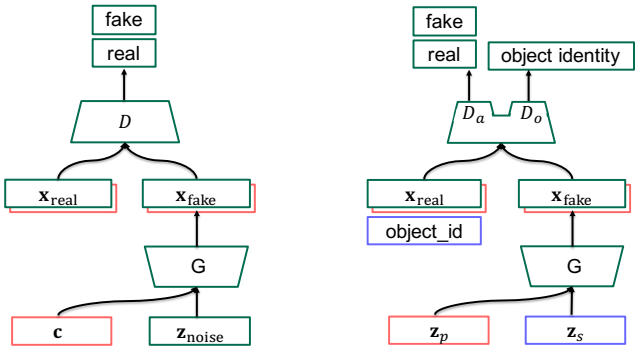
As introduced in Section II, Conditional GAN (CGAN) [2] can generate images corresponding to the given conditions, $\mathbf{c}$. Arbitrarily posed object images can be generated by considering the pose parameter as the condition. However, there are two difficulties: The pose parameter has circularity, and there is no guarantee that the object instance's shape in the images is preserved when only the pose parameter is changed.

To tackle these difficulties, in this paper, by extending CGAN [2], we propose the Object Manifold Embedding GAN ($\mathbf{\Omega}$-GAN) that samples noise variables from the product manifold and explicitly disentangles the pose and shape parameters of the objects. Using the proposed $\mathbf{\Omega}$-GAN, we aim to find the transformation between the shape and pose parameters to the generated images. We modify the original CGAN based on the following concepts:

- Parametric Manifold Sampling: To handle the pose circularity and shape continuity,
- Object Identity Loss: To maintain the object instance's shape when only the pose parameter is changed and to maintain the pose when only the shape parameter is changed.

The loss function of the proposed $\mathbf{\Omega}$-GAN is defined as follows:

$$\begin{aligned}
&\min_G \max_D V(D,G) \\
&= \min_G \max_D \Big(\mathrm{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x}|\mathbf{c})] \\
&\quad + \mathrm{E}_{\mathbf{z}\sim p_{\mathcal{M}_\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z}|\mathbf{z}_p)|\mathbf{z}_p))] \\
&\quad + \alpha \mathrm{E}_{\mathbf{x}_1,\mathbf{x}_2\sim p_{\text{data}}(\mathbf{x})}[O_r(\mathbf{x}_1,\mathbf{x}_2)] \\
&\quad + \alpha \mathrm{E}_{\mathbf{z}_1,\mathbf{z}_2\sim p_{\mathcal{M}_\mathbf{z}}(\mathbf{z})}[O_f(G(\mathbf{z}_1|\mathbf{z}_{1p}), G(\mathbf{z}_2|\mathbf{z}_{2p}))] \Big),
\end{aligned} \tag{4}$$



(i) Conditional GAN [2]. (ii) Proposed **Ω**-GAN.

Fig. 4. GAN architectures.

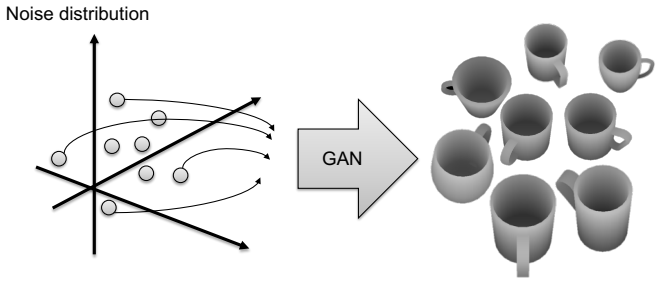


Fig. 5. Image generation by the traditional GAN [1].

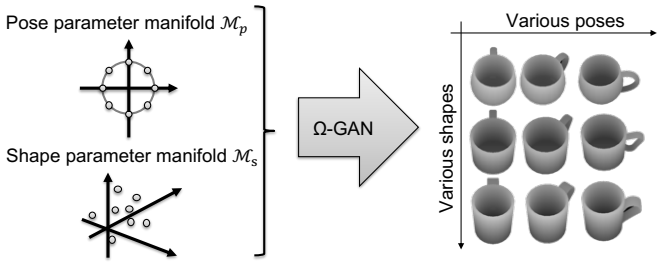


Fig. 6. Proposed **Ω**-GAN can generate variously shaped objects in various poses. Especially, the poses can be controlled by a parameter independently.

where $\mathbf{c}$ is the pose parameter assigned to the real data $\mathbf{x}$, $\mathbf{z}_p$ the pose parameter of $\mathbf{z}$, α a weight parameter, and $O_r(\mathbf{a}, \mathbf{b})$ and $O_f(\mathbf{a}, \mathbf{b})$ the Object Identity Losses for real and fake data, respectively.

The network architecture of the proposed **Ω**-GAN compared to that of CGAN is shown in Fig. 4.

In the following subsections, the detail of the proposed **Ω**-GAN is explained.

A. Parametric Manifold Sampling

As mentioned in Section III, the appearance variations can be described by the product manifold of a "pose manifold" $\mathcal{M}_p$ and the "shape manifold" $\mathcal{M}_s$. The former describes the pose variations of a specific object, while the latter describes the shape variations of objects. Different from the generator in the traditional GAN [1], as shown in Fig. 5, by sampling from a distribution over the product manifold $\mathcal{M} = \mathcal{M}_p \otimes \mathcal{M}_s$, specifically, by replacing $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ in Equation (1) with $\mathbf{z} = (\mathbf{z}_p, \mathbf{z}_s) \sim p_{\mathcal{M}}(\mathbf{z})$, we realize image generation of various shapes and poses (Fig. 6). For example, because the pose parameter, such as the rotation angle, can be mapped onto the pose manifold $\mathcal{M}_p$, we can control an object's poses in the generated images with the pose parameter.

In the case of a single-axis rotation, the pose parameter can be described as a variable on a one-dimensional manifold. By describing the pose variation on a unit circle, a one-dimensional manifold in a two-dimensional space, we can continuously describe the object rotation as discussed in [22]. A point on a unit circle can be described as,

$$\mathbf{z}_p = (z_1, z_2) \quad (z_1^2 + z_2^2 = 1, \quad z_1, z_2 \in [-1, 1]). \tag{5}$$

Once a random variable $\theta \in [0°, 360°)$ is sampled from a uniform distribution, the pose parameter $\mathbf{z}_p = (\cos\theta, \sin\theta)$ can be obtained.

On the other hand, we sample a noise variable $\mathbf{z}_s$ from a Gaussian distribution over the N dimensional space $V_s = \mathbb{R}^N$ for the shape variation.

Finally, a noise variable $\mathbf{z}$ is obtained by combining the pose and shape's noise variables as

$$\mathbf{z} = (\mathbf{z}_p, \mathbf{z}_s) \sim p_{\mathcal{M}}(z). \tag{6}$$

This is input into the generator G.

In the image generation phase, by changing the parameter θ in $[0°, 360°)$, images as if we were capturing a rotating object are expected to be generated. Similarly, by changing the parameter $\mathbf{z}_s$, images as if we were capturing various objects are expected to be generated.

B. Object Identity Loss

To explicitly disentangle the shape and pose parameters, the discriminator D should not only judge whether the input $(\mathbf{x}, \mathbf{z}_p)$ is real or fake (adversarial loss) but should also evaluate whether the images are from the same object or not (Object Identity Loss). Therefore, it has two outputs D_a and D_o, as shown in Fig. 4; D_a is for the adversarial loss calculation while D_o is for the Object Identity Loss calculation.

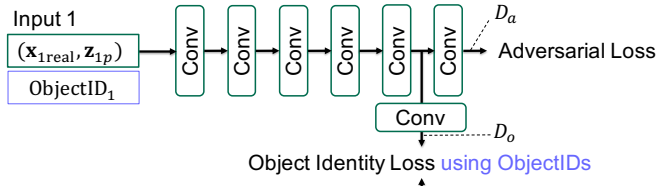
The Object Identity Loss evaluates the object's identity, namely whether the shapes of two given images are identical or not. The loss evaluation is based on the Siamese Network [23] with the contrastive loss [24]. If the shapes of the objects in two images are identical, the loss forces the distance of the network's output features to be small even if they are observed in different poses, and vice versa. Generally, contrastive loss L_c is defined as

$$L_c = y_{ab} d_{ab} + (1 - y_{ab}) \max(0, \tau - d_{ab}), \tag{7}$$

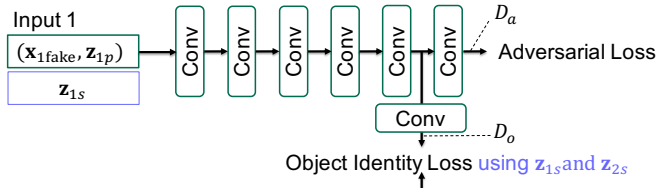
$$d_{ab} = d(\mathbf{f}_a, \mathbf{f}_b), \tag{8}$$

where $y_{ab} \in \{0, 1\}$ indicates whether the two inputs a and b are identical or not, while $\mathbf{f}_a$ and $\mathbf{f}_b$ are the network's output features.

In the discriminator training phase, we expect that the object IDs are given as additional input, and the Object Identity Loss



(i) Loss calculation for real data.



(ii) Loss calculation for fake data.

Fig. 7. Discriminator of the proposed **Ω**-GAN and the loss calculation structures.

is evaluated referring to the information (Fig. 7 (i)). The loss function O_r given the two outputs $\mathbf{a} = \mathbf{x}_{1_{\text{real}}}$ and $\mathbf{b} = \mathbf{x}_{2_{\text{real}}}$ for the discriminator training is defined as follows:

$$O_r(\mathbf{a}, \mathbf{b}) = y_r(\mathbf{a}, \mathbf{b}) d(\mathbf{a}, \mathbf{b})^2 + (1 - y_r(\mathbf{a}, \mathbf{b})) \max(0, \tau - d(\mathbf{a}, \mathbf{b}))^2, \quad (9)$$

$$d(\mathbf{a}, \mathbf{b}) = ||D_o(\mathbf{a}) - D_o(\mathbf{b})||, \quad (10)$$

$$y_r(\mathbf{a}, \mathbf{b}) = \begin{cases} 1 & \text{if } l_r(\mathbf{a}) = l_r(\mathbf{b}) \\ 0 & \text{otherwise} \end{cases}, \quad (11)$$

where τ is a margin parameter, and $l_r(\mathbf{a})$ is a function that returns the corresponding object ID of the input. The loss is not evaluated for the generated data. Through the training with this loss, the discriminator learns the shape similarity metric.

In the generator training phase, the loss is evaluated using the shape parameters $\mathbf{z}_s$ (Fig. 7 (ii)). If the two input's shape parameters are the same, then the distance of the two features is expected to be small. The loss function, O_f, given $\mathbf{a} = \mathbf{x}_{1_{\text{fake}}} G(\mathbf{z}_1|\mathbf{z}_{1p})$ and $\mathbf{b} = \mathbf{x}_{2_{\text{fake}}} = G(\mathbf{z}_2|\mathbf{z}_{2p})$ for the generator training is defined as follows:

$$O_f(\mathbf{a}, \mathbf{b}) = y_f(\mathbf{a}, \mathbf{b}) d(\mathbf{a}, \mathbf{b})^2 + (1 - y_f(\mathbf{a}, \mathbf{b})) \max(0, \tau - d(\mathbf{a}, \mathbf{b}))^2, \quad (12)$$

$$y_f(\mathbf{a}, \mathbf{b}) = \begin{cases} 1 & \text{if } l_f(\mathbf{a}) = l_f(\mathbf{b}) \\ 0 & \text{otherwise} \end{cases}, \quad (13)$$

where $l_f(\mathbf{a})$ is a function that returns the corresponding shape parameter of the input ($\mathbf{z}_p = l_f(\mathbf{z})$). This loss function aims to train the generator G to generate similarly shaped objects when the shape parameters are similar and differently shaped objects when they are different. Through the training with this loss, the generator learns to generate images by preserving the shape if the shape parameter is the same.

V. Evaluation

A. Dataset

To evaluate the image generation, we used two datasets. One is a subset ("cat", "duck", and "pig" images) of COIL-20 [25], which is a well-known image dataset for object pose estimation. The other is a "Mug" depth-image dataset generated from CAD models. As the CAD models, we selected 133 "Mug" models from the ShapeNet dataset [26], which is known as a large 3D-object dataset. Out of the 133 mugs, we selected 100 mugs for training. We rendered the depth images from the CAD models with z-axis rotation as a simulation of observing real objects. The rendering was performed at rotation angles of $0°, 10°, \ldots, 350°$ for each of the 100 training objects, whose setting is the same as the COIL-20 dataset. As a result, we obtained 36 images $\times$ 100 objects for training.

B. Network Implementation

Although the proposed Object Manifold Embedding GAN (**Ω**-GAN) can make use of various kinds of GAN architectures, we implemented it based on Self-Attention GAN (SAGAN) [27]. SAGAN has self-attention layers to capture global information for generation/discrimination. It also employs spectral normalization to make the training more stable.

We modified the distribution of the random variables $\mathbf{z}$ for the generator input to the distribution over the product parameter manifold $\mathcal{M}$ which was defined in a 256-dimensional space $V = \mathbb{R}^{256}$. The pose manifold $\mathcal{M}_p$ was defined as a one-dimensional manifold in a two-dimensional space $V_p = \mathbb{R}^2$. To emphasize the pose parameter, we extended the pose parameter's dimension ten times by repeating the values and obtained a twenty-dimensional vector $\mathbf{z}_p$ from a pose parameter. On the other hand, the shape manifold $\mathcal{M}_s$ was defined in a 236 $(= 256 - 2 \times 10)$ dimensional space $V_s = \mathbb{R}^{236}$, the remaining part of V. The dimension of V was tuned empirically.

For the training of the proposed **Ω**-GAN, we ran 5,000 epochs.

C. Qualitative Evaluation: Image Generation Results

To confirm that the **Ω**-GAN can interpolate the training data, we trained the proposed **Ω**-GAN with the COIL-20 dataset. The transition of the adversarial loss of the proposed **Ω**-GAN is shown in Fig. 8. From the graph, we can see that the training of the GAN converged. We then generated fifty images by changing the pose parameter in $[0°, 360°)$ with an interval of $7.2°$ while fixing the shape parameter. Examples of the generated images are shown in Fig. 9. This result shows that the proposed **Ω**-GAN can successfully separate the object pose and shape.

A comparison with the existing methods is shown in Fig. 1. We generated twelve images by changing the pose parameter

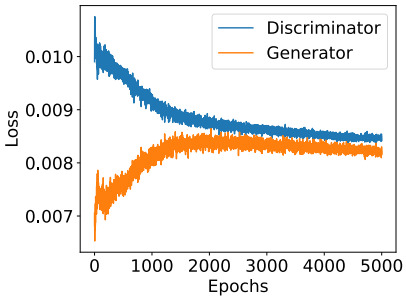


Fig. 8. Adversarial loss for the $\boldsymbol{\Omega}$-GAN training.

with an interval of 30° while fixing the shape parameters. By comparing Fig. 1 (i) and (ii), we can see that the Conditional GAN with the continuous condition could interpolate the pose well; however, the shape sometimes changed. On the other hand, the proposed $\boldsymbol{\Omega}$-GAN maintained their shape while changing only the pose parameters, as shown in Fig. 1 (iii).

For the "Mug" dataset, we also generated 100 images by changing the pose parameter in $[0°,\ 360°)$ with 3.6° while fixing the shape parameter. The result is shown in Fig. 10.

Fig. 11 shows the generated images by changing the shape parameters while fixing the pose parameter. We can see that the pose and shape variations were successfully disentangled. These results show that the proposed $\boldsymbol{\Omega}$-GAN can successfully separate the object pose and shape even though the object shape variation is small.

As the proposed $\boldsymbol{\Omega}$-GAN is based on the simple SAGAN, the generated images were not in good quality. However, if we used a more sophisticated GAN, we will surely be able to generate more realistic images.

VI. Quantitative Evaluation: Application to Object Pose Estimation

To quantitatively evaluate the objects' pose in the generated images for our purpose, we use the generated images for data augmentation in the training of an object pose estimator. For the object pose estimator training, if the poses of the augmented (added) images generated by the GAN are accurate, the trained pose estimator should be more accurate. Note that the generated images by GANs are usually evaluated by several metrics such as Inception score [28] and Fréchet Inception Distance (FID) [29] quantitatively. While these scores aim to evaluate the quality of the generated images, the goal is to generate realistic images in the targeted pose controllably in this paper. Thus, the accuracy of the object's poses in the generated images is our main interest, but such existing evaluation metrics cannot evaluate such accuracy of the pose of generated images.

A. Dataset

For the evaluation, we used the "Mug" dataset from the ShapeNet dataset [26]. Out of the 133 mugs, we selected 100 mugs for training data and 33 mugs for testing data.

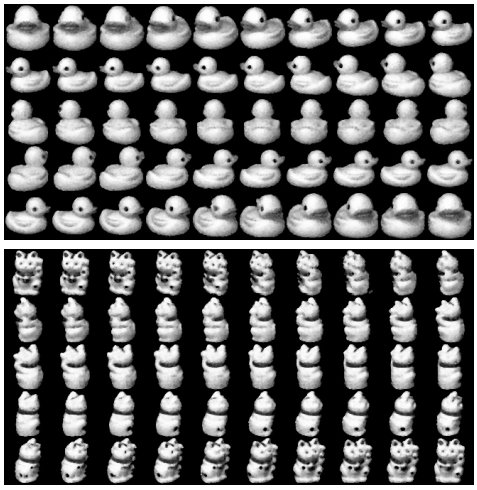


Fig. 9. Examples of the generated images by the proposed $\boldsymbol{\Omega}$-GAN trained with the COIL-20 dataset [25] using pose parameters sampled from $[0°, 360°)$ with an interval of 7.2° and two different shape parameters.

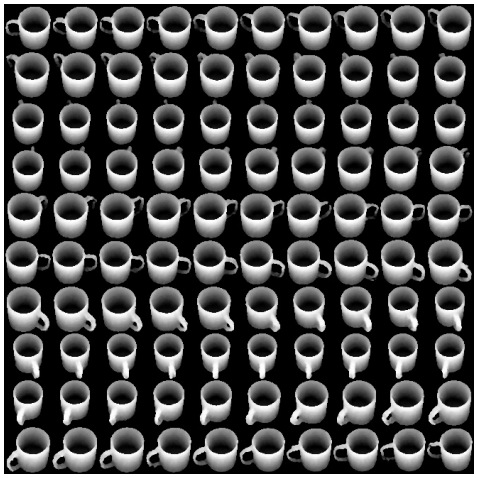


Fig. 10. Examples of the generated depth images by the proposed $\boldsymbol{\Omega}$-GAN. From the upper left to the lower right in a raster scan manner, images were generated by changing only the pose parameter in $[0°, 360°)$ with an interval of 3.6°.

Fig. 11. Examples of the generated images in various shapes by the proposed $\boldsymbol{\Omega}$-GAN.

For the evaluation of the pose estimation by using unknown shapes and poses, the rendering was performed at rotation angles of $0°, 30°, \ldots, 330°$ for each of the 100 training objects, and at $15°, 45°, \ldots, 345°$ for each of the 33 testing objects. By changing the observation's elevation angle at $\phi = 0°, 30°, 45°, 60°$, we generated four datasets. Here, in case of the elevation angle of $\phi = 0°$, the objects were observed from the side, and larger elevation angles indicated observing the object from higher angles. The rendering environment is illustrated in Fig. 12.

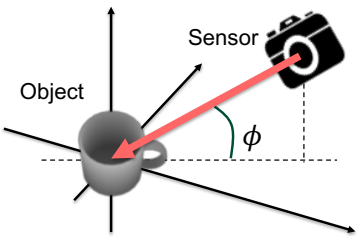


Fig. 12. Rendering environment and the elevation angle.

TABLE I
POSE ESTIMATION RESULTS (MEAN ABSOLUTE ERROR).

(i) By elevation angle ("Mug" class)

Elevation angle	60°	45°	30°	0°
Baseline	11.43	17.63	18.03	21.88
DA	10.37	18.02	17.83	21.99
Proposed ($\boldsymbol{\Omega}$-GAN)	**6.68**	**12.35**	**16.94**	**19.64**

(ii) By object class (Elevation angle 60°)

Object class	Mug	Car	Bike	Chair
Baseline	11.43	23.35	16.02	4.37
DA	10.37	14.58	18.72	3.77
Proposed ($\boldsymbol{\Omega}$-GAN)	**6.68**	**4.76**	**9.05**	**2.66**

B. Comparative Methods

There is a large number of methods for object pose estimation from an image. Currently, deep learning-based approaches are actively developed [30], [31], [32], [33]. Ninomiya et al. [34] proposed the Pose-CyclicR-Net for regression to cyclic objective variables using a convolutional neural network. The network outputs quaternion as the pose representation to handle the pose circularity.

We used a modified version of this Pose-CyclicR-Net as a baseline method. This network receives a depth image and outputs its pose parameter in a $(\cos\theta, \sin\theta)$ format, where θ is the rotation angle from the reference pose, instead of the quaternion representation as our dataset is restricted to the single-axis rotation.

For comparison, we trained the modified Pose-CyclicR-Net using different data as follows:

- **Baseline**: Trained with the original depth images only.
- **DA**: Trained with naïve data augmentation using the original depth images.
- **Proposed**: Trained with the original depth images and generated depth images by the proposed $\boldsymbol{\Omega}$-GAN.

In the case of DA, the original images were randomly shifted at most 10% of the image size and zoomed in the range of $[0.9, 1.1]$ and fed to the network. In the case of Proposed, the original images X_{real} and the generated images X_{fake}, which were randomly generated by the proposed $\boldsymbol{\Omega}$-GAN, were used equally.

The original images X_{real} are annotated with the ground-truth poses Y_o. On the other hand, for each generated image in X_{fake}, the pose parameter $\mathbf{z}_p$ was used as the ground-truth pose. For training the pose estimator with various training samples, the parameters $\mathbf{z}_p$ and $\mathbf{z}_s$ were randomly sampled from the distributions over the pose and the shape manifolds.

The training procedure of the pose estimator is as follows. First, N_o images for a mini-batch were sampled from the original images X_{real}. Then, the same number of images in a batch were randomly generated by the generator G of the proposed $\boldsymbol{\Omega}$-GAN and added to the mini-batch. Here, we applied a median filter to all the generated images to reduce the small noises. The training was repeated for 500 epochs.

C. Pose Estimation Results

The mean absolute error of the pose estimation results considering the pose circularity, *e.g.* the error between 5° and 355° is 10°, are shown in Table I (i).

For all the elevation angles, we confirmed that the proposed method, which is based on training with the images generated by the proposed $\boldsymbol{\Omega}$-GAN, achieved the best performance. This is because the proposed $\boldsymbol{\Omega}$-GAN successfully generated the poses and shapes not included in the training data. As the data captured from the small elevation angles were observed almost from the side, they contained depth images that were difficult to distinguish. Therefore, the pose estimation errors were relatively higher than in other situations.

We also evaluated the pose estimation accuracy for other object classes using the proposed $\boldsymbol{\Omega}$-GAN. As with the "Mug" models, we also prepared several models such as "Car," "Bike," and "Chair" selected from the ShapeNet dataset [26]. They were rendered from the elevation angle of $\phi = 60°$. We trained the proposed $\boldsymbol{\Omega}$-GAN for each object class. The evaluation results are shown in Table I (ii). We confirmed that the proposed method is also effective for them.

From the results, we confirmed that the pose estimation results improved even though the generated depth images are not in high quality. If the image generation's quality is improved, we can expect that the pose estimation accuracy would also improve.

VII. CONCLUSION

We proposed the Object Manifold Embedding GAN ($\boldsymbol{\Omega}$-GAN) that generates an image from a distribution in the pose and the shape manifolds. The generator of the proposed $\boldsymbol{\Omega}$-GAN maps the parameters on these manifolds to images. For clearly disentangling these parameters, we also introduced Object Identity Loss to preserve the object instance's shape when only the pose parameter is changed.

We confirmed that the proposed $\boldsymbol{\Omega}$-GAN could generate realistic images according to the pose and shape parameters through evaluation. For evaluating the pose accuracy of the generated images, we trained an object pose estimator with the generated images as data augmentation. We confirmed that the pose estimator trained with the generated images achieves

improved pose estimation accuracy compared to that trained with only the training images and naïve data augmentation, that is, the poses of the generated images are accurate enough for training an object pose estimator.

In the current work, the objects' rotaiton is restricted to a single-axis rotation; the generator of the Ω-GAN samples a pose parameter from a distribution on a unit circle in the two-dimensional space. We plan to extend this to sample from a unit hypersphere, namely the two- or three-dimensional manifold, to handle more complicated rotations such as around two-dimensional or three-dimensional axes in the future.

ACKNOWLEDGMENT

Parts of this research were supported by MEXT (17H00745), Grant-in-Aid for Scientific Research.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Adv. in Neural Info. Process. Syst. 27*, Dec. 2014, pp. 2672–2680.

[2] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Comput. Res. Repos.*, no. arXiv:1411.1784, Nov. 2014. [Online]. Available: http://arxiv.org/abs/1411.1784

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

[4] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *Comput. Res. Repos.*, no. arXiv:1701.07875, Jan. 2017. [Online]. Available: http://arxiv.org/abs/1701.07875

[5] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," *Comput. Res. Repos.*, no. arXiv:1703.10717, Mar. 2017. [Online]. Available: http://arxiv.org/abs/1703.10717

[6] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE Access*, vol. 7, pp. 36 322–36 333, Mar. 2019.

[7] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *Comput. Res. Repos.*, no. arXiv:1511.06434, Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.06434

[8] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image translation with conditional adversarial networks," *Comput. Res. Repos.*, no. arXiv:1611.07004, Nov. 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[9] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. 15th Int. Conf. on Comput. Vision*, Oct. 2017, pp. 2242–2251.

[10] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," *Comput. Res. Repos.*, no. arXiv:1610.09585, Oct. 2016. [Online]. Available: http://arxiv.org/abs/1610.09585

[11] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Adv. in Neural Info. Process. Syst. 29*, Dec. 2016, pp. 2172–2180.

[12] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. 2019 IEEE Conf. on Comput. Vision and Pattern Recognit.*, Jun. 2019, pp. 4401–4410.

[13] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato, "Fader Networks: Manipulating images by sliding attributes," in *Adv. in Neural Info. Process. Syst. 30*, Dec. 2017, pp. 5967–5976.

[14] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3D models from single images with a convolutional network," in *Proc. 14th European Conf. on Comput. Vision*, vol. 7, Oct. 2016, pp. 322–337.

[15] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang, "HoloGAN: Unsupervised learning of 3D representations from natural images," in *Proc. 17th Int. Conf. on Comput. Vision*, Oct. 2019, pp. 7588–7597.

[16] B. Liu, X. Wang, M. Dixit, R. Kwitt, and N. Vasconcelos, "Feature space transfer for data augmentation," in *Proc. 2018 IEEE Conf. on Comput. Vision and Pattern Recognit.*, Jun. 2018, pp. 9090–9098.

[17] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *Comput. Res. Repos.*, no. arXiv:1611.02163, 2016. [Online]. Available: http://arxiv.org/abs/1611.02163

[18] C. Xiao, P. Zhong, and C. Zheng, "BourGAN: Generative networks with metric embeddings," in *Adv. in Neural Info. Process. Syst. 31*, Dec. 2018, pp. 2269–2280.

[19] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proc. 2019 IEEE Conf. on Comput. Vision and Pattern Recognit.*, Jun. 2019, pp. 1429–1437.

[20] Z. Shu, M. Sahasrabudhe, R. Alp Güler, D. Samaras, N. Paragios, and I. Kokkinos, "Deforming autoencoders: Unsupervised disentangling of shape and appearance," in *Proc. 15th European Conf. on Comput. Vision*, Sep. 2018, pp. 650–665.

[21] X. Xing, T. Han, R. Gao, S.-C. Zhu, and Y. N. Wu, "Unsupervised disentangling of appearance and geometry by deformable generator network," in *2019 IEEE/CVF Conf. on Comput. Vision and Pattern Recognit.*, Jun. 2019, pp. 10 346–10 355.

[22] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proc. 2019 IEEE/CVF Conf. on Comput. Vision and Pattern Recognit.*, Jun. 2019, pp. 5738–5746.

[23] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "Siamese" time delay neural network," in *Adv. in Neural Info. Process. Syst. 6*, Dec. 1993, pp. 737–744.

[24] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. 2005 IEEE Conf. on Comput. Vision and Pattern Recognit.*, Jun. 2005, pp. 539–546.

[25] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Department of Computer Science, Columbia University, Tech. Rep. CUCS-005-96, Feb. 1996.

[26] 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," *Comput. Res. Repos.*, no. arXiv:1512.03012, Dec. 2015. [Online]. Available: http://arxiv.org/abs/1512.03012

[27] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *Comput. Res. Repos.*, no. arXiv:1805.08318, May 2018. [Online]. Available: https://arxiv.org/abs/1805.08318

[28] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Adv. in Neural Info. Process. Syst. 29*, Dec. 2016, pp. 2234–2242.

[29] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Adv. in Neural Info. Process. Syst. 30*, Dec. 2017, pp. 6626–6637.

[30] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *Proc. 16th IEEE Int. Conf. on Comput. Vision*, Oct. 2017, pp. 1530–1538.

[31] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *Proc. 2015 IEEE Int. Conf. on Robot. and Autom.*, May 2015, pp. 1329–1335.

[32] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3D pose estimation," in *Proc. 2015 IEEE Conf. on Comput. Vision and Pattern Recognit.*, Jun. 2015, pp. 3109–3118.

[33] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," *Comput. Res. Repos.*, no. arXiv:1711.00199, Nov. 2017. [Online]. Available: https://arxiv.org/abs/1711.00199

[34] H. Ninomiya, Y. Kawanishi, D. Deguchi, I. Ide, H. Murase, N. Kobori, and Y. Nakano, "Deep manifold embedding for 3D object pose estimation," in *Proc. 12th Joint Conf. on Comput. Vision, Imaging and Comput. Graphics Theory and Appl.*, Feb. 2017, pp. 173–178.