

# Self-supervised Learning of 3D Objects from Natural Images

Hiroharu Kato<sup>1</sup> and Tatsuya Harada<sup>1,2</sup>

<sup>1</sup>The University of Tokyo, <sup>2</sup>RIKEN

{kato, harada}@mi.t.u-tokyo.ac.jp

## Abstract

We present a method to learn single-view reconstruction of the 3D shape, pose, and texture of objects from categorized natural images in a self-supervised manner. Since this is a severely ill-posed problem, carefully designing a training method and introducing constraints are essential. To avoid the difficulty of training all elements at the same time, we propose training category-specific base shapes with fixed pose distribution and simple textures first, and subsequently training poses and textures using the obtained shapes. Another difficulty is that shapes and backgrounds sometimes become excessively complicated to mistakenly reconstruct textures on object surfaces. To suppress it, we propose using strong regularization and constraints on object surfaces and background images. With these two techniques, we demonstrate that we can use natural image collections such as CIFAR-10 and PASCAL objects for training, which indicates the possibility to realize 3D object reconstruction on diverse object categories beyond synthetic datasets.

## 1. Introduction

By looking at an object at a glance, we humans can understand its 3D shape, orientation, and appearance on surfaces. Implementing this ability in machines, known as single-view 3D object reconstruction and object pose estimation in computer vision, has many practical applications such as robot grasping and augmented reality. Since this is a severely ill-posed problem, learning and leveraging the prior knowledge of objects is the key to this task.

Most works in this field use ShapeNet [2], a large-scale 3D CAD dataset, for training. Though recent technical advancement has realized to generate a high-quality 3D model from an image in object categories that are contained in ShapeNet [4, 7, 9, 18, 42, 46], because creating a large amount of 3D models is very costly, 3D object reconstruction in more diverse categories beyond ShapeNet is difficult. While the majority of methods use 3D shapes as supervision [4, 7, 9], several works aim to reduce 3D supervision by using 2D images for training [18, 42, 46]. However, they

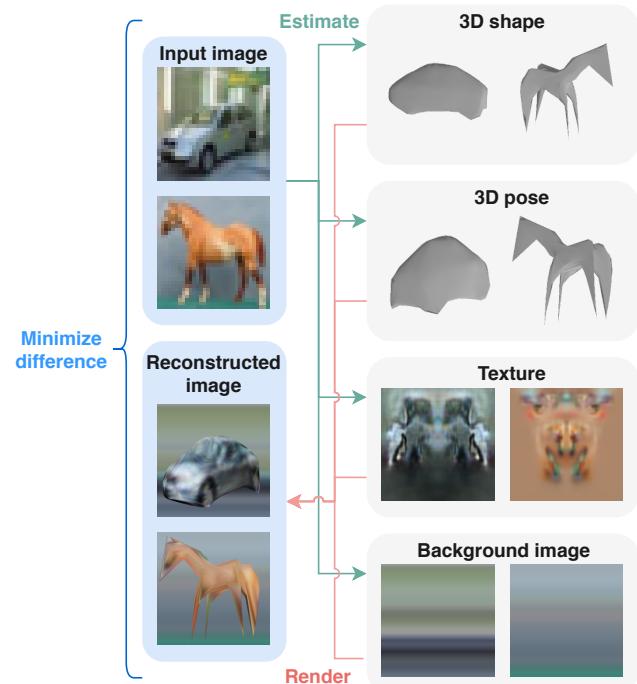


Figure 1. Given an object image, our proposed model estimates its 3D shape, pose, texture, and background. Only categorized object images are required for training. This figure shows the system architecture and result of our model on CIFAR-10.

typically require images with foreground masks of objects, which are still not easy to obtain.

In this work, we demonstrate that 3D shape, pose, and texture can be learned from categorized natural images without supervision. Fig. 1 shows our result on the test set of CIFAR-10 dataset [19]. This dataset of categorized natural images has difficulties in several aspects. Ground-truth 3D shapes are not given, there are no multiple views of the same object, foregrounds and backgrounds are not separated, viewpoints are unknown, objects have various shapes and sizes, and they locate and rotate freely. Success in estimating 3D elements on this challenging dataset indicates the possibility to leverage natural images for 3D understanding and realize 3D object reconstruction on diverse object categories.

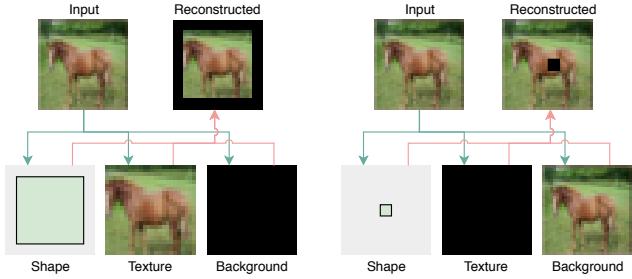


Figure 2. Examples of trivial and poor solutions. We know these are unrealistic, however, neural networks cannot know it by self-supervision. Therefore, we induct several kinds of structural knowledge of 3D scenes into our model.

We train this model by comparing input images with reconstructed images. Given an image, 3D shape, pose, texture image, and background image are estimated by neural networks. Then, an image is rendered using these estimated elements. Reconstruction error is computed by extracting and comparing features of the input image and reconstructed image, and neural networks are optimized by minimizing it. Though this framework is quite simple, a way to obtain a meaningful model is not straightforward because this problem has several trivial and poor solutions, as depicted in Fig. 2. One example is to expand an object to cover the whole image and copy the input image into the texture of the object. Another example is to shrink an object under one pixel and copy the input image into the background. Though image reconstruction is almost perfectly in both cases, we know that this reconstruction is unlikely as realistic 3D scenes. The technical key point of this paper is to induct such knowledge about 3D structures into neural networks in the form of training methods, constraints, and regularization. Our main assumptions are (1) all shapes in the same object category are similar, and they can be made by slightly deforming a category-specific base shape, (2) surfaces of objects are smooth, and (3) background images are sufficiently simple. Based on these assumptions, we carefully design constraints and regularization, and separate training steps into category-specific base shape generation and full training given a base shape. While these assumptions are not always correct in real scenes, they are practically useful for training, as we demonstrate in experiments using CIFAR-10 and PASCAL objects [45].

The major contributions can be summarized as follows.

- To the best of our knowledge, this is the first study to train single-view reconstruction of 3D shape, pose, and texture by using only categorized natural images.
- We demonstrate that self-supervised learning can be achieved by (1) training shapes first and subsequently training poses and textures using the obtained shapes, and (2) introducing regularization and constraints of shapes, textures, and backgrounds.

Supervision	[28]	[40]	[13]	[50] <sup>†</sup>	[25] <sup>‡</sup>	ours
Natural images			✓	(✓)	✓	✓
Viewpoint-free	✓	✓	✓	(✓)	✓	✓
Silhouette-free				(✓)	✓	✓

Table 1. Works that aim supervision reduction in single-view training of single-view 3D object reconstruction. *Silhouette-free* includes works that use images without backgrounds. <sup>†</sup>Zuffi *et al.* [50] leverages simulators. <sup>‡</sup>Nguyen-Phuoc *et al.* [25] cannot represent 3D shapes explicitly.

## 2. Related work

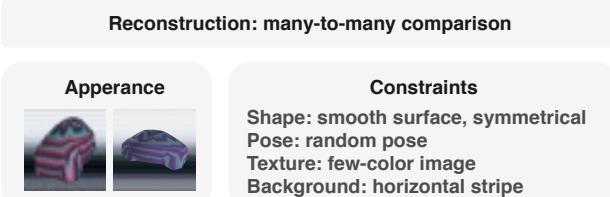
There are a vast number of problems and approaches in 3D reconstruction since this is a long-standing topic in computer vision. In this section, we focus on supervision for single-view 3D object reconstruction.

**3D supervision** When plenty of 3D object models are available, using them as training signals would be the best option because that approach does not suffer from shape ambiguities found in 2D images. This approach has been popularized with the advent of large 3D shape datasets, such as ShapeNet [2]. One research direction is how neural networks handle irregular 3D representations, such as high-resolution voxels [11, 37], point clouds [7], meshes [9, 43] and implicit functions [3, 24, 26]. Another path is the generalization of learning algorithms to novel objects [34, 38, 49].

**Multi-view training** Because a 3D shape is understandable from its multiple 2D projections, object silhouettes from multiple viewpoints have been used as alternative training signals. Different from 3D supervision, this view supervision requires a differentiable 3D-to-2D projection module. Therefore, several differentiable projection modules have been developed for voxels [42, 46], meshes [18, 22], point clouds [14], and implicit functions [23, 35]. Though annotation cost is lower than 3D supervision, multiple views are still costly because collecting them requires a specialized 3D capture system.

**Single-view training** Training using image collections would be the lowest cost choice. However, since it is not an easy task, additional supervision is typically required. Kar *et al.* [16] demonstrated that 3D shapes and viewpoints can be recovered from natural images when silhouettes and keypoints of objects are available. Kanazawa *et al.* [15] translated this framework into neural networks and incorporated texture prediction in addition. Tulsiani *et al.* [42] applied their multi-view training method that uses silhouette and viewpoint supervision onto a single-view dataset. Later, they relaxed this dataset requirement by integrating pose prediction [40]. Kato and Harada [17] demonstrated that a similar approach tends to result in unrealistic-looking shapes and improved it by adversarial training.

## First step: generating a base shape



## Second step: full training with the base shape

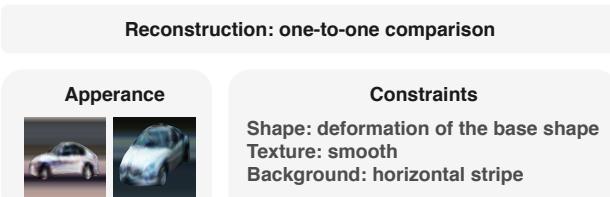


Figure 3. Training steps and constraints in our proposed method.

Rezende *et al.* [28] trained 3D structure from images, however, their dataset is composed of simple primitives without backgrounds. Henzler *et al.* [13] trained 3D object reconstruction from natural images with silhouette annotations. Nguyen-Phuoc *et al.* [25] learned implicit neural 3D representation from natural images. Zuffi *et al.* [50] leveraged simulators to learn rare 3D objects. Contrary to these works, our method can learn explicit 3D structures from natural images, and it does not require any supervision except for categorized object images. Table 1 shows a summary.

## 3. Method

Our method trains single-view reconstruction of 3D shape, pose, texture, and background with self-supervision as shown in Fig. 1 while avoiding unrealistic solutions like those shown in Fig. 2. One difficulty is training all elements at the same time because neural networks easily fall in the easiest solution of copying an input image into pixel arrays (textures or backgrounds). Therefore, we propose a two-stage training method that focuses on shapes first. Fig. 3 illustrates the overview of our proposed approach. In the first step, a category-specific 3D base shape is generated by maximizing the similarity between images in a dataset and images of the shape. We use randomly sampled viewpoints and strongly limited textures. In the second step, the whole model is trained limiting generated shapes to deformations of the obtained base shape. Another difficulty is that shapes and backgrounds sometimes become excessively complicated to mistakenly reconstruct textures on object surfaces. To suppress it, we propose using strong regularization and constraints on object surfaces and background images. We use a mesh as a 3D representation to introduce constraints and regularization on texture and surfaces.

### 3.1. Learning category-specific base shape

In training a category-specific base 3D shape, we strongly limit the representation capacity of textures in order to focus on shapes. Because this limitation makes it impossible to reconstruct images with results close to the input images, adoption of an auto-encoder architecture in Fig. 1 is infeasible. Instead, as shown in Fig. 4, we propose a model that generates a shape, texture, and background from random noise by minimizing the difference between the set of rendered images and the set of images in a dataset. In the following sections, we explain each component along with the additional constraints and regularization needed to obtain a meaningful shape.

#### 3.1.1 Shape generation

We generate a shape by deforming vertices of a pre-defined sphere, as was done in several existing works [15, 18, 43]. Additionally, we manually set an initial dimension of the sphere in each category. With a pre-defined shape of  $N_v$  vertices,  $N_v \times 3$  variables are generated by a neural network and added to vertex coordinates in 3D space. Then, the generated shape is scaled to fit into a unit cube. In addition, we employ the following constraints and regularization.

**Smoothness of objects** Because the representation capacity of textures is limited, the generated shapes try to be very complicated in order to represent edges in images. However, most of the edges in natural images are actually caused by textures or backgrounds, not by shapes. Therefore, we assume that the surfaces of objects are smooth and regularize curvature of them, which is a common approach in modeling object surfaces [1, 15]. Specifically, we minimize graph Laplacian of a mesh, which represents approximated mean curvature at each vertex [39], and angles between two neighboring triangle polygons, which implies smoothness at each edge. We denote this loss term as  $\mathcal{L}_s$ . More details are in the appendix.

**Symmetry of objects** Though object shapes in natural images are not always symmetrical (e.g. horses), category-specific base shapes are often symmetrical (e.g. the average shape of horses). Therefore, we constrain generated shapes to be symmetrical.

#### 3.1.2 Texture generation

We assume that UV-mapping of a texture image and surface is pre-defined and fixed during training. To generate a texture image, we employ DCGAN [27]-like architecture with residual connections [12].

**Simplicity of object textures** To reduce the representation capacity of texture images, we propose using a single color or only a few colors when making a texture image. A

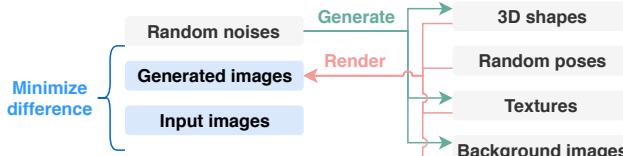


Figure 4. Our proposed model for learning category-specific base shapes. To focus on learning shapes, viewpoints are randomly sampled from a fixed distribution, and the representation capacity of textures is limited.

similar assumption is used for reflectance maps in intrinsic image decomposition [1]. Specifically, instead of generating a three-channel RGB image, we generate a  $N_c$ -channel image by a neural network and normalize each pixel, so that the sum of the channel values is one. Additionally, a color palette of  $N_c$  colors is generated by another neural network. Then, an RGB image is generated by mixing the  $N_c$  colors according to the  $N_c$ -channel image. Though the representation capability of this reparameterization is the same as the original network when  $N_c \geq 3$ , it generates few-color images in practice.

### 3.1.3 Pose generation

To represent the 6DoF pose of an object, we assume that a camera is always directed to the center of the object, the upward direction of the camera is always  $(x, y, z) = (0, 1, 0)$ , the distance between the object and the camera is fixed, and only azimuth and elevation of viewpoints can be changed. During training, viewpoints are sampled randomly from a manually-designed viewpoint distribution.

### 3.1.4 Background generation

To prevent the solution shown in the right of Fig. 2, we need to limit the representation capability of background images. Therefore, we introduce the following constraint.

**No vertical lines in backgrounds** The representation capacity of a background image must be high enough to express the structure of a scene, however, it must not be too high in order not to represent foreground objects. To achieve this, we constrain background images to be horizontal stripes without vertical lines. Images with this constraint can express the rough scene structure, such as the sky and grasses, however, they cannot express objects.

### 3.1.5 Rendering

We render an image using a generated shape, pose, texture, and background with random directional lighting and smooth shading. Using light is essential to express shapes

with limited few-color textures. We use a differentiable renderer developed by Kato *et al.* [18] to back-propagate the gradient from the loss function into generators.

### 3.1.6 Comparison between real and generated images

Since one-to-one comparison of real and generated images is impossible, we match distributions of them similar to generative adversarial networks (GANs) [8]. However, adversarial training used in GANs does not work well for our problem because the limited representation capacity of the image generator makes the minimax game too advantageous for discriminators. Instead, we use feature matching [30] and the Chamfer distance of real and generated image minibatches to compute a reconstruction loss  $\mathcal{L}_{\text{rec}}$ . More details are in the appendix.

### 3.1.7 Summary and post-processing

In this step, to obtain a category-specific base shape, a shape generator, a texture generator, and a background generator are trained by minimizing the sum of reconstruction loss  $\mathcal{L}_{\text{rec}}$  and smoothing loss  $\mathcal{L}_s$  under the constraints of shape symmetricity, and texture and background simplicity. Though the input is random noise, generated shapes converge to a single shape after training, which is similar to mode collapse in GANs.

For texture mapping, a smaller variance of polygon sizes is better. To accomplish this, we use silhouettes of the obtained mesh to generate another mesh by minimizing several factors: the difference of the silhouettes, the variance of sizes of the triangle polygons, and the whole area of the surfaces. This post-processing significantly reduces the variance of polygon sizes while maintaining the whole shape.

## 3.2 Full training with base shape

In the second step, we train the pipeline in Fig. 1 while limiting the generated shapes to deformations of a category-specific base shape. We use encoder-decoder architecture for shape, pose, texture, and background prediction. We use a texture generator without the few-color constraint because providing base shapes prevents results like those found in the left of Fig. 2. In addition, we regularize the total variation [29] of textures to reduce noise. We also use the same background generator as we did for the previous step so as to prevent results like those found in the right of Fig. 2. We render images without using directional lighting because textures are able to represent shadings in this step.

### 3.2.1 Shape prediction

Instead of predicting a mesh directly, we predict shape deformations using free-form deformation [33] similar to several other object reconstruction works [21, 47]. We use a

spatial grid of  $4 \times 4 \times 4$  vertices, and regress the difference between the original grid and a deformed grid using a neural network. In addition, we use another network to regress the relative height, width, and length of shapes. After deformation, the size of the predicted shape is scaled to fit a unit cube.

**Exploring best shape** The variation between generated shapes tends to be very small because exploring various shapes using only a differentiable renderer and gradient descent is difficult due to local minima. To overcome this problem, we explore and record the best shape for each input image at each training iteration. Specifically, we render images using an estimated shape, a recorded best shape, a slightly perturbed the best shape, and random shapes. Then, we compute reconstruction loss to find the best one and record it.

### 3.2.2 Pose prediction

In this step, we parameterize the 6DoF object/camera pose by azimuth and elevation as with Section 3.1.3, in-plane rotation of an object, center point of an object in 2D image coordinates, and scale of an object. We train a decoder that outputs these six parameters. We adopt multiple regressor approach used in [14].

**Exploring best pose** Similarly to shape prediction, we also need to actively explore the best poses. At each training iteration, we explore and record the best pose for each input image by rendering images using estimated, recorded, random, and perturbed poses.

### 3.2.3 Training

In addition to the components described above, we employ view prior learning (VPL) [17] to reduce overfitting to the observed views. Summarily, a loss function is composed of the following four terms. (1) Reconstruction loss. Reconstructed images using the best shapes, estimated textures, the best poses, and estimated backgrounds are compared with input images. In addition, feature matching is also used. (2) Mean absolute error between estimated shapes/poses and the best shapes/poses that are recorded during training. (3) Total variation of estimated texture images for denoising. (4) VPL loss. To facilitate an early phase of training, at the  $i$ -th iteration, training samples are randomly selected from first to  $i$ -th data in the dataset. This makes the model see the same sample frequently in an early stage, which simplifies finding the best poses and makes the estimated poses diverse.

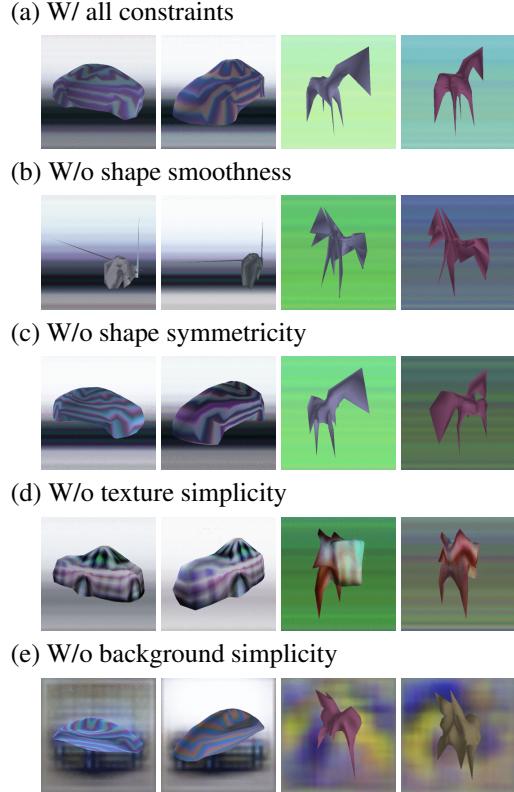


Figure 5. Generated category-specific base shapes on CIFAR-10 dataset. These images are rendered in  $256 \times 256$  resolution with upsampled background images. (a) shows a result of our proposed method, and (b–e) are ablation studies that clarify contributions of introduced constraints and regularization.

### 3.2.4 Photometric per-instance fine-tuning

In inference, similar to [50], we slightly adjust predictions by optimizing the outputs of encoders to minimize the reconstruction loss. This is possible because we do not need silhouette or viewpoint annotations to compute the loss. We successively optimize the outputs of the background encoder, pose encoder, and shape decoder.

## 4. Experiments

### 4.1. CIFAR-10

We mainly tested our method on the CIFAR-10 [19] dataset because it is composed of natural images and contains thousands of images per object category. Among ten object categories, we focused on *car* and *horse* classes because *car* is an artificial and rigid object and one of the most commonly used categories on the synthetic ShapeNet dataset [2] and *horse* is a deformable natural object not contained in ShapeNet. For feature extraction, we trained WRN-16-4 [48] on the CIFAR-10 training set. We used three layers right before sub-sampling as feature maps.

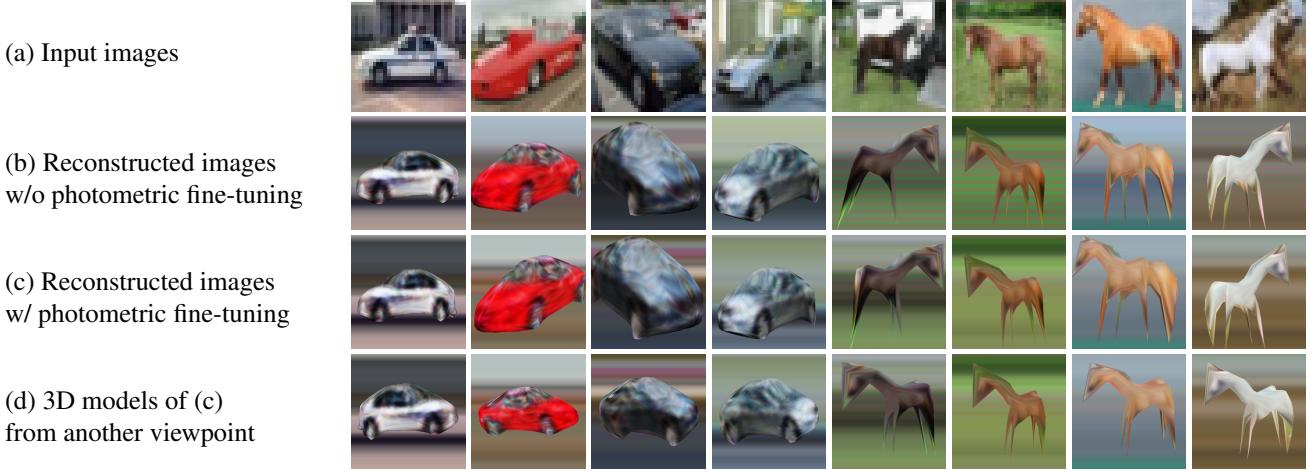


Figure 6. Representative results of 3D shape, pose, texture, and background estimation on CIFAR-10 test set. To understand the shapes and textures better, images are rendered at a higher resolution with upsampled backgrounds. Since the input images (a) are explicitly disentangled into 3D object elements, objects can be rendered from another viewpoint (d). Randomly selected results are in the appendix.

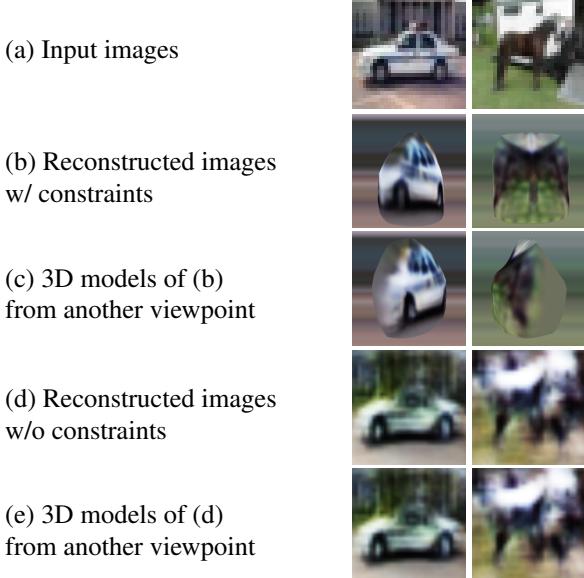


Figure 7. Training without our proposed two-stage training. (b–c) and (d–e) correspond to the left and right of Fig. 2 respectively. These results confirm the importance of training shapes explicitly.

#### 4.1.1 Base shape learning

First, we evaluate the first step described in Section 3.1. We set the number of colors of *car* texture to four, and that of *horse* to one. We trained 10 models for each category using different random seed and selected the best-looking one. Fig. 5 (a) shows generated base shapes by our proposed method using different random noise. The generated shapes, textures, and backgrounds look plausible. Particularly, the *horse* correctly has four legs, and the *car* has four

tires on texture. The background image of *car* represents the sky as a bright region and roads as dark, and the background of *horse* shows grasses. This result indicates that the generators work properly.

Fig. 5 (b–e) shows ablation study. When the regularization of shape smoothness is removed, thin lines are generated to represent edges, which results in unrealistic shapes (b). The shape symmetricity constraint seems unimportant for *car*, but it helps to generate legs on *horse* regularly. (c). Even when the texture simplicity constraint is removed, the texture does not represent the whole scene as in the left of Fig. 2 because of constraints on shapes. However, the texture of *horse* contains the colors of horses and grasses, that results in the incorrect shape (d). When the background simplicity constraint is not used, the background generator tries to represent shapes, especially in *horse* (e). These results indicate introducing our knowledge about 3D scenes into a model is essential in self-supervised shape learning, and all of the constraints and regularization used are indispensable.

#### 4.1.2 Full training using base shapes

Secondly, we evaluate the second step using the base shapes obtained in the previous step. Fig. 6 shows representative results on the test set. Reconstructed images demonstrate that the estimators trained by our method are able to reconstruct images that look similar to input images (a–b). Estimated shapes, poses, and backgrounds can be further improved by simple gradient descent and photometric reconstruction loss (c). Rendered images from other viewpoints show that these objects have correct 3D shapes, which are slightly different among different input images (d).

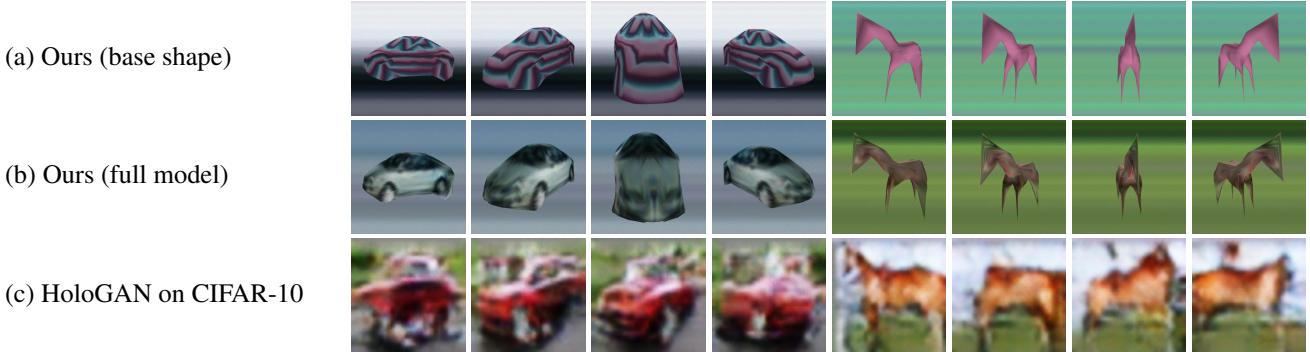


Figure 8. Comparison of ours with HoloGAN on CIFAR-10. Images are rendered at 50 degree intervals.

#### 4.1.3 Effectiveness of two-stage training

One of the most important techniques of our method is to separate training into two stages. To validate its effectiveness, we trained our models in a single stage. Fig. 7 shows the reconstruction results by the learned models. When our proposed constraints, such as surface smoothness and background simplicity, are used, the textures represent edges of shapes, which results in incorrect shapes (a–b). When these constraints are not used, because the background estimator copies input images, the reconstructed images look the same from any viewpoint (c–d). Apparently, neither model understands these 3D scenes correctly. These results correspond to the left and right of Fig. 2 respectively.

#### 4.1.4 Comparison with existing works

To the best of our knowledge, this is the first work to learn single-image reconstruction of 3D shape, pose, and texture from natural image collections without supervision. Therefore, we cannot conduct fair comparison between our work and existing works. One related approach would be structure-from-motion because it can recover an object shape from a photo collection. Therefore, we tested COLMAP [31, 32] on CIFAR-10, however, it failed to reconstruct a shape because it cannot find initial corresponding image pairs. Though slightly different from 3D reconstruction, HoloGAN [25] learns a generative model of images with implicit, but manipulable, 3D representation from natural images. Fig. 8 shows comparison between images of our base shapes and images by HoloGAN trained on CIFAR-10. While the shapes produced by our method are consistent from multiple viewpoints, the shapes produced by HoloGAN are not. This result implies the effectiveness of having explicit 3D representations.

### 4.2. PASCAL

We also evaluated our method on PASCAL dataset pre-processed by Tulsiani *et al.* [42]. This dataset contains

Method	Set	<i>airplane</i>	<i>car</i>	<i>chair</i>
With pose and keypoint supervision				
K&V [41]	validation	.81	.90	.80
Self-supervised				
Ours	training	.04	.71	.51
Ours	validation	.04	.65	.38

Table 2. Quantitative evaluation of pose estimation on the PASCAL 3D+ dataset. The used metric is  $\text{acc}_{\frac{\pi}{6}}$  in [41].

three object categories *aeroplane*, *car*, and *chair*, and it is composed of images from PASCAL VOC [6], additional images from ImageNet [5], and their shape and pose annotations provided by PASCAL 3D+ [45]. Object images were cropped using bounding boxes and resized to the same size. We used ResNet-18 architecture as encoders, and the same decoders as the CIFAR-10 experiments. As feature maps, we used five layers after convolution of pre-trained AlexNet [20]. Though this requires additional supervision provided by ImageNet, this would be replaceable by any kind of self-supervised representation learning methods. Since input images were cropped and resized using bounding boxes, we also cropped and resized rendered images. This reduced the degree of freedom of poses from six to three.

Fig. 9 shows obtained category-specific base shapes and Fig. 10 shows several results of single-view reconstruction by our fully-trained model on the validation set. These results demonstrate that our proposed method works properly on PASCAL dataset. Table 2 shows pose estimation accuracy of found best shapes during training on the training set and prediction by the pose estimator on validation set. For *car* and *chair*, our model found correct shapes during training in the majority of cases, and the pose estimator learned the correspondence of images and poses properly. However, our method cannot find poses of *aeroplane* because the ambiguity of poses is very high in this category. Though many of the reconstructed *aeroplane* images look plausible at a glance, the accuracy of pose estimation does not correlate



Figure 9. Generated base shapes of *aeroplane*, *car*, and *chair* on PASCAL dataset.



Figure 10. Representative results of 3D shape, pose, texture, and background estimation on PASCAL validation set. Randomly sampled results are in the appendix.

with this intuitive evaluation.

### 4.3. Discussion

Though we believe that this work is an important step toward 3D understanding without supervision, as the task is very challenging, the accuracy of our method is not very high. In this section, we list our observations in the experiments and possible solutions. Please see the appendix for more qualitative results.

- Our method works well for estimating rough shapes and poses, however, it is not very good at reconstructing small details (cf. legs of *horses*) and estimating accurate poses in ambiguity (cf. *aeroplane*). Also, the intra-class variance of generated shapes is very small. One reason for these problems is that the reconstruction loss using pre-trained image features does not capture category-specific fine-grained features. Actually, a reliable measure of image reconstruction is quite important in self-supervised learning based on render-and-compare loss because reconstruction loss is the only supervision. Incorporating unsupervised learning of keypoints [36] would alleviate this problem.
- We introduced three assumptions in the introduction section. As demonstrated in experiments, these do not prevent learning in *car*, *horse*, *chair*, and *aeroplane* categories. However, our proposed method does not work well in *cat* and *dog* on CIFAR-10 because the deformation of shapes is relatively large. One possible way to learn large deformation would be using videos for training.

- We used manually-designed pose distributions in the base shape learning. However, designing them is not straightforward in some categories. For example, the viewpoint distribution of *horse* images are far from uniform because there are many photos of zoom up of heads, but fewer photos of tails. How to deal with biased distributions would be an important and interesting problem.

- The introduced constraints and regularization may sound too naive and intuitive. Actually, the proposed base shape learning is a bit sensitive to hyperparameters of surface smoothness. This is because of the difficulty to design a robust measure of object naturalness. Learning object naturalness from data [10, 44] would be a promising direction.

## 5. Conclusion

In this study, we presented a method to learn single-view reconstruction of the 3D shape, pose, and texture of objects from categorized natural images in a self-supervised manner. The two main techniques were two-stage training to focus on shapes, and inducing strong regularization and constraints to the surface of shapes and background images. Results of experiments on CIFAR-10 and PASCAL confirm the importance of our proposed techniques. In addition, we summarized observations and possible research directions.

## Acknowledgment

This work was partially supported by JST CREST Grant Number JPMJCR1403, and partially supported by JSPS KAKENHI Grant Number JP19H01115.

## References

- [1] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *PAMI*, 37(8):1670–1687, 2014. 3, 4
- [2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv*, 2015. 1, 2, 5
- [3] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 2
- [4] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 1
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 7
- [6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 7
- [7] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 1, 2
- [8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 4
- [9] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Atlasnet: A papier-mache approach to learning 3d surface generation. In *CVPR*, 2018. 1, 2
- [10] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *3DV*, 2017. 8
- [11] Christian Hane, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction. *TPAMI*, (1):1–1, 2019. 2
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3
- [13] Philipp Henzler, Niloy Mitra, and Tobias Ritschel. Escaping plato’s cave using adversarial training: 3d shape from unstructured 2d image collections. In *ICCV*, 2019. 2, 3
- [14] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *NeurIPS*, 2018. 2, 5
- [15] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 2, 3
- [16] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015. 2
- [17] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *CVPR*, 2019. 2, 5
- [18] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *CVPR*, 2018. 1, 2, 3, 4
- [19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 1, 5
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 7
- [21] Andrey Kurenkov, Jingwei Ji, Animesh Garg, Viraj Mehta, JunYoung Gwak, Christopher Choy, and Silvio Savarese. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. In *WACV*, 2018. 4
- [22] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. In *ICCV*, 2019. 2
- [23] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In *NeurIPS*, 2019. 2
- [24] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2
- [25] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019. 2, 3, 7
- [26] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2
- [27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 3
- [28] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *NIPS*, 2016. 2
- [29] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 4
- [30] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016. 4, 11
- [31] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 7
- [32] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 7
- [33] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. *ACM Transactions on Graphics*, 20(4):151–160, 1986. 4
- [34] Daeyun Shin, Charless C Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *CVPR*, 2018. 2
- [35] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 2

- [36] Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d key-points via end-to-end geometric reasoning. In *NeurIPS*, 2018. 8
- [37] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2017. 2
- [38] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, 2019. 2
- [39] Gabriel Taubin. A signal processing approach to fair surface design. In *SIGGRAPH*, 1995. 3, 11
- [40] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *CVPR*, 2018. 2
- [41] Shubham Tulsiani and Jitendra Malik. Viewpoints and key-points. In *CVPR*, 2015. 7
- [42] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 1, 2, 7
- [43] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 2, 3
- [44] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In *ECCV*, 2018. 8
- [45] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. 2, 7
- [46] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS*, 2016. 1, 2
- [47] M Ersin Yumer and Niloy J Mitra. Learning semantic deformation flows with 3d convolutional networks. In *ECCV*, 2016. 4
- [48] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 5
- [49] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In *NeurIPS*, 2018. 2
- [50] Silvia Zuffi, Angjoo Kanazawa, Tanja Berger-Wolf, and Michael J Black. Three-d safari: Learning to estimate zebra pose, shape, and texture from images” in the wild”. In *ICCV*, 2019. 2, 3, 5

## A. Appendix

### A.1. Implementation details

#### A.1.1 Smoothness regularization

As described in Section 3.1.1, we minimize graph Laplacian of a mesh, which represents approximated mean curvature at each vertex [39], and angles between two neighboring triangle polygons, which implies smoothness at each edge. These regularizations are represented by a loss term  $\mathcal{L}_s$ . For a mesh of  $N_v$  vertices, let  $L \in \mathcal{R}^{N_v \times N_v}$  be a Laplacian matrix of vertices  $v \in \mathcal{R}^{N_v \times 3}$  and let  $t_l \in \mathcal{R}^+$  be a hyper parameter that controls tolerance. We minimize  $\mathcal{L}_{g_2} = \sum_i \|L_i v\|_2^2$  and  $\mathcal{L}_{g_1} = \max((\sum_i \|L_i v\|_2^2) - t_l, 0)^2$ . Similarly, let  $\theta_{i,j}$  be the angle between normal vectors of two neighboring triangle polygons  $f_i$  and  $f_j$  and let  $t_a \in \mathcal{R}^+$  be a hyper parameter, we minimize  $\mathcal{L}_{a_2} = \sum_{i,j} \theta_{i,j}^2$  and  $\mathcal{L}_{a_1} = \max(\|\sum_{i,j} \theta_{i,j}\| - t_a, 0)^2$ . With weighting parameters  $\lambda_{g_1}, \lambda_{g_2}, \lambda_{a_1}, \lambda_{a_2}$ , the sum of these regularization terms is

$$\mathcal{L}_s = \lambda_{g_1} \mathcal{L}_{g_1} + \lambda_{g_2} \mathcal{L}_{g_2} + \lambda_{a_1} \mathcal{L}_{a_1} + \lambda_{a_2} \mathcal{L}_{a_2}. \quad (1)$$

#### A.1.2 Initial object dimensions

As described in Section 3.1.1, we manually give a rough shape in base shape learning by setting initial dimensions of a predefined sphere to alleviate the difficulty of learning aspect ratio of objects. Table 3 shows the setting. Our method worked well even though the specified values were very rough.

#### A.1.3 Viewpoint distribution

As described in Section 3.1.3, we use predefined viewpoint distributions in learning base shapes. Table 4 shows the setting.

#### A.1.4 Reconstruction loss

As described in Section 3.1.6, we use feature matching [30] and the Chamfer distance of real and generated image minibatches to compute a reconstruction loss  $\mathcal{L}_{\text{rec}}$  in the base shape learning. Let  $H, W$  be the height and width of images assuming all images are resized to the same size. Given  $N_c$ -channel feature maps of  $N_b$  real images  $f \in \mathcal{R}^{N_b \times N_c \times H \times W}$  and  $N_b$  generated images  $\hat{f}$ , we define the distance of two image features  $f_i, \hat{f}_{i'}$  as

$$\mathcal{D}(f_i, \hat{f}_{i'}) = \frac{1}{HW} \sum_{k=1}^H \sum_{l=1}^W \sqrt{\sum_{j=1}^{N_c} (\hat{f}_{i'jkl} - f_{ijkl})^2}. \quad (2)$$

Using this distance, feature matching loss of  $f$  and  $\hat{f}$  is defined as

$$\mathcal{L}_{\text{fm}} = \mathcal{D}\left(\sum_{i=1}^{N_b} \frac{f_i}{N_b}, \sum_{i=1}^{N_b} \frac{\hat{f}_i}{N_b}\right), \quad (3)$$

and Chamfer distance is defined as

$$\mathcal{L}_{\text{cd}} = \frac{1}{N_n} \left( \sum_{i=1}^{N_b} \min_{i'} \mathcal{D}(f_i, \hat{f}_{i'}) + \sum_{i'=1}^{N_b} \min_i \mathcal{D}(\hat{f}_{i'}, f_i) \right). \quad (4)$$

Category	Width	Height	Depth
<i>car</i>	0.5	0.5	1.0
<i>horse</i>	1.0	1.0	1.0
<i>aeroplane</i>	1.0	0.5	1.0
<i>chair</i>	1.0	1.0	1.0

Table 3. Initial dimensions in base shape learning.

Category	Elevation	Azimuth
<i>car</i>	Uniform (0, 30)	Uniform(0, 360)
<i>horse</i>	Uniform (-10, 10)	Beta(1.5, 1.5) * 180
<i>aeroplane</i>	Uniform (-60, 60)	Beta(1.5, 1.5) * 180
<i>chair</i>	Uniform (0, 30)	Uniform(0, 360)

Table 4. Predefined distribution of viewpoints. When azimuth is zero, the camera is located in front of the object.

With a hyper parameter  $\lambda_{\text{rec}}$ , we define reconstruction loss as  $\mathcal{L}_{\text{rec}} = \lambda_{\text{rec}} \mathcal{L}_{\text{cd}} + (1 - \lambda_{\text{rec}}) \mathcal{L}_{\text{fm}}$ . We set  $\lambda_{\text{rec}}$  to 0.9, 0.3, 0.3, 0.9, and 0.8 for CIFAR-10 *car*, CIFAR-10 *horse*, PASCAL *aeroplane*, PASCAL *car*, and PASCAL *chair* respectively.

#### A.1.5 Optimization

In all experiments, we used Adam optimizer with  $\alpha = 0.0001$ ,  $\beta_1 = 0.5$ , and  $\beta_2 = 0.99$ . In base shape learning, batch size is set to 64, and the number of training iterations of the base shape learning is set to 1200, 300, 200, 800, and 200 for CIFAR-10 *car*, CIFAR-10 *horse*, PASCAL *aeroplane*, PASCAL *car*, and PASCAL *chair* respectively. In full model training, the number of iterations is set to 10000 in all categories.

## A.2 Additional experimental results

### A.2.1 Training data

Fig. 11 shows randomly selected training samples from our experiments. Different from commonly used datasets such as ShapeNet, foregrounds and backgrounds are not separated, viewpoints are unknown, objects have various shapes and sizes, and they locate and rotate freely. Though learning about 3D from these datasets is very challenging due to these characteristics, we think that trying this is a necessary step toward fundamental 3D understanding in machines.

### A.2.2 Comparison with HoloGAN on PASCAL

We showed comparison between ours and HoloGAN on CIFAR-10 in Fig. 8. For this comparison, we used codes provided by the authors. Specifically, we used default settings for CelebA dataset except for elevation and azimuth, which are set to the values for the cars included in the paper. In addition, we show comparison on PASCAL dataset in Fig. 12. Though images generated by HoloGAN are improved, they still lack consistency from multiple viewpoints. Especially, this method seems not good at generating front images of cars.

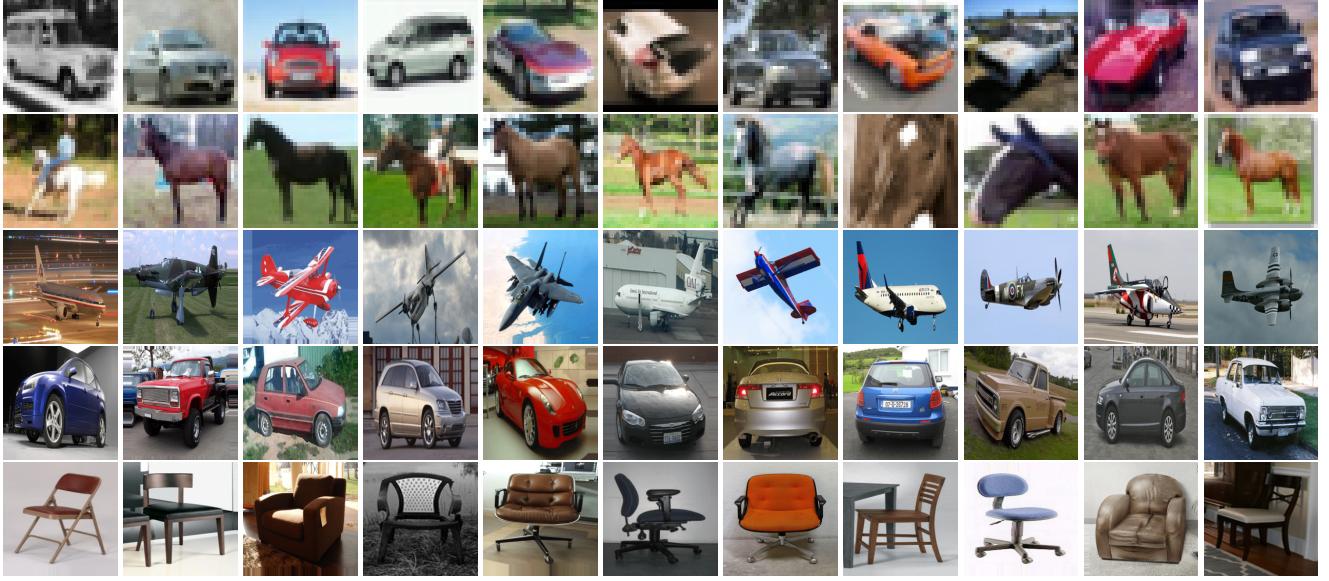


Figure 11. Randomly selected images in our training dataset. From top to bottom: CIFAR-10 *car*, CIFAR-10 *horse*, PASCAL *aeroplane*, PASCAL *car*, and PASCAL *chair*.



Figure 12. Comparison of ours with HoloGAN on PASCAL dataset. Images are rendered at 50 degree intervals.

### A.2.3 Randomly selected results

In Fig. 6 and Fig. 10, selected results on CIFAR-10 and PASCAL datasets are shown. For further qualitative evaluation, randomly selected results are shown in Fig. 13 to Fig. 22. Input images are shown in the top rows, reconstructed images using estimated shapes, poses, textures, and backgrounds are shown in the middle rows, and reconstructed images using a fixed viewpoint are shown in the bottom rows. For training set, the best shapes and viewpoints found during training are used, and for validation set, predicted shapes and viewpoints by the shape estimator and viewpoint estimator are used. Photometric fine-tuning are used for only validation set.

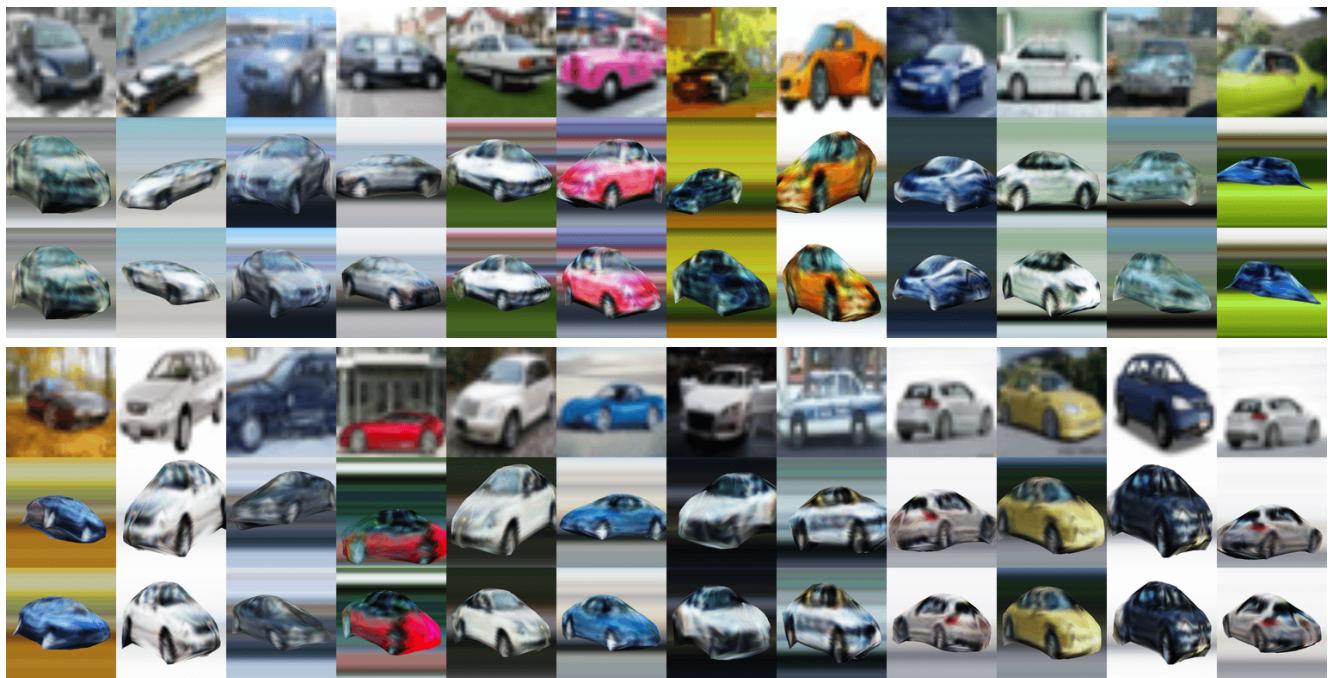


Figure 13. Randomly selected results on CIFAR-10 *car* training set.

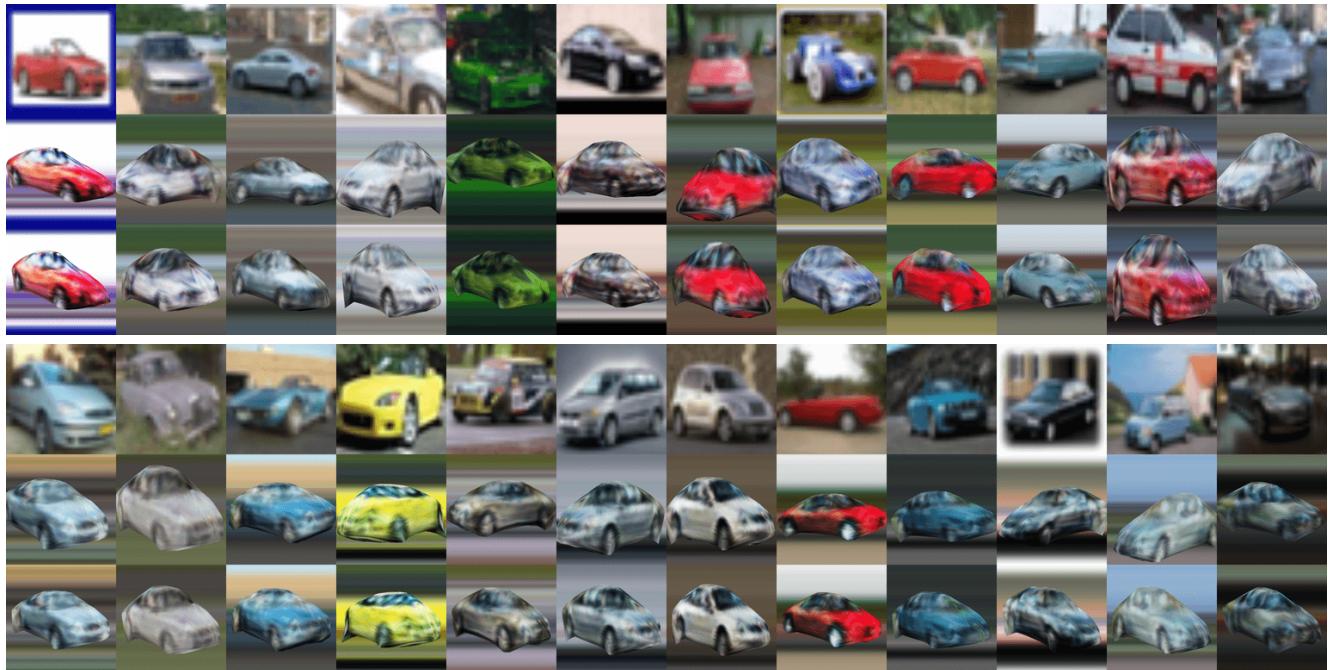


Figure 14. Randomly selected results on CIFAR-10 *car* validation set.

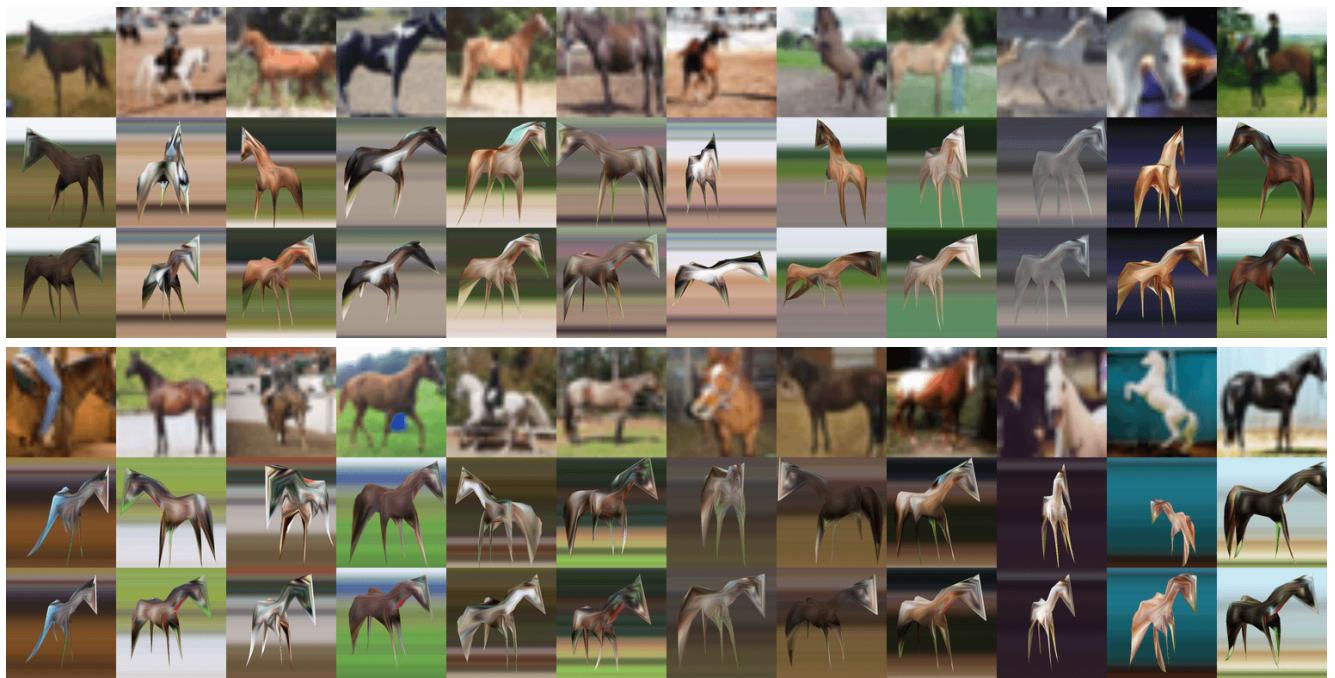


Figure 15. Randomly selected results on CIFAR-10 *horse* training set.

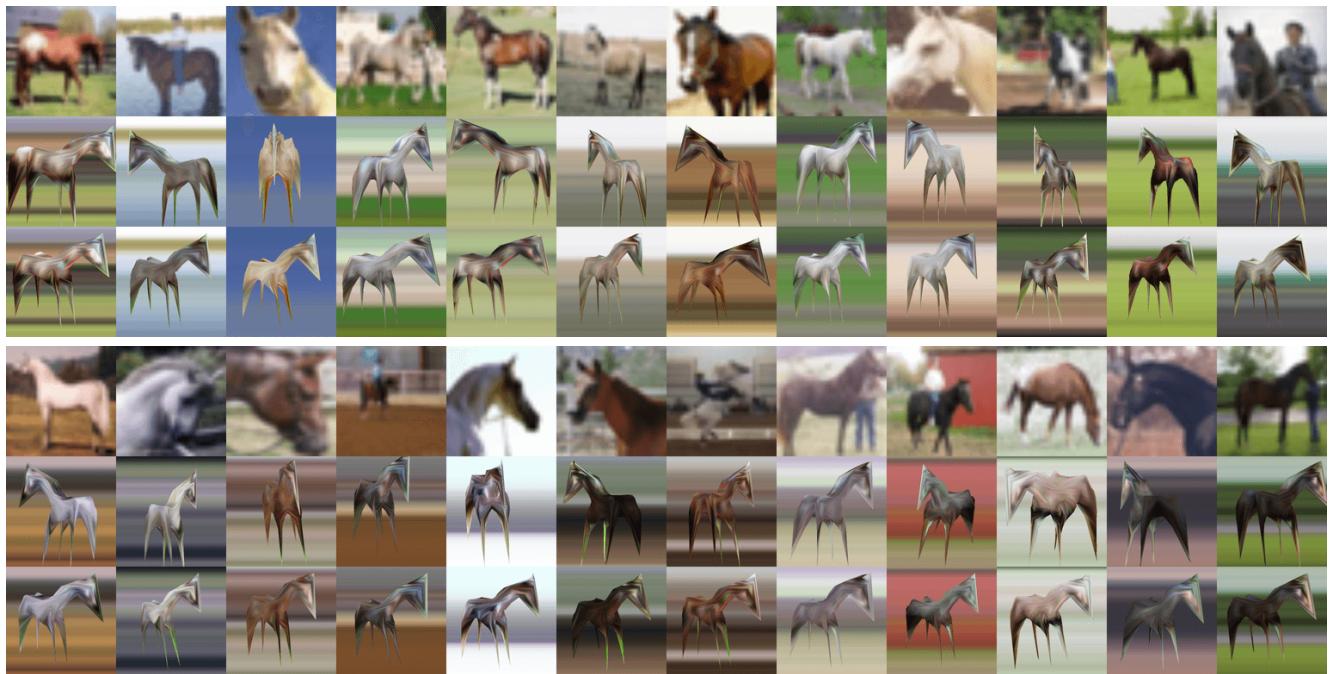


Figure 16. Randomly selected results on CIFAR-10 *horse* validation set.



Figure 17. Randomly selected results on PASCAL *aeroplane* training set.



Figure 18. Randomly selected results on PASCAL *aeroplane* validation set.

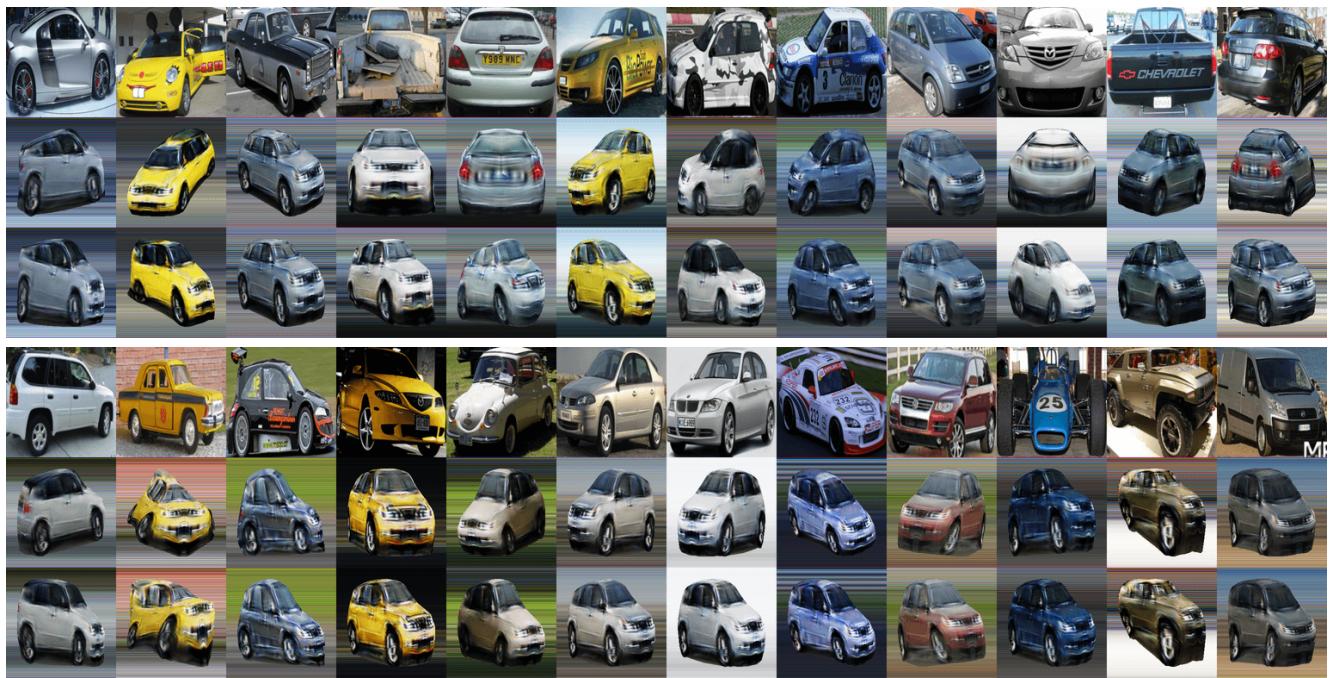


Figure 19. Randomly selected results on PASCAL *car* training set.

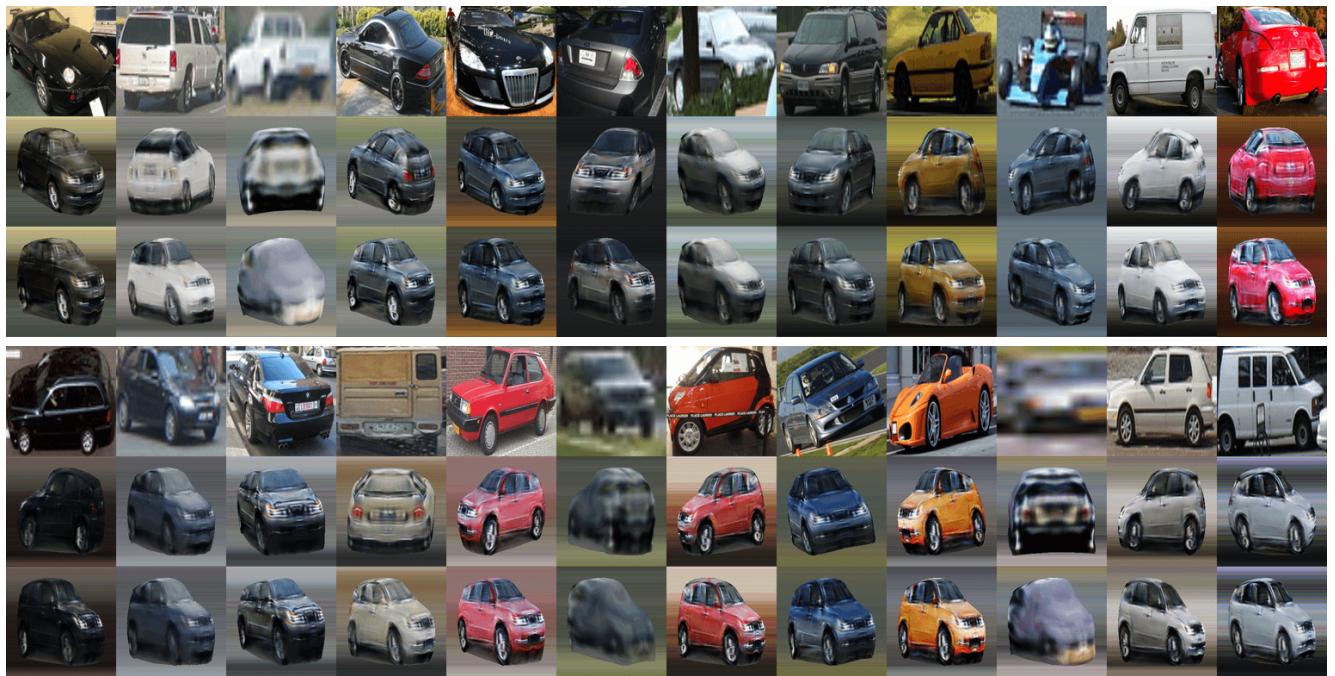


Figure 20. Randomly selected results on PASCAL *car* validation set.

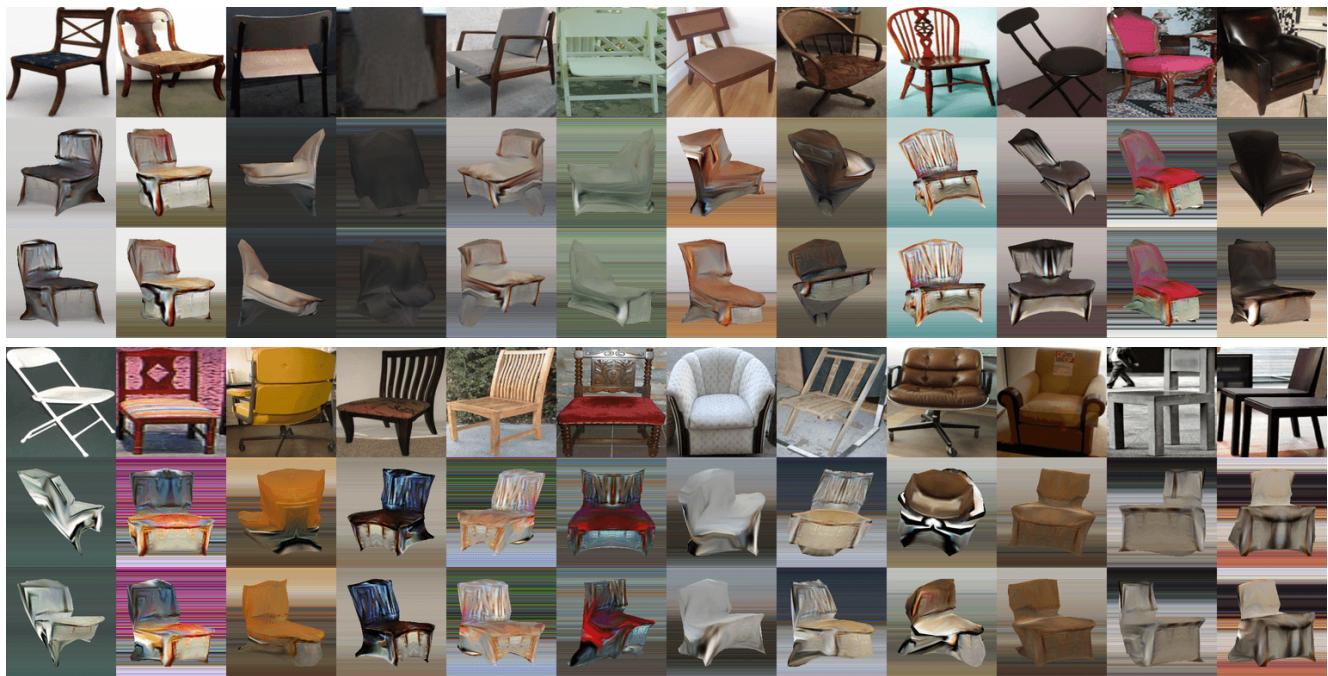


Figure 21. Randomly selected results on PASCAL *chair* training set.

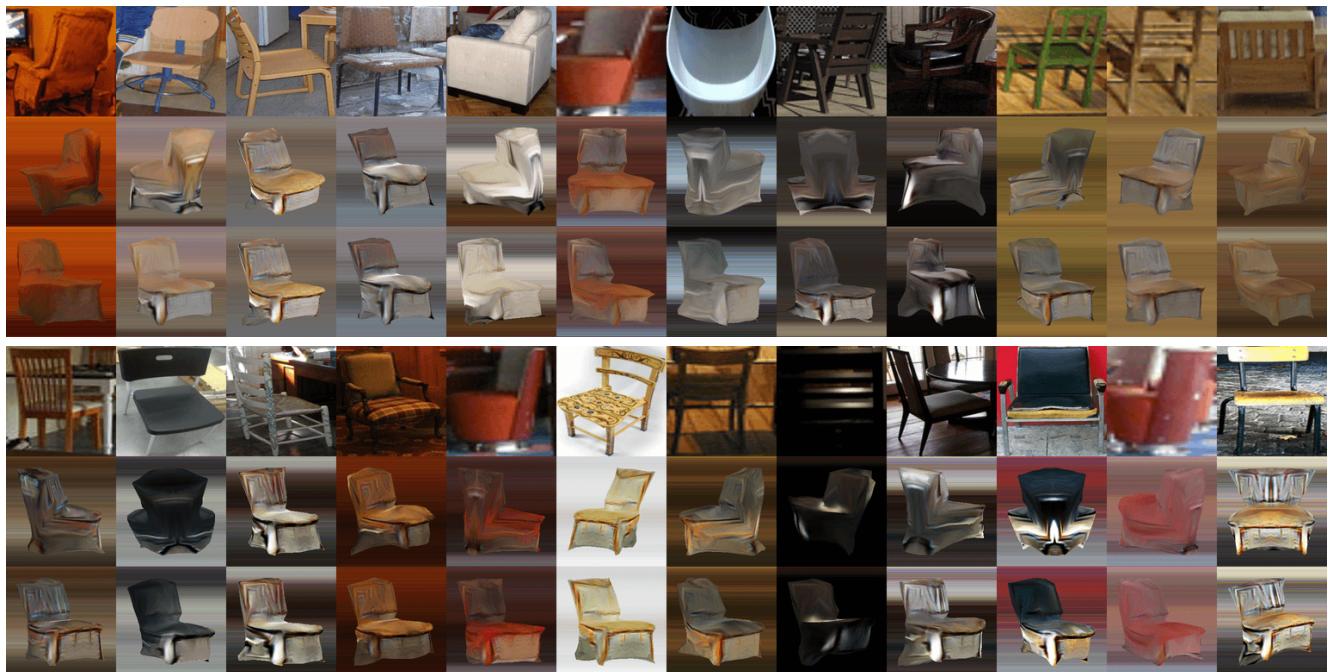


Figure 22. Randomly selected results on PASCAL *chair* validation set.