LEARNING POSE-AWARE 3D RECONSTRUCTION VIA 2D-3D SELF-CONSISTENCY

Yi-Lun Liao[*1], *Yao-Cheng Yang*[*1], *Yuan-Fang Lin*[1], *Pin-Jung Chen*[1],
Chia-Wen Kuo[2], *Wei-Chen Chiu*[3], *Yu-Chiang Frank Wang*[1]

[1]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[2]Department of Computer Science, Georgia Institute of Technology, USA
[3]Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
{b03901001, b03901161}@ntu.edu.tw

ABSTRACT

3D reconstruction, inferring 3D shape information from a single 2D image, has drawn attention from learning and vision communities. In this paper, we propose a framework for learning pose-aware 3D shape reconstruction. Our proposed model learns deep representation for recovering the 3D object, with the ability to extract camera pose information but without any direct supervision of ground truth camera pose. This is realized by exploitation of 2D-3D self-consistency between 2D masks and 3D voxels. Experiments qualitatively and quantitatively demonstrate the effectiveness and robustness of our model, which performs favorably against state-of-the-art methods.

Index Terms— deep learning, 3D shape reconstruction, camera pose estimation, perspective projection

1. INTRODUCTION

3D modeling and reconstruction can be applied to a variety of real-world applications, including visual rendering, modeling, and robotics. Over the past few years, convolution neural networks (CNN) have shown impressive progress in the areas of computer vision and image processing. For the task of 3D reconstruction, with the development of large-scale shape repository like ShapeNet [1], several deep learning methods have been proposed [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Many previous works predict pose-invariant 3D shapes in the volumetric form, voxels, aligned with a pre-defined canonical frame. That is, the predicted shapes would be similar for multiple images of the same instance but taken from different camera viewpoints. This can facilitate the efficiency of CNN.

Different settings such as the number of 2D input images and the viewpoint information of images are observed. 3D-R2N2 [3] uses a 3D-Convolutional Recurrent Neural Network to reconstruct a 3D volumetric model from multiple images of the same object. Other works [4, 5, 6, 7, 9] manage to learn shape reconstruction in weakly supervised settings. Guided by the projected 2D masks and given camera viewpoints, PTN [4] learns to reconstruct 3D voxels using 2D sihlouettes without supervision of ground truth (GT) shapes. MVC [9] utilizes the consistency between shape and viewpoint information independently predicted from multiple images of the same instance taken from different camera viewpoints. Although these weakly-supervised works can achieve satisfactory visual results, they either require GT viewpoint or achieve lower Intersection over Union (IoU) results.

As for representation disentanglement, many methods have been proposed [12, 13, 14, 15, 16]. For instance, Deep Disentangled Representations for Volumetric Reconstruction [15] takes 2D images as inputs and produces separate representations for 3D shapes and parameters of viewpoint and lighting. However, their learned representation of viewpoints cannot convert into exact azimuth or elevation angles.

In order to enable reconstructed shapes to interact with other shapes, for example placing them in 3D scenes with relative poses, the predicted shapes should contain pose information. Therefore, we propose a learning paradigm that enables pose-aware 3D shape reconstruction using ground truth voxels and masks but without ground truth pose. The proposed network predicts pose-invariant voxels and estimates corresponding camera pose. The predicted voxels are then transformed with estimated poses into pose-aware 3D shapes. Experiments show that the proposed method is able to achieve state-of-the-art 3D reconstruction performance and can produce satisfactory pose-aware 3D shape reconstruction.

2. PROPOSED METHOD

2.1. Notations and Architecture

The network architecture is composed of five components as shown in Fig. 1. We explain the functionality of each component and why the method achieves 3D shape reconstruction with the ability to extract camera pose information and enables pose-aware shape reconstruction.

(a) Image Encoder E: The encoder consists of residual structures [17]. It maps an input RGB image x of size $64 \times 64 \times 3$ to intrinsic shape representation, shape code z_s, and extrinsic viewpoint representation, pose code z_p. The former is a 512-dimensional vector, and the latter is a 16-dimensional vector. The pose code z_p is then converted into camera pose

* Indicates equal contribution.

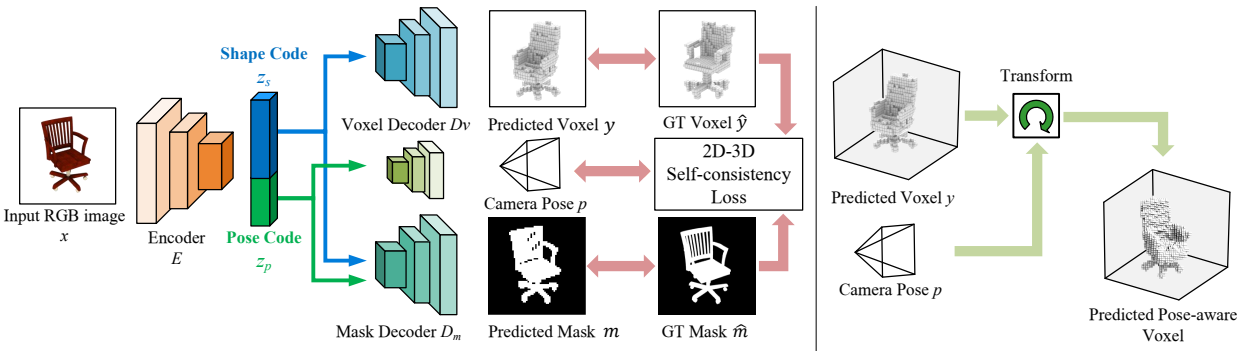

Fig. 1: The proposed method for learning pose-aware 3D reconstruction. The network learns pose estimation through 2D-3D self-consistency loss between masks and voxels. The predicted voxel is transformed with the estimated camera pose to produce the pose-aware voxel.

p through one FC layer. As translation does not affect the quality of shape reconstruction, we consider only elevation and azimuth for camera pose. In addition, we also assume camera intrinsic parameters are given as MVC [9].

(b) Voxel Decoder D_v: We use similar 2D deconvolution layers for voxel decoder D_v as [10]. It takes only shape code z_s as input and outputs voxels y of size $32 \times 32 \times 32$. We utilize GT voxels $\hat{y}$ to learn to predict pose-invariant shapes aligned with a certain frame, where, for instance, the Z axis represents the upward direction. This alignment can make the shape reconstruction using CNN more tractable and efficient.

(c) Mask Decoder D_m: We utilize U-Net [18] decoder structure for 2D mask segmentation. Both shape code z_s and pose code z_p are input latent vectors. The decoder outputs masks m of size 48×48 and has GT mask $\hat{m}$ supervision.

(d) 2D-3D Self-consistency Loss: The loss evaluates the inconsistency between GT 3D voxels $\hat{y}$ viewed from a predicted camera pose p and GT mask $\hat{m}$. That is, if the camera pose is incorrect, the projection of shapes would not be aligned properly with the associated GT 2D masks. This is how pose estimation can be learned without GT pose.

(e) Transform: The predicted voxels are transformed using estimated camera poses to produce pose-aware 3D voxels.

We detail (d) and (e) in the following subsection.

2.2. Consistency for Pose-aware Shape Reconstruction

We mention the ray consistency used in [9] and [19] with some modifications and describe the transformation of voxels using predicted pose into pose-aware voxels here.

Given camera intrinsic parameters (f_u, f_v, u_0, v_0), where (f_u, f_v) is focal length of camera and (u_0, v_0) is optical center of camera, the ray passing through the GT mask pixel (u, v) travels along the direction ($\frac{u-u_0}{f_u}$, $\frac{v-v_0}{f_v}$, 1). We consider discrete points along this ray path and sample points at a fixed set of depth values $\{\ d_i = \frac{i}{N},\ 1 \le i \le N\ \}$. The i-th point in the ray path is at $l_i = (\frac{u-u_0}{f_u}d_i,\ \frac{v-v_0}{f_v}d_i,\ d_i)$. Given camera rotation matrix R and camera translation t, the point would be mapped to the location $R\times(l_i + t)$. The camera rotation matrix R is parameterized by camera pose p defined in this paper, and the translation t is a fixed constant in this work. With GT voxel $\hat{y}$, trilinear sampling Tri is then used to determine the occupancy y_i^p of the i-th point on the ray passing through the pixel (u, v) as showed below.

$$y_i^p = Tri(\hat{y}, R \times (l_i + t)) \tag{1}$$

Therefore, the probability $q^p_{(u,v)}(i)$ that the ray passing through the pixel (u, v) stops at the i-th point would be determined.

$$q^p_{(u,v)}(i) = y_i^p \prod_{j=1}^{i-1}(1 - y_i^p) \quad \forall i \le N \tag{2}$$

The probability $q^p_{(u,v),pen}$ that the ray penetrates the voxel is shown here.

$$q^p_{(u,v),pen} = \prod_{j=1}^{i}(1 - y_i^p) \tag{3}$$

The loss of this ray is defined as below and is a function of GT voxel $\hat{y}$, GT mask $\hat{m}$, and predicted camera pose p.

$$\begin{aligned} L_{(u,v)}(\hat{y}, p, \hat{m}) = \hat{m}(u,v) q^p_{(u,v),pen} \\ + (1 - \hat{m}(u,v)) \sum_{i=1}^{N} q^p_{(u,v)}(i) \end{aligned} \tag{4}$$

Please note that $\hat{m}(u, v)$ represents the (u, v) pixel of $\hat{m}$ and is set to 1 if it is an object pixel and 0 otherwise. When $\hat{m}(u, v)= 1$, the ray should stop and therefore $q^p_{(u,v),pen}$ would be low. The ray consistency loss $L_{ray}(\hat{y}, p, \hat{m})$ is defined as the mean of $L_{(u,v)}(\hat{y}, p, \hat{m})$ over all pixels of $\hat{m}$. As Eqs. (1)-(4) are differentiable w.r.t. camera pose p, this loss can supervise pose estimation.

Besides, $(1 - q^p_{(u,v),pen})$ represents the (u, v) pixel of 3D-2D projection $P(\hat{y}, p)$. That is, if the ray terminates in voxels, $q^p_{(u,v),pen}$ would be low, and $(1 - q^p_{(u,v),pen})$ would be near 1 and represents an object pixel. In addition to ray consistency loss, we also adopt the Intersection-over-Union (IoU) loss [10] between 3D-2D projection $P(\hat{y}, p)$ and GT mask $\hat{m}$ defined as below.

$$L_{proj}(P(\hat{y}, p), \hat{m}) = exp(1 - \frac{\sum_i P_i \hat{m}_i}{\sum_i P_i + \hat{m}_i + P_i \hat{m}_i}) - 1 \tag{5}$$

We refer the combination of this loss and ray consistency loss to as 2D-3D self-consistency loss.

$$L_{sc} = (\frac{\lambda_{ray} L_{ray} + \lambda_{proj} L_{proj}}{2}) \tag{6}$$

In the proposed method, we use $N = 64$ instead of 80 as in [9]. As some fine structures like the bases of chairs would diminish when the GT mask is of size 32×32 as used in [9], we use GT masks of size 48×48.

The transformation of predicted shapes with estimated pose into pose-aware shapes is similar to ray consistency. Given a predicted voxel y and camera pose p, which parametrizes camera rotation matrx R, the occupancy ${y_i}'$ at any location of pose-aware voxel grid ${l_i}'$ is determined as below.

$$ {y_i}' = Tri(y, R \times ({l_i}')) \tag{7} $$

Please note that there is no translation in transformation.

2.3. Loss Functions

We describe the loss functions used to train the network here.

The fine structures such as the base of a chair or the wing of an airplane occupy a relatively small portion of the whole shape, and thus the model tends to predict nothing for these fine structures to minimize the penalty of traditional cross entropy loss. To solve this issue, we adopt different penalties for false positive and true negative of each voxel and utilize weighted cross entropy loss $L_{3drecon}$ defined as below where λ_p is the positive weight.

$$ L_{3drecon} = -\lambda_p \cdot \hat{y} \cdot log(y) - (1 - \hat{y}) \cdot log(1 - y) \tag{8} $$

We also consider IoU loss for 3D reconstruction.

$$ L_{IoU}(y, \hat{y}) = exp(1 - \frac{\sum_i y_i \hat{y_i}}{\sum_i y_i + \hat{y_i} + y_i \hat{y_i}}) - 1 \tag{9} $$

where y_i and $\hat{y_i}$ are the $i\text{-}th$ voxel of predicted shapes and GT shapes. Please note that y is the output of sigmoid function.

To regularize the distribution of shape code z_s and better model ambiguity of 3D reconstruction due to unseen parts of shapes, we adopt conditional variational autoencoder [12, 20] for E and D_v with the Kullback-Leibler (KL) divergence loss.

$$ L_{KL} = \mathcal{KL}(\mathcal{N}(z_{s,u}, z_{s,var}) \| \mathcal{N}(0, 1)) \tag{10} $$

where the original shape code z_s would consist of mean $z_{s,mu}$ and variance $z_{s,var}$, each is a 512-dimension vector. Only the mean $z_{s,mu}$ would be utilized by mask decoder D_m as mask segmentation is under-determined.

The 2D mask segmentation loss is calculated as:

$$ L_m = BCE(D_m(E(x)), \hat{m}) \tag{11} $$

where BCE indicates binary cross entropy loss.

As for camera pose p supervision, we use 2D-3D self-consistency loss L_{sc} as described previously. Please note that both L_{ray} and L_{proj} are differentiable w.r.t. camera pose p and thus can supervise pose estimation.

The loss functions for image encoder E, voxel decoder D_v and mask decoder D_m are shown as below:

$$ L_E = (\frac{L_m + (L_{3drecon} + L_{IoU})}{2}) + L_{sc} + \lambda_{kl} L_{KL} \tag{12} $$

Method	airplane	car	chair
3D-R2N2 [3]	0.513	0.798	0.466
DRC [19]	0.570	0.760	0.470
PTN [4]	0.584	0.738	0.507
Voxel Tube [10]	0.671	**0.821**	0.550
MVC(3D) [9]	0.570	0.790	0.490
Ours	**0.688**	0.807	**0.572**

Table 1: Comparison with other methods in terms of IoU. MVC (3D) indicates MVC trained with GT 3D voxels. MVC without 3D supervision produces lower IoU.

	airplane	car	chair
Rotation error	6.47°	89.74°	5.19°
Elevation error	2.79°	4.21°	1.73°
Mask IoU	0.930	0.987	0.970

Table 2: Quantitative result of pose estimation and mask segmentation. Mean values for each category are shown.

$$ L_{D_v} = L_{3drecon} + L_{IoU} \tag{13} $$

$$ L_{D_m} = L_m \tag{14} $$

We set $\lambda_p = 3$, $\lambda_{ray} = 10$, $\lambda_{proj} = 0.25$, and $\lambda_{KL} = 0.02$.

3. EXPERIMENTS

We evaluate the performance of our proposed method on pose-invariant voxel reconstruction, pose estimation, and pose-aware voxel reconstruction.

Dataset: The ShapeNet dataset [1] contains a rich collection of 3D CAD models. The three categories, airplane, car, and chair, are selected for our experiment. We use the data split from [4]. For each CAD model, we generate 24 rendered images and corresponding GT masks using the same camera pose information as in [4]. We use the same GT voxels in [9] to fit the projection module, and the grid size of voxels is $32 \times 32 \times 32$. We use Adam optimizer [21] with a learning rate of 2×10^{-4} for image encoder and decoders.

3.1. Pose-Invariant 3D Shape Reconstruction

To evaluate the performance, we consider the IoU between predicted 3D voxels and corresponding GT 3D voxels. We compare our approach with several state-of-the-art learning-based methods: 3D-R2N2 [3], Differentiable Ray Consistency (DRC) [19], Perspective Transformer Nets (PTN) [4], voxel tube [10] and Multi-view Consistency (MVC) [9]. All the works except [9] and [19] scale GT voxels to fit inside the whole $32 \times 32 \times 32$ grids. In order to utilize the projection module, [9], [19] and our work do not scale voxels, and the GT voxels are slightly smaller. This difference results in a little drop in IoU as each voxel occupies a larger portion of reconstructed shapes and thus one incorrect voxel would reduce IoU more. Even with this constraint, the performance of our method is still better than or equal to other methods. This shows that the separation of shape and pose representation can help 3D reconstruction.

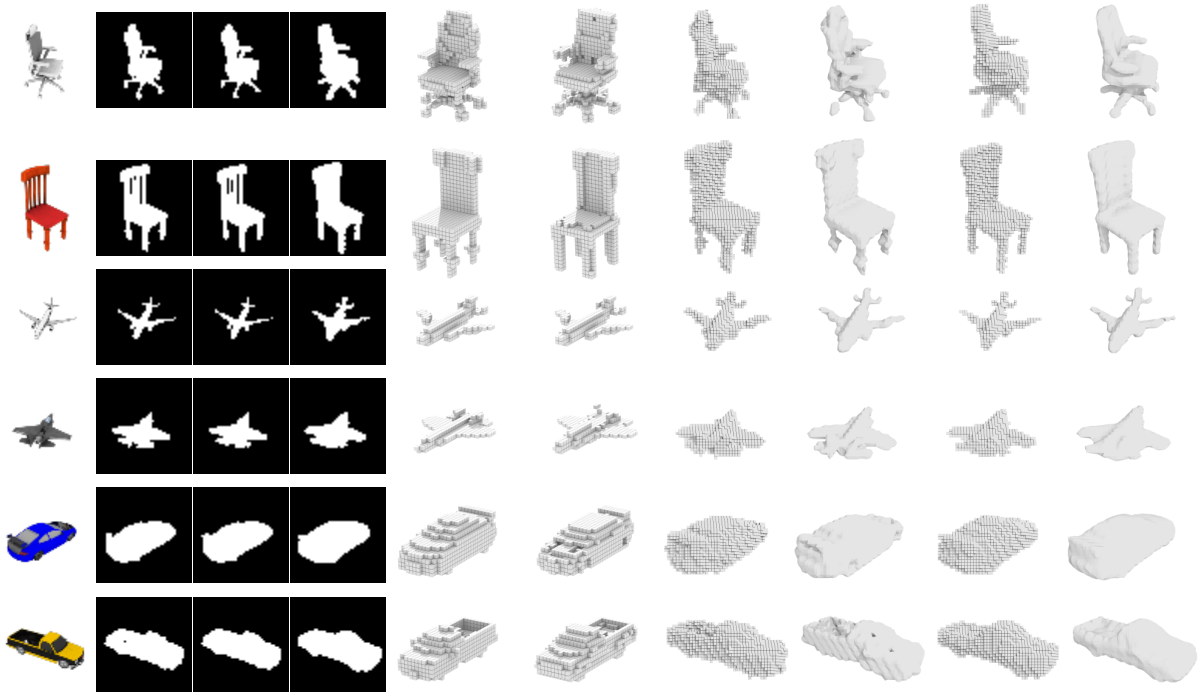

Fig. 2: Predictions on testing data with a single RGB input image. (a) Input image x. (b) GT mask $\hat{m}$. (c) Predicted mask m. (d) Projection of predicted shapes using estimated camera pose $P(y, p)$. (e) GT voxel $\hat{y}$. (f) Predicted voxel y. (g) GT pose-aware voxel. (h) Mesh from GT pose-aware voxel. The mesh is obtained by applying marching cubes to the pose-aware voxel and is shown here for better visual effects. (i) Predicted pose-aware voxel using predicted shape and pose. (j) Mesh from predicted pose-aware voxel.

	airplane	car	chair
pred shape & pred pose	0.569	0.691	0.525
pred shape & GT pose	0.640	0.792	0.550
GT shape & pred pose	0.728	0.723	0.806

Table 3: Mean IoU result of pose-aware 3D shape reconstruction on testing data. $pred$ means that shapes or poses are predicted.

	airplane	car	chair
pred shape & pred pose	0.709	0.864	0.774
pred shape & GT pose	0.715	0.864	0.773
GT shape & pred pose	0.683	0.827	0.777
GT shape & GT pose	0.684	0.817	0.777

Table 4: Mean IoU result of 3D-2D projection on testing data. We evaluate IoU between GT masks and different projections.

3.2. Pose Estimation and Pose-Aware 3D Reconstruction

The result of pose estimation and mask segmentation is shown in Table 2. The rotation error of car is close to 90° because 50.10% of pose predictions are rotated with 180° due to the symmetry of shapes viewed from some angles. Other pose errors are small as we do not utilize direct pose supervision.

We transform voxels of grid size $32 \times 32 \times 32$ into pose-aware voxels of size $48 \times 48 \times 48$. The performance of pose-aware 3D shape reconstruction from different voxels and poses evaluated by IoU is shown in Table 3. GT pose-aware voxels are obtained by transforming GT voxels with GT poses. This demonstrates the effectiveness of the proposed method to learn pose-aware shape reconstruction as we rely on only addition GT masks to learn pose information.

The performance of 3D-2D projections is shown in Table 4. We evaluate mean IoU between GT masks and projections. The result shows that the projections from predicted shapes with predicted poses can explain GT masks as well as projections of GT voxels with GT poses.

Visualization of predictions is shown in Fig. 2. We can see that our pose-aware 3D shape reconstruction contains fairly accurate camera pose information.

4. CONCLUSION

We propose a framework for pose-aware single-image 3D reconstruction. Our proposed model is able to learn deep representation from a single 2D input for recovering its 3D voxels, with the ability to extract camera pose information. More importantly, ground truth camera pose information is never observed during training of our proposed model. This is achieved by exploiting 2D-3D self-consistency between 2D masks and 3D voxels. Both quantitative and qualitative results demonstrate that our method is able to produce satisfactory results when compared to state-of-the-art approaches. Thus, the effectiveness and robustness of our model can be successively verified.

5. REFERENCES

[1] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015.

[2] Rohit Girdhar, David F. Fouhey, Mikel Rodriguez, and Abhinav Gupta, "Learning a predictable and generative vector representation for objects," in *European Conference Computer Vision (ECCV)*, 2016.

[3] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *European Conference Computer Vision (ECCV)*, 2016.

[4] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," in *Neural Information Processing Systems (NIPS)*, 2016.

[5] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese, "Weakly supervised 3d reconstruction with adversarial constraint," in *3D Vision (3DV), International Conference on 3D Vision*, 2017.

[6] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T Freeman, and Joshua B Tenenbaum, "MarrNet: 3D Shape Reconstruction via 2.5D Sketches," in *Neural Information Processing Systems (NIPS)*, 2017.

[7] Amir Arsalan Soltani, Haibin Huang, Jiajun Wu, Tejas D. Kulkarni, and Joshua B. Tenenbaum, "Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2511–2519.

[8] Abhishek Kar, Christian Häne, and Jitendra Malik, "Learning a multi-view stereo machine," in *Neural Information Processing Systems (NIPS)*, 2017.

[9] Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik, "Multi-view consistency as supervisory signal for learning shape and pose prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Stephan R. Richter and Stefan Roth, "Matryoshka networks: Predicting 3d geometry via nested shape layers," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Haoqiang Fan, Hao Su, and Leonidas J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2463–2471.

[12] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow, "Adversarial autoencoders," in *International Conference on Learning Representations (ICLR)*, 2016.

[13] Tejas D. Kulkarni, William F. Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum, "Deep convolutional inverse graphics network," in *Neural Information Processing Systems (NIPS)*, 2015.

[14] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Neural Information Processing Systems (NIPS)*, 2016.

[15] Edward Grant, Pushmeet Kohli, and Marcel van Gerven, "Deep disentangled representations for volumetric reconstruction," in *ECCV Workshops*, 2016.

[16] Chaoyue Wang, Chaohui Wang, Chang Xu, and Dacheng Tao, "Tag disentangled generative adversarial network for object image re-rendering," in *IJCAI*, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015, pp. 234–241, Springer International Publishing.

[19] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 209–217.

[20] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[21] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.