SICGAN - Single Image 3D Reconstruction based on Conditional GAN

Danish Syed*
University of Michigan
Ann Arbor
dasyed@umich.edu

Hansal Shah*
University of Michigan
Ann Arbor
shansal@umich.edu

Manas Jyoti Buragohain*
University of Michigan
Ann Arbor
manasjb@umich.edu

Abstract

3D shape reconstruction from single image has been a promising research area in recent times. However, most of the older approaches focus on generating a voxel based representation of objects which end up capturing only low resolution details due to data sparsity and computation cost of 3D convolution. More recently, there have been works on generating meshes of objects which are a more natural representation of 3D surfaces but these methods are based on handcrafted geometric losses. We propose an end-to-end Single Image Conditional GAN (SICGAN) framework for generating realistic meshes of 3D objects using a single RGB image. It consists of a Generator which is based on Pixel2Mesh and a Discriminator which employs graph based convolution for processing irregular mesh objects. Our SICGAN framework can be modified for different Generator or Discriminator architectures by swapping with corresponding modules to achieve high quality mesh reconstruction. We validate our mesh prediction on ShapeNet, where we were able to get a slight jump in reconstruction metrics for single image shape prediction and more realistic looking meshes. Code for our paper is publicly available at https://github.com/dysdsyd/SICGAN.

1. Introduction

The world around us is in 3D and thus working towards algorithms which help machines understand this inherent 3D structure in the world is an important area of research in Computer Vision. Inferring 3D shape from 2D images has always been an important research direction in this area. Early works [29, 22] as well as more recent works [16, 2, 41], explore representations of 3D shapes by inferring observable 2D properties. 3D shapes can be represented in many ways. [3, 6, 28, 43] use voxels and [9, 26, 21, 1, 37, 44] use point clouds to represent 3D structures. Voxel representation is conceptually simple but needs high spatial resolutions to capture fine structures and scaling to these high resolutions is nontrivial. Point Clouds can represent fine structures without huge number of points but they don't explicitly represent the surface of the shape. Also, extracting a mesh from them for rendering or other applications requires post-processing. Meshes on the other hand can explicitly represent 3D shapes and are standard representations used in graphics applications.

Our work focuses on producing a mesh representation of the 3D Object using a Conditional Generative Adversarial Networks (CGANs) [24] framework. There has been extensive research done in the past [20, 12, 15, 39, 10, 25, 33] to generate meshes from 2D representation of the objects. However, they all use some handcrafted loss functions for achieving realistic mesh reconstructions. For instance, [39] uses surface normal loss to favor smooth surface, an edge loss to encourage uniform distribution of mesh and a laplacian loss to prevent mesh faces from intersecting each other. Inspired by [14], it would be preferable to have a framework to which a high-level goal to make the output realistic could be specified and the system then would learns an appropriate loss function to satisfy this requirement. Generative Adversarial Networks (GANs) [11] achieve this through its loss function, a two player minmax game between Generator G and Discriminator D which encourages G to produce realistic outputs. Thus, we use a CGAN framework for generating meshes of 3D objects which will encourage the Generator to predict realistic meshes. Prior to us, GANs have been used to generate voxel representation [42, 36, 34, 19, 38] but voxel representation has its drawbacks as mentioned. For our experiments, we use the model in [39] as our Generator to create 3D meshes and a network consisting mainly of a Graph Convolution Networks (GCN) [7, 18, 4] as our Discriminator. Note here that our CGAN framework is general and apart from the models used by us, we can plug in any model which is used to produce a mesh in place of the Generator and similarly for the Discriminator variants of GCN layers can be used.

* indicates equal contribution

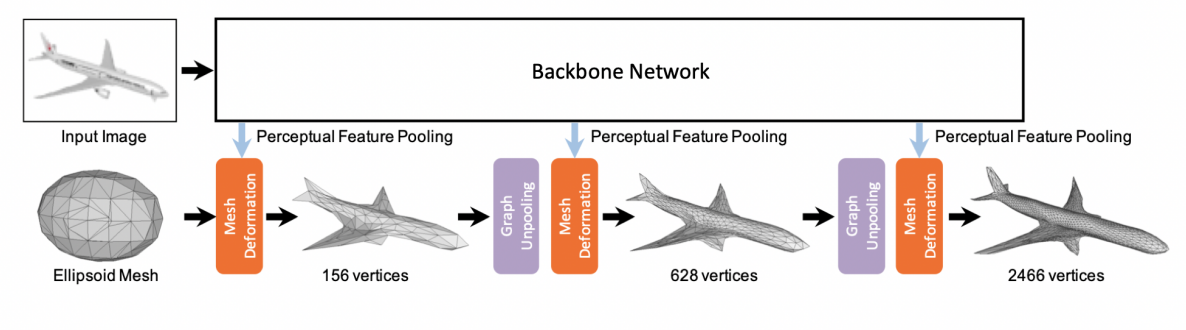

Figure 1. The Generator used (mainly based on [39]) in our CGAN framework

In summary, our main contribution is to propose a general CGAN framework which can be used to generate realistic 3D meshes. Additionally, we use the model in [39] as our Generator and a simple GCN as our Discriminator and show that this produces comparable and more realistic meshes.

2. Related Work

Our proposed approach is based on a mixture of two research areas: 3D Reconstruction and Conditional GANs. In this section we are going to discuss previous work in the aforementioned areas.

3D Reconstruction. Reconstructing 3D objects from color images has been around since the beginning of the field [30]. In recent years, learning based approaches have stood out as a major trend. With the advent of deep neural networks and large scale 3D shape collections, e.g. ShapeNet [5] and Pix3D [35], deep learning based 3D shape generation has made great progress.

To represent 3D structure various forms of representations like voxels [6, 40, 42] and point clouds [9, 21] have been explored for 3D reconstruction. However, deep voxel generators are constrained by it's resolution due to the data sparsity and computation cost of 3D convolution. Although, point clouds provides a flexible form of representation due to high memory efficiency and simple structure, they are not well suited for posing geometric constraints.

Mesh representations have been widely used in recent work [39, 10, 25, 33] on generation and reconstruction tasks. This was possible because of two reasons: 1) Mesh representation can model fine shape details and is compatible with various geometric regularizers 2) Graph Convolution Networks [18] provides an effective way to process them.

Our work closely relates to Pixel2Mesh [39], which deforms a generic pre-defined input mesh using Graph Convolutions to form 3D structures. We are building up on that by adding a discriminator D, thereby transforming the whole system into a Conditional GAN (CGAN) to achieve better reconstruction. Although we chose a simple model based on Pixel2Mesh[39], recent advancements over Pixel2Mesh such as [10, 25], can easily be used as the generator G since our approach can be generalised to other 3D Reconstruction state-of-the art methods.

Conditional GANs. The adversarial architecture was first proposed by *Goodfellow et al.* [11], and its main idea is to simultaneously train two models, the generator G and the discriminator D, and make them both stronger in adversarial learning. GANs under conditional settings have been extensively used for various tasks in image domain [14, 24, 27, 31, 45].

3D-GAN [42] applied GAN in learning latent 3D space, and it can generate 3D voxel models from the latent space by extending 2D convolution into 3D convolution. Building upon the work on 3D-GAN, *Edward et al.* [34] proposed conditional 3D-GAN for generating 3D object from images, similar work [19] also aimed at generating 3D objects from images or labels. However, aforementioned models are based on conditionally generating voxels, which are usually of low resolution due to the memory constraint on a modern GPU. Although, there are efforts in the direction to tackle this using octree representation [36] which allows reconstructing higher resolution outputs with a limited memory budget but they are not effective shape representation as per industrial standards. To avoid these drawbacks, we are focusing on processing only mesh polygons using GCNs in both the generator G and discriminator D of our 3D-CGAN pipeline.

3. Methods

3.1. Graph Convolution Network (GCN)

Since we are operating on meshes, we first provide some background on Graph-based Convolution. GCN [4, 7, 18] is a neural network that operates on graph. Given a Mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ where $\mathcal{V}$, $\mathcal{E}$ and $\mathcal{F}$ are the vertices, edges

and feature vectors attached to the vertices of the mesh respectively, a graph convolution for a layer l is defined as:

$$f_p^{l+1} = w_0 f_p^l + \sum_{q \in \mathcal{N}(p)} w_1 f_q^l \tag{1}$$

where $f_p^l \in \mathbb{R}^{d_l}, f_p^{l+1} \in \mathbb{R}^{d_{l+1}}$ are the feature vectors on vertex p before and after the convolution, and $\mathcal{N}(p)$ is the neighboring vertices of p; w_0 and w_1 are the learnable parameter matrices of $d_l \times d_{l+1}$ that are applied to all vertices. Note that w_1 is shared for all edges, and thus (1) works on nodes with different vertex degrees. Running convolutions updates the features, which is equivalent as applying a deformation.

3.2. Network Architecture

Our model is an end-to-end deep learning framework which takes an image as the input and yields a 3D mesh as the output. It consists of two major modules: a Generator (G) and a Discriminator (D).

3.2.1 Generator (G)

We reimplemented the Pixel2Mesh [39] network based on the implementation given by the Mesh R-CNN [10] code. Figure 1 illustrates the modified version of Pixel2Mesh which we used as our Generator. In this implementation, we used ResNet-50 [13] as our backbone instead of VGG-16 [32] architecture for pooling perceptual features. This implementation outperforms the original model due to a deeper backbone, better training recipe and minimizing Chamfer on sampled rather than vertex positions. Other than above mentioned modifications, rest of the implementation is same as the original paper. Please refer to the original paper [39] for a detailed understanding of the architecture.

3.2.2 Discriminator (D)

We implemented a shallow seven layer network which takes the generated object from G as input and classify whether the object is real (1) or fake (0). As depicted in the figure 2, it consisted of three graph convolution layers with features of size 16, 32 and 64 followed by max pooling and three linear layer which condenses to a single value and a sigmoid activation for real vs fake prediction. Our decision for a shallow network was motivated by the success of PatchGAN in [39] which models the image as Markov random field. Similarly, a shallow three layer graph convolution network would have a smaller receptive field of the mesh topology.

3.3. Objective

Our model is trained using a CGAN framework which generates objects through adversarial process estimation conditioned on some prior. CGANs learn a mapping from observed image x and random noise vector z, to y, $G : \{x, z\} \rightarrow y$. G and D can be regarded as two players of a min-max game where G is trying to "fool" D and are trained jointly. The objective of conditional GAN is stated as follows:

$$\begin{aligned} l_{cGAN}(G, D) =& \mathbb{E}_{x,y}[\log D(x, y)] + \\ & \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))] \end{aligned} \tag{2}$$

where G tries to minimize this objective against an adversarial D that tries to maximize it, i.e. $G^* = \arg\min_G \max_D l_{cGAN}(G, D)$

Similar to [39], we found that it is beneficial to mix geometric losses and regularizations with GAN objective. The Discriminator's job remains unchanged, but the generator not only have to fool the discriminator but also be geometrically similar to the ground truth mesh. To define geometric closeness, we are going to define losses that we can apply on mesh objects.

Mesh Losses: Similar to [10] we used differentiable mesh sampling to sample point clouds on surface of meshes. Given two pointclouds P, Q with normal vectors, let $\Lambda_{P,Q} = \{(p, \arg\min_q \|p - q\|) : p \in P\}$ be the set of pairs (p, q) such that q is the nearest neighbor of p in Q, and let u_p be the unit normal to point p. The chamfer distance between pointclouds P and Q is given by

$$\begin{aligned} l_{cham}(P, Q) = & |P|^{-1} \sum_{(p,q) \in \Lambda_{P,Q}} \|p - q\|^2 + \\ & |Q|^{-1} \sum_{(q,p) \in \Lambda_{Q,P}} \|q - p\|^2 \end{aligned} \tag{3}$$

and the (absolute) normal distance is given by

$$\begin{aligned} l_{norm}(P, Q) = & -|P|^{-1} \sum_{(p,q) \in \Lambda_{P,Q}} |u_p \cdot u_q| \\ & -|Q|^{-1} \sum_{(q,p) \in \Lambda_{Q,P}} |u_q \cdot u_p| \end{aligned} \tag{4}$$

The chamfer and normal distances penalize mismatched positions and normals between two pointclouds, but minimizing these distances alone results in degenerate meshes (intersecting faces). High-quality mesh predictions require additional shape regularizers: To this end we use an edge loss:

$$l_{edge}(V, E) = \frac{1}{|E|} \sum_{(v,v') \in E} \|v - v'\|^2 \tag{5}$$

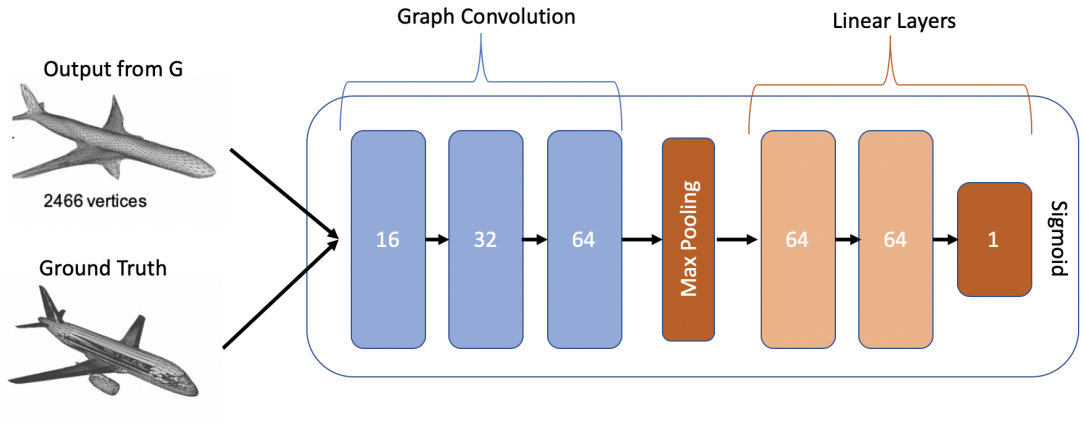

Figure 2. The Discriminator model used in our CGAN framework. Consists of a series of Graph Convolution layers followed my max-pooling and linear layers which output the probability of the 3D mesh generated by the generator being real.

where $E \subseteq V \times V$ are the edges of the predicted mesh. Alternatively, a Laplacian loss l_{lap} [8] also imposes smoothness constraints.

The total mesh loss is a weighted sum of all four losses, $l_{mesh} = l_{cham} + \lambda_1 l_{norm} + \lambda_2 l_{lap} + \lambda_3 l_{edge}$ where $\lambda_1 = 1.6e^{-4}, \lambda_2 = 0.3$ and $\lambda_3 = 0.1$ are the hyperparameters which balance the losses and fixed for all the experiments.

Final Objective: Our final objective is

$$G^* = \arg\min_G \max_D l_{cGAN}(G, D) + \lambda l_{mesh}(G) \quad (6)$$

As shown in [39, 23], generator simply learned to ignore the noise which was consistent with our experiments; the net could still learn a mapping from x to y, but would produce deterministic outputs, and therefore fail to match any distribution other than a delta function. This is still an open research problem, we are going to continue our work in exploring ways to induce stochasticity in the network.

4. Experiments

We benchmark our predicted meshes on the *Table* and *Chair* classes from ShapeNet [5] and compare our model with [39].

4.1. Dataset

We use the subset of the ShapeNet [5] dataset provided by [6] which contains a collection of 3D shapes, represented as textured CAD models organised into different categories following the WordNet hierarchy, and their rendered images. We render each mesh from up to 24 random viewpoints, giving RGB images of size 137×137. We filter out meshes with more than 6000 vertices due to computational constraints. For the same reason, we only use the models contained in the *Table* and *Chair* classes with a train/val/test split of size $3500/1000/1000$ for our experiments.

		Chamfer	$F1^{\tau}$	$F1^{2\tau}$
chair	Pixel2Mesh [39]	0.00062	55.98	70.27
	SICGAN (Ours)	**0.00059**	**56.99**	**71.35**
table	Pixel2Mesh [39]	0.00056	66.31	77.53
	SICGAN (Ours)	**0.00052**	**68.39**	**79.48**

Table 1. F-score (%) at different thresholds where $\tau = 10^{-4}$ and Chamfer Distance on test set from Shapenet *Table* and *Chair* classes using the evaluation protocol from Pixel2Mesh.[39].

4.2. Training and Optimization

To optimize our network, we follow the approach mentioned in [39]: we alternate between one gradient descent step on D, then one step on G. As suggested in the original GAN paper, rather than training G to minimize $\log(1 - D(x, G(x, z))$, we instead train to maximize $\log D(x, G(x, z))$[11]. In addition, we divide the objective by 2 while optimizing D, which slows down the rate at which D learns relative to G. We use minibatch SGD and apply the Adam solver [17], with a learning rate of 0.001, and momentum parameters $\beta_1 = 0.5, \beta_2 = 0.999$.

4.3. Evaluation

We compare our presented approach with Pixel2Mesh [39], which predicts meshes by deforming and subdividing an initial ellipsoid. We incorporate changes to made to the original Pixel2Mesh architecture by MeshRCNN [10] code for higher computational efficiency and robust feature extraction.

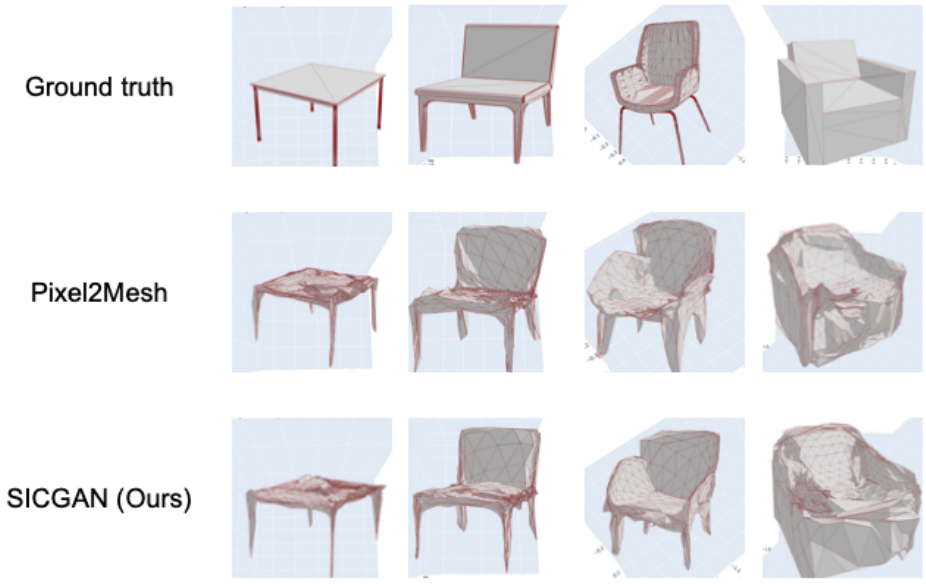

Figure 3. Comparison of results between Ground Truth, Pixel2Mesh baseline [39] and SICGAN (ours). We see that the meshes produced by our model look more realistic than the baseline.

We study the effect of using a conditional generative architecture on the single-image 3D reconstruction by implementing the following models:

- **Pixel2Mesh Baseline:** We implement Pixel2Mesh [39] and use the output mesh directly to minimize the Chamfer on the sampled points.
- **SICGAN Vanilla (ours):** We use Pixel2Mesh as the generator network for predicting meshes. In addition, we use a shallow GCN with FC layers as a discriminator to determine whether the predicted meshes are real or fake.
- **SICGAN with random noise (ours):** This is the same as SICGAN Vanilla, except the fact that the output of the generator is conditioned on additional random noise z.

We follow the evaluation metrics adopted by recent works [33, 39]. 10k points are uniformly sampled at random from the surface of the predicted and ground-truth meshes, and are used to compute Chamfer distance (equation 3). We also compute the F1-score using various distance threshold τ. The harmonic mean of the percentage of predicted points within τ of the ground-truth point gives the precision while the vice versa gives the recall. For F1-score, higher is better, while for Chamfer distance, lower is better.

From Table 1 we see that both our Vanilla SICGAN model performs better than the Pixel2Mesh [39] baseline with respect to both the F-score and the Chamfer distance metric. Consistent with the observations of [23] and [45] we observe no change in results when the output from the generator is conditioned on random noise z which corresponds to our SICGAN with random noise implementation.

We realize that the evaluation metrics for 3D shape generation may not thoroughly reflect the shape quality. The metrics often capture occupancy or point-wise distance rather than surface properties, such as continuity, smoothness, high-order details. Thus, we show some qualitative results for better understanding of these aspects in figure 3. We observe that the meshes produced by SICGAN seem more realistic than those generated using Pixel2Mesh [39].

5. Conclusion

In this paper we outline a new framework, SICGAN, which is successful in 3D generation from single image. Although we tested our model on a subset of ShapeNet [5], we were able combine CGAN objective with geometric losses and regularizers to achieve slightly better reconstruction. Also, the Generator and Discriminator modules can be replaced with different networks to transfer the CGAN framework for other combinations. We then demonstrate this system's generative power by recovering 3D objects from images, to achieve a slightly better performance on ShapeNet dataset [5].

Although, we were not able to add stochasticity in the network, we will continue to work on finding new ways to induce randomness in reconstruction such as VAE-CGAN, graph dropouts, random intial meshes or a combination of them.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017. 1

[2] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 5965–5974. IEEE, Jun 2016. 1

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. 1

[4] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34:18–42, 2017. 1, 2

[5] 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. *ArXiv*, abs/1512.03012, 2015. 2, 4, 5

[6] 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. *ArXiv*, abs/1604.00449, 2016. 1, 2, 4

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016. 1, 2

[8] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99*, 1999. 4

[9] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2017. 1, 2

[10] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9784–9794, 2019. 1, 2, 3, 4

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1, 2, 4

[12] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. 3

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv:1611.07004 [cs]*, Nov. 2018. arXiv: 1611.07004. 1, 2

[15] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 1

[16] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1966–1974. IEEE, Jun 2015. 1

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 4

[18] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2016. 1, 2

[19] Haisheng Li, Yanping Zheng, Xiaoqun Wu, and Qiang Cai. 3d model generation and reconstruction using conditional generative adversarial network. 12(2):697. 1, 2

[20] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018. 1

[21] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI*, 2018. 1, 2

[22] David Marr and Herbert Keith Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200(1140):269–294, 1978. 1

[23] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2016. 4, 5

[24] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]*, Nov. 2014. arXiv: 1411.1784. 1, 2

[25] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9963–9972, 2019. 1, 2

[26] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1

[27] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*, Jan. 2016. arXiv: 1511.06434. 2

[28] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances in neural information processing systems*, pages 4996–5004, 2016. 1

[29] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 1

[30] Lawrence G. Roberts. Machine perception of three-dimensional solids. In *Outstanding Dissertations in the Computer Sciences*, 1963. 2

[31] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from Simulated and Unsupervised Images through Adversarial Training. *arXiv:1612.07828 [cs]*, July 2017. arXiv: 1612.07828. 2

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 3

[33] Edward Smith, Scott Fujimoto, Adriana Romero, and David Meger. Geometrics: Exploiting geometric structure for graph-encoded objects. In *ICML*, 2019. 1, 2, 5

[34] Edward Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. 1, 2

[35] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018. 2

[36] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. 1, 2

[37] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 1

[38] Qun Wan, Yidong Li, Haidong Cui, and Zheng Feng. 3d-mask-GAN:unsupervised single-view 3d object reconstruction. In *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, pages 1–6. IEEE. 1

[39] 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. 1, 2, 3, 4, 5

[40] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Joshua B. Tenenbaum. Marrnet: 3d shape reconstruction via 2.5d sketches. In *NIPS*, 2017. 2

[41] Jiajun Wu, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. Tenenbaum, Antonio Torralba, and William T. Freeman. 3d interpreter networks for viewer-centered wireframe modeling. *International Journal of Computer Vision*, 126(9):1009–1026, Sep 2018. 1

[42] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, 2016. 1, 2

[43] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2690–2698, 2019. 1

[44] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4541–4550, 2019. 1

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv:1703.10593 [cs]*, Nov. 2018. arXiv: 1703.10593. 2, 5