MuDeGAN: Shape synthesis with Multiview Depthmaps and Reprojection Consistency

Kamal Gupta
University of Maryland, College Park
kampta@cs.umd.edu

Susmija Reddy
University of Maryland, College Park
jsreddy@cs.umd.edu

Matthias Zwicker
University of Maryland, College Park
zwicker@cs.umd.edu

Abstract

We present MuDeGAN (Multiview Depthmap GAN), a general purpose framework for generating 3D shapes by leveraging recent advances in 2D image generation using Generative Adversarial Networks (GANs). Volumetric representation such as Voxels to represent 3D shapes is not scalable. We instead propose to synthesize shapes in 2D space using a multiview depthmap representation. In order to bind the representations generated by a GAN conditioned on different viewpoints, we introduce a novel re-projection consistency loss. Our experimental results demonstrate the two-fold advantage of our approach. First, we can directly borrow architectures that work well in 2D image domain to 3D. Second, we can generate high resolution 3D shapes with low memory effectively.

1 Introduction

Generative Adversarial Networks (Goodfellow et al. [2014]) have shown tremendous growth in recent years (Brock et al. [2018], Karras et al. [2017, 2018]) which can be attributed at least partly to improved GAN training architectures along with large scale, high resolution image datasets. However, thanks to the curse of dimensionality, we are still far from generating novel and diverse 3D shapes at scale. Two fundamental problems that are preventing us from bridging the gap between 2D and 3D shape synthesis are - first, finding an appropriate 3D representation and second, availability of a large scale 3D dataset. In this paper, we look at first of the two problems and postulate multiview depthmaps as a viable 3D representation for generating high quality 3D shapes.

Before the advent of deep learning, data driven methods of synthesizing 3D shapes involved using class-specific individual component level segmentation of models to build a probabilistic model of components' appearance and relative placements. In last few years, attempts have been made to use voxel grids and meshes within a GAN framework for the generations task. Each of these 3D representations poses restrictions on the kind of neural network architectures that can be deployed. Voxel grids for example are memory inefficient and hard to scale beyond 128^3 sizes with existing hardware. Meshes or point clouds on the other hand can be used with graph convolutional networks, they, however are not yet as effective as convolutional networks for euclidean data types such as images. Meshes additionally can represent only fixed topology and while work well for shapes such as faces, can't generalize to simple objects with varying topology such as chairs or tables. Multiview depthmaps offer a promising alternative. They are memory efficient and can be directly used to store and visualize 3D objects and scenes without the need of projecting to 3D space.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

In this work, we show that all image based GAN architectures can be directly used to generate depthmaps with little or no modifications. We introduce a simple reprojection consistency loss that ensures depthmaps generated with same input latent noise but different viewpoints correspond to same 3D objects. This loss function is generic, takes very little memory and compute overhead and can be used in conjunction with various generative models.

2 Related Work

Synthesizing 3D shapes from complex domains such as objects, humans, clothes etc. is a problem of great interest because of its immense potential applications in entertainment and fashion industry. Traditional approaches to shape synthesis has focused on identifying new plausible combinations of components from existing shapes. Chaudhuri et al. [2011] developed a probabilistic representation of shape structure that can be used to suggest relevant components during an interactive assembly-based modeling session. Kalogerakis et al. [2012] proposed a generative model based on component-based structure. The model learns the probabilistic relationships between properties of shape components, and relates them to learned underlying causes of structural variability(latent variables) within the domain. Huang et al. [2015] uses part-based templates to construct a probabilistic deformation model for generating shapes.

Since the advent of deep learning, research community has moved to the use of Variational Autoencoders (VAEs) [Kingma and Welling [2013]] and Generative Adversarial Networks (GANs) [Goodfellow et al. [2014]] as the de-facto approaches to data-driven generative modeling. These approaches have worked well especially for sampling realistic data in 2D domains such as images. Adopting these approaches for 3D however has remained a challenge. First GAN based work to synthesize 3D shapes was perhaps Wu et al. [2016]'s 3DGAN, which encodes 3D shapes using a voxel-based representation. Their work is a simple extension of DCGAN [Radford et al. [2015]] to voxel volumes with use of 3D convolutions. However, use of 3D convolutions is very memory intensive and hard to scale to higher resolution voxel grids.

View based representations have demonstrated potential for 3D shape understanding and reconstruction. Several works have explored reconstructing 3D shapes from single or multi-view images. One of the early works from Dosovitskiy et al. [2017] attempted to generated depthmaps or images given properties of 3D shapes, albeit in a deterministic manner. Soltani et al. [2017] proposed a VAE based approach to model 3D shapes using multi-view depth maps and silhouettes. While their method works well to perform reconstruction of depthmaps images, VAEs have a tendency to produce blurry outputs and haven't shown as much promise as GANs.

Some of the recent work from Cheng et al. [2019], Bouritsas et al. [2019], Ranjan et al. [2018] have exploited mesh representation of 3D shapes along with Graph Convolution Networks [Henaff et al. [2015]] for modelling human faces and body. Mesh convolutions, as mentioned previously, have a drawback that they can generate shapes of a fixed topology only as of now.

Idea of reprojection consistency has been used almost ubiquitously in 3D reconstruction using multiview supervision from images, depthmaps or surface normals [Wu et al. [2017], Ollinger [1990], Tulsiani et al. [2017], Vogiatzis et al. [2007], Seitz et al. [2006]]. Couple of challenges before us is absence of supervision since only we don't have any groundtruth targets while training a GAN model. Only feedback that model receives is ease of classifying the depthmap as real vs fake. Another challenge before us is to be able to backpropagate the reprojection consistency feedback to improve the generator. In Section 3, we discuss our approach on how we handle above challenges. In Section 4, we evaluate our framework both quantitatively and qualitatively. Finally we summarize and discuss future work in Section 5.

3 Our Approach

In this section, we will discuss our framework for 3D shape generation. Our objective is to be able to generate high resolution shapes of various objects using just the depthmaps. While GANs are effective at generating sharp images that look realistic, asking a GAN to generate depthmaps from different viewpoints is not good enough. This is because in existing form, GAN framework is oblivious to the 3D structure described by the depthmap. In order to deal with this shortcoming, we propose to use a novel reprojection consistency loss term as discussed in Section 3.2.

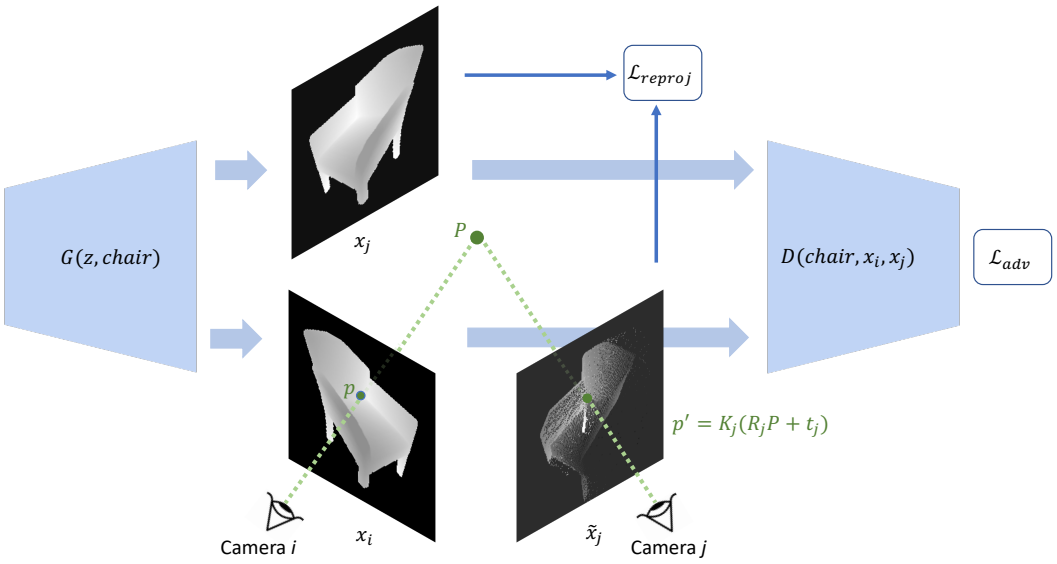

Figure 1: **MuDeGAN architecture**: Our conditional generator G takes a latent noise variable z and one-hot coded class category as input to generate V different depthmaps of a same object. Discriminator then takes all V depthmaps as input along with class information to classify between real and fake shapes. In order to compute reprojection consistency loss term, we reproject depthmap from one view to another nearby view. $\mathcal{L}_{\text{reproj}}$ is then simply a smooth L_1 distance between depth values of reprojected depthmap and generated depthmap. We resolve the depth ambiguity which arise during the reprojection depth by choosing the minimum depth value for a pixel location.

3.1 Network Structure

In our GAN framework, we use a conditional generator G which generates depthmaps of a 3D shape $\mathcal{S}$ given a noise vector z and shape category. Let $\{\mathbf{x}_i\}, i \in [1, V]$ denote the set of multi-view depthmaps rendered from a known number of V different camera view-points generated by G. Discriminator simply uses these depthmaps along with the shape category and tries to discriminate between set of real depthmaps vs set of fake depthmaps. We use $V = 20$ viewpoints and explore 2 different GAN backbones in our experiments - DCGAN by Radford et al. [2015] and ProGAN by Karras et al. [2017]. While this framework is sufficient to generate realistic looking depthmaps, actual depth values don't need to be consistent across all the views to ensure consistency in 3D. To this end, we generate pseduo-groundtruth depthmaps for each viewpoint to compute the proposed reprojection consistency loss function as described below.

3.2 Reprojection Consistency Loss

Given a depthmap $\mathbf{x}_i$, let $\mathbf{p}_i = [\mathbf{u}\ \mathbf{v}\ \mathbf{d}]$ denote all foreground pixels where $\mathbf{d}$ is a one dimensional vector containing depths of points at pixel locations $\mathbf{u}, \mathbf{v}$ in image. In order to get the foreground pixels, we simply threshold the depth values (instead of generating a separate silhouette mask as done by Soltani et al. [2017]). All foreground pixels $\mathbf{p}_i$ can be projected to corresponding 3D points in world coordinate system $\mathbf{P}_i$ using inverse perspective transformation, i.e.,

$$\mathbf{P}_i = \mathrm{R}_i^{-1}(\mathrm{K}_i^{-1}\mathbf{p}_i - \mathrm{t}_i)$$

where K_i, R_i, and t_i are camera intrinsic matrix, rotation matrix and translation for the i^{th} view-point. These points can be reprojected to another view-point j using a simple perspective transformation as follows.

$$\begin{aligned}\mathbf{p}_{ji} &= \mathrm{K}_j(\mathrm{R}_j\mathbf{P}_i + \mathrm{t}_j) \\ &= \mathrm{K}_j(\mathrm{R}_j(\mathrm{R}_i^{-1}(\mathrm{K}_i^{-1}\mathbf{p}_i - \mathrm{t}_i)) + \mathrm{t}_j)\end{aligned}$$

where $\mathbf{p}_{ji}$ represents 2D representation of 3D points $\mathbf{P}_i$ in j^{th} view-point's frame. These points will correspond to the reprojected depthmap $\mathbf{x}_{ji}$. Note that during this operation, multiple depth values can be assigned to same pixel location in $\mathbf{x}_{ji}$. We resolve the ambiguity by taking minimum of these depth values (since the points closer to the camera will get rendered preferentially). We treat $\mathbf{x}_{ji} = \tilde{\mathbf{x}_j}$ as pseudo-groundtruth for j^{th} depthmap $\mathbf{x}_j$. For all the (foreground) pixel locations in this pseudo-groundtruth, we compute our Reprojection Consistency Loss $\mathcal{L}_{\text{reproj}}$ by simply computing a soft $\mathcal{L}_1$ distance (or Huber Loss) between $\tilde{\mathbf{x}_j}$ and $\mathbf{x}_j$.

To generate pseudo-groundtruth for each view point j, we consider 3 neighboring viewpoints of j. We denote all the foreground pixel locations of $\tilde{\mathbf{x}_j}$ with $z^l_{ji}, l \in [1, L]$. Our final loss function now becomes

$$\mathcal{L}_{\text{reproj}} = \frac{1}{3 \times V} \sum_{j=1}^{V} \sum_{i=1}^{3} f(\mathbf{x}_j, \mathbf{x}_{ji})$$

$$f(\mathbf{x}_j, \mathbf{x}_{ji}) = \sum_{L} z^l_{ji}$$

$$z_{ji} = \begin{cases} 0.5(d_{ji} - d_j)^2, & \text{if } |d_{ji} - d_j| < 1 \\ |d_{ji} - d_j| - 0.5, & \text{otherwise} \end{cases}$$

where $\tilde{d_j}$ and d_j are depth values in $\mathbf{x}_{ji}$ and $\mathbf{x}_j$ respectively.

3.3 Training details

Our reprojection formulation is agnostic to the choice of GAN architecture. In our experiments, we try two of the popular architectures - DCGAN and ProGAN. DCGAN (Radford et al. [2015]) is a straight-forward application of GAN framework. It mainly composes of convolution layers without max pooling or fully connected layers. It uses convolutional stride and transposed convolution for the downsampling and the upsampling. We use a 5-layer deep DCGAN and 100 dimensional noise vector z. Generator and Discriminator architectures are mirror of each other.

ProGAN has a similar architecture as GAN, however differs in training methodology. Starting with lower resolution 7×7 depthmaps, we progressively add a layer to both generator and discriminator and also increase the resolution of generated depthmaps. As discussed by the authors, we use Wassertein Loss (Arjovsky et al. [2017]) and Gradient Penalty (Gulrajani et al. [2017]) to improve the GAN training. Finally, we use auxillary class-conditioning as described by Odena et al. [2017] for conditional synthesis.

4 Experiments

4.1 Dataset

Figure 2: **Rendered Depth Maps:** Depth Maps are rendered from voxel grid representations of 3D models by placing 20 virtual cameras at 20 vertices of a regular dodecahedron. The areas closer to the camera are considered to be bright and the areas farther away appear dark in the depth maps

We train our MuDeGAN on ShapeNet CoreV2 dataset [Chang et al. [2015], Wu et al. [2015]] which consists of almost 50000 3D aligned models from 50 categories. We considered 5 major categories

Figure 3: Inception Score and Frećhet Distance Scores on the generated depth maps

Model-generated-data	IS (higher the better)	FID (lower the better)
ShapeNet Models (reference)	3.107	-
3DGAN [Wu et al. [2016]]	1.627	25.094
SDS [Soltani et al. [2017]]	-	-
Ours DCGAN	3.029	0.389
Ours DCGAN + $\mathcal{L}_{\text{reproj}}$[1]	2.926	1.449
Ours ProGAN [2]	2.964	1.445
Ours ProGAN + $\mathcal{L}_{\text{reproj}}$	-	-

for all our experiments - airplane, car, chair, sofa, table of the Shapenet dataset with a total of 25938 models.

To train our model, we render the depth maps for each 3D model in our dataset as follows. Given the 3D voxel occupancy grid representation of the 3D object, we consider 20 different view points to generate the depth maps, following the approach in Zhu et al. [2018]. The renderings are generated by placing 20 virtual cameras at 20 vertices of a regular dodecahedron enclosing the shape [Soltani et al. [2017]]. Figure 2 shows the depth maps rendered from an airplane model.

To render the depth maps from the voxel grid representation from a given view point, we first generate a collection of rays, each originating from the camera's center and going through a pixel's center in the image plane. To render the depth maps, we need to calculate whether a given ray would hit the voxels, and if so, calculate the corresponding depth value for that ray. To this end, we first sample a collection of points at evenly spaced depth along each ray. Next, for each point, we calculate the probability of hitting the input voxels using a trilinear interpolation (Jaderberg et al. [2015]). We then calculate the expectation of visibility and depth along each ray.

All cameras are assumed to be fixed at a distance of 2.5m from the center of the voxel grid and point towards the center of the grid. The focal length of the camera used is 50mm. The near clipping plane is fixed at 1.5m from the camera and the far clipping plane at 3.5m. For all our experiments, we use rendered images of resolution 224×224. For training purposes, we set our background pixels to -1 to differentiate background from the object.

4.2 Quantitative Evaluation

Evaluating the quality and diversity of generated 3D shapes is a challenging task. We adopt the common evaluation metrics used in image domain i.e. Inception Score (IS) [Salimans et al. [2016]] and Fréchet Inception Distance (FID) [Heusel et al. [2017]] for our purpose. Off-the-shelf Inception network is trained on 2D RGB color images, so we can't use them directly. We trained our Inception network on the rendered depthmaps from 5 categories of the ShapeNet dataset. We use a 80:20 split for training and testing and obtain 98.8% accuracy on the dataset (using 20 depthmaps for each object). We then generated 500 shapes of each class from our model. We report results on following 4 variations of our model (1) DCGAN (2) DCGAN + $\mathcal{L}_{\text{reproj}}$ (3) ProGAN (4) ProGAN + $\mathcal{L}_{\text{reproj}}$.

We also compare our results with two state of the art approaches for 3D shape generation Wu et al. [2016] and Soltani et al. [2017]. We use the pre-trained models provided by the authors to generate 500 shapes of each category for fair comparison.

Table 3 shows the relative performance of each of the models on the 2 metrics. We note that our method performs comparable to state of the art and is close to real world data. We do however want to point out that these metrics are not perfect and miss out on a lot of important qualities of 3D shapes. For example, if a generated chair is missing a leg, it can still easily be classified as chair without being a good quality generation. Similarly in most cases shape of the depthmap alone might be an indicator for classification, while actual depth values need not be perfect for a model to report good numbers on above metrics. Hence, we next see some qualitative results from our model.

[1] Intermediate results for model trained with $\lambda_{reproj} = 0.2$

[2] Intermediate results from model trained till 112×112 resolution

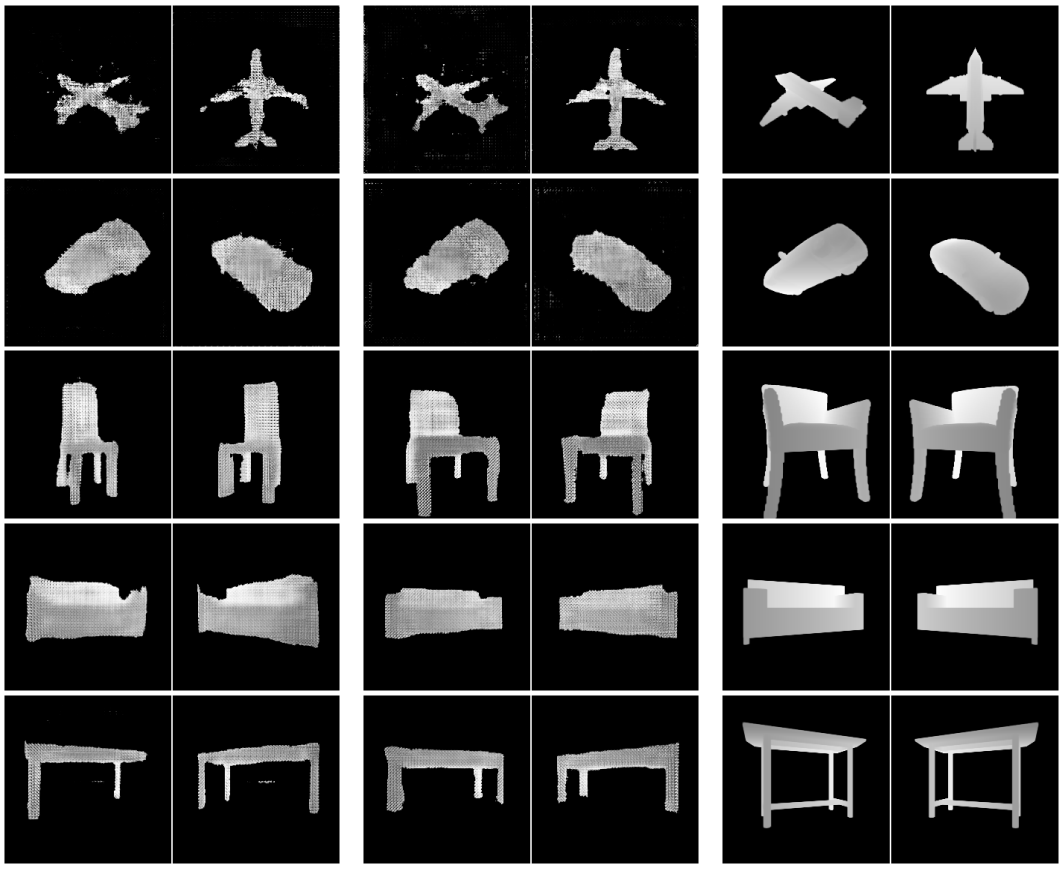

Figure 4: **Generates Samples.** One generated sample from each of the category: airplane, car, chair, sofa and table respectively. First two columns correspond to 2 different depth views of sample generated from DCGAN, next two columns correspond to 2 different depth views of sample generated from DCGAN + $\mathcal{L}_{\text{reproj}}$, and the last two columns correspond to 2 different depth views of sample from ShapeNet dataset.

4.3 Qualitative Evaluation

Figure 4 shows a sample shape generated from MuDeGAN with DCGAN backend (ProGAN experiments are still running) from five different categories on which the model was trained (airplane, car, chair, sofa and table). We can see that a single conditional GAN model is able to synthesize shapes from 5 different categories with a simple GAN framework. Adding the reprojection consistency term enables to recover depth values more faithfully. Each generated depthmap is of size 224×224 and we used reprojection loss weight of 0.2 in our experiments.

5 Conclusion

In summary, we have provided a framework for generating depthmaps for new shapes with reprojection consistency using GANs. Our framework is agnostic to and works well with various GAN backbones developed for generating high resolution real world images. Our method does have some limitations. For instance, in its current form we use aligned voxel volumes with fixed viewpoints to generate our dataset. This can be handled with a simple modification by making GAN condition on camera pose. In future we would like to extend our technique to more complex data domains such as full 3D indoor and outdoor scenes.

References

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

G. Bouritsas, S. Bokhnyak, M. Bronstein, and S. Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. *arXiv preprint arXiv:1905.02876*, 2019.

A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *ACM Transactions on Graphics (TOG)*, volume 30, page 35. ACM, 2011.

S. Cheng, M. Bronstein, Y. Zhou, I. Kotsia, M. Pantic, and S. Zafeiriou. Meshgan: Non-linear 3d morphable models of faces. *arXiv preprint arXiv:1903.10384*, 2019.

A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko, and T. Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):692–705, 2017.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

H. Huang, E. Kalogerakis, and B. Marlin. Analysis and synthesis of 3d shape families via deep-learned generative models of surfaces. In *Computer Graphics Forum*, volume 34, pages 25–38. Wiley Online Library, 2015.

M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun. A probabilistic model for component-based shape synthesis. volume 31, page 55. ACM, 2012.

T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

J. M. Ollinger. Iterative reconstruction-reprojection and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 9(1):94–98, 1990.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720, 2018.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006.

A. A. Soltani, H. Huang, J. Wu, T. D. Kulkarni, and J. B. Tenenbaum. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1511–1519, 2017.

S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2626–2634, 2017.

G. Vogiatzis, C. H. Esteban, P. H. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12): 2241–2246, 2007.

J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Advances in neural information processing systems*, pages 540–550, 2017.

Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

J.-Y. Zhu, Z. Zhang, C. Zhang, J. Wu, A. Torralba, J. B. Tenenbaum, and W. T. Freeman. Visual object networks: Image generation with disentangled 3D representations. In *Advances in Neural Information Processing*

Systems (NeurIPS), 2018.