3SGAN: 3D Shape Embedded Generative Adversarial Networks

Fengdi Che, Xiru Zhu, Tianzi Yang, and Tzu-Yang Yu
McGill University
Montreal, Quebec, Canada

Abstract

Despite recent advances in Generative Adversarial Models(GAN) for image generation, significant gaps remain concerning the generation of boundary and spatial structure. In this paper, we propose a new approach to generate edge and depth information combined with an RGB image to solve this problem. More specifically, we propose two new regularization models. Our first model enforces image-depth-edge alignments by controlling the second-order derivative of depth and the first-order derivative of RGB maps, enforcing smoothness and consistency. The second model leverages multiview synthesis to regularize RGB and depth by computing the difference between an expected rotated object compared to a conditionally generated view of the object; enforcing projection consistency enables the model to directly learn spatial structures and depths. To evaluate our approach, we generated an RGB-D dataset with edge contours from ShapeNet models. Furthermore, we utilized an existing RGB-D dataset, NYU Depth V2 with edges learned by the Holistically-nested Edge Detection model.

1. Introduction

Deep learning leverages hierarchical models to analyze high-dimensional inputs, such as images, speech audio, or natural languages. One of the popular streams in this line of research is the design of generative models that approximate real data distribution and synthesize unseen data. With the emergence of adversarial generative nets(GAN) [12], researchers can now develop models generating novel data that directly computes and minimizes the distance between the generated data distribution and the real data distribution without explicitly approximating the density functions. GAN models have produced impressive results in a wide variety of fields from image generation, image style transfer, image inpainting, super-resolution to even text generation [6, 7, 13, 17].

One of the weaknesses of GAN models when generating images is that they often do not produce a clear boundary, lack reasonable shape, and spatial structure information. For instance, when generating faces, parts of the face such as eyes are frequently mispositioned, warped and blurry [24]. Even state of the art models such as BigGAN and StyleGAN suffers from such artifacts [3, 15]. The cause of this problem is due to convolution neural networks' inherent properties; its receptive fields are often limited to local structure and patterns. GANs tends to generate well locally but fails when holistically considering an image. For instance, StyleGAN can generate a baby with wrinkles! Learning a global spatial structure with limited data is a challenging endeavor. Furthermore, the lack of 3D geometric information increases the difficulty of inferring object shapes and distances from various view angles. In many cases, GAN seemingly fuses different objects into one image. Figure 1 showcases samples generated for airplanes of orthogonal projection of CAD models from ShapeNet [4] 3D dataset with different architectures. As shown in figure 1, the baseline Improved Wasserstein GAN (WGAN) [2] [10] generates blurry geometry of airplanes. Depth map information has been proved to contain spatial information from various fields, such as 3D reconstruction [26] and spatial relations learning [14]. Furthermore, it has been showed that there exists a clear correspondence between RGB images and depth maps from depth estimation research [30] [1]. Besides, using a depth map is a memory-efficient approach to represent 3D information compared with other approaches such as voxels [5] and point clouds. The number of computations for voxels increases in a cubic order of the image size, leading to expensive representation at high resolutions. In contrast, point clouds are memory-efficient but require the use of graph neural networks to represent local structure which suffers from difficulty spreading information. Other approaches consist of training specific models to model the point system, compounding errors when used with other models [22, 23].

In this paper, we propose two novel regularizations GAN models that embed 3D information into the generator networks based on depth maps and also edge maps. The former enforces image-depth-edge alignments by controlling the second-order derivative of depth maps and the first-order

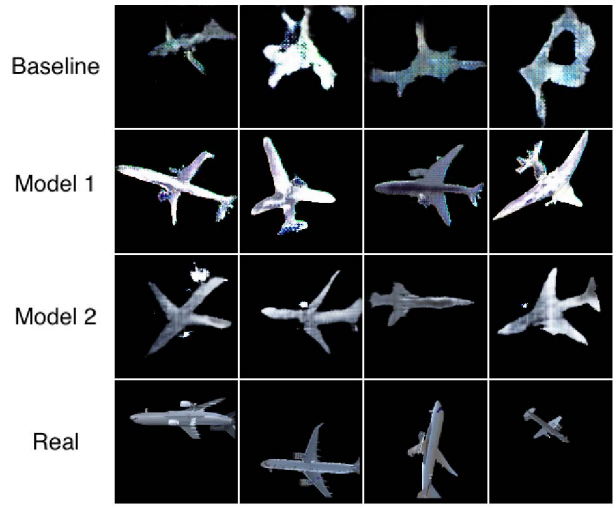

Figure 1. Comparison of results for our airplane ShapeNet dataset

derivative of RGB maps. This enforces that objects are made of locally planar surfaces and colors of objects do not change rapidly; thus it ensures smoothness and consistency within an edged area [30]. Our latter model leverages conditional GAN [21] to produce multiple views [28] at different angles. It regularizes the generator by including projection consistency loss with the assistance of depth information. Generated multiview images should be equivalent to rotated images gained by projection from 2D images to 3D objects. The purpose of our approach is not for 3D reconstruction or generating a new view. Our goal is to regularize GAN learning to create more consistent images.
The major contribution of this paper can be summarized as follows:

1. We incorporate 3D information into GAN models through regularization as a way of reducing inconsistent artifacts GAN generates.

2. Improves the learning of object shapes and spatial structures.

3. Generates depth maps, edges maps along with RGB images which can help visualize GAN's understanding of the object structure.

The remainder of the paper is organized as follows.

1. In Section II, we overview related works in the field of GAN, multiview synthesis and depth reconstruction.

2. In Section III, we discuss in detail our two proposed models and the associated loss functions.

3. In section IV, we discuss the model performance from our experiments with ShapeNet dataset [4] and NYU Depth Dataset V2 [25] with Frechet Inception Distance [11] provided.

4. In section V, we conclude this work and discuss future works.

2. Related Work

A major problem of GAN models is training instability and sample diversity. GAN models such as DCGAN or LSGAN [20, 24] suffer from mode collapse, affecting both variety and quality of images generated. Thus, multiple restarts are often required to obtain good results. However, the Wasserstein GAN(WGAN) and its extension, the improved Wasserstein GAN bypass such a problem by leveraging the earthmover loss [2, 10]. The advantages of the WGAN are its stable training and its high sample diversity at the expense of slower training speeds needed to clip the weights and to train the discriminator to near-optimal. The improved WGAN removed the need for weight clipping to enforce the Lipschitz constraint, improving training speeds. However, it still requires more time to train compared to the softmax loss of DCGAN.

To generate images with spatial structure, one solution proposed by Kossaifi et al. is to utilize pre-existing geometric information to enforce such structure [16]. By leveraging a wealth of existing data and models on facial structures, Kossaifi et al. managed to constrain GAN outputs to a reasonably shaped face with correct positions for eyes, nose, and mouth. The issue with this work is that it cannot be extended beyond faces; other objects do not have the existing models to support them. For instance, our datasets do not have such information and cannot use GAGAN's approach. Another approach for using generative models with spatial structure was proposed by Yao et Al [31]. However, their goal is to generate 3D data while ours is to regularize using 3D information to improve GAN performance in general. Similarly, Yan et al. attempt to learn 3D using multiview similar to our paper. However, their goal is also to generate multiview shapes while we are using the multiview only as a regularization. Finally, Shubham et Al proposed a multiview consistency approach for learning shape. This is similar to our idea for multiview but we apply our approach for GAN regularization rather than only learning shapes.

In contrast, SAGAN proposes a self-attention mechanism in which the GAN learns by itself which distant relationships it should focus upon [34]. Such an approach allows the GAN self to learn structural consistency and can be used for any data. As an extension to SAGAN, BigGAN dramatically expands the scale in both neural network size and dataset. It greatly improved results for generating ImageNet but its performance is affected by mode collapse. This can be verified from the weights of BigGAN provided by DeepBrain which shows that about 15-20 classes,

e.g keyboards, pickelhaubes generated are completely mode collapsed and are of low quality [3]. We find the results often lack structural consistency. For instance, a goldfish generated from BigGAN have mispositioned eyes, no heads. Furthermore, another weakness stems from the use of 3 billion images and 512 TPUs to train. Such a model is beyond to reach of most and lacks practicality for many datasets. In contrast, StyleGAN generates hyper-realistic samples at high resolution by leveraging high-level information such as pose, identity, and variation in the low level such as hair [15]. Despite its impressive performance, it can still generate interesting results like a camera instead of a mouth and other awkward artifacts. Note that for both BigGAN and StyleGAN such errors can be reduced by controlling the diversity but leads to significant mode collapse. Thus, our approach should be considered orthogonal to these work where we seek to find image regularization which enforces more consistency.

Our work is also similar to inpainting approaches in the sense that generating multiview is like inpainting the sides of an image. Goals for inpainting is simply to fill in the missing areas based on a given mask [19]. In many cases, the inpainting approach also relies on GAN based approach [32, 33] to fill in the gaps. However, our approach's main purpose is instead to regularize GAN and improve GAN performance rather than either making a 3D reconstruction or inpainting some information. The output of our multiview may be poor which would be acceptable so long it improves base results without our regularization. Similarly, the GAN discriminator may be a poor predictor of image quality but can, in general, provide good gradient for training the generator. Another work by Almalioglul et al. proposed the use of GAN for assisting depth estimation by generating warped views [1]. The model proposed is similar to research on structure from motion where pose and depth are estimated. However, unlike our paper, their model contains the ground truth for the warped image while we must generate it unsupervised. Thus, the generation of such an image should not suffer from as much structural consistency problems. Furthermore, our paper seeks to combine both RGB and depth maps to improve results.

Finally, an important source of inspiration for our work is the recent depth estimation approach from LEGO [30]. This work combines self-learned edges, depth, normals, and multiview synthesis to create sharper boundaries and smooth depths maps. Unlike our work, LEGO seeks to estimate depth from monocular motion and although unsupervised, does not attempt to generate new data as a GAN does. Another similar work relating to multiview synthesis by Zhou et al. proposes to generate a different view of the same object by copying pixels of an existing image. Furthermore, it also utilizes ShapeNet as a dataset [36]. In contrast, our work introduces similar concepts to GAN and can generate new data.

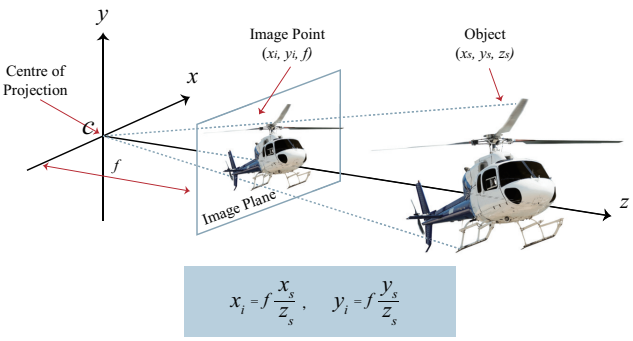

Figure 2. 2D to 3D viewpoint transformation

3. Models

In this section, we detail our 3D-information embedding models. First, we cover the transformation needed for projection between 3D objects and 2D images. This will be leveraged by our second model to obtain projected multiview by rotating and projecting the 3D object. Second, we present the first model which enforces locally planar surfaces and non-rapidly changing colors of objects. Finally, we review the design of the multiview synthesis model which has the potential to generate the same object but at different angles.

3.1. 3D Projection to 2D

The models assume that 2D images rely on some underlying 3D objects since 2D images are formed by mapping 3D objects onto a focal plane, as shown in figure 2. A 3D coordination system is introduced here to assist with computations. It is assumed that we are using a pinhole camera positioned at the origin or the center of the projection in 3D coordinates. The camera looks at the scene along the z-axis with the focal plane positioned at $z = f$. The coordinates transformation between 3D objects (x_s, y_s, z_s) and 2D images (x_i, y_i) on the focal plane $z = f$ obey the following equation:

$$\begin{aligned} x_i &= f\frac{x_s}{z_s} \\ y_i &= f\frac{y_s}{z_s} \end{aligned} \tag{1}$$

We can express this transformation by formulating the intrinsic camera matrix. Here, we use homogeneous coordinates which expand dimension by 1 to weight all coordinates value. Thus, the 3D to 2D projection using homogeneous coordinates can be captured by the intrinsic camera matrix K of size 3 by 4 where K is equal to

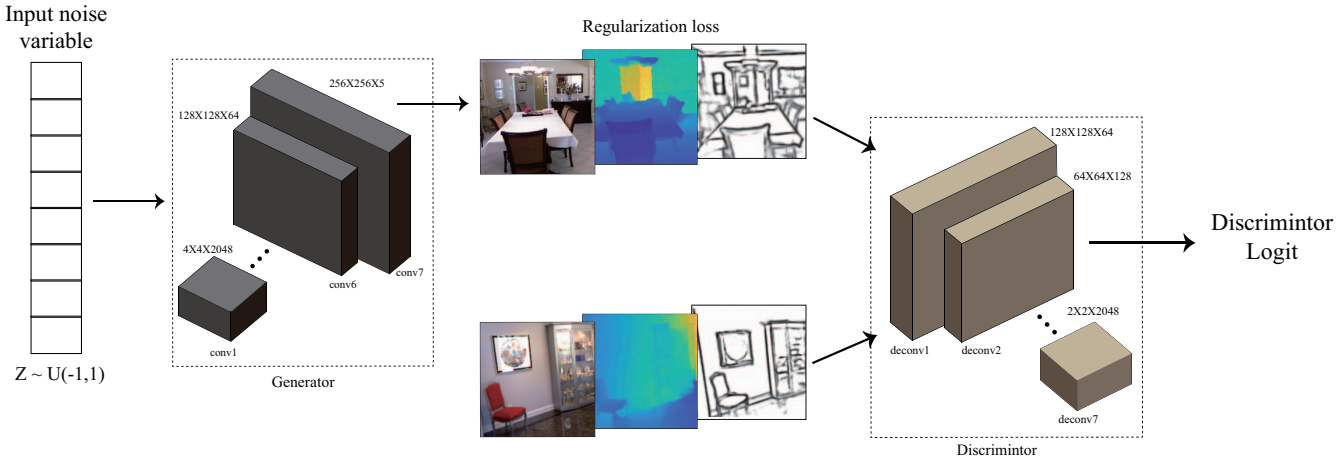

Figure 3. The architecture of model 1. Our model generates an RGB, depth and edge output compared to the normal RGB output. Also, this DC-GAN architecture with 7 layers is used in all our models.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

We define $f_X,\ f_y$ as the focal plane distances and they should have the same value f. However, they can be influenced by camera errors or non-uniformly scale during preprocessing. Finally, we define $c_x,\ c_y$ as the principal point coordinates which are coordinates of the projection center. Inversely, 2D to 3D projection can also be computed by matrix multiplications given depth values. Given a pixel $p_{(i,j)}$ and its depth d, its 3D coordinate is evaluated as

$$\phi(p_{(i,j)}) = d * K^{-1} * [i \ \ j \ \ 1] \tag{2}$$

For both datasets in this paper, the intrinsic camera matrices are given.

3.2. Image-Depth-Edge Alignment Model

The input to our GAN generator is a randomly generated uniform noise of size 512. The generator outputs a (256,256,5) image where the channels are RGB values, the depth map, and the edge map. The discriminator receives and trains on both generated and real data of 5 channels. The discriminator is assigned the role of the critic to minimize the distribution distance between fake and real data. The model can be seen in figure 3

Rather than use depth and edge map as only extra channels, we leverage a common strategy to regularize the second-order derivative of depth and the first-order derivative of RGB [30] [9] [35]. Specifically, we are inspired by the ideas from Yang et al. [30] on how to utilize edge information to improve regularization. It is assumed that each edge contour defines a surface and all surfaces are locally planar with color not rapidly changing. In a more mathematical way for each pixel $p_{(i,j)}$, the neighborhood of that pixel on the same surface as $p_{(i,j)}$ should inherit the same depth map gradient and the same image color.

Note that all the derivatives are approximated by numerical differentiation and the process is shown as follows. Given a pixel $p_{(i,j)}$, the neighbourhood is defined as $p_{(i-1,j)}$,$p_{(i+1,j)}$,$p_{(i,j)}$,$p_{(i,j)}$. The second-order derivative of depth maps is approximated along x-axis and y-axis by its neighbours.

$$\nabla_x^2 D(p_{(i,j)}) = \frac{D(p_{(i+1,j)}) - D(p_{(i,j)})}{\left\|\phi(p_{(i+1,j)}) - \phi(p_{(i,j)})\right\|^2} - \frac{D(p_{(i,j)}) - D(p_{(i-1,j)})}{\left\|\phi(p_{(i,j)}) - \phi(p_{(i-1,j)})\right\|^2} \tag{3}$$

Similarly, the first order derivative of RGB can be approximated along x+ axis or x- axis.

$$\nabla_{x+} RGB(p_{(i,j)}) = \frac{RGB(p_{(i+1,j)}) - RGB(p_{(i,j)})}{\left\|\phi(p_{(i+1,j)}) - \phi(p_{(i,j)})\right\|} \tag{4}$$

In general, the regularization loss L on depth and edge maps is computed along both x-axis and y-axis.

$$\begin{aligned} L_x &= |\nabla_x^2 D(p_{(i,j)})| + \frac{|\nabla_{x+} RGB(p_{(i,j)})| + |\nabla_{x-} RGB(p_{(i,j)})|}{2} \\ L_y &= |\nabla_y^2 D(p_{(i,j)})| + \frac{|\nabla_{y+} RGB(p_{(i,j)})| + |\nabla_{y-} RGB(p_{(i,j)})|}{2} \end{aligned} \tag{5}$$

$$L = L_x + L_y \tag{6}$$

However, directly applying this loss will lead to too many edges being created. As such, a regularization is needed to constrain the number of edges.

$$L+ = |E(p_{(i,j)})|^2 \tag{7}$$

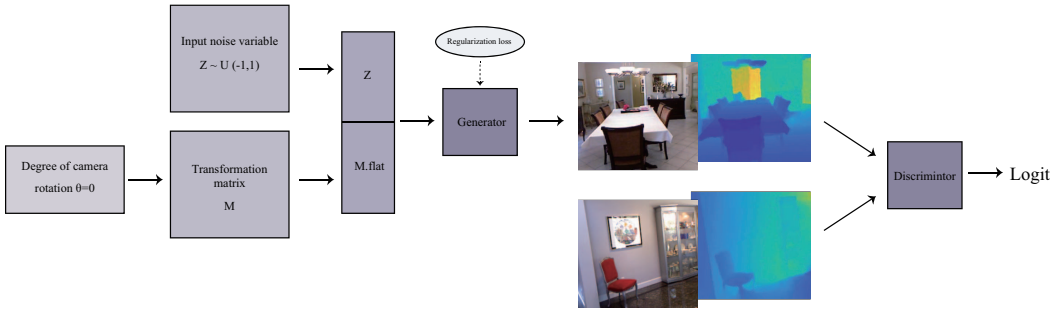

Figure 4. The architecture model of our conditional GAN model 2. The model takes a conditional transformation matrix as part of its inputs and generates RGB-D outputs.

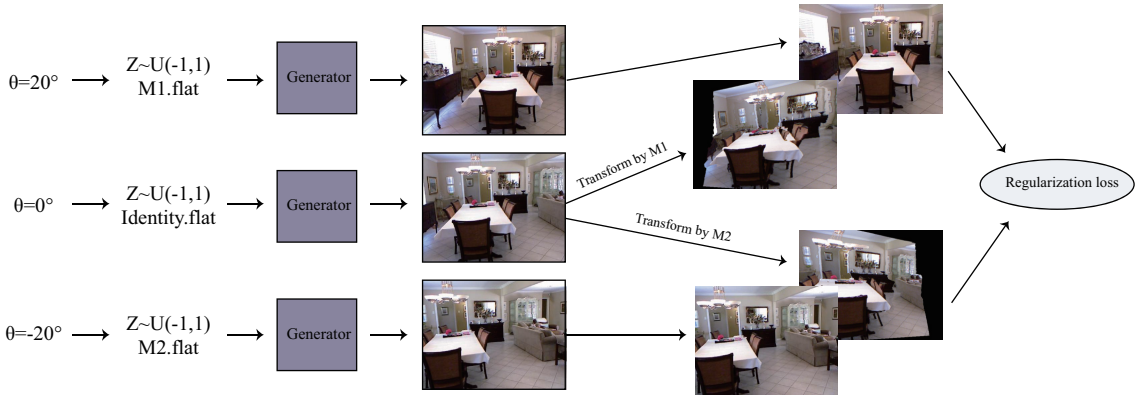

Figure 5. Our model 2 regularization loss. The initial view can be rotated to directly match a conditionally generated sample of the same rotation angle with a mask applied for out of bounds indices.

3.3. Multiview Synthesis Model

For the second model, we propose a multiview synthesis [28] [27] approach where we can force the model to embed depth information inside the generator. The depth information is used to first re-project a 2D image into 3D space. We then re-project back to obtain multiview images of the input image at different angles. Meanwhile, the model generates images from multiple views and create a generator regularization loss which verifies that the generated RGB-D images match the projected RGB-D images gained under the help of the depth information.

This model is designed as a conditional GAN [21] which adds some conditions as input along with the noise vector z into the generator; in this paper, it is chosen to be the standard transformation matrix $T(\theta)$ of view change by degree of θ. The generator will output an image at θ degree rotated view. We can then generate multiple views by keeping the z vector fixed while modifying the rotation matrix. The generator outputs an RGB-D image with the specified rotation. Meanwhile, only images with a rotation degree of 0 and real RGB-D images are passed to the discriminator. The architecture is shown in figure 4. Although generated images at different angles $\theta \neq 0°$ are not passed to the discriminator, they are supervised by our multiview synthesis loss. We first generate images at angles $\theta \in [0, 10]°, \theta = 0°$ and $\theta \in [-20, 0]°$. Then, we can transform the initial image at rotation $\theta = 0°$ to the left and the right by projecting 2D images into 3D coordinates and rotating it in 3D coordinates [8]. Let m be the 2D homogeneous coordinate of an image pixel in the original view and m_1 be the corresponding coordinate in the rotated view with the transform matrix T. As introduced in subsection 3.1, the 3D coordinate M is equivalent to

$$\begin{aligned} M &= d * K^{-1} * m \\ T * M &= d * K^{-1} * m_1 \end{aligned} \tag{8}$$

After simplification, we are able to get

$$m = K * T * k^{-1} * m_1 + \frac{K * C}{d} \tag{9}$$

where R is the rotation matrix and C is the translation matrix from T [8]. Given that rotation can include new information which the original view cannot see, a mask is necessary to handle pixels indexed outside the boundaries of the image

and prevent any loss penalties. This regularization loss is shown in detail in figure 5 along with images created by our transformations.

4. Experimental Settings and Evaluations

In this section, we introduce the experiment settings, training procedure. We further present the results of our two models which are evaluated qualitatively and quantitatively. The models are shown to outperform the state-of-arts WGAN under the same settings which generate clearer shapes and arrange objects with spatial orders. We also use the Frechet Inception distance [11] to provide comparable performance metrics for our models. Note that our goal here is not to generate multiview or 3D as a goal but to improve the results for the normal GAN output using 3D information as regularization.

4.1. Experiment setting

Datasets and pre-processing

For our training dataset, we generated our RGB-D data using ShapeNet CAD models [4]. To do so, we used Unity Engine and Ray tracing with an orthogonal projection. We positioned 5 cameras at the +z, +x, -x, +y, and -y-axis. For each model, 15 shots are taken at 3 sets of different rotation, translation and scale picked from a uniform random distribution. Each image's background was set to be completely black to improve learning speed. The resolution of the RGB-D obtained was 256 by 256. Due to time constraints and technical difficulties, we limited to the plane models of ShapeNet. The airplane models are highly varied ranging from jumbo jets to fighter planes to even helicopters and show considerably more variation compared to other shapes such as cars from ShapeNet. After pruning images where the airplane was scaled too small or at strange angles, we were left with 57,088 RGB-D images. As we set the background to black, obtaining the contour edges of the plane is straightforward resulting in a 5-channel image.

To study complex scenes with variegated objects, we used the NYU Depth Dataset V2 [25] for various indoor scenes recorded by a Microsoft Kinect RGB and depth cameras. It is comprised of video sequences of 1449 pairs of aligned RGB and depth images labeled by object classes and object contours and video sequences of raw datasets without labels. The raw datasets used here for bedrooms, living rooms, dining rooms, and offices contain depths taken at different angles from the RGB images with missing values. We selected images from the video sequences while attempting to avoid repetitive scenes. Then, those images were pre-processed by the given toolbox to project the depths onto the coordinates of the images and to fill missing values of depth by colorization [18]. Next, the edge labels were generated by side outputs of the Holistically-nested Edge Detection models [29]. The edge detection model is pre-trained by the paper authors Xie and Tu. Finally, 5-channel images of size 4,460 were re-scaled to 256 by 256 pixels to match our models.

Note that for both our ShapeNet dataset and NYU dataset, depth was normalized to [0, 1] where the closer the data, the smaller its depth. However, 0 is considered a special value where the depth is infinity.

Training settings

All the experiments are completed on either an NVIDIA GEFORCE 1080 Ti or a cloud server using a single NVIDIA TESLA P100. Both generator and discriminator utilized an improved WGAN architectures and loss scaled for images of 256 by 256 instead of 64 by 64. We set $lambda$ to 10 plus our regularization losses. The models are trained by RMSProp optimization with learning rate 0.0002 and batch size 32. We found RMSProp worked better compared to Adam even though [10] claimed Adam works well with their loss. The intrinsic camera matrix K we used is the scaled Kinect camera matrix according to the size of images. The Kinect intrinsic camera matrix for (640,480) pixels image is

$$\begin{bmatrix} 518.8579 & 0 & 325.5824 \\ 0 & 519.4696 & 253.7362 \\ 0 & 0 & 1 \end{bmatrix}$$

This matrix was provided from the NYU dataset.

Model 1 sets the weight of the regularization loss to 0 during the first 7 epochs, increasing gradually to 0.55 and finally reduced to 0.3 after 135 epochs. We do so much that the generator can first stabilize and then be steadily affected by our regularization loss. To improve results towards the end of the training, we reduce regularization constraints so to allow the model to learn more details. For Model 2, instead of adjusting loss weight, we slowly adjust the rotation range starting from 0. This functions similarly to a weight adjustment as a loss with smaller rotations will be reduced. In this manner, the model can smoothly learn the rotation. Otherwise, a problem we faced is that the model could not properly learn the rotation and resulted in blank generated images since it could only minimize loss that way. Finally, the rotation coefficients were set to be between [-20, 20]. This small rotation angle avoids having objects rotated completely out of view which can occur depending on the depth of the object. Note that the baseline model has the same setting except that it does not add the regularization losses.

4.2. ShapeNet Results

We first trained on trained on ShapeNet data with 57,088 images. The results can be seen in figure 6. All the models

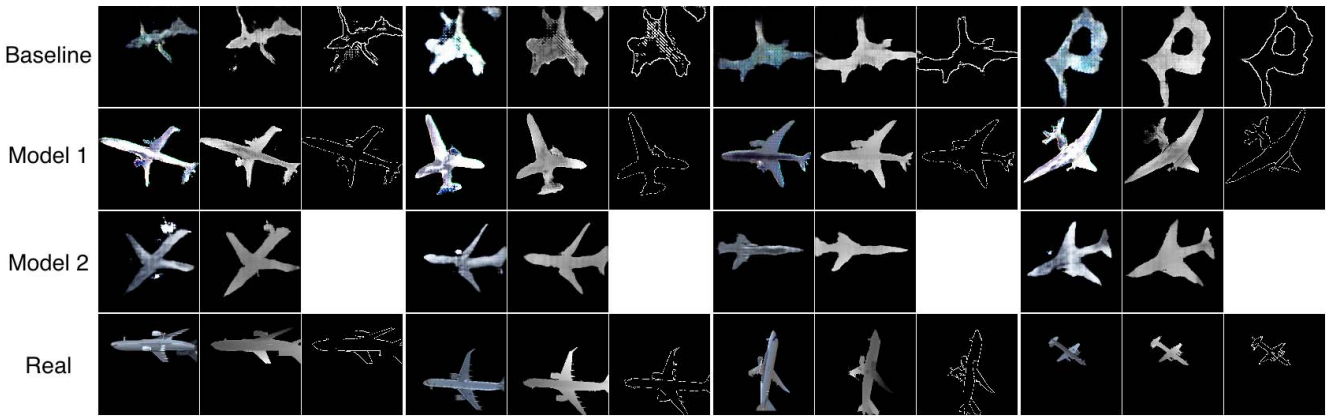

Figure 6. Model results for ShapeNet Dataset. The 1st row is from baseline WGAN, 2nd row is from Model 1, 3rd row is from Model 2, and the last row is real images.

Figure 7. Sampled 3D reconstruction results from generated depth + RGB from both Model 1 and Model 2. The 3D view was generated from the Unity Engine. Note that there is some roughness in the generated depth

are trained using the adjusted DC-GAN architecture with the same number of layers and filter number on the same data. We noticed during training was that the GAN quickly was associating the edge and depth with its RGB. This means for edges generated, the results were always following the contour of the object. The addition, our regularization losses led to more constrained and reasonable shapes of planes generated, enhancing the quality of generated images dramatically from the baseline.

To further verify the correctness of depth generation, 3D reconstruction from the depth and RGB views are provided in figure 7. We can see the general shape and spatial structure from the object. We rotated all objects to the right to show the contour of the airplane. Note that since a view is only from one side, the depth does not create a complete object but a partial 3D structure. An interesting point about the depth map generated is that when displayed in 3D, there are frequent rough spots and spikes in depth. At worst, this can lead to points completely detached from the object itself. This is not noticeable from the 2D depth map which seems reasonably smooth. The most common spike occurs around the contour of the object where the depth switches from distance to the object to infinity. This problem is likely due to the fact these points are right between the depth of the object and infinity. We believe a smoothing of the area around the contour can reduce the impact of this problem.

Model	Score
Real Data	8.5
Model 1	122.56
Model 2	125.29
Baseline	175.88

Table 1. Frechet Distance for ShapeNet Dataset

Finally, Frechet Inception Distances(FID) [11] were computed for the ShapeNet models with 4,000 fake and 4,000 real images. We can see that the performance of both Model 1 and Model 2 shows no obvious difference. However, both models surpass the baseline model significantly 1. However, with the current FID score, there re-

Figure 8. Model 1 generation results trained on NYU Depth V2 Dataset

mains plenty of room for improvements.

4.3. NYU Depth V2 Results

Our model 1 is also trained with 4,460 NYU Depth V2 images in figure 8. Though the generated textures of objects are blurry, we can discern each object in a complex scene; this is especially evident from observing depth or edge outputs. The matching between images and edge maps is reasonable and present us with meaningful object contours. This is in contrast to the baseline model where only color blocks are learned without clear objects. Moreover, spatial structures of rooms such as depths of rooms, orders of objects and flows of lights are cleanly learned in the generated images, while the baseline model does not perform as well in terms of distinguishable depths. The imperfect results, especially the lack of details can be partially explained to the nature of the dataset. The data is drawn from video sequences with unavoidable repetitions and sometimes only shows the corners of a room. Furthermore, despite being a fairly small dataset, the NYU dataset shows significant scene diversity ranging from bedrooms, living rooms, dining rooms, and offices which can be difficult for a GAN to learn with so few data points. Overall, we can see that adding our regularization did improve overall performance even if somewhat poor.

5. Discussion and Future Works

In this paper, we have introduced two new regularization models to improve GAN image generation ability by using 3D information as regularization. Our generator is therefore trained to embed both edge information and depth information and is regularized by that information. Our models possess major potentials for performance improvements using deeper network architectures and improving datasets. Though our models require more experimental results and analysis, the designs of our models do work as expected and the improvements in objects' shapes and spatial structures from the baseline are visible. The edge and depth information not only brings extra contour and position information but also sketches out a global spatial structure of the image. Furthermore, the alignment of RGB, depth and edge maps are useful for understanding what the GAN is trying to create, especially for a complex scene such as the NYU Depth V2 dataset.

Another observation is that Model 1 seems to supervise shapes and structures more proficiently while model 2 is more adept at generating smoother depth maps. We reason that it is because model 2 does not regularize edge-depth values but instead focuses on continuous transformations in 3D which smooths its final result. Therefore for future work, it would be natural to combine these 2 models further improve generated details in RGB-D images. Moreover, during the training of model 1, we noticed that regularization loss did not decrease much itself during training but led to adjustments in the generator WGAN loss. During this process, the loss would become unstable and then slowly readjust; studying such a process and how it deviates from the normal learning process can be a further topic.

Although our view synthesized results are not state of the art, the multiview regularization has shown good performance in improving GAN results. Further investigation should be made for whether improved generated multiview can augment regularization performance. In addition, current angles changes are small but perhaps larger angles can improve regularization. Another approach for improving our model is to train the discriminator with generated images at different angles. This can be further expanded if we add real images at different angles but it requires adjustments of architectures to efficiently take advantages of multiview real images. Furthermore, a model using the attention architecture could also be advantageous as shown in SAGAN.

References

[1] Yasin Almalioglu, Muhamad Risqi U Saputra, Pedro PB de Gusmao, Andrew Markham, and Niki Trigoni. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. *arXiv preprint arXiv:1809.05786*, 2018.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] 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 preprint arXiv:1512.03012*, 2015.

[5] 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 *European conference on computer vision*, pages 628–644. Springer, 2016.

[6] Zongyuan Ge Michael Milford "Fangyi Zhang, Jrgen Leitner and Peter Corke". Adversarial discriminative sim-to-real transfer of visuo-motor policies. 2018.

[7] William Fedus, Ian Goodfellow, and Andrew M. Dai. maskGAN: Better text generation via filling in the _____. In *International Conference on Learning Representations*, 2018.

[8] Christoph Fehn. Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. In *Stereoscopic Displays and Virtual Reality Systems XI*, volume 5291, pages 93–105. International Society for Optics and Photonics, 2004.

[9] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp 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.

[12] Mehdi Mirza Bing Xu David Warde-Farley Sherjil Ozair Aaron Courville "Ian Goodfellow, Jean Pouget-Abadie and Yoshua Bengio". Generative adversarial nets. In *Neural Information Processing Systems (NIPS)*, 2014.

[13] Phillip Isola "Jun-Yan Zhu, Taesung Park and Alexei A. Efros". Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2017.

[14] Philipp Jund, Andreas Eitel, Nichola Abdo, and Wolfram Burgard. Optimization beyond the convolution: Generalizing spatial relations with end-to-end metric learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7. IEEE, 2018.

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[16] Jean Kossaifi, Linh Tran, Yannis Panagakis, and Maja Pantic. Gagan: Geometry-aware generative adversarial networks. *arXiv preprint arXiv:1712.00684*, 2017.

[17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[18] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM transactions on graphics (tog)*, volume 23, pages 689–694. ACM, 2004.

[19] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[20] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE, 2017.

[21] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[22] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[23] 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.

[24] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[25] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012.

[26] Edward Smith, Scott Fujimoto, and David Meger. Multi-view silhouette and depth decomposition for high resolution 3d object representation. In *Advances in Neural Information Processing Systems 31*, 2018.

[27] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J. Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[28] Dong Tian, Po-Lin Lai, Patrick Lopez, and Cristina Gomila. View synthesis techniques for 3d video. In *Applications*

of Digital Image Processing XXXII, volume 7443, page 74430T. International Society for Optics and Photonics, 2009.

[29] Saining "Xie and Zhuowen" Tu. Holistically-nested edge detection. In *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[30] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. Lego: Learning edge with geometry all at once by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 225–234, 2018.

[31] Shunyu Yao, Tzu Ming Hsu, Jun-Yan Zhu, Jiajun Wu, Antonio Torralba, Bill Freeman, and Josh Tenenbaum. 3d-aware scene manipulation via inverse graphics. In *Advances in Neural Information Processing Systems*, pages 1887–1898, 2018.

[32] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5485–5493, 2017.

[33] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018.

[34] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[35] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017.

[36] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016.