3D-Aware Indoor Scene Synthesis with Depth Priors
– *Supplementary Material* –

Zifan Shi[1], Yujun Shen[2], Jiapeng Zhu[1], Dit-Yan Yeung[1], and Qifeng Chen[1]

[1]The Hong Kong University of Science and Technology [2]ByteDance Inc.
{vivianszf9, shenyujun0302, jengzhu0}@gmail.com
{dyyeung, cqf}@cse.ust.hk

A Overview

This supplementary material is organized as follows. Sec. B describes the implementation details of our DepthGAN, including the network structures and the training configurations. Sec. C introduces the datasets we used for experiments. Sec. D provides the implementation of our baseline approaches. Sec. E introduces the rotation consistency loss used for training. Sec. F discusses how we use the synthesized RGBD images for 3D visualization. Sec. G evaluates our switchable discriminator on the task of depth estimation. Sec. H discusses the out-of-distribution generation of our method. Sec. I provides more qualitative results to demonstrate the continuous 3D control achieved by our DepthGAN.

B Implementation Details

B.1 Network Architectures

Dual-path Generator. We use the generator of StyleGAN2 [5] as the backbone of the depth generator and the appearance renderer, which produce the 1-channel depth and the 3-channel image, respectively. The path length regularization is removed from both of them.

Switchable Discriminator. The switchable discriminator follows the structure of the discriminator in StyleGAN2 [5]. Given an image, the native discriminator employs an input layer, $\mathbf{\Gamma}_0$, to project it to the feature space. We supplement $\mathbf{\Gamma}_0$ with an additional layer, $\mathbf{\Gamma}_1$, for the toggle between the RGB image and the depth image. $\mathbf{\Gamma}_1$ takes in a depth image and outputs the same number of channels as $\mathbf{\Gamma}_0$. In this way, if the input is an RGB image, only $\mathbf{\Gamma}_0$ will be activated. If the input is an RGBD image, the outputs of $\mathbf{\Gamma}_0$ and $\mathbf{\Gamma}_1$ are aggregated using element-wise sum. On the other hand, the architecture of depth prediction branch is shown in Fig. S1. It uses the features at resolutions 16^2, 32^2, and 64^2 as the inputs. Starting from the lowest resolution (*i.e.*, 16^2) each feature is first transformed using a small network with skip connection, then upsampled to double the resolution, and finally concatenated onto the feature from the next

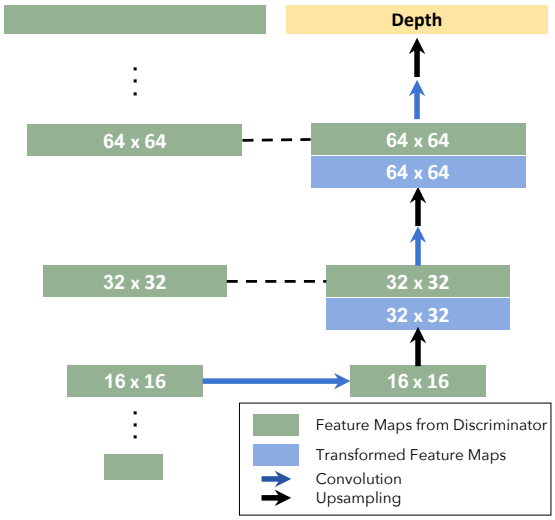

Fig. S1. Depth prediction branch in the proposed switchable discriminator

stage. Finally, this branch outputs a k-channel probability map, formulating depth prediction as a classification problem [3].

Codebase. Our implementation is based on Hammer [13].

B.2 Training Configurations

All the components of our proposed DepthGAN are trained in turn. When only one part is trained, the gradients of other parts will be turned off. Our training process is as follows: (1) The dual-path generator is updated with adversarial loss $\mathcal{L}_{adv}^{g}$. (2) The depth generator is updated with rotation consistency loss $\mathcal{L}_{rot}^{d}$. (3) The appearance renderer is updated with rotation consistency loss $\mathcal{L}_{rot}^{rgb}$ and depth prediction loss $\mathcal{L}_{dp}^{f}$. (4) The discriminator is updated with adversarial loss $\mathcal{L}_{adv}^{d}$, depth prediction loss $\mathcal{L}_{dp}^{r}$ and R_1 regularization [6]. We train with Adam optimizer and use a batch size of 64. The learning rate for both the generator and the discriminator is 1.5e-3. The weight of the R1 regularization is 0.3 for resolution 128^2 and 0.5 for resolution 256^2. $\{\lambda_i\}_{i=1}^{4}$ are set to 50, 0.3, 1e-3 and 0.8 at resolution 128^2. At resolution 256^2, $\{\lambda_i\}_{i=1,i\neq 2}^{4}$ are set to 50, 0.001, and 0.8, while λ_2 is 0.5 for LSUN bedroom and 0.4 for LSUN kitchen. The focal length is fixed to 26mm. The angles are uniformly sampled from -15° to 15°. The whole network is trained from scratch. All the experiments run on 8 Tesla V100 GPUs for about 2-3 days.

C Datasets

We use LSUN bedroom and kitchen datasets [18] for all experiments, which contain about 3M and 2M RGB images, respectively. Compared with object

datasets used in prior arts [1,8,11], indoor scenes are far more challenging due to their high diversity. We take the last 50k images from each dataset as the validation set, while the rest are used for training. We utilize the pre-trained model from [17] to predict the depth for each RGB image. The images are resized and center-cropped to the resolutions of 128 and 256.

D Baselines

All the baselines are trained from scratch and we ensure that all the training of baselines is converged. The field of views are the same as ours. When testing, the range of azimuth is set to 30 degrees and the elevation is fixed to the frontal view. The details for each baseline are as follows:

SeFa [14]. We factorize the weights in the first three layers of StyleGAN2 [5] and find the most relevant direction that affects the pose of the bedroom or kitchen. The pre-trained models can be found in GenForce [12]. When editing the pose, the truncation is set to 0.8, and FID is computed on the edited images. We use the pre-trained model [17] to estimate a depth map from the generated image first, and then calculate the RP and RC using the predicted depth.

HoloGAN [7]. We fail to reproduce HoloGAN with the official implementation, hence we do not report the quantitative results. The qualitative results of bedrooms are borrowed from the original paper [7], while those of kitchens are generated by our poorly reproduced model.

GRAF [11]. We use the official implementation of GRAF. Since we observe that a slightly larger range of azimuth produces better results, we set the range of azimuth to 0°-60° and the elevation to 80°-95° in the training phase. However, when testing, we randomly sample the azimuth from the 30 degrees in the middle and fix the elevation to 90°. White background is set to false. Others are kept as the default ones.

GRAF(D). We modify the official implementation of GRAF by adding an extra discriminator to lead the learning of depth image generation. The new discriminator shares the same architecture as the original one, except that the input is changed to the one-channel depth image. We obtain the depth image by accumulating the depth with the help of sigma value. Other configurations are the same as those we used for GRAF. Depths are rendered from the learnt radiance field, and then normalized to [0, 1] to match the range of ground-truth depth.

GIRAFFE [8]. We use the official implementation of GIRAFFE. The settings are kept the same as those for LSUN church. The depth image is obtained on the feature volume before 2D neural rendering by accumulating the depth with the help of sigma value. Since the output depth map is of resolution 16^2, we then upsample it to the size of the input image through bi-linear interpolation.

π-GAN [1]. We use the official implementation of π-GAN. The range of azimuth is set to $-30°$-$+30°$, and the elevation is fixed to 90°. During testing, the azimuth is sampled from the 30 degrees in the middle, and the elevation is set to 90°.

White background is set to false. Other settings follow the configuration for CARLA dataset provided by the authors.

π-GAN(D). We make modifications on the original implementation of π-GAN. To incorporate the depth information into training, the input of the discriminator is changed from the three-channel RGB image into the four-channel RGBD image. Other hyper-parameters are kept the same as those we used for π-GAN. Depths are rendered from the learnt radiance field, and then normalized to [0, 1] to match the range of ground-truth depth.

RGBD-GAN [9]. We use the official implementation of RGBD-GAN. The range of azimuth is set to $-30°$-$+30°$, while rotation angles around x-axis and z-axis are set to zero. During testing, the azimuth is sampled from the 30 degrees in the middle. Other settings are kept the same as the official configuration.

StyleNeRF [4]. We use the official implementation of StyleNeRF. During training, the azimuth is randomly sampled from $-30°$ to $+30°$, and the elevation is fixed to $90°$. When testing, we sample the azimuth from the middle 30 degrees. Other settings follow the configuration provided by the authors. Due to the limited time, we report results on LSUN bedroom [18] only.

VolumeGAN [16]. We use official implementation of VolumeGAN. The range of azimuth is set to $-30°$-$+30°$, and the elevation is fixed to $90°$. During testing, the azimuth is sampled from the 30 degrees in the middle. Other settings follow the configuration provided by the authors. Due to the limited time, we report results on LSUN bedroom [18] only.

E Rotation Consistency Loss

In this section, we discuss the implementation details of rotation consistency loss. The images $\mathbf{I}^f_{rgbd,1}$ and $\mathbf{I}^f_{rgbd,2}$ are generated under angles θ_1 and θ_2. $\mathbf{I}^f_{rgbd,2}$ will first be projected to the 3D space as a point cloud using the fixed camera intrinsic parameter $\mathbf{K}$. Then we rotate the point cloud around the central axis, which passes through the center point of xz-plane and is parallel to y-axis. The rotation angle is the difference between θ_1 and θ_2. The rotation axis and the rotation angle form the rotation matrix $\mathbf{R}$, which is then used to transform the points accordingly. After the rotation, we get the new coordinates for each pixel in $\mathbf{I}^f_{rgbd,2}$. Then, we use the *grid_sample* function in PyTorch to query the RGB value in $\mathbf{I}^f_{rgbd,1}$ according to the new coordinates from $\mathbf{I}^f_{rgbd,2}$, which gives us the rotated image $\mathbf{I}^{f,rot}_{rgbd,1}$. We get a mask $\mathbf{M}$ through coordinate comparison to filter out the out-of-boundary regions simultaneously. Therefore, the output image $\mathbf{I}^{f,rot}_{rgbd,1}$ should be the same as $\mathbf{I}^f_{rgbd,2}$ in the valid regions. The rotation consistency loss is then calculated between $\mathbf{I}^{f,rot}_{rgbd,1} \circ \mathbf{M}$ and $\mathbf{I}^f_{rgbd,2} \circ \mathbf{M}$, where $\circ$ denotes element-wise multiplication.

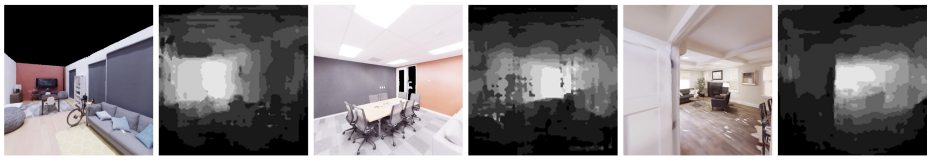

Fig. S2. Estimated depth by our switchable discriminator on the Replica dataset [15]

Fig. S3. Out-of-distribution generation. Recall that our model is trained by setting the rotation range as $-15^\circ \sim 15^\circ$. At the inference stage, our approach allows to synthesize a sample from a novel viewpoint

F 3D Visualization

To visualize the point clouds of each synthesized scenes, we first project each generated RGBD images the 3D space as point clouds using a fixed camera intrinsic parameter $\mathbf{K}$. Then, we use ICP registration [10] implemented in Open3D [19] for point cloud registration. Finally, we fuse these point clouds into one point cloud and show it from different viewpoints in Fig. 1 of the submission.

G Depth Estimation

Though depth estimation is not under the main scope of this work, we evaluate the switchable discriminator on Replica dataset [15] to validate its transferability. We get 2.36 for 10-class cross-entropy error over 10K samples provided by [2]. Some examples are visualized in Fig. S2.

H Out-of-Distribution Generation

During training, we sample the angle of rotation from -15° to 15°. When extrapolating the angle outside of that range, as shown in Fig. S3, the model can generate the rotated geometry but lacks the 3D consistency as expected.

I More Results

More qualitative results are shown in Fig. S4 and Fig. S5. A demo video is also available to show the continuous 3D control achieved by our DepthGAN.

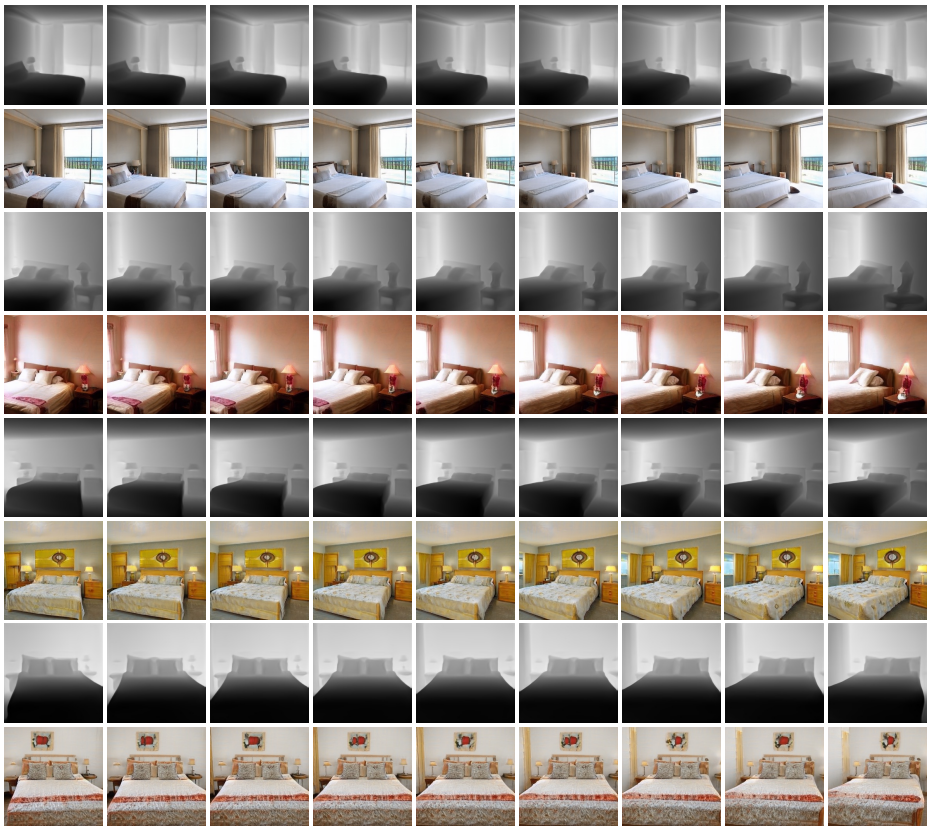

Fig. S4. Qualitative results on LSUN bedrooms [18]

Fig. S5. Qualitative results on LSUN kitchens [18]

References

1. Chan, E., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-GAN: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
2. DeVries, T., Bautista, M.A., Srivastava, N., Taylor, G.W., Susskind, J.M.: Unconstrained scene generation with locally conditioned radiance fields. In: Int. Conf. Comput. Vis. (2021)
3. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2018)
4. Gu, J., Liu, L., Wang, P., Theobalt, C.: Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. In: Int. Conf. Learn. Represent. (2022)
5. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
6. Mescheder, L., Nowozin, S., Geiger, A.: Which training methods for gans do actually converge? In: Int. Conf. Mach. Learn. (2018)
7. Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.L.: HoloGAN: Unsupervised learning of 3d representations from natural images. In: Int. Conf. Comput. Vis. (2019)
8. Niemeyer, M., Geiger, A.: GIRAFFE: Representing scenes as compositional generative neural feature fields. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
9. Noguchi, A., Harada, T.: RGBD-GAN: Unsupervised 3d representation learning from natural image datasets via rgbd image synthesis. In: Int. Conf. Learn. Represent. (2020)
10. Rusinkiewicz, S., Levoy, M.: Efficient variants of the icp algorithm. In: Int. Conf. on 3-D digital imaging and modeling (2001)
11. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: GRAF: Generative radiance fields for 3d-aware image synthesis. In: Adv. Neural Inform. Process. Syst. (2020)
12. Shen, Y., Xu, Y., Yang, C., Zhu, J., Zhou, B.: Genforce lib for generative modeling. https://github.com/genforce/genforce (2020)
13. Shen, Y., Zhang, Z., Yang, D., Xu, Y., Yang, C., Zhu, J.: Hammer: An efficient toolkit for training deep models. https://github.com/bytedance/Hammer (2022)
14. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in GANs. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
15. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
16. Xu, Y., Peng, S., Yang, C., Shen, Y., Zhou, B.: 3d-aware image synthesis via learning structural and textural representations. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
17. Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S., Shen, C.: Learning to recover 3d scene shape from a single image. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)

18. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
19. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018)