Supplementary Materials for: Neural Wireframe Renderer: Learning Wireframe to Image Translations

Yuan Xue, Zihan Zhou, and Xiaolei Huang

College of Information Sciences and Technology,
The Pennsylvania State University, University Park, PA 16802, USA

1 Joint Wireframe and Image Synthesis

Our joint representation learning framework is general and may also benefit other image synthesis tasks. In this section, we report preliminary results on extending this framework to the noise-to-image task. Recently, coarse-to-fine multi-scale models [1–4] have been shown to produce visually pleasing results. However, such models often rely on having a large training set and do not explicitly take structural integrity into consideration.

In this work, we choose the state-of-art StackGANv2 [4] model as the baseline model for our experiments. In the baseline model, the generator takes a random noise vector as input and output an image. Instead of generating images only, we propose a GAN model to generate images and their corresponding wireframes simultaneously. An illustration of our proposed GAN model with joint representation learning can be found in Fig. 1. Unlike the original StackGANv2 model, we first map the input noise vector to a shared latent space of wireframe and image through the first generator G_0, then two separate branches of coarse-to-fine generators take the joint representation and generate wireframes and images, respectively. Although we do not have explicit supervision upon the joint representation, the wireframe-based adversarial learning guarantees that the learned representation contains enough structure information.

Note that our GAN model for image and wireframe generation does not require paired wireframes and images during training, as it uses separate discriminators for wireframes and images. Thus, we can potentially use wireframes and images from different sources, which makes the model scalable to much larger datasets.

Following [4], our GAN objective consists of two parts: the traditional adversarial loss and a color-consistency regularization term. Since we are generating both wireframe and image, we also apply a structure-consistency regularization term to the generated wireframes. More specifically, the adversarial objective for the ith generator G_i and the ith discriminator D_i is defined as

$$\begin{aligned}\max_{\theta_D}\min_{\theta_G}\mathcal{L}_i^{\text{adv}} =& \mathbb{E}_{x_i}[\log D_i^w(x_i)] + \mathbb{E}_{z_i^w}\log(1 - D_i^w(G_i^w(z_i^w)))] \\ &+ \mathbb{E}_{y_i}[\log D_i^s(y_i)] + \mathbb{E}_{z_i^s}\log(1 - D_i^s(G_i^s(z_i^s)))],\end{aligned} \tag{1}$$

where z_i is the input to the ith generator, x and y represent real wireframes and images, respectively. The superscript indicates the index of the generator/discriminator branch and G_0 is shared by both branches.

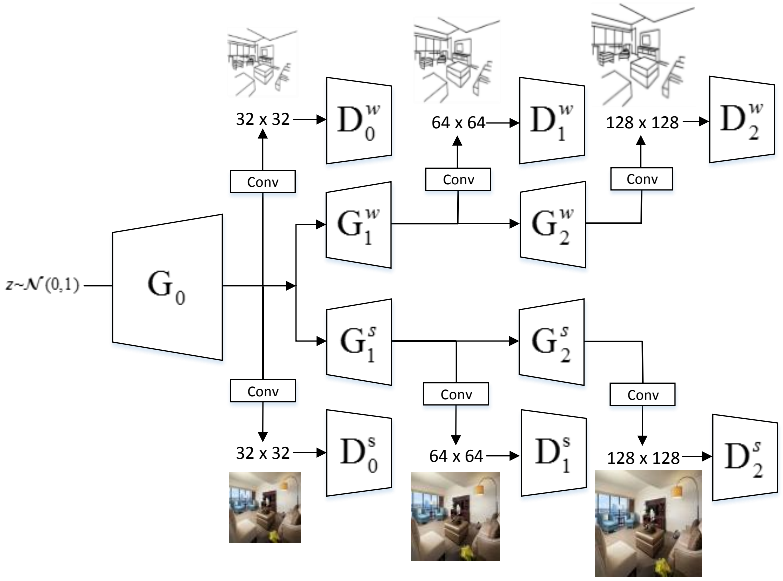

Fig. 1. Network architecture of our GAN model for joint wireframe and image generation. The backbone generators and discriminators are the same as StackGANv2 [4].

Given a mini-batch of N generated wireframes $\hat{x}_i^n$ and images $\hat{y}_i^n$ at the ith scale, the color- and structure-consistency regularization term is defined as

$$\begin{aligned}\mathcal{L}_i^{\text{con}} = \frac{1}{N}\sum_{n=1}^{N} & ||\lambda_1 \boldsymbol{\mu}_{\hat{x}_i^n} - \boldsymbol{\mu}_{\hat{x}_{i-1}^n}||_2^2 + ||\lambda_2 \boldsymbol{\Sigma}_{\hat{x}_i^n} - \boldsymbol{\Sigma}_{\hat{x}_{i-1}^n}||_2^2 \\ & + ||\lambda_1 \boldsymbol{\mu}_{\hat{y}_i^n} - \boldsymbol{\mu}_{\hat{y}_{i-1}^n}||_2^2 + ||\lambda_2 \boldsymbol{\Sigma}_{\hat{y}_i^n} - \boldsymbol{\Sigma}_{\hat{y}_{i-1}^n}||_2^2,\end{aligned} \tag{2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the mean and covariance of pixel values of the given wireframe or image.

During the training of each discriminator in our model, only the adversarial loss $\mathcal{L}^{\text{adv}}$ is applied. When we train the ith generator, the total loss is the sum of the adversarial loss and the consistency regularization loss, *i.e.*, $\mathcal{L}_i^G = \mathcal{L}_i^{\text{adv}} + \alpha_{\text{con}}\mathcal{L}_i^{\text{con}}$, where α_{con} is the scaling factor that controls the relative influence of the two loss terms.

1.1 Implementation Details

Our proposed GAN model is built upon a StackGANv2 [4] backbone. After the shared generator G_0 which maps the input vector z to a joint embedding, the wireframe generator and image generator use separate coarse-to-fine generators G_1^w, G_1^s and G_2^w, G_2^s to generate wireframes and images at different scales. Before generating each wireframe/image, the learned features will go through a 3×3 convolution block including batch normalization and relu activation, then followed by a 7×7 convolution and tanh activation to generate the wireframe or image. We set $\lambda_1 = 1$, $\lambda_2 = 5$ and $\alpha_{\text{con}} = 50$ to

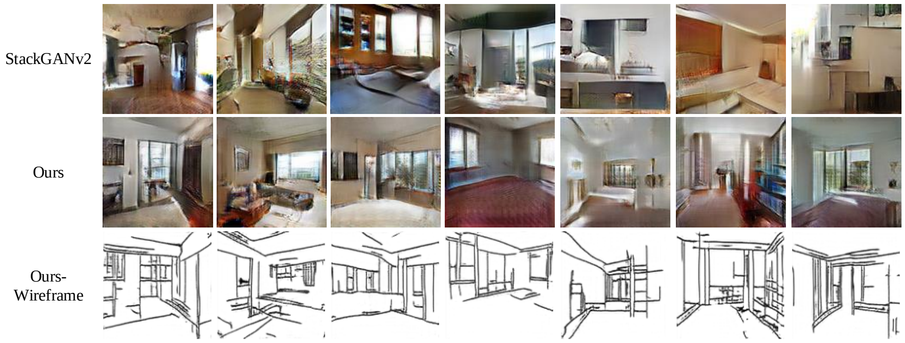

Fig. 2. Qualitative comparisons of image generation models. The first row contains generated images by the baseline StackGANv2 [4] model. The second and third rows are paired images and wireframes generated by our model.

Table 1. Quantitative comparisons between image generation models. The inception score of the real images in the test set is provided as reference.

Method	IS↑	FID↓
StackGANv2 [4]	2.92	**49.76**
Ours	**3.08**	50.96
GT	3.21	-

be consistent with the original StackGANv2. The training is done by Adam optimizer with fixed learning rate $2e-3$. The batchsize is 64 and the maximum number of training epochs is 500. No LSGAN loss is applied during training.

The data pre-processing is the same as in the image translation experiments except that we do not apply color jitter augmentation for GAN training. During inference, only the highest resolution images (we use 128×128 in joint synthesis experiments) are evaluated.

1.2 Experiment Results

Fig. 2 shows example image synthesis results of StackGANv2 and our model. As one can see, our model generates images with room layouts which are more geometrically meaningful and better align with the typical layout of real rooms. It is also worth noting that the wireframes generated by our model align quite well with the images, despite that fact that no direct supervision is provided w.r.t. the alignment. This is a strong indication of the effectiveness of the joint representation learning module.

Table 1 reports quantitative comparison results between the baseline StackGANv2 model and our GAN model. Specifically, we randomly generate 500 images for each model, then calculate the IS and FID scores based on the generated images and real images in the test set. Here we note that, the focus of our work is more on the geometric constraints and structural integrity of the generated images. However, existing

GAN metrics, such as the IS and FID scores, are mainly designed to measure the perceptual quality and the diversity of the generated images, and cannot well capture the structure information. While preserving structural integrity in image synthesis remains a challenge for current GAN models, we hope that our proposed model and preliminary results provide some useful insight for future research. We also expect that our model can be improved by training with larger datasets from multiple sources and utilizing advanced GAN models.

2 Additional Wireframe Rendering Results

We provide more wireframe-to-image translation results and wireframe detection results in Fig. 3. Note the structural similarity between our synthesized images and the real images. Also, by comparing the wireframe detection results from synthetic images with real images, we can observe that wireframes detected from real images contain more false positives. This may explain why our synthetic images achieves higher sAP scores than real images.

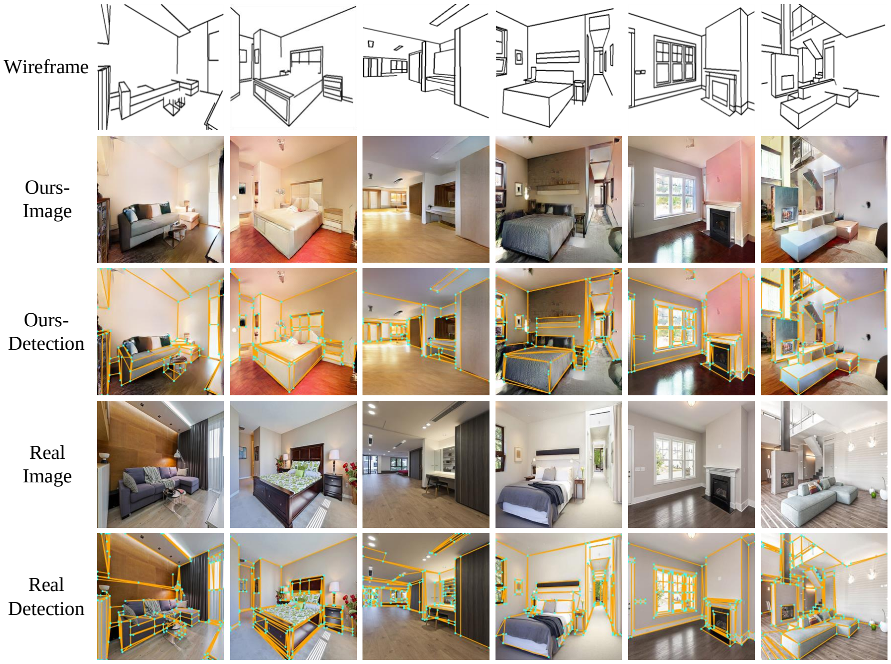

Fig. 3. Additional wireframe-to-image translation results. The first row is the input wireframe; second and third rows are images generated by our model and the corresponding wireframe detection results; the rest are real images and the corresponding detection results. We use wireframe parser from [5] to detect wireframes from synthetic/real images. For fair comparison, no post-processing is done for the wireframe parser.

References

1. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: ICCV. pp. 1511–1520 (2017)
2. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: ICLR (2018)
3. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: CVPR. pp. 8798–8807 (2018)
4. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.: Stackgan++: Realistic image synthesis with stacked generative adversarial networks. arXiv preprint arXiv:1710.10916 (2017)
5. Zhou, Y., Qi, H., Ma, Y.: End-to-end wireframe parsing. In: ICCV 2019 (2019)