Learning Structure-Appearance Joint Embedding for Indoor Scene Image Synthesis

Yuan Xue, Zihan Zhou, and Xiaolei Huang
The Pennsylvania State University

Abstract

Advanced image synthesis methods can generate photo-realistic images for faces, birds, bedrooms, and more. However, these methods do not explicitly model and preserve essential structural constraints such as junctions, parallel lines, and planar surfaces. In this paper, we study the problem of structured indoor image generation for design applications. We utilize a small-scale dataset that contains both images of various indoor scenes and their corresponding ground-truth wireframe annotations. While existing image synthesis models trained on the dataset are insufficient in preserving structural integrity, we propose a novel model based on a structure-appearance joint embedding learned from both images and wireframes. In our model, structural constraints are explicitly enforced by learning a joint embedding in a shared encoder network that must support the generation of both images and wireframes. We demonstrate the effectiveness of the joint embedding learning scheme on the indoor scene wireframe to image translation task. While wireframes as input contain less semantic information than inputs of other traditional image translation tasks, our model can generate high fidelity indoor scene renderings that match well with input wireframes. Experiments on a wireframe-scene dataset [13] show that our proposed translation model significantly outperforms existing state-of-the-art methods in both visual quality and structural integrity of generated images.

1. Introduction

Recently, driven by the success of generative adversarial networks (GANs) [8] and image translation techniques [16, 57], there has been a growing interest in developing data-driven methods for a variety of image synthesis applications, such as image style transfer [17, 19], super-resolution [22], enhancement [52], text-to-image generation [53], domain adaption [12, 34], just to name a few. In this work, we study a new image synthesis task, dubbed wireframe-to-image translation, in which the goal is to convert a line-drawing (i.e., a wireframe) of a man-made environment to a photo-realistic rendering of the scene (Figure 1). In the fields of visual arts, architecture, and computer-aided design, the wireframe representation is an important intermediate step for producing novel designs of man-made environments. For designers to quickly validate their wireframe design and obtain feedback from customers, there is a great demand for conversion of such a wireframe into a photo-realistic rendering of the scene in real-time.

In this paper, we address the need for wireframe to image translation. While several methods have been developed recently to parse wireframes from images [13, 50, 55, 56], few work has studied how to translate wireframes to photo-realistic images. Compared to edge maps and sketches, wireframes contain precise information that encodes 3D geometric structure such as salient straight lines and junctions while being more sparse and ignoring lines due to planar
texture. As such, an image generated given a wireframe input should respect the geometry constraints encoded in it, and should have pixel-level correspondence around straight lines and junctions where lines intersect. This requirement arises from the fact that human perception of 3D is highly dependent on recognizing structures like those encoded in a wireframe; even small violations of geometric constraints would make a generated image look unnatural.

State-of-the-art image translation models such as pix2pix [16] have difficulty in generating images that preserve structures such as straight lines and their intersections (Figure 1). This may be due to that these models are designed for other types of input modalities, as illustrated in Figure 2. Inputs that are semantic segmentation maps emphasize object instance- or part-level correspondence rather than pixel-level correspondence; scribbles in free-hand sketches usually do not strictly map to lines or curves in photographic images; and edges often do not contain complete and accurate structure information and make no distinction between salient structural lines and planar texture-induced lines.

In this work, we propose a structure-appearance joint embedding learning scheme that utilizes a paired wireframe-image dataset to learn to generate images with structural integrity. Our assumption is that there exists a shared latent space for encoding both structural and appearance constraints of a scene. Accordingly, we design our wireframe-to-image translation model to include one encoder and two decoders (see Figure 3). The encoder encodes the input wireframe to a joint embedding, the wireframe decoder reconstructs the original wireframe from the joint embedding, and the scene decoder transforms the joint embedding into a photo-realistic indoor image. Further, the jointly generated wireframe-image pairs are used to train a cGAN-like discriminator, which takes the generated pairs as fake samples and ground truth wireframe-image pairs as real samples. Such a design enables us to better preserve structural integrity and pixel-level correspondences in two ways. First, the encoder together with the wireframe decoder branch can be regarded as an autoencoder for wireframes, which helps enforce precise pixel-level correspondences for salient structures. Second, the cGAN-like discriminator provides an adversarial loss that can help train the model to adaptively learn the difference between the re-constructed wireframe-image pairs and ground truth pairs.

We demonstrate the effectiveness of our proposed model by conducting extensive experiments on a dataset of man-made scenes including various indoor scenes with ground truth wireframe annotations [13]. As shown in Figure 1 and results in Section 4, by introducing a new joint embedding learning scheme combined with both adversarial loss and perceptual loss, our proposed wireframe-to-image translation model generates images that not only have higher visual realism than prior arts [1, 16, 46], but also adhere much better to structural constraints encoded in the input wireframes.

To summarize, the main contributions of our work are:

- We propose a supervised image to image translation model which generates realistic image renderings from wireframe inputs. The architecture including a novel structure-appearance joint embedding and multiple loss functions for the end-to-end network are carefully designed to ensure that the generated synthetic images adhere to wireframe structural constraints.
- To the best of our knowledge, we conduct wireframe-to-image translation experiments for high-fidelity indoor scene rendering using a challenging indoor scene wireframe dataset for the first time. Both quantitative and qualitative results of our experiments indicate the superiority of our proposed method compared with previous state-of-the-art methods.

2. Related Work

2.1. Generative Adversarial Networks

Generative adversarial networks (GANs) [8], especially the conditional GANs [32], have been widely used in image synthesis applications such as text-to-image generation [53] and image-to-image translation [16, 57]. However, training GANs is known to be difficult and often requires a large training set in order to generate satisfactory results. Some attempts have be made to stabilize the GAN training [9, 31], as well as use coarse-to-fine generation to get better results [18, 53]. One work that explores structure informa-
tion in GAN training is [47]. It utilizes RGB-D data and factorizes the image generation process into synthesis of a surface normal map and then the conditional generation of a corresponding image.

2.2. Supervised Image-to-Image Translation

The line of research that most closely relate to our work is supervised image-to-image translation, in which input-output image pairs are available during training. Prior work [1, 16, 46] has been focusing on leveraging different losses to generate high-quality output images. While pixel-wise losses, such as the ℓ1 loss, are the most natural choices, using ℓ1 loss alone has been shown to generate blurry images [16, 17]. To mitigate the problem, Isola et al. [16] uses a combination of ℓ1 loss and a conditional adversarial loss. To avoid the instability of adversarial training, Chen and Koltun [1] implement a cascaded refinement network trained via feature matching based on a pre-trained visual perception network. Recently, the perceptual loss [6] has been shown to be effective in measuring the perceptual similarity between images [34]. Wang et al. [45] integrates the perceptual adversarial loss and the generative adversarial loss to adaptively learn the discrepancy between the output and ground-truth images. Combining the merits from previous works, Wang et al. [46] generate high quality images with coarse-to-fine generation, multi-scale discriminators, and an improved adversarial loss.

Other works focus on improving the performance for a certain input modality. For semantic maps, Qi et al. [38] first retrieve segments from external memory, then combine the segments to synthesize a realistic image. Liu et al. [26] predict convolutional kernels from semantic labels and use a feature-pyramid semantics-embedding discriminator for better semantic alignment. Park et al. [37] modulate the normalization layer with learned parameters to avoid washing out the semantic information. For sketches, Sangkloy et al. [41] generate realistic images by augmenting the training data with multiple sketch styles; SketchyGAN [2] improves the information flow during training by injecting the input sketch at multiple scales; Lu et al. [28] use sketch as context in a joint image completion framework to handle the misalignment between sketches and photographic objects.

2.3. Joint Representation Learning

For applications that involve two or more variables, the traditional one-way mapping of GANs may be insufficient to guarantee the correspondence between the variables. Conditional GANs [32] such as InfoGAN [3], ACGAN [36], and StarGAN [5] learn to infer one variable from another in both directions. ALI [7], CycleGAN [57], and their variants (e.g., [24, 20, 51]) learn the cross-domain joint distribution matching via bidirectional mapping of two examples.

In unsupervised image-to-image translation, several works [14, 25, 23] propose to map images from multiple domains to a joint latent space. To further learn instance level correspondences, DA-GAN [29] incorporate a consistency loss in the latent space between the input and output images. However, due to the lack of paired training data, it is hard for these methods to generate outputs that match all the details (e.g., semantic parts) in the input images. When paired data is available, learning a joint embedding has been proved to be an effective way to capture the correspondences. To promote instance awareness in unsupervised image translation, InstaGAN [33] simultaneously translates image and the corresponding segmentation mask. Recent work on domain adaption [4] jointly predict segmentation and depth maps in order to better align the predictions of the task network for two domains.

3. Methodology

The core idea of our work is to add an intermediate step in the image synthesis process to improve structural integrity and pixel-level correspondence. Specifically, we learn a structure-appearance joint embedding from the input wireframe, and use the joint embedding to simultaneously generate corresponding scene images and reconstructed wireframes as output. As shown in Figure 3. The overall pipeline of our image translation model consists of an encoder, a wireframe decoder, a scene image decoder, and a discriminator.

In the following, we introduce the theoretical background and architecture of our proposed model in Section 3.1, and discuss implementation details in Section 3.2.

3.1. Learning Joint Embedding for Wireframe-to-Image Translation

Formally, we measure the uncertainty of generating the correct wireframe from a joint embedding of wireframe and scene image using Conditional Entropy. The conditional entropy of an input wireframe x conditioned on its corresponding joint embedding e is defined as

\[
H(x|e) = \mathbb{E}_{x \sim P(x|e)} \log P(x|e),
\]

(1)

where e \sim Q(x, y) follows an estimated joint distribution Q of wireframe x and indoor scene image y, and is computed by an encoder network Enc. Under a supervised training scenario with paired wireframe and scene image, for simplicity, we assume that the mapping from x to e is deterministic so that e = Enc(x) is a joint embedding of x and y. Since the mapping from e to x should also be deterministic when e contains a certain input, we have

\[
H(x|e = Enc(x)) = 0.
\]

Since we do not have the ground truth distribution of P(x|e), we approximate it with a decoder network Decw.
Figure 3. Network architecture of our wireframe-to-image translation model. The numbers above each block indicate the kernel size and the output channel number. Best viewed in color.

for reconstructing the wireframe from the joint embedding. The conditional entropy of $\text{Dec}_w$ is

$$L_{ce} = \mathbb{E}_{\hat{x} \sim P(x|e)} \log \text{Dec}_w(\hat{x}|e)$$

$$= H(x|e) + KL(P(x|e)||\text{Dec}_w(\hat{x}|e)) \quad (2)$$

$$\geq H(x|e) = 0.$$ 

Thus, minimizing the conditional entropy is equivalent to reducing the KL divergence between the decoder and the ground truth posterior. To approximate the $L_{ce}$, given a mini-batch of $N$ wireframes $x_n$, we define the wireframe reconstruction objective as

$$\min_{\theta, \theta_w} \mathcal{L}_{rec} = \frac{1}{N} \sum_{n=1}^{N} \alpha_w \| x_n - \text{Dec}_w(\text{Enc}(x_n)) \|_1$$

$$+ \beta_w \text{MS-SSIM}(x_n, \text{Dec}_w(\text{Enc}(x_n))) \quad (3)$$

where $\theta, \theta_w$ are the parameters of encoder and wireframe decoder, respectively. The first term is the $L_1$ distance between the original wireframe and the reconstructed wireframe. The second term is the Multiscale Structural Similarity (MS-SSIM) loss to compensate the $L_1$ distance, as MS-SSIM is more perceptually preferable. More details of MS-SSIM can be found in [49]. $\alpha_w$ and $\beta_w$ are scaling factors to balance the two loss terms.

In addition to the decoder branch that reconstructs the wireframe, we have another decoder $\text{Dec}_s$ that generates the corresponding scene image from the learned joint embedding. By having the two decoder branches share the same encoder, the encoder network is forced to learn both structure and appearance information as well as their correspondence so that the generated image can have better structural alignment with the reconstructed wireframe. Given a mini-batch of $N$ wireframes $x_n$ and corresponding scene images $y_n$, we define the objective for scene generation as

$$\min_{\theta, \theta_s} \mathcal{L}_{gen} = \frac{1}{N} \sum_{n=1}^{N} \left( \alpha_s \| y_n - \text{Dec}_s(\text{Enc}(x_n)) \|_1 \right.$$  

$$\left. + \beta_s \text{MS-SSIM}(y_n, \text{Dec}_s(\text{Enc}(x_n))) \right), \quad (4)$$

where the scene decoder network is parameterized by $\theta_s$. The perceptual loss $D_{perc}$ is defined as

$$D_{perc}(y, \hat{y}) = \sum_l \frac{1}{H_l(W_l)} \| \phi_l(y) - \phi_l(\hat{y}) \|_2^2, \quad (5)$$

where $\phi_l$ is the activations of the $l$th layer of a perceptual network with shape $C_l \times H_l \times W_l$. In our experiments, we use the 5 convolutional layers from VGG16 [43] pre-trained on ImageNet [40] to extract visual features, and unit-normalize the activations in the channel dimension as in [54].

Further, we propose an adversarial loss [8] to adaptively learn the difference between the reconstructed wireframe/generated image and the groundtruth. Denote $\hat{x}$ and $\hat{y}$ as the reconstructed wireframe and generated scene image, the adversarial objective is

$$\max_{\theta_d} \min_{\theta, \theta_w, \theta_s} \mathcal{L}_{adv}$$

$$= \mathbb{E}_{x,y} \log \sigma(\text{Dis}(x, y)) + \mathbb{E}_{x,y} \log(1 - \sigma(\text{Dis}(\hat{x}, \hat{y}))), \quad (6)$$

where $\sigma(\cdot)$ is the sigmoid function and $\theta_d$ represents the parameters of the conditional discriminator network, Dis. For simplicity, we omit the representations such as $x \sim P_x$ in all adversarial objectives.

Therefore, the full objective for end-to-end training of our translation model is

$$\max_{\theta_d} \min_{\theta, \theta_w, \theta_s} \mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{gen} + \lambda \mathcal{L}_{adv}, \quad (7)$$

where $\lambda$ is another scaling factor to control the impact of the adversarial loss.
3.2. Implementation Details

In our image translation model, the encoder network consists of 5 convolution blocks. The first block uses $7 \times 7$ convolution kernels with stride 1 and reflection padding 3. The remaining 4 downsample blocks have kernel size 3, stride 2 and reflection padding 1. Each convolutional layer is followed by one batch normalization [15] layer and one LeakyReLU [30] activation. The last downsample block is followed by 4 residual blocks [10] with $3 \times 3$ convolution and ReLU activation.

The decoder network consists of 4 upsample blocks. To avoid the characteristic artifacts introduced by the transpose convolution [35], each upsample block contains one $3 \times 3$ sub-pixel convolution [42] followed by batch normalization and ReLU activation. The last block uses a $7 \times 7$ convolution directly followed by a tanh activation without normalization. The two decoder networks have similar architecture except in the last layer where the outputs have different channel dimensions.

We follow [16] and use the PatchGAN [27] discriminator for adversarial training. We use LSGAN [31] for stabilizing the adversarial training. The scaling factors in our final model are $\alpha_w = 1, \beta_w = 1, \alpha_s = 15, \beta_s = 4$ and $\lambda = 1$. These values are determined through multiple runs of experiments. The training is done using Adam optimizer [21] with initial learning rate $2e^{-3}$. The learning rate is decayed every 30 epochs with rate 0.5. The batch size is 16 and the maximum number of training epochs is 500.

All training images are first resized to $307 \times 307$, then randomly cropped to $256 \times 256$. A random horizontal flipping and random adjustment of brightness, contrast and saturation are applied for data augmentation. During inference, all images are re-scaled to $256 \times 256$ with no further processing.

4. Experiments
4.1. Experiment Settings

Dataset. The wireframe dataset [13] consists of 5,462 images of man-made environments, including both indoor and outdoor scenes, and manually annotated wireframes. Each wireframe is represented by a set of junctions, a set of line segments, and the relationships among them. Note that, unlike general line segments, the wireframe annotations consider structural elements of the scene only. Specifically, line segments associated with the scene structure are included, whereas line segments associated with texture (e.g., carpet), irregular or curved objects (e.g., humans and sofa), and shadows are ignored. Thus, to translate the wireframe into a realistic image, it is critical for a method to handle incomplete information about scene semantics and objects.

As we focus on the indoor scene image generation task in this paper, we filter out all outdoor or irrelevant images in the dataset. This results in 4,511 training images and 422 test images. The dataset contains various indoor scenes such as bedroom, living room, and kitchen. It also contains objects such as humans which are irrelevant to our task. The limited size and the scene diversity of the dataset make the task of generating interior design images even more challenging.

Baselines. We compare our image translation models with several state-of-the-art models, namely the Cascaded Refinement Network (CRN) [1], pix2pix [16], and pix2pixHD [46]. For fair comparison, we use authors’ original implementations whenever possible. For CRN, we use six refine modules, staring from $8 \times 8$ all the way up to $256 \times 256$. For pix2pix model, we use UNet [39] backbone model as in the original paper. We decrease the weight of pixel loss from 100 to 50 since the original weight fails to generate any meaningful results. For pix2pixHD model, we use two discriminators with different scales. Since there is no instance map available for our problem, we train the pix2pixHD model with wireframes only.

Besides, to verify the benefit of joint embedding, we also train a variant of our method in which we remove the wireframe decoder branch. All the other components in the network are the same as our full model and we train it with the same image generation loss and adversarial loss for wireframe-to-image translation. For all baseline models involve adversarial training, since there is no wireframe predicted, generated images are paired with their input wireframes as the input to the discriminator.

4.2. Qualitative Comparisons

The qualitative comparisons for the translation models are shown in Figure 4. We first note that the CRN model trained on the wireframe dataset fails to generate meaningful results, despite that we have experimented with different hyper-parameter settings. One possible reason is that the CRN is originally designed for image synthesis based on semantic layouts. However, the wireframe itself contains little semantic information (e.g., object categories), thus the model has to infer such information from the structure information presented in the wireframe. Moreover, CRN model is the only model which does not use adversarial training. This may suggest that adversarial training is important in the wireframe-to-image translation task.

Except for the CRN, all other models are able to generate meaningful synthetic images. However, in the images generated by pix2pix and pix2pixHD, structural integrity is not always well preserved. In general, the generated images of these models cannot align well with the input wireframes, especially when structure information is complicated (e.g., the furniture areas in the first and second rows of Figure 4). Further, these methods generate noticeable artifacts in regions where structure information is sparse (e.g., the white
walls in the third row of Figure 4). Compared with the previous models, our model without joint embedding generates more semantically meaningful images. But it still fails to capture the structure information and the correspondence to the input wireframe in some cases (e.g., blurry edges and distorted walls in the third and fourth rows of Figure 4).

In contrast, our full model generates images with best quality among all models and preserves the structure and correspondence very well. Compared with the real images in the test set, the synthetic images of our final model are almost photo-realistic.

4.3. Quantitative Evaluations

**FID, LPIPS, and SSIM scores.** We first report results based on various standard metrics for image synthesis.

Fréchet inception distance (FID) [11] is a popular evaluation metric for image synthesis tasks, especially for GAN models. It computes the divergence between the synthetic data distribution and the real data distribution:

\[
\text{FID} = \left\| \tilde{m} - m \right\|^2_2 + \text{Tr}(\tilde{C} + C - 2(\tilde{C}C)^{1/2}),
\]

where \(m, C\) and \(\tilde{m}, \tilde{C}\) represent the mean and covariance of the feature embeddings of the real and the synthetic distributions, respectively. The feature embedding is extracted from an Inception-v3 [44] model pre-trained on ImageNet. Since the dataset contains various indoor scenes, we use the pre-trained model without fine-tuning. Lower FID score indicates a better generation result.

For our task, since we have the groundtruth images associated with the input wireframes, we also calculate paired LPIPS and SSIM scores between the synthetic images and the real images. The learned perceptual image patch similarity (LPIPS) recently proposed by Zhang et al. [54] is essentially a perceptual loss. It has been shown to have a better agreement with human perception than traditional
perceptual metrics such as SSIM [48] and PSNR. We use Eq. (5) to calculate the perceptual distance between the synthetic image and the real image. Note that in our experiments we calculate the perceptual distance instead of the similarity, thus the lower the LPIPS score, the better quality of the generated images. The feature extractor is a pre-trained VGG16 model as in the training.

In Table 1, we report results of all methods except for CRN, since CRN fails to generate meaningful results. As one can see, pix2pixHD outperforms pix2pix in all metrics. Compared with the pix2pix, pix2pixHD adopts multi-scale discriminators and use the adversarial perceptual loss, leading to better performance in the image translation task. However, since the training dataset in our experiments has a limited size, a perceptual loss learned by adversarial training may not work as well as a perceptual loss computed by a pre-trained feature extractor. As shown in Table 1, our model without the joint embedding learning achieves better performance than the pix2pixHD model.

Finally, our full model with the joint embedding learning achieves the best performance across all metrics, as the images generated by the model better preserve the structure information encoded in the wireframes.

**Wireframe detection score.** Since the focus of this work is to preserve structure information in the wireframe-to-image translation task, an important and more meaningful evaluation metric would be whether we can infer correct wireframes from the generated images or not.

To this end, we propose a wireframe detection score as a complimentary metric for evaluating the structural integrity in image translation systems. Specifically, we apply the state-of-the-art wireframe parser [55] to detect wireframes from the generated images. The wireframe parser outputs a vectorized wireframe that contains semantically meaningful and geometrically salient junctions and lines (Figure 5). To evaluate the wireframe detection results, we follow [55] and use the *structural average precision* (sAP), which is defined as the area under the precision-recall curve computed from a scored list of detected line segments on all test images. Here, a detected line is considered as a true positive if the distance between the predicted and ground truth end points is within a threshold $\theta$.

Table 2 reports the sAP scores at $\theta = \{5, 10, 15\}$. As one can see from Table 2 and Figure 5, our full model outperforms all other methods by large margins. In the last row of Table 2, we also report sAP scores obtained by applying the same wireframe parser [55] to the corresponding real images. Rather surprisingly, the images generated by our method even achieve higher sAP scores than the real images. After a close inspection of the results, we find that it is mainly because, when labeling wireframe, human annotators tend to miss some salient lines and junctions in the real images. In other words, there are often more salient lines and junctions in real images than those labelled in the ground truth. As a result, the detected wireframes from real images contain more false positives. In the meantime, the input provided to our model is just the annotated wireframes. And our model is able to faithfully preserve such information in the generated images.

**Human studies.** We also conduct a human perception evaluation to compare the quality of generated images between our method and pix2pixHD [46]. we show the ground truth wireframes paired with images generated by our method

| Method         | FID $\downarrow$ | LPIPS $\downarrow$ | SSIM $\uparrow$ |
|----------------|------------------|-------------------|-----------------|
| pix2pix [16]   | 186.91           | 3.34              | 0.091           |
| pix2pixHD [46] | 153.36           | 3.25              | 0.080           |
| Ours w/o JE    | 97.49            | 2.85              | 0.092           |
| Ours           | 70.73            | 2.77              | 0.102           |

Table 1. Quantitative evaluation of image translation models on the wireframe-to-image translation task. For SSIM, the higher the better; For FID and LPIPS, the lower the better.

| Method         | sAP $^5$ | sAP $^{10}$ | sAP $^{15}$ |
|----------------|---------|-------------|-------------|
| pix2pix [16]   | 7.8     | 10.0        | 11.1        |
| pix2pixHD [46] | 10.6    | 13.6        | 15.1        |
| Ours w/o JE    | 26.7    | 34.4        | 37.5        |
| Ours           | 60.1    | 64.1        | 65.7        |
| Real images    | 58.9    | 62.9        | 64.7        |

Table 2. Wireframe parser scores using [55] for all models. For sAP scores, the higher the better.

![Figure 5. Example wireframe detection results on the synthesized images.](image-url)
and pix2pixHD to three workers. Workers are asked to evaluate synthetic images based on fidelity and the alignment to wireframes. Workers are given unlimited time to choose between our method, pix2pixHD, or “none” if both methods fail to generate realistic enough image or preserve the wireframe structure. We use all 422 test images for this evaluation. On average, the preference rates of pix2pixHD, our method, and “none” are 3.7%, 65.1%, 31.2%, respectively. The human study score further proves that our method can not only generate realistic rendering, but also respect the structure information encoded in the wireframes.

4.4. Wireframe Manipulation Results

To provide additional insight to our trained models, and also to illustrate the potential use of our method in a realistic design application setting, we incrementally modify the input wireframes and check whether the generated scene images are updated consistently. In the first row of Figure 6, we show the original input wireframe and the images generated by pix2pixHD and our method. In the second and third rows, we manually remove and add some lines/junctions to the original wireframe. These components may correspond to objects like a heater and a decorative picture on the right wall. As shown in Figure 6, our model captures the changes in the input wireframe and update the generated image in a consistent fashion. On the contrary, pix2pixHD responses poorly to such changes.

4.5. Discussion

Failure cases. While our model is able to capture the structure information and ensures that generated images match well with input wireframes, the wireframes can be sparse and contain little semantic information such as objects. As shown in Figure 7, when there is no wireframe information provided to the model, especially in the corner part of the image, our model sometimes fail to generate visually meaningful results. We expect to mitigate this issue by training on a larger dataset containing more diverse images and wireframes, and providing other semantic inputs such as types of furniture to be included to make the learning task easier.

Extensions. Our joint embedding learning framework is general and may also benefit other image synthesis tasks. In fact, we have conducted preliminary experiments on the noise-to-image generation task, in which our model is trained to simultaneously generate paired scene image and wireframe from a noise input using the joint embedding. We have obtained improved results compared to a baseline which generates the scene image only. More details are provided in the supplementary material.

5. Conclusion

In this paper, we study a new image synthesis task for design applications in which the input is a wireframe representation of the scene. By learning the joint embedding in the shared latent space of wireframes and images, our wireframe-to-image translation model generates photorealistic scene images with high structural integrity. In the future, we plan to extend our model to a wider range of applications and considering semantic constraints alongside with structural constraints. One such application is to generate image rendering directly from computer-aided design (CAD) models.

Acknowledgement. This work is supported in part by NSF Award #1815491.
References

[1] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In ICCV, pages 1511–1520, 2017. 1, 2, 3, 5, 7

[2] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In CVPR, pages 9416–9425, 2018. 3

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, pages 2172–2180, 2016. 3

[4] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In CVPR, pages 1841–1850, 2019. 3

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sung hun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In CVPR, pages 8789–8797, 2018. 3

[6] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In NIPS, pages 658–666, 2016. 3

[7] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. In ICLR, 2017. 3

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NIPS, pages 2672–2680, 2014. 1, 2, 4

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In NIPS, pages 5767–5777, 2017. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016. 5

[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 NIPS, pages 6626–6637, 2017. 6

[12] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In ICMLO, pages 1994–2003, 2018. 1

[13] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. Learning to parse wireframes in images of man-made environments. In CVPR, pages 626–635, 2018. 1, 2, 5

[14] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In ECCV, pages 172–189, 2018. 3

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, pages 448–456, 2015. 5

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In CVPR, pages 1125–1134, 2017. 1, 2, 3, 5, 7

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, pages 694–711. Springer, 2016. 1, 3

[18] Tero Karras, Samuli Laine, and Janne Hellinen. Progressing growing of gans for improved quality, stability, and variation. In ICLR, 2018. 2, 11

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In CVPR, pages 4401–4410, 2019. 1

[20] Taeksoo Kim, Moonsu Cha, Hyunsu Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In ICML, pages 1857–1865. JMLR, org, 2017. 3

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015. 5

[22] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, pages 4681–4690, 2017. 1

[23] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In ECCV, pages 36–52, 2018. 3

[24] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In NIPS, pages 5495–5503, 2017. 3

[25] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In NIPS, pages 700–708, 2017. 3

[26] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, and Hongsheng Li. Learning to predict layout-to-image convolutional networks for semantic segmentation. In CVPR, pages 3431–3440, 2015. 3

[27] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. Image generation from sketch constraint using conditional GAN. In ECCV, pages 213–228, 2018. 2, 3

[28] Shuang Ma, Jianlong Fu, Chang Wen Chen, and Tao Mei. LSGAN: Least squares generative adversarial networks. In CVPR, pages 5657–5666, 2018. 3

[29] Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, pages 4681–4690, 2017. 1

[30] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. A style-based generator architecture for generative adversarial networks. In ICML, pages 1841–1850, 2019. 1

[31] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014. 2, 3
[33] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. In ICLR, 2019. 3
[34] Zak Murez, Soheil Kolouri, David J. Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In CVPR, pages 4500–4509, 2018. 1
[35] Augustus Odena, Vincent Dumoulin, and Chris Olah. De-convolution and checkerboard artifacts. Distill, 1(10):e3, 2016. 5
[36] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In ICML, pages 2642–2651. JMLR. org, 2017. 3
[37] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In CVPR, pages 2337–2346, 2019. 3
[38] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In CVPR, pages 8808–8816, 2018. 3
[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015. 5
[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, et al. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3):211–252, 2015. 4
[41] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In CVPR, pages 6836–6845, 2017. 2, 3
[42] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In CVPR, pages 1874–1883, 2016. 5
[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 4
[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In CVPR, pages 2818–2826, 2016. 6
[45] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. Perceptual adversarial networks for image-to-image transformation. IEEE Transactions on Image Processing, 27(8):4066–4079, 2018. 3
[46] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In CVPR, pages 8798–8807, 2018. 2, 3, 5, 7, 11
[47] Xiaolong Wang and Abhinav Gupta. Generative image modeling using style and structure adversarial networks. In ECCV, pages 318–335. Springer, 2016. 3
[48] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004. 7
[49] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, volume 2, pages 1398–1402. Ieee, 2003. 4
[50] Nan Xue, Song Bai, Fudong Wang, Gui-Song Xia, Tianfu Wu, and Liangpei Zhang. Learning attraction field representation for robust line segment detection. In CVPR, pages 1595–1603, 2019. 1
[51] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In ICCV, pages 2849–2857, 2017. 3
[52] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. arXiv preprint arXiv:1701.05957, 2017. 1
[53] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stacgan++: Realistic image synthesis with stacked generative adversarial networks. arXiv preprint arXiv:1710.10916, 2017. 1, 2, 11, 12
[54] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In CVPR, pages 586–595, 2018. 3, 4, 6
[55] Yichao Zhou, Haozhi Qi, and Yi Ma. End-to-end wireframe parsing. In ICCV 2019, 2019. 1, 7, 13
[56] Yichao Zhou, Haozi Qi, Simon Zhai, Qi Sun, Zhili Chen, Li-Yi Wei, and Yi Ma. Learning to reconstruct 3d manhattan wireframes from a single image. In ICCV, 2019. 1
[57] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV, pages 2223–2232, 2017. 1, 2, 3
A. Joint Wireframe and Image Synthesis

Our joint embedding 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, 18, 46, 53] 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 [53] 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 embedding learning can be found in Figure 8. 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 G₀, then two separate branches of coarse-to-fine generators take the joint embedding and generate wireframes and images, respectively. Although we do not have explicit supervision upon the joint embedding, the wireframe-based adversarial learning guarantees that the learned embedding 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 [53], 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 i-th generator Gᵢ and the i-th discriminator Dᵢ is defined as

$$\max_{\theta_D} \min_{\theta_G} L_{\text{adv}}^{i}$$

$$= E_{x_{n}}[\log D_{n}^{i}(x_{n})] + E_{z_{n}}[\log(1 - D_{n}^{i}(G_{n}(z_{n})))],$$

$$+ E_{y_{n}}[\log D_{n}^{i}(y_{n})] + E_{z_{n}}[\log(1 - D_{n}^{i}(G_{n}(z_{n})))],$$

where $z_{n}$ is the input to the i-th 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.

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

$$L_{\text{con}}^{i} = \frac{1}{N} \sum_{n=1}^{N} \left( \lambda_{1} \mu_{\hat{x}_{n}^{i}} - \mu_{x_{n}} \right)^{2} + \lambda_{2} \Sigma_{\hat{x}_{n}^{i}} - \Sigma_{x_{n}} \right)^{2}$$

$$+ \left( \lambda_{1} \mu_{\hat{y}_{n}^{i}} - \mu_{y_{n}} \right)^{2} + \lambda_{2} \Sigma_{\hat{y}_{n}^{i}} - \Sigma_{y_{n}} \right)^{2},$$

where $\mu$ and $\Sigma$ 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 $L_{\text{adv}}^{i}$ is applied. When we train the i-th generator, the total loss is the sum of the adversarial loss and the consistency regularization loss, i.e. $L_{\text{total}}^{i} = L_{\text{adv}}^{i} + \alpha_{\text{con}} L_{\text{con}}^{i}$, where $\alpha_{\text{con}}$ is the scaling factor that controls the relative influence of the two loss terms.

A.1. Implementation Details

Our proposed GAN model is built upon a StackGANv2 [53] backbone. After the shared generator G₀ which maps the input vector $z$ to a joint embedding, the wireframe generator and image generator use separate coarse-to-fine generators $G_{w}^{1}, G_{1}^{s}$ and $G_{w}^{2}, G_{2}^{s}$ 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 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...
Figure 9. Qualitative comparisons of image generation models. The first row contains generated images by the baseline StackGANv2 [53] model. The second and third rows are paired images and wireframes generated by our model.

| Method          | IS↑  | FID↓ |
|-----------------|------|------|
| StackGANv2 [53] | 2.92 | 49.76|
| Ours            | 3.08 | 50.96|
| GT              | 3.21 | -    |

Table 3. Quantitative comparisons between image generation models. The inception score of the real images in the test set is provided as reference.

A.2. Experiment Results

Figure 9 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 embedding learning module.

Table 3 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.

B. Additional Image Translation Results

We provide more wireframe-to-image translation results and wireframe detection results in Figure 10. 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 achieve higher sAP scores than real images.
Figure 10. Additional wireframe-to-image translation results. The first row is the input wireframe; second and third row are results generated by our model; the rest are real images and real wireframe deduction results. We use wireframe parser from [55] to detect wireframes from synthetic/real images. For fair comparison, no post-processing is done for the wireframe parser.