STATE: Learning Structure and Texture Representations for Novel View Synthesis

Xinyi Jing[†]
Tianjin University
Tianjin, China
jingxinyi@tju.edu.cn

Qiao Feng[†]
Tianjin University
Tianjin, China
exculibar@tju.edu.cn

Yu-Kun Lai
Cardiff University
Wales, UK
LaiY4@cardiff.ac.uk

Jinsong Zhang
Tianjin University
Tianjin, China
jinszhang@tju.edu.cn

Yuanqiang Yu
Tianjin University
Tianjin, China
yuyuanqiang@tju.edu.cn

Kun Li[*]
Tianjin University
Tianjin, China
lik@tju.edu.cn

Abstract

Novel view synthesis, especially from sparse view images, is very challenging due to large view shifting and occlusions. Existing image-based methods fail to generate reasonable results for invisible regions, while geometry-based methods have difficulties synthesizing detailed textures. In this paper, we propose STATE, an end-to-end deep neural network, for sparse view synthesis by learning STructure And TExture representations. The structure is encoded as a hybrid feature field to predict reasonable structures for invisible regions and maintain original structures for visible regions, and the texture is encoded as a deformed feature map to preserve detailed textures. We propose a hierarchical fusion scheme with intra-branch and inter-branch aggregation, in which spatio-view attention is designed for multi-view fusion at the feature level to adaptively select important information by regressing pixel-wise or voxel-wise confidence maps. Through decoding the aggregated features, STATE is able to generate realistic images with reasonable structures and detailed textures. Experimental results demonstrate that our method achieves better performance than state-of-the-art methods in both qualitative and quantitative evaluations. Our method also enables texture and structure editing applications benefitting from implicit disentanglement of structures and textures. The code will be available online at https://github.com/jingxinyi/STATE.

1. Introduction

Novel view synthesis aims to generate a new image for an object at a new viewpoint from a single image or multi-view images, which has a wide range of applications in virtual reality, education and movie production. It is a very challenging problem for sparse view cases due to large view variation and occlusions.

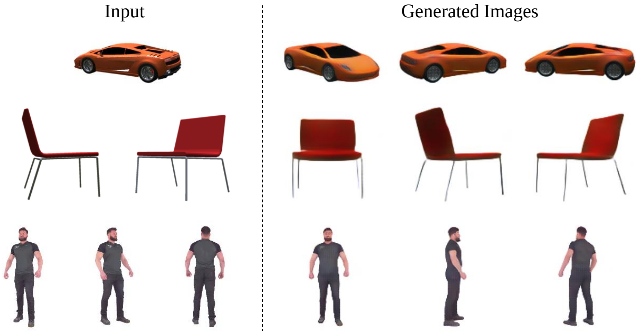

Figure 1. Our STATE model is able to generate realistic images from sparse view images or even a single image.

Existing methods on novel view synthesis can be classified into image-based and geometry-based methods. Image-based methods warp the source image from the source viewpoint to the target viewpoint by estimating an affine transformation [39, 45] or an appearance flow field [32, 38, 53]. Flow-based methods are more flexible to deal with complex deformations than affine transformation methods. However, due to lack of geometry information, image-based methods tend to generate unsatisfactory results for invisible regions, especially for sophisticated objects or sparse views. Geometry-based methods first estimate the 3D structure of the object in an explicit [6, 42, 20] or implicit [37, 29, 49] manner, and then generate the target image by rotation and projection. Explicit representations use discrete volumes while implicit methods use continuous implicit functions. Along with neural rendering based methods [5], the latter can be trained without 3D supervision. Although geometry-

[†] Contributed equally.
[*] Corresponding author.

based methods can keep the consistency of the structure and predict reasonable shapes for the invisible regions, they would deteriorate with sparse views and lose texture details due to the limited representation resolution.

It is very important to find an effective way to make better use of the multi-view information, especially for sparse views. Most works [22, 23, 29, 41, 44] directly average the representations of all inputs, where all locations of inputs are taken as valid values. However, not all locations of inputs have positive impacts on the target image. To solve this problem, Sun *et al*. [38] propose a self-learned confidence method to fuse the resulting images generated by each input at the pixel level. However, this fusion scheme requires large memory and cannot deal with the unavoidable misalignment problem.

The aforementioned methods encounter three challenges to synthesize satisfactory images: 1) the coupling of the shape and the texture in the input images, 2) potential uncertainties in invisible regions, and 3) difficulty to achieve color, texture and shape consistency.

To address these problems, in this paper, we propose an end-to-end deep neural network, STATE, for sparse view synthesis by disentangling the input images into *STructure And TExture representations* to ensure both shape and texture consistency. Although our method does not explicitly control disentanglement, the two branches with proper design achieve effective disentanglement of structures and textures as verified by experimental results (in Section 4.2 and 4.5). In the structure-aware encoder, we represent structure as a hybrid feature field, which can predict reasonable structure for invisible regions. In the texture-aware encoder, we estimate an appearance flow field and warp the source image feature from the source viewpoint to the target viewpoint at the feature level. To make the best use of multi-view images, we also propose *spatio-view attention* aggregation to adaptively fuse multi-view information at the feature level by regressing pixel-wise or voxel-wise confidence maps. The final image is delivered by decoding the aggregated feature of structure-aware representation and texture-aware representation. Our model works well for both single view and multi-view inputs. Experimental results demonstrate that our method achieves better performance than state-of-the-art methods. We also verify our hypothesis by comprehensive ablation studies. Figure 1 gives some examples of our results.

The main contributions are summarized as follows:

- We propose STATE, an end-to-end deep neural network, to disentangle the sparse input images into two embedding neural representations: structure and texture representations, which helps to predict reasonable regions invisible in the source image, while also recovering detailed textures.
- We propose a hierarchical fusion scheme with intra-branch and inter-branch aggregation. Spatio-view attention is designed for multi-view fusion at the feature level to adaptively select important information by regressing pixel-wise or voxel-wise confidence maps.
- Our model can realize texture or structure swapping without training in stages due to effective disentanglement of structures and textures. Our model is easy and robust to train with a hybrid loss including cosine loss to achieve color, texture and shape consistency, leading to state-of-the-art performance.

2. Related work

In this section, we review the existing work on novel view synthesis for objects or humans from a single image or multiple images, which can be classified into image-based and geometry-based novel view synthesis methods. Image-based methods can maintain the appearance consistency by transferring the pixels in the source images to the target image, while geometry-based methods can maintain the structure consistency by reconstructing the 3D structure of the object before rendering the novel view image.

2.1. Image-based Novel View Synthesis

Image-based novel view synthesis methods directly generate pixels or move pixels from the source images to the target image. Tatarchenko *et al*. [39] and Yang *et al*. [45] generate pixels with affine transformation. Instead of learning to synthesize pixels from scratch, Zhou *et al*. [53] prove that the visual appearance of the same instance from different views is highly correlated, and such correlation can be explicitly learned to predict appearance flow [17, 32, 38], *i.e*., 2D coordinate vectors specifying which pixels in the input view can be used to reconstruct the target view. To use features at different scales, Yin *et al*. [48] estimate appearance flows with different resolutions to warp the source view to the target view. According to the appearance flow, bilinear sampling is used to move pixels from the source images to the target image [17, 19, 38, 53]. To avoid the poor gradient propagation of the bilinear sampling, Ren *et al*. [32] propose a content-aware sampling method by adopting a local attention mechanism. As described in [21, 50], most flow-based methods [38, 53] warp the input images pixel-wisely, which prevents the network from generating new content for invisible pixels. Warping the input images at the feature level can solve this problem [14, 17, 32]. There are also some methods synthesizing invisible pixels without warping the input features. Park *et al*. [30] use a completion network to hallucinate the empty parts. In summary, image-based methods can generate detailed textures by moving pixels from the source images to the target image, but the results generated by the above methods lack a

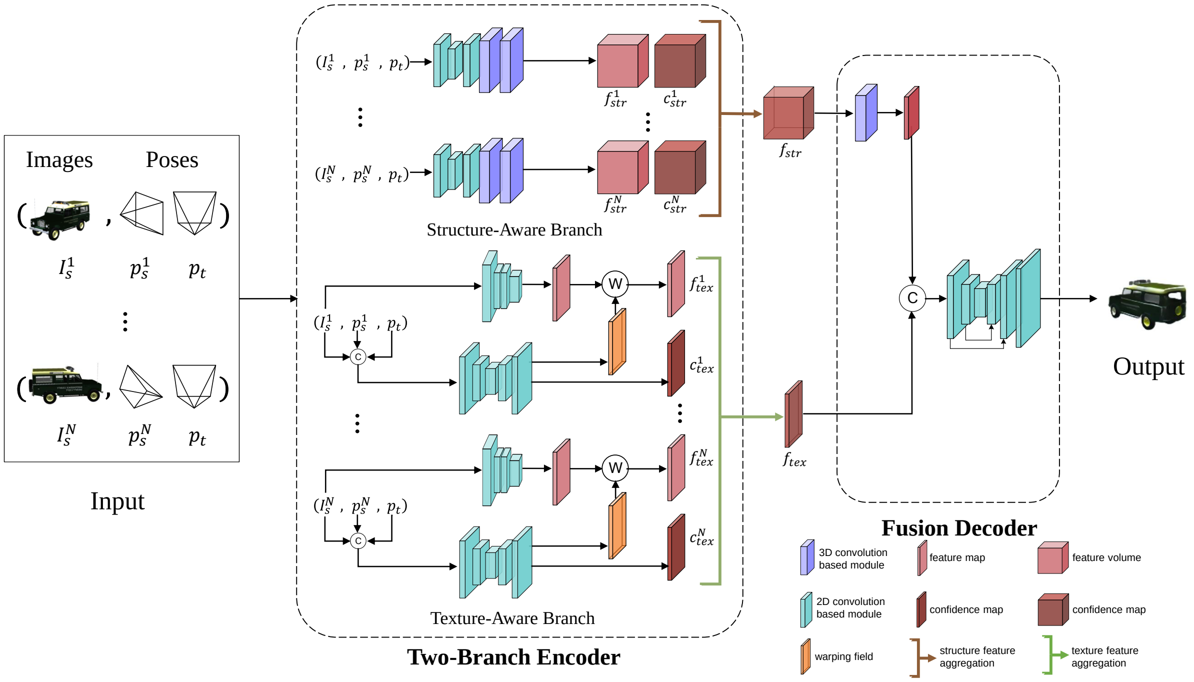

Figure 2. Overview of our STATE model.

consistent shape and hence may have some artifacts along the silhouette.

2.2. Geometry-based Novel View Synthesis

Geometry-based novel view synthesis methods estimate the 3D structure of the instance in an explicit or implicit manner, and then generate the target image by rotation and projection. Two strategies are adopted: depth maps and 3D models (textured occupancy volumes, colored point clouds or neural scene representations). The depth-map-based approaches [2, 6, 13] typically estimate the depth map [46, 47] of each input view as a 2.5D intermediate representation which captures hidden surfaces from one or multiple viewpoints. The point-cloud-based methods [20] estimate a point cloud to be transformed into the target view. In addition, several recent methods [4, 8, 16, 42] reconstruct an explicit occupancy volume from the input images, and render it using traditional rendering techniques. To overcome the memory limitation of volume representations, some methods leverage signed distance field encoded volumes [31, 35] or RGBα-encoded volumes [11, 24] and achieve good performance. Zhao *et al*. [52] adopt parametric human model as the representation together with neural texture to avoid memory limitation problem. Since explicit volumes are discrete, several methods [26, 27, 28, 29] based on implicit volume representations are proposed without any 3D supervision. In order to have a more accurate understanding of the structure of objects, Galama and Mensink [7] propose IterGANs to learn an implicit 3D model of the object in an iterative manner. Implicit volume representation has gained popularity due to continuous shape and texture representation. Some methods [25, 37, 40, 49] predict continuous neural scene representations, and then render them to the novel view image through neural rendering. Geometry-based methods can keep consistent structure and predict reasonable shapes for invisible regions, but the generated textures tend to lose fine details.

In this paper, we propose an end-to-end deep neural network for sparse view synthesis by learning structure and texture representations. Structure is encoded as a hybrid feature field while texture is encoded as a deformed feature map. Each representation is generated by spatio-view attention aggregation for multi-view cases. The results generated by our approach have consistent structures and detailed textures.

3. Method

3.1. Overview

The inputs of novel view synthesis from N images are a target camera pose p_t and N pairs of source images and camera poses $(I_s^1, p_s^1), (I_s^2, p_s^2), ..., (I_s^N, p_s^N)$. Our goal is to synthesize the target image $\hat{I}_t$ in the target camera pose p_t. Denote by I_t and $\hat{I}_t$ the ground truth and synthesized target images. In order to generate the result with reasonable structure and fine texture, we propose a new network STATE that aggregates information from both structure and texture representations. As shown in Figure 2, STATE con-

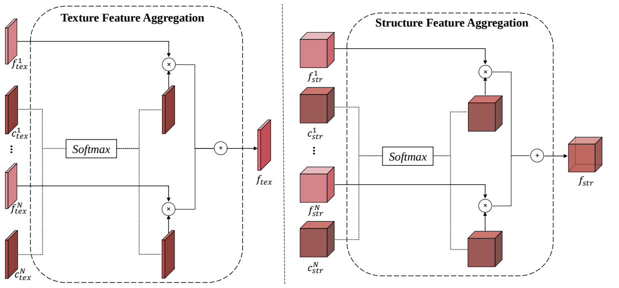

Figure 3. Detailed structures of spatio-view attention aggregation in texture-aware branch (left) and structure-aware branch (right).

sists of a two-branch encoder and a fusion decoder.

1. The two-branch encoder $E(\cdot)$, consisting of a structure-aware branch and a texture-aware branch, encodes the inputs into a structure feature volume f_{str} and a texture feature map f_{tex}. It can be written as:

$$(f_{str}, f_{tex}) = E(p_t, (I_s^1, p_s^1), (I_s^2, p_s^2), ..., (I_s^N, p_s^N)). \tag{1}$$

   The structure-aware branch estimates a hybrid feature field for each view, and then rotates and adaptively aggregates them to a single feature volume f_{str} containing structure information. The texture-aware branch estimates a single feature map f_{tex} containing texture information by adaptively fusing the flow-warped features of N views.

2. The fusion decoder $D(\cdot)$ takes the feature volume f_{str} and the feature map f_{tex} as input and generates the target image by

$$\hat{I}_t = D(f_{str}, f_{tex}). \tag{2}$$

The adaptive fusion of multi-view inputs will be introduced in detail in Section 3.3. Please note that our model is able to extend to an arbitrary number of inputs for both training and testing without modifying the encoder or decoder.

3.2. Two-Branch Encoder

We design a two-branch encoder to disentangle texture and structure from the sparse input images, which includes a texture-aware branch and a structure-aware branch. For both branches, to cope with occlusion and large view difference, pixels in the input images should not have the same contributions. So we design a spatio-view attention by calculating confidence maps for multi-view images to obtain the final texture representation f_{tex} and structure representation f_{str}, which will be presented in Section 3.3 in detail.

In the texture-aware branch, as shown in Figure 2, We use an hourglass network F_{warp} to predict a warping field w_i and a confidence map c_{tex}^i for each input view i, which takes the target pose p_t, the i-th source image I_s^i and the i-th source pose p_s^i as inputs:

$$(w_i, c_{tex}^i) = F_{warp}(p_t, I_s^i, p_s^i), \tag{3}$$

where the warping field w_i is represented by displacements between the source image and the target image. Camera poses p_t and p_s^i are represented by quaternions. We expand the dimensions of the quaternion to match the dimensions of the image, and then concatenate them to form the input. The confidence map c_{tex}^i is used to fuse the feature maps from different views. c_{tex}^i and w_i share all weights of F_{warp} except for their output layers. We use a fully convolutional network F_{tex} to extract features $\tilde{f}_{tex}^i$ from the source images, and then warp the features to get the target features f_{tex}^i, which can be formulated as

$$\tilde{f}_{tex}^i = F_{tex}(I_s^i), \tag{4}$$

$$f_{tex}^i = \mathcal{W}(w_i, \tilde{f}_{tex}^i), \tag{5}$$

where $\mathcal{W}(\cdot)$ is the warping function, and bilinear sampling is used in our network.

In the structure-aware branch, we use an encoder F_{str} [29] consisting of a series of 2D convolutions, reshaping, and 3D convolutions to extract a hybrid feature field represented as a structure feature volume for each image:

$$\tilde{f}_{str}^i = F_{str}(I_s^i), \tag{6}$$

where $\tilde{f}_{str}^i$ is the structure feature volume in the corresponding pose p_s^i. Then, we rotate $\tilde{f}_{str}^i$ from the source pose p_s^i to the target pose p_t:

$$f_{str}^i = \mathcal{R}(\tilde{f}_{str}^i, p_s^i, p_t), \quad c_{str}^i = 3DConv(f_{str}^i), \tag{7}$$

where $\mathcal{R}(\cdot)$ is the rotation operation with trilinear sampling, f_{str}^i is the transformed feature volume having the same shape as $\tilde{f}_{str}^i$, and $3DConv(\cdot)$ represents 3D convolution. The confidence map c_{str}^i is used to fuse the feature maps from different views.

The texture representation f_{tex} and the structure representation f_{str} are decoded by a fusion decoder described in Section 3.4.

3.3. Spatio-View Attention Aggregation

Due to occlusions and large view variation, the texture representation f_{tex}^i of view i may be incomplete. The missing regions should not have the same weighting as the other regions. Moreover, the visible view should have more impact on the final result. Similarly, the structure-aware branch requires different weights for different regions of f_{str}^i and different views. Therefore, instead of simply averaging the encoded feature maps, we design an adaptive aggregation with spatio-view attention for the texture-aware

Table 1. Quantitative comparison with four alternative designs.

Dataset	Method	1 view		2 views		3 views		4 views	
		LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓
Car	w/o Tex.	0.139	79.143	0.104	57.997	0.096	54.261	0.092	52.961
	w/o Str.	0.127	64.788	0.098	44.501	0.089	39.765	0.084	37.901
	w/o SVA	0.118	62.619	0.090	42.023	0.081	38.642	0.078	37.258
	w/o Cos.	0.136	82.208	0.104	57.810	0.096	53.844	0.092	52.462
	Full	**0.117**	**60.387**	**0.089**	**39.052**	**0.080**	**34.472**	**0.075**	**32.290**
Chair	w/o Tex.	0.250	64.584	0.113	21.622	0.096	19.488	0.092	18.898
	w/o Str.	0.166	33.330	0.141	26.628	0.133	25.145	0.129	24.443
	w/o SVA	0.209	48.731	0.100	19.228	0.086	17.336	0.081	16.730
	w/o Cos.	0.246	62.418	0.109	20.006	0.093	17.998	0.088	17.461
	Full	**0.159**	**30.936**	**0.096**	**18.486**	**0.080**	**16.547**	**0.074**	**15.881**
Human	w/o Tex.	0.118	70.431	0.087	64.174	0.082	64.860	0.081	65.550
	w/o Str.	0.106	82.642	0.088	76.567	0.081	75.137	0.078	75.357
	w/o SVA	**0.102**	61.274	0.078	57.386	0.072	57.710	0.069	58.330
	w/o Cos.	0.110	62.791	0.082	56.604	0.077	56.487	0.076	**56.525**
	Full	0.105	**60.056**	**0.076**	**55.802**	**0.070**	**56.469**	**0.068**	57.055

encoder and the structure-aware encoder by calculating a confidence map for each view, as shown in Figure 3. The pixel-wise or voxel-wise confidence maps $\{c^i_{tex}\}_{1\leq i\leq N}$ and $\{c^i_{str}\}_{1\leq i\leq N}$ are used to fuse the texture features and structure features of all the views by

$$f_{tex} = \sum_{i=1}^{N} f^i_{tex} \odot Softmax_i(c^1_{tex}, ..., c^N_{tex}). \qquad (8)$$

$$f_{str} = \sum_{i=1}^{N} f^i_{str} \odot Softmax_i(c^1_{str}, ..., c^N_{str}). \qquad (9)$$

We normalize the predicted confidence maps $\{c^i_{tex}\}_{1\leq i\leq N}$ and $\{c^i_{str}\}_{1\leq i\leq N}$ by applying $Softmax(\cdot)$ across them. The normalized confidence maps can then be used as the weights to aggregate the feature maps. This mechanism enables the weights to be automatically adjusted for any number of input views, which is very flexible. Moreover, the fusion at the feature level costs less memory but is able to produce a more continuous result.

3.4. Fusion Decoder

The fusion decoder fuses the texture feature map and the structure feature volume, and then generates the final image. After several 3D convolutions, the structure feature volume is turned into a structure feature map by merging the depth dimension into the channel dimension. We concatenate the structure feature map and the texture feature map, and then get the final image after a U-Net decoder. Instead of fusion at the pixel level, we fuse the structure representation and the texture representation at the feature level. This has three reasons: 1) it is difficult to ensure the alignment of two-branch results; 2) the features before the decoder contain more information than the decoded images; 3) fusion at the feature level enables the network to generate new contents, especially for the invisible regions.

3.5. Loss Functions

Because our STATE is an end-to-end trainable network, we directly define several losses in the image space to train our network. Our full training loss consists of a reconstruction term, a structural term, a perceptual term, a cosine term and an adversarial term. The full loss is formulated as

$$\mathcal{L} = \lambda_r \mathcal{L}_R + \lambda_s \mathcal{L}_S + \lambda_p \mathcal{L}_P + \lambda_c \mathcal{L}_C + \lambda_a \mathcal{L}_A, \qquad (10)$$

where λ_r, λ_s, λ_p, λ_c and λ_a indicate the weights of five loss terms.

Reconstruction Loss. The reconstruction loss directly guides the similarity between the generated image $\hat{I}_t$ and the ground-truth image I_t at the pixel level, which can accelerate the convergence process. $\mathcal{L}_R$ is defined as the ℓ_1 distance:

$$\mathcal{L}_R = \left\| \hat{I}_t - I_t \right\|_1 . \qquad (11)$$

Structural Loss. We also use the structural similarity (SSIM) loss $\mathcal{L}_S$ [43] with the window size of 11×11 to improve the structural similarity, which is more consistent with human perception. We compute the structural dissimilarity between the generated image $\hat{I}_t$ and the ground-truth image I_t by

$$\mathcal{L}_S = 1 - SSIM(\hat{I}_t, I_t). \qquad (12)$$

Perceptual Loss. In addition to the low-level constraints at the pixel level, we adopt the perceptual loss [15] to compute the difference between the deep features of the generated

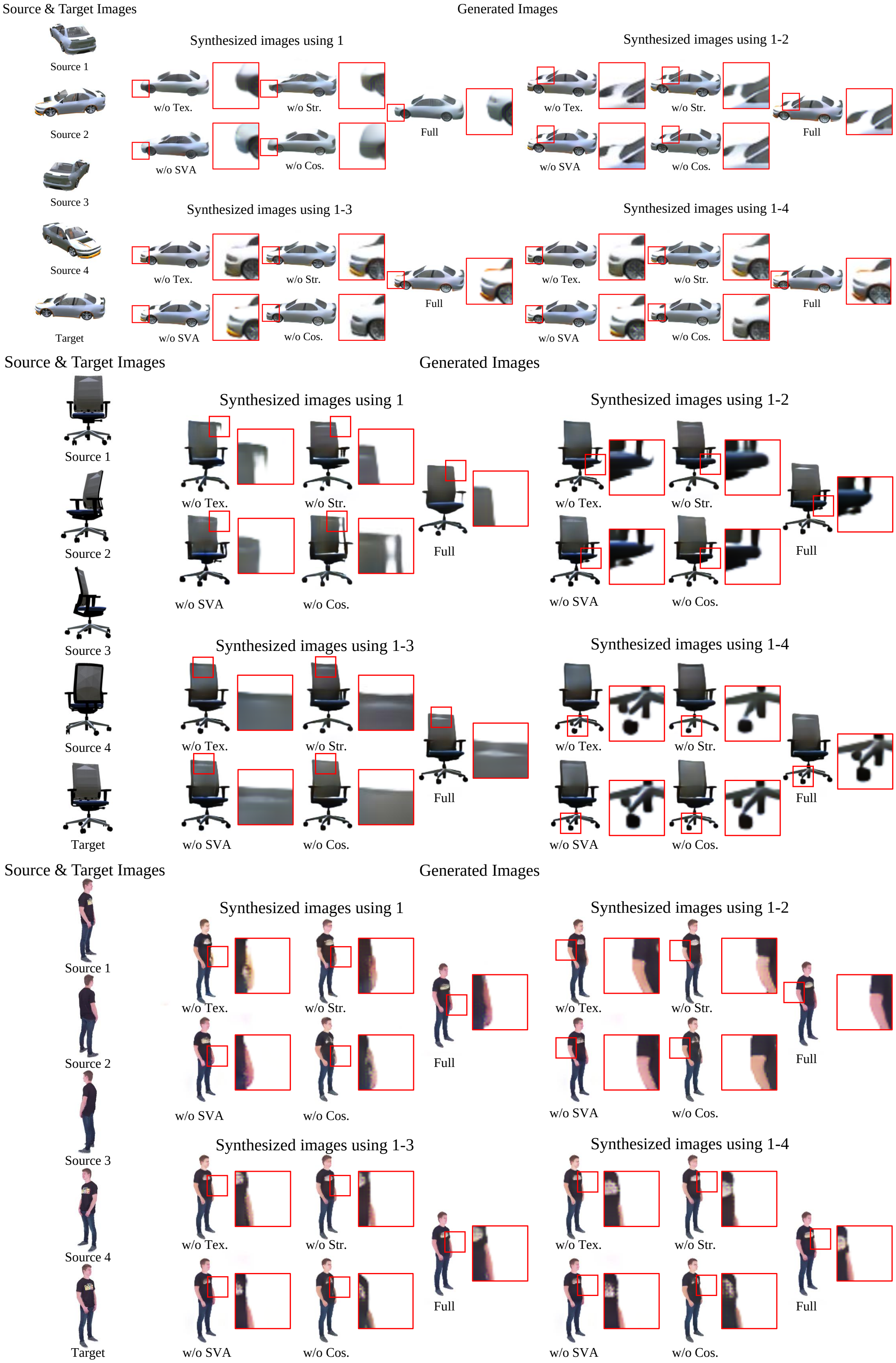

Figure 4. Qualitative comparison with four alternative designs.

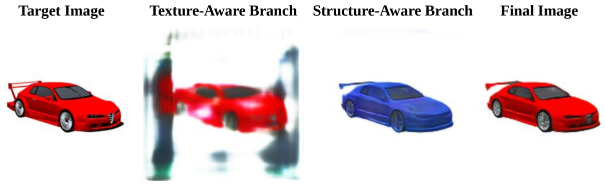

Figure 5. Disentanglement of textures and structures.

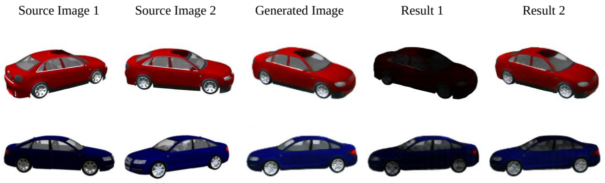

Figure 6. Confidence map of different views.

image $\hat{I}_t$ and the ground-truth image I_t in perceptual level, which is formulated as

$$\mathcal{L}_P = \sum_i \left\| \phi_i(\hat{I}_t) - \phi_i(I_t) \right\|_2 , \tag{13}$$

where ϕ_i is the output of the i-th layer of the VGG-19 [36] which is pre-trained on ImageNet [34]. We use [1, 6, 11, 16]-th layers to supervise our network.

Cosine Loss. To ensure the color consistency, we calculate the cosine similarity between the generated image $\hat{I}_t$ and the ground-truth image I_t. Cosine similarity measures the similarity between two vectors by measuring the cosine of the angle between them:

$$\mathcal{L}_C = 1 - cos(\hat{I}_t, I_t). \tag{14}$$

Discriminator Loss. We adopt the discriminator from generative adversarial networks [9], which has achieved great progress in image synthesis. It constrains the distance between the distributions of the generated image $\hat{I}_t$ and the ground-truth image I_t, which is defined as

$$\mathcal{L}_A = \mathbb{E}[\log(1 - D(\hat{I}_t))] + \mathbb{E}[\log D(I_t)], \tag{15}$$

where $D(\cdot)$ is a patch discriminator, $\log(\cdot)$ is the logarithm of base 2 and $\mathbb{E}[\cdot]$ is the expectation.

3.6. Implementation Details

Our framework is implemented in PyTorch. The hyperparameters $[\lambda_r, \lambda_s, \lambda_p, \lambda_c, \lambda_a]$ are set to be $[1, 10, 0.5, 1, 1]$ in our training. Adam optimizer [18] is used to optimize our network with the default parameters (β_1 = 0.9 and β_2 = 0.999) and the learning rate is $2e-4$. We trained our model with four source view images until convergence on the training data, which takes approximately 7 days using a single GeForce GTX 2080 Ti GPU. At the test time, generating an image takes about 90 milliseconds using a single GeForce GTX 2080 Ti GPU.

4. Experiments

4.1. Setup

Datasets. To evaluate the performance of our view synthesis approach, we conduct experiments on ShapeNet (*Chair* and *Car*) [1], in which the camera poses are represented by the rotation components around the object's central axis. We use the same training and testing splits used in [53, 38, 29, 30] (80% of models for training and the remaining 20% for testing). Each model is rendered as 256 × 256 RGB images at 18 azimuth angles sampled at 20-degree intervals and 3 elevations (0, 10 and 20 degrees), for a total of 54 viewpoints per model.

We also synthesize a dataset *Human* from 496 real scanned 3D human models[1]. Each model is rendered as 256 × 256 RGB images at 18 azimuth angles sampled at 20-degree intervals and 3 elevations (0, 10 and 20 degrees), for a total of 54 viewpoints per model. We use 80% of the models for training and the remaining 20% for testing.

Note that the models in the test images are not included in the training set.

Metrics. We use two popular metrics, Learned Perceptual Image Patch Similarity (LPIPS) [51] and Fréchet Inception Distance (FID) [12], which are generally considered to be closer to human perception, to calculate the reconstruction errors. LPIPS computes the distance between the generated image and the ground-truth image in the perceptual domain. FID calculates the Wasserstein-2 distance between the distributions of the generated images and the ground-truth images, which measures the realism of the generated images.

4.2. Ablation Study

In this section, we evaluate our method with four alternative models to assess the factors that contribute to achieving reasonable view synthesis from sparse input images. These models use the same setup, training schedule, and sequence of input images as STATE. We use the same training and test scheme as that in state-of-the-art methods [29, 38] on *Chair*, *Car* and *Human* datasets: training with 4 views and testing with 1-4 views.

The Model without Texture-Aware Branch (w/o Tex.). The model, deleting the texture-aware branch but retaining the multi-view adaptive weighting, is designed to assess the importance of the texture-aware branch, and to verify the necessity of the combination of both texture representation and structure representation.

The Model without Structure-Aware Branch (w/o Str.). The model, deleting the structure-aware branch but retaining the multi-view adaptive weighting, is designed to assess the importance of the structure-aware branch, and to verify

[1] https://web.twindom.com

Table 2. Quantitative comparison on *Chair*, *Car* and *Human* datasets.

Dataset	Method	1 view		2 views		3 views		4 views	
		LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓
Chair	TBN [29]	0.182	38.446	0.109	21.159	0.093	18.891	0.086	18.051
	pixelNeRF [49]	0.183	40.515	0.181	71.560	0.095	28.588	**0.068**	18.118
	Ours	**0.159**	**30.936**	**0.096**	**18.486**	**0.080**	**16.547**	0.074	**15.881**
Car	TBN [29]	**0.112**	**46.401**	0.091	40.404	0.084	38.841	0.080	38.129
	pixelNeRF [49]	0.155	91.252	0.145	89.553	0.101	55.887	0.083	41.496
	Ours	0.117	60.387	**0.089**	**39.052**	**0.080**	**34.472**	**0.075**	**32.290**
Human	TBN [29]	0.187	92.368	0.093	**51.535**	0.083	**51.573**	0.080	**52.262**
	pixelNeRF [49]	0.137	84.211	0.102	67.718	0.078	60.250	**0.068**	61.453
	Ours	**0.105**	**60.056**	**0.076**	55.802	**0.070**	56.469	**0.068**	57.055
Average	TBN [29]	0.160	59.072	0.098	**37.699**	0.087	36.435	0.082	36.147
	pixelNeRF [49]	0.158	71.993	0.143	76.277	0.091	48.242	0.073	40.256
	Ours	**0.127**	**50.460**	**0.087**	37.780	**0.077**	**35.829**	**0.072**	**35.075**

the necessity of the combination of both texture representation and structure representation.

The Model without Spatio-View Attention (w/o SVA). The model is trained with multi-view averaging fusion, to assess the importance of spatio-view attention.

The Model without Cosine Loss (w/o Cos.). The model with cosine loss removed is designed to assess the importance of cosine loss.

Full Model (Full). Our full model includes the two-branch encoder and the multi-view fusion at the feature level with adaptive weighting.

Table 1 gives quantitative results compared with four alternatives on *Chair*, *Car* and *Human* datasets. Our full model outperforms all the alternatives on *Chair* and *Car* datasets in terms of LPIPS and FID that are the most recently used metrics to measure the results from perception and realism. Note that spatio-view attention aggregation is not used when the test input is single view. Therefore, the LPIPS values of the w/o SVA model and the Full model are similar on *Human* dataset. On the other hand, all the models in ablation study are trained on the input of four views, and different confidences are assigned to different views due to the SVA module of full model. However, when the test input is single view that has low confidence, the results may be affected. Besides, the clothed posed human has complex color and is asymmetric, which influences the learning of structures. Therefore, the FID of the Full model is slightly worse than that of w/o Cos. model for the input of four views on *Human* dataset.

Some visual results are shown in Figure 4. It can be seen that the w/o Tex. model can generate correct structures, but the textures in the source images cannot be well maintained, *e.g.*, the head of car. The w/o Str. model can recover the detailed textures, especially on *Car* and *Human* datasets, but fails to keep the shape consistency. The w/o SVA model fails to effectively fuse the results of two branches, and thus the results lose some textures or structures, such as the texture of car, the back and the legs of chair and the arms of human. The w/o Cos. model cannot ensure the color consistency, such as the head of car. On the contrary, our full model can achieve the consistency of color, texture and structure.

To verify the disentanglement of textures and structures, we also visualize the results of two branches. We output the result of one branch by zeroing out the features of the other branch. Figure 5 demonstrates that our method can effectively disentangle textures and structures to generate realistic images with correct shapes and textures.

We visualize the confidence maps to demonstrate the effect of spatio-view attention aggregation in Figure 6. Taking novel view synthesis with two views as an example, the first two columns are the source images, the third column is the generated image, and the last two columns give the visualizations of the confidence maps which are computed by multiplying the confidence map with the generated image. As shown in the figure, the generated image obtains more texture information from the source image 2 due to the similarity of the target view and the source view 2, which demonstrates that our spatio-view attention aggregation can select more relevant information from the inputs of different views.

More results can be found on the project website[2].

4.3. Comparisons

We compare our method with TBN [29] and pixelNeRF [49]. For simplicity, we omit comparisons with earlier works [37, 38] that have already been compared in TBN or pixelNeRF, and the methods that do not work well for sparse views [3, 24, 25, 33]. We use the same training and test scheme as that in TBN [29] on *Chair* and *Car* datasets: training with 4 views and testing with 1-4 views. For the case of single view input, we use single view for training,

[2] http://cic.tju.edu.cn/faculty/likun/projects/STATE/

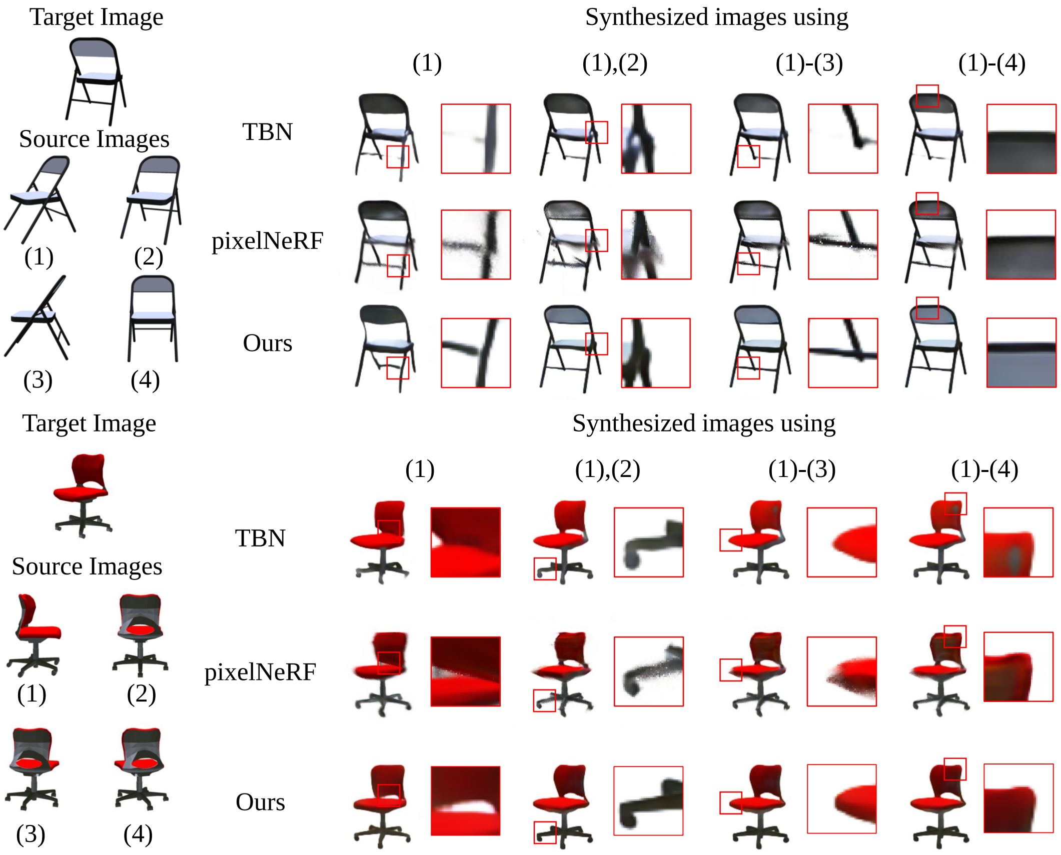

Figure 7. Qualitative comparison on *Chair* dataset.

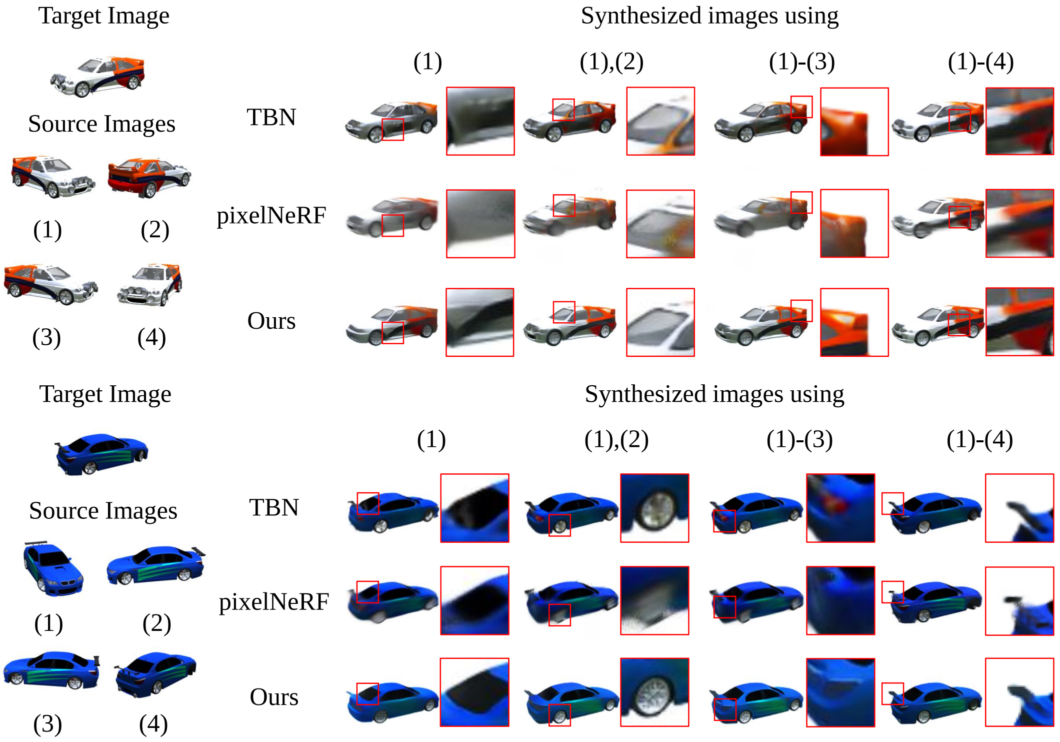

Figure 8. Qualitative comparison on *Car* dataset.

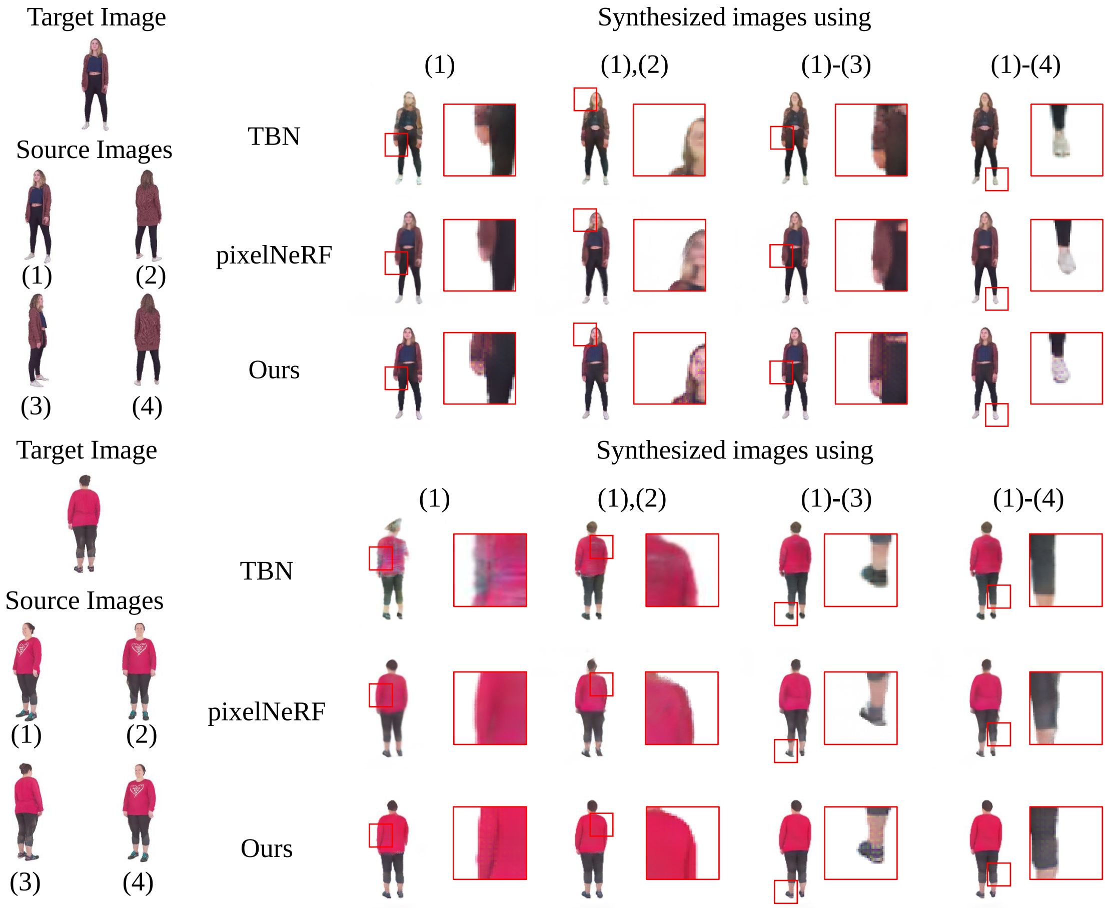

Figure 9. Qualitative comparison on *Human* dataset.

because multi-view adaptive weighting is not used. The pre-trained models of TBN [29] on *Chair* and *Car* datasets are used and we re-train TBN [29] on *Human* dataset for fair comparison with the same training and test scheme: training with 4 views and testing with any other views. We also re-train pixelNeRF [49] on the *Car*, *Chair* and *Human* datasets for fair comparison: training with 4 views and testing with 2-4 views. For the case of single view input, we use single view for training according to the suggestion of the author.

Table 2 gives the quantitative comparison results on *Chair*, *Car* and *Human* datasets. It can be seen that our proposed method outperforms the other methods in terms of FID by a significant margin on *Chair* dataset, even in the challenging case of single-view input. For the *Car* dataset, benefiting from the spatio-view attention, our method achieves the best performance for the multi-view inputs. The cars are left-right symmetrical, not front-to-back. Therefore, our texture-aware branch is difficult to estimate reasonable textures when there are lots of occlusions in front of or behind the car for single view, which leads to some deviations of the final textures, even if the shape estimated by the structure-aware branch is accurate. For the *Human* dataset, our method achieves the best performance for all the cases in terms of LPIPS. The clothed posed human has complex color and is asymmetric, which influences the learning of structures. Therefore, the FIDs of our method are not the best for the multi-view inputs on *Human* dataset. From the average results of all datasets, our method achieves the best performance for nearly all the views except for slight performance variation in terms of FID at two views.

Visual results for several challenging examples with large viewpoint transformations on *Chair*, *Car* and *Human* datasets are shown in Figure 7, Figure 8 and Figure 9. Due to the limited representation resolution, TBN [29] is difficult to recover the details of the image, such as the chair legs, and the textures of car and people. PixelNeRF [49] generates some artifacts along the structural edges.

In contrast, our method can obtain detailed textures while maintaining the structures of the objects, *e.g.*, the stripes on the car and the suit on the person. Thanks to the disentangled learning of the structure representation and the texture representation, the invisible regions and detailed textures are successfully recovered by our method for any

Table 3. Statistical results of the user study.

Cases	Females			Males			Independent T Test	
	Method A	Method B	Method C	Method A	Method B	Method C	*t*	*p*
Case 1	27.12%	16.95%	**55.93%**	20.51%	9.62%	**69.87%**	-1.487	0.140
Case 2	16.38%	15.82%	**67.80%**	6.41%	15.38%	**78.21%**	-1.875	0.064
Case 3	13.56%	14.12%	**72.32%**	8.97%	11.54%	**79.49%**	-1.072	0.286
Case 4	20.34%	12.99%	**66.67%**	19.23%	14.10%	**66.67%**	-0.086	0.932
Case 5	16.38%	13.00%	**70.62%**	18.59%	7.69%	**73.72%**	-0.067	0.946
Case 6	9.61%	10.73%	**79.66%**	13.46%	8.33%	**78.21%**	0.510	0.611
Case 7	16.95%	9.60%	**73.45%**	7.05%	11.54%	**81.41%**	-1.685	0.095
Case 8	15.25%	14.13%	**70.62%**	8.33%	9.62%	**82.05%**	-1.774	0.079
Case 9	15.82%	14.69%	**69.49%**	10.90%	8.33%	**80.77%**	-1.437	0.154
Case 10	16.95%	15.82%	**67.23%**	16.67%	10.25%	**73.08%**	-0.489	0.626
Case 11	14.69%	14.69%	**70.62%**	9.62%	13.46%	**76.92%**	-0.979	0.330
Case 12	12.99%	9.61%	**77.40%**	12.82%	8.97%	**78.21%**	-0.087	0.931
Average	16.40%	13.47%	**70.13%**	12.89%	10.76%	**76.35%**	-1.115	0.267

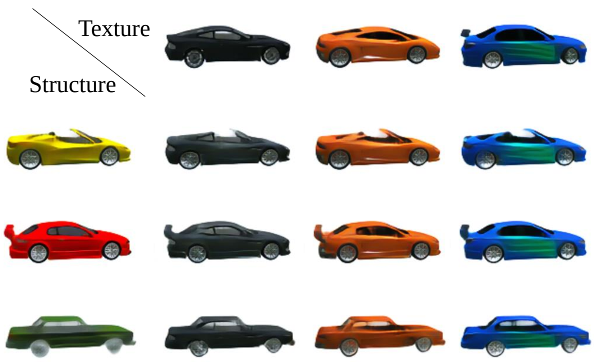

Figure 10. The results of texture or structure swapping on *Car* dataset.

number of input views. By fusing and decoding the two representations, our method does not suffer from the missing pixels. This proves that our method can generate visually better and more realistic images.

More results can be found on the project website[3].

4.4. User Study

To better evaluate the results of our method, we perform perceptual evaluation with a user study, compared with state-of-the-art methods. In the user study, we show the results of TBN [29] (Method A), pixelNeRF [49] (Method B), our method (Method C) and the ground-truth for the same input images in twelve cases with three questions per case (38 questions in total including the questions related to gender and age of the participant): 1-4 views as input on the *Car*, *Chair* and *Human* datasets. The results shown are randomly selected, and the users are required to choose the one closest to the ground-truth in terms of texture, structure and overall quality for each case from A, B and C. We have collected 111 answers, including 59 females and 52 males with different ages (108 users between 18 and 40, 1 user between 40 and 60, 2 users beyond 60). Table 3 presents statistical results of the user study. For each of the two genders, we gives the percentage of participants who choose the particular method for each case, and the average results over the twelve cases are also shown. In addition to the percentage, we also make independent T test [10] between the result and the gender. For the independent T test, *t* is a statistical vari-

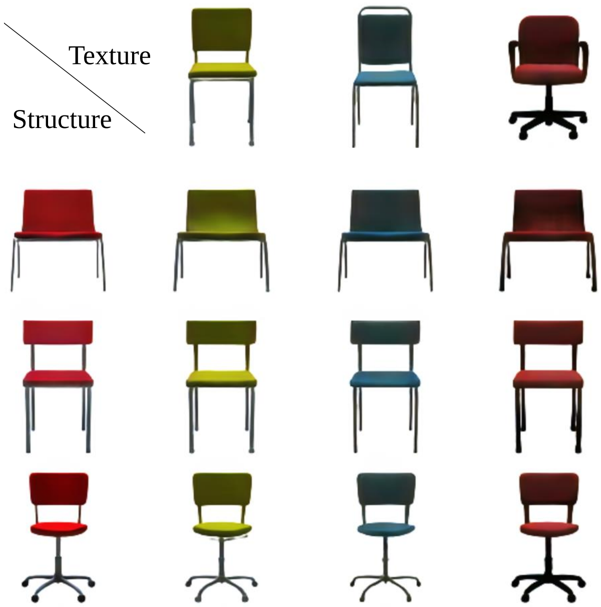

Figure 11. The results of texture or structure swapping on *Chair* dataset.

[3] http://cic.tju.edu.cn/faculty/likun/projects/STATE/

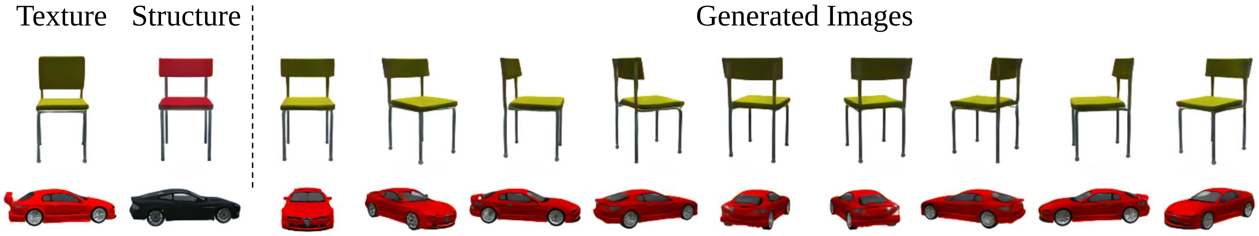

Figure 12. The results of texture or structure swapping for various views.

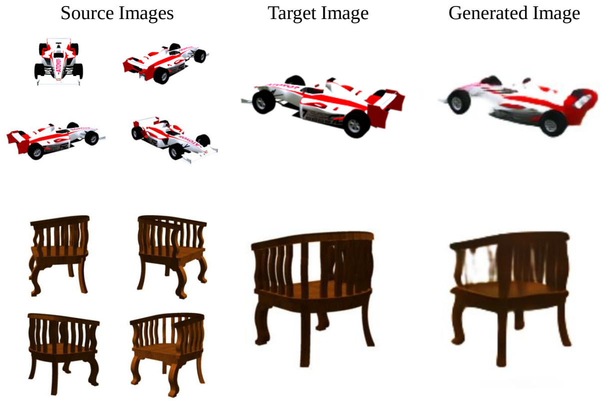

Figure 13. Failure cases.

able calculated from the results and we can get p by looking up the table according to t. The p value greater than 0.05 means there is no significant difference between the results from the two genders. We use 1, 2 and 3 to represent Methods A, B and C, respectively, and we average the results of the three questions in each case. As shown in Table 3, the p values are greater than 0.05, which demonstrates that there is no significant difference between the results of females and males, and the user study results are not dependent on the gender. In a word, our method achieves better results in the user study.

More results can be found on the project website[4].

4.5. Applications

Our method does not explicitly constrain texture and structure, however, as the branches are capable of generating better structure and texture respectively, this implicitly leads to disentanglement. We also achieve texture or structure swapping with trained model for novel view synthesis.

With the texture branch and the structure branch, we can easily edit the texture and the structure by changing the inputs of each branch. Figure 10 and Figure 11 show some disentangled results on *Car* and *Chair* datasets. The first row provides the texture information and the first column gives the structure information. Each result in the other position (row i and column j) is a decoded result of the combination of the structure representation of the first column image of row i with the texture representation of the first row image of column j. It can be seen that the structure of the result in each row is consistent with that of the left in this row, and the texture of the result in each column is consistent with that of the top in this column. Figure 12 shows some disentangled results for various views, which proves that our method achieves the disentanglement of texture and structure.

4.6. Failure Cases

Although our method generates realistic images with reasonable structures and detailed textures in most cases, it cannot cope well with the structures and textures that deviate extremely from the training set distribution. The neural network predicts the outputs by the interpolation operator in the manifold built on the training data. Therefore, it is difficult to predict reasonable results for some challenging cases, especially with extremely complex structures and textures. Figure 13 shows some failure cases of our method. It can be seen that our model fails to predict correct textures and shapes for extremely complex cases.

5. Conclusions

In this paper, we propose STATE, an end-to-end deep neural network, for sparse view synthesis from input images by learning structure and texture representations. Specifically, we propose a two-branch encoder to extract implicit structure representation and deformed texture representation. We also propose spatio-view attention to adaptively fuse multi-view information at the feature level by regressing pixel-wise or voxel-wise confidence maps. By decoding the aggregated feature, STATE can generate realistic images with reasonable structures and detailed textures. Experimental results demonstrate that our method achieves better performance than state-of-the-art methods. We verify our hypothesis by comprehensive ablation study. Our method also enables texture and structure editing applications benefitting from implicit disentanglement of structures and textures.

Acknowledgements. This work was supported in part

[4] http://cic.tju.edu.cn/faculty/likun/projects/STATE/

by the National Natural Science Foundation of China (62171317 and 62122058).

References

[1] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 7

[2] X. Chen, J. Song, and O. Hilliges. Monocular neural image based rendering with continuous view control. In *International Conference on Computer Vision*, pages 4090–4100, 2019. 3

[3] J. Chibane, A. Bansal, V. Lazova, and G. Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Computer Vision and Pattern Recognition*, pages 7911–7920, 2021. 8

[4] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. 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. 3

[5] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 1

[6] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: learning to predict new views from the world's imagery. In *Computer Vision and Pattern Recognition*, pages 5515–5524, 2016. 1, 3

[7] Y. Galama and T. Mensink. IterGANs: Iterative GANs to learn and control 3D object transformation. *Computer Vision and Image Understanding*, 189:102803, 2019. 3

[8] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016. 3

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 7

[10] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, A. J. Smola, et al. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, volume 20, pages 585–592. Citeseer, 2007. 11

[11] P. Guo, M. A. Bautista, A. Colburn, L. Yang, D. Ulbricht, J. M. Susskind, and Q. Shan. Fast and explicit neural view synthesis. *arXiv preprint arXiv:2107.05775*, 2021. 3

[12] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 7

[13] Y. Hou, A. Solin, and J. Kannala. Novel view synthesis via depth-guided skip connections. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3119–3128, 2021. 3

[14] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015. 2

[15] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 5

[16] A. Kar, C. Häne, and J. Malik. Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems*, pages 365–376, 2017. 3

[17] J. Kim and Y. M. Kim. Novel view synthesis with skip connections. In *IEEE International Conference on Image Processing*, pages 1616–1620. IEEE, 2020. 2

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[19] Y. Kwon, S. Petrangeli, D. Kim, H. Wang, H. Fuchs, and V. Swaminathan. Rotationally-consistent novel view synthesis for humans. In *ACM International Conference on Multimedia*, pages 2308–2316, 2020. 2

[20] H.-A. Le, T. Mensink, P. Das, and T. Gevers. Novel view synthesis from single images via point cloud transformation. *arXiv preprint arXiv:2009.08321*, 2020. 1, 3

[21] K. Li, J. Zhang, Y. Liu, Y.-K. Lai, and Q. Dai. PoNA: Pose-guided non-local attention for human pose transfer. *IEEE Transactions on Image Processing*, 29:9584–9599, 2020. 2

[22] X. Liu, Z. Guo, J. You, and B. V. Kumar. Dependency-aware attention control for image set-based face recognition. *IEEE Transactions on Information Forensics and Security*, 15:1501–1512, 2019. 2

[23] X. Liu, B. Vijaya Kumar, C. Yang, Q. Tang, and J. You. Dependency-aware attention control for unconstrained face recognition with image sets. In *European Conference on Computer Vision*, pages 548–565, 2018. 2

[24] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Transactions on Graphics*, 2019. 3, 8

[25] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020. 3, 8

[26] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang. HoloGAN: Unsupervised learning of 3D representations from natural images. In *International Conference on Computer Vision*, pages 7588–7597, 2019. 3

[27] T. Nguyen-Phuoc, C. Richardt, L. Mai, Y.-L. Yang, and N. Mitra. BlockGAN: Learning 3D object-aware scene representations from unlabelled images. *arXiv preprint arXiv:2002.08988*, 2020. 3

[28] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 3

[29] K. Olszewski, S. Tulyakov, O. Woodford, H. Li, and L. Luo. Transformable bottleneck networks. In *International Conference on Computer Vision*, pages 7648–7657, 2019. 1, 2, 3, 4, 7, 8, 10, 11

[30] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3D view synthesis. In *Computer Vision and Pattern Recognition*, pages 3500–3509, 2017. 2, 7

[31] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Computer Vision and Pattern Recognition*, pages 165–174, 2019. 3

[32] Y. Ren, X. Yu, J. Chen, T. H. Li, and G. Li. Deep image spatial transformation for person image generation. In *Computer Vision and Pattern Recognition*, pages 7690–7699, 2020. 1, 2

[33] G. Riegler and V. Koltun. Free view synthesis. In *European Conference on Computer Vision*, pages 623–640. Springer, 2020. 8

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 7

[35] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *International Conference on Computer Vision*, pages 2304–2314, 2019. 3

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[37] V. Sitzmann, M. Zollhöfer, and G. Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*, 2019. 1, 3, 8

[38] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *European Conference on Computer Vision*, pages 155–171, 2018. 1, 2, 7, 8

[39] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3D models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016. 1, 2

[40] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020. 3

[41] A. Trevithick and B. Yang. GRF: Learning a general radiance field for 3D scene representation and rendering. *arXiv preprint arXiv:2010.04595*, 2020. 2

[42] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Computer Vision and Pattern Recognition*, pages 2626–2634, 2017. 1, 3

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[44] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016. 2

[45] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee. Weakly-supervised disentangling with recurrent transformations for 3D view synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015. 1, 2

[46] J. Yang, X. Ye, and P. Frossard. Global auto-regressive depth recovery via iterative non-local filtering. *IEEE Transactions on Broadcasting*, 65(1):123–137, 2018. 3

[47] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang. Color-guided depth recovery from RGB-D data using an adaptive autoregressive model. *IEEE Transactions on Image Processing*, 23(8):3443–3458, 2014. 3

[48] M. Yin, L. Sun, and Q. Li. ID-Unet: Iterative soft and hard deformation for view synthesis. In *Computer Vision and Pattern Recognition*, pages 7220–7229, 2021. 2

[49] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 1, 3, 8, 10, 11

[50] J. Zhang, K. Li, Y.-K. Lai, and J. Yang. PISE: Person image synthesis and editing with decoupled GAN. In *Computer Vision and Pattern Recognition*, pages 7982–7990, 2021. 2

[51] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Computer Vision and Pattern Recognition*, pages 586–595, 2018. 7

[52] H. Zhao, J. Zhang, Y.-K. Lai, Z. Zheng, Y. Xie, Y. Liu, and K. Li. High-fidelity human avatars from a single RGB camera. In *Computer Vision and Pattern Recognition*, 2022. 3

[53] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *European Conference on Computer Vision*, pages 286–301. Springer, 2016. 1, 2, 7