3D-aware Image Synthesis via Learning Structural and Textural Representations

The Chinese University of Hong Kong / Zhejiang University / Bytedance Inc

CVPR 2022

Contents

1. Introduction

2. Related Works

3. Method

4. Experiments

5. Conclusion

1. Introduction

I. Nerf-GAN

Maps raw 3D coordinates to density and color conditioned on the given latent code.

Problems?

① Only input 3D coordinate.

→With a very local receptive field, hard for MLP to represent the underlying global structure.

② Requires sampling numerous points along the camera ray regarding each pixel.

→ computational cost / unsatisfying performance for high-resolution image generation

1. Introduction

II. 2D GAN

Benefits from valid representations learned by the generator.

Representative features encode rich texture and structure information, thereby enhancing the synthesis quality and the controllability of image GANs.

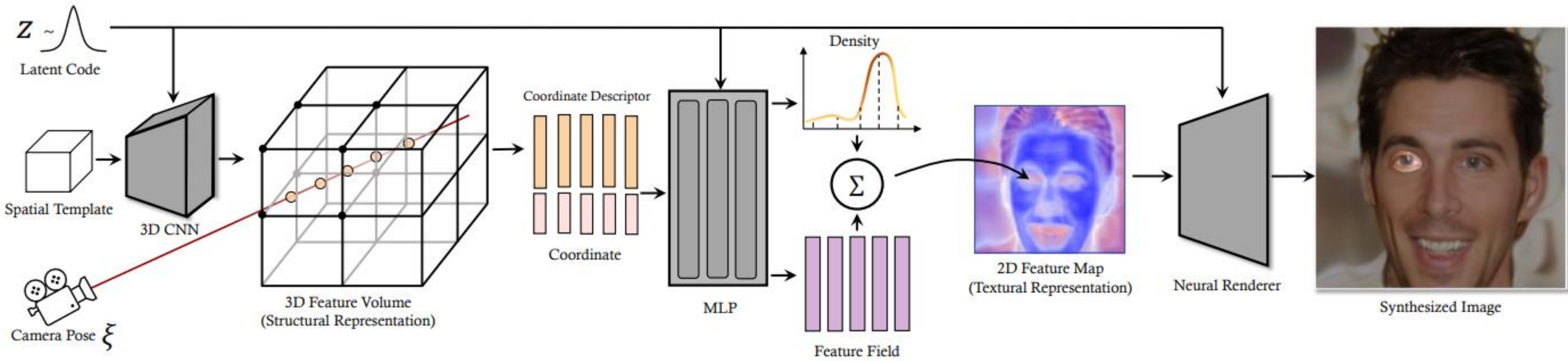
Ex)

Xu et al. : Face synthesis model is aware of the landmark positions of the output face.

Yang et al. : Identify the multilevel variation factors emerging from generating bedroom images.

References)
Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou. Generative hierarchical features from synthesizing images.
Ceyuan Yang, Yujun Shen, and Bolei Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis.

1. Introduction



III. Volume-GAN

3D-aware image synthesis through explicitly learning a structural and a textural representation, enabling the disentangled control of the shape and the appearance.

Method

1. Generate a feature volume using a 3D convolutional network (structural representation).

2. Query a coordinate descriptor from the feature volume for each 3D point.

3. Input coordinate descriptor and raw coordinate through a NeRF-like model to create a feature field.

4. The feature field is accumulated into a 2D feature map (textural representation), followed by a CNN with 1 × 1 kernel size to render the final image.

1. Introduction

IV. Contributions

– Taking the FFHQ dataset under 256 × 256 resolution as an instance, improve FID from 36.7 to 9.1.

– 3D-aware image synthesis on the challenging indoor scene dataset (LSUN bedroom).

– Stable control of the object pose and better consistency across different viewpoints, benefiting from the learned structural representation (feature volume).

– Empirical study on the learned structural and textural representations, and analyze the trade-off between the image quality and the 3D property.

2. Related Works

I. NeRF

Given a sampled ray, predict the colors and densities of all the points, which are then accumulated into the pixel via volume rendering.

$$\mathbf{c}(\mathbf{x}, \mathbf{d}) = \phi_c(\Phi(\mathbf{x}), \mathbf{d}),$$
$$\sigma(\mathbf{x}) = \phi_d(\Phi(\mathbf{x})),$$

3D reconstruction and novel view synthesis

II. NeRF-GAN

Geometry and appearance of the rendered image will vary according to the input z, resulting in diverse generation.

$$F(\mathbf{x}, \mathbf{d}, \mathbf{z}) = (\mathbf{c}, \sigma)$$

Generator is asked to compete with a discriminator of GANs to mimic the distribution of real 2D images.

2. Related Works

III. 3D-Aware Image Synthesis

Some prior works use voxel.

VON : generates a 3D shape represented by voxels which is then projected into 2D image space by a differentiable renderer.

HoloGAN : propose voxelized and implicit 3D representations and then render it to 2D space with a reshape operation.

→ Synthesized images suffer from the fine details and identity shift because of the voxel resolution restriction.

2. Related Works

III. 3D-Aware Image Synthesis

GRAF / π-GAN : model 3D shapes by neural implicit representation, which maps the coordinates to the RGB color.

GOF / ShadeGAN : occupancy field and albedo field instead of radiance field for image rendering.

→ Due to the computationally intensive rendering process, cannot synthesize high resolution images with good visual quality.

Giraffe : first render low-resolution feature maps with neural feature fields and then generate high-resolution images with 2D CNNs.

→ severe artifacts across different camera views are introduced because CNN-based decoder harms the 3D consistency.

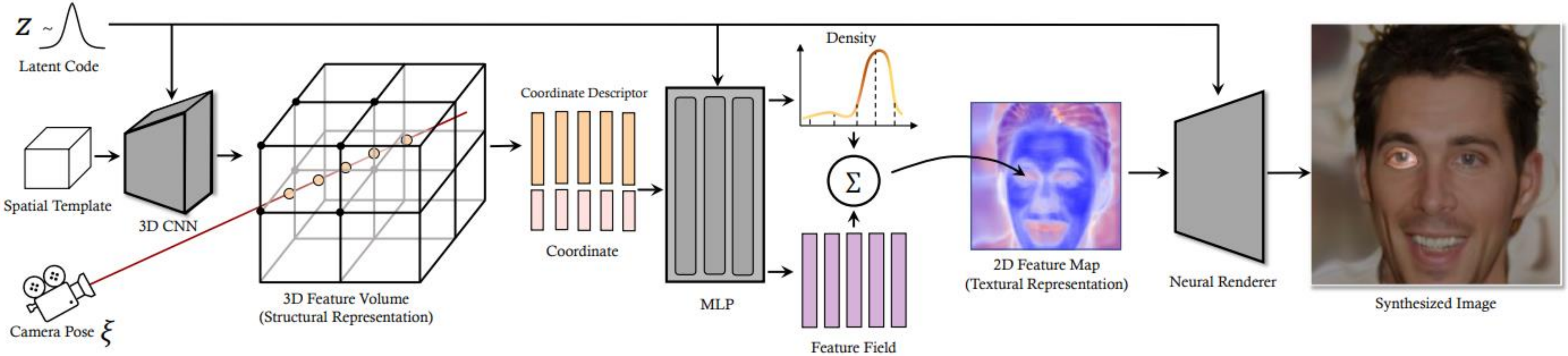
2. Related Works

III. 3D-Aware Image Synthesis

StyleNeRF : also adopts 1 × 1 convolution block to synthesize high-quality images.

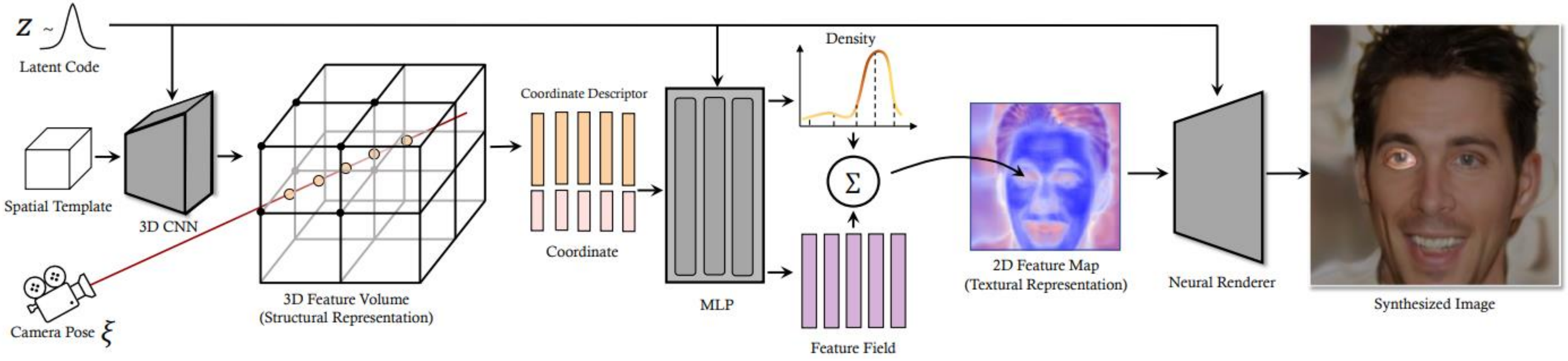
However, we adopt the feature volume to provide the structural description for the synthesized object instead of using regularizers to improve the 3D properties.

3. Method



z ~
Latent Code
Spatial Template
3D CNN
Camera Pose ξ
3D Feature Volume
(Structural Representation)
Coordinate Descriptor
Coordinate
MLP
Density
Σ
Feature Field
2D Feature Map
(Textural Representation)
Neural Renderer
Synthesized Image

3. Method



I. 3D Feature Volume as Structural Representation

Learn a 3D feature volume V, as the structural representation which characterizes the underlying 3D structure.

Use 3D convolutional layers with the Leaky ReLU functions.

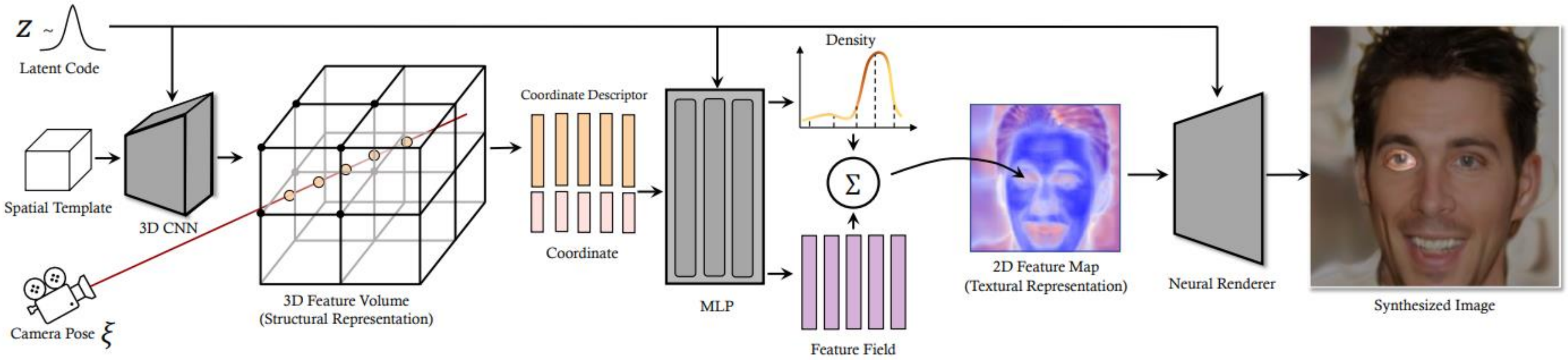
Apply AdaIN to the output of each layer to introduce diversity to the feature volume.

$$\mathbf{V} = \psi_{n_s-1} \circ \psi_{n_s-2} \circ \ldots \circ \psi_0(\mathbf{V}_0),$$

$$\psi_i(\mathbf{V}_i) = \mathtt{AdaIN}\Big(\mathtt{LReLU}\big(\mathtt{Conv}(\mathtt{Up}(\mathbf{V}_i, \underline{s_i}))\big), \mathbf{z}\Big),$$

upsampling
scale

3. Method



II. 2D Feature Map as Textural Representation

Query the coordinate descriptor v from the feature volume V, given a 3D coordinate x.
Concatenate it with x to obtain v^x as the input.
The implicit function transforms v^x to the density and feature vector.

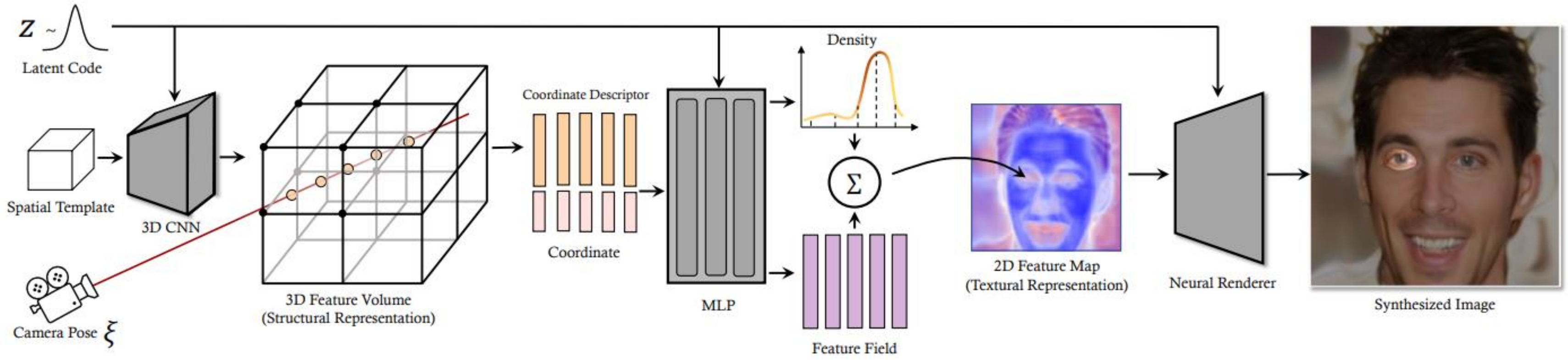
$$
\begin{aligned}
\mathbf{v} &= \mathtt{trilinear}(\mathbf{V}, \mathbf{x}),\\
\mathbf{v}^{\mathbf{x}} &= \mathtt{Concat}(\mathbf{v}, \mathbf{x}),\\
\Phi(\mathbf{v}^{\mathbf{x}}) &= \phi_{n-1} \circ \phi_{n-2} \circ \ldots \circ \phi_0(\mathbf{v}^{\mathbf{x}}),\\
\phi_i(\mathbf{v}_i^{\mathbf{x}}) &= \sin\left(\gamma_i(\mathbf{z}) \cdot (\mathbf{W}_i \mathbf{v}_i^{\mathbf{x}} + \mathbf{b}_i) + \beta_i(\mathbf{z})\right),\\
\mathbf{f}(\mathbf{x}, \mathbf{d}) &= \phi_f(\Phi(\mathbf{v}^{\mathbf{x}}), \mathbf{d}),\\
\sigma(\mathbf{x}) &= \phi_d(\Phi(\mathbf{v}^{\mathbf{x}})),
\end{aligned}
$$

A per-pixel final feature m(r) can be obtained via volume rendering along a ray.
A collection of m regarding different rays become a 2D feature map M.

$$\mathbf{m}(\mathbf{r}) = \sum_{k=1}^{N} T_k (1 - \exp(-\sigma(\mathbf{x}_k)\delta_k))\mathbf{f}(\mathbf{x}_k, \mathbf{d}),$$

$$T_k = \exp(-\sum_{j=1}^{k-1} \sigma(\mathbf{x}_j)\delta_j).$$

3. Method



III. Final Image Synthesis

Learn a feature map at a low resolution, followed by a CNN to render a high fidelity result.

CNN consists of Modulated Convolutional Layers & LReLU.

To avoid the CNN from weakening the 3D consistency, use 1×1 kernel size for all layers such that the per-pixel feature can be processed independently.

$$\mathbf{I}^f = f_{n_t-1} \circ f_{n_t-2} \circ \ldots \circ f_0(\mathbf{M}),$$

$$f_i(\mathbf{M}_i) = \mathtt{LReLU}\big(\mathtt{ModConv}(\mathbf{M}_i, \underline{t_i}, \mathbf{z})\big),$$

upsampling scale

3. Method

Training

The whole generation process : $I^f = G(z, \xi)$

z : latent code sampled from a Gaussian distribution N(0, 1)

ξ : camera pose sampled from a prior distribution p_ξ. p_ξ is tuned for different datasets as either Gaussian or Uniform.

Loss

$$\min \mathcal{L}_G = \mathbb{E}_{\mathbf{z}\sim p_z, \xi\sim p_\xi}[f(D(G(\mathbf{z}, \xi)))],$$

$$\min \mathcal{L}_D = \mathbb{E}_{\mathbf{I}^r\sim p_D}[f(-D(\mathbf{I}^r)) + \lambda \underbrace{||\nabla_{\mathbf{I}^r} D(\mathbf{I}^r)||_2^2}_{\text{gradient penalty}}],$$

4. Experiments

Datasets

CelebA, Cats, FFHQ, CompCars, LSUN bedroom, Carla

Baselines

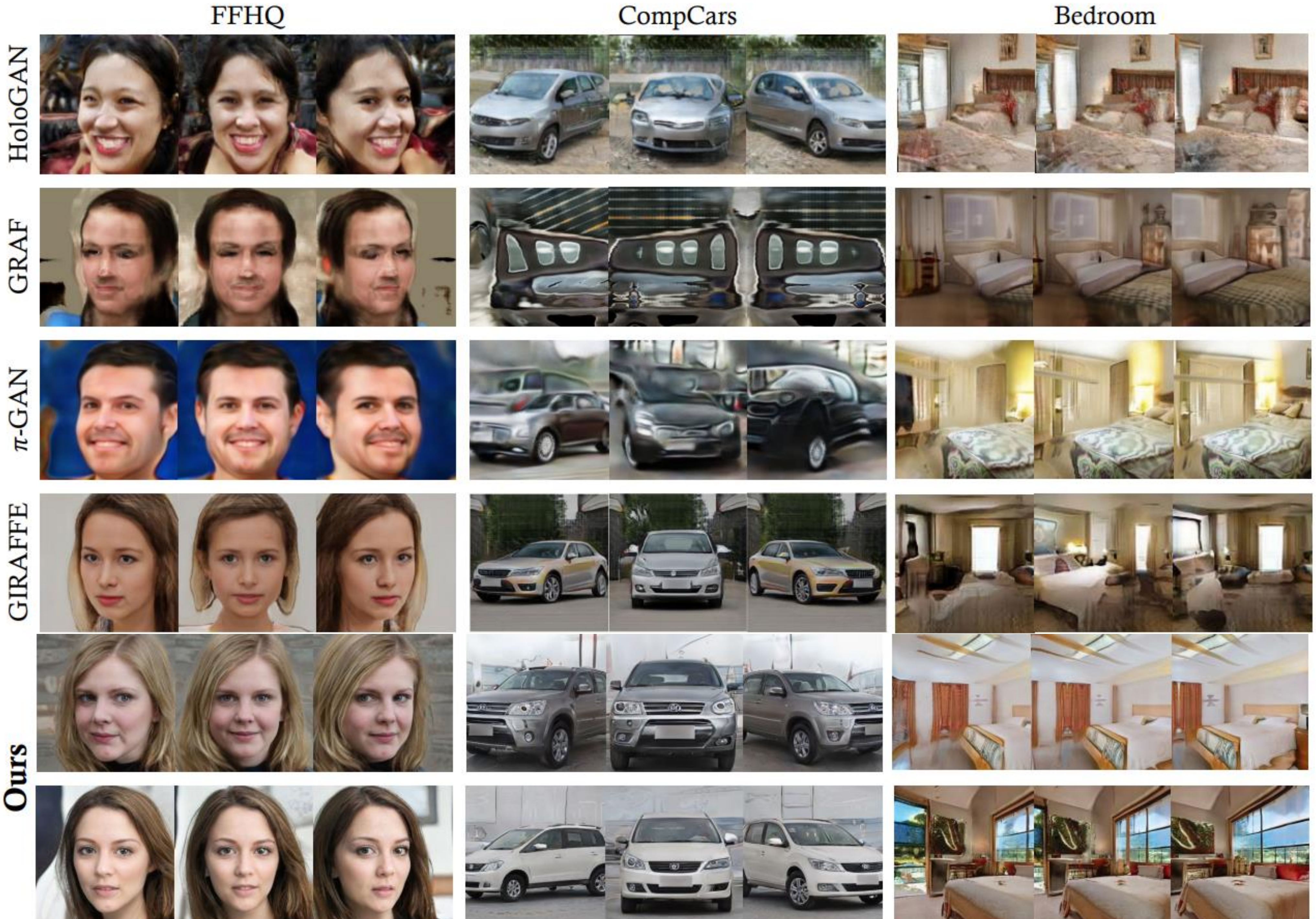
HoloGAN, GRAF, π-GAN, GIRAFFE

Implementation Details

The learnable 3D template V_0 are randomly initialized in $4 \times 4 \times 4$ shape and 3D convolutions with kernel size $3 \times 3 \times 3$ are stacked to embed the template, resulting in the feature volume in $32 \times 32 \times 32$ resolution.

Sample rays in a resolution of 64×64, and 4 conditioned MLPs (SIREN) with 256 dimensions are adopted to model the feature field.

Apply progressive training strategy used in PG-GAN to achieve better image qualities.


FFHQ
CompCars
Bedroom
HoloGAN
GRAF
π-GAN
GIRAFFE
Ours

4. Experiments

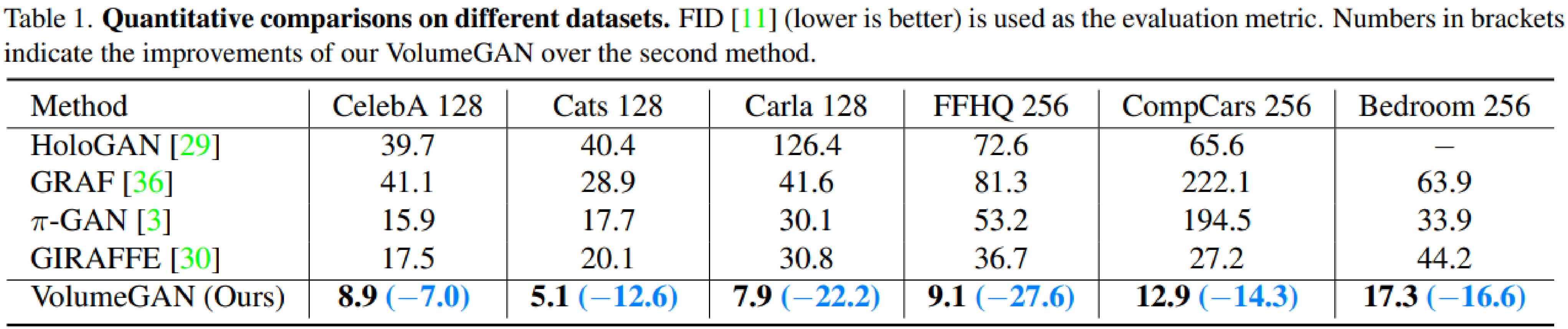
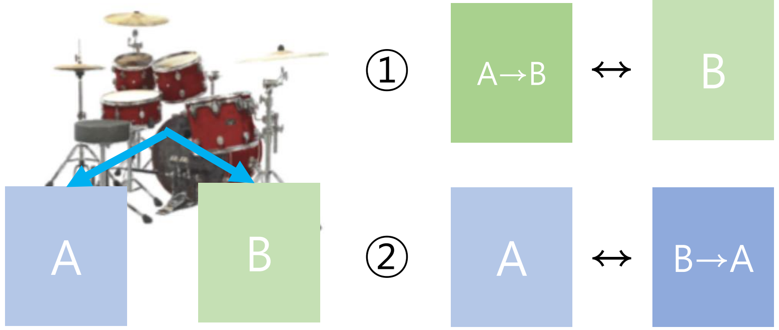


Table 1. **Quantitative comparisons on different datasets.** FID [11] (lower is better) is used as the evaluation metric. Numbers in brackets indicate the improvements of our VolumeGAN over the second method.

Method	CelebA 128	Cats 128	Carla 128	FFHQ 256	CompCars 256	Bedroom 256
HoloGAN [29]	39.7	40.4	126.4	72.6	65.6	–
GRAF [36]	41.1	28.9	41.6	81.3	222.1	63.9
π-GAN [3]	15.9	17.7	30.1	53.2	194.5	33.9
GIRAFFE [30]	17.5	20.1	30.8	36.7	27.2	44.2
VolumeGAN (Ours)	**8.9** **(−7.0)**	**5.1** **(−12.6)**	**7.9** **(−22.2)**	**9.1** **(−27.6)**	**12.9** **(−14.3)**	**17.3** **(−16.6)**

4. Experiments



Ablation on CelebA 128 × 128

Measure multi-view consistency & precision of 3D control

- Reprojection error : Extract the geometry of an object from the generated density using marching cubes. Then, sample five viewpoints uniformly to synthesize the images.
The depth of each image is rendered from the resulting extracted mesh, which is used to calculate the reprojection error on two consecutive views by warping them each other.

- Pose error : The L1 distance between the given camera pose and the predicted pose from the head pose estimator (GT).

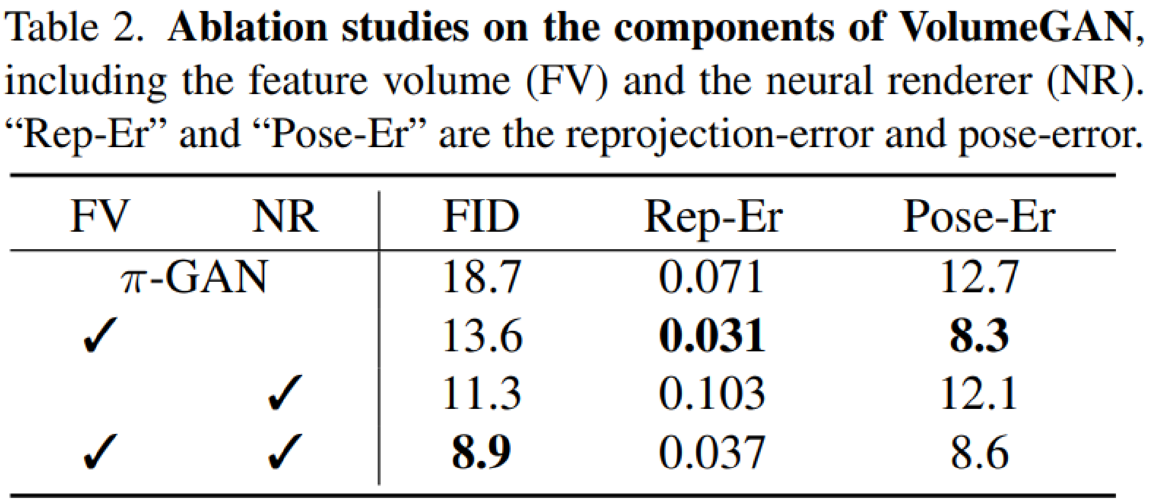


Table 2. **Ablation studies on the components of VolumeGAN**, including the feature volume (FV) and the neural renderer (NR). “Rep-Er” and “Pose-Er” are the reprojection-error and pose-error.

FV	NR	FID	Rep-Er	Pose-Er
π-GAN		18.7	0.071	12.7
✓		13.6	**0.031**	**8.3**
	✓	11.3	0.103	12.1
✓	✓	**8.9**	0.037	8.6

4. Experiments

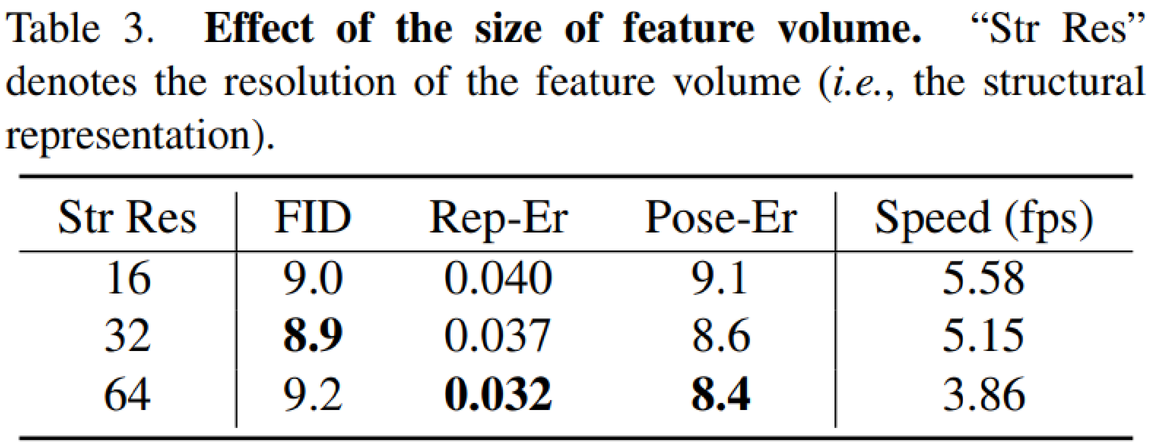


Table 3. **Effect of the size of feature volume.** "Str Res" denotes the resolution of the feature volume (*i.e.*, the structural representation).

Str Res	FID	Rep-Er	Pose-Er	Speed (fps)
16	9.0	0.040	9.1	5.58
32	**8.9**	0.037	8.6	5.15
64	9.2	**0.032**	**8.4**	3.86

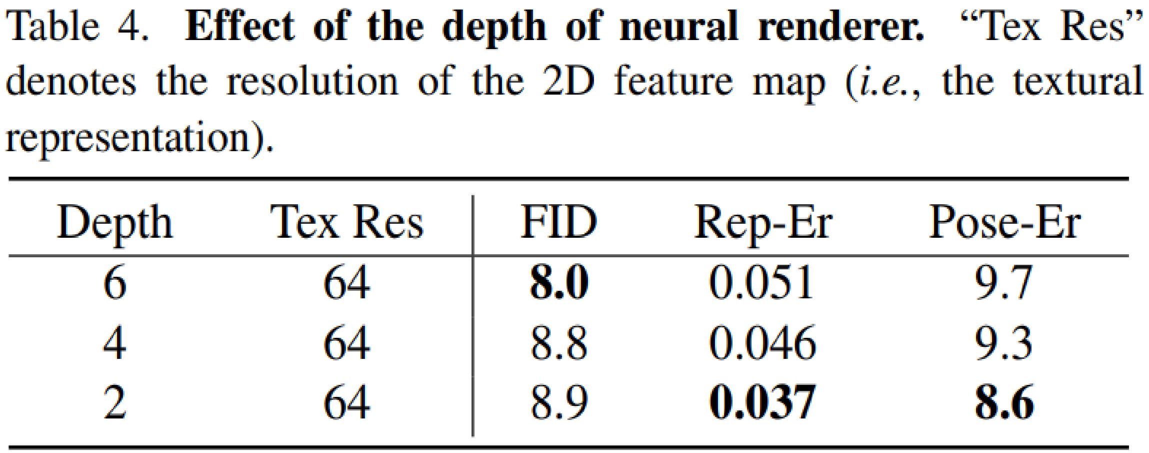


Table 4. **Effect of the depth of neural renderer.** "Tex Res" denotes the resolution of the 2D feature map (*i.e.*, the textural representation).

Depth	Tex Res	FID	Rep-Er	Pose-Er
6	64	**8.0**	0.051	9.7
4	64	8.8	0.046	9.3
2	64	8.9	**0.037**	**8.6**

4. Experiments

Combine the structural representation (feature volume code) of one with the textural (generative feature field and neural renderer code) of another.

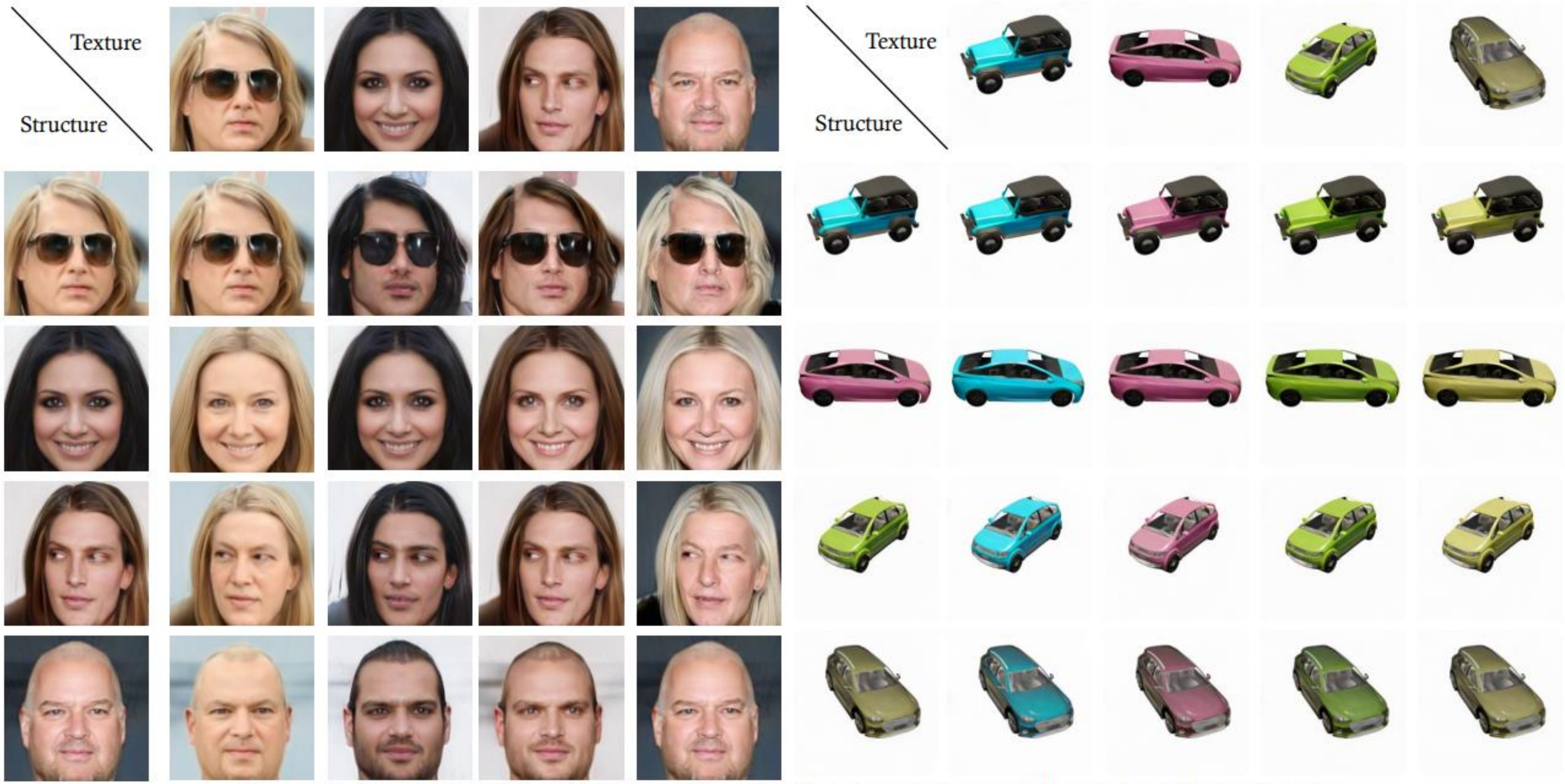


Figure 5. **Synthesized results by exchanging the structural and the textural latent codes.**

4. Experiments

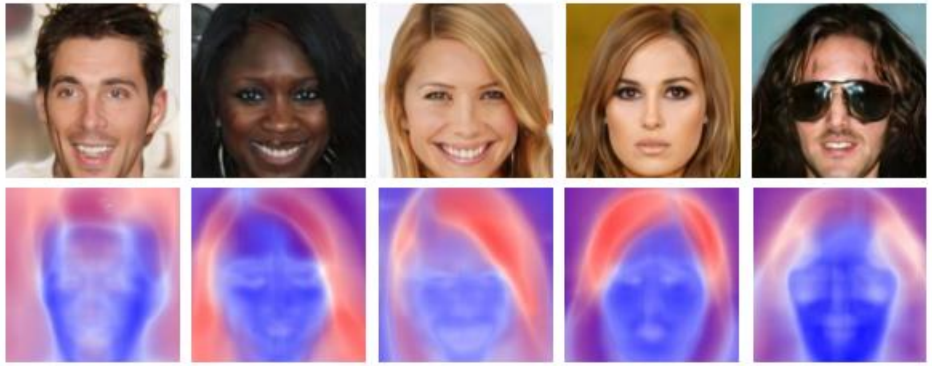


Figure 6. **Visualization of coordinate descriptor.** PCA is used to reduce the feature dimension.

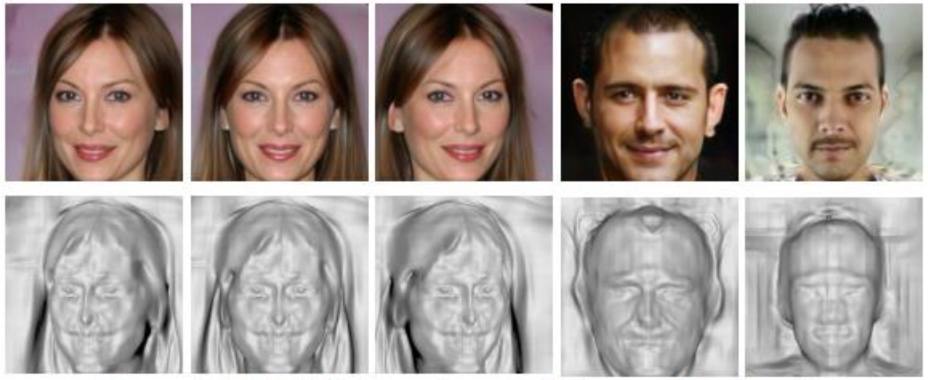


Figure 7. **3D Mesh extracted from the density.**

Accumulate coordinate descriptors on each ray, resulting in a high dimensional feature map. PCA is utilized to reduce the dimension to 3 for visualization.

5. Conclusion

Limitations

- Despite the structural representation learned by VolumeGAN, the synthesized 3D mesh surface is still not smooth and lacks fine details.

- Even though we can improve the synthesis resolution via deeper CNN, it may weaken the multi-view consistency and 3D control.

Future research will focus on generating fine-grained 3D shape as well as making the tailing CNN in VolumeGAN with improved 3D properties through introducing regularizers.