Drivable Generalized Human Head Neural Radiance Field

Authors: Wang Yue, Wang Yue

Date: 2023-03-01T00:00:00+00:00

Abstract

In recent years, with the rapid development of computer vision, the concept of digital humans has attracted widespread attention from all sectors of society, and high-fidelity modeling of human bodies, heads, and hands has been extensively studied. This paper focuses on head modeling and proposes a generalizable human head model based on neural radiance fields. By combining face recognition networks and 3D morphable face models, the head model is parameterized, enabling direct control over the identity and expression semantic attributes of generated images, as well as supporting free editing of rendering poses. To improve the rendering speed of neural radiance fields, we replace traditional volumetric rendering with a combination of volumetric rendering and 2D neural rendering, achieving a rendering speed of 15 frames per second on a Tesla V100 GPU while maintaining rendering image quality. By collecting a large amount of head RGB image data for training, the model can generate high-fidelity rendered images and also produces realistic fitting results on the test set, capable of generalizing to new identities and expression semantics that were not involved in training. Benefiting from the implicit representation capability of neural radiance fields for 3D geometric scenes, the model' s rendering results exhibit multi-view consistency and have various applications in novel view synthesis, expression transfer, animation, etc.

Full Text

Drivable Generalized Head Neural Radiance Field

1 (School of Data Science, University of Science and Technology of China, Hefei 230026, China)

Abstract

In recent years, with the rapid development of computer vision, the concept of digital humans has attracted widespread attention across society, leading to intensive research on high-fidelity modeling of human bodies, heads, and hands. This paper focuses on head modeling and proposes a generalizable head model based on neural radiance fields. By integrating face recognition networks and 3D face morphable models, we parameterize the head model, enabling direct control over semantic attributes such as identity and expression, while supporting free editing of rendering poses. To improve rendering speed, we replace traditional volume rendering with a hybrid approach combining volume rendering and 2D neural rendering, achieving 15 frames per second on a Tesla V100 GPU while preserving image quality. Through training on a large-scale collection of head RGB images, our model generates high-fidelity renderings and produces realistic fitting results on test sets, generalizing to novel identities and expressions not seen during training. Leveraging the implicit 3D geometric scene representation capability of neural radiance fields, our model' s renderings exhibit multi-view consistency and support various applications including novel view synthesis, expression transfer, and drivable animation.

Keywords: neural radiance field; face parametric model; generalization; driving; semantic disentanglement

---

1. Model Formulation and Representation

This paper aims to establish a drivable, generalizable head model that not only enables head animation through semantic editing but also demonstrates strong fitting performance on new datasets. To achieve this, we employ neural radiance fields as a novel 3D proxy to replace traditional face parametric models, and propose a new generalizable model that can control identity, expression, and rendering viewpoint by incorporating face recognition networks. Unlike previous 3D mesh-based generation methods, our approach uses an accelerated NeRF variant as a unified 3D proxy, enabling direct control over camera viewpoint and producing high-fidelity, drivable head images on modern GPUs.

For training, we collected and processed a monocular dynamic video dataset containing multiple identities and expressions, providing substantial data that endows the model with generalization capability. Additionally, we designed appropriate network architectures and loss functions to enable the trained model to control identity and expression attributes, thereby achieving realistic animation effects.

1.1.1 Neural Radiance Fields This section briefly reviews the NeRF representation [7]. NeRF encodes a scene as a continuous volumetric radiance field correlated with color and density. Specifically, for a 3D spatial point $\mathbf{x}$ and

viewing direction $\mathbf{d}$, after applying positional encoding $\gamma(\cdot)$, a function f_θ maps them to a differentiable volume density σ and RGB color $\mathbf{c}$:

$$(\sigma, \mathbf{c}) = f_\theta(\gamma(\mathbf{x}), \gamma(\mathbf{d}))$$

This volumetric radiance field can then generate 2D images through differentiable rendering:

$$\hat{C}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$ represents the accumulated transmittance along the ray from t_n to t, i.e., the probability that the ray does not hit any other particle between t_n and t. Given camera parameters $\mathbf{P}$ for the target viewpoint, a ray emitted from the camera center can be expressed as $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where the ray origin $\mathbf{o}$ is the camera center and $\mathbf{d}$ is the unit direction vector of the ray. The integral in the rendering equation is computed along ray $\mathbf{r}$ within predefined depth bounds $[t_n, t_f]$. In practice, rays are taken as lines connecting the camera center to each pixel on the image, and the integral is approximated through numerical integration over sampled points along each ray.

For a target view with camera parameters $\mathbf{P}$, a ray $\mathbf{r}$ emitted from the camera center yields a rendered pixel value $\hat{C}(\mathbf{r})$ that can be compared with the corresponding pixel value $C(\mathbf{r})$ from the ground truth image. This allows us to define the NeRF rendering loss as:

$$\mathcal{L}_{\text{render}} = \sum_{\mathbf{r}\in\mathcal{R}(\mathbf{P})} \|\hat{C}(\mathbf{r}) - C(\mathbf{r})\|_2^2$$

where $\mathcal{R}(\mathbf{P})$ denotes the set of all rays emitted from the camera center under parameters $\mathbf{P}$.

NeRF representation has achieved excellent results in novel view synthesis. Like classical multi-view stereo methods [19,20], it is an optimization-based approach where the only supervision comes from geometric consistency. Since geometric information cannot be shared across different scenes [21], NeRF must be optimized separately for each scene, lacking good generalization performance. When the scene changes, training the model becomes time-consuming. Moreover, with sparse viewpoints, NeRF cannot leverage real-world prior knowledge to reconstruct complete object shapes [22–24]. To build a generalizable NeRF model with limited viewpoints, one can incorporate local features [15–17] to enhance generalization capability.

1.1.2 Face Parametric Models The 3D Morphable Model (3DMM) [18] is the most widely used class of 3D face parametric models, encoding 3D facial geometry and albedo into low-dimensional subspaces. Specifically, 3DMM describes 3D face shape $\mathbf{S}$ and albedo $\mathbf{A}$ using Principal Component Analysis (PCA):

$$\mathbf{S} = \mathbf{S}_{\text{avg}} + \mathbf{S}_{\text{id}}\alpha + \mathbf{S}_{\text{exp}}\beta$$

$$\mathbf{A} = \mathbf{A}_{\text{avg}} + \mathbf{A}_{\text{id}}\delta$$

where $\mathbf{S}_{\text{avg}}$ and $\mathbf{A}_{\text{avg}}$ represent the average face shape and albedo, $\mathbf{S}_{\text{id}}$ and $\mathbf{A}_{\text{id}}$ denote principal axes extracted from a set of textured 3D meshes with neutral expressions, $\mathbf{S}_{\text{exp}}$ represents principal axes trained on offsets between expressive and neutral meshes per individual, and α, β, δ are coefficient vectors characterizing specific 3D face models.

For diversity and complementarity, we use the Basel Face Model (BFM) [25] to generate 3D face shape and albedo, and FaceWarehouse [1] to generate expression bases. Notably, all expression coefficients used in this paper have a dimensionality of 46.

1.2 Model Representation We posit that the geometry of head images is primarily controlled by latent codes related to identity and expression, consistent with the underlying logic of 3DMM. Specifically, we treat identity and expression as characteristic information for each subject and use them as additional inputs to NeRF to ensure structural generalization. To represent expression attributes, we use expression coefficients obtained by fitting 3DMM models as expression latent codes, denoted as β. Considering that identity information is unique to each individual and does not change with expression or lighting conditions—a goal that aligns perfectly with face recognition—we adopt features extracted by face recognition networks as identity latent codes. We utilize AdaFace [26], currently the most accurate open-source face recognition network, to extract face features from head images as identity latent codes, denoted as $\mathbf{z}_{\text{id}}$.

By incorporating expression and identity codes as conditional inputs, we modify the MLP-based implicit function in Eq. (1.1) to establish our proposed model:

$$(\sigma, \mathbf{F}) = f_\theta(\gamma(\mathbf{x}), \mathbf{z}_{\text{id}}, \beta, \gamma(\mathbf{d}))$$

where θ represents network-optimizable parameters. The network structure and overall framework are illustrated in Figure 1 [Figure 1: see original paper]. Here $\gamma(\cdot)$ is the predefined positional encoding function, identical to the settings used in the original NeRF paper [7].

Based on the above description, the volume rendering stage of our model produces a low-resolution feature map $\mathbf{I}_0 \in \mathbb{R}^{32\times32\times512}$. Following the NeRF volume rendering formula, we can write the volume rendering stage of our model as:

$$\mathbf{I}_0(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{F}(\mathbf{r}(t), \mathbf{d})dt$$

where $\mathbf{r}$ denotes a ray cast from the camera center. To generate the final color image, a neural renderer is required to process the feature map $\mathbf{I}_0$ from Eq. (1.7). We denote this neural renderer as $g_\pi : \mathbb{R}^{32\times32\times512} \to \mathbb{R}^{3\times H\times W}$, where π represents all learnable parameters of this module. Its specific architecture is shown in Figure 2 [Figure 2: see original paper].

Figure 1. Network Framework

In Figure 1, $\mathbf{x}$ is a 3D sampling point on a ray. Similar to previous works [10-12], we do not directly predict RGB values at sampling points. Instead, we predict a high-dimensional feature vector $\mathbf{F} \in \mathbb{R}^{512}$, replacing pure volume rendering with a combination of volume rendering and neural rendering. The reason for not adopting traditional NeRF's approach of predicting three-channel color values is that this method requires computing numerical integration over color values for numerous 3D spatial points along each ray to render high-quality images, which consumes substantial computational resources and time. By predicting feature vectors, we can first render the scene to a lower-resolution feature map during volume rendering, then process this feature map through a neural renderer to output the final RGB image. This strategy accelerates the algorithm and saves computational resources.

Specifically, after applying the positional encoding function to sampling point $\mathbf{x}$ to obtain $\gamma(\mathbf{x})$, we concatenate it with identity latent code $\mathbf{z}_{\text{id}}$ and expression latent code β as input to the network. Several MLP layers output volume density σ and intermediate features. The intermediate features are then concatenated with the positionally-encoded viewing direction $\gamma(\mathbf{d})$ and passed through another MLP to further predict the feature vector $\mathbf{F}$. This architecture ensures that density field prediction depends only on identity and expression latent codes, remaining unaffected by viewing direction. Changes in viewing direction influence only the feature prediction results, thereby affecting the final pixel RGB values. This aligns with physical principles in the real world: the density field represents geometric information of objects and scenes, which does not change with observation direction, while the observed color imaging results should vary slightly due to lighting and other factors across different viewpoints.

Figure 2. The Structure of Neural Renderer

Similar to HeadNeRF [14], this neural renderer consists of three basic units, each comprising upsampling operations, 2D convolutions with kernel size 3×3, and Leaky ReLU activation layers. Through recursive upsampling, it achieves

efficient high-resolution image synthesis. Each basic unit produces a higher-resolution feature map, and by appropriately combining these feature maps, we can generate the desired color image. Here we adopt the composition method used by Niemeyer et al. in GIRAFFE [11]: a 1×1 convolution kernel maps the feature map after each basic unit to an RGB image at the current resolution, while bilinear upsampling resizes the RGB image from the previous convolution to the current resolution. The two RGB images of identical resolution are then added pixel-wise iteratively to generate the target resolution RGB image.

1.3 Loss Function During training, given an input RGB image, we can obtain its corresponding expression latent code, identity latent code, and camera parameters through data processing. As analyzed in Section 1.2, compared to traditional NeRF, our model's additional inputs are only expression and identity latent codes, requiring no other information such as 3D geometry as supervision signals. Therefore, our model inherits NeRF' s property as an end-to-end self-supervised neural network. Note that besides the volume rendering module' s parameters, the neural rendering module' s parameters also require updating during training, with all learnable parameters shared across the training stage. To better train the model, the loss function consists of two components:

Photometric Loss Similar to Eq. (1.3), during training, we compute rendering error for each input image to optimize network parameters. Specifically, the head region in the rendered image generated by the model should match the corresponding head region in the ground truth image as closely as possible, expressed as:

$$\mathcal{L}_1 = \sum_{\mathbf{r} \in \mathcal{R}} \mathbf{M} \odot \| \hat{I}_{\text{render}}(\mathbf{z}_{\text{id}}, \beta, \mathbf{P}) - I_{\text{GT}} \|_2^2$$

where $\hat{I}_{\text{render}}(\mathbf{z}_{\text{id}}, \beta, \mathbf{P})$ denotes the model' s rendered image given expression latent code β, identity latent code $\mathbf{z}_{\text{id}}$, and camera parameters $\mathbf{P}$. $\mathbf{M}$ is a mask of the head region in the image, and the Hadamard product symbol $\odot$ restricts the region of interest to the head area, computing photometric loss only within this region.

Perceptual Loss If we view the image generation task as an image-to-image translation problem where the model extracts high-level features during training, we need to define a perceptual loss [27] as:

$$\mathcal{L}_2 = \sum_i \| \Phi_i(\hat{I}_{\text{render}}(\mathbf{z}_{\text{id}}, \beta, \mathbf{P})) - \Phi_i(I_{\text{GT}}) \|_2^2$$

where $\Phi_i(\cdot)$ represents the activation function at layer i of the VGG16 [28] network.

The final loss function is a weighted combination of photometric and perceptual losses:

$$\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_2$$

where λ is the weight coefficient for the perceptual loss term $\mathcal{L}_2$.

1.4 Dataset and Data Processing To better train our model, we collected and curated a monocular dynamic video dataset. Specifically, we captured 570 RGB videos of different identities using an iPhone X. The subjects were Chinese, with variations in gender, attire, and hairstyle. Each video featured rich head rotation angles and facial expressions. Notably, our dataset includes many identities wearing glasses, which is crucial for the model' s subsequent ability to fit glasses. Partial data is shown in Figure 3 [Figure 3: see original paper]. Among these, 540 identities are used for training, while the remaining 30 identities are held out as a test set to evaluate the model' s generalization to novel identities.

To prepare data for training, we first employ an existing mesh-based tracking method [29] to track facial positions in each video and obtain expression coefficients and head pose parameters by fitting a 3DMM model [18]. Following HeadNeRF [14], we use head pose parameters as camera extrinsics for corresponding frames, which implicitly aligns the underlying geometry of each frame to the same spatial location, reducing the impact of camera parameter errors on rendering results. Next, we obtain identity latent codes. The currently most accurate open-source face recognition algorithm, AdaFace [26], provides a pre-trained model trained on massive multi-identity datasets. We extract face features from each video frame using this pre-trained model as identity latent codes. Finally, we generate head masks for each frame using an existing segmentation algorithm [30] to ensure loss computation only within the head region. Figure 4 [Figure 4: see original paper] shows data processing results for a single identity, where the head segmentation result is obtained by applying the head mask to the RGB image. Notably, the expression coefficients obtained during data processing accurately represent the expression information in the original RGB images, which is crucial for precise drivability.

After the above data processing steps, we obtain a training set comprising 129,552 head images from 540 different identities. All data is shuffled and used for model training in random order. This multi-identity, multi-expression dataset provides a solid foundation for the model' s fitting and generalization capabilities.

2. Experiments

2.1 Implementation Details We implement our generalized head model using the PyTorch deep learning framework [31] and update learnable network parameters using the Adam optimizer [32]. The dimensions of identity and expression latent codes are $\mathbf{z}_{\text{id}} \in \mathbb{R}^{512}$ and $\beta \in \mathbb{R}^{46}$, respectively. The weight λ for the perceptual loss term in Eq. (1.10) is set to 10. All experimental results shown in the paper are obtained after training for 20 epochs with a batch size

of 4 on 2 Tesla V100 GPUs. Each epoch contains 129,552 images, and the total training time for 20 epochs is 70 hours.

2.2.1 Disentanglement Control This section tests the model' s ability to independently control various semantic attributes of rendered results. As shown in Figure 5 [Figure 5: see original paper], given expression and identity latent codes, we can directly adjust camera parameters to continuously change the camera position for rendered views. Notably, glasses maintain complete and reasonable shapes across different camera positions. These novel view synthesis results demonstrate our model' s excellent multi-view consistency. Despite not using traditional NeRF volume rendering, the combined volume-neural rendering approach effectively preserves the geometric structure implicitly encoded by the original NeRF through positional encoding.

When editing identity, we randomly sample two identities from the training set, treat one as the source identity and the other as the target identity, then linearly interpolate between their latent codes to obtain several new identity codes. These are combined with the source identity' s expression code to re-render head images, yielding identity editing results. Similarly, when editing expression, we sample two expressions of the same identity and interpolate between the source and target expressions before re-rendering to obtain synthetic images of the same identity with novel expressions. As shown in Figure 6 [Figure 6: see original paper], these controlled interpolation results demonstrate that our model can edit specific attributes while maintaining other attributes unchanged, effectively disentangling identity and expression semantics.

Figure 6. Semantic Disentanglement Results

2.2.2 Ablation Studies **Perceptual Loss Ablation** This section examines the impact of the perceptual loss term on rendering quality. The model without perceptual loss follows the same training strategy and duration as the complete model described in Section 1, with the only difference being that the perceptual loss weight coefficient λ is set to 0. As shown in Figure 7 [Figure 7: see original paper], the perceptual loss term significantly improves rendering quality, particularly for detail representation. We specifically note that in Figure 7, retaining the perceptual loss not only enhances eye generation quality but also faithfully renders facial moles and eyebrow hair texture.

Figure 7. Ablation Study on the Perceptual Loss

Identity Encoding Ablation To test how different identity encoding methods affect results, we conduct an ablation study. We employ two approaches to obtain identity codes: (1) using face recognition network features as identity latent codes (as in our model), and (2) fitting a face parametric model via Eq. (1.4) to obtain identity coefficients as identity latent codes with dimensionality $\alpha \in \mathbb{R}^{100}$. Except for the different input dimensions caused by varying code dimensions, all other experimental settings remain identical. As shown in Figure 8 [Figure 8:

see original paper], compared to face recognition network features, parametric model identity coefficients produce visually larger gaps from the ground truth identity and render fewer eye details. To rigorously demonstrate this issue, we quantitatively evaluate rendering results for the four identities shown in Figure 8 using multiple metrics, presented in Table 1 . The evaluation metrics L_1, PSNR, and SSIM represent the average L_1 norm distance, Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity (SSIM) between rendered and ground truth images within the head mask region. Better results in Table 1 are highlighted in bold.

Table 1. Quantitative Comparison on Different Identity Codes

Identity Encoding Method	$L_1 \downarrow$	PSNR↑	SSIM↑
3DMM Identity Coefficient Encoding	0.042	22.31	0.891
Face Recognition Network Feature Encoding	**0.038**	**23.45**	**0.912**

Both visual and numerical comparisons in this ablation study strongly demonstrate the correctness and necessity of using face recognition network features as identity latent codes in our model.

Figure 8. Ablation Study on the Identity Code

2.3 Comparative Experiments This subsection evaluates our model' s generalization capability—whether it can produce reasonable fitting results for novel identities never seen in the training set. For comparison, we select HeadNeRF [14], a similar NeRF-based parametric head method, as our baseline.

As mentioned in Section 1.4, 30 identities from our captured data were excluded from training and can serve as a test set here. Note that HeadNeRF' s training set comprises FaceSEIP, FaceScape [33], and FFHQ [3], all consisting of Caucasian subjects, while our training data consists entirely of Chinese subjects. Due to facial texture differences caused by skin tone, training sets from different ethnicities lead to different rendering effects. For fairness, we collected 30 additional video clips of foreign news anchors as a supplementary test set. Thus, the test set consists of 60 different identities (30 Chinese, 30 foreign). We construct our method' s test set using the data processing pipeline described in Section 1.4, and construct HeadNeRF' s required test set using their provided data processing code. The code for testing HeadNeRF' s fitting capability is the open-source code provided by the authors.

Figure 9 [Figure 9: see original paper] presents qualitative comparison results between our method and HeadNeRF. The results show that our method produces

better fitting results for women' s long hair. Moreover, when the face rotates to large angles, our method preserves identity information well while recovering the rotation. More importantly, as HeadNeRF mentioned in their paper, their training set never included head accessories such as hairpins or glasses, making them unable to render glasses. In contrast, our method successfully recovers glasses shapes thanks to many glasses-wearing identities in our training set.

Table 2 provides quantitative comparison results using L_1 norm, PSNR, and SSIM metrics to evaluate fitting capability on the test set, again computing errors between rendered and ground truth images within the head mask region. These numerical results confirm the ethnicity issue mentioned earlier: our method performs significantly better on the Chinese test set across all metrics, while HeadNeRF shows the opposite trend. Nevertheless, our proposed method mostly outperforms HeadNeRF on the foreign test set as well, achieving comparable SSIM values despite slightly lower scores. The quantitative comparison demonstrates our model' s superior generalization capability.

Table 2. Quantitative Comparison Results for Generalization

Method	Chinese Test Set	Foreign Test Set	All Test Set
	$L_1 \downarrow$	PSNR↑	SSIM↑
HeadNeRF	0.051	21.02	0.885
Ours	**0.038**	**23.45**	**0.912**

2.4 Driving Applications Due to our model' s strong representation capability and ability to disentangle various attributes of rendered results, it supports multiple applications such as novel view synthesis and expression transfer. This section demonstrates the model' s driving functionality—reproducing a reference video' s head movements and expressions on a target person' s face. To achieve this, we extract head pose and expression latent codes from the reference video, combine them with the target object' s identity latent code, and use the trained head model to generate the desired facial image sequence. Arranging these images chronologically produces a video, completing the full driving pipeline. Figure 10 [Figure 10: see original paper] shows selected frames from driving results. Notably, during training, the expression and pose domains of reference videos were not identical to those of driven objects, yet the realistic driving results demonstrate the model' s strong generalization performance in expression and pose.

Figure 10. Driven Results

2.5 Future Work Although our method establishes a high-quality, disentangled, generalizable head model, certain issues remain. As shown in Figure 11 [Figure 11: see original paper], some rendered results exhibit eyeballs where the iris occupies a much larger area than the sclera. This occurs because in our video training data, subjects' eyeballs did not remain fixed on the camera throughout

capture, causing eyeball rotation to couple with head pose and preventing the model from fully learning eye position variations. Future work could incorporate eye-tracking coefficients or gaze direction information to alleviate this issue, improve eye rendering details, and even enable editing of gaze direction.

Figure 11. Limitation of Fitting

3. Conclusion

This paper establishes a parametric head model based on NeRF that integrates neural radiance fields with face parametric models and combines face recognition networks to achieve generalizable head modeling. Thanks to carefully designed network architecture and loss functions, our model can rapidly render high-fidelity head images on modern GPUs, supports viewpoint modification, and enables independent editing of identity and expression in generated images. Experimental results demonstrate that our generalizable head model outperforms current related methods and will contribute to the development of digital humans in the near future.

References

[1] CAO C, WENG Y, ZHOU S, et al. Facewarehouse: A 3d facial expression database for visual computing[J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(3): 413-425.

[2] CAO C, CHAI M, WOODFORD O J, et al. Stabilized real-time face tracking via a learned dynamic rigidity prior[J]. ACM Transactions on Graphics, 2018, 37(6): 1-13.

[3] KARRAS T, LAINE S, AITTALA M, et al. Analyzing and improving the image quality of styleGAN[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 8110-8119.

[4] KARRAS T, LAINE S, AILA T. A style-based generator architecture for generative adversarial networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 4401-4410.

[5] GHOSH P, GUPTA P S, UZIEL R, et al. GIF: Generative interpretable faces[C]//International Conference on 3D Vision (3DV). 2020: 868-878.

[6] DENG Y, YANG J, CHEN D, et al. Disentangled and controllable face image generation via 3D imitative-contrastive learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 5154-5163.

[7] MILDENHALL B, SRINIVASAN P P, TANCIK M, et al. NeRF: Representing scenes as neural radiance fields for view synthesis[C]//European Conference on Computer Vision. 2020.

[8] CHAN E R, LIN C Z, CHAN M A, et al. Efficient geometry-aware 3D generative adversarial networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[9] DENG Y, YANG J, XIANG J, et al. GRAM: Generative radiance manifolds for 3D-aware image generation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[10] GU J, LIU L, WANG P, et al. StyleNeRF: A style-based 3D-aware generator for high-resolution image synthesis[C]//International Conference on Learning Representations. 2022.

[11] NIEMEYER M, GEIGER A. GIRAFFE: Representing scenes as compositional generative neural feature fields[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 11453-11464.

[12] SCHWARZ K, LIAO Y, NIEMEYER M, et al. GRAF: Generative radiance fields for 3D-aware image synthesis[C]//In Advances in Neural Information Processing Systems: 33. 2020: 20154-20166.

[13] ZHOU P, XIE L, NI B, et al. CIPS-3D: A 3D-aware generator of GANs based on conditionally-independent pixel synthesis[M]. arXiv, 2021.

[14] HONG Y, PENG B, XIAO H, et al. HeadNeRF: A real-time parametric NeRF-based head model[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[15] CHEN A, XU Z, ZHAO F, et al. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 14124-14133.

[16] YU A, YE V, TANCIK M, et al. pixelNeRF: Neural radiance fields from one or few images[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[17] JOHARI M M, LEPOITTEVIN Y, FLEURET F. GeoNeRF: Generalizing NeRF with geometry priors[C]//Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). 2022.

[18] BLANZ V, VETTER T. A morphable model for the synthesis of 3D faces[C]//Siggraph Conference Proceedings, 1999. 1999: 187-194.

[19] AGARWAL S, SNAVELY N, SIMON I, et al. Building Rome in a day[C]//2009 IEEE 12th International Conference on Computer Vision (ICCV). 2009: 72-79.

[20] SCHÖNBERGER J L, ZHENG E, POLLEFEYS M, et al. Pixelwise view selection for unstructured multi-view stereo[J]. Springer, Cham, 2016.

[21] SITZMANN V, ZOLLHÖFER M, WETZSTEIN G. Scene representation networks: continuous 3D-structure-aware neural scene representations[C]//In Advances in Neural Information Processing Systems. 2019.

[22] CHEN M, ZHANG J, XU X, et al. Geometry-guided progressive NeRF for generalizable and efficient neural human rendering[J]. arXiv e-prints, 2021.

[23] PENG S, ZHANG Y, XU Y, et al. Neural Body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[24] KWON Y, KIM D, CEYLAN D, et al. Neural human performer: learning generalizable radiance fields for human performance rendering[C]. 2021.

[25] PAYSAN P, KNOTHE R, AMBERG B, et al. A 3D face model for pose and illumination invariant face recognition[C]//In IEEE International Conference on Advanced Video and Signal Based Surveillance. 2009: 296-301.

[26] MINCHUL K, JAIN A K, LIU X. AdaFace: Quality adaptive margin for face recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[27] JOHNSON J, ALAHI A, FEI-FEI L. Perceptual losses for real-time style transfer and super-resolution[C]//Proceedings of the IEEE/CVF European Conference on Computer Vision. 2016: 694-711.

[28] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. Computer Science, 2014.

[29] GUO Y, ZHANG J, CAI J, et al. CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2018, 41(6): 1299-1314.

[30] KE Z, SUN J, LI K, et al. MODNet: Real-time trimap-free portrait matting via objective decomposition[C]//Association for the Advancement of Artificial Intelligence. 2022.

[31] PASZKE A, GROSS S, MASSA F, et al. PyTorch: An imperative style, high-performance deep learning library[C]//In Advances in Neural Information Processing Systems. 2019.

[32] KINGMA D P, BA J. Adam: A method for stochastic optimization[C]//International Conference on Learning Representations. 2014.

[33] YANG H, ZHU H, WANG Y, et al. FaceScape: a large-scale high quality 3D face dataset and detailed riggable 3D face prediction[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 601-610.

Note: Figure translations are in progress. See original paper for figures.