JUSTC

Article

http://justc.ustc.edu.cn

Received: April 22, 2023; Accepted: November 06, 2023

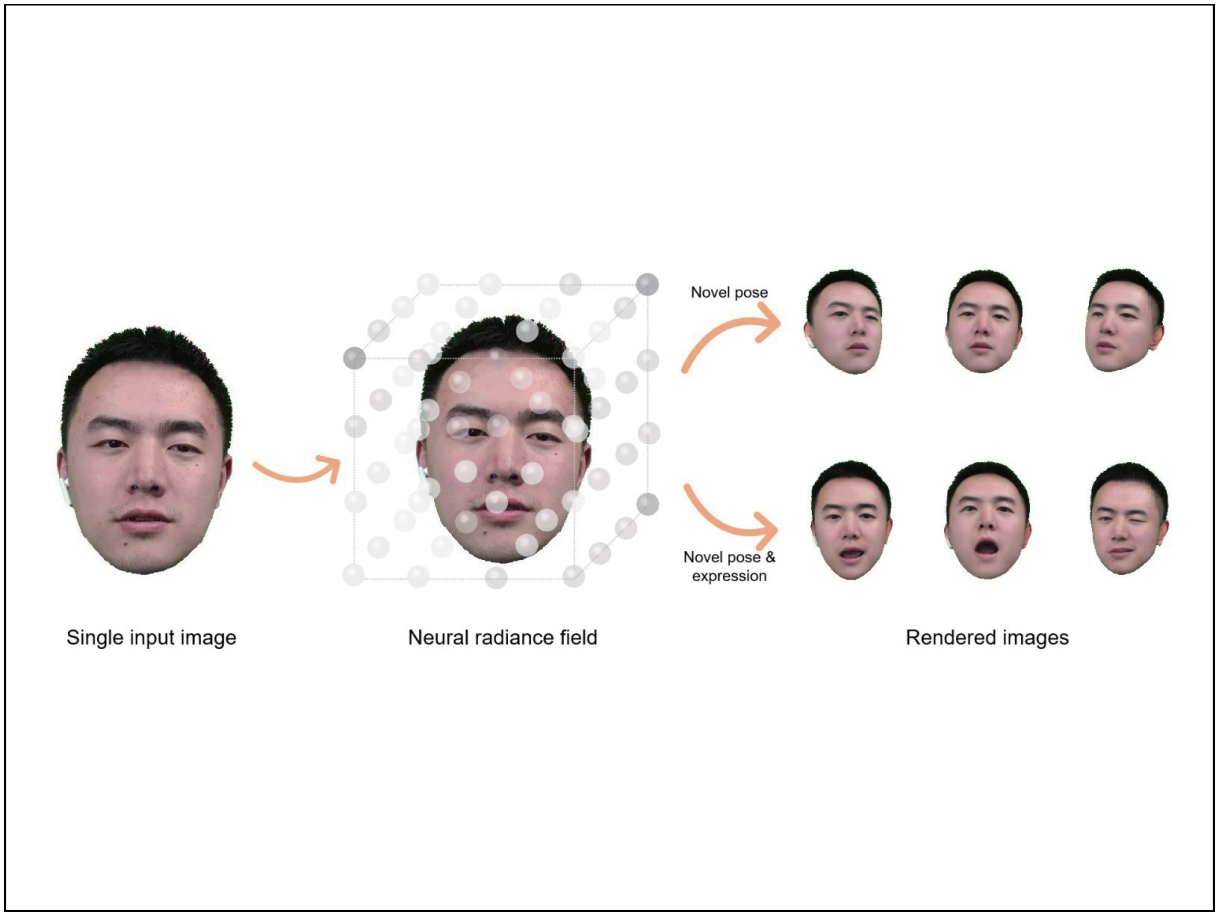
Drivable generalized NeRF-based head model

Yue Wang[1], and Yudong Guo[2] ✉

[1]*School of Data Science, University of Science and Technology of China, Hefei 230026, China;*
[2]*Image Derivative Inc., Hangzhou 311100, China*

✉Correspondence: Yudong Guo, E-mail: gyd2011@mail.ustc.edu.cn

Graphical abstract



The NeRF-based head model can generate various novel poses and expressions.

Public summary

- We propose a drivable generalized NeRF-based head model, which is capable of generating high fidelity human head images.
- By using the face recognition network to encode the identity semantic, the representation ability of our head model is improved, and it has good fitting results for new input images.
- By using the facial expression database FaceWarehouse, the model's ability to fit different expressions is improved and our expression fitting results are significantly better than other similar methods.

Citation: Wang Y, Guo Y D. Drivable generalized NeRF-based head model. *JUSTC*, **2025**, 55(1): 0104. CSTR: 32290.14.JUSTC-2023-0070. DOI: 10.52396/JUSTC-2023-0070

JUSTC

Article

http://justc.ustc.edu.cn

Received: April 22, 2023; Accepted: November 06, 2023

Drivable generalized NeRF-based head model

Yue Wang[1], and Yudong Guo[2] ✉

[1]*School of Data Science, University of Science and Technology of China, Hefei 230026, China;*
[2]*Image Derivative Inc., Hangzhou 311100, China*

✉Correspondence: Yudong Guo, E-mail: gyd2011@mail.ustc.edu.cn

Cite This: *JUSTC*, **2025**, 55(1): 0104 (12pp)

Read Online

Abstract: In recent years, the concept of digital human has attracted widespread attention from all walks of life, and the modelling of high-fidelity human bodies, heads, and hands has been intensively studied. This paper focuses on head modelling and proposes a generic head parametric model based on neural radiance fields. Specifically, we first use face recognition networks and 3D facial expression database FaceWarehouse to parameterize identity and expression semantics, respectively, and use both as conditional inputs to build a neural radiance field for the human head, thereby improving the head model's representation ability while ensuring editing capabilities for the identity and expression of the rendered results; then, through a combination of volume rendering and neural rendering, the 3D representation of the head is rapidly rendered into the 2D plane, producing a high-fidelity image of the human head. Thanks to the well-designed loss functions and good implicit representation of the neural radiance field, our model can not only edit the identity and expression independently, but also freely modify the virtual camera position of the rendering results. It has excellent multi-view consistency, and has many applications in novel view synthesis, pose driving and more.

Keywords: neural radiance fields; head parametric model; semantic disentanglement; novel view synthesis

CLC number: TP391.4 **Document code:** A

1 Introduction

In recent years, the concept of digital human has been highly discussed in the field of computer vision, with 3D face/head representation attracting much attention and its research results being widely used in augmented reality (AR)/virtual reality (VR), digital games, film production, and many other various applications. How to reconstruct the human head model in video or image with high accuracy and high fidelity is still a very challenging research topic.

Based on the assumption that the human head model can be embedded in a low-dimensional space, parametric semantic human head models, such as the blendshape model, have been studied and optimized by many scholars for a long time. Here, the blendshape model is a head model which different facial expressions are combined linearly/bi-linearly, and users can control the facial expressions by combination coefficients. What's more, this head model constructs a meaningful shape space. This enables personalized face editing. Building on this, semantic head models, such as FaceWarehouse[1], which aim to model different objects by different expressions, achieve initial generalization properties of head models, but they ignore possible geometric and textural details. In order to construct more expressive head models, traditional mesh-based approaches consider the inclusion of 3D morphable models (3DMMs)[2] based on multi-linear tensor[3], but this modelling approach usually ignores non-facial parts such as teeth and hair, and in addition, it is difficult for models to represent high-frequency details such as wrinkles due to the resolution limitations of mesh rendering. Besides, mesh rendering is not differentiable, if RGB image is the only supervision signal in the training stage, some approximation methods must be used to alleviate the non-differentiable problem.

In recent years, with the rapid development of deep learning, two-dimensional generative adversarial networks (2D GANs)[4–6] are able to generate high-quality face images directly without 3D models, bringing a great improvement to the quality of human head image generation. However, 2D GANs run only in 2D and does not explicitly model the underlying 3D scenes, so the rendering results of 2D GANs are often not multi-view consistent across different views[7]. What's more, these methods are not capable of parameter editing, even if the generated face image is more realistic, there is no way to control the details such as the facial features. Building on the advantages of 2D GANs, recent works on 3D-aware GANs[7–11] have taken image generation synthesis techniques into 3D with explicit attributes control. EG3D[7] proposed tri-plane hybrid 3D representation to model scene, which can synthesize high resolution images with multi-view consistency. Next3D[10] leveraged neural textures[12] to represent deformable facial parts to control over facial deformations and thus can be directly applied to animation tasks. But obtaining rendering for this method is time-consuming, so the training and inference are still expected to be accelerated.

In 2020, Mildenhall et al.[13] proposed to represent 3D scenes with neural radiance fields (NeRFs), a way to implicitly encode 3D space with excellent multi-view consistency. NeRF uses the volume rendering equation to render scenes

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

and naturally supports differentiable rendering, so it can be trained in an end-to-end self-supervised manner without any 3D ground truth. Because of its ability to generate high-fidelity novel view images, NeRF has quickly become a compelling research method in the field of image generation, and a large body of works on NeRF-based face/head image generation[14–16] has also emerged. Some scholars have also considered combining GAN and NeRF[17–21] to generate high-fidelity face images, but this generation model still has the same drawbacks as the base GAN model, coupling identity, expression and appearance all together, thus preventing semantic editing. To disentangle different semantic attributes, build a parametric head model, and generate a semantically editable neural radiance field of the human head, HeadNeRF[14] proposed to integrate 3DMM into a neural radiance field, and to use neural rendering to accelerate the rendering. It was trained on a large number of high-resolution single face images to ensure generalizability. However, the 3DMM[2] used by HeadNeRF is a linear parametric model that still lacks the ability to represent expression and identity information.

Based on the above observations, in order to build a generic human head model with semantic editing capabilities and improve the representation ability of the model, we propose a parametric human head model based on HeadNeRF[14] in combination with face recognition network[22] and the bi-linear expression parametric model FaceWarehouse[1]. To train this model, we collect and process a monocular dynamic video dataset containing various expressions, identities, and poses, then obtain identity and expression latent codes, and use them as conditional inputs to the NeRF structure to optimize the representation of the head model. Furthermore, like HeadNeRF[14], we combine volume rendering with 2D neural rendering, which can increase the rendering speed of NeRF several times faster during the inference stage, reaching 15 fps on a Tesla V100 GPU. Thanks to a carefully designed network structure as well as loss functions, identity and expression features, and a large amount of training data, the generalization performance of the model is guaranteed, and it successfully disentangles identity and expression, further enabling drive capability.

2 Methods

2.1 Recall on NeRF and face parametric model

2.1.1 Neural radiance field

In this section we will briefly review the NeRF representation[13]. NeRF encodes the scene as a continuous volumetric radiance field f of color and density. Specifically, for a point $\boldsymbol{x}\in\mathbb{R}^3$ and a viewpoint direction unit vector $\boldsymbol{d}\in\mathbb{R}^3$, they together are mapped to a pair, i.e., a differentiable volume density σ and an RGB color $\boldsymbol{c}$.

$$f_\theta=(\gamma(\boldsymbol{x}),\gamma(\boldsymbol{d}))\mapsto(\sigma,\boldsymbol{c}), \tag{1}$$

where $\gamma(*)$ is a position encoding function, mapping $\mathbb{R}^3$ into a higher dimensional space $\mathbb{R}^{2L}$, which can be expressed as follows.

$$\gamma(p)=(\sin(2^0\pi p),\cos(2^0\pi p),\cdots,\sin(2^{L-1}\pi p),\cos(2^{L-1}\pi p)), \tag{2}$$

where p denotes one of the coordinate values in the 3D Cartesian coordinate system. The function $\gamma(*)$ should be applied separately to each dimension of the three coordinate values in $\boldsymbol{x}$ and $\boldsymbol{d}$. It thus allows the network represented by the multilayer perceptron (MLP) weights to learn higher frequency details, which improves the network's ability.

Then 2D image can be generated from the volume radiance field by differentiable rendering using the following equation.

$$\hat{\boldsymbol{C}}(\boldsymbol{r})=\int_{t_n}^{t_f}T(t)\sigma(t)\boldsymbol{c}(t)\mathrm{d}t, \tag{3}$$

where $T(t)=\exp\left(-\int_{t_n}^{t}\sigma(s)\mathrm{d}s\right)$ denotes the cumulative transparency of the ray from t_n to t, i.e., the probability that the ray will not hit any other particle from t_n to t. $\boldsymbol{r}$ denotes a ray coming from the camera center $\boldsymbol{o}\in\mathbb{R}^3$, it can be represented as $\boldsymbol{r}(t)=\boldsymbol{o}+t\boldsymbol{d}$ where $\boldsymbol{d}\in\mathbb{R}^3$ is the unit direction vector of the ray. The integral in Eq. (3) is calculated along the ray within a given depth boundary $[t_n,\ t_f]$.

For a target view with camera parameters $\boldsymbol{P}$, a ray emitted from the center of the camera is denoted as $\boldsymbol{r}$, and the pixel values $\hat{\boldsymbol{C}}(\boldsymbol{r})$ rendered by Eq. (3) on this ray can be compared with the corresponding pixel values $\boldsymbol{C}(\boldsymbol{r})$ on the ground truth image, from which the rendering loss of NeRF can be written as follows.

$$\mathcal{L}=\sum_{\boldsymbol{r}\in\mathcal{R}(\boldsymbol{P})}\|\hat{\boldsymbol{C}}(\boldsymbol{r})-\boldsymbol{C}(\boldsymbol{r})\|_2^2, \tag{4}$$

where $\mathcal{R}(\boldsymbol{P})$ is the set of all rays emitted from the camera center when the camera parameter is $\boldsymbol{P}$.

NeRF representation has a highly desirable result in the work on novel view synthesis. Like the classical multi-view stereo methods[23, 24], it is an optimization-based approach where the only signal comes from geometric consistency. The geometric information of different scenes cannot be shared[25], so it must be optimized separately for each scene and does not have good generalization performance. When the scenes are different, it will take a lot of time to train the model. What's more, once the viewpoints are sparse, it is impossible to reconstruct the complete shape of the object using any prior in the real world[26–28]. To reconstruct a NeRF model with generalization performance under condition of limited viewpoints, local features[29–31] can be considered to enhance the generalization ability of the model.

2.1.2 Face parametric model

3D morphable model[2] is the most widely used class of 3D face parametric models, which encodes the geometry and albedo of 3D faces in space into a low-dimensional subspace. Specifically, the 3DMM describes the face geometry and albedo using principal component analysis (PCA) like following equation.

$$\begin{aligned}\boldsymbol{S}&=\overline{\boldsymbol{S}}+\boldsymbol{\alpha}\boldsymbol{A}_{\mathrm{id}}+\boldsymbol{\beta}\boldsymbol{A}_{\mathrm{exp}},\\ \boldsymbol{b}&=\overline{\boldsymbol{b}}+\boldsymbol{\delta}\boldsymbol{A}_{\mathrm{alb}},\end{aligned} \tag{5}$$

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

where $\overline{\boldsymbol{S}}$ and $\overline{\boldsymbol{b}}$ denote the shape and albedo of the mean face, respectively. $\boldsymbol{A}_{\text{id}}$, $\boldsymbol{A}_{\text{alb}}$ are the principal axes extracted from a set of 3D textured-meshes with neutral expression, $\boldsymbol{A}_{\text{exp}}$ is the principal axes trained on the offsets between each individual mesh with expressions and the mesh with neutral expressions, and the corresponding coefficient vectors, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$, characterize a particular 3D face model. For diversity and complementarity, we use the Basel face model (BFM)[32] in our data processing to generate the shape and albedo of 3D faces to obtain the camera parameters. In order to improve the ability of the model to represent expression information, we use FaceWarehouse[1] to generate the expression bases.

2.2 Network architecture

2.2.1 Implicit function

In our opinion, the head geometry is mainly encoded by the identity and expression, which is consistent with the underlying logic of 3DMM[2]. However, 3DMM as a linear model is still not powerful enough. Therefore, inspired by the generalized NeRF[14, 27–31], we use the neural radiance field as a 3D proxy for the human head. Furthermore, we replace the identity coefficients in 3DMM with identity features extracted from the head image by an face recognition network AdaFace[22], denoted as $\boldsymbol{z}_{\text{id}}$. To improve the representation of the expression information, we use the bi-linear model FaceWarehouse[1] to obtain the facial expression coefficients for each frame, denoted as $\boldsymbol{\beta}$. Like HeadNeRF[14], by using expression code and identity code as conditional inputs, the MLP-based implicit function in Eq. (1) can be rewritten as the following to build our parametric human head model.

$$f_\theta : (\gamma(\boldsymbol{x}), \gamma(\boldsymbol{d}), \boldsymbol{\beta}, \boldsymbol{z}_{\text{id}}) \mapsto (\sigma, \boldsymbol{F}), \tag{6}$$

where θ denotes the parameters optimized by network. $\gamma(*)$ is the same positional encoding function as in NeRF. The overall framework of the complete model and the structure of the implicit function f_θ are both shown in Fig. 1.

2.2.2 Volume rendering

In Fig. 1, $\boldsymbol{x} \in \mathbb{R}^3$ is a 3D point sampled on the ray. Similar to previous works[18–20], we predict a high-dimensional feature vector $\boldsymbol{F}(\boldsymbol{x})$ instead of directly predicting the RGB value $\boldsymbol{c}$. In this way, at the volume rendering stage, we can render the feature vectors $\boldsymbol{F}(\boldsymbol{x})$ into a low-resolution feature map, and then use the neural renderer to process the feature map to generate the final color image, rather than rendering the colors of the sampled points directly into the final color image, as is the case with vanilla NeRF. The reason for this is that directly generating a high-resolution color image requires that every pixel in the image has a ray passing through it, and a large number of spatial points must be sampled on each ray to approximate the volume rendering integral in Eq. (3), consuming considerable computing resources and time. In contrast, if we generate only a low-resolution feature map at the volume rendering stage, then the number of pixels that the rays must pass through is greatly reduced, so the number of numerical integrals that need to be calculated is also greatly reduced, speeding up the inference and saving computational resources. Specifically, we apply the positional encoding function $\gamma(*)$ to the sampled points $\boldsymbol{x}$, then concatenate it with the identity code $\boldsymbol{z}_{\text{id}}$ and expression code $\boldsymbol{\beta}$ as the whole input of the network, after the volume density σ is output through several layers of MLP, the positional encoding of view direction $\gamma(\boldsymbol{d})$ is fed into the network to further predict feature vector $\boldsymbol{F}(\boldsymbol{x})$. Such a network structure allows the prediction of the density field to be related only to the identity and expression codes, and is not affected by the view direction $\boldsymbol{d}$; changes in view direction only affect the prediction of the feature vector $\boldsymbol{F}(\boldsymbol{x})$, which in turn affects the final rendered image. This is consistent with the physical nature of the real world. The density field represents the geometry of the object itself, which is isotropic and does not change depending on

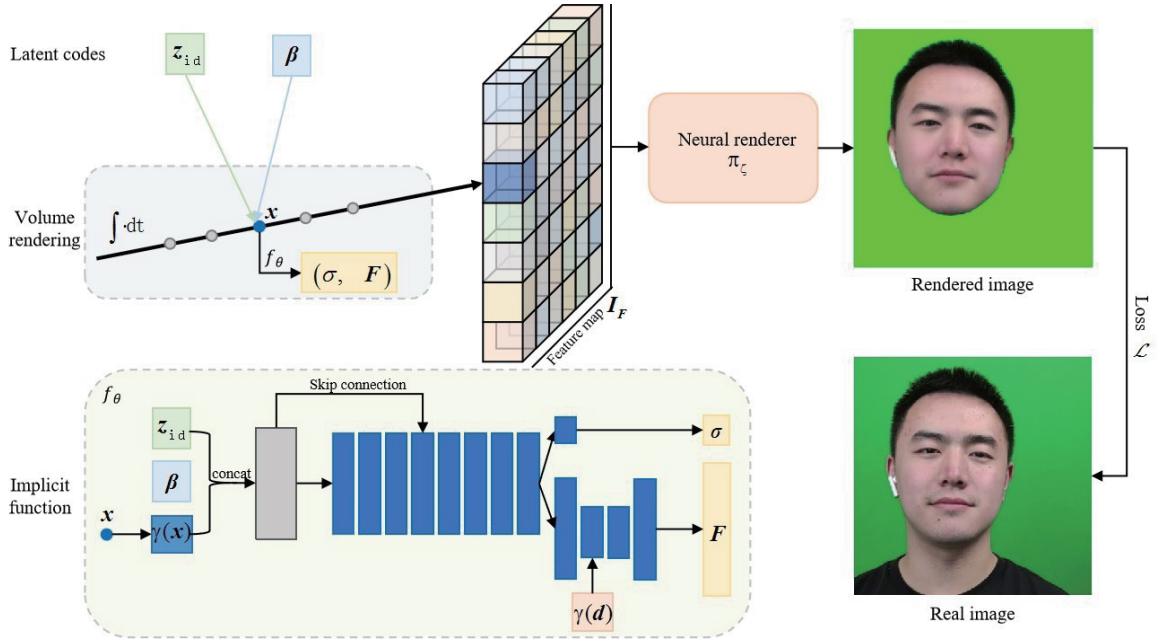


Fig. 1. Network framework.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

the view direction. The image result, on the other hand, is anisotropic and will naturally show different visual effects when viewed from different viewpoints. This is due to differences in the material and geometry of each part of the object, and differences in light propagation such as refraction and reflection of ambient light on different surfaces of the object.

According to the above description, a low-resolution feature map $\boldsymbol{I}_F \in \mathbb{R}^{512\times32\times32}$ will be generated in volume rendering stage. With reference to the rendering equation in NeRF, we can write a new equation for our model as follows.

$$\boldsymbol{I}_F(\boldsymbol{r}) = \int_0^{\infty} w(t)\cdot \boldsymbol{F}(\boldsymbol{r}(t))\,\mathrm{d}t, \tag{7}$$

where $w(t) = \exp\left(-\int_0^t \sigma(\boldsymbol{r}(s))\,\mathrm{d}s\right)\cdot\sigma(\boldsymbol{r}(t))$, $\boldsymbol{r}(t)$ denotes the ray emitted from the center of the camera and passing through one pixel of the image.

2.2.3 Neural rendering

To generate the final color image, a neural renderer is needed to process the feature map in Eq. (7).

$$\begin{aligned} &\pi_\xi : \mathbb{R}^{512\times32\times32} \to \mathbb{R}^{3\times512\times512}, \\ &\boldsymbol{I}_F \mapsto \pi_\xi(\boldsymbol{I}_F) = \boldsymbol{I}_{\text{render}}, \end{aligned} \tag{8}$$

where $\boldsymbol{I}_{\text{render}}$ denotes the final RGB image, π_ξ is the neural renderer, and ξ denotes all the learnable parameters of this module. The structure of the neural renderer is shown in Fig. 2.

Similar to GIRAFFE[19], the neural renderer consists mainly of four basic units, each consisting of an upsampling operation, a two-dimensional convolution with a convolution kernel size of 3×3, and a leaky ReLU activation function layer, which can be applied recursively to achieve efficient high-resolution image synthesis. It is worth noting that two convolution kernels are used here, one is the feature map convolution kernel, which is used in conjunction with the upsampling operation to double the resolution of the feature map from $H\times W$ to $2H\times2W$, with the aim of increasing the resolution of the feature map. The other is a color image convolution kernel which applies directly on the feature map to decode it into an RGB image of the same resolution, with the aim of mapping a multi-channel feature map into a three-channel RGB image. So that two RGB images with same resolutions are decoded from $\boldsymbol{I}_F \in \mathbb{R}^{D\times H\times W}$ in different ways, and then they are added pixel by pixel to produce the RGB image at the double resolution of $2H\times2W$. Since the initial feature map has a resolution of 32×32 and the resolution is doubled after each application of the basic unit, and the experiments in this paper all produce color images with a resolution of 512×512, it is only necessary to repeat the above steps four times to produce the final RGB image at the target resolution.

2.2.4 Loss function

Our parametric model is a NeRF-based head model that represents the head structure as an implicit radiance field related to identity and expression. The modules in the model that require optimizing parameters include the implicit function and the neural rendering module, all of which are shared during the training stage. As each step is differentiable, it inherits the advantages of NeRF as an end-to-end self-supervised neural network. In order to better optimize the network parameters, our loss function consists of the following two components.

Photometric loss. In terms of fitting the input image, we require the rendered image to be close enough to the input image. For this purpose we use photometric loss to define the difference between the rendered result and the input image.

$$\mathcal{L}_{\text{pho}} = |\mathcal{M}_{\text{h}} \oplus \mathcal{I}_{\text{render}}(\boldsymbol{\beta}, \ddagger_{\text{id}}, \mathcal{P}) - \mathcal{I}_{\text{GT}}|. \tag{9}$$

where $\boldsymbol{I}_{\text{render}}(\boldsymbol{\beta}, \boldsymbol{z}_{\text{id}}, \boldsymbol{P})$ denotes the rendered image with expression code $\boldsymbol{\beta}$, identity code $\boldsymbol{z}_{\text{id}}$ and camera parameter $\boldsymbol{P}$. $\boldsymbol{M}_{\text{h}}$ is a mask for the head region on the image, which in combination with the Hadamard product symbol $\oplus$ allows the region

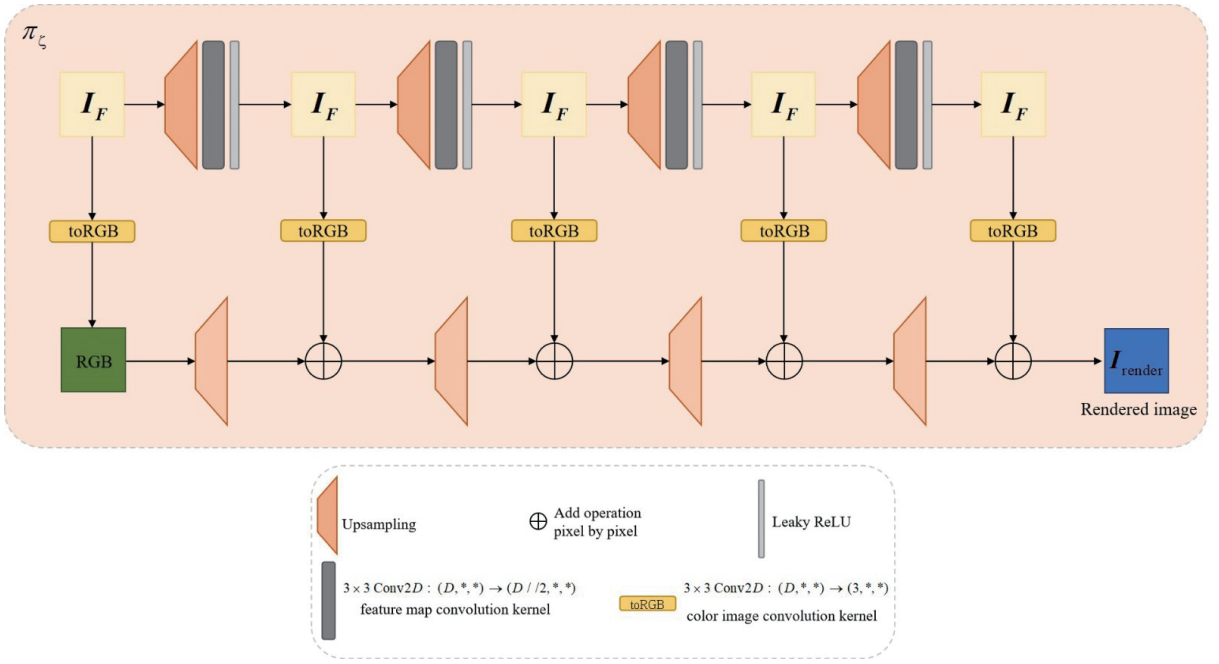


Fig. 2. The structure of neural renderer.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

of interest to be restricted to the head part and the photometric loss to be calculated only in this region.

Perceptual loss. To generate more detailed rendered images, we consider adding the perceptual loss term[33] as follows.

$$\mathcal{L}_2 = \sum_i \|\Phi(\boldsymbol{I}_{\text{render}}(\boldsymbol{\beta}, \boldsymbol{z}_{\text{id}}, \boldsymbol{P})) - \Phi(\boldsymbol{I}_{\text{GT}})\|_2^2, \tag{10}$$

where $\Phi(*)$ denotes the activation function of the ith layer in the VGG16 network[34].

The final loss function is the weighted sum of photometric loss and perceptual loss.

$$\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_2, \tag{11}$$

where λ is the weighting factor for perceptual loss term.

3 Dataset and processing

3.1 Dataset

We collected and processed a monocular dynamic video dataset. Specifically, we used an iPhone X to capture 570 RGB videos of different identities, mostly Chinese, with a 1∶1 ratio of men to women. The captured subjects differ in gender, clothing, and hairstyle. Each video contains rich head rotation angles and facial expressions of the subjects. The multi-identity, multi-expression, and multi-pose dataset can provide a solid foundation for model fitting and generalization. In particular, it should be noted that there are many subjects wearing glasses in our data, which benefits the ability of our model to fit glasses. Some of the data is shown in Fig. 3. Of these, 540 identities are used for training, while the remaining 30 are not present during training and can be a test set to evaluate the generalization ability of our parametric head model.

3.2 Data processing

Firstly, we use an existing mesh-based tracking method[35] to track the face in each video and obtained the expression coefficients and head pose for each frame by fitting 3DMM[2] and FaceWarehouse[1] expression parametric model. Like HeadNeRF[14], we view the head pose parameters as camera extrinsic parameters for the corresponding frames. This operation implicitly aligns the base geometry of each frame to the same spatial location, thus avoiding the effect of different coordinates between different data on the camera parameters.

Secondly, we need to obtain the identity latent code. AdaFace[22], the current open source face recognition algorithm with the highest accuracy, provides a pre-trained model. We use the face feature information extracted from each frame of the video by this pre-trained model as the identity code.

Finally, we obtain head mask for each frame by existing segmentation algorithm[36] to ensure that the loss function is computed only in the head area. Fig. 4 shows the results of data processing on a single identity, where the head segmentation results are obtained by the head mask acting on the RGB image. In particular, the fit results from the parametric model show that the expression coefficients obtained in the data processing stage can accurately represent the expression information of the original RGB image, which is important for the model to be accurately driven.

After the above data processing steps, we obtained a training set consisting of 129552 human head images from 540 different identities, which were all used in random order for model training. The various identities, expressions, and poses provide a solid foundation for the model's fitting ability and generalization capability.

4 Experiments and results

4.1 Implementation details

We used the PyTorch deep learning framework[37] to implement the parametric human head model built in this paper, which was trained on 2 NVIDIA Tesla V100 GPUs based on the Adam optimizer[38] to update the learnable network parameters. The batch size was set to 4. The expression code dimension is $\boldsymbol{\beta} \in \mathbb{R}^{46}$ and the identity code dimension is

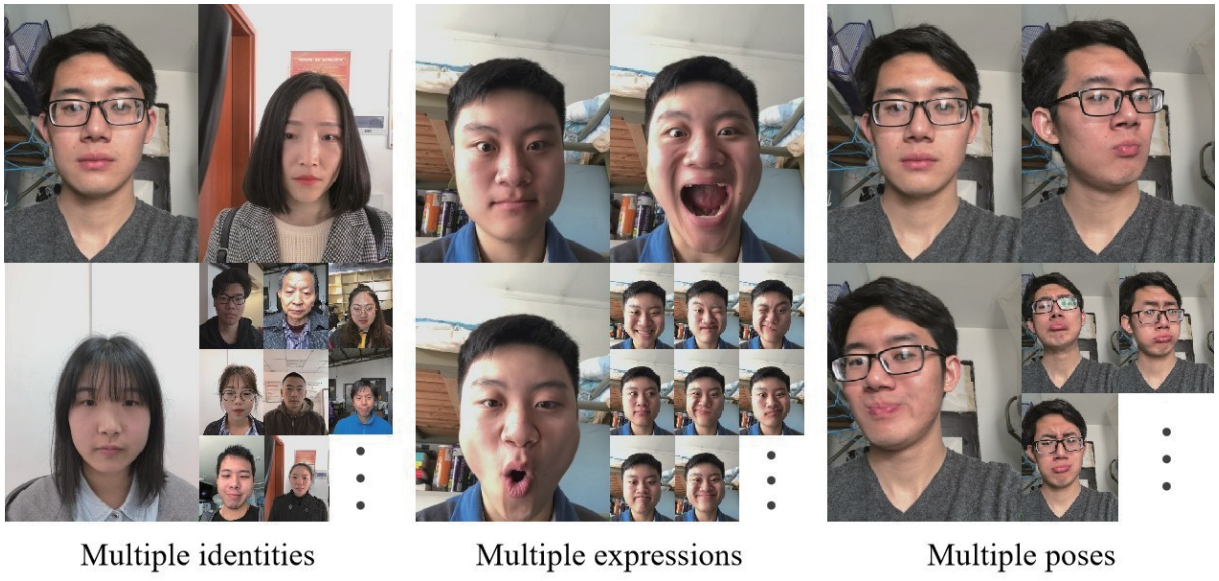


Fig. 3. Dataset.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

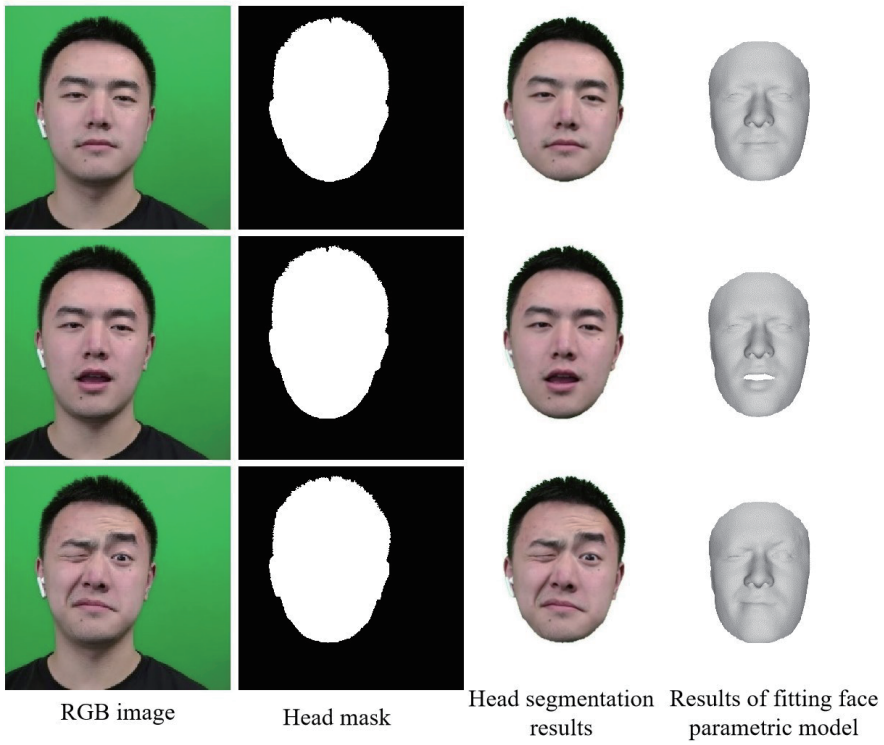


Fig. 4. Dataset processing results.

$z_{\mathrm{id}} \in \mathbb{R}^{512}$. The weight factor of the perceptual loss term in the loss function Eq. (11) is $\lambda = 10$. There were 129552 images in each epoch of training, and we consider the model to have converged after 20 epochs of training, which took a total of 70 h. Unless otherwise stated, the experimental results shown in the next section were carried out with the settings described above.

4.2 Evaluations

4.2.1 Disentangled control

In the part, we test the ability of our model to independently control different semantic attributes of the rendered results. As shown in Fig. 5, for a given expression and identity pair $(\boldsymbol{\beta}, \boldsymbol{z}_{\mathrm{id}})$, we can continuously modify the camera parameters to generate rendering images with continuously viewpoints. In particular, the glasses retain the full and sensible shape under different camera pose. These rendering results for the novel view synthesis demonstrate the excellent multi-view consistency of our model, despite not using volume rendering of the traditional NeRF, combined with a 2D neural rendering module that still effectively preserves the geometry of the NeRF via positional implicit encoding.

Furthermore, we can use the trained model to achieve semantic disentanglement and edit the identity and expression attributes independently. That is, if either identity or expression is given, smooth changes in the other attribute can be achieved while the given attribute remains unchanged. In detail, when editing identity semantics, two different identities are sampled at random from dataset, one is treated as a reference identity, while the other is treated as a target identity, and linear interpolation is performed between reference identity and target identity semantics to produce several new identity codes, each of which is concatenated with the expression code of the reference identity, and then re-render the head images with the front view to obtain identity editing results. Similarly, do the same when expression editing is required. As shown in Fig. 6, the disentanglement results show that our model is able to edit specific attributes while maintaining other attributes unchanged, effectively decoupling the semantic information of identity and expression.

4.2.2 Ablation study

To analyse the effect of certain components on experimental results, we perform ablation analysis on these components and train different models on the same training dataset for different ablation studies. We use three evaluation metrics, mean average error $\mathcal{L}_1$, peak signal to noise ratio (PSNR), and structural similarity (SSIM) to quantitatively assess the image quality of the results generated by the ablation studies. Since the training dataset consists of ~130000 images, it will be time-consuming and unnecessary to calculate these metrics on all images. Therefore, we construct a small dataset, called FaceData, with 1000 randomly selected images from the training dataset and evaluate the results of the ablation studies on this small dataset.

Ablation study on perceptual loss. In this section we test the effect of the perceptual loss on the rendering results of our model. We train the model without the perceptual loss term using exactly the same training strategy and epochs as the full model, the only difference being that the weight factor of the perceptual loss term in the loss function of the model without perceptual loss is set to $\lambda = 0$. Fig. 7 shows the qualitative results of the model without perceptual loss on the FaceData. The qualitative results show that the perceptual loss term significantly improves the quality of the generated images,

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

mainly in terms of improving details, e.g., the eye part of the full model generated image has a more detailed rendering result, and even the moles on a person's face and the hairy flu of the eyebrows are well rendered.

Furthermore, we quantitatively evaluate the ablation results of the perceptual loss term on the FaceData, results are presented in Table 1. The line that the perceptual loss is "×" and the identity is encoded as "AdaFace", is the quantitative result for the model trained without the perceptual loss term, while the last row is the quantitative result for the full model, indicating that the loss function includes the perceptual loss term. As can be seen in Table 1, the perceptual loss term leads to an improvement in the quality of the model for all three metrics.

Ablation study on identity code. At the beginning of the model construction, we use NeRF as a 3D proxy, arguing that

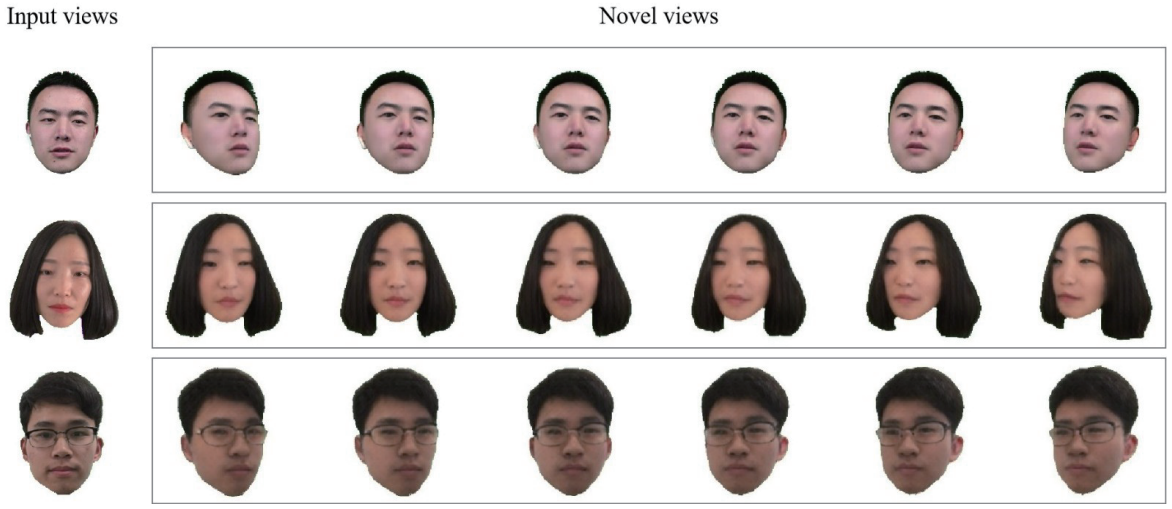


Fig. 5. Novel view synthesis.

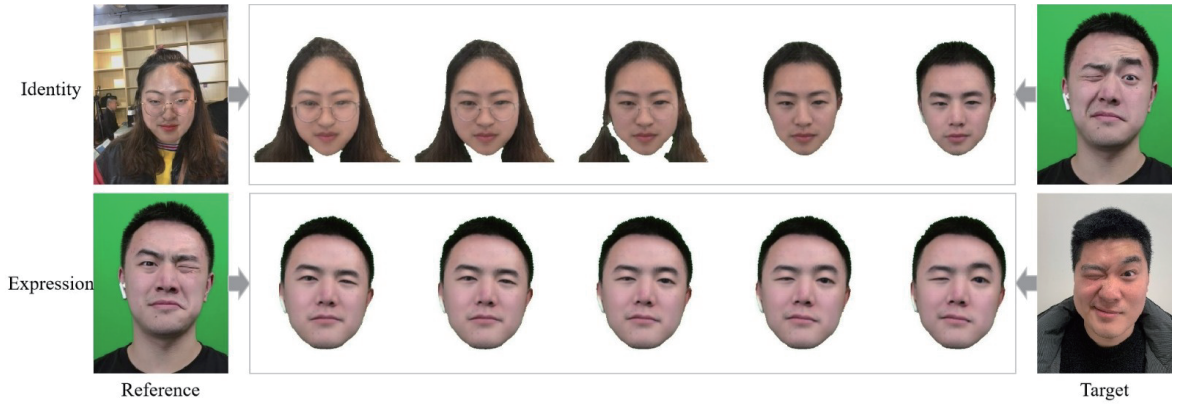


Fig. 6. Semantic disentanglement results.

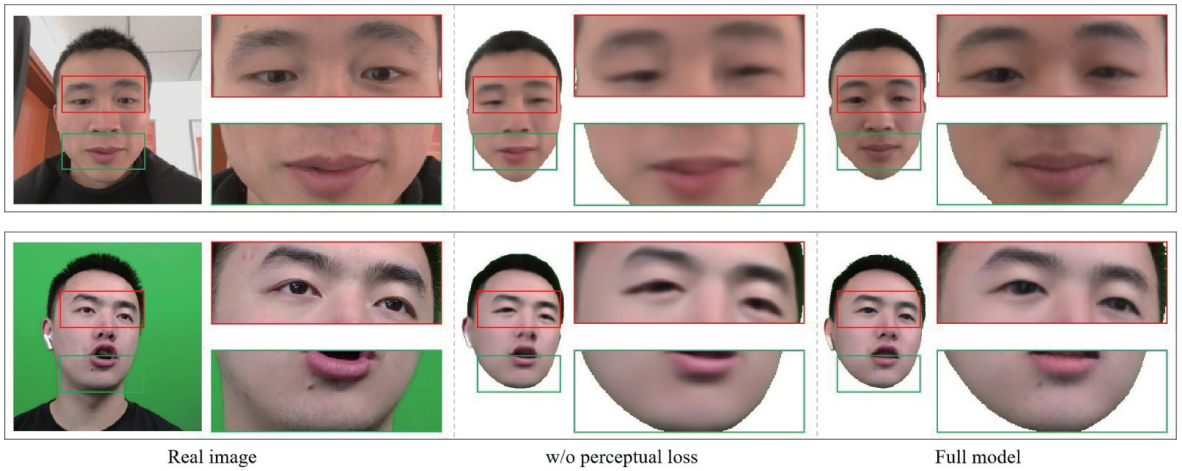


Fig. 7. Ablation study on perceptual loss.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

the geometry of human head is primarily controlled by the semantic coding related to identity and expression. Therefore, for a high quality human head model, the accurate representation of identity and expression semantics is crucial. In this section, we perform an ablation study on the effect of encoding the identity semantics in different ways. Here we use two different ways to encode the identity semantic information, one as used in the full model, where the face recognition network AdaFace[22] is used as an encoder to extract the image features as the identity latent code. The other is commonly used in face parametric semantic models, where the identity coefficient fitting the 3DMM by Eq. (5) is used to encode identity. In particular, it should be noted that the dimension of identity which is encoded by AdaFace is $\boldsymbol{z}_{\mathrm{id}} \in \mathbb{R}^{512}$, while the dimension of identity encoded by the 3DMM is $\boldsymbol{\alpha} \in \mathbb{R}^{100}$. All other experimental settings are the same except for this. Fig. 8 shows the qualitative results. As can be seen in the figure, compared to the way of using 3DMM coefficients to encode identity semantics, our model uses AdaFace, a face recognition network as the identity information encoder, the generated image is then visually closer to the identity of the real image, and the eye part is rendered with more details. Quantitative results are shown in Table 1. Both qualitative and quantitative results above strongly demonstrate the correctness and necessity of using the face recognition network as an encoder to encode identity semantic when building our human head parametric model.

Table 1. Quantitative results of the different components of our model on the FaceData.

Perceptual loss	Way of encoding identity	$\mathcal{L}_1 \downarrow$	PSNR ↑	SSIM ↑
×	AdaFace	0.075	23.9	0.940
√	3DMM	0.077	24.1	0.952
√	AdaFace	**0.073**	**24.4**	**0.955**

↓ means less is better, ↑ means more is better, and the bold font means the best results.

4.3 Comparisons

This subsection evaluates the generalization ability of our model, including fitting ability to the test dataset, and the image quality of novel view synthesis results.

4.3.1 Fitting

We compare our method with two common classes of image generation methods. One class is the NeRF-based methods,

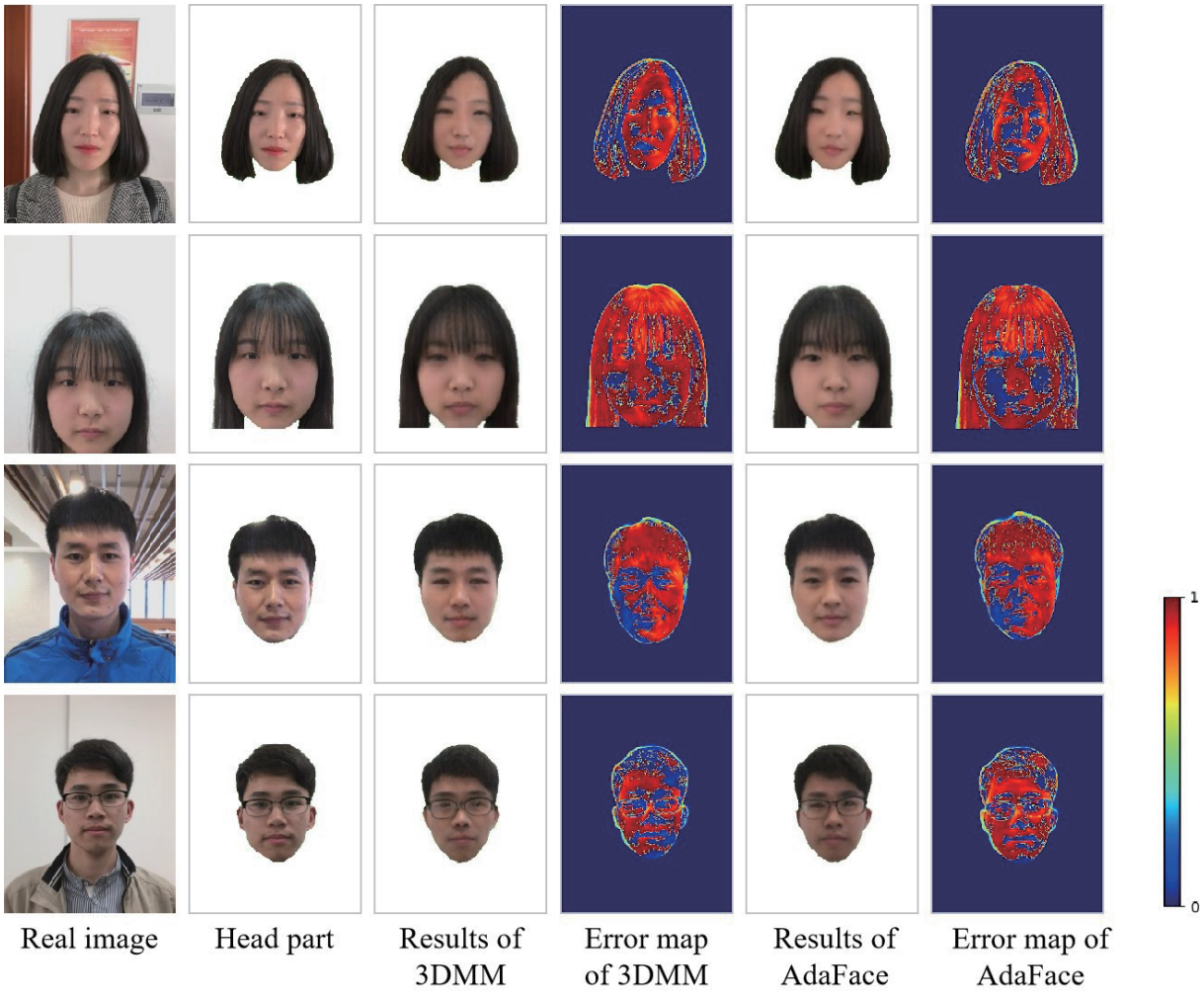


Fig. 8. Ablation study on the identity code.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

and we choose the representative GIRAFFE[19] and HeadNeRF[14]. The other class is the 3D-aware GAN-based methods, and we choose EG3D[7] and its following work Next3D[10]. GIRAFFE represents the scene as a combination of multiple objects by combining the NeRF and 2D GAN. HeadNeRF integrates 3DMM representation[2] into NeRF and establishes a semantic parametric head model. EG3D proposes the tri-plane hybrid representation to improve quality of novel view synthesis. Next3D is based on EG3D and leverages the neural texture representation to control facial deformations.

In order to better display the evaluation results, we take 30 monocular videos from our dataset that does not participate in model training at all as the test dataset. In addition, to be fair, an additional 30 news broadcast videos from different countries and identities are collected online as a supplementary test dataset. Thus the test dataset consists of 60 different videos. Due to the fact that all the test data are brand new for all methods, in order to fit each image, we need to reverse optimize the rendering process to generate the fitting results. Specifically, for GIRAFFE[19], we use the pre-trained model provided by the official and write our own fitting code to achieve the above optimization process. For HeadNeRF[14], we directly use the pre-trained model and fitting code provided by the official to fit the input image. As for EG3D[7] and Next3D[10], we slightly modified the official generation code to be suitable for fitting the input image with the officially provided pre-trained model. In order to compare these methods more comprehensively, we record the epochs and time required in their optimization process in Table 2, where GIRAFFE, HeadNeRF, and our method are all optimized 300 epochs on a Tesla V100 GPU in 30 s, 36 s, and 55 s, respectively, EG3D is optimized 1000 epochs on a 3090 GPU in 166 s, and Next3D is optimized 2000 epochs on a 3090 GPU in 504 s. The quantitative and qualitative comparison results are shown in Table 2 and Fig. 9, respectively.

Because GIRAFFE is based on 2D GAN, its latent codes don't have semantic information, and therefore can't fit the input images accurately even if the latent codes are optimized. HeadNeRF can fit the input images better during the optimization process by using 3DMM coefficients as conditional inputs, significantly improving the generation quality and visual effect of fitting results. However, HeadNeRF is limited in the representation due to the use of coefficients from the linear 3DMM model to encode semantic information, so the fitting results are still somewhat different from the input. EG3D benefits from the generative power of GAN-based approach, and after optimizing the latent codes, it is able to fit the input images well, even to the extent that it is indistinguishable to the naked eye. However, Eg3D is unable to disentangle the semantic information, and its training pressure is much less than that of our method. In addition, its novel view synthesis results is not as good as it could have been, which will be demonstrated in the subsequent content. Next3D proposes to use the neural texture representation based on EG3D, which disentangles some of the semantic information and is able to control facial deformation. Although Next3D obtains optimal results in almost metrics, the optimization process is time-consuming due to certain disentanglement difficulties inherent in the GAN-based methods, even though it is able to achieve facial control, it sacrifices the generation quality of

Table 2. Quantitative comparison with four image generation methods.

Method	$\mathcal{L}_1$ ↓	PSNR ↑	SSIM ↑	Time (s) ↓
GIRAFFE[19]	0.227	17.1	0.826	30
HeadNeRF[14]	0.110	22.7	0.925	36
EG3D[7]	0.101	29.5	0.944	166
Next3D[10]	0.025	33.9	0.963	504
Ours	0.056	27.6	0.939	55

↓ means less is better, and ↑ means more is better.

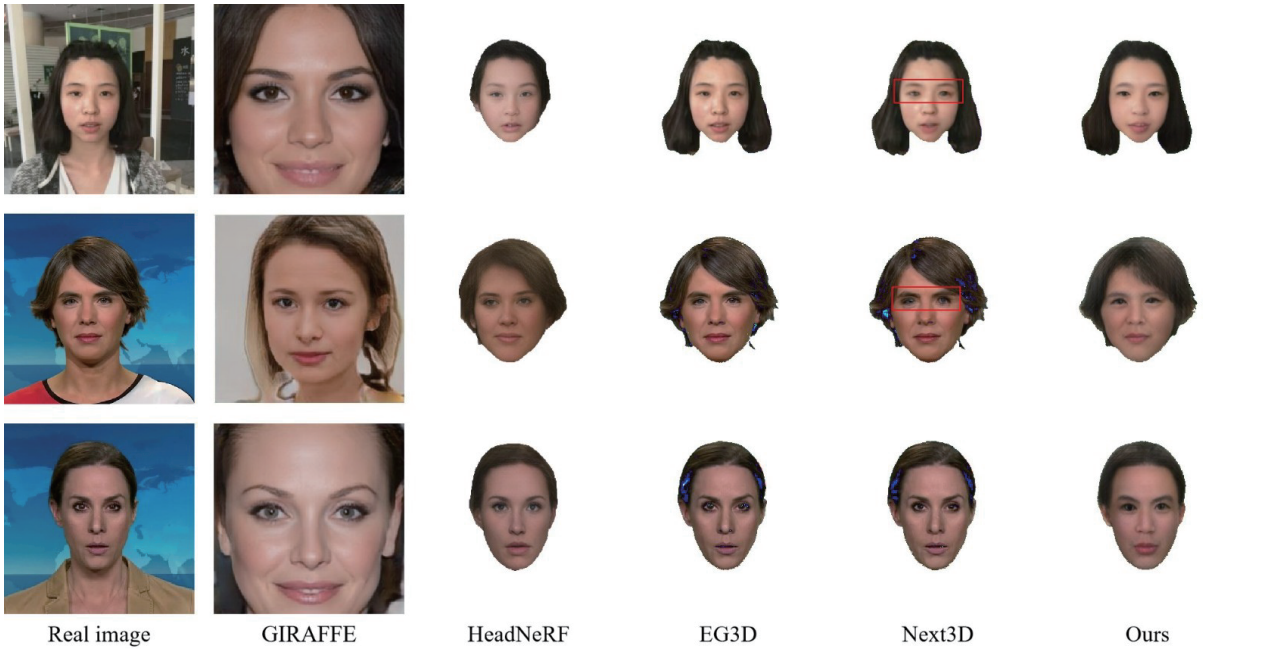


Fig. 9. Fitting results of different methods on test dataset.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

the fitting results, such as the eye part in the red box in Fig. 9, which lacks details. Our method builds on HeadNeRF, it takes much less time to optimize than Next3D because of the simplicity of the representation and the ease of disentangling in principle. Benefiting from the diversity of training data and the richness of the expression database FaceWarehouse[1], our method can not only fits the input images well, but also disentangles the identity and expression semantics.

Furthermore, to illustrate that FaceWarehouse[1], the bi-linear model used in our method, improves the representation ability of the parametric head model relative to the 3DMM[2] linear model used by HeadNeRF[14], we show the fitting results of HeadNeRF and our method for more complex expressions in Fig. 10, with the numerical value at the bottom-right corner of each fitting images indicating its PSNR. As can be seen from the figure, HeadNeRF is unable to accurately fit relatively complex facial expressions such as closing one eye, skimming, and pouting mouth. In contrast, our method better recovers these expressions due to the use of the bi-linear model FaceWarehouse, where the model's representation ability for expressions is improved, increasing HeadNeRF's PSNR value of around 22 to around 26–27.

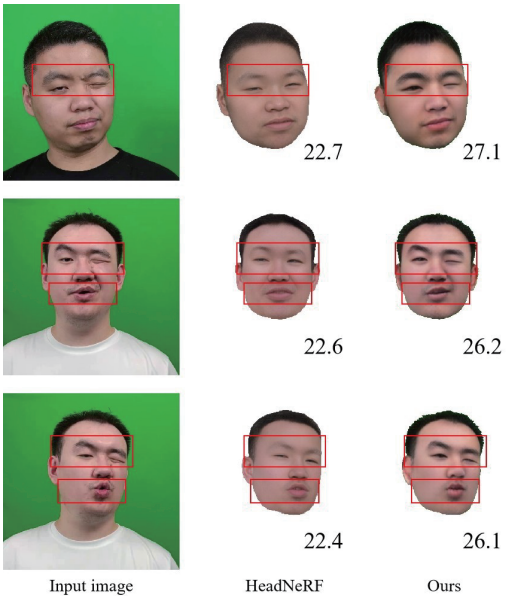


Fig. 10. Fitting results of HeadNeRF and our method for more expressions.

4.4 Novel view synthesis

Moreover, we also test the novel view synthesis results of those methods mentioned above except GIRAFFE on single input image, as shown in Fig. 11. In particular, we don't show the results of GIRAFFE here because the quality of its fitting results is relatively the worst. In Fig. 11, the first columns of each method is a direct fitting to the input image, the second and third columns are the synthesised images of the left and right new viewpoints, respectively. As can be seen from the figure, although EG3D and Next3D are able to fit the input images well, due to the lack of data from different viewpoints in the training dataset and the fact that their representations still lack 3D consistency, the missing content in the novel view synthesis results is very serious, which is completely impossible to be applied in practice. On the other hand, HeadNeRF and our method are based on the parametric model, which is equivalent to an explicit 3D geometry, inherently has multi-view consistency, and the results of novel view synthesis are more satisfactory. In addition, thanks to the diversity of the training dataset in this paper, the shape of the glasses is also well fitted.

4.5 Driving application

This article establishes a parametric head model with strong representation ability, which can disentangle various semantic

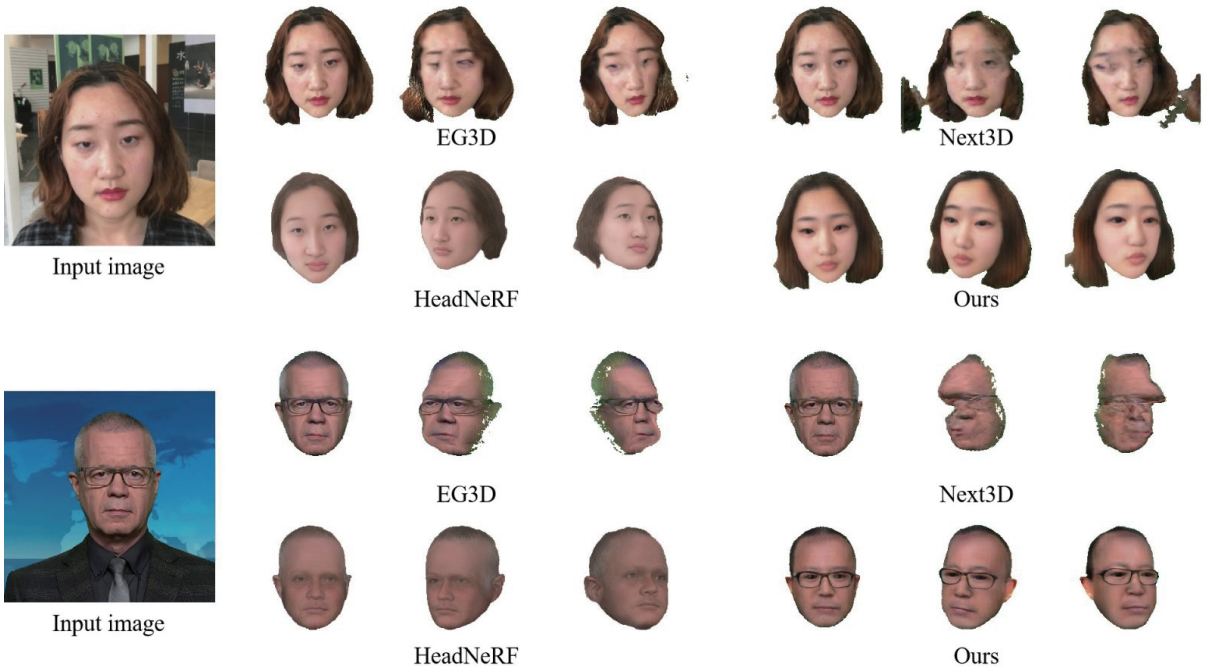


Fig. 11. Novel view synthesis results compared with different methods.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

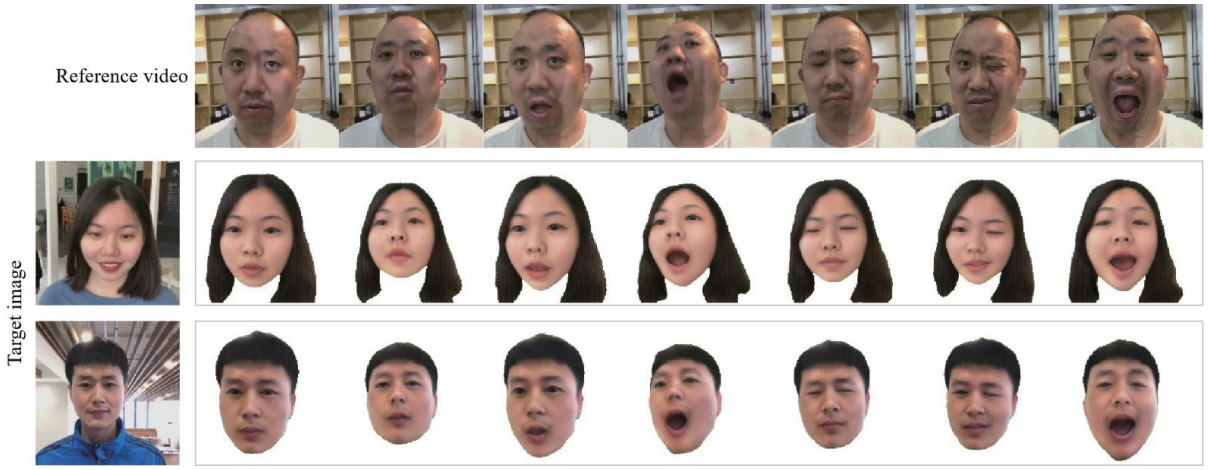


Fig. 12. Driven results.

attributes and achieve controllability of rendered content. Therefore, it has various applications in the field of image and video generation, such as facial expression transfer and novel view synthesis of single images. In this section, we will demonstrate driving applications. To achieve this, we need to use data processing to obtain head poses and facial expression latent codes from reference videos, combine it with the identity code of the target subject, use the trained head model to generate the desired facial image sequences, and then form the video in the corresponding chronological order to achieve a complete driving process. Fig. 12 shows the rendering results of some frames in the video driving. It is worth noting that the expressions and poses domains in the reference video do not completely overlap with them of the target subject, but the driving results are still coherent and natural, indicating that our model also has good fitting ability for new expression semantics and head poses.

5 Conclusions

In this paper, we propose a new head model, which takes the neural radiance field as a 3D implicit representation, and establishes a parametric model of human head. Combined with face recognition network AdaFace[22] and bi-linear expression model FaceWarehouse[1], the representation ability of our head model is further improved. Due to the large amount of different data participating in the training, the model also has good generalizability. Thanks to the well-designed network structure and loss function, this model can quickly render high fidelity head images on modern GPUs. In addition, it also supports the modification of rendering viewpoints, and can independently edit the identity and facial expressions of the generated images. The experimental results indicate that our parametric head model is superior to current relevant methods and is expected to contribute to the development of digital humans in the near future.

Conflict of interest

The authors declare that they have no conflict of interest.

Biographies

Yue Wang is now working at China Telecom. She received her Bachelor's degree from Anhui University and Master's degree from the University of Science and Technology of China. Her research mainly focuses on 3D vision.

Yudong Guo is now working at the University of Science and Technology of China (USTC). He received his Ph.D. degree from USTC. His research mainly focuses on 3D vision.

References

[1] Cao C, Weng Y, Zhou S, et al. FaceWarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, **2014**, *20*: 413–425.
[2] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques—SIGGRAPH'99. New York: ACM, **1999**: 187–194.
[3] Cao C, Chai M, Woodford O, et al. Stabilized real-time face tracking via a learned dynamic rigidity prior. *ACM Transactions on Graphics*, **2018**, *37*: 1–11.
[4] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, volume 27. New York: ACM, **2014**, *2*: 2672–2680.
[5] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA: IEEE, **2019**: 4396–4405.
[6] Karras T, Laine S, Aittala M, et al. Analyzing and improving the image quality of StyleGAN. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, **2020**: 8107–8116.
[7] Chan E R, Lin C Z, Chan M A, et al. Efficient geometry-aware 3D generative adversarial networks. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, **2022**: 16102–16112.
[8] Ghosh P, Gupta P S, Uziel R, et al. GIF: Generative interpretable faces. In: 2020 International Conference on 3D Vision (3DV). Fukuoka, Japan: IEEE, **2020**: 868–878.
[9] Deng Y, Yang J, Chen D, et al. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA: IEEE, **2020**: 5153–5162.
[10] Sun J, Wang X, Wang L, et al. Next3D: Generative neural texture rasterization for 3D-aware head avatars. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, **2023**: 20991–21002.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104

[11] An S, Xu H, Shi Y, et al. PanoHead: Geometry-aware 3D full-head synthesis in 360°. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Vancouver, BC, Canada: IEEE, **2023**: 20950–20959.
[12] Thies J, Zollhöfer M, Nießner M. Deferred neural rendering. *ACM Transactions on Graphics*, **2019**, *38*: 1–12.
[13] Mildenhall B, Srinivasan P P, Tancik M, et al. NeRF: Representing scenes as neural radiance fields for view synthesis. In: Vedaldi A, Bischof H, Brox T, et al. editors. Computer Vision–ECCV 2020. Cham: Springer, **2020**: 405–421.
[14] Hong Y, Peng B, Xiao H, et al. HeadNeRF: A realtime NeRF-based parametric head model. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, **2022**: 20342–20352.
[15] Athar S, Xu Z, Sunkavalli K, et al. RigNeRF: Fully controllable neural 3D portraits. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, **2022**: 20332–20341.
[16] Guo Y, Chen K, Liang S, et al. AD-NeRF: Audio driven neural radiance fields for talking head synthesis. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, **2021**: 5764–5774.
[17] Deng Y, Yang J, Xiang J, et al. GRAM: Generative radiance manifolds for 3D-aware image generation. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, **2022**: 10663–10673.
[18] Gu J, Liu L, Wang P, et al. StyleNeRF: a style-based 3D aware generator for high-resolution image synthesis. arXiv: 2110.08985, **2021**.
[19] Niemeyer M, Geiger A. GIRAFFE: Representing scenes as compositional generative neural feature fields. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, **2021**: 11448–11459.
[20] Schwarz K, Liao Y, Niemeyer M, et al. GRAF: Generative radiance fields for 3D-aware image synthesis. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. New York: ACM, **2020**: 20154–20166.
[21] Zhou P, Xie L, Ni B, et al. CIPS-3D: A 3D-aware generator of GANs based on conditionally-independent pixel synthesis. arXiv: 2110.09788, **2021**.
[22] Kim M, Jain A K, Liu X. AdaFace: Quality adaptive margin for face recognition. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, **2022**: 18729–18738.
[23] Agarwal S, Snavely N, Simon I, et al. Building Rome in a day. In: 2009 IEEE 12th International Conference on Computer Vision. Kyoto, Japan: IEEE, **2009**: 72–79.
[24] Schönberger J L, Zheng E, Frahm J M, et al. Pixelwise view selection for unstructured multi-view stereo. In: Leibe B, Matas J, Sebe N, et al. editors. Computer Vision–ECCV 2016. Cham: Springer, **2016**: 501–518.
[25] Sitzmann V, Zollhöfer M, Wetzstein G. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM, **2019**: 1121–1132.
[26] Chen M, Zhang J, Xu X, et al. Geometry-guided progressive NeRF for generalizable and efficient neural human rendering. In: Avidan S, Brostow G, Cissé M, et al. editors. Computer Vision–ECCV 2022. Cham: Springer, **2022**: 222–239.
[27] Peng S, Zhang Y, Xu Y, et al. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, **2021**: 9050–9059.
[28] Kwon Y, Kim D, Ceylan D, et al. Neural human performer: learning generalizable radiance fields for human performance rendering. In: Ranzato M, Beygelzimer A, Dauphin Y, et al. editors. Advances in Neural Information Processing Systems. New York: Curran Associates, Inc. **2021**: 24741–24752.
[29] Chen A, Xu Z, Zhao F, et al. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, **2021**: 14104–14113.
[30] Johari M M, Lepoittevin Y, Fleuret F. GeoNeRF: Generalizing NeRF with geometry priors. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). New Orleans, LA, USA: IEEE, **2022**: 18344–18347.
[31] Yu A, Ye V, Tancik M, et al. pixelNeRF: Neural radiance fields from one or few images. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA: IEEE, **2021**: 4576–4585.
[32] Paysan P, Knothe R, Amberg B, et al. A 3D face model for pose and illumination invariant face recognition. In: 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance. Genova, Italy: IEEE, **2009**: 296–301.
[33] Johnson J, Alahi A, Li F F. Perceptual losses for real-time style transfer and super-resolution. In: Leibe B, Matas J, Sebe N, et al. editors. Computer Vision–ECCV 2016. Cham: Springer, **2016**: 694–711.
[34] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015. San Diego, CA, USA: ICLR, **2015**.
[35] Guo Y, zhang J, Cai J, et al. CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2019**, *41*: 1294–1307.
[36] Ke Z, Sun J, Li K, et al. MODNet: Real-time trimap-free portrait matting via objective decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*, **2022**, *36*: 1140–1147.
[37] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. New York: ACM, **2019**: 8026–8037.
[38] Kingma D P, Ba J L. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015. San Diego, CA, USA: ICLR, **2015**.

CSTR: 32290.14.JUSTC-2023-0070
DOI: 10.52396/JUSTC-2023-0070
JUSTC, **2025**, 55(1): 0104