Generating Animatable 3D Cartoon Faces from Single Portraits

Chuanyu Pan · Guowei Yang · Taijiang Mu · Yu-Kun Lai

Abstract With the booming of virtual reality (VR) technology, there is a growing need for customized 3D avatars. However, traditional methods for 3D avatar modeling are either time-consuming or fail to retain similarity to the person being modeled. We present a novel framework to generate animatable 3D cartoon faces from a single portrait image. We first transfer an input real-world portrait to a stylized cartoon image with a StyleGAN. Then we propose a two-stage reconstruction method to recover the 3D cartoon face with detailed texture, which first makes a coarse estimation based on template models, and then refines the model by non-rigid deformation under landmark supervision. Finally, we propose a semantic preserving face rigging method based on manually created templates and deformation transfer. Compared with prior arts, qualitative and quantitative results show that our method achieves better accuracy, aesthetics, and similarity criteria. Furthermore, we demonstrate the capability of real-time facial animation of our 3D model.

Keywords 3D Reconstruction, Cartoon Face Reconstruction, Face Rigging, Stylized Reconstruction, Virtual Reality

1 Introduction

Virtual and augmented reality (VR/AR) has developed rapidly in recent years. An essential and challenging task in this field is to create virtual 3D faces for users and avatars. These faces should achieve high performance on aesthetics and recognizability, resembling the person being modeled. They should also be animatable for many downstream applications. However, traditional methods either require heavy manual modeling, which is time consuming, or rely on existing general templates, and thus tend to lose recognizability. With the development of deep learning techniques, a few methods [12, 10, 14] have been proposed that automatically reconstruct realistic 3D faces from images. However, due to numerous facial details, realistic 3D reconstruction becomes extremely hard to reach high similarity with the reference face. In comparison, cartoon faces are easier to reach high visual performance and can be represented with less memory. Therefore, many VR/AR applications choose 3D cartoon faces as avatars for user images.

Our work focuses on automatically creating 3D animatable cartoon faces based on a single real-world portrait. As Fig. 1 shows, we split our pipeline into the following steps: we first generate a stylized cartoon image from the input image with a StyleGAN [20]; then we reconstruct a static 3D cartoon face from the stylized image; finally, we generate semantic-preserving facial rigs to make the 3D face animatable.

Existing face reconstruction methods [12, 10] perform poorly in reconstructing cartoon faces because they introduce strong real-world priors that are hard to generalize to the cartoon domain. Some works [33] that reconstruct 3D caricatures fail to perform well on real-world portrait images due to domain gaps as well. However, to obtain accurate texture mapping and natural facial animation, precise correspondences between the reconstructed 3D face and semantic labels on the 2D image are required. These correspondences are usually acquired by projecting the model back to the image. Therefore, wrong shapes would cause wrong correspondences, highlighting the necessity for accurate reconstruction in this task.
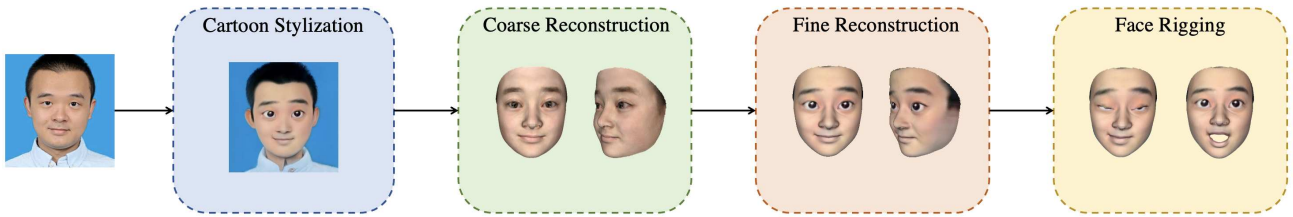
To solve this problem, we propose a two-stage reconstruction method. In the first stage, we utilize face templates and a reconstruction network to make a coarse estimation.

Chuanyu Pan
University of California Berkeley

Guowei Yang, Taijiang Mu
Tsinghua University

Yu-Kun Lai
Cardiff University

Fig. 1 The pipeline of our animatable 3D cartoon face generation method. We first transform an input portrait to a 2D cartoon image. Then we conduct template-based coarse reconstruction and deformation-based fine reconstruction to build an elaborate 3D cartoon face. Finally, we generate semantic face rigs for facial animation, making the static 3D model animatable.

In the second stage, our non-rigid deformation refinement adjusts the 3D model under the supervision of accurate 2D annotations. This refinement is not restricted to a specific domain. Some works [14,7] introduced a similar idea of adding a refinement network to adjust the 3D model. However, these works constrain the refinement on depth or normal directions. As a result, they are effective in reconstructing face details like wrinkles and moles. But for cartoon faces, which usually contain larger eyes and exaggerated expressions, these refinements are insufficient to handle. Our method conducts more general refinement, making accurate alignment without unnatural distortions. We show that our method performs well on both cartoon and real-world data.

Face rigging is the last part of our pipeline, which is the basis for facial animation. Facial animation methods [3] that use 3D morphable models (3DMM) [5] usually lack semantics, making it hard to apply them to industrial applications. Some face rigging methods [26,43] can generate semantic rigs but require user-specific training samples. Our semantic-preserving rigging method conducts deformation transfer from a set of hand-made expression models to the target. The expression models are predefined and built by professional modelers, and the rigging process is free from any reference samples.

Our work is industry-oriented, aiming to realize high-quality customized cartoon face reconstruction with real-time animation capability. Experiments show that our method outperforms prior arts on both reconstruction accuracy and user subjective evaluation. We show visualization results and an application of real-time video driven animation. In summary, our main contributions are:

1. We develop a complete system that generates a user-specific 3D cartoon face from a single portrait, which is real-time animatable. It can be directly applied to VR/AR applications such as virtual meetings and social networking for avatar customization.
2. To achieve this, we propose a two-stage 3D face reconstruction scheme that produces high-quality results on both real-world portrait images and cartoon images. Our deformation-based refinement in the second stage evi-

dently improves the performance of texture mapping and facial animation.
3. We further provide a solution for semantic-preserving face rigging without reference samples.

2 Related Work

Model-based Single Image 3D Face Reconstruction. 3D face reconstruction has been studied extensively in 3D computer vision, which is widely applied in face recognition, character generation, facial data collection, etc. The reconstructed 3D faces are usually represented as 3D meshes with a large number of vertices. To reduce the complexity of face representation, 3D Morphable Models (3DMM) [5] have been proposed for face modeling. 3DMM is a set of basis that constructs a low-dimensional subspace of 3D faces. The geometry and texture of the faces that reside in the manifold can be expressed by linear combinations of the basis. Some works [4,46,17,2] align the reconstructed face model with facial landmarks on the input image to regress 3DMM coefficients. However, these methods have difficulties capturing detailed geometry of the faces due to the landmark sparsity. Other works use features like image intensities and edges [35] to preserve facial fidelity. With the development of deep learning and differentiable rendering, some recent works [22, 19,45] use Convolutional Neural Networks (CNNs) to learn the 3DMM coefficients and pose parameters. To alleviate the lack of training data, Deng et al. [10] utilize photometric information to train CNNs in a weakly-supervised manner. All these 3DMM-based methods are facing the same problem: exaggerated shapes and geometry details can hardly be preserved due to the lack of expressivity of the low-dimensional linear models. To tackle this problem, Guo et al. [14] propose a finetuning network to recover geometry details, such as wrinkles and moles, after 3DMM coarse reconstruction. However, this method restricts the finetuning displacement to the depth direction, so still not capable of reconstructing exaggerated expressions and shapes, like large eyes and big mouth, which are quite common in cartoon images. There are also some model-free single-image reconstruction methods [16,21,15,12]. However, the results of these methods
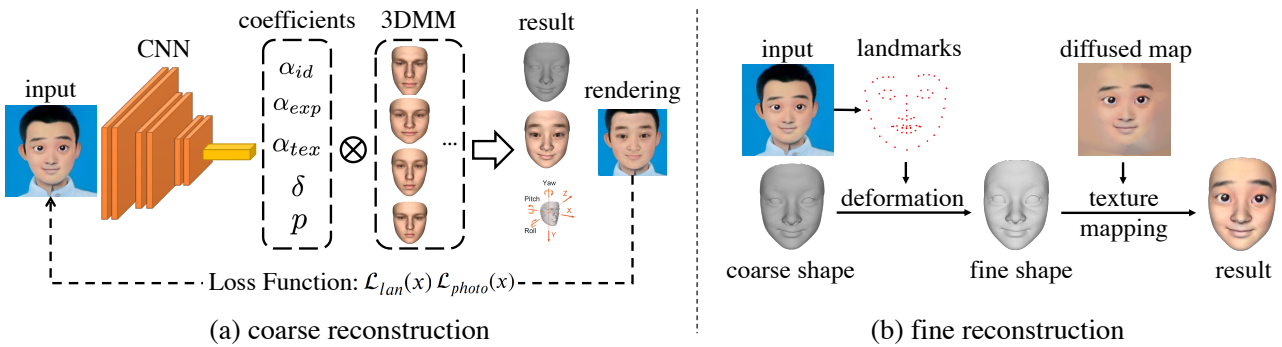
Fig. 2 Overview of our two-stage 3D cartoon face reconstruction. (a) Our coarse reconstruction method utilizes a CNN to predict 3DMM coefficients from an input image. The output coefficients contain a combination of parameters for identity α_{id}, expression α_{id}, texture α_{tex}, lighting δ, and pose p. (b) Our fine reconstruction method refines the coarse shape using landmark supervision with Laplacian deformation. The refined model is then colored by diffused texture.

are hard to be aligned or animated due to the topological inconsistency of the output meshes.

Stylized Face Reconstruction. Stylized faces usually come with larger variation of shapes and expressions, making it difficult to transfer realistic reconstruction methods to the cartoon domain directly. Liu et al. [27] represent 3D caricatures with 3DMM. Since 3DMM is low-dimensional, the reconstructed geometry varies little. Wu et al. [39] reconstruct 3D stylized faces from 2D caricature images. To address the limited deformation space of 3DMM for 3D caricatures, their approach deforms a 3D standard face by optimizing deformation gradients under the constraints of facial landmarks. The follow-up work [6] utilizes a CNN to learn the deformation gradients. These methods suffer from poor reconstruction accuracy due to the sparsity of supervision, and the large gap between the standard face and the target. Following [36], Qiu et al. [33] predict the surface of 3D caricatures with an implicit function, which is then aligned with 3DMM. However, this method requires a large amount of 3D training data, which is difficult to collect. Overall, research on reconstructing 3D stylized faces is still quite limited, and cartoon reconstruction remains to be a challenging task.

Face Rigging. Face rigging is a crucial step for 3D facial animation. By introducing 3DMM, facial expressions can be represented by linear combinations of PCA (Principal Component Analysis) basis [5,3]. Vlasic et al. [38] propose a multi-linear model to encode facial identity, expression and viseme. Synthesizing from large quantity of real-world data, these PCA models are generally built without semantics, increasing difficulty of using them to drive facial animation. To generate user-specific blendshapes for each neutral face, hand-crafted or 3D-scanned blendshape models are required [1,25]. Li et al. [26] generate facial blendshape rigs from

sparse exemplars. However, it still relies on existing well-crafted face models, and preparing exemplars for each subject is impractical. Pawaskar et al. [31] transfer a set of facial blendshapes from one identity to another; however, the topological difference between the two models could have negative impact on its performance. Some other works [13,18,8] automatically generate personalized blendshapes from video sequences or RGBD frames. Although these works achieve impressive performance, they require temporally continuous data, so are not applicable to single image reconstruction.

3 Method

As shown in Fig. 1, our pipeline can be split into three parts: stylization, reconstruction, and rigging. For stylization, existing methods like StyleGAN [20] have achieved impressive performance. Therefore, we directly apply a StyleGAN-based style transfer method [32] to generate cartoon images from real-world portraits. In this section, we will focus on our reconstruction and rigging methods.

To recover accurate geometry and detailed texture from a single cartoon image, we split the reconstruction into two stages. The first stage is to make a coarse estimation of the face geometry with CNN-based 3DMM coefficients regression. The second stage is to align the face geometry to the input image with fine-grained Laplacian deformation. The two-stage reconstruction is designed for cartoon faces with exaggerated shapes by extending the representation space of the low-dimensional 3DMM. Finally, to animate the reconstructed model, we transfer the pre-defined expression basis from the standard face to the user-specific face for semantic-preserving facial rig generation.

3.1 Model-based Coarse Reconstruction

3.1.1 Template Models: 3DMM

Expressed by 3D meshes, human faces generally consist of a large quantity of vertices and faces to show facial details. During reconstruction, directly predicting each vertex's position is a daunting and time-consuming task. However, human faces share some common geometrical features, such as the eyes and nose, making it possible to reduce the representation complexity. 3DMM [5] was then proposed to encode 3D faces into low-dimensional subspace through linear combinations of shape and texture bases:

$$\mathcal{S} = \overline{\mathcal{S}} + \alpha_{id}A_{id} + \alpha_{exp}A_{exp} \tag{1}$$

$$\mathcal{T} = \overline{\mathcal{T}} + \alpha_{tex}A_{tex} \tag{2}$$

where $\overline{\mathcal{S}}$ and $\overline{\mathcal{T}}$ represent the shape and texture of the standard face. A_{id}, A_{exp}, and A_{tex} are 3DMM bases for identity, expression, and texture respectively. These bases are extracted and synthesized from a large amount of real facial scans. α_{id}, α_{exp}, and α_{tex} are combination coefficients of the bases. $\mathcal{S}$ and $\mathcal{T}$ are the shape and texture of a 3D face. Our model-based reconstruction utilizes 3DMM to make a coarse estimation of the face geometry due to its expressiveness and simplicity.

3.1.2 Coarse 3D Cartoon Face Reconstruction

Based on previous CNN-based methods [14,10], we utilize a CNN to predict 3DMM coefficients. As Fig. 2(a) shows, the network takes a 2D cartoon image as input, and predicts a vector of coefficients $x = (\alpha_{id}, \alpha_{exp}, \alpha_{tex}, \delta, p)$. The 3D face pose p in the world coordinate system is defined as a rigid body transformation with rotation $R \in SO(3)$ and translation $t \in \mathbb{R}^3$. δ is the Sphere Harmonic (SH) coefficients to estimate the global illumination of a Lambertian surface on each vertex as $\Phi(n_i, b_i|\delta) = b_i \cdot \sum_{k=1}^{B^2} \delta_k \phi_k(n_i)$, where human faces are assumed to be Lambertian surfaces [14, 34], $\phi_k : \mathbb{R}^3 \to \mathbb{R}$ represents SH basis functions ($1 \leq k \leq B^2$), and $\Phi(n_i, b_i|\delta)$ computes the irradiation of a vertex with normal n_i and scalar albedo b_i. Applying these coefficients to 3DMM gives the reconstructed 3D face.

To train the network, we first render the face image from the predicted 3D face model at pose p and lighting approximation δ using differential rendering [24] techniques. The rendered image I_{render} is then compared with the input image I_{in} to calculate the loss.

Specifically, the loss function consists of three parts:

$$\mathcal{L}(x) = \omega_l \mathcal{L}_{lan}(x) + \omega_p \mathcal{L}_{photo}(x) + \omega_r \mathcal{L}_{reg}(x) \tag{3}$$

The first part is landmark loss:

$$\mathcal{L}_{lan}(x) = \frac{1}{N} \sum_{n=1}^{N} \omega_n \|q_n - \Pi(R\mathbf{p_n} + t)\|^2 \tag{4}$$

where $q_n \in \mathbb{R}^2$ is the true position of the nth 2D facial landmark on the original image, $\mathbf{p_n} \in \mathbb{R}^3$ is the nth 3D facial landmark on the face mesh, which is pre-defined by 3DMM. Note that 3DMM base models share identical topology, and the related vertices on each base model have the same semantics. Therefore, the 3D landmarks could be defined as certain vertices on the mesh. N is the number of landmarks, ω_n is the loss weight for each landmark, R and t denote the rotation and the transformation of the pose p respectively. $\Pi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ is the orthogonal projection matrix from 3D to 2D. The second part is photometric loss:

$$\mathcal{L}_{photo}(x) = \frac{1}{|\mathcal{A}_m|} \|\mathcal{A}_m \cdot (I_{render} - I_{in})\|^2 \tag{5}$$

which calculates the color difference between I_{render} and I_{in} per pixel. $\mathcal{A}_m$, acquired by face parsing [40], is the confidence map that evaluates whether an image pixel belongs to a human face. This strategy helps improve robustness in low-confidence areas, like glasses or beards. Compared to the landmark loss, the photometric loss constrains the reconstructed texture and geometry at a fine-grained level. The final part is regularization loss on 3DMM coefficients to avoid getting far from the standard face:

$$\mathcal{L}_{reg}(x) = \omega_{id}\|\alpha_{id}\|^2 + \omega_{exp}\|\alpha_{exp}\|^2 + \omega_{tex}\|\alpha_{tex}\|^2 \tag{6}$$

3.1.3 Training with Cartoon Data

Most CNN-based methods train their reconstruction network with normal face images. However, domain gaps exist between real and cartoon faces. To solve this problem, we propose a cartoon face dataset with landmark labels for network training.

Cartoon face images are not as common as real-world images. To gather a large amount of cartoon data, we utilize a pre-trained StyleGAN [20] for cartoon face generation. Specifically, a StyleGAN is trained on a set of cartoon face images collected from the internet. Then we randomly sample latent codes from the input latent space $\mathcal{Z}$, forward them to the StyleGAN and get the cartoon face images. To ensure a clear face appears on each image, we filter out images where face detection confidence is lower than a threshold ϵ using a face detector [41].

Fig. 3 shows some examples of our cartoon dataset, which contains 73852 images at the resolution of 1024×1024. The faces of different colors and ages are uniformly distributed in the dataset to minimize the bias caused by the data distribution. For each image, to calculate the landmark loss (Eq. 4), 68 landmarks are labeled by a landmark detector [23]; see section 3.2.1 for more details.

In addition, we use the same StyleGAN structure with a "layer swapping" interpolation scheme [32] to stylize users' real-world portraits. These images then become the input
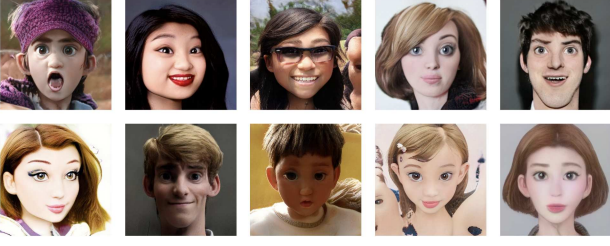
Fig. 3 Examples of our cartoon training dataset. For each sample, we ensure that a clear face exists through a face detector and apply annotations of 68 facial landmarks through a landmark detector.

of the coarse reconstruction process in our complete application pipeline. The size of the stylized image is currently fixed in this work. However, works of image enhancement [42,29] have shown the potential to increase the size and resolution of the image. Thus the image size won't be a limitation of this work.

3.2 Deformation-based Fine Reconstruction

Although using 3DMM for coarse reconstruction yields accurate results on the overall shape of the face, we find it fails to recover some fine face structures, especially the eyes. The low-dimensional parametric face model lacks expressivity for exaggerated facial parts, which are common in cartoon portraits. These reconstruction errors cannot be ignored because even a tiny misalignment would significantly affect the model appearance and facial animation.

To tackle this issue, we introduce deformation-based fine reconstruction. As Fig. 2(b) shows, we align the 3D reconstructed face to the 2D landmarks on the input image with non-rigid deformation. We minimize the misalignment with accurate landmark supervision and a local deformation method. We show that our facial alignment strategy significantly improves texture mapping performance.

3.2.1 Cartoon Face 2D Landmark Annotation

Accurate 2D landmark annotation is crucial to the alignment. We observe that significant misalignment appears in the eye areas after projecting the predicted 3D face to the image space. Some mainstream 68-landmark detectors [23], which are trained on ordinary face images, could provide landmark annotations on the image. However, the annotation is not accurate on cartoon images, especially in the eye areas, because of the domain gap. To solve this problem, we combine landmark detection with a state-of-the-art pixel-level face parsing method [40]. We first obtain the prediction of 68 facial landmarks from the detectors and acquire the face parsing result, which contains eye segmentation. Then, for each eye landmark, we snap its position to the

nearest point on the boundary of the segmented eye area if the boundary exists. Utilizing color clues, we set the eye landmarks to lie on the border of the eye.

3.2.2 Face Alignment with Laplacian Deformation

An intuitive way to align a 3D face with 2D landmark labels is to optimize the 3DMM coefficients by minimizing the distance between the projected 3D landmarks and the 2D labels:

$$\alpha_{id}^*, \alpha_{exp}^* = \arg\min_{\alpha_{id},\alpha_{exp}} \sum_{n=1}^{N} \omega_n \|q_n - \Pi(R\mathbf{p_n} + t)\| \tag{7}$$

$$\mathbf{p_n} = K(\bar{\mathcal{S}} + \alpha_{id}A_{id} + \alpha_{exp}A_{exp}; n) \tag{8}$$

where q_n, $\mathbf{p_n}$, Π, and (R, t) share the same definition as Eq. 4. $K(\mathcal{S}; n) \in \mathbb{R}^3$ is to get the nth 3D landmark position on shape $\mathcal{S}$. However, adjusting 3DMM coefficients in this way will cause distortion and unnatural folds on the face due to the global nature and geometric restrictions of the template models, which will be demonstrated in the experiments (section 4.2.2).

We instead exploit Laplacian deformation [44] to align the landmarks accurately and locally without affecting the overall shape. The deformation is driven by anchors, which are landmarks in this context. The goal is to preserve the local normal of each vertex on the mesh as much as possible while moving the anchors. Specifically, the Laplacian coordinates of vertex $\mathbf{v_i}$ are defined as:

$$L(\mathbf{v_i}) = \frac{1}{|\mathcal{N}(\mathbf{v_i})|} \sum_{\mathbf{v_j} \in \mathcal{N}(\mathbf{v_i})} (\mathbf{v_i} - \mathbf{v_j}) \tag{9}$$

where $\mathcal{N}(\mathbf{v_i})$ is the set of vertices that share common edges with $\mathbf{v_i}$ (i.e., 1-ring neighboring vertices). Preserving $L(\mathbf{v_i})$ during deformation imposes a constraint on local geometry that prevents unnatural distortions. Meanwhile, to be driven by anchors, corresponding vertices should follow the anchors and stay close. Therefore, the objective function to be minimized is:

$$\min_{\mathbf{v} \in V} \left(\sum_{i=1}^{|V|} \|L(\mathbf{v_i}) - L_i'\|^2 + \lambda \sum_{k \in M} \|\mathbf{v_k} - \mathbf{p_k}\|^2 \right) \tag{10}$$

where L_i' is the initial value of $L(\mathbf{v_i})$, M is the set of vertex indices for 3D landmarks on the mesh, and $\mathbf{v_k} \in \mathbb{R}^3$ is a 3D landmark position, and $\mathbf{p_k} \in \mathbb{R}^3$ is the corresponding ground truth 3D position. Transforming 2D landmark supervision q_n to 3D anchors $\mathbf{p_k}$ requires depth information. We use the depth value of the initial 3D landmark vertex $\mathbf{v_k}$ as an approximation of $\mathbf{p_k}$'s:

$$d_{cam} - (R\mathbf{p_k} + t)\big|_z = d_{cam} - (R\mathbf{v_k} + t)\big|_z \tag{11}$$

where d_{cam} is the depth of the camera center, (R, t) is a rigid body transformation to the camera coordinate system.
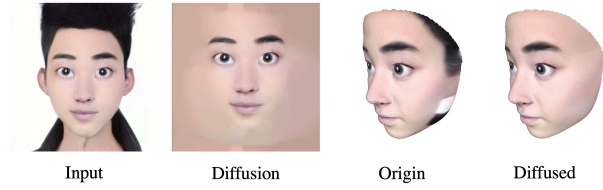
Fig. 4 Texture diffusion (left 2). Comparison of the results without (Origin) and with (Diffused) diffused texture (right 2).

3.2.3 Texture Mapping

Texture plays a decisive role in improving the visual quality of the reconstructed model. The texture we acquired from coarse reconstruction is a combination of 3DMM texture basis, which is too rough to express an elaborate cartoon face. Therefore, to maximize the similarity of the model with the input cartoon image, we project each vertex to the image with the transformation (R, t) predicted in the coarse reconstruction stage. The normalized 2D projected position is then used as the texture coordinates of the vertex:

$$\text{tex\_coord}(\mathbf{v}) = \text{Norm}(\Pi(R\mathbf{v} + t)) \tag{12}$$

Diffused Texture. Due to tiny reconstruction inaccuracy, some background pixels might be mistakenly mapped as part of the face texture. This error will be amplified on the 3D model, as Fig. 4 (Origin) shows. To tackle this problem, we first segment the cartoon face from the background with face parsing [40], and then replace the background with diffusion of the face color, as Fig. 4 (Diffusion) shows. Each background pixel is traversed by a Breadth-First-Search, and its color is replaced with the average color of the surrounding visited pixels. The processed image is then used for texture mapping.

3.3 Semantic-preserving Facial Rig Generation

Animating a static 3D cartoon face requires additional action guidance. Motivated by 3DMM, we utilize a template-based method for facial animation:

$$S^* = S_0 + B_{exp}\beta \tag{13}$$

where S_0 is the neutral 3D face, and B_{exp} is the expression basis. Controlled by coefficients β, the output face S^* changes expression accordingly. Normally, the expression components of 3DMM basis lack semantics and are mutually coupled, making it difficult to control each part of the face independently. Inspired by FACS [11], we manually construct a set of standard face models $\{S_i\}, i = 1, 2, 3..., m$, each of which represents a specific movement of a single face part, such as 'left eye close' and 'mouth open.' Then
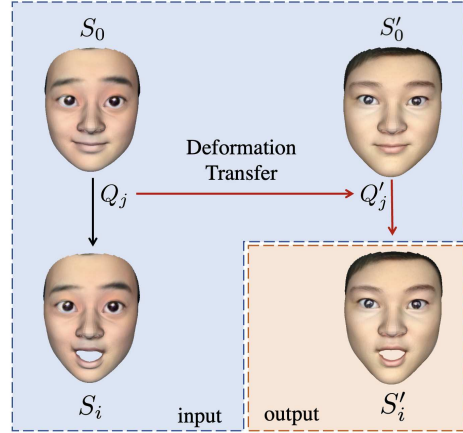
Fig. 5 Expression Transfer. The deformation from S_0 to S_i (Q_j for face f_j) is transferred to the deformation from S_0' to S_i' (Q_j' for face f_j). By generating new expression models S_i' using the subject-specific expression models, S_0' can be animated.

we have $B_{exp} = (S_1 - S_0, S_2 - S_0, ..., S_m - S_0), \beta = (\beta_1, \beta_2, ..., \beta_m)$. Generally, β_i ranges from 0 to 1.

However, directly applying the standard expression models $\{S_i\}$ to an arbitrary neutral face will cause unnatural expressions, due to differences of facial shapes. So we utilize deformation transfer [37] to generate user-specific face rig. As Fig. 5 shows, the deformation from S_0 to S_i is transferred to adapt to the newly reconstructed S_0' and generate S_i'. The expression transfer is based on the geometric relations between the standard neutral face S_0, S_i and S_0'.

As to the deformation from S_0 to S_i, since they are topologically consistent, vertices and faces between them correspond to each other. For a triangular face f_j, suppose $\mathbf{v_i}$ and $\tilde{\mathbf{v}_i}$, $i = 1, 2, 3$, are undeformed and deformed vertices of f_j respectively. To include normal information, [37] introduces the fourth vertex $\mathbf{v_4}$ in the direction perpendicular to f_j with a unit distance as:

$$\mathbf{v_4} = \mathbf{v_1} + \frac{(\mathbf{v_2} - \mathbf{v_1}) \times (\mathbf{v_3} - \mathbf{v_1})}{\sqrt{|(\mathbf{v_2} - \mathbf{v_1}) \times (\mathbf{v_3} - \mathbf{v_1})|}} \tag{14}$$

Then the deformation of f_j can be described with a 3×3 matrix Q_j and a translation vector t_j as:

$$\tilde{\mathbf{v}_i} = Q_j\mathbf{v_i} + t_j, i = 1, 2, 3, 4 \tag{15}$$

As to the transformation from S_i to S_i', the goal is to preserve Q_j's:

$$\min_{\tilde{\mathbf{v}}_1', ..., \tilde{\mathbf{v}}_n'} \sum_{j=1}^{m} \|Q_j - Q_j'\| \tag{16}$$

where Q_j is the transformation matrix of the jth triangular face on mesh from S_0 to S_i, Q_j' is for S_0' to S_i'; m is the number of faces and $\{\tilde{\mathbf{v}}_1', ..., \tilde{\mathbf{v}}_n'\}$ are vertices of S_i'. t_j remains unchanged when transferred to S_0'.

We now can obtain the expression models $\{S_i'\}$ for the newly reconstructed model by applying the above expression transfer to each $\{S_i\}$. Then the 3D face can be animated in real-time driven by the input coefficients β.

4 Experiments

4.1 Setup

Implementation Details. We implement the coarse reconstruction network using the PyTorch framework [30]. The network takes a stylized face image with size $224 \times 224 \times 3$ as input, and outputs a coefficient vector $x \in \mathbb{R}^{239}$, with $\alpha_{id} \in \mathbb{R}^{80}$, $\alpha_{exp} \in \mathbb{R}^{64}$, $\alpha_{tex} \in \mathbb{R}^9$, $\delta \in \mathbb{R}^6$ respectively. In our experiment, we set the weights to $\omega_{id} = 1.2$, $\omega_{exp} = 1.0$, $\omega_{tex} = 1.2e - 3$, $\omega_l = 2e - 3$, $\omega_p = 2.0$, $\omega_r = 3e - 4$. Similar to [10], we use a ResNet-50 network as the backbone followed by a fully-connected layer to regress the coefficients. For the fine reconstruction stage, the optimization problem in Eq. 10 can be transformed into a linear equation by the least squares method. We solve the linear equation with sparse matrices and Cholesky decomposition. The same processing is applied to the expression transfer optimization problem in Eq. 16 in facial rig generation. Our manually constructed standard expression models are built on blender [9] by professional modelers, containing 46 different expressions defined by FACS [11].

Data Collection. As introduced in section 3.1.3, we built a training dataset with 73852 cartoon face images for coarse reconstruction training. For testing data, we collect real-world portraits and stylize them using a pretrained StyleGAN [20]. We then annotate 68 facial landmarks for each stylized cartoon image with the landmark detector[23] and manually adjust their positions. The test set contains 50 images with various lighting conditions and shapes.

4.2 Results on Cartoon Face Reconstruction

4.2.1 Comparison with Prior Art

We compare our method with PRN [12], a template-free method that predicts face shapes with a CNN, and Deep3D [10], a baseline that predicts 3DMM coefficients in an unsupervised manner. Both works have been proposed recently, showing impressive performance on 3D face reconstruction. We also report the results of our two stages: coarse reconstruction and fine reconstruction, to validate the effectiveness of the two-stage design. We measure the reconstruction quality by computing the 2D landmark and photometric errors on the test set. Specifically, for each test image, we project
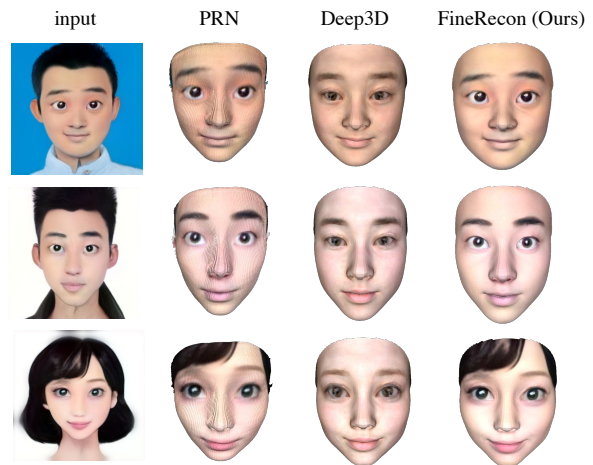
Fig. 6 Comparison of our results with PRN and Deep3D.

the result to the image plane after reconstruction. The landmark error measures the Euclidean distances between the projected landmarks and the annotations, evaluating the correspondence and shape accuracy. We evaluate the error of different face parts separately. We also use the photometric error, which is the average Manhattan distance of the pixel colors between the rendered image and the input image, to evaluate the appearance similarity. We show the average results over the test data.

As Table 1 shows, our method achieves a much lower landmark error than PRN and Deep3D. Although sharing a similar network structure, our coarse reconstruction slightly outperforms Deep3D due to the cartoon data training. Compared with coarse reconstruction, our fine reconstruction significantly improves the eyes' alignment accuracy. The accuracy of other facial parts like the nose, eyebrow, and mouth are also improved. It validates the effectiveness of our deformation-based alignment strategy. To map texture from the input image, alignment with the image should be accurate. Otherwise, it would cause evident unnatural facial colors. Our fine reconstruction also achieves the lowest photometric error due to accurate reconstruction, alignment, and texture mapping. Although PRN utilizes the input image for texture mapping like our method, which is the reason why PRN's result looks similar to the input image, it has a larger photometric error because the inaccuracy of the shape and alignment causes background pixels to be mistakenly mapped to the texture. We show visualization comparisons in Fig. 6.

4.2.2 Evaluation on Face Alignment

Comparison with the template-based method. Eq. 7 shows an intuitive way of adjusting 3DMM coefficients α_{id} and α_{exp} to align with 2D landmark labels. There are two schemes to optimize the coefficients based on templates: adjusting

Method	Landmark Error↓						Photometric Error↓
	eyes	nose	brow	mouth	contour	total	
PRN [12]	269.45	200.41	156.34	397.67	435.86	1459.73	1.06
Deep3D [10]	100.23	0.51	11.60	1.57	16.17	130.08	3.31
CoarseRecon (Ours)	98.33	0.55	11.40	1.54	16.14	127.96	3.27
FineRecon (Ours)	**8.27**	**0.23**	**11.33**	**1.50**	**16.11**	**37.44**	**0.83**

Table 1 Comparison with prior arts.

	Adjust Exp	Adjust Id+Exp	Ours
Landmark Error↓	143.38	110.70	**37.44**

Table 2 Comparison with template-based method.

input Adjust Id+Exp Adjust Exp Ours
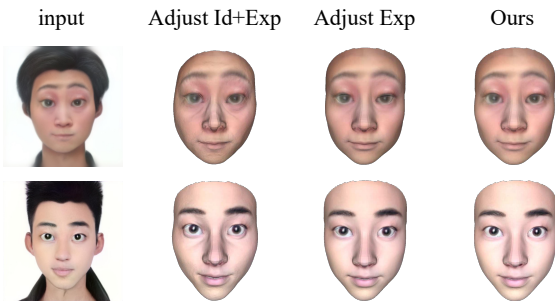
Fig. 7 Comparison of the results using different face alignment strategies: template-based method (Adjust Id+Exp/Adjust Exp) and deformation-based method (ours).

α_{exp} only (Adjust Exp), and adjusting both α_{id} and α_{exp} at the same time (Adjust Id+Exp). Table 2 shows a comparison of our deformation-based method with these two template-based methods. We use landmark errors as the criteria with the same definition in Table 1. Our method has lower landmark errors than these baselines. Interestingly, 'Adjust Id+Exp' gains lower landmark error than 'Adjust Exp', due to a higher degree of freedom (DoF) and larger representation space. In this regard, our method has the highest DoF and shows the lowest error. We also demonstrate with visualization results in Fig. 7. Although 'Adjust Id+Exp' gains lower landmark error than 'Adjust Exp', the visualization shows unnatural wrinkles and distortion due to the restrictions of templates. This suggests that the refinement exceeds the template's representation capability. On the other hand, our method can retain high-quality visual performance, while making accurate adjustments simultaneously.

Fig. 8 shows the comparison of the results on eye areas with or without face alignment. It is clear that before the alignment, part of the eye texture is mistakenly mapped to the face skin, because the coarse reconstructed eyes are too small. During animation, the wrongly mapped texture will be amplified, for example, when the eyes are closing.

(a) without alignment   (b) with alignment
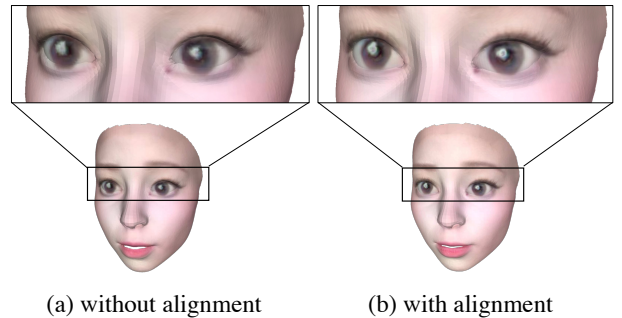
Fig. 8 Comparison of the results on eye areas without (a) and with (b) face alignment.

	Aesthetics	Accuracy	Similarity
PRN [12]	2.63/1.07	2.99/1.04	3.12/1.17
Deep3D [10]	2.66/1.22	2.62/1.01	2.46/1.02
Ours	**3.75/0.92**	**3.88/0.81**	**3.95/0.87**

Table 3 User subjective evaluations. The table shows mean and standard deviation (mean/std. dev.) of users' evaluation scores. Our method achieves the highest ratings on all three subjective criteria (aesthetics, accuracy and similarity).

4.2.3 User Subjective Evaluation

To make a more comprehensive evaluation of our reconstruction results, we conduct a user study to collect subjective evaluations of the reconstruction. For each participant, we send out a questionnaire with six independent questions. For each question, we randomly select a cartoon face image from the test set, and reconstruct its 3D model with PRN, Deep3D and our method. We show the results of these methods in a random order and ask participants to rate for aesthetics, accuracy and similarity. Aesthetics evaluates whether the 3D model is aesthetically pleasing. Accuracy evaluates the correctness of the overall shape and the position of each face part. Similarity evaluates whether the 3D model appears similar to the input image. Participants are asked to rate in the range of 1-5 for each aspect, where 1 for very poor, and 5 for very good.

We invited 55 participants in total, 30 males and 25 females distributed from diverse backgrounds. Table 3 shows
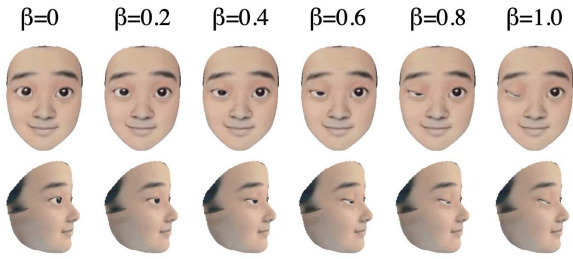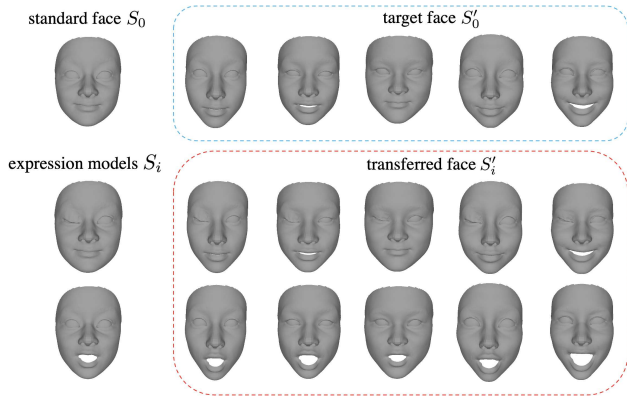
Fig. 9 Template-based facial animation.

Fig. 10 Visualization results of expression transfer on part of the test data. We demonstrate two expressions: 'mouth open' and 'close right eye'.

	Reconstruction		Run-time
CoarseRecon	FineRecon	FaceRigging	Deformation
1.46s	20.02s	1.84s	3.56ms

Table 4 Efficiency Evaluation. We show that our pipeline can reconstruct an arbitrary face model within 30s, and perform real-time facial animation over 280 FPS.

the average score and standard deviation on all participants for each question and method. Our method evidently outperforms the other two methods on subjective criteria, including aesthetics, accuracy and similarity. To our observation, face shape and texture play important roles in improving performance on these subjective criteria. The results suggest that our method does not overfit the landmark constraints, but rather it uses the appropriate constraints to achieve overall high visual quality.

4.3 Results on Face Rigging and Animation

Visualizations on Template-based Facial Animation. Fig. 9 shows the linear combination of the neutral face S_0 and an expression template model S_i with coefficient β, according to Eq. 13. The semantic of S_i is 'right eye close', which allows us to control the right eye independently. We have 46 template models with different semantics such as mouth open, left brows up, lip funnel, etc.

Results on Expression Transfer. We demonstrate the effectiveness of our expression transfer method in Fig. 10. We hide the texture to show the geometry of the faces clearly.

We demonstrate the transfer of two typical expressions: 'right eye close' and 'mouth open'. The results show that, although the target model S_0' varies in shape, the transferred expression can adapt pretty well. This is because, instead of simply applying the vertex shift to the target model, we transfer the transformation of the triangular faces on the mesh.

Eye-ball Modeling. To animate the eyes without eyeball distortion, we model the eyeballs independently during face rigging. A sphere fits the eyeball area, and then we move the sphere inside the head for a small distance Δ to avoid collision with the eyelids. The texture is correspondingly mapped to the sphere, and the invisible parts are set to white by default.

4.4 Application Results

Efficiency Evaluation. Generally, applications require high efficiency of reconstruction and animation. Our experiment is carried out on an computer with an Intel(R) Xeon(R) E5-2678 v3 @ 2.50GHz CPU and a TITAN RTX GPU. We repeat ten times on each test sample and show the average time consumption. As Table 4 shows, our method takes 24 seconds on average for reconstruction and face rigging, which is acceptable for a user to wait. Currently, the fine reconstruction algorithm is implemented on CPU, and we believe that the efficiency will be largely improved if this step is sped up by GPU. For the run-time, results show that our reconstructed model can change its expression with a real-time performance of over 280 FPS.

Real-time Video Driven Face Animation. Utilizing a fast expression animation driver [28], we show the potential of real-time video driven facial animation in Fig. 11. The upstream driver predicts expression coefficients β from a real human face. A reconstructed 3D cartoon face is then animated by β. The driving process can be implemented online with a separate frontend and backend, where the driver serves as the backend, and the animatable 3D model serves as the frontend. Intuitively, with this functionality, users can drive their own avatars to follow their facial actions in a VR application.
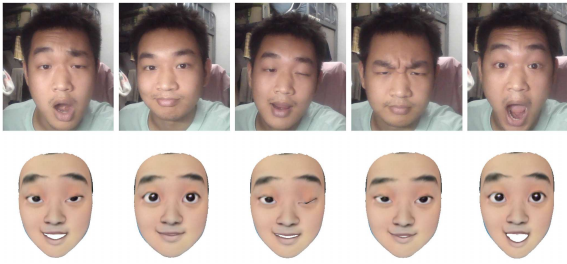
Fig. 11 Visualizations of real-time face-to-face animation. Our reconstructed model can be driven by a real-world reference face, utilizing an upstream expression driver.
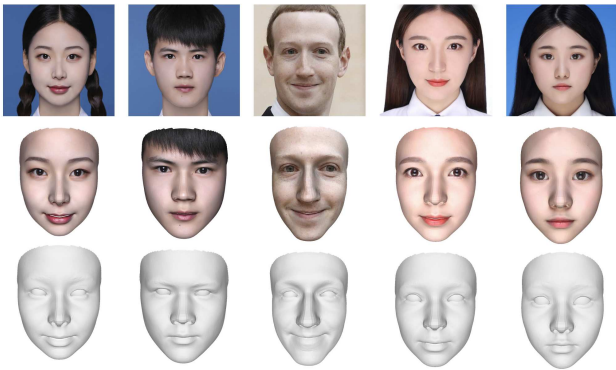
Fig. 12 Realistic 3D face reconstruction results from ordinary portraits. The upper row is the input image. The middle row and the bottom row are reconstructed models with and without texture respectively.

Results on ordinary portrait images. Although we focus on cartoon face reconstruction, our method can also reconstruct high-quality realistic faces. Fig. 12 shows examples of single-view 3D face reconstruction from ordinary portraits with our method.

4.5 More results

Fig. 13 shows more visualization results on cartoon images with different styles. Our method is robust to exaggerated face parts like large eyes and unnatural face shapes.

5 Conclusion

In this paper, we introduce a novel pipeline to generate animatable 3D cartoon faces from a single real-world portrait. To achieve high-quality 3D cartoon faces, we propose a two-stage face reconstruction scheme. We generate semantic-preserving face rigs with manually-created models and expression transfer. Quantitative and Qualitative results show that our reconstruction achieves high performance on accuracy, aesthetics and similarity. Furthermore, we show the ca-
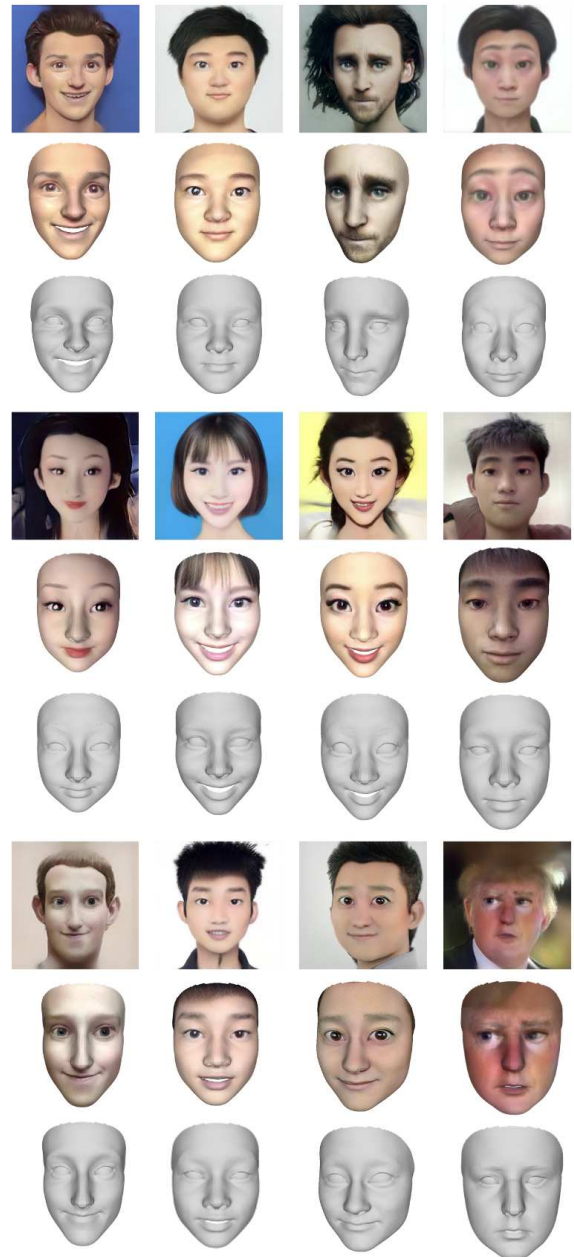
Fig. 13 More visualization results. We conduct reconstruction on images with different styles. For every three rows, the first row shows the input cartoon images, the second and third row show 3D models with and without texture.

pability of real-time animation of our model. Our pipeline can be applied to creating user 3D avatars in VR/AR applications. Generating high-quality animatable 3D faces with various styles is a difficult task, and we would like to generalize our method to a larger range of styles in our future research agenda.

References

1. Alexander, O., Rogers, M., Lambeth, W., Chiang, J.Y., Ma, W.C., Wang, C.C., Debevec, P.: The digital Emily project: Achieving a photorealistic digital actor. IEEE Computer Graphics and Applications **30**(4), 20–31 (2010)
2. Bas, A., Smith, W.A., Bolkart, T., Wuhrer, S.: Fitting a 3D morphable model to edges: A comparison between hard and soft correspondences. In: Asian Conference on Computer Vision, pp. 377–391. Springer (2016)
3. Blanz, V., Basso, C., Poggio, T., Vetter, T.: Reanimating faces in images and video. In: Computer graphics forum, vol. 22, pp. 641–650. Wiley Online Library (2003)
4. Blanz, V., Mehl, A., Vetter, T., Seidel, H.P.: A statistical method for robust 3D surface reconstruction from sparse data. In: Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004., pp. 293–300. IEEE (2004)
5. 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, pp. 187–194 (1999)
6. Cai, H., Guo, Y., Peng, Z., Zhang, J.: Landmark detection and 3D face reconstruction for caricature using a nonlinear parametric model. Graphical Models **115**, 101,103 (2021)
7. Cao, C., Bradley, D., Zhou, K., Beeler, T.: Real-time high-fidelity facial performance capture. ACM Transactions on Graphics (ToG) **34**(4), 1–9 (2015)
8. Casas, D., Feng, A., Alexander, O., Fyffe, G., Debevec, P., Ichikari, R., Li, H., Olszewski, K., Suma, E., Shapiro, A.: Rapid photorealistic blendshape modeling from RGB-D sensors. In: Proceedings of the 29th International Conference on Computer Animation and Social Agents, pp. 121–129 (2016)
9. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018). URL http://www.blender.org
10. Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y., Tong, X.: Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 0–0 (2019)
11. Ekman, P., Friesen, W.V.: Facial action coding system. Environmental Psychology & Nonverbal Behavior (1978)
12. Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X.: Joint 3D face reconstruction and dense alignment with position map regression network. In: Proceedings of the European conference on computer vision (ECCV), pp. 534–551 (2018)
13. Garrido, P., Zollhöfer, M., Casas, D., Valgaerts, L., Varanasi, K., Pérez, P., Theobalt, C.: Reconstruction of personalized 3D face rigs from monocular video. ACM Transactions on Graphics (TOG) **35**(3), 1–15 (2016)
14. Guo, Y., Cai, J., Jiang, B., Zheng, 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 **41**(6), 1294–1307 (2018)
15. Hassner, T.: Viewing real-world faces in 3D. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3607–3614 (2013)
16. Hassner, T., Basri, R.: Example based 3D reconstruction from single 2D images. In: 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), pp. 15–15. IEEE (2006)
17. Hassner, T., Harel, S., Paz, E., Enbar, R.: Effective face frontalization in unconstrained images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4295–4304 (2015)
18. Ichim, A.E., Bouaziz, S., Pauly, M.: Dynamic 3D avatar creation from hand-held video input. ACM Transactions on Graphics (ToG) **34**(4), 1–14 (2015)
19. Jourabloo, A., Liu, X.: Large-pose face alignment via CNN-based dense 3D model fitting. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4188–4196 (2016)
20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4401–4410 (2019)
21. Kemelmacher-Shlizerman, I., Seitz, S.M.: Face reconstruction in the wild. In: 2011 international conference on computer vision, pp. 1746–1753. IEEE (2011)
22. Kim, H., Zollhöfer, M., Tewari, A., Thies, J., Richardt, C., Theobalt, C.: Inversefacenet: Deep single-shot inverse face rendering from a single image. arXiv preprint arXiv:1703.10956 (2017)
23. King, D.E.: Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research **10**, 1755–1758 (2009)
24. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. ACM Transactions on Graphics **39**(6) (2020)
25. Lewis, J.P., Anjyo, K., Rhee, T., Zhang, M., Pighin, F.H., Deng, Z.: Practice and theory of blendshape facial models. Eurographics (State of the Art Reports) **1**(8), 2 (2014)
26. Li, H., Weise, T., Pauly, M.: Example-based facial rigging. Acm transactions on graphics (tog) **29**(4), 1–6 (2010)
27. Liu, J., Chen, Y., Miao, C., Xie, J., Ling, C.X., Gao, X., Gao, W.: Semi-supervised learning in reconstructed manifold space for 3D caricature generation. In: Computer Graphics Forum, vol. 28, pp. 2104–2116. Wiley Online Library (2009)
28. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.L., Yong, M.G., Lee, J., et al.: Mediapipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172 (2019)
29. Muslim, H.S.M., Khan, S.A., Hussain, S., Jamal, A., Qasim, H.S.A.: A knowledge-based image enhancement and denoising approach. Computational and Mathematical Organization Theory **25**, 108–121 (2019)
30. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
31. Pawaskar, C., Ma, W.C., Carnegie, K., Lewis, J.P., Rhee, T.: Expression transfer: A system to build 3D blend shapes for facial animation. In: 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013), pp. 154–159. IEEE (2013)
32. Pinkney, J.N., Adler, D.: Resolution dependent GAN interpolation for controllable image synthesis between domains. arXiv preprint arXiv:2010.05334 (2020)
33. Qiu, Y., Xu, X., Qiu, L., Pan, Y., Wu, Y., Chen, W., Han, X.: 3DCaricShop: A dataset and a baseline method for single-view 3D caricature face reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10,236–10,245 (2021)
34. Ramamoorthi, R., Hanrahan, P.: An efficient representation for irradiance environment maps. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 497–500 (2001)
35. Romdhani, S., Vetter, T.: Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, pp. 986–993. IEEE (2005)
36. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2304–2314 (2019)

37. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. ACM Transactions on graphics (TOG) **23**(3), 399–405 (2004)

38. Vlasic, D., Brand, M., Pfister, H., Popovic, J.: Face transfer with multilinear models. In: ACM SIGGRAPH 2006 Courses, pp. 24–es (2006)

39. Wu, Q., Zhang, J., Lai, Y.K., Zheng, J., Cai, J.: Alive caricature from 2D to 3D. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7336–7345 (2018)

40. Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., Sang, N.: BiSeNet v2: Bilateral network with guided aggregation for real-time semantic segmentation. International Journal of Computer Vision **129**(11), 3051–3068 (2021)

41. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters **23**(10), 1499–1503 (2016)

42. Zhang, Y., Di, X., Zhang, B., Li, Q., Yan, S., Wang, C.: Self-supervised low light image enhancement and denoising. arXiv preprint arXiv:2103.00832 (2021)

43. Zhou, J., Wu, H.T., Liu, Z., Tong, X., Guo, B.: 3D cartoon face rigging from sparse examples. The Visual Computer **34**(9), 1177–1187 (2018)

44. Zhou, K., Huang, J., Snyder, J., Liu, X., Bao, H., Guo, B., Shum, H.Y.: Large mesh deformation using the volumetric graph Laplacian. In: ACM SIGGRAPH 2005 Papers, pp. 496–503 (2005)

45. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: A 3D solution. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 146–155 (2016)

46. Zhu, X., Lei, Z., Yan, J., Yi, D., Li, S.Z.: High-fidelity pose and expression normalization for face recognition in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 787–796 (2015)