TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 16/24 pp506–516
DOI: 10.26599/TST.2023.9010028
Volume 29, Number 2, April 2024

# Synthesis, Style Editing, and Animation of 3D Cartoon Face

Ming Guo, Feng Xu*, Shunfei Wang, Zhibo Wang, Ming Lu, Xiufen Cui, and Xiao Ling

**Abstract:** As a popular kind of stylized face, cartoon faces have rich application scenarios. It is challenging to create personalized 3D cartoon faces directly from 2D real photos. Besides, in order to adapt to more application scenarios, automatic style editing, and animation of cartoon faces is also a crucial problem that is urgently needed to be solved in the industry, but has not yet had a perfect solution. To solve this problem, we first propose "3D face cartoonizer", which can generate high-quality 3D cartoon faces with texture when fed into 2D facial images. We contribute the first 3D cartoon face hybrid dataset and a new training strategy which first pretrains our network with low-quality triplets in a reconstruction-then-generation manner, and then finetunes it with high-quality triplets in an adversarial manner to fully leverage the hybrid dataset. Besides, we implement style editing for 3D cartoon faces based on k-means, which can be easily achieved without retrain the neural network. In addition, we propose a new cartoon faces' blendshape generation method, and based on this, realize the expression animation of 3D cartoon faces, enabling more practical applications. Our dataset and code will be released for future research.

**Key words:** cartoon; 3D face; animation; computer graphics

## 1 Introduction

Cartoon face is a prevalent kind of stylized face, which is widely used in movies, advertisements, games, and virtual reality. Generating 3D cartoon faces directly and automatically from real faces can largely extend the ability of digital content creation. Although many previous works can generate high-quality 2D cartoon faces[1–3], 3D cartoon faces generation still mainly relies on tedious manual works by artists using professional 3D modelling software. Therefore, high-quality 3D cartoon faces are usually only used in high-end fields such as game and film production. As for ordinary users, it is challenging to customize 3D cartoon faces with personalized features from 2D real face images. Although there are some 3D stylized face generation methods, they all focus on caricatures, which are quite different from cartoon faces. Besides, they usually require sketches as additional input[4, 5] or need to know the 3D model of the input real face[6]. In addition, none of the existing works can generate cartoon faces in various and user-defined styles or personalized face animations, which limits their application range.

Deep learning has been proven to be successful in many fields these years[7, 8]. In this paper, we introduce a learning-based method to generate both the shape and texture of 3D cartoon faces directly from 2D real face images. To achieve this, we built a hybrid dataset consisting of 6842 low-quality and 130 high-quality face triplets. Each triplet contains a 2D real face image, its corresponding 2D cartoon face, and textured 3D cartoon face. For geometry, learning the relationship between 3D cartoon faces and 2D real faces suffers

• Ming Guo, Feng Xu, and Zhibo Wang are with School of Software and BNRist, Tsinghua University, Beijing 100084, China. E-mail: gm20@mails.tsinghua.edu.cn; xufeng2003@gmail.com; sireerw@gmail.com.
• Shunfei Wang, Xiufen Cui, and Xiao Ling are with Department of Multimedia and Smart, Guangdong OPPO Mobile Telecommunications Corp. Ltd., Dongguan 523860, China. E-mail: wangshunfei@oppo.com; cuixiufen@oppo.com; lingxiao@oppo.com.
• Ming Lu are with Intel Labs, Intel, Beijing 100089, China. E-mail: lu199192@gmail.com.
* To whom correspondence should be addressed.
Manuscript received: 2023-02-19; revised: 2023-04-03; accepted: 2023-04-04

not only the 2D-to-3D ambiguity but also the real-to-cartoon transformation. Besides, it is not easy to learn to generate high-quality results with a limited number of high-quality data in our dataset. We address this by a novel training strategy, named Recon2AGen, that learns reconstruction before generation by just using the low-quality dataset and then utilize high-quality data in an adversarial manner to further fine-tune our geometric synthesis module. As for texture generation, we integrate style transfer and texture generation into a single geometry-aware synthesis approach, which utilizes the powerful generation capability of Generative Adversarial Networks (GANs).

Based on our generation technique, we achieve style editing of cartoon faces using a simple k-means based method. This greatly increases the flexibility of our generation results, making it possible to obtain various types of cartoon faces without retraining the network. Besides, users can also customize the template style of the cartoon face according to their own preferences and get the user-defined style of cartoon faces.

We also make the generated 3D cartoon faces animatable, which further increases the practical usability of our method. We propose a novel method for constructing blendshapes for cartoon faces that reflect the performance results of each generated cartoon face under a variety of different expressions. Then we capture the expressions and motion sequences of live face videos with a blendshape-based real-time face tracking method, and migrate them to cartoon faces for realistic expression animation.

In summary, the contributions can be concluded as follows:

(1) We propose "3D face cartoonizer", which is the first method that generates 3D cartoon faces directly from 2D facial images with high-quality geometry and texture to the best of our knowledge.

(2) In order to make our results more diverse, we implement style editing for 3D cartoon faces based on k-means.

(3) We animate the generated 3D cartoon faces by a novel user-specific expression blendshapes construction method.

## 2 Related Work

### 2.1 3D stylized face reconstruction

Thanks to the rapid development of 3D human face reconstruction, several stylized face reconstruction works emerge recently. However, almost all of them focus on caricatures. Due to the considerable geometrical difference between caricatures and real faces, previous work[9] has demonstrated that directly using normal face models, such as 3DMM[10] and FaceWareHouse[11], cannot fit them correctly. Approaches that aim to address this limitation can be divided into two categories. In the first category, the problem is solved by manually making 3D meshes for caricatures and using the results to build a specific parametric model, such as that in Ref. [12]. However, it is usually time-consuming and costly for artists to make 3D caricatures. In the second category, the problem is solved by designing parametric models beyond the scope of normal face models[9, 13]. However, we observe that these methods will introduce artifacts like wrinkled surfaces and self-intersections when applied to cartoon faces. While plenty of methods are proposed for 3D caricature faces, as one popular style type, 3D cartoon faces are never studied by existing methods to the best of our knowledge.

### 2.2 3D stylized face generation

Similar to 3D stylized face reconstruction, existing works on 3D stylized face generation primarily focus on the caricature face generation. Some of them are based on the 3D models of real faces, and generate caricatures by exaggerating the difference between the input face and the mean face[6, 14]. Ye et al.[15] used a cycle-form network to get 3D caricature from 2D real faces, but lake of paired data made the relevance of the input and their generation result not obvious. Guo et al.[16] used normal faces video as input to obtain 3D caricature sequence, but had to reconstruct the real faces first which may lead to error accumulation and time-consuming. Other methods require additional input like sketches drawn by users[4, 5]. Although they can obtain 3D caricatures, the above methods are highly dependent on the quality of the input sketch and not totally automatic. Reference [17] used 2D real faces as input, while the generated results are limited in a predefined Principal Component Analysis (PCA) space. Reference [18] proposed an end-to-end method that can generate 3D caricatures directly from 2D face photos, which builds a 3D caricature dataset through an automatic 3D caricature reconstruction method[9], and learns a PCA-based parametric model for the 3D caricature from the dataset. However, since the 3D caricatures in the dataset do not have their corresponding 2D real faces, the identity similarity between the synthetic result and

the input real face is not very high. In addition, the 3D caricatures in the dataset are all reconstructed by the automatic method based on sparse constraints, and the lack of high-quality data in the dataset may limit the quality of the generated results. Reference [19] presented a framework for one-shot 3D portrait style transfer, which can generate 3D stylized faces with only one arbitrary style image. However, it is difficult to decouple the style information and identity information of the arbitrary style image. The generated results easily carry the unpleasant identity information of the arbitrary style image, especially texture results such as skin color.

### 2.3 3D face animation

Blendshape is a commonly used face parametric model in animation production, which indicates different facial expressions through the linear combination of multiple given standard meshes[20], it usually expresses typical facial expressions, such as FaceWarehouse[11] and Facescape[21]. Blendshape has many applications, such as facial retargeting[22] and animation[23]. Combined with the latest facial expression tracking method, Blendshape can explicitly describe the facial expression change process in a video[24]. Compared with the PCA-based face parametric models, such as Flame[25], Blendshape can easily be used to retarget from a video with continuously changing face to another virtual avatar, so it has a wider range of applications in the field of facial expression animation generation[24, 26]. However, it is not easy to automatically build blendshapes for any 3D face. For example, Ref. [11] made blendshapes for real faces based on mesh deformation. But cartoon faces usually have a more exaggerated collection of faces, this approach does not work well. Reference [27] proposes a blendshape-generation method for caricatures, but it requires more complex additional calculations and is not suitable for cartoon faces.

## 3 Hybrid Cartoon Dataset

To facilitate the learning of 3D cartoon face generation, we construct a hybrid cartoon dataset with both low and high-quality data. It connects real and cartoon face domains, providing both 2D and 3D information with different quality. Specifically, our dataset contains 6972 data triplets, each of which includes a real facial image $I_r$ (Fig. 1a), its corresponding 2D cartoon image $I_c$ generated by a web app, ToonMe (Fig. 1b)†, and its 3D

† https://toonme.com

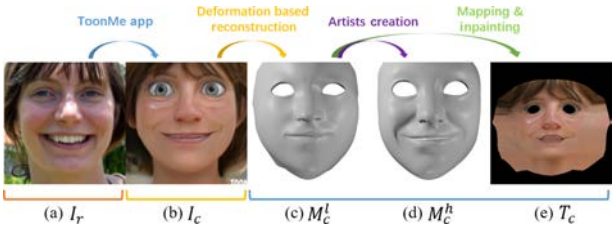



**Fig. 1 Pipeline for building the triplets in our hybrid dataset. (a) 2D real face, (b) corresponding 2D cartoon face, (c) and (d) low and high-quality 3D cartoon meshes, respectively, and (e) cartoon texture.**

cartoon face mesh $M_c$ with texture $T_c$ (Fig. 1e). For the 6842 low-quality data triplets, the 3D cartoon face meshes $M_c^l$ (Fig. 1c) are generated by a modified denser-landmark guided face fitting algorithm[9]. To provide more explicit 3D cartoon guidance, our dataset also contains 130 high-quality 3D cartoon faces $M_c^h$ created by expert artists based on $M_c^l$, forming the high-quality triplets (Fig. 1d). More details about the dataset can be found in Ref. [28].

### 3.1 Dense landmark annotation

Here, we would like to detail the process of denser landmark annotation. As the cartoon face reconstruction process is guided by face landmarks, and the standard 68-point landmarks are not sufficient to express the shape of a cartoon face, especially around the eyes and nose. Therefore, we enrich the landmarks from 68 points to 106 points as shown in Fig. 2b. Blue points are the standard 68-point landmarks, and the red and green ones are our added landmarks. Extra landmarks are on the nose and eyelids, which are important face regions for human perception and the key to distinguishing between a cartoon face and a real one. Landmarks on the nose are obtained by manually labeling the four end ones and interpolating the intermediate ones. For the landmarks on eyelids, as the area is too small to perform accurate manual labeling, we apply a method based on edge detection and curve fitting. We first use the sparse

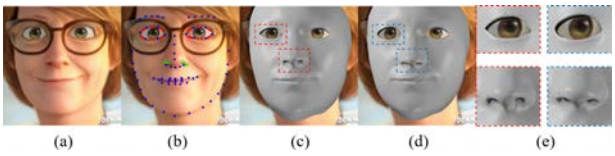



**Fig. 2 106-point landmarks with the fitting results. (a) Cartoon image, (b) image with 106-point landmarks, (c) and (d) fitting results using 68-point and 106-point landmarks, respectively, and (e) zoom-in results. Notice that using 106-point landmarks, the eye contour, and nose shape in (d) and (e) are more consistent with cartoon image (a).**

landmarks to extract small eye patches, and then we use the Sobel operator[29] to detect the edges of eye patches. Next, we fit a third-order polynomial curve for the edge map of each eyelid (upper eyelid and lower eyelid for one eye) by a weighted least square method, which solves the following minimization problem:

$$\arg\max_{A} \sum_{k \in C} w_k \, \| P(A, x_k) - y_k \| \qquad (1)$$

where $A = \{a_i | i = 0, 1, 2\}$ stands for the polynomial parameter, $P$ is the polynomial function, $C$ is the set of valid pixels obtained by the edge detector. $(x_k, y_k)$ is the 2D coordinates of each valid pixel, and the weight parameter $w_k$ is the intensity in the edge map (a kind of edge possibility). Finally, we evenly sample the fitted curves to increase the number of landmarks on each eyelid from 4 points to 12 points. After dense landmark annotation, we adopt the method proposed by Ref. [9] to obtain the 3D cartoon faces by fitting the 2D cartoon faces using the landmarks. As shown in Figs. 2c–2e, dense landmark annotation helps the fitting algorithm reconstruct better 3D cartoon faces with smoother and more natural eyelid contours and the characteristic flat nose shapes in this cartoon style.

### 3.2 Texture completion

Here we prepare the "ground truth" texture of the 3D cartoon faces, which is used in training the texture synthesis module. As the 3D cartoon face is fitted to the 2D cartoon face, we can obtain a texture map for the 3D cartoon face by sampling the 2D cartoon face image as Fig. 3a. However, errors will be involved without considering the visibility of surface points on the 3D cartoon face in Fig. 3b. We further notice that due to the errors in the fitting, only considering the visibility obtained by the fitting still involves errors. To better handle this, we propose a more "strict" visibility detection that excludes the visible points whose normals are almost orthogonal to the camera viewing directions. In this case, there are still some missing regions in the map as Fig. 3c. To solve for the missing regions, we apply a texture completion method based on even extensions to process the missing regions. For a certain point in the missing area of the texture map, we first find its closest point on the visible boundary. Then the color of its symmetrical point relative to the boundary point is used as its color. In this way, we remove the artifacts on the side face and chin while maintaining the colors' continuity at the visible boundaries as shown in Fig. 3d.

(a) (b) (c) (d)

**Fig. 3 Example of texture completion. (a) Input image, (b) and (c) texture maps generated without and with the consideration of visibility, respectively, and (d) final texture map obtained by the symmetry-guide hallucination scheme.**

### 3.3 Style of our dataset

To show the style of our dataset, we put some examples in Fig. 4 to highlight the characteristic of our dataset. Besides, to show the differences between cartoon faces in our dataset and caricatures, we also put 3D caricatures from a typical 3D caricature dataset[13] The style characteristics of cartoon faces and caricatures are quite different. The geometrical features are different. The shape of the cartoon face is rounded and cute, but the shape of caricatures is much more exaggerated and sharp. The texture features are different, too. Cartoon-style textures are usually crisp, smooth, and bright, while caricature-style textures are usually sketchy and unsmooth.

## 4 Method

### 4.1 3D cartoon face synthesis

The architecture of the proposed 3D face cartoonizer

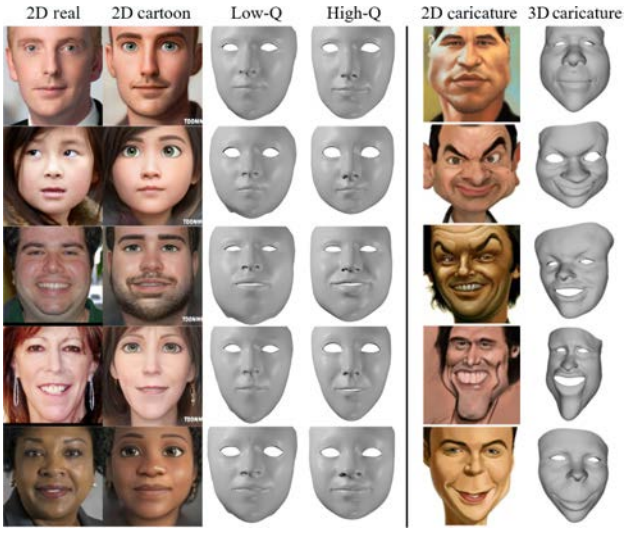



**Fig. 4 Examples of our 3D cartoon dataset and 3D caricature dataset[13]. The four columns on the left are examples from our 3D cartoon dataset. From left to right: 2D real faces, 2D cartoon faces, low-quality (namely low-Q) 3D cartoon faces, and high-quality (namely high-Q) 3D cartoon faces. The two columns on the right are examples from 3D caricature dataset[13], which are 2D caricatures and 3D caricatures, respectively.**

is shown in Fig. 5. Our method consists of a geometry module and a texture module which are trained on our hybrid cartoon dataset.

#### 4.1.1 Geometry module

The geometry module is the most critical component of our method. It is trained to output a 3D cartoon facial geometry from a single real facial image. In order to generate high-quality outputs, we carefully design its network structures and train it using a novel training strategy Recon2AGen.

As shown in Fig. 5, our geometry module has a multi-encoder and single-decoder architecture. It contains two encoders $E_{\text{recon}}$ and $E_{\text{gen}}$, which can regress the feature vectors from 2D cartoon images and 2D real images respectively. The feature vectors can be divided into a geometry feature vector which only encodes the 3D cartoon facial geometry, and a 3D pose $P \in \text{se}(3)$, that is special Euclidean group. The geometry feature, either output by $E_{\text{recon}}$ or $E_{\text{gen}}$, is then be fed into a shared decoder $D_{\text{share}}$. Similar to Ref. [9], the decoder $D_{\text{share}}$ predicts deformation representation instead of directly outputting the vertex positions of the 3D cartoon face. By applying the estimated deformation gradient to the mean 3D cartoon face, the geometry module can generate the final shape of the 3D cartoon face.

Besides, in order to generate high-quality outputs, we train the geometry module using a novel training strategy Recon2AGen, which fully explores our hybrid dataset and solves both the 2D-to-3D ambiguity and the real-to-cartoon transformation. The training process has three different stages. In Stage 1, the geometry module learns reconstruction using $E_{\text{recon}}$ and $D_{\text{share}}$. It is trained to reconstruct the 3D cartoon face from a given 2D cartoon image, supervised by the large quantity of low-quality training data. In Stage 2, we fix the decoder $D_{\text{share}}$ and train a new encoder $E_{\text{gen}}$ from scratch to transfer the input 2D real image to 3D cartoon domain. In Stage 3, to enhance the quality of the 3D cartoon face generation, we finetune the geometry module on the artist-made high-quality data in an adversarial manner. More details of the training strategy and loss setting can be found in Ref. [28].

#### 4.1.2 Geometry-aware texture synthesis

The goal here is to synthesize a full texture map in the UV-space of the 3D cartoon face. As textures are strongly correlated to facial geometry, the texture module is guided by the geometric information of the 3D cartoon face predicted by the geometry module, as shown in Fig. 5. Our geometry-aware GAN does not directly concatenate the input image with the geometry guidance. Instead, we first use two shallow encoders, noted as $E_{\text{img}}$ and $E_{\text{normal}}$, to transfer the input image and the normal map into two feature maps. They will be added and then injected into pSp[30] which is the state-of-the-art encoder for StyleGAN (noted as $E_{\text{tex}}$). Finally the pretrained StyleGAN using our texture dataset will generate texture maps in the UV-space with the input feature map output by $E_{\text{tex}}$.

### 4.2 Style editing

We implement a simple but effective method for typical cartoon style editing of the generated 3D cartoon faces. Specifically, we use k-means algorithm to find several typical styles of 3D cartoon faces represented by the cluster centers of the low-quality data of our hybrid dataset. Our geometry module generates a cartoon face

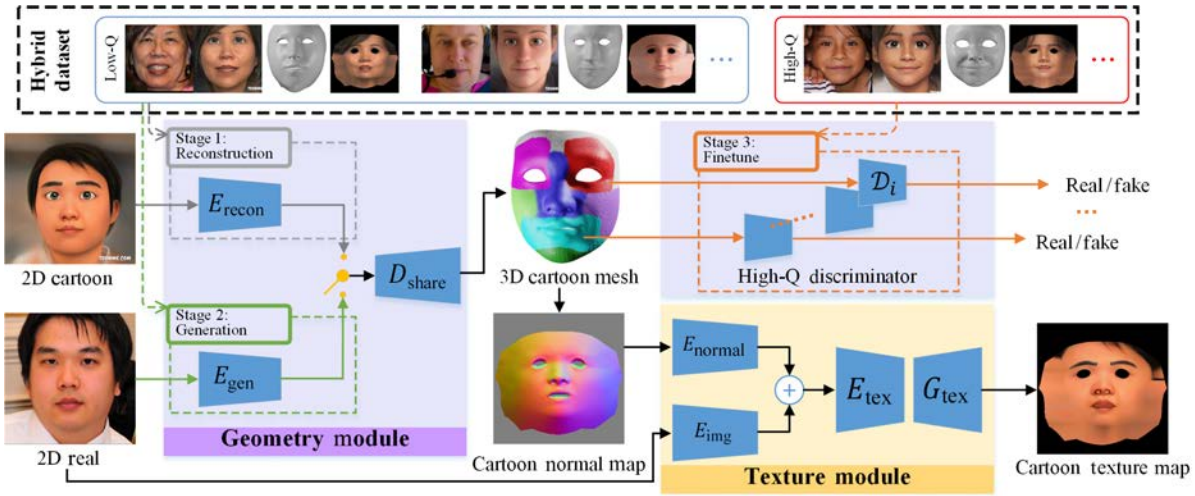



**Fig. 5 Overview of our 3D face cartoonizer framework. Our framework contains two modules for geometry generation and texture generation. Notice that we only need 2D cartoon faces for training, which are not required for real usage.**

by first estimating a suitable deformation and then applying it to the mean cartoon face. To enable the generation of different styles, we replace the mean cartoon face with the aforementioned different typical 3D cartoon faces, and thus the geometry module is able to flexibly produce 3D cartoon faces with the corresponding typical styles. We also design a user interface, so that users can easily and freely edit the style of the generated cartoon face (see Fig. 6).

### 4.3 Facial animation

We animate the generated 3D cartoon faces by constructing user-specific expression blendshapes. Specifically, we first ask artists to make a set of expression blenshapes for the mean face of our 3D cartoon dataset. The semantic definition of the expression blendshapes is the same as that in Ref. [31]. Then, we apply the deformation transfer algorithm[32] to transfer the artist-made expression blendshapes to fit the identity of the generated 3D cartoon face. Next, we design landmark-guided blendshape correction method using Laplacian deformation[33] to automatically fix the artifacts in the results of deformation transfer[32], e.g., open eyes in the "eye closing" blendshape. In detail, we try to minimize the distance between the landmarks on the upper eyelids and the corresponding ones on the lower eyelids while forcing Laplacian coordinates of all vertices almost unchanged. In addition, we only set vertices close to the "closing eyelid" deformable. Finally, we adopt a face tracking algorithm[11] to extract the facial rigid motion and expression blendshape coefficients to drive the user-specific blendshapes, generating 3D cartoon faces under the target facial poses and expressions.

## 5 Experiment

In this section, we evaluate the proposed 3D cartoon face synthesis, style editing, and animation method with thorough qualitative and quantitative experiments.

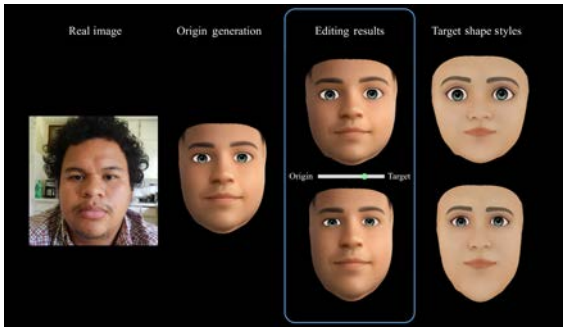



**Fig. 6 User interface of style editing of 3D cartoon faces.**

### 5.1 Implementation details

#### 5.1.1 Network architecture

For the geometry module, as mentioned in Section 4.1.1, two encoders $E_{\text{recon}}$ and $E_{\text{gen}}$ are fed in with 2D cartoon images and 2D real images, respectively, and output feature vectors contain a geometry part and a pose part. Both of them use ResNet-34[34] as the backbone and are enhanced with an attention mechanism. The channel attention layer and the spatial attention layer[35] are inserted both before and after the "Resblocks" of ResNet-34. The result of $E_{\text{recon}}$ or $E_{\text{gen}}$ is a vector $\boldsymbol{f} \in \mathbf{R}^{100}$. Eight independent discriminators $\mathcal{D}_i$, $i \in \{1, 2, \ldots, 8\}$ try to distinguish the generated 3D face shapes from the artist-made ones by different local regions of a mesh.

For the texture module, it has totally three encoders, namely $E_{\text{img}}$, $E_{\text{normal}}$, and $E_{\text{tex}}$. $E_{\text{tex}}$ adopts the same architecture as pSp but removes the input layer. Please refer to Ref. [30] for more details.

#### 5.1.2 Training details

For the geometry module, we randomly choose 6140 triplets for training and the rest 702 triplets for testing from the low-quality data. In the high-quality data, there are 90 triplets for training and 40 triplets for testing. We use Adam optimizer in the training process. When training the texture generation module, we remove all triplets with eyeglasses to achieve clean facial textures. Finally, 4889 triplets are used for training leaving 556 triplets for testing. Here, we use Ranger optimizer, a synergistic optimizer combining Rectified Adam[36] and LookAhead[37]. The total training process costs about 6 days on a single GTX 2080 GPU. The average time to process an image in the inference stage is 0.02 ms for the geometry module and 0.08 s for the texture module.

### 5.2 Results of synthesis

To the best of our knowledge, there is no previous method that can directly convert a real portrait to a 3D cartoon face. Therefore, we compare our generation results with two state-of-the-art stylized facial image reconstruction methods, the fitting-based method "Alive"[9] and the learning-based method "Cari"[13]. Both methods have to get 2D cartoon faces by ToonMe APP first and then perform reconstruction. Note that Ref. [13] is retrained on our proposed dataset.

#### 5.2.1 Qualitatively comparisons

The qualitative comparisons among our "3D face

cartoonizer" and these two indirect solutions (Alive[9] and Cari[13]) are shown in Fig. 7. The proposed 3D face cartoonizer outperforms all other methods in generating more vivid cartoon faces. Although most training samples are low-quality 3D cartoon faces, our geometry module still learns the style characteristics of the small number of high-quality artist-made 3D cartoon faces, and even synthesizes more expressive cartoon style details than the artist-made 3D cartoon faces, especially around the nasolabial folds. We also demonstrate the effect of our texture generation module in the last column in Fig. 7 by rendering the 3D cartoon face results with the generated textures. The generated textures dramatically enhance aesthetics and the identity similarity with the input face image.

We need to briefly clarify the inconsistent expression between the generated 3D cartoon face and the input 2D real face. Since the 3D cartoon face we generated is based on the 2D cartoon face generated by ToonMe. However, ToonMe itself cannot maintain the same expression between the input 2D real face and the generated 2D cartoon face. No matter what expression the input face is, the output is a result of neutral expression. The results of our generated and the two indirect solutions (Alive[9] and Cari[13]) are based on the results of ToonMe, so they also have this feature. This feature is beneficial for driving, however. The result of neutral expression generation will further help us obtain the blendshapes of each typical expression, which is more convenient to drive.

### 5.2.2 Quantitatively comparisons

We also quantitatively compare our method with those in Refs. [9] and [13]. As there is no "accurate" ground truth for 3D cartoon face generation, we conduct a perceptual study to demonstrate the visual quality of these methods. To present a thorough evaluation of different methods, each participant rates each 3D cartoon face on the artistry, the identity similarity with the real face, and the style similarity with the cartoon face, scoring from 1–5 (1 for the worst and 5 for the best). Our method achieves the highest artistry score when comparing with not only the alternative solutions, but also the manual work of artists (high-quality 3D cartoon faces) in our proposed dataset. In identity similarity and style similarity, our method still achieves dramatic improvement on these two metrics among all the automatic 3D cartoon face generation methods. The quantitative results in Table 1 exhibit that our method can greatly improve the quality

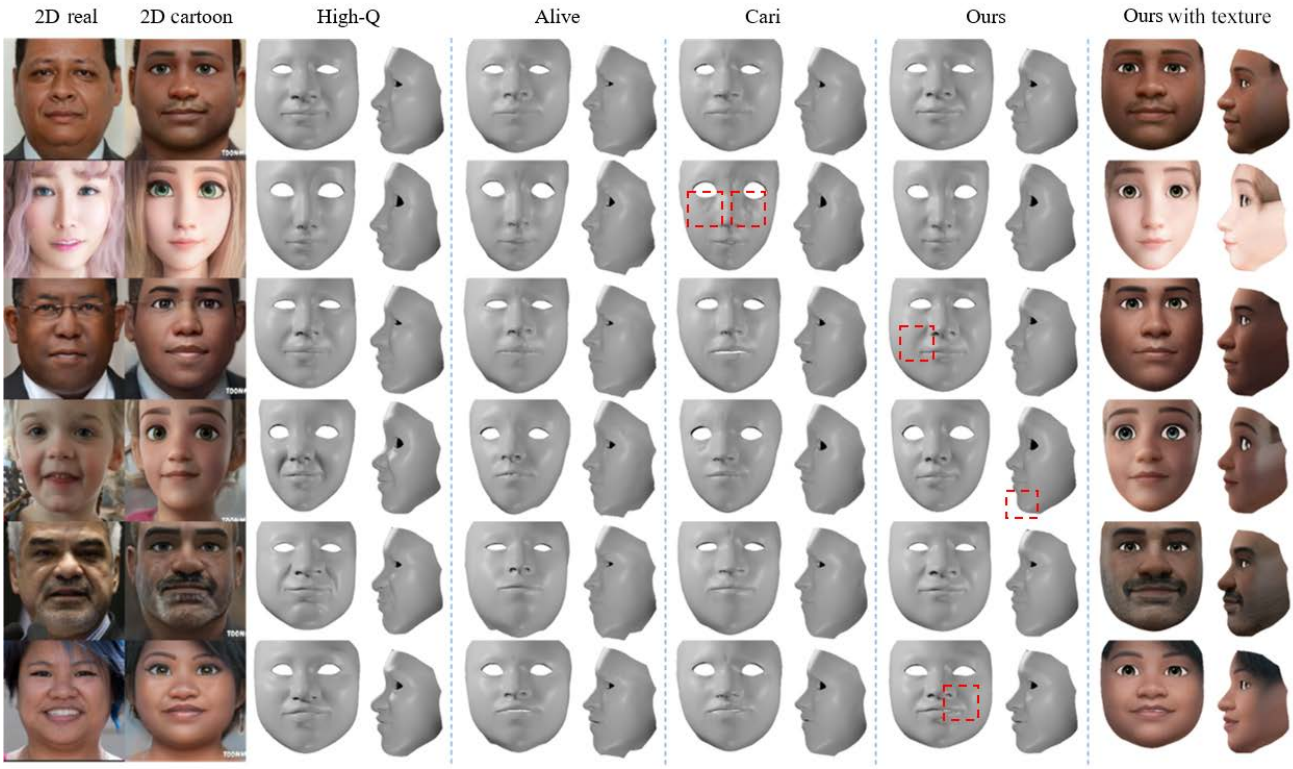



**Fig. 7 Qualitative comparisons of our 3D cartoon face generation with other 3D catoon face reconstruction methods ("Alive"[9] and "Cari"[13]). Notice that neither "Alive" nor "Cari" can directly obtain the results from real image inputs as we do; they rely on the 2D cartoon face as inputs.**

**Table 1 Perceptual study of different 3D cartoon face generation methods.**

| Metric | Alive[9] | Cari[13] | Ours | High-Q |
|---|---|---|---|---|
| Artistry | 2.67 | 2.88 | **4.39** | 4.27 |
| Identity similarity | 2.56 | 2.75 | 3.72 | **4.01** |
| Style similarity | 2.84 | 3.31 | 4.23 | **4.28** |

in automatically synthesizing a 3D cartoon from a real facial image. More details about the user study can be found in Ref. [28].

### 5.2.3 More results

To further demonstrate the generalization of our method, Fig. 8 shows that our method can successfully generate high-quality cartoon faces from real facial images with different genders, ages, and races. Figure 8 presents the front view and side view without head movement of the origin generated results, with a better observation of generated effects.

**(1) Style editing**

We present some results of the 3D cartoon face style editing in Fig. 9. This style editing method can synthesize 3D cartoon faces with diverse types of style while preserving the identity and the expression of the original faces.

**(2) Facial animation**

We further conduct an ablation study in Fig. 9 to show with and without laplacian correction effects. Without Laplacian deformation, the corresponding closing eyes in "left eye closing" or "right eye closing" blendshapes cannot fully "be closed". On the contrary, the landmark-guided Laplacian deformation helps greatly to correct these errors (Fig. 9, 4-th and 5-th columns). We provide the results of facial animation retarget in Fig. 10. We migrate motions from a real face video to our generated 3D cartoon faces in Fig. 11.

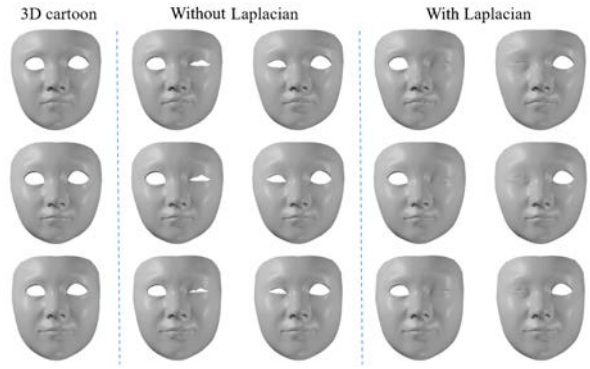



**Fig. 9 Qualitative comparison between blendshapes without and with Laplacian deformation correction.**

**Fig. 10 Cartoon facial animation driven by real images.**

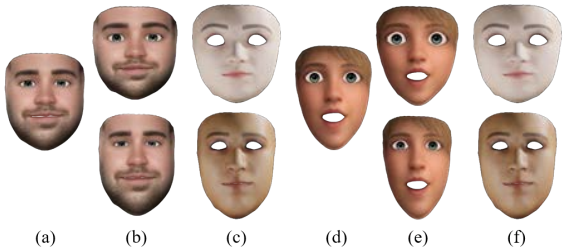



**Fig. 11 Examples of our 3D cartoon face style editing. (a) and (d) Original generated 3D cartoon faces, (b) and (e) edited results of style transfer, and (c) and (f) target styles.**

## 6 Conclusion

We propose 3D face cartoonizer, the first automatic solution that directly generate a high-quality 3D cartoon face from a single facial image. A hybrid dataset and a novel training strategy are proposed to help us achieve this. Based on our synthesis technique, we achieve easily style editing and animation of the generated 3D cartoon faces, enabling more applications for game production and virtual communication. Furthermore, the idea of leveraging a hybrid dataset and the k-means based editing method is a good trade-off among high-quality, various types, and low cost, and may be extended to other kinds related tasks in the future.

**Fig. 8 Gallery of our generated 3D cartoon faces.**

## Acknowledgment

This work was supported by the National Key R&D Program of China (No. 2018YFA0704000), the Beijing Natural Science Foundation (No. M22024), the National Natural Science Foundation of China (No. 62021002), and the Key Research and Development Project of Tibet Autonomous Region (No. XZ202101ZY0019G). This work was also supported by the Institute for Brain and Cognitive Sciences, BNRist, Tsinghua University, BLBCI, and Beijing Municipal Education Commission.

## References

[1] J. L. Gong, Y. Hold-Geoffroy, and J. W. Lu, AutoToon: Automatic geometric warping for face cartoon generation, in *Proc. IEEE Winter Conf. Applications of Computer Vision*, Snowmass, CO, USA, 2020, pp. 360–369.

[2] K. D. Cao, J. Liao, and L. Yuan, CariGANs: Unpaired photo-to-caricature translation, *ACM Trans. Graph.*, vol. 37, no. 6, p. 244, 2018.

[3] Y. C. Shi, D. Deb, and A. K. Jain, WarpGAN: Automatic caricature generation, in *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 10762–10771.

[4] X. G. Han, C. Gao, and Y. Z. Yu, DeepSketch2Face: A deep learning based sketching system for 3D face and caricature modeling, *ACM Trans. Graph.*, vol. 36, no. 4, p. 126, 2017.

[5] X. G. Han, K. C. Hou, D. Du, Y. D. Qiu, S. G. Cui, K. Zhou, and Y. Z. Yu, CaricatureShop: Personalized and photorealistic caricature sketching, *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 7, pp. 2349–2361, 2020.

[6] R. C. C. Vieira, C. A. Vidal, and J. B. Cavalcante-Neto, Three-dimensional face caricaturing by anthropometric distortions, in *Proc. XXVI Conf. Graphics, Patterns and Images*, Arequipa, Peru, 2013, pp. 163–170.

[7] Y. L. Xing and J. Zhu, Deep learning-based action recognition with 3D skeleton: A survey, *CAAI Trans. Intell. Technol.*, vol. 6, no. 1, pp. 80–92, 2021.

[8] A. Wani, S. Revathi, and R. Khaliq, SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL), *CAAI Trans. Intell. Technol.*, vol. 6, no. 3, pp. 281–290, 2021.

[9] Q. Y. Wu, J. Y. Zhang, Y. K. Lai, J. M. Zheng, and J. F. Cai, Alive caricature from 2D to 3D, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 7336–7345.

[10] V. Blanz and T. Vetter, A morphable model for the synthesis of 3D faces, in *Proc. 26$^{th}$ Ann. Conf. Computer Graphics and Interactive Techniques*, Los Angeles, CA, USA, 1999, pp. 187–194.

[11] C. Cao, Y. L. Weng, S. Zhou, Y. Y. Tong, and K. Zhou, FaceWarehouse: A 3D facial expression database for visual computing, *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 3, pp. 413–425, 2014.

[12] Y. D. Qiu, X. J. Xu, L. T. Qiu, Y. Pan, Y. S. Wu, W. K. Chen, and X. G. Han, 3DCaricShop: A dataset and a baseline method for single-view 3D caricature face reconstruction, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021, pp. 10236–10245.

[13] H. R. Cai, Y. D. Guo, Z. Peng, and J. Y. Zhang, Landmark detection and 3D face reconstruction for caricature using a nonlinear parametric model, *Graph. Models*, vol. 115, p. 101103, 2021.

[14] T. Lewiner, T. Vieira, D. Martínez, A. Peixoto, V. Mello, and L. Velho, Interactive 3D caricature from harmonic exaggeration, *Comput. Graph.*, vol. 35, no. 3, pp. 586–595, 2011.

[15] Z. P. Ye, M. F. Xia, Y. N. Sun, R. Yi, M. J. Yu, J. Y. Zhang, Y. K. Lai, and Y. J. Liu, 3D-CariGAN: An end-to-end solution to 3D caricature generation from normal face photos, *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 4, pp. 2203–2210, 2023.

[16] Y. D. Guo, L. Jiang, L. Cai, and J. Y. Zhang, 3D magic mirror: Automatic video to 3D caricature translation, arXiv preprint arXiv: 1906.00544, 2019.

[17] J. F. Liu, Y. Q. Chen, C. Y. Miao, J. J. Xie, C. X. Ling, X. Y. Gao, and W. Gao, Semi-supervised learning in reconstructed manifold space for 3D caricature generation, *Comput. Graph. Forum*, vol. 28, no. 8, pp. 2104–2116, 2009.

[18] Z. P. Ye, M. F. Xia, Y. N. Sun, R. Yi, M. J. Yu, J. Y. Zhang, Y. K. Lai, and Y. J. Liu, 3D-CariGAN: An end-to-end solution to 3D caricature generation from normal face photos, *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 4, pp. 2203–2210, 2023.

[19] F. Z. Han, S. Q. Ye, M. M. He, M. L. Chai, and J. Liao, Exemplar-based 3D portrait stylization, *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 2, pp. 1371–1383, 2023.

[20] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, Making faces, in *Proc. 25$^{th}$ Annu. Conf. Computer Graphics and Interactive Techniques*, Orlando, FL, USA, 1998, pp. 55–66.

[21] H. T. Yang, H. Zhu, Y. R. Wang, M. K. Huang, Q. Shen, R. G. Yang, and X. Cao, FaceScape: A large-scale high quality 3D face dataset and detailed riggable 3D face prediction, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 601–610.

[22] T. Weise, S. Bouaziz, H. Li, and M. Pauly, Realtime performance-based facial animation, *ACM Trans. Graph.*, vol. 30, no. 4, p. 77, 2011.

[23] C. Cao, H. Z. Wu, Y. L. Weng, T. J. Shao, and K. Zhou, Real-time facial animation with image-based dynamic avatars, *ACM Trans. Graph.*, vol. 35, no. 4, p. 126, 2016.

[24] L. Y. Mo, H. K. Li, C. Y. Zou, Y. B. Zhang, M. Yang, Y. H. Yang, and M. K. Tan, Towards accurate facial motion retargeting with identity-consistent and expression-exclusive constraints, in *Proc. 36th AAAI Conf. Artificial Intelligence*, Palo Alto, CA, USA, 2022, pp. 1981–1989.

[25] T. Y. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, Learning a model of facial shape and expression from 4D scans, *ACM Trans. Graph.*, vol. 36, no. 6, p. 194, 2017.

[26] B. Chaudhuri, N. Vesdapunt, and B. Y. Wang, Joint face detection and facial motion retargeting for multiple faces, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 9719–9728.

[27] K. Y. Chen, J. M. Zheng, J. F. Cai, and J. Y. Zhang, Modeling caricature expressions by 3D blendshape and dynamic texture, in *Proc. 28th ACM Int. Conf. Multimedia*, Seattle, WA, USA, 2020, pp. 3228–3236.

[28] M. Guo, S. F. Wang, Z. B. Wang, M. Lu, X. F. Cui, X. Ling, and F. Xu, 3D face cartoonizer: Generating personalized 3D cartoon faces from 2D real photos with a hybrid dataset, in *Proc. 2nd CAAI Int. Conf. Artificial Intelligence*, Beijing, China, 2022, pp. 356–367.

[29] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York, NY, USA: Wiley, 1973.

[30] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, Encoding in style: A StyleGAN encoder for image-to-image translation, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, TN, USA, 2021, pp. 2287–2296.

[31] Z. B. Wang, J. W. Ling, C. Z. Feng, M. Lu, and F. Xu, Emotion-preserving blendshape update with real-time face tracking, *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 6, pp. 2364–2375, 2022.

[32] R. W. Sumner and J. Popović, Deformation transfer for triangle meshes, *ACM Trans. Graph.*, vol. 23, no. 3, pp. 399–405, 2004.

[33] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H. P. Seidel, Laplacian surface editing, in *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Processing*, Nice, France, pp. 175–184, 2004.

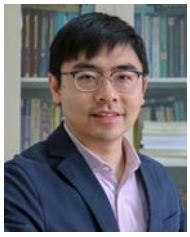
[34] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, Deep residual learning for image recognition, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 770–778.

[35] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, CBAM: Convolutional block attention module, in *Proc. $15^{th}$ European Conf. Computer Vision*, Munich, Germany, 2018, pp. 3–19.

[36] L. Y. Liu, H. M. Jiang, P. C. He, W. Z. Chen, X. D. Liu, J. F. Gao, and J. W. Han, On the variance of the adaptive learning rate and beyond, presented at the 8th Int. Conf. Learning Representations, Addis Ababa, Ethiopia, 2020.

[37] M. R. Zhang, J. Lucas, G. Hinton, and J. Ba, Lookahead optimizer: $k$ steps forward, 1 step back, in *Proc. $33^{rd}$ Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2019, p. 861.

**Ming Guo** received the BEng degree from Shanghai Jiao Tong University, China in 2020. Currently, she is a master student in software engineering at Tsinghua University, Beijing, China. Her research interests include computer graphics and 3D vision.



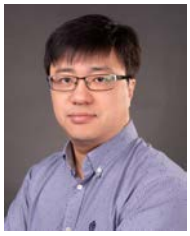
**Feng Xu** received the BS degree in physics from Tsinghua University, Beijing, China in 2007, and the PhD degree in automation from Tsinghua University, Beijing, China in 2012. He is currently an associate professor at School of Software, Tsinghua University. His research interests include face animation, performance capture, and 3D reconstruction.

**Shunfei Wang** received the BEng and MEng degrees from Nanjing University of Aeronautics and Astronautics, China in 2014 and 2017, respectively. He is a senior algorithm engineer at Guangdong OPPO Mobile Telecommunications Corp. Ltd. His research interests include computer vision and 3D vision.

**Zhibo Wang** received the BEng degree in microelectronics science and technology from Nanjing University, Nanjing, China in 2017. He is a PhD candidate in software engineering at Tsinghua University, Beijing, China. His research interests include facial animation and face reconstruction.

**Ming Lu** received the PhD degree in information and communication engineering from Tsinghua University, Beijing, China in 2019. He is currently a researcher at Intel Labs, Intel, China. His research interests include computer graphics and 3D vision. He is particularly interested in improving the workloads at crucial visual synthesis systems, such as AI + Chips (ISP/Codec/GPU), AIGC, and neural field.

**Xiufen Cui** received the MEng degree from Zhejiang University, China in 2017. She is a multimedia algorithm expert at Guangdong OPPO Mobile Telecommunications Corp. Ltd. Her research interests include digital human, XR, and 3D vision.



**Xiao Ling** received the MEng degree from Nanjing University, China in 2007. He is the director of Game and Graphics Development Group in Software Engineering Division at Guangdong OPPO Mobile Telecommunications Corp. Ltd. His research interests include graphics, 3D vision, and XR.