Pacific Graphics (2021)
M. Okabe, S. Lee, B. Wuensche, and S. Zollmann (Editors)

Short Paper

3D-CariNet: End-to-end 3D Caricature Generation from Natural Face Images with Differentiable Renderer

Meijia Huang[1], Ju Dai[2], Junjun Pan[†1,2], Junxuan Bai[1,2] and Hong Qin[‡3]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[2] Peng Cheng Laboratory [3] Stony Brook University

Abstract

Caricatures are an artistic representation of human faces to express satire and humor. Caricature generation of human faces is a hotspot in CG research. Previous work mainly focuses on 2D caricatures generation from face photos or 3D caricature reconstruction from caricature images. In this paper, we propose a novel end-to-end method to directly generate personalized 3D caricatures from a single natural face image. It can create not only exaggerated geometric shapes, but also heterogeneous texture styles. Firstly, we construct a synthetic dataset containing matched data pairs composed of face photos, caricature images, and 3D caricatures. Then, we design a graph convolutional autoencoder to build a non-linear colored mesh model to learn the shape and texture of 3D caricatures. To make the network end-to-end trainable, we incorporate a differentiable renderer to render 3D caricatures into caricature images inversely. Experiments demonstrate that our method can achieve 3D caricature generation with various texture styles from face images while maintaining personality characteristics.

CCS Concepts

• ***Computing methodologies*** → ***Image processing; Mesh geometry models;***

1. Introduction

A caricature is an art form of depicting persons by exaggerating or abstracting the features of a face. As a medium to convey sarcasm or humor, caricatures are mainly used in daily life, such as video games, entertainment activities, advertisements, and personalized avatars in the film. The creation of caricature is complicated, and it is often designed carefully by the artists. With the development of deep learning, 2D caricatures generation [SDJ19, JJJ*21] using image translation and GAN has gained impressive performance in computer graphics. However, many scenarios, including virtual reality, cartoon character creation, animated film, 3D printing, etc., show that the 3D caricature is an indispensable representation rather than a 2D caricature. Modeling a high-quality 3D caricature requires professional 3D artists and their labor-intensive and time-consuming work. Therefore, generating expressive 3D face caricatures from minimal input, such as a single natural face image, is a highly demanding but challenging task. However, obtaining 3D caricatures from face images is still a step-by-step process in existing methods, which firstly apply style transfer to face photos to generate caricature images [SDJ19] and then perform 3D reconstruction for caricature images [ZCGP21]. Because of the intrinsically double heterogeneous domain (cross-style and cross-dimension) between face images and 3D caricatures, it is very challenging to construct 3D caricatures from a single face image directly. There are two key issues that need to be addressed in solving the dual cross-domain problems. One is how to define the style of caricatures, and the other is how to develop a medium from 2D domain to 3D space. The caricature style can be divided into geometric shapes and color textures. The shape usually refers to the exaggeration of a person's facial contours and facial features on the basis of maintaining personality characteristics; color texture style is the appearance, for example, Picasso's abstract style or the black and white style of old photos, etc.

2D caricature generation works have defined a variety of caricature styles. To solve the cross-style issue, we make an intermediary bridge between the face image and the 3D caricature, *i.e.*, the caricature image. [SDJ19] decouples the caricature image into content and color style. The two aspects are combined to obtain different styles of caricature. We employ the work to generate diverse styles of 2D caricature on face images, and then use the 3D caricature reconstruction method [ZCGP21] to obtain 3D caricatures, so that the style of our 3D model comes from 2D caricature. We obtain a synthetic dataset at the same time, which is very meaningful. To deal with the cross-dimension problem, we construct a 3D caricature model as a bridge between 2D and 3D domains. The conventional PCA-based face model like 3DMM [BV99] has limited extrapolation ability. It cannot express the exaggerated part that exceeds the normal face threshold or is not 'seen'. Experiments have proved that the auto-encoder based on mesh convolution proposed by [RBSB18] has better reconstruction ability and extrap-

† Corresponding Author
‡ Corresponding Author

DOI: 10.2312/pg.20211387

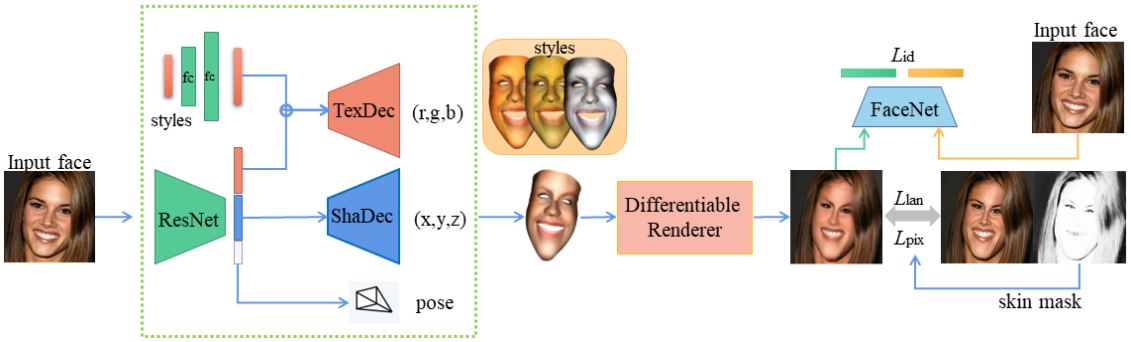

Figure 1: *Overview of our framework. We use Resnet50 to extract embeddings of texture, shape, and pose. Styles are encoded by two fully connected layers. The texture (TexDec) and shape (ShaDec) decoders are respectively used to learn the texture and shape mesh of a 3D caricature. The differentiable renderer renders the 3D caricature into a 2D caricature image. FaceNet aims to maintain identity consistency.*

olation ability than PCA. Inspired by [RBSB18], we propose an autoencoder that joint encodes and separately decodes the shape and texture. The latent vectors of the shape and texture are used as low-dimensional representations of 3D caricatures.

In this paper, we propose an end-to-end network to generate 3D caricatures from face photos. ResNet is employed to extract parameters of the 3D caricature model from face images, and then we use two decoders to restore colored meshes of the caricature. Because the synthetic dataset has different caricature styles, our framework can generate 3D caricatures with various texture styles. However, the 3D synthetic data is not the real ground truth. We only use this data as weak supervision of the network. To refine the generated results and make the network end-to-end trainable, we introduce a differentiable renderer to render the 3D object into an image. Experiments show that our method can quickly generate 3D caricatures of different styles.

The contribution of this article can be summarized as:

- We propose a novel end-to-end neural network that only inputs a face image to quickly generate personalized 3D caricatures with an exaggerated shape and textures of four different styles.
- We construct a synthetic dataset, including 2D natural face images, caricature images, and 3D caricatures. The caricature data consists of four different styles. These synthesized data can be used to assist network training.
- With the graph convolutional auto-encoder, we design a new 3D caricature representation of shape and texture.

2. Related Work

2D Caricature Generation. With the development of GAN and style transfer, 2D caricature generation [CLY18, HHD*18] has achieved impressive performance. Compared with standard face images, caricatures have two main differences: exaggerated shape and color style difference. WarpGAN [SDJ19] generates caricatures from face photos with different exaggerated shapes and various appearance styles. In geometry, WarpGAN predicts a set of control points that could simultaneously distort photos and maintain identities. Furthermore, WarpGAN combines random-styled latent space with the content component of the input face to get different styled caricatures. Similarly, StyleCariGAN [JJJ*21] is a novel caricature generation framework based on StyleGAN [KLA19], which can automatically generate caricature and control the shape degree and color stylization. In particular, it can create more realistic and detailed caricatures. However, these works all focused on the image domain without involving the 3D caricature.

3D Caricature Reconstruction. Following the excellent work on 2D caricature generation, reconstructing 3D caricatures receives increasing attention. Recently, Wu [WZL*18] introduced a 3D caricature geometry model on vertex based on intrinsic deformation representation built from a standard face database FaceWareHouse [CWZ*13]. Zhang [ZCGP21] proposed an automatic caricature reconstruction method. The method first built a database with caricature images and corresponding caricature meshes. It then utilized the caricature model in Wu [WZL*18] to regress the 3D caricature shape and orientation from the input 2D caricature. Due to the lack of high-quality and diversified 3D caricature mesh, Qiu proposed the 3DCariShop [QXQ*21], which contained 2000 3D caricatures manually crafted by professional artists. 3DCariShop was a parametric mesh representation based on PCA, which employed the landmark-guide strategy to reconstruct caricature mesh. While these methods have spared no effort on caricature reconstruction instead of 3D caricature generation from face photos and none of these architectures build a caricature texture model. In contrast, we aim to generate 3D caricatures from face images in an end-to-end manner. To achieve the goal, we first construct a synthetic dataset, based on which we design the mesh model and texture model of 3D caricatures simultaneously. The two models are obtained by training the graph convolutional auto-encoder.

3. Methodology

Given a face photo, we aim to generate a 3D caricature automatically. Our framework is illustrated in the Figure 1. A CNN is employed to regress the coefficients of the 3D caricature model from face images. To further optimize the generated results, we use a differentiable renderer to render the estimated 3D caricature to a 2D image. Since the synthetic 3D data is not the real ground truth, our method is weakly supervised. We will introduce each component in the following chapters. First, the synthetic dataset is explained in chapter 3.1. After constructing 3D caricatures, we train an auto-encoder to obtain shape and texture representations of car-

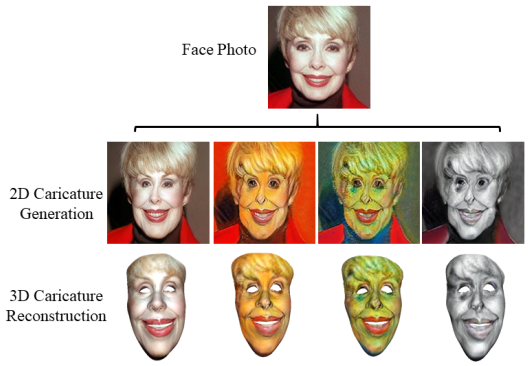

Figure 2: *The process of data preparation.*

icature, which is presented in chapter 3.2. The first two sections are the preparation stage. The details of the entire end-to-end network framework will be introduced in section 3.3. The loss functions that constrain the entire network are described in section 3.4.

3.1. Data Preparation

The dataset contains three parts: face images, caricature images with four different styles, and 3D caricatures with four different styles (including textures). The three types of data have identity consistency. The process of constructing data is shown in Figure 2.

Face Photo. We employ the face dataset CelebA [LLWT15], a large-scale face attribute dataset containing 200k faces, covering large-scale pose changes and different backgrounds.

2D Caricature. We utilize the WarpGAN [SDJ19] to generate 2D caricatures from face images. We produce four styles of caricatures, one that exaggerates the shape without changing the color style, and the other three stylize the color of the skin on top of the exaggerated shape. Since the caricature images generated by the neural network may be blurred or lost important details, we use image enhancement algorithms [WZC*20] to refine the caricature images.

3D Caricature. We use the method [ZCGP21] to reconstruct the 3D mesh from the caricature image. However, it only builds a geometry model without the texture part. To obtain the mesh colors, we project the mesh to get the pixel value from the caricature image for each mesh vertex. Due to the limited accuracy of the 3D reconstruction, the projected mesh may deviate from the face region, which inevitably allows the mesh color to capture the background or hair. Therefore, we deleted the bad 3D data through manual screening. Through the above procedures, we obtain 10k face images, 40k caricature images, and 40k 3D caricatures to train our network end-to-end. As the 2D caricature generated from the neural network is not exaggerated enough, we additionally reconstruct 7k 3D caricatures from caricature images drawn by artists. All the 3D caricature colored meshes are sent into an auto-encoder to learn a 3D caricature representation.

3.2. Caricature Shape and Texture Representation

Mesh Convolution. A mesh with vertices and edges can be defined as a graph $F=(V,A)$, where $V \in R^{n\times F^{in}}$ is a set of n vertices with F^{in} attributes like vertex positions $\{x,y,z\}$ or texture color $\{r,g,b\}$ and $A \in \{0,1\}^{n\times n}$ is a sparse adjacency matrix, which represent the connections status between vertices. The graph Laplacian is defined as $L = D - A$, where D is the diagonal matrix that represents the vertex degree $D_{ii} = \sum_j A_{ij}$. Since L is a real symmetric matrix that can diagonalized by Fourier basis $U = [u_0, u_1, ..., u_{n-1}] \in R^{n\times n}$, we can compute $L = U\Lambda U^T \in R^{n\times n}$, with the non-negative eigenvalues matrix Λ. The Fourier transform of our 3D face representation $x \in R^{n\times F^{in}}$ is defined as $x_w = U^T x$, with the inverse Fourier transform $x = Ux_w$. We formulate the mesh filter g_θ with a Chebyshev polynomial [DBV16] of order K given by

$$g_\theta(L) = \sum_{k=0}^{K-1} \theta_k T_k(L_w). \quad (1)$$

where θ is a vector of Chebyshev coefficients and T_k is the Chebyshev polynomial of order k, which can be recursiverly computed by $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$. Therefore, the spectral convolution can be written as in [DBV16]

$$y_i = \sum_{i=1}^{F_{in}} g_{\theta_{i,j}}(L) x_i \in R^n. \quad (2)$$

where $x \in R^{n\times F_{in}}$ is the input and $y \in R^{n\times F_{out}}$ is the output of the mesh convolution operations.

Mesh Sampling. We utilize the surface simplification algorithm [GH97] especially quadric error metrics to define our sampling operations through two kinds of matrix Q_d and Q_u. To perform down-sampling of a colored mesh with m vertices from n vertices, we employ a binary transformation matrix $Q_d \in \{0,1\}^{n\times m}$, which is calculated by contracting vertex pairs iteratively using quadric matrices. The q-th vertex is kept when $Q_d(p,q) = 1$, or discarded if $Q_d(p,q) = 0, \forall p$ .The Q_u matrices is built during the process of down-sampling, since the q-th vertex is discarded during down-sampling and the vertex is restored in Q_u. The up-sampled mesh with vertices V_u is calculated through $V_u = Q_u V_d$.

Shape & Texture Auto-Encoder. We learned a latent representation of a colored mesh using a graph convolutional auto-encoder that consists of an encoder and two decoders (Figure 3). Both the encoder and decoder contain 4 mesh convolutions with $K = 6$ Chebyshev polynomials. Each of the convolutions is followed by a biased Relu and a sampling operation which reduces or increased the number of mesh vertices by approximately 4 times. The encoder transforms the colored mesh from $R^{n\times 6}$ with input mesh vertices $v = \{x,y,z,r,g,b\}$ to a 256 latent vector using a fully connected layer at the end. The shape decoder reconstructs the mesh $v_g = \{x,y,z\} \in R^{n\times 3}$ from the first 128 numbers of the vector and the texture decoder obtains the texture $v_t = \{r,g,b\} \in R^{n\times 3}$ from another 128 latent parameters. To train the network we employ the $L1$ loss between the predicted colored mesh by contacting the v_g and v_t and the input caricature vertices $v \in R^{n\times 6}$. We utilize the two decoders as our 3D caricature model that can construct colored mesh from the latent vector in the whole architecture.

3.3. 3D Caricature Generation Network

With the dataset and 3D caricature model, we use ResNet [HZRS16] as the face encoder to regress the latent coefficients by modifying the last fully connected layer to 262 neurons, which are further split into three parts 128, 128, and 6. The first 128-dimensional latent embedding is used to generate a mesh, and the other 128-dimensional representation is sent to the texture decoder. To control the texture styles of 3D caricature, we use one-hot encoding to project the styles. For example, the skin texture styles can be expressed as the style vector $(1,0,0,0)$, and the following three

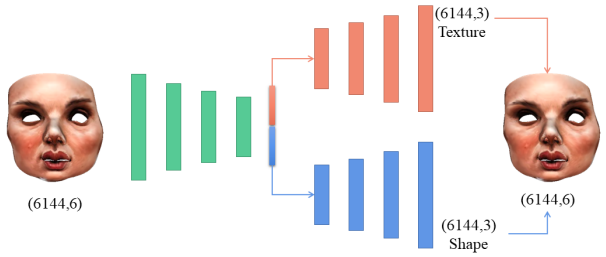

Figure 3: *The auto-encoder of caricature colored mesh.*

styles are green style and orange style, and black-and-white style. We use two fully connected layers to map the four style vectors to the same dimension as the texture model and add them to the texture representation. In this way, we can control the styles of 3D caricatures. The 6-dimensional vector represents the pose.

Camera Model. Since the output of the shape texture decoder is a colored mesh within a unit sphere. A camera model is required to project the 3D mesh from the object-centered Cartesian coordinates into an image plane in the same Cartesian coordinates. We utilize the perspective camera model with an empirically selected camera station and focal length for the projection process. The pose $p \in R^6$ is represented by rotation $R \in SO(3)$ and translation $t \in R^3$.

Differentiable Render. To train our network end-to-end, we employ a differentiable rendering layer, through which we project the 3D caricature into a 2D image with the predicted colored mesh and pose. The differentiable renderer [GCM*18], also known as the rasterizer-based deferred shading model, generates barycentric coordinates and corresponding triangle IDs for each pixel at the image plane. Since the normal and color attributes of mesh vertices are interpolated at the corresponding pixels, gradients can be easily backpropagated through the renderer to the latent parameters that make the architecture trained end-to-end.

FaceNet. In particular, a pre-trained FaceNet is used to extract face and caricature features to guarantee identity consistency which is introduced in detail in section 3.4.

3.4. Loss Function

Pixel-wise Loss. A straightforward way to measure the photometric discrepancy is to compute the difference between the rendered image and the input caricature. We formulated a pixel loss based on the L2 norm. Since caricature images may have occlusions like glasses, we introduce a skin attention mask following [JR02b] to focus on the face region. The pixel-wise loss is defined as:

$$L_{pix}(I,I') = \frac{\sum_{i \in M_{proj}} M_i \left\| I_i - I'_i \right\|_2}{\sum_{i \in M_{proj}} M_i}. \tag{3}$$

where I is the 2D caricature and I' is the rendered image, especially i is the pixel index, M_{proj} is the face area and M is the skin mask.

Identity-Preserving Loss. Although the rendered image may be similar to the input caricature image through the constrain of pixel-level loss during the training, the generated 3D caricature might not look like the input face photo, especially under extreme circumstances. To ensure identity consistency and lower photometric errors with smoother textures, we introduce an identity-preserving loss. We use FaceNet [SKP15] to extracted a 128-feature vector for the input face image and the rendered image, and we compute the cosine distance to measure the similarity. The function is:

$$L_{id}(I_f, I') = 1 - \frac{\langle F(I_f), F(I') \rangle}{\left\| F(I_f) \right\| . \left\| F(I') \right\|}. \tag{4}$$

where I_f is the input face photo, I' is the rendered caricature image and $F(.)$ denotes the feature encoding by FaceNet. In particular, $\langle F(I_f), F(I') \rangle$ is the inner product.

Landmark Loss. The alignment error is treated as the point-to-point distance between the 68 2D landmarks of the input caricature and the projected 3D landmarks, which are specially labeled on the caricature template mesh. To obtain the accurate 2D landmark positions of a caricature, we employ the method in [YNS19] which is especially proposed for caricature landmark detection. Since the 3D landmark indexes are fixed on the mesh vertex, the 3D landmark locations heavily rely on the shape of the mesh and pose. Therefore, the landmark loss can better constrain the parameters related to geometry and the camera. The landmark term is in the following:

$$L_{lan} = \frac{1}{n} \sum_{i=1}^{n} \left\| q_i - q'_i \right\|_2. \tag{5}$$

where q_i indicates the i-th landmark position of 2D caricature and q'_i is the 3D landmark projection location.

In summary, we constrain our network by minimizing weighted combination of above loss terms in the following:

$$L = \lambda_{pix} L_{pix} + \lambda_{id} L_{id} + \lambda_{lan} L_{lan}. \tag{6}$$

4. Experiments

4.1. Training Details

Since our 3D caricature model is constructed by neural networks instead of the PCA morph model, we firstly pre-train the colored mesh auto-encoder introduced at Section 3.2 for 300 epochs with a learning rate of 8e-3 and a learning rate decay of 0.99 . We use stochastic gradient descent to optimize the L1 loss between predicted colored mesh and input samples. The training data is the 3D caricature colored mesh which is represented as $v \in \{x, y, z, r, g, b\}$. We use the two decoders as our 3D caricature model and the trained parameters as the initial parameters for the entire framework.

For the whole pipeline, the inputs are the face photo and 2D caricatures of 4 styles, and the outputs are 3D caricatures of 4 styles. We detect and crop out the face region of input images, which are then resized to 224×224. Since our framework is too large and difficult to train, we pre-train the Resnet for 10 epochs by minimizing the L2 loss between the predicted 3D caricature and the colored mesh we constructed. The weights pre-trained in auto-encoder are as two decoders initialization. Then we trained the whole pipeline with the pre-trained weights for 200 epochs using the ADAM optimizer with lr = 1e-4. Our algorithm is implemented in TensorFlow and trained on a GTX 2080Ti. Furthermore, we set different weights for the losses, where $L_{pix} = 2.5$, $L_{id} = 0.2$ and $L_{lan} = 1e-6$.

4.2. Generated 3D Caricatures

During testing, only one face image is needed to generate 3D caricatures. In particular, our method can capture personality character-

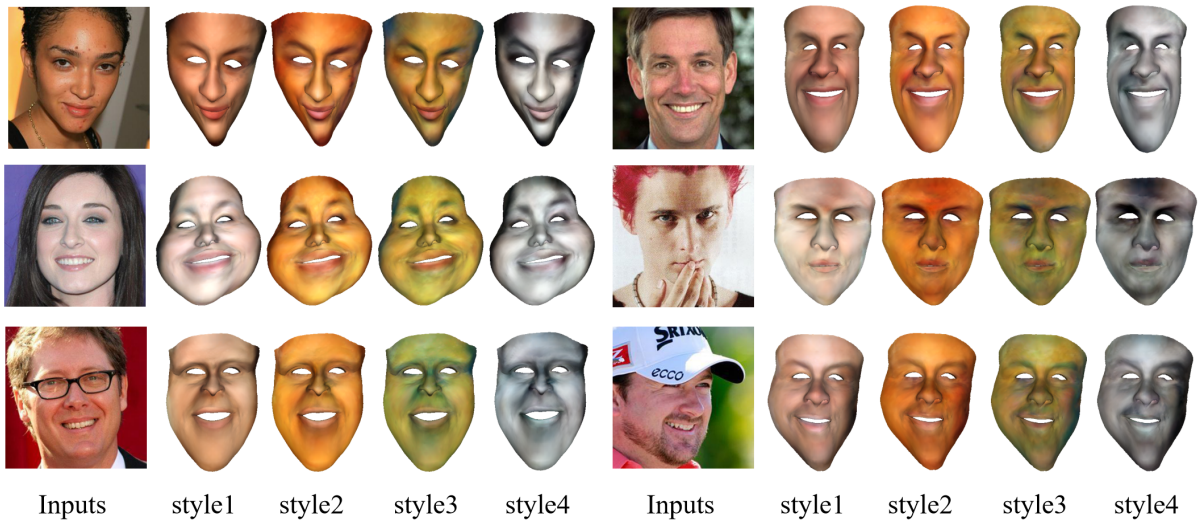

Figure 4: *Results of our algorithm.*

istics, exaggerate face shapes, and transform texture styles of portrait images. For example, pointy chin faces produce more pointy chin faces, and long faces produce longer faces, which can be observed in the first row of Figure 4. Furthermore, when face images are collected without controlled environments, such as wearing glasses or hats and strong lights, our network can still achieve a good appearance, demonstrating the powerful generation abilities when facing challenging inputs.

4.3. Comparisons

Runtime. Our method is an end-to-end network without the process of reconstruction or face image stylization. The simplest way to create the 3D exaggerated face can be achieved by combining several steps we conducted in section 3.1. To verify the time efficiency of our framework, we conduct step-by-step experiments and compute the running time of every stage. Then we compute the total time cost with each step for comparison. The results are displayed in Table 1. We can observe that the total running time of the step-by-step method is 3.343s, while our method only needs 0.233s, which verifies the high efficiency and superiority of our method.

Expression Ability. We change the caricature decoders to the BFM face model [JR02a] in our architecture, which can not generate a face with an abnormal shape like a pointed chin or a very long face as we can see in the Figure 5. The PCA-based linear model BFM can only generate faces within the normal range and does not have the ability to exaggerate faces. Experiments show that our model has outstanding extrapolation capabilities, which is able to express extremely exaggerated geometric shapes.

4.4. Ablation Study

To demonstrate the effectiveness of each loss term in our network, we perform an ablation study with different losses in Figure 6. The pixel loss only ensures the appearance of an exaggerated face, but the contours of the generated caricature are rather rough, which is far from sufficient to express a caricature. The landmark loss constrains that the generation results have a smooth facial geometry,

Methods	Time(s)
2D caricature generation	3.113s
3D mesh reconstruction	0.048s
Colored mesh	0.185s
Sum time of above	3.343s
Ours	0.233s

Table 1: *Runtime comparison of step-by-step methods and our algorithm.*

making the outputs relatively good. However, it is still less expressively with pixel loss and landmark loss. To further enhance the quality of the generated 3D caricature, we employ FaceNet to compute the identity loss to guarantee the identity consistency between the input image and the 3D output of our method. As we can see in the figure, the generations of the three-loss combined are much more detailed. In particular, the mouse part is closed with identity loss which is the same as the input, while the result without identity loss is a little open in the third line. It demonstrates that our generated 3D caricatures preserve the identity well with the FaceNet. We successively add pixel loss, landmark loss, and identity loss to the objective function.

5. Conclusion

This paper presents an end-to-end automatic framework that transforms a normal face photo into a styled 3D caricature with texture, which is an extremely cross-domain task of dimension and style. We build a caricature shape model and a texture model, which are two decoders actually trained in the GCN auto-encoder. Our algorithm is self-learned without the supervision of 3D caricature ground truth. In particular, our 3D caricature generation system only needs one natural face photo as input and outputs a 3D face with exaggerated shape and detailed texture while preserving the people's identity of the face image.

Our technique still has limitations. Firstly, caricature is a styled

Figure 5: *Comparison of our decoders and PCA model.*

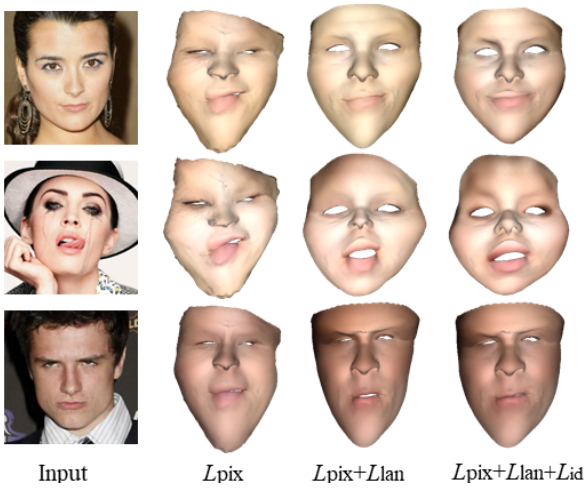

Figure 6: *Comparison of our method with different losses.*

form of the face, and different artists can create different caricature styles. Our algorithm can only generate 3D caricatures whose styles are only from our 2D caricature images. We will conduct experiments to change the style vector and improve the algorithm to generate more styles in the future. In particular, a user study will be employed to verify the generation ability of identity consistency and exaggeration. Another limitation is that occasionally, the shape of generated 3D caricatures may not be exaggerated enough and expressive, since the 2D caricatures of training data are not always of high quality. We will solve this limitation in future work. A considerable solution is to use the improved 2D caricature generation method or caricatures drawn by artists rather than the generation of neural networks. Furthermore, we will try to generate 3D caricatures of different exaggeration degrees by augmenting the dataset to better express satire or humor.

6. Acknowledgements

This work was supported by National Natural Science Foundation of China(No.61872020, U20A20195), Beijing Natural Science Foundation Haidian Primitive Innovation Joint Fund (L182016), Shenzhen Research Institute of Big Data, Shenzhen, 518000, China Postdoctoral Science Foundation (2020M682827), Baidu academic collaboration program, and Global Visiting Fellowship of Bournemouth University.

References

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3d faces. In *Conference on Computer graphics and interactive techniques* (1999), pp. 187–194. 1

[CLY18] CAO K., LIAO J., YUAN L.: Carigans: Unpaired photo-to-caricature translation. In *ACM Transactions on graphics* (2018). 2

[CWZ*13] CAO C., WENG Y., ZHOU S., TONG Y., ZHOU K.: Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2013), 413–425. 2

[DBV16] DEFFERRARD M., BRESSON X., VANDERGHEYNST P.: Convolutional neural networks on graphs with fast localized spectral filtering. *NeurIPS 115*, 101103 (2016), 3844–3852. 3

[GCM*18] GENOVA K., COLE F., MASCHINOT A., SARNA A., VLASIC D., FREEMAN W. T.: Unsupervised training for 3d morphable model regression. In *cvpr* (2018), p. 8377–8386. 4

[GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *Conference on Computer graphics and interactive techniques* (1997), pp. 209–216. 3

[HHD*18] HAN X., HOU K., DU D., QIU Y., CUI S., ZHOU K., YU Y.: Caricatureshop: Personalized and photorealistic caricature sketching. *IEEE Transactions on Visualization and Computer Graphics 26*, 7 (2018), 2349–2361. 2

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *CVPR* (2016), p. 770–778. 3

[JJJ*21] JANG W., JU G., JUNG Y., YANG J., TONG X., LEE S.: Stylecarigan: Caricature generation via stylegan feature map modulation. In *Proc. SIGGRAPH 2021* (2021). 1, 2

[JR02a] JONES M. J., REHG J. M.: Morphable face models - an open framework. *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018) 46*, 1 (2002), 81–96. 5

[JR02b] JONES M. J., REHG J. M.: Statistical color models with application to skin detection. *International Journal of Computer Vision 46*, 1 (2002), 81–96. 4

[KLA19] KARRAS T., LAINE S., AILA T.: A style-based generator architecture for generative adversarial networks. In *CVPR* (2019), pp. 4401–4410. 2

[LLWT15] LIU Z., LUO P., WANG X., TANG X.: Deep learning face attributes in the wild. In *ICCV* (2015), pp. 3730–3738. 3

[QXQ*21] 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 *CVPR* (2021), pp. 10236–10245. 2

[RBSB18] RANJAN A., BOLKART T., SANYAL S., BLACK M. J.: Generating 3d faces using convolutional mesh autoencoders. In *ECCV* (2018), pp. 704–720. 1, 2

[SDJ19] SHI Y., DEB D., JAIN A. K.: Warpgan: Automatic caricature generation. In *CVPR* (2019), pp. 10762–10771. 1, 2, 3

[SKP15] SCHROFF F., KALENICHENKO D., PHILBIN J.: Facenet: A unified embedding for face recognition and clustering. In *CVPR* (2015), pp. 815–823. 4

[WZC*20] WAN Z., ZHANG B., CHEN D., ZHANG P., CHEN D., LIAO J., WEN F.: Bringing old photos back to life. In *CVPR* (2020), pp. 2747–2757. 3

[WZL*18] WU Q., ZHANG J., LAI Y.-K., ZHENG J., CAI J.: Alive caricature from 2d to 3d. In *CVPR* (2018), pp. 7336–7345. 2

[YNS19] YANIV J., NEWMAN Y., SHAMIR A.: The face of art: landmark detection and geometric style in portraits. In *ACM Transactions on graphics* (2019), pp. 1–15. 4

[ZCGP21] ZHANG J., CAI H., GUO Y., PENG Z.: Landmark detection and 3d face reconstruction for caricature using a nonlinear parametric model. *Graphical Models 115*, 101103 (2021). 1, 2, 3