
Article:
Li, S., Su, S., Lin, J. et al. (2 more authors) (2021) Deep 3D caricature face generation with identity and structure consistency. Neurocomputing, 454. pp. 178-188. ISSN 0925-2312

https://doi.org/10.1016/j.neucom.2021.05.014

Deep 3D Caricature Face Generation with Identity and Structure Consistency

Song Li[a], Songzhi Su[a,*], Juncong Lin[a], Guorong Cai[b], Li Sun[c]

[a]*School of Informatics, Xiamen University, China*
[b]*School of Computer Engineering, Jimei University, China*
[c]*Department of Computer Science, University of Sheffield, UK*

Abstract

This paper proposed a novel approach to generate face caricatures automatically from a single portrait image. We decompose the process of 3D face caricatures generation into two independent subtasks: appearance transfer of texture and the geometry transfer of mesh. For the appearance transfer, we design a GAN-based network named CariFaceGAN to learn the style mapping from portrait to caricature, in which facial features are leveraged to preserve identity consistency. For geometry transfer, we first learn the transformation of the landmarks between portraits and caricatures in an embedded space obtained with Locally Linear Embedding method, and then Kriging interpolation is used to manipulate the portrait mesh constructed from a single image. The experimental results show that our proposed CariFaceGAN outperforms the state-of-the-art methods in terms of maintaining identity consistency and providing satisfactory visual effects.

Keywords: Style transfer, Generative Adversarial Networks, 3D face mesh, Caricature

1. Introduction

Caricature is a kind of artistic work that describes a person with exaggerated characteristics to create a comic or grotesque effect. Since it can highlight people's personality, caricature is widely used in the entertainment

*Corresponding author at: School of Informatics, Xiamen University, Xiamen 361001, China.

Email address: ssz@xmu.edu.cn (Songzhi Su)

May 17, 2021

industry, e.g. movies, television works, and games, etc. In recent years, with the rapid development of the computer animation industry, there is a high demand for 3D caricatures. Compared to 2D caricatures, 3D caricatures are more effective as they can produce images of variational appearance by changing the camera pose, and generate images of richer expressions by deforming the face mesh. Moreover, our proposed method is able to automatically output 3D caricature faces from 2D portrait images, without the need to build 3D face model manually.

In recent years, pioneer researchers start tackling the 3D caricature problem. The previous research [1, 2, 3] need heuristics to craft the face generation, such as manipulating the facial sketch or exaggerating specific parts. Although they can effectively control the process of model generation, it requires skilled operators and these operations are not user-friendly. Other methods like [4, 5, 6, 7, 8] focus on automatic generation of 3D caricature face, whereas the texture of the model is not considered or fixed using a predefined one. Our proposed method uses a discriminative model with local information (k nearest neighbor face keypoints) to generative caricature model, and our model can easily control the degree of exaggeration. For appearance transfer, there are some style transfer methods [9, 10, 11, 12, 13] and cross-domain image transfer methods [14, 15, 16, 17, 18, 19, 20] based on Generative Adversarial Networks (GAN) [21], which theoretically suits the appearance transfer task. However, compared to other applications e.g. landscapes, dogs, and cats, the effectiveness of these methods in transferring portrait to caricature is not significant due to the complexity of facial details. Hence, a significant evolution is required to advance the state-of-the-art methods for 3D face caricature. Based on the loss functions of MUNIT and CariGANs, we add identity loss to ensure the identity consistency of deep facial features, which can generate new face style but preserve the identity information.

In addition, there exist other challenges in 3D face caricature. Firstly, the ground truth of 3D caricature face is scarce as drawing 3D caricature faces manually is time-consuming for 3D modelers. Secondly, different from 3D face reconstruction based on 3D Morphable Models (3DMM) [22], which can be accomplished by refining the parameters of Basel Face Model (BFM) [23] to match the input 2D portrait, 3D caricature face does not have any parameterized morphable models since the styles of caricature are varied by different artists.

Following [25, 8, 18], we decouples the task into two subtasks of appear-

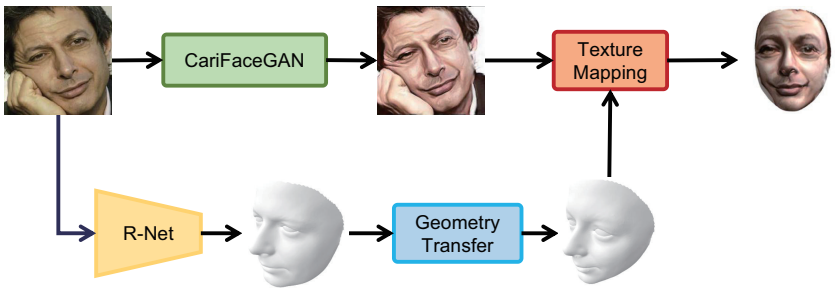

Figure 1: The System overview. We proposed a two-stage method to generate 3D caricature from a single image. First, we train a network named CariFaceGAN to transform portrait image into caricature image. Then we use R-NET [24] to build portrait mesh and design a pipeline to deform the mesh. Finally, the 3D caricature face can be obtained by combining the deformed mesh with the caricature texture. The input image is from WebCaricature dataset.

ance transfer and geometry transfer. Similar to style transfer of image, the goal of the appearance transfer is to transform the portrait style of facial photos to caricature style. We designed a network named CariFaceGAN inspired by MUNIT [17], which can translate the portrait to caricature with the maximization of preserving the facial identity information. The goal of the geometry transfer is to exaggerate and deform the 3D portrait mesh to generate the 3D caricature mesh. By using Locally Linear Embedding (LLE) [26], we embed the 2D landmarks of these two domains into low dimensional space to learn the mapping function and then use Kriging interpolation [27] to finish the deformation. Finally, the 3D caricature face can be obtained by simply combining the 3D caricature mesh with the caricature style texture.

To summarize, main contributions of this paper are:

- We propose a CariFaceGAN appearance transfer network utilizing FaceNet [28] to supervise the generation process of the caricature image, which can enhance the identity consistency between the original face and the generated caricature face. It is novel to use deep face features to supervise the generation of caricature images.

- We propose a novel approach that leverages an intermediate domain to eliminate the geometric variance in the domain transferring from portraits to caricatures. This intermediate representation provides an underlying geometrical constraint between the original and target domain, thereby facilitating the training of CariFaceGAN.

- We propose a geometry transfer pipeline that can generate 3D caricature mesh by inputting a portrait image. Our method is fully automatic without any need for the ground truth of 3D caricatures and user participation.

2. Related Work

2.1. 3D Face Reconstruction.

The task of 3D face reconstruction is to predict a complete facial structure with texture from single or multiple images. Since 3DMM[22] was proposed in 1999, it has been the technical basis of this task and a vast array of CNN based methods [29, 30, 31, 24, 32, 33, 34, 29, 35] were proposed to predict 3DMM coefficients to reconstruct 3D face models in recent years. In addition, in order to break through the limitation of 3DMM, some model-free methods [36, 37] were proposed. Although they achieve great results in face reconstruction, the appearance and geometry of portrait and caricature are absolutely different, these methods are not suitable for caricature face generation. [24] proposed R-Net based on ResNet[38] which is used to reconstruct the portrait mesh in this paper.

2.2. 3D Caricature Generation.

2.2.1. Semi-Automatic Methods

For the sake of accuracy, there are some semi-automatic methods that require user operation in the process of generating 3D caricature. [1] proposed a two-step method, which first allows users to drag and drop the average model to get a rough model and its landmarks, and then uses the Kriging Interpolation to refine the face model according to landmarks. [2] extracts the human identity from a 3D human face model, then transfer it to a 3D artistic face model, where the transformation degree can be controlled by user. [3] proposed a sketching system that allows users to change facial feature curves to exaggerate the specific features, and then sketch with 3D portrait face is fed into their network to create a caricature face. Although they can control the process of model generation manually, these operations are unfriendly to users.

2.2.2. Automatic Methods

Many research has studied generating 3D caricatures automatically. [4, 5] use LLE to embed the 2D faces and 3D caricatures into a low dimension

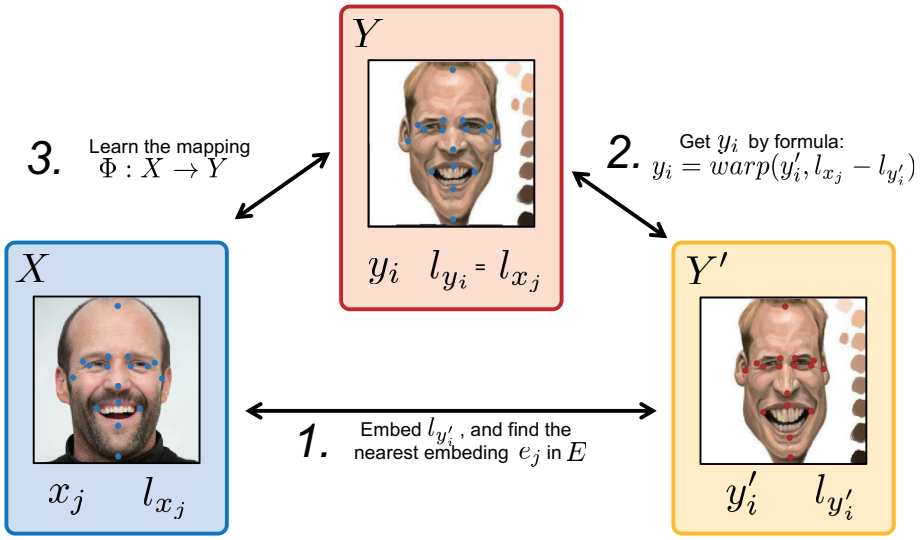

Figure 2: The construction of Intermediate Domain. The images are from WebCaricature dataset.

space, and learn the mapping between these two space. [6] divides the whole face into several regions, and then regresses a set of linear mappings for each region, and finally constructs all transformed regions. [7] proposed a deformation representation coordinates system to formulate the 3D caricature generation as an optimization problem. [8] proposed VAE-CycleGAN to solve this cross-domain problem, which converts a 3D portrait mesh into 3D caricature mesh. [25] builds a PCA model for 3D caricature meshes and trains a GAN which can automatically translate 2D pictures to 3D caricature meshes while the exaggerated shape of meshes can be controlled. Although these methods can generate 3D models well, the required ground truth of 3D caricatures is expensive. In contrast, our method just needs caricature images with landmarks.

2.3. Style Transfer.

[9] first proposed an effective style transfer method from one to another, and many follow-up research has been proposed to improve the quality of transformation. [10] presented a solution for real-time style transformation and perceptual loss to measure high-level perceptual and semantic differences between images. [11] designs an explicit representation for styles, which enables the network to completely extract the style from the content. Owing

to the success of GAN, image translation has made much progress in recent years. [14], [15] and [16] proposed methods to transfer unpaired images from one domain to another. [17] proposed MUNIT, in which the style of the translation outputs can be controlled by a user-provided example image or a random style code in the target domain. Although they can achieve satisfactory results in some domain like landscape paintings or oil paintings, they show limited capability of translating image from portrait to caricature. For caricature generating, [18] and [19] use the perceptual loss to supervise the training of caricature generator which can produce good visual effect, but lack of identity consistency. To tackle this, we use the pretrained FaceNet to enhance the identity consistency.

3. Method

In this section, we demonstrate the framework and the details of our proposed method. As shown in Fig. 1, our goal is to generate a 3D caricature face from a single face image. To tackle this challenging problem, we decompose this task into two subtasks, namely appearance transfer and geometry transfer. For appearance transfer, we designed a cross domain transfer network named CariFaceGAN to transform input portrait into caricature style. And for geometry transfer, we firstly use LLE to embed the landmarks of the face, then learn the mapping between portraits and caricatures, and finally use Kriging interpolation to deform the dense points of the face mesh.

3.1. Appearance Transfer

In this section, we introduce CariFaceGAN, which can transform the appearance style of portrait into caricature with the preservation of identity consistency.

3.1.1. Construct an Intermediate Domain.

To facilitate the neural network to transfer the appearance, we build an intermediate domain which bridges the geometric differences between portrait domain and caricature domain. As shown in Fig. 2, let $x \in X$ denotes an image in portrait domain X, $y' \in Y'$ an image in caricature domain Y', and $y \in Y$ an image in the constructed intermediate domain Y. Using landmarks of 200 portraits from X, where the number of portraits is selected empirically and data dependent, we establish a manifold space. For the rest portraits in X, we calculate the embedding e for each portrait, and E denotes

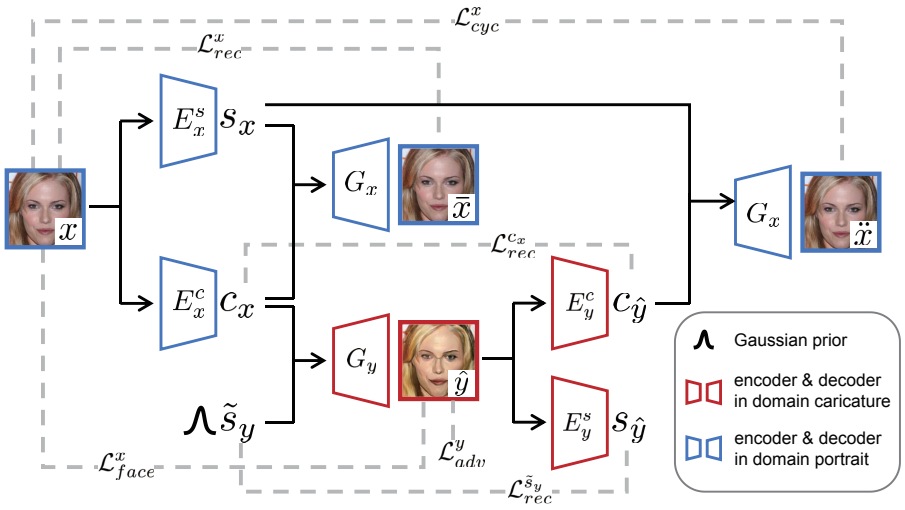

Figure 3: The architecture of CariFaceGAN. For brevity, we only show the network architecture of the mapping $\Phi : X \to Y$. The blue and red arrows from x or y to c or s represent encoder and vice versa. The input image is selected from the WebCaricature dataset.

the collection of e. And for each image y'_i with its landmarks $l_{y'_i}$, we embed $l_{y'_i}$ to the manifold space, and find the nearest embedding e_j in E by Euclidean distance, where j is the index of the nearest embedding. Under the guidance of the displacement between $l_{y'_i}$ and l_{x_j} according to e_j, we use differentiable spline interpolation module from [39] to calculate the displacements on horizontal and vertical. And using the displacements of dense points, we rerender the image l_{y_i} with simple bilinear interpolation. Then we get image domain Y, which is same with Y' in appearance style and similar to X in shape. The goal of CariFaceGAN is to learn the mapping $\Phi : X \to Y$ to transform a portrait style image into caricature style. And for brevity, we will only introduce $\Phi : X \to Y$, and the opposite mapping is the same.

3.1.2. Architecture.

Architecture of CariFaceGAN is shown in Fig. 3. Following MUNIT, we assume that X and Y share the same content space, and that they have their own style space independently, the distribution of the style code is assumed to be standard Gaussian distribution. What we need to do is to extract their content code noted as c and style code noted as s from image, and combine the content code of x with the style code of y to get the target image. To

this end, we train a decoder G and two encoders E^s and E^c for each domain, and for mapping $\Phi : X \to Y$, their relationship can be formulated as follows:

$$\begin{gathered} c_x = E_x^c(x), \quad s_x = E_x^s(x), \quad \tilde{s}_y \in \mathcal{N}(0,1), \\ \bar{x} = G_x(c_x, s_x), \quad \hat{y} = G_y(c_x, \tilde{s}_y) \end{gathered} \tag{1}$$

where $\bar{x}$ is a reconstruction image generated from content code c_x and style code s_x, which are encoding from x by E_x^c and E_x^s. x and $\bar{x}$ should be same because D^x and $E^{c|s}$ are a pair of inverse functions. $\tilde{s}_y$ is sampled from standard Gaussian distribution, and $\hat{y}$ is the target image.

With a large enough scale of network, an input portrait can be mapping to any different random images in the target domain, without considering whether the content of the generated image is same to input ones. Cycle consistency has been proved to be effective way to maintain the content information in cross-domain image transfer problem. In order to use cycle consistency, we reconstruct the input image using the content code from $\hat{x}$ and the style code from x. And this process can be formulated as follows:

$$c_{\hat{y}} = E_y^c(\hat{y}), \quad s_x = E_x^s(x), \quad \ddot{x} = G_x(c_{\hat{y}}, s_x) \tag{2}$$

where $c_{\hat{y}}$ is the content code of $\hat{y}$ encoded by the content encoder of domain Y. $\ddot{x}$ is the reconstructed image from content code $c_{\hat{y}}$ and style code s_x using generator G_x, and cycle consistency will be calculated between $\ddot{x}$ and x.

In summary, there are 4 images in this process, in which x, $\bar{x}$ and $\ddot{x}$ are consistent in theory because they have the same content and style code. $\hat{y}$ is the target image which combined from content code of portrait and style code of caricature. The encoders, decoders and the discriminators have the same structure as MUNIT.

3.1.3. Loss.

The loss of CariFaceGAN consists of four parts as follows.

Adversarial loss. In order to make the image generated by CariFaceGAN indistinguishable from images in caricature domain, we employ adversarial loss to match the translated image $\hat{y}$ to the target caricature distribution.

$$\mathcal{L}_{adv}^y = \mathbb{E}_{x\sim X,\ \tilde{s}_y\sim\mathcal{N}(0,1)}\ [log\ (1 - D_y(\hat{y}(x, \tilde{s}_y))] + \mathbb{E}_{y\sim Y}\ [log\ (D_y(y)] \tag{3}$$

where D_y is the discriminator of domain Y. $\hat{y}(x, \tilde{s}_y)$ is defined in Equation 1.

Table 1: Mainly differences among MUNIT, CariGANs and ours.

Methods	MUNIT	CariGANs	ours
Structure information	×	Generative and global	Discriminant and local
Bidirectional reconstruction loss	✓	×	✓
Cycle loss	×	✓	✓
Perceptual loss	×	✓	×
Identity loss	×	×	✓

Bidirectional reconstruction loss. Reconstruction loss is designed to ensure the encoder and decoder are a pair of inverse function. We employ the bidirectional reconstruction loss for both image → code → image and code → image → code:

$$\begin{aligned}
\mathcal{L}_{rec}^{x} &= \mathbb{E}_{x\sim X}\ [\parallel \bar{x}(x) - x \parallel_1] \\
\mathcal{L}_{rec}^{c_x} &= \mathbb{E}_{x\sim X,\ \tilde{s}_y\sim\mathcal{N}(0,1)}\ [\parallel E_c^y(\hat{y}(x,\tilde{s}_y)) - E_c^x(x) \parallel_1] \\
\mathcal{L}_{rec}^{\tilde{s}_y} &= \mathbb{E}_{x\sim X,\ \tilde{s}_y\sim\mathcal{N}(0,1)}\ [\parallel E_s^y(\hat{y}(x,\tilde{s}_y)) - \tilde{s}_y \parallel_1]
\end{aligned} \tag{4}$$

where $\bar{x}(x)$ is defined in Equation 1.

Cycle loss. As mention above, bidirectional reconstruction loss can be used to limit that encoder and decoder are a pair of inverse function in each domain, and we also need to make sure that (E, G) in two domains are the opposite processes. Cycle consistency reconstructs the source domain image from the target domain, which can ensure the generation processes of these two domains are opposite to each other. We employ cycle loss as follows:

$$\mathcal{L}_{cyc}^{x} = \mathbb{E}_{x\sim X,\ \tilde{s}_y\sim\mathcal{N}(0,1)}\ [\parallel \ddot{x}(x,\tilde{s}_y) - x \parallel_1] \tag{5}$$

where $\ddot{x}(x,\tilde{s}_y)$ is defined in Equation 2.

Identity loss. While only low-level information (pixel-level) is utilized to supervise the process of image transfer, we find that the target image generated by the network loses the facial details of origin input image. Inspired by [40] and [24] using FaceNet to supervise the process of 3D face reconstruction, we adopted it to supervise the process of facial appearance transfer. Specifically, we extract deep facial features and compute the cosine consistency as follows:

$$\begin{aligned}
cos_dist(a,b) &= 1 - \frac{< a, b >}{\parallel a \parallel \cdot \parallel b \parallel} \\
\mathcal{L}_{id}^{x} &= \mathbb{E}_{x\sim X,\ \tilde{s}_y\sim\mathcal{N}(0,1)}\ [cos_dist(\ f(\hat{y}(x,\tilde{s}_y)),\ f(x))]
\end{aligned} \tag{6}$$

where f is the function that embedding the facial image into facial features by FaceNet and $< \cdot, \cdot >$ is the vector inner product. The FaceNet is pretrained by VGGFace2[41] dataset.

Overall loss. Overall, the aforementioned loss function is defined for $\Phi : X \to Y$, and the opposite mapping is the same. The final loss function of our CariFaceGAN is defined as follows:

$$\begin{aligned}\mathcal{L}_{overall} = &\mathcal{L}^{y}_{adv} + \mathcal{L}^{x}_{adv} + w_{rec}(\mathcal{L}^{x}_{rec} + \mathcal{L}^{y}_{rec}) \\ &+ w_{code}(\mathcal{L}^{c_x}_{rec} + \mathcal{L}^{c_y}_{rec} + \mathcal{L}^{\tilde{s}_x}_{rec} + \mathcal{L}^{\tilde{s}_y}_{rec}) \\ &+ w_{cyc}(\mathcal{L}^{x}_{cyc} + \mathcal{L}^{y}_{cyc}) + w_{id}(\mathcal{L}^{x}_{id} + \mathcal{L}^{y}_{id})\end{aligned} \tag{7}$$

where w_{rec}, w_{code}, w_{cyc} and w_{id} are weights to control the importance of loss of each parts. The main differences among MUNIT, CariGANs and CariFaceGAN are summerized in Table 1. Firstly, CariGANs use generative model to construct an intermediate domain with global information, while our proposed method uses discriminative model with local information (k nearest neighbor face keypoints). Secondly, as for the loss function, MUNIT utilize bidirectional reconstruction loss to ensure the encoder and decoder are a pair of inverse function, CariGANs utilize cycle loss to ensure transfer function of two domains are the opposite processes, while we use both these two loss functions. Finally, CariGANs use perceptual loss to keep the content consistency of input and output, aiming at reducing reconstruction error, while we use identity loss to ensure the identity consistency with deep facial feature, aiming at reducing semantic error.

3.1.4. Training detail.

We follow the training strategy of MUNIT. The weights of loss are set as $w_{rec} = 10$, $w_{code} = 1$, $w_{cyc} = 10$ and $w_{id} = 0.8$ respectively in all our experiments. Learning rate begins at $1.5e-4$ and decays 70% after each $50K$ iterations, and $500K$ total iterations. And use the Adam optimizer [42] with a batch size of 2.

3.2. Geometry Transfer

It is challenging to directly recover 3D mesh from a single caricature image. In this section, we present a geometry transfer pipeline to build the 3D caricature mesh.

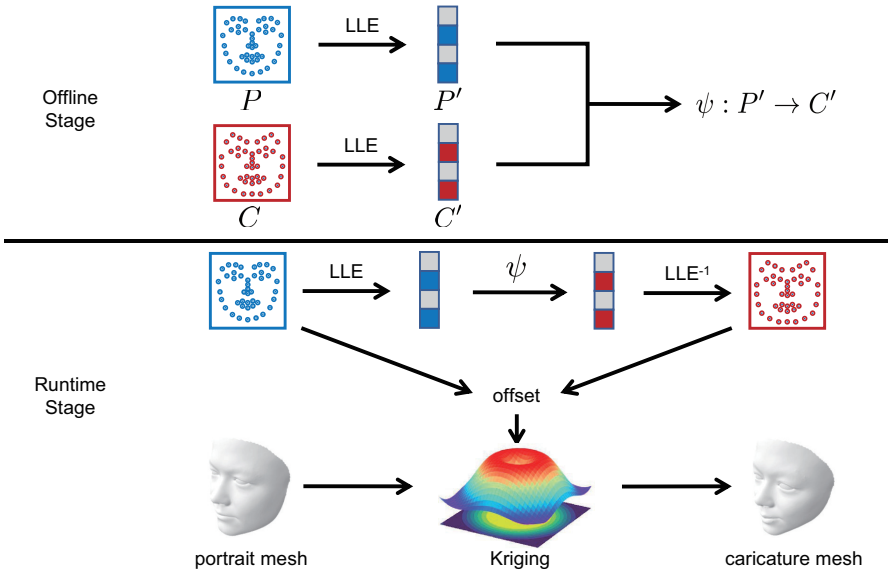

Figure 4: Overview of the pipeline of geometry transfer. The upper part is the offline stage, and the lower part is the runtime stage.

3.2.1. Data preparation.

Unlike the data used in CariFaceGAN, we only use frontal facial portrait and caricature in training of geometry transfer. We collect images in portrait domain and caricature domain, in which images of the same index are paired. In other words, one caricature is paired with one portrait. We use the landmarks detector designed by [43] to detect 68 landmarks d_p for each portrait, and label landmarks d_c manually for each caricature. d_p and d_c are 136-d vectors defined as $[x_0, y_0, x_1, y_1, ...x_{67}, y_{67}]$ where (x_i, y_i) is the 2D coordinates of the ith landmarks. Before using these landmarks, it is necessary to align them to a uniform center point, angle and scale, since these images come from various sources. We use the method designed by [44] to align these landmarks. Finally, P denotes the matrix consisting by aligned d_p, and C by aligned d_c.

3.2.2. Overview.

As shown in Fig. 4. In the offline stage, we use LLE to embed landmarks of faces in two domains and learn the mapping between P and C inspired by [5]. And in the deployment stage, we firstly reconstruct 3D face following [24], in which also outputs the facial landmarks, so we do not need to detect

them using another 2D landmarks detector. Then using the learned mapping, we predict the target caricature landmarks corresponds to the input image. Finally, Kriging interpolation is employed to deform the dense point of mesh.

It is worth noting that in our method, we only use two-dimensional information to manipulate the 3D portrait mesh, that is, we do not manipulate in the z-axis direction. The reason for this is that the ground truth of 3D caricature mesh is rare, and the deformation on the x and y axises can represent the exaggeration to some extent.

3.2.3. Mapping based on Locally Linear Embedding.

In offline stage, we establish a manifold space using LLE for P and C. Then matrix W^P which record the neighborhood weight for each data and d dimension embeddings P' can be calculated, so as W^C and C'. And the least square method is utilized to find the mapping $\psi : P' \rightarrow C'$.

In runtime stage, we denote the input portrait landmarks as p, and the predicted caricature landmarks c can be calculated as follow steps: 1) Calculating the weight vector w^p by minimizing the cost function:

$$\begin{aligned} \epsilon(w^p) &= (p - \textstyle\sum_{j \in n(p)} w_j^p P_j)^2, \\ s.t.\ &\textstyle\sum_{j \in n(p)} w_j^p = 1 \end{aligned} \tag{8}$$

where $n(p)$ denotes the indices of k nearest neighbours of p in P, and w_j^p is the weight summarize the contribution of the j-th data point to the reconstruction of p following [26]. 2) Since P and $P'$are corresponding, the embedding can be calculated by $p' = w^p P'$. 3) By using mapping $\psi : P' \rightarrow C'$, we can get the caricature embedding c'. 4) Searching within C', we also get the k nearest neighbours of c' and calculate the weight vector $w^{c'}$ similar to Equation 8. 5) The predicted caricature landmarks can be calculated by $c = w^{c'} C$. The parameters are set to $k = 5$ and $d = 15$ respectively in our implementation.

3.2.4. Kriging interpolation.

Kriging interpolation [27] is a regression algorithm for spatial modeling and interpolation of stochastic process or random field based on covariance function, which can calculate the best linear unbiased prediction and is applied widely in geostationary research. Inspired by [1], we used 3D Universal Kriging [45] to deform the 3D face mesh into the target 3D caricature mesh. As caricature landmarks c have been predicted, we can calculate the offset

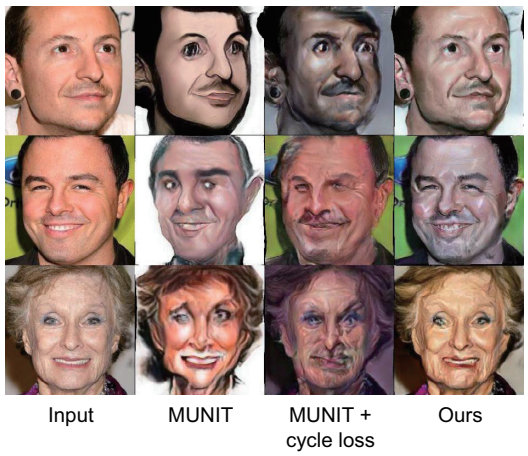

Figure 5: Some qualitative results of the ablation study on CariFaceGAN. We generate these caricatures for each method using randomly sampled style codes. Inputs are selected from CelebA dataset.

$\Delta l = c - p$. And we denote the x axis offset of landmarks as Δl_x, and y as Δl_y. Landmarks in 3D face mesh are regarded as observed points, and Δl_x and Δl_y as observed values respectively. Then the offset of x and y axis of dense vertices can be calculated using the 3D Universal Kriging interpolation. Finally, the target 3D caricature mesh can be calculated by adding the offset to the 3D face mesh.

4. Experimental Results and Analysis

In this section, we evaluate the performance of our proposed method on the tasks of appearance transfer and geometry transfer respectively. We first introduce the dataset used to train and evaluate in our experiments in Sec. 4.1. We then show the performance of our system in Sec. 4.2. Then in Sec. 4.3, we analyze our CariFaceGAN and compare it with other methods. Finally, we compare our geometry transfer with baseline method and show the 3D caricatures built by our proposed method in Sec. 4.4. The results shown in all figures are randomly generated.

4.1. Dataset

WebCaricature WebCaricature is a large portrait-caricature dataset consisting of 6042 caricatures and 5974 photographs from 252 people col-

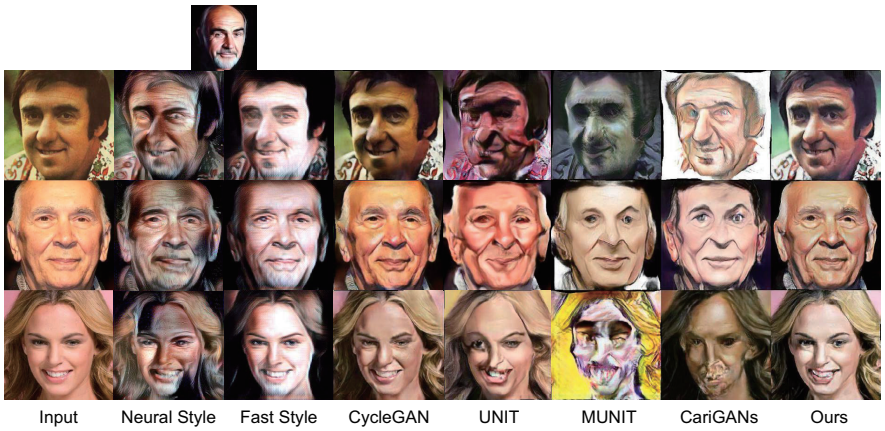

Figure 6: The comparison of state-of-the-art image-to-image translation methods. All these models are trained on the dataset which is used by CariFaceGAN. The style images of Neural Style and Fast Style are shown in the first row. We only use CariStyGAN of CariGANs for appearance transferring. The inputs are selected from CelebA dataset.

lected from the Internet. For each image in this dataset, 17 labelled facial landmarks are provided.

CelebFaces Attributes Dataset CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images and each with 40 attribute annotations. Images in this dataset cover large pose variations and background clutter.

To train the CariFaceGAN, we utilize 3540 caricatures from WebCaricature as domain Y', where we remove 1) caricatures with large poses; 2) similar to the portraits; 3) with a single eye. And select 5000 portraits randomly from CelebA as domain X, whose number of pictures is the same as that of the Y' domain. For all these images, the face region is cropped according to the landmarks and rotated horizontally according to the angle between the binocular line and the horizontal direction. 2000 randomly selected portraits in CelebA are used for testing.

To train the geometry transfer algorithm, we build a paired portrait-caricature mini dataset. We provided a painter with 70 images of frontal portraits, and he created the corresponding caricature for each portrait. We used the landmarks detector[43] to detect 68 landmarks for portraits, and labelled 68 landmarks manually for caricatures.

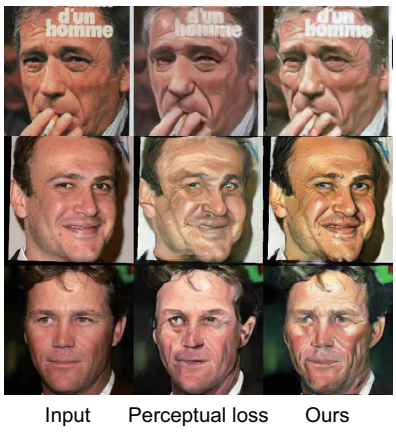

Figure 7: Result of the comparison of facial loss and perceptual loss. Inputs are from CelebA dataset.

4.2. System Performance

Our main code is developed in PyTorch [46]. A desktop with a NVIDIA GeForce RTX 2080Ti and Intel i7 9700k (3.6GHz) CPU is used for evaluation. The total time used to predict a 256×256 image is approximately 0.421 seconds including 0.143 seconds for appearance transfer and 0.278 seconds for geometric transfer.

4.3. Experiments on CariFaceGAN

Our system allows user customization on appearance style as aforementioned. In Fig. 11, we show different results with three random style codes.

4.3.1. Ablation Study.

To verify our loss function used in CariFaceGAN, we design an ablation study on loss function. Fig. 5 present the qualitative result, which shows that cycle loss can retain most of the content information, but it is not effective for some facial details such as eyes and mouths. And after joining the identity loss, the details of the eyes and mouths are well preserved. In addition, in order to verify capability of our method on maintaness of identity consistency, we use pretrained FaceNet to map the original image and its corresponding caricature output to a compact Euclidean space, where distances directly correspond to a measure of face similarity. We calculate the average distance in test set is 0.573±0.136, which is less than the threshold 1.242 defined by FaceNet.

4.3.2. Comparison with State-of-the-Art Methods.

We qualitatively compare our CariFaceGAN with previous state-of-the-art methods in Fig. 6. All these methods are based on the implementations provided by the author except CariGANs [18] which has not released its codes. The dataset used to train these networks is same with us. In these works, Neural Style [9] and Fast Style [10] are CNN-based style transfer network, and the rest are GAN-based cross-domain image transfer network.

Neural Style, the first style transfer network, achieves satisfactory results in this experiment, but sadly it is computationally expensive. Fast Style is an improved version of Neural Style, which can complete the style transfer task in real-time, but the effect on the portrait is not significant, it changes the hue only. CycleGAN [14] and UNIT [16] train a bidirectional conversion network based on GAN, which has more creative freedom than Neural Style and Fast Style, but they are not good at creating portrait and caricature. MUNIT [17] is based on UNIT, which can transfer one portrait into multiple results. We randomly sample style code for MUNIT, and the results show clear artefact. CariGANs [18] based on MUNIT and CycleGAN is the first deep neural network for unpaired portrait-to-caricature translation, which consists of CariGeoGAN for geometry exaggeration and CariStyGAN for appearance stylization. It is worth noting that the training caricatures used by CariStyGAN are deformed by CariGeoGAN, while ours is constructed by the method mentioned in Sec. 3.1, since their training dataset has not released. The result shows that CariStyGAN works properly in most cases, but the identity consistency of input and output images cannot be guaranteed.

We have compared the inference time of our CariFaceGAN with Fast Style, CycleGAN, UNIT, MUNIT, and CariStyGAN of CariGAN, performing on NVIDIA GeForce RTX 2080Ti and Intel i7 9700k (3.6GHz), results are shown in Table 2. The inference time of CycleGAN is minimal, and that of other networks are almost the same. Furthermore, we conduct a user study for Fig. 6 with 50 participants (including 32 artists and art students and 18 people without art background; 22 males and 28 females; aged from 18 to 38) to measure the visual effect of the generated caricatures. We use a score ranging from 0 to 5 to represent the criteria, from the worst to the best. Results are shown in Table 2, and ours achieves the highest score while CariGANs is the second. Kruskal-Wallis test is then used to test the scores of all methods ($P < 0.05$), and Nemenyi test is used in pairwise comparison where the difference between CariGANs and ours is not statistically significant (P

Figure 8: Results of our proposed method. The caricatures in the second column are generated by CariFaceGAN. Portrait meshes in the third column are built by R-NET, and caricature meshes are deformed from portrait meshes using our proposed geometry transfer pipeline. The last three columns are the final 3D caricature. Input images are selected from CelebA and WebCaricature dataset.

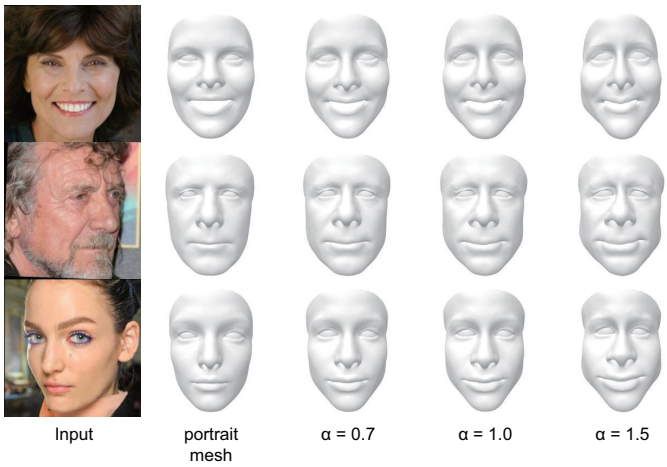

Figure 9: Results of adjusting the geometric exaggeration extent parameter α. The first column on the left shows the input images from CelebA dataset, and the second is the result of face reconstruction using R-NET. The three columns on the right are deformation results under different α. Input images are selected from CelebA dataset.

Table 2: Comparison of inference time, human score and Inception score of the state-of-the-art methods and ours. We only use CariStyGAN of CariGANs for appearance transferring.

Methods	Fast Style	CycleGAN	UNIT	MUNIT	CariGANs	ours
Time/sec	0.179	**0.011**	0.159	0.161	0.143	0.139
Human score	1.24 ± 0.46	2.76 ± 0.83	0.36 ± 0.50	0.24 ± 0.43	3.14 ± 1.23	$\mathbf{3.36 \pm 0.96}$
Inception score	2.29 ± 0.15	3.05 ± 0.08	$\mathbf{3.12 \pm 0.13}$	2.97 ± 0.14	2.92 ± 0.10	3.02 ± 0.17

> 0.05). For the quantitative result, we utilize the Inception score which uses a pre-trained classifier and sampled images in [47] for the evaluation of the entire test set. The comparison results with the state-of-the-art methods are shown in Table 2, where UNIT get the highest score. Kruskal-Wallis test is then used to test the scores of all methods ($P < 0.05$), and Nemenyi test is used for pairwise comparison where the difference between UNIT, CycleGAN and ours is not significant ($P > 0.05$). Although UNIT and CycleGAN also get high scores, they do not meet our requirements in terms of identity consistency.

4.3.3. Comparison with Perceptual Loss.

Perceptual loss, which is adopted by Fast Style and CariStyGAN, is a successful loss function in the task of image transfer. By inputting the original

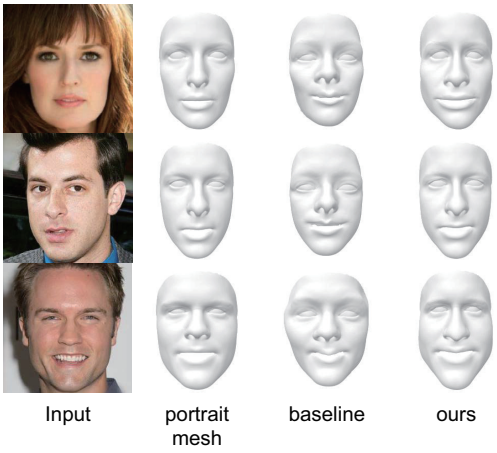

Figure 10: The comparison between baseline method and ours. Parameter α is set to 1. Input images are selected from CelebA dataset.

image and the translated result into the pretrained VGG [48] and comparing the differences of two output features in the last convolution layer of VGG, perceptual loss can explicitly keep the content consistency of these two images. To compare the role of identity loss and perceptual loss, we replace the identity loss of CariFaceGAN with perceptual loss and leave other configurations unchanged. The experimental results are shown in Fig. 7 which shows that identity loss can better maintain the content consistency in the task of caricature generation.

4.4. Experiments on 3D Caricature Generation

Fig. 8 shows the final 3D caricature, which is built by mapping the caricatures texture to the caricature mesh.

4.4.1. Results with Control.

Our proposed geometry transfer can be controlled by user. To tweak the geometric exaggeration extent, we introduce a parameter α to adjust the offset of landmarks. As in the previous representation, we use c to represent the landmarks of the input image, and p the target landmarks calculated by c, $\Delta l = c - p$ the offset of landmarks. The observed values finally used in 3D Universal Kriging can be obtained by $\alpha\Delta l$. Setting α to 0.5, 1 and 2 respectively, and the results are shown in Fig. 9.

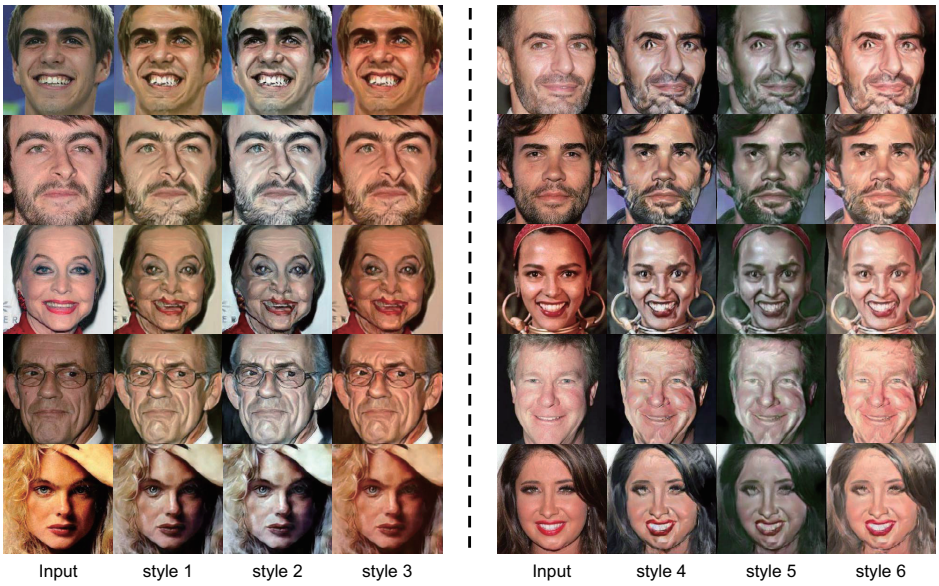

Figure 11: Our system allows the user to control the appearance style. Results are generated with random style codes and input images are from CelebA dataset.

4.4.2. User Study.

Since caricature is a kind of artistic work, its quality should be judged by the users. Therefore, we conduct a user study with 40 participants (including 18 artists and art students and 22 volunteers without art background; 17 males and 23 females; aged from 19 to 35) to compare our method with a baseline method. A straight-forward method to build 3D caricature mesh is to use Laplacian deformation algorithm[49], whose results are shown in Fig. 10. Baseline results are built by deforming the portrait mesh with the offset of paired portrait and caricature landmarks. We randomly pick 10 paired results of baseline and our method, and every participant is requested to give scores to the following two questions: 1) the degree of similarity between the 3D caricature model and the 2D caricature geometric structure in the data set; 2) the visual effect of the 3D caricature model. We use a score ranging from 0 to 5 to represent the criteria (i.e. worst to the best), and the whole results are shown in Table 3. In order to verify whether the difference between these two methods is statistically significant, we use Mann-Whitney test to test the scores of these two questions. The result show that our approach is superior to baseline both in visual effects ($P<0.001$) and

Table 3: Quantitative results of the user study of 3D caricature built by baseline and our method.

Methods	Q1	Q2
Baseline	3.37 ± 0.9	2.91 ± 0.81
Ours	3.75 ± 1.13	3.48 ± 0.97

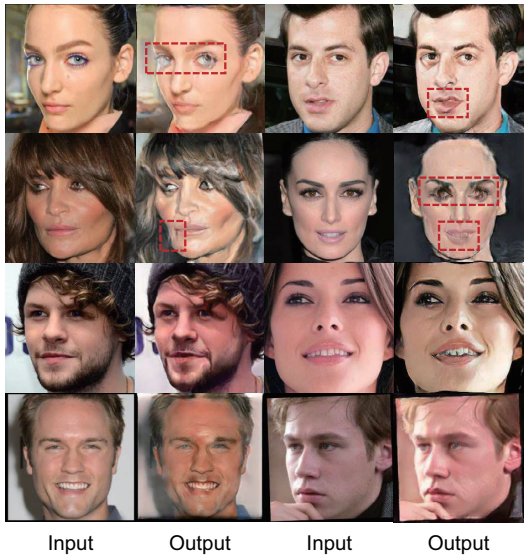

Figure 12: Failure cases of CariFaceGAN. Outputs in the first two rows show a certain degree of artefacts in the mouth and eyes, while the change of the styles in the last two rows is not significant enough. Input images are selected from CelebA dataset.

structural consistency ($P<0.001$).

5. Conclusions

In this paper, we propose a method to build 3D caricature which consists of a GAN-based cross-domain transfer network (CariFaceGAN) and a geometry transfer pipeline. The experimental results show that our CariFaceGAN outperforms previous state-of-the-art methods in both maintaining identity consistency and transferring the style. Moreover, the 3D caricatures constructed by our approach show satisfactory visual effects.

There still exists some room for improvement. Firstly, the low resolution of generated images from CariFaceGAN (i.e. 256 x 256) can not provide sufficient details when mapping the texture to the deformed mesh, thereby

leading to some blurred areas in the final 3D caricatures. Secondly, since translating portrait to caricature is a very challenging task, our model also fails in some cases as shown in Fig. 12. There are mainly two types of failure cases, one is the occurrence of artefacts on the eyes or mouth, and the other is the unexpected variance during the conversion. Thirdly, since labelled caricature data is scarce in the offline stage of the geometry transfer pipeline, the deformation result is relatively monotonous. These will be considered in future work.

Acknowledgements

We intend to thank all participants in our user study. This work is supported by the Project of Transformation for Ocean science and Technology of Xiamen (No.18CZB033HJ11) and National Natural Science Foundation of China under grants of No.41971424 and No.61902330.

References

[1] J. Xie, Y. Chen, J. Liu, C. Miao, X. Gao, Interactive 3d caricature generation based on double sampling, in: Proceedings of the 17th ACM international conference on Multimedia, 2009, pp. 745–748.

[2] T. Sucontphunt, A practical approach for identity-embodied 3d artistic face modeling, International Journal of Computer Games Technology 2014 (2014).

[3] X. Han, K. Hou, D. Du, Y. Qiu, S. Cui, K. Zhou, Y. Yu, Caricatureshop: Personalized and photorealistic caricature sketching, IEEE transactions on visualization and computer graphics (2018).

[4] P. Li, Y. Chen, J. Liu, G. Fu, 3d caricature generation by manifold learning, in: 2008 IEEE International Conference on Multimedia and Expo, IEEE, 2008, pp. 941–944.

[5] J. Liu, Y. Chen, C. Miao, J. Xie, C. X. Ling, X. Gao, W. Gao, Semi-supervised learning in reconstructed manifold space for 3d caricature generation, in: Computer Graphics Forum, Vol. 28, Wiley Online Library, 2009, pp. 2104–2116.

[6] J. Zhou, X. Tong, Z. Liu, B. Guo, 3d cartoon face generation by local deformation mapping, The Visual Computer 32 (6-8) (2016) 717–727.

[7] Q. Wu, J. Zhang, Y.-K. Lai, J. Zheng, J. Cai, Alive caricature from 2d to 3d, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7336–7345.

[8] Y. Guo, L. Jiang, L. Cai, J. Zhang, 3d magic mirror: Automatic video to 3d caricature translation, arXiv preprint arXiv:1906.00544 (2019).

[9] L. A. Gatys, A. S. Ecker, M. Bethge, Image style transfer using convolutional neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2414–2423.

[10] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European conference on computer vision, Springer, 2016, pp. 694–711.

[11] D. Chen, L. Yuan, J. Liao, N. Yu, G. Hua, Stylebank: An explicit representation for neural image style transfer, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1897–1906.

[12] J. Liao, Y. Yao, L. Yuan, G. Hua, S. B. Kang, Visual attribute transfer through deep image analogy, ACM Transactions on Graphics (TOG) 36 (4) (2017) 1–15.

[13] D. Chen, J. Liao, L. Yuan, N. Yu, G. Hua, Coherent online video style transfer, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1105–1114.

[14] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[15] Z. Yi, H. Zhang, P. Tan, M. Gong, Dualgan: Unsupervised dual learning for image-to-image translation, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2849–2857.

[16] M.-Y. Liu, T. Breuel, J. Kautz, Unsupervised image-to-image translation networks, in: Advances in neural information processing systems, 2017, pp. 700–708.

[17] X. Huang, M.-Y. Liu, S. Belongie, J. Kautz, Multimodal unsupervised image-to-image translation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 172–189.

[18] K. Cao, J. Liao, L. Yuan, Carigans: Unpaired photo-to-caricature translation, ACM Transactions on Graphics 37 (6) (2018) 244.1–244.14.

[19] Z. Zheng, C. Wang, Z. Yu, N. Wang, H. Zheng, B. Zheng, Unpaired photo-to-caricature translation on faces in the wild, Neurocomputing 355 (2019) 71–81.

[20] H. Hou, J. Huo, J. Wu, Y.-K. Lai, Y. Gao, Mw-gan: Multi-warping gan for caricature generation with multi-style geometric exaggeration, arXiv preprint arXiv:2001.01870 (2020).

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[22] V. Blanz, T. Vetter, A morphable model for the synthesis of 3d faces, in: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, 1999, pp. 187–194.

[23] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, T. Vetter, A 3d face model for pose and illumination invariant face recognition, in: 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, Ieee, 2009, pp. 296–301.

[24] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, X. Tong, Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.

[25] Z. Ye, R. Yi, M. Yu, J. Zhang, Y.-K. Lai, Y.-j. Liu, 3d-carigan: An end-to-end solution to 3d caricature generation from face photos, arXiv preprint arXiv:2003.06841 (2020).

[26] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, science 290 (5500) (2000) 2323–2326.

[27] M. A. Oliver, R. Webster, Kriging: a method of interpolation for geographical information systems, International Journal of Geographical Information System 4 (3) (1990) 313–332.

[28] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[29] E. Richardson, M. Sela, R. Kimmel, 3d face reconstruction by learning from synthetic data, in: 2016 Fourth International Conference on 3D Vision (3DV), IEEE, 2016, pp. 460–469.

[30] A. Tewari, M. Zollhofer, H. Kim, P. Garrido, F. Bernard, P. Perez, C. Theobalt, Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 1274–1283.

[31] A. Chen, Z. Chen, G. Zhang, K. Mitchell, J. Yu, Photo-realistic facial details synthesis from single image, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9429–9439.

[32] A. Jourabloo, X. Liu, Pose-invariant 3d face alignment, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3694–3702.

[33] A. Jourabloo, X. Liu, Large-pose face alignment via cnn-based dense 3d model fitting, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4188–4196.

[34] F. Liu, D. Zeng, Q. Zhao, X. Liu, Joint face alignment and 3d face reconstruction, in: European Conference on Computer Vision, Springer, 2016, pp. 545–560.

[35] E. Richardson, M. Sela, R. Or-El, R. Kimmel, Learning detailed face reconstruction from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1259–1268.

[36] A. S. Jackson, A. Bulat, V. Argyriou, G. Tzimiropoulos, Large pose 3d face reconstruction from a single image via direct volumetric cnn

regression, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1031–1039.

[37] Y. Feng, F. Wu, X. Shao, Y. Wang, X. Zhou, Joint 3d face reconstruction and dense alignment with position map regression network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 534–551.

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[39] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, W. T. Freeman, Synthesizing normalized faces from facial identity features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3703–3712.

[40] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, W. T. Freeman, Unsupervised training for 3d morphable model regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8377–8386.

[41] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman, Vggface2: A dataset for recognising faces across pose and age, in: International Conference on Automatic Face and Gesture Recognition, 2018.

[42] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: the International Conference on Learning Representations, 2015.

[43] A. Bulat, G. Tzimiropoulos, How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks), in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1021–1030.

[44] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, Active shape models-their training and application, Computer vision and image understanding 61 (1) (1995) 38–59.

[45] P. K. Kitanidis, Introduction to geostatistics: applications in hydrogeology, Cambridge university press, 1997.

[46] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch (2017).

[47] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, Advances in neural information processing systems 29 (2016) 2234–2242.

[48] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2015).

[49] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, H.-P. Seidel, Laplacian surface editing, in: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, 2004, pp. 175–184.