The Visual Computer (2019) 35:783–795
https://doi.org/10.1007/s00371-019-01695-6

ORIGINAL ARTICLE

Expressive facial style transfer for personalized memes mimic

Yanlong Tang[1] · Xiaoguang Han[2] · Yue Li[3] · Liqian Ma[4] · Ruofeng Tong[1]

Published online: 11 May 2019

Abstract

Meme, usually represented by an image of exaggerated expressive face captioned with short text, are increasingly produced and used online to express people's strong or subtle emotions. Meanwhile, meme mimic apps continuously appear, such as the meme filming feature in WeChat App that allow users to imitate meme expressions. Motivated by such scenarios, we focus on transferring exaggerated or unique expressions which is rarely noticed by previous works. We present a technique—"expressive style transfer"—which allows users to faithfully imitate popular memes' unique expression styles both geometrically and textually. To conduct distortion-free transferring of exaggerated geometry, we propose a novel accurate feature curve-based face reconstruction algorithm for 3D-aware image warping. Furthermore, we propose an identity preserving blending model, based on a deep neural network, to enhance facial expressive textural details. We demonstrate the effectiveness of our method on a collection of Internet memes.

Keywords Expressive style transfer · Meme generation · Curve-based 3D face reconstruction · Neural-style alpha-blending

1 Introduction

Memes are widely used in social media to express complex feelings in funny ways. Figure 1a shows two memes of characters with different expressions. Because each expression is exaggerated and uniquely owned by specific character, it is difficult for another person to replicate it faithfully with their own face, as shown in Fig. 1b.

We propose a novel expressive style transfer framework to transfer a meme—image A (Fig. 1a)—to arbitrary faces for producing personalized memes. The pipeline requires the user to mimic the expression of the target meme to get an approximation—image B (Fig. 1b). After this, our goal is to transfer the expression of A to B while preserving the identity of B. This is done in two steps: First, we deform the geometry of B to approximate A's expression, and then, we blend the result with A to capture the details contained in the texture. After the two steps, a more expressive expression is produced. Note that the final expression does not correspond to an expression that the person can actually have as image features from the meme image are introduced to the final result in the blending step.

For the first step, methods based on 2D image warping usually cause the resulting face to have an unrealistic appearance [44]. Instead, we conduct the geometric transfer in 3D space (as in [41]) by taking a textured 3D mesh recovered from B and deforming it to fit A. Since most of the target memes feature highly exaggerated expressions, current landmark-based 3D morphable model (3DMM) fitting algorithms [8,9] fail to capture the correct geometry of memes, causing mismatching along some feature curves such as mouth and eye boundaries. These approaches also result in artifacts in the final blending. To tackle this issue, a curve-aware 3D face reconstruction framework can locate feature curves in the image and then move vertices on the 3D model to match the extracted curves, conducted with a

✉ Xiaoguang Han
hanxiaoguang@cuhk.edu.cn

Yanlong Tang
yanlongtang@gmail.com

Yue Li
yueli.cg@gmail.com

Liqian Ma
maliqian@kuaishou.com

Ruofeng Tong
trf@zju.edu.cn

[1] Zhejiang University, Hangzhou, China

[2] Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong (Shenzhen), Shenzhen, China

[3] University of Pennsylvania, Philadelphia, USA

[4] Beijing Kuaishou Technology Ltd., Beijing, China

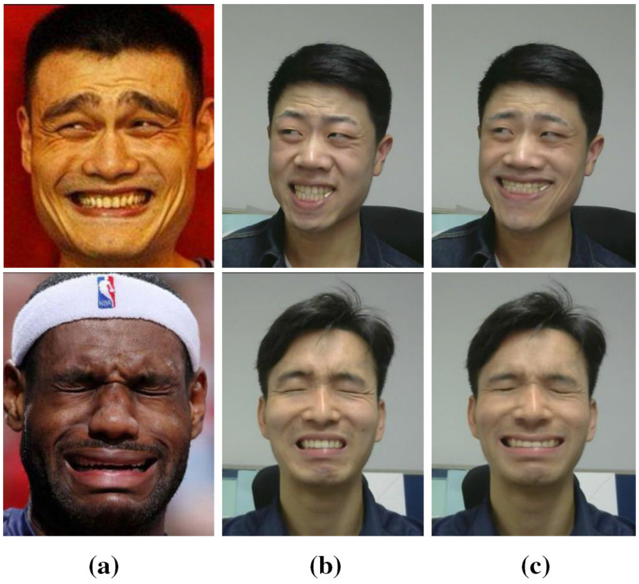


Fig. 1 Expressive facial style transfer. Users can create personalized memes (**c**) by imitating popular ones (**a**) with their own faces (**b**)

coarse-to-fine optimization mechanism. Visual comparisons show our approach outputs more accurate results than existing ones.

The second step of the process aims to transfer the expression-related texture details from A to C, generating the final result. Taking the first row in Fig. 1 as an example, details such as the wrinkles on the cheeks are essential to transfer if the meme is to be mimicked convincingly. On the other hand, texture over-blending will erode the user's identity-related features, making the result seem less personalized. To strike a balance, we formulate the problem as an optimization of a pixel-wise alpha map. This is solved by trading off between an identity preserving loss and an expression transferring loss within a neural style transfer network architecture [16]. Another coarse-to-fine optimization strategy is used at this stage. Visual comparisons and user case studies indicate that every step of our framework is necessary to achieving final result.

Our contributions can be summarized as follows:

– We propose the first exaggerated expression transfer approach for customized meme generation.
– We propose a novel 3D face reconstruction algorithm based on feature curve matching to produce accurate fitting.
– We propose a neural style transfer network to optimize alpha blending that can transfer expression texture details while preserving identity.

2 Related works

2.1 3D face reconstruction from images

Existing methods for recovering 3D facial geometry fall into two major categories: reconstruction via shape from shading (SfS) and fitting via a parametric model (3DMM). SfS methods estimate the normal of each pixel based on the image's illumination, and they then recover the depth information according to the inferred normal [23,37,38]. However, these approaches tend to fail especially for in-the-wild images (e.g., Internet memes), since it is a non-trivial process to decompose the shading information.

Most parametric fitting methods [5,8,9,11,44] use sparse landmarks as constraints for reconstruction; the works of [4,15,24,31,36,41] introduce pixel information as additional constraints; [2,36] take edge constraints into account; and [6,19,27,28] show that medium-level correction could refine a coarse-level fitting result. Recently, learning-based methods [7,45] have been proposed to speed up 3D face reconstruction. These methods usually use data produced by optimization-based methods. All the above methods approximate facial feature regions (such as the mouth, nose and eyes) as sparse points, which make the reconstructed 3D face fall short of matching the expected geometry, especially when the expression is exaggerated. To deal with this facial region mismatch problem, we propose a feature curve-based 3D facial reconstruction method, which achieves more accurate results.

2.2 Facial expression and texture transfer

Existing approaches that can perform facial expression transferring mainly focus on facial shape or movement without blending textures. This does not enable them to faithfully reproduce texture changes caused by expressions. There are mainly three ways of performing expression transfer: 2D warp, 3D warp and image synthesis with generative adversarial networks (GANs).

2D warping methods [1,14,17] usually translate the movements of the detected feature points for guiding a warping on the image domain. These tend to result in artifacts especially when the expression's changing is very large. Thus, 3D warping approaches [41,44] are presented, which firstly recover the 3D textured model, resorting to 3DMM, and then re-render it after a geometry-domain deformation. Limited by the representational ability of 3DMM, these methods are difficult to modeling exaggerated face's geometry, making them not applicable for our settings. The GAN-based methods [10,33,34,43] usually require a large well-collected dataset for training a deep ConvNets, making them difficult to use for our goal.

In terms of texture transfer, existing methods [12,13,16,26,29,30,42] mainly focus on low-level texture transfer, which usually destroys identity information heavily. Among them, works [12,16,26] are more suitable for texture transfer of non-photorealistic images as they lack spatial correspondence control, and [29] improves style transfer stability by adding semantic correspondence in feature maps, but identity is still not well controlled and preserved when applied on facial images. Recent work [25] can change texture by blending two face images, but this method changes the identity and image tone greatly. Compared with all these methods, our expression style transfer algorithm better preserves spatial correspondence and identity.

3 Algorithm

3.1 Overview

Different from previous works that focus on transferring common expressions to a neutral expression face, our goal is to transfer exaggerated or unique expressions that widely seen in popular memes, so that the results can faithfully restore memes' funny or expressive emotion. In Fig. 2, we illustrate the overall pipeline.

The pipeline takes two images as input: One is the source meme image I_s which is extremely expressive, and the other is a user image I_t. We require users to input a raw mimic expression instead of neutral expression, because currently geometric expression transfer works well with the assumption that source expression does not differ too much from target expression. Otherwise, it will lead to obvious distortion artifacts.

With the two input images, we first propose an improved sophisticated geometric transfer approach that deforms the user image I_t to an exaggerated expression I'_t. The deformation achieves delicate expression shape correction that better match the expression of original meme I_s (such as trademark eyebrows and unique mouth shape). Such delicate geometric transfer mainly benefits from the proposed feature curve-based 3D face reconstruction that can model 3D source face M_s and target face M_t more accurately. With accurate 3D faces, dense 3D warp [40] can generate artifact-free exaggerated 3D expression M'_t. Then M'_t is rendered to I'_s and I'_t

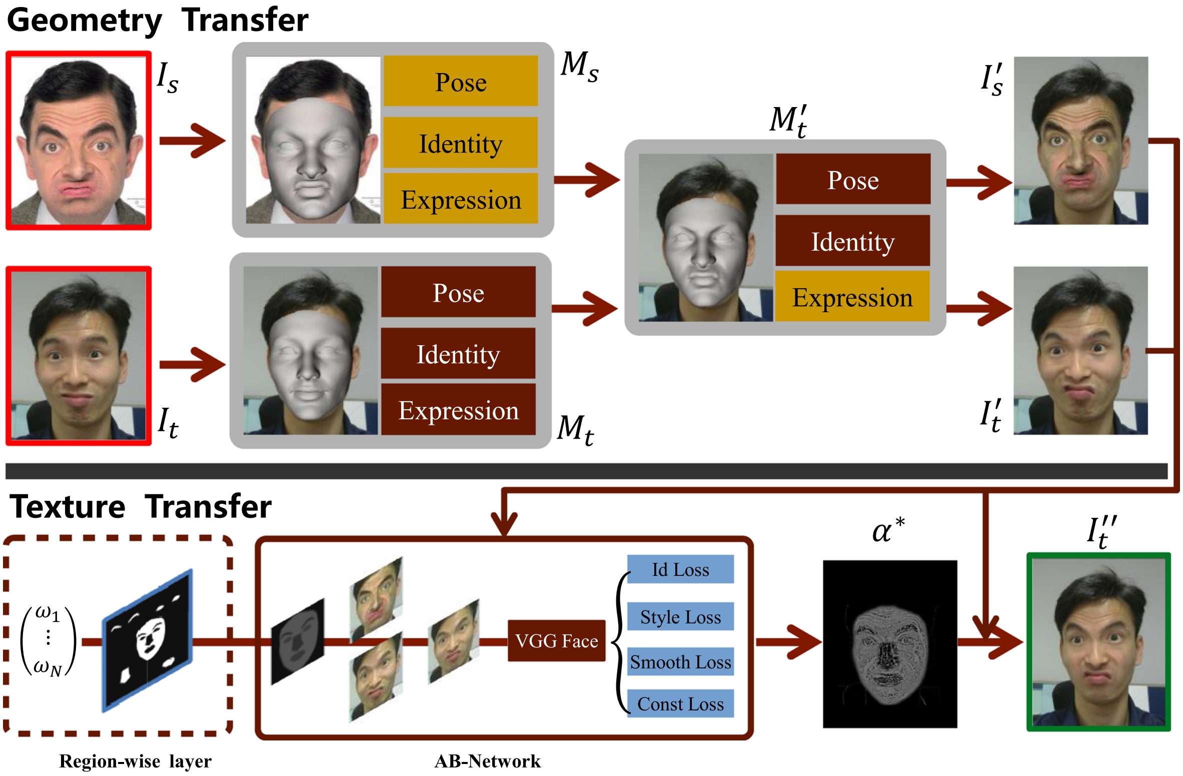


Fig. 2 Overall pipeline of our algorithm. Given a reference meme image I_s in exaggerated expression, the user firstly roughly mimics its expression and gives rise to an image I_t. Then accurate facial geometries are recovered from these two images, respectively, resulting in 3D faces M_s and M_t. In the step of geometric transfer, the expression-related geometry of M_s is extracted and transferred to M_t, generating an expression-corrected 3D face M'_t. Two new images I'_s and I'_t are then obtained by texturing I_s and I_t on M'_t individually. The step of texture transfer is then applied to transfer the texture details of I'_s to I'_t, by preserving the facial identity. Our final output is I''_t

using meme's and user's texture, respectively. I'_s is rendered with color-corrected texture.

We then propose textural enhancement to further improve the expressiveness after the geometric correction step. However, naive fix threshold-based alpha blending cannot well balance the identity persevering and texture enhancement for all cases. We thus optimize a pixel-wise alpha map α^* to blend I'_t and I'_s in a VGG-Face [32] face recognition neural network, so that the final result I''_t can sufficiently transfer meme's texture detail while preserving the user's identity.

3.2 Geometric expression transfer

3.2.1 3D face reconstruction

Landmark-based 3DMM fitting We use a parametric model [45] to reconstruct the target 3D face, which is expressed as:

$$M = M_{\text{mean}} + B_{\text{id}} * \alpha + B_{\text{exp}} * \beta \tag{1}$$

where M is 3D face and M_{mean} is mean face. B_{id} and B_{exp} are shape and expression basis, while α and β are shape and expression parameters. Weekly perspective camera model is proposed to transform 3D face to 2D:

$$V(\mathcal{T}) = f * \text{Pr} * R * (M_{\text{mean}} + B_{\text{id}} * \alpha + B_{\text{exp}} * \beta) + t_{2d} \tag{2}$$

where f is the scaling parameter, Pr is an orthogonal projection matrix, R is a rotation matrix (represented by $pitch$, yaw and $roll$) and t_{id} is 2D translation. Model parameters can be expressed as $\mathcal{T} = [f, pitch, yaw, roll, t_{\text{id}}, \alpha, \beta]^T$.

Usually, image-based 3DMM fitting is based on sparse landmarks and the landmark-based energy is:

$$E_{\text{lan}}(\mathcal{T}) = \sum_{f_k \in \mathcal{F}} \omega_{\text{conf},k} \| f_k - \Pi(\Phi(v_k)) \|_2 \tag{3}$$

where $\Phi = \Phi(R, t)$ is the modelview matrix and $\Pi = \Pi(f, \text{Pr})$ is the projection matrix. $\omega_{\text{conf},k}$ is the confidence of landmark f_k. We use Face++ commercial landmark detector to robustly detect 83 landmarks. In the fitting, camera parameters and shape parameters are updated in an alternative manner. When updating camera parameters, landmarks of rigid regions, such as face contours, nose contours and eye corners, are assigned with larger confidence value 2.0, while the rest landmarks are assigned with confidence value 1.0. When updating shape parameters, all landmarks are assigned with confidence value 1.0.

In addition to landmark energy, a face prior constraint [41] is also introduced to prevent generating an unrealistic face that is outside the data distribution. The face prior energy is expressed as:

$$E_{\text{reg}}(\mathcal{T}) = \sum_{i=1}^{N_\alpha} \left(\frac{\alpha_i}{\sigma_{\text{id},i}} \right)^2 + \sum_{i=1}^{N_\beta} \left(\frac{\beta_i}{\sigma_{\text{exp},i}} \right)^2 \tag{4}$$

Feature curve fitting Given the landmark and face prior constraints, a coarse 3D model is estimated. The landmark-based fitting is, however, not accurate enough to support the texture blending procedure. This is because sparse landmarks cannot faithfully represent the facial feature shapes in exaggerated expressions. Therefore, we propose feature curve-based 3D face reconstruction, which can achieve better contour matching. We first extract feature curves based on the landmarks to better represent the 2D facial shape. Then, the fitting is conducted in an iterative way, by updating 3D feature curves and the 2D/3D curve correspondence for each iteration.

2D feature curve extraction We firstly define 14 semantic facial feature curves, by connecting the landmarks with lines. These curves are then optimized to match local edges with the snake algorithm [22], as shown in Fig. 8c. For each semantic curve, the raw initial positions are multiple short straight lines by connecting adjacent landmarks. Then the curve optimization is conducted in a local-to-global manner in two steps. According to [22], α and β in equation (2) control the shape length and smoothness, while ω_{line} and ω_{edge} in equation (3) control attraction to brightness and edges. Adjusting these energy weights can create a wide range of snake behavior. In the first step, we locally optimize each piece of short straight line by fixing the two adjacent landmarks. And the short straight line will be bended and locked onto local edges by setting large edge energy weight $\omega_{\text{edge}} = 12.0$, and other parameters are set as $\alpha = 0.1$, $\beta = 0.1$, $\omega_{\text{line}} = -5.0$. Then in the second step, the curve will be further globally smoothed by fixing two curve end points with larger smoothness energy weight $\beta = 0.5$, and other parameters are set as $\alpha = 0.1$, $\omega_{\text{line}} = -5.0$, $\omega_{\text{edge}} = 8.0$. With the curve optimization operation, it can effectively eliminate the reconstruction errors caused by inaccurate curves approximated by straight lines.

3D feature curve updating Thanks to the fixed topology of 3D face models, we can pre-define some 3D feature curves, such as the boundaries of the mouth, eyes and nose, by connecting a sequence of vertices. However, it is difficult to fix the vertex sequence for occluded edges as they will change when the model is changing. Regarding this, the method of [36] is used to calculate the edges.

3D/2D curve matching In each iteration, the updated 3D curves are firstly projected to image plane and are then matched to the extracted feature curves using the iterative closest point (ICP) method [3], resulting in dense correspondences.

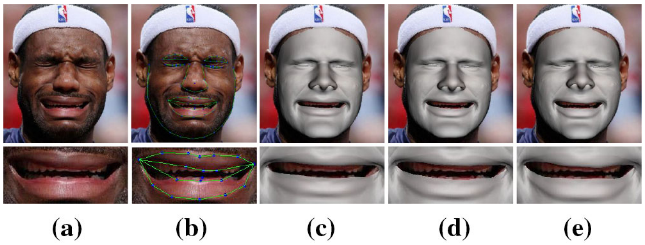


Fig. 3 Input an image (**a**), 83 landmarks using Face++ detector are shown in **b**, the 3D fitting result only using landmarks is shown in **c**, the proposed feature-based fitting result is shown in **d**, the final result after curve-based deformation is shown in **e**

Given the correspondences, the curve energy can be formulated as:

$$E_{\text{cur}}(\mathcal{T}) = \sum_{k=1}^{N_k} \sum_{c_i \in \mathcal{C}_k, v_j \in \mathcal{V}_k} \|c_i - \Pi(\Phi(v_j))\|_2 \tag{5}$$

where N_k is the number of semantic curves. $\mathcal{C}_k/\mathcal{V}_k$ is the kth 2D curve/3D curve in 2D image/3D mesh. By applying the above 2D/3D feature curve correspondence updating, for each point v_j in $\mathcal{V}_k$, we find a corresponding point c_i in $\mathcal{C}_k$. In curve-based fitting, landmark and prior term constraints are also added in the total energy with smaller weights. And weights for landmark term, prior term and curve term are 0.005, 2.0 and 15.0, respectively. Afterward, the reconstructed face better matches the facial feature shapes. See Fig. 3d.

Curve-based deformation Since an exaggerated facial expression may exceed the representational power of 3D face morphable model, as a post-processing, we propose to deform the reconstructed face model to match the target face's feature curves. The Laplacian deformation method [39] is used to reach our goal, outputting a more accurate result. Figure 3e shows the necessity of our curve-based deformation for face reconstruction.

3.2.2 Geometric warp

We denote the reconstructed face model from I_s as M_s and the reconstructed model from I_t as M_t. Then, the deformation transfer method [40] is applied to transfer the expression of M_s to M_t, resulting in M_t'. Afterward, we map the user's face texture I_t on M_t' and re-render it to get a new image $I_\text{t}\prime$. On the other hand, the meme's face I_s is also mapped to M_t' for re-rendering, which produces another image $I_\text{s}\prime$. Before conducting texture mapping, the color correction method in [35] is applied to make the meme face's color consistent with the user's face. This lets us avoid tone changing in the following texture transfer step.

3.3 Texture expression transfer

3.3.1 Id preserving blending

Problem formulation The goal of our texture expression transfer is to solve a pixel-wise alpha map α^* to blend I_t' and I_s':

$$I_\text{t}''(i,j) = \alpha_{ij}^* I_\text{s}'(i,j) + (1-\alpha_{ij}^*) I_\text{t}'(i,j) \tag{6}$$

In order to obtain more textured expression details of the meme face I_s while maintaining the identity of the user's face I_t, we borrow the idea of neural style transfer [16] to optimize the α-map with a deep neural network.

Network architecture Different from [16], our optimization objective is solving an alpha map instead of the final output image. This setting better preserves spatial semantic information that traditional neural style transfer [16] lacks, as alpha blending can implicitly build and preserve spatial semantic dense correspondence. For example, traditional method [16] may transfer wrinkles to wrong regions, such as mouth contour regions in Fig. 12e, while our method will constraint wrinkles in the right regions; see Fig. 12h. We propose an alpha blending network (AB-Network) based on VGG-Face [32]. The AB-Network is compose of two modules: alpha blending module and VGG-Face module, as shown in Fig. 2 highlighted as a solid red box. An input alpha map is firstly passed to the alpha blending module, which produces a blended image. This is then sent to VGG-Face module. (We take the pre-trained model as provided.) In this network, the alpha map is optimized by back-propagating the gradients with respect to the loss functions, with the parameters of VGG-Face module fixed.

Loss definition The loss function used in this work is a weighted sum of four different loss functions, and we will give the details of the each individual loss. The loss functions used in this work are defined as below.

Identity loss Identity loss is used to ensure that the face in the output image I_t'' has the same overall identity as the user in the image I_t. We use high-level layers (conv8) features F_{ij}^I to form the identity loss:

$$\mathcal{L}_{\text{id}}(\alpha) = \sum_{ij} \left(\frac{F_{ij}^I(\alpha)}{\|F^I(\alpha)\|_2} - \frac{F_{ij}^I(\alpha^0)}{\|F^I(\alpha^0)\|_2} \right)^2 \tag{7}$$

where α^0 means the alpha map with zero value and $F_{ij}^I(\alpha^0)$ stands for the feature maps of the processed user image I_t', while $F_{ij}^I(\alpha)$ means the feature maps of I_t''.

Style loss Style loss is used to blend texture areas such as wrinkles in a meme image. However, based on our experiments, the classical style loss proposed by the work [16] often produces uncontrollable alpha map and causes artifacts to the blended image. This is because the Gram-based style representation captures global statistics and does not preserve local structures [20].

Hence, we design a new loss formula as below, which better extracts wrinkle-level style with using the feature maps of low-level layers (conv4) in the VGG-Face module.

$$\mathcal{L}_{\text{style}}(\alpha) = \sum_{ij}(F^S_{ij}(\alpha) - F^S_{ij}(\alpha^1))^2 \tag{8}$$

where α^1 means the map with all pixels own the value 1; thus, the blended image is the meme image and $F^S_{ij}(\alpha^1)$ is the feature map of the meme image.

Const loss The const loss is used as a regulation term to globally ensure that the optimized result does not shift too much away from the initial alpha map input:

$$\mathcal{L}_{\text{const}}(\alpha) = \sum_{ij}(\alpha_{ij} - \alpha^{\text{init}}_{ij})^2 \tag{9}$$

where $\alpha^{\text{init}}_{ij}$ is the initial alpha map input.

TV smooth loss Total variation loss (TV loss) [21] is used as another regulation term to guarantee the local smoothness of the optimized alpha map.

$$\mathcal{L}_{\text{smooth}}(\alpha) = \sum_{ij}(\alpha_{i+1,j} - \alpha_{i,j})^2 + (\alpha_{i,j+1} - \alpha_{i,j})^2 \tag{10}$$

Overall loss The final overall loss is a weighted sum of the above four losses:

$$\mathcal{L}_{\text{total}} = \omega_1\mathcal{L}_{\text{id}} + \omega_2\mathcal{L}_{\text{style}} + \omega_3\mathcal{L}_{\text{const}} + \omega_4\mathcal{L}_{\text{smooth}} \tag{11}$$

3.3.2 Two-stage optimization

Optimizing the pixel-wise alpha map straightforwardly will result in locally saturated pixels, which greatly changes the identity and leads to artifacts. To tackle this problem, we propose a coarse-to-fine optimization strategy. We first solve a coarse region-wise alpha map to guarantee global identity preserving (Fig. 4b). Then we solve the optimal pixel-wise alpha map by taking the region-wise alpha map as an initial value, where regularizers (Fig. 4d) are exploited to ensure as much facial details as possible can be transferred while making artifacts reduced.

Region-wise optimization We firstly introduce our region-wise alpha map optimization. To do so, an additional region-wise layer (dotted red box in Fig. 2) is added into the original AB-Network (solid red box in Fig. 2). In this region-wise layer, alpha map α is produced by combining discrete alpha values $\{\omega_i\}$ of five regions $\{\alpha_i\}$ using the formula $\alpha = \sum_{i=1}^5 \omega_i\alpha_i$. We create five region masks (nose, mouth, eyes, eyebrows and skin) based on the landmark semantic information, where α_i corresponds to the ith region mask. Thus, the optimization objective turns to be solving the optimal values of only five variables.

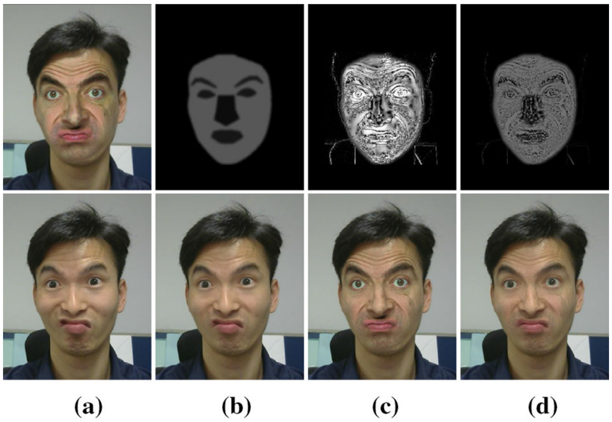


Fig. 4 Results using different alpha map optimization strategies. **a** Two images obtained after geometric transfer step. **b** Region-wise optimization only, which leads to insufficient transferring of texture details. **c** Region-wise and pixel-wise optimization without regularizing. It causes locally blending artifacts. **d** The proposed method enhances texture details without introducing too much noise and identity distortions

Pixel-wise optimization After the region-wise alpha map optimization, we get coarse identity preserving result where some textured details are still not transferred (Fig. 4b). We thus apply pixel-wise optimization to make a further enhancement for local texture transferring. During the optimization procedure, we only use the AB-Network without the region-wise layer. However, without constraint of each pixel's alpha value, the result pixel-wise alpha map may be locally excessive, which can damage the identity and involve too much artifacts (Fig. 4c). To solve this problem, we introduce const loss and TV loss as regulation, which can finally produce texture-enhanced result, while identity is well preserved with reduced artifacts (Fig. 4d).

4 Experiments

4.1 Qualitative results

We perform a series of qualitative evaluations to validate the effectiveness of the proposed personalized meme mimic approach.

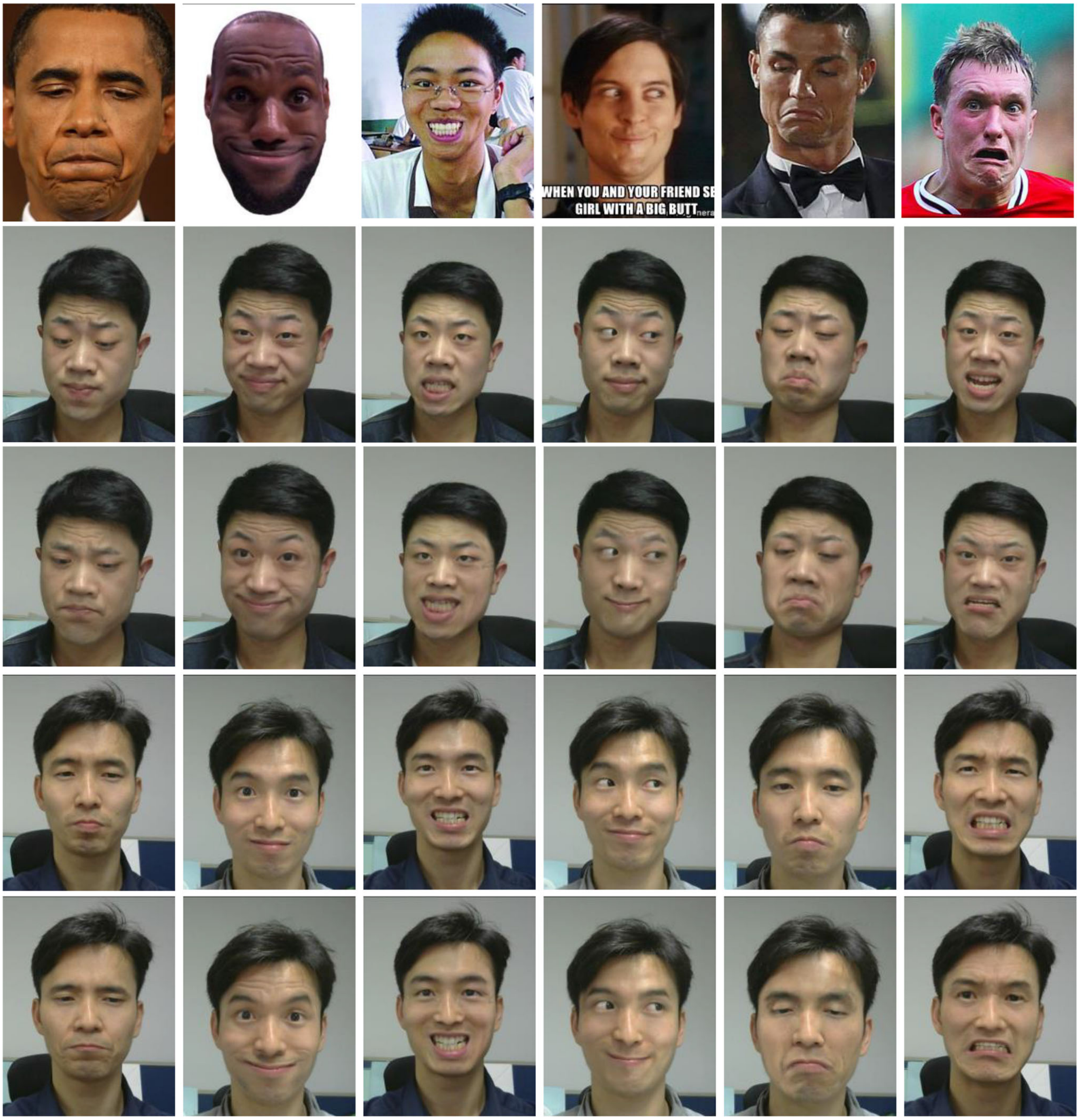


Fig. 5 Result gallery of personalized memes generation. First row: source meme images. Second/fourth row: user's imitation of the memes. Third/fifth row: results

4.1.1 Result gallery exhibition

In order to validate our algorithm, we collected popular meme images from the Internet. The expression styles of these memes are then transferred to various users. The results are shown in Fig. 5. The results show that the proposed expressive facial style transfer approach can faithfully transfer the original meme's exaggerated and unique expression style to various users, making the mimic more emotional and expressive.

4.1.2 Pipeline evaluation user study

We select $N_1 = 5$ typical meme images; then, we ask $N_2 = 3$ users to mimic each meme. For each user/meme pair, we generate $N_3 = 3$ results of various algorithm stages; thus,

Table 1 User study of various stages of the algorithm pipeline

	0	1	2	3	4	Mean
Stage0	3.67	3.54	3.15	3.61	3.14	3.462
Stage1	4.13	3.97	4.03	4.13	4.05	4.062
Stage2	4.25	4.68	4.27	4.35	4.34	4.378

Volunteers score the meme imitations (0–4) quality with score in the range of (1–5) by considering the expressiveness fidelity as well as identity preservation. Statistics shows that both geometric (stage1) and texture transfers (stage2) obviously improve meme imitation, compared with the raw mimic (stage0)

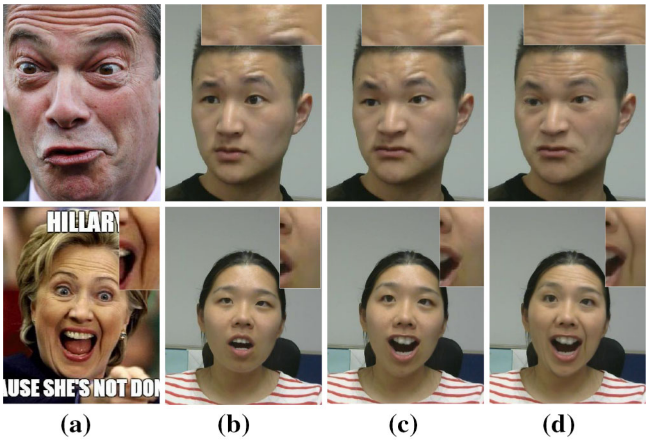


Fig. 6 **a** Input meme images. **b** Input user images. **c** Results of geometric transfer only. **d** Results of applying both geometric and texture transfers

totally $N = N_1 * N_2 * N_3$ images are generated. We then found $N_4 = 10$ volunteers to score each image result to get $N_1 * N_2 * N_3 * N_4$ scores. When scoring, each time a volunteer is shown with the N_3 phase results that produced by $imitator_i$ imitating $meme_j$. Each volunteer is asked to rate the N_3 results by score 1–5 by considering imitation expressiveness, identity preserving degree. The results are shown in Table 1. It shows both geometric and texture transfer steps contribute to good meme mimic. Visual result validation is illustrated in Fig. 6.

4.1.3 Input-invariant test

Although an initial imitation is required as input, we demonstrate that our method can mostly produce stable and reasonable results when the user mimic in various expressions, poses and lighting conditions, as shown in Fig. 7. A failure example (top right of Fig. 7) shows that it produces a weird laughing without obvious nasolabial folds as the input meme and user expressions differ too much. This shows the necessity of using similar expression as input.

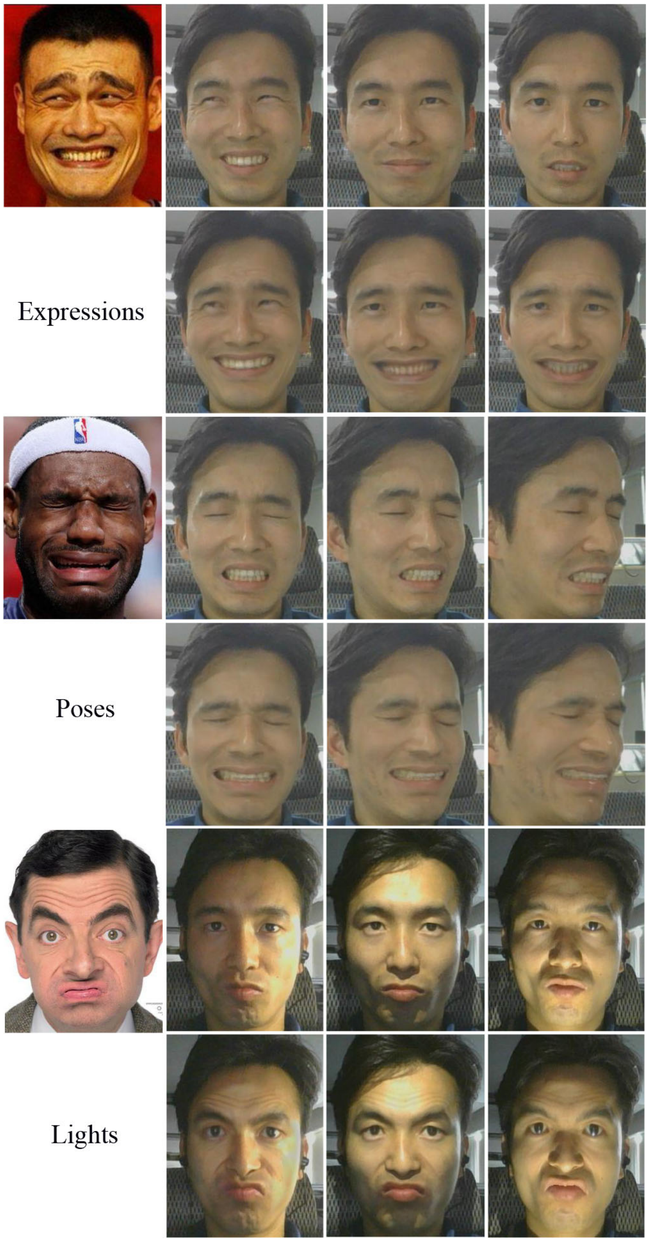


Fig. 7 Results of our algorithm when taking input images with different expressions, poses and lightings. Generally speaking, our algorithm is insensitive to poses and lightings. In the first row, it is noticed that the right top example does not produce a reasonable laughing with obvious nasolabial folds. This shows the necessity of using similar expression as input

4.1.4 3D face reconstruction demonstration

To model exaggerated expressions, good 3D face reconstruction quality is essential. We show the effect of the proposed curve-based 3D face reconstruction method in Fig. 8. The results show that curve-based reconstruction obviously improves the reconstruction quality and more faithfully reconstruct the exaggerated and subtle expressions in 3D.

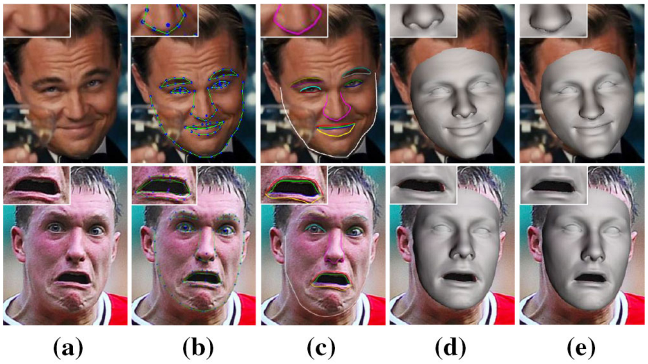


Fig. 8 Demonstration of feature curve-based 3D face reconstruction. **a** Input images. **b** 83 landmarks using Face++ detector. **c** Extracted semantic curves. **d** Reconstruction without curve constraints. **e** Reconstruction with curve constraints

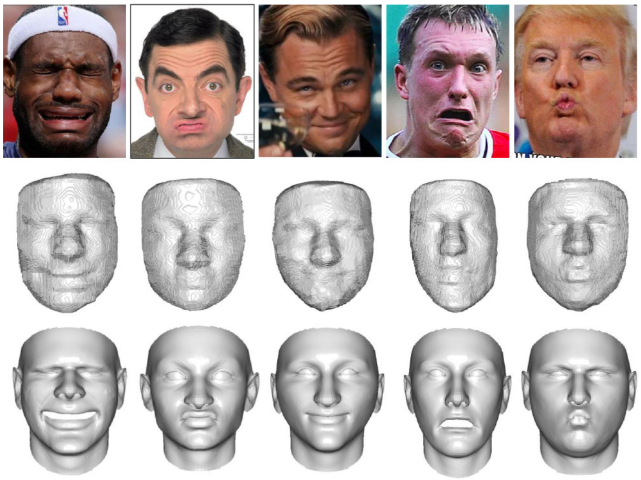


Fig. 9 3D face reconstruction comparisons. With single-image input (first row), our method (third row) obviously outperforms state-of-the-art method [18] (second row), by reconstructing 3D faces with more accurate feature shapes and less noise

We further compare the proposed curve-based reconstruction method with state-of-the-art single-image-based reconstruction method [18], as shown in Fig. 9. The results of [18] are generated using the code released by the authors, and all parameters are set by default. The comparisons in Fig. 9 show that our method can reconstruct higher-quality 3D faces with more accurate feature shapes and less noise, which demonstrates the advantage of our method.

4.1.5 Texture blending losses evaluation

We evaluate the variation of $\omega_1, \omega_2, \omega_3, \omega_4$ of equation (11), as shown in Fig. 10. The proposed baseline result is illustrated in top right, with losses weights $\omega_1 = 0.075, \omega_2 = 0.1$, $\omega_3 = 0.25$, $\omega_4 = 0.15$. The proposed parameters produce reasonable results with enhanced texture details, without introducing too much noise and identity distortions. From second to fourth rows, we double or reduce by half the single weight ω_i in each row and illustrate the corresponding results. The second row shows that when identity weight ω_1 increases, texture detail will be insufficiently transferred, while when identity weight ω_1 decreases, the identity will be less preserved. The third row shows the results of altering style weight ω_2, and it basically shows the opposite effect to the second row, as style loss and identity loss are adversarial. The fourth row shows that when constant loss weight ω_3 increases, the result will be more close to region-wise alpha map result with few textural details, and when the weight decreases, the result will shift away from region-wise result and tend to be overly saturated. The fifth row shows that when TV loss increases, the result alpha map will be over-smoothed and few textual details will be transferred, while when TV loss weight decreases, the result alpha map will contain too much noise and thus leads to heavy artifacts in final result. The evaluation experiment shows the selected

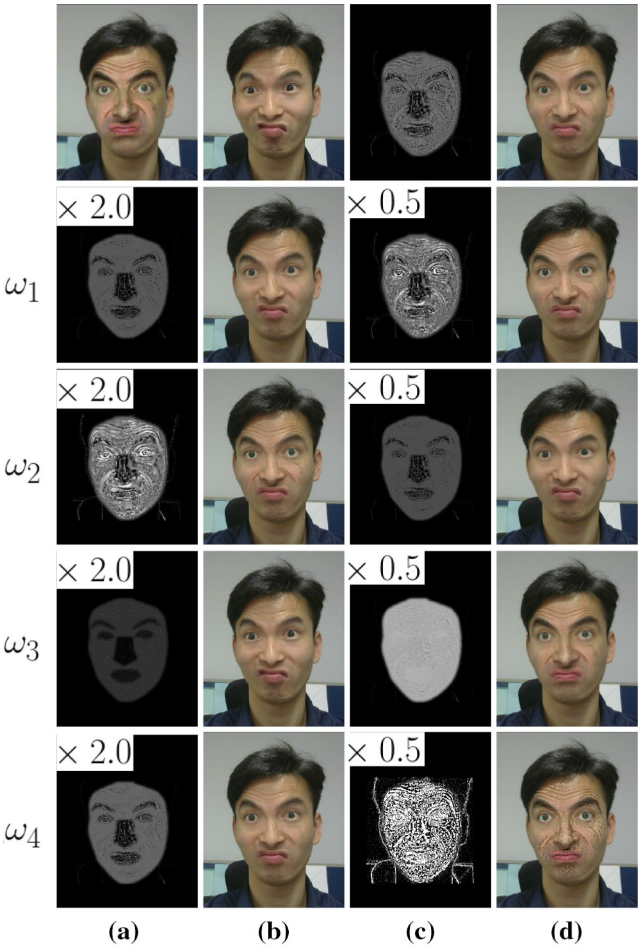


Fig. 10 Effect of variation of $\omega_1, \omega_2, \omega_3$ and ω_4 in equation 11. The first row: **a** and **b** are images before blending. **c** and **d** are blending results with proposed parameters: $\omega_1 = 0.075, \omega_2 = 0.1, \omega_3 = 0.25, \omega_4 = 0.15$. Second to fourth rows: blending results by altering $\omega_1, \omega_2, \omega_3$ and ω_4

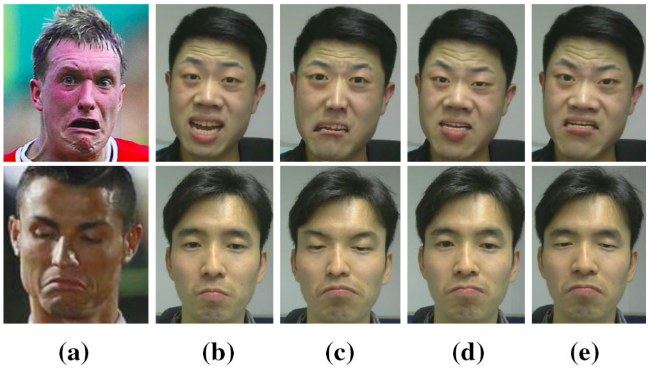


Fig. 11 Geometric transfer comparisons. **a** Input meme image. **b** Input user image. **c** 2D warp [14]. **d** 3D warp [41]. **e** 3D warp (the proposed method). Our method **e** has fewer distortion artifacts than **c** and is more expressive than **d**

baseline weights produce the best balanced result among all the weights combinations and we fix the weights in all our experiments.

4.2 Comparisons against state of the art

We compared our geometric transfer and texture transfer results with the state-of-the-art approaches, respectively.

4.2.1 Geometric transfer

We compare our geometric warp method with state-of-the-art warp methods [14,41], as shown in Fig. 11. As both works required video input, we modify their original implementations to adapt our single-image pair scenario. For work [14], we remove the adjacent frame constraints by setting $\alpha_1 = \alpha_3 = 0.0$, $\alpha_2 = 1.0$ in Eq. (5) and setting $\beta_1 = 0.0$, $\beta_2 = 1.0$ in Eq. (6). For work [41], we reconstruct 3D face in single image without temporally refinement and set the weights as $\omega_{\text{col}} = 1$, $\omega_{\text{lan}} = 10$, $\omega_{\text{reg}} = 2.5 \times 10^{-5}$ following the authors' implementation. Comparisons show the effectiveness of proposed method in geometric warp.

4.2.2 Textural transfer

We also compare our texture transfer approach with three other texture transfer methods, as shown in Fig. 12. To make fair comparisons, input images for all methods are preprocessed by our geometric transfer step. The results shows that our method achieves better semantic texture transfer than [16] and better identity preserving than [25,29].

4.3 Identity preserving effectiveness

We conduct another user study to test the identity preserving effect with various identity loss weights in the texture transfer step. We selected three different users to imitate three different meme images, respectively. Then five identity loss weights are applied to generate various texture blending results. Totally, 45 meme imitations are generated. Then we asked 10 volunteers (five males and five females) to identify the three identities of 45 images and totally get 450 results. Before the identification, we crop and save only face region of each image avoiding the influence of hairstyles and image backgrounds. Based on the statistics, we show the relation between identity loss weight and identification accuracy in Fig. 13. It shows that higher identity loss weight preserves identity better. Without identity loss, people can still identify the images mainly based on the facial geometry with accuracy 68.2%. Based on this user study, the identity loss

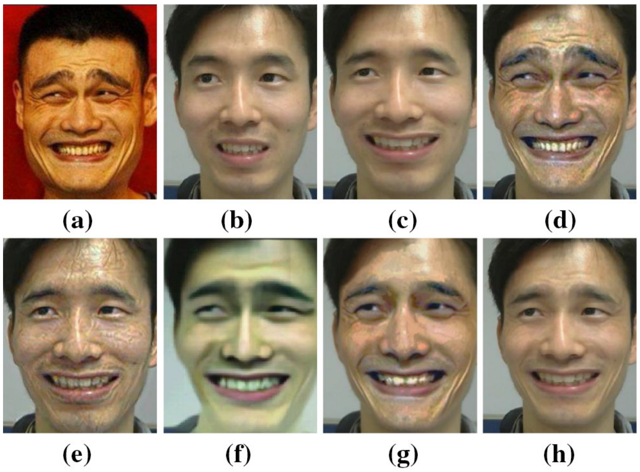


Fig. 12 Comparisons against different texture transfer approaches. **a** Input meme image. **b** The input user image. **c** and **d** are the two textured images after applying geometric transfer. **e** Result of NST [16]. **f** Result of Glow [25]. **g** Result of deep image analogy [29]. **h** Our method, which better transfers the expression textures than **e** while preserving the user's identity more faithfully than **f** or **g**

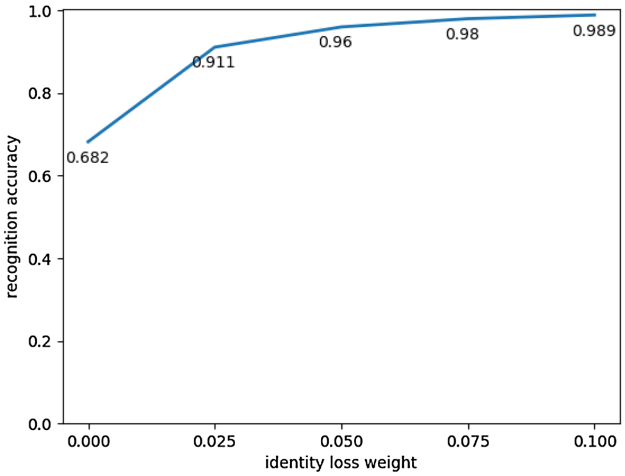


Fig. 13 User study of the effectiveness when increasing the identity loss weight

weight is set as 0.075 in our experiment, which achieves 98% identification accuracy.

5 Conclusions and future works

We present a technique to transfer stylized expressions, which allow users to imitate unique or extreme expressions (trademark eyebrows, glancing and wrinkles) of Internet memes. In contrast to previous methods that focus on transferring common expressions based on multiple images or video, our challenge is to transfer extreme and unique expressions only based on single image pair without neutral expressions. To handle the neutral expression absence problem, we introduce 3D parametric face model, which implicitly encodes a corresponding neutral expression for arbitrary expression. To faithfully transfer the exaggerated geometry, we propose a feature curve-based 3D face reconstruction algorithm for more accurate 3D warping. We further transfer texture details to enhance the expressiveness and preserve the identity. To this end, we optimize pixel-wise alpha blending in face recognition neural network. We demonstrate the necessity and contribution of every step of our approach on a collection of Internet memes. In the future, we plan to introduce GAN-based method to produce better results for arbitrary inputs, especially for neutral expression inputs.

Acknowledgements We thank the anonymous reviewers for the insightful and constructive comments, Matt Boyd-Surka for proofreading this manuscript, Jinghui Zhou, Keli Cheng and Xiaodong Gu for valuable discussions, and Yuan Yao for providing their deep image analogy result [29]. This paper was supported by the National Natural Science Foundation of China (No. 61832016) and the Science and Technology Project of Zhejiang province (No.2018C01080). This work was also funded in part by the Pearl River Talent Recruitment Program Innovative and Entrepreneurial Teams in 2017 under Grant No. 2017ZT07X152 and the Shenzhen Fundamental Research Fund under Grants No. KQTD2015033114415450 and No. ZDSYS201707251409055.

Compliance with ethical standards

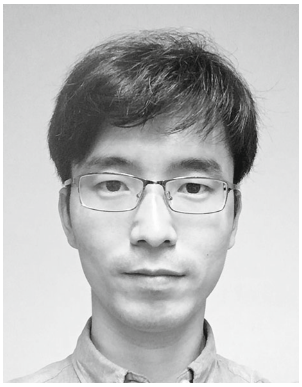
Conflict of interest The authors declare that they have no conflict of interest.

References

1. Averbuch-Elor, H., Cohen-Or, D., Kopf, J., Cohen, M.F.: Bringing portraits to life. ACM Trans. Graph. (TOG) **36**(6), 196 (2017)
2. Bas, A., Smith, W.A., Bolkart, T., Wuhrer, S.: Fitting a 3d morphable model to edges: a comparison between hard and soft correspondences. In: Asian Conference on Computer Vision, pp. 377–391. Springer (2016)
3. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures, vol. 1611, pp. 586–607. International Society for Optics and Photonics (1992)
4. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194. ACM Press/Addison-Wesley Publishing Co. (1999)
5. Booth, J., Antonakos, E., Ploumpis, S., Trigeorgis, G., Panagakis, Y., Zafeiriou, S., et al.: 3d face morphable models in-the-wild. In: Proceedings of the IEEE Conference on ComputerVision and Pattern Recognition (2017)
6. Bouaziz, S., Wang, Y., Pauly, M.: Online modeling for realtime facial animation. ACM Trans. Graph. (TOG) **32**(4), 40 (2013)
7. Cao, C., Hou, Q., Zhou, K.: Displaced dynamic expression regression for real-time facial tracking and animation. ACM Trans. Graph. (TOG) **33**(4), 43 (2014)
8. Cao, C., Weng, Y., Lin, S., Zhou, K.: 3d shape regression for real-time facial animation. ACM Trans. Graph. (TOG) **32**(4), 41 (2013)
9. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: a 3d facial expression database for visual computing. IEEE Trans. Vis. Comput. Graph. **20**(3), 413–425 (2014)
10. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: unified generative adversarial networks for multi-domain image-to-image translation. arXiv preprint **1711** (2017)
11. Ding, L., Ding, X., Fang, C.: 3d face sparse reconstruction based on local linear fitting. Vis. Comput. **30**(2), 189–200 (2014)
12. Fišer, J., Jamriška, O., Simons, D., Shechtman, E., Lu, J., Asente, P., Lukáč, M., Sỳkora, D.: Example-based synthesis of stylized facial animations. ACM Trans. Graph. (TOG) **36**(4), 155 (2017)
13. Frigo, O., Sabater, N., Delon, J., Hellier, P.: Video style transfer by consistent adaptive patch sampling. Vis. Comput. **35**(3), 429–443 (2019)
14. Garrido, P., Valgaerts, L., Rehmsen, O., Thormahlen, T., Perez, P., Theobalt, C.: Automatic face reenactment. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4217–4224 (2014)
15. Garrido, P., Zollhöfer, M., Casas, D., Valgaerts, L., Varanasi, K., Pérez, P., Theobalt, C.: Reconstruction of personalized 3d face rigs from monocular video. ACM Trans. Graph. (TOG) **35**(3), 28 (2016)
16. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414–2423 (2016)
17. Geng, J., Shao, T., Zheng, Y., Weng, Y., Zhou, K.: Warp-guided GANs for single-photo facial animation. In: SIGGRAPH Asia 2018 Technical Papers, p. 231. ACM (2018)
18. Jackson, A.S., Bulat, A., Argyriou, V., Tzimiropoulos, G.: Large pose 3d face reconstruction from a single image via direct volumetric CNN regression. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1031–1039 (2017)
19. Jiang, L., Zhang, J., Deng, B., Li, H., Liu, L.: 3d face reconstruction with geometry details from a single image. arXiv preprint arXiv:1702.05619 (2017)
20. Jing, Y., Yang, Y., Feng, Z., Ye, J., Song, M.: Neural style transfer: a review. CoRR **abs/1705.04058** (2017)
21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, pp. 694–711. Springer (2016)
22. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vision **1**(4), 321–331 (1988)
23. Kemelmacher-Shlizerman, I.: Internet based morphable model. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3256–3263 (2013)
24. Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Nießner, M., Pérez, P., Richardt, C., Zollhöfer, M., Theobalt, C.: Deep video portraits. arXiv preprint arXiv:1805.11714 (2018)

25. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1 × 1 convolutions. arXiv preprint arXiv:1807.03039 (2018)
26. Li, C., Wand, M.: Precomputed real-time texture synthesis with Markovian generative adversarial networks. In: European Conference on Computer Vision, pp. 702–716. Springer (2016)
27. Li, H., Yu, J., Ye, Y., Bregler, C.: Realtime facial animation with on-the-fly correctives. ACM Trans. Graph. **32**(4), 42–1 (2013)
28. Li, Y., Ma, L., Fan, H., Mitchell, K.: Feature-preserving detailed 3d face reconstruction from a single image. In: Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production, pp. 1:1–1:9. ACM, New York, NY, USA (2018)
29. Liao, J., Yao, Y., Yuan, L., Hua, G., Kang, S.B.: Visual attribute transfer through deep image analogy. arXiv preprint arXiv:1705.01088 (2017)
30. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. CoRR, arXiv:1703.07511 **2** (2017)
31. Ma, M., Peng, S., Hu, X.: A lighting robust fitting approach of 3d morphable model for face reconstruction. Vis. Comput. **32**(10), 1223–1238 (2016)
32. Parkhi, O.M., Vedaldi, A., Zisserman, A., et al.: Deep face recognition. In: BMVC, vol. 1, p. 6 (2015)
33. Pumarola, A., Agudo, A., Martinez, A.M., Sanfeliu, A., Moreno-Noguer, F.: Ganimation: anatomically-aware facial animation from a single image. arXiv preprint arXiv:1807.09251 (2018)
34. Qiao, F., Yao, N., Jiao, Z., Li, Z., Chen, H., Wang, H.: Emotional facial expression transfer from a single image via generative adversarial nets. Comput. Anim. Virtual Worlds **29**(3–4), e1819 (2018)
35. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Comput. Graph. Appl. **21**(5), 34–41 (2001)
36. Romdhani, S., Vetter, T.: Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 2, pp. 986–993. IEEE (2005)
37. Snape, P., Panagakis, Y., Zafeiriou, S.: Automatic construction of robust spherical harmonic subspaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 91–100 (2015)
38. Snape, P., Zafeiriou, S.: Kernel-PCA analysis of surface normals for shape-from-shading. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1059–1066 (2014)
39. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 175–184. ACM (2004)
40. Sumner, R.W., Popović, J.: Deformation transfer for triangle meshes. In: ACM Transactions on Graphics (TOG), vol. 23, pp. 399–405. ACM (2004)
41. Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of RGB videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2387–2395 (2016)
42. Wang, L., Wang, Z., Yang, X., Hu, S.M., Zhang, J.: Photographic style transfer. Vis. Comput. (2018). https://doi.org/10.1007/s00371-018-1609-4
43. Wu, W., Zhang, Y., Li, C., Qian, C., Loy, C.C.: Reenactgan: Learning to reenact faces via boundary transfer. arXiv preprint arXiv:1807.11079 (2018)
44. Yang, F., Wang, J., Shechtman, E., Bourdev, L., Metaxas, D.: Expression flow for 3d-aware face component transfer. ACM Trans. Graph. (TOG) **30**(4), 60 (2011)
45. Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S.Z.: Face alignment across large poses: A 3d solution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 146–155 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

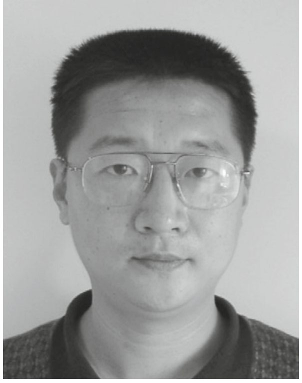


Yanlong Tang is a Ph.D. candidate of Zhejiang University. He received his B.Sc. from Shandong University in 2013. His research interests include 3D face reconstruction, image processing and computer vision.

Xiaoguang Han received his B.Sc. in mathematics in 2009 from NUAA and his M.Sc. in applied mathematics in 2011 from Zhejiang University. He obtained his Ph.D. degree in 2017 from HKU. He is currently a Research Assistant Professor at Shenzhen Research Institute of Big Data, the Chinese University of Hong Kong (Shenzhen). His research mainly focuses on computer vision, computer graphics and 3D deep learning.

Yue Li is currently a master candidate at University of Pennsylvania. He received his B.Sc. in 2018 from Beijing University of Technology. His research interests includes computer vision, computer graphics and optimization.

Liqian Ma is currently a senior researcher at Beijing Kuaishou Technology Limited. He received his B.Sc. in 2010 from Tsinghua University and obtained his Ph.D. degree from Tsinghua University in 2015. His research interests include computer graphics and computer vision.



Ruofeng Tong is a professor in Department of Computer Science, Zhejiang University. He received his B.Sc. from Fudan University in 1991 and obtained his Ph.D. degree from Zhejiang University in 1996. His research interests include image and video processing, computer graphics and computer animation.