Computational Visual Media
https://doi.org/10.1007/s41095-021-0237-5 Vol. 8, No. 2, June 2022, 225–237

Research Article

3D corrective nose reconstruction from a single image

Yanlong Tang[1], Yun Zhang[2] (✉), Xiaoguang Han[3], Fang-Lue Zhang[4], Yu-Kun Lai[5], and Ruofeng Tong[6]

Abstract There is a steadily growing range of applications that can benefit from facial reconstruction techniques, leading to an increasing demand for reconstruction of high-quality 3D face models. While it is an important expressive part of the human face, the nose has received less attention than other expressive regions in the face reconstruction literature. When applying existing reconstruction methods to facial images, the reconstructed nose models are often inconsistent with the desired shape and expression. In this paper, we propose a coarse-to-fine 3D nose reconstruction and correction pipeline to build a nose model from a single image, where 3D and 2D nose curve correspondences are adaptively updated and refined. We first correct the reconstruction result coarsely using constraints of 3D–2D sparse landmark correspondences, and then heuristically update a dense 3D–2D curve correspondence based on the coarsely corrected result. A final refinement step is performed to correct the shape based on the updated 3D–2D dense curve constraints. Experimental results show the advantages of our method for 3D nose reconstruction over existing methods.

Keywords nose shape recovery; single image 3D reconstruction; contour correspondence; Laplacian deformation

1 Tencent Games Lightspeed & Quantum Studios, Shenzhen, China. E-mail: yanlongtang@gmail.com.

2 Communication University of Zhejiang, Hangzhou, China. E-mail: zhangyun_zju@zju.edu.cn (✉).

3 Shenzhen Research Institute of Big Data, the Chinese University of Hong Kong (Shenzhen), Shenzhen, China. E-mail: hanxiaoguang@cuhk.edu.cn.

4 Victoria University of Wellington, Wellington, New Zealand. E-mail: fanglue.zhang@ecs.vuw.ac.nz.

5 Cardiff University, Wales, UK. E-mail: Yukun.Lai@cs.cardiff.ac.uk.

6 Zhejiang University, Hangzhou, China. E-mail: trf@zju.edu.cn.

Manuscript received: 2021-03-20; accepted: 2021-04-29

1 Introduction

Faces have a high degree of freedom to allow humans to express emotions, making the reconstruction of facial geometry from 2D images difficult. Despite the vast amount of work that attempts to utilize a large photo collection to resolve ambiguities when building the 3D geometry of faces, accurately reconstructing a face model from a single 2D image still remains challenging. 3D morphable model (3DMM) based fitting techniques are normally used when we only have access to a single facial image. They work to match the reconstructed 3D face mesh with the 2D contours in a facial image, including those of the face, eyes, and nose. In applications using dynamic facial models, such as facial motion re-targeting, researchers mainly focus on the reconstruction quality of parts with frequent movement, like the eyes and mouth; little attention has been paid to the nose. However, with the steadily growing range of applications that can benefit from face reconstruction techniques, the demand for accurate reconstruction of nose shapes is increasing. For example, face re-lighting requires a precise nose shape to produce a natural lighting effect in the area surrounding the nose. When creating virtual avatars in computer games, the nose shape needs to be customized by automatically manipulating bone controllers to match the input selfie. The ability to reconstruct recognizable 3D nose shapes is also important to improve recognition accuracy [1, 2], used, e.g., for 3D face unlocking of smart phones.

It is non-trivial to reconstruct accurate and identifiable 3D nose shapes from single images. There are two major challenges. On the one hand, 3D parametric face models (such as 3DMM) are unable to represent complex and diverse nose shapes due to

their limited representational power; on the other hand, more importantly, it is more difficult to establish sufficient feature constraints in the nose region than in the regions of eyes, mouth, and facial silhouette. To deal with the first challenge, previous works mainly use non-parametric deformation to correct the parametric reconstruction for further model enhancement [3–5]. However, they focus on only correcting the shape of the whole face, not just the nose, and their sparse landmarks and dense pixels are not semantically informative enough to represent various nose shapes. Recently, Tang et al. [6] introduced dense semantic curve constraints for 3D face reconstruction and correction, which makes the reconstructed mesh better match the face contours in the input image. However, their method mainly works for expressive face regions, such as eyebrows, eyes, and mouth, where the curve features are simple and salient, as shown in the middle row of Fig. 1. In the nose region, the curves can be very complex and diverse due to variations in shape and perspective, leading to erroneous matching between a pre-defined 3D nose contour and the nose contour on the 2D input image. Finally, compared with eye and mouth regions, 2D curve features on nose regions are less salient due to the similarity in color to neighboring regions: both face and nose have the color of the skin.

To tackle the aforementioned problems, we propose a coarse-to-fine 3D nose reconstruction and correction method, in which 3D and 2D nose curve correspondences can be adaptively updated and refined. Although correct dense correspondences between 3D and 2D nose curves are not easy to establish, it is observed that sparse landmarks of 3D and 2D nose shapes can be accurately established to support the reconstruction. Based on this observation, our idea is to use the sparsely reconstructed result to guide the estimation of the dense 3D–2D correspondences. We first correct the reconstruction result coarsely using constraints on 3D–2D sparse landmark correspondences, and then heuristically update dense 3D–2D curve correspondences based on the coarsely corrected result. A final refinement step is performed to correct the shape based on the updated dense 3D–2D curve constraints.

There are three problems to be solved for effectively updating dense 3D–2D curve correspondences: (i) how to determine the 3D nose contour, due to self-occlusion and variations in nose shape and pose, (ii) how to extract a precise 2D nose contour using the non-salient curve features of the boundary of the nose region, and (iii) how to establish accurate correspondences between the 3D and 2D nose contours. To extract 3D contours, Tang et al. [6] used predefined vertex indices on a template mesh as a fixed 3D nose contour, but this method is not flexible for varied nose shapes and poses. Instead, we render the sparsely corrected nose into a depth map, which can naturally form self-occlusion edges. We heuristically use this edge as the 3D nose contour to update. For 2D contour extraction, Tang et al. [6] applied snakes [7] on a feature map, but the curve features here are not distinctive enough. We produce an enhanced feature map using an RGB-D foreground enhancement method [8], where we render a depth map using the sparsely corrected 3D face mesh. Then a snake is able to extract a more accurate 2D contour. To determine 3D–2D contour correspondences, we integrate 3D contour information with 2D contour extraction, rather than dealing with them separately as in Ref. [6]. Specifically, we initialize the active contour in the snake algorithm using the projection

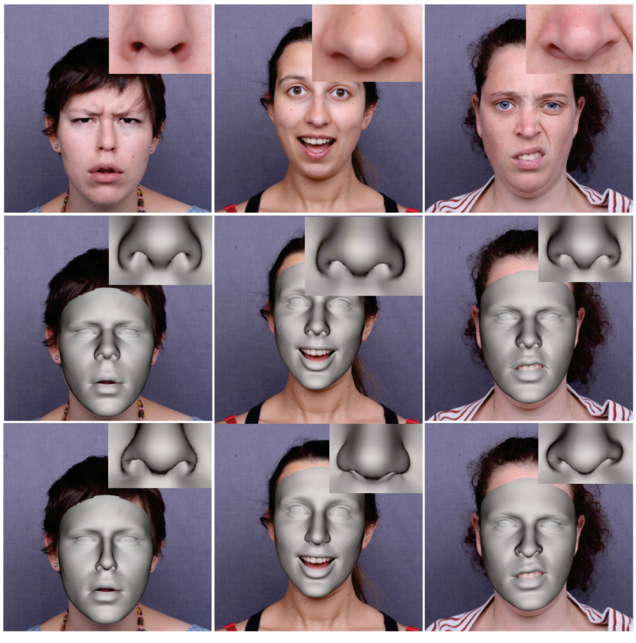


Fig. 1 Top: input images. Middle: baseline 3D face reconstruction without nose correction [6]. Bottom: our 3D face reconstruction with nose correction.

of the heuristically determined 3D contour. In this way, no matter how complex the 3D nose curve is, proper correspondences can be preserved. In contrast, the matching method used in Ref. [6] may produce erroneous correspondences when the curve is complex.

We believe our work to be the first to reconstruct accurate 3D noses from single images. Experiments show that our method outperforms the state-of-the-art. We make the following technical contributions:

- a coarse-to-fine 3D nose reconstruction approach, which can adaptively and heuristically build and correct dense 3D–2D nose contour correspondences to adapt to different face poses and nose shapes, and
- an improved 2D nose contour feature detection method integrating the RGB-D foreground enhancement method.

2 Related work

Low-dimensional parametric 3D face models [9–16] are widely used for 3D face reconstruction for their simplicity, compactness, and effectiveness. However, limited by the wide range of types of models and their formats in model databases, low-dimensional models cannot be used to reconstruct sufficiently accurate face shapes, especially when the face greatly differs from those in the model database. Therefore, it is a necessary step to further correct the reconstructed low-dimensional 3D faces to better match the input data.

Numerous studies [3, 4, 17] have investigated how to use Laplacian deformation [18] to correct low-dimensional 3D face reconstruction results. Their idea is to correct the position of each vertex in a high-dimensional feature space to better match the input data, where the local structure is maintained by a Laplacian coordinate regularization term. Li et al. [3] used RGB-D data to correct the whole face, and correct the nose depending on the dense depth data, which is however unavailable when only a single image can be accessed. Thus, for single image input, Li et al. [4] approximately converted detected 2D sparse landmarks to 3D space to correct the whole face. However, sparse landmarks in the nose area are not dense enough to describe the nose shape, and the corrective effect is thus limited. For video input, Garrido et al. [17] corrected the whole face based on dense optical flow constraints, but the optical flow calculation depends on having video input and is not applicable for single image input. As high dimensional Laplacian deformation [18] in vertex space has a high computational cost and is not robust to noise, some researchers have suggested [5, 17, 19] solving Laplacian deformation in a low-dimensional subspace [20] to speed up the computation and/or reduce noise. In work like that of Li et al. [3] and Bouaziz et al. [19], producing corrected meshes relies on depth data, which is again not applicable to a single image. For single image input, a series of recent studies has indicated that the deformation problem can be solved by utilizing the dense pixel difference between the rendered image and input image [5, 17]. However, it needs to solve parametric albedo and illumination models at the same time, so is also greatly affected by the representational power of the parametric illumination and albedo model. Pixel level dense constraints (depth or image pixels) are usually used to supplement sparse landmark constraints, and are especially suitable to represent medium level wrinkle deformations in skin regions such as the forehead and cheek, where sparse landmark constraints cannot model them well. On the other hand, pixel level dense constraints usually contain a lot of noise and do not show salient contour-level semantic features, so cannot correct feature regions properly. In addition, although low-dimensional subspace Laplacian deformation [20] is more efficient and smooth, the deformation is limited to a narrow range.

The above works aim to correct the whole face to fit the sparse or dense input data. However, in their reconstructed results, local feature regions such as the eyes, mouth, and nose are still not identifiable or expressive enough. Compared to sparse landmarks and dense pixel features, contour features contain more semantic information so can model parts of the face better, and thus can be used to further correct local shapes. For eyelid correction, Wen et al. [21] built a parametric eyelid model to fit the extracted 2D eyelid contour, but their 2D eyelid contour extraction relies on manually labeled data for training. For lip correction, Garrido et al. [22] learned a mapping from inaccurate 3D lips to accurate 3D lips. But the accurate 3D lip data set needs to be collected and processed by complex and expensive equipment, and they also required manually labeled data to train the 2D lip contour

extraction model. Dinev et al. [23] also corrected lips using a data-driven method. Differing from Ref. [22], they constructed a training dataset using lightweight Laplacian deformation techniques [18]. However, they need to manually extract the 2D lip contour, and sometimes need to heuristically label lips due to occlusion between upper and lower lips. All the above correction methods involve some manual intervention for 2D contour extraction; more lightweight and fully automatic 2D contour extraction methods would be preferable to reduce the manual burden. More recently, Tang et al. [6] proposed a lightweight 2D contour extraction approach to correct local facial features. When extracting 2D contour, they used a local-to-global snake algorithm [7] to refine the initial connection lines between landmarks. However, their method is more suitable for eye and mouth regions where the features are salient and simple. It does not work well for noses because of their more complex shape.

To the best of our knowledge, no previous works target to correct nose reconstruction in the field of single-image-based facial reconstruction. Compared to eye and lip correction [21–23], it is more challenging to establish accurate dense 3D–2D contour correspondences for nose correction. To deal with this challenge, we couple 3D reconstruction and 2D feature extraction instead of dealing with them separately [21–23], which effectively improves the dense 3D–2D nose correspondence. In our approach, in order to allow a flexible 3D nose contour for varied face poses and nose shapes, we heuristically refine the 3D nose contour in a coarse-to-fine scheme during reconstruction. To mitigate the ambiguity when extracting the 2D nose contour using less salient curve features, we combine the reconstructed depth information to improve 2D contour extraction instead of extracting features based only on 2D input data [6, 21, 22]. For 3D–2D one-to-one contour correspondences, as the iterative closest point (ICP) method may find wrong correspondences for complex nose shapes, we implicitly preserve correct correspondence by deforming the 2D projection of the 3D nose contour to produce the final 2D contour using a snake algorithm [7].

3 Method

3.1 Overview

Previously, single-image-based 3D face reconstruction commonly encountered difficulties in reconstructing accurate and identifiable 3D nose shapes. In this paper, we propose and develop a method which makes the reconstructed 3D nose accurately match the 2D nose contour in the input image, as shown in Fig. 2. The key challenge in 3D nose reconstruction

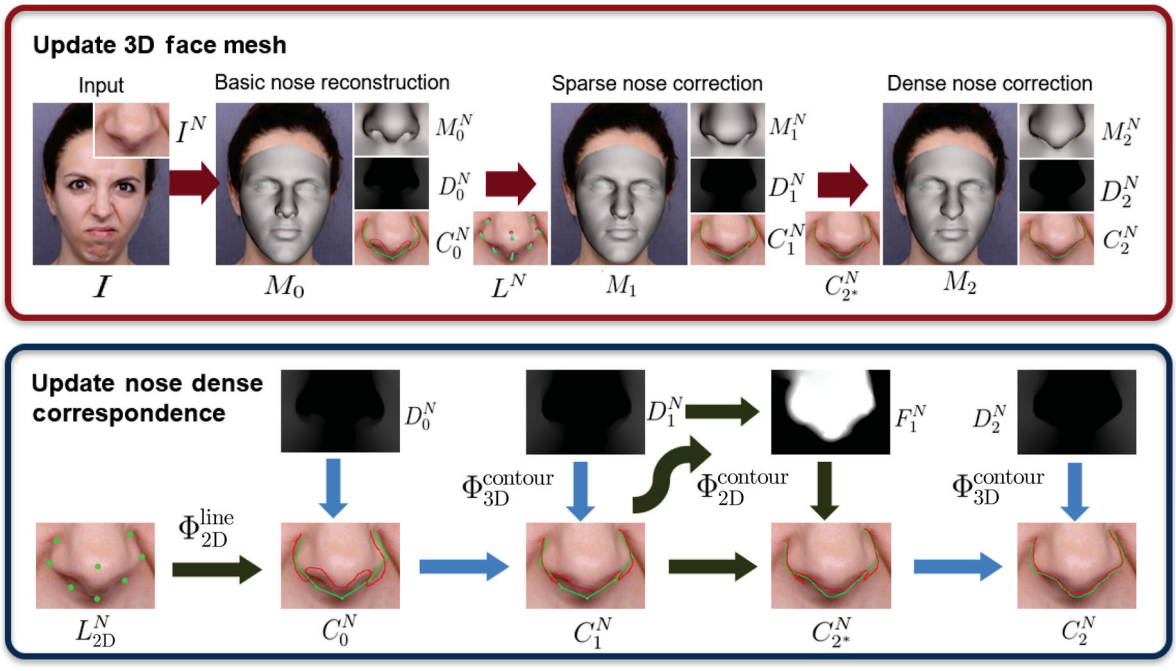


Fig. 2 Pipeline of proposed 3D corrective nose reconstruction method.

is to establish sufficiently accurate 3D–2D feature correspondences that can adapt to varied face poses and nose shapes. Our basic idea is to update the 3D nose shape M_i^N and the 3D–2D nose correspondence C_i^N in a coarse-to-fine manner. In the process, the 3D–2D correspondence is heuristically updated based on 3D nose shape changes. Then, the 3D nose shape is iteratively refined based on the updated nose correspondences. Overall, the process has three stages: basic nose reconstruction, sparse nose correction, and dense nose correction.

The mathematical notation used in this paper is summarized in Table 1. The nose reconstruction process is formulated as a three-stage optimization problem with the following objective:

$$\begin{aligned} E(P, M, C^N) = {} & \sigma_0 E_{\text{basic}}(P, M, C^N) \\ & + \sigma_1 E_{\text{sparse}}(M^N, C^N) \\ & + \sigma_2 E_{\text{dense}}(M^N, C^N) \end{aligned} \quad (1)$$

where the targets to be determined include camera parameters P, the 3D face mesh M (with nose part M^N), and 3D–2D nose correspondences C^N. $C^N = (C^{N,2D}, C^{N,3D})$ contains one-to-one nose correspondences between the 2D point set $C^{N,2D}$ and the 3D mesh vertex set $C^{N,3D}$. In each reconstruction stage, only a single energy term in Eq. (1) is activated. We now consider each stage.

(1) *Basic nose reconstruction stage*. In this stage, an initial 3D nose is reconstructed with energy weights $\sigma_0 = 1$, $\sigma_1 = 0$, $\sigma_2 = 0$. The optimization objective is $E(P, M, C^N) = E_{\text{basic}}(P, M, C^N)[L^A, C_0^{A-N}]$ [6], where camera parameters P, whole face mesh M, and nose correspondence C^N are all found based on all 3D–2D sparse correspondences $L^A = (L^{A,2D}, L^{A,3D})$ and partial 3D–2D dense correspondences $C_0^{A-N} = (C_0^{A-N,2D}, C_0^{A-N,3D})$ (excluding the nose dense correspondences, as they are not accurate yet). This stage outputs the basic 3D nose shape M_0^N and 3D–2D nose dense correspondences C_0^N.

(2) *Sparse nose correction stage*. In this stage, we refine the results of the first stage using energy weights $\sigma_0 = 0$, $\sigma_1 = 1$, $\sigma_2 = 0$. The optimization is formulated as $E(P, M, C^N) = E_{\text{sparse}}(M^N, C^N)[L^N]$, where camera parameters P are fixed, and only 3D nose shape M^N and nose correspondence C^N are determined. The nose 3D–2D sparse correspondence $L^N = (L^{N,2D}, L^{N,3D})$ is used as a constraint. This stage outputs the roughly corrected 3D nose M_1^N and updated nose correspondence C_1^N.

(3) *Dense nose correction stage*. In this stage, we further refine the second stage results, with energy weights $\sigma_0 = 0$, $\sigma_1 = 0$, $\sigma_2 = 1$. The optimization becomes $E(P, M, C^N) = E_{\text{dense}}(M^N, C^N)[C_{2^*}^N]$, where we first update the nose correspondences from C_1^N to $C_{2^*}^N$ as energy constraints, and then solve for the 3D nose shape M^N and update the nose correspondences C^N, giving the final results M_2^N and C_2^N.

We use the 3D *face model* from Ref. [12] for reconstruction. In this model, a 3D face mesh can be represented in two forms, in high-dimensional space and low-dimensional space. In the former, a 3D face is represented by all of its vertices, while in the latter,

Table 1 Notation

Notation	Description
Camera	
P	camera parameters, including $P = \{Pr, R, t\}$
Pr	weak perspective projection matrix
R	rotation matrix
t	translation vector
Π	obtain projected 3D point in image space
Π_{xy}	obtain 2D position (x–y components) of projected 3D point
Π_z	obtain depth value (z component) of projected 3D point
Image	
I	input face image
I^N	nose region of input face image (N indicates "Nose")
F_1^N	enhanced nose feature map during first optimization stage
Mesh	
M	target 3D face mesh to be found
M_i	3D face mesh found by optimization stage i
M_i^N	nose region of 3D face mesh found by optimization stage i
D_i^N	rendered nose depth map of 3D face mesh in optimization stage i
Correspondences	
L^A	correspondence of all 3D–2D sparse landmarks (A indicates "All")
L^N	correspondence of nose 3D–2D sparse landmarks
C^N	target 3D–2D dense nose correspondences to be found
C_i^N	3D–2D dense nose correspondence result of stage i
C_i^A	3D–2D dense face correspondence result of stage i
C_i^{A-N}	3D–2D dense correspondence result for face excluding nose, for stage i
Operation	
Φ_{2D}^{line}	generate 2D nose contour by connecting landmarks
$\Phi_{2D}^{\text{contour}}$	update 2D nose contour using snake
$\Phi_{3D}^{\text{contour}}$	extract 3D nose contour from depth map
$\mathcal{F}$	obtain enhanced feature map using RGB-D image

it is represented by a small number of parameters. In the basic nose reconstruction stage, the 3D face mesh M is first obtained in the low-dimensional space, which is represented by the following set of parameters: $M(\alpha,\beta) = M_{\text{mean}} + B_{\text{id}} \cdot \alpha + B_{\text{exp}} \cdot \beta$ [12], where α and β represent identity and expression parameters respectively. In all three stages, the 3D face mesh is corrected in the high-dimensional space. The face mesh is represented in the form of a high-dimensional vector of vertex positions: $M(V) = \{v_i\}_{i=1}^{n}$, where v_i is the 3D position of the i-th vertex.

In the basic nose reconstruction stage, the *camera parameters* P are determined and fixed; in the next two stages, P is used to inversely project 2D points in image space to 3D space. P is represented by $\{Pr, R, t\}$, including a weak perspective projection matrix Pr, a rotation matrix R, and a translation vector t. We formulate the weak perspective projection from 3D to 2D as

$$v^{\text{proj}} = \Pi(v_{\text{3D}}) \tag{2}$$

which can be further expanded as

$$\begin{pmatrix} v_{\text{2D}} \\ d \end{pmatrix} = Pr \cdot (R \cdot v_{\text{3D}} + t) \tag{3}$$

where $v^{\text{proj}} = \begin{pmatrix} v_{\text{2D}} \\ d \end{pmatrix}$ represents the position after 3D point v_{3D} is projected to 2D image space. v_{2D} is the projected 2D position and d is the depth value. $\Pi = \Pi(Pr, R, t)$ represents the model-view matrix. $Pr = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}$ represents the weak perspective projection matrix. R represents a 3D rotation matrix and t is a 3D translation. For convenience, we decompose the 3D projection formula into

$$v_{\text{2D}} = \Pi_{xy}(v_{\text{3D}}) \tag{4}$$

and

$$d = \Pi_{z}(v_{\text{3D}}) \tag{5}$$

To get a unique result when inversely projecting a 2D point to 3D, the 2D point's depth value should be known in advance. Thus the inverse projection from v_{2D} to the 3D point v_{3D} is

$$v_{\text{3D}} = \Pi^{-1}(v^{\text{proj}}) \tag{6}$$

which can be further expanded as

$$v_{\text{3D}} = R^{-1}\left(Pr^{-1}\begin{pmatrix} v_{\text{2D}} \\ d \end{pmatrix} - t\right) \tag{7}$$

3.2 Basic nose reconstruction

Tang et al.'s recent work [6] proposed a 3D facial reconstruction method based on dense contour features, which can faithfully reconstruct 3D faces, especially exaggerated faces. Such a method of establishing 3D–2D dense contour correspondences does not produce good correspondences for nose reconstruction, as the 2D nose contour is more difficult to extract and 3D nose contour varies with different poses and shapes. Therefore, we just apply the method of Ref. [6] for initialization, and facial regions except for the nose are corrected. The optimization objective of the initial nose reconstruction is formulated as

$$\begin{aligned} E(P, M, C^N) &= E_{\text{basic}}(P, M, C^N)[L^A, C_0^{A-N}] \\ &= \omega_1 E_{\text{sparse}}^{\text{fit}}[L^A] + \omega_2 E_{\text{dense}}^{\text{fit}}[C_0^{A-N}] \\ &\quad + \omega_3 E_{\text{reg}}^{\text{fit}} + \omega_4 E_{\text{dense}}^{\text{correct}}[C_0^{A-N}] \end{aligned} \tag{8}$$

where P, M, and C^N are camera parameters, objective 3D face mesh, and nose correspondences respectively as in Eq. (1). ω_i is the weight of each energy term. $E_{\text{sparse}}^{\text{fit}}[L^A]$ is the low-dimensional fitting energy using all sparse landmarks L^A as constraints. $E_{\text{dense}}^{\text{fit}}[C_0^{A-N}]$ is the low-dimensional fitting energy using all dense contours except for the nose contour C_0^{A-N} as constraints. $E_{\text{reg}}^{\text{fit}}$ is the low-dimensional regularization energy which keeps the parameters in a reasonable range. $E_{\text{dense}}^{\text{correct}}[C_0^{A-N}]$ represents the high-dimensional correction energy based on all dense contours excluding the nose C_0^{A-N}.

We solve the above optimization problem in three stages following Ref. [6]. In the first stage, we estimate a 3D mesh in a low-dimensional space using sparse constraints, with energy weights $\omega_1 = 1.0$, $\omega_2 = 0.0$, $\omega_3 = 0.05$, and $\omega_4 = 0.0$. In the second stage, dense constraints are introduced to the fitting for refinement. Energy weights are $\omega_1 = 0.005$, $\omega_2 = 15.0$, $\omega_3 = 2.0$, and $\omega_4 = 0.0$. In the third stage, high-dimensional correction is based on dense constraints, with energy weights $\omega_1 = 0.0$, $\omega_2 = 0.0$, $\omega_3 = 0.0$, and $\omega_4 = 1.0$.

Our initial results show that except for the nose region, the other regions better match the feature contours of the image. Based on the initial reconstructed mesh, we initialize the dense 3D–2D nose contour correspondence as follows:

$$C_0^N = (C_0^{N,\text{2D}}, C_0^{N,\text{3D}}) = (\Phi_{\text{2D}}^{\text{line}}(L^{N,\text{2D}}), \Phi_{\text{3D}}^{\text{contour}}(D_0^N)) \tag{9}$$

where C_0^N is the initialized nose dense 3D–2D correspondence, $C_0^{N,2D} = \Phi_{2D}^{line}(L^{N,2D})$ represents the initialized 2D nose contour, generated by connecting nose landmarks $L^{N,2D}$ with straight lines. $C_0^{N,3D} = \Phi_{3D}^{contour}(D_0^N)$ represents the 3D nose contour, extracted from the rendered nose depth map D_0^N. The nose depth map D_0^N is rendered from the reconstructed nose region mesh M_0^N. In D_0^N, pixels belonging to nose regions are set to white and other pixels are set to black. The 2D contour is detected from the binary mask and the projected 3D nose vertices that are closest to the contour are found by nearest neighbor search, giving the initial 3D nose contour $C_0^{N,3D}$.

3.3 Sparse nose correction

The nose shape reconstructed by the method in Ref. [6] appears quite different from the ground truth. However, as stated before, dense nose 3D–2D contour correspondences cannot be directly generated like those for the eyes and lips due to the difficulties in extracting both 2D and 3D nose contours. While sparse nose landmarks are insufficient to describe nose shape, they usually can be accurately detected. Based on this observation, weak nose correction [18] is performed using the sparse nose landmarks, allowing the reconstructed 3D nose shape to be roughly corrected to fit the 2D nose shape better. Moreover, with this sparse correction result, dense nose correspondences can be further refined. This sparse nose correction optimization can be formulated as

$$\begin{aligned} E(P, M, C^N) &= E_{\text{sparse}}(M^N, C^N)[L^N] \\ &= \sum_i^n \|\pounds(v_i^*) - \pounds(v_i)\|_2 \\ &\quad + \omega \sum_{l_j^{2D} \in L^{N,2D}} \left\| v_j^* - l_j^{3D} \right\|_2 \end{aligned} \tag{10}$$

where M^N is the nose mesh represented by its vertices. L^N is sparse landmark correspondence, used as optimization constraints. $\pounds$ is the Laplacian operator [18]. ω is a weight to balance the landmark matching term and the Laplacian term, with an experimentally determined value of 5.0. Using the inverse projection Eq. (6), each 2D point l_j^{2D} in the sparse correspondence can be approximately converted to a 3D point:

$$l_j^{3D} = \Pi^{-1} \begin{pmatrix} l_j^{2D} \\ \Pi_z(v_j) \end{pmatrix} \tag{11}$$

The sparse nose correction not only makes the reconstructed 3D nose approach the 2D shape, but also heuristically updates the 3D nose contour for better dense 3D–2D nose correspondences. The nose correspondence is updated using

$$C_1^N = (C_1^{N,2D}, C_1^{N,3D}) = (C_0^{N,2D}, \Phi_{3D}^{contour}(D_1^N)) \tag{12}$$

where C_1^N is the updated nose dense correspondence in the sparse nose correction stage. $C_1^{N,2D} = C_0^{N,2D}$ is the 2D nose contour before updating. $C_1^{N,3D} = \Phi_{3D}^{contour}(D_1^N)$ indicates the heuristically updated 3D nose contour using the sparsely corrected nose result D_1^N.

3.4 Dense nose correction

After sparse nose correction, the 3D nose shape is closer to the 2D input, but the quality of the result is not sufficient for use in personalized applications. Thus, we further perform dense nose correction to get accurate dense 3D–2D nose contour correspondence.

3.4.1 Updating dense nose correspondences

In the previous sparse correction stage, the 3D nose contour is heuristically updated to better match the 2D input. However, the 2D nose contour is still inaccurate. Traditional works use a low-level edge detection method [27] to detect 2D facial contours. The resulting contours may be noisy or jagged due to the lack of a shape prior. We overcome this problem by employing a snake algorithm [7] to combine both low-level image features and a high-level shape prior. A snake is an active contour model which introduces an external fitting energy term to optimize the objective contour to match the low-level image features, such as edges and brightnesss. An internal regular energy term preserves the contour shape and smoothness. Snake-based 2D contour updating can be formulated as

$$C = \Phi_{2D}^{contour}(C_{\text{init}}, F) \tag{13}$$

where C is the updated 2D contour, C_{init} is the initial contour, and F is the feature map of the target image used to fit the active contour.

Previous work [6] has also employed snakes to extract facial contours. In that work, the initial contour is composed of straight lines connecting nose landmarks, and the feature map is the intensity map of the gray image. Their method produces good results for expressive regions, such as eyes and lips, but is not applicable to extracting the nose contour.

Unlike the eyes and lips, edge features are indistinct in nose regions because the skin colors of the nose and its surrounding regions are similar. We thus generate an enhanced feature map F using the RGB-D saliency detection method in Ref. [8], where the depth map D_1^N is rendered from the reconstructed 3D face mesh. Furthermore, as the shape of the nose is more complex than the eyes and lips, the ICP method used in Ref. [6] may result in incorrect 3D–2D correspondences. We instead set the initial contour C_{init} as the 2D projection of the 3D nose contour $\Pi_{xy}(C_1^N)$, which can implicitly establish accurate 3D–2D correspondences in an adaptive way. The above dense nose 3D–2D correspondence update process can be formulated as

$$\begin{aligned} C_{2^*}^N &= (C_{2^*}^{N,2\text{D}}, C_{2^*}^{N,3\text{D}}) \\ &= (\Phi_{2\text{D}}^{\text{contour}}(\Pi_{xy}(C_1^{N,3\text{D}}), F_1^N), C_1^{N,3\text{D}}) \end{aligned} \tag{14}$$

where $C_{2^*}^N$ is the updated dense correspondence, and $C_{2^*}^{N,3\text{D}} = C_1^{N,3\text{D}}$ represents the 3D nose contour in the previous sparse correction stage. $C_{2^*}^{N,2\text{D}} = \Phi_{2\text{D}}^{\text{contour}}(\Pi_{xy}(C_1^{N,3\text{D}}), F_1^N)$ indicates the updated 2D nose contour based on the snake method (Eq. (13)). In 2D nose updating, the initial nose contour $\Pi_{xy}(C_1^{N,3\text{D}})$ is the 2D projection of the 3D nose contour $C_1^{N,3\text{D}}$, which can implicitly preserve the 3D–2D correspondences when the 2D contour deforms. The feature map F_1^N used for the snake algorithm is an enhanced feature map generated by the RGB-D saliency detection method [8]. $F_1^N = \mathcal{F}(I^N, D_1^N)$ represents the feature map calculated from the RGB image I^N and the depth map D_1^N of the nose. As both 3D and 2D contours are evolved from $C_1^{N,3\text{D}}$, accurate dense 3D–2D nose correspondences can be implicitly preserved without any additional computation such as ICP.

When calculating the enhanced feature map F_1^N using the RGB-D saliency detection method, we compute the probability of each pixel belonging to the foreground, resulting in enhanced edges. We modify the original method [8] to better suit our task. Specifically, the random walk seeds for foreground and background are sampled on different sides of the banded area formed by $C_1^{N,2\text{D}}$ and $\Pi_{xy}(C_1^{N,3\text{D}})$, and we set the random walk weight graph using the depth information for regularization, to constrain the resulting foreground boundary to be close to the input nose boundary in the depth map.

3.4.2 Dense nose correction

With the updated dense nose 3D–2D contour correspondences, we correct the nose shape in the high-dimensional space:

$$\begin{aligned} E(P, M, C^N) &= E_{\text{dense}}(M^N, C^N)[C_{2^*}^N] \\ &= \sum_{i=1}^{n} \|\mathcal{L}(v_i^*) - \mathcal{L}(v_i)\|_2 \\ &\quad + \omega \sum_{c_j^{2\text{D}} \in C_{2^*}^{N,2\text{D}}} \left\| v_j^* - c_j^{3\text{D}} \right\|_2 \end{aligned} \tag{15}$$

where M^N is the target 3D nose to be corrected. $C_{2^*}^N$ is the 3D–2D correspondence of nose contour (Eq. (14)), used as constraints. ω is a weight to balance the landmark matching term and Laplacian term, with an experimentally determined value of 5.0. Each 2D point $c_j^{2\text{D}}$ in the dense correspondence can be converted into a 3D point approximately by

$$c_j^{3\text{D}} = \Pi^{-1} \begin{pmatrix} c_j^{2\text{D}} \\ \Pi_z(v_j) \end{pmatrix} \tag{16}$$

where the depth value is rendered using the corresponding 3D vertex $\Pi_z(v_j)$.

After dense nose correction, an accurate 3D nose shape M_2^N is generated. As in Eq. (12), the dense correspondence can be further updated by

$$C_2^N = (C_2^{N,2\text{D}}, C_2^{N,3\text{D}}) = (C_{2^*}^{N,2\text{D}}, \Phi_{3\text{D}}^{\text{contour}}(D_2^N)) \tag{17}$$

to give the final output of the dense 3D–2D contour correspondence.

4 Experiments

4.1 Comparison to the state-of-the-art

We compared our method with Tang et al.'s state-of-the-art image-based 3D face reconstruction method [6] using the Stirling ESRC 3D face dataset [24]: see Fig. 3. The experimental results demonstrate that our method outperforms it by reconstructing better, personalized, distinctive nose shapes. Further quantitative comparisons with optimization based methods [6, 26] on the BU-3DFE dataset [25] numerically demonstrate the advantage of our method: see Fig. 4. Additionally, we compared our method to recent learning based methods [15, 16], again showing the better performance of our method: see Fig. 5.

4.2 Ablation study

We conduct ablation experiments to demonstrate the

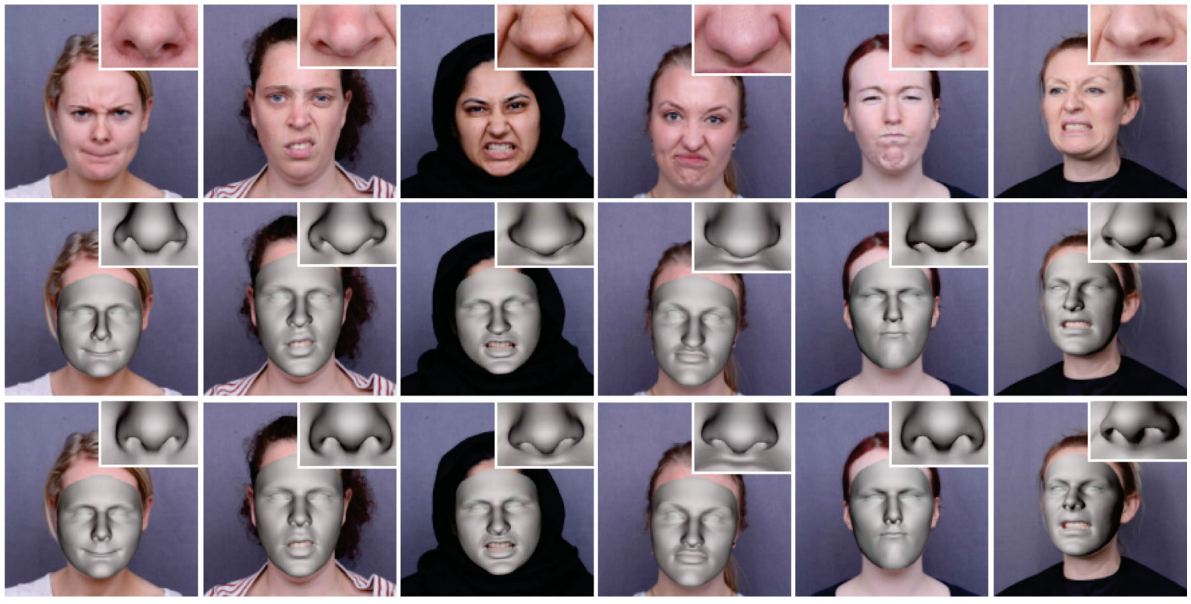


Fig. 3 Comparison with Tang et al.'s state-of-the-art method [6] using the Stirling ESRC dataset [24]. Top: input images. Middle: results of our method. Bottom: results using Tang et al.'s method.

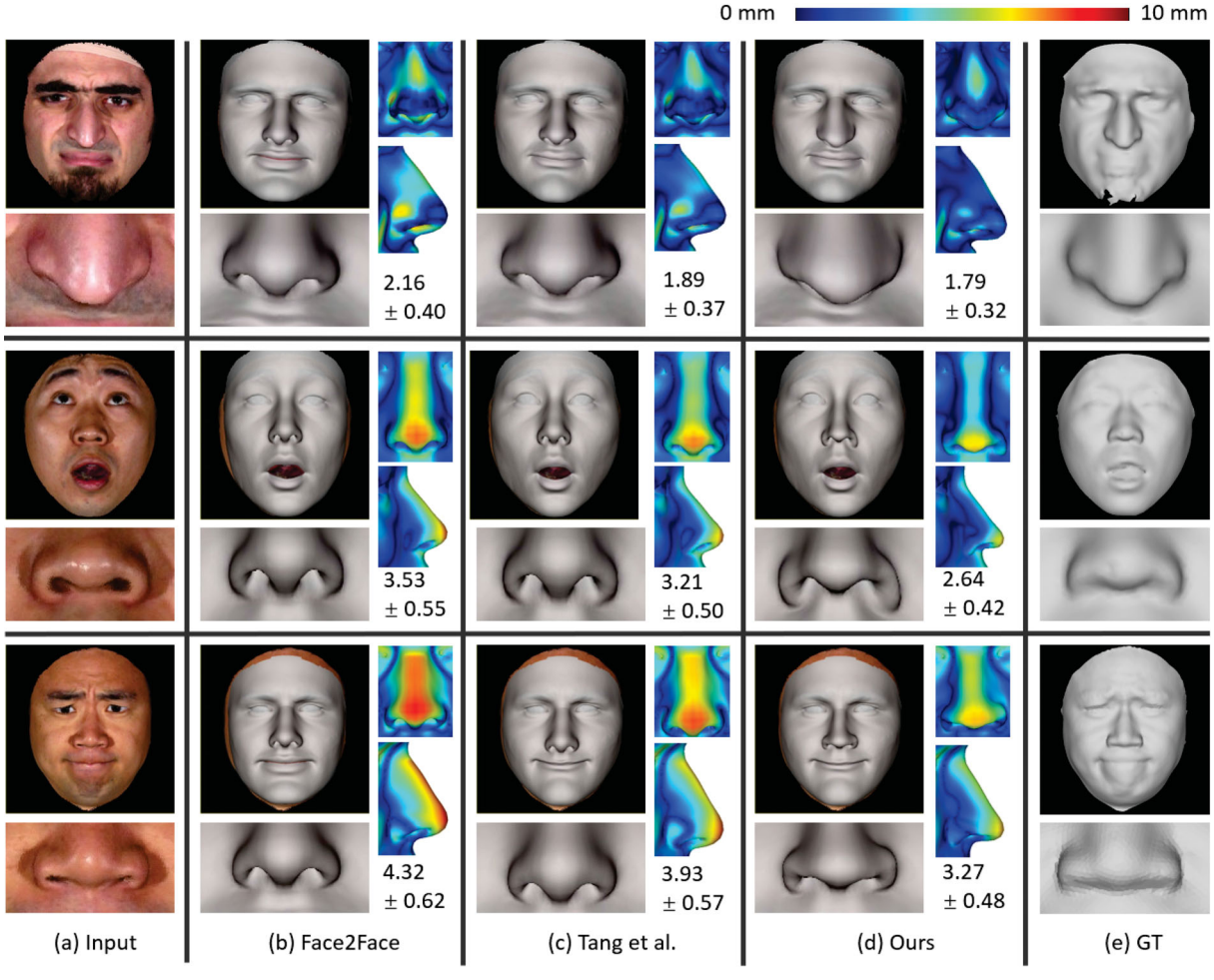


Fig. 4 Comparison with state-of-the-art optimization based methods using the BU-3DFE dataset [25]: (a) input images; (b) results of Face2Face [26]; (c) results of Tang et al. [6]; (d) results of our method; (e) ground truth 3D meshes. Reconstruction error (in mm) is visualized in red/blue color maps, with root mean squared error and standard deviation given below the color maps.

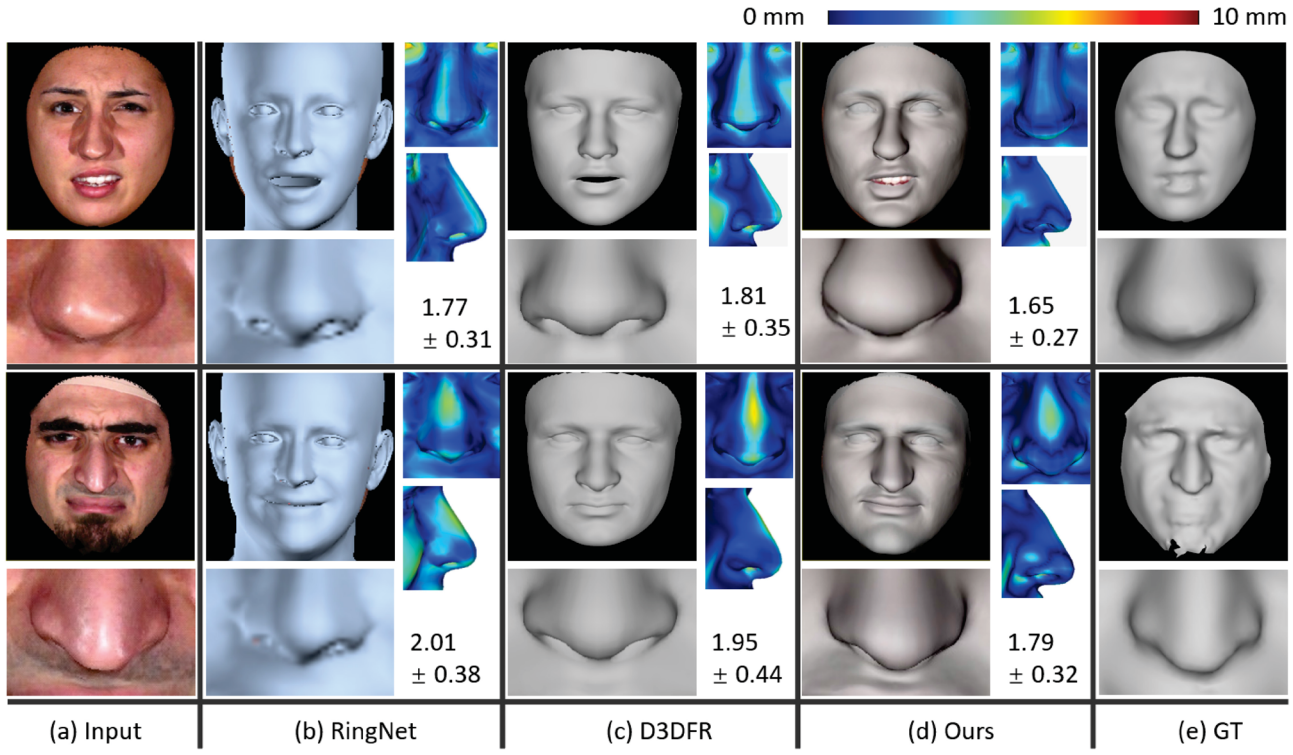


Fig. 5 Comparison with state-of-the-art learning based methods using the BU-3DFE dataset [25]: (a) input images; (b) results of RingNet [15]; (c) results of D3DFR [16]; (d) results of our method; (e) ground truth 3D meshes. Reconstruction error (in mm) is visualized in red/blue color maps, with root mean squared error and standard deviation given below the color maps.

roles of all three stages of our method. The results after each stage are shown in Fig. 6. It demonstrates that both sparse and dense correction can significantly improve nose reconstruction. In the first example, nose wings are improved in the final result. In the second example, the overall shape and position of the model are improved. In the third example, the final reconstructed results have lower nostrils, better matching the input image.

4.3 Fixed versus updated 3D contour

Successful nose correction relies on adequate accuracy of matched features in the nose region. The 3D nose contour must match the 2D contour; otherwise, the reconstructed results cannot accurately recover the shape of the nose in the 2D image. Our 3D contour updating scheme is designed with that aim. In Fig. 7, we compare the results of using a fixed 3D nose contour and our proposed heuristic 3D nose contour updating scheme, where we can see that our method provides much better results.

4.4 2D contour updating

The traditional snake method is used to update the 2D contour based on the intensity feature map of the image. However, features on the intensity map are not distinctive, often leading to poor nose boundaries. Our enhanced feature map generated from RGB-D data is designed to cope with this problem. In Fig. 8, we compare the results based on feature maps generated from the intensity map, the RGB saliency map and the RGB-D saliency map, showing that the RGB-D saliency map significantly improves the quality of the 2D contour and further improves the quality of nose correction. The 3D nose tip shape generated by the proposed method better matches the input image for pointed noses.

5 Conclusions

In this paper, we proposed a 3D nose reconstruction method which adaptively updates the nose model to better match the input 2D facial image. Our method utilizes coarse-to-fine 3D nose correction in its reconstruction approach, which adaptively and heuristically builds and updates dense 3D–2D nose contour correspondences to adapt to different face poses and nose shapes. We also improve 2D nose contour detection using an enhanced feature map generated from RGB-D data rendered from the

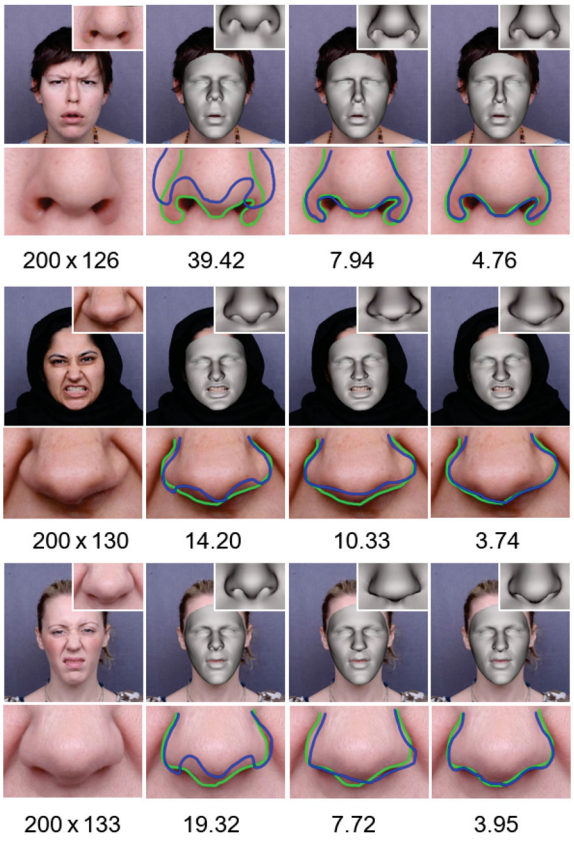


Fig. 6 Ablation study results. Columns, left to right: input images, first stage results, second stage results, third stage (final) results of the proposed method. Numbers below the first column give the resolution of the nose region; numbers below other columns are mean pixel errors between the reconstructed nose contour (blue) and the ground truth nose contour (green).

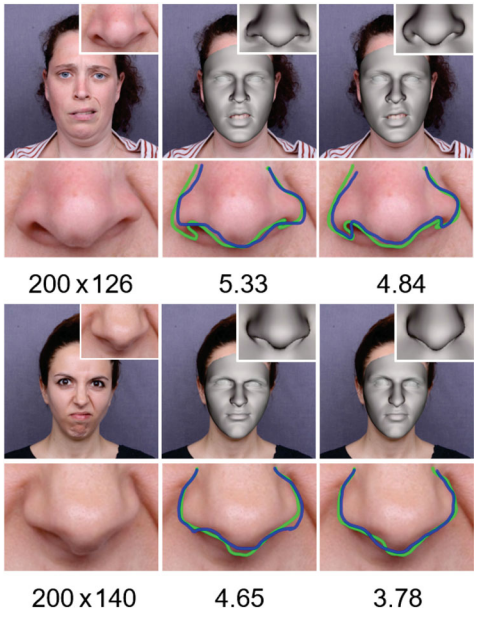


Fig. 7 3D nose contour updating benefits. Left to right: input images, results without 3D contour updating, results with 3D contour updating. Numbers have meanings as in previous figure.

Fig. 8 Utility of 2D contour update. Left to right: input image, results using intensity map, results using RGB saliency map, results using RGB-D saliency map.

intermediate nose model. Our experiments show the improved quality of noses reconstructed using our method, compared to the current state-of-the-art facial reconstruction method.

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61972342, 61602402, and 61902334), Zhejiang Provincial Basic Public Welfare Research (Grant No. LGG19F020001), Shenzhen Fundamental Research (General Project) (Grant No. JCYJ20190814112007258), and the Royal Society (Grant No. IES\R1\180126).

References

[1] Samad, M. D.; Iftekharuddin, K. M. Frenet frame-based generalized space curve representation for pose-invariant classification and recognition of 3-D face. *IEEE Transactions on Human-Machine Systems* Vol. 46, No. 4, 522–533, 2016.

[2] Werghi, N.; Tortorici, C.; Berretti, S.; Del Bimbo, A. Boosting 3D LBP-based face recognition by fusing shape and texture descriptors on the mesh. *IEEE Transactions on Information Forensics and Security* Vol. 11, No. 5, 964–979, 2016.

[3] Li, H.; Yu, J. H.; Ye, Y. T.; Bregler, C. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics* Vol. 32, No. 4, Article No. 42, 2013.

[4] Li, Y.; Ma, L. Q.; Fan, H. Q.; 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, Article No. 1, 2018.

[5] Jiang, L.; Zhang, J. Y.; Deng, B. L.; Li, H.; Liu, L. G. 3D face reconstruction with geometry details from a single image. *IEEE Transactions on Image Processing* Vol. 27, No. 10, 4756–4770, 2018.

[6] Tang, Y. L.; Han, X. G.; Li, Y.; Ma, L. Q.; Tong, R. F. Expressive facial style transfer for personalized memes mimic. *The Visual Computer* Vol. 35, 783–795, 2019.

[7] Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: Active contour models. *International Journal of Computer Vision* Vol. 1, No. 4, 321–331, 1988.

[8] Tang, Y. L.; Tong, R. F.; Tang, M.; Zhang, Y. Depth incorporating with color improves salient object detection. *The Visual Computer* Vol. 32, 111–121, 2016.

[9] 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, 187–194, 1999.

[10] Paysan, P.; Knothe, R.; Amberg, B.; Romdhani, S.; Vetter, T. A 3D face model for pose and illumination invariant face recognition. In: Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance, 296–301, 2009.

[11] Booth, J.; Roussos, A.; Zafeiriou, S.; Ponniah, A.; Dunaway, D. A 3D morphable model learnt from 10,000 faces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5543–5552, 2016.

[12] Zhu, X. Y.; Lei, Z.; Liu, X. M.; Shi, H. L.; Li, S. Z. Face alignment across large poses: A 3D solution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 146–155, 2016.

[13] Cao, C.; Weng, Y. L.; Zhou, S.; Tong, Y. Y.; Zhou, K. FaceWarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* Vol. 20, No. 3, 413–425, 2014.

[14] Feng, Y.; Wu, F.; Shao, X. H.; Wang, Y. F.; Zhou, X. Joint 3D face reconstruction and dense alignment with position map regression network. In: *Computer Vision–ECCV 2018. Lecture Notes in Computer Science, Vol.11218.* Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 557–574, 2018.

[15] Sanyal, S.; Bolkart, T.; Feng, H. W.; Black, M. J. Learning to regress 3D face shape and expression from an image without 3D supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 7755–7764, 2019.

[16] Deng, Y.; Yang, J. L.; Xu, S. C.; Chen, D.; Jia, Y. D.; Tong, X. Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 285–295, 2019.

[17] Garrido, P.; Zollhöfer, M.; Casas, D.; Valgaerts, L.; Varanasi, K.; Pérez, P.; Theobalt, C. Reconstruction of personalized 3D face rigs from monocular video. *ACM Transactions on Graphics* Vol. 35, No. 3, Article No. 28, 2016.

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

[19] Bouaziz, S.; Wang, Y. G.; Pauly, M. Online modeling for realtime facial animation. *ACM Transactions on Graphics* Vol. 32, No. 4, Article No. 40, 2013.

[20] Vallet, B.; Lévy, B. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum* Vol. 27, No. 2, 251–260, 2008.

[21] Wen, Q.; Xu, F.; Lu, M.; Yong, J. H. Real-time 3D eyelids tracking from semantic edges. *ACM Transactions on Graphics* Vol. 36, No. 6, Article No. 193, 2017.

[22] Garrido, P.; Zollhöfer, M.; Wu, C. L.; Bradley, D.; Pérez, P.; Beeler, T.; Theobalt, C. Corrective 3D reconstruction of lips from monocular video. *ACM Transactions on Graphics* Vol. 35, No. 6, Article No. 219, 2016.

[23] Dinev, D.; Beeler, T.; Bradley, D.; Bächer, M.; Xu, H.; Kavan, L. User-guided lip correction for facial performance capture. *Computer Graphics Forum* Vol. 37, No. 8, 93–101, 2018.

[24] Feng, Z. H.; Huber, P.; Kittler, J.; Hancock, P.; Wu, X. J.; Zhao, Q. J.; Koppen, P.; Raetsch, M. Evaluation of dense 3D reconstruction from 2D face images in the wild. In: Proceedings of the 13th IEEE International Conference on Automatic Face & Gesture Recognition, 780–786, 2018.

[25] Yin, L. J.; Wei, X. Z.; Sun, Y.; Wang, J.; Rosato, M. J. A 3D facial expression database for facial behavior research. In: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, 211–216, 2006.

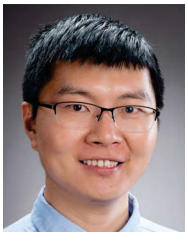
[26] Thies, J.; Zollhöfer, M.; Stamminger, M.; Theobalt, C.; Nießner, M. Face2Face: Real-time face capture and reenactment of RGB videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2387–2395, 2016.

[27] Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. PAMI-8, No. 6, 679–698, 1986.

Yanlong Tang is currently a researcher at Tencent. He obtained his Ph.D. degree in 2019 from Zhejiang University, and his B.Sc. degree from Shandong University in 2013. His research interests include 3D face reconstruction, image processing, and computer vision.

Yun Zhang is an associate professor at Zhejiang Communication University. He received his doctoral degree from Zhejiang University in 2013, and bachelor and master degrees from Hangzhou Dianzi University in 2006 and 2009, respectively. In 2018, he was a visiting scholar at Cardiff University. His research interests include computer graphics, image and video editing, computer vision, and virtual reality. He is a member of the CCF.

Xiaoguang Han received his B.Sc. degree in mathematics in 2009 from Nanjing University of Aeronautics and Astronatics and his M.Sc. degree in applied mathematics in 2011 from Zhejiang University. He obtained his Ph.D. degree in 2017 from the University of Hong Kong. He is currently an 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.



Fang-Lue Zhang is currently a lecturer at Victoria University of Wellington, New Zealand. He received his bachelor degree from Zhejiang University in 2009, and his doctoral degree from Tsinghua University in 2015. His research interests include image and video editing, computer vision, and computer graphics. He is a member of the IEEE and ACM. He received a Victoria Early-Career Research Excellence Award in 2019 and a Marsden Fast-Start grant from the New Zealand Royal Society in 2021.

Yu-Kun Lai received his bachelor degree and Ph.D. degree in computer science from Tsinghua University in 2003 and 2008, respectively. He is currently a professor in the School of Computer Science & Informatics, Cardiff University. His research interests include computer graphics, geometry processing, image processing, and computer vision. He is on the editorial boards of *Computer Graphics Forum* and *The Visual Computer*.

Ruofeng Tong is a professor in the Department of Computer Science, Zhejiang University. He received his B.Sc. degree 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.