Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Article Type (Research/Review)

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], Ruofeng Tong[6]

Abstract There has been a steadily growing range of applications that can benefit from the facial reconstruction techniques, which brings higher demand for reconstruction of high-quality 3D face models. As an important expressive part of the human face, nose receives less attention than other expressive regions in the literature of face reconstruction. When applying existing reconstruction methods on facial images, the reconstructed nose models are 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 we then heuristically update 3D-2D dense 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 in the 3D nose reconstruction than the current state-of-the-art methods.

Keywords 3D Face, Nose Correction, Single Image.

Fig. 1 First row: input images. Second row: baseline 3D face reconstruction without nose correction [20]. Third row: our 3D face reconstruction with nose correction.

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 solve the ambiguities when building the 3D geometry of faces, accurately reconstructing the face model from single 2D images still remains challenging. 3D Morphable Model (3DMM)-based fitting techniques are normally used when we only have access to one facial image, which allows us to make the reconstructed 3D face mesh match the 2D contours in a facial image, such as the contours of face, eyes and nose. In the applications with dynamic facial models, such as facial motion re-targeting, researchers mainly focus on the reconstruction quality of the parts with frequent movements, like the eyes and mouth. But

1 Tencent, Shenzhen, China.yanlongtang@gmail.com
2 Communication University of Zhejiang, Hangzhou, China. zhangyun_zju@zju.edu.cn
3 Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong (Shenzhen), Shenzhen, China. hanxiaoguang@cuhk.edu.cn
4 Victoria University of Wellington, Wellington, New Zealand. fanglue.zhang@ecs.vuw.ac.nz
5 Cardiff University, Wales, UK. Yukun.Lai@cs.cardiff.ac.uk
6 Zhejiang University, Hangzhou, China. trf@zju.edu.cn
Manuscript received: 2014-12-31; accepted: 2015-01-30.

little attention has been paid to the nose. However, with the steadily growing range of applications that can benefit from the face reconstruction techniques, the demand for accurate reconstruction of nose shapes is increasing. For example, post face re-lighting requires a precise nose shape to produce a natural lighting effect in the area surrounding the nose; when creating virtual avatar 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 the recognition accuracy [17, 25], which could be applied 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 representative power; on the other hand, more importantly, it is more difficult to establish sufficient feature constraints in the nose region than 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 [12, 14, 15]. However, they focus on only correcting the shape of the whole face instead of the nose, and their sparse landmarks and dense pixels are not semantically informative enough to represent various nose shapes. Recently, Tang et al. [20] 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 second row of Figure 1. While in the nose region, the curves can be very complex and diverse due to various shapes and perspectives, leading to an erroneous match 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 color similarity with its neighboring regions, i.e., face and nose are both with the color of 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 shapes of nose 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 of 3D-2D sparse landmark correspondences, and then heuristically update 3D-2D dense curve correspondences 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.

There are three problems to be solved for effectively updating 3D-2D dense curve correspondence: 1) How to determine the 3D nose contour due to the self-occlusion and the variance of nose shapes and poses. 2) How to extract a precise 2D nose contour with the non-salient curve features of the boundary of the nose region. 3) How to establish accurate correspondences between the 3D and 2D contours of nose. In terms of extraction of 3D contours, Tang et al. [20] used predefined vertex indices on a template mesh as a fixed 3D nose contour, but this method is not flexible for various nose shapes and poses. Instead, we render the sparsely corrected nose to 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. [20] applied Snake [13] on a feature map, but the curve feature here is not distinctive enough. We produce an enhanced feature map using RGB-D foreground enhancement method [21], where we render a depth map using the sparsely corrected 3D face mesh. Then Snake is able to extract a more accurate 2D contour. In terms of the estimation of 3D-2D contour correspondences, we integrate 3D contour information to 2D contour extraction, rather than dealing with them separately as in [20]. Specifically, we initialize the active contour in Snake algorithm using the projection 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 [20] may produce erroneous correspondences when the curve is complex.

Our work is the first attempt to reconstruct accurate 3D noses from single images to our knowledge. Experiments show that our method outperforms the state-of-the-art methods. We have the following technical contributions:

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

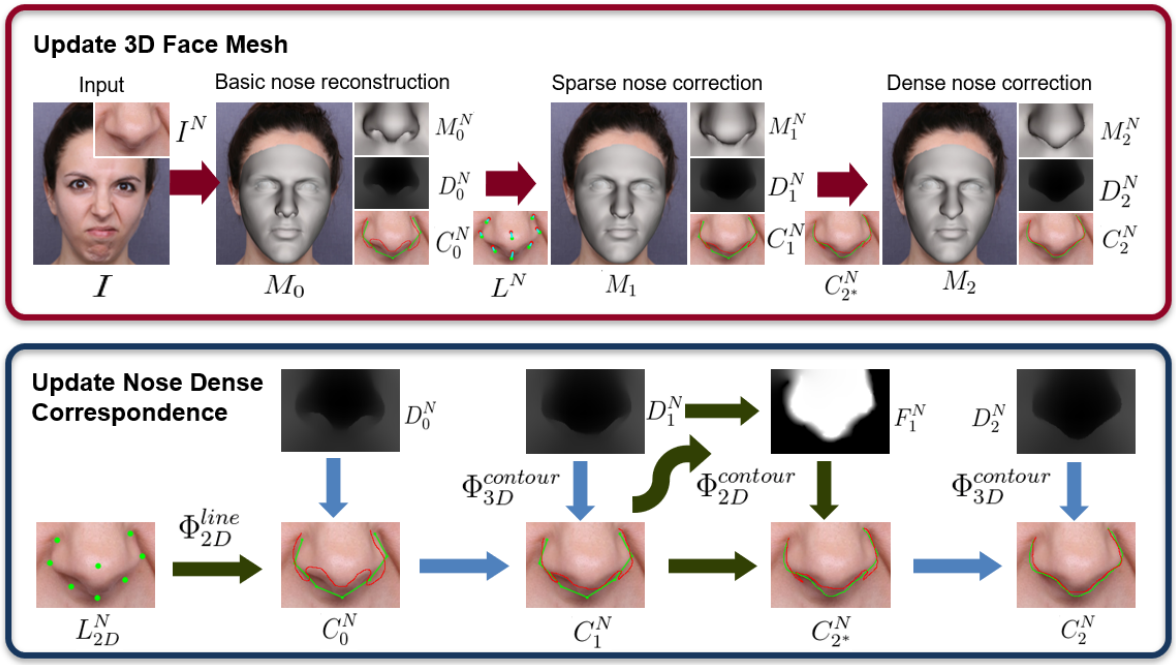


Fig. 2 Pipeline of proposed 3D corrective nose reconstruction method.

2 Related Work

Low-dimensional parametric 3D face models [1, 2, 5, 6, 8, 16, 18, 27] are widely used in the 3D face reconstruction task for their simplicity, compactness and effectiveness. However, limited by the wide range of types of models and their formats in the model databases, low-dimensional models cannot be used to reconstruct sufficiently accurate face shapes, especially when the face is greatly different 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.

There has been numerous studies [10, 14, 15] to investigate how to use Laplacian deformation [19] to correct low dimension 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 coordinates regulation term. Li et al. [14] 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.[15] approximately converted detected 2D sparse landmarks to the 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 correction effect is thus limited. For video input, Garrido et al. [10] corrected the whole face based on dense optical flow constraints. But the optical flow calculation depends on the video input and is not applicable for the single image input. Considering that the high dimensional Laplacian deformation [19] in vertex space is of high computational cost and not robust to noise, some researchers have also suggested [3, 10, 12] solving Laplacian deformation in a low-dimensional subspace [23] to speed up the computation and/or reduce noise. Similar to the work of Li et al. [14], Bouaziz et al. [3] corrected meshes relying on depth data, which is not applicable to a single image either. For single image input, a series of recent studies has indicated that the deformation problem can be solved by utilizing the dense pixels difference between the rendered image and input image [10, 12]. However, it needs to solve parametric albedo and illumination model at the same time, thus is also greatly affected by the representation power of parametric illumination and albedo model. Pixel level dense constraint (depth or photo pixel) is usually a supplement to sparse landmark constraints, and is especially suitable to represent medium level wrinkle deformations in skin regions, such as forehead and cheek, where sparse landmarks constraints cannot model them well. On the other hand, pixel level dense constraint usually contains a lot of noise and does not show salient contour-level semantic features, thus cannot correct feature regions properly. In addition, although low dimensional subspace Laplacian deformation [23] is more efficient and smooth, the deformation is limited to a narrow range.

The above works aim at correcting the whole face to fit the input sparse or dense data. However, local feature regions in

their reconstructed results, such as eyes, mouth and nose, are still not identifiable or expressive enough. Compared with sparse landmarks and dense pixel feature, contour features contain more semantic information to model parts of the face better, thus can be used to further correct local shapes. For eyelid correction, Wen et al. [24] build 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. [11] learned a mapping from inaccurate 3D lips to accurate 3D lips. But the accurate 3D lips data set needs to be collected and processed by complex and expensive equipment, and they also require manually labeled data to train the 2D lip contour extraction model. Dinev et al. [7] also corrected lips using a data-driven method. Different from [11], they constructed a training data set using lightweight Laplacian deformation techniques [19]. However, they need to manually extract 2D lip contour, and sometimes they need to heuristically label lips due to the occlusion between upper and lower lips. All the above correction methods involving some manual intervention in 2D contour extraction. Thus, more lightweight and fully automatic 2D contour extraction methods are preferable to reduce manual burden. More recently, Tang et al. [20] propose a lightweight 2D contour extraction approach to correct local facial features. When extracting 2D contour, they propose a local-to-global snake algorithm [13] 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 its more complex shape.

To the best of our knowledge, there are no previous works aiming at correct nose reconstruction in the field of single-image-based facial reconstruction. Compared with eye and lip correction [7, 11, 24], it is more challenging to establish accurate 3D-2D dense contour correspondence for nose correction. To deal with this challenge, we couple the 3D reconstruction and 2D feature extraction instead of dealing with them separately [7, 11, 24], which effectively improves dense 3D-2D nose correspondence. In our approach, in order to allow a flexible 3D nose contour for various face poses and nose shapes, we heuristically refine the 3D nose contour in a coarse-to-fine scheme in reconstruction. To mitigate the ambiguity when extracting 2D nose contour with 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 [11, 20, 24]. For 3D-2D one-to-one contour correspondence, considering that 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 the snake algorithm [13].

3 Method

3.1 Overview

Previously, single image based 3D face reconstruction commonly faced the difficulty of reconstructing accurate and identifiable 3D nose shape. 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 Figure 2. The key challenge in 3D nose reconstruction is to establish sufficiently accurate 3D-2D feature correspondences that can adapt to various 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 the 3D nose shape change. Then, the 3D nose shape is iteratively refined based on the updated nose correspondences. Overall, the process is composed of three stages: basic nose reconstruction, sparse nose correction and dense nose correction.

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

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

where the targets to be solved include camera parameters P, 3D face mesh M (nose part is M^N), and 3D-2D nose correspondence C^N. $C^N = (C^{N,2D}, C^{N,3D})$ contains one-to-one nose correspondence between 2D point set $C^{N,2D}$ and 3D mesh vertex set $C^{N,3D}$. In each reconstruction stage, only a single energy term in Equation 1 is activated. **(1) Basic Nose Reconstruction Stage**. In this stage, an initial 3D nose is reconstructed with energy weights as $\sigma_0 = 1.0$, $\sigma_1 = 0.0$, $\sigma_2 = 0.0$. The optimization objective is $E(P,M,C^N) = E_{basic}(P,M,C^N)[L^A, C_0^{A-N}]$ [20], where camera parameters P, whole face mesh M and nose correspondence C^N are all solved based on all 3D-2D sparse correspondence $L^A = (L^{A,2D}, L^{A,3D})$ and partial 3D-2D dense correspondence $C_0^{A-N} = (C_0^{A-N,2D}, C_0^{A-N,3D})$ (excluding nose dense correspondence, as it is not accurate yet). This stage outputs the basic 3D nose shape M_0^N and 3D-2D nose dense correspondence C_0^N. **(2) Sparse Nose Correction Stage**. In this stage, we refine the results of first stage with energy weights as $\sigma_0 = 0.0$, $\sigma_1 = 1.0$, $\sigma_2 = 0.0$. The optimization is formulated as $E(P,M,C^N) =$

Tab. 1 Mathematical notations used in this paper.

Notations	Description
Camera	
P	camera parameters, include $P = \{Pr, R, t\}$
Pr	weak perspective projection matrix
R	rotation matrix
t	translation vector
Π	get projected 3D point in image space
Π_{xy}	get 2D position (xy component) of projected 3D point
Π_z	get depth value (z component) of projected 3D point
Image	
I	input face image
I^N	nose region of input face image
F_1^N	enhanced nose feature map in optimization stage 1
Mesh	
M	target 3D face mesh to be solved
M_i	solved 3D face mesh of optimization stage i
M_i^N	nose region of solved 3D face mesh of optimization stage i, 'N' indicate 'Nose'
D_i^N	rendered nose depth map of 3D face mesh in optimization stage i, 'N' indicate 'Nose'
Corres.	
L^A	all 3D-2D sparse landmarks correspondence, 'A' indicate 'All'
L^N	nose 3D-2D sparse landmarks correspondence, 'N' indicate 'Nose'
C^N	target 3D-2D dense nose correspondence to be solved, 'N' indicate 'Nose'
C_i^N	3D-2D dense nose correspondence result of stage i , 'N' indicate 'Nose'
C_i^A	3D-2D dense face correspondence result of stage i, 'A' indicate 'All'
C_i^{A-N}	3D-2D dense face (excluding nose) correspondence result of stage i, 'A-N' indicate 'All' face dense correspondence excluding 'Nose' part
Operation	
Φ_{2D}^{line}	generate 2D nose contour by connecting landmarks
$\Phi_{2D}^{contour}$	update 2D nose contour using snake
$\Phi_{3D}^{contour}$	extract 3D nose contour from depth map
$\mathcal{F}$	get enhanced feature map using RGB-D image

$E_{sparse}(M^N, C^N)[L^N]$, where camera parameters P are fixed, and only 3D nose shape M^N and nose correspondence C^N are solved. The constraint is nose 3D-2D sparse correspondence $L^N = (L^{N,2D}, L^{N,3D})$. 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 settings $\sigma_0 = 0.0$, $\sigma_1 = 0.0$, $\sigma_2 = 1.0$. And the optimization becomes $E(P, M, C^N) = E_{dense}(M^N, C^N)[C_{2*}^N]$, where we first update nose correspondence from C_1^N to C_{2*}^N as energy constraints, and then solve the 3D nose shape M^N and update the nose correspondence C^N, and get the final results M_2^N and C_2^N.

Face Model. We propose 3D face model [27] for reconstruction. In the model, a 3D face mesh can be represented in two forms: high dimensional space and low dimensional space. In high dimensional space, a 3D face is represented by all the vertices, while in the low dimensional space, it can be represented by a small number of parameters. In the basic nose reconstruction stage, 3D face mesh M is first obtained in low dimensional space, which is represented by the following set of parameters: $M(\alpha, \beta) = M_{mean} + B_{id} \cdot \alpha + B_{exp} \cdot \beta$ [27], where α and β represent identity and expression parameters respectively. In all the three stages, 3D face mesh is also corrected in high dimensional space. The face mesh is represented in the form of high-dimensional vector containing positions of vertices: $M(V) = \{v_i\}_{i=1}^n$, where v_i represents the 3D position of the i-th vertex.

Camera Model. In the basic nose reconstruction stage, camera parameters P are solved, and in the next two stages, P is fixed and is used to inversely project 2D points of image space to 3D space. It can be represented as $P = \{Pr, R, t\}$, 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:

$$v^{proj} = \Pi(v_{3d}), \tag{2}$$

which can be further expanded as:

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

where $v^{proj} = \begin{pmatrix} v_{2d} \\ d \end{pmatrix}$ 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 3D rotation matrix and t is 3D translation. For convenience,

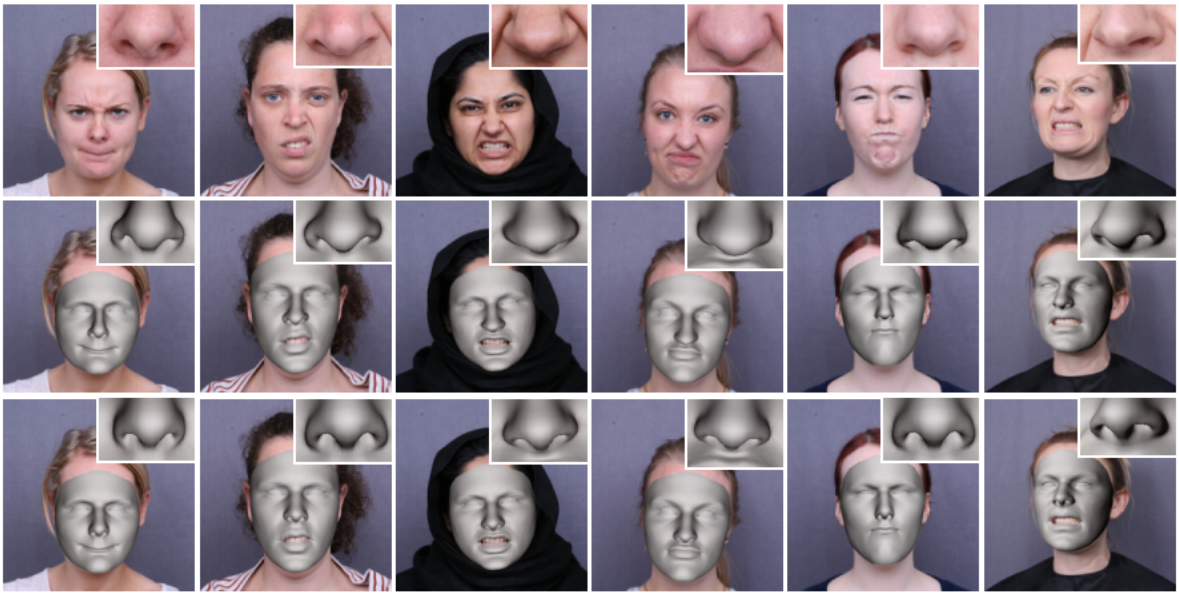


Fig. 3 Comparison with the state-of-the-art method on Stirling ESRC dataset [9]. First row: input images. Second row: results of our method. Third row: results of method [20]

we decompose the 3D projection formula as:

$$v_{2d} = \Pi_{xy}(v_{3d}), \tag{4}$$

and

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

On the other hand, 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:

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

which can be further expanded as:

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

3.2 Basic Nose Reconstruction

Recent work [20] proposed a 3D facial reconstruction based on dense contour features, which can produce faithfully reconstructed 3D faces especially for exaggerated faces. Such a method of establishing 3D-2D dense contour correspondence 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 [20] for initialization, and facial regions except nose are corrected. The optimization objective of the initial nose reconstruction is formulated as:

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

where P, M and C^N are camera parameters, objective 3D face mesh and nose correspondence respectively as in Equation 1. ω_i is the weight of each energy term. $E_{sparse}^{fit}[L^A]$ is the low-dimensional fitting energy using all sparse landmarks L^A as constraints. $E_{dense}^{fit}[C_0^{A-N}]$ indicates the low-dimensional fitting energy using all dense contours except for the nose contour C_0^{A-N} as constraints. E_{reg}^{fit} is the low-dimensional regulation energy which keeps the parameters in a reasonable range. $E_{dense}^{correct}[C_0^{A-N}]$ 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 according to the work [20]. In the first stage, we estimate a 3D mesh in a low dimensional space with sparse constraints, and the energy weights are $\omega_1 = 1.0$, $\omega_2 = 0.0$, $\omega_3 = 0.05$, 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 proposed based on dense constraints. Energy weights are $\omega_1 = 0.0$, $\omega_2 = 0.0$, $\omega_3 = 0.0$, and $\omega_4 = 1.0$.

Our initial results show that except for the nose region, the other regions can 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:

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

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

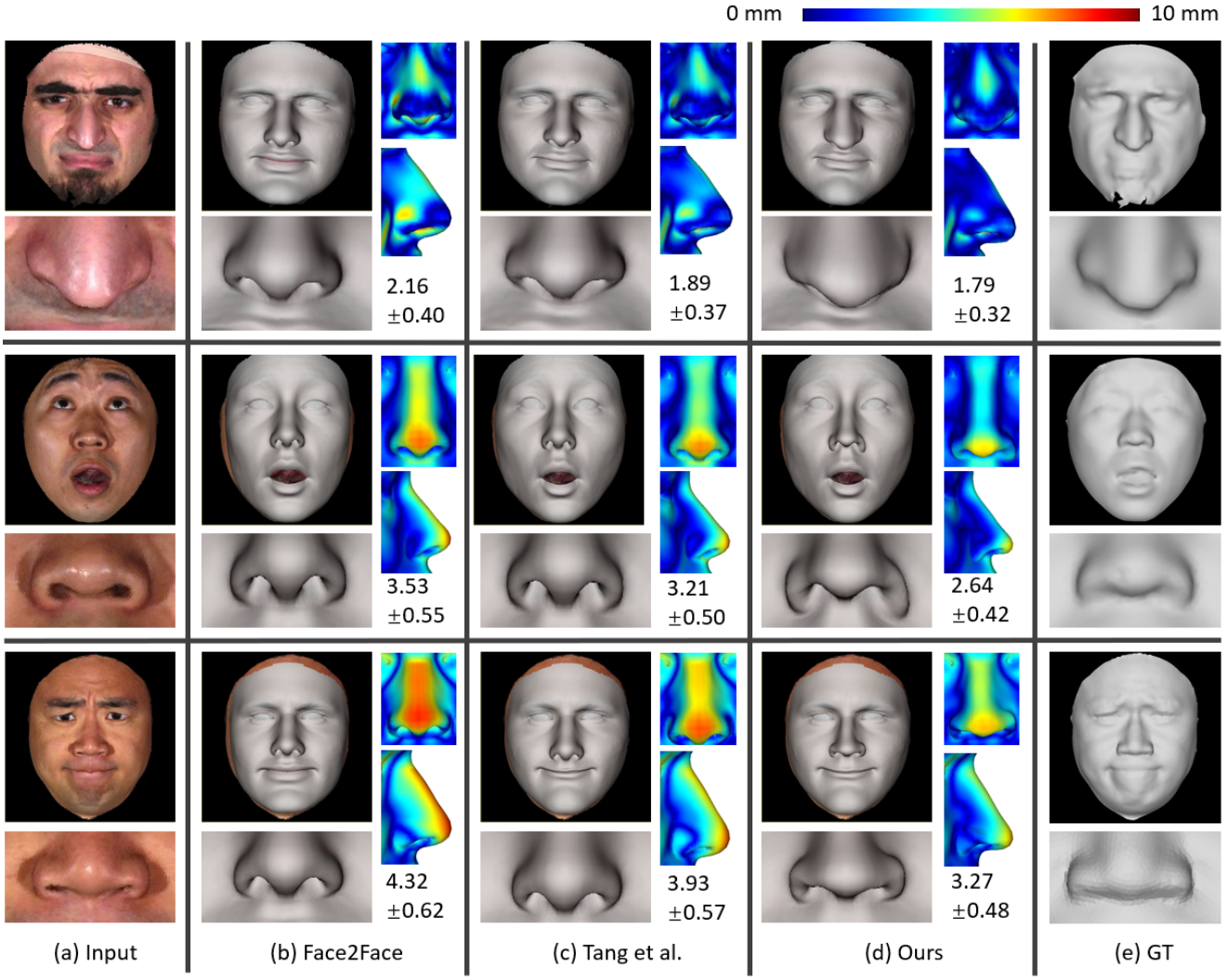


Fig. 4 Comparison with state-of-the-art optimization based methods on BU-3DFE dataset [26]. (a) input images; (b) results of Face2Face [22]; (c) results of Tang et al. [20]; (d) results of our method; (e) ground truth 3D meshes. The reconstructed error (unit is millimeter) can be visualized in red/blue color maps, and the Root Mean Squared Error (RMSE) and Stanard Deviation are shown below the color maps.

landmarks $L^{N,2D}$ with straight lines. $C_0^{N,3D} = \Phi_{3D}^{contour}(D_0^N)$ represents the 3D nose contour, extracted from rendered nose depth map D_0^N. The nose depth map D_0^N is rendered from reconstructed nose region mesh M_0^N. In D_0^N, pixels belonging to nose regions are set as white and non-nose pixels are set as 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 searching, which results in the initial 3D nose contour $C_0^{N,3D}$.

3.3 Sparse Nose Correction

The nose shape reconstructed from [20] has a largely different look with ground truth. However, as stated before, dense nose 3D-2D contour correspondence cannot be directly generated like eyes and lips due to the difficulties of extracting both 2D and 3D nose contours. It is observed that although sparse nose landmarks are not sufficient to describe a nose shape, they usually can be accurately detected. Based on the observation, weak nose correction [19] is performed using the sparse nose landmarks, thus the reconstructed 3D nose shape can 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:

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

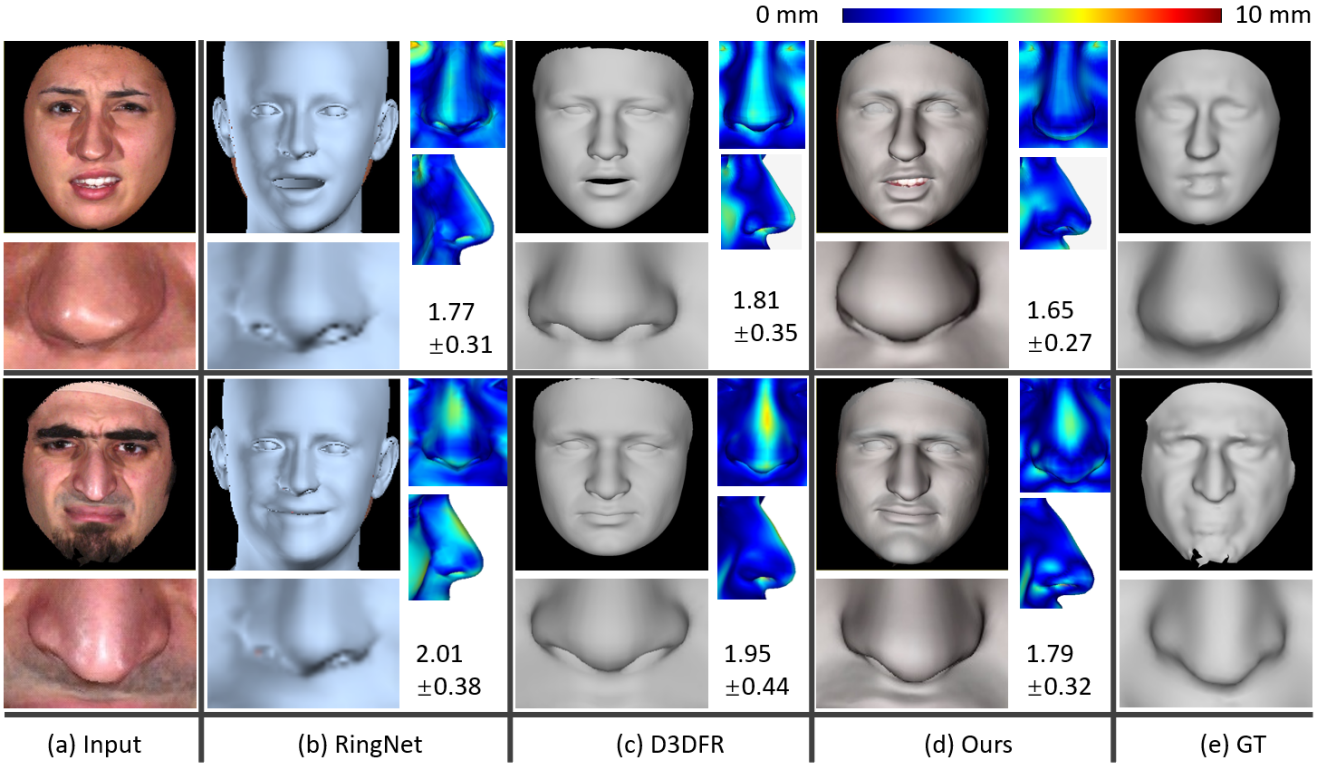


Fig. 5 Comparison with state-of-the-art learning based methods on BU-3DFE dataset [26]. (a) input images; (b) results of RingNet [18]; (c) results of D3DFR [6]; (d) results of our method; (e) ground truth 3D meshes. The reconstructed error (unit is millimeter) can be visualized in red/blue color maps, and the Root Mean Squared Error (RMSE) and Stanard Deviation are shown below the color maps.

where M^N is the nose mesh represented by its vertices. L^N is sparse landmark correspondence as optimization constraints. $\pounds$ is the Laplacian operator [19]. ω is a weight to balance the landmark matching term and the Laplacian term, with an experimentally determined value 5.0. With the inverse projection Equation 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 a better dense 3D-2D nose correspondence. The nose correspondence is updated by:

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

where C_1^N is the updated nose dense correspondence in stage of sparse nose correction. $C_1^{N,2D} = C_0^{N,2D}$ 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 sparsely corrected nose result D_1^N.

3.4 Dense Nose Correction

After sparse nose correction, the 3D nose shape gets closer to 2D input, but the quality of the result is not sufficiently high to be used in personalized applications. Thus, we further perform dense nose correction to get accurate dense 3D-2D nose contour correspondence.

Update dense nose correspondence. In the previous sparse correction stage, 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 [4] to detect 2D facial contours. Their resulting contours may be noisy or jaggy due to the lack of shape prior. Thus, we deal with this problem by employing the snake algorithm [13] which can combine both low-level image features and high-level shape prior. On the one hand, 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 edge and brightness. On the other hand, the internal regular energy term can preserve the contour shape and smoothness. The snake-based 2D contour updating can be formulated as:

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

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

Previous work [20] also employed snakes to extract the

facial contour. In their work, the initial contour is composed of the 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 not applicable to extracting nose contour. Different from eyes and lips, edge features are not salient in nose regions because the skin colors are similar between a nose and its neighboring region. We thus generate an enhanced feature map F using the RGB-D saliency detection method in [21], where the depth map D_1^N is rendered from reconstructed 3D face mesh. Furthermore, as the shape of nose is more complex than eyes and lips, ICP method used in work [20] may result in wrong 3D-2D correspondences. We instead set the initial contour C_{init} as the 2D projection of 3D nose contour $\Pi_{xy}(C_1^N)$, which can implicitly establish accurate 3D-2D correspondence in an adaptive way. The above dense nose 3D-2D correspondence updating process can be formulated as:

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

where $C_{2^*}^N$ is the updated dense correspondence, $C_{2^*}^{N,3D} = C_1^{N,3D}$ represents the 3D nose contour in the previous sparse correction stage. $C_{2^*}^{N,2D} = \Phi_{2D}^{contour}(\Pi_{xy}(C_1^{N,3D}), F_1^N)$ indicates the updated 2D nose contour based on the snake method (Equation 13). In the 2D nose updating, the initial nose contour $\Pi_{xy}(C_1^{N,3D})$ is the 2D projection of 3D nose contour $C_1^{N,3D}$, which can implicitly preserves the 3D-2D correspondence when 2D contour deforms. The feature map F_1^N used for the snake algorithm is an enhanced feature map which is generated by the RGB-D saliency detection method [21]. $F_1^N = \mathcal{F}(I^N, D_1^N)$ represents the feature map that is calculated based on the RGB image I^N and the depth map D_1^N of the nose. As both 3D and 2D contours are evolved from $C_1^{N,3D}$, 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, which thus results in enhanced edges. We modify the original method [21] to better suit our task. Specifically, the random walker seeds for foreground and background are sampled on different sides of the banded area formed by $C_1^{N,2D}$ and $\Pi_{xy}(C_1^{N,3D})$, and we set the random walker weight graph using the depth information for regularization, which can constrain the resulted foreground boundary close to the input nose boundary in the depth map.

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

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

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

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

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

After the dense nose correction, accurate 3D nose shape M_2^N is generated. Similar to Equation 12, the dense correspondence can be further updated by:

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

which is the final output of the dense 3D-2D contour correspondence.

4 Experiment

4.1 Comparison with the state of the art

We compare our method with state-of-the-art image-based 3D face reconstruction method [20] on Stirling ESRC 3D face dataset [9], as shown in Figure 3. The experimental results demonstrate that our method outperforms it by reconstructing better personalized and distinctive nose shapes. Further quantitative comparison with optimization based methods [20, 22] on BU-3DFE dataset [26] is performed, which numerically demonstrate the advantage of our method, as shown in Figure 4. Additionally, we compare our method with recent learning based methods [6, 18], as shown in Figure 5, which also shows the better performance of our method.

4.2 Ablation study

We conduct ablation experiments to demonstrate the roles of all the three stages of our method. The results after each stage are shown in Figure 6. It demonstrates that both sparse and dense correction can significantly improve nose reconstruction. In the first row, nose wings are improved in the final result. In the second row, the overall shape and position of the model are improved. In the third row, final reconstructed results have lower nose tips, which better match the input images.

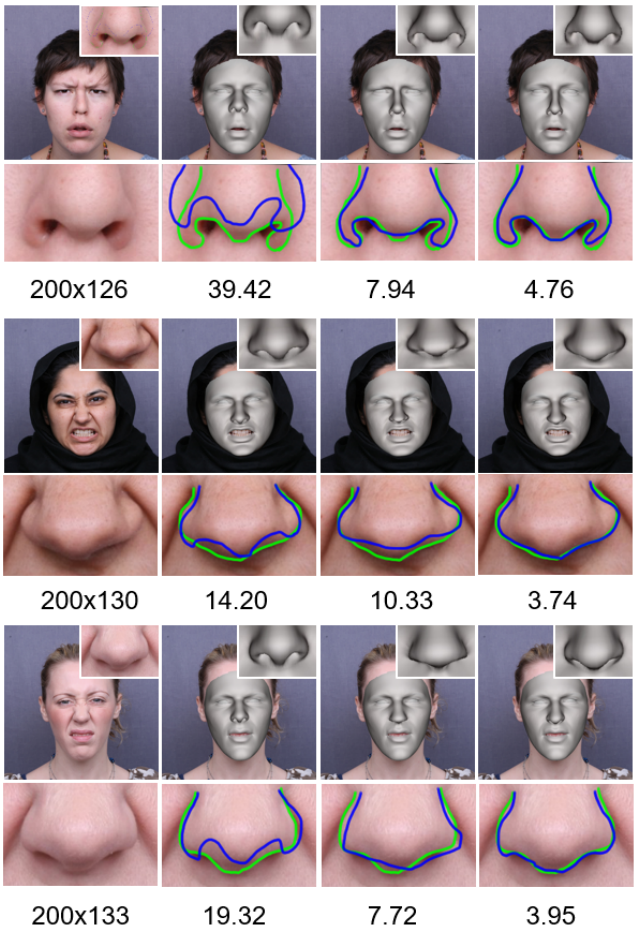


Fig. 6 Results of the ablation study. First column: input images. Second column: results of first stage. Third column: results of second stage. Fourth column: results of third stage (final results of the proposed method). Numbers below the first column are the resolution of nose region, while numbers below the other columns are the mean pixel errors between the reconstructed nose contour (blue) and the ground truth nose contour (green).

4.3 Fixed vs. updated 3D contour

A successful nose correction relies on adequate accurately matched features on the nose region. The 3D nose contour must match the 2D contour, otherwise the reconstructed results can not accurately recover the shape of the nose in the 2D image. Our 3D contour updating scheme is designed for that aim. In Figure 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 can get much better results.

4.4 Discussion on the 2D contour updating

The traditional Snake method is to update the 2D contour using the intensity feature map of the image. However, the feature on the intensity map is not significant, which often leads to an undesirable nose boundaries. Our enhanced feature map generated from RGB-D data is designed to

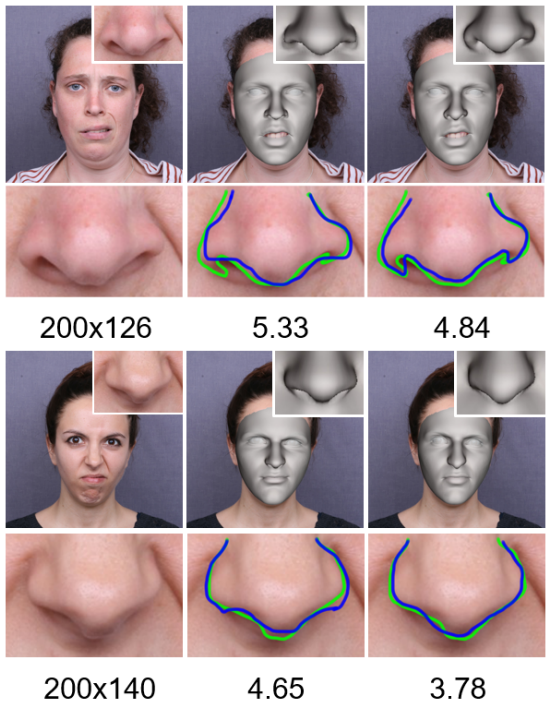


Fig. 7 Comparative results for 3D nose contour updating. First column: input images. Second column: results without 3D contour updating. Third column: results with 3D contour updating. Numbers below the first column are the resolution of nose region, while numbers below the other columns are the mean pixel errors between the reconstructed nose contour(blue) and ground truth nose contour (green).

cope with this problem. In Figure 8, we compare the results based on feature maps generated from the intensity map, the RGB saliency map and the RGB-D saliency map respectively. The experimental results show that the RGB-D saliency map significantly improves the quality of 2D contour and further improves the quality of nose correction. The 3D nose head shape generated by the proposed method is more approximate to hook nose that better matches the input image.

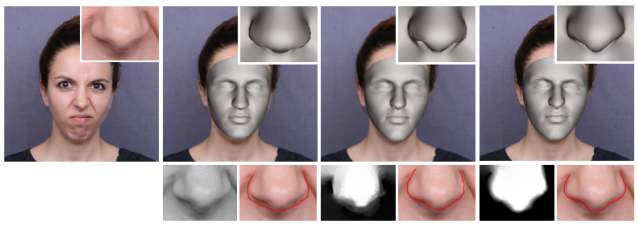


Fig. 8 Demonstration of the effectiveness of 2D contour update. First column: input images. Second column: results on intensity map. Third column: results on RGB saliency map. Fourth column: results on RGB-D saliency map.

5 Conclusion

In this paper, we propose a 3D nose reconstruction method which adaptively updates the nose model to

better match the input 2D facial image. Our method utilizes a coarse-to-fine 3D nose corrective reconstruction approach, which can adaptively and heuristically build and update dense 3D-2D nose contour correspondences to adapt to different face poses and nose shapes. We also improve 2D nose contour detection using the enhanced feature map generated from RGB-D data that is rendered using intermediate nose model. The experiments show our advantage over the current state-of-the-art facial reconstruction method in terms of the quality of reconstructed noses.

Acknowledgements

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

References

[1] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

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

[3] S. Bouaziz, Y. Wang, and M. Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics (TOG)*, 32(4):40, 2013.

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[5] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2014.

[6] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, and X. Tong. 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*, pages 0–0, 2019.

[7] D. Dinev, T. Beeler, D. Bradley, M. Bächer, H. Xu, and L. Kavan. User-guided lip correction for facial performance capture. In *Computer Graphics Forum*, volume 37, pages 93–101. Wiley Online Library, 2018.

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

[9] Z.-H. Feng, P. Huber, J. Kittler, P. Hancock, X.-J. Wu, Q. Zhao, P. Koppen, and M. Rätsch. Evaluation of dense 3d reconstruction from 2d face images in the wild. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 780–786. IEEE, 2018.

[10] P. Garrido, M. Zollhöfer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, and C. Theobalt. Reconstruction of personalized 3d face rigs from monocular video. *ACM Transactions on Graphics (TOG)*, 35(3):28, 2016.

[11] P. Garrido, M. Zollhöfer, C. Wu, D. Bradley, P. Pérez, T. Beeler, and C. Theobalt. Corrective 3d reconstruction of lips from monocular video. *ACM Trans. Graph.*, 35(6):219–1, 2016.

[12] L. Jiang, J. Zhang, B. Deng, H. Li, and L. Liu. 3d face reconstruction with geometry details from a single image. *arXiv preprint arXiv:1702.05619*, 2017.

[13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[14] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.*, 32(4):42–1, 2013.

[15] Y. Li, L. Ma, H. Fan, and K. Mitchell. Feature-preserving detailed 3d face reconstruction from a single image. In *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production*, pages 1:1–1:9, New York, NY, USA, 2018. ACM.

[16] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*, pages 296–301. Ieee, 2009.

[17] M. D. Samad and K. M. Iftekharuddin. Frenet frame-based generalized space curve representation for pose-invariant classification and recognition of 3-d face. *IEEE Transactions on Human-Machine Systems*, 46(4):522–533, 2016.

[18] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black. 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*, pages 7763–7772, 2019.

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

[20] Y. Tang, X. Han, Y. Li, L. Ma, and R. Tong. Expressive facial style transfer for personalized memes mimic. *The Visual Computer*, pages 1–13, 2019.

[21] Y. Tang, R. Tong, M. Tang, and Y. Zhang. Depth incorporating with color improves salient object detection. *The Visual Computer*, 32(1):111–121, 2016.

[22] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[23] B. Vallet and B. Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Wiley Online Library, 2008.

[24] Q. Wen, F. Xu, M. Lu, and J.-H. Yong. Real-time 3d eyelids tracking from semantic edges. *ACM Transactions on Graphics (TOG)*, 36(6):193, 2017.

[25] N. Werghi, C. Tortorici, S. Berretti, and A. Del Bimbo. Boosting 3d lbp-based face recognition by fusing shape and texture descriptors on the mesh. *IEEE Transactions on Information Forensics and Security*, 11(5):964–979, 2016.

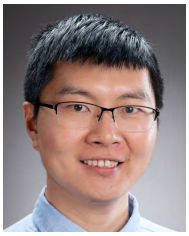
[26] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 211–216. IEEE, 2006.

[27] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155, 2016.

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

Yun Zhang Yun Zhang is an associate professor at Communication University of Zhejiang. He received his doctoral degree from Zhejiang University in 2013. Before that, he received Bachelor and Master degrees from Hangzhou Dianzi University in 2006 and 2009, respectively. From Feb. to Aug. 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 CCF.

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 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 Fang-Lue Zhang is currently a Lecturer with Victoria University of Wellington, Wellington, New Zealand. He received the Bachelor's degree from Zhejiang University in 2009, and the 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 IEEE and ACM. He received Victoria Early-Career Research Excellence Award in 2019 and Marsden Fast-Start grant from New Zealand Royal Society in 2021.

Yu-Kun Lai Yu-Kun Lai received his bachelor's degree and PhD 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 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.