Pose-Free Face Swapping Based on a Deformable 3D Shape Morphable Model

Yuan LIN†(a), Nonmember and Shengjin WANG†(b), Member

SUMMARY Traditional face swapping technologies require that the faces of source images and target images have similar pose and appearance (usually frontal). For overcoming this limit in applications this paper presents a pose-free face swapping method based on personalized 3D face modeling. By using a deformable 3D shape morphable model, a photo-realistic 3D face is reconstructed from a single frontal view image. With the aid of the generated 3D face, a virtual source image of the person with the same pose as the target face can be rendered, which is used as a source image for face swapping. To solve the problem of illumination difference between the target face and the source face, a color transfer merging method is proposed. It outperforms the original color transfer method in dealing with the illumination gap problem. An experiment shows that the proposed face reconstruction method is fast and efficient. In addition, we have conducted experiments of face swapping in a variety of scenarios such as children’s story book, role play, and face de-identification striping facial information used for identification, and promising results have been obtained.

key words: 3D face reconstruction, face swapping, color transfer, multi-resolution spline, human-computer interaction

1. Introduction

Avatar technology has increasingly appealing applications in virtual worlds, online makeover, virtual video conference and face recognition. In various usage scenarios or situations involving security concerns, consumers may require face recognition. In various usage scenarios or situations involving security concerns, consumers may require face recognition. In various usage scenarios or situations involving security concerns, consumers may require face recognition. In various usage scenarios or situations involving security concerns, consumers may require face recognition. In various usage scenarios or situations involving security concerns, consumers may require face recognition. Avatars can be used for identification, and promising results have been obtained.

Aiming to reach a balance between the reconstruction accuracy and the reconstruction reality [7] by taking it as an optimization problem [9], [10]. Accuracy is the similarity between the projection of the reconstructed face and the input image. Reality is the plausibility of being faces, which is quantified with the probability distribution of the shape and texture coefficients of the adopted 3D database. The optimization can generate realistic faces, but is likely to fall into local minima and is quite time-consuming. The well-known statistical morphable model, 3D morphable model (3DMM) [7], requires around 4.5 min to reconstruct one 3D face on a 2 GHz Pentium IV according to Levine’s survey [8].

Shape deformation of a generic face [11]–[14] is very popular for its efficiency. At first, the face is detected with skin color [11] or a boost cascade detector [15]. Then the facial feature points are extracted and new coordinates of the feature vertices on the face model are computed [12]. After that, the displacements of the non-feature vertices are interpolated with the displacements of the feature vertices. There are different interpolation methods, such as radial basis function (RBF) [12], Dirichlet free-form deformations (DFFD) [13] and morphing by control lines [14]. Compared with morphable models, these reconstruction methods have analytic solutions, which make them very fast.

Aiming to reach a balance between the reconstruction accuracy and the computational efficiency, in this paper a deformable 3D shape morphable model combining the advantages of 3DMM and shape deformation is proposed for 3D face reconstruction. Compared with 3DMM, our core idea is to sacrifice some reconstruction accuracy to accelerate the reconstruction process. The MATLAB implementation of the algorithm takes about 2.8 s on a 3.2 GHz Intel Core i5 processor and the results are quite photo-realistic. Aided by the generated 3D faces, face swapping [16], [17] and face augmented makeover [18] can be accomplished as
Generally, face swapping replaces a face/faces in the target image with the face in the source image. It can be used in many applications, such as children’s role play in story books, face replacement of the stunt man’s face with the user’s face and face de-identification. Traditional 2D image based face swapping requires the similarity of pose and appearance between the target face and the source face. Bitouk et al. [16] proposed a face swapping method based on a large database of face images. For a target face image, the candidate face images with similar pose and appearance were selected from the database. Then, the pose, lighting and color of these candidate face images were adjusted to match the appearance of those in the target image and seamlessly blended in the results. This method generated great results for some images, but they were restricted to frontal poses, which were within ±25° in yaw and ±15° in pitch. This method also highly relied on the coverage of the face database. Besides, the quality of the face swapping result between two particular face images would not be guaranteed, since the pose and appearance restrictions were implicit in the method. Another face swapping method proposed by Zhu et al. [17] used unsupervised face alignment technology. It could generate impressive results when the source face and the target face had similar pose and appearance. However, when there was a noticeable pose variation between inputs, the algorithm often failed. To solve these problems, our proposed 3D face reconstruction method is designed to generate arbitrary viewpoint face images for face swapping in the case of large-view pose differences. Thus, the application of face swapping is not subject to the pose similarity restrictions.

2. Overview of the Proposed System

The proposed system contains two main parts. The first part is to reconstruct a 3D face automatically based on a single frontal view image with a deformable 3D shape morphable model, shown in Fig. 1. The reconstruction focuses on building a high-quality photo-realistic 3D face in an efficient way. First, the facial feature points are extracted from the input image. It contains face detection, face alignment and facial feature point matching three sub steps. Second, one 3D face database, in which all the faces are in full correspondence is adopted. 3D feature vertices of each face are extracted as well. Shape information of these 3D faces is learned with Principal Component Analysis (PCA) to construct a 3D shape morphable model. Third, based on the feature points in the input image and their corresponding feature vertices of the 3D faces, a personalized 3D face is reconstructed with the 3D shape morphable model for the shape initialization. Fourth, shape deformation is applied to this reconstructed 3D face for the shape adjustment to gain the final result. Finally, the texture is generated and mapped onto the final reconstructed 3D face model.

The second part is to apply the reconstructed 3D face to face swapping. In our system, for doing face swapping to a target image the user only needs to upload one frontal view image (source image). At first, a personalized 3D face is reconstructed using the aforementioned method. Once the pose of the target face has been estimated, the virtual source image with the same pose will be generated by rotating and rendering the reconstructed 3D face. Then, both face regions are extracted to gain the facial masks. For each target image, this work only needs to be done once for all the users. The face region on the average 3D face is predefined as well. Thus, the facial mask of the virtual source image can be generated simultaneously when the reconstructed 3D face is rendered. Then the virtual source image, the target image and their facial masks are aligned to a common coordinate system. The overlapping part of the virtual source face region and the target facial region are taken as the merge mask. Based on the two face images and their facial masks, further color adjustment and images blending are used to get the final replacement result. The process is shown in Fig. 2.

3. 3D Personalized Face Reconstruction Based on a Single Frontal View Image

The reconstruction process is performed by combining three algorithms: a random forest embedded active shape model [19] for face alignment, a 3D shape morphable model for 3D shape initialization, and a shape deformation algorithm [12] for further shape adjustment.

In this paper, USF Human ID 3D face database [20] is
adopted. It contains 100 exemplar faces, and all of them are in full correspondence. Figure 3 shows the average face. Forty four 3D feature vertices are predefined on the average face. Then, the feature vertices of each 3D face can be gotten by its corresponding relationship with the average face.

The 3D face reconstruction flowchart is shown in Fig. 1. The succeeding subsections discuss how facial feature points are extracted, and how the 3D personalized face is constructed.

3.1 Face Detection and Face Alignment

Face detection is to locate the face in the image. It provides an initialization for the face alignment. Face detections can be done based on skin color detection[11] or with a boost cascade detector[15]. In this paper, the boost cascade detector[15] is used.

Face alignment is to extract facial feature points, which serve as fundamental reference points demonstrating the subject’s face shape. The accuracy of the face alignment influences the 3D face reconstruction quality directly. Base on this consideration, a random forest embedded active shape model (RFE-ASM)[19] is adopted for its efficiency and accuracy. In this RFE-ASM method, random forest classifiers with pair-compare features are adopted to replace the traditional appearance model in ASM. It avoids the deficiency of local minima in appearance models by using a general statistical model. The pair-compare features are simple and efficient, so facial features localization can process in real time. The overall optimization procedure then incorporates the classifier outputs and shape constraint into a quadratic function which can largely eliminate uncertainty shapes. Some results are shown in Fig. 4. All the images are from the database of [17]. An interface is also provided to modify the alignment result if necessary.

3.2 Feature Point Matching

Feature point matching is to find one corresponding image feature point for each 3D feature vertex, which is used as one important restriction for the 3D face reconstruction later. Many researches[5],[18] match the 3D feature vertices with the face alignment result directly. It means the matching accuracy will highly depend on the face alignment’s accuracy.

We believe that there is not a fixed corresponding relationship between the 3D feature vertices and the face alignment result. Instead, the image feature points are on the corresponding interpolation curves of the face alignment result. One resampling method is proposed for the matching. Firstly, the 3D face shape is projected on the input image plane. Secondly, interpolation is done for both the projection of the 3D feature vertices and the face alignment result. The points of the same part is interpolated to form a successive feature contour. They are named as 3D feature contours and image feature contours respectively. Thirdly, each image feature contour is resampled based on the distribution of the 3D feature vertices on its corresponding 3D feature contour. For a 3D feature vertex v on a 3D feature contour C, the arc length of curve J from the start point of C to v is calculated. Thus, the arc length ratio of J to C can be gotten. Multiplying this ratio by the arc length of its corresponding image feature contour Q, the arc length from the start point of Q to the extracted image feature point is obtained. In this way, the image feature point can be gotten. The matching process is shown in Fig. 5. It presents the face alignment result, the 3D face, the image feature contours, and the face alignment result marked with blue triangles and the final image feature points marked with yellow dots.

3.3 3D Shape Morphable Model

3D morphable model[7] is based on a set of 3D faces, in which all of them are in full correspondences. The face geometry is represented with a shape vector $S = (X_1, Y_1, Z_1, X_2, Y_2, Z_2, \ldots, X_n, Y_n, Z_n) \in \mathbb{R}^{3n}$, contains the $X$, $Y$ and $Z$ coordinates of its $n$ vertices. And the face texture is represented with a texture vector $T = (R_1, G_1, B_1, R_2, G_2, B_2, \ldots, R_n, G_n, B_n) \in \mathbb{R}^{3n}$, that contains the $R$, $G$ and $B$ color values of the $n$ vertices. Based on the averages of shape $\bar{S}$ and texture $\bar{T}$, and the shape and texture covariance matrices, Principal Component Analysis (PCA) performs a basis transformation to an orthogonal coordinate...
system formed by the eigenvectors \( s_i \) and \( t_i \) of the covariance matrices on the database of \( m \) faces.

\[
S = \tilde{S} + \sum_{i=1}^{m-1} \alpha_i s_i, \ T = \tilde{T} + \sum_{i=1}^{m-1} \beta_i t_i,
\]
\[
\tilde{a} = (\alpha_1, \alpha_2, \ldots, \alpha_{m-1})^T, \ \tilde{b} = (\beta_1, \beta_2, \ldots, \beta_{m-1})^T,
\]
(1)

where \( \tilde{a} \in \mathbb{R}^{m-1} \) is the shape coefficients, and \( \tilde{b} \in \mathbb{R}^{3m-3} \) is the texture coefficients. 3D morphable model (3DMM) is defined as a multidimensional linear combination function, which is given in Eq. (1). The probability for \( \tilde{a} \) is defined as

\[
p(\tilde{a}) \propto \exp\left[-\frac{1}{2} \sum_{i=1}^{m-1} (\alpha_i/\sigma_i)^2\right],
\]
(2)

with \( \sigma_i^2 \) being the eigenvalues of the shape covariance matrix. The probability for \( \tilde{b} \) is computed similarly.

To reconstruct a 3D face from the input image, the original 3DMM [7] optimizes the shape and texture coefficients along with a set of rendering parameters such that they produce an image as close as possible to the input image. It builds relatively accurate and realistic 3D faces, but it is liable to fall into local minima and quite time consuming.

To make the reconstruction fast and reliable, a 3D shape morphable model, a 3DMM focusing only on the shape part is proposed. It is defined with the shape part of Eq.(1). Further the step, this method only uses the feature points as constraints on the set of the solutions. The optimization function is given as

\[
\tilde{a}^* = \arg \min_{\tilde{a}} \sum_{i=1}^{m-1} \frac{1}{\sigma_i} (\alpha_i)^2,
\]

\[s.t. \ \ \ t_i^{\text{img}} = \Gamma \left( \tilde{S}(f_i) + \sum_{k=1}^{m-1} \alpha_k s_k(f_i) \right), i \in [1, N],\]
(3)

where \( f_i \) is the \( i \)-th 3D feature vertex on the face. \( \tilde{S}(f_i) \) is the \( f_i \) of the average shape, \( s_k(f_i) \) is the \( f_i \) of \( s_k \). \( N \) is the feature number. \( t_i^{\text{img}} \) is the \( i \)-th image feature point matched with \( f_i \). The mapping from a 3D feature vertex to its 2D feature point \( \Gamma \) is assumed as a linear transformation:

\[
\Gamma(X, Y, Z) = M(s, \theta)[X, Y]^T + t,
\]

\[
M(s, \theta)[X, Y]^T = \begin{bmatrix}
  s \cos \theta \cdot X - s \sin \theta \cdot Y \\
  s \sin \theta \cdot X + s \cos \theta \cdot Y
\end{bmatrix},
\]

\[
t = (t_x, t_y)^T,
\]
(4)

where \( t \) is translation, \( \theta \) is plane rotation and \( s \) is scale. \( \Gamma(X, Y, Z) \) is the projection of \( (X, Y, Z)^T \) on the image. Then \( \Gamma \) can be obtained by solving Eq. (5) with least square method.

\[
\arg \min_{s, \theta, t} \sum_{i=1}^{N} \| t_i^{\text{img}} - \Gamma(\tilde{S}(f_i)) \|^2.
\]
(5)

After that, the constraints of Eq. (3) are solved with Singular Value Decomposition for the initialization of \( \tilde{a} \). And the optimum \( \tilde{a}^* \) of Eq. (3) can be found by Levenberg-Marquardt method. Thus, the face shape \( \hat{S} \) can be reconstructed with Eq. (1).

3.4 3D Face Deformation

The \( X, Y \) coordinates of the feature vertices on the reconstructed face are somewhat different from the \( X, Y \) coordinates of the feature points on the image, because the shape space is limited by the 3D face database. Thus, we want to deform the reconstructed face to ensure that the feature vertices are exactly correct. The \( X, Y \) coordinates of the feature vertices on the face are adjusted to be aligned to the \( X, Y \) coordinates of the feature points on the image at first. Then, according to the displacements of the feature vertices, one interpolation method is used to compute the displacements of the non-feature vertices. The process is the shape adjustment for the 3D reconstructed face. It is like the deformation of \( \hat{S} \). The deformation function based on radial basis functions (RBF) has the form

\[
p_j^{\text{new}} = \sum_{i=1}^{N} \gamma_i \phi(r_j - r_i) + a + bx_j + cy_j, j \in [1, n],
\]
(6)

where \( p_j \) is the \( j \)-th vertex of \( \hat{S} \), \( p_j^{\text{new}} \) is its new position, \( f_i \) is the \( i \)-th feature vertex of \( \hat{S} \), \( \gamma_i, i \in [1, N] \) are the deformation coefficients, and \( a, b, c \) are the affine components. \( \phi(r) \) is a radially symmetric basis function. The form \( \phi(r) = \exp(-k||r||) \) is used here. \( k \) is a positive coefficient related with image size. \( x_j \) and \( y_j \) is the coordinates of \( p_j \). The feature point matching results are used to calculate all these coefficients with:

\[
\begin{bmatrix}
  \phi(f_i^{x, y} - f_j^{x, y}) \\
  \vdots \\
  \phi(f_i^{x, y})
\end{bmatrix}
= \begin{bmatrix}
  1 & f_i^{x, y} & \gamma_1 \\
  \vdots & \vdots & \vdots \\
  1 & f_i^{x, y} & \gamma_N
\end{bmatrix}
\begin{bmatrix}
  f_i^{\text{img}} \\
  \gamma
\end{bmatrix},
\]
(7)

where \( \hat{S}(f_i^{x, y}) \) is written as \( f_i^{x, y} \), it is a row vector with the \( X, Y \) coordinates of \( \hat{S}(f_i) \). After \( \gamma_i, i \in [1, N] \) and \( a, b, c \) are gotten from Eq. (7), \( \hat{S} \) is deformed with Eq. (6). In this way, the final 3D face shape model is obtained.

3.5 Texture Generation and Texture Mapping

The input frontal view image is taken as the texture of the generated 3D face. Base on the matching between image feature points and the 3D feature points, the orthogonal projection matrix from the 3D face to the image plane can be calculated. Thus, the texture coordinate of every 3D vertex can be gotten. Then, the orthographic projection is used to map the input image to the generated 3D face shape model. In this way, the photo-realistic 3D face model is generated.
4. Face Swapping Based on 3D Faces

The traditional 2D image based face swapping [16, 17] suffers from the pose and appearance similarity restriction of the input faces. Our reconstructed 3D face can deal with this problem by rendering virtual images with any required poses. The proposed face swapping method is shown in Fig. 2. It focuses on swapping frontal source faces to the pose-free target faces. The succeeding subsections discuss how the virtual source image is generated and how the color transfer and image blending methods are used for the face swapping.

4.1 Virtual Source Image Generation

The target face is detected to generate its facial mask and its pose is estimated. There are many methods for estimating the head pose, such as Li’s work [21]. The virtual source image with this estimated pose can be rendered. At the same time, the facial mask of the virtual source image is generated with the predefined face region on the 3D face. To replace the target face with the source face, both images are needed to be aligned to a common coordinate system. In this paper, the virtual source image will be aligned to the coordinate system of the target image through three fiducial points: both iris centers and the mouth center. Least square estimation is adopted for this rigid transformation.

4.2 Traditional Color Transfer and Image Blending

The reconstructed face usually has different skin color with the target face. The target skin color need to be transferred to the source face at first. Usually, in RGB space, most pixels have large values for the red and green channel if the blue channel is large. This implies that all the color channels must be modified in tandem if we want to change the appearance of a pixel’s color in a coherent way. In this case, color transfer [22] is used to solve this problem. The color transfer method processes images in laβ space. This space minimizes correlation between channels for many natural scenes, which let us apply different operations in different color channels with some confidence that undesirable cross channel artifacts won’t occur. The process is done pixel by pixel. It is as follows:

1. The images are translated from RGB space to laβ space.
2. The means \( \langle l_s \rangle, \langle \alpha_s \rangle, \langle \beta_s \rangle \) and the standard deviations \( \sigma_{ls}^2, \sigma_{\alpha s}^2, \sigma_{\beta s}^2 \) for each axis of the source face (extracted with the source facial mask) are calculated separately in laβ space.
3. The means \( \langle l_t \rangle, \langle \alpha_t \rangle, \langle \beta_t \rangle \) and the standard deviations \( \sigma_{lt}^2, \sigma_{\alpha t}^2, \sigma_{\beta t}^2 \) for each axis of the target face (extracted with the target facial mask) are calculated separately in laβ space.
4. The means and the standard deviations of the source face region are replaced with those of the target face region. \( l', \alpha', \beta' \) are the color transfer result in laβ space.

\[
\begin{bmatrix}
    l'(x, y) \\
    \alpha'(x, y) \\
    \beta'(x, y)
\end{bmatrix} =
\begin{bmatrix}
    \frac{\langle l_t \rangle}{\sigma_{lt}^2} & 0 & 0 \\
    0 & \frac{\langle \alpha_t \rangle}{\sigma_{\alpha t}^2} & 0 \\
    0 & 0 & \frac{\langle \beta_t \rangle}{\sigma_{\beta t}^2}
\end{bmatrix} \cdot
\begin{bmatrix}
    l_s(x, y) + \langle l_s \rangle \\
    \alpha_s(x, y) + \langle \alpha_s \rangle \\
    \beta_s(x, y) + \langle \beta_s \rangle
\end{bmatrix}
\]

where \( l_s(x, y) = l_t(x, y) - \langle l_t \rangle, \alpha_s(x, y) = \alpha_t(x, y) - \langle \alpha_t \rangle, \beta_s(x, y) = \beta_t(x, y) - \langle \beta_t \rangle \).

5. Then the result is converted back to RGB space by using Eq. (8) and Eq. (9).

Color transfer makes the source face have the same skin color distribution as the target face. However, directly replacement may still generate a non-smooth boundary between the face and the non-face region. To make the replacement seamlessly, multi-resolution spline technique [23] is adopted for image blending. The multi-resolution spline technique goes as follows.

1. For both the target image and the virtual source image, Gaussian pyramid and Laplacian pyramid are calculated. They are defined as \( G_k \) and \( L_k \). \( G_0 \) is the original image. Each Gaussian pyramid level is generated from its predecessor with a REDUCE operation. The level number is denoted by \( n \). For \( 0 < k < n \):

\[
G_k = \text{REDUCE}[G_{k-1}].
\]

by which we mean:

\[
G_k(x, y) = \frac{1}{4} \sum_{a=0}^{1} \sum_{b=0}^{1} G_{k-1}(2x + a, 2y + b).
\]

The Laplacian pyramid is constructed with the Gaussian pyramid and an EXPAND operation. Say \( \text{EXPAND}[G_{k+1}] = G_{k+1}^{\text{EP}} \), \( G_{k+1}^{\text{EP}} \) is defined as

\[
G_{k+1}^{\text{EP}}(2x, 2y) = G_{k+1}(x, y),
\]

\[
G_{k+1}^{\text{EP}}(2x + 1, 2y) = \frac{1}{2}[G_{k+1}(x, y) + G_{k+1}(x + 1, y)],
\]

\[
G_{k+1}^{\text{EP}}(2x, 2y + 1) = \frac{1}{2}[G_{k+1}(x, y) + G_{k+1}(x, y + 1)],
\]

\[
G_{k+1}^{\text{EP}}(2x + 1, 2y + 1) = \frac{1}{4}\left( G_{k+1}(x+1, y) + G_{k+1}(x, y+1) + G_{k+1}(x+1, y+1) \right).
\]

\( L_n = G_n \). For \( 0 < k < n \):

\[
L_k = \text{EXPAND}[L_{k+1}].
\]
Fig. 6 The face swapping based on color transfer and image blending. (a) The target image. (b) The virtual source image. (c) The face swapping result with the source image directly. (d) The face swapping result with only color transfer. (e) The final face swapping result with both color transfer and image blending.

2. At each level, Laplacian images are combined using a weighted average with a weight map. The merge mask is a binary image, 1 for the face region and 0 for the non-face region. It will be filtered with Gaussian function and serve as the weight map. At level \( k \), \( W_k \) is the weight map, and \( L_k \) is the combined result. For \( 0 < k < n \):

\[
L_k(x, y) = L_k^*(x, y) - W_k(x, y) + (1 - W_k(x, y)) \cdot L_k^*(x, y)
\]

(15)

\( L_k^* \) and \( L_k^\prime \) are the Laplacian image at level \( k \) of the source image and the target image respectively.

3. The combined image \( L_k \) is augmented with \( R_{k+1} \) to get \( R_k \), which is the result for each level. The image reconstruction is from level \( n \) to level \( 0 \). \( R_n = L_n \). The final result is \( R_0 \). The reconstructed image \( R_{k+1} \) is expanded with Eq. (13) to the same size of \( L_k \) and adds with it. \( R_{k}^{EP} \) stands for \( EXPAND[R_{k+1}] \). For \( 0 < k < n \), the reconstruction can be formulated as

\[
R_k(x, y) = L_k(x, y) + R_{k+1}^{EP}(x, y).
\]

(16)

Figure 6 gives one example for this part. It presents the target image, the virtual source image, the face replacement result with the source face directly, the face replacement result with only color transfer, and the final face swapping result. This experiment shows the significance of adopting both the color transfer and image blending method. The source image is from the database of [17].

In the previous works, the illumination was either estimated by a linear combination of spherical harmonics [16] or a 3D morphable model [7]. These methods are quite complicated. Since color transfer can transmit illumination to a certain degree, no further adjustment is done.

4.3 A Color Transfer Merging Method

Convenient as it is, color transfer has its own limitations. Its result’s quality depends on the images’ similarity in composition. For example, if the illumination of the target face and the source face are quite different, the transfer of statistics is expected to fail. To remedy this issue, Reinhard et al. [22] proposed a modified color transfer based on different clusters. Both images were separated into different swatches, leading to several pairs of clusters. Each of them had similar composition. The source image was adjusted according to the statistics associated with each of the cluster pairs by using Eq. (10). Then all the color adjusted images were blended in \( l \alpha \beta \) space, yielding the final color. The final result \( l \alpha \beta \) for pixel \((x, y)\) is calculated as

\[
\begin{align*}
I'(x, y) &= \frac{\sum_{l=1}^{K} w_l(x, y) \cdot I_l(x, y)}{\sum_{l=1}^{K} w_l(x, y)} \\
\alpha'(x, y) &= \frac{\sum_{l=1}^{K} w_l(x, y) \cdot \alpha_l(x, y)}{\sum_{l=1}^{K} w_l(x, y)} \\
\beta'(x, y) &= \frac{\sum_{l=1}^{K} w_l(x, y) \cdot \beta_l(x, y)}{\sum_{l=1}^{K} w_l(x, y)}
\end{align*}
\]

(17)

where \( K \) is the cluster number, \( I_l \), \( \alpha_l \), \( \beta_l \) are the color transfer result with the \( i \)-th cluster pair, and \( \omega_l \), \( \alpha_l \), \( \beta_l \) are the weights of \( I_l \), \( \alpha_l \), \( \beta_l \) for the final result. These weights are inversely proportional to their normalized distances \( I_l, \alpha_l, \beta_l \), \( \langle I_l \rangle, \langle \alpha_l \rangle, \langle \beta_l \rangle \) and \( \sigma^2_\alpha, \sigma^2_\beta, \sigma^2_\gamma \) are the means and the standard deviations of the \( i \)-th cluster of the source image.

In this paper, the face part is divided into seven independent regions: two eyes, the nose, the mouth, two cheeks, and a surrounding region. Each region forms a cluster. For the target image, these regions are generated based on the face alignment results. And for the generated virtual source image, these regions are gained by the projection of the pre-defined facial parts on the 3D face shape. Then the image can be generated with Eq. (17).

However, the traditional color transfer Eq. (17) implies that this process might generate pixels with RGB value out of the range of [0, 255] due to the influences by other regions based color transfer results. Basically, the more statistically different the two regions are, the more this possibility is. To deal with this problem, this article proposes a color transfer merging method. Instead of the weighted blending the color adjusted images, a seamless texture merging method is used to blend them together in RGB space. The basic idea is to preserve the correct color transfer result generated with the corresponding cluster pair and eliminate the errors introduced by other cluster pairs. Thus, for every region of the result image, only the color transfer result generated by its corresponding cluster pair is preserved. And along the boundaries between different regions, bilinear interpolations are done for merging. The merging function is defined as

\[
F_{new}(x, y) = \sum_{i=1}^{K} I_i(x, y) \cdot W_i(x, y)
\]

\[
+ \sum_{j=1}^{K} \frac{I_j(x, y) - (1 - W_j(x, y))}{K - 1} \cdot H_j(x, y)
\]

(18)

where \( I_i \) is the color transfer result in RGB space with the \( i \)-th cluster pair. \( H_i \) is the binary mask for the \( i \)-th cluster. That is to say, when a certain region is considered, only the pixels of that region are 1, and others are 0. \( W_i \) is \( H_i \) filtered with Gaussian function. \( F_{new} \) is the final result.

Comparison of different color adjustment methods based face swapping results are shown in Fig. 7. It shows the face swapping result with the source image, with the
result of a single face region based color transfer by using Eq. (10), with the result of the seven regions based color transfer by using Eq. (17) and with the proposed method by using Eq. (18). In Fig. 7, the 1st row gives the both images and their seven selected regions. The 2nd row gives the original source image, a single face region based color transfer result, seven regions based color transfer result and the proposed merging result. The 3rd row gives the face swapping results based on the corresponding color adjustment results in the 2nd row. The traditional regions based color transfer (the 3rd image of the 2nd row) has color discontinuity and noises, which is eliminated by the proposed color transfer merging method (the 4th image of the 2nd row). That is to say, the proposed method can generate more satisfied result and more suitable for the face swapping application. The target and the source image are from Zhu’s dataset [17].

5. Experimental Results and Evaluations

5.1 Results of the 3D Face Reconstruction

The proposed algorithm is first tested on images taken under controlled viewing conditions by rendering images of faces from the USF 3D face database [20]. For this experiment, all 100 faces are used for PCA, and the first 79 eigenvectors are used for constructing the 3D shape morphable model. All the rendered images are taken as input images to reconstruct 3D faces. Since the ground-truth shape is available, the accuracy of the reconstruction can be demonstrated with error maps. The reconstruction error maps are calculated per vertex as $\frac{\|p_j - p_{gt,j}\|}{\|p_{gt,j}\|}$, where $p_{gt,j} = [x_{gt,j}, y_{gt,j}, z_{gt,j}]^T$ denotes the ground-truth value of the $j$-th vertex, and $p_j = [x_j, y_j, z_j]^T$ is its reconstructed value. $j \in [1, n], n$ is the vertices number. The facial feature points are generated by projecting the 3D feature vertices on the image. The reconstruction accuracy evaluation is done with leave-one-out cross validation. To demonstrate the effect of adopting the 3D face deformation, two reconstruction methods are considered: one is the reconstruction method without the deformation (3D shape morphable model only), named Rec.1, and the other is the proposed reconstruction method (3D shape morphable model plus deformation), named Rec.2.

In Fig. 8 four examples of typical reconstructions from the USF 3D face database are presented. In each example the input image, its facial feature points and two triplets (two viewpoints) of shapes: the ground-truth shape, the reconstructed shape of Rec.1 and the reconstructed shape of Rec.2 are shown. Two error maps: between the ground-truth and the reconstructed shape of Rec.1 and between the ground-truth and the reconstructed shape of Rec.2 are presented as well. The numbers under each of the error maps indicate the mean error and the standard deviation (in percents). By
The overall mean reconstruction error for each of the 100 faces in the USF 3D face database. The blue dots mark the errors between the results of Rec.1 and the ground-truth, and the red squares mark the errors between the results of Rec.2 and the ground-truth. (a) The shape reconstruction error. (b) The texture reconstruction error.

Comparing the two error maps in each example we can see how the reconstructed shape is modified by the deformation algorithm. The first example has the smallest mean error and the fourth one has the biggest mean error in all the 100 faces. The first two recovered shapes are consistently similar to the ground-truth shapes. However, the registration method between faces has errors for some faces like the last two faces (see the facial feature points). That’s one of the reasons why the last two shapes are not well recovered.

In Fig. 9, the overall mean shape and texture reconstruction error for each face are presented. In both figures, the blue dots mark the errors between the results of Rec.1 and the ground-truth values, and the red squares mark the errors between the results of Rec.2 and the ground-truth values. In Fig. 9(a) the shape reconstruction error for each face multiplied by 100 to indicate percents is presented. The overall means and standard deviations are $4.47 \pm 2.14\%$ and $4.45 \pm 2.11\%$ for Rec.1 and Rec.2 respectively. In Fig. 9(b) the texture reconstruction error for each face is presented. It is calculated per vertex as $\|q_j - q_{gt}^j\|_1$, where $q_{gt}^j = [r_{gt}^j, g_{gt}^j, b_{gt}^j]^T$ denotes the ground-truth texture value of the $j$-th vertex, and $q_j = [r_j, g_j, b_j]^T$ denotes its mapped texture value. $j \in [1, n]$. The overall means and standard deviations are $17.08 \pm 21.33$ and $16.65 \pm 20.46$ for Rec.1 and Rec.2. Figure 9 shows Rec.2 has equal or better performance than Rec.1 in most cases. That is to say, it can generate more personalized models. The reconstruction process runs 2.0 s and 2.8 s with and without deformation separately on the average on a 3.2 GHz Intel Core i5 processor, which is quite fast.

### 5.2 Results of Face Swapping

To demonstrate the proposed method, several cartoon images with predefined head poses are used for face swapping at first, shown in Fig. 11. It demonstrates our user can upload their images and do any kind of role play. The parents may generate one storybook easily for the child using their favorite character. An example of a storybook is shown in Fig. 12. It is from a famous German fairy tale: Rapunzel. In Fig. 13, it shows our method can deal with target images with large-view point differences which traditional 2D face swapping methods will fail. The poses of the target faces are around $\pm 40^\circ$ in yaw or $\pm 30^\circ$ in pitch. They are presented with (Yaw, Pitch, Roll). The illumination conditions are quite different between the inputs in most cases. However, the result shows the proposed color transfer method can solve the lighting inconsistency problem to some degree.

Some comparison experiments with two present face swapping methods are done as well. In Fig. 14 the face de-
identification comparative results between our method and Bitouk et al.’s work [16] are shown. Our method seems to be equally effective with theirs. In Fig. 15 the comparative results between our method and Zhu et al.’s work [17] are shown. It shows that our results are more similar with the input face, and their results seem a little distorted.

The input images of Fig. 11 and Fig. 13 are photographed by us. The inputs of Fig. 12, Fig. 14 and Fig. 15 are from the database of [17]. These results show this method can deal with a wide range of images. In addition, the traditional face swapping cannot replace the faces with significant pose variations, such as pairs in Fig. 13. That is to say our method is more robust to pose differences.

Since it is impossible to get ground-truth for the swapped face, the best way to evaluate is still subjective. For the face swapping results from Fig. 11 to Fig. 15, fourteen testers are asked to give scores with 5, 4, 3, 2, 1. 5 stands for the wonderful performance, and 1 stands for the poor performance. The average score for every result is shown in Fig. 16. The total average score for all the results is 4.2, which shows our system is quite impressive. The scores for real images face swapping are quite high (4.5), while the scores for the cartoon images are relative low (3.6). The lowest score is for first replaced face in Fig. 12. The reason is that the cartoon faces are usually quite different from real human faces. We only adopt 3 feature points to align the input faces, which will generate not that satisfied result. More representing feature points will be selected in the future.

6. Conclusion

A deformable 3D shape morphable model for building 3D faces from a single frontal view image is proposed. The proposed algorithm models a personalized 3D face using the extracted image feature points and a priori knowledge of the face shape provided by the 3D face training data set. The experiment shows the reconstruction is fast and effi-
cient. Then, a 3D face based target pose-free face swapping method is presented. It can eliminate the traditional pose and appearance similarity restriction. Thus, the applications of the traditional 2D face swapping method are further extended. Applications such as generating story books, role play, and face de-identification are accomplished. The results show our method is effective even when the target character has a non-frontal face. One limitation of the proposed method is its dependence on the training data set. More 3D faces with various expressions will be added to form a better model. And for those quite unique faces, such as faces with many wrinkles or scars, our method may not generate satisfied results. More inputs will be needed to deal with this problem. Besides, the color transfer merging method might fail under complex illumination. In the future, a more powerful lighting adjustment method will be introduced.

Acknowledgments

This work was supported by the National High Technology Research and Development Program of China (863 program) under Grant No. 2009AA112214, the National Natural Science Foundation of China under Grant No. 61071135, and USA Hewlett-Packard Development Company.

References

[1] I.Y. Jang and K.H. Lee, “Depth video based human model reconstruction resolving self-occlusion,” IEEE Trans. Consum. Electron., vol.56, no.3, pp.1933–1941, Aug. 2010.
[2] B. Amberg, A. Blake, A. Fitzgibbon, S. Romdhani, and T. Vetter, “Reconstructing high quality face-surfaces using model based stereo,” Proc. IEEE Int. Conf. Computer Vision, Rio de Janeiro, Brazil, Oct. 2007.
[3] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. GROSS, “High-quality single-shot capture of facial geometry,” Proc. ACM SIGGRAPH, Los Angeles, California, USA, July 2010.
[4] A. Ansari and M. Abdel-Mottaleb, “Automatic facial feature extraction and 3D face modeling using two orthogonal views with application to 3D face recognition,” Pattern Recognit., vol.38, no.12, pp.2549–2563, Dec. 2005.
[5] X. Sun, Y. Zheng, and Z. Wang, “Model-assisted face reconstruction based on binocular stereo,” Proc. Visual Communications and Image Processing, Huang Shan, An Hui, China, July 2010.
[6] S. Ryu, S.H. Lee, and J. Park, “Real-time 3D surface modeling for image based relighting,” IEEE Trans. Consum. Electron., vol.55, no.4, pp.2431–2435, Nov. 2009.
[7] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3D faces,” Proc. ACM SIGGRAPH, pp.187–194, Los Angeles, California, USA, Aug. 1999.
[8] M.D. Levine and Y.C. Yu, “State-of-the-art of 3D facial reconstruction methods for face recognition based on a single 2D training image per person,” Pattern Recognit. Lett., vol.30, no.10, pp.908–913, July 2009.
[9] S. Romdhani and T. Vetter, “Estimating 3-D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior,” Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp.986–993, San Diego, CA, USA, June 2005.
[10] S. Romdhani, V. Blanz, and T. Vetter, “Face identification by matching a 3D morphable model using linear shape and texture error functions,” Proc. European Conf. Computer Vision, pp.5–19, Copenhagen, Denmark, May, June 2002.
[11] I.K. Park, H. Zhang, and V. Vezhnovets, “Image-based 3D face modeling system,” EURASIP J. Applied Signal Processing, vol.2005, pp.2072–2090, Jan. 2005.
[12] Y. Lin, S. Wang, Q. Lin, and F. Tang, “Face swapping under large pose variations: A 3D model based approach,” Proc. IEEE Int. Conf. Multimedia and Expo 2012, Melbourne, Australia, July 2012.
[13] W.-S. Lee and N. Magnenat-Thalmann, “Fast head modeling for animation,” Image Vis. Comput., vol.18, no.4, pp.355–364, March 2000.
[14] S. Chin and K.Y. Kim, “Expressive 3D face for mobile devices,” IEEE Trans. Consum. Electron., vol.54, no.3, pp.1294–1302, Aug. 2008.
[15] P. Viola and M.J. Jones, “Robust real time face detection,” Int. J. Comput. Vis., vol.57, no.2, pp.137–154, May 2004.
[16] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S.K. Nayar, “Face swapping: automatically replacing faces in photographs,” Proc. ACM SIGGRAPH, L.A., CA, USA, Aug. 11-15, 2008.
[17] J. Zhu, L.V. Gool, and S.C.H. Hoi, “Unsupervised face alignment by robust nonrigid mapping,” Proc. IEEE Int. Conf. Computer Vision, pp.1265–1272, Kyoto, Japan, Sept.–Oct. 2009.
[18] P. Wang, X. Tong, Y. Du, J. Li, W. Hu, and Y. Zhang, “Augmented makeover based on 3D morphable model,” Proc. ACM Int. Conf. Multimedia, pp.1569–1572, Scottsdale, Arizona, USA, Nov.–Dec. 2011.
[19] L. Wang, X. Ding, and C. Fang, “Generic face alignment using an improved active shape model,” Proc. Int. Conf. Audio, Language and Image Processing, pp.317–321, Shanghai, China, July 2008.
[20] V. Blanz and T. Vetter, “Face recognition based on fitting a 3D morphable model,” IEEE Trans. Pattern Anal. Mach. Intel., vol.25, no.9, pp.1063–1074, Sept. 2003.
[21] Y. Li, S. Wang, and X. Ding, “Person-independent head pose estimation based on random forest regression,” Proc. IEEE Int. Conf. Image Processing, pp.1521–1524, H.K., China, Sept. 26–29, 2010.
[22] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, “Color transfer between images,” IEEE Comput. Graph. Appl., vol.21, no.5, pp.34–41, Sept. 2001.
[23] P.J. Burt and E.H. Adelson, “A multiresolution spline with applications to image mosaics,” ACM Trans. Graphics, vol.2, no.4, pp.217–236, Oct. 1983.

Yuan Lin received the B.S. degree in electronic engineering in 2008 from Dalian University of Technology, China. Currently, she is pursing the PhD degree in the State Key Laboratory of Intelligent Technology and Systems, Department of Electronic Engineering, Tsinghua University, China. Her interests include human computer interactive, computer vision and image processing.

Shengjin Wang received the B.E. degree from Tsinghua University, China in 1985 and the Ph.D. degree from Tokyo Institute of Technology, Japan in 1997. From May 1997 to Aug. 2003, he was a Member of the Senior Research Staff in the Internet System Research Laboratories, NEC Corporation, Nara, Japan. Since Sept. 2003, he has been a Professor with the Department of Electronic Engineering, Tsinghua University. His current research interests include computer vision, video surveillance, and virtual reality.