Removal of 3D Facial Expressions: A Learning-based Approach

Gang Pan, Song Han, Zhaohui Wu, Yuting Zhang
Department of Computer Science, Zhejiang University, China
{gpan,hansong1983,wzh,zyt@zju.edu.cn}

Abstract

This paper focuses on the task of recovering the neutral 3D face of a person when given his/her 3D face model with facial expression. We propose a learning-based expression removal framework to tackle this task. Our basic idea is to model expression residue from samples, and then use the inferred expression residue from the input expressional face model to recover the neutral one. A two-step non-rigid alignment method is introduced to make all the face models topologically share a common structure. Then we construct two spaces, normal space and expression residue space, for modeling expression. Therefore, the expression removal problem can be formalized as the inference of expression residue from normal spaces. The neutral face model can be generated in a Poisson-based framework by the inferred expression residue. The experimental results on BU-3DFED database demonstrate the effectiveness of our approach.

1. Introduction

Facial expression is one of the most common nonverbal communication manners in our daily life, which has a close relationship with psychological activities and physiological signals. It exists ubiquitously in our living environments. It has attracted many researchers in the area of facial expression analysis, recognition, and synthesis [7, 18, 23]. Facial expressions occurs so frequently in the social context, however, sometimes we actually would not want the existence of expressions since it challenges our tasks, for example, automatic face recognition.

This paper addresses the problem of removing expression from a 3D expressional face model. Detailedly, given a 3D face model of a person with certain expression, we need to wipe off the expression in order to get a neutral face model which still preserves the original facial characteristics of the person.

Although 3D facial expression removal has been little addressed in the literature, however, its benefits can be in many aspects. 1) It may enhance the performance of 3D face recognition. Expression variation is one of the most factors that degrade the recognition accuracy of 3D face recognition[4, 29, 15, 1]. Expression removal technique can eliminate the unwanted expressional information while at the same time preserving details of facial characteristics of a person. 2) It could play a vital role for 3D gender classification. Current 3D gender classification methods can only deal with neural face[14, 13]. Gender classification in the existence of expression is still a challenging problem. 3) It might be a promising way to analyze complex 3D facial expressions. Most work on expression analysis is restricted to tackle prototypical expressions[28, 9, 26], whereas realistic expression could be more complicated so that it is difficult to handle. 4) It is helpful for face synthesis. Most of the existing methods for face synthesis work with neutral faces [11, 21, 33, 17]. Given an expressional face, changing it to another expression is beyond the capability of the current methods. While the expression removal technology can be employed to obtain a neutral face, and then various state-of-the-art face synthesis approach can be utilized.

Expression removal requires to get its neutral face while still preserving its person-specific facial characteristics. However expressions generally bring large shape deformation which change the original facial appearance. To the best of our knowledge, there is few effort focusing on 3D facial expression removal. Three-dimensional facial expression synthesis [12], which is a quite close area to 3D facial expression removal, can be deemed as the reverse process of expression removal. The current 3D expression synthesis approaches can be roughly classified into four categories: 1) *Interpolation-based methods* define an interpolation function to specify smooth motions between two key frames [18, 21]. Although interpolation usually is fast, they cannot generate arbitrary realistic facial expressions ; 2)*Approaches based on physical muscle models* propagate muscle strength in an elastic spring mesh to model facial expressions [6, 30, 10]; 3) *Example-based methods* synthesize expressions from a neutral face based on a collections of examples [11, 17]. It can synthesize expressions in real-time with low computational cost; 4) *Pseudo muscle models* simulate muscle forces by splines, tensors and free deformation models [27, 33, 24]. Although it can generate smooth and

flexible facial expressions, tuning parameters is a difficult process. Specifically, Blanz et al[3] proposed a face reanimation method for 2D images and videos. By the aid of 3D face models, it's effective in adding expressions.

This paper is organized as follows. In the next section, the framework of the proposed approach for 3D facial expression removal is introduced. Section 3 gives a non-rigid alignment method, which aligns an expressional face model to the generic face model, thus all the face models can topologically share a common structure. Section 4 describes two spaces for modeling face expressions. Section 5 infers the expression residue when an expressional face as an input, with which the neutral face can be reconstructed in a Poisson-based framework, given in Section 6. The experimental results are presented in Section 7. Finally, conclusion is given in Section 8.

2. Framework of the proposed approach

Given an expressional 3D face as the input, our goal is to obtain its neutral 3D face by removing its expressional residues. Our basic idea is to infer the expressional residue from the input by learning, and then synthesize the neutral face by subtracting the expressional residue from the expressional 3D face of input.

In order to infer the expressional residue from the input, we build two spaces, *normal space* and *expression residue space*. Firstly, a 3D face is adapted to a generic 3D face by the proposed two-step non-rigid alignment method, so that all the 3D faces can share a common geometric structure. Thus, each 3D face can be described by a point in the normal space, and its expression residue compared with its neutral face can also be represented by a point in the expression residue space. Then, we learn the relationship between the two spaces by samples. With the learned relationship model, the expression residue of the input can be computed from the input.

Logically, synthesis of the neutral face can be achieved by "subtract" the expressional residue from the given expressional face. However, it is a non-trival task. This paper employs a Poisson-based deformation framework, which is a linear system, to reconstruct the neutral face with the given expression face as the deformation source and the expressional residue as the deformation path. The proposed framework is shown in Fig. 1.

3. Non-rigid Alignment of Facial Expressions

Original captured 3D face models are irregular and posture-variant. It's difficult to propose a proper mapping scheme [7]. Sun, Y. and Yin, L. proposed a interpolation function for 3D face adaptation [23]. The interpolation function is a combination of an affine transformation and a RBF regression. Although the adaptation process is

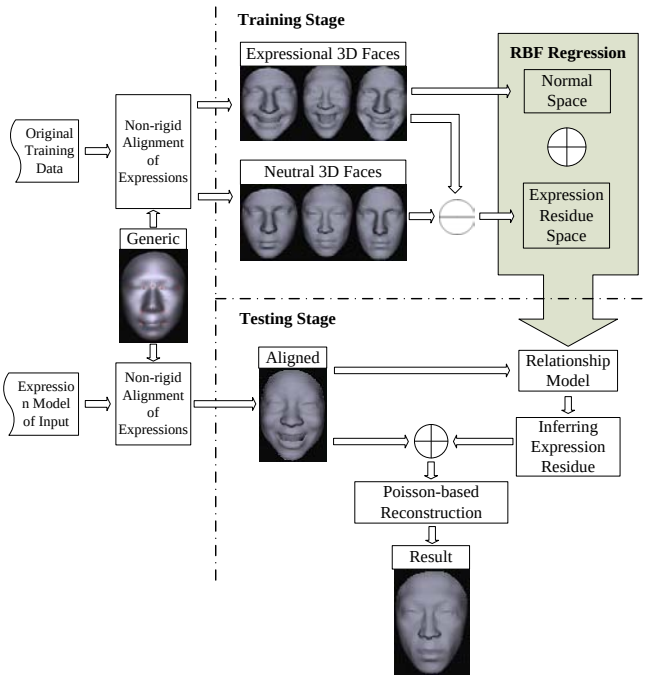

Figure 1. Diagram of our framework

really fast, the accuracy can't meet our requirement. And the tradeoff between time consuming and accuracy was not considered simultaneously. In order to eliminate this defect and improve the quality of adaptation, Rigid Adjustment and Energy-based Generic Model Adaptation is designed to "resample" each original 3D face model by the generic model. The generic model used in our experiment is shown in Fig. 2. Notice that the mouth of the generic model has been "cut" open to avoid effect of unstable changes. We propose a method to generate a mesh from a point cloud by fitting a generic mesh model. The fitting is constrained by a predefined set of landmarks located respectively on the corresponding positions of the generic model and the point cloud. We use G to represent the generic model whose point set is P_G and use O to represent the original model whose point set is P_O, L_G and L_O represent the landmark set of G and O respectively. Our expression alignment consists of two steps as follow.

3.1. Landmark-constrained Rigid Adjustment

We employ 11 prominent landmarks in a 3D face for accurate alignment. We assume that all the landmarks (shown in Fig. 2) are available. So we can focus on the problem of facial expression removal itself. In our experiment, we adopt the triangular mesh object which conveys both the geometric information of points and the topological information between points.

In order to adjust the posture of O, we employ the Iterative Closest Point (ICP)[2] method to move O towards G. We combine the landmark-constraint with ICP. For each

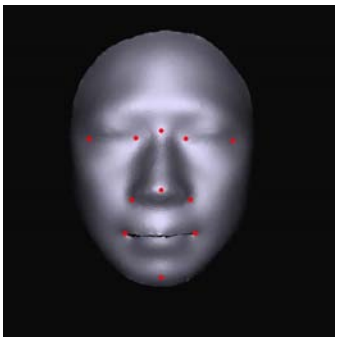

Figure 2. The generic model used in this paper, which has 7686 vertices and 15045 triangles. The 11 landmarks are marked in red, namely inner and outer corners of left and right eyes, nose tip, left and right mouth corners, bottom of chin, and left and right corners of nose wing.

point $x_i \in P_O$, if $x_i \in L_O$, find its corresponding landmark $y_i \in L_G$; else find the nearest point $y_i \in P_G$ as its corresponding point.

After the posture adjustment process, the posture of O coincides with that of G. Thus the affect of rigid variation component can be eliminated.

3.2. Energy-based Generic Model Adaptation

After the above adjustment, each O holds the same posture with the generic model G. Then for each O, the generic model G is deformed in order to fit the shape of O. This adaptation problem can be converted into an energy minimization problem. The goal of the energy function is to smoothly wrap the shape of G so as to match that of adjusted O. To meet this goal, we define two energy measures, specifically geometric error E_g and smooth error E_s. The geometric error E_g which measures the quality of wrapping has the following form:

$$E_g(G,O) = \delta \sum_{y_i \in L_G} (||y_i + t_i - x_i||^2) + \sum_{y_i \in \bar{L_G}} (||y_i + t_i - x_i||^2) \quad (1)$$

where t_i denotes the offset of y_i, which needs to be solved, δ is the weight of landmarks against common point.

The smooth error E_s is used to measure the synchronicity and smoothness of the adaptation process, which is defined as follows:

$$E_s(G,O) = \delta \sum_{y_i \in L_G} (\sum_{j \in N(i)} ||t_i - t_j||^2) + \sum_{y_i \in \bar{L_G}} (\sum_{j \in N(i)} ||t_i - t_j||^2) \quad (2)$$

where $N(i)$ denotes the 1-ring neighbor of point i, t_i and t_j are offsets of point i and j respectively.

The final energy function is defined as

$$E(\lambda, \delta) = \lambda E_g(G,O) + (1-\lambda) E_s(G,O) \quad (3)$$

Notice that $0 \leq \lambda \leq 1$ is used as a tradeoff between geometric error and smooth error. We start the energy minimization process with large value of λ and δ to quickly yield a proper initial correspondence. Then we gradually decrease the value of λ and δ to ensure a steady and accurate adaptation process. By adjusting λ and δ, we can measure the tradeoff between time-consuming and accuracy of the adaptation process. The adaptation process is performed by algorithm 1.

Algorithm 1: Energy-based Generic Model Adaptation

Inputs: G: the generic 3D face model; L_G: the landmark set of G; O: the original 3D face model; L_O: the landmark set of O;

Output: M: the aligned 3D face

Variables: x_i is a point in P_O; y_i is a point in P_G; t_i: the offset for point y_i

1 For each point $y_i \in P_G$, If $y_i \notin L_G$, find its nearest point $x_i \in P_O$ to form a pair (y_i, x_i); else choose its corresponding point $x_i \in L_O$ to form a pair (y_i, x_i).

2 For each point y_i, calculate its offset t_i by minimizing the energy function:

$$\arg \min_t E(\lambda, \delta). \quad (4)$$

3 Update each point y_i by adding its offset t_i

$$y_i = y_i + t_i. \quad (5)$$

4 Compute the root mean squared distance ε_k between P_O and P_G. If ε_k is less than a preset threshold, then go to step 5; else, go to step 1.

5 Obtain the aligned 3D face M of O by $M = O$.

After the expression alignment process, the generic model G, like an elastic net, matches the shape of O and moreover its posture keeps in a fixed orientation. Therefore we obtain an aligned 3D face model M for an original 3D face model O.

4. Two Spaces for Expression Modeling

We propose to model expression variations by constructing two spaces: normal space and expression residue space. Normal space is used to reveal properties of different facial expressions. Expression residue space is used to model expression variations compared with their neutral faces. From the geometrical perspective, expression variation can be

considered as a certain shape variation. Since normal direction is stable against noises and capable of preserving sensible shape variations, it is a fundamental attribute for shape representation.[16]. Suppose we have an aligned 3D face model M. We use T to represent the triangle set of M and $n^j = (n_x^j, n_y^j, n_z^j)$ to represent the normal of jth triangle on M.

4.1. Normal Space

Let K represent the number of triangles on M. We use $\{T^j\}_{j=1}^K$ to denote the set of all triangles on M collected in a certain order. Then the regular representation for M in the normal space has the form of $C = \{n^j\}_{j=1}^K$ with $C^j = n^j$.

4.2. Expression Residue Space

Since a 3D facial surface is comprised of various triangles. Facial expression on a 3D facial surface can be perceived as a combination of various movements of each triangle. We proposed to use a 5-tuple to describe this movement. On the unit sphere, the elevation angle and azimuth angle of n^j has the form of

$$\begin{aligned} \phi^j &= \arctan \frac{n_y^j}{n_x^j} \\ \theta^j &= \frac{\pi}{2} - \arcsin n_z^j \end{aligned} \tag{6}$$

respectively.

Thus the residue between a neutral 3D face M_{neu} and its corresponding expressional 3D face M_{exp} on the jth triangle is a 5-tuple which has the following form,

$$\begin{aligned} \mathrm{d}_j(M_{exp}, M_{neu}) = (&\theta_{exp}^j - \theta_{neu}^j, \phi_{exp}^j - \phi_{neu}^j, \\ &x_{exp}^j - x_{neu}^j, y_{exp}^j - y_{neu}^j, z_{exp}^j - z_{neu}^j) \end{aligned} \tag{7}$$

where $(x_{exp}^j, y_{exp}^j, z_{exp}^j)$ and $(x_{neu}^j, y_{neu}^j, z_{neu}^j)$ are barycenters of jth triangle on M_{exp} and M_{neu} respectively. Therefore the regular expression residue between M_{exp} and M_{neu} has the form of $\Delta(M_{exp}, M_{neu}) = \{\mathrm{d}_j(M_{exp}, M_{neu}) | 1 \leq j \leq K\}$, where neu and exp are short for neutral and expressional respectively. $\Delta(M_{exp}, M_{neu})$ is a point in the expression residue space.

5. Inferring Expression Residue

After the expression spaces have been constructed, our expression removal problem can be formalized as:

> *Given N expressional 3D faces and their corresponding neutral 3D faces, how can we infer the expression residue for an incoming expressional 3D face?*

From the definition of expression residue, the computation of neutral 3D face can be formalized as

$$M_{neu} = M_{exp} \ominus \Delta(M_{exp}, M_{neu}), \tag{8}$$

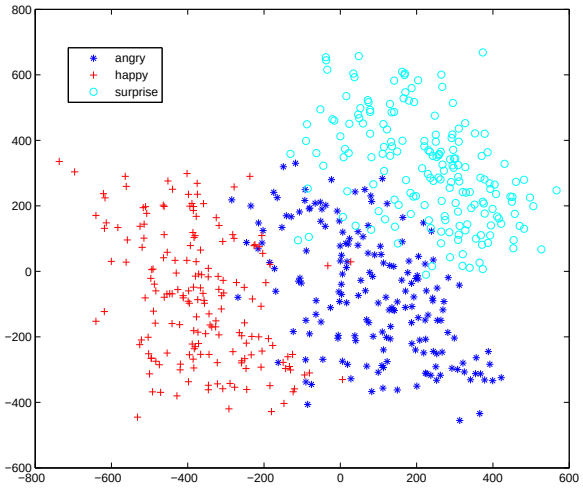

Figure 3. Illustration of three expressions (angry, happy, and surprise) in the reduced two-dimensional space R^2.

where $\ominus$ denotes the removal of expression residue from M_{exp}.

Let C_{exp} be the normal representation of M_{exp} in the normal space, the expression removal problem can be converted to infer the expression residue from C_{exp}.

$$\Delta(M_{exp}, M_{neu}) \approxeq \hat{\Delta}(C_{exp}) \tag{9}$$

If the expression residue can be effectively inferred, we can subtract the residue from the expressional 3D face M^{exp} to obtain the neutral 3D face M^{neu} by

$$M_{neu} = M_{exp} \ominus \hat{\Delta}(C_{exp}). \tag{10}$$

We approximate the correlations of normal space and expression residue space using a couple of basis functions with RBF kernels. It is worth noting that the sample space may not be uniformly distributed. In order to adapt the distribution of sample space, we adopt the Mahalanobis distance measurement for RBF regression.

5.1. Dimension Reduction of Normal Space

Since the dimensionality of the normal space is very large. The normal space is sparse which means that it contains redundant or even noisy information. Therefore we need to reduce the dimensionality of the Normal Space to get a compact representation with noise excluded. Fig. 3 shows a demonstration of the reduced normal space in R^2, which can reflect the distribution of different expressions. Notice that the normal space is not an Euclidean space, which means that we can't directly employ PCA for dimension reduction. Suppose $n = (n_x, n_y, n_z)$ is a normal vector, μ is a base point for reference, the log map and expo-

nential map which were first proposed by Fletcher [8] are given by

$$LOG_\mu(n) = (n_x \frac{\arccos n_z}{\sin \arccos n_z}, n_y \frac{\arccos n_z}{\sin \arccos n_z}) \quad (11)$$

$$EXP_\mu(\Delta\mu) = (\Delta\mu_1 \frac{\sin \|\Delta\mu\|}{\|\Delta\mu\|}, \Delta\mu_2 \frac{\sin \|\Delta\mu\|}{\|\Delta\mu\|}, \cos \|\Delta\mu\|) \quad (12)$$

respectively. Since the log map and exponential map are able to measure distance of the non-linear normal space, we adopt these measurements for dimension reduction. Suppose there are N samples $\{M_i\}_{i=1}^N$ in the training set and we use n_i^j to denote the unit normal direction of the jth triangle on the ith sample. Consider μ^j as the geodesic mean of the jth normal vectors of all training samples. Then the modeling is divided into three parts as follow:

1. Solving Geodesic Mean Initially set $\mu^j = n_1^j$ and solving for μ^j is actually a refining process proposed in [8]. After the refinement process, we can obtain a geodesic mean for each triangle, namely $\{\mu^j\}_{j=1}^K$.

2. Geodesic Analysis by PCA For each training sample M^i , calculate its centralized geodesic coordinates $u_i = (LOG_{\mu^1}(C_i^1), \ldots, LOG_{\mu^K}(C_i^K))$ by counting its deviation from geodesic means. After that, the covariance matrix can be constructed in the form of $S = \frac{1}{N}\sum_{i=1}^N u_i u_i^T$. Finally we perform the SVD (singular value decomposition) on the covariance matrix S to acquire eigenvectors $(v_1, ..., v_N)$ and their corresponding eigenvalues $(\lambda_1, ..., \lambda_N)$ with $\lambda_1 \geq \lambda_2 \cdot \cdot \geq \lambda_N$.

3. Intrinsic Subspace Projection Let the matrix $U = (u_1, ..., u_N)$ represent those centralized geodesic coordinates of N samples in the training set. The top V eigenvectors are selected to form the intrinsic projection subspace which is $P = (v_1, ..., v_V)$, where $V = \arg\limits_V \frac{\sum_i^V \lambda_i}{\sum_i^K \lambda_i} \cong \xi$ and the threshold ξ is a predefined energy preserving parameter. Then coordinates of N training samples in the intrinsic subspace have the form of $U^P = U \cdot P^T = (u_1^P, ..., u_N^P)$.

5.2. Inference of Expression Residue

After the normal space reduction, we employ RBF regression to build relationship between normal space and expression residue space. The RBF networks are able to model complex mappings and have the advantage of being much simpler than perceptrons while keeping the major property of universal approximation of functions[19]. The RBF regression algorithm is to firstly choose some proper data points as radial basis function centers and then use singular value decomposition to solve for the weights of the network. The most distinguishing feature of RBF functions is that they are local, or at least their response decreases monotonically away from a central point. Since the sample space may not be uniformly distributed. In order to adapt the distribution of sample space, we adopt the Mahalanobis distance measurement for RBF regression. Specifically, the distance measurement for each central point is approximated by its neighborhood. This local preserving advantage is essential for a precise prediction.

The RBF regression has the following form,

$$\Delta_i = \sum_{j=1}^n w_{ij} k(u_i^P, u_j^P) \quad (13)$$

where $k(u_i^P, u_j^P) = \exp(-(u_i^P - u_j^P)^T \cdot \Sigma_i^{-1} \cdot (u_i^P - u_j^P))$, Σ_i denotes the Mahalanobis distance matrix constructed as follows:

$$\Sigma_i = \frac{1}{k} \sum\nolimits_{u_j^P \in neighbors\ of\ u_i^P} (u_j^P - c_i)(u_j^P - c_i)^T \quad (14)$$

and

$$c_i = \frac{1}{k} \sum\nolimits_{u_j^P \in neighbors\ of\ u_i^P} u_j^P \quad (15)$$

where k is the number of neighbors.

The matrix form of RBF can be written as

$$\Delta = W * K = [w_1, ..., w_n] * K \quad (16)$$

Given samples C of normal space and the corresponding Δ of expression residue space, we can train the RBF weight matrix W by orthogonal least squares methods [5]. When u^P, dimension reduction of a probe C_{in}, is as an input, the corresponding expression residue $\hat{\mathbf{\Delta}}_{\mathbf{in}}$ can be obtained using the matrix W.

6. Poisson-based Reconstruction from Expression Residue

After the expression residue $\hat{\mathbf{\Delta}}_{\mathbf{in}} = (\hat{d}_1, ..., \hat{d}_K)$ has been inferred, we can modify the input expressional 3D face M_{in} by rotating and translating each triangle i with $\hat{d}_i = (\Delta\hat{\theta}^i, \Delta\hat{\phi}^i, \Delta\hat{x}^i, \Delta\hat{y}^i, \Delta\hat{z}^i)$. For each triangle, we set up a local coordinate system by taking the barycenter as the origin point. After rotating and translating each triangle, the 3D face is broken into pieces.

Then we can employ the Poisson deformation for reconstruction [32]. Some prerequired knowledge of Poisson deformation[32, 22, 25, 20] are provided as follows.

The discrete Poisson equation has the form of

$$\Delta(u) \equiv Div(\nabla u) = Div(\xi), \quad (17)$$

where $\Delta(u)$ is the discrete Laplace-Beltrami operator, ∇u is the discrete gradient operator, $Div(\xi)$ is the discrete divergence operator. Now we can rewrite the discrete Poisson

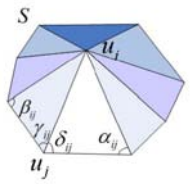

Figure 4. 3D 1-ring neighborhood

equation to a sparse linear system:

$$AU = b \tag{18}$$

where U is the coordinates of unknown deformed mesh, b is the divergence of the gradient fields modified by inferred expression residue. The sparse matrix A is defined as:

$$A_{ij} = \begin{cases} -\frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij}) & if\ j \in N(i) \\ -\sum_{k\in N(i)} A_{ik} & if\ i = j \\ 0 & otherwise \end{cases} \tag{19}$$

where $alpha_{ij}$ and β_{ij} are shown in Fig. 4

Then we compute the three modified gradient fields to construct the discrete Poisson equation. By solving the linear system in equation 18, we can "paste" the broken pieces together so as to obtain the neutral face.

7. Experiments

7.1. Data

We use the BU-3DFED(Binghamton University 3D Facial Expression Database)[31] to evaluate our algorithm. The BU-3DFED database was built for the purpose of 3D facial expression research. There are 44 males and 56 females in the database. Each subject in the database was required to perform 6 different expressions (i.e. anger, disgust, fear, happiness, sadness, surprise) and a neutral expression. Each of the 6 prototypical expressions has 4 levels of intensity. To evaluate the performance of the proposed algorithm, we enroll all the 3D face models of six expressions of the highest intensity and the neutral expression. Thus, the total number of face models for experiments is 100*6+100=700.

We employ the leave-one-person-out methodology to evaluate our approach. For each test, only one expressional face is chosen as the probe, and all the other 99 subjects' models are for training. So we can guarantee that the subject who acts as the probe will never appear in the corresponding training set. The RMS (root mean square) is used to quantitatively measure the performance of the expression removal algorithm. The RMS between two face models X and Y is defined as:

$$\mathrm{RMS}(X,Y) = (Dist(X,Y) + Dist(Y,X))/2 \tag{20}$$

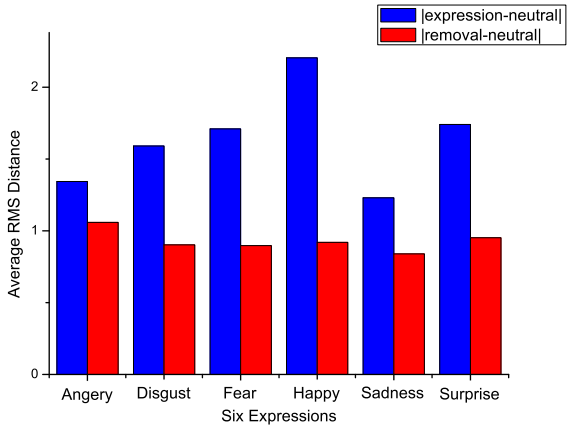

Figure 5. Comparison of average RMS distance for $|expression - neutral|$ and $|removal - neutral|$

$$Dist(X,Y) = \sqrt{\frac{\sum_{i=1}^{n} \|x_i - y_i\|^2}{n}} \tag{21}$$

where n is the number of points on X, x_i is a point on X, y_i is a point of Y which is nearest to x_i.

7.2. Results

Fig. 6 shows some results of expression removal. Each row demonstrates one expression with two samples, in the top-down order of anger, disgust, fear, happy, sadness, and surprise. Fig. 6(a) and (d) are the expressional faces of input, Fig. 6(b) and (e) are the neutral faces of ground truth, Fig. 6(c) and (f) are the expression removal results. From the figure, our expression removal approach achieves promising results, in despite of existence of large expression variation such as fear, happy, and surprise.

To quantitatively measure the performance, for each expression removal test, we compute the RMS distance between the expression removal face and its neutral face of ground truth. The RMS distance between the expressional face of input and its neutral face is also computed. Fig. 5 shows the comparison result of the average RMS distance of $|expression - neutral|$, and that of $|removal - neutral|$.

The experiments are carried out on our system of CPU Intel Core 2 Quad 2.66GHz with 2048MB RAM. Each expression removal test completes in less than one minute.

8. Conclusion

This paper provides a 3D facial expression removal algorithm. Given an expressional face model of input, it can wipe off the corresponding expression to get a neutral output. We build two spaces, *normal space* and *expression residue space*, for statistically modeling the relationship be-

(a) (b) (c) (d) (e) (f)

Figure 6. Some results of expression removal for six expressions. Each row is for one expression, in the top-down order of anger, disgust, fear, happy, sadness, and surprise. (a) and (d): expressional faces of input; (b) and (e): neutral faces of ground truth; (c) and (f): the expression removal results.

tween expressional faces and expression residue. A two-step non-rigid alignment method is presented to adapt a 3D face to the generic 3D face, so that 3D faces can share a common geometric structure, which facilitates the representation in the two spaces. Eventually, the expression removal problem is solved in a Poisson-based deformation framework using the inferred expression residue. The proposed approach is evaluated by the public 3D face database *BU-3DFED*. The results shows the effectiveness of the approach.

Acknowledgements

This work is partly supported by NSFC (60525202), National 863 Program (2009AA011900), and PCSIRT Program (IRT0652).

References

[1] F. Al-Osaimi, M. Bennamoun, and A. Mian. An Expression Deformation Approach to Non-rigid 3D Face Recognition. *IJCV*, 81(3):302–316, 2009. 1

[2] P. Besl and H. McKay. A method for registration of 3-D shapes. *IEEE T-PAMI*, 14(2):239–256, 1992. 2

[3] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *Computer Graphics Forum*, volume 22, pages 641–650, 2003. 2

[4] K. Chang, K. Bowyer, P. Flynn, et al. Multiple nose region matching for 3D face recognition under varying facial expression. *IEEE T-PAMI*, 28(10):1695, 2006. 1

[5] S. Chen, C. Cowan, and P. Grant. Orthogonal least squares learning algorithm for radial basis function network. *IEEE Trans. Neural Networks*, 2(2):302–309, 1991. 5

[6] P. Ekman, W. Friesen, and J. Hager. Facial action coding system. 1978. 1

[7] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36(1):259–275, 2003. 1, 2

[8] P. Fletcher, C. Lu, S. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE T-MI*, 23(8):995–1005, 2004. 5

[9] B. Gong, Y. Wang, J. Liu, and X. Tang. Automatic Expression Recognition on a Single 3D Face by Exploring Shape Deformation. In *ACM Multimedia*, 2009. 1

[10] K. Kahler, J. Haber, H. Yamauchi, and H. Seidel. Head shop: Generating animated head models with anatomical structure. In *SCA '02*, pages 55–63, 2002. 1

[11] C. Kouadio, P. Poulin, and P. Lachapelle. Real-time facial animation based upon a bank of 3D facial expressions. In *Proc. of the Computer Animation*, page 128, 1998. 1

[12] S. Krinidis, I. Buciu, and I. Pitas. Facial expression analysis and synthesis: A survey. In *HCI'03*, pages 1432–1433. 1

[13] Y. Liu and J. Palmer. A quantified study of facial asymmetry in 3d faces. In *AMFG'03*. 1

[14] X. Lu, H. Chen, and A. Jain. Multimodal facial gender and ethnicity identification. *Advances in Biometrics*, 2006. 1

[15] X. Lu and A. Jain. Deformation modeling for robust 3D face matching. *IEEE T-PAMI*, 30(8):1346–1356, 2008. 1

[16] M. Meyer, M. Desbrun, P. Schroder, and A. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and mathematics*, 3:35–57, 2002. 4

[17] J. Minoi, S. Amin, C. Thomaz, and D. Gillies. Synthesizing realistic expressions in 3D face data sets. In *BTAS'08*. 1

[18] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH*, pages 75–84, 1998. 1

[19] T. Poggio and F. Girosi. Networks for approximation and learning. In *Proceedings of the IEEE*, 1990. 5

[20] K. Polthier and E. Preuß. Identifying vector field singularities using a discrete Hodge decomposition. *Visualization and Mathematics III*, 2002. 5

[21] H. Pyun, Y. Kim, W. Chae, H. Kang, and S. Shin. An Example-Based Approach for Facial Expression Cloning. ACM SIGGRAPH. In *Eurographics Symposium on Computer Animation*, pages 167–176, 2003. 1

[22] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986. 5

[23] Y. Sun and L. Yin. Facial expression recognition based on 3D dynamic range model sequences. In *ECCV'08*. 1, 2

[24] D. Tao, M. Song, and X. L. Bayesian Tensor Approach for 3D Face Modelling. *IEEE Transactions on CSVT*, 10-12(71):1397–1410, 2008. 1

[25] Y. Tong, S. Lombeyda, A. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition. *ACM Transactions on Graphics*, 22(3):445–452, 2003. 5

[26] Y. Venkatesh, A. Kassim, and O. Ramana Murthy. A novel approach to classification of facial expressions from 3D-mesh datasets using modified PCA. *Pattern Recognition Letters*, 30(12):1128–1137, 2009. 1

[27] D. Vlasic, M. Brand, H. Pfister, and J. Popovic. Face transfer with multilinear models. *ACM Trans. Graphics*, (3):426–433, 2005. 1

[28] J. Wang, L. Yin, X. Wei, and Y. Sun. 3D facial expression recognition based on primitive surface feature distribution. In *CVPR'06*, volume 2, pages 1399–1406, 2006. 1

[29] Y. Wang, G. Pan, Z. Wu, and Y. Wang. 3D face recognition in the presence of expression: A guidance-based constraint deformation approach. In *IEEE CVPR*, pages 1–7, 2007. 1

[30] K. Waters. A muscle model for animation three-dimensional facial expression. *ACM SIGGRAPH*, 21(4):17–24, 1987. 1

[31] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato. A 3D facial expression database for facial behavior research. In *FGR'06*, pages 211–216, 2006. 6

[32] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H. Shum. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graphics*, (3):644–651, 2004. 5

[33] Q. Zhang, Z. Liu, B. Guo, D. Terzopoulos, and H. Shum. Geometry-driven photorealistic facial expression synthesis. *IEEE T-VCG*, 12(1):48–60, 2006. 1