

Regression-Based Facial Expression Optimization

Hui Yu, Honghai Liu, Senior *Member*, *IEEE*

Abstract—This paper presents an approach for reproducing optimal 3D facial expressions based on blendshape regression. It aims to improve fidelity of facial expressions but maintain the efficiency of the blendshape method, which is necessary for human-machine interaction and online avatars etc. The proposed method intends to optimize the given facial expression using Action Units (AUs) based on the Facial Action Coding System (FACS) which were recorded from real human beings. To help capture facial movements for the target face, an intermediate model space has then been generated, in which both the target and source AUs have the same mesh topology and vertex number. The optimization can be conducted interactively in the intermediate model space through adjusting the regulating parameter. Finally, the optimized facial expression model is transferred back to the target facial model to produce the final facial expression. We have demonstrated that given a sketched facial expression with rough vertex positions indicating the intended facial expression, the proposed method is able to approach the sketched facial expression through automatically selecting blendshapes with corresponding weights. The sketched expression model is finally approximated through AUs representing true muscle movements, which is able to significantly improve the fidelity of facial expressions.

Index Terms—Blendshape, facial expression, FACS action unit, online avatar, optimization, regression

I. INTRODUCTION

FACIAL EXPRESSION techniques play a very important role in the success of human-machine interaction, online virtual worlds, the animation industry, computer games and online avatars for mobile devices etc. [1, 3]. In the early stages, to create a facial animation, animators had to tediously sculpt facial expressions by using hand crafted skills. Since the pioneering work conducted by Parke [9], the blendshape method has been popular in industries due to its efficiency and intuitive controllability. However, this method still suffers from several disadvantages making it difficult to achieve ideal facial expressions. On one hand, the post-processing and

corrective procedure for the blended expressions is time-consuming. Digital artists need to spend massive amount of time to manually correct vertex positions. Since human beings are very sensitive to even subtle glitches of facial movements, it is non-trivial to place vertices in the proper positions to satisfy the sensitivity of human perception. On the other hand, since there is no pure boundary for vertex deviations among different blendshapes, deviation of overlap vertices will accumulate to generate unrealistic or exaggerated facial movements with more than one blendshape being added to the facial expression. This problem becomes worse when more blendshapes are involved in facial expression generation. Artists have to repeatedly adjust positions for those overlapped vertices, which makes the facial animation generation a painful task. Therefore, how to solve the aforementioned problems has become a challenging and interesting piece of work.

In this paper, the proposed method aims to solve these problems and improve the fidelity of facial expressions but maintain the efficiency of the blendshape method, with the hope of significantly improving the users' experience. The artist sketched facial expression can then be updated with true muscle movements but keeps the key characteristics of the original expression. To this end two questions need to be solved. First, how to transfer muscle movements as much as possible from the source AUs to the target face. Second, how to keep the characteristics of the original artist's sketched facial expression.

For the first question, we created an intermediate model space through nonrigid registration using a dense template mesh. To solve the second question, we proposed an optimization scheme in the intermediate model space. We developed an approach to automatically correct vertex positions through iteratively optimizing blendshapes using captured facial movements from real actors. In the application, we use AUs as blendshapes based on FACS. Those AUs were recorded from certified FACS actors. Our method will not completely replace artist's manual work for facial animation generation. Instead, it intends to help digital artists with their post-processing, which will significantly reduce the manual work and processing time.

We propose to optimize the sketched facial expression based on the regression concept. Relevant blendshapes and their corresponding weights are automatically produced during

Manuscript received November, 2013, revised Feb. 2014.

H. Yu is with Intelligent Systems and Robotics Group, School of Creative Technologies, CCI, University of Portsmouth, Portsmouth, UK (phone: +44(0)239 284 5470; email: hui.yu@port.ac.uk).

H. H. Liu is with Intelligent Systems and Robotics Group, School of Creative Technologies, CCI, University of Portsmouth, Portsmouth, UK (phone: +44(0)23 9284 5541; email: honghai.liu@port.ac.uk).

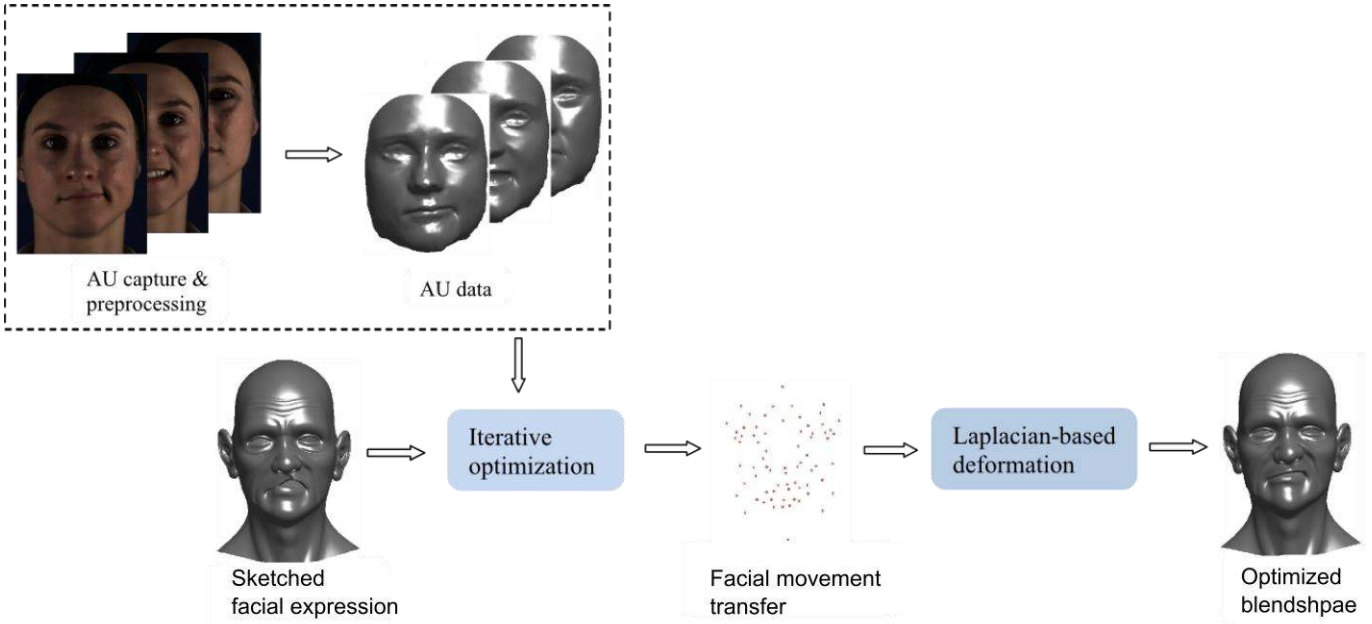


Fig. 1. Schematic diagram for facial expression optimization

the optimization, which are then used to generate the target facial expression based on the blendshape method. An intermediate optimized facial expression model is obtained during this stage. And the final optimized facial expression is achieved through mapping the intermediate optimized facial expression to the target neutral face. The schematic diagram is illustrated in Fig. 1.

Our contributions are triple fold. Firstly, a regression-based blending method has been developed to approximate sketched facial deviations. This method can effectively avoid the challenging problem in shape blending: negative and cancelling combinations. The combination of automatically selected blendshapes with their corresponding weights can approximate the sketched facial expression with high quality. There is no need to specifically constrain the weights in a certain range. Relevant AUs and their corresponding weights are produced during the optimization procedure. Secondly, smooth weight propagation for blending shapes has been considered. Vertices' positions on blendshapes represent facial muscle movements, which can nicely match the source face model. Finally, since those blendshapes used in this paper are peak frames of AUs recorded from real human beings, they represent the maximum muscle movements for normal facial expressions. This actually sets movement constraints for facial shape vertices. Any vertex movements that goes beyond the deviation boundary will be forced to fall back. Thus, unrealistic facial vertex movements can be avoided. As a dense template mesh is used to transfer the muscle movement to the target face mode, it can capture some wrinkles on the face.

II. RELATED WORK

Various techniques have been developed to create blendshape facial animations since Parke's work [9]. Many blending shape models have been explored from user-defined blendshapes, facial action units to scanned data [3, 12, 14, 21, 22, 28, 30]. Pighin et al. [4] developed a system for creating photorealistic 3D facial animation by manually assigning blending weights. In their method, a blending interface has been provided to specify the blending function including global blend, regional blend and a painting tool. Principal Component Analysis (PCA) has been used to compute weights for blendshape face model generation [14]. These principal components correspond to each key blendshape, which are then combined to generate the new face model through adjusting corresponding weights [29]. Joshi et al. [12] proposed to segment the face into subregions with similar amount of deformation. Those subregions fitted to the motion capture data by finding their blendshape weights.

Choe et al. [31] developed a method for synthesizing facial expressions using hand-generated muscle actuation bases. An iterative algorithm was used, taking the form of fixed point iteration as input for a set of markers. However, it lacks flexibility in practice since the 3D model is constrained to the actor's face. Liu et al. [6] presented a optimization method using motion capture data. It then used an optimization algorithm to find the transformation matrix including rigid and nonrigid deformation and relied on Radial Basis Function (RBF) to transfer the optimized shape back to the target face. However, the final facial movements may not fully reflect the true facial muscle movement due to the RBF-based transfer based on sparse landmarks. Lewis et al. [27] proposed a system for direct manipulation of vertices for blendshape facial expression generation. Chuang et al. [24] applied the



Fig. 2. Illustration of recorded AU image (row 1 and 2) and 3D shape (row 3 and 4) examples: AU1, AU4, AU5, AU10 right, AU12, AU14 left, AU23 and AU25.

source weights to the target face to generate facial animations. But due to large positive and negative blendshape combinations, their method led to poor animation results. Some efforts have been taken trying to overcome the extrapolation problem, such as constraining blendshape weights in a certain range [25, 26]. Li et al. [28] constructed an orthogonal blendshape model base on facial regions, which also applied constraints on weight propagation.

Those methods mentioned above have different focuses and details when creating facial animation based on the principle of blendshape. However, most of them require proficient skills making those methods less feasible for professional projects. Some methods lack realistic representation of true facial muscle movements, thus leaving too much flexibility to users to obtain the desired facial expression. In our method, nonprofessional users can explore arbitrary facial expressions without considering the plausible face space.

FACS invented by Ekman et al. [13], is a widely accepted technique for analyzing facial expressions and activities. It defines a set of Action Units encoding any facial action in terms of small visible units of muscle movements. FACS intends to represent possible visually detectable facial variations in terms of 44 AUs. With FACS, human coders can manually code nearly any anatomically possible facial

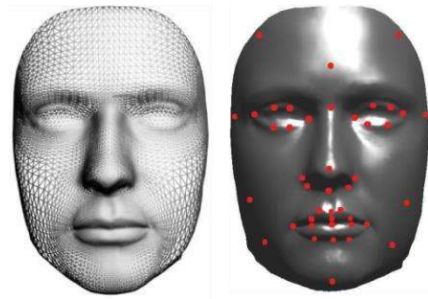


Fig. 3. Illustration of the template face model and landmarks

expressions, which can then be decoded into the specific AU. Thus, it is intuitive to represent facial expressions by combinations of certain AUs. FACS has already been successfully applied in computer games and movies [10]. And some facial expression generation methods have been developed based on the principle of FACS [18]. In this paper an AU database recorded from certified FACS actors is used [16]. In the database, captured data has been processed to produce a sequence of 3D face models for each AU with the same mesh topology. Fig. 2 demonstrates some example images of those recorded AUs. Facial movements represented by vertex deviations from the neutral face can be obtained by using various techniques [16, 30]. Since those AUs are recorded from real human beings, vertex movements of those AUs represent true muscle motions.

III. GENERATE THE INTERMEDIATE MODEL SPACE

As the optimization of sketched facial expression is actually taken place in the intermediate space, the optimization is more efficient and flexible. To this end, we generate intermediate face models for the target neutral face and the facial expression face using a template mesh with the same vertex number (4375 vertices) and mesh topology of the source AU mesh. Fig. 3 shows the template face model and manually labeled landmarks. Landmarks on the face are used as assistance during shape fitting for more accurate and efficient shape fitting. The facial expression optimization procedure and the vertex “boundary” regulation are both implemented in the intermediate model space. Though we use FACS AUs in this paper, the deviation “boundary” definition can be generalized to any blendshapes created by digital artists.

Both source AUs and target face models have the same vertex number and mesh topology in the intermediate model space, which gives more flexibility to process the facial expression. After the model fitting, two new intermediate target face models are generated. Because the intermediate target face model has the same vertex number and topology with the source AU models, we transfer the source AUs to the intermediate target model using the method proposed by Yu et al. [11]. A set of AUs for the target face and the target facial expression model from artist in the intermediate model space have been obtained through the fitting procedure (Fig. 4).



Fig. 4. Illustration of target AUs in the intermediate model space

IV. FACIAL EXPRESSION OPTIMIZATION

The blendshape facial animation method is a general technique for combining weighted underlying parameters from various methods eg. FACS, PCA, Facial Animation Parameters (FAP) specified in MPEG-4 etc. It assumes that facial expressions can be interpreted and parameterized in terms of different definitions, eg. PCA, FACS and so on. Through controlling these underlying parameters, animators are able to create various facial expressions. Various methods have been proposed for animating and manipulating blendshape faces [5,15]. The blendshape method can be expressed using the following equation:

(1)

where S is the blendshape expression, S_0 the neutral face model, d_i the set of difference between the neutral face and the captured 3D AU shape S_i^{AU} , $d_i = S_i^{AU} - S_0$, and w_i the set of corresponding weights of each blendshape. Therefore, the blendshape quality largely depends on the given blendshapes and blending weights. In this paper, we have developed an iterative scheme to find the combination of base shapes involved in the facial expression and their corresponding blending weights.

Given an expressive facial expression of the target, the proposed scheme interprets it in terms of AUs. The corresponding weights of AUs are obtained through this iterative procedure by introducing constraints.

And a deformation method is developed to deform the target face model using the optimized deformation of markers. To generate a facial animation, we adopt the performance-driven method by adapting the recorded AU motion to the synthesized facial expression frame.

We rely on the optimization procedure to determine the blendshapes and corresponding weight distribution. Through shape fitting, dense meshes for both target face AUs and target facial expression have been obtained in the intermediate model space. Thus, we have options on how many vertices are involved in the optimization procedure. We can use the full mesh with 4375 vertices or the simplified mesh with fewer vertices. With the full mesh, we can have better representation of muscle movements for facial expression but takes more time to converge during the optimization compared with the simplified mesh.

The idea of using the optimization of AUs to approximate the sketched facial expression is based on the regression concept that patterns from a single object class lie in a linear space [20]. And this concept is consistent with the blendshape method. Thus any given facial expression can be represented by linearly combining the expression samples from the same identities as follows:

$$\tilde{S} = WS + \tau \quad (2)$$

Where $W \in \mathcal{R}^N$ is the set of unknown weights. More specifically, W can be expressed as $W = (w_1, w_2, \dots, w_N)$, N the number of AU involved in the blendshape blending ($N=40$). And \tilde{S} is the sketched facial expression with S being AUs, $\tau \in \mathcal{R}^n$ a residual term with a bounded energy $\|\tau\|_2 \leq \varepsilon$.

Rather than optimizing vertex positions to approximate the target expression, we choose to use deviations of each blendshape. Since vertex deviations represent the movement of a set of facial muscles, by optimizing deviations, we intend to approximate the target movements using blendshape movements. We use AUs as blendshapes, the deviations of each blendshape is actually the AU deviations from the neutral face. Thus, equation (2) can be rewritten as follows:

$$\tilde{D} = WD + \tau \quad (3)$$

Where $D = (d_1^i, d_2^i, \dots, d_n^i)$ is deviations of n vertices of the AU, n being the total number of shape vertices ($n \leq 4735$).

And \tilde{D} is the difference between the sketched face and the neutral face. The term $W \in \mathcal{R}^N$ is the set of unknown weights as explained above in (2). The problem for matching the right AUs and computing their corresponding weights can be solved as an optimization problem. The error function can be formulated as follows:

$$F_w(w_i) = \|WD - \tilde{D}\|^2 \quad (4)$$

However, the error function in (4) does not guarantee a valid and ideal result [19]. This is because the data can be overfitted during the process of least square error minimization. As a result, it sometimes stretches the weighting W to unreasonably large values, which results in unrealistic AU deviations beyond the full limitation. When the space $n > N$, the weight matrix W can be estimated in the least square sense as follow:

$$\hat{W} = (D^T D)^{-1} D^T \tilde{D} \quad (5)$$

However, in practice the inverse operator of the matrix in above equation often fails due to the singularity of the matrix $D^T D$, since the matrix of blendshapes cannot guarantee a full column rank, $rank(D) < n$. This could happen even when the feature dimension is greater than the blendshape space, $n > N$. The reason behind this is that those blendshapes may not be orthogonal, which causes linear correlations among the data.

To solve the problem in the above equation, we suppose that the approximation includes two parts: the coarse and refined regression. The coarse regression part determines the dominant blendshapes while the refined part is used to approximate the minor deviations. Since the vertex deviations of the blendshapes may not orthogonal, it is possible to linearly combine this subspace for refined approximation. Thus equation (3) can be expressed as follows:

$$\hat{D} = W_1 D + W_r D_r + \tau \quad (6)$$

Where D_r can be seen as the blending shape set without those dominant blendshapes. And $W_1 \in \mathfrak{R}^N$ and $W_r \in \mathfrak{R}^N$ are weightings for shape D and D_r respectively.

Equation (6) is a $n \times 40$ ($n > 40$), overdetermined linear system. Theoretically, the above equation can be estimated in the least-square sense as follows:

$$\begin{bmatrix} \hat{W}_1 \\ \hat{W}_2 \end{bmatrix} = \begin{bmatrix} D^T \\ D_r^T \end{bmatrix} [D, D_r]^{-1} \begin{bmatrix} D^T \\ D_r^T \end{bmatrix} \quad (7)$$

However, equation (7) is not practical, since we never know D_r until we obtain D . To overcome this problem, we introduce the following regularization term:

$$F(w, \alpha) = \left\| \sum_{i=1}^N w_i - \alpha \right\|^2 \quad (8)$$

where α is a control parameter. With the value of α , we can control the refined blendshapes and their corresponding

weights. We can obtain a coarse regression and \hat{W}_1 by setting $\alpha = 0$. Through adjusting α , we can further adjust the involved blendshapes and their corresponding weights. Thus, the minimization function can be written as follows:

$$\arg \min_{w_i, \lambda} \left\| \sum_{i=1}^N w_i D_i^{au} - \tilde{D} \right\|^2 + \lambda^2 \left\| \sum_{i=1}^N w_i - \alpha \right\|^2 \quad (9)$$

The regulating parameter λ is an experimental value. Equation (9) can be minimized interactively to find the ‘‘best match’’ optimized facial expression model with the sketched one. The error between the sketched facial expression and the optimized one can be computed using the root-mean-square (RMS) error e_{rms} :

$$e_{rms} = \frac{1}{n} \left(\sum_{i=1}^n \left\| \hat{S} - \tilde{S} \right\|_2^2 \right)^{\frac{1}{2}} \quad (10)$$

where \hat{S} and \tilde{S} are the optimized shape and the target shape respectively.

V. FACE DEFORMATION

If full meshes are used in the intermediate model space during optimization, we obtain an optimized facial expression with a full mesh in the intermediate model space. Then the optimized facial expression model is mapped to the target neutral face. In the case where simplified face models are used for optimization, we deform the full mesh in the intermediate model space using the Laplacian deformation method.

A. Deform Face in the Intermediate Model Space

This step is carried out only when simplified face models are used. The target face model is deformed using Laplacian deformation based on the optimized vertex positions.

Since Laplacian coordinates embed geometric information of the local shape, it can preserve the local detail. The Laplacian of a shape is invariant to the locally linearized rigid transformations. The local geometry can be preserved as much as possible through reconstruction of global coordinates. In addition, the sparse linear system can be solved efficiently. The idea is that we deform the target neutral face model in the intermediate model space using Laplacian coordinates with deformed vertices being anchored. New vertices can be reconstructed based on Laplacian coordinates.

In the intermediate model space, the mesh model can be represented as $M = (V, E)$ with V being vertices and E the set of triangle edges.

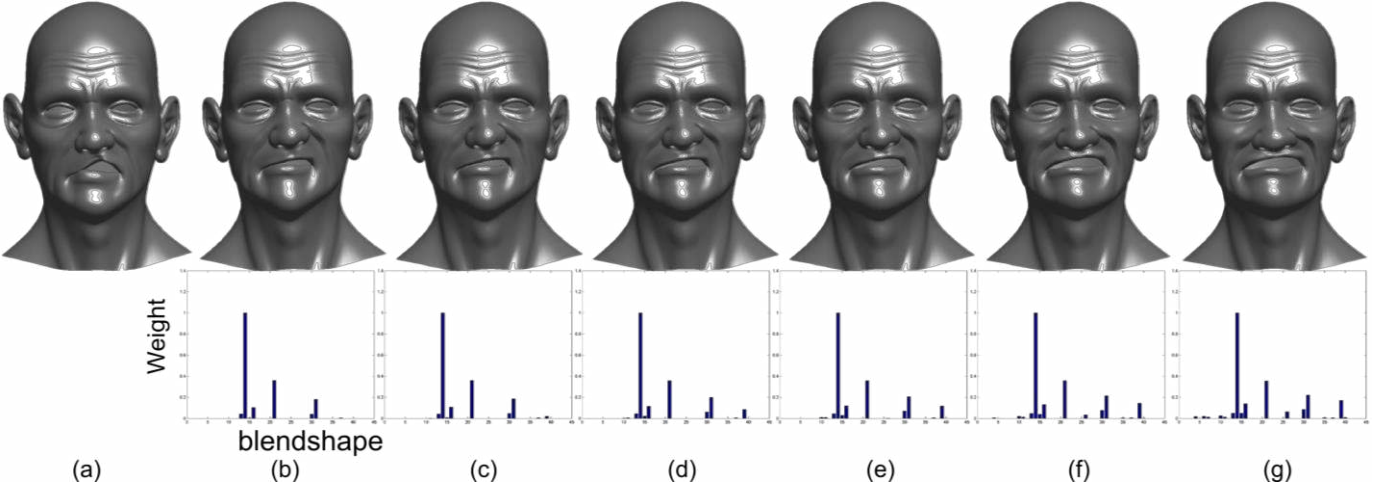


Fig. 5. Blendshape example – expression exp 1. The top row is the sketched face and optimized facial expressions. The bottom row is the blendshapes distribution corresponding to each facial expression above. For each blendshape figure, the horizontal axis is the selected blendshape – AUs. And the vertical axis is the weight values for those selected blendshapes. (a) The sketched facial expression. Facial expressions from (b) to (g) are the optimized facial expressions with $\lambda=1$. The control parameter α takes values 0, 5 10, 14, 20 and 25 respectively. The dominant blendshapes for this facial expression is blendshape 14 (AU10R – right upper lip raiser) and a small value of blendshape 21 (AU12 – the right lip corner puller). With the increase of α value the weight for some blendshape increase accordingly such as blendshape 16 (AU11L – Nasolabial Deepener) and 31 (AU20R – right lip stretcher) etc.

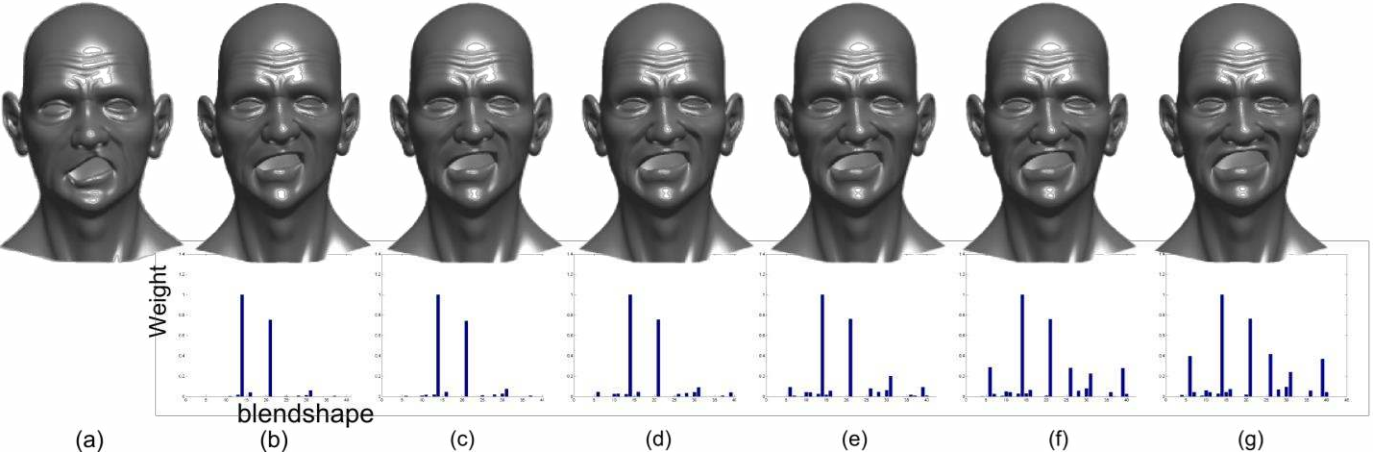


Fig. 6. Blendshape example - expression exp 2. The top row is the sketched face and optimized facial expressions. The bottom row is the blendshapes distribution corresponding to each facial expression above. For each blendshape figure, the horizontal axis is the selected blendshape – AUs. And the vertical axis is the weight values for those selected blendshapes. (a) The sketched facial expression by an artist with unrealistic expression. Figures from (b) to (g) are optimized facial expressions with control parameter values α being 0, 5 14, 20, 27 and 35 respectively. We set $\lambda=1$ for those cases. Facial expressions in (b), (c) and (d) mainly involve two blendshapes – blendshape 14 (AU10 – Upper Lip Raiser) and 21 (AU12 – Lip Corner Puller) with a small value of blendshape 31 (AU20 – Lip stretcher). Facial expressions in (d), (f) and (g) involve more blendshapes such as blendshape 6 (AU5 – Upper Lid Raiser) and 26 (AU15 – Lip Corner Depressor) etc. with the increase of α value.

Laplacian coordinates, denoted by $\delta_d = (\delta_{1d}, \delta_{2d}, \dots, \delta_{nd})^T, d \in \{x, y, z\}$ can be computed using the following equation $\delta_d = LV_d, d \in \{x, y, z\}$, where $V_d = (v_{1d}, v_{2d}, \dots, v_{nd})^T$ are vertex coordinates of the mesh model. L is the n by n Laplacian matrix with elements as follows:

$$L_{ij} = \begin{cases} 1, & i = j \\ -\alpha_{ij}, & (i, j) \\ 0, & \text{others} \end{cases} \quad (11)$$

This can be achieved through minimizing the quadratic energy:

$$E = \|L\tilde{V}_d - \delta_d\|^2 + \sum_{i=1}^m \alpha^2 \|\tilde{v}_{id} - v_{id}\|^2 \quad (12)$$

where \tilde{v}_d is the unknown deformed vertex set, \tilde{v}_{id} and v_{id} are optimized feature points and feature points on the neutral face model respectively. And α is a weighting parameter for feature points. The above quadratic energy minimization can be achieved by solving a linear system:

$$A\tilde{V} = b \quad (13)$$

Specifically, the above equation can be rewritten as follows:

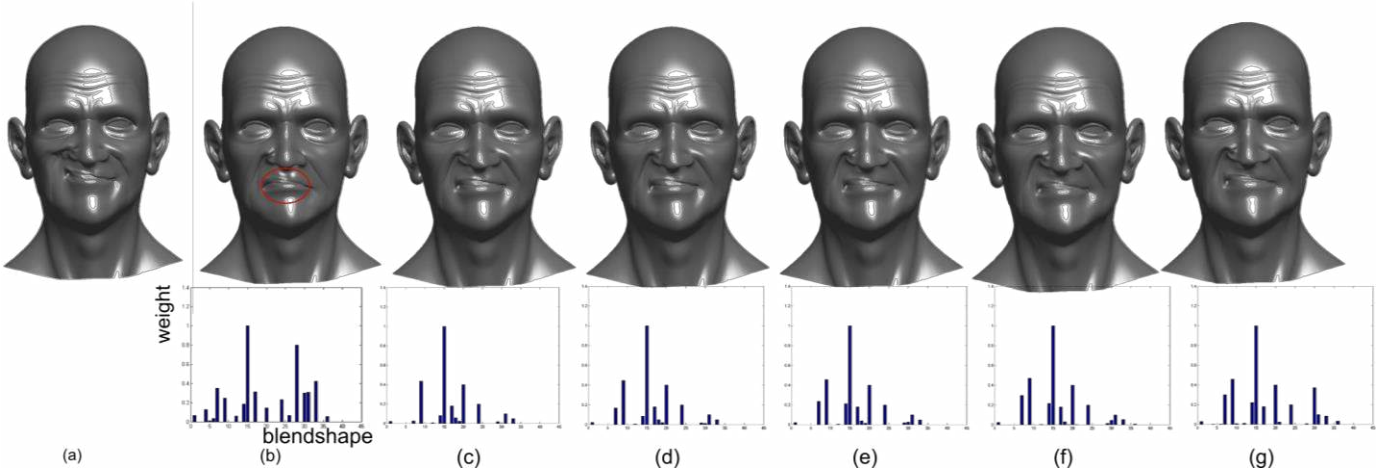


Fig. 7 Examples of extreme - facial expression exp 3 optimization with corresponding weights on the bottom row. (a) The sketched face with an extreme facial expression with very rough vertex positions around the mouth area. (b) optimized facial expression using equation (4) with over 10 blendshapes. The main blendshapes involved are blendshape 7 (AU6 – cheek raiser), 9 (AU6R), 15 (AU10R – right upper Lip Raiser), 17 (AU11R – right Nasolabial Deepener), 24 (AU14L – left Dimpler), 28 (AU17 – Chin Raiser), 30 (AU20L – left Lip stretcher), 31 (AU20R – right Lip stretcher); Figures from (c) to (g) show the optimized facial expressions using equation 9 with increasing α values from left to right: 0, 5, 14, 20, 27 and 35. The main blendshapes involved in the expression are blendshape 9 (AU6R), 15(AU10ROpen), 20(AU12L – left Lip Corner Puller) and 24(AU14L). Blendshape weight distribution changes slightly when adjusting the value of α . For example, the weights for blendshape 7 (AU6) and 30 (AU20L) increase slightly with the value of α , which results in a slightly stronger cheek raising and left lip stretch.

$$\begin{bmatrix} L \\ \alpha I_{m \times n} \end{bmatrix} V = \begin{bmatrix} \delta_d \\ \alpha V_d \end{bmatrix} \quad (14)$$

Since this is a $(n+m)$ by n , overdetermined linear system, it can be solved in the least square sense using its normal equation:

$$\tilde{V} = (A^T A)^{-1} A^T b \quad (15)$$

The weighting parameter is determined through experiments. In this paper, we set $\alpha = 0.5$ for all experiments. The full mesh deformation in the intermediate model space is then used to map to the target neutral face to get the final facial expression through calculating barycentric coordinates using the method proposed in [30].

The procedure for the optimization of sketched facial expression can be described in Algorithm 1.

Algorithm 1 The procedure for sketched facial expression optimization

1. Initialization. Set initial values $w_i = (0, 0, \dots, 0)$, $\varepsilon = 0.01$, $\alpha = 0$;
 2. Minimization. Update w_i by minimizing quadratic equation (9). This constrained linear optimization is solved by using reflective Newton method [23].
 3. Target shape deformation. The target shape is deformed based on the optimized landmark model using (12)
 4. Error evaluation. Compute the root-mean-square (RMS) error e_{rms} using (10).
 5. Termination. Loop α until $\alpha = 100$. We set 100 as the maximum value for the control parameter α .
-

VI. EXPERIMENT AND DISCUSSION

We have done a set of experiments using both full mesh and simplified mesh for several face models. Two ways have been used to generate the raw facial expressions. In the first case, we want to test raw facial expression with known AUs. We created some facial expressions by combining different AUs. And then we intentionally offset some vertices to create twists in the facial expressions. In this experiment, since AUs involved in the facial expression generation are known, we want to test whether the proposed method can accurately filter the right AUs. We tried not to completely change the AU selection during the approximation.

The experiments have been conducted in the MatlabR2012 environment on a computer with the specifications of CPU 2.6GHz and RAM 8.00GB. The average runtime is 1.52 seconds for each expression for the model used in this paper.

We asked artists to briefly sketch some facial expressions without further sculpting fine detail. Those sketched facial models represented rough facial expressions only with main features of the ideal expressions. A significant number of vertices, which represent detailed muscle movements for those expressions, have not been finely adjusted. Expressions shown from Fig. 5 to Fig. 7 are getting more complex with more AUs involved in the expression.

Fig. 5 shows a simple example of blendshape optimization for the sketched face from an artist. There is only one dominant blendshape - AU10L. It is very clear that with the increase of α , the weight of the dominant blendshape does not change much. But the weights for the auxiliary blendshapes have slightly changed. There are more blendshapes involved in the approximated facial expression from column 4 (Fig. 5) though with very small weight values. Fig. 8 shows the MSE errors for expressions in Fig. 5, Fig. 6 and Fig. 7 with different α values. The optimal α value is

the one with a minimal error calculated using equation (10). Through experiment, we find that there is no need to go

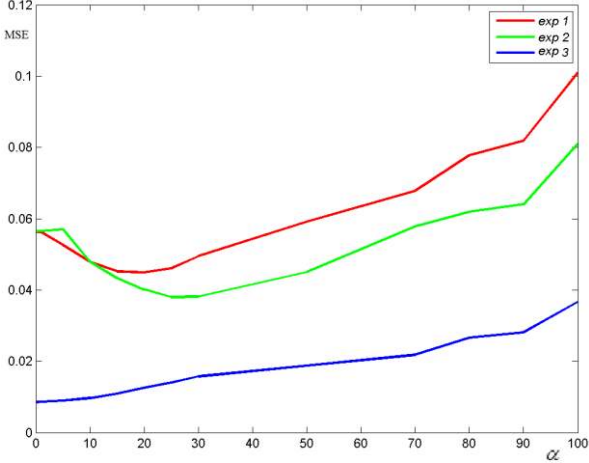


Fig. 8. MSE e_{rms} for three expressions in Fig. 5, Fig. 6 and Fig. 7.



Fig. 9 A close-up view of AU17 and AU23. (a) AU 17. (b) AU 23.

beyond 100 for α . Thus during the experiment we terminate the iteration at $\alpha = 100$. For exp 1 (Fig. 5), the facial expression in column 5 with $\alpha = 14$ has the minimum MSE. Thus, in terms of MSE, we can obtain the “best” regression facial expression model when $\alpha = 14$. For exp 2 (Fig. 6), the blendshape has the minimum MSE when $\alpha = 27$. This “best” facial expression just provides a suggestion for the user to make a decision. Fig. 7 shows an optimization example for an extreme facial expression. The sketched facial expression generated by the artist shows overstretched facial actions, which is beyond the normal muscle movements. In most cases, we want to avoid the unrealistic facial movements, though it is not an easy task to manually control the movements within the “boundary”. As it is demonstrated in Fig. 5, Fig. 6 and Fig. 7, the proposed optimization scheme can automatically maintain the vertex “boundary”. With the increasing of the α value, the blendshapes involved and their corresponding weights also change accordingly. However, those main blendshapes are not replaced by new blendshapes but by their weights changing slightly. With the increasing of α , those auxiliary blendshapes have a larger change than the main ones. The weight for blendshape 30 (AU20L) in (f) and (g) Fig. 7 is 0.4, but it is zero when $\alpha = 0$ in (b). Since AU20L is left lip stretcher on the corner, we can see that facial expression in (f) and (g) slightly stretches at the left lip corner compared with the one in column 1. And the weight for blendshape 7 (AU6) increases from nearly zero to 0.3 when α value increases from 0 to 35. There are about 18 blendshapes involved in the facial expression in (b) Fig. 7 optimized using equation (4), which

leads to an unrealistic regression result. As we can see from the mouth area in the figure – there is clear lip tightening on the lower lip caused by blendshape 28 (AU17 – Chin Raiser) and 33 (AU23 – Lip Tightener) with weight being 0.8 and 0.43 respectively. The illustration of these two AUs can be found in Fig. 9. This problem can be avoided using the proposed optimization method in equation (9). The result is illustrated in (c)-(g) in Fig. 7.

Fig. 10 shows an optimization example mainly involving eyebrow raising (AU4), lip corner puller (AU12), dimpler (AU14). Eyebrow raiser causes small movements compared with that around the mouth and cheek areas. Our method can effectively approximate these minor movements on the sketched face shown on the left block (bottom row) in Fig. 10. As shown in right block of Fig. 10, with the increase of α , the movements of AU4, AU12 and AU14 increases slightly. Fig. 11 demonstrates another example of eyebrow lower (AU4) and eyelid tightener (AU7). The proposed method can capture minor movements such as frown and eyelid tightening.

VII. CONCLUSION AND FUTURE WORK

In this paper, a method for facial expression optimization based on the regression concept has been presented. This technique has various potential applications such as real time human machine interaction, telecommunication and online avatars for mobile devices combining facial expression recognition techniques etc. [5, 8]. Through capturing rough positions of facial expression features of the end users, the proposed method could quickly reproduce the optimal facial expression model with high fidelity. Since the blendshapes are predefined on the local end, the reproduction is very efficient. We demonstrated the feasibility of the techniques for reproducing optimal facial expressions for a single frame. Dynamic facial expressions can be generated through iterating this procedure for the image sequence. Not only is the proposed scheme able to optimize expressions with wrong vertex positions but it also “compensates” fine muscle movements on the face model. Since the source AUs are recorded from real human actors, it is confirmed to capture expression wrinkles using a dense template mesh. The optimization is able to set a vertex “boundary” for each individual AU with the introduction of a regulating term. It effectively avoids the unrealistic regression. Since the optimization is carried out in the intermediate model space, it is flexible to use templates with different densities.

We have demonstrated the feasibility of the techniques for reproducing optimal facial expressions for a single frame. Hence, to generate dynamic facial expressions and give users more effective experience when interacting with computers, the dynamic facial expression model will also be investigated and the human age factor will even be taken into account [2] in the future work. Eye movements have not been experimented in the proposed method due to the limited 3D data captured for eyeballs. Since eye movements convey important emotional information, we also would like to incorporate this part in our future work.

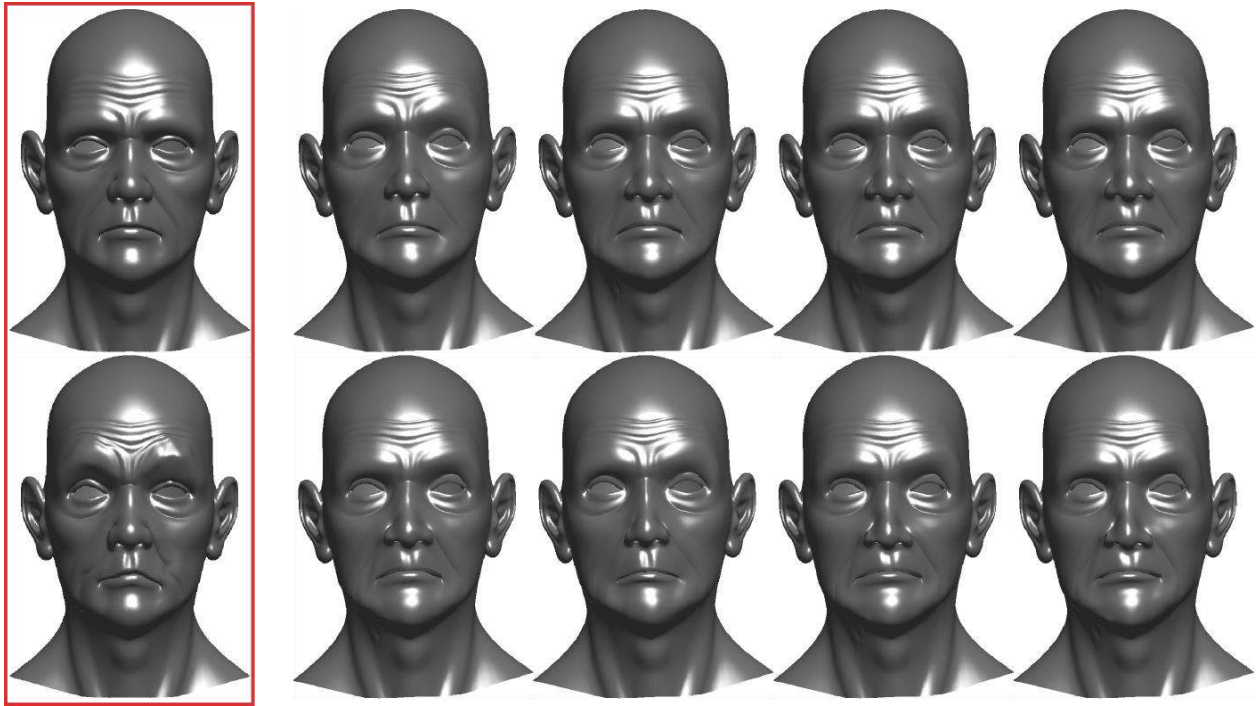


Fig.10 Optimization of minor eyebrow movements. This example mainly involves eyebrow raising (AU4), lip corner puller (AU12), dimpler (AU14). The left block shows the neutral face (top row) and the sketched face (bottom row). The facial movements involved in the sketched face mainly in the eyebrow area and upper lip and mouth corners. The right block illustrates the optimized expressions with parameter α increasing from left to right where α takes the value of 1, 3, 5 and 10 in the top row and 15, 20, 25 and 30 in the bottom row. The movements of AU4, AU12 and AU14 changes slightly with the increase of α as shown in the right block.

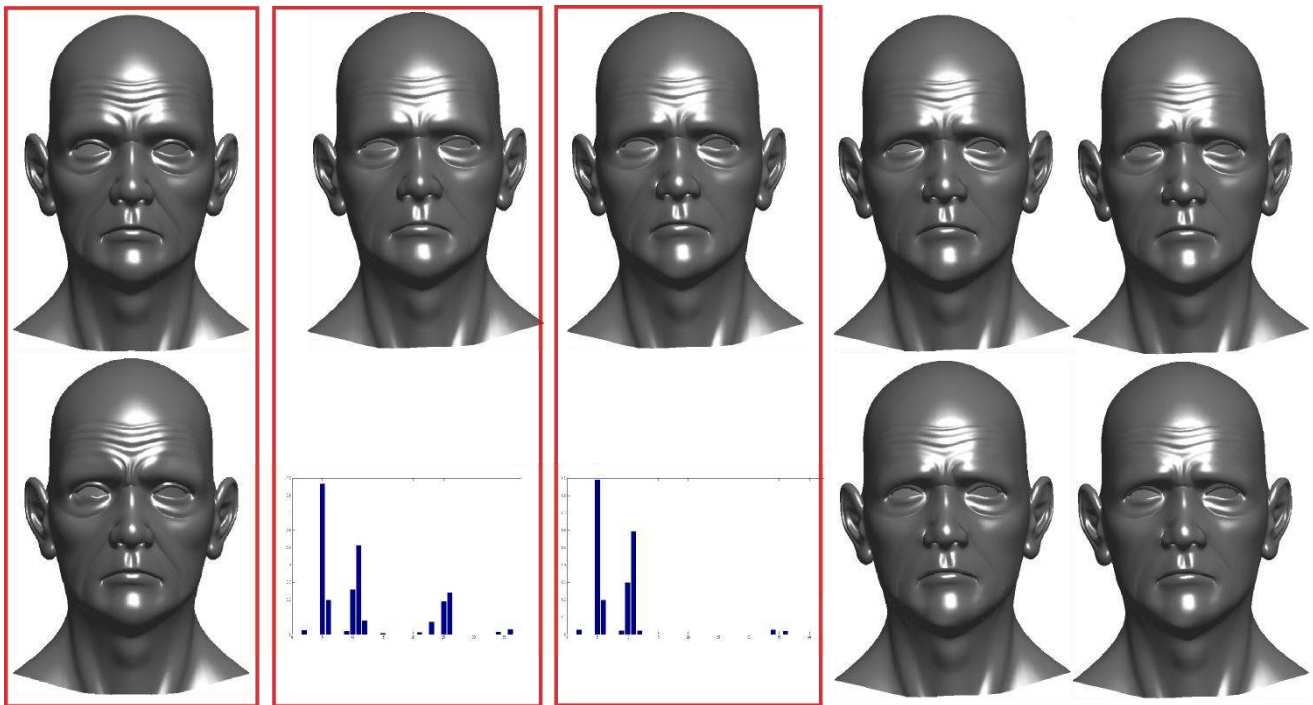


Fig. 11 Approximating frown expression. The left block in the red frame shows the neutral face (top row) and the sketched face (bottom row). The sketched face shows a rough frown expression with eyelid tightening. Block 2 and 3 show approximated facial expressions with $\alpha = 0$ and $\alpha = 1$ respectively. The bottom row of these two blocks shows the weights for these two expressions. Both expressions contain AU4 (brow lowerer), AU5 (upper lid raiser), AU7 (lid tightener) for left and right eyes. But two more unwanted AUs involved in the second block expression, AU14 (dimpler) and AU15 (lip corner depressor) with small values. The right two expressions in both rows were generated with α takes 5, 10, 20 and 25 respectively.

ACKNOWLEDGEMENT

The authors would like to thank Prof. Philippe Schyns, Dr. Rachael Jack and Dr. Oliver Garrod from University of Glasgow for allowing to access to the AU dataset. The authors also thank Dr. Brett Stevens with University of Portsmouth for his suggestions.

REFERENCES

- [1] Rincon-Nigro, M.; Zhigang Deng, "A Text-Driven Conversational Avatar Interface for Instant Messaging on Mobile Devices," *Human-Machine Systems, IEEE Transactions on* , vol.43, no.3, pp.328,332, May 2013.
- [2] Lu J. W; Tan Y.P, "Ordinary Preserving Manifold Analysis for Human Age and Head Pose Estimation," *Human-Machine Systems, IEEE Transactions on* , vol.43, no.2, pp.249,258, March 2013.
- [3] Sagar, M. And Grossman, R. 2006. Facial performance capture and expressive translation for King Kong. *In SIGGRAPH 2006 Sketches*.
- [4] Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., And Salesin, D. H. 1998. Synthesizing Realistic Facial Expressions from Photographs. *Proc. Of ACM SIGGRAPH'98*, 75–84.
- [5] Anderson, K.; McOwan, Peter W., "A real-time automated system for the recognition of human facial expressions," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* , vol.36, no.1, pp.96,105, Feb. 2006
- [6] Liu X., Mao T., Xia S, Yu Y. and Wang Z, Facial animation by optimized blendshapes from motion capture data, *Computer Animation And Virtual Worlds*, Vol 19, Issue 3-4, 2008, Pp: 235–245,
- [7] Ma W., Fyffe G, Debevec P., Optimized Local Blendshape Mapping for Facial Motion Retargeting, *SIGGRAPH 2011*, August 7 – 11, 2011. ISBN 978-1-4503-0921-9/11/0008
- [8] Medioni, G.; Jongmoo Choi; Cheng-Hao Kuo; Fidaleo, D., "Identifying Noncooperative Subjects at a Distance Using Face Images and Inferred Three-Dimensional Face Models," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* , vol.39, no.1, pp.12,24, Jan. 2009
- [9] Parke FI (1972) Computer Generated Animation Of Faces. In: *Proceedings Of ACM Annual Conference*. ACM, New York, Pp 451–457
- [10] Sagar, M. 2006. Facial performance capture and expressive translation for king kong. *In ACM SIGGRAPH 2006 Sketches*, 26.
- [11] Yu, H., Garrod, O., Jack, R., Schyns, P., Realistic Facial Expression Generation Based on Movement Mapping, *5th International Conference on Graphic and Image Processing*. Hong Kong, 2013
- [12] Joshi P., Tien W. C., Desbrun M., Pighin F.: Learning Controls For Blend Shape Based Realistic Facial Animation. *In Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 187–192. 1
- [13] Ekman P. & Friesen W.V. (1969). The repertoire of nonverbal behaviour: Categories, origins, usage, and coding. *Semiotica*, 1. pp. 49–98.
- [14] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. In: *Proceedings of the ACM SIGGRAPH*; 1999.
- [15] Besl PJ, McKay ND. A method for registration of 3D shapes. *IEEE Trans Pattern Anal Mach Intell* 1992;14(2):239–56.
- [16] Yu, H. Garrod O. and Schyns P., Perception-Driven Facial Expression Synthesis, *Computer and Graphics*, Vol. 36, Issue 3, May 2012, pp152–162.
- [17] Li Q., Deng Z.: Orthogonal-Blendshape-Based Editing System for Facial Motion Capture Data. *IEEE Computer Graphics and Applications* 28(6): 76–82 (2008).
- [18] Wojdel A. and Rothkrantz L. J. M., Parametric Generation of Facial Expressions Based on *Facs*, *Computer Graphics Forum*, Vol 24, Issue 4, 2005, Pp: 743–757
- [19] Bregler, C., Loeb, L., Chuang, E., and Deshpande, H. 2002. Turning To The Masters: Motion Capturing Cartoons. *ACM Trans. Graph.* 21,3, 399–407.
- [20] Barsi R and Jacobs D, "Lambertian Reflection and Linear Subspaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, pp. 218–233, 2003.
- [21] Pyun, H., Im K, Y., Hae C, W., Ang K, H. W., And Shin, S. Y. 2003. An example-based approach for facial expression cloning. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 167–176.
- [22] Deng, Z., C Hiang, P.-Y., F Ox, P., And Neumann, U. 2006. Animating blendshape faces by cross-mapping motion capture data. *In 13D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*. ACM, New York, NY, USA, 43–48.
- [23] Coleman, T.F. and Y. Li, "A Reflective Newton Method for Minimizing a Quadratic Function Subject to Bounds on Some of the Variables," *SIAM Journal on Optimization*, Vol. 6, Number 4, pp. 1040-1058, 1996F.
- [24] Chuang, E. And Bregler, C. 2002. Performance driven facial animation using blendshape interpolation. *Tech. rep.*, Department of Computer Science, Stanford University.
- [25] Bregler, C., L Oeb, L., C Huang, E., and Deshpande, H. 2002. Turning to the masters: motion capturing cartoons. *ACM Trans. Graph.* 21, 3, 399–407.
- [26] Choe, B., L Ee, H., And Seok Ko, H. 2001. Performance-driven musclebased facial animation. *The Journal of Visualization and Computer Animation* 12, 67–79.
- [27] Lewis, J. And Anjyo, K. 2010. Direct Manipulation Blendshapes. *IEEE Computer Graphics and Applications* 30, 4, 42–50.
- [28] Li , H., W Eise, T., And Pauly, M. 2010. Example-based facial rigging. *ACM Trans. Graph.* 29, 4, 1–6.
- [29] Lewis, J. P., Mooser, J., Deng, Z., And Neumann, U. 2005. Reducing Blendshape Interference By Selected Motion Attenuation. In *Proc. Of Acm Siggraph Symposium On Interactive 3D Graphics And Games (I3D)*, 25-29.
- [30] Sun Y.; Chen X.; Rosato, M.; Yin L., "Tracking Vertex Flow and Model Adaptation for Three-Dimensional Spatiotemporal Face Analysis," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* , vol.40, no.3, pp.461,474, May 2010
- [31] Choe, B. W., And Ko, H. S. 2001. Analysis And Synthesis Of Facial Expressions With Hand-Generated Muscle Actuation Basis. *In Ieee Computer Animation Conference*, 12–19