Towards Accurately Extracting Facial Expression Parameters

Xuecheng Liu†, Dinghuang Ji‡, Zhaoqi Wang and Shihong Xia‡
†Institute of Computing Technology, Chinese Academy of Sciences, China
Graduate School of the Chinese Academy of Sciences, China
KeXueYuan South Road 6, Haidian District, Beijing, China
†{liuxuecheng,jidinghuang}@ict.ac.cn
‡Institute of Computing Technology, Chinese Academy of Sciences, China
KeXueYuan South Road 6, Haidian District, Beijing, China
‡{zqwang,xsh}@ict.ac.cn

Abstract—Existing methods extract facial expression parameters from captured expressions data in two steps. First, head absolute orientation is acquired, the captured expression is then transferred from world coordinates to local coordinates. Second, expression parameters are extracted from local coordinate data. However, above methods result to severe error accumulation. In the second step, the error of head absolute orientation is amplified when extracting expression parameters, which results to error accumulation of the synthesized facial expression. In this paper, we propose an optimization-based parameters extraction method to prevent large error accumulation. In our optimization model, we use the error generated from synthesized and captured expressions as optimization objective. The head absolute orientation and expression parameters were regarded as optimization variables, which were simultaneously computed through optimization. The optimization scheme reduces error accumulation effectively. The experiments have shown that, in the case of linear blend-shape expression parameterization, the proposed method was able to promote the accuracy and efficiency relative to existing methods; in the case of nonlinear blendshape parameterization, our method was able to synthesize even higher accuracy results than the previous work.

Keywords— computer facial animation, expression capture, blendshape weights extraction, optimization, error accumulation

I. INTRODUCTION

In order to synthesize highly realistic facial animation, it is necessary to capture accurate and subtle expressions of true face and transfer it to the facial model. In general, the captured expression data can be expressed as world coordinates of markers on face surface. Taking passive optical motion capture system as example, a frame of captured expression data is composed of the world coordinates of markers on human face, as illustrated in Fig. 1.

The captured expression data is dependent on specific face and contains head motion. In order to transfer the captured expression to other feasible facial model, it is necessary to extract face-independent expressions parameters. Existing works extract facial expressions parameters through two steps. Firstly, head absolute orientation is extracted. The captured facial expression data is composed by world coordinates of facial markers, it is necessary to transform them to local face coordinates system. This could be easily done by attaching some markers which are relative fixed to the head as reference and then extracting head absolute orientation [1]. Above orientation is composed by head rotation matrix $R$ and transition vector $t$. Based on head orientation, the coordinates of face markers are transformed from world coordinates to local coordinates.

Then, facial expressions parameters are extracted. After getting expression data of facial local coordinates system, the captured expressions are transformed to face-independent expression parameters. In many existing works, this could be expressed as an optimization problem, as illustrated in Eq. 1.

$$
\min_{w} \left| \left( R^{-1} M - t - M_{neutral} - E(w) \right) \right|
\text{s.t. } 0 \leq w_i \leq 1
$$

(1)

In above optimization, $M$ is a frame of captured expression data, $R$ and $t$ is the rotation matrix and transition vector of facial local coordinates system, respectively. $M_{neutral}$ is the captured data of neutral facial expression. $w$ is facial

Fig. 1: Illustration of facial expression capture. The left image shows the infrared cameras arrangement. The right one shows optical markers distribution on actor’s face.
expression parameters vector. $E(w)$ is the synthesized facial expression according to parameters vector $w$. In this optimization, in order to avoid meaningless facial expression, $w$ is constraint in a close interval [0, 1]. In above optimization, the optimization solution method is dependent on the specific form of $E(w)$.

Through above two steps, the face-independent expression parameters are extracted from captured expression data. Given another face model, facial expressions are synthesized from parameters $w$ and head orientation $R$ and $t$, as illustrated in Fig. 2.

![Fig 2: Facial expression parameters extraction and expression synthesis. (a) extracting head orientation and expression parameters from captured expression data. (b) cloning captured expression onto another face model.](image)

Above two-steps method results to large error accumulation. Firstly, relative fixed markers results to error of computing head orientation. Then, above error will be amplified during computing expressions parameters $w$. The error accumulation results to the loss of precision from captured facial expression.

To address above issue, we propose a one-step optimized scheme. In our optimization model, the error transferred between captured expression and the synthesized one by expression parameters is set as objective. The optimization variables include head orientation (rotation matrix $R$ and transition vector $t$) and expression parameters $w$ which are simultaneously optimized. The proposed one-step method could reduce error accumulation.

In order to demonstrate the advantages of our method, we have applied our optimization model to linear and nonlinear blendshape facial animation methods. In the experiments, we contrast our method with existing ones in aspects of precision and efficiency of expression parameters extraction. Experiments have shown that, in the case of linear blendshape animation, our method promote the precision and efficiency. In the case of nonlinear one, we have promoted the precision of expression parameters extraction.

The remainder of this paper is organized as follows: Section 2 reviews related works. Section 3 illustrates our optimization scheme. Section 4 shows the experiments results. Finally, we conclude and discuss our work in section 5.

II. RELATED WORKS

The expression parameters extraction method is related to the specific parameterization method. In existing expression parameterization methods, linear blendshape method is widely used due to its intuition and convenience. In this section, we introduce related works of the linear and nonlinear blendshape methods, and then introduce the corresponding parameters extraction methods.

Since linear blendshape facial animation was firstly proposed by Parke’s pioneer work in [2] and [3], it has been widely used in both research and industry domains of facial animation. The linear blendshape method is illustrated in Eq. 2.

$$E(w) = \sum_{i=1}^{Nbs} w_i e_i, \quad 0 \leq w_i \leq 1$$

In Eq. 2, facial animation $E$ is synthesized as linear combination of basic expressions $e_i$, they are expressed as vectors of coordinates increments relative to the ones of neutral expression model. $Nbs$ represents the number of key shapes. $w_i$ is blending weight (expression parameters).

Principle components analysis (PCA) based methods construct basic shapes of facial animation through dimension reduction of facial expression samples. Its primary advantage is the rigid orthogonality of the constructed space. The disadvantage is that the principle components don’t possess visual intuition and are not suitable for manual manipulation. Chuang has thus designed three schemes that select key shapes from facial expression samples based on PCA results [4]; Li has used region-based PCA to automatically construct local orthogonal space from motion capture data [5].

Since facial action coding system (FACS) was firstly proposed by Ekman in [6], it has been very popular in facial animation. According to FACS, facial expressions can be presented as combinations of distinct Action Units. Each Action Unit intuitively corresponds to a basic shape of facial animation. In specific applications, reduced or modified versions of FACS also animate face well with improved usability [7] [8] [9] [10].

In light of the importance of facial animation, MPEG-4 has specified criterion of facial animation synthesis for network transmission[11]. In essence, MPEG-4 defines a system of linear blendshape facial animation. MPEG-4 has defined 68 parameters (FAP) to animate face. In these parameters, FAP1 and FAP2 depict fourteen static visemes and six basic facial expressions respectively, and the other FAPs defined 66 blending weights that enable synthesizing arbitrary facial expressions. In MPEG-4 facial animation, it is a vital issue that how to construct the facial animation table (FAT) that defines rules of facial motion. Kishiragar, Fratarcangeli and Jiang have made their efforts to explore FAT [12] [13] [14].

Besides that, other works also have strived to construct linear space of facial expressions. For example, Joshi has proposed an automatic physically-motivated scheme that segments blendshapes into smaller regions [15]. Cao and Shin have used independent components analysis (ICA) to extract a
set of meaningful parameters that were independent to each other as much as possible [16] [17].

Some methods are proposed to extract linear blendshape parameters from given expression samples. A usual method extracts parameters \( w_i \) through optimization [7] [18] [19] [9] [20] [21] [22], as illustrated in Eq. 3.

\[
\min_w \| (R^{-1}M - t - M_{neutral} - \sum_{i=1}^{N_b} w_i e_i) \| \\
\text{s.t. } 0 < w_i < 1
\]  

(3)

In Eq. 3, \( R \) and \( t \) is rotation matrix and transition vector of head respectively. \( M \) is an expression sample which is composed by facial markers coordinates of the world coordinates system. Similarly, \( M_{neutral} \) is the neutral expression data. In this optimization, the algorithm proposed by [23] is widely used.

Besides above optimization based methods, a learning based method is proposed by [24]. Firstly, the expression samples are transformed from world coordinates system to the local facial one. Secondly, some representative samples are selected out as training samples of learning algorithm. Then, the corresponding facial expressions are obtained by manually adjusting blending weights \( w_i \) and taking training samples as reference. After that, above blending weights \( w_i \) along with expressions samples are referred as training data of learning algorithm. At last, the radial basis functions (RBF) are trained to get the mapping relationship between expression parameters (\( w_i \)) and expression samples. This method needs massive manual work.

Other than linear blendshape method, other works use nonlinear blending of basic expressions to synthesize more realistic facial animation. The nonlinear blendshape method is illustrated in Eq. 4.

\[
E(w) = \sum_{i=1}^{N_b} f_i(w_i), \quad 0 \leq w_i \leq 1
\]  

(4)

Different from Eq. 2, in Eq. 4, linear blending function \( f_i(w_i) \) is substituted by nonlinear one \( f_i(w_i) \) to allow the greatest generality and fidelity of facial expressions.

In existing works, the cubic vector polynomials are used to express nonlinear blending functions [25] [17], as illustrated in Eq. 5. The constant items of cubic vector polynomials are left out because zero blending weights correspond with neutral facial expression.

\[
f_i(w_i) = a_i w_i^3 + b_i w_i^2 + c_i w_i
\]  

(5)

To extract the nonlinear blending weights \( w_i \) from given expression sample, the optimization method is used [25] [17], as illustrated in Eq. 6. In this optimization, a usual solution is proposed by [26].

\[
\min_w \| (R^{-1}M - t - M_{neutral} - \sum_{i=1}^{N_b} (a_i w_i^3 + b_i w_i^2 + c_i w_i)) \| \\
\text{s.t. } 0 < w_i < 1
\]  

(6)

To extract expression parameters through Eq. 3 and 6, the two steps method is used as illustrated in Section 1. The absolute orientation of head must be computed, it is followed by optimization. The two steps methods bring the error of computing head orientation into expression parameters extraction, which results to error accumulation.

To address above issue, this paper propose a one step method to prevent error accumulation. In our method, the head orientation and expression parameters are extracted simultaneously. The algorithm is illustrated in the following section.

III. EXPRESSION PARAMETERS EXTRACTION

A. Optimization model

In this paper, the proposed optimization model is illustrated in Eq. 7. In this model, the error between captured expression and the synthesized one is set as objective, the optimization variables includes head orientation (\( R \) and \( t \)) and expression parameters \( w \). Different from the model of Eq. 1, in our method, the head orientation is computed in optimization model.

\[
\min_{w,R,t} \| (R^{-1}M - t - M_{neutral} - E(w)) \| \\
\text{s.t. } 0 < w_i < 1
\]  

(7)

The inverse rotation matrix \( R^T \) has a sophisticated form, and above optimization model is difficult to solve. Therefore, we simplified it as follows.

B. Optimization simplification

We decompose rotation matrix \( R \) into two matrixes \( R_e \) and \( R_c \), which are estimation value and revision value of matrix \( R \) respectively. We compute the estimation matrix \( R_e \) through absolute orientation algorithm. It is done by taking the markers which are relatively fixed to local facial coordinates system as reference. For the revision rotation matrix \( R_c \), we set it as optimization variables. The transition vector \( t \) is similar decomposed into estimation vector \( t_e \) and revision vector \( t_c \). \( t_c \) is computed with absolute orientation algorithm and \( t_e \) is set as optimization variable. Based on above simplification, the optimization model is simplified as Eq. 8.

\[
\min_{w,R_e,t_e} \| (R_e^{-1}R_c^{-1}M - t_c - t_e - M_{neutral} - E(w)) \| \\
\text{s.t. } 0 \leq w_i \leq 1
\]  

(8)

In Eq. 8, we refer that the estimation matrix \( R_e \) is well estimated with small rotation error. Therefore, the revision rotation value \( R_c \) is matrix with tiny rotation. Then, we can simplify the inverse revision matrix \( R_c^{-1} \) as Eq. 9.

\[
R_c^{-1} \approx \begin{pmatrix}
1 & -\theta_z & \theta_y \\
\theta_z & 1 & -\theta_x \\
-\theta_y & \theta_x & 1
\end{pmatrix}
\]  

(9)
Based on above simplification, the optimization of Eq. 8 is transformed as the one of Eq. 10.

$$\min_{w, \theta_x, \theta_y, \theta_z, t_x, t_y, t_z} \left| (R_c^{-1} R_c^{-1} M - t_x - t_y - t_z - M_{\text{neutral}} - E(w)) \right|$$ \hspace{1cm} (10)

s.t. $0 \leq w_i \leq 1$

In Eq. 10, the optimization variables $\theta_x$, $\theta_y$, and $\theta_z$ are the angles of revision rotation $R_c$ respectively. $t_x$, $t_y$, and $t_z$ are revision transitions along x, y, and z axis respectively.

C. Optimization solution

In Eq. 10, the solution method of optimization model is relative to the specific form of expression parameterization method $E(w)$. Therefore, we applied the optimization model to linear and nonlinear blendshape methods which are usually employed to efficiently synthesize realistic facial animation. When applying to linear blendshape method, the optimization model is illustrated in Eq. 11.

$$\min_{w, \theta_x, \theta_y, \theta_z, t_x, t_y, t_z} \left| (R_c^{-1} R_c^{-1} M - t_x - t_y - t_z - M_{\text{neutral}} - \sum_{i=1}^{N_x} w_i \beta_i) \right|$$ \hspace{1cm} (11)

s.t. $0 \leq w_i \leq 1$

The optimization model of Eq. 11 is a bound constraint linear least square optimization. A common solution is proposed by [23]. The universal optimization model of [23] is illustrated in Eq. 12.

$$\min_{x} \sum_{j=1}^{N} F_j^2(x)$$ \hspace{1cm} (16)

s.t. $\lambda^u < x < \lambda^l$.

We transform our optimization model to the one of [26] through Eq. 17. Based on that, the optimization model is efficiently solved.

$$F_j(x) = d_j - \sum_{i=1}^{N_x} (a_{i,j} w_i^1 + b_{i,j} w_i^2 + c_{i,j} w_i);$$

$x = (\theta_x, \theta_y, \theta_z, t_x, t_y, t_z, w^f)$;

$x_{lb} = (\theta_{lb}, t_{lb}, 0, 0, 0, 0, 0)^T$;

$x_{ub} = (\theta_{ub}, t_{ub}, 1, 1, 1, 1, 1)^T$;

$d = (R_c^{-1} R_c^{-1} M - t_x - t_y - t_z - M_{\text{neutral}})$.

In Eq. 17, $a_{ij}$, $b_{ij}$ and $c_{ij}$ is the $j$th component of vector polynomials coefficients $a_i$, $b_i$ and $c_i$.

IV. EXPERIMENTS

We show the experiment results of our methods in this section. We have designed two experiments. In the one experiment, we applied our method into linear blendshape facial expression synthesis (Eq. 11). Then, we contrasted it with existing method (Eq. 3), which is widely used in existing works [7][8][9][10][11][12]. In the other experiment, we have applied our method into nonlinear blendshape expression synthesis (Eq. 15), and then contrasted it with existing work (Eq. 6), which is used in [25][17].

In this experiment, we have captured 10 clips of facial motion data (more than 38,000 frames of facial expressions). These expressions samples were able to cover the facial expression space. The main hardware of this experiment is 2.67G CPU and 2G RAM. The main software is Matlab 7.3.

To evaluate the experiment results of different methods, we have taken two quantitative measures as references. One measure is precision of the synthesized facial expression, which was the most important factor to synthesize realistic facial animation. Another one is computing efficiency of different method due to the importance of real-time request in many applications.

To the precision, we firstly got each frame error between the synthesized expression and the captured one. Then, we average the errors of each clip of facial motion and each marker. After that, we got the absolute and relative error of each clip of data.
To the efficiency, we firstly got iterations and computing time of each frame optimization solving. Then, we average them to each clip of data. The iterations of optimization solving is independent of hardware, therefore is suitable to evaluate the efficiency of different methods. The frame rate is the most obvious measure of real-time computing.

A. Linear blendshape experiment

For the linear blendshape application, we contrast our method with a representative existing one (Eq.11 v.s. Eq.3).

We have solved the optimizations of different methods by [23]. Based on the optimization solution, we firstly got the absolute error of each frame through Eq. 18. Then, we got the relative error through Eq. 19. At last, we average above errors on each clip of data, and it was set as the measure of precision of different expression synthesis methods. Above errors were shown in Fig. 3.

\[
| (\mathbf{R}^{-1} \mathbf{M} - \mathbf{t} - \mathbf{M}_{\text{neutral}}) - \sum_{i=1}^{N_{hs}} w_i \mathbf{e}_i | \tag{18}
\]

\[
\frac{| (\mathbf{R}^{-1} \mathbf{M} - \mathbf{t} - \mathbf{M}_{\text{neutral}}) - \sum_{i=1}^{N_{hs}} w_i \mathbf{e}_i |}{| (\mathbf{R}^{-1} \mathbf{M} - \mathbf{t} - \mathbf{M}_{\text{neutral}}) |} \tag{19}
\]

From Fig. 3, we can see that, the synthesis error of our method is less than the one of existing work [9]. That is, our method is able to synthesize more realistic facial animation relative to existing method.

We have also demonstrated the expressions respectively synthesized by different methods. A pair of representative results of facial expressions was shown in Fig. 4. More results were shown in Fig. 5.

![Figure 3: Contrast of expression synthesis precision through different methods (linear blendshape application). (a) The absolute error contrast. (b) The relative error contrast](image)

![Figure 4: contrast of synthesized expressions respectively by our method and [9]. In (a) and (b), the left image is synthesized by [9], the right one is synthesized by our method, the middle image is synchronized video of facial motion capture. The red spheres represent motion capture data, the green ones represent synthesized expressions.](image)
Figure 5: More results contrast of synthesized expressions respectively by our method and [9]. In (a) and (b), the second line images are synthesized by [9], the bottom ones are synthesized by our method, the top images are synchronized video of facial motion capture. The red spheres represent motion capture data, the green ones represent synthesized expressions.

From above synthesis precision, we have also contrasted the computing efficiency of different methods. On the one hand, we have contrasted the average iterations of optimization solving. The results are shown in Fig. 6a. On the other hand, we have contrasted the average frame rates, and they were demonstrated in Fig. 6b.

From above experiment, we can see that, in linear blendshape application, our method is superior to existing work [9]. On the one hand, our method is able to synthesize more realistic facial expressions, the synthesis precision increases 12.3% relative to [9] (the absolute error decreases 11.0%). On the other hand, the computing efficiency increases 86.0% relative to [9], and our method is able to perform real-time facial animation (about 85.8f/s).

B. Nonlinear blendshape experiment

In the nonlinear blendshape application, we contrast our method with a representative work (Eq.15 VS. Eq.6).

We have solved the optimizations of different methods by [26]. Based on the optimization solution, we firstly got the absolute error of each frame through Eq. 20. Then, we got the relative error through Eq. 21. At last, we averaged above errors on each clip of data, and it was set as the measure of precision of different expression synthesis methods. Above errors were shown in Fig. 7.

<table>
<thead>
<tr>
<th>Facial animation clip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute error (mm)</td>
</tr>
<tr>
<td>[LXFW11]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Facial animation clip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative error (%)</td>
</tr>
<tr>
<td>[LXFW11]</td>
</tr>
</tbody>
</table>

Figure 7: Contrast of expression synthesis precision through different methods (nonlinear blendshape application). (a) The absolute error contrast. (b) The relative error contrast.

From figure 7, we can see that, the synthesis error of our method is less than the one of existing work [25]. That is, our method is able to synthesize more realistic facial animation relative to existing method.

We have also demonstrated the expressions respectively synthesized by different methods. A pair of representative results of facial expressions was shown in figure 8. More results were shown in Fig. 9.

From above synthesis precision, we have also contrasted the computing efficiency of different methods. On the one hand, we have contrasted the average iterations of optimization solving. The results are shown in Fig. 10a. On the other hand, we have contrasted the average frame rates, and they were demonstrated in Fig. 10b.

<table>
<thead>
<tr>
<th>Facial animation clip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame rates (fps)</td>
</tr>
<tr>
<td>[Hav06]</td>
</tr>
</tbody>
</table>

Figure 6; Contrast of computing efficiency of different methods (linear blendshape application). (a) Contrast of average iteration of optimization solving. (b) Contrast of average frame rates.
Figure 8: contrast of synthesized expressions respectively by our method and [25]. In (a) and (b), the left image is synthesized by [25], the right one is synthesized by our method, the middle image is synchronized video of facial motion capture. The red spheres represent motion capture data, the green ones represent synthesized expressions.

Figure 9: More results contrast of synthesized expressions respectively by our method and [25]. In (a) and (b), the second line images are synthesized by [25], the bottom ones are synthesized by our method, the top images are synchronized video of facial motion capture. The red spheres represent motion capture data, the green ones represent synthesized expressions.

Figure 10: Contrast of computing efficiency of different methods (nonlinear blendshape application). (a) Contrast of average iteration of optimization solving. (b) Contrast of average frame rates.

From above experiment, we can draw the conclusion as follows. On the one hand, our method is able to synthesize more realistic facial expressions, the synthesis precision increases 23.5% relative to [25] (the absolute error decreases 19.0%). On the other hand, although the computing efficiency decreases 18.2% relative to [25], our method is able to perform real-time facial animation (about 34.6f/s).

V. CONCLUSION

Existing works extract facial expression parameters from captured expressions data through two steps method. However, this scheme results to heavy error accumulation. In this paper, we have propose a more accurate method of parameters extraction to prevent error accumulation.

The experiments have shown that, when applying to linear blendshape expression parameterization, the proposed method was able to promote the accuracy and efficiency relative to existing methods. When applying to nonlinear blendshape parameterization, our method was able to synthesize more accuracy results than the previous work. The conclusion is quantitatively shown in Table 1.

<table>
<thead>
<tr>
<th>Application</th>
<th>Methods</th>
<th>Precision pro motion</th>
<th>Efficiency pro motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear blendshape</td>
<td>Our Method VS. [9]</td>
<td>+12.3%</td>
<td>+86.0%</td>
</tr>
<tr>
<td>Nonlinear blendshape</td>
<td>Our Method VS. [25]</td>
<td>+23.5%</td>
<td>-18.2%</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

This work is supported and funded by National Natural Science Foundation of China Grant Nos. 60970086, NSFC-Guangdong Joint Fund (U0935003) and National Science and Technology Key Projects 2012ZX03002005. Also, we would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES