

Research Article

An Improved Surface Simplification Method for Facial Expression Animation Based on Homogeneous Coordinate Transformation Matrix and Maximum Shape Operator

Juin-Ling Tseng

Department of Computer Science and Information Engineering, Minghsin University of Science and Technology, No. 1, Xinxing Road, Xinfeng Township, Hsinchu County 304, Taiwan

Correspondence should be addressed to Juin-Ling Tseng; flysun@must.edu.tw

Received 28 September 2015; Revised 2 January 2016; Accepted 10 January 2016

Academic Editor: Fazal M. Mahomed

Copyright © 2016 Juin-Ling Tseng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Facial animation is one of the most popular 3D animation topics researched in recent years. However, when using facial animation, a 3D facial animation model has to be stored. This 3D facial animation model requires many triangles to accurately describe and demonstrate facial expression animation because the face often presents a number of different expressions. Consequently, the costs associated with facial animation have increased rapidly. In an effort to reduce storage costs, researchers have sought to simplify 3D animation models using techniques such as Deformation Sensitive Decimation and Feature Edge Quadric. The studies conducted have examined the problems in the homogeneity of the local coordinate system between different expression models and in the retainment of simplified model characteristics. This paper proposes a method that applies Homogeneous Coordinate Transformation Matrix to solve the problem of homogeneity of the local coordinate system and Maximum Shape Operator to detect shape changes in facial animation so as to properly preserve the features of facial expressions. Further, root mean square error and perceived quality error are used to compare the errors generated by different simplification methods in experiments. Experimental results show that, compared with Deformation Sensitive Decimation and Feature Edge Quadric, our method can not only reduce the errors caused by simplification of facial animation, but also retain more facial features.

1. Introduction

Concomitant with the rapid advancements in information technology, the application of 3D animation has become increasingly popular in fields such as the movie industry [1, 2], gaming [3–5], arts [6, 7], and education [8, 9]. Among the various 3D animation technologies, facial animation is the most commonly used technique. However, because facial animation contains a variety of expressional contents, the surface relief of 3D models in facial animation undergoes considerable changes.

Parke [10] was the first to propose the concept establishing facial animation. Subsequently, many related studies and techniques, including Head Pose Estimation [11], Facial 3D Shape Estimation [12], Head Shop [13], The Digital Emily Project [14], Automatic Generation [15], Kinect-Based Facial Animation [16], Real-Time Facial Animation [17–21], and 3D Facial Similarity Measure [22], emerged.

However, in order to elaborate facial animation, numerous triangles are needed, which significantly increases storage costs. Thus, in order to reduce the number of triangles needed to describe the face animation, many experts and scholars have proposed simplified 3D animation methods, including Deformation Sensitive Decimation (DSD) [23, 24], Feature Edge Quadric (FEQ) [25], Facial Features Region Partition [26], and MPEG-4 Quadric-based LoD Simplification [27]. However, most of them are derived from QSlim [28], proposed by Garland and Heckbert, and DSD [23], proposed by Mohr and Gleicher.

QSlim [28] is one of the most famous 3D model simplification methods [29]. This method can not only execute rapidly, but also reduce the errors caused by model simplification. Unfortunately, QSlim can only simplify 3D static models, whereas 3D animation typically contains various frame models. Consequently, in order to utilize QSlim for

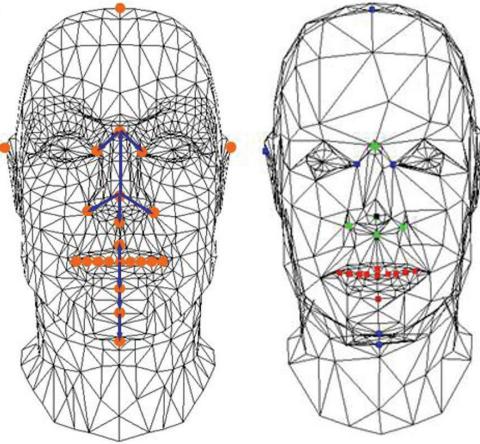


FIGURE 1: Simplification results using Feature Edge Quadric [25].

simplification of 3D animation, Mohr and Gleicher proposed the DSD method [23]. The method calculates and aggregates the error matrix of each vertex in different frame models to serve as a basis for simplification. However, DSD can cause destruction of the appearance characteristics of the model because it lacks complete homogeneous coordinate transformation.

In order to preserve the facial expression features when simplifying face animation, Kim et al. [25] established 32 feature points and adopted FEQ to preserve facial animation features, as shown in Figure 1; Wang et al. [26] used Facial Features Region Partition to set a scope on the face, which is intended to preserve the desired facial features. Its simplification results are shown in Figure 2. However, in these methods, the feature regions must be set by the user before the shapes of the regions are preserved. In other words, these methods can only preserve general facial features, such as the eyes and nose. For nongeneral features, such as wrinkles, it is most likely that they are not in the set range and their triangles can easily be removed, resulting in damage to the features of the nonsetting regions.

In order to overcome this limitation of feature regions having to be set by users and to preserve the facial features during the simplification process, this paper proposes a DSD-based method that uses Homogeneous Coordinate Transformation Matrix (HCTM) to solve DSD's homogeneous coordinate transformation problems and uses Maximum Shape Operator (MSO) to estimate the changes in each vertex point in the various frame models. In this way, the intensity values of each facial characteristic are automatically quantified and the characteristics of facial animation are moderately preserved during the simplification process.

2. Related Works

2.1. Deformation Sensitive Decimation (DSD). QSLIM [28] is a Quadric Error Metrics- (QEMs-) based 3D model

simplification method. This method first calculates the K_f matrix of the adjacent triangular piece of each vertex: $f : ax + by + cz + d = 0$. In this formula, $a^2 + b^2 + c^2 = 1$ and the matrix is based on the calculation of the distance of any point v from the f plane, $(f^T v)^2 = v^T (ff^T) v = v^T K_f v$. The sum of distances of any point to its adjacent triangle is computed by adding all the K_f matrices of all adjacent triangular patches. This sum of distances, which can be obtained using formula (1), is the error matrix values resulting from the simplification of vertex pairs. In the QSLIM process, the simplification is performed in an orderly manner from the vertex pair of the lowest error to the highest, as shown in Figure 3. Consequently, this method can efficiently generate a simplified low error model as follows:

$$\text{Error}(v) = v^T \cdot Q \cdot v. \quad (1)$$

In this formula,

$$Q = \sum_{f \in \text{faces}(v)} K_f \quad (2)$$

$$K_f = f \cdot f^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}.$$

QSLIM is utilized to simplify 3D animation models by DSD [23]. This method mainly calculates the matrix $Q_{v,q}$ of each vertex in different frame models of animation, as shown in formula (3), to estimate the errors each vertex pair (v_1, v_2) produces during the simplification process and then to decide which vertex pair should be simplified first, as shown in formula (4). This method inherits the speedy execution of QSLIM but lacks complete homogeneous coordinate transformation [30]. Consequently, it cannot generate the best vertex point after the simplification of vertex pairs, as illustrated in Figures 4 and 5. Consider

$$\text{Error}(v) = \sum_{q=1}^k v^T \cdot Q_{v,q} \cdot v = v^T \cdot \left(\sum_{q=1}^k Q_{v,q} \right) \cdot v, \quad (3)$$

$$\sum_{q=1}^k \tilde{v}_q^T \cdot (Q_{v_1,q} + Q_{v_2,q}) \cdot \tilde{v}_q, \quad (4)$$

where k is the number of frame models in 3D animation; \tilde{v}_q refers to the new vertices, v_1 and v_2 , generated in the q th frame model after simplification; and $Q_{v_1,q}$ and $Q_{v_2,q}$ are the respective error matrices of v_1 and v_2 in the overall 3D animation model.

In addition, the DSD simplification method is not aimed at analyzing the surface of the 3D animation model. Consequently, it is relatively easy for the important features of the face to be damaged during the facial animation simplification process. To solve this problem, Kim et al. proposed Feature Edge Quadric method with the aim of defining the important feature regions of the face and further preserving the main facial animation features during the simplification process.

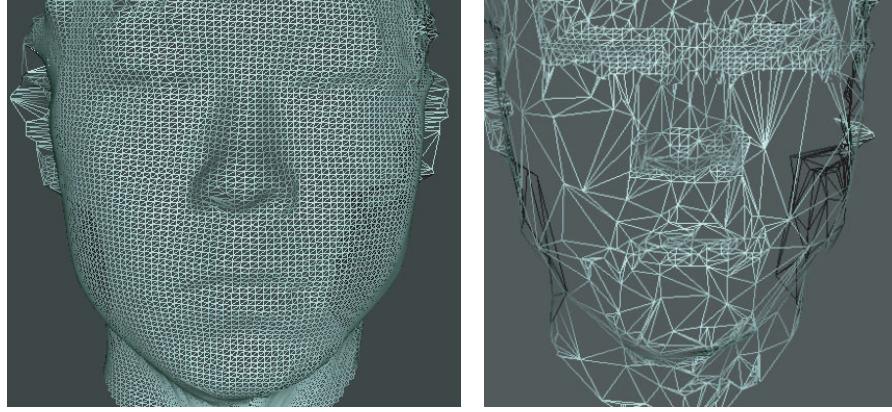


FIGURE 2: Simplification results using Facial Features Region Partition [26].

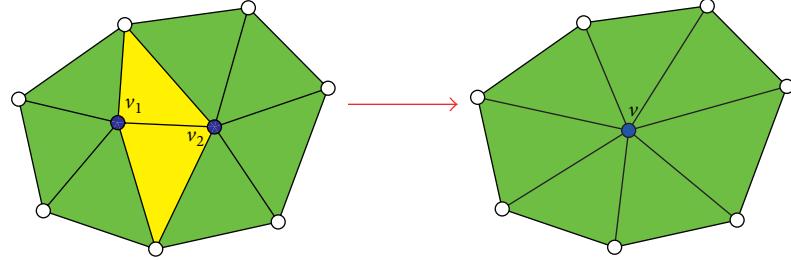


FIGURE 3: Surface simplification using Quadric Error Metrics [28].

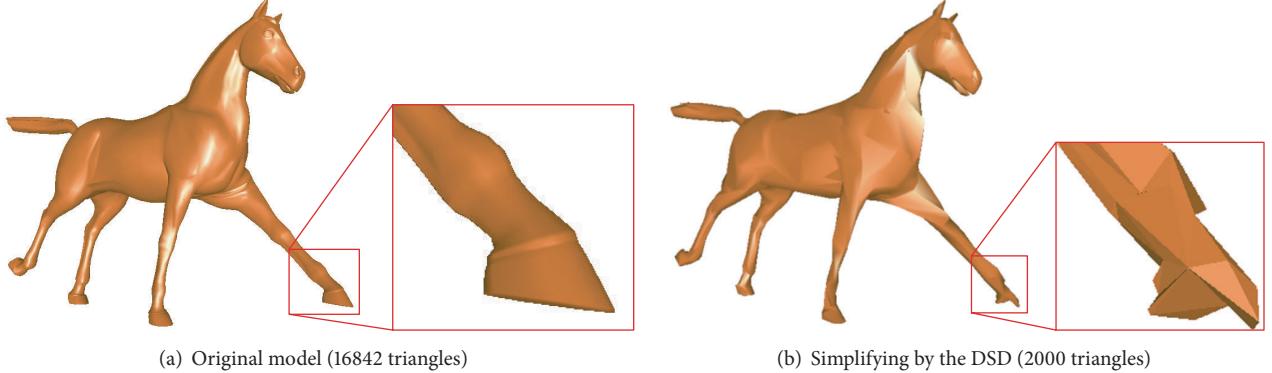


FIGURE 4: The shape of horse hoof is destroyed gradually when simplifying by the DSD [30].

2.2. Feature Edge Quadric (FEQ). FEQ was proposed by Kim et al. for preservation of facial features. In the proposed method, the feature points of 10 parts in the face, including tip of the nose, top of the head, left/right side of the face, top/bottom of the nose, left/right eye socket, bottom left/right of the nose, lip contact line, top/bottom of the lip, chin, and throat, are set as shown in Figure 1. For error estimation, this method contains basic quadric error and Feature Edge Quadric. The former is based on QSlim and calculates the error matrix generated by each vertex pair (v_1, v_2) during the simplification process, as shown in the following formula:

$$Q_p = \sum_{f \in nghb(v_1)} \text{area}(f) \cdot Q^f + \sum_{f \in nghb(v_2)} \text{area}(f) \cdot Q^f, \quad (5)$$

where $nghb(v_1)$ is the set of triangles adjacent to v_1 , $nghb(v_2)$ is the set of triangles adjacent to v_2 , $\text{area}(f)$ is the area of triangle f , and Q_f is the error matrix of triangle f .

In the latter, the error matrix of the adjacent feature edges of each feature point is calculated to properly preserve feature edges during the facial animation simplification process.

To calculate the error matrix of feature edge e_i , FEQ first computes the planes m_e , which are orthogonal to the edge and can be calculated by e_i and the average normal vector of its two adjacent triangles, e_i^n , as shown in the following formula:

$$m_e = e_i \times e_i^n, \quad (6)$$

where $e_i^n = (n_1^f + n_2^f)/2$, where n_1^f and n_2^f are the normal vectors of two triangles adjacent to e_i .

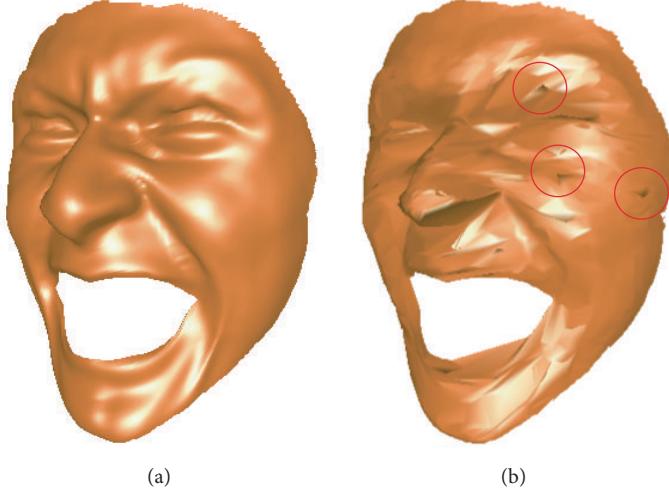


FIGURE 5: DSD cannot generate the best vertex point after the simplification of vertex pairs. (a) Original model (57835 triangles); (b) simplifying by the DSD (5000 triangles).

Through the integration of basic quadric error and Feature Edge Quadric, the formula of the error matrix can be deduced, as shown in

$$Q^v(v) = (1 - \alpha) \cdot Q_p + \alpha \cdot Q_F, \quad (7)$$

where $Q_F = \sum_{e \in fe(v)} (m_e) \cdot (m_e)^T$, where $fe(v)$ is the set formed by the adjacent feature edge of vertex point v , and α is the weight value (0-1) customized by users (0.5 by default).

Although the FEQ simplification method aims to preserve important features of the eyes, nose, mouth, and so forth, there are many more facial features than these organ features. The forehead, cheeks, and other areas also change significantly with changes in facial expressions. However, FEQ easily ignores the expression features and causes a rapid increase in the simplified errors during the process of simplification. Therefore, our paper proposes a novel simplification method that integrates MSO and HCTM and properly takes into account the overall facial features, thereby reducing errors generated during the simplification of facial animation.

3. Homogeneous Coordinate Transformation Matrix (HCTM)

Because 3D facial animation is composed of various different expression models, the tangent plane and normal vector of

the same vertex point also vary with changes in expressions, as shown in Figure 6. In other words, the local coordinate system for the same vertex point varies in different frame models. Therefore, if the coordinate transformation system is not accurate, errors can be easily caused by the simplification of the facial animation models. For this reason, the DSD cannot accurately calculate the optimal vertex position after the simplification of vertex pairs. This paper adopts Theorems 1 and 2 and proposes HCTM to solve this problem.

Theorem 1. *The Homogeneous Coordinate Transformation Matrix $T(d)$ that corresponds to a translation by $d = (d_x, d_y, d_z)$ is given by the following formula [31]:*

$$T(d) = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

Theorem 2. *The Homogeneous Coordinate Transformation Matrix $R(a, \theta)$ that corresponds to a rotation of θ about the axis in the direction of the unit vector $a = (a_x, a_y, a_z)$ is given by the following formula [31]:*

$$R(a, \theta) = \begin{bmatrix} a_x^2 + c_\theta(1 - a_x^2) & a_x a_y (1 - c_\theta) - a_z s_\theta & a_x a_z (1 - c_\theta) + a_y s_\theta & 0 \\ a_x a_y (1 - c_\theta) + a_z s_\theta & a_y^2 + c_\theta(1 - a_y^2) & a_y a_z (1 - c_\theta) - a_x s_\theta & 0 \\ a_x a_z (1 - c_\theta) - a_y s_\theta & a_y a_z (1 - c_\theta) + a_x s_\theta & a_z^2 + c_\theta(1 - a_z^2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

where $c_\theta = \cos \theta$ and $s_\theta = \sin \theta$.

In this paper, HCTM sets the local coordinate space for each vertex point in the first expression model of the facial animation as the main coordinate space. Then, the vertex points in all other expression models are transformed into the corresponding local coordinate space in the first expression model.

Suppose that $v_{p,q}$ is the vertex point of v_p in frame q ; $M_{p,q}$ is the Mesh formed by the adjacent vertex points of $v_{p,q}$; and $t_{p,q}$ and $n_{p,q}$ are the tangent plane and normal vector in $v_{p,q}$. HCTM must convert $M_{p,q}$ from its original local coordinate system to the local coordinate system in the first expression model. This process contains two steps: translation of $M_{p,q}$ to the coordinate system that takes $v_{p,1}$ as the origin and $t_{p,1}$ and $n_{p,1}$ as the tangent plane and normal vector and rotation of the tangent plane $t_{p,q}$ and normal vector $n_{p,q}$ to overlap with $t_{p,1}$ and $n_{p,1}$. The transformation matrix in step one can be defined by Theorem 1 as follows:

$$T(\Delta v) = \begin{bmatrix} 1 & 0 & 0 & \Delta v_x \\ 0 & 1 & 0 & \Delta v_y \\ 0 & 0 & 1 & \Delta v_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

In this formula,

$$\Delta v = v_{p,1} - v_{p,q} = [\Delta v_x \ \Delta v_y \ \Delta v_z \ 1]^T. \quad (11)$$

The transformation matrix in step two can be defined by Theorem 2 as in formula (12). In this formula, $n = n_{p,q} \times n_{p,1} = [n_x \ n_y \ n_z \ 1]^T$ and θ is the angle included between $n_{p,q}$ and $n_{p,1}$. Consider

$$R(n, \theta) = \begin{bmatrix} n_x^2 + c_\theta(1 - n_x^2) & n_x n_y (1 - c_\theta) - n_z s_\theta & n_x n_z (1 - c_\theta) + n_y s_\theta & 0 \\ n_x n_y (1 - c_\theta) + n_z s_\theta & n_y^2 + c_\theta(1 - n_y^2) & n_y n_z (1 - c_\theta) - n_x s_\theta & 0 \\ n_x n_z (1 - c_\theta) - n_y s_\theta & n_y n_z (1 - c_\theta) + n_x s_\theta & n_z^2 + c_\theta(1 - n_z^2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (12)$$

Integrating formulas (10) and (12), HCTM can be defined, as is shown in (13). Mesh $M_{p,q}$ and the error matrix $Q_{v_p,q}$ can

be converted from one to the other with (13), as is shown in formulas (14) and (15) as follows:

$$H(n, \theta, \Delta v) = R(n, \theta) \times T(\Delta v) = \begin{bmatrix} n_x^2 + c_\theta(1 - n_x^2) & n_x n_y (1 - c_\theta) - n_z s_\theta & n_x n_z (1 - c_\theta) + n_y s_\theta & 0 \\ n_x n_y (1 - c_\theta) + n_z s_\theta & n_y^2 + c_\theta(1 - n_y^2) & n_y n_z (1 - c_\theta) - n_x s_\theta & 0 \\ n_x n_z (1 - c_\theta) - n_y s_\theta & n_y n_z (1 - c_\theta) + n_x s_\theta & n_z^2 + c_\theta(1 - n_z^2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & \Delta v_x \\ 0 & 1 & 0 & \Delta v_y \\ 0 & 0 & 1 & \Delta v_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ = \begin{bmatrix} n_x^2 + c_\theta(1 - n_x^2) & n_x n_y (1 - c_\theta) - n_z s_\theta & n_x n_z (1 - c_\theta) + n_y s_\theta & \Delta v_x n_x^2 + \Delta v_x c_\theta(1 - n_x^2) + \Delta v_y n_x n_y (1 - c_\theta) - \Delta v_y n_z s_\theta + \Delta v_z n_x n_z (1 - c_\theta) + \Delta v_z n_y s_\theta \\ n_x n_y (1 - c_\theta) + n_z s_\theta & n_y^2 + c_\theta(1 - n_y^2) & n_y n_z (1 - c_\theta) - n_x s_\theta & \Delta v_x n_x n_y (1 - c_\theta) + \Delta v_x n_z s_\theta + \Delta v_y n_y^2 + \Delta v_y c_\theta(1 - n_y^2) + \Delta v_z n_y n_z (1 - c_\theta) - \Delta v_z n_x s_\theta \\ n_x n_z (1 - c_\theta) - n_y s_\theta & n_y n_z (1 - c_\theta) + n_x s_\theta & n_z^2 + c_\theta(1 - n_z^2) & \Delta v_x n_x n_z (1 - c_\theta) - \Delta v_x n_y s_\theta + \Delta v_y n_y n_z (1 - c_\theta) + \Delta v_y n_x s_\theta + \Delta v_z n_z^2 + \Delta v_z c_\theta(1 - n_z^2) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (13)$$

$$M'_{p,q} = H(n, \theta, \Delta v) \times M_{p,q}, \quad (14)$$

$$Q'_{v_p,q} = H(n, \theta, \Delta v) \times Q_{v_p,q}. \quad (15)$$

With HCTM, the same Mesh M_p in different expression models can be converted to the local coordinate system in the first expression model, and its error calculation can be revised using formula (3) defined by DSD, as shown in formula (16). This method can solve the error value increase problem resulting from incomplete homogeneous coordinate transformation in DSD as follows:

$$\text{Error}(v) = v^T \cdot \left(\sum_{q=1}^k (H(n, \theta, \Delta v) \cdot Q_{v_p,q}) \right) \cdot v. \quad (16)$$

4. Maximum Shape Operator (MSO)

To detect the shape features of the facial animation model, this paper presents a shape operator that estimates the changes in the 3D surface. In general, shape operator is mainly used to calculate the shape change of the specific tangent direction of the 3D object's surface, as in Definition 3. However, in this paper, the shape change of the 3D object's surface, on which Mesh lies, is estimated by the edge of the Mesh.

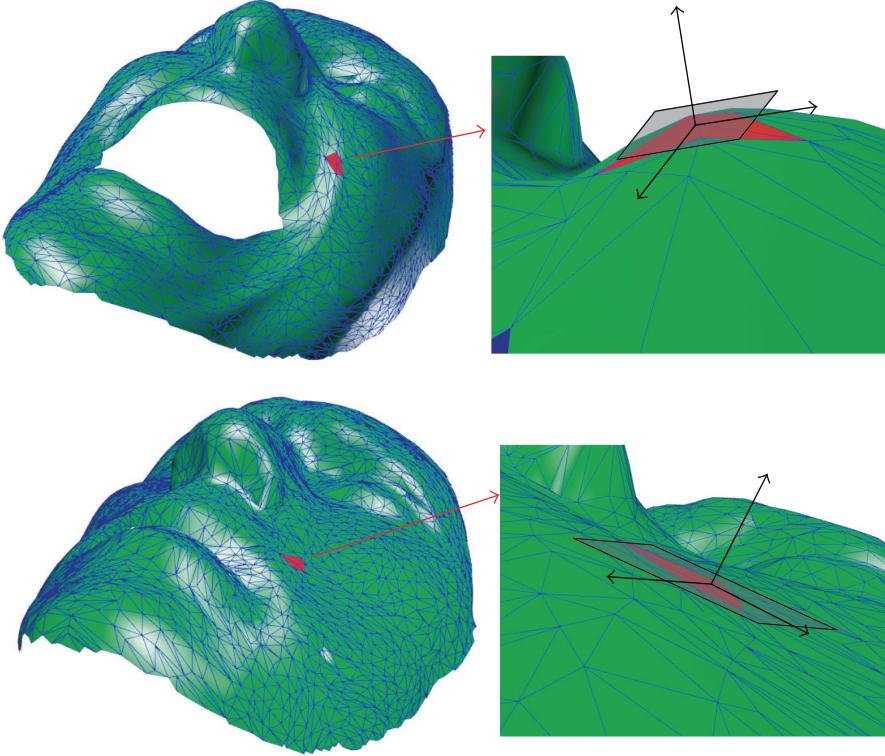


FIGURE 6: The local coordinate system for the same vertex point varies in different expression models.

Definition 3. Let $M \subset R^3$ be a regular surface, and let U be a surface normal to M defined in a neighborhood of a point $\tilde{v} \in M$. For a tangent vector t to M at \tilde{v} , we put

$$S(t) = -D_t U. \quad (17)$$

Then, S is called the *shape operator* [32, 33].

Shape operator refers to the change degree of the normal vector field of surface M 's vertex point \tilde{v} in its tangent direction t . To calculate the adjacent surface changes of the vertex points in the 3D model, in this paper, shape operator is extended to estimate and integrate the shape operator values of the vertex point \tilde{v} and its adjacent vertex points. Suppose that the number of adjacent vertex points of vertex point \tilde{v} is m , called $\tilde{v}_1, \tilde{v}_2, \tilde{v}_3, \dots, \tilde{v}_m$, and its tangent vector to each vertex point is, respectively, t_1, t_2, t_3, \dots , and t_m , as shown in Figure 7. Consequently, $S_{\tilde{v}}(t_1), S_{\tilde{v}}(t_2), S_{\tilde{v}}(t_3), \dots, S_{\tilde{v}}(t_m)$, which signify the shape operator value of vertex \tilde{v} in the t_1, t_2, t_3, \dots , and t_m directions can be generated according to the definition of shape operator, as shown in formula (18). After integrating the shape operator in each of these different tangent directions, the surface change of the region in which the vertex lies is shown in formula (19) as follows:

$$S_{\tilde{v}}(t_i) = \frac{\|U(\tilde{v}_i) - U(\tilde{v})\|}{\|\tilde{v}_i - \tilde{v}\|}, \quad (18)$$

$$\begin{aligned} S_{\tilde{v}} &= \frac{\sum_{i=1}^m (\|\tilde{v}_i - \tilde{v}\| \times S_{\tilde{v}}(t_i))}{\sum_{i=1}^m \|\tilde{v}_i - \tilde{v}\|} \\ &= \frac{\sum_{i=1}^m \|U(\tilde{v}_i) - U(\tilde{v})\|}{\sum_{i=1}^m \|\tilde{v}_i - \tilde{v}\|}. \end{aligned} \quad (19)$$

Shape operator can automatically analyze the surface changes of the facial expression model and extract the important features, such as eyes, nose, and mouth, as shown in Figure 8. This method covers the drawback in FEQ of each feature region having to be defined by the user. However, facial animation is not static. As time passes, the face may produce various facial features, including nasolabial folds and forehead wrinkles. Therefore, in order to preserve facial expression features during simplification of the facial animation, this paper uses MSO to extract the facial animation feature. This method calculates the shape operator value of each vertex point v in k (the number of expression models) expression models and takes the maximum value as the eigenvalue for appearance analysis of facial animation. The MSO is defined as follows:

$$S_v = \max_{1 \leq i \leq k} S_{v,i}, \quad (20)$$

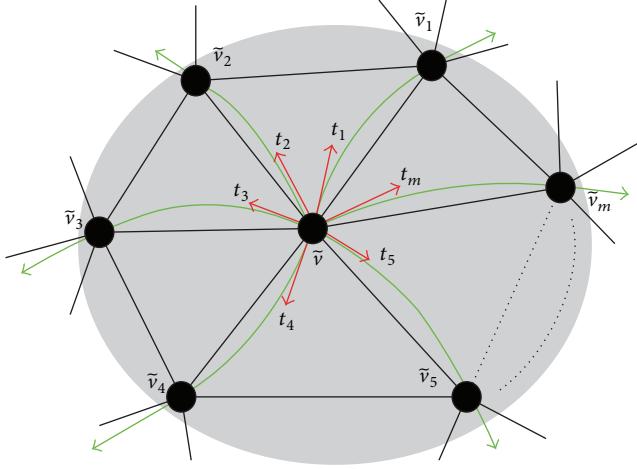


FIGURE 7: Shape operators of point p using the tangent vectors to each neighboring point to estimate local surface variation.

where $S_{v,i}$ is the shape operator value of vertex v in frame i . Figure 9 shows the results of the surface changes of the face animation extracted by MSO. It can be seen from the result that nasolabial folds are obvious in expressions of surprise, grin, laugh, and smile, but not obvious in expressions of fury, anger, rage, and sadness. However, if the features of those expressions are ignored during the face animation simplification process, a rapid increase in simplification errors can easily occur. With MSO, the proposed method can extract the features of nasolabial folds to reduce the errors caused by the simplification of the facial animation.

5. The Proposed Algorithm

The algorithm proposed in this paper is mainly based on DSD, with the introduction of MSO and HCTM to reduce the errors caused by the simplification of the face animation model. The main steps in this algorithm are as follows.

- (1) Calculate the QEM $Q_{v_p,q}$ of vertex v_p in each facial expression model:

$$Q_{v_p,q} = \sum_{f \in \text{faces}(v_p,q)} K_f, \quad (21)$$

where $\text{faces}(v_p,q)$ is the set of triangles adjacent to v_p,q in facial expression model q and K_f is the matrix formed in plane f , as is shown in formula (1).

- (2) Calculate HCTM $H(n, \theta, \Delta\nu)$ and update QEM $Q_{v_p,q}$ to obtain the following:

$$Q'_{v_p,q} = H(n, \theta, \Delta\nu) \times Q_{v_p,q}. \quad (22)$$

- (3) Estimate the MSO S_{v_p} of each vertex point v_p :

$$\begin{aligned} S_{v_p} &= \max_{1 \leq q \leq k} S_{v_p,q}, \\ S_{v_p,q} &= \frac{\sum_{i=1}^m \|U(\tilde{v}_i) - U(v_{p,q})\|}{\sum_{i=1}^m \|\tilde{v}_i - v_{p,q}\|}, \end{aligned} \quad (23)$$

where m is the number of vertex points adjacent to $v_{p,q}$ in facial expression model q , \tilde{v}_i is the vertex point adjacent to $v_{p,q}$ in facial expression model q , and $S_{v_p,q}$ is the shape operator of $v_{p,q}$.

- (4) Sum all QEMs of v_p and import MSO into the QEM:

$$\begin{aligned} Q''_{v_p} &= \sum_{q=1}^k Q'_{v_p,q} + S'_{v_p} \times \sum_{q=1}^k Q'_{v_p,q} \\ &= \left(1 + S'_{v_p}\right) \times \sum_{q=1}^k \left(H(n, \theta, \Delta\nu) \times Q_{v_p,q}\right), \end{aligned} \quad (24)$$

$$S'_{v_p} = \begin{cases} 0, & S_{v_p} \leq 0, \\ \frac{S_{v_p}}{(\mu + 3\sigma)}, & 0 < S_{v_p} < \mu + 3\sigma, \\ 1, & S_{v_p} \geq \mu + 3\sigma, \end{cases}$$

where S_{v_p} is the MSO value of v_p , μ is the mean value of MSO, and σ is the standard deviation of MSO. The original MSO value is between zero and 1,241.77, which is scattered and has some outliers. In order to balance the effect of QEM $Q'_{v_p,q}$ and MSO, in this paper, the MSO is standardized to [0, 1], and the outliers that are higher than $S_{v_p}/(\mu + 3\sigma)$ are set to the maximum value of one.

- (5) Calculate the minimum simplification error of each vertex pair with the following conditions:

- (a) (v_a, v_b) is an edge; or
- (b) (v_a, v_b) is not an edge, but its distance $\|v_a - v_b\|$ is smaller than the threshold value set by users.

- (6) Choose the vertex pair with the smallest error in step (5) for simplification.
- (7) Simplify the vertex pair (v_a, v_b) into a vertex point v , and update the QEM Q''_v of v , where $Q''_v = Q''_{v_a} + Q''_{v_b}$.
- (8) Update all the information adjacent to the vertex points v_a or v_b .
- (9) Repeat steps (5) to (8) until the number of triangles has been reduced to the designated value.

6. Experimental Results

This study used an Intel Core 2.2 GHz CPU with 1 GB RAM as the main execution environment. The eight experimental

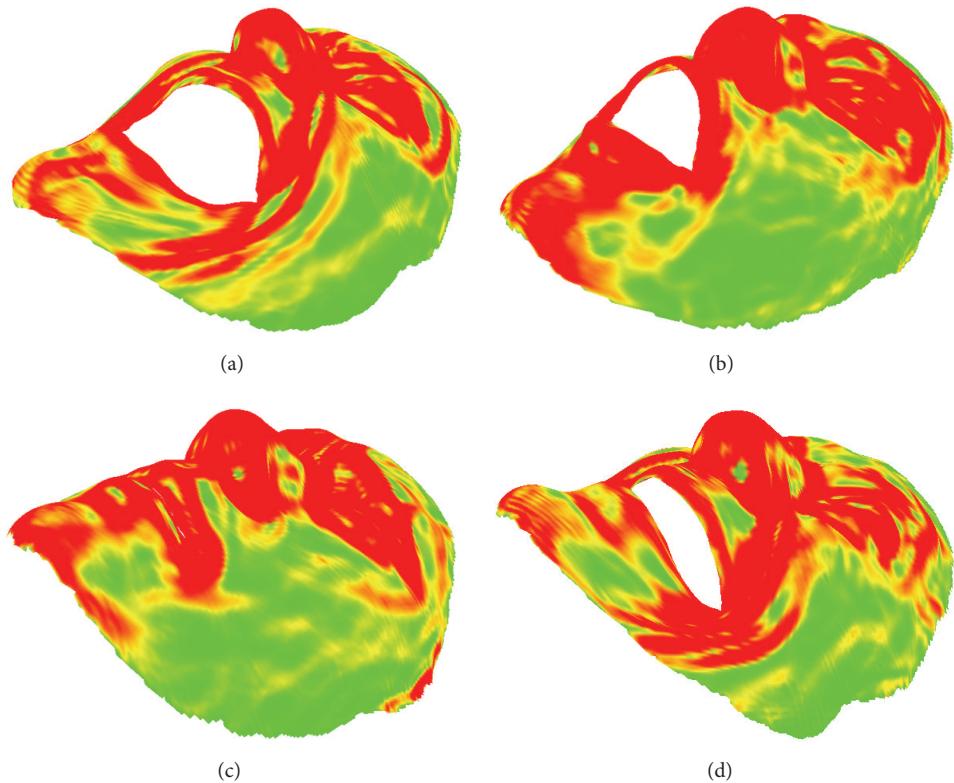


FIGURE 8: Shape operator can automatically analyze the surface changes of the facial expression model. (Red indicates the high-variation surfaces. Yellow indicates medium-variation regions. Green represents low-variation areas.)

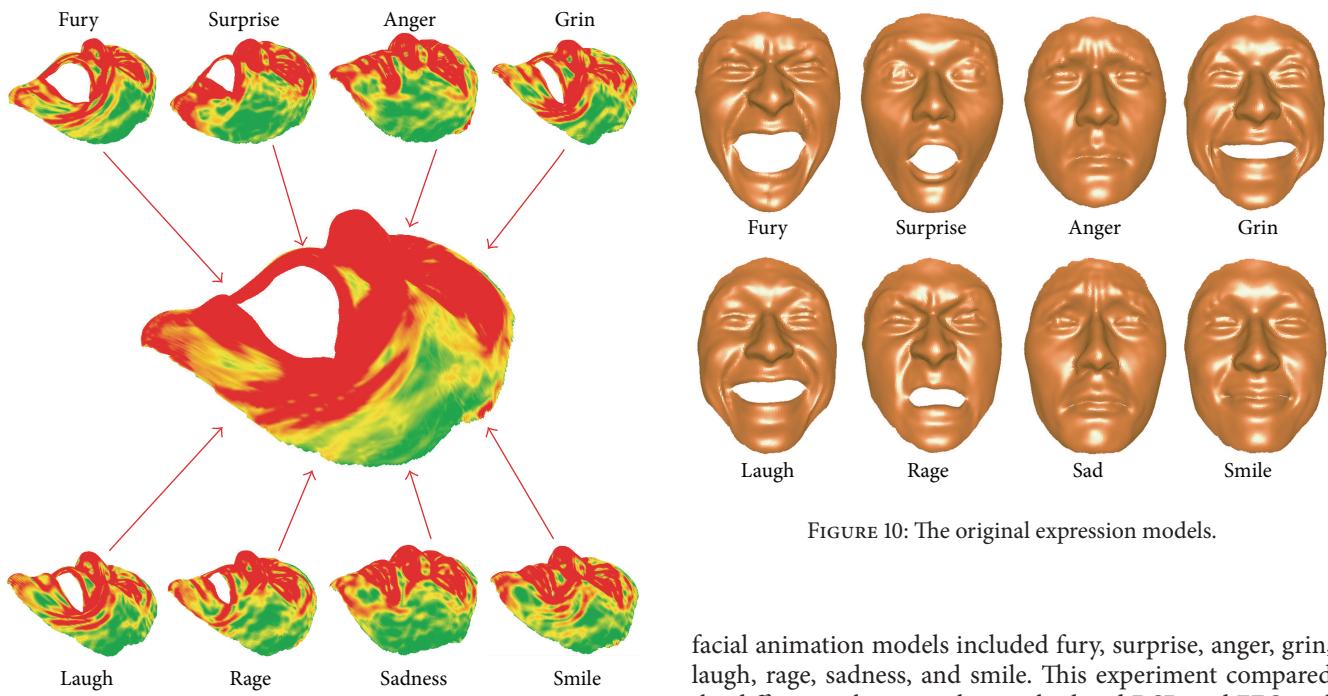


FIGURE 9: This method calculates the shape operator value of each vertex point in every expression model and takes the maximum value as the eigenvalue for appearance analysis of facial animation.

facial animation models included fury, surprise, anger, grin, laugh, rage, sadness, and smile. This experiment compared the differences between this method and DSD and FEQ and used root mean square (RMS) error and perceived quality to analyze the distortion errors caused by these methods so as to verify the superiority of this method over DSD and FEQ.

TABLE 1: RMS error comparison between our method and DSD (unit: 10^{-2}).

Triangles	The DSD	Our method	Improvement rate
Face 01 (fury)			
5000	5.7313	0.8427	85.30%
2000	11.3899	1.6240	85.74%
1000	22.9495	2.6013	88.67%
500	42.7859	4.7140	88.98%
Face 02 (surprise)			
5000	4.8805	1.4011	71.29%
2000	9.6882	2.7791	71.31%
1000	21.3109	4.2362	80.12%
500	37.8799	6.6230	82.52%
Face 03 (anger)			
5000	4.6126	1.1371	75.35%
2000	9.1516	2.3843	73.95%
1000	19.3625	3.9031	79.84%
500	35.8760	6.6498	81.46%
Face 04 (grin)			
5000	5.2081	1.3187	74.68%
2000	10.2029	2.4012	76.47%
1000	21.2078	3.8791	81.71%
500	39.2217	6.6697	82.99%
Face 05 (laugh)			
5000	4.8700	1.2308	74.73%
2000	10.1936	2.4448	76.02%
1000	21.3471	3.8759	81.84%
500	39.3825	6.5277	83.42%
Face 06 (rage)			
5000	5.3178	1.1113	79.10%
2000	10.2902	2.3494	77.17%
1000	21.0917	3.6617	82.64%
500	37.7458	5.6601	85.00%
Face 07 (sad)			
5000	4.6088	1.4588	68.35%
2000	9.9824	2.9477	70.47%
1000	21.0213	4.4211	78.97%
500	38.0176	7.1686	81.14%
Face 08 (smile)			
5000	4.6887	1.1734	74.97%
2000	9.8362	2.4347	75.25%
1000	21.4011	3.7970	82.26%
500	39.8993	5.9372	85.12%

The eight original expression models adopted in this experiment are shown in Figure 10. Each model had 29,299 vertex points and 57,836 triangles. When the number of triangles in the facial animation model is reduced to 5,000, 2,000, 1,000, and 500 through DSD, the generated RMS error values are shown in Table 1.

With changes in the expression model, the error caused by DSD and this method will be different. Take face 01 (fury) as an example: when the number of triangles in the facial

TABLE 2: Average RMS error comparison between our method and DSD (unit: 10^{-2}).

Triangles	The DSD	Our method	Improvement rate
5000	4.9897	1.2092	75.77%
2000	10.0919	2.4207	76.01%
1000	21.2115	3.7969	82.10%
500	38.8511	6.2438	83.93%

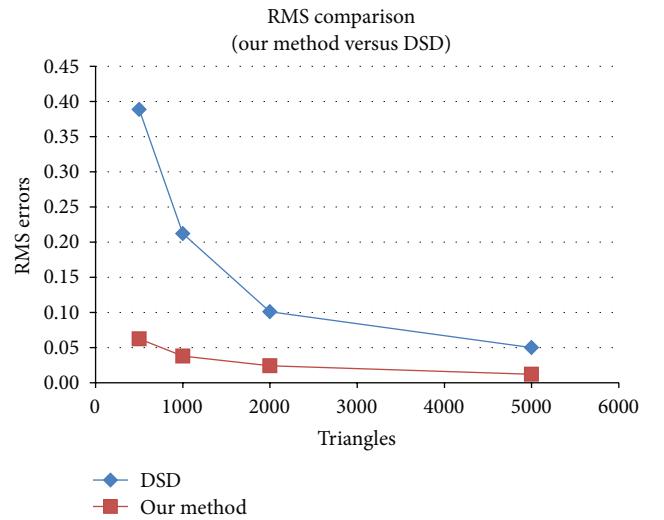


FIGURE 11: RMS error comparison between our method and DSD.

animation model is reduced to 5,000, 2,000, 1,000, and 500 through DSD, the generated RMS error values are 5.7313×10^{-2} , 11.3899×10^{-2} , 22.9495×10^{-2} , and 42.7859×10^{-2} , respectively. However, the error can be reduced to 0.8427×10^{-2} , 1.6240×10^{-2} , 2.6013×10^{-2} , and 4.7140×10^{-2} using this method, resulting in an improvement rate of between 85.30% and 88.98%. For other expression models, this method is better than the DSD method, with an improvement rate of more than 68.35%. In addition, with the reduction in the number of triangles, the improvement rate is increasingly better. Take face 01 (fury) as an example: when the number of triangles is reduced to 5,000, the improvement rate of this method is 85.30% compared with the DSD; and when the number is 500, the improvement rate is 88.98%. In other words, this method is better than DSD in low triangle numbers and can retain more facial shapes than DSD.

The average errors generated by all the facial expressions in the facial animation are shown in Table 2 and Figure 11. The table shows the errors generated from simplifying the entire 3D facial animation using DSD and this method, respectively. In Table 2, it can be seen that when the number of triangles is simplified to 5,000, 2,000, 1,000, and 500, the errors generated by DSD are 4.9897×10^{-2} , 10.0919×10^{-2} , 21.2115×10^{-2} , and 38.8511×10^{-2} , whereas those of this method are 1.2092×10^{-2} , 2.4207×10^{-2} , 3.7969×10^{-2} , and 6.2438×10^{-2} , which are improved by 75.77% to 83.93% compared with DSD.

In addition to the use of QEM for simplification like DSD, FEQ also estimates the simplification errors of important

TABLE 3: Average RMS error comparison among our method, FEQ, and DSD (unit: 10^{-2}).

Triangles	The DSD	The FEQ	Our method
5000	4.9897	1.2658×10^{-2}	1.2092×10^{-2}
2000	10.0919	2.7134×10^{-2}	2.4207×10^{-2}
1000	21.2115	5.2143×10^{-2}	3.7969×10^{-2}
500	38.8511	10.6767×10^{-2}	6.2438×10^{-2}

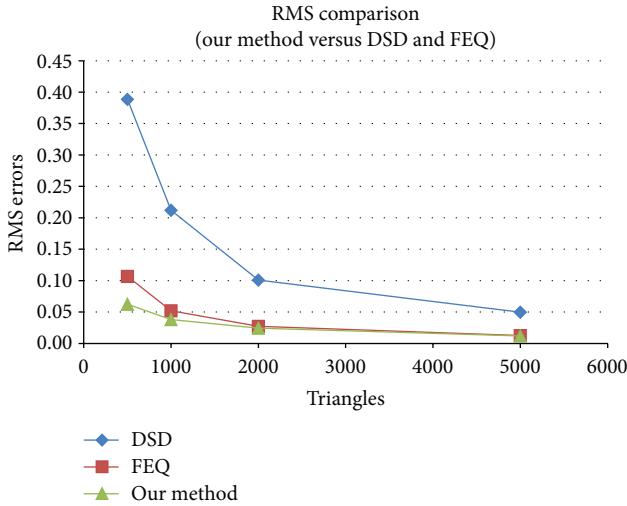


FIGURE 12: Average RMS error comparison among our method, FEQ, and DSD.

facial features such as the eyes, nose, mouth, and ears. Therefore, FEQ is also better than DSD in terms of simplification results. However, the features of facial animation include not only the eyes, nose, mouth, and ears, but also wrinkles on an angry forehead, protuberant cheeks, and obvious nasolabial folds on a smiling face.

The data in Table 3 and Figure 12 indicate that when the number of triangles is simplified to 5,000, 2,000, 1,000, and 500, the errors generated by FEQ are 1.2658×10^{-2} , 2.7134×10^{-2} , 5.2143×10^{-2} , and 10.6767×10^{-2} , which are better than DSD's but are only limited to general facial features such as eyes, nose, mouth, and ears, having no obvious improvement in other expression features like wrinkles in forehead and nasolabial folds in cheeks.

The error comparison results in Table 4 indicate that when the number of triangles is simplified to 5,000, 2,000, 1,000, and 500, the respective errors generated by this method are 1.2092×10^{-2} , 2.4207×10^{-2} , 3.7969×10^{-2} , and 6.2438×10^{-2} , which are all better than EFQ's. The improvement rate reaches 4.47% when the number of triangles is simplified to 5,000, and it even reaches 41.52% when the number is 500.

In the preservation of expression features, as shown in Figure 13, when the model is simplified to 5,000 triangular pieces, as shown in Figure 13(b), the FEQ method retains 129 triangles in the forehead wrinkled area; however, this method has 123 fully covered and 37 half-covered triangles in the same area. If we use 0.5 as a unit for each half-covered triangle, then our method obtains 141.5 triangles in the forehead wrinkled

TABLE 4: Average RMS error comparison between our method and FEQ (unit: 10^{-2}).

Triangles	The FEQ	Our method	Improvement rate
5000	1.2658×10^{-2}	1.2092×10^{-2}	4.47%
2000	2.7134×10^{-2}	2.4207×10^{-2}	10.79%
1000	5.2143×10^{-2}	3.7969×10^{-2}	27.18%
500	10.6767×10^{-2}	6.2438×10^{-2}	41.52%

TABLE 5: Perceived quality comparison between our method and DSD (unit: 10^{-1}).

Triangles	DSD	Our method	Improvement rate
5000	2.4633×10^{-1}	1.7413×10^{-1}	29.31%
2000	3.2981×10^{-1}	2.2074×10^{-1}	33.07%
1000	4.0961×10^{-1}	2.5682×10^{-1}	37.30%
500	4.6055×10^{-1}	3.0275×10^{-1}	34.26%

area. In other words, for this feature area, this method retains 10% more triangular pieces than the FEQ. Moreover, when the model is reduced to 1,000 triangles, as shown in Figure 13(c), the FEQ only has 15.5 triangles in the forehead wrinkled area (including 11 fully covered and 9 half-covered triangles), but this method has 28.5 triangles (20 fully covered and 17 half-covered triangles); 84% more triangles are retained compared with the FEQ.

In addition, in the cheeks and the nasolabial fold areas, as shown in Figure 14, when the number of triangles in the model is reduced to 1,000 using the FEQ, these areas retain 61 triangles. However, by using the proposed method to simplify the model, 113.5 triangles are retained (including 97 fully covered and 33 half-covered triangles), which is approximately 86% more triangles as compared to the FEQ.

From the simplified results of forehead wrinkles, cheeks, and nasolabial fold areas, it can be seen that this method retains a greater number of triangles than the FEQ. It also shows that, after simplification, this method gets closer to expressing the original facial model in these feature areas than the FEQ method.

To verify the effectiveness of our method, we also adopted perceived quality to further compare it with DSD and FEQ. Perceived quality is mainly used to compare visual differences. We used tensor-based perceptual distance measure (TPDM) [34], proposed by Torkhani et al., to evaluate the perceived quality values of the model before and after simplification. The perceived quality value is between zero and one. A value closer to one signifies that the appearance of the simplified model is very similar to the original, whereas a value closer to zero signifies that the appearance of the simplified model is very different from the original.

In Table 5, it can be seen that the perceived value generated by our method is 1.7413×10^{-1} to 3.0275×10^{-1} , results that are clearly superior to DSD's 2.4633×10^{-1} to 4.6055×10^{-1} , with its improvement rate of 29.31% to 37.30%. In Table 6, it can be seen that the perceived value generated by FEQ is 1.7413×10^{-1} to 3.0275×10^{-1} . Thus, this method generates a lower perceived value than FEQ and can

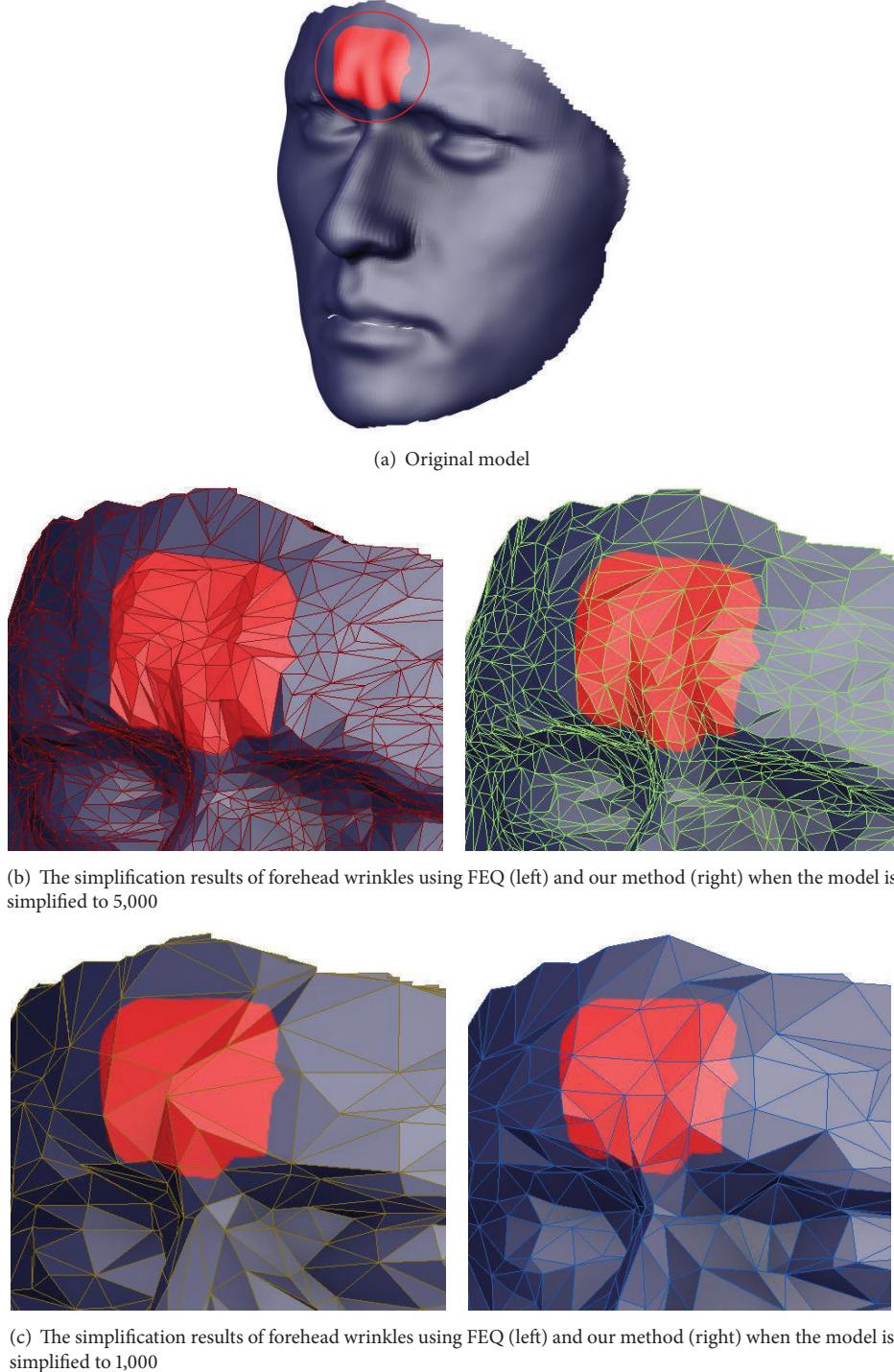


FIGURE 13: Comparison between FEQ and our method when the number of triangles in the face 03 (anger) model is simplified to 5,000 and 1,000.

achieve an improvement rate of 2.33% to 12.88%, as shown in Figure 15.

In addition, in terms of time cost, given that this improved method is based on DSD, it inherits benefits such as lower computation time. In order to perform the time cost analysis,

this paper divided the entire implementation process into four phases, namely, the setup time, initialization time, simplification running time, and output time.

In the setup time phase, the main work is to input the original facial models, including vertex coordinates and triangle

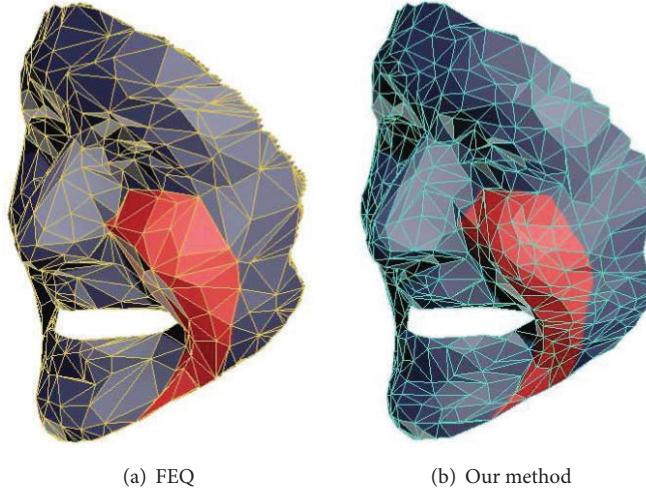


FIGURE 14: Comparison between FEQ and our method when the number of triangles in the face 04 (grin) model is simplified to 1,000.

TABLE 6: Perceived quality comparison between our method and FEQ (unit: 10^{-1}).

Triangles	FEQ	Our method	Improvement rate
5000	1.7828	1.7413	2.33%
2000	2.3440	2.2074	5.83%
1000	2.8950	2.5682	11.29%
500	3.4751	3.0275	12.88%

information. In the initialization time phase, this method calculates relevant information needed to simplify the model, including QEM, HCTM, and MSO. In the simplification running time phase, the method mainly records the execution time needed to simplify the model until it reaches a specific required number of triangles.

In the final output time phase, it records the time needed to output the simplified facial model. The average time required for each execution process is shown in Table 7. As is evident from the table, the execution speed of this method is significantly fast. The overall simplification time is only about 8 seconds, in which the initialization time for HCTM and MSO is just over 2-3 seconds, accounting for about 37% of the total simplification time. Undoubtedly, this method is very efficient in model simplification.

7. Conclusions

In order to analyze the shape changes of facial animation and to reduce simplification errors, this paper proposed HCTM to modify the homogeneity of the local coordinate system for different models and adopted MSO to automatically analyze the degree of facial animation shape change, to locate the region with most expression changes and to rectify

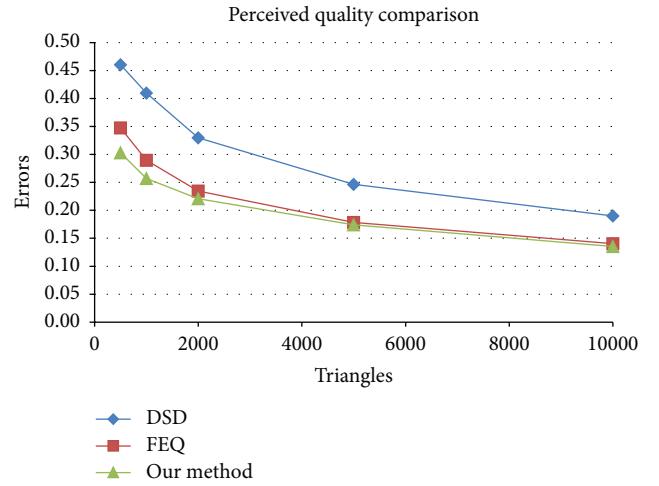


FIGURE 15: Perceived quality comparison among our method, FEQ, and DSD.

the drawback wherein feature regions such as the eyes, nose, mouth, and ears had to be defined by users. In experiments conducted, RMS and perceived quality errors were utilized to compare the simplification results of the proposed method with those of DSD and FEQ. The experimental results show that the errors caused by this method are lower than those of DSD and FEQ. Furthermore, this method can not only properly retain the facial features of fixed positions such as the eyes, nose, mouth, and ears, but also preserve more triangles than other methods in other important feature regions such as wrinkles on the forehead, cheeks, and nasolabial folds. Thus, it satisfies the requirement that the simplified facial animation should be as elaborate and natural as possible.

TABLE 7: Time analysis for each execution process in facial model simplification (unit: second).

Triangles	Setup time	Initialization time	Running time	Output time	Total time
5000	2.95	2.92	1.95	0.22	8.04
2000	2.78	3.05	2.09	0.17	8.09
1000	2.53	2.97	2.13	0.11	7.74
500	2.78	2.88	2.19	0.09	7.94
Average	2.76	2.96	2.09	0.15	7.95

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author thanks Professors Chassery, Wang, and Torkhani for providing the TPDM codes. Additionally, this research was supported by the Minghsin University of Science and Technology, Taiwan, under Grant MUST-104CSIE-2.

References

- [1] S. Karunaratne and H. Yan, “3D animated movie actor training using fuzzy logic,” in *Proceedings of the IEEE Computer Graphics International*, pp. 23–30, IEEE, Hong Kong, July 2011.
- [2] M. Salvati, M. Kinoshita, Y. Katsura, K. Anjyo, T. Yotsukura, and H. Uchibori, “Developing tools for 2D/3D conversion of Japanese animations,” in *Proceedings of the ACM SIGGRAPH Talks (SIGGRAPH ’11)*, Article no. 14, ACM, Vancouver, Canada, August 2011.
- [3] M. Song and P. Grogono, “Application of advanced rendering and animation techniques for 3D games to softbody modeling and animation,” in *Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering (C3S2E ’09)*, pp. 89–100, ACM, Montreal, Canada, May 2009.
- [4] M. Zhao and J. Zhang, “Rapidly product and optimize facial animation methods for 3D game,” in *Proceedings of the International Conference on Internet Computing in Science and Engineering (ICICSE ’08)*, pp. 136–139, IEEE, Harbin, China, January 2008.
- [5] L. Bangquan and M. Yun, “A facial animation based on emotional model for characters in 3D games,” in *Proceedings of the IEEE International Conference on Computer Science and Information Processing (CSIP ’12)*, pp. 1304–1307, Xi’an, China, August 2012.
- [6] I. Conradi and Y. Y. X. Ivan, “The undiscovered country: the art of pictorial 3-D stereo animation,” in *Proceedings of the International Conference on Cyberworlds*, pp. 181–188, IEEE, Ontario, Canada, October 2011.
- [7] V. Janarthanan, “Innovations in art and production: sound, modeling and animation,” in *Proceedings of the IEEE 9th International Conference on Information Technology: New Generations (ITNG ’12)*, pp. 879–882, IEEE, Las Vegas, Nev, USA, April 2012.
- [8] M. Aoki, W. Koning, A. Miyai, and T. Kamihira, “3D animation education in the US and Japan: different environments, similar issues,” in *Proceedings of the SIGGRAPH Asia Sketches (SA ’11)*, ACM, Hong Kong, December 2011.
- [9] H. Croft, R. Rasiah, J. Cooper, and K. Nesbitt, “Comparing animation with video for teaching communication skills,” in *Proceedings of the ACM Conference on Interactive Entertainment (IE ’14)*, pp. 1–10, Newcastle, Australia, December 2014.
- [10] F. I. Parke, “Computer generated animation of faces,” in *Proceedings of the ACM Annual Conference (ACM ’72)*, vol. 1, pp. 451–457, ACM, Boston, Mass, USA, August 1972.
- [11] Y. Cai, M. Yang, and Z. Li, “Robust head pose estimation using a 3D morphable model,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 678973, 10 pages, 2015.
- [12] U. Musti, Z. Zhou, and M. Pietikainen, “Facial 3D shape estimation from images for visual speech animation,” in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR ’14)*, pp. 40–45, Stockholm, Sweden, August 2014.
- [13] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel, “Head shop: generating animated head models with anatomical structure,” in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 55–63, San Antonio, Tex, USA, July 2002.
- [14] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec, “Creating a photoreal digital actor: the Digital Emily project,” in *Proceedings of the 6th European Conference for Visual Media Production (CVMP ’09)*, pp. 176–187, IEEE, London, UK, November 2009.
- [15] A. Maejima, H. Yarimizu, H. Kubo, and S. Morishima, “Automatic generation of head models and facial animations considering personal characteristics,” in *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology (VRST ’10)*, pp. 71–78, ACM, Hong Kong, November 2010.
- [16] T. Weise, S. Bouaziz, H. Li, and M. Pauly, “Kinect-based facial animation,” in *Proceedings of the SIGGRAPH Asia Emerging Technologies (SA ’11)*, Article No. 1, Hong Kong, December 2011.
- [17] T. Weise, S. Bouaziz, H. Li, and M. Pauly, “Realtime performance-based facial animation,” *ACM Transactions on Graphics*, vol. 30, no. 4, article 77, 2011.
- [18] S. Bouaziz, Y. Wang, and M. Pauly, “Online modeling for realtime facial animation,” *ACM Transactions on Graphics*, vol. 32, no. 4, article 40, 2013.
- [19] C. Cao, Y. Weng, S. Lin, and K. Zhou, “3D shape regression for real-time facial animation,” *ACM Transactions on Graphics*, vol. 32, no. 4, article 41, 2013.
- [20] C. Luo, J. Yu, C. Jiang, R. Li, and Z. Wang, “Real-time control of 3D facial animation,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME ’14)*, pp. 1–6, IEEE, Chengdu, China, July 2014.
- [21] C. Cao, Q. Hou, and K. Zhou, “Displaced dynamic expression regression for real-time facial tracking and animation,” *ACM Transactions on Computer Systems*, vol. 33, no. 4, article 43, 2014.
- [22] J. Zhao, C. Liu, Z. Wu et al., “3D facial similarity measure based on geodesic network and curvatures,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 832837, 17 pages, 2014.

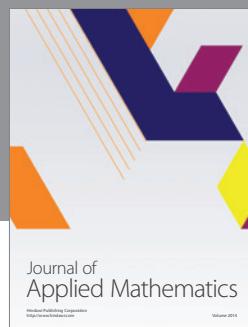
- [23] A. Mohr and M. Gleicher, “Deformation sensitive decimation,” Tech. Rep., University of Wisconsin, 2003.
- [24] S. Zhang and E. Wu, “Deforming Surface Simplification Based on Feature Preservation,” in *Entertainment Computing—ICEC 2007*, vol. 4740 of *Lecture Notes in Computer Science*, pp. 139–149, Springer, Berlin, Germany, 2007.
- [25] S.-K. Kim, S.-O. An, H.-T. Kim, and M. Hong, “Efficient modeling for animating human face,” in *Proceedings of the IEEE International Symposium on Ubiquitous Multimedia Computing (UMC ’08)*, pp. 35–38, IEEE, October 2008.
- [26] A.-B. Wang, B. Yu, and Z.-J. Liu, “Mesh simplification based on facial features region partition,” in *Proceedings of the International Conference on Image Analysis and Signal Processing (IASP ’09)*, pp. 269–272, IEEE, Taizhou, China, April 2009.
- [27] R. L. P. Duarte, A. El Rhalibi, C. Carter, S. Cooper, and M. Merabti, “An MPEG-4 quadric-based LoD simplification for facial animation,” in *Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS ’12)*, pp. 743–748, Tangier, Morocco, May 2012.
- [28] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics,” in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 209–216, Los Angeles, Calif, USA, August 1997.
- [29] B.-S. Jong, J.-L. Tseng, and W.-H. Yang, “An efficient and low-error mesh simplification method based on torsion detection,” *The Visual Computer*, vol. 22, no. 1, pp. 56–67, 2006.
- [30] J. L. Tseng, “Surface Simplification of 3D animation models using robust homogeneous coordinate transformation,” *Journal of Applied Mathematics*, vol. 2014, Article ID 189241, 14 pages, 2014.
- [31] M. D. Adams, *Multiresolution Signal and Geometry Processing: Filter Banks, Wavelets, and Subdivision*, University of Victoria, 2013.
- [32] A. Gray, E. Abbena, and S. Salamon, *Modern Differential Geometry of Curves and Surfaces with Mathematica*, Chapman & Hall, CRC Press, 2006.
- [33] J.-L. Tseng and Y.-H. Lin, “3D surface simplification based on extended shape operator,” *WSEAS Transactions on Computers*, vol. 12, no. 8, pp. 320–330, 2013.
- [34] F. Torkhani, K. Wang, and J.-M. Chassery, “A curvature-tensor-based perceptual quality metric for 3D triangular meshes,” *Machine Graphics and Vision*, vol. 23, no. 1-2, pp. 59–82, 2014.



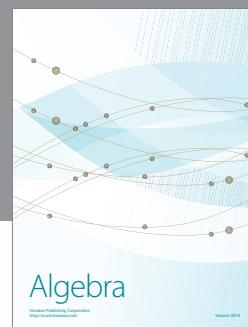
Advances in
Operations Research



Advances in
Decision Sciences



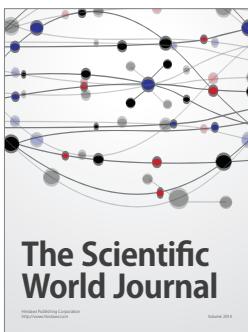
Journal of
Applied Mathematics



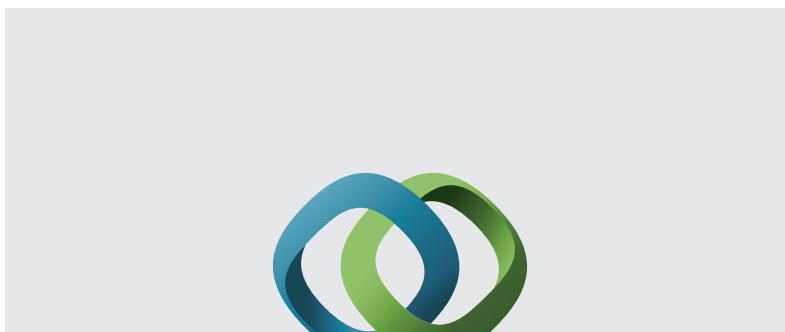
Algebra



Journal of
Probability and Statistics



The Scientific
World Journal

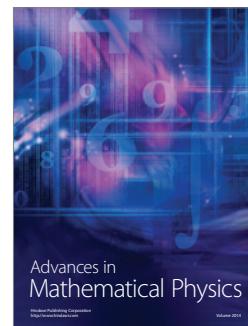


Hindawi

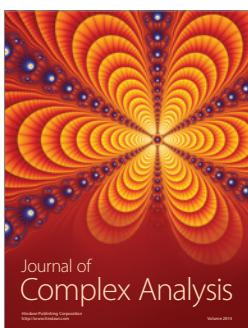
Submit your manuscripts at
<http://www.hindawi.com>



International Journal of
Combinatorics



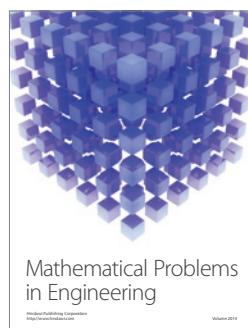
Advances in
Mathematical Physics



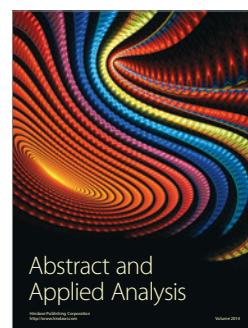
Journal of
Complex Analysis



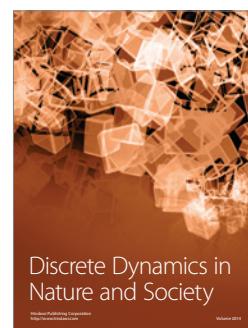
Journal of
Mathematics



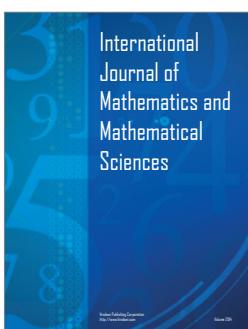
Mathematical Problems
in Engineering



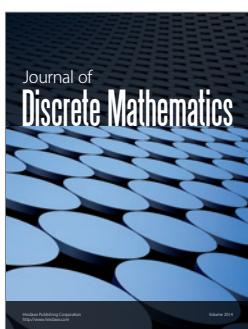
Abstract and
Applied Analysis



Discrete Dynamics in
Nature and Society



International
Journal of
Mathematics and
Mathematical
Sciences



Journal of
Discrete Mathematics



Journal of
Function Spaces



International Journal of
Stochastic Analysis



Journal of
Optimization