Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 189241, 14 pages
http://dx.doi.org/10.1155/2014/189241

Research Article

Surface Simplification of 3D Animation Models Using Robust Homogeneous Coordinate Transformation

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 13 June 2014; Accepted 10 August 2014; Published 1 September 2014

Academic Editor: Zhenfu Cao

The goal of 3D surface simplification is to reduce the storage cost of 3D models. A 3D animation model typically consists of several 3D models. Therefore, to ensure that animation models are realistic, numerous triangles are often required. However, animation models that have a high storage cost have a substantial computational cost. Hence, surface simplification methods are adopted to reduce the number of triangles and computational cost of 3D models. Quadric error metrics (QEM) has recently been identified as one of the most effective methods for simplifying static models. To simplify animation models by using QEM, Mohr and Gleicher summed the QEM of all frames. However, homogeneous coordinate problems cannot be considered completely by using QEM. To resolve this problem, this paper proposes a robust homogeneous coordinate transformation that improves the animation simplification method proposed by Mohr and Gleicher. In this study, the root mean square errors of the proposed method were compared with those of the method proposed by Mohr and Gleicher, and the experimental results indicated that the proposed approach can preserve more contour features than Mohr's method can at the same simplification ratio.

1. Introduction

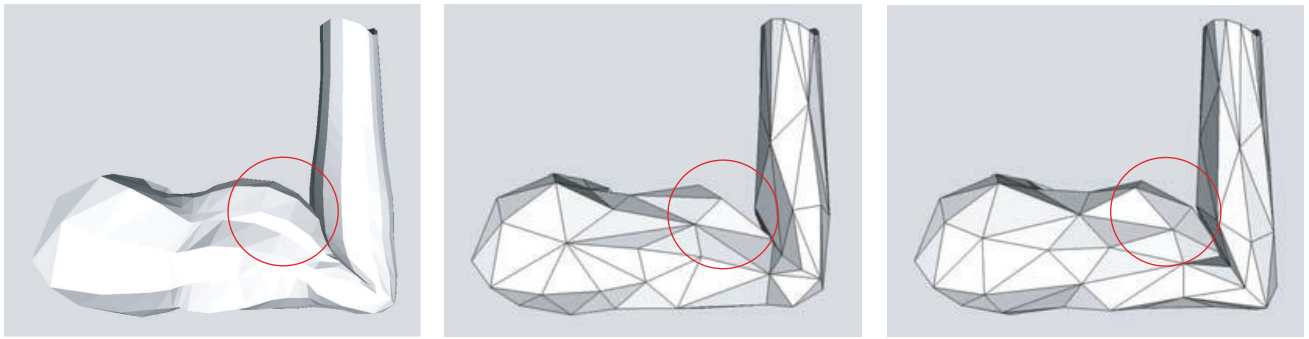
The development of information technology has caused 3D techniques to be applied often in numerous aspects of digital life [1–5], especially computer graphics [6–9]. A major component of 3D animation is 3D objects, and realistic animation requires many frames, each containing numerous vertices and triangles. Therefore, several surface simplification methods have been proposed to reduce the storage costs [10–16].

These simplification methods include vertex decimation [17], vertex clustering [18], edge contraction [19], triangle contraction [20], and quadric error metrics (QEM) simplification [21]. QEM is one of the accurate methods for simplifying static 3D models, providing low-error simplifications of high-resolution static models. Although QEM is suited to simplifying static models, it cannot be applied alone to simplify animation models directly. Because each frame model in an animation is static, QEM must simplify one of all frame models in an animation model first and then simplify other frame models by following the simplification sequence of the first simplified one if the animation model has to be directly simplified by QEM. The errors caused by simplification increase rapidly and the shape features are destroyed easily, in spite of the fact that an animation model can be simplified by the previous way, as shown in Figure 1.

To reduce these errors, Mohr and Gleicher [22] proposed the deformation sensitive decimation (DSD) method. This method is an extension of the QEM algorithm that involves constructing a metamesh after summing the quadric errors incurred by all frames. Because vertices with the same index do not have the same local space coordinate in different frame models, the quadric errors cannot only be summed. To solve this problem, Mohr and Gleicher replaced the final row of the pair's summed quadric matrix Q with $[0\ 0\ 0\ 1]$ and used Q^{-1} to determine the optimal vertex position. However, Q^{-1} does not always exist. To resolve this problem, this paper proposes a robust homogeneous coordinate transformation (RHCT) that improves the DSD method.

2. Related Works

2.1. Quadric Error Metrics. Because the QEM method proposed by Garland and Heckbert [21] has recently been



(a) (b) (c)

Figure 1: Simplification results between the single-pose QEM and the DSD method. (a) Original model; (b) simplifying by single-pose QEM; (c) the DSD method [22].

identified as one of the most effective simplification methods, numerous experts have studied it, extending its applications in level of detail (LOD) [23], radiosity rendering [23], and animation model simplification [22]. QEM involves contracting vertex v_1 and vertex v_2 into a new vertex, v, the position of which is obtained according to the minimal distance between a vertex and the conjoining triangles, as shown in Figure 2. This minimal distance represents the error of simplification. This method enables the outer shape of a 3D model to be retained after the model is simplified and reduces the errors caused by surface simplification. The new estimated position of v strongly influences changes in the surrounding triangles.

QEM involves measuring the error by using a quadric metric as follows:

$$\text{Error}(v) = v^T \cdot \left(\sum K_p\right) \cdot v, \tag{1}$$

where v is an estimated point and K_p represents a quadric metric defined by plane p: $ax + by + cz + d = 0$. When $a^2 + b^2 + c^2 = 1$, the distance from any vertex v to the plane p is represented as $(p^T v)^2 = v^T (pp^T) v = v^T K_p v$. In other words, K_p can be represented as follows, and the initial matrix Q can be obtained by conglomerating the planes for the triangles meeting at v:

$$Q = \sum_{p \in \text{planes}(v)} K_p,$$

$$K_p = p \cdot p^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}, \tag{2}$$

where planes(v) indicates the triangles meeting at the vertex v. When the vertex pair (v_1, v_2) is simplified into a new vertex v, the matrix Q of vertex v is converted into $(Q_1 + Q_2)$, where Q_1 and Q_2 are the quadric matrices of vertices v_1 and v_2, respectively. In other words, the cost of contraction is simply computed as $v^T(Q_1 + Q_2)v$. This method rapidly provides a low-error solution for surface simplification.

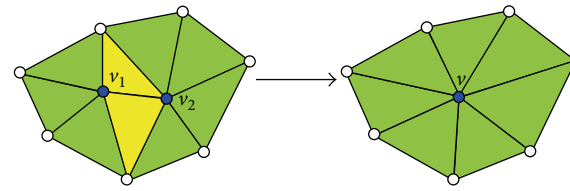


Figure 2: Surface simplification using quadric error metrics [21].

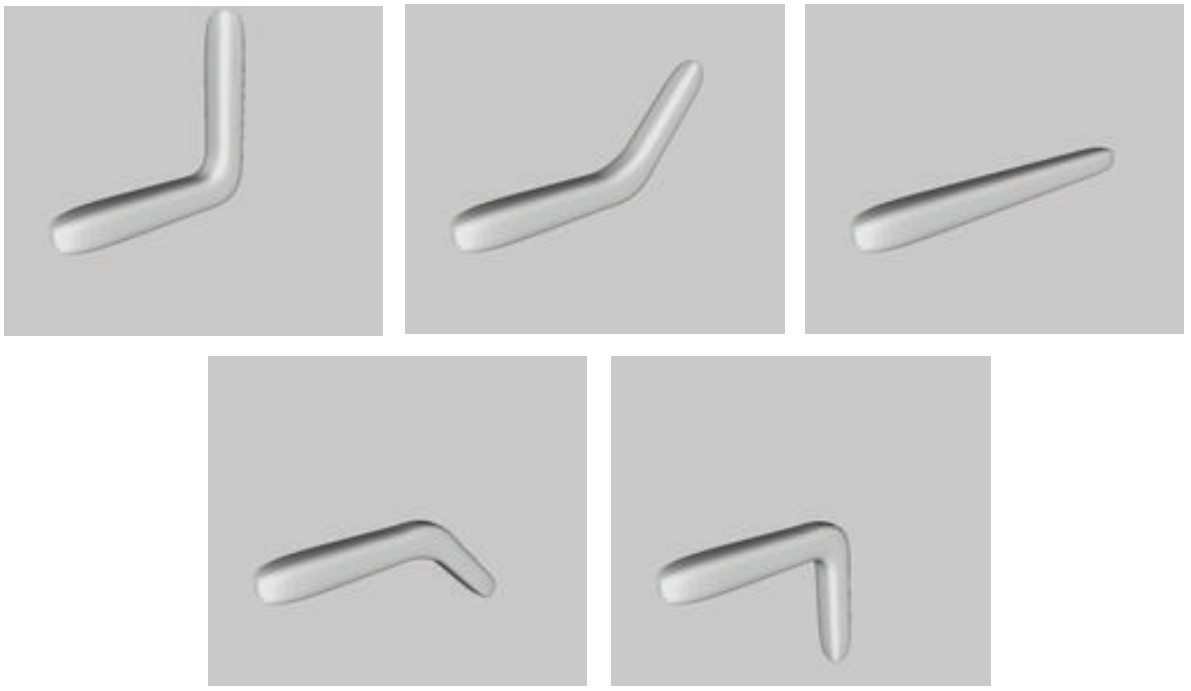
2.2. Deformation Sensitive Decimation. Although QEM features advantages such as a high computation speed and a low average error, it can be used to simplify only static models. That is, QEM can obtain a good simplified result for one of all frame examples in an animation but cannot work well for each frame example at the same time. For example, in the animation model containing five frame examples shown in Figure 3(a), the first frame example is simplified first, and the fifth frame example is simplified using the collapse sequence based on the first frame example, as shown in Figure 3(b). The first frame example can be simplified adequately by using this approach, but the fifth frame example cannot.

To apply QEM in simplifying all frame examples in an animation model, Mohr and Gleicher [22] proposed DSD method. This method is used to generate the contraction cost by considering all frame examples in an animation model. The method involves tracking the error quadric and summing the matrices per vertex for each frame example. The error measurement is modified as follows:

$$\sum_{i=1}^{k} \tilde{v}_i^T \left(Q_{i,v_1} + Q_{i,v_2}\right) \tilde{v}_i, \tag{3}$$

where k is the number of frame examples, $\tilde{v}_i$ represents the new vertex formed by collapsing vertices v_1 and v_2 in the ith frame example, and Q_{i,v_1} and Q_{i,v_2} represent the error quadrics of vertices v_1 and v_2, respectively.

Experimental results indicated that when DSD is used to simplify the animation model in Figure 3(c), the simplification of the fifth frame example is superior to that generated using the QEM of first frame example.



(a) An animation model with five frame examples

(b) Using the QEM of example 1 only (c) Using the DSD

Figure 3: Simplification comparison between using the QEM of the first frame example and using the DSD [21].

2.3. Progressive Multiresolution Meshes. To apply QEM in high degree of nonrigid deformation, Kircher and Garland proposed a new multiresolution method to deform surfaces [24]. This method comprises two major steps. First, a multilevel mesh is created for the first frame model in the deformation sequence. Each level is numbered in an increasing order from fine to coarse, starting with M_0. This method uses QEM to cluster all vertices in M_0, generate a simplified model M_1, and complete the initial hierarchy H_0 gradually.

In the second step, a reclustering algorithm is employed to improve the time-varying hierarchies for each subsequent frame model. To update the time-varying hierarchies, a swap approach is used to recluster vertices; in other words, the $(i+1)$th hierarchy H_{i+1} is generated from the ith hierarchy H_i, and all swap operations varying from H_i to H_{i+1} are recorded, as shown in Figure 4.

According to the experimental results [24], the simplification errors generated by progressive multiresolution meshes were between the DSD and Direct QSlim. Because this method updated hierarchies from the first frame hierarchy, the models similar to the first frame model (frames 1, 13, 25, 37, and 49) were simplified well. By contrast, the models that greatly differed from the first frame model, such as frames 6, 18, 30, and 42, could not be simplified effectively using progressive multiresolution meshes. Local maximal simplification errors were observed in frames 6, 18, 30, and 42, and these errors were nearly equal to those of the DSD.

Many studies have recently referenced DSD in proposing numerous related works, including progressive multiresolution meshes [24], articulated meshes [25, 26], animation key-frame extraction [27], and analytic error metrics [28]. However, one shortfall of the DSD method, the homogeneous coordinate problem, must be resolved.

To manage homogeneous coordinates, DSD entails replacing the final row of the summed quadric matrix Q with $[0\ \ 0\ \ 0\ \ 1]$ and determining the Q^{-1} to obtain the optimal

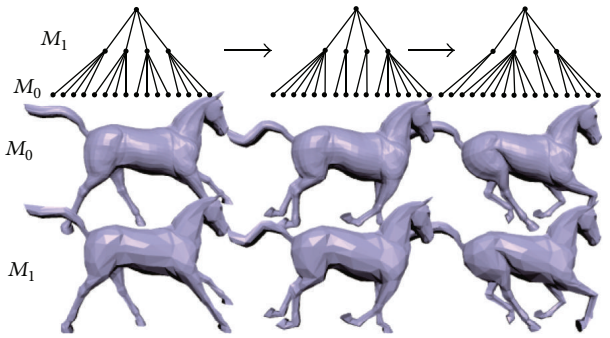


Figure 4: Multiresolution representation proposed by Kircher and Garland [24].

vertex position. However, Q^{-1} does not always exist. Therefore, we propose a RHCT to improve the DSD method.

3. Robust Homogeneous Coordinate Transformation

An animation model generally comprises several frame examples, and each frame example has a distinct pose. The surface information, such as information on the position and normal vector, of a vertex differs among frame examples. In brief, the local coordinate system of a vertex varies among the frame examples of an animation model. Before an animation model is simplified, a vertex present in different frame examples should be transformed to one local coordinate space. This transformation is conducted by applying the following theorems.

Theorem 1 (Euler's rotation [29]). *In 3D space, the transform for rotating a point around the origin of a plane is expressed as follows:*

$$R_\beta = \begin{bmatrix} \cos\beta & -\sin\beta & 0 & 0 \\ \sin\beta & \cos\beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

Any rotation may be described using three angles. If the rotations are written using rotation matrices D, C, and B, then a general rotation A can be written as $A = B \times C \times D$.

Theorem 2 (equivalence of an orthogonal matrix to a rotation matrix [31]). *A rotation matrix is a matrix that, when multiplied by a vector, rotates that vector in its n-dimensional domain. A rotation matrix is equivalent to an orthogonal matrix, and an orthogonal matrix is defined as a matrix of which the transpose is equal to its determinant, which is equal to 1:*

$$Q^{-1} = Q^T, \qquad |Q| = 1. \tag{5}$$

According to Theorem 1, the matrices for rotation around the x-axis by α, around the y-axis by β, and around the z-axis by γ are expressed as

$$R_{\alpha,x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_{\beta,y} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

$$R_{\gamma,z} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Theorem 3 (rotating a point around an arbitrary axis [30]). *Assume the axis of rotation is given by the unit vector: $\hat{n} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$. $P(x_p, y_p, z_p)$ is the point that is rotated by angle θ to $P'(x'_p, y'_p, z'_p)$, as shown in Figure 5. The rotation transform is*

$$\begin{bmatrix} x'_p \\ y'_p \\ z'_p \\ 1 \end{bmatrix} = \begin{bmatrix} a^2K + \cos\theta & abK - c\sin\theta & acK + b\sin\theta & 0 \\ abK + c\sin\theta & b^2K + \cos\theta & bcK - a\sin\theta & 0 \\ acK - b\sin\theta & bcK + a\sin\theta & c^2K + \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}, \tag{7}$$

where $K = 1 - \cos\theta$.

To transform the vertex in different frame examples to one local coordinate space, the local coordinate spaces in

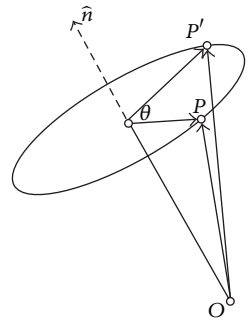


Figure 5: Rotating a point about an arbitrary axis [30].

the first frame example are used as major coordinate spaces. Each local coordinate laid on each frame example must be transformed to that on the first frame example. Suppose that the normal vector and tangent plane of a vertex $v_{i,j}$ in frame j are notated in $n_{i,j} = \begin{bmatrix} n_{x,i,j} & n_{y,i,j} & n_{z,i,j} & 1 \end{bmatrix}^T$ and $t_{i,j}$, respectively, and $M_{i,j}$ represents the mesh of connecting vertex $v_{i,j}$. To transform the mesh $M_{i,j}$ to the coordinate space of $M_{i,1}$, the normal vector $n_{i,j}$ and tangent plane $t_{i,j}$ at vertex $v_{i,j}$ must be transformed to $n_{i,1}$ and $t_{i,1}$ at vertex $v_{i,1}$. According to Theorem 3, the vertex $M_{i,j}$ can be transformed to the local coordinate space of $M_{i,1}$ by rotating the normal vector $n_{i,j}$:

$$T_{i,j} = \begin{bmatrix} \hat{n}_{x,i,j}^2 K + \cos\theta_{i,j} & \hat{n}_{x,i,j}\hat{n}_{y,i,j}K - \hat{n}_{z,i,j}\sin\theta_{i,j} & \hat{n}_{x,i,j}\hat{n}_{z,i,j}K + \hat{n}_{y,i,j}\sin\theta_{i,j} & 0 \\ \hat{n}_{x,i,j}\hat{n}_{y,i,j}K + \hat{n}_{z,i,j}\sin\theta_{i,j} & \hat{n}_{y,i,j}^2 K + \cos\theta_{i,j} & \hat{n}_{y,i,j}\hat{n}_{z,i,j}K - \hat{n}_{x,i,j}\sin\theta_{i,j} & 0 \\ \hat{n}_{x,i,j}\hat{n}_{z,i,j}K - \hat{n}_{y,i,j}\sin\theta_{i,j} & \hat{n}_{y,i,j}\hat{n}_{z,i,j}K + \hat{n}_{x,i,j}\sin\theta_{i,j} & \hat{n}_{z,i,j}^2 K + \cos\theta_{i,j} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

$$T_{i,j}^{-1} = T_{i,j}{}^T = \begin{bmatrix} \hat{n}_{x,i,j}^2 K + \cos\theta_{i,j} & \hat{n}_{x,i,j}\hat{n}_{y,i,j}K + \hat{n}_{z,i,j}\sin\theta_{i,j} & \hat{n}_{x,i,j}\hat{n}_{z,i,j}K - \hat{n}_{y,i,j}\sin\theta_{i,j} & 0 \\ \hat{n}_{x,i,j}\hat{n}_{y,i,j}K - \hat{n}_{z,i,j}\sin\theta_{i,j} & \hat{n}_{y,i,j}^2 K + \cos\theta_{i,j} & \hat{n}_{y,i,j}\hat{n}_{z,i,j}K + \hat{n}_{x,i,j}\sin\theta_{i,j} & 0 \\ \hat{n}_{x,i,j}\hat{n}_{z,i,j}K + \hat{n}_{y,i,j}\sin\theta_{i,j} & \hat{n}_{y,i,j}\hat{n}_{z,i,j}K - \hat{n}_{x,i,j}\sin\theta_{i,j} & \hat{n}_{z,i,j}^2 K + \cos\theta_{i,j} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

Assume that the included angle between $n_{i,j}$ and $n_{i,1}$ is $\theta_{i,j} = \cos^{-1}(n_{i,j} \cdot n_{i,1})$. The cross product, $\hat{n}_{i,j} = n_{i,j} \times n_{i,1} = \begin{bmatrix} \hat{n}_{x,i,j} & \hat{n}_{y,i,j} & \hat{n}_{z,i,j} & 1 \end{bmatrix}^T$, is a vector orthogonal to the plane of $n_{i,j}$ and $n_{i,1}$. The transform matrix that rotates the normal vector $n_{i,j}$ by angle $\theta_{i,j}$ around the axis $\hat{n}_{i,j}$ is expressed as (8). According to Theorem 2, the matrix $T_{i,j}$ is an orthogonal matrix. In other words, the inverse matrix of $T_{i,j}$ exists and is defined as (9).

All faces incident to vertex $v_{i,j}$ in different frame examples can be transformed to the local coordinate space at vertex $v_{i,1}$ by using the matrix $T_{i,j}$. This approach ensures that the inverse transform matrix $T_{i,j}^{-1}$ exists. Therefore, this method can be used to identify the optimal contraction vertex when DSD is used in summing the QEM to simplify an animation model. The collapse cost for a pair becomes

$$\sum_{i=1}^{k} \tilde{v}_i^T \left(T_{v_1,i} \times Q_{i,v_1} + T_{v_2,i} \times Q_{i,v_2} \right) \tilde{v}_i, \tag{10}$$

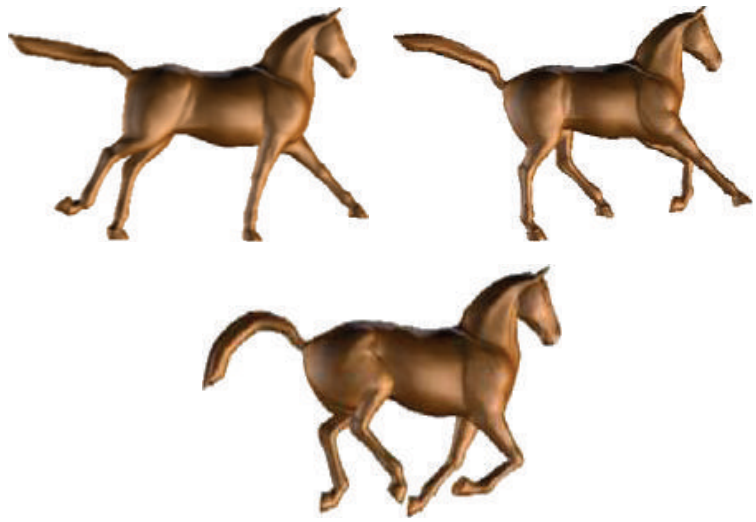
where k is the number of frame examples, $\tilde{v}_i$ is the new vertex formed by collapsing vertices v_1 and v_2 in the ith frame example, Q_{i,v_1} and Q_{i,v_2} represent the error quadrics of vertices v_1 and v_2, respectively, $T_{v_1,i}$ is the matrix that transforms all faces incident to v_1 from the ith frame example to the first frame example, and $T_{v_2,i}$ is the matrix that transforms all faces incident to v_2 from the ith frame example to the first frame example.

4. The Proposed Algorithm

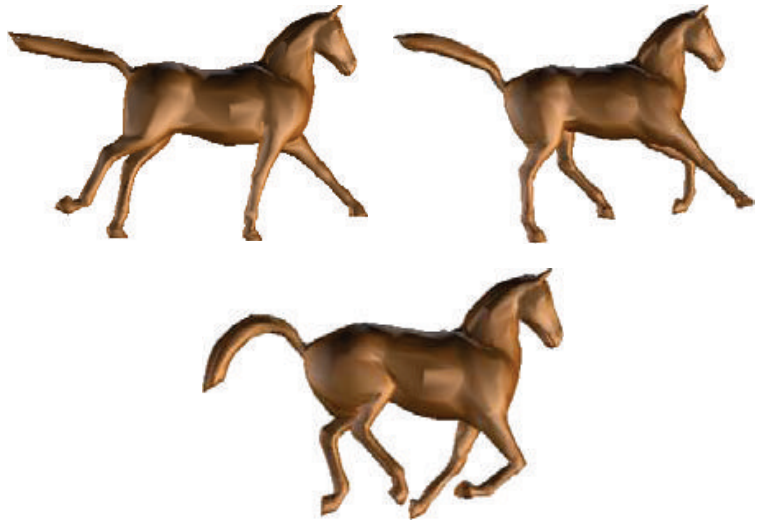
The proposed algorithm refers to and amends DSD methods and includes eight major steps described as follows.

Step 1 (calculate the QEM of each vertex $v_{i,j}$). Its QEM is $Q(v_{i,j}) = Q_{j,v_i} = \Sigma K_p$, where p denotes the triangle plane that contains the point $v_{i,j}$, and K_p represents 4×4 metric of the plane p, as shown in (2).

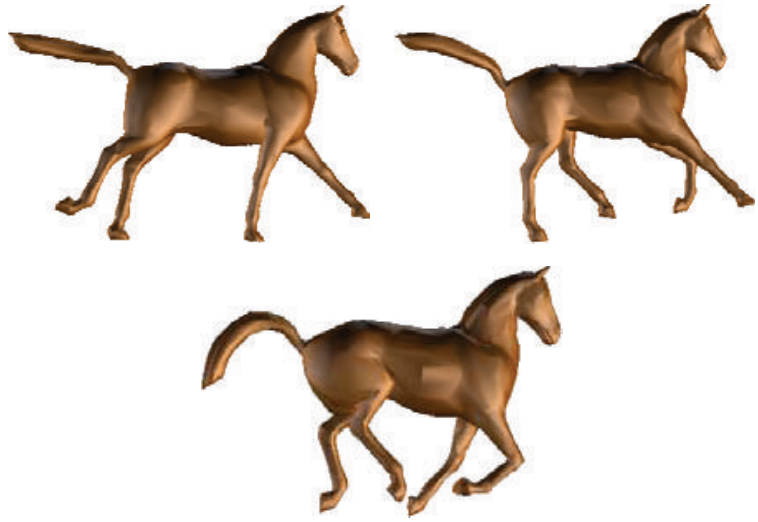
Step 2 (calculate the transformation matrix and modify the QEM). The transformation matrix $T_{i,j}$ shown in (8) must be



(a) Original horse animation model (with 16842 triangles)



(b) Simplified horse animation model using the DSD (with 5000 triangles)



(c) Simplified horse animation model using the RHCT (with 5000 triangles)

Figure 6: Simplification results of the horse animation model (frame examples 1, 21, and 41).

calculated, and then the new QEM is represented as follows. Additionally, $Q'(v_{i,j})$ will be $Q(v_{i,1})$ when j equals 1:

$$Q'\left(v_{i,j}\right) = \sum_{p\in\text{planes}(v_{i,j})} T_{i,j} \times K_p = T_{i,j} \times \left(\sum_{p\in\text{planes}(v_{i,j})} K_p \right) = T_{i,j} \times Q\left(v_{i,j}\right). \tag{11}$$

Step 3. Sum all QEMs of vertex v_i in different frame example:

$$Q'\left(v_i\right) = \sum_{j=1}^{k} Q'\left(v_{i,j}\right). \tag{12}$$

Step 4 (choose each (v_m, v_n) pair to calculate the minimum error produced due to simplification). The (v_m, v_n) pair must satisfy the following conditions:

(a) (v_m, v_n) is an edge or

(b) (v_m, v_n) is not an edge and $\|v_m - v_n\| < t$, where t denotes a threshold value specified by the user.

Step 5. Choose the lowest error vertex-pair (v_m, v_n) as an object of simplification.

Step 6. Contract the vertex-pair (v_m, v_n) into v and calculate its QEM $Q(v)$, where $Q(v) = Q(v_m) + Q(v_n)$.

Step 7. Update all the information of the vertices adjacent to v_m and v_n.

Step 8. Repeat the previous steps until the assigned number of triangles is reached after simplification.

Through Steps 1–3 calculation, the summed QEM with RHCT is obtained, and it is used to contract vertex pairs. The experimental results presented here demonstrate that the RHCT helps to truly reduce the error due to simplification.

5. Experimental Results

This study used the Intel Core i7-2670QM 2.2 GHZ CPU, which features 4 GB of memory, as the main experimental environment, and Visual C++ was used as a programming development tool. Animation models of a horse, a camel, and an elephant were the subjects of the experiment. Each animated sequence comprised 48 frame examples. In the experiment, the root mean square (RMS) errors generated when using DSD to simplify the three animation models were compared with those generated by the proposed method. The experimental results obtained when the horse model was tested are shown in Figure 6, where Figure 6(a) is the original model which consisted of 8431 vertices and 16842 triangles. Figures 6(b) and 6(c) show the results obtained when the horse model was simplified into 5000 triangles by using DSD and the proposed method, respectively.

To estimate the distortion errors of simplified animation models, this study used Metro [32] to calculate the RMS error for each simplified frame example in an animation model and then to determine the mean of these RMS errors. Table 1 and Figure 7 show the mean distortion errors that occurred when the horse animation was simplified into 10000 triangles, 5000 triangles, 2000 triangles, and 1000 triangles by using the DSD and RHCT methods. The RMS errors of the two methods differed substantially when the models were simplified. The proposed method produced 0.828×10^{-3} RMS errors when it was used to simplify the horse model into 5000 triangles and, compared with the model simplified using DSD and the original model comprising 16842 triangles, reduced RMS errors by 53.9% to within an error range of 1.796×10^{-3}. When the model was simplified into 2000 and 1000 triangles, the proposed method achieved improvements of 63.7% and 72%, respectively, compared with DSD.

Table 1: RMS error comparison for simplifying horse animation model (unit: 10^{-3}).

Triangles	The DSD	The RHCT	Improvement rate
10000	0.316	0.316	0.0%
5000	1.796	0.828	53.9%
2000	5.650	2.051	63.7%
1000	12.186	3.322	72.7%

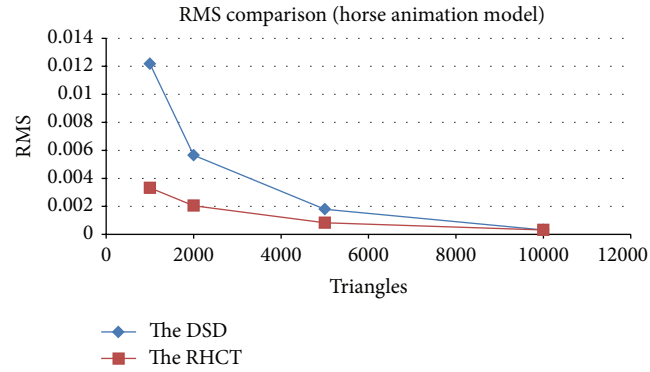
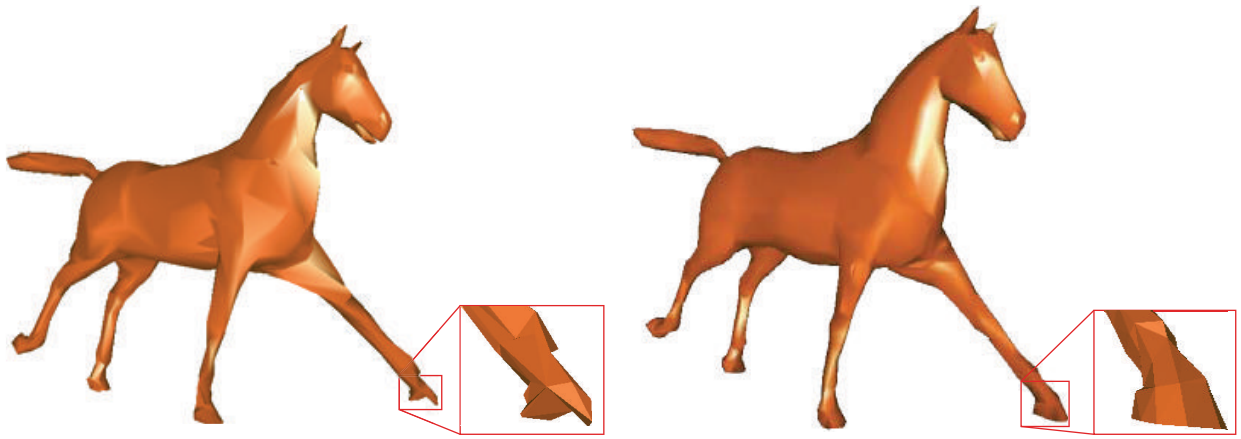


Figure 7: RMS error comparison for simplifying horse animation model.

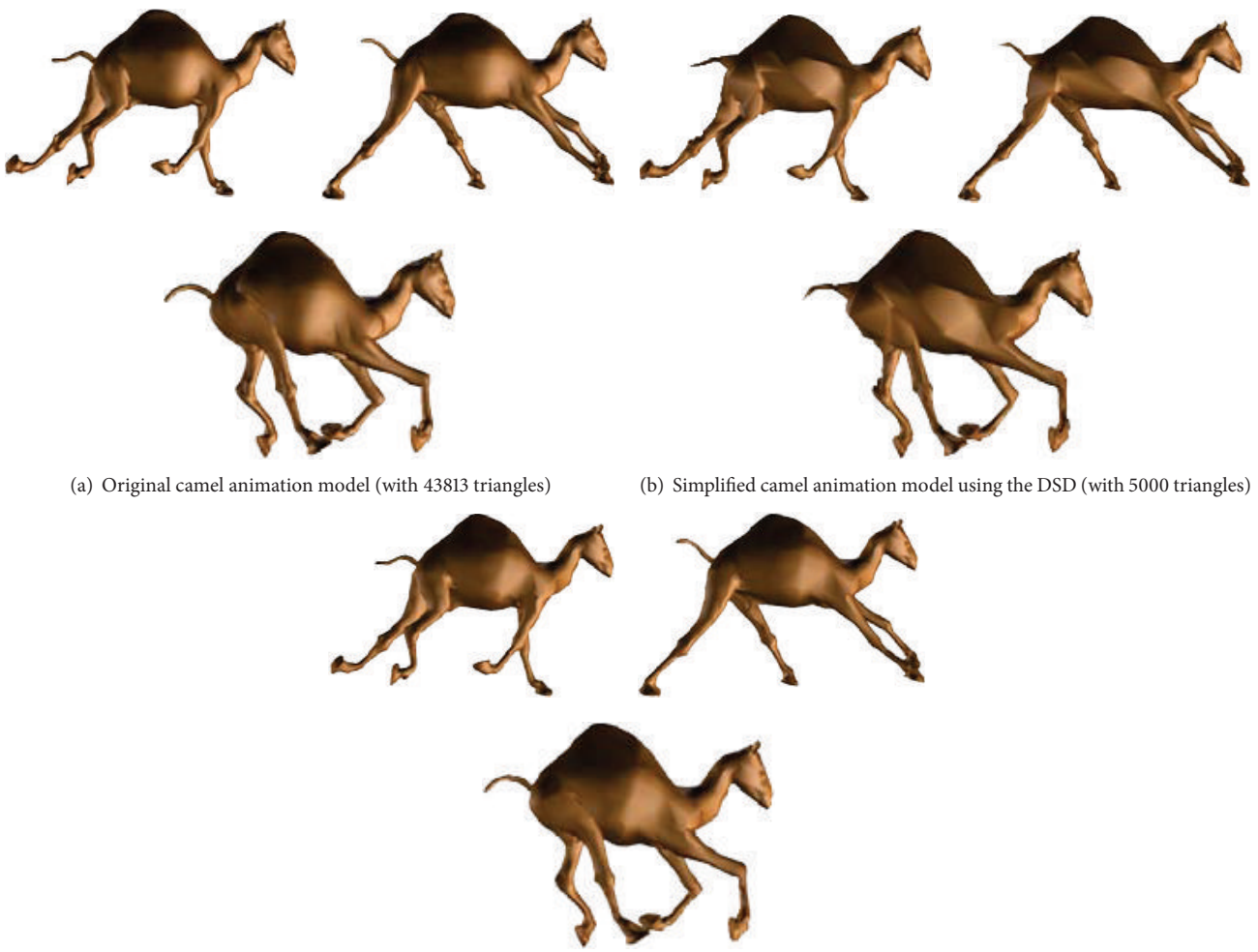
According to the experimental results, when the horse animation model was simplified into 5000 triangles by using DSD, all simplified frame examples remained similar to the original frame examples. However, the shape features deteriorated gradually when the horse examples were simplified into 2000 triangles by using DSD. RHCT was used to determine a suitable vertex position for each simplified vertex pair and retain more shape features than DSD,does as shown in Figure 8.

Each original frame example in the camel model contained 43813 triangles and 21887 vertices. Figure 9 shows the original examples and the simplification results generated using DSD and RHCT. Table 2 and Figure 10 show comparisons of the errors produced by DSD and the proposed



(a) Simplifying by the DSD (2000 triangles) (b) Simplifying by the RHCT (2000 triangles)

Figure 8: The shape of horse hoof is destroyed gradually when simplifying by the DSD.



(a) Original camel animation model (with 43813 triangles) (b) Simplified camel animation model using the DSD (with 5000 triangles)

(c) Simplified camel animation model using the RHCT (with 5000 triangles)

Figure 9: Simplification results of the camel animation model (frame examples 3, 23, and 43).

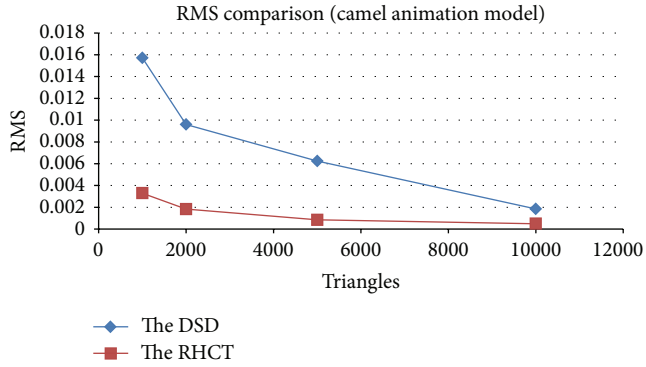
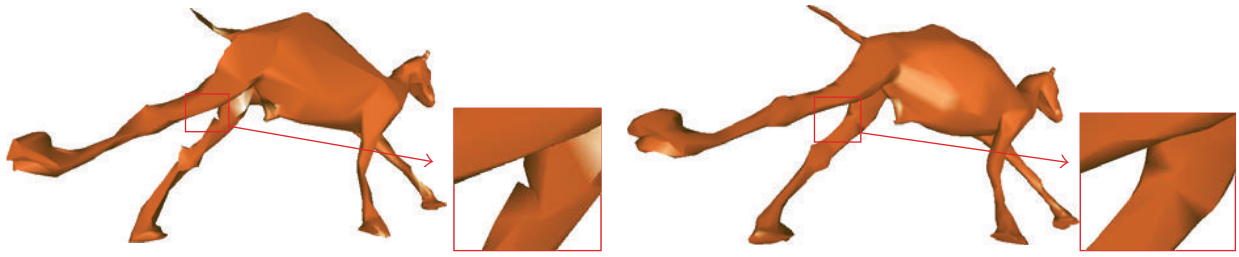


Figure 10: RMS error comparison for simplifying camel animation model.



(a) Simplifying by the DSD (2000 triangles)

(b) Simplifying by the RHCT (2000 triangles)

Figure 11: The thigh shape of camel is destroyed after simplifying by the DSD.

Table 2: RMS error comparison for simplifying camel animation model (unit: 10^{-3}).

Triangles	The DSD	The RHCT	Improvement rate
10000	1.852	0.494	73.3%
5000	6.243	0.845	86.5%
2000	9.599	1.834	80.9%
1000	15.715	3.294	79.0%

Table 3: RMS error comparison for simplifying elephant animation model (unit: 10^{-3}).

Triangles	The DSD	The RHCT	Improvement rate
10000	2.468	0.751	69.6%
5000	4.779	1.390	70.9%
2000	10.909	2.920	73.2%
1000	17.370	4.957	71.5%

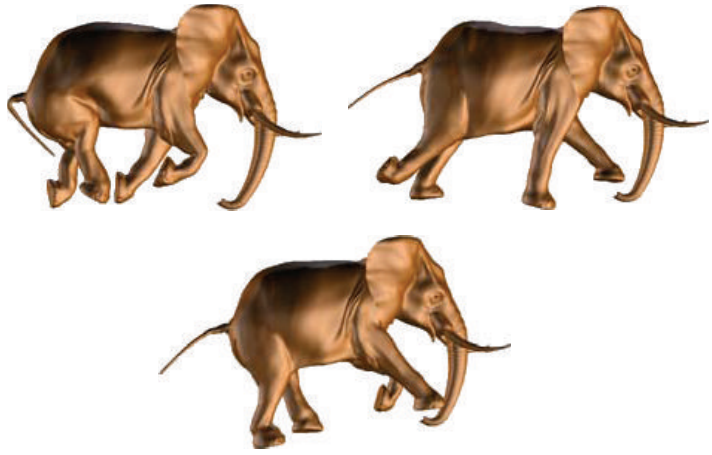
method; the table shows that the proposed method achieved 73–86% error reductions.

In addition, when the camel animation model was simplified into 2000 triangles by using DSD, the outline of the camel leg deteriorated severely. By contrast, the simplified examples produced using the RHCT method retained their major shapes, as shown in Figure 11.

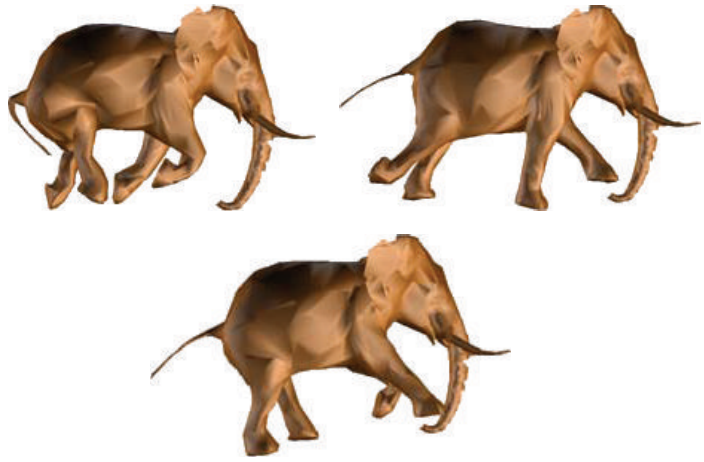
In the experiment, besides taking the horse and camel models for verification, the elephant animation model is taken for experimental testing. The original elephant model contained 84637 triangles and 42321 vertices. Figure 12 shows the simplification results. Table 3 and Figure 13 show a comparison of the error produced using DSD and the proposed method; when the elephant examples were simplified to 10000 triangles by using DSD, the mean RMS error was 0.002468. When the proposed method was used, the error was 0.000751, which is 69.6% lower than that of DSD. Moreover, when the model was simplified to 2000 triangles, the proposed method reduced the error by 73.2% compared with DSD.

Regarding the shape preservation of the simplified elephant example, the shape of the trunk deteriorated severely and even became hollow when the model was simplified to 5000 triangles by using the DSD method, as shown in Figure 14(a). By contrast, RHCT identified a suitable position for the new vertex after simplifying a vertex pair. Therefore, the outline of trunk was kept as intact as possible when it was simplified using the proposed method, as shown in Figure 14(b).

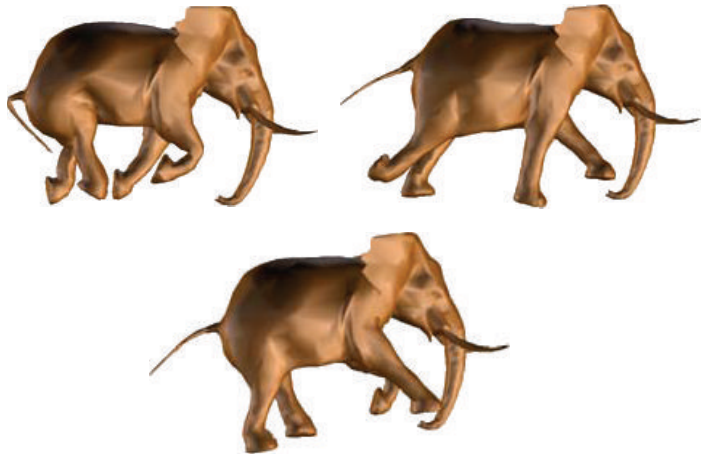
In addition, the progressive multiresolution meshes [24] mentioned in Section 2 also provided wonderful simplification results. This method reduced the local minimal simplification errors of the models similar to the first frame model. However, the local maximal simplification errors, which were



(a) Original elephant animation model (with 84637 triangles)



(b) Simplified elephant animation model using the DSD (with 5000 triangles)



(c) Simplified elephant animation model using the RHCT (with 5000 triangles)

Figure 12: Simplification results of the elephant animation model (frame examples 5, 25, and 45).

nearly equal to those generated by the DSD, could not be reduced. By contrast, RHCT reduced the considerable local maximal simplification errors, as shown in Figure 15.

Moreover, to compare the appearance changes in the simplified models, the experiment employed the tensor-based perceptual distance measure (TPDM) [33] and mesh structural distortion measure 2 (MSDM2) [34] to estimate perceived quality. Both measures were executed in MEPP software [35]. The obtained perceived quality values were between 0 and 1, tending toward 1 when the visualization of the measured objects differed and equaled 0 when they were identical. This experiment simplified the horse sequence

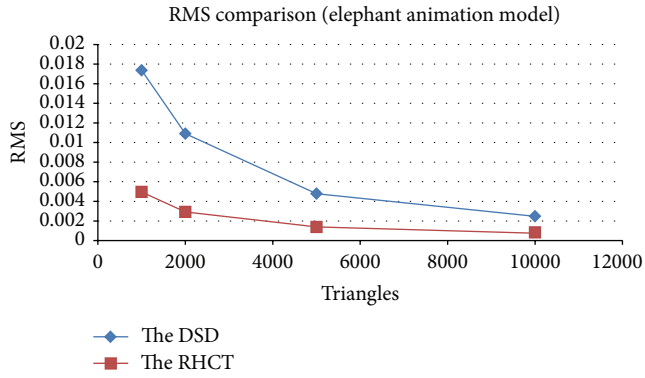
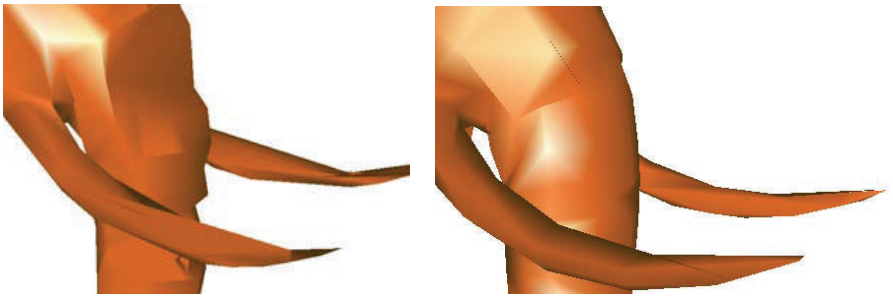


Figure 13: RMS error comparison for simplifying elephant animation model.



(a) Simplifying by the DSD (5000 triangles) (b) Simplifying by the RHCT (5000 triangles)

Figure 14: The elephant trunk is damaged and even hollow when simplifying by the DSD.

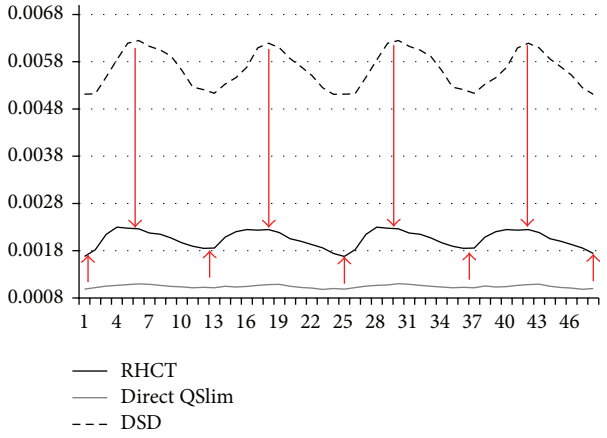


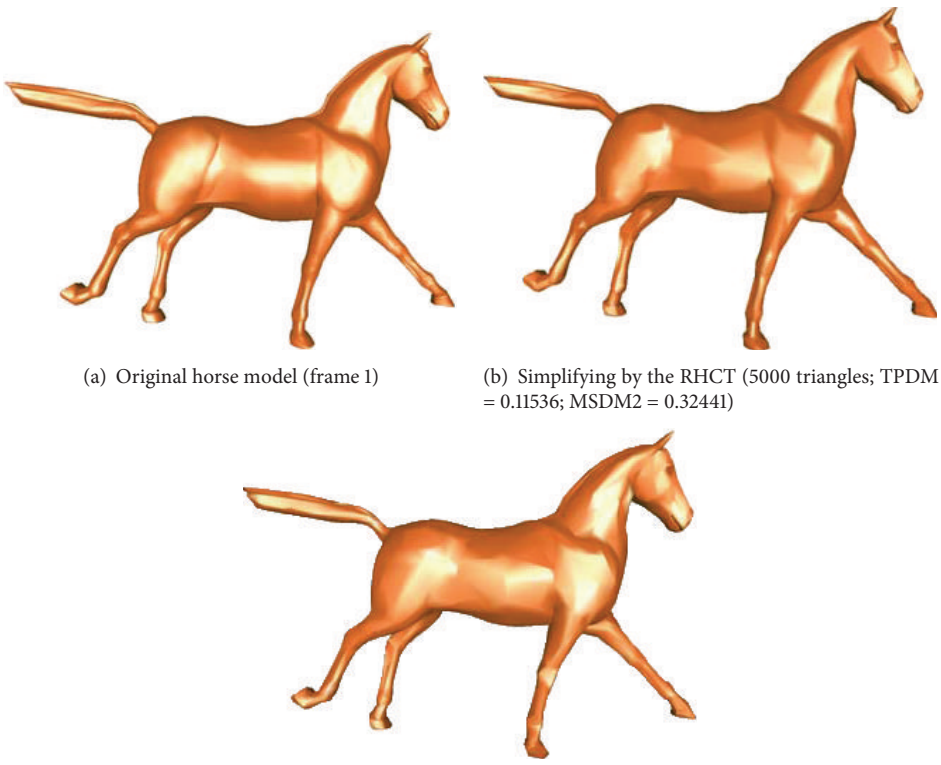
Figure 15: Simplification error per frame for a 2000-triangle approximation of a galloping horse.

to 5000 triangles and used TPDM and MSDM2 to measure the perception distortion, as shown in Tables 4 and 5 and Figure 16.

According to the results in Table 4, the average perception distortion caused by the simplification of the DSD was 0.14588, and the distortion was 0.12097 when the RHCT was employed, a value that was 17.07% smaller than that obtained using the DSD. In MSDM2, the DSD and RHCT produced 0.39796 and 0.33999 average perception distortions, respectively, when they were employed to simplify the horse model into 5000 triangles; in other words, the proposed method achieved improvements of 11.87% to 16.68% compared with the DSD.

6. Conclusions

The QEM method has recently been proven to be one of the most effective simplification methods. DSD uses summed QEM to simplify and reduce the storage cost of animation models. It then uses revised quadric matrices to resolve



(a) Original horse model (frame 1)

(b) Simplifying by the RHCT (5000 triangles; TPDM = 0.11536; MSDM2 = 0.32441)

(c) Simplifying by the DSD (5000 triangles; TPDM = 0.14417; MSDM2 = 0.38888)

Figure 16: Perceived quality comparison.

Table 4: Perceived quality comparison for simplified horse animation models using TPDM.

Frame	The DSD	The RHCT	Improvement rate
1	0.14417	0.11536	19.98%
2	0.14574	0.1169	19.79%
3	0.14756	0.11965	18.91%
4	0.14641	0.12315	15.89%
5	0.14726	0.12353	16.11%
6	0.14799	0.1246	15.81%
7	0.14695	0.12536	14.69%
8	0.14601	0.1232	15.62%
9	0.14547	0.1233	15.24%
10	0.14529	0.11978	17.56%
11	0.1431	0.11931	16.62%
12	0.14455	0.11749	18.72%
Average	0.14588	0.12097	17.07%

Table 5: Perceived quality comparison for simplified horse animation models using MSDM2.

Frame	The DSD	The RHCT	Improvement rate
1	0.38888	0.32441	16.58%
2	0.39903	0.3395	14.92%
3	0.40074	0.34292	14.43%
4	0.40216	0.34614	13.93%
5	0.40539	0.35548	12.31%
6	0.40217	0.35443	11.87%
7	0.39941	0.34918	12.58%
8	0.40186	0.34418	14.35%
9	0.39619	0.33311	15.92%
10	0.39283	0.3273	16.68%
11	0.3927	0.32993	15.98%
12	0.39418	0.33327	15.45%
Average	0.39796	0.33999	14.57%

the problem of homogeneous coordinates. However, the revised quadric matrices cannot be used to estimate a suitable position for the contraction vertex formed by a vertex pair, leading to severe deterioration of the outlines of simplified animation models. To solve this problem, this paper proposes using RHCT to determine the appropriate position of the contraction vertex. Experimental results indicated that the proposed method can reduce the errors caused by simplification and enables the shape features of simplified animation to be retained.

Conflict of Interests

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

Acknowledgments

The author thanks the anonymous reviewers for helping to improve this paper. The author also thanks Professors Chassery, Wang, and Torkhani for providing the TPDM codes. Additionally, this research was supported by the National Science Council, Taiwan, under Grants NSC98-2221-E-159-019 and NSC100-2622-E-159-004-CC3.

References

[1] G. K. L. Tam, Z.-Q. Cheng, Y.-K. Lai et al., "Registration of 3D point clouds and meshes: a survey from rigid to Nonrigid," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 7, pp. 1199–1217, 2013.

[2] Y.-Z. Song, D. Pickup, C. Li, P. Rosin, and P. Hall, "Abstract art by shape classification," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 8, pp. 1252–1263, 2013.

[3] M. Figueiredo, J. I. Rodrigues, I. Silvestre, and C. Veiga-Pires, "A topological framework for interactive queries on 3D models in the qeb," *The Scientific World Journal*, vol. 2014, Article ID 920985, 10 pages, 2014.

[4] G. Mei, "Summary on several key techniques in 3D geological modeling," *The Scientific World Journal*, vol. 2014, Article ID 723832, 11 pages, 2014.

[5] J. T. Wang, Y. C. Chang, C. Y. Yu, and S. S. Yu, "Hamming code based watermarking scheme for 3D model verification," *Mathematical Problems in Engineering*, vol. 2014, Article ID 241093, 7 pages, 2014.

[6] Y. Seol, J. Seo, P. H. Kimz, J. P. Lewisx, and J. Noh, "Artist friendly facial animation retargeting," *ACM Transactions on Graphics*, vol. 30, no. 6, article 162, 2011.

[7] D. Gerszewski and A. W. Bargteil, "Physics-based animation of large-scale splashing liquids," *ACM Transactions on Graphics*, vol. 32, no. 6, article 185, 2013.

[8] T. Weise, S. Bouaziz, and H. L. M. Pauly, "Realtime performance-based facial animation," *ACM Transactions on Graphics*, vol. 30, no. 4, article 77, 2011.

[9] S. Sun and C. Ge, "A new method of 3D facial expression animation," *Journal of Applied Mathematics*, vol. 2014, Article ID 706159, 6 pages, 2014.

[10] B.-S. Jong, J.-L. Tseng, W.-H. Yang, and T.-W. Lin, "Extracting features and simplifying surfaces using shape operator," in *Proceedings of the 5th International Conference on Information, Communications and Signal Processing*, pp. 1025–1029, Bangkok, Thailand, December 2005.

[11] 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.

[12] X.-D. Sun and H.-B. Zhang, "Fast simplification of scanned 3D human body for animation," in *Proceedings of the 8th International Conference on Computer Graphics, Imaging and Visualization (CGIV '11)*, pp. 70–75, Singapore, August 2011.

[13] M. Larkin and C. O'Sullivan, "Perception of simplification artifacts for animated characters," in *Proceedings of the 8th Annual Symposium on Applied Perception in Graphics and Visualization (APGV '11)*, pp. 93–100, August 2011.

[14] J. L. Tseng and Y. H. Lin, "Low-resolution surface simplification using shape operators with large-scale surface analysis," *WIT Transactions on Information and Communication Technologies*, vol. 58, pp. 105–113, 2014.

[15] E. Mendi, "A 3D face animation system for mobile devices," *Journal of Intelligent and Fuzzy Systems*, vol. 26, no. 1, pp. 11–18, 2014.

[16] 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.

[17] T. S. Gieng, B. Hamann, K. I. Joy, G. L. Schussman, and I. J. Trotts, "Constructing hierarchies for triangle meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 2, pp. 145–161, 1998.

[18] J. Rossignac and P. Borrel, "Multi-resolution 3D Approximations for Rendering Complex Scenes," in *Modeling in Computer Graphics: Methods and Applications*, pp. 455–465, 1993.

[19] A. Guéziec, "Surface simplification with variable tolerance," in *Proceedings of the 2nd Annual International Symposium on Medical Robotics and Computer Assisted Surgery*, pp. 132–139, 1995.

[20] B. Hamann, "A data reduction scheme for triangulated surfaces," *Computer Aided Geometric Design*, vol. 11, no. 2, pp. 197–214, 1994.

[21] 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 (IGGRAPH '97)*, pp. 209–216, August 1997.

[22] A. Mohr and M. Gleicher, "Deformation sensitive decimation," Tech. Rep., University of Wisconsin, 2003.

[23] M. Garland, A. Willmott, and P. S. Heckbert, "Hierarchical face clustering on polygonal surfaces," in *Proceedings of the Symposium on Interactive 3D Graphics*, pp. 49–58, March 2001.

[24] S. Kircher and M. Garland, "Progressive multiresolution meshes for deforming surfaces," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '05)*, pp. 191–200, July 2005.

[25] C. DeCoro and S. Rusinkiewicz, "Pose-independent simplification of articulated meshes," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D '05)*, pp. 17–24, April 2005.

[26] E. Landreneau and S. Schaefer, "Simplification of articulated meshes," *Computer Graphics Forum*, vol. 28, no. 2, pp. 347–353, 2009.

[27] T.-Y. Lee, C.-H. Lin, Y.-S. Wang, and T.-G. Chen, "Animation key-frame extraction and simplification using deformation analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 4, pp. 478–486, 2008.

[28] B. Merry, P. Marais, and J. Gain, "Simplifying character skins with analytic error metrics," *Computer Graphics Forum*, vol. 29, no. 1, pp. 13–24, 2010.

[29] J. Vince, *Rotation Transforms for Computer Graphics*, Springer, 2011.

[30] J. A. Vince, *Mathematics for Computer Graphics*, Springer, 2010.

[31] D. Simon, *Evolutionary Optimization Algorithms*, John Wiley & Sons, 2013.

[32] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998.

[33] F. Torkhani, K. Wang, and J.-M. Chassery, "A curvature-tensor-based perceptual quality metric for 3D triangular meshes," *Machine Graphics & Vision*, vol. 23, no. 1, 2014.

[34] G. Lavoué, "A multiscale metric for 3D mesh visual quality assessment," *Computer Graphics Forum*, vol. 30, no. 5, pp. 1427–1437, 2011.

[35] G. Lavoué, M. Tola, and F. Dupont, "MEPP-3D mesh processing platform," in *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications (GRAPP & IVAPP '12)*, pp. 206–210, SciTePress, Rome, Italy, February 2012.