Feature Aware Multiresolution Animation Models Generation

Shixue Zhang
Changchun Institute of Optics, Fine Mechanics and Physics, Changchun 130033, China
Email: shixuezhang@tom.com

Jinyu Zhao
Changchun Institute of Optics, Fine Mechanics and Physics, Changchun 130033, China
Email: zhaojy@ciomp.ac.cn

Abstract—In computer graphics, investigation has been made in many years in LOD (Level of Detail) models generation to alleviate the workload of visualization processing. However, most of the existing algorithms only focus on static models, and very little works are to deal with animation models. In this paper, we propose a feature aware method for generating multiresolution animation models. By the method we first use a normal deviation based quadric error metric to generate a base hierarchy, and then update the edge contraction order progressively from the first frame to the last one. Thus, we can get the simplified versions for each frame in the animation sequence. Finally, a mesh optimization method is proposed to allow visual distortion of the simplified models to be further alleviated. We demonstrate that our method works efficiently and can generate better approximations with well-preserved features for each frame than previous methods.

Index Terms—animation models, progressive mesh, edge collapse, level of detail

I. INTRODUCTION

In computer animation and virtual reality, polygonal mesh is most commonly used to construct three-dimensional models. Also, more and more time-varying surfaces, which are also called animation models, are frequently used from scientific applications to movies, games, and so on. In many cases, high-resolution models are required to present details and fine structures. However, some details might be unnecessary especially when viewing from a distance. Mesh simplification is a process of eliminating such unnecessary or redundant details from high-resolution 3D models by repeated removing vertices, edges, or faces. Most of the existing simplification algorithms are to deal with a single static mesh, and very little work has been proposed to address the problem of accurate approximation for time-varying surfaces. In this paper, we propose an improved method for generating progressive animation models.

To simplify animation meshes, one naive way is to simplify the models for each frame independently. This solution can generate the approximations with the minimum distortion. However, since it does not exploit the temporal coherence of the data, it can involve the unpleasant visual artifact, causing the surface to vibrate and twitch. So traditional simplification algorithm can’t be directly applied to time-varying models.

Some previous methods focus on preserving the static connectivity, i.e. the connectivity of the animation models remains unchanged for all frames. Such adaptations are inadequate and would cause arbitrarily bad approximations when deformation is highly non-rigid, since it does not take time-varying deformation into consideration.

We therefore propose a new method for generating multiresolution animation models. Our method is a better tradeoff between the temporal coherence and geometric distortion, i.e. we try to maximize the temporal coherence while minimizing the visual distortion during the simplification process. We propose the use of normal deviation to improve the quadric error metric of edge collapse operation. After getting the edge collapse sequence for the first frame, we can progressively get the edge contraction sequence for the subsequent frames by measuring the frame-to-frame deformation degree. Finally, an optimization algorithm is performed on the simplified models to further reduce its visual distortion. We demonstrate that this provides an efficient means of multiresolution representation of deforming meshes over all frames of an animation.

The rest of this paper is organized as follows: Section 2 will review the previous related work. Section 3 will mainly introduce the procedure of our algorithm in detail and discuss its advantage. Section 4 will show the experimental results and compare our method with previous methods. Finally, conclusion will be made and some future work will be given in Section 5.

II. RELATED WORK

A. Mesh Simplification

There are now extensive papers on the approximation of dense polygonal meshes by coarser meshes that preserve surface detail. These methods can be roughly divided into five categories: vertex decimation [1, 2],...
vertex clustering [3, 4], region merging [5, 6], subdivision meshes [7, 8], and iterative edge contraction [9, 10, 11, 12, 13, 14]. A complete overview of the methods has been given in [15, 16, 17]. Among these methods, the process of iterative edge contraction is predominantly utilized. Representative algorithms include those from Hoppe [11] and Garland [9]. In such methods, a simple multiresolution structure is generated on the surface that can be used for adaptive refinement of the mesh [18, 19]. Traditional mesh simplification algorithm works fine on a single static model, but it is unable to be directly applied to deforming surfaces since no temporal coherence has been considered.

B. Mesh Optimization

Most of the existing mesh optimization methods are based on the idea of local optimization and require an improvement of such mesh quality parameters as aspect ratio, area, etc. Hoppe et al. [20] described an energy minimization approach to solve the mesh optimization problem. The energy function consists of three terms: a distance energy that measures the closeness of fit, a representation energy that penalizes meshes with a large number of vertices, and a regularization term that conceptually places springs of rest length zero on the edges of the mesh.

Turk [21] proposed an approach for distributing a given number of points over a mesh surface evenly. These points are connected to one another to create a tiling of a surface that is faithful to both the geometry and the topology of the original mesh surface.

Recently, Garimella [22] proposed a numerical optimization method to improve the quality of complex polygonal surface meshes without an underlying smooth surface.

C. Approximation of Animation Models

Shamir et al. [23, 24] are the first to address the problem of simplifying deforming surfaces. They designed a global multiresolution structure named Time-dependent Directed Acyclic Graph (TDAG) which merges each individual simplified model of each frame into a unified graph. TDAG is a data structure that stores the life time of a vertex, which is queried for the connectivity update. Unfortunately this scheme is very complex, and can not be easily handled.

Mohr and Gleicher [25] proposed a deformation sensitive decimation (DSD) method, which directly adapts the quadric error metric(QEM) [9] by summing the quadric errors over each frame of the animation. The result is a single mesh that attempts to provide a good average approximation over all frames. Consequently this technique provides a pleasant result only when the original surfaces do not present strong deformation.

DeCoro and Rusinkiewicz [26] introduced a method of weighing possible configuration of poses with probabilities. With articulated meshes, skeleton transformation is incorporated into standard QEM algorithm, and users must specify the probability distribution for each joint. This method works quite well, but is limited to a very specific class of deformations.

Kircher and Garland [27] proposed a multiresolution representation with a dynamic connectivity for deforming surfaces. By their method, the simplified model for the next frame is obtained by a sequence of edge-swap operations from the simplified model at the current frame. They treat a sequence of vertex contraction and their resulting hierarchies as a reclustering process [15]. This method seems to be particularly efficient because of its connectivity transformation.

Huang et al. [28] proposed a method based on the deformation oriented decimation and dynamic connectivity update. They added deformation weight to the collapse cost to generate the unified collapse sequence, and then use vertex tree to further reduce geometric distortion by allowing the connectivity to change. Their method can provide a good approximation of deforming surfaces, but the update process requires a complex structure.

Recently, Landreneau et al. [29] propose a method for simplifying a polygonal character with an associated skeletal deformation. This method works well but can be only applied to articulated meshes.

III. OUR ALGORITHM

Our algorithm consists of three components: a basic simplification hierarchy, an update scheme, and a mesh optimization method. The basic simplification uses a normal deviation based quadric error metric; the update scheme is used to adjust the edge contraction sequence between adjacent frames based on the deformation information; and the optimization method is used to further minimize the geometric distortion. Next we will introduce the algorithm in detail.

A. Review of QEM

Our algorithm is based on the famous QEM metric, so before describing our method, we should first have a quick review of QEM [9]. QEM method iteratively selects an edge \((v_i, v_j)\) with the minimum contraction cost to collapse and replace this edge by a new vertex \(u\) which minimizes the contraction cost. For measuring the contraction cost of an edge, it utilizes the quadric error metric (QEM) to measure the total squared distance of a vertex to the two sets of planes \(P(v_i)\) and \(P(v_j)\) adjacent to \(v_i\) and \(v_j\), respectively. A plane can be represented with a 4D vector \(p\), consisting of the plane normal and the distance to the origin. Hence, the squared distance of a vertex \(v\) to a plane \(p\) equals \(v^T (pp^T)v\). The QEM cost function \(QEM_\theta\) for a vertex \(v\) to replace the edge \((v_i, v_j)\) is

\[
QEM_\theta(v) = \sum_{p \in P(v_i)} v^T (pp^T)v + \sum_{p \in P(v_j)} v^T (pp^T)v
= v^T Q v + v^T Q v
\]

Hence, the QEM cost \(QEM_\theta\) for contracting an edge \((v_i, v_j)\) is defined as \(QEM_\theta(u_i, u_j)\), in which \(u_j\) is the vertex minimizing \(QEM_\theta(u_i, u_j)\). This algorithm simplifies a mesh by iteratively finding the edge \((v_i, v_j)\) with the minimum \(QEM_\theta\), performing an edge-collapse operation to replace
(v_i, v_j) with a new vertex u_{ij} and updating the edge contraction costs related to u_{ij} until the desired vertex count m is reached.

B. Normal deviation Based Metric

After an edge collapse operation, the normal vectors of the incident triangles are changed. This deviation in normal field can be used to measure the error introduced in the mesh. The normal deviation across the triangle p is:

\[ D(p) = A(p) \cdot (1 - n^p \cdot n^p) \]  

where A(p) is current area of triangle p, n^p and n^p' are the normals of triangle p before and after the edge contraction respectively. And we add this deviation to the edge contraction calculation.

\[ COST_j(v) = \sum_{p \in P(v_j)} v^t (pp^t \cdot D(p))v + \sum_{p \in P(v_j)} v^t (pp^t \cdot D(p))v \]  

where D(p) is the normal deviation for triangle p. The total distance between the vertex v and the triangles in P(v_j) and P(v_j') is zero when the neighborhoods of v_j and v_j' are coplanar. In this case, the normal deviation caused by the edge collapse will also be zero. So our improved error metric does not discard one value when the other is zero.

C. Update Scheme

Once the coarse model of the first frame is obtained, we have to handle the meshes in the subsequent frames. If we directly apply this edge contraction sequence to all of them, it will inevitably produce poor approximations since no any deformation has been taken into consideration, and the features in the first frame might not be the features in other frames.

We therefore propose an edge sequence update scheme which is based on the deformation degree. In order to preserve the areas with large deformation, we should postpone the contraction of predominantly not be the features in other frames.

We first define density function for each edge and vertex. This density function, denoted as \( \rho \), is defined based on Gaussian and mean curvature. Let v be any interior vertex that does not lie on the boundary, v_1, v_2, ..., v_n are the ordered neighboring vertices (1-ring) of v (as shown in Fig. 1). We define the edge connecting v and \( v_i \) as \( e_i = v_i - v \), and \( \alpha_i = e_i \cdot e_i \) as the angle between two successive edges \( e_i \) and \( e_{i+1} \), \( n_i = e_i \times e_{i+1} / \| e_i \times e_{i+1} \| \) as the normal of triangle face (v, v_i, v_{i+1}). The dihedral angle \( \beta_i = n_i \cdot n_{i+1} \) at an edge \( e_i \) is the angle between the normals of the adjacent triangles. The discrete Gaussian curvature K and the absolute mean curvature H [29] with respect to the region S attributed to vertex v is then given by:
\[ K(v) = \frac{K}{\text{area}(S)} = \frac{2\pi - \sum_{i=1}^{n} \alpha_i}{\text{area}(S)} \]  \tag{5} \\
\[ H(v) = \frac{H}{\text{area}(S)} = \frac{\sum_{i=1}^{n} |e_i| \|\beta\|}{4 \cdot \text{area}(S)} \]  \tag{6} 

In this work the area of \( S \) is defined by using barycentric region which is one third of the area of the triangles adjacent to \( v \). The density function for vertex \( v \) is then defined by

\[ \rho(v) = \mu[K(v)] + (1 - \mu)H^2(v) \]  \tag{7} 

where \( \mu \) is a positive weight factor less than unity. For every edge \( e_{ij} = v_i - v_j \), we assume the density distribution linearly changes along the edge. Then the density of an edge can be expressed by the density of its midpoint \( \rho_{ij} = \rho(v_i) + \rho(v_j)/2 \). So we can define an edge marking function \([30]\) combining the length \( L_{ij} \) and density by:

\[ M_{ij} = \frac{1}{\sqrt{2\pi\sigma}} \left(1 - \exp\left(-\frac{\rho_{ij}^2 + L_{ij}^2}{2\sigma^2}\right)\right) \]  \tag{8} 

where \( \sigma \) is the Gaussian standard deviation. Based on the above density and marking function, we define a threshold \( T \). If the density value of a vertex is higher than \( T \), we should perform the vertex split operation on this vertex. One vertex split operation will increase the total number of vertices by 1. In order to maintain this number unchanged, we should perform a corresponding edge collapse operation. The edge with the lowest mark \( M_{ij} \) will be selected for contraction.

It should be pointed out that the threshold \( T \) should be great enough to guarantee that only very little connectivity update is operated, otherwise the temporal coherence can’t be maintained between adjacent frames. In general, we should guarantee that the percentage of connectivity update should be no more than 5%.

### E. Algorithm Outline

The algorithm can be quickly summarized as the following steps:

1) Calculate the normal deviation based edge contraction cost for the original animation mesh sequence.

2) Get the basic edge contraction order of the first frame model, and record it as a reference.

3) For a given frame, apply the edge contraction order in the previous frame while testing if the edge change error \( \eta \) is beyond the threshold \( \epsilon \). The edges with large deformation will not be contracted.

4) Continue the edge contraction operation using the improved quadric error metric until the desired resolution is reached.

5) Perform the mesh optimization algorithm to smooth the coarse models.

6) Repeat Step(3)-Step(5) until all the frames in the animation sequence have been handled.

### IV. EXPERIMENTAL RESULT

We test the result of our algorithm on a computer with Pentium4 3.2G CPU and 4G memory. We use OpenGL to render the models. The simplification of the lion-pose animation is shown in Fig. 2. Compared to the upper full resolution models, the bottom shows our simplified results when 90% of its components are reduced. We could see that most of the features in the original sequence are preserved.

In Fig. 3 we show our algorithm applied on a more complicated animation sequence. The top row shows the
original elephant-gallop models with 17489 vertices. The middle row and bottom row show the simplified models with 80% and 95% of its components removed respectively. We can see that even very little component is left, most of the outline features are preserved very well.

Next we show the comparison result between our method and Kircher’s method [27]. Fig. 4 shows the simplification result of the horse-to-human animation models. By enlarging the detail of some human features, we could see that our result (right) preserves the hands and feet better, moreover the triangulation of this area is more reasonable than Kircher’s result. In the area of the human’s feet, our result obviously contains more triangles. The reason is that the feet and hands are morphing from the hoof of the horse, and deformation degree is relative larger, so more edge collapse operation is postponed in such areas. Fig. 5 shows another example of the elephant-to-horse animation. Our result obviously preserves the features on the horse ear and tail better.

Fig. 6 shows the RMS error statistics of horse-gallop animation results using different methods. The RMS error line of our result (red) lies between DSD method and independent QEM method on the whole, and is obvious lower than other methods. So our method can be demonstrated to generate better result both in visual effect and error statistic.

To demonstrate the various applicability of our algorithm, we finally show a skinning animation in Fig. 7. It demonstrates an example of simplifying a facial expression animation. Even after removing 95% of the
Figure 6. RMS error metric results of the horse-gallop animation (48 frames). Vertical axis indicates the error value, and the horizontal axis indicates the animation frame.

Figure 7. A facial expression animation with 192 frames. Top: The original sequence with 23725 vertices and 46853 faces. Middle: 3201-vertices and 5825-faces approximation (80% reduced) using our method. Bottom: 1200-vertices and 1874-faces approximation (95% reduced).

vertices, the simplified meshes can still be rendered faithfully to the original ones.

V. CONCLUSION AND FUTURE WORKS

In this paper we propose a feature-aware method for generating progressive multiresolution animation models. Given a sequence of meshes that represent the time-varying data, our method can produce a sequence of the simplified versions that well approximate the original model at any given frame. We propose the use of normal deviation to calculate the edge contraction cost. The proposed method first generates a basic edge collapse sequence for the first frame. This basis edge contraction sequence is adjusted for the subsequent frames using an update scheme, which can preserve the deformation features while maintaining the frame-to-frame temporal coherence. A mesh optimization method is proposed to further reduce the visual distortion. Our algorithm is easy to implement, and can produce better results than previous methods.

There are certainly further improvements that could be made to our algorithm. For example, we believe that there must be a way to extend our algorithm to be view-dependent.

ACKNOWLEDGMENT

The authors wish to thank Scott Kircher for providing the horse-to-man and elephant-to-horse morphing...
REFERENCES


Shixue Zhang was born in Jilin province, China, in 1980. He received his Bachelor of Science degree from Jilin University in 2003, and his Ph.D. degree from Department of Computer and Information Science, University of Macau in 2009. He is currently an assistant researcher in Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His research interests include computer graphics and image processing.

Jinyu Zhao was born in Inner Mongolia province, China, in 1976. He received his Bachelor of Science degree from Inner Mongolia University of Technology in 1997, and his Ph.D. degree from Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, in 2006. He is currently an associate researcher in Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His research interests include electricity-optics measure and image processing.