Feature Aware Dynamic Mesh Approximation with Local Area and Deformation Control

Shixue Zhang¹, Qingyun Yang¹ and Wenhao Lu²*

¹Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, P.R. China
²Changchun University of Science and Technology College Of Optical And Electronical Information, Changchun 130022, P.R. China
*Corresponding author’s e-mail:34426797@qq.com

Abstract. In computer animation applications, simplifying the mesh is an important technique to reduce the workload of visualization processing. In this paper, we propose a new method to simplify the dynamic model using local area and deformation information control. Our method uses an improved quadratic error metric with the local distortion measurement as the basic level. In addition, the deformation parameters are defined to be embedded in the summarized quadratic error, so that the regions with large deformations during the animation can be successfully retained. The new method is efficient and easy to implement, and in average can generate high-quality animated models with well-preserved local and deformation features on each frame.

1. Introduction

In the fields of computer animation, from scientific applications to 3D animation (games, deformations, etc.), more and more dynamic meshes are widely used. In many cases, high-resolution geometric models are required to present details and fine structures. However, some details on the models may be unnecessary, especially when viewed from a long distance. Mesh simplification is the process of removing these unnecessary details from high-resolution 3D models by repeatedly deleting vertices, edges or faces. Most of the existing simplification algorithms are for a single static mesh, and little research has been done to solve the problem of accurate approximation of dynamic surfaces. In this paper, we propose a new method for generating the best multi-resolution dynamic meshes. Many approaches focus on maintaining static connectivity, that is, connectivity on a animation surface remains constant for all frames. Such adjustments are inadequate and lead to approximation differences in highly non-rigid deformations because it does not consider dynamic information. We propose a new method for generating the approximation for animated meshes. The method focus on preserve local features and deformation regions. It uses local distortion control to preserve highly curved areas, and also deformation weight is defined to be considered to the total cost, so that areas with greater deformation can be preserved. We prove that this provides an efficient multi-resolution representation for dynamic models on all frames of the animation.

2. Related works

2.1. Mesh simplification
There is now a wide range of papers involving approximating dense polygonal meshes through rough meshes that retain surface feature. These methods can be divided into five categories: vertex extraction [1], vertex clustering [2], region merging [3], subdivision mesh [4], and iterative edge contraction [5]. These methods have been comprehensively reviewed in reference [6, 7, 8]. In these methods, the process of iterative edge contraction or vertex remove is mainly used. Such mesh simplification method can't directly used to dynamic models, since the deformation information is not considered.

2.2. Mesh optimization
There are many optimization techniques for vertex relocation. Most of them are based on the idea of local optimization, and need to improve grid quality parameters such as aspect ratio, area, etc. Hoppe et al. [9] described an energy minimization method to solve the grid optimization problem. Their minimization algorithm divides the problem into two nested subproblems: internal continuous minimization and external discrete minimization.

2.3. Approximation of dynamic surfaces
Shamir et al. [10, 11] designed a global multi-resolution structure called time-varying directed acyclic graph (TDAG). The advantage of T-DAG is that it can handle arbitrary topological changes and geometric deformations. But this method has the disadvantages that it is non-incremental and space inefficient. Mohr and Gleicher [12] proposed a deformation-sensitive extraction (DSD) method, which directly adapted the Qslim algorithm by summing the secondary error of each frame of the animation [5]. The result is that a single grid tries to provide a good average approximation across all frames. As a result, this technique can provide pleasing results only when the original surface is not strongly deformed. DeCoro and Rusinkiewicz [13] introduced a method of weighing the possible configurations of poses with probability. When using an articulated mesh, the skeletal transformation has been incorporated into the standard QEM (quadratic error metric) algorithm [5]. Kircher and Garland [14] proposed a multi-resolution representation with dynamic connectivity for deformed surfaces. This method works well and can generate optimal animation models due to its connection transformation.

3. Our algorithm
Our algorithm consists of two parts: (1) Use the local area distortion measurement to improve QEM edge contraction cost. (2) Define a deformation degree weight during the whole animation to preserve the areas with large deformation.

3.1. Weighted quadric error metrics
So far, the QEM algorithm is widely considered as one of the most efficient methods to simplify static meshes. Before introducing our method, we should have a quick review of QEM.

The QEM method iteratively selects an edge \((v_i, v_j)\) with the minimum contraction cost to collapse and replace this edge with a new vertex \(u\) which minimizes the contraction cost. QEM measure the edge contraction cost as 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 \(\Delta_{ij}(v)\) for a vertex \(v\) to replace the edge \((v_i, v_j)\) is

\[
\Delta_{ij}(v) = \sum_{p \in P(v_i)} v^T (pp^T) v + \sum_{p \in P(v_j)} v^T (pp^T) v
\]

\[
= v^T Q_i v + v^T Q_j v \tag{1}
\]

QEM iteratively finds the edge \((v_i, v_j)\) with the smallest \(\Delta_{ij}\), performs edge contraction operations to replace \((v_i, v_j)\) with new vertices \(u_i\) and updates the edge shrinkage cost associated with \(u_i\) until the required resolution is reached to simplify the network grid. Arrivals. QEM can produce good results, but it ignores some basic features, such as curvature.
Next, we will define local distortion. Let $n$ be the surface normal at vertex $v$, let $k_1$ and $k_2$ be its principal curvatures, and $e_1$ and $e_2$ be their corresponding principal directions. In an orthogonal coordinate system whose origin is $v$ and the axes are $e_1, e_2, n$, the second-order local approximation of the surface is

$$S(x, y) = \left(x, y, \frac{k_1 x^2 + k_2 y^2}{2}\right)$$

$$(x, y) \in [-\omega_1, \omega_1] \times [-\omega_2, \omega_2]$$

By considering only the lower terms of the Taylor series approximation, we define $A$ is a diagonal matrix with entries

$$a_{11} = \frac{4}{3} \omega_1^3 \omega_2 k_1^2$$
$$a_{22} = \frac{4}{3} \omega_1 \omega_2^3 k_2^2$$
$$a_{33} = 4 \omega_1 \omega_2 - \frac{a_{11} + a_{22}}{2}$$

The area of the surface is

$$a_{11} + a_{22} + a_{33} = 4 \omega_1 \omega_2 \left(1 + \frac{\omega_1^2 k_1^2 + \omega_2^2 k_2^2}{6}\right)$$

Thus, the distortion factor between the area $4 \omega_1 \omega_2$ of the domain and the area of the surface patch is

$$\frac{a_{11} + a_{22} + a_{33}}{4 \omega_1 \omega_2} = 1 + \frac{\omega_1^2 k_1^2 + \omega_2^2 k_2^2}{6}$$

Observe that the area distortion depends on the term $(\omega_1^2 k_1^2 + \omega_2^2 k_2^2)/6$, which we call the local area distortion measure and can be computed by

$$\frac{a_{11} + a_{22}}{a_{11} + a_{22} + a_{33}} = \frac{\text{trace}(A) - a_{33}}{\text{trace}(A) + a_{33}}$$

Due to the invariance of the eigenvalues of a matrix under bijective linear transformations, the local area distortion measure can be computed in the canonical coordinate frame by

$$\frac{\text{trace}(A) - \lambda}{\text{trace}(A) + \lambda}$$

where $\lambda$ is $A$’s eigenvalue corresponding to the eigenvector closest to the surface normal at $v$.

In our algorithm, we can use an additional weight to improve the quadric $w_i Q_i + w_j Q_j$, where the weights $w_i$ are computed by

$$w_i = \frac{\text{trace}(A_i) - \lambda_i}{\text{trace}(A_i) + \lambda_i}$$

Consequently, the contraction cost of every edge $(v_i, v_j)$ is computed by

$$v_i^T (w_i Q_i + w_j Q_j) v + v_j^T (w_i Q_i + w_j Q_j) v$$

Moreover, to guarantee that the weight won’t be overestimated, the quadric assigned to the new vertex $v$ has to be

$$w_i Q_i + w_j Q_j$$

With this improved edge contraction cost, more local features can be preserved during the simplification.
3.2. Deformation degree measurement

The deformation-sensitive extraction (DSD) [20] algorithm solves this problem by summing the edge shrinkage cost of all frames. Based on the DSD method, we add extra weight to the DSD cost to measure the degree of deformation. Distortion is measured by changes in the cost of edge folding between frames.

For areas with large deformation, the change in collapse cost must be prominent, while in areas with less deformation, the collapse cost may change slightly. Therefore, the shrinkage cost of edge \((v_i, v_j)\) can be expressed as

\[
\text{cost}_{ij} = DSD_{ij} + k_1 \sum_{t=1}^{f} \left| \Delta_{ij}^t - \bar{\Delta}_{ij} \right|
\]

(11)

where \(\Delta_{ij}^t\) is the collapse cost of edge \((v_i, v_j)\) in frame \(t\), and \(\bar{\Delta}_{ij}\) is the average collapse cost of edge \((v_i, v_j)\) over all of the frames. And \(k_1\) is a user-specified coefficient to adjust the influence the deformation degree. In our experiment, we set \(k_1\) to around 1.

4. Experimental result

![Figure 1. Experiment result of horse-gallop sequence.](image1)

We implement our algorithm using MS Visual Studio developing tool. Figure 1 shows the animation of horse gallops. The total animation contains 48 frames. The first row shows the original animation models, while the second row shows the approximations with 90% vertex removed. And it is also very close to the original models. And most of the local and deformation features are well preserved.

![Figure 2. Experiment result of elephant to horse animation.](image2)
5. Conclusion
This paper proposes a new dynamic mesh simplification method. The method considers the local area features and deformation information. The experiment results demonstrate the proposed method can generate optimal simplified versions of the original animation models.

Acknowledgments
We would like to thank Scott Kircher for the elephant-to-horse sequence. This work is supported by China’s national high-tech research and development program.

References
[1] Schroeder, W.J., Zarge, J.A., Lorensen, W.E., (1992) Decimation of Triangle Meshes, ACM SIGGRAPH 1992 Conference Proceedings, 26(2), 65-70.
[2] Low, K.L., Tan, T.S., (2007) Model Simplification Using Vertex-Clustering, Proc. ACM Symp. Interactive 3D Graphics, 75-82.
[3] Garland, M., Willmott, A., Heckbert, P.S. (2001) Hierarchical Face Clustering on Polygonal Surfaces, Proc. ACM Symp. Interactive 3D Graphics, 49-58.
[4] Lee, A., Moreton, H., Hoppe, H. (2000) Displaced Subdivision Surfaces, ACM SIGGRAPH 2000 Conference Proceedings, 85-94.
[5] Garland M., Heckbert P. S. (1997) Surface Simplification using Quadric Error Metrics, ACM SIGGRAPH 1997 Conference Proceedings, 209-216.
[6] Garland, M. (1999) Multiresolution Modeling: Survey & future opportunities, Proceedings of Eurographics 1999, 49-65.
[7] Luebke, D., Reddy, M., Cohen, J. (2002) Level of Detail for 3-D Graphics, Morgan Kaufmann, San Francisco.
[8] Oliver, M. van K., Hélio, P. (2006) A Comparative Evaluation of Metrics for Fast Mesh Simplification, Computer Graphics Forum, 197-210.
[9] Hoppe, H., DeRose, T., Dunchamp, T., McDonald, J., Stuetzle W. (2003) Mesh Optimization, SIGGRAPH ’93 Conference Proceedings, 19-25.
[10] Shamir, A., Bajaj, C., Pascucci, V. (2000) Multi-resolution Dynamic Meshes with Arbitrary Deformations, IEEE Visualization 2000 Conference Proceedings, 423-430.
[11] Shamir, A., Pascucci, V. (2001) Temporal and Spatial Level of Details for Dynamic Meshes, Proceedings of ACM Symposium on Virtual Reality Software and Technology, 77-84.
[12] Mohr, A., Gleicher, M. (2003) Deformation Sensitive Decimation, Tech. rep., University of Wisconsin.
[13] Decoro C., Rusinkiewicz S. (2005) Pose-independent Simplification of Articulated Meshes, Proceedings of Symposium on Interactive 3D Graphics and Games, 17-24.
[14] Kircher, S., Garland, M. (2005) Progressive Multiresolution Meshes for Deforming Surfaces, Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 191-200.