

# Chapter 70

## 3D Model Simplification Method with Maintaining Local Features

Fangyi Liu, Xing Chen, Weiyuan Sun and Xingchen Gu

**Abstract** Current 3D model simplification algorithms only maintain the principal characteristics of the models, leading to the loss of the important characteristics of the local region. Therefore, these algorithms do not facilitate the human understanding of objects. The paper proposes a novel algorithm for simplifying 3D models, which can maintain the local important characteristics. The presented method combines the mesh segmentation algorithm and the simplification algorithm to extract the local visual features while ensuring the high fidelity of the whole mode. Experiments show the proposed algorithm is effectiveness.

**Keywords** Local feature • Mesh simplification • Mesh segmentation

### 70.1 Introduction

With the development of 3D scanning device and relevant technology, models are highly complex and detailed. However, the full complexity of such models is not always required, and it is useful to have simpler versions of complex models, as the computational cost of using a model is directly related to its complexity.

Nowadays, Triangular meshes are the most common drawing manner in computer graphics. Mesh simplification attempts to reduce its polygonal number while maintaining the appearance of the final object as much as possible. Since James Clark proposed the idea of 3D model simplification, global scholars have presented many model simplification algorithms. Typical algorithms [1, 2] include vertex deletion, vertex clustering, edge collapse and triangle collapse and so on. However, these simplification algorithms only focus on the fidelity of the whole model.

---

F. Liu (✉) · X. Chen · W. Sun · X. Gu  
School of Software Technology, Dalian University of Technology, Dalian, China  
e-mail: liufangyi911@gmail.com

According to the study of the psychophysics of human cognitive process, human understanding of things primarily concerned with the overall outline of things and then depends on the division that a complex object is divided into a few simple parts, and last focuses on the local features of each part. However, current 3D model simplification algorithms only focus on the principal characteristics of the whole models, ignoring the importance of local characteristics. How to extract the local features of the models is the key to this problem.

The paper proposes a novel algorithm for simplifying 3D models, which can remain the local important characteristics. The presented method combines the mesh segmentation algorithm and the simplification algorithm to extract the local visual features while ensuring the high fidelity of the whole mode.

## **70.2 Relevant Knowledge**

### ***70.2.1 Mesh Segmentation***

Mesh segmentation partitions a mesh into a set of connected and visually meaningful regions depending on specific geometric characteristics and topological characteristics.

According to the different segmentation goals, mesh segmentation are classified as patch-type segmentation and part-type segmentation. Patch-type segmentation creates disk-like patches which obey certain geometric properties such as planarity, size or convexity. Part-type segmentation is targeted more at partitioning the object defined by the mesh into meaningful components [3]. In order to detect the sharp regions of models, we use patch-part segmentation.

### ***70.2.2 Edge Collapse and Mesh Simplification***

Edge collapse, first proposed by Hoppe [4], is an optimization algorithm suitable to simplify any two-dimensional manifold mesh. This algorithm iteratively selects the edge with the least error for collapsing, removes all adjacent vertices and faces and re-triangulates the resulting hole. As the algorithm proceeds, the number of the vertices and faces reduces and the model can be simplified.

QEM algorithm, a kind of edge collapse algorithm proposed by Garland in 1997 [5], uses iterative contractions of vertex pairs to simplify models and maintains surface error approximations using quadric matrices. We use the QEM algorithm to simplify the no-sharp regions of models because this algorithm is able to simplify complex models quite rapidly and preserved the primary features of models.

For every model, each vertex is the solution of the intersection of a set of planes, namely the planes of the triangles that meet at that vertex. QEM algorithm

associates a set of planes with each vertex and defines the error of the vertex with respect to this set as the sum of squared distances to its planes:

$$\Delta(v) = \sum_{p \in \text{plane}(v)} (p^T v)^2 = \sum_{p \in \text{plane}(v)} v^T (pp^T) v = v^T \left( \sum_{p \in \text{plane}(v)} K_p \right) v \quad (70.1)$$

where  $p$  represents the plane defined by the equation:

$$ax + by + cz + d = 0 (a^2 + b^2 + c^2 = 1)$$

where  $K_p$  is the fundamental error quadric matrix of the plane  $p$ :

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (70.2)$$

Sum these fundamental quadrics together and represent an entire set of planes by a single matrix  $Q_v$ :

$$Q_v = \sum_{p \in \text{plane}(v)} K_p \quad (70.3)$$

For a given contraction  $(v_1, v_2) \rightarrow \bar{v}$ , the cost of the contraction is defined as  $\Delta(\bar{v})$ :

$$\Delta(\bar{v}) = (\bar{v})^T (Q_{v_1} + Q_{v_2}) \bar{v} \quad (70.4)$$

The error quadric matrix of  $\bar{v}$  is  $Q_{\bar{v}}$ :

$$Q_{\bar{v}} = Q_{v_1} + Q_{v_2} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \quad (70.5)$$

In order to minimize the computing works during choosing a position for  $\bar{v}$ , we select either  $v_1$  or  $v_2$  depending on which one of these produces the lowest value of  $\Delta(\bar{v})$ .

### 70.3 3D Models Simplification Method With Lossless Sharp Regions

In this section, a novel algorithm is proposed to simplify a model with keeping the local important features of the model, as shown in Fig. 70.1.

**Fig. 70.1** Shows the diagram for this approach



Starting from a complete 3D surface model represented by smooth triangle meshes, the improved watershed segmentation is used to segment the model. Next, simplify each part of the model using QEM simplification.

### 70.3.1 Mesh Segment

Watershed segmentation [6] is a simple and efficient algorithm for 3D model segmentation. However, watershed segmentation tends to cause over segmentation. The paper [7] improved watershed segmentation, using multi-level neighborhood of a point to estimate curvature, which made discrete principal curvature of a point reflect the geometric characteristics of models. Although the improved algorithm solved this problem, the improved algorithm often produces isolated patch. In order to achieve the meaningful segmentation of 3D models, isolated patch should be merged into the neighborhood patch. Thus, we improved watershed segmentation further.

*Curvature computing.* Compute the curvature at each vertex according to the following equation:

$$K_v = \frac{2\pi - \sum_n a_n}{S_{\Sigma}(v)} \quad (70.6)$$

*Initial Labeling.* Find the local minima and assign each a unique label. Next, find each flat area and classify it as a minimum or a plateau.

*Descent.* Loop through plateaus and allow each one to descend until a labeled region is encountered. And then allow all remaining unlabeled vertices to similarly descend and join to labeled regions.

*Region merging.* Merge regions whose watershed depth is below a preset threshold. After merging, every region is a basic patch.

*Basic patch detection.* For each basic patch, detect whether its borders are adjacent to the same basic patch.

*Patch generation.* If the borders of one basic patch are adjacent to the same basic patch, then it will be merged into its adjacent patch.

Information about edges and triangles generations.

### 70.3.2 Simplify Each Part of the Model Dependently

Select one part which is not simplified.

Simplify the part using QEM algorithm depending on the following steps.

- Step 1: Create the vertex set, edge set and triangle set of the part by reading the model file.
- Step 2: For each vertex, compute its error quadric matrix.
- Step 3: For each edge, compute its collapse cost.
- Step 4: Select an edge with the least QEM collapse cost for collapsing.
- Step 5: Compute the QEM collapse cost of the produced edge and update the edge set.
- Step 6: If simplification ration is larger than predetermined threshold, end the algorithm, or perform Step 4.

Predetermined threshold is defined as the ratio of the number of the vertices of the simplified model to that of the primitive model.

Check whether all the parts are simplified, if yes, then stop the algorithm; otherwise turn to 1.

## 70.4 Experiments

We tested our proposed algorithm using java program in Windows platform and use QEM algorithm as the control group. Examples of results are illustrated in Figs. 70.2 and 70.3.

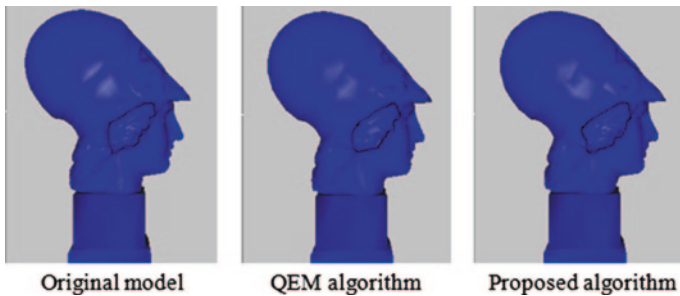


Fig. 70.2 Head model

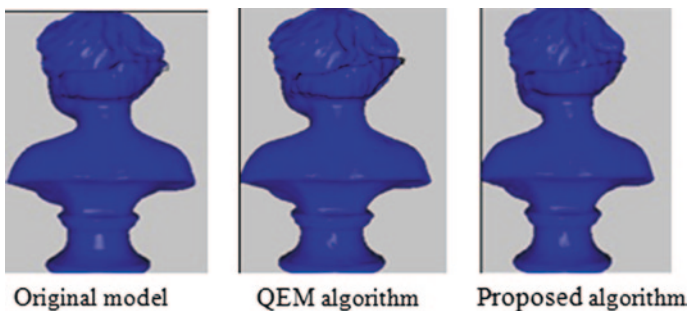


Fig. 70.3 Body model

In these models, we use the dark coil to mark the local part with the important local features of the model. Experimental results show that the approximations produced by our algorithm maintain higher fidelity to the original model than QEM algorithm.

## 70.5 Conclusion

According to the study of the psychophysics of human cognitive process, human understanding of things primarily concerned with the overall outline of things and then depends on the division that a complex object is divided into a few simple parts, and last focuses on the local features of each part. The paper proposes a novel algorithm for simplifying 3D models, which can remain the local important characteristics. The presented method combines the mesh segmentation algorithm and the simplification algorithm to extract the local visual features while ensuring the high fidelity of the whole mode. Experiments show the proposed algorithm is effectiveness.

## References

1. Dong F, Zhang RLY (2007) A summarization of mesh simplification algorithms. *J Chin Three Gorges Univ* 29(1):54–59
2. He H, Tian J (2002) A survey on mesh simplification. *J Softw* 13(12):2215–2224
3. Shamir A (2004) A formulation of boundary mesh segmentation. *3D Data Proc Visual Transm* 578:32–38
4. Hoppe H (1996) Progressive meshes. In: *Proceedings of SIGGRAPH 96*, vol 06. ACM Press, New York, pp 99–206
5. Garland M, Heckbert P (1997) Surface simplification using quadric error metrics. *Comput Graph (SIGGRAPH Proc)* 17:209–216
6. Mangan A, Whitaker R (1999) Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans Visual Comput Graph* 5(4):308–321
7. Liu Z, Tang Y (2010) 3D model simplification algorithm based on meaningful segmentation. *J Southwest Chin Norm Univ* 35(1):172–174