3D Mesh Simplification for Freeform Surfacing

J. Zhu¹, T. Tanaka² and Y. Saito³

Intelligent and Integrated Manufacturing,
Department of Mechanical and Control Engineering,
Tokyo Institute of Technology, Tokyo, Japan
Email: zhujiang@mep.titech.ac.jp

Abstract

Automatic simplification of highly detailed mesh models has been gaining considerable interest in the research field of reverse engineering and rapid prototyping recent years. However, almost all the present simplification algorithms are only focus on the geometry similarity. The triangle quality and the machinability of the mesh are less considered. We have developed a simplification algorithm from the character of freeform surfacing. The algorithm uses iterative edge collapse to simplify models, and use vertex importance value to control the error of the model as it is being simplified. It can rapidly produce faithful approximations of the triangle mesh models. Moreover, the simplified mesh generated by our algorithm is composed of regular shaped and regular sized triangles, which will enhance the model's machinability greatly.

Keywords: Mesh Simplification, CAD, Reverse Engineering, Freeform Surfacing

1 Introduction

Rapid prototyping manufacturing, and freeform surfacing often requires detailed geometric models for three-dimensional objects[1]. Such models are typically created using CAD software or 3D scanning systems by reverse engineering[2]. For efficient rendering, these models are typically converted to the triangle mesh approximations. Recently, with the rapid development of 3D scanning technology, an accurate representation of a 3D model can easily contain a million triangles. Complex triangle meshes are notoriously difficult to render, store, and transmit. Especially in the field of rapid prototyping manufacturing, over sampled mesh data will cause great inconvenience and low efficiency for later processing such as slicing. One approach to narrow this discrepancy is mesh simplification. Mesh simplification is the act of transforming a 3D mesh model into a simpler version. It reduces the number of triangles needed to represent a model while trying to retain a good approximation to the original shape and appearance[3].

In recent years, the problem of mesh simplification has received increasing attention. Several different algorithms have been formulated for simplifying mesh models, such as vertex clustering proposed by Rossignac[4], vertex decimation proposed by Schroeder[5], triangle decimation proposed by Gieng[6], edge collapse proposed by Hoppe[7], and vertex pair contraction proposed by Garland[8].

Noticed almost all the present algorithms are based on the application of computer graphics, what the algorithm interests are time efficiency and geometry likeness.

¹ - Doctor Course Student.
² - Research Associate, Ph. D of Engineer
³ - Professor, Ph. D of Engineer
However, for freeform surfacing, the triangle quality of the mesh models are also very important, they are expected to have regular shape and relatively small size range. Unfortunately, few algorithms were developed from the characteristic of manufacturing. The machinability of the approximation is less considered. In this paper, a mesh simplification algorithm based on vertex importance value and edge collapse transformation is introduced. In the algorithm, the special requirement for freeform surfacing is considered. The experiments results proved that the approximations not only keep the original model feature very well, but also provide high quality triangle mesh which is suitable for freeform surfacing. The remainder of the paper is organized as follows. We first discuss the basic requirement for freeform surfacing, then describe the basic simplification algorithm and data structure. Finally, the experiment results are given.

2 Basic Requirement for Freeform Surfacing
We propose to manufacture the 3D triangle mesh model by surface sculpture machining method [10]. This attempt brings some new tasks for the simplification algorithm. Originally, the artist used chisels to make the sculpture. They made some large flat faces first, and then refined the details. For surface sculpture machining, the large flat face is also welcomed. Firstly, to cut the model face-by-face from the raw material, the triangles of the mesh model are expected to have regular shapes, for instance an equilateral triangle is much easier to machine than the sliver triangle, whose vertices are almost collinear. Secondly, the sizes of triangles are also expected to have a small range. If some triangles used to represent the local detail are too small, while some triangles used to represent an approximate plane are too large, it will be very difficult for the tool to manufacture. Based on the above requirements, we developed a simplification algorithm, which can produce high quality approximations with high quality triangles. It provides a practical mix of quality and machinability.

3 Algorithm Description
A triangle mesh consists of a set of vertices and a set of triangles. Each vertex \( V \) specifies the \((x, y, z)\) coordinates of a point in space, and each face \( T \) defines a triangle by connecting together an ordered subset of the vertices. We consider in this paper only the special case of triangle meshes. However, arbitrary meshes can be easily converted into triangle meshes through a triangulation process.

3.1 Edge Collapse

![Figure 1: Edge Collapse.](image)

Our simplification algorithm is based on edge collapse transformation[7]. As shown in Figure 1. An edge collapse transformation \( \text{ecol}(\{V_s, V_t\}) \) unifies two adjacent vertices \( V_s \) and \( V_t \) into a single vertex \( V_s \). The vertex \( V_t \) and the two adjacent faces \( \{V_s, V_t, V_l\} \) and \( \{V_t, V_s, V_r\} \) vanish in the process. A position \( V_s \) is specified for the new unified
vertex. Therefore, the initial mesh can be simplified into a coarser approximation by applying a sequence of successive edge collapse transformations. After settled the basic transformation of the algorithm, an importance issue remains: how to select the candidate edge to collapse. In our algorithm, each vertex is assigned an importance value according to the geometry and topology feature of its neighborhood area. The vertex with the minimal importance value in the mesh will be firstly collapsed, and it will be collapsed with its nearest neighbor vertex. Therefore, there are two loops to search the candidate edge to collapse. The outer loop is searching among all the vertices in the mesh to locate the vertex with the minimal importance value. The inner loop is searching around the vertex’s neighbor vertices to find out the minimal length edge. For example in Figure 1, if $V_t$ has the minimal importance value of all the vertices, after searching all its neighbor vertices, the edge connects $V_t$ and its nearest neighbor $V_e$ will be selected to be the candidate edge.

### 3.2 Evaluation of Vertex Importance Value

![Figure 2: Local Curvature Based on Dihedral Angle.](image)

First of all, let us consider the local curvature value of a vertex. For two triangles in space $T_a, T_b$, if we know the normal vectors of these two triangles, the cosine value of the dihedral angle composed by these two triangles can be calculated as the dot product of the two normal vectors.

$$
cos(\angle(T_a, T_b)) = \frac{\text{Normal}(T_a) \cdot \text{Normal}(T_b)}{\parallel \text{Normal}(T_a) \parallel \parallel \text{Normal}(T_b) \parallel} \quad (0 \leq \angle(T_a, T_b) \leq \pi) \tag{3-1}
$$

The sharper the dihedral angel is, the higher the dot product is. For a vertex, if the dihedral angles of its around triangle pairs are all very large, almost 180 degree. The neighborhood area of that vertex can be considered as an approximate plane, and that vertex can be deleted firstly. Oppositely, if the dihedral angles of a vertex’s around triangle pairs are sharp, this neighborhood area must have a relatively high local curvature. The vertex should be preserved. Therefore, if vertex $V_i$ is surrounded by a triangle set $T_a, T_b, T_c, ..., T_e \in T$, the local curvature value of $V_i$ can be approximately represented by the triangle pair which have the minimal dihedral angles among all its around triangles. In order to unitize the curvature value, it can be described as follows.

$$
\text{Curvature}(V_i) = \max \left( \frac{\arccos(\text{Normal}(T_a) \cdot \text{Normal}(T_b))}{\pi} \right) \quad (T_a, T_b \in T, T_a \neq T_b) \tag{3-2}
$$

In order to generate a faithful approximation to the original model, the vertex with low local curvature value should be collapsed first, since it will cause less change to the model feature. On the other hand, considering the character of freeform surfacing, the edge with short edge length should also be collapsed first to generate regular shape triangle mesh. Therefore, in order to find a mix of both quality and machinability, the vertex importance value should be given as follows:

$$
\text{Importance} (V_i) = \text{Curvature}(V_i) \times \text{Edgelength}(V_i) \tag{3-3}
$$
where $\text{Edgelength}(V_i)$ is the length between vertex $V_i$ and its nearest neighbor vertex.

From equation (3-3), it can be found that the vertex importance value not only reflects the local model feature, but also eliminates the generation of sliver triangles.

The remaining problem is how to place the new vertex after the collapse transformation. One solution is using the optimal placement strategy; another one is using the original vertices subset as the new vertex position. Considering the complex models usually have tremendous data quantity, using the original vertices subset will highly increase the simplification efficiency. Therefore, one vertex of collapsed edge, which has a higher importance value, is selected as the position of new vertex.

### 3.3 Boundary Maintenance

Here in freeform surfacing application, only the solid mesh model is considered. The open boundary surface can be easily converted into a solid mesh model through a triangulation process.

### 3.4 Basic Data Structure

Data structure has important influence to the efficiency of the algorithm. Using too simple data structure to represent the mesh model, cannot access the geometry element among the mesh efficiently. However, using too complex data structure will cost great time to adjust the data after each simplification transformation. Therefore, finding a proper data structure to represent the vertices, edges and faces of the mesh is necessary. The data structure used in this algorithm is listed as follows.

1. Vertex List. It records the (x, y, z) coordinate of the vertex, the pointers of its neighbor vertices and adjacent triangles, and the geometry attribute of the vertex, such as nearest neighbor vertex pointer, importance value and so on.

2. Triangle List. It records pointers of the triangle’s three vertices, and the geometry attribute of the triangle, such as normal vector, surface area, and so on.

3. Collapse Record List. It records the pointers of two vertices to collapse and the pointer of the new vertex. It is a sequence of collapse records, and will be useful for the reverse operation or reconstruction of the original mesh model.

### 3.5 Validity Check

Some times in the simplification program, the edge collapse transformation will cause some undesirable inconsistencies to the mesh model. The most common problem is the mesh inversion shown in Figure 3.

For example, $\{V_i, V_j\}$ is the candidate edge to collapse. Notice the triangle highlighted by the red line. After the edge collapse transformation, that triangle will fold over on
itself. And this will produce a crack on the model surface. It is not permitted during the simplification process; it must be prevented. Therefore, before the edge collapse is executed, the validity check process is necessary to add into the program. If a face’s normal vector changes more than some significant threshold, it is regarded to fail the validity check, and that collapse is discarded. The program will continually pick the next proposed candidate edge to excuse the validity check, until the proposed edge passes the validity check. If none of the edge existed in the approximation passes the validity check. The program will stop and output the simplified model. In fact, to prevent fold-over phenomenon, a careful selection of the threshold is required. Here in the algorithm, the threshold is selected as \( \pi /3 \) to make experiments.

Sliver triangles, which have very small angles, are also undesirable in our application. Here a measure of triangle compactness is introduced [9],

\[
\gamma = \frac{4\sqrt{3} \omega}{l_1^2 + l_2^2 + l_3^2}
\]

where \( l_i \) are the lengths of the edges and \( \omega \) is the area of the triangle. This will assign a compactness of 1 to an equilateral triangle and 0 to a triangle whose vertices are collinear. Using this heuristic, we can also regard a collapse fail the validity check, which produce triangles whose compactness \( \gamma \) falls below some threshold.

### 3.6 Algorithm Summary

Having described the specific details of the algorithm, the algorithm process can be presented as the following outlines:

1. Read the initial mesh model.
2. Calculate the importance value of each vertex.
3. Make an order list for all the vertices by the importance value.
4. Repeat the following steps until the desired approximation is reached.
   - Select the top vertex from the order list, and take the edge with its nearest neighbor as the candidate edge to collapse.
   - Validity check for the candidate edge. If failed, select the next vertex from the order list, until there is one vertex passes the validity check.
   - Perform the edge collapse.
   - Select one vertex from the original vertices pair as the position of the new generated vertex by the importance value.
   - Recalculate the importance values of the neighbor vertices affected by this transformation.
   - Update the order list.

### 4 Experiment Results

The algorithm is implemented on a P4 2.8GHz PC, using JAVA language. Experiments with models of cow, horse and bunny are executed; the results are shown in Figure 4, 5, 6. In general, we let the simplification proceed until no more legal edge collapse transformations are possible. In order to obtain a visually aesthetic approximation, we stopped at 500 faces. Figure 4, 5, 6.a represent the original models. Figure 4, 5, 6.b represent the significant simplifications of the models using only 1000 triangles, the features of the original models are kept very well. Figure 4, 5, 6.c present more drastic approximations, using only 500 triangles. While most of the detail of the models has disappeared, the basic structures of the objects
are still intact. The major features such as head, legs, ears are all apparent, although highly simplified.

On the other hand, in the result shown in Figure 4, 5, 6, it can be found that the algorithm always tries to use approximately equilateral triangles to represent the approximations. Moreover, the triangles in the simplified mesh also have a regular size. The difference between the largest triangle and the smallest triangle won’t runs too far. This will do a great favour for later freeform surfacing process. Noticed there is one exception, which is on the leg part of the cow and horse, small triangles are used. However, this is to keep the contour of the legs. Using larger triangles will increase the legs sizes.

5 Error Analysis
As stated earlier, the primary aim of simplification is to produce an approximation which is as similar as possible to the original. In order to assess the quality of an approximation, we need some means of quantifying the notion of similarity. We have
chosen a metric $E_{\text{max}}$, which measures the maximal distance between the approximation and the original model, and a metric $E_{\text{avg}}$, which measures the average squared distance between the approximation and the original model. We define the approximation error $E_{\text{max}}(M_1, M_2)$ and $E_{\text{avg}}(M_1, M_2)$ between the original model $M_1$ and the approximation $M_2$ as:

$$E_{\text{max}}(M_1, M_2) = \max_{v \in X_1} \max_{v \in X_2} \{ d_v(M_2), d_v(M_1) \}$$

(5-1)

$$E_{\text{avg}}(M_1, M_2) = \frac{1}{k_1 + k_2} \left( \sum_{v \in X_1} d_v^2(M_2) + \sum_{v \in X_2} d_v^2(M_1) \right)$$

(5-2)

Where $M_1$ and $M_2$ are sets of vertices sampled on the models $M_1$ and $M_2$ respectively. $k_1$ and $k_2$ are numbers of vertices in subset $M_1$ and $M_2$. The distance $d_v(M)$ is the minimum distance from $v$ to the closest vertex of $M$.

The $E_{\text{max}}$ metric provides absolute distance bounds, which is a useful error guarantee, but it can be overly influenced by noise and local deviations. On the other hand, the $E_{\text{avg}}$ metric provides a better estimate of the overall fit and is more tolerant of noise, but it may discount local deviations. Neither of these error metrics is completely ideal. Therefore, a combination of these two metrics is preferable. We use these metrics for evaluation purposes only; they play no parts in the simplification algorithm. Figure 7 shows $E_{\text{max}}$ and $E_{\text{avg}}$ analysis of the simplification experiments described in the previous section.

![Figure 7: Error Analysis.](image)

6 Conclusion
Our algorithm is developed based on the application of freeform surfacing. It provides a mix of approximation quality and mesh triangle quality. Our algorithm uses iterative edge collapse to simplify models and use vertex importance value to control the error of the model as it is being simplified. The experiment results proved that the approximations not only keep the original model feature very well, but also provide high quality triangle mesh which is suitable for freeform surfacing. The algorithm performs well on both high curvature models, such as cow and horse, and low
curvature models, such as bunny. We envision that the algorithm presented in this paper will not only contribute to manufacturing, including freeform surfacing and rapid prototyping, but also have applications in computer graphics.

Acknowledgement
I would like to thank the Stanford University Computer Graphics Laboratory, for the horse and bunny rabbit models used in the experiments. The cow model is from Mr. Garland's homepage.

Reference


