Geometry Modeling for Virtual Reality Based on CAD Data

Yong Tang*, Yan Xu and Li-li Yuan

College of Automobile and Traffic Engineering, Nanjing Forestry University, Nanjing 210037, China

Abstract: In order to satisfy the requirement of geometry modeling for virtual reality system, we process a modeling method based on CAD data. Firstly, the media file that is exported from the CAD data is parsed to gain the mesh data. And then the mesh data was simplified through the improved quadric error metrics algorithm, which makes better shape of triangle. At last, the mesh data is wrote to standard virtual reality model with lever of details via OpenFlight API. Experiments show that triangular mesh shape by our algorithm is more reasonable than QEM algorithm, but with higher quality of rendering. With large simplification, rendering frame rate and loading efficiency of model are high improved, and visualization of the virtual reality model almost have no change, which will be sufficient to meet the demand of virtual reality system.

Keywords: Virtual reality, CAD data, quadric error metrics, mesh simplification, model conversion.

1. INTRODUCTION

Geometrical modeling is a basic work in Virtual Reality (VR) system. The number and the quality of patches of geometrical models directly exert an effect on the rendering efficiency and scene reality of VR system. Now commercial software is generally used to construct 3D models manually in VR system. But this method is laborious and of high human cost, which also has high requirements on skills [1]. In some special applications, especially in the field of industrial simulation, special soft and hardware are used for automatic 3D modeling. Techniques such as visual geometrical measurement, 3D laser scanner or depth camera are used to construct point cloud for 3D reconstruction [2]. In fact, there have been CAD data in productive enterprises and design departments. Therefore, if they can directly transform the available CAD data to geometrical models for VR system, modeling efficiency can be greatly improved and model precision can also be ensured. So there are many researches on using CAD data for geometrical modeling in VR system now. For examples, Wan Bi-le et al. [3] studied on obtaining geometrical model data required in virtual assembly environment through CAD software interface through CAD software interface; Sun Li-juan et al. [4] achieved transformation from CAD data to VR models and real-time visual by using MEMPIS middle ware. Literature [5, 6] successfully applied geometrical models and assembly information into distributed virtual assembly system by extracting CAD data. In addition, Atul Thaku et al. [7] also made conclusions and contrast of VR geometrical modeling methods based on current CAD data. Apart from research institutes, there are also some commercial companies are engaged in this work. For example, Spatial Corporation and ITI Corporation provide commercialized modules or software for geometrical modeling, to achieve for model transformation, simplification, restoration and optimization of CAD data.

There are various transformation methods for conversion from available CAD data to VR models, but most of them serve specific systems and requirements. In CAD software, mathematical version is used to express geometrical information of models. Though geometrical profiles of geometrical models directly exported from CAD software are very precise, the number of triangle is very large, which brings great computational burdens to real-time rendering in VR system. Some algorithms are used for mesh simplification in some applications and researches, but the simplification algorithms are generally applicable to commonly-used irregular geometrical models while it is quite possible that some very important geometrical elements for design features should be deleted in the simplification process of CAD data. Some boundary features of CAD models will be destroyed, which also tends to generate abnormal triangle, making shapes or distribution of triangles unreasonable and influencing visual effects [8]. In order to achieve a rapid and efficient method for VR modeling based on CAD data, the quadric error metrics algorithm (QEM) is modified for mesh simplification, and the OpenFlight API is used to export the mesh data to standard VR model.

2. IMPORT GEOMETRICAL DATA FROM CAD MODEL

The first step of model simplification is getting geometrical data. To simplify the process, first, the thesis directly export the CAD data into VRML (virtual reality modeling languages) neutral model format (CAD software such as CATIA and Pro/Engineer all have the function); then analyze the neutral files to obtain geometrical vertexes and patch information of original CAD models. VRML is short for virtual reality modeling languages [9], whose format actually is a kind of simply defined text file, the extension of file is “.wrl” and the file content mainly consists of different kinds
of nodes which used to illustrate head, data annotation, node, domain, event and route. The following is VRML node schematic diagram [10], as shown in Fig. (I).

![VRML node schematic diagram](image)

**Fig. (I). Node of VRML format.**

The 3D models in VR system are generally described by point, line and face. We used C++ programming language to analyze the model file. Therefore, in programming, design model class, triangle faces class and vertex class should be handled respectively. To accelerate loading speed and reduce memory usage, we set up index number for every node, then we can gain each triangle vertex index and 3D coordinate corresponding to each vertex by looking through index. As for a simple CAD model, data of vertexes in point node field are actual coordinates of model vertexes. But child nodes of complex assembly parts may be involved with scale, translation and rotation transformation, so subordinate vertex coordinates should be transformed into actual coordinates of model vertexes through coordinate transformation.

3. MESH SIMPLIFICATION

3.1. Modified QEM Algorithm

In the field of mesh simplification, Garland’s Quadric Error Metrics (QEM) simplification algorithm [11] is a classic simplified algorithm, which was first applied in a 3D game. In the algorithm, the quadratic sum of distance from the new vertex after collapsing the two sides to the plane which is adjacent to the two points of the side before collapsing as the error metrics, namely using local error to measure the side collapse cost, sorting collapse costs and making collapsing strategies. As for common irregular models like 3D reconstruction of point cloud, mesh simplification effect by using QEM algorithm is good, but as for regular CAD models, it’s easy to generate abnormal long and narrow triangles in the process of simplification. Therefore, the thesis makes some modifications based on QEM algorithm, adding morphologic change cost of triangle faces in collapse cost to improve simplification strategies of triangles and make shapes and distribution of model triangle faces more reasonable in the process of simplification.

Since the basic idea of QEM algorithm is side collapse, two vertexes (A and B) of one side are merged each time to generate a new vertex (V). The number of triangles reduces by one for every collapse process, as shown in Fig. (2).

![Fig. (2). Principles of edge collapse.](image)

In QEM algorithm, it is quadric error cost to determine whether the side can be collapsed and the collapse order. The detailed algorithm is as follows:

Suppose $v = [v_x, v_y, v_z]^T$ is a vertex in the mesh, $p$ is a plane in the 3D space, and the coefficient of the plane equation is $p = [a, b, c, d]^T$, then the square of distance from point $v$ to plane $p$ is:

$$d^2(v) = v^T K_p v$$

where

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

Make those adjacent to point $v$ belong to all triangle plane set $\text{planes}(v)$ making quadric error metrics of point $v$ the qua-dratic sum of distance from $v$ to these triangle faces, namely:

$$\Delta'(v) = \sum_{p \in \text{planes}(v)} d^2(v) = v^T (\sum_{p \in \text{planes}(v)} K_p) v$$

Make

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

Then $Q'(v)$ is $4 \times 4$ symmetric matrix, and make it quadric error metrics matrix of point $v$ to measure side collapse cost. The collapse cost of side $(v_x, v_y)$ collapsed into a new vertex is $\Delta'(\bar{v}) = \bar{v}^T (Q' + \bar{Q}') \bar{v}$ and $Q' + \bar{Q}'$ is used to represent the quadric error metrics of the new vertex $\bar{v}$.

The core of side collapse algorithm is determining or selecting the position of a vertex after side collapse. The general handling method is selecting two endpoints of collapsed side or some point in the ligature of two endpoints as the new vertex, then computing and sorting the collapse
costs of all the candidate points, and making the point of minimum cost the final new vertex.

The experiment found that when simplifying mesh data read out of CAD data, collapse costs tend to be the same or similar. There are several triangle patches in Fig. (3), when side AB is the collapse side, taking different points in side AB as collapse points, quadratic error costs of all the collapse sides are similar and the number of patches after simplification is also the same, but final shapes of triangles after simplification differ greatly. It can be seen that when a side is collapsed to the midpoint of side AB, the shape of the triangle is the best and all the triangles tend to be equilateral triangles, which is beneficial for calculating illumination in graphics engine and improving the quality of rendering.

\[
2 = \frac{4\sqrt{3}A}{l_0^2 + l_1^2 + l_2^2} \quad (5)
\]

\( A \) is the triangle area in formula (5). \( l_0, l_1, l_2 \) are three sides of the triangle, and \( c \) is triangle compactness ranging from 0 to 1. When the triangle tends to be an equilateral triangle, the value of \( c \) tends to approach 1, and when the triangle tends to be long and narrow, the value of \( c \) tends to approach 0. Then we define some vertex \( v \) as the average compactness of all triangle faces in which the vertex is in, namely:

\[
C(v) = \frac{\sum_{p \in \text{plane}(v)} C(p)}{n} \quad (6)
\]

\( n \) is the number of triangles, and \( C(p) \) is the compactness of the triangle \( p \). When computing and sorting the collapse cost, if collapse costs of several new vertexes are the same or similar, then compute and sort the compactness of new vertexes, make the vertex of maximum compactness the final new collapse vertex, and push the collapse cost into the queue. The final algorithm flow of model mesh simplification is as follows in Fig. (4).

![Fig. (3). Simplification effects at the same edge cost.](image1)

![Fig. (4). Algorithm flow of model simplification.](image2)

In the simplification process of patches, after sorting the sides, we can set simplification objectives, namely simplification rate or the number of patches after simplification to generate mesh models of different simplification rate to meet requirements for Lever of Details models (LOD) in VR system.

4. GENERATE VIRTUAL REALITY STANDARD MODEL

In virtual reality system, industrial standard model format of geometrical models is OpenFlight format. OpenFlight uses geometrical hierarchical structures, node base and node attributes to describe 3D bodies. The model includes root nodes, group nodes, object nodes subordinate to group nodes, face nodes subordinate to object nodes and other special nodes (such as DOF node, light dot node, etc.). Users are allowed to operate and manage the nodes through visual interface to ensure the rationality of model structure and modeling efficiency. By using OpenFlight API interface function, we can directly transform simplified data into OpenFlight standard model format through program interface. The procedure of establishing standard OpenFlight model file is as follows (See Fig. 5). We went through procedures such as program initialization, establishing head nodes of database, establishing group nodes, object nodes and face nodes, setting up node attributes, binding the hierarchical relationship
among nodes, and saving and closing database to achieve the save and read-in of models [13].

5. EXPERIMENTS

To verify the validity of the method mentioned in the thesis, we verified the final VR model effect though two experiments:

5.1. Contrast Experiments Among Simplification Algorithms

QEM algorithm and modified algorithm QEM algorithm are used respectively to simplify a simple part model (shaft), and then we make a subjective contrast of the final distribution of triangle mesh and visual effects of illumination. The experimental results are as follows (See Fig. 6). The number of original model patches is 23716, meshes are intensively distributed, and rendering effect is good. QEM algorithm and the algorithm proposed in the thesis are used respectively to simplify the model (all the simplification rates are set as 90%). We can see from the Figure that under the condition of the same simplification rate, light uniformity, shapes of meshes and distribution of models are better after using the simplification algorithm proposed in the thesis.

In the second experiment, a complex CAD final assembly model (cockpit) is tested. Virtual engine - Vega Prime is used to load and render models of different simplification rate, and make a contrast of real-time rendering frame rate, loading time of models and sizes of model files. The final result is as follow in Table 1.

It can be seen from the Table 1 that rendering frame rate and loading efficiency of the simplified generated VR model in virtual engine have been greatly improved, which can bet-

ter meet the requirements of virtual reality system. In addition, file size in hard disk was also effectively reduced, which is good to memory and transmission in distributed

![Fig. (5). Procedures of establishing open flight models.](image)

![Fig. (6). Model simplification result of QEM algorithm and modified QEM algorithm.](image)

<table>
<thead>
<tr>
<th>Simplified ratio(%)</th>
<th>Triangle count</th>
<th>Frame rate(fps)</th>
<th>Load time(s)</th>
<th>File size(m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>47512</td>
<td>66</td>
<td>0.53</td>
<td>6.4</td>
</tr>
<tr>
<td>50</td>
<td>23189</td>
<td>79</td>
<td>0.47</td>
<td>3.7</td>
</tr>
<tr>
<td>60</td>
<td>8852</td>
<td>83</td>
<td>0.31</td>
<td>2.3</td>
</tr>
<tr>
<td>70</td>
<td>6189</td>
<td>95</td>
<td>0.26</td>
<td>2.0</td>
</tr>
<tr>
<td>80</td>
<td>4339</td>
<td>102</td>
<td>0.19</td>
<td>1.3</td>
</tr>
<tr>
<td>90</td>
<td>4672</td>
<td>117</td>
<td>0.11</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Fig. (7). Real-time rendering effects in virtual engine.

CONCLUSION

As for modeling requirements in the VR system, we introduce triangle compactness cost based on the existing mesh simplification algorithm, achieving a rapid VR modeling method based on CAD data, which can generate levels-of-detail virtual reality models. Compared with other similar methods, after simplification, distribution shape of model triangle mesh becomes more reasonable. When visual effects almost have no change, both rendering efficiency and loading time are greatly improved, and model files account for little hard-disk space, which is beneficial for storage and network transmission. The method can be seen as a supplementary method for modeling in virtual reality system to improve modeling efficiency.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

The research has been supported by the fund of natural science fund for colleges and universities in Jiangsu Province (14KJB520017), project of Jiangsu Students’ innovation and entrepreneurship training (201510298028Z).

REFERENCES


