Send Orders for Reprints to reprints@benthamscience.ae

The Open Cybernetics & Systemics Journal, 2015, *9*, 2339-2343

Open Access

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, this study processed a modeling method based on CAD data. Firstly, the media file that is exported from the CAD data was parsed to gain the mesh data. Following this, the mesh data was simplified through the improved quadric error metrics algorithm, to obtain a better shape of triangle. At last, the mesh data was analyzed to establish a standard virtual reality model with Level of Details via OpenFlight API. Experiments showed that the triangular mesh shape obtained by using the proposed algorithm is more reasonable than QEM algorithm, but had higher quality of rendering. With extensive simplification, rendering frame rate and loading efficiency of model were observed to be highly improved, and visualization of the virtual reality model experienced almost 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 task involved in Virtual Reality (VR) system. The number and the quality of patches of geometrical models directly exert an influence on the rendering efficiency and scene reality of VR system. Nowadays, 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 requires f great skills [1]. In some special applications, especially in the field of industrial simulation, special software 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, CAD data are used in productive enterprises and design departments. Therefore, if they can directly transform the available CAD data into the geometrical models for VR system, modeling efficiency can be greatly improved and model precision can also be ensured. Recently, many researches have been conducted based on CAD data for geometrical modeling in VR system . For examples, Wan Bi-le *et al.* [3] studied how to obtain geometrical model data required in virtual assembly environment through CAD software interface. Sun Lijuan *et al.* [4] achieved transformation of CAD data into VR models and real-time visualization by using MEMPHIS 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 regarding VR geometrical modeling methods based on current CAD data. Apart from research institutes, there are also some commercial companies engaged in this work. For example, Spatial Corporation and ITI Corporation provide commercialized modules or software for geometrical modeling to achieve model transformation, simplification, restoration and optimization of CAD data.

There are various transformation methods available for conversion of CAD data into VR models, but most of them have 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 triangles is very large, which brings great computational burden to real-time rendering in VR system. Some algorithms have been 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 likely that some very important geometrical elements for design features could be deleted in the simplification process of CAD data. Some boundary features of CAD models are destroyed, which also tends to form an 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 to obtain geometrical data. To simplify the process, first, the thesis directly exported the CAD data into VRML (virtual reality modeling languages) neutral model format (CAD software such as CATIA and Pro/Engineer all have the function) and analyzed the neutral files to obtain geometrical vertexes and patch information of original CAD models. VRML is short for virtual reality modeling languages [9], the format of which is actually a kind of simply defined text file, with the extension of file being ".wrl" and the file content mainly

*Address correspondence to this author at the College of Automobile and Traffic Engineering, Nanjing Forestry University, Nanjing, 210037, P.R. China; Nanjing, 201137, P.R. China; Tel: 025-85428593;
E-mail: tangyong159@163.com

1874-110X/15 2015 Bentham Open

consists of different kinds of nodes used to illustrate head, data annotation, node, domain, event and route. The following is the VRML node's schematic diagram [10], as shown in Fig. (1).

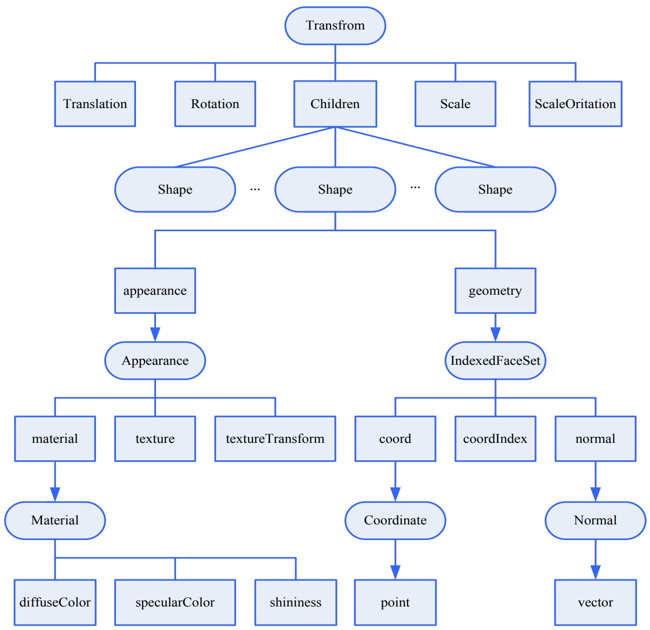


Fig. (1). Node of VRML format.

The 3D models used in VR system are generally described by point, line and face. The study used C++ programming language to analyze the model file. Therefore, in programming, design model class, triangle faces class and vertex class should be handled carefully. To accelerate the loading speed and reduce memory usage, an index number was set up for every node, in order to gain each triangle's vertex index and 3D coordinate corresponding to each vertex by referring to the index. As for a simple CAD model, 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, therefore the 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 the distance from the new vertex after collapsing of the two sides to the plane which is adjacent to the two points of the sides before collapsing, determines 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 achieved by using QEM algorithm is good, but as for regular CAD models, it is easy to generate abnormal long and narrow triangles in the process of simplification. Therefore, the thesis made some modifications based on QEM algorithm, by adding morphologic changed cost of triangle faces in collapse cost to improve simplification strategies of triangles to make shapes and distribution of model triangle faces more reasonable in the process of simplification.

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

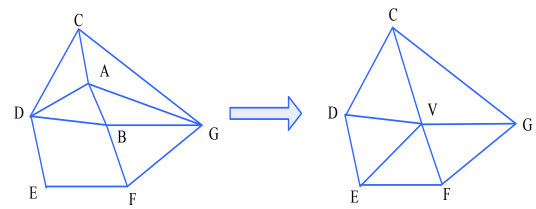


Fig. (2). Principles of edge collapse.

In QEM algorithm, quadric error cost determines whether the side can be collapsed by analyzing the collapse order. The detailed algorithm is as follows:

Assuming $v=\begin{bmatrix} v_x & v_y & v_z & 1 \end{bmatrix}^T$ is a vertex in the mesh, p is a plane in the 3D space, and the coefficient of the plane equation is $p=\begin{bmatrix} a & b & c & d \end{bmatrix}^T$, then the square of distance from point v to plane p is:

$$d^2(v)=v^T K_p v \tag{1}$$

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

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

$$\Delta'(v)=\sum_{p\in planes(v)} d_p^2(v)=v^T(\sum_{p\in planes(v)} K_p)v \tag{3}$$

Make

$$Q'(v)=\sum_{p\in planes(v)} K_p \tag{4}$$

Following this, $Q'(v)$ is 4 * 4 symmetric matrix, and making it quadric error matrix of point v to measure side collapse cost. The collapse cost of side (v_a, v_b) collapsed with a new vertex is $\Delta'(\bar{v})=\bar{v}^T(Q'_a+Q'_b)\bar{v}$ and $Q'_a+Q'_b$ is used to represent the quadric error metrics of the new vertex $\bar{v}$.

The core of side collapse algorithm is to determine or select the position of a vertex after side collapse. The general handling method to select two endpoints of collapsed side or some point in the ligature of two endpoints as the new vertex, for computing and sorting out 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 obtained from CAD data, collapse costs were observed to be exactly the same or similar. There are several triangle patches shown in Fig. (**3**); when side AB was the collapse side, by taking different points in side AB as collapse points, quadratic error costs of all the collapse sides were observed to be similar and the number of patches after simplification was also the same, but the final shapes of triangles after simplification differed greatly. It can be seen that when a side was collapsed to the midpoint of side AB, the shape of the triangle was the best and all the triangles were observed as equilateral triangles, which was beneficial for calculating illumination in graphics engine and improving the quality of rendering.

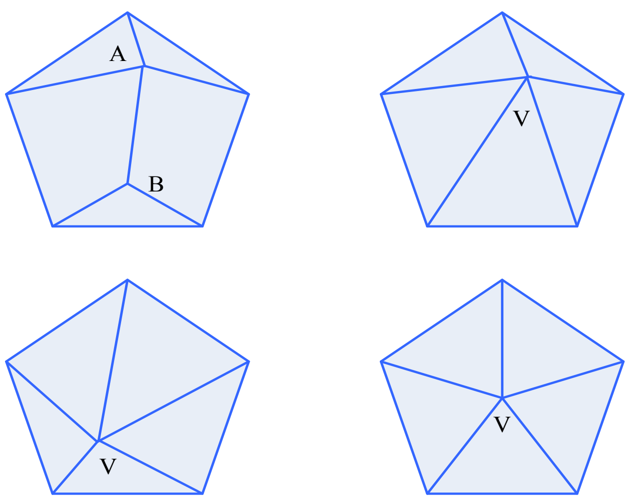


Fig. (3). Simplification effects at the same edge cost.

As for the above features in the simplification process, to optimize the shapes of triangles and evade abnormal long and narrow triangles, the thesis introduced evaluation index of shapes of triangles, evaluating compactness of triangles through the QEM algorithm modified in the thesis [12].

$$c = \frac{4\sqrt{3}A}{l_0^{\ 2} + l_1^{\ 2} + l_2^{\ 2}} \tag{5}$$

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. Following this, vertex v was defined as the average compactness of all triangle faces in which the vertex is included,i.e:

$$C(v) = \sum_{p \in planes(v)} C(p)/n \tag{6}$$

n is the number of triangles, and $C(p)$ is the compactness of the triangle p. When computing and sorting the collapse cost, if the collapse costs of several new vertexes were the same or similar, the compactness of new vertexes was computed and sorted out to make the vertex of maximum compactness the final new collapse vertex, as a result, transferring the collapse cost into the queue. The final algorithm flow of model mesh simplification is as follows in Fig. (**4**).

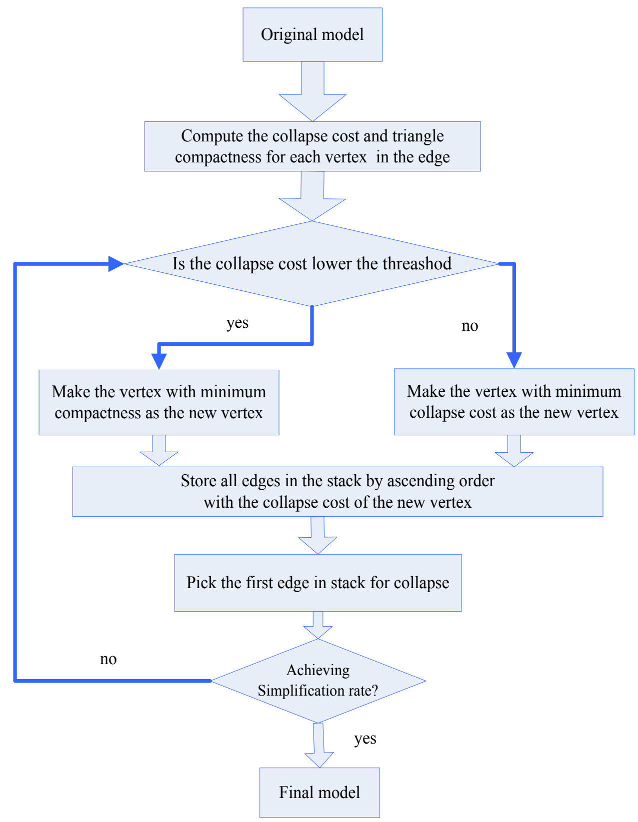


Fig. (4). Algorithm flow of model simplification.

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

4. GENERATION OF VIRTUAL REALITY STANDARD MODEL

In virtual reality system, industrial standard model format of geometrical models includes 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 the model structure and modeling efficiency. By using OpenFlight API interface function, it is possible to directly transform simplified data into an OpenFlight standard model format through program interface. The procedure of establishing standard OpenFlight model file is as follows (See Fig. **5**). This study followed 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].

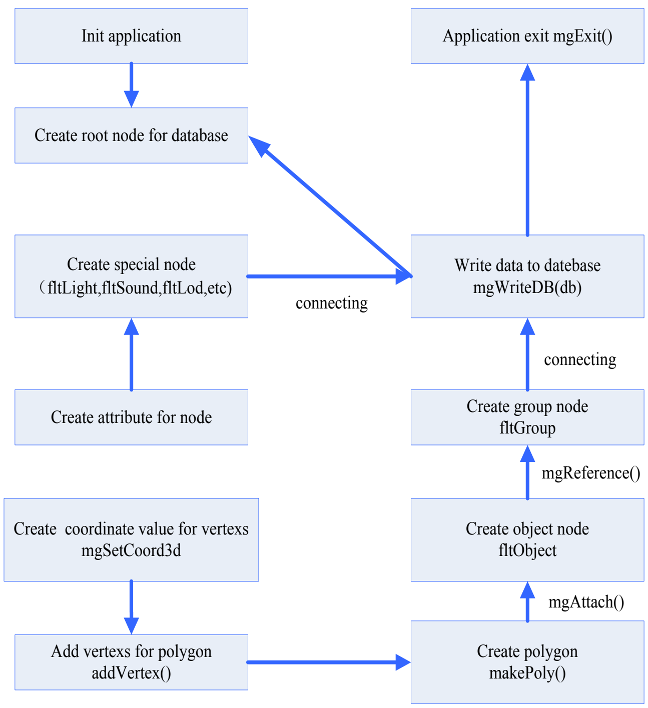


Fig. (5). Procedures of establishing open flight models.

5. EXPERIMENTS

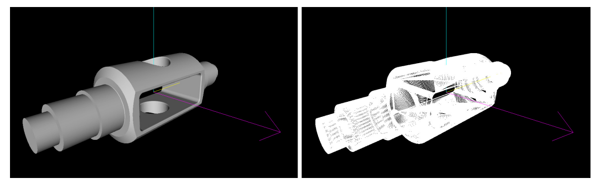
To verify the validity of the method mentioned in the thesis, the study verified the final VR model effect by two experiments:

5.1. Contrast Experiments Among Simplification Algorithms

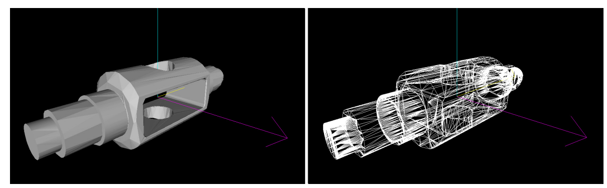
QEM algorithm and modified QEM algorithm were used respectively to simplify a simple part model (shaft), which resulted in a subjective contrast of the final distribution of triangle mesh and visual effects of illumination. The experimental results are as follows (See Fig. **6**). The number of original model patches was 23716, and meshes were intensively distributed, which showed good rendering effect . QEM algorithm and the algorithm proposed in the thesis were used to simplify the model (all the simplification rates were set as 90%). It can be seen from the figure that under the condition of same simplification rate, light uniformity, shapes of meshes and the distribution of models were improved after using the simplification algorithm proposed in the thesis.

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

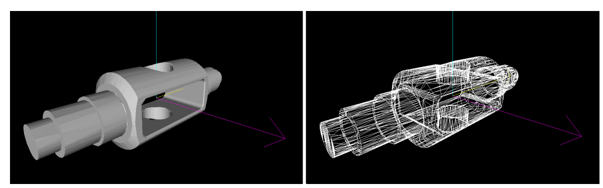
It can be seen from the Table **1** that rendering frame rate and loading efficiency of the simplified generated VR model in virtual engine have been greatly improved, which can better meet the requirements of virtual reality system. In addition, file size in hard disk was also effectively reduced, which is beneficial for enhancing memory and transmission of distributed



(**a**) Original Model



(**b**) Simplified by QEM Algorithm



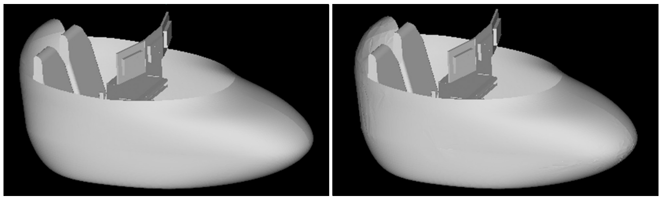
(**c**) Simplified by the modified QEM Algorithm

Fig. (6). Model simplification result of QEM algorithm and modified QEM algorithm.

Table 1. Models at different simplification rate

Simplified ratio(%)	Triangle count	Frame rate(fps)	Load time(s)	File size(m)
0	47512	66	0.53	6.4
50	23189	79	0.47	3.7
60	8852	83	0.31	2.3
70	6189	95	0.26	2.0
80	4339	102	0.19	1.3
90	4672	117	0.11	0.75

rendering systems. At the same time, shapes of models almost showed no change after simplifying 90% of the patches, which also meets visual requirements, as shown in Fig. (**7**).



(**a**) Original model (**b**) Simplification model

Fig. (7). Real-time rendering effects in virtual engine.

CONCLUSION

For modeling requirements in the VR system, this study introduced 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, shape of model triangle mesh becomes moreappropriate. When visual effects experienced no change, both rendering efficiency and loading time were greatly improved, and model files accounted for little hard-disk space, which is beneficial for storage and network transmission. The method can be seen as a supplementary method for modeling with 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

[1] Y. Tang, and H. B. Gu, "CAD Model's Simplification and Conversion for Virtual Reality," *Information and Computing (ICIC)*, 2010 *3rd International Conference on, IEEE*, Wuxi, 2010, pp. 265-268.

[2] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," In: *Proceedings of the 24th annual ACM Symposium on User Interface Software and Technology*, 2011, pp. 559-568.

[3] B. L. Wan, J. H. Liu, and Y. X. Ning, "Research and realization on CAD model transformation interface for virtual assembly," *Journal of Simulation*, vol. 18, pp. 391-394, 2006.

[4] L. X Sun, and R. F. Li, "Visualization research of transform from CAD model to VR mode," *Computer Engineering and Applications*, vol. 44, pp. 117-120, 2008.

[5] J. S. Liu, Y. X. Yao, S.A. Pahlovy, and J. G. Li, "A novel data decomposition and information translation method from CAD system to virtual assembly application," *International Journal of Advanced Manufacturing Technology*, vol. 28, pp. 395-402, 2006.

[6] W. D. Li, Y. L. Cai, and W. F. Lu, "A 3D simplification algorithm for distributed visualization," *Computers in Industry*, vol. 58, pp. 211-226, 2007.

[7] A.Thakur, A. G. Banerje, and S. K. Gupta, "A survey of CAD model simplification techniques for physics-based simulation applications," *Computer-Aided Design*, vol. 41, pp. 65-80, 2009.

[8] Z.M. Qiua, Y.S. Wonga, J.Y.H. Fuha, Y.P. Chenb, Z.D. Zhoub, W.D. Lic, and Y.Q. Luc, "Geometric model simplification for distributed CAD," *Computer-Aided Design*, vol. 36, pp. 809-819, 2004.

[9] K. Walczak, and W. Cellary, 'X-VRML for advanced virtual reality applications," *Computer*, vol. 36, pp. 89-92, 2003.

[10] Q. Yu, H. B. Gu, and Y. Tang, "Algorithm of transform from CAD to VR model based on sorting," *Journal of WUT (Information & Management Engineering*, vol. 33, pp. 548-552, 2011.

[11] M. Garland, and P. S. Heckbert, "Surface simplification using quadric error metrics," *Proceedings of the Computer Graphics*, pp. 209-216, 1997.

[12] A. Guziec, "Locally toleranced surface simplification,"*IEEE Transactions on Visualization and Computer Graphics*, vol. 5 pp. 168-189, 1999.

[13] C.J. Chen, Y. F. Wu, and Y. Yang, "A modeling and representation method for virtual assembly system," *Applied Mechanics and Materials*, vol. 29, pp. 1057-1062, 2010.

Received: June 10, 2015 Revised: July 29, 2015 Accepted: August 15, 2015