STL data visualized by triangular facet algorithm

ZENG Feng

School of Computer, Jia Ying University, Meizhou 514015, China

28251245@qq.com

Keywords: STL, Triangular facet, OpenGL

Abstract. Aiming at the demerits of STL data, a visualization algorithm of STL model is presented. Valid data consisting the vertexes and normal vector of triangular facet were separated out from STL file. The sequential file was used to save the valid data, in order to avoid excessive use of memory. The algorithm was developed through C++ and OpenGL, and the system has the functions such as translations, scales and rotations. The case studies show that the error of STL files can be detected and the optimization of process can be visual supported by the system.

1 Introduction

STL(STereo Lithography) point cloud data approximate the 3D model surface by a series of triangular facets, and each triangle facet is defined by 3 vertexes and 1 normal vector. The STL data is unrelated to any one CAD system. The representation and transmission of 3D model surface are simplified by STL. The file format of STL is widely used for 3D printing, rapid prototyping, reverse engineering and the data exchange in different CAD systems [1-3]. Before the STL model is inputted into the CNC machine, it will be cut into slices to obtain the 2D outline data [1-4]. It could generate some errors such as gaps, cracks and vector reversal when transforming STL data [5]. The 3D model can not be visualized in present slicing process, so, the flaws of STL point cloud data could be an under-estimate.

Based on above-mentioned circumstance, STL data visualized algorithm is developed through C++ and OpenGL. The algorithm completely out of CAD system and the 3D model is quickly visualized to defect detecting, confirming slicing direction and products preview. This work provides support for optimizing design & process.

2 STL point cloud data analysis

There are two styles for STL file, which are binary file and ASCII text. The binary STL file begins with file name and other properties which consume 80 bytes. The second element includes 4 bytes to store the total number of triangular facets. In the final section of STL files, geometrical information of each triangular facet is described by fixed-length bytes. ASCII STL files present information of component and triangular facets line by line. Geometrical information of each triangular facet is composed by 1 normal vector and 3 vertexes, as follows.

```
solid
  facet normal 0.000000e+000 1.000000e+000 0.000000e+000
    outer loop
      vertex 7.382817e+001 2.158745e+002 2.254879e+000
      vertex 5.014560e+001 2.654712e+002 3.254789e+000
      vertex 7.174124e+001 3.548971e+002 4.257899e+000
    endloop
  endfacet
   ………
endsolid
```

All triangular facets' data are between the first line "solid" and the last line "endsolid", while one triangular facet's geometrical information is between each "facet" and the next "endfacet", which is defined by coordinate values. For example, the 3 values behind the word "normal" in the second line define the normal vector of first triangular facet. And the coordinates of 3 vertexes (*x*, *y*, *z*) are between “outer loop” and “endloop”. One STL file contains geometrical information of one 3D model, which triangular mesh is used to approximate any 3D model surface. The more triangular facets there are, the higher approximation’s accuracy and model’s quality will be. The relationship between normal vector and 3 vertexes conforms to right-hand rule, as shown in Fig.1.

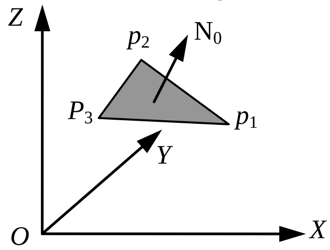
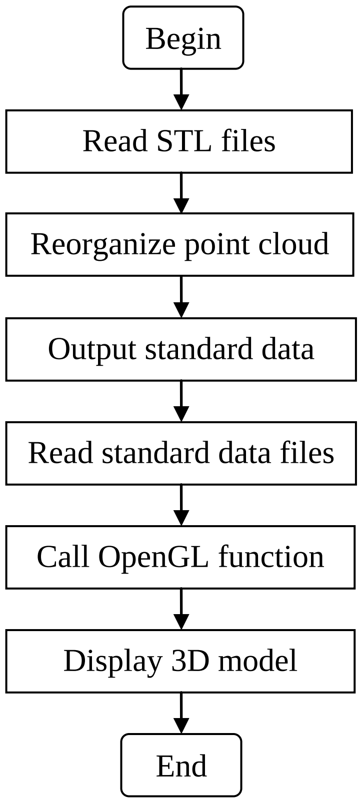


Fig.1 Normal vector and 3 vertexes of 1 triangular facet

3 Visualizing process and key algorithm

STL files can not be used immediately which contain some key words besides the coordinates of normal vector and vertexes. The coordinates are extracted and reorganized by filtering out key words. Total number of triangular facets would be unusually large when the surface quality of 3D model is very high. So, in order to avoid using large array and memory overusing, writing files technology based on external memory must be used to store data. The flow chart of STL model visualizing is shown in Fig.2.



3.1 Data filtering algorithm

In STL files, all coordinates of normal vectors and vertexes begin at 17th character in certain line. The left 16-count characters are the “space” and “facet normal” or “vertex”. Define *k* as line number of STL files, *j* as the line number of sequential file by reorganized, *str1* and *str2* as the string variable, n as the total number of rows. Data filtering algorithm is shown as follows, while initializing *k*=1, *j*=1.

Step1. Read the character string of line *k* and assign it to *str1*; assign left 16-count characters of *str1* to *str2*.

Step2. If *str2* is “***facet normal” (1 * means 1 space), then go to Step4, else, go to Step3.

Step3. If *str2* is “**********vertex” (1 * means 1 space), then go to Step4, else, go to Step5.

Step4. Delete the *str2* in *str1*; write remaining data output to line k of sequential file; $j=j+1$, $k=k+1$; return to Step1.

Step5. IF $k=n$, that means the last line of STL files is reached, then end the algorithm, else, $k=k+1$, and return to Step1.

Line-by-line buffering technology of sequential file is used to store the valid data, without the substantial memory overhead. The data in each low of sequential file is coordinates when the algorithm is ended, as shown in Fig.3. The first row of data while $k=4i+1$ and $i\in[0,1,\ldots(n-1)/4]$ is the normal vector, and the other row of data are vertex's coordinates.

```
0.000000e+000 1.000000e+000 0.000000e+000
7.382817e+001 2.158745e+002 2.254879e+000
5.014560e+001 2.654712e+002 3.254789e+000
7.174124e+001 3.548971e+002 4.257899e+000
0.000000e+000 1.000000e+000 0.000000e+000
7.254879e+001 1.854795e+002 1.254879e+001
9.326458e+001 2.687894e+002 2.658746e+001
5.574894e+001 1.254874e+002 2.365874e+001
```

Fig.3 The sequential file after filtering STL file

3.2 Triangular facet parameter identifying and drawing algorithm

Each row of sequential file define the (x,y,z) coordinate of normal vector or vertex. Known from filtering algorithm, the parameter type is closely related to the line number. Every 4 rows is 1 triangular facet's information, which the first row is normal vector, and the other 3 rows are the 3 vertexes. Define *normal* as array to save the normal vector coordinates, *v1*&*v2*&*v3* as array to save the 3 vertexes' coordinates, j as current row number, m as total number of rows. The algorithm is as follows while initializing $j=1$.

Step1. Assign row j to array *normal*.

Step2. Assign row $j+1$ to array *v1*.

Step3. Assign row $j+2$ to array *v2*.

Step4. Assign row $j+3$ to array *v3*.

Step5. Call OpenGL function to draw the current triangular facet. The graphics have the effect of perspective or solid.

Step6. IF $j=3+m$, that means the last line of sequential files is reached, then end the algorithm, else, go to Step7.

Step7. $j=j+4$, return to Step1.

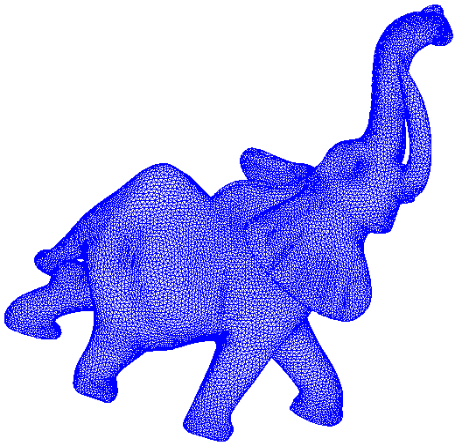
Parameters of 1 triangular facet are read line-by-line, and then the triangular facet is drawn in RC (Rendering Context). Repeat above process until the all triangular are drawn completely.

4 Application instance

The algorithm is realized at C++ and OpenGL. Besides the drawing functions, there are many other powerful functions such as graph editing, photometric properties, and so on. The triangular facets of foot and elephant are shown in Fig.4 and Fig.5, while the glcolor3f is (0.5,0.5,0.5) in illumination display. Table 1 shows the relevant information of program running system.

(a) Point cloud of foot (b) Triangular mesh of foot

Fig.4 Visualizing of foot STL data



(a) Triangular mesh of elephant (b) Triangular facets' illumination display of elephant

Fig.5 Visualizing of elephant STL data

Table 1 Calculated result of STL point cloud data
(at Inter Dual-Core 2.5G CPU, 3.25GB RAM)

STL model	Total number of Points	Total number of triangle	Running time(S)
Foot	48847	17428	2.5
Elephant	185371	47857	5.7

The visualizing system of STL model is unrelated to any one CAD system. STL point cloud can be display quickly. The defects such as normal vector reversal, gaps and cracks can be detected by this system, which can be used in 3D printing, RP and finite element analysis.

5 Summary

STL format is widely used for 3D printing, RP, reverse engineering and data exchange in different CAD systems. In this paper, the contents of triangle information are analyzed, and the valid data are filtered out to store in secondary store. The algorithm was realized at C++ and OpenGL, which the STL data were transformed to 3D model. Before the STL data entering process, the defects can be quickly detected by this work.

Acknowledgement

This work was supported by Natural Science Foundation of Guangdong Province #2014A030307038

References

[1] W. Jingya, F. Ling and H.J. Bin. Adaptive slicing algorithm of STL model based on feature facets. Application Research of Computer, Vol. 28 (2011), p. 2361-2364

[2] Y. Guang, L. W. Jun, W. Wei, T.F. Jie. Research on the rapid slicing algorithm based on STL topology construction. Modern Manufacturing Engineering, (2009), No.10, P. 32-35

[3] Z. Hu, Y.Z. Feng, Z. Wei. Progress in Study of STL File and Its Application. MACHINE TOOL&HYDRAULICS, Vol. 37, No.6, (2009), P. 186-189

[4] W.Z. You, H. L. Lin, Z.G. Xian. 3D sculpture algorithm based on hierarchical slicing theory. Journal of Computer Applications, Vol. 31, (2011), No. 2, P. 379-382

[5] Kang-Soo Lee, Sung-Hwan Kim. Non-uniform deformation of STL model satisfying error criteria. Computer-Aided Design, Vol. 42, (2010), No.3, P.238-247