

2012 AASRI Conference on Computational Intelligence and Bioinformatics

Research and Application on the Method of 3DS File Read, Render and Control

LiuFang^{a*}, LiuXianmei^a

^a:School of Computer and Informaiton Technology,Northeast university of petroleum,Daqing,Heilognjiang 163318,China;

Abstract

OpenGL is a strong there-the graph develop tools,3DS format is a common format for saving there d model Based on OpenGL, the paper researches on the method of 3DS model file read, render and control. First it analyzes the file structure of 3DS, and abstracts the data structure of face, material, object and model by chunk structure used to read in data. The file data is read in with the method of recursion, the read data renders in screen after data disposal. The interactive control of model is accomplished by three kinds of geometric transformation, translate, rotate and scale and OpenGL select mechanism. Finally, those methods are applied into the system of the borehole operation operate simulation and drilling, and achieve better results.

© 2012 Published by Elsevier B.V. Open access under [CC BY-NC-ND license](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Selection and/or peer review under responsibility of American Applied Science Research Institute

Keyword: 3DS file, Chunk structure, OpenGL, Read, Render, Interactive control

1. Introduce

OpenGL is a strong there-the graph develop tools and a software interface of graph hardware, Points, lines, polygons and these basic geometries make up of models in OpenGL^[1], so it' s difficult to construct complex models directly in OpenGL, such as bent or beveling objects. Compare with OpenGL, 3DS Max has strong ability to construct models with the method of lathe, lofting and so on in short time. The models in 3DS Max could be exported as .3DS file, if the complex models constructed in 3DS Max could be import into OpenGL,

* Corresponding author. Tel.: +861-13936923027; fax: +861-0459-6503553;

E-mail address: lfliufang1983@126.com.

the efficiency of developing 3d graphics using OpenGL could be improved. So the article researches the methods of 3ds file read, render and control in OpenGL, and applies the method into the system of the borehole operation operate simulation and drilling, and achieve better results^[2].

2. 3DS file format

3DS file is saved as chunk structure. Beginning of a chunk is 2 bytes ID, 4 bytes chunk length information, and then is the main data of the chunk. 3DS file is opened in hex format, the data is saved according to the lower is in the front, and the higher is in the back. For example, 2 bytes FF AF is integer in hex, AF is higher byte and FF is lower byte; 93 8E 06 00 is long integer, 8E 93 is lower byte and 00 06 is higher byte.

There is a basic chunk in every 3DS file, The ID of the chunk is 4D4D, the subchunks are the main edit chunk with ID 3D3D, the key chunk with ID B000. The structure of the chunk and nested relationship of every chunk is complex. Now let's briefly describe several main chunks.

2.1. The main edit chunk (0X3D3D)

The states of current edited scene and the configuration information of current window are saved in the main edit chunk (0X3D3D), which is used to describe scene information, these information is not only important geometry information used to display three dimensional graph but also the key point research part of loading 3DS file based on OpenGL.

2.2. The material chunk(0XAFFF)

Material chunk(0XAFFF) defines the material attribute of objects, which includes the name,color and texture of material. If the material includes texture,the name, the alpha information, the position and the UV map of textures are also saved in the subchunk of material chunk.

2.3. The Object Chunk (0X4000)

Object chunk includes object mesh chunk (0X4100) and object material chunk (0X4130). The vertex of model, the UVW of vertex, face information and edge information are saved in object chunk. The endowed material information in current object is saved in object material information. The difference between material chunk and object material chunk is that 3DS file's material library information is saved in material chunk, compare with it, material information in current object is saved in object material chunk.

2.4. The Other Chunks

If the information of chunk is unknown or invalid for displaying model, we should skip current chunk and directly read next chunk.

3. Read 3DS File

According to the characteristics of the 3DS file structure, we will illustrate the basic data structure and loading flow in reading file^[3].

3.1. The basic data structure

3DS file is saved as nested chunk structure, so it should be divided into chunk. Defining the data structure of basic chunk as follows:

```

struct tChunk
{
    unsigned short int ID; // ID the ID of chunk
    unsigned int length; // the length of chunk
    unsigned int bytesRead; // the number of bytes readed;
};

```

After abstracting the face chunk, material chunk and object chunk structure, the structure definition of basic model in scene is arrived, as flows:

```

struct tModel
{
    int numOfObj; //The number of objects
    int numOfMat; //the number of material
    vector<tMat> pMat; //Material linked list
    vector<tObj> pObj; //object linked list
    CVector3 Pos; //position
    CVector3 Rot; //rotation
    CVector3 Sca; //Scale };

```

3.2. Reading and loading file flow

The chunk structure of 3DS is fixed and nested, we should read file in recursive rule, and the condition of return back is whether the bytes read is equal to the chunk length. Jump from father chunk to subchunk could be realized by switch statement, which subchunk branch could be enter is through the subchunk ID. If we want to skip an unused chunk, the pointer point to file could be moved forward chunk length-6 bytes, and then skip this chunk by break statement, and process next chunk. If we want to read a chunk, for example material chunk, when the ID is 0xAFFF, first self add variable numOfMat used to record material number, then distribute an empty structure to current material chunk, and then Recurs call function ProcessNextMaterialChunk() to read relative information^[4]. Whole program will end after main chunk is all read. The main flow of read program is as figure 1.

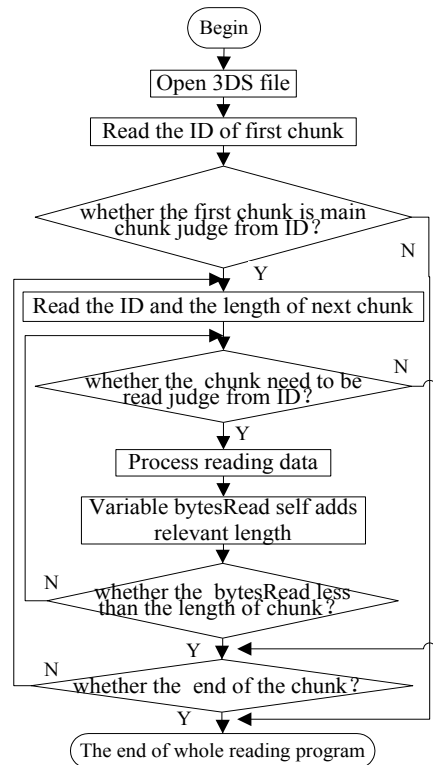


Fig. 1 3DS file process flow

4. Draw and Control Model

4.1. Data Processing

The information directly getted from the 3DS file can not reach requirement of rendering model in OpenGL, so the data is must to be processed, including the Coordinate conversion and the vertex normal vector calculation.

1. The Coordinate conversion

The coordinate system of OpenGL is different to 3DS MAX. In this article, the reading program is designed like that, swapping the coordinate of y and z, and then saving the swapped data when reading vertex list data. Compare with processing after reading data, this method could save time and space in processing.

2. The vertex normal vector calculation

The render effect of the vertex- triangle 3D model depends on vertex normal vector's setting, so we must allocate a normal vector for every vertex in model, which could be used to compute lighting effects and judge the orientation of the triangle face relative to the observer. From solid geometry knowledge, we could know that all vertex normal vector in the same face is not necessarily same, the normal vector of a point is equal to the average of all face to this point for vertex. Reading program first initialized all model vertex normal vector as(0.0,0.0,0.0),then computed normal vector of every face separately ,and then add it to vertex normal vector of this face, finally normalized all vertex normal vector, the final compute result of the vertex normal vector will arrived.

4.2. Render

Model's geometry, material information are saved in relevant data structure, rendering is according to these information, draw scene to the screen with graphic command provided by OpenGL^[5].

Primitive's render way is setted up as triangle strip, when rending, program judges whether there is texture mapping, If there is open texture mapping. Model renders according to the following 4 steps:

- (1) configure head file。 #include<gl/gl.h>,#include<gl/glu.h>,#include<glaux.h>.
- (2)call function bSetupPixelFormat(HDC hdc) to set pix format
- (3)call function SizeOpenGLScreen(int width,int height) for the projection transformation
- (4)Call function RenderScene() for drawing and rendering model.

4.3. Interactive control

1. Geometry of matrix

Translation, rotation and scale is the basic geometry of matrix, in the process of interaction, every operation to model could be converted into translation, rotation and scale the three matrix transformation combination. Here hypothesis $[x' \ y' \ z' \ 1]$ as the current matrix, $[x \ y \ z \ 1]$ as the goal matrix, it has the following matrix transformation.

2. The choice mechanism of OpenGL

If we want to operate the 3d object interactively in OpenGL, the first to be sure is the object operated. The method is the choice mechanism of OpenGL. The choice is an operation which the operator chick in a certain position of the window's internal with the mouse to determine which object he chicks^[6].

The step of choice mechanism in OpenGL is as follows:

- (1)Enter choice pattern. Call function glRenderMode(); it's archetype is glRenderMode(GLenum mode); the mode is setted up as GL_SELECT.

(2) Create name stack. It is the object choiced by operator saved in name stack. First call function `glInitNames()` to initialize name stack, that is clear name stack, second call function `glPushName()` to press object name into the stack when sending command to render all kinds of models, call `glPopName()` function to push the object name from stack.

(3)Generate hitted record. The hitted record is a chosen to sign which is in view port or pixel intersect with view port.

(4)Define the view port used to be choosed. We usually call `glPushMatrix()` and `glPopMatrix()` to save and recover the current view port state.

(5) Alternately release pixel render command and name stack control command to give a suitable name to every pixel.

(6)exit choice pattern, deal with the choiced data, that is hitted record.

3. Main code

According to the geometry of matrix and OpenGL selection mechanism, the core code in OpenGL to realize interactive control is as follow:

(1) Choice funcion

```
hits = glRenderMode (GL_RENDER); //collect mouse click message
processHits (hits, selectBuf); //deal with the interactive after click
```

(2)Translate function Translation().Generate the translate of model in three axis.

```
g_Model[hits].Pos=TraDelta;
```

(3) Rotate function Rotation().

```
g_Model[hits].Rot=RotDelta;
```

(4)Scale function Scale(). The zoom factor is Sca_gene

5. The application in simulation system

Based on those methods, we design and realize borehole operation simulation and drill system. The simulation system realizes the operation of picking the donkey head, up tube and so on, 20 kinds of operation in borehole operation process. When applying, first the borehole operation model constructed in 3DS Max could be imported into OpenGL, then translate, rotate and scale model by geometric transform, adjust the size and the position of model to meet the practical requirements. Second layout scene, the method of texture mapping is applied to Drill background, namely choose 2n size bmp picture format as the texture, and then select the suitable mapping coordinates for the texture .Finally lighting the scene, open the function of scene lighting.

The critical equipment in borehole process is pumping Wells, and its surrounding related equipment, and a beam pumping unit is composed by the donkey head, balance piece, swim beam and control box components. Based on the OpenGL interactive control mechanism, we could interactively control the critical equipment in process. In the operation simulation process, first we should click the equipment model need to be controlled in mouse left key, in the choice mode, equipment models hit records have chosen to sign, and the corresponding name is pushed into the name of the stack, second exit the choice model and return the choiced data. Enter into the rendering model after the equipment model motion events are triggered, call render function Translation(hits), Rotation(hits) or Scale(hits) to translate, rotate or zoom the corresponding equipment to realize the aim of control model.

Figure 2 is picking the donkey head operation simulation of movement when rendering, the process of donkey head move down call the function of translation. Figure 3 is operating the hanmer to hit the pip when rendering, the process of wield and hit call the function of rotate.



Fig. 2 picking the donkey head operation simulation of movement

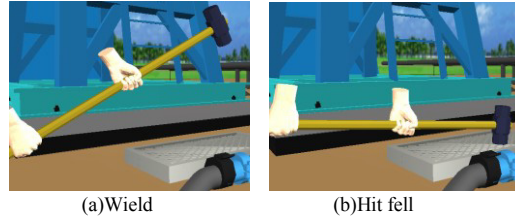


Fig.3 Operate the hammer to hit pipeline

6. Conclusion

3DS is the universal data format to save 3d model, so it's important to research the reading and rendering. The article researches on the method of reading, rendering and interactive controlling 3DS file, first divided the 3DS file into chunk, read the chunk based on recursion call, after coordinate change and vertex normal vector processing, rendered static 3d model, realized the interactive control based on geometry of matrix and OpenGL selection mechanism. Finally, those methods are applied into the system of the borehole operation operate simulation and drilling, and achieve better results.

Acknowledgements

One reviewer made several helpful suggestions which I hope I have implemented satisfactorily to clarify this paper.

References

- [1] Richard S. Wright, Jr. Benjamin Lipchak, OpenGL SuperBible (4th Edition), Addison Wesley, 2007, 30-40
- [2] Donald Hearn, M. Pauline Baker. Computer Graphics. Beijing: Publishing house of electronics industry, 1998: 128~134.
- [3] Flanery Jr, Raymond E, Immersive and 3D viewers for CUMULVS, Future Generation Computer Systems, Jul. 2003, 19(5), 701-719
- [4] Haeberli, Painting and Texturing on 3D Shape, Computer Graphics, 2000, 24(4), 64-73
- [5] Zhu, Yimin, Viewer controllable visualization for construction document processing, Automation in Construction Volume, 2003, 12(3)
- [6] Kelsey J. Sinclair, Improving computer-assisted instruction in teaching, Computer & Education, 2004, Vol(42), 169-18