

# 3D object surface color extraction based on 3D printing equipment

Yi Wang, Yang Xiao, Fanyao Jin, Jie Wei

College of Mechanical and Electric Engineering, Southwest Petroleum University,  
Chengdu Sichuan 610500, China

---

## Abstract

with the continuous development of 3D printing technology, the demand of industry has gradually changed from single-color 3D printing to 3D color printing. This paper proposes a method to read and display the 3ds model file based on OpenGL, and extract the surface color information of the model taken out according to the slice form required by 3D printing, and finally submit it to the 3D printing equipment to realize 3D color printing.

## Keywords

3D printing; 3ds model file; Color printing method.

---

## 1. Introduction

3D color printing technology can print the product parts and get the color appearance of the product at the same time. Currently, 3D color printing is mainly divided into two types. The first type is the multi-material 3D printing technology, which realizes the color of products by printing mixed materials of different colors. The second type is color ink-jet printing, which is achieved by spraying color adhesive on the powder bed. In recent years, through the research on 3D color printing, there have been 3D color printing equipment for specific environmental requirements. This paper proposes a 3D color printing method based on OpenGL to read and display the 3ds model file. Then extract the surface color of the model taken out according to the slice form required by the 3D printing equipment. Finally submit it to the 3D printing equipment, and print out the color and pattern at the same time according to the 3D printing model.

## 2. OpenGL reads and draws 3ds model files

OpenGL is a software interface for graphics hardware. It is not a programming language but more like a C runtime function library<sup>[1-2]</sup>. OpenGL constructs models through the basic geometric primitives of points, lines and polygons, so it is very difficult to directly construct complex objects, especially those with curved radii or chamfers<sup>[3]</sup>. However, 3ds model file, as a 3dmax file format with powerful modeling functions, is widely used, and 3dmax modeling is easier and faster than OpenGL. In addition, this paper proposes a method for 3D color printing. 3dmax modeling software is more mature in color rendering than other modeling software. Therefore, this paper proposes OpenGL to read and display 3ds model files.

### 2.1 3ds file structure description

3ds file is a kind of universal binary file to save the model property information, which contains all the property information of the model such as point, line, surface, texture, material and color<sup>[4]</sup>. A 3ds file is made up of blocks, so blocks are the basic building blocks of a 3ds file. The block contains the ID of the block and the address of the next block. Blocks have a hierarchy, child blocks are nested within the parent block, and different categories have different data block hierarchies. Each 3ds file contains a master block, because each block of a 3ds file contains the ID of that block and the address of the next block, so when we read in a 3ds file we can skip it and read in the next block if we

encounter an unwanted block. For OpenGL to read 3ds files, you only need to read the following major blocks.

The main edit block (0X3D3D) describes the scene data and stores the status of the current edit scene and the configuration data of the current window. These data are important geometric information for displaying 3d graphics, and also the key research part of OpenGL loading 3DS files.

The material block (0XAFFF) defines the material properties to use on an object, including the name of the material, the color, the texture, and so on. If the material being read contains a map, the file name, transparency information, location, and wrapping method for the map are also stored in the corresponding subblock of the material block.

Under object block (0X4000) are object mesh block (0X4100) and object material block (0X4130). The object grid block stores the vertices, texture coordinates, surface information and edge information of the model. The object material block stores the material assigned to the current object. The difference between this block and the material block is that the material block stores the information in the material library in the 3DS file, while the object material block stores the material information on the current object.

## 2.2 3ds file reading

According to the characteristics of 3DS file structure, the basic data structure and reading process needed in the file reading process are described [5]. Because 3DS files are stored in nested block structures, you can partition 3DS files into blocks. Therefore, the basic block structure is defined as follows:

```
struct tChunk{
unsigned short int ID;// Piece of ID
unsigned int length;// The length of the block
unsigned int bytesRead;// The number of bytes read into the block
};
```

After abstracting the structure of Face block, material block and object block (obj), the structure of the basic model of the scene is obtained:

```
struct t3DModel {
int numOfObjects; // Number of objects in the model
int numOfMaterials;// Number of materials in the model
vector<tMaterialInfo> pMaterials;// Material list information
vector<t3DObject> pObject;// Object linked list information in the model
};
```

The block structure of file reading process 3ds is fixed and nested, so recursive method is selected to read, and the condition of returning the upper level is whether the number of bytes of the block that has been read is equal to the length of the block. Jumping from a parent block to a child block can be done with a switch statement, and the ID of the child block determines which branch to enter. When you encounter a block that you do not want to read in, you can judge by the ID. If you do not need this block information, you can move the file pointer forward (block length -- 6) bytes [6], and then break the block and process the next block. When a block needs to be read in, such as MATERIAL block, by judging when the ID is 0XAFFF (edit-material), first add the variable numOfMat recording the number of materials, then allocate a blank structure for the current MATERIAL block, and then call ProcessNextMaterialChunk() function recursively to read the relevant information. When the main block is finished reading, the whole reading program ends.

After 3ds file is read in, OpenGL will render and draw. The information obtained directly from 3ds file cannot meet the format requirements of OpenGL drawing model. Therefore, corresponding data processing should be done before drawing, including coordinate system transformation and vertex normal vector calculation. OpenGL has a different coordinate system than 3DMAX. Therefore, in

this paper, the read-in program is designed to directly store the coordinates of Y and Z when reading the data in the point list, which saves the time and space for processing after reading all the data [7]. Vertex normal vector calculation. The rendering effect of the vertex triangle 3d model depends on the setting of the vertex normal vector. Therefore, a normal vector must be specified for each vertex in the model. From our knowledge of solid geometry, we know that the normal vectors of all the vertices on the same plane are not necessarily the same, and the normal vector of a point is equal to the average of the normal vectors of all the faces with that point as the vertex. This paper first initializes all vertex normal vectors of the model, then calculates the normal vectors of each surface, and adds them to the vertex normal vectors of the surface. Finally, normalizes all vertex normal vectors, and obtains the results of vertex normal vector calculation. The geometric and material information of the rendering model is stored in the corresponding data structure. The rendering is based on this information and drawn with the drawing command provided by OpenGL [8~9]. Set the rendering mode of the primitive as triangle strip, and judge whether there is texture mapping in the rendering process. If there is, open texture mapping. Model rendering is performed in the following four steps. First, configure the header file. The second call to the `bSetupPixelFormat(HDC hdc)` function sets the pixel format. Third, `SizeOpenGLScreen(int width,int height)` function is called to perform the projection transformation and viewport transformation. The fourth call `RenderScene()` function to render the model [10~11].

### 3. 3ds file read result

Main code:

```
glBegin
g_ViewMode // Start drawing with g_ViewMode
glNormal3f // Give me the normal vector
glTexCoord2f
glColor3ub
glVertex3f
glEnd // End to draw
```

The final drawing result of OpenGL is as follows:



After reading and drawing by OpenGL, this paper proposes to extract the surface color of 3D objects on this basis, and finally extract the color information to be stored in the format of 3D printer printing model, and realize 3D color printing. Of course, this paper has not completed the final step and is still

in the process of research and exploration. But this idea proposed in this paper is not available before, and it is believed that the final implementation is not far away.

## References

- [1] Sugavaneswaran M, Arumaikkannu G. Modelling for randomly oriented multi material additive manufacturing component and its fabrication. *Mater Design*, 2014, 54: 779–78
- [2] Shreiner, D. *OpenGL Programming guide* [M]. Beijing: Machinery industry press, 2006.
- [3] Zhengbo Zhang, Ye Mu. OpenGL Free rotation of models in 3D files [J]. *Computer engineering and applications*, 2005, 41(13):98-100.
- [4] Lin Du, Hui Jiang. VC++ using DirectX to achieve 3DS file reading and control [J]. *Computer age*, 2010(1):41-43
- [5] HERTZMANN A, JACOBS C E, OLIVER N, et al. Image analogies[C]. *Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH*, 2001:327-340
- [6] Irony R, Cohen-Or D, Lischinski D. Colorization by example[C]. *Eurographics Symposium on Rendering*, 2005:201-210.
- [7] Yusheng Shi, Lichao Zhang. Development of 3D printing technology and its software implementation [J]. *Science in China: information science*, 2015, 45(2):197-203
- [8] Xuejun Zhang, Siyi Tang. Research status and key technologies of 3D printing technology [J]. *Materials engineering*, 2016, 44(2):122-128
- [9] Xia Han, Enyuan Yang. *Rapid prototyping technology and application*. Beijing: China machine press, 2012.
- [10] Jimin Chen. *Basic course of 3D printing technology*. [M]. Beijing: national defense industry press, 2016.
- [11] Ronald Goldman. *Introduction to computer graphics and geometric modeling* [M]. Beijing: Tsinghua university press, 2011.