Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2013, 5(9):420-427

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Reading and voxelization of 3D models

Xu Haili, Wang Heng, Zhu Longbiao and Hua Guoran

School of Mechanical Engineering, Nantong University, Nantong, Jiangsu, China

---

ABSTRACT

3D models have got widespread attention in graphics currently. Most 3D models are used for visual calculation, but they only contain geometric and appearance properties and the description of high-level semantic features suitable for automatically matching are very little. So, how to reasonably describe 3D models (feature extraction) is a difficult problem of the prime importance. Voxelization just can solve this problem. The voxel model is helpful for regularization of 3D models and is convenient to extract models' characteristic signals. This paper proposed a voxelization method based on edclidean distance measurement to voxelize models' vertices, edges and faces. Experiment results indicated that the method is efficient and effective for generating correct entity voxel models.

Keywords: 3D models, voxelization, polygonal meshes, OpenGL

INTRODUCTION

Following sound, images and video signal, 3D models have got widespread attention in graphics since 90's of 20th century. 3D models are widely used in image processing, animation simulation, entertainment and gaming and geometric modeling etc, which tend to replace 2D drawings in engineering design.

Most 3D models are used for visual calculation. They only contain geometric and appearance properties, and the description of high-level semantic features suitable for automatically matching are very little. So, how to reasonably describe 3D models (feature extraction) is a difficult problem of the prime importance[1]. Voxelization just can solve this problem. Using identical little cubes called 'voxel' to represent 3D models' surface is called Voxelization. If such model data is transformed into rules signal, many classic analysis tools such as Fourier transform, unequal moments can be directly extended to 3D cases, which will benefit the extraction of effective feature descriptor.

The term 'voxelization' was first put forward by Arie Kaufman[2]. The simplest voxelization is binary voxelization which means the voxel values either 0 or 1. Binary voxelization is used most commonly currently. Wu Xiaojun[3][4] proposed a voxelization algorithm to produce the entire 3D model relying on octree encoding features. This algorithm employs polygon mesh contour voxels and the flag characteristics of the model's inner and outer voxel sequences to voxelize the 3D mesh model inside, which corrected the mistake of Flooding algorithm in dealing with the closed inner cavity. However, this algorithm is designed for the body voxelization rather than face voxelization. Rueda [5] proposed a simple and robust algorithm without meshdivision, but this method can only generate regular grids. Huang [6] also presented a rapid voxelization algorithm which meets the demands of fidelity, minimality and reciprocity in voxelization by selecting 6 -neighborhood closed or 26 - neighborhood closed. And some researchers [7][9]proposed other methods. Relying on Huang's algorithm, this paper presents a voxelization method based on edclidean distance measurement to voxelize models' vertices, edges and faces. The method can generate a correct entity voxel model, and is convenient for the extraction of model structure.

The structure and reading method of 3DS models

3D Studio MAX (3dsmax) is a powerful graphic software developed by Autodesk, the graphic file format it generates is 3DS. 3DS format is a very common data format, and the information of a 3D graphic file saved in 3DS format is very rich [11]. Other formats of 3D models can be converted into 3DS model easily .

3DS file format: A 3DS file is composed of many blocks, and each block describes the information category firstly, that is，how the block is composed of .The first two items of a block are the ID of the block and the length of the block. The ID of the block is a 2-byte integer, as an identification of the block. The length of the block is a 4-byte long integer that represents the offset bytes of the next block relative to the block's initial position. Accordingly, even if you do not understand the meaning of a block, you can easily skip it.

The main block of a 3DS file that appears firstly is a key block which contains the entire document. There is an another block called 3D editing program block (EDIT3DS) whose ID is 3D3D. The block mainly defines the physical data of the object and contains the triangle information of the model surface for voxelization, such as vertices and faces. EDIT3DS block contains an object descriptor block EDIT - OBJECT, and its ID is 0x4100. Other sub-blocks of EDIT3DS are concerning the features of light, material rendering and windows. In order to obtain the geometric data of the 3D model , we are concerned about the triangle list sub-block OBJ-TRIMESH under EDIT-OBJECT. OBJ-TRIMESH mainly contains seven sub-blocks.

Reading of a 3DS model: We start to design the reading programs of a 3DS model and define a reading class of a 3DS file after we have understood the structure and content of the 3DS file format as well as the data structure of related vertices and faces. C3DSReader class has many reading functions, such as block reading, vertex data reading, face data reading, and material reading and so on.

• Reading of Blocks. Member functions of C3DSReader class, such as Reader， Read3DSFile and Is3DSFile, are used to read the file and to judge whether the file is 3DS format or not . Member functions such as ReadPointArray, ReadFaceArray, ReadColor and ReadMeshMatGroup , are used to read the vertices, faces, color and material of the model. The relationship between the function callings is shown in Fig.1.The entry of the Reader function is filename. The algorithm is described as follows:

Step 1, Use fp(fp = fopen (filename,'rb')) function to open a 3ds file with Read-Only method;
Step 2, Use Is3DSFile(fp) function to verify the file type and to judge whether the file is 3DS file or not;
Step 3, If the ID is 0x4d4d, then the file format is 3DS, then call sub-blocks such as ReadMDATA, ReadNamedObject, ReadTriObject, ReadPointArray to read the corresponding 3DS file, otherwise return false;
Step 4, Use fclose(fclose(fp)) function to close the file.

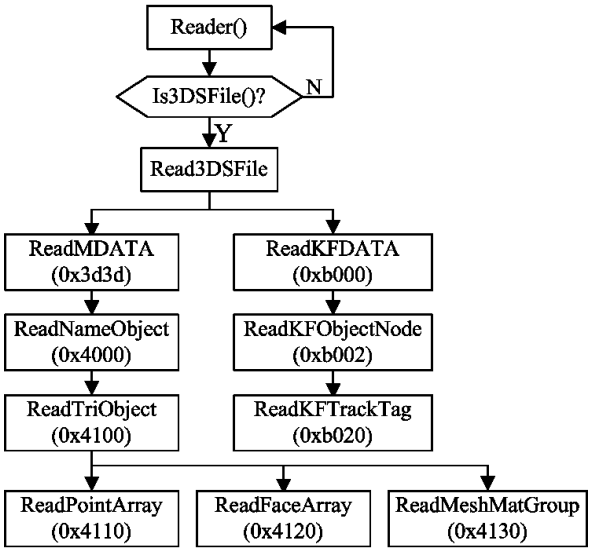


Fig.1 The relationship between the function callings

• Reading of vertex data. ReadPointArray function is used to read the vertex list of the 3DS file. The algorithm is described as follows:

Step 1, Define three dynamic pointers to point to three dynamic arrays respectively. The arrays are used to store every vertex of the model. In addition, define a variable named count to save the total number of the model's vertices.
Step 2, Read the data of all vertices into designated memories;
Step 3, Delete the pointer variables and free their memories.

• Reading of face data. ReadFaceArray function is used to read the face list of the 3DS file. The algorithm is similar to above.

VOXELIZATION OF 3DS MODELS

The basic concepts of voxel and voxelization: Voxel is the basic data unit to describe a volume model in volumetric graphics. Voxelization of 3D Models has significant value. It is more simple, stable and topology information is not required when a geometric object is described with Voxels rather than with triangular patches. After voxelization, model data are transmitted to regular signal, which makes it possible for the use of many signal analysis tools and the efficient extraction of feature descriptor .

Voxelization serves as the process that geometric models described by triangular patches or other boundary convert to a set of discrete voxels with the guaranty of precision. It is the extension of 2D rasterization in 3D space, that is to say, $L \times L \times L$ voxel grids are used to approximate the surface of the 3DS models (m triangular patches), which is the process of an A / D conversion essentially.

Similar to the 4-neighborhood relationship and 8-neighborhood relationship of pixels in 2D graphic filling, voxel neighborhood relationship exists in the voxelization of 3D models. The voxel neighborhood relationship can be divided into three kinds：6-neighborhood, 2 voxels share a face; 18-neighborhood, 2 voxels share a face or an edge; 26-Neighborhood, 2 voxels share a face, an edge or a vertex. In the three kinds of voxel neighborhood relationship, the constraint of 6-neighborhood is the strongest, that of 18-neighborhood comes next, the weakest is that of 26-neighborhood. Fig.2 shows one voxel and all of its N-neighborhood voxels

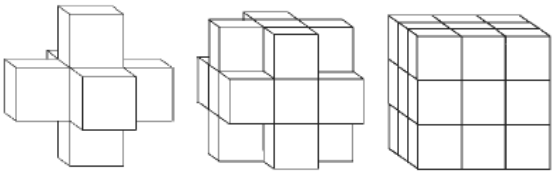


Fig.2 N-neighborhood diagram of the voxel N ∈ {6, 18, 26}

Voxelization convert a continuous description of geometric objects to a set of discrete data. Voxels generated can be binary or multi-valued. In the binary voxelization , one voxel can only be 0 or 1. When a voxel is fully occupied, its value is 1. When a voxel is not occupied by any object, its value is 0. A multivalued voxel can be any value between 0 and 1, which is helpful for anti-aliasing. Multivalued voxelization can be realized by smooth filtering. A precise voxelization method for 3D surface discussed in this paper belongs to binary voxelizztion.

The voxelization algorithm for triangular patches: In this paper, the data source is a 3D mesh model. After voxelization, a 3D volume model can be obtained. Because the feature extraction of 3D model retrieval only needs surface date, voxelization is actually the process to voxelize all the triangle patches which compose the model surface. The intersection of a voxel and a triangle plane is shown in Fig.3. Where, L is the voxel's width, C is the voxel's center, K is the voxel's diagonal, N is the normal to the plane through the voxel's center C, α is the angle between N and K, β is the minimum angle between N and the normal to the faces of the voxel.

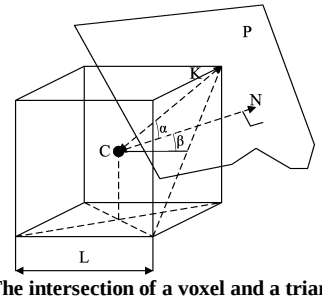


Fig.3 The intersection of a voxel and a triangle face

The voxelization algorithm used in this paper can select 6 - neighborhood closed or 26 - neighborhood closed. There are two theorems for these two representations.

Theorem 1 The set of voxels: $\tilde{S} = \{(x, y, z) \mid -t_{26} < A_p + B_p + C_p + D_p < t_{26}\}$ is a 26-separating and 26-minimal representation of the plane S defined by A_p, B_p, C_p and D_p.

Theorem 2 The set of voxels: $\tilde{\tilde{S}} = \{(x, y, z) \mid -t_6 < A_p + B_p + C_p + D_p < t_6\}$ is a 6-separating and 6-minimal representation of the plane S defined by A_p, B_p, C_p and D_p

We consider three steps of voxelization:
Step 1 Voxelizing Vertices. For each vertex, we define a bounding sphere of radius R_c as shown in Fig.4. All the voxels whose voxel centers fall inside any one of the bounding spheres equal to 1.

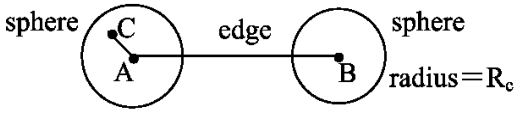


Fig.4 Voxelizing Vertices

In Fig.4, point A (x_1, y_1, z_1)、 point B (x_2, y_2, z_2) is two vertices of a triangular patches ,point C (x_0, y_0, z_0) is any voxel's center, then the distance between voxel's center C to the vertices A is

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2} \tag{1}$$

Step 2 Voxelizing edges. For each edge, we define a regular bounding cylinder of radius R_c and length L, where L is the length of the edge, as shown in Fig.5. All the voxels whose voxel centers fall inside any one of the bounding cylinders equal to 1.

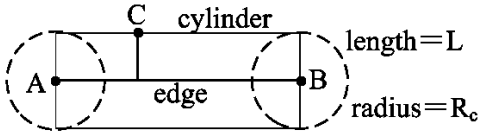


Fig.5 Voxelizing edges

In Fig.5, point A (x_1, y_1, z_1)、 point B (x_2, y_2, z_2) is two vertices of a triangular patches， point C (x_0, y_0, z_0) is any voxel's center, then the distance from voxel's center C to the edge AB is:

$$d = \frac{\sqrt{\begin{vmatrix} x_0 - x_1 & y_0 - y_1 \\ x_2 - x_1 & y_2 - y_1 \end{vmatrix}^2 + \begin{vmatrix} y_0 - y_1 & z_0 - z_1 \\ y_2 - y_1 & z_2 - z_1 \end{vmatrix}^2 + \begin{vmatrix} z_0 - z_1 & x_0 - x_1 \\ z_2 - z_1 & x_2 - x_1 \end{vmatrix}^2}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}} \tag{2}$$

Step 3 Voxelizing surface. For each triangular patch S, we construct two parallel planes such that S lies between these planes and is parallel to both of them, and the distance from S to G or H is *t*, as shown in Fig. 6. All the voxels with their voxel centers falling inside this triangular prism equal to 1.

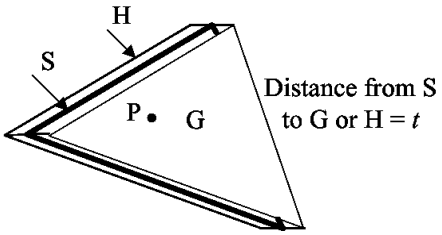


Fig.6 Voxelizing surface

In Fig.6, surface S is a triangular patch, ($S = Ax + By + Cz + D = 0$), point P (x_0, y_0, z_0) is any voxel's center, then the distance from voxel's center C to the triangular patch S is:

$$d = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}} \tag{3}$$

For 26-neighborhood representation, let's use formula (4) and (5) to calculate t_{26} and R_c.

$$t = K\cos\alpha = \sqrt{3}(L/2)\cos\alpha \tag{4}$$

$$R_c = (\sqrt{3}/2)L \tag{5}$$

For 6-neighborhood representation, let's use formula (6) and (7) calculate t_{26} and R_c.

$$t = (L/2)\cos\beta \tag{6}$$

$$R_c = L/2 \tag{7}$$

The minimality and reciprocity of this method has been proved in [6]by Huang .

The realization of the voxelization algorithm: A voxel class Cvoxel used to voxelize the input model is defined in this paper. The voxelization process can be described as follows. First, take each triangle which compose the surface of 3DS model as the basic unit and obtain the data of triangle vertices and surfaces. Then, for each triangle, traverse all voxels to test the intersecting relationship between each voxel and triangle and determine whether the triangle plane possesses the voxel or not. Final, determine the set of voxels that compose the triangle. The algorithm flow chart is as shown in Fig.7.

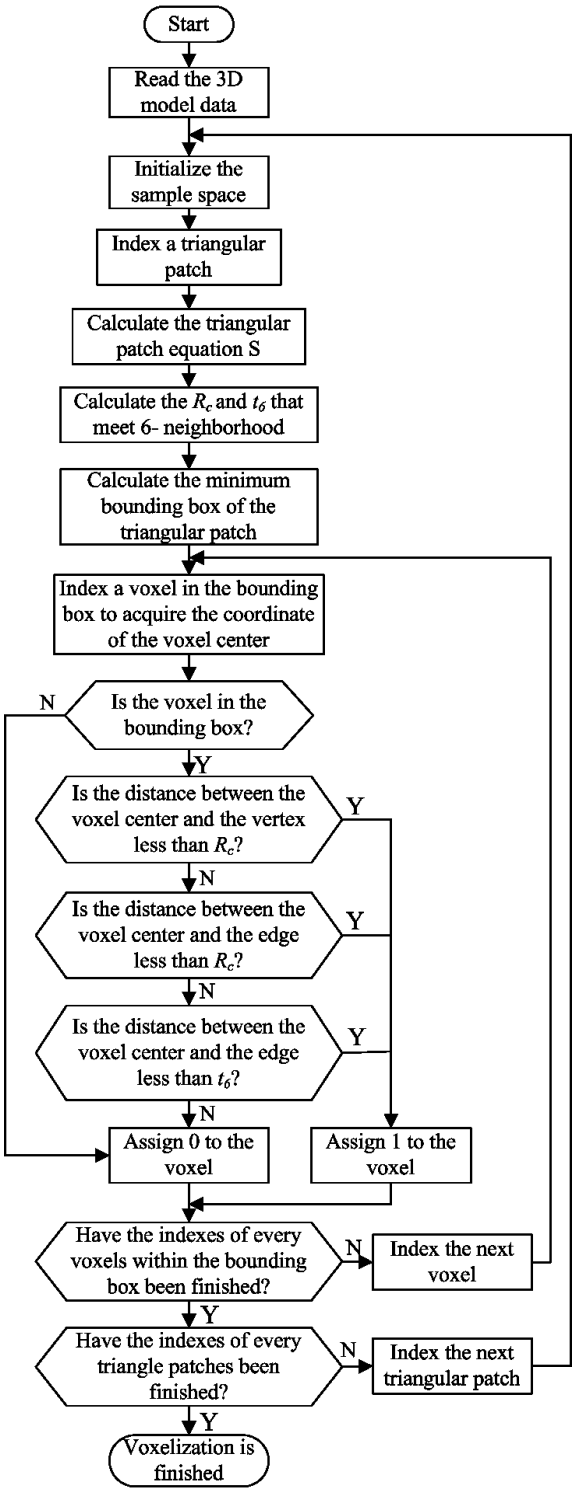
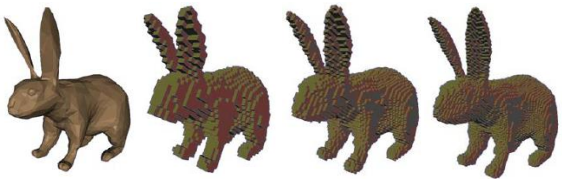


Fig. 7 The algorithm flow chart

EXPERIMENT AND ANALYSIS

The algorithm is run in VC++6.0 by calling glut and glut32 function library of OpenGL. The hardware environment is: Intel (R) Rentium (R) IV 2.80 GHz processor, NVIDIA GeForce 9600GT display card, 1015 MB RAM.

First experiment is the voxelization of a rabbit with $l \times l \times l$ resolution. The result is as shown in Fig.8. In order to see different voxelization results with different resolutions clearly, each voxel is drawn as a cube by using OpenGL functions.



(a) 3D model (b) *l*=64 (c) *l*=100 (d) *l*=150

Fig.8 voxelization results with three resolutions

Second experiment is the voxelization of four different parts with 512×512×512 resolution. The voxel images are shown in Fig.9. The model parameters and voxelization time are given in table 1. The experiment demonstrate that the voxels model can well approximate the original model with a faster speed by using our voxelization algorithm. In addition, the voxelizing time is approximately proportional to the number of triangular patches with a certain resolution.

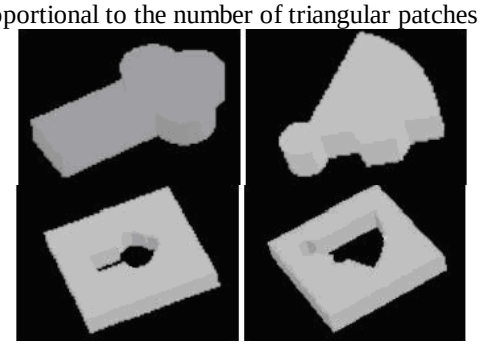


Fig.9 The voxel models

Table 1 The model's parameters and voxelization time

Model	The model's physical size (x, y, z)	The number of triangular patches	Resolution	Voxelization time/s
A part in the shape of plum-blossom	59.9,67.7,15	356	512	12.6
An assembly part in the shape of plum-blossom	100,100,16	196	512	7.5
A part in the shape of fan with a boss	60.6, 68.2,15	216	512	8.9
An assembly part in the shape of fan with a boss	100,100,16	208	512	7.7

CONCLUSION

A voxelization method of 3D polygon mesh models has been described in this paper. According to our research work, we summarized as follows:

- The 3D mesh Models were read Layer-by-layer to acquire the information of points, lines, surfaces, materials and textures by using OpenGL. Such information is irregular.
- The voxelization method presented in this paper can generate 6- neighborhood closed and 26-neighborhood closed voxels model. Very prominent or isolated single voxels could not exist in the voxel model due to the constraint between adjacent voxels. This method has strong anti-interference and meets the demands of fidelity, minimality and reciprocity in voxelization.
- The voxel models can approximate the original model without distortion and higher the resolution is, more authentic the model is. The voxelizing time is approximately proportional to the number of triangular patches with a certain resolution.
- The characteristic signals of the voxels model can be used to replace the characteristics of 3D mesh models approximately, which result to the convenient use of many classical signal processing method and to simplify the complexity of the problem.
- The experimental results indicate that the algorithm runs fast and can obtain good approximation effect.

REFERENCES

[1] Cui, C.Y., Shi, J.Y., **2004**. *Journal of Computer Aided Design & Graphics*, 16: 882-888.
[2] Kaufman, A., Shimony, E., **1986**. 3D scan- conversion algorithms for voxel- based graphics. Proceedings of the 1986 Workshop on Interactive 3D Graphics. New York.ACM:45-76.
[3] Wu, X.J., Liu W.J., Wang T.R., *et al*.,**2004**. *Journal of Computer Aided Design & Graphics*,4: 592-597.

[4] Wu, X.J., Liu, W.,**2005**. *Journal of Engineering Graphics*, 26:1-7.
[5] Rueda, A.J., Segura, R.J., Feito, F.R., *et al.*,**2004**. Voxelization of solids using simplicial coverings. The 12th Internet Conference on Computer Graphics, Visualization & Computer Vision. Plzen, Czech Republic. February 2-6, 2004, Science Press.
[6] Huang, J., Yagel, R., Filippov, V., **1998**. An accurate method for voxelizing polygon meshes. 1998 IEEE Symposium on Volume Visualization. New York, 1998. IEEE: 119-126.
[7] Hao, C.Z.,Yang, Q.F.,**2007**. *Modern Manufacturing Engineering*, 7: 122-125.
[8] Lv, G.X., Pan, M., Wang Z.G., *et al.*, **2007**. *Geography and Geo-Information Science*, 23: 5-9.
[9] Lu, H.X., Zheng, W.T., Chen, W., *et al.*,**2010**. *Journal of Computer-Aided Design & Graphics*, 22: 373-381.
[10] Mu, B., Pan, M., Deng J., **2010**. *Geography and Geo-Information Science*, 26: 27-31.
[11] Peace dove studio, **2006**. OpenGL advanced programming and development of visualization system. Beijing: China Water Power Press: 68-107.