

基于 VC++ 和 OpenGL 的 STL 文件读取显示^①

STL Files Reading and Visualization Based on VC++ and OpenGL

严栳铭 钟艳如 (桂林电子科技大学 计算机与控制学院 广西 桂林 541004)

摘要: STL 是三维模型常用的文件格式。对 STL 文件进行读取和显示, 是对模型进行后续操作的前提。在对 STL 文件格式进行详细分析的基础上, 以 VC++ 作为开发平台; 利用 C++ 标准 IO 库的 `ifstream` 类型定义对象并绑定 STL 文件; 再用标准库 `string` 类型中的 `getline` 逐行读取绑定的 STL 文件; 最后, 应用 OpenGL 中绘制三角面片编程技术实现对 STL 文件格式的直观显示。通过试验验证了读取和显示效果。

关键词: STL 文件 ASCII 三维模型 三角面片 OpenGL

1 引言

STL (Stereo lithographic) 文件格式是美国 3D SYSTEMS 公司提出的三维实体造型系统的一个接口标准, 其接口格式规范, 目前已被工业界认为是快速成形(rapid prototyping)领域的标准描述文件格式。在逆向工程、有限元分析、医学成像系统、文物保护等方面有广泛的应用^[1]。对 STL 文件的读取与显示是其应用的基础与前提, 本文对 STL 文件特点进行了详细的分析并在 VC++ 平台上, 利用 C++ 标准库中提供的文件流读取及字符操作等功能结合 OpenGL 中三角面片绘制的编程技术实现了对 STL 文件的读取与显示。

2 STL 文件格式的结构

为了正确地读取三维模型原始数据, 更好地利用三维图形的知识重现三维模型原型, 首先必须充分了解三维模型原始数据文件的格式, 理解三维数据内部的组织结构。

STL 文件是一种用许多空间小三角形面片逼近三维实体表面的数据模型, STL 模型的数据通过给出组成三角形法向量的 3 个分量(用于确定三角面片的正反方向)及三角形的 3 个顶点坐标来实现, 一个完整的 STL 文件记载了组成实体模型的所有三角形面片的法向量数据和顶点坐标数据信息。目前的 STL 文件格式包括二进制文件(BINARY)和文本文件(ASCII)两种^[2]。

2.1 TL 的二进制格式

二进制 STL 文件用固定的字节数来给出三角面片的几何信息^[3]。文件起始的 80 个字节是文件头, 用于存贮零件名; 紧接着用 4 个字节的整数来描述模型的三角面片个数, 后面逐个给出每个三角面片的几何信息。每个三角面片占用固定的 50 个字节, 依次是 3 个 4 字节浮点数(角面片的法向量)3 个 4 字节浮点数(1 个顶点的坐标)3 个 4 字节浮点数(2 个顶点的坐标)3 个 4 字节浮点数(3 个顶点的坐标)个三角面片的最后 2 个字节用来描述三角面片的属性信息。一个完整二进制 STL 文件的大小为三角形面片数乘以 50 再加上 84 个字节, 总共 134 个字节。

2.2 TL 的 ASCII 文件格式

ASCII 码格式的 STL 文件逐行给出三角面片的几何信息, 每一行以 1 个或 2 个关键字开头。在 STL 文件中的三角面片的信息单元 `facet` 是一个带矢量方向的三角面片, STL 三维模型就是由一系列这样的三角面片构成。整个 STL 文件的首行给出了文件路径及文件名。在一个 STL 文件中, 每一个 `facet` 由 7 行数据组成, `facet normal` 是三角面片指向实体外部的法向量坐标, `outer loop` 说明随后的 3 行数据分别是三角面片的 3 个顶点坐标, 3 顶点沿指向实体外部的法向量方向逆时针排列^[4]。

ASCII 格式的 STL 文件结构如下:

^① 基金项目: 广西自然科学基金项目(桂科自 0728211)

收稿时间: 2008-09-10

```

solid filename stl //文件路径及文件名
facet normal x y z //三角面片法向量的 3 个分量值
    outer loop
    vertex x y z //三角面片第一个顶点坐标
    vertex x y z //三角面片第二个顶点坐标
    vertex x y z //三角面片第三个顶点坐标
    endloop
endfacet //完成一个三角面片定义
---
endsolid filename stl //整个 STL 文件定义结束

```

一个具体 STL 文件如下:

```

solid C:\Temp\NOELParts\HMJ_PRT.STL
created by CtrlView V2.70
facet normal 0 1 0
    outer loop
    vertex 67.4752 34.7981 31.693
    vertex 68.4358 34.7981 30.3941
    vertex 69.1631 34.7981 28.9516
    endloop
endfacet
---

```

通过对 STL 两种文件格式的分析可知,二进制格式文件较小(通常是 ASCII 码格式的 1/5),节省文件存储空间,而 ASCII 码格式的文件可读性更强,更容易进行进一步的数据处理。

3 STL文件的读取与显示

本文以 STL 三维模型文件为数据源,根据对文件格式和内部数据结构的分析,采用 OpenGL 作为三维图形接口,并以 VC++6.0 为开发平台^[5]。利用 C++ 标准 IO 库的 ifstream 类型定义对象并绑定 STL 文件;再用标准库 string 类型中的 getline 逐行读取绑定的 STL 文件;然后对读入的数据进行是否为顶点信息的判断,存储其中的顶点数据。再应用 OpenGL 编程技术实现对 STL 三维模型文件的显示。

读取 STL 文件时,只需要读取 STL 文件中表示向量和三角形顶点的相应数据,不需要读文件中的其它信息。依次按逆时针方向读入各个三角形面片的 3 顶点坐标值。由于三角面片外法矢量可以通过右手螺

旋法则由 3 顶点坐标值计算出来,因此可不对其进行存储,以节省存储空间。如果后续处理需用到法矢量,可利用以下的外法矢量计算公式:

$$\begin{cases} n_x = (v1_y - v3_y)(v2_z - v3_z) - (v1_z - v3_z)(v2_y - v3_y) \\ n_y = (v1_z - v3_z)(v2_x - v3_x) - (v2_z - v3_z)(v1_x - v3_x) \\ n_z = (v1_x - v3_x)(v2_y - v3_y) - (v2_x - v3_x)(v1_y - v3_y) \end{cases} \quad (1)$$

3.1 定义顶点

STL 三维模型文件由一系列的三角面片组成,每一个三角面片由三维空间中对应的三个顶点组成。STL 文件的读取与显示实质就是对 STL 文件中的顶点信息进行读取并直观显示,因此,读取显示 STL 文件首先要解决的就是顶点的定义问题,本文通过类 Vertex 定义三角面片顶点,Vertex 类定义的部分代码如下:

```

class Vertex
{
public:
    Vertex(double _x=0, double _y=0, double
_z=0)
        :x(_x), y(_y), z(_z)//三坐标变量赋初值
    {}
    ---
private://定义顶点三坐标
    double x_;
    double y_;
    double z_;
};

```

3.2 定义读取功能

由前面的分析我们知道,STL 文件以行为标准存储不同信息。根据文件格式的该特点,本文首先应用 C++ 标准 IO 库的 ifstream 类型定义对象绑定待读取的 STL 文件;再采用逐行读取的方式用标准库 string 类型中的 getline 函数从 ifstream 对象绑定的 STL 文件中读入数据。由于只需用到其中的顶点信息,因此,通过判断字符 Vertex,然后读取其后的数据即可。本文在类 VertexReader 中定义了读取 STL 文件的相关操作函数,VertexReader 类定义的部分代码如下:

```

class VertexReader
{
public:

```

```

---
bool Read(VertexArray& vertices);//顶点信息读取函数
private:
void ReadLine(string& line);//读取行信息函数
void PushLine(VertexArray& vertices);//读取顶点函数
---
};

```

在函数 `Read(VertexArray& vertices)` 中通过 `ifstream` 定义对象绑定并打开 STL 文件, 为进行数据信息的读取、保存做准备。

```

bool VertexReader::Read(VertexArray& vertices)
{
    ifstream          infile(FileName.c_str()),
    ifstream::in;//打开要读取的 STL 文件
    ---
    ReadLine(line);
    PushLine(vertices);
    ---
}

```

`ReadLine(string& line)` 函数不对 `Read(VertexArray& vertices)` 函数中读入的行字符串信息进行是否为三角形顶点字符串信息的判断, 而只对读入的字符串进行存储, 并通过 `strtok(buff, Del)` 函数对读入的每行字符串信息进行分隔, 以区别读入的每个字符, 为下一步对三角形顶点字符串信息的判断做准备。部分实现程序如下:

```

void VertexReader::ReadLine(string& line)
{
    ---
    char* token = strtok(buff, Del);//分解字符串为一组标记串
        while (token) {
            Tken.push_back(token);
            token = strtok(NULL, Del);
        }
}

```

通过前面对 STL 文件的结构分析可知, STL 文件由文件路径及文件名、三角面片法向量、三角形面片的顶点、三角形顶点定义开始标识符(`outer loop`)及结束标识符(`endloop`)和三角形面片定义结束标识符(`endfacet`)等字符串信息构成。在本文中只需要其中的三角形顶点信息, 因此要对读入的字符串进行是否为三角形顶点信息的判断, 这里通过 `PushLine(VertexArray& vertices)` 函数来实现判断。文中我们具体用函数 `stricmp()` 来判断读入信息是否为顶点; 由于读入的顶点信息为字符类型而在显示时要用到的是浮点类型的顶点信息, 因此要通过函数 `atof()` 进行类型的转化。部分实现程序如下:

```

void VertexReader::PushLine(VertexArray& vertices)
{
    if (stricmp(Tken[0].c_str(), "vertex")==0)
    { //判断读取的是否为顶点信息
        double x = atof(Tken[1].c_str());//把字符串转化成浮点数
        double y = atof(Tken[2].c_str());
        double z = atof(Tken[3].c_str());
        vertices.push_back(Vertex(x, y, z)); //对点进行读入保存
    }
}

```

3.3 STL 文件的显示

在 OpenGL 函数库中, 提供了直接渲染三角面片的方法 `glBegin(GL_TRIANGLES)` 和 `glEnd()`, 因而利用 OpenGL 来实现 STL 三维模型的真实感图形显示^[6], 更加具有优势。部分实现程序如下:

```

void CWelcomeView::myplane()
{
    ---
    for(int i=0; i<vertices.size(); i=i+3)
    { glBegin(GL_TRIANGLES);//绘制三角面片
        glVertex3f(vertices[i].X0,vertices[i].Y0,vertices[i].Z0);
        glVertex3f(vertices[i+1].X0,vertices[i+1].Y0,vertices[i+1].Z0),

```

```
);
glVertex3f(vertices[i+2].X0,vertices[i+2].Y0,ver
tices[i+2].Z0
);
    glEnd();
    }
    }
```

注意,在 OpenGL 中投影是进行模型显示的关键技术^[7],主要有两种类型,一种是正投影(orthographic projection)另一种是透视投影(perspective projection)。由于透视投影更符合人类视觉,使模型显示更具有真实感,因此本文采用了该种投影方式。透视投影通过函数 void gluPerspective(Gldouble fovy, Gldouble aspect, Gldouble zNear, Gldouble zFar)其中的参数分别表示垂直方向的视野角度、高度到宽度的纵横比,以及近端和远端裁剪平面之间的距离。恰当的参数设置可以得到更好的显示效果。

3.4 试验

本文在对 STL 文件特点进行详细分析的基础上,利用 Visual C++ 6.0 平台和 OpenGL 编程技术实现了对 STL 文件的读取与显示功能。为了验证系统显示三维模型的实际效果,以 purdue 大学提供的 3D 模型库作为数据源,进行了读取显示试验。图 1 和图 2 是其中齿轮和弯管零件三维模型的显示效果图。

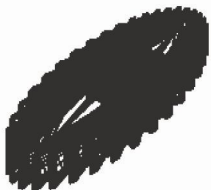


图 1 齿轮



图 2 弯管

4 结束语

对 STL 三维模型文件的读取与显示是对模型一切后续操作的前提和基础。本文通过对 STL 文件的研究;以 VC++ 作为开发平台;利用 C++ 标准 IO 库的 ifstream 类型定义对象并绑定 STL 文件;再用标准库 string 类型中的 getline 逐行读取绑定的 STL 文件并对其中的三角面片顶点数据进行存储;再结合 OpenGL 编程技术实现了对 STL 文件读取与显示的功能。为进一步对 STL 文件进行冗余数据处理以及其它相关领域如基于内容的三维模型检索的研究做了必要的准备。今后,在 STL 文件快速读取显示方面还有待于进一步的研究。

参考文献

- 1 卫伟,周来水,张丽艳.海量 STL 文件的快速读取与显示.机械科学与技术,2006,25(8):935-938.
- 2 郎兴华,郭阳,林亨,张伟.STL 模型的立体显示及其多屏拼接.系统仿真学报,2004,16(4):740-744.
- 3 杨晟院,舒适.基于 STL 文件的三角形表面网格的特征线提取.计算机工程与应用,2008,44(4):14-19.
- 4 纪峰,陈荔,李占利.基于 STL 文件的模型及应用.长安大学学报,2006,26(1):104-107.
- 5 马安鹏.Visual C++6.0 程序设计导学.北京:清华大学出版社,2002.
- 6 郭兆荣.Visual C++ OpenGL 应用程序开发.北京:人民邮电出版社,2006.
- 7 Richard S.OpenGL 超级宝典.徐波,译.3 版.北京:人民邮电出版社,2005.