在STL文件渲染程序开发中的动态链接库创建*

曹　驰[1,2]，齐国庆[1]，刘洪军[1]，郝　远[1]

(1. 兰州理工大学 甘肃省有色金属新材料省部共建国家重点实验室，甘肃 兰州 730050；2. 长风机器厂，甘肃 兰州 730070)

摘　要：STL文件的渲染是快速成型三维模型数据处理的重要内容之一。本文应用OpenGL开放性图形库和Visual C++编程语言来实现STL数据模型的渲染，并且将OpenGL的一些函数封装在一起，创建动态链接库，使得程序更具模块性，软件的可移植性加强，有利于软件的维护和进一步开发。

关键词：快速成型；OpenGL；STL；模型；动态链接库

中图分类号：TP 311　　**文献标志码**：A

Dynamic Link Library Creation in Development of STL File Rendering Program

CAO Chi[1,2], QI Guoqing[1], LIU Hongjun[1], HAO Yuan[1]

(1. State Key Laboratory of Gansu Advanced Non-ferrous Metal Materials, Lanzhou University of Technology, Lanzhou 730050, China; 2. Changfeng Machine Factory, Lanzhou 730070, China)

Abstract: The rendering process of STL file is an important content of data treatment for 3D model in Rapid Prototyping process. In this paper, the rendering of STL data model was carried out by using the tool of function library in OpenGL and Visual C++ 6.0 program language, and then the Dynamic Link Library (DLL) was created by encapsulating some OpenGL graphic function. This method makes the program more modularity and can enhance the portability of software, so it is beneficial to maintenance and further development of software.

Key words: Rapid prototyping, OpenGL, STL Model, Dynamic link library

快速成型技术可以迅速实现从CAD模型到实体零件的制造，是集CAD、CAM、激光技术、数控技术、材料科学等为一体的先进制造技术[1]，目前已经成为产品创新和快速反应市场的重要智能工具，获得了越来越广泛的应用。快速成型系统一般以STL文件作为加工的数据文件，在对STL模型文件进行分层切片后生成扫描或者加工轨迹数据，然后，输入到快速成型机进行快速原型零件的制作。由于从三维CAD模型转换而来的STL文件需要在加工前进行评估、检测、定位等工作，因此有必要对STL文件进行处理，使其能够以三维真实感图形显示，并能进行平移、旋转和缩放等模型辅助变换。在此过程中，对STL模型进行有效地渲染非常关键，针对STL模型的渲染已经做过很多相关的研究工作[2-6]，可以结合OpenGL等实现STL格式的实体显示和变换。但是STL渲染程序要用于其他相关程序的开发，就要具备较强的可移植性，否则开发新的程序时往往要重新编写STL显示程序，由此会带来很多不必要的重复性工作。

动态链接库(Dynamic Link Library，缩写为DLL)，是一个可以被其他应用程序共享的程序模块，其中封装了一些可以被共享的例程和资源。它和可执行文件(exe)非常类似，区别在于DLL中虽然包含了可执行文件代码却不能单独执行，而应由其他应用程序直接或间接调用。在动态链接的情况下，一个库的函数和数据并不复制到可执行文件中，而是仅仅当应用程序开始运行时，才在应用程序和相应的DLL之间建立链接关系，去执行DLL中的函数代码[7]。如果采用动态链接库将STL渲染相关的类封装，就可以重复利用创建的类和函数，可方便地用于别的相关软件的开发。因此本文中将DLL技术用于STL渲染程序的开发，使程序可移植性弱的问题得到解决。STL模型的三维图形开发环境用VC++和OpenGL来创建，程序中包括2个函数库(几何基本工具库和OpenGL显示库)，采用模块化的层次结构，由多个动态链接库(DLL)组成，从而节省程序运行时的内存消耗，并且使应用程序易于开发和维护。

1　几何基本工具库的设计

点、矢量、矩阵是几何变换中常用的对象，涉及的计算多、运算量大，要为其设计相应的C++类，并以这些类和相关的计算函数为主要内容开发几何基本工具库——Calu.Dll，该工具库输出基本几何对象类和几何计算函数，如描述点、矢量、矩阵的类以及相关计算函数等，可实现系统中的几何造型、操作和显示等功能。

点是构成所有几何对象的最基本元素，对于一个空间的点(x,y,z)其数据结构如下：

```
Struct Point3D{
    double x;
    double y;
```

```
double z;
}POINT3D;
```

对于点，设计一个C++类——CPoint3D，在这个类中可以实现对点的操作，还可以添加函数用于点与矢量的加减运算、点和矩阵之间的运算以及矩阵变换的运算。对它的类定义如下：

```
Class CPoint3D{
  public:
    CPoint3D();
    ~CPoint3D();
    ……
    ……
  };
```

同样可以定义矢量和矩阵的数据结构以及构造矢量和矩阵的类CVector3D和CMetrix3D。

在这个几何基本工具库中包括的运算关系有：点与矢量的运算、矢量之间的运算、矢量与标量的运算、矩阵的平移运算、矩阵的旋转运算和矩阵的缩放运算等。

2 OpenGL显示库的设计

OpenGL即开放图形库(Open Graphics Library)是美国SGI公司开发的一个功能强大的图形软件应用程序接口。它实际上是一个三维图形渲染库和三维模型库，有着强大的图形函数，开发者不但可以使用自己的数据，而且还可以使用其他不同格式的数据源，如STL、DXF和3D格式文件等。这种灵活性节约了开发时间，提高了开发效率[8]。

OpenGL显示库为STLView.dll，封装了OpenGL中的类，实现OpenGL的初始化、三维模型的显示、光照材质的设置和交互式选择等。

在MFC环境下创建OpenGL应用程序的步骤在文献[9]中有描叙，在此就不再赘述。在建立了基本的应用程序框架之后，就要进行STL文件的渲染。在动态链接库中要进行的操作主要有STL文件的读取显示、三维观察、材质及光照的设置、交互式选择等。

2.1 STL文件的读取

STL文件是用许多三角面片来逼近三维实体表面的数据模型，STL文件中记录了三角形的顶点坐标和法向矢量，STL文件分为二进制格式和ASCⅡ格式。ASCⅡ格式的STL文件结构为：

```
solid PRT0001
facet normal nx ny nz
    outer loop
      vertex v1x v1y v1z
      vertex v2x v2y v2z
      vertex v3x v3x v3x
    endloop
  endfacet
  ……
  ……
end solid PRT0001
```

利用ASCⅡ格式可以定义三角形的数据结构，若STL数据模型中三角形数目不大，则可在程序中定义一个数组来存放数据。在通常情况下，由于三角形数目大小不固定，所以定义一个动态数组或采用链表结构来存储数据更实用。

```
//STL文件的读取
BOOL CSTLView::ReadSTLFile(LPCTSTR stlfile)
{ FILE *file;
if((file=fopen(stlfile,"r"))==NULL
return FALSA;
char str[80];
if((str,"normal",6)==0){
fscanf(file,"%lf %lf %lf",&(tri->normal.dx),&(tri->normal.dy),&(tri->normal.dz));

fscanf(file,"%lf %lf %lf",&(tri->vex[0].x),&(tri->vex[0].y),& tri->vex[0].z),);
fscanf(file,"%lf %lf %lf",&(tri->vex[1].x),&(tri->vex[1].y),& tri->vex[1].z),);
fscanf(file,"%lf %lf %lf",&(tri->vex[2].x),&(tri->vex[2].y),& tri->vex[2].z),);
}
Return TRUE;
}
```

2.2 STL数据模型显示

用OpenGL函数对数据模型进行显示时，多边形分为正面和反面，在使用多边形的过程中可以对多边形的2个面分别进行操作。OpenGL以glPolygonMode(GLenum face, GLenom mode)函数来设置多边行的渲染模式。Draw()函数是用于绘制读入的STL数据模型。关键代码如下：

```
Void CSTLView::Draw()
{
  glBegin(GL_TRIANGLES);
for (i=1;i<n;i++) {
//绘制三角形的方向矢量
glNormal3f(pDoc->NormalArray[i].x, pDoc->NormalArray[i].y,  pDoc->NormalArray[i].z);
//绘制三角形的第一点坐标
glVertex3f(pDoc->VertexArray[p].x,pDoc->VertexArray[p].y,  pDoc->VertexArray[p].z);
//绘制三角形的第二点坐标
glVertex3f(pDoc->VertexArray[p+1].x,pDoc->
```

```
VertexArray[p+1].y,    pDoc->VertexArray[p+1].z);
    //绘制三角形的第三点坐标
    glVertex3f(pDoc->VertexArray[p+2].x,pDoc->
VertexArray[p+2].y,pDoc->VertexArray[p+2].z);
      p=p+3;
     }
      glEnd();
    }
```

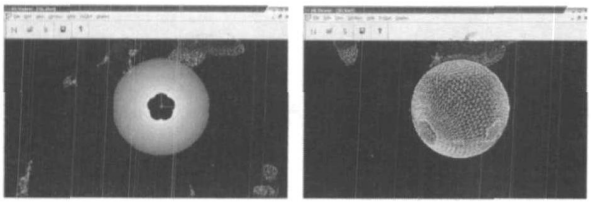
2.3 STL 文件的三维观察

在 OpenGL 中，屏幕所显示的景物范围是由 OpenGL 中的取景设置决定的。取景操作是通过视口变换、投影变换来实现的。视口变换可设置应用程序的窗口大小，通过函数 void glViewport()实现。

在图形学中，几何变换包括视点变换和模型变换。几何变换指三维场景中的物体运动姿势的变化，包括物体的平移、旋转和缩放。OpenGL 中提供了 3 个函数来实现几何变换，它们分别是 glTranslate()、glRotate()和 glScale()，从而可以确定一个物体在场景中的位置、旋转角度和缩放比例。

利用 OpenGL 函数进行多边形的模型显示时，多边形是有前、后面之分的，通常把顶点排序为逆时针的多边形定义为朝前。OpenGL 以 glPolygonMode(Glenum face，Glenum mode)设置多边形渲染模式。参数 face 取 GL_FRONT_AND_BACK 表示多边形的正面和背面都使用填充或线框模式；参数 mode 取 GL_LINE 表示模型以线框模式显示，取 GL_FILL 表示以填充模式显示。

在程序的工具条中添加按钮，并对其添加相应的消息响应函数，以实现模型的不同角度和大小。为了更加详细的观察模型的信息，可以分别使用填充模式和线框模式来表示模型。图 1 为加载的一个 STL 格式模型分别用以上 2 种模式表示。



a) 填充模式下的模型显示　　b) 线框模式下的模型显示

图 1　STL 模型显示及视角转换

2.4 交互式选择

在软件的使用中，通常要用到用户和图形之间的交互式操作，例如在此程序中实现的功能，如在多个对象中实现单个模型的选取、模型颜色的设置等，可以通过工具条、对话框、键盘、鼠标来实现用户和图形对象之间的交互操作。通过交互式选择可以让用户更加方便、直观地查看模型信息。

在 OpenGL 中有 3 种操作模型：渲染模式(render mode)、选择模式(selection mode)和反馈模式(feedback mode)，由于图形的交互操作涉及的内容多，在此主要描叙使用选择模式的方法。在 OpenGL 中使用选择模式的具体步骤和程序片段如下：

1) 在使用选择模式之前，首先调用 glSelectBuffer()函数指定存放返回命令中记录的数组；

2) 调用 glRenderModel(GL_SELECT)函数进入选择模式；

3) 调用 glInitNames()初始化名字堆栈；

4) 调用 glPushName()函数将物体的名字压入名字堆栈中；

5) 定义选择体。选择体和场景的观察体有所不同，可以用函数 glPushMatrix()和 glPopMatrix()来保存和恢复观察体；

6) 交替使用绘图命令和名字堆栈命令，使每一个图元都被赋予一个适当的名字；

7) 调用 glRenderMode(GL_RENDER)函数退出选择模式并重新回到一般绘图模式；

8) 处理命中记录数组中的数据。

```
GLuint  selectBuff[64];
GLint  hits,viewport[4];
//指定存放返回命令中记录的数组
glSelectBuffer(64,selectBuff);
//取得视景体
glGetIntegerv(GL_VIEWPORT,viewport);
// 选择投影坐标系
glMatrixMode(GL_PROJECTION);
// 压入投影矩阵
glPushMatrix();
//进入选择模式
glRenderMode(GL_SELECT);
//重设数组
glLoadIdentity();
//在鼠标位置生成 2×2 像素
gluPickMatrix(xPos,yPos,2,2,viewport);
//调用 RenderScene()函数绘图
RenderScene();
//返回到一般绘图模式
hits = glRenderMode(GL_RENDER);
//重新回到投影坐标系
glMatrixMode(GL_PROJECTION);
glPopMatrix();
```

3 动态链接库的创建

当几何基本工具库 Calu.dll 和显示库 STLView.dll 中的类和函数创建完毕后，就可在主程序

中调用这 2 个动态链接库。动态链接库的创建和调用步骤如下。

3.1 动态链接库的创建步骤

1) 进入 VC++集成开发环境;

2) 在新建对话框工程中选择"MFC AppWizard(dll)"来创建 DLL,命名为 STLView;

3) 选择 MFC Extension DLL 来创建 MFC 扩展 DLL;

4) 在随后的对话框中单击 OK 按钮,就可生成 DLL 的框架文件。

编译项目一切正常后,就可以在项目中添加类和函数,其方法和在普通的 MFC 中插入类的方法一样,只需要将类的头文件(*h)和源文件(*cpp)加入到项目中即可,需要注意的是,对于要输出的类,需要在声明类时添加 AFX_EXT_CLASS 的标识符。对于要输出的全局函数,需要在声明类时添加 AFX_EXT_API 的标识符。

```
Class AFX_EXT_CLASS CPoint3D
  { public:
    CPoint3D();
    ~CPoint3D();
    ……
    ……
  };
```

完成类和函数的添加后,编译链接这个 DLL 项目,若不出现错误,即可生成 STLView·dll 文件和 STLView·lib 文件。

3.2 动态链接库的调用

调用 DLL 分为隐式链接和显式链接,由于本文采用的是扩展 DLL,所以只能用显式链接。显式加载 DLL 的步骤如下:

1) 调用 LoadLibrary 函数。这个函数返回一个该库的句柄。如果函数失败,返回 NULL,并且调用函数 GetLastError,以确定失败的原因;

2) 用 LoadLibrary 函数返回来的句柄调用函数 GetProcAddress,把想用的函数地址传递过来,GetProcAddress 函数返回该函数的地址,然后可用该地址来调用此函数;

3) 当用完 DLL 后,使用函数 FreeLibrary,把该库从内存中卸载掉,并释放所有与之关联的资源。

4 结语

1) 用 VC++和 OpenGL 实现了 STL 模型的渲染程序设计和编写,使用了动态链接库,创建了几何基本工具库 Calu·dll 和显示库 STLView·Dll,使程序的结构模块划分更清晰。

2) 动态链接库使程序的可移植性增强,而且根据软件设计的要求还可以对动态链接库进行后续的扩充和增强,用来完善其功能。

3) 本文开发的程序实现了 STL 实体模型的读取和显示,可以对 STL 模型进行不同角度、不同大小比例和不同绘图模式的观察,还可以使用不同的颜色以便更好地区分多个模型,从而使 STL 数据模型的显示功能更加完善。

参考文献

[1] Xue Y, Gu P·A review of rapid prototyping technologies and systems[J]·Computer-Aided Design, 1996, 28(4): 307-318.

[2] 李春雷,倪俊芳·基于 Open GL 对 STL 文件描述实体的渲染技术研究[J]·苏州大学学报(工科版),2006,26(4): 24-26.

[3] 孙建平,曹志清,张家军·基于 OpenGL 的快速成型软件可视化研究[J]·机械工程师,2006(8):41-42.

[4] 李一,杜焱,杨大为·STL 格式实体真实感图形的显示工具开发[J]·沈阳工业学院学报,2003,22(3): 30-32.

[5] 黄常标,林俊义,江开勇·对 STL 模型的交互选择功能的实现[J]·计算机工程与应用,2004,40(17): 121-123.

[6] LI Jianguo, XU Mingheng·3D Flash display for STL model in rapid prototyping system[J]·International Journal of Plant Engineering and Management, 2003, 8(4): 206-209.

[7] 文欣绣,米西峰,赫枫龄·基于动态链接库实现软件见面组件化方法研究[J]·计算机应用与软件,2007,24(7):18-20.

[8] Bo Chen, Harry H Cheng·Interpretive openGL for computer graphics[J]·Computer & Graphics, 2005(29): 331-339.

[9] 刘金义,侯宝明·STL 格式实体的快速拓扑重建[J]·工程图学学报,2003(4):34-39.

* **甘肃省科技支撑项目**(0708GKCA057)

兰州理工大学博士基金项目(SB01200414)

作者简介:曹驰(1969-),男,高级工程师,博士研究生,主要从事模具设计与快速制造研究。

收稿日期:2008 年 12 月 26 日

责任编辑　吕菁