

# 用面向对象的方法实现 3DS 文件的读取与操纵<sup>\*</sup>

王益群<sup>1, 2</sup>, 黄 诚<sup>3</sup>

{<sup>1</sup> 南京工业大学 信息科学与工程学院, 江苏 南京 210009;  
<sup>2</sup> 盐城工学院 计算机科学技术系, 江苏 盐城 224003;  
<sup>3</sup> 上海盛大网络发展有限公司, 上海 201203

**摘 要:**介绍了利用面向对象的程序设计方法实现了 3DS 文件的读取。与面向过程的方法比较,面向对象方法对于理解 3DS 文件的读取、重绘及控制显得条理清晰,易于掌握,同时又方便移植到其它应用程序中。基于这一点,提出了将 3DS 文件的读取工作封装在一个类中,从而提高了程序的可读性。

**关键词:**面向对象;OpenGL;3DMAX;法向量

**中图分类号:** 文献标识码:A 文章编号:1671-5322(2005)03-0041-04

随着计算机技术的飞速发展,计算机动画、科学计算可视化和虚拟现实逐渐成为计算机图形学领域中三大重要技术,而三维图形又是这三大技术的核心内容<sup>[1]</sup>。三维图形现在已广泛应用于建筑学、医学、CAD/CAM/CAE、分子结构研究、模拟仿真、虚拟现实、广告业和娱乐业等诸多领域<sup>[2]</sup>,编程实现 3DS 文件的读取、重绘及控制具有重要意义。传统的基于过程的设计方法,由于相关函数多而且散杂同时数据又与函数分离,导致不便于程序的移植。为了克服这些缺点,本文采用了面向对象的程序设计方法,将 3DS 文件的形体数据读入到类中,然后由其成员函数利用 OpenGL 实现对 3DS 文件的读取、重绘及操纵。

## 1 3DS 文件结构及 OpenGL 简介

目前三维图形制作软件很多,如 AutoCAD、UG、3DMAX、Wavefront 等,因而其文件格式非常丰富。3DStudio 是 Autodesk 公司开发的一套在微机上制作三维动画的应用程序。本文利用由该软件产生的三维图形文件为例来实现三维图形的读取、操纵与控制。

3DS 文件由许多块组成,每个块首先描述其

信息类别,即该块是如何组成的。块的信息类别用 ID 来标识,块还包含了下一个块的相对位置信息。因此,即使不理解一个块的含义,也可以很容易地跳过它。要想对 3DS 文件的读写有一个明确的认识,必须准确知道它的含义。

OpenGL 全称为“开放式图形库”,是由 SGI 公司开发的三维图形 API,目前在图形开发领域已经成为工业标准;是国际上通用的图形硬件的软件接口,独立于操作系统和硬件环境,是一个完全可移植并且速度很快的 3D 图形建模库<sup>[3]</sup>。但 OpenGL 没有提供三维模型的高级命令,它是与通过基本的几何图元即点、线及多边形来建立三维模型的。目前有许多三维图形软件如 3DS MAX, AUTOCAD, 可以方便地建立物体模型,但又难以对其进行控制,比如 3DS MAX 的动画是基于关键帧而不是人机交互的。把这些模型转换成 OpenGL 程序,再对其进行控制则是一种比较理想的方法。在建模阶段可以利用三维软件建立模型,避免在 OpenGL 中用点、线和多边形来建立三维立体模型的繁重的劳动;在控制阶段,则可以通过 OpenGL 程序实现对模型的人机交互调整,从而可以随心所欲地利用现成的模型实现

<sup>\*</sup> 收稿日期:2005- -

万方数据  
作者简介:王益群(1978-),男,江苏建湖人,南京工业大学硕士研究生,高级程序员,研究方向:计算机应用技术。

虚拟现实。

## 2 实现方法<sup>[4~6]</sup>

### 2.1 文件内容的读取

#### 2.1.1 数据说明

在文件读取前,首先需要定义一系列的结构用来存放 3D 数据,如对象的材质、位置矢量、对象类(CTriObject)和对象列表类(CTriList)等,下面对两个重要的类作简单说明:

(1) CTriObject 类:主要用于处理 3DS 文件中的各种对象。

```
class CTriObject
{
    public:
        void drawGL(): //利用 OpenGL 绘画三角形
        void calNormals();//计算法向量
        .....
    Private://对象基本的成员变量
        float * x;float * y;float * z;//三角形的顶点
        float * nx;float * ny;float * nz;//面的法向量
        int * faces; //面信息
        tMaterial * materials;//材质信息
        .....
};
```

(2) CtriList 类:用于处理对象序列,它是 CTriObject 的集合,保存了三维模型的形体数据。

```
class CTriList
{
    public:
        void drawGL(); //绘制 3D 模型
        BOOL add(CTriObject * __object); //增加对象
        .....
    private:
        int numobjects; //CTriObject 对象的数目
        CTriObject * objects[100];
};
```

#### 2.1.2 对象的读取

文件的读取工作实质上是如何构造 CTriList 列表,也就是如何从 3DS 文件中读取数据来构造

CTriObject 对象,然后再组织到 CtriList 列表中。文件的读取主要包括循环读取 3D 对象所使用的材质,形体数据以及关键帧数据等,同时忽略其它一些不需要的数据,这一切都是通过 C3DSReader 类来实现的。

#### (1) C3DSReader 类的简单说明

```
class C3dsReader
{
    public:
        int ReadKFDATA(.....); //读取关键帧数据
        int Read3DSChunk(.....); //读取 3DS 块结构
        int ReadPointArray(.....); //读取顶点数据
        int ReadFaceArray(.....); //读取面数据
        int ReadMeshMatGroup(.....); //读取对象所用的材质
        int ReadTriObject(.....); //读入对象数据
        int ReadMatEntry(.....); //读入材质定义,并将其添加到材质库中
        int ReadMDATA(.....); //读取 3D 对象模型数据
        int Read3DSFile(.....); //读取 3DS 文件
        .....
    private:
        CTriList * DaList; //此链表用于存放形体对象数据
};
```

#### (2)读取的过程

```
int C3dsReader:: Read3DSFile ( long fileSize, long fileStart, long fileLen, FILE * fp)
{
    long chunkStart = ftell(fp);
    Chunk3DS chunk;
    MaterialDict * matdict = new MaterialDict();
    unsigned long version;
    while (chunkStart < fileStart + fileLen && Read3DSChunk(fp, chunk))
    {
```

```

switch (chunk.id)
{
case M3D_VERSION:
if ( ! ReadULong(fp, version)) goto
error;

break;
case MDATA:
if ( ! ReadMDATA(matdict, fileSize,
chunkStart, chunk.len, fp)) goto error;

break;
case KFDATA:
if ( ! ReadKFDATA(fileSize, chunk-
Start, chunk.len, fp)) goto error;

break;
default: // 忽略一些不需要的块
fseek(fp, chunkStart + chunk.len,
SEEK_SET);
}
chunkStart = ftell(fp);
}
if (matdict != NULL) delete matdict;
return TRUE;
error:
if (matdict != NULL) delete mat-
dict;

return FALSE;
}

```

## 2.2 算法向量

在将 3DS 文件读取后,就可以计算各个面的方向向量。法线矢量指定哪个方向垂直于多边形表面,用于光线和阴影计算。它与实体的光照模型密切相关,光线从不同入射角度到模型表面时,显示结果会有差别。如果忽略了法向量的计算,将不能重现模型的三维效果。

```

void CTriObject:: CalcNormal (int entry, float
out [ 3 ])

```

```

{
float v1[3],v2[3]
; // 根据空间三个点计算两个矢量值
v1[0] = x[ faces[entry]] - x[ faces[entry
+1] ];
v1[1] = y[ faces[entry]] - y[ faces[entry
+1] ];
v1[2] = z[ faces[entry]] - z[ faces[entry
+1] ];

```

```

+1] ];
v2[0] = x[ faces[entry+1] ] - x[ faces
[entry+2] ];
v2[1] = y[ faces[entry+1] ] - y[ faces
[entry+2] ];
v2[2] = z[ faces[entry+1] ] - z[ faces
[entry+2] ];
// 计算法向向量
out[0] = v1[1] * v2[2] - v1[2] * v2[1];
out[1] = v1[2] * v2[0] - v1[0] * v2[2];
out[2] = v1[0] * v2[1] - v1[1] * v2[0];
}

```

## 2.3 3D 对象的绘制

读取了模型数据,计算了法向量后,就可以在 VC++6.0 中重绘模型,下面分三步进行:

(1)处理 OnCreate 事件,在视图文件中初始化 OpenGL 绘图环境。该程序中将设置像素格式并创建 OpenGL 绘制描述表等,而只有创建了 OpenGL 绘制描述表和设置了像素格式后,OpenGL 的绘制命令才有效。同时还对 OpenGL 的一些默认属性进行了一些设置,如绘图模式、光照及材质属性等。

(2)处理 OnSize 事件,当窗口大小发生改变时,为了避免比例失调,需要对视口和显示比例进行一些调整,用 glViewport 函数指定视口矩形,用 gluPerspective 函数指定仰角、宽高比和深度裁剪值。

(3)处理 OnPaint 事件,这是重绘工作的最后一步了。当读入 3DS 文件后,由 OnPaint 函数来重画。该事件处理程序是所有 OpenGL 绘制程序的真正源头,所有绘制命令都直接或间接地由该函数调用。绘制时是通过每一个 CTriObject 对象调用成员函数 CTriObject::drawGL() 绘制每一个对象,从而在窗口中绘制出 3D 图形。其中绘制函数如下:

```

void CTriObject::drawGL()
{
int j;
glBegin(GL_TRIANGLES);
for (i=0; i < numfaces/3; i++)
{
j = 3 * i;
if (materialsapplied) glColor4f ( materials
[matfaces[i]].diffuseColor[0], materials[mat-

```

```
faces[i]].diffuseColor[1],
    materials[matfaces[i]].diffuseColor[2], 1/
materials[matfaces[i]].transparency );
    else glColor3f( 0.0f, 0.0f, 1.0f );
    ::glNormal3f( nx[i], ny[i], nz[i]);
    ::glVertex3f( x[faces[j]], y[faces[j]], z
[faces[j]]);
    ::glVertex3f( x[faces[j+1]], y[faces[j+
1]], z[faces[j+1]]);
    ::glVertex3f( x[faces[j+2]], y[faces[j+
2]], z[faces[j+2]]);
}
glEnd();
glPopMatrix();
}
```

3 结束语

本文主要讲述了结合面向对象和 OpenGL

参考文献：

[1] 孙波. OpenGL 编程实例学习教程[M]. 北京:北京大学出版社,2000.

[2] 王莹莹. 在 VC 中利用 OpenGL 实现动态效果图像的技巧[J]. 微型电脑应用,2002,18(6):50—52.

[3] 王雪莉,李宗民. Visual C++下 OpenGL 开发框架与应用[J]. 电脑学习,2004,(5):34—36.

[4] 郭景,雷鸣. 3DSMAX 模型在 OPENGL 中的读取与重现[J]. 自动化与仪表,2002,17(5):46—49.

[5] Richard S. Wright,Jr. Michael Sweet. OpenGL 超级宝典[M]. 北京:人民邮电出版社,2003.

[6] 和平鸽工作室. OpenGL 高级编程与可视化系统开发[M]. 北京:中国水利水电出版社,2001.

The Access to 3DS Files by the Method of  
Object Oriented Programming and its Operation

WANG Yi-qun<sup>1,2</sup>,HUANG Cheng<sup>3</sup>

1. College of Information Science and Engineering,Nanjing University of Technology ,Jiangsu Nanjing 210009,China;

2. Department of Computer Science and Technology ,Yancheng Institute of Technology ,Jiangsu Yancheng 224003,China;

3. Shanghai ShanDa Networking Development Limited Company, Shanghai 201203, China

**Abstract:**In this paper,there is a method of object oriented programming introducing how to access 3DS files. Compared with the method of procedure oriented programming,the former has many advantages. It is easier to understand how to access,draw and control this type of files and more convenient to replant it to other programs. Based on this,the authors put forward a method which encapsulates a class to finish the task of accessing,which improves the readability of the program.

**Keywords:**Object oriented;Open Graphic Library;3D MAX;Normal Vector

在 VC++6.0 环境下实现三维图像的读取,并在窗口中重绘。通过对本方法的了解,可以使用 OpenGL 的基本操作,来实现对 3DS 文件随心所欲的控制。把 3DMAX 模型应用于 OpenGL 中,降低了 OpenGL 复杂建模的难度,可以得到较真实的复杂物体模型。在飞行运动模拟的具体应用中,利用在 OpenGL 程序中引入 3DMAX 各种模型,得到了满意的三维动画效果。同样,我们还可以编写 AutoCAD、UG 等多种图形接口程序,把在 AutoCAD、UG 等 CAD 软件中建立的物体模型引入 OpenGL 的应用程序中,将可应用于飞行模拟、运动虚拟仿真、商业广告、游戏制作、影视采集、内装饰系统设计、服装设计等多个领域。使用现成的 3DS 文件,可以节省大量的重复且繁重的劳动,极大地提高了工作效率。