文章编号: 1673-1549(2011) 02-0182-04

数据库中三维模型的内存显示方式

程 旭, 赵 刚, 曾博才

(四川大学电子信息学院, 成都 610064)

摘 要: 在三维模型的应用中, 将模型文件存储在数据库中是一种很好的管理方式。一般显示模型的做法是把数据库中的模型存储在临时文件中, 再从这个文件中读取数据。文章提出了一种避免临时文件的显示方式, 即将数据库中的模型暂存在内存中的一块区域, 然后从内存中读取数据的显示方式。在这种方式下, 内存中的数据以字节为单位赋值给用于显示的数据结构。由于 STL(STereo Lithography)格式的三维模型文件有两种形式, 所以内存显示方式也有两种形式, 经过比较发现它们在读取时间上有比较明显的差异。

关键词: 数据库; STL; 存取; 显示

中图分类号: TP311.1 文献标识码: A

引 言

在三维模型的诸多应用中, 如何管理数量众多的模型文件是一个重要的问题。将模型存储在数据库中可以有效解决文件方式管理模型的弊端, 保证数据的安全性、一致性与完备性[1-2]。然而, 如何查看数据库中的模型, 又成为一个新的问题。

Microsoft的 ADO是用于访问信息的 API, 本文利用这个接口来访问数据库获得三维模型, 并探索了一种不经过临时文件显示三维模型的方法, 同时对该方法的效率作了一些分析。

1 三维模型的存取

用 Microsoft的 ADO(ActiveX Data Objects)连接数据库, 这样不必考虑因为数据库之间的差异带来的麻烦。

1.1 存 储

在存储模型的时候, 用 CFile类的 read函数将模型文件读入内存中的一片区域, 其首地址是一个 unsigned char类型的指针 pBuf指向的地址, 文件的长度是 nSize. 然而在利用 C++实现存取的时候, 并不能把 pBuf指向的内容直接用于 ADO的操作, 这是因为 ADO是以 COM技术为基础的, C++中的变量要应用到 ADO编程中必须对数据类型进行转换。

类 _variant_t是 C++类型, 封装了 COM中使用的 VARIANT数据类型, 管理资源的分配和释放, 通过这个类可以进行这样的转换。

VARIANT中有一个 SAFEARRAY结构体指针 parray, 可以用 SafeArrayPutElement方法把资源分配到 parray指向的空间, 由于 VARIANT已经被封装到了 _variant_t, 所以就相当于把资源分配到了类 _variant_t之中, 完成了 C++类型变量与 COM变量之间的转换。

那么在向数据库中存储模型的时候, 就可以先创建一个 _variant_t对象, 然后用 SafeArrayPutElement方法把 pBuf指向的数据按字节分配到 _variant_t对象之中, 最后用 AppendChunk方法把模型存入数据库。

1.2 读 取

在读取的时候应先把模型数据用 GetChunk方法读到 _variant_t类的对象中, 然后用 SafeArrayAccessData方法把 _variant_t对象的 parray指针指向的数据复制到 pBuf所指向的内存区域内, 这时候的 pBuf就可以在 C++中用于显示了。

收稿日期: 2011-01-05

作者简介: 程 旭(1985-), 男, 重庆垫江人, 硕士生, 主要从事三维图形方面的研究。

2 三维模型的显示

显示三维模型的时候,首先要把 pBuf指向的数据根据 STL格式赋给一个数据结构。由于 STL格式有 ASCII和二进制两种形式,所以赋值也有两种方法。

2.1 数据结构

STL格式的三维模型文件以三角面片为单位存储,每个存储单元包括这个三角面片的三个顶点和这个面片的法向。作为面向对象的应用,本文把这个结点装到了类 CTriChip里了:

```
Class CTriChip: public CObject
{
public:
CVector3D FaceNormal;
CPoint3D vex[3];
};
```

其中 CVector3D是向量类,封装了三个表示坐标的浮点数; CPoint3D是点类,也封装了三个表示坐标的浮点数。为了方便操作,本文用类模板 CTypedPtrArray来组织结点。

2.2 赋 值

2.2.1 ASCII形式

ASCII形式的文件结构如下[1]:

```
solid[零件名]
facet normal x y z
outer loop
vertex x y z
vertex x y z
vertex x y z
endloop
endfacet
……
endsolid[零件名]
```

对这种文件结构进行扫描的时候,有一个比较通用的步骤: (1)识别关键字“ nomal”; (2)读入面片的法向量; (3)跳过“ outer”和“ loop”两个关键字; (4)读入三个顶点; (5)把结点加到面表; (6)判断是否结束,如果没有结束,返回 (1). 流程图如图 1所示。

pBuf指向的内存区域是之前读取的 ASCII形式的文件,可以用 sscanf函数扫描其中的内容:形如 sscanf((char*)pBuf, "% lf% lf% lf% n", &x, &y, &z, &L)的语句读入坐标数据;形如 sscanf((char*)pBuf, "% * s% * s% n", &L)的语句跳过关键字。参数 L记录 sscanf每次扫描过的字节数。在每次扫描之后,都要把 pBuf后移 L位。

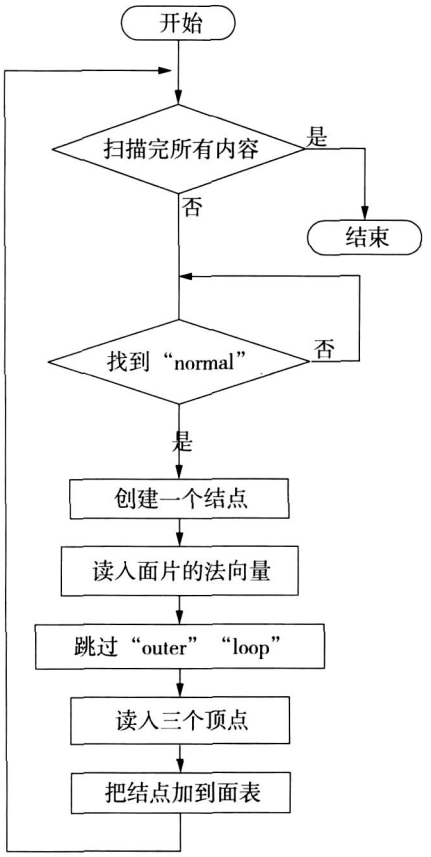

图 1 ASCII模型的赋值过程

2.2.2 二进制形式

二进制形式的文件是紧密排列的,四个字节表示一个浮点数,没有关键字,也没有空格和换行[3]。对这种文件结构进行扫描的时候,也有一个比较通用的步骤: (1)读取三角面片个数; (2)读入面片的法向量; (3)读入三个顶点; (4) 把结点加到面表; (5)三角面片个数减一; (6)判断三角面片个数是否小于 1,如果不是,返回 (2)。流程图如图 2所示。

pBuf指向的内存区域是之前读取的文件,可以用 memcpy来读入其中的内容:形如 memcpy(&x, pBuf+ = 4, 4)的语句读入坐标数据,并把指针后移四个字节。文件的头 80个字节用来存储其他附加信息,每个三角面片之后还有两个描述性的字节,这些地方在读入的时候都要跳过去。

2.2.3 两种形式的比较

在 ASCII形式的 STL文件中,一个字节表示一个浮

点数的一位,一个浮点数要占用多个字节,而且还有关键字,与二进制格式的文件相比,占用的存储空间大;从ASCII形式的 STL文件中读入数据的时候,由于 sscanf是一个格式化输入函数,有一个转换过程,与二进制形式的 STL文件相比,所花费的时间也比较长。由于这样的原因,ASCII形式一般只在调试的时候用到,而文件的传输用的是二进制形式[3]。

从图 3和图 4可以看到,在实际应用中,无论是存储空间还是读入时间,二进制形式都要优于 ASCII形式。

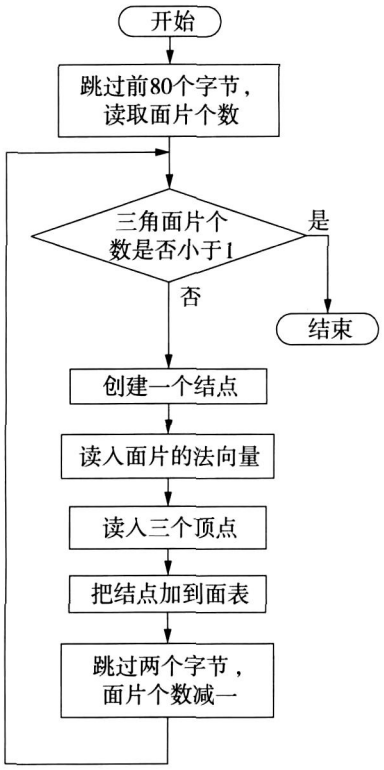

图 2 二进制模型的赋值过程

2.3 显 示

数据结构赋值过后,就可以用来显示了。OpenGL提供了直接绘制三角面片的方法,本文用它来渲染三维模型。以下是一段显示三维模型的代码[4],其中的 m_TriList就是由之前封装的结点类所构成的面表. 由于CTypedPtrArray这个类模板重载了"[ ]"这个操作符,可以像操作数组一样使用其中的元素,在操作面表的时候很方便。

```
for( int i= 0; i< m_TriList GetSize( ); i+ + ) {
glBegin( GL_TRIANGLES);
//开始绘制三角面片
glNormal3d( m_TriList[ i] - > FaceNormal dx,
m_TriList[ i] - > FaceNormal dy,
m_TriList[ i] - > FaceNormal dz);
//确定三角面片的法线方向
glVertex3d( m_TriList[ i] - > vex[ 0]. x,
m_TriList[ i] - > vex[ 0]. y,
m_TriList[ i] - > vex[ 0]. z);
//确定三角面片的第一个顶点
glVertex3d( m_TriList[ i] - > vex[ 1]. x,
m_TriList[ i] - > vex[ 1]. y,
m_TriList[ i] - > vex[ 1]. z);
//确定三角面片的第二个顶点
glVertex3d( m_TriList[ i] - > vex[ 2]. x,
m_TriList[ i] - > vex[ 2]. y,
m_TriList[ i] - > vex[ 2]. z);
//确定三角面片的第三个顶点
glEnd( );
//结束三角面片的绘制
}
```

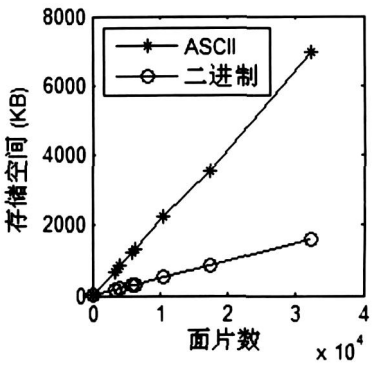

图 3 两种形式所占存储空间的比较

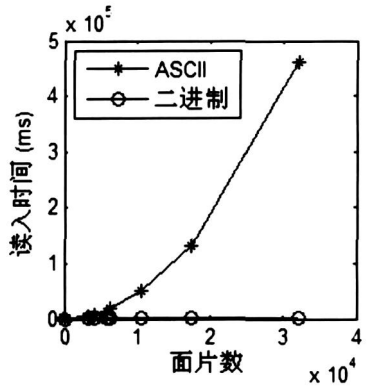

图 4 两种形式读入时间的比较

3 结束语

本文在将 STL模型文件在数据库中进行存取的基

础上,提出了一种把读取的数据不经过硬盘缓存,直接用于显示的方法。经过比较发现,STL文件的两种形式在存储空间和赋值时间上的差异,ASCII形式的 STL文件通常只是用于调试。

目前在快速成型[5]、虚拟制造[6]等领域中 STL模型有广泛应用。由于需要频繁的存取操作,所以在内存中组织数据,并把模型显示出来的方法在实际应用中有重要意义。

参 考 文 献:

[1] 胡逢凯,赵 刚,程 旭,等.基于 SQL Server数据库的三维模型存取研究与实现[J].四川理工学院学报:自然科学版,2010,23(2):212-215.

[2] 周松涛.基于关系数据库的三维模型库技术[J].测绘信息与工程,2005,30(6):30-31.

[3] 董得义,谢 川,李彦生,等.基于 MFC的 STL格式数据文件接口问题研讨[J].现代制造工程,2005(5):53-55.

[4] 王清辉,王 彪. Visual C++ CAD应用程序开发技术[M].北京:机械工业出版社,2003.

[5] Kang-Soo Lee, Sung-Hwan Kim. Non-uniform deformation of an STL model satisfying error criteria[J]. Computer-Aided Design, 2008, 42: 238-247.

[6] (韩)李建雨,著.袁清珂,译. CAD/CAM/CAE系统原理[M].北京:电子工业出版社,2006.

Memory Display of STL Model in Database

CHENG Xu, ZHAO Gang, ZENG Bo-cai

(School of Electronics and Information Engineering, Sichuan University, Chengdu 610064, China)

Abstract In the use of three-dimensional model, it is a good way to store the model files in a database to manage these files. When displaying a model which is stored in the database, it is commonly to store the model in a temporary file, and then read the data from the file to display it. This paper presents a way to display a model from the memory, instead of from a temporary file. In this way, the data stored in the memory is assigned to the data structure in bytes. As the STL format of three-dimensional model has two forms, there are two corresponding ways when displaying, they have obvious differences in reading speed.

Key words database; STL; access; display