

基于 SQL Server 数据库的三维模型存取研究与实现

胡逢恺, 赵刚, 程旭

(四川大学电子信息学院, 成都 610064)

摘要: 如何有效管理三维模型是三维图形应用中的一个重要问题。过去常用数据库结合文件系统存储和管理模型, 数据完备性和安全性较差, 而完全利用数据库存取模型可解决这些问题。本文给出了 1 种在数据库中存取三维模型的方法, 将 STL 模型文件以二进制大对象的形式存放在 SQL Server 数据库中, 通过 ADO 接口访问数据库, 利用 Stream 对象读取并输出模型文件。通过读取和显示模型, 验证了该方法的有效性。

关键词: SQL Server 数据库; 三维模型; STL; ADO

中图分类号: TP311.1

文献标识码: A

引言

在 CAD、CAM、GIS 等三维图形应用中, 通常先用三维建模软件如 AutoCAD、3D MAX、Solidworks 等创建实体对象的三维模型, 再调用模型数据进行显示、加工、导航等操作。建模软件生成的模型通常以文件形式存放在磁盘上, 如何有效地管理这些模型是一个重要问题。过去的管理系统常采用数据库 + 文件服务器的模式来管理模型文件, 即用数据库存储模型编号、名称、种类等相关信息, 模型本身仍以文件形式存放在服务器上, 在数据库中利用一个字段来记录模型文件的具体路径, 从而将两部分数据关联起来。文献 [1] 和文献 [2] 即采用这种方式。这样做有以下缺点: (1) 难以保证数据的一致性和完备性, 可能存在有记录而无文件或文件而无记录的情况; (2) 难以保证数据安全性, 对模型文件存在误删除、越权访问、病毒破坏等风险; (3) 数据的发布、备份和恢复较复杂, 需要考虑数据库和文件服务器两端的情况。解决以上问题的方法是将模型文件和模型信息存放在数据库的同一条记录中。文献 [3] 提出了一个利用关系数据库来实现三维模型库的方法, 解决了文件方式管理三维模型数据的弊端, 但没有具体讨论模型文件在数据库中的存取方法。

本文采用 SQL Server 数据库平台和 Visual C++

6.0 开发平台, 针对 STL 模型文件研究了三维模型在数据库中的存取方法, 并基于 OpenGL 三维图形接口实现了模型文件的读取及显示, 验证了该方法的有效性。

1 STL 模型文件简介

常见的三维模型文件有 3ds、dwg、stl 等, 本文采用较简单的 STL (StereoLithography 立体光造型) 文件格式。STL 是快速成型领域的标准格式, 目前主流的三维几何造型软件都支持 STL 文件的输出。STL 文件是一种用许多空间三角形小平面来逼近三维实体的数据模型, 每个三角面片由三个顶点和一个法向矢量构成, 如图 1 所示。顶点的排列顺序符合右手法则, 即沿指向实体外部的法向矢量方向逆时针排列。

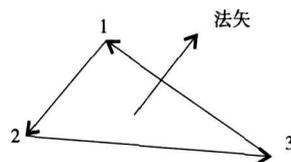


图 1 STL 模型中的三角面片

STL 格式分为 ASCII 码和二进制码两种格式。其中 ASCII 码格式是逐行给出三角面片的几何信息, 每行以关键字开头。一个 ASCII 码格式的 STL 文件如下所示^[4]:

```

solid< name> /三维模型实体名称
facet normal  $N_x N_y N_z$  /单位法矢量
outer loop
vertex  $V_{1x} V_{1y} V_{1z}$  /顶点 1坐标
vertex  $V_{2x} V_{2y} V_{2z}$  /顶点 2坐标
vertex  $V_{3x} V_{3y} V_{3z}$  /顶点 3坐标
endloop
endfacet//一个三角面片结束
.....
.....
endsolid < name> //文件结束

```

2 三维模型在数据库中的存取

2.1 数据类型及表结构

由 STL文件的数据格式可以看出, STL文件实际上是一长串的二进制码或字符串。在 SQL Server数据库中, 系统提供了 `text` `ntext` `image`三种 BLOB(binary large object二进制大对象)数据类型用于存储巨大的不定长度的二进制或者字符数据。其中, `text`数据类型用于存储非 Unicode的 BLOB数据, `ntext`数据类型用于存储 Unicode的 BLOB数据, `image`数据类型用于存储二进制 BLOB数据, 最大长度为 2GB。 `image`类型常用于存储图片, 本文用来存储三维模型。数据库中记录模型编号、名称、种类、属性等信息, 表结构如表 1所示。

表 1 三维模型库表结构

字段名称	字段类型	字段长度	备注
ID	int	10	模型编号
Name	varchar	100	模型名称
Type	varchar	100	模型种类
Property	varchar	200	模型属性
Model	image		模型文件

2.2 数据库访问接口

本文采用微软提供的 ADO (ActiveX Data Objects)访问 SQL Server数据库, 它是建立在 OLE DB底层技术之上的高级编程接口。OLE DB是一组 COM (Component Object Model组件对象模型)接口, 是新的数据库底层接口, 它封装了 ODBC的功能, 并以统一的方式访问存储在不同数据源中的数据。ADO继承了 OLE DB的技术优点, 并对 OLE DB的接口作了封装, 定义了 ADO对象, 使得开发流程得以简化。

ADO中包含 Connection(连接)、Command(命令)、Recordset(记录集)、Record(记录)、Stream(流)、Parameter(参数)、Field(字段)、Property(属性)、Error(错误)等九种对象以及相应的 Errors、Parameters、Fields、Properties等四种集合^[5], 可利用 `_ConnectionPtr` `_CommandPtr`

`_RecordsetPtr`和 `_StreamPtr`等智能指针操作它们。ADO对象模型如图 2所示。

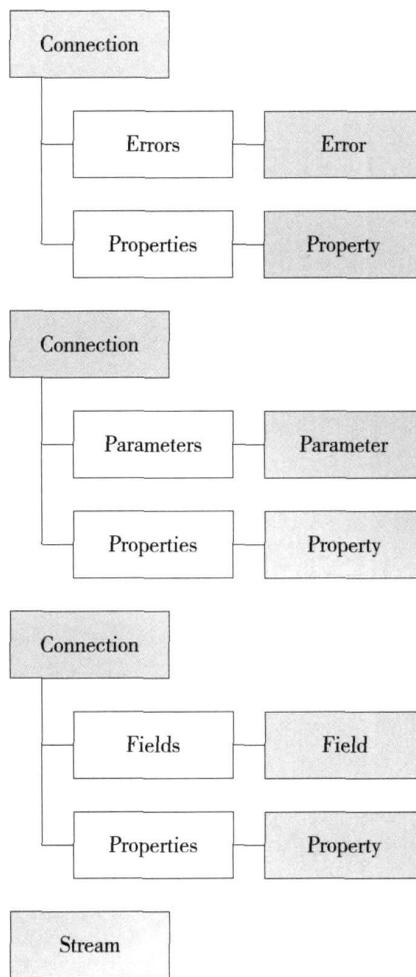


图 2 ADO对象模型

2.3 读取并输出模型文件

在 Visual C++ + 6.0中要使用 ADO接口, 首先要引入 ADO类型库并初始化 OLE/COM库环境^[5], 代码如下:

```

# import "c:\program files\common files\system\ado\msado15.dll" no_namespace rename ("EOF", "adoEOF") /引入 ADO库文件

```

```

:: CoInitialize(NULL); /初始化 COM库环境

```

接着声明 Connection对象并用 Connection对象的 Open方法连接数据库, 函数原型如下:

```

HRESULT _ConnectionPtr: Open(_bstr_t ConnectionString _bstr_t UserID, _bstr_t Password, lng Options);

```

其中, `ConnectionString`包含连接信息的字符串, `UserID`是用户名, `Password`是登录密码, `Options`是连接选项, 用于指定 `Connection`对象对数据的更新许可权。若 `UserID`和 `Password`已经在 `ConnectionString`参数中给

出,那么 User ID和 Password可以传入一个长度为 0的字符串。

然后声明一个 Recordset对象并打开记录集,代码如下:

```
_RecordsetPtr m_pRecordset //声明记录集对象
m_pRecordset.CreateInstance( _uuidof( Recordset)); //创建记录集对象实例
m_pRecordset->Open( _bstr_t( sql),
m_pConnection.GetInterfacePtr(), adOpenStatic, adLockOptimistic, adCmdText);
//执行一条 SQL语句得到记录集
```

在以前版本的 ADO (2.0 2.1和 2.1 SP2)中,一般使用 Field对象的 GetChunk和 AppendChunk方法以固定块区大小从 BLOB字段中读写 BLOB数据,这种方法比较复杂,且容易出错。使用 ADO 2.5中新引入的 Stream对象可以大大简化访问和修改 SQL Server数据库中的 BLOB数据需要编写的代码^[6]。利用 Stream对象从 image类型字段中读取并输出模型文件的步骤如下:

(1)构造一个流实例,设置流类型为二进制型,接着打开流。

(2)调用 Field对象的 GetValue()方法,返回 Model字段数据的变体对象。

(3)调用流对象的 Write()方法把返回的变体对象写入流中。

(4)把流中的数据保存到文件中。关闭流,释放所占的资源。代码如下:

```
_StreamPtr pStm; //声明流对象
pStm.CreateInstance( "ADODB.Stream");
//创建流对象实例
_variant_t varOptional( DISP_E_PARAMNOTFOUND, VT_ERROR);
pStm->PutType( adTypeBinary);
//设置类型为二进制
Long lDataSize= m_pRecordset->GetFields()->GetItem( "Model")->ActualSize
//获取 Model字段内容的大小
pStm->Open( varOptional, adModeUnknown, adOpenStreamUnspecified, _bstr_t(), _bstr_t());
//打开流对象
pStm->Write( _variant_t(m_pRecordset->GetFields()->GetItem( "Model")->Value));
//把 Model字段的内容写入流对象
pStm->SaveToFile( "c:\\stl\\m.p.stl", adSaveCreate
```

ateOverWrite); //将流数据存入文件

```
pStm->Close(); //关闭流
```

通过以上方法,读出了数据库中的一个 STL模型,并将数据输出到一个临时文件中,供进一步操作使用。读取其他模型时,将会覆盖此文件。

3 三维模型读取及显示

为了验证输出的模型文件的有效性,可以读取该文件,重建数据结构并渲染模型。

首先建立如下的数据结构用于描述一个三角面片:

```
struct TrChip
{
float n[ 3]; //三角面片法向量
float v[ 3][ 3]; //三角面片的三个顶点坐标
};
```

将组成 STL模型的一系列三角面片记录在如下的链表结构中:

```
typedefCList< TrChip TrChip> m_TRList
```

读取文件时,首先找到关键字“normal”,读取其后的法向量数据。接着跳过“outer loop”、“vertex”等关键字,读取顶点坐标数据。然后再跳过若干关键字,找到下一个三角面片的“normal”关键字,如此循环,直到读到“endsolid”关键字,结束文件读取。

本文基于 OpenGL三维图形接口渲染 STL模型,它提供了直接绘制三角形的方法。绘制一个三角片面的代码如下:

```
glBegin( GL_TRIANGLES);
glNormal3f(TrChip n[ 0], TrChip n[ 1], TrChip n[ 3]); //三角面片法矢量
for( int i= 0; i< 3; i++)
glVertex3f(TrChip v[ i][ 0], TrChip v[ i][ 1], TrChip v[ i][ 2]); //三个顶点坐标
glEnd();
```

为了便于观察,还可以利用图形操作函数如 glTranslate() (平移)、glRotate() (旋转)、glScale() (缩放)进行模型变换,或使用 glLookAt()进行视图变换,实现对模型的交互控制^[7]。另外还可以用 glPolygonMode(GLenum face, GLenum mode)函数来改变绘制方式。参数 mode取 GL_FILL,表示多边形以填充形式显示,取 GL_LINE表示以线框模式显示。最终效果如图 3所示,图中的齿轮模型显示正确,表明文件中包含了完整准确的模型信息。

4 结束语

利用数据库直接存取三维模型可解决文件方式管

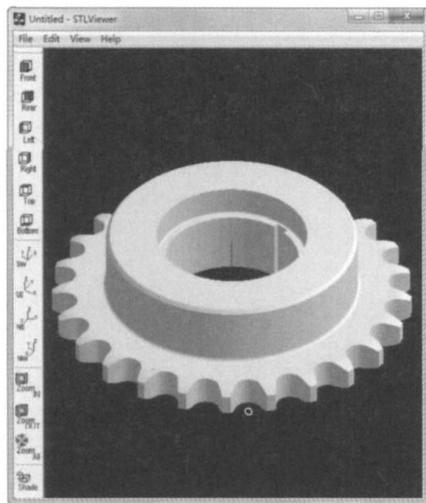


图 3 模型显示结果

理三维模型数据的弊端。本文给出了一种在 SQL Server 数据库中存取三维模型的方法, 并验证了其有效性。基于该方法可进一步设计更加完善的模型管理系统。另外, 利用相同的方法还可以在数据库中存取模型的纹理文件, 通过纹理贴图技术提高模型的真实感。

参考文献:

- [1] 褚彦军, 康凤举, 高立斌, 等. 视景仿真的三维模型库管理系统的研究与开发 [J]. 计算机仿真, 2004, 21(5): 174-176
- [2] 赵韩, 曾文莹, 黄方毅, 等. 基于 Web 的三维模型库系统的研究与开发 [J]. 合肥工业大学学报: 自然科学版, 2005, 28(11): 1384-1388.
- [3] 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.
- [4] 宫法明, 李海生, 杨钦, 等. 基于 OpenGL 的 STL 文件浏览器的设计与实现 [J]. 计算机工程与应用, 2002, 38(6): 116-117.
- [5] 李涛, 刘凯奎, 王永皎. Visual C++ + SQL Server 数据库开发与实例 [M]. 北京: 清华大学出版社, 2006
- [6] Microsoft Corporation. How To Access and Modify SQL Server BLOB Data by Using the ADO Stream Object [DB/OL]. (2004-6-15) [2009-12-23]. <http://support.microsoft.com/kb/258038/en-us>
- [7] 李保杰, 马明栋. OpenGL 实现三维可视化工作流程 [J]. 四川理工学院学报: 自然科学版, 2005, 18(3): 54-58

Research and Implementation of 3D Model Access Based on SQL Server Database

HU Feng-kai, ZHAO Gang, CHENG Xu

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

Abstract How to manage the 3D models effectively is an important problem in the 3D graphics applications. It used to store and manage the models with database and file systems, but the data completeness and security is poor. Instead, accessing models in the database can totally resolve these problems. A method is proposed to access 3D models in the database, which stores the STL file in SQL Server database as BLOB, accessing to the database via ADO, reading and exporting the model file by using Stream object. The validity of the method is verified by reading and visualising the model.

Key words SQL Server databases; 3D model; STL; ADO

(上接第 211 页)

Research on Ant Colony Algorithm for Image Edge Detection

XU Jin-long, CHEN Ming-ju, HAO Ming

(School of Automation and Electronic Information, Sichuan University of Science & Engineering, Zigong 643000, China)

Abstract The Ant Colony Optimization (ACO) to detect the image edge is researched in this paper. Different neighborhood and heuristic matrix to get the image edge are proposed in the algorithm. According to the simulation with MATLAB, how to get the optimal neighborhood and heuristic's method in this algorithm is obtained. Detecting different image edge is also received by the optimal method.

Key words neighborhood; heuristic information; Ant Colony Optimization; image edge