# 两种三维图形库的数据共享①

## Data Sharing Between Two 3D Graphic Libraries

左海龙 罗红霞 匡鸿海 (西南大学 地理科学学院 重庆 400715)

**摘 要：** 虚拟现实的构建离不开 3D 图形库的支持， OpenGL 和 Direct3D 两种主流 3D 图形库还没有实现数据共享，存在着资源浪费问题。针对上述问题，分析了 X 模型文件的文件格式和工作原理，实现了 X 模型文件在 Oracle 数据库中存储，并在 OpenGL 环境中表现。以 X 模型文件为桥梁，解决了两种图形库间的数据共享问题，为更高效的搭建 VR 环境及避免重复建模提供了技术支持，具有实际意义。

**关键词：** 虚拟现实 OpenGL Direct3D X 模型文件

## 1 引言

虚拟现实，也称“虚拟环境”,“灵境技术”或“赛伯空间”,于 1965 年,由美国科学家 Ivan Sutherland 博士在一篇名为“The Ultimate Display”的论文中提出。美国军方首先应用这项技术进行了军事仿真，但由于当时计算机硬件性能的局限，只能应用于极少部分的领域。随着计算机技术的发展，虚拟现实技术(VR——virtual reality)已脱去其神秘的面纱，逐渐走入了人们的生活[1]。城市规划、室内设计、文物保护、交通、房地产、游戏、军事、地理、教育、工业等领域都能看虚拟现实的身影。而三维图形库是实现虚拟现实的基础，无论是建模软件还是 VR 软件，都要基于 OpenGL 或 Direct3D。

目前，已有多种方法将 3DSMAX 建模软件的 3DS 和 OBJ 两种格式的文件导入 OpenGL 中,为资源共享，避免资源浪费提供了一定的支持。但关于两种图形库的数据共享及通信问题却鲜见研究成果，且 3DS 和 OBJ 文件中不包含动画、材质特性、贴图路径、动力学、粒子等信息，不益于高效的搭建虚拟场景。本文直接从 Direct3D 的 X 模型文件入手，研究并实现了利用 VB 开发工具，将 X 模型文件导入 OpenGL 环境中，解决了两种图形库通信的基础数据格式问题,为进一步研究图形库通信奠定了基础，为更高效的搭建 VR 环境及避免重复建模提供了技术支持，具有实际意义。

## 2 OPENGL图形库

OpenGL (Open Graphics Library)，是一个用于三维图形显示的 API 函数库，是程序员连接图形硬件的软件接口，它由几百个指令或函数组成。因其在三维图形方面的出色表现，OpenGL 被业界推崇为该领域的工业标准[2]。

与其它三维图形开发工具相比，OpenGL 具有更优越的跨平台能力，可以应用于 UNIX、WINDOWS 和 MACOS 几乎所有主流操作系统：同时 OpenGL 具有应用图形软件与硬件无关的特性，只要硬件支持 OpenGL 的 API 就可以运行，这使 OpenGL 具有更高的可移植性和可靠性，扩大了它的应用范围。

OpenGL 被设计为状态机(state machine)函数。许多函数的调用建立了一些参数，这些参数可在其它函数子调用中被重新调用，例如设定当前的颜色和当前的纹理贴图用于将要进行的图元光栅处理。这就使 OpenGL 应用程序可在网络上工作，在网络上，一台计算机上的程序生成了一些 OpenGL 命令，而命令的翻译工作在另一台计算机上完成[3]。

## 3 X模型文件格式

### 3.1 X 文件概述

与 OpenGL 不同，微软公司的 Direct3D 带有三维模型文件格式(X 模型文件)， 该文件由 Templates

① 收稿时间:2009-03-01

(模板)驱动运行。不同的模板装载相应的模板数据，包括网络数据，材质数据，贴图数据，网格面数据，框架数据和动画数据等。其中动画数据中包含了可以进行实时回放的动画路径数据。程序读取模板数据，并调用绘图和渲染算法，在 OpenGL 中绘制三维模型。微软公司提供了二进制和文本两种格式的 X 模型文件，本文所用的是文本形式的 X 模型文件[4]。

### 3.2 X 文件格式

X 文件由三部分组成：备注信息声明部分(文件开头第一行)，模板声明部分和模板应用部分。X 文件的基本格式如下：

```
xof0303 txt0032……备注信息声明部分
template FVFData{……模板声明部分
<b6e70a0e-8ef9-4e83-94ad-ecc8b0c04897>
template EffectInstance {
<e331f7e4-0559-4cc2-8e99-1cec1657928f>
STRING EffectFilename; [...]}……..
Frame
Box01{ ……模板应用部分
FrameTransformMatrix {
1.000000,0.000000,0.000000,0.000000,0
.000000,1.000000,0.000000,22.020401,-9.29
2849,0.000000,1.000000;;
}
Mesh Box01 {
  8;
  -16.344843;-13.936215;0.000000;,
  4.003885;-13.936215;0.000000;,
  ……..
  12;
  3;0,2,3;,
  3;3,1,0;,
……..
}
}
```

第一行是备注信息声明部分，xof 是 X 文件的标识码，0303 表示文件的版本号，txt 说明该文件为可读的文本文件，如果是 bin 则说明是一个 2 进制文件，0032 表示浮点数的精度为单精度，如果是 0064 则表明是双精度。

X 文件的模板由三部分构成：模板名称，模板 GUID 和模板用到的数据类型。模板 GUID 由于系统生成，用来标识模板名称，GUID 与模板是一一对应的。

模板应用部分是 X 文件的核心区域，其模板嵌套结构如图 1：

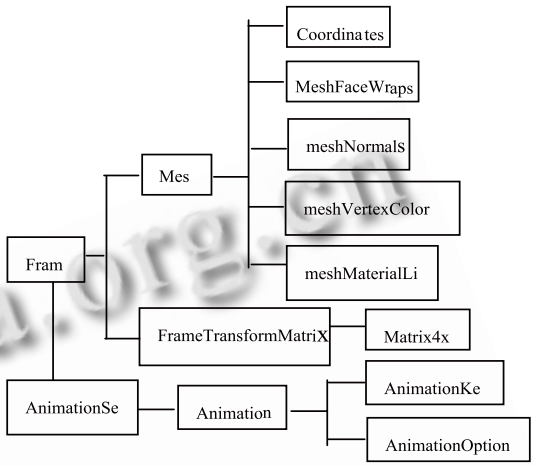


图 1　X 文件模板关系图

Frame 模板定义了一个对象框架，可以容纳任何对象，每个 Frame 包括一个 Mesh 和一个 FrameTransformMatrix 模板，当有新的 Frame 实例导入时，原有 Frame 实例将作为新实例的子对象，即 Frame 模板具有自嵌套的特性。

Mesh 模板是模板应用部分最主要的模板，它的第一个矩阵记录了模型的顶点数和顶点坐标，第二个矩阵记录了模型中所有面的个数和它们的索引顶点。

FrameTransformMatrix 模板定义了一个对当前 Frame 及其所有子对象的位置转换矩阵，这个矩阵的数据存储在一个 Matrix4x4 模板中。

AnimationSet 模板包含一个或多个 animation 模板，并且其中所有的 animation 对象在任何给定点都有相同的时间。即各个对象的时间关键点都是一样的。如何增加了 AnimationSet 的时间，也同样面增加所有对象的时间。

## 4　X文件导入OpenGL环境的方法设计

明确了 X 文件的文件结构和以模板驱动的工作原理后，文本以 VB 为例，实现了 X 文件模型在 OpenGL 环境中的显示，放大，缩小，旋转等功能，并提出了一种面片光照模型，通过对部分面片的模糊处理，提高了程序运行效率。

### 4.1 数据结构设计

三维模型的表面是由大量面片构成的，因而面片越多，模型就越真实，越细腻[5]。且面片由顶点组成，所以设计面片和顶点的数据结构如下：

```
Type face3d
    pointnumber As Integer
    indexnumber As Integer
    pointdata() As Double
    indexdata() As Integer
End Type                          /*面片结构
Type point3d
    pointID As double
    pointX As double
    pointY As double
    pointZ As double
End Type                          /*顶点结构
```

### 4.2 存储结构设计

X 文件中存储的空间数据是顶点的三维笛卡儿坐标值，以及面的点索引值，根据上文的数据结构，本文以数组为存储单位，设计存储结构如下：

```
Pointarray(N, 4) As Double
Polgonarray(M, 4) As Integer
```

X文件以二进制格式无损存储在Oracle数据库的BLOB 字段中，通过程序动态载入到 OpenGL 环境中。采用数据库保存，有益于后期海量模型文件的管理，在此不再赘述。

### 4.3 数据读取过程设计

在 X 文件中，模板名称及 GUID 具有唯一性，可作为标识，来定位模板数据的位置。首先读取文件首行，判断是否为标准 X 文件，并获取数据精度等相关信息。再向下逐行搜索模板名称，根据定义的宏来确定模板种类，调用相应的读取函数读取出数据，保存到存储数组中。读取的流程如图 2。

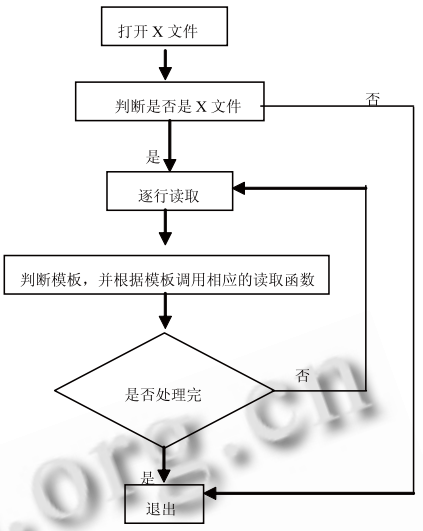


图 2 X 文件的读取流程

不同的模板所包含的数据结构是不一样的，针对不同的模板，设计不同的数据读取函数，当程序逐行读取并确定模板种类后，调用相应函数进行读取。读取过程的主程序代码如下：

```
Public function load( )
Open FileName For Input As #1     /*  打开 X 文件
m = 0                             /*  记数器
While Not EOF(1)
  Line Input #1, strData          /*按行读入数据
  m=m + 1                         /* 记数器增 1
  case “ frame ”                  /* 模板识别
      readframedata()             /*读取 frame 模板文件
      break
  case “ mesh ”                   /* 模板识别
      readmeshdata()              /* 读取 mesh 模板文件
      break
  case “ animation ”              /* 模板识别
          readanimation()         /* 读取 animation 模板文件
      break
      ……                        /*读取其它模板文件
  Wend
Close #1
```

### 4.4 模型属性设计

只有空间数据，还不足以表现出三维立体的效果，要将一个 X 模型文件表现在屏幕上，还需要对模型的颜色、亮度、灯光、相机等属性设计。

#### 4.4.1 颜色和亮度

颜色是模型的一个重要特征，模型的颜色用(r，g，b)表示，r 为红颜色的值，g 为绿颜色的值，b 为蓝颜

色的值。本文将 12 种常见颜色进行宏定义，可以任意选择模型的颜色[6]。

模型显示的另一个特性是亮度，用 alp 表示，alp∈[0,1]，亮度值为 1 表示模型的颜色最亮，为 0 表示颜色最暗。可以通过模型不同面上的亮度变化来表现立体感[7]。

4.4.2 相机

OpenGL 中用相机的概念来表示视角，即以观察者的角度将模型显示在屏幕上，通过对相机的显示范围，位置，视野等参数的设定，可以从不同角度观察模型，本文将模型的显示范围设定在[1，2000]，超出这个范围，模型将不被显示。视野设为 90，位置(40，0，120)。

4.4.3 灯光

灯光用来模拟现实中的光照，只有被光照到的面片才能被看到，且受光多的部分发亮，受光弱的部分发暗[8]。本文没有调用 OPENGL 自带的四种光照模型，而是结合实际，用镜面反射(图 3)原理，模拟了一个面片光照模型，由于 X 模型是由大量面片组成的，对每个面片进行镜面反射模型算法，其效果与漫反射大致相似，因为在现实中，漫反射就是大量镜面反射的集合。

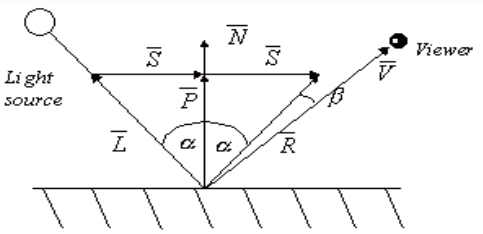


图 3　镜面反射图示

面片光照模型公式如下：

$I_{reflect} = AI\cos\alpha\cos\beta$

A 为校正因子。I reflect 为面片的亮度值。I 为入射亮度值。

对每个面进行镜图反射公式，由于在模型背面，是没有光源照射的，所以可以根据面法线与光源的夹角的大少放弃一部分面，对这部分面的亮度赋以一个较小的亮度值，从提高程序的运行效率。

**4.5 设计实现**

本文以茶壶的模型文件为例，根据上述设计方法，实现了 X 模型文件在 OpenGL 环境中的显示、放大、缩小、旋转等功能(图 4)。

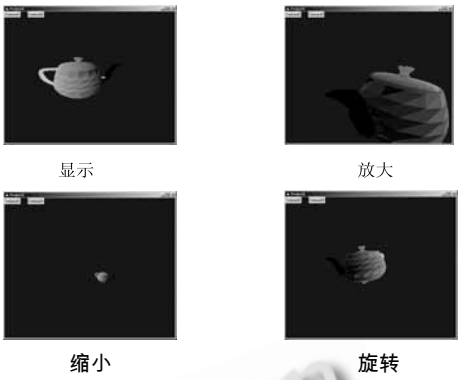


图 4　功能实现

## 5　结论

在虚拟现实的设计与建设过程中，存在重复建模的资源浪费问题，其根本原因是数据格式的不一致。本文借助 Direct3D 的 X 模型文件，解决了 X 模型文件在 OpenGL 环境下的显示与操作问题，使两种图形库有了统一的数据格式，为进一步研究两种图形库的通信奠定了基础。此外还提出了一种光照模型，该光照模型模拟了漫反射，用不同面片亮度差别来表现光照，增强模型立体感同时降低系统消耗。

本文设计的方法，一方面可以利用 3DSMAX 等建模工具导出 X 文件，在 OpenGL 中进行交互，克服了 OpenGL 建模不灵活的弊端；另一方面也为分别在两种图形库上开发的系统之间进行数据共享，更高效地建设虚拟现实环境提供了技术支持。

## 参考文献

1 胡小强.虚拟现实技术.北京:北京邮电大学出版社, 2005.1 - 2.
2 阎锋欣,侯增选,张定华,方淳,周育红,等.Open Inventor 程序设计从入门到精通.北京:清华大学出版社, 2007.21 - 23.
3 刘长松,程连冀(译).3D 图形编程指南.西安:西北工业大学, 1998.2 - 3.
4 马继东,王立海.VB 环境下的 OpenGL 的使用.森林工程, 2007,23(3):91 - 93.
5 唐荣锡,汪嘉业,彭群生,汪国昭.计算机图形学教程.北京:科学出版社, 1990.58 - 59.
6 Hill FS, JR.计算机图形学 OpenGL 版.北京:科学出版社, 2004.2 - 4.
7 Angel E. OpenGL 程序设计指南.李桂琼,张文祥,译.北京:清华大学出版社, 2005.95 - 98.
8 Rogers DF. 计算机图形学的算法基础.北京:机械工业出版社, 2002.431 - 433.