

基于 VC、Opengl 和 Directx 的三维景观浏览

张 奇¹, 屈年赦²

(¹ 辽宁省第三测绘院, 辽宁 沈阳 110034; ² 辽宁工程技术大学测量工程系, 辽宁 阜新 123000)

摘要 以 VC、Opengl 和 Directx 为基础, 创建三维景观浏览场景; 结合 3DS MAX 6.0 和 Photoshop, 三维建模、纹理映射, 处理模型纹理, 建立三维景观模型; 并将构建的精细三维景观以 3DS 模型形式读入到创建的场景中, 并最终实现在场景中任意漫游、碰撞检测、雾的探测、阴影、阳光光源位置的控制、播放音乐等诸多功能。

关键词 Opengl Directx 纹理映射 三维景观模型 3DS 模型 碰撞检测 雾的探测

中图分类号: P208 **文献标识码:** B **文章编号:** 1672-4097(2006)01-0018-03

1 建立三维景观模型

3D Studio MAX 是一套三维建模软件, 3DS MAX 的几何建模方法主要有多边形 (Polygon) 建模、非均匀有理 B 样条曲线建模 (NURBS)、细分曲面技术建模 (Subdivision Surface)。VR 系统要求实时动态逼真地模拟环境, 考虑到硬件的限制和虚拟现实系统的实时性的要求, 建立一个模型可以分别通过以上几种方法得到, 但有优劣、繁简之分。要求选取合适的建模途径, 多快好省地完成虚拟模型制作。3DS MAX 的出色的纹理贴图, 强大的贴图控制能力, 各种空间扭曲和变形, 都提供了对图像和环境映照的容易的处理途径。^[1]

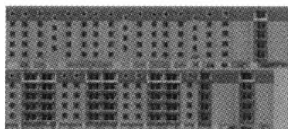


图 1 合成后的模型纹理

模型创作好后, 根据 VR 系统的编辑环境将模型输出为编辑环境所能接收的文件类型, 这里采用 3DS 格式的文件。对 3DS 模型进行修改, 包括省面, 模型 UVW 坐标调整等。

2 创建三维景观场景

2.1 Opengl 与 Directx

OpenGL 即是 Open Graphics Libraries 的缩写, 它是一套「计算机三维图形」处理函数库, 由于它是由各家显示厂商所共同定义的共通函数库, 所以它也是绘图成像的工业标准。

OpenGL 经过对 GL 的进一步发展, 实现二维和三维的高级图形技术, 在性能上表现得异常优越, 它包括建模、变换、光线处理、色彩处理、动画以

及更先进的能力, 如纹理影射、物体运动模糊等。OpenGL 的这些能力为实现逼真的三维渲染效果、建立交互的三维景观提供了优秀的软件工具。OpenGL 在硬件、窗口、操作系统方面是相互独立的。因此, OpenGL 已成为目前的三维图形开发标准, 是从事三维图形开发工作的技术人员所必须掌握的开发工具。

DirectX 是一种 Windows 系统的应用程式介面 (简称 API), 它可以以 Windows 为作业平台的游戏或多媒体程式获得更高的执行效率, 还可以加强 3D 图形成像和丰富的声音效果, 并提供设计人员一个共同的硬件驱动标准。

Microsoft DirectX 提供了一套非常好用的应用程序介面, 其中包含了设计高性能、即时应用的程式码, 为「DirectX SDK」(俗称「DirectX 开发包」)。此技术能够轻易地建构电脑游戏和多媒体的应用程序, 其中包括了 DirectDraw、DirectSound、DirectPlay、Direct3D 和 DirectInput 等部分的 API 指令及媒体相关的元件。^[2]

目前 3D 图形领域, 专业用户 (CAD/CAM 等) 3D 绘图大部分利用 OpenGL, 而微软致力于推广家庭与个人市场的多媒体娱乐平台, DirectX 几乎就是 Game、CAI 等应用程序的绘图标准接口。但是因为 OpenGL 较早完成, 且技术层次熟练深受开发者和用户喜爱, 甚至包含许多游戏开发者。但 DirectX 不只有绘图处理, 还包含 Video、Sound 等多媒体影音处理, 因此其涵盖面超过 OpenGL, 可以说是各有所长。因此, DirectX 和 OpenGL 一个都不要少。

三维地图模型可以借助三维图形库 (如 OpenGL、DirectX 等) 来进行显示。显示流程如图 2 所示^[3]:

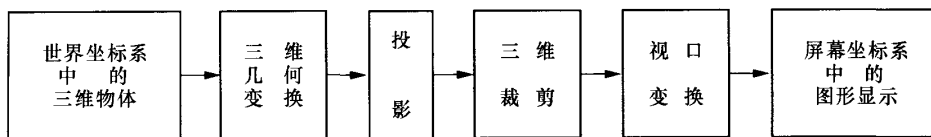


图 2 三维图形的显示流程

2.2 程序实现的构建思想

任何一种程序都有它的运行环境。OpenGL 的所有绘图命令(函数)都必需在 OpenGL 的运行环境中使用,这个运行环境称为 OpenGL 的框架。现在使用的是微软公司的 Windows 系统,所有程序又都是在 Windows 系统下运行的,所以 OpenGL 程序还得建立在 Windows 的程序框架上。在 Windows 系统下编制 OpenGL 程序必需首先建立 Windows 框架,再在 Windows 框架下建立 OpenGL 的框架。

(1) Windows 框架是在 Windows 系统下运行的程序必需的架构。Windows 框架主要由三个函数构建。

① WinMain() 程序入口。程序从这里开始运行。WinMain() 的功能为定义一个 Windows 的窗口的样式、大小等等,并建立这个窗口。

② MsgProc() 窗口回调函数。Windows 程序是基于事件响应的运行机制,MsgProc() 窗口回调函数在整个运行周期中随时对回调函数所定义的外部事件(鼠标移动、点击,窗口变化,键盘和其它函数发出的消息)做出反应。

③ GameLoop() 消息循环。这是利用 Windows 的消息机制做成的主循环函数,它被程序反复执行。它的作用是侦听消息,有消息时返回 Windows 的消息链,没有消息时就执行 OpenGL 的图形处理 Render()。

(2) OpenGL 的基本框架由四个函数组成。

① SetupPixelFormat (HDC hDC) 检测安装 OpenGL。主要设置 OpenGL 的参数,并测试显卡安装 OpenGL 的图形系统。OpenGL 安装成功后就不再使用。

② init(int Width, int Height) OpenGL 视口进行变换。根据 Windows 窗口的变化对 OpenGL 视口进行变换。这里的视口就是看到的三维世界视觉效果窗口。OpenGL 安装成功后就不再使用。

③ CleanUp() 清除 OpenGL 的连接。在程序退出时调用,清除 OpenGL 的视口和图形环境。

④ Render() OpenGL 图形处理。这是处理 OpenGL 图形的关键函数,它在程序的整个程序运行时间中都在调用,它的调用周期也就是屏幕刷新周期。

2.3 3DS 模型读入到场景中

3DS 文件是由许多块(chunk)组成的(大块中

镶嵌子块)。由于至今为止,没有一个官方的文献说明其格式,所以还有很多未知的块。不过这并不影响我们读入 3DS 文件中的模型。因为我们在读入时,可以根据自己的需要选择性地读入自己需要的块,而忽略掉那些不感兴趣或未知的块。这正是块结构带来的好处。

一个块由块信息和块数据组成。块信息又由块的 ID(两个字节长的标识,如 4D4D)和块的长度(四个字节,其实也就是下一个块的偏移字节数)组成。用 VC++ 以十六进制方式打开一 3DS 文件可以很清楚的看到其结构。在读入这种块结构(大块中嵌套小块,而块的结构固定)的文件时,完全可以用递归的方法实现,而返回上一级(子块读完,返回父块)的条件则是当前已经读入的块的字节数是否等于块的长度。从父块转向读入其子块,则可用 switch 语句实现,通过子块的 ID 判断进入哪个分支。

3DS 模型是由模型文件“*.3DS”和贴图文件“*.bmp”(或其它图形)构成。贴图文件“楼房.bmp”名是包含在“楼房.3DS”文件中的。调入并显示 3DS 模型有一套复杂的算法,鉴于篇幅的原因不在此处详细讲解它。

只要在调用它的类文件(sheji)的头文件“sheji.h”中:引用 #include “./include/3DS.H”并定义类名 C3DS Load * 3ds;在 3ds.cpp 中就可以使用“C3DS Load”类中的功能函数了。

2.4 深度检测、雾的探测

在 opengl 中,通过变量:fogMode 来保存 3 种有关雾的类型:GL_EXP, GL_EXP2, GL_LINEAR。实现代码如下:

```
glClearColor(0.5f, 0.5f, 0.5f, 1.0f); // We'll Clear To The Color Of The Fog (Modified)
glFogf ( GL_ FOG_ MODE, fogMode [ fogfilter ] ); // Fog Mode
glFogfv(GL_ FOG_ COLOR, fogColor); // Set Fog Color
glFogf(GL_ FOG_ DENSITY, 0.35f); // How Dense Will The Fog Be
glHint (GL_ FOG_ HINT, GL_ DONT_ CARE); // Fog Hint Value
glFogf(GL_ FOG_ START, 1.0f); // Fog Start Depth
glFogf(GL_ FOG_ END, 5.0f); // Fog End Depth
glEnable(GL_ FOG); // Enables GL_ FOG
第一行 glEnable(GL_ FOG) 解释起来非常容易,它告诉 OPENGL 开始进行雾的计算。
```

第二行 `glFogf(GL_FOG_MODE, fogMode[fogfilter])` 确定了雾的类型。开始的时候我们声明了数组 `fogMode`, 他保存了 `GL_EXP, GL_EXP2, GL_LINEAR`。

第三行 `glFogfv(GL_FOG_COLOR, fogcolor)` 设定了雾的颜色, 开始的时候我们把它设定为 `(0.5, 0.5, 0.5, 1.0)`, 使用变量 `FOG_COLOR` 将给予我们漂亮的灰色。

`glFogf(GL_FOG_DENSITY, 0.35f)` 确定了雾的密度, 增大这个数值雾将会变的更浓, 减小它雾将会变的更淡。

`glHint(GL_FOG_HINT, GL_DONT_CARE)` 确定了雾的渲染方式

`glFogf(GL_FOG_END, 5.0)`, 它告诉 `OPENGL` 雾能离开屏幕有多远。

2.5 场景中的碰撞检测得实现

为了创造逼真的效果, 在场景中漫游有真实的感觉, 在场景加入碰撞检测。基本思路是这样的: 给定一个障碍范围, 判断物体在这次移动后会不会进入这个范围, 如果会, 就发生碰撞, 否则不发生碰撞。碰撞检测: $\text{if } |X_a - X_b| > C$ (其中 C 为可以容纳两人通过的最小正方形区域的边长) then {不会发生碰撞, `goto`: 获取目标}。

2.6 在3D空间中播放音乐

一般来说使用 `DirectX Audio` 来播放 `WAV` 和 `MIDI` 文件, 使用 `DirectShow` 来播放 `MP3` 文件。 `DirectX Audio` 和 `DirectShow` (`DirectX Audio` 和 `DirectShow`)。另外两个 `DirectX` 组件: `DirectX Audio` 和 `DirectShow`。用 `DirectX Audio` 来播放 `WAV` 和 `MIDI` 文件; 用 `DirectShow` 来播放媒体流, 例如 `AVI` 和 `MP3`。

2.6.1 `Wav files (WAV 文件)` `WAV` 是一种未经压缩的数字音频, `CD` 音质, 但是体积非常庞大。

2.6.2 `Midi files (MIDI 文件)` `MIDI` 文件并没有保存音频记录, 它实际上更像是一套演奏指令, 所以, 它的体积是非常小的。 `MIDI` 的音质完全取决于演奏设备 (声卡), 它在高端的设备上能表现出色, 而在低端设备上则表现较差。

2.6.3 `Mp3 files (MP3 文件)` 同 `WAV` 文件一样, `MP3` 也是一种数字音频格式。不同的是, `MP3` 文件是经过压缩的, 而且是有损压缩, 这意味着它

的体积将大大地减小, 而音质上将会有一些失真 (实际上接近 `CD` 音质, 基本听不出失真)。而且, `MP3` 是一种媒体流, 这意味着在播放它的时候不会将它整个的读入, 取而代之的是分期的做部分读入。

2.6.4 本文中的选择 本文中选择 `MP3` 来作背景音乐。 `MP3` 和 `MIDI` 格式文件的体积都比较小, 它们适合做长时间的背景音乐播放。 而当需要播放简短的声效时, 例如爆炸声等, 更趋向于使用 `WAV` 文件, 品质能稍好一些。 如果文件的体积不是问题的话, 实际上可以使用 `WAV` 文件来播放所有的声音和音乐。

创建一个 `CSound` 对象, 然后根据想播放的声音类型来调用 `InitialiseForWavMidi` 或是 `InitialiseForMP3`, 二者选其一。接着, 调用 `LoadSound` 读取声音文件。最后, 调用 `Play` 模块来播放声音即可。

3 结束语

三维可视化技术是随着计算机科学的进步而迅速发展起来的一门新的计算机图形技术。本文对数字地图的三维可视化作了一些初步探讨, 着重于地图各主要要素的三维模型的建立、模型的显示以及若干应用, 尚未讨论 `LOD (Level of Detail)` 技术的应用以及图形显示的速度等有关问题。可以根据地形图为地图模型添加树木, 树木也是三维的。考虑到模拟飞行的方向是任意的, 为了让树木显的更加真实, 可以实时根据飞行的位置计算一个角度, 通过旋转这个角度让树木始终朝向观察者。

参考文献

- 1 吴瑞燕, 余强, 张国权, 等. `3ds max6 应用培训教程 [M]`. 北京: 电子工业出版社, 2004
- 2 LaMothe. `游戏开发专业人员指南系列 [M]`. 北京: 中国电力出版社, 2004
- 3 江文萍. `城市地形图及市政设施的三维可视化研究`. 武汉测绘科技大学硕士学位论文, 1999, 6~8, 22~23

Construction of Three — Dimensional Landscape Model by Using VC、Opengl and Directx

Zhang Qi¹, Qu Nianshe²

(¹The Third Surveying Institute of Liaoning Province, Shengyang 110034;

²Department of Surveying Engineering, Liaoning Technical University, Fuxin 123000, China)

Abstract In this article, the authors construct a three — dimensional landscape using VC、Opengl and Directx, constructing the three — dimensional model、projecting the texture, dealing with textures of models by using `3DS MAX 6.0` and `Photoshop`. This paper also solves how to read the three — dimensional model into landscape in the format of `3DS`, roaming about in the landscape at will, collision detecting, fog detecting, shadow detecting, controlling position of the sun、playing music and other functions.

Key words Opengl, Directx, Texture projection, Three — dimensional landscape model, 3D model, Collision detecting, Fog detecting