

硕 士 研 究 生 读 书 报 告



题目 基于OpenGL的虚拟现实技术研究

作者姓名 刘凯敏

作者学号 21651074

指导教师 李启雷

学科专业 移动互联网与游戏开发

所在学院 软件学院

提交日期 二○ 一七 年 四 月

# Virtual Reality Technology Research Based on OpenGL

Major Subject: Software Engineering

Advisor: LiQilei

By

LiuKaimin

Zhejiang University, P。R。 China

2017

摘要

近年来虚拟现实技术不断发展，在城市规划方面也具有及其广阔的应用前景，在此基础上利用VC和OpenGL开发的三维场景构建的虚拟系统VR\_City为城市规划提供了直观的视觉享受，该系统不仅可以进行场景显示，还提供场景观察和城市漫游等其他功能。文中主要介绍了虚拟场景的构建方法，并对显示效率的提高进行了探讨，最后就系统在北京某地的运用给出了实例介绍。

关键词：OpenGL、虚拟现实、城市规划

Abstract

The development of virtual reality technology in recent years, in the planning of the city also has and its broad application prospects, based on the use of VC and OpenGL 3 d scene to build virtual system for urban planning VR\_City provides intuitive visual enjoyment, the system can not only detect scene shows that also provide other functions such as city scene observation and roaming.This paper mainly introduces the method of building virtual scene, and to show the efficiency are discussed in this paper, finally the system somewhere in Beijing applying examples are given.

Keywords: OpenGL, virtual reality, and urban planning.

1 引言

OpenGL是美国高级图形和高性能计算机系统公司（SGI）所开发的三维图形库，它是图形硬件的一种软件接口，可以在许多不同的硬件平台上实现。由于OpenGL拥有大量功能强大的图形函数，可以对整个三维图形进行光色渲染，从而方便的绘制出客观世界逼真的三维景象[1]。

利用OpenGL和VC++开发的三维场景构建的虚拟系统，主要是用于显示道路及两侧模型，为城市规划提供直观的视觉感受。该系统不仅可以逼真地显示场景，同时具有视点观察功能，可以从不同的方向和角度观察建筑模型和沿途景观，还可以通过鼠标和键盘的操作来完成模型的位置调整，并且还具有沿着道路进行城市漫游的功能，另外考虑到道路上拥有大量的标志牌等通用模型，在内部添加了模型构建模块，只要给出标志牌的规格和其上的文字内容就可以在指定位置自动生成一个新的标志牌。本文主要介绍虚拟场景的构建及如何提高显示效率。

2 虚拟场景构建

2.1 模型数据导入

虚拟场景构建的第一步就是建立整个虚拟空间的模型。OpenGL虽然提供了120个图形函数用于绘制点、线和多边形等基本图元，但如果直接利用OpenGL的绘图函数建立二维实体模型是一个非常繁杂的过程，因此考虑采用其他的建模软件构建该虚拟场景中的模型。本系统采用UC-winRoad，这是Forum8 公司的一款三维虚拟现实的软件，不仅能方便的建立道路模型，而且还可以导入3DMAX所建立的模型，因此首先在UC-winRoad中建立好复杂模型后，然后再OpenGL中读出该场景数据，以实现对其控制和变换。

在软件开发时，利用建模软件UC-winRoad建立实际场景的模型，由于UC-winRoad的文件格式至今尚未公开，需要通过转换工具进行文件格式的转换，才可以读入数据。

经过转换生成的模型数据以下面的格式给出了模型所有三角形片面的数据，包括顶点坐标、法向坐标、UV坐标以及纹理序号等。

Smooth\_triangle

{

<5159.9116,553.5000,4999.6533>,<0.0000,0.0000,1.0000>,

<5159.9116,553.7000,4999.6533>,<0.0000,0.0000,1.0000>,

<5150.1636,553.5000,4999.6533>,<0.0000,0.0000,1.0000>,

uv\_vectors<-1.0000,-4.0867>,<0.0000,-4.0867>,<-1.0000,-3,1119>

Texture{texture\_29}

}

按照以上的数据格式，由于同一个顶点会被好几个三角形同时用到，必然会造成大量顶点数据的冗余，这样无论是最初读入数据还是在程序中使用数据都需要消耗大量的是时间，影响执行效率，因此程序中采用了链表的方法，提出冗余的部分并将数据格式转换为OpenGL顶点数组形式，然后利用写文件程序将处理后的数据保存为二进制文件，此文件中包含了三角形片面的顶点数据及材质信息。具体的过程如图1所示。

转换工具

数据读入程序

模型数据

三维模型

二进制文件

数据处理程序

写文件程序

图1 模型数据处理

经过处理之后生成的二进制文件可以通过读文件函数几口读入程序之中，并生成OpenGL可以控制的显示列表，进行虚拟场景的绘制。

2.2 场景绘制

数据成功导入后，就需要进行场景的绘制。利用OpenGL绘图时，必须进行初始化工作，如设置像素格式、定义光源、定义视口等[2]，同时添加窗口OnCreate、OnSize、OnPaint等事件处理程序。完成这些之后就要进行模型的绘制，根据现有的数据格式，采用了三角形的造型函数来创建对象的数据模型，然后在三维空间上放置对象，同时选择有利的观察点，接着根据纹理贴图或是颜色信息以及光照条件来计算对象的颜色，随后进行光栅化，把对象的几何数据和像素数据转换为片断，之后可能要执行混合、抖动等操作，最后将经过完整处理的片段装换为屏幕的像素。

场景绘制中所使用的纹理来自转换工具所生成的一系列图片文件，利用自己编写的程序读取图片中的数据，包括RGB值以及部分图片格式的Alpha值，利用OpenGL的纹理映射功能将这些纹理映射到模型的表面。纹理映射中的值处理可以直接用于绘制面着色，还可以通过调用OpenGL中的函数来完成混合像素等操作，从而达到所需要的效果。

2.3 场景观察

图形显示出来后，可以利用鼠标和键盘操作来完成场景的观察功能，包括绕着某个指定的建筑物进行旋转观察以及对模型位置的动态改变等。当鼠标点下伙食选中某个建筑物时，此时移动鼠标将以鼠标位置或是选中的模型为中心进行图形的旋转，而键盘中的上下键将用来改变旋转的速度。在此首先要添加鼠标和键盘事件，并添加变量来记录此时的旋转速度以及初始的旋转中心，当进行模型位置变化时，通过使用Ctrl、Shift、Alt和鼠标移动来实现模型的全方位移动。

另外该软件中还具有漫游功能，初始时为每一条道路进行了命名，在选中道路名称后，点击确定按钮来沿着道路进行漫游。

3 场景显示优化

3.1 提高显示的真实感

为了获得逼真的显示效果以及流畅的动画，需要进行优化，主要考虑到以下几个方面。

在绘制直线尤其是接近水平或垂直的直线时会出现锯齿状边缘，这种锯齿现象时由于理想的直线是由一系列的像素进行模拟的，而这些像素又必须位于它们各自的像素网格中[3]。OpenGL中提供了对点、直线和多边形进行抗锯齿的多种方法，在这里我们使用混色函数来消除锯齿，在硬件辅助消除锯齿的系统中，混色会产生类似使用累积缓冲区速度上要快得多[4]。在动态场景中，当纹理对象快速的接近或是远离观察点时，会出现闪烁现象，这是由于在这个过程中，当观察点位置改变时，模型的投影大小会发生变化，纹理对象也必须会随着一起放大或是缩小，否则就会产生令人不快的视觉效果，如闪烁或抖动等。为了避免这种情况的产生，需要先制定一系列的预先过滤的分辨率递减的纹理图像，称为Mipmap，在OpenGL中使用Mipmap，他会根据被贴图的物体大小自动确定该使用哪个纹理[3]。

最后为了获得流畅的动画效果，使用了双缓存技术。OpenGL中的动画实现是使用了重绘和交换的方法，它提供两个完整的颜色缓冲区，当一个缓冲区被显示时，另一个在后台进行绘制工作，当一个帧绘制完成后，两个缓冲区进行交换，这样就可以平滑地显示动画。

3.2 显示效率的优化

在导入外部数据后，利用OpenGL绘制场景，并对场景进行观察，这时显示效率就显得尤为重要，下面就本软件如何提高显示效率介绍一些方法。

首先采用了显示列表技术。显示列表是一组预先存储起来的OpenGL函数，以便在需要使用的地方进行调用。OpenGL函数可以存储在显示列表中也可以在立即模式下执行，但是在立即模式下执行函数操作会产生较大计算量，而显示列表中由于只储存了最后所得到的的结果矩阵，使用起来就可以提高执行速度。以旋转函数glRotate为例，在立即模式下，它需要首先生成如下的矩阵，glRotate \* ：计算量较大，而如果好几个变换命令连续执行，必然进一步增加计算量，显示列表只储存最后的旋转矩阵，甚至能将相邻的几个变换命令并置为单个矩阵相乘，大大提高了性能[1]。

其次采用了视野裁减，以观察点位置为圆心，只显示指定范围内的场景，而位于视野外的东西则被丢弃，同时考虑到天空的位置距离观察点比较远，为了在视野裁减的同时能够显示完整的天空，在程序中单独对天空定义了一个视野剪切半径。

4 结论

本文使用VC和OpenGL来完成道路场景的显示，为城市规划提供直观的依据，成功应用于北京复兴门地段道路显示。基于该技术的软件VR\_City可以形象的显示道路周围的特征建筑和景观点，还具有通用性标志设计功能，以后的工作主要是进一步完善系统的其他功能，包括空中漫游、天气模拟等，建设成集漫游、设计与一体的三维道路系统。

[参考文献]

[1] 乔林，费广正，林杜等.OpenGL程序设计[M].北京：清华大学出版社，2000.

[2]吕恒，江南.基于OpenGL和地形图支撑下的地形三维显示[J].计算机工程，2004,30(7):174-176.

[3] Dave Shreiner,Mason Woo,Jackie Neider等著.OpenGL编程指南[M].徐波等译.北京：机械工业出版社，2006.

[4] Richand S.Wright,Jr.Michael Sweet.萧湘工作室译.OpenGL超级宝典[M].北京：人民邮电出版社，2001.