

# 基于 OpenGL 技术的隧道仿真系统的研究与开发

刘鹏程<sup>1</sup>, 艾廷华<sup>1</sup>, 胡晋山<sup>1,2</sup>

(1. 武汉大学资源与环境科学学院, 武汉 430079; 2. 徐州师范大学国土信息与测绘工程系, 徐州 221116)

**摘要:** 科学计算可视化、计算机动画和虚拟现实是现代计算机图形学研究的 3 个热点, 其核心都是三维真实感图形的绘制。该文论述了隧道构造物的三维模型的建立, 探讨了 OpenGL 的建模、光照、材质、模型变换、视点变换等技术在实现隧道可视化中的应用, 并且通过具体的实例将工程隧道的三位场景形象直观地表现出来, 具有很好的应用性。

**关键词:** OpenGL 技术; 隧道建模; 光照; 模型变换; 视点变换

## Research and Development of Tunnel Simulation System Based on OpenGL

LIU Peng-cheng<sup>1</sup>, AI Ting-hua<sup>1</sup>, HU Jing-shan<sup>1,2</sup>

(1. School of Resource and Environment Science, Wuhan University, Wuhan 430079;

2. Territory Resource Information and Surveying Engineering Department, Xuzhou Normal University, Xuzhou 221116)

**【Abstract】** Scientific visualization, computer animation and virtual reality are the three focuses on study of modern computer graphics, and 3D realistic image rendering is the core of the three focuses. The detail modeling of tunnel structure attached establishment is presented and how the technique of OpenGL, including modeling, lighting, material, model transformation and view point transformation, is applied on visualization of tunnel is described. The simulation of tunnel is realized. Practical application shows that this approach has good adaptability.

**【Key words】** OpenGL; modeling of tunnels; lighting; model transformation; view point transformation

随着国民经济的发展, 公路、铁路越来越多, 地下隧道是公路、铁路不可缺少的重要部分。数字铁路、数字公路的兴起, 研究开发一种快速三维可视化仿真方法实现公路、铁路及其隧道的可视化具有相当的意义。近十年来, 计算机系统、图形输入、图形输出设备的不断发展和计算机图形软件及其生成、控制图形的算法不断优化, 无疑为引入计算机辅助设计创造了条件。Open Graphics Library(OpenGL)是 SGI 在 1983 年所开发的三维图形库——GL 的基础上发展而来的。它是一种多平台、高性能的三维图形软件开发系统, 对网络是透明的, 采用客户/服务器(client-server)体系结构, 提供很强的绘制二维和三维图形能力, 并在 OpenGL 的 API 顶部设有实用程序库支持绘制二次曲线和曲面、NURBS 曲线和曲面及若干其他高级图元。现已成为开放式的国际图形标准。本文研究的即是通过 OpenGL 来实现隧道仿真系统。

### 1 野外数据的采集及描述隧道的数据结构

野外隧道测量先必须建立控制网, 一般选用导线网, 与地面的控制网进行联测, 建立隧道控制网。在隧道控制点上架设电子全站仪, 考虑到立镜的不方便, 最好采用无反射棱镜的全站仪。一个测站可测量若干个断面, 测量的距离以通视为前提, 一般不超过 1 km。隧道断面是隧道几何建模中的重要参数, 主要有拱形、矩形、梯形、斜梯形等形态(见图 1), 测量断面的特征点, 在下文中均以拱形断面为例加以说明: 拱形断面上部为半圆形, 下部为矩形, 测量图 1 中所示的 4 点的坐标。测量时应控制其误差, 使其 4 点近似在同一竖直面上, 并使断面近似垂直于隧道地面的中心线。在测量数据的整理中, 要将数据归算到同一竖直断面中。断面间的距离依照隧道的曲折程度而变化, 对于直线隧道部分可达

200 m, 所有特征弯曲的地方均要采集。为了研究的方便, 本文只研究圆转角一种情况(如图 2)。

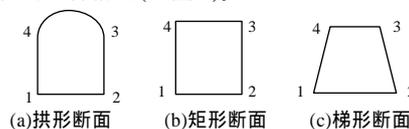


图 1 隧道断面

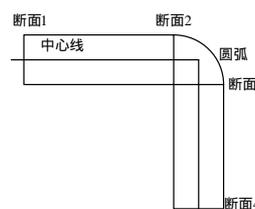


图 2 隧道平面

根据断面数据及隧道的图形特点建立相关的数据结构, 如下:

点结构:

```
struct SectorPoint
{
    double x; //平面 X 坐标
    double y; //平面 Y 坐标
    double h; //高程坐标
};
```

**作者简介:** 刘鹏程(1968 - ), 男, 工程师、博士研究生, 主研方向: GIS 可视化, 地图综合; 艾廷华, 教授、博士; 胡晋山, 讲师、博士研究生

**收稿日期:** 2007-04-23 **E-mail:** liupeng3000@tom.com

断面结构：

```
struct Sector
{
    SectorPoint P1; //断面第 1 点
    SectorPoint P2; //断面第 2 点
    SectorPoint P3; //断面第 3 点
    SectorPoint P4; //断面第 4 点
    BOOL bArcSector; //是否为圆弧段
};
将各个断面以列表类 CArray 来组织, typedef CArray<
Sector*,Sector*> CSectorArray。
```

## 2 隧道面建模

### 2.1 OpenGL 的建模技术

OpenGL 是一个硬件图形开发器的软件界面。就是说,它的应用编程界面是图形硬件的软件界面,因此它实际上是一种与硬件无关的编程界面。它由一系列 OpenGL 命令组成,程序员利用 OpenGL 命令轻松地实现图形系统大部分功能,创建动态的三维彩色图形的交互程序。OpenGL 的基本几何图元共 10 类,如图 3 所示<sup>[1]</sup>。

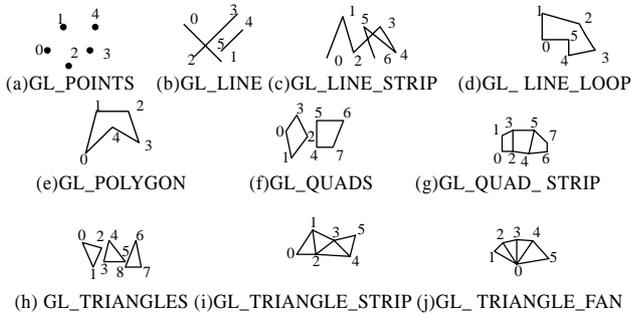


图 3 几何图元

图元以 glBegin 和 glEnd 语句为一个完整的单元。在这些简单的图元基础上可以绘制出复杂的图形,由点生成线,线生成面,面组成体,最终构造所需要的三维仿真模型。考虑光照的效果,对于面的各个定点需要用 glNormal3fv() 函数增加法线方向。

### 2.2 隧道的建模

考虑隧道各部的形状的特点,建模时本系统考虑了 4 种子模型,分别为矩形平面模型底部矩形(图 4 中 1)、两侧壁矩形(图 4 中 2);扇形平面模型(底部扇形(图 4 中 5);圆柱曲面矩形模型包括隧道交叉处的圆壁(图 4 中 4)、隧道半圆顶壁(图 4 中 3),圆柱曲面扇形模型隧道交叉处的顶壁(图 4 中 6)。

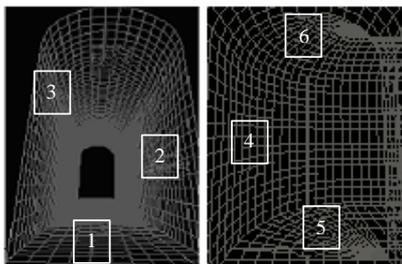


图 4 隧道建模示意图

各个子模型均以 GL\_QUAD\_STRIP 为几何单元,为了配合材质与光照,在 GL\_QUAD\_STRIP 中各个矩形单元的规格大小应近似相等,各个子模型中各点的法线方向应仔细计算,圆弧上的点的法线指向圆心,平面上的各点的法线方向相同,通过 glNormal3fv() 函数加入法线方向后,还需用 glEnable

(GL\_NORMALIZE) 函数进行单位化,否则同样的光及同样的材质所表现的颜色及光强均不相同。

下面以隧道顶(如图 5)为例,来说明程序代码建模的相关细节。

```
GLdouble ang=PI/num2; //圆弧上角度偏移量
GLdouble delD=l/num1; //纵向距离偏移量
for(i=0;i<num1;i++)
{
    glBegin(GL_QUAD_STRIP);
    for(int j=0;j<=(num2);j++)
    {
        float y = delD*(i); //第 1 个点的坐标
        float x = cos(ang*j)*w/2;
        float z = sin(ang*j)*w/2;
        GLfloat v1[] ={-x,0,-z}; //发方向指向圆心
        glNormal3fv(v1); //确定法方向
        glVertex3d(x,y,z); //加入第 1 个点
        y = delD*(i+1);
        glVertex3d(x,y,z); //第 2 个点的坐标
    }
    glEnd();
}
```

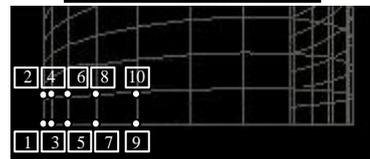
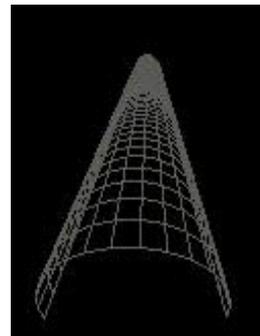


图 5 隧道顶的建模

## 3 光照与材质的应用

光照是增强物体真实感最重要的手段,对于大多数物体,只有对其加入光照效果之后,才会有鲜明的三维效果。而物体三维显示效果的不同,跟光源位置、光照颜色、物体材质的设置有很大的关系,光分为 4 种不同的成分:辐射光,环境光,漫反射光,镜面光。

材质是指物体表面对光的反射特性。一种材质由环境色、散射色、镜面高亮色等组成,分别说明了它对环境光、漫反射光和镜面反射光反射的多少,即反射率。材质影响物体的颜色、反光度和透明度等。一个只反射红光的物体在一束白光的照射下将呈现为红色,若放在纯绿色的光中观察,物体看起来将会是黑色的。

光源决定了光的颜色、强度和入射方向,而物体的几何性质及材质决定反射光的颜色、反射特性及强度等。光和物体间的相互作用远比能够模拟的情况复杂,比如在不同光照条件下模拟人的皮肤的真实颜色将会是一件十分困难的事情。在逼真显示和实时绘制之间通过以下方法实现均衡:只

在几何顶点进行光照计算对所绘制的每个像素上所执行的复杂的光照运算进行了简化处理。因而，尽管光照只是一个近似模型，但却相当实用，而且处理速度快，能够比较清晰地再现自然界中的研究对象。光和材质的颜色都采用红、绿、蓝三原色合成，如果光源的参数为 $(R_L, G_L, B_L)$ ，材质的参数为 $(R_M, G_M, B_M)$ ，若不考虑其他影响因素，则观察到的光照的参数为 $(R_L \times R_M, G_L \times G_M, B_L \times B_M)^{[2]}$ 。

OpenGL 能够构建 7 束光源，代号分别为  $GL\_LIGHT0$ ,  $GL\_LIGHT1$ ,  $GL\_LIGHT2$ ,  $GL\_LIGHT3$ ,  $GL\_LIGHT4$ ,  $GL\_LIGHT5$ ,  $GL\_LIGHT6$ ,  $GL\_LIGHT7$ ,  $glLight()$  函数用来定义光源， $glEnable()$  函数启动光源。在本隧道仿真系统中，定义了  $yh6$  束位置光源，下面的代码描述的是第一束光的情况：

```
GLfloat light_Ka[]={ 0.3f,0.3f,0.3f,1.0f }; // 环境光
GLfloat light_Ks[]={ 0.0f,0.0f,0.0f,1.0f }; // 镜面光
.....
GLfloat light3_Ka[]={ 1.0f,1.0f,0.0f,1.0f }; // 环境光
GLfloat light3_Kd[]={ 1.0f,1.0f,0.0f,1.0f }; // 漫反射光
GLfloat attenuation[]={ 1.0f,0.5f }; // 光的强度
```

下面定义  $GL\_LIGHT0$  的辐射光、环境光、漫反射光、镜面光及位置：

```
glLightfv(GL_LIGHT0, GL_AMBIENT, light0_Ka);
glLightfv(GL_LIGHT0, GL_DIFFUSE, light0_Kd);
glLightf(GL_LIGHT0, GL_CONSTANT_ATTENUATION,
attenuation[0]);
glLightf(GL_LIGHT0, GL_LINEAR_ATTENUATION,
attenuation[1]);
glLightfv(GL_LIGHT0, GL_SPECULAR, light_Ks);
glLightfv(GL_LIGHT0, GL_POSITION, new0_pos);
下面的代码定义隧道面的材质：
glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT, wall_Ka);
glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, wall_Kd);
glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, wall_Ks);
glMaterialfv(GL_FRONT_AND_BACK, GL_EMISSION, wall_Ke);
glMaterialf(GL_FRONT_AND_BACK, GL_SHININESS, wall_Se);
```

试验效果如图 6 所示。



图 6 试验效果

## 4 隧道的快速浏览

该功能的实现可通过模型变换或视点变换来实现。

### 4.1 模型变换

模型变换是在世界坐标系中进行的。缺省时，物体模型的中心定位在坐标原点处。OpenGL 在这个坐标系中，有

3 个命令，可以模型变换：

模型平移，函数为  $glTranslate\{fd\}(TYPE\ x, TYPE\ y, TYPE\ z)$ ；模型旋转，函数为  $glRotate\{fd\}(TYPE\ angle, TYPE\ x, TYPE\ y, TYPE\ z)$ ；模型缩放，函数为  $glScale\{fd\}(TYPE\ x, TYPE\ y, TYPE\ z)$ 。

### 4.2 视点变换

视点变换确定了场景中物体的视点位置和方向。执行视点变换的命令和执行模型变换的命令是相同的，在用相机拍摄物体时，可以保持物体的位置不动，而将相机移离物体，这就相当于视点变换；另外，也可以保持相机的固定位置，将物体移离相机，这就相当于模型变换。在 OpenGL 中，以逆时针旋转物体就相当于以顺时针旋转相机。因此，必须把视点变换和模型变换结合在一起考虑，而对这 2 种变换单独进行考虑是毫无意义的。

除了用模型变换命令执行视点变换之外，OpenGL 实用库还提供了  $gluLookAt()$  函数，该函数有 3 个变量，分别定义了视点的位置、相机瞄准方向的参考点以及相机的向上方向。该函数的原型为： $void\ gluLookAt(GLdouble\ eyex, double\ eyey, double\ eyez, double\ centerx, GLdouble\ centery, GLdouble\ upx, GLdouble\ upy, GLdouble\ upz)$ ，该函数定义了视点矩阵，并用该矩阵乘以当前矩阵。

本系统中便是通过这一函数实现隧道的全程浏览， $upx$ ,  $upy$ ,  $upz$  变量指定了向上向量的方向； $eyex$ ,  $eyey$ ,  $eyez$  为视点的位置； $centerx$ ,  $centery$  和  $centerz$  变量指定了参考点的位置；在软件中将  $eyex$ ,  $eyey$ ,  $eyez$ ,  $centerx$ ,  $centery$  和  $centerz$  这 6 个参数设为变量，在程序中通过设置  $:OnTimer(UINT\ nIDEvent)$  函数，每隔一定时间自动产生一个消息，在该消息中变化视点或参考点的位置，进而实现隧道的快速浏览。实现效果如图 7 所示。



图 7 隧道的全程浏览

## 5 结束语

OpenGL 三维图形库为在微机实现三维真实感图形的绘制与显示提供了有效的手段。本文所涉及的程序在系统 WindowsXP 下，用 VC++6.0 调试通过。运用 OpenGL 的建模、光照、材质、消隐等技术实现了隧道的屏幕仿真，获得了效果良好的三维真实感图形，并可进行全程浏览，使隧道工作人员可以更好地观察隧道的基本情况，从而可以更好地为工程管理和工程施工服务，具有重大的理论和现实意义。

### 参考文献

[1] 江 早, 王洪成. OpenGL VC/VB 图形编程[M]. 北京: 科学出版社, 2001.  
 [2] 乔 林, 费广正. OpenGL 程序设计[M]. 北京: 清华大学出版社, 2000.