

基于 OpenGL 的真实感图形绘制技术及应用\*

僧德文, 李仲学, 李春民, 李翠平  
( 北京科技大学 资源工程系, 北京 100083)

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的三个热点, 而这三个热点的核心都是三维真实感图形的绘制。通过 VC++ 环境下的编程实践, 系统地总结了基于 OpenGL 技术实现真实感图形绘制 (包括反走样、消隐以及光照模型等) 的基本步骤和方法, 并结合具体实例, 介绍了真实感图形绘制技术在地矿工程三维实体构模中的应用。

关键词: 真实感图形; 可视化; 消隐; 反走样; 光照模型; OpenGL

中图法分类号: TP391.41      文献标识码: A      文章编号: 1001-3695(2005)03-0173-03

OpenGL-based Realistic Image Rendering Technique and Its Application

SENG De-wen, LI Zhong-xue, LI Chun-min, LI Cui-ping  
( Dept. of Resources, Beijing University of Science & Technology, Beijing 100083, China)

**Abstract:** Scientific visualization, computer animation and virtual reality are the three focuses of modern computer graphics, and 3D realistic image rendering is the core of the three focuses. This paper summaries the experience of utilizing OpenGL to generate highly realistic images with realistic image rendering techniques, including antialiasing, culling and shading. A case study based on VC++ integrated development environment is provided to demonstrate the effectiveness and efficiency of OpenGL as an interfacing tool for realistic image rendering of mineral deposits.

**Key words:** Realistic Image; Visualization; Culling; Antialiasing; Illumination Model; OpenGL

1 引言

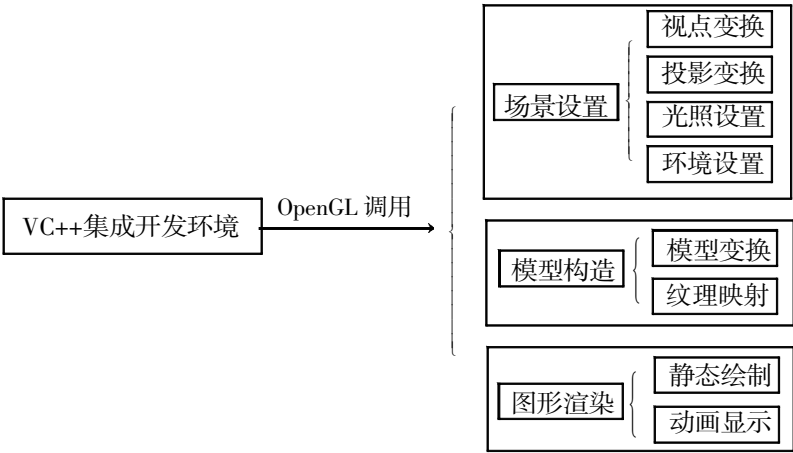
真实感图形的绘制是计算机图形学的一个重要研究领域<sup>[1]</sup>, 也是三维实体造型系统和特征造型系统的重要组成部分。三维实体在计算机显示屏上可以有三种表现形式: 简单线框图、线框消隐图和真实感图形。简单线框图能够粗略表达实体的形状, 但由于简单线框图的二义性从而导致表达的实体形状具有不确定性。线框消隐图能反映实体各表面间的相互遮挡关系, 从而可以消除简单线框图产生的二义性, 但是, 简单线框图和线框消隐图一样都只能反映实体的几何形状和实体间的相互关系, 而不能反映实体表面的特性, 如表面的颜色、材质、纹理等。只有真实感图形才能表现实体的这些特性, 因此, 在三维实体造型系统中, 生成三维实体的光照模型, 进行实体的真实感绘制与显示占有重要的地位。

OpenGL 作为一种强大的三维图形开发工具, 通过便捷的编程接口提供了处理光照和物体材质、颜色属性等通用功能<sup>[2]</sup>。本文通过 VC++ 环境下的编程实践, 系统地总结了基于 OpenGL 的真实感图形绘制技术的实现方法。

2 基于 OpenGL 的真实感图形绘制技术

用计算机在图形设备上生成连续色调的真实感三维图形

必须完成四个基本任务<sup>[3]</sup>: 用数学方法建立所需三维场景的几何描述, 并将它们输入计算机。这部分工作可由三维立体造型或曲面造型系统来完成。场景的几何描述直接影响了图形的准确和图形绘制的计算耗费, 选择合理有效的数据表示和输入手段极其重要。将三维几何描述转换为二维透视图, 可通过对场景的透视变换来完成。确定场景中的所有可见面, 这需要使用隐藏面消除算法将视域之外或被其他物体遮挡的不可见面消去。计算场景中可见面的颜色, 严格地说, 就是根据基于光学物理的光照模型计算可见面投射到观察者眼中的光亮度大小和色彩组成, 并将它转换成适合图形设备的颜色值, 从而确定投影画面上每一像素的颜色。接着通过明暗处理模型确定画面上每一个面的颜色, 最终生成图形。在 VC++ 集成开发环境下, 基于 OpenGL 技术绘制三维真实感图形的基本框架如图 1 所示。



(1) 建模。根据基本几何图元(点、线、多边形等)建立景物模型,并根据具体位置及形态对所建立的模型进行数学描述。

(2) 设置场景。把模型放置在三维空间的适当位置,并设置视点以观察感兴趣的景观。

(3) 效果处理。计算并设置模型的视觉特征,包括颜色、光照条件以及纹理映射方式等,并在场景中加入光照、雾化、融合以及反走样等处理,使场景更具有真实感。

(4) 光栅化。把模型的数学描述及其色彩信息转换为可以在计算机屏幕上显示的像素信息,实现模型的渲染。

### 2.1 反走样

反走样是真实感图形研究的重要内容。OpenGL 中对图形进行反走样,先调用 `glEnable()` 函数激活反走样(参数为 `GL_POINT`, `GL_LINE_SMOOTH` 或 `GL_POLYGON_SMOOTH`, 分别表示对点、线和多边形进行反走样),同时,调用图形显示质量控制函数 `glHint()` 提供一个图像质量提示。在 `RGBA` 颜色模式下,图形反走样还必须启动混合,最常用的混合因子是 `GL_SRC_ALPHA`(源)和 `GL_ONE_MINUS_SRC_ALPHA`(目的)。另外,也可以使目的因子为 `GL_ONE`,使线段的交点处显得更亮一些。基于 OpenGL 的直线段反走样例程如下:

```
...
glEnable( GL_LINE_SMOOTH); //激活反走样
glEnable( GL_BLEND); //启动混合
glBlendFunc( GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
//控制融合效果
glHint( GL_LINE_SMOOTH_HINT, GL_NICEST);
//控制图形绘制质量
glLineWidth( 2.0); //以一定的宽度绘制线段
... //以下进行图形绘制
```

### 2.2 消隐

消隐是增强物体立体感、消除三维物体二义性的重要手段。OpenGL 中通过 Z 缓冲区算法解决消隐问题<sup>[4]</sup>。Z 缓冲区(即深度缓冲区)存储了屏幕上每个像素的深度值,即观察点与像素之间的距离。如果当前所画的图形元素通过了深度检测,则用输入的深度值替换深度缓冲区的原有值,而所画的图形元素也在最终的图像上保留下来。在 OpenGL 中,首先打开深度检测开关,并设置深度检测方式,如下:

```
glEnable( GL_DEPTH_TEST); //启用消隐
glDepthFunc( GL_LEQUAL); //设置深度检测方式
```

这样设置了以后,即启动了深度检测,并且只有距离观察点较近的图形元素所对应的像素才保留下来,存储在深度缓存中,并最终绘制在显示屏上。

### 2.3 光照与材质模拟

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<sup>[6]</sup>。下面重点介绍 OpenGL 中对物体光照颜色以及材质属性的模拟:

(1) OpenGL 光照组成。当光照射到物体表面时,光线可能被吸收、反射和透射。被物体吸收的部分转换为热,反射、透射的光进入人的视觉系统,使我们能看见物体。在 OpenGL 光照模型中,光分为四种不同的成分:辐射光、环境光、漫反射光、镜面光。

辐射光(Emitted Light)是最简单的一种光,它直接从物体自身发出并且不受任何光源的影响。环境光(Ambient Light)是由光源发出经环境多次散射而无法确定其方向的光,它似乎来自所有的方向。一般说来,房间里的环境光成分要多些,户外的相对要少得多。当环境光照到物体表面时,它在各个方向上均匀地发散(类似于无影灯光)。漫反射光(Diffuse Light)来自于一个固定的方向,它垂直于物体时比倾斜时更明亮。然而,一旦照射到物体上,则在各个方向均匀地发散出去,因此,无论观察点处于哪个位置,它的亮度都是一样的。来自特定位置和特定方向的任何光线都可能有散射光成分。镜面光(Specular Light)也来自于某个特定的方向,它几乎全部反射到对称的方向上去。光亮的镜面和金属具有很高的镜面反射成分,而像粉笔和地毯等几乎不发生镜面反射。

OpenGL 独立计算四种成分的光强,然后再将其结果叠加到一起产生最终的光照效果。

(2) 光源的创建。OpenGL 中光源有许多特性,如颜色、位置、方向等。选择不同的特性值,则对应的光源作用在物体上的效果也不一样。定义光源特性的函数如下:

```
void glLight( if) [ v] ( GLenum light , GLenum pname, TYPE param)
其中第一个参数 light 指定所创建的光源号,OpenGL 中可以同时创建八个不同的光源,如 GL_LIGHT0, GL_LIGHT1, ..., GL_LIGHT7。第二个参数 pname 指定光源特性,最后一个参数设置相应的光源特性值。
```

在 OpenGL 中,必须明确指出光照是否有效或无效。如果光照无效,则只是简单地将当前颜色映射到当前顶点上去,不进行方向、光源、材质等复杂计算,那么显示的图形就没有真实感。要使光照有效,首先应启动光照,即

```
glEnable( GL_LIGHTING);
然后,必须使所定义的每个光源有效,即
glEnable( GL_LIGHT0);
```

其他光源类似,只是光源号不同而已。若使光照无效,则调用 `glDisable( GL_LIGHTING)` 来关闭当前光照。

(3) 材质属性。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。

(4) 材质和光的相互作用。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我们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。OpenGL 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 OpenGL 的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( $R_L * R_M, G_L * G_M, B_L$

\*  $B_M$ )。另外对于多个光源来说, 它们的各个成分满足叠加原理, 叠加之和超过 1 的颜色分量将被置为 1。用 OpenGL 绘制三维场景时, 光照及材质属性设置的示例程序如下:

```
...
// 光照属性值的设置
GLfloat ambientProperties[ ] = { 0.5f, 0.3f, 0.3f, 1.0f };
//环境光属性值设置
GLfloat diffuseProperties[ ] = { 0.8f, 0.7f, 0.8f, 1.0f };
//漫射光属性值设置
GLfloat specularProperties[ ] = { 1.0f, 1.0f, 1.0f, 1.0f };
//镜面光属性值设置
GLfloat position[ ] = { - 1.0, 1.0, 1.0, 0.0 };
//设置光源位于无穷远处
glLightfv( GL_LIGHT0, GL_AMBIENT, ambientProperties );
//环境光
glLightfv( GL_LIGHT0, GL_DIFFUSE, diffuseProperties ); //漫射光
glLightfv( GL_LIGHT0, GL_SPECULAR, specularProperties );
//镜面光
glLightfv( GL_LIGHT0, GL_POSITION, position ); //光源位置
glEnable( GL_LIGHTING ); //激活光照
glEnable( GL_LIGHT0 );
glColorMaterial( GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE );
glEnable( GL_COLOR_MATERIAL );
//物体材质的颜色跟踪当前颜色
... //以下进行物体的绘制
```

### 3 实例验证

笔者在最近开发的地矿工程三维可视化仿真系统中, 运用前面介绍的方法方便而高效地实现了地勘钻孔、三维矿体的不同显示效果。图 2 所示为进行反走样之后的钻孔效果图, 其中, 不同的颜色表示不同的采样品位值; 图 3 所示为进行隐藏面消隐以及光照处理之后的矿体真三维显示效果图, 其中, 物体材质属性设置为当前颜色的跟踪色。本系统为具有自主版权且功能较为完善的地矿三维可视化仿真系统, 现已进入实际应用中。

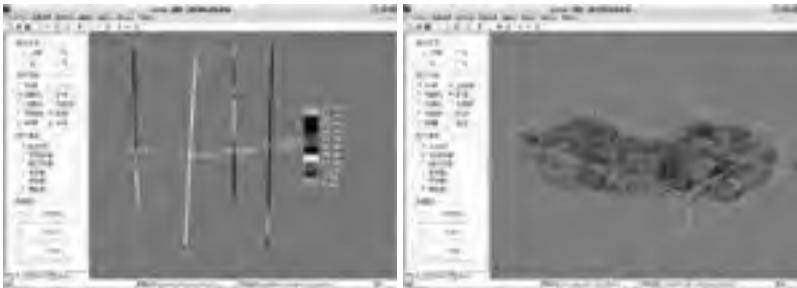


图 2 地勘钻孔效果图



图 3 矿体真三维显示效果图

### 4 技术关键

#### 4.1 环境设置

在 Windows 操作平台、VC++ 集成开发环境下, 利用 MFC 与 OpenGL 技术相结合的方法开发应用程序, 必须进行一系列初始化设置, 主要工作如下:

(1) 创建工程项目, 添加 OpenGL 程序必须的库文件 `opengl32.lib`, `glu32.lib`, `glaux.lib` 和相应的头文件 `gl.h`/`gl.h`, `gl\glu.h`, `gl\glaux.h`。

(2) OpenGL 只对 `WS_CLIPCHILDREN` | `WS_CLIPSIBLINGS` 类型的窗口有效, 必须将以下代码添加到 View 类的 `PreCreateWindows()` 函数中: `cs.style |= (WS_CLIPCHILDREN | WS_`

`CLIPSIBLINGS)`。

(3) 响应 `WM_CREATE` 消息, 获取设备描述表 (DC)、设置像素格式, 检测用户当前的显示环境, 并创建着色描述表 (RC), 并使之当前化。

(4) 响应 `WM_SIZE`, `WM_ERASEBKGD` 消息, 在 `OnSize()` 函数中建立视点、启动投影变换并建立视景物, 之后重载 `OnEraseBkgnd()` 函数, 使它返回真值, 以消除重绘画面所引起的闪烁。

(5) 为了绘制真实感图形, 在绘制所有的物体前需要设置光源, 并在 `OnDraw()` 函数中调用。

(6) 响应 `WM_DESTROY` 消息, 释放 RC 和 DC。

#### 4.2 显示列表

显示列表是 OpenGL 为了提高程序的运行效率而引入的批处理技术, 是预先存储的用于稍后执行的一组 OpenGL 指令序列<sup>[5]</sup>。激活一个显示列表后, 就按照显示列表中预先排好的次序执行其存储的指令。显示列表是不可修改的, 一旦建立, OpenGL 就将其处理成适合于图形硬件的格式, 而且可以避免在绘图过程中因主机计算量过大而影响图像生成的速度, 从而提高了程序运行的效率。

#### 4.3 双缓存技术

双缓存技术是用 OpenGL 实现动画的关键技术<sup>[5]</sup>, 其原理类似于电影放映, 它在屏幕上实现绘制图像以前, 分配了两个颜色缓存 (即前台缓存和后台缓存), 在后台缓存中执行绘制指令, 在前台缓存中进行图像显示。由于计算机的计算速度较快, 所以可以实现显示图像的连续变化。OpenGL 中的双缓存是通过调用 `SwapBuffers()` 函数来实现的。该函数的作用就是交换两个缓存, 把已经生成的图像从内存拷贝到屏幕上显示, 这样就可以方便地实现动画效果。

### 5 结论

OpenGL 三维图形库为在微机上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 Windows 2000 下, 用 Visual C++ 6.0 调试通过。研究中运用 OpenGL 技术进行真实感图形的绘制与显示, 实现了消隐、光照下三维矿体的真实感绘制以及反走样下的钻孔图绘制, 获得了效果良好的地矿三维真实感图形, 使地矿工作人员可以更好地观察和理解矿体在地下空间的贮存情况, 从而可以更好地为采矿工程设计和工程施工服务, 具有重大的理论和现实意义。

#### 参考文献:

[1] 孙家广. 计算机图形学[M]. 北京: 清华大学出版社, 1998.  
[2] 乔林, 费广正, 等. OpenGL 程序设计[M]. 北京: 清华大学出版社, 2000.  
[3] 唐荣锡, 汪嘉业, 彭群生. 计算机图形学教程[M]. 北京: 科学出版社, 2001.  
[4] 王玉华, 杨克俭, 王玲. 基于 OpenGL 的光照处理技术绘制真实感图形[J]. 现代计算机, 2002, 147(9): 72-75.  
[5] 刘敏莺, 黄文骞. 基于 OpenGL 的地形三维可视化[J]. 海洋测绘, 2002, (2): 26-29.

#### 作者简介:

僧德文(1977-), 男, 福建人, 博士, 研究方向为科学计算可视化、系统建模与仿真等。