

Visual C++ 三维图形 与动画功能库 Open GL 的使用

徐建波 (湘潭矿业学院自动化系 411201)

Open GL 在三维图形显示与动画领域已成为事实上的工业标准。现在微机公司在它的 Visual C++ V4.0 上附带了从 SGI 移植到微机上的 Open GL 三维图形库。尽管目前只能在 Windows NT 上使用(笔者在 Windows NT V3.51 上使用),但终于使广大程序员能在微机上很容易地产生三维图形与动画,并可以对目标施加消隐、加光照、纹理、反走样等高级图形处理技术。在微机上利用 Visual C++ V4.0 的 Open GL 编写出来的源程序几乎可以不加修改就能在 SGI 工作站上编译执行,很适合作三维图形与动画应用系统的前期开发工作,同时考虑到目前国内尚无作为高校计算机图形学课程合适的上机实验环境,Open GL 的推出为该课程提供了经济理想的上机实验环境。下面就笔者在使用 Open GL 当中获取的一些经验介绍给读者。

1. 运行在 Windows NT 上的 Open GL 的组成

微软公司完全按照工业标准在微机 Windows NT 上实现了三维图形软件接口 Open GL,使程序员能在微机上产生高质量的三维图形与动画,并使 SGI 工作站上的纯 Open GL 应用(不采用 X-window 与 Open GL 混合编程)能很容易地移植到微机上运行。微软的 Open GL 包括下列组件:

(1)是目前 Open GL 命令的全集。这是 Open GL 核心,包括 115 个用于 3D 图形操作的库函数,处理象物体形状描述、矩阵变换、加光加色,纹理、剪裁。位图、雾效果和反走样技术,每个函数前缀“gl”。如果考虑这些函数参数类型的不同,这 115 个函数可以生出 300 多个 OpenGL 命令。

(2)包括 43 个 Open GL 实用函数,用于处理坐标变换、多边形网络、产生球、园柱体和园碟、NURBS 曲线和曲面以及错误处理。

(3)Open GL 程序设计指导辅助库 Auxiliary Library 包括 31 个与平台无关的函数,负责窗口管理、事件处理、画经典 3D 目标、背景处理、程序运行。

(4)包括 9 个联系 Open GL 和 Windows NT 窗口系统的函数集,用于输出文字、显示列表、字型。每个函数前缀“wgl”。

(5)5 个用于象表格式和双缓冲技术的 Wins32APLs 函数,仅用于 Open GL 图形窗口。

2. Windows NT 上使用 Open GL 应注意之处

(1)一个应用程序在一个单缓冲窗口中既可以画 Open GL 图形,也可以画 GDI 图形,但不能在双缓冲窗口中混合使用这两种方式。

(2)源程序必须包含头文件(Windows.h)。产生可执行文件时,必须在连接库表中增加三个 Open GL 库文件,它们分别是:opengl32.lib、glu32.lib、glaux.lib。

(3)Open GL 提供了两种颜色方式:RGB 方式和颜色映射方式。其中在 RGB 方式中 Open GL 将每个象素用 24 位来对待,分别用 8 位(255 个值)表红、绿、蓝三成份。如果微机上配有真彩色图形卡,则用这种方式能产生十分丰富彩色效果。而颜色映射方式提供了位面中存放的值与屏幕上显示的 RGB 值之间的一种间接映射方式。如 0 对应黑色,1 对应红色,2 对应绿色,3 对应黄色,4 对应蓝色等等,这种方式对于没有足够的位面使用 RGB 方式的系统和对于闪烁及其他需要快速改变颜色的应用程序都是有用的。

(4)Open GL 提供双缓冲技术用来产生图形场景的连续运动,其效果看上去就象电影一样光滑运动。在双缓冲方式中只显示前台位面,绘图例程正常情况下只更新后台位面,然后调用 swapbuffer()函数切换两个位面,速度之快,人眼根本感觉不到任何闪烁感。

3. Windows NT 上的一个 Open GL 应用实例

在计算机飞行器仿真和游戏程序设计中,常常要为背景设计“云团”。我们不妨将“云团”看成是由许多较小的“云团”组成的,而后者又是由更小的“云团”组成。“云团”的基本形状可以用简单的球体在 XYZ 三方向作不同的缩放构成变形球(通过 XYZ 三方向作比例变换)。将变形球以随机方位偏离父球微小位移(通过 XYZ 三方向作平移变换)进行多次随机复制,最终形成了三维形状的“云团”。源程序中的 GreatNewCloud()函数产生组成“云团”的变形球坐标在 XYZ 三方向的变形系数和变形半径。函数 display()画出所有的变形球,以形成“云团”。为了增加真实感,本程序应用了 Open GL 提供的光照技术,光源的定义采用无穷光源,代表一个无穷远处的太

阳。通过调节定义材料的四个数组 `mat-ambient[]`、`mat-diffuse[]`、`mat-shininess[]` 跟随的参数,可以产生从乌云到白云的任何外观逼真的云团。为了使每次生成的“云团”都不在同一位置上,我们取 `srand()` 函数中产生伪随机数的种子参数为系统时间,这样产生的 XYZ 坐标每次都发生变化“云团”就不会出现在相同位置上。产生“云团”的源程序(略)。