

文章编号:1671-191X(2001)04-0365-04

# OpenGL 及其基于 VC++6.0 的开发

田红鹏, 马 苗

(西安科技学院 网络中心 陕西 西安 710054)

**摘 要:** 简述了计算机 3D 图形、图像与 OpenGL 的关系, 给出 OpenGL 的明确定义以及它与 DirectX 的区别。并进一步描述了 OpenGL 跨各种平台的移植性。文章从 OpenGL 的工作顺序、状态机制及构成出发, 介绍了 OpenGL 的编程原理。最后, 讨论了在 VC++6.0 中开发 OpenGL 程序的步骤。

**关键词:** OpenGL; 图形/图像; 图形显示卡; VC++6.0

**中图分类号:** TP 391.41 **文献标识码:** A

## 1 OpenGL 概述

目前, 计算机图形学已进入 3D 时代, 色彩缤纷的计算机广告、扣人心弦的 3D 游戏、震撼人心的影视特效, 无一不是三维图形的力作。总之, 3D 已无处不在。随着计算机图形、图像技术的迅猛发展, 科学可视化、计算机动画和虚拟现实成为计算机图形学的三大热门话题, 而这三大热门话题的技术核心均为 3D 图形。那么如何快速实现 3D 图形/图像? OpenGL 即可作为主要途径之一。

### 1.1 OpenGL 的概念

OpenGL 是一种与硬件、窗口系统和 OS 相独立的一系列 API。它的前身是由 SGI 公司为其图形工作站开发的 IRIS GL, 是一个工业标准 3D 计算机软件接口。后来, 为了实现向其它平台移植, 开发了 OpenGL。OpenGL 不仅有 GL 的功能, 而且是开放的, 可用于多种硬件及 OS。

### 1.2 OpenGL 与 DirectX 的区别

**相异点** DirectX 是一种 Windows 环境下标准的高性能游戏、多媒体开发工具包。它是由微软公司提供的 Windows 9X 下开发高性能图形、声音、输入、输出和网络游戏的接口, 是在 Windows 下开发游戏的基本工具软件。

DirectX 的结构由两个驱动程序构成, 硬件抽象层 HAL 和硬件模拟层 HEL, 当创建 DirectX 对象时, 同时建立一张“兼容表”, 其中记录了当前硬件系统支持的功能, 当 DirectX 需要实现某个功能时就查询该表, 得到硬件对功能的支持信息, 如果功能能够得到硬件支持, 则向 HAL 发出请示, 以得到硬件支持, 否则向 HEL 发出请求, 以模拟方式实现功能。

**相同点** DirectX 也可以利用硬件厂商提供的驱动程序接口实现直接硬件访问, 具备程序与硬件设备间相对独立等特点。

### 1.3 OpenGL 的优点

用 OpenGL 开发的应用程序可以方便地在 Unix, Windows NT, Windows 9X 等不同的平台间移植。OpenGL 也可工作于网络之上, 即采用 Client/Server 模式, 只要遵从某一个特定的协议, 客户机和服务机

可以是异构的计算机。另外,OpenGL 使用简便,功能强大,具有建模、变换、颜色模式设置、光照和材质设置、纹理映射、双缓存动画等基本功能,并提供了线框绘制、深度优先线框绘制、反走样线框绘制、平面明暗处理、光滑明暗处理、加阴影和纹理、运动模糊绘制、大气环境效果、深度域效果等丰富的绘图方式。用 OpenGL 开发的图像可以创建出接近光线跟踪的高质量静止或动画的 3D 彩色图像,而且比光线跟踪算法快一个数量级。另外,OpenGL 不但可以使用自己的数据,而且可利用其它不同格式的数据源,从而编程人员可省去编写外部设备访问函数。

## 2 Win 32 环境下运行 OpenGL 与其它软硬件间的关系

通常,计算机的硬盘、内存、显示卡的性能都对 OpenGL 的实现有重要影响。在此,仅从与 OpenGL 密切相关的显示卡来讨论图形性能。

1) OpenGL 硬件加速方式:最快的显示速度,能够极大地提高图形显示性能,获得工作站级的图形效果。一些显示芯片针对 OpenGL 进行了优化,如 Matrox 公司的 MGA G400,支持 5 种过滤效果(纹理贴图、雾化、Z 缓存等),3dfx 公司的 Voodoo 3 还提供了 Windows 98 下使用 OpenGL ICD 的驱动程序。这种情况下,由硬件实现大部分功能,少量功能由 OS 完成。

2) 3D 图形加速模式:显示速度较快。有些中低档的图形芯片也具备一定的三维加速功能,由硬件来完成一些较为复杂的图形操作。部分重要的 OpenGL 操作,如 Z 缓存等就能够直接由显示卡硬件完成;显示卡所不支持的图形功能以软件模拟的方式实现。

3) 纯软件模式:显示速度最慢,该模式是针对不具备 3D 加速功能的显示卡,又运行 OpenGL 的用户。由于所有复杂的 OpenGL 图形功能均通过主机 CPU 来执行,故速度受到很大的影响。但是由于有了软件仿真模式,才使得能在硬件性能较差的机器上对 OpenGL 进行开发。

另外,如果采用 VRAM(专门为显示卡设计的高速内存,这种双端口的存储器能让一个端口传送数据给显存,另一个端口将显存中的数据传送给 RAMDAC 同时运行,以减少图像处理的等待时间)或者 WRAM 的显示内存,会进一步提高 OpenGL 的性能。

## 3 OpenGL 的工作流程

在屏幕上显示图像的主要步骤是:

- 1) 构造几何要素(点、线、多边形、图像、位图),创建对象的数学描述;
- 2) 在三维空间上放置对象,选择有利的场景观察点;
- 3) 直接定义或通过光照条件及纹理间接给出对象的颜色;
- 4) 光栅化。

## 4 OpenGL 及相关库

OpenGL 核心函数包括 115 个函数,这些函数均以 gl 开头,这些函数是最基本的,它们可以在任何 OpenGL 的工作平台上应用。该库用于建立各种形体,产生光照效果,进行反走样及纹理映射、投影

变换等。由于这些核心函数有多种形式并能够接受不同类型的参数,所以它们派生出 300 多个函数。OpenGL 除了具有基本的 OpenGL 核心函数外,还支持其它几类函数。

1) OpenGL 实用库:包括 43 个函数,每个函数以 glu 开头。该库在核心函数库的上一层,这类函数提供了简单的调用,其实是调用核心函数。合理使用实用库函数,可减少开发者的编程工作量。

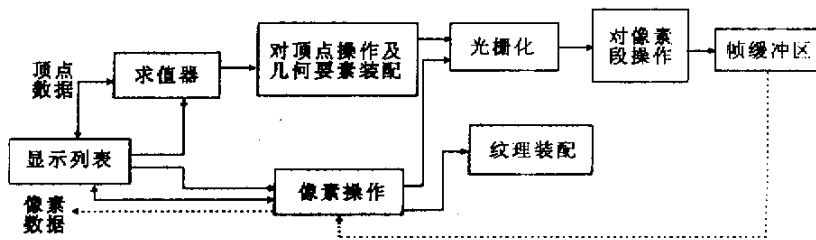


图 1 OpenGL 工作流程

Fig. 1 The working flow chart of OpenGL

2) OpenGL 辅助库 包括 31 个函数,每个函数以 aux 开头。该库可以分为 6 类:窗口初始化函数、窗口处理和事件处理函数、定义场景绘制循环函数、三维物体绘制函数、颜色索引表装入函数和空闲事件处理函数。提供了一些基本的窗口管理和基本三维图形绘制,可方便地创建完整的 OpenGL 应用程序,但对复杂的应用程序及一些深入的需求,辅助库则远远不能满足。

3) Windows 专用库函数 包括 6 个函数,每个以 wgl 开头。该库中的函数针对 Windows NT/9X 环境的 OpenGL 函数调用,用来管理显示列表、字体位图、绘图描述表及扩展函数。

4) Win 32 API 函数 包括 5 个函数,函数前面没有专门的前缀。主要用于处理像素存储格式、双缓存等函数调用。

5) 对 X - Window 系统扩展的函数库:每个函数以 glx 开头,类似于 WGL 函数,主要是针对 X - Window 系统的。

## 5 OpenGL 在 Windows 的状态机制

OpenGL 的绘图方式是由一系列状态决定的,如果设置了一种状态或模式而不改变它,OpenGL 在绘图过程中将一直保持这种模式。

前面已经提到 OpenGL 是 C/S 工作模式,即客户机向服务器发出命令时,服务器负责解释这些命令。客户机和服务器通常在同一台 PC 上运行,但两者并不要求一定在同一台机器上,这使得用户能够十分方便地在网络环境下使用 OpenGL。

OpenGL 的图形库函数封装在动态链接库 OpenGL 32.DLL 中,客户中的所有 OpenGL 函数调用都被传送到服务器上,由 WinSrv.DLL 实现功能,再将经过处理的指令发送到 Win 32 设备驱动接口(DDI),从而在计算机屏幕上产生图像。如果采用 OpenGL 图形加速卡,那么将在客户端安装 OpenGL 可装载客户模块 OpenGLICD,在服务器安装两个硬件指定模块 DDI 驱动器。

## 6 VC++ 6.0 环境下基于 OpenGL 的编程步骤

1) 创建项目文件 选择 File|New 菜单项,建立一个基于单文档(SDI)的项目文件 MyOpenGLTest,View 类基于 Cview。

2) 选择 Build|Setting 菜单项,在 Link 选项组的 Lib 列表框中添加 opengl 32.lib,glu 32.lib,若使用 OpenGL 的辅助库函数,则还需添加 glaux.lib。

3) 选择 View|ClassWizard 菜单项,打开 MFC 对话框,在 ClassName 列表框中选择 CMyOpenGLTestView 类,进行以下操作:选择 WM-CREATE 消息,单击 Edit Code,将 OpenGL 初始化代码添加到 OnCreate()函数中。

定义像素存储格式即 PIXELFORMATDESCRIPTOR

```
CCLient d( this );
```

```
int pixelFormat = ChoosePixelFormat( dc ,m_ hDC ,&pfd );
```

```
BOOL success = SetPixelFormat( dc ,m_ hDC ,pixelFormat ,&pfd );
```

```
M_ hRC = wglCreateContext( dc ,m_ hDC );
```

选择 WM\_DESTROY 消息,在 OnDestroy()中添加以下代码:

```
wglDeleteContext( m_ hDC );
```

在 MyOpenGLTestView.cpp 中的 PreCreateWindows()函数中添加代码:

```
cs.style |= WS_CLIPCHILDREN|WS_CLIPSIBLINGS;
```

OpenGL 只对 WS\_CLIPCHILDREN|WS\_CLIPSIBLINGS 类型窗口有效;

在 MyOpenGLTestView.cpp 中,将以下代码添加到 OnDraw()函数中:

```
wglMakeCurrent( pDC - m_ hDC ,m_ hRC );
```

```
DrawScene( );
```

```
WglMakeCurrent( pDC - m_ hDC ,NULL );
```

在 MyOpenGLTestView.cpp 中,添加成员函数 DrawScene( ):

```
void cMyOpenGLTestView::DrawScene( )
{
    //绘制三维场景
}
```

4) 在 MyOpenGLTestView.h 中包含以下头文件并添加类成员说明:

```
#include <gl.h>
#include <glu.h>
#include <glaux.h>
```

在 CMyOpenGLTestView 类中的 protected 段中添加成员变量声明:

```
HGLRC m_hRC;
```

同时添加成员函数声明 DrawScene( );

用户只需在 DrawScene( )函数中添加程序代码即可。

### 参考文献:

- [1] 贾志刚.精通 OpenGL[M].北京:电子工业出版社,1998.
- [2] 乔林.程序设计 OpenGL[M].北京:清华大学出版社,2000.
- [3] 白建军.OpenGL 三维图形设计与制作[M].北京:人民邮电出版社,1999.
- [4] 赵礼海.最新硬件技术及其疑难解答[M].武汉:华中理工大学出版社,2000.
- [5] 陈文英.谈 OpenGL 非矩形光栅图像的绘制[J].电脑编程技巧与维护,1998(12):64~66.
- [6] 刘瑰.OpenGL 三维图形在 Visual C++ 中的实现[J].电脑编程技巧与维护,1998(12):53~55.
- [7] 张利浩.基于 OpenGL 的三维字体显示方法[J].电脑编程技巧与维护,2000(4):75~77,85.
- [8] 毕胜.Visual C++ 6.0 下 OpenGL 屏幕保护程序的实现[J].电脑编程技巧与维护,2000(11):69~72.

## OpenGL overview and development based on VC++ 6.0

TIAN Hong-peng, MA Miao

(Xi'an University of Science & Technology, Xi'an 710054, China)

**Abstract:** The relationship between computer 3D graphics, image and OpenGL is presented, and the definition of OpenGL and difference between OpenGL and DirectX are given. Furthermore, migration of OpenGL between different systems is described and programming method is introduced from flow of work, status mechanism and architecture angles. Finally, programming steps are discussed in VC++ 6.0.

**Key words:** OpenGL; graphics/image; display card; VC++ 6.0