

精通

贾志刚 编著

OpenGL



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

精通 OpenGL

贾志刚 编著

電子工業出版社

Publishing House of Electronics Industry

内 容 简 介

OpenGL是一个工业标准的三维计算机图形软件接口,用户可以方便地利用它创建出接近光线跟踪的高质量静止或动画的三维彩色图像。

本书分为三个部分:第一部分为快速入门篇——学习使用 OpenGL,介绍使用 OpenGL 绘制具有色彩及光照的三维物体的基本内容;第二部分为提高篇——深入研究 OpenGL,介绍绘制具有特殊视觉效果的 OpenGL 图像,以及 Windows 95 OSR2 环境下的 OpenGL 实现,举例说明如何开发具有交互式功能的 OpenGL 程序;第三部分为 OpenGL 参考手册,按照英文字母顺序给出 OpenGL 核心库函数、OpenGL 应用程序库函数以及 Windows95 OSR2 的 WGL、Win32 库函数的详细说明。本书内容详实,实例丰富,给出了大量的源程序,这些程序几乎包括了 OpenGL 应用的各个方面。

本书适合具有初级计算机图形学、C 语言、FORTRAN 语言设计基础的读者学习 OpenGL 使用,也是广大工程技术人员进行 OpenGL 程序设计的详细指南。

书 名:精通 OpenGL

编 著 者:贾志刚

责任编辑:文宏武

特约编辑:张激扬

印 刷 者:北京科技印刷厂印刷

装 订 者:

出版发行:电子工业出版社出版、发行 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话:68214070

经 销:各地新华书店经销

开 本:787×1092 1/16 印张:26.5 字数:672 千字

版 次:1998 年 8 月第 1 版 1998 年 8 月第 1 次印刷

书 号:ISBN 7-5053-4910-4
TP·2402

定 价:38.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

前 言

OpenGL 的前身是由 SGI 公司为其图形工作站开发的 IRIS GL, 是一个工业标准的三维计算机图形软件接口。但是 SGI 公司向其他平台移植时, 遇到了问题。为改进其移植性, 开发了 OpenGL, 它有 GL 的功能, 而且是开放的, 适用于多种硬件平台及操作系统, 用户可以方便地利用这个图形库, 创建出接近光线跟踪的高质量静止或动画的三维彩色图像, 而且是要比光线跟踪算法快一个数量级。

1992 年 7 月, SGI 发布了 OpenGL 的 1.0 版本, 后来 SGI 与微软共同开发了 Windows NT 下的新版本。1995 年 12 月, 由 OpenGL ARB (Architecture Review Board) 批准了 OpenGL 1.1 版本, 这一版本的 OpenGL 性能得到了加强并引入了一些新功能, Microsoft 把 OpenGL 集成到 Windows NT 中, 后来又把它集成到新版本的 Windows 95(OEM Service Release 2)中, 用户既可以在 Windows 95、Windows NT 环境下开发 OpenGL 应用程序, 又可以很方便地把已有的工作站上的程序移植过来。

中国计算机应用的现状, 一方面是使用工作站的用户较少, 另一方面, 微机的硬件、性能在近几年有显著的提高, 配合 32 位操作系统的出现, 整体性能已经接近或超过了早期的工作站水平; 另外, 使用 Microsoft Visual C++、Fortran Powerstation 集成软件开发环境, 在微机上实现高品质、交互式三维图像也更加方便。所以, 本书主要是针对 Windows 95 环境下 OpenGL 应用程序, 尤其注重 Microsoft 的 OpenGL 新增加功能的介绍, 这些内容比较新颖, 也很重要。

本书分为三个部分, 第一部分为快速入门篇——学习使用 OpenGL, 第二部分为提高篇——深入研究 OpenGL, 第三部分为 OpenGL 参考手册。

第一部分 快速入门篇——学习使用 OpenGL, 介绍使用 OpenGL 绘制具有色彩及光照的三维物体的基本内容, 这些内容包括: OpenGL 辅助库函数的使用、几何要素的使用、设置物体的颜色、OpenGL 的矩阵变换、显示列表的基本概念、设置光源及材质的属性、绘制有光照的场景。

第二部分 深入研究 OpenGL——提高篇, 介绍绘制有特殊视觉效果 OpenGL 图像, 这些特殊效果包括: 获得半透明效果的混合操作、绘制反走样图形、在场景中使用雾效果、进行纹理映射, 在物体上绘制纹理、对物体的抖动操作、利用累加缓冲区产生的运动模糊、得到景深效果等。详细介绍像素段的操作及缓冲区的使用, 绘制 NURBS 曲线、曲面; 为了设计交互式的 OpenGL 程序, 将介绍 OpenGL 的选择、反馈机制, 另外, 还介绍 Windows 95 OSR2 环境下的 OpenGL 实现, 举例说明 WGL 及 Win32 库函数的使用方法, 同时给出利用 Fortran Powerstation 环境开发的 OpenGL 程序以及利用 Visual C++ 开发的具有交互式功能的 OpenGL 程序。

第三部分 OpenGL 参考手册, 按照英文字母顺序给出 OpenGL 核心库函数、OpenGL 应用程序库函数以及 Windows95 OSR2 的 WGL、Win32 库函数的详细说明。

为了能够详细、直观地说明 OpenGL 的使用, 本书给出了大量的源程序, 这些程序几乎包括了 OpenGL 应用的各个方面, 对于读者在短时间内掌握 OpenGL 是会有很大帮助的。为了方便读者使用, 上述程序收入软盘, 有需要者可与电子工业出版社发行部软件邮购科联

系购买。

参加本书编写工作的同志还有：周越、赵双勤、张砚臣、左键、田志伟、朱文羽、沈金来、郑文字、李大东、康凯，在此表示感谢。

在本书的编写过程中得到了北京航空航天大学蔡国飙博士和国家计算流体力学实验室周越女士的支持和帮助，在此表示诚挚的谢意。

作者
1998年3月

目 录

第一部分 快速入门篇——学习使用 OpenGL

第一章 OpenGL 概述	3
1.1 OpenGL 入门	3
1.1.1 什么是 OpenGL	3
1.1.2 OpenGL 的工作顺序	3
1.2 一个简单的 OpenGL 程序	5
1.2.1 一个简单的程序	5
1.2.2 OpenGL 的语法规则	8
1.2.3 OpenGL 的当前状态	9
1.3 OpenGL 程序设计的预备知识	10
1.3.1 与 OpenGL 相关的库函数	10
1.3.2 使用 aux 库	10
1.3.3 在 OpenGL 中使用颜色	17
第二章 用 OpenGL 画几何体	23
2.1 绘图前的一些准备工作	23
2.2 OpenGL 的几何要素	23
2.2.1 OpenGL 的几何要素	23
2.2.2 如何使用几何要素	30
2.3 法向矢量	37
2.4 用多边形建立曲面的技巧	38
第三章 在 OpenGL 中观察物体	40
3.1 OpenGL 基本变换命令	41
3.1.1 OpenGL 变换过程概述	41
3.1.2 OpenGL 的基本变换命令	41
3.2 取景与模型变换	44
3.2.1 模型变换	44
3.2.2 取景变换	47
3.3 投影变换	48
3.3.1 透视投影	48
3.3.2 正交投影	48
3.4 视见区变换	49
3.5 变换操作的技巧	50

3.5.1	变换操作的注意事项	50
3.5.2	操作矩阵堆栈	50
3.5.3	裁剪平面	50
3.6	应用变换的一个实例	53
第四章	显示列表	57
4.1	显示列表的基本概念	57
4.2	创建并执行一个显示列表	58
4.3	进一步使用显示列表遇到的问题	64
第五章	光照处理	69
5.1	OpenGL 光照概念	69
5.1.1	OpenGL 光照基本概念	69
5.1.2	光照处理的步骤	70
5.2	如何定义光源的特性	72
5.3	定义材料属性	81
5.4	光照处理的注意事项	92
5.4.1	OpenGL 顶点的颜色值	92
5.4.2	色彩指数模式下的光照处理	93
 第二部分 提高篇——深入研究 OpenGL		
第六章	混合、反走样和雾	99
6.1	混合	99
6.1.1	混合操作的基本方法	99
6.1.2	混合操作的实例	100
6.2	反走样	106
6.2.1	反走样点、线	106
6.2.2	反走样多边形	111
6.3	雾	114
第七章	像素、位图、字体和图像	122
7.1	位图和字体	122
7.2	图像	130
7.3	像素操作进阶	130
第八章	纹理映射	135
8.1	初步使用纹理映射	135
8.2	定义一个纹理	138
8.3	给纹理坐标赋值	141

8.4	自动生成纹理坐标	144
第九章	帧缓冲区	150
9.1	缓冲区及其用法	150
9.1.1	缓冲区	150
9.1.2	缓冲区的用法	151
9.2	对像素段的检验和操作	152
9.2.1	对像素段的检验	152
9.2.2	混合、抖动和逻辑操作	158
9.3	累加缓冲区	158
9.3.1	场景反走样	159
9.3.2	运动模糊	173
9.3.3	景深	174
第十章	求值器和 NURBS	181
10.1	求值器	181
10.1.1	一维求值器	181
10.1.2	二维求值器	185
10.1.3	在纹理坐标上使用求值器	191
10.2	GLU NURBS 接口	195
10.2.1	使用 NURBS 绘制图形	195
10.2.2	修剪	199
第十一章	选择与反馈	204
11.1	选择	204
11.1.1	使用选择机制的基本步骤	204
11.1.2	创建名称堆栈	205
11.1.3	命中记录	206
11.1.4	选择的例子	206
11.1.5	拾取	211
11.1.6	利用选择编程的提示	220
11.2	反馈	222
11.2.1	反馈数组	223
11.2.2	在反馈模式下使用标记	223
11.2.3	反馈的例子	224
第十二章	Windows NT 和 Windows 95 环境下的 OpenGL	228
12.1	Win32 环境下的 OpenGL 扩展	228
12.2	Win32 环境下的 OpenGL 专题	229
12.2.1	绘图描述表	229

12.2.2	像素格式	230
12.2.3	前后缓冲区和其它缓冲区	231
12.2.4	字体和文本	232
12.2.5	OpenGL 颜色模式和窗口调色板管理	233
12.3	Windows NT、Windows 95 环境的 OpenGL 使用	240
第十三章	Windows NT 和 Windows95 环境下的 OpenGL 程序设计	247
13.1	FORTRAN 程序中 OpenGL 函数的调用	247
13.2	OpenGL 交互式程序设计	263
13.2.1	跟踪球程序	263
13.2.2	创建菜单界面	276
13.3	使用 OpenGL 的技巧	292
 第三部分 OpenGL 参考手册		
第十四章	OpenGL 核心库函数	295
14.1	OpenGL 命令概述	295
14.2	OpenGL 核心函数	300
第十五章	OpenGL 应用程序库函数	376
15.1	OpenGL 应用程序库命令	376
15.2	OpenGL 应用程序库函数	378
第十六章	Win32 对 OpenGL 的扩展 (Windows95 OSR2)	393
16.1	Win32 函数	394
16.2	WGL 函数	399
16.3	结构	409

第一部分 快速入门篇——学习使用 OpenGL

第一章 OpenGL 概述

1.1 OpenGL 入门

1.1.1 什么是 OpenGL

OpenGL 的前身是由 SGI 公司为其图形工作站开发的 IRIS GL, 是一个工业标准的三维计算机图形软件接口。但是 SGI 公司向其他平台移植时, 遇到了问题, 为改进其移植性, 开发了 OpenGL, 它有 GL 的功能, 而且是开放的, 适用于多种硬件平台及操作系统, 用户可以方便地利用这个图形库, 创建出接近光线跟踪的高质量静止或动画的三维彩色图像, 而且是要比光线跟踪算法快一个数量级。它的主要特点是:

①OpenGL 可以在网络上工作, 即客户机/服务器型, 显示图形的计算机(客户机)可以不是运行图形程序的计算机(服务器), 客户机与服务器可以是不同类型的计算机, 只要两者服从相同的协议。

②OpenGL 是与硬件无关的软件接口, 可以在多种硬件平台上运行, 使得 OpenGL 的应用程序有较好的移植性。

1992 年 7 月, SGI 发布了 OpenGL 的 1.0 版本, 后来 SGI 与微软共同开发了 Windows NT 下的新版本。1995 年 12 月, 由 OpenGL ARB (Architecture Review Board) 批准了 OpenGL 1.1 版本, 这一版本的 OpenGL 性能得到了加强并引入了一些新功能, 其中包括: 在增强元文件中包含 OpenGL 调用, 改进打印机支持, 顶点数组的新特征, 提高顶点位置, 法线, 颜色及色彩指数, 纹理坐标, 多边形边缘标识的传输速度, 引入新的纹理特性。

Microsoft 开始把 OpenGL 集成到 Windows NT 中, 后来又把它集成到新版本的 Windows 95 OEM Service Release 2(简称为 OSR2)中, 用户既可以在 Windows 95、Windows NT 环境下开发 OpenGL 应用程序, 又可以很方便地把已有的工作站上的程序移植过来。

中国计算机应用的现状, 一方面是使用工作站的用户较少, 另一方面, 微机的硬件、性能在近几年有显著的提高, 配合 32 位操作系统的出现, 整体性能已经接近或超过了早期的工作站水平; 另外, 使用 Visual C++, Fortran Powerstation 集成软件开发环境, 在微机上实现高品质、交互式三维图像也更加方便。所以, 本书主要介绍针对 Windows 95 环境下 OpenGL 应用程序, 尤其注重 Microsoft 的 OpenGL 新增加功能及其开发应用的介绍, 这些内容比较新颖也很重要。

1.1.2 OpenGL 的工作顺序

本节介绍从定义几何要素到把像素段写入帧缓冲区的过程, 即 OpenGL 的工作顺序。

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计算对象的颜色，这些颜色可能直接定义，或由光照条件及纹理间接给出。
- ④光栅化，把对象的数学描述和颜色信息转换到屏幕的像素。另外，也可能执行消隐，以及对像素的操作，如图 1-1 所示。

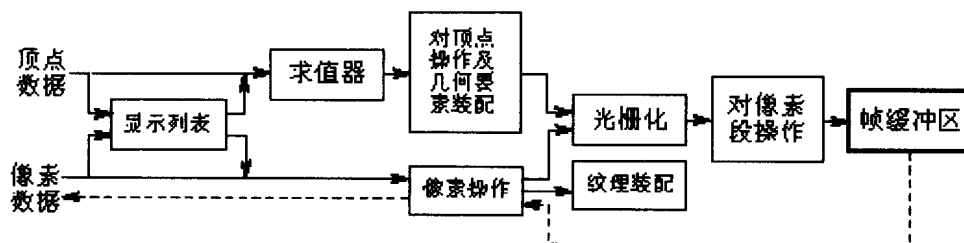


图 1-1 OpenGL 的操作顺序

(1) 几何操作

1) 针对每个顶点的操作

每个顶点的空间坐标经模型取景矩阵变换，法向矢量由逆矩阵变换，若允许纹理自动生成，则由变换后的顶点坐标生成新的纹理坐标，替代原有的纹理坐标，经过当前纹理矩阵变换，传递到几何要素装配步骤。

2) 几何要素装配

根据几何要素类型的不同，几何要素装配也不同。若使用平直明暗处理，线或多边形的所有顶点颜色相同，若使用裁剪平面，裁剪这些几何要素，此后每个顶点的空间坐标由投影矩阵变换，由标准取景平面裁剪 $x = \pm w$, $y = \pm w$, $z = \pm w$ 。若使用选择模式，没被裁剪掉的几何要素生成一个选中报表，否则投影矩阵除以 w ，做视见区和深度范围操作，若几何要素是多边形，还要做剔除检验。最后根据点图案、线宽、点尺寸等生成像素段，并给其赋上颜色、深度值。

(2) 像素操作

由主机读入的像素首先解压缩成适当的组份数目，然后数据放大、偏置并经过像素映射处理，根据数据类型限制在适当的取值范围内，最后写入纹理内存，在纹理映射中使用或光栅化成像素段。

若由帧缓冲区读入像素数据，则执行像素传输操作（放大、偏置、映射、调整），结果以适当的格式压缩并返回给处理器内存。

像素拷贝操作相当于解压缩和传输操作的组合，只是压缩和解压缩不是必需的，数据写入帧缓冲区前的传输操作只有一次。

(3) 像素段操作

若使用纹理化，每一个像素段由纹理内存产生纹素，如果还允许下面的操作，将做雾效果计算、反走样处理。其后进行裁剪处理、 α 检验(只在 RGBA 模式下使用)、模板检验、深度缓冲区检验、抖动处理，若在指数模式下，对指定的值进行逻辑操作，若在 RGBA 模式下则进行混合操作。

根据 OpenGL 所处的模式不同，由颜色屏蔽或指数屏蔽这个像素段，写入适当的帧缓

缓冲区，若写入模板或深度缓冲区，在模板和深度检验后进行屏蔽，结果写入帧缓冲区而不做混合、抖动或逻辑操作。

1.2 一个简单的 OpenGL 程序

OpenGL 是一个功能强大的图形库，用户可以很方便地开发所需要的有多种特殊视觉效果三维图形。作为简单的入门介绍，本节将结合一个简单程序，介绍与 OpenGL 相关的库函数、OpenGL 的语言规则、OpenGL 系统的状态。通过这个简单的程序，可以看到，OpenGL 程序的基本结构有两部分：初始化 OpenGL 绘图的状态和描述要绘制的物体。这样，读者可以初步掌握 OpenGL 程序设计的方法，设计功能相对简单、完整的应用程序。

1.2.1 一个简单的程序

下面给出一个绘制圆球线框图的完整程序（程序 1-1），尽管这个程序比较简单，但是具有一个完整的程序结构，如果在这个程序的基础上，加以修改，可以实现一些简单的功能，如绘制二维、三维曲线等。

程序 1-1

```
#define rad 3.14159265/180.
#include <windows.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <GL/glaux.h>
#include <math.h>

void myinit(void);
void CALLBACK myReshape(GLsizei w, GLsizei h);
void CALLBACK display(void);

/* Clear the screen. Set the current color to white.
 * Draw the wire frame sphere.
 */
void CALLBACK display (void)
{
    int i,j;
    float x,y,z,r;
    glClearColor(0.5,0.5,0.5,1.0);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 1.0, 0.0);
    glLoadIdentity (); /* clear the matrix */
```

```

glTranslatef (0.0, 0.0, -5.0); /* viewing transformation */
glRotatef(45.0,-45.0,0.0,1.0);
glScalef (1.0, 1.0, 1.0); /* modeling transformation */
for (i=0;i<180;i+=5)
{
    glBegin(GL_LINE_LOOP);
    r=2.*sin(i*rad);
    z=2.*cos(i*rad);
    for (j=0;j<360;j+=5)
    {
        x=r*cos(j*rad);
        y=r*sin(j*rad);
        glVertex3f(x,y,z);
    }
    glEnd();
}
for (j=0;j<360;j+=5)
{
    glBegin(GL_LINE_LOOP);
    for (i=0;i<=180;i+=5)
    {
        r=2.*sin(i*rad);
        z=2.*cos(i*rad);
        x=r*cos(j*rad);
        y=r*sin(j*rad);
        glVertex3f(x,y,z);
    }
    glEnd();
}
glFlush();
}

void myinit (void) {
    glShadeModel (GL_FLAT);
}
/* Called when the window is first opened and whenever
 * the window is reconfigured (moved or resized).
 */
void CALLBACK myReshape(GLsizei w, GLsizei h)
{

```

```

    glMatrixMode (GL_PROJECTION); /* prepare for and then */
    glLoadIdentity (); /* define the projection */
    glFrustum (-1.0, 1.0, -1.0, 1.0, 1.5, 20.0); /* transformation */
    glMatrixMode (GL_MODELVIEW); /* back to modelview matrix */
    glViewport (0, 0, w, h); /* define the viewport */
}
/* Main Loop
 * Open window with initial window size, title bar,
 * RGBA display mode, and handle input events.
 */
int main(int argc, char** argv)
{
    auxInitDisplayMode (AUX_SINGLE | AUX_RGB | AUX_DEPTH);
    auxInitPosition (0, 0, 400, 400);
    auxInitWindow ("wireframe of a sphere");
    myinit ();
    auxReshapeFunc (myReshape);
    auxMainLoop(display);
    return(0);
}

```

其中主程序的功能是分别用三个前缀为“aux”的函数设置 OpenGL 窗口的显示模式，打开窗口的左下角坐标为 (0, 0)，右上角坐标为 (400, 400)，auxInitWindow 函数给出窗口的标题，myinit 子程序设定 OpenGL 的明暗处理方式，auxReshapeFunc 函数在窗口移动、变形后，调用 myReshape 子程序重新计算模型、取景变换及投影变换，auxMainLoop 循环调用 display 子程序，设定颜色、投影变换、描述几何要素，display 子程序首先设置 OpenGL 窗口的背景及绘图颜色，然后对物体、场景进行旋转、缩放、平移操作。有关函数的使用请参见本书的第三章。glBegin 与 glEnd 之间的语句用于给定的顶点绘制圆球的经线和纬线，其中 x, y, z 为顶点坐标，由 glVertex 函数定义所描述几何要素的顶点，在本书第 2.2 节将介绍几何要素的使用，上述前缀为“aux”的函数是 OpenGL 的辅助库函数，有关函数的使用及说明请参见本书的第 1.3.2 节。图 1-2 是这个程序运行的结果，由此不难看出，在 OpenGL 中绘制三维物体与绘制二维物体复杂程度基本相当，OpenGL 的图示显示功能是强大的。实现这些功能也很方便。另外，这个例子显示的圆球只是一个线框图，并不是三维实体，如果不对其做适当的旋转变换，看不出来是三维物体，即使在一个特定角度上进行观察，又因为没做消除，立体感也不强，即需要解决的问题是做消隐操作和构造曲面，这些内容将在后面的章节详细介绍。值得注意的是，与 OpenGL 辅助函数库中的 auxWireSphere 函数相似，而另一个函数 auxSolidSphere 可以绘制一个实体圆球。除此之外，辅助库中还有很多绘制其他三维物体线框图、实体图的子程序可供调用，具体内容请参见下一节。

运行 OpenGL 的软件及硬件需要是：

操作系统：Windows 95 (OEM Service Release 2) 或 Window NT 4.0 版本；

软件开发环境：Microsoft Visual C++ 4.0 及以上版本；

硬件：奔腾级微机，最好配有支持 OpenGL 硬加速的图示卡。

windows.h 支持微软 Windows 95 或 Windows NT 环境下使用 OpenGL 的头文件，gl.h 是 OpenGL 核心函数的头文件，glaux.h 是辅助库函数的头文件，对于所有的 OpenGL 应用程序，这 3 个头文件是必需的。在创建执行文件时，要另外连接 opengl32.lib, glu32.lib, glaux.lib 3 个函数库。运行已创建的执行文件时，在 windows95\system 目录下要有 opengl32.dll, glu32.dll 两个动态连接库。

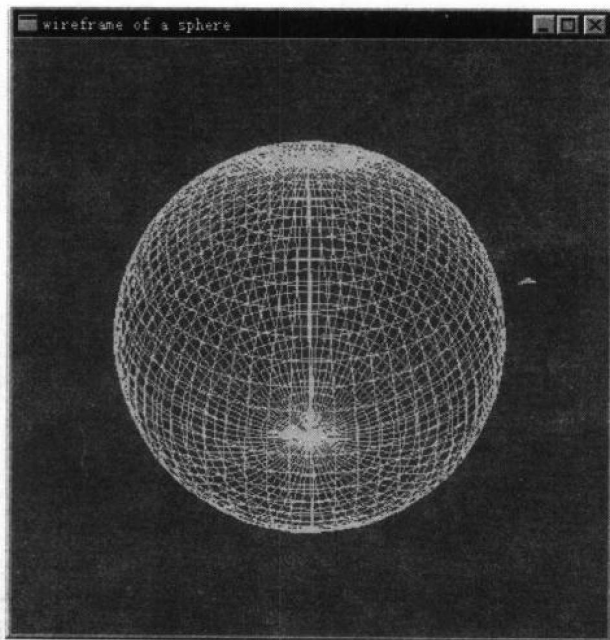


图 1-2 绘制一个线框图

1.2.2 OpenGL 的语法规则

由上一节的程序可以看出，OpenGL 核心函数命令的前缀为“gl”，组成命令的首字母大写，如 glColor3f，其常量是以 GL_ 开头，均用大写字母，用下划线把关键词分开，如 GL_LINE_LOOP。有些函数如 glVertex3f，为定义几何要素顶点的函数，“3”表示有 3 个参量，“f”说明类型为浮点，由于顶点的参量数不同(2 个或 3 个)，数据类型不同(整数、浮点或双精度)，参见表 1-1。

表 1-1 OpenGL 命令后缀及参量数据类型

后缀的数据类型	C 语言类型	OpenGL 类型定义
b 8 位整数	signed char	GLbyte
s 16 位整数	short	GLshort
l 32 位整数	long	GLint, GLsizei
f 32 位浮点数	float	GLfloat, GLclampf
d 64 位浮点数	Double	GLdouble, GLclampd
ub 8 位整数	unsigned char	GLubyte, GLboolean
us 16 位整数	unsigned short	GLushort
ui 32 位整数	unsigned long	GLuint, GLenum, GLbitfield

glVertex 函数有多种形式，适用于具体的情况，有时会出现在上述命令后加上“v”的形式，如 glVertex3fv，其后缀“v”表示，其参量是一个矢量或矩阵的指针，对于 glVertex3fv，