

## <<OpenGL编程技术详解>>

### 图书基本信息

书名：<<OpenGL编程技术详解>>

13位ISBN编号：9787122090805

10位ISBN编号：7122090809

出版时间：2010-8

出版时间：化学工业出版社

作者：和克智

页数：419

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<OpenGL编程技术详解>>

### 内容概要

OpenGL图形系统是一个连接图形硬件的软件接口。

利用该系统，程序员可以极为方便地开发出交互式程序，生成三维动态对象的彩色图像。

使用OpenGL可以利用计算机图形学理论和技术来生成具有真实感的图形，或构建出现实世界中并不存在的图形图像。

本书通过大量的实例，深入浅出、从简到繁地介绍了OpenGL编程技术。

通过本书的学习，读者可以掌握利用OpenGL API编写三维图形及三维动画应用程序的方法。

本书是作者积多年教学、科研的成果，在认真研读了OpenGL官方资料并分析了数百个OpenGL应用程序源代码的前提下精心撰写而成。

书中利用60多个实例详细介绍了OpenGL各类应用程序的编写方法，同时对大量OpenGL命令进行了详细的介绍，尤其是对GLUT库所提供的命令的介绍，是在其他资料中难以见到的。

本书中所有的实例程序均是作者精心设计并认真调试，在VC6环境下开发成功的，具有原创性，从而可供为OpenGL程序员直接引用和参考。

本书涉及面较广、内容翔实、实用性强，可作为广大计算机工作者的参考资料，亦可作为大学计算机专业或他需要使用OpenGL技术的专业本科生或研究生的教材。

## 书籍目录

第1章 OpenGL详解	1
1.1 OpenGL的发展历史	1
1.2 OpenGL规范	2
1.2.1 数据类型	2
1.2.2 命令	3
1.3 OpenGL执行模型	4
1.4 帧缓冲区	4
1.5 状态	6
1.6 OpenGL实用工具包	6
1.7 OpenGL管线	7
1.7.1 绘制几何对象	8
1.7.2 绘制图像	12
1.8 坐标变换	13
1.9 纹理	16
第2章 图元绘制	20
2.1 点和线段的绘制	20
2.1.1 程序入口	22
2.1.2 回调函数OnDisplay	23
2.1.3 回调函数OnReshape	25
2.1.4 点的大小与线的宽度	27
2.1.5 图形对象的颜色	28
2.2 折线的绘制	29
2.2.1 使用双缓冲区	31
2.2.2 改变背景色	31
2.2.3 设置视区坐标系	31
2.2.4 绘制折线	32
2.3 三角形的绘制	32
2.3.1 设置白色背景	35
2.3.2 绘制简单三角形	35
2.3.3 绘制连续三角形	35
2.3.4 绘制三角扇形	36
2.4 四边形的绘制	37
2.4.1 绘制简单四边形	38
2.4.2 绘制连续四边形	38
2.5 多边形的绘制	38
2.6 轮廓与填充	39
2.6.1 使用菜单	43
2.6.2 多边形光栅模式	45
2.6.3 隐藏轮廓线	47
2.7 圆的绘制	47
2.7.1 利用绘制正多边形函数绘制圆	47
2.7.2 绘制圆	51
第3章 三维绘图	52
3.1 绘制三维点	52
3.1.1 旋转变换	54
3.1.2 矩阵堆栈	54
3.1.3 三维投影变换	55
3.1.4 绘图	55
3.2 利用硬件信息改变点的大小	56
3.3 利用平面图元构造三维实体	58
3.3.1 设置光照	63
3.3.2 设置三维对象的材质	64
3.3.3 法向量及其归一化	65
3.3.4 绘制三维实体	66
3.3.5 深度检测	66
3.3.6 实体图与线框图	67
3.4 GLUT提供的实体对象	68
3.4.1 三维实体	73
3.4.2 重新归一化法向量	78
3.4.3 设置视点	78
3.4.4 高光反射	78
第4章 观察与几何变换	80
4.1 变换观察点	80
4.2 几何变换	84
4.2.1 平移变换	85
4.2.2 旋转变换	87
4.2.3 缩放变换	90
4.3 用户自定义几何变换	93
4.3.1 用户自定义变换矩阵	94
4.3.2 使用用户定义的矩阵	95
4.4 投影变换	96
4.4.1 正交投影	96
4.4.2 透视投影	99
第5章 颜色、光照和材质	104
5.1 OpenGL中的颜色	104
5.1.1 计算机的颜色模式	104
5.1.2 颜色的使用	105
5.1.3 着色模式	111
5.2 OpenGL中的光照	113
5.2.1 光源类型	113
5.2.2 场景中的光源	114
5.3 OpenGL中的材质	118
5.3.1 材质与光照的关系	118
5.3.2 表现材质的方法	120
5.3.3 表面法线与正反面	126
第6章 颜色的高级应用	132
6.1 混合与透明	132
6.1.1 颜色混合	132
6.1.2 透明	136
6.2 反走样	138
6.3 雾化	142
第7章 图像及图像处理	149
7.1 位图图像	149
7.1.1 定义位图图像	151
7.1.2 显示位图图像	152
7.2 字体	153
7.2.1 定义字体	156
7.2.2 显示列表	156
7.2.3 构造光栅字体	159
7.2.4 显示光栅字体	160
7.3 像素图像	161
7.3.1 载入图像	163
7.3.2 显示图像	163
7.4 对像素图像的一些操作	165
7.4.1 缩放图像	169
7.4.2 反射图像	169
7.4.3 颜色通道分离	170
7.4.4 灰度图变换	172
7.4.5 绘制负像	173
第8章 纹理映射	176
8.1 概述	176
8.1.1 纹理映射的一般步骤	176
8.1.2 纹理映射实例	177
8.2 纹理环境	182
8.2.1 纹理函数	188
8.2.2 “替换”映射	189
8.2.3 “调节”映射	189
8.2.4 “贴纸”映射	189
8.2.5 “混合”映射	190
8.3 纹理对象	190
8.3.1 绑定纹理	194
8.3.2 映射	195
8.4 透明纹理	196
8.5 一维纹理	201
8.5.1 创建一维纹理	205
8.5.2 一维映射	205
第9章 纹理映射的特殊应用	208
9.1 利用一维纹理制造光影效果	208
9.1.1 生成并载入一维纹理	216
9.1.2 绘制圆环	216
9.1.3 显示及动画	217
9.1.4 辅助函数	217
9.2 绘制自发光体	218
9.2.1 空纹理图像	225
9.2.2 绘制螺旋管	225
9.2.3 绘制光芒	225
9.2.4 动态显示	227
9.3 各向异性过滤和常规过滤	227
9.3.1 各向异性过滤	233
9.3.2 常规过滤	234
9.4 利用立方体映射实现球体反射	235
9.4.1 “帧”类	249
9.4.2 立方体映射	250
9.4.3 绘制背景和球体	250
9.4.4 移动观察	251
9.5 多重纹理	251
9.5.1 使用多重纹理	261
9.5.2 在Win32下使用多重纹理	261
第10章 曲线与曲面	263
10.1 二次曲面	263
10.1.1 创建二次曲面对象	266
10.1.2 设置二次曲面绘制参数	266
10.1.3 绘制二次曲面对象	267
10.1.4 以不同形式绘制二次曲面	268
10.2 利用变形生成对象	270
10.3 样条	273
10.3.1 Bezier曲线与曲面	273
10.3.2 B样条曲线与曲面	280
10.4 修剪NURBS曲面	290
10.4.1 修剪曲线	293
10.4.2 曲面修剪	293
10.5 曲面纹理映射	294
10.5.1 三维点结构及Bezier混合函数计算	298
10.5.2 Bezier曲面面片结构	299
10.5.3 绘制带有纹理的Bezier曲面	299
第11章 选择和反馈	301
11.1 选择	301
11.1.1 创建名字堆栈	305
11.1.2 设置选择缓冲区	306
11.1.3 拾取及其处理	306
11.1.4 层次选择	308
11.2 反馈	312
11.2.1 建立反馈缓冲区	318
11.2.2 反馈数据	318
11.2.3 反馈过程及其处理	319
第12章 Windows下的OpenGL编程	320
12.1 利用MFC进行编程	320
12.1.1 对View类进行增补	327
12.1.2 初始化	328
12.1.3 善后处理	330
12.1.4 窗口尺寸的改变	331
12.1.5 绘图	331
12.1.6 其他修改	331
12.2 利用Windows API进行编程	332
12.2.1 生成应用程序窗口	340
12.2.2 定义窗口过程	342
12.2.3 绘制场景	343
12.2.4 处理键盘消息	344
第13章 案例集锦(一)	346
13.1 网格曲面及其纹理映射	346

13.1.1 初始化 349 13.1.2 绘图 350 13.2 纹理映射变换 351 13.2.1 加载纹理图像 356 13.2.2 变换纹理映射 357 13.2.3 绘图 357 13.3 使用模板缓冲区 359 13.3.1 启用模板缓冲区 361 13.3.2 设置模板测试参数及行为 362 13.3.3 绘图 363 13.4 多窗口界面及交互 364 13.4.1 创建、显示子窗口 381 13.4.2 文字输出 383 13.4.3 菜单及其处理 384 13.4.4 绘图 386 13.4.5 修改绘图参数 387 第14章 案例集锦(二) 388 14.1 利用抖动进行反走样 388 14.1.1 两个辅助函数 392 14.1.2 抖动 393 14.1.3 绘图 394 14.2 焦点处理 395 14.3 运动模糊 398 14.3.1 设置灯光与背景色 399 14.3.2 绘图 399 14.4 阴影 400 14.4.1 数据结构 410 14.4.2 使用数据文件 410 14.4.3 处理对象数据 413 14.4.4 绘制对象 413 14.4.5 动画 413 附录 OpenGL命令 415 A.1 OpenGL核心命令 415 A.2 实用库命令 416 A.3 辅助库命令 417 A.4 实用工具包命令 417 A.5 Windows专用命令 417 参考文献 419

## 章节摘录

大多数图形硬件都支持一个前台缓冲区和一个后台缓冲区，以便执行“双重缓存（DoubleBuffering）”。

这允许应用程序在显示（可见的）前台缓冲区的同时渲染到（屏幕外的）后台缓冲区，当渲染完成后，将交换两个缓冲区，以便使完成后的渲染显示为前台缓冲区，同时使渲染工作可以在后台缓冲区中重新开始。

在使用双重缓存时，最终用户永远不会看到正在进行绘制的图形，而只会看到完成后的图形，这种技术常被用来实现动画技术。

通过对左眼使用一个颜色缓冲区，对右眼使用另一个颜色缓冲区，就可以实现立体视图。

双重缓存则是通过使用一个前台缓冲区和一个后台缓冲区来支持的，因此，双重缓存的立体窗口就有4个颜色缓冲区：前左、前右、后左和后右。

普通的（非立体）双重缓存窗口有一个前台缓冲区和一个后台缓冲区，单独的缓存窗口只有一个前台缓冲区。

若要绘制不显示隐藏表面的3D对象，则会涉及到深度缓冲区。

该缓冲区中存储了所显示对象各个像素的深度值。

绘制额外的对象时，为了确定新对象的可见性，可以在各个像素间执行深度比较。

模板缓冲区用来执行复杂的遮挡操作。

模板缓冲区中可能存储一个复杂的形状，后续的绘制操作可以使用模板缓冲区的内存来确定是否需要更新各个像素的显示。

累积缓冲区是一个颜色缓冲区，通常具有比颜色缓冲区精度更高的组件。

这就允许几幅图像累积在一起，组成一个复合图像。

它的一个用途是将几帧运动的对象绘制到缓冲区中。

将累积缓冲区中的各个像素除以帧数，结果就是一幅显示出运动模糊效果的最终图像。

类似的技术可以用来模拟景深效果和执行高质量的全屏幕反走样操作。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>