

高等院校计算机专业教育改革推荐教材

计算机图形学 实验教程

(OpenGL 版)

李胜睿 等编著



附光盘

ISBN 7-111-15278-6/TP · 3977

◎ 策划 胡毓坚

◎ 封面设计 旭洲企划 刘吉维

高等院校计算机专业教育改革推荐教材

基础知识模块

计算机电路与电子技术基础
信号、系统与数字信号处理
计算机科学中的离散结构
计算机基础教程
数据结构与算法
操作系统简明教程
编译方法
微型计算机硬件技术教程——原理·汇编·接口及体系结构
计算机网络技术
计算机专业英语

程序设计模块

Windows 编程技术
新编 C 语言程序设计
Visual Basic 6.0 程序设计
面向对象程序设计与 Java 语言
面向对象程序设计基础
计算机网络程序设计
分布式对象技术

应用技术模块

数据库应用技术基础
多媒体数据库技术
计算机图形学
多媒体技术及其应用
互联网应用与技术
人机交互教程
人工智能技术及其应用

软件工程模块

软件工程方法与实践
ROSE 对象建模方法与技术
统一建模语言 UML 导论

实践模块

计算机网络实验教程
新编 C 语言学习指导与习题
计算机图形学实验教程 (OpenGL 版)
微型计算机硬件技术实验教程
信号、系统与数字信号处理学习指导与实践
编译方法学习指导与实践

ISBN 7-111-15278-6



9 787111 152781 >

地址: 北京市百万庄大街22号

联系电话: (010) 68326294

邮政编码: 100037

网址: <http://www.cmpbook.com>

E-mail: online@cmpbook.com

定价: 27.00 元 (含 1CD)

高等院校计算机专业教育改革推荐教材

计算机图形学实验教程

(OpenGL 版)

李胜睿 王乘恩 王智高

编著

许鹏展 郑芳友 袁志明 黄荔丽



机械工业出版社

本书以 OpenGL 开发为核心, 精心设计了 36 个实验。全书共分 10 章, 既介绍了基本的图形学要素, 包括 2D 图形和 3D 物体的绘制, 坐标系变换, 光照, 材质, 2D 纹理, 曲线和曲面等, 又进一步引入利用 OpenGL 开发图形工程的各种技术, 例如复杂 3D 建模, 音频系统, 摄像机控制, 粒子系统, 人机交互技术等。

本书图文并茂、实例丰富, 所有例程都是基于一个功能齐全的编程框架文件 (SDG 编程框架文件) 开发的, 代码有详细的中文注释, 可读性很强。本书既可作为高校的图形学实验教材, 也是学习 OpenGL 的理想参考书, 适合具有 C 语言基础的读者使用。

图书在版编目 (CIP) 数据

计算机图形学实验教程 (OpenGL 版) / 李胜睿等编著. —北京: 机械工业出版社, 2004.9

(高等院校计算机专业教育改革推荐教材)

ISBN 7-111-15278-6

I. 计... II. 李... III. 计算机图形学—教材 IV. TP391.41

中国版本图书馆 CIP 数据核字 (2004) 第 095370 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策 划: 胡毓坚

责任编辑: 陈振虹

责任印制: 石 冉

北京中兴印刷有限公司印刷·新华书店北京发行所发行

2004 年 10 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·15 印张·371 千字

0001—5000 册

定价: 27.00 元 (含 1CD)

凡购本图书, 如有缺页、倒页、脱页, 由本社发行部调换

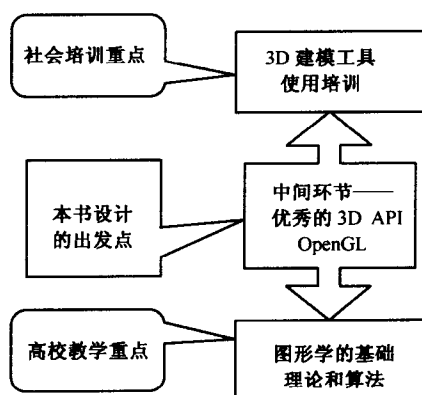
本社购书热线电话 (010) 68993821、88379646

封面无防伪标均为盗版

前 言

计算机图形学已经相当成熟，它广泛应用于广告、娱乐、医学、教育、科学、工程、航空等领域，但是国内图形学的高级人才十分匮乏。因此，培养合格的图形学人才，是高等院校计算机教学急待解决的问题。

目前国内多数计算机图形学教材都主要讨论图形学的基础理论和算法，而社会上的培训班多以培训 3d max, AutoCAD, Maya 等建模工具的使用为主。本书希望在二者之间求得较好的平衡，以培养具备一定理论基础，又具有较强动手能力的图形开发的人才为目标（见下图）。



OpenGL 是由计算机图形和动画领域公认的业界领袖 Silicon Graphics (SGI) 公司发布的 3D API。作为一个性能优越的 3D API, OpenGL 适合于广泛的计算环境, 从个人计算机到工作站和超级计算机, OpenGL 都能实现高性能的三维图形功能。OpenGL 已成为目前的三维图形开发标准, 是从事三维图形开发工作的技术人员必须掌握的开发工具。

本书以 OpenGL 开发为核心, 精心设计了 36 个实验, 旨在通过翔实精彩的例程, 层层深入地讲解图形学编程开发的各个要素, 循序渐进地引导读者将图形学理论与实际图形工程开发相结合, 为从事图形学理论研究或更大规模的 3D 工程开发打下坚实的基础。

为了使读者一开始就将精力集中在图形学相关的知识点和 OpenGL 自身的开发细节上, 而不必关心复杂繁琐的 Win32 编程, 本书提供了一个功能齐全的编程框架文件 (5DG 编程框架文件), 本书所有的实验例程都是基于该框架开发的。值得一提的是, 本书的所有例程都有详细的中文注释, 代码的可读性很强。

本书第 1 章介绍了图形学的各种基本要素在 OpenGL 中的实现效果, 11 个基本实验涵盖了在 Win32 下基于 OpenGL 开发的基本要素。第 2~8 章分别探讨了 OpenGL 的各技术要素, 包括光照、材质、纹理、缓冲区、3D 投影、建模、粒子系统等技术。第 9 章讨论了难度较大的技术, 包括音频系统、拣选和反馈等。第 10 章为综合实验, 运用本书所介绍的技术开发了一个简单的小飞机游戏。

本书可以作为中、高年级本科生计算机图形学的实验教材，也可作为具有 C 语言基础的读者学习图形学基本原理和 OpenGL 图形工程开发的参考书。

本书由 5DG 团队[⊖]编著，36 个实验例程均由团队成员创作。本书的编写得到了李翠华教授和李名世教授的鼎力支持以及刘丽老师和斐佳老师的帮助，在此表示最诚挚的谢意。

编 者

[⊖] 5DG 是 the 5th Dimension Group 的缩写。这是一个由 7 个富有激情的，热爱 OpenGL 的青年组成的团队。

目 录

前言

绪论	1
第 1 章 OpenGL 基本要素	4
1.1 5DG 的 Win32 编程框架	4
1.1.1 定义	4
1.1.2 ResizeWindowGL()设置 OpenGL 绘制场景的大小	6
1.1.3 创建 Win32 窗口	6
1.1.4 WinMain()	14
1.1.5 Initialize()	16
1.1.6 DrawSceneGL()	17
1.1.7 Update()	17
1.1.8 DeInitialize()	18
1.2 2D 图形	18
1.2.1 定义	18
1.2.2 DrawSceneGL()	18
1.2.3 疑难解答	20
1.3 阴暗处理	20
1.3.1 DrawSceneGL()	21
1.3.2 彩色三角形	21
1.3.3 单色正方形	22
1.3.4 绿色椭圆	22
1.3.5 疑难解答	22
1.4 3D 图形	23
1.4.1 改窗口标题	24
1.4.2 DrawCube()绘制绿色立方体	24
1.4.3 绘制金字塔和十字架	25
1.4.4 疑难解答	26
1.5 旋转	26
1.5.1 定义控制旋转的变量	27
1.5.2 glPushMatrix()和 glPopMatrix()	27
1.5.3 glRotatef()旋转坐标轴	27
1.5.4 Update()中的按键控制	28
1.5.5 疑难解答	29
1.6 纹理贴图	29

1.6.1	定义	29
1.6.2	LoadImage()载入 BMP 图片	30
1.6.3	LoadTexture()创建纹理	30
1.6.4	在 Initialize()里载入纹理	32
1.6.5	第一个纹理矩形	32
1.6.6	纹理金字塔	33
1.7	3D 建模与绘制	34
1.7.1	物体建模	34
1.7.2	定义	35
1.7.3	LoadTextureGL()创建纹理	35
1.7.4	Initialize()	36
1.7.5	香皂	36
1.7.6	香皂盒	38
1.7.7	香皂盒的开关动画控制	39
1.7.8	疑难解答	40
1.8	简单的光照	41
1.8.1	定义	41
1.8.2	设置光源属性	42
1.8.3	DrawCube()绘制长方体	42
1.8.4	使用光源	43
1.8.5	Update()	44
1.8.6	疑难解答	44
1.9	雾	45
1.9.1	定义	45
1.9.2	LoadTextureGL()创建纹理	45
1.9.3	初始化雾	46
1.9.4	三个金字塔	47
1.9.5	Update()	48
1.10	透明	48
1.10.1	定义	49
1.10.2	DrawPic()绘制平面	50
1.10.3	指定混合算子	50
1.10.4	DrawSceneGL()	51
1.10.5	Update()	51
1.10.6	疑难解答	52
1.11	地月系	52
1.11.1	定义	53
1.11.2	Initialize()	53
1.11.3	DrawSceneGL()	54

1.11.4 疑难解答	57
1.12 习题	57
第2章 光照和材质专题	58
2.1 聚光	58
2.1.1 定义	58
2.1.2 初始化光源	59
2.1.3 绘制机器人	60
2.1.4 绘制场景	61
2.1.5 按键控制	64
2.1.6 疑难解答	64
2.2 材质	64
2.2.1 光源和材质定义	65
2.2.2 Initialize()	65
2.2.3 光源的移动控制	66
2.2.4 提示信息	66
2.2.5 绘制小球	67
2.2.6 按键控制	69
2.2.7 疑难解答	70
2.3 习题	71
第3章 显示表专题	72
3.1 显示表的应用	72
3.1.1 定义	72
3.1.2 建立显示表	72
3.1.3 调用显示表	73
3.1.4 删除显示表	74
3.1.5 疑难解答	75
3.2 Bitmap 和 Outline 字体	75
3.2.1 位图字体 (Bitmap Fonts)	76
3.2.2 轮廓字体 (Outline Fonts)	80
3.3 中文字体	81
3.3.1 定义	81
3.3.2 建立字体	82
3.3.3 删除字体	83
3.3.4 显示字体	83
3.3.5 中文的输出	85
3.3.6 疑难解答	86
3.4 习题	86
第4章 纹理专题	87
4.1 纹理导入	87

4.1.1	本工程文件	87
4.1.2	创建纹理	87
4.1.3	获取导入的图片信息	89
4.1.4	建立临时位图	90
4.1.5	TGA 格式图片导入	91
4.1.6	定义 (Main.cpp)	93
4.1.7	载入纹理图片	94
4.1.8	绘制主场景	95
4.1.9	纹理切分	96
4.1.10	动画控制	97
4.1.11	疑难解答	99
4.2	户外地形	100
4.2.1	工程文件构成	100
4.2.2	导入文件数据	100
4.2.3	获取高度值	101
4.2.4	绘制地形	102
4.2.5	绘制主场景	104
4.2.6	疑难解答	104
4.3	天空纹理	105
4.3.1	天空盒 (Skybox)	105
4.3.2	设置纹理滤波方式	105
4.3.3	绘制天空盒	106
4.3.4	疑难解答	109
4.4	纹理特效综合	109
4.4.1	定义	110
4.4.2	建立黑色纹理 (BuildBlackTexture())	110
4.4.3	程序初始化	111
4.4.4	运动模糊	111
4.4.5	运动的隧道	114
4.4.6	绘制电影胶片	115
4.4.7	电视雪花特效	116
4.4.8	蝶形曲线和曲面	117
4.4.9	主场景绘制	118
4.5	习题	119
第 5 章	缓冲区专题	120
5.1	切割技术	120
5.1.1	深度测试	120
5.1.2	利用深度测试切割	121
5.1.3	使用自定义的切割面 (clipping plane)	122

5.1.4 按键控制	124
5.2 透视技术	124
5.2.1 Stencil 测试	124
5.2.2 透视技术	125
5.2.3 按键控制	127
第 6 章 坐标系和视图专题	128
6.1 gluLookAt() 详解	128
6.1.1 gluLookAt() 介绍	128
6.1.2 定义	129
6.1.3 Initialize()	129
6.1.4 移动 Camera	130
6.1.5 坐标系及锥体绘制	131
6.1.6 总场景绘制	133
6.2 透视投影和正射投影	134
6.2.1 定义	134
6.2.2 修改 ResizeWindowGL()	134
6.2.3 绘制公路路标	135
6.2.4 绘制公路和汽车	135
6.2.5 绘制视口场景	136
6.2.6 汽车引擎声音的播放	136
6.2.7 建立视口	137
6.2.8 设置正射投影	137
6.2.9 透视投影	139
6.3 阴影的实现	140
6.3.1 定义	141
6.3.2 阴影矩阵的计算	141
6.3.3 绘制投影平面	143
6.3.4 绘制十字架及阴影	143
6.3.5 绘制直升飞机及其阴影	144
6.3.6 绘制模拟光源	145
6.3.7 按键控制	146
6.3.8 疑难解答	146
6.4 习题	146
第 7 章 复杂物体建模专题	148
7.1 二次对象	148
7.1.1 操作二次对象流程	148
7.1.2 定义	148
7.1.3 ChangeStatus()	149
7.1.4 resetstatus()	150

7.1.5	SetBusy()	151
7.1.6	创建二次曲面对象	151
7.1.7	按键控制	151
7.1.8	场景绘制	153
7.1.9	释放空间	154
7.2	曲线和曲面	155
7.2.1	定义	155
7.2.2	Bezier 曲线	156
7.2.3	Bezier 曲面	157
7.2.4	NURBS 对象	159
7.2.5	NURBS 曲线	159
7.2.6	NURBS 曲面	160
7.2.7	按键控制	161
7.3	OBJ 文件载入	161
7.3.1	标准模板库	164
7.3.2	重要定义	164
7.3.3	函数声明	165
7.3.4	关键函数定义	165
7.3.5	法向量计算	168
7.3.6	初始化设置	171
7.3.7	绘制场景	171
7.3.8	释放存储空间	172
7.4	3ds 文件载入	173
7.4.1	3ds 文件格式简介	173
7.4.2	3ds 文件导入及使用	175
7.5	习题	177
第 8 章	粒子系统专题	179
8.1	简单粒子系统——星云	179
8.1.1	粒子的定义	179
8.1.2	粒子系统初始化	180
8.1.3	绘制星空粒子	181
8.1.4	更新粒子属性	181
8.1.5	粒子死亡和再生	182
8.1.6	绘制星空粒子系统	182
8.1.7	疑难解答	183
8.2	5DG'S Matrix 粒子系统	183
8.2.1	Matrix01 粒子系统	183
8.2.2	5DG 粒子系统	187
8.2.3	绘制 5DG 和 Matrix01 粒子系统	189

8.3	习题	190
第9章	高级专题	191
9.1	音频系统	191
9.1.1	链接音频库	191
9.1.2	定义	191
9.1.3	初始化音频系统	192
9.1.4	释放音频文件	193
9.1.5	初始化场景	193
9.1.6	绘制钢琴	194
9.1.7	播放控制	195
9.1.8	疑难解答	196
9.2	检选	196
9.2.1	实现检选的基本步骤	197
9.2.2	主要函数	197
9.2.3	定义	197
9.2.4	背景绘制	198
9.2.5	检选的实现	198
9.2.6	DrawObjects()	200
9.2.7	获取鼠标的屏幕坐标	202
9.2.8	转换鼠标的坐标	203
9.2.9	绘制鼠标指针	203
9.2.10	疑难解答	204
9.3	反馈	204
9.3.1	定义	205
9.3.2	插入物体标记	205
9.3.3	MakeSelection(int nChoice)	206
9.3.4	绘制物体外形方框	209
9.3.5	显示物体信息和帮助	210
9.3.6	疑难解答	211
9.4	习题	211
第10章	小飞机游戏设计专题	212
10.1	小飞机游戏的设计说明	212
10.1.1	需求	212
10.1.2	类的设计	212
10.1.3	技术要点	216
10.2	小飞机游戏的实现	217
10.2.1	模块设计	217
10.2.2	游戏的初始化	217
10.2.3	Opening	219

10.2.4 Running 219

10.2.5 Ending 221

10.2.6 DrawSceneGL() 222

10.3 习题 222

附录 224

附录 A 建立 5DG 工程..... 224

 A.1 使用简单的 5DG 框架 224

 A.2 使用复杂的 5DG 框架 226

附录 B 光盘资源说明 227

参考文献 228

绪 论

计算机图形学 (Computer Graphics)

计算机图形学属于可视化计算领域,是研究如何用计算机生成可视图形和如何用计算机模拟现实世界的科学。它的应用范围很广,包括 3D 的实时绘制(视频游戏)、视频捕捉和视频重建、动画特效处理(电影和电视)、图像处理、建模(工程和医学领域)等等。

计算机图形学源于学术兴趣,起初依靠政府的资助发展,但随着图形学软件在广播电视和电影领域内的广泛运用,越来越多的商业团体投资该领域,最后商业投资成为图形学发展的主要因素。

20 世纪 50 年代 CRT 显示器的发明促进了 2D 计算机图形学(2D Computer Graphics)的发展;1963 年,麻省理工学院的 Ivan E.Sutherland 发表了“画板(Sketchpad):第一个人机图形通信系统”的博士论文,标志着计算机图形学作为一个全新研究领域的起步;20 世纪 70 年代后,计算机工作站[⊖]的诞生带来了基于矢量的 3D 计算机图形学(3D Computer Graphics);现在计算机图形学已经相当成熟,它广泛应用于广告、娱乐、医学、教育、科学、工程、航空等领域。

2D 和 3D 计算机图形学

2D 计算机图形学源起 20 世纪 50 年代,最初是基于矢量图形设备(vector graphics devices)的。后来矢量图形设备逐步被光栅图形设备(Raster-based Devices)所取代。2D 图形分为矢量图和光栅图两大类。该领域的标志性的产品有 PostScript 语言和 X window system 协议。

3D 计算机图形学研究如何运用计算机和特殊的 3D 软件生成 3D 图形。3D 之于 2D 就像雕塑之于绘画,其独特之处在于采用三维的方式来表示和保存虚拟物体。迄今为止,3D 计算机图形学几个重要的技术包括:平滑阴暗处理(Flat shading)、Gourand 阴暗处理(Gourand shading)、纹理映射(Texture Mapping)、Phong 阴暗处理(Phong shading)、Bump 映射(Bump mapping)、光线跟踪(Ray Tracing)和射线算法(Radiosity)。

此外,2D 和 3D 领域还有许多相关的研究课题,比如色彩理论、光栅理论、矢量理论、物体表面的几何表示(曲线和曲面)、材质属性、图像压缩、动画、渲染、投影和坐标变换、消隐等等。

APIs 和各种工具、协议和软件

图形学经过多年的发展日趋成熟,不仅有 OpenGL 和 DirectX 两个优秀的 3D API,还有大量的图形开发工具,协议和软件产品,比如 Open Inventor, Crystal Space, Adobe Systems, Pixia, Macromedia Flash, Macromedia Shockwave, GKS(Graphical Kernel System), PostScript,

⊖ LISP machines, paintbox computers 和 Silicon Graphics workstations 等。

X Window System, Computer Graphics Metafile (CGM) 等等。

本书的内容

本书并不具体介绍各种底层的图形学算法原理,而是另辟蹊径,以如何运用 OpenGL 开发图形学应用程序为例,重点培养读者使用图形库开发图形工程的能力,精心设计的例程又涉及各种基本的图形学要素,读者可以在提高开发能力的同时掌握基本的图形学概念。

第 1 章介绍了图形学的几个基本要素,包括 2D 基本图形的绘制(1.2 节),3D 物体的建模和绘制(1.4, 1.7 节),阴暗处理(1.3 节),坐标变换(1.5 节),纹理映射(1.6 节),光照模型(1.8 节),自然现象的模拟(1.9 节),曲面(1.11 节)等。

第 2 章~第 8 章深入讨论第 1 章引入的各个主题,并且提供了颇具技巧性的例程和各种实用的技术:

第 2 章深入讨论光照(2.1 节)和材质(2.2 节)。

第 3 章介绍了典型的光栅图形——字体,并且提供了可输出英文字体和中文字体的代码(3.2, 3.3 节)。

第 4 章探讨各种纹理技术,提供了导入各种格式图片的纹理导入工程(4.1 节),介绍模拟现实世界的户外地形的绘制技术(4.2 节)和天空盒的技术(4.3 节),并提供了第一人称视角控制的工程代码,在 4.4 节综合了几种特殊的纹理技术,涉及纹理坐标控制,运动模糊等多个要素。

第 5 章的 5.1 节讨论隐藏面切割的相关技术,包括 z-buffer,切割面等,5.2 节介绍了一种 OpenGL 的 stencil 测试,利用 stencil 测试实现的透视技术相当有趣。

第 6 章深入探讨 3D 投影和坐标变换。6.1 节介绍的 `gluLookAt()` 是 OpenGL 中摄像机控制的基础,6.2 节讨论的正射投影和透视投影是整个 3D 场景应用的核心概念,6.3 节则介绍了阴影的实现,属于真实感图形和几何建模相关的内容。

第 7 章重点介绍了典型的 3D 建模技术,7.1 节讨论了二次曲面,7.2 节讨论了基本的 Bezier 曲线和曲面,NURBS 曲线和曲面,7.3 节介绍了 OBJ 文件的导入,7.4 节则介绍了复杂的 3DS 文件的导入,提供的 3DS 导入代码相当强大,利用该技术可以利用现有的 3DS 文件建立相当复杂的 3D 场景。

第 8 章介绍了自然现象模拟的最重要的技术——粒子系统,8.1 节的内容较基础,有助于读者迅速把握概念,8.2 节稍难,将粒子系统技术同纹理特效技术结合,效果绚丽。

第 9 章相对独立,9.1 节引入了强大的音频播放引擎,支持各种格式的音频文件。9.2 节和 9.3 节介绍了最复杂的 OpenGL 机制——检选和反馈,该机制为 OpenGL 对人机交互技术的支持。

第 10 章综合运用本书提供的大部分技术开发了一个简单的小飞机游戏,即作为本书的总结,又希望成为读者迈入更复杂的 3D 工程开发,乃至 3D 游戏开发的敲门砖。

如何使用本书

全书例程都基于 5DG 编程框架开发,读者无需白手起家,可以迅速地利用 5DG 框架开发自己的应用程序,同时可以把精力集中在 OpenGL 的开发上,不必过多地考虑复杂繁琐的 win32 编程。

本书的所有代码结构合理，有详细的中文注释，可读性很强。建议读者先从阅读例程的代码入手，编译运行例程代码，然后再阅读本书，这样学习可以事半功倍。同时第 2~8 章的内容相对独立，读者可以有选择的阅读，无需拘泥于章节顺序。

最后用摘自《real-time rendering》的话结尾，希望读者喜欢本书的例程，享受创造的快乐！

“No one knows, and each person can have a significant effect on the way the future turns out. There is no one future, no course that must occur. You create it.”