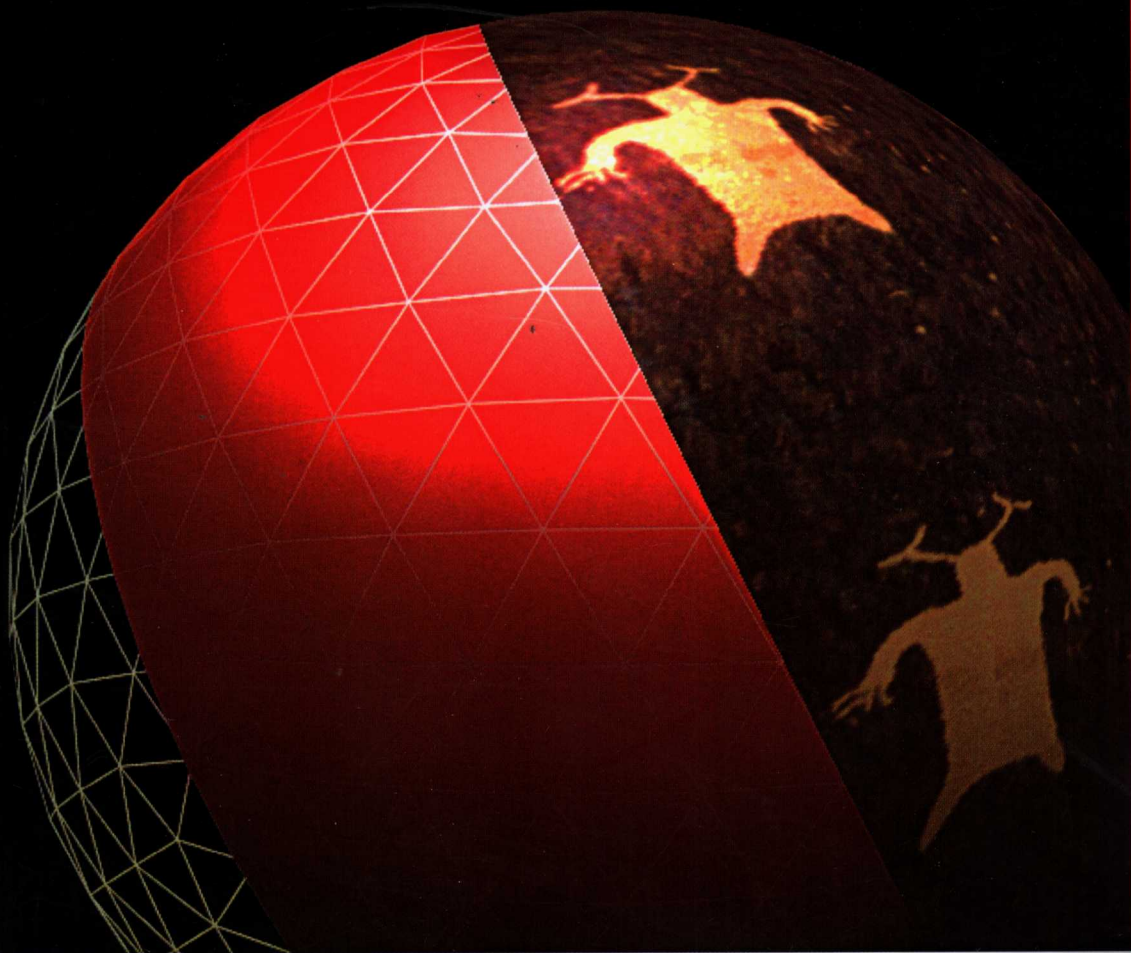


OpenGL 2.0

PEARSON
Addison
Wesley

精髓

■ Paul Martz 著 ■ 邓郑祥 译

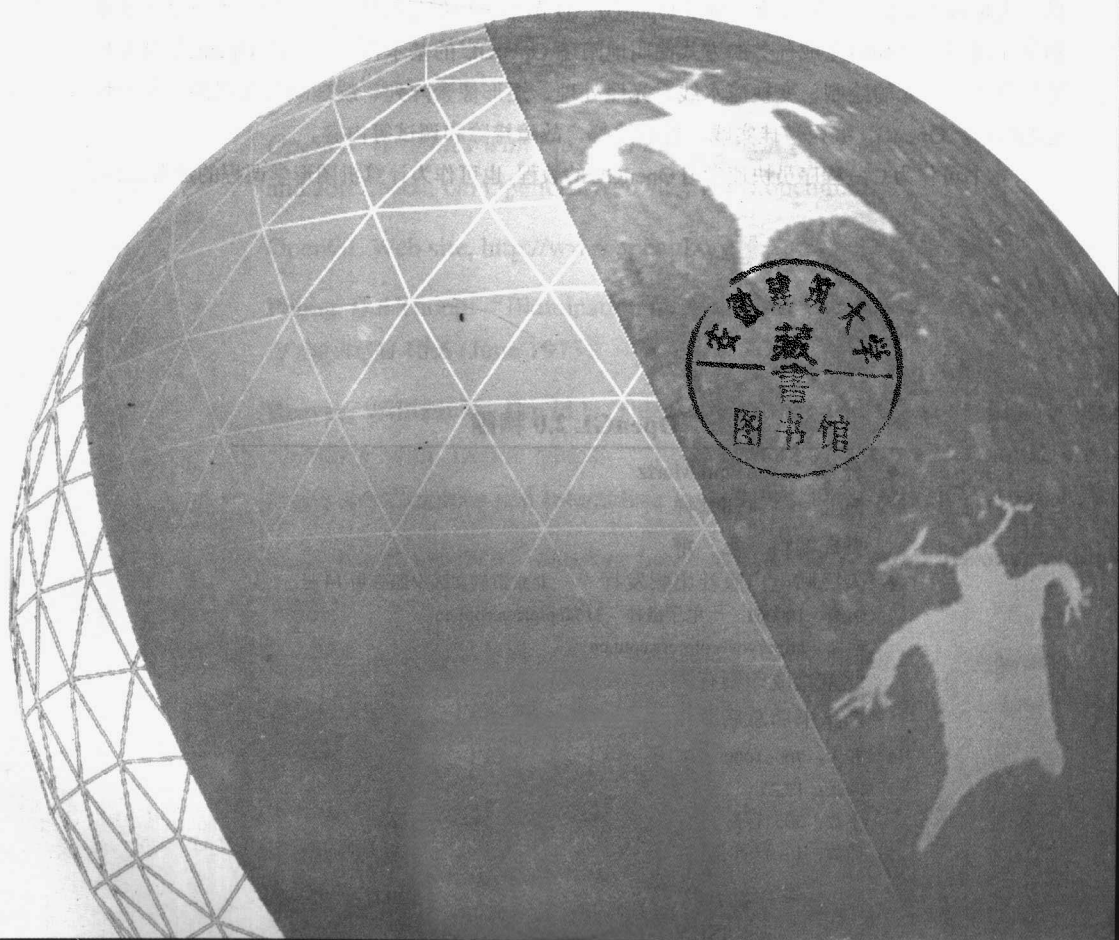


人民邮电出版社
POSTS & TELECOM PRESS

OpenGL 2.0

精髓

■ Paul Martz 著 ■ 邓郑祥 译



定价：39.00元

印刷：(010) 67151702 北京人民邮电出版社 电话：(010) 67151702

人民邮电出版社

图书在版编目 (CIP) 数据

OpenGL 2.0 精髓 / (美) 马茨 (Marts, P.) 著; 邓郑祥译.

—北京: 人民邮电出版社, 2006.7

ISBN 7-115-14908-9

I. O... II. ①马... ②邓... III. 图形软件, OpenGL

2.0—程序设计 IV. TP319.41

中国版本图书馆 CIP 数据核字 (2006) 第 066691 号

内 容 提 要

本书是针对初学者编写的, 旨在引导读者快速掌握 OpenGL 2.0 的精华和最常用特性, 以获得使用 OpenGL 进行图形编程所必需的知识; 同时, 为方便读者更深入地学习, 本书提供了大量参考资源, 为读者获取更多 OpenGL 3D 图形编程信息提供了路线图。本书分为 8 章和 4 个附录, 以如何完成任务的方式简明地阐述 OpenGL 的基本特性, 包括 OpenGL 开发环境的设置、图元的绘制、变换流水线、光照处理、矩形像素阵列、纹理映射和阴影、平台特定的接口、OpenGL 编程最佳实践、性能改善、故障排除和调试等内容。

本书可作为 C++ 程序员快速学习 OpenGL 的教程, 也可作为计算机图形学课程的参考资料。

OpenGL 2.0 精髓

-
- ◆ 著 Paul Martz
 - 译 邓郑祥
 - 责任编辑 王 琳
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
 - 邮编 100061 电子函件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京铭成印刷有限公司印刷
 - 新华书店总店北京发行所经销
 - ◆ 开本: 700×1000 1/16
 - 印张: 17.25
 - 字数: 305 千字 2006 年 7 月第 1 版
 - 印数: 1—4 000 册 2006 年 7 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-3204 号

ISBN 7-115-14908-9/TP · 5489

定价: 36.00 元

读者服务热线: (010) 67132705 印装质量热线: (010) 67129223

版 权 声 明

Authorized translation from the English language edition, entitled OPENGL® DISTILLED, 1st Edition, 0321336798 by MARTZ, PAUL, published by Pearson Education, Inc, Publishing as Addison Wesley Professional, Copyright © 2006.

All rights reserved. No part of this book may be reproduced, or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and POSTS & TELECOMMUNICATIONS PRESS Copyright ©.

版权所有。未经出版者书面许可，对本书任何部分不得以任何方式或任何手段复制和传播。

人民邮电出版社经 Addison-Wesley, Pearson Education 公司授权出版。

版权所有，侵权必究。

作者简介

Paul Martz 是 SimAuthor 公司的资深软件工程师，负责开发基于 OpenGL 的飞机数据可视化软件。加入 SimAuthor 之前，Paul 受聘于惠普公司的图形软件实验室和 Evans&Sutherland 的工作站图形分部。

Paul 从 1986 年就涉足计算机图形学领域，在其职业生涯中，主要致力于 3D 图形系统软件和 OpenGL 设备驱动程序的开发。1993 年，Paul 为 Evans&Sutherland 的第一个 OpenGL 产品开发了性能敏感的代码。在 20 世纪 90 年代末，Paul 致力于为惠普基于 OpenGL 的工作站图形硬件开发 HP-UX 和 Microsoft Windows OpenGL 设备驱动程序。Paul 参与了 OpenGL 规范修订的审阅工作，当前是 OpenGL ARB 的独立撰稿人。

Paul 领导了 OpenGL Technical FAQ 的编写工作，OpenGL 网站 (<http://www.opengl.org>) 提供了这些内容；他还编写了多部技术白皮书、Internet 教程和计算机图形学图书书评。

序 言

很久以前（大约 15 年前），即使是委婉地说，部署渲染图形的跨平台应用程序也是一种挑战。那时候，人们认为这种问题是由于软件和 API，而不是计算机硬件引起的。每家厂商都有自己的库和方法，用于控制硬件来绘制图像，但都是专用的，要将应用程序部署到多家厂商的硬件中，必须使用大量 `#ifdef` 代码块。现在，这种代码虽然还未绝迹，但就图形方面而言，情况显然要好得多。

也就是在那时候，一些聪明的设计者在开发计算机硬件和接口方面拥有丰富经验，他们致力于解决这种问题。在那个库总是有一家厂商设计和实现的时代，**Silicon Graphics Computer System** 却接受了竞争对手的理念和方法并将其集成到库中，也许该公司最有创意的理念是同竞争对手共享接口。有鉴于此，**OpenGL** 降生于一个和睦的大家庭中，这个家庭就是 **OpenGL** 架构委员会（ARB），它们深知，倘若没有应用程序来驱动，计算机和图形学将毫无价值。

设计出一个编程接口，将不同图形学算法实现之间的差别抽象掉，这不是一项简单任务。另外，这种接口要被开发人员接受并采用，必须能够简单地完成任务，且效率尽可能高，能够满足交互式 and 3D 计算机图形的需求。既要简单又要高效，这可不容易。从某种意义上说，这是软件工程的本质，而最初设计该接口时，该学科还处于萌芽状态。

更为复杂的是，那时的计算机图形学领域还处于其起步时代。图形学技术及

其硬件实现正处于如火如荼的快速发展中，OpenGL 要成为成功的竞争者，必须紧跟该领域的发展步伐。退一步说，这一切都令人兴奋不已。

回到现在。你手捧的这本书介绍 OpenGL，这是跨平台 3D 计算机图形和图像处理领域中占统治地位的编程接口，让新手能够轻松地创建使用 3D 图形的交互式应用程序，其功能非常强大，足以驱动大部分对性能敏感的图形应用程序。然而，为实现这种目标，该接口（编写本书时为 2.0 版）包含 800 多个入口，以提供最新的计算机图形学功能。学习所有这些函数和方法是一项令人畏缩的任务，阅读经验丰富者编写的指南将大有裨益。

Paul Martz 长期致力于图形学领域，见证了 OpenGL 的发展历程，并为之做出了贡献。他将自己的宝贵经验拿出来供大家分享，使读者的征服 OpenGL 之路尽可能平坦。Paul 全面考虑 OpenGL 后，从中提取 OpenGL 编程和最佳实践的精华，集结成本书作为读者的指南。祝贺 Paul 取得的成就，并祝读者在使用 OpenGL 的过程中好运气。

渲染愉快！

Dave Shreiner

2006 年 1 月于加州 Mountain View

前 言

和其他计算机技术一样，计算机图形技术也在以惊人的速度发展。在 20 世纪 90 年代末，诞生了第一个针对消费者的硬件加速 3D 图形卡，这标志着 3D 图形软件开发进入了一个新时代。这些图形卡早已进入垃圾堆，取而代之的是更强大和复杂的硬件。

如果读者是 OpenGL 新手，可能想马上开始编写复杂的 OpenGL 应用程序。虽然市面上有很多高深的图形编程图书，但大部分要求读者熟悉 OpenGL 或其他 3D 图形应用程序编程接口（API）。在向复杂的编程任务发起挑战之前，快速地掌握 OpenGL 是有必要的。

本书简介

OpenGL 是跨平台的高性能 3D 计算机图形 API，是行业标准。本书简明地阐述了常用的 OpenGL 基本特性。本书重点介绍 OpenGL 基本特性，而不讨论算法，并指出了更深入的参考文献，从而快速、准确地向读者提供了有帮助的信息。

OpenGL 1.0 版是 SGI 于 1992 年推出的，很快在 3D 软件开发行业占据了统治地位。有多个用于 Microsoft Windows、Linux、UNIX 和 Apple Mac OS 的版本，支持 OpenGL 的图形硬件厂商包括 3Dlabs、ATI、Intel、Matrox、NVIDIA 和 SGI。

OpenGL 是一种开放标准，这无疑是其得以流行的原因之一。OpenGL 特性集由 OpenGL 架构委员会（ARB）确定，这是一个管理机构，由来自多家重要硬件和软件公司的代表组成¹。这些成员定期集会，讨论和批准对 OpenGL 规范的增补和修订。ARB 由多家厂商组成，这确保 OpenGL 能够在各种图形架构上运行。

本书介绍 2004 年 9 月发布的 OpenGL 2.0 版。虽然自面世以来 OpenGL 得到了重大的发展，但每个版本都与以前的版本兼容。向后兼容的好处是显而易见的，很多公司利用 OpenGL 的稳定性来降低软件开发的成本。

随着时间的推移，ARB 在 OpenGL 标准中加入了新的渲染方法和模式，并摒弃了一些古老的 OpenGL 特性。尽管如此，这些特性仍不能完全满足开发人员的需要，其原因各不相同，如特例或平台特定的适用性、重叠或有限的功能、普遍的低效实现等。

由于需要在新增功能的同时，保持与日益过时的老式特性兼容，这导致 OpenGL 规范极其庞大和复杂。看看下面的数据吧：

在 1992 年，OpenGL 1.0 规范仅 163 页，而现在的 OpenGL 2.0 规范为 368 页，如果加上新增的 OpenGL 着色语言规范，总共为 474 页。

虽然如此，详尽的文档对经验丰富的程序员来说是不可或缺的；但对于 OpenGL 新手而言，日益庞大和复杂的文档增加了学习难度。另外，由于信息量急剧增加，OpenGL 新手难以获得有关公认新用法的建议，或将大量时间浪费在学习过时的方法或模式上。

为解决这种问题，本书集中介绍用于 3D 软件开发的 OpenGL 基本特性，现代 OpenGL 编程中常用的特性和公认的最佳实践；对于那些已被摒弃的特性，除非有助于阐述公认的最佳实践或重要概念，否则不做介绍。

为进一步方便读者学习，本书以如何完成工作的方式提供信息，让读者无需在浩瀚的信息中艰难跋涉就能够快速找到所需的信息。因此，本书很少讨论和解释图形学算法，但由于阴影算法的普适性而 OpenGL 又不直接支持它，在第 6

¹ 在 2006 年 1 月，有投票权的成员为 3Dlabs、Apple、ATI、Dell、IBM、Intel、NAVIDIA、SGI 和 Sun。

章对其进行了讨论。对于其他算法，本书假设读者已熟悉计算机图形学，或将本书作为大学计算机图形学课程的参考资料。

本书介绍了 OpenGL 的部分特性，大多数程序员都能够使用它们来完成其主要的 OpenGL 编程工作。由于本书只介绍部分特性，读者可能想了解更详细的信息。如果是这样，本书将是获取更多 OpenGL 和 3D 图形编程信息的入口或路线图。

本书经常引用下述资源。

OpenGL 官方网站 (<http://www.opengl.org>)，这里有开发人员论坛、示例程序、实用库、白皮书、文档、超链接以及其他宝贵的 OpenGL 资源。

*OpenGL Graphics System: A Specification*¹，它是全面的 OpenGL 官方规范，虽然是针对 OpenGL 实现厂商编写的，但对于应用程序开发人员来说也是不可或缺的。OpenGL 网站以 PDF 方式提供了该文档，可免费下载。

《OpenGL 编程指南》，也被称为红皮书，它是权威的 OpenGL 编程指南。另外，它广泛地讨论了图形学算法及其在 OpenGL 中的实现。

OpenGL Shading Language（其中文版即将由人民邮电出版社出版）²，也被称为橘皮书，详细介绍了 OpenGL 的可编程图形硬件接口，这种功能是 OpenGL 2.0 规范新增的。

本书针对的读者

本书是为不熟悉 OpenGL 的 C++ 程序员编写的。

本书假设读者有一定的计算机图形学知识，熟悉图形学资料、教材和算法，但不要求读者是图形学专家。例如，读者无需知道如何编写 Bresenhan 算法，但应熟悉扫描转换的概念。如果读者使用过其他图形 API 将很有帮助，但这并非必要条件。

¹ 译者注：正文译为《OpenGL 图形系统》。

² 译者注：正文译为《OpenGL 着色语言》。

读者应熟悉 C++ 编程语言。本书配套网站中的示例程序是使用 C++ 编写的, 可在 Microsoft Windows、Apple Mac OS 以及大部分 Linux 和 UNIX 平台上编译和运行。如果读者熟悉 C 语言且手边有 C++ 参考手册, 应该能够理解书中的代码片段。

读者应熟悉向量和矩阵方面的数学知识, 这些知识在典型的线性代数课程中都会介绍。读者应能够理解矩阵乘法和向量-矩阵乘法。如果读者已通过使用其他 3D 图形 API 理解了变换和坐标系, 将发现本书很容易理解。

当然, 读者还应具备基本的数学知识, 如代数学、几何学和三角学。

本书内容

本书共分 8 章并有 4 个附录, 内容简介如下。

第 1 章“OpenGL 简介”, 概述 OpenGL 架构和概念。简要地介绍支持库 GLUT 和 GLU, 讨论了如何设置 OpenGL 开发环境, 并提供一个简单的 OpenGL 示例; 最后, 介绍 OpenGL API 的历史。

第 2 章“绘制图元”, 介绍如何使用 OpenGL 来渲染几何数据。它描述 OpenGL 图元类型以及如何使用顶点数组和缓存对象等特性来渲染它们, 还讨论其他绘图细节, 如清空帧缓存、顺序依赖性和深度测试。

第 3 章“变换和观察”, 描述 OpenGL 变换流水线。该章介绍如何设置观察位置和方向、设置透视和正交投影以及控制视口, 并对如何用鼠标进行交互式观察进行介绍。

第 4 章“光照”, 解释如何设置光照参数, 这些参数如何控制光源的位置、方向、颜色以及材质的反射属性。

第 5 章“矩形像素阵列”, 介绍 OpenGL 绘制、读取和拷贝像素数据块的方式。

第 6 章“纹理映射”, 阐述如何创建纹理对象并将其应用于渲染的几何体。纹理映射是一个功能强大的工具, 可提高图像的逼真度。本章将讨论改善镜面反射点、实现环境映射和创建阴影的技巧。

第 7 章“扩展和版本”, 描述如何编写独立于平台的版本安全的代码, 它们

使用实现特定的图形硬件功能。

第 8 章“平台特定的接口”，描述特定于 Apple Mac OS、Linux 和 Microsoft Windows 中的 API 接口，并将讨论如何创建和使用多个 OpenGL 渲染上下文以及在它们之间共享对象。

附录 A “其他特性”，将简要介绍多个不在本书范围内的特性，如 OpenGL 着色语言、裁剪面、雾效、多重采样等。

附录 B “最佳实践”，列举公认的 OpenGL 编程最佳实践，帮助确保 OpenGL 应用程序是健壮且可移植的。

附录 C “性能”，将介绍如何获得准确的计时指标以及如何找出并消除性能瓶颈。

附录 D “故障排除和调试”，帮助读者解决常见的问题。

约定

本书使用了下述约定：

- OpenGL 入口、函数名和矩阵使用**粗体**表示；
- 变量和参数名、空间维和矩阵分量使用*斜体*表示；
- 枚举类型、枚举值、常量、源代码、参考文献、文件名和系统路径使用常规体表示；
- 程序代码使用灰色底纹，以便将其与正文区分开来。

本书以命令总结的方式介绍 OpenGL 命令，包含函数原型、参数描述和 OpenGL 版本等内容。

下面是一个命令总结模板：

命令的函数原型

对命令及其参数的描述。

► OpenGL 版本：支持该命令的 OpenGL 版本。

在函数原型中，用方括号括起的是一组选项，例如，`glLight*v()`表示一组函数。

```
void glLight[fd]v(GLenum light, GLenum pname, const TYPE* param);
```

该原型定义了两个函数：`glLightfv()`和`glLightdv()`。`f`和`d`指出了参数的数据类型——分别是`GLfloat`和`GLdouble`。在函数原型的参数列表中，将`TYPE`替换为实际的参数类型。

命令总结可用作简明的命令参考。

本书配套网站

本书配套网站可以提供本书的所有示例代码，其英文版网址为 <http://www.opengl distilled.com>，单击链接 **Download** 可下载这些文件。所有代码都在下述平台上进行了编译和测试：Microsoft Windows（使用 Visual Studio 6.0、.NET 2003 和 Cygwin 开发环境）、Apple Mac OS X（使用 Xcode 2.0）和 Red Hat Linux（使用 GNU 编译器）。

示例程序需要 GLUT、TIFF 库和 GNU Make，配套网站提供了下载这些库和实用程序的链接。

本书引用了一些彩图，本书配套网站和 Addison-Wesley 的网站以 PDF 文件的方式提供了这些彩图，后者的英文网址为 <http://www.awprofessional.com/title/0321336798>。

如果在本书中发现了错误，将在配套网站上张贴勘误。

目 录

第 1 章 OpenGL 简介	1
1.1 OpenGL 是什么	2
1.1.1 基本知识和架构	4
1.1.2 语法	8
1.1.3 状态和查询	10
1.2 GLUT	17
1.3 GLU	18
1.4 开发环境	18
1.4.1 Apple Mac OS X	19
1.4.2 Linux	20
1.4.3 Microsoft Windows	20
1.5 简单示例	21
1.6 OpenGL 的历史	27
1.6.1 1992 年之前	28
1.6.2 OpenGL 与 Iris GL 之比较	29
1.6.3 OpenGL 与 PEX 之比较	30
1.6.4 不断发展的 OpenGL	31
1.7 更详细的信息	31
参考文献	32

第2章 绘制图元	35
2.1 OpenGL 图元	36
2.1.1 图元类型	37
2.1.2 顶点共享	39
2.2 指定顶点数据	39
2.2.1 使用 glBegin()/glEnd()绘制图元	40
2.2.2 使用顶点数组绘制图元	42
2.3 绘图细节	57
2.3.1 清空帧缓存	57
2.3.2 模型变换	58
2.3.3 平滑着色和恒定着色	59
2.3.4 多边形模式	60
2.3.5 深度测试	60
2.3.6 共面图元	61
2.3.7 alpha 和透明度	63
2.4 性能问题	65
2.4.1 显示列表	65
2.4.2 面剔除	68
2.4.3 顶点数组的大小	69
2.5 更详细的信息	69
参考文献	70
第3章 变换和观察	71
3.1 坐标系和矩阵	73
3.1.1 旋向性	73
3.1.2 矩阵	73
3.1.3 矩阵堆栈和矩阵状态	75
3.1.4 矩阵操纵命令	76
3.2 变换流水线	78
3.2.1 物体坐标	79
3.2.2 模型变换	80
3.2.3 世界坐标	81
3.2.4 视点变换	81

3.2.5	眼坐标	82
3.2.6	投影变换	82
3.2.7	裁剪坐标	82
3.2.8	透视除法	83
3.2.9	归一化的设备坐标	83
3.2.10	视口变换	83
3.2.11	窗口坐标	84
3.2.12	控制变换	84
3.3	设置模型视点矩阵	86
3.3.1	使用 <code>gluLook()</code> 指定视点变换	87
3.3.2	创建视点变换矩阵	88
3.3.3	示例代码中的视点变换	89
3.4	透视和平行投影	90
3.5	视口	92
3.6	选择	93
3.6.1	执行选择	93
3.6.2	示例程序中的选择	98
3.7	更详细的信息	99
	参考文献	99
第 4 章	光照	101
4.1	概述	102
4.1.1	环境光、散射光和镜面反射光	103
4.1.2	控制 OpenGL 光照	104
4.1.3	最简单的光照代码	105
4.1.4	内部光照计算	105
4.2	法线	106
4.3	光源参数	107
4.3.1	典型用法	108
4.3.2	更详细的信息	108
4.4	材质参数	109
4.4.1	使用 <code>glColor*()</code> 修改材质参数	111
4.4.2	典型用法	112
4.4.3	更多的信息	113

4.5	定位光源和定向光源	113
4.6	调试光源	114
4.6.1	调试空白窗口	115
4.6.2	法线	115
4.6.3	错误的面剔除	115
4.6.4	调试位置和方向	115
4.6.5	调试光源颜色	116
4.6.6	顶点光照异常	116
4.6.7	没有镜面反射点	117
4.6.8	直线和点的颜色	118
4.7	更详细的信息	118
	参考文献	119
第 5 章	矩形像素阵列	121
5.1	绘制像素	122
5.1.1	当前光栅位置	123
5.1.2	使用 <code>glDrawPixels()</code> 绘制矩阵像素阵列	126
5.2	读取像素	128
5.3	复制像素	129
5.4	性能问题	130
5.4.1	使用可替代 <code>glDrawPixels()</code> 的方式	130
5.4.2	刷新 OpenGL 流水线	131
5.5	调试	131
5.5.1	光栅位置不正确	131
5.5.2	裁剪光栅位置	132
5.5.3	禁用纹理映射和雾效	132
5.5.4	深度测试	133
5.5.5	像素数据对齐	133
5.5.6	被遮住的窗口	134
5.5.7	内存分配问题	134
5.6	更详细的信息	135
	参考文献	135
第 6 章	纹理映射	137