

高等学校数字媒体技术系列教材

图形 程序设计

*Graphics
Programming*

李祥 刘琳 朱梅 编著



高等教育出版社
HIGHER EDUCATION PRESS

高等学校数字媒体技术系列教材

图形程序设计

Tuxing Chengxu Sheji

李祥 刘琳 朱梅 编著



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书是高等学校数字媒体技术系列教材之一,依据教育部示范性软件学院建设工作办公室组织编制的《高等学校数字媒体技术专业规范》中对图形程序设计课程的要求编写而成。全书共9章,分别为 OpenGL 基础知识,OpenGL 建模技术,坐标变换,OpenGL 颜色,光照与材质处理,纹理映射技术,帧缓存技术与动画,渲染器的实现,混合、反走样与雾化。全书覆盖数字媒体技术应用领域图形程序设计中建模、绘制等基本问题,包括多边形建模、几何变换、光照模型、材质处理、纹理映射、帧缓存技术、动画、裁剪、隐藏、反走样、图像合成等全部内容。同时,通过标准图形库 OpenGL 的教学和编程作业,可帮助学生掌握基本的三维图形编程技能,为今后深入图形相关的应用开发与研究奠定基础。

本书获“东华理工大学教材建设基金项目”资助,可作为高等院校本科计算机科学与技术、软件工程、数字媒体技术等专业相关课程的教材,也可供相关技术人员参考。

图书在版编目(CIP)数据*

图形程序设计/李祥,刘琳,朱梅编著.——北京:
高等教育出版社,2013.12

高等学校数字媒体技术系列教材

ISBN 978-7-04-038611-0

I. ①图… II. ①李… ②刘… ③朱… III. ①图形软件—程序设计—高等学校—教材 IV. ① TP391.41

中国版本图书馆 CIP 数据核字(2013)第 244328 号

策划编辑 倪文慧
插图绘制 尹莉

责任编辑 倪文慧
责任校对 刘春萍

封面设计 于涛
责任印制 张福涛

版式设计 杜微言

出版发行 高等教育出版社
社址 北京市西城区德外大街 4 号
邮政编码 100120
印刷 北京七色印务有限公司
开本 787 mm × 1092 mm 1/16
印张 18.5
字数 410 千字
购书热线 010-58581118

咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
版次 2013 年 12 月第 1 版
印次 2013 年 12 月第 1 次印刷
定价 29.00 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物料号 38611-00

高等学校数字媒体技术系列教材编委会

主 任：卢 苇

副主任：吴中海(北京大学)

委 员：(按姓氏拼音排序)

董槐林(厦门大学)

贾金原(同济大学)

李 祥(东华理工大学)

王崇文(北京理工大学)

王方石(北京交通大学)

王万良(浙江工业大学)

叶德健(复旦大学)

曾智勇(福建师范大学)

朱 青(北京工业大学)

序

数字媒体包括文字、图形、图像、音频、视频和动画等各种媒体形式，互联网、移动、IPTV、数字出版等各种传播形式，以及媒体的采集、存取、加工和分发的数字化过程。近 20 年来，随着互联网应用的普及，数字媒体技术发展迅速，已成为推动我国文化创意产业和电子信息产业发展的重要力量。为此，迫切需要培养一支适应时代要求和产业需求、具有创新能力、复合交叉型的数字媒体技术专业人才培养队伍。

数字媒体技术是信息技术和媒体艺术相结合的一门交叉学科。美国麻省理工学院的媒体艺术与科学实验室是该学科的探索者和领跑者。我国早期的数字媒体技术专业人才培养起源于计算机辅助设计与图形学专业方向和数字图像处理专业方向。2004 年教育部批准设置数字媒体技术专业后，许多高校纷纷开设数字媒体技术专业，面向动画、游戏、新媒体、虚拟现实等领域培养本科生或研究生，受到了社会和企业的欢迎。

为了进一步推动数字媒体技术专业人才培养，教育部示范性软件学院建设工作办公室于 2010 年 3 月成立了数字媒体技术专业规范研制专家组，旨在进一步加强高等学校数字媒体技术专业建设和人才培养，组织高水平教材的编写工作。2011 年 10 月，专家组编制完成《高等学校数字媒体技术专业规范》，同时依据专业规范，委托有丰富教学实践经验的教师编写一批专业基础课程和专业核心课程教材。这套教材正是过去两年来的重要工作成果之一。

数字媒体技术的发展日新月异，新概念、新方法、新技术层出不穷，这一切需要我们与时俱进地反映到大学课程的教学计划和教学内容中。相信这套教材的出版能够起到积极推动各高校数字媒体技术专业建设，提高教学水平和人才培养质量的作用。

希望广大教师在教学过程中对教材提出宝贵的意见和建议，使其在广泛使用过程中不断完善。

教育部示范性软件学院建设工作办公室
2013 年 7 月

前 言

计算机图形学是随着计算机技术在图形处理领域中的应用而发展起来的新技术,在航空航天、影视制作、汽车制造、工业生产、游戏制作、科学仿真等社会各个行业得到广泛应用。图形程序设计已成为计算机类专业及其他工科专业的重要专业课程,更是数字媒体技术工作者必须掌握的基础技能。

OpenGL (Open Graphics Library, 开放式的图形程序接口) 源于 SGI 公司为其图形工作站开发的 IRIS GL, 在跨平台移植过程中发展而成。OpenGL 作为一个性能优越的图形应用程序设计接口 (API), 具有广泛的可移植性, 独立于硬件系统、操作系统和窗口系统, 适应于广泛的计算机环境, 从嵌入式应用平台、个人计算机、工作站到超级计算机, 用户都可以利用 OpenGL 创建漂亮的三维图形, 具有良好的跨平台特性。因此, OpenGL 是从事三维图形开发工作的技术人员所必须掌握的基础性开发工具之一。

为适应数字媒体技术专业图形程序技能培养的需求, 进一步提高计算机图形处理等课程的教学质量, 并为相关行业的工程技术人员提供有益参考, 作者根据多年教学经验, 结合当前计算机图形处理研究的一些热点问题, 以 OpenGL 开发为切入点, 在分析国内外多种同类教材的基础上编写了本书, 相信大多数图形类编程爱好者、IT 专业人员和愿意接触图形编程的学生都会从本书中获益。

本书共 9 章, 内容覆盖数字媒体技术应用领域图形程序设计中建模、绘制等领域的基本问题, 包括多边形建模、几何变换、光照模型、材质处理、纹理映射、帧缓存技术、动画、裁剪、隐藏、反走样、图像合成等全部内容。同时, 通过标准图形库 OpenGL 的教学和编程作业, 可帮助学生掌握基本的三维图形编程技能, 为今后深入图形相关的数字媒体领域的应用开发与研究奠定基础。使用本书教学的规划学时为 38 学时, 其中实验 8 学时, 任课教师可根据教学实际情况进行相应调整。

本书由李祥、刘琳、朱梅编著。李祥负责全书的总体规划设计、实例编制、各章内容修改等工作, 并完成第 1、2 章编写; 刘琳完成第 5~9 章编写, 朱梅完成 3、4 章编写。在本书的创作过程中, 还得到了教育部示范性软件学院数字媒体技术专业教材编写组、东华理工大学软件学院同行、朋友的大力帮助, 试用过程中软件学院学生更提出了很多宝贵的修改意见。在此, 向他们表示感谢。还要感谢高等教育出版社相关领导和编辑的精心组织、细心审阅和修改, 正是因为他们辛苦的付出, 才使本书能在第一时间和读者见面。书中还参考了大量国内外专著、教材和图片, 在此一并表示衷心感谢。

本书的内容涉及面广、专业性强, 虽几经斟酌, 多方查找资料, 编程核对, 但由于作者的教学和科研水平有限, 书中难免有不足之处, 敬请读者不吝指正。

编 者
2013 年 5 月

郑重声明

高等教育出版社依法对本书享有专有出版权。任何未经许可的复制、销售行为均违反《中华人民共和国著作权法》，其行为人将承担相应的民事责任和行政责任；构成犯罪的，将被依法追究刑事责任。为了维护市场秩序，保护读者的合法权益，避免读者误用盗版书造成不良后果，我社将配合行政执法部门和司法机关对违法犯罪的单位和个人进行严厉打击。社会各界人士如发现上述侵权行为，希望及时举报，本社将奖励举报有功人员。

反盗版举报电话 (010)58581897 58582371 58581879

反盗版举报传真 (010)82086060

反盗版举报邮箱 dd@hep.com.cn

通信地址 北京市西城区德外大街4号 高等教育出版社法务部

邮政编码 100120

目 录

第 1 章 OpenGL 基础知识	1	2.1.3 线段的绘制	41
1.1 OpenGL 概述	1	2.1.4 多边形的绘制	42
1.1.1 OpenGL 是直观的三维图形		2.1.5 规则三维物体绘制函数	48
开发环境	1	2.1.6 开发实例：基本三维几何	
1.1.2 OpenGL 是三维图形的		物体绘制	50
开发标准	2	2.2 曲线与曲面绘制	61
1.1.3 OpenGL 的基本功能与操作	2	2.2.1 曲线的基本理论	61
1.1.4 OpenGL 的体系结构	3	2.2.2 样条曲线的绘制	63
1.1.5 创建 OpenGL 控制台应用		2.2.3 样条曲面的绘制	67
程序框架	4	2.2.4 NURBS 曲线和曲面绘制	72
1.2 OpenGL 图形的实现方式	8	2.2.5 二次曲面	81
1.2.1 设备描述表与渲染描述表	8	2.3 显示列表	83
1.2.2 OpenGL 像素格式设置	11	2.3.1 显示列表的概念	83
1.2.3 OpenGL 图形处理流程	13	2.3.2 显示列表的创建	84
1.2.4 OpenGL 图形绘制方式	14	2.3.3 显示列表的执行	85
1.2.5 OpenGL 程序运行方式	15	2.3.4 多重显示列表	85
1.3 OpenGL 图形开发库	16	2.3.5 显示列表的索引	86
1.3.1 开发库的组成	16	2.3.6 显示列表的嵌套	87
1.3.2 基本数据类型	20	2.3.7 实例介绍	88
1.3.3 OpenGL 库函数命名规则	21	2.4 位图、图像与文本绘制	90
1.4 基于 OpenGL 的高层图形库	21	2.4.1 位图绘制	90
1.5 OpenGL 应用程序框架	24	2.4.2 图像绘制	96
1.6 开发实例：基本二维几何		2.4.3 文本绘制	101
物体绘制	29	2.5 开发实例：植物生长 L 系统——	
本章小结与习题	32	花朵与叶脉绘制	106
第 2 章 OpenGL 建模技术	33	本章小结与习题	111
2.1 基本图元及规则物体绘制	33	第 3 章 坐标变换	112
2.1.1 图形显示控制	33	3.1 从三维图形到二维图像	112
2.1.2 点的绘制	39	3.1.1 三维图形的输出过程	112

3.1.2 坐标系与坐标变换	113	5.1.2 材质颜色	153
3.1.3 矩阵操作	113	5.2 创建光源	155
3.2 几何变换	116	5.2.1 定位和衰减	155
3.2.1 平移变换	116	5.2.2 聚光灯	155
3.2.2 旋转变换	117	5.2.3 多光源	156
3.2.3 缩放变换	118	5.2.4 控制光源的位置与方向 ...	160
3.2.4 变换次序	119	5.3 选择光照模型	163
3.2.5 实例介绍	120	5.4 开发实例：植物质地与环境 光线的呈现	168
3.3 投影变换	121	本章小结与习题	170
3.3.1 透视投影	122	第6章 纹理映射技术	174
3.3.2 正交投影	124	6.1 纹理映射的定义	175
3.4 视窗变换	125	6.1.1 一维纹理映射的定义	175
3.4.1 定义视窗	125	6.1.2 二维纹理映射的定义	179
3.4.2 变换 z 坐标	126	6.2 纹理数据的获取	180
3.5 附加裁剪面	126	6.2.1 直接创建法	180
3.6 矩阵堆栈	129	6.2.2 读取外部文件	181
3.7 组合变换举例	136	6.3 纹理坐标	182
3.8 开发实例：植物生长 L 系统—— 图形与场景变换	139	6.3.1 纹理坐标的指定	182
本章小结与习题	139	6.3.2 纹理坐标的自动计算	183
第4章 OpenGL 颜色	141	6.4 纹理控制	187
4.1 RGBA 模式与颜色索引模式	141	6.4.1 缩小与放大滤波	188
4.1.1 RGBA 显示模式	143	6.4.2 纹理重复和截取	189
4.1.2 颜色索引显示模式	144	6.5 多重纹理	195
4.1.3 RGBA 模式与颜色索引 模式的对比	146	6.6 开发实例：植物生长 L 系统—— 天空、土地等背景纹理映射	199
4.1.4 抖动操作	147	本章小结与习题	204
4.2 指定阴影模型	148	第7章 帧缓存技术与动画	213
4.3 开发实例：植物生长 L 系统—— 植物颜色的渲染	149	7.1 OpenGL 中的各种缓存	213
本章小结与习题	150	7.2 缓存操作技术	214
第5章 光照与材质处理	151	7.2.1 清除缓存	214
5.1 真实世界与 OpenGL 光照	151	7.2.2 选择绘图颜色缓存	215
5.1.1 发射光、环境光、漫反射 光和镜面反射光	152	7.2.3 屏蔽缓存	216
		7.3 测试操作	217
		7.3.1 裁剪测试	218

7.3.2 Alpha 测试	218	8.7.4 深度缓冲器算法	255
7.3.3 模板测试	219	8.7.5 深度排序算法	256
7.3.4 深度测试	225	8.7.6 区间扫描线算法	257
7.3.5 混合、抖动、逻辑操作 ...	225	8.8 多边形的扫描转换	258
7.4 双缓存动画	226	8.8.1 内外测试法	258
7.5 开发实例：植物生长 L 系统—— 缓存区的清除与测试操作	229	8.8.2 区域填充	259
本章小结与习题	230	8.8.3 种子填充算法	259
第 8 章 渲染器的实现	237	8.8.4 扫描线种子填充算法	261
8.1 实现的基本策略	237	8.9 植物生长系统——多边形的绘制 ...	261
8.2 光栅化	238	本章小结与习题	262
8.2.1 数值微分法	239	第 9 章 混合、反走样与雾化	263
8.2.2 中点画线法	240	9.1 混合	263
8.2.3 Bresenham 画线算法	241	9.1.1 源因子和目标因子	263
8.3 线段裁剪	243	9.1.2 混合的简单使用	266
8.3.1 Cohen – Sutherland 裁剪 算法	243	9.1.3 具有深度缓存的三维混合 ...	267
8.3.2 Liang – Barsky 裁剪算法 ...	246	9.2 反走样	268
8.4 多边形裁剪	249	9.2.1 行为控制函数	269
8.5 其他图元的裁剪	252	9.2.2 点和线的反走样	269
8.6 三维裁剪	253	9.2.3 多边形的反走样	271
8.7 隐藏面消除	254	9.3 雾化	274
8.7.1 景物空间和图像空间	254	9.4 植物生长 L 系统——远景雾化与 边界反走样	279
8.7.2 多边形区域排序算法	254	本章小结与习题	280
8.7.3 背面消除	255	参考文献	283

第 1 章 OpenGL 基础知识

学习目标：

- ☆ 掌握图形学编程的基本概念。
- ☆ 了解 OpenGL 的基本功能与操作。
- ☆ 学习图形系统和模型的基本原理、结构。

1.1 OpenGL 概述

1.1.1 OpenGL 是直观的三维图形开发环境

长期以来，人们致力于计算机三维图形技术的研究。由于生成真实感的图形需要较深的数学知识（如三角学、线性代数、微分几何等），在很大程度上制约了计算机三维图形技术的发展。今天，很多三维图形开发工具已经隐藏了这些细节，编程人员仅需要考虑如何创作自己的作品即可。OpenGL 就是这样的一个工具。

OpenGL（Open Graphics Library，开放式的图形程序接口）是美国 SGI 公司为图形工作站开发的一种功能强大的三维图形机制，其目的是将用户从具体的硬件系统和操作系统中解放出来，可以完全不去理解这些系统的结构和指令系统，只要按规定的格式书写应用程序，就可以在任何支持该语言的硬件平台上运行。经过长期的发展，OpenGL 已经成为高性能图形和交互式视景处理的工业标准，借助 OpenGL 的帮助，编程人员可以轻松地进行三维图形（动画）的设计工作。

OpenGL 实际上是一种图形与硬件的接口。它包括了几百个指令和函数，开发者可以用这些函数来建立三维模型和进行三维实时交互。与其他图形程序设计接口不同，OpenGL 提供了十分清晰明了的图形函数，因此，初学的程序设计人员也能利用 OpenGL 的图形处理能力和 1 670 万种色彩的调色板很快地设计出三维图形以及三维交互软件。

OpenGL 的图形函数不要求开发者把三维物体模型的数据写成固定的数据格式，这样开发者不但可以直接使用自己的数据，而且可以利用其他不同格式的数据源。这种灵活性极大地节省了开发者的时间，提高了软件开发效率。

OpenGL 提供一种直观的编程环境，它提供的一系列函数大大地简化了三维图形程序。例如：

- OpenGL 提供一系列的三维图形单元供开发者调用。
- OpenGL 提供一系列的图形变换函数。
- OpenGL 提供一系列的外部设备访问函数，使开发者可以方便地访问鼠标、键盘、空间球、数据手套等。

这种直观的三维图形开发环境体现了 OpenGL 的技术优势，这也是许多三维图形开发者热衷于使用 OpenGL 的缘由所在。

1.1.2 OpenGL 是三维图形的开发标准

在计算机发展初期，人们就开始进行计算机图形的开发，并尝试用图形来表示各种类型的数据及结果，以增强直观性。到 20 世纪 90 年代，随着计算机软硬件和计算机图形学技术的高度发展，三维图形得以迅猛发展，于是各种三维图形工具软件包相继推出，如 PHIGS、PEX、RenderMan 等。这些软件包有些侧重于使用方便，有些侧重于渲染效果或与应用软件的连接，但没有一种软件包能够在交互式三维图形建模能力、外部设备管理以及编程方便程度上与 OpenGL 相比拟。

OpenGL 通过对图形库（Graphics Library，GL）的进一步发展，实现了二维和三维的高级图形技术，在性能上表现得异常优越，它包括建模、变换、光线处理、色彩处理、动画以及更先进的处理能力，如纹理影射、物体运动模糊等。OpenGL 的这些能力为实现逼真的三维渲染效果、建立交互的三维景观提供了优秀的软件工具。OpenGL 在硬件、窗口、操作系统方面是相互独立的。

许多计算机公司已经把 OpenGL 集成到各种窗口和操作系统中，其中操作系统包括 UNIX、Windows NT、DOS 等，窗口系统有 X 窗口、Windows 等。为了实现一个完整功能的图形处理系统，设计一个与 OpenGL 相关的系统结构如下：其最底层是图形硬件，第二层为操作系统，第三层为窗口系统，第四层为 OpenGL，第五层为应用软件。OpenGL 是网络透明的，在客户 - 服务器（Client - Server）体系结构中，OpenGL 允许本地和远程绘图。所以在网络系统中，OpenGL 在 X 窗口、Windows 或其他窗口系统下都可以以一个独立的图形窗口出现。

OpenGL 作为一个性能优越的图形应用程序设计接口（Application Programming Interface，API）适用于广泛的计算环境，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。由于许多在计算机界具有领导地位的计算机公司纷纷采用 OpenGL 作为三维图形应用程序设计界面，OpenGL 应用程序具有广泛的移植性。因此，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所必须掌握的开发工具。

1.1.3 OpenGL 的基本功能与操作

OpenGL 能够对整个三维模型进行渲染着色，从而绘制出与客观世界十分类似的三维场景。另外 OpenGL 还可以进行三维交互、动作模拟等。主要包括如下一些具体功能。

（1）模型绘制

OpenGL 能够绘制点、线和多边形，应用这些基本的形体可以构造出几乎所有的三维模型。

(2) 模型观察

在建立了三维景物模型后,就需要用 OpenGL 描述如何观察所建立的三维模型。观察三维模型是通过一系列的坐标变换进行的。模型的坐标变换使观察者能够在视点位置观察与视点相适应的三维模型景观。在整个三维模型的观察过程中,投影变换的类型决定三维模型的观察方式,不同的投影变换得到的三维模型的景象也是不同的。最后的视窗变换则对模型的景象进行裁剪缩放,即决定整个三维模型在屏幕上的图像。

(3) 颜色模式的指定

OpenGL 应用了一些专门的函数来指定三维模型的颜色。程序开发者可以选择两个颜色模式,即 RGBA 模式和颜色索引模式。在 RGBA 模式中,颜色直接由 RGB 值来指定;在颜色索引模式中,颜色值由颜色表中的一个颜色索引值来指定。开发者还可以选择平面着色和光滑着色两种着色方式对整个三维景观进行着色。

(4) 光照应用

用 OpenGL 绘制的三维模型必须加上光照才能与客观物体更加相似。OpenGL 提供了 4 种光(辐射光、环境光、镜面光和漫反射光)的使用,另外还可以指定模型表面的反射特性。

(5) 图像效果增强

OpenGL 提供了一系列增强三维景观图像效果的函数,这些函数通过反走样、混合和雾化来增强图像的效果。反走样用于改善图像中线段图形的锯齿而使其更平滑,混合用于处理模型的半透明效果,雾化使得影像从视点到远处逐渐褪色,更接近于真实。

(6) 位图和图像处理

OpenGL 还提供了专门对位图和图像进行操作的函数。

(7) 纹理映射

三维景物因缺少景物的具体细节而显得不够真实,为了更加逼真地表现三维景物,OpenGL 提供了纹理映射的功能。OpenGL 提供的一系列纹理映射函数使得开发者可以十分方便地把真实图像贴到景物的多边形上,从而可以在视窗内绘制逼真的三维景物。

(8) 实时动画

为了获得平滑的动画效果,需要先在内存中生成下一幅图像,然后把已经生成的图像从内存复制到屏幕上,这就是 OpenGL 的双存技术(double buffer)。OpenGL 提供了一系列双缓存技术的函数。

(9) 交互技术

目前有许多图形应用需要人机交互,OpenGL 提供了三维图形人机交互接口,用户可以方便地选择修改三维景物中的物体。

1.1.4 OpenGL 的体系结构

OpenGL 是一套图形标准,它是按照计算机图形学的原理设计而成的,符合光学和视觉原理,非常适合可视化仿真系统。

在 OpenGL 中允许视景对象用图形方式表达, 如由物体表面顶点坐标集合构成的几何模型, 这类图形数据含有丰富的几何信息, 得到的仿真图像能充分表达出其形体特征; 而且在 OpenGL 中有针对三维坐标表示的顶点的几何变换, 通过该变换可使顶点在三维空间内进行平移和旋转, 对于由顶点的集合表示的物体则可以实现其在空间中的各种运动。另外 OpenGL 通过光照处理计算各顶点的颜色, 可以表达出物体的三维特性, 绘制出真实感更强的三维图形。

由于 OpenGL 是一个独立于硬件的高效接口, 其中没有执行窗口任务或获取用户输入的函数, 程序员必须通过窗口系统来控制硬件。因此在三维图形的绘制过程中, 应用程序需要利用 OpenGL 与操作系统的基本指令, 使窗口系统与操作系统进行交互控制, 通过硬件驱动程序操作各个图形硬件, 进而完成三维图形的绘制。一个完整的窗口系统的 OpenGL 图形处理系统的层次结构如图 1.1 所示。

OpenGL 采用 C-S 机制, 即客户 (用 OpenGL 绘制场景的应用程序) 向服务器 (即 OpenGL 内核) 发布 OpenGL 命令, 服务器则解释这些命令。大多数情况下, 客户和服务在同一机器上运行。正是 OpenGL 的这种 C-S 机制, 使其可以十分方便地在网络环境下使用。因此 Windows 下的 OpenGL 是网络透明的。图 1.2 给出了网络下的 OpenGL 体系结构, 在实际操作中, 应用程序发出 OpenGL 命令, 由动态链接库 OpenGL32.dll 接受打包后, 发送到服务器端的 WINSRV.DLL, 然后由它通过 DDI (Device Driver Interface, 设备驱动程序接口) 层发往视频显示驱动程序。如果系统安装了硬件加速器, 则由硬件相关的 DDI 来处理。

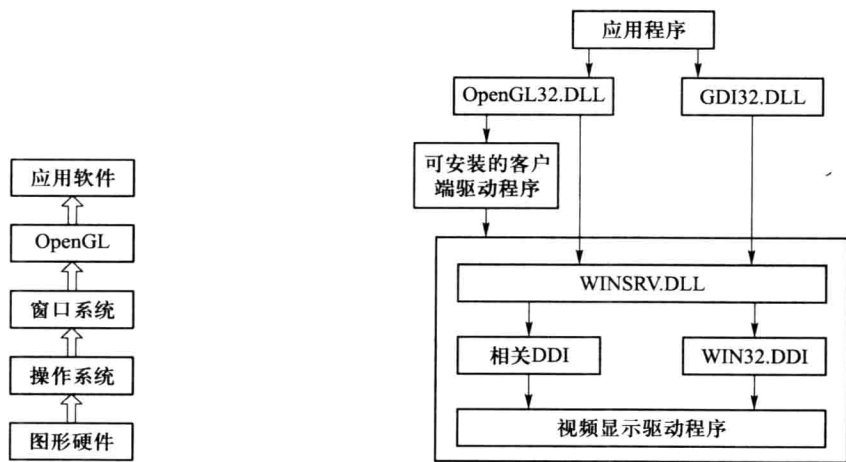


图 1.1 OpenGL 图形处理系统的层次结构

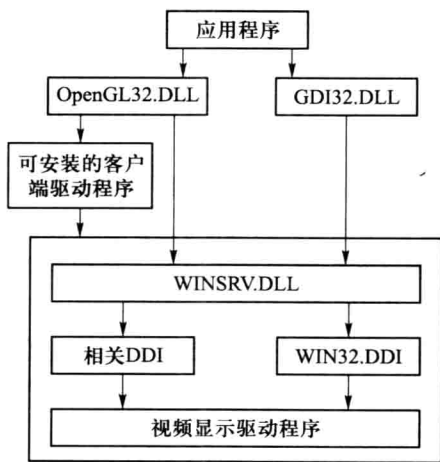


图 1.2 OpenGL/Windows 体系结构

1.1.5 创建 OpenGL 控制台应用程序框架

1. 在 Visual Studio 2008 环境下创建 OpenGL 控制台应用程序

目前主流的集成开发环境已经开始使用 Visual Studio.net 2008, 这里介绍一下如何在该环

境下创建 OpenGL 控制台应用程序。

(1) OpenGL 安装

① 从 <http://download.csdn.net/source/274113> 下载 OpenGL 类库, 该类库是 1.4 版本的, 也可以到 OpenGL 官网下载其他版本。将下载的文件解压到一个临时文件夹中。

② 将解压文件中的所有 .h 头文件复制到 C:\Program Files\Microsoft Visual Studio 9.0\VC\include\GL 目录中 (没有 GL 目录就自己创建一个, 这里的具体路径由计算机中 Visual Studio 2008 的安装位置而定); 将 .lib 文件复制到 C:\Program Files\Microsoft Visual Studio 9.0\VC\lib 目录中; 将 .dll 文件复制到 C:\Windows\System32 目录中。

(2) OpenGL 工程配置与测试

① 创建一个 Win32 控制台应用程序, 工程名命名为 FirstOpenGL。

② 链接 OpenGL 函数库, 该操作主要使编译器能找到 OpenGL 的函数库编译程序。

a. 创建一个 CPP 文件, 如 FirstOpenGL.cpp。

b. 在 Visual C++ 2008 中右击项目, 再单击“属性”, 如图 1.3 所示。

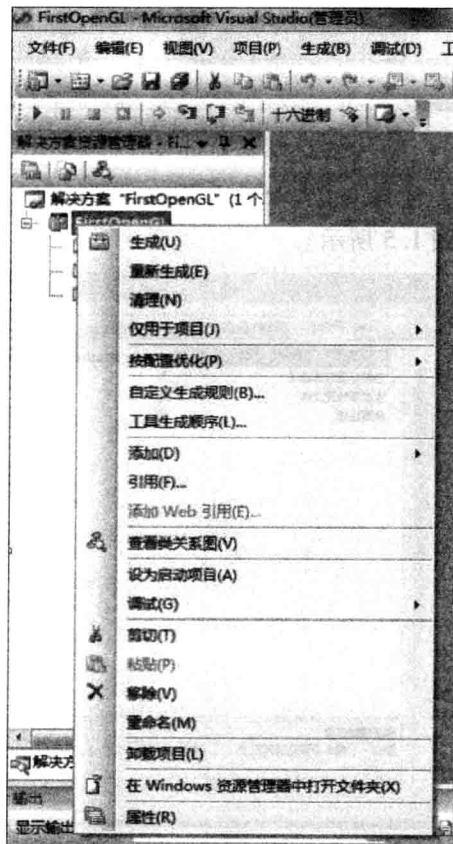


图 1.3 “属性”菜单

这样，就完成了 OpenGL 开发环境的配置。

c. OpenGL 程序测试：在 .cpp 文件中输入以下代码，如能出现图 1.6 所示的结果表示安装正确。

```
#include <gl/glut.h>
void myDisplay( void )
{
    glClear( GL_COLOR_BUFFER_BIT );
    glRectf( -0.5f, -0.5f, 0.5f, 0.5f );
    glFlush();
}

int main( int argc, char * argv[ ] )
{
    glutInit( &argc, argv );
    glutInitDisplayMode( GLUT_RGB | GLUT_SINGLE );
    glutInitWindowPosition( 100, 100 );
    glutInitWindowSize( 400, 400 );
    glutCreateWindow( "第一个 OpenGL 程序" );
    glutDisplayFunc( &myDisplay );
    glutMainLoop();
    return 0;
}
```

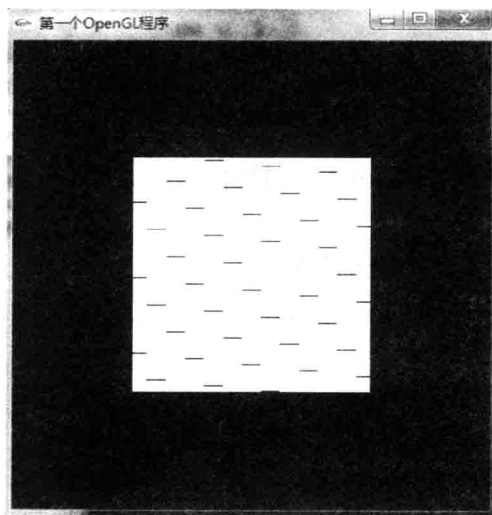


图 1.6 效果图