

程序设计

OpenGL

乔林 费广正 林杜 编著
芦丽丹 马晓峰



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



内附
光盘

OpenGL 程序设计

乔林 费广正等 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

OpenGL 是目前应用最为广泛的开放式图形编程标准, 用户可以利用它编写出高质量的图形绘制程序。

本书从编写 OpenGL 程序的基础知识和基本概念出发, 讲解了如何在 Win32 环境下正确设置环境以进行 OpenGL 程序设计; 然后介绍了 OpenGL 的基本内容, 如何在程序中读取三维图形数据, 以及 OpenGL 的高级编程方法, 并编写了一个屏幕保护程序和一个虚拟建筑物漫游程序。最后, 重点介绍了 Windows 环境下对 OpenGL 功能的扩展及程序改进。

本书示例丰富、实用, 语言通俗易懂, 介绍全面, 无论对大专院校的大学生、研究生, 还是对科研院所的研究人员, 都是一本三维图形技术难得的指导和参考书。此外, 本书亦可作为从事计算机图形工作一般人员的三维图形技术引导, 也非常适合作为 OpenGL 三维图形编程的培训教程。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: OpenGL 程序设计

作 者: 乔林、费广正等

出 版 者: 清华大学出版社 (北京清华大学学研楼, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 胡伟卷

印 刷 者: 北京市人民文学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 29.5 字数: 714 千字

版 次: 2000 年 4 月第 1 版 2000 年 4 月第 1 次印刷

书 号: ISBN 7-900625-15-1

印 数: 0001~5000

定 价: 54.00 元

前 言

目前，计算机图形学已进入三维时代，三维图形在人们周围无所不在。魅力无穷的计算机广告，扣人心弦的三维游戏，声势浩大、震撼人心的影视特效，无一不是三维图形大显身手的得意之作。

科学可视化、计算机动画和虚拟现实已成为近年来计算机图形学的三大热门话题，而这三大热门话题的技术核心均为三维图形。当前，三维图形在军事、航空、航天、医学、地质勘探、文化艺术和艺术造型等方面有着十分广泛的应用。

众多的计算机工作者，尤其是广大的计算机爱好者，对计算机图形、图像怀着浓厚的兴趣。许多计算机爱好者在图形学这片土地上苦苦耕耘，创作出许多优秀的软件作品。他们在享受由此带来的乐趣的同时，也真真切切地感受到其中的艰辛。他们盼望有一种简单易用的编程标准，正像许多人当年盼望 Windows 界面编程标准一样，因为他们深知：只有标准的才是有生命力的。

1992年，OpenGL 正式成为适用于各种计算机环境下的三维应用程序接口（3D API）。目前，它已成为国际上通用的开放式三维图形标准。Compaq, IBM, Intel, Microsoft, DEC, HP 和 SUN 等在计算机界具有主导地位的公司纷纷采用 OpenGL 图形标准。各种支持 OpenGL 的图形卡，3D 加速卡层出不穷，已成为计算机界继 CPU、内存、硬盘之后的又一快速增长点。逼真的 3D 游戏让人目不暇接，令人身临其境的虚拟现实也正向人们走来。这一切都必然吸引着更多的人加入到图形这一行列中，然而，国内介绍 OpenGL 的书可谓少而又少，致使不少人仍在苦苦地寻觅，焦急地等待。

事实上 OpenGL 是一种与硬件、窗口系统和操作系统相独立的一系列 API，它可以被集成到 Unix, Windows 95, Windows NT 和 X Window 等窗口系统中（这里尤其值得指出的是：作为软件界全球霸主的 Microsoft 为表示其支持 OpenGL 的决心，已将 OpenGL 作为核心模块之一加入到 Windows NT 3.5 以后的版本中）。OpenGL 由大量功能强大的图形函数组成，开发人员可以利用这些函数对整个三维图形进行光色渲染，从而方便地绘制出客观世界逼真的三维景象。

本书作者结合自己的实际开发经验，以及使用时的大量心得，通过丰富的教学实例，系统地讲述了 OpenGL 三维图形设计的基本概念和方法。

本书共由 20 章组成，第 1~第 4 章从编写 OpenGL 程序的基础知识和基本概念出发，讲解了如何在 Win32 环境下正确设置环境以进行 OpenGL 程序设计；第 5~第 8 章介绍了 OpenGL 的基本内容，包括几何图元的构造、坐标变换、颜色和光照等；第 9 章介绍了如何在程序中读取三维图形数据；第 10~第 16 章讲解了 OpenGL 的高级编程方法，内容包括：位图、字体和图像、纹理映射、各种特殊效果、帧缓存技术与动画、显示列表、参数曲面和曲线以及利用选择与反馈进行交互式图形程序设计；第 17 和第 18 章编写了两个实

际应用程序,一个屏幕保护程序和一个虚拟房间漫游程序。第 19、20 章重点讲解了 Windows 环境下对 OpenGL 功能的扩展及程序改进。

本书由乔林、费广正、林杜、芦丽丹和马晓峰合力编写而成,示例极为丰富、相当实用。无论是对大专院校的大学生、研究生,还是对科研院所的研究人员,都无疑是一本难得的三维图形技术指导和参考书。本书亦可作为从事计算机图形工作的一般人员的三维图形技术引导,此外,也非常适合作为 OpenGL 三维图形编程的培训教程。

为了便于用户使用,本书附有示例程序的光盘。所附程序盘按书中章节分目录存放,每章中的例子程序以 examxx 的名称顺序命名,分别表示书中具有完整代码的程序。程序使用 Visual C++ 5.0 以上的版本编译,运行环境是 Windows 98 或 Windows NT。书中的程序均由作者调试通过。

目 录

| | |
|--|----|
| 第 1 章 OpenGL 编程基础知识 | 1 |
| 1.1 图形学的基本发展状况 | 1 |
| 1.1.1 蓬勃发展的计算机图形学 | 1 |
| 1.1.2 计算机图形学的广泛应用 | 2 |
| 1.2 OpenGL 概述 | 3 |
| 1.2.1 OpenGL 基本概念 | 4 |
| 1.2.2 OpenGL 命令语法及各种状态的含义 | 7 |
| 1.3 运用辅助库创建规则几何对象 | 8 |
| 1.3.1 辅助库简介 | 8 |
| 1.3.2 辅助库分类 | 8 |
| 1.3.3 辅助库应用示例 | 8 |
| 1.4 OpenGL 辅助库函数详解 | 16 |
| 1.4.1 窗口初始化函数 | 16 |
| 1.4.2 窗口处理和事件处理函数 | 17 |
| 1.4.3 定义场景绘制循环函数 | 17 |
| 1.4.4 三维物体绘制函数 | 18 |
| 1.4.5 颜色索引表装入函数 | 19 |
| 1.4.6 空闲事件处理函数 | 19 |
| 1.5 Windows 环境下 OpenGL 编程简介 | 19 |
| 1.5.1 OpenGL 在 Windows 下的运行机制 | 19 |
| 1.5.2 Windows 下新增 OpenGL 相关函数 | 20 |
| 1.5.3 VC 环境下基于 OpenGL 的编程步骤 | 20 |
| 1.5.4 建立三维实体模型 | 22 |
| 第 2 章 Win32 环境下 OpenGL 编程概论 | 25 |
| 2.1 Windows 环境下 OpenGL 基本程序结构 | 25 |
| 2.1.1 Windows 环境下 OpenGL 工作机制 | 25 |
| 2.1.2 OpenGL 与 Windows 的绘图方式 | 25 |
| 2.1.3 像素格式的结构和参数设置 | 26 |
| 2.2 Windows 环境下 OpenGL 编程方法 | 28 |
| 2.2.1 示例源程序 | 28 |
| 2.2.2 建立 GDI 窗口 | 35 |

| | | |
|--------------|--|-----------|
| 2.2.3 | 建立用于 OpenGL 绘制的窗口 | 36 |
| 2.2.4 | 用 OpenGL 绘图 | 37 |
| 2.2.5 | 释放设备描述表和绘制描述表 | 39 |
| 2.3 | 利用 Visual C++ 6.0 的 AppWizard 编写 OpenGL 应用程序 | 39 |
| 2.3.1 | 创建一个新工程 | 40 |
| 2.3.2 | 设置编译环境 | 43 |
| 2.3.3 | 加入 OpenGL 库 | 43 |
| 第 3 章 | Win32 环境下 OpenGL 相关函数及结构 | 44 |
| 3.1 | 与 OpenGL 相关的 Win32 函数 | 44 |
| 3.1.1 | ChoosePixelFormat 函数 | 44 |
| 3.1.2 | DescribePixelFormat 函数 | 46 |
| 3.1.3 | GetPixelFormat 函数 | 47 |
| 3.1.4 | GetEnhMetaFilePixelFormat 函数 | 48 |
| 3.1.5 | SetPixelFormat 函数 | 48 |
| 3.1.6 | SwapBuffers 函数 | 49 |
| 3.2 | 与 OpenGL 相关的 wgl 函数 | 50 |
| 3.2.1 | wglCreateContext 函数 | 50 |
| 3.2.2 | wglCreateLayerContext 函数 | 50 |
| 3.2.3 | wglCopyContext 函数 | 51 |
| 3.2.4 | wglDeleteContext 函数 | 52 |
| 3.2.5 | wglDescribeLayerPlane 函数 | 52 |
| 3.2.6 | wglGetCurrentContext 函数 | 52 |
| 3.2.7 | wglGetCurrentDC 函数 | 53 |
| 3.2.8 | wglGetLayerPaletteEntries 函数 | 53 |
| 3.2.9 | wglGetProcAddress 函数 | 53 |
| 3.2.10 | wglMakeCurrent 函数 | 54 |
| 3.2.11 | wglRealizeLayerPalette 函数 | 54 |
| 3.2.12 | wglSetLayerPaletteEntries 函数 | 55 |
| 3.2.13 | wglShareLists 函数 | 55 |
| 3.2.14 | wglSwapLayerBuffers 函数 | 56 |
| 3.2.15 | wglUseFontBitmaps 函数 | 56 |
| 3.2.16 | wglUseFontOutlines 函数 | 58 |
| 3.3 | 与 OpenGL 相关的结构 | 59 |
| 3.3.1 | GLYPHMETRICSFLOAT 结构 | 59 |
| 3.3.2 | LAYERPLANEDESCRIPTOR 结构 | 60 |
| 3.3.3 | PIXELFORMATDESCRIPTOR 结构 | 62 |
| 3.3.4 | POINTFLOAT 结构 | 64 |
| 第 4 章 | OpenGL 与 Visual C++ 6.0 事件编程 | 66 |

| | |
|---|------------|
| 4.1 简单编程思路 | 66 |
| 4.1.1 程序功能简介 | 66 |
| 4.1.2 简要实现步骤 | 66 |
| 4.2 实例过程 | 67 |
| 4.2.1 建立一个单文档 | 67 |
| 4.2.2 用资源编辑器制作一个命令操作对话框 | 67 |
| 4.2.3 为命令操作对话框建立一个视图类 | 67 |
| 4.2.4 在 CMainFrame 中创建一个 CSplitterWnd 类 | 68 |
| 4.2.5 编写命令操作对话框的事件处理程序 | 69 |
| 第 5 章 基本几何图元的定义 | 84 |
| 5.1 绘制初始化命令 | 84 |
| 5.2 基本图元的绘制 | 85 |
| 5.2.1 点、线与多边形的简单定义 | 85 |
| 5.2.2 特殊多边形的绘制方法 | 87 |
| 5.2.3 点、线与多边形属性的设置 | 88 |
| 5.2.4 法向量的计算方法 | 89 |
| 5.2.5 利用简单几何图元构造较复杂形体 | 90 |
| 5.3 基本图元绘制实例程序 | 93 |
| 第 6 章 坐标变换 | 109 |
| 6.1 坐标变换的基本概念 | 109 |
| 6.2 通用变换命令 | 112 |
| 6.3 视图和造型变换 | 113 |
| 6.3.1 变换的顺序 | 113 |
| 6.3.2 造型变换 | 114 |
| 6.3.3 视图变换 | 117 |
| 6.4 投影变换 | 119 |
| 6.4.1 透视投影 | 119 |
| 6.4.2 正交投影 | 121 |
| 6.5 视口变换 | 122 |
| 6.5.1 定义视口 | 122 |
| 6.5.2 变换 z 坐标 | 123 |
| 6.6 操作矩阵栈 | 123 |
| 6.7 附加裁剪面 | 125 |
| 6.8 利用几种变换综合绘制机器人手臂 | 127 |
| 第 7 章 OpenGL 颜色 | 140 |
| 7.1 色彩视觉 | 140 |
| 7.2 计算机颜色 | 141 |

| | |
|-----------------------------|------------|
| 7.2.1 颜色生成原理 | 141 |
| 7.2.2 RGB 颜色模型 | 142 |
| 7.3 颜色模式 | 142 |
| 7.3.1 RGBA 颜色模式 | 143 |
| 7.3.2 颜色索引模式 | 143 |
| 7.3.3 两种模式的应用场合 | 143 |
| 7.4 指定颜色和明暗处理模型 | 144 |
| 7.4.1 在 RGBA 模式下指定颜色 | 144 |
| 7.4.2 在颜色索引模式下指定颜色 | 145 |
| 7.5 颜色索引模式下指定颜色实例过程 | 145 |
| 7.5.1 在颜色索引模式下指定颜色 | 145 |
| 7.5.2 包含文件及全局变量 | 146 |
| 7.5.3 绘制圆环 | 147 |
| 7.5.4 设置投影方式 | 149 |
| 7.5.5 初始化 OpenGL 环境 | 149 |
| 7.5.6 变换后重绘圆环 | 150 |
| 7.5.7 窗口改变大小 | 150 |
| 7.5.8 设置逻辑调色板 | 151 |
| 7.5.9 设置像素格式 | 153 |
| 7.5.10 主窗口事件处理函数 | 154 |
| 7.5.11 主函数 | 156 |
| 7.5.12 程序执行结果 | 158 |
| 第 8 章 光照高级编程 | 159 |
| 8.1 真实世界与 OpenGL 中的光照 | 159 |
| 8.1.1 光照分量 | 160 |
| 8.1.2 材质颜色 | 160 |
| 8.1.3 光线与材质的 RGB 值 | 161 |
| 8.2 简单示例 | 161 |
| 8.3 创建光源 | 163 |
| 8.3.1 颜色 | 164 |
| 8.3.2 位置与衰减 | 165 |
| 8.3.3 聚光灯 | 165 |
| 8.3.4 多光源 | 166 |
| 8.3.5 控制光源位置和方向 | 167 |
| 8.4 选择光照模型 | 171 |
| 8.4.1 全局环境光 | 171 |
| 8.4.2 局部和无穷远视点 | 171 |
| 8.4.3 双面光照 | 172 |

| | |
|------------------------------------|------------|
| 8.4.4 启用光照 | 172 |
| 8.5 定义材质特性 | 172 |
| 8.5.1 漫射和环境反射 | 173 |
| 8.5.2 镜面反射 | 174 |
| 8.5.3 发射光 | 174 |
| 8.5.4 改变材质特性 | 174 |
| 8.6 光照中的数学 | 179 |
| 8.6.1 材质发射光 | 179 |
| 8.6.2 放缩的全局环境光 | 179 |
| 8.6.3 光源的贡献 | 179 |
| 8.6.4 合成 | 181 |
| 8.7 颜色索引模式中的光照 | 181 |
| 8.8 发光小球碰撞实验 | 182 |
| 第 9 章 读取三维图形数据 | 194 |
| 9.1 三维图形数据文件格式简介 | 194 |
| 9.1.1 Wavefront 模型简介 | 194 |
| 9.1.2 3DS 文件格式 | 196 |
| 9.1.3 各种 3D 数据模型相互转换 | 203 |
| 9.2 在 OpenGL 中读取和操作 3DS 模型文件 | 203 |
| 第 10 章 位图、字体和图像编程 | 236 |
| 10.1 位图和字体 | 236 |
| 10.1.1 OpenGL 对字符显示的支持 | 236 |
| 10.1.2 当前光栅位置 | 238 |
| 10.1.3 绘制位图 | 238 |
| 10.1.4 字体和显示列表 | 239 |
| 10.1.5 定义和使用一个完整的字体 | 240 |
| 10.2 OpenGL 中生成图像的打印 | 247 |
| 10.2.1 OpenGL 中图像的绘制 | 247 |
| 10.2.2 OpenGL 中生成图像打印的基本步骤 | 251 |
| 10.2.3 实例过程 | 251 |
| 10.2.4 程序执行结果 | 260 |
| 第 11 章 纹理映射高级编程 | 261 |
| 11.1 定义纹理 | 261 |
| 11.2 控制纹理滤波 | 262 |
| 11.2.1 滤波 | 262 |
| 11.2.2 重复与缩限 | 263 |
| 11.2.3 映射方式 | 263 |

| | |
|--------------------------------------|------------|
| 11.2.4 纹理坐标定义 | 263 |
| 11.2.5 自动产生纹理坐标 | 263 |
| 11.3 用位图文件创建纹理 | 270 |
| 11.3.1 调用函数来创建纹理 | 270 |
| 11.3.2 从文件中直接读写 | 272 |
| 第 12 章 各种特殊效果的实现 | 273 |
| 12.1 融合 | 273 |
| 12.1.1 源和目的因子 | 274 |
| 12.1.2 颜色融合的应用 | 275 |
| 12.1.3 一个颜色融合的例子 | 275 |
| 12.1.4 利用深度缓存的三维融合 | 277 |
| 12.2 反走样 | 283 |
| 12.2.1 反走样点或线 | 285 |
| 12.2.2 反走样多边形 | 289 |
| 12.3 雾化 | 292 |
| 12.3.1 雾的用法 | 292 |
| 12.3.2 雾方程 | 295 |
| 第 13 章 帧缓存技术与 OpenGL 动画 | 299 |
| 13.1 缓存及其使用 | 299 |
| 13.1.1 OpenGL 中的各种缓存 | 300 |
| 13.1.2 清除缓存 | 301 |
| 13.1.3 选择绘图的颜色缓存 | 302 |
| 13.1.4 屏蔽缓存 | 303 |
| 13.2 检验和操作片断值 | 303 |
| 13.2.1 剪裁检验 | 304 |
| 13.2.2 a 检验 | 304 |
| 13.2.3 模板检验 | 305 |
| 13.2.4 深度检验 | 320 |
| 13.2.5 融合、抖动和逻辑操作 | 321 |
| 13.3 累积缓存 | 322 |
| 第 14 章 巧用显示列表 | 324 |
| 14.1 显示列表的基本概念 | 324 |
| 14.2 显示列表的建立、执行和管理 | 327 |
| 14.2.1 显示列表的建立 | 327 |
| 14.2.2 显示列表中的存储内容 | 329 |
| 14.2.3 执行显示列表 | 330 |
| 14.2.4 显示列表的层次 | 332 |

| | |
|---------------------------------------|------------|
| 14.2.5 管理显示列表及其索引 | 333 |
| 14.3 改变封装方式的显示列表 | 338 |
| 第 15 章 参数曲面和曲线编程 | 340 |
| 第 16 章 选择与反馈实现图形交互 | 350 |
| 16.1 选择模式 | 350 |
| 16.1.1 选择的基本步骤 | 351 |
| 16.1.2 建立名字栈 | 352 |
| 16.1.3 命中记录 | 352 |
| 16.1.4 选择的例子 | 353 |
| 16.1.5 拾取 | 353 |
| 16.1.6 利用选择编程的提示 | 354 |
| 16.2 反馈模式 | 355 |
| 16.2.1 反馈数组 | 356 |
| 16.2.2 在反馈模式下利用标记 | 357 |
| 16.2.3 反馈的例子 | 357 |
| 16.2.4 命中记录 | 359 |
| 16.3 一个选择的例子 | 359 |
| 16.4 拾取 | 363 |
| 16.4.1 简单示例 | 363 |
| 16.4.2 从多个名称和一个层次结构中进行拾取 | 367 |
| 第 17 章 用 OpenGL 编写屏幕保护程序 | 373 |
| 17.1 Windows 环境下的屏幕保护程序 | 373 |
| 17.1.1 设置屏幕保护程序 | 373 |
| 17.1.2 安装屏幕保护程序 | 375 |
| 17.2 屏幕保护程序编写的基本步骤 | 376 |
| 17.2.1 屏幕保护响应程序 | 376 |
| 17.2.2 屏幕保护设置程序 | 382 |
| 17.2.3 图形绘制程序 | 382 |
| 17.3 屏幕保护程序实例 | 384 |
| 17.3.1 屏幕保护程序主程序 | 384 |
| 17.3.2 图形绘制程序 | 386 |
| 第 18 章 建筑物虚拟漫游编程 | 394 |
| 18.1 虚拟现实系统简介 | 394 |
| 18.1.1 虚拟现实技术的产生、基本特征及构成 | 394 |
| 18.1.2 多维信息空间 | 395 |
| 18.1.3 实现虚拟现实的软硬件平台 | 396 |

| | |
|--------------------------------------|------------|
| 18.1.4 虚拟环境人机交互系统 | 397 |
| 18.2 虚拟漫游技术简介 | 398 |
| 18.2.1 虚拟漫游系统示例 | 398 |
| 18.2.2 建筑物虚拟漫游 | 399 |
| 18.3 虚拟建筑物漫游编程实例 | 400 |
| 18.3.1 辐射度方法简介 | 400 |
| 18.3.2 数据存储结构 | 400 |
| 18.3.3 读取辐射度数据 | 402 |
| 18.3.4 纹理映射相关程序 | 404 |
| 18.3.5 绘制程序 | 407 |
| 18.3.6 漫游动作控制 | 409 |
| 第 19 章 OpenGL 新版本的新增功能 | 412 |
| 19.1 新增纹理处理功能 | 412 |
| 19.1.1 新增纹理格式 | 412 |
| 19.1.2 使用纹理环境的新方法 | 413 |
| 19.1.3 纹理代理机制 | 415 |
| 19.1.4 使用纹理物体 | 418 |
| 19.1.5 使用子纹理 | 421 |
| 19.1.6 从帧缓存中装入纹理 | 422 |
| 19.1.7 其他新的纹理扩展函数 | 423 |
| 19.2 多边形偏移功能 | 424 |
| 19.2.1 使用多边形偏移功能 | 424 |
| 19.2.2 多边形偏移编程示例 | 425 |
| 19.3 OpenGL 新版本顶点数组 | 428 |
| 19.3.1 指定顶点数组 | 428 |
| 19.3.2 变换独立的数组单元 | 430 |
| 19.3.3 连接不同数组的数组单元 | 430 |
| 19.3.4 通过建立顶点网格绘制实体 | 431 |
| 19.3.5 使用隔行数组 | 431 |
| 19.3.6 顶点数组扩展功能 | 432 |
| 第 20 章 OpenGL 扩展功能及程序改进 | 434 |
| 20.1 Win32 环境下 OpenGL 扩展集 | 434 |
| 20.1.1 确定扩展集可用 | 434 |
| 20.1.2 编译顶点数组扩展集 | 437 |
| 20.1.3 物体空间顶点拾取扩展集 | 438 |
| 20.1.4 调色板纹理扩展集 | 439 |
| 20.1.5 BGRA 扩展集 | 440 |

| | | |
|--------|------------------------------------|-----|
| 20.1.6 | 索引纹理扩展集 | 441 |
| 20.1.7 | 索引材质扩展集 | 442 |
| 20.1.8 | 索引功能扩展集 | 442 |
| 20.1.9 | 索引数组格式扩展集 | 443 |
| 20.2 | OpenGL 程序性能的改进 | 444 |
| 20.2.1 | 改进程序流程 | 444 |
| 20.2.2 | 改进几何体子系统 | 446 |
| 20.2.3 | 改进栅格化子系统 | 449 |
| 20.2.4 | 提高图像和像素绘制性能 | 451 |
| 20.2.5 | 改进动画性能 | 453 |
| 20.2.6 | 进行时间测定 | 454 |
| 20.2.7 | SGI OpenGL for Windows 库使用技巧 | 455 |

第 1 章 OpenGL 编程基础知识

随着计算机软、硬件突飞猛进的发展，计算机图形学在各个行业的应用也得到迅速普及和深入。目前，计算机图形学已进入三维时代，三维图形在人们周围无所不在。魅力无穷的计算机广告，扣人心弦的三维游戏，声势浩大、震撼人心的影视特效，无一不是三维图形大显身手的得意之作。

科学可视化、计算机动画和虚拟现实已成为近年来计算机图形学的三大热门话题，而这三大热门话题的技术核心均为三维图形。当前，三维图形已在军事、航空、航天、医学、地质勘探、文化艺术和艺术造型等方面有着十分广泛的应用。

OpenGL 是一种与硬件、窗口系统和操作系统相独立的一系列 API，它可以被集成到 UNIX, Windows 95, Windows NT 和 X Window 等窗口系统中。OpenGL 由大量功能强大的图形函数组成，开发人员可以利用这些函数对整个三维图形进行光色渲染，从而方便地绘制出客观世界逼真的三维景象。

以下先简要介绍图形学的基本发展状况，然后通过一个简单的示例向读者展示 OpenGL 的诱人魅力，衷心希望本书能抛砖引玉，引导读者在 OpenGL 的广阔天地中深入探索。

1.1 图形学的基本发展状况

1.1.1 蓬勃发展的计算机图形学

计算机图形学是一个最令人振奋且发展迅速的计算机领域。计算机图形学的发展历史可以追溯到 50 年代末至 60 年代初期。当时的计算机主要应用于科学计算，使用尚不普及，但已开始配置了图形显示器。例如，美国麻省理工学院旋风计算机就配置了由计算机驱动的阴极射线管式的显示器，但不具备人机交互能力。50 年代末期，美国麻省理工学院林肯实验室研制的 SAGE 空中防御系统，就已具备了指挥和控制功能。这个系统能将雷达信号转换为显示器上的图形，操作者可以借用光笔指向屏幕上的目标图形来获得所需要的信息，这一功能的出现预示着交互式图形生成技术的诞生。

1963 年，美国麻省理工学院的 I.E.Sutherland 在他的博士论文中提出了 SKETCHPAD 系统。这一系统用的是 TX.2 型计算机及阴极射线管式图形显示器，它可用光笔在图形显示器上实现选择、定位等交互功能。计算机还可以跟踪光笔，从原来所在的点到所指定的点画直线，或在给定圆心和半径后画圆等。这一系统还引入了分层存储符号和图形元素的数据结构，一幅完整的图可以通过分层调用若干子图来产生。这些基本概念和技术至今仍有用。因此，I.E.Sutherland 的 SKETCHPAD 系统被公认为对交互式图形生成技术的发展

奠定了基础。

在 60 年代, 计算机主要以大型机的形式出现, 大多数用于科学计算及事务管理。因而以大型计算机连接图形终端成为计算机图形生成技术的运行环境。进入 70 年代以后, 由于集成电路技术的发展, 计算机硬件性能不断提高、体积缩小、价格降低, 特别是廉价的图形输入、输出设备及大容量磁盘的出现, 以小型计算机及超级小型机为基础的图形生成技术开始进入市场并形成主流。由于这种系统比起大型计算机来, 价格相对便宜, 维护使用也比较简单。因而, 70 年代以来, 计算机图形生成技术在计算机辅助设计、事务管理和过程控制等领域得到了比较广泛的应用。

进入 80 年代以后, 工作站的出现, 极大地促进了计算机图形学的发展。比起小型计算机来, 工作站在图形生成上具有显著的优点。首先, 工作站是一个用户使用一台计算机交互作用时, 响应时间短; 其次, 工作站联网后可以共享资源, 如, 大容量磁盘、高精度绘图仪等; 再有, 它便于逐步投资、逐步发展、使用寿命较长。因而工作站已经取代小型计算机而成为图形生成的主要环境。目前, 工作站正朝着提高速度、扩大容量、提高分辨率、具有标准化的友好用户界面以及多媒体等方向发展。

80 年代后期至今, 微机的性能也迅速提高, 无论是整数运算速度、内存容量还是磁盘容量, 都可以和 workstation 相媲美, 而且大多数微机都配备了高分辨率显示器及窗口管理系统, 因此微机也已经成为图形生成的重要环境。由于微机价格比 workstation 更加便宜, 因而受到广大用户的欢迎, 更易于普及和推广。随着微机性能的进一步提高, 可以预计, 它将在计算机图形技术的应用中扮演更为重要的角色。

1.1.2 计算机图形学的广泛应用

计算机图形系统的软硬件性能日益提高, 而价格却逐步降低, 这必然促使计算机图形生成技术的应用日益广泛。当前计算机图形学已在航空、航天、汽车制造、电子、机械、影视广告、地理信息和数据可视化等领域取得了广泛的应用。

计算机图形学主要应用在设计方面, 尤其是工程和建筑系统, 现在几乎所有的产品都是由计算机来设计。计算机辅助设计(CAD)方法现在已频繁地应用于大楼、汽车、飞机、轮船、宇宙飞船、计算机、纺织品和许许多多其他产品的设计中。对某些设计来说, 设计对象以线框图形式显示出来, 以展示出其整个外形及该对象的内部特征。显示线框图可让设计者较快地看到对设计的外形交互修改的结果。CAD 的应用还通常使用动画, 如果不需要表面绘制, 则可以较快地生成平稳的实时动画。在虚拟现实环境中使用动画确定飞行器的内部结构以及运动时内部构件的行为。在设计完成或将要完成的时候, 应用真实感光照模型和曲面绘制技术可给出最终产品的外貌。制造过程也和设计对象的计算机描述联系起来以提供产品的自动化程度。例如, 电路板布局可转换为构造该布局的各个过程的描述。某些机械零件按照如何使用机床加工曲面的描述来制造。建筑设计师使用交互式图形技术进行平面布局, 给出房间、门、窗、楼梯、橱和柜等的位置及其他特征。面对屏幕上显示的大楼平面布局图, 电器设计师可以设计如何布线, 安装电源插座以及防火报警系统等。

计算机图形学广泛地应用于美术和商业艺术产品中。美术师使用各种计算机方法, 包括专用硬件、艺术家画笔程序和其他绘图软件包和动画软件包等来设计物体的外形及描述

物体的运动。优秀的美术师可以使用各种计算机技术生成图像，他们可以混合使用各种建模软件包、绘图软件和 CAD 软件等。许多商业美术应用(动画及影视广告等)使用逼真照片绘制产品的图片。广告中经常使用动画，电视广告是通过一帧一帧地绘制并存储在一个文件中来生成的。运动的模拟通过在后继帧中的物体相对于前一帧作微小的位移来实现。绘制好动画序列的所有帧以后，将这些帧传送到胶片上或存到视频缓存中重播。电影动画需要顺序播放 24 帧每分，如果在视频上重播动画则需要 30 帧每分。

现在计算机图形学方法常用于制作动画、音乐录像带和电视片。图形场景有时单独显示，有时则和演员及实际场景混合显示。许多电视连续剧中经常使用计算机图形方法。音乐录像片以几种不同方法来使用图形学。图形对象可以被混合进实景中，图形学中的变形技术可被用来将一个人或对象变成另一个。

科学家、工程师、药剂师、商业分析员等常要分析大量的信息或研究特定处理的行为。在巨型机上进行数值计算不断生成包含成千上万数据项的数据文件。同样地，卫星摄像机等也以比解释速度快得多的速度积聚大量数据文件。扫描众多数据以确定趋势及相互关系是一个乏味而低效的过程，但是，如果将这些数据转换为可视形式，则趋势和模式就可以立即显现。存在许多不同类型的数据集，而高效的可视化方法则依赖于数据的特征。一组数据可以包含标量、向量、高次张量或这些数据类型的组合。

尽管在计算机图形学和图像处理中所使用的技术有重叠，但两者有不同的基本操作。在计算机图形学中使用计算机来生成图。而图像处理使用有关技术修改或解释现有图片，比如照片和电视扫描片。图像处理的两个主要应用是：改善图片质量和对视觉信息的机械感觉。

应用计算机图形生成技术产生高精度的地理图形或其他自然资源的图形是另一个重要的应用领域。包括地理图、地形图、矿藏分布图、海洋地理图、气象图和植被分布图等。目前，地理信息管理系统已在很多方面得到广泛的应用，它就是建立在地理图形基础上的信息管理系统。

随着微型计算机及桌上印刷设备的发展，计算机图形学及人机交互技术在办公自动化及电子出版系统中得到广泛的应用。昔日需要提交给专门的印刷机构出版的资料，现在可以在办公室内印刷了。

国外资料表明，在 21 世纪，计算机图形生成技术的应用仍将不断发展，计算机图形系统的市场将以每年 15%~20% 的递增速度不断扩大。

1.2 OpenGL 概述

为了用计算机模拟人们生活的三维世界，必须在三维空间中描绘这些物体。对于这个充满了信息的世界，人们渴望尽快地理解并运用这些信息，因此需要一种直接的、易于理解的形式来表达这些信息，计算机图形学在这方面起着举足轻重的作用。许多人在这方面作出了卓越的贡献，例如，长期以来，从事三维图形开发的技术人员都编写了大量的诸如矩阵变换、外部设备访问等函数。然而，他们做的很多工作与自己的目标关系都不是很大，在这方面费脑筋着实是极大的脑力浪费。OpenGL 提供了一种直观的编程环境，它提供的