



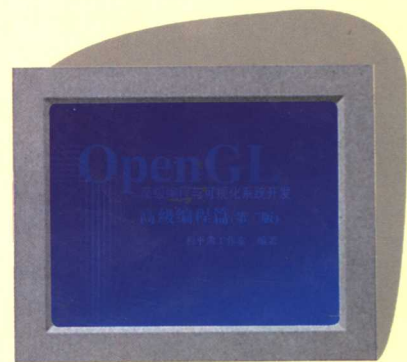
万水计算机编程技术与应用系列

OpenGL

高级编程与可视化系统开发

高级编程篇(第二版)

和平鸽工作室 编著



中国水利水电出版社
www.waterpub.com.cn

万水计算机编程技术与应用系列

OpenGL 高级编程与可视化系统开发

——高级编程篇

(第二版)

和平鸽工作室 编著

中国水利水电出版社

内 容 提 要

OpenGL 是最近几年发展起来的一个性能卓越的开放式三维图形标准,利用它可以创作出具有照片质量的、独立于窗口系统、操作系统和硬件平台的三维彩色图形和动画。目前,OpenGL 在可视化系统、虚拟现实系统和三维游戏方面得到了广泛应用。

本书及其姊妹篇的主要内容是介绍如何利用 OpenGL 图形库开发可视化应用系统,主要以实例的方式讲述一些特殊可视化效果的编程方法,包括地形、云彩、海浪、烟雾、粒子系统、飞行器尾焰等,掌握这些特效的制作对于开发高逼真度图形非常重要。

本书姊妹篇《OpenGL 高级编程与可视化系统开发——系统开发篇(第二版)》介绍了多个可视化仿真实例,包括 1 个场景编辑系统、2 个三维游戏和 5 个可视化仿真项目。充分利用这些源代码和编程思想,可以大大节省读者开发可视化仿真项目的时间。

本书深入浅出、内容广泛,可供从事可视化系统开发、三维游戏开发或其他图形应用程序开发的各大专院校学生、教员和研究人员参考,也可作为 OpenGL 三维图形编程的培训教程以及其他相关专业人士和计算机爱好者阅读。

本书所附光盘包含书中全部应用实例的源代码。

图书在版编目(CIP)数据

OpenGL 高级编程与可视化系统开发. 高级编程篇 / 和平鸽工作室编著. —2 版. —北京: 中国水利水电出版社, 2005

(万水计算机编程技术与应用系列)

ISBN 7-5084-3335-1

I. O... II. 和... III. 图形软件, OpenGL IV. TP391.41

中国版本图书馆 CIP 数据核字(2005)第 119323 号

书 名	OpenGL 高级编程与可视化系统开发——高级编程篇(第二版)
作 者	和平鸽工作室 编著
出版 发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net(万水) sales@waterpub.com.cn
经 售	电话: (010) 63202266(总机)、68331835(营销中心)、82562819(万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 28.5 印张 702 千字
版 次	2003 年 1 月第 1 版 2006 年 1 月第 2 版 2006 年 1 月第 2 次印刷
印 数	4001—8000 册
定 价	58.00 元(含 1CD)

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

第一版前言

科学可视化、计算机动画和虚拟现实是近年来在计算机图形学领域内的三大热门研究方向，它们的技术核心都是三维真实感图形。当前，三维图形在军事、航天、航空、医学、地质勘探、三维游戏和工业 CAD 设计等方面有着十分广泛地应用。

OpenGL（即开放性图形库 Open Graphics Library）是近几年发展起来的一个性能卓越的三维图形标准，它源于 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成为 OpenGL。SGI 公司在 1992 年 7 月发布 1.0 版，后成为工业标准。1995 年 12 月批准了 1.1 版本，最新版规范是 1999 年 5 月通过的 1.2.1。OpenGL 作为一个性能优越的图形应用程序设计界面（API），具有广泛的可移植性，它独立于硬件系统、操作系统和窗口系统。OpenGL 适用于广泛的计算机环境，从个人计算机到工作站和超级计算机，用户都可以利用 OpenGL 创建漂亮的三维图形。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，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开发工具。

由于 OpenGL 能实现高性能的三维图形功能，这使得科学计算可视化、仿真可视化技术发展迅速。可视化系统不仅可以帮助技术人员形象直观地查看仿真结果，而且在一定程度上可以提高设计水平，减少错误的发生。因此计算机图形技术将在各行业、各个领域内得到越来越广泛地应用。有人说 20 世纪是计算机多媒体技术发展的世纪，21 世纪将是虚拟现实飞速发展的世纪，因此，积极推动计算机图形领域的发展对于我国的经济建设、科学研究具有重要的现实意义。

本套丛书主要介绍如何利用 OpenGL 开发可视化应用系统，它定位于 OpenGL 的高级编程和可视化系统开发，在书中并不对 VC++ 开发和 OpenGL 的基础知识进行讲解。因此要求阅读本套丛书的读者必须具有使用 VC++ 和 OpenGL 进行编程的基础知识。

《OpenGL 高级编程与可视化系统开发（高级编程篇）》共由 15 章组成。第 1 章至第 2 章简要地介绍了 OpenGL 与图形可视化的关系以及 OpenGL 编程的基础知识；第 3 章介绍了利用 OpenGL 读取各种外部三维模型的方法，如 3DS、OBJ、WRL 等；第 4 章至第 7 章介绍了三维地形、蓝天白云、星空和水的波动的模拟；第 8 章介绍了粒子系统的编程方法；第 9 章至第 10 章介绍了碰撞检测编程、LOD 编程技术；第 11 章至第 15 章介绍了一些特殊效果的编程方法，如爆炸效果、霓虹效果、对象捕捉和自然现象的编程。

《OpenGL 高级编程与可视化系统开发（系统开发篇）》共由 16 章组成。第 1 章至第 11 章全面系统地介绍了一个三维场景编辑软件系统的编程实现过程，该软件系统结构清晰、功能齐全、可扩充性强，对于开发应用系统具有很高的借鉴价值；第 12 章介绍了一个三维空战游戏的开发过程；第 13 章介绍了一个利用 OpenGL 开发的游戏引擎和一个利用该引擎开发的三维射击游戏“恐怖之战”；第 14 章至第 16 章介绍了三个 OpenGL 开发的可视

化仿真实例，包括导弹飞行试验系统、飞行漫游系统、飞机飞行动力学系统。

为了便于用户使用，在书后附有包含该书全部应用实例的源代码光盘。全部程序在 Visual C++ 6.0 中编译通过，运行环境为 Windows 98 或 Windows 2000。读者可以充分利用其中的代码，有些只需稍许改动便可应用到自己的系统中，这些都是笔者多年的积累，以供读者参考。

本书由和平鸽工作室全体成员齐心协力编写完成，是大家精诚团结才使本书顺利完成。感谢对本书的出版提供支持的所有老师、编辑。由于水平有限，书中难免存在疏漏和错误之处，敬请各位读者朋友批评指正。

和平鸽工作室

2002 年 5 月

第二版前言

科学计算可视化、计算机动画和虚拟现实是近年来在计算机图形学领域内的三大热门研究方向，它们的技术核心都是三维真实感图形。当前，三维图形在军事、航天、航空、医学、地质勘探、三维游戏和工业 CAD 设计等方面有着十分广泛的应用。

OpenGL（即开放性图形库 Open Graphics Library）是近几年发展起来的一个性能卓越的三维图形标准，它源于 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成为 OpenGL。SGI 在 1992 年 7 月发布 1.0 版，后成为工业标准。1995 年 12 月批准了 1.1 版本，1999 年 5 月通过了 1.2.1 版本，现在的最新版本是 2.0 版。OpenGL 作为一个性能优越的图形应用程序设计界面（API），具有广泛的可移植性，它独立于硬件系统、操作系统和窗口系统。OpenGL 适用于广泛的计算机环境，从个人计算机到工作站和超级计算机，用户都可以利用 OpenGL 创建漂亮的三维图形。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，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开发工具。

由于 OpenGL 能实现高性能的三维图形功能，这使得科学计算可视化、仿真可视化技术发展迅速。可视化系统不仅可以帮助技术人员形象直观的查看仿真结果，而且在一定程度上可以提高设计水平，减少错误的发生。因此计算机图形技术将在各行各业、各个领域内得到越来越广泛地应用。有人说 20 世纪是计算机多媒体技术发展的世纪，21 世纪将是虚拟现实飞速发展的世纪，因此，积极推动计算机图形领域的发展将对我国的经济建设、科学研究具有重要的现实意义。

丛书特色

本套丛书在第一版的基础上经过修订后内容更加全面，更加紧密结合发展前沿，增加的源代码更加具有实用性、趣味性、新颖性。与第一版相比，第二版主要具有以下特色。

● 更加紧扣前沿

全面介绍了 OpenGL 的发展历史、目前的发展现状和今后 OpenGL 2.0 版本的发展方向，使读者更加了解 OpenGL 的来龙去脉。其中还详细讲解了不同版本的 OpenGL 的体系结构、实现方法和函数库组成等。

● 应用更加广泛

为了适应不同读者开发的需要，丛书第二版在第一版的基础上增加了 Visual Basic、Fortran 和 Delphi 编程语言开发 OpenGL 三维图形程序的框架。同时对第一版的 Visual C++ 开发 OpenGL 三维程序模本增加了更为详细的讲解。

● 实例更加新颖

OpenGL 在 3D 屏幕保护程序开发中的应用非常广泛，在第二版的“高级编程篇”中特意增加了一章这方面的内容。紧密联系研究热点，在第二版的“系统开发篇”中增加了三

维地形模拟系统和三维织物模拟系统。书中详细介绍每一个实例系统开发的理论基础、模拟方法以及编程过程,使读者可以轻松地掌握其中的内容。

● 源码更加丰富

为了便于用户使用,随本书附有包含该书全部应用实例的源代码光盘。全部程序均编译通过,运行环境为 Windows 9x 和 Windows 2000/XP。读者可以充分利用其中的代码,有些只需稍许改动便可应用到自己的系统中,这些都是笔者多年的积累,非常珍贵,非常有用。

主要内容

《OpenGL 高级编程与可视化系统开发——高级编程篇(第二版)》共由 16 章组成。第 1 章至第 2 章简要地介绍了 OpenGL 与图形可视化的关系以及 OpenGL 编程的基础知识;第 3 章介绍了利用 OpenGL 读取各种外部三维模型的方法,如 3DS、OBJ、WRL 等;第 4 章至第 7 章介绍了三维地形、蓝天白云、星空和水的波动的模拟;第 8 章介绍了粒子系统的编程方法;第 9 章至第 10 章介绍了碰撞检测编程、LOD 编程技术;第 11 章至第 15 章介绍了一些特殊效果的编程方法,如爆炸效果、霓虹效果、对象捕捉和自然现象模拟的编程;第 16 章介绍了如何使用 OpenGL 开发 3D 屏幕保护程序。

《OpenGL 高级编程与可视化系统开发——系统开发篇(第二版)》共由 18 章组成。第 1 章至第 11 章全面系统地介绍了一个三维场景编辑软件系统的编程实现过程,该软件系统结构清晰、功能齐全、可扩充性强,对于开发应用系统具有很高的借鉴价值;第 12 章介绍了一个三维空战游戏的开发过程;第 13 章介绍了一个利用 OpenGL 开发的游戏引擎和一个利用该引擎开发的三维射击游戏“恐怖之战”;第 14 章至第 18 章介绍了 5 个 OpenGL 开发的可视化仿真实例,包括导弹飞行试验系统、飞行漫游系统、飞机飞行动力学系统、三维随机分形地形生成系统和三维织物动感模拟系统。

读者要求

本套丛书主要介绍如何利用 OpenGL 开发可视化应用系统,它定位于 OpenGL 的高级编程和可视化系统开发,在书中并不对 VC++ 开发和 OpenGL 的基础知识进行讲解。因此要求阅读本套丛书的读者必须具有使用 VC++ 和 OpenGL 进行编程的基础知识。

本书由和平鸽工作室全体成员齐心协力编写完成,是大家精诚团结才使本书顺利完成。参加本书编写工作的人员有李海平、李贞、李雪梅、张冰、刘一平、谢平、郭健康、牛威、张健、秦一凡、张建国、衣裘、郝冰、李惠彬、周大志、刘文涛、王瑞雪、张志刚、袁世杰、董慧、龙亦敏、韩鹏程、孟子刚、张宏斌、杨海涛、康敏、刘志强、李娟、陈爱华、周伟、杨小伟、曾国强、李卫东、张慧、李恒、刘国建、谢凯旋、张立国、刘立群、牛兴旺等。感谢对本书的出版提供支持的所有老师、编辑。

由于水平有限,书中难免存在疏漏和错误之处,敬请各位读者朋友批评指正。读者如有疑问、意见或建议,请发邮件至 pingtaizheng@tom.com 或 liyuhua1973@yahoo.com。

和平鸽工作室

2005 年 10 月

目 录

第二版前言

第一版前言

第 1 章 OpenGL 与图形可视化概述	1	2.1.9 帧缓存与动画	27
1.1 图形可视化概述	1	2.2 基于对话框的 OpenGL 图形程序的	
1.2 OpenGL 的发展	1	基本框架	29
1.2.1 关于 OpenGL	1	2.2.1 程序说明	29
1.2.2 OpenGL 1.X 回顾	2	2.2.2 理论基础	29
1.2.3 从 1.X 到 2.0	3	2.2.3 编程步骤	30
1.3 OpenGL 2.0 的新技术	3	2.2.4 加入到 AppWizard	36
1.3.1 OpenGL 2.0 由来	3	2.3 基于单文档的 OpenGL 图形程序的	
1.3.2 OpenGL 2.0 特点与新技术	4	基本框架	37
1.4 OpenGL 基本操作	6	2.3.1 程序说明	37
1.5 OpenGL 图形的实现	7	2.3.2 理论基础	38
1.5.1 OpenGL 体系结构	7	2.3.3 编程步骤	38
1.5.2 渲染上下文 (RC)	8	2.3.4 加入到 AppWizard	42
1.5.3 调色板的使用	10	2.4 基于多文档的 OpenGL 图形程序的	
1.5.4 像素格式设置	11	基本框架	42
1.5.5 OpenGL 处理流程与图形		2.4.1 程序说明	42
绘制方式	14	2.4.2 理论基础	43
1.6 OpenGL 开发库的基本组成	15	2.4.3 编程步骤	43
1.6.1 OpenGL 开发组件	15	2.4.4 加入到 AppWizard	50
1.6.2 OpenGL 函数	16	2.5 Visual Basic 语言开发 OpenGL 程序的	
1.7 OpenGL 程序运行方式	17	基本框架	51
第 2 章 OpenGL 编程基础	18	2.5.1 程序说明	51
2.1 OpenGL 绘图基本知识	18	2.5.2 添加 OpenGL 引用	51
2.1.1 基本数据类型	18	2.5.3 OpenGL 窗口的初始化	52
2.1.2 图形绘制	19	2.5.4 OpenGL 图形的绘制	53
2.1.3 显示列表	20	2.6 Fortran 语言开发 OpenGL 程序的	
2.1.4 OpenGL 颜色	22	基本框架	55
2.1.5 OpenGL 光照模型	23	2.6.1 程序说明	55
2.1.6 材质设置	24	2.6.2 添加 OpenGL 开发库	55
2.1.7 纹理映射	25	2.6.3 OpenGL 窗口的初始化	55
2.1.8 选择与反馈	26	2.6.4 OpenGL 图形的绘制	59

2.7 Delphi 语言开发 OpenGL 程序的 基本框架.....	61	5.1.2 理论基础.....	165
2.7.1 程序说明.....	61	5.1.3 编程步骤.....	166
2.7.2 添加 OpenGL 开发库.....	61	5.2 模拟实例二.....	170
2.7.3 OpenGL 窗口的初始化.....	61	5.2.1 程序说明.....	170
2.7.4 OpenGL 图形的绘制.....	65	5.2.2 理论基础.....	170
第 3 章 外部三维图形文件的输入与处理.....	68	5.2.3 编程步骤.....	174
3.1 3DS 文件格式.....	68	第 6 章 星空的模拟.....	180
3.1.1 3DS 文件格式简介.....	68	6.1 模拟实例一.....	180
3.1.2 3D 编辑程序块.....	70	6.1.1 程序说明.....	180
3.1.3 关键帧块.....	74	6.1.2 理论基础.....	180
3.2 3DS 文件输入程序介绍.....	75	6.1.3 编程步骤.....	182
3.2.1 程序说明.....	75	6.2 模拟实例二.....	188
3.2.2 理论基础.....	75	6.2.1 程序说明.....	188
3.2.3 编程步骤.....	77	6.2.2 理论基础.....	189
3.3 OBJ 文件格式介绍.....	107	6.2.3 编程步骤.....	190
3.4 OBJ 文件输入程序介绍.....	111	第 7 章 水的波动的模拟.....	200
3.4.1 程序说明.....	111	7.1 编程实例一.....	200
3.4.2 理论基础.....	112	7.1.1 程序说明.....	200
3.4.3 编程步骤.....	113	7.1.2 理论基础.....	200
3.5 其他格式 3D 文件的处理方法.....	133	7.1.3 编程步骤.....	202
3.5.1 各种格式的 3D 文件的相互转换.....	133	7.2 编程实例二.....	206
3.5.2 WRL 文件的读入.....	134	7.2.1 程序说明.....	206
3.5.3 ASE 文件的读入.....	136	7.2.2 理论基础.....	206
3.5.4 3DS 转换成 OpenGL 文件的 读入.....	138	7.2.3 编程步骤.....	207
第 4 章 三维地形的模拟.....	140	第 8 章 粒子系统的模拟.....	215
4.1 地形模拟概述.....	140	8.1 粒子系统概述.....	215
4.2 编程实例一.....	141	8.1.1 过程模型.....	215
4.2.1 程序说明.....	141	8.1.2 粒子系统.....	215
4.2.2 理论基础.....	141	8.2 编程实例一.....	216
4.2.3 编程步骤.....	142	8.2.1 程序说明.....	216
4.3 编程实例二.....	156	8.2.2 理论基础.....	217
4.3.1 程序说明.....	156	8.2.3 编程步骤.....	218
4.3.2 理论基础.....	156	8.3 编程实例二.....	224
4.3.3 编程步骤.....	159	8.3.1 程序说明.....	224
第 5 章 蓝天白云的模拟.....	164	8.3.2 理论基础.....	225
5.1 模拟实例一.....	164	8.3.3 编程步骤.....	227
5.1.1 程序说明.....	164	第 9 章 碰撞检测编程.....	237
		9.1 碰撞检测概述.....	237
		9.2 编程实例一.....	238

9.2.1 程序说明	238	13.2:1 程序说明	367
9.2.2 理论基础	239	13.2.2 理论基础	368
9.2.3 编程步骤	241	13.2.3 编程步骤	370
9.3 编程实例二	270	13.3 编程实例三	376
9.3.1 程序说明	270	13.3.1 程序说明	376
9.3.2 理论基础	271	13.3.2 理论基础	377
9.3.3 编程步骤	271	13.3.3 编程步骤	378
第 10 章 LOD 技术编程	277	第 14 章 对象捕捉的编程	390
10.1 LOD 技术概述	277	14.1 编程实例一	390
10.2 编程实例一	280	14.1.1 程序说明	390
10.2.1 程序说明	280	14.1.2 理论基础	390
10.2.2 理论基础	280	14.1.3 编程步骤	392
10.2.3 编程步骤	281	14.2 编程实例二	408
第 11 章 火焰的模拟	309	14.2.1 程序说明	408
11.1 编程实例一	309	14.2.2 理论基础	408
11.1.1 程序说明	309	14.2.3 编程步骤	410
11.1.2 理论基础	309	第 15 章 自然现象的模拟	414
11.1.3 编程步骤	310	15.1 编程实例一	414
11.2 编程实例二	314	15.1.1 程序说明	414
11.2.1 程序说明	314	15.1.2 理论基础	414
11.2.2 理论基础	315	15.1.3 编程步骤	415
11.2.3 编程步骤	316	15.2 编程实例二	420
第 12 章 爆炸效果的模拟	333	15.2.1 程序说明	420
12.1 编程实例一	333	15.2.2 理论基础	420
12.1.1 程序说明	333	15.2.3 编程步骤	421
12.1.2 理论基础	333	第 16 章 OpenGL 实现 3D 屏幕保护程序	429
12.1.3 编程步骤	335	16.1 屏幕保护程序编写方法	429
12.2 编程实例二	355	16.1.1 屏幕保护开发库	429
12.2.1 程序说明	355	16.1.2 MFC 开发屏幕保护程序	430
12.2.2 理论基础	355	16.2 编程实例一	430
12.2.3 编程步骤	357	16.2.1 程序说明	430
第 13 章 霓虹效果的模拟	363	16.2.2 理论基础	431
13.1 编程实例一	363	16.2.3 编程步骤	432
13.1.1 程序说明	363	16.3 编程实例二	436
13.1.2 理论基础	363	16.3.1 程序说明	436
13.1.3 编程步骤	364	16.3.2 理论基础	436
13.2 编程实例二	367	16.3.3 编程步骤	439

第 1 章 OpenGL 与图形可视化概述

1.1 图形可视化概述

所谓可视化 (Visualization),《牛津英语词典》解释为“构成头脑情景的能力或过程,或不可直接觉察的某种东西的视觉”。此术语亦指本来不可见的东西成为可见图像的过程, X 光片即为其例。有人指出:可视化是一系列的转换,这种转换将原始模拟数据转换成可显示的图像,这种转换的目的在于将信息转换成可被人类感应系统所领悟的格式。

可视化成为一种技术与方法应用于有关科学和工程技术各个领域,开始于利用计算机图形来加强信息的传递和理解。随后,计算机图像处理技术和计算机视觉也成功地用来处理各类医学图像和卫星图片,以帮助人们理解和利用各类图像数据。1998 年 2 月由 B.H.McCormick 等根据美国国家科学基金会召开的“科学计算可视化研讨会”的内容撰写的一份报告中正式提出了“科学计算可视化”(Visualization in Scientific Computing, 简称 ViSC)的概念,从此标志着一门新的可视化学科的问世。

科学计算可视化的产生是当代科学技术飞速发展的结果,而推动科学可视化这一新的研究方向发展的一个重要因素就是实现科学计算过程可视化方面的需求。科学家使用计算机进行计算时,不仅仅要对最终结果进行解释,而且希望能对计算的中间结果进行解释,即希望对整个计算过程进行实时的监控,例如,改变其参数、调整其显示的分辨率及视觉的效果等。对此,科学计算可视化提供了帮助科学家实现其驾驭计算过程模型研究系统行为的一个定量分析过程,在这个过程中无论是各子系统的运行流程还是系统的总体控制都是由大量计算数据所表现和完成的,其实现仍是一个科学计算过程。这样就决定了科学计算可视化技术在系统仿真方法领域中广泛应用的必然性,并最终导致可视化仿真 (Visual Simulation) 的出现。

科学计算可视化、计算机动画和虚拟现实是近年来在计算机图形学领域内 3 大活跃的发展方向,它们的技术核心都是三维真实感图形。无论是丝丝入扣的机械零件设计、惊心动魄的虚拟战场演习,还是如同身受的 3D 游戏,各种生动而精妙的图形应用使人目不暇接,计算机似乎为人们打开了一个五彩缤纷的第四维空间,而 OpenGL 正是这个“缤纷世界”的构造者之一。

1.2 OpenGL 的发展

1.2.1 关于 OpenGL

OpenGL 的前身是 SGI 公司为其图形工作站开发的 IRIS GL。IRIS GL 是一个工业标准的 3D 图形软件接口,功能虽然强大但是移植性不好,于是 SGI 公司便在 IRIS GL 的基础上开发了 OpenGL。OpenGL 的英文全称是 Open Graphics Library,顾名思义,OpenGL 便是“开放的图形程序接口”。

SGI 在 1992 年 7 月发布 OpenGL 1.0 版，后来成为工业标准，由成立于 1992 年的独立财团 OpenGL Architecture Review Board (ARB) 控制。SGI 等 ARB 成员以投票方式产生标准，并制成规范文档公布，各软硬件厂商据此开发自己系统上的实现。只有通过了 ARB 规范全部测试的实现才能称为 OpenGL，现在的 ARB 投票成员包括 SGI、Intel、IBM、nVIDIA、ATI、Microsoft、Apple、3DLabs 等业界群英。

计算机硬件性能的提高和 OpenGL 本身的不断发展，使得 OpenGL 不再只属于专用图形工作站。如今，开发人员可以在各种硬件平台利用 OpenGL 进行图形软件开发。

OpenGL 可以运行在当前各种流行操作系统之上，如 Mac OS、Unix、Windows 95/98、Windows NT/2000、Linux、OPENStep、Python、BeOS 等。各种流行的编程语言都可以调用 OpenGL 中的库函数，如 C、C++、Fortran、Ada、Java。OpenGL 完全独立于各种网络协议和网络拓扑结构。目前，Microsoft 公司、SGI 公司、ATT 公司的 Unix 软件实验室、IBM 公司、DEC 公司、SUN 公司、HP 公司等几家在计算机市场占主导地位的大公司都采用了 OpenGL 图形标准。值得一提的是，由于 Microsoft 公司在 Windows NT 和 Windows 95/98 中提供 OpenGL 标准，使得 OpenGL 在微机中得到了广泛应用。尤其是在 OpenGL 三维图形加速卡和微机图形工作站推出后，人们可以在微机上实现 CAD 设计、仿真模拟、三维游戏等，从而使得应用 OpenGL 及其应用软件来创建三维图形变得更有机会、更为方便。

1.2.2 OpenGL 1.X 回顾

自从 90 年代初以来，OpenGL API 就一直是专业 3D 制图技术中的重要组成部分，从专业的电影特效制作到游戏，在很多领域中 OpenGL 都大显身手，当时用的就是 OpenGL 1.X 标准。

在计算机 3D 领域，OpenGL 之所以能很快抛开其他竞争对手而流行开来，很大程度上是由于从一开始 OpenGL 就被设计成独立于硬件和操作系统的 3D API。所以 OpenGL 本身并不拘泥于操作系统的种类，可运行在采用不同操作系统的各种计算机上，并能在网络环境下以客户/服务器模式工作，于是，自然而然地 OpenGL 就成了专业图形处理、科学计算等高端应用领域的标准图形 API。

1992 年 7 月，SGI 公司发布了 OpenGL 的 1.0 版本，随后又与微软公司共同开发了 Windows NT 版本的 OpenGL，从而使一些原来必须在高档图形工作站上运行的大型 3D 图形处理软件，如用于制作电影《侏罗纪公园》、《玩具总动员》、《泰坦尼克号》，而大名鼎鼎的 Softimage 3D 也可以在微机上运用。

1995 年 OpenGL 的 1.1 版本面市，该版本比 1.0 的性能有许多提高，并加入了一些新的功能。其中包括改进打印机支持，在增强元文件中包含 OpenGL 的调用，顶点数组的新特性，提高顶点位置、法线、颜色、色彩指数、纹理坐标、多边形边缘标识的传输速度，引入了新的纹理特性等。

1999 年 4 月，SGI 和 ARB 公布了 1.2 版本的 OpenGL 标准，这是使用软件实现 OpenGL 三维图形处理的最高版本，以后的 OpenGL 版本都需要硬件的支持。

2001 年 8 月 17 日，SGI 和 ARB 公布了 1.3 版本的 OpenGL 标准，其主要技术特点包括立方体纹理贴图，应用于点、线和多边形的独立反锯齿渲染的多重采样以及其他一些有关纹理的技术。

2002 年 7 月 17 日，SGI 和 ARB 公布了 OpenGL 1.4 规范。该版本带有增强图形质量等级

和真实性的新型特性及功能设计,是业界领先的 2D、3D 图形 API 交叉平台的代表。为仍然保持完整的向下兼容性,ARB 开始让 OpenGL 2.0 加入可以提高目前及将来图形架构可编程序性效果的高级语言。

2003 年 7 月 28 日,SGI 和 ARB 公布了 OpenGL 1.5 规范。OpenGL 1.5 中包括 OpenGL ARB 的正式扩展规格绘制语言——OpenGL Shading Language。该语言将作为即将发布的 OpenGL 2.0 的底核。

1.2.3 从 1.X 到 2.0

继 1.5 版本后,ARB 又展示了具有美好前程的 OpenGL 2.0 版本。2004 年 8 月 12 日,ARB 发布了 OpenGL 2.0 版标准。OpenGL 2.0 标准的主要制订者并非原来的 SGI,而是逐渐在 ARB 中占据主动地位的 3Dlabs。

OpenGL 2.0 标准要求图形芯片厂商需要专门开发支持 OpenGL 2.0 的驱动程序,同时也要求编程者遵循 OpenGL 2.0 标准,这样编写出来的程序能正确地在 ATi、nVIDIA 和 Intel 图形芯片硬件上运行。

可编程着色语言是 OpenGL 2.0 的一大主要功能,但除此之外 OpenGL 2.0 还有其他几项重要的功能,包括顶角处理器、区段处理器、封装和解封装运算、数据移动与内存管理。顶角处理器功能在于照明、材质和几何图形的弹性;区段处理器功能为材质存取、内插器与像素运算;封装和解封装运算的目的在于将应用像素转换成一致性的像素群数据流;数据移动与内存管理则是为了增进效能,数据移动量必需减至最低并且能消除为增进数据流量而产生的数据备份,从而大幅度提高性能。

1.3 OpenGL 2.0 的新技术

1.3.1 OpenGL 2.0 由来

OpenGL 2.0 标准的主要制订者并非原来的 SGI,而是逐渐在 ARB 中占据主动地位的 3Dlabs。3Dlabs 认为 OpenGL 1.X 面临的主要问题是:

(1) 系统和图像架构自 1992 年来有了重大改变,以 Glide 和 DirectX 为主的游戏 3D API 开始成为家用市场的主流,很显然,OpenGL 没有跟上节奏,自始至终都未能在家用游戏市场获得足够影响力。

(2) 现在的显卡功能和性能已经越来越先进,但 OpenGL 的函数却无法发挥出这些优秀硬件的性能,所以说目前的 OpenGL 1.X 已经落后于今天先进硬件的功能和性能。

(3) 虽然在 OpenGL 下也有一些优秀的游戏,比如 QuakeII,但它实现和应用的复杂性却远远超过其他 3D API,所以易用性是 OpenGL 当前的一个大问题。

在 3Dlabs 的构想中,OpenGL 2.0 版本的目标是:

(1) 将 OpenGL 提升到一个新的高度,像 Direct3D 那样由核心提供全部硬件功能,以配合目前的和将来的显示硬件。

(2) 展示下一代可编程图像硬件的图景,使用高级语言而不是汇编语言对显卡编程。

(3) 由定义良好的核心功能取代大部分扩展功能,以可编程性取代复杂性。

- (4) 对已有程序保持相当理想的兼容性，以使老程序不经修改便能正常运行，并保持高效。
- (5) 对不需要兼容性的场合，提供简洁干净的核心，剔除过时的函数。
- (6) 提高动态材质等动态媒体的功能和效率，以满足对动态媒体处理的需要。

ARB 接受了 3Dlabs 的提议，进行 OpenGL 2.0 架构的开发，也描绘出一个清楚的发展路线。并且在其他地方也进行了一些更新，并于 2004 年 8 月公布了其最新的 2.0 版本。OpenGL 的主要且近期的目标，就在于启动并开发硬件的可编程能力，这一点恐怕是 OpenGL 在游戏市场对抗微软 D3D 的主要武器。当然不可否认的是，OpenGL 在其他地方也需要进行更新。

1.3.2 OpenGL 2.0 特点与新技术

成形后的 OpenGL 2.0 最重要的更新是加入了可编程能力，为使其符合大部分应用程序与用户的标准，它将像以前的 OpenGL 1.X 一样提供一个丰富的功能集，独立于硬件并专门在 OpenGL 架构下使用。

1. 兼容性

新的 OpenGL 2.0 标准首先要做的是与 OpenGL 1.3 之间的兼容性，而且应该是完整的兼容性。为了达到这一目的，3Dlabs 设计的 OpenGL 2.0 将由 OpenGL 1.3 的现有功能加上与之完全兼容的新功能所组成，这样不但可保持兼容性，还能对各家推出的各种纠缠不清的扩展指令集进行一次彻底精简。

2. OpenGL 2.0 的 Shading 语言

2002 年，NVIDIA 在推动交互式 3D 程序开发方面投入了大量精力，其中最主要的就是 Cg 图形语言（即“图形技术的 C 语言”——C for Graphics，这是 NVIDIA 为超越 DirectX 和 OpenGL 而开发的高级光影效果程序语言）。NVIDIA 试图把这个语言递交给 ARB，希望其成为 OpenGL 组织认可的标准。但最终 ARB 采用的 Shading 语言却是 3Dlabs 的 Glslang，这套语言会虚拟资源管理，由于 OpenGL 的内存管理还是黑箱作业，所有事项都会自动处理，因此，对大多数程序设计师来说便不用去考虑资源管理。

3. 扩展性

如果只是简单的精简，必然会导致 OpenGL 指令的功能下降，道理很简单，因为可使用的指令集减少了。所以在精简的同时必须扩展指令的功能，也就是说为留下来的“旧”指令定义新的功能。OpenGL 2.0 增加了片段处理器的能力，取代旧有的内插值顶点数据运算、像素缩放、质材存取及应用、雾化等。可编程图像格式将取代固定格式封装和解封装运算，在传送或接收像素数据时，将允许类型与格式进行任意组合。

4. 可编程性

在所有这些扩展功能中，可编程性是最吸引人的。开发者们可以借助 OpenGL 2.0 提供的可编程能力，减少对现有及未来扩展指令的需求。

最著名的可编程功能应该是可编程顶点，它允许进行随机个别顶点运算，同时也会取代部分旧有的 OpenGL 管线，如顶点转换、正规转换、照明、色彩强化、材质坐标产生及转换等。与之相关的还有可编程片断处理和可编程图像格式，其中前者主要负责材质存取、插值运算与像素运算弹性，它将取代内插值顶点数据运算。

OpenGL 支持两种可编程模型：

顶点编程（Vertex Program）：对应于 Direct 3D 中的 Vertex Shader，提供可编程的 T&L

能力，代替了传统渲染流水线中的 T&L，处理顶点的变换、光照计算。

片断编程 (Fragment Program)：对应于 Direct 3D 的 Pixel Shader，提供可编程的光栅化操作，代替了传统流水线中的纹理映射、逐像素颜色及雾化处理。

5. 多渲染目标支持 (Multiple Render Targets)

这种技术的作用就是将每像素的数据保存到不同的缓冲区中，这些缓冲区数据由此可成为照片级光照效果着色器的参数。此技术能有效节省硬件资源，达到更有效的运算。

6. 对非二次幂纹理技术的支持

以前版本的 OpenGL 仅支持分辨率为 $2^m \times 2^n$ 的纹理，但现在我们可以使用“非二次幂”的纹理。这样使得与之匹配的屏幕分辨率或包围区域不一定非要等于二次幂，以此来减小显存消耗。非二次幂的纹理将会带来一些纹理坐标贴图的差异，主要体现在两个方面——没有纹理细化过滤处理和不支持镶边纹理或重映纹理封装模式。

7. 支持点纹理 (Point Sprites)

目前已有许多新游戏支持这项技术，是一种快速的粒子渲染，它表示在渲染粒子效果时显卡所能绘制点的数量，例如，火花、灰尘或爆炸效果。此外，对于天空雪花飘舞的渲染，也是通过这种技术来实现的。但是，点状粒子效果有很大限制，它们不能被独立构成、旋转或运动起来，这是因为它们被确立为一个独立的点。当在这些特性需要的场景中，例如爆炸，我们将其渲染成四边形的粒子，它们被设定为两个正好相配的三角形。

8. 支持双面模板 (Two-Sided Stencil) 技术

在 DOOM3 中，我们可看到大量的动态光影效果，这种动态光源、即时阴影的效果主要是通过 Shadow Volume (阴影锥) 技术实现的。一个物体之所以会处在阴影当中，是由于在它与光源之间存在着障碍物。而障碍物产生的阴影范围从几何角度来看就是一个锥形，所以我们称其为 Shadow Volume。Shadow Volume 其实也是一个几何构成，也是需耗费系统资源来生成的。有了阴影锥后，游戏会对所有物体进行即时检测，来判断是否有物体处在阴影锥内，是否需要即时投射阴影。

判断物体是否在阴影锥内，需要使用 Stencil Buffer (模板缓冲区)。当物体进入阴影锥时，Stencil Buffer 模板值加 1；而当它离开阴影锥时，模板值减 1。如果模板值为 0，则表示实现进入和离开阴影锥的次数相等，自然就表示物体不在阴影锥内了。如此复杂的 Shadow Volume 算法才能描绘出那些逼真的动态阴影，随之而来的肯定是极大的资源消耗。此外，频繁的模板缓冲调用操作也会占据一部分显存带宽。

而双面模板 (Two-Sided Stencil) 技术则可在处理阴影时减少因顶点带宽不足对速度的消极影响。它建立了两套 Stencil Buffer，一套为 Front Face 三角形 (即面对光源的这一面)，另一套是为 Back Face 三角形 (即背向光源的一面)。原来需两遍的渲染计算，在 Two-Sided Stencil 的模式下，则需一遍就够了。这样可极大地减少对资源的消耗，能显著提高阴影加速性能。

9. 增强数据管理性能

在 OpenGL 中的数据类型主要有顶点数据 (色彩、正规、位置、用户定义等) 以及图像数据 (材质、图像、像素缓冲区)。OpenGL 2.0 提供了更好的数据移动和内存管理功能，加强了应用程序对数据移动的控制能力和更好的顶点处理能力，能消除为增进数据流量而产生的数据备份，大幅度提高性能。

1.4 OpenGL 基本操作

OpenGL 作为一个性能优越的图形应用程序设计界面（API），适用于广泛的计算机环境。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。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。

客观世界和各种事物的形状虽然千变万化，但用计算机将之描述出来却只需要把一系列基本操作组合起来。OpenGL 提供了以下基本操作。

1. 绘制物体

真实世界里的任何物体都可以在计算机中用简单的点、线、多边形来描述。OpenGL 提供了丰富的基本图元绘制命令，从而可以方便地绘制物体。

2. 变换

可以说，无论多复杂的图形都是由基本图元组成并经过一系列变换来实现的。OpenGL 提供了一系列基本的变换，如取景变换、模型变换、投影变换及视口变换。

3. 光照处理

正如自然界不可缺少光一样，绘制有真实感的三维物体必须做光照处理。

4. 着色

OpenGL 提供了两种物体着色模式，一种是 RGBA 颜色模式，另一种是颜色索引模式。

5. 反走样

在 OpenGL 绘制图形过程中，由于使用的是位图，所以绘制出的图像的边缘会出现锯齿形状，称为走样。为了消除这种缺陷，OpenGL 提供了点、线、多边形的反走样技术。

6. 融合

为了使三维图形更加具有真实感，经常需要处理半透明或透明的物体图像，这就需要用到融合技术。

7. 雾化

正如自然界中存在烟雾一样，OpenGL 提供了 fog 的基本操作来达到对场景进行雾化的效果。

8. 位图和图像

在图形绘制过程中，位图和图像是非常重要的一个方面。OpenGL 提供了一系列函数来实现位图和图像的操作。

9. 纹理映射

在计算机图形学中，把包含颜色、alpha 值、亮度等数据的矩形数组称为纹理。而纹理映射可以理解为将纹理粘贴在所绘制的三维模型表面，以使三维图形显得更生动。

10. 动画

出色的动画效果是 OpenGL 的一大特色，OpenGL 提供了双缓存区技术来实现动画绘制。

OpenGL 并没有提供三维模型的高级命令，它也是通过基本的几何图元——点、线及多边形来建立三维模型的。目前，有许多优秀的三维图形软件（如 3DMAX）可以较方便地建立物体模型，但又难以对建立的模型进行控制，若把这些模型转化为 OpenGL 程序，则可随心所欲地控制这些模型来制作三维动画，实现仿真数据的可视化和虚拟现实。

1.5 OpenGL 图形的实现

1.5.1 OpenGL 体系结构

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

首先,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达的物体则可以实现其在空间的各种运动。

其次,OpenGL 通过光照处理能表达出物体的三维特性,其光照模型是整体光照模型,它把顶点到光源的距离、顶点到光源的方向向量以及顶点到视点的方向向量等参数代入该模型,计算顶点颜色。因此,可视化仿真图像的颜色体现着物体与视点以及光源之间的空间位置关系,具有很强的三维效果。

另外,为弥补图形方法难于生成复杂自然背景的不足,OpenGL 提供了对图像数据的使用方法,即直接对图像数据读、写和拷贝,或者把图像数据定义为纹理与图形方法结合在一起生成视景图像以增强效果。为增强计算机系统三维图形的运算能力,有关厂家已研制出了专门对 OpenGL 进行加速的三维图形加速卡,其效果可与图形工作站相媲美。

一个完整的窗口系统的 OpenGL 图形处理系统的结构为:最底层为图形硬件,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,如图 1.1 所示。

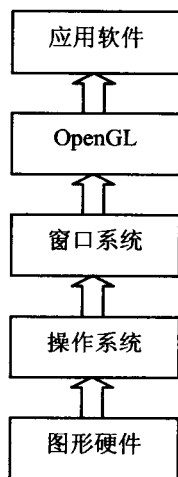


图 1.1 OpenGL 图形处理系统的层次结构

OpenGL 是网络透明的,在客户机/服务器体系结构中,允许本地或远程调用 OpenGL。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。

由于 OpenGL 是一个与平台无关的三维图形接口,操作系统必须提供像素格式管理和渲