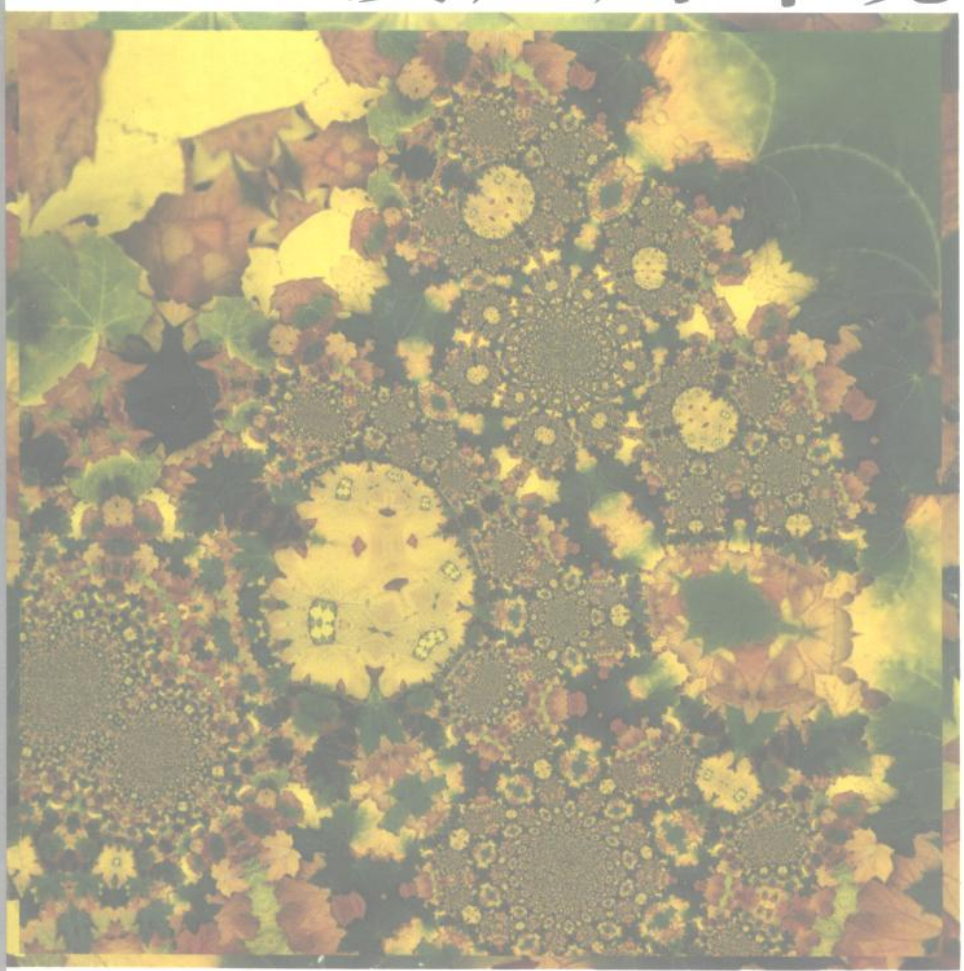
吴海平 罗红兵 张映平 朱敏 张小强

OpenGL

图形程序设计及应用环境



国防科技大学出版社

OpenGL 图形程序设计与应用环境

吴海平　罗红兵　张映平
朱　敏　张小强　编著

国防科技大学出版社
·长　沙·

图书在版编目(CIP)数据

OpenGL 图形程序设计与应用环境/吴海平等编著. —长沙:国防科技大学出版社,1999.6

ISBN 7-81024-544-9

Ⅰ. O… Ⅱ. 吴… Ⅲ. 图形软件, OpenGL-程序设计 Ⅳ. TP391.4

中国版本图书馆 CIP 数据核字(1999)第 18397 号

国防科技大学出版社出版发行

电话:(0731)4555681 邮政编码:410073

E-mail:gfkdcbs@public.cs.hn.cn

责任编辑:黄 煌 责任校对:文 慧

新华书店总店北京发行所经销

国防科技大学印刷厂印装

*

787×1092 1/16 印张:23.5 字数:543 千

1999 年 6 月第 1 版第 1 次印刷 印数:1—2000 册

*

定价:28.00 元

内 容 提 要

OpenGL(Open Graphics Library)是目前应用最广泛的二维和三维图形程序设计标准。OpenGL 应用程序界面(API)是一个多平台工业图形标准。本书由 OpenGL 导论、OpenGL 程序设计方法、OpenGL 与窗口系统的结合技术和 OpenGL 的实际应用四个部分组成,系统翔实地介绍了 OpenGL 的程序设计方法与应用技术。

本书内容丰富,叙述深入浅出,有大量的实例分析。它既可以作为计算机图形领域内大学本科生和研究生的教材,也可以作为相关专业科研教学人员的实用参考书,还可以作为具备 OpenGL 运行支持环境的任何一种机器的随机手册。

前　言

1962 年美国麻省理工学院林肯实验室的 Ivan E. Sutherland 的博士论文《Sketchpad：一个人－机通信的图形系统》的发表，标志了计算机图形学作为一个崭新的科学分支的确定。30 多年来，计算机图形学飞速发展并得到了广泛的应用。计算机图形学的研究成果不仅在计算机用户界面设计、计算机辅助设计与制造(CAD/CAM)、地理勘探测量、过程控制与模拟、办公自动化、艺术模拟、软件与科学计算可视化、计算机辅助教学等方面得到了大量的应用，在其它方面的应用也在不断深入。例如在医学方面，利用计算机三维图形展示人体各部位的解剖模拟，可以精确地判断病情或进行手术；在娱乐方面，计算机游戏、电影电视特技、计算机动画等等更是多姿多彩，令人陶醉；在刑事侦破方面，可以利用计算机图形学来处理现场数据、再现当事人的图像及犯罪场景。总之，计算机图形学的应用极大地提高了人们理解数据、观察现实和想象形体的能力。随着个人计算机性能的不断发展、计算机图形软件的不断丰富，计算机图形学的应用前景不可估量。

由于计算机图形学的广泛应用，计算机图形学的功能除了随着计算机图形设备的发展而提高外，其自身也在不断地在朝着标准化、集成化和智能化的方向发展。有关计算机图形学方面的标准越来越多，给计算机生产厂家和计算机图形用户提供了较大的选择空间。另一方面，软件与科学计算可视化、虚拟现实环境、计算机仿真模拟等的应用使三维计算机图形学在真实性和实时性方面的表现技术越来越完善，三维计算机图形技术已经成为计算机图形学的一个最主要的组成部分。

在诸多计算机三维图形标准的角逐中，OpenGL 三维图形开发标准崭露头角，目前已经成为三维图形事实上的通用标准。

OpenGL(开放式图形库的简称)是在 SGI 公司 GL 图形库的基础上扩展而来，并且被几乎所有的计算机厂家采纳的一个计算机图形标准。OpenGL 所具有的功能基本上涵盖了计算机图形学所要求提供的方方面面的功能，包括基本图形元素的生成(例如点、线、多边形、二次曲线曲面生成)，封闭边界内的填色、纹理、反走样等；基本图形元素的几何变换、投影变换、窗口裁剪等；NURBS 曲线曲面处理；隐藏线、隐藏面消除以及具有光照颜色效果的真实图形显示；自然界效果(例如云彩、薄雾、烟霭)的景象生成等等。

本书系统翔实地介绍了 OpenGL 的程序设计方法、OpenGL 与目前主流窗口系统及高级程序设计语言的捆绑应用，是一本相当实用的 OpenGL 使用资料。该书用了四个部分来介绍 OpenGL 的功能与使用方法。第一部分包含前三章的内容，综合介绍了 OpenGL 的作用、意义、发展历史、总体结构和内部处理流程。通过这一部分的学习，读者可以对 OpenGL 有一个全局的掌握和了解。对这一部分的学习可以采用螺旋式的方法，也就是 OpenGL 的初学者首先通过学习本部分来获得对 OpenGL 总体的轮廓性了解，以便为后面的学习建立一个全局的概念；对已经熟悉 OpenGL 或学习了其余部分内容后再学习本部分的读者则能从总体上对 OpenGL 进行概括和综合。第四章到第十四章属于第二部分，

也是本书的主体部分。这一部分基本上是按三维图形技术构件来编排章节的，每一章对应三维图形技术的一个或若干个构件来介绍 OpenGL 的相应成分和程序设计方法。第十五章和第十六章属于第三部分，分别介绍了 OpenGL 与 X Window 和 Windows 95/NT 两个主流窗口系统的结合技术。这部分的学习要求读者比较熟悉 OpenGL 的各个功能。第十七章到第二十章是第四部分，这部分内容主要是为了扩展读者应用 OpenGL 的视野，对从事 OpenGL 应用的读者具有启迪和参考作用。本书最后给出了两个附录，其中附录 A 按功能列出了 OpenGL 各种库中所包含的命令及其功能简述，附录 B 给出了本书所用到的一些术语的解释。

本书前十四章的学习只要求读者有一定的 C 语言程序设计知识，而后几章由于涉及到 OpenGL 的实际应用，因此要求读者必须具备一些相应的知识，主要是 Windows 95 下的 Viscal C++程序设计、UNIX 环境下基于 X Window 的 C 语言程序设计等。当然，如果读者具有计算机图形学的知识，对本书的学习将具有极大的帮助。

本书的第一部分、第四、五、十、十三、十四、十七、二十章和附录由吴海平编著，第七、八、九、十二和十五章由罗红兵编著，第十一、十八和十九章由张映平编著，第十六章由朱敏编著，第六章由张小强编著，本书的章节编排和全书的统稿由吴海平负责。朱敏和罗红兵在 DEC 工作站的 UNIX 环境下和微机的 Windows 95/NT 环境下调试通过了书中的全部实例。为了使实例简单易读，前十四章中的实例均采用了 OpenGL 辅助库。这里要强调指出的是，OpenGL 辅助库没有什么实际价值，只是用来帮助 OpenGL 的学习，建议读者结合后几章的学习将这些实例进行改写。

本书在编写过程中得到了许多同事的关心和帮助，其中杨学军教授、夏卫民研究员、杨桃栏教授、胡子昂副教授、黄瑞芳副教授等同志为本书的成文与出版提供了大力的支持，张奇飞、胡晓静、乔川龙、陈琳、郭歌、杨进、邹捷、何蓉晖等同志为本书作了很多事务性工作。谨此表示感谢。

吴海平
国防科技大学计算机研究所
E-mail:hpwu@nudt.edu.cn
电话:(0731)4506605
1999.3

目　录

第一篇　OpenGL 导论

第二篇　OpenGL 程序设计方法

第三篇　OpenGL 与窗口系统的结合技术

第四篇　OpenGL 的实际应用

第一篇

OpenGL 导论

第一章 OpenGL 素描

1.1 OpenGL 概述

随着计算机图形(尤其是交互式 3D 计算机图形)应用的日益普及,大量图形应用程序已经或正在各类机器平台上开发。从简单的平面图形程序到高级模型模拟仿真程序与可视化软件,计算机图形的应用领域越来越广,对人们的吸引力也越来越大。对计算机图形开发人员来说,他们非常希望有一种交互式 3D 图形标准,能够保证他们的应用程序不作修改即可在具有不同图形处理功能的不同机器平台上运行,并且产生相同的显示结果。

一个具有生命力的 3D 图形标准必须满足三个条件:它必须能在具有不同图形处理能力的各种平台上实现而不会损失底层的硬件性能;它必须提供友好的用户界面和简捷方便的图形操作描述手段;它必须具有灵活的可扩展包容性,使得新的图形子系统所描述的操作可以正常执行而不会扰乱原来的标准。OpenGL(Open Graphics Library)就是满足这些条件的一个 3D 图形标准,它是图形硬件与应用程序之间的一个抽象界面,它支持诸如点、线段、多边形和图像一类的基本图原。OpenGL 既具有诸如仿射(Affine)、投影变换、光照计算等基本绘制操作,也支持纹理映射、反走样一类的高级绘制特性。

OpenGL 是目前用于开发可移植、可交互的 2D 和 3D 图形应用程序的首选环境,也是目前最广泛采用的计算机图形标准。OpenGL 不仅具有高质量的可视效果和性能支持,无论是 3D 动画、CAD,还是可视仿真模拟,OpenGL 都能应付自如;同时 OpenGL 的程序设计界面又相当简单,这使得 OpenGL 开发人员可以在诸如 CAD/CAM/CAE、娱乐、医学图像、虚拟现实等各种图形应用领域中生成和显示令人神往的梦幻般的 2D 和 3D 图形。因此,从 1992 年推出以来,OpenGL 就成为图形领域中最广泛使用和支持的 2D 和 3D 图形应用程序设计界面(API),成千上万种 OpenGL 应用程序已经在各类计算机平台上开发出来。OpenGL 利用其特有的绘制手段、纹理映射、特殊效果和其它强有力的图形描

绘功能，提供了崭新而高效的计算机图形应用程序开发支持。OpenGL 应用程序开发人员可以在各种普通微机、工作站平台或超级计算机上执行其开发的应用程序而无须考虑平台相关性问题。

OpenGL 被越来越多的计算机图形应用程序开发人员选用的主要理由在于 OpenGL 提供了如下的一些主要支持。

1. 图形工业标准

OpenGL 是目前唯一真正具有开放性、平台无关性和厂商公平性的一个图形工业标准。包括 AccelGraphics、DEC、HP、Microsoft、NEC、Sony、SGI、SUN 在内的几乎所有的微机、工作站和超级计算机厂家都已经把 OpenGL 作为其高性能 2D 和 3D 图形的直接标准。另外还有一些第三方公司也在为一些著名的机器平台实现 OpenGL，例如 Conix Graphics 已经在 Apple 上实现了 OpenGL。

作为一个图形工业标准，OpenGL 由成立于 1992 年的 OpenGL Architectural Review Board(简称为 OpenGL ARB)来管理和发布。OpenGL ARB 负责制定 OpenGL 的规范测试标准(能够判断 OpenGL 的实现是否成功的一组测试程序，其目的在于保证应用程序源代码在任何一个 OpenGL 的实现环境中都是一致的)、升级描述、功能审定和版本发布。OpenGL ARB 在 1992 年公布了第一个版本 OpenGL 1.0，该版本包含了 OpenGL 的核心函数、实用函数的描述和定义，这是目前最广泛实现的一个 OpenGL 版本，包括与 Windows NT、Windows 95 的捆绑。1994 年发布的 OpenGL 1.1 版对前一个版本作了细微的修改，扩充了处理纹理的临界性能特性和顶点数据封装(本书介绍的内容就是基于这个版本)。1996 年 OpenGL ARB 发布了 OpenGL 1.2 版，这也是目前最新的版本。该版本新增加了一组可选图像处理函数，扩充了部分核心函数(包括三维纹理、RGBA 像素格式和封装像素格式、顶点法向的自动缩放等)。

2. 兼容性

尽管 OpenGL 来源于 SGI 公司的 IRIS GL 库，但它们之间有明显的区别：OpenGL 的 API 界面经过了一系列的整理，去掉了 IRIS GL 中的一些冗余结构和冲突函数；所有 OpenGL 命令均有前缀字母"gl"，这就避免了与其它图形界面 API 的名字冲突；OpenGL 在 IRIS GL 基础上增加了一些新的功能，包括纹理映射、反走样多边形、模板缓冲区(Stencil Buffer)和累积缓冲区(Accumulation Buffer)等。所以 OpenGL 不是 IRIS GL，OpenGL 遵循 ARB 制定的标准，具有规范的版本描述和及时的修订通告。OpenGL 版本升级要求保证向下的兼容性，这就使得已开发出来的应用程序不会被丢弃。

3. 可靠性和可移植性

任何一个 OpenGL 应用程序在任何一种遵循 OpenGL API 标准的环境下都会产生相同的可视显示结果，无须考虑环境中的操作系统和窗口系统。OpenGL 没有附属任何窗口函数，它必须在具体平台上与特定的窗口系统结合才能执行。每一个遵从 ARB 规范的 OpenGL 实现都会包含 OpenGL 描述的全体函数。从平台配置来看，所有 UNIX 工作站、标准的 Windows NT 和 Windows 95 都支持 OpenGL。OpenGL 已经在当前所有主流操作系统下运行，包括 MacOS、OS/2、UNIX、Windows 95、Windows NT、Linux、OPENStep、Python 和 BeOS；它也与主要的窗口系统进行了结合，包括 Presentation Manager、Win32

和 X Window 系统。OpenGL 完全独立于网络协议和拓扑。

4. 可扩展性

因为 OpenGL 本身设计的完备性和前瞻性，可以按照 OpenGL 的扩展机制通过 API 去访问新的图形硬件。按照这一特性，新硬件总是适时地出现在 API 中，OpenGL 开发人员和硬件厂商可以将新的硬件特性并入到其正常的产品生产中。虽然 OpenGL 描述定义了一个特定的图形处理流水线，厂家仍然可以对 OpenGL 的实现采取特殊处理以满足对特定系统性能的要求。OpenGL 的某个具体函数可以作为一个软件程序在特定的硬件上执行，也可以将 OpenGL 函数在特定硬件中固化。OpenGL 实现中的可扩展性意味着 OpenGL 从简单的绘制到完整的几何处理都可以利用硬件加速，实际上采用硬件加速实现 OpenGL 也已经广泛使用在普通微机、工作站和超级计算机上。OpenGL 应用程序开发人员可以不关心环境中 OpenGL 是如何实现的，因为他的程序的显示结果在不同环境下都是相同的。

5. 广延性

基于 OpenGL API 建立起来的图形应用程序可以在普通的娱乐器具(例如电子游戏机)、微机、工作站及超级计算机等各种层次的计算机上运行。也就是说，无论 OpenGL 程序开发人员选择什么环境，OpenGL 都可以外延覆盖。

6. 易用性

OpenGL 具有精简直观的函数式命令结构。在同等编程条件下，OpenGL 编写的应用程序代码行要少于用其它图形库或图形包编制的程序。此外，OpenGL 驱动程序封装了底层的硬件信息，使应用程序开发人员无须针对特定的硬件特性来编程。OpenGL 使图形软件的开发变得简单易行——从简单的几何点、线或填充多边形到最复杂的光照和 NURBS (Non-Uniform Rational B-Spline)曲面纹理映射。OpenGL 给图形软件开发人员提供了对几何和图像图原、显示列表、模型变换、光照和纹理、反走样、融合(Blending)以及其它许多特性的直接控制访问。所有 OpenGL 的状态分量(包括纹理存储器和帧缓冲区内容等)都可以通过 OpenGL 的应用来获得。OpenGL 还支持 2D 图像的可视化处理，它将 2D 图像处理成图原，以便能像处理 3D 几何对象那样进行 2D 图像操作。OpenGL 可以在 C、C++、Fortran、Ada、Java 等高级语言中调用。

1.2 OpenGL 的概念与特征

1.2.1 OpenGL 的基本概念

1. 顶点(Vertex)

在 OpenGL 中，所有几何对象最终都是由一组有一定顺序的顶点坐标来描述的。顶点用来定义一个点、线段的端点或多边形两条边相交的一个角，每一个顶点可以用 2～4 个坐标来描述(4 个坐标指的是三维齐次坐标顶点)。在每个顶点的处理中可以用到当前法线、当前纹理坐标和当前颜色。OpenGL 在进行光照计算时使用法线，当前法线是由描述它的三个坐标构成的一个三维向量；颜色或者由红、绿、蓝和 Alpha 值构成，或者由单个颜

色索引值构成；纹理坐标(1～4 个)决定纹理图像如何映射到图原上。

2. 像素(Pixel)

OpenGL 最后绘制的图像是屏幕上的一系列像素值，因此像素(图像元素的简称)是显示设备能够显示到屏幕上的最小可见单位。有关像素的信息，如它们支持什么元颜色等，是由系统显示存储器的位面数来决定的。位面是内存中的一块区域，它保留每个像素点在屏幕上的一位信息，该位可以决定某一像素点应该显示什么颜色。位面的总和构成了帧缓冲区，其中保存了图形显示所需要的屏幕上的所有像素值。

3. 帧缓冲区(Frame Buffer)

OpenGL 的帧缓冲区既可以是运行环境中窗口系统构造的一个窗口，也可以是存储器中的一个数据结构。OpenGL 的帧缓冲区逻辑上是由一组缓冲区组成的，每个缓冲区逻辑上就是一个二维数组。OpenGL 的帧缓冲区从功能上可以分为两种：图像缓冲区和辅助缓冲区。图像缓冲区是 OpenGL 最重要的缓冲区，它包含实际的颜色信息和 Alpha 分量；辅助缓冲区包括深度缓冲区(Depth Buffer)、模板缓冲区(Stencil Buffer)、累积缓冲区(AccumulationBuffer)、双缓冲区和立体缓冲区等。深度缓冲区也称为 z 缓冲区，它存储每个像素的深度值。模板缓冲区用来限制绘制到屏幕某个区域中的内容，这有点像用一个有图案的纸模板精确地绘制一幅图画一样。累积缓冲区存储 RGBA 颜色数据，用来累积一系列图像，形成一个最终的合成图像。双缓冲区指的是前端可视缓冲区与后台可绘制缓冲区的组合，它可以使得在显示一幅图像的同时绘制另一幅图像，这是通过将后台可绘制缓冲区绘制到内存中，然后将缓冲区内容快速拷贝到屏幕缓冲区来实现的。立体缓冲区类似于双缓冲区，它组合了多个缓冲区，包括前端、后端以及左右缓冲区。大多数 OpenGL 的实现都支持双缓冲区，而立体缓冲区要求特殊的硬件支持，因而当前大多数 OpenGL 的实现还不支持立体缓冲区。

4. 矩阵堆栈(Matrix Stack)

OpenGL 用到三种 4×4 的变换矩阵：模式观察(Model-view)矩阵用于顶点坐标、纹理矩阵应用于纹理坐标、投影(Projection)矩阵描述观察截头体(Viewing-frustum)，它必须在模式观察矩阵对顶点坐标变换后，才能应用于该顶点坐标。这三种矩阵都可以按照通常的变换方式来加载或相乘。它们中的每一个都是一种矩阵堆栈，栈顶矩阵就是当前应用于顶点坐标并由相关矩阵操作和控制命令所施加的对象。

5. 状态查询与属性栈

OpenGL 的工作方式是一种状态机机制，用户可以设置各种状态或模式并将状态值存放在栈中。几乎所有的 OpenGL 状态变量值都可以通过 OpenGL 的查询命令来获得。所有状态从功能上分为 21 组，它们的结合可以压入属性栈中。查询命令与状态栈的使用使得各种库都可以有效地使用 OpenGL 而不会彼此混淆。

6. 颜色模式

OpenGL 颜色模式指 RGBA 模式或颜色索引模式。在 RGBA 模式下，所有的颜色定义全用 R(红)、G(绿)、B(蓝)和 Alpha 值来表示，其中 Alpha 表示颜色的透明度，在颜色融合操作时使用。在颜色索引模式下，每一个像素的颜色是用颜色索引表中的某个颜色索引值表示，而这个索引值指向相应的 R、G、B 值。