

华章程序员书库

Pearson

HZ Books
华章IT

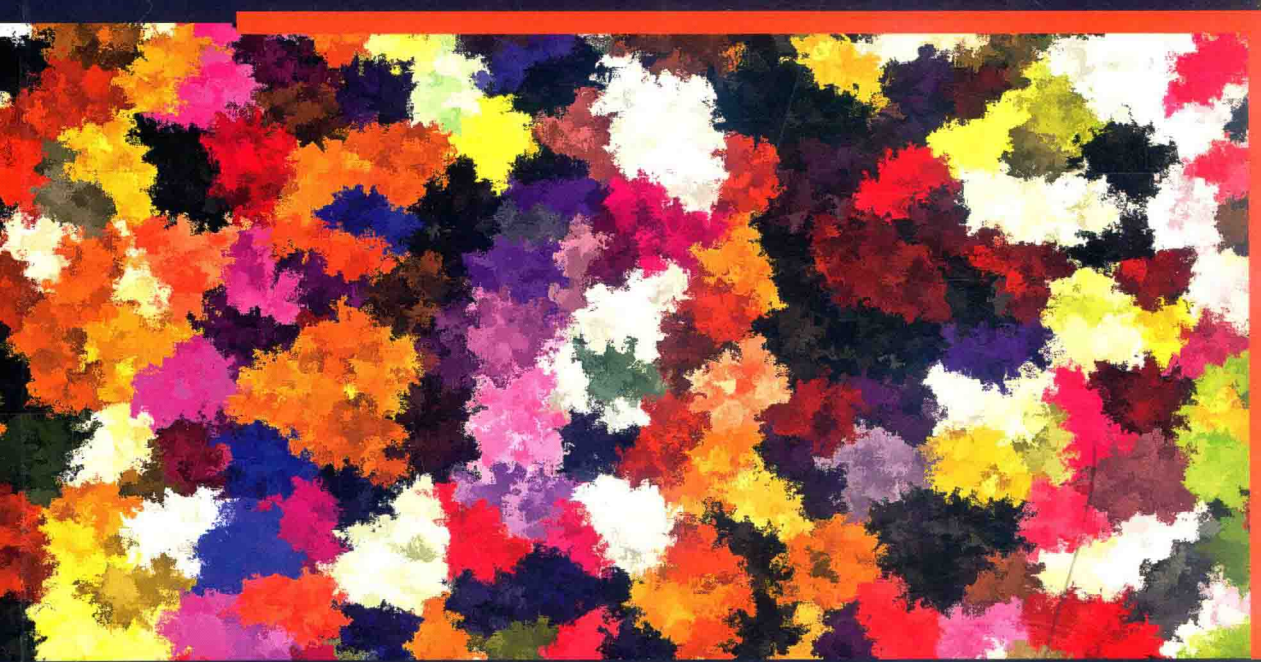
OpenGL编程指南

(原书第9版)

OpenGL Programming Guide

The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V

Ninth Edition



约翰·克赛尼希 (John Kessenich)

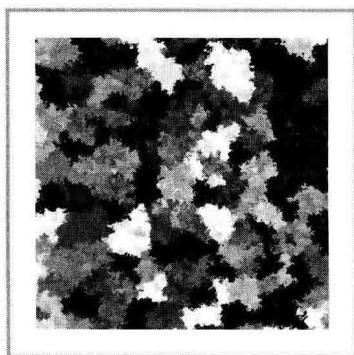
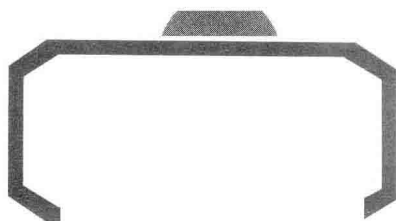
[美] 格雷厄姆·塞勒斯 (Graham Sellers) 著

戴夫·施莱尔 (Dave Shreiner)

王锐 等译



机械工业出版社
China Machine Press



OpenGL Programming Guide

The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V

Ninth Edition

OpenGL编程指南

(原书第9版)

约翰·克赛尼希 (John Kessenich)

[美] 格雷厄姆·塞勒斯 (Graham Sellers) 著

戴夫·施莱尔 (Dave Shreiner)

王锐 等译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

OpenGL 编程指南 (原书第 9 版)/(美) 约翰·克赛尼希 (John Kessenich) 等著; 王锐等译. —北京: 机械工业出版社, 2017.7

(华章程序员书库)

书名原文: OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V

ISBN 978-7-111-57511-5

I. O… II. ①约… ②王… III. 图形软件—指南 IV. TP391.41-62

中国版本图书馆 CIP 数据核字 (2017) 第 176109 号

本书版权登记号: 图字: 01-2016-8657

Authorized translation from the English language edition, entitled OPENGL PROGRAMMING GUIDE: THE OFFICIAL GUIDE TO LEARNING OPENGL, VERSION 4.5 WITH SPIR-V, 9th Edition, ISBN: 0134495497 by KESSENICH, JOHN M.; SELLERS, GRAHAM; SHREINER, DAVE, published by Pearson Education, Inc, Copyright © 2017.

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 CHINA MACHINE PRESS/HUAZHANG CO., Copyright © 2017.

本书中文简体字版由 Pearson Education (培生教育出版集团) 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

OpenGL 编程指南 (原书第 9 版)

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 缪 杰

责任校对: 李秋荣

印 刷: 三河市宏图印务有限公司

版 次: 2017 年 8 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 42.5

书 号: ISBN 978-7-111-57511-5

定 价: 139.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

Praise 推荐语

“这是一本一站式服务的 OpenGL 书籍，它就是我梦寐以求的那种图书。感谢 Dave、Graham、John 和 Bill，感谢你们做出的了不起的贡献。”

——Mike Bailey，俄勒冈州立大学教授

“最近出版的这本红宝书依然遵循了 OpenGL 的伟大传统：不断进化让它拥有了更为强大的力量和效率。书中包含了最前沿的接口标准和新特性的内容，以及对于应用在各行各业的现代 OpenGL 技术的脚踏实地的讲解。红宝书依然是我的公司中所有新员工的必备参考书。还有其他任何一本书可以说的上是必备的指南书吗？它让我喜极而泣，让我觉得无与伦比——我会一遍又一遍地阅读这本书。”

——Bob Kuehne，Blue Newt Software 总裁

“OpenGL 在这 20 年来已经有了巨大的发展。这次的修订版是一本学习使用现代 OpenGL 的实用指南书。现代 OpenGL 侧重于着色器的使用，而这一版的编程指南准确地对应了这一点，它在第 2 章对着色器进行了深入的叙述。而后继的章节里，它继续深入到方方面面，从纹理到计算着色器。无论你对 OpenGL 了解多少，或者你准备深入到何种程度，只要你准备开始编写 OpenGL 程序，你就一定需要《OpenGL 编程指南》这本手边书。”

——Marc Olano，UMBC 副教授

“如果你正在寻找有关 OpenGL 最新版的编程权威指南，那么你已经找到了。本书的作者深入参与了 OpenGL 4.3 标准的创立，而这本书中恰恰包含了你所需要了解的一切，它将使用一种清晰、富有逻辑性和见解性的方式，介绍这个行业领先的 API 标准的最新知识。”

——Neil Trevett，Khronos Group 总裁

译者序 *The Translator's Words*

本书是《OpenGL Programming Guide, Eighth Edition》的修订版本，也就是知名的“OpenGL 红宝书”的第9版。由于API接口的颠覆性变化，以及第8版写作者的大胆革新和推翻重写，让很多原本自认为经验丰富的从业者（包括笔者在内）对OpenGL这个已经存续了20余年的图形API一下子有了几分陌生感。而第9版的及时发布修正了之前的很多结构问题和概念错误，让全新的OpenGL架构更加明晰地展现在读者面前，不过对于那些早就熟悉了传统兼容模式（compatible profile）的开发者，以及原本对图形学一无所知急需精进的初学者来说，新的OpenGL看起来依然高不可攀，需要太长的时间去消化、实验、理解，直至应用。

随着图形硬件的飞速演进，以及虚拟现实（VR）和增强现实（AR）技术在这几年的火热发展，支持可视化编辑、多种脚本语言以及各种逼真渲染效果和优化算法的商业图形引擎（例如Unity和Unreal）开始逐渐占据主要的市场份额。众多开发者开始更多地着眼于如何实现优质的内容，而不是纠结于底层接口的封装、渲染批次的优化，或者光影效果的叠加与取舍。强大的中间层引擎解放了更多人的头脑，让他们不必把时间消耗在大量的底层逻辑处理、状态切换、数据管理以及跨平台测试上——这显然是一件好事，但是这也带来了一个新的话题，学习和熟悉底层图形接口——OpenGL或者DirectX，包括新的Vulkan，是不是已经不重要了？

笔者以为不然。事实上，大多数Unity和Unreal开发者经常遇到的那些问题，例如Draw Call的优化、纹理的分类和特性、相机参数的管理、着色器的编写等，依然都需要回归到最本质的图形学问题的范畴，回归到OpenGL接口所定义和执行的阶段。对于一名初学者而言，具备完善的可视化界面和快速搭建场景能力的商业引擎是一个入门的不错阶梯。他们可以避开枯燥的接口和数学概念的学习，快乐地做一些自己想要做的内容，例如简易的电脑游戏，或者一个可以用VR头盔观看的全景画面。在那之后，他们有必要带着一些

也许是一知半解的理念回归，重新去理解底层接口和定义，这些实现方案与底层硬件密不可分的关系，以及与他们曾经遇到的问题的千丝万缕的联系。这样的学习过程才是事半功倍的，也是一名开发者从爱好和浅显应用开发的阶段迈向资深图形工作者阶段的重要历程。而从繁花乱象中返璞归真，再去理解图形学中各种晦涩术语和复杂公式的时候，想必很多读者也会有一番全新的感受了。希望到那个时候，本书将会再次成为你的良师益友。

本书第9版的翻译和修订工作由王锐完成，并且根据章节需要重新翻译了第6章和附录A的部分。参与过本书前一版（第8版）翻译的还有郭华、苏明南、张静、王凯、陈节、龙海鹰和毕玉玲。感谢他们的辛苦付出，也感谢机械工业出版社华章公司的编辑老师们的信任与帮助！

王锐

2017年4月

前 言 *Preface*

OpenGL 图形系统是图形硬件的一种软件接口（GL 表示 Graphics Library，即图形库）。它使得用户可以创建交互式的程序以产生运动的三维对象的颜色图像。通过 OpenGL，我们可以使用计算机图形学技术产生逼真的图像，或者通过一些虚构的方式产生虚拟的图像。这本指南将告诉你如何使用 OpenGL 图形系统进行编程，得到你所期望的视觉效果。

本书的主要内容

本书中包含以下章节：

- 第 1 章对 OpenGL 可以完成的工作进行了概览。它还提供了一个简单的 OpenGL 程序并解释了一些本质性的编程细节，它们可能会用于后续的章节中。
- 第 2 章讨论了 OpenGL 中最主要的特性——着色语言和 SPIR-V，并介绍了它们在应用程序中的初始化和使用方法。
- 第 3 章介绍了使用 OpenGL 进行几何体绘制的各种方法，以及一些可以让渲染更为高效的优化手段。
- 第 4 章解释了 OpenGL 对于颜色的处理过程，包括像素的处理、缓存的管理，以及像素处理相关的渲染技术。
- 第 5 章给出了将三维场景表现在一个二维计算机屏幕上的操作细节，包括各种几何投影类型的数学原理和着色器操作。
- 第 6 章讨论了将几何模型与图像结合来创建真实的、高质量的三维模型的方法。
- 第 7 章介绍了计算机图形的光照效果模拟方法，主要是这类方法在可编程着色器中的实现。
- 第 8 章介绍了使用可编程着色器生成纹理和其他表面效果的方法细节，从而增强真实感和其他的渲染特效。

- 第 9 章解释了 OpenGL 管理和细分几何表面的着色器功能。
- 第 10 章介绍了在 OpenGL 渲染流水线中使用着色器进行几何体图元修改的一种特殊技术。
- 第 11 章介绍了使用 OpenGL 帧缓存和缓存内存实现高级渲染技术和非图形学应用的相关方法。
- 第 12 章介绍了最新的着色器阶段，将通用计算的方法融合到 OpenGL 的渲染管线当中。此外，我们也提供了一系列作为参考的附录内容。
- 附录 A 介绍了本书示例程序中用到的一些第三方支持库。GLFW 是可移植的，它可以用来实现更简短也更加可读的代码示例。而 GL3W 负责处理应用程序与 OpenGL 函数之间的绑定关系。
- 附录 B 介绍了 OpenGL 体系中的其他 API，包括用于嵌入式和移动平台系统的 OpenGL ES，以及用于 Web 浏览器内的交互式 3D 应用程序的 WebGL。
- 附录 C 提供了有关 OpenGL 着色语言的详细参考文档。
- 附录 D 列出了 OpenGL 维护的所有状态变量，并介绍了获取其值的方法。
- 附录 E 介绍了矩阵变换相关的一些数学方法。
- 附录 F 对于 OpenGL 中所用到的浮点数格式做出了概述。
- 附录 G 介绍了 OpenGL 中最新的调试特性。
- 附录 H 给出了有关 uniform 缓存的使用的参考文档，其中使用了 OpenGL 定义的标准内存布局。

本版中的新特性

本版《OpenGL 编程指南》已经针对最新的 OpenGL 4.5 进行了修订和更新。我们知道上一版是对以前版本的《OpenGL 编程指南》内容的一次颠覆，而这一版则是在此基础上进行了校对，修订了错误，重写了一些内容以便让读者更加愉悦地阅读。从程序开发的角度来说，OpenGL 4.5 版本所带来的最重大的特性变更就是直接状态访问（direct state access），这是对 OpenGL 程序开发模型和对象访问机制的一次重大革新。此外，我们也继续尝试将越来越多的功能移植到图形处理器硬件中，因此本书将着重于对着色器功能和 GPU 处理机制的讲解。

需要在阅读本书之前掌握的知识

本书假设你已经了解了使用 C++ 语言进行编程的方法（我们将使用少量的 C++ 程序，如果你对 C 语言已经比较熟悉的话，应该会比较容易理解它们），并且具有一定的数学背景（几何、三角学、线性代数、微积分以及微分几何）。即使对计算机图形学技术没有太多的经

验或者一无所知，你也可以学习和理解本书中讨论的大部分内容。当然，计算机图形学是一个不断延展的学科，因此你也许还需要阅读以下补充内容来丰富自己的知识。

- 《Computer Graphics: Principles and Practice》第3版，John F. Hughes、Andries van Dam、Morgan McGuire、David F. Sklar、James D. Foley、Steven K. Feiner 和 Kurt Akeley 著 (Addison-Wesley, 2013 年出版)：这本书是有关计算机图形学的一本百科全书，它包含了大量有价值的信息，不过在阅读之前，你最好已经对这门学科有了一定的了解。
- 《OpenGL SuperBible: Comprehensive Tutorial and Reference》第7版，Graham Sellers、Richard S. Wright Jr.、Nicolas Haemel 著 (Addison-Wesley, 2015 年出版)：这本书采用教程的形式编写，即使是对计算机图形学一无所知的读者也可以从这本循循善诱的指导书中开始学习 OpenGL。
- 《OpenGL Insights》，Patrick Cozzi 和 Christophe Riccio 著 (A. K. Peters, 2012 年出版)：这是一本有关 OpenGL 高级技巧的论文集，包括一些资深开发者、研究者以及一线工作者的感悟。每篇文章都会专注于某一个特定的技术领域，而这本书也会成为从事相关行业的读者的极佳的灵感来源。

另一个可以有组织地进行系统学习的地方就是 OpenGL 网站。该网站包含软件、示例程序、文档、FAQ、讨论版以及新闻页面。如果想要搜索 OpenGL 相关问题的答案，那么这里是一个好的开始：

<http://www.opengl.org/>

此外，OpenGL 的官方网站中还包含了 OpenGL 最新版本对应的所有函数和着色语言语法的完整文档。这些网页内容完整地涵盖了《OpenGL Reference Manual》的内容，该书由 OpenGL Architecture Review Board 和 Addison-Wesley 出版。

OpenGL 是一个与硬件密切相关的编程接口标准，我们可能会在某一类特定的硬件上使用一个特定的 OpenGL 实现。本书将会介绍如何使用任意的 OpenGL 实现进行开发。但是，因为这些实现之间会存在细微的差异（包括性能上的差异，以及额外的特性支持），你可能需要阅读自己所用的特定设备实现所对应的补充文档。此外，某个特定实现的供应商网站上，也可能也会提供一些 OpenGL 相关的功能、工具包、编程和调试支持、窗口组件、示例代码以及演示程序。

如何获取示例代码

本书中包含很多示例程序，它们演示了特定 OpenGL 编程技术的用法。本书的读者

群体在计算机图形学和 OpenGL 方面可能有着巨大的经验差异，有的人是新手，而有的人是多年的老手，因此这些章节里给出的示例都会使用最简单的方法去实现一个特定的渲染形式，并且全部使用 OpenGL 4.5 版本的接口。这么做主要是为了确保那些刚开始学习 OpenGL 的读者也能够顺利地阅读相关的内容。对于那些已经有了足够的经验，只是希望了解最新的 API 特性实现的读者，我们首先感谢你能够耐心阅读本书前面的内容，之后你可以访问我们的网站：

<http://www.opengl-redbook.com/>

在这里你会找到本书中所有示例的源代码，它们均使用最新的特性进行实现，而后文的讨论中也会涉及从一个 OpenGL 版本移植到另一个版本所需的修改。

本书中的所有程序都使用了 GLFW 工具库，它最初的作者是 Marcus Geelnard，现在由 Camilla Berglund 维护。GLFW 是一个持续在进行改进的开源项目，你可以在下面的地址里找到 GLFW 的项目页面：

<http://www.glfw.org/>

你可以在这个网站中找到相应的代码和二进制程序。

本书第 1 章和附录 A 中介绍了更多有关 GLFW 库的信息。我们可以在 OpenGL 网站的资源页面找到更多帮助你学习和使用 OpenGL 与 GLFW 的资源：

<http://www.opengl.org/resources/>

OpenGL 的很多实现也包含了一些系统相关的代码实例。这些源代码可能是你实现程序时最好的资源，因为它们已经针对系统进行了优化。你可以阅读与自己的系统相关的 OpenGL 文档来了解如何获取这些代码实例。

勘误

即使在本书的出版期间，OpenGL 也是不断更新的：有一些错误被修正，并且标准文档中也做出了澄清，同时还有新的标准被发布。我们将在网站 <http://www.opengl-redbook.com/> 上维护一个错误和更新列表，同时我们也会提供一些功能让用户提交自己发现的错误。如果你发现了本书的错误，我们首先向你郑重道歉，并且非常感谢你的报告。我们将尽快对其进行更正。

致 谢 *Acknowledgements*

John Kessenich

感谢 Graham 完成了很多写作的工作。感谢 Alison 牺牲了她的周末时间参与到这个项目中，并且帮助我将各个部分整合到一起。同时感谢 Google 对我的时间规划提供了便利和支持。最后，我要特别感谢 Khronos 一直以来对 OpenGL 的完美推行，并且对 Neil Trevett 和 Barthold Lichtenbelt 等人致以最高的敬意。

Graham Sellers

感谢我的妻子 Chris、我的孩子，以及我的其他家庭成员，他们让我能够安心地在每个早晨、夜晚、周末和放假时间进行写作。我也要感激我在 AMD 的同事以及一直持续开发 OpenGL 的 Khronos 成员们。而对于作为读者的你，同样感谢你的关注。来吧，这就是 OpenGL。

Dave Shreiner

首先也是最重要的一点，感谢 John 和 Graham，这两位了不起的作者完成了这样一部伟大的著作。我同样要感谢 Vicki 和 Cookie，他们在与我共同工作的时间里体现出了充分的支持和耐心。此外，对于我的父母 Bonnie 和 Bob，他们一直对我的努力满怀热忱，我作为他们的孩子感到非常幸运和骄傲。对于每一版著作的读者，以及全世界的 OpenGL 参与者们，我同样要致以最诚挚的谢意，希望你们能够使用 OpenGL 完成同样伟大的作品，高兴地开始图形编程吧！

推荐语
译者序
前 言
致 谢

第 1 章 OpenGL 概述	1
1.1 什么是 OpenGL	1
1.2 初识 OpenGL 程序	2
1.3 OpenGL 语法	6
1.4 OpenGL 渲染管线	7
1.4.1 准备向 OpenGL 传输数据	8
1.4.2 将数据传输到 OpenGL	8
1.4.3 顶点着色	9
1.4.4 细分着色	9
1.4.5 几何着色	9
1.4.6 图元装配	9
1.4.7 剪切	9
1.4.8 光栅化	9
1.4.9 片元着色	10
1.4.10 逐片元的操作	10
1.5 第一个程序：深入分析	11
1.5.1 进入 main() 函数	11

1.5.2 OpenGL 的初始化过程	12
1.5.3 第一次使用 OpenGL 进行 渲染	21

第 2 章 着色器基础	24
2.1 着色器与 OpenGL	25
2.2 OpenGL 的可编程管线	25
2.3 OpenGL 着色语言概述	27
2.3.1 使用 GLSL 构建着色器	27
2.3.2 存储限制符	33
2.3.3 语句	36
2.3.4 计算的不变性	40
2.3.5 着色器的预处理器	42
2.3.6 编译器的控制	44
2.3.7 全局着色器编译选项	44
2.4 数据块接口	45
2.4.1 uniform 块	45
2.4.2 指定着色器中的 uniform 块	46
2.4.3 从应用程序中访问 uniform 块	47
2.4.4 buffer 块	53
2.4.5 in/out 块、位置和分量	53
2.5 着色器的编译	54

2.6 着色器子程序	59	4.2 缓存及其用途	114
2.6.1 GLSL 的子程序设置	59	4.2.1 缓存的清除	116
2.6.2 选择着色器子程序	60	4.2.2 缓存的掩码	117
2.7 独立的着色器对象	62	4.3 颜色与 OpenGL	118
2.8 SPIR-V	64	4.3.1 颜色的表达与 OpenGL	118
2.8.1 选择 SPIR-V 的理由	64	4.3.2 平滑数据插值	119
2.8.2 SPIR-V 与 OpenGL	66	4.4 片元的测试与操作	121
2.8.3 使用 GLSL 在 OpenGL 中 生成 SPIR-V	67	4.4.1 剪切测试	122
2.8.4 Glslang	69	4.4.2 多重采样的片元操作	123
2.8.5 SPIR-V 中包含了什么	69	4.4.3 模板测试	124
第 3 章 OpenGL 绘制方式	70	4.4.4 模板的例子	125
3.1 OpenGL 图元	70	4.4.5 深度测试	127
3.1.1 点	71	4.4.6 融混	129
3.1.2 线、条带与循环线	72	4.4.7 逻辑操作	133
3.1.3 三角形、条带与扇面	72	4.4.8 遮挡查询	134
3.2 OpenGL 缓存数据	75	4.4.9 条件渲染	138
3.2.1 创建与分配缓存	75	4.5 多重采样	139
3.2.2 向缓存输入和输出数据	77	4.6 逐图元的反走样	141
3.2.3 访问缓存的内容	81	4.6.1 线段的反走样	142
3.2.4 丢弃缓存数据	86	4.6.2 多边形的反走样	143
3.3 顶点规范	86	4.7 像素数据的读取和拷贝	143
3.3.1 深入讨论 VertexAttribPointer	87	4.8 拷贝像素矩形	145
3.3.2 静态顶点属性的规范	90	第 5 章 视口变换、裁减、剪切与 反馈	147
3.4 OpenGL 的绘制命令	92	5.1 观察视图	148
3.4.1 图元的重启动	99	5.1.1 视图模型	148
3.4.2 多实例渲染	102	5.1.2 相机模型	148
第 4 章 颜色、像素和片元	112	5.1.3 正交视图模型	151
4.1 基本颜色理论	113	5.2 用户变换	152
		5.2.1 矩阵乘法的回顾	153

5.2.2	齐次坐标	155	6.7	采样器对象	218
5.2.3	线性变换与矩阵	157	6.8	纹理的使用	220
5.2.4	法线变换	167	6.8.1	纹理坐标	222
5.2.5	OpenGL 矩阵	168	6.8.2	排列纹理数据	225
5.3	OpenGL 变换	170	6.8.3	使用多重纹理	227
5.3.1	高级技巧: 用户裁减和剪切	172	6.9	复杂纹理类型	229
5.3.2	OpenGL 变换的控制	173	6.9.1	3D 纹理	229
5.4	transform feedback	174	6.9.2	纹理数组	230
5.4.1	transform feedback 对象	175	6.9.3	立方体映射纹理	231
5.4.2	transform feedback 缓存	176	6.9.4	阴影采样器	236
5.4.3	配置 transform feedback 的 变量	179	6.9.5	深度 - 模板纹理	237
5.4.4	transform feedback 的启动和 停止	185	6.9.6	缓存纹理	238
5.4.5	transform feedback 的示例: 粒子系统	187	6.10	纹理视图	240
第 6 章	纹理与帧缓存	192	6.11	滤波方式	243
6.1	纹理综述	193	6.11.1	线性滤波	243
6.2	基本纹理类型	194	6.11.2	使用和生成 mipmap	245
6.3	创建并初始化纹理	195	6.11.3	计算 mipmap 层次	249
6.4	指定纹理数据	200	6.11.4	mipmap 细节层次的控制	250
6.4.1	显式设置纹理数据	200	6.12	高级纹理查询函数	250
6.4.2	从缓存中加载纹理	202	6.12.1	显式的细节层次控制	250
6.4.3	从文件加载图像	203	6.12.2	显式的梯度设置	251
6.4.4	获取纹理数据	206	6.12.3	带有偏移参数的纹理获取 函数	251
6.4.5	纹理数据的排列布局	207	6.12.4	投影纹理	252
6.5	纹理格式	211	6.12.5	在着色器中执行纹理查询	253
6.5.1	内部格式	211	6.12.6	纹素收集	255
6.5.2	外部格式	214	6.12.7	组合功能的特殊函数	255
6.6	压缩纹理	216	6.13	无绑定纹理	256
			6.13.1	纹理句柄	256
			6.13.2	纹理驻留	257
			6.13.3	采样无绑定纹理	258

6.14	稀疏纹理	259	7.3.1	半球光照	303
6.14.1	稀疏纹理的数据提交	259	7.3.2	基于图像的光照	306
6.14.2	稀疏纹理的页面	260	7.3.3	球面光照	310
6.15	点精灵	261	7.4	阴影映射	313
6.15.1	纹理点精灵	261	7.4.1	创建一张阴影贴图	314
6.15.2	控制点的显示	263	7.4.2	使用阴影贴图	316
6.16	帧缓存对象	264			
6.17	渲染到纹理贴图	266	第8章 程序式纹理		320
6.17.1	抛弃渲染数据	268	8.1	程序式纹理	321
6.17.2	渲染缓存	269	8.1.1	规则的花纹	322
6.17.3	创建渲染缓存的存储空间	270	8.1.2	玩具球	328
6.17.4	帧缓存附件	272	8.1.3	晶格	335
6.17.5	帧缓存的完整性	274	8.1.4	程序式着色方法的总结	336
6.17.6	帧缓存的无效化	275	8.2	凹凸贴图映射	336
6.17.7	多重渲染缓存的同步写入	276	8.2.1	应用程序设置	338
6.17.8	选择颜色缓存来进行读写 操作	277	8.2.2	顶点着色器	340
6.17.9	双源融混	279	8.2.3	片元着色器	341
6.18	本章总结	281	8.2.4	法线贴图	343
6.18.1	纹理回顾	281	8.3	程序式纹理的反走样	343
6.18.2	纹理的最佳实践	281	8.3.1	走样的来源	344
			8.3.2	避免走样问题	345
			8.3.3	提高分辨率	346
第7章 光照与阴影		283	8.3.4	高频率的反走样	347
7.1	光照介绍	284	8.3.5	频率截断	354
7.2	经典光照模型	284	8.3.6	程序式反走样的总结	356
7.2.1	不同光源类型的片元着色器	285	8.4	噪声	356
7.2.2	将计算移到顶点着色器	294	8.4.1	噪声的定义	358
7.2.3	多个光源和材质	296	8.4.2	噪声纹理	362
7.2.4	光照坐标系	302	8.4.3	权衡	365
7.2.5	经典光照模型的局限	302	8.4.4	一个简单的噪声着色器	366
7.3	光照模型进阶	303	8.4.5	湍流	368

8.4.6	大理石	369	10.2	几何着色器的输入和输出	396
8.4.7	花岗岩	370	10.2.1	几何着色器的输入	396
8.4.8	木纹	370	10.2.2	特殊的几何着色器图元	399
8.4.9	噪声的总结	373	10.2.3	几何着色器的输出	403
8.5	更多信息	373	10.3	产生图元	405
第 9 章 细分着色器		375	10.3.1	几何体的裁减	405
9.1	细分着色器	375	10.3.2	几何体的扩充	406
9.2	细分面片	376	10.4	transform feedback 高级篇	410
9.3	细分控制着色器	377	10.4.1	多重输出流	411
9.3.1	生成输出面片的顶点	378	10.4.2	图元查询	414
9.3.2	细分控制着色器的变量	378	10.4.3	使用 transform feedback 的 结果	416
9.3.3	细分的控制	379	10.5	几何着色器的多实例化	423
9.4	细分计算着色器	383	10.6	多视口与分层渲染	425
9.4.1	设置图元生成域	384	10.6.1	视口索引	425
9.4.2	设置生成图元的面朝向	384	10.6.2	分层渲染	429
9.4.3	设置细分坐标的间隔	384	10.7	本章总结	432
9.4.4	更多的细分计算着色器 layout 选项	384	10.7.1	几何着色器回顾	433
9.4.5	设置顶点的位置	385	10.7.2	几何着色器的最佳实践	433
9.4.6	细分计算着色器的变量	385	第 11 章 内存		435
9.5	细分实例：茶壶	386	11.1	使用纹理存储通用数据	436
9.5.1	处理面片输入顶点	386	11.1.1	将纹理绑定到图像单元	440
9.5.2	计算茶壶的细分坐标	387	11.1.2	图像数据的读取和写入	441
9.6	更多的细分技术	389	11.2	着色器存储缓存对象	445
9.6.1	视口相关的细分	389	11.3	原子操作和同步	446
9.6.2	细分的共享边与裂缝	391	11.3.1	图像的原子操作	446
9.6.3	置换贴图映射	392	11.3.2	缓存的原子操作	454
第 10 章 几何着色器		393	11.3.3	同步对象	455
10.1	创建几何着色器	394	11.3.4	图像限定符和屏障	459
			11.3.5	高性能的原子计数器	467

11.4 示例：顺序无关的透明	470	12.5.2 计算着色器的最佳实践	500
11.4.1 工作原理	471	附录 A 第三方支持库	502
11.4.2 初始化	472	附录 B OpenGL ES 与 WebGL	514
11.4.3 渲染	473	附录 C 内置 GLSL 变量与函数	525
11.4.4 排序和融混	477	附录 D 状态变量	576
11.4.5 结果	480	附录 E 齐次坐标与变换矩阵	620
第 12 章 计算着色器	481	附录 F 纹理、帧缓存与渲染缓存 的浮点格式	625
12.1 概述	481	附录 G OpenGL 程序的调试与 优化	631
12.2 工作组及其执行	482	附录 H 缓存对象的布局	645
12.3 通信与同步	487	术语表	648
12.3.1 通信	488		
12.3.2 同步	489		
12.4 示例	490		
12.4.1 物理模拟	491		
12.4.2 图像处理	496		
12.5 本章总结	500		
12.5.1 计算着色器回顾	500		