

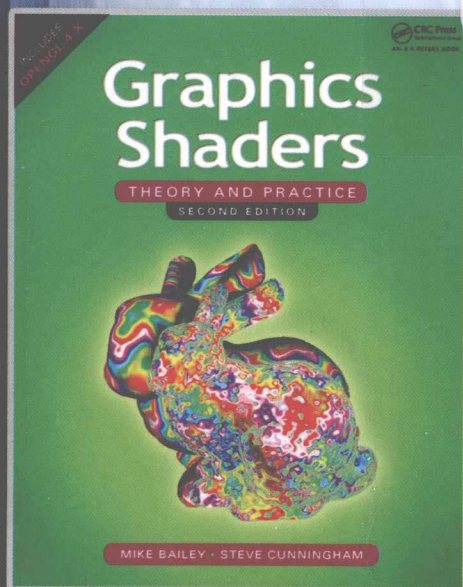


CRC Press
Taylor & Francis Group

图形着色器——理论与实践 (第2版)

Mike Bailey
Steve Cunningham
刘鹏 著
译

Graphics Shaders, Second Edition



清华大学出版社

图形着色器——理论与实践

(第2版)

Mike Bailey

Steve Cunningham 著

刘 鹏 译

清华大学出版社

北 京

内 容 简 介

本书详细阐述了与着色器程序设计相关的高效解决方案及相应的数据结构和算法,主要包括固定功能管线、OpenGL着色器的发展、着色器基本概念、使用glman、GLSL着色器语言、光照、顶点着色器、片元着色器和表面外观、片元着色器中的表面纹理、噪声、基于着色器的图像处理、几何着色器的概念和示例、细分着色器、GLSL API、基于着色器的科学可视化计算以及着色器应用等内容。此外,本书还提供了相应的算法、代码以及伪代码,以帮助读者进一步理解相关方案的实现过程。

本书适合作为高等院校计算机及相关专业的教材和教学参考书,也可作为相关开发人员的自学教材和参考手册。

Graphics Shaders 2nd Edition/by Mike Bailey,Steve Cunningham/ISBN:978-1-56881-434-6

Copyright© 2012 by CRC Press.

Authorized translation from English language edition published by CRC Press, part of Taylor & Francis Group LLC;All rights reserved;

本书原版由Taylor & Francis出版集团旗下CRC出版公司出版,并经其授权翻译出版。版权所有,侵权必究。

Tsinghua University Press is authorized to publish and distribute exclusively the Chinese(Simplified Characters) language edition.This edition is authorized for sale throughout Mainland of China.No part of the publication may be reproduced or distributed by any means,or stored in a database or retrieval system,without the prior written permission of the publisher.

本书中文简体翻译版授权清华大学出版社独家出版并限在中国大陆地区销售。未经出版者书面许可,不得以任何方式复制或发行本书的任何部分。

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书封面贴有Taylor & Francis公司防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

图形着色器——理论与实践(第2版)/(美)贝利(Bailey,M.), (美)坎宁安(Cunningham,S.)著;刘鹏译.
—北京:清华大学出版社,2013.3

(书名原文:Graphics shaders:theory and practice:2nd edition)

ISBN 978-7-302-31599-5

I. ①图… II. ①贝… ②坎… ③刘… III. ①计算机图形学 IV. ①TP391.41

中国版本图书馆CIP数据核字(2013)第030647号

责任编辑:赵洛育

封面设计:刘超

版式设计:文森时代

责任校对:柴燕

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京鑫丰华彩印有限公司

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:203mm×260mm

印 张:22.5

字 数:597千字

版 次:2013年3月第1版

印 次:2013年3月第1次印刷

印 数:1~4000

定 价:79.00元

产品编号:046291-01

译者序

着色器应用于计算机图形学领域，指一组供计算机图形资源在执行渲染任务时使用的指令。程序员将着色器应用于图形处理器（GPU）的可编程管线，来实现三维应用程序。这样的图形处理器有别于传统的固定管线处理器，为GPU编程带来更高的灵活性和适应性。

本书详细阐述了着色器程序设计方案，涵盖了丰富的内容，主要包括固定功能管线、OpenGL着色器的发展、着色器基本概念、使用glman、GLSL着色器语言、光照、顶点着色器、片元着色器和表面外观、片元着色器中的表面纹理、噪声、基于着色器的图像处理、几何着色器的概念和示例、细分着色器、GLSL API、基于着色器的科学可视化计算以及着色器应用等内容。值得一提的是，本书并非一本纯理论书籍，除了对相关内容进行全面、系统的讲解以外，其设计思想、数据结构和算法均辅以对应的代码示例，以帮助读者进一步理解计算方案的实现过程。

本书深入着色器这一核心问题的内部，内容涉及数学、物理、数据结构、算法、计算机图形学以及美学等知识。世界是如此的美好，它们都是了解世界、模拟世界甚至是创造世界的重要知识。基于着色器的程序设计使我们重返昔日时光，唤起我们的回忆并重新审视某些古老的问题。本书正是向读者传达这样的信息：我们所学习的知识是如此的重要，而大多数人却从未认识到这一点，并将它们遗忘。这里是我们重新认识、应用所学知识的地方，一切都是真实的——这的确令人激动！作为一名读者，此刻，你值得拥有！

在本书的翻译过程中，除刘鹏外，翁桂荣、康熙丽、孙颖、万立梅、陈丽丽、龙清伟、邵利萍、高影、安娇娇、李霞、刘笑彤、周宏丽、魏晶、李芳芳、张凤欣、高阳、胡艳娇、商婵婵、武变琴、顾庆娟、蒋丽丽、张超、赵雅利、吕莹莹、贾亚飞、古庆梅等人也参与了部分翻译工作，在此一并表示感谢。

由于译者水平有限，书中难免有疏漏和不妥之处，恳请广大读者批评指正。

译者

序



非常高兴读者选择阅读本书。

计算机图形学是一个令人极度着迷的领域，其历史久远且一直处于快速发展中。鉴于计算机图形学综合了工程技术以及艺术设计，本人被其深深吸引。在计算机图形学工业中，具有工程技术背景的研发人员负责设计图形软件和硬件，并使得计算性能和图像质量获得了长足的进步。同时，这也激发了设计人员的灵感并开创了令人惊奇的视觉体验，内容涉及娱乐和教育。受此鼓励，工程技术人员不断改善硬件和软件环境，以满足艺术设计者日益增长的设计需求。工程研发人员与艺术设计者之间的这种相辅相成的关系从未止步，两者间的碰撞火花使得影片具有照片级效果、计算机游戏呈现出剧场般的体验感。

或许，读者对计算机图形学怀有莫大的兴趣；或许，读者的身份是一名工程开发人员，正在为基于计算机图形学的软件开发工具箱研发一种新的工具；或许，读者作为一名艺术家，迫切需要了解计算机图像的生成方式；或许，读者身兼开发人员和设计者的双重身份，脑海中已呈现出设计结果，且需要深入了解此种新型媒介，并将创意转化为最终产品……无论如何，本书都将是一部极具指导意义的专著。

本书由Mike Bailey和Steve Cunningham两位作者联袂完成，并释放了两种截然不同的“激情”。Mike Bailey和Steve Cunningham对计算机图形学怀有无比的热爱并热衷于教育事业，本书尝试将这两种感情进行完美结合且一并呈现给读者。

Mike和Steve在计算机图形学界所作出的贡献远不止于此。本人于19年前与Mike相识，并在计算机图形系统业界标准等问题上求教于他，且得到了他的欣然首肯。作为一名计算机图形学教授，Mike在教育界和图形界已然奋斗了30余载并获奖无数。每年，Mike都会向SIGGRAPH（美国计算机协会计算机绘图专业组）旗下的Introduction to Computer Graphics杂志发表相关文章。

Steve的经历也与此类似，他是SIGGRAPH教育委员会的联合创始人，在SIGGRAPH组织和SIGGRAPH大会中身兼多项要职（主要涉及计算机图形学领域）。SIGGRAPH于2004年授予Steve“美国计算机协会计算机绘图专业组”杰出贡献奖，以表彰他长期不懈的努力。Steve在计算机图形界的影响遍布全球，例如，他担任Eurographics教育委员会首届主席并被授予资深委员这一称号。

本书由两位作者联手打造，旨在揭示计算机图形学的真相，更为重要的是，Mike和Steve向读者阐述了问题的理解、记忆方式。

因此，本书的重要性和及时性不言而喻。OpenGL及其着色语言均符合业界标准，这也意味着，两者均得到了硬件厂商以及各种操作平台的大力支持，其中包括Mac、PC、Linux系统、工作站、主机系统、桌面系统、膝上型系统以及手持设备。该标准的目标十分简单，即简化程序员的开发任务，在不改变源代码的前提下，可在各种产品上发布代码。因此，可移植性可有效地分摊软件开发费用。也就是说，基于业界标准的软件产品可显著地提升其市场份额。

本书的另一个重要特征是，读者在编写着色器代码时，还将进一步了解计算机图形学的基础知识。据此，读者可轻松地尝试其他着色语言及其图形学示例。考虑到图形硬件的通用编程能力不断提升，以及硬件内建固定功能不断减少，着色器程序设计将变得异常重要。换言之，我们将重返昔日美好的时光，并在软件中再次体验计算机图形学的重大革新，以提前适应未来的通用图形编程环境。

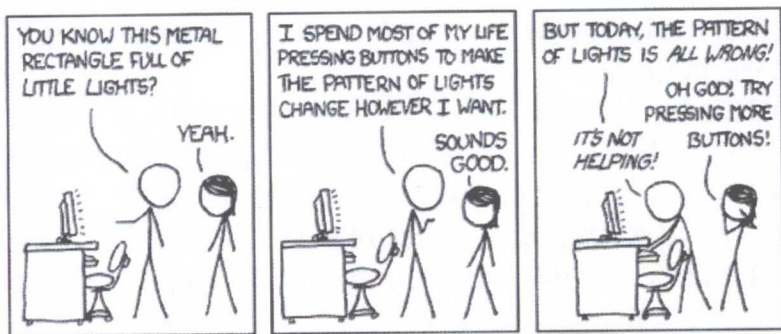
本书各章后均提供了相关练习，以帮助读者巩固图形学以及着色器编程知识，其目的在于：

- （1）阅读本书。
- （2）通过计算机图形学知识以及可编程的着色语言创造美好事物。
- （3）与他人分享作品和知识。
- （4）享受编程的乐趣——这是最为重要的。

Randi Rost

前言

下面的漫画作品是否唤起了你的某些回忆？



<http://xkcd.com>

你的脑海中是否萦绕着许多伟大、创新的理念却苦于无法实现？那么，你就是我们正在寻找的人，本书专门为你而打造。

欢迎阅读《图形着色器——理论与实践（第2版）》。顾名思义，本书将讨论着色器理论以及实践过程，并辅以大量的实现代码。为了方便读者阅读，本书采用全彩印刷，亦即多数示例均配以效果图像，以便于读者深入理解相关概念。因此，放慢你的脚步，静下心来开始阅读，并从中获得极大的享受吧！

与第1版相比，本书增加了100多幅图像。除此之外，本书还对以下内容进行了改版：

- (1) 本书遵循OpenGL最新规范，即OpenGL 4.x和GLSL 4.x0。
- (2) 全部示例代码根据最新GLSL语言标准加以编写。
- (3) 新增了与细分着色器相关的部分章节（及其示例代码）。
- (4) 各章增加了示例和练习。
- (5) 修改了大量的示意图，并将GLSL功能级别调整至4.x0。

(6) OpenGL架构评审委员会（ARB）弃用了OpenGL中的某些功能，但尚未将其完全去除。本书将对此加以阐述且对应代码集中体现了这一重要变化。另外，本书全部示例代码均遵循这一理念。在该策略的引导下，读者还可向OpenGL ES 2.0实现平稳过渡。

(7) 本书附录添加了与着色器对应的C++类，进而简化OpenGL着色器应用程序的编写过程，并对上述弃用策略提供某种支持。

可编程图形着色器经历了一段有趣的发展历程。在以往的岁月中，计算机图形内容均需要通过编程实现，相信某些读者对此仍有清晰的记忆。鉴于图像创建过程中存在可编程选项，这也意味着“可编程”往往包含某种贬义成分——事实并非如此。若期望生成某种图像结果，除了亲自动手编写程序之外，别无他法。

对于大多数图形实践者而言，计算机图形API可在一定程度上改变这一现状。优秀的API使得程序设计人员可轻松并出色地完成图形程序设计任务，期间，API功能项可掌控大部分图形处理操作。然而，程序设计人员也为此付出了一定的代价，即放弃API无法处理的某些功能项，如表面着色技术。当时（20世纪90年代），大多数API仅支持简单的平滑光照表面操作。

然而，无论是计算机图形开发社区还是高级图形程序员，对此都难以感到满意，这种满意度的欠缺分别来自软件和硬件方面。随着图形处理器的不断发展，基于特定功能的程序设计逐渐打破固定功能图形API这一桎梏，此类功能项逐渐形成自己的标准，包括隶属于OpenGL标准的GLSL着色器语言。可编程图形着色器已成为计算机图形学的关键特征，其中，程序可载入图形处理器，并执行早期固定功能管线标准之外的相关操作。

在教学以及学习过程中，计算机图形学也可采用并行方式实现。初学者通常会通过相关标准（如OpenGL）了解计算机图形学。当前，初学者还应理解可编程着色器所饰演的角色，并尝试编写和使用着色器。值得注意的是，基于着色器的程序设计往往会使我们重返昔日时光，并重新审视某些古老的图形学问题（例如，某些问题出现于20世纪70年代）。时至今日，读者可在OpenGL的全力支持下实现顶点和独立像素操作，这种体验往往来自于游戏、科学可视化计算以及通用图形计算。

本书可用于传统的教学课程中，其内容自成一体，向学生全面阐述了计算机图形学着色器程序设计。同时，本书还可作为基于固定功能API（尤其是OpenGL）的辅助读物。本书从整体上系统地讲述了着色器程序设计，并重点讲解了GLSL着色器语言。除此之外，本书还介绍了灵活、易用的编程工具glman，以帮助读者在应用程序外部开发、调试着色器。

本书可作为本科、研究生层次计算机图形学课程的高级教程，并假设读者已了解OpenGL以及相关图形学知识。本书重点讨论GLSL语言并统一使用OpenGL术语，对应内容同样适用于其他图形API，如Direct3D。鉴于着色器程序设计揭示了API所隐藏的事实真相，因而读者需要进一步了解某些基础理论，包括几何、光照以及着色等内容，相信读者会加深对此类问题的理解。同时，着色器程序设计还会帮助读者剖析API的工作原理，因而有必要对上述内容加以回顾。

本书采用GLSL作为示例着色器，其原因在于，OpenGL跨平台API具有较为广泛的应用范围且兼具高效性。鉴于着色器理念之间的相似性，相关概念同样适用于其他着色器API，如Cg和HLSL。本书首先通过着色器程序的基本角色和功能介绍固定功能图形管线，随后将在glman和应用程序环境下编写顶点着色器、片元着色器和几何着色器。

尽管按照扩展图形管线中的应用顺序考察着色器更符合逻辑性，即顶点着色器—几何着色器—片元着色器，但本书的处理方案则稍显不同，即按照使用频率这一顺序考察着色器（片元着色器—顶点着色器—几何着色器）。需要说明的是，该顺序无法实现正确的工作流，其原因在于，片元着色器所接收的数据源自顶点着色器，因而顶点着色器位于初始位置，随后依次为片元着色器、几何着色器和细分着色器。

本书整体内容脉络鲜明，第1章涵盖了后续章节所需要的背景知识，即基于固定功能的OpenGL图形管线。第2章讲述了OpenGL的演化历史。第3章讲解了顶点着色器、片元着色器和细分着色器的基本原理，以及基于GLSL着色器语言的多个示例。第4章以使用手册方式引入了glman软件。第5章讨论了GLSL着色器语言，并针对C语言比较了两者间的相似性和差异性。

第6~8章涉及顶点着色器和片元着色器等概念。其中，第6章阐述着色器的视点光照，并引入了

ADS（环境光、漫反射光和镜面光）光照函数，此类光照环境将在后续章节中多次提及。由于顶点着色器需要计算各顶点的光照，片元着色器负责计算各像素的光照，因而光照可视为顶点着色器和片元着色器的基础内容。第7章将重点探讨顶点着色器，对应内容主要集中于输入数据、输出数据以及顶点几何对象的调整方式。第8章将讨论片元着色器，包括输入数据、输出数据以及固定功能片元操作的替换方式。

第9~11章将分析片元着色器的特定功能项。第9章将描述片元着色器与纹理贴图之间的处理方式，包括凹凸贴图、立方体贴图以及场景的纹理渲染。第10章将阐述噪声函数及其在纹理和着色器中所扮演的角色。除此之外，第10章还介绍了noisegraph工具，以使读者可对1D和2D噪声函数属性进行调试。第11章则根据片元着色器工具讲述2D图像（纹理）的操控方式。

第12章讲述了几何着色器，包括几何着色器与顶点着色器和片元着色器之间的关系，以及几何着色器自身的功能项。本章通过多个示例强调几何着色器针对几何模型的扩展能力，以及细节级别的处理能力。第13章介绍了细分着色器，着重说明该类型着色器与几何着色器之间的相似性以及重大改进。

第14~16章集中讨论应用程序中的图形着色器。第14章定义了GLSL API，读者可在应用程序中编译、链接并使用着色器。另外，本章还分析了着色器程序的数据和图形状态信息的传递方式，并使用C++类封装着色器程序的整合操作。第15章阐述了着色器在可视化应用程序中的应用方式，并演示了多个可视化操作示例。第16章则考察了基于图形着色器的多个有趣示例。

附录部分显示了C++类的应用方案，以方便读者编写OpenGL应用程序并处理某些API弃用问题。

本书多数内容均较为直观，部分内容则包含一定的技巧性，因而应引起读者的足够重视。本书中的数学知识源自Nicholas Bourbaki所著教材，并采用  符号加以标识，进而强调该问题的重要性。

由于着色器函数仍处于不断发展中，因而新鲜事物以及对应处理方案层出不穷。相应地，本书采用  符号加以标识，希望读者对此有充分的认识。

通过本书的阅读，读者可进一步提升图形编程能力，并尽享图形编程的无限乐趣。OpenGL前进的步伐从未停止，不久的将来，着色器或许是几何对象以及渲染操作的唯一处理方式，我们有理由相信，读者将会再次认识到本书的价值所在。

致谢

本书的出版得益于多方的支持。这里，首先对Mike Bailey表达最真挚的谢意。

另外，还要感谢俄勒冈州立大学同仁们的支持，他们是Ron Adams、Bella Bose、Terri Fiez、Karti Mayaram、Ron Metoyer、Eric Mortensen、Cherri Pancake、Sinisa Todorovic以及Eugene Zhang。

鸣谢UCSD和OSU的同学们，他们才华横溢并自始至终伴我左右，他们是Tim Bauer、William Brendel、Guoning Chen、Matt Clothier、John Datuin、Will Dillon、Jonathan Dodge、Chuck Evans、Nick Gebbie、Kyle Hatcher、Nick Hogle、Chris Janik、Ankit Khare、Vasu Lakshmanan、Adam Leibel、Jessica McGregor、Daniel Moffitt、Chris Moore、Patrick Neill、Jonathan Palacios、Nadia Payet、Randy Rauwendaal、Dwayne Robinson、Avneet Sandhu、Nick Schultz、Sudarshanram Shetty、Evon Silvia、Ian South-Dickinson、Madhu Srinivasan、Michael Tichenor、Christophe Tome、Ben Tribelhorn、Ben Weiss以及Alex Wiggins。

感谢我的同事Ryan Bailey教授、Mike Gannis教授、Jenny Orr教授、Todd Shechter教授以及Justin

Spencer教授。

同时，还要感谢NVIDIA公司的鼎力支持，尤其是Gary Brown、Greg Gritton、Jen-Hsun Huang、David Kirk、Dave Luebke以及David Zier。

特别感谢来自AMD/ATI公司的Bill Licea-Kane。

感谢Randi Rost在3D Labs和Intel工作时给予的支持，感谢他所撰写的“橘皮书”，本书从中受益匪浅。

感谢派拉蒙电影公司授权使用本书中的图2.2。同时，皮克斯动画工作室提供了原始图像，在此一并表示感谢。

感谢xkcd.com授权使用前言首页处的漫画作品。

最后，还要感谢Alice Peters和Sarah Cutler为本书提供的建议以及所付出的努力；感谢本书的审稿人员，他们的宝贵意见使得本书内容得到了进一步的完善。

Mike Bailey
Corvallis, Oregon

Steve Cunningham
Coralville, Iowa

目 录

第1章 固定功能管线.....	1
传统的视见方案.....	1
顶点操作.....	1
管线的片元处理.....	3
图形管线中的状态.....	4
传统视见机制的实现方案.....	5
顶点处理操作.....	5
渲染处理操作.....	6
固定管线中的齐次坐标.....	9
顶点数组.....	10
本章小结.....	13
本章练习.....	14
第2章 OpenGL着色器的发展.....	16
着色器发展史.....	17
OpenGL着色器发展史.....	19
OpenGL 2.0/GLSL 1.10.....	19
OpenGL 3.x/GLSL 3.30.....	19
OpenGL 4.0/GLSL 4.00.....	20
OpenGL 4.x/GLSL 4.x0.....	21
内部原因.....	21
OpenGL ES.....	22
处理不同版本.....	22
本书方案.....	22
变量名命名规则.....	23
本章练习.....	23
第3章 着色器基本概念.....	24
图形管线中的着色器.....	24
顶点着色器.....	26
片元着色器.....	29
细分着色器.....	31
几何着色器.....	34

GLSL着色器语言.....	35
应用程序与着色器之间的数据传递.....	38
在应用程序中定义attribute变量.....	38
在应用程序中定义uniform变量.....	40
GLSL新版本之间的转换方式.....	42
本章练习.....	44
第4章 glman的使用.....	45
使用glman.....	46
加载GLIB文件.....	47
编辑GLIB文件和着色器文件.....	47
生成GLIB场景.....	47
窗口和视见操作.....	47
转换操作.....	48
定义几何对象.....	48
定义纹理.....	50
定义着色器名.....	50
其他命令.....	51
定义uniform变量.....	51
GLIB文件示例.....	53
纹理和噪声.....	54
使用纹理.....	55
使用噪声纹理.....	55
glman界面窗口功能.....	56
生成并显示场景的硬拷贝.....	56
全局场景转换.....	57
眼睛转换.....	57
对象拾取和转换.....	57
纹理转换.....	58
显示帧速率.....	58
其他功能项.....	59
本章练习.....	59
第5章 GLSL着色器语言.....	61
着色器语言的成因.....	61
图形卡功能项.....	62
通用GLSL语言.....	63
共享命名空间.....	64
函数扩展和操作符功能项.....	64

新增函数.....	64
新增变量类型.....	65
新增函数参数类型.....	65
具体实现.....	65
忽略的语言功能项.....	66
新增矩阵和向量类型.....	66
名称集.....	67
向量构造函数.....	67
矩阵和向量扩展函数.....	68
矩阵和向量的扩展操作.....	70
新增函数.....	71
混合操作.....	75
新增函数参数类型.....	75
const变量.....	76
兼容模式.....	76
定义兼容模式.....	76
OpenGL 2.1内建数据类型.....	77
综述.....	81
本章练习.....	81
第6章 光照.....	82
ADS光照模型.....	82
ADS光照模型函数.....	83
光源类型.....	85
固定光源.....	85
有向光源.....	86
聚光灯.....	86
构建基于着色器的光照机制.....	88
固定着色.....	88
平滑着色.....	89
Phong着色方案.....	89
各向异性着色.....	90
本章练习.....	91
第7章 顶点着色器.....	92
图形管线中的顶点着色器.....	93
顶点着色器的输入数据.....	93
源自顶点着色器的输出数据.....	94
几何体.....	94
顶点着色器之后的固定功能处理.....	96

顶点着色器和细分着色器之间的关系.....	96
顶点着色器和几何着色器之间的关系.....	96
利用顶点着色器替换固定功能图形操作.....	97
标准的顶点处理过程.....	97
基于顶点着色器的固定功能管线扩展.....	97
顶点修正.....	98
顶点着色器中的问题.....	100
生成法线.....	100
综述.....	101
本章练习.....	102
第8章 片元着色器和表面外观.....	103
片元着色器的基本功能.....	104
片元着色器的输入数据.....	104
片元着色器的in变量.....	105
坐标系.....	106
片元着色器处理机制.....	107
源自片元着色器的输出数据.....	107
利用片元处理器替换固定功能处理.....	107
着色机制.....	107
传统的纹理贴图.....	108
伪色.....	108
片元着色器后续步骤.....	110
其他着色器效果.....	110
像素丢弃.....	111
Phong着色.....	111
基于解析法线的着色机制.....	112
各向异性着色.....	113
基于数据驱动的色彩方案.....	114
使用其他数据生成图像.....	115
本章练习.....	116
第9章 片元着色器中的表面纹理.....	117
纹理坐标.....	117
传统的纹理贴图.....	117
GLSL纹理贴图.....	118
纹理上下文.....	120
固定功能管线中的纹理环境.....	121
纹理采样参数.....	121

采样器.....	122
过程纹理.....	122
凹凸贴图.....	127
立方体贴图.....	132
渲染至纹理.....	135
渲染至纹理 (glman多路渲染)	138
本章练习.....	140
第10章 噪声	141
噪声的基本概念.....	141
3种噪声类型: 数值、梯度以及数值+梯度.....	142
三次插值和五次插值.....	142
噪声方程.....	143
其他噪声.....	145
分形布朗运动 (FBM, $1/f$, 倍频程)	146
二维噪声和三维噪声.....	146
基于glman的噪声.....	148
基于内建GLSL函数使用噪声纹理.....	149
湍流.....	149
不同环境中的噪声示例.....	151
大理石着色器.....	153
云彩着色器.....	153
木材着色器.....	155
与噪声相关的高级话题.....	156
noisegraph应用程序.....	156
本章练习.....	157
第11章 基于着色器的图像处理	159
基本概念.....	160
单幅图像处理.....	160
亮度.....	160
CMYK转换.....	161
色调转换.....	163
图像过滤机制.....	165
图像模糊操作.....	166
色度键图像.....	167
立体视图.....	169
3D电视.....	171
边检测技术.....	173

浮雕效果.....	174
卡通着色器.....	176
艺术视觉效果.....	177
图像的翻转、旋转和扭曲.....	178
图像混合处理.....	182
基于固定基图像的图像混合操作.....	183
反像.....	183
亮度.....	184
对比度.....	184
图像的自身混合操作.....	185
饱和度.....	185
锐利度.....	186
不同图像间的混合操作.....	187
其他合成方案.....	188
图像渐变.....	190
内容提示.....	194
本章练习.....	195
第12章 几何着色器的概念和示例.....	198
几何着色器的功能.....	198
新型邻接图元.....	200
基于输入和输出变量的布局结构.....	201
新增OpenGL API函数.....	201
新增GLSL变量以及变量类型.....	203
顶点着色器或细分着色器与几何着色器之间的通信.....	203
几何着色器中的法线.....	204
应用示例.....	205
Bezier曲线.....	205
收缩三角形.....	207
球体细分.....	208
3D对象轮廓边.....	211
本章练习.....	213
第13章 细分着色器.....	214
细分着色器定义.....	214
细分着色器与几何着色器.....	215
细分着色器概念.....	216
设置细分级别时的问题.....	219
应用示例.....	219

等值线.....	219
Bezier曲面.....	222
球体细分.....	228
基于屏幕覆盖率的整球细分.....	233
PN三角形.....	236
综述.....	241
本章练习.....	241
第14章 GLSL API.....	243
OpenGL程序处理中的着色器.....	243
处理OpenGL扩展.....	244
GLSL着色器程序的创建方式.....	245
创建并编译着色器对象.....	245
CheckGLErrors()函数.....	247
创建、绑定、链接并激活着色器程序.....	248
生成着色器程序并绑定着色器对象.....	248
着色器程序的链接操作.....	249
激活着色器程序.....	250
向着色器传递数据.....	250
在应用程序中定义uniform变量.....	251
兼容模式下的uniform变量.....	253
在应用程序中定义attribute变量.....	254
兼容模式下的attribute变量.....	255
着色器程序的C++控制类.....	256
本章练习.....	257
第15章 基于着色器的科学可视化计算.....	258
基于图像的可视化技术.....	258
负像.....	258
图像边检测.....	259
卡通渲染.....	259
双曲线几何学.....	260
3D标量数据可视化计算.....	263
点云.....	264
剖切面.....	267
体探测.....	270
直接体渲染.....	271
其他转换函数.....	276
通过几何对象传递数据值.....	279