HZ BOOKS
华章科技

PEARSON

资深iOS开发专家根据OpenGL ES最新版本撰写，详细讲解了OpenGL ES与GLKit的结合使用

系统讲解OpenGL ES的核心概念、技术，以及iOS的图形机制，通过大量案例讲解了在iOS上进行OpenGL ES开发的方法和技巧

华章程序员书库

Learning OpenGL ES for iOS

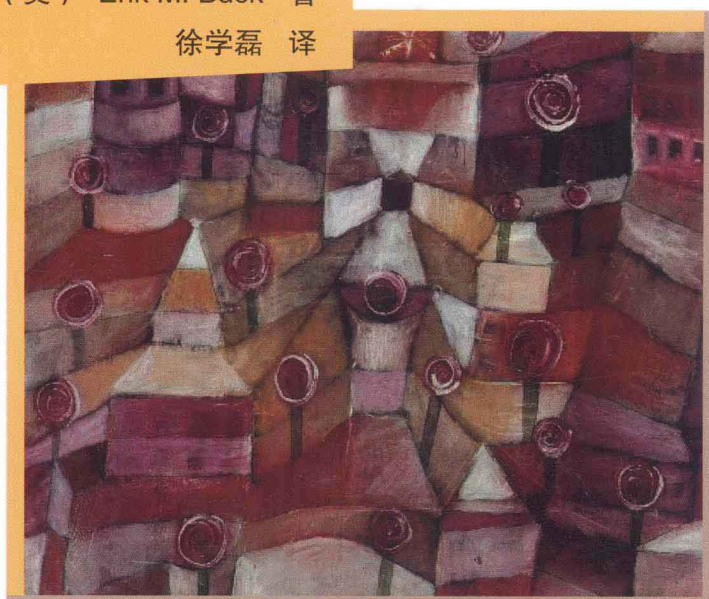
A Hands-On Guide to Modern 3D Graphics Programming

OpenGL ES应用开发实践指南

iOS卷

（美） Erik M. Buck 著

徐学磊 译



机械工业出版社
China Machine Press

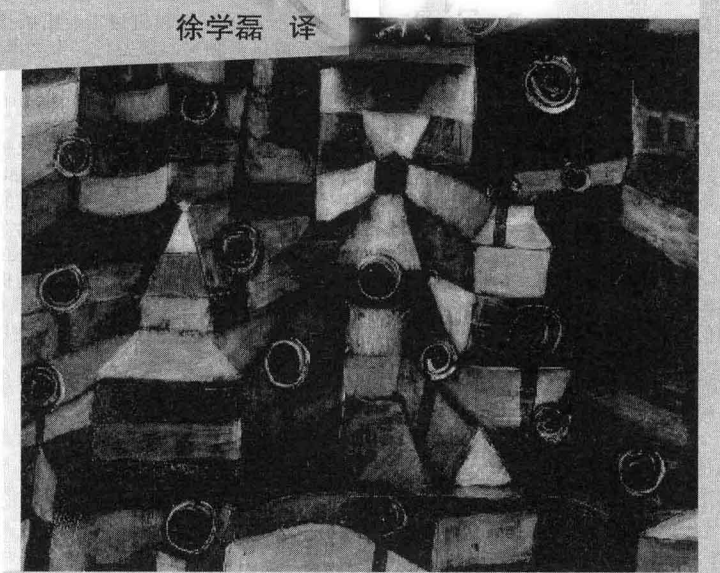
Learning OpenGL ES for iOS

A Hands-On Guide to Modern 3D Graphics Programming

OpenGL ES应用开发实践指南 iOS卷

（美） Erik M. Buck 著

徐学磊 译



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

OpenGL ES 应用开发实践指南：iOS 卷 /（美）巴克（Buck, E. M.）著；徐学磊译 . —北京：机械工业出版社，2013.6

（华章程序员书库）

书名原文：Learning OpenGL ES for iOS: A Hands-On Guide to Modern 3D Graphics Programming

ISBN 978-7-111-42867-1

I. O… II. ① 巴… ② 徐… III. ① 图形软件 – 指南 ② 移动电话机 – 应用程序 – 程序设计 – 指南 IV. ① TP391.41-62 ② TN929.53-62

中国版本图书馆 CIP 数据核字（2013）第 126145 号

本书版权登记号：图字：01-2012-7590

这是一本系统的具备实战性的 OpenGL ES 3D 图形开发指南。由资深 iOS 开发专家根据 OpenGL ES 最新版本撰写，不仅详细讲解了 OpenGL ES 与 GLKit 的结合使用，而且还系统讲解 OpenGL ES 的核心概念、技术，以及 iOS 的图形机制，并通过大量案例讲解了在 iOS 上进行 OpenGL ES 开发的方法和技巧。

全书共分 12 章。第 1 章介绍了使用嵌入式图形硬件绘制 3D 图形的最新方法；第 2 章讲解了如何使用苹果 Xcode 开发工具和 Cocoa Touch 面向对象的框架在 iPhone、iPod Touch 和 iPad 中开发包括 3D 图形的程序；第 3 章涵盖了纹理的底层概念和常用选项；第 4 章介绍灯光模拟背后的概念，以及利用 GLKit 并使用相对简单的应用代码演示灯光效果；第 5 章讲解并演示从任意视点渲染几何对象的技术；第 6 章介绍如何制作动画；第 7 章介绍了如何加载并使用模型；第 8 章讲解了特效的使用；第 9 章介绍能够提高 iOS 设备上 OpenGL ES 2.0 渲染性能的优化策略；第 10 章讲解了地形和拾取；第 11 章回顾了 3D 渲染所需的常见数学运算；第 12 章涵盖了一个结合地形渲染、天空盒、粒子系统、动画、变化视点、灯光、模型和碰撞检测技术的实例。

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：陈佳媛

三河市杨庄长鸣印刷装订厂印刷

2013 年 7 月第 1 版第 1 次印刷

186mm×240mm • 17.5 印张

标准书号：ISBN 978-7-111-42867-1

定 价：69.00 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）88378991 88361066 投稿热线：（010）88379604

购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

译 者 序

OpenGL ES 是桌面版 OpenGL 的一个子集，是现代移动设备绘图功能的基础。其中“ES”代表嵌入式系统。OpenGL ES 定义了嵌入式 3D 绘图的标准。GLKit 是苹果公司的 iOS 5 中引入的一个软件框架，这个框架简化了很多常用的编程任务，隐藏了 OpenGL ES 版本间的差异，使用这个框架可以提高软件开发的生产率。本书主要讲解的是 iOS 5 中的 GLKit 软件框架，以及 GLKit 所支持的 OpenGL ES 2.0 版本。

本书从基本概念开始，全面介绍了 OpenGL ES 计算机图形的核心概念以及 iOS 的图形机制，包括地形、特效、模型、动画、视点变化、纹理、灯光和优化等。本书例子丰富有趣，一般每章都有多个例子，并且源码结构合理，注释全面。本书会告诉你如何在苹果公司的 iOS 环境中充分地利用好 OpenGL ES。本书专注于最新版 OpenGL ES 的最新使用方法，因此可以让你避开散落在互联网上的不相关的、过时的，或者误导性的方法。本书的第 11 章为数学速查章节，非常方便，如果你是计算机图形的新手，可以先阅读这一章以帮助你更好地理解本书的内容。如果你熟悉桌面版 OpenGL，但是不了解 OpenGL ES 以及 GLKit，那么这本书也非常适合你。

本书的翻译经历了 3 个月，非常感谢华章公司的编辑们在翻译过程中给予我的支持和帮助。对于译者来说，能够给读者带来切实的帮助是我最大的荣幸。虽然在翻译的过程中竭力以信、达、雅为目标，但是由于水平有限，失误和遗漏在所难免，恳请读者批评指正。

译者

前　言

OpenGL ES 技术是苹果 iOS 设备（iPhone、iPod Touch 以及 iPad）上的用户界面和图形绘制能力的基础。“ES”代表嵌入式系统（Embedded System），这个术语适用于视频游戏机、飞机驾驶员座舱显示器，并且广泛适用于几乎所有生产商的手机。OpenGL ES 是桌面操作系统 OpenGL 版本的一个子集。因此，OpenGL ES 应用通常也适用于桌面系统。

本书介绍了最新图形编程，同时对 iOS 设备中 OpenGL ES 的有效使用做了简洁说明。书中有很多用于演示图形编程概念的例子程序。在网站 http://opengles.cosmicthump.com/ 上保存着很多例子和相关文章。本书对于从底层位操作到高级主题的图形技术都做了详细的解释。

学习图形编程的重大挑战体现在当你第一次试图整理散落在互联网上的成堆的误导性信息和过时的例子时。最初 OpenGL 是 1992 年的最先进图形工作站中的一个小型软件库。由于图形硬件改进得频繁且更新较快，以至于现在的移动设备已经胜过 OpenGL 刚出现时能够买到的最好的硬件。随着硬件技术的提高，OpenGL 设计者当时所做的一些折中方法和假设已不再有意义了。现在至少存在着 12 种不同的 OpenGL 标准，不过最新的 OpenGL ES 省略了很多对于以前版本中的常见技术的支持。不幸的是，在谷歌的搜索结果中还存在相当多的过时的代码和次优方案，以及几十年来形成的过时的经验。本书将主要关注最新、最高效的方法，以避免分心于过时且不相干的练习上。

读者对象

本书的读者包括学习编程的学生以及精通其他学科又想要学习图形的程序员。读者不需要有计算机图形的经验，但需要熟悉 C 或者 C++ 以及面向对象编程的概念。有 iOS、Objective-C 编程语言和 Cocoa Touch 框架的使用经验是最好的，但不是必需的。在学完本书后，你将有能力在你自己的 iOS 应用中使用高级计算机图形技术。

示例代码

本书提供的很多例子可以用作你自己的工程的起始点。可以从 http://opengles.cosmicthump.com/learning-opengl-es-sample-code/ 下载本书例子的源代码，这些源码使用 MIT 软件许可协议：http://www.opensource.org/licenses/mit-license.html。

这些例子是使用苹果的免费开发者工具建立的，用的是 Objective-C 编程语言，以及苹果的 Cocoa Touch 面向对象的软件框架。OpenGL ES 应用程序编程接口（API）由美国国家标准协会（ANSI）/ 国际标准化组织（ISO）C 编程语言的数据类型和函数组成。作为 ANSI/ISO C 的一个超集，Objective-C 程序原生地支持与 OpenGL ES 交互。

所有 iOS 应用都或多或少地依赖于 Cocoa Touch 框架，该框架是基于苹果的 Objective-C 语言的。一些开发者通过重用现存的用 C 或者 C++ 写就的跨平台库来最小化自己的应用与 Cocoa Touch 的融合。作为 UNIX 操作系统的一个派生物，iOS 包含了标准 C 库和 UNIX API，这使得移植跨平台代码到苹果设备上变得出人意料的简单。OpenGL ES 自身的一部分也是由跨平台 C 库组成的。尽管如此，不想使用 Cocoa Touch 和 Objective-C 的开发者几乎总是给自己帮倒忙。苹果面向对象的框架史无前例地提升了程序员的生产率。更重要的是，Cocoa Touch 提供了用户对 iOS 应用所期望的平台一体性和精良性。

本书包含了 Objective-C 和 Cocoa Touch。苹果的基于 Objective-C 的 GLKit 框架的强大和简洁是如此令人信服，以至于它明确地指出了图形编程的未来发展方向。如果不使用 GLKit 而只关注操作系统的底层 C 接口和 OpenGL ES，本书几乎无法声称自己教授的是最新技术。

Objective-C

与 ANSI/ISO C 一样，Objective-C 是一个非常小型的语言。有经验的 C 程序员通常会发现，他们可以在至多几个小时内很容易地学会 Objective-C。Objective-C 在开启了一个富有表现力的面向对象的编程风格的同时，最低限度地扩充了 C 语言。本书详述了图形编程，同时在需要的时候会对 Objective-C 语言的特性做适当的介绍。你并不需要精通 Objective-C 或者 Cocoa Touch，但是你需要熟悉 C 或者 C++ 以及面向对象编程的概念。你会发现使用 Objective-C 语言实现应用逻辑是非常容易和简洁的。Cocoa Touch 经常简化应用的设计，尤其是在响应用户输入的时候。

C++

ANSI/ISO C++ 编程语言是 ANSI/ISO C 的一个不是很完美的超集，但它几乎总是

可以自由地与 C 语言混合。OpenGL ES 与 C++ 可以无缝配合，并且 OpenGL 结构审查委员会（ARB）会监督 OpenGL ES 的规范，以保证其未来与 C++ 的兼容性。

C++ 编程语言是用于图形编程的最常见的编程语言之一。但是，C++ 是一个非常大型的编程语言，充满了惯用语法和精妙法则。对于 C++ 语言要达到中等掌握水平可能要花费数年的时间。使用 C++ 做图形编程有许多优势，例如，使用 C++ 操作符重载功能可以让图形程序中数学运算的表达更加简洁。

混合使用 C++ 与 Objective-C 代码并没有任何障碍。苹果开发者工具甚至支持 Objective-C++ 形式，这种形式允许在一个语句中混合使用 C++ 和 Objective-C 代码。但是 Objective-C 是 iOS 的主要编程语言。在苹果和第三方提供的几乎所有 iOS 示例代码中你都可以发现 Objective-C 的代码。如果你想使用 C++ 也是可以的，但这超出了本书讨论的范围。

使用 GLKit 作为导向

本书通过对苹果 GLKit 的探索来讲解图形编程的最新概念。在一些情况下，某些章节会通过实现 GLKit 的部分对象来讲解和演示这些概念。这样做有几个目的：使用 GLKit 来简化启动项目所需的步骤。在第 2 章末尾你会建立起 3 个 OpenGL ES 应用，并在你的 iOS 设备上运行。一章接着一章，逐渐创建彼此关联的很多主题，最终创建一个可重用的知识和代码的基础结构。当付出努力以从零开始创建时，可以帮助你获取一个想要的最终结果的清晰概念。为了获得有价值的最终结果，GLKit 设置了一个高质量的现代基准。

本书会消除你关于怎么使用 OpenGL ES 来实现和扩展 GLKit 的所有疑云。学完本书后，你会成为一个 GLKit 专家，彻底理解 GLKit，同时拥有在你的 iOS 应用中使用 GLKit 的能力。GLKit 演示了当前对于 OpenGL ES 的最好做法，同时如果你需要，甚至可以将其作为你自己的跨平台库的一个模板。

致谢

写一本书需要很多人的支持。首先，感谢我的妻子 Michelle，以及我的孩子 Joshua、Emma 和 Jacob，谢谢你们的理解和支持。其次，感谢出版社的编辑们，他们为本书的写作提供了非常宝贵的建议。最后，感谢那些在学术上、专业上、精神上、道德上和艺术上将影响我一生的人。

推荐阅读

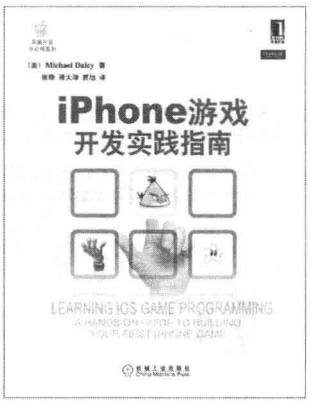

Cocoa设计模式
COCOA DESIGN PATTERNS

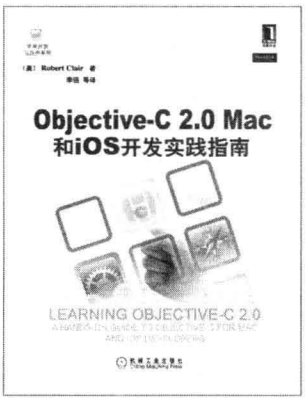

iPad
应用开发实战

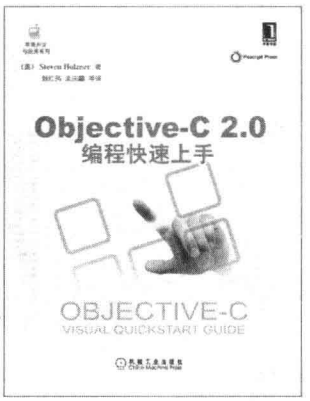

iPhone 3D Game Programming All in One
iPhone 3D游戏
编程指南

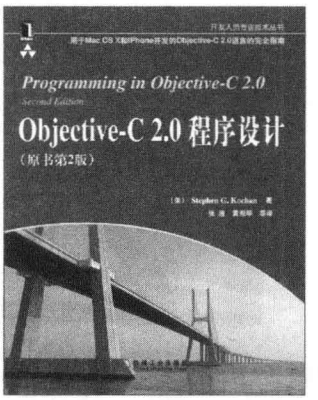

iPhone
应用开发
从入门到精通


iPhone
应用用户体验设计
实战与案例
Designing the iPhone
User Experience


iPhone游戏
开发实践指南
LEARNING IOS GAME PROGRAMMING


Objective-C 2.0 Mac
和iOS开发实践指南
LEARNING OBJECTIVE-C 2.0


Objective-C 2.0
编程快速上手
OBJECTIVE-C
VISUAL QUICKSTART GUIDE


Programming in Objective-C 2.0
Second Edition
Objective-C 2.0 程序设计
（原书第2版）

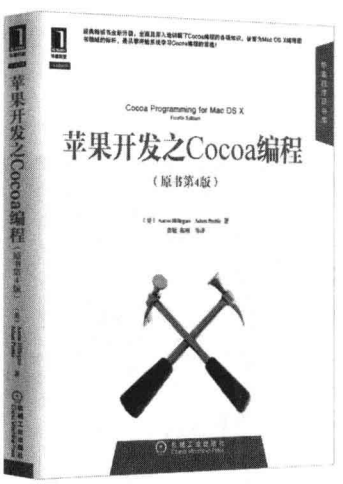
推荐阅读

内容全面，系统讲解开发企业级iPhone应用所需掌握的各项核心技术，以及各种工具和框架的用法，包含大量技巧和最佳实践

实战性强，不仅为各个知识点精心设计了能辅助读者理解的小案例，而且还有能指导读者实践的大案例，具备极强的可操作性

Amazon五星级畅销书，作者权威，在全球iOS/Mac开发者社区享有盛誉！

完美地展现了测试驱动开发方法与iOS开发的结合，能使iOS开发者在产品需求、软件设计、测试有效性与开发效率之间达成很好的平衡



经典畅销书全新升级，系统且深入地讲解了Cocoa编程的各项知识，被誉为Mac OS X编程图书领域的标杆，被公认为从零开始学习Cocoa的首选！

资深iOS开发工程师撰写，Amazon五星级畅销书，国际Mac和iPhone开发者社区CocoaHeads联合创始人Mark Dalrymple等数位专家联袂推荐！

目　录

第 1 章　使用现代移动图形硬件

本章介绍使用嵌入式图形硬件绘制 3 维（3D）图形的最新方法。嵌入式系统涵盖了范围广泛的设备，从飞机驾驶员座舱显示器到自动售货机。绝大多数具有 3D 功能的嵌入式系统都是手持电脑，比如苹果的 iPhone、iPod Touch、iPad，或者是基于谷歌的 Android 操作系统的手机。索尼、任天堂及其他的手持设备也具备强大的 3D 图形能力。

用于嵌入式系统的 OpenGL（OpenGL ES）定义了嵌入式 3D 图形的标准。基于 iOS 5 的 iPhone、iPod Touch 以及 iPad 设备支持的是 OpenGL ES 2.0。苹果的设备也支持旧的 OpenGL ES 1.1 版本。iOS 5 引入了 GLKit 软件框架，这个框架简化了很多常用的编程任务，同时部分隐藏了所支持的这两个 OpenGL ES 版本间的差异。本书主要关注带有 GLKit 的 iOS 5 所支持的 OpenGL ES 2.0 版本。

为了能在 ANSI C 编程语言中使用，OpenGL ES 定义了一个应用程序编程接口。通常用来开发苹果产品的 C++ 和 Objective-C 编程语言可以与 ANSI C 无缝交互。特定的转换层或者粘合层的存在使 OpenGL ES 可以用在 JavaScript 和 Python 中。新兴的 Web 编程标准，如非营利性 Web3D 联盟的 WebGL 标准，也正准备在网页上实现对于 OpenGL ES API 的跨平台标准化访问。本书中讲解的 3D 图形概念适用于所有具有 3D 功能的嵌入式系统。

本章会讲解使用 OpenGL ES 和 iOS 5 来实现 3D 图形的一般方法，但并不会深入特定的编程细节。最新的 3D 图形硬件加速是所有高级移动产品的可视化效果的基础。学习本章是能够从移动硬件中萃取出最好的 3D 图形和可视效果的第一步。

1.1　3D 渲染

图形处理单元（GPU）就是能够结合几何、颜色、灯光和其他数据而产生一个屏幕图像的硬件组件。屏幕只有 2 维，因此显示 3D 数据的技巧就在于产生能够迷惑眼睛使其看到丢失的第 3 维的一个图像，参见图 1-1 中的例子。

用 3D 数据生成一个 2D 图像的过程就叫渲染。在计算机上显示的图片是由矩形的颜色点组成的，这些矩形的颜色点叫做像素。图 1-2 放大了图像中的一部分来显示

单独的像素。如果通过放大镜仔细观察显示器，你会看到每个像素都是由 3 个颜色元素组成的，即一个红点、一个绿点和一个蓝点。图 1-2 还显示了一个进一步放大的像素来描述单独颜色的元素。在一个全彩色的显示器上，一个像素通常含有红、绿、蓝 3 个元素，但这些元素的排列样式可能与图 1-2 中显示的一个挨一个的排列方式不同。

图 1-1　用 3D 数据生成的一个示例图像

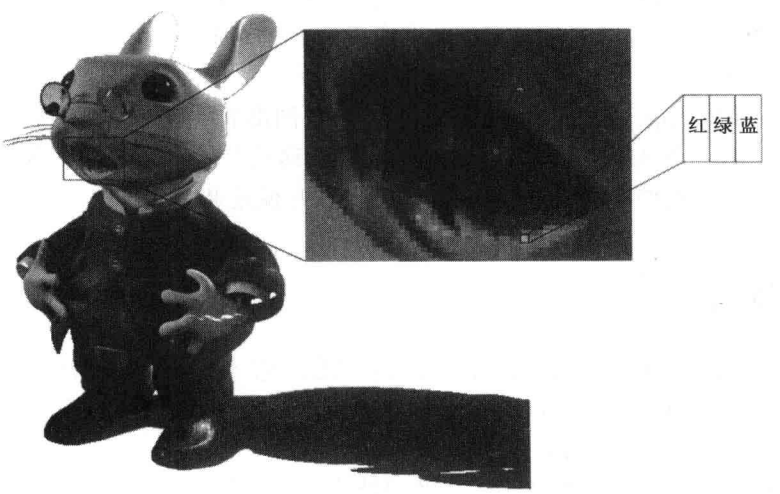


图 1-2　图像是由像素构成的，每个像素有红、绿、蓝 3 个元素

图像是以每个像素至少包含 3 个值的一个数组存储在电脑的存储器中的。第一个值指定了像素的红色元素的强度，第二个值代表绿色强度，第三个值是蓝色强度。一个包含 10000 个像素的图像能够以一个拥有 30000 个强度值的数组的形式存储在存储器中，像素的 3 个元素每个需要一个值。以不同的强度结合红绿蓝 3 个值就足以产生彩虹的所有颜色。如果 3 个元素的强度都是 0，结果颜色就是黑色。如果 3 个颜色都是强度的最大值，结果颜色就是白色。黄色是通过丢掉蓝色并混合红色和绿色得到的。图 1-3 中的 Mac OS X 标准颜色面板用户界面包含能够调节相关的红、绿、蓝强度的图像滑块。

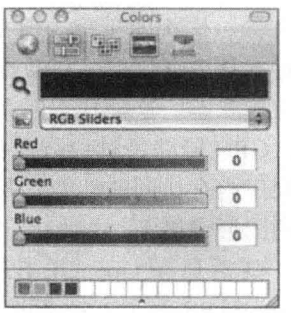


图 1-3　用于调节红、绿、蓝颜色元素强度的用户界面

渲染 3D 数据为一个 2D 图像通常发生在几个不同的步骤中，包括设置图像中的每个像素的红、绿、蓝颜色强度的计算。总的来说，本书会讲解怎么让程序在渲染过程的每一步中尽可能地利用好 OpenGL ES 和图形硬件。第一步是为 GPU 提供要处理的 3D 数据。

1.2　为图形处理器提供数据

程序会保存 3D 场景数据到硬件随机存取存储器（RAM）中。嵌入式系统的中央处理单元有专门为其分配的 RAM。在图形处理的过程中，GPU 也有专门为其分配的 RAM。使用现代硬件渲染 3D 图形的速度几乎完全取决于不同的内存区域被访问的方式。

OpenGL ES 是一种软件技术。OpenGL ES 部分运行在 CPU 上，部分运行在 GPU 上。OpenGL ES 横跨在两个处理器之间，协调两个内存区域之间的数据交换。图 1-4 中的箭头代表了与 3D 渲染相关的硬件组件之间的数据交换。每个箭头也代表着一个渲染性能的瓶颈。OpenGL ES 通常会高效地协调数据交换，但是程序与 OpenGL ES 的交互方式会明显地增加或者减少所需的数据交换的数量和类型。对于渲染速度，最快的数据交换方式是没有数据交换。

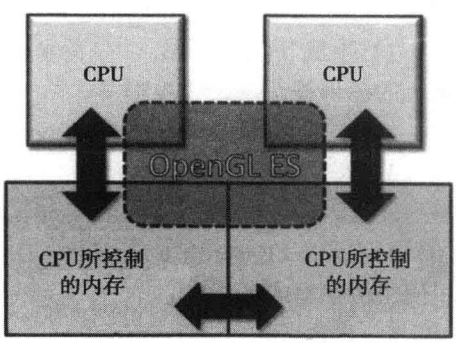


图 1-4　硬件组件和 OpenGL ES 之间的关系

首先，从一个内存区域复制数据到另一