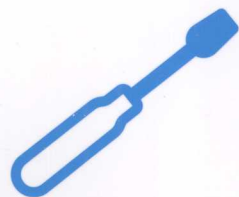


Make: 爱上制作  
makezine.com

格物致知

# 3D图形与人机交互技术应用 Making Things See



边探索边学习

[美] Greg Borenstein 著

张天雷 何雯 译  
郭沐 龚江涛



O'REILLY

人民邮电出版社  
POSTS & TELECOM PRESS

014010560

MAKES

TP391.411

06

格物致知

# 3D图形与人机交互技术应用 Making Things See



[美] Greg Borenstein 著

张天雷 何 雯 译  
郭 沐 龚江涛



北航

C1696977

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权人民邮电出版社出版

人民邮电出版社

北京

TP391.411  
06

## 图书在版编目 (CIP) 数据

格物智知：3D图形与人机交互技术应用 / (美) 博伦斯坦 (Borenstein, G.) 著；张天雷等译. — 北京：人民邮电出版社，2013. 11

(爱上制作)

ISBN 978-7-115-32459-7

I. ①格… II. ①博… ②张… III. ①三维—计算机图形学②人—机系统 IV. ①TP391.41②TB18

中国版本图书馆CIP数据核字(2013)第160862号

## 版权 声 明

Copyright ©2012 by O'Reilly Media, Inc. Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2013. Authorized translation of the English edition, 2013 O'Reilly Media, Inc., the owner of all rights to publish and sell the same. All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2012。简体中文版由人民邮电出版社出版 2013。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者—O'Reilly Media, Inc. 的许可。版权所有，未经书面许可，本书的任何内容不得以任何形式重制。

## 内 容 提 要

本书是关于 3D 图形、3D 打印、人机交互技术的应用宝典。以 Kinect 为主，另涉及 Processing、Arduino、MakerBot 等热门技术。本书由浅入深，从 Kinect 的基本原理讲起，并通过应用实例，对 Kinect 相关知识进行详细地教学，使你制作出来的作品能够对外界进行智能的立体感知。本书还包含了对相关行业的专家和艺术家的采访内容，可以让你进一步了解该领域从业者的工作状态。本书是给创客们以启迪的、创客圈里有口皆碑的经典之作。

- 
- ◆ 著 [美] Greg Borenstein
  - 译 张天雷 何雯 郭沐 龚江涛
  - 责任编辑 周桂红
  - 执行编辑 马涵
  - 责任印制 彭志环 焦志炜
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京精彩雅恒印刷有限公司印刷
  - ◆ 开本：800×1000 1/16  
印张：27  
字数：592 千字 2013 年 11 月第 1 版  
印数：1-3 000 册 2013 年 11 月北京第 1 次印刷

---

著作权合同登记号 图字：01-2012-3280 号

定价：150.00 元

读者服务热线：(010) 67132837 印装质量热线：(010) 67129223

反盗版热线：(010) 67171154

广告经营许可证：京崇工商广字第 0021 号

谨以此书献给 Ellie、Sophie 和 Amalia，未来是你们的。

# 前言

Preface

在微软公司首发 Kinect 的时候，Matt Webb——伦敦 Berg 设计公司的 CEO，敏锐地预见到了 Kinect 会让众多程序员、硬件爱好者们激动不已的可能性。

在微软公司首发 Kinect 的时候，Matt Webb——伦敦 Berg 设计公司的 CEO，敏锐地预见到了 Kinect 会让众多程序员、硬件爱好者们激动不已的可能性。

## “为何 Kinect 如此重要

Kinect 的发布宣告了一场技术革命的开始，如同那些带来了 20 世纪最重要科技突破的革命潮流一样。与个人计算机、互联网的初次亮相相似，Kinect 的发布代表着耗费数十年时间和上百万美元资金的研究成果，将被普及到日常生活中。

面部识别、情报分析、骨骼学、深度图像，这些应用在军用领域的技术，而今可以用在生活中，使人们用它开发更有创造力的项目。为软件构建人体姿态的输入接口，为个人制造业提供低成本的 3D 扫描仪，将动作捕捉用于非常易用的 3D 角色动画建模，用生物学为残障人士制造个性化定制的辅助器械等。

这些应用涉及领域非常广泛，内容各有不同，但是可以被总结为它们使计算机有了“视觉”。在过去的几十年里，我们能够通过计算机处理静态图片和视频，但是简单地在红绿蓝三原色像素上操作，实际上失去了人类视觉系统中我们引以为荣的重要能力——双目视觉，区分空间中的物体，随时间、空间的变化来跟踪人体，识别肢体语言等。而伴随这次成像和图像处理的技术革命，我们这些只有业余时间能做些制作项目的爱好者也能操作上述应用了。

正如此前的个人计算机和互联网革命带来的影响，视觉革命必将引发百花齐放般的创新产品井喷。将 Kinect 的到来同个人计算机和互联网相比较，现在可能听起来有点无法同日而语。但是想想个人计算机第 1 次出现的时候，它也只是一些狂热的“极客”爱好者的玩具。互联网的初衷是一个让政府研究人员可以互相访问计算机的方法。所有这些技术，在当时也都是迎合客观需求出现的，随着人们不断使用它们制作出有创造性的应用，它们最终变成了我们日常生活不可割舍的一部分。几十年以后，也许你会发现，Kinect 可以被比作 Altair 或 ARPA 网，成为名副其实的“新技术革命导火索”。

本书旨在讲解以下技术：

- 使用来自 3D 摄像头的深度图像
- 分析和操作点云
- 跟踪人体关节的运动
- 背景移除和场景分析
- 姿态和手势识别

本书的前 3 章为大家介绍上述技术。你会从中学到如何在 Processing 编程环境中实现这些技术。我们将从最基本的访问 Kinect 数据开始，让你逐步学会构建更复杂程序的能力。学习这些技术并不单单意味着掌握某个特定的软件库或者 API 编程接口，而是要理解技术背后的原理，这样一来，即使技术的细节发生快速演变，你仍然能够运用自如。

不过仅仅掌握这些基本的技术并不足以充分利用这次视觉革命带来的全部优势。为了功能最大化，你还需要理解这个领域中更多的相关知识，它们因成本低廉而且易用的深度数据和骨骼信息而发生了革命性的改变。为此，本书还提供了

3D 扫描、数字化制造、机器人视觉及相关的概念与技术。你可以将这些章理解为教你如何使用到手的深度和骨骼数据。这些概念和技术包括：

- 构建网格
- 准备用于制造的 3D 模型
- 定义和检测手势
- 展示和操控 3D 模型
- 为有动作障碍的人士设计个性化输入设备
- 正向和反向动力学

在这些话题中，我们的焦点将从简单地使用 Kinect，向外扩展到一些相关软件和技术的使用上。本书的最后 3 章，将通过一系列深入的项目来对上述进行探索。我们会写一个用 Kinect 做扫描仪，然后用 3D 打印机来制造真实物体的程序，我们还会创造一个游戏，用来帮助中风病人进行理疗，最后我们将制造一个机器人手臂，它会模仿你手臂的运动。在这些项目过程中，我们以介绍其基本原理开始，然后尝试利用我们所学到的 Kinect 编程的新知识将这些原理付诸实践。我们不会仅使用 Processing 和 Kinect，而是使用所有在构建应用过程中所需要的工具，从 3D 模型构建程序到单片机。

本书在上述领域涉猎广泛并深入，更有着自身独特而迷人的错综复杂，旨在给你提供足够的相关技术，让你能够使用 Kinect 制作有趣的项目，并衷心期待你的所学技术能够激发你的灵感，鼓励你追随先进，探索更深远的知识。

## “ 本书面向读者

从宗旨来讲，本书面向任何想要学习使用 Kinect 创建交互应用的人，无论是想要制作手势接口的交互游戏设计师、使用 3D 扫描仪的创客，还是想要使用计算机视觉技术的艺术家。

期待学到 3D 空间图形和交互技术的初级程序员，还有想要找到学习使用 Kinect 输入、输出捷径以及了解 Kinect 特定使用领域的自身程序员，都会从本书中收获颇丰。

要学习本书中涉及的技术，你无须是图形学专家，或是 Processing 的资深使用者，但还是需要你懂得一些编程知识。

作为开始，我们假设你已经接触过 Processing 这个很有创造力的编程语言（或者你有能力一边深入了解本书内容，一边学习，你可以参阅《爱上 Processing》学习 Processing 的基础知识）。

本书在阅读上的设计由浅入深，从介绍性的章节到复杂的代码和概念的章节，既给出对基础知识的介绍，让你能够制作交互图形应用，同时会教你 Kinect 的相关知识。在开始的时候，我会详细介绍每一个例子的每一个细节，随着章节的

前进，我会不时留出一些细节让你自己思考。最后的目标是让你从一个初学者成为一个信心满满的中级互动图形学程序员。

## “ ” 本书结构

本书的目标是教会你使用 Kinect 构建交互应用的能力。这意味着我们要将你打造成上文介绍过的视觉革命中的先锋。作为视觉革命的先锋，有一系列的好处。一旦你达到了先锋的水平，你就能够用看不见的鼓来发出真实的声音，可以扫描 3D 物体，打印它们的复制品，还可以教会机器人模仿你手臂的动作。

然而，“先锋”这个称号，可不是平白来的。要想步入先锋的行列，你需要学习一系列的基础编程概念和技术。这些技术是先锋这个头衔所带来的所有好处的基础，如果没有这些基础技术，先锋所拥有的那些酷炫的功力都是痴人说梦。本书旨在逐一为你讲授这些基础技术，从最简单、最基础的开始，朝着更复杂更深入进发。我们将从最基本的像素开始，努力朝着错综复杂的 3D 手势进发。

为了完成这个目的，本书的第 1 部分会作为这些基本编程技术的启蒙篇。在我们深入钻研控制机器人或者 3D 打印人脸之前，我们需要学习这些基本编程技术。本书的前 4 章就是介绍使用 Processing 处理 Kinect 数据的相关基础。

Processing 是一个超级有创造力的编程环境，使用 Java 语言，让初学者更加容易地开发出简单的交互应用，包括图形等其他形式的富媒体。正如上文所述，本书假设你具有 Processing 的基础知识（或者类似编程经历），而且随着前 4 章的跟进，我还会给你讲述 Processing 的一些高级概念，尤其是与 Kinect 相关的。这些概念包括在像素数组里循环遍历、基本的 3D 绘制和 3D 朝向控制以及一些简单的几何计算。

我会尝试着把每一个相关概念都解释得透彻深入。目的不仅是想让你最后手里有几个可以作为基础的项目框架，而是让你真正地掌握足够多的基础知识，从而自己创建更好的功能，以及修改我给你在书中所提供的这些功能。在阅读过程中，你可能会觉得我抓住某些细节不放，翻来覆去地解释，却不明就里。请你坚持住，你之后会发现，这些细节会频繁地出现在你自己创造的应用中，而且都是相当重要的环节。

本书这种教学方式还带来一个很好的相关作用。这些所提及的基础技术，与很多领域都有关系，不光是 Kinect。如果你能在本书的教程中通过 Kinect 的使用而掌握它们，那么它们会伴随你到其他 Processing 应用的开发中，在你未来的工作里释放出更多更新的能量，绝对会大幅度地将你从初学阶段推向高层次。

为了构建 Kinect 带来的这些令人振奋的酷炫应用，我们有 3 个基本技术需要学习——深度图像处理、3D 数据的使用和访问骨骼数据。从 3D 扫描到机器人视觉，所有这些应用都使用深度图像来计算物体之间的距离，然后将图像重建成一个 3D 场景，最后跟踪用户身体的每一个独立关节的运动。本书的第 1 部分，



是对这几个技术的介绍。我会着重解释从 Kinect 获得的数据，是如何让这些技术成为可能，也会用编程展示如何对其进行实现，最后教你几个简单的例程，来了解这些技术的好处。

## 玩转深度摄像头

开始的时候，你将学习如何使用 Kinect 提供的深度数据。Kinect 使用一个红外传感器和摄像头制造出其面前的“深度图像”。深度图像与用像素来记录摄像头所捕捉到的颜色和亮度的传统图像不同，它的每一个像素都记录了 Kinect 看到的场景中物体的距离。观察深度图像，你会发现它看起来像不规则的黑白图片的拼接。它们看起来很奇怪，因为图像的颜色并不代表物体明亮的程度，而是代表物体距离 Kinect 的远近。图像中最亮的部分，距离我们最近，而最暗的部分则距离我们最远。如果我们写一个 Processing 程序，来检查深度图像中每一个像素的亮度，我们就能够知道 Kinect 前面每一个物体距离我们的远近了。使用同样的技术，以及一些编程技术，我们甚至可以跟踪最近点的运动，这就可以作为简单交互应用中跟踪用户的很好的一个手段。

## 玩转点云

上面的过程将深度图像看作是一个 2D 空间的数据。它会把 Kinect 所提供的深度信息当作 2D 平面图像来看待，虽然，图片实际上是 3D 的。在第 3 章，我们会学习将这些 2D 像素数据转换成 3D 空间中点集的方法，比如我们可以将像素在图像中的位置作为它的  $x$  和  $y$  轴坐标。也就是说，当我们研究那个距离左上角原点靠右 50，靠下 100 这个像素的时候，它的  $x$  坐标就是 50， $y$  坐标就是 100。而且，别忘了像素是有灰度值的。从我们之前的讨论中得知，每个像素所对应的灰度值，就是它们在 3D 空间中距离我们的远近。这么一来，这个灰度值，就可以代表点的  $z$  轴了。

一旦我们将手里所有的 2D 灰度像素点转换成 3D 空间中的点，我们就生成了称作“点云”的玩意儿。即在 3D 空间中相互不接触的，漂浮在彼此附近的，代表 Kinect 所看到的人和事物的点的集合。你可以将这个点云认作是一个 3D 空间的像素化。从远处看，貌似浑然一体，如果靠近看，图像就会变成点云，中间露出空隙。如果你想要将点云转换成更平滑的连续表面，那就需要找出一种方法，将这些点使用多边形链接起来，填充它们之间的空隙。这个过程叫作“网格化”，我们会在本书后面的章节，物理制造和动画中附加的进行说明。

就目前来讲，哪怕仅针对点云本身，我们都是有很多事情可以做。首先，点云本身非常酷，给你自己及所在的环境做一个实时的 3D 展示，甚至你还可以对其进行操作，从不同的视角进行观察，听起来有点玄幻。这是你在使用 Kinect 的过程中，首次以其本质完全不同于以往传统摄像头的视角，来观察你周围的世界。

为了将这个新的视角发挥到极致，你还需要学习一些在 3D 空间游走和绘画的基础代码。当你开始使用 3D 空间工作时，我会帮你避免一些常见的误区。例如，在 3D 空间游走的过程中，你很有可能迷失当前的方向，你所绘制的图形也会因此不能示人。我会给你讲解 3D 空间的 3 个坐标轴在 Processing 中是如何使用的，还会给你展示一些用于 3D 空间游走和绘画的工具，它们用起来不会让你再困惑。另外一个 3D 绘画中常见的困惑就是摄像机的概念。为了将 3D 空间中的点转换成 2D 图像，以便在计算机显示屏上绘制，Processing 使用了摄像机这一概念。即，当我们把所需要的点都投影到 3D 空间以后，有一个假设的摄像机会被放置在该空间的某个特定地方，把它对准绘制好的点云，然后，简而言之，用摄像机拍照。正如真正的摄像机会将其面前的物体转换成 2D 图像一样，这个虚拟的摄像机也对点云所在的 3D 空间做了相同的转换。摄像机看到的每一样东西，都以摄像机自己的角度和方式，被渲染到计算机的屏幕上。任何摄像机视角之外的东西，都不会被渲染。我将给你展示如何在 3D 空间中移动该摄像机，以不同的角度，而无需真实地移动 Kinect。

## 玩转骨骼数据

从某种角度来讲，第 3 个技术既是最简单的一个，又是最有用途的一个。除了我们一直掌握的原始深度信息之外，在一些软件的帮助下，Kinect 还能够识别人体，告诉我们人体在空间中的位置。特别要指出的是，我们的 Processing 代码能够以 3D 坐标的形式访问到用户身体每一部分的位置——我们能得到手部、头部、肘部、脚步等的确切位置。

深度图像最大的一个优势，在于计算机视觉算法在其上会比传统颜色图像有更好的效果。微软公司研制和发布的用于 Xbox 游戏机的这个深度摄像机，可不仅仅是给玩家展现看起来有点酷的点云，而是因为工程师可以利用深度图像来定位人体并找到对应的关节部位。这个过程被称为“骨骼化”，因为软件会利用深度数据计算推测用户骨架的位置（确切地说是用户各个关节及其之间的位置）。

通过使用正确的 Processing 库，我们可以访问到这个用户骨骼数据，而不用自己着手实现那些极其复杂的骨骼化算法。我们可以通过调用函数来请求任何感兴趣的 3D 关节数据，将其用在我们的交互应用上。在第 4 章，我将给你展示如何通过 Processing 的 Kinect 软件库来访问骨骼数据，以及如何使用它来让我们的应用愈发交互。当然，为了创造出真正有交互感的应用，我们还需要学习一些更复杂的 3D 编程技术。在第 3 章，学习使用点云的时候，我们就学习了基本的 3D 游走和绘画。那么第 4 章我们会通过学习 3D 点比较、点跟踪、点轨迹回放等技术，来将这些基础进一步深化。这些新入手的技术，将作为工程中那些令人振奋的新式接口的基础，使我们能够通过用户姿态、舞姿以及体育运动来进行交互（还有许许多多自然的人体运动）。

一旦我们掌握了 Kinect 使用过程中这 3 个基本技术，我们就可以着手制作

那些酷炫应用了。有了 Kinect，类似 3D 扫描，高级机器人的视觉突然出现在任何一个拥有它的用户面前，只需理解我们上文所描述的基本技术，就可以灵活运用。当然，为了充分利用 Kinect 所提供的可能性，你还需要一些具体应用领域的背景知识。例如，构建模仿人类运动的机器人，仅仅知道如何访问 Kinect 的骨骼数据是不够的，你还需要熟悉逆向动力学，它会告诉你为了让机器人做出特定的姿态，你需要如何摆放机器人的关节。为了创建 3D 扫描，用于制造业或计算机图形学，仅仅知道 Kinect 点云什么的也是不够的，你还需要知道如何利用这些点生成网格，以及如何使用 MakerBot 3D 打印机、CNC 激光切割机，或者其他 3D 打印机来制造。

本书最后 2 章会教你 3D 扫描制造和机器人 3D 视觉的技术。

## 用于数字化制造的 3D 扫描

在第 5 章中，我们的注意力从人体转向物体。我们将 Kinect 作为 3D 扫描仪，以数字格式捕捉真实物体的几何特征，并用 3D 打印机复制。我们会学到如何访问 Kinect 的深度点集，并将它们转换成连续的表面或网格。然后我们将学习如何用标准的文件格式把数据从 Processing 导出，以便在其他软件中使用。我将给你介绍一系列免费的程序，用于预处理这个网格数据，以便后期制造。一旦我们完成了网格数据处理，我们会尝试各种快速原型制作平台对其打印。例如，使用 MakerBot 3D 打印机将其输出成塑料制品，或者提交到 Shapeways 网站，它们可以用一系列从石头到金属等不同的材料打印我们的物体。

## 机器人计算机视觉

在第 6 章中，我们将学习 Kinect 在机器人学上的应用。机器人视觉是有着 50 多年发展历史的深度学科。它的成果包括月球行走机器人、汽车装配流水线等。在本章中，我们将制作一个简单的机器人手臂，它可以复现 Kinect 检测的用户手臂的姿态。我们将 Processing 获取的关节数据，通过串口连接发送给机器人。机器人控制器是一个 Arduino 单片机。Arduino 是 Processing 电子学方面的强力搭档。它使得交互电子制作变得非常容易，正如 Processing 让图形化交互应用更容易。Arduino 平台会侦听从 Processing 发来的命令，并控制机器人的电动机来执行。

我们将使用两种不同的方式来实现这个项目。首先我们会计算 Kinect 检测出的你的关节之间的夹角。这个过程被称为正向动力学，在机器人学中，这是一种由已知角度来控制机器人最终关节位置的方法。然后我们实现机器人端的程序，使得它能够跟随模仿你关节的运动。这里我们用到了逆向动力学。意思是我们不需要确切地知道机器人应该如何运动，我们仅知道它最终的姿态是什么。那么我们就教会机器人自己计算需要达到这个姿态之前的每一个动作、每一个角度变

化。比起前面的正向动力学，这就复杂多了。它的解决方案极其严谨，包含了大量的复数计算和令人迷惑的代码。当然，我们要写的代码比起专业程序要简单很多。但是在此过程中，你也将体验一些在未来复杂的机器人学应用中可能会碰到的问题。

上述章节中，将会对相关领域的知识做一些背景介绍，让你能够从容使用 Kinect 设备，构建自己的想法。

和前 4 章不同，之前我们是想要给你不断灌输和加深相关的基础知识，而这后面 3 章则是想要介绍 Kinect 广泛和多样的应用性。在这 3 个章节中，不再是缓慢而透彻地讲解原理，而是由独立项目构成。每章都从相关领域中选择一个想法，从头到尾实现一遍。在对这些项目的学习过程中，你将不止一次发现我们不仅是在写 Processing 代码。我们需要采访职业医师，帮助使用辅助医疗技术的病人，清理 3D 网格，使用 3D 动画程序，画电路，给 Arduino 单片机编程。在这个过程中，你将体验到很多新奇的想法和工具的用法，但是无一不是前 4 章那样事无巨细了。我们很快掠过这些基础知识，直接讲解制作方法，这个过程特别令人激动，你都无法相信你的双手可以制作出如此杰出的作品。

在这些项目构建中的每一个步骤中，都会用到本书前 4 章所涉及的知识。所以，在学习这些基础知识的时候，千万要全神贯注，它们可是本书所有项目的基石，而且对这些知识的牢靠掌握，也会对你在未来项目中的好想法起到促进作用。

然后，在本书的结尾部分，我们还会更加拓宽视野。学习了这么多 3D 编程技术，也深入理解了 Kinect 知识，我会给你指出在未来的下一步你还可以前进的方向。我们将讨论除了 Processing 以外的其他编程语言和环境，用以下一步的 Kinect 开发。例如 C++ 语言中的交互程序库，还有 Max/MSP 这样的图形化环境，以及 Quartz 软件。微软公司也提供了自己的一套开发工具，把 Kinect 和 Windows 系统深度整合在了一起。我会给你逐一讲解这些工具的优点，告诉你为什么以后有必要试一试。另外，我还会给你提供一些上述相关领域的入门级别资料。

除了对其他编程环境的探讨，你还可以继续学习 Kinect 3D 绘画。对于高级用户，Processing 的 3D 绘画代码是基于 OpenGL 函数库实现的，这个函数库称得上是计算机图形学的标准。OpenGL 函数库很庞大、复杂，也很强大，而 Processing 只给你展现了九牛一毛而已。多学习一些 OpenGL 函数，有助于你的 Kinect 应用进入更高境界。我会给你提供一些 Processing 内部和外部的图形学教程，让你能够写出美妙而吸引人的 3D 图形。

## “ 声明

所有那些只署了一个作者的书，背后都有着广泛的合作力量。同样地，我也不例外。撰写本书所需的知识和技巧，都是我在美国纽约大学交互电信项目组的

老师和同事们耐心协作的产物。如果没有他们的帮助，本书根本无法成文。

Dan Shiffman 的热情激发了我对 Kinect 最初的兴趣。作为教授、本书的技术编辑和朋友，他给予我很大帮助。同时，他鼓舞人心的能力也是我不断追求和渴望的。

Kyle McDonald 和 Zach Lieberman 在 2011 年春天讲授了一门为期 7 周的短期课程，这彻底改变了我的人生。该课程中介绍了很多技术和概念，在本书中都有所提及。我谨希望本书中对于这些知识的阐述能够达到他们课程的一半那么清楚就知足了。进一步讲，Zach 给出了艺术家采访那个点子成为本书中我最喜欢的部分。在帮助我翻译 3D 扫描制造的材料中，Kyle 的价值是不可估量的，这些材料最终成为第 5 章的核心。

Dan O’Sullivan——ITP 主席、Red Burns 创立者和拥护者，给我提供了本书内容所需的环境和支持，还给我建立起了一定能够完成目标的信心。

Lily Szajnberg 是我的第 1 个学生，也是一位理想的读者。本书中那些出色的解释都源自她对知识的渴望，以及指出我那些莫名其妙片段的坦诚。

我还想要感谢 O’Reilly 公司的 Andrew Odewahn 和 Brian Jepsen。Andrew 是相信我能够完成本书的第 1 人，甚至早于我自己。他早些时候给我的反馈使得这个项目从一篇非常长的博客变成了这本书。Brian 坚持不懈地工作，让本书从多种我始料未及的方式有了很大的提高。

Max Rheiner 创建了贯穿本书始终的 SimpleOpenNI 程序库，他作为本书的技术编辑，帮助我确保很多细节的准确性。如果没有他的工作，本书的撰写会很难、很糟糕。

我还想要感谢愿意接受我采访的艺术家们：Robert Hodgins、Elliot Woods、blablalLAB、Nicolas Burrus、Oliver Kreylos、Alejandro Crawford、Kyle McDonald、Josh Blake，还有来自 Adafruit 电子公司的 Phil Torrone 和 Limor Fried。你们的帮助和对作品的期待，是我撰写本书的一大动力。

这里还要大力感谢提供 MakerBot 3D 打印机的 Liz Arum 和 Matt Griffin，以及 Catarina Mota，他们帮助我加快了打印好作品的速度，还有来自 Shapeways 网站的 Duann Scott，他确保了作品的及时出品。

最后，我想对忍受我写作的我的家人、朋友和 ITP 的人们说一声：“我爱你们。”

## “ 代码示例的使用

你可以在程序和文档中使用本书的代码，无需联系我们取得授权，除非你要连续使用本书的大部分代码并用于商业用途。例如，用本书中的一些代码片段写一个程序，这完全无需通知我们。但是出售 O’Reilly 公司的图书包含的 CD-

ROM 就需要我们的许可了。引用本书的代码和示例来回答问题无需许可，但是要将代码放在其他商业化或者产品的文档中就需要许可了。

我们鼓励你标注代码的出处，包括书名、作者、出版社和 ISBN。

如果你觉得在上述许可范围之外使用了我们的代码，请联系 [permissions@oreilly.com](mailto:permissions@oreilly.com)。

## 001

## 第 1 章 什么是 Kinect

- 002 Kinect 的来历和工作原理
- 009 Kinect 艺术家

## 045

## 第 2 章 处理深度图像

- 046 图像和像素
- 047 项目 1: 安装 SimpleOpenNI 编程程序
- 053 项目 2: 编写你的第 1 个 Kinect 程序
- 061 项目 3: 像素
- 066 将像素换算为实际距离
- 067 项目 4: 无线录音测量
- 074 项目 5: 追踪最近的物体
- 083 项目 6: 隐形铅笔
- 092 项目 7: 《极少数派》风格的照片
- 104 练习

## 105

## 第 3 章 利用点云编程

- 106 本章中你将学到
- 107 欢迎来到 3D 世界
- 110 绘制第 1 个点云
- 115 让点云移动
- 121 以彩色的形式呈现点云
- 124 让点云具有交互性
- 134 项目
- 135 项目 8: 空气架子鼓
- 155 项目 9: 虚拟的 Kinect
- 180 总结

## 181

## 第 4 章 玩转骨骼数据

- 188 关于标定的说明
- 189 标定过程的步骤
- 190 用户检测
- 197 访问关节位置
- 204 骨骼结构课程
- 213 测量两个关节之间的距离
- 223 3D 方向转换
- 242 背景去除、用户像素与场景图
- 251 免标定的追踪: 手跟踪及质心
- 260 项目
- 261 项目 10: 训练度量
- 276 练习
- 276 项目 11: “Stayin’ Alive”: 舞步触发 MP3
- 298 练习
- 299 总结

# 301

## 第 5 章 制作扫描

- 306 Modelbuilder 介绍
- 313 MeshLab 介绍
- 316 为 Kinect 数据构建网格
- 322 我们的第 1 次扫描
- 323 清理网格
- 331 校正后的模型
- 332 准备打印
- 333 用 MeshLab 减少多边形
- 335 使用 MakerBot 打印模型
- 338 把模型发送至 Shapeways
- 340 总结

# 343

## 第 6 章 将 Kinect 用于机器人

- 345 正向运动学
- 363 逆向运动学
- 374 总结

# 375

## 第 7 章 结论：下一步要做什么

- 376 超越 Processing：其他框架与语言
- 379 需探索的 3D 编程中的话题
- 384 项目

# 388

## 附录

- 390 SimpleOpenNI 备忘录
- 393 第 2 章
- 397 第 4 章
- 408 第 6 章



# 01

Chapter 第 1 章

## 什么是 Kinect

我们已经讲了一些有关类似 Kinect 那样的深度摄像机的神奇功能了。不过，Kinect 实际上是如何工作的呢？它采集的是什么样的影像？又是为什么这样采集呢？它是如何收集每个场景的深度数据呢？这样一个光滑的小黑盒子中究竟隐藏了什么样的秘密呢？

---

### 本章中你将学到：

*Kinect 是如何工作的*

*Kinect 的来历*

*Kinect 艺术家*

---