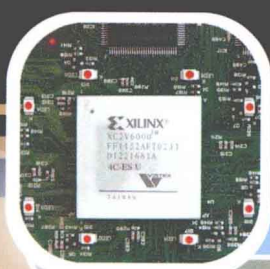


# 基于FPGA的 嵌入式图像处理系统设计

## Design for Embedded Image Processing on FPGAs



◎ Donald G. Bailey 著  
◎ 原魁 何文浩 肖晗 译



電子工業出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY <http://www.phei.com.cn>

# 基于 FPGA 的嵌入式图 像处理系统设计

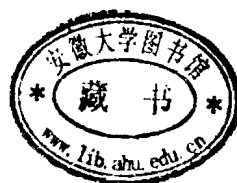
Design for Embedded Image Processing on FPGAs

Donald G. Bailey 著  
原 魁 何文浩 肖 晗 译

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING



## 内 容 简 介

本书详细介绍 FPGA (Field Programmable Gate Array, 现场可编程门阵列) 这种新型可编程电子器件的特点, 对 FPGA 的各种编程语言的发展历程进行回顾, 并针对嵌入式图像处理系统的特点和应用背景, 详细介绍如何利用 FPGA 的硬件并行性特点研制开发高性能嵌入式图像处理系统。作者还结合自己的经验, 介绍了研制开发基于 FPGA 的嵌入式图像处理系统所需要的正确思路以及许多实用性技巧, 并给出了许多图像处理算法在 FPGA 上的具体实现方法以及多个基于 FPGA 实现嵌入式图像处理系统的应用实例。

本书对 FPGA 技术的初学者以及已经具有比较丰富的设计经验的读者来说都有很好的参考价值, 也为从事基于 FPGA 的嵌入式系统开发和应用的软、硬件工程师和科研人员提供一种比较系统、全面的学习材料。

Design for Embedded Image Processing on FPGAs, ISBN 978-0-470-82849-6

Donald G. Bailey

© 2011 John Wiley & Sons (Asia) Pte Ltd

All Rights Reserved. This translation published under license. No part of this book may be reproduced in any form without the written permission of the original copyrights holder.

本书简体中文版专有翻译版权由 John Wiley & Sons, Inc. 授予电子工业出版社。未经许可, 不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字: 01-2011-8009

### 图书在版编目 (CIP) 数据

基于 FPGA 的嵌入式图像处理系统设计/贝利 (Bailey, D.G.) 著; 原魁, 何文浩, 肖晗译. —北京: 电子工业出版社, 2013.2

书名原文: Design for embedded image processing on FPGAs

ISBN 978-7-121-19597-6

I. ①基… II. ①贝… ②原… ③何… ④肖… III. ①数字图象处理—系统设计 IV. ①TN911.73

中国版本图书馆 CIP 数据核字 (2013) 第 029082 号

策划编辑: 曲 昕 (quxin@phei.com.cn)

责任编辑: 王春宁

印 刷: 北京中新伟业印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 27.5 字数: 700 千字

印 次: 2013 年 2 月第 1 次印刷

定 价: 79.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

# 译者序

---

基于 FPGA 的嵌入式系统具有体积小、功耗低、速度快等许多优点，在图像处理以及许多需要高速计算的嵌入式系统（例如智能相机、智能手机等）中有着广阔的应用前景，也是高性能嵌入式系统未来的技术发展趋势。

目前国内外关于 FPGA 应用系统开发的图书很少，各种技术资料也比较分散。因此，虽然许多科研人员和工程技术人员已经充分认识到了基于 FPGA 的嵌入式系统的重要性，但却因为缺乏比较系统完整的学习资料等原因而难以深入了解和掌握相关技术，也难以研制开发出具有实用价值的 FPGA 应用系统。

本书对 FPGA 的发展历程以及如何利用 FPGA 技术设计开发嵌入式视觉系统进行了比较详尽的介绍，不但介绍了相关方法，而且给出了许多具体算法和应用实例。本书对 FPGA 技术的初学者以及已经具有比较丰富的设计经验的读者来说都有很好的参考价值，也将为从事基于 FPGA 的嵌入式系统开发和应用的软、硬件工程师和科研人员提供一本比较系统、全面的学习材料。

本书的专业性很强，对于普通读者来说，理解书中给出的所有知识点可能存在一定的困难。作者在本书的前言中已经相当详细地介绍了他本人写作本书的出发点，以及面向的主要读者群。建议读者先认真读一下作者所写的前言，然后再根据自己的需要开始本书的学习。

对 FPGA 技术应用感兴趣的初学者可以先读一下本书的前 4 章和第 14 章，对 FPGA 及其应用建立一些感性认识，了解一下 FPGA 的基本特点及其主要用途，然后根据自己的需要选择书中的相关章节进行学习；而对于已经具有一定的 FPGA 知识并希望进行 FPGA 系统开发的读者来说，则可以根据自己的需要，直接选择书中的相关章节进行学习，重点了解相关算法的具体实现，以及是否可以根据书中所提供的方法来解决自己所面对或可能面对的问题。

译者所在的课题组近 10 年来一直在从事基于 FPGA 的嵌入式视觉系统的研究，完成了多种基于 FPGA 的图像处理算法，研制开发出两种基于 FPGA 和 DSP 技术的高性能嵌入式图像处理系统并在多种机器人系统中进行了成功应用，可以说已经在一定程度上掌握了基于 FPGA 的嵌入式图像处理系统的相关知识。为了尽可能准确地表达作者的原意，此次译者所在课题组中从事相关研究工作的主要研究人员都参加了本书的翻译工作，并由主要翻译人员原魁、何文浩、肖晗、徐正东、宋海涛、王照、高舒对译稿进行了认真校对。但由于译者的水平有限，仍然难免出现一些错误，因此，译文中如有不当之处，还希望能够得到读者的原谅。

原魁

2012.9

# 前 言

我认为介绍一下本书面世的原因和过程的背景对读者是有帮助的，因为这样有可能使读者能够更好地理解本书内容的构成方式，以及为什么要采用这种方式。

## 背景

首先，介绍一点历史。我自己在图像处理方面有比较广泛的背景，尤其在图像分析、机器视觉和机器人视觉等领域，而上述领域都有很强的应用背景。在超过 25 年的时间里，通过把图像处理技术应用到各种各样的问题中，我在算法开发方面积累了大量经验，不仅在图像处理应用层面，而且在图像处理操作层面。在解决一个应用问题时，我往往更倾向于采用实证方法而不是理论方法——我的着眼点一直是，研制开发可以解决所面对的问题的图像处理算法。这通常会涉及对现有的图像处理操作方法进行组合，但是偶尔也需要开发新的算法和技术来解决问题的某一个方面。通过机器视觉和机器人应用方面的工作，我开始意识到了基于软件的方法的局限性，尤其是计算速度和算法效率方面的局限性。

这就很自然地使我想到了使用 FPGA 作为开发嵌入式图像应用的实现平台。许多图像处理操作本身就是并行的，并且 FPGA 提供的可编程硬件，其本质上也是并行的。因此，将其中一个映射到另一个上应该很简单，对吧？当我开始在 FPGA 上实现图像处理算法时我有很多想法，但是对相关知识了解得却很少。我很快就发现实现一个有效的设计需要很多技巧，因此，在进行了许多实验并经历了许多失败后，我和我的学生很艰难地学到了一些技巧。

通过成为一个电子工程师所接受的基本训练，我很容易就适应了这种硬件的思维模式。但通过观察我的学生，包括本科水平和研究生水平的，我发现这种思维模式可能是完成一个有效设计的最大障碍。图像处理在传统上被认为是软件领域的工作，然而基于 FPGA 的设计却完全是硬件领域的工作。为了在两者之间搭起桥梁，在考虑算法时不仅要考虑算法本身，更要考虑算法的基本计算结构。

因此，在 FPGA 上实现一个图像处理算法（或者实际上任意算法）包括确定法的基本结构，将该结构映射到 FPGA 中的可用资源上，最后将算法映射到硬件结构上等步骤。遗憾的是，目前只有很少的可用材料可以帮助初学者开始该领域的学习，而可以深入了解设计思路及过程的文献资料实际上很难找到，虽然工作在该领域的大多数人都会不自觉地（无论有意还是无意）遵循一定的设计思路。

## 可用文献

虽然发表在会议论文集和期刊上的研究论文已有很多,但是侧重于怎样将图像处理算法映射到FPGA上的并不多。在文献上能找到的研究论文可被分为几个大组。

(1) 第一组论文侧重于研究FPGA本身的结构。该组的大部分论文分析了与逻辑块的结构和粒度、布线网络以及嵌入式内存有关的一系列技术;除了FPGA结构,还涵盖了各种论题,包括基本方法、电源问题、过程可变性和动态可重构性的影响。许多论文仅仅只是建议或与FPGA原型设计有关而与商用芯片无关。虽然从作者自己的角度来看这些论文是令人感兴趣的,而且也给出了非常合适的研究课题,但是,从应用的角度来看,这些论文基本上是不能直接提供解决方案的。虽然这些论文也对一些可能用于设备的下一代产品的特点进行了分析,但大部分论文讨论的主题水平不高。

(2) 第二组论文是对可重构计算进行的研究。这里的侧重点在于怎样使用FPGA能加速计算密集型的任务或者一系列任务。虽然可以将图像处理看作这样的一个任务,但是大部分的研究是针对高性能(并且大功耗)计算而不是低功耗嵌入式系统的。该组的主题包括软、硬件划分,软、硬件的协同设计,动态可重构性,FPGA和CPU之间的通信,FPGA、GPU和CPU性能之间的比较,以及对用于可重构计算应用及研究的操作系统和特定平台的设计。从许多论文中可以得到重要的设计原则和技巧,虽然这可能并不是论文本身的主要着眼点。

(3) 这一组论文和前面一组密切相关,主要考虑FPGA和应用程序编程所用的工具。该组论文侧重于提高开发过程的效率,提出了多种硬件描述语言,而这些语言则以软件开发语言如C、Java甚至Prolog为模型。许多论文的开发工作仅是作为科研工具,因而这些工作很少走出实验室而被用于商业目的。此外,也有大量论文对编译方法进行研究,使其能够将标准软件语言映射到硬件上。这些编译器尝试采用如循环展开、带状开发和流水线等方法产生并行的硬件。同样,许多该类型的论文介绍了有助于实现更有效设计的重要的原则和方法。然而,现有的编译器还不够成熟,难以自动使用各种并行性,它们仅能对所给算法完成相对简单的转换,而不能对算法进行重新设计。

(4) 最后一组论文侧重于各种应用,包括图像处理以及对图像处理操作和系统的实现。令人遗憾的是,由于篇幅限制和空间约束,许多论文只给出了各种系统实现的结果,而很少给出设计细节。这些论文通常描述了最终产品,而并没有说明采用该设计的理由和决策过程。在没有专门的平台和工具,或者对许多丢失的细节进行推断的情况下,并不能重新实现这些设计。虽然许多细节对于专业人士来说是显而易见的,但是,如果没有这方面的知识,仅仅阅读这些论文则完全不能得到同样结论。在该组论文中,比较好的论文会将问题聚焦,并考虑一个单一的图像处理操作的实现。

综上所述,虽然目前已经可以得到大量的相关文献,但这些文献非常分散。在很多情况下,明确知道你要找什么是很有必要的,否则只能靠运气来找。

在开始本领域的研究后不久,我和我的学生列出了一个我们已经知道的研究主题和方法的清单。随着研究工作的进展,我们的清单也在增长。我们最初的想法就是将这些资料编成一本书去帮助其他像我们一样不得不艰苦地自学的人。从本质上讲,这本书反映出了我们在该领域积累的经验,并结合了我们从文献中收集到的(包括FPGA设计和图像处理)方法。

## 面向的读者

这本书主要面向对图像处理的基本操作比较熟悉并且想将其用 FPGA 实现的人群。我们希望读者在刚开始涉足这一领域时能知道本书提到的这些技术和方法。可能从软件思维模式转换到硬件思维模式是最大的障碍。通常来说，我们在进行软件编程时并不需要认真考虑其基本结构，可能这是因为大多数的软件处理器的结构非常相似，因此其差别也仅仅是一个次要因素，无论其对计算机工程师来说多么重要。一个好的编译器能相对有效地将编程语言描述的算法映射到结构上，因此我们没有必要对其考虑太多。但是当编写硬件时，结构则是最重要的问题。这不仅仅是将软件映射到硬件上的问题，还需要对基本的硬件结构进行设计。特别是，对硬件进行编程经常需要将算法转换成一个合适的并行结构，这通常就造成算法本身很大的改变。这不是目前的编译器所能够实现的，因为该工作需要完成大量的设计工作而不是仅仅对数据流进行分解。本书针对这一问题不仅给出了用于图像处理操作的算法，而且给出了有效实现这些算法所使用的基本结构。

本书也将有益于那些熟悉编程并想用 FPGA 解决其他问题的人和那些正考虑图像处理应用的人。虽然本文中许多技术与很多应用领域相关并且是适用的，但是大部分的重点和例子都取自图像处理应用中。本书中给出了构成许多算法及其实现的足够多的细节。但是，我认为对图像处理的学习不仅仅是收集一组算法的过程，还需要有一些数量的优秀的图像处理文本能提供这些算法。成像是一种实用性的学科，动手做去学习最有效，并且在软件环境下学习图像处理比通过 FPGA 学习能提供更大的灵活性和交互性。

这就是说，本书所涉及的是 FPGA 发挥重要作用的嵌入式图像处理领域，一个在实现高效低功耗的最终解决方案时需要硬件工程师和软件工程师将其技术进行紧密结合的技术领域。

## 内容大纲

本书旨在为实现基于 FPGA 的图像处理所用的算法和技术提供一个全面的概述，尤其是针对低级和中级视觉。然而，同任何领域中的设计一样，实现某一特定的任务不是只有一个办法。本书将重点放在了基于流的图像处理方法实现上，因为该方法能有效地利用并行性。这种强调反映了我在该领域的研究背景和经验，但并不意味着是关于该研究方向的最终结论。

第 1 章广泛概述了图像处理操作，包括一段简短的历史背景，解释了许多基本的图像处理的术语，并且用一个例子论证说明了图像处理算法的不同阶段。本章介绍了实时嵌入式图像处理遇到的问题，说明了用传统的串行处理器解决该问题的局限性。高速图像处理必须利用处理图像过程中固有的并行性。本章回顾了并行图像处理系统的一段简短的历史，并给出了使用 FPGA 进行图像处理的缘由。

因为 FPGA 提供可重复编程的硬件，因此它结合了软、硬件系统的优点。第 2 章介绍了 FPGA 技术。虽然其中的一些技术对于实现算法来说并不是必需的，但是了解构建模块和底层架构对于开发一个高效利用资源的解决方案来说是很重要的。在实现图像处理算法部分还回顾了目前可用的 FPGA 的主要特点。

基于 FPGA 的设计是一种硬件设计，因此需要用某些形式的硬件描述语言来表示这些硬

件。第3章回顾了一些主要的语言，同时特别强调了实现算法的设计流程。传统的硬件描述语言如VHDL和Verilog需要对所有的控制进行清楚的编程，因此属于低级编程语言。在过去的15年里，主要是基于C语言的硬件编程的算法研究已经有了很多。文中阐述了其中的一些研究，并且最后简单介绍了几种商业产品。

第4章详细描述了在FPGA上设计和实现一个图像处理应用程序的过程，并特别强调了基于FPGA的实现和基于一个标准软件实现的不同点。初始阶段最重要的一步是要清楚地说明待解决的图像处理的问题是什么。这一步必须足够的详细，提供一个可以评估解决方案的规范说明。本章详述了开发图像处理算法的过程，概述了许多图像处理算法的基本步骤。最终产生的算法必须能够定义系统和计算架构。算法映射过程不是仅仅将算法移植给一种硬件描述语言即可，还需要将算法转换，使得FPGA能充分利用可用资源。实现算法的最后一步是将其映射到计算架构上。

映射过程中的三种约束为：有限的处理时间、受限的数据访问，以及有限的系统资源。第5章介绍了几种可克服或者缓解这些约束的方法，并给出了计算机视觉算法中常见的几种数据结构的可行的FPGA实现。这些有助于在软、硬件实现之间搭起桥梁。图像处理部分介绍了数字表示方法和数字系统，讨论了一系列高效的硬件计算技术。其中的一些技术虽是用硬件实现的，但在高效实现常用函数时可等效于软件库函数。

本书的接下来部分介绍了许多常用图像处理操作的实现，并考虑了一些设计选择和将操作映射到FPGA上时的可选方案。虽然本书介绍得已经相当全面，尤其是在低级图像与图像之间的转换方面，但还是不可能涵盖了每个可能的的设计。文中讨论的例子旨在为其他相关操作提供基础。

第6章介绍了点操作，只考虑输出仅取决于输入图像中相应像素的情况，对直接计算法和查表法都做了介绍。对于多输入图像的情况，详细介绍了如图像平均及背景去除等技术。本章的最后一节将之前的讨论延伸到对彩色图像的处理。重点讨论的特定主题是颜色空间转换、颜色分割和颜色均衡。

第7章讨论了直方图和基于直方图处理的实现，描述了累积一个直方图然后从中抽取数据的一些技术细节，讲述了直方图均衡化、阈值选择以及使用直方图进行图像匹配等具体任务。此外，文中将标准一维直方图的概念扩展到了多维直方图，详细讨论了色彩分割时的聚类和分类的使用。本章结束部分介绍了从多维直方图中提取特征用于纹理分析的过程。

第8章重点介绍了各种局部滤波器，包括线性滤波器和非线性滤波器。本章重点分析了基于流实现的高速缓存技术以及用于高效处理图像边界周围操作的方法，介绍了排序滤波器和一种相关的排序网络架构。形态学滤波器是另一类非常重要的滤波器。形态学滤波器的状态机实现为经典的滤波器实现提供了一种选择方法，对分离性以及串行和并行分解技术的描述使得得到更有效的实现成为可能。

第9章涵盖了图像整形及相关技术。本章对几何变换中的正向和逆向映射方法进行了比较详细的比较，并特别强调了采用流操作的实现方法。插值技术通常与几何变换相关联。本章介绍了用硬件来实现双线性插值、双三次插值以及基于样条的插值算法。本章的末尾还介绍了图像配准的相关技术，包括对尺度不变特征变换和超分辨率的讨论。

第10章介绍了线性变换，将重点放在了快速傅里叶变换、离散余弦变换以及小波变换。



本章介绍了 FFT 和 DCT 的并行实现和流水线实现，并对频率域的滤波和逆滤波进行了比较详细的介绍。为小波变换开发的基于提升的滤波，可将基于直接有限脉冲响应实现所需的逻辑最高减至原来的四分之一。本章最后讨论了图像和视频编码的步骤，并且概述了每个步骤可以用到的一些技术。

第 11 章介绍了与区域检测和标记有关的中间级操作的一种选择方案。对链式编码和连通部分标记所用的标准软件算法进行相应的改造，从而实现了高效的流操作。这能在很大程度上缩短潜伏期并且减少应用程序所需的内存。本章还介绍了距离变换、分水岭变换以及霍夫变换的硬件实现。

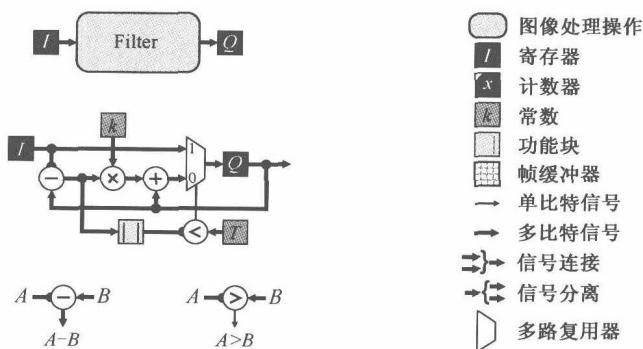
任何一个嵌入式应用程序必须具有与现实世界相连接的接口。第 12 章介绍了一系列的常用外设，并给出了关于它们如何连接到 FPGA 上的一些建议。本章重点介绍了摄像机和视频输出设备接口，同时也介绍了若干种用户接口和存储器设备。本章还重新探讨了图像处理中的隔行扫描技术和拜尔模式的马赛克技术。

接下来的一章对前面提到的与测试和整定技术有关的一些问题进行了扩展。本章论证了可导致实现方案无法按预期的方式运行的四种情况，即设计时的错误、实现时的漏洞、错误的参数选择以及不符合时间约束。几种清单为测试和调试 FPGA 算法提供了指南和一些提示。

本书的最后选择了一些案例，说明怎样把前面章节介绍的材料和技术集成在一个完整的应用中。这些应用简短地说明了设计步骤，并且在整个算法级而并非仅在操作级解释了映射过程。将操作结合在一个相兼容的整体架构中能获益很多。所介绍的这些应用是用于基于手势的用户接口的彩色区域跟踪、校准和纠正镜头的桶形畸变、由人类视觉系统的属性特点联想到开发一种中央窝图像传感器、从一段时间的飞行距离图像系统中提取范围，以及用于实时生产分级的一种机器视觉系统。

## 使用约定

本书内容独立于任何特定的 FPGA 或者 FPGA 供应商，或任何特定的硬件描述语言。该书并不限制读者的范围，其主题就已经足够专业！因此，许多函数和操作使用模块化示意图的形式表示，这就使得用一种独立于语言的表示成为可能，可以把重点放在算法的硬件实现上，并便于移植。这些示意图的基本组成元素如图 P.1 所示。 $I$  通常表示一个图像处理操作的输入， $Q$  通常表示输出。



P.1 本书使用的约定。左上方：一个典型的图像处理操作；左边中间：P.1 所示方程的模块化示意图；左下方：操作数的顺序很重要时操作符的表示形式。右图：模块示意图内各种模块所用符号

对于一些数学运算来说，例如减法运算和比较运算，操作数的顺序是很重要的。鉴于这个原因，第一个操作数用一个点表示而不是箭头表示，如图 P.1 左下方所示。

考虑对流数据进行操作的一个递归滤波器：

$$Q_n = \begin{cases} I_n, & |I_n - Q_{n-1}| < T \\ Q_{n-1} + k(I_n - Q_{n-1}), & \text{其他} \end{cases} \quad (\text{P.1})$$

这里的下标表示流图像的第  $n$  个像素。在较高的级别上，这可以被视为一个图像处理操作，并用一个单独的模块表示，如图 P.1 左上图所示。其左边中间模块给出了上式的低级实现。输入和输出， $I$  和  $Q$ ，用黑框的寄存器表示，而其他可选的寄存器用白框表示。图中下标都省略掉了，因为它们在流操作中都是隐含的。在一些例子中，可能会出现其他的控制输入： $CE$  表示时钟使能， $RST$  表示置位，等等。用中灰色块表示常数，浅灰色表示其他的功能模块。

当用等式表示逻辑函数时， $\vee$  表示逻辑或， $\wedge$  表示逻辑与。这样就可以避免与加法和乘法符号混淆。

## 致 谢

---

我要感谢所有那些帮助并使我对基于 FPGA 设计的理解达到目前水平的人们。我特别想感谢我的研究生们 (David Johnson, Kim Gribbon, Chris Johnston, Aaron Bishell, Andreas Buhler, Ni Ma)。当我们一起努力找出实现图像处理算法的有效方法时, 是他们帮助我形成了对 FPGA 开发的理解和方法。本书反映了多少我的工作, 也就反映了多少他们的工作。

我们用于 FPGA 上的大多数算法是用 Handel-C 编写的, 并且用由 Celoxica 公司提供的板卡进行了测试。我想要对 Roger Gook 和他的团队所提供的支持表示感谢, 起初提供了 Celoxica University Programme, 后来又提供了 Agility Design Solutions。Roger 为开发套件提供了费用大幅度打折的许可, 没有这些, 本书中提到的很多想法就不会得到充分的探讨。

Massey 大学为我提供了一个有利的环境并允许我对这一领域进行自由探索。尤其是 Serge Demidenko, 他给了我开始研究 FPGA 的动力和决心。自那时以来, 他一直给我们启发并提出一些挑战性的问题。其他的同事中尤其值得感谢的是 GourabSen Gupta 和 Richard Harris, 同时我还要感谢 Paul Lyons, 他帮助我一起监督我的学生。

本书一些材料的早期版本作为为时半天的辅导材料, 出现在 2005 年在澳大利亚墨尔本举办的 IEEE 第十区大会 (TenCon)、2007 年在美国德克萨斯州圣安东尼奥举办的 IEEE 图像处理国际会议 (ICIP) 以及 2010 年在新西兰皇后镇举办的亚洲计算机视觉会议 (ACCV) 上。我想感谢这些研讨会的与会者, 感谢他们提供的宝贵的反馈意见以及他们的令人兴奋的讨论。

2008 年期间, 我到英国伦敦帝国学院电路和系统组休假。我非常感谢 Peter Cheung, 他招待了我, 给我提供了一个安静的办公室, 使我免受分心和干扰。我在那里真正开始写这本书, 并至少完成了大部分的框架内容。我要特别感谢 Peter Cheung、Christos Bouganis、Peter Sedcole 和 George Constantinides, 他们一起参与讨论才给了我反复推敲想法的机会。

我的妻子 Robyn 两年来给了我很多晚上和周末的自由时间来完成这篇稿子。我对她的耐心和支持表示感谢。她现在知道了 FPGA 并不是带有射线枪的短吻鳄陷进沼泽地的意思。我将这本书献给她。

Donald Bailey

# 目 录

# C O N T E N T S

## 1

### 图像处理

1.1 基本定义	2
1.2 图像形成	3
1.3 图像处理操作	6
1.4 应用实例	8
1.5 实时图像处理	10
1.6 嵌入式图像处理	11
1.7 串行处理	12
1.8 并行性	14
1.9 硬件图像处理系统	17

## 2

### 现场可编程门阵列

2.1 可编程逻辑器件	19
2.1.1 FPGA 与 ASIC	21
2.2 FPGA 和图像处理	23
2.3 FPGA 的内部	24
2.3.1 逻辑器件	24
2.3.2 互连	26
2.3.3 输入和输出	27
2.3.4 时钟	28
2.3.5 配置	28
2.3.6 功耗	29
2.4 FPGA 产品系列及其特点	31
2.4.1 Xilinx	31
2.4.2 Altera	35
2.4.3 Lattice 半导体公司	41
2.4.4 Achronix	43

2.4.5	SiliconBlue	44
2.4.6	Tabula	44
2.4.7	Actel	45
2.4.8	Atmel	46
2.4.9	QuickLogic	46
2.4.10	MathStar	47
2.4.11	Cypress	47
2.5	选择 FPGA 或开发板	48

### 3

#### 编程语言

3.1	硬件描述语言	53
3.2	基于软件的语言	58
3.2.1	结构化方法	60
3.2.2	扩展语言	61
3.2.3	本地编译技术	66
3.3	Visual 语言	69
3.3.1	行为式描述	69
3.3.2	数据流	70
3.3.3	混合型	71
3.4	小结	73

### 4

#### 设计流程

4.1	问题描述	76
4.2	算法开发	77
4.2.1	算法开发过程	77
4.2.2	算法结构	78
4.2.3	FPGA 开发问题	81
4.3	结构选择	82
4.3.1	系统级结构	82
4.3.2	计算结构	84
4.3.3	硬件和软件的划分	88
4.4	系统实现	90
4.4.1	映射到 FPGA 资源	91
4.4.2	算法映射问题	93
4.4.3	设计流程	95

4.5 为调整和调试进行设计	96
4.5.1 算法调整	96
4.5.2 系统调试	97

## 5

## 映射技术

5.1 时序约束	99
5.1.1 低级流水线	99
5.1.2 处理同步	101
5.1.3 多时钟域	103
5.2 存储器带宽约束	104
5.2.1 存储器架构	105
5.2.2 高速缓存	107
5.2.3 行缓冲	108
5.2.4 其他存储器结构	110
5.3 资源约束	113
5.3.1 资源复用	113
5.3.2 资源控制器	116
5.3.3 重配置性	120
5.4 计算技术	122
5.4.1 数字系统	122
5.4.2 查找表	127
5.4.3 CORDIC	131
5.4.4 近似	137
5.4.5 其他方法	138
5.5 小结	140

## 6

## 点操作

6.1 单幅图像上的点操作	141
6.1.1 对比度和亮度调节	142
6.1.2 全局阈值化和等高线阈值化	145
6.1.3 查找表实现	147
6.2 多幅图像上的点操作	148
6.2.1 图像均值	149
6.2.2 图像相减	150
6.2.3 图像比对	154

6.2.4	亮度缩放	155
6.2.5	图像掩模	156
6.3	彩色图像处理	157
6.3.1	伪彩色	158
6.3.2	色彩空间转换	160
6.3.3	颜色阈值化	172
6.3.4	颜色校正	173
6.3.5	颜色增强	176
6.4	小结	176

## 7

### 直方图操作

7.1	灰度级直方图	177
7.1.1	数据汇集	179
7.1.2	直方图均衡化	183
7.1.3	自动曝光	187
7.1.4	阈值选择	188
7.1.5	直方图相似性	194
7.2	多维直方图	194
7.2.1	三角阵列	195
7.2.2	多维统计信息	196
7.2.3	颜色分割	200
7.2.4	颜色索引	203
7.2.5	纹理分析	204

## 8

### 局部滤波器

8.1	缓存	207
8.2	线性滤波器	211
8.2.1	噪声平滑	212
8.2.2	边缘检测	213
8.2.3	边缘增强	215
8.2.4	线性滤波器技术	215
8.3	非线性滤波器	219
8.3.1	边缘方向	221
8.3.2	非极大值抑制	222
8.3.3	零交点检测	222

8.4	排序滤波器	223
8.4.1	排序滤波器的排序网络	225
8.4.2	自适应直方图均衡化	229
8.5	颜色滤波器	230
8.6	形态学滤波器	232
8.6.1	二值图像的形态学滤波	233
8.6.2	灰度图像形态学	236
8.6.3	颜色形态学滤波	238
8.7	自适应阈值分割	238
8.7.1	误差扩散	239
8.8	小结	240

## 9

## 几何变换

9.1	前向映射	242
9.1.1	可分离映射	243
9.2	逆向映射	248
9.3	插值	250
9.3.1	双线性插值	251
9.3.2	双三次插值	253
9.3.3	样条插值	254
9.3.4	通过插值压缩数据	256
9.4	映射的优化	256
9.5	图像配准	258
9.5.1	基于特征的方法	258
9.5.2	基于区域的方法	262
9.5.3	应用	267

## 10

## 线性变换

10.1	傅里叶变换	270
10.1.1	快速傅里叶变换	271
10.1.2	滤波	277
10.1.3	逆滤波	279
10.1.4	插值	280
10.1.5	配准	281
10.1.6	特征提取	281



10.1.7	哥兹柔算法	282
10.2	离散余弦变换	283
10.3	小波变换	286
10.3.1	滤波实现	288
10.3.2	小波变换的应用	292
10.4	图像和视频编码	293

## 11

### 目标检测和标记

11.1	包围盒	300
11.2	游程编码	302
11.3	链码	303
11.3.1	串行实现算法	304
11.3.2	单次扫描算法	304
11.3.3	特征提取	307
11.4	连通域标注	309
11.4.1	随机读取算法	309
11.4.2	多次扫描算法	309
11.4.3	两次扫描算法	310
11.4.4	单次扫描算法	312
11.4.5	多输入标号	314
11.4.6	进一步优化	315
11.5	距离变换	315
11.5.1	形态学方法	316
11.5.2	倒角距离	316
11.5.3	可分离变换	318
11.5.4	应用	320
11.5.5	测地线距离变换	321
11.6	分水岭变换	321
11.6.1	流水算法	322
11.6.2	浸入算法	323
11.6.3	应用	324
11.7	哈夫变换	325
11.7.1	直线哈夫变换	326
11.7.2	圆哈夫变换	328
11.7.3	广义哈夫变换	329
11.8	小结	329