国外电子与通信教材系列

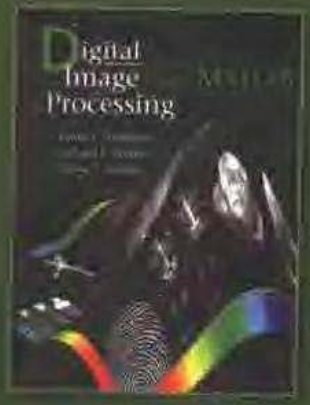
PEARSON
Prentice Hall

冈萨雷斯

数字图像处理

（MATLAB版）

Digital Image Processing Using MATLAB



Rafael C. Gonzalez
[美] Richard E. Woods 著
Steven L. Eddins

阮秋琦 等译

電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

数字图像处理（MATLAB版）

Digital Image Processing Using MATLAB

本书是把图像处理基础理论论述与软件实践方法相结合的第一本书，它集成了冈萨雷斯和伍兹所著的《数字图像处理》一书中的重要内容和MathWorks公司的图像处理工具箱。本书的特色在于它重点强调了怎样通过开发新代码来增强这些软件工具。本书在介绍MATLAB编程基础知识之后，讲述了图像处理的主要内容，具体包括亮度变换、线性和非线性空间滤波、频率域滤波、图像复原与配准、彩色图像处理、小波、图像数据压缩、形态学图像处理、图像分割、区域和边界表示与描述以及对象识别等。

本书主要特点

- 本书自成体系
- 开发了超过60个新的图像处理函数
- 所开发的每个新函数均带有完整的代码清单
- 详细涉及了在MATLAB中使用C代码的方法
- 提供了114个示例、400多幅图像、150多幅图形和线条图
- 书中使用的所有MATLAB函数、图像处理工具箱函数以及新函数，均已在附录中列出
- 详细探讨了图形用户界面（GUI）的设计
- 本书的配套网站提供全面支持（M文件、图像文件、教辅材料、更新等）

本书支持网站：www.imageprocessingplace.com

相关图书

作者简介

Rafael C. Gonzalez：于佛罗里达大学电气工程系获博士学位，田纳西大学电气和计算机工程系教授，田纳西大学图像和模式分析实验室、机器人和计算机视觉实验室的创始人及IEEE会士，国际模式识别、图像处理和机器人领域知名专家与人合著了100多篇技术文章、两本书和4本教材，他的书已被世界500多所大学和研究所采用。

Richard E. Woods：于田纳西大学电气工程系获博士学位，IEEE会员。

译者简介

阮秋琦：北方交通大学博士生导师，计算机与信息技术学院院长，信息科学研究所所长，长期从事信号与信息处理领域的教学和科研工作，先后承担过国家自然科学基金重大项目，国家自然科学基金项目，国家863项目等科研项目40余项。

ISBN 7-121-01456-4
9 787121 014567 >

华信教育资源网 www.hxedu.com.cn

PEARSON Prentice Hall

责任编辑：杜闽燕
责任美编：毛惠庚

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书
ISBN 7-121-01456-4　定价：50.00元

国外电子与通信教材系列

数字图像处理

（MATLAB版）

Digital Image Processing

Using MATLAB

Rafael C. Gonzalez
[美] Richard E. Woods 著
Steven L. Eddins

阮秋琦 等译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内容简介

本书是把图像处理基础理论论述与软件实践方法相结合的第一本书，它集成了冈萨雷斯和伍兹所著的《数字图像处理》一书中的重要内容和MathWorks公司的图像处理工具箱。本书的特色在于它重点强调了怎样通过开发新代码来增强这些软件工具。本书在介绍MATLAB编程基础知识之后，讲述了图像处理的主要内容，具体包括亮度变换、线性和非线性空间滤波、频率域滤波、图像复原与配准、彩色图像处理、小波、图像数据压缩、形态学图像处理、图像分割、区域和边界表示与描述以及对象识别等。

本书概念清晰，层次分明，可供从事信号与信息处理、计算机科学与技术、通信工程、地球物理、医学等专业的大专院校师生学习参考，也可供相应的工程技术人员参考使用。

版权贸易合同登记号　图字：01-2004-0954

图书在版编目（CIP）数据

数字图像处理（MATLAB版）/（美）冈萨雷斯（Gonzalez, R. C.）等著；阮秋琦等译。
北京：电子工业出版社，2005.9
（国外电子与通信教材系列）
书名原文：Digital Image Processing Using MATLAB
ISBN 7-121-01456-4

Ⅰ.数… Ⅱ.①冈… ②阮… Ⅲ.数字图像处理-计算机辅助计算-软件包，MATLAB-教材 Ⅳ.TP391.41

中国版本图书馆CIP数据核字（2005）第067623号

责任编辑：杜闽燕
印　　刷：北京市顺义兴华印刷厂
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
经　　销：各地新华书店
开　　本：787×1092　1/16　　印张：30.5　　字数：781千字　　彩插：2页
印　　次：2005年9月第1次印刷
定　　价：50.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换；若书店售缺，请与本社发行部联系。联系电话：（010）68279077。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。

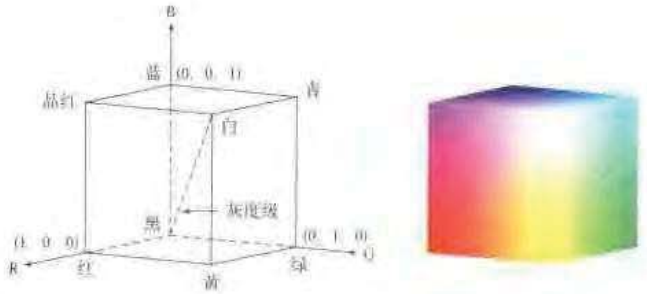


图 6.2

a b

图 6.5

a b

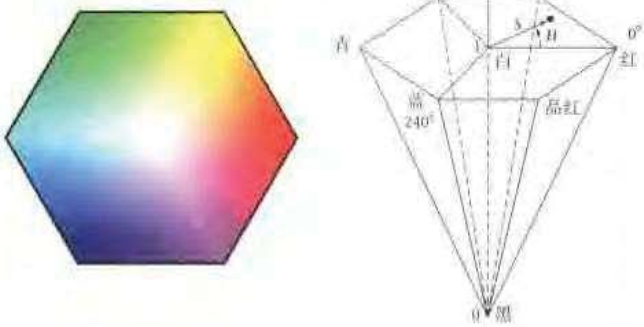


图 6.6

a b

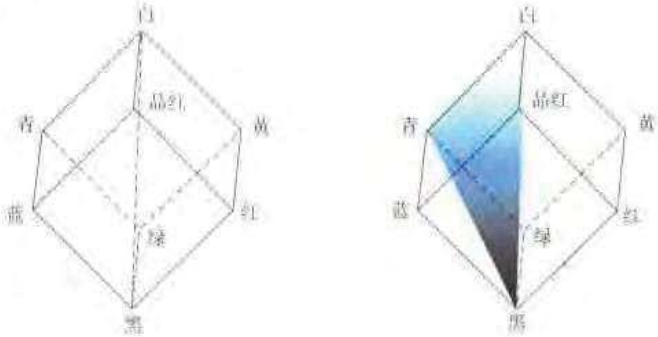


图 6.8

a

b

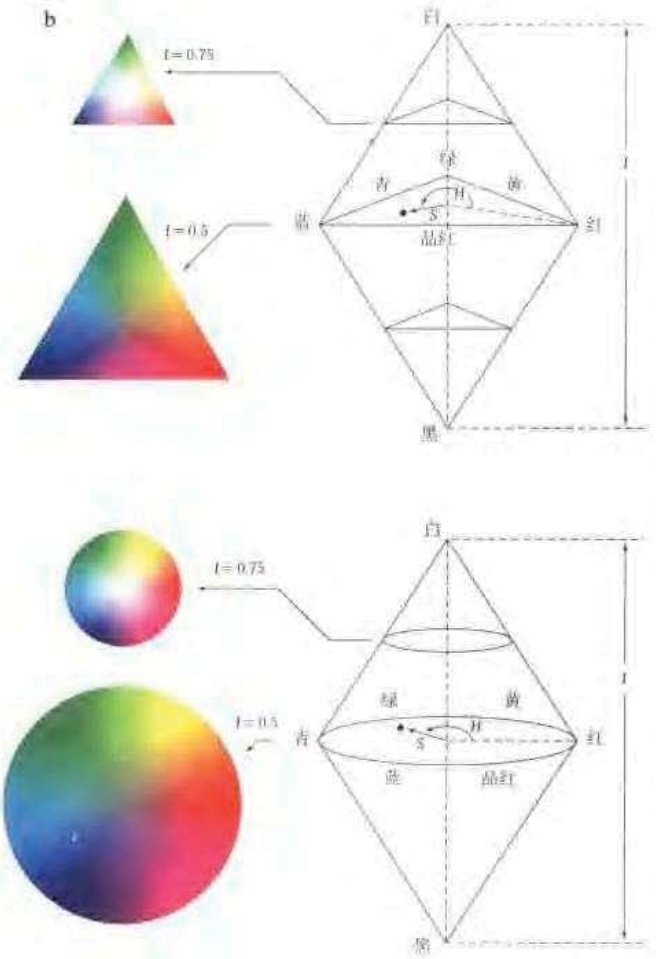
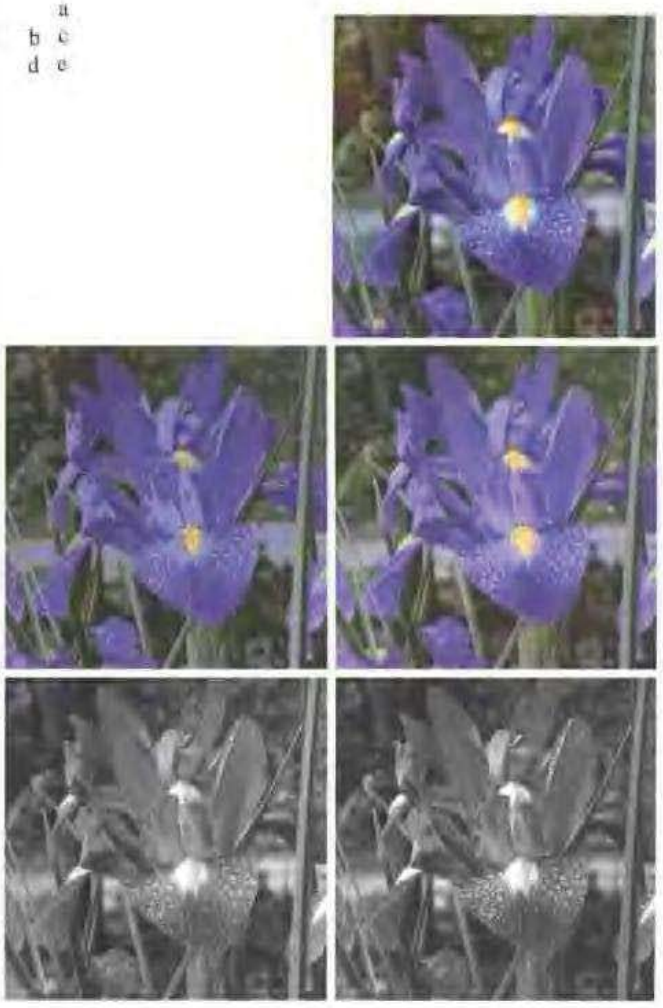


图 6.4

a

b c

d e



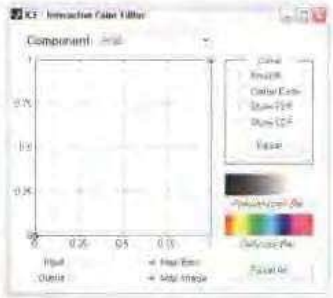


图 6.12

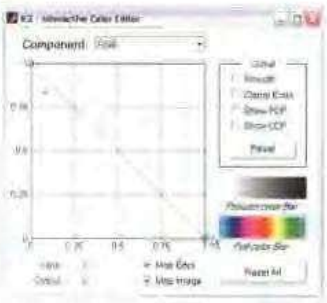
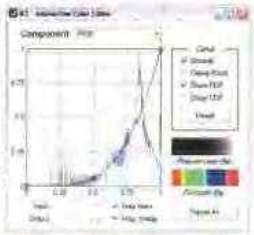


图 6.13

a b

图 6.14

a b



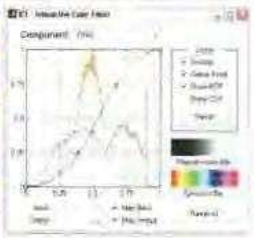
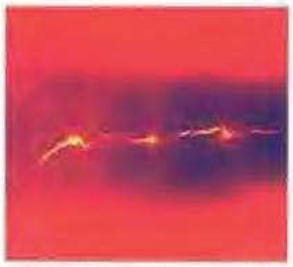
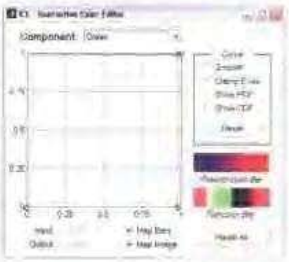


图 6.15

a b c
d e f





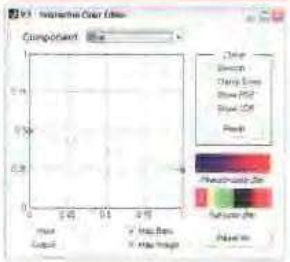


图 6.16

a b
c d

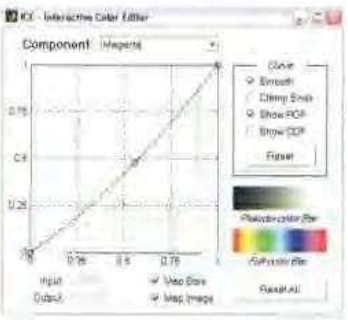


图 6.17

a b c

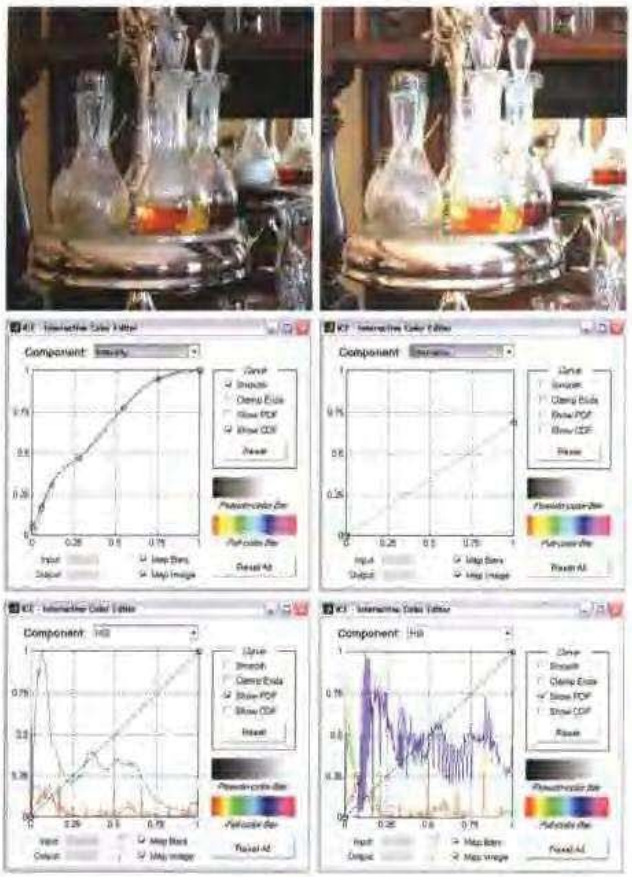


图 6.18

a b
c d
e f

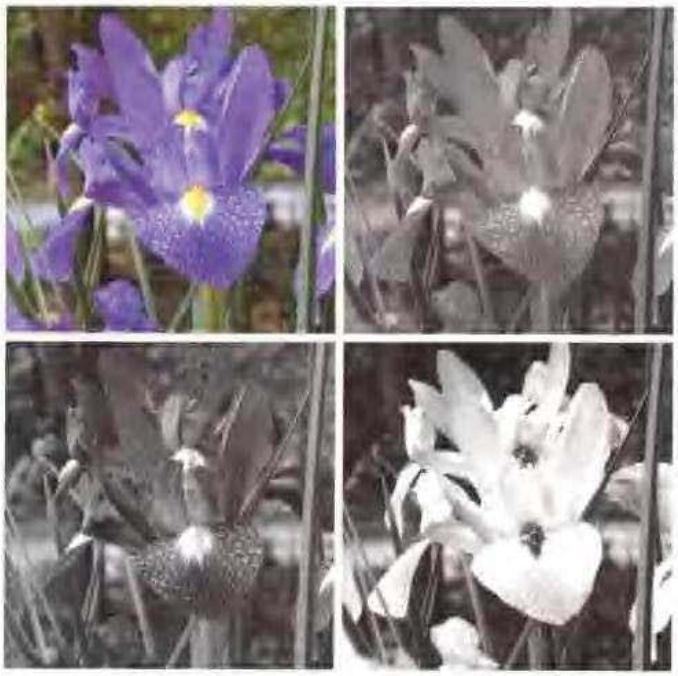


图 6.19

a b
c d

图 6.21

a b c

图 6.22

a b

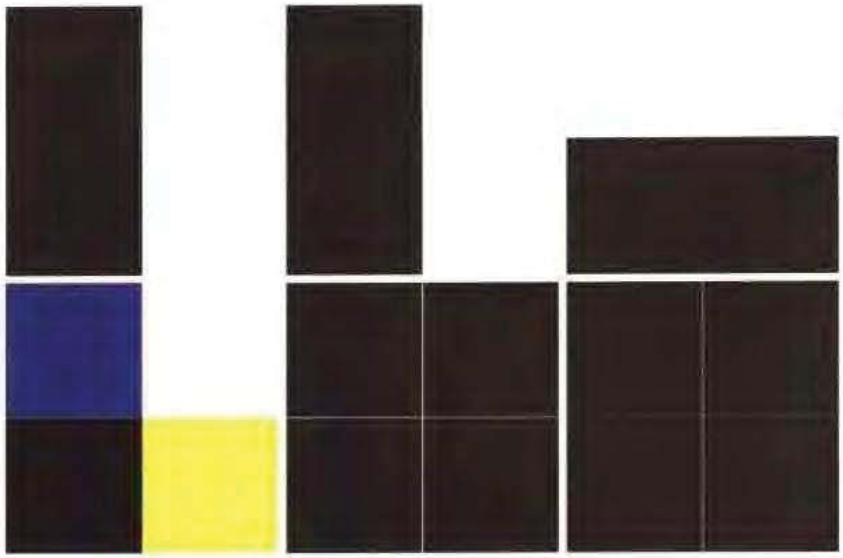


图 6.24

a b c
d e f

图 6.25

a b
c d

图 6.27

a b

序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

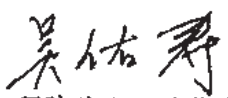
我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个“国外电子与通信教材系列”项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝“国外电子与通信教材系列”项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。



中国工程院院士、清华大学教授

“国外电子与通信教材系列”出版委员会主任

出 版 说 明

进入21世纪以来，我国信息产业在生产和科研方面都大大加快了发展速度，并已成为国民经济发展的支柱产业之一。但是，与世界上其他信息产业发达的国家相比，我国在技术开发、教育培训等方面都还存在着较大的差距。特别是在加入WTO后的今天，我国信息产业面临着国外竞争对手的严峻挑战。

作为我国信息产业的专业科技出版社，我们始终关注着全球电子信息技术的发展方向，始终把引进国外优秀电子与通信信息技术教材和专业书籍放在我们工作的重要位置上。在2000年至2001年间，我社先后从世界著名出版公司引进出版了40余种教材，形成了一套“国外计算机科学教材系列”，在全国高校以及科研部门中受到了欢迎和好评，得到了计算机领域的广大教师与科研工作者的充分肯定。

引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，将有助于我国信息产业培养具有国际竞争能力的技术人才，也将有助于我国国内在电子与通信教学工作中掌握和跟踪国际发展水平。根据国内信息产业的现状、教育部《关于“十五”期间普通高等教育教材建设与改革的意见》的指示精神以及高等院校老师们反映的各种意见，我们决定引进“国外电子与通信教材系列”，并随后开展了大量准备工作。此次引进的国外电子与通信教材均来自国际著名出版商，其中影印教材约占一半。教材内容涉及的学科方向包括电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等，其中既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。我们还将与国外出版商一起，陆续推出一些教材的教学支持资料，为授课教师提供帮助。

此外，“国外电子与通信教材系列”的引进和出版工作得到了教育部高等教育司的大力支持和帮助，其中的部分引进教材已通过“教育部高等学校电子信息科学与工程类专业教学指导委员会”的审核，并得到教育部高等教育司的批准，纳入了“教育部高等教育司推荐——国外优秀信息科学与技术系列教学用书”。

为做好该系列教材的翻译工作，我们聘请了清华大学、北京大学、北京邮电大学、南京邮电大学、东南大学、西安交通大学、天津大学、西安电子科技大学、电子科技大学、中山大学、哈尔滨工业大学、西南交通大学等著名高校的教授和骨干教师参与教材的翻译和审校工作。许多教授在国内电子与通信专业领域享有较高的声望，具有丰富的教学经验，他们的渊博学识从根本上保证了教材的翻译质量和专业学术方面的严格与准确。我们在此对他们的辛勤工作与贡献表示衷心的感谢。此外，对于编辑的选择，我们达到了专业对口；对于从英文原书中发现的错误，我们通过与作者联络、从网上下载勘误表等方式，逐一进行了修订；同时，我们对审校、排版、印制质量进行了严格把关。

今后，我们将进一步加强同各高校教师的密切关系，努力引进更多的国外优秀教材和教学参考书，为我国电子与通信教材达到世界先进水平而努力。由于我们对国内外电子与通信教育的发展仍存在一些认识上的不足，在选题、翻译、出版等方面的工作中还有许多需要改进的地方，恳请广大师生和读者提出批评及建议。

电子工业出版社

教材出版委员会

主　任	吴佑寿	中国工程院院士、清华大学教授
副主任	林金桐	北京邮电大学校长、教授、博士生导师
	杨千里	总参通信部副部长，中国电子学会会士、副理事长 中国通信学会常务理事
委　员	林孝康	清华大学教授、博士生导师、电子工程系副主任、通信与微波研究所所长 教育部电子信息科学与工程类专业教学指导分委员会委员
	徐安士	北京大学教授、博士生导师、电子学系主任 教育部电子信息与电气学科教学指导委员会委员
	樊昌信	西安电子科技大学教授、博士生导师 中国通信学会理事、IEEE 会士
	程时昕	东南大学教授、博士生导师、移动通信国家重点实验室主任
	郁道银	天津大学副校长、教授、博士生导师 教育部电子信息科学与工程类专业教学指导分委员会委员
	阮秋琦	北京交通大学教授、博士生导师 计算机与信息技术学院院长、信息科学研究所所长
	张晓林	北京航空航天大学教授、博士生导师、电子信息工程学院院长 教育部电子信息科学与电气信息类基础课程教学指导分委员会委员
	郑宝玉	南京邮电大学副校长、教授、博士生导师 教育部电子信息与电气学科教学指导委员会委员
	朱世华	西安交通大学副校长、教授、博士生导师、电子与信息工程学院院长 教育部电子信息科学与工程类专业教学指导分委员会委员
	彭启琮	电子科技大学教授、博士生导师、通信与信息工程学院院长 教育部电子信息科学与电气信息类基础课程教学指导分委员会委员
	毛军发	上海交通大学教授、博士生导师、电子信息与电气工程学院副院长 教育部电子信息与电气学科教学指导委员会委员
	赵尔沅	北京邮电大学教授、《中国邮电高校学报（英文版）》编委会主任
	钟允若	原邮电科学研究院副院长、总工程师
	刘　彩	中国通信学会副理事长、秘书长
	杜振民	电子工业出版社原副社长
	王志功	东南大学教授、博士生导师、射频与光电集成电路研究所所长 教育部电子信息科学与电气信息类基础课程教学指导分委员会主任委员
	张中兆	哈尔滨工业大学教授、博士生导师、电子与信息技术研究院院长
	范平志	西南交通大学教授、博士生导师、计算机与通信工程学院院长

译 者 序

数字图像处理起源于20世纪20年代，当时通过海底电缆从英国伦敦到美国纽约采用数字压缩技术传输了第一幅数字照片。此后，由于遥感等领域的应用，使得图像处理技术逐步受到关注并得到了相应的发展。第三代计算机问世后，数字图像处理便开始迅速发展并得到普遍应用。由于CT的发明、应用及获得了备受科技界瞩目的诺贝尔奖，使得数字图像处理技术大放异彩。目前数字图像处理科学已成为工程学、计算机科学、信息科学、统计学、物理、化学、生物学、医学甚至社会科学等领域中各学科之间学习和研究的对象。随着信息高速公路、数字地球概念的提出以及Internet的广泛应用，数字图像处理技术的需求与日俱增。其中，图像信息以其信息量大、传输速度快、作用距离远等一系列的优点成为人类获取信息的重要来源及利用信息的重要手段。因此，图像处理科学与技术逐步向其他学科领域渗透并为其他学科所利用是必然的。图像处理科学又是一门与国计民生紧密相联的应用科学，它已为人类带来了巨大的经济和社会效益，不久的将来不仅在理论上会有更深入的发展，在应用上亦是科学研究、社会生产乃至人类生活中不可缺少的强有力工具。它的发展及应用与我国现代化建设的联系之密切、影响之深远是不可估量的。在信息社会中，数字图像处理科学无论是在理论上还是实践上都存在着巨大的潜力。

本书是冈萨雷斯博士继《数字图像处理》问世后的又一力作。这是图像处理基础理论论述和以MATLAB为主要工具的软件实践方法相对照的一本书，将Gonzalez and Woods[2002]所著的《数字图像处理》的重要理论和MATLAB的图像处理工具有机地结合在一起，为图像处理领域的科技工作者提供了一本通俗实用的参考书。众所周知，MathWorks公司是公认的科学计算方面的引领者，它开发的MATLAB图像处理工具为数字图像处理提供了一个稳定、宽泛的软件实现平台。本书的特色在于它重点强调了怎样通过开发新代码来增强这些软件工具的功能。

本书在介绍MATLAB编程基础知识之后，主要围绕数字图像处理的主干内容展开。这些内容包括：亮度变换、线性和非线性空间滤波、频率域滤波、图像复原与配准、彩色图像处理、小波、图像数据压缩、形态学图像处理、图像分割、区域和边界的表示与描述以及目标识别。全书共分为12章，组织格局与《数字图像处理》一书基本一致，为读者的学习、参考带来了极大的方便。

由于时间仓促，本书的翻译难以达到“信、达、雅”的高标准，退而求其次，尽量做到了译文准确，译文风格统一。此外，为了本书的可实现性，本应对书中的程序进行运行实验，但由于时间关系也没能实现，在此深表歉意。不过大部分语法问题在读者进行实验时都会有相应的提示，加之在本书的网站可以下载所有的可执行文件，因此，不会对读者造成太大的困难。

本书由北京交通大学的阮秋琦教授负责翻译，在翻译过程中得了阮宇智、仵冀颖、江立、陆俊、王萌、张成元、王艳霞等人的帮助，在此深表感谢。由于译者水平所限，书中难免有一些错误及不当之处，恳请读者提出宝贵的建议和批评。

前　言

在数字图像处理领域对问题的求解通常需要宽泛的实验工作，包括软件模拟和大量样本图像的测试。虽然典型算法的开发是基于理论支持的，但这些算法的实现几乎总是要求参数估计，并常常进行算法修正与候选求解方案的比较。这样，灵活的、综合的以及由许多资料证明的软件开发环境就是一个关键因素。这些因素在开销、开发时间和图像处理求解方法上都具有重要意义。

尽管它很重要，但却很少有以教材形式编写的涉及数字图像处理的理论原理和软件实现方面的材料。而本书恰好是为此目的而编写的。它的主要目标是提供一个可用现代软件工具实现图像处理算法的基础。本书自成体系，并且对于具有数字图像处理、数学分析及计算机编程基础知识背景的人来说更易阅读，所有这些内容在技术学科初级或高级课程中都可以找到。同时，也希望读者具备MATLAB的初级知识。

为了达到这一目的，需要两个关键要素。其一是选择图像处理材料，它在该领域中涵盖在正规课程中；其二是选择被充分支持和证明了的软件工具，该工具在现实世界中有着广泛的应用。

为了满足第一个目的，本书后续章节中的大多数理论概念是从冈萨雷斯和伍兹所著的《数字图像处理》一书中选择的，而该书在20多年中被全世界教育工作者选用为引领性的教材。所选择的软件工具来自MATLAB图像处理工具箱，其在教育和工业应用中同样占有优势。编写本书的基本策略是在建立理论概念与用软件工具实现技巧间提供一个无缝的集成。本书沿用《数字图像处理》一书的主线组织。通过这种方法，读者很容易参考这里讨论的数字图像处理的概念，并作为进一步阅读的最新参考。

遵循这种方法可使得我们有可能以简明扼要的方式提供理论材料，从而集中精力解决图像处理问题的软件实现。由于图像处理工作于MATLAB计算环境下，所以图像处理工具箱提供了极大的便利，不仅体现在计算工具的宽泛性上，而且还体现在它支持今天所用的大多数操作系统上。本书的特点是强调如何开发新的代码以便增强已有MATLAB和IPT的功能，这在图像处理中也是一个重要的特性。

介绍MATLAB函数和编程基础之后，本书致力于图像处理的主流领域论述。其涵盖的主要领域包括亮度变换、线性和非线性空间滤波、频域滤波、图像复原和配准、彩色图像处理、小波图像数据压缩、数学形态学图像处理、图像分割、区域和边界的表示和描述及目标识别。这些材料是作为如何用MATLAB和IPT函数来解决图像处理问题的大量论述的补充。在没有所需函数的情况下，编写一个新的函数和文本也是本书所强调的内容。本书后面包含有60多个新函数，这些函数使IPT的范围增加了35%，并且解决了更多新的图像处理问题。

这些以教材形式出现的材料并不能作为软件手册。虽然本书自成体系，但我们还是建成了一个综合网站（见1.5节），该网站被设计用于支持许多领域。对于学生来说，该网站包括背景材料的辅导和综述，以及本书中的所有图像。对于教师来说，网站包含课堂上讲授的材料和书中所用图像、图形的PPT。个别熟悉图像处理和IPT基础的人员将会发现该网站包含有最新参考、最新技术以及在其他地方不容易找到的热点支持材料，读者可适当下载本书开发的所有新函数的可执行文件。

在本书的手稿完成之前，我们一直在努力地修改它。因此，我们在内容的取舍方面已尽了最大努力，我们相应这些内容均是基本的内容，读者在了解这些内容后就可快速地掌握知识。此外，我们相信本书的读者将受益于这种努力，并因此可及时地找到有用的资料。

致谢

我们要感谢学术机构、业界以及政府中的许多人，感谢他们为本书所做的贡献。我们要衷心感谢 Mongi A. Abidi, Peter J. Acklam, Serge Beucher, Ernesto Bribiesca, Michael W. Davidson, Courtney Esposito, Naomi Fernandes, Thomas R. Gest, Roger Heady, Brian Johnson, Lisa Kempler, Roy Lurie, Ashley Mohamed, Joseph E. Pascente, David R. Pickens, Edgardo Felipe Riveron, Michael Robinson, Loren Shure, Jack Sklanski, Sally Stowe, Craig Watson 和 Greg Wolodkin。我们还要感谢本书的图题中所引用的公司，是这些公司允许我们使用了这些图片。

衷心感谢 Prentice Hall 出版公司的 Tom Robbins, Rose Kernam, Alice Dworkin, Xiaohong Zhu, Bruce Kenselaar 和 Jayne Conte，感谢他们为本书所付出的努力。

Rafael C. Gonzalez
Richard E. Woods
Steven L. Eddins

关于作者

Rafael C. Gonzalez（R. C. 冈萨雷斯）

1965 年于迈阿密大学获得电气工程学士学位。1967 年和 1970 年在佛罗里达大学分别获得电气工程硕士和博士学位。自 1970 年起，他一直任教于田纳西大学电气和计算机工程系。1973 年晋升为副教授，1978 年晋升为教授，1984 年被评为杰出贡献教授，1994 年到 1997 年任系主任，现已退休。

冈萨雷斯博士是田纳西大学图像和模式分析实验室、机器人和计算机视觉实验室的创始人。1982年他创建了Perceptics公司，至1992年一直任董事长。1989年，西屋股份有限公司收购了这家公司。在他的指导下，Perceptics公司在图像处理、计算机视觉、光盘存储技术方面获得了极大成功。他还是模式识别、图像处理和机器学习领域企业和政府的常任顾问。他曾获得 1992 年 IEEE 第三区杰出工程师奖等多个获项，并且是 IEEE 会士。

冈萨雷斯博士在模式识别、图像处理和机器人领域编写或与人合著了 100 多篇技术文章、两本书和 5 本教材。他的书已被全世界 500 多所大学和研究所使用。

Richard E. Woods（R. E. 伍兹）

伍兹在田纳西大学获得电气工程学士、硕士和博士学位，做过企业家、科学工作者、政府顾问和管理者。他是 MedData Interactive 公司的创建人，还是 Perceptics 公司的奠基人和副总裁，负责多家公司的定量图像分析和自动判定产品的开发工作。伍兹曾是田纳西大学电气工程和计算机工程系的助理教授，并担任过 Union Carbide 公司的计算机应用工程师。

伍兹博士发表了大量有关数字信号处理方面的文章，并且是 IEEE 等多个专业学会的会员。

Steven L. Eddins（S. L. 艾丁斯）

艾丁斯是 MathWorks 公司图像处理开发组的项目经理。他领导开发了公司多个版本的图像处理工具箱。他的专业兴趣包括构建基于最新研究的图像处理算法的软件工具及其宽泛的科学和工程应用。1993 年加盟 MathWorks 公司之前，艾丁斯博士是芝加哥伊利诺伊大学电气工程和计算机科学系的教师，他为研究生和高年级学生讲授数字图像处理、计算机视觉、模式识别、滤波器设计等课程，并完成了图像压缩方面的研究。艾丁斯是 IEEE 会员。

目　录

第1章　绪言 1
前言 1
1.1　背景知识 1
1.2　什么是数字图像处理 2
1.3　MATLAB 和图像处理工具箱的背景知识 2
1.4　本书涵盖的图像处理范围 3
1.5　本书的 Web 站点 4
1.6　MATLAB 工作环境 4
1.6.1　MATLAB 桌面 5
1.6.2　使用 MATLAB 编辑器创建 M 文件 6
1.6.3　获得帮助 6
1.6.4　保存和检索工作会话 7
1.7　参考文献的组织方式 7
小结 7

第2章　基本原理 8
前言 8
2.1　数字图像的表示 8
2.1.1　坐标约定 8
2.1.2　图像的矩阵表示 9
2.2　读取图像 9
2.3　显示图像 11
2.4　保存图像 12
2.5　数据类 16
2.6　图像类型 17
2.6.1　亮度图像 17
2.6.2　二值图像 17
2.6.3　术语注释 17
2.7　数据类与图像类型间的转换 17
2.7.1　数据类间的转换 18
2.7.2　图像类和类型间的转换 18
2.8　数组索引 21
2.8.1　向量索引 21
2.8.2　矩阵索引 22
2.8.3　选择数组的维数 26
2.9　一些重要的标准数组 26

2.10 M 函数编程简介 27
2.10.1 M 文件 27
2.10.2 运算符 28
2.10.3 流控制 34
2.10.4 代码优化 39
2.10.5 交互式 I/O 42
2.10.6 单元数组与结构简介 44
小结 45

第 3 章 亮度变换与空间滤波 46
前言 46
3.1 背景知识 46
3.2 亮度变换函数 47
3.2.1 函数 imadjust 47
3.2.2 对数和对比度拉伸变换 48
3.2.3 亮度变换的一些实用 M 函数 50
3.3 直方图处理与函数绘图 54
3.3.1 生成并绘制图像的直方图 54
3.3.2 直方图均衡化 58
3.3.3 直方图匹配（规定化） 61
3.4 空间滤波 64
3.4.1 线性空间滤波 65
3.4.2 非线性空间滤波 70
3.5 图像处理工具箱的标准空间滤波器 72
3.5.1 线性空间滤波器 72
3.5.2 非线性空间滤波器 75
小结 77

第 4 章 频域处理 78
前言 78
4.1 二维离散傅里叶变换 78
4.2 在 MATLAB 中计算并可视化二维 DFT 80
4.3 频域滤波 83
4.3.1 基本概念 83
4.3.2 DFT 滤波的基本步骤 87
4.3.3 用于频域滤波的 M 函数 88
4.4 从空间滤波器获得频域滤波器 89
4.5 在频域中直接生成滤波器 92
4.5.1 建立用于实现频域滤波器的网格数组 92
4.5.2 低通频域滤波器 94
4.5.3 线框图与表面图 96
4.6 锐化频域滤波器 99
4.6.1 基本的高通滤波器 99

4.6.2 高频强调滤波 101
小结 102
第5章 图像复原 103
前言 103
5.1 图像退化/复原处理的模型 103
5.2 噪声模型 104
5.2.1 使用函数 imnoise 添加噪声 104
5.2.2 使用指定的分布产生空间随机噪声 105
5.2.3 周期噪声 111
5.2.4 估计噪声参数 113
5.3 仅有噪声的复原：空间滤波 116
5.3.1 空间噪声滤波器 117
5.3.2 自适应空间滤波器 121
5.4 通过频域滤波来降低周期噪声 122
5.5 退化函数建模 123
5.6 直接逆滤波 125
5.7 维纳滤波 126
5.8 约束的最小二乘方（正则）滤波 128
5.9 使用 Lucy-Richardson 算法的迭代非线性复原 130
5.10 盲去卷积 133
5.11 几何变换与图像配准 134
5.11.1 空间几何变换 134
5.11.2 对图像应用空间变换 139
5.11.3 图像配准 141
小结 143
第6章 彩色图像处理 144
前言 144
6.1 MATLAB 中彩色图像的表示方法 144
6.1.1 RGB 图像 144
6.1.2 索引图像 146
6.1.3 用来处理 RGB 图像和索引图像的 IPT 函数 148
6.2 转换至其他彩色空间 151
6.2.1 NTSC 彩色空间 151
6.2.2 YCbCr 彩色空间 152
6.2.3 HSV 彩色空间 152
6.2.4 CMY 和 CMYK 彩色空间 153
6.2.5 HSI 彩色空间 154
6.3 彩色图像处理基础 160
6.4 彩色变换 161
6.5 彩色图像的空间滤波 167
6.5.1 彩色图像平滑 168

6.5.2 彩色图像锐化 171
6.6 在RGB向量空间直接处理 171
6.6.1 使用梯度的彩色边缘检测 172
6.6.2 RGB向量空间中的图像分割 175
小结 178

第7章 小波 179
前言 179
7.1 背景知识 179
7.2 快速小波变换 181
7.2.1 使用小波工具箱的快速小波变换 182
7.2.2 不使用小波工具箱的快速小波变换 186
7.3 小波分解结构的运算 193
7.3.1 不使用小波工具箱编辑小波分解系数 194
7.3.2 显示小波分解系数 198
7.4 快速小波反变换 202
7.5 图像处理中的小波 206
小结 210

第8章 图像压缩 211
前言 211
8.1 背景知识 211
8.2 编码冗余 214
8.2.1 霍夫曼码 216
8.2.2 霍夫曼编码 220
8.2.3 霍夫曼解码 225
8.3 像素间的冗余 232
8.4 心理视觉冗余 236
8.5 JPEG压缩 239
8.5.1 JPEG 239
8.5.2 JPEG 2000 245
小结 251

第9章 形态学图像处理 252
前言 252
9.1 预备知识 252
9.1.1 集合论中的基本概念 252
9.1.2 二值图像、集合和逻辑运算符 254
9.2 膨胀和腐蚀 255
9.2.1 膨胀 255
9.2.2 结构元素的分解 256
9.2.3 函数strel 257
9.2.4 腐蚀 259

9.3 膨胀与腐蚀的组合 261
9.3.1 开运算和闭运算 261
9.3.2 击中或击不中变换 264
9.3.3 使用查找表 266
9.3.4 函数 bwmorph 268
9.4 标注连接分量 270
9.5 形态学重构 273
9.5.1 由重构做开运算 274
9.5.2 填充孔洞 275
9.5.3 清除边界对象 276
9.6 灰度图像形态学 276
9.6.1 膨胀和腐蚀 276
9.6.2 开运算和闭运算 278
9.6.3 重构 282
小结 284

第 10 章 图像分割 285
前言 285
10.1 点、线和边缘检测 285
10.1.1 点检测 286
10.1.2 线检测 287
10.1.3 使用 edge 函数的边缘检测 289
10.2 使用 Hough 变换的线检测 296
10.2.1 使用 Hough 变换做峰值检测 300
10.2.2 使用 Hough 变换做线检测和链接 302
10.3 阈值处理 305
10.3.1 全局阈值处理 305
10.3.2 局部阈值处理 307
10.4 基于区域的分割 307
10.4.1 基础公式 307
10.4.2 区域生长 308
10.4.3 区域分离和合并 311
10.5 使用分水岭变换的分割 315
10.5.1 使用距离变换的分水岭分割 315
10.5.2 使用梯度的分水岭分割 317
10.5.3 控制标记符的分水岭分割 318
小结 320

第 11 章 表示与描述 321
前言 321
11.1 背景知识 321
11.1.1 单元数组与结构 321
11.1.2 本章中使用的其他一些 MATLAB 和 IPT 函数 325

11.1.3 一些基本的M函数 326
11.2 表示 328
11.2.1 链码 328
11.2.2 使用最小周长多边形的多边形近似 330
11.2.3 标记 337
11.2.4 边界片断 340
11.2.5 骨骼 340
11.3 边界描绘子 342
11.3.1 一些简单的描绘子 342
11.3.2 形状数 342
11.3.3 傅里叶描绘子 343
11.3.4 统计矩 347
11.4 区域描绘子 348
11.4.1 函数regionprops 348
11.4.2 纹理 349
11.4.3 不变矩 353
11.5 主分量描述 356
小结 363

第12章 对象识别 364
前言 364
12.1 背景知识 364
12.2 在MATLAB中计算距离度量 365
12.3 基于决策理论方法的识别 367
12.3.1 形成模式向量 367
12.3.2 使用最小距离分类器的模式匹配 367
12.3.3 相关匹配 368
12.3.4 最优统计分类器 370
12.3.5 自适应学习系统 374
12.4 结构识别 375
12.4.1 MATLAB中的串操作 375
12.4.2 串的匹配 382
小结 386

附录A 函数汇总 387

附录B ICE和MATLAB图形用户界面 399

附录C M函数 421

参考文献 461

索引 463

1.2 什么是数字图像处理

一幅图像可以定义为一个二维函数 $f(x, y)$，其中 x 和 y 是空间坐标，而 f 在任意一对坐标 (x, y) 处的幅度称为该点处图像的亮度或灰度。当 x, y 和 f 的幅值都是有限的离散值时，称该图像为数字图像。数字图像处理就是用计算机处理数字图像。注意，数字图像是由有限数量的元素组成的，每个元素都有一个特殊的位置和数值。这些元素称为画素或像素。像素是广泛用于定义数字图像元素的术语。第 2 章中将正式地讨论这些定义。

视觉是我们感觉中最高级的，因此，图像在人类感知中起着最重要的作用并不令人奇怪。然而，人类的视觉被限制在电磁波谱的可视波段，而成像机器几乎覆盖了全部电磁波谱，其范围从伽马射线到无线电波。它们还可以在人类不常涉及的图像源所产生的图像上进行处理，包括超声波、电子显微镜和计算机产生的图像。这样，数字图像处理就包含了很宽的应用领域。

图像处理所涉及的领域到底有多广，作者们并无统一的见解。有时，人们将图像处理定义为其输入和输出均是图像的一个学科。但我们认为这存在局限性，并有点人为界定的意思。例如，在这种定义之下，计算图像的平均亮度这种简单任务将不被认为是图像处理操作。另外，存在像计算机视觉这样的领域，其最终目的是用计算机来模仿人类视觉，包括学习和推理，并根据视觉输入采取相应的行动。该领域本身是人工智能的一个分支，其目的是模仿人类智能。人工智能的研究领域从发展的意义上看还处于初始阶段，其进展要比预期的慢得多。图像分析领域（也称为图像理解）介于图像处理和计算机视觉之间。

图像处理和计算机视觉之间并没有明显的界限，但我们可通过考虑三种类型的计算机化处理来加以划分：低级、中级和高级处理。低级处理包括原始操作，如降低噪声的图像预处理、对比度增强和图像锐化。低级处理的特点是其输入与输出均为图像。图像的中级处理涉及诸如分割这样的任务，即把图像分为区域或对象，然后对对象进行描述，以便把它们简化为适合计算机处理的形式，并对单个对象进行分类（识别）。中级处理的特点是，其输入通常是图像，但输出则是从这些图像中提取的属性（如边缘、轮廓以及单个对象的特性）。最后，高级处理通过执行通常与人类视觉相关的感知函数，来对识别的对象进行总体确认。

基于前面的注释，我们可知图像处理和图像分析之间的重叠之处是图像中单个区域或对象的识别。这样，本书中所谓的数字图像处理就包含了其输入和输出都是图像的过程，从图像中提取特性的过程，以及对单个对象进行识别的过程。为说明这些概念，我们现在考虑文本的自动分析这一领域。该领域的图像获取过程，包括获取文本、预处理图像、提取（分割）个别字符、以适合计算机处理的形式描述字符以及识别这些个别字符，就在本书中所谓的数字图像处理范围之内。弄清这些内容后就了解了图像分析和计算机视觉的领域。正像我们所定义的那样，数字图像处理已成功用于许多领域，给人们带来了巨大的社会和经济价值。

1.3 MATLAB 和图像处理工具箱的背景知识

MATLAB 对于技术计算来说是一种高性能的语言。它以易于应用的环境集成了计算、可视化和编程，在该环境下，问题及其解以我们熟悉的数学表示法来表示。典型的应用包括如下方面：

- 数学和计算
- 算法开发
- 数据获取
- 建模、模拟和原型设计
- 数据分析、研究和可视化
- 科学和工程图形
- 应用开发，包括图像用户界面构建

MATLAB是一种交互式系统，其基本数据元素是并不要求确定维数的一个数组。这就允许人们用公式化方法求解许多技术计算问题，特别是涉及矩阵表示的问题。有时，MATLAB可调用使用C或Fortran这类非交互式语言所编写的程序。

MATLAB是matrix laboratory的缩写。MATLAB由LINPACK（Linear System Package）和EISPACK(Eigen System Package)项目开发，最初用于矩阵处理。今天，MATLAB已集成了LAPACK和BLAS库，并成为了矩阵计算的首选软件。

在高等院校中，对于数学、工程和科学理论中的入门课程和高级课程，MATLAB都是标准的计算工具。在工业领域，MATLAB对于研究、开发和分析也是首选的计算工具。MATLAB中补充了许多针对于特定应用的工具箱。图像处理工具箱是一个MATLAB函数（称为M函数或M文件）集，它扩展了MATLAB解决图像处理问题的能力。其他有时用于补充IPT的工具箱是信号处理、神经网络、模糊逻辑和小波工具箱。

MATLAB学生版MATLAB的一个极具特色的版本。学生版可以通过大学书店和MathWorks网站(www.mathworks.com)以较大的折扣购买到。包括图像处理工具箱在内的附加软件的学生版，也可通过类似的方式购买到。

1.4 本书涵盖的图像处理范围

本书中的每一章都包含有相关的MATLAB和IPT材料，以实现我们已讨论过的图像处理方法。当实现某种特殊方法的MATLAB或IPT函数不存在时，我们会开发一个新的函数并对之加以说明。正如前面提到的那样，本书中包括了所有的新函数。剩下的11章则涉及了如下内容。

第2章：基本原理。本章涵盖了MATLAB表示法、索引和编程概念的基础知识。这些内容是后续内容的基础。

第3章：亮度变换和空间滤波。本章详细讨论了使用MATLAB和IPT实现亮度变换函数的方法，并详细讨论与演示了线性和非线性滤波器。

第4章：频域处理。本章讨论了使用IPT函数计算傅里叶变换及其逆变换的方法，可视化傅里叶频谱的方法，以及在频域实现滤波的方法。此外，还讨论了由特定空间滤波器生成频域滤波器的方法。

第5章：图像复原。本章讨论了传统的线性复原方法，如维纳滤波。讨论并说明了非线性方法，如用于盲去卷积的Richardson-Lucy方法和最大似然估计。此外，还涉及了几何校正和配准。

第6章：彩色图像处理。本章讨论了伪彩色和全彩色图像处理，探讨了可用于数字图像处理的彩色模型，而附加彩色模型的实现则扩展了IPT的功能。本章还探讨了边缘检测和区域分割中彩色的应用。

第7章：小波。IPT目前没有任何小波变换。本章开发了与小波工具箱兼容的小波函数集，这些函数可帮助读者理解Gonzalez and Woods所著的书中讨论的所有小波变换概念。

第 8 章：图像压缩。工具箱中没有数据压缩函数。在这一章中，我们开发了一个可用于这一目的的函数集。

第 9 章：形态学图像处理。本章解释并说明了 IPT 中大量用于二值和灰度图像形态学图像处理的函数。

第 10 章：图像分割。本章解释并说明了用于图像分割的 IPT 函数集。此外，还开发了用于 Hough 变换处理和区域生长的新函数。

第 11 章：表示和描述。本章开发了几个用于对象表示和描述（包括链码和多边形表示）的新函数，还开发了几个用于对象描述的函数，如傅里叶描绘子、纹理和矩不变量。这些函数是对 IPT 中的区域特性函数集的补充。

第 12 章：对象识别。本章的重要特点之一是计算欧几里得距离和 Mahalanobis 距离的函数的有效实现。这些函数在模式匹配中扮演着重要角色。本章还包含有在 MATLAB 中如何操作符号串的广泛讨论。串操作和匹配在结构模式识别中很重要。

除前述内容外，本书还包括有三个附录。

附录 A：包含了所有 IPT 和本书中开发的新图像处理函数，还包括了相应的 MATLAB 函数。这些有用的参考资料提供了工具箱和本书中所有函数的概览。

附录 B：包含有在 MATLAB 中如何实现图形用户界面（GUI）的讨论。GUI 是对本书材料的有用补充，因为它们简化了交互式函数的使用。

附录 C：当在某章中解释一个新概念时，在该章的正文中会包含新函数的列表。除此之外，附录 C 中也包含了这些新函数的列表。过长的函数列表也包含在此处。某些函数的列表只在该附录中列出，目的在于不影响正文的连续阅读性。

1.5 本书的 Web 站点

本书特色之一是提供网站支持。网站地址为 www.prenhall.com/gonzalezwoodseddins。该网站在如下领域为本书提供支持：

- 可下载的 M 文件，包括书中的所有 M 文件
- 培训
- 计划
- 授课材料
- 数据库链接，包括本书中的所有图像
- 书的更新
- 出版背景

该网站与 Gonzalez and Woods 所著的《数字图像处理》一书的网站集成在一起：

www.prenhall.com/gonzalezwoods

它为教学和研究提供了额外的支持。

1.6 MATLAB 工作环境

本节简要介绍使用 MATLAB 的一些重要操作。

启动一个菜单，从菜单中可选择除执行命令外的各种选项。当在工作会话中试用各种命令时，这是很有用的特性。

1.6.2 使用MATLAB编辑器创建M文件

MATLAB编辑器既是用于创建M文件的文本编辑器，也是图形MATLAB调试器。编辑器可以自己以一个窗口出现出，或者以桌面上的子窗口出现。M文件用扩展符.m来表示，如pixeldup.m。MATLAB编辑器窗口有许多下拉菜单，用于保存、查看和调试文件。因为MATLAB编辑器可执行某些简单的检查，并且可用彩色区分各种编码元素，因此，在编写和编辑M函数时，应首选使用该文本编辑器。要打开该编辑器，可在命令窗口的提示符处键入edit命令。类似地，在提示符下键入edit filename，会在编辑器窗口打开M文件filename.m，编辑工作准备就绪。正像前面提到的那样，文件必须在当前目录中，或者在搜索路径的目录中。

1.6.3 获得帮助

获得在线帮助的主要方法是应用MATLAB帮助浏览器[①]，要打开帮助浏览器，可在桌面工具条上双击问号符（?），或在命令窗口提示符处键入helpbrowser。帮助浏览器是集成到MATLAB桌面的Web浏览器，它显示超文本标记语言（HTML）文档。帮助浏览器由两个面板组成，即用于寻找信息的帮助导航面板和用于查看信息的显示面板。导航面板上的自我解释标签用于执行搜索。例如，要得到特殊函数的帮助，可选择**Search**标签，并为**Search Type**选择**Function Name**，然后在**Search for**域中键入该函数的名称。在MATLAB会话开始后，最好打开帮助浏览器，以便在编码开发或其他MATLAB任务期间得到帮助。

获得某个函数的帮助的另一种方法是，在提示符处键入doc及该函数名。例如，若键入doc format，则会在帮助浏览器的显示面板中显示format函数的说明。若浏览器未打开，则该命令会打开浏览器。

M函数有两种可以由用户显示的信息类型。第一种信息类型称为H1行，它包含函数名和一行描述。第二种信息类型称为帮助文本块（详细内容将在2.10.1节中讨论）。在提示符处键入help及函数名，就会在命令窗口显示函数的H1行和帮助文本。有时，这种信息可能比帮助浏览器中的信息更新、更多，因为它是直接从相关M函数的文档中提取的。键入lookfor及一个关键字，会显示所有包含该关键字的H1行。在寻找特殊主题但又不知适用函数的名称时，该函数很有用。例如，在提示符处键入lookfor edge，会显示所有包含该关键字的H1行。因为H1行包含函数名，所以就有可能使用其他的帮助方法来查找指定的函数。在提示符处键入lookfor edge -all，会显示所有函数的H1行，而这些函数会在H1行或帮助文本块中包含单词edge。还会检测到包含字符edge的单词。例如，在H1行或帮助文本中包含单词polyedge的函数的H1行，也会显示出来。

除lookfor方法外，在使用前面描述的任何方法显示一个M函数的信息时，“帮助页”是MATLAB的常用术语。这里建议读者熟悉这些获取信息的方法。因为在后续章节中，我们常常只给出MATLAB和IPT函数的语法。这样做的原因有两个，一是本书的篇幅限制，二是防止讨论离题。在这些情况下，我们仅给出执行函数的语法。通过在线搜索方法，读者可以更详细地研究感兴趣的函数。

最后，1.3节中提到的MathWorks网站包含有丰富的帮助材料、他人开发的函数以及其他资源。

---

① 在本书中应用在线这一术语来指在本地计算机系统中可用的信息，而不是互联网上可用的信息。

1.6.4 保存和检索工作会话

在MATLAB中，保存和载入一个完整的工作会话或选取的工作空间变量有几种方法。最简单的方法如下。

为保存一个完整的工作空间，可简单地在工作空间浏览器窗口中的任何空白处右键单击，并在出现的菜单中选择**Save Workspace As**。这会打开一个目录窗口，该窗口允许命名文件及在系统中选择任何文件夹，并在文件夹中保存文件。然后，可简单地点击**Save**按钮。为了从工作空间保存一个所选的变量，可左键单击以选择该变量，然后在突出显示的区域右键单击。然后，在出现的菜单中选择**Save Selection As**。这将再次打开一个窗口，从中可选择一个文件夹来保存该变量。要选择多个变量，可Shift-单击或Control-单击，然后使用保存单个变量的过程。所有文件都以双精度二进制格式保存，扩展名为`.mat`。这些已保存的文件通常称为MAT文件。例如，名为mywork_2003_02_10的会话在保存时，将会出现MAT文件mywork_2003_02_10.mat。类似地，称为`final_image`的已保存图像（在工作空间中它是一个单变量）在保存时，将会以final_image.mat的形式出现。

要载入保存过的工作空间和/或变量，可在工作空间浏览器窗口的工具条上左键单击文件夹图标。这将打开一个窗口，这时可从中选择一个含有感兴趣MAT文件的文件夹。双击选中的MAT文件或选择**Open**按钮，可在工作空间浏览器窗口中恢复文件的内容。

在提示符处键入带有合适文件名及路径信息的`save`和`load`命令，也可以实现前几段描述的相同结果。这种方法虽然不太方便，但也有其优点。这里建议并鼓励读者使用帮助浏览器来更多地了解这两个函数的相关信息。

1.7 参考文献的组织方式

本书的所有参考文献都按作者和日期这种格式列出，如Soille[2003]。本书理论内容的大多数背景材料源于Gonzalez and Woods[2002]。非此情形时，会在讨论的适当位置给出新的参考文献。适用于所有章节的参考文献，如MATLAB手册和其他常用的参考文献，均标识在本书后面的“参考文献”中。

小结

除简单介绍表示法和基本的MATLAB工具外，本章还强调了在求解数字图像处理问题时理解原型环境的重要性。在后续各章中，我们将开始安排理解IPT函数所需的基础内容，并介绍一组贯穿全书的基本编程概念。第3章到第12章的内容跨越了很宽的主题，它们是数字图像处理应用的主流。尽管涉及的主题不同，但这些章节的讨论却遵循相同的基本主题，即演示并说明如何将MATLAB和IPT函数与新代码结合起来，以便求解宽泛的图像处理问题。

第2章 基本原理

前言

正如前一章所述，MATLAB为数字图像处理带来了一套广泛的函数，这些函数处理的是多维数组，而图像（二维数值数组）正是多维数组的一种特例。图像处理工具箱（IPT）是扩展MATLAB数值计算能力的函数集，这些函数与MATLAB语言的简洁表示，使得大量的图像处理操作可以按简洁明了的编码方式进行，从而为求解图像处理问题提供了一个理想的软件原型环境。在这一章中，我们将介绍MATLAB表示法的基本知识，讨论大量的IPT基本属性和函数，并介绍能进一步增强IPT的程序设计概念。因此，本章中的内容是后续大部分章节的内容的基础。

2.1 数字图像的表示

一幅图像可以被定义为一个二维函数$f(x, y)$，其中x和y是空间（平面）坐标，f在任何坐标点(x, y)处的振幅称为图像在该点的亮度。灰度是用来表示黑白图像亮度的一个术语，而彩色图像是由单个二维图像组合形成的。例如，在RGB彩色系统中，一幅彩色图像是由三幅独立的分量图像（红、绿、蓝）组成的。因此，许多为黑白图像处理开发的技术适用于彩色图像处理，方法是分别处理三幅独立的分量图像即可。彩色图像处理将在第6章详细讲解。

图像关于x和y坐标以及振幅连续。要将这样的一幅图像转换成数字形式，就要求数字化坐标和振幅。将坐标值数字化称为取样；将振幅数字化称为量化。因此，当f的x, y分量和振幅都是有限且离散的量时，称该图像为数字图像。

2.1.1 坐标约定

取样和量化的结果是一个实数矩阵。在本书中，我们使用两种主要的方法来表示数字图像。假设对一幅图像$f(x, y)$取样后，得到了一幅有着M行和N列的图像。我们称这幅图像的大小为$M \times N$。坐标(x, y)的值是离散量。为使符号表示清晰和方便，我们为这些离散坐标使用整数值。在很多图像处理书籍中，图像原点定义在$(x, y) = (0, 0)$处。沿图像第一行的下一坐标值为$(x, y) = (0, 1)$。注意，符号$(0, 1)$用来表示沿第一行的第二个取样，而不表示图像在取样时的实际物理坐标值。图2.1(a)显示了这种坐标约定。注意，x的范围是从0到$M-1$的整数，y的范围是从0到$N-1$的整数。

工具箱中用于表示数组的坐标约定与前段所述的坐标约定有两处不同。首先，工具箱使用(r, c)而不是(x, y)来表示行与列，但坐标顺序与前段所述的坐标顺序一致。在这种情况下，坐标元组(a, b)的第一个元素表示行，第二个元素表示列。另一区别是该坐标系统的原点在$(r, c) = (1, 1)$处。因此，r是从1到M的整数，c是从1到N的整数，如图2.1(b)所示。

IPT文档将图2.1(b)中的坐标称为像素坐标。IPT还采用另一种较少使用的坐标约定，称为空间坐标，这种坐标使用x来表示列，使用y来表示行。这与我们所用的变量x与y正好相反。在本书中，除少量例外外，我们将不使用IPT的空间坐标约定，但读者在IPT文档中一定会遇到这种术语。

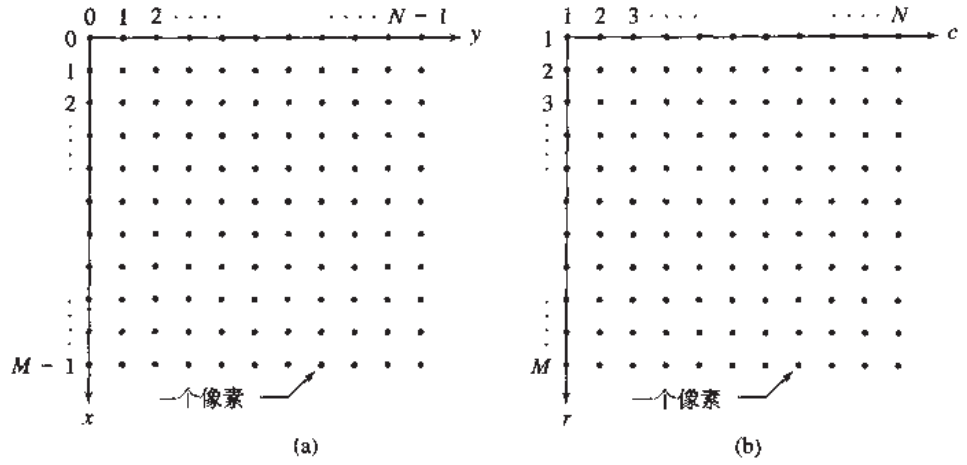


图2.1 所用的坐标约定：(a)多数图像处理书籍中所用的坐标约定；(b)图像处理工具箱中所用的坐标约定

2.1.2 图像的矩阵表示

由图2.1(a)所示的坐标系统和前述讨论，我们可以得到如下数字化图像函数的表示：

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

等式右边是由定义给出的一幅数字图像。该数组的每一个元素都称为像元、图元或像素。图像和像素这两个术语在本书后面的讨论中，将用来表示一幅数字图像及其元素。

一幅数字图像在MATLAB中可以很自然地表示成矩阵

$$\mathtt{f}=\begin{bmatrix} \mathtt{f(1,1)} & \mathtt{f(1,2)} & \cdots & \mathtt{f(1,N)} \\ \mathtt{f(2,1)} & \mathtt{f(2,2)} & \cdots & \mathtt{f(2,N)} \\ \vdots & \vdots & & \vdots \\ \mathtt{f(M,1)} & \mathtt{f(M,2)} & \cdots & \mathtt{f(M,N)} \end{bmatrix}$$

其中f(1,1)=$f(0,0)$，注意，等宽字体用来表示MATLAB的量。很明显，这两种表示是相同的，只是原点不同。符号f(p,q)表示位于p行和q列的元素。例如，f(6,2)是指矩阵f中位于第6行和第2列的元素。一般来说，我们分别用字母M和N来表示矩阵中的行与列。一个1 × N矩阵称为一个行向量，而一个M × 1矩阵称为一个列向量。一个1 × 1矩阵是一个标量。

在MATLAB中，矩阵以变量的形式来存储，名称诸如A,a,RGB,real_array等。变量必须以字母开头，且只能由字母、数字和下划线组成。如在前段中注释的那样，本书中的所有MATLAB量都用等宽字体来表示。此外，我们使用常见的斜体罗马字母来表示数学表达式，如$f(x,y)$。

2.2 读取图像

使用函数imread可以将图像读入MATLAB环境，imread的语法为

```
imread( 'filename' )
```

表 2.1 MATLAB 6.5 中的函数 imread 和 imwrite 所支持的一些常用图像 / 图形格式。早期版本支持这些格式的子集。所支持格式的完整清单请参阅在线帮助

格式名称	描述	可识别扩展符
TIFF	加标识的图像文件格式	.tif,.tiff
JPEG	联合图像专家组	.jpg,.jpeg
GIF	图形交换格式 *	.gif
BMP	Windows 位图	.bmp
PNG	可移植网络图形	.png
XWD	X Window 转储	.xwd

* imread 支持 GIF 格式，但 imwrite 不支持该格式。

其中，filename 是一个含有图像文件全名的字符串（包括任何可用的扩展名）。例如，命令行

```
>> f = imread('chestxray.jpg');
```

将 JPEG 图像（见表 2.1）chestxray 读入图像数组 f。注意，这里使用单引号(')来界定 filename 字符串。命令行结尾处的分号在 MATLAB 中用于取消输出。若命令行中未包含分号，则 MATLAB 会立即显示该行中指的运算的结果。在 MATLAB 命令行窗口中出现的提示符(>>)指明了命令行的开始（如图 1.1 所示）。

就像上面的这个命令行一样，当 filename 中不包含任何路径信息时，imread 会从当前目录中寻找并读取图像文件（见 1.7.1 节）。若当前目录中没有所需要的文件，则它会尝试在 MATLAB 搜索路径中寻找该文件（见 1.7.1 节）。要想读取指定路径中的图像，最简单的办法就是在 filename 中输入完整的或相对的路径。例如，

```
>> f = imread('D: \myimages\chestxray.jpg');
```

从驱动器 D 上名为 myimages 的文件夹中读取图像文件 chestxray.jpg；而

```
>> f = imread('.\myimages\chestxray.jpg');
```

从当前的工作目录中名为 myimages 的子目录中读取图像文件 chestxray.jpg。MATLAB 桌面工具条上的当前目录窗口会显示 MATLAB 的当前工作路径，并提供一种非常简单的方法来手工改变当前的路径。表 2.1 列出了函数 imread 和 imwrite 所支持的常用图像/图形格式（函数 imwrite 将在 2.4 节中讨论）。

函数 size 可给出一幅图像的行数和列数：

```
>>size(f)
ans =
     1024   1024
```

在使用如下格式来自动确定一幅图像的大小时，该函数很有用：

```
>> [ M, N ] = size(f);
```

该语法将返回图像的行数（M）和列数（N）。

函数 whos 可以显示出一个数组的附加信息。例如，语句

```
>> whos f
```

会给出下列结果：

```
  Name         Size                Bytes         Class
  f            1024 x 1024           1048576       uint8 array
  Grand total is 1048576 elements using 1048576 bytes
```

结果中的uint8是指几种MATLAB数据类之一，详见2.5节的讨论。whos行结尾处的分号对结果没有影响，因此我们一般将其省略。

2.3 显示图像

在MATLAB桌面上图像一般使用函数imshow来显示，该函数的基本语法为

```
imshow(f, G)
```

其中，f是一个图像数组，G是显示该图像的灰度级数。若将G省略，则默认的灰度级数是256。语法

```
imshow(f, [low high])
```

会将所有小于或等于low的值都显示为黑色，所有大于或等于high的值都显示为白色。界于low和high之间的值将以默认的级数显示为中等亮度值。最后，语法

```
imshow(f, [ ])
```

可以将变量low设置为数组f的最小值，将变量high设置为数组f的最大值。函数imshow的这一形式在显示一幅动态范围较小的图像或既有正值又有负值的图像时非常有用。

函数pixval经常用来交互地显示单个像素的亮度值。该函数可以显示覆盖在图像上的光标。当光标随着鼠标在图像上移动时，光标所在位置的坐标和该点的亮度值会在该图形窗口的下方显示出来。处理彩色图像时，红、绿、蓝分量的坐标也会显示出来。若按下鼠标左键不放，则pixval将显示光标初始位置和当前位置间的欧几里得距离。

此处应注意的是，语法

```
pixval
```

会在上次显示的图像上显示光标。单击光标窗口上的X按钮可将其关闭。

例2.1 读入和显示图像

(a) 下列语句会从磁盘中读入一幅名为rose_512.tif的图像，提取该图像的基本信息，并使用imshow将其显示出来：

```
>> f = imread('rose_512.tif');
>> whos f
  Name        Size              Bytes    Class
  f           512 x 512         262144     uint8 array
  Grand total is 262144 elements using 262144 bytes
>> imshow(f)
```

由于imshow命令行结尾处的分号对结果无影响，所以一般将其省略。屏幕输出如图2.2所示。窗口左上角显示了图像编号。窗口上有各种下拉菜单和工具按钮，用于缩放、保存及输出显示窗口的内容等。特别地，在将结果打印或存盘之前，**Edit**菜单有对结果进行编辑或格式化的功能。

当用imshow显示另一幅图像g时，MATLAB会在屏幕上用新图像替换旧图像。为保持第一幅图像并同时显示第二幅图像，可以使用如下的figure函数：

```
>> figure, imshow(g)
```

使用语句

```
>> imwrite(f, 'patient10_run1', 'tif' )
```

或

```
>> imwrite(f, 'patient10_run1.tif' )
```

若 filename 中不包含路径信息，则 imwrite 会将文件保存到当前的工作目录中。

函数 imwrite 可以有其他的参数，具体取决于所选的文件格式。后续章节中的大部分工作都是处理 JPEG 或 TIFF 格式的图像，所以我们将注意力放在这两种格式上。

另一种常用但只适用于 JPEG 图像的函数是 imwrite，其语法为

```
imwrite(f, 'filename.jpg ', 'quality', q )
```

其中，q 是一个在 0 到 100 之间的整数（由于 JPEG 压缩，q 越小，图像的退化就越严重）。

例 2.2 使用函数 imfinfo 保存一幅图像

图 2.4(a)所示的图像 f 是由给定的化学过程得到的典型序列图像之一。我们期望定期将这些图像传送到中心站点，以便进行显示或自动检查。为降低存储量并减少传输时间，可在图像的外观质量不低于某个合理的级别时，尽量压缩图像。其中，“合理的”意味着没有可察觉的伪轮廓线。图 2.4(b)到图 2.4(f)示出了分别以 q = 50, 25, 15, 5 和 0 将图像 f 写到磁盘的结果（JPEG 格式）。例如，q = 25 时的语法结构为

```
>> imwrite(f, 'bubbles25.jpg', 'qyality', 25 )
```

以 q = 15 存储的图像［见图 2.4(d)］中几乎看不清伪轮廓线。当 q = 5 和 q = 0 时，伪轮廓线就变得非常明显。因此，消除伪边缘的一种可接受方案就是用参数 q = 25 来压缩图像。要了解所实现的压缩并获得图像文件的其他详细信息，可以使用 imfinfo 函数，其语法结构为

```
imfinfo filename
```

其中，filename 是存储在磁盘中的图像的全名。例如，

```
>> imfinfo bubbles25.jpg
```

会输出如下信息（注意，在这种情况下，有些域不包含信息）：

```
            Filename:   'bubbles25.jpg'
         FileModDate:   '04-Jan-2003 12:31:26'
            FileSize:   13849
              Format:   'jpg'
       FormatVersion:   ''
               Width:   714
              Height:   682
            BitDepth:   8
           ColorType:   'grayscale'
     FormatSignature:   ''
             Comment:   {}
```

其中，FileSize 以字节为单位。通过简单地使用 width 乘以 Height，再乘以 BitDepth，然后将结果除以 8，就可以计算出原图像中的字节数。计算的结果是 486 948。用这个结果除以 FileSize 就可以得到压缩比：(486 948/13 849) = 35.16。这一压缩比是在保持图像质量与应用要求一致的前提下得到的。除了在存储空间方面有明显的优势之外，这种压缩还可使得单位时间内传输的数据量大约是压缩前的 35 倍。

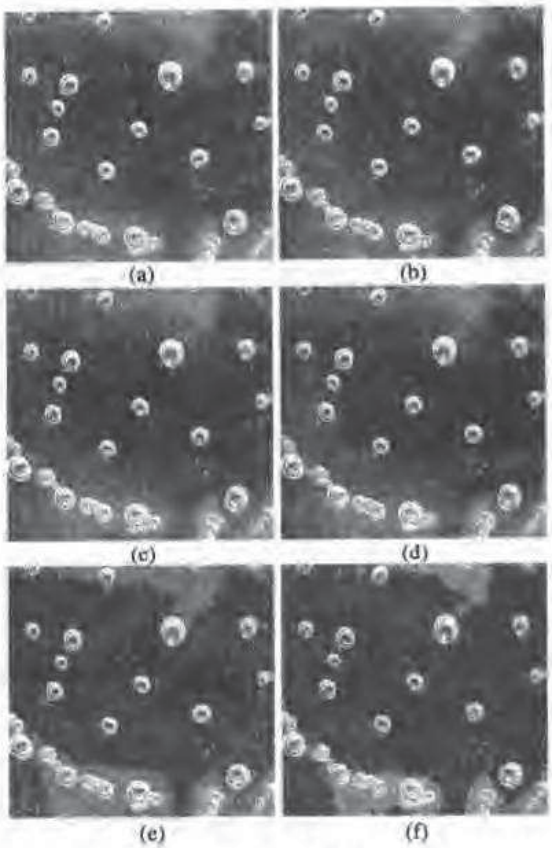


图 2.4 (a)原图像；(b)~(f)分别使用jpg质量参数q=50,25,15,5,0的结果。q=15时，伪轮廓并不明显[见图像(d)]，而在q=5和0时则十分明显

由函数imfinfo显示的信息域可捕获至所谓的结构变量中，以便用于后续的计算。若以前面的这幅图像为例，并将相应的结构变量取名为K，则使用语法

```
>> K = imfinfo('bubbles25.jpg');
```

可将由命令imfinfo产生的所有信息存入变量K。由imfinfo产生的信息附加到了形式为K域的结构变量中。例如，图像的高度和宽度现在存储在结构域K.Height与K.Width中。下面我们通过计算图像bubbles25.jpg的压缩比来考虑使用结构变量K的一个示例：

```
>> K = imfinfo('bubbles25.jpg');
>> image_bytes = K.Width*K.Height*K.BitDepth/8;
>> compressed_bytes = K.FileSize;
>> compression_ratio = image_bytes/compressed_bytes

compression_ratio =
    35.1612
```

注意，imfinfo有两种不同的用法。第一种用法是在命令行键入imfinfo bubbles25.jpg，其结果是在屏幕上显示出信息。第二种用法是键入K=imfinfo('bubbles25.jpg')，其结果是把由imfinfo产生的信息存入K。调用imfinfo的这两种不同方法是命令-函数二元性的一个例子，这个重要的概念在MATLAB的在线文档中有更加详细的解释。

函数imwrite的另一种常用但只适用于tif图像的语法为

```
imwrite(g, 'filename.tif', 'compression', 'parameter', ...
                                'resolution', [colres rowres])
```

其中，'parameter'可以是如下主要的值之一：'none'表示无压缩；'packbits'表示比特包压缩(非二值图像的默认参数)；'ccitt'表示ccitt压缩(二值图像的默认参数)。1×2矩阵[colres rowres]包含两个整数，分别以每单位中的点数给出图像的列分辨率和行分辨率(默认值为[72 72])。例如，若一幅图像的大小以英寸来表示，则colres是垂直方向上每英寸的点(像素)数(dpi)，而rowres是水平方向上每英寸的点数。使用标量res来指定分辨率与使用[res res]是等价的。

例2.3 使用imwrite的参数

图2.5(a)是检查一块电路板的质量时产生的一幅8比特X光图像。其格式为jpg，分辨率为200 dpi。图像大小为450 × 450像素，即其尺寸为2.25 × 2.25英寸。我们想把这幅图像以tif格式存储为无压缩的名为sf的图像，并且想将图像的尺寸减小到1.5 × 1.5英寸，但其像素数仍保持为450 × 450。下列语句可产生我们所希望的结果：

```
>> imwrite(f,'sf.tif','compression','none','resolution', ...
                                                    [300 300])
```

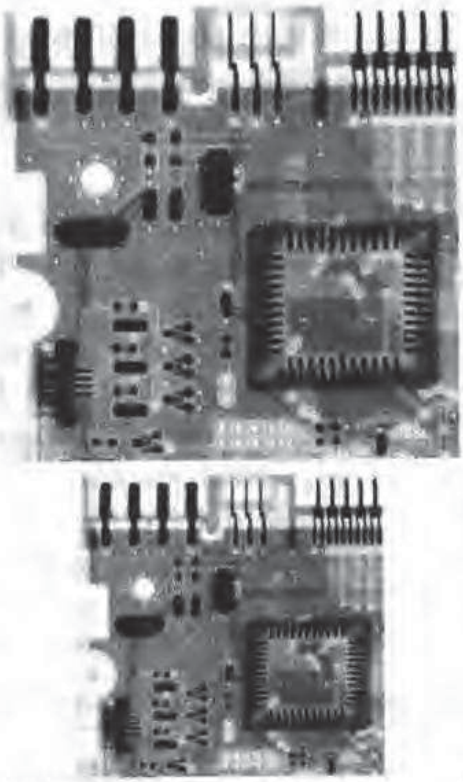


图2.5 更改dpi分辨率并同时保持像素数不变时的效果：(a)分辨率为200 dpi且大小为450 × 450的图像(大小为2.25 × 2.25英寸)；(b)大小为450 × 450(1.5 × 1.5英寸)且分辨率为300 dpi的同一幅图像(原图像由Lixi公司提供)

向量[colres rowres]的值由分辨率200 dpi乘以压缩比2.25/1.5得到，即300 dpi。可以用下列语句来代替手工计算：

```
>> res = round(200*2.25/1.5);
>> imwrite(f, 'sf.tif', 'compression', 'none' ,'resolution', res)
```

其中，函数round会将其参量舍入为最接近的整数。注意，这些命令并不改变像素数，而只改变图像的大小。原图像的像素数为450 × 450，分辨率为200 dpi，尺寸为2.25 × 2.25英寸。

分辨率为 300 dpi 的新图像与原图像完全一致，只是其 450 × 450 像素分布在 1.5 × 1.5 英寸的区域内。这样的过程在打印文档时控制图像的大小而不牺牲其分辨率是非常有用的。

通常，有必要将图像按它们在MATLAB桌面上显示的那样输出到磁盘中。对于下一章将要讲到的绘图，更应如此。图形窗口的内容可以按两种方法输出到磁盘。第一种方法是在图形窗口的**File**下拉菜单（见图2.2）中选择**Export**。使用这一选项，用户可以选择保存路径、文件名以及文件格式。使用`print`命令可以获得更多的输出参数：

```
print -fno -dfileformat -rresno filename
```

其中，*no*是感兴趣的图形窗口的图形编号，*fileformat*是表2.1中的一种文件格式，*resno*是单位为dpi的分辨率，`filename`是我们希望为文件指定的文件名。例如，要将图2.2所示图形窗口的内容以300 dpi的分辨率输出到一个名为`hi_res_rose`的`tif`文件中，可以键入

```
>> print -f1 -dtiff -r300 hi_res_rose
```

该命令会把文件`hi_res_rose`发送到当前目录中。

若在命令行中简单地键入`print`，则MATLAB会（使用默认的打印机）打印上一个显示的图形窗口的内容。也可以指定`print`的其他选项，如某个指定的打印设备。

2.5 数据类

虽然我们处理的是整数坐标，但MATLAB中的像素值本身并不是整数。表2.2中列出了MATLAB和IPT为表示像素值所支持的各种数据类[①]。表中的前8项称为数值数据类，第9项称为字符类，最后一项称为逻辑数据类。

MATLAB中所有的数值计算都可用`double`类来进行，所以它也是图像处理应用中最常使用的数据类。`uint8`数据类也是一种频繁使用的数据类，尤其是在从存储设备中读取数据时，因为8比特图像是实际中最常用的图像。`logical`类和使用较少的`uint16`类构成了本书集中讨论的基础数据类。但是，很多IPT函数都支持表2.2中列出的所有数据类。`double`数据类需要使用8个字节来表示一个数字，而`uint8`和`int8`只需要1个字节，`uint16`和`int16`需要2个字节，`uint32`、`int32`和`single`则需要4个字节。`char`数据类用来表示Unicode字符。一个字符串就是一个$1 \times n$字符矩阵。`logical`类矩阵中每个元素的取值只能是0和1，并且每个元素都用1字节存储在存储器中。逻辑矩阵的创建可通过函数`logical`（见2.6.2节）或相关的运算符来实现（见2.10.2节）。

表2.2 数据类。前8项称为数值数据类，第9项称为字符类，最后一项称为逻辑数据类

名称	描述
`double`	双精度浮点数，范围为 $-10^{308} \sim 10^{308}$（8比特每像素）
`unit8`	无符号8比特整数，范围为 [0, 255]（1比特每像素）
`uint16`	无符号16比特整数，范围为 [0, 65 535]（2比特每像素）
`uint32`	无符号32比特整数，范围为 [0, 4 294 967 295]（4比特每像素）
`int8`	有符号8比特整数，范围为 [-128, 127]（1比特每像素）
`int16`	有符号16比特整数，范围为 [-32 768, 32 767]（2比特每像素）
`int32`	有符号32比特整数，范围为 [-2 147 483 648, 2 147 483 647]（4比特每像素）
`single`	单精度浮点数，范围为 $-10^{38} \sim 10^{38}$，（4比特每像素）
`char`	字符（2比特每像素）。
`logical`	值为0或1（1比特每像素）

① MATLAB文献经常互换地使用术语“数据类”和“数据类型”。本书中，如2.6节讨论的那样，我们把“类型”术语留给图像使用。

2.6 图像类型

工具箱支持以下4种图像类型：

- 亮度图像（Intensity images）
- 二值图像（Binary images）
- 索引图像（Indexed images）
- RGB图像（RGB images）

大多数单色图像的处理运算是通过二值图像或亮度图像来进行的，所以我们首先重点研究这两种图像。索引图像和RGB图像将在第6章中讨论。

2.6.1 亮度图像

一幅亮度图像是一个数据矩阵，其归一化的取值表示亮度。若亮度图像的像素都是uint8类或uint16类，则它们的整数值范围分别是[0, 255]和[0, 65 535]。若图像是double类，则像素的取值就是浮点数。规定双精度型归一化亮度图像的取值范围是[0, 1]。

2.6.2 二值图像

二值图像在MATLAB中具有非常特殊的意义。一幅二值图像是一个取值只有0和1的逻辑数组。因而，一个取值只包含0和1的uint8类数组，在MATLAB中并不认为是二值图像。使用logical函数可以把数值数组转换为二值数组。因此，若A是一个由0和1构成的数值数组，则可使用如下语句创建一个逻辑数组B：

```
B = logical(A)
```

若A中含有除了0和1之外的其他元素，则使用logical函数就可以将所有非零的量变换为逻辑1，而将所有的0值变换为逻辑0。使用关系和逻辑运算符也可以创建逻辑数组（见2.10.2节）。

要测试一个数组是否为逻辑数组，可以使用函数islogical：

```
islogical(c)
```

若C是逻辑数组，则该函数返回1；否则，返回0。使用2.7.1节所讨论的数据类转换函数，可将逻辑数组转换为数值数组。

2.6.3 术语注释

前两节用了大量的笔墨来阐明术语“数据类”和“图像类型”。通常，在我们提到一幅图像时，是指一幅“data_class image_type图像”，其中的data_class是表2.2中的类之一，而image_type则是本节开始时定义的图像类型之一。因此，一幅图像的特性是由数据类和图像类型这两者来表征的。例如，“uint8亮度图像”表示一幅像素都是uint8数据类的亮度图像。工具箱中的有些函数支持所有的数据类，而有些函数只支持特殊的数据类。例如，前面提到的二值图像中的像素只能是logical数据类。

2.7 数据类与图像类型间的转换

在IPT应用中，数据类与图像类型间的转换是非常频繁的操作。在数据类间转换时，记住表2.2中详细列出的每种数据类的取值范围是很重要的。

2.7.1 数据类间的转换

数据类间的转换相当直接。通用的语法为

```
B = data_class_name(A)
```

其中，data_class_name 可以是表 2.2 中第一列的任何一项。例如，假设 A 是一个 uint8 类数组，则命令 B = double(A) 会产生一个双精度数组 B。这种转换贯穿全书，因为 MATLAB 希望数值计算中的所有操作数都是双精度浮点数。假设 C 是一个取值范围为[0, 255]（很有可能包含小数）的 double 类数组，则命令 D = uint8(C)可将其转换为一个 uint8 类数组。

若一个 double 类数组包含有区间[0, 255]之外的值，则在使用上述方法将其转换成 uint8 类数组时，MATLAB 会将所有小于 0 的值转换为 0，所有大于 255 的值转换为 255，而在 0 和 255 之间的值将全部舍去小数部分转换为整数。因此，在将 double 类数组转换成 uint8 类数组之前，有必要先对其进行适当的缩放，以使其元素的取值尽量在区间[0, 255]内。如2.6.2节中提到的那样，在将任何数值数据类转换为 logical 类时，数组中的所有非 0 值将转换为逻辑 1，0 值将转换为逻辑 0。

2.7.2 图像类和类型间的转换

工具箱中提供了执行必要缩放的函数（见表 2.3），以在图像类和类型间进行转换。函数 im2uint8 可以检测出输入的数据类，并进行所有必要的缩放,以便使工具箱能将这些数据识别为有效的图像数据。例如，考虑下面这个 double 类 2 × 2 图像 f，它可以是中间计算的结果：

```
f =
    -0.5   0.5
    0.75   1.5
```

执行转换

```
>> g = im2uint8(f)
```

得出结果

```
g =
     0     128
   191     255
```

可以看出，函数 im2uint8 将输入中所有小于 0 的值设置为 0，而将输入中所有大于 1 的值设置为 255，再将所有的其他值乘以 255。将得到的结果四舍五入为最接近的整数后，就完成了转换。注意，im2uint8 的舍入行为与前一节中讨论的数据类转换函数 uint8 是不一样的，后者只是简单地将小数部分全部舍去。

表 2.3 IPT 中用于进行图像类和类型间的转换的函数。适用于彩色图像的转换请参阅表 6.3

名称	将输入转换为	有效的输入图像数据类
im2uint8	uint8	logical, uint8, uint16 和 double
im2uint16	uint16	logical, uint8, uint16 和 double
mat2gray	double，范围为[0, 1]	double
im2double	double	logical, uint8, uint16 和 double
im2bw	logical	uint8, uint16 和 double

要把一个 double 类的任意数组转换成取值范围为[0, 1]的归一化 double 类数组，可以通过函数 mat2gray 完成，其基本语法为

转换为一幅这样的二值图像，即原图像中的1和2变为0，其余两个值变为1。首先，我们将原图像的取值范围变换为区间[0, 1]：

```
>> g = mat2gray(f)
g =
         0    0.3333
    0.6667    1.0000
```

然后，我们使用阈值0.6将其转换为二值图像：

```
>> gb = im2bw(g, 0.6)
gb =
     0   0
     1   1
```

正如在2.5节中提到的那样，我们可以使用关系运算符（见2.10.2节）直接生成一个二值数组。这样，我们就可使用如下语句得到相同的结果：

```
>> gb = f > 2
gb =
     0   0
     1   1
```

我们可以使用函数islogical将gb是一个逻辑数组的事实存储为一个变量（如gbv）：

```
>> gbv = islogical(gb)
gbv =
     1
```

(b) 假设我们现在要将gb转换为一个值为0和1的double类数值数组。实现的方法如下：

```
>> gbd = im2double(gb)
gbd =
     0   0
     1   1
```

若gb是一个uint8类数值数组，则对其使用函数im2double可以得到取值为

```
     0         0
0.0039    0.0039
```

的数组，因为函数im2double会将所有的元素都除以255。这在前面的转换中都没有出现过，因为那时函数im2double检测到输入是一个值只能为0和1的逻辑数组。若输入是一个uint8类数值数组，则在保持其值为0和1的前提下要将它转换为double类数组，可以使用如下语句完成：

```
>> gbd = double(gb)
gbd =
     0   0
     1   1
```

最后要指出的是，MATLAB支持嵌套语句。因此，上述对图像f产生相同结果的所有操作，可只用单行语句

```
>> gbd = im2double(im2bw(mat2gray(f), 0.6));
```

来完成，或者使用这些函数的部分组合来完成。当然，在这种情况下，整个过程也可以用一条更简单的命令实现：

```
>> gbd = double(f > 2);
```

这又一次证明了MATLAB语言的简洁性。

2.8 数组索引

MATLAB支持大量功能强大的索引方案，这些索引方案不仅简化了数组操作，而且提高了程序的运行效率。本节将讨论并举例说明一维和二维（向量和矩阵）的基本索引。后面的讨论将根据需要介绍更复杂的技术。

2.8.1 向量索引

如我们在2.1.2节中讨论的那样，维数为$1 \times N$的数组称为行向量。行向量中元素的存取是使用一维索引进行的。因此，v(1)是向量v的第一个元素，v(2)是第二个元素，依次类推。MATLAB中向量的元素使用方括号括起，并由空格或逗号隔开。例如，

```
>> v = [1  3  5  7  9]
v =
    1  3  5  7  9
>> v(2)
ans =
    3
```

使用转置运算符（.'）可将行向量转换为列向量：

```
>> w = v.'
w =
   1
   3
   5
   7
   9
```

要存取元素的数据块，我们可使用MATLAB的冒号。例如，要存取v的前三个元素，可使用语句

```
>> v(1:3)
ans =
    1  3  5
```

类似地，也可以使用如下语句存取第二个到第四个元素：

```
>> v(2:4)
ans =
    3  5  7
```

或使用如下语句存取第三个到最后一个元素：

```
>> v(3:end)
ans =
    5  7  9
```

其中，end表示向量中的最后一个元素。若v是一个向量，则

```
>> v(:)
```

产生一个列向量，而语句

```
>> v(1:end)
```

产生一个行向量。

索引并不限于连续的元素。例如，

```
>> v(1:2:end)
ans =
    1   5   9
```

注意，符号1:2:end表示索引从1开始计数，步长为2，直到最后一个元素时停止。步长也可以为负：

```
>> v(end:-2:1)
ans =
    9   5   1
```

这时，索引从最后一个元素开始计数，步长为-2，直到第一个元素时停止。

函数linspace的语法为

```
x = linspace(a, b, n)
```

该语句产生一个含有n个元素的行向量x，这n个元素之间线性地隔开并且包含a与b。后续章节会在一些地方使用到该函数。

一个向量也可用做另一个向量的索引。例如，我们可以使用如下命令挑出向量v的第一个、第四个和第五个元素：

```
>> v([1   4   5])
ans =
    1   7   9
```

就像下一节所示的那样，使用一个向量作为另一个向量的索引这种功能，在矩阵索引中起着非常重要的作用。

2.8.2 矩阵索引

在MATLAB中，矩阵可以很方便地用一列被方括号括起并用分号隔开的行向量表示。例如，

```
>> A = [1   2   3;   4   5   6;   7   8   9]
```

显示了3 × 3矩阵

```
A =
   1   2   3
   4   5   6
   7   8   9
```

注意，此处分号的作用，与前面提到的取消命令行的输出或在一行中写入多条命令时所用分号的作用是不同的。

从矩阵中选取元素和从向量中选取元素是一样的，但我们现在需要两个索引：一个用于确定行位置，另一个用于确定相应的列位置。例如，要提取第2行第3列的元素，可使用语句

```
5
8
3
6
9
```

使用单个下标存储A可直接对该列索引。例如，A(3)存取的是列中的第三个值，即数字7；A(8)存取的是第8个值，即数字6，如此等等。当我们使用这个列符号时，表示我们正简单地对所有元素A(1:end)寻址。这种类型的索引是为优化程序而使循环向量化的基本成分，详见2.10.4节中的讨论。

例2.5 使用数组索引进行简单的图像操作

图2.6(a)是一幅大小为1024 × 1024的uint8类亮度图像f。图2.6(b)是该图像经过如下语句操作后垂直翻转的图像：

```
>> fp = f(end:-1:1, :);
```

图2.6(c)所示的图像是图像(a)中的一部分，由以下语句获得：

```
>> fc = f(257:768, 257:768);
```

类似地，图2.6(d)是使用如下语句进行二次取样后得到的图像：

```
>> fs = f(1:2:end, 1:2:end);
```

最后，图2.6(e)显示了通过图2.6(a)中部的一条水平扫描线，它是使用如下语句得到的：

```
>> plot(f(512, :))
```

函数plot将在3.3.1节中详细讨论。

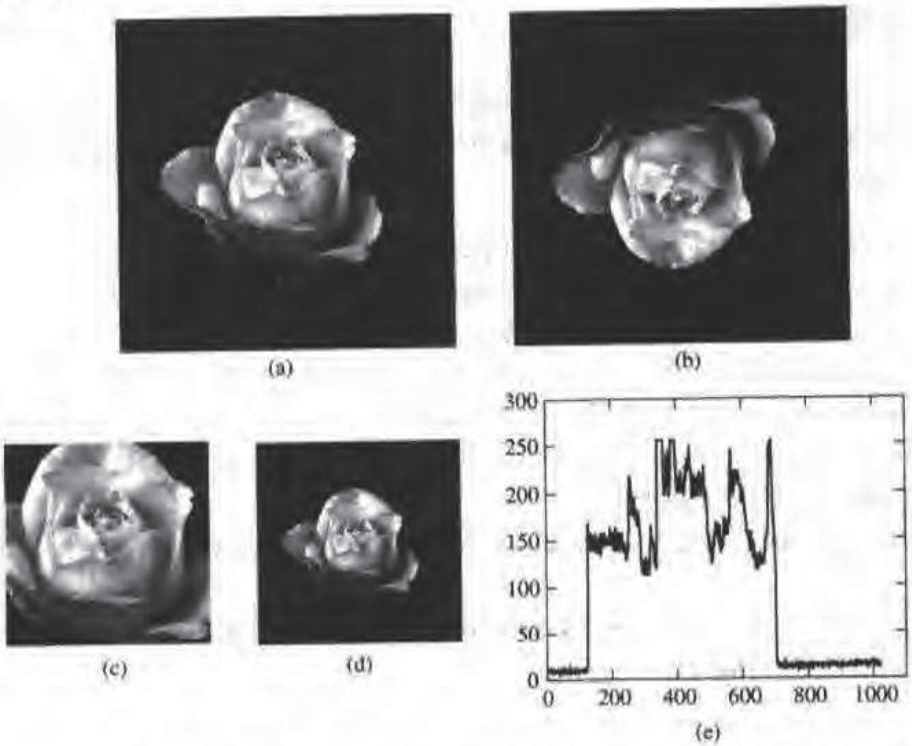


图2.6 使用数组索引得到的结果：(a)原图像；(b)垂直翻转的图像；(c)裁剪后的图像；(d)二次取样后的图像；(e)通过图(a)中间的水平扫描线

2.8.3 选择数组的维数

本书中频繁地使用了形如

```
operation(A, dim)
```

的操作，operation表示MATLAB中的一种可用操作，A是一个数组，dim是一个标量。例如，假设A是一个大小为$M \times N$的数组。命令

```
>> k = size(A, 1);
```

沿A的第一个维数（在MATLAB中定义为垂直方向）给出A的大小。换言之，该命令给出A的行数。类似地，数组的第二个维数为水平方向，所以语句size(A,2)给出A的列数。单一维数是任意维数dim，且size(A,dim) = 1。使用这些概念，我们可将例2.5中的最后一条命令写为

```
>> plot(f(size(f, 1)/2, : ))
```

MATLAB并不限制数组的维数，因此，以任何一个维数来提取某个数组的分量就成为了一个重要的特性。大部分情况下，我们只处理二维数组，但有时（如处理彩色图像或者多谱段图像时）需要将图像“堆叠”到第三维或更高维上去。这些内容将会在第6章、第11章和第12章中详细解释。函数ndims的语法为

```
d = ndims(A)
```

它将给出数组A的维数。函数ndims返回的值不会小于2，因为即使是标量，我们也认为它有两个维数，这时的标量是大小为1 × 1的数组。

2.9 一些重要的标准数组

在开发期间，用一些简单的图像数组来检验算法并测试函数的语法常常是很有用的。本节将介绍7种数组生成函数，这些函数将在后面的章节中用到。在下面的任何一个函数中，若只包含一个参量，则结果将是一个方阵。

- zeros(M,N)生成一个大小为M × N的double类矩阵，其元素均为0。
- ones(M,N)生或一个大小为M × N的double类矩阵，其元素均为1。
- ture(M,N)生成一个大小为M × N的logical类矩阵，其元素为1。
- false(M,N)生成一个大小为M × N的logical类矩阵，其元素为0。
- magic(M)生成一个大小为M × N的“魔术方阵”。在该方阵中，每一行中的元素之和、每一列中的元素之和以及主对角线中的元素之和均相等。魔术方阵可用于测试目的，因为它们易于生成，且其元素均为整数。
- rand(M,N)生成一个大小为M × N的矩阵，矩阵中的元素都是在区间[0, 1]中均匀分布的随机数。
- randn(M,N)生成一个大小为M × N的矩阵，矩阵中的元素是正态分布（如高斯分布）的随机数，随机数的均值为0，方差为1。

例如，

```
>> A = 5*ones(3, 3)
A =
```

```
     5     5     5
     5     5     5
     5     5     5
>> magic(3)
ans =
     8     1     6
     3     5     7
     4     9     2
>> B = rand(2, 4)
B =
    0.2311    0.4860    0.7621    0.0185
    0.6068    0.8913    0.4565    0.8214
```

2.10 M函数编程简介

图像处理工具箱最强大的特征之一是其对MATLAB编程环境的透明访问。MATLAB函数编程非常灵活，并且非常容易学习，随着我们学习的深入，这一特性将会很快呈现。

2.10.1 M文件

MATLAB中的M文件可以是简单执行一系列MATLAB语句的脚本，也可以是接受变量并产生一个或多个输出的函数。本节重点介绍M文件函数。这些函数将MATLAB和IPT的功能扩展到了寻址用户定义的特定应用。

M文件由文本编辑器创建，并以filename.m形式的文件名存储，如average.m和filter.m。M文件函数的组成部分为

- 函数定义行
- H1行
- 帮助文本
- 函数体
- 命令

函数定义行的形式为

```
function [outputs] = name(inputs)
```

例如，计算两幅图像的和与积（两个不同的输出）的函数的形式为

```
function [s, p] = sumprod(f, g)
```

其中f和g是输入图像，s是和图像，p是积图像。名称sumprod可任意定义，但字function必须出现在左侧，如上所示。注意，输出变量必须位于方括号内，而输入变量必须位于圆括号内。若函数只有单个输出变量，则也可不使用括号而直接列出。若函数没有输出，则只需要使用字function，而无须括号或等号。函数名必须以字母开头，后面可以跟字母、数字、下划线的任意组合，但不允许有空格。MATLAB可以识别长达63个字符的函数名，超过此长度的字符将被忽略。

函数可以在命令提示符处调用；例如，

```
>> [s, p] = sumprod(f, g);
```

也可以用做其他函数的元素，此时的函数就成为了子函数。正如前一段提到的那样，若输出只有一个变量，则也可不使用括号，例如

```
>> y = sum(x);
```

H1 行是第一个文本行。它是单个注释行，其前面为函数定义行。H1 行与函数定义行间无空行或前导空格。H1 行的一个例子为

```
% SUMPROD Computes the sum and product of two images.
```

如我们在 1.7.3 节中提到的那样，当用户在 MATLAB 提示符处键入

```
>> help function_name
```

时，H1 行是最先出现的文本。此外，该节还提到键入 `lookfor keyword` 会显示出所有含有字符串 `keyword` 的 H1 行。该行提供了关于 M 文件的重要摘要信息，所以应尽可能地描述它。

帮助文本是紧跟在 H1 行后面的文本块，二者之间无空行。帮助文本用来为函数提供注释或在线帮助。当用户在提示符处键入 `help function_name` 时，MATLAB 会显示函数定义行和第一个非注释（可执行或空白的）行之间的全部注释行。帮助系统会忽略帮助文本块后的任何注释行。

函数体包含了所有执行计算并给输出变量赋值的 MATLAB 代码。MATLAB 代码的几个例子将在本章后面给出。

符号“%”后面非 H1 行或帮助文本的所有行可看做是函数注释行，它们不是帮助文本的一部分。代码行的末尾可附加注释。

M 文件可以使用任何文本编辑器创建和编辑，并以扩展名 `.m` 保存到指定的目录下，一般会保存在 MATLAB 搜索路径中。创建 M 文件的另一种方法是在提示符处使用 `edit` 函数。例如，若文件存在于 MATLAB 搜索路径的目录中或者在当前目录中，则键入

```
>> edit sumprod
```

会打开文件 `sumprod.m` 并进行编辑。若找不到该文件，MATLAB 会为用户提供创建该文件的选项。如 1.7.2 节中提到的那样，MATLAB 编辑器窗口有很多下拉菜单，可以完成诸如保存文件、查看文件以及调试文件等任务。由于文本编辑器可以执行一些简单的检查并使用不同的颜色来区分各种代码元素，所以建议用户在写 M 文件或编辑 M 文件时使用该文本编辑器。

2.10.2 运算符

MATLAB 运算符可以分为以下三种主要类别：

- 执行数值计算的算术运算符
- 在数量上比较操作数的关系运算符
- 执行函数 AND、OR 和 NOT 的逻辑运算符

本节的剩余部分将讨论这些运算符。

算术运算符

MATLAB 有两种不同类型的算术运算符。矩阵算术运算符由线性代数的规则定义。数组算术运算符可以逐个元素地执行，并且可以用于多维数组。句点（圆点）字符（.）用来区分数组运算符与矩阵运算符。例如，`A*B` 表示传统意义上的矩阵乘法，而 `A.*B` 则表示数组乘法，这种乘法的

乘积是与A和B大小相同的数组，其每个元素都是A和B中相应元素的乘积。换言之，若C=A.*B，则C(I,J)=A(I,J)*B(I,J)。由于对加法和减法来说，矩阵运算和数组运算是相同的，所以不使用字符对.+和.-。

在编写如B=A这样的表达式时，MATLAB将做一个B等于A的“记录”，但实际上并不将数据复制到B，除非在程序中A的内容以后有了变化。这一点很重要，因为使用不同的变量来“存储”相同的内容有时可以增强代码的清晰性和可读性。因此，在编写MATLAB代码时，一定要记住MATLAB不复制信息这一功能。表2.4列出了MATLAB的算术运算符，其中A与B是矩阵或数组，a与b是标量。所有的操作数可以是实数或复数。若操作数是标量，则数组运算符中的小圆点可以省略。记住，图像是等价于矩阵的二维数组，因此，表中的所有运算符都适用于图像。

表2.4 数组和矩阵算术运算符。涉及这些运算符的计算可使用运算符本身实现，如A+B，或使用所示的MALAB函数实现，如plus(A,B)。为数组显示的示例使用矩阵来简化表示，但它们可很容易地扩展到高维情形

运算符	名称	MATLAB函数	注释和示例
+	数组和矩阵加	plus(A,B)	a+b,A+B或a+A
-	数组和矩阵减	minus(A,B)	a-b,A-B,A-a或a-A
.*	数组乘	times(A,B)	C=A.*B,C(I,J)=A(I,J)*B(I,J)
*	矩阵乘	mtimes(A,B)	A*B,即标准矩阵乘；或a*A，即一个标量乘以A的所有元素
./	数组右除	rdivide(A,B)	C=A./B,C(I,J)=A(I,J)/B(I,J)
.\	数组左除	ldivide(A,B)	C=A.\B,C(I,J)=B(I,J)/A(I,J)
/	矩阵右除	mrdivide(A,B)	A/B与A*inv(B)大致相同，具体取决于计算精度
\	矩阵左除	mldivide(A,B)	A\B与inv(A)*B大致相同，具体取决于计算精度
.^	数组求幂	power(A,B)	若C=A.^B，则C(I,J)=A(I,J)^B(I,J)
^	矩阵求幂	mpower(A,B)	该运算符的讨论请参阅在线帮助
.'	向量和矩阵转置	transpose(A)	A.'。标准的向量和矩阵转置
'	向量和矩阵复共轭转置	ctranspose(A)	A'。标准的向量和矩阵共轭转置。当A是实数时，A.'=A'
+	一元加	uplus(A)	+A与0+A相同
-	一元减	uminus(A)	-A与0-A或-1*A相同
:	冒号		详见2.8节的讨论

表2.5 IPT支持的图像算术函数

函数	描述
imadd	两幅图像相加或把常数加到图像
imsubtract	两幅图像相减或从图像中减去常数
immultiply	两幅图像相乘，其中相乘是在相应的像素对间进行的；或图像乘以一个常数
imdivide	两幅图像相除，其中相除是在相应的像素对间进行的，或图像除以一个常数
imabsdiff	计算两幅图像间的绝对差
imcomplement	对图像求补，参见3.2.1节
imlincomb	计算两幅或多幅图像的线性组合，示例请参阅5.3.1节

工具箱支持表2.5中列出的图像算术函数。尽管这些函数可以使用MATLAB算术运算符直接实现，但使用IPT函数的优点在于它们支持整数数据类，而等同的MATLAB数学运算符要求输入是double类数据。

后面的例2.6用到了函数max和min。前者的语法形式有如下几种：

```
C = max(A)
C = max(A, B)
C = max(A, [ ], dim)
[C, I] = max(...)
```

在第一种形式中，若A是一个向量，则max(A)返回其最大元素；若A是一个矩阵，则max(A)将A的列作为向量来处理，并返回一个包含了每列最大值的行向量。在第二种形式中，max(A,B)返回一个与A和B大小相同的数组，该数组由A与B中最大的元素组成。在第三种形式中，max(A, [ ],dim)返回沿标量dim指定的A的维度上的最大元素。例如，max(A,[ ],1)产生的是A的第一维（行）上的最大值。最后，[C,I]=max(...)会找出A的最大值的索引，并将它们返回到输出向量I中。若有几个相等的最大值，则返回找到的第一个最大值的索引。圆点表示使用的语法是前三种形式的右侧这一。函数min的语法形式与函数max的语法形式类似。

例2.6 算术运算符与函数max和min的示例

假设我们要编写一个称为fgprod的M函数，该函数的功能是将两幅输入图像相乘，并输出图像的乘积、乘积的最大值与最小值以及一幅归一化的乘积图像（其取值范围为[0, 1]）。使用文本编辑器，我们可写出期望的函数，如下所示：

```
function [p, pmax, pmin, pn] = improd(f, g)
%IMPROD Computes the product of two images.
%  [P, PMAX, PMIN, PN] = IMPROD(F, G)① outputs the element-by-
%  element product of two input images, F and G, the product
%  maximum and minimum values, and a normalized product array with
%  values in the range [0, 1]. The input images must be of the same
%  size.  They can be of class uint8, unit16, or double. The outputs
%  are of class double.
fd = double(f);
gd = double(g);
p = fd.*gd;
pmax = max(p(:));
pmin = min(p(:));
pn = mat2gray(p);
```

注意，输入图像是使用函数double而不是函数im2double来转换为double类图像的。因为，若输入是uint8类图像，则函数im2double会将它们转换到范围[0, 1]。假定我们想将原始值的乘积放在p中。为得到一个取值范围为[0, 1]的归一化数组pn，我们可以使用函数mat2gray。还要注意单个冒号索引的使用，详见2.8节中的讨论。

假设f=[1 2;3 4]，g=[1 2;2 1]。在提示符处键入上述函数可以得到如下输出：

```
>> [p, pmax, pmin, pn] = improd(f, g)
p =
     1     4
     6     4
pmax =
```

① 在MATLAB文献中，在指函数名和参量时，在H1行和帮助文本中常常使用大写字母，以避免程序名或变量与通常的解释性文本相混淆。

```
     6
pmin =
     1
pn =
           0    0.6000
      1.0000    0.6000
```

在提示符处键入 help improd 可以得到如下输出：

```
>> help improd

IMPROD Computes the product of two images.
  [P, PMAX, PMIN, PN] = IMPROD(F, G) outputs the element-by-
  element product of two input images, F and G, the product
  maximum and minimum values, and a normalized product array with
  values in the range [0, 1]. The input images must be of the same
  size.  They can be of class uint8, unit16, or double. The outputs
  are of class double.
```

关系运算符

表2.6列出了MATLAB的关系运算符。这些运算符可以对相同维数的数组中的相应元素进行比较，或逐个元素地进行比较。

例2.7 关系运算符

尽管关系运算符的关键应用是在2.10.3节讨论的流控制中（如在if语句中），但这里还是简单地演示一下这些运算符是怎样直接用于数组的。考虑下面的输入和输出序列：

```
>> A = [1 2 3; 4 5 6; 7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
>> B = [0 2 4; 3 5 6; 3 4 9]
B =
     0     2     4
     3     5     6
     3     4     9
>> A == B
ans =
     0     1     0
     0     1     1
     0     0     1
```

可以看到，运算符A == B生成一个与A和B维数相同的逻辑数组，当A与B的相应元素匹配时，新数组中的相应位置取1，其余位置取0。考虑另一个例子，语句

```
>> A >= B
ans =
     1     1     0
     1     1     1
     1     1     1
```

生成一个逻辑数组，当A的元素大于或等于B的相应元素时，新数组中的相应位置取1，其余位置取0。

表2.6 关系运算符

运算符	名称
<	小于
<=	小于等于
>	大于
>=	大于等于
==	等于
~=	不等于

对于向量和矩形数组，两个操作数必须有相同的维数，或者其中一个操作数是标量。此时，MATLAB将这个标量与另一个操作数的每一个元素相比较，产生一个与操作数大小相同的逻辑数组，在满足指定关系的位置取1，其余位置取0。若两个操作数都是标量，则当指定关系满足时结果为1，否则为0。

逻辑运算符与函数

表2.7列出了MATLAB的逻辑运算符，且下面的例子说明了其中一些运算符的性质。不同于逻辑运算符的通用描述，表2.7中的运算符既能作用于逻辑数据又能作用于数值数据。在所有的逻辑测试中，MATLAB将逻辑1或非零数值量作为true来处理，将逻辑0和数值0作为false来处理。例如，当两个操作数都为逻辑1或非零数值时，两个操作数AND运算的结果为1；当两个操作数中的任何一个是逻辑0或数值0时，或两个操作数都为逻辑0或数值0时，AND运算的结果为0。

表2.7 逻辑运算符

运算符	名称
&	AND（与）
|	OR（或）
~	NOT（非）

例2.8 逻辑运算符

考虑下面两个数值数组的AND运算：

```
>> A = [1 2 0; 0 4 5];
>> B = [1 -2 3; 0 1 1];
>> A & B

ans =
     1     1     0
     0     1     1
```

可以看出，AND运算符产生一个与输入数组大小相同的逻辑数组，并在两个输入数组相应操作数都为非零值的位置取1，其他位置取0。像前面一样，每次运算都在输入数组的相应位置的一对操作数上进行。

OR运算符的使用方式与AND运算符的使用方式类似。当两个操作数中任何一个为逻辑1或非零数值时，或两个操作数都为逻辑1或非零数字时，OR运算表达式的值为ture；否则为false。NOT运算符只作用于单个输入。逻辑上，若一个操作数为ture，则NOT运算符会将其转换为false。当将NOT作用于数值数据时，所有非零操作数将变为0，零操作数变为1。

MATLAB也支持表2.8中总结的逻辑函数。在编程中，函数all与any是非常有用的。

例2.9 逻辑函数

考虑简单的数组A=[1 2 3;4 5 6]与B=[0 -1 1;0 0 2]。将表2.8中的函数作用于这两个数组，可得到如下结果：

```
>> xor(A, B)
ans =
     1     0     0
     1     1     0
>> all(A)
ans =
     1     1     1
>> any(A)
ans =
     1     1     1
>> all(B)
ans =
     0     0     1
>> any(B)
ans =
     0     1     1
```

表2.8 逻辑函数

函数	注释
xor（异或）	若两个操作数逻辑上不同，则函数xor返回1；否则，返回0
all	若一个向量中的所有元素都非零，则函数all返回1；否则，返回0
any	若一个向量中的任何元素都非零，则函数any返回1；否则，返回0。该函数在矩阵中按列操作

仔细观察函数all与any是如何作用于A和B的列的。例如，由all(B)产生的向量的前两个元素为0，因为B的前两列中每一列都至少含有一个0；最后一个元素为1，因为B的最后一列的所有元素都非零。

除了表2.8中列出的函数，MATLAB还提供了很多其他的函数，这些函数可以检测到特殊条件或特殊值的存在并返回逻辑的结果。表2.9列出了一些这样的函数。其中一小部分涉及到了本章前面提到的一些术语和概念（如2.6.2节中的islogical函数）；其他的函数将在后面的讨论中用到。记住，表2.9中列出的函数在条件满足时将返回逻辑1，否则将返回逻辑0。当变量是数组时，表2.9中的一些函数会产生一个与输入变量大小相同的数组，在满足函数条件的位置取逻辑1，其他位置取逻辑0。例如，假设A=[1 2;3 1/0]，函数isfinite(A)会返回矩阵[1 1;1 0]，该矩阵中的0（假）表示A的最后一个元素不是有限值。

表2.9 在变量为ture或false时，根据数值或条件返回逻辑1或0的某些函数

函数	描述
iscell(C)	若C是单元数组，则为真
iscellstr(s)	若s是字符串单元数组，则为真
ischar(s)	若s是字符串，则为真
isempty(A)	若A是空数组[]，则为真
isequal(A,B)	若A和B有相同的元素和维数，则为真
isfield(S,'name')	若'name'是结构S的一个域，则为真
isfinite(A)	若数组A的元素有限，则为真

```
    error('The dimensions of the input cannot exceed 2.')
end
% Compute the average
av = sum(A(:))/length(A(:));
```

注意，输入已使用A(:)转换为一个一维数组。一般来说，length(A)返回数组A的最长维的大小。在该例中，由于A(:)是一个向量，所以length(A)给出了A中的元素个数，从而无须测试输入是一个向量还是一个二维数组。另一种直接获得数组中的元素个数的方法是使用函数numel，其语法为

```
n = numel(A)
```

因此，若A是一幅图像，则numel(A)将给出它的像素数。使用这个函数时，前一个程序中最后一条可执行的语句就变为

```
av = sum(A(:))/numel(A);
```

最后要注意的是，函数error将终止程序的执行并且输出括号内的信息（引号不能省略）。

for

如表2.11所示，一个for循环按指定的次数执行一组语句。其语法为

```
for index = start:increment:end
    statements
end
```

可以嵌套两个或更多的for循环，如下所示：

```
for index1 = start1:increment1:end
   statements1
   for index2 = start2:increment2:end
      statements2
   end
   additional loop1 statements
end
```

例如，下面的循环将执行11次：

```
count = 0;
for k = 0:0.1:1
   count = count + 1;
end
```

若省略了循环增量，则其默认为1。循环增量也可以为负，如k = 0:-1:-10。注意，每个for行的结尾无须加分号。MATLAB能够自动地停止打印循环索引的值。如我们将在2.10.4节中讨论的那样，利用所谓的向量化代码代替for循环，程序的执行速度会有明显的提高。

例2.11 使用for循环将多幅图像写入文件

例2.2中比较了几幅使用不同JPEG质量值的图像。这里，我们将给出使用for循环来把这些文件写入磁盘的方法。假设存在一幅图像f，我们想将其写为一系列JPEG文件，这些文件用范围在0至100、增量为5的质量因子表示。进而，假设我们想将这些JPEG文件用series_xxx.jpg来命名，其中xxx是品质因子。使用下面的for循环，我们可完成这项任务：

```
for q = 0:5:100
   filename = sprintf('series_%3d.jpg', q);
   imwrite(f, filename, 'quality', q);
end
```

在此例中，函数sprintf的语法为

```
s = sprintf('characters1%ndcharacters2', q)
```

它会将格式化的数据写为一个字符串s。在这种语法形式中，characters1和characters2是字符串，%nd表示一个n位十进制数（由q决定）。在此例中，characters1代表series_，n的值是3，characters2代表.jpg，q的值在循环中指定。

while

只要控制循环的表达式为true，while循环就会执行一组语句。其语法为

```
while expression
   statements
end
```

与for循环类似，while循环也可以嵌套使用：

```
while expression1
   statements1
   while expression2
      statements2
   end
   additional loop1 statements
end
```

例如，如下嵌套的while循环会在a和b均降至0时终止：

```
a = 10;
b = 5;
while a
   a = a - 1;
   while b
      b = b - 1;
   end
end
```

注意，为了控制循环，我们采用了MATLAB以逻辑形式来表达数值的习惯，即对非零数用true表示，0用false表示。换言之，只要a与b非零，则while a和while b均为真。

与for循环类似，利用向量化代码（见2.10.4节）代替while循环，程序的执行速度会有明显的提高。

break

正如其名称所示的那样，break终止for或while循环的执行。遇到break语句时，程序会继续执行循环外的下一条语句。在嵌套循环中，break语句仅退出包含它的最内层循环。

continue

continue语句将控制传递给for或while循环的下一次迭代，而跳过循环体中的任何其他语句。在嵌套循环中，continue语句会将控制传递给包含该语句的循环的下一次迭代。

switch

这是一个基于不同类型的输入来控制 M 函数的流的选择语句。其语法为

```
switch switch_expression
   case case_expression
      statement(s)
   case {case_expression1, case_expression2,...}
      statement(s)
   otherwise
      statement(s)
end
```

switch构造基于变量或表达式的值来执行语句组。关键词case和otherwise用于描述语句组。仅执行首先匹配的case[①]。通常必须有end来匹配switch语句。当同一个case语句中存在多个表达式时，则必须使用大花括号。例如，假设存在一个M函数，该函数接收一幅图像f，并将其转换为一个指定的类，称该类为newclass。该转换只对三种图像类有效，即uint8类图像、uint16类图像和double类图像。下列代码片断执行我们所期望的转换，并且会在输入图像不是所允许转换的类时输出错误信息：

```
switch newclass
   case 'uint8'
      g = im2uint8(f);
   case 'uint16'
      g = im2uint16(f);
   case 'double'
      g = im2double(f);
   otherwise
      error('Unknown or improper image class.')
end
```

全书广泛使用了switch构造。

例 2.12 从给定图像中提取子图像

在本例中，我们将基于for循环来编写一个M函数，以便从一幅图像中提取一幅矩形子图像。正如下一节将要讨论的那样，尽管我们可以使用单条MATLAB语句来完成提取，但使用这个例子可在稍后比较for循环和向量化代码的运行速度。函数的输入是一幅图像、所提取子图像的大小（行数和列数）以及子图像左上角的坐标。记住，MATLAB 中图像的原点在(1, 1)处，详见2.1.1节的讨论。

```
function s = subim(f, m, n, rx, cy)
%SUBIM Extracts a subimage, s, from a given image, f.
%  The subimage is of size m-by-n, and the coordinates
%  of its top, left corner are (rx, cy).

s = zeros(m, n);
rowhigh = rx + m - 1;
colhigh = cy + n - 1;
xcount = 0;
```

① 与C语言中的switch构造不同，MATLAB中的switch不会“失败”。换言之，switch仅执行第一个匹配的情形，而不执行后续的匹配情形。因此，未使用break语句。