Algorithms for Image Processing and Computer Vision, Second Edition

图像处理与计算机视觉算法及应用（第2版）

[加拿大] J. R. Parker 著　　景　丽　译

清华大学出版社

图像处理与计算机视觉算法及应用

(第 2 版)

[加拿大] J. R. Parker 著

景 丽 译

清华大学出版社

北 京

J. R. Parker
Algorithms for Image Processing and Computer Vision，Second Edition
EISBN：978-0-470-64385-3

图书在版编目(CIP)数据

图像处理与计算机视觉算法及应用 /(加拿大) 帕科尔(Parker，J. R.) 著；景丽 译. —2 版.
—北京：清华大学出版社，2012.5

书名原文：Algorithms for Image Processing and Computer Vision，Second Edition

ISBN 978-7-302-28222-8

Ⅰ. ①图… Ⅱ. ①帕… ②景… Ⅲ. ①图像处理 ②计算机视觉 Ⅳ. ①TP391.41

中国版本图书馆 CIP 数据核字(2012)第 038935 号

责任编辑：王　军　吴　乐
装帧设计：牛艳敏
责任校对：邱晓玉
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：北京密云胶印厂
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：25.25　　**字　　数**：614 千字
版　　次：2012 年 5 月第 1 版　　**印　　次**：2012 年 5 月第 1 次印刷
印　　数：1～4000
定　　价：49.00 元

产品编号：041244-01

作者简介

J. R. Parker 是一名计算机专家和教师，对图像处理和视觉、视频游戏技术以及计算机仿真有着浓厚的兴趣。在根特州立大学获得信息学博士学位之后，Parker 博士在加拿大卡尔加里大学任全职教授，讲授计算机科学、艺术和戏剧。他的作品包括 150 多篇技术论文和 4 本书，他还是视频游戏 Booze Cruise 的作者，这个游戏模拟了酒后驾车的情形，用于演示酒后驾车行为的危险性。此外他还开发了很多其他教育游戏。

技术编辑简介

Kostas Terzidis 是哈佛大学设计研究生院的副教授。他拥有密歇根大学的建筑学博士学位(1994 年)，俄亥俄州立大学的建筑学硕士学位(1989 年)以及亚里士多德大学的工程学本科学位(1986)。他最近热衷于开发建筑学相关的算法的理论和技术。他的著作 *Expressive Form: A Conceptual Approach to Computational Design*(2003 年由伦敦的出版社 Spon Press 出版)提供了独特的视角，将计算技术应用于美学，特别是建筑学和设计；*Algorithmic Architecture* 一书(2006 年由 Architectural Press/Elsevier 出版)为算法建筑学的术语、概念和过程进行了本体论的研究，并为设计实现提供了理论框架。他的最新著作 Algorithms for Visual *Design*(2009 年由 Wiley 出版)为学生、程序员和研究者提供了技术性、理论性和设计性的方法，用于开发实验设计问题的计算机代码。

致　谢

这一次要感谢 Sonny Chan，是他激发了我并行计算这一章的灵感；感谢 Jeff Boyd，多次向我介绍 OpenCV；感谢 Ralph Huntsinger 和 Ghislain C. Vansteenkiste，帮助我进入并成功地完成博士计划。

本书中使用的几乎所有图像都是我自己创建的，使用的工具包括一台带有帧采集器和 Sony CCD 相机的 IBM PC，一台 HP 扫描仪和一台 Sony Eyetoy 网络摄像机。不是以这种方式获得的新图像都要归功于：

Corel 公司让我使用图 3-33 所示的树叶上的蚂蚱的图像，同时也提供了图 10-5 所示的示例搜索图像。

图 10-1 中的示例图像来自于 ALOI 数据集，这个数据集的使用经过了 J. M. Geusebroek 的同意。

感谢加拿大艾伯塔省科克伦的 Big Hill 兽医诊所，提供了图 3-10(e)所示的 X 射线图像。

最后，感谢卡尔加里大学地质系的 N. Wardlaw 博士，提供了图 3-16 的地质学微孔图像。

最重要的，我要感谢我的家庭：我的妻子 Katrin 和孩子们 Bailey 和 Max。若没有他们做出的时间和精力上的牺牲，我无法完成这本书。我非常感谢他们，也希望他们的付出有所值。

前　言

人类主要的感官输入来源是视觉系统，人们付出了很多努力致力于通过人工的方式提升这种感官能力。眼镜、望远镜、雷达、红外线传感器和光电倍增器等设备的功能都用于提升我们对世界和宇宙的观察能力。我们甚至在轨道上也建立了望远镜(外太空的眼睛)，通过很多这种望远镜能“看到”其他频谱：红外线、紫外线和 X 射线。这些设备给予了我们几年前想都不敢想的视野，还能让我们观测到裸眼无法观察到的色彩。计算机成为了通过这些设备创建我们能看见的伟大图像的必备工具。

在本书第一版写作的时候，哈勃空间望远镜就已经在轨，并能以很高的速率生成图像。哈勃空间望远镜和欧洲的依巴谷(Hipparcos)望远镜是当时在大气层以上仅有的光学设备。现在有了 COROT、Kepler、MOST(加拿大的空间望远镜)和雨燕伽马射线暴探测器(Swift Gamma Ray Burst Explorer)。此外，还有 Spitzer(红外线)、Chandra(X 射线)、GALEX(紫外线)探测器以及很多其他探测器。本书第一版是在一台带有 256MB 内存、450Mhz 奔腾 III 处理器的计算机上编写的。1999 年，第一台数字单反相机投放市场：Nikon D1。这台相机只有 274 万像素，价格为 6000 美元。一台典型的 PC 机硬盘容量为 100～200MB。网络摄像机在 1997 年就出现了，但是非常昂贵而且分辨率很低。通过计算机处理图像的人需要有一个特殊的图像获取卡和比较昂贵的照相机才能开展自己的工作，总共需要大约 1000～2000 美元的设备。相比那个年代，个人计算机和图像获取的技术已经取得了长足的发展。

1997 年，在我多次浏览与图像处理和计算机视觉相关的 Internet 新闻组的时候，萌生了写这本书第一版的想法。我注意到有一些提问请求反复出现，它们有时候能得到回答，而有时候得不到回答，因此我考虑是否能写一本书来回答这些常见的问题，书籍的形式能够提供完整解释所需要的一些必备的背景信息。然而，由于我当时刚刚完成了另一本书(*Practical Computer Vision Using C*)，因此我没心情再写一本书了。我继续在网上收集信息，希望有朝一日能以合理的方式整理这些信息。我做到了——本书第一版广受好评。感谢你们！

15 年后，技术发生了变化，但我惊讶地发现，计算机视觉和图像处理领域的变化却如此之小，至少其变化在可接受的范围内。的确，理论变得更加复杂，而且三维视觉技术也肯定得到了提升。一些机器人视觉系统已经能完成非常有意思的事情，人脸识别也步入了一个新的台阶。然而，简单的字符识别仍然非常简单，还没有进步到能够在大部分情况下可靠工作的地步。和其他类型的软件不一样，视觉系统并没有深入日常的生活。为什么呢？可能是因为视觉问题的确是一个非常棘手的问题。或许有必要对原书进行一次修订？

我的目标有所改变。我现在对这项技术的“民主化”同样感兴趣——也就是说，让任何人都能使用这项技术，不论是在家里、在商业场合还是在学校。当然，您要懂得计算机编程，不过编程技能已经比过去更加普遍。这一版中构建程序所需的所有软件都可以在 Internet 上免费获得。我使用了免费的编译器(Microsoft Visual Studio Express)，OpenCV 也能免费下载。阻碍您开发自己的图像分析系统的唯一障碍就是您自己的编程技能。

在这一版中，有一些原始的材料并没有太大的改变。边缘检测、细化、阈值化和形态学一直都不是研究的热点，因此它们在这一版中的相关章节和初版中的相关章节非常接近。软件已经升级为使用 Intel 的 OpenCV 系统，对于程序员来说，使用这个系统可以使得图像 IO 和显示的工作简单得多。甚至从网络摄像机实时获得图像并作为程序的输入也成为了一件简单的事情。第 1 章讨论了 OpenCV 使用的基础知识，本书所有的软件都使用 OpenCV 作为基础。

本书中出现的大部分数学内容都是详细理解书中所描述的算法的必备内容。只有数学才能提供图像处理和视觉系统中的高级方法所需要的推导和证明。在某些情况下，我只涉及了表面知识，更细节的研究留给那些愿意根据每章末的参考文献按图索骥的读者。我精心选择了描述不同方法的文献，有的文献提供了详细且复杂的数学分析，有的文献提供了清晰简洁的讲解。然而，对于某些情况，在文献中几乎没有非常清晰的描述，而且至少需要具备大学数学水平才能理解。在这本书中，我尝试用直观的方式描述问题，为了尽可能地描述清楚而牺牲了一些严谨性(几乎所有其他文献都很严谨)。和正文描述相伴的软件是对数学的替代，可以提供算法的分步描述。

我彻底删除了本书第一版中的一些材料。书中再也没有关于小波的章节了，也没有遗传算法的章节。从另一方面说，本书新添了关于分类器的章节，我认为在第一版中这显然是一个遗漏。这一版中新添的重要的一章是关于利用并行计算解决图像处理问题的章，其中包括通过图形卡(GPU)加速计算高达 200 倍。还有全新的关于基于内容的搜索的第 11 章，基于内容的搜索指的是通过图像中的信息获取其他图像。就好像是说：“找出另一张与这张图片类似的图片”。基于内容的搜索会成为未来 20 年的重要技术。这项技术可以有效地使用现代大容量的磁盘驱动器；而且随着低成本高分辨率数码相机的普及，可以预见人们越来越需要在大量大图像(海量像素数)中搜索图像。

这一版中讨论的大部分算法的源码都可以在本书的网站上找到。关于阈值处理的那一章提供了 17 个程序，每一个程序实现了不同的阈值算法。现在在 Internet 上都可以找到细化程序、边缘检测程序以及形态学程序。

关于图像还原的一章仍是该主题实用信息的少有来源之一。符号识别的章节被更新了；然而，由于很多方法都是商用的，因此由于专利和版权的原因，不能描述这些方法，也不能提供相关的软件。不过，其基础内容都得到了描述，而且可以关联到与分类器有关的内容。

我认为利用并行计算处理视觉问题的那一章是本书的亮点。再次，利用可以下载到的工具，这一章展示了如何将网络上的所有计算机都连接起来组建成一个大型的图像处理集群。这一章还讲解了如何利用多核处理器上的所有 CPU；最重要的是，这一章还介绍了有关如何利用 GPU 编程的非常实用的信息，通过 GPU 可以进行图像处理和视觉任务处理，而不仅仅是显示图形。

最后，我还写了包含了一组精选的图像搜索方法的一章内容。这些方法都有对应的代码展示其算法的实现，再结合本书中的其他代码，您可以花很多时间实验自己的关于组织和搜索图像数据集的想法和算法。

读者可以在本书网站(http://www.wiley.com/go/jrparker)或者 http://www.tupwk.com.cn/downpage 上下载本书中提到的所有源代码和示例图像。您还可以链接到我的个人主页，在这上面我会添加新的代码和新的图像，甚至还会添加一些新写的材料补充和修订印刷的材料。通过这个网站还可以交流评论和错误，勘误表也会发布在这个网站上，另外读者贡献的软件和在其他系统和编译器上使用这些代码的新思想也会放在这个网站上。

我诚邀您通过网站向我建议您想读到的新的章节的主题。我会注意选择最热的请求，然后在未来在网站上发布那个主题的新章节。一本书，尽管主要通过纸张的形式发布，也不一定是完全静止的！

Jim Parker
加拿大艾伯塔省科克伦

目　录

第1章

视觉系统实践——图像显示、输入/输出和库函数调用

在利用视觉和图像分析系统进行实验或为了实际应用实现一个这种系统的时候，都需要一个基本的软件基础框架。图像由像素组成，在一幅来自于数码相机的典型图像中，约有400～600万个像素，每一个像素表示图像中那一个点的颜色。这样大量的数据都以一种适合于类似Photoshop和Paint这样的商业软件包操作的格式(例如GIF或JPEG)存储在文件中。开发一个新的图像分析软件意味着首先要能把这些文件读取为能直接访问像素值的内部格式。实现这些功能的代码并没有什么特别的地方，也不会涉及任何实际的图像处理，但是这是必不可少的第一步。同理，图像分析软件还需要将图像显示在屏幕上，并且能够以标准格式保存这些图像。有时候可能还需要使用图像捕捉的设施。所有这些操作都不会对图像进行修改，这些操作的作用只是按照必要的方式将图像在不同部分之间转移。

在图像程序中，这些枯燥的工作需要大量的代码。类似于把所有红色像素转换为黄色像素的代码可能只需要区区10行；但是，程序还是需要读取图像、显示图像并且将图像处理的结果输出，这些工作可能另外需要2000多行代码，甚至更多。

当然，这些基础设施代码(可以看做应用程序编程接口，即API)可以被所有的应用程序使用；因此，一旦开发好，就可以一直使用这些API了，而不需要做任何修改，除非需要进行更新。如果操作系统和底层库发生了变化，或者需要新的功能，就需要更新版本的API。如果处理得当，依赖于这些API的视觉程序只需要很小的修改甚至不需要修改。OpenCV系统就是这样一种API。

1.1 OpenCV

OpenCV最初由Intel开发。在本书写作的时候，其最新版本为2.0版，可以在http://sourceforge.

net/projects/opencvlibrary/下载。

尽管如此，2.0 版本相对来说较新，但依旧不能在所有主流系统上安装，也不能通过所有主流编译器的编译。本书的所有例子使用的都是 1.1 版(可以在 http://sourceforge.net/projects/opencvlibrary/files/opencv-win/1.1pre1/OpenCV_1.1pre1a.exe/下载)，并且通过 Microsoft Visual C++ 2008 Express Edition 的编译(在 www.microsoft.com/express/Downloads/#2008-Visual-CPP 下载)。

本书网站(www.wiley.com/go/jrparker)会维护并保存这些工具最新版本的链接。该网站讲解了如何安装编译器和 OpenCV。使用这种工具组合的好处在于，这些工具仍然非常流行，可用而且是免费的。

1.2 基本的 OpenCV 代码

OpenCV 是一个 C 函数库，既实现了基础设施的操作，也实现了图像处理和视觉函数。当然，开发者还可以混合加入自己编写的函数。因此，本书中描述的所有代码都可以在使用了 OpenCV 编程方式的程序中调用，这意味着本书中的方法都可以作为 OpenCV 中函数的补充使用。使用者只需要了解如何调用库函数，以及 OpenCV 使用的基本数据结构。

OpenCV 是一个大型的、复杂的库。为了帮助初学者开始使用这个库，下面这个简单的程序可以作为几乎所有程序的模板：

```
// basic.c : A 'wrapper' for basic vision programs.
#include "stdafx.h"
#include "cv.h"
#include "highgui.h"
int main (int argc, char* argv[])
{
    IplImage *image = 0;
    image = cvLoadImage( "C:\AIPCV\image1.jpg", 1 );
    if( image )
    {
        cvNamedWindow( "Input Image", 1 );
        cvShowImage( "Input Image", image );
        printf( "Press a key to exit\n");
        cvWaitKey(0);
        cvDestroyWindow( "String");
    }
    else
        fprintf( stderr, "Error reading image\n" );
    return 0;
}
```

这个程序和 Internet 上很多示例程序类似。这个程序读入一个图像(C:\AIPCV\image1.jpg 是表示图像文件路径名的字符串)，并且把这个图像显示在屏幕上的一个窗口中。当用户按下任意键后，这个程序销毁该显示窗口并且终止执行。

下面首先对数据结构和函数进行解释，这样才能正确地修改这段代码。

1.2.1　IplImage 数据结构

IplImage 数据结构是图像在内存内的数据组织形式。IplImage 形式的图像可以转换为像素数组的形式，不过 IplImage 还包含了很多和图像数据有关的结构信息，而图像数据可以有很多种形式。例如，一个从 GIF 文件读入的图像可能带有 256 个灰阶，每一个像素大小为 8 个比特。一个从 JPEG 文件读入的图像可能是每个像素 24 比特的彩色图像。这两种文件都可以表达为 IplImage。

IplImage 的基本结构和其他内部图像表达方式类似。必要的字段包括：

- Width——一个整数，保存了图像的宽度(以像素计)。
- Height——一个整数，保存了图像的高度(以像素计)。
- ImageData——指向字符数组的指针，每一个字符表示一个实际的像素或颜色值。

如果每一个像素为 1 字节，这样就能满足我们的需要了。然而，OpenCV 中的图像可以有很多种数据类型，例如：字节、整型、浮点型和双精度浮点型。可以是灰阶(1 字节)或 3 字节色彩(RGB)，还可以是 4 字节等。最后，有一些图像格式的原点在左上角(事实上大部分都是)，还有一些图像格式的原点在左下角(只有 Microsoft 是这样的)。

还有一些有用的字段包括：

- nChannels——一个整数，表示每一个像素的颜色数(1~4)。
- depth——一个整数，表示每一个像素的比特数。
- origin——表示坐标系统的原点。这是一个整数，0 表示左上角，1 表示左下角。
- widthStep——一个整数，表示图像一行所占的字节数。
- imageSize——一个整数，表示一个图像所占的字节数(= widthStep×height)。
- imageDataOrigin——指向图像原始数据(也称为根数据、基数据)的指针。
- roi——指向定义了正在处理的图像中兴趣区(region of interest)的结构体的指针。

当创建一个图像或从文件中读取一个图像的时候，会为这个图像创建一个 IplImage 实例，并且为相应的字段填上值。考虑以下定义：

```
IplImage* img = 0;
```

通过以下代码可以从一个文件读取一个图像(后面会详述)：

```
img = cvLoadImage(filename);
```

其中变量 filename 是保存了图像文件名称的字符串。如果运行成功，那么

```
img->imageData
```

指向的是保存像素的内存块。图 1-1 展示了用做示例的 JPEG 文件，名为 marchA062.jpg。

读入这个图像之后会创建一个特定类型的内部表达形式，这种表达形式适用于基本的 RGB 图像，这种形式可能是 IplImage 结构体在真实情形中最常见的一种形式了。这种表达形式中的每一个像素都表达为 3 个字节：一个表示红色，一个表示绿色，还有一个表示蓝色。字节的顺序为 b、g、r，从图像的第 1 行开始，按列步进，然后再显示下一行。因此 img->imageData 指向的数据按照以下顺序存储。

$b_{0,0}$ $g_{0,0}$ $r_{0,0}$ $b_{0,1}$ $g_{0,1}$ $r_{0,1}$ $b_{0,2}$ $g_{0,2}$ $r_{0,2}$ ...

这意味着第 1 行(行 0)中像素的 RGB 值表示为那一行所有像素的反序(b，g，r)。然后继续表示下一行，从列 0 开始，以此类推，直到最后一行。

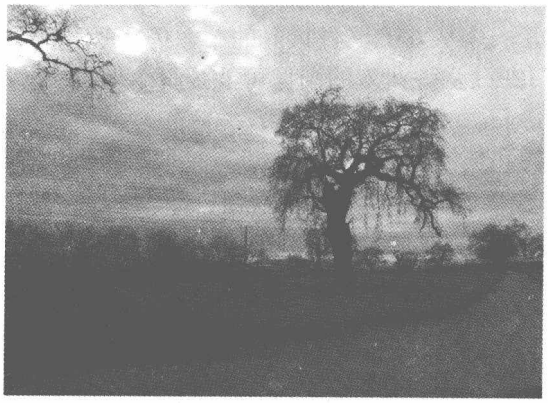

图 1-1 本章使用的示例数码图像。这是 Chico，CA 的一棵树的图像，由 HP Photosmart M637 相机拍摄。这一款相机是典型的现代中等质量的相机

如何访问其中的某一个像素？因为 widthStep 字段表示一行的大小，因此图像第 i 行的开始位置应该在：

```
img->imageData + i*img->widthStep
```

列 j 就位于这个位置之后的第 j 个像素处；如果像素为字节，那么这个像素的位置为：

```
img->imageData + i*img->widthStep + j
```

如果像素为 RGB 值(例如上面读出的 JPEG 图像)，那么每一个像素为 3 个字节长，因此第 j 个像素的位置为：

```
img->imageData + i*img->widthStep + j*3
```

nChannels 字段的值表示每个像素占用的字节数，因此某一个像素的位置可以归纳为：

```
img->imageData + i*img->widthStep))[j*img->nChannels]
```

最后，颜色分量的顺序为蓝、绿和红。因此，像素[i,j]的蓝色值为：

```
(img->imageData + i*img->widthStep)[j*img->nChannels + 0]
```

那么绿色和红色值分别为：

```
(img->imageData + i*img->widthStep)[j*img->nChannels + 1]
(img->imageData + i*img->widthStep)[j*img->nChannels + 2]
```

一个像素每个通道的数据类型为无符号字符型(或 uchar)。

有一种通用的方式可以访问图像中的像素，这种方式能够自动地根据图像类型和格式返回或修改指定的像素。这种方法非常方便，因为像素的类型可能是字节、RGB、浮点数