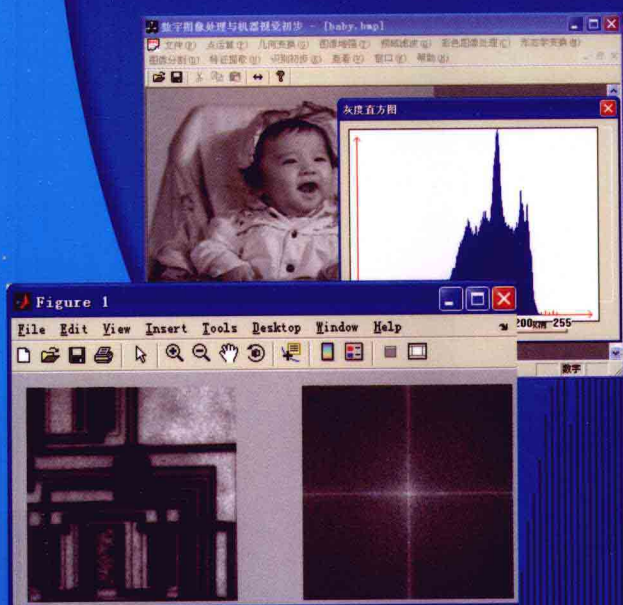


数字图像处理 与机器视觉

——Visual C++与Matlab实现

■ 张铮 王艳平 薛桂香 编著
白刚 顾泽苍 童欣 赵政 审



- 为想要了解数字图像处理与机器视觉领域的读者提供了一条扎实的进阶之路，从基本的图像处理算法，到投影校正、顶帽变换等高级应用，再到特征提取、分类器设计，以及像光学字符识别、人脸识别等综合案例，一步步地引导读者从阅读中获得知识，于实践中升华感悟。
- **Matlab**与**Visual C++**两种语言描述的无缝连接，体现出科学研究和工程实践在图像处理与机器视觉领域的完美结合。

- ✓ 各种标准图像处理算法 (Matlab/Visual C++)
- ✓ 汽车牌照的投影失真校正 (Matlab/Visual C++)
- ✓ 顶帽变换 (Matlab/Visual C++)
- ✓ 主成份分析 (Matlab)
- ✓ LBP特征提取 (Matlab)
- ✓ 基于人工神经网络的数字识别系统 (Visual C++)
- ✓ 基于支持向量机的人脸识别系统 (Matlab)



数字图像处理 与机器视觉

——Visual C++与Matlab实现

■ 张铮 王艳平 薛桂香 编著
白刚 顾泽苍 童欣 赵政 审

人民邮电出版社
北京

图书在版编目(CIP)数据

数字图像处理与机器视觉: Visual C++与Matlab实现 / 张铮, 王艳平, 薛桂香编著. — 北京: 人民邮电出版社, 2010.4

ISBN 978-7-115-22023-3

I. ①数… II. ①张… ②王… ③薛… III. ①C语言—程序设计②数字图像处理③计算机视觉 IV. ①TP312②TN911.73③TP302.7

中国版本图书馆CIP数据核字(2010)第003807号

内 容 提 要

本书将理论知识、科学研究和工程实践有机结合起来,介绍了数字图像处理和识别技术的方方面面,内容包括图像的点运算、几何变换、空域和频域滤波、图像复原、形态学处理、图像分割以及图像特征提取。本书还对于机器视觉进行了前导性的探究,重点介绍了两种在工程技术领域非常流行的分类技术——人工神经网络(ANN)和支持向量机(SVM),并在配套给出的识别案例中直击光学字符识别(OCR)和人脸识别两大热点问题。

全书结构紧凑,内容深入浅出,讲解图文并茂,适合于计算机、通信和自动化等相关专业的本科生、研究生以及工作在图像处理和识别领域一线的广大工程技术人员阅读。

数字图像处理与机器视觉——Visual C++与Matlab实现

- ◆ 编 著 张 铮 王艳平 薛桂香
审 白 刚 顾泽苍 童 欣 赵 政
责任编辑 刘 浩
- ◆ 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 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京铭成印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 34.75
字数: 853千字
印数: 1-3 500册

2010年4月第1版

2010年4月北京第1次印刷

ISBN 978-7-115-22023-3

定价: 69.00元(附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223
反盗版热线: (010)67171154

前 言

图像处理与机器视觉是当前计算机科学的一个热门研究方向，它应用广泛，发展前景乐观。近年来，伴随着人工智能、模式识别学科以及人机智能接口技术的飞速发展，机器视觉的研究正在不断升温——从日常生活中与我们息息相关的光学字符识别（OCR）和汽车自动驾驶，到医学应用中的病灶检测与分析，再到未来人机智能交互领域中的人脸识别，情感计算，等等。而图像处理是机器视觉研究中必不可少的图像预处理环节，是我们需要掌握的首要技术。

和图像相关的东西往往容易引起计算机初学者的兴趣，我在读大学本科的时候就觉得能让计算机理解所“看”到的东西是一件非常神秘和令人兴奋的事情；但同时它的理论性较强，门槛较高，在各个高校中，这门课程大多也是作为计算机专业研究生的选修课程。要理解该领域的知识，读者需要具有一定的数学基础，除此之外还涉及到信号处理、统计分析、模式识别和机器学习等专业领域知识，因此令很多人望而却步。

其实“难以理解”关键在于缺乏必要的先序知识，这才是难以跨越的鸿沟。我们在撰写本书过程中，对于可能造成理解困难的地方，均尽可能地给出了必要的先序知识，深入浅出，尽量定性地进行描述；对于那些并不一目了然的结论均给出了思路和解释，必要的还在附录中提供了证明；对于某些非常专业已经超过本书讨论范围的相关知识则在最后给出了参考文献，供有兴趣的读者进一步学习和研究。

本书的宗旨是在向读者介绍知识的同时，培养读者的思维方法，使读者知其然还要知其所以然，并在解决实际问题中能有自己的想法。

——内容安排

全书共分 14 章，具体内容安排如下。

第 0~2 章介绍了数字图像处理的基础知识和编程基础，使读者第一步能够建立起对于数字图像本质的正确认识，了解和掌握必要的术语和预备知识并且熟悉本书自始至终需要使用的两大工具 Matlab 和 Visual C++，掌握功能强大的 C++ 数字图像处理基类 CImg 及其派生类 CImgProcess 的框架。

第 3~4 章分别介绍了图像的灰度变换和几何变换。通过灰度变换可以有效改善图像的外观，并在一定程度上实现图像的灰度归一化；几何变换则主要应用在图像的几何归一化和图像校准当中。总体而言，这些内容大多是图像的前期预处理工作的一部分，是图像处理中相对固定和程式化的内容。

第 5~6 章分别从空间域和频率域两个角度考量图像增强的各个主要方面。图像增强作为数字图像处理中相对简单却最具艺术性的领域之一，可理解为根据特定的需要突出一幅图像中的某些信息，同时，削弱或去除某些不需要的信息的处理方法。其主要目的是使处理后的图像对某种特定的应用来说，比原始图像更适用。

第 7 章是本书中相对独立的一章，以介绍色彩模型之间的相互转换以及彩色图像处理方

面的基本概念和基本方法为主。随着基于互联网的图像处理应用的不断增长，彩色图像处理已经成为一个重要领域。

第 8~10 章（形态学处理、边缘检测与图像分割、特征提取）可以说是从单纯图像处理向图像识别（机器视觉）的过渡，这一阶段的特点是输入是图像，输出则是在识别意义上我们感兴趣的图像元素。形态学处理是提取图像元素的有力技术，它在表现和描述形状方面非常有用；分割过程则将一幅图像划分为组成部分或目标对象；研究特征提取则是要将前面提取出来的图像元素或目标对象表示为适合计算机后续处理的数值形式，最终形成能够直接供分类器使用的特征。

第 11 章在前面知识的基础之上，引出了机器视觉的前导性内容，给出了解决识别问题的一般思路。

最后两章（人工神经网络、支持向量机）介绍了两种十分强大的分类技术，并在手写数字字符识别和人脸识别这样的经典案例中结束本书。

——读者对象

- 大学二年级以上（具备必要的数学基础）的相关专业的本科生、研究生。
- 工作在图像处理和识别领域一线的工程技术人员。
- 对于数字图像处理和机器视觉感兴趣的并且具备必要预备知识的所有读者。

——在阅读本书之前，读者最好具有如下的预备知识

- 读者应该熟悉 C++ 编程语言和面相对象的基本思想。书中的相当一部分示例是以 C++ 语言描述的。
- 读者应具备一定的数学基础，主要的高等数学知识、少量的线性代数基本概念加上对于概率理论主要思想的理解（识别部分）。

——附书光盘和读者反馈

本书所有 Visual C++ 实例的源代码和大部分 Matlab 实例的源代码均可在随书附赠的光盘中找到，代码全部经 Visual C++ 6.0 编译通过或在 Matlab 中执行正确。虽然本书中的所有例子都已经在 Windows XP 和 Windows 2003 下测试通过，但由于许多算法比较复杂且笔者水平局限，也有存在 Bug 的可能，即便正确也可能存在更加优化的算法或更加合理的程序结构，如发现任何上述问题，请您不吝告知本书的作者（book_better@sina.com）以便我们做出改进。

——致谢

首先要感谢我的授业恩师南开大学的白刚教授和天津大学的赵政教授，是他们引导我进入了图像处理与机器视觉的研究领域。同时，他们在我写作过程中的指点和教诲确保了本书的权威性和严谨性。

感谢我的好友王艳平提供并调试了许多实例代码；感谢河北工业大学的薛桂香老师参与编写了本书中第 3、4、6、7、9 章的内容；感谢我的师弟和师妹——王宪、王玲、金洋、梁

燕飞、黄毛毛参与了部分章节的编写和修改；感谢罗小科先生为本书制作了很多插图；感谢我的兄长张钊为本书提供了部分照片；还要感谢朱毅、丁宗尧、刘群忠、贾万宏等为本书的编写提出了很多宝贵的意见和建议。

最后感谢我的妻子马宏及我的家人，没有你们的鼓励和支持就不会有我的这部作品。

作者

2010年1月

目 录

第 0 章 数字图像处理概述	1	1.2 Matlab 图像类型及其存储 方式	29
0.1 数字图像	1	1.3 Matlab 的图像转换	33
0.1.1 什么是数字图像	1	1.4 读取和写入图像文件	35
0.1.2 数字图像的显示	1	1.5 图像的显示	37
0.1.3 数字图像的分类	2	第 2 章 Visual C++ 图像处理编程基础	39
0.1.4 数字图像的实质	3	2.1 位图文件及其 C++ 操作	39
0.1.5 数字图像表示	4	2.1.1 设备无关位图	39
0.1.6 图像的空间和灰度级 分辨率	5	2.1.2 BMP 图像文件数据 结构	39
0.2 数字图像处理与识别	6	2.2 认识 CImg 类	42
0.2.1 从图像处理到图像识别	6	2.2.1 主要成员函数列表	42
0.2.2 什么是机器视觉	8	2.2.2 公有成员	43
0.2.3 数字图像处理和识别的 应用实例	8	2.3 CImg 类基础操作	44
0.3 数字图像处理的预备知识	8	2.3.1 加载和写入图像	44
0.3.1 邻接性、连通性、 区域和边界	8	2.3.2 获得图像基本信息	47
0.3.2 距离度量的几种方法	10	2.3.3 检验有效性	48
0.3.3 基本的图像操作	10	2.3.4 按像素操作	49
第 1 章 Matlab 图像处理编程基础	12	2.3.5 改变图像大小	50
1.1 Matlab 操作简介	12	2.3.6 重载的运算符	51
1.1.1 Matlab 软件环境	12	2.3.7 在屏幕上绘制位图图像	51
1.1.2 文件操作	13	2.3.8 新建图像	52
1.1.3 在线帮助的使用	14	2.3.9 图像类型的判断与转化	54
1.1.4 变量的使用	16	2.4 DIPDemo 工程	55
1.1.5 矩阵的使用	18	2.4.1 DIPDemo 主界面	55
1.1.6 细胞数组 (Cell Array) 和 结构体 (Structure)	21	2.4.2 图像操作和处理类—— CImg 和 CImgProcess	56
1.1.7 关系运算与逻辑运算	22	2.4.3 文档类—— CDIPDemoDoc	57
1.1.8 常用图像处理数学函数	22	2.4.4 视图类—— CDIPDemoView	58
1.1.9 Matlab 程序流程 控制	23	2.5 CImg 应用示例	58
1.1.10 M 文件编写	27	2.5.1 打开图像	58
1.1.11 Matlab 函数编写	28	2.5.2 清空图像	60

2.5.3	像素初始化方法	61	4.2.2	图像平移的实现	104
2.5.4	保存图像	61	4.3	图像镜像	106
第 3 章	图像的点运算	63	4.3.1	图像镜像的变换公式	106
3.1	灰度直方图	63	4.3.2	图像镜像的实现	106
3.1.1	理论基础	63	4.4	图像转置	109
3.1.2	Matlab 实现	64	4.4.1	图像转置的变换公式	110
3.1.3	Visual C++实现	67	4.4.2	图像转置的实现	110
3.2	灰度的线性变换	69	4.5	图像缩放	111
3.2.1	理论基础	69	4.5.1	图像缩放的变换公式	112
3.2.2	Matlab 实现	70	4.5.2	图像缩放的实现	112
3.2.3	Visual C++实现	72	4.6	图像旋转	114
3.3	灰度对数变换	73	4.6.1	以原点为中心的图像 旋转	114
3.3.1	理论基础	74	4.6.2	以任意点为中心的图像 旋转	115
3.3.2	Matlab 实现	74	4.6.3	图像旋转的实现	116
3.3.3	Visual C++实现	75	4.7	插值算法	118
3.4	伽玛变换	76	4.7.1	最近邻插值	118
3.4.1	理论基础	76	4.7.2	双线性插值及其 Visual C++实现	119
3.4.2	Matlab 实现	77	4.7.3	高阶插值	121
3.4.3	Visual C++实现	79	4.8	图像配准	124
3.5	灰度阈值变换	80	4.8.1	什么是图像配准	124
3.5.1	理论基础	80	4.8.2	人脸图像配准的 Matlab 实现	124
3.5.2	Matlab 实现	81	4.9	Visual C++实用案例——汽车 牌照的投影失真校正	128
3.5.3	Visual C++实现	82	4.9.1	系统分析与设计	129
3.6	分段线性变换	83	4.9.2	系统实现	130
3.6.1	理论基础	83	4.9.3	功能测试	136
3.6.2	Matlab 实现	84	第 5 章	空间域图像增强	140
3.6.3	Visual C++实现	88	5.1	图像增强基础	140
3.7	直方图均衡化	90	5.1.1	为什么要进行图像 增强	140
3.7.1	理论基础	90	5.1.2	图像增强的分类	140
3.7.2	Matlab 实现	91	5.2	空间域滤波	141
3.7.3	Visual C++实现	93	5.2.1	空间域滤波和邻域 处理	141
3.8	直方图规定化(匹配)	94	5.2.2	边界处理	142
3.8.1	理论基础	95			
3.8.2	Matlab 实现	95			
3.8.3	Visual C++实现	97			
第 4 章	图像的几何变换	101			
4.1	解决几何变换的一般思路	101			
4.2	图像平移	103			
4.2.1	图像平移的变换公式	103			

5.2.3	相关和卷积	143	6.3	快速傅立叶变换 (Fast Fourier Transform, FFT) 及实现	187
5.2.4	滤波操作的 Matlab 实现	143	6.3.1	FFT 变换的必要性	187
5.2.5	滤波操作的 Visual C++ 实现	146	6.3.2	常见的 FFT 算法	188
5.3	图像平滑	148	6.3.3	按时间抽取的基-2 FFT 算法	188
5.3.1	平均模板及其实现	148	6.3.4	离散反傅立叶变换的快速算法 (IFFT)	192
5.3.2	高斯平滑及其实现	151	6.3.5	N 维快速傅里叶变换 (FFT _N)	192
5.3.3	通用平滑滤波的 Visual C++ 实现	154	6.3.6	Matlab 实现	192
5.3.4	自适应平滑滤波	156	6.3.7	Visual C++ 实现	198
5.4	中值滤波	156	6.4	频域滤波基础	206
5.4.1	性能比较	156	6.4.1	频域滤波与空域滤波的关系	206
5.4.2	一种改进的中值滤波策略	161	6.4.2	频域滤波的基本步骤	206
5.4.3	中值滤波的工作原理	162	6.4.3	频域滤波的 Matlab 实现	207
5.5	图像锐化	162	6.4.4	频域滤波的 Visual C++ 实现	208
5.5.1	理论基础	162	6.5	频域低通滤波器	210
5.5.2	基于一阶导数的图像增强——梯度算子	163	6.5.1	理想低通滤波器及其实现	210
5.5.3	基于二阶微分的图像增强——拉普拉斯算子	167	6.5.2	高斯低通滤波器及其实现	215
5.5.4	基于一阶与二阶导数的锐化算子的比较	169	6.6	频率域高通滤波器	220
5.5.5	高提升滤波及其实现	171	6.6.1	高斯高通滤波器及其实现	220
5.5.6	高斯-拉普拉斯变换 (Laplacian of a Gaussian, LoG)	175	6.6.2	频域拉普拉斯滤波器及其实现	224
第 6 章	频率域图像增强	178	6.7	Matlab 综合案例——利用频域滤波消除周期噪声	227
6.1	频率域滤波——与空间域滤波殊途同归	178	6.7.1	频域带阻滤波器	227
6.2	傅立叶变换基础知识	178	6.7.2	带阻滤波消除周期噪声	229
6.2.1	傅立叶级数	179	6.8	频域滤波器与空域滤波器之间的内在联系	232
6.2.2	傅立叶变换	181	第 7 章	彩色图像处理	233
6.2.3	幅度谱、相位谱和功率谱	183	7.1	彩色基础	233
6.2.4	傅立叶变换的实质——基的转换	185			

7.1.1	什么是彩色	233	实现	316
7.1.2	我们眼中的彩色	234	8.3.8 bwmorph 函数	319
7.1.3	三原色	234	8.4 灰度图像中的基本形态学	
7.1.4	计算机中的颜色表示	235	运算	320
7.2	彩色模型	236	8.4.1 灰度膨胀及其实现	320
7.2.1	RGB 模型	236	8.4.2 灰度腐蚀及其实现	324
7.2.2	CMY、CMYK 模型	238	8.4.3 灰度开、闭运算及其实现	328
7.2.3	HSI 模型	241	8.4.4 顶帽变换 (top-hat) 及其实现	331
7.2.4	HSV 模型	249	8.5 小结	334
7.2.5	YUV 模型	254	第 9 章 图像分割	335
7.2.6	YIQ 模型	260	9.1 图像分割概述	335
7.2.7	Lab 模型简介	264	9.2 边缘检测	336
7.3	全彩色图像处理基础	264	9.2.1 边缘检测概述	336
7.3.1	彩色补偿及其 Matlab 实现	265	9.2.2 常用的边缘检测算子	337
7.3.2	彩色平衡及其 Matlab 实现	267	9.2.3 Matlab 实现	340
第 8 章 形态学图像处理		270	9.2.4 Visual C++实现	343
8.1	预备知识	270	9.3 霍夫变换	350
8.2	二值图像中的基本形态学运算	272	9.3.1 直线检测	350
8.2.1	腐蚀及其实现	273	9.3.2 曲线检测	352
8.2.2	膨胀及其实现	280	9.3.3 任意形状的检测	353
8.2.3	开运算及其实现	284	9.3.4 Hough 变换直线检测的 Matlab 实现	354
8.2.4	闭运算及其实现	287	9.3.5 Hough 变换直线检测的 Visual C++实现	357
8.3	二值图像中的形态学应用	289	9.4 阈值分割	361
8.3.1	击中与击中不中变换及其实现	289	9.4.1 阈值分割方法	361
8.3.2	边界提取与跟踪及其实现	291	9.4.2 Matlab 实现	364
8.3.3	区域填充及其 Visual C++实现	296	9.4.3 Visual C++实现	366
8.3.4	连通分量提取及其实现	299	9.5 区域分割	368
8.3.5	细化算法及其 Visual C++实现	305	9.5.1 区域生长及其实现	368
8.3.6	像素化算法及其 Visual C++实现	310	9.5.2 区域分裂与合并及其 Matlab 实现	373
8.3.7	凸壳及其 Visual C++实现	316	9.6 小结	378
			第 10 章 特征提取	379
			10.1 图像特征概述	379
			10.1.1 什么是图像特征	379
			10.1.2 图像特征的分类	379

10.1.3	特征向量及其几何解释	379	11.1.2	图像识别	422
10.1.4	特征提取的一般原则	380	11.1.3	关键概念	422
10.1.5	特征的评价标准	381	11.1.4	识别问题的一般描述	423
10.2	基本统计特征	381	11.1.5	过度拟合 (Overfit)	424
10.2.1	简单的区域描绘子及其 Matlab 实现	381	11.1.6	模式识别系统结构	425
10.2.2	直方图及其统计特征	383	11.1.7	训练/学习方法分类	425
10.2.3	灰度共现矩阵及其 Visual C++ 实现	385	11.2	模式识别方法分类	426
10.3	特征降维	388	11.2.1	统计模式识别	426
10.3.1	维度灾难	388	11.2.2	句法模式识别	426
10.3.2	特征选择简介	389	11.2.3	小结	427
10.3.3	主成份分析 (Principal Component Analysis, PCA)	390	11.3	最小距离分类器和模板匹配	428
10.3.4	快速 PCA 及其实现	397	11.3.1	最小距离分类器及其 Matlab 实现	428
10.4	综合案例——基于 PCA 的人脸特征抽取	399	11.3.2	基于相关的模板匹配	430
10.4.1	数据集简介	399	11.3.3	相关匹配的计算效率	436
10.4.2	生成样本矩阵	400	第 12 章	人工神经网络	438
10.4.3	主成份分析	401	12.1	人工神经网络简介	438
10.4.4	主成份脸可视化分析	402	12.1.1	仿生学动机	438
10.4.5	基于主分量的人脸重建	404	12.1.2	人工神经网络的应用实例	440
10.5	局部二进制模式	406	12.2	人工神经网络的理论基础	441
10.5.1	基本 LBP	406	12.2.1	训练线性单元的梯度下降算法	441
10.5.2	圆形邻域的 LBP_{PR} 算子	407	12.2.2	多层人工神经网络	447
10.5.3	统一化 LBP 算子——Uniform LBP 及其 Matlab 实现	407	12.2.3	sigmoid 单元	448
10.5.4	MB-LBP 及其 Matlab 实现	411	12.2.4	反向传播 (BP, Back Propagation) 算法	450
10.5.5	图像分区及其 Matlab 实现	417	12.2.5	训练中的问题	453
第 11 章	图像识别初步	421	12.3	基于 ANN 的数字字符识别系统 DigitRec——分析与设计	454
11.1	模式识别概述	421	12.3.1	任务描述	454
11.1.1	模式与模式识别	421	12.3.2	数据集简介	455
			12.3.3	设计要点	455
			12.4	基于 ANN 的数字字符识别系统 DigitRec——实现	457
			12.4.1	构建神经元结构——SNeuron	457
			12.4.2	构建神经网络网络层——	

SNeuronLayer.....	459	影响.....	506
12.4.3 神经网络信息头——		12.7.2 学习率的影响.....	508
NEURALNET_		12.7.3 训练时代数目的影响...	508
HEADER.....	460	第 13 章 支持向量机	511
12.4.4 神经网络类——		13.1 支持向量机的分类思想.....	511
CNeuralNet.....	460	13.1.1 分类模型的选择.....	511
12.4.5 神经网络的训练		13.1.2 模型参数的选择.....	512
数据类—CNeural		13.2 支持向量机的理论基础.....	512
Data.....	473	13.2.1 线性可分情况下的	
12.4.6 误差跟踪类——		SVM.....	512
CValueTrack.....	478	13.2.2 非线性可分情况下的	
12.4.7 训练对话框类——		C-SVM.....	516
CTrainDlg.....	481	13.2.3 需要核函数映射	
12.4.8 测试对话框类——		情况下的 SVM.....	518
CTestDlg.....	484	13.2.4 推广到多类问题.....	521
12.5 基于 ANN 的数字字符识别		13.3 SVM 的 Matlab 实现.....	523
系统 DigitRec——测试.....	487	13.3.1 训练——svmtrain.....	523
12.5.1 训练.....	488	13.3.2 分类——svmclassify...	525
12.5.2 测试.....	489	13.3.3 应用实例.....	526
12.6 改进的 DigitRec.....	490	13.4 综合案例——基于 PCA 和	
12.6.1 数字字符图像的预		SVM 的人脸识别系统.....	526
处理类——COCRImage		13.4.1 人脸识别简介.....	527
Process.....	490	13.4.2 前期处理.....	527
12.6.2 输入图像的预处理——		13.4.3 数据规格化	
实现.....	491	(Scaling).....	528
12.6.3 输入图像的预处理——		13.4.4 核函数的选择.....	531
测试.....	504	13.4.5 参数选择.....	532
12.7 神经网络参数对训练和		13.4.6 构建多类 SVM	
识别的影响.....	506	分类器.....	535
12.7.1 隐藏层单元数目的		13.4.7 实验结果.....	537

第 0 章 数字图像处理概述

图像是指能在人的视觉系统中产生视觉印象的客观对象，包括自然景物、拍摄到的图片、用数学方法描述的图形。图像的要素有几何要素（刻画对象的轮廓、形状等）和非几何要素（刻画对象的颜色、材质等）。

本章中，我们主要讲解数字图像的实质和数字图像处理的一般步骤，以及后面会经常使用到的基本概念。

0.1 数字图像

自然界中的图像都是模拟量，在计算机普遍应用之前，电视、电影、照相机等图像记录与传输设备都是使用模拟信号对图像进行处理。但是，计算机只能处理数字量，而不能直接处理模拟图像。所以我们要在使用计算机处理图像之前进行图像数字化。

0.1.1 什么是数字图像

简单地说，数字图像就是能够在计算机上显示和处理的图像，根据其特性可分为两大类——位图和矢量图。位图通常使用数字阵列来表示，常见格式有 BMP、JPG、GIF 等；矢量图由矢量数据库表示，我们接触最多的就是 PNG 图形。



提示：本书只涉及数字图像中位图图像的处理与识别，如无特别说明，后文提到的“图像”和“数字图像”都仅仅是指位图图像。一般而言，使用数字摄像机或数字照相机得到的图像都是位图图像。

我们可以将一幅图像视为一个二维函数 $f(x, y)$ ，其中 x 和 y 是空间坐标，而在 $x-y$ 平面中的任意一对空间坐标 (x, y) 上的幅值 f 称为该点图像的灰度、亮度或强度。此时，如果 f 、 x 、 y 均为非负有限离散，则称该图像为数字图像（位图）。

一个大小为 $M \times N$ 的数字图像是由 M 行 N 列的有限元素组成的，每个元素都有特定的位置和幅值，代表了其所在行列位置上的图像物理信息，如灰度和色彩等。这些元素称为图像元素或像素。

0.1.2 数字图像的显示

不论是 CRT 显示器还是 LCD 显示器，都是由许多点构成的，显示图像时这些点对应图像的像素，显示器被称为位映像设备。所谓位映像，就是一个二维的像素矩阵，而位图则采用位映像方法显示和存储的图像。当一幅数字图像被放大后就可以明显看出图像是由很多方格形状的像素构成的，如图 0.1 所示。

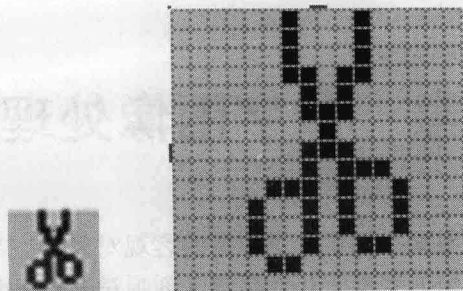


图 0.1 位图图像示例

0.1.3 数字图像的分类

根据每个像素所代表信息的不同, 可将图像分为二值图像、灰度图像、RGB 图像以及索引图像等。

1. 二值图像

每个像素只有黑、白两种颜色的图像称为二值图像。在二值图像中, 像素只有 0 和 1 两种取值, 一般用 0 来表示黑色, 用 1 表示白色。

2. 灰度图像

在二值图像中进一步加入许多介于黑色与白色之间的颜色深度, 就构成了灰度图像。这类图像通常显示为从最暗黑色到最亮的白色的灰度, 每种灰度(颜色深度)称为一个灰度级, 通常用 L 表示。在灰度图像中, 像素可以取 $0 \sim L-1$ 之间的整数, 根据保存灰度数值所使用的数据类型不同, 可能有 256 种取值或者 2^k 种取值, 当 $k=1$ 时即退化为二值图像。

3. RGB 图像

众所周知, 自然界中几乎所有颜色都可以由红 (Red, R)、绿 (Green, G)、蓝 (Blue, B) 3 种颜色组合而成, 通常称它们为 RGB 三原色。计算机显示彩色图像时采用最多的就是 RGB 模型。对于每个像素, 通过控制 R、G、B 三原色的合成比例则可决定该像素的最终显示颜色。

对于三原色 RGB 中的每一种颜色, 可以像灰度图那样使用 L 个等级来表示含有这种颜色成份的多少。例如对于含有 256 个等级的红色, 0 表示不含红色成份, 255 表示含有 100% 的红色成份。同样, 绿色和蓝色也可以划分为 256 个等级。这样每种原色可以用 8 位二进制数据表示, 于是 3 原色总共需要 24 位二进制数, 这样能够表示出的颜色种类数目为 $256 \times 256 \times 256 = 2^{24}$, 大约有 1600 万种, 已经远远超过普通人所能分辨出的颜色数目。

RGB 颜色代码可以使用十六进制数以减少书写长度, 按照两位一组的方式依次书写 R、G、B 三种颜色的级别。例如: 0xFF0000 代表纯红色, 0x00FF00 代表纯绿色, 而 0x00FFFF 是青色(这是绿色和蓝色的加和)。当 RGB 三种颜色的浓度一致时, 所表示的颜色就退化为灰度, 比如 0x808080 就是 50% 的灰色, 0x000000 为黑色, 而 0xFFFFFFFF 为白色。常见颜色的 RGB 组合值如表 0.1 所示。

表 0.1 常见颜色的 RGB 组合值

颜 色	R	G	B
红 (0xFF0000)	255	0	0
蓝 (0x00FF00)	0	255	0
绿 (0x0000FF)	0	0	255
黄 (0xFFFF00)	255	255	0
紫 (0xFF00FF)	255	0	255
青 (0x00FFFF)	0	255	255
白 (0xFFFFFFFF)	255	255	255
黑 (0x000000)	0	0	0
灰 (0x808080)	128	128	128

未经压缩的原始 BMP 文件就是使用 RGB 标准给出的 3 个数值来存储图像数据的,称为 RGB 图像。在 RGB 图像中每个像素都是用 24 位二进制数表示,故也称为 24 位真彩色图像。

4. 索引图像

如果对每个像素都直接使用 24 位二进制数表示,图像文件的体积将变得十分庞大。举个例子,对一个长、宽各为 200 像素,颜色数为 16 的彩色图像,每个像素都用 RGB 3 个分量表示,这样每个像素由 3 个字节表示,整个图像就是 $200 \times 200 \times 3 = 120\text{kB}$ 。这种完全未经压缩的表示方式,浪费了大量的存储空间,下面简单介绍另一种更节省空间的存储方式:索引图像。

同样还是 200 像素 \times 200 像素的 16 色图像,由于这张图片中最多只有 16 种颜色,那么可以用一张颜色表 (16 \times 3 的二维数组) 保存这 16 种颜色对应的 RGB 值,在表示图像的矩阵中使用这 16 种颜色在颜色表中的索引 (偏移量) 作为数据写入相应的行列位置。例如,颜色表中第 3 个元素为 0xAA1111,那么在图像中所有颜色为 0xAA1111 的像素均可以由 3-1=2 表示 (颜色表索引下标从 0 开始)。如此,每一个像素需要使用的二进制数就仅仅为 4 位 (0.5 字节),整个图像只需要 $200 \times 200 \times 0.5 = 20\text{kB}$ 就可以存储,而不会影响显示质量。

上文所指的颜色表就是常说的调色板 (Palette),另一种说法叫做颜色查找表 (LUT, Look Up Table)。Windows 位图中就应用调色板技术。其实不仅是 Windows 位图,许多其他的图像文件格式比如 PCX, TIF, GIF 都应用了这种技术。

在实际应用中,调色板中通常不足 256 种颜色。在使用许多图像编辑工具生成图像或者编辑 GIF 文件时,系统常常会提示用户选择文件包含的颜色数目。当选择较低的颜色数目时,会有效地降低图像文件的体积,在一定程度上也会降低图像的质量。

使用调色板技术减小图像文件体积的条件是图像的像素数目相对较多,而颜色种类相对较少。如果一个图像中用到了全部的 24 位真彩色,则对其使用颜色查找表技术完全没有意义,单纯从颜色角度对其进行压缩是不可能的。

0.1.4 数字图像的实质

实际上,0.1.1 小节中对于数字图像 $f(x, y)$ 的定义仅适用于最为一般的情况,即静态的灰度图像。更严格地说,数字图像可以是 2 个变量 (对于静止图像, static image) 或 3 个变量

(对于动态画面, video sequence) 的离散函数。在静态图像的情况下是 $f(x, y)$, 但在动态画面下, 则还需要时间参数 t , 即 $f(x, y, t)$; 函数值可能是一个数值 (对于灰度图像), 也可能是一个向量 (对于彩色图像)。



提示: 静态的灰度图像是本书研究的主要对象, 对于函数值为向量的情况会在第 7 章彩色图像处理中阐述, 而有关动态图像处理的内容请参见第 9 章。

图像处理是一个涉及诸多研究领域的交叉学科, 下面就让我们从不同的角度来审视数字图像:

- 从线性代数和矩阵论的角度, 数字图像就是一个由图像信息组成的二维矩阵, 矩阵的每个元素代表对应位置上的图像亮度和/或色彩信息。当然, 这个二维矩阵在数据表示和存储上可能不是二维的, 因为每个单位位置的图像信息可能需要不只一个数值来表示, 这样可能需要一个三维矩阵来对其进行表示 (参见 1.2 节关于 Matlab 中 RGB 图像表示的介绍)。
- 由于随机变化和噪声的原因, 图像在本质上是统计性的。因而有时将图像函数作为随机过程的实现来观察其存在的优越性。这时有关图像信息量和冗余的问题可以用概率分布和相关函数来描述和考虑。例如, 如果知道概率分布, 可以用熵 (Entropy) H^1 来度量图像的信息量, 这是信息论中一个重要的思想。
- 从线性系统的角度考虑, 图像及其处理也可以表示为用狄拉克冲激公式表达的点展开函数的叠加, 在使用这种方式对图像进行表示时, 可以采用成熟的线性系统理论研究。大多数时候, 我们都考虑使用线性系统近似的方式对图像进行近似处理以简化算法。虽然实际的图像并不是线性的, 图像坐标和图像函数的取值都是有限的和非连续的。

0.1.5 数字图像的表达

为了表述像素之间的相对和绝对位置, 通常还需要对像素的位置进行坐标约定。本书中所使用的坐标约定如图 0.2 所示。但在 Matlab 中坐标的约定会有变化, 具体请参见 1.1.5 小节。

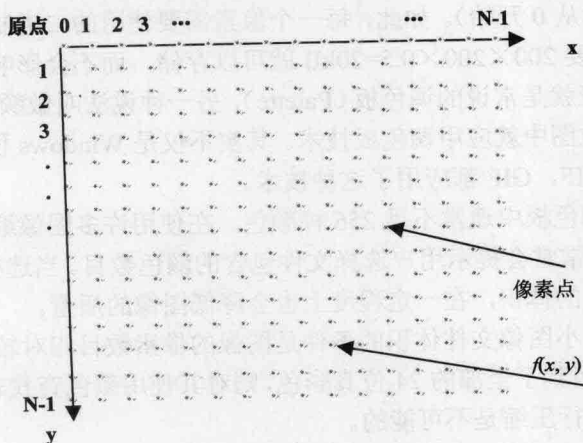


图 0.2 数字图像的坐标约定

¹ 熵 Entropy: 熵是信息论中用于度量信息量的一个概念。一个系统越是有序, 信息熵就越低; 反之, 一个系统越是混乱, 信息熵就越高。所以, 信息熵也可以说是系统有序化程度的一个度量。

在这之后，一幅物理图像就被转化成了数字矩阵，从而成为计算机能够处理的对象。数字图像 f 的矩阵表示如下所示：

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

有时也可以使用传统矩阵表示法来表示数字图像和像素：

$$A = \begin{bmatrix} a_{0,0} & \cdots & a_{0,N-1} \\ \vdots & \ddots & \vdots \\ a_{M-1,0} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (0-2)$$

其中，行列 (M 行 N 列) 必须为正整数，离散灰度级数目 L 一般为 2 的 k 次幂， k 为整数 (因为使用二进制整数表示灰度值)，图像的动态范围为 $[0, L-1]$ ，那么图像存储所需的比特数为 $b = M \times N \times k$ 。需要注意的是在矩阵 $f(y, x)$ 中，一般采用先行下标、后列下标的表示方法，因此这里先是纵坐标 y (对应行)，然后才是横坐标 x (对应列)。

而有些图像矩阵中，很多像素的值都是相同的。例如一个在纯黑背景上使用不同灰度勾勒的图像，其多数像素的值都会是 0。这种矩阵称为稀疏矩阵 (Sparse Matrix)，可以通过简单描述非零元素的值和位置来代替大量地写入 0 元素。这时存储图像需要的比特数可能大大减少。

0.1.6 图像的空间和灰度级分辨率

1. 图像的空间分辨率 (Spatial Resolution)

图像的空间分辨率是指图像中每单位长度所包含的像素或点的数目，常以像素/英寸 (pixels per inch, ppi) 为单位来表示。如 72ppi 表示图像中每英寸包含 72 个像素或点。分辨率越高，图像越清晰，图像文件所需的磁盘空间也越大，编辑和处理所需的时间也越长。

像素越小，单位长度所包含的像素数据就越多，分辨率也就越高，但同样物理大小范围内所对应图像的尺寸也会越大，存储图像所需要的字节数也越多。因而，在图像的放大缩小算法中，放大是对图像的过采样，缩小则是对图像的欠采样，这些内容会在“4.5 节图像缩放”中进一步介绍。

通常在没有必要对涉及像素的物理分辨率进行实际度量时，通常会称一幅大小为 $M \times N$ 的数字图像的空间分辨率为 $M \times N$ 像素。

图 0.3 给出了同一幅图像在不同的空间分辨率下呈现出的不同效果。当高分辨率下的图像以低分辨率表示时，在同等的显示或打印输出条件下，图像的尺寸会变小，细节变得不明显；当将低分辨率下的图像放大时，则会导致图像的细节仍然模糊，只是尺寸变大。这是因为缩小的图像已经丢失了大量的信息，在放大图像时只能通过复制行列的插值方法来确定新增像素的取值。