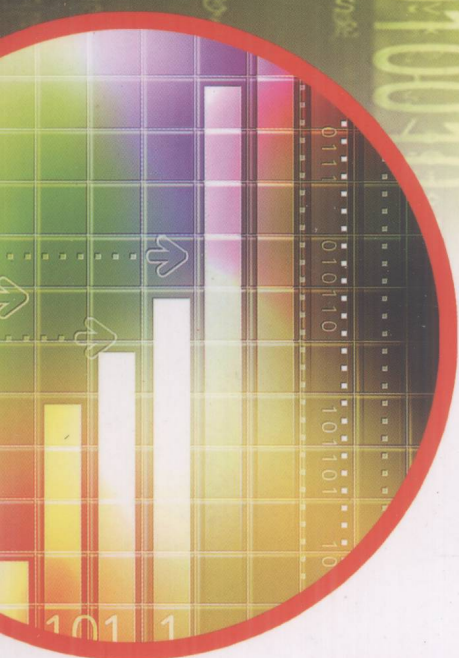




SHUZI TUXIANG
CHULI JISHU

数字图像处理技术

编著 穆晓芳 赵月爱
邓红霞 刘耀军

煤炭工业出版社

山西省青年科技研究基金项目资助(2008021025)

数字图像处理技术

穆晓芳 赵月爱 邓红霞 刘耀军 编著

煤炭工业出版社

·北京·

图书在版编目 (CIP) 数据

数字图像处理技术/穆晓芳等编著.—北京: 煤炭工业出版社, 2009

ISBN 978-7-5020-3548-8

I. 数… II. 穆… III. 数字图像处理 IV. TN911.73

中国版本图书馆 CIP 数据核字 (2009) 第 079555 号

煤炭工业出版社 出版
(北京市朝阳区芍药居 35 号 100029)

网址: www.cciph.com.cn

北京房山宏伟印刷厂 印刷

新华书店北京发行所 发行

*

开本 787mm×1092mm^{1/16} 印张 29^{1/4}

字数 692 千字 印数 1—500

2009 年 6 月第 1 版 2009 年 6 月第 1 次印刷

社内编号 6358 定价 60.00 元

版权所有 违者必究

本书如有缺页、倒页、脱页等质量问题, 本社负责调换

内 容 提 要

本书全面系统地介绍了数字图像处理的基本概念、基本理论、编程方法和应用实例。

全书共分 12 章，主要内容包括：数字图像处理概述、Visual C++ 数字图像编程基础、图像的几何变换、图像的点运算、图像的正交变换、图像增强、图像复原、图像边缘检测与边界跟踪、图像分割与匹配、图像数学形态学运算、MATLAB 图像处理简介、图像处理新技术等。

为了方便读者掌握图像处理技术，本书另配有实例光盘。

本书文字流畅，通俗易懂，是图像处理和多媒体编程的实用参考书，可作为高等院校计算机工程、信息工程、生物医学工程及其他相关专业高年级本科生和研究生的教学参考用书，也可供从事图像处理研究和开发的科技人员参考。

前 言

21 世纪是科学技术高速发展的时代, 图像作为现代信息社会中最基本的信息之一得到了广泛的应用。数字图像处理这门信息技术中的新兴学科, 几乎在各个行业里都起到了重要作用。它已成为计算机科学、信息科学、生物科学、空间科学、气象学、工程科学、统计学、物理科学、医学等许多学科的重要基础, 并正在向传统的学科甚至社会科学等领域不断地渗透。数字图像处理同时也是一种具有巨大的社会效益和经济效益的实用技术, 在军事、工业、科研、医学等领域获得了广泛运用。如今, 该学科的运用范围和规模又有了进一步延伸, 特别是在依据“视觉”感知事物的领域, 如遥感、气象、视频特技、体视技术、工业检测等领域。随着因特网的广泛应用和全球数字化的不断逼近, 数字图像处理这个交叉学科必将得到更加充分的发展, 并在现代信息社会中占据更加重要的地位。

数字图像处理既是一门实用的学科, 又需要一定的理论基础支持。目前数字图像处理的书籍可分为两大类: 一类是纯学术理论性的, 主要侧重于理论推导和分析, 而没有包含实际图像处理技术编程的实现; 另一类是数字图像处理技术的编程实现的讲解, 其内容也包含了必要的理论支持。本书属于第二种类型, 是实用型的数字图像处理技术基础教材, 专门介绍数字图像处理技术及其编程实现, 以及当前较为流行的图像处理新技术。

本书实用性较强, 读者利用本书可以轻松地掌握该领域的基本概念和基本编程方法以及数字图像处理技术发展的趋势和新方法。本书在介绍图像处理基本知识的同时, 着重介绍了相关算法的分析、描述了算法实现步骤和技巧, 并给出实现算法的简洁代码, 使读者能很快掌握编程的核心。针对每一种图像处理技术, 清晰地分为理论基础、编程实现两部分。读者在掌握了某种图像处理技术的基本理论之后, 按照编程实现步骤的指导, 可以很容易地了解算法的实现思路和方法, 再进一步体会核心代码, 可以很快掌握该图像处理技术, 从而实现从理论和实用的编程技巧两方面轻松掌握数字图像处理。

本书首先介绍了数字图像处理的基本概念和 Visual C++ 数字图像编程基础, 在此基础上分别介绍了图像几何变换、图像点运算、图像正交变换、图像增强、图像复原、图像边缘检测与边界跟踪、图像分割与匹配、图像数学形态学运算

等具体图像处理相关内容。之后对 MATLAB 图像处理进行了简单介绍,并对当前数字图像处理中 3 种常用的新技术——小波分析、神经网络、遗传算法,进行了相应讲解。

由于 Visual C++ 具有面向对象的编程特点,其编写得到的程序执行速度快、可移植能力强,使用较为广泛。由美国 Math Works 公司推出的可视化科学计算软件 MATLAB,近年来也得到了广泛的应用。MATLAB 工具箱对相关学科和各种基本技术都采用了当今最先进的算法,有极强的图形和图像处理功能,其语法结构简单易学,可大大节省编写低层算法代码的时间,避免程序设计中的重复劳动, MATLAB 的这些特点使其在数字图像处理的开发和研究中起到了重要作用。本书的程序大多都是由 Visual C++ 编程实现,书中提供了详细使用的编程代码。同时本书也简单介绍了 MATLAB 图像编程相关知识,并针对相关知识点和工程实例,提供了相应的 MATLAB 调试代码,以便指导读者在实际应用中,学会利用 MATLAB 图像处理工具快速、方便地调试算法,缩短开发周期。

本书基本上采用了一个图像处理算法对应一个编程实例的编写方法,使读者能够快速查到相应图像处理技术的编程代码。读者在学习过程中可以通过编程代码来加深理解本书内容,快速掌握图像处理编程的方法,为本领域的研究打下坚实的基础,提高独立承担科研项目开发的能力。

本书由山西省青年科技研究基金项目(2008021025)资助。

本书由穆晓芳、赵月爱、邓红霞、刘耀军共同编著,其中穆晓芳负责编写第 8 章、第 9 章和第 12 章,赵月爱负责编写第 2 章、第 6 章和第 7 章,邓红霞负责编写第 3 章、第 4 章、第 5 章和第 11 章,刘耀军负责编写第 1 章和第 10 章。

本书编写过程中参考了国内外有关数字图像处理的书刊及文献资料,在此对书刊及文献的作者表示衷心的感谢!

由于作者水平有限,书中难免会有缺陷和不足之处,敬请广大读者批评和指正。

编 者

2009 年 6 月

目 次

1 数字图像处理概述	1
1.1 数字图像处理相关概念	1
1.1.1 数字图像	1
1.1.2 图像处理	4
1.1.3 图像识别和解释	5
1.2 数字图像的存储	6
1.2.1 位映射图像	6
1.2.2 矢量图像	7
1.3 BMP 图像文件格式	7
1.3.1 位图文件头	8
1.3.2 位图信息头	8
1.3.3 调色板	9
1.3.4 实际的位图数据	10
1.4 TIFF 图像文件格式	10
1.4.1 TIFF 图像文件头	10
1.4.2 TIFF 标识信息区目录	11
1.4.3 TIFF 图像数据	12
1.4.4 TIFF 图像文件分类及数据压缩方法	13
1.5 JPEG 图像文件格式	14
1.5.1 JPEG 编码过程	14
1.5.2 JPEG 文件结构	15
本章小结	17
2 Visual C++ 数字图像编程基础	18
2.1 设备相关位图 (DDB)	18
2.1.1 DDB 概述	18
2.1.2 显示 DDB	19
2.2 设备无关位图 (DIB)	21
2.2.1 DIB 的结构	21
2.2.2 Win32 SDK 中的 DIB 访问函数	22
2.3 构造 CDIB 类	25
2.3.1 创建 CDIB 类的头文件	26

2.3.2 构造和析构函数	27
2.3.3 获取 DIB 属性函数	28
2.3.4 DIB 调色板操作函数	31
2.3.5 DIB 文件操作与数据串行化函数	33
2.3.6 DIB 显示函数	35
2.4 基于 CDIB 类读写 BMP 图像文件示例程序	36
2.4.1 创建工程	36
2.4.2 协调调色板	37
2.4.3 设置程序启动的最初参数	38
2.4.4 文档类中的函数	38
2.4.5 视图类中的函数	42
本章小结	44
3 图像的几何变换	45
3.1 图像几何变换的基础知识	45
3.1.1 几何变换的概念	45
3.1.2 几种常见几何变换	46
3.2 图像平移	48
3.2.1 算法剖析	48
3.2.2 编程实现	49
3.3 图像旋转	53
3.3.1 算法剖析	53
3.3.2 编程实现	56
3.4 图像镜像	61
3.4.1 算法剖析	61
3.4.2 编程实现	62
3.5 图像转置	66
3.5.1 算法剖析	66
3.5.2 编程实现	67
3.6 图像缩放	70
3.6.1 算法剖析	70
3.6.2 编程实现	71
3.7 插值算法	75
3.7.1 最近邻插值	75
3.7.2 双线性插值	76
3.7.3 高阶插值	81
本章小结	85

4 图像的点运算	86
4.1 图像点运算概述	86
4.2 灰度直方图	86
4.2.1 灰度直方图的定义	87
4.2.2 编程绘制灰度直方图	87
4.3 图像灰度阈值变换	94
4.3.1 算法剖析	94
4.3.2 编程实现	95
4.4 图像灰度窗口变换	97
4.4.1 算法剖析	97
4.4.2 编程实现	98
4.5 图像灰度线性变换	100
4.5.1 算法剖析	100
4.5.2 编程实现	101
4.6 图像灰度分段线性变换	104
4.6.1 算法剖析	104
4.6.2 编程实现	105
4.7 图像直方图均衡	108
4.7.1 算法剖析	108
4.7.2 编程实现	110
本章小结	114
5 图像的正交变换	115
5.1 正交变换的概念	115
5.1.1 连续函数的正交性	115
5.1.2 二维离散正交变换	116
5.2 傅里叶变换	117
5.2.1 连续傅里叶变换的基本概念	117
5.2.2 离散傅里叶变换的基本概念	117
5.2.3 离散傅里叶变换的性质	118
5.2.4 快速傅里叶变换	122
5.2.5 编程实现图像的快速傅里叶变换	126
5.3 离散余弦变换	132
5.3.1 一维离散余弦变换的基本概念	132
5.3.2 二维离散余弦变换的基本概念	133
5.3.3 快速离散余弦变换	134
5.3.4 编程实现图像的快速离散余弦变换	135
5.4 离散沃尔什-哈达玛变换	140

5.4.1 沃尔什函数	140
5.4.2 离散沃尔什-哈达玛变换	142
5.4.3 快速沃尔什-哈达玛变换	144
5.4.4 编程实现图像的快速沃尔什-哈达玛变换	145
5.5 离散 K-L 变换	150
5.5.1 离散 K-L 变换的基本概念	151
5.5.2 编程实现图像的离散 K-L 变换	152
本章小结	163
6 图像增强	164
6.1 图像灰度变换增强	165
6.2 图像方差均衡	165
6.2.1 方差均衡的基本概念	165
6.2.2 编程实现图像的方差均衡	166
6.3 图像平滑	170
6.3.1 图像平滑的基本概念	170
6.3.2 邻域平均法	170
6.3.3 模板操作	171
6.3.4 编程实现图像平滑	174
6.4 图像的中值滤波	183
6.4.1 中值滤波的基本概念	183
6.4.2 编程实现图像的中值滤波	185
6.5 图像锐化	190
6.5.1 梯度锐化的基本概念	190
6.5.2 编程实现图像的梯度锐化	192
6.5.3 拉普拉斯锐化的基本概念	195
6.5.4 编程实现图像的拉普拉斯锐化	196
6.6 低通滤波	197
6.6.1 低通滤波的基本概念	197
6.6.2 编程实现图像的低通滤波	199
6.7 高通滤波	208
6.7.1 高通滤波的基本概念	208
6.7.2 编程实现图像的高通滤波	209
本章小结	217
7 图像复原	219
7.1 图像复原概述	219
7.1.1 图像退化连续函数的数学模型	219
7.1.2 图像退化离散函数的数学模型	221

7.2 图像的退化	222
7.2.1 图像的卷积模糊	222
7.2.2 编程实现图像的卷积模糊	223
7.2.3 图像的运动模糊	227
7.2.4 编程实现图像的运动模糊	228
7.2.5 图像的噪声	232
7.2.6 编程实现图像的噪声	233
7.3 图像逆滤波复原	237
7.3.1 逆滤波复原的基本概念	237
7.3.2 编程实现图像的逆滤波复原	238
7.4 图像的维纳滤波复原	244
7.4.1 维纳滤波复原的基本概念	244
7.4.2 编程实现图像的维纳滤波复原	247
7.5 图像的运动模糊复原	252
7.5.1 运动模糊复原的基本概念	253
7.5.2 编程实现图像运动模糊复原	254
7.6 其他图像复原方法	259
7.6.1 最大后验复原	259
7.6.2 最大熵复原	260
7.6.3 图像盲复原	262
7.6.4 图像几何畸变校正	264
本章小结	265
8 图像边缘检测与边界跟踪	266
8.1 梯度算子边缘检测	266
8.1.1 图像梯度的基本概念	266
8.1.2 边缘检测梯度算子	267
8.1.3 编程实现图像的梯度算子边缘检测	269
8.2 Kirsch 算子边缘检测	279
8.2.1 Kirsch 算子的基本概念	279
8.2.2 编程实现图像的 Kirsch 算子边缘检测	280
8.3 Laplacian-Gauss 算子边缘检测	289
8.3.1 Laplacian-Gauss 算子的基本概念	289
8.3.2 编程实现图像的 Laplacian-Gauss 算子边缘检测	289
8.4 Canny 算法边缘检测	292
8.4.1 Canny 算法的基本概念	292
8.4.2 编程实现图像的 Canny 算法边缘检测	294
8.5 哈夫变换	305
8.5.1 哈夫变换的基本概念	305

8.5.2 编程实现图像的哈夫变换目标检测	307
8.6 图像灰度值探测法边界跟踪	316
8.6.1 灰度值探测法的基本概念	316
8.6.2 编程实现图像的灰度值探测法边界跟踪	317
8.7 图像梯度值探测法边界跟踪	322
8.7.1 梯度值探测法的基本概念	322
8.7.2 编程实现图像的梯度值探测法边界跟踪	323
本章小结	327
9 图像分割与匹配	329
9.1 阈值分割法	330
9.1.1 阈值分割法的基本概念	330
9.1.2 编程实现图像的阈值分割	331
9.2 特征空间聚类法	337
9.2.1 特征空间聚类法的基本概念	337
9.2.2 编程实现特征空间聚类法图像分割	338
9.3 区域生长法	343
9.3.1 区域生长法的基本概念	343
9.3.2 编程实现区域生长法图像分割	344
9.4 图像匹配	348
9.4.1 模板匹配的基本概念	348
9.4.2 编程实现图像的模板匹配	350
本章小结	365
10 图像数学形态学运算	366
10.1 数学形态学概述	366
10.2 图像腐蚀	368
10.2.1 图像腐蚀的基本概念	368
10.2.2 编程实现图像腐蚀	370
10.3 图像膨胀	379
10.3.1 图像膨胀的基本概念	379
10.3.2 编程实现图像膨胀	381
10.4 图像开运算与闭运算	390
10.4.1 开运算与闭运算的基本概念	390
10.4.2 编程实现图像开运算和闭运算	391
10.5 图像的数学形态学边缘检测	395
10.5.1 数学形态学边缘检测的基本概念	395
10.5.2 编程实现图像的数学形态学边缘检测	396
本章小结	399

11 MATLAB 图像处理简介	400
11.1 MATLAB 概述	400
11.2 MATLAB 编程基础	401
11.2.1 MATLAB 数据结构	401
11.2.2 MATLAB 的基本运算	404
11.2.3 MATLAB 的程序结构与程序流控制	406
11.2.4 MATLAB 的数据接口与文件操作	409
11.2.5 MATLAB 的函数与 M 文件	411
11.3 MATLAB 图像处理简单实例	414
11.3.1 读写和显示图像文件	414
11.3.2 图像增强	417
本章小结	422
12 图像处理新技术	423
12.1 小波分析	423
12.1.1 连续小波变换 (CWT)	424
12.1.2 离散小波变换 (DWT)	427
12.1.3 小波分析在图像处理中的应用	430
12.2 神经网络	433
12.2.1 神经网络的基本概念	433
12.2.2 神经网络在图像处理中的应用	440
12.3 遗传算法	441
12.3.1 遗传算法的基本概念	441
12.3.2 遗传算法在图像处理中的应用	446
本章小结	449
附录 MATLAB 图像处理工具箱函数	450
参考文献	453

1 数字图像处理概述

1.1 数字图像处理相关概念

数字图像处理 (Digital Image Processing) 是通过计算机对图像进行去除噪声、增强、复原、分割、提取特征等处理的方法和技术。数字图像处理技术起源于 20 世纪 20 年代, 随着计算机科学技术的发展, 数字图像处理技术近几年来得到了长足的发展, 并广泛应用于科学研究和生产实践的各个领域。以下介绍数字图像处理相关的基础概念。

1.1.1 数字图像

数字图像 (Digital Image) 在信息时代的今天, 与人们的生活和工作已紧密相连。其涉及面非常广, 影响到了科学技术和社会的各个方面。在通信技术、宇宙探索、遥感技术、生物医学、工业生产、军事技术、侦缉破案、考古等许多领域, 数字图像都起着不可替代的作用。

1. 图像数字化

人眼看到的任何自然界的图像都是连续的模拟图像, 其形状和形态表现由图像各位置的颜色所决定。把连续空间的图像在坐标空间 (x, y) 和性质空间 F 都离散化, 便于计算机进行加工处理, 这种离散化的图像就是数字图像。数字图像是数字图像处理的主要研究对象。

将连续的图像函数必须转化为离散的数据集的过程叫做图像采集。图像采集由图像采集系统完成, 如图 1-1 所示。

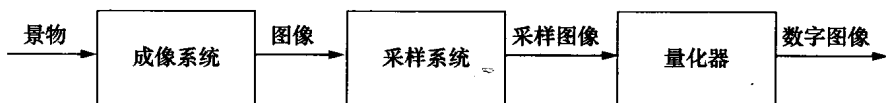


图 1-1 图像采集系统

图像采集系统包括 3 个基本单元, 即成像系统、采样系统和量化器。采样实际上就是一个空间坐标的量化过程, 量化则是对图像函数值的离散化过程。采样和量化系统统称为数字化。

数字图像是连续图像的一种近似表示, 通常由采样点的值所组成的矩阵来表示:

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

矩阵中每个采样点叫做一个像素 (pixel), M, N 分别为数字图像在横、纵方向上的像素数。

把像素按不同的方式进行组织或存储,就得到不同的图像格式,把图像数据存成文件就得到图像文件。在 Windows 系统中,最常用的图像格式是位图格式,其文件名以 BMP 为扩展名。

图像数字化的精度取决于分辨率和颜色深度两个方面。分辨率是指图像数字化的空间精细度,有显示分辨率和图像分辨率两种不同的分辨率。图像分辨率实质是数字化图像时划分的图像的像素密度,即单位长度内的像素数,通常用每英寸的点数 DPI (Dots per Inch) 表示。显示分辨率则是数字图像在输出设备(如显示器或打印机)上能够显示的像素数目和所显示像素之间的点距。图像分辨率说明了数字图像的实际精细度,显示分辨率说明了数字图像的表现精细度。具有不同的图像分辨率的数字图像在同一设备上的显示分辨率相同。显示器是常见的图像输出设备,现在常见的显示器的分辨率一般可达 1024×768 ,点距为 0.28mm。

颜色深度是指表示每一像素的颜色值的二进制位数。颜色深度越大则能表现的像素的颜色数目越多,它们之间的关系取决于数字图像采用的颜色表示法。常用的颜色表示法有 RGB, CMYK, HSI 和 YUV 等。

2. 色彩系统

根据人眼的结构,所有颜色都可看做是 3 种基本颜色——R 表示红 (Red)、G 表示绿 (Green) 和 B 表示蓝 (Blue),按照不同的比例组合而成。红、绿、蓝被称为三原色。为了科学地定量描述和使用颜色,人们提出了各种色彩系统。目前常用的色彩系统按用途可分为两类:一类面向诸如视频监视器、彩色摄像机或打印机之类的硬件设备;另一类面向以彩色处理为目的的应用,如动画中的彩色图形。面向硬件设备的最常用色彩系统是 RGB 色彩系统,而面向彩色处理的最常用色彩系统是 HSI 色彩系统。另外,在印刷工业上和电视信号传输中,经常使用 CMYK 和 YUV 色彩系统。下面简单介绍常用的几个色彩系统。

RGB 是最常用的色彩系统,此系统图像的每个像素都用其 R, G, B 颜色分量来表示。计算机屏幕显示通常用 RGB 色彩系统,它是通过相加来产生其他颜色,这种做法通常称为加色合成法 (Additive Color Synthesis)。

HSI 色彩系统是 Munseu 提出的,它反映了人的视觉系统观察彩色的方式。HSI 色彩系统中, H 表示色调 (Hue), S 表示饱和度 (Saturation), I 表示亮度 (Intensity, 对应成像亮度和图像灰度)。色调表示基本的纯色;饱和度是指一个颜色的鲜明程度,饱和度越高,颜色越深,其反映的是颜色中掺入白光的比例;亮度是指光波作用于感受器所发生的效应,其大小由物体反射系数来决定,反射系数越大,物体的亮度愈大,反之愈小,其反映的是颜色中掺入黑色的比例。HSI 色彩系统非常适合借助人的视觉系统来感知彩色特性的图像处理算法。HIS 与 RGB 之间可进行转换,其转换关系为

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{3} & 0 & \frac{2}{\sqrt{6}} \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{2}}{2} & -\frac{1}{\sqrt{6}} \\ \frac{\sqrt{3}}{3} & -\frac{\sqrt{2}}{2} & -\frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} I \\ S \cdot \cos H \\ S \cdot \sin H \end{bmatrix}$$

CMYK 色彩系统也是一种常用的表示颜色的方式。与 RGB 色彩系统对比,它是通过颜色相减来产生其他颜色的,所以称这种方式为减色合成法 (Subtractive Color Synthesis)。CMYK 色彩系统的原色为青色 (Cyan)、品红色 (Magenta)、黄色 (Yellow) 和黑色 (Black)。在处理图像时,一般不用 CMYK 模式,主要是因为这种模式的文件大,占用的磁盘空间和内存大,这种模式一般在印刷时使用。CMYK 与 RGB 之间的对应关系如下:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \rightarrow \begin{bmatrix} C' = F - R \\ M' = F - G \\ Y' = F - B \end{bmatrix} \rightarrow \begin{bmatrix} K = \min(C', M', Y') \\ C = C' - K \\ Y = Y' - K \\ M = M' - K \end{bmatrix} \rightarrow \begin{bmatrix} C \\ M \\ Y \\ K \end{bmatrix}$$

其中, F 为满色度数值,通常 RGB 色彩系统满色度数值为 255。

YUV 色彩系统在欧洲彩色电视信号传输时广泛使用 (PAL 系统), PAL 制式将 R, G, B 三色信号改组成 Y, U, V 信号,其中 Y 信号表示亮度, U, V 信号是色差信号。YUV 与 RGB 之间的对应关系如下:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YIQ 色彩系统在北美彩色电视信号传输时广泛使用,其将 R, G, B 改组成亮度信号和色度信号。Y 表示颜色的明视度 (Luminance),即亮度 (Brightness),实际就是图像的灰度值 (Gray Value); I 和 Q 表示色调 (Chrominance),描述图像色彩和饱和度。YIQ 与 RGB 之间的对应关系如下:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

YCbCr 色彩系统也是一种常见的色彩系统,这是常用于彩色图像压缩时的一种色彩系统, JPEG 采用的正是该系统 (它是从 YUV 色彩系统衍生出来的)。与 YUV 色彩系统不同的是它充分考虑了色彩组成时 RGB 三色的重要因素。YUV 考虑的是简单, YCbCr 考虑的是压缩时可以充分取出冗余量。其中 Y 还代表亮度,而 Cb 和 Cr 代表色差,是将 U 和 V 作少量调整而得到的。RGB 和 YCbCr 之间的对应关系如下:

$$\begin{bmatrix} Y \\ Cb \\ Cr \\ 1 \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 & 0 \\ -0.1687 & -0.3313 & 0.5000 & 128 \\ 0.5000 & -0.4187 & -0.0813 & 128 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \\ 1 \end{bmatrix}$$

3. 调色板

现实世界的颜色种类是无限的,但计算机显示系统所能表现的颜色数量是有限的。同时图像数据比较大,计算机的存储空间也是有限的。因此,为了使计算机能最好地重现实际图景,对图像数据进行存储管理操作,就必须采用一定的技术来管理和取舍颜色。

当图像每个像素都用 RGB 分量来表示时, 图像文件会变得很大。例如, 1 个长宽各为 100 个像素, 颜色数为 16 色的彩色图, 每一个像素都用 R, G, B 3 个分量表示, 因为每个分量有 256 个级别, 要用 8 个比特 (bit), 即 1 个字节 (byte) 来表示, 所以每个像素需要用 3 个字节。整个图像要用 $100 \times 100 \times 3$, 约 30k 字节。而 16 色图最多只有 16 种颜色, 可以用 1 个颜色表来进行操作, 设置颜色表中的每一行记录一种颜色的 R, G, B 值, 要表示 1 个像素的颜色时, 只需要指出该颜色是在表中的第几行, 即该颜色在表中的索引值。例如, 如果表的第 0 行为 255, 0, 0 (红色), 那么当图像中的像素为红色时, 利用索引值 0 表示即可。这样 16 种状态就可以用 4 个比特 (bit) 表示, 1 个像素只需 0.5 个字节确定其颜色。图像需用 $100 \times 100 \times 0.5$, 约 5k 字节表示颜色, 再加上索引表的字节数: $3 \times 16 = 48$ 字节, 整个图像占用的字节数约为前面的 1/6, 这样在满足图像色彩表示范围的情况下, 能节省很大的存储空间。调色板 (Palette) 即是这个 RGB 的颜色表, 也称为颜色查找表 LUT (Look Up Table)。Windows 位图及其他许多图像文件格式如 pcx, tif, gif 等都用到调色板技术。

需要注意的是, 真彩色图 (True Color) 每个像素的值都用 24 位来表示, 也称 24 位图像, 真彩色图的像素值与真彩色颜色值可以一一对应, 所以像素值就是所表现的颜色值。因此真彩色系统不需要调色板, 其中的像素值就是 24 位的颜色值。16 色系统通常采用 Windows 的内部系统调色板, 一般并不直接操作调色板。所以, 通常仅在 256 色显示系统中操作调色板。

4. 灰度图像

彩色图像比较复杂, 与之相对存在处理方便一些的简单图像。单色图像是最简单的格式, 它一般由黑色区域和白色区域组成, 可以用 1 个比特表示 1 个像素, “0” 表示黑色, “1” 表示白色, 当然也可以倒过来表示, 这种图像称为二值图像。我们也可以用 8 个比特 (1 个字节) 表示 1 个像素, 相当于把黑和白等分为 256 个级别, “0” 表示为黑, “255” 表示为白, 该字节的数值表示相应像素值的灰度值或亮度值, 数值越接近 “0”, 对应像素点越黑, 相反则对应像素点越白。这种每个像素的信息由 1 个量化的灰度级来描述的图像称之为灰度 (Grayscale) 图像。单色图像和灰度图像又统称为黑白图像。

在 Windows 系统中, 最常用位图格式 (BMP), 并没有用到灰度图像这个概念。而 BMP 文件所采用的 RGB 色彩系统中, 如果 $R=G=B$, 则颜色 (R, G, B) 即为 1 种灰度颜色, 这样, 当 $R=G=B$ 时的值就可定义为灰度值。RGB 值为 (0, 0, 0) 是全黑色, (255, 255, 255) 是全白色, 中间的是灰色。这样灰度图就可以用 256 色图来表示了。本书中对 BMP 格式的图像进行灰度级处理时都是采用这种方法, 因此灰度图处理比较方便, 其 RGB 的值都一样, 图像数据即调色板的索引值, 在图像数据中 1 个字节代表了 1 个像素。图像处理中通常是对灰度图进行操作。

1.1.2 图像处理

图像处理就是对图像信息进行加工处理, 以满足人的视觉心理和实际应用的要求。一般来讲, 对图像进行处理或加工、分析的主要目的有 3 个方面:

(1) 提高图像的视感质量, 如进行图像的亮度、彩色变换, 增强、抑制某些成分, 对图像进行几何变换等, 以改善图像的质量等。

(2) 提取图像中所包含的某些特征或特殊信息, 这些被提取的特征或信息往往为计算