

精通

Visual C++

数字图像处理技术 与工程案例

王占全 徐慧 编著

本书介绍了9个综合性的商业案例，如相机自动调焦系统、人脸检测系统、车牌定位系统、牌照自动识别监控系统等，每个案例都涵盖了数字图像处理技术的关键。

详细讲解了常用数字图像处理技术的基本方法，如点运算、几何变换、图像增强、边缘检测等。

所有案例均按照工程设计思路分步详细说明，读者能够很容易地掌握整个工程的思路以及实现方法。

给出了较完整的基于Visual C++的源代码。



人民邮电出版社
POSTS & TELECOM PRESS



精通

Visual C++

数字图像处理技术 与工程案例

王占全 徐慧 编著

人民邮电出版社

北 京

图书在版编目 (C I P) 数据

精通Visual C++数字图像处理技术与工程案例 / 王占全, 徐慧编著. —北京: 人民邮电出版社, 2009.1
ISBN 978-7-115-18968-4

I. 精… II. ①王…②徐… III. C语言—数字图像处理—程序设计 IV. TP391.41 TP312

中国版本图书馆CIP数据核字 (2008) 第156375号

内 容 提 要

本书是一本详细介绍 Visual C++数字图像编程的书籍。书中通过大量的综合性实例, 向读者展示了如何开发一套完整的数字图像处理应用程序。

书中开始介绍了数字图像处理的基本技术和典型应用, 然后介绍了 9 个综合性的商业案例, 分别是相机自动调焦系统、计算机集成数控技术、细胞识别统计系统、人脸检测系统、车牌定位系统、基于神经网络的文字识别系统、牌照自动识别监控系统、运动检测系统、运动人体跟踪系统等, 并提供了相应的源代码文件, 便于读者轻松实现程序的改编和移植。

本书内容实用、结构清晰、案例丰富, 可以作为高等院校计算机及相关专业的课程设计、毕业设计指导用书, 同时也可以作为科研单位、企业进行数字图像处理相关技术开发的指导用书。

精通 Visual C++数字图像处理技术与工程案例

- ◆ 编 著 王占全 徐 慧
责任编辑 汤 倩
- ◆ 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>
三河市海波印务有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 29.5
字数: 781 千字 2009 年 1 月第 1 版
印数: 1—3 500 册 2009 年 1 月河北第 1 次印刷

ISBN 978-7-115-18968-4/TP

定价: 59.00 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

随着计算机软硬件技术的高速发展,计算机数字图像处理技术在各个领域得到了广泛的应用,如计算机图像识别、图像检索、图像工业化应用等。大部分的高校计算机及相关专业都开设了计算机数字图像处理技术课程。很多研究所也在从事这方面的研究。

为什么写本书

目前进行计算机数字图像编程的主要工具是 Visual C++。虽然其他开发工具,如 Delphi、Visual Basic 等也可以进行数字图像处理,但是在实现速度以及灵活性方面,远远比不上 Visual C++。

虽然市面上数字图像处理的书籍比较多,但大多是介绍基本的数字图像处理算法,要把这些算法应用到实际的工程中,还需要很多复杂的步骤和设计。本书主要介绍如何通过 Visual C++ 实现数字图像处理技术在车牌定位、细胞识别、运动识别等方面的应用。

本书特色

书中首先介绍了数字图像处理的各种基本技术,然后介绍了数字图像处理的一些基本应用,如滚动图片、滚动文字以及小波处理等,最后通过 9 个综合案例全面展示数字图像处理的综合应用,基本上涵盖了当前的热点问题。

本书的特色概括如下。

- 提供了大量的实用程序,如车牌定位、细胞识别、数码相机调焦、运动识别、图像数控编程、人体跟踪系统等,每个系统都涵盖了数字图像处理的关键技术。
- 除了介绍比较完整的系统之外,还详细介绍了常用的数字图像处理技术,如点计算、几何变换、图像增强、边缘检测等。
- 所有的案例均按照工程设计思路,分步解释说明,使读者能够很容易地掌握整个工程的思路以及实现方法。

本书主要内容

本书共 11 章, 各章内容简介如下。

第 1 章 介绍数字图像处理的基本知识, 同时提供了 30 多个数字图像处理中常用的技术, 在提供详细理论分析的同时, 给出了对应的程序实现方法。

第 2 章 介绍典型的数字图像处理技术, 如滚动浏览图片、在图片上显示透明文字以及小波变换技术等。

第 3 章 商业化的基于图像处理的数码相机自动调焦系统。该系统是在现有的相机结构基础上, 改变传统的有源调焦方式(如红外线测距方式), 从获取的数字图像中提取表征图像清晰度的特征, 并以此为判别依据实现自动调焦。该系统通过光学镜头和 CCD 采集一系列的数字图像, 对每一帧数字图像进行实时处理, 判断对焦是否准确, 成像是否清晰, 并给出反馈信号控制 CCD 的位置, 直到采集的图像的清晰度达到最佳。

第 4 章 将图像应用到数控系统的商业程序。在计算机控制的加工系统中, 利用图像信息作为输入数据时, 需要转化成控制系统可以识别的代码。在二维图像坐标中, 颜色信息间接表示图像对象的第三维坐标的几何信息, 由此可将二维灰度图像经矢量化处理转成三维几何图形, 然后进行数控加工编程处理; 也可以利用数学方法将图像处理生成骨架或者轮廓, 再进行处理。

第 5 章 一个功能完善的细胞识别统计系统。该系统是医院中血液红细胞计数的实用系统, 主要是以病人的血液样本(玻璃切片)为原始数据, 通过一系列的图像处理和分析, 最后识别出血液中的红细胞, 同时给出红细胞的个数。得到红细胞个数之后, 通过检测的血液量, 就可以得出血液中红细胞的密度, 大大方便了临床诊断。

第 6 章 人脸检测系统。本系统是为了实现对图片中的人脸进行检测而设计的。人脸跟踪是目前图像研究领域的热门话题, 而人脸检测是其中最基础的工作。系统通过对图片进行一系列的处理, 提取出人脸的大致框架, 然后通过眼睛、嘴巴的建模, 最后得到人脸的比较精确的位置。本系统可以单独使用, 也可以与人脸识别系统组合使用, 在人脸检测的基础上实现人脸识别, 实现更为广泛的应用。

第 7 章 车牌定位系统。本章是进行车牌识别一个基础部分, 只有将车牌上面的文字部分正确地分割出来, 才能使后面的文字、数字识别成为可能。

第 8 章 基于神经网络的文字识别系统。该系统是为了辨认、识别图像中的数字而设计的, 它通过对图片的一系列处理, 最后识别得出图片中显示的数字。系统既可以单独使用, 又可以作为识别系统的软件核心应用到车牌识别等系统中去。

第 9 章 一个综合的牌照自动识别监控系统。该系统是在交通监控的基础上, 引入了数字摄影技术和计算机信息管理技术; 采用先进的图像处理、模式识别和人工智能技术, 卫星网络通信技术, 通过对图像的采集和处理, 获得更多的信息, 从而达到高智能化管理。汽车牌照自动识别系统本身就是一个全数字化的智能系统, 在它上面只要做不多的扩充, 就可以衍生出一些其他功能。汽车牌照自动识别系统可广泛安装于公路收费站、停车场、十字路口等交通关卡处, 使收费的管理更严密、更科学, 能有效地监控车辆的收费、闯关、欠费以及各种舞弊现象的发生。

第 10 章 运动检测系统。该系统是为了实现对一系列图片中运动物体的跟踪而设计的。通过对图片的处理, 提取出图片背景, 并识别出运动物体, 进而对运动物体进行跟踪, 得出物体运动的轨迹。系统既可以单独使用, 比如截取电影中的一系列图片进行分析, 得到某个运动物体的

运行路线；也可以作为监视系统的软件核心，应用到办公室、走廊等地的监视系统中。

第 11 章 运动人体跟踪系统。本章涉及的 CT (Computer Tracking) 系统是一种适用于室内外环境的视频检测与跟踪系统。整个系统由运动物体检测、运动物体分析、人体运动跟踪 3 部分组成。系统完全由软件实现，可运行于普通个人计算机上，使用单个静止 CCD 数字摄像机得到的灰度图像序列作为输入，也可以使用红外线图像序列，对单人或多人的运动行为进行监控。

本书由王占全、徐慧主编，参与写作的人员还有卜利君、蔡敏、陈华智、陈连根、陈涛、邓兴业、何贤辉、刘红涛、刘海英、厉蒋、金芳芳、姜琴英、苏淑滢、张固、张静、赵海波、周洁、余声扬、叶达峰、杨易、赵斯思和朱慧敏等。

在编写过程中，我们力求精益求精，但难免存在一些不足之处，如果读者使用本书时遇到问题，可以发 E-mail 到 computerbook@126.com 与我们联系。

编 者

目 录

第 1 章 数字图像处理技术	1	1.7 图像的点运算	22
1.1 数字图像	1	1.7.1 图像的反色	22
1.1.1 单色图像	1	1.7.2 灰度阈值变换	23
1.1.2 灰度图像	1	1.7.3 灰度拉伸	24
1.1.3 彩色图像	2	1.7.4 灰度均衡	26
1.2 Windows 颜色和调色板	3	1.7.5 256 色转灰度图	28
1.2.1 视频卡和调色板	3	1.7.6 亮度增减	29
1.2.2 Windows 系统调色板和逻辑 调色板	4	1.7.7 取对数	31
1.2.3 Windows 的调色板管理	6	1.7.8 取指数	32
1.2.4 使用调色板	7	1.8 图像几何变换技术	33
1.3 BMP 图像文件	8	1.8.1 图像的平移变换	33
1.3.1 BMP 文件格式	8	1.8.2 图像的旋转变换	36
1.3.2 Windows 中的位图格式	11	1.8.3 图像的镜像变换	42
1.4 DDB 位图	11	1.8.4 图像的缩放变换	44
1.4.1 创建 CBitmap 对象和加载位 图数据	11	1.8.5 插值算法简介	46
1.4.2 将 CBitmap 绘制到屏幕上	13	1.9 图像的增强技术	47
1.5 DIB 位图	14	1.9.1 高斯平滑处理	47
1.5.1 DIB 文件格式	14	1.9.2 中值滤波	50
1.5.2 DIB 访问函数	17	1.9.3 图像的锐化处理	52
1.6 使用 Visual C++ 打开并显示 BMP 图像	19	1.10 边缘检测及轮廓跟踪技术	54
1.6.1 CBMPViewerDoc 头文件	20	1.10.1 边缘检测算子	54
1.6.2 浏览 BMP 文件	20	1.10.2 Hough 变化	60
		1.10.3 轮廓提取	64
		1.10.4 轮廓跟踪	66
		1.10.5 模板匹配	69

1.10.6 用 Canny 算子提取边缘	72	4.2.3 CAM 知识	156
1.11 图像分割技术	74	4.2.4 数学形态学的理论基础	157
1.11.1 投影法	74	4.3 关键技术和程序实现	158
1.11.2 差影法	77	4.3.1 Windows 下的图像表示和 操作	158
1.12 图像腐蚀、膨胀、细化算法	78	4.3.2 直接加工算法	158
1.12.1 图像的腐蚀	79	4.3.3 细化算法	159
1.12.2 图像的膨胀	82	4.3.4 矢量化算法	161
1.12.3 图像的细化	86	4.4 程序实现	162
第 2 章 数字图像处理典型应用	92	4.4.1 直接加工	162
2.1 透明位图背景上的滚动字幕实例	92	4.4.2 细化 CAM	170
2.1.1 编程原理	92	4.4.3 矢量 CAM	183
2.1.2 实例详解	99	4.5 数控代码仿真器	186
2.2 滚动浏览位图	99	第 5 章 细胞识别统计系统	189
2.2.1 动态装载位图到对话框	100	5.1 血液红细胞识别计数系统	189
2.2.2 利用滚动技术显示原始 尺寸的位图	101	5.2 系统基本技术要求	189
2.2.3 双缓存技术防止闪烁	103	5.3 系统架构及实现	189
2.3 小波算法的 Visual C++ 实现	104	5.3.1 系统硬件平台	190
2.3.1 设计思路	104	5.3.2 系统软件平台	190
2.3.2 编程原理	105	5.3.3 程序实现	190
2.3.3 实例详解	109	第 6 章 人脸检测系统	222
第 3 章 相机自动调焦系统	113	6.1 系统简介	222
3.1 系统简介	113	6.2 系统基本技术要求	222
3.2 系统基本技术要求	113	6.3 系统架构及实现	222
3.3 系统实现平台	113	6.3.1 系统硬件平台	222
3.3.1 系统硬件平台	114	6.3.2 系统用到的关键技术	223
3.3.2 系统软件平台	115	6.3.3 程序实现	223
3.4 系统关键技术	115	第 7 章 车牌定位系统	255
3.4.1 图像预处理	115	7.1 系统简介	255
3.4.2 消除光线的影响	115	7.2 系统基本技术要求	255
3.4.3 消除噪声的影响	117	7.3 系统架构及实现	255
3.4.4 清晰度计算	118	7.3.1 系统硬件平台	255
3.4.5 反馈控制	121	7.3.2 系统软件平台	255
3.4.6 程序实现	124	7.3.3 系统中用到的关键技术	256
第 4 章 计算机集成数控技术	153	7.3.4 系统实现	256
4.1 技术要点	153	7.3.5 程序效果测试	272
4.2 基本理论	153	第 8 章 基于神经网络的文字识别系统	278
4.2.1 数字控制技术	153	8.1 系统简介	278
4.2.2 数控代码	154	8.2 系统基本技术要求	278

8.3 系统中用到的关键技术	278	9.5 系统中其他关键技术与算法	388
8.4 系统软硬件平台	278	第 10 章 运动检测系统	390
8.4.1 系统的硬件平台	278	10.1 运动判定系统	390
8.4.2 系统的软件平台	279	10.2 系统基本技术要求	390
8.5 系统实现	279	10.3 系统架构及其实现	390
8.5.1 系统流程图	279	10.3.1 硬件系统平台	390
8.5.2 图像数据的读取、保存与 屏幕显示等基本函数	279	10.3.2 系统用到的关键技术	391
8.5.3 图像的预处理	288	10.3.3 程序实现	391
8.5.4 其他相关的重要图像处理 技术	320	第 11 章 运动人体跟踪系统	420
8.5.5 用神经网络进行字符识别	328	11.1 系统简介	420
8.5.6 程序总体编程框架	350	11.2 系统应用范围及意义	421
8.5.7 程序使用说明、测试及 注意事项	351	11.2.1 视频监控	421
第 9 章 牌照自动识别监控系统	355	11.2.2 人机界面	421
9.1 系统简介	355	11.2.3 运动诊断与辨别	421
9.2 系统基本技术要求	355	11.3 系统软硬件平台	421
9.3 系统架构及实现	356	11.3.1 系统的硬件平台	421
9.3.1 系统硬件平台	356	11.3.2 系统的软件平台	421
9.3.2 系统软件平台	357	11.4 人体运动目标检测与跟踪 理论概述	421
9.4 牌照识别关键技术与算法	361	11.4.1 运动目标检测	421
9.4.1 车辆图像采集	361	11.4.2 运动目标跟踪	422
9.4.2 牌照区域提取	363	11.5 系统实现	423
9.4.3 牌照图像二值化	371	11.5.1 系统流程图	423
9.4.4 牌照字符分割	376	11.5.2 读取与屏幕显示等基本函数	424
9.4.5 牌照字符识别 (神经网络)	377	11.5.3 人体运动检测部分的理论 简介和基本函数	433
9.4.6 车辆视频检测	381	11.5.4 运动区域分析部分的理论 简介和基本函数	449

第 1 章

数字图像处理技术

传统的计算机只能处理文字、数字或是简单的图形。随着计算机硬件技术的飞速发展和更新,计算机处理图形图像的能力大大增强。以前要用大型图形工作站来运行的图形应用软件,或是生成特殊的文件格式,及对图形做各种复杂的处理和转换,如今,很普遍的家用电脑就可以胜任,通过 Photoshop、CorelDRAW、3ds MAX 等软件轻松做出精美的图片或是逼真的三维物体。

在当今信息社会,以多媒体为代表的信息技术和信息产业的发展对人类社会产生的影响愈来愈明显,愈来愈重要。多媒体的发展和应用,极大地推动了诸多产业的相互渗透和飞速发展,逐步改变了整个人类社会的工作结构和生活方式。

所谓多媒体,即多种信息媒介,通常包括文本、图形、影像、声音、视频和动画等。可以看出,多媒体的应用在很大程度上依赖于丰富多彩的图形和图像。也就是说,图形图像技术的飞速发展也将是必然趋势。

1.1 数字图像

图像由栅格点(像素点)组成,每点的颜色(亮度或者颜色值)都单独记录。由于利用栅格数据点映射图像中的像素点,因此又名位图,其中位图区域中数据点的位置即为对应像素点的位置。

1.1.1 单色图像

图像都具有颜色,单色图像一般由黑色和白色组成,图像中每个像素点都用单独一位进行描述:1 表示该点为黑色,0 表示该点为白色。当然,也可以用 1 表示该点为白色,而 0 表示该点为黑色。典型的单色图像如扫描文档等。

1.1.2 灰度图像

很多的非彩色图像用单色不能表达图像的真实视觉效果,这时用灰度来表达。当在不支持彩色的显示设备上显示彩色图像时,一般需要利用相应的灰度图像对其进行近似处理,因此灰度图像

是经常使用到的一种图像表示方式。

描述灰度图像有两种比较典型的方案。由于人眼会将一个很细致的黑白图案的像素解释为灰色，因此可以利用单色图像描述灰度图像，数据点仍然可以是黑或者白。这种利用黑色或者某一种单色的点描述连续的该色灰度的过程称为半色调处理（half toning）或者抖动处理（dithering），该技术适用于绝大多数的打印图像，因为打印图像时，通常一次只能使用一种颜色的墨水。

但是，在单色图像中使用半色调处理技术获得灰度图像的方法存在着很多的缺点。一方面，它使图像的编辑工作相当困难，因为大多数图像处理程序无法区分带阴影的点和用于描绘直线的点；另一方面，使用半色调处理的图像在显示灰度时层次太明显，不具备连续性和光滑性；而且，半色调处理技术会降低灰度图像的分辨率，因为它必须使用几个像素点，才能描述单独一个点的灰度值。因此，半色调处理一般很少使用，绝大多数的灰度图像描述方法都是采用下面的灰度级记录方法。

将灰度图像记录为真正的灰度级，从而简化灰度图像的编辑工作。任何一幅真正的灰度级图像都具有一定的位深度。一幅灰度图像一般根据 3 个尺度进行描绘：列的像素数（宽）、行的像素数（高），以及用于表示像素亮度的位数——图像的位深度，即灰度级分辨率（grayscale resolution）。例如，一个 400 列、600 行和 16 个深度级的图像粗略大小为 $400 \times 600 \times (1/2)$ 字节，即 120 000 字节。由于人眼可以很容易地分辨出所给图像的 64 级灰度，所以对于一个真实的灰度图像，经常需要 6 个位的深度。对于灰度图像，所使用的像素位一般都多于一位，这些位的内容即代表颜色号，通常 0 表示黑色，最高值（8 位时为 255）表示白色，中间值则代表灰度的级别。

典型的灰度图如遥感图像、医学图像等。

1.1.3 彩色图像

现实世界中大多数图像都是彩色的。要描述彩色图像，必须先确定颜色模式。彩色图像中最常用的颜色模式是 RGB 和 CMYK。

RGB 模式也称为添加模式，主要用于发光设备，如显示器等。RGB 分别代表 3 种基色：红色、绿色、蓝色。图像颜色形成过程为：从黑色开始，逐渐增加每种基色的颜色值，从而产生指定的像素颜色。绝大多数图像都采用这种颜色模式描述像素点的颜色。

CMYK 则称为削减模式，常用于打印、感光材料等。它使用 4 种基色，分别为青色（Cyan）、品红（Magenta）、黄色（Yellow）和黑色（Black），其颜色形成过程为：从白色开始，逐渐削减 4 种基色的颜色值。这种颜色形成方式类似于在一张打印纸上使用彩色喷墨技术，所以这种模式主要用于将一幅生成的图像输出到彩色打印机中。

IHS 是另一种重要而且应用广泛的彩色模式，在后面 IHS 融合图像一节将会有详细介绍。另外还有很多彩色模式，如 YIQ、YUV、YcbCr 等，本书就不涉及了。

Windows 中数字图像都采用 RGB 模式，图像中每个像素点被赋予不同的 RGB 值，如 (255,128,0)，255 代表红色分量的值，128 代表绿色分量的值，0 代表蓝色分量的值。因为一个字节 8 位，所以通常用 1 个字节来表示每种分量，这样每种分量值的范围为 0 到 255，共 256，即 2^8 个等级，0 表示不含该颜色成分，255 表示含有 100% 的该颜色成分。这样红、绿、蓝各种不同的组合可以产生 $256 \times 256 \times 256$ 种颜色。

表 1-1 所示是常见的一些颜色的 RGB 组合值。

表 1-1 常见颜色的 RGB 组合

颜 色	红 色 分 量	绿 色 分 量	蓝 色 分 量
黑色	0	0	0
白色	255	255	255
红色	255	0	0
绿色	0	255	0
蓝色	0	0	255
青色	0	255	255
紫色	255	0	255
黄色	255	255	0
灰色	128	128	128

如果图像中每个像素直接用 RGB 组合值表示,则称之为真彩色图像。但如果一幅图像每个像素都用其 RGB 分量来表示,那么所有的图像文件都将变得非常庞大。例如对一幅 200×200 的 16 色图像,它共有 40 000 个像素,如果每一个像素都用 R、G、B 3 个分量表示,则一个像素需要 3 个字节。这样保存整个图像要用 $200 \times 200 \times 3$,即 120 000 字节。图像只有 16 种颜色,完全可以用 4 位的组合来区分 16 种不同的颜色,因此这幅图像也可以用下面的方法来表示:采用一个颜色表,表中的每一项记录一种颜色的 R、G、B 值,当表示一个像素的颜色时,只需要指出该颜色是在第几项,即该颜色在表中的索引值即可。例如,如果表的第 0 项为 (255,0,0),那么当某个像素为红色时,只需要标明 0 即可。通过颜色索引表来表示图像,16 种状态可以用 4 位 (bit) 表示,所以一个像素要用半个字节。整个图像要用 $200 \times 200 \times 0.5$,即 20 000 字节,再加上颜色表占用 $3 \times 16 = 48$ 字节,也不过 20 048 字节。这样一幅图像整个占用的字节数只是前面的 1/6。这张 RGB 表就是通常所说的调色板 (palette),它还有另外一个名称——颜色查找表 LUT (Look Up Table)。

由于输入输出设备的限制,最初的颜色都是使用调色板的。尽管许多现代的视频卡提供 24 位颜色 (真彩色) 或 16 位颜色 (增强色),可以不用操作调色板,但还有很多视频卡或提供真彩的视频卡在高分辨率模式中,每个像素只允许 8 位或更少,在这样的系统中还必须操作调色板。

1.2 Windows 颜色和调色板

调色板是提供给图像使用的一系列颜色的集合表。可以根据需要,从所有可以利用的颜色中挑出一组所需的,形成一个调色板,以产生所需的颜色效果。

1.2.1 视频卡和调色板

调色板的出现和发展是与视频卡的发展息息相关的。CGA 是最早的彩色图形显示卡,其颜色由 iRGB (加亮和蓝、绿、红) 4 个变量定义,能表示 16 种颜色。由于显存空间的限制,屏幕上每个像素只用两位表示,因此同屏只能显示 4 种颜色。CGA 内部有一张表,可根据两位的像素值从 16 种颜色中找出所表示的颜色。

VGA (视频图形阵列) 视频卡有 256 个 18 位的颜色寄存器,每个寄存器中红绿蓝三原色各占 6 位,这样就可以表示 262 144 (即 2^{18}) 种颜色。颜色寄存器中的 RGB 值最后决定

显示的颜色。由于显存空间的限制，VGA 卡在高分辨率（ 640×480 ）下，屏幕上每个像素在显存中只用 4 位表示，只能同屏显示 16 种颜色；在低分辨率（ 320×200 ）下，屏幕上每个像素在显存中用 8 位表示，可同屏显示 256 色。低分辨率下，VGA 使用全部的 256 个颜色寄存器，8 位的像素值直接被当做索引来检索对应的颜色寄存器，从而得到该像素的颜色。高分辨率下，VGA 只使用 16 个颜色寄存器，并且可能不是前 16 个连续的颜色寄存器，所以 4 位的像素值并不能作为索引直接检索对应的颜色寄存器。这时候 VGA 使用了一张颜色查找表，查找表由 16 个调色板寄存器组成。屏幕像素值作为索引指向了所对应的调色板寄存器，调色板寄存器中的一个 6 位值和 VGA 属性控制器中的彩色选择寄存器一起构成一个 8 位的索引指向颜色寄存器，从而得到像素的颜色。这 16 个颜色寄存器就组成了 VGA 的硬件调色板（Hardware Palette）。

TVGA 视频卡增强了 VGA，它提供了一张 256 项的颜色查找表，也就是一个 256 项的硬件调色板。TVGA 256 色模式时，8 位的像素值就是相应的调色板编号，由这个编号可以找到相应的颜色寄存器。

视频卡硬件调色板支持的 RGB 颜色位数又称为视频卡的颜色分辨率。SVGA 以前的 VGA 卡，包括 TVGA 等，颜色分辨率为 18；从 SVGA 开始，大多数视频卡颜色寄存器有 24 位，其颜色分辨率为 24，能提供 $16\,777\,216$ （即 2^{24} ）种颜色。

真彩色视频卡提供的图像质量比 SVGA 更高，它有足够大的显存以支持每像素 24 位（有的显卡用 16 位）的颜色信息，它能同时显示 2^{24} 种颜色。图像中每个像素有其独立的彩色定义，而不会由于调色板的项数而受限制。在真彩色视频卡中显示图像无需借助调色板。

1.2.2 Windows 系统调色板和逻辑调色板

下面介绍 Windows 系统调色板的相关知识。

1. Windows 颜色

不同视频卡表示颜色的方式不同，因此 Windows 定义了一种设备无关的颜色表示方式。人们可以方便地使用这些颜色，而不必管视频卡的颜色是如何表示的，Windows 会在这两种颜色之间建立映射。Windows 中用一个名为 COLORREF 的 32 位值来定义颜色。COLORREF 存储着 RGB 值或调色板索引，它有 3 种结构。

0x00BBGRRR：最高字节为 0x00，这种结构后 3 个字节分别存储 BGR 值。

0x0100nnnn：最高字节为 0x01，这种结构最后两个字节为指向系统调色板的索引。

0x02BBGRRR：最高字节为 0x02，这种结构后 3 个字节存储调色板的 BGR 值。

每一种结构，Windows 都提供一个对应的宏来创建。第一种最简单，可以用 RGB 宏创建。例如下面代码创建一个纯红刷子。

```
brRed.CreateSolidBrush(RGB(255,0,0));
```

如果 RGB 宏指定的颜色与硬件调色板中某种颜色相同，则刷子的颜色为 RGB 宏指定的颜色（ 0×00000011 ）；如果没有相同的颜色，则由硬件调色板中已存在的颜色抖动来匹配。

如果用户知道硬件调色板特定的索引位置有自己需要的颜色，可以用 PALETTEINDEX 宏创建一个 COLORREF 结构，比如如果硬件调色板第 16 号是纯红色，可用如下代码。

```
brRed.CreateSolidBrush(PALETTEINDEX(16));
```

来创建一个纯红色的刷子，此宏创建的颜色为 0×01000010 。

PALETTERGB 宏指定颜色与 RGB 宏类似，但在硬件调色板中没有相同的颜色时，不是通过抖动来匹配，而是在硬件调色板中选择与之最接近的颜色，如下所示。

```
brRed.CreateSolidBrush(PALETTERGB ( 255,0,0 ) );
```

2. Windows 系统调色板

在 256 色模式下运行时，为了方便管理，Windows 还定义了一个调色板，称为系统调色板 (System Palette)，此调色板与显示卡上的硬件调色板相对应。系统调色板有固定的 20 个默认的颜色，对 EGA 和标准 VGA (16 色模式)，这 20 种颜色中只有 16 种可显示的颜色，剩下的 4 种颜色由抖动来显示。对支持 256 色及更多颜色的视频卡，20 种默认颜色作为确定的单一颜色出现，不能修改。由于 20 种默认颜色 (见表 1-2) 不能由程序改变，因此称之为“Windows 标准颜色”，Windows 使用这些颜色绘制对话框和图标等。这 20 种颜色驻留在系统调色板的最前面和最后面，即第 0~9 和第 246~255 共 20 项。在 Windows 2000 下，这 20 种颜色值如表 1-2 所示。

表 1-2

Windows 2000 系统调色板默认条目

序 号	红	绿	蓝	颜 色
0	0	0	0	黑色
1	128	0	0	暗红色
2	0	128	0	暗绿色
3	128	128	0	暗黄色
4	128	0	0	暗蓝色
5	128	0	128	暗紫色
6	0	128	128	蓝青色
7	192	192	192	亮灰色
8	192	220	192	淡绿色
9	212	208	200	中灰色
246	255	251	240	乳白色
247	58	110	165	桌面蓝
248	128	128	128	暗灰色
249	255	0	0	红色
250	0	255	0	绿色
251	255	255	0	黄色
252	0	0	255	蓝色
253	255	0	255	品红色
254	0	255	255	青色
255	255	255	255	白色

图 1-1 是在 8 位显示模式下显示的系统调色板 (按从左到右，从上到下的顺序)，其中前后各 10 项是 Windows 2000 的标准颜色。不同的情况下，除这 20 项，其他的调色板项也是不同的。可

以通过下一节讲解的程序，来显示当前的系统调色板。

3. 逻辑调色板

应用程序要通过修改硬件调色板来实现正确的显示，但 Windows 能同时运行多个程序，如果每个程序都按自己的需要改变硬件调色板，就会影响系统自身的和其他应用程序的显示。所以 Windows 3.0 版本开始支持每个应用程序创建一个逻辑调色板 (Logical Palette)，该调色板包含了该应用程序使用的颜色集，所有的应用程序和系统自身都可以在任何时候使用这个颜色集而不必考虑当前系统调色板的状态。

在 WINGDI.H 文件中定义了逻辑调色板结构 LOGPALETTE。

```
typedef struct tagLOGPALETTE{  
    WORD palVersion; // windows 版本  
    WORD palNumEntries; // 调色板条目数  
    PALETTEENTRY palPalEntry[1];  
}LOGPALETTE;
```

在当前所有 Windows 版本中，palVersion 都是 0x300。PALETTEENTRY 表示调色板条目，结构定义如下。

```
typedef struct tagPALETTEENTRY{  
    BYTE peRed; //调色板表项的红色分量  
    BYTE peGreen; //调色板表项的绿色分量  
    BYTE peBlue; //调色板表项的蓝色分量  
    BYTE peFlags;  
}PALETTEENTRY;
```

PeFlags 是个标志，确定如何使用调色板表项。取 NULL 表示这是普通的调色板表项，还可取其他几个值，对应含义如表 1-3 所示。

表 1-3 调色板项标志

标 志	值	说 明
PC_EXPLICIT	0 × 02	该调色板表项的低字节 WORD 是一个硬件调色板索引，应用程序能取得当前显示的调色板的表项
PC_NOCOLLAPSE	0 × 04	创建一个被映射到系统调色板中未用条目上的调色板条目，即使该颜色条目已经存在。用于当两个条目映射到同一个颜色上时，确保调色板颜色的唯一性
PC_RESERVED	0 × 01	表示颜色将用于调色板动画制作，因此不能将其他应用程序的颜色映射到此项上

应用程序通过将自己的逻辑调色板映射到系统调色板，以实现正确显示而不会不影响其他程序和 Windows 系统。

1.2.3 Windows 的调色板管理

在大于 256 色的显示系统中，像素值就是 24 位的 RGB 颜色值，不需要操作调色板。16 色显示系统通常采用 Windows 的默认调色板，也不需要操作调色板。所以，仅在 256 色显示设备中运行

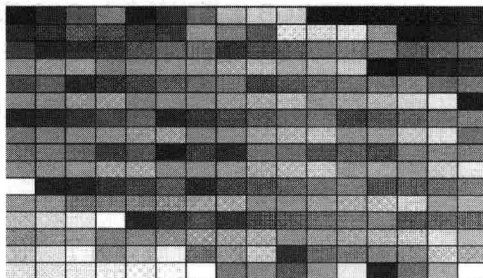


图 1-1 Windows 2000 系统调色板

程序,才需要操作调色板。

每款 Windows 的应用程序都有自己的调色板,在使用调色板时,首先要向 Windows 提出申请,Windows 根据该应用程序的优先级来对应用程序调色板进行分配。前台程序在选择系统调色板时有优先权,后台程序必须设法处理提供给它的颜色集。如果前台程序没有把系统调色板(236 色)用完,它们可以根据先到先服务的原则使用余下的项;如果用完,则用系统调色板中最接近的颜色匹配,即映射到系统调色板中已确定的颜色项上。Windows 使用“调色板管理器(Palette Manager)”对调色板进行管理。

创建了程序的逻辑调色板,然后调用 `RealizePalette`,调色板管理器就将程序逻辑调色板上的每种颜色都映射到系统调色板的颜色上。映射调色板的过程从逻辑调色板的第一个颜色开始,直到最后一个颜色。因此,逻辑调色板开始部分的颜色被准确实现的可能性比后面的大。如果一些颜色对用户比其他颜色更重要,可将它们放在逻辑调色板的开始部分。

为了将颜色从逻辑调色板映射到系统调色板的颜色项上,调色板管理器首先寻找系统调色板上其红绿蓝值与逻辑调色板完全匹配的已确定的颜色项(对于前景程序,调色板中间的 236 项都还未被确定)。首先从 0 项开始直至 255 项。如果找到完全匹配的,则将颜色映射到该系统调色板颜色项上。如果没有找到完全匹配的颜色项,调色板管理器又从 0 开始寻找系统调色板上没有用过的颜色项。如果找到一个没用过的项,就将颜色的 RGB 值复制到系统调色板上,这样也将逻辑颜色映射到了系统调色板上。如果系统调色板上没有空余项,调色板管理器再从 0 开始寻找与逻辑调色板要求的最接近的颜色,并且将逻辑调色板上的颜色映射到该系统调色板颜色项上。两种颜色的接近度由红、绿、蓝矢量差平方和的平方根来度量,此值越小,颜色越接近。这个算法能在合理的计算时间内提供一个可接受的结果,但这并不是最佳的算法。可以看到,调色板映射包括颜色的复制,相同颜色的映射和相近颜色的映射。

对于前台程序,除了 20 种 Windows 标准颜色,系统调色板中的 236 项都还没有确定,映射过程会尽量把所有的颜色复制到系统调色板中。

对于后台程序,如果前台程序使用了 236 种颜色,那我们只能通过调色板管理器把逻辑调色板中的颜色映射到系统调色板中已确定的颜色上。如果前台程序没有占用所有的 236 项系统调色板,那么后台程序中最早申请调色板的程序将通过调色板管理器把自己的逻辑调色板按照规则映射系统调色板中。

映射的过程还受一些标志的控制。如果系统调色板某项上设置了 `PC_RESERVED` 标志,其他颜色就不能映射到它上面。如果逻辑调色板某项上设置了 `PC_NOCOLLAPSE` 标志,那么,调色板管理器将不会把该颜色映射到任何已存在的颜色项上,而是在系统调色板上寻找空余项。

1.2.4 使用调色板

大于 256 色的显示设备不支持调色板管理器,即使是显示调色板图像,也不需要对调色板进行操作。因此在使用调色板之前,首先得确定显示设备是否支持调色板。

1. 判断显示设备是否支持调色板

可以通过下面的方法确定显示设备是否支持调色板操作:

判断 `GetDeviceCaps(hdc, RASTERCAPS)&RC_PALETTE` 的值,如果此值非 0,则显示设备支持调色板。

`GetDeviceCaps` 是个非常有用的函数,通过它可以得到其他一些系统的参数。例如:

GetDeviceCaps(hdc, BITSPIXEL)返回系统每种颜色需要的位数,即位深度。GetDeviceCaps(hdc, SIZEPALETTE)返回系统调色板的总项数,8位和8位模式以下值为“ $1 \ll \text{bits}$ ”(bits为位深度),大于8位值为0。GetDeviceCaps(hdc, NUMRESERVED)返回系统调色板的默认颜色数,为20。GetDeviceCaps(hdc, COLORRES)返回硬件调色板支持的RGB分辨率,为18或者24。还有其他的应用,可以参阅MSDN。

2. 在不匹配的显示系统上显示图形

Windows 3.0 开始定义了设备无关位图(DIB)。DIB的数据与设备无关,它能在与其配置不匹配的显示系统中显示。当颜色深度(每个像素的位数)不匹配时,颜色匹配算法被使用,以获取最好的颜色匹配。在不支持调色板的显示系统上显示调色板图像,系统会直接使用调色板的RGB值来显示相应的像素。在调色板显示系统上显示真彩图时,颜色减少操作在绘图操作中被执行,但通常效果都不理想。另一种Windows位图——设备相关位图(DDB)则只能绘制到与其自身配置匹配的显示系统上。

3. 使用调色板

使用调色板的第一步就是调用CreatePalette函数来创建逻辑调色板。初始化LOGPALETTE结构,并将这个结构的指针传递给CreatePalette函数。CreatePalette返回逻辑调色板的句柄(HPALETTE)。

在使用调色板之前,必须确保将其选入设备描述表。因此在响应WM_PAINT消息时,调用SelectPalette将逻辑调色板选进设备描述表。SelectPalette的第3个参数通常设为FALSE,以确保当应用程序在前台运行时,相应的调色板被实现为前台调色板。如果使用TRUE为参数,那么调色板将始终是“后台调色板”,这意味着当其他所有程序都实现了各自的调色板后,该调色板才可以获得位于系统调色板中的未使用的条目。

将逻辑调色板选入设备描述表之后,RealizePalette函数将逻辑调色板映射到系统调色板上。

1.3 BMP 图像文件

前面介绍了Windows图像处理的一些基本知识,下面介绍数字图像处理最经常使用的BMP文件的基本知识。数字图像文件的格式多达几十种,平时经常遇到的图像格式有JPG、GIF、BMP、PCX等。但是这些图像格式并不是都适合进行复杂的图像处理技术,因为有些图像都是经过压缩的,如JPG、GIF等文件格式。本书主要介绍的是数字图像处理技术,进行数字图像处理的前提是图像必须是未经压缩的,BMP文件通常都是以未压缩方式存储的,因此最适合进行图像处理。



提示 压缩过的图像是无法直接进行数字化处理的,如JPEG、GIF等格式,此时首先要对图像文件进行解压缩,将图像还原成BMP文件格式,读者可以参考相关书籍和资料。

1.3.1 BMP 文件格式

BMP格式是微软公司为其Windows操作系统设置的标准图像格式,在Windows系统软件中包含了一系列支持BMP图像处理的API函数。随着Windows操作系统在PC机上的流行,BMP