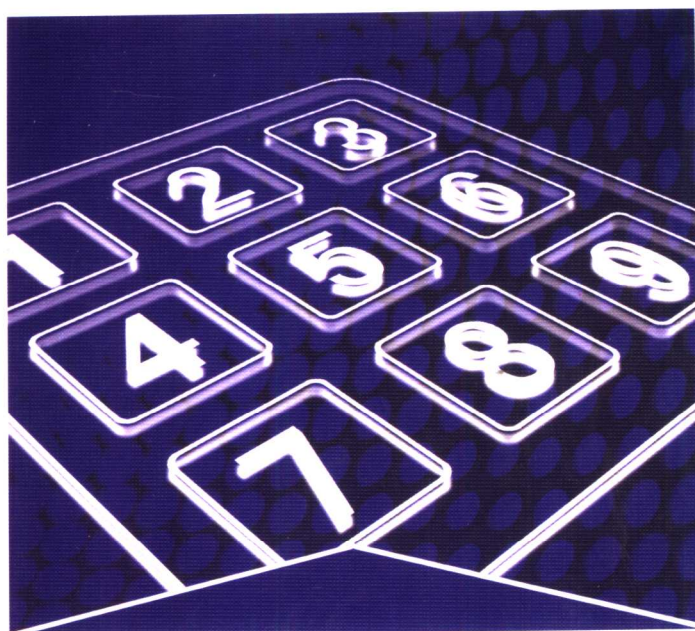


电脑编程实例导航丛书



Visual C/C++图形图像与游戏编程

典型实例解析

《电脑编程技巧与维护》杂志社 编著



中国水利水电出版社
www.waterpub.com.cn

电脑编程实例导航丛书

Visual C/C++ 图形图像与游戏编程 典型实例解析

《电脑编程技巧与维护》杂志社 编著

中国水利水电出版社

内 容 提 要

本书本着实用的原则,通过一个个典型案例的解析,深入地介绍了使用 Visual C/C++ 进行图形图像处理和游戏编程的方法及技巧。

全书分三章共计 56 个实例。第 1 章是图形图像、游戏编程基础与应用实例,该章使用 Visual C/C++ 进行图形图像、游戏编程基础与应用列举了 14 个典型而实用的编程实例,为图形图像处理应用和游戏编程奠定了基础。第 2 章是图形图像编程实例,该章精选了 34 个 Visual C/C++ 在图形图像处理应用中典型而实用的编程实例。第 3 章是游戏编程实例,该章精选了 8 个 Visual C/C++ 在游戏编程应用中典型而实用的编程实例。

本书内容丰富,实用性强,所有实例的源代码都经过了测试和运行,并给出了详细解释,浓缩了 Visual C/C++ 图形图像处理与游戏程序设计的精华,是图形图像处理与游戏编程人员的经验之作。

本书适合具有一定经验的 Visual C/C++ 图形图像处理与游戏编程人员,对初学 Visual C/C++ 进行图形图像处理与游戏编程的新手也有很高的参考价值。

本书所有实例源代码可以从中国水利水电出版社网站下载,网址是: <http://www.waterpub.com.cn/softdown/>。

图书在版编目(CIP)数据

Visual C/C++ 图形图像与游戏编程典型实例解析/
《电脑编程技巧与维护》杂志社编著. —北京:中国水
利水电出版社, 2006

(电脑编程实例导航丛书)

ISBN 7-5084-4082-X

I.V… II.电… III.C 语言—程序设计

IV.TP312

中国版本图书馆 CIP 数据核字 (2006) 第 113337 号

书 名	Visual C/C++ 图形图像与游戏编程典型实例解析
作 者	《电脑编程技巧与维护》杂志社 编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787mm×1092mm 16 开本 19.75 印张 594 千字
版 次	2006 年 10 月第 1 版 2006 年 10 月第 1 次印刷
印 数	0001—4000 册
定 价	34.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换

版权所有·侵权必究

丛 书 序

《电脑编程技巧与维护》杂志是为从事电脑编程、系统应用开发的人员创办的专业性和实用性都很强的技术刊物。自 1994 年创刊以来，始终以“实用第一，智慧密集”为宗旨，坚持“质量第一”、“读者第一”的原则，为广大电脑编程爱好者、软件开发人员和专业计算机系统维护人员提供第一手技术资料、编程技巧和维护经验；紧密跟踪计算机软硬件技术发展和应用趋势，不断求变创新，针对软件开发过程中的许多关键技术问题着重提供各类解决方案，在业内获得一致好评，是广大编程和维护人员的首选刊物。在栏目内容上，选题覆盖面广，涉及技术领域宽、信息量大，帮助程序员开阔视野；在技术水平上，始终把握计算机技术发展的大方向，提供先进、详尽、准确的技术指导，并在长期工作中与国际性大公司建立了良好的合作关系，为读者提供全球最新、最全的实用信息；在实用性上，稿源来自于专业开发和维护人员的实践经验，是普通书籍难以获得的编程心得、体会与技巧。

2006 年是《电脑编程技巧与维护》创刊十二周年，为了最大限度地开发和利用本刊宝贵而丰富的资源，更好地服务和真诚回报多年来一直关爱和支持本刊的广大读者，《电脑编程技巧与维护》杂志社和中国水利水电出版社共同策划出版了本套“电脑编程实例导航丛书”。

本套丛书包括《Visual C/C++ 系统开发典型实例解析》、《Visual C/C++ 图形图像与游戏编程典型实例解析》、《Visual Basic 编程典型实例解析》、《Delphi 编程典型实例解析》、《C#编程典型实例解析》、《Java 编程典型实例解析》、《计算机系统安全与维护编程典型实例解析》、《计算机网络与通信编程典型实例解析》、《编程疑难问题解析 126 例》，一套 9 册共 680 多个典型实例。每册书的编程实例均依不同的编程应用分成若干章，条目清晰可查，使用极为方便。

这套丛书选编了《电脑编程技巧与维护》杂志近两年发表的和一部分尚未发表而又极为实用、精彩的典型编程实例。该套书的特点是：其各册内容来自编程高手的智慧和经验总结，其中不少文章的作者是业界资深程序员和技术专家，内容有深度、思路有新意、讲解深入浅出，编程技巧新颖实用，构思巧妙；丛书中的实例都是作者从实际项目提炼出的开发范例，实例讲解部分先给出设计目标，然后介绍实现目标的基本思想和方法，最后详细给出其核心程序的源代码，对程序的关键部分进行讲解并给出程序的运行效果；丛书中每一个实例

的程序源代码都经过上机调试通过，对编程中的疑难问题进行了深入解答，给程序开发人员移植源代码和学用编程带来了方便，加快了编程应用的步伐。全套书既讲究内容的深入性、专业性、权威性和实用性，同时兼顾轻松、通俗易懂、时效性强的特点。

本套丛书是《电脑编程技巧与维护》资源的二次深入开发，浓缩了当前主流编程语言 Visual C/C++、Visual Basic、Delphi、Java、C#等程序设计的精华，其目的是力求为读者建造一个真正的知识整合、编程思想、编程技术、技巧交流的平台，让读者从中学习到编程高手的诀窍，丰富编程技巧，拓宽编程思路，迅速提升程序开发能力。对电脑编程人员来说，程序开发能力的提高，除了对语言和算法的不断钻研学习、不断实践、不断总结提高，练好基本功，打好基础外，还要集思广益，善于学习，善于借鉴参考别人的经验，深入透彻地理解其中的精髓，然后融入到自己的设计方案中去，这无疑是一条有效的学习途径，对于自身编程能力的增强和编程水平的迅速提高十分重要，这也正是我们编写本套丛书想要达到的目的。

本套丛书可作为高等院校学生进行课程项目开发、毕业项目设计的参考书，也可作为软件从业人员及编程爱好者的珍藏宝典，还可作为高等培训学校的实例教程。

《电脑编程技巧与维护》杂志社

中国水利水电出版社

2006年5月

前 言

Visual C/C++ 是微软推出的一种面向对象的可视化编程工具，它作为 Microsoft Visual Studio 开发工具包中的一员，利用其提供的强大功能可以很容易地开发出功能完善、界面友好的 Windows 应用程序，是 Windows 系统下 32 位程序开发的理想工具。特别是 Visual C++ 6.0 版本问世以来，更是增加了许多功能，如代码重用、动态输入感应、对象模块化和组件共享等，可以大大提高软件系统设计的效率和速度，更易于实现大型软件系统的管理。

在《电脑编程实例导航丛书》中，Visual C/C++ 编程典型实例解析精选了《电脑编程技巧与维护》杂志近两年共 24 期已发表的精彩编程实例共 178 例。根据 Visual C/C++ 的不同应用对象，将其分为《Visual C/C++ 系统开发典型实例解析》和《Visual C/C++ 图形图像与游戏编程典型实例解析》两本书，介绍了使用 Visual C/C++ 开发应用程序关键技术的应用实例，每一个实例都给出了开发过程、技术难点及其解决的方法和技巧，涉及到 Visual C/C++ 应用程序设计的新思路、方法和技巧。这些典型实例所涵盖的编程技巧是编程高手经验的总结，具有一定的典型性和代表性，值得借鉴。本着实用原则，所有实例紧紧围绕一个主题展开，循序渐进、由浅入深地介绍使用 Visual C/C++ 进行应用程序开发的思想方法与编程技巧，在此我们愿与更多的 Visual C/C++ 编程爱好者分享这些宝贵经验。

本书分三章共计 56 个实例。第 1 章是图形图像、游戏编程基础与应用实例，该章使用 Visual C/C++ 进行图形图像、游戏编程基础与应用列举了 14 个典型而实用的编程实例，为图形图像处理应用和游戏编程奠定了基础。第 2 章是图形图像编程实例，该章精选了 34 个 Visual C/C++ 在图形图像处理应用中典型而实用的编程实例。第 3 章是游戏编程实例，该章精选了 8 个 Visual C/C++ 在游戏编程应用中典型而实用编程实例。

本书的特色体现如下几点：第一，每一章都通过一个个实例来介绍 Visual C/C++ 系统开发编程方法和技巧，避免枯燥、空洞的理论，并且每一个实例都具有很强的实用性和代表性。在实例的讲解上一般都是先给出设计目标，接着介绍实现该目标的基本思想和方法，然后详细给出其核心程序的源代码，对程序的关键部分进行讲解并给出程序的运行结果。第

二，所选的每一个实例都是从事 Visual C/C++ 系统开发编程人员的经验总结，具有很强的实用性，其中很多编程技巧可供借鉴。第三，每一个实例的程序源代码都经过上机调试通过，给程序开发人员移植源代码带来了方便，加快编程应用的步伐。

本书是《电脑编程技巧与维护》资源的二次开发，内容丰富，实用性强，所有实例中的源程序代码都经过了测试和使用，并给出了详细解释，浓缩了 Visual C/C++ 系统开发程序设计的精华，是系统开发编程人员的经验之作。

本书适合具有一定 Visual C/C++ 基础应用编程人员和系统开发人员，对初学 Visual C/C++ 进行系统开发编程的新手也有很好的参考价值。书中不足甚至疏漏之处在所难免，望读者和各位专家不吝赐教。

《电脑编程技巧与维护》杂志社

2006 年 5 月

目 录

丛书序

前 言

第 1 章 图形图像、游戏编程基础与应用实例

实例 1 Visual C 图像读写的编程实现	2
实例 2 Visual C 图像显示的编程实现	8
实例 3 Visual C 图像增强的编程实现	16
实例 4 Visual C 傅立叶变换的编程实现	26
实例 5 Visual C 图像快速傅立叶变换篇	34
实例 6 Visual C 图像界面设计的编程实现	44
实例 7 Visual C 图像边缘检测的编程实现	53
实例 8 图像的数字化艺术	63
实例 9 图库管理系统的设计	74
实例 10 三维地形实时动态显示的核心技术研究	78
实例 11 在 Visual C 中定制 DIB 类	82
实例 12 图论的两个实用算法的编程	85
实例 13 在 GDI + 中双缓存的使用	89
实例 14 基于 CMYK 图像的格式分析与读取实现	91

第 2 章 图形图像编程实例

实例 15 用 Visual C + + 处理灰度位图	95
实例 16 炭笔画效果	100
实例 17 Visual C 编程实时缩放及平移	104
实例 18 启动画面与鸣谢画面的编程	108
实例 19 Visual C 编程在状态条中加入位图	111
实例 20 用 ShadowMap 投射阴影	117
实例 21 Visual C + + 6.0 中实现运行程序图标在任务栏中的变换及去掉标题图标	121
实例 22 Visual C + + 中双缓存滚动视图类开发及应用	123
实例 23 利用内存多重复制实现多位图拖动显示	127
实例 24 Visual C 下 OpenGL 开发框架与应用	132

实例 25	OpenGL 中纹理贴图、滤波、光照和雾化效果初步编程	137
实例 26	基于 MFC 框架的 OpenGL 三维图形编程	142
实例 27	基于 OpenGL 编程实现三维虚拟现实	149
实例 28	在 Visual C 中将客户区图像保存到 BMP 位图文件	163
实例 29	Visual C + + 6.0 中实现矢量图的分层绘制、打印	167
实例 30	利用位图实现大数据量绘图快速显示	175
实例 31	基于 ObjectARX 2004 的任意复杂窗口剪裁技术	180
实例 32	Visual C + + 与 Matlab 混合编程实现卫星遥感影像的三维显示	185
实例 33	基于 Visual C + + 和 L 系统的自然景物模拟	188
实例 34	Visual C + + 6.0 下的图像处理及在多文档和无模态对话框中的实时显示	193
实例 35	增值税发票抵扣联字符识别中的图像倾斜校正方法	196
实例 36	用 GDI + 实现数字水印编程	199
实例 37	基于 GDI + 的分形图形绘制及处理	203
实例 38	自动纹理映射的实现	208
实例 39	用 Visual C 实现矢量图形的开发	209
实例 40	圆形半透明画笔功能的实现	215
实例 41	Visual C 编程实现多交互绘图程序的设计技术	223
实例 42	Visual C 中特效动画制作技巧	228
实例 43	用 Visual C 实现 AutoCAD 再次开发	240
实例 44	模仿深度剥离效果	246
实例 45	在 Visual C + + 中借助 DHTML 增加特效	248
实例 46	在 Visual C 下通过直方图变换对图像进行有效增强	251
实例 47	实现洒水车喷洒特殊效果三维仿真	256
实例 48	Visual C + + .NET 中基于 ADO 的数据库图像输入方法	259

第 3 章 游戏编程实例

实例 49	游戏开发中粒子系统分析	267
实例 50	3D 游戏中的粒子系统	273
实例 51	基于 MFC 开发三维动画	278
实例 52	简单骨骼动画的实现	284
实例 53	三维游戏中动态水面的模拟	289
实例 54	实现卡通渲染的简单方法	294
实例 55	基于 Symbian OS 的 S60 手机游戏图像处理	295
实例 56	用 Visual C + + 编游戏	300

第 1 章 图形图像、游戏编程 基础与应用实例

实例 1 Visual C 图像读写的编程实现

一、图像处理简介

视觉是人类从大自然获取信息的最主要手段。据统计,在人类获取的信息中,视觉信息约占 60%,听觉信息约占 20%,通过其他手段获取的信息约占 20%。“百闻不如一见”说的就是这个道理。由此可见视觉对人类的重要性,而图像正是人类获取视觉信息的主要途径。然而,直接获取的图像通常不能令人满意,为了满足人们视觉上或者心理上的需求,还要对图像进行加工,这就是图像处理(Image Processing)。现在是计算机时代,通常所说的图像处理主要是指数字图像处理——DIP(Digital Image Processing),也就是利用计算机对图像进行处理。

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的伦敦到美国的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、医学等领域的应用,图像处理技术得到广泛关注并迅速发展。从 20 世纪 60 年代起,随着计算机电子技术和计算机技术的发展,数字图像处理进入高速发展时期。目前,数字图像处理的应用越来越广泛,在医疗保健、资源调查、气象监测、航空航天、军事等领域都发挥了很大的作用。

一般来说,图像处理包括以下几个常见的方面:

(1) 点运算:主要是针对图像的像素进行加、减、乘、除等运算。点运算可以有效地改变图像的直方图分布。

(2) 几何处理:主要包括图像的坐标转换,例如:平移、缩放、旋转等。几何处理是最常见的图像处理手段。

(3) 图像增强:主要是为了突出图像中的重要信息,同时减弱或者除去次要信息。

(4) 正交变换:主要是把图像信息从空间域变换到频率域进行研究,例如:傅立叶变换、小波变换等。

(5) 图像复原:主要是去除干扰和模糊,从而恢复图像的本来面目。

(6) 图像重建:主要是利用采集的数据来重建出图像。

(7) 图像编码:主要是利用图像信息的统计特征和人类视觉系统的特性对图像进行编码,以达到压缩图像的目的。

除此之外,模式识别、形态学处理和分形算法也是当前研究的热点。当然,图像处理的内容还有很多,由于篇幅有限,就不在本实例进行讨论了。在后续的一系列实例中,会逐步地介绍图像处理的相关概念以及用 Visual C 实现的方法。

二、常见图像格式

在图像处理中,比较常见的图像格式有 BMP、JPG 和 TIFF 等,下面对这几类图像进行简要介绍。此外,还有其他许多类型的图像,例如:PNG、PCX、GIF 等,在此不作讨论。

1. BMP 图像

BMP 格式的图像在 Windows 系统比较常见,几乎所有的 Windows 上的应用软件都支持这种图像文件。Windows 中有两种位图,一种是 DDB(Device Dependent Bitmap)——设备相关位图,一种是 DIB(Device Independent Bitmap)——设备无关位图。由于 DIB 位图包含调色板信息,并且可以在不同的机器或者系统中显示位图所固有的图像,因此我们主要研究 DIB 位图。DIB 位图存储时通常以 BMP 为

后辍。

2. JPEG 图像

JPEG(Joint Photographic Experts Group)是由国际标准组织(ISO)和国际电报电话咨询委员会(CCITT)为静态图像所建立的第一个数字图像压缩标准。JPEG 压缩的效果比较好,能达到较高的压缩比。通过 JPEG 进行压缩后的图像信息可能有所丢失,但这是在人眼允许的范围内的。鉴于 JPEG 良好的压缩性能, JPEG 格式的图像在网络上非常流行。

3. TIFF 图像

TIFF(Tag Image File Format)图像可以在一个图像文件中存储多幅图像,除了文件头之外, TIFF 中的数据没有固定顺序。TIFF 图像的格式比较复杂,但是主要由文件头标志信息和图像数据组成。另外,在 TIFF 文件中还可以定义个人的标志信息。

三、BMP 格式读写

目前,微软的 Windows 操作系统已经占据了桌面操作系统的大部分市场。Visual C 作为该系统下的集成开发环境(IDE, Integrated Development Environment),其本身就是一个图形开发界面,而且 MFC(Microsoft Foundation Class)中封装了大量的位图操作,这给开发图像处理程序带来了很大方便。由于 BMP 图像格式在 Windows 中比较常见,因此我们主要以该格式的图像为例来讲解在 Visual C 下实现图像处理的基本方法。

首先对 BMP 图像的文件结构进行分析。在了解了该图像的结构之后,设计了一个 CBmp 类,并使用 Visual C 中封装的文件操作函数对 BMP 图像进行读写。

1. BMP 图像结构

BMP 格式的图像文件包含了一个 DIB, DIB 是标准的 Windows 位图格式。BMP 图像一般分为四部分:文件头、信息头、调色板和图像数据。值得注意的是有些 BMP 图像,例如 24 色(真彩色)图像不需要调色板,这种图像的信息头之后紧跟着的就是图像数据。BMP 图像的文件结构如图 1-1 所示。

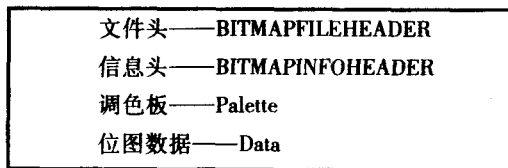


图 1-1 BMP 文件结构示意图

如果 Visual C 程序采用的是默认安装路径,那么在 c:\program files\microsoft visual studio\vc98\include\wingdi.h 中,可以找到除位图数据以外上面 3 个结构的定义。在分析这三个结构之前,先对这些结构定义中用到的数据类型进行一些说明。

在这些结构定义中,用到了 Visual C 中的几个数据类型,例如:DWORD, WORD 和 BYTE。在 c:\program files\microsoft visual studio\vc98\include\windef.h 中可以找到这几个类型的定义。编程初学者对于这几个类型代表的含义可能不是很清楚,而且经常和 C++ 的数据类型混淆。本实例作者在学习 Visual C 很长一段时间之后才弄清楚它们的意思。为了方便以后对文件结构的分析,现在让我们一起来看看这几个数据类型的庐山真面目。在 windef.h 中有如下代码:

```
typedef unsigned long   DWORD;  
typedef unsigned char  BYTE;  
typedef unsigned short WORD;
```

我们已经知道 long, char, short 是标准 C++ 中的数据类型,而 Visual C 中又定义了自己的数据类型。DWORD 就是 unsigned long, 占四个字节; BYTE 就是 unsigned char, 占一个字节; WORD 就是

unsigned short，占两个字节。用 DWORD(DOUBLE WORD)代替 unsigned long，用 BYTE 代替 unsigned char，这样不仅方便记忆而且编码量也减少了。要打出“unsigned char”需要十几个字符，而 BYTE 只需要四个字符！其实 Visual C 中还有很多自己定义的数据类型，感兴趣的读者可以用下面的方法来查看它们的来龙去脉，例如，想看看 RGBQUAD 的数据类型到底是什么，可以在 Visual C 中把鼠标移到 RGBQUAD，接着单击鼠标右键，从弹出菜单中选择 Go To Definition Of RGBQUAD，这时 IDE 就会自动定位到 RGBQUAD 的定义处。

下面开始进入正题，先来分析一下 BMP 图像的结构。

(1) 位图文件头。

在 wingdi.h 中有如下定义：

```
typedef struct tagBITMAPFILEHEADER {
    WORD bfType;
    DWORD bfSize;
    WORD bfReserved1;
    WORD bfReserved2;
    DWORD bfOffBits;
} BITMAPFILEHEADER, FAR * LPBITMAPFILEHEADER, * PBITMAPFILEHEADER;
```

这就是位图文件头的结构，它的长度是固定的。3 个 WORD 是 6 字节，2 个 DWORD 是 8 字节，共 14 个字节。各个域的说明如表 1-1 所示。

表 1-1 位图文件头结构说明

名称	偏移量	数据类型	长度	说明
bfType	0	WORD	2	ASCII 字符串“BM”
bfSize	2	DWORD	4	文件大小
bfReserved1	6	WORD	2	保留值，0
bfReserved2	8	WORD	2	保留值，0
bfOffBits	10	DWORD	4	到图像数据的偏移量

值得说明的是，所有的 BMP 图像文件的头两个字节都是 BM，也就是 0x424D。0x 前缀表示是 16 进制，0x42 就是字符“B”的 ASCII 码，0x4D 就是字符“M”的 ASCII 码。可以用 UltraEdit 打开任意一幅 BMP 图像，第一行的数据应该是以 42 4D 起始，如果不是，说明图像很有可能已经损坏。

(2) 位图信息头。

接下来是位图信息头，同样在 wingdi.h 中有如下定义：

```
typedef struct tagBITMAPINFOHEADER {
    DWORD biSize;
    LONG biWidth;
    LONG biHeight;
    WORD biPlanes;
    WORD biBitCount;
    DWORD biCompression;
    DWORD biSizeImage;
    LONG biXPelsPerMeter;
    LONG biYPelsPerMeter;
    DWORD biClrUsed;
    DWORD biClrImportant;
} BITMAPINFOHEADER, FAR * LPBITMAPINFOHEADER, * PBITMAPINFOHEADER;
```

位图信息头的大小也是固定的。LONG 也是 Visual C 中定义的数据类型，也就是 C++ 中的 long，它的长度是 4 个字节。5 个 DWORD 是 20 个字节，4 个 LONG 是 16 个字节，两个 WORD 是 4 个字节，

共 40 个字节。各个域的说明如表 1-2 所示。

表 1-2 位图信息头结构说明

名称	偏移量	数据类型	长度	说明
biSize	14	DWORD	4	该结构长度
biWidth	18	LONG	4	图像宽度，单位是像素
biHeight	22	LONG	4	图像高度，单位是像素
biPlanes	26	WORD	2	图像平面数，固定为 1
biBitCount	28	WORD	2	每个像素的位数
biCompression	30	DWORD	4	压缩类型
biSizeImage	34	DWORD	4	图像长度
biXPelsPerMeter	38	LONG	4	水平分辨率
biYPelsPerMeter	42	LONG	4	垂直分辨率
biClrUsed	46	DWORD	4	使用的色彩数
biClrImportant	50	DWORD	4	“重要”色彩数

表 1-2 中的偏移量是相对于文件起始位置的。这里要说明的是，biSizeImage 中指定的是图像数据实际占用的字节数。因为在 Win32 环境中为了使内存对齐，每一行字节数必须是 4 的整数倍（4 个字节恰好是 32 位）。以 256 色图像为例，如果该图像的 biWidth 是 255，则实际的图像宽度应该占 256 个字节。

(3) 调色板。

调色板实际上是一个数组（参考相关术语部分对调色板概念的说明），数组中的每个元素都是一个 RGBQUAD 结构。该结构在 wingdi.h 中有如下定义：

```
typedef struct tagRGBQUAD {
    BYTE  rgbBlue;
    BYTE  rgbGreen;
    BYTE  rgbRed;
    BYTE  rgbReserved;
} RGBQUAD;
typedef RGBQUAD FAR * LPRGBQUAD;
```

该结构的各个域的说明如表 1-3 所示。

表 1-3 RGBQUAD 结构说明

数据名称	偏移量	数据类型	说明
rgbBlue	0	BYTE	蓝色值
rgbGreen	1	BYTE	绿色值
rgbRed	2	BYTE	红色值
rgbReserved	3	BYTE	保留值

显然 RGBQUAD 结构占 4 个字节。这里要说明的是，对于真彩色图像是不需要调色板的，因此该类图像的信息头之后紧接着的就是图像数据。

(4) 图像数据。

如果图像中用到了调色板，那么图像数据中存储的是调色板中的索引值（即该像素值位于调色板中的第几行）；如果是真彩色图像，那么图像数据中存储的就是实际的 RGB 值。对于二值图像，用 1 位就可以表示该像素的颜色，所以 1 个字节可以表示 8 个像素；对于 16 色图像来说，用 4 位就可以

表示该像素的颜色, 所以 1 个字节可以表示 2 个像素。这里为了便于讲解原理, 把重点放在 256 色图像上, 它的 1 个字节恰好表示 1 个像素。

还要说明的是, BMP 文件中的图像数据是从图像的最下面一行开始的, 并且是按照从左到右的方式存储的。

2. BMP 图像读写

理解了 BMP 图像的结构之后, 编写 BMP 图像的读写程序就比较容易了。在 CBmp 类中定义了两个私有变量 m_pBMP 和 m_pBits。m_pBMP 是指向位图信息头的指针, m_pBits 是指向位图数据的指针。该类中涉及位图读写的函数是 Read, Write 和 Serialize。为了支持串行化, 让 CBmp 继承自 CObject, CBmp 类的头文件定义如下:

```
class CBmp: public CObject
{
public:
    DECLARE_SERIAL(CBmp)
    .....
    //Operations
    BOOL Read(CFile * pFile);
    BOOL Write(CFile * pFile);
    void Serialize(CArchive & ar);
    //Attributes
    int GetPaletteSize();    //调色板大小
    DWORD GetImageSize();   //图像大小
private:
    LPBITMAPINFOHEADER m_pBMP;    //位图信息头
    LPBYTE              m_pBits;   //位图数据
};
```

(1) Read 函数。

Read 函数主要是把位图文件从磁盘读入内存。在前面对 BMP 位图文件进行分析时已经知道, BMP 位图包括 4 个部分。因此在读取 BMP 位图时, 可以按照顺序把这 4 个部分读入内存。Read 函数的代码如下:

```
BOOL CBmp::Read(CFile * pFile)
{
    // 进行读操作
    try
    {
        BITMAPFILEHEADER bmfh;
        // 步骤 1 读取文件头
        int nCount = pFile->Read((LPVOID) & bmfh, sizeof(BITMAPFILEHEADER));
        // 判断是否是 BMP 格式的位图
        if(bmfh.bfType != BMP_HEADER_MARKER)
        {
            throw new CException;
        }
        // 计算信息头加上调色板的大小并分配内存
        int nSize = bmfh.bfOffBits - sizeof(BITMAPFILEHEADER);
        m_pBMP = (LPBITMAPINFOHEADER) new BYTE[nSize];
        // 步骤 2 读取信息头和调色板
        nCount = pFile->Read(m_pBMP, nSize);
        // 步骤 3 读取图像数据
        m_pBits = (LPBYTE) new BYTE[m_pBMP->biSizeImage];
```

```

        nCount = pFile -> Read(m_pBits, m_pBMPH -> biSizeImage);
    }
    catch(CException * pe)
    {
        AfxMessageBox("Read error");
        pe -> Delete();
        return FALSE;
    }
    return TRUE;
}

```

为了判断所读文件是否是 BMP 格式, 预先定义一个宏来作为 BMP 文件头的标志。

```
#define BMP_HEADER_MARKER ((WORD) ('M' << 8) | 'B')
```

在读取的过程中, 可以和这个标志进行比较来确定图像是否是 BMP 格式。Read 函数用到了 sizeof 运算符, 它可以计算某种类型所占的字节数, 例如, sizeof(BITMAPFILEHEADER) 的值是 14。通过调用 Read 函数, 把图像文件读到内存中, 并把 m_pBMPH 指向图像文件的信息头, 而 m_pBits 指向位图数据。

(2) Write 函数。

Write 负责把内存中的数据写入图像文件中。按照 BMP 图像的文件格式, 依次把文件头、信息头、调色板和图像数据写入文件中。

```

BOOL CBmp::Write(CFile * pFile)
{
    BITMAPFILEHEADER bmfh;
    // 设置文件头中文件类型 0x424D = "BM"
    bmfh.bfType = BMP_HEADER_MARKER;
    // 计算信息头和调色板大小
    int nSizeHeader = sizeof(BITMAPINFOHEADER) + sizeof(RGBQUAD) * GetPaletteSize();
    // 设置文件头信息
    bmfh.bfSize = sizeof(BITMAPFILEHEADER) + nSizeHeader + GetImageSize();
    bmfh.bfReserved1 = 0;
    bmfh.bfReserved2 = 0;
    // 计算偏移量 文件头大小 + 信息头大小 + 调色板大小
    bmfh.bfOffBits = sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER) + sizeof(RGBQUAD)
    * GetPaletteSize();
    // 进行写操作
    try
    {
        pFile -> Write((LPVOID) &bmfh, sizeof(BITMAPFILEHEADER));
        pFile -> Write((LPVOID) m_pBMPH, nSizeHeader);
        pFile -> Write((LPVOID) m_pBits, GetImageSize());
    }
    catch(CException * pe)
    {
        pe -> Delete();
        AfxMessageBox("write error");
        return FALSE;
    }
    return TRUE;
}

```

(3) Serialize 函数。

为了使文档支持串行化, 添加了 Serialize 函数, 并且要在 CBmp 类的 .H 文件中添加宏 DE-

CLARE_SERIAL(CBmp), 在 .CPP 文件中添加宏 IMPLEMENT_SERIAL(CBmp, CObject, 0); 下面是串行化的代码。

```
// 串行化
void CBmp::Serialize(CArchive & ar)
{
    if(ar.IsStoring())
    {
        Write(ar, GetFile());
    }
    else
    {
        Read(ar, GetFile());
    }
}
```

(4) 创建 ImageTest 工程。

利用 MFC AppWizard (EXE) 创建了一个多文档的工程 ImageTest。在 CImageTestApp 类中添加了 ID_FILE_OPEN 的消息——OnFileOpen。在 Doc 类中包含类 CBmp 的头文件, 并添加一个变量 m_pBmp, 它是 CBmp 类的指针。在 Doc 类的构造函数中对它进行初始化。

```
m_pBmp = new CBmp;
```

然后, 在 Doc 类的 Serialize 函数中添加如下代码:

```
m_pBmp->Serialize(ar);
```

现在, 可以测试一下所编写的函数是否成功。

因为现在还没有实现图像显示的功能, 所以图像并不能显示出来。运行上面的程序, 打开标准 Lena 图像。此时, 可以单击“另存为”, 将图像以文件名 Lena1 保存。这时, 用 Windows 自带的图片浏览器可以看到 Lena1 与原图像 Lena 是一样的, 这就验证了我们编写的函数是可行的。

四、小结

本实例讨论了图像处理的基本概念, 并用 Visual C 实现了 BMP 图像读写功能。但是, 对于图像处理来说, 这只是万里长征的开始。接下来, 会进一步介绍如何显示图像, 如何对图像进行处理等内容。编程环境为 Visual C 6.0, 实验所用图像保存在 ImageTest 子文件夹 Data 中。

(耿 迅)

实例 2 Visual C 图像显示的编程实现

掌握了 BMP 图像的读写方法, 这仅仅是把图像文件读入内存, 还没有看到图像的显示。本实例讨论图像显示的技术, 开发环境仍然是 Windows + VC。

一、相关术语

在图像处理中, 可能会经常遇到一些专业词汇。为了方便交流, 先来了解一下术语。

(1) 像素: 数字化图像是对物理图像进行采样, 把物理图像划分为若干个采样方格, 这些小的采样方格就是像素。通常所说的屏幕分辨率为 1024×768 , 也就是说屏幕上每行有 1024 个像素, 共有 768 行。