

数字图像

处理方法及程序设计

陆 玲 周书民◎著

HEU 哈尔滨工程大学出版社
Harbin Engineering University Press

数字图像处理方法及程序设计

陆 玲 周书民 著

哈尔滨工程大学出版社

内 容 简 介

本书主要介绍数字图像处理的一些常用方法及应用,以及相应方法的 VC ++ 6.0 程序设计。常用方法主要包括 BMP 图像文件的存取方法、彩色图像的特效处理方法、灰度图像的平滑与锐化、灰度图像的投影、灰度图像的边缘检测、二值图像的形态学处理、二值图像的几何特征与形状特征。应用实例包括彩色人像的美容与美发处理、人像的人眼定位处理、手写字体的识别、车牌的定位及车牌号码的识别。

本书的特点是所有方法都附有相应的 VC ++ 源程序代码,且含有 VC ++ 6.0 的编程过程。

本书可作为高等学校计算机科学与技术专业、电子信息类专业的本科生和研究生教材,也可作为从事图像处理工作工程技术人员的参考书。

图书在版编目(CIP)数据

数字图像处理方法及程序设计/陆玲,周书民著.
—哈尔滨:哈尔滨工程大学出版社,2011.12
ISBN 978-7-5661-0283-6

I. ①数… II. ①陆… ②周… III. ①数字图像处理—程序设计 IV. ①TN911.73

中国版本图书馆 CIP 数据核字(2011)第 244742 号

出版发行 哈尔滨工程大学出版社
社 址 哈尔滨市南岗区东大直街 124 号
邮政编码 150001
发行电话 0451-82519328
传 真 0451-82519699
经 销 新华书店
印 刷 黑龙江省地质测绘印制中心
开 本 787mm × 960mm 1/16
印 张 10
字 数 202 千字
版 次 2011 年 12 月第 1 版
印 次 2011 年 12 月第 1 次印刷
定 价 19.8 元
<http://press.hrbeu.edu.cn>
E-mail: heupress@hrbeu.edu.cn

前 言

数字图像处理技术在各个行业里都有着广泛的应用。随着科学技术的不断发展以及人们对图像信息的不断需求,数字图像信息及其处理技术正在发挥着越来越重要的作用。数字图像处理已经在计算机科学、信息科学、生物学、统计学、气象学、工程学、物理学、医学等领域里得到广泛的重视和应用,并正在向传统的学科甚至社会科学等领域不断地渗透。在信息社会中,数字图像处理科学无论是在理论上还是实践上都存在着巨大的发展潜力。

本书是作者在多年研究的基础上,总结过去的一些研究成果,结合教学经验和已出版的相关教材,经过整理与汇总编写而成的。

本书首先介绍了 BMP 图像文件格式及存取的程序设计方法,为后续的图像处理作准备;然后对常用的图像预处理方法进行介绍,主要包括彩色图像向灰度图像及灰度图像向二值图像的转换、灰度直方图及灰度变换、图像的平滑与锐化、二值图像的形态学处理等;最后分别介绍彩色图像、灰度图像及二值图像的常用处理方法及应用实例。应用实例包括人像美容与美发处理,人像的人眼定位处理、手写字体的识别、车牌的定位及车牌号码的识别。这些实例所使用的方法都是作者提出的经验方法,虽然并不一定是最佳方法,但是简单易行,特别适用于数字图像处理的初学者,可以起到抛砖引玉的作用。

本书力求体现以下特点:

(1)内容通俗易懂 对读者的要求起点低,只要求读者有 C 语言程序设计的基础。读者通过本书的学习,可掌握数字图像处理常用的基本方法。

(2)重点难点突出 侧重介绍形成数字图像处理基本方法的思路及实现算法。

(3)程序设计完整 本书涉及的图像处理方法都配有相应的 VC++6.0 程序设计过程及源代码,使读者能够掌握图像处理的编程方法。

(4)处理结果展示 每个图像处理方法都附有相应程序的运行结果图,使读者对图像处理的方法有直观的认识。

本书在编写和出版的过程中,得到了很多朋友的支持与帮助,在此一并表示感谢。

由于作者水平有限,加之时间仓促,书中难免有不妥和错误之处,恳请广大读者批评指正。

著 者
2011 年 11 月

目 录

第 1 章 数字图像文件的存取	1
1.1 BMP 图像文件的格式	1
1.2 BMP 文件的读取	3
第 2 章 数字图像预处理	15
2.1 彩色图像转灰度图像	15
2.2 灰度直方图处理	22
2.3 灰度变换	27
2.4 图像的平滑处理	34
2.5 图像的锐化处理	39
2.6 灰度图像转二值图像	42
2.7 二值图像的形态学处理	46
第 3 章 彩色图像处理及应用	61
3.1 颜色模型	61
3.2 彩色图像的特效处理	64
3.3 人像美容处理	72
3.4 人像美发处理	82
第 4 章 灰度图像处理及应用	96
4.1 灰度图像的投影	96
4.2 图像的边缘检测	99
4.3 人眼定位处理	107
4.4 车牌定位处理	117
第 5 章 二值图像处理及应用	124
5.1 二值图的几何特征	124
5.2 二值图的形状特征	133
5.3 手写字体的识别	135
5.4 车牌号码识别	142
参考文献	152

第 1 章 数字图像文件的存取

数字图像(以下简称图像)有多种存储格式,每种格式一般由不同的开发商支持。随着信息技术的发展和图像应用领域的不断拓宽,还会出现新的图像格式。因此,要进行图像处理,必须了解图像文件的格式,即图像文件的数据构成。每一种图像文件均有一个文件头,在文件头之后才是图像数据。文件头的内容由制作该图像文件的公司决定,一般包括文件类型、文件制作者、制作时间、版本号、文件大小等内容。各种图像文件的制作还涉及到图像文件的压缩方式和存储效率等。本章将介绍未经压缩的 24 位真彩 BMP 图像的存取方法。

1.1 BMP 图像文件的格式

BMP 类型图像文件又称位图文件,它是 Windows 采用的图像文件存储格式,在 Windows 环境下运行的所有图像处理软件都支持这种格式。BMP 位图文件默认的文件扩展名是 BMP 或者 bmp。

位图文件可看成由 4 个部分组成:位图文件头(bitmap-file header)、位图信息头(bitmap-information header)、颜色表(color table)和位图数据。

1. 位图文件头

文件头的信息主要包括与文件相关的信息,如文件类型、文件大小等,该部分信息共占 14 个字节。其具体结构如表 1-1:

表 1-1 位图文件头结构

起始字节	所占字节数	具体内容	标志
1	2	文件类型(Windows 位图为“BM”)	bfType
3	4	文件大小	bfSize
7	4	保留	bfReserved
11	4	第一个位图数据的偏移量(一般为 54)	bfOffBits

2. 位图信息头

位图信息头主要包括与图像相关的信息,如图像的高度与宽度,图像的大小等,该部分信息共占 40 个字节。其具体结构如表 1-2:

表 1-2 位图信息头结构

起始字节	所占字节数	具体内容	标志
15	4	位图信息头的长度(一般为 40)	biSize
19	4	位图的宽度	biWidth
23	4	位图的高度	biHeight
27	2	位图的位面数(=1)	biPlanes
29	2	每个像素所占位数(=24 真彩图)	biBitCount
31	4	位图压缩类型(=0 未压缩)	biCompression
35	4	图像的大小(以字节为单位,必须是 4 的倍数)	biSizeImage
39	4	位图水平分辨率(像素/米)	biXPelsPerMeter
43	4	位图垂直分辨率(像素/米)	biYPelsPerMeter
47	4	位图实际使用的颜色数 (=0 使用所有颜色)	biClrUsed
51	4	指定重要的颜色数(=0 都重要)	biClrImportant

3. 颜色表

该部分含有多个表项,每个表项占 4 个字节。24 位真彩图像是没有这部分内容的。

4. 位图数据

对于 24 位真彩图,位图数据存储图像中每个像素的 RGB 颜色值(例如,白色 $R = G = B = 255$,黑色 $R = G = B = 0$,红色 $R = 255$ 和 $G = B = 0$)。一个像素占 3 个字节,每个字节分别表示 R, G, B 三分量的值,且存放顺序是 B, G, R 。

值得注意的是,像素的存放顺序是从图像中的最后一行到第一行,每一行的顺序是从左到右。并且,每一行像素所占的字节数必须是 4 的整倍数,如果实际像素所占字节数不足 4 的倍数,则多余的字节不存像素颜色,下一行像素值从 4 的倍数字节处后开始存放。

例如,对于图 1-1 的 24 位真彩图像,其文件中各字节的存储内容如图 1-2。

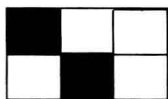


图 1-1 24 位真彩图像

字节起始位	1	3	7	11	15	19	23	27	
存储值	BM	78		54	40	3	2	1	
字节起始位	29	31	35	39	43	47	51		
存储值	24	0	24	0	0	0	0		
字节起始位	55	56	57	58	59	60	61	62	63 64-66
存储值	255	255	255	0	0	0	255	255	255
字节起始位	67	68	69	70	71	72	73	74	75 76-78
存储值	0	0	0	255	255	255	0	0	0

图 1-2 图像内容

1.2 BMP 文件的读取

在 VC++ 中使用 MFC 方式编写图像文件的应用程序。

1.2.1 VC++ 应用程序框架的生成

(1) 创建一个图像应用程序

进入 Visual C++ 集成开发环境后,选择“文件|新建”菜单,弹出“新建”对话框,单击“工程”标签,打开其选项卡,在其左边的列表框中选择 MFC AppWizard (exe) 工程类型,在“工程名称”文本框输入工程名 Image,在“位置”文本框输入工程路径。如果是第一个工程文件,则必须创建一个新的工作区,选择“创建新的工作空间”,如图 1-3 所示。

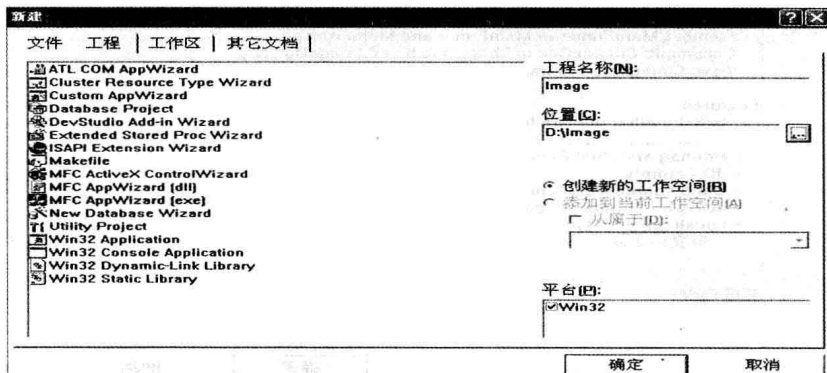


图 1-3 新建工程对话框

(2) 单击“确定”按钮后,系统显示“MFC 应用程序向导 - 步骤 1”对话框,选择“单文档”选项,如图 1-4 所示。

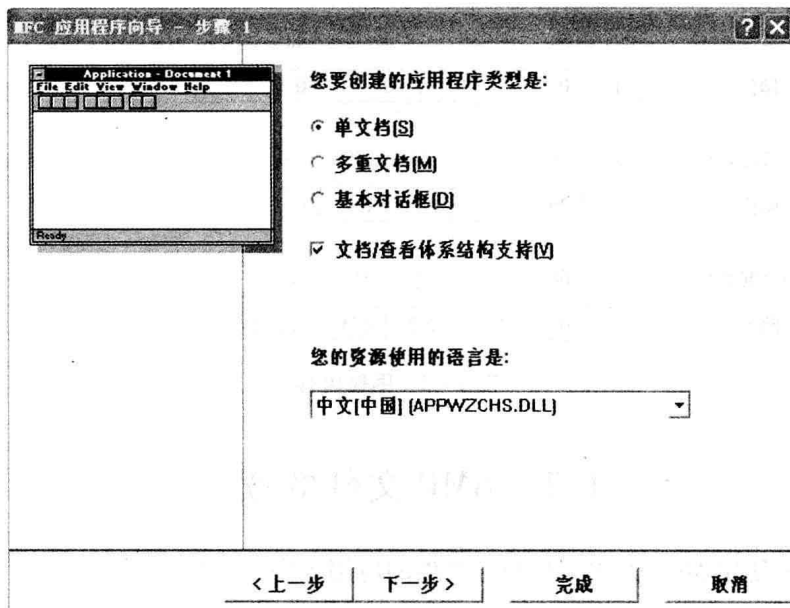


图 1-4 MFC 应用程序向导 - 步骤 1 对话框

(3) 单击“完成”按钮,系统弹出如图 1-5 所示“新建工程信息”对话框。

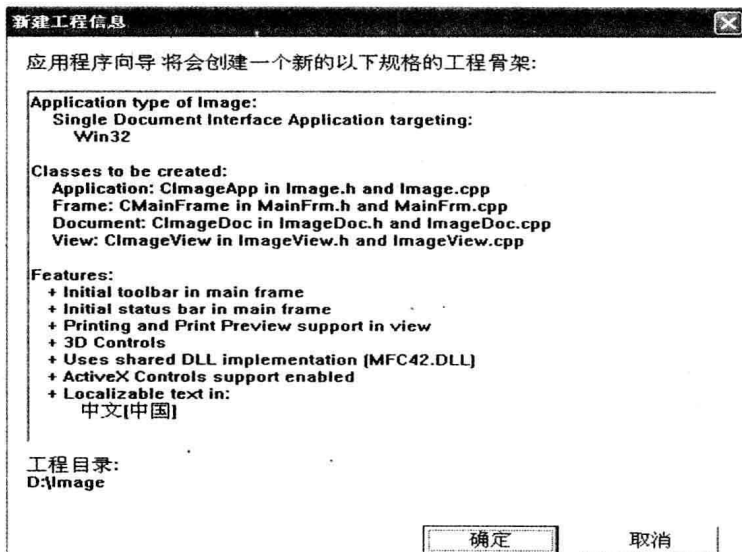


图 1-5 新建工程信息对话框

(4) 单击“确定”按钮后,就完成了应用程序大体框架的自动生成,在指定的目录下生成了应用程序框架所必需的全部文件,如图 1-6 所示。

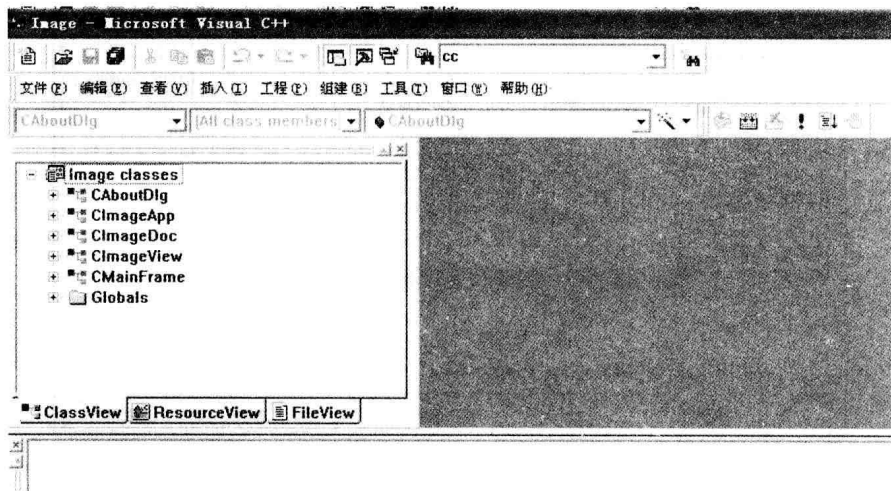


图 1-6 应用程序编辑界面

图 1-6 界面左栏中的 ClassView 选项卡显示了该程序框架的四个大类: CimageApp(应用类)、CimageDoc(文档类)、CimageView(视图类)和 CimageFrame(框架类)。我们主要使用文档类(用于图像文件的打开与存储)与视图类(用于图像处理及显示)。

1.2.2 图像文件的打开

选择主菜单中“查看”下的“建立类向导”,如图 1-7 所示。在 Message Maps 标签下,进行如下操作:

(1) 在 Class name 区选择 CimageDoc(文档类),表示在文档类中设计成员函数,实现打开图像文件的功能。

(2) 在 Message 区双击 OnOpenDocument(该消息映射“打开文件”菜单,同时激活“打开文件对话框”),这时在 Member function 区内出现 OnOpenDocument 成员函数名。该成员函数是响应 Message 区中 OnOpenDocument 消息的函数。

(3) 单击“Edit Code”按钮,进入代码编辑界面(如图 1-8 所示)。在“// TODO: Add your specialized creation code here”处可编写打开图像文件的程序代码。注意,该成员函数中的参数 lpszPathName 就是用户在“打开文件对话框”中选择的文件名。

在“// TODO: Add your specialized creation code here”下方输入如下代码:

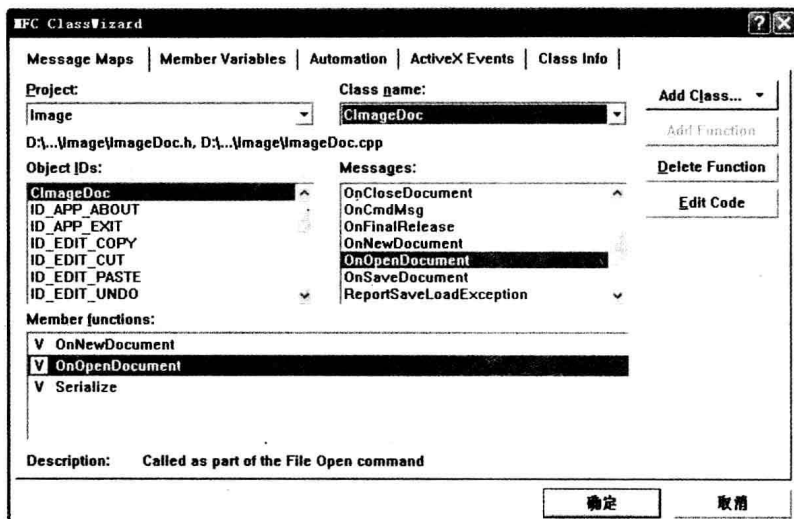


图 1-7 建立类向导界面

```

{
    CDocument::Dump(dc);
}
#endif // _DEBUG

// CImageDoc commands

BOOL CImageDoc::OnOpenDocument(LPCTSTR lpszPathName)
{
    if (!CDocument::OnOpenDocument(lpszPathName))
        return FALSE;

    // TODO: Add your specialized creation code here

    return TRUE;
}

```

图 1-8 代码编辑界面

```
int x,y,k;
```

```
FILE *fp = fopen(lpszPathName,"rb"); //以二进制的方式打开文件对话框中的图像文件
```

```
fread(bfType,2,1,fp); //读取从第 1 个字节开始的 2 个字符读 1 次存入 bfType 中
```

```
if (bfType[0] == 'B' && bfType[1] == 'M') //如果文件类型为"BM",则该文件是 BMP 文件
```

```
{ fread(&bf_h,12,1,fp); //读文件信息头
```

```

fread(&bi_h,40,1,fp); //读图像信息头
if (bi_h.biBitCount == 24) //像素位数等于 24,是 24 位真彩图,可以进行处理
{
    flag = 1; //24 位真彩图标志量赋值
    for (y = bi_h.biHeight - 1; y >= 0; y--) //图像从最后一行开始存储,因此先读的存在数组后面
    {
        for (x = 0; x < bi_h.biWidth; x++)
        //分别读(x,y)处像素的 B、G、R 三个颜色分量分别存入 pic[y][x][2]、pic[y][x][1]、pic[y][x][0]中
        for(k = 2; k >= 0; k--)
            fread(&pic[y][x][k],1,1,fp);
        if(((bi_h.biWidth * 3) % 4) != 0)
            fseek(fp,4 - (bi_h.biWidth * 3) % 4,1); //跳过无图像数据的字节
    }
}

fclose(fp); //关闭文件

```

上述程序中未定义的变量或数组须在文档类中定义,具体操作如下。

(1) 在程序编辑界面(图 1-6)左区选择 FileView 选项卡,则列出本工程文件中所有程序文件,如图 1-9 中的左区域。

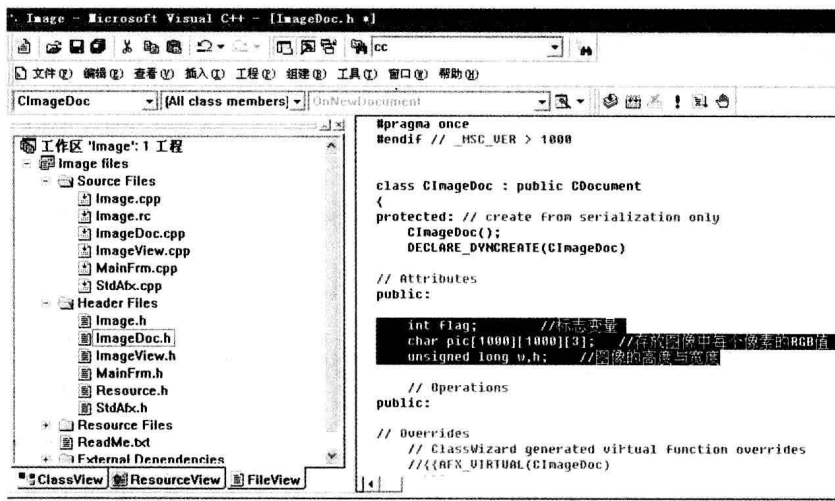


图 1-9 文档类头文件代码编辑界面

(2) 选择 ImageDoc.h(文档类头文件),右区域出现该程序文件的所有代码,其中就包括文档类的定义,在文档类的定义中,添加公有成员变量如下(如图 1-9 右区域中的选取部分):

```

char bfType[2];           //存放图像类型标志
struct bf_head            //定义文件信息头结构体
{
    long int bfSize;       //文件大小
    long bfReserved;
    long bfOffBits;       //第一个位图数据的开始字节位置
    bf_h;
    struct bi_head        //定义图像信息头结构体
    {
        long biSize;      //图像大小
        long biWidth;     //图像高度
        long biHeight;    //图像宽度
        short int biPlanes; //位图的位面数
        short int biBitCount; //每个像素所占位数
        long biCompression; //位图压缩类型
        long biSizeImage;   //图像的大小
        long biXPelsPerMeter; //位图水平分辨率
        long biYPelsPerMeter; //位图垂直分辨率
        long biClrUsed;     //位图实际使用的颜色数
        long biClrImportant; //指定重要的颜色数
    } bi_h;
    BYTE pic[1000][1000][3]; //存放图像中每个像素的 RGB 值
    int flag;                //标志变量
}

```

1.2.3 图像文件的显示

图像文件打开后存储在文档类中的 pic 数组里,如果要在屏幕中显示出来,可设计通过菜单选择在视图类中通过视图区输出。其具体过程如下:

(1) 单击图 1-9 左区中的“ResourceView”选项卡,选择“Menu”,双击其下的“IDR_MAIN-FRAME”,如图 1-10 所示,进入菜单编辑界面。

(2) 选择右边菜单栏的空栏并单击右键,点击“属性”,弹出如图 1-11 所示对话框,输入一级菜单内容“图像处理”,该级菜单只是显示提示内容。

(3) 单击“图像处理”菜单的子菜单空栏处再单击右键,选择“属性”选项,对如图 1-12 对话框,设置其子菜单属性。

(4) 在菜单编辑界面处点击右键,选择“建立类向导”选项,操作如图 1-13 所示。

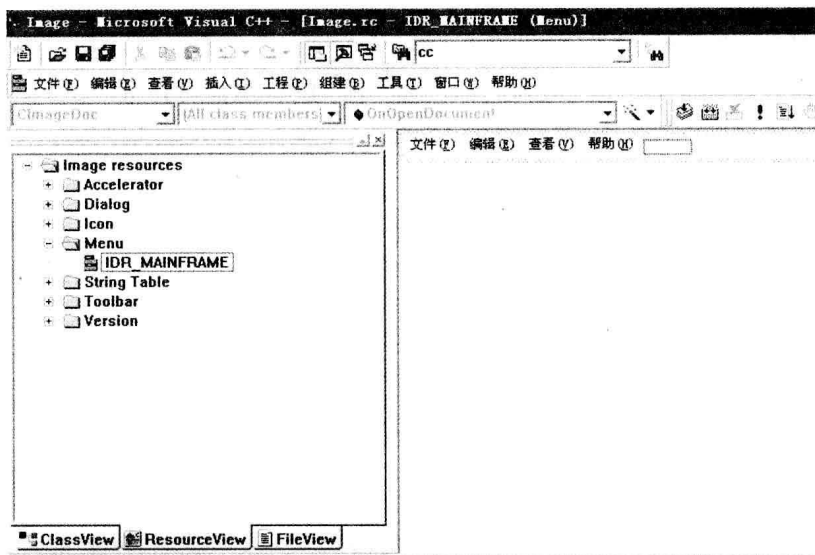


图 1-10 菜单编辑界面

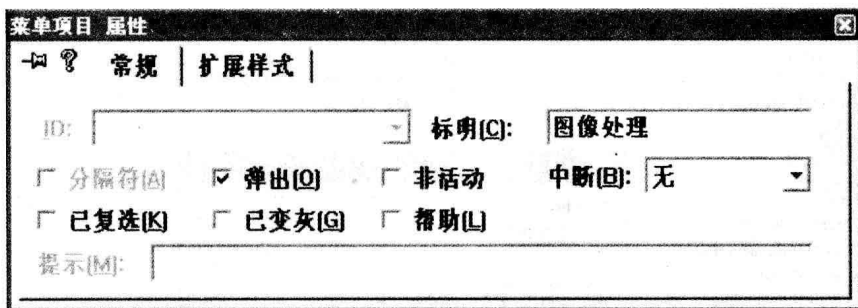


图 1-11 菜单项目属性对话框

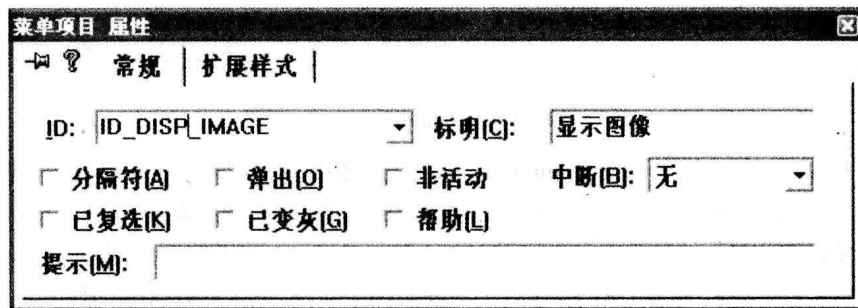


图 1-12 子菜单属性设置

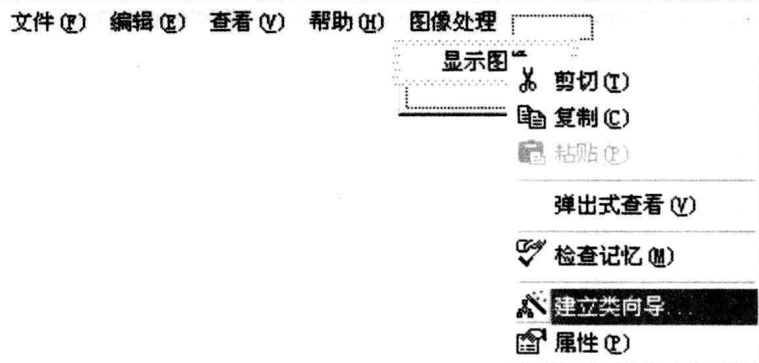


图 1-13 建立类向导

(5) 在弹出的类向导对话框中,进行如图 1-14 所示的设置。

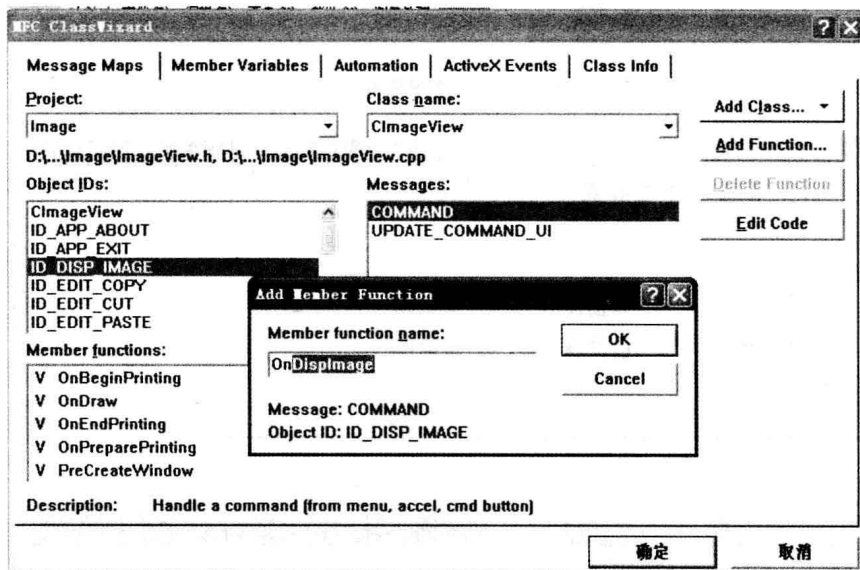


图 1-14 MFC ClassWizard 对话框

如图 1-14 中的“Add Member Function”对话框,是选择“Add Function”按钮时出现的,选择该对话框中的“OK”按钮后,在“MFC ClassWizard”对话框的 Member Function 区内出现 OnDispImage 成员函数名。该成员函数响应 ID_DISP_IMAGE 菜单项单击事件。

(6) 点击“Edit Code”按钮(图 1-15),进入如图 1-16 所示代码编辑界面。

在“// TODO: Add your command handler code here”下方输入如下代码:

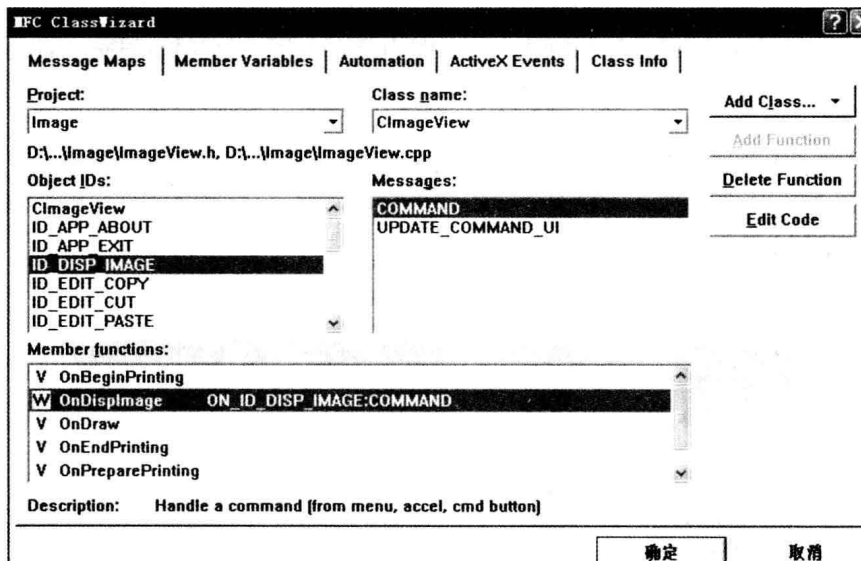


图 1-15 MFC ClassWizard 对话框生成成员函数

```

        ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CImageDoc))
        return (CImageDoc*)m_pDocument;
    }
#endif // _DEBUG

////////////////////////////////////
// CImageView message handlers

void CImageView::OnDispImage()
{
    // TODO: Add your command handler code here
}

```

图 1-16 视图类文件代码编辑界面

```

CImageDoc * pDoc = GetDocument(); //pDoc 指向文件类对象
CDC * pDC = GetDC(); //获取输出设备指针
if(pDoc->flag == 1) //读取文件成功
    { for(int y=0;y < pDoc->bi_h.biHeight-1;y++)

```

```

for(int x=0;x<pDoc->bi_h.biWidth-1;x++)
    pDC->SetPixel(x,y,RGB(pDoc->pic[y][x][0],pDoc->pic[y][x][1],pDoc->pic[y][x][2]));
}
else
    MessageBox("不能打开图像文件");

```

程序运行实例如图 1-17、图 1-18、图 1-19。

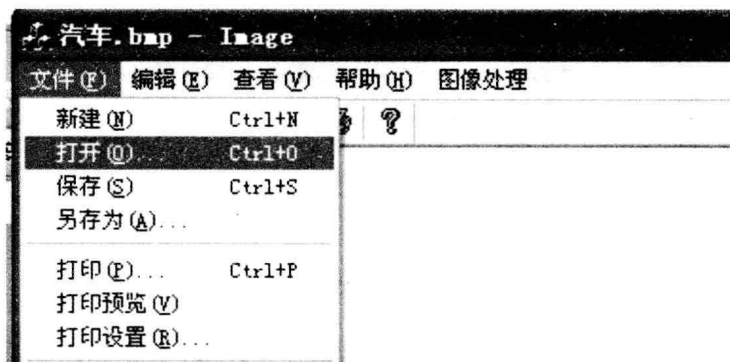


图 1-17 打开文件界面

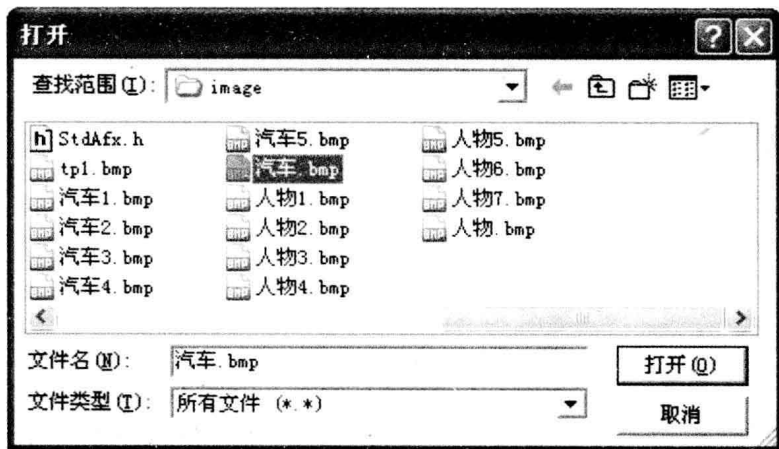


图 1-18 打开文件对话框界面