

直方图均衡化处理编程指南

本程序将在 Microsoft Visual C++6.0 开发环境下构建，所以，在开始之前，请确定您所使用的计算机已经正确安装了 Microsoft Visual C++6.0（具体的安装方法请详见 Microsoft Visual C++6.0 用户手册，低于 6.0 的版本是否可以运行在此不做讨论）。下面，请用该指南构建一个直方图均衡化处理的程序。

一、创建对话框资源

利用 Microsoft Visual Studio 的对话框编辑器来设计对话框并创建对话框资源。步骤如下：

1、Visual C++6.0 启动步骤：

- ① 安装好 Visual C++6.0；
- ② 按路径开始——程序——Microsoft Visual Studio 6.0——Microsoft Visual C++ 6.0 启动 VC；
- ③ 出现 VC 界面后，按路径“File---New”进入创建项目对话框；

2、在弹出的对话框中单击 Projects 选项卡，并单击 MFC AppWizard (exe)，在右侧的 Project name 一栏中输入工程文件名，在此我们输入 HistDemoA（读者可以输入自己的工程文件名，但为了统一起见，建议初学者输入和本文相同的名字，以便于后面的理），如下图所示：

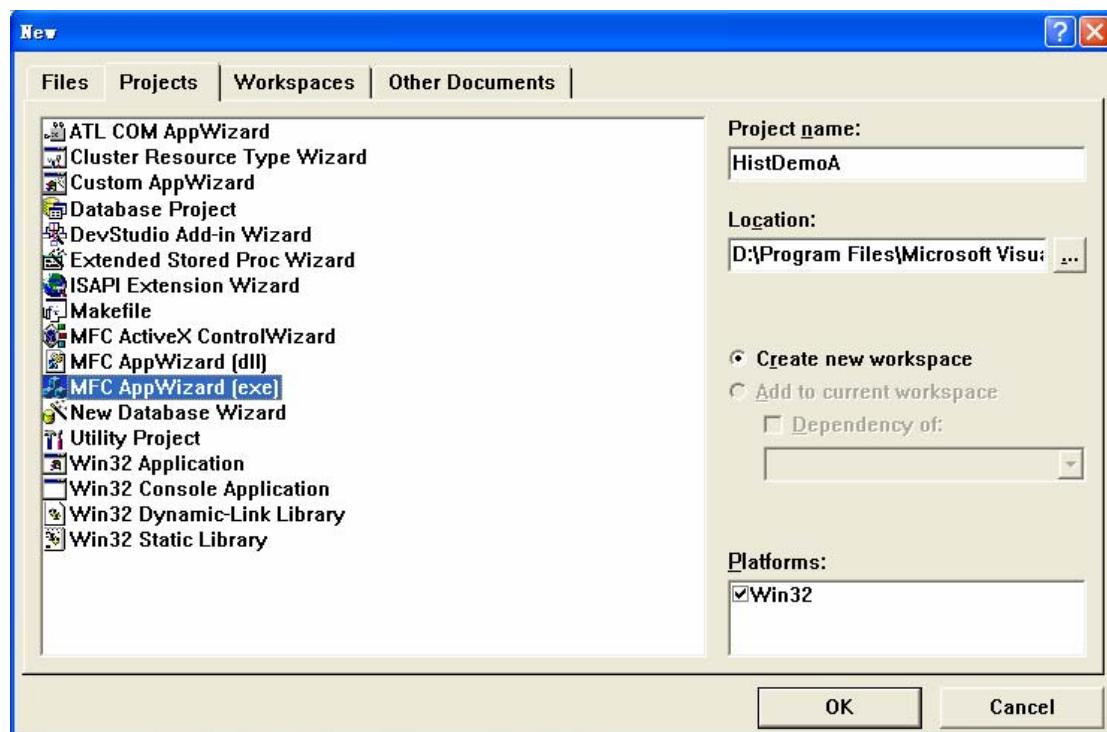


图 1

完成后，单击 OK 按钮。

3、在弹出的 MFC AppWizard 中，选择 Dialog based，如下图所示，

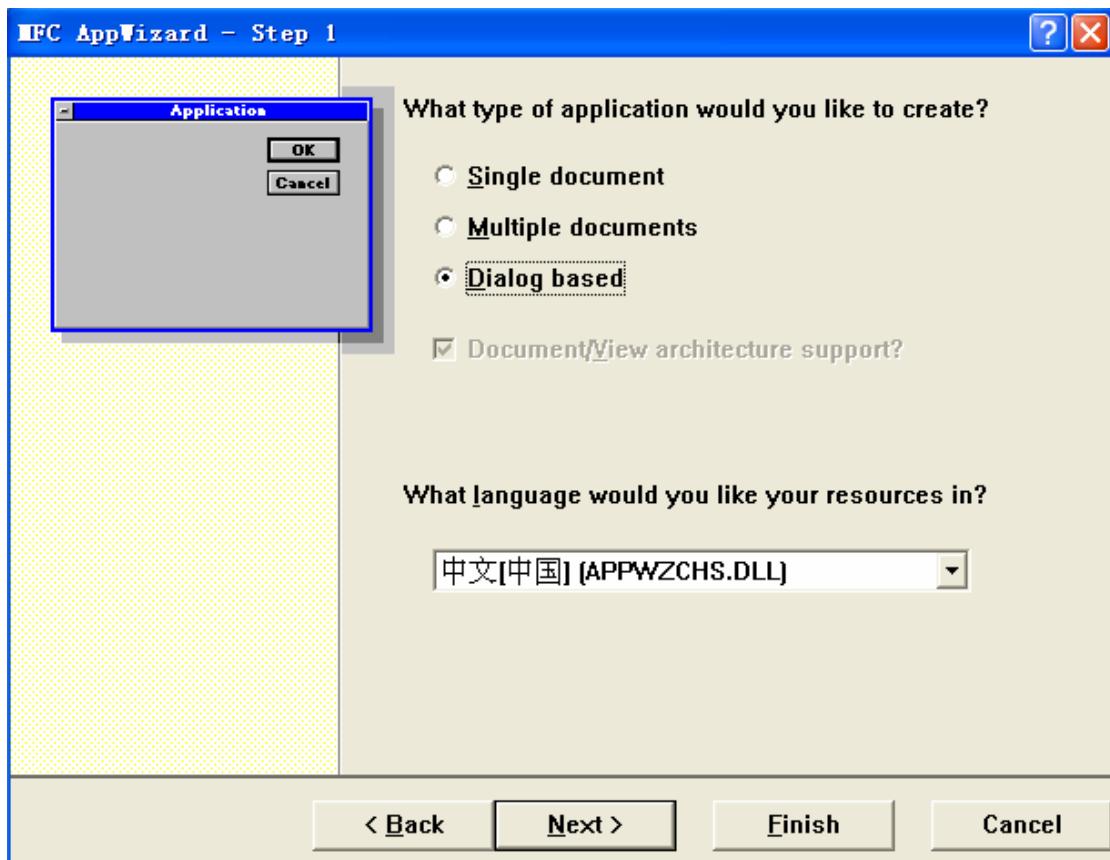


图 2

其它各项均接受默认的设置。生成对话框应用程序界面如下所示：

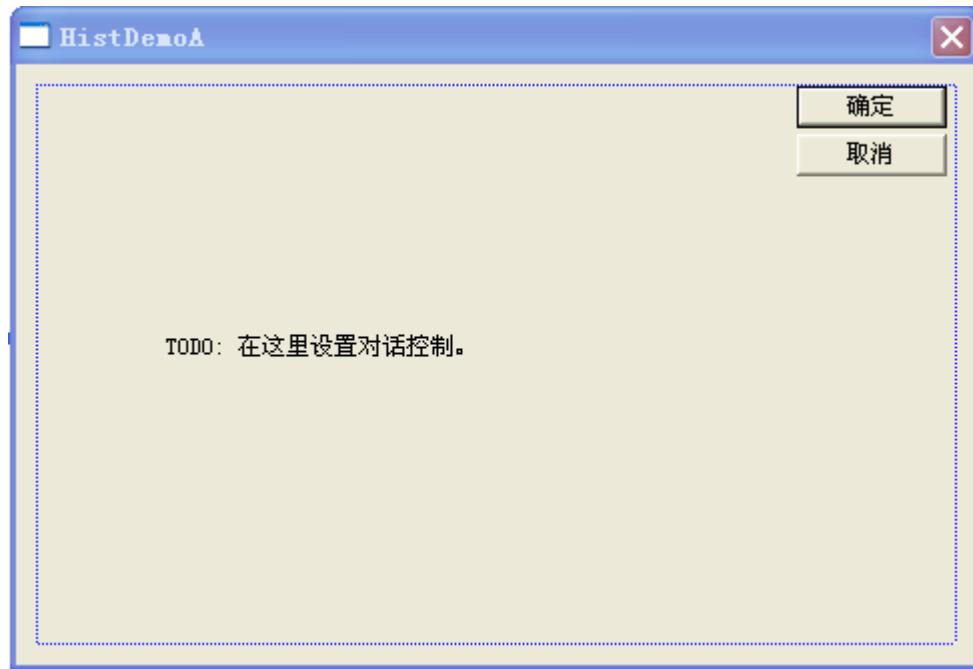


图 3

- 4、删除“取消”按钮及“TODO: 在这里设置对话框”静态文本框。右键单击“确定”按钮，在弹出的菜单中选择 Properties，在弹出的按钮属性对话框中将“确定”按钮的 Caption 改为“关闭”。

5、改变对话框的大小。用鼠标直接拖拉就可以调整这个对话框的大小。在这里，我们通过鼠标的拖拉，将对话框的大小改为 500×270 （为了后面能合适地显示图像。本程序在 1024×768 的分辨率下调试，如果读者使用不同的分辨率，请根据具体情况做相应的调整），这个大小将在窗口右下侧的状态栏中显示出来。

6、添加对话框控件。通过控件调色板来添加控件（如果控件调色板不可见的话，在工具栏上单击右键，从列表中选择 Controls 即可）。可以把控件直接从控件调色板上拖拉到新的对话框中，然后放到合适的位置，并且改变控件大小。在这里，我们添加两个静态文本框，右键单击 Properties，将它们的 Caption 分别设置为：“原始图像”和“直方图均衡化处理后的图像”。然后添加两个按钮，右键单击 Properties，在属性设置中分别将它们的 ID 设置为“IDC_OPEN”和“IDC_HIST”，Caption 分别设置为“打开文件”和“直方图均衡化”。将新添加的这些控件拖拉到合适的位置后，对话框界面如下所示：

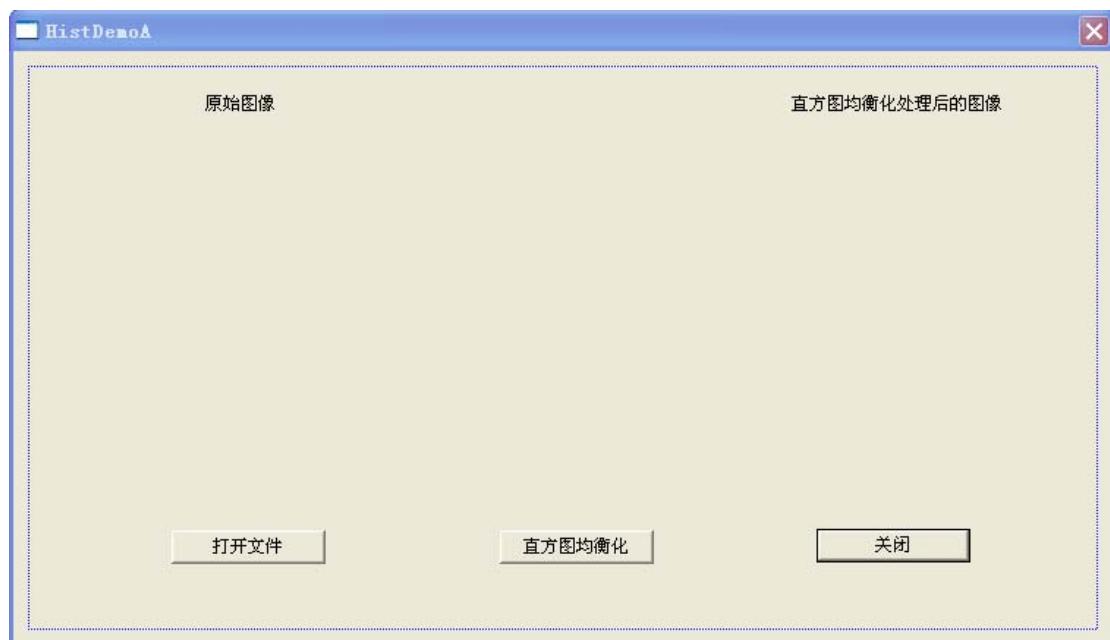


图 4

二、创建 CImageWnd 类

1、单击菜单栏的 Insert，选择 New Class，在弹出的对话框中，将 Class type 选为 Generic Class，并在 Name 栏中输入 CImageWnd，然后点击 OK 按钮。如下图所示：

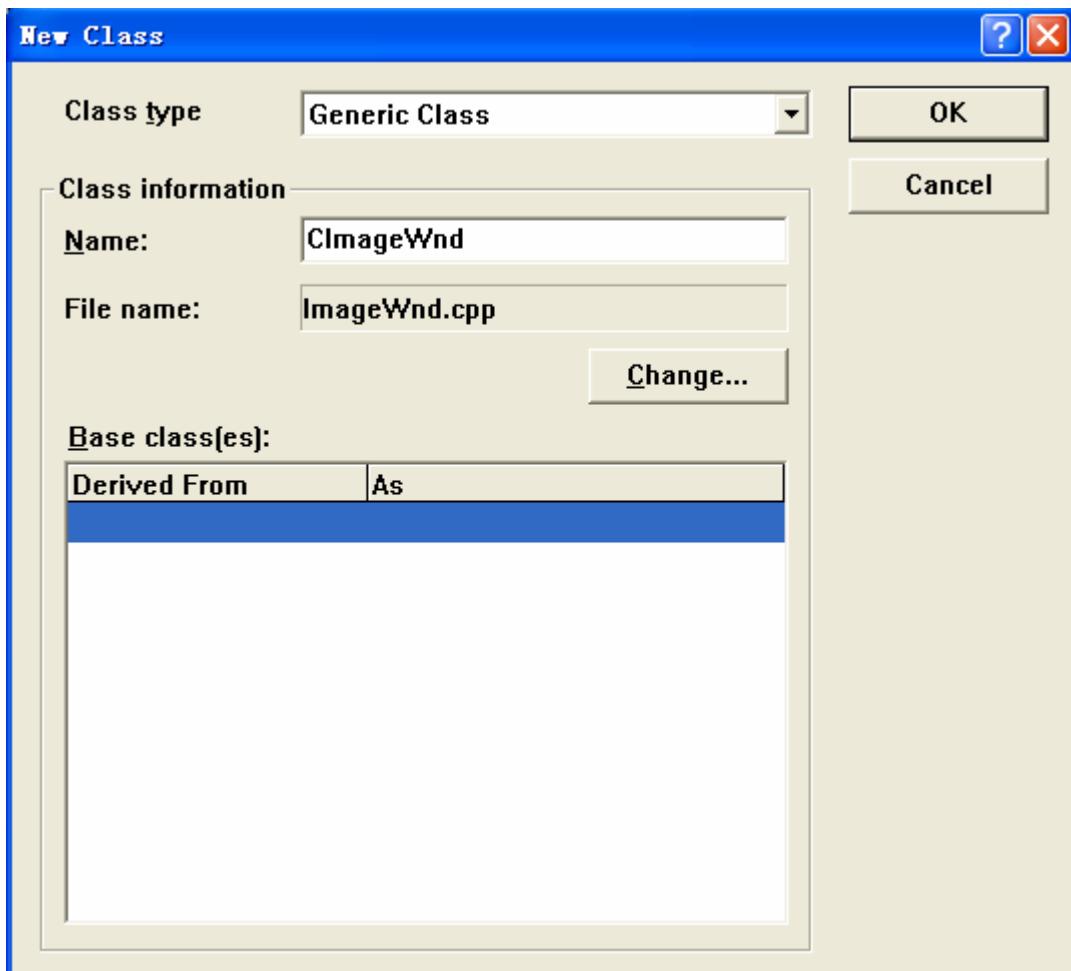


图 5

2、通过工作区的 FileView 找到 ImageWnd.h 文件，打开它并将类定义部分的 class CImageWnd 修改为 class CImageWnd:public CWnd。

3、为类 CImageWnd 添加成员变量。通过工作区的 ClassView，找到类 CImageWnd，单击右键，在弹出的菜单中选择 Add Member Variable，在弹出的对话框中输入要添加的成员变量的类型及名字，及选择访问限定语。在这里，我们依次加入访问限定语为 public 的下列 int 型变量：HCurrentPosition, HScrollMax, HScrollPosition, VCurrentPosition, VScrollMax, VScrollPosition, nWidth, nHeight, nByteWidth。然后再添加访问限定语为 public，类型为 BYTE 的成员变量*lpBits。完成类 CimageWnd 的成员变量的添加。

4、为类 CimageWnd 添加成员函数。通过工作区的 ClassView，找到类 CImageWnd，单击右键，在弹出的菜单中选择 Add Member Function，在弹出的对话框中输入要添加的成员函数的类型及声明，及选择访问限定语。在此我们添加以下的成员函数：访问限定语为 public，类型为 void 的 SetImage(int cx,int cy,const void *bits)，在函数的定义部分加入如下的代码：

```
nWidth=cx;  
nHeight=cy;  
nByteWidth=nWidth*3;  
    if (nByteWidth%4) nByteWidth+=4-(nByteWidth%4);  
if (lpBits) delete lpBits;  
lpBits=new BYTE[nByteWidth*nHeight];
```

```
memcpy(lpBits,bits,nByteWidth*nHeight);
if (cx>320 || cy>240) SetScroll(cx,cy);
else SetScroll(0,0);
Invalidate();
```

然后添加访问限定语为 public, 类型为 void 的 SetScroll(int cx,int cy), 并在函数的定义部分加入如下代码:

```
HScrollPosition=0;
HCurrentPosition=0;
VScrollPosition=0;
VCurrentPosition=0;
SetScrollPos(SB_HORZ,0);
SetScrollPos(SB_VERT,0);

SCROLLINFO sinfo;
sinfo.cbSize=sizeof(SCROLLINFO);
sinfo.fMask=SIF_PAGE|SIF_RANGE;
sinfo.nMin=0;

//HScroll
sinfo.nMax=cx-1;
sinfo.nPage=302;
SetScrollInfo(SB_HORZ,&sinfo);
HScrollMax=cx-302;

//VScroll
sinfo.nMax=cy-1;
sinfo.nPage=222;
SetScrollInfo(SB_VERT,&sinfo);
VScrollMax=cy-222;
```

5、为类 CimageWnd 添加构造函数和析构函数。

在构造函数 CImageWnd::CImageWnd()的定义体内加入下面的代码:

```
HScrollPosition=0;
HCurrentPosition=0;
HScrollMax=0;
VScrollPosition=0;
VCurrentPosition=0;
VScrollMax=0;
lpBits=0;
```

在析构函数 CImageWnd::~CImageWnd()的定义体内加入下面的代码:

```
if (lpBits) delete lpBits;
```

6、为类 CImageWnd 添加消息映射。

打开文 ImageWnd.h, 并添加如下代码:

```
protected:  
    //{{AFX_MSG(CImageWnd)  
    afx_msg BOOL OnEraseBkgnd(CDC* pDC);  
    afx_msg void OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar);  
    afx_msg void OnVScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar);  
    afx_msg void OnPaint();  
    afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);  
    //}}AFX_MSG  
DECLARE_MESSAGE_MAP()
```

打开文件 ImageWnd.cpp, 在析构函数 CImageWnd::~CImageWnd()的定义后面, 加入下面的代码:

```
BEGIN_MESSAGE_MAP(CImageWnd, CWnd)  
    //{{AFX_MSG_MAP(CImageWnd)  
    ON_WM_ERASEBKGND()  
    ON_WM_HSCROLL()  
    ON_WM_VSCROLL()  
    ON_WM_PAINT()  
    ON_WM_CREATE()  
    //}}AFX_MSG_MAP  
END_MESSAGE_MAP()
```

7、添加消息处理函数。打开文件 ImageWnd.cpp, 加入下面的代码:

```
#define BACKGROUND RGB(128,128,128)  
BOOL CImageWnd::OnEraseBkgnd(CDC* pDC)  
{  
    if (!lpBits) pDC->FillSolidRect(0,0,320,240,BACKGROUND);  
    return TRUE;  
}  
  
void CImageWnd::OnPaint()  
{  
    CPaintDC dc(this); // device context for painting  
    if (lpBits)  
    {  
        int x,y;  
        x=-HScrollPosition;  
        y=-VScrollPosition;  
  
        if (nWidth<320) x=(320-nWidth)/2;  
        if (nHeight<240) y=(240-nHeight)/2;  
        BITMAPINFOHEADER bmi;  
        bmi.biSize=sizeof(BITMAPINFOHEADER);
```

```

        bmi.biWidth=nWidth;
        bmi.biHeight=nHeight;
        bmi.biPlanes=1;
        bmi.biBitCount=24;
        bmi.biCompression=BI_RGB;
        bmi.biSizeImage=0;
        bmi.biXPelsPerMeter=0;
        bmi.biYPelsPerMeter=0;
        bmi.biClrUsed=0;
        bmi.biClrImportant=0;
        StretchDIBits(dc.m_hDC,x,y,nWidth,nHeight,0,0,nWidth,nHeight,
        lpBits,
        (BITMAPINFO *)&bmi,
        DIB_RGB_COLORS,
        SRCCOPY);
    }

}

void CImageWnd::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    switch(nSBCode)
    {
    case SB_LINEDOWN:
        HScrollPosition+=5;
        break;
    case SB_LINEUP:
        HScrollPosition-=5;
        break;
    case SB_PAGEDOWN:
        HScrollPosition+=302;
        break;
    case SB_PAGEUP:
        HScrollPosition-=302;
        break;
    case SB_THUMBPOSITION:
    case SB_THUMBTRACK:
        HScrollPosition=nPos;
        break;
    }
    if (HScrollPosition<0) HScrollPosition=0;
    if (HScrollPosition>HScrollMax) HScrollPosition=HScrollMax;

    SetScrollPos(SB_HORZ,HScrollPosition);
}

```

```

if (HCurrentPosition!=HScrollPosition)
{
    ScrollWindow((HCurrentPosition-HScrollPosition),0);
    HCurrentPosition=HScrollPosition;
}
CWnd::OnHScroll(nSBCode, nPos, pScrollBar);
}

void CImageWnd::OnVScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    switch(nSBCode)
    {
    case SB_LINEDOWN:
        VScrollPosition+=5;
        break;
    case SB_LINEUP:
        VScrollPosition-=5;
        break;
    case SB_PAGEDOWN:
        VScrollPosition+=222;
        break;
    case SB_PAGEUP:
        VScrollPosition-=222;
        break;
    case SB_THUMBUPOFFSET:
    case SB_THUMBPOSITION:
    case SB_THUMBTRACK:
        VScrollPosition=nPos;
        break;
    }
    if (VScrollPosition<0) VScrollPosition=0;
    if (VScrollPosition>VScrollMax) VScrollPosition=VScrollMax;

    SetScrollPos(SB_VERT,VScrollPosition);
    if (VCurrentPosition!=VScrollPosition)
    {
        ScrollWindow(0,(VCurrentPosition-VScrollPosition));
        VCurrentPosition=VScrollPosition;
    }
    CWnd::OnVScroll(nSBCode, nPos, pScrollBar);
}

int CImageWnd::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CWnd::OnCreate(lpCreateStruct) == -1)

```

```
    return -1;
    return 0;
}
```

三、为 CHistDemoADlg 添加代码

1、找到并双击打开文件 HistDemoADlg.h (通过工作区的 FileView)，在 class CHistDemoADlg : public CDialog 前添加#include "ImageWnd.h"

2、在 HistDemoADlg.h 的 public 部分，CHistDemoADlg(CWnd* pParent = NULL);之后加入下面的语句：

```
int nWidth;
int nHeight;
int nLen;
int nByteWidth;
BYTE *lpBackup;
BYTE *lpBitmap;
BYTE *lpBits;
CString FileName;
CImageWnd source,dest;
```

3、单击菜单栏的 View，在弹出的菜单中选择 ClassWizard，然后在弹出的对话框中选择 Message Maps 选项卡，在 Class name 下拉列表框中选择 CHistDemoADlg，在 Object IDs 中选择 CHistDemoADlg，在 Member functions 中选择双击 OnInitDialog，(如果没有的话，在 Messages 中 双击 WM_INITDIALOG 添加)，窗口将跳转到 BOOL CHistDemoADlg::OnInitDialog()的定义部分。在“// TODO: Add extra initialization here”后面加入下面的代码：

```
source.Create(0,"Source",WS_CHILD|WS_VISIBLE,CRect(40,40,360,280),this, 10000);
dest.Create(0,"Destination",WS_CHILD|WS_VISIBLE,CRect(400,40,720,280),      this,
10001);
```

至此，编译我们的建立的程序，界面如下图所示：(如有任何错误不能正常编译通过，请读者仔细检查前面的编程步骤，笔者按此建立过程成功编译通过并显示出下面的界面)



图 6

单击按钮“打开文件”和“直方图均衡化”，没有任何反映，因为我们还没有给它们添加响应。单击“关闭”按钮，对话框关闭。

四、为“打开文件”按钮添加响应。

1、为类 CHistDemoADlg 添加成员函数 void LoadBitmap()。打开工作区的 ClassView，右键单击类 CHistDemoADlg，在弹出的菜单中选择 Add Member function，在弹出的对话框中加入访问限定语为 public，函数类型为 void，函数声明为 LoadBitmap 的成员函数，点击“OK”按钮，编辑窗口将跳转到函数 void LoadBitmap() 的定义部分。在函数 void LoadBitmap() 的定义部分的定义部分加入下面的代码：

```
BITMAPINFOHEADER *pInfo;
pInfo=(BITMAPINFOHEADER*)(lpBitmap+sizeof(BITMAPFILEHEADER));
nWidth=pInfo->biWidth;
nByteWidth=nWidth*3;
if (nByteWidth%4) nByteWidth+=4-(nByteWidth%4);
nHeight=pInfo->biHeight;
if (pInfo->biBitCount!=24)
{
    if (pInfo->biBitCount!=8)
    {
        AfxMessageBox("无效位图");
        delete lpBitmap;
        lpBitmap=0;
        return;
    }
    unsigned int PaletteSize=1<<pInfo->biBitCount;
    if      (pInfo->biClrUsed!=0          &&      pInfo->biClrUsed<PaletteSize)
        PaletteSize=pInfo->biClrUsed;
    lpBits=lpBitmap+sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER);
```

```

RGBQUAD *pPalette=(RGBQUAD *)lpBits;
lpBits+=sizeof(RGBQUAD)*PaletteSize;
nLen=sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER)+nByteWidth*
nHeight;
BYTE *lpTemp=lpBitmap;
lpBitmap=new BYTE[nLen];
BITMAPFILEHEADER bmh;
BITMAPINFOHEADER bmi;
bmh.bfType='B'+'M'*256;
bmh.bfSize=nLen;
bmh.bfReserved1=0;
bmh.bfReserved2=0;
bmh.bfOffBits=54;
bmi.biSize(sizeof(BITMAPINFOHEADER));
bmi.biWidth=nWidth;
bmi.biHeight=nHeight;
bmi.biPlanes=1;
bmi.biBitCount=24;
bmi.biCompression=BI_RGB;
bmi.biSizeImage=0;
bmi.biXPelsPerMeter=0;
bmi.biYPelsPerMeter=0;
bmi.biClrUsed=0;
bmi.biClrImportant=0;
int nBWidth=pInfo->biWidth;
if (nBWidth%4) nBWidth+=4-(nBWidth%4);
memset(lpBitmap,0,nLen);
memcpy(lpBitmap,&bmh,sizeof(BITMAPFILEHEADER));
memcpy(lpBitmap+sizeof(BITMAPFILEHEADER),&bmi,sizeof(BITMAPINFOHEA
DER));
BYTE
*lpBits2=lpBitmap+sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER);
int x,y,p1,p2,Palette;
for(y=0;y<nHeight;y++)
{
    for(x=0;x<nWidth;x++)
    {
        p1=y*nBWidth+x;
        p2=y*nByteWidth+x*3;
        if (lpBits[p1]<PaletteSize) Palette=lpBits[p1];
        else Palette=0;
        lpBits2[p2]=pPalette[Palette].rgbBlue;
        lpBits2[p2+1]=pPalette[Palette].rgbGreen;
        lpBits2[p2+2]=pPalette[Palette].rgbRed;
    }
}

```

```

    }
}

delete lpTemp;
}

lpBits=lpBitmap+sizeof(BITMAPFILEHEADER)+sizeof(BITMAPINFOHEADER);
if (lpBackup) delete lpBackup;
lpBackup=new BYTE[nLen];
memcpy(lpBackup,lpBitmap,nLen);

```

2、单击菜单栏的 View，在弹出的菜单中选择 ClassWizard，然后在弹出的对话框中选择 Message Maps 选项卡，在 Class name 下拉列表框中选择 CHistDemoADlg，在 Object IDs 中选择 IDC_OPEN，在 Messages 中单击 BN_CLICKED，然后再单击 ADD Function 按钮，将弹出添加成员函数的对话框，如下图所示：

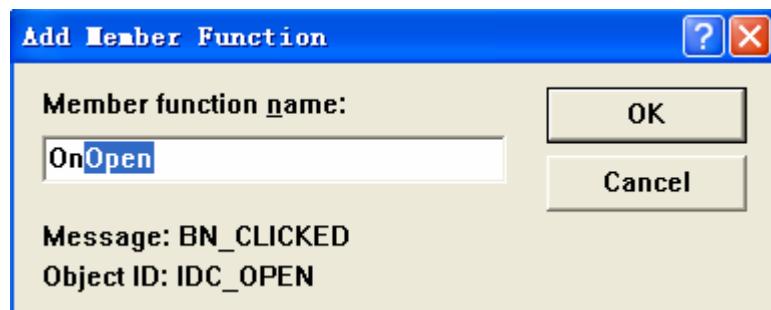


图 7

接受默认的函数名，单击“OK”按钮，就成功地添加了该成员函数。单击“Edit Code”按钮，窗口将跳转到该成员函数的定义部分，加入下面的代码：

```

CFile File=0;
CFileDialog dlg(TRUE,0,0,OFN_HIDEREADONLY,"位图文件 (*.bmp)|所有文件|*.*||",this);
if (dlg.DoModal()==IDOK)
{
    FileName=dlg.GetPathName();
    if (!File.Open(FileName,CFile::modeRead)) return;
    // TODO: add loading code here
    if (lpBitmap) delete lpBitmap;
    nLen=File.GetLength();
    lpBitmap=new BYTE[nLen];
    File.Read(lpBitmap,nLen);
    LoadBitmap();
    if (lpBitmap) source.SetImage(nWidth,nHeight,lpBits);
}

```

3、在类 CHistDemoADlg 的构造函数 CHistDemoADlg(CWnd* pParent /*=NULL*/)中加入下面的代码：

```
lpBitmap=0;
```

```
lpBackup=0;
```

到此为止，我们所构建的程序已经能实现对单击“打开文件”按钮的响应。编译程序，并打开 BMP 文件，如下图所示：



图 8

五、为“直方图均衡化”按钮添加响应

1、单击菜单栏的 View，在弹出的菜单中选择 ClassWizard，然后在弹出的对话框中选择 Message Maps 选项卡，在 Class name 下拉列表框中选择 CHistDemoADlg，在 Object IDs 中选择 IDC_HIST，在 Messages 中单击 BN_CLICKED，然后再单击 ADD Function 按钮，将弹出添加成员函数的对话框，如下图所示：



图 9

接受默认的函数名，单击“OK”按钮，就成功地添加了该成员函数。单击“Edit Code”按钮，窗口将跳转到该成员函数的定义部分，加入下面的代码：

```
HistogramEq();
```

2、刚刚添加的 HistogramEq() 实现直方图均衡化处理的算法。我们在 HistDemoADlg.cpp 中加入下面的代码：

```
#define Point(x,y) lpPoints[(x)+(y)*nWidth]
#define Point1(x,y) lpPoints1[(x)+(y)*nWidth]

void GetPoints(int nWidth,int nHeight,BYTE *lpBits,BYTE *lpPoints)
```

```

{
    int x,y,p;
    int nByteWidth=nWidth*3;
    if (nByteWidth%4) nByteWidth+=4-(nByteWidth%4);
    for(y=0;y<nHeight;y++)
    {
        for(x=0;x<nWidth;x++)
        {
            p=x*3+y*nByteWidth;
            lpPoints[x+y*nWidth]=(BYTE)(0.299*(float)lpBits[p+2]+0.587*(float)lpBits[p+1]
                +0.114*(float)lpBits[p]+0.1);
        }
    }
}

void PutPoints(int nWidth,int nHeight,BYTE *lpBits,BYTE *lpPoints)
{
    int nByteWidth=nWidth*3;
    if (nByteWidth%4) nByteWidth+=4-(nByteWidth%4);
    int x,y,p,p1;
    for(y=0;y<nHeight;y++)
    {
        for(x=0;x<nWidth;x++)
        {
            p=x*3+y*nByteWidth;
            p1=x+y*nWidth;
            lpBits[p]=lpPoints[p1];
            lpBits[p+1]=lpPoints[p1];
            lpBits[p+2]=lpPoints[p1];
        }
    }
}
}

```

3、为类 CHistDemoADlg 添加成员函数 HistogramEq()。打开工作区的 ClassView，右键单击类 CHistDemoADlg，在弹出的菜单中选择 Add Member function，在弹出的对话框中加入访问限定语为 public，函数类型为 void，函数声明为 HistogramEq 的成员函数，点击“OK”按钮，编辑窗口将跳转到函数 void CHistDemoADlg::HistogramEq()的定义部分。在函数 void CHistDemoADlg::HistogramEq()的定义部分加入下面的代码：

```

if (lpBitmap==0) return;
BYTE *lpOutput=new BYTE[nByteWidth*nHeight];
HistogramEq1(nWidth,nHeight,lpBits,lpOutput);
dest.SetImage(nWidth,nHeight,lpOutput);
delete lpOutput;
NoColor();

```

4、为类 CHistDemoADlg 添加成员函数 HistogramEq1(int nWidth,int nHeight,BYTE *lpInput,BYTE *lpOutput)。打开工作区的 ClassView，右键单击类 CHistDemoADlg，在弹出的菜单中选择 Add Member function，在弹出的对话框中加入访问限定语为 public，函数类型为 void，函数声明为 HistogramEq1(int nWidth,int nHeight,BYTE *lpInput,BYTE *lpOutput) 的成员函数，点击“OK”按钮，编辑窗口将跳转到函数 void CHistDemoADlg::HistogramEq1(int nWidth, int nHeight, BYTE *lpInput, BYTE *lpOutput) 的定义部分。在函数 void CHistDemoADlg::HistogramEq1(int nWidth, int nHeight, BYTE *lpInput, BYTE *lpOutput) 的定义部分加入下面的代码：

```
int x,y;
BYTE *lpPoints=new BYTE[nWidth*nHeight];
GetPoints(nWidth,nHeight,lpInput,lpPoints);

int r[256],s[256];
ZeroMemory(r,1024);
ZeroMemory(s,1024);
for(y=0;y<nHeight;y++){
    for(x=0;x<nWidth;x++){
        r[Point(x,y)]++;
    }
}
s[0]=r[0];
for(y=1;y<256;y++)
{
    s[y]=s[y-1];
    s[y]+=r[y];
}
for(y=0;y<nHeight;y++){
    for(x=0;x<nWidth;x++){
        Point(x,y)=s[Point(x,y)]*255/nWidth/nHeight;
    }
}

PutPoints(nWidth,nHeight,lpOutput,lpPoints);
delete lpPoints;
```

5、为类 CHistDemoADlg 添加成员函数 NoColor()。打开工作区的 ClassView，右键单击类 CHistDemoADlg，在弹出的菜单中选择 Add Member function，在弹出的对话框中加入访问限定语为 public，函数类型为 void，函数声明为 NoColor 的成员函数，点击“OK”按钮，编辑窗口将跳转到函数 void CHistDemoADlg:: NoColor() 的定义部分。在函数 void CHistDemoADlg:: NoColor() 的定义部分加入下面的代码：

```
if (lpBitmap==0) return;
int x,y,p;
BYTE Point;
for(y=0;y<nHeight;y++)
```

```
{  
    for(x=0;x<nWidth;x++)  
    {  
        p=x*3+y*nByteWidth;  
        Point=(BYTE)(0.299*(float)lpBits[p+2]+0.587*(float)lpBits[p+1]+0.114*(float)lpBits[p]+0.1);  
        lpBits[p+2]=Point;  
        lpBits[p+1]=Point;  
        lpBits[p]=Point;  
    }  
}  
source.SetImage(nWidth,nHeight,lpBits);
```

至此，编译我们所构建的程序，并打开一个 bmp 文件，点击“直方图均衡化”按钮，程序就对打开的 bmp 文件进行直方图均衡化处理，并将处理的结果显示在右侧的窗格中，如下图所示：



图 10

这样，我们就成功的构建了一个直方图均衡化处理的程序。