【数字图像处理】四.MFC对话框绘制灰度直方图

本文主要讲述基于VC++6.0 MFC图像处理的应用知识，主要结合自己大三所学课程《数字图像处理》及课件进行回忆讲解，主要通过MFC单文档视图实现点击弹出对话框绘制BMP图片的灰度直方图，再获取平均灰度、中指灰度和标准差等值。文章比较详细基础，希望该篇文章对你有所帮助~

【数字图像处理】一.MFC详解显示BMP格式图片

【数字图像处理】二.MFC单文档分割窗口显示图片

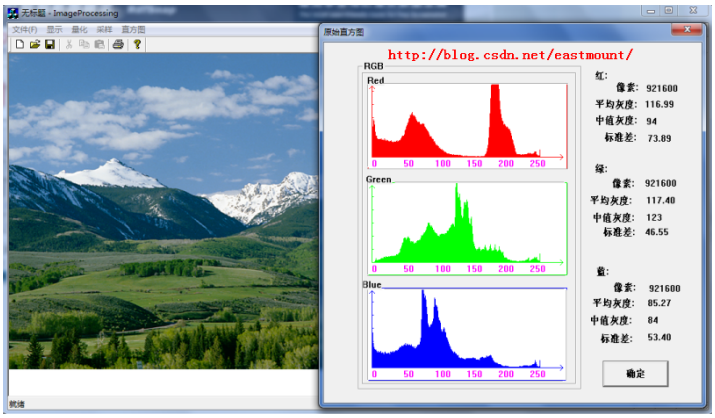
【数字图像处理】三.MFC实现图像灰度、采样和量化功能详解

免费资源下载地址：

http://download.csdn.net/detail/eastmount/8757243

一. 程序运行结果

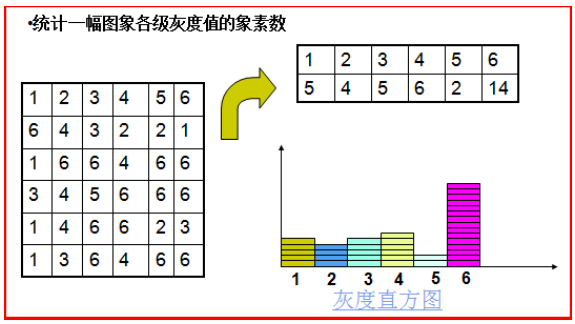
该篇文章主要是在上一篇文章基础上进行的讲解，其中当打开一张BMP图像后，点击”直方图“-》”显示原图直方图“如下。



二. 灰度直方图原理

什么是灰度直方图?

灰度直方图（histogram）是灰度级的函数，描述的是图像中每种灰度级像素的个数，反映图像中每种灰度出现的频率。横坐标是灰度级，纵坐标是灰度级出现的频率。



对于连续图像，平滑地从中心的高灰度级变化到边缘的低灰度级。直方图定义为：

$$H(D)=\lim_{\Delta D\to 0}\frac{A(D)-A(D+\Delta D)}{D-(D+\Delta D)}=\lim_{\Delta D\to 0}\frac{A(D)-A(D+\Delta D)}{-\Delta D}=-\frac{d}{dD}A(D)$$

其中A(D)为阈值面积函数：为一幅连续图像中被具有灰度级D的所有轮廓线所包围的面积。对于离散函数，固定ΔD为1，则：H(D)=A(D)-A(D+1)

色彩直方图是高维直方图的特例，它统计色彩的出现频率，即色彩概率分布信息。

通常这需要一定的量化过程，将色彩分成若干互不重叠的种类。一般不直接在RGB色彩空间中统计，而是在将亮度分离出来后，对代表色彩部分的信息进行统计，如在HSI空间的HS子空间、YUV空间的UV子空间，以及其它反映人类视觉特点的彩色空间表示中进行。

其中直方图的计算方法如下：

依据定义，若图像具有L（通常L=256,即8位灰度级）级灰度，则大小为MxN的灰度图像f(x,y)的灰度直方图hist[0...L-1]可用如下计算获得。

1、初始化 hist[k]=0; k=0,...,L-1

2、统计 hist[f(x,y)]++; x=0,...,M-1, y =0,...,N-1

3、归一化 hist[f(x,y)]/=M*N

那么说了这么多，直方图究竟有什么作用呢?

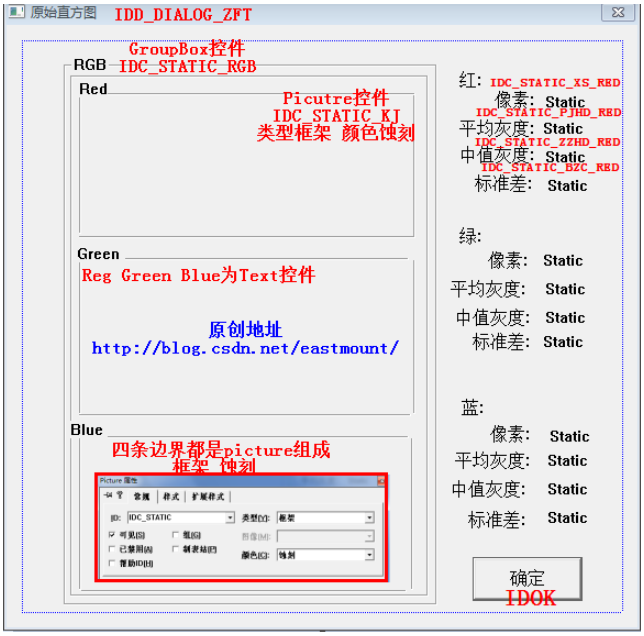
在使用轮廓线确定物体边界时，通过直方图更好的选择边界阈值，进行阈值化处理；对物体与背景有较强对比的景物的分割特别有用；简单物体的面积和综合光密度IOD可以通过图像的直方图求得。

三. 程序实现

1.建立直方图对话框

第一步：创建Dialog

将视图切换到ResourceView界面，选中Dialog右键鼠标新建一个Dialog，并新建一个名为IDD_DIALOG_ZFT，设置成下图对话框。



右键添加属性如下:

对话框-原始直方图-IDD_DIALOG_ZFT

组框-RGB-IDC_STATIC_RGB

图像-框架-IDC_STATIC_KJ-蚀刻(重点:有它才能添加直方图在此处，注意GetDlgItem()函数中是IDC而不是IDD对话框)

添加蚀刻线(图像蚀刻形成的直线)形如图中的3个矩形框，并添加静态文本：Red、Green、Blue、红、绿、蓝、像素、平均灰度、中值灰度、标准差；这些静态文本都是IDC_STATIC且为默认属性

添加红色4个值(Static)、绿色4个值、蓝色4个值，分别为：

IDC_STATIC_XS_RED(GREEN BLUE)对应像素XS

IDC_STATIC_PJHD_RED(GREEN BLUE)对应平均灰度PJHD

IDC_STATIC_ZZHD_RED(GREED BLUE)对应中值灰度ZZHD

IDC_STATIC_BZC_RED(GREEN BLUE)对应标准差BZC

第二步：建立类向导MFC ClassWizard

(1) 在对话框资源模板空白区双击鼠标(Ctrl+W)，创建一个新类，命名为CImageZFTDlg会自动生成它的.h和.cpp文件。在类向导中选中类名CImageZFTDlg，IDs为CImageZFTDlg，WM_INITDIALOG建立这个函数用于初始化。

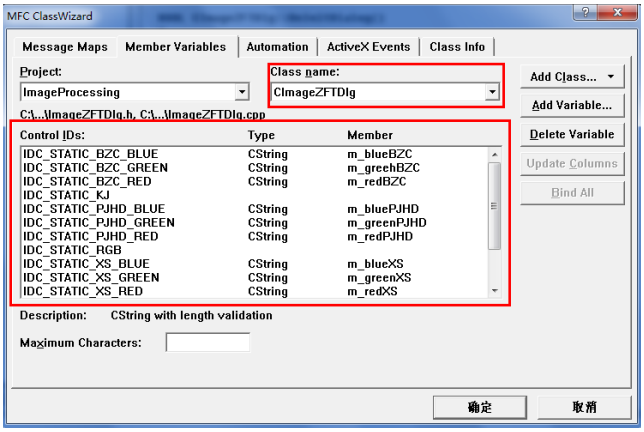
(2) 打开类向导，选择Member Variables页面，添加如下变量，类型均为CString。

像素 m_redXS、m_greenXS、m_blueXS

标准差 m_redBZC、m_greeenBZC、m_blueBZC

平均灰度 m_redPJHD、m_greenPJHD、m_bluePJHD

中值灰度 m_redZZHD、m_greenZZHD、m_blueZZHD

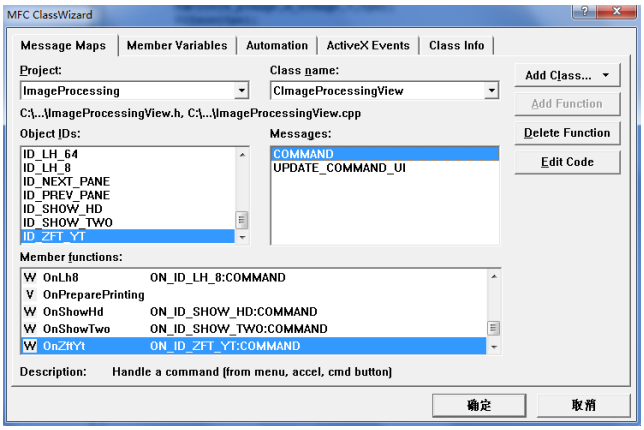


(3) 在View.cpp中添加直方图的头文件 #include "ImageZFTDlg.h"

第三步：设置菜单栏调用直方图对话框

(1) 将视图切换到ResourceView界面，选中Menu，在IDR_MAINFRAM中添加菜单项“直方图”，菜单属性中选择“弹出”，在“直方图”中添加子菜单“显示原图直方图”。

(2) 设置其属性为ID_ZFT_YT(显示直方图原图)，同时建立类向导，选择ID_ZFT_YT(IDs)，通过COMMAND建立显示直方图函数OnZftYt()。



第四步：添加代码及计算4个值

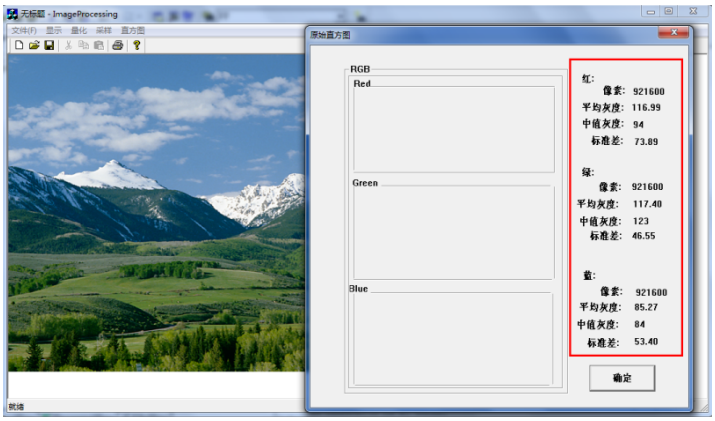
在ImageProcessingView.cpp中添加如下代码，注释中有如何求平均灰度、中值灰度和标准差的消息算法过程。

```
//引用显示直方图头文件
#include "ImageZFTDlg.h"
#include "math.h"

/*全局变量在TestZFTDlg.cpp中引用 用extern*/
int Red[256],Green[256],Blue[256];

/************************************************/
/* 添加直方图显示功能，并在直方图下方显示相关信息
/* 如平均灰度、中值灰度、标准差和像素总数
/* ID_ZFT_YT:直方图原图显示
/************************************************/
void CImageProcessingView::OnZftYt()
{
	if(numPicture==0) {
		AfxMessageBox("载入图片后才能显示原图直方图!",MB_OK,0);
		return;
	}
	AfxMessageBox("显示原图直方图!",MB_OK,0);
	CImageZFTDlg dlg;

	//打开临时的图片
	FILE *fpo = fopen(BmpName,"rb");
	fread(&bfh,sizeof(BITMAPFILEHEADER),1,fpo);
	fread(&bih,sizeof(BITMAPINFOHEADER),1,fpo);

	int i,j;
	for(j=0;j<256;j++) { //定义数组并清零
		Red[j]=0;
		Green[j]=0;
		Blue[j]=0;
	}

	//计算4个数据
```

```
unsigned char red,green,blue;
                                int IntRed,IntGreen,IntBlue;                    //强制转换
double sumRedHD=0,sumGreenHD=0,sumBlueHD=0;   //记录像素总的灰度值和
for(i=0; i<m_nImage/3; i++ )
{
        fread(&red,sizeof(char),1,fpo);
        IntRed=int(red);
        sumRedHD=sumRedHD+IntRed;
        if( IntRed>=0 && IntRed<256 ) Red[IntRed]++; //像素0-255之间

        fread(&green,sizeof(char),1,fpo);
        IntGreen=int(green);
        sumGreenHD=sumGreenHD+IntGreen;
        if( IntGreen>=0 && IntGreen<256 ) Green[IntGreen]++;

        fread(&blue,sizeof(char),1,fpo);
        IntBlue=int(blue);
        sumBlueHD=sumBlueHD+IntBlue;
        if( IntBlue>=0 && IntBlue<256 ) Blue[IntBlue]++;
}
fclose(fpo);

//像素:int型转换为CString型
dlg.m_redXS.Format("%d",m_nImage);
dlg.m_greenXS.Format("%d",m_nImage);
dlg.m_blueXS.Format("%d",m_nImage);

//平均灰度值:计算24位bmp图片的灰度值,我是记录RGB中的所有平均值
float pinRedHD,pinGreenHD,pinBlueHD;
pinRedHD=sumRedHD*3/m_nImage;
pinGreenHD=sumGreenHD*3/m_nImage;     //平均灰度=总灰度/总像素
pinBlueHD=sumBlueHD*3/m_nImage;

dlg.m_redPJHD.Format("%.2f",pinRedHD);
dlg.m_greenPJHD.Format("%.2f",pinGreenHD);
dlg.m_bluePJHD.Format("%.2f",pinBlueHD);

/*****************************************************************/
/* 中值灰度:算法重点(黄凯大神提供)
/* 中值灰度:所有像素中的中位数,应该所有像素排序找到中间的灰度值
/* 算法:num[256]记录各灰度出现次数,sum+=num[i],找到sum=总像素/2
/*****************************************************************/
int sumRedZZHD=0,sumGreenZZHD=0,sumBlueZZHD=0;
int redZZHD,greenZZHD,blueZZHD;
for(i=0;i<256;i++)
{
        sumRedZZHD=sumRedZZHD+Red[i];
        if(sumRedZZHD>=m_nImage/6)          //m_nImage被分成3份RGB并且sum=总像素/2
        {
                redZZHD=i;
                break;
        }
}
for(i=0;i<256;i++)
{
        sumGreenZZHD=sumGreenZZHD+Green[i];
        if(sumGreenZZHD>=m_nImage/6)          //m_nImage被分成3份RGB并且sum=总像素/2
        {
                greenZZHD=i;
                break;
        }
}
```

```
        for(i=0;i<256;i++)
                                {
                sumBlueZZHD=sumBlueZZHD+Blue[i];
                if(sumBlueZZHD>=m_nImage/6)          //m_nImage被分成3份RGB并且sum=总像素/2
                {
                        blueZZHD=i;
                        break;
                }
        }

        dlg.m_redZZHD.Format("%d",redZZHD);
        dlg.m_greenZZHD.Format("%d",greenZZHD);
        dlg.m_blueZZHD.Format("%d",blueZZHD);

        /*****************************************************************/
        /*标准差:标准差=方差的算术平方根
        /*        方差s^2=[(x1-x)^2+(x2-x)^2+.....(xn-x)^2]/n
        /* 算法:不用开m_nImage数组进行计算 用num[256]中数进行
        /* 方差=(平均灰度-i)*(平均灰度-i)*Red[i]   有Red[i]个灰度值为i的数
        /*****************************************************************/
        float redBZC,greenBZC,blueBZC;        //标准差
        double redFC=0,greenFC=0,blueFC=0;    //方差
        for(i=0;i<256;i++)
        {
                redFC=redFC+(pinRedHD-i)*(pinRedHD-i)*Red[i];   //有Red[i]个像素i
                greenFC=greenFC+(pinGreenHD-i)*(pinGreenHD-i)*Green[i];
                blueFC=blueFC+(pinBlueHD-i)*(pinBlueHD-i)*Blue[i];
        }

        redBZC=sqrt(redFC*3/m_nImage);
        greenBZC=sqrt(greenFC*3/m_nImage);
        blueBZC=sqrt(blueFC*3/m_nImage);

        dlg.m_redBZC.Format("%.2lf",redBZC);
        dlg.m_greenBZC.Format("%.2lf",greenBZC);
        dlg.m_blueBZC.Format("%.2lf",blueBZC);

        //重点必须添加该语句才能弹出对话框
        if(dlg.DoModal()==IDOK)
        {

        }
}
```

第五步：此时运行结果如下图所示，打开图片可以显示参数。



2.建立对话框与View联系并绘制直方图

重点（极其重要*）

(1) 如何在MFC中(View中)实现对子对话框的画图或直方图响应?

解决方法:在子对话框中.cpp文件中实现画图响应，不要再View.cpp中实现，否则图像会以menu背景为坐标，而在ImageZFTDlg.cpp中建立OnPaint函数实现画图，它默认会以子对话框为标准。

(2) 如何把View.cpp中的图片像素直方图信息传递给子对话框ImageZFTDlg.cpp呢?

解决方法:如果自定义ImageStruct.h中建立全局变量，每个.cpp中引用该头文件调用总是报错(未知)，所以我在View.h中建立一个全局变量int Red[256];再在子文件.cpp中函数里调用该全局变量即可extern int Red[256]，这是非常重要的一个C语言知识。

(3) 画图函数OnPaint()参考源代码中详细注释。

如何绘制坐标轴、文字、图像，其实自己绘制而没调用第三方库还是挺有意思的。

第一步：建立画直方图函数OnPaint

打开类向导(Ctrl+W)，类名选择CImageZFTDlg，IDs选择CImageZFTDlg，在Message函数中建立WM_PAINT映射，默认函数名为OnPaint建立函数void CImageZFTDlg::OnPaint()

第二步：绘制直方图大致思想如下

(1) 重点：获取要绘制直方图的位置和图像资源的对应号ID(IDC_STATIC_KJ 框架)，我当时认为绘制直方图只能绘制到”图像“控件IDC中，不能是对话框IDD。

```
CWnd *pWnd = GetDlgItem(IDC_STATIC_KJ);
CDC *pDC = pWnd->GetDC();
```

(2) 获取对话框矩形的长和宽

```
CRect rectpic;
GetDlgItem(IDC_STATIC_KJ)->GetWindowRect(&rectpic);
```

(3) 创建画笔对象并对画笔进行颜色设置

```
CPen *RedPen = new CPen();
RedPen->CreatePen(PS_SOLID,1RGB(255,0,0));
```

(4) 选中当前画笔并保存以前画笔

```
CGdiObject *RedOlderPen = pDC->SelectObject(RedPen);
```

(5) 绘制直方图(图像坐标自己算)

```
矩形 pDC->Rectangle(9,327,312,468);
移动 pDC->MoveTo(15,331);
直线 pDC->LineTo(15,488);
文字 pDC->TextOut(15+48*i,450,str);
```

(6) 恢复以前画笔

```
pDC->SelectObject(RedOlderPen);
delete RedPen;
ReleaseDC(pDC);
```

第三步：源代码与详细注释思想

在ImageZFTDlg.cpp中修改OnPaint函数：

```
//****************绘制原图直方图*********************//
void CImageZFTDlg::OnPaint()
{
        CPaintDC dc(this); // device context for painting

        // TODO: Add your message handler code here
```

```
/*****************************************************************************/
/* 重点知识:(百度)
/* 如何在View.cpp中把一个变量的值传给其它对话框
/*
/* 错误一:在View.h中定义的pubic变量只能在View.cpp中用
/* 错误二:定义一个Struct.h中存全局变量,在2个函数中分别调用#include "Struct.h"
/*
/* 解决方法一: (CSDN 不会)参数用 A& a 两个对话框里都可以访问a
/* 解决方法二: (CSDN 不会)重载
/*
/* 解决:在View.cpp中定义全局变量 void CBmpDrawView::OnZftYt() 前面 并函数中操作
/*       在dialog的cpp中即void CTestZFTDlg::OnPaint()中在定义一个extern int a
/*****************************************************************************/
extern int Red[256],Green[256],Blue[256];

/*写在该空间中可以省略Invalidate()语句*/
/*获取控件的CDC指针*/
CRect rectpic;
GetDlgItem(IDC_STATIC_KJ)->GetWindowRect(&rectpic);

int x,y;
x=rectpic.Width();
y=rectpic.Height();

CWnd *pWnd=GetDlgItem(IDC_STATIC_KJ);
CDC *pDC=pWnd->GetDC();

/***********************/
/*重点:画直方图 红色
/***********************/
CPen *RedPen=new CPen();                                //创建画笔对象
RedPen->CreatePen(PS_SOLID,1,RGB(255,0,0));             //红色画笔
CGdiObject *RedOlderPen=pDC->SelectObject(RedPen);    //选中当前红色画笔并保存以前的画笔

/*画图*/
pDC->Rectangle(9,16,312,147);      //画一个矩形框
pDC->MoveTo(15,20);                //绘制坐标轴
pDC->LineTo(15,128);               //Y竖轴
pDC->LineTo(305,128);              //X横轴

pDC->MoveTo(305,128);              //绘制X箭头
pDC->LineTo(300,123);              //绘制上边箭头
pDC->MoveTo(305,128);
pDC->LineTo(300,133);              //绘制下边箭头

pDC->MoveTo(15,20);                //绘制Y箭头
pDC->LineTo(10,25);                //绘制左边箭头
pDC->MoveTo(15,20);
pDC->LineTo(20,25);                //绘制右边箭头

/*************************************************************************/
/* TextOut函数功能:
/* 该函数用当前选择的字体、背景颜色和正文颜色将一个字符串写到指定位置
/* BOOL TextOut(HDC hdc,int x,int y,LPCTSTR str,int numStr)
/* 表示:x起始坐标,y起始坐标,字符串,字符串中字符个数
/*
/* SetTextColor函数功能:
/* 设置指定设备环境(HDC)的字体颜色
/* SetTextColor (HDC, COLORREF) 如:SetTextColor(HDC,RGB(255,0,0));
/*************************************************************************/
```

```
CString str;
                    int i;
for(i=0;i<=5;i++)                    //写X轴刻度线
{
        str.Format("%d",i*50);               //0-255之间添加6个刻度值
        pDC->SetTextColor(RGB(255,0,255));   //设置字体颜色
        pDC->TextOut(15+48*i,130,str);       //输出字体
        pDC->MoveTo(15+48*i,128);            //绘制X轴刻度
        pDC->LineTo(15+48*i,125);
}
for(i=0;i<=5;i++)                        //写Y轴刻度线
{
        pDC->MoveTo(15,128-20*i);            //绘制Y轴刻度
        pDC->LineTo(18,128-20*i);
}

/*绘制直方图主要的代码*/
for(i=1;i<256;i++)
{
        pDC->MoveTo(15+i,128);
        if( (128-16) > (Red[i]/40) )
                pDC->LineTo(15+i,128-(Red[i]/40));
        else
                pDC->LineTo(15+i,16);            //超过矩形的画矩形高
}


/**********************/
/*重点:画直方图 绿色
/**********************/
CPen *GreenPen=new CPen();                              //创建画笔对象
GreenPen->CreatePen(PS_SOLID,1,RGB(0,255,0));           //绿色画笔
CGdiObject *GreenOlderPen=pDC->SelectObject(GreenPen);

pDC->Rectangle(9,167,312,308);     //画一个矩形框
pDC->MoveTo(15,171);               //绘制坐标轴
pDC->LineTo(15,288);               //Y竖轴
pDC->LineTo(305,288);              //X横轴

pDC->MoveTo(305,288);              //绘制X箭头
pDC->LineTo(300,283);              //绘制上边箭头
pDC->MoveTo(305,288);
pDC->LineTo(300,293);              //绘制下边箭头

pDC->MoveTo(15,171);                //绘制Y箭头
pDC->LineTo(10,176);                //绘制左边箭头
pDC->MoveTo(15,171);
pDC->LineTo(20,176);                //绘制右边箭头

for(i=0;i<=5;i++)                   //写X轴刻度线
{
        str.Format("%d",i*50);               //0-255之间添加6个刻度值
        pDC->SetTextColor(RGB(255,0,255));   //设置字体颜色
        pDC->TextOut(15+48*i,290,str);       //输出字体

        pDC->MoveTo(15+48*i,288);            //绘制X轴刻度
        pDC->LineTo(15+48*i,285);
}
for(i=0;i<=5;i++)                        //写Y轴刻度线
{
        pDC->MoveTo(15,288-20*i);            //绘制Y轴刻度
```

```
        pDC->LineTo(18,288-20*i);         }

/*绘制直方图主要的代码*/
for(i=1;i<256;i++)
{
        pDC->MoveTo(15+i,288);
        if( (288-167) > (Green[i]/40) )
                pDC->LineTo(15+i,288-(Green[i]/40));
        else
                pDC->LineTo(15+i,167);            //超过矩形的画矩形高
}


/**********************/
/*重点:画直方图 蓝色
/***************((*****/
CPen *BluePen=new CPen();                              //创建画笔对象
BluePen->CreatePen(PS_SOLID,1,RGB(0,0,255));           //蓝色画笔
CGdiObject *BlueOlderPen=pDC->SelectObject(BluePen);

pDC->Rectangle(9,327,312,468);      //画一个矩形框
pDC->MoveTo(15,331);                //绘制坐标轴
pDC->LineTo(15,448);                //Y竖轴
pDC->LineTo(305,448);               //X横轴

pDC->MoveTo(305,448);               //绘制X箭头
pDC->LineTo(300,443);               //绘制上边箭头
pDC->MoveTo(305,448);
pDC->LineTo(300,453);               //绘制下边箭头

pDC->MoveTo(15,331);                //绘制Y箭头
pDC->LineTo(10,336);                //绘制左边箭头
pDC->MoveTo(15,331);
pDC->LineTo(20,336);                //绘制右边箭头


for(i=0;i<=5;i++)                   //写X轴刻度线
{
        str.Format("%d",i*50);               //0-255之间添加6个刻度值
        pDC->SetTextColor(RGB(255,0,255));   //设置字体颜色
        pDC->TextOut(15+48*i,450,str);       //输出字体

        pDC->MoveTo(15+48*i,448);            //绘制X轴刻度
        pDC->LineTo(15+48*i,445);
}
for(i=0;i<=5;i++)                       //写Y轴刻度线
{
        pDC->MoveTo(15,448-20*i);            //绘制Y轴刻度
        pDC->LineTo(18,448-20*i);
}

/*绘制直方图主要的代码*/
for(i=1;i<256;i++)
{
        pDC->MoveTo(15+i,448);
        if( (448-327) > (Blue[i]/40) )
                pDC->LineTo(15+i,448-(Blue[i]/40));
        else
                pDC->LineTo(15+i,327);            //超过矩形的画矩形高
}
```

```
                                        //恢复以前的画笔
            pDC->SelectObject(RedOlderPen);
            pDC->SelectObject(GreenOlderPen);
            pDC->SelectObject(BlueOlderPen);
            delete RedPen;
            delete GreenPen;
            delete BluePen;
            ReleaseDC(pDC);
            return;

            // Do not call CDialog::OnPaint() for painting messages
    }
```

此时运行程序即可显示直方图。

最后还是希望文章对你有所帮助，如果文章有不足或错误之处，请海涵~文章不仅仅讲述了直方图相关的知识，同时文章也给你提供了一种绘制坐标图像的思想和详细注释。有时候一直怀疑回忆这些知识会让我停滞不前，但心安即好，何必在意！

从来没有什么终南捷径和大神，真正的捷径只有三个：坚持、专注、认真。其他的都是细枝末节，做到这三个，其他的自然而然都会拥有。——同学CY

(By:Eastmount 2015-5-31 下午3点 http://blog.csdn.net/eastmount/)