基于 VC++在数字图像处理中的格式转换和图像增强处理

朱喆
武汉大学遥感信息工程学院,湖北武汉 430072

摘要:数字图像处理运用在当今社会的许多方面,许多的图形软件都有图像处理的部分,那么如果要有自己的处理方法就只有运用语言编程了,在这里主要介绍用 VC++对图像进行格式转换和增强处理。

关键词:格式转换;线性拉伸;滤波
中图分类号:TP311　　文献标识码:A

Abstract:Digital image process is used in many aspects in out society. A lot of image process software have this part. But if we want to have out own way to deal with the pictures, we have to use some computer languages. This passage is mainly about how to transform formats of images and to strengthen the images.

Key Words:transform formats;linear extend;filtering

本文详细说明一个图形处理的小程序的生成,它可以将 RAW 图像格式转换成 BMP 图像格式,并显示出来,对 BMP 图像进行灰度的线性拉伸的操作,对 BMP 图像进行图像增强处理高通滤波的操作。

1 编写程序将 RAW 图像格式转换成 BMP图像格式,并显示出来

主要原理:数字图像是连续图像的一种近似表示,通常用由采样点所组成的矩阵来表示。按不同的方式进行存储数字图像的灰度,就得到了不同格式的图像文件。这个实习是先读入 RAW 图像的文件,然后再写入 BMP 图像文件的四个部分(前三部分要自己赋值);读、写入 BMP 图像后,然后再显示 BMP 图像文件。

1.1 主要步骤

(1)根据学号和班级建立自己的文件夹。

(2)使用 AppWizard 生成一个基于多文档的项目(以自己命名)。

靠的帮助文档,在用户使用产生迷惑时可以自己寻求解决方法。帮助设施细则如下:

(1)帮助文档中的性能介绍与说明要与系统性能配套一致。

(2)打包新系统时,对作了修改的地方在帮助文档中要做相应的修改。

(3)操作时要提供及时调用系统帮助功能,常用 F1。

(4) 在界面上调用帮助时应该能够及时定位到与该操作相对的帮助位置。也就是说帮助要有即时针对性。

(5)最好提供目前流行的联机帮助格式或 HTML 帮助格式。

(6)用户可以用关键词在帮助索引中搜索所要的帮助,当然也应该提供帮助主题词。

(7)如果没有提供书面的帮助文档的话,最好有打印帮助的功能。

(8)在帮助中应该提供我们的技术支持方式,一旦用户难以自己解决可以方便的寻求新的帮助方式。

4 小结

本文主要讨论了软件界面设计的易用性概念,并提出了软件界面设计的一般应该遵循的原则。通常在设计界面时,还要充分考虑到用户的机器配置,在设计字体和图片时要注意分辨率的选择,这样才能使应用程序界面获得最佳的显示效果。在视窗技术飞速发展的今天,讲究软件界面设计的易用性显得非常重要。这就要求我们在今后的学习与工作中不断磨合,把我们的应用程序做的更好。

参考文献:

[1]吴柏鸿等.公共管理知识手册 MPA 全集.管理出版社,2001.

[2]郑人杰,殷人昆,陶永雷.实用软件工程.清华大学出版社,1997.

[3]软件业 ISO9000 质量体系的建立和认证.

收稿日期:2005 年 3 月

(3)建立 MFC 项目,给程序添加 dibap.h、dibapi.cpp、和 myfile.cpp 三个文件。

(4)添加菜单项"RAW->BMP 和相应对话框,并建立对话框的类和在 View 中加菜单项的函数,并建立相应的消息响应函数。

(5)在对话框内输入相应的值并在 OK 键里处理,程序如下:

```
void CDlgRawtoBmp::OnOK()
{UpdateData(TRUE); //读 RAW 格式文件
CFile file_raw,file_bmp;//定义文件对象
CFileException fe;//定义异常对象
HDIB m_HDib,m_HDIB;//定义句柄
if (! file_raw.Open(m_RawName, CFile::modeRead-
Write  CFile::shareDenyNone|CFile::typeBinary, &fe))
{ return ;}
DWORD  dwBitsSize = file_raw.GetLength();//得到
RAW 的大小
m_HDib = (HDIB) ::GlobalAlloc(GMEM_MOVE-
ABLE | GMEM_ZEROINIT, dwBitsSize);//分配句柄空间
if (m_HDib == 0)//如为 0 返回
{ return ; }
LPSTR pDIB = (LPSTR) ::GlobalLock((HGLOBAL)
m_HDib);//得到对应指针
if (file_raw.ReadHuge(pDIB, dwBitsSize ) ! =dwBits-
Size )//如果不相等返回
{ ::GlobalUnlock((HGLOBAL) m_HDib);
::GlobalFree((HGLOBAL) m_HDib);
return ; } //写位图
if (! file_bmp.Open(m_BmpName, CFile::modeCreate|
CFile::modeReadWrite | CFile::shareDenyNone,  &fe))//打
开异常处理
{ return ; }
BITMAPFILEHEADER bmfHeader;
//定义文件头对象
BITMAPINFOHEADER bmpHeader;
//定义信息头对象
RGBQUAD rgb[256];//定义调色版
bmfHeader.bfType=0x4D42;//文件头赋值
bmpHeader.biSizeImage=(((m_width*8) + 31) / 32 *
4)*m_length;
if(m_grind==0)
{ bmfHeader.bfSize =14 +40 +256*4 +bmpHeader.bi-
SizeImage;
bmfHeader.bfOffBits=14+40+256*4;}
if(m_grind==1)
{ bmfHeader.bfSize =14 +40 +bmpHeader.biSizeIm-
age*3;
bmfHeader.bfOffBits=14+40; }
bmfHeader.bfReserved1=0;
bmfHeader.bfReserved2=0;
bmpHeader.biSize=40;//信息头赋值
bmpHeader.biWidth=m_width;
bmpHeader.biHeight=m_length;
bmpHeader.biPlanes=1;
if(m_grind==0)
bmpHeader.biBitCount=8;
if(m_grind==1)
bmpHeader.biBitCount=24;
bmpHeader.biCompression=BI_RGB;
bmpHeader.biXPelsPerMeter=0;
bmpHeader.biYPelsPerMeter=0;
bmpHeader.biClrUsed=0;
bmpHeader.biClrImportant=0;
if(m_grind==0)//为 256 色时对调色板赋值
{ for(int i=0;i<256;i++)
{ rgb[i].rgbBlue=i;
rgb[i].rgbGreen=i;
rgb[i].rgbRed=i;
rgb[i].rgbReserved=0;}  }
file_bmp.Write(&bmfHeader,14);//写信息头
file_bmp.Write(&bmpHeader,40);//写文件头
if(m_grind==0)如过为 256 色写调色版
file_bmp.Write(rgb,256*4);
m_HDIB = (HDIB) ::GlobalAlloc(GMEM_MOVE-
ABLE | GMEM_ZEROINIT, bmpHeader.biSizeImage);//分
配句柄的空间
if (m_HDIB == 0)
{ return ; }
LPSTR pDIB2 = (LPSTR) ::GlobalLock((HGLOBAL)
m_HDIB);//得到句柄的指针
int swidth=(((m_width*8) + 31) / 32 * 4);
//计算得到的 width'
for(int j=0;j<m_length;j++)
//对小于 width 的象素加 0
for(int i=0;i<swidth;i++)
{ if(i<m_width)
```

```
*(pDIB2+swidth*j+i)=*(pDIB+swidth*j+i);
else  *(pDIB2+swidth*j+i)=0;}
file_bmp.WriteHuge(pDIB2,bmpHeader.biSizeImage);
//写入第四部分
GlobalUnlock(m_HDib);//句柄的释放
GlobalFree(m_HDib);
GlobalUnlock(m_HDIB);
GlobalFree(m_HDIB);
CDialog::OnOK();}
```

(6)在 Doc 中加入读、写函数和初始化函数,并在 View 中的 OnDaw 函数中加入相应的函数,使之能够显示出来。

(7)转换与显示成功。

1.2 注意事项与心得

(1) 注意 BMP 中宽度只可能是 4 的整数倍, 如果 RAW 中不是要补 0。

(2) 在程序的使用中句柄和指针的使用容易混淆。比如, 在程序中有这样的 LPSTR pDIB = (LPSTR) :: GlobalLock((HGLOBAL) m_HDib); pDIB 为通过::GlobalLock((HGLOBAL) m_HDib)全局函数得到 m_Hdib 句柄对应的指针,并且要知道指针指向的确切位置。

(3)对异常的情况要注意进行处理,要不然就会有程序的漏洞。比如:

```
m_HDib = (HDIB) ::GlobalAlloc (GMEM_MOVEABLE | GMEM_ZEROINIT, dwBitsSize);
if (m_HDib == 0)
{ return ; }
```

在分配内存时,如果没有分配到内存返回。

(4) 在编程的过程中对 C++有了更深入的了解,对图形的结构也更加清楚了。

2 编写程序对 BMP 图像进行灰度的线性拉伸的操作

实习原理:线性拉伸是一种常用的点运算的方法。是对图像中的所有点的灰度按照线性灰度变换函数进行处理,调整图像的对比度和亮度。在本程序中是对每一个原图像的像素做线性变换:x'=a*x+b, 这样 a 就可以代表对比度,b 就可以用来代表亮度了。

2.1 主要步骤:

(1)添加菜单项“线性变换”和相应对话框,并建立对话框的类和在 View 中加菜单项的函数, 同时得到 Doc 的指针,将得到的第二部分的指针传给对话框中定义的指针变量 m_hdib,并建立相应的消息响应函数。程序如下:

```
CPrs021200232590035Doc* pDoc = GetDocument();
//在 View 里得到 Doc 的指针
HDIB hDIB = pDoc->GetHDIB();
//得到第二部分的句柄
CLIneChan linechandlg;//定义对话框类的对象
linechandlg.m_hdib=hDIB;// 第二部分的指针传给对话框中定义的指针变量
linechandlg.DoModal();//弹出对话框
pDoc->UpdateAllViews(NULL);//刷新所有视图
```

(2) 在对话框内输入相应斜率与斜距值并在 OK 键里处理,程序如下:

```
void CLIneChan::OnOK()
{ UpdateData(TRUE);//对输入值反应
LPSTR lphDIB;//定义指针
unsigned char *lphDIB4;
lphDIB=(LPSTR)::GlobalLock((HGLOBAL) m_hdib);//句柄得到第二部分的指针
lphDIB4=(unsigned char *)(lphDIB+40+4*256);//由句柄得到第四部分的指针
double R;
int width=DIBWidth(lphDIB);//宽度
int height=DIBHeight(lphDIB);//高度
for(int j=0;j<height;j++)//对每个象数进行 y=ax+b 的线性变换
for(int i=0;i<width;i++)
{ R=*(lphDIB4+width*j+i)*m_a+m_b;
if(R<0)R=0;
if(R>255)R=255;
*(lphDIB4+width*j+i)=R;}
CDialog::OnOK(); }
```

2.2 注意事项与心得

(1)在新建的对话框的类里处理,需要将图像的句柄得到,可以在视图类中得到文档类的指针,然后再将得到的第二部分的指针传给对话框中定义的指针变量 m_hdib,这样就能在对话框的类中对图像处理了。

(2) 要把得到的第二部分的指针移动到第四部分,这样才能对 BMP 的象素部分进行线性的处理。

(3)由于线性变换是点处理,处理时不会影响到其它点,因此不必进行句柄的拷贝。

(4)由于线性变换时,变换后的象素可能会大于 255 或小于 0,这时对超限的象素一定要进行处理,使它在 0~255 之间。当大于 255 时, 给它赋值 255, 当小于 0 时,给它赋值为 0。

(5)对于有符号指针和无符号指针的强制转换要特别注意。由于灰度值是由字节形式存储的,而灰度值是在 0~255 之间的, 因此在用指针处理象素时要将指针变换为无符号指针。

(6)在赋值的时候不能直接对变化后的象素值赋为新值,然后才进行判断,因为如果真的超过了限值,是不能赋值成功的,因为无符号指针的范围也是 0~255。

3 编写程序对 BMP 图像进行图像增强处理高通滤波的操作

原理:高通滤波是局部处理的一种方法,它通过原图像与高通滤波算子进行卷积运算得到新的锐化了的图像。在这里我采用的是 3*3 的卷积核与图象求卷积(同理可以得到低通滤波处理的方法)。

3.1 主要步骤

(1)添加菜单项"高通滤波",并在 View 中加菜单项的函数, 同时得到 Doc 的指针, 得到的第二部分的指针,并建立相应的消息响应函数。

(2)通过拷贝句柄得到两个相同的句柄,同时由句柄得到两个指向第二部分的指针,程序如下:

```
LPSTR lphDIB,lphDIBAK;
//定义两个有符号的指向第二部分的指针
unsigned char *lphDIB4,*lphDIBAK4;
// 定义两个有符号的指向第四部分的指针
lphDIB=(LPSTR)::GlobalLock((HGLOBAL) hDIB);
//由句柄得到第二部分的指针
lphDIB4=(unsigned char *)(lphDIB+40+4*256);
//得到第四部分的指针
HDIB hDIBAK=(HDIB)CopyHandle(hDIB);
//拷贝句柄
lphDIBAK =(LPSTR)::GlobalLock ((HGLOBAL) hDIBAK);//由拷贝句柄得到指针
lphDIBAK4 =(unsigned char *) (lphDIBAK +40 + 4*256);//得到第四部分指针
```

(3)得到行与列的象素数,通过两个 for 循环与定义的高通滤波的算子进行卷积的运算, 刷新后就可以得到高通滤波后的图像了,程序如下:

```
int width=DIBWidth(lphDIB);//得到每行象素的个数
int height=DIBHeight (lphDIB);//得到每列象素的个数
for (int j=1;j<height-1;j++)//通过 for 循环与高通滤波算子进行卷集的运算
for(int i=1;i<width-1;i++)
{ R=(*(lphDIBAK4+width*(j-1)+(i-1))*(-1)\
+*(lphDIBAK4+width*(j-1)+i)*(-1)\
+*(lphDIBAK4+width*(j-1)+(i+1))*(-1)\
+*(lphDIBAK4+width*j+(i-1))*(-1)\
+*(lphDIBAK4+width*j+i)*9\
+*(lphDIBAK4+width*j+(i+1))*(-1)\
+*(lphDIBAK4+width*(j+1)+(i-1))*(-1)\
+*(lphDIBAK4+width*(j+1)+i)*(-1)+\
*(lphDIBAK4+width*(j+1)+(i+1))*(-1));
if(R<0)R=0;
if(R>255)R=255;
*(lphDIB4+width*j+i)=R;}
pDoc->UpdateAllViews(NULL);}
```

3.2 注意事项与心得

(1) 要把得到的第二部分的指针移动到第四部分,这样才能对 BMP 的象素部分进行卷积的处理。

(2) 最为关键的是由于高通滤波变换是局部处理,处理时会影响到其它点, 在计算下一点的时候不能代入已经变换后的点,因此必进行句柄的拷贝,得到两个相同的指针,新得到的指针用来计算,赋值给原来的指针,这样用来计算的指针所指向的象素是不会变的,而我们赋值的指针用来指向变化后的象素。

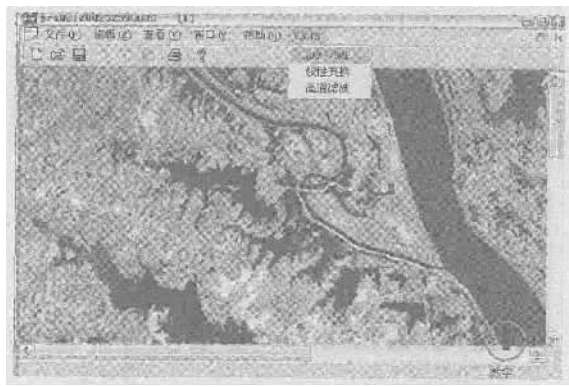
(3)由与进行卷积变换时,变换后的象素可能会大于 255 或小于 0, 这时对超限的象素一定要进行处理,使它在 0~255 之间。当大于 255 时,给它赋值 255,当小于 0 时,给它赋值为 0。

(4)在进行卷积变换时,一定要注意括号和运算的书写。如果一个括号或一个运算符号写错了,运行时没有报错信息,可是得不到需要的结果,这会浪费掉很多的时间查错。

4 成果显示

4.1 实习一:转换后的 BMP 图像的显示

第一步(图 1):



第二步(图 2):

用 VFP 与 Word 开发题库系统

王宝艳　叶玉萍
武汉大学,湖北武汉 430072

摘要:介绍了用 VFP6.0 与 Word 开发的一种题库系统,对系统中几种重要功能模块作了较详尽地阐述。

关键词:VFP;Word;数据库;模块设计
中图分类号:TP311　**文献标识码**:A

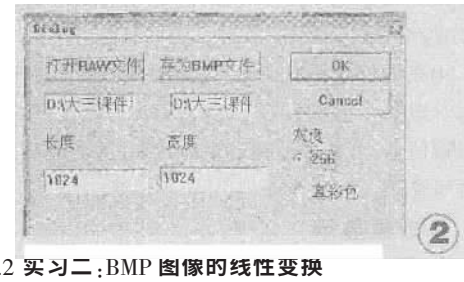
The development of Examination Question Bank Via VFP and Word

Abstract:The system for examination question bank is developed via VFP and Word,and some function modules in the system are introduced in more detail.

Key Words:VFP;Word;database;module design

测验是教学过程中必备的教学手段,不但能及时反馈学生对知识的掌握情况,而且能够帮助教育工作者改进教学方法,提高教学质量。现有题库系统很多,各有优缺点, 主要的缺点是处理公式、图形难。我用 Visual FoxPro 6.0 与 Word 设计的一种题库系统,利用 Word 的优势,解决了以上的缺点。本系统具有题库管理、试卷操作和试卷分析功能,可适用于多种学科。题目库采用 Word 文件,生成的试卷也为 Word 格式,生成的试卷图文并茂,美观大方,便于编辑。

VFP 是微软公司推出的关系型数据库管理系统,具有强大的数据库管理性能、友好的图形界面、面向对象的程序设计及汉字处理功能,具有数据存取速度快、兼容性好等特点,并能与一些软件友好连接。提供了许多基于 Windows 的崭新功能。VFP 主要提供了三种工



4.2 实习二:BMP 图像的线性变换

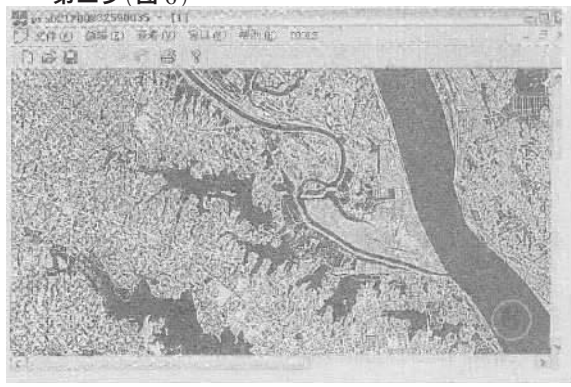
第一步(图 3)

第二步(图 4)

4.3 实习三:高通滤波变换

第一步(图 5)

第二步(图 6)



收稿日期:2005 年 3 月