# 24位真彩色DIB图像的显示

1.24再次复习BMP图像的储存格式：

BTMAPFILEHEADER

图像文件头

BTMAPINFOHEADER

图像文件信息头

调色板

8位位图:256个表项

24位位图:没有表项

位图数据

24位位图按BGR排列

每个颜色1个字节

要说明的是，24位位图没有调色板，所以从文件头找位图数据地址，可以偏移sizeof(BTMAPINFOHEADER)+sizeof(BTMAPFILEHEADER)或者偏移文件头里的bfOffBits个字节便可。

2.在Doc.h文件中定义以下变量：

public:

BYTE\* lpBMP;//指向BMP文件指针

BYTE\* lpDIB;//指向DIB指针

BITMAPFILEHEADER \*lpBmpFile;//指向文件头指针

BITMAPINFOHEADER \*lpBmpInfo;//指向文件信息头指针

unsigned char\* lpBmpBits;//指向像素数据指针

LONG nWidth;

LONG nHeight;

LONG LineByte;

int BitCount;

3.在Doc.cpp中构造函数和析构函数：

CBmpDisplayExDoc::CBmpDisplayExDoc()

{

// TODO: 在此添加一次性构造代码

lpDIB=NULL;

lpBmpFile=NULL;

lpBmpInfo=NULL;

lpBmpBits=NULL;

nWidth=0;

nHeight=0;

LineByte=0;

BitCount=0;

}

CBmpDisplayExDoc::~CBmpDisplayExDoc()

{

if (lpBMP)delete lpBMP;

}

4.在Doc.cpp中重载打开

BOOL CBmpDisplayExDoc::OnOpenDocument(LPCTSTR lpszPathName)

{

if (!CDocument::OnOpenDocument(lpszPathName))

return FALSE;

CFile file;

if(!file.Open(lpszPathName,CFile::modeReadWrite|CFile::shareDenyWrite,NULL,NULL))return FALSE;

lpBMP=new BYTE[file.GetLength()];

file.Read(lpBMP,file.GetLength());

file.Close();

//BMP文件头指针

lpBmpFile=(BITMAPFILEHEADER\*)lpBMP;

//DIB指针

lpDIB=(unsigned char\*)lpBmpFile+sizeof(BITMAPFILEHEADER);

//文件信息头指针

lpBmpInfo=(BITMAPINFOHEADER\*)lpDIB;

//图像像素数据指针

lpBmpBits=(unsigned char\*)lpBmpFile+lpBmpFile->bfOffBits;

nWidth=lpBmpInfo->biWidth;

nHeight=lpBmpInfo->biHeight;

BitCount=lpBmpInfo->biBitCount;

if (BitCount!=8&&BitCount!=24)

{

AfxMessageBox(\_T("程序只支持8位和24位位图！"));

delete lpDIB;lpDIB=NULL;

delete lpBmpFile;lpBmpFile=NULL;

delete lpBmpInfo;lpBmpInfo=NULL;

nWidth=0;

nHeight=0;

LineByte=0;

BitCount=0;

return FALSE;

}

int ColorTableLng=0;

if (BitCount==8)

{

ColorTableLng=pow(float(2),BitCount);

}

if (BitCount==24)

{

ColorTableLng=0;

}

LineByte=(nWidth\*BitCount/8+3)/4\*4;

SetPathName(lpszPathName);

return TRUE;

}

4.在View.cpp中：

void CBmpDisplayExView::OnDraw(CDC\* pDC)

{

CBmpDisplayExDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

if (!pDoc)

return;

// TODO: 在此处为本机数据添加绘制代码

pDC->SetStretchBltMode(COLORONCOLOR);

LONG nWidth=pDoc->nWidth;

LONG nHeight=pDoc->nHeight;

unsigned char\* lpBit=pDoc->lpBmpBits;

LPBITMAPINFO lpBmpInfo=(BITMAPINFO\*)pDoc->lpDIB;

::StretchDIBits(pDC->GetSafeHdc(),0,0,nWidth,nHeight,0,0,nWidth,nHeight,

lpBit,lpBmpInfo,DIB\_RGB\_COLORS,SRCCOPY);

}

5.本程序使用cfile对象将BMP文件读入到内存中保存，程序保存文件数据的变量是一个BYTE\*指针。读入数据的大小为file.GetLength()，单位为字节，于是可以使用DOC中的变量：

BYTE\* lpBMP;//指向BMP文件指针

BYTE\* lpDIB;//指向DIB指针

BITMAPFILEHEADER \*lpBmpFile;//指向文件头指针

BITMAPINFOHEADER \*lpBmpInfo;//指向文件信息头指针

unsigned char\* lpBmpBits;//指向像素数据指针

指向内存数据，这样可以方便读取图像中的各种信息。

6.将图像显示到屏幕上用StretchDIBits函数，使用上与显示灰度图像没有区别。

7.BYTE的本质就是unsigned char，8位。