

MFC 对话框Picture Control(图片控件)中静态和动态显示Bmp图片

原创 Eastmount 最后发布于2014-05-21 02:29:21 阅读数 28804 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏,采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



Eastmount

¥9.90

订阅

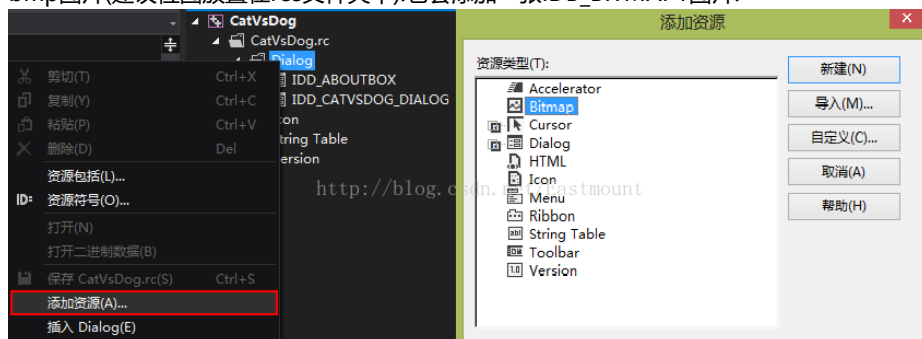
最近有同学问我如何实现MFC基于对话框在图片控件中加载图片?其实使用MFC显示图片的方法各种各样,但是还是有些同学不知道怎样显示.以前在《数字图像处理》课程中完成的软件都是基于单文档的程序,这里介绍两种在对话框picthre控件中显示BMP图片的最简单基础的方法.

~~方法可能并不完美,高手忽略,但是提供一种能运行的方法,希望对刚接触这方面知识的同学有所帮助.可能你觉得文章过于简单或者有些过于详细叙述(点到即可我我并不反对),但也为哪些入门同学想想,当初自己也是一头雾水.

一.静态显示bmp图片

1.创建MFC基于对话框程序,添加一个button1和picture控件,其中“资源视图”中编辑picture控件ID为:IDC_STATIC_PIC.

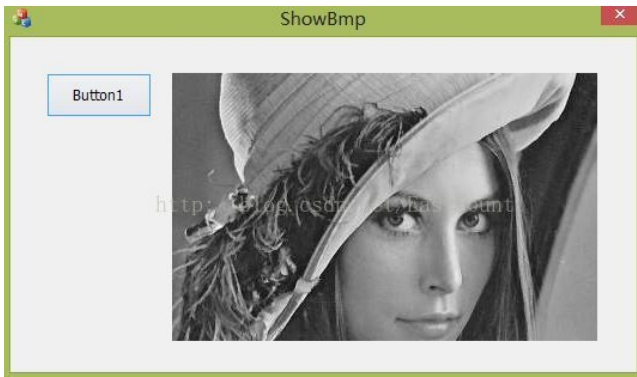
2.在“资源视图”(如果没有,视图中可调用出来)中右键鼠标Dialog处,选择“添加资源”.在对话框中选择Bitmap,点击“导入”,选择本地bmp图片(建议位图放置在res文件夹中).它会添加一张IDB_BITMAP1图片.



3.点击button1按钮,在生成的函数中添加如下代码:

```
void CShowBmpDlg::OnBnClickedButton1()
{
    // 从资源中加载图片
    CBitmap bitmap;
    // 加载指定位图资源 Bmp 图片ID
    bitmap.LoadBitmap(IDB_BITMAP1);
    // 获取对话框上的句柄 图片控件ID
    CStatic *p=(CStatic *)GetDlgItem(IDC_STATIC_PIC);
    // 设置静态控件窗口风格为位图居中显示
    p->ModifyStyle(0xf,SS_BITMAP|SS_CENTERIMAGE);
    // 将图片设置到Picture控件上
    p->SetBitmap(bitmap);
}
```

4.运行结果如下图所示:



5.总结

通过上面的截图我们可以发现:该方法是通过资源ID来获取BMP图片的,所以我称之为静态的方法.同时,可能你见到过在添加Picture控件时修改其Type为Bitmap在通过image选择ID的方法.但这里是通过代码来实现的,而且ModifyStyle函数的功能就相当于设置其属性.

~~上图很明显的反应了它显示图片时,如果图片尺寸超过控件size只能显示部分图片.方法我只叙述到这,需要你自己去尝试.曾经在单文档中显示BMP图片时,是通过重载onpaint事件,GetClientRect获取尺寸调用StretchBlt(图像可拉伸压缩)实现(参照前文).

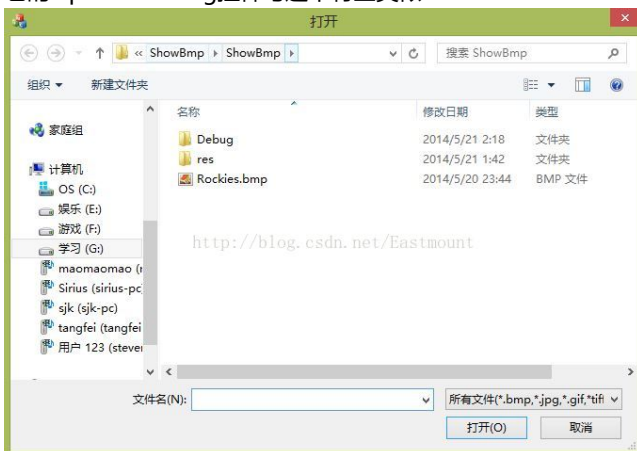
其实LoadBitmap函数已被LoadImage替代,该函数用于装载图标\光标\位图.如下代码所示可以实现同样的效果,其中Rockies.bmp为本地文件,与程序相同文件夹.

```
void CShowBmpDlg::OnBnClickedButton1()
{
    // 获取对话框上图片控件的句柄
    CStatic* pWnd = (CStatic*)GetDlgItem(IDC_STATIC_PIC);
    // 设置静态控件窗口风格为位图居中显示
    pWnd->ModifyStyle(0xf, SS_BITMAP|SS_CENTERIMAGE);
    // 显示图片
    pWnd->SetBitmap((HBITMAP)::LoadImage(NULL,
        _T("Rockies.bmp"), // 资源号或本地文件名
        IMAGE_BITMAP,      // 装载位图 IMAGE_CURSOR光标 IMAGE_ICON图标
        0,                  // 宽度 0为默认大小
        0,                  // 高度 像素为单位
        LR_CREATEDIBSECTION | LR_DEFAULTSIZE | LR_LOADFROMFILE));
}
```

二.动态加载bmp图片

动态添加的方法一样,只是添加代码如下(这结合我以前的代码完成).主要的步骤是:

1.先定义打开文件对话框,通过它获取bmp图片,其中dlg.DoMal()==IDOK表示按下确认按钮,实现后面操作.如果你学过C#就会发现它的OpenFileDialog控件与这个有些类似.



2.然后,获取读取文件的路径和后缀,确保其为bmp时执行读取图片信息,在"2.MFC-bmp图片读取保存"中,我曾详细讲述过如何读取bmp图片,这里也是二进制读取文件,读取其文件头(BITMAPFILEHEADER)、信息头(BITMAPINFOHEADER)、获取其位图数据.

3.最后显示图像到Picture控件中,并通过GetClientRect函数获取矩形区域,显示图像.

```

void CShowBmpDlg::OnBnClickedButton1()
{
    // 打开文件 定义四种格式的文件bmp gif jpg tiff
    CString filter;
    filter="所有文件(*.bmp;*.jpg;*.gif;*.tiff)|*.bmp;*.jpg;*.gif;*.tiff| BMP(*.bmp)|*.bmp|
JPG(*.jpg)|*.jpg| GIF(*.gif)|*.gif| TIFF(*.tiff)|*.tiff|";
    CFileDialog dlg(TRUE,NULL,NULL,OFN_HIDEREADONLY,filter,NULL);
    // 按下确定按钮 dlg.DoModal() 函数显示对话框
    if( dlg.DoModal() == IDOK )
    {
        // 打开对话框获取图像信息
        CString BmpName = dlg.GetPathName(); // 获取文件路径名 如D:\pic\abc.bmp
        CString EntName = dlg.GetFileExt(); // 获取文件扩展名
        EntName.MakeLower(); // 将文件扩展名转换为一个小写字符

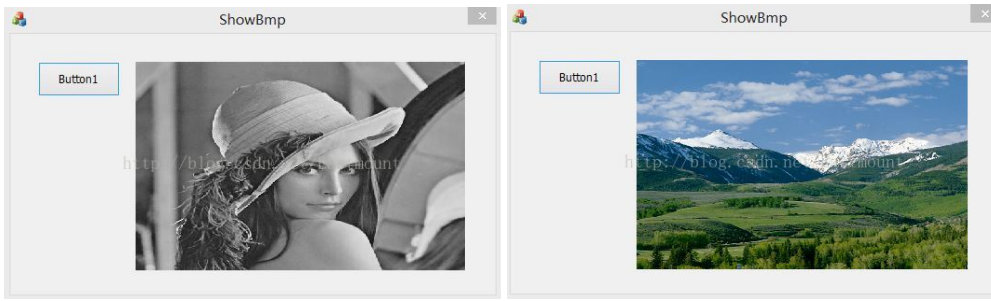
        if(EntName.Compare(_T("bmp")) == 0)
        {
            // 定义变量存储图片信息
            BITMAPINFO *pBmpInfo; // 记录图像细节
            BYTE *pBmpData; // 图像数据
            BITMAPFILEHEADER bmpHeader; // 文件头
            BITMAPINFOHEADER bmpInfo; // 信息头
            CFile bmpFile; // 记录打开文件

            // 以只读的方式打开文件 读取bmp图片各部分 bmp文件头 信息 数据
            if(!bmpFile.Open(BmpName, CFile::modeRead|CFile::typeBinary))
                return;
            if (bmpFile.Read(&bmpHeader,sizeof(BITMAPFILEHEADER)) != sizeof(BITMAPFILEHEADER))
                return;
            if (bmpFile.Read(&bmpInfo,sizeof(BITMAPINFOHEADER)) != sizeof(BITMAPINFOHEADER))
                return;
            pBmpInfo = (BITMAPINFO *)new char[sizeof(BITMAPINFOHEADER)];
            // 为图像数据申请空间
            memcpy(pBmpInfo,&bmpInfo,sizeof(BITMAPINFOHEADER));
            DWORD dataBytes = bmpHeader.bfSize - bmpHeader.bfOffBits;
            pBmpData = (BYTE*)new char[dataBytes];
            bmpFile.Read(pBmpData,dataBytes);
            bmpFile.Close();

            // 显示图像
            CWnd *pWnd=GetDlgItem(IDC_STATIC_PIC); // 获得pictrue控件窗口的句柄
            CRect rect;
            pWnd->GetClientRect(&rect); // 获得pictrue控件所在的矩形区域
            CDC *pDC=pWnd->GetDC(); // 获得pictrue控件的DC
            pDC->SetStretchBltMode(COLORONCOLOR);
            StretchDIBits(pDC->GetSafeHdc(),0,0,rect.Width(),rect.Height(),0,0,
                bmpInfo.biWidth,bmpInfo.biHeight,pBmpData,pBmpInfo,DIB_RGB_COLORS,SRCCOPY);
        }
    }
}

```

运行程序显示结果如下图所示,点击按钮打开图片,显示图片.



总结:通过这个程序你可以很明显的看到动态加载图片由于是通过获取picture控件矩形来显示图的,图像被拉伸了.而且代码很好的反应了如何获取bmp图像的方法.通过静态的和动态的两种方法讲述.

~~最后希望该文章对大家有所帮助,如果文章中有错误或不足之处,还请海涵.

(By:Eastmount 2014-5-21 夜2点半 原创CSDN <http://blog.csdn.net/eastmount/>)

PS:《数字图像处理》课程希望大家也可以学习下,你可以自己完成一个软件,你将对图像有更深入的认识.以前我们要求实现的功能就包括:图像载入保存、采样量化、直方图显示、线性和非线性变换(对比度拉伸、图像均衡化)、几何运算(最近邻差值和双线性差值)、几何变换(平移、反转、旋转)、图像增强(平滑、锐化Laplacian等各种算子)、边缘检测、图像编码压缩(Huffman编码和游程编码)、图像分割复原、二值图像细化等功能.记得最蛋疼的是DICOM医学图像在Android显示与处理花了很长时间才完成的.

👍 点赞 14 ☆ 收藏 🔄 分享 ...



Eastmount 博客专家

发布了455 篇原创文章 · 获赞 6523 · 访问量 509万+

他的留言板

关注