



写留言 邀请进圈子  
发消息 加友情链接

进家园 加好友

2012年度IT博客大赛  
十大杰出IT博客诞生

去看看

## 转载 bitmap格式分析

2009-11-18 14:08:05

标签: bitmap 图片格式 度写 休闲

本文转自本人Blog: Sinlgerblog.mblogger.cn

最近正在着手开发一个图片库，也就是实现对常见图片格式的度写操作。作为总结与积累，我会把这些图片格式以及加载的实现写在我的Blog上。

说到图片，位图（Bitmap）当然是最简单的，它Windows显示图片的基本格式，其文件扩展名为\*.BMP。在Windows下，任何各式的图片文件（包括视频播放）都要转化为位图个时候才能显示出来，各种格式的图片文件也都是在位图格式的基础上采用不同的压缩算法生成的（Flash中使用了适量图，是按相同颜色区域存储的）。

一、下面我们来看看位图文件（\*.BMP）的格式。

位图文件主要分为如下3个部分：

块名称	对应Windows结构体定义	大小（Byte）
文件信息头	BITMAPFILEHEADER	14
位图信息头	BITMAPINFOHEADER	40
RGB颜色阵列	BYTE*	由图像长宽尺寸决定

1、文件信息头BITMAPFILEHEADER

结构体定义如下：

```
typedef struct tagBITMAPFILEHEADER { /* bmfh */  
    UINT bfType;  
    DWORD bfSize;  
    UINT bfReserved1;  
    UINT bfReserved2;  
    DWORD bfOffBits;  
} BITMAPFILEHEADER;
```

其中：

bfType	说明文件的类型，该值必需是0x4D42，也就是字符'BM'。
bfSize	说明该位图文件的大小，用字节为单位
bfReserved1	保留，必须设置为0
bfReserved2	保留，必须设置为0
	说明从文件头开始到实际的图象数据之间的字节的偏移量。这个

- 智能交通监控系统观后日志
- InstallShield8.0制作自..
- 图模板（二）——遍历
- 详述trait和policy演化（..
- 二叉树模板
- 按摩小姐气得移动公司老..
- 队列模板


搜索BLOG文章

搜索

我的技术圈(1)更多>>

网络沙龙


最近访客




dylinshi




reall..




silen..




qinyunti




73163..




lin19..




wangj..




simon..




feng..



zhans..



朋御天下



shipe..

最新评论

- aniuzhang: 首先感谢博主给了这么简洁的一个介..
- bluebee: 你好像对文档类还不是很熟悉，加油。
- 359342916: 厉害~~
- w156445045: 哈哈你想笑死我哦~
- xusilu: 文章太牛了，这样的老总是要受下平..

51CTO推荐博文更多>>

- 【CocosBuilder 开发系列之一】co..
- Ext JS 4.2 Grid预览
- STL——hash\_map【十全十美】
- 一个图形数据维护工具架构设计
- oracle备份恢复之rman恢复到异机
- [C#基础知识]专题十三：全面解析..
- Android开发动画效果被遮掉的解决..
- Redis主从自动failover
- .NET深入解析LINQ框架（四：IQuer..
- .net内存管理与指针

bfOffBits	参数是非常有用的，因为位图信息头和调色板的长度会根据不同情况而变化，所以你可以用这个偏移值迅速的从文件中读取到位数据。
-----------	---

2、位图信息头BITMAPINFOHEADER

结构体定义如下：

```
typedef struct tagBITMAPINFOHEADER { /* bmih */
    DWORD biSize;
    LONG biWidth;
    LONG biHeight;
    WORD biPlanes;
    WORD biBitCount;
    DWORD biCompression;
    DWORD biSizeImage;
    LONG biXPelsPerMeter;
    LONG biYPelsPerMeter;
    DWORD biClrUsed;
    DWORD biClrImportant;
} BITMAPINFOHEADER;
```

其中：

biSize	说明BITMAPINFOHEADER结构所需要的字数。
biWidth	说明图象的宽度，以像素为单位。
biHeight	说明图象的高度，以像素为单位。注：这个值除了用于描述图像的高度之外，它还有另一个用处，就是指明该图像是倒向的位图，还是正向的位图。如果该值是一个正数，说明图像是倒向的，如果该值是一个负数，则说明图像是正向的。大多数的BMP文件都是倒向的位图，也就是时，高度值是一个正数。
biPlanes	为目标设备说明位面数，其值将总是被设为1。
biBitCount	说明比特数/像素，其值为1、4、8、16、24、或32。但是由于我们平时用到的图像绝大部分是24位和32位的，所以我们讨论这两类图像。
biCompression	说明图象数据压缩的类型，同样我们只讨论没有压缩的类型：BI_RGB。
biSizeImage	说明图象的大小，以字节为单位。当用BI_RGB格式时，可设置为0。
biXPelsPerMeter	说明水平分辨率，用像素/米表示。
biYPelsPerMeter	说明垂直分辨率，用像素/米表示。
biClrUsed	说明位图实际使用的彩色表中的颜色索引数（设为0的话，则说明使用所有调色板项）。
biClrImportant	说明对图象显示有重要影响的颜色索引的数目，如果是0，表示都重要。

3、RGB颜色阵列

有关RGB三色空间我想大家都很熟悉，这里我想说的是在Windows下，RGB颜色阵列存储的格式其实BGR。也就是说，对于24位的RGB位图像素数据格式是：

蓝色B值	绿色G值	红色R值
------	------	------

对于32位的RGB位图像素数据格式是：

蓝色B值	绿色G值	红色R值	透明通道A值
------	------	------	--------

透明通道也称Alpha通道，该值是该像素点的透明属性，取值在0（全透明）到255（不透明）之

间。对于24位的图像来说，因为没有Alpha通道，故整个图像都不透明。

二、搞清了文件格式，下一步我们要实现加载。

加载文件的目的是要得到图片属性，以及RGB数据，然后可以将其绘制在DC上(GDI)，或是生成纹理对象(3D:OpenGL/Direct3D)。这两种用途在数据处理上有点区别，我们主要按前一种用法讲，在和3D有不同的地方，我们再提出来。

#### 1、加载文件头

```
//Load the file header
BITMAPFILEHEADER header;
memset(&header, 0, sizeof(header));
inf.read((char*)&header, sizeof(header));
if(header.bfType != 0x4D42)
    return false;
```

这个很简单，没有什么好说的。

#### 2、加载位图信息头

```
//Load the image information header
BITMAPINFOHEADER infoheader;
memset(&infoheader, 0, sizeof(infoheader));
inf.read((char*)&infoheader, sizeof(infoheader));
m_imageWidth = infoheader.biWidth;
m_imageHeight = infoheader.biHeight;
m_iBitsPerPixel = infoheader.biBitCount;
```

这里我们得到了3各重要的图形属性：宽，高，以及每个像素颜色所占用的位数。

#### 3、行对齐

由于Windows在进行行扫描的时候最小的单位为4个字节，所以当

图片宽 X 每个像素的字节数 != 4的整数倍

时要在每行的后面补上缺少的字节，以0填充（一般来说当图像宽度为2的幂时不需要对齐）。位图文件里的数据在写入的时候已经进行了行对齐，也就是说加载的时候不需要再做行对齐。但是这样一来图片数据的长度就不是：宽 X 高 X 每个像素的字节数 了，我们需要通过下面的方法计算正确的数据长度：

```
//Calculate the image data size
int iLineByteCnt = (((m_imageWidth*m_iBitsPerPixel) + 31) >> 5) << 2;
m_imageDataSize = iLineByteCnt * m_imageHeight;
```

#### 4、加载图片数据

对于24位和32位的位图文件，位图数据的偏移量为sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER)，也就是说现在我们可以直接读取图像数据了。

```
if(m_pImageData) delete []m_pImageData;
m_pImageData = new unsigned char[m_imageDataSize];
inf.read((char*)m_pImageData, m_imageDataSize);
```

如果你足够细心，就会发现内存m\_pImageData里的数据的确是BGR格式，可以用个纯蓝色或者是纯红色的图片测试一下。

#### 5、绘制

好了，数据和属性我们都有了，现在就可以拿来随便用了，就和吃馒头一样，爱粘白糖粘白糖，爱粘红糖粘红糖。下面是我的GDI绘制代码，仅作参考。

```
void CImage::DrawImage(HDC hdc, int iLeft, int iTop, int iWidth, int iHeight)
```

```
{  
  
    if(!hdc || m_pImageData == NULL)  
        return;  
  
    BITMAPINFO bmi;  
    memset(&bmi, 0, sizeof(bmi));  
    bmi.bmiHeader.biSize = sizeof(BITMAPINFO);  
    bmi.bmiHeader.biWidth = m_iImageWidth;  
    bmi.bmiHeader.biHeight = m_iImageHeight;  
    bmi.bmiHeader.biPlanes = 1;  
    bmi.bmiHeader.biBitCount = m_iBitsPerPixel;  
    bmi.bmiHeader.biCompression = BI_RGB;  
    bmi.bmiHeader.biSizeImage = m_iImageDataSize;  
    StretchDIBits(hdc, iLeft, iTop, iWidth, iHeight,  
                  0, 0, m_iImageWidth, m_iImageHeight,  
                  m_pImageData, &bmi, DIB_RGB_COLORS, SRCCOPY);  
}
```

## 6、3D(OpenGL)的不同之处

如果你想用刚才我们得到的数据生成纹理对象，那么你还得请出下面的问题。

首先，用来生成纹理的数据不需要对齐，也就是说不能在每行的后面加上对齐的字节。当然在OpenGL里要求纹理图片的尺寸为2的幂，所以这个问题实际上不存在；

其次，我们得到的图形数据格式是BGR(BGRA)，所以在生成纹理的时候，需指定格式为GL\_BGR\_EXT(GL\_BGRA\_EXT)；否则需要做BGR->RGB(BGRA->RGBA)的转化。

分享至:

0

收藏 +



tongy0 1人

了这篇文章

类别: [数字电视](#) | [技术圈\(00\)](#) | [阅读\(33403340\)](#) | [评论\(00\)](#) | [推送到技术圈](#) | [返回首页](#)

上一篇 [Mpeg2 视频编码概述](#) 下一篇 [bitmap分析——另一篇](#)

### 相关文章

[TEST用windows live客户端写博客.](#)

[年轻时给百度写的一封信,想想那时候的自己真..](#)

[Bitmap](#)

[GDI Graphics In Delphi\(Drawing Bitmaps, t..](#)

[快速浏览Silverlight3 beta:鸡肋一样的Writa..](#)

[测试客户端写日志功能!](#)



### 文章评论

#### 发表评论

[点击回复你评论的文章地址及楼层，赢51CTO新年大礼包！](#)

昵 称:

[登录](#) [快速注册](#)

验证码:

请点击后输入验证码 [博客过2级，无需填写验证码](#)

内 容: