


# BMP格式详解

原创 Ezioooooo 于 2015-12-12 08:58:14 发布 44893 收藏 222 版权

分类专栏: BMP解析 文章标签: bmp

BMP解析 专栏收录该内容

4 订阅 1 篇文章 订阅专栏

BMP（全称 **Bitmap**）是Windows操作系统中的标准图像文件格式，可以分成两类：设备相关位图（DDB）和设备无关位图（DIB），使用非常广。它采用位映射存储格式，除了图像深度可选以外，不采用其他任何压缩，因此，BMP文件所占用的空间很大。BMP文件的图像深度可选1bit、4bit、8bit及24bit。BMP文件存储数据时，图像的扫描方式是按从左到右、从下到上的顺序。由于BMP文件格式是Windows环境中交换与图有关的数据的一种标准，因此在Windows环境中运行的图形图像软件都支持BMP图像格式。

这里通过一个具体的例子对BMP格式做一个简单的介绍。

## 1、整体信息

BMP格式的文件从头到尾依次是如下信息：

- bmp文件头（bmp file header）：共14字节；
- 位图信息头（bitmap information）：共40字节；
- 调色板（color palette）：可选；
- 位图数据；

最常见的就是24位图，所谓的24位图，就是说一个像素的颜色信息用24位来表示，也就是说，对于三原色BRG，每一个颜色都用以字节（8）位来表示。除了24位图，还有1位（单色），2位（4色，CGA），4位（16色，VGA），8位（256色），16位（增强色），24位（真彩色）和32位等。  
下面通过下面的图片做详细介绍：



图像的部分信息如下：

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f																																																																																																																																																																																																												
00000000h:	42	4D	36	B4	04	00	00	00	00	00	36	04	00	00	28	00	; BM6?....6...(. 00000010h:	00	00	80	02	00	00	E0	01	00	01	00	08	00	00	00	; ..€...?..... 00000020h:	00	00	00	00	04	00	C4	0E	00	00	C4	0E	00	00	00	01	; ...?.?..?.... 00000030h:	00	00	00	01	00	00	01	10	37	00	00	10	49	00	00	18	; .....7...I... 00000040h:	44	00	01	1D	58	00	08	23	4D	00	01	22	55	00	00	18	; D...X...#M.."U... 00000050h:	64	00	04	25	51	00	00	22	5F	00	04	26	63	00	0E	2E	; d..%Q..."_..&c... 00000060h:	5D	00	00	26	72	00	0D	2C	62	00	06	2D	63	00	00	2A	; ]..&r...b...-c...* 00000070h:	6E	00	18	31	58	00	01	2E	68	00	02	30	68	00	0C	2E	; n..1[...k...0h... 00000080h:	6B	00	04	27	81	00	04	2B	7D	00	03	32	78	00	03	35	; k..'?.+}..2x..5 00000090h:	74	00	0A	35	72	00	19	31	75	00	0C	32	78	00	00	32	; t..5r...1u...2x..2 000000a0h:	7E	00	00	2E	89	00	00	38	79	00	00	36	7D	00	17	38	; ~...?.8y..6}...; 000000b0h:	6C	00	15	38	73	00	00	33	86	00	04	38	85	00	0A	3C	; l..8s...3?.8?.< 000000c0h:	7C	00	0D	39	81	00	03	3B	82	00	00	39	88	00	00	3C	;  ..9?;?.9?.<

## 2、bmp文件头 ( bmp file header )

bmp文件头包含如下信息：

- bfType：2字节，文件类型；
- bfSize：4字节，文件大小；
- bfReserved1：2字节，保留，必须设置为0；
- bfReserved2：2字节，保留，必须设置为0；
- bfOffBits：4字节，从头到位图数据的偏移；

下图的数据就是bmp文件头：

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f																																																																																						
00000000h:	42	4D	36	B4	04	00	00	00	00	36	04	00	00	28	00		; BM6?....6...(. 00000010h:	00	00	80	02	00	00	E0	01	00	00	01	00	08	00	00	00	; ..€...?..... 00000020h:	00	00	00	00	04	00	C4	0E	00	00	C4	0E	00	00	00	01	; ...?.?..?.... 00000030h:	00	00	00	01	00	00	01	10	37	00	00	10	49	00	00	18	; .....7...I... 00000040h:	44	00	01	1D	58	00	08	23	4D	00	01	22	55	00	00	18	; D...X...#M.."U... 00000050h:	64	00	04	25	51	00	00	22	5F	00	04	26	63	00	0E	2E	; d..%Q..."_..&c...

一共14字节，下面逐个解释。

0-1: bfType，表示文件类型，BMP格式的文件这两个字节是0x4D42，10进制就是19778，字符显示就是'BM'；

2-5: bfSize，表示文件的大小，这里的是0x0004B436，十进制是308278，也就是301kb，检查文件信息，验证正确；

6-7: bfReserved1，保留位，必须设置为0；

8-9: bfReserved2，保留位，必须设置为0；

a-d: bfOffBits，4字节的偏移，表示从文件头到位图数据的偏移，这里是0x00000436，十进制是1078，后面会做验证；

## 3、位图信息头 ( bitmap information )

位图信息头一共40字节，包含如下内容：

- biSize：4字节，信息头的大小，即40；
- biWidth：4字节，以像素为单位说明图像的宽度；
- biHeight：4字节，以像素为单位说明图像的高度，同时如果为正，说明位图倒立（即数据表示从图像的左下角到右上角），如果为负说明正向；
- biPlanes：2字节，为目标设备说明颜色平面数，总被设置为1；
- biBitCount：2字节，说明比特数/像素数，值有1、2、4、8、16、24、32；
- biCompression：4字节，说明图像的压缩类型，最常用的就是0（BI\_RGB），表示不压缩；
- biSizeImages：4字节，说明位图数据的大小，当用BI\_RGB格式时，可以设置为0；
- biXPelsPerMeter：表示水平分辨率，单位是像素/米，有符号整数；
- biYPelsPerMeter：表示垂直分辨率，单位是像素/米，有符号整数；
- biClrUsed：说明位图使用的调色板中的颜色索引数，为0说明使用所有；
- biClrImportant：说明对图像显示有重要影响的颜色索引数，为0说明都重要；

下图数据是位图信息头：

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f																																																																					
00000000h:	42	4D	36	B4	04	00	00	00	00	00	36	04	00	00	28	00	; BM6?....6...(. 00000010h:	00	00	80	02	00	00	E0	01	00	00	01	00	08	00	00	00	; ..€...?..... 00000020h:	00	00	00	00	04	00	C4	0E	00	00	C4	0E	00	00	00	01	; ...?.?..?.... 00000030h:	00	00	00	01	00	00	01	10	37	00	00	10	49	00	00	18	; .....7...I... 00000040h:	44	00	01	1D	58	00	08	23	4D	00	01	22	55	00	00	18	; D...X...#M.."U...

一共40字节，解释如下：



Ezioooooo

关注



68

0e-11: 4字节的biSize, 这里是0x28, 即十进制的40, 验证正确;

12-15: 4字节的biWidth, 这里是0x00000280, 即十进制的640, 用像素表示图像的宽度, 查看文件信息验证正确;

16-19: 4字节的biHeight, 这里是0x000001E0, 即十进制的480, 用像素表示图像的高度, 查看文件信息验证正确; 同时, 这是一个正数, 表示图像是倒立的, 即图像数据是从左下角到右上角排列的;

1a-1b: 2字节的biPlanes, 值为0x0001;

1c-1d: 2字节的biBitCount, 值是0x0008, 即8, 表示每个像素用8位表示, 一共有256个颜色;

1e-21: 4字节的biCompression, 值是0, 即BI\_RGB格式, 不压缩;

22-25: 4字节的biSizelImage, 图像的大小, 值是0x0004B000, 十进制为307200, 由上面的bfSize (文件大小) 和bfOffBits (文件头到数据的偏移) 分别是308278和1078可以得到, biSizelImage=bfSize-bfOffBits, 即图像大小=文件大小-偏移量;

26-29: 4字节的biXPelsPerMeter, 水平分辨率, 值是0x00000EC4, 十进制3780;

2a-2d: 4字节的biYPelsPerMeter, 垂直分辨率, 值是0x00000EC4, 十进制3780;

2e-31: 4字节的biClrUsed, 使用的颜色索引数, 值是0x00000100, 十进制256, 与1c-1d得到的结论一致;

32-35: 4字节的biClrImportant, 重要的颜色索引数, 值是0x00000100, 十进制256;

4、调色板 (Color Palette)

调色板是可选的, 不过这里的8位色图有调色板。那么接下来的数据就是调色板了。调色板就是一个颜色的索引, 这里是8位色图, 一共有256中颜色, 由于每个颜色都有RGB三原色, 也就是要3个字节表示, 这样的话256个颜色就不能表示所有的颜色, 所以就需要一个索引, 用一个字节的索引指向4个字节表示的颜色 (RGB加上Alpha值)。如果把这4个字节表示为一个Color类型, 那么调色板就是Color的数组。由于Color类型也是一个数组, 调色板就像一个二维数组palette[N][4], 其中N是颜色的数量, 这里就是256。因此, 这个例子中的调色板的大小就是256x4=1024字节, 在调色板之前, 有14字节的bmp文件头, 40字节的位图信息头, 加上1024字节的调色板, 一共1078字节, 也就是说真正的图像数据前面有1078字节, 这和bmp文件头中的bfOffBits相符, 验证了我们的讨论。

有的图像没有调色板, 比如下面的24位色图:



头部数据如下:

```
0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 42 4D 72 BA 01 00 00 00 00 00 36 00 00 00 28 00 ; BMr?.....6...(.
00000010h: 00 00 C6 01 00 00 53 00 00 00 01 00 18 00 00 00 ; ..?.S.....
00000020h: 00 00 3C BA 01 00 00 00 00 00 00 00 00 00 00 00 ; ..<?.....
00000030h: 00 00 00 00 00 00 01 A7 E4 01 A7 E4 01 A7 E4 01 ; .....T.T.T.
00000040h: A7 E4 01 A7 E4 01 A7 E4 01 A7 E4 01 A7 E4 01 A7 ; T.T.T.T.T.?
```

根据上面的讨论可以知道, biBitCount是24 (0x18), bfOffBits是54 (0x36), 即没有调色板, 位图信息头接下来就是图像数据了。

调色板中的数据每4字节一组, 分别表示蓝、绿、红和Alpha值。按照第一个图像举例来说:

索引	蓝	绿	红	Alpha
0	01	10	37	00
1	00	10	49	00
2	00	18	44	00
3	01	1D	58	00

5、位图数据

接下来就是位图数据了。由于是8位色图, 所以每个像素用1个字节表示, 取出每个字节, 显示到相应的设备上就可以了。

注意, 这里的biHeight为正数, 说明图像倒立, 从左下角开始到右上角, 以行为主序排列。

如果是24位色图, 按照BGR的顺序排列, 32位色图按照BGRAAlpha排列。

位图数据排列还有一个规则, 就是对齐。

Windows默认的扫描的最小单位是4字节, 如

获取速度等都是有很大的增益的。因此, BMI

据的长度必须是4的倍数，如果不够需要进行比特填充（以0填充），这样可以达到按行的快速存取。这样的话，位图数据的大小就不一定是宽x高x每像素字节数了，因为每行还可能有0填充。  
填充后的每行数据如下：

$$RowSize = 4 * \left\lceil \frac{BPP * Width}{32} \right\rceil$$

其中，BPP是每像素的比特数（Bits Per Pixel），即biBitCount，Width是宽度，单位是像素即bfWidth。  
对于我们这个例子，BPP是8，Width是480，正好是4的倍数，也就是没有填充。来计算一下：  
RowSize=4\*(8\*480/32)=480字节，验证没有填充。  
那么以上面第二个图片24位色图为例，按照数据可以得到：

- biBitCount=0x0018=24;
- bfWidth=0x000001c6=454;
- bfHeight=0x00000053=83;
- biSizeImage=0x0001BA3c=113212;

按照没填充计算：454\*83\*3=113046 bytes，与真实值相差166字节。  
按照填充公式，每行有数据4\*(24\*454/32)=1364 字节，真正的数据有454\*3=1362字节，也就是说每行填充了2字节0，一共83行，共填充83\*2=166字节，验证了我们的讨论。  
在程序中，我们可以用下面的代码计算每行的数据：

```
int bytesPerLine=((bfWidth*biBitCount+31)>>5)<<2;
```

那么，位图数据大小为：

```
int imageSize=bytesPerLine*bfHeight;
```

这样的话，每扫描完一行数据，最后的几个字节可能是填充的0，需要跳过：

```
int skip=4-((bfWidth*biBitCount)>>3)&3;
```

C语言读取、存储、显示BMP图像08-26

在C语言下，可读取、存储、显示BMP图像，对BMP图像格式进行了详细剖析。

BMP图片详解weixin\_46987028的博客 682

BMP文件格式，又称为Bitmap（位图）或是DIB(Device-Independent Device，设备无关位图)，...

评论 12

您还未登录，请先

登录

后发表或查看评论

bmp格式说明\_android\_cai\_niao的博客\_bmp图片格式详解6-7

bmp图片格式详解 1. BMP文件简介 BMP(Bitmap-File)图形文件是Windows采用的图形文件格式...

BMP文件格式解析游一法师 1142

BMP文件格式解析 作者：水木子 文章目录一、图像...

BMP图像格式详解yesheng1989 2819

一. 简介 BMP(Bitmap-File)图形文件是Windows采用的图形文件格式，在Windows环境下运行...

图形图像基础 之 bmp介绍runafterhit的博客 1740

文章目录一、bmp相关概念位图Bitmap 与 矢量图Vector--用点表示 还是 用公式表示bmp（Bit...


BMP文件格式详解（BMP file format）qq\_41137110的博客 3145

BMP文件格式，又称为Bitmap（位图）或是DIB(Device-Independent Device，设备无关位图)，...

bmp图片文件格式 最新发布qq\_50998958的博客 655

（参考大佬的博客，自己做个记录）经典图片为例：建议用010EDITOR打开，再套上官方BM...

BMP图片-文件头部信息

 Ezioooooo

关注

 68

5/7