位图文件格式浅析和图形颜色转换

Prepared by Jetty
Date:2005.1.14

SW Department
PCEG LDC

Agenda

- BMP图象文件格式分析
- 真彩色BMP图象到256色BMP的转换
- 真彩色BMP图象到16色BMP的转换

- BMP是bitmap的缩写形式，bitmap顾名思义，就是位图也即Windows位图。它一般由4部分组成：文件头信息块、图像描述信息块、颜色表（在真彩色模式无颜色表）和图像数据区。在系统中以BMP为扩展名保存。
- 位图是最常用的windows图形格式,通过windows API函数可以直接读取并绘制,不过,有时(比如使用windows API有限制的时候),我们还是需要自己控制,那么,就让我们看看它的格式吧!

BMP图象文件格式

bitmap-file header
bitmap-information header
color table
图象数据阵列字节
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx

位图文件的组成	结构名称
位图文件头(bitmap-file header)	BITMAPFILEHEADER
位图信息头(bitmap-information header)	BITMAPINFOHEADER
彩色表(color table)	RGBQUAD
图象数据阵列字节	BYTE

位图文件头

- 位图文件头(bitmap-file header)包含有关于文件类型、文件大小、图象信息偏移量等信息：
- typedef struct tagBITMAPFILEHEADER { /* bmfh */

  UINT bfType;
  DWORD bfSize;
  UINT bfReserved1;
  UINT bfReserved2;
  DWORD bfOffBits;

  } BITMAPFILEHEADER;

位图信息

- 位图信息(bitmap-information)用BITMAPINFO结构来定义，它由位图信息头(bitmap-information header)和彩色表(color table)组成，前者用BITMAPINFOHEADER结构定义，后者用RGBQUAD结构定义。BITMAPINFO结构具有如下形式：
- typedef struct tagBITMAPINFO { /* bmi */

  BITMAPINFOHEADER bmiHeader;
  RGBQUAD bmiColors[1];

  } BITMAPINFO;

位图信息头

- 位图信息头(bitmap-information header) BITMAPINFOHEADER结构包含有位图文件的大小、压缩类型和颜色格式，其结构定义为:
- 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;

彩色表

- 彩色表包含(color table)的元素与位图所具有的颜色数相同，象素的颜色用RGBQUAD结构来定义。对于24-位真彩色图象就不使用彩色表（同样也包括16位、和32位位图），因为位图中的RGB值就代表了每个象素的颜色。彩色表中的颜色按颜色的重要性排序，这可以辅助显示驱动程序为不能显示足够多颜色数的显示设备显示彩色图象。RGBQUAD结构描述由R、G、B相对强度组成的颜色，定义如下：
- typedef struct tagRGBQUAD { /* rgbq */

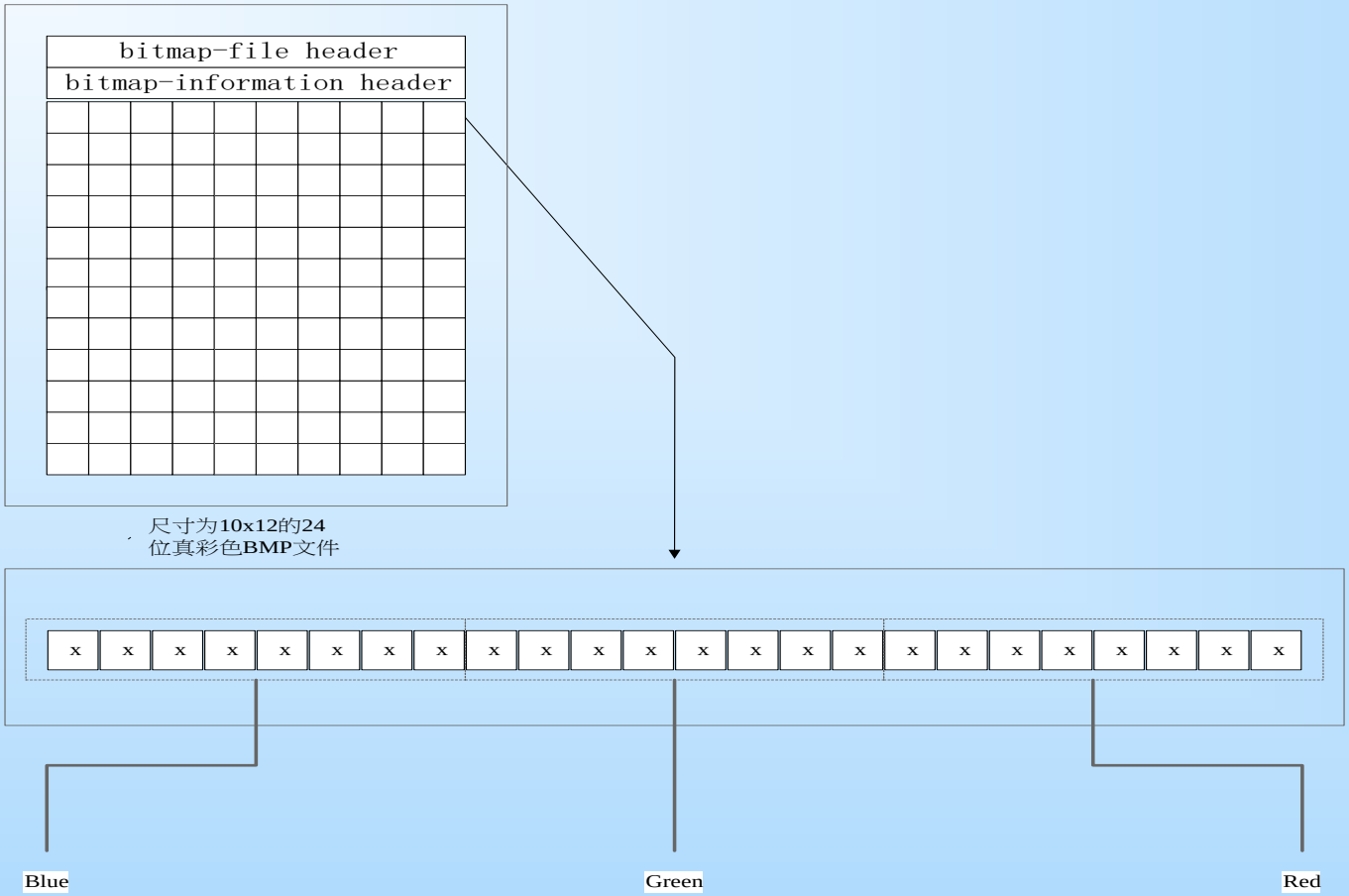
  BYTE rgbBlue;
  BYTE rgbGreen;
  BYTE rgbRed;
  BYTE rgbReserved;

  } RGBQUAD;

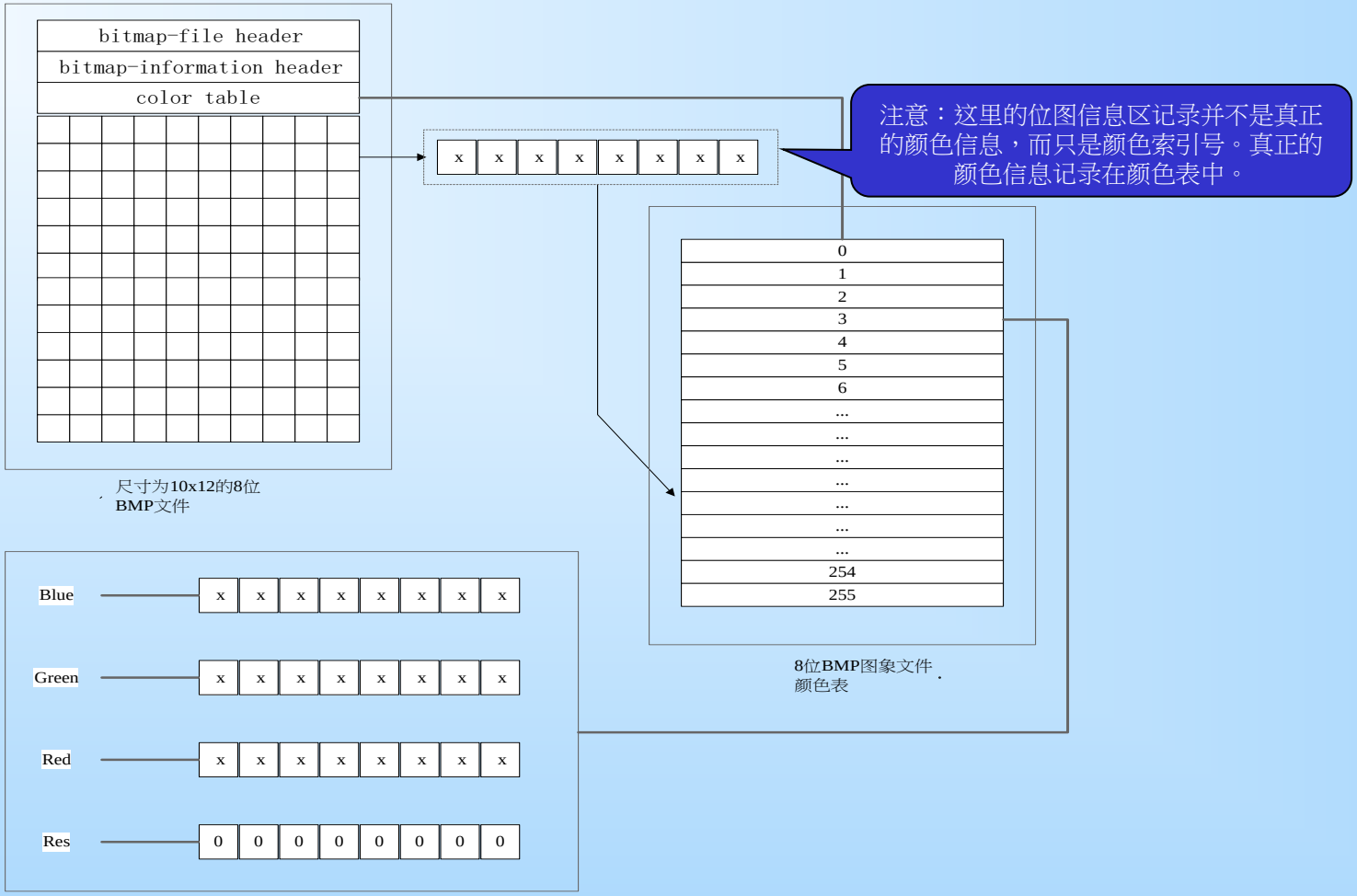
位图数据

- 紧跟在彩色表之后的是位图文件的图像数据区。图象的每一扫描行由表示图象象素的连续的字节组成，每一行的字节数取决于图象的颜色数目和用象素表示的图象宽度。扫描行是由底向上存储的，这就是说，阵列中的第一个字节表示位图左下角的象素，而最后一个字节表示位图右上角的象素。
- 在此部分记录着每点像素对应的颜色号，其记录方式也随颜色模式而定，即2色图像每点占1位（8位为1字节）；16色图像每点占4位（半字节）；256色图像每点占8位（1字节）；真彩色图像每点占24位（3字节）。所以，整个数据区的大小也会随之变化。究其规律而言，可的出如下计算公式：图像数据信息大小=（图像宽度*图像高度*记录像素的位数）/8。

24位真彩色BMP文件格式



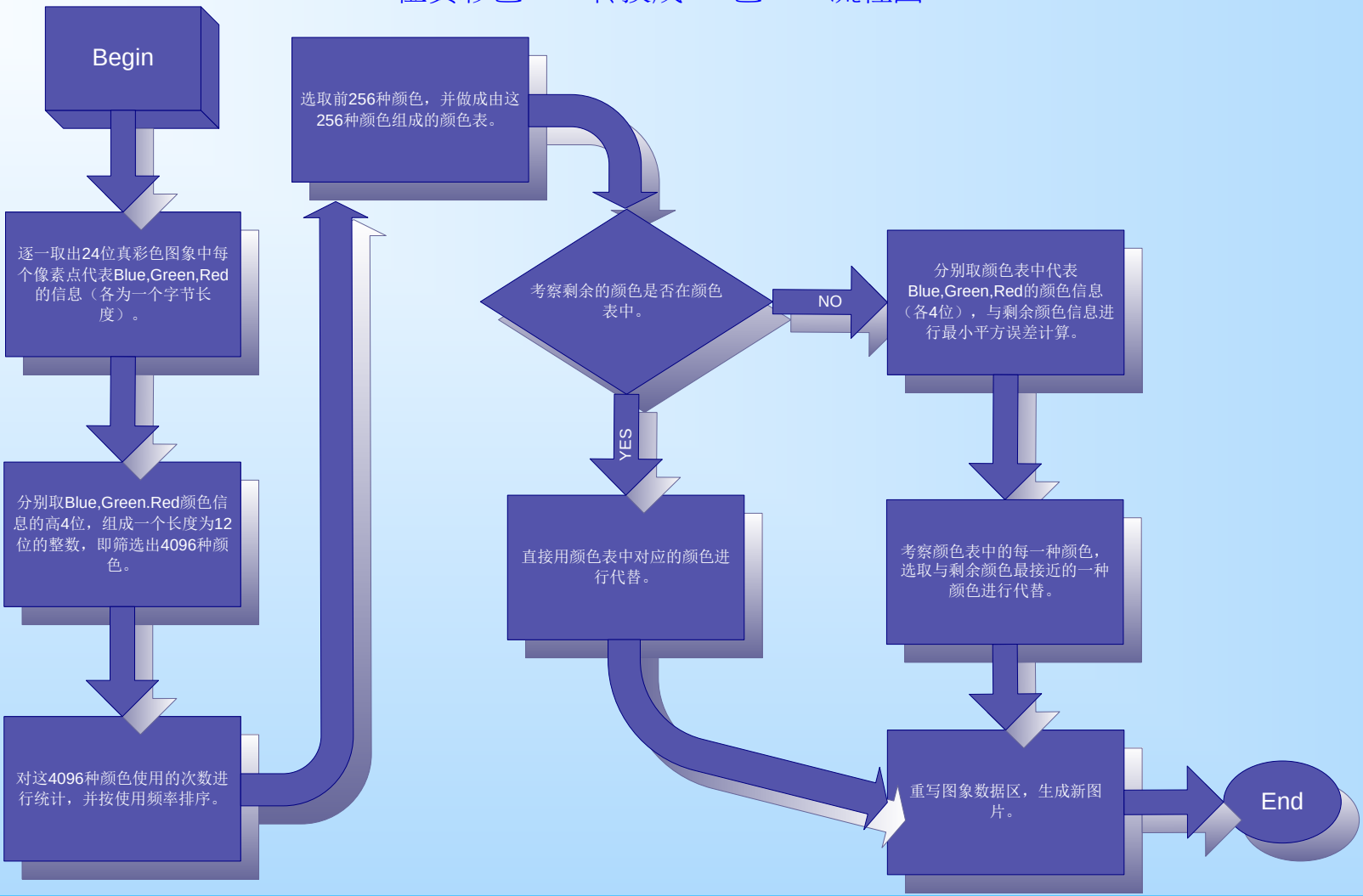
8位BMP文件格式


bitmap-file header
bitmap-information header
color table
x x x x x x x x
注意：这里的位图信息区记录并不是真正的颜色信息，而只是颜色索引号。真正的颜色信息记录在颜色表中。
0
1
2
3
4
5
6
...
...
...
...
...
...
...
254
255
尺寸为10x12的8位BMP文件
8位BMP图象文件颜色表
Blue
Green
Red
Res
0 0 0 0 0 0 0 0

- 了解了BMP图象文件格式后，我们再讨论一下如何将真彩色BMP图片转换成256色BMP图片，甚至16色BMP图片的问题。
- 虽然真彩色图片所占比例越来越大，但在某些场合，仍需要256色甚至16色图片，于是便需要对真彩色图片进行转换。
- 我们已经知道，24位真彩色BMP图片最多可以显示2的24次方（16777216）种颜色，怎样从中选出256种颜色，又要使颜色的失真比较小，这是一个比较复杂的问题。
- 一种简单的做法是将Red:Green:Blue以3:3:2表示，即取Red，Green的高3位，Blue的高2位，组成一个字节，这样就可以表示256种颜色了。但不难想象，这种方法三原色失去平衡，失真肯定很严重。这种方法将不被采取。

- 我们的做法是，对真彩色原图中的每一个像素点，分别取Blue,Green,Red的高4位，组成一个12位整数。这样原图缩减为4096种颜色，这其中，有些颜色可能用了很多次，有些颜色可能一次没用过。再在这4096种颜色中选出使用得最多的256种颜色。剩下的颜色我们也不能简单的丢弃，而是要用已经筛选出的256种颜色来进行替代。替代的算法是用最小平方误差找出最接近的颜色值。

24位真彩色BMP转换成256色BMP流程图



原图对比

- 看看转换成256色的BMP图和24位真彩色原图的比较吧。

真彩色原图

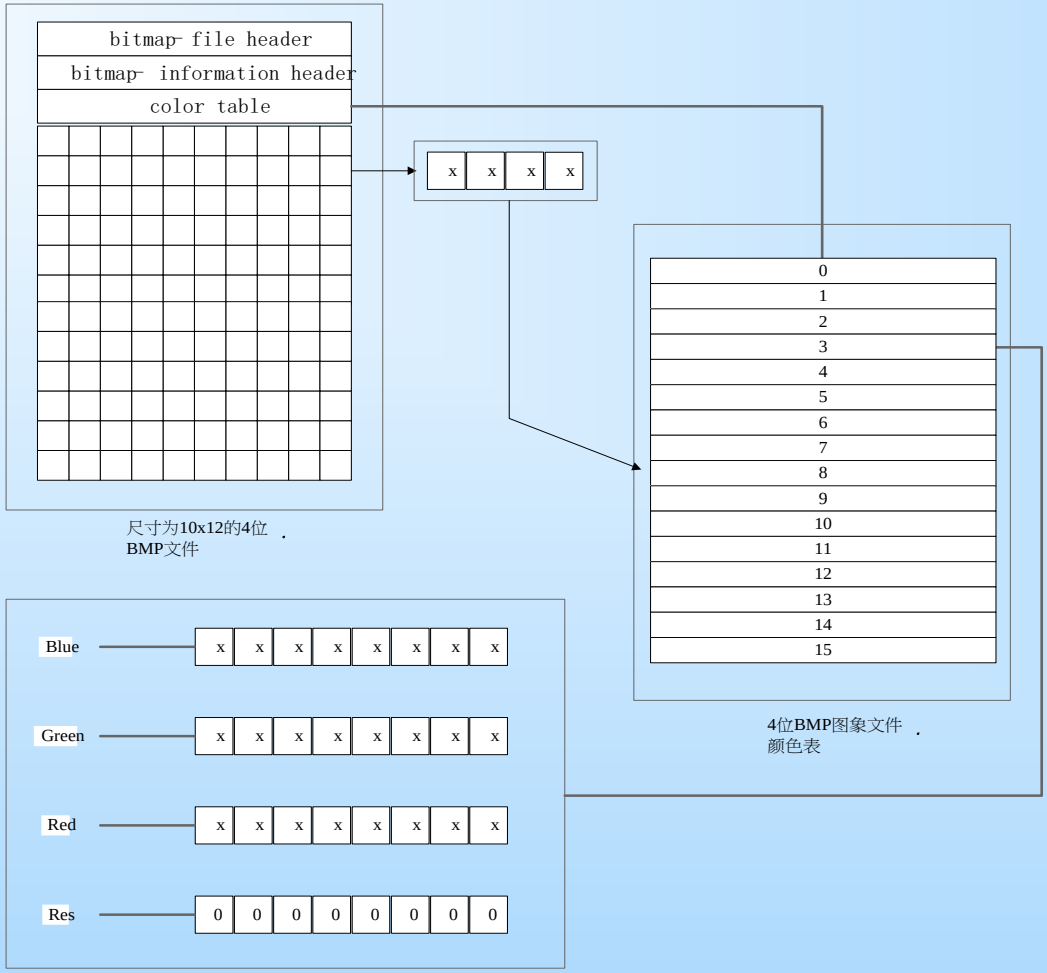
256色BMP

- 接下来讨论24位真彩色图片到16色BMP图片的转换。
- 我们知道，16色BMP图片只能显示16种颜色，这样的图片在显示效果上显然很难让人满意，但在某些场合却必须使用16色图片。
- 根据前面的了解，我们可以知道16色BMP图片至少应该有如下特点：

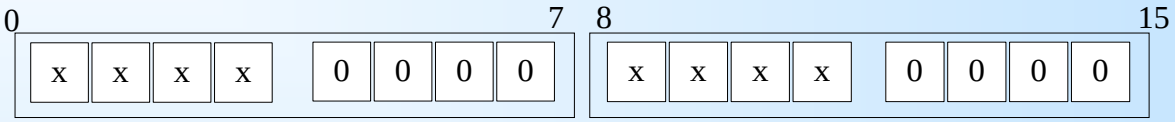
  颜色表中有16种颜色，即颜色表的大小为16*4Bytes；

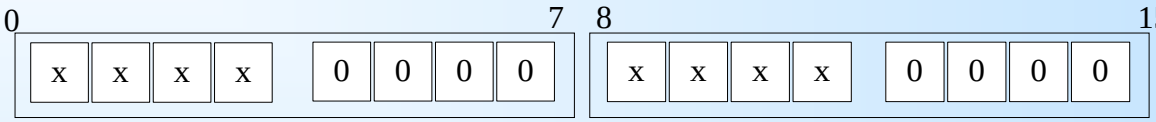
  每个像素点的大小为4位，即半个字节。
- 来看一下它的文件格式。

4位BMP文件格式


bitmap- file header
bitmap- information header
color table
x x x x
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
尺寸为10x12的4位BMP文件
4位BMP图象文件颜色表
Blue
Green
Red
Res
x x x x x x x x
0 0 0 0 0 0 0 0

- 怎样将24位真彩色BMP图片转换成16色BMP图片？事实上，我们使用的转换方法和前面提到的将24位真彩色BMP图片转换成256色BMP图片的基本相同。
- 首先也要将真彩色图片中2的24次方种颜色缩减为4096种颜色，再从其中筛选中使用得最多的16种颜色，也用最小平方误差的方法找出近似颜色进行替代。
- 但是，在重写图象数据区，生成新图片的时候，我们就要注意了。新的16色BMP图片图象数据区的颜色索引号只有4位，即半个字节，但在进行内存读写的时候，最小单位是8位，即1个字节。那么，如果按老办法进行，将会出现什么样的情况呢？

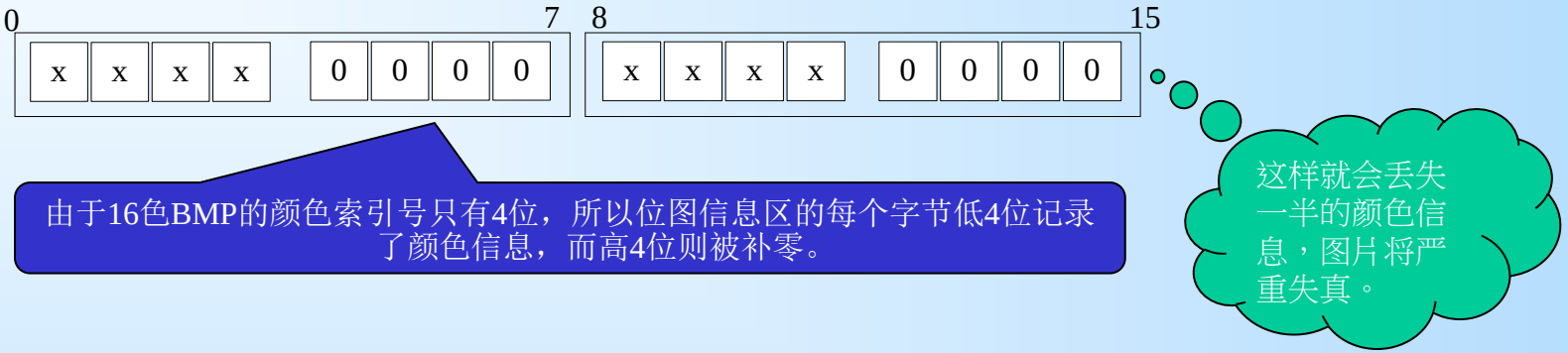

0
7
8
15
x x x x 0 0 0 0
x x x x 0 0 0 0

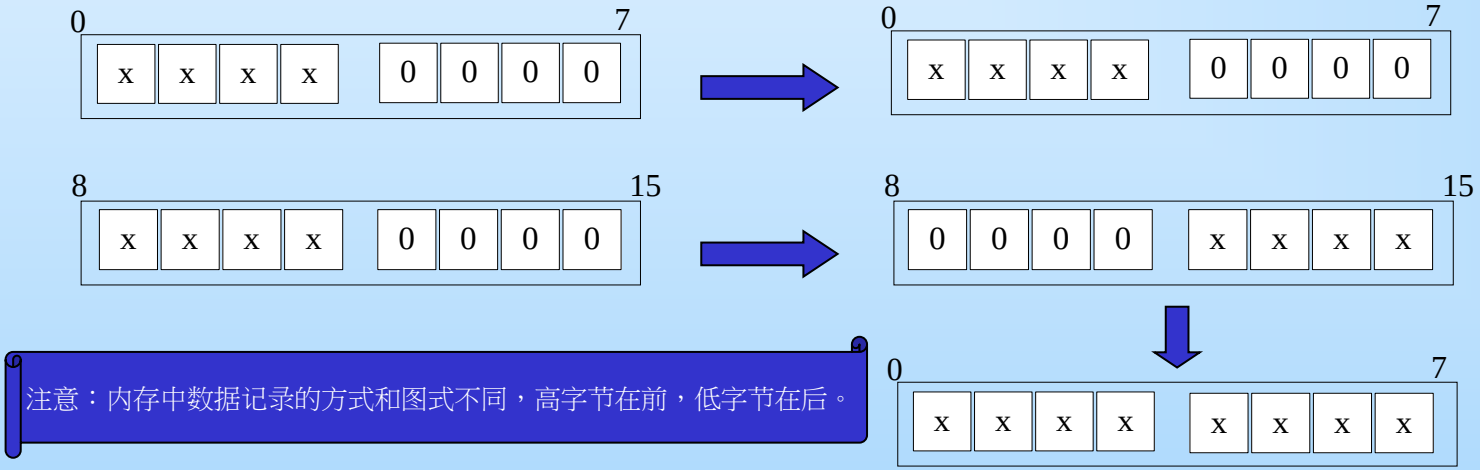


由于16色BMP的颜色索引号只有4位，所以位图信息区的每个字节低4位记录了颜色信息，而高4位则被补零。

这样就会丢失一半的颜色信息，图片将严重失真。

所以，我们要这样做……


0
7
8
15
x x x x 0 0 0 0
x x x x 0 0 0 0
由于16色BMP的颜色索引号只有4位，所以位图信息区的每个字节低4位记录了颜色信息，而高4位则被补零。
这样就会丢失一半的颜色信息，图片将严重失真。


0
7
x x x x 0 0 0 0
0
7
x x x x 0 0 0 0
8
15
x x x x 0 0 0 0
8
15
0 0 0 0 x x x x
注意：内存中数据记录的方式和图式不同，高字节在前，低字节在后。
0
7
x x x x x x x x

原图对比

- 让我们再看看转换成16色的BMP图和24位真彩色原图的比较吧。

真彩色原图

16色BMP

效果差得比较远，但是对于只有16种颜色的图片，也只能如此了……

总结

BMP图形颜色转换的几个关键：

- 重写位图文件头和信息头，重点是信息头。
- 重建颜色表。
- 重写位图信息区，即填写颜色索引号。位图是否能正确显示，关键在于填写颜色索引号是否正确。