The BMP (bitmap) Image-file Format

The BMP (bitmap) format is a standard image-file format developed by Microsoft and used by Windows Operating systems and others. It is a very simple format that offers a choice of several simple compression schemes as well as an option for no-compression. The BMP format provides a good introduction to the use of headers and look-up tables in uncompressed and compressed image and video files. Video teleconferencing and MPEG video coding, as we'll see later, make a similar use of headers in their formats.

5.1 Five parts of a BMP File

Every BMP file has 5 parts (as explained later, parts 3 and 4 may be absent).

1) A bitmap file header (14 bytes) is the 1st 14 bytes in a BMP image-file. The bitmap-file header, called a BITMAPFILEHEADER structure, contains information about the type, size, and layout of the bitmap file and image as shown in Figure 5.1below.

2) A bitmap information header (40 bytes) follows the BITMAPFILEHEADER. The bitmap-information header, defined as a BITMAPINFOHEADER structure, specifies the dimensions, compression type, and color format for the bitmap as shown in Figure 5.2 below.

3) A color table (may be absent).

The color table is not present in the file for bitmaps that use 24-color-bits per pixel.

In this class, we will only work with BMP images that have 24-bits per pixel; therefore we will not use color tables.

4) Filler-bytes (may be absent) may follow the color table (if present) or the BITMAPINFOHEADER if the color table is not present. These filler-bytes would typically be a company name, or a product name, regarding some aspect of the BMP picture.

In this class, we will not use any filler-bytes; i.e. we will not load any additional bytes for a company name, etc.

5) Image-data, a sequence of bytes that define the value at each pel in the image, completes the BMP-image-file. The image-data stores the image in row-order beginning at the bottom of the image. The number of bytes representing an image-row depends on the number of colors in the image, the width of the image in pixels, and the choice of compression. Each image-row must be zero-padded to end on a 32-bit boundary. That is, the number of bytes required to store each image-row must be an integer multiple of 4. You must append binary 0's, as needed, to the end of each image-line (in the BMP file) to satisfy this constraint. For pels that occupy a portion of a byte, the bits are filled from MSb to LSb, with zero-padding bits filling out the byte where necessary.

$$\#bytes\_in\_imagedata\_row = 4\ Int\left\{\frac{Int\left\{\frac{(width)(bpp)+7}{8}\right\}+3}{4}\right\}$$

$$\#\_filler\_bits = 32 \quad Int\left\{\frac{(width)(bpp)+31}{32}\right\} - (width)(bpp)$$

BYTE locations	Description	Binary Value	Comments
bytes 1-2 (2) (**Type**)	BMP file designation	01001101 01000010 a 16-bit word, LSB 1st	(66) (77) decimal 66 is stored as 0th byte
bytes 3-6 (4) (**Size**)	total # bytes in BMP image-file	0 to 4,294,967,295 (decimal) = max unsigned 4-byte value	LeastSigByte 1st; MostSigByte last
bytes 7-8 (2) (**Reserved1**)	reserved bytes	0000 0000 0000 0000	must be 0's
bytes 9-10 (2) (**Reserved2**)	reserved bytes	0000 0000 0000 0000	must be 0's
bytes 11-14 (4) (**OffsetBytes**)	offset (in bytes) from start of file to image-data.	0 to 4,294,967,295 (decimal)	Least Signif Byte 1st; Most Signif Byte last

Figure 5.1 The 14-byte BITMAPFILEHEADER that begins a BMP file.

BYTE locations	Description	Binary Value	Comments
bytes 1-4 (4) (**Size**)	# bytes in BITMAPINFOHEADER	(0000 0000 0000 0000 0000 0000 0010 1000)	= 40 LSByte 1st
bytes 5-8 (4) (**Width**)	Image-width in pixels		LSByte 1st
bytes 9-12 (4) (**Height**)	Image-height in pixels		LSByte 1st
bytes 13-14 (2) (**Planes**)	# planes in image	(0000 0000 0000 0001)	typically = 1 LSByte 1st
bytes 15-16 (2) (**BitCount**)	# of bits per pixel		= 1, 4, 8, 16, 24, or 32 LSByte 1st
bytes 17-20 (4) (**Compression Type**)	Type of Compression LSByte 1st		0: no compression 1: RLE8 2: RLE4 3: Bitfield compr. for BitCount=1
bytes 21-24 (4) (**ImageDatasize**)	Total # bytes in compressed/uncompressed imagedata (to include padding bits in each image row to 4-byte boundaries). LSByte 1st		**ImageDatasize** = (bytes per line)* (# lines) If ImageDatasize = 0, display SW computes ImageDatasize
bytes 25-28 (4) (**PixelsPerMeter_X**)	# horiz pixels per meter	arbitrary value	(Not used). LSByte 1st
bytes 29-32 (4) (**PixelsPerMeter_Y**)	# vert pixels per meter	arbitrary value	(Not used). LSByte 1st
bytes 33-36 (4) (**Colors_Used**)	# of entries in palette; i.e. # RGBQUAD structures in the color table	LSByte 1st	# of RGBQUAD structures to follow
bytes 37-40 (4) (**Colors_Important**)	# of significant colors LSByte 1st		typically = # of entries in palette.

Figure 5.2 The 40-byte BITMAPINFOHEADER that follows the BITMAPFILEHEADER in a BMP file.

All of components of the BITMAPFILEHEADER are explained within Figure 5.1. Have a close look at this figure. The same holds for the BITMAPINFOHEADER, except for its entry BitCount, which we'll discuss here. In the BITMAPINFOHEADER, the entry for BitCount (in the 15th and 16th bytes) can be set to 1, 4, 8, 16, 24, or 32 and its value tells an image display software how to decode the bits that follow the BITMAPINFOHEADER.
We will only use BitCount=24.

BitCount=24 denotes 24 bpp (24-bits-per-pixel)
In 24-bit BMP files, the color for each pel is denoted by 3-BYTES containing a red, green and blue values. Blue is stored as the least-significant-byte, green next, then red. The least significant byte of each pel is written to the BMP file first, the most significant byte is written 3rd. No color table is stored in the image file. No padding-bytes are added between pels in the image data. Of course, BMP images stored in 24bpp must have each image-line end on a 4-byte boundary. That is, each image-line of the image-data should be zero-padded (where necessary) to so that the image-data for each image line is an integer-multiple of 4-bytes. Also, with BitCount=24, bytes 33-36 (**Colors_Used**) and bytes 37-40 (**Colors_Important**) of the BITMAPINFOHEADER should be set to 0.

Again, the BMP format stores the image in "bottom-up" order. That is, the bottom row of pels is stored first, the top row of pels is stored last. Also, each image-line of the image-data must be zero-padded (where necessary) so that the image-data for each image line occupies an integer-multiple of 4-bytes.