Chapter 4
Storing a Digital Image

4.1 Storing an Image as a File

Each of the methods of creating a digital image described in Chap. 3, (and of transforming an image in Chaps. 5 and 7), concludes with a display of the image, usually in an editor such as Paint. To store an image, the option is as follows:

File
—Save as
——Filename
———Save as type.

The stored image is then accessible as a file with the chosen filename and extension.

4.2 Image File

Computer storage is organized as nested *directories* or folders, containing *files*, each with a *filename* and *extension*. The extension indicates the type of file, each of which has a specific *file format* [1]. We can access stored files by means of a file manager utility, such as Microsoft File Explorer, which opens, closes, copies, pastes or deletes files.

A file holds a sequence of *fields*; a field holds one or more *bytes*; and a byte holds eight *binary digits* 0 or 1. The file format defines the position of each field, and what each field represents. For example, Fig. 4.1 shows the .BMP image file format. For user convenience, a file is usually displayed in hexadecimal digits 0 to F [2], so a byte is written as two hex digits 00 to FF. For example, Fig. 4.2 shows a simple image and its .BMP file.

4.3 File Format .BMP

.BMP (for bitmap) is a venerable image file format developed by Microsoft for Windows operating systems, in successive versions [3]. It is an uncompressed format, and has large file size. It is a simple format, very widely available for ordinary use,

A. Parkin, *Computing Colour Image Processing*,
https://doi.org/10.1007/978-3-319-74076-8_4

Group	*Field*	*Type*	*dec.*	*hex.*	*Remarks*
FileHeader			1-14	01-0E	File type, properties
	bfType	integer	1-2	01-02	Always 'BM'
	bfSize	long	3-6	03-06	File length, bytes
	bfReserved1	integer	7-8	07-08	Always 0
	bfReserved2	integer	9-10	09-0A	Always 0
	bfOffBits	long	11-14	0B-0E	Header length, bytes
InfoHeader			15-54	0F-36	Image size, properties
	biSize	long	15-18	0F-12	Infoheader len., bytes
	biWidth	long	19-22	13-16	Image width, pixels
	biHeight	long	23-26	17-1A	Image height, pixels
	biPlanes	integer	27-28	1B-1C	Always 1
	biBitCount	integer	29-30	1D-1E	Bits per pixel = 24
	biCompression	long	31-34	1F-22	Not compr. = 0
	biSizeImage	long	35-38	23-26	Image data len., bytes
	biXPelsPerMeter	long	39-42	27-2A	Printer p.p.m., horiz.
	biYPelsPerMeter	long	43-46	2B-2E	Printer p.p.m., vert.
	biClrUsed	long	47-50	2F-32	Always 0
	biClrImportant	long	51-54	33-36	Clrs considered imp.
Image data			55-	37-	In scan sequence
	aBitmapBits[0]	byte	55	37	Pixel 0, blue value
	aBitmapBits[1]	byte	56	38	Pixel 0, green value
	aBitmapBits[2]	byte	57	39	Pixel 0, red value
	aBitmapBits[3]	byte	58	3A	Pixel 1, blue value
	aBitmapBits[4]	byte	59	3B	Pixel 1, green value
	aBitmapBits[5]	byte	60	3C	Pixel 1, red value
	etc. ...				

Fig. 4.1 The .BMP 24-bit format, with two headers and a body as long as it takes

though other formats with compression are often preferred. Figure 4.1 shows the structure. Figure 4.3 shows a typical camera image stored in .BMP format displayed in a Paint editor, at 100% size and magnified to 400% to show the pixel detail.

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	42	4D	F6	00	00	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	08	00	00	00	08	00	00	00	01	00	18	00	00	00
00000020	00	00	C0	00	00	00	C3	0E	00	00	C3	0E	00	00	00	00
00000030	00	00	00	00	00	00	FF	FF	FF	00	00	00	FF	FF	FF	00
00000040	00	00	FF	FF	FF	00	00	00	FF	FF	FF	FF	00	00	00	00
00000050	00	FF	FF	FF	00	00	00	FF	FF	FF	00	00	00	FF	FF	FF
00000060	00	00	00	FF	FF	FF	FF	FF	FF	00	00	00	FF	FF	FF	00
00000070	00	00	FF	FF	FF	00	00	00	FF	FF	FF	00	00	00	00	00
00000080	00	FF	FF	FF	1D	E6	B5	1D	E6	B5	1D	E6	B5	FF	FF	FF
00000090	00	00	00	FF	FF	FF	FF	FF	FF	00	00	00	1D	E6	B5	00
000000A0	FF	00	1D	E6	B5	00	00	00	FF	FF	FF	00	00	00	00	00
000000B0	00	FF	FF	FF	1D	E6	B5	1D	E6	B5	1D	E6	B5	FF	FF	FF
000000C0	00	00	00	FF	FF	FF	FF	FF	FF	00	00	00	FF	FF	FF	00
000000D0	00	00	FF	FF	FF	00	00	00	FF	FF	FF	00	00	00	00	00
000000E0	FF	FF	FF	FF	00	00	00	FF	FF	FF	00	00	00	FF	FF	FF
000000F0	00	00	00	FF	FF	FF										

Fig. 4.2 A simple image and its .BMP file in hexadecimal dump, two hex digits per byte. In multi-byte fields, most-significant bytes are to the right, least-significant left. Header fields, above the thick line, are: two-byte "magic cookie" 42 4Dh = ASCII BM; four-byte file size F6h = 246d; two two-byte fields ignored; four-byte header length 36h = 54d; four-byte info header 28h = 40d; four-byte image width 08h = 8d; four-byte image height 08h = 8d; two-byte planes field 01h = 1d; two-byte bits per pixel 18h = 24d; four-byte compression 00h = 0d; image data length C0h = 192d; two four-byte pixels per meter C3 0Eh = 49934d; four-byte colors used 00h = 0d; and four-byte colors important 00h = 0d. After the thick line: sixty-four triples of one-byte fields showing the (B, G, R) values of the image pixels in scan-sequence starting at bottom left

4.4 File Format .GIF

.GIF (for graphic interchange format) was developed in 1987 for 8-bit displays limited to 256 colours [4]. It has been largely replaced by .PNG, but is still widely available for website use. It has lossless compression, hence small file size. Figure 4.4 shows a typical camera image stored in .GIF format displayed in a Paint editor, at 100%

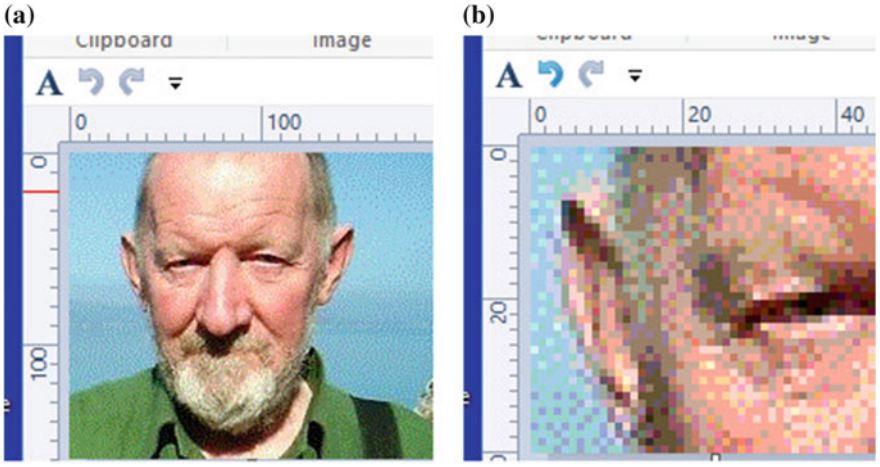

Fig. 4.3 BMP format. **a** Image 190 × 160 px stored in .BMP format; file size 89 KB. **b** Magnified to 400%

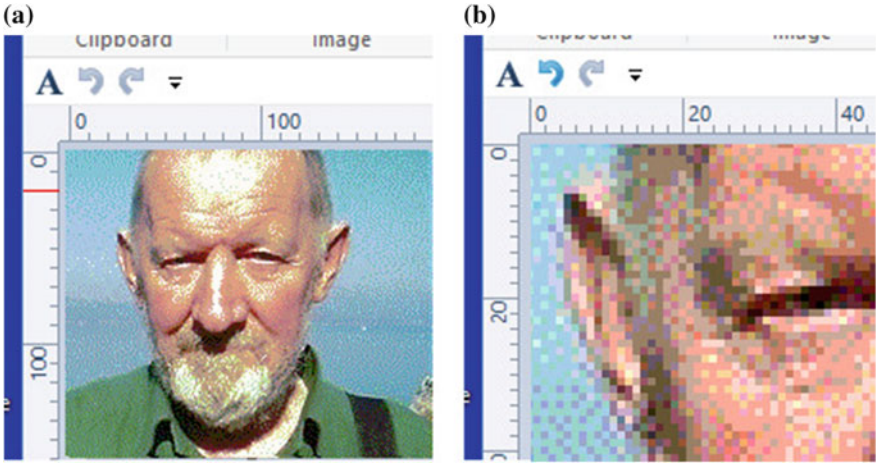

Fig. 4.4 GIF format. **a** Image 190 × 160 px stored in .GIF format; file size 20 KB. **b** Magnified to 400%

size and magnified to 400% to show the pixel detail. The original sRGB colours are severely altered.

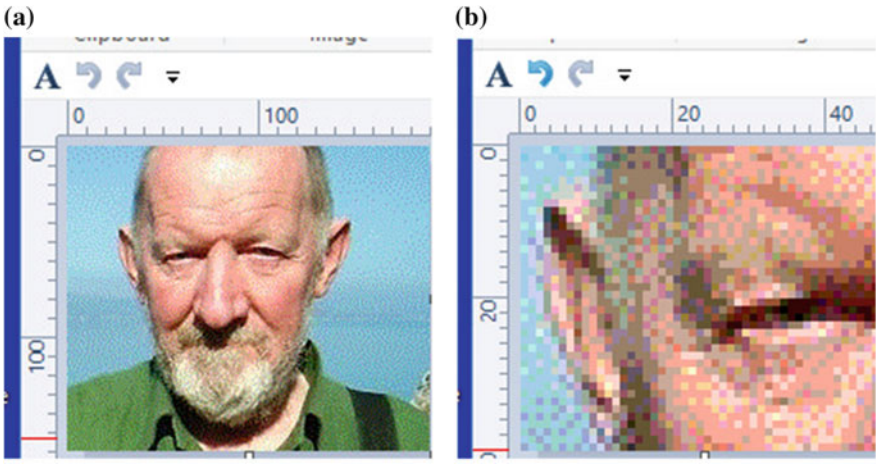

Fig. 4.5 PNG format. **a** Image 190 × 160 px stored in .PNG format; file size 23 KB. **b** Magnified to 400%

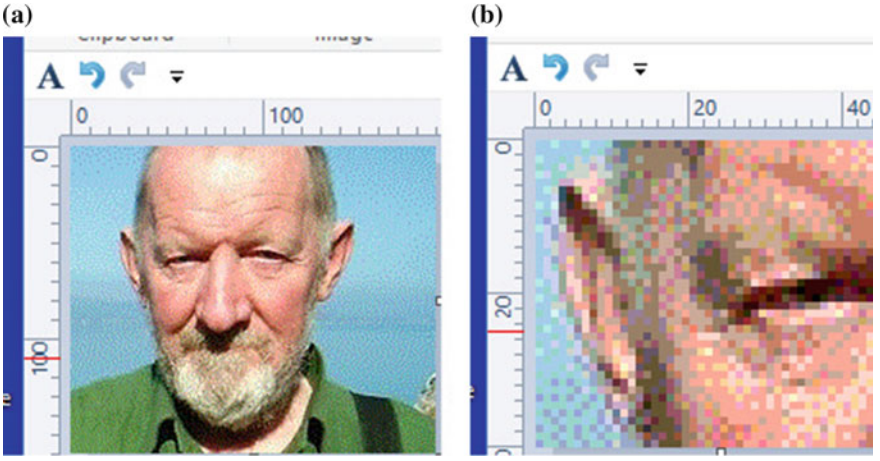

Fig. 4.6 TIF format. **a** Image 190 × 160 px stored in .TIF format; file size 26 KB. **b** Magnified to 400%

4.5 File Format .PNG

.png (for portable network graphics) was developed in 1996 as a replacement for .BMP and .GIF [5]. It has lossless compression, hence small file size. Figure 4.5 shows a typical camera image stored in .png format displayed in a Paint editor, at 100% size and magnified to 400% to show the pixel detail.

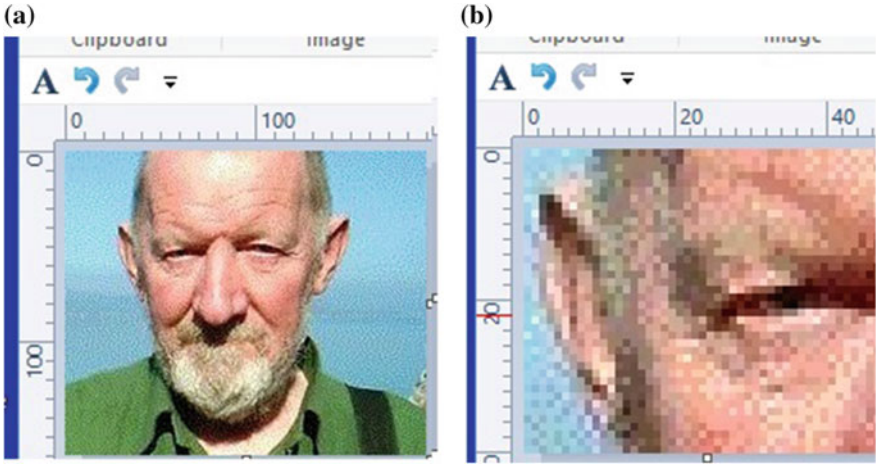

Fig. 4.7 JPG format, high quality. **a** Image 190 × 160 px stored in .JPG format with 94% quality; file size 20 KB. **b** Magnified to 400%

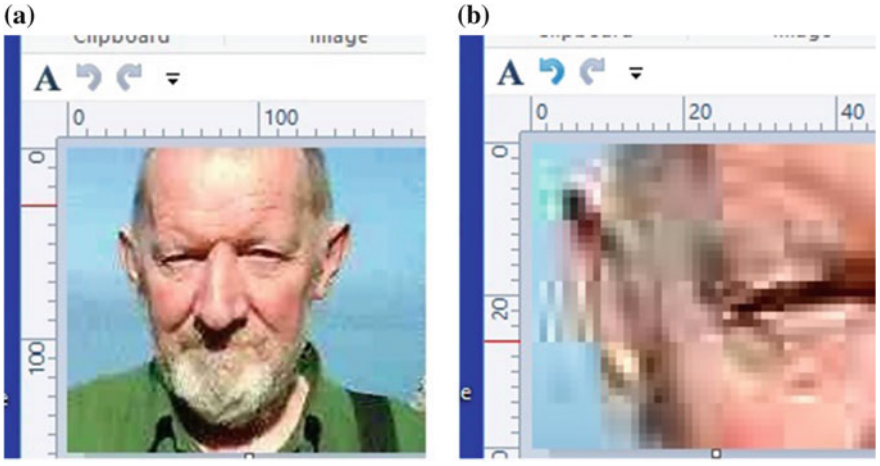

Fig. 4.8 JPG format, low quality **a** Image 190 × 160 px stored in .JPGG format with 20% quality; file size 2 KB. **b** Magnified to 400%

4.6 File Format .TIF

.TIF (for tagged image file) was developed in 1986 for desktop and commercial printing, where it remains the preferred format [6]. It has lossless compression, hence small file size, and can contain additional image information. Figure 4.6 shows a typical camera image stored in .TIF format displayed in a Paint editor, at 100% size and magnified to 400% to show the pixel detail.

4.7 File Format .JPG

.JPG (for joint photographic experts group) was developed in 1992 for making adjustable-quality compressions of camera and scanner images by removing the least visually significant image data [7]. It is very widely used to store and transfer such images. Figure 4.7 shows a typical camera image stored in .JPG format with 94% quality, and Fig. 4.8 with 20% quality.

References

1. Image file formats. https://en.wikipedia.org/wiki/Image_file_formats
2. Hexadecimal number system. https://en.wikipedia.org/wiki/Hexadecimal
3. BMP file format. https://en.wikipedia.org/wiki/BMP_file_format
4. GIF file format. https://en.wikipedia.org/wiki/GIF
5. PNG file format. https://en.wikipedia.org/wiki/Portable_Network_Graphics
6. TIF file format. https://en.wikipedia.org/wiki/TIFF
7. JPEG file format. https://en.wikipedia.org/wiki/JPEG