Robotic Vision:
Assignment 2:
Graphic Files Representation

José Ramón Romero Chávez (A01700318)
A01700318@itesm.mx

Tecnológico de Monterrey, Campus Querétaro
5th February 2019

Introduction

The image file format allows us to organize and store digital data (images), that can be rasterized for use on a computer display or printer.
An image file format may store data in multiple ways: uncompressed or compressed and vector or rasterized format.

Compression

There are two types of image file compression algorithms: lossless and lossy.

Lossless compression algorithms reduce file size while preserving a perfect copy of the original uncompressed image. Lossless compression generally, but not always, results in larger files than lossy compression. Lossless compression should be used to avoid accumulating stages of re-compression when editing images.

Lossy compression algorithms preserve a representation of the original uncompressed image that may appear to be a perfect copy, but it is not a perfect copy. Often lossy compression is able to achieve smaller file sizes than lossless compression. Most lossy compression algorithms allow for variable compression that trades image quality for file size.

Raster Graphics / Vector Graphics

Vector graphics are computer graphics images that are defined in terms of 2D points, which are connected by lines and curves to form polygons and other shapes. Its importance relies in the capacity changing their size without any problem, compared with raster graphic that have a defined size, position and color.

BMP (BitMaP image file)

- Raster/ vector: Raster
- Color depth: 1, 4, 8, 16, 24, 32, and 64
- Indexed color: Yes
- Transparency: Yes
- Metadata: No
- Interlacing: No
- Multi-page: No
- Animation: No
- Layers: No
- Color management: Yes
- Extendable: No
- HDR format: No
- Compression algorithm: None, RLE, JPEG, and PN

Basic BMP File Format

Name			Size	Description
Header			14 bytes	Windows Structure: BITMAPFILEHEADER
	Signature		2 bytes	'BM'
	FileSize		4 bytes	File size in bytes
	reserved		4 bytes	unused (=0)
	DataOffset		4 bytes	File offset to Raster Data
InfoHeader			40 bytes	Windows Structure: BITMAPINFOHEADER
	Size		4 bytes	Size of InfoHeader =40
	Width		4 bytes	Bitmap Width
	Height		4 bytes	Bitmap Height
	Planes		2 bytes	Number of Planes (=1)
	BitCount		2 bytes	Bits per Pixel 1 = monochrome palette. NumColors = 1 4 = 4bit palletized. NumColors = 16 8 = 8bit palletized. NumColors = 256 16 = 16bit RGB. NumColors = 65536 (?) 24 = 24bit RGB. NumColors = 16M
	Compression		4 bytes	Type of Compression 0 = BI_RGB no compression 1 = BI_RLE8 8bit RLE encoding 2 = BI_RLE4 4bit RLE encoding
	ImageSize		4 bytes	(compressed) Size of Image It is valid to set this =0 if Compression = 0
	XpixelsPerM		4 bytes	horizontal resolution: Pixels/meter
	YpixelsPerM		4 bytes	vertical resolution: Pixels/meter
	ColorsUsed		4 bytes	Number of actually used colors
	ColorsImportant		4 bytes	Number of important colors 0 = all
ColorTable			4 * NumColors bytes	present only if Info.BitsPerPixel <= 8 colors should be ordered by importance
		Red	1 byte	Red intensity
		Green	1 byte	Green intensity
		Blue	1 byte	Blue intensity
		reserved	1 byte	unused (=0)
	repeated NumColors times			
Raster Data			Info.ImageSize bytes	The pixel data

GIF (Graphics Interchange Format)

- Raster/ vector: Raster
- Color depth: Indexed (1bpc to 8bpc) and grayscale (8bpc)
- Indexed color: Yes
- Transparency: Yes; 1bpp plane-only mask using color index
- Metadata: Yes
- Interlacing: Yes
- Multi-page: Yes
- Animation: Yes
- Layers: Yes
- Color management: No
- Extendable: Yes (GIF89a)
- HDR format: No
- Compression algorithm: LZW

GIF File Format
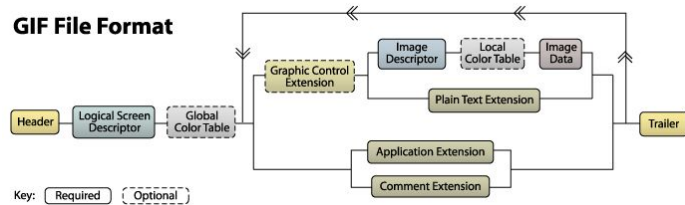
```
byte#  hexadecimal  text or
(hex)               value      Meaning
0:     47 49 46
       38 39 61     GIF89a     Header
                               Logical Screen Descriptor
6:     90 01        400          – width in pixels
8:     90 01        400          – height in pixels
A:     F7                        – GCT follows for 256 colors with resolution 3 x 8bits/primary
B:     00           0            – background color #0
C:     00                        – default pixel aspect ratio
D:                              Global Color Table
:
30D:   21 FF                    Application Extension block
30F:   0B           11            – eleven bytes of data follow
310:   4E 45 54
       53 43 41
       50 45        NETSCAPE     – 8-character application name
       32 2E 30     2.0          – application "authentication code"
31B:   03           3            – three more bytes of data
31C:   01           1            – data sub-block index (always 1)
31D:   FF FF        65535        – unsigned number of repetitions
31F:   00                        – end of App Extension block
320:   21 F9                    Graphic Control Extension for frame #1
322:   04           4            – four bytes of data follow
323:   08                        – bit-fields 3x:3:1:1, 000|010|0|0 -> Restore to bg color
324:   09 00                     – 0.09 sec delay before painting next frame
326:   00                        – no transparent color
327:   00                        – end of GCE block
328:   2C                       Image Descriptor
329:   00 00 00 00  (0,0)        – NW corner of frame at 0, 0
32D:   90 01 90 01  (400,400)    – Frame width and height: 400 x 400
331:   00                        – no local color table; no interlace
332:   08           8           LZW min code size
333:   FF           255          – 255 bytes of LZW encoded image data follow
334:                data
433:   FF           255          – 255 bytes of LZW encoded image data follow
                    data

                  :
92BA:  00                        – end of LZW data for this frame
92BB:  21 F9                    Graphic Control Extension for frame #2
 :                                   :
153B7B:21 F9                    Graphic Control Extension for frame #44
 :
15CF35:3B                       File terminator
```

JPEG / JFIF (Joint Photographic Experts Group / File Interchange Format)

- Raster/ vector: Raster
- Color depth: 8-bit(greyscale), 12-bit, and 24-bit
- Indexed color: No
- Transparency: No
- Metadata: Yes
- Interlacing: Yes
- Multi-page: No
- Animation: No
- Layers: No
- Color management: Yes
- Extendable: No
- HDR format: No (see unofficial JPEG-HDR)
- Compression algorithm: Lossy (and partly lossless), DCT, RLE, and Huffmanpredictive nearest neighbor
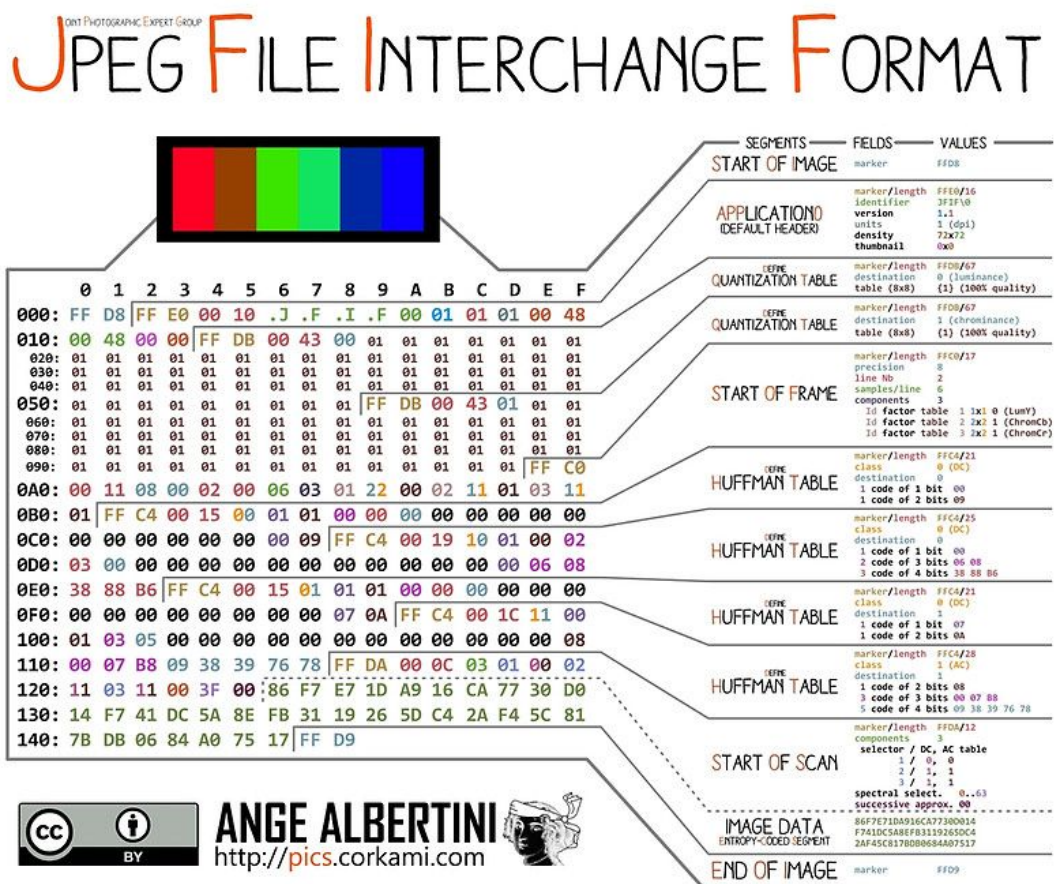
JPEG IS THE ENCODING STANDARD, JFIF IS THE FILE FORMAT

PGM (Portable GrayMap), PPM (Portable PixMap) & PBM (Portable BitMap)

- Raster / vector: Raster
- Indexed color: No
- Transparency: No
- Metadata: Yes
- Interlacing: No
- Multi-page: Yes
- Animation: No
- Layers: No
- Color management: No
- Extendable: No
- HDR format: No
- Compression algorithm: None
- Structure
  - Identification (PX)
 - Cols and Rows
  - Max value for each color

Type	Magic number		Extension	Colors
	ASCII	Binary		
Portable BitMap[1]	P1	P4	.pbm	0–1 (white & black)
Portable GrayMap[2]	P2	P5	.pgm	0–255 (gray scale)
Portable PixMap[3]	P3	P6	.ppm	0–255 (RGB)

PNG (Portable Network Graphics)

- Raster / vector: Raster
- Color depth: bitmap(1bpc), indexed(8bpc), grayscaleand RGB(8bpc, 16bpc)
- Indexed color: Yes
- Transparency: Yes; indexed, grayscale and RGB
- Metadata: Yes
- Interlacing: Yes, Adam7 algorithm
- Multi-page: No
- Animation: No (but see MNG and APNG)
- Layers: No
- Color management: Yes
- Extendable: Yes, via chunks
- HDR format: No, see discussion
- Compression algorithm: Lossless and DEFLATE
- Structure:
  - A PNG file consists of a PNG signature followed by a series of chunks. This chapter defines the signature and the basic properties of chunks.
  - The first eight bytes of a PNG file always contain the following (decimal) values:
    - 137 80 78 71 13 10 26 10
  - This signature indicates that the remainder of the file contains a single PNG image, consisting of a series of chunks beginning with an IHDR chunk and ending with an IEND chunk.
  - Chunk layout:
    - Length.:
      - A 4-byte unsigned integer giving the number of bytes in the chunk's data field. The length counts only the data field, not itself, the chunk type code, or the CRC. Zero is a valid length.
    - Chunk Type:
      - A 4-byte chunk type code. For convenience in description and in examining PNG files, type codes are restricted to consist of uppercase and lowercase ASCII letters (A-Z and a-z, or 65-90 and 97-122 decimal). However, encoders and decoders must treat the codes as fixed binary values, not character strings.
    - Chunk Data:
      - The data bytes appropriate to the chunk type, if any. This field can be of zero length.
    - CRC:
      - A 4-byte CRC (Cyclic Redundancy Check) calculated on the preceding bytes in the chunk, including the chunk type code and chunk data fields, but not including the length field. The CRC is always present, even for chunks containing no dat

TIF o TIFF (Tagged Image File Format)

- Compression algorithm: None, LZW, RLE, ZIP, and other
- Raster/ vector: Both
- Color depth: 1, 2, 4, 8, 16, 24, and 32
- Indexed color: Yes (1–8 bit modes)
- Transparency: Yes
- Metadata: Yes
- Interlacing: Yes, for JPEG compression
- Multi-page: Yes
- Animation: No
- Layers: Yes
- Color management: Yes
- Extendable: Yes, via tags
- HDR format: Yes, TIFF float
- Structure:
    - Today, TIFF is a standard file format found in most paint, imaging, and desktop publishing programs and is a format native to the Microsoft Windows GUI. TIFF's extensible nature, allowing storage of multiple bitmap images of any pixel depth, makes it ideal for most image storage needs
    - The image data is stored using a bitmap or a compression algorithm
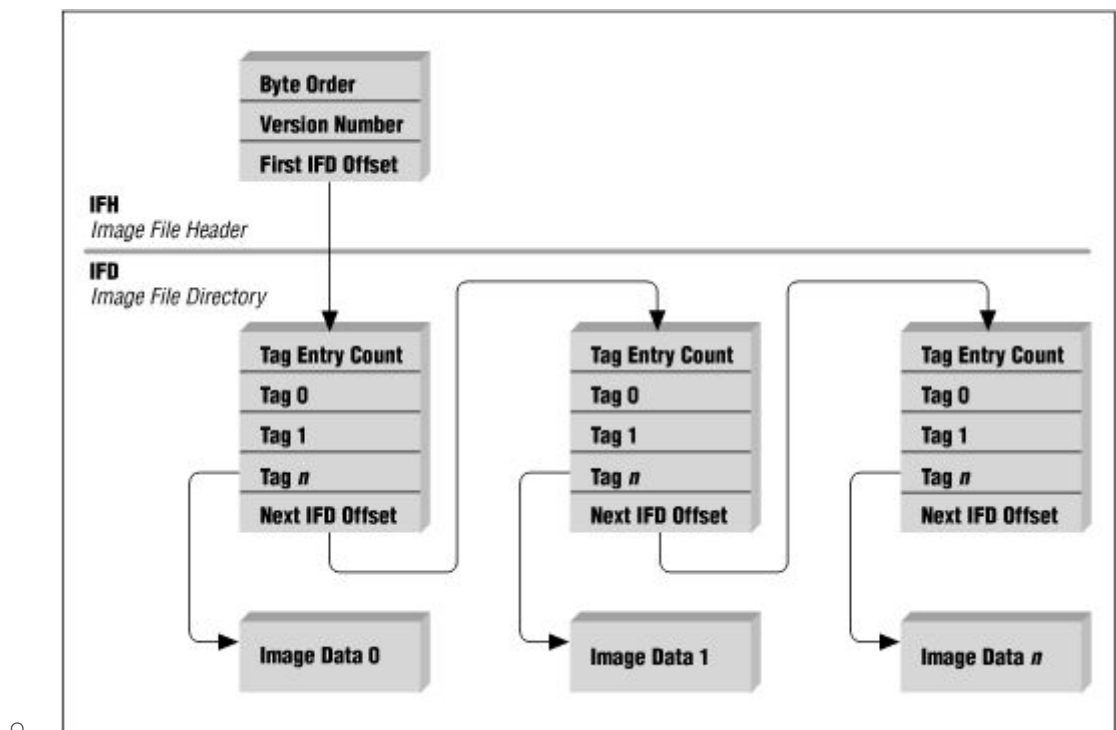
References:

[1]"THE BMP FILE FORMAT", *Ece.ualberta.ca*, 2019. [Online]. Available: http://www.ece.ualberta.ca/~elliott/ee552/studentAppNotes/2003_w/misc/bmp_file_format/ bmp_file_format.htm. [Accessed: 05- Feb- 2019].

[2]"Structure of BMP file", *Ue.eti.pg.gda.pl*, 2019. [Online]. Available: http://www.ue.eti.pg.gda.pl/fpgalab/zadania.spartan3/zad_vga_struktura_pliku_bmp_en.htm l. [Accessed: 05- Feb- 2019].

[3]"The Book of GIF - Rigor", *Rigor*, 2019. [Online]. Available: https://rigor.com/resources/ebooks/the-book-of-gif. [Accessed: 05- Feb- 2019].

[4]"Cover Sheet for the GIF89a Specification", *GIF89*, 2019. [Online]. Available: https://www.w3.org/Graphics/GIF/spec-gif89a.txt. [Accessed: 05- Feb- 2019].

[5]"3MF Project: What's In A GIF - Bit by Byte", *Matthewflickinger.com*, 2019. [Online]. Available: http://www.matthewflickinger.com/lab/whatsinagif/bits_and_bytes.asp. [Accessed: 05- Feb- 2019].

[6]"GIF", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/GIF#Image_coding. [Accessed: 05- Feb- 2019].

[7]"Netpbm format", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Netpbm_format. [Accessed: 05- Feb- 2019].

[8]"PNG Specification: File Structure", *Libpng.org*, 2019. [Online]. Available: http://www.libpng.org/pub/png/spec/1.2/PNG-Structure.html. [Accessed: 05- Feb- 2019].

[9]"TIFF: Summary from the Encyclopedia of Graphics File Formats", *Fileformat.info*, 2019. [Online]. Available: https://www.fileformat.info/format/tiff/egff.htm. [Accessed: 05- Feb- 2019].

[10]"Comparison of graphics file formats", *En.wikipedia.org*, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Comparison_of_graphics_file_formats#Technical_details. [Accessed: 05- Feb- 2019].