

# ECE160

# Multimedia

## Lecture 3: Spring 2011

### Graphics and Image Data Representations

# Graphics and Image Data Representations

- **Graphics/Image Data Types**
- The number of file formats used in multimedia continues to proliferate. For example, a list of some file formats used in the popular product Macromedia Director.

| File Import   |              |                              |              |  | File Export |              | Native               |
|---|--------------|------------------------------|--------------|--|-------------|--------------|----------------------|
| Image   | Palette      | Sound                        | Video        | Anim.  | Image       | Video        |                      |
| .BMP, .DIB,<br>.GIF, .JPG,<br>.PICT, .PNG,<br>.PNT, .PSD,<br>.TGA, .TIFF,<br>.WMF | .PAL<br>.ACT | .AIFF<br>.AU<br>.MP3<br>.WAV | .AVI<br>.MOV | .DIR<br>.FLA<br>.FLC<br>.FLI<br>.GIF<br>.PPT | .BMP        | .AVI<br>.MOV | .DIR<br>.DXR<br>.EXE |

# 1-bit Images

- Each pixel is stored as a single bit (0 or 1), so also referred to as **binary image**.
- Such an image is also called a 1-bit **monochrome** image or a pure black/white image since it contains no color.
- We show a sample 1-bit monochrome image (called “Lena” by multimedia scientists - this is a standard image used to illustrate many algorithms).



# 8-bit Gray-scale Images

- Each pixel has a gray-value between 0 and 255. Each pixel is represented by a single byte; e.g., a dark pixel might have a value of 10, and a bright one might be 230.
- **Bitmap:** The two-dimensional array of pixel values that represents the graphics/image data.
- **Image resolution** refers to the number of pixels in a digital image (higher resolution always yields better quality).
  - Fairly high resolution for such an image might be 1600x1200, whereas lower resolution might be 640x480.

# Multimedia Presentation

- Each pixel is usually stored as a byte (a value between 0 to 255), so a 640x480 grayscale image requires 300 kB of storage ( $640 \times 480 = 307200$ ).
- We show the Lena image again, but this time in grayscale.



# Dithering

- When an image is printed, the basic strategy of **dithering** is used, which trades intensity resolution for spatial resolution to provide ability to print multi-level images on 2-level (1-bit) printers.
- **Dithering** calculates patterns of dots such that values from 0 to 255 correspond to patterns that are more and more filled at darker pixel values, for printing on a 1-bit printer.
- The main strategy is to replace a pixel value by a larger pattern, say 22 or 44, such that the number of printed dots approximates the varying-sized disks of ink used in analog, in **halftone printing** (e.g., for newspaper photos).
  - Half-tone printing is an analog process that uses smaller or larger filled circles of black ink to represent shading, for newspaper printing.

# Dithering

- For example, if we use a 2x2 **dither matrix**  $\begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$

we can first re-map image values in 0..255 into the new range 0..4 by (integer) dividing by  $256/5=51.2$ . Then, if the pixel value is 0 we print nothing, in a 2x2 area of printer output. But if the pixel value is 4 we print all four dots.

- The rule is:
  - If the intensity is  $>$  the dither matrix entry then print an *on* dot at that entry location: replace each pixel by an  $n \times n$  matrix of dots.
- Note that the image size may be much larger, for a dithered image, since replacing each pixel by a 4x4 array of dots, makes an image 16 times as large.

# Dithering

- A clever trick can get around this problem. Suppose we use a larger, 4x4 dither matrix, such as

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

- An **ordered dither** consists of turning on the printer output bit for a pixel if the intensity level is greater than the particular matrix element just at that pixel position





# Dithering

- A grayscale image of “Lena” and an ordered dither version, with a detail of Lena's right eye.



# Image Data Types

- The most common data types for graphics and image file formats - 24-bit color and 8-bit color.
- Some formats are restricted to particular hardware/operating system platforms, while others are “cross-platform” formats.
- Even if some formats are not cross-platform, there are conversion applications that will recognize and translate formats from one system to another.
- Most image formats incorporate some variation of a **compression** technique due to the large storage size of image files. Compression techniques can be classified into either **lossless** or **lossy**.

# 24-bit Color Images

- In a color 24-bit image, each pixel is represented by three bytes, usually representing RGB.
  - This format supports  $256 \times 256 \times 256$  possible combined colors, or a total of 16,777,216 possible colors.
  - However such flexibility does result in a storage penalty: A 640x480 24-bit color image would require 921.6 kB of storage without any compression.
- **An important point:** many 24-bit color images are actually stored as 32-bit images, with the extra byte of data for each pixel used to store an *alpha* value representing special effect information (e.g., transparency).

# 24-bit Color Images

- The image **forestfire.bmp.**, a 24-bit image in Microsoft Windows BMP format. Also shown are the grayscale images for the Red, Green, and Blue channels, for this image.

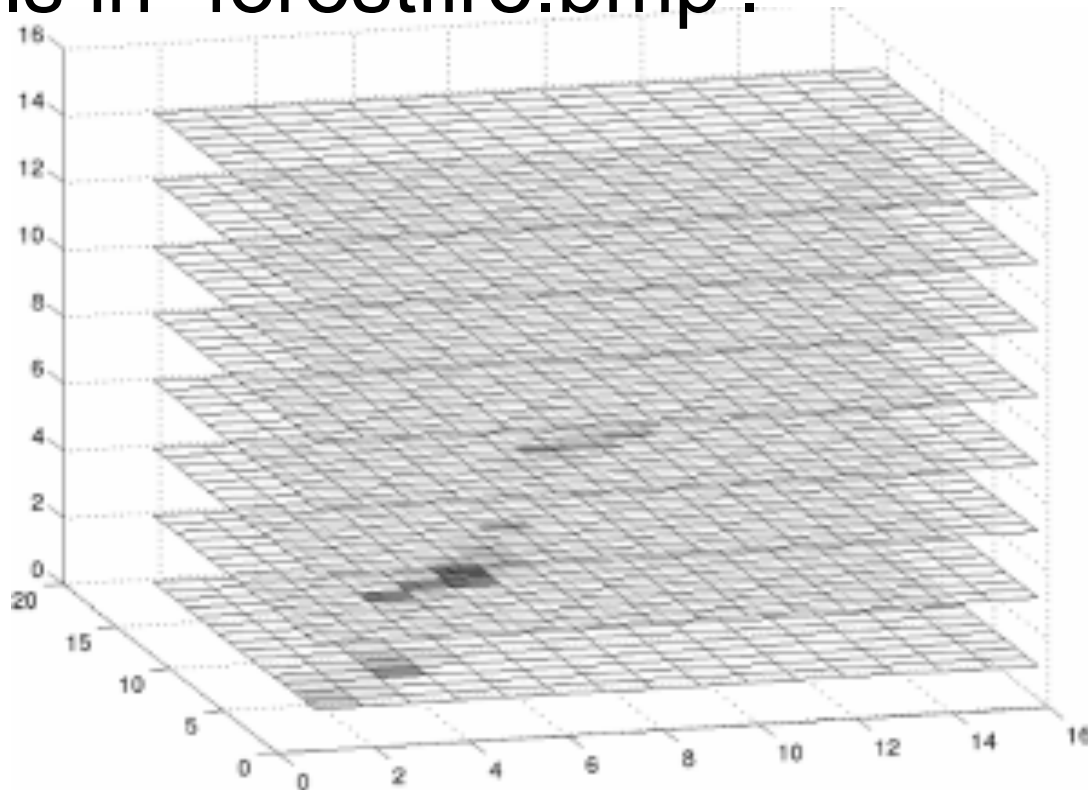


# 8-bit Color Images

- Many systems can make use of 8 bits of color information (the so-called “256 colors”) in producing a screen image.
- Such images use the concept of a **lookup table** to store color information.
  - Basically, the image stores not color, but instead just a set of single bytes, each of which is actually an index into a table of 3-byte values that specify the pixel color for that lookup table index.

# 8-bit Color Images

- A 3D histogram of the RGB values of the pixels in “forestfire.bmp”.



# 8-bit Color Images

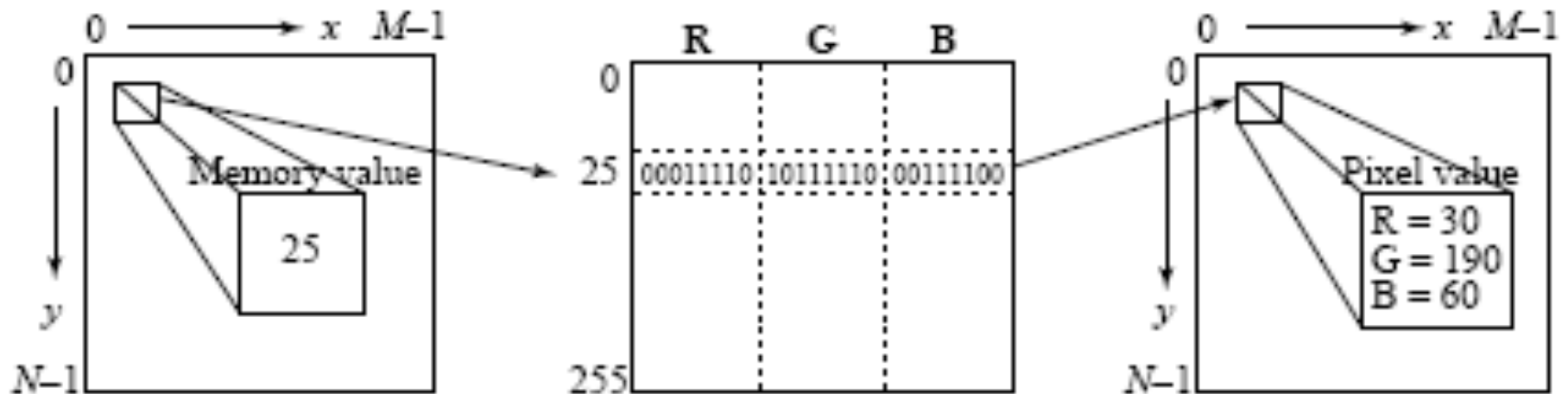
- The resulting 8-bit image, in GIF format for “forestfire.gif” .





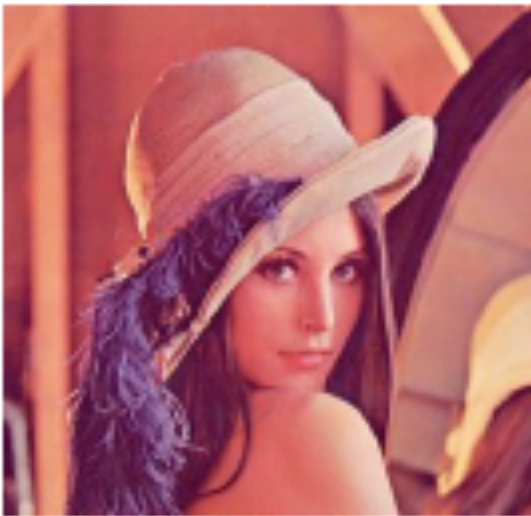
# Color Look-up Tables (LUTs)

- The idea used in 8-bit color images is to store only the index, or code value, for each pixel. Then, e.g., if a pixel stores the value 25, the meaning is to go to row 25 in a color look-up table (LUT).



# Color Look-up Tables (LUTs)

- A 24-bit color image of “Lena”, and the same image reduced to only 5 bits via color lookup and dithering. A detail of the left eye is shown.



# How to devise a color look-up table

- The most straightforward way to make 8-bit look-up color out of 24-bit color would be to divide the RGB cube into equal slices in each dimension.
  - (a) The centers of each of the resulting cubes would serve as the entries in the color LUT, while simply scaling the RGB ranges 0..255 into the appropriate ranges would generate the 8-bit codes.
  - (b) Since humans are more sensitive to R and G than to B, we could shrink the R range and G range 0..255 into the 3-bit range 0..7 and shrink the B range down to the 2-bit range 0..3, thus making up a total of 8 bits.
  - (c) To shrink R and G, we could simply divide the R or G byte value by  $(256/8)=32$  and then truncate. Then each pixel in the image gets replaced by its 8-bit index and the color LUT serves to generate 24-bit color.

# How to devise a color look-up table

- **Median-cut algorithm:** A simple alternate solution that does a better job for this color reduction problem.
  - (a) The idea is to sort the R byte values and find their median; then values smaller than the median are labeled with a “0” bit and values larger than the median are labeled with a “1” bit.
  - (b) This type of scheme will indeed concentrate bits where they most need to differentiate between high populations of close colors.
  - (c) One can most easily visualize finding the median by using a histogram showing counts at position 0..255.

# How to devise a color look-up table

- Histogram of R bytes for the 24-bit color image “forestfire.bmp” results in a “0” bit or “1” bit label for every pixel.
- For the second bit of the color table index being built, we take R values less than the R median and label just those pixels as “0” or “1” according as their G value is less than or greater than the median of the G value, just for the “0” Red bit pixels.
- Continuing over R, G, B results in eight possible combinations, or 3 bits, and gives a color LUT with 8 rows.
- In practice, divide the histogram of red bytes into eight regions with equal numbers of pixels in each region, the green histograms into eight regions and the blue histograms into four regions.

# Popular File Formats

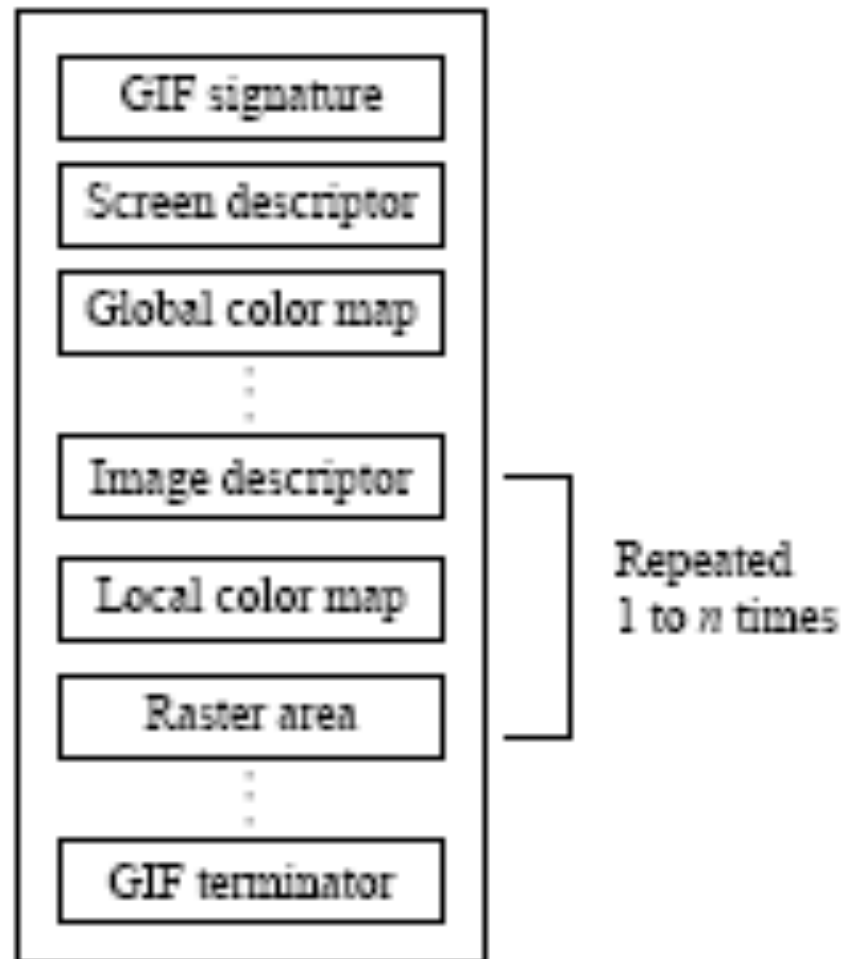
- **8-bit GIF** : one of the most important format because of its historical connection to the WWW and HTML markup language as the first image type recognized by net browsers.
- **JPEG**: currently the most important common file format.

# GIF

- **GIF standard:** (We examine GIF standard because it is so simple yet contains many common elements.)
- Limited to 8-bit (256) color images only, which, while producing acceptable color images, is best suited for images with few distinctive colors (e.g., graphics or drawing).
- GIF standard supports **interlacing** - successive display of pixels in widely-spaced rows by a 4-pass display process.
- GIF actually comes in two flavors:
  1. **GIF87a:** The original specification.
  2. **GIF89a:** The later version. Supports simple animation via a Graphics Control Extension block in the data, provides simple control over delay time, a transparency index, etc.

# GIF87

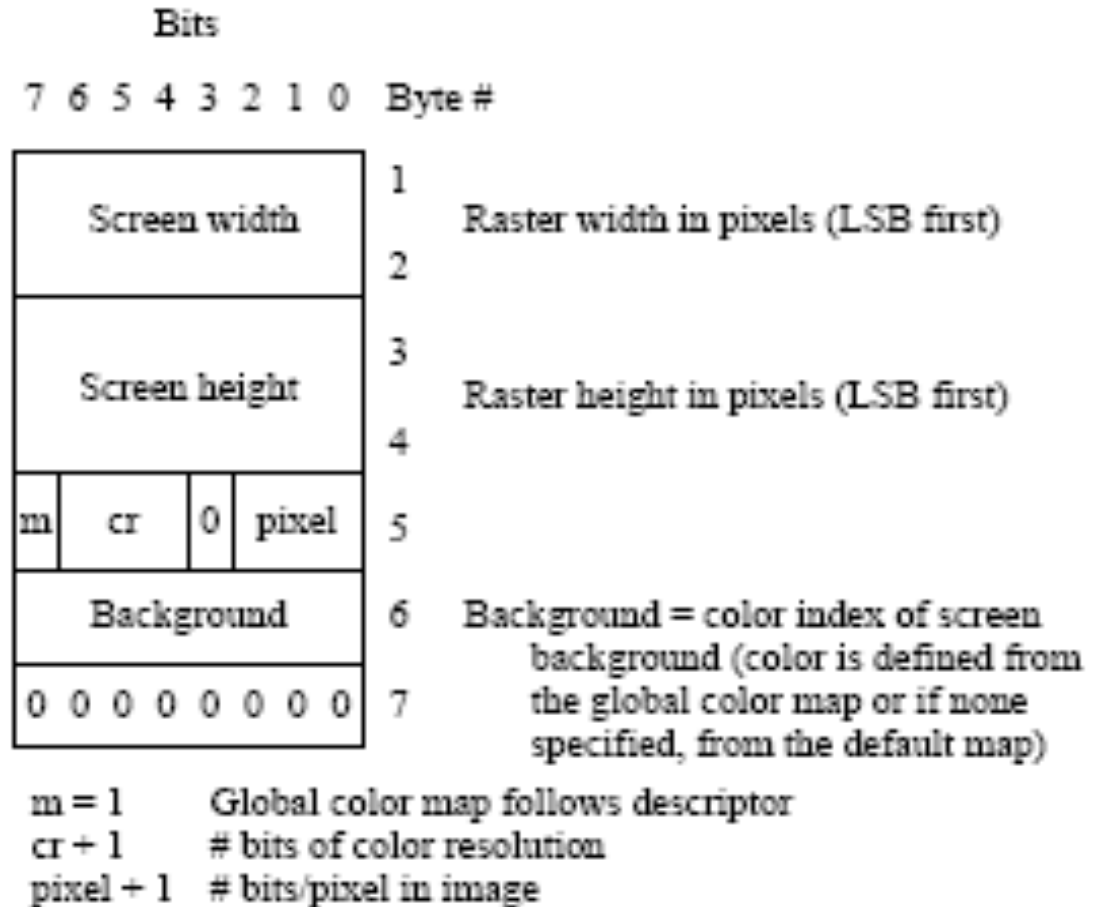
- For the standard specification, the general file format of a GIF87 file is





# GIF87

- **Screen Descriptor** comprises a set of attributes that belong to every image in the file.



# GIF87

- **Color Map** is set up in a very simple fashion. The actual length of the table equals  $2^{\text{pixel}+1}$  as given in the Screen Descriptor.

| Bits            |               | Byte # |                                  |
|-----------------|---------------|--------|----------------------------------|
| 7               | 6 5 4 3 2 1 0 |        |                                  |
| Red intensity   |               | 1      | Red value for color index 0      |
| Green intensity |               | 2      | Green value for color index 0    |
| Blue intensity  |               | 3      | Blue value for color index 0     |
| Red intensity   |               | 4      | Red value for color index 1      |
| Green intensity |               | 5      | Green value for color index 1    |
| Blue intensity  |               | 6      | Blue value for color index 1     |
| ⋮               |               |        | (continues for remaining colors) |

# GIF87

- Each image in the file has its own Image Descriptor

| Bits         |   |   |   |   |       | Byte # |   |    |   |
|--------------|---|---|---|---|-------|--------|---|----|---|
| 7            | 6 | 5 | 4 | 3 | 2     | 1      | 0 |    |   |
| 0            | 0 | 1 | 0 | 1 | 1     | 0      | 0 | 1  | Image separator character (comma)   |
| Image left   |   |   |   |   |       |        |   | 2  | Start of image in pixels from the left side of the screen (LSB first)   |
| Image top    |   |   |   |   |       |        |   | 3  | Start of image in pixels from the top of the screen (LSB first)   |
| Image width  |   |   |   |   |       |        |   | 4  | Width of the image in pixels (LSB first)  |
| Image height |   |   |   |   |       |        |   | 5  | Height of the image in pixels (LSB first)   |
| m            | i | 0 | 0 | 0 | pixel |        |   | 6  |   |
|              |   |   |   |   |       |        |   | 7  |   |
|              |   |   |   |   |       |        |   | 8  |   |
|              |   |   |   |   |       |        |   | 9  |   |
|              |   |   |   |   |       |        |   | 10 | m = 0    Use global color map, ignore 'pixel'<br>m = 1    Local color map follows, use 'pixel'<br>i = 0    Image formatted in Sequential order<br>i = 1    Image formatted in Interlaced order<br>pixel + 1   # bits per pixel for this image |

# GIF87

- If the “interlace” bit is set in the local Image Descriptor, then the rows of the image are displayed in a four-pass sequence

| Image row | Pass 1 | Pass 2 | Pass 3 | Pass 4 | Result |
|-----------|--------|--------|--------|--------|--------|
| 0         | *1a*   |        |        |        | *1a*   |
| 1         |        |        |        | *4a*   | *4a*   |
| 2         |        |        | *3a*   |        | *3a*   |
| 3         |        |        |        | *4b*   | *4b*   |
| 4         |        | *2a*   |        |        | *2a*   |
| 5         |        |        |        | *4c*   | *4c*   |
| 6         |        |        | *3b*   |        | *3b*   |
| 7         |        |        |        | *4d*   | *4d*   |
| 8         | *1b*   |        |        |        | *1b*   |
| 9         |        |        |        | *4e*   | *4e*   |
| 10        |        |        | *3c*   |        | *3c*   |
| 11        |        |        |        | *4f*   | *4f*   |
| 12        |        | *2b*   |        |        | *2b*   |
| ⋮         |        |        |        |        |        |

# JPEG

- **JPEG**: The most important current standard for image compression.
- The human vision system has some specific limitations and JPEG takes advantage of these to achieve high rates of compression.
- JPEG allows the user to set a desired level of quality, or compression ratio (input divided by output).

# JPEG

- As an example, we show our **forestfire.** image, with a quality factor  $Q=10\%$ .
  - This image is a mere 1.5% of the original size. In comparison, a JPEG image with  $Q=75\%$  yields an image size 5.6% of the original, whereas a GIF version of this image compresses down to 23.0% of uncompressed image size.



# 8-bit Color Images

- An 8-bit image, in GIF format for “forestfire.gif”.



# PNG

- **PNG format:** standing for **Portable Network Graphics** - meant to supersede the GIF standard, and extend it in important ways.
- Special features of PNG files include:
  1. Support for up to 48 bits of color information - a large increase.
  2. Files may contain gamma-correction information for correct display of color images, as well as alpha-channel information for such uses as control of transparency.
  3. The display progressively displays pixels in a 2-dimensional fashion by showing a few pixels at a time over seven passes through each 8x8 block of an image.



# TIFF

- **TIFF**: stands for **Tagged Image File Format**.
- The support for attachment of additional information (referred to as “tags”) provides a great deal of flexibility.
  1. The most important tag is a format designifier: what type of compression etc. is in use in the stored image.
  2. TIFF can store many different types of image: 1-bit, grayscale, 8-bit color, 24-bit RGB, etc.
  3. TIFF was originally a lossless format but now a new JPEG tag allows one to opt for JPEG compression.
  4. The TIFF format was developed by the Aldus Corporation in the 1980's and was later supported by Microsoft.

# EXIF

- **EXIF** (Exchange Image File) is an image format for digital cameras:
  1. Compressed EXIF files use the baseline JPEG format.
  2. A variety of tags (many more than in TIFF) are available to facilitate higher quality printing, since information about the camera and picture-taking conditions (flash, exposure, light source, white balance, type of scene, etc.) can be stored and used by printers for possible color correction algorithms.
  3. The EXIF standard also includes specification of the format for audio that accompanies digital images. As well, it also supports tags for information needed for conversion to FlashPix (initially developed by Kodak).

# Graphics Animation Files

- A few dominant formats aimed at storing graphics animations (i.e., series of drawings or graphic illustrations) as opposed to video (i.e., series of images).
- **Difference:** cartoon-like animations are considerably less demanding of resources than video files.
  1. FLC is an animation or moving picture file format; it was originally created by Animation Pro. Another format, FLI, is similar to FLC.
  2. GL produces somewhat better quality moving pictures. GL animations can also usually handle larger file sizes.
  3. Many older formats: such as DL or Amiga IFF files, Apple Quicktime files, as well as animated GIF89 files.

# PS and PDF

- **Postscript** is an important language for typesetting, and many high-end printers have a Postscript interpreter built into them.
- Postscript is a vector-based picture language, rather than pixel-based: page element definitions are essentially in terms of vectors.
  1. Postscript includes text as well as vector/structured graphics.
  2. GL bit-mapped images can be included in output files.
  3. Encapsulated Postscript files add some additional information for inclusion of Postscript files in another document.

# PS and PDF

4. Postscript page description language itself does not provide compression; in fact, Postscript files are just stored as ASCII.
- Another text + figures language has begun to supersede or at least parallel Postscript: Adobe Systems Inc. includes LZW compression in its Portable Document Format (**PDF**) file format.
    - PDF files that do not include images have about the same compression ratio, 2:1 or 3:1, as do files compressed with other LZW-based compression tools.

# Some Other Image Formats

- **Microsoft Windows: WMF:** the native vector file format for the Microsoft Windows operating environment:
  1. Consist of a collection of GDI (Graphics Device Interface) function calls, also native to the Windows environment.
  2. When a WMF file is “played” (typically using the Windows PlayMetaFile() function) the described graphics is rendered.
  3. WMF files are ostensibly device-independent and are unlimited in size.

# Some Other Image Formats

- **Microsoft Windows: BMP:** the major system standard graphics file format for Microsoft Windows, used in Microsoft Paint and other programs. Many sub-variants within the BMP standard.
- **Macintosh: PAINT and PICT:**
  1. PAINT was originally used in the MacPaint program, initially only for 1-bit monochrome images.
  2. PICT format is used in MacDraw (a vector-based drawing program) for storing structured graphics.
- **X-windows: PPM:** the graphics format for the X Window system. PPM supports 24-bit color bitmaps, and can be manipulated using many public domain graphic editors, e.g., xv.