Fundamentals of Multimedia

3rd Edition + supplementary material

Chapter 3 :

Graphics and Image Data Representations

- This chapter introduces:
  - how best to represent the graphics and image data since it is of crucial importance in the study of multimedia.
  - The specifics of file formats for storing such images are also discussed

3.1 Graphics/Image Data Types

- Table 3.1 shows a list of file formats used in the popular product Adobe Premiere.
- We concentrate on GIF and JPG image file formats, since the **GIF** file format is one of the simplest and contains several fundamental features,
- and the **JPG** file format is arguably the most important overall.

Table 3.1 Some popular Adobe Premiere file formats

Image	Audio	Video
BMP, DIB,	AIFF, AAC,	AVI, DV,
GIF, HEIF,	AC3, BWF,	FLV, HEVC,
JPG, PICT,	MP3, M4A,	M4V, MOV, MP4,
PNG, PSD,	WAV, WMA	MPG, MTS, MXF,
TGA, TIF		SWF, WMV

3.1.1 1-Bit Images

- Images consist of pixels (picture elements in digital images).
- A 1-bit image (also called binary image) consists of on and off bits only and thus is the simplest type of image.
- Each pixel is stored as a single bit (0 or 1)
- It is also sometimes called a *1-bit monochrome (called Lena image by scientists)* image since it contains no color. See Figure in next slide.
- Monochrome
- 1-bit images can be satisfactory for pictures containing only simple graphics and text.
- fax machines use 1-bit data, so in fact 1-bit images are still important.

Monochrome 1-bit Lena image

A 640×480 monochrome image requires 38.4 kB of storage

3.1.2 8-Bit Gray-Level Images

- 8-bit image is one for which each pixel has a *gray value* between 0 and 255.
- Each pixel is represented by a single byte.
- The entire image can be thought of as a two-dimensional array of pixel values referred to as a *bitmap*.
- *Image resolution* refers to the number of pixels in a digital image (higher resolution always yields better quality but increases size)

Grayscale image of Lena

640×480 grayscale image requires 300kB of storage

3.1.4 24-Bit Color Images

- In a color 24-bit image, each pixel is represented by three bytes, usually representing RGB.

- Since each value is in the range 0–255, this format supports 256×256×256, or a total of 16,777,216, possible combined colors; which increases storage size.

- a 640 × 480 24-bit color image would require 921.6 kB of storage. (without any compression applied)
- Compression is used to decrease the image size by simply grouping pixels effectively. (chapter 7).

24-bit color image forestfire.bmp

Microsoft Windows BMP format

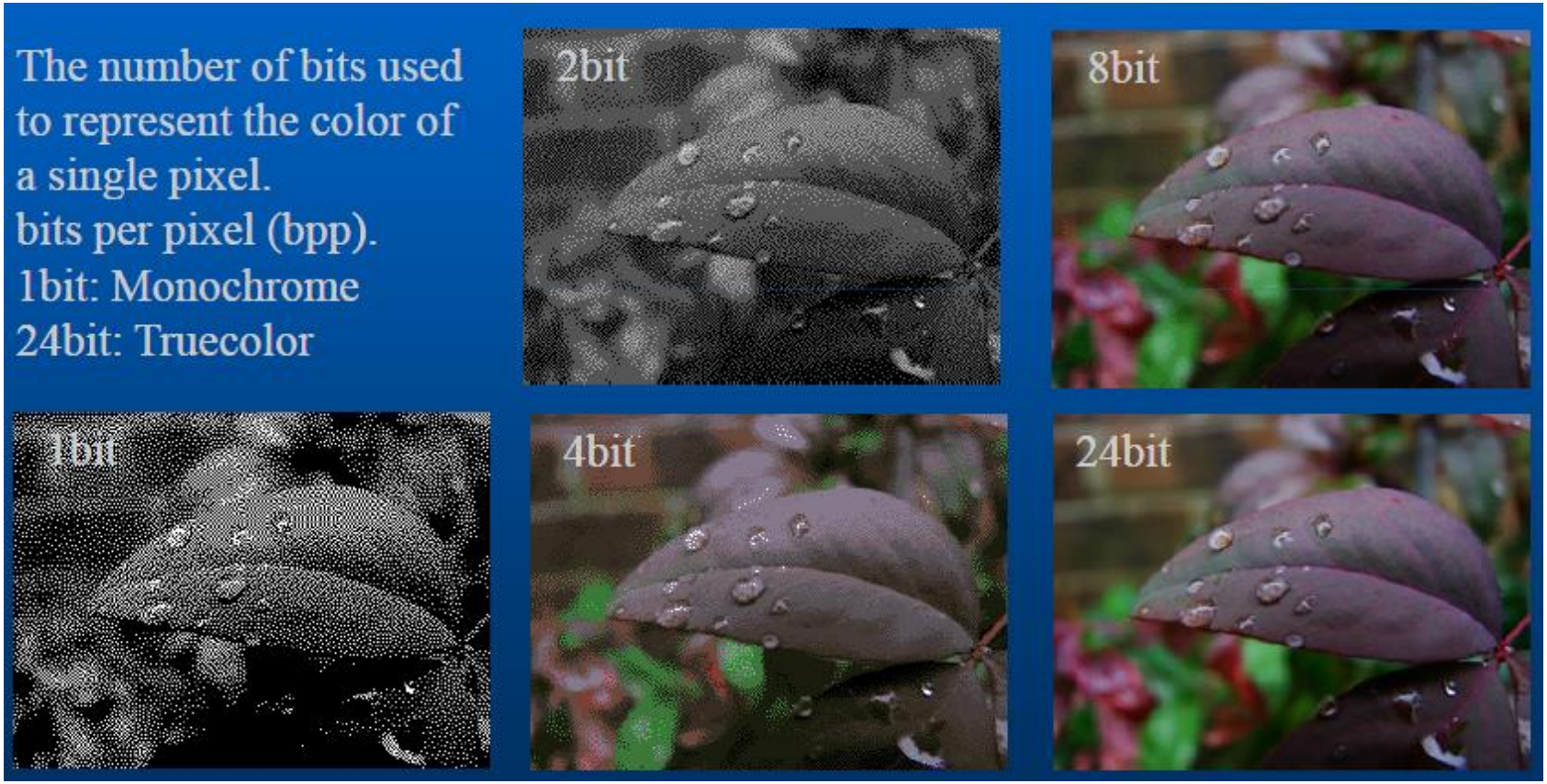
3.1.5 Higher Bit-Depth Images

- In some fields such as medicine (security cameras, satellite imaging) more accurate images are required to see the patient's liver, for example.

- To get such images, special cameras that view more than just 3 colors (RGB) are used.

- Such images are called *multispectral* (more than three colors) or *hyperspectral* (224 colors for satellite imaging).

تاثیر عمق رنگ در کیفیت تصویر



عمق رنگ یا رزولوشن رنگ به تعداد بیت به کار رفته برای نمایش اطلاعات رنگ هر پیکسل گفته می‌شود.

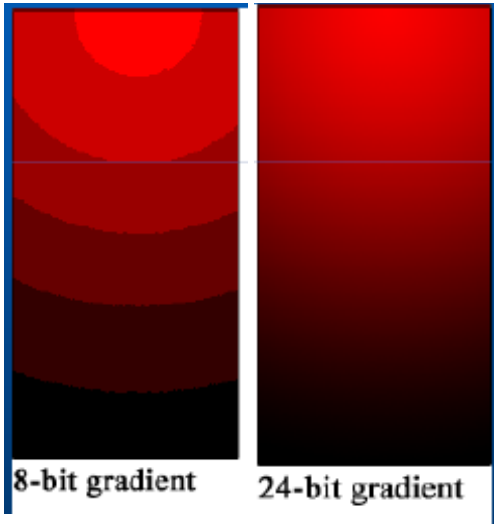
3.1.6 Quantizing Color Information

- Reasonably accurate color images can be obtained by *quantizing* the color information.
- Color quantizing example: reducing the number of colors required to represent a digital image makes it possible to reduce its file size.
- Example: decreasing the number of bits per pixel from 24 to 8 (decreasing number of colors from 16million to 256)

- Great savings in space for 8-bit images over 24-bit ones:
  - a 640 × 480 8-bit color image requires only 300 kB of storage,
  - compared to 921.6 kB for a color image (without any other compression applied).

3.1.6 Quantizing Color Information

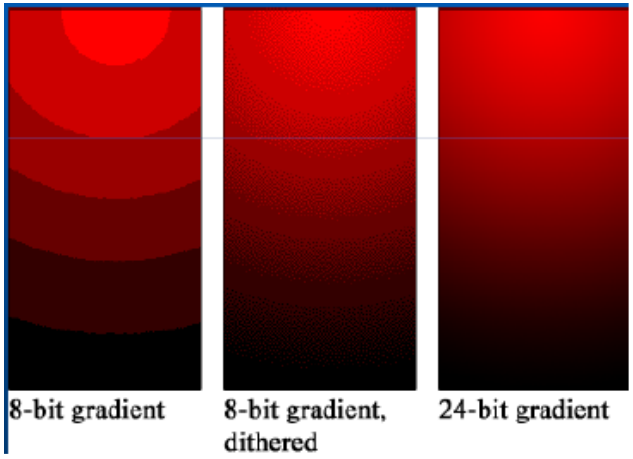
- First approach: straightforward quantization→ decrease the number of bits for each color component:
- From 24 bits per pixel (8 bits for each RGB color components per pixel) to 8 bits per pixel (3 bits for R and G and 2 bits for B component)
- Drawback: "banding" effect



3.1.6 Quantizing Color Information

- Two basic techniques are proposed to reduce the banding effect:
  - a) Quantization using Look Up Table (LUT)
  - b) Dithering

Using LUT



Using Dithering

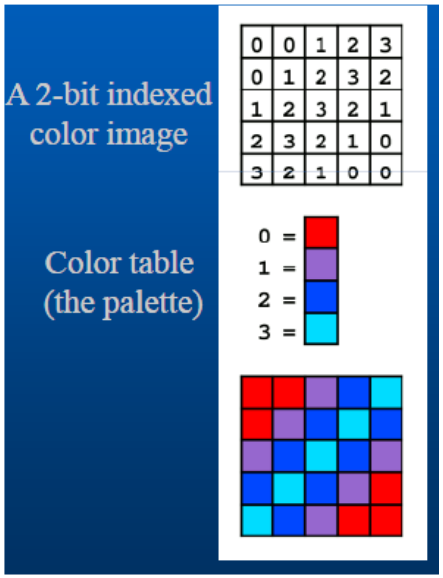
3.1.6 Quantizing Color using LUT

- Use the concept of a *lookup table (LUT)* to store color information:
- Example: we choose 256 most representative colors and store them in an LUT

  → each color can be coded with 8 bits

  → hence it is called 8-bit color image

  → this is called Vector Quantization



Notice that the difference between Fig. 3.5a, the 24-bit image, and Fig. 3.7, the 8-bit image, is reasonably small.

Fig. 3.5a, the 24-bit image

Fig. 3.7, the 8-bit image

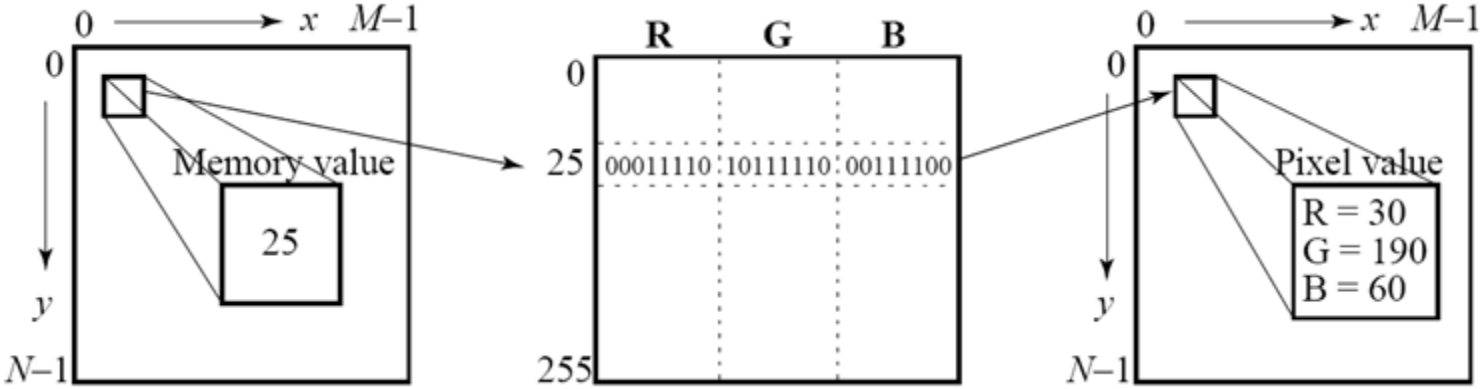
Another example for difference between Fig. 3.5a, the 24-bit image, and Fig. 3.7, the 8-bit image, is reasonably small.

Fig. 3.5a, the 24-bit image

Fig. 3.7, the 8-bit image

3.1.7 Color Lookup Tables

- The LUT is often called a *palette*.
- The idea is to store only the index, or code value, for each pixel.
- if a pixel stores, say, the value 25 (Figure 3.8), the meaning is to go to row 25 in a color lookup table (LUT).



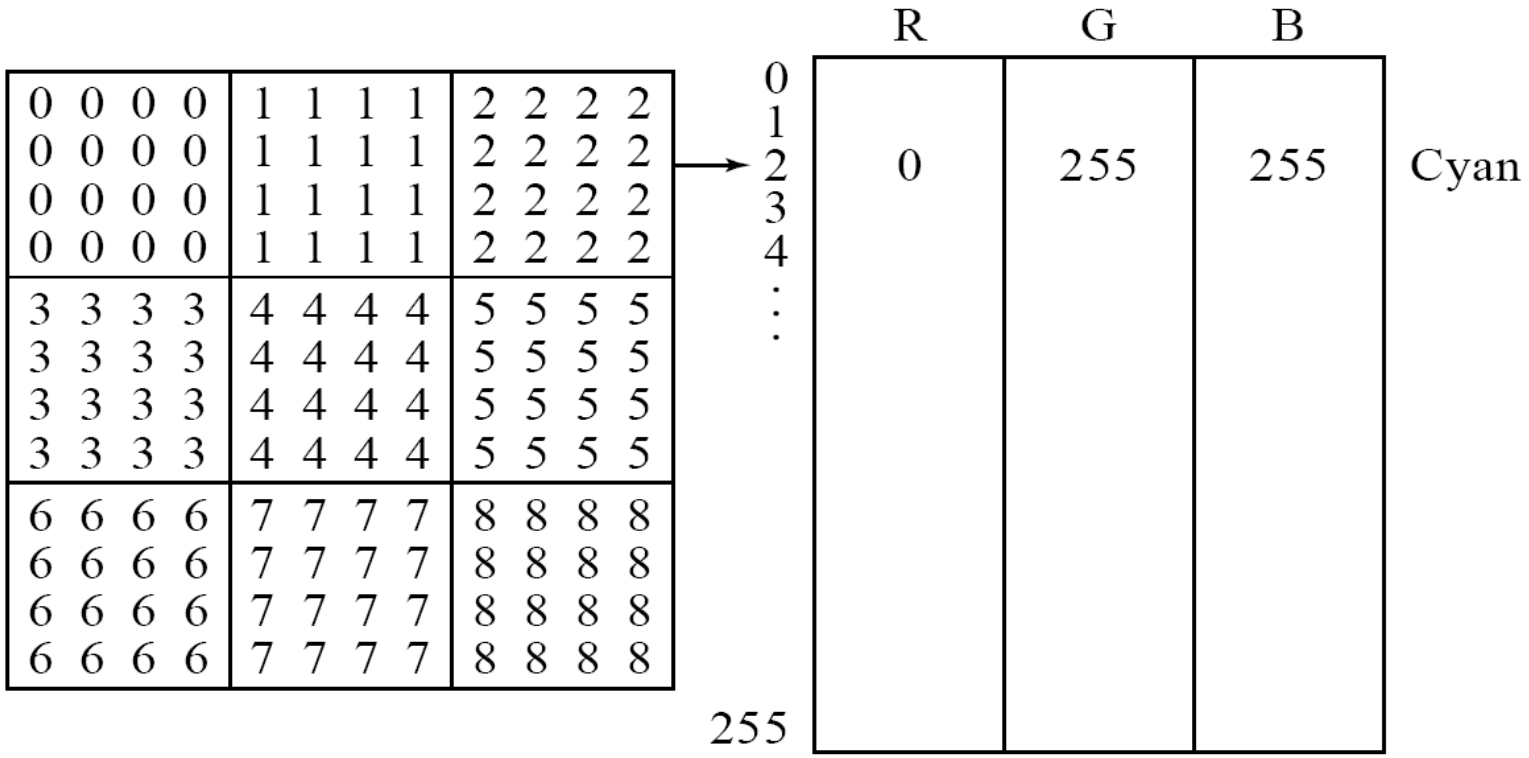


Fig. 3.9: Color-picker for 8-bit color: each block of the color-picker corresponds to one row of the color LUT

→ if the user selects the color block with index value 2, then the color meant is cyan, with RGB values (0, 255, 255).

3.1.6 LUT نحوه تشکیل جدول

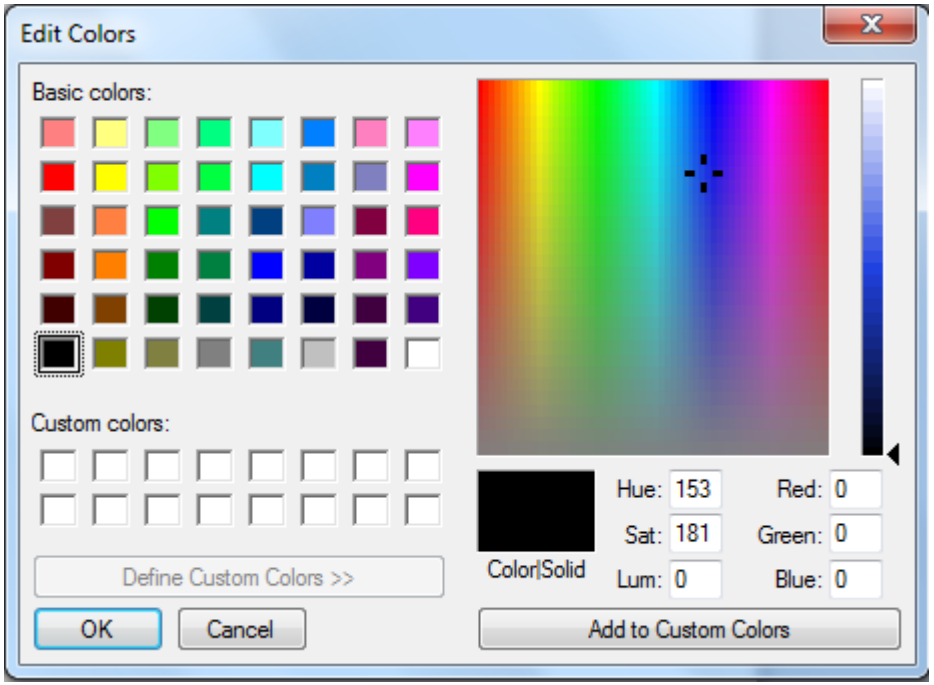
سه روش پایه:

- انتخاب دستی از طریق یک واسط Color picker
- استفاده از روش خوشه‌بندی Clustering
- استفاده از روش Median Cut

روش دستی انتخاب مجموعه رنگ از طریق پلت رنگ

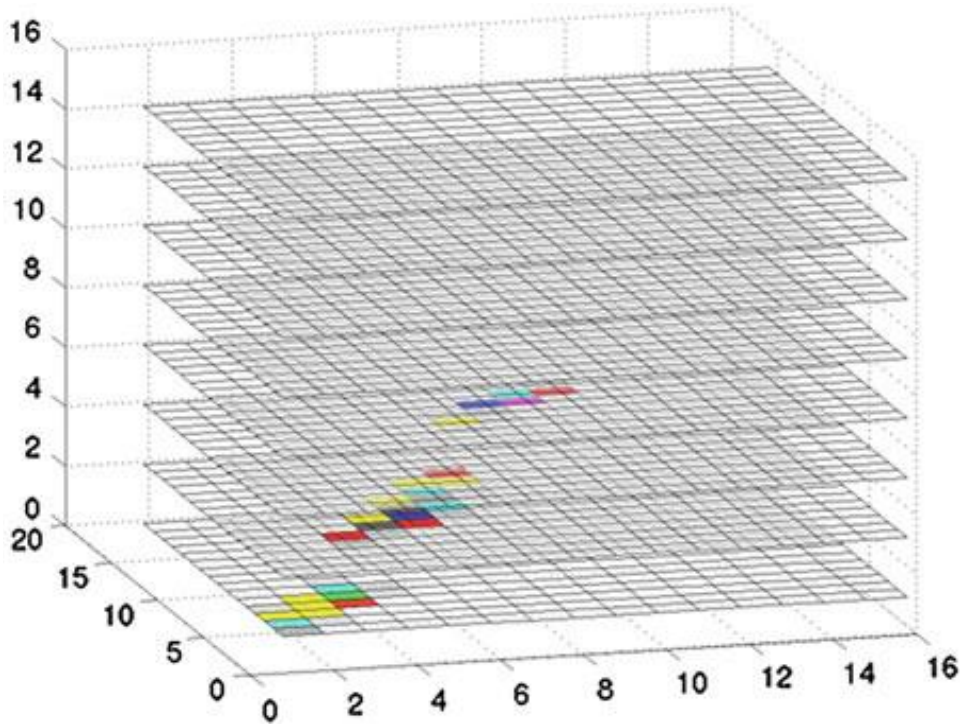
•A **Color-picker** consists of an array of fairly large blocks of color (or a semi-continuous range of colors) such that a mouse-click will select the color indicated.

- In reality, a color-picker displays the palette colors associated with index values from 0 to 255.



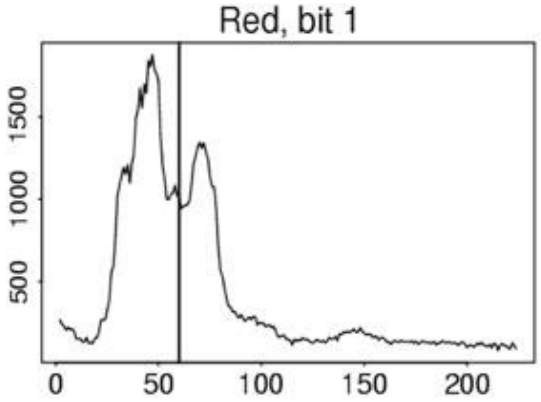
روش خوشه‌بندی انتخاب مجموعه رنگ

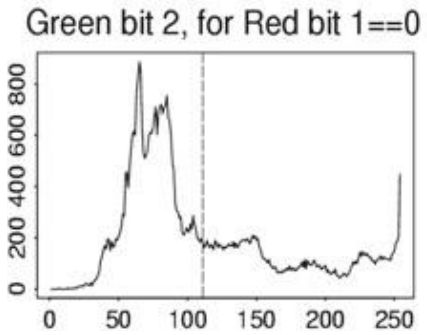
تمرکز رنگ یک تصویر در فضای سه بعدی RGB نشان داده شده است. ملاحظه می شود قسمت عمده ای از فضای رنگ استفاده نشده است. از بین رنگ های استفاده شده و در نقاط تمرکز رنگ های استفاده شده، مجموعه ای از رنگ انتخاب می شود.

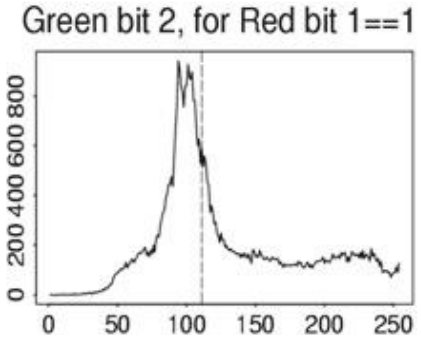


روش Median-Cut برای انتخاب مجموعه رنگ

نقاط میانه (Median) در هیستوگرام رنگ (Color Histogram) تصویر می‌تواند برای انتخاب مجموعه رنگ لازم برای نمایش نزدیک به واقعیت یک تصویر استفاده شود. بطور پی در پی (successive)، از میانه برای تقسیم فضای رنگ به ۲ و ۴و ۸ و ... قسمت استفاده می شود. پس از k بار، فضای رنگ به 2^k قسمت تقسیم می شود و برای هر قسمت یک رنگ بعنوان نماینده ان ناحیه انتخاب می شود. در قسمت‌هایی که تمرکز رنگ بیشتری هست، تعداد رنگ بیشتری انتخاب می شود.

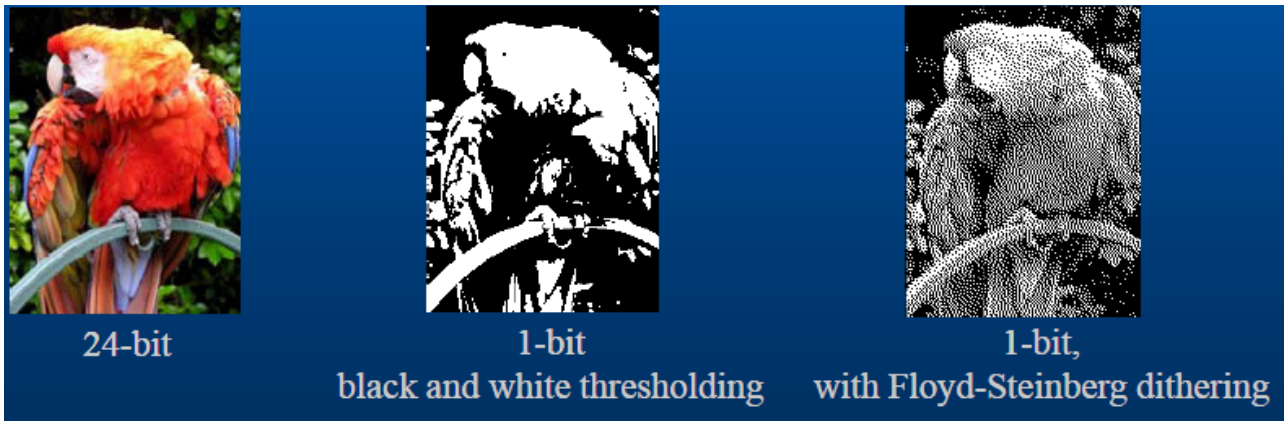






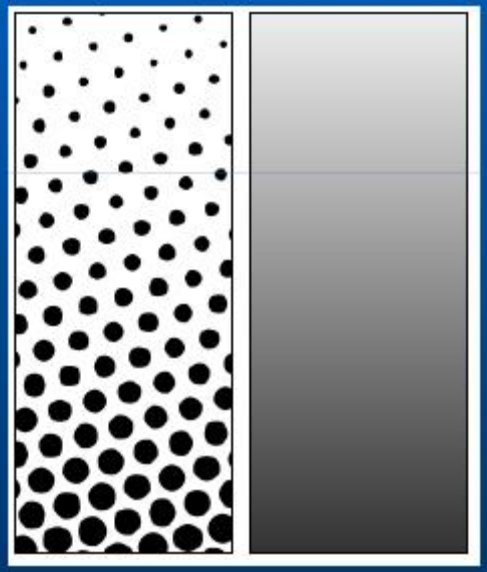
Dithering

- دیترینگ روشی برای کاهش اثر banding در شرایط کاهش رزولوشن رنگ است (برای کاهش حجم فایل تصویر یا بدلیل محدودیت رسانه نمایش تصویر)
- دیترینگ از تکنیک Halftone برای افزایش سطوح رنگ (عمق رنگ) از نظر بصری استفاده می‌کند: مثلا ایجاد ۱۶ سطح خاکستری با استفاده از دو رنگ سیاه و سفید (ایجاد سطوح خاکستری با توزیع نقاط سیاه و سفید با چگالی مختلف)
- دیترینگ جزییات اطلاعات مکانی تصویر را کاهش می دهد تا اثر کاهش رزولوشن رنگ را جبران کند.



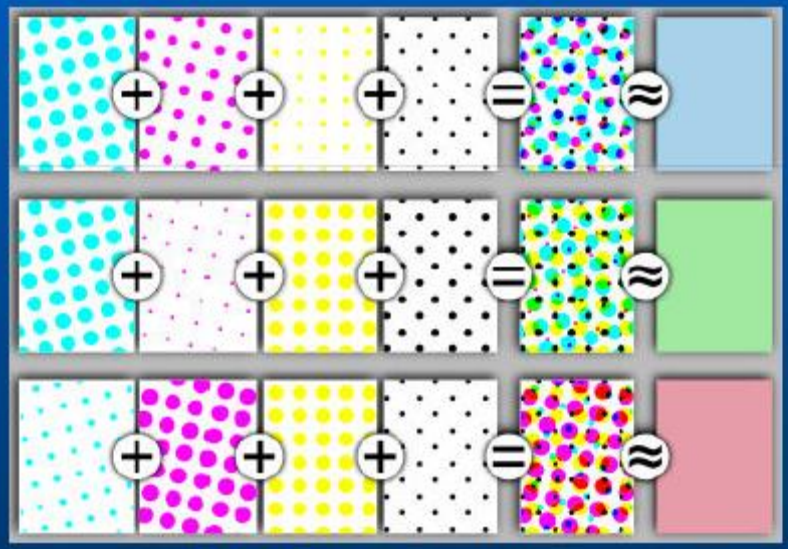
تکنیک Halftone مورد استفاده در دیترینگ

A technique that simulates continuous tone imagery through the use of dots, varying either in size, in shape or in spacing.



Halftone dots

How the human eye would see this sort of arrangement from a sufficient distance.



Color Halftoning

Three examples of color halftoning with CMYK separations. From left to right: The cyan separation, the magenta separation, the yellow separation, the black separation, the combined halftone pattern and finally how the human eye would observe the combined halftone pattern from a sufficient distance.

روش پایه دیترینگ بر پایه تکنیک Halftone

- یک ماتریس دیترینگ nxn در نظر می گیریم.
- درایه های این ماتریس مقادیر 0 تا n^2-1 را می گیرند:

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

- در هر یک از نقاط این ماتریس می‌تواند یک نقطه سیاه باشد یا نباشد. پس از نظر تعداد نقاط سیاه موجود n^2+1 حالت می تواند وجود داشته باشد.
- اگر به فرض از ماتریس 2x2 استفاده کنیم، ۵ حالت قابل تعریف است. پس محدوده مقادیر ممکن رنگ را به ۵ ناحیه تقسیم می کنیم و برای هر کدام یکی حالت‌های ماتریس دیترینگ چاپ میشود. بطور مثال اگر مقادیر دامنه رنگ هر پیکسل ۴ بیتی (m بیت) باشد، ۱۶ حالت ممکن داریم (مقادیر 0 تا 15). اندازه هر ناحیه برابر ۱۶÷۵=۳.۲=L . اگر مقدار هر پیکسل (y) را به L تقسیم کنیم، مقدار خارج قسمت (z) عددی بین 0 تا 4 خواهد بود (۵ حالت).
- در هر یک از نقاط ماتریس دیترینگ (i,j) اگر $z \le D(i,j)$ یک نقطه سیاه در آن چاپ می‌شود.

روش دیترینگ بدون افزایش تعداد پیکسل‌ها

- عیب روش قبل این است که تعداد نقاط پیکسل n^2 برابر خواهد شد.
- می‌توان از این مسئله جلوگیری کرد: روش Ordered Dithering
- در این روش، ماتریس دیترینگ بر روی تصویر لغزانده میشود و اگر $z \leq D(i,j)$ در آن محل یک نقطه سیاه چاپ می شود:

Algorithm 3.1 Ordered Dither

```
begin
    for x = 0 to x_max          // columns
        for y = 0 to y_max      // rows
            i = x mod n
            j = y mod n
            // I(x, y) is the input, O(x, y) is the output, D is the dither matrix.
            if I(x, y) > D(i, j)
                O(x, y) = 1;
            else
                O(x, y) = 0;
end
```

دیترینگ بر روی تصاویر رنگی

Floyd–Steinberg Algorithm

- **Distribute the quantization residual to neighboring pixels that have not yet been processed.**

$$\frac{1}{16}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

Distribution matrix

- **Pseudocode:**

```
for each y from top to bottom
   for each x from left to right
      oldpixel := pixel[x][y]
      newpixel := find_closest_palette_color(oldpixel)
      pixel[x][y] := newpixel
      quant_error := oldpixel - newpixel
      pixel[x+1][y] := pixel[x+1][y] + 7/16 * quant_error
      pixel[x-1][y+1] := pixel[x-1][y+1] + 3/16 * quant_error
      pixel[x][y+1] := pixel[x][y+1] + 5/16 * quant_error
      pixel[x+1][y+1] := pixel[x+1][y+1] + 1/16 * quant_error
```

رزولوشن تصویر

Pixels per inch (ppi)

- **The average human eye can only detect 300 ppi.**

iPhone 5
4"
1136x640
326 ppi

iPad 4
9.7"
2048x1536
264 ppi

HTC One
4.7"
1920x1080
468 ppi

Lumia 920
4.5"
1280x768
332 ppi

Galaxy Note II
5.55"
720x1280
267 ppi

Xperia Z
5"
1920x1080
443 ppi

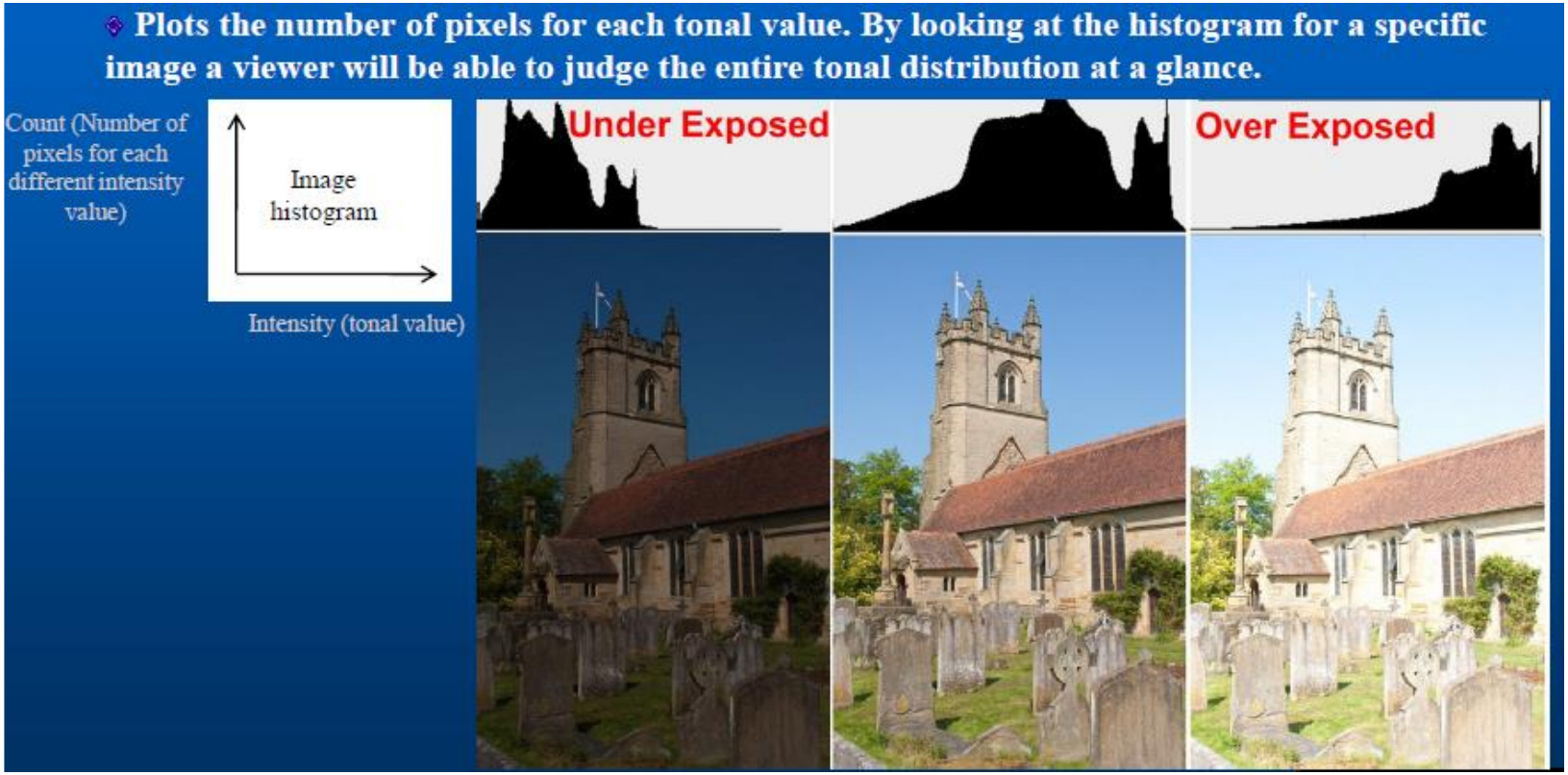
List of displays by pixel density
http://en.wikipedia.org/wiki/List_of_displays_by_pixel_density

$$d_p = \sqrt{w_p^2 + h_p^2}$$

$$PPI = \frac{d_p}{d_i}$$

- d_p is diagonal resolution in pixels,
- w_p is width resolution in pixels,
- h_p is height resolution in pixels and
- d_i is diagonal size in inches. (This is the number advertised as the size of the display.)

پردازش تصویر مبتنی بر شیفت هیستوگرام



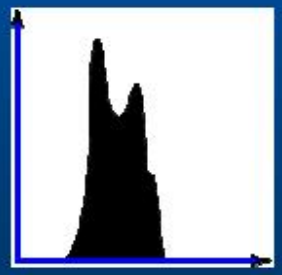
پردازش تصویر مبتنی بر شیفت هیستوگرام

Contrast

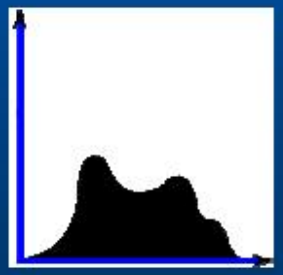
Formula

$$\frac{\text{Luminance difference}}{\text{Average luminance}}.$$

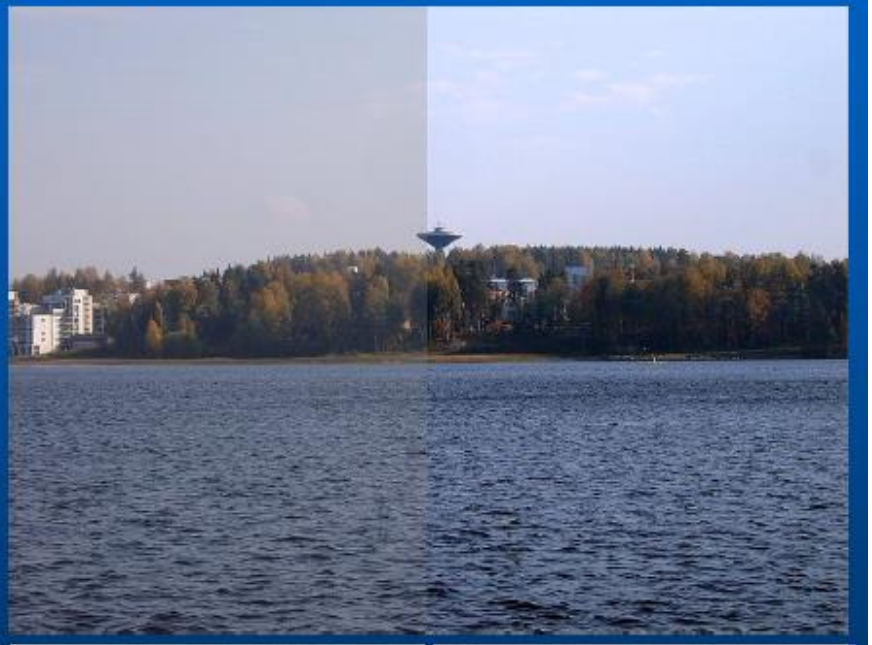
$$\frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}},$$

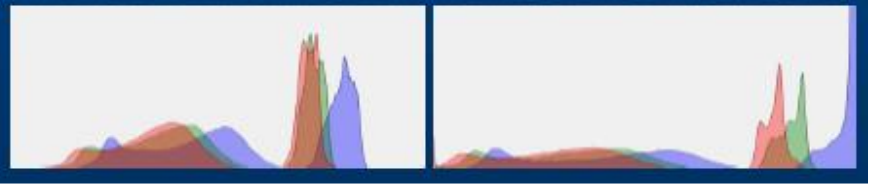


Typ. histogram of
a low contrast image

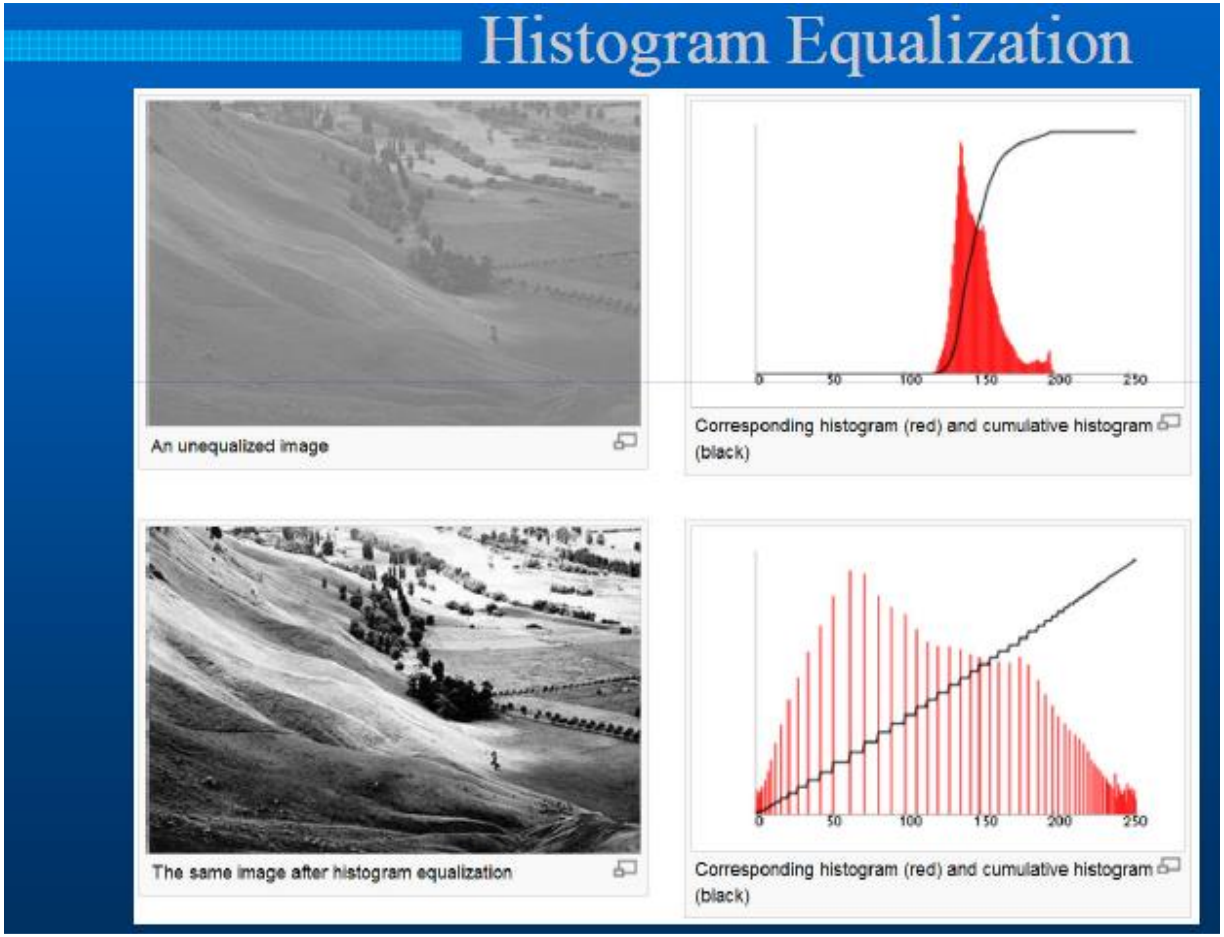


Typ. histogram of
a high contrast image





پردازش تصویر مبتنی بر شیفت هیستوگرام



3.2 Popular File Formats

- **8-bit GIF** : one of the most important formats 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.

3.2.1 GIF

- **GIF standard (Graphics Interchange Format)**: (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. (Figure 3.16, slide 25)

- **interlacing** allows a quick sketch to appear when a web browser displays the image, followed by more detailed fill-ins.

- The JPEG standard (below) has a similar display mode, denoted *progressive mode*.
- GIF has two formats GIF87 (standard) and GIF89 supports simple animation.

3.2.1 GIF

GIF87

- For the standard specification, the general file format of a GIF87 file is as in Fig. 3.12.
- The *Signature is six bytes*
- the *Screen Descriptor is a seven-byte*
- *Local Color Map (if does not exist*

A global color map can be defined)

- A GIF87 file can contain more than one image definition, usually to fit on several different parts of the screen.
- *actual raster data itself is first compressed using the LZW compression* scheme (see Chap. 7)

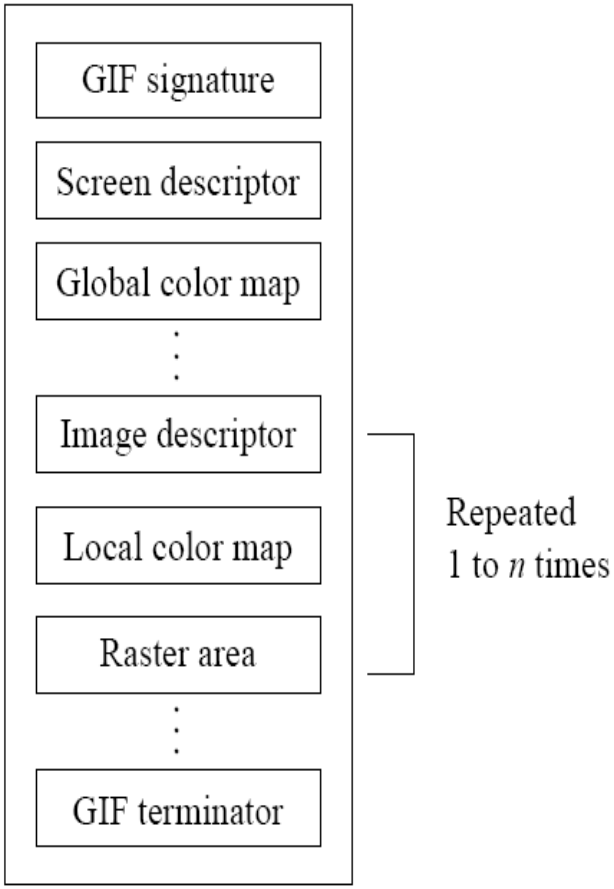


Fig. 3.12: GIF file format.

- **Screen Descriptor** comprises a set of attributes that belong to every image in the file. According to the GIF87 standard, it is defined as in Fig. 3.13.
- LSB/ MSB :

Least/Most Significant Byte

- Bit 7 is filled with zeros

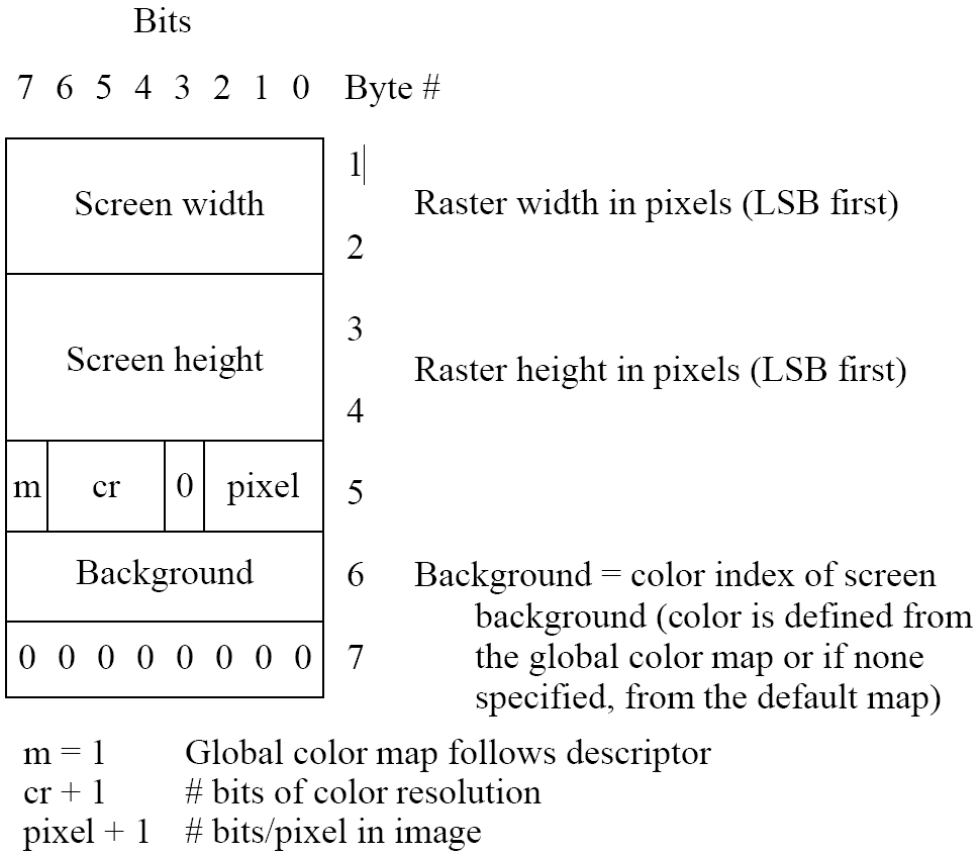


Fig. 3.13: GIF screen descriptor.

- **Color Map** is set up in a very simple fashion as in Fig. 3.14. However, the actual length of the table equals $2^{(pixel+1)}$ as given in the Screen Descriptor.

Bits 7 6 5 4 3 2 1 0	Byte #	
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)

Fig. 3.14: GIF color map.

- Each image in the file has its own **Image Descriptor**, defined as in Fig. 3.15.

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

Fig. 3.15: GIF image descriptor.

interlace

- If the *interlace* bit is set to (1), then the local Image Descriptor, the rows of the image are displayed in a four-pass sequence, as in Fig. 3.16. (next slide)
- Here, the first pass displays rows 0 and 8, the second pass displays rows 4 and 12, and so on.

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*
⋮					

Fig. 3.16: GIF 4-pass interlace display row order.

3.2.2 JPEG

- **JPEG (Joint Photographic Experts Group)**: The most important current standard for image compression (.jpg, .jpeg, .jpe).

- The human vision system has some specific limitations (The eye–brain system cannot see extremely fine detail; those are dropped) 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).

- As an example, Fig. 3.17 shows 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.

A photo of a flower compressed with successively more lossy compression ratios from left to right.

Fig. 3.17: JPEG image with low quality specified by user.

PNG

- **PNG format**: standing for **Portable Network Graphics** — meant to supersede the GIF standard, and extends 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 8 X 8 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 signifier: 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.

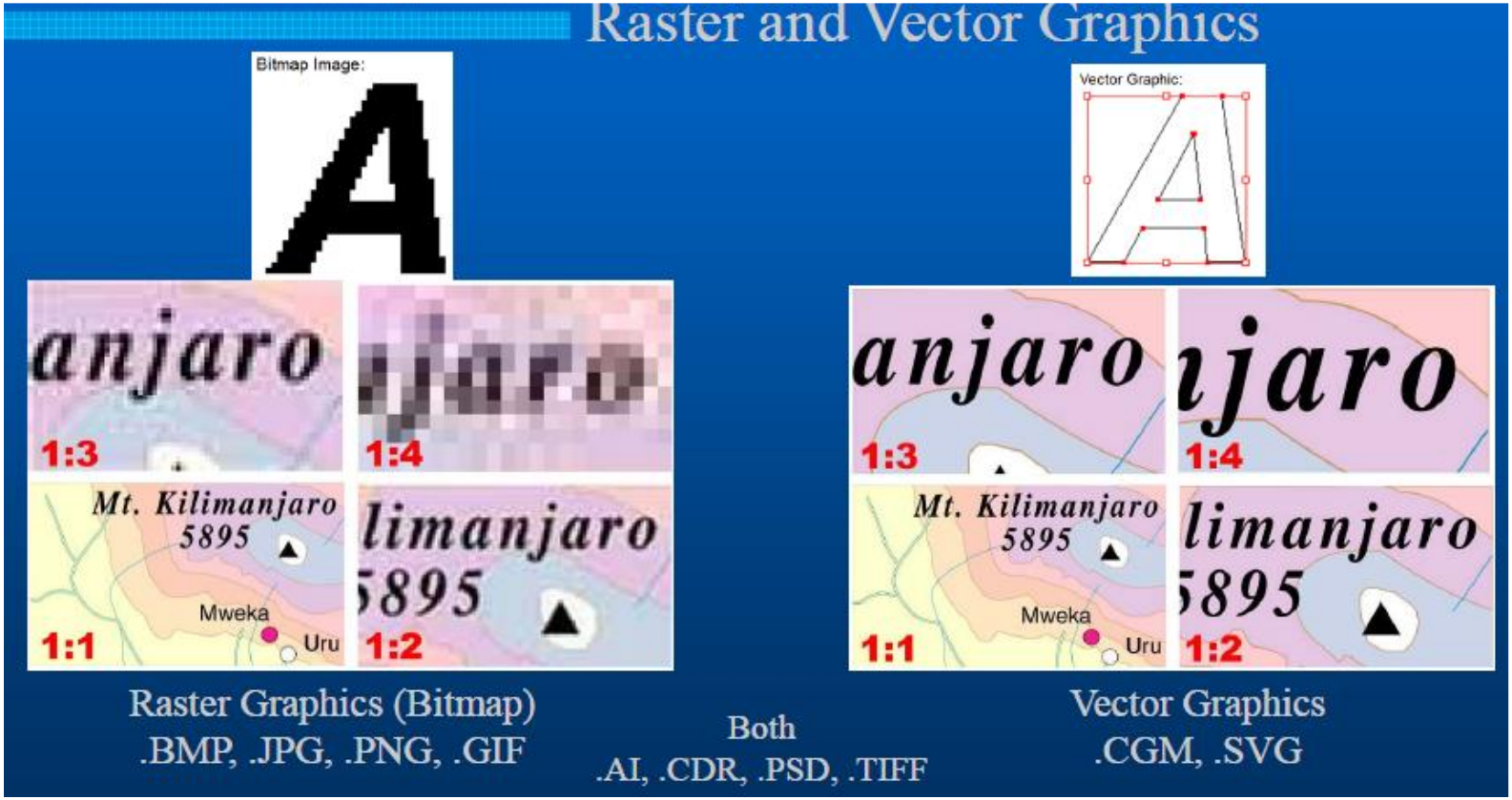
3.2.9 PS and PDF

- PostScript is an important language for typesetting, and many high-end printers have a PostScript interpreter built into them.
- PostScript is a vector-based, rather than pixel based, picture language: page elements are essentially defined in terms of vectors.
- PostScript includes vector/structured graphics as well as text
- Several popular graphics programs, such as Adobe Illustrator, use PostScript.
- Note, however, that the PostScript page description language does not provide compression; in fact, PostScript files are just stored as ASCII.

3.2.9 PS and PDF

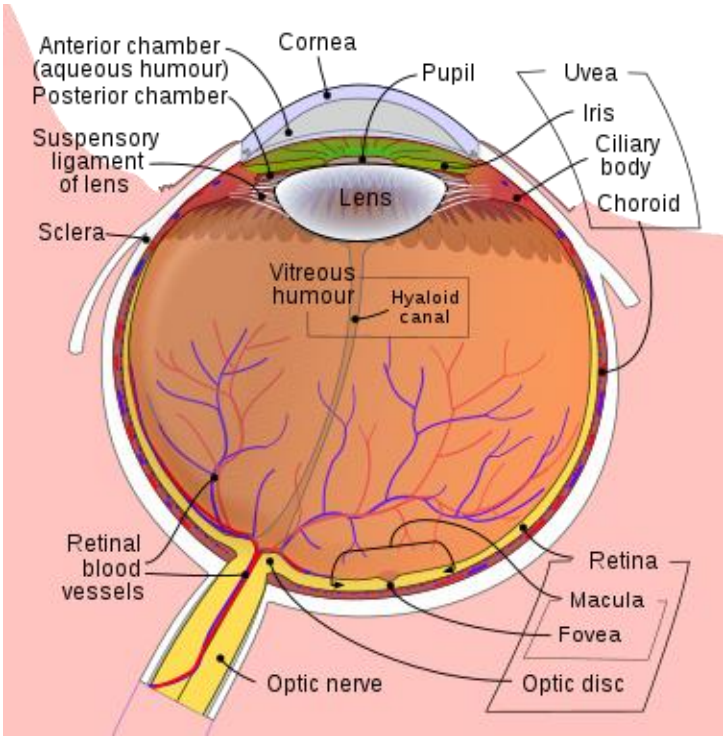
- Therefore, another text + figures language has largely superseded PostScript is *Portable Document Format (PDF) file format.*
- PDF files that do not include images have about the same compression ratio, while
- For files containing images, PDF may achieve higher compression ratios by using separate JPEG compression for the image content

Raster vs. Vector Graphics



نگاهی به آناتومی چشم انسان

- در کالبدشناسی چشم انسان، **گودهٔ مرکزی** (انگلیسی: Fovea centralis) به فرورفتگی ریزی در مرکز لکهٔ زرد گفته می‌شود که حاوی یاخته‌های مخروطی است که به علت افزایش طولشان به یاخته‌های استوانه‌ای شبیه شده‌اند.[۱]
- در مرکز شبکیه، ناحیه کوچکی به نام لکه زرد وجود دارد که در آن فقط نورون‌های مخروطی یافت می‌شود. در نواحی دورتر از لکه زرد به‌تدریج از تعداد نورون‌های مخروطی کاسته شده و بر تعداد نورون‌های استوانه‌ای افزوده می‌شود.



End of Chapter 3