Multimedia Systems

Media Representation: Image

Dr. Mojtaba Aajami

Graphics/Image Data Types

- The number of file formats used in multimedia continues to proliferate.

Adobe Premiere file formats.

Image	Sound	Video
BMP, GIF, JPG, EPS, PNG, PICT, PSD, TIF, TGA	AIFF, AAC, AC3, MP3, MPG, M4A, MOV, WMA	AVI, MOV, DV, FLV, MPG, WMA, WMV, SWF, M4V, MP4, MXF

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 since it contains no color.

8-bit Gray-level 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.

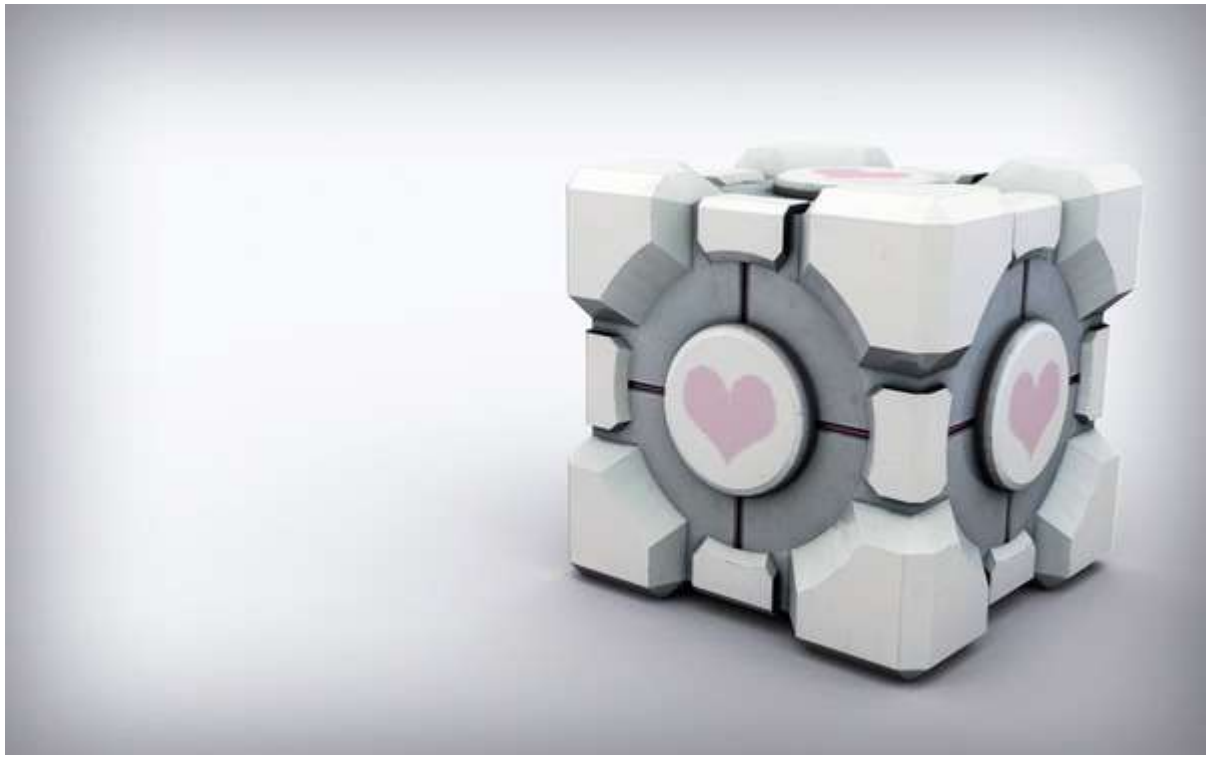
Grayscale Image

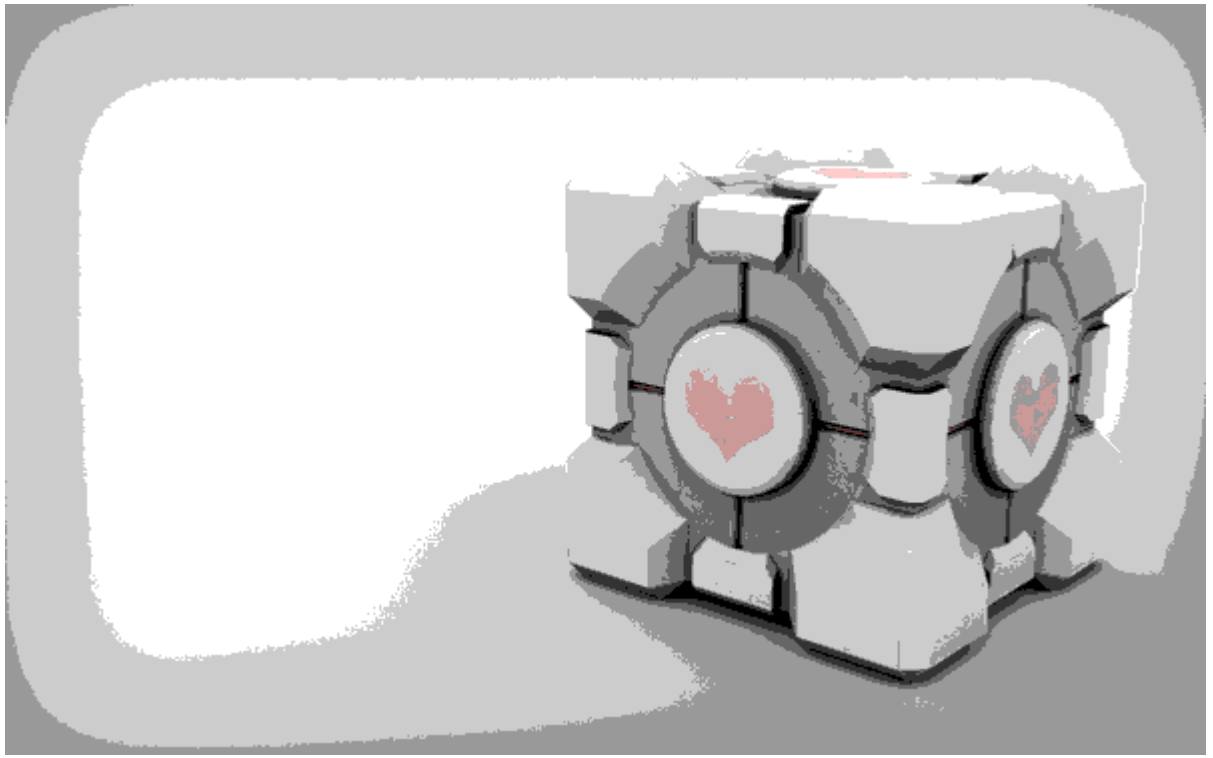
- Each pixel -> a byte (a value between 0 to 255)
  - 640x480 grayscale image requires 300 kB of storage (640 x 480 = 307, 200).
- Print a grayscale image on Black/White newspaper (or laser printer) ?

Grayscale Image: Dithering

- **Dithering** is a process which trades intensity resolution for spatial resolution to provide ability to print multi-level images on 2-level (1-bit) printers.

Dithering of Colored Images





Dithering of Colored Images

- Floyd-Steinberg dithering is a technique for reducing the color palette of an image (for example, to reduce its file size) whilst keeping as much of the perceived detail as possible.
- For each pixel in the original image, the nearest color to that pixel is chosen from a restricted palette and any "error" (difference in pixel color value, original - new) is distributed across the neighboring pixels as follows

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & * & \frac{7}{16} \\ \frac{3}{16} & \frac{5}{16} & \frac{1}{16} \end{array}$$

Floyd-Steinberg dithering: example

	X	+42
+18	+30	+6

24-bit Color Images

- In a color 24-bit image, each pixel is represented by three bytes, usually representing RGB.
  - This format supports 256x 256x 256 possible combined colors, or a total of 16,777,216 possible colors.
  - A 640×480, 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

(a)

(b)

(c)

(d)

High-resolution color and separate R, G, B color channel images.
(a): Example of 24-bit color image "forestfire.bmp".
(b, c, d): R, G, and B color channels for this image

Beyond 24-bit Color Images

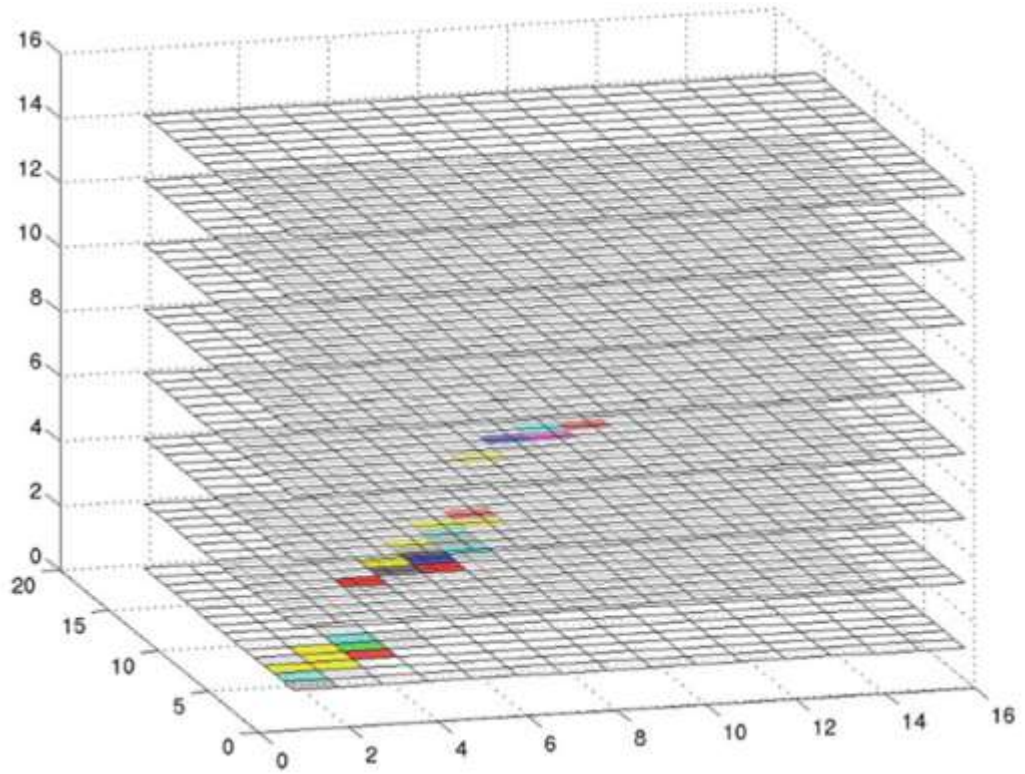
- More information about the scene being imaged can be gained by using more accuracy for pixel depth (64 bits, say); or by using special cameras that view more than just three colors (i.e., RGB).
  - use invisible light (e.g., infra-red, ultraviolet) for security cameras: “dark flash”.
  - use higher-dimensional medical images of skin (> 3-D) to diagnose skin cancer.
  - in satellite imaging, use high-Depth to obtain types of crop growth, etc.
  - Multispectral/ Hyperspectral

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 image files use the concept of a **(color index) lookup table** to store color information.
  - Basically, the image stores not color, but instead just a set of bytes, each of which is actually an index into a table with 3-byte values that specify the color for a pixel with that lookup table index.

Color histogram

- All the colors in a 24-bit image were collected in a 256 × 256 × 256 set of cells, along with the count of how many pixels belong to each of these colors stored in that cell.



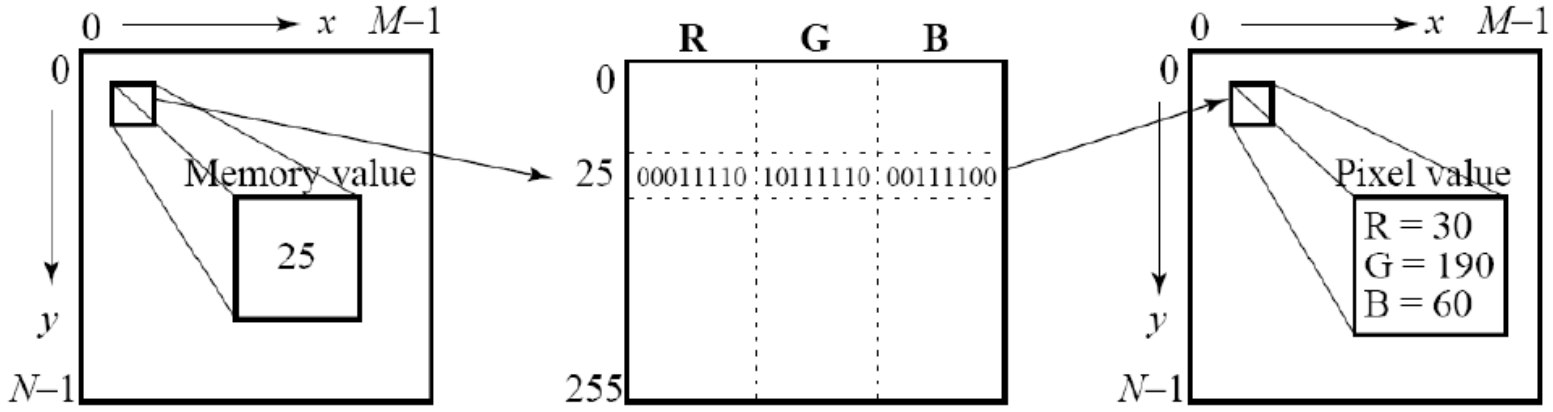
Splitting and Merging

- The algorithm starts from the initial assumption that the entire image is a single region.
- It then computes the homogeneity criterion to see if it is TRUE.
- If FALSE, then the square region is **split** into four smaller regions.
- This process is then repeated on each of the sub-regions until no further splitting is necessary.
- These small square regions are then **merged** if they are similar to give larger irregular regions.

Example of an 8-bit color image

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 LUT for 8-bit color images.

How to devise a color look-up table

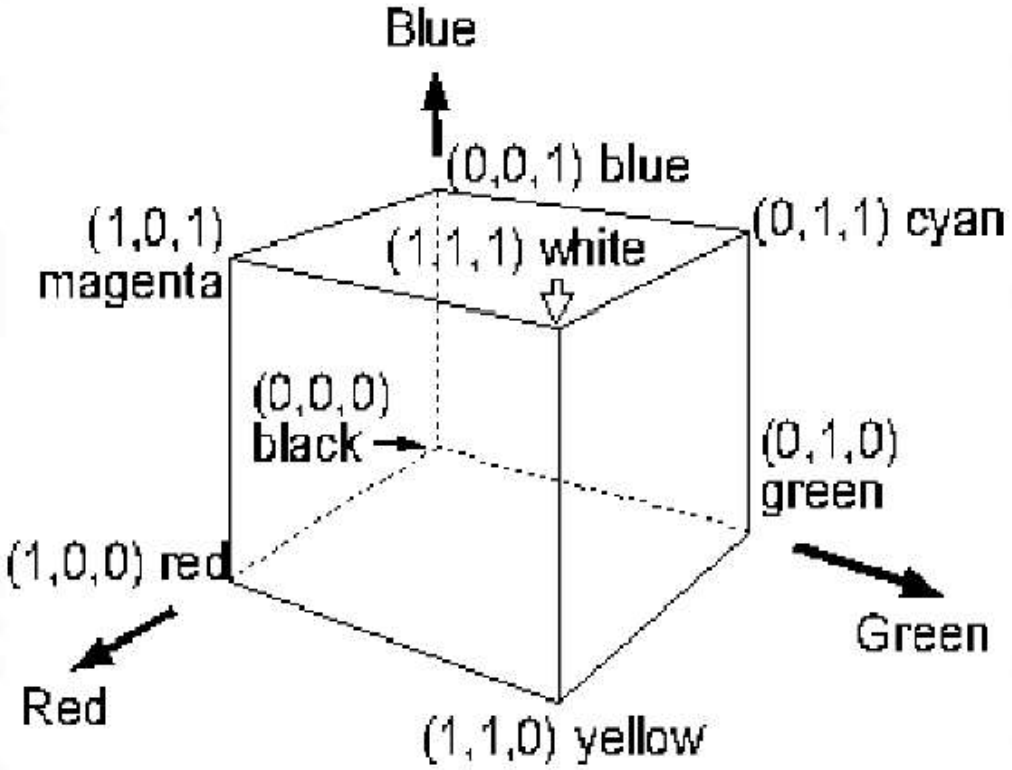
- The most straightforward way to make 8-bit look-up color out of 24-bit color would be to divide the RGB cube into equal slices in each dimension.
  - The centers of each of the resulting cubes would serve as the entries in the color LUT, while simply scaling the RGB ranges 0..255 into the appropriate ranges would generate the 8-bit codes.
  - 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.
  - To shrink R and G, we could simply divide the R or G byte value by (256/8)=32 and then truncate.

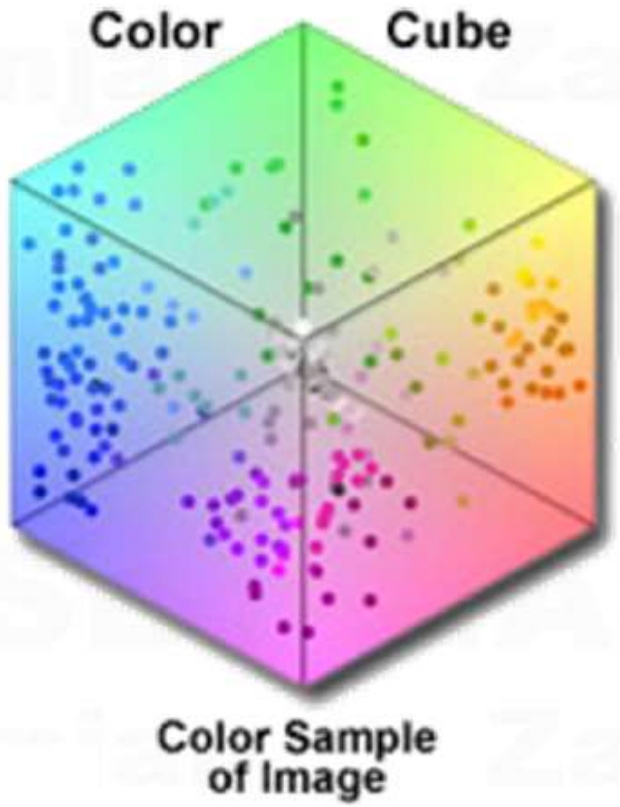
Problem of LUT

- However, what tends to happen with this simple scheme is that edge artifacts appear in the image.
- The reason is that if a slight change in RGB results in shifting to a new code, an edge appears, and this can be quite annoying perceptually. For Example:
  - Using 3 bits for R, two color RGB(31,128,128), RGB(32,128,128) would become RGB(0,128,128), RGB(32,128,128)

Median-cut algorithm

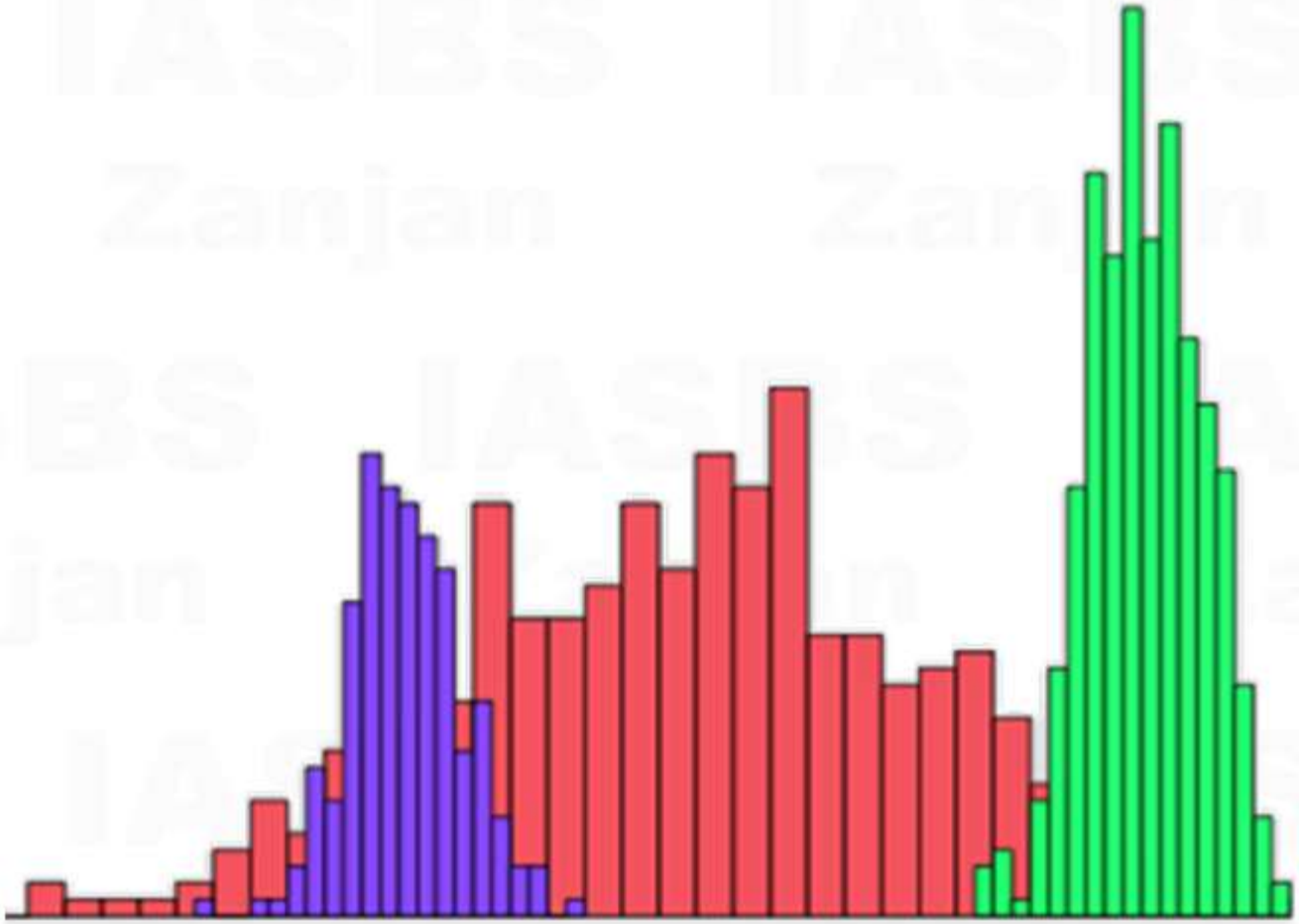
Step 1: Create a "cube" of the colors in the pixels of an image by using each color component (R, G, and B) as an axis (e.g. x,y,z)





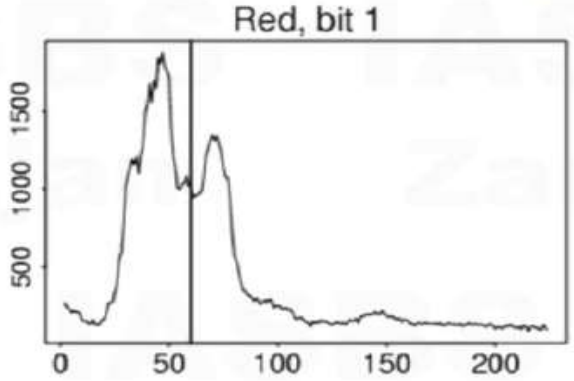
Median-cut algorithm

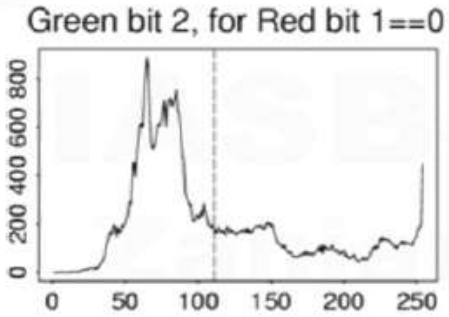
Step 2: Calculate the range of each color component (R, G, B)

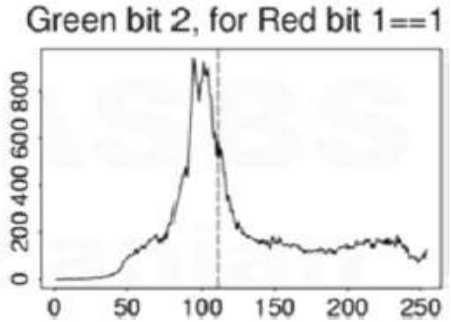


Median-cut algorithm

Step 3: For the component with the largest range, calculate the median value, M

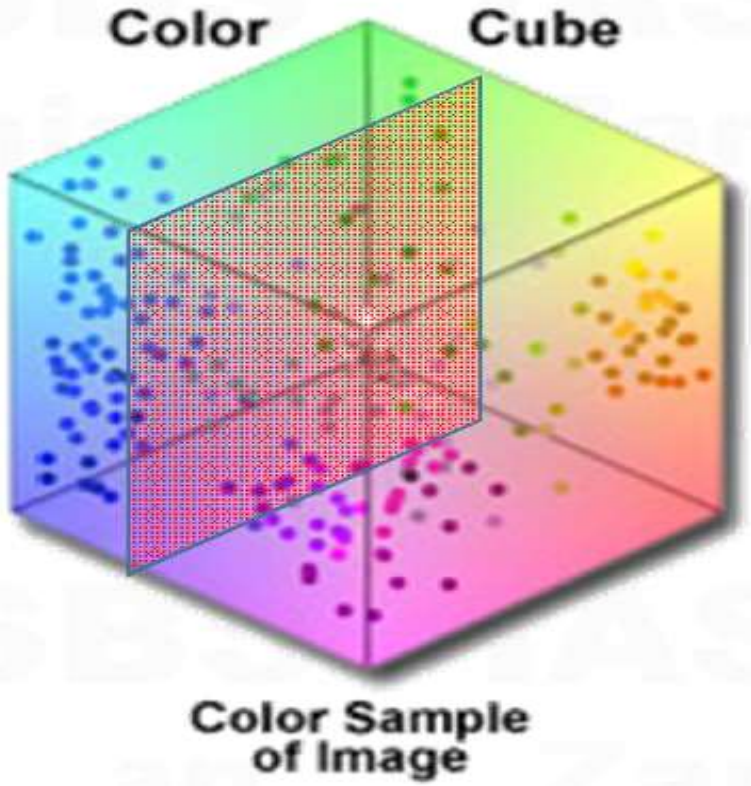






Median-cut algorithm

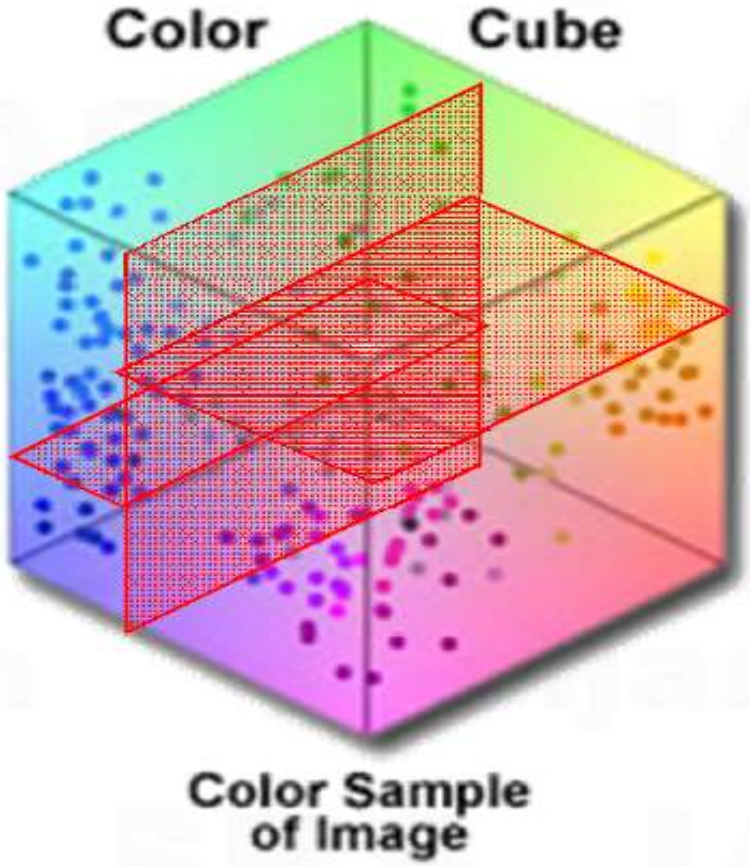
Step 4: Split the "cube" of colors according to the median



Median-cut algorithm

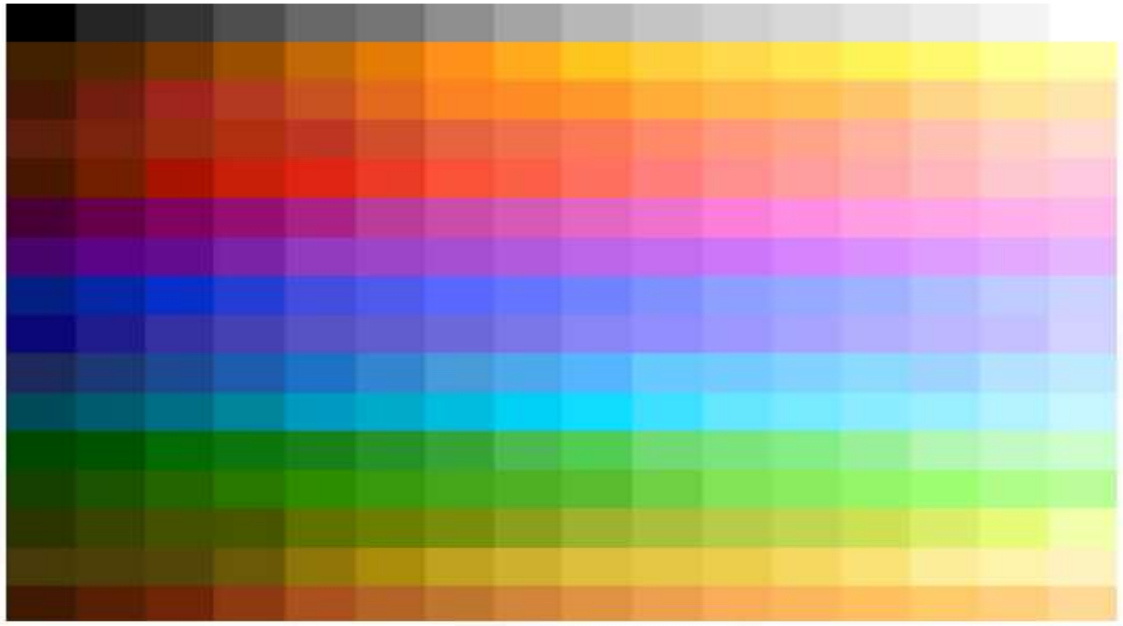
If the number of cubes is equal to our chosen number of desired colors, exit the loop

Step 5: For each color cube, calculate the range of each component, choose the cube which contains the largest range, and repeat.



Median-cut algorithm

Step 6: For each cube, averaging the colors in each box.



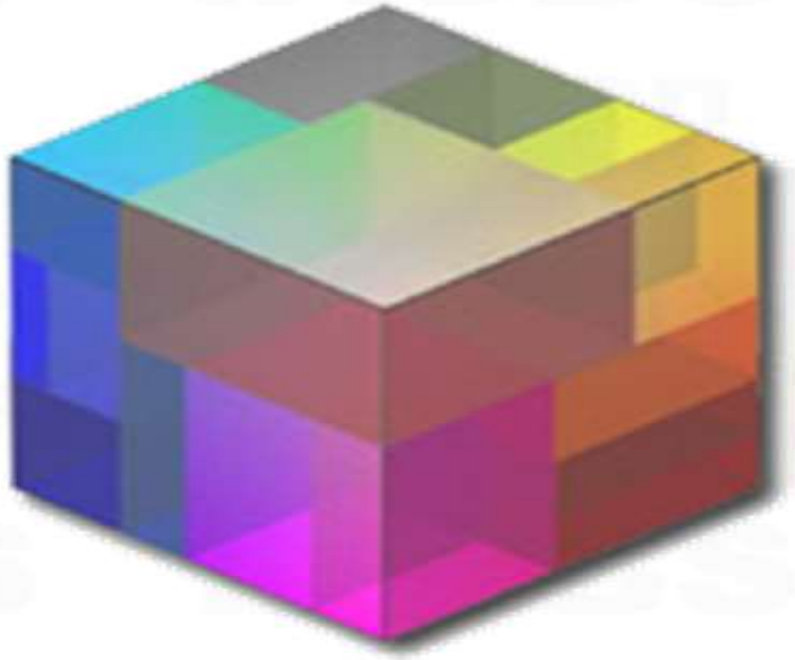


Image Formats: GIF

- Stands for **G**raphics **I**nterchange **F**ormat.
- Limited to 8-bit (256) color images only, best suited for images with few distinctive colors (e.g., graphics or drawing).
- GIF standard supports **interlacing**
- Available in two versions
  - **GIF87a**: The original specification.
  - **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.

GIF: interlacing

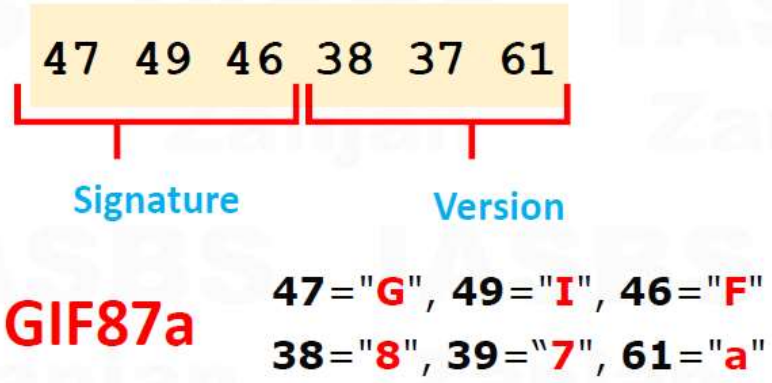
- A mechanism that makes images appear faster on screen and gives the reader a quick preview of the full area of the picture.
- A four-pass sequence

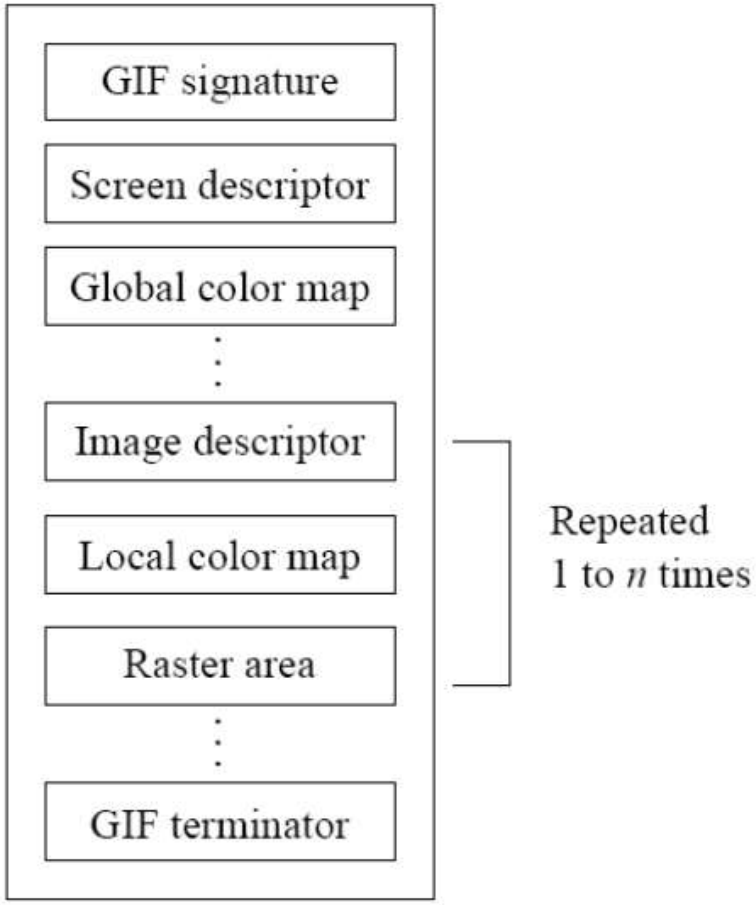
GIF: animation

- Allows timed display of images in sequence, giving the effect of animation.

GIF: File format

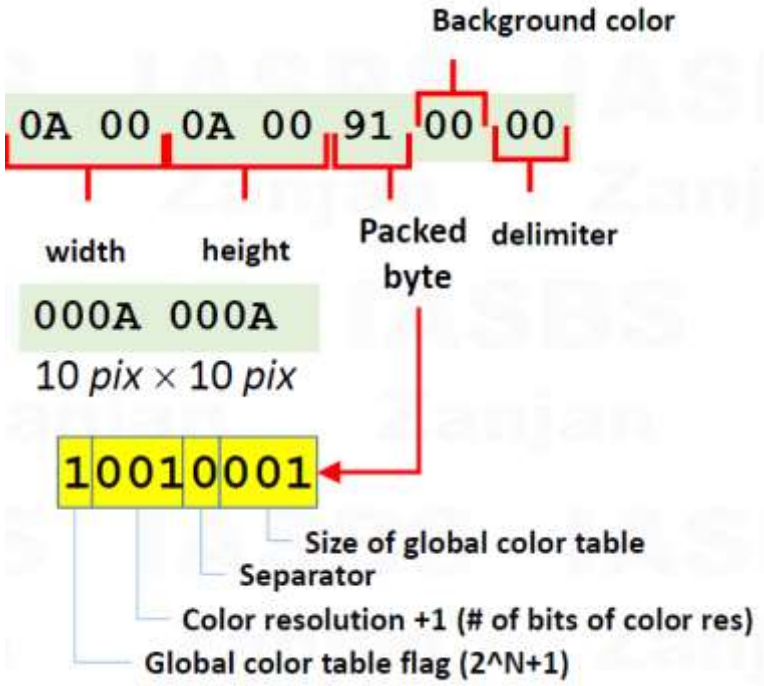
- All GIF files must start with a header or ***Signature*** block. The header takes up the first six bytes of the file and its corresponding bytes:





GIF: File format

- The *Screen Descriptor*
  - immediately follows the header
  - a seven-byte set of flags



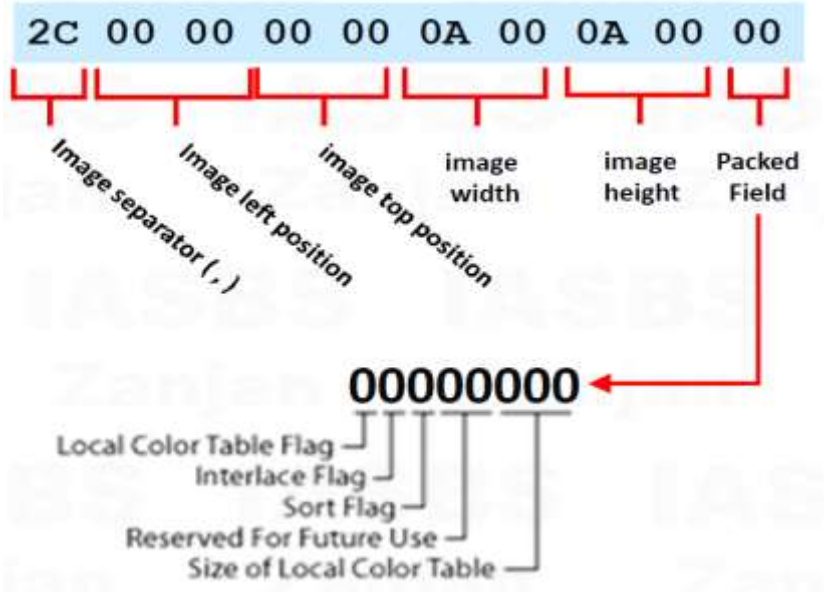
7	6	5	4	3	2	1	0	Byte
Screen width								1
								2
Screen height								3
								4
m	cr			0	pixel			5
Background								6
0	0	0	0	0	0	0	0	7

GIF: File format

- the *Global Color Table*
  - A list of all the colors that can be in the image
  - Each GIF has its own color palette
  - Not every GIF has to specify a global color table

GIF: File format

- The *Image descriptor*
  - A set of attributes that belong to every image in the file.
  - Each image begins with an image descriptor block.

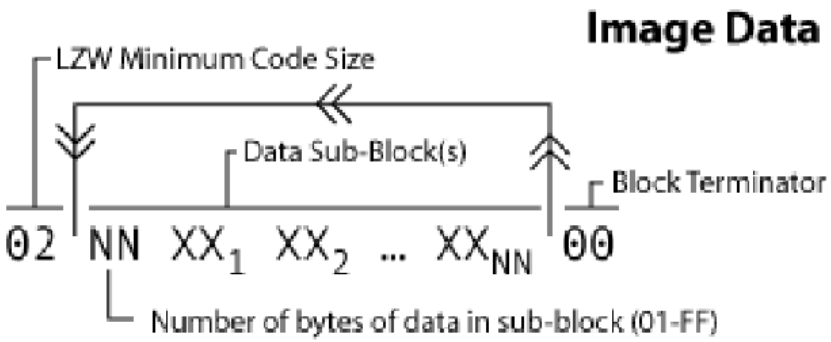


Bits 7	6	5	4	3	2	1	0	Byte
0	0	1	0	1	1	0	0	1
Image left								2 3
Image top								4 5
Image width								6 7
Image height								8 9
m	i	0	0	0	pixel			10

GIF: File format

- the *Raster Area* or *Image Data*

```
02 16 8C 2D 99 87 2A 1C DC
33 A0 02 75 EC 95 FA A8 DE
60 8C 04 91 4C 01 00
```



When to use GIF?

- Images with areas of flat color
  - logos, line art, icons & cartoon-like illustrations
- Not much suited for photographic images
- Suited for adding simple animation
- Whenever transparency in image is required

PNG

- **Portable Network Graphics** — meant to supersede the GIF standard, and extends it in important ways.
  - Support for up to 48 bits of color information
  - Files may contain gamma-correction information for the correct display of color images, as well as alpha-channel information for control of transparency.
  - The display progressively displays pixels in a 2-dimensional fashion by showing a few pixels at a time over seven passes