Image Representation

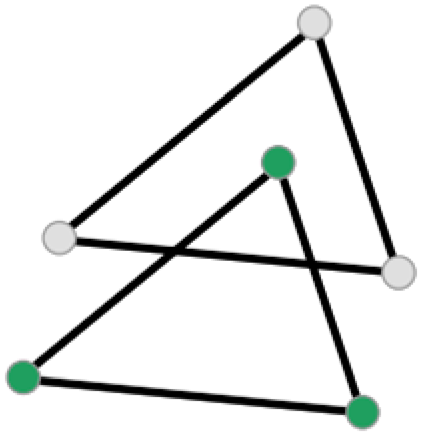
LECTURE-2

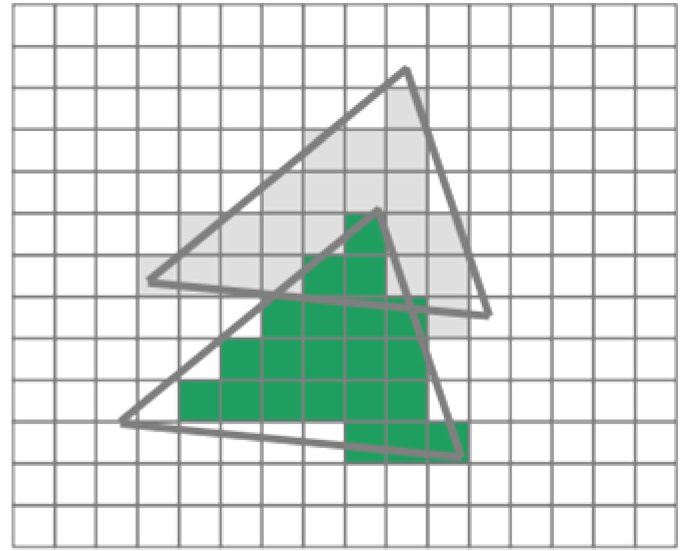
Digital Image

- Virtually all computing devices in use today are digital.
- Data is representation in a discrete form using patterns of binary digits (bits) that can encode numbers within finite ranges and with limited precision.
- The images we perceive in our environment are analogue and formed by the complex interaction between light and physical objects.
- Suppose we wish to capture an image and represent it in a computer.
- Since we do not have infinite storage (bits), it follows that we must convert that analogue signal into more limited digital form. This conversion process is called **sampling**.
  - That means the process of getting finite number of values from image (cont. func.) is called **sampling.**

Raster Image Representation

- The problem of digital image representation is that it can not encode fine levels of details, nor dynamic color ranges.
- The solution of this problem is to break the image up into a regular grid that we call **raster**.
- A **raster image**, stored in the form of colored pixels on a fixed and finite grid.
- Each grid cell is “Picture Cell” often known as **pixel**.
  - Smallest element of the screen.
  - Atomic unit of the image.
- Pixels are arranged in a row column fashion to form a rectangular picture area (raster).





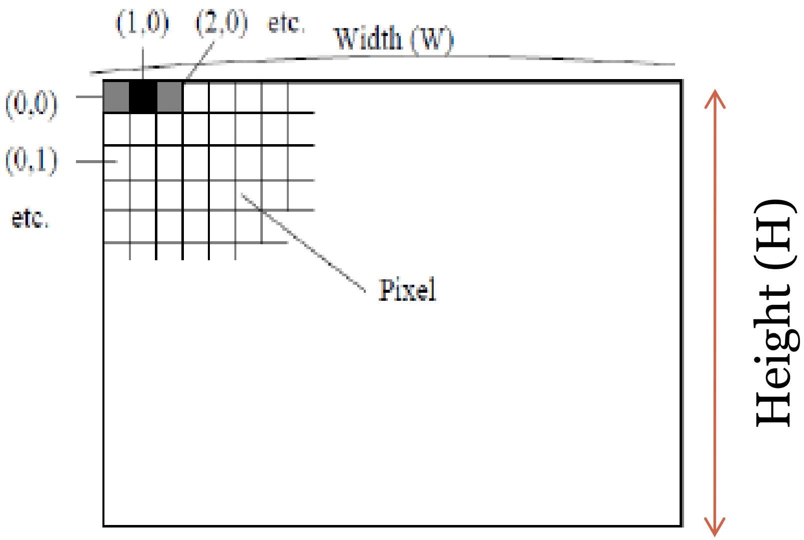
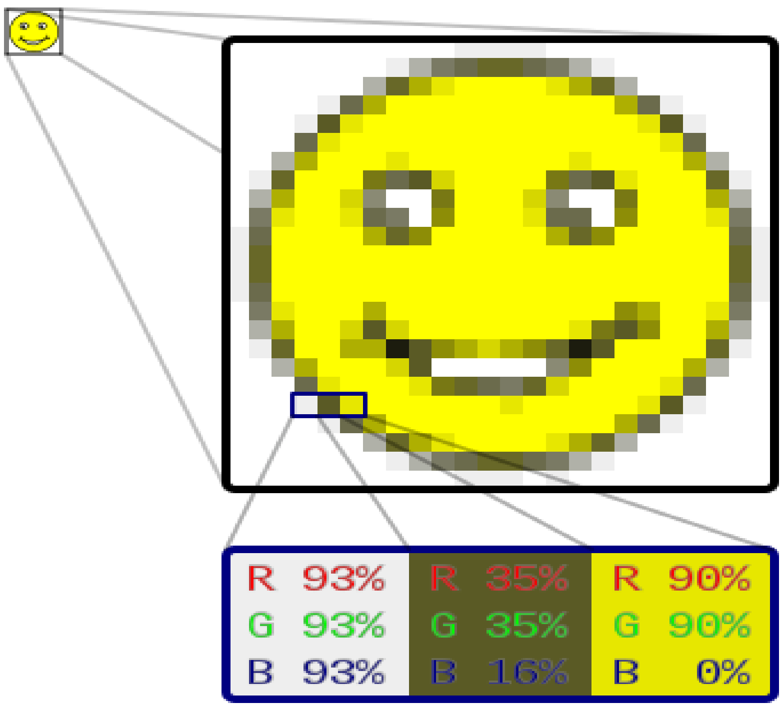


Image Raster



Raster Image Representation

- Total number of pixels in an image is a function of size of the image and resolution.
  - 3x2 inch image at a resolution of 300 pixels per inch would have total pixel= (3*300)* (2*300)=540000 pixel.
- Number of pixels per unit length (e.g. inch) in the horizontal as well as vertical direction is known as **resolution**.
- **Image size** is given by-

  Total number of pixels in the horizontal direction **x** Total number of pixels in the vertical direction
  - Example: 512 x 512 or 640 x 480
- Aspect Ration= Width/Height

  [measured in unit length or number of pixels]
  - 2 x2 inch image and 512 x 512 image have aspect ratio of 1/1

Raster Image Representation

- Individual pixels in an image can be referenced by their coordinates.
- Left Handed System:
  - Origin->top left corner
  - (W-1, H-1) pixel->bottom right corner.
- Right Handed System:
  - Origin->lower left corner
  - (W-1, H-1) pixel->upper right corner
- Raster images are pixel based and format is resolution specific (Better resolution->better quality).
- Common raster format: TIFF, JPEG, GIF, PNG and BMP

Examples

- Compute the size of a 640 x 480 image at 240 pixels per inch.

  (640/240) x (480/240)= 2 2/3 x 2 inch
- Compute the resolution of a 2 x 2 inch image that has 512 x 512 pixels.

  512/2 = 256 pixels per inch
- If an image has a height of 2 inches and aspect ratio of 1.5, what is its width?

  Aspect Ratio= Width/Height

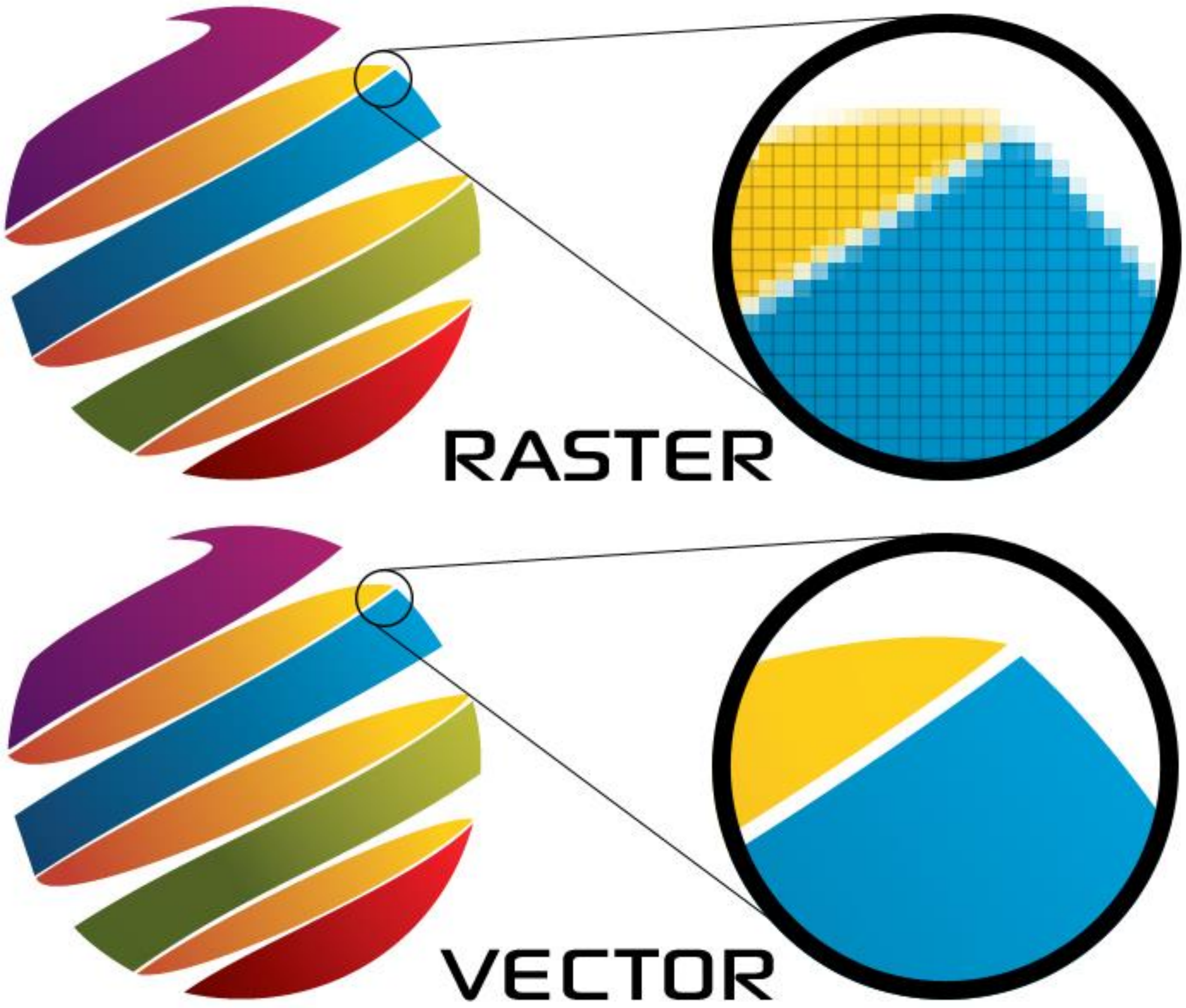
  Width=Aspect Ratio x Height= 1.5 x 2= 3 inches

Problems

- If we want to resize a 1024 x 768 image to one that is 640 pixels wide with the same aspect ratio, what would be the height of the resized image?
- If we want to cut a 512 x 512 sub image out from the center of an 800 x 600 image, what are the coordinates of the pixel in the large image that is at the lower left corner of the small image (Right Handed System)?
- How to convert the coordinate of a pixel at (x, y) in the left handed coordinate system into its coordinate (x´, y´) in the lower left corner as origin coordinate system (right handed system)?

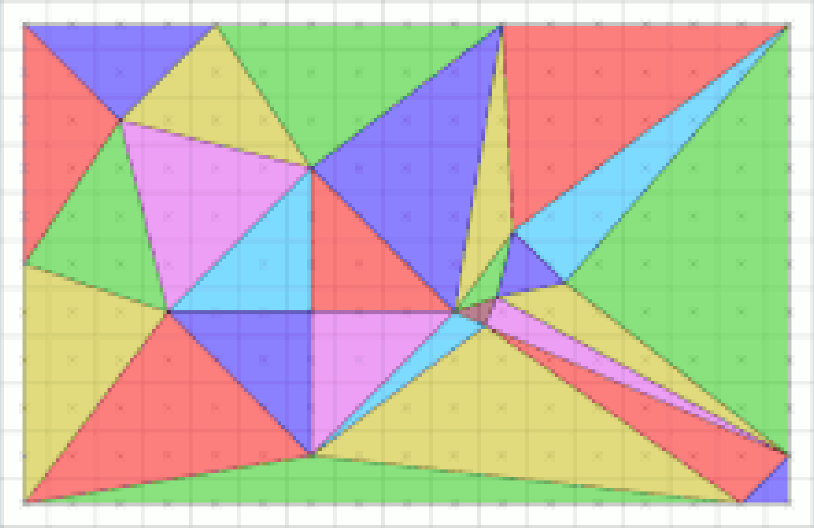
Vector Graphics Representation

- Based on the mathematical formulas that define geometric primitives such as polygons, lines, curves, circles & rectangles.
- Stored in the form of mathematical representation of the image element.
- Points, lines and curves may be scaled up or down to any resolution with no aliasing.
- The points determine the direction of vector path.
  - Each path may have properties including values for stroke color, shape, curve, thickness and fill.


RASTER
VECTOR

Rasterization & Vectorization

- Rasterization:
  - Process of converting vector graphics format to raster graphics.
- Vectorization:
  - Process of converting raster graphics to vector graphics format

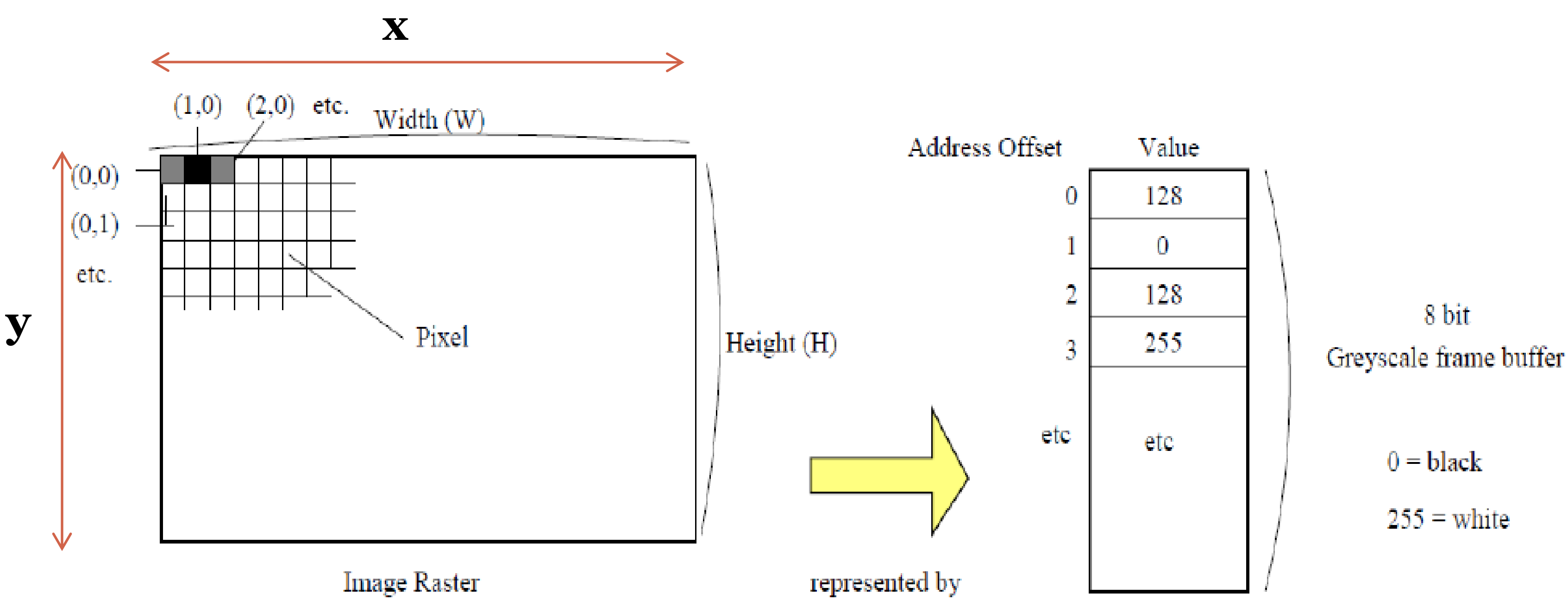


Hardware Frame Buffer

- A large piece of memory reserved for holding the bit mapped image (color of each pixel) that is continuously being sent to the display device is called frame buffer.
- Frame buffer is stored in main memory.

Grayscale Frame Buffer

- Simplest form of frame buffer.
- Encodes pixels using various shades of gray.
- Pixels are encoded as an unsigned integer using 8 bit (1 byte) and can present 256 (2^8) different shades.
- **0** means black, **255** means white and mid intensity grey is **128**.
- Image of width W pixels & height H pixels requires W x H bytes of memory for its frame buffer.



- Frame buffer is arranged so that the first byte of memory corresponds to the pixel at coordinates (0,0) [left handed system].
- Addressing then proceed in a left to right and then top to down manner.
- Pixel (x, y) would be stored at buffer offset A where

$$A = x + W y$$

A bytes from the start of the frame buffer.
- Sometimes, we use the term scanline to refer to a full row of pixels. A scanline is therefore W pixels wide.

True Color Frame Buffer

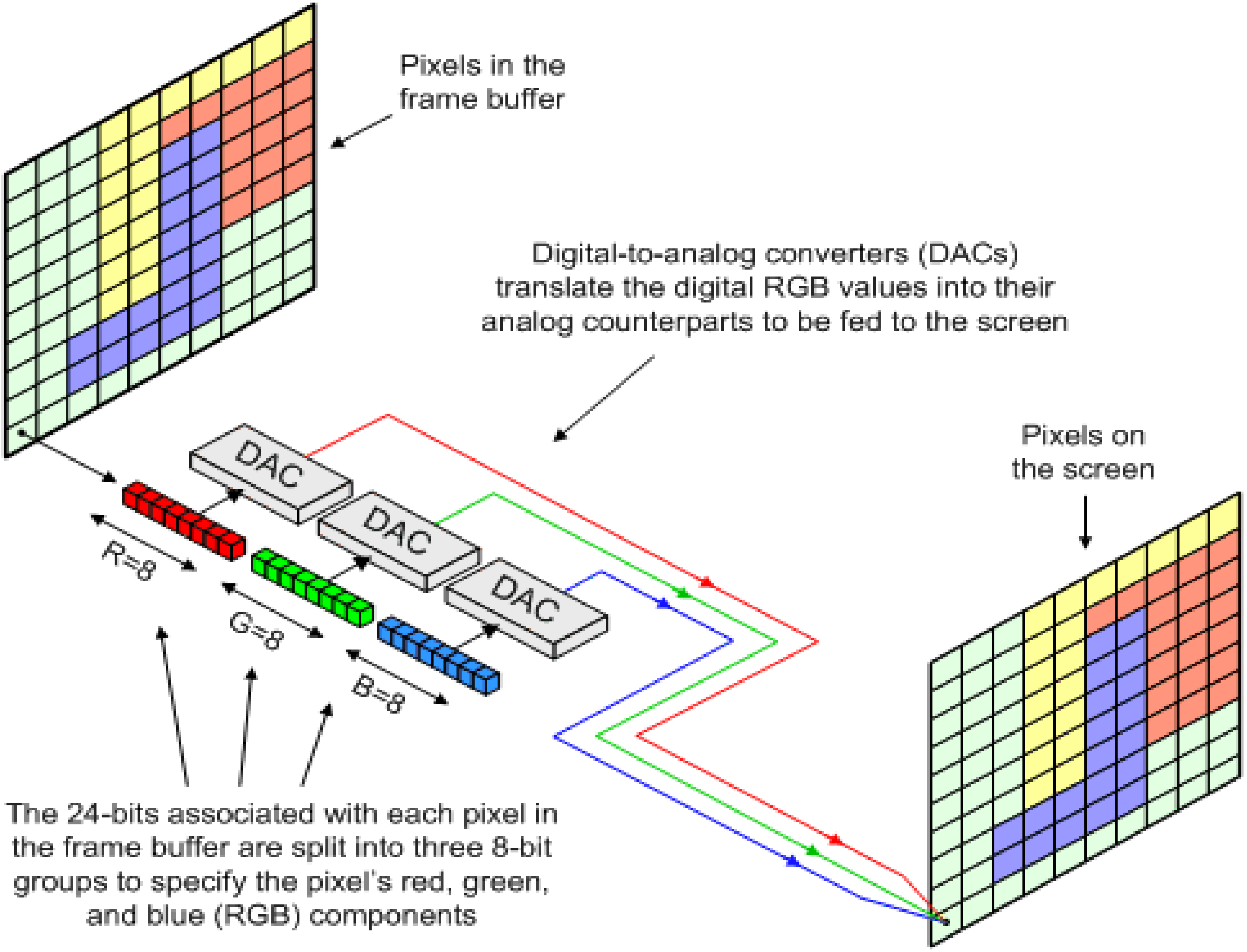
- Represents color images.
- RGB color value for each pixel is stored directly within the frame buffer.
- Each pixel requires at least 3 bytes. One byte for each primary color.
  - Possible variation of color 2^{24} (3 byte=24 bit)
  - Require 24 bit of storage per pixel.
- Pixel (0,0)->Buffer location 0,1,2

  Pixel (1,0)-> Buffer location 3,4,5

So, Pixel (x,y)-> S=3W

A= 3x+Sy

Where S is the stride of the display.



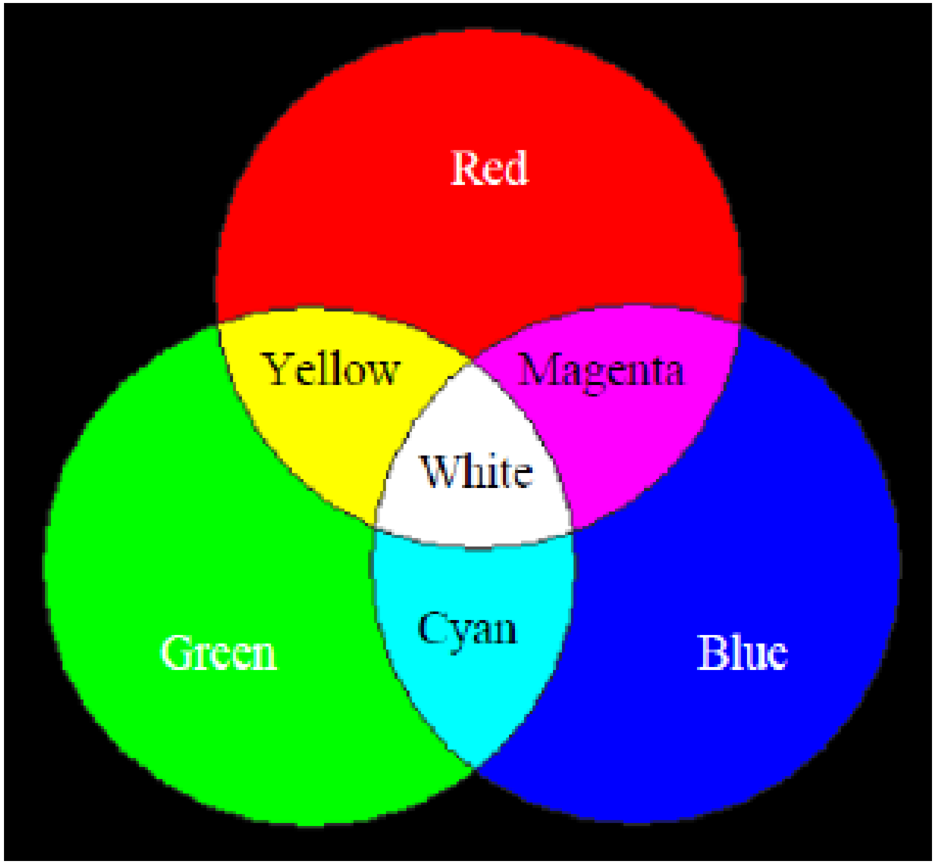
Graphical representation of 24-bit "true color"

Representation of Color

- Red, Green and Blue are called “additive primaries” because we obtain other, non primary color by blending (adding) different quantities of Red, Green and Blue light.
- We can make the additive secondary colors by making pairs of primaries: Red, Green and Blue.
  - Red + Green=Yellow
  - Green + Blue= Cyan
  - Red + Green +Blue = White
  - No mix of RGB= Black

Additive Vs Subtractive Primaries

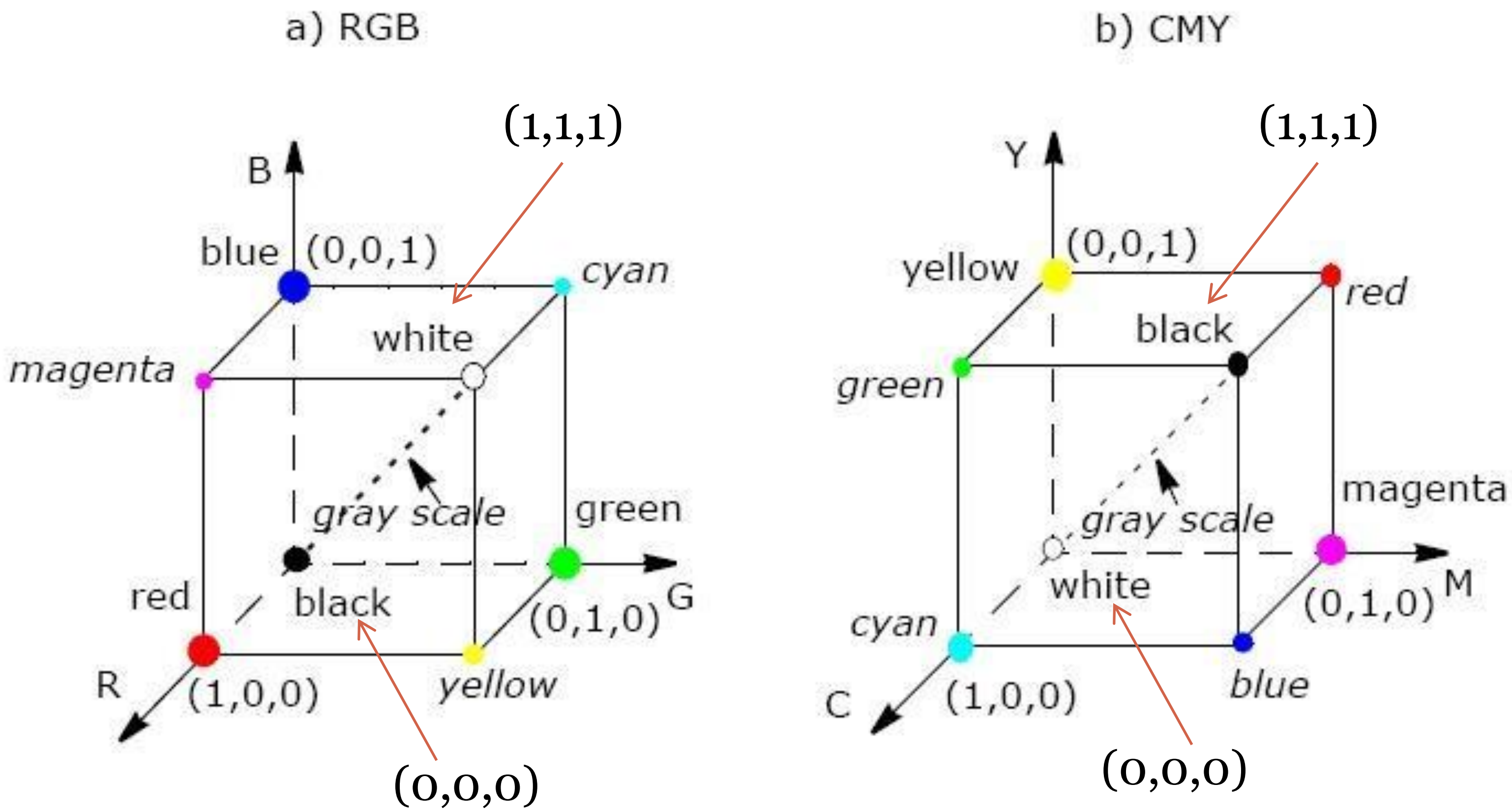
- Additive color model are used in light.
  - Start with black and add primary component to make desired shade.
- Subtractive color models are used with paint.
  - Start with white and take away appropriate primary components to yield a desired color
  - Example: Subtract Red from White, remains Green + Blue which is Cyan.


Red
Yellow
Magenta
White
Green
Cyan
Blue

RGB Color Space

- Each primary color can take intensity value ranging from 0 (lowest) to 1 (highest). [R,G,B values divided by 255 to change to 0..1]
- Additive color model
  - R + G +B->white
  - (0,0,0)->Black , (1,1,1)->white
  - (1,1,0)->Yellow, (0.7,0.7,0.7)-> Gray
- Used in display monitor.

Color Cube


a) RGB
(1,1,1)
B
blue
(0,0,1)
cyan
white
magenta
gray scale
green
red
black
G
(0,1,0)
R
(1,0,0)
yellow
(0,0,0)
b) CMY
(1,1,1)
Y
(0,0,1)
yellow
red
black
green
magenta
gray scale
white
M
(0,1,0)
cyan
C
(1,0,0)
blue
(0,0,0)

CMYK Color Space

- Complementary color model of RGB.
- Subtractive color model.
  - C=W-R, M=W-G, Y=W-B
  - Ink reduces the light that would otherwise be reflected. Ink subtracts brightness from white.
  - K= min (c, y, m) , K for key which is a black color for printing purpose.
- Used in printer.

Formula for Conversion

Suppose, RGB value of violet is (238,130,238)or (0.933,0.509,0.933).

Now we want to convert this value to CMY.

C=1-R=1-0.933=0.067
M=1-G=1-0.509=0.491
Y=1-B=1-0.933=0.067

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

Self Study

Grayscale Conversion

the end