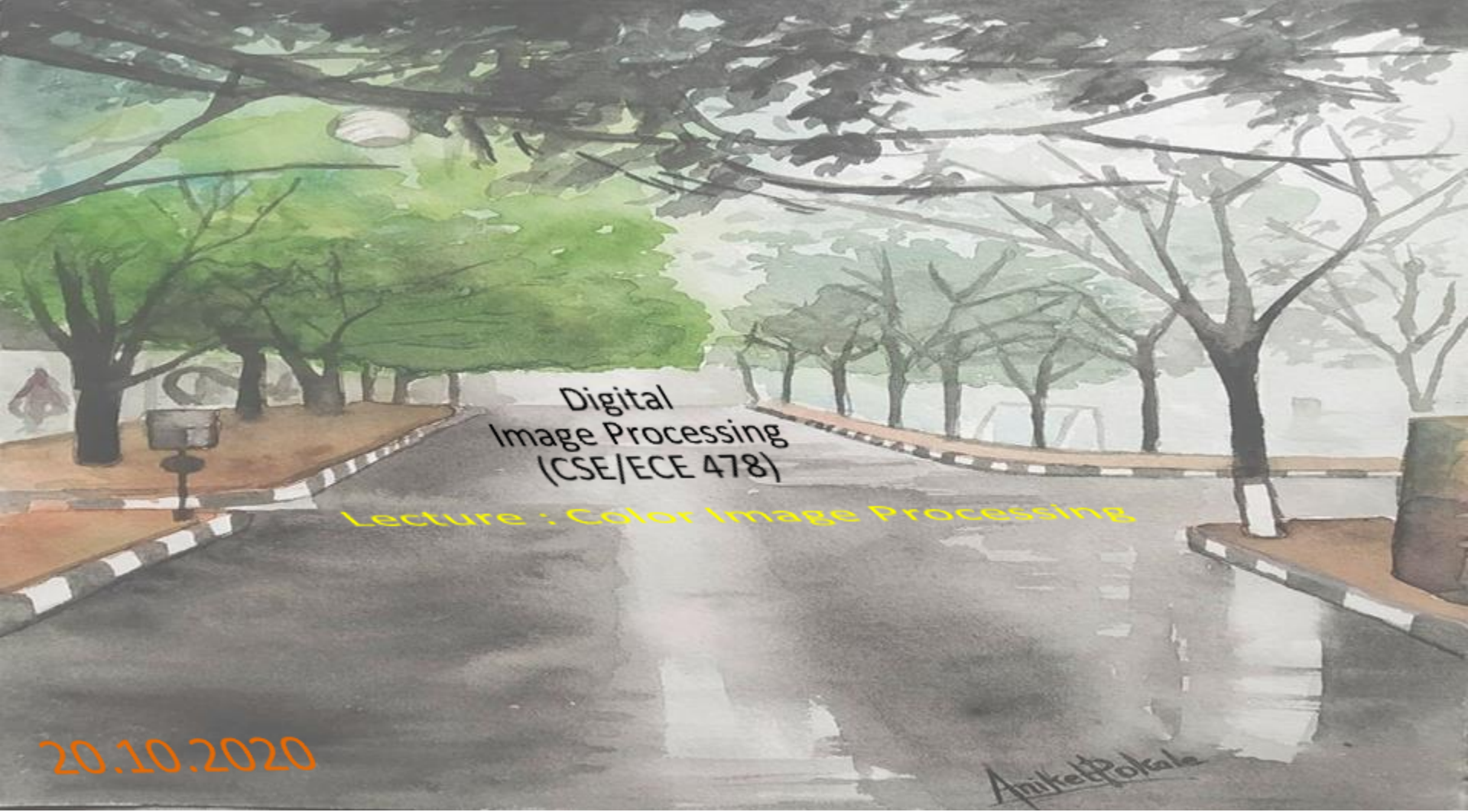
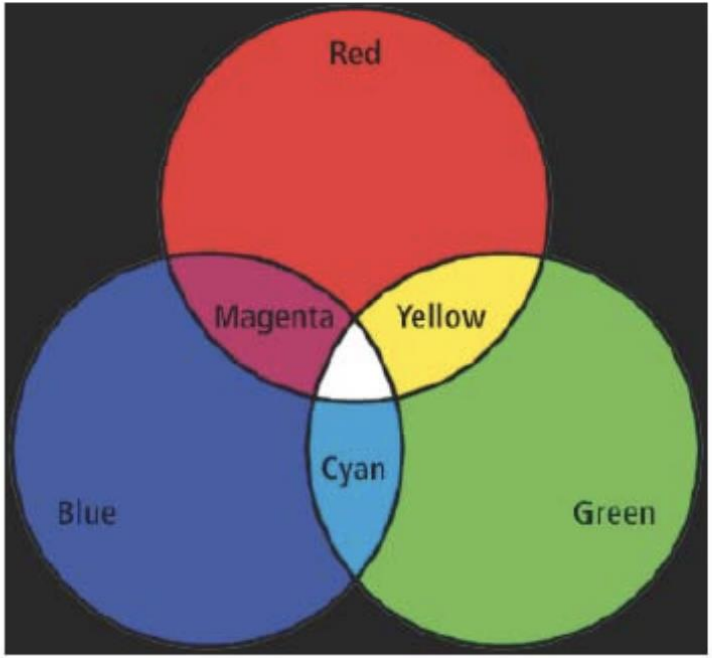
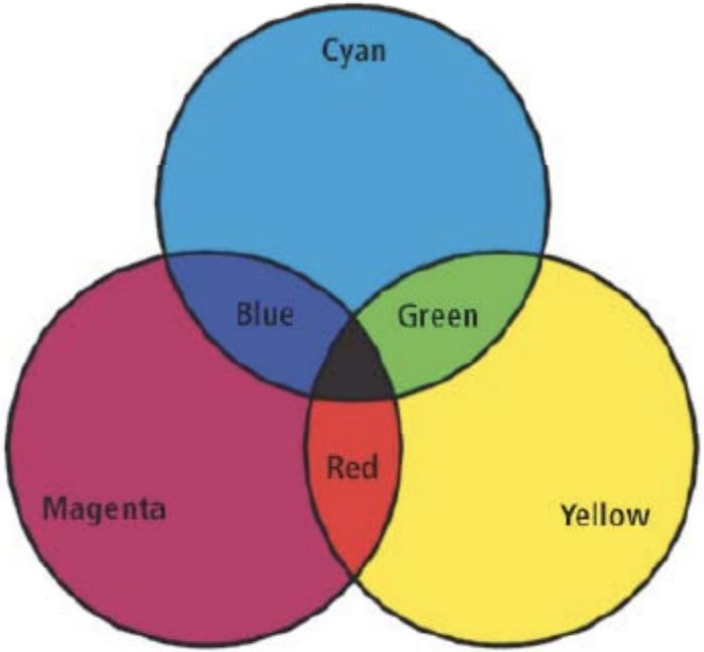

Digital
Image Processing
(CSE/ECE 478)
Lecture : Color Image Processing
20.10.2020
Anirudh Pokala

Additive



Magenta = Red + Blue
Cyan = Blue + Green
Yellow = Green + Red

Subtractive



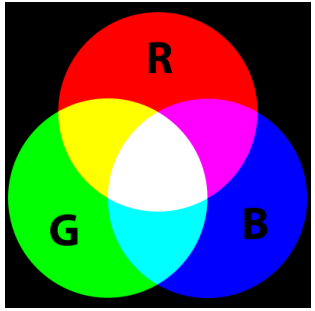
Magenta = White - Green
Cyan = White - Red
Yellow = White - Blue

Full color

Red

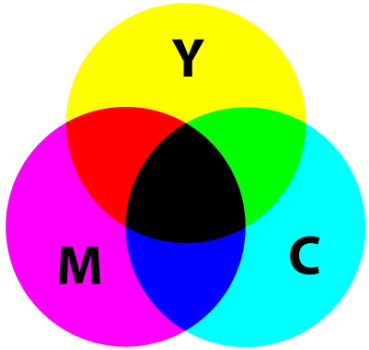
Green

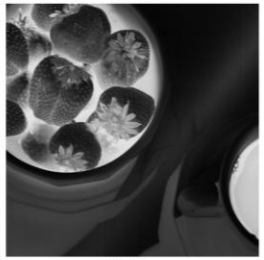
Blue



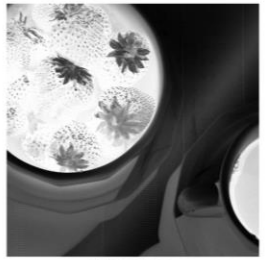
White

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



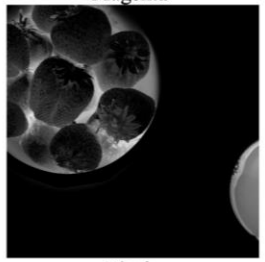


Cyan



Magenta

Yellow



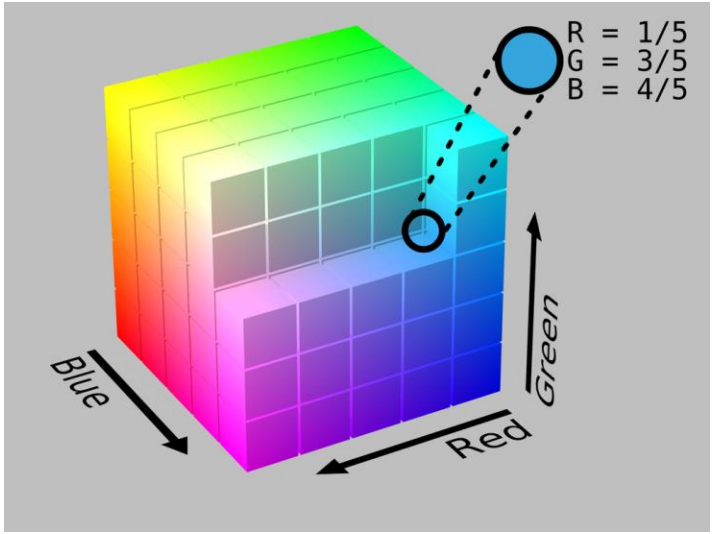
Black

Color Models

- RGB
- HSI / HSY
- CIE LAB

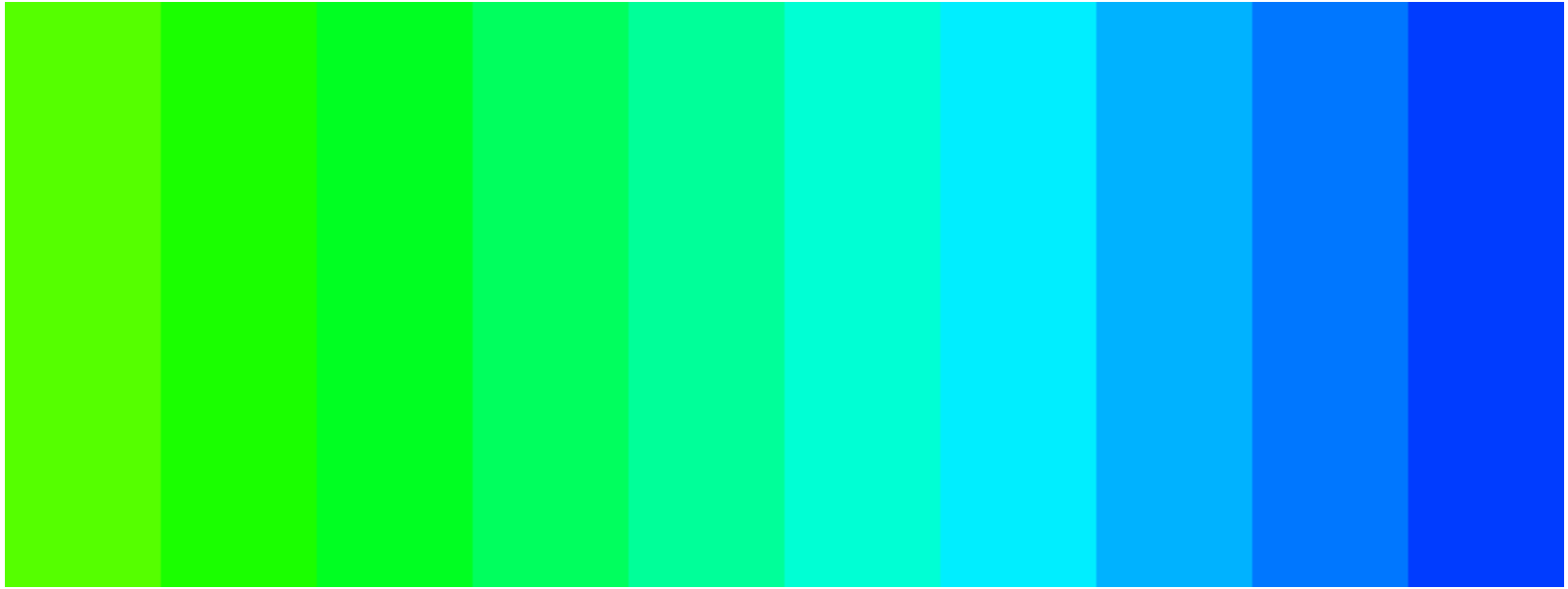
RGB color space

- Primary colors
- Additive color model $f(x, y) = \alpha_1 R + \alpha_2 G + \alpha_3 B$
- Perceptually non uniform



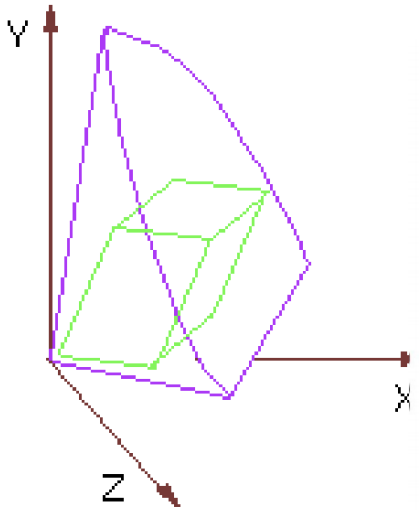
Courtesy: wikipedia

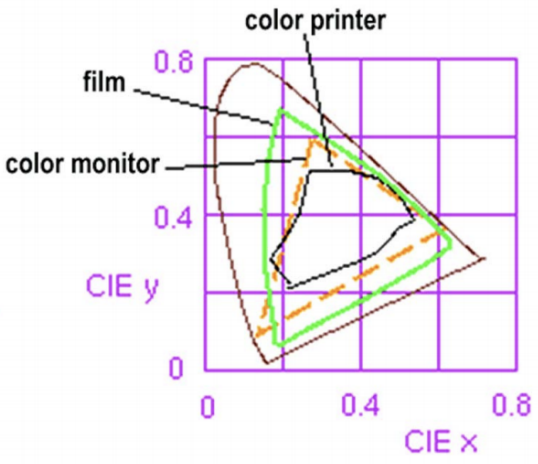
RGB : Non-Uniform Perceptual Space

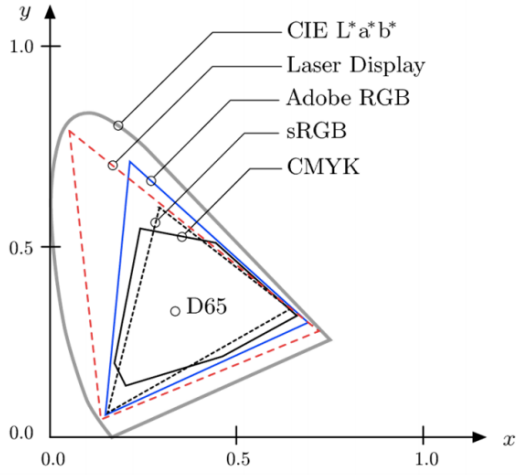


Device Color Gamuts

- The RGB color cube sits within CIE color space
- We can use the CIE chromaticity diagram to compare the gamuts of various devices
- E.g. compare color printer and monitor color gamuts





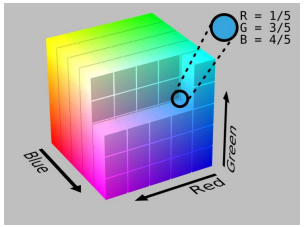


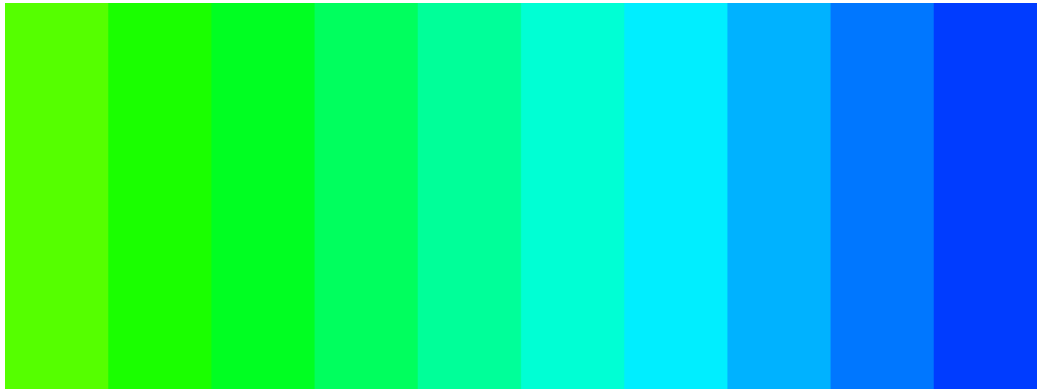
Measuring Color Differences

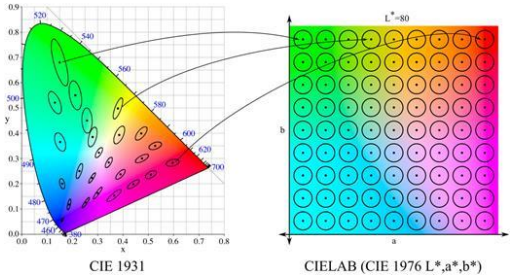
- Due to its uniformity with respect to human perception, differences between colors in L*a*b* color space can be determined as **euclidean distance**

$$\begin{aligned}\mathrm{ColorDist}_{\mathrm{Lab}}(\mathbf{C}_1, \mathbf{C}_2) &= \|\mathbf{C}_1 - \mathbf{C}_2\| \\ &= \sqrt{(L_1^* - L_2^*)^2 + (a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2}\end{aligned}$$

Perceptually Uniform Color Spaces



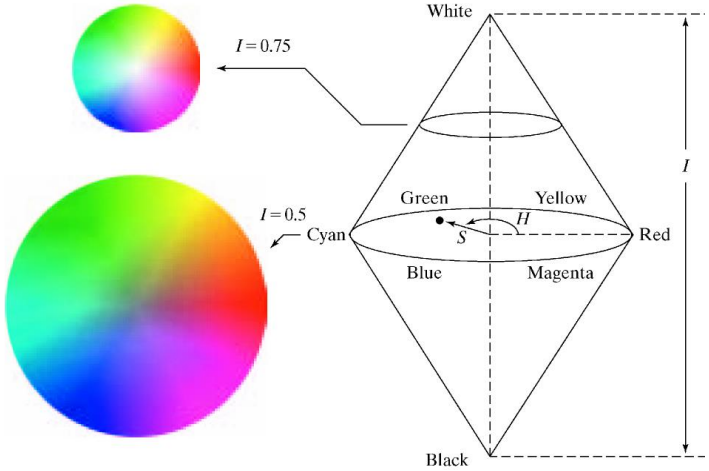




https://matplotlib.org/3.1.1/tutorials/colors/colormaps.html

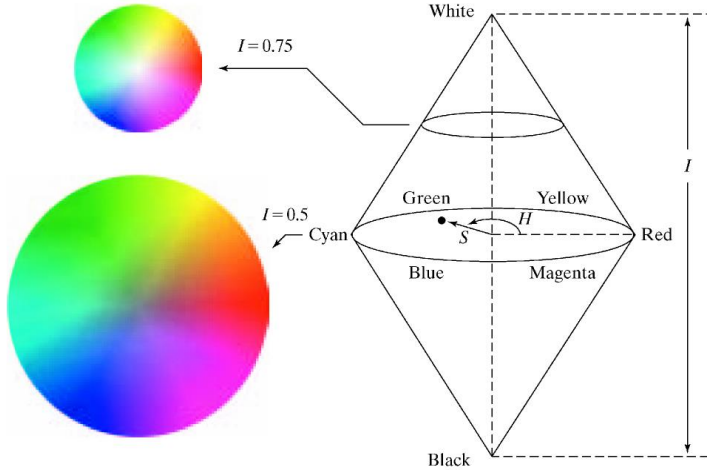
HSI color space

- Hue (H):
  - dominant colour perceived by an observer
  - When we call an object red or orange, we refer to its hue.
- Saturation (S):
  - Amount of white light mixed with a hue
  - Pure colors are fully saturated.
    - Pink (red+white) is less saturated.
- Brightness (I) : achromatic notion of intensity



HSI color space

- **HUE**
  - A subjective measure of color
  - Average human eye can perceive ~200 different colors
- **Saturation**
  - Relative purity of the color. Mixing more "white" with color reduces its saturation.
  - Pink has the same hue as red but less saturation
- **Intensity**
  - The brightness or darkness of an object

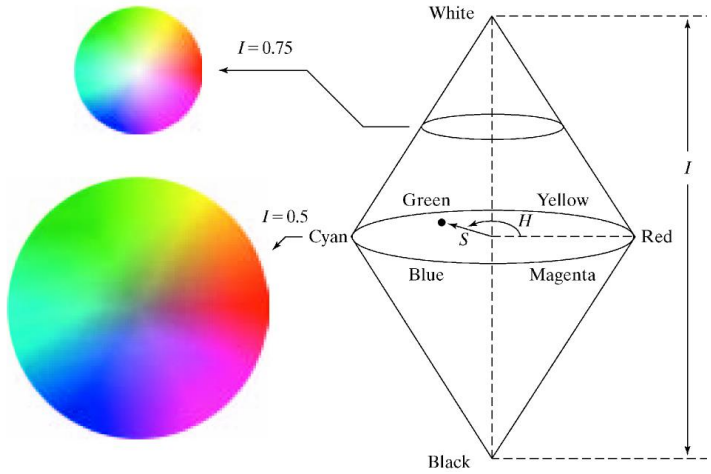


RGB : great for color generation
HSI : great for color description

HSI color space

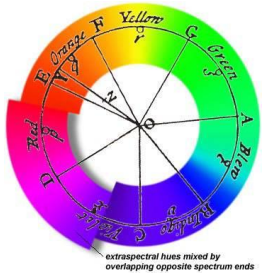
- **Hue** is defined as an angle
  - 0 degrees is **RED**
  - 120 degrees is **GREEN**
  - 240 degrees is **BLUE**

Saturation = % of distance from center : [0,1]



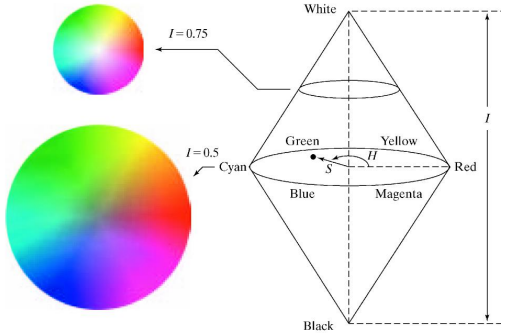
- **Intensity** is denoted as the distance "up" the axis from black.
  - Values range from 0 to 1

RGB : great for color generation
HSI : great for color description



Example: RGB to HSV Conversion

Original RGB image



H_{HSV} S_{HSV} V_{HSV}

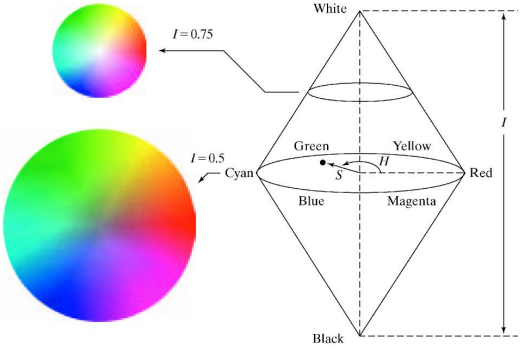
HSV values in grayscale

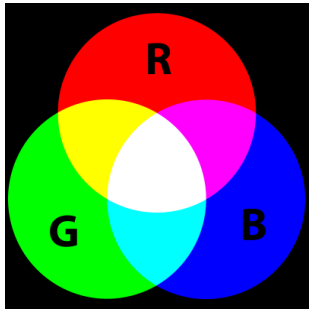
Full color

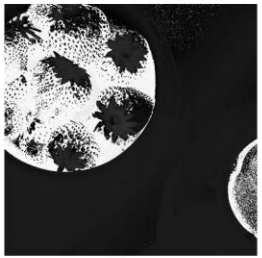
Red

Green

Blue





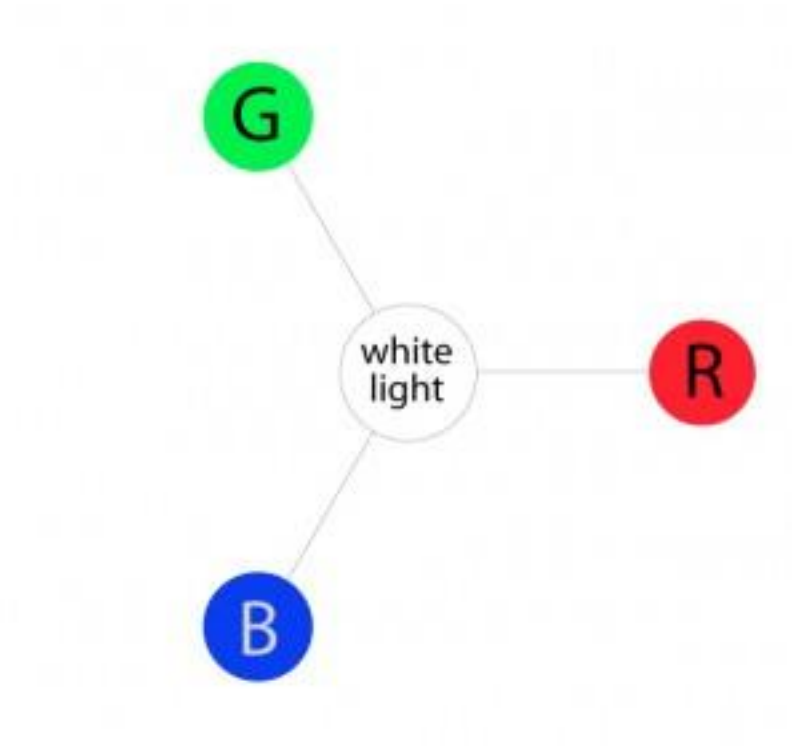


Hue

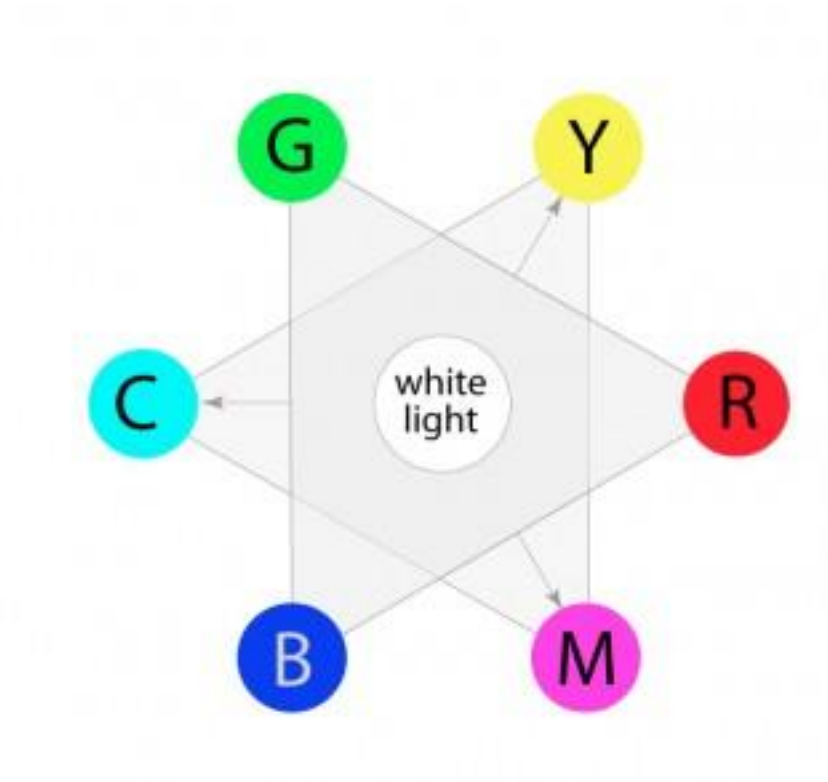
Saturation

Intensity

Primary Colors

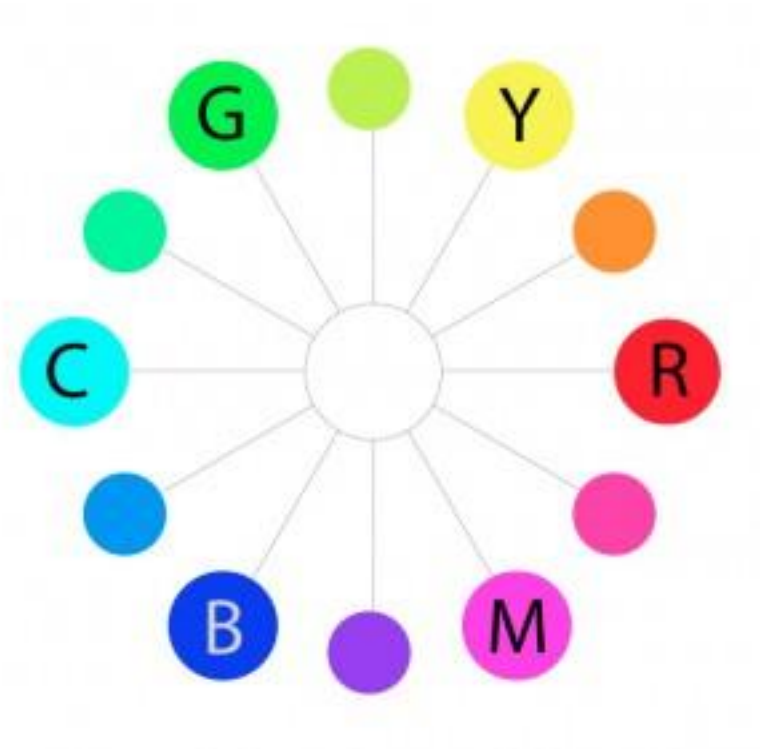


Secondary Colors

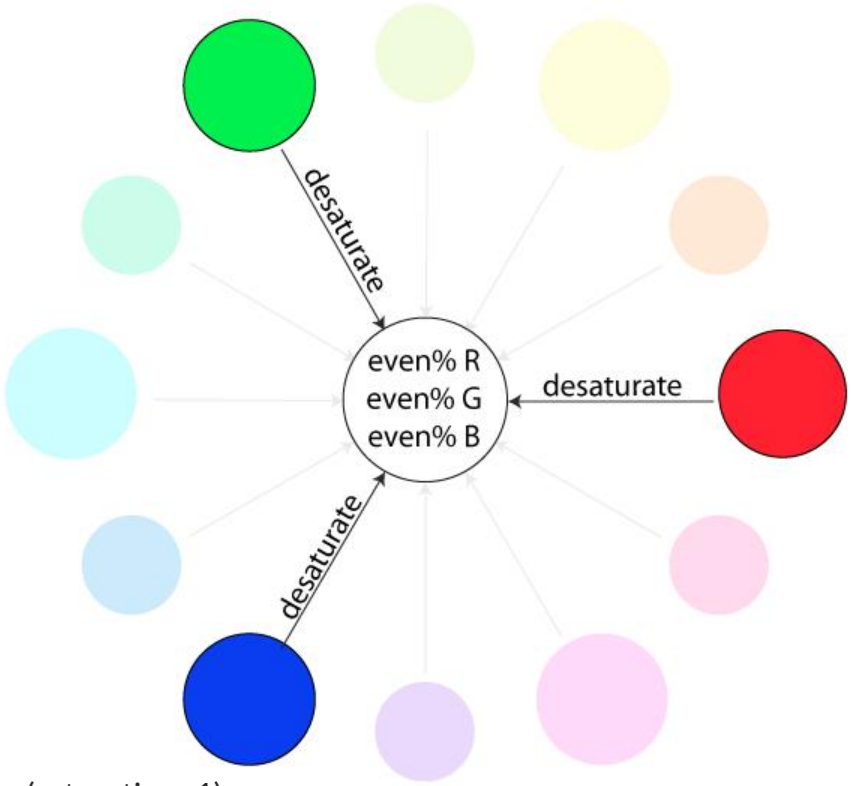


Tertiary Colors

‘Hue’ circle

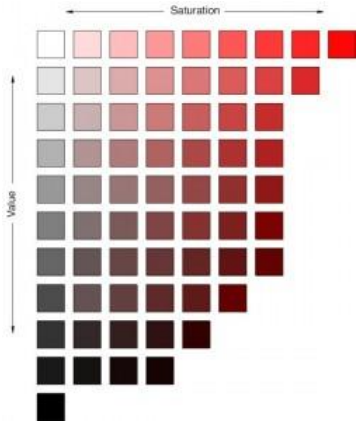


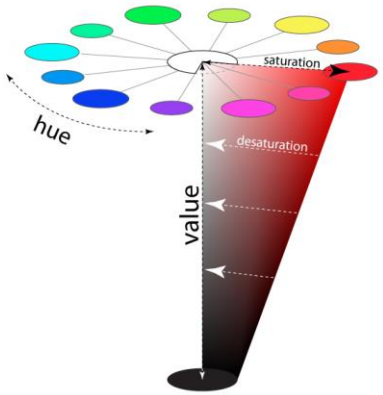
Saturation

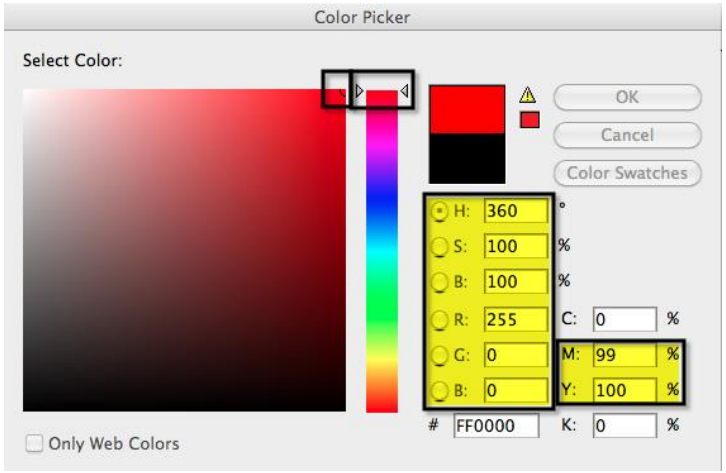


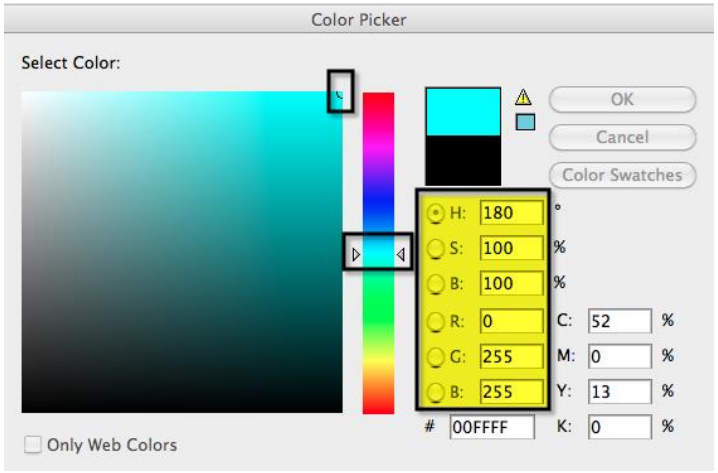
- At the rim of circle are pure hues (saturation=1)
- Hue dominates less as you move into the center
- At centre of the wheel, no hue dominates (saturation=0)

HSI


Saturation
Value


saturation
hue
desaturation
value


Color Picker
Select Color:
OK
Cancel
Color Swatches
H: 360 °
S: 100 %
B: 100 %
R: 255
G: 0
B: 0
C: 0 %
M: 99 %
Y: 100 %
K: 0 %
FF0000
Only Web Colors


Color Picker
Select Color:
OK
Cancel
Color Swatches
H: 180 °
S: 100 %
B: 100 %
R: 0
G: 255
B: 255
C: 52 %
M: 0 %
Y: 13 %
K: 0 %
00FFFF
Only Web Colors

RGB --> HSV

RGB to HSV conversion formula

The *R*,*G*,*B* values are divided by 255 to change the range from 0..255 to 0..1:

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$Cmax = \max(R', G', B')$$

$$Cmin = \min(R', G', B')$$

$$\Delta = Cmax - Cmin$$

Hue calculation:

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

Value calculation:

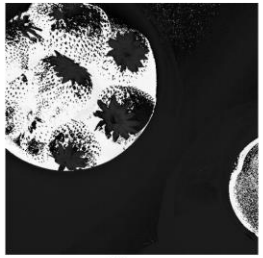
$$V = Cmax$$

Full color

Red

Green

Blue



Hue

Saturation

Intensity

Full color

Full color image

Cyan Magenta Yellow

C,M,Y components.

Red Green Blue

R,G,B components.

Hue Saturation Intensity

H,S,I components.

color space

The colourisation team spent 18 months developing software for colouring the frames, called "Effects Plus", which was designed to accept only those colours whose hue would match the shade of grey present in the original film. This ensured that the colours added were as close to the real colour as possible;[113] the authenticity of the colouring was later verified when a costume used in the film was retrieved from a warehouse, and its colours were found to closely match those in the film. Every shot was finally hand-corrected to perfect the look.[116] The actual colourisation process took a further 10 months to complete.[113] The exact cost of the colourisation is disputed, with a wide variety of estimates ranging from ₹20 million (US$290,000)[117] to ₹50 million,[46][118] or ₹100 million.[76]

https://en.wikipedia.org/wiki/Mughal-e-Azam

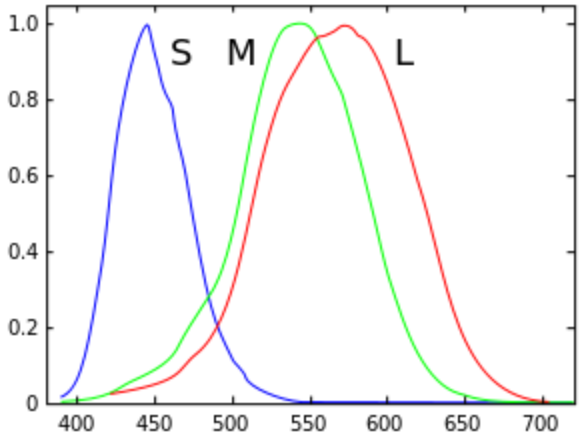
Mayabazar (1957), colorized (2007)

The Film Heritage Foundation announced in March 2015 that they would be restoring *Mayabazar*, along with a few other Indian films from 1931 to 1965, as a part of their restoration projects carried out in India and abroad in accordance with international parameters. **The foundation opposed digital colourisation, stating that they "believe in the original repair as the way the master or the creator had seen it"**

LMS color space

- Used when performing chromatic adaptation (estimating appearance of a sample under a different illuminant)
- Also useful in the study of color blindness, when one or more cone types are defective.

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.7328 & 0.4296 & -0.1624 \\ -0.7036 & 1.6975 & 0.0061 \\ 0.0030 & 0.0136 & 0.9834 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$



White Balancing

Courtesy: wahlmanphotography.com

Courtesy: wallcreations.com.au

White Balancing

White Balancing

White Balancing

```
im = double(imread('lighthouse.jpg'));

RGBw = [246 169 87];

im1(:,:,1) = im(:,:,1)*255/RGBw(1);
im1(:,:,2) = im(:,:,2)*255/RGBw(2);
im1(:,:,3) = im(:,:,3)*255/RGBw(3);
```

Von Kries Method

- Scaling operation is performed in LMS space

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 1/L'_w & 0 & 0 \\ 0 & 1/M'_w & 0 \\ 0 & 0 & 1/S'_w \end{bmatrix} \begin{bmatrix} L' \\ M' \\ S' \end{bmatrix}$$

White Balancing

Incandescent lighting

Fluorescent lighting

Sunlight

Camera Flash

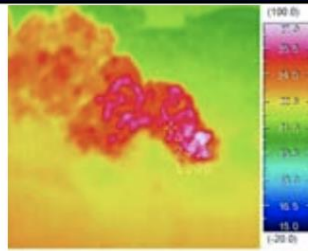
Cloudy

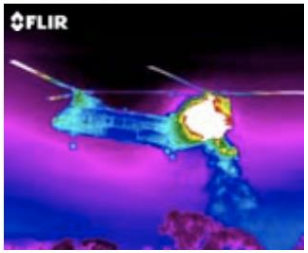
Shadow

courtesy: seventhgrove.com

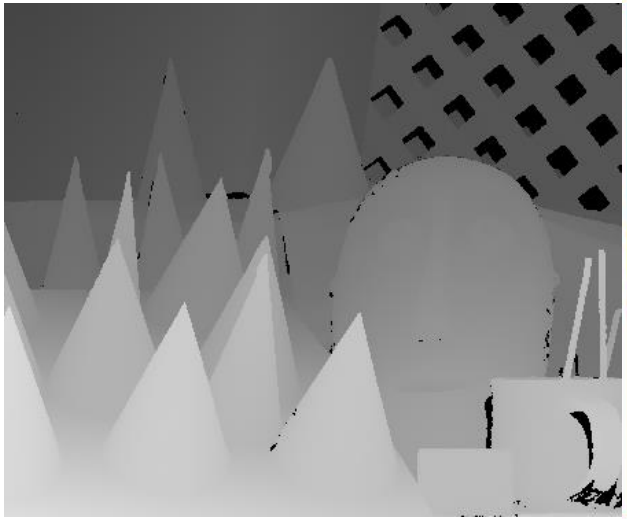
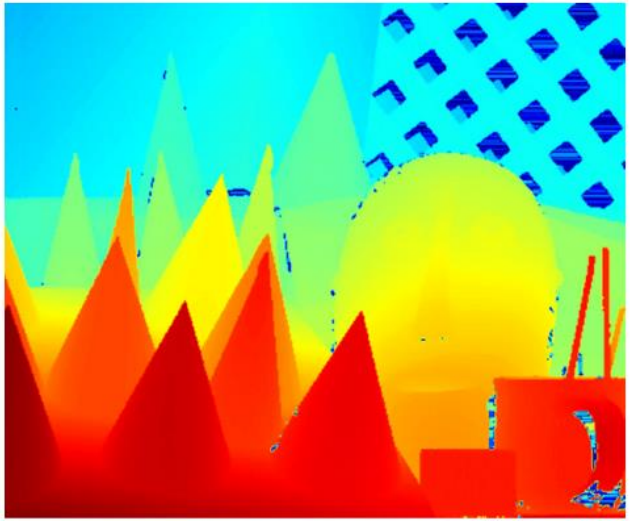
Pseudocolor Image Processing

- Pseudocolor (also called false color) image processing consists of assigning colors to grey values based on a specific criterion.
- The principle use of pseudocolor image processing is for human visualisation.
  - Humans can discern between thousands of color shades and intensities, compared to only about two dozen or so shades of grey.



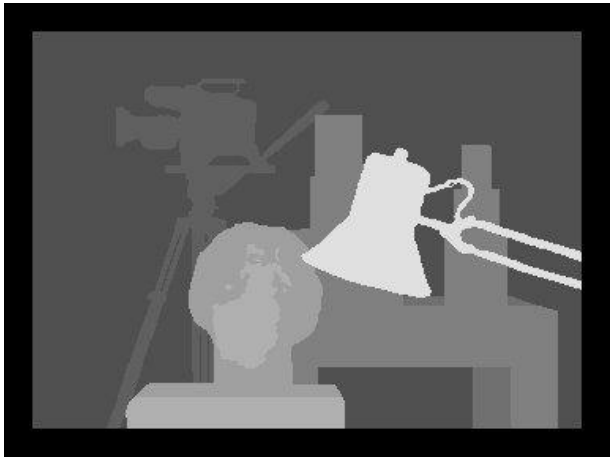


Pseudo color Image Processing

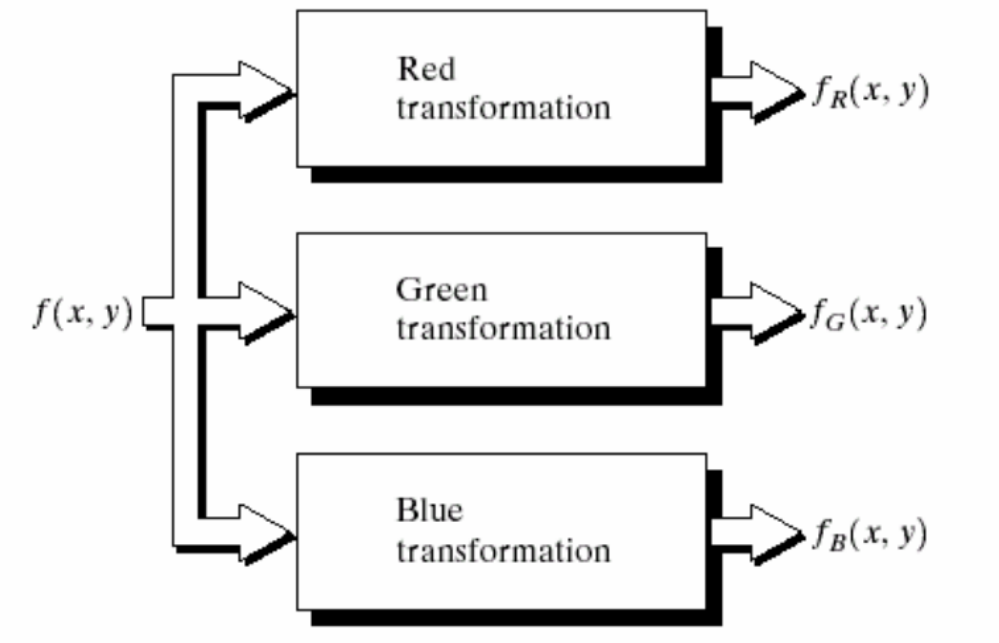
courtesy: middlebury dataset

Pseudo color Image Processing



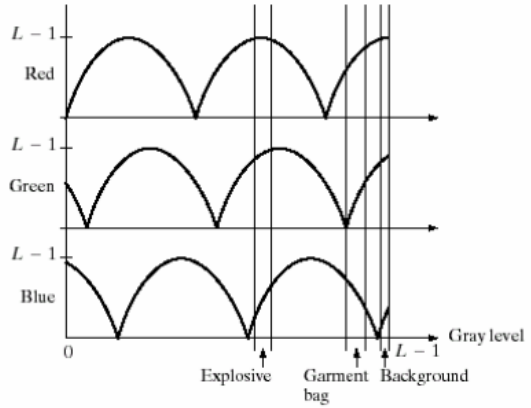
courtesy: middlebury dataset

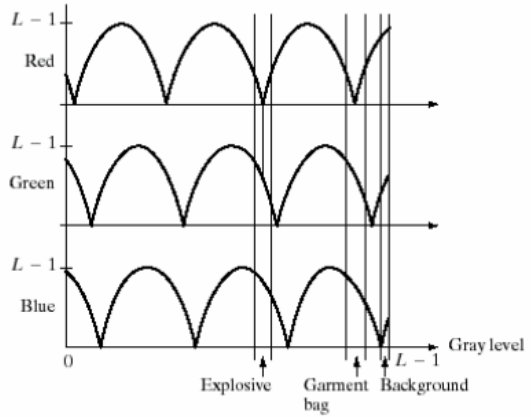
Pseudo color Image Processing (Transformations)

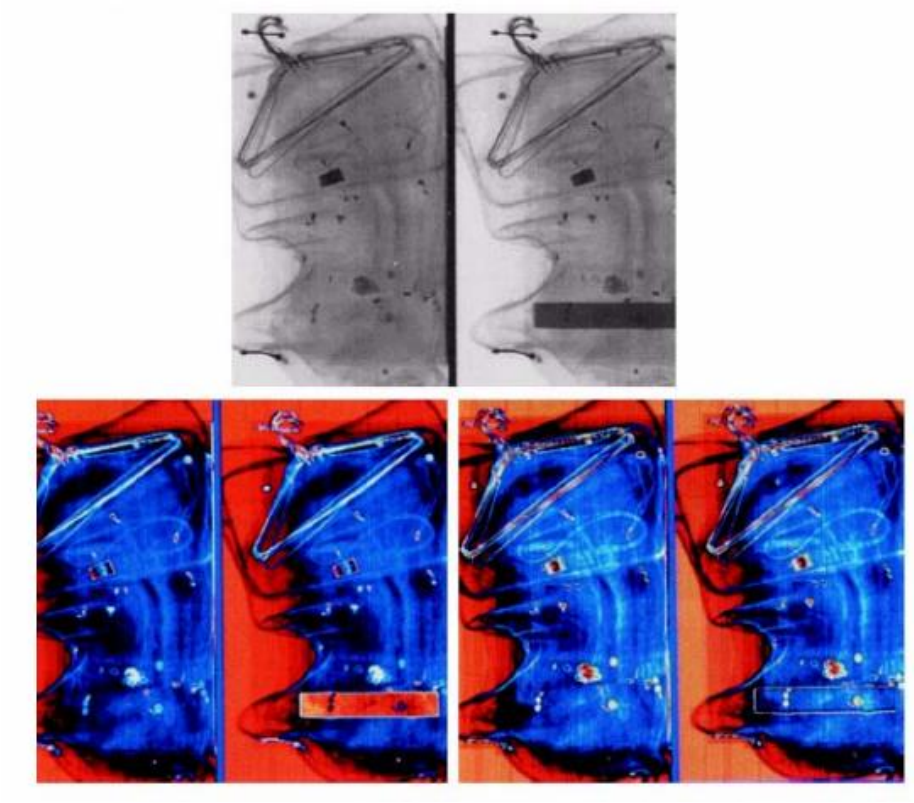


courtesy: Gonzalez and Woods

Pseudo color Image Processing (Transformations)

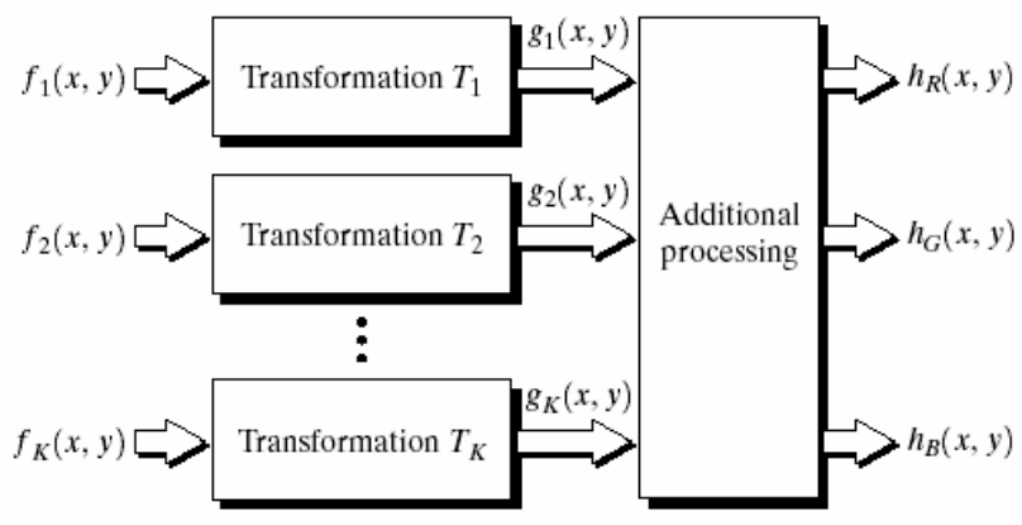






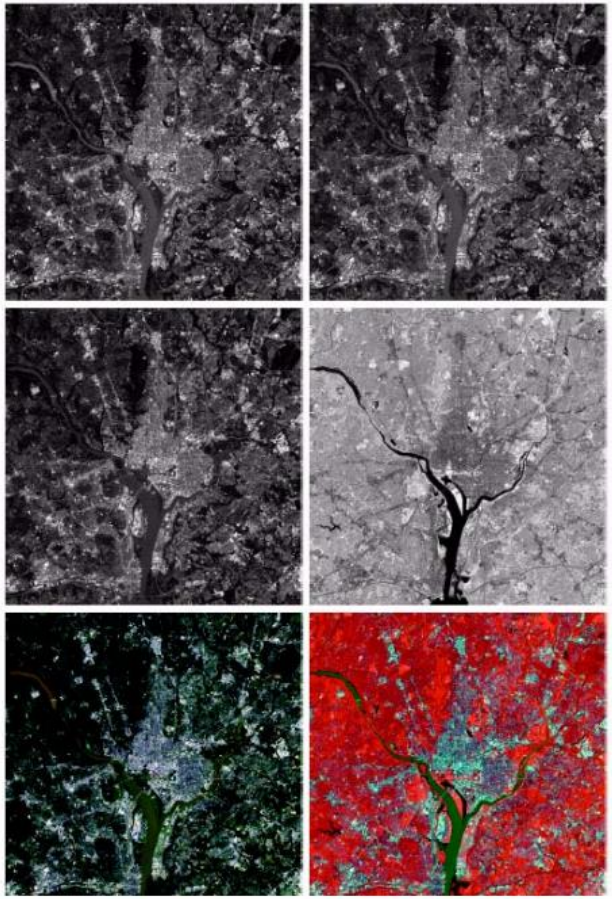
courtesy: Gonzalez and Woods

Pseudo color Image Processing (Multi Spectral)



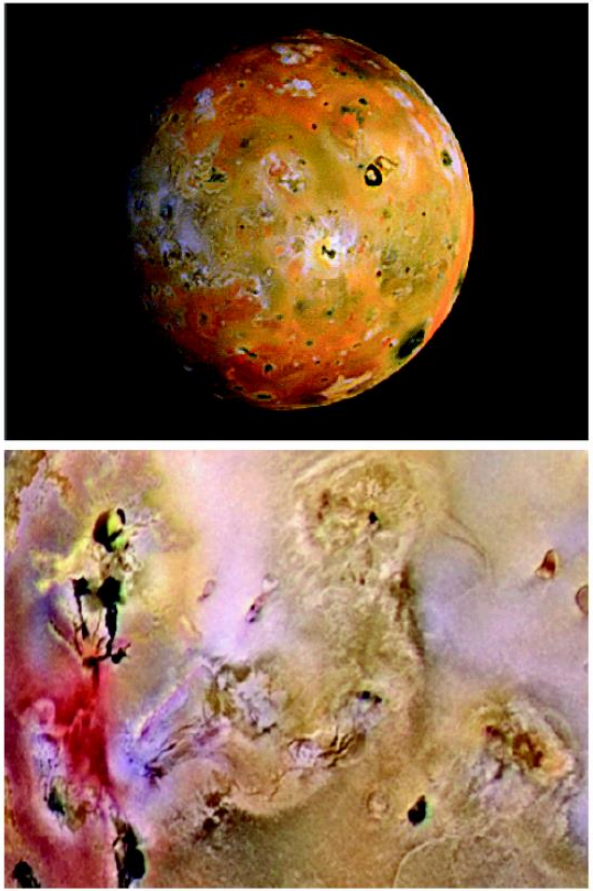
courtesy: Gonzalez and Woods

Pseudo color Image Processing (Multi Spectral)



courtesy: Gonzalez and Woods

Pseudo color Image Processing (Multi Spectral)



courtesy: Gonzalez and Woods

RGBA space

- A (alpha) for transparency (important in image editing)

$$I_{out} = \alpha I_{foreground} + (1 - \alpha) I_{background}$$

Trending applications: Image enhancement in RGB

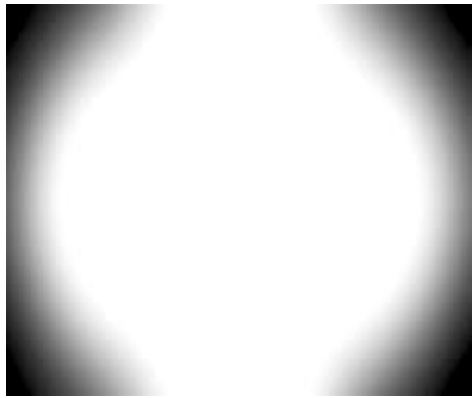
Example: Vintage effect

Example: Vintage effect

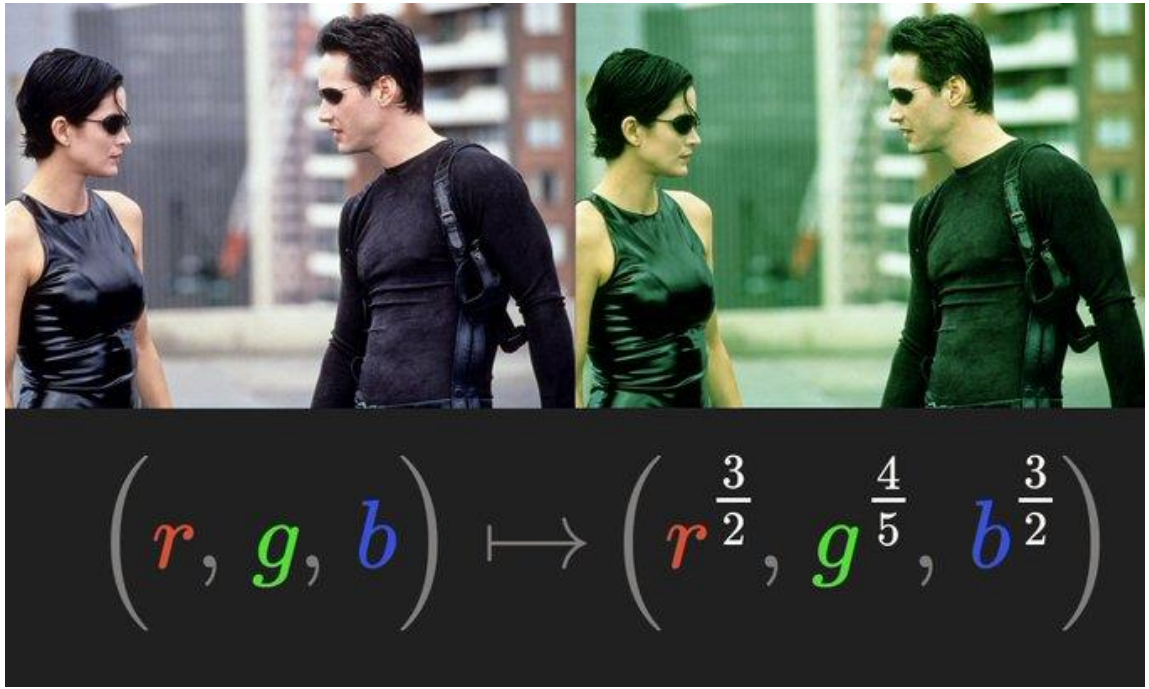
```
im = double(imread('bike.jpg'));

% Extract each colour plane
R = im(:,:,1); % Red
G = im(:,:,2); % Green
B = im(:,:,3); % Blue

% Create sepia tones for each channel
%(these number can be edited to create different styles)
outR= (R * .293) + (G *.769) + (B * .210);
outG = (R * .249) + (G *.686) + (B * .188);
outB = (R * .172) + (G *.534) + (B * .151);
```

Example: Vignetting effect

```
texture = imread(texture_path);
texture = imresize(texture,[size(out,1) size(out,2)]);
texture = double(rgb2gray(texture))/255;

out1(:,:,1) = double(out(:,:,1)) .* double(texture);
out1(:,:,2) = double(out(:,:,2)) .* double(texture);
out1(:,:,3) = double(out(:,:,3)) .* double(texture);
```


‘Matrix’ effect



https://pbs.twimg.com/media/E_7qDWVVIAMTpgT?format=jpg&name=small

E.g. Contrast enhancement in color images:

- **Basic (popular) approach:**
  - ⇒ human eye - 5 more sensitive to brightness contrast then color contrast
  - ⇒ can achieve good contrast enhancement on brightness component alone!
  - ⇒ typically:

Tone and Color Corrections

$$L^* = 116h\left(\frac{Y}{Y_W}\right) - 16$$

$$a^* = 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right]$$

$$b^* = 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{X_W}\right)\right]$$

$$h(q) = \begin{cases} \sqrt[3]{q}\,0.5 & q > 0.008856 \\ 7.787q + 16/116 & q \le 0.008856 \end{cases}$$

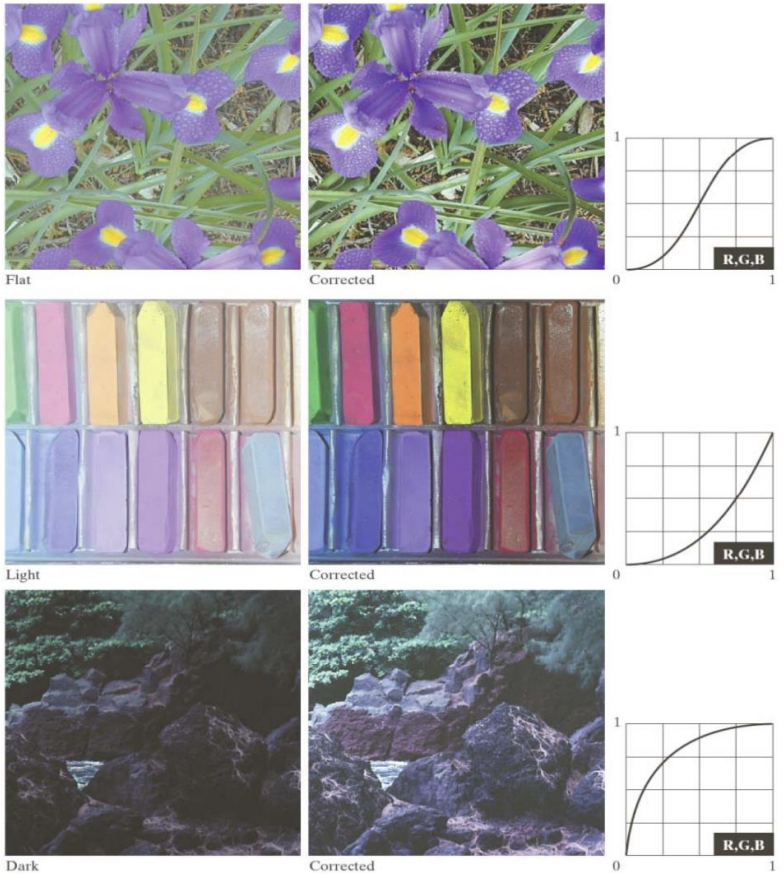
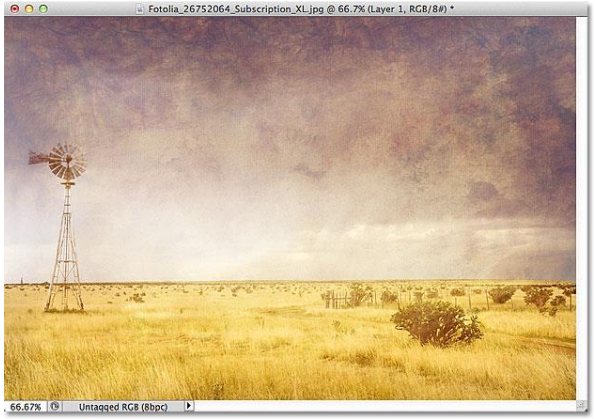


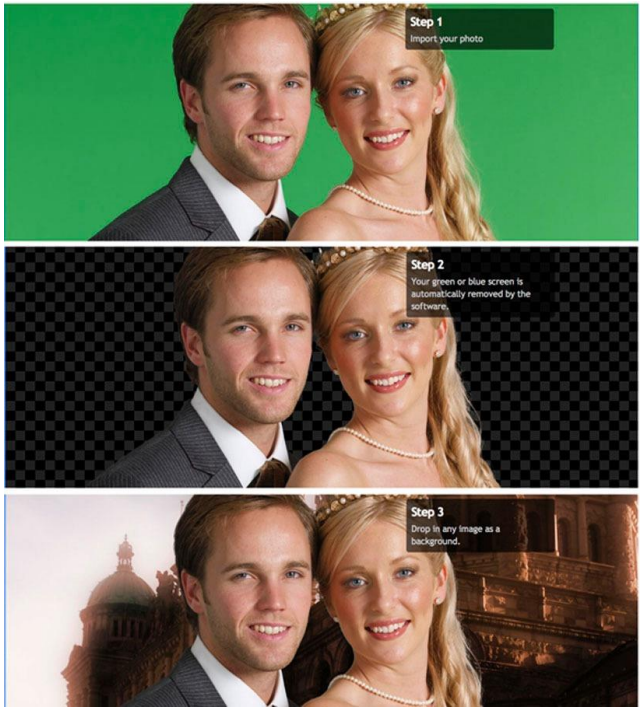
FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not always alter the image hues significantly.

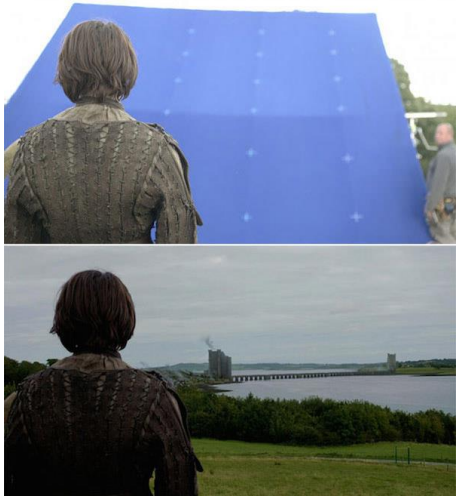
Other such image effects

1. Change the transformation matrix, to suit the desired color tones
2. Choose or design different textures and blend them with original image
3. Repeat 1 and 2 in innovative ways



Chroma Keying

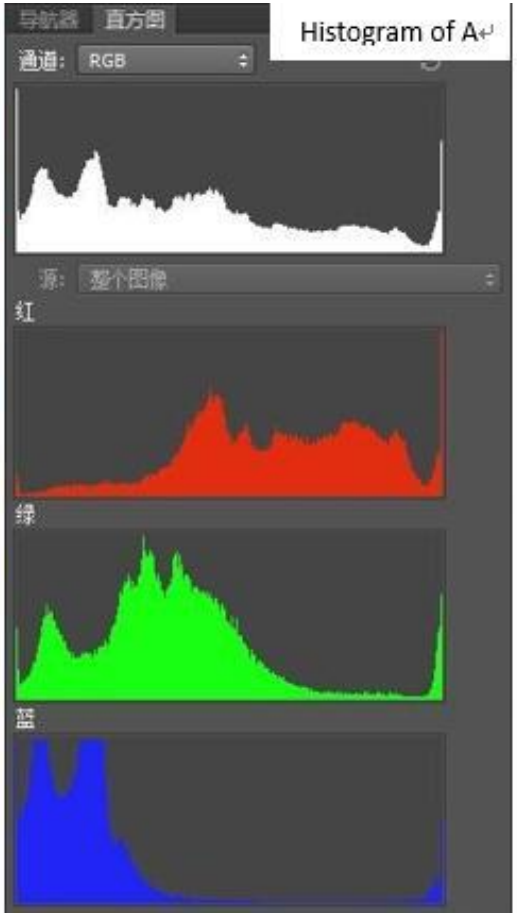




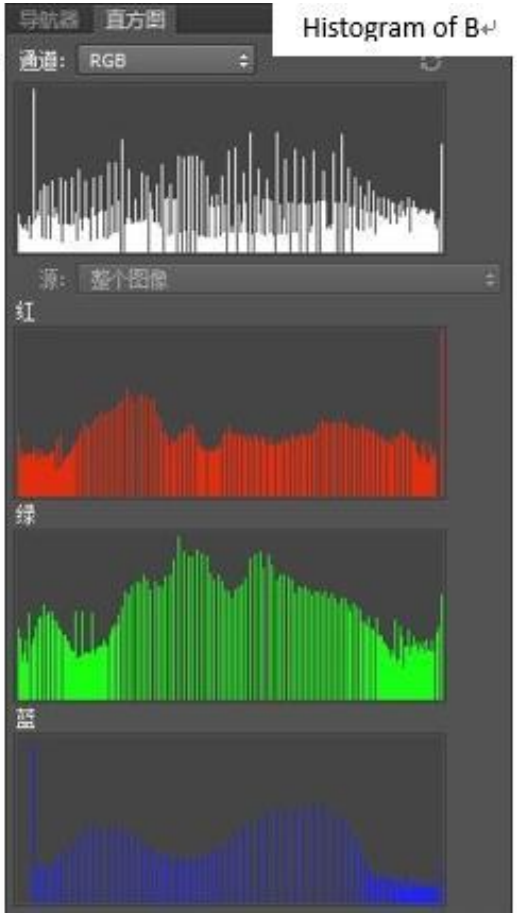
Histogram equalization (per-channel)

A: original rgb image

B: equalized by independent channel



Histogram of A

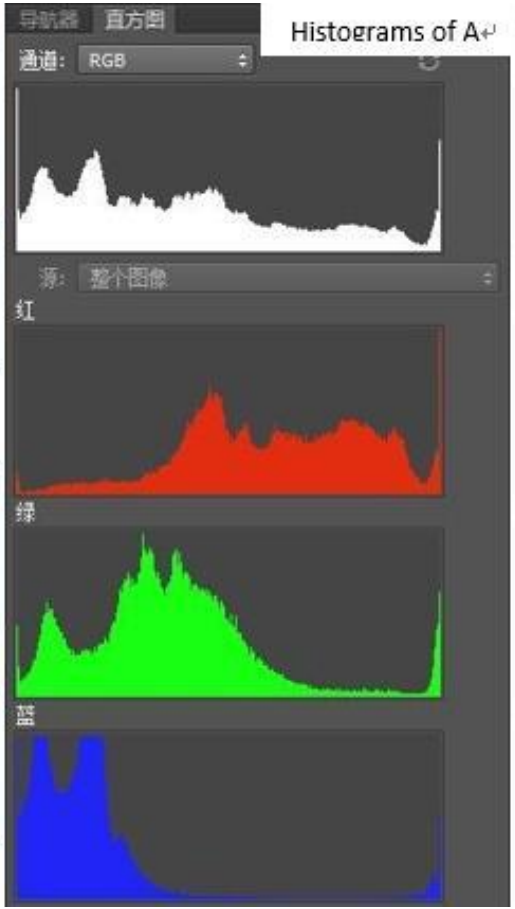


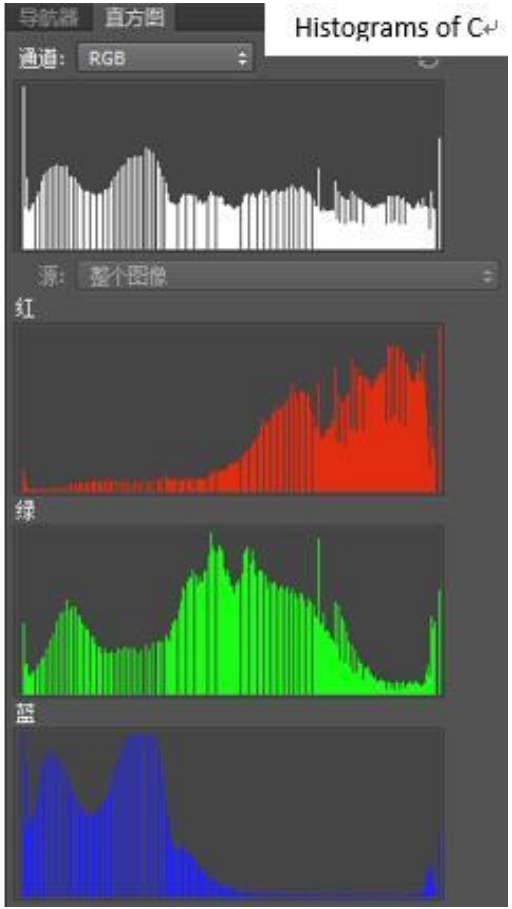
Histogram of B

Histogram equalization (ref histogram = average of R,G,B)

A: original rgb image

C: equalized by average histogram



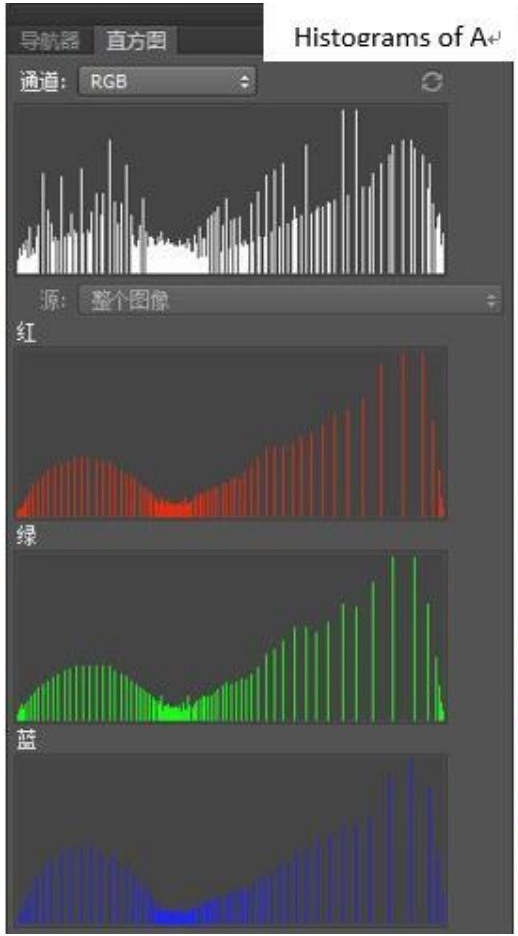


Histogram equalization (ref histogram = I of HSI)

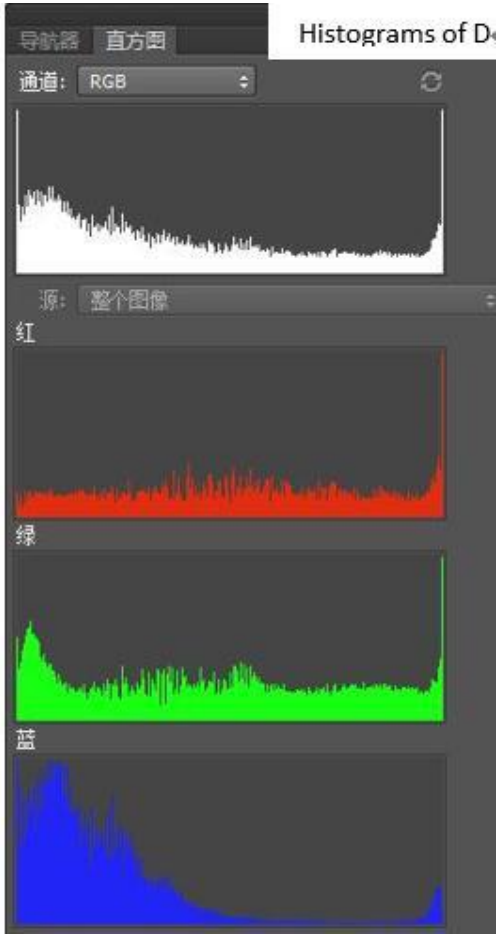
A: original rgb image

D: Intensity component equalization

Luminance histogramsof D



Histograms of A



Histograms of D

References

- Gonzalez & Woods, Chapter-6
  - 6.1, 6.2, 6.3.2, 6.5
- http://www.inf.u-szeged.hu/ssip/2008/presentations2/Gordan_LectureSSIP08.pdf
- https://hypjudy.github.io/2017/03/19/dip-histogram-equalization/
- “Colorimetry”, Ohta and Robertson, John Wiley and Sons Ltd