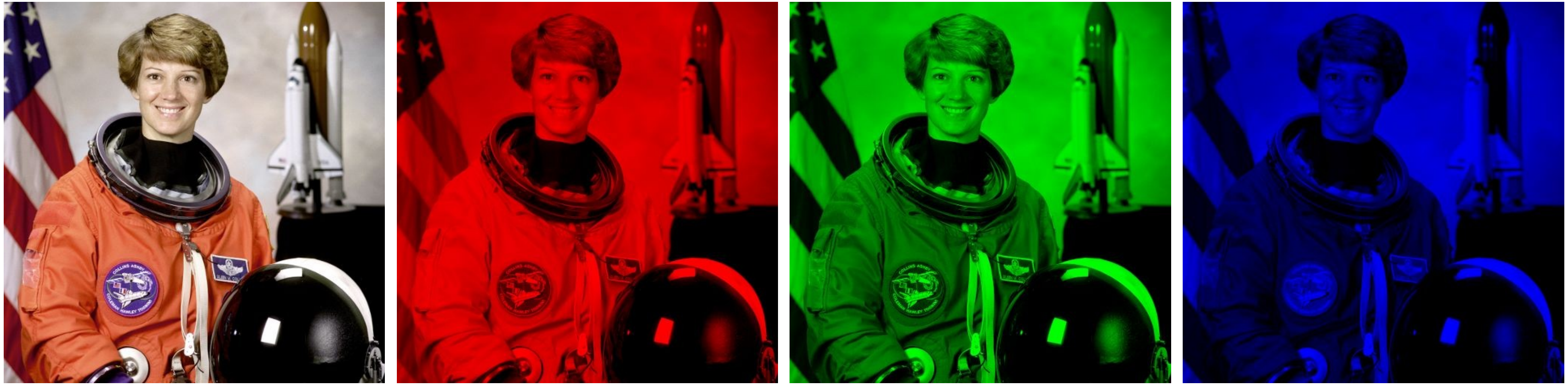
# Colour

Deepayan Bhowmik

# Colour



# Colour

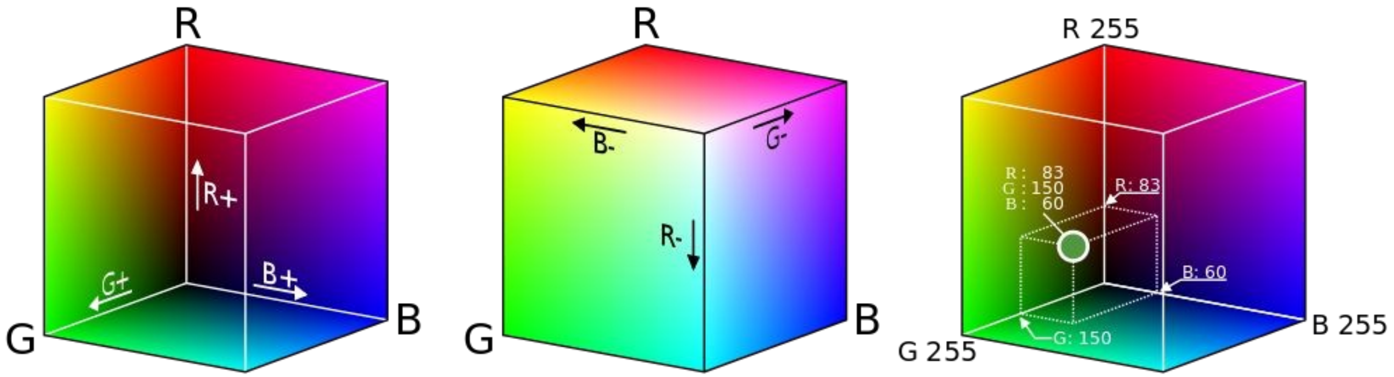
RGB colour representation:

- Red, Green and Blue light.
- Each (x,y) pixel = {r,g,b} integer vector.
- Colour image = 3-channel.
- Image: (x,y,i) for i={0,1,2}.

# Colour Spaces: RGB

Colours represented as R,G,B intensities:

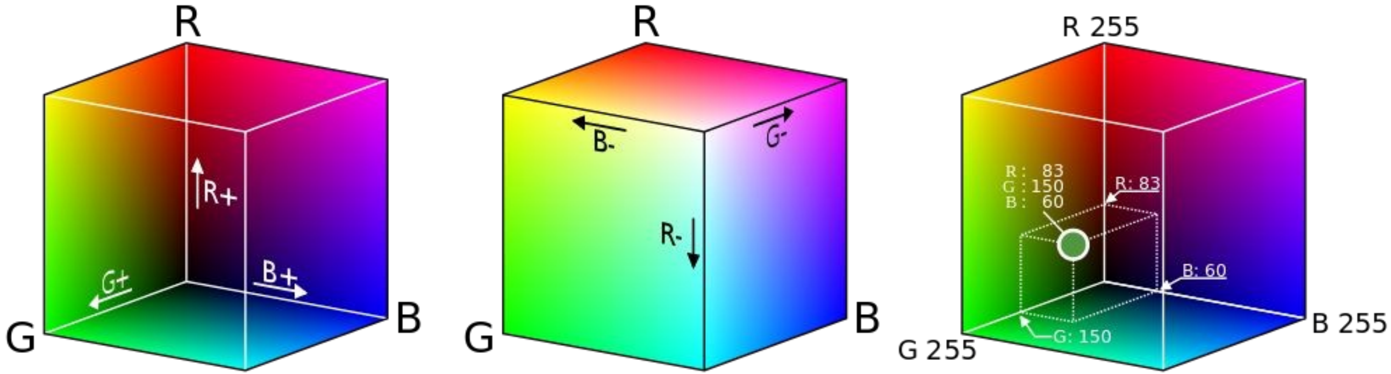
- 3D colour space (cube) with axes R, G and B.
- Each axis 0 → 255 (8-bit colour).
- Black = (0,0,0) (origin); White = (255,255,255) (opposite corner).




Key problem for image processing tasks: RGB is perceptually non-linear.

# Colour Spaces: RGB

What do we mean by perceptually non-linear ?



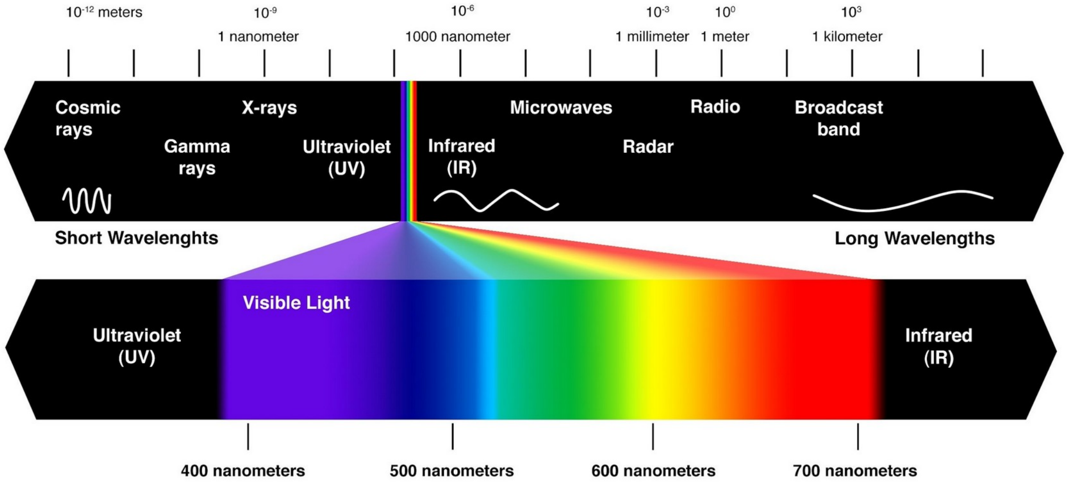

- Starting at white and subtracting the blue component produces ..... yellow!
- Starting at white and subtracting the red component produces ..... blue!
- Starting at red and adding the blue component produces ..... pink!

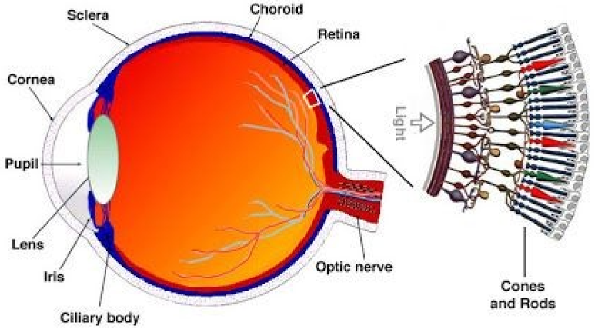
RGB representation does not follow a natural representation of colours (as humans would reason about them).

# Colour Spaces: why use RGB?

Based on human perception of visible spectrum

- Human eye has R, G and B receptors
  - Limited, non-uniform absorbance.
- Computer science - simplified and standardised RGB model.



# Colour Spaces: HSV

Colour represented in H,S,V parameterized space:

- Commonly modelled as a cone.
- “Perceptual colour space”
  - more closely follows how we reason about colour.

H (Hue) = dominant wavelength of colour:

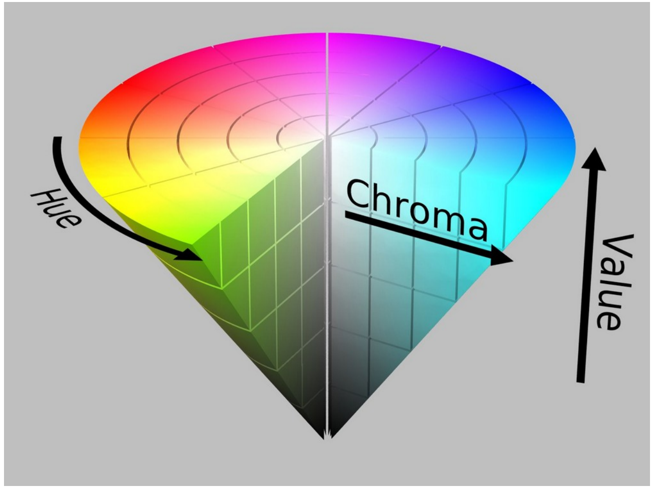
- Colour type {e.g. red, blue, green...}.

S (Saturation) = amount of Hue present:

- “Vibrancy” or purity of colour.

V (Value) = brightness of colour:

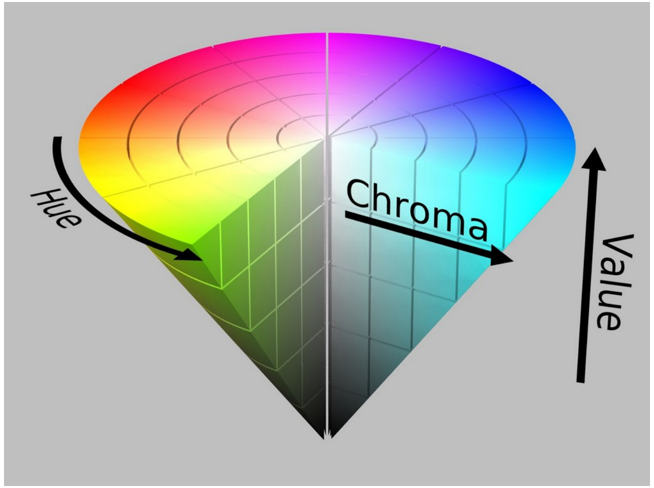
- Brightness of the colour.



# Colour Spaces: HSV

HSV Component / Channel Ranges:

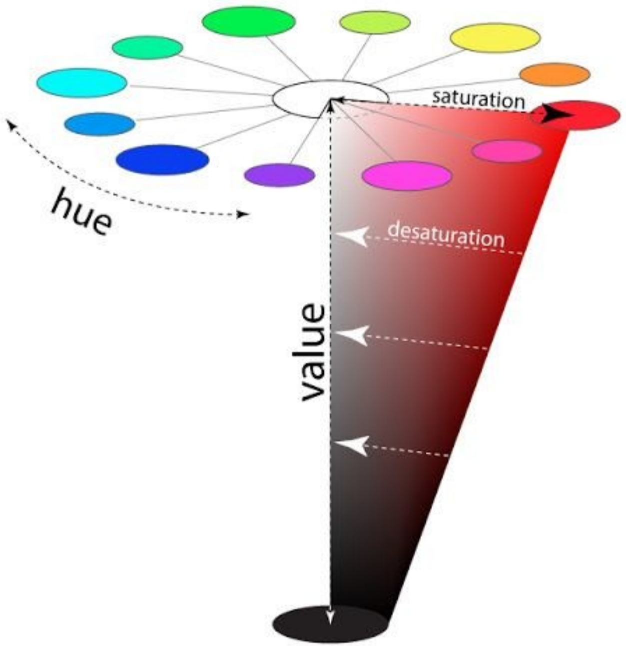
- Hue = 0 → 360° (0-1 in some imp.).
- Saturation = 0 → 1:
  - E.g. for Hue ≈ blue.
  - 0.5 = sky colour; 1.0 = primary blue.
- Value = 0 → 1 (amount of light):
  - E.g. 0 = black, 1 = bright.



# Colour Spaces: HSV

Bright colours (value = 1) with decreasing values of saturation for all values of Hue.

Pure colours (saturation = 1) with decreasing values of “value” (brightness) for all values of Hue.



# Colour Spaces: HSV

*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')

Δ = Cmax - Cmin

Hue calculation:

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

Saturation calculation:

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

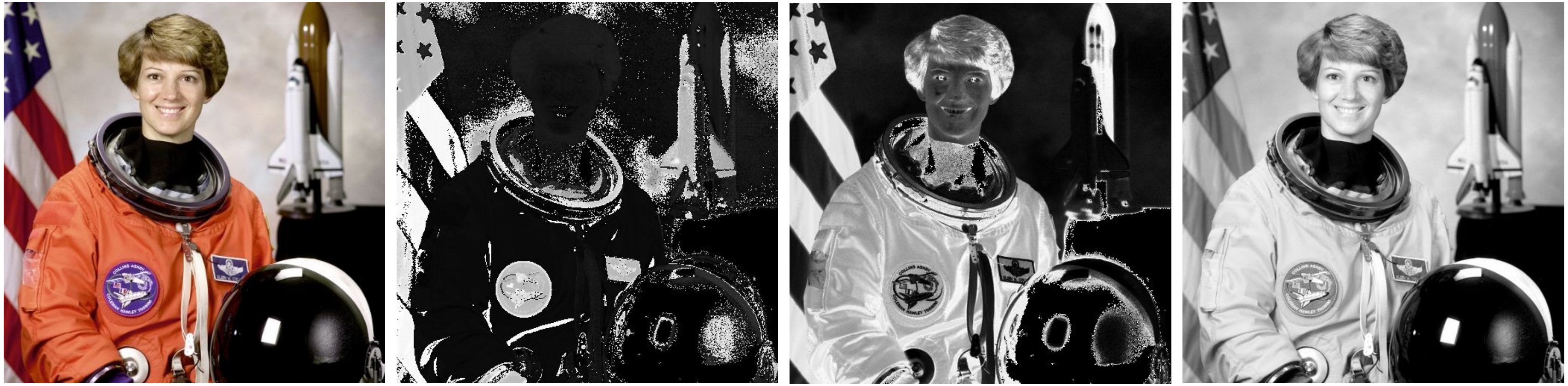
Value calculation: V = Cmax

$$h_i = \left\lfloor \frac{h}{60} \right\rfloor \mod 6$$

$$f = \frac{h}{60} - h_i$$
$$p = v \times (1 - s)$$
$$q = v \times (1 - f \times s)$$
$$t = v \times (1 - (1 - f) \times s)$$

$$(r, g, b) = \begin{cases} (v, t, p), & \text{if } h_i = 0 \\ (q, v, p), & \text{if } h_i = 1 \\ (p, v, t), & \text{if } h_i = 2 \\ (p, q, v), & \text{if } h_i = 3 \\ (t, p, v), & \text{if } h_i = 4 \\ (v, p, q), & \text{if } h_i = 5 \end{cases}$$

# Colour Spaces: HSV



# Colour Spaces: CIE Lab

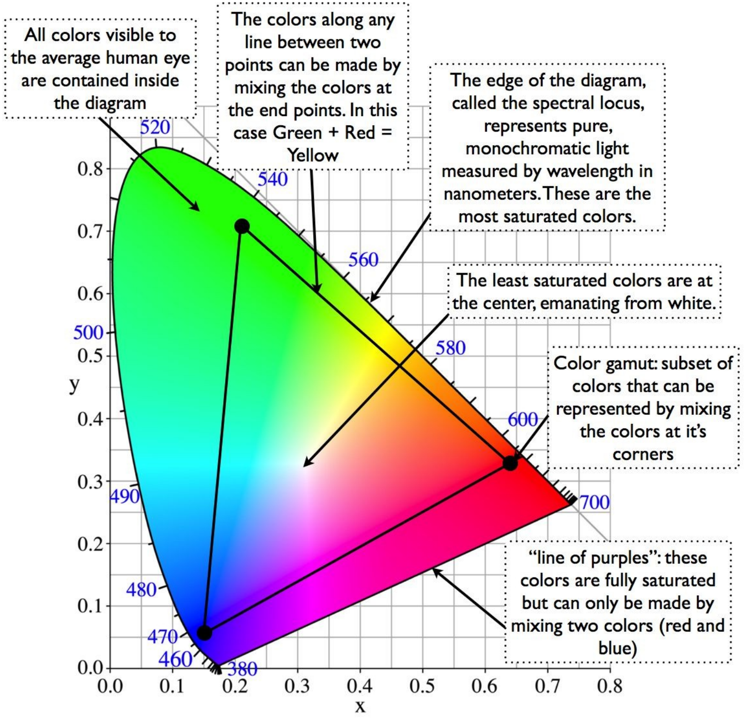
Perceptual Uniformity:

- Change in human perception of colour == change in colour space values.
- Euclidean distance between colour parameterization.

Extends RGB (!):

- L*a*b covers all visible colours, RGB does not.
- L*a*b parameter space can specify imaginary (non-visible colours!).

Device independent.




Anatomy of a CIE Chromaticity Diagram

# Colour Spaces: CIE Lab

# Colour Spaces: CIE Lab

$$L^\star = 116\, f\left(\frac{Y}{Y_\mathrm{n}}\right) - 16$$

$$a^\star = 500\left(f\left(\frac{X}{X_\mathrm{n}}\right) - f\left(\frac{Y}{Y_\mathrm{n}}\right)\right)$$

$$b^\star = 200\left(f\left(\frac{Y}{Y_\mathrm{n}}\right) - f\left(\frac{Z}{Z_\mathrm{n}}\right)\right)$$

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29} & \text{otherwise} \end{cases}$$

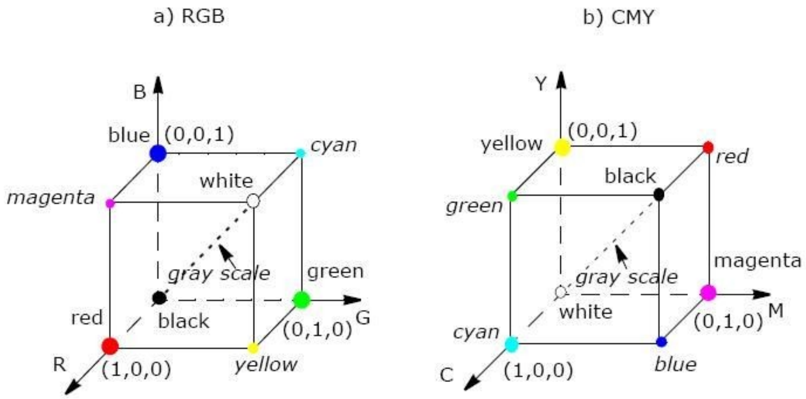
$$\delta = \frac{6}{29}$$

$$X_\mathrm{n} = 95.0489,$$
$$Y_\mathrm{n} = 100,$$
$$Z_\mathrm{n} = 108.8840$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{b_{21}} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{0.176\,97} \begin{bmatrix} 0.490\,00 & 0.310\,00 & 0.200\,00 \\ 0.176\,97 & 0.812\,40 & 0.010\,63 \\ 0.000\,00 & 0.010\,00 & 0.990\,00 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

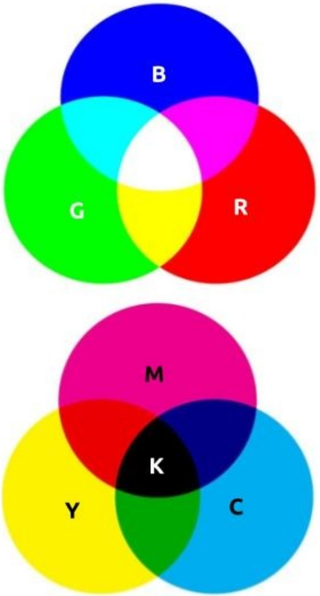
# Colour Spaces: CMY & CMYK

The CMY colour space is also represented by a unit cube. The white is the (0,0,0) and the black is the (1,1,1).

# Colour Spaces: CMY & CMYK

The CMY colour space is also represented by a unit cube. The white is the (0,0,0) and the black is the (1,1,1).




$$R' = \frac{R}{255}$$

$$G' = \frac{R}{255}$$

$$B' = \frac{R}{255}$$

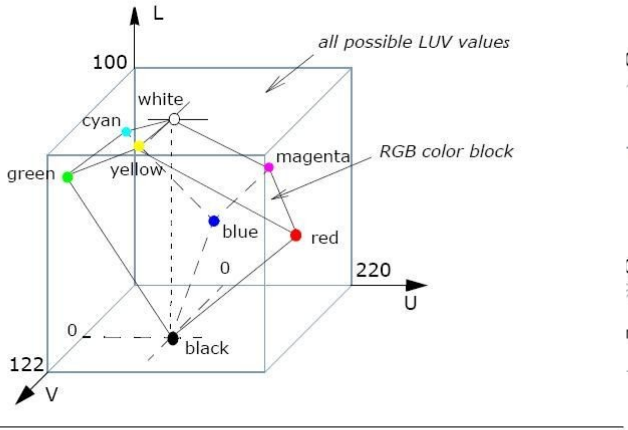
$$K = 1 - max(R', G', B')$$

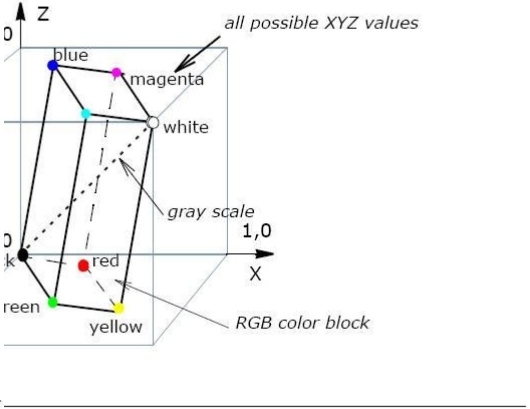
$$C = \frac{1 - R' - K}{1 - K}$$

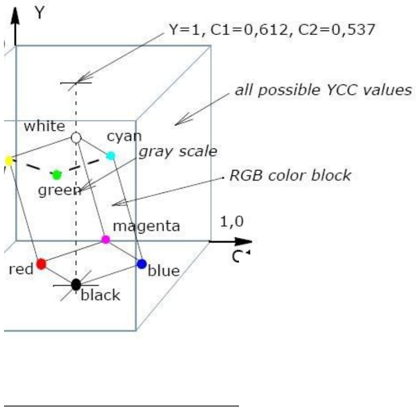
$$M = \frac{1 - G' - K}{1 - K}$$

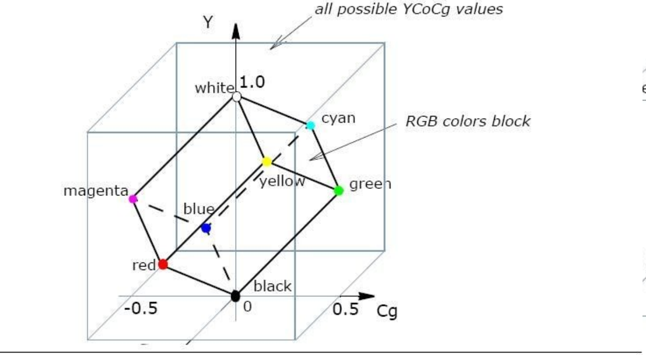
$$Y = \frac{1 - B' - K}{1 - K}$$

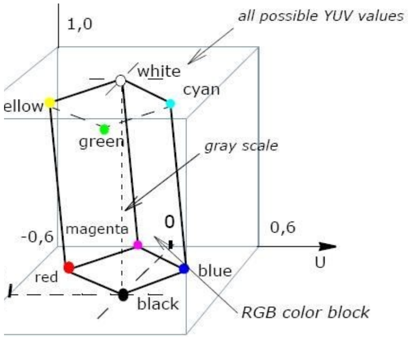
# Colour Spaces: Others

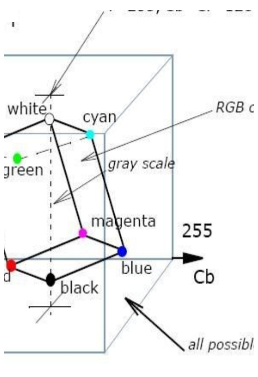


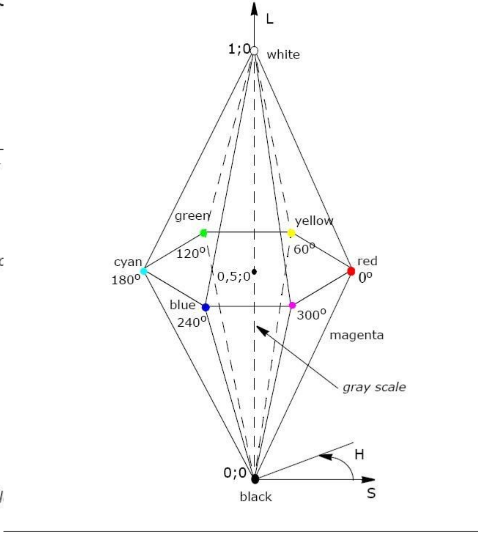



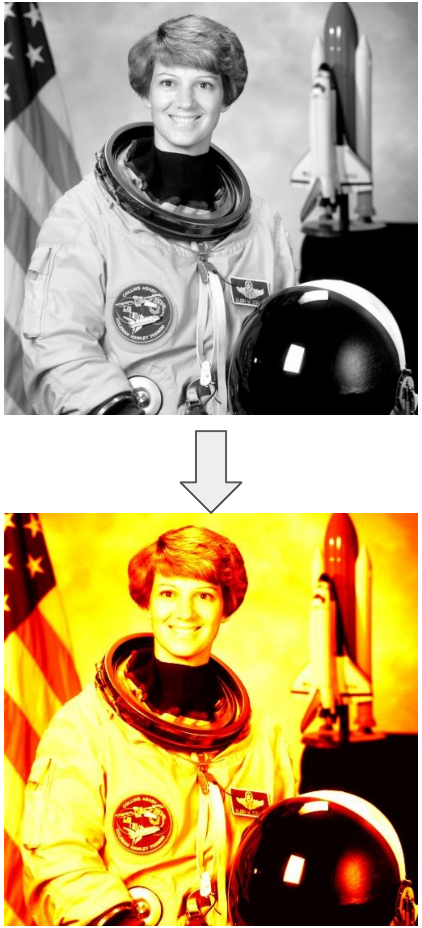
# Colour Mapping

Map scalar value to colour range for display:

- Scalar value = grayscale image value.
- Colour range = purple → yellow.
- Set H channel to grayscale image.
- Set V and S to fixed values.

Colour Look-up Tables (LUT):

- Provide scalar to colour conversion.
- Scalar values = indices into LUT.
- Define colour transfer function.



# Colour Mapping

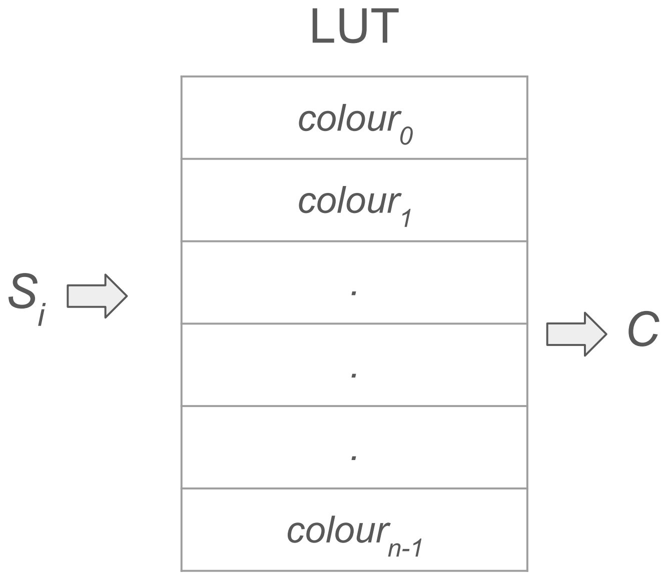
Assume

- Scalar (image) values $S_i$ in range {min→max}.
- $n$ unique colours, $\{colour_0,...,colour_{n-1}\}$ in LUT.

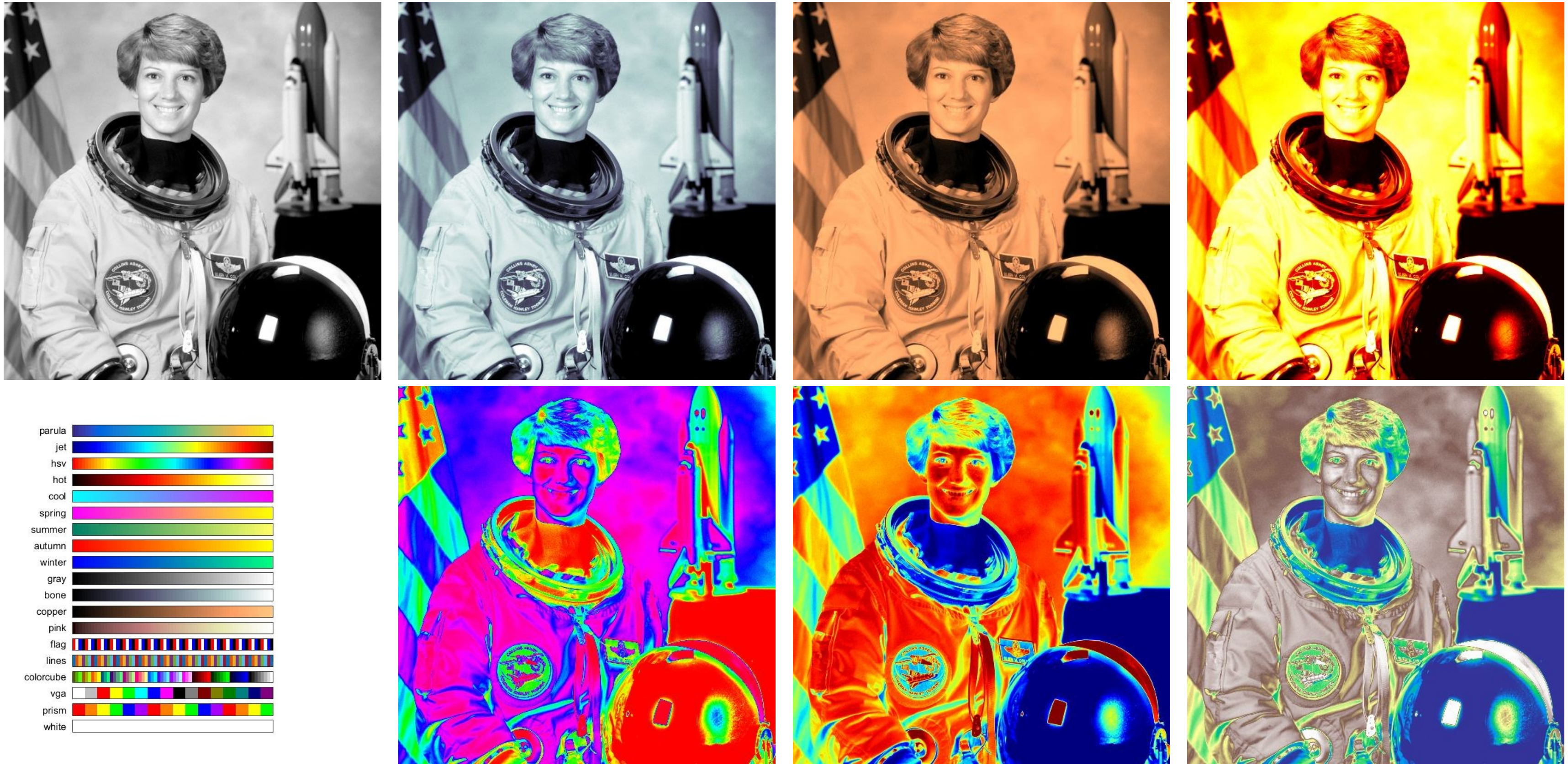
Define mapped colour C:

- If $S_i$ < min then $C = colour_{min}$.
- If $S_i$ > max then $C = colour_{max}$.
- Else $C = colour( (S_i - min) / (max - min) )$.

Quantising values into given colour range.

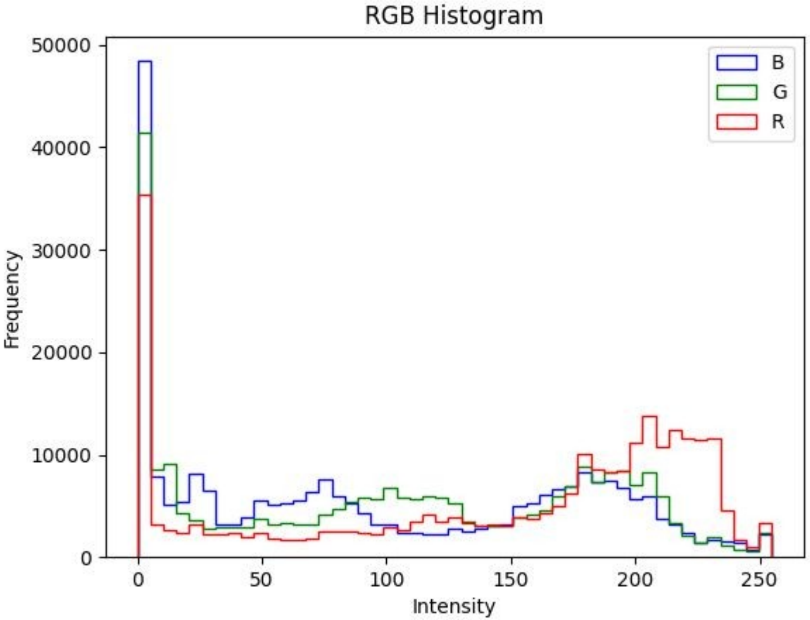


# Colour Mapping



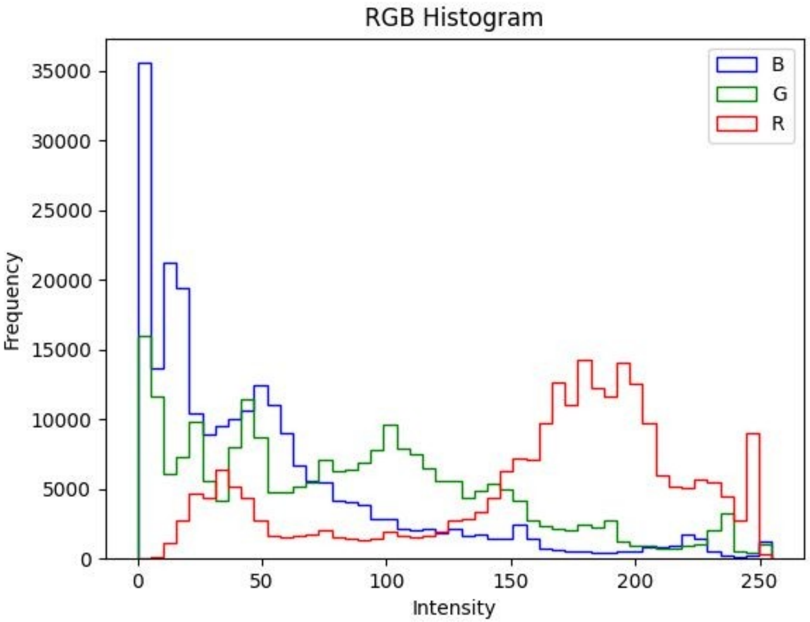
# Colour Spaces: RGB Histogram

Extend to each R, G, B colour channel (separately).



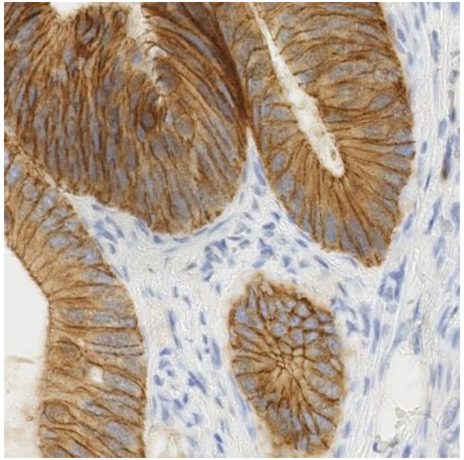
# Colour Spaces: RGB Histogram

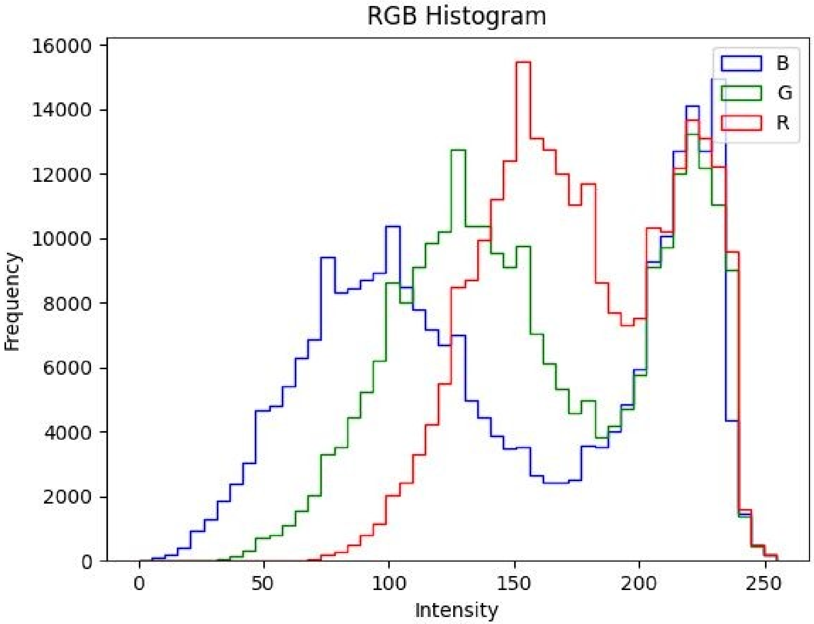
Extend to each R, G, B colour channel (separately).



# Colour Spaces: RGB Histogram

Extend to each R, G, B colour channel (separately).





# Colour Spaces: Histogram

Histogram = statistical distribution of image content:

- No structural representation.
- Different images have very different histograms.
- Similar images have similar histograms.

Histogram Comparison:

- Compare stat. dist. of images.
- Use normalised histograms.
- E.g. chi-squared, Euclidean, … .

# Colour Spaces: Histogram Similarity Distances

Correlation:

$$d(H_1, H_2) = \frac{\sum_{i=1}^{N} H_1'(i) - H_2'(i)}{\sqrt{\sum_{i=1}^{N} [H_1'(i)]^2 \sum_{i=1}^{N} [H_2'(i)]^2}}$$

$$H_k' = \frac{H_k(i) - 1}{N \sum_{i=1}^{N} H_k(i)}$$

Chi-squared:

$$d(H_1, H_2) = \sum_{i=1}^{N} \frac{H_1(i) - H_2(i)}{H_1(i) + H_2(i)}$$

Intersection:

$$d(H_1, H_2) = \sum_{i=1}^{N} min(H_1(i), H_2(i))$$

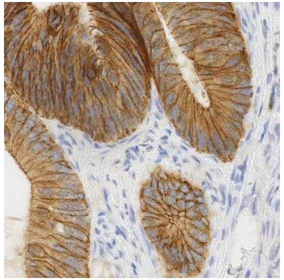
Bhattacharyya:

$$d(H_1, H_2) = \sqrt{1 - \sqrt{\sum_{i=1}^{N} H_1(i) - H_2(i)}}$$

# Colour Spaces: Histogram Similarity Distances

Intersection:

0.649



0.707

0.846

# Colour Spaces: Image Processing

- Contrast Enhancement.
- Histogram Based Transforms.

For processes altering the dynamic range of the image, in order to preserve the relative weights of the colour information in the image, we first convert to a perceptual colour space that isolates the scene illumination in a single channel (e.g. V of HSV or L of L*a*b).

We subsequently perform enhancement on this channel in isolation, add it back to the other channels that contain the image colour (chromatic) information and transform back to RGB.

# Colour Spaces: Histogram Similarity Distances

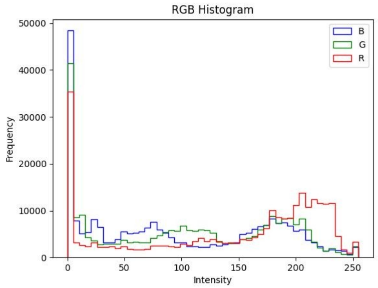
- Logical Arithmetic
- Spatial Filtering.
- Fourier Filtering + Correlation.

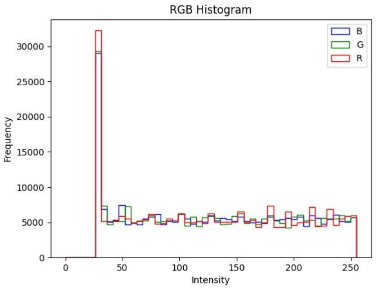
We can just apply these operations to individual colour channels in any colour space representation.

Often we convert to grayscale to reduce computation if colour output is not required.

# Colour Spaces: Histogram Equalisation

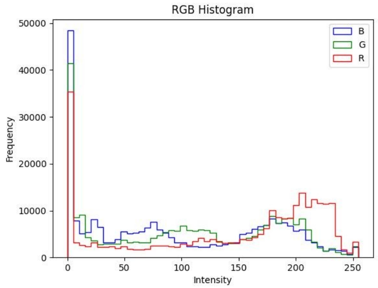
Histogram equalisation on each RGB channel (gives strange results).

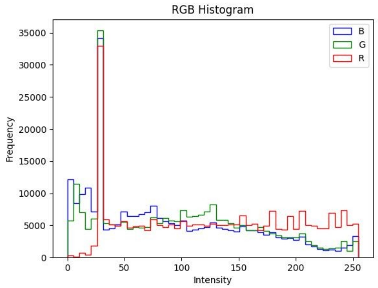


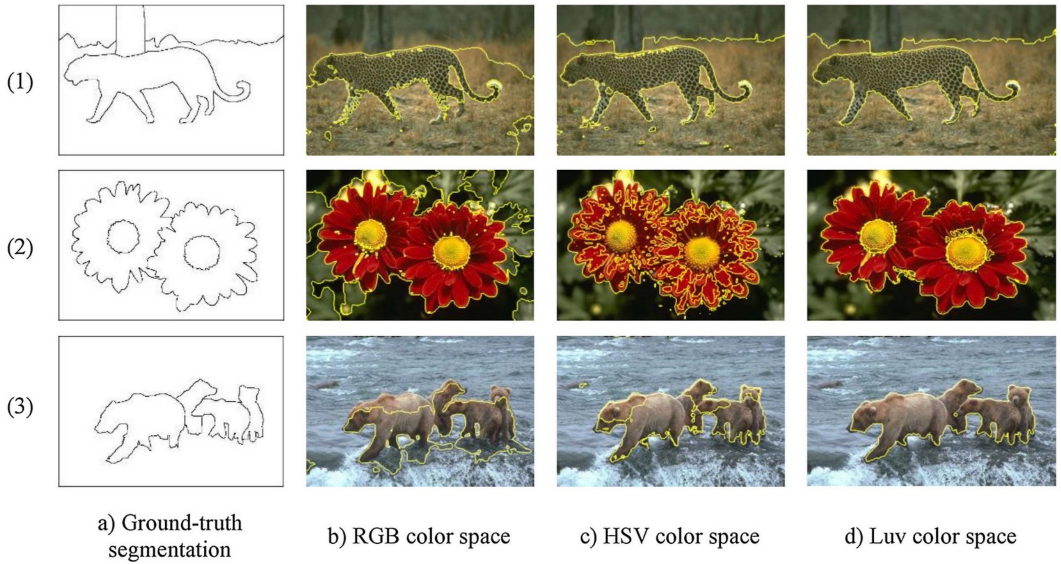
# Colour Spaces: Histogram Equalisation

1. RGB → HSV.
2. Histogram Equalize V channel.
3. HSV → RGB.



# Applications




Akbulut Y, Guo Y, Şengür A, Aslan M (2018) An effective color texture image segmentation algorithm based on hermite transform. Appl Soft Comput 67:494–504