Leow Wee Kheng

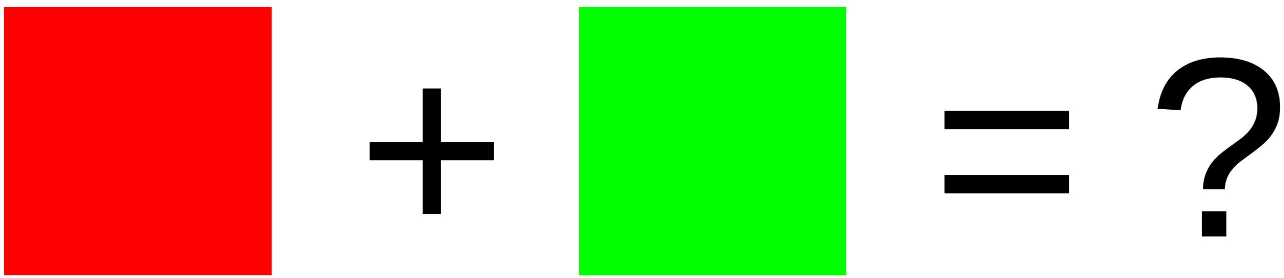
CS4243 Computer Vision and Pattern Recognition

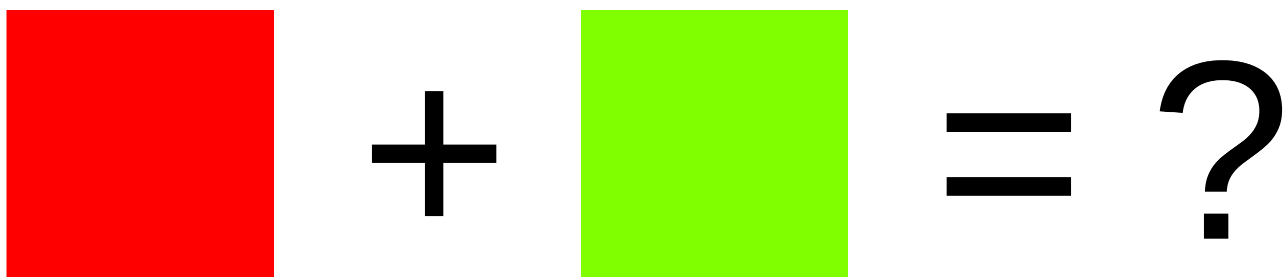
Colour

Colour is everywhere!

You already know enough about colour?

Let’s do a test…

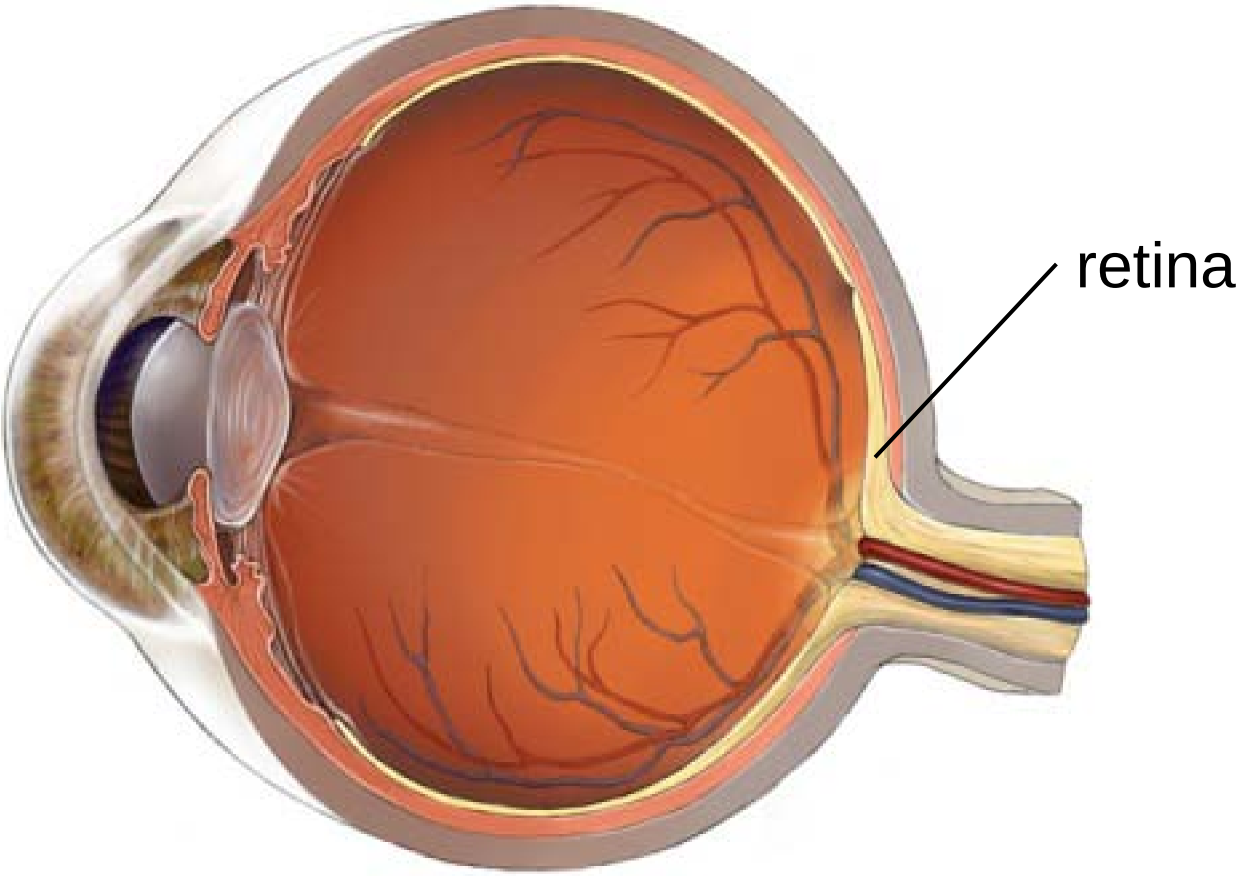




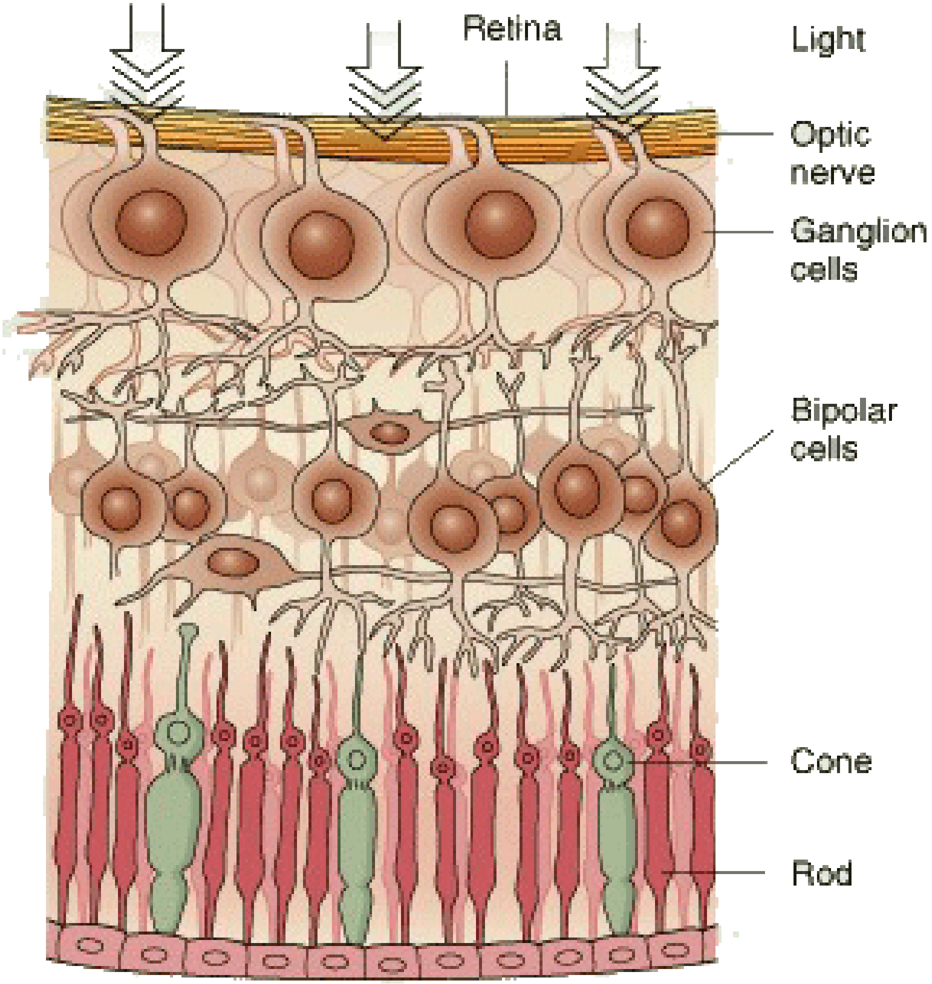
So, there’s a lot more to colour!

Human Colour Perception

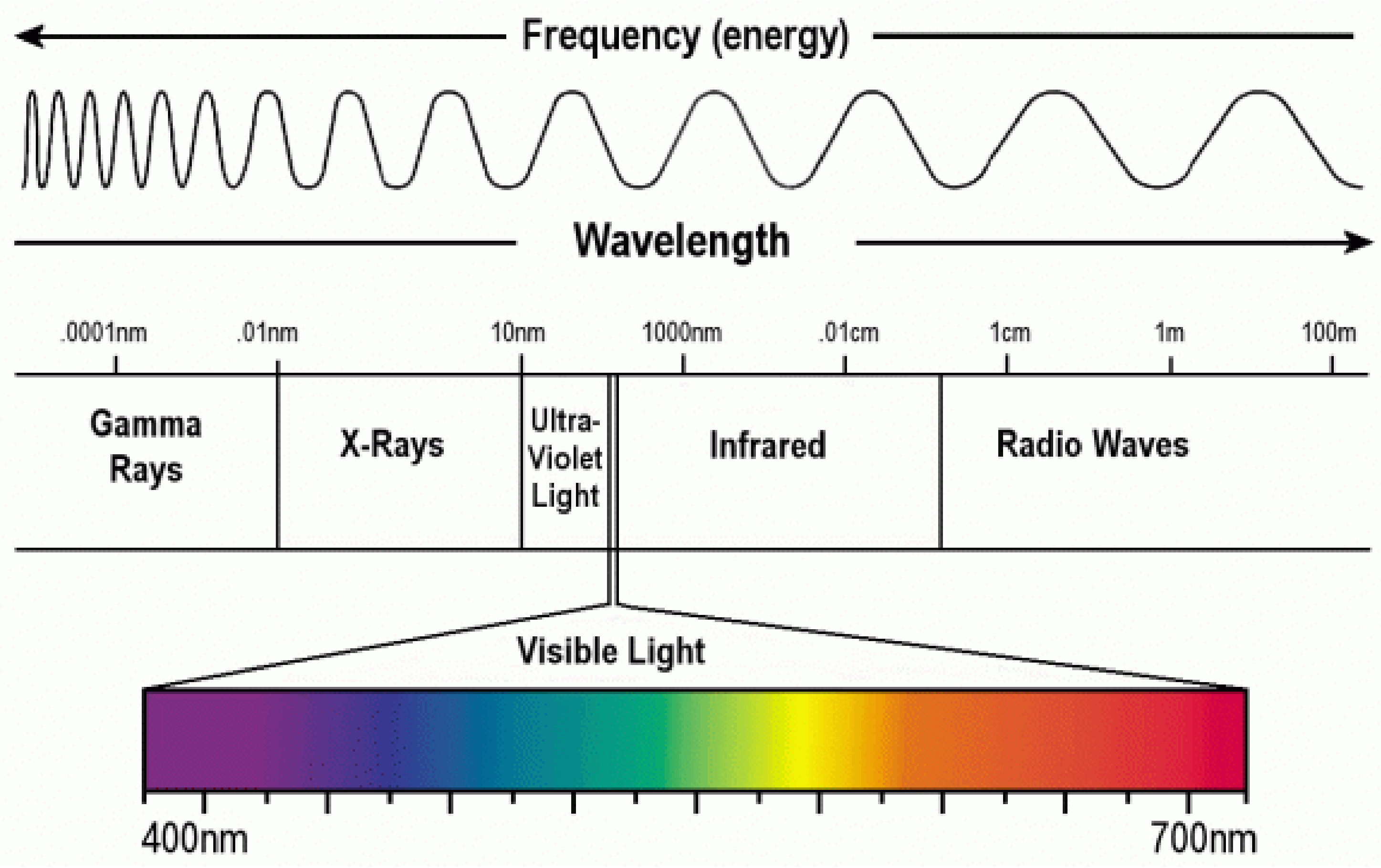
- Human's eye has retina that senses light.



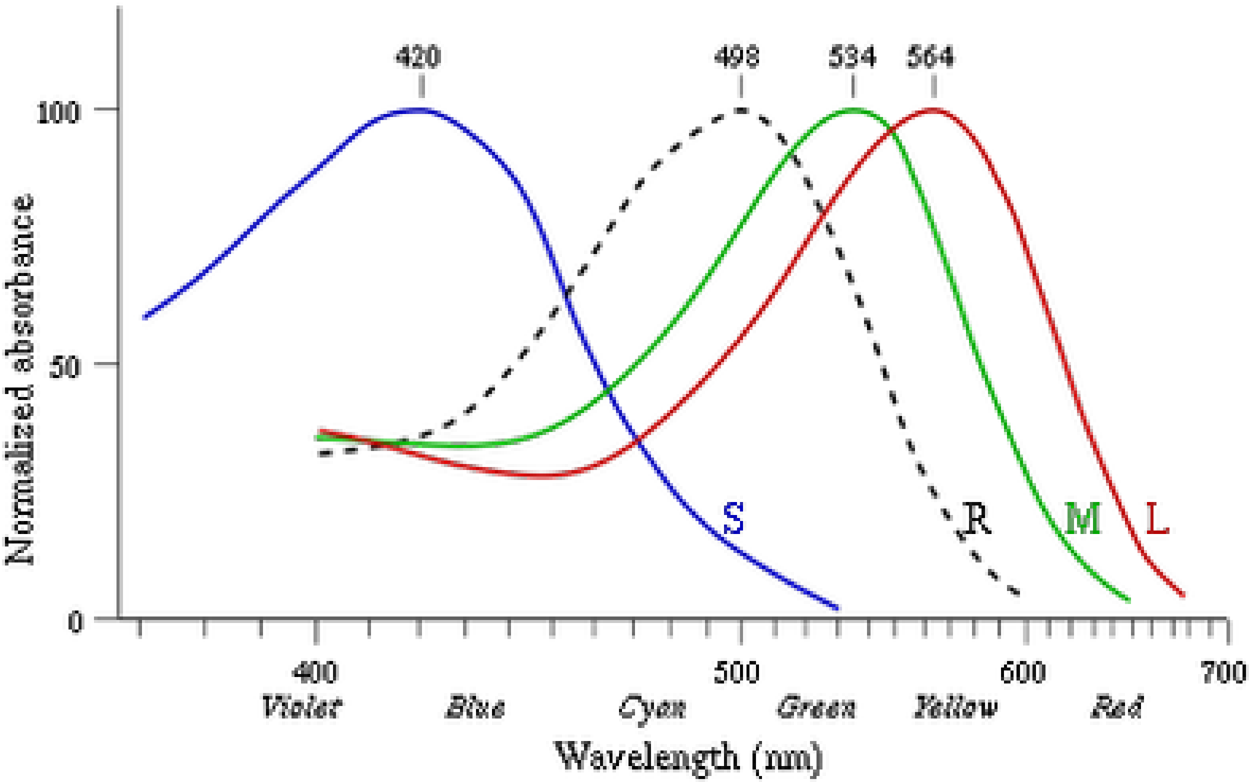
- Human's retina has 2 kinds of light receptors:
  - Rods:
    sensitive to amount of light
  - Cones:
    sensitive to lights of different wavelengths



- Cones are sensitive to various colours
  - Wavelengths from 400nm (violet) to 700nm (red)



- 3 kinds of cones:
  - Short (S): most sensitive to blue
  - Medium (M): most sensitive to green
  - Long (L): most sensitive to red

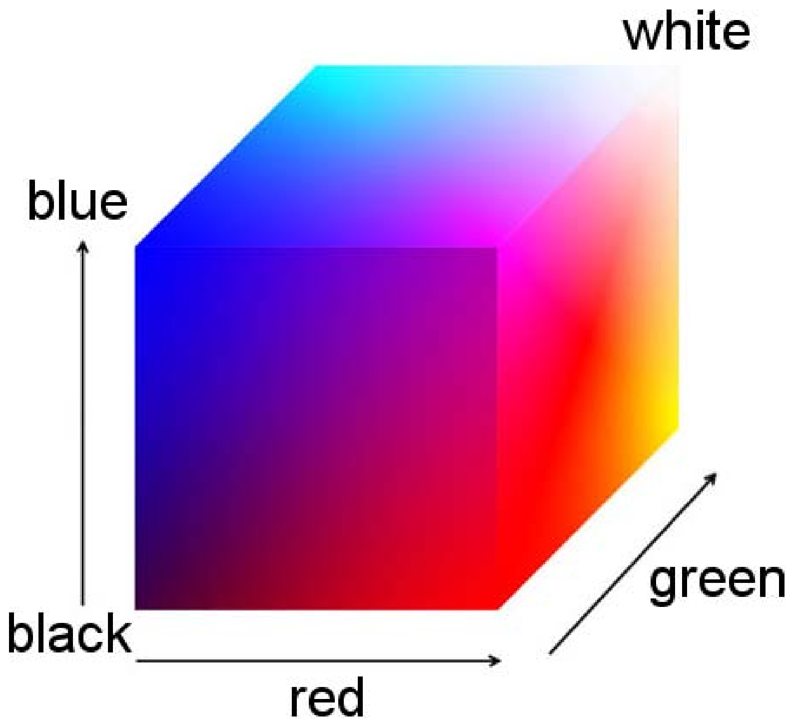


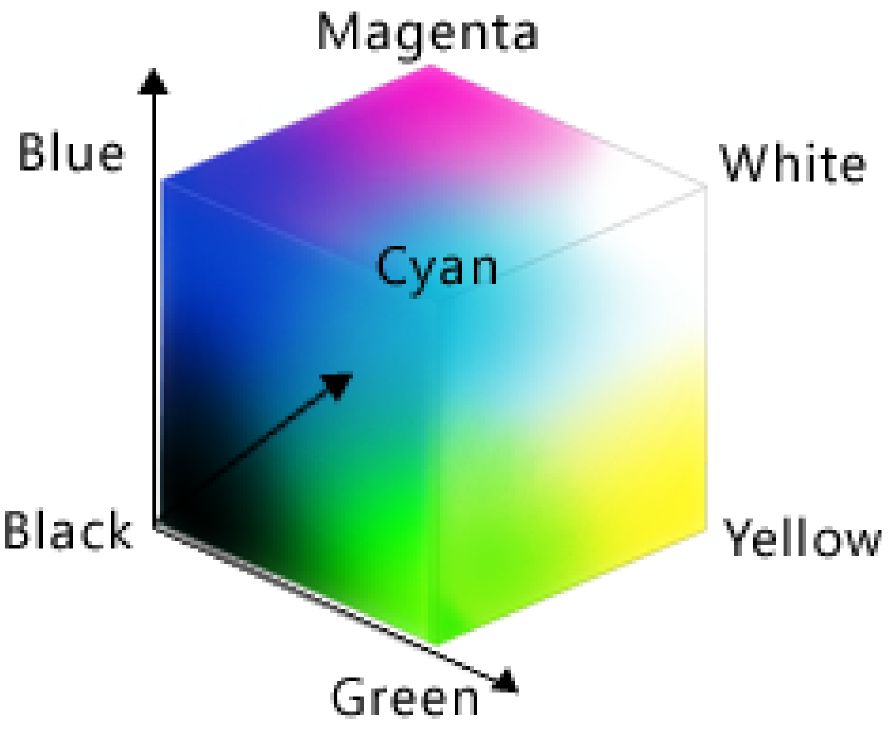
- Cones send signals to brain.
- Brain interprets mixture of signals as colours.
- That’s why colours are coded with 3 values.
- Different coding schemes give different colour spaces.

RGB Colour Space

- Colour is coded with 3 values

(red, green, blue)

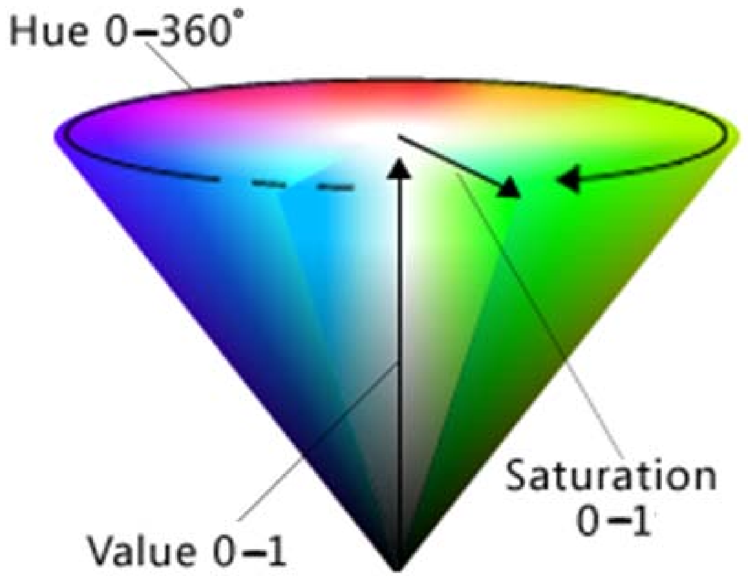


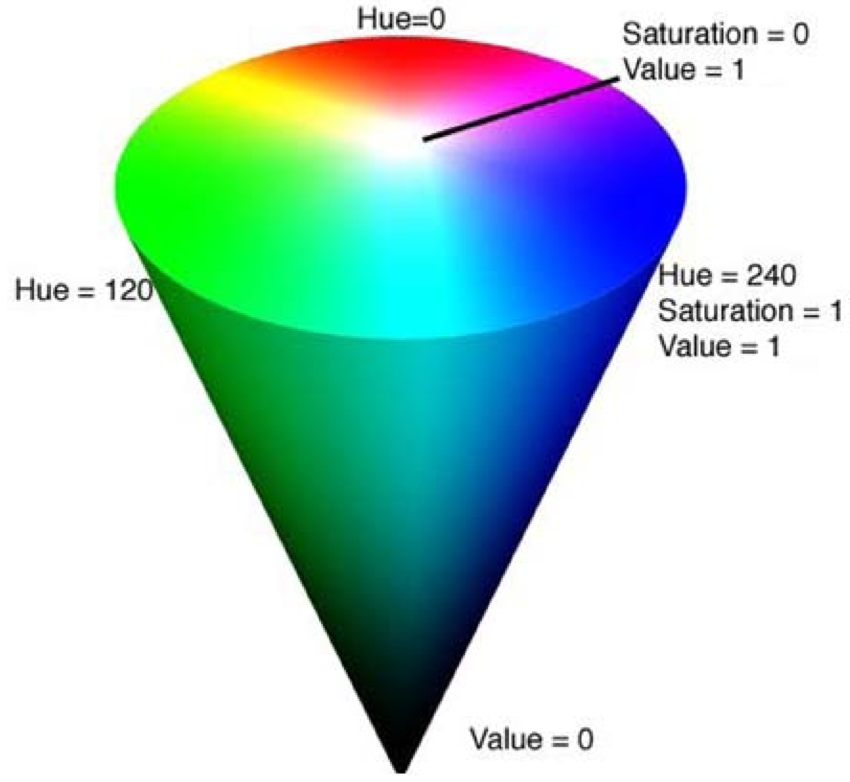


- Each value is an unsigned 8-bit value:
  - (0, 0, 0) is black
  - (255, 0, 0) is red
  - (0, 255, 0) is green
  - (0, 0, 255) is blue
  - (128, 128, 128) is gray
  - (255, 255, 255) is white

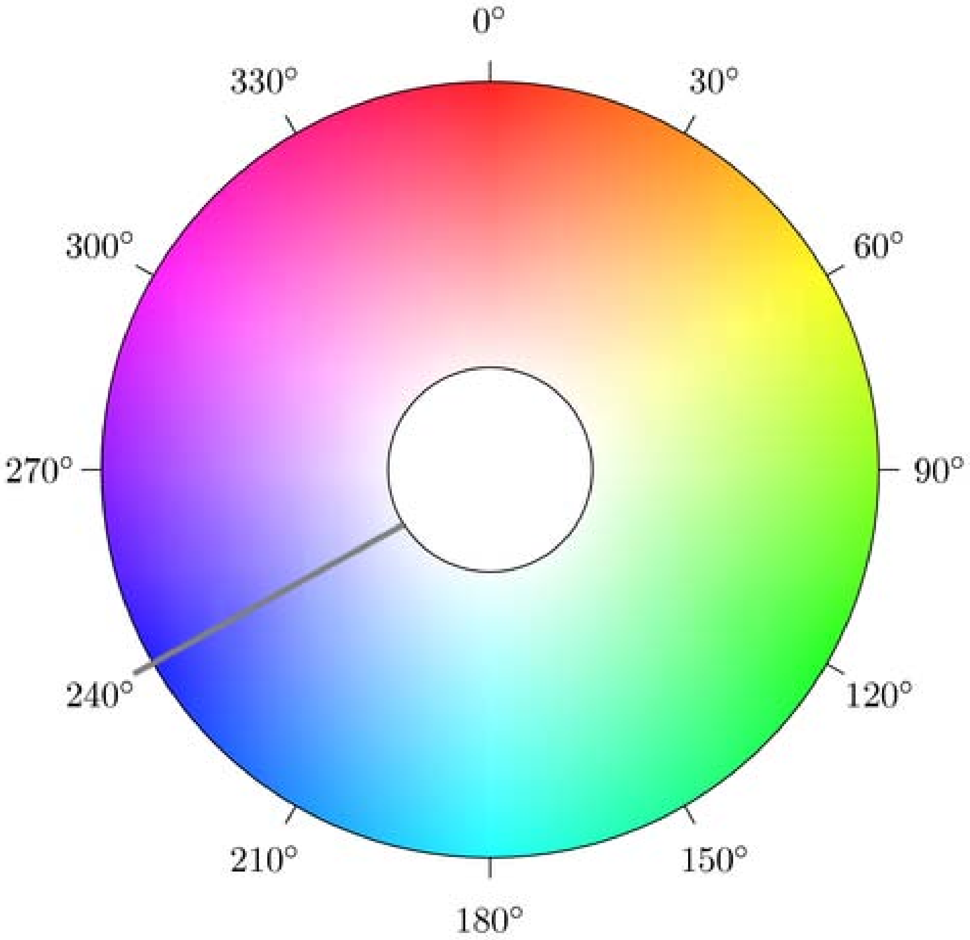
HSV Colour Space

- Code colour as hue, saturation, value.
  - Also called HSB (B for brightness).





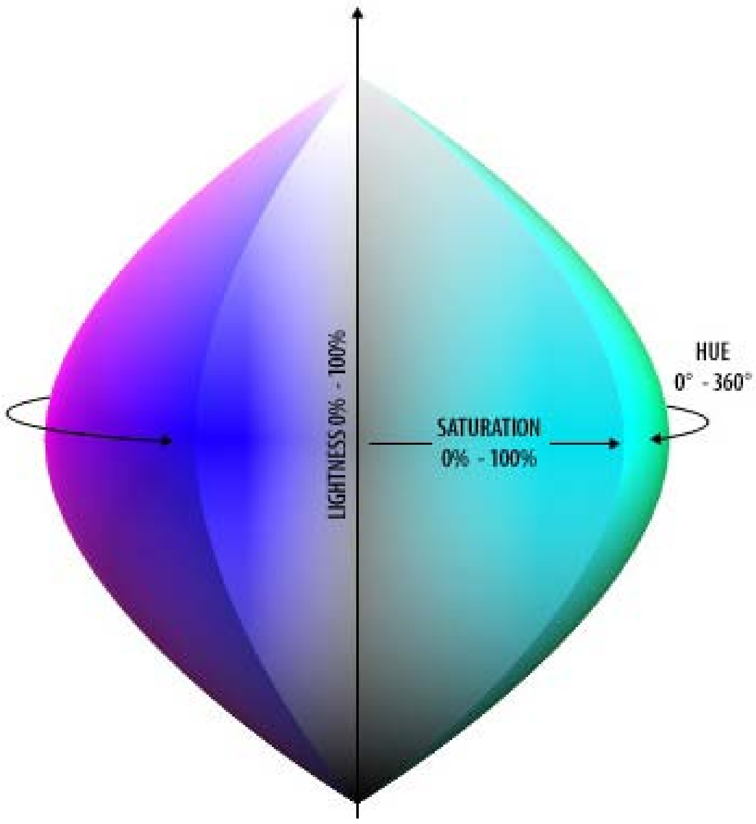
- Hue
  - colour type
  - 0° (red) to 360°
- Saturation
  - Colourfulness
  - 0 to 1 (full colour)
- Value
  - Brightness
  - 0 (black) to 1 (white)

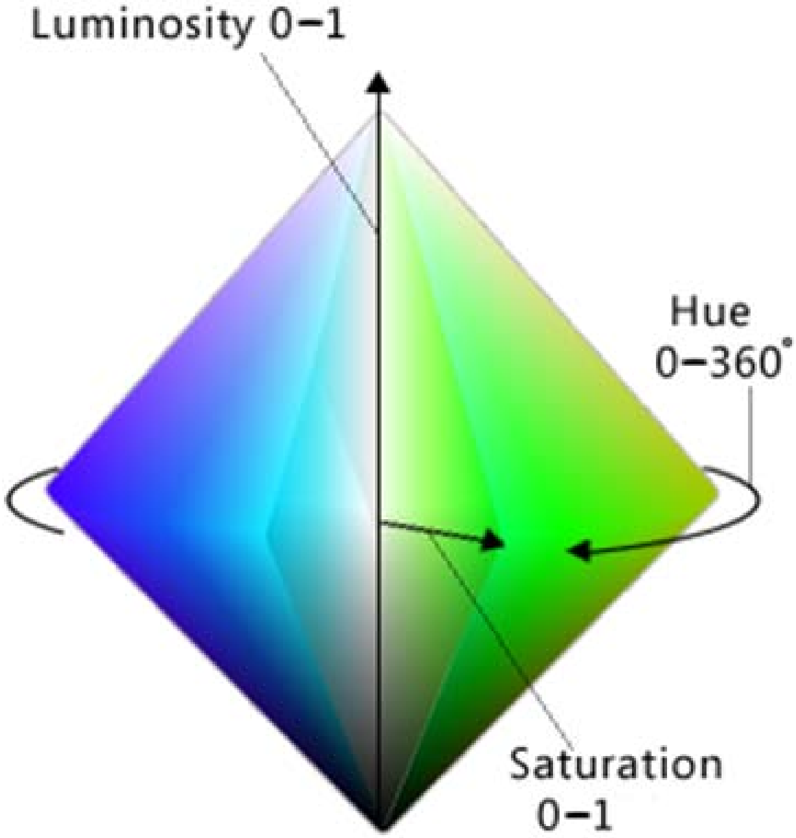


- More intuitive than RGB.

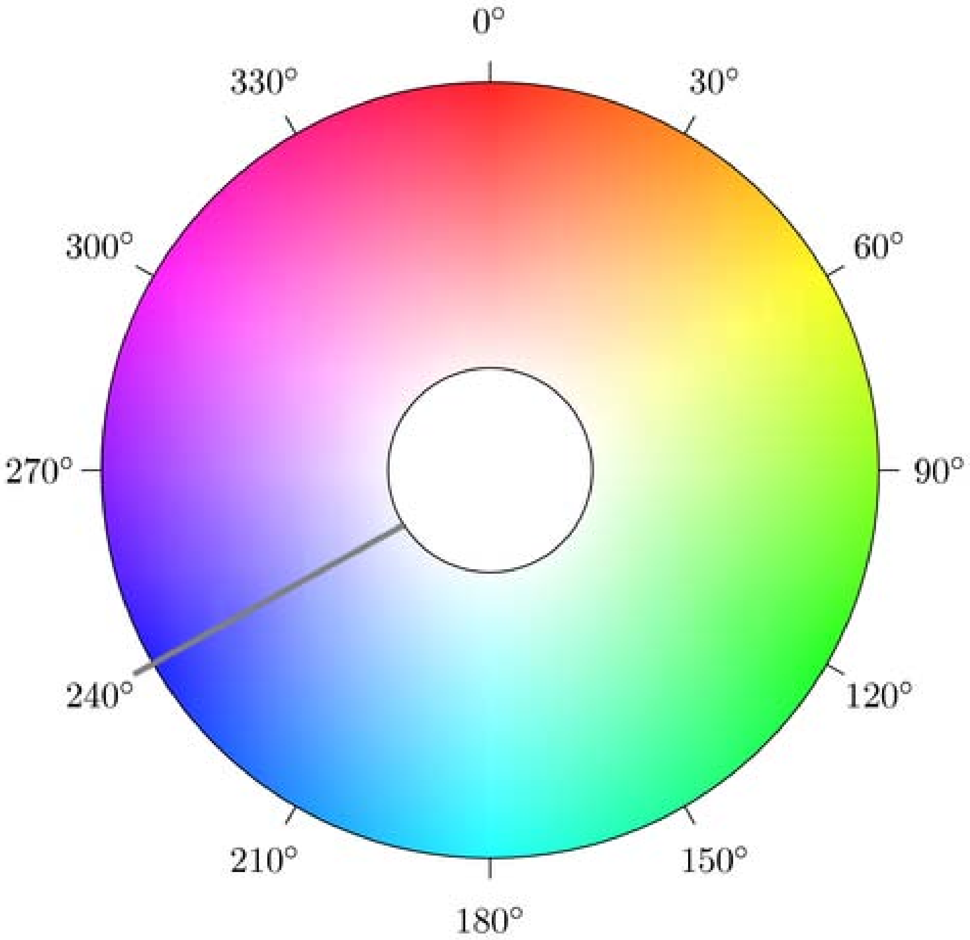
HSL Colour Space

- Code colour as hue, saturation, lightness.
  - Similar to HSV, also called HLS.



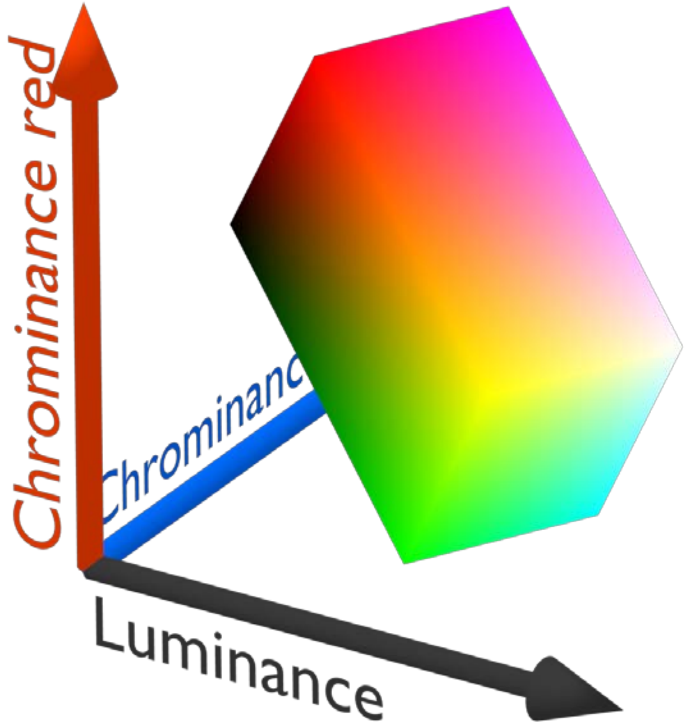


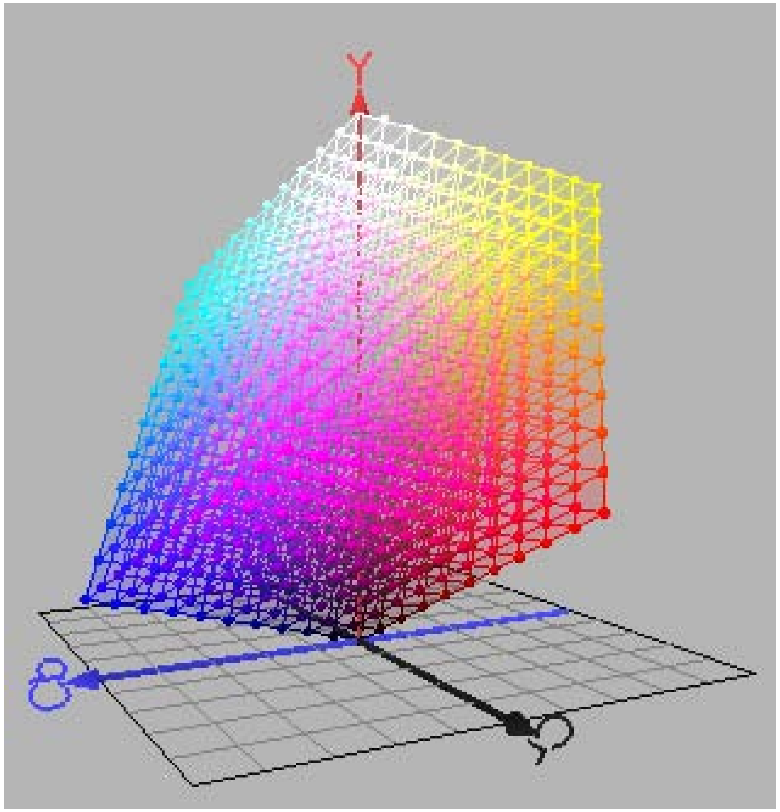
- Hue
  - colour type
  - 0° (red) to 360°
- Saturation
  - Colourfulness
  - 0 to 1 (full colour)
  - Full saturation defined at L = 0.5
- Lightness
  - 0 (black) to 1 (white)



YCbCr Colour Space

- Code colour as Y, Cb, Cr.
  - Used for TV, video.





- Y: luminance
- Cb: blue difference
- Cr: red difference
- For unsigned 8-bit encoding, these values range from 0 to 255.
- Often confused with YUV.

Colour Difference

- Consider two colours C_1 and C_2.
- How to measure difference between C_1 and C_2?
- Simplest difference measure: Euclidean distance

$$d(C_1, C_2) = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$$

  - Straight line distance in RGB space.

Let's do a test…

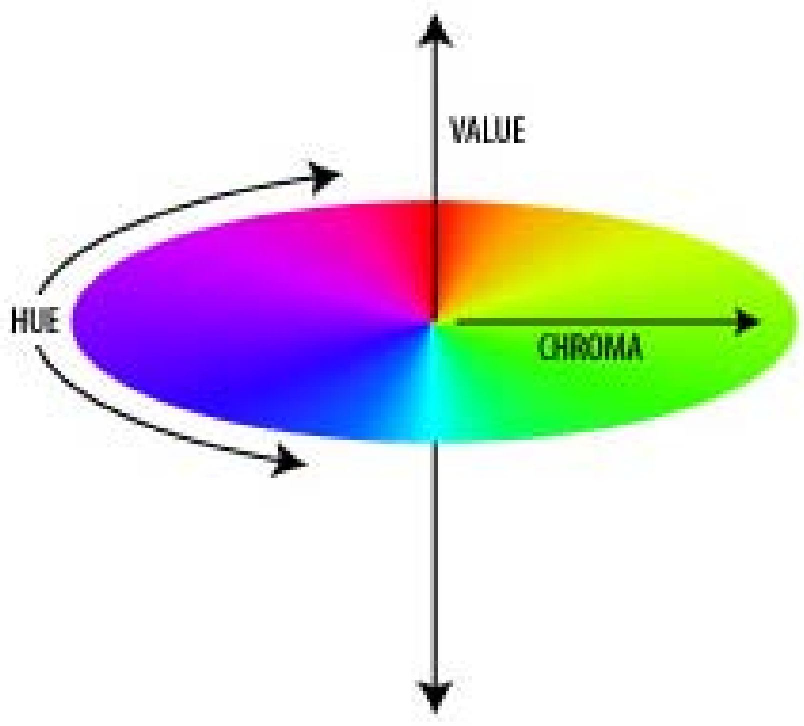
Which colour looks more similar to the middle colour, left or right?

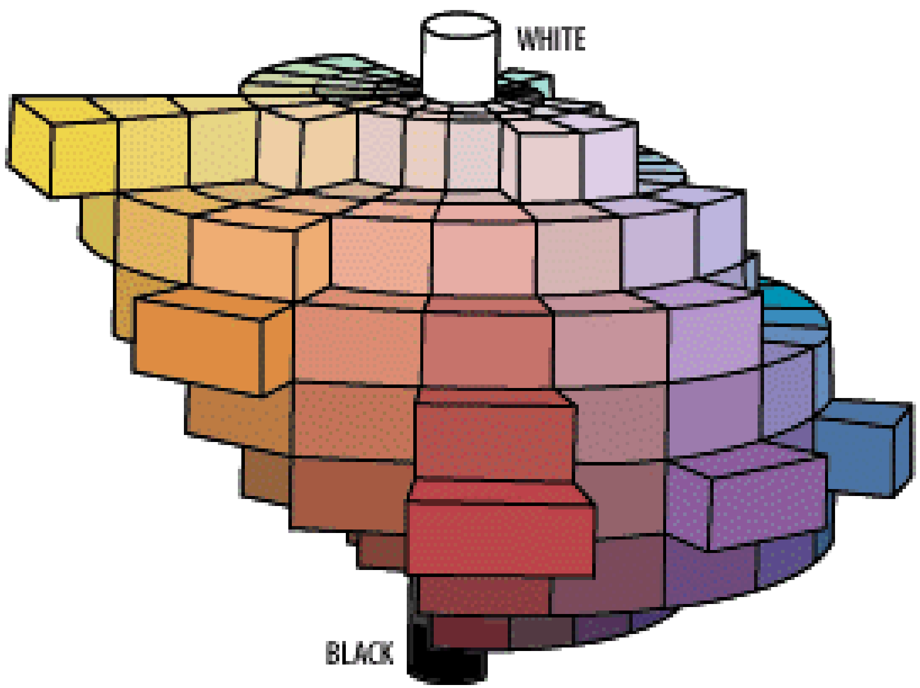
Perceptually Uniform Colour Spaces

- RGB space is not perceptually uniform
  - Equal colour distance $\not\Rightarrow$ equal perceptual difference
  - Inappropriate if need to match human perception.
- HSV, HSL, YCbCr also not perceptually uniform.
- Perceptually (more) uniform colour spaces:
  - Munsell colour space
  - CIELAB
  - CIELUB

Munsell Colour Space

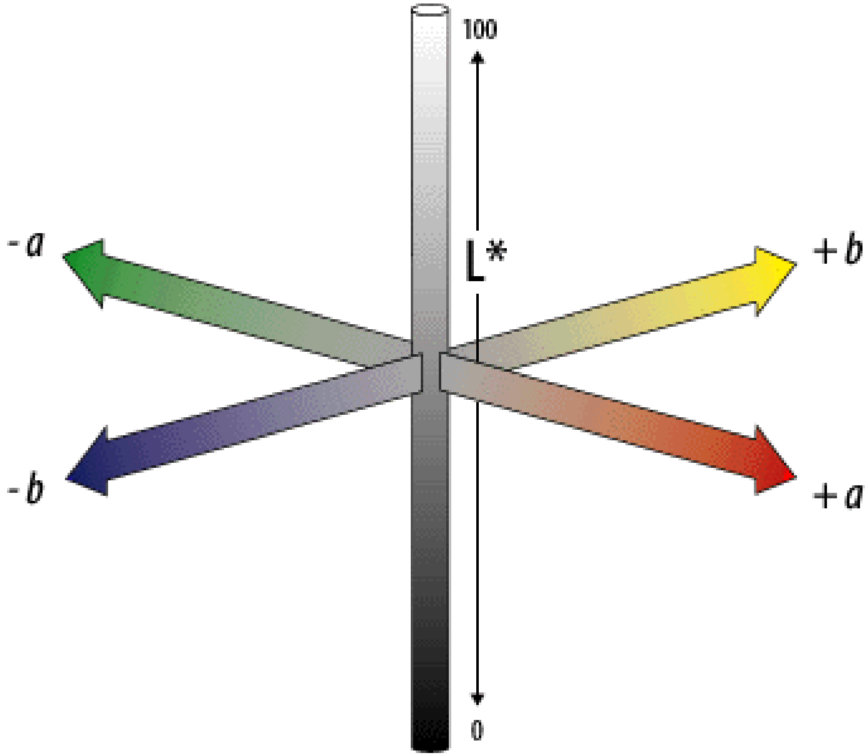
- Code colour as hue, value, chroma.
  - chroma: colourfulness
  - Difficult to use.

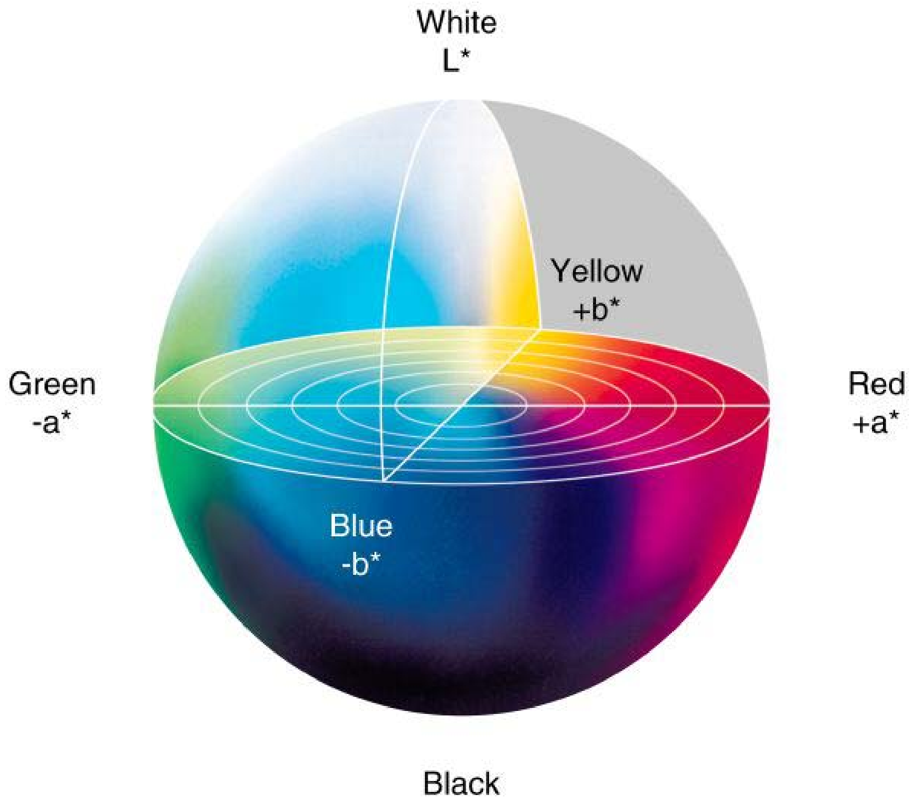




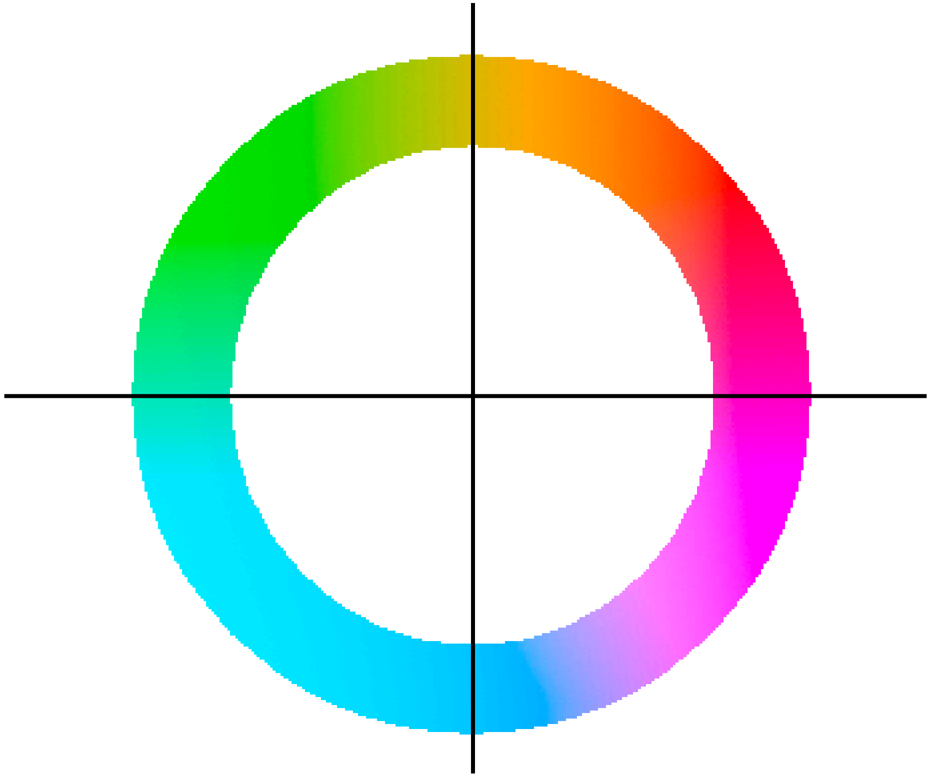
CIELAB

- CIE 1976 L*a*b* colour space





- L* : matches human perception of lightness
  - From 0 (black) to 100 (white).
- a*, b*: hue



CIELUV

- CIE 1976 L*u*v*
  - Similar to CIELAB colour space.
  - L* : range from 0 to 100.
  - u*, v*: typically range from –100 to +100.

Colour Arithmetic

- How to add, subtract, average colours correctly?
- If not careful, can produce invalid colours.

Addition

- With unsigned 8-bit, cannot have value > 255.
- Usually clip to maximum value.
- For example,

$$R = \begin{cases} R_1 + R_2 & \text{if } R_1 + R_2 < 255 \\ 255 & \text{otherwise} \end{cases}$$

- Similarly for *G*, *B*.

image 2

image 1
source 2

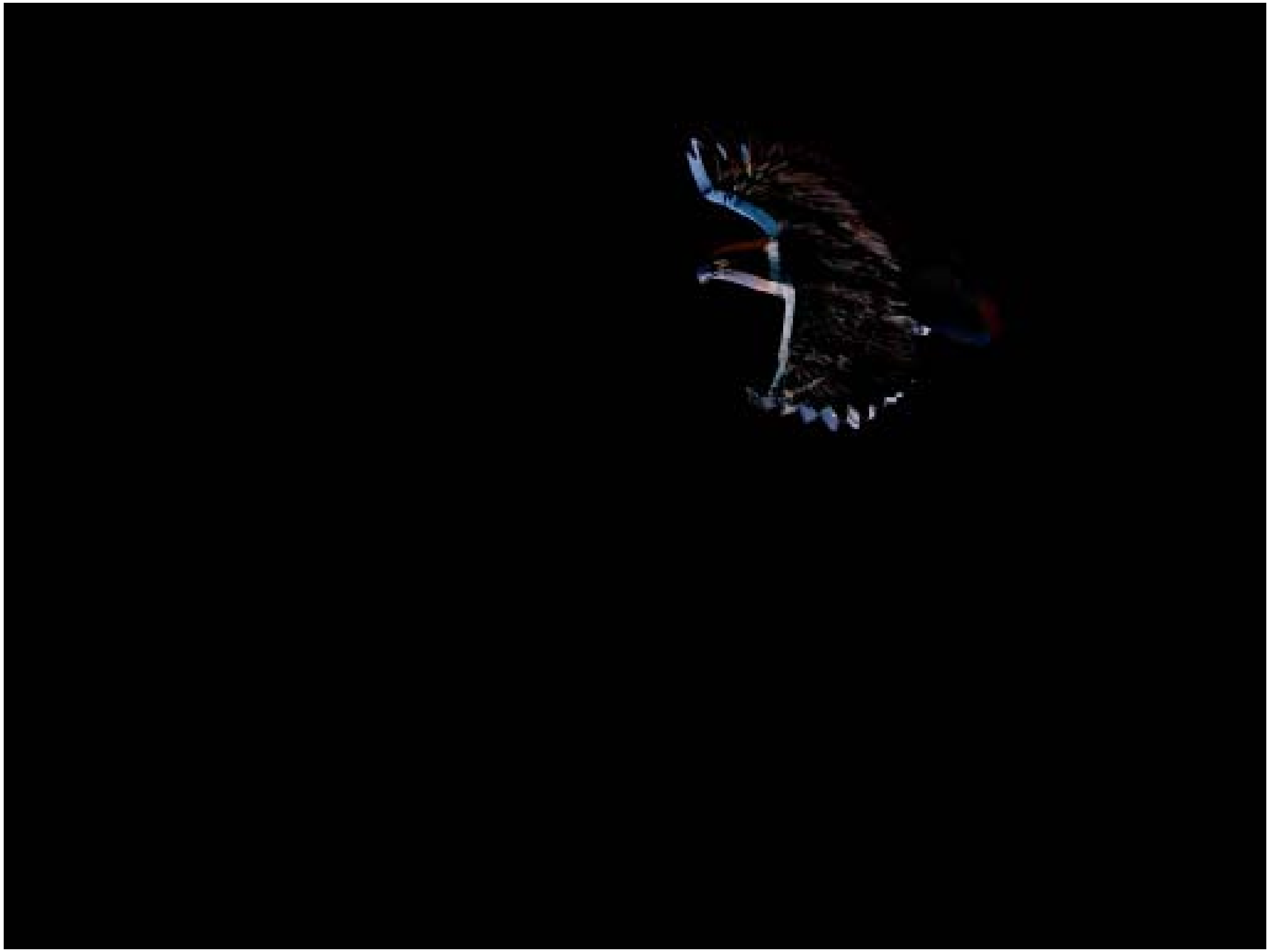
Subtraction

- With unsigned 8-bit, cannot have value < 0.
- Usually clip to minimum value.
- For example,

$$R = \begin{cases} R_1 - R_2 & \text{if } R_1 - R_2 > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Similarly for *G*, *B*.

difference image 2



difference
image 2

Average

- Usual way of computing mean:

sum $$S = \sum_{i=1}^{n} R_i$$

divide $$M = \frac{1}{n} S$$

  - With unsigned 8-bit value, S can overflow for small n.
  - Clipping S produces inaccurate average.
  - Need better methods.

Average

- Method 1: Floating point representation.
  - Then, S doesn’t overflow unless n is very large.
  - S is not a valid colour value.
  - Mean $M = \frac{1}{n}S$ truncated to unsigned 8-bit is valid.

Average

- Method 2: Incremental average
  - With only 1 colour R_1, mean = R_1

$$M_1 = R_1$$

  - With two colours R_1, R_2,

$$M_2 = \frac{1}{2}\left(R_1 + R_2\right) = \frac{1}{2}M_1 + \frac{1}{2}R_2$$

  - With k colours $R_1, R_2, \ldots, R_k$,

$$M_k = \frac{1}{k}\left(R_1 + \cdots + R_k\right) = \frac{k-1}{k}M_{k-1} + \frac{1}{k}R_k$$

  - $M_k, \frac{k-1}{k}M_{k-1}, \frac{1}{k}R_k$ are all valid colours.

image 1
mean 2

image 1
image 2

mean over 8-sec video

Colour-Difference Equations

- Consider a reference colour C_0.
- Euclidean distance of C_1 from C_0 in CIELAB / CIELUV space:

$$\Delta E^*_{ab} = \left[(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2\right]^{1/2}$$

$$\begin{aligned} \Delta L^* &= L^*_1 - L^*_0 \\ \Delta a^* &= a^*_1 - a^*_0 \\ \Delta b^* &= b^*_1 - b^*_0 \end{aligned}$$

  - More perceptually uniform than in RGB space.

- Can also define as:

$$\Delta E^*_{ab} = \left[(\Delta L^*)^2 + (\Delta C^*_{ab})^2 + (\Delta H^*_{ab})^2\right]^{1/2}$$

$$\Delta C^*_{ab} = C^*_{ab,1} - C^*_{ab,0} = ({a_1^*}^2 + {b_1^*}^2)^{1/2} - ({a_0^*}^2 + {b_0^*}^2)^{1/2}$$

$$\Delta H^*_{ab} = [(\Delta E^*_{ab})^2 - (\Delta L^*)^2 - (\Delta C^*_{ab})^2]^{1/2}$$

- Further improvements: CIE94, CMC, BFD.
- More perceptually uniform than Euclidean distance in CIELAB / CIELUV colour space.
- CIE94

$$
\begin{aligned}
\Delta E^*_{94} &= \left[\left(\frac{\Delta L^*}{k_L S_L} \right)^2 + \left(\frac{\Delta C^*_{ab}}{k_C S_C} \right)^2 + \left(\frac{\Delta H^*_{ab}}{k_H S_H} \right)^2 \right]^{1/2} \\
S_L &= 1 \\
S_C &= 1 + 0.045\, \bar{C}^*_{ab} \\
S_H &= 1 + 0.015\, \bar{C}^*_{ab} \\
k_L &= k_C = k_H = 1 \text{ for reference conditions} \\
\bar{C}^*_{ab} &= \sqrt{C^*_{ab,0}\, C^*_{ab,1}}
\end{aligned}
$$

Summary

- Colours have 3 components.
- Most colour spaces are not perceptually uniform.
- CIELAB/CIELUV are perceptually more uniform.
- Be careful with colour arithmetic.
- Use appropriate colour-difference equation.

Further Reading

- Conversion formulae for colour spaces
  - OpenCV user guide
  - Wikipedia
- Colour-difference equations: [Leow2002]
- CIE94, CMC, BFD: [Berns2000, Leow2002]

References

- R. S. Berns. Billmeyer and Saltzman's Principles of Color Technology. John Wiley & Sons, 3 edition, 2000.
- W. K. Leow, Color Spaces and Color-Difference Equations. Tech Report, NUS, 2002.