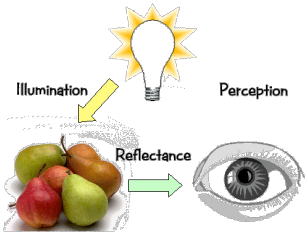
Color

Basics Of Color



Color/Lightness Constancy

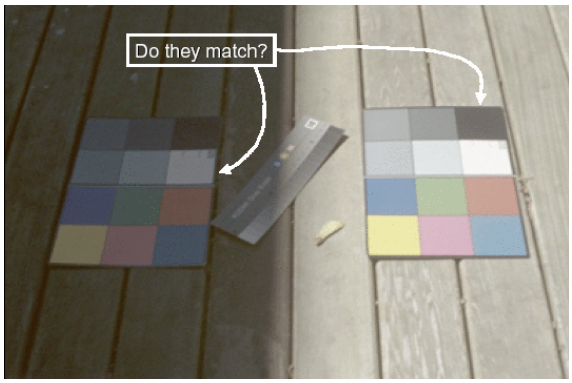


Image courtesy of John McCann

Stroop Effect

- **blue**
- **green**
- **purple**
- **red**
- **orange**

- interplay between cognition and perception

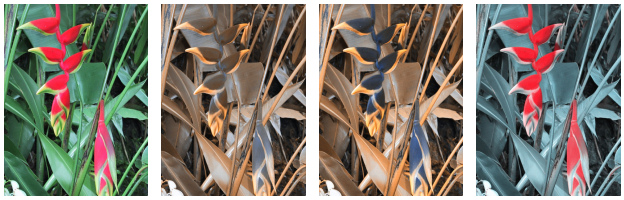
Color Perception

- dogs ?
- birds ?
- your perception of “red” vs my perception of “red”?
- human color perception ?
- why not have RGB inks in printers?

- why not have CMY displays? why 3 colour displays?

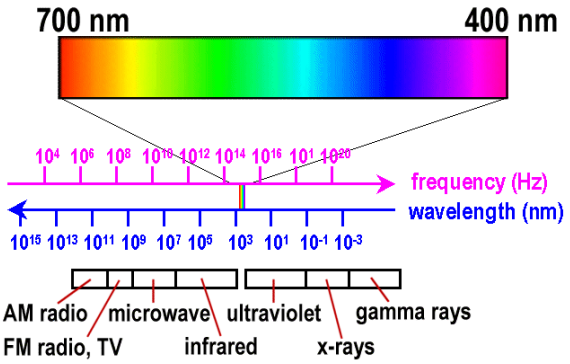
Colour blindness, tetrachromats

- simulating color vision deficiencies



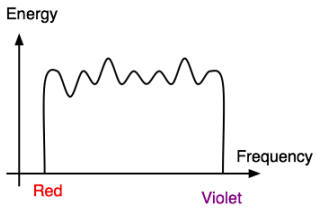
vischeck.com

Electromagnetic Spectrum



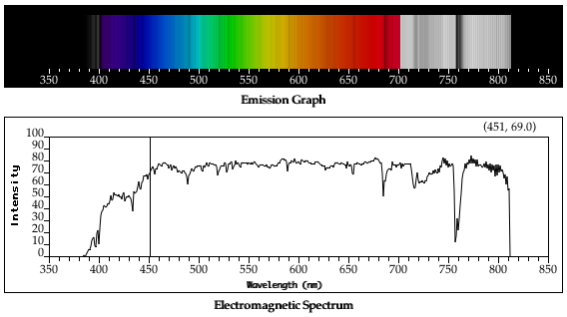
White Light

- sun or light bulbs emit many frequencies within the visible range to produce what we perceive as "white light"



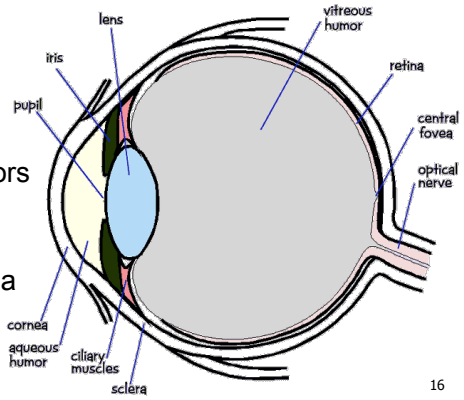
Sunlight Spectrum

- spectral distribution: power vs. wavelength

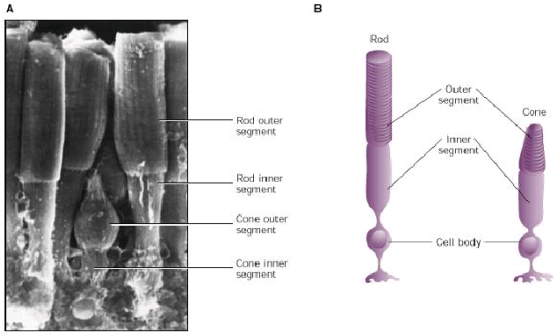


Physiology of Vision

- the retina
  - rods
    - b/w, edges
  - cones
    - 3 types
    - color sensors
  - uneven distribution
    - dense fovea

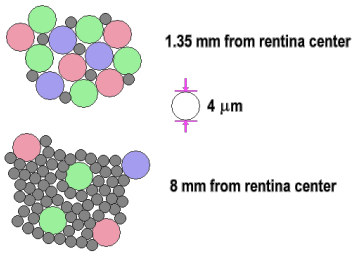


Rods and Cones



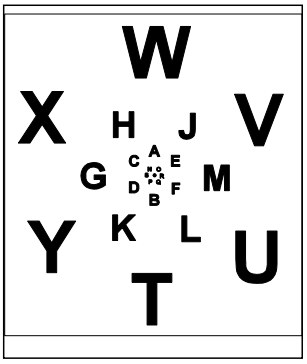
Physiology of Vision

- Center of retina is densely packed region called the *fovea*.
  - Cones much denser here than the *periphery*



Foveal Vision

- hold out your thumb at arm's length



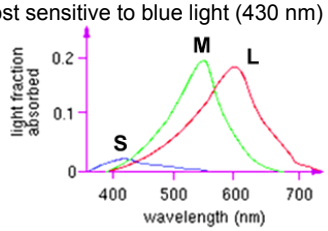
Tristimulus Theory of Color Vision

- Although light sources can have extremely complex spectra, it was empirically determined that colors could be described by only 3 primaries

- Colors that look the same but have different spectra are called metamers

Trichromacy

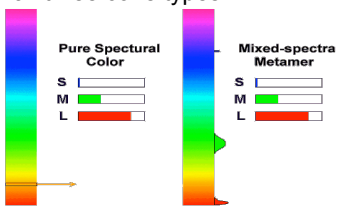
- three types of cones
  - L or R, most sensitive to red light (610 nm)
  - M or G, most sensitive to green light (560 nm)
  - S or B, most sensitive to blue light (430 nm)



  - color blindness results from missing cone type(s)

Metamers

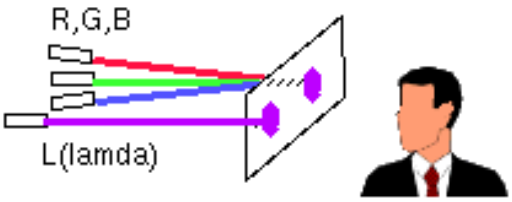
- a given perceptual sensation of color derives from the stimulus of all three cone types



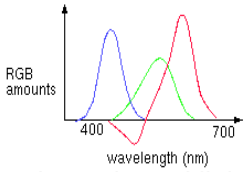
- identical perceptions of color can thus be caused by very different spectra
- demo

http://www.cs.brown.edu/exploratories/freeSoftware/catalogs/color_theory.html

Color Spaces



Negative Lobes



- sometimes need to point red light to shine on target in order to match colors
  - equivalent mathematically to "removing red"
    - but physically impossible to remove red from CRT phosphors
- can't generate all other wavelenths with any set of three positive monochromatic lights!
- solution: convert to new synthetic coordinate system to make the job easy

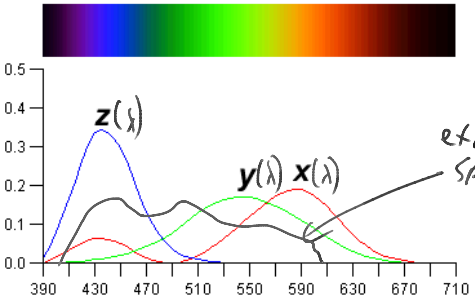
Commission Internationale d'Eclairage

CIE Color Space

- CIE defined 3 "imaginary" lights X, Y, Z
  - any wavelength λ can be matched perceptually by positive combinations

$$Y = \int_\lambda y(\lambda) S(\lambda)\, d\lambda$$

$$Z = \int_\lambda z(\lambda) S(\lambda)\, d\lambda$$



Note that:
X ~ R
Y ~ G
Z ~ B

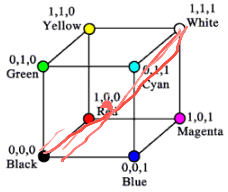
example spectral distribution $S(\lambda)$

what is XYZ color?

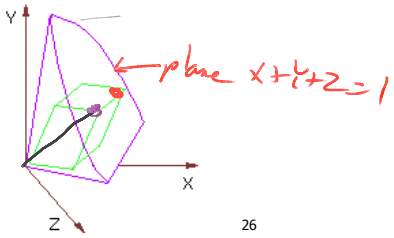
$$X = \int_\lambda x(\lambda) S(\lambda)\, d\lambda$$

RGB Color Space (Color Cube)

- define colors with (r, g, b) amounts of red, green, and blue
  - used by OpenGL
  - hardware-centric



- RGB color cube sits within CIE color space
  - subset of perceivable colors
  - scale, rotate, shear cube



plane $x+y+z=1$

XYZ and RGB colour spaces

- colour transformation matrix:
  (for monochromatic R=700nm, G=546nm,B=436nm)

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} 2.36460 & -0.51515 & 0.00520 \\ -0.89653 & 1.42640 & -0.01441 \\ -0.46807 & 0.08875 & 1.00921 \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{G} \\ \mathbf{B} \end{bmatrix}$$

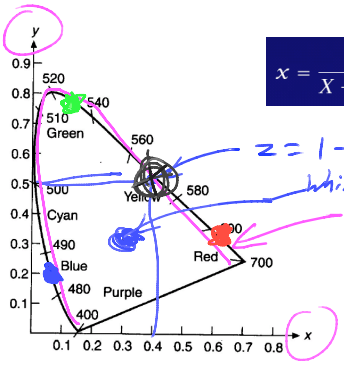
- each monitor has its own RGB-to-XYZ transformation matrix
  - suppose we have a colour $R_A G_A B_A$ on monitor A and wish to view the same colour on monitor B:

$$\textcircled{2}\ \begin{bmatrix} R_B \\ G_B \\ B_B \end{bmatrix} = \begin{bmatrix} & M_B^{-1} & \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad \textcircled{1}\ \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} & M_A & \end{bmatrix} \begin{bmatrix} R_A \\ G_A \\ B_A \end{bmatrix}$$

$$C_B = M_B^{-1} M_A C_A$$

CIE Chromaticity Diagram

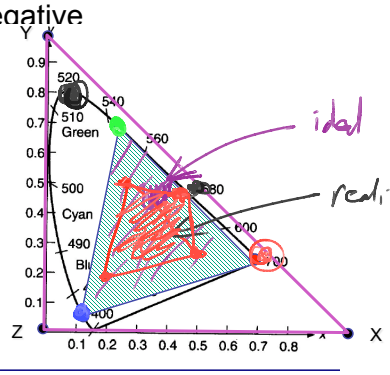
- produce a 2D colour space by projecting onto the plane given by X+Y+Z = 1



$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}$$

RGB vs XYZ revisited

- another view of why the R curve goes negative

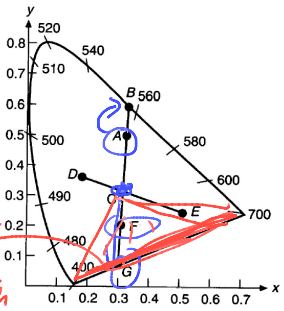


ideal RGB color gamut

realistic color gamut

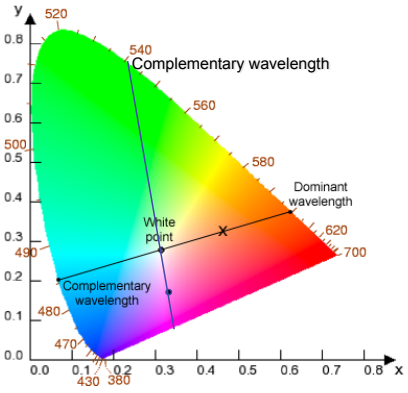
CIE Chromaticity Diagram

- C: white point
- complementary colours → colours that can be mixed to produce white
  e.g. A + F
  B + G
  B + F
  D + E
- dominant wavelength
- non-spectral colors



colors that have no dominant wavelength

Color Interpolation, Dominant & Opponent Wavelength

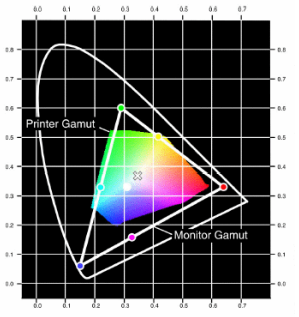


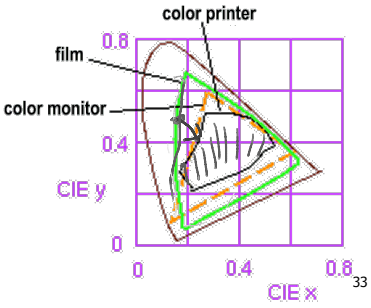
CIE "Horseshoe" Diagram Facts

- all visible colors lie inside the horseshoe
  - result from color matching experiments
- spectral (monochromatic) colors lie around the border
  - straight line between blue and red contains purple tones
- colors combine linearly (i.e. along lines), since the xy-plane is a plane from a linear space

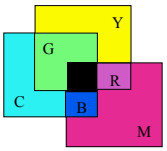
Device Color Gamuts

- gamut is polygon, device primaries at corners
  - defines reproducible color range
  - gamut mapping → mapping into a target colour gamut





The CMY Color Model

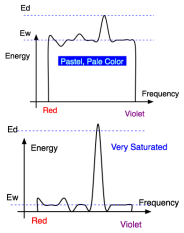


- Used mainly in color printing, where light is absorbed by dyes
- Cyan, Magenta and Yellow primaries are complements of Red, Blue and Green
- Primaries (dyes) subtracted from white paper which absorbs no energy
  - Red = White-Cyan = White-Green-Blue (0,1,1)
  - Green = White-Magenta = White-Red-Blue (1,0,1)
  - Blue = White-Yellow = White-Red-Green (1,1,0)
  - (r,g,b) = (1-c,1-m,1-y)

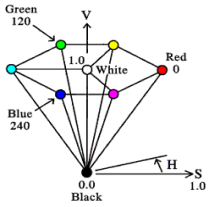
(non-linear wrt XYZ & RGB)

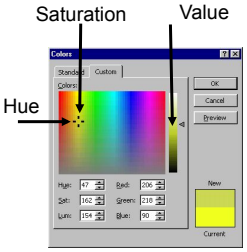
HSV Color Space

- more intuitive color space for people
  - H = Hue
    - dominant wavelength, "color"
  - S = Saturation
    - how far from grey/white
  - V = Value
    - how far from black/white
    - also: brightness B, intensity I, lightness L



polar coords for colour.

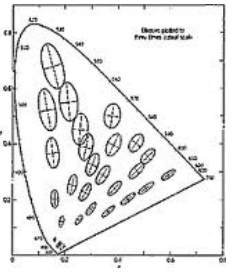


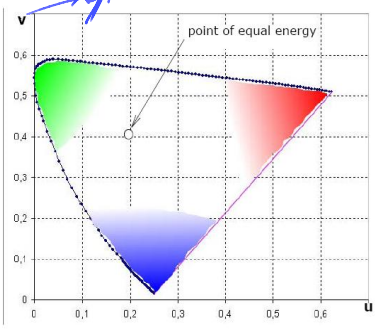


also non-linear wrt XYZ, RGB

LUV color space

Perceptually scaled version of CIE XYZ colour space





→ x'