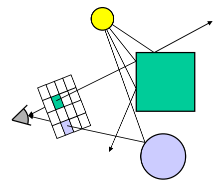
Light & Colors

Light in Computer Graphics

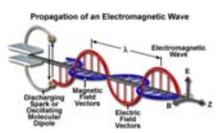


© 1995 by Henrik Wann Jensen

- Computer graphics "=" generating images
- Image = array of pixels
- Each pixel represents one light ray (or more)

Light in Physics

- A light ray is an electromagnetic wave



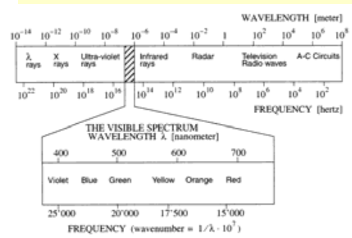
- Propagation speed in vacuum: c
- In general arbitrary shape

→ Sum of harmonic waves (spectrum)

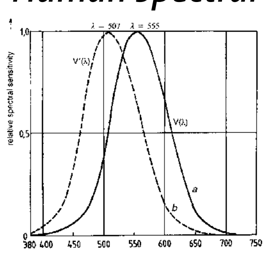
The Visible Spectrum

- Energy is proportional to the frequency f

$$E_{photon} = \frac{hc}{\lambda} = hf \quad (h: \text{Planck's constant}, \lambda: \text{wavelength})$$



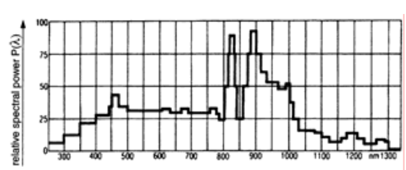
Human Spectral Sensitivity



- V'(λ): By night, photopic (rods, Stäbchen)
- V(λ): Daylight, scotopic (cones, Zäpfchen)

Power Spectrum of Light Sources

- Relative Spectral Power Density $P(\lambda)$



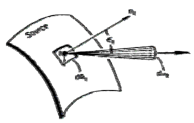
Measuring Light

- Luminous Flux F [lumen] *(Lichtfluss)*

$$F = \text{const} \cdot \int_{380\text{nm}}^{780\text{nm}} P(\lambda)\, V(\lambda)\, d\lambda \qquad \text{const} : 683 \frac{\text{lm}}{\text{W}}$$

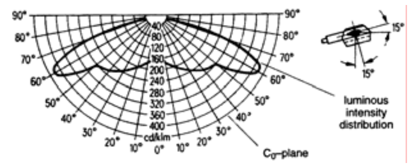
- Luminous Intensity I [candela] *(Lichtstärke)*

$$I = \frac{dF}{d\omega_1}$$

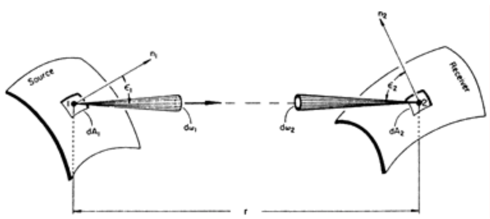


Luminous Intensity Diagram

- Angular distribution of luminous flux for real world light sources



Two Radiant Surface Patches



Measuring Light

- Luminance Y [candela/m²] *(Leuchtdichte)*

$$Y = \frac{d^2F}{dA_1 \cos\varepsilon_1\, d\omega_1}$$

- Illumination B [lux] *(Beleuchtungsstärke)*

$$B = \frac{dF}{dA_2}$$

Measuring Color

- A Definition:

Color is that aspect of visual perception by which an observer may distinguish differences between two structure-free fields of view of the same spatial and temporal properties, such as may be caused by differences in spectral composition of the radiant energy.

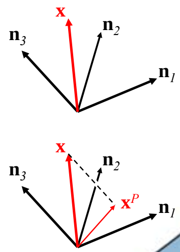
(from: Handbook of Perception and Visual Performance)

Measuring Color

- Each ray carries a spectrum P(λ)
- So far we compressed it to one scalar: $\int_{380\text{nm}}^{780\text{nm}} P(\lambda)\, V(\lambda)\, d\lambda$
- P(λ) contains more information than humans can and need to process
- Humans project P(λ) into a 3D subspace
- Fangschreckenkrebs uses 8D space:

Excursus to 3D Vector Spaces

- $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$ orthonormal basis vectors:
  $\mathbf{x} = x_1\mathbf{n}_1 + x_2\mathbf{n}_2 + x_3\mathbf{n}_3$
- Coordinates are inner products:
  $\mathbf{x} = (\mathbf{x}\cdot\mathbf{n}_1)\mathbf{n}_1 + (\mathbf{x}\cdot\mathbf{n}_2)\mathbf{n}_2 + (\mathbf{x}\cdot\mathbf{n}_3)\mathbf{n}_3$
- Projection onto 2D subspace
  $\mathbf{x}^P = (\mathbf{x}\cdot\mathbf{n}_1)\mathbf{n}_1 + (\mathbf{x}\cdot\mathbf{n}_2)\mathbf{n}_2$

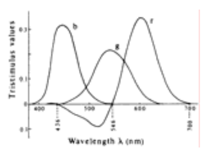


Infinite Dimensional Space

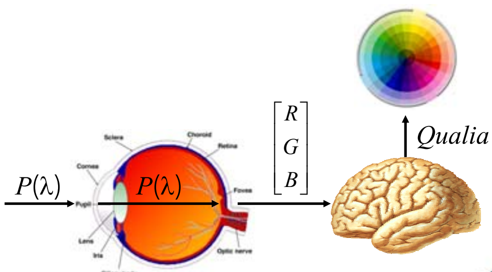
- Infinite dimensional vector is a function:
  $\mathbf{x}^{3D} = (x_1, x_2, x_3) \quad \rightarrow \quad \mathbf{x}^{\inf} = x(\lambda)$
- Infinite number of basis functions needed
- Projection onto 3D subspace with
  $n_1(\lambda)$, $n_2(\lambda)$, $n_3(\lambda)$ orthonormal basis functions:
  $x^P(\lambda) = x_1 n_1(\lambda) + x_2 n_2(\lambda) + x_3 n_3(\lambda)$
- Coordinates are continuous inner products:
  $x_i = \int x(\lambda) n_i(\lambda) d\lambda$

The Human Eye

- Spectrum $P(\lambda)$ is infinite dimensional
- Eye projects $P(\lambda)$ into 3D subspace
- Three types of cones (photopic vision) are three basis functions r(λ), g(λ), b(λ)

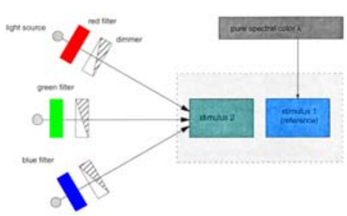


Qualia

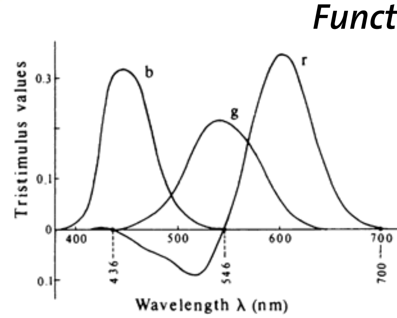


The CIE Primary System (1931)

- Commission Internationale de l'Eclairage
- Setup for measuring human color sensitivity
  (435.8 nm, 546.1 nm, 700.0 nm)



Spectral Sensitivity Functions



The CIE Spectral Response Functions

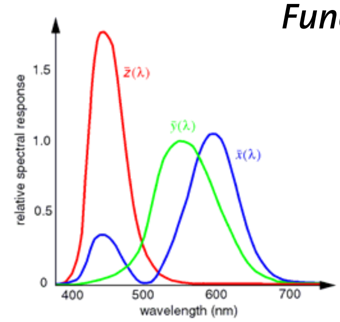
- Normalized, positive definite functions
- Y =const. * F
- Linear (matrix) transform is standardized

$$\bar{x}(\lambda) = +2.36\,\bar{r}(\lambda) \quad -0.515\,\bar{g}(\lambda) \quad +0.005\,\bar{b}(\lambda)$$
$$\bar{y}(\lambda) = -0.89\,\bar{r}(\lambda) \quad +1.426\,\bar{g}(\lambda) \quad +0.014\,\bar{b}(\lambda)$$
$$\bar{z}(\lambda) = -0.46\,\bar{r}(\lambda) \quad +0.088\,\bar{g}(\lambda) \quad +1.009\,\bar{b}(\lambda)$$

- Linear combinations

→ new basis spans same 3D subspace

CIE Spectral Response Functions



CIE Primaries of a Color Stimulus

- Vector (X, Y, Z) provides a ***quantification*** of any spectral color stimulus $P(\lambda)$
- Compute by inner products of $x,y,z(\lambda)$ and $P(\lambda)$

$$X = \int_{380nm}^{780nm} P(\lambda)\bar{x}(\lambda)d\lambda$$
$$Y = \int_{380nm}^{780nm} P(\lambda)\bar{y}(\lambda)d\lambda$$
$$Z = \int_{380nm}^{780nm} P(\lambda)\bar{z}(\lambda)d\lambda$$

The CIE Chart

- 2D chart in practice by projection into the plane perpendicular to spatial diagonal

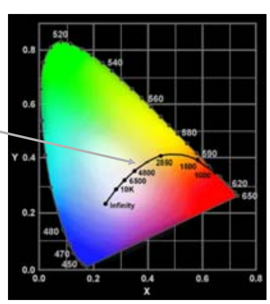
$$x + y + z - 1 = 0$$
$$\text{plane normal}: \mathbf{n} = [1 \quad 1 \quad 1]^T$$

- (x, y) pair characterizes color

$$x = \frac{X}{X+Y+Z} \qquad y = \frac{Y}{X+Y+Z} \qquad z = 1 - x - y$$

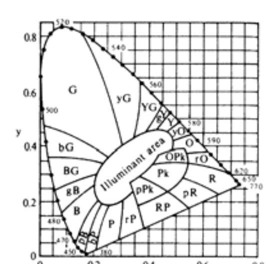
The CIE Chart

Color of Planck's black body

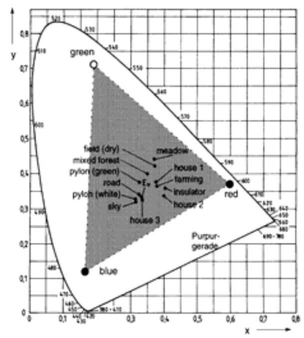


Color Definitions

R – Red
B – Blue
G – Green
Y – Yellow
O – Orange
P – Purple
Pk – Pink
Lower case takes suffix ish



Examples of Real World Objects

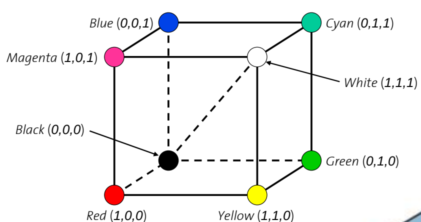


Features

- White Point
- Isolines of saturation
- Hue (Farbart)
- Color Temperature
- Purple line
- Dominant wavelength
- Domain of visible colors
- Inverse color

RGB Color Space

- Unit cube with R,G,B as basis vectors



XYZ to RGB Transform

- Measure the phosphor coordinates of your monitor (see manual)
- Take them as basis vectors of the transform matrix
- Compute inverse

B/W image

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \begin{matrix} = \\ = \\ = \end{matrix} \begin{pmatrix} X_R & X_G & X_B \\ Y_R & Y_G & Y_B \\ Z_R & Z_G & Z_B \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- B/W conversion: $Y = 0.3R + 0.59G + 0.11B$

Alternative

- Given CIE chart coordinates (x, y) of the primaries and the white point (X_w, Y_w, Z_w)
- Compute equations below
- Used for active color systems (monitors, projectors)

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{bmatrix} x_R C_R & x_G C_G & x_B C_B \\ y_R C_R & y_G C_G & y_B C_B \\ (1-x_R-y_R)C_R & (1-x_G-y_G)C_G & (1-x_B-y_B)C_B \end{bmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

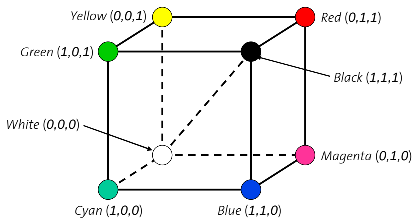
$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = \begin{bmatrix} x_R & x_G & x_B \\ y_R & y_G & y_B \\ (1-x_R-y_R) & (1-x_G-y_G) & (1-x_B-y_B) \end{bmatrix} \cdot \begin{pmatrix} C_R \\ C_G \\ C_B \end{pmatrix}$$

CMY Color Space

- Used in passive color systems (printers)
- Inverse to RGB
- Transform given by

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad resp. \quad \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} C \\ M \\ Y \end{pmatrix}$$

CMY Color Space

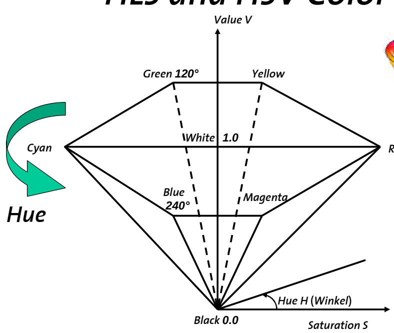


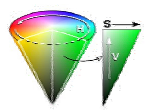
YIQ Color Space

- Uses Luminance, inphase (green-orange) and quadrature (blue-yellow) components
- Advantages for natural and skin colors
- NTSC US-color TV standard
- Bandwidth partitioning (2.4 MHz, 1.5 MHz, 0.6 MHz)

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

HLS and HSV Color Spaces





HLS and HSV Color Spaces

- Perceptual color spaces
- More intuitive for interactive color synthesis
- Hue, Lightness and Saturation explicitly given
- Approximation of CIE lightness

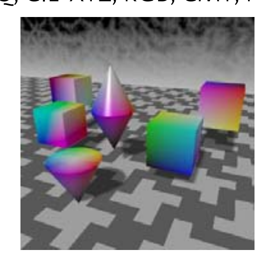
Conversion Procedures

- Conversion procedure (RGB→HSV)

```
min = min(R, G, B);
max = max(R, G, B);
V = max;
If (max != 0)
  S = (max - min)/max;
else
  S = 0;
H = Hue (V, S, R, G, B); //procedural
  comp.
```


3D Color Bodies

- YIQ, CIE-XYZ, RGB, CMY, HSV, HLS

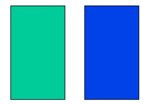


www.okino.com/slidshow/colspace.htm

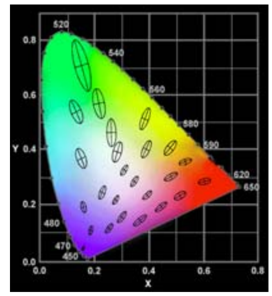
Higher Order Colorimetry

- Measuring “perceptual distance” in color spaces
- Important for many industrial branches (textile, automotive etc.)
- Experiments of McAdams
- Ellipsoidal perceptual thresholds in CIE chart

McAdams Ellipses



Test patches



L*a*b* Color Space

- Nonlinear Warp of RGB

$$L^* = 25\left[\frac{100Y}{Y_W}\right]^{1/3} - 16$$

$$a^* = 500\left[\left(\frac{X}{X_W}\right)^{1/3} - \left(\frac{Y}{Y_W}\right)^{1/3}\right]$$

$$b^* = 200\left[\left(\frac{Y}{Y_W}\right)^{1/3} - \left(\frac{Z}{Z_W}\right)^{1/3}\right]$$

(X_W, Y_W, Z_W): Coordinates whitepoint

L*u*v* Color Space

$$u = \frac{4X}{X + 15Y + 3Z}$$

$$v = \frac{9Y}{X + 15Y + 3Z}$$

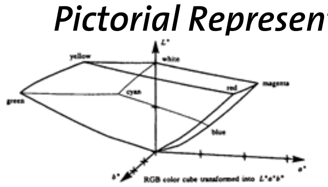
$$L^* = 25\sqrt[3]{\frac{100Y}{Y_W}} - 16$$

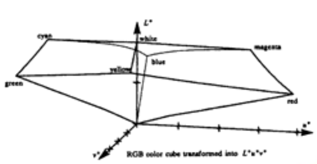
$$u^* = 13L^*(u - u_W)$$

$$v^* = 13L^*(v - v_W)$$

(Y_W, u_W, v_W): Coordinates of Whitepoint

Pictorial Representation





OpenGL Color

- Primitive oriented (vertex)

```
glColor3f(r,g,b);
glColor4f(r,g,b,a);
```

- Normalized to [*0*,..,*1*]
- RGBA mode versus color index mode
- Depending on number of bitplanes per pixel
- n bitplanes gets 2^n colors
- *8* Bits/component -> true color
- dithering option

```
glEnable(GL_DITHER);
```

OpenGL Color

- Color index mode uses lookup table
  `glIndex(I); glutSetColor();`
- Optimal lookup tables refer to clustering algorithms (median cut)
- Size between 2^8 and 2^{12}
- Mode Specification using
  `glutInitDisplayMode();`
- Color interpolation by
  `glShadeModel(GL_SMOOTH);`

Example

- A smooth triangle