Basic Techniques in Computer Graphics

Winter 2017 / 2018

The slide comments are not guaranteed to be complete, they are no alternative to the lectures itself. So go to the lectures and write down your own comments!

Colors

- **Art**
- Physics
- Biology
- Computer graphics

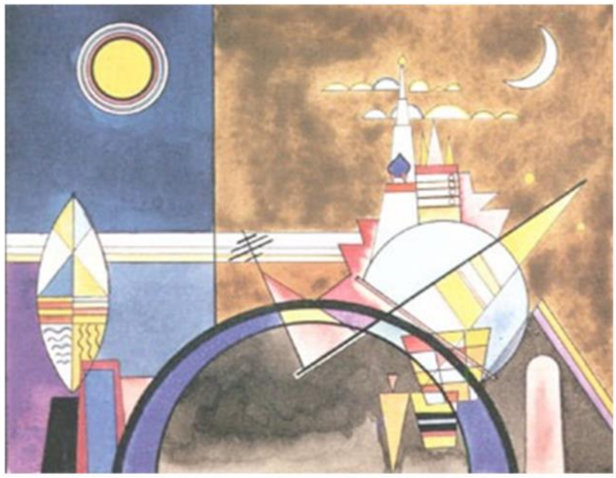
First we want to look at how color is seen from the standpoint of artists.

Color Circle

Goethe

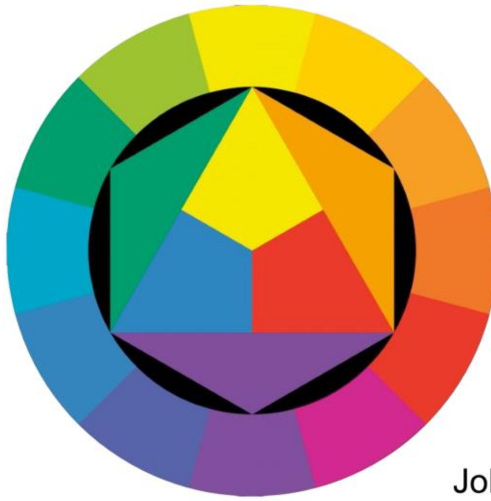
Goethe invented the concept of a color circle (or color wheel). For Goethe his scientific work on colors were more imported to him than his literature. His physical explanation however were less accurate than Newtons theory. He parted the color circle into warm (red, yellow, orange) and cold colors (blue, green).

Art



One example: „cold“ colors in a painting by Kandinsky on the left side depicting a day scene and “warm” colors on the right side depicting a night scene. So in this example the more common usage of colors is switched intentionally by the artist.

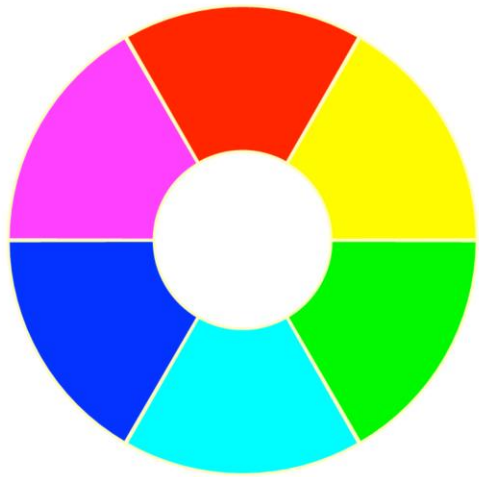
Color Circle



Johannes Itten

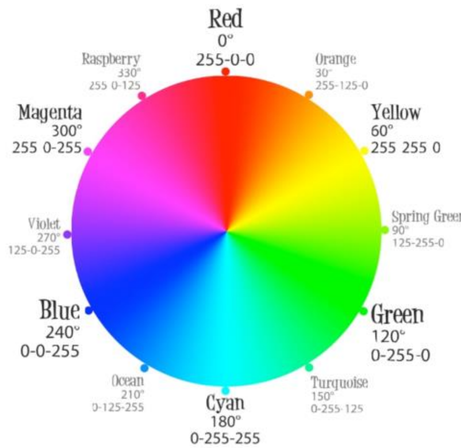
Starting with the primary colors yellow, red and blue (middle), we learned in school that you can mix together all colors (middle triangle and outer circle). The resulting color circle is similar to Goethes but is derived in a more systematic way.

Color Circle



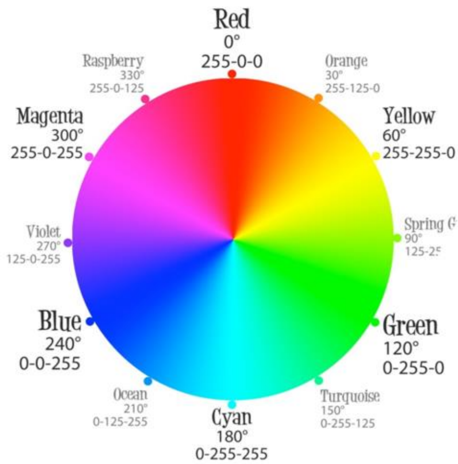
Most modern PC displays as well as TVs use red, green and blue as there primary colors. Those mixed will create magenta, yellow and cyan. Those three colors are the primary colors of printers (with the addition of black).

Color Circle



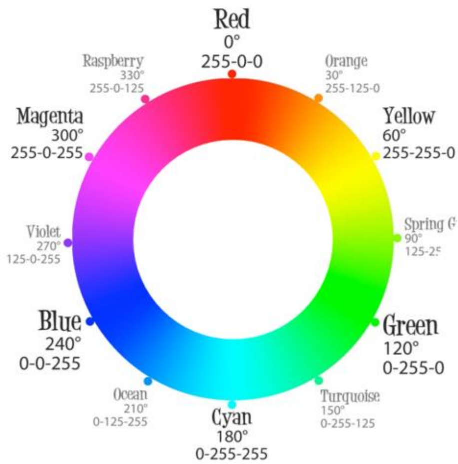
Mixing the primary colors leads to a continuous circle in which a color can get described as a mix of the primary colors (in this example as integers from 0 to 255) or as an angle on the circle.

Color Circle



Itten

Color Circle



Itten

Terminology

- hue (Farbton)
- saturation (Sättigung)
- brightness (Helligkeit)

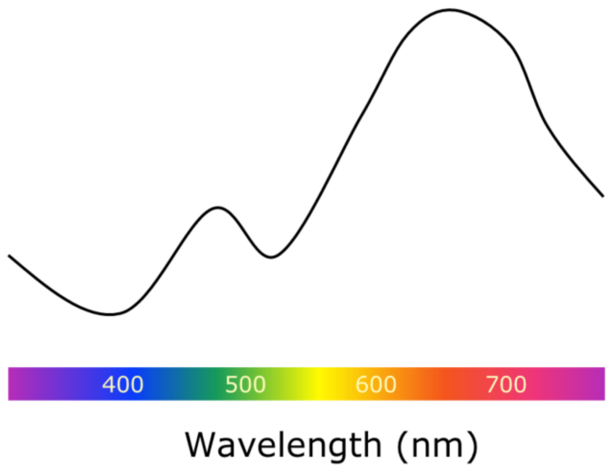
Artists differentiate colors based on hue, saturation and brightness.
In german: Farbton, Sättigung, Helligkeit.
Hue is the base color, saturation a measure of how pure that color is and brightness a measure of the perceived amount of energy of the color.

Colors

- Art
- **Physics**
  - Electromagnetic waves
  - Wavelength: 380 nm – 780 nm
    (violet, blue, green, yellow, orange, red)
- Biology
- Computer graphics

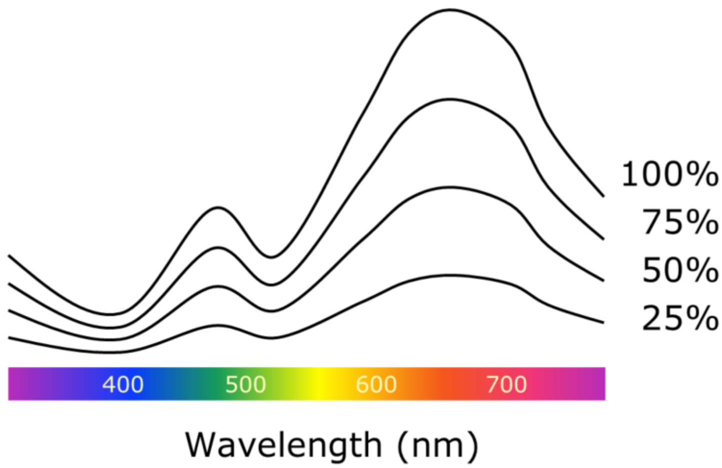
Light and colors are just electromagnetic waves. The human eye is able to see wavelength of around 380 nm to 780 nm.

Spectral Distribution Function



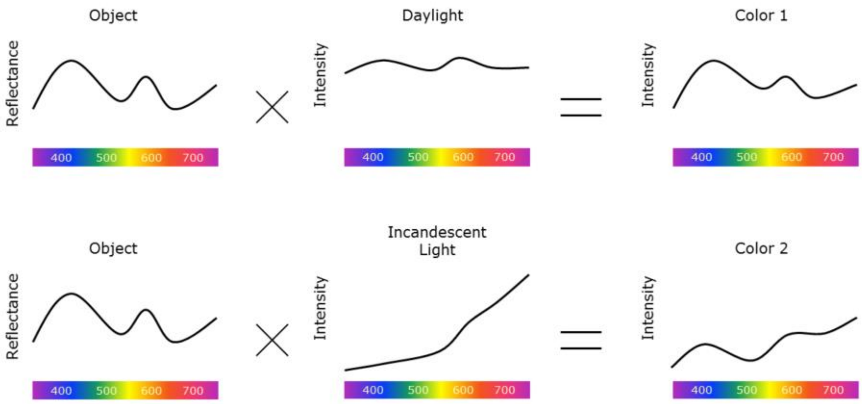
Light of just one wavelength is very rare (one example would be a laser). Normally we encounter a mixture of wavelength. Such a spectrum will create one specific color impression and in contrast to a abstract color description in terms of hue/saturation/brightness this gives us a mathematical, physically based function to describe a color.

Spectral Distribution Function



Scaling down all values of that function does not change the ‚color' but only the brightness. So the integral of the lightspectrum can be seen as the brightness.

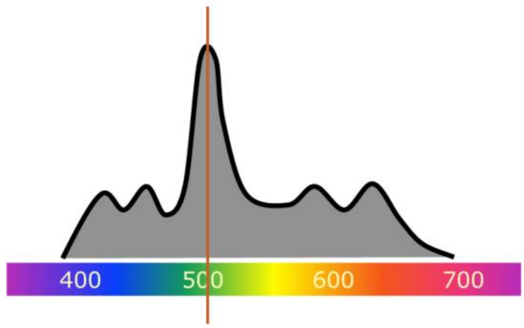
Visible Color



Light-Object interaction: The same object as seen under different lighting conditions can create different color impressions. One example of incandescent light is a normal (not energy saving) light bulb. The colors that we see will always be a product of the light source characteristics and the material properties (unless we see directly into the light source).

Physical Terms

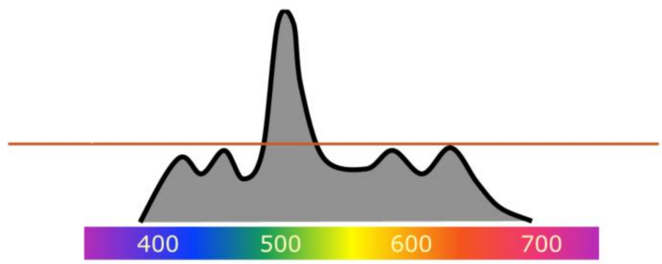
- **hue = dominant wavelength**
- saturation = excitation purity
- lightness/brightness = luminance



The dominant wavelength of a spectrum is what the artists mean by hue.

Physical Terms

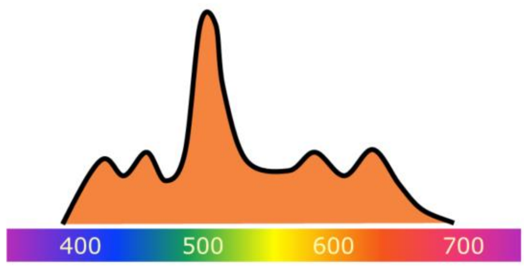
- hue = dominant wavelength
- **saturation = excitation purity**
- lightness/brightness = luminance



The saturation is the purity of a color. The higher the contribution of the dominant wavelength is to the full spectrum, the purer the color. Because we can‘t distinguish all individual wavelength, you would want to look at a small wavelength range.

Physical Terms

- hue = dominant wavelength
- saturation = excitation purity
- **lightness/brightness = luminance**

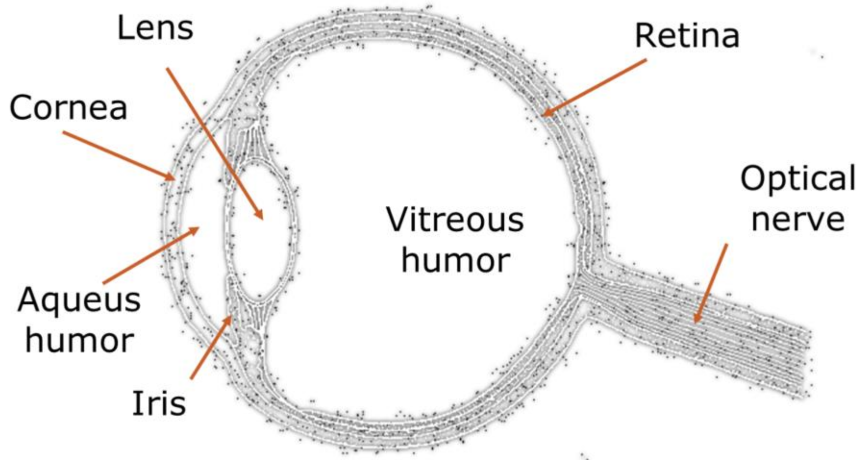


As already stated, the area under the function is the brightness.

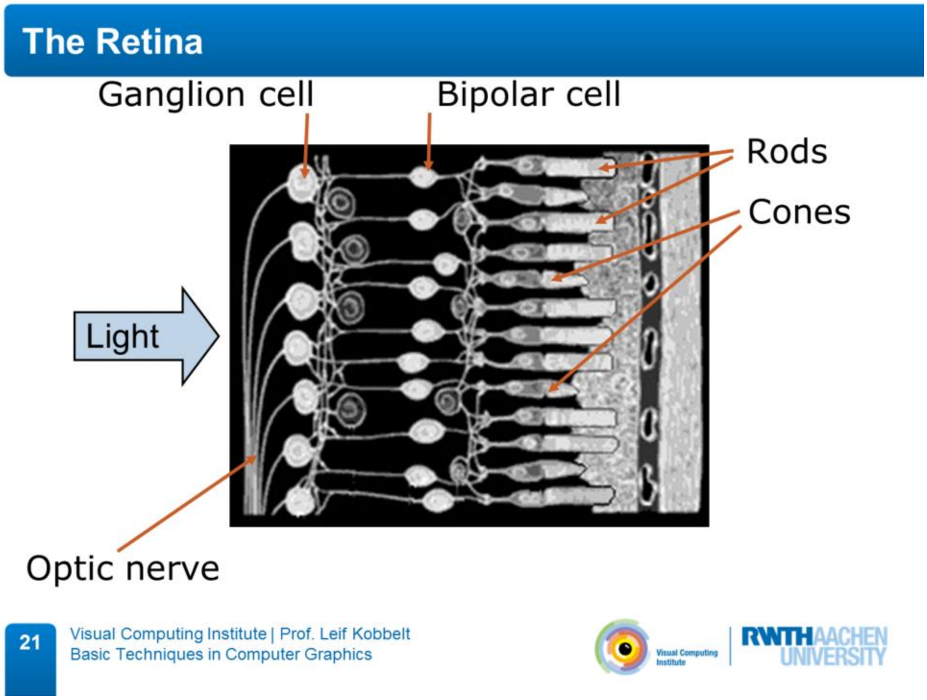
Colors

- Art
- Physics
- **Biology**
  - Three types of cones
  - Luminance-efficiency function
  - CIE Standard
- Computer Graphics

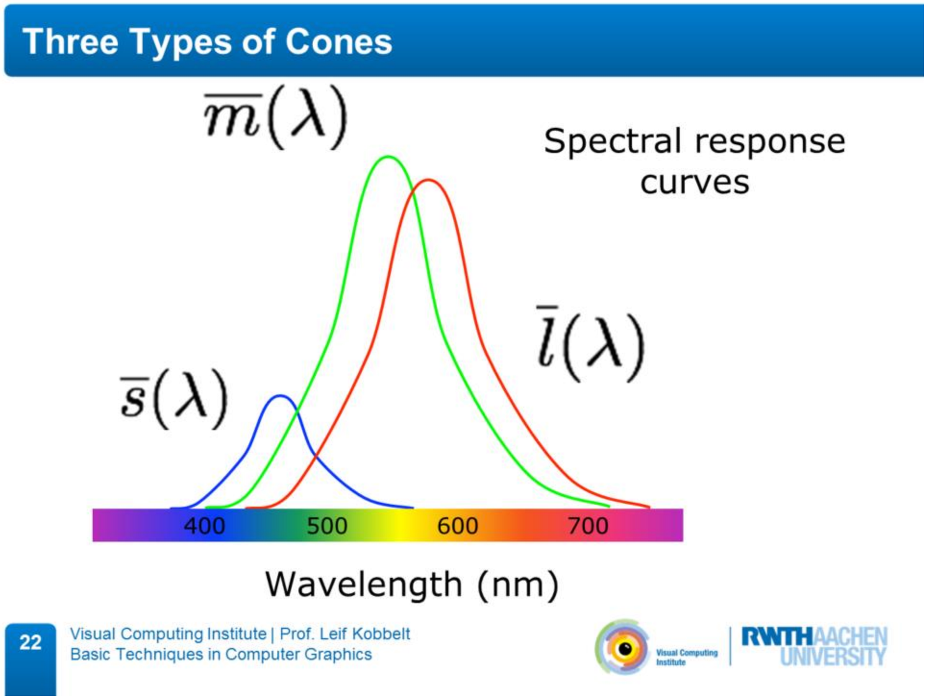
The Human Eye



Sidecut through the human eye.



The sensors on the retina are already connected to each other to perform some simple forms of image filtering before the information even gets to the brain. We have two different sensors in our retina: rods and cones. Cones can distinguish between red, green and blue while the rods only measure the brightness.



The retina has less cones that are sensitive to blue than red and green. To green the human eye is most sensitive. This explains why the red-yellow part of the color circle from Itten was bigger than the blue part (note: yellow is seen when red and green cones fire from which we have a lot).

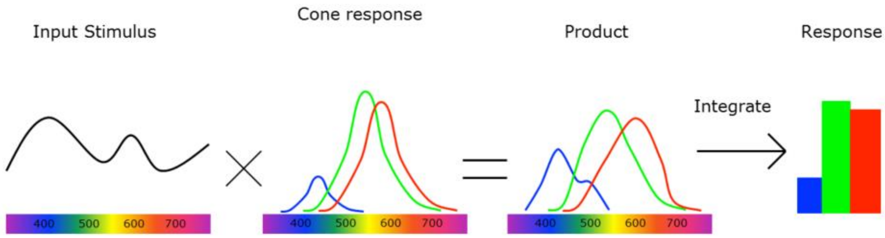
Other animals

- many mammals have only two cones (blue/green)
- non-mammals have often more than three cones

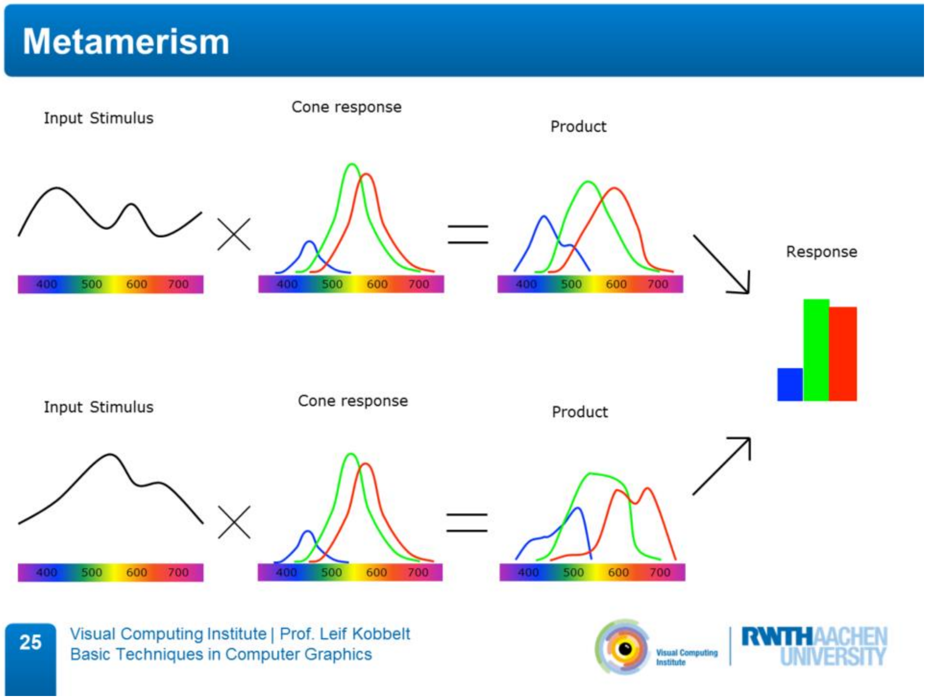
Many mammals have only two kinds of cones, due to a genetic mutation in primates, humans now have three cones (some of the green seeing cones shifted into the red spectrum which was an advantage as those primates could distinguish fruit from leaves better). About 8% of males and 0.5% of females do not have this mutation and have a form of red-green color blindness.

In extremely rare cases is it possible for humans to have a fourth kind of cone sensitive between the red and green cones which can result in differentiating colors better (but not being able to see additional colors). Tetrachromacy (having four kinds of cones) on the other hand exists in fish, insects, birds, reptiles and amphibians.

Trichromacity



As we have seen that we have only three cones to see colors, we have to multiply the input wavelength by the three response functions of our cones to get the three cone responses. The continuous input stimulus is reduced to a three dimensional space of perceivable colors.



If we do that for different input stimuli we can in fact get to the same response. That is two physically different colors can be perceived as the same color. So when we want to simulate light and color to be seen by humans it normally doesn‘t make any sense to simulate more than the three base colors red, green and blue.

Mathematics

Stimulus $\sigma(\lambda)$

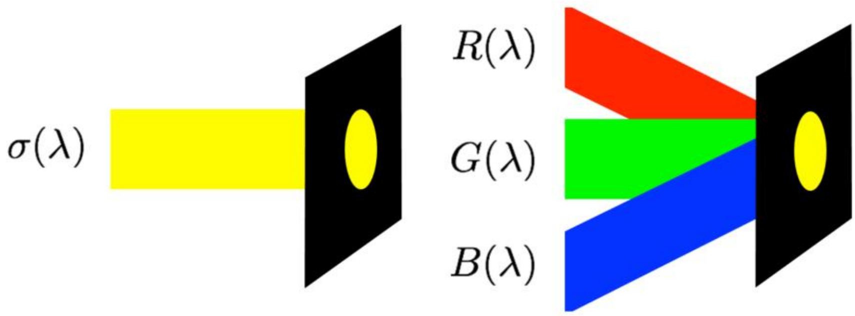
Cone responses $\overline{s}(\lambda), \overline{m}(\lambda), \bar{l}(\lambda)$

$$S(\sigma) = \int \sigma(\lambda)\overline{s}(\lambda)$$
$$M(\sigma) = \int \sigma(\lambda)\overline{m}(\lambda)$$
$$L(\sigma) = \int \sigma(\lambda)\bar{l}(\lambda)$$

Linear map $\sigma \rightarrow (S(\sigma), M(\sigma), L(\sigma))$

The input stimulus is a function of the wavelength. Cone responses are the three cone response functions (s,m,l) for the three different kinds of cones for red, green and blue.
So in the end the input stimulus gets mapped to three values (S,M,L) which are the integrals of the product of the input stimulus and the response functions.

Color Matching



Given a color of pure wavelength (left), a human can mix together amounts of three colors to create the same color impression (right). In this way the response function of a human can be ‚reverse engineered' from the tristimulus on the right.

Mathematics

Stimulus $\sigma(\lambda)$

Primary colors $r(\lambda)g(\lambda)b(\lambda)$

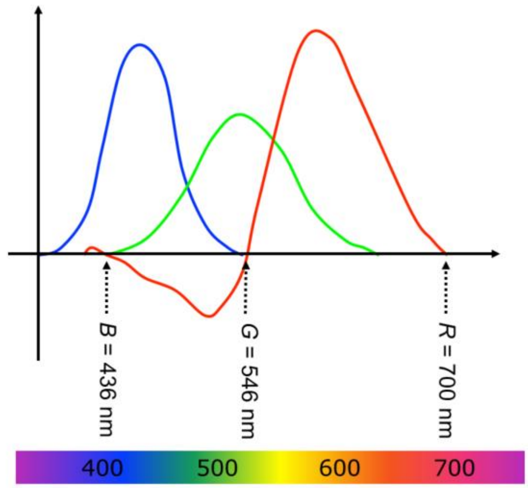
Goal: find tristimulus values $R(\sigma)G(\sigma)B(\sigma)$ that

$$\begin{bmatrix} S(\sigma) \\ M(\sigma) \\ L(\sigma) \end{bmatrix} = \begin{bmatrix} S(R(\sigma)r + G(\sigma)g + B(\sigma)b) \\ M(R(\sigma)r + G(\sigma)g + B(\sigma)b) \\ L(R(\sigma)r + G(\sigma)g + B(\sigma)b) \end{bmatrix}$$

$$\begin{bmatrix} S(\sigma) \\ M(\sigma) \\ L(\sigma) \end{bmatrix} = \begin{bmatrix} S(r) & S(g) & S(b) \\ M(r) & M(g) & M(b) \\ L(r) & L(g) & L(b) \end{bmatrix} \begin{bmatrix} R(\sigma) \\ G(\sigma) \\ B(\sigma) \end{bmatrix}$$

We want to create the same impression as the stimulus sigma(lambda) using the primary colors (of our display).

Pure + mixed colors = all colors ?

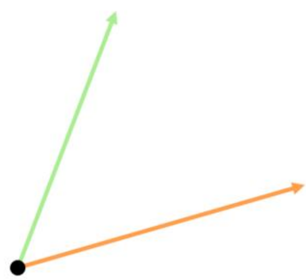


test for each frequency λ_0

$\sigma(\lambda) = \sigma(\lambda - \lambda_0)$

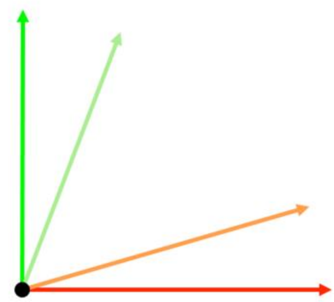
The result of doing those “calibrations” with many test subjects.
To create all possible color impressions of humans, we would need to subtract red light to create certain colors. Of course that‘s not possible so sadly it‘s not possible to simulate all colors this way.
The reason to this is that the red, green, blue colors of the monitor do not exactly produce the same spectrum that our cones react to.

Pure + mixed colors = all colors ?



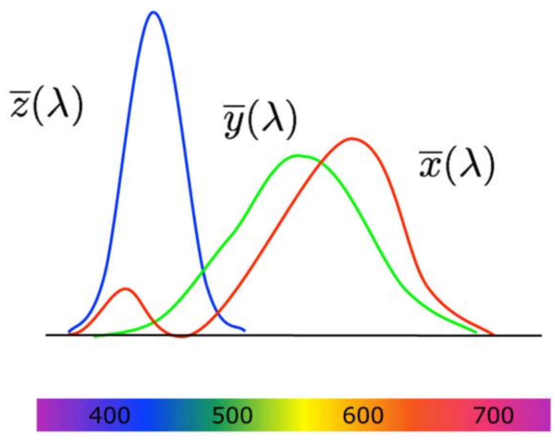
All reproducible colors from two screen-colors span a area in the actual color spectrum. Colors outside of the area on the right top can't be reproduced (but seen by humans) as we are limited to positive values for the vectors (again: no negative light). Here only red and green are shown to keep the example in 2D.

Pure + mixed colors = all colors ?



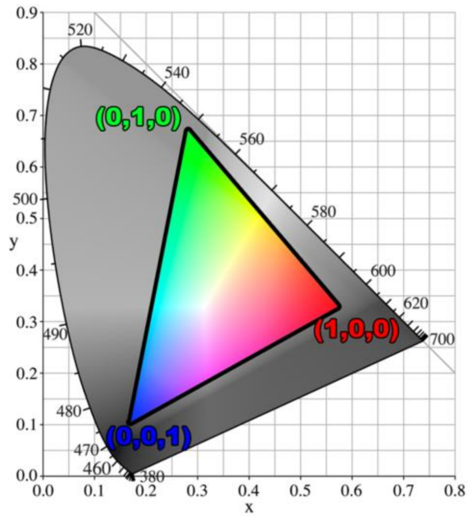
The space seeable by humans shown in darker red/green. Such base colors would be ideal for a monitor, but technically it’s too hard to produce (e.g.) LEDs that produce these kinds of lights.

CIE Model



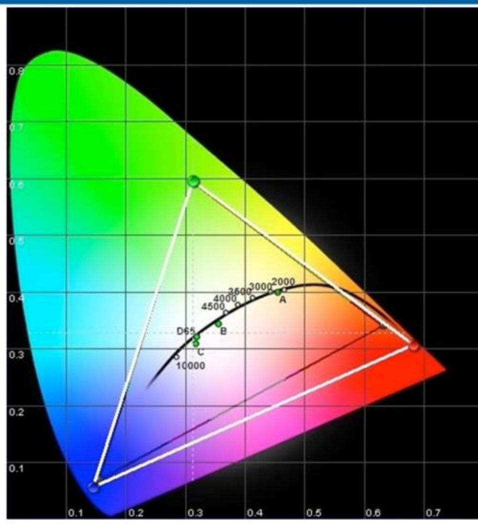
With the theoretical three base colors defined by the CIE we could create all colors (visible by humans). We don't have monitors with such colors but we can use this model to compare the colors which are reproducible by different monitors (or printers etc.).

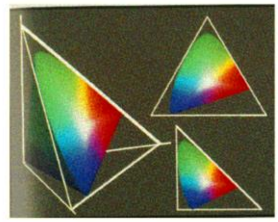
CIE Diagram



The CIE diagram shows the range of fully saturated colors that are visible by humans. The triangles shows the subset that is displayable by a monitor. As colors are defined by red, green and blue components, this spans a triangle. There are colors outside of this triangle (= not reproducible by a PC monitor) that are visible by humans. Each display (and printer in fact) would span a slightly different polygon on this plane. Even when there missing huge parts of the green spectrum, the brain can hallucinate the missing colors so we don‘t recognize that much colors are missing when e.g. watching a movie.

CIE Diagram





$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

The commonly shown version of this diagram shows the visible colors scaled to the whole diagram. Brightness would be an additional dimension (right). The diagram does not show a perfect triangle because not all values would be visible to humans. Pure colors lie around the edges of this diagram. The line in the middle is the ‚white curve‘. All colors on this line can perceived as white (depending on the environment).

CIE Chart and Energy Spectrum

- color temperature / white balance
- pure and saturated colors
- RGB triangle
- non colors

Different tones can be perceived as white and those whites are defined by the light spectrum emitted by an ideal black body when being heated to a certain temperature. Similar to metal which glows at high temperatures, a black body glows in a physical ideal way without reflecting external light. So the color temperature is the temperature in Celvin you would have to heat up an ideal black body to recreate the same color.
Metamerism means that two spectra (S1, S2) can lead to the same color impression. S1 - S2 would result in no stimulus, the spectrum of it can even contain negative values -> these are non colors.

Colors

- Art
- Physics
- Biology
- **Computer Graphics**
  - Gamma Correction
  - Calibration
  - Color Models

Gamma Correction

- Choose **n+1** intensities in [0, 1]
- Eye reacts on intensity **ratios**, not **differences** (50W, 100W, 150W bulb).
- Logarithmic scale:

$$I_0 \neq 0,\ I_n = 1$$
$$I_j = r \times I_{j-1} = r^j \times I_0$$
$$r = \sqrt[n]{1/I_0}$$

- Ratio I_n / I_0 : **dynamic range (contrast)**

I_n is the maximum brightness normalized to 1.

Gamma Correction

- Display intensity I on CRT.
- I does not depend linearly on voltage V

$$I = KV^{\gamma} \Leftrightarrow V = (I/K)^{\gamma}$$

- Find voltage V for intensity I:
  1. $j = round(\log_r(I / I_0))$
  2. $I_j = r^j \cdot I_0$
  3. $V_j = (I_j / K)^{1/\gamma}$

High Dynamic Range Images

Different exposures of the same scene.

High Dynamic Range Images

Multiple images are needed to represent the full dynamic range of the scene. By taking a series of photos with different exposure times the HDR of the scene can be reconstructed.
As the screen still can‘t represent the full dynamic range a new image can be calculated that compresses the full range into the visible range of the screen.

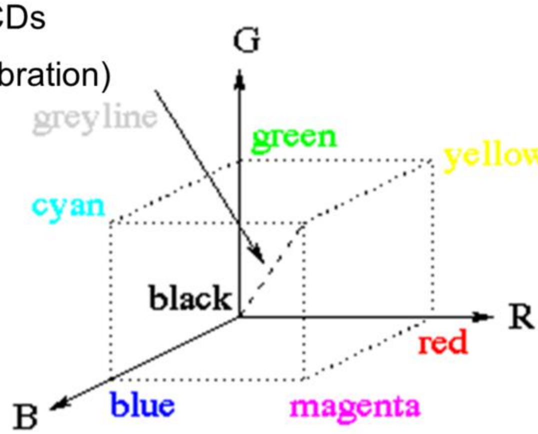
Color Models

- Hardware oriented
  - RGB
  - CMY, CMYK
  - YIQ
- Scientificly founded
  - CIE
  - Lab*
- User oriented
  - HSV
  - HLS

So far we have mentioned a RGB based color model as well as CMYK for printing and CIE XYZ, but there a more than those.

RGB Color Model

- Additive system: black + [r,g,b]
- Color CRTs & LCDs
- CIE → RGB (calibration)



The discussed RGB model is the most common one as it’s close to how the human eyes work and also the ones monitors use.

CIE → RGB

- Problem: $$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
- Given:
  - chromaticity (x_r, y_r), (x_g, y_g), (x_b, y_b) (CIE diagram)
  - white (X_w, Y_w, Z_w) (CIE space)

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

Given are only the x,y values from the calibration but not the X,Y,Z values which also contain the brightness. So we have the colors of RG and B but not the relative brightnesses of them. Given is also the white the user wants to set in X,Y,Z as the mapping from Kelvin to X,Y,Z is well defined in the CIE chart. The problem is to find the maximal R,G,B values the display can use to create the most bright white of the desired white temperature.

CIE → RGB

- Problem: $$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
- Given:
  - chromaticity $(x_r, y_r), (x_g, y_g), (x_b, y_b)$ (CIE diagram)
  - white (X_w, Y_w, Z_w) (CIE space)

$$x_i + y_i + z_i = 1 \qquad z_i = 1 - x_i - y_i$$

As we know that x+y+z=1 we get derive z from x and y.

CIE → RGB

- Problem: $$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$
- Given:
  - chromaticity $(x_r, y_r), (x_g, y_g), (x_b, y_b)$ (CIE diagram)
  - white (X_w, Y_w, Z_w) (CIE space)

- Find luminances $C_i = X_i + Y_i + Z_i, \quad i \in \{r, g, b\}$
  $\Rightarrow X_i = x_i C_i, \quad Y_i = y_i C_i, \quad Z_i = z_i C_i$

CIE → RGB

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{bmatrix} x_r C_r & x_g C_g & x_b C_b \\ y_r C_r & y_g C_g & y_b G_b \\ z_r C_r & z_g C_g & z_b C_b \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = \begin{bmatrix} x_r & x_g & x_b \\ y_r & y_g & y_b \\ z_r & z_g & z_b \end{bmatrix} \begin{bmatrix} C_r \\ C_g \\ C_b \end{bmatrix}$$

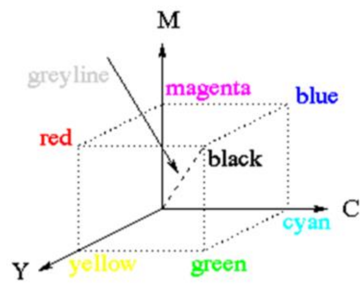
→ Solve for C_r, C_g, C_b

Cr, Cg, Cb are the relative brightness, a change of the whitepoint will change the rel. brightness in return.

CMY(K) Color Model

- Complementary colors to RGB:
  - cyan, magenta, yellow
- Subtractive system for color printers

- Extension: **CMYK**
  - K = min(C,M,Y)
  - C = C - K
  - M = M - K
  - Y = Y - K



Used for printers (that‘s why a 4th color got added, black ink is cheaper than printing 3 colors together and the black ink is normally darker). With perfect cyan, yellow and magenta ink a black ink would not be needed (perfect in color purity and cheap).

YIQ

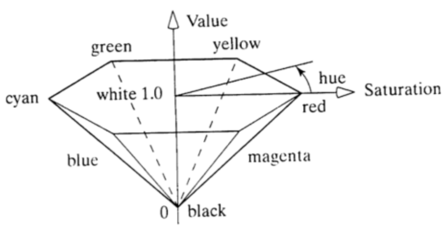
- Additive system for television (B/W)
- Y = brightness (CIE), weighting acc. to spectral response function

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Used for color TV which was compatible to black&white TV: The B/W TV would just use the Y channel.
The Y channel represents the brightness of the colors. Here we see again that blue has a smaller impact to the brightness as red or even green.

HSV Color Model

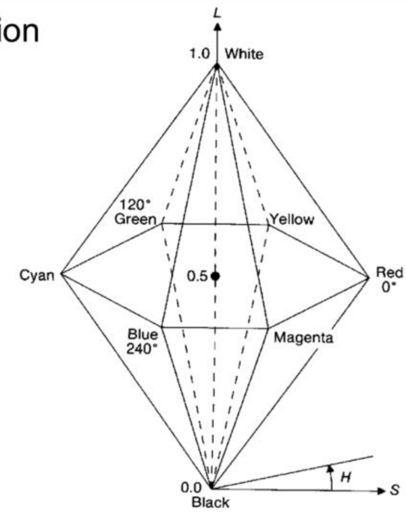
- Hue, Saturation, Value
- User oriented, intuitive
- Projection of RGB cube
- HSV ↔ RGB : Foley



The HSV cone can be interpreted as a deformed version of the RGB-color-cube.
The HSV to RGB conversion is not given here but can be looked up in the literature (e.g. Foley "Computer Graphics: Principles and Practice in C").
This model is more intuitive to humans as it works with Hue, Saturation & Value (brightness) in the same way as we have discussed in the art section at the beginning of this chapter.

HLS Color Model

- Hue, Lightness, Saturation



The pure colors are in the middle outer radius. Towards the bright and dark colors humans can‘t distinguish all different color tones, so the diagram gets more narrow. Such models are not meant to represent the physics of the colors exactly, but to represent the artistic view of colors.

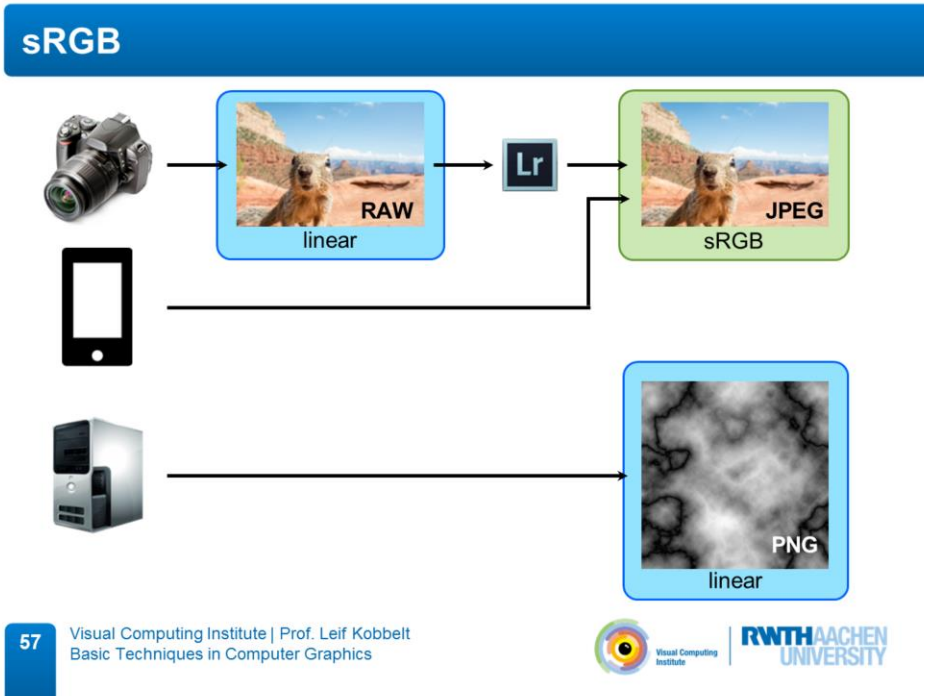
L a*b* Color Space

- CIExyz ... contains all visible colors
- perceptual uniformity
  - color difference measure
  - color table discretization
- conversion

$$\begin{aligned} L &= 116\, f(Y/Y_w) - 16 \\ a* &= 500\,(f(X/X_w) - f(Y/Y_w)) \\ b* &= 200\,(f(y/Y_w) - f(Z/Z_w)) \end{aligned}$$

$$f(t) = \begin{cases} t^{1/3} & t > (\frac{6}{29})^3 \\ \frac{1}{3}\,(\frac{6}{29})^2\, t + \frac{4}{29} & otherwise \end{cases}$$

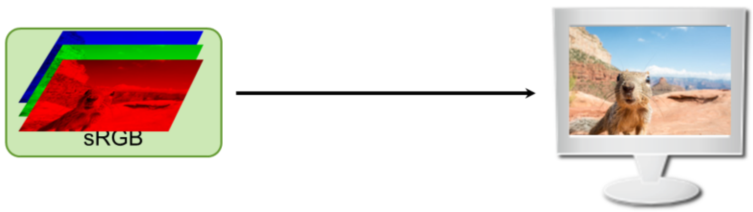
The euclidian distance between two Lab values represent how different the colors are, a property the other models do not have.



Digital cameras capture images linearly (twice the photons result in twice the numerical value read from the sensor) but most image formats store the images in sRGB or another gamma corrected space. If the camera can store the original data from the sensor (RAW format) the linear data can get used but most consumer cameras and smartphones store directly in sRGB and perform the transformation from linear to sRGB internally.

Artificial data like a height map or a normal map is stored in a linear way.

sRGB in OpenGL



The display will always assume an sRGB framebuffer so the last renderpass should be converted to sRGB.

Hint

- Best practice:
  - Store image data in sRGB
  - Store non-image data in linear
  - Store intermediate results in linear
  - Convert to sRGB as late as possible

Storing compressed or 8bit/color image data in sRGB is effectively a compression of the brightness range resulting in finer tone differences in the dark areas and fewer tone differences in the bright areas. This results in less banding, as the human eyes perceive brightness in a non-linear way (similar to the sRGB gamma curve!).
This kind of storage is of course not useful for non-image data.
If the linear and sRGB textures are set up correctly, all texture accesses in a shader will return linear data (as float point so the conversion will be lossless). The lighting calculations will expect all values to be linear!
Intermediate results in applications with multiple render passes (we will discuss those later) are best stored in linear buffers with > 8bit/color, e.g. 16bit floats.
The operating system expects the final result to be in sRGB.

Hint

Lighting algorithms assume linear space!

Phong in gamma space - Phong in linear space

The light source is white, the specular material color is also white. The diffuse material color is based on a green texture. If the lighting is performed on gamma corrected materials (textures) the result will look greenish. If the input material is linearized and the output framebuffer is gamma corrected at the end the highlights will be white (correct rendering).