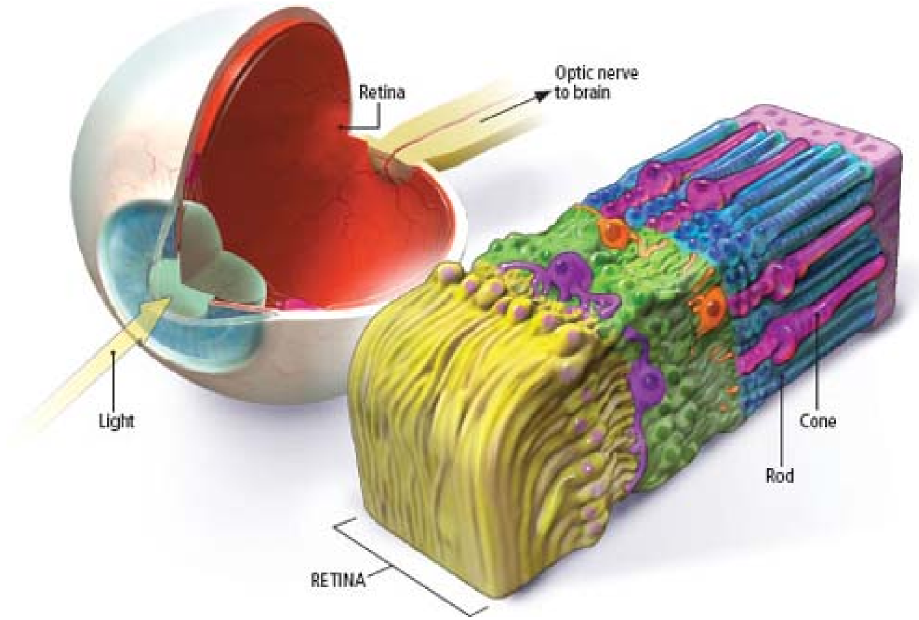
# Color Perception and Representation

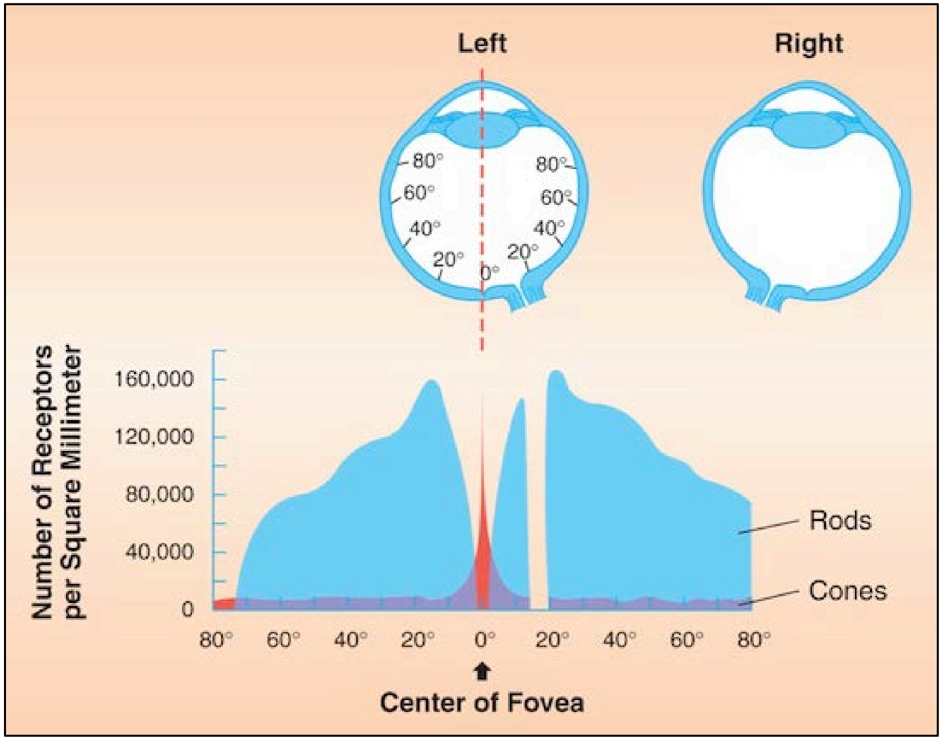
Georgios Papaioannou - 2014

# LIGHT PERCEPTION

# The Human Visual System

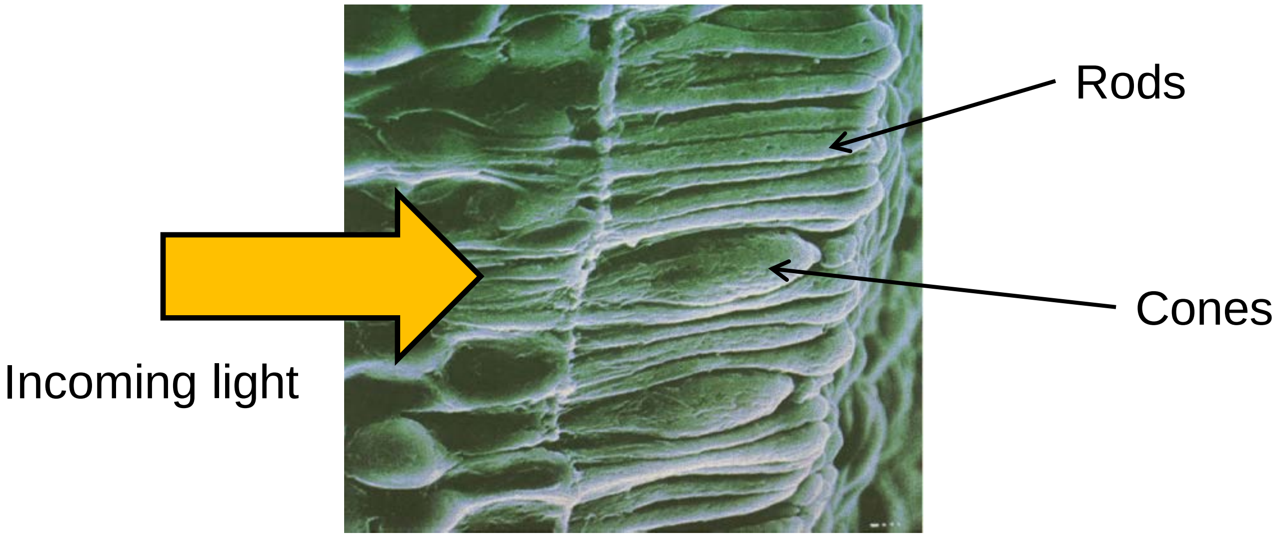
- We perceive light intensity and chromaticity via our photoreceptors: cones and rods



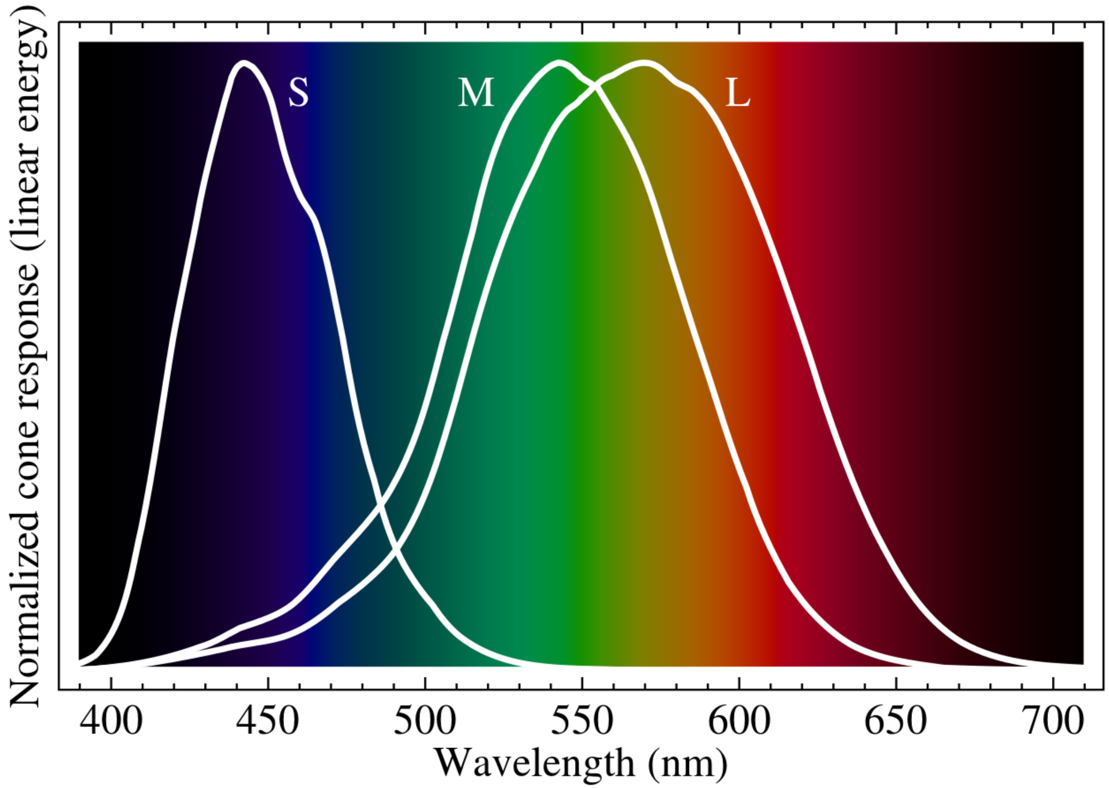
# The Human Visual System – Cones (1)

- Cones primarily responsible for our photopic vision
- They are tuned to specific light wavelengths → responsible for color sensing



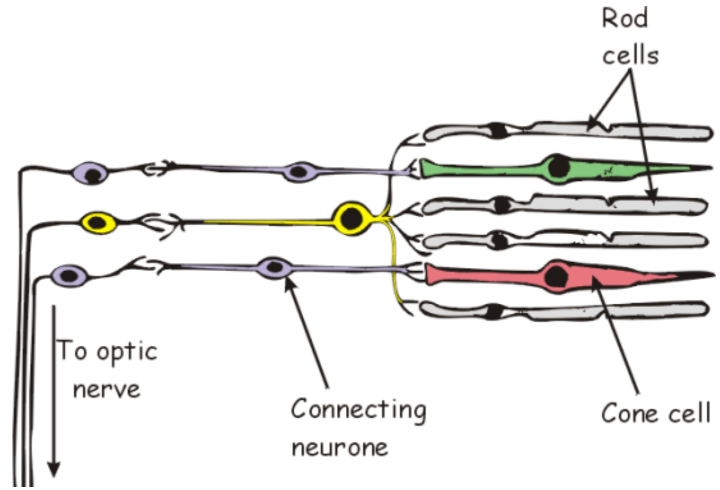
# The Human Visual System – Cones (2)

- Cone wavelength response centered at: blue, green and red



# The Human Visual System – Cones (3)

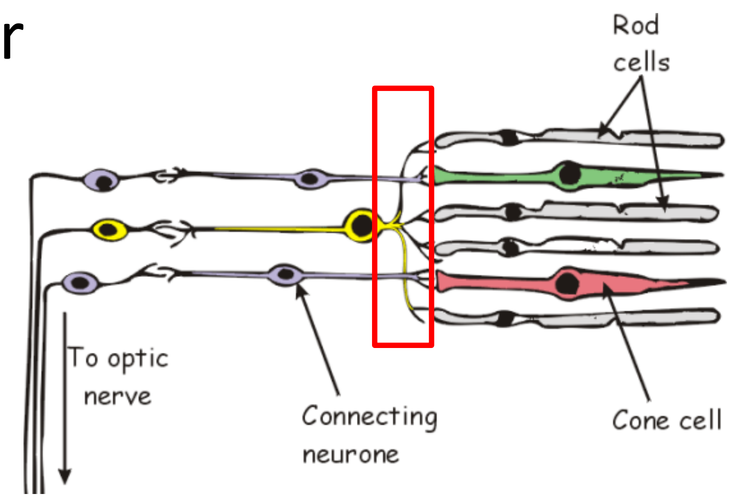
- Cones are tightly packed and dominant near the fovea (center of visual field)
- They better discriminate detail (high frequencies) and temporal changes due to single connectivity to the optic nerve via the retinal ganglion cells



# The Human Visual System – Rods (1)

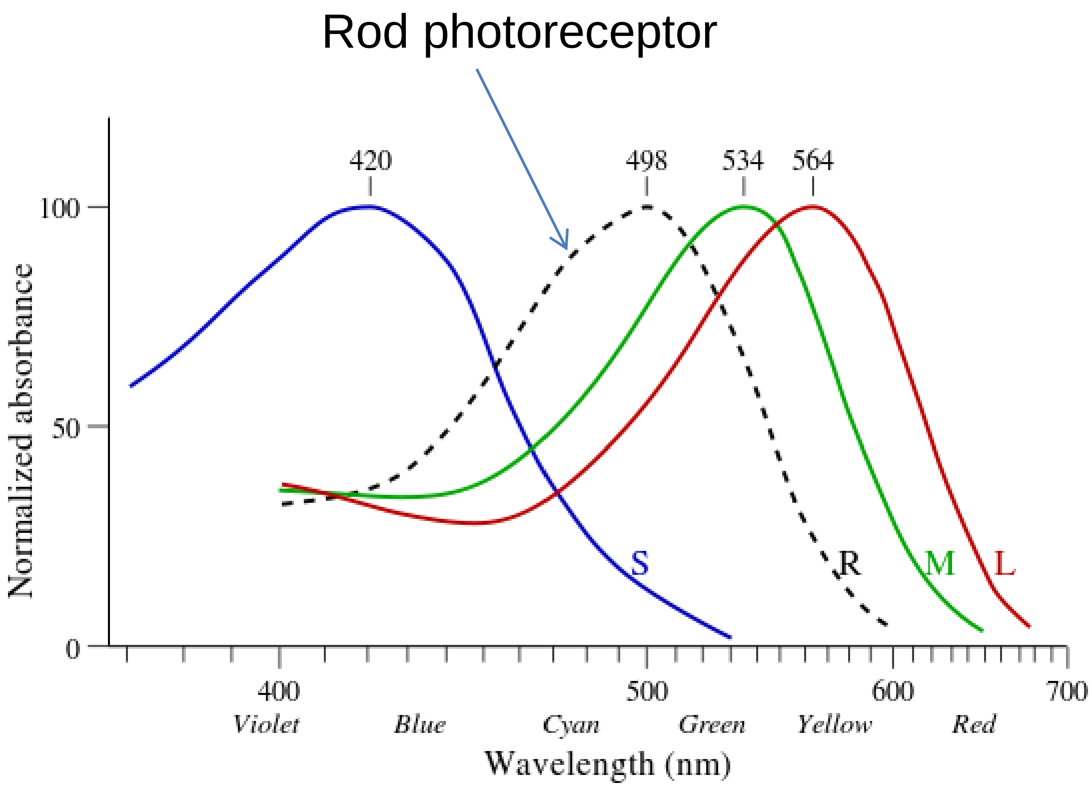
- Rods can function in lower intensity
- → responsible for our scotopic (night) vision
- More concentrated to the outer regions of our field of vision (dominant in peripheral vision)
- Lower visual acuity (detail) due to averaging effect of bunching their signals together

  → low detail in dim light



# The Human Visual System – Rods (2)

- Rod frequency response is centered at bluish-green

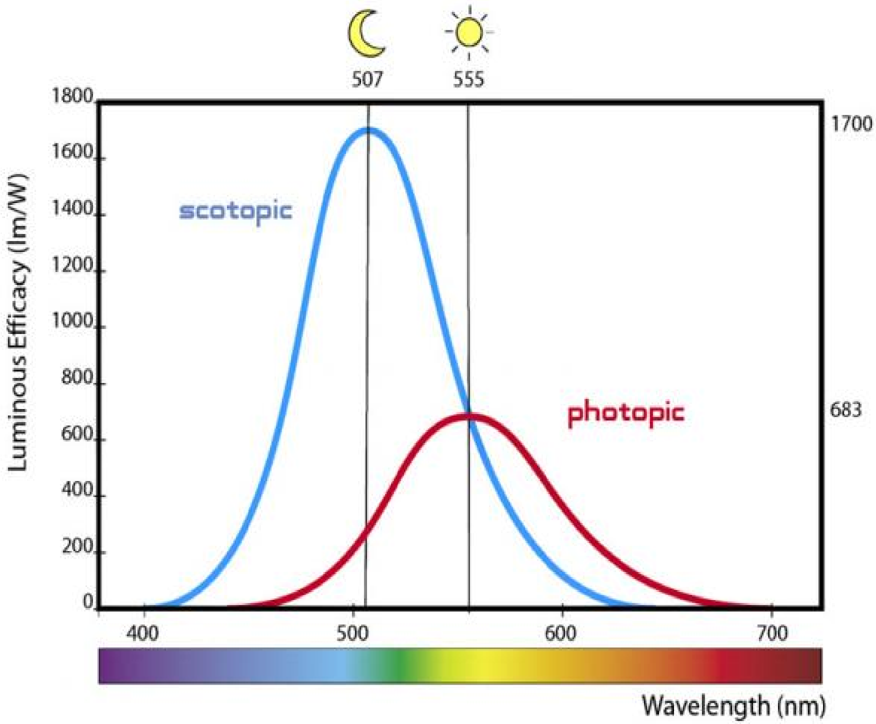


# HVS – Total Color Response

- The cumulative effect of all receptors combined is a frequency response mainly centered at green hues

→ We can better discriminate shades and intensity values of green

- Why?



# Perceived Brightness (1)

- Perceived light ≠ actual incident light

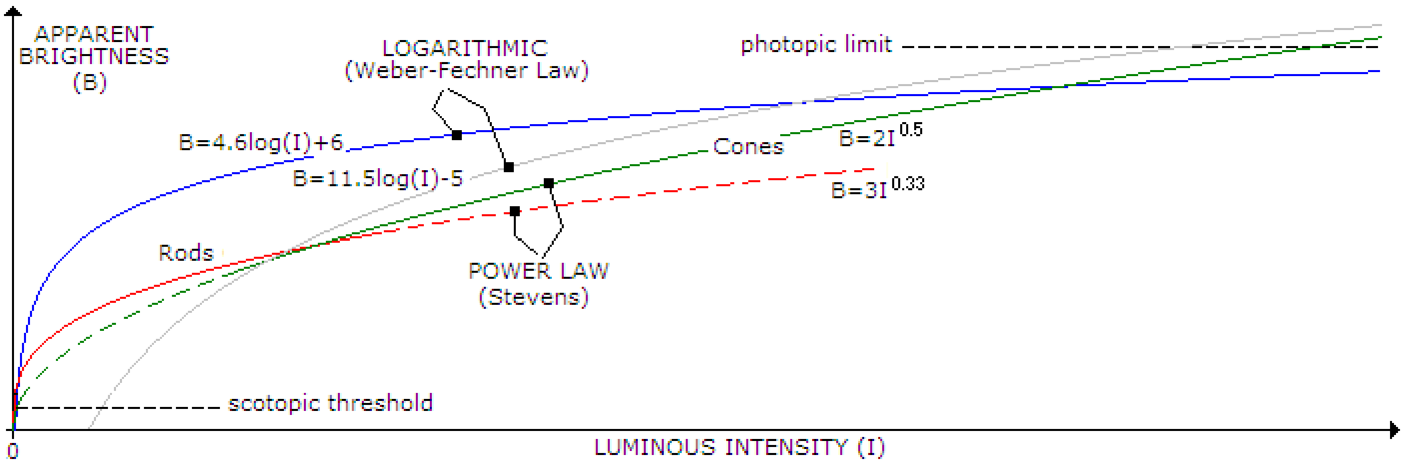
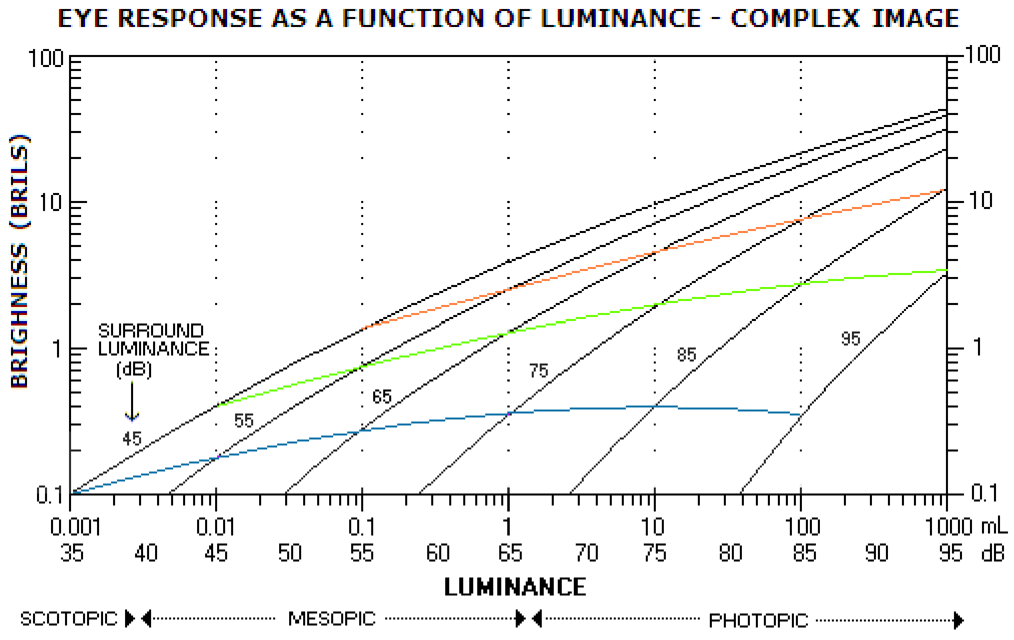
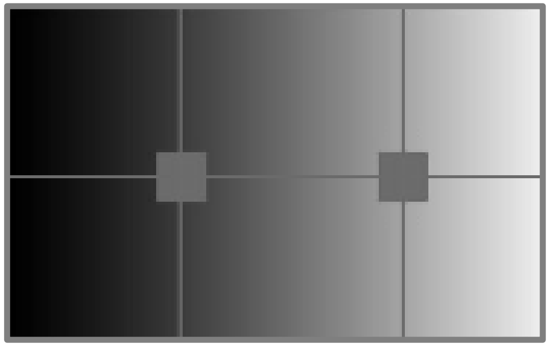


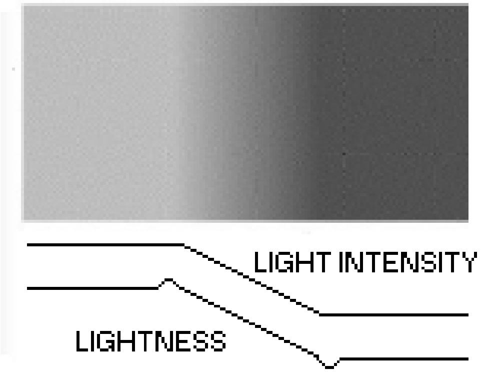

Diagram refers to steady background illumination

# Perceived Brightness (2)

- Perceived brightness is affected by background level
- Brighter background → Darker hotspot brightness



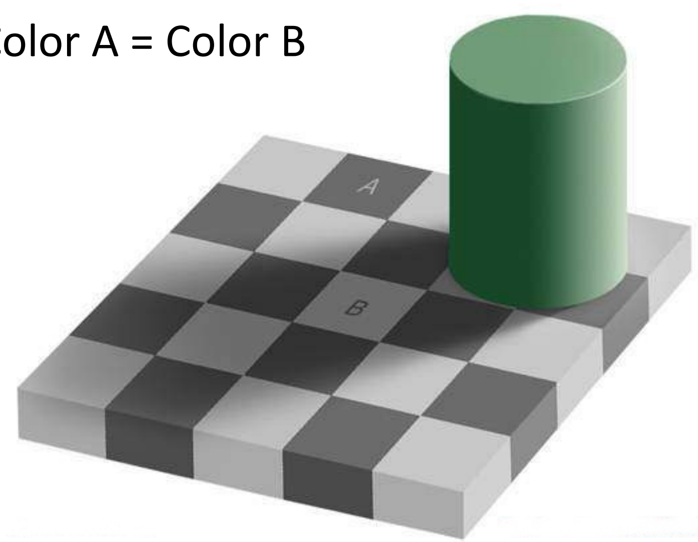



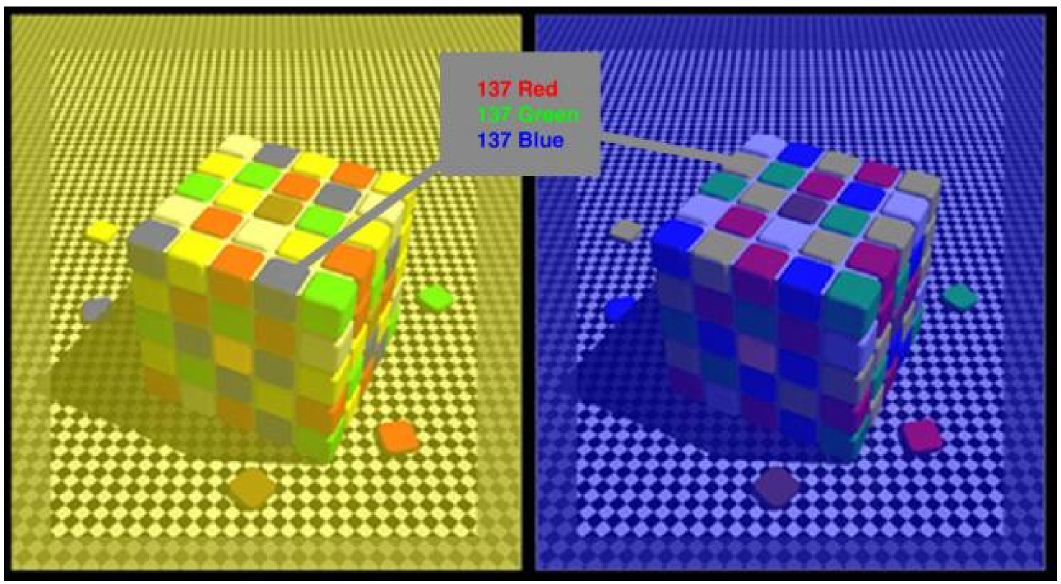


# Perception of Contrast

- HVS is not good at interpreting absolute color values
- It is driven by contrast differences
- Color and shape discrimination relies on contrast
- Many visual illusions are based on the above

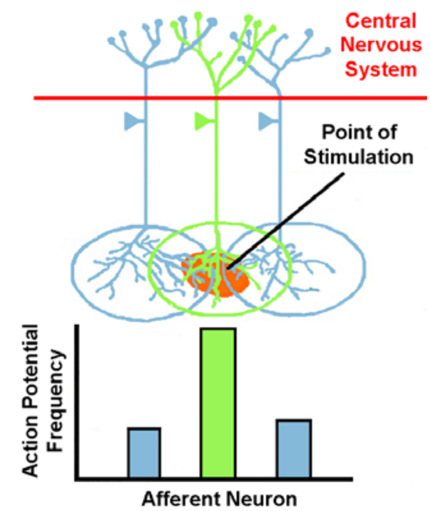
Color A = Color B

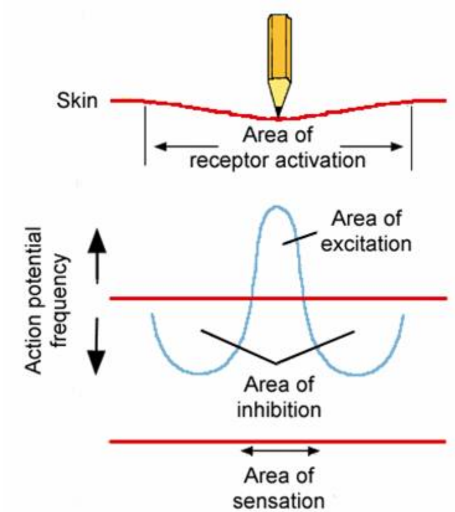


# Why do we Accentuate Contrast?

- The area of sensation is less than the area of receptor activation, due to the areas of inhibition that flank the center of stimulus
- Activation of the central neuron negatively affects the action potential frequency of the flanking neurons → locally increasing stimulation contrast



# Dynamic Range

- Dynamic range: the minimum to maximum luminance level achieved by a system
  - Dynamic → adaptive
- The human visual system adapts to the level of illumination incident to the photoreceptors
  - Rods (scotoptic light): $10^{-6}cd/m^2 – 10cd/m^2$
  - Cones (photoptic light): $10^{-2}cd/m^2 – 10^8 cd/m^2$
- Total luminance range: $10^8:10^{-6}$
- Cannot achieve these levels simultaneously!

# Dynamic Range Example

Cannot correctly visualize the entire linear luminance scale simultaneously

# COLOR REPRESENTATION

# Color Representation (1)

- Color is represented via a color model
- A color model is a mathematical mapping of the spectrum of visible light (by the HVS) to a set of components
- Color models can represent either the perceived color or the stimulus (produced light)
- Remember: Perceived light $\neq$ actual incident light

# Color Representation (2)

- We need color models to:
  - Describe
  - Compare
  - Order
  - Classify

  colors

# Color Representation (3)

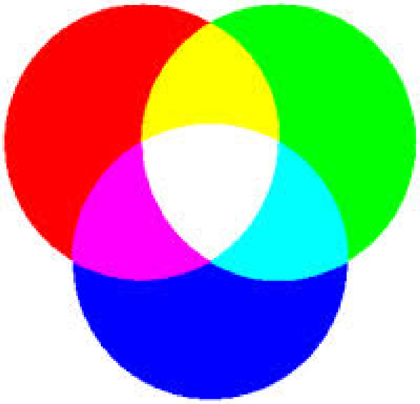
- Each color model defines a color space, i.e. the range of valid values for each component
- Some color spaces are bounded, others allow only positive values etc.
- The coverage of a particular color space by a certain device or sensor (generally, a system) is its color gamut

# Color Model Classification (1)

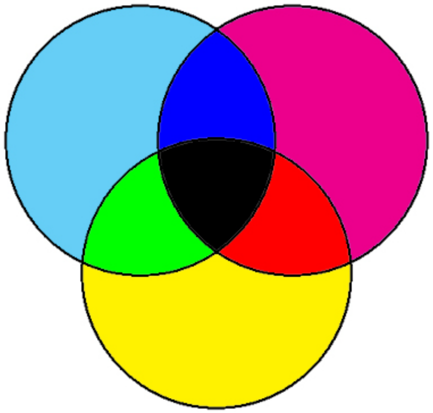
- Device-independent
  - The coordinates (components) of a color will represent a unique color value, according to human perception
  - Useful, among other things, for the consistent conversion between device-dependent color models
- Device-dependent
  - The same color coordinates will produce a slightly different visible color value on different display devices or media

# Color Model Classification (2)

- Additive models encapsulate the way color is produced on a computer display by *adding* the contributions of the primaries



- Subtractive models resemble the working of a painter or a printer, where color mixing is achieved through a *subtractive* (filtering / painting) process.

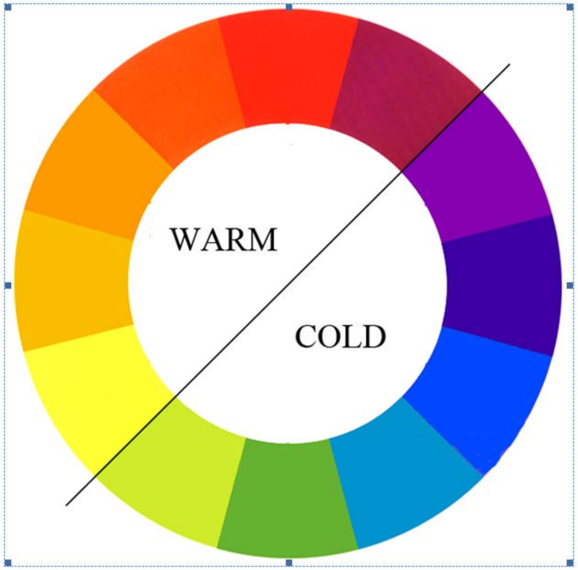


# Color Spaces

- Color models define the primary components (primaries) that form a basis for representing all other colors
- Primaries are a basis for this space:
  - No primary can be produced as a linear combination of the other
  - Addition and linear mixing are always well-defined in a color space
  - Linear operations in a color space are not necessarily perceptually linear!

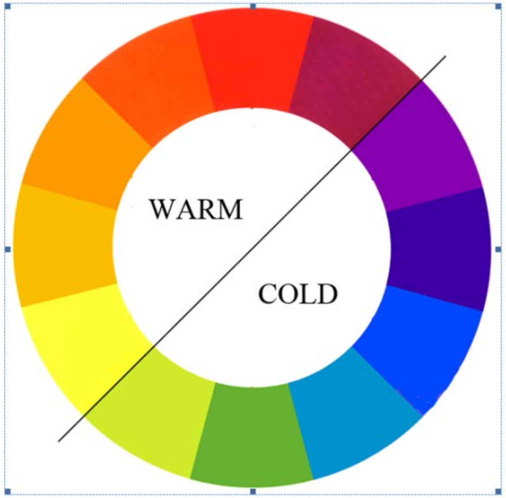
# Hue

- Hue defines which color in the range of available tones a signal represents
- It is typically represented in a circular arrangement, not as wavelength but rather as the color mixing result



# Warm and Cool Colors

- The categorization of hues to warm and cold (cool) colors is a psychological mapping of hue to certain events and emotional states
  - This can be useful in visualization, to convey the appropriate meaning for visualized information



# Color Models – RGB (1)

- Device-dependent
- Color images are typically stored as RGB (red, green, blue) triplets per pixel
- RGB values match our tri-stimulus vision
- Displays emit light in 3 separate RGB components
- The RGB model represents the generated flux and is therefore linear with regard to the emitted light at the source
- RGB is not perceptually linear

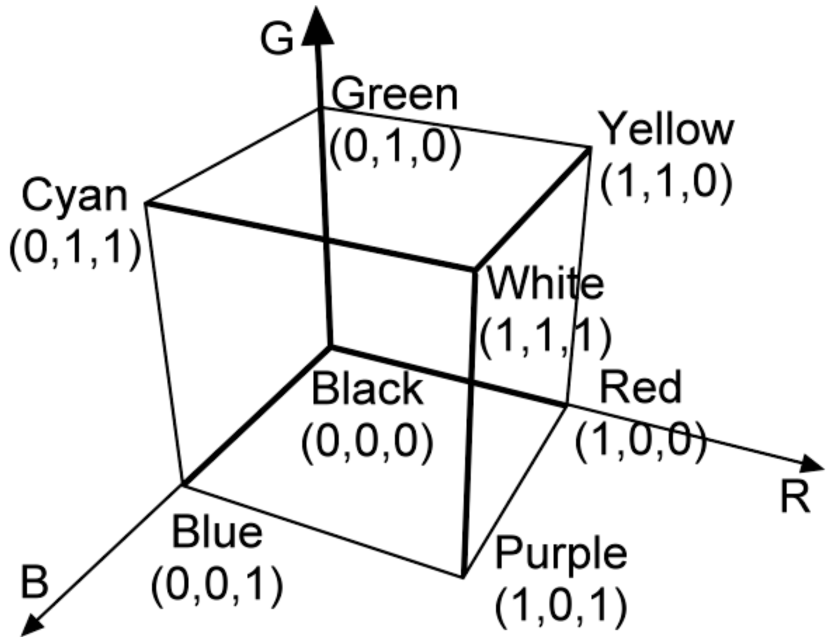
# Color Models – RGB (2)

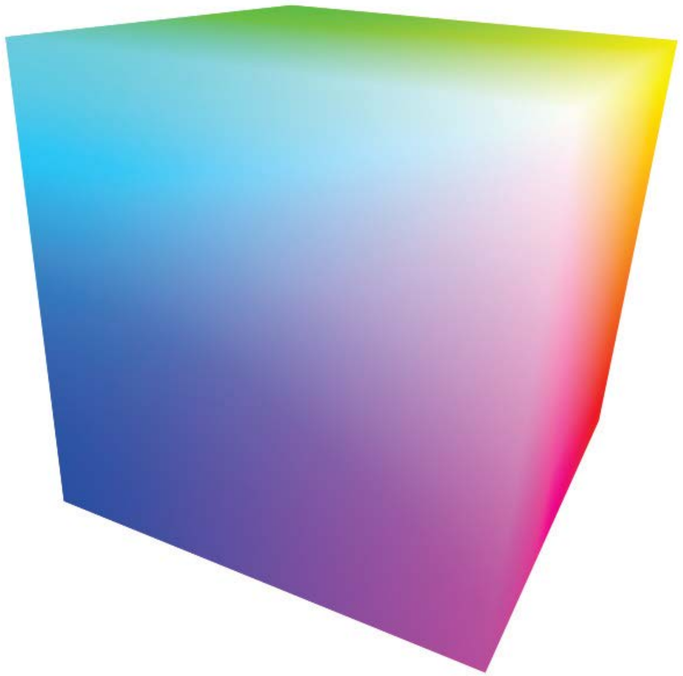
- Typical model for:
  - Storing color information in images
  - Keeping color information in memory buffers
  - Display systems (active)

- Usually, a bounded (normalized maximum) range of values is represented and stored
- Floating-point arithmetic representation allows also to store virtually any value for RGB components

# RGB Model – RGB Color Cube

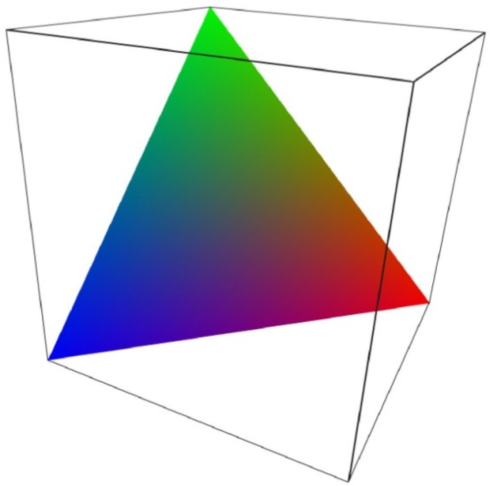
- The 3 primary colors (RGB) form a basis for the RGB color space
- Unit values form the RGB color cube



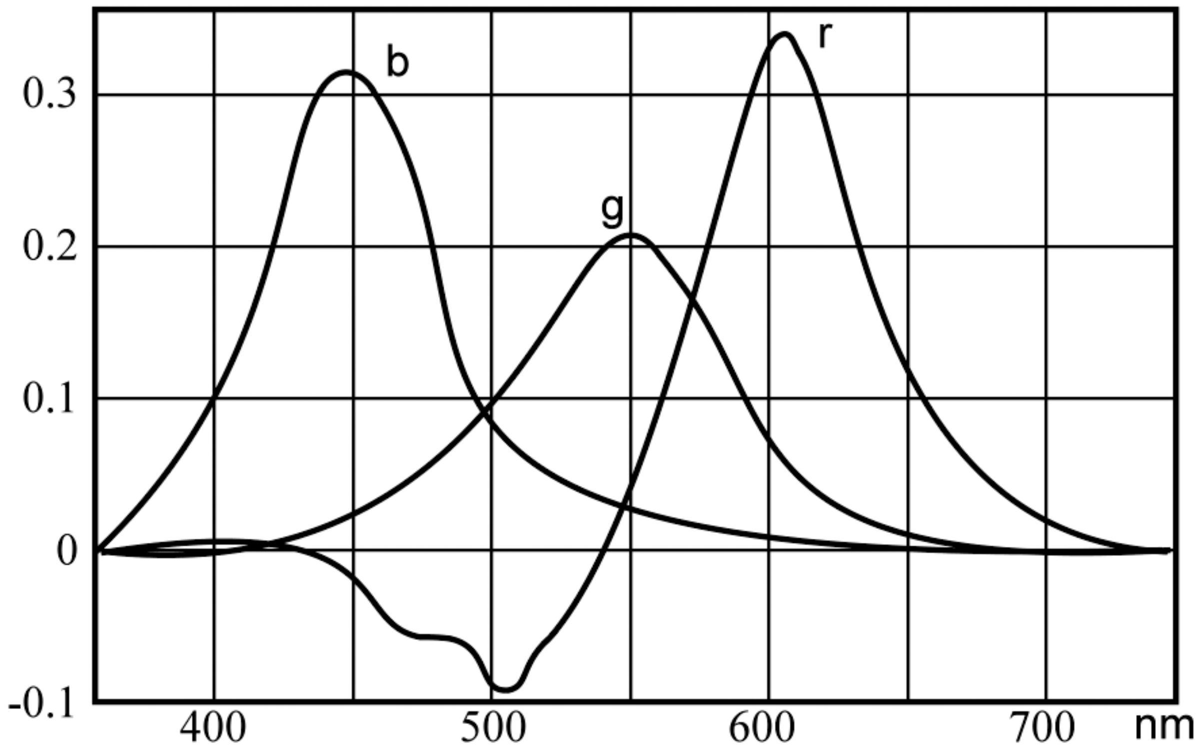



# RGB Model – RGB Triangle

- Joining the 3 primaries we obtain the RGB triangle
- Hue (different color) is represented at the perimeter of the triangle
- Saturation is increased off center (towards the edges) and neutralized (gray) towards the center



# Color Models – RGB (3)



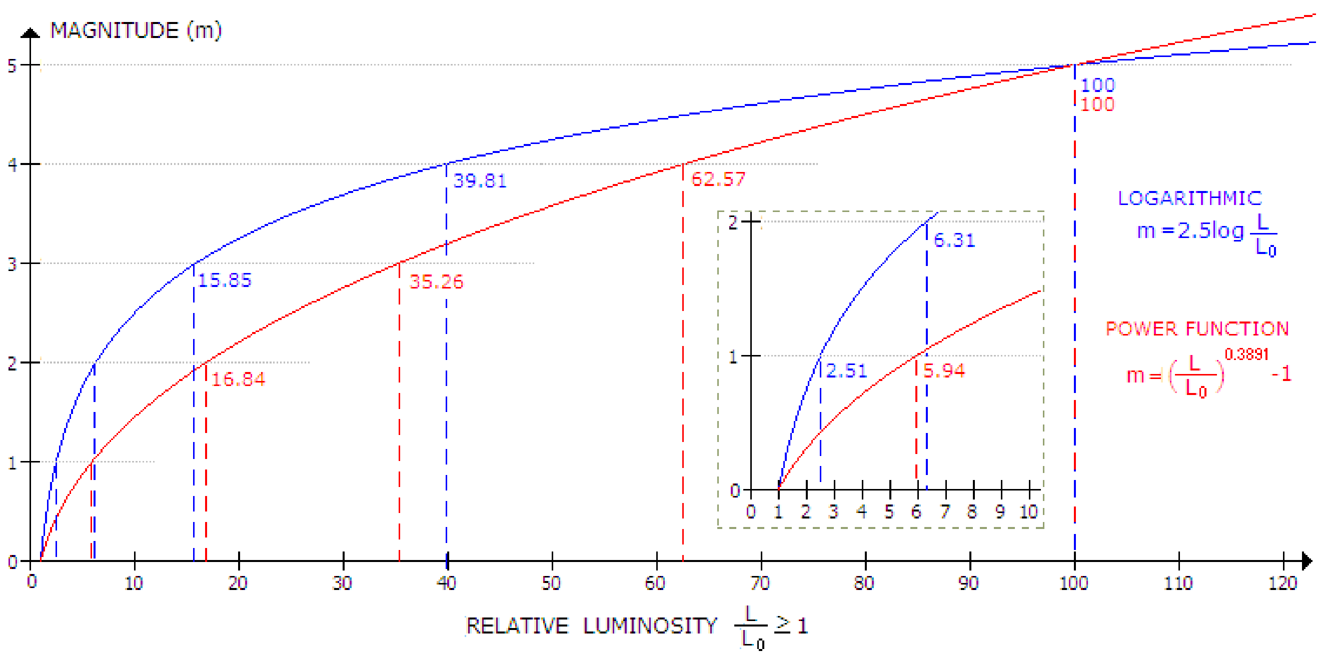

The mixing curves to produce a particular single wavelength in the HVS sensors.
The RGB model is not linear.

# Gamma Correction (1)

- As explained, eye response to light intensity (*brightness)* is not linear. Rather, it is well approximated by a power function of light intensity, and in many cases it can be also described as logarithmic

# Gamma Correction (2)

- i.e.: Brightness is determined by the change of incident flux relative to an initial flux and not the nominal change

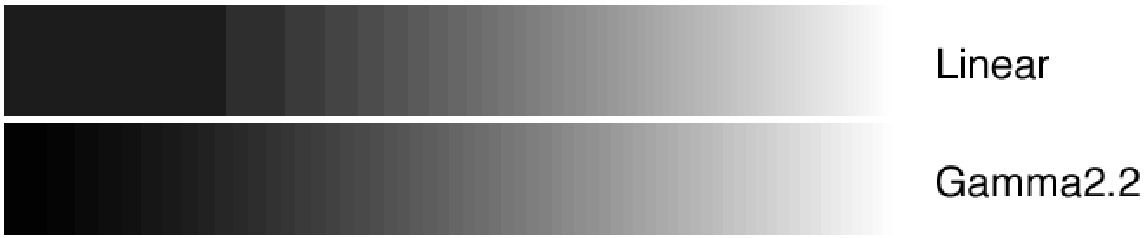


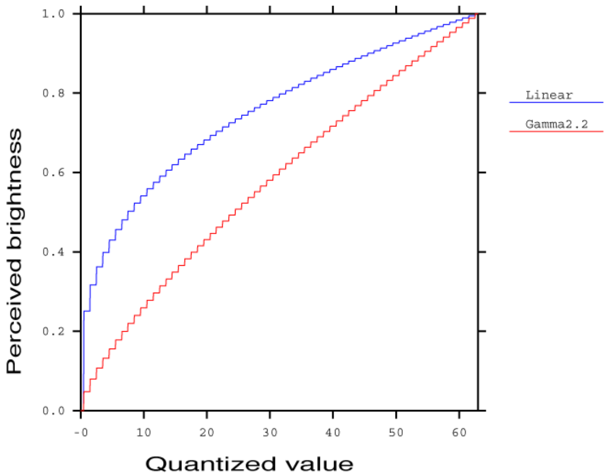
# Gamma Correction (3)

- Photography and computer-generated images capture light in linear luminance space (i.e. as received by the “sensor”) →
- But: we perceive these values non-linearly

# Gamma Correction (4)

- If we convert the linear range to a fixed-quantization representation (e.g. 24bit integer RGB representation):
  - We discriminate dim color transitions (images appear quantized)
  - We fail to differentiate bright differences (waste of bits)



# Gamma Correction (5)

- Gamma correction transforms the linear luminance according to a power law (our perception response)
- And then stores the encoded results
  - This results in sufficient quantized values being allocated for all brightness levels
- The reverse process is performed during the display of the image

$$v_{out} = {v_{in}}^{\gamma}$$

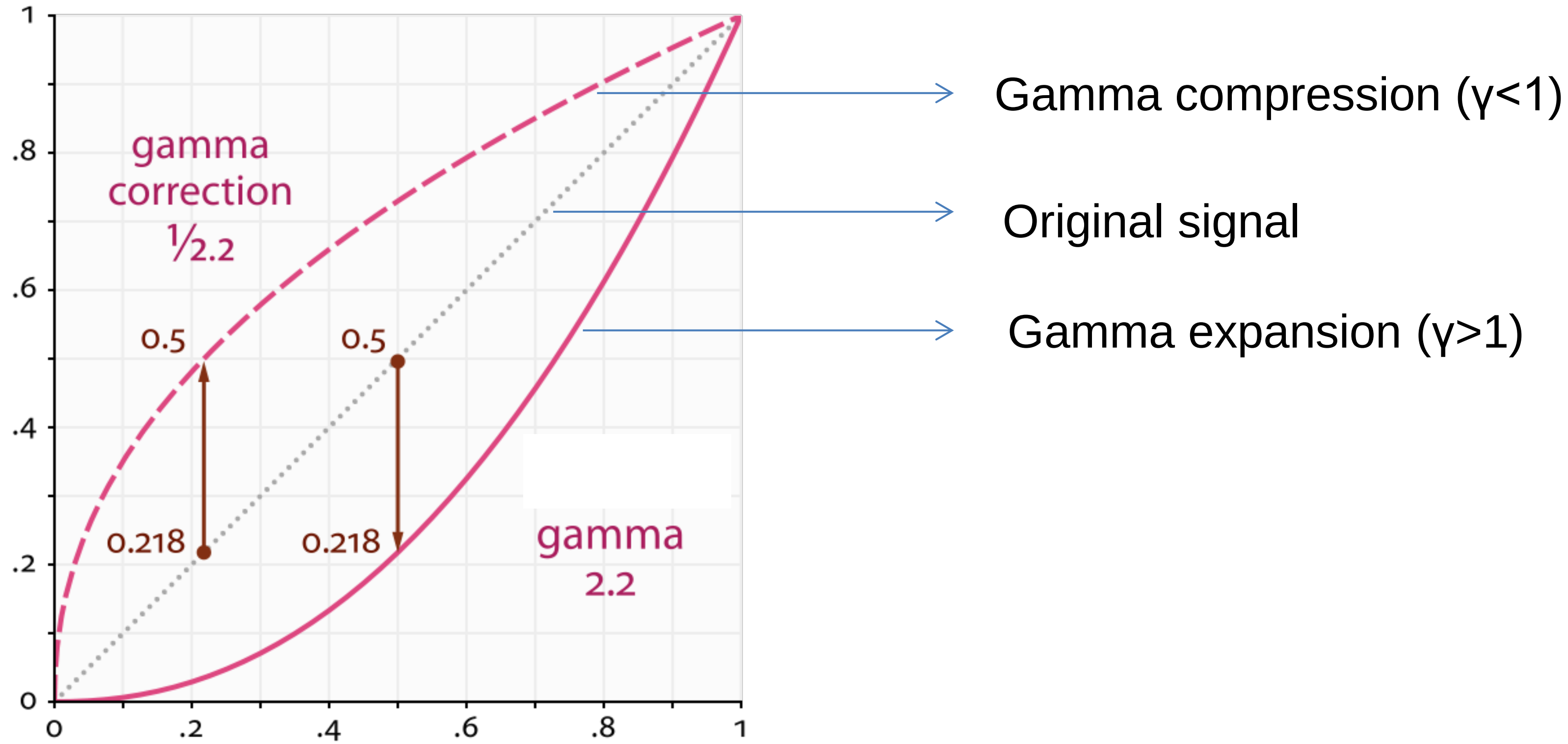
# Gamma Correction (6)

- Gamma correction transforms the linear luminance according to a power law (our perception response)
- And then stores the encoded results
  - This results in sufficient quantized values being allocated for all brightness levels
- The reverse process is performed during the display of the image

$$v_{out} = {v_{in}}^{\gamma}$$

# Gamma Correction (7)

$$v_{out} = {v_{in}}^{\gamma}$$

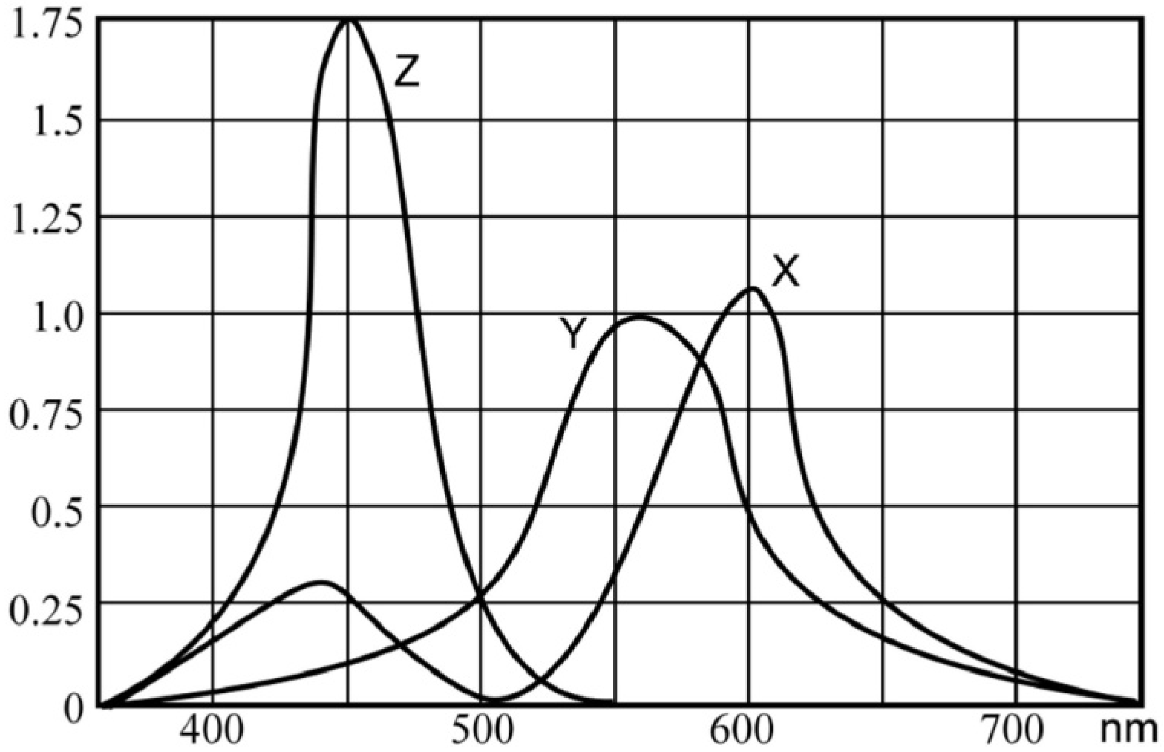


# The XYZ Model (1)

- Device-independent
- Not perceptually linear
- Quantifies luminance (Y) and chromaticity (X and Z coordinates)
- XYZ coordinates are not primary colors, rather computational quantities

# The XYZ Model (2)

- Mixing XYZ values produces visible colors:



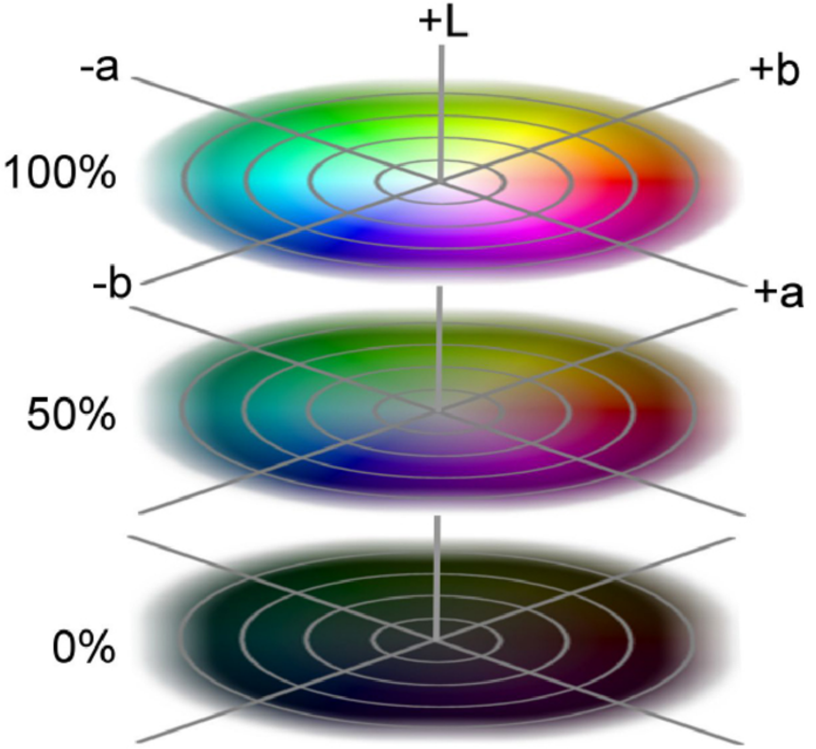
# Conversion to/from RGB

- XYZ model can be used to convert an RGB color between two devices
- Each device specifies an invertible conversion matrix $\mathbf{M}$ from RGB to XYZ
- Then, given $(r_1, g_1, b_1)$ of device with $\mathbf{M}_1$, we convert to RGB of a device with $\mathbf{M}_2$ with:

- $(r_2, g_2, b_2) = \mathbf{M}_2^{-1}\mathbf{M}_1(r_1, g_1, b_1)$

# The CIE L*a*b* Model (1)

- Similar to XYZ, separates luminance (L* here) from chromaticity (a*,b*), but
- It is perceptually linear
- It is defined w.r.t. the white point of a given device
- a* axis: green-magenta
- b* axis: blue-yellow



# White Point

- The color that is displayed when all color components take their max value
- Usually when r = g = b = 1 (normalized max)
- Is expressed in CIE XYZ as $(X_n, Y_n, Z_n)$

# The CIE L*a*b* Model (2)

- The coefficients of the L*a*b* color model are defined w.r.t. the XYZ coordinates and the white point as (reversible transformation):

$$L^* = \begin{cases} 116\sqrt[3]{Y_r} - 16, & if \quad Y_r > 0.008856, \\ 903.3Y_r, & if \quad Y_r \leq 0.008856, \end{cases}$$

$$a^* = 500(f(X_r) - f(Y_r))$$

$$b^* = 200(f(Y_r) - f(Z_r))$$

$$X_r = \frac{X}{X_n} \quad Y_r = \frac{Y}{Y_n} \quad Z_r = \frac{Z}{Z_n},$$

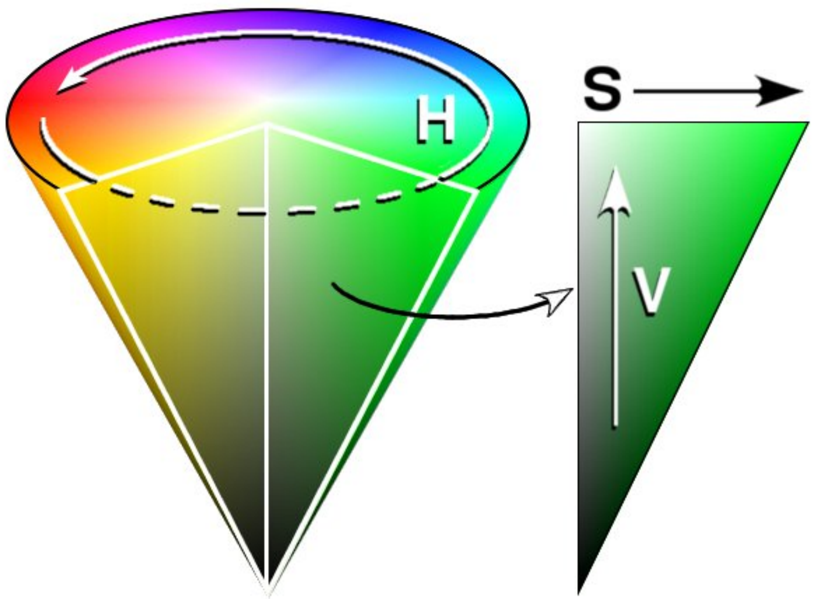
$$f(t) = \begin{cases} \sqrt[3]{t}, & if \;\; t > 0.008856 \\ 7.787t + 16/116, & if \;\; t \leq 0.008856 \end{cases}$$

# The HSV Model (1)

- RGB, XYZ and L*a*b* color models are not intuitive to work with (i.e. to specify a desired color)
- The HSV model attempts a more human-centric color definition approach:
  - (H)ue specifies what the color is
  - (S)aturation specifies how intense the coloration is (as opposed to muted / gray)
  - (V)alue specifies the color's produced intensity
- Alternatively: HSB, (B)rightness being the respective perceived light intensity

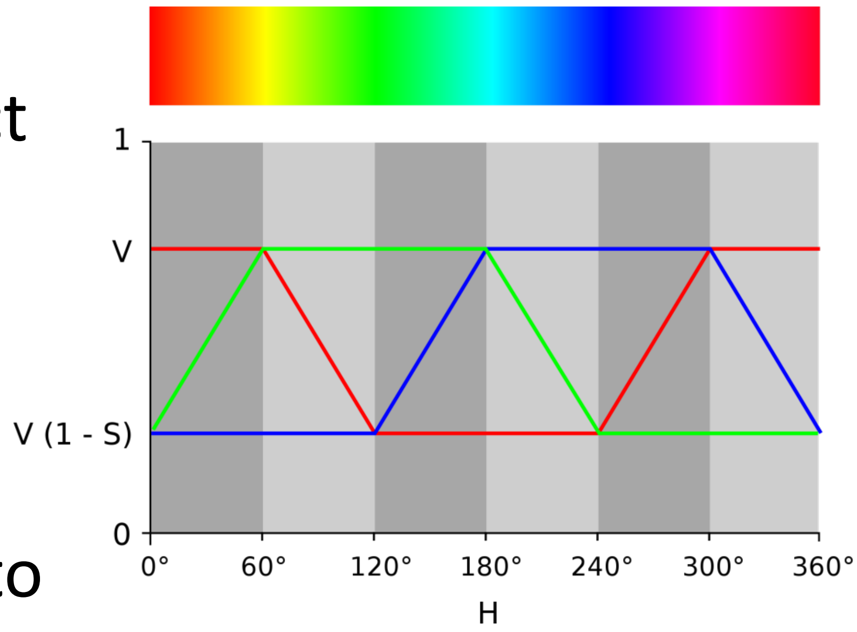
# The HSV Model (2)

- It is common to specify a color based on the above characteristics
- Colors are geometrically represented on a cone



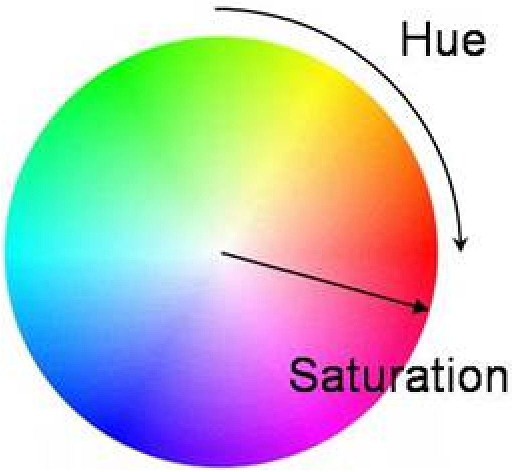
# The HSV Model - Hue

- Colors arranged on a circle (like a color wheel)
- Hue is the angle with respect to an initial position on the circle
  - E.g. red is at 0°, green is at 120°, blue is at 240°
- The hue circle corresponds to a cross section of the cone



# The HSV Model - Saturation

- Is max on the surface of the cone (minus the base) → represents pure colors with maximum “colorfulness”
- The axis of the cone represents the min saturation (shades of gray)



# The HSV Model - Value

- Corresponds to intensity
- Min value (0) : absence of light (black)
- Max value:  the color has its peak intensity
- Is represented along the axis of the cone:
  - 0 : the cone's apex
  - Max value : the center of the cone's base

# HSV to RGB

$$C = V \times S_{HSV}$$

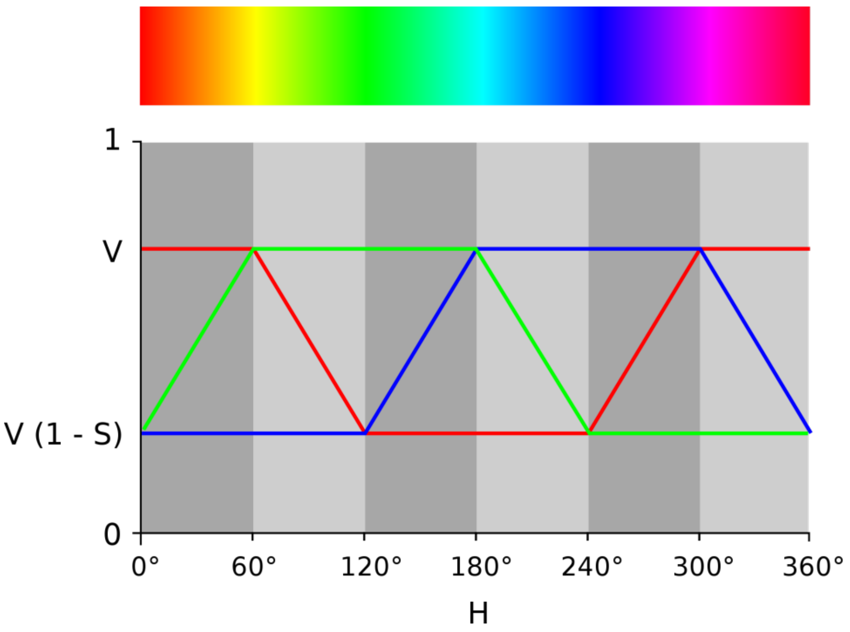
$$H' = \frac{H}{60^\circ}$$

$$X = C(1 - |H' \bmod 2 - 1|)$$

$$(R_1, G_1, B_1) = \begin{cases} (0,0,0) & \text{if } H \text{ is undefined} \\ (C, X, 0) & \text{if } 0 \le H' < 1 \\ (X, C, 0) & \text{if } 1 \le H' < 2 \\ (0, C, X) & \text{if } 2 \le H' < 3 \\ (0, X, C) & \text{if } 3 \le H' < 4 \\ (X, 0, C) & \text{if } 4 \le H' < 5 \\ (C, 0, X) & \text{if } 5 \le H' < 6 \end{cases}$$

$$m = V - C$$

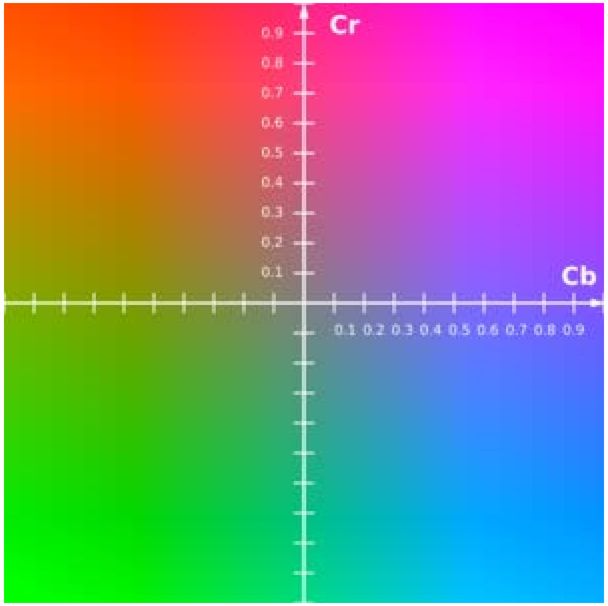
$$(R, G, B) = (R_1 + m, G_1 + m, B_1 + m)$$



# The YCbCr Model (1)

- Heavily used in video and digital photography.
- Y′ is the luma component and $C_B$ and $C_R$ are the blue-difference and red-difference chroma components.

Chroma plane




<u>Note:</u>
Y′ (with prime) is distinguished from Y (luminance), as light intensity is encoded using gamma corrected RGB primaries

# The YCbCr Model (2)

- Conversion from RGB:

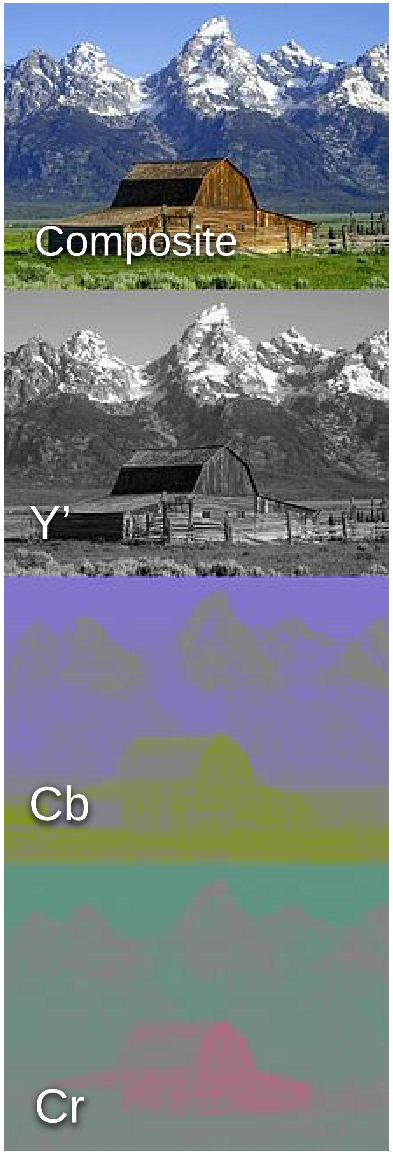
$$Y' = K_R \cdot R' + (1 - K_R - K_B) \cdot G' + K_B \cdot B'$$

$$P_B = \frac{1}{2} \cdot \frac{B' - Y'}{1 - K_B}$$

$$P_R = \frac{1}{2} \cdot \frac{R' - Y'}{1 - K_R}$$

where:

- $P_B$ and $P_R$ are the “analog” color offsets before adjustment for integer representation)
- $K_R$ and $K_B$ are determined by the color matrix for a particular device or format



# The YCbCr Model (3)

- Example - YCbCr in the JPEG format:

$$
\begin{aligned}
Y' &= \quad 0 + (0.299 & \cdot R'_D) + (0.587 & \cdot G'_D) + (0.114 & \cdot B'_D) \\
C_B &= 128 - (0.168736 & \cdot R'_D) - (0.331264 & \cdot G'_D) + (0.5 & \cdot B'_D) \\
C_R &= 128 + (0.5 & \cdot R'_D) - (0.418688 & \cdot G'_D) - (0.081312 & \cdot B'_D)
\end{aligned}
$$

$$
\begin{aligned}
R &= Y & & + 1.402 & \cdot (C_R - 128) \\
G &= Y - 0.34414 & \cdot (C_B - 128) & - 0.71414 & \cdot (C_R - 128) \\
B &= Y + 1.772 & \cdot (C_B - 128) & &
\end{aligned}
$$

# Color-space Compression

- Why use a luma-chroma model?
  - It allows the efficient compression of image information in a perceptually optimal manner
- The HVS luminance visual acuity is greater than the discrimination of chrominance variations (why?)
- → We can subsample the

# Chroma Subsampling Example (1)

Original image: luma/chroma subsampling ratio = 1:1

# Chroma Subsampling Example (2)

Compressed image: luma/chroma subsampling ratio = 1:16

# Chroma Subsampling – Example 2

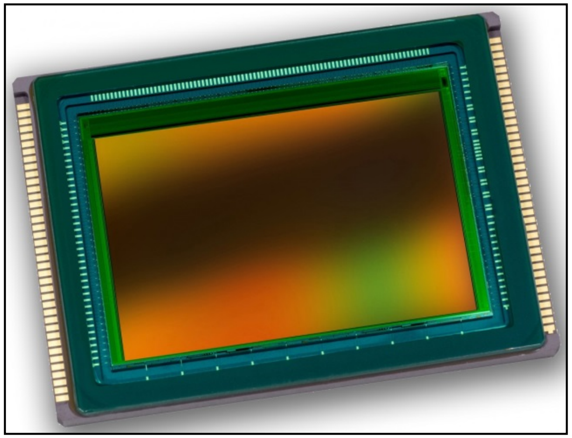
Original

Subsampled chroma

Subsampling causes color bleeding and desaturation in high chroma contrast areas

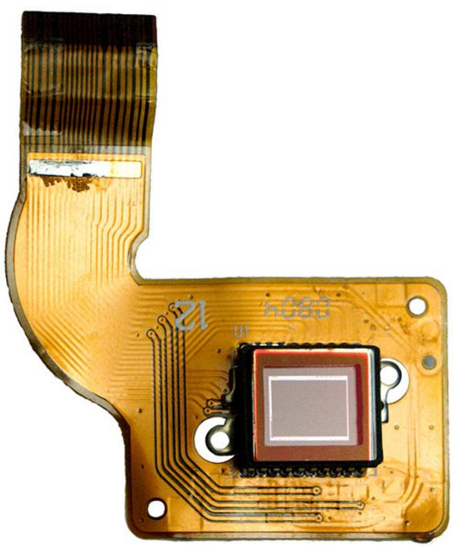
# Digital Photography Sensors

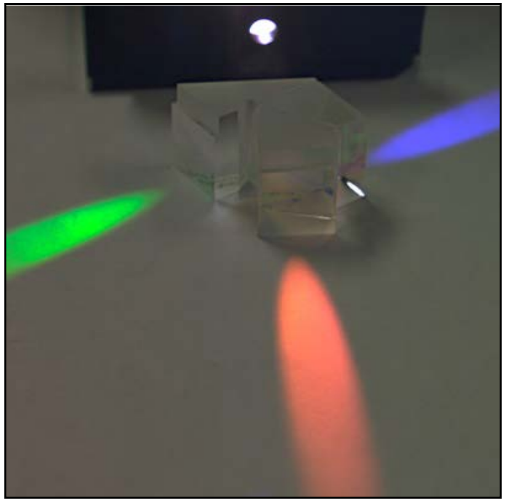
- Digital camera sensors produce a voltage for each “sensed” pixel cell on their sensor array
  - This signal is further digitized
- Technologies: CCD and CMOS devices
  - They provide more or less the same quality
  - Relatively linear response to incident light



# Digital Photography and Color

- Sensors cannot inherently separate color!
- Strategies:
  - Color filter arrays → Typical camera
  - X3 sensor arrays + prism → Bulky construction: high-end video cameras





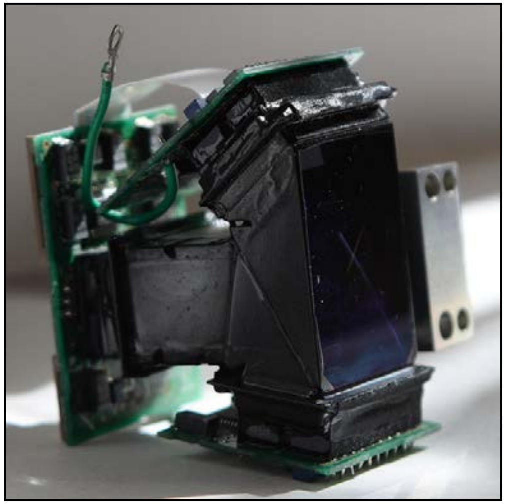
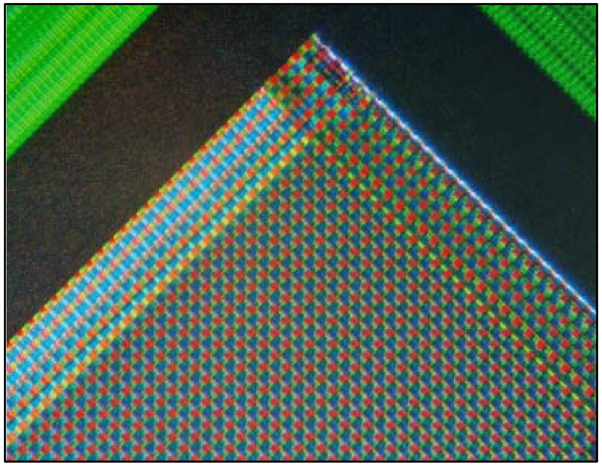
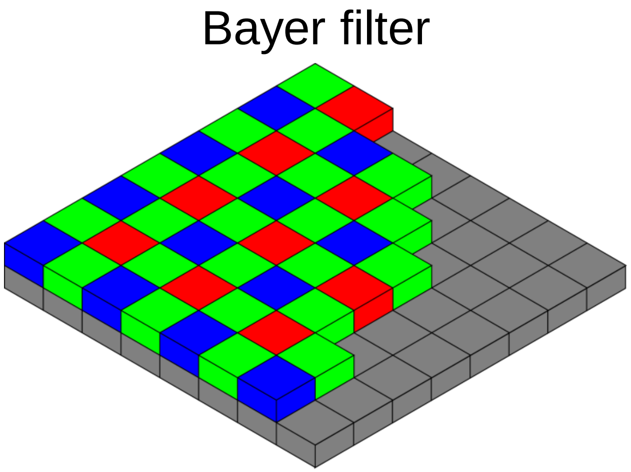


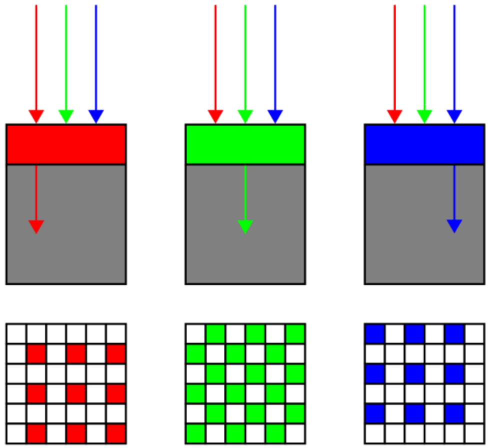
Image source: Wikipedia

# Color Filter Arrays

- In order to capture color information on a single sensor array:
  - Single sensors are grouped into clusters (e.g. quads)
  - A color filter is applied to each cell
- This way, color information is subsampled!





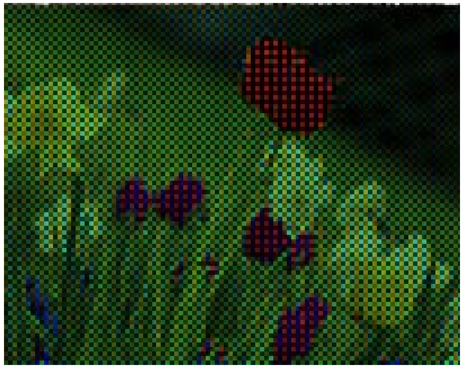
# Pixel Color Reconstruction

- To estimate the color information at each pixel, a reconstruction filter is used
  - Missing colors are weighted and interpolated from neighboring cells

Input

Sensor output

Color-coded output



Reconstructed image

# High Dynamic Range Images - Why

- Physically measured or simulated radiance (therefore luminance) in a natural environment matches the HVS levels
- Typical displays can achieve a dynamic contrast ratio of 6000:1 and an actual luminance level of 1-150cd/m$^2$

→ Screens are far from capable to display physically correct images!

- We need methods to adapt the computed radiance to the output intensity of a graphics system

# High Dynamic Range Images - Storage

- To be able to adjust the tonal range of the image output we need:
  - High precision (float/double) imaging algorithms
  - More than 8bits/color for storage (>255 levels)
  - Floating point precision buffers
- Either physical or canonical scale is assumed
- Frame buffers store values higher than [0-1] or
- Compressed ranges 0-1

# Tone Mapping

- Is the process of fitting a potentially huge luminance level to the tonal range of graphics display hardware
- Can be
  - Static
  - Adaptive
  - Delayed adaptive (to simulate the time required for the eyes to adjust to sudden change of illumination levels)
- According to image coverage, it can be
  - Global (same equation and params for all pixels)
  - Local (different adaptation for each pixel)

# Tone Mapping - Goals

- De-saturate useful range of information
- Enhance contrast of useful ranges
- Human visual system discriminates changes, not absolute values→
- Local contrast enhancement:
  - Separates tone levels of adjacent pixels →
  - accentuates details
- Simulate the retinal response to physical luminance levels (see blurring and bloom)

# Tone Mapping – Maximum to white

- Global operator
- Simple to implement (offline/real-time)
- Assuming normalized output: $L_o = L_i / L_{\max}$
- Ensures mapping of entire range to visible scale
- Reduces contrast for $L_{\max} > 1$
- Increases contrast for $L_{\max} < 1$
- Prone to significantly reduce levels if isolated high values are present

# Tone Mapping – Average Luminance

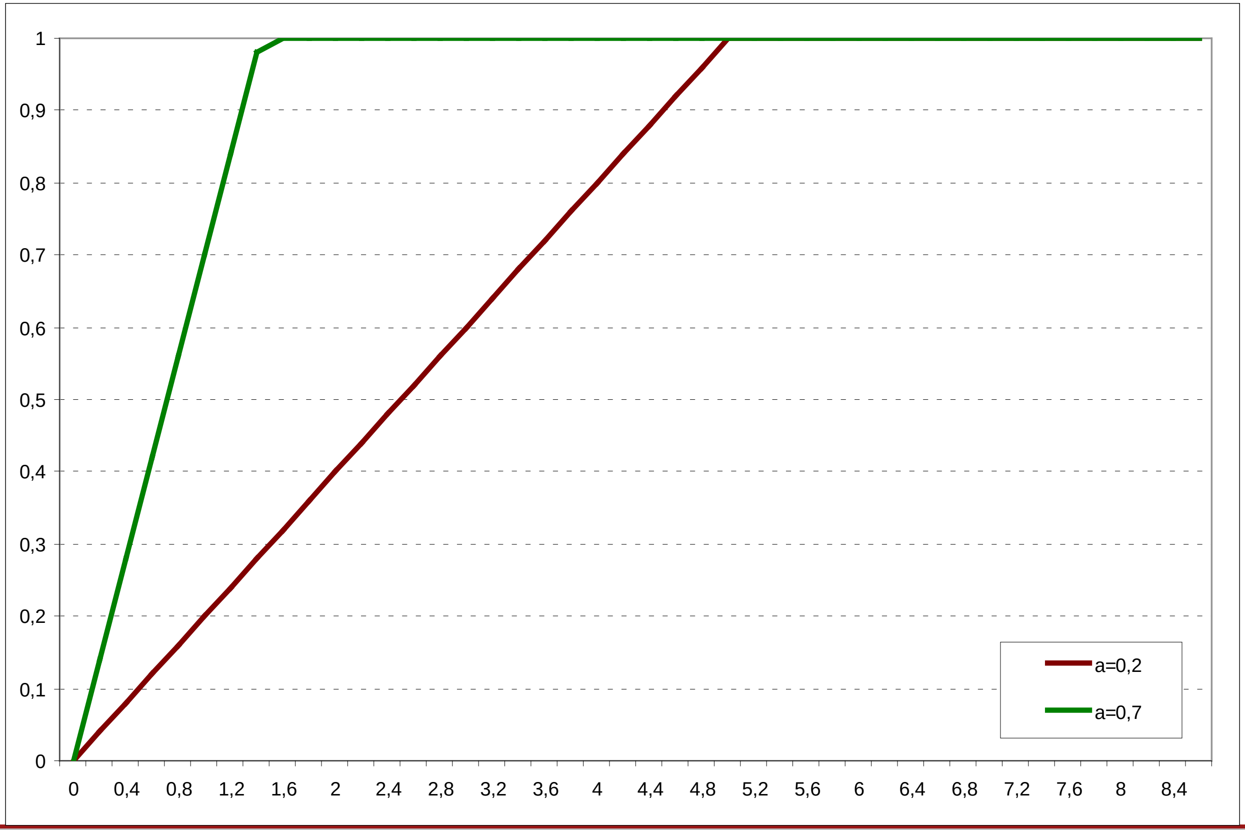
- In more sophisticated global tone mapping approaches, we evaluate the "general appearance" of an image instead of strict ranges
- We need to evaluate average luminance
- It is preferable to find the log-average of luminance and not the linear one:

$$\bar{L}_w = \exp\left(\frac{1}{N}\sum_{x,y}\log(\delta + L_w(x,y))\right), \qquad \delta = \text{small float}$$

- Because:
  - Perceived intensity on photoreceptors follows the power law
  - So does the working luminance $L_w$ (isolated pixel luminance against a uniform – average – background)

# Tone Mapping – Linear Mapping (1)

$$L_o(x,y) = \min\left\{ \frac{a}{\bar{L}_w} L_w(x,y), L_{o,\max} \right\}$$

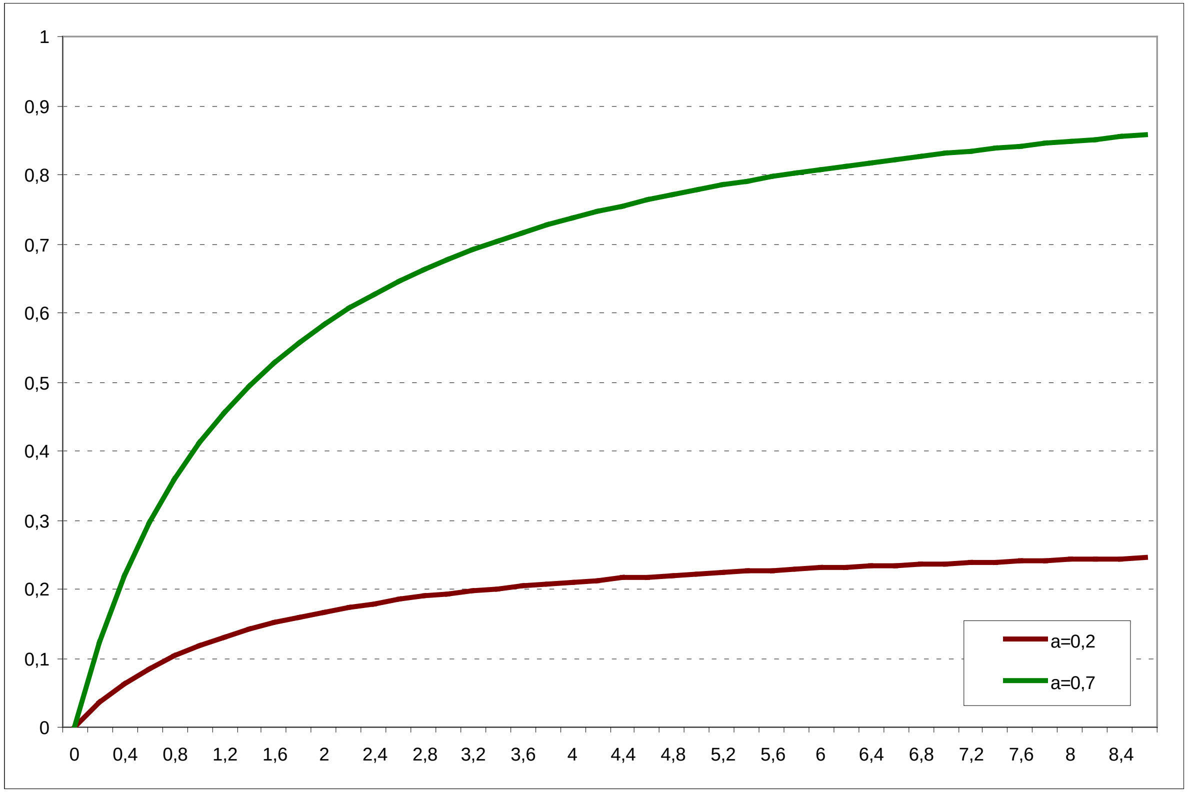


# Tone Mapping – Linear Mapping (1)

- $a$ is the tonal "key"
- Clipping
- Global technique
- Easy to implement (off-line/real-time)

# Tone Mapping – Non-linear Compression (1)

$$L_d(x,y) = \frac{L_o(x,y)}{1+L_o(x,y)} \qquad L_o(x,y) = \min\left\{\frac{a}{\bar{L}_w} L_w(x,y), L_{o,\max}\right\}$$

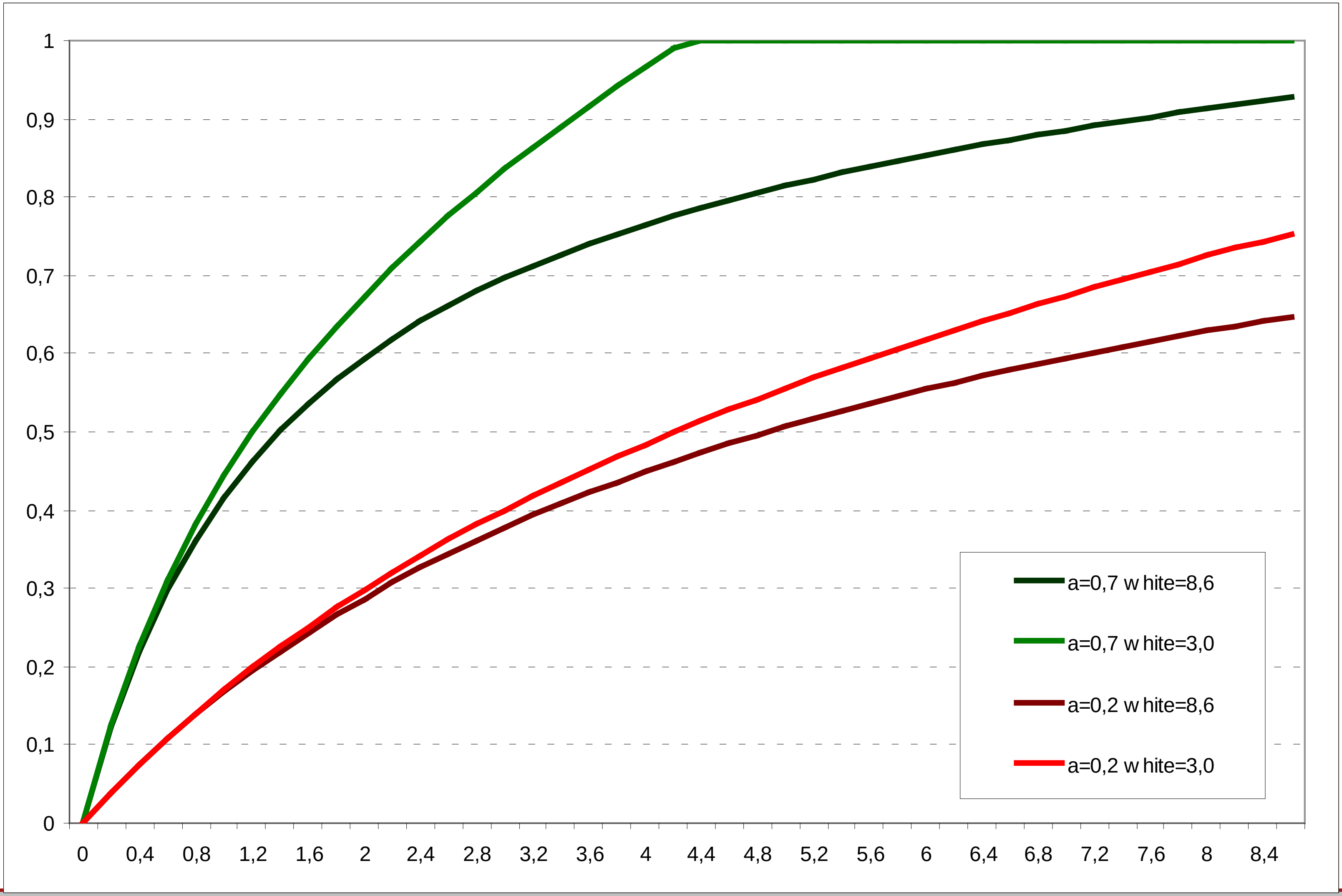


# Tone Mapping – Non-linear Compression (2)

- Enhances low-key tonal range
- No clipping
- Better used with a **white point** reference value (expected RGB luminance of "white" – background luminance):

$$L'_d(x,y) = \frac{L_o(x,y)\left(1 + \frac{L_o(x,y)}{L_{white}^2}\right)}{1 + L_o(x,y)}$$

# Tone Mapping – Non-linear Compression (3)



# Post-processing Enhancements

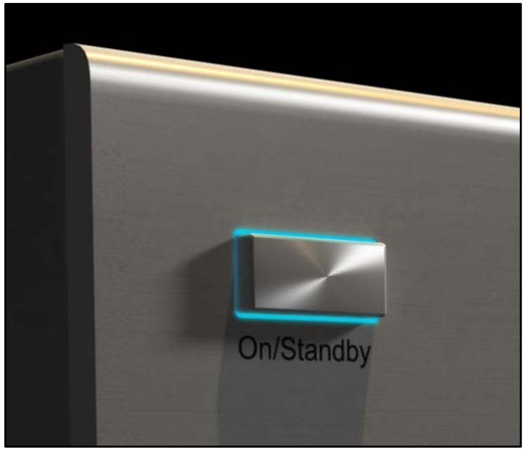
- “Visual tricks” can enhance the tonal discrimination and interpretation of an image
- Two dominant techniques:
  - Bloom
  - Unsharp mask filtering

# Bloom (1)

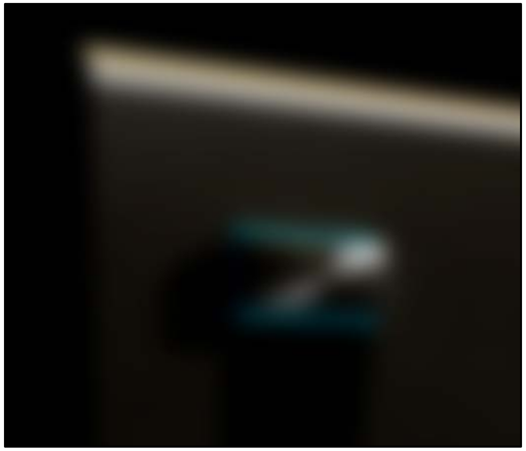
- When very bright light is perceived by the human eye, a noticeable glow or intensity “spill” is spread towards the darker regions
- This effect is called bloom and when artificially reproduced in synthetic images, can fool the HVS that an image region is brighter than it really is

# Bloom (2)

- To simulate bloom:
  - Subtract a high threshold from the image
  - Blur the result to spread the intensity
  - Modulate the blurred image to achieve the desired effect presence
  - Add to original image




Original



Blurred original + threshold


Original + blurred

# Real-time Bloom (1)

- For real-time rendering bloom is performed similar to off-line rendering
- Blurring (convolution) is an expensive operation
- Requires look-ups and updates over the image → better separate read/store images→ use a “blur buffer”
- Steps:
  - Use a low-resolution frame buffer to store the clipped image
  - Perform upscaling (via bilinear interpolation or/and multisampling) of the low-res buffer
  - Add the result to the image

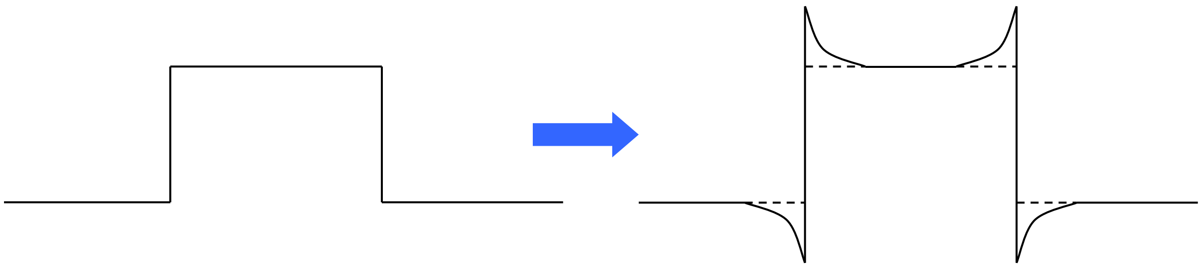
# Real-time Bloom (2)

Alan Wake

https://udn.epicgames.com/Three/ContentBlogArchive.html

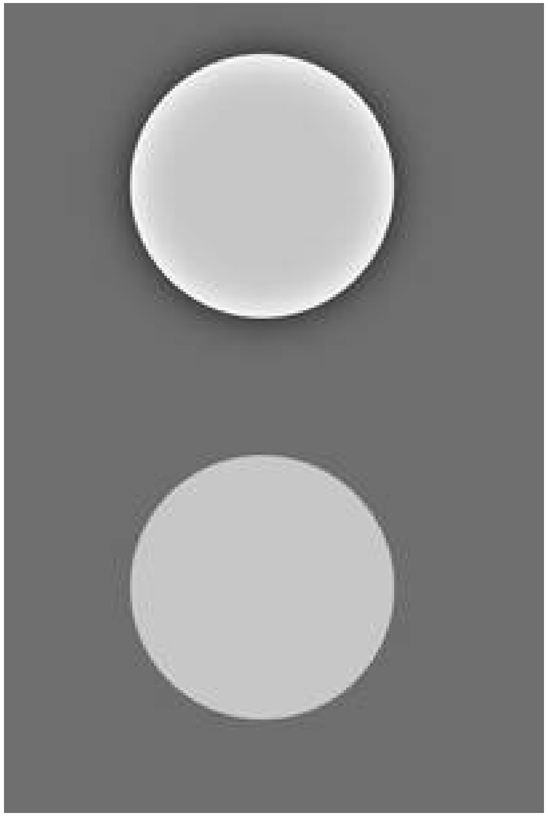
# Local Contrast Enhancement

- Local sharpening of the image features gives the illusion of greater dynamic range:

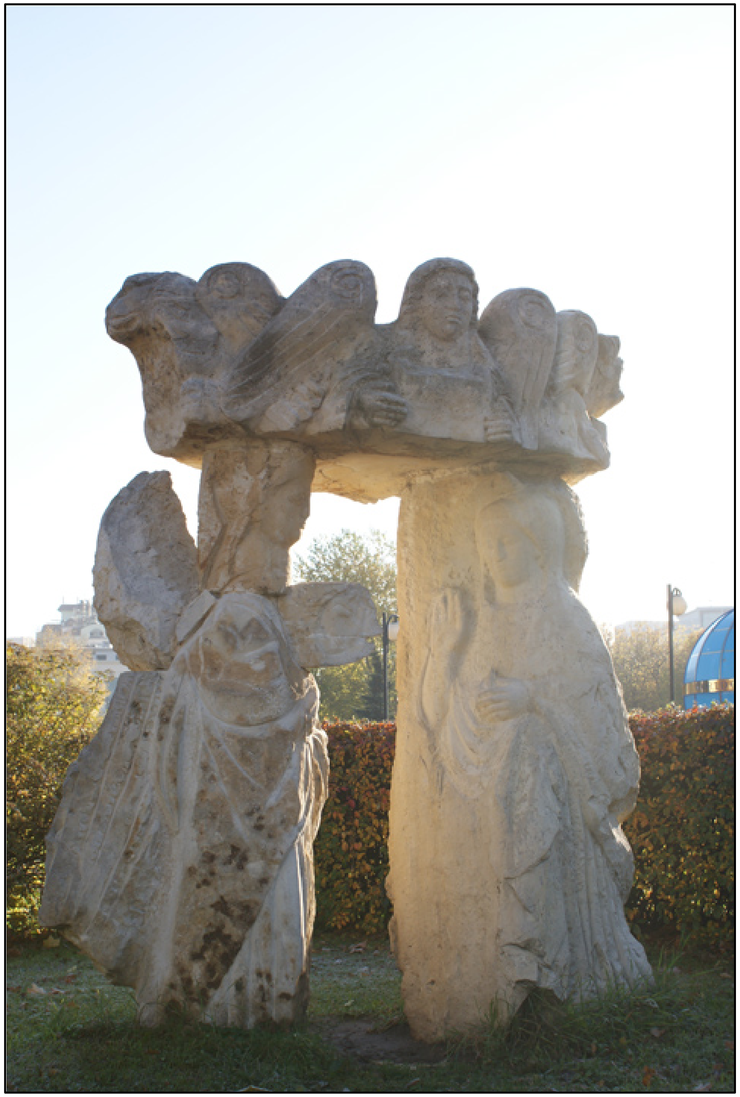


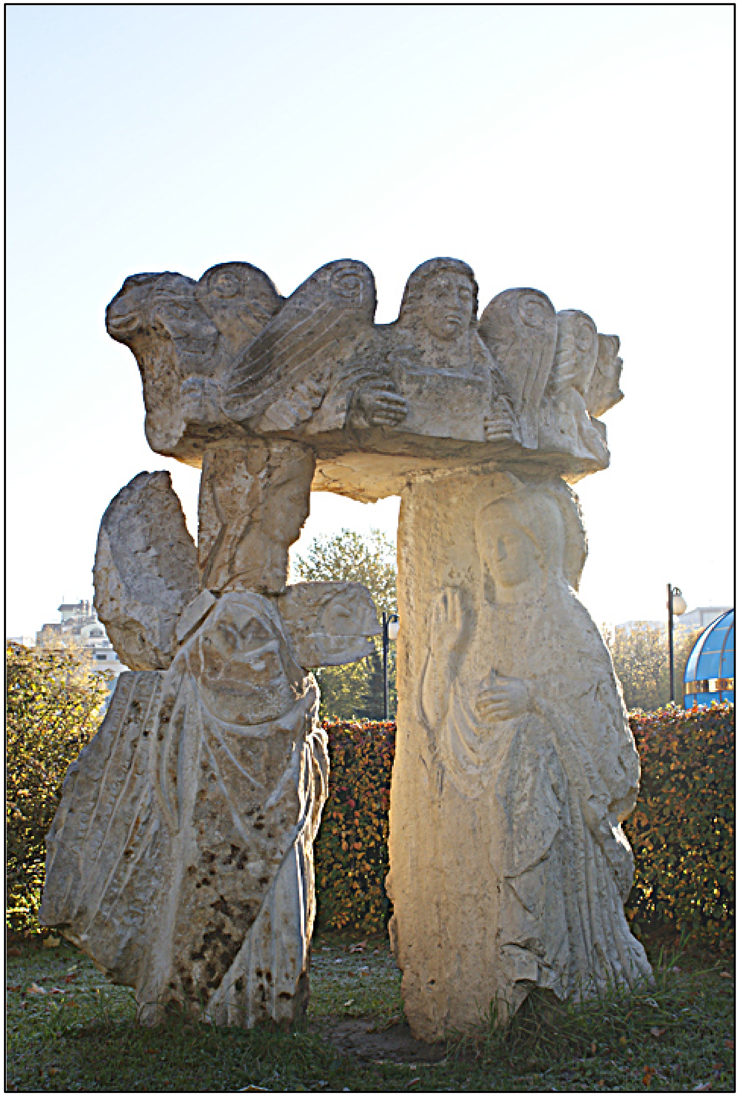
$$I'(x,y) = I(x,y) - s \cdot \nabla^2 I(x,y)$$

$$\nabla^2 I(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2}$$



# Local Contrast Enhancement Example





# Contributors

- Georgios Papaioannou

# References


- T. Akenine-Moller, E. Haines, N. Hoffman, Real-Time Rendering, 3rd Ed., AK Peters, 2008
- M. Pharr, G. Humphreys, Physically Based Rendering, Morgan Kaufmann, 2004
- J. Tumblin, H. Rushmeier, Tone reproduction for computer generated images. *IEEE Computer Graphics and Applications 13*, 6 (November), 42–48, 1993
- E. Reinhard, M. Stark, P. Shirley, J. Ferwerda, Photographic tone reproduction for digital images. *ACM Trans. Graph.* 21, 3 (Jul. 2002), 267-276, 2002