Computer Graphics

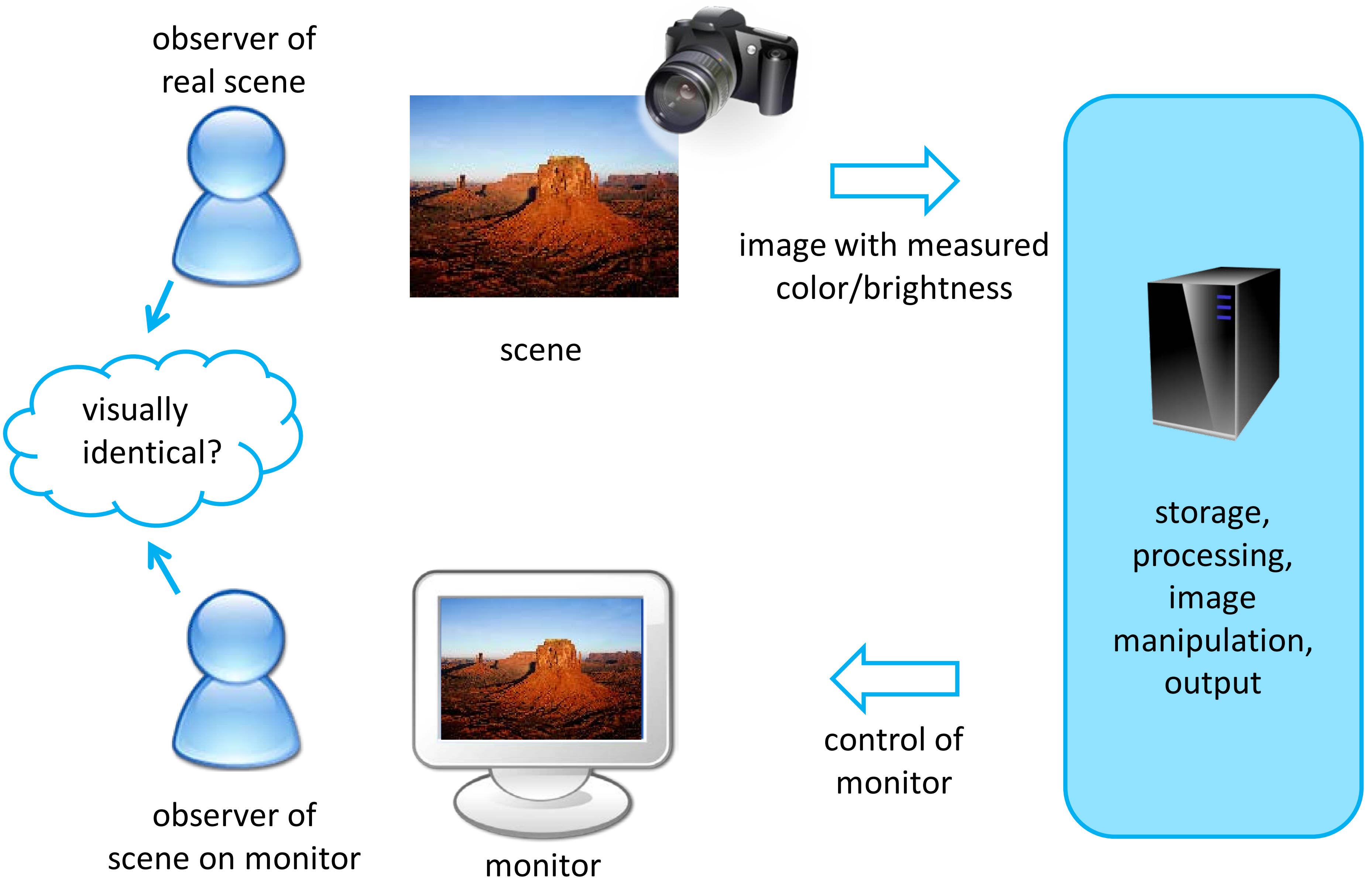
Chapter 2: Perception and Color

Winter Term 2022/23

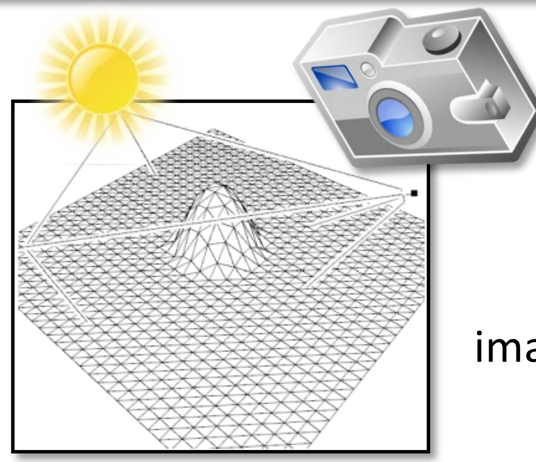
Filip Sadlo

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Computer Graphics and Color Images



Computer Graphics and Color Images



virtual scene
(geometry, material, camera, light sources, …)

image synthesis

storage,
processing,
image
manipulation,
output

realistic image?
illumination,
material, …?

observer of
scene on monitor

monitor

control of
monitor

Computer Graphics and Color Images

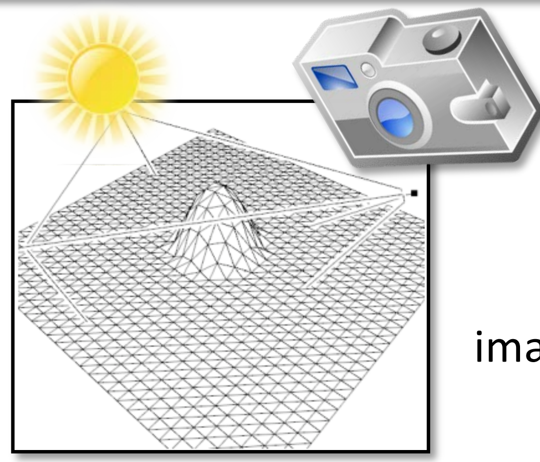


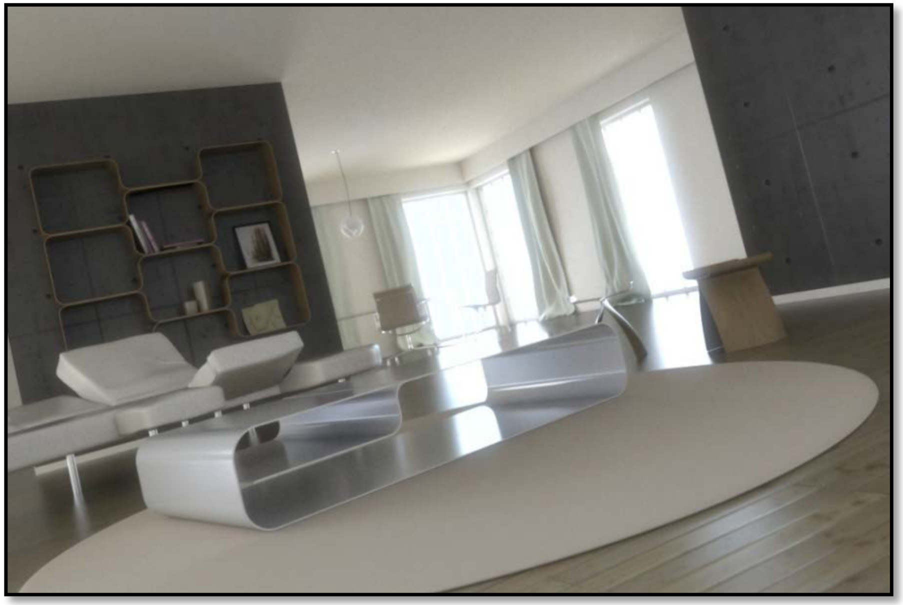
image synthesis

virtual scene
(geometry, material, camera, light sources, …)

storage,
processing,
image
manipulation,
output

… complemented with real data
(textures, environment maps, 3D scanner, …)

Example: Physically Correct Illumination



www.luxrender.net

Example: Augmented Reality

image: Thorsten Grosch

Example: Augmented Reality

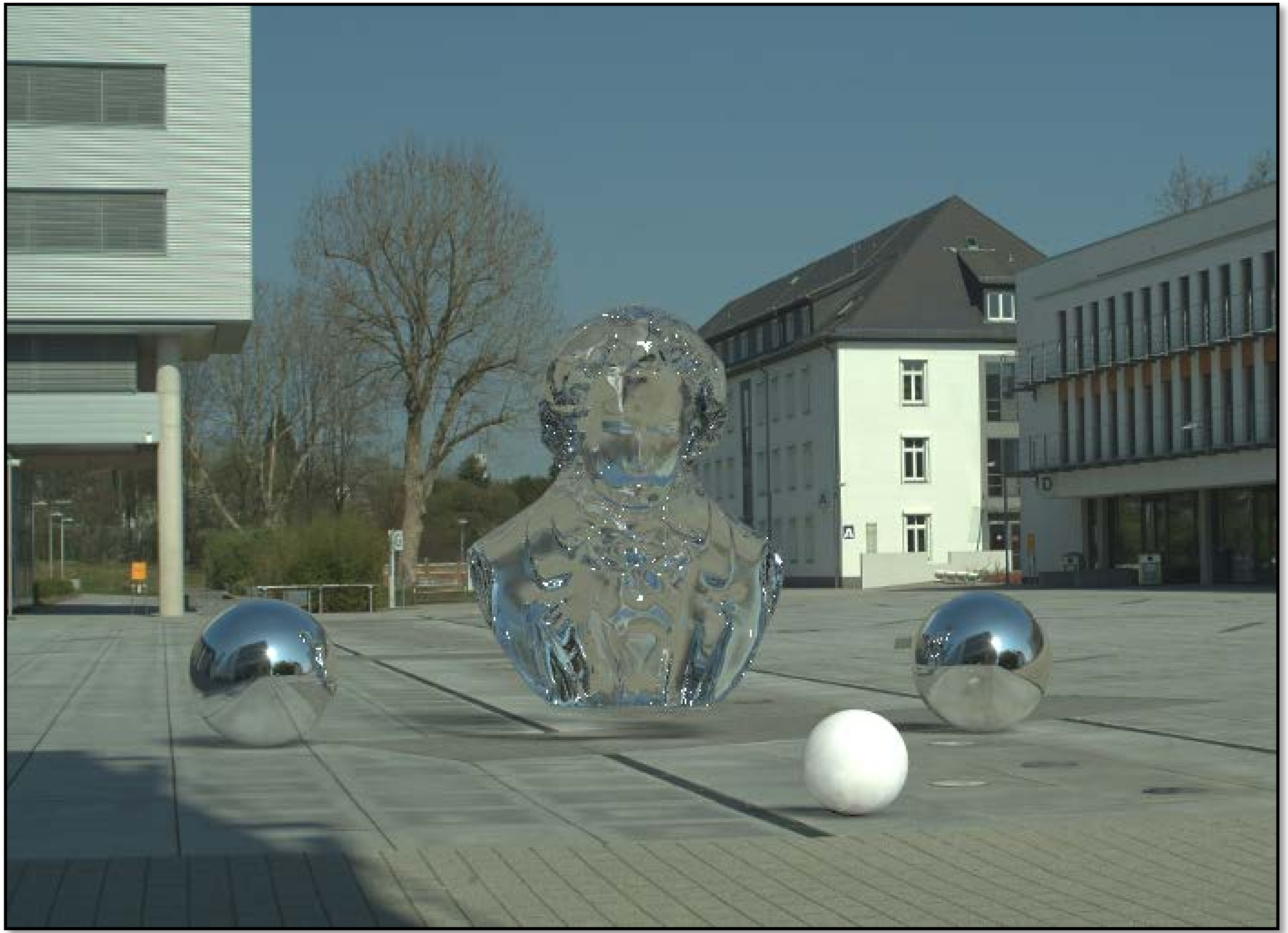
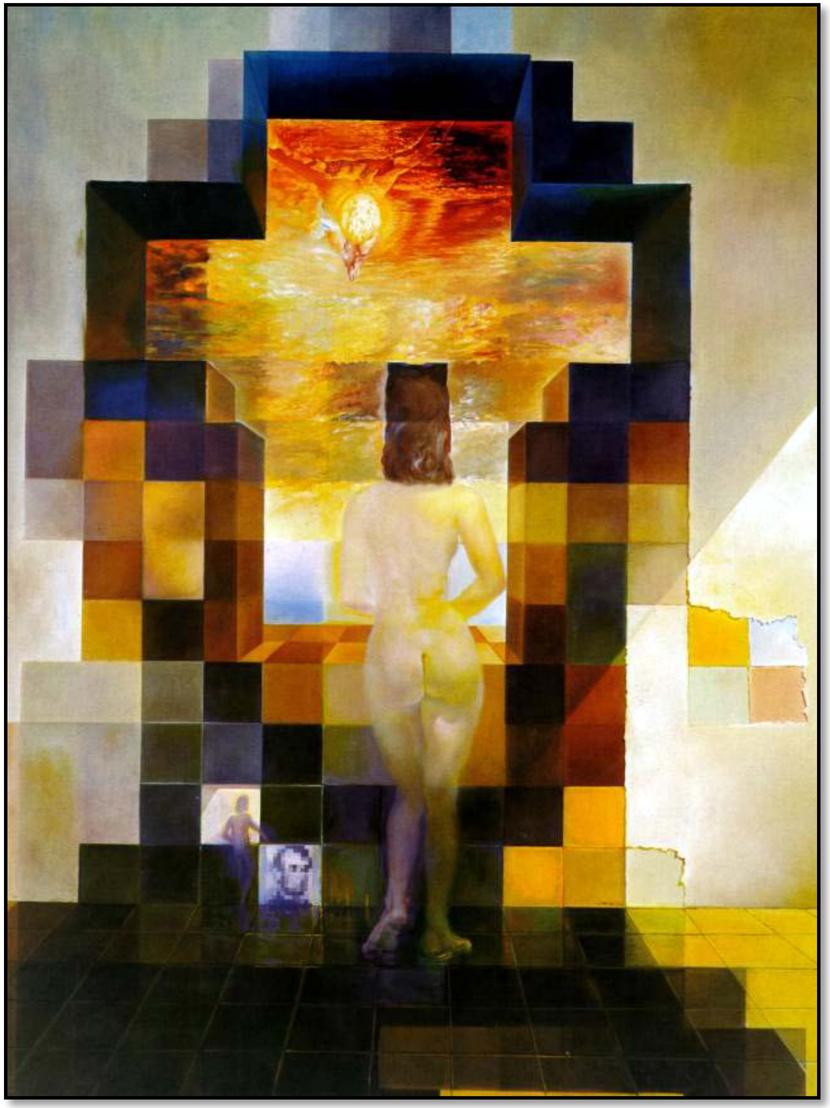


image: Thorsten Grosch

Outline

- Images and representation
- Light, vision, and perception
- Color vision, representation of color, and color spaces
- Raster images
  - Sampling
  - Simple operations on images



Salvador Dali
"Gala Contemplating the Mediterranean Sea, which at 20 meters becomes the portrait of Abraham Lincoln", 1976 (Dalí Museum, St. Petersburg, Florida)

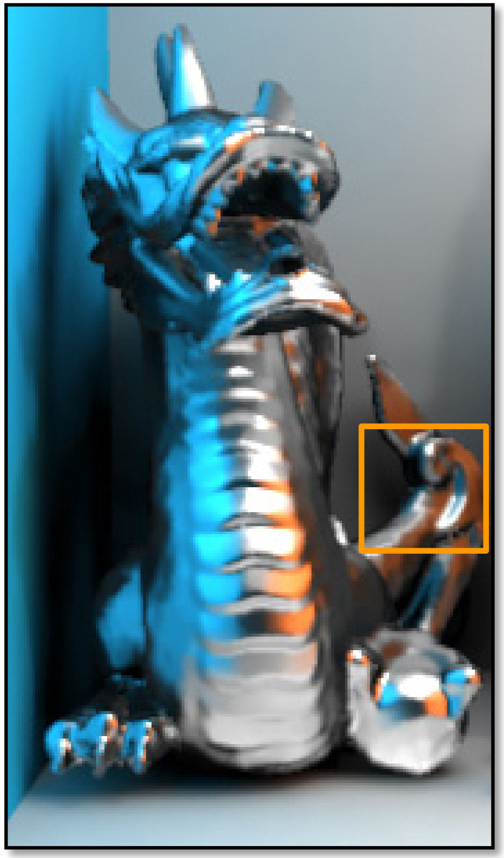
Outline

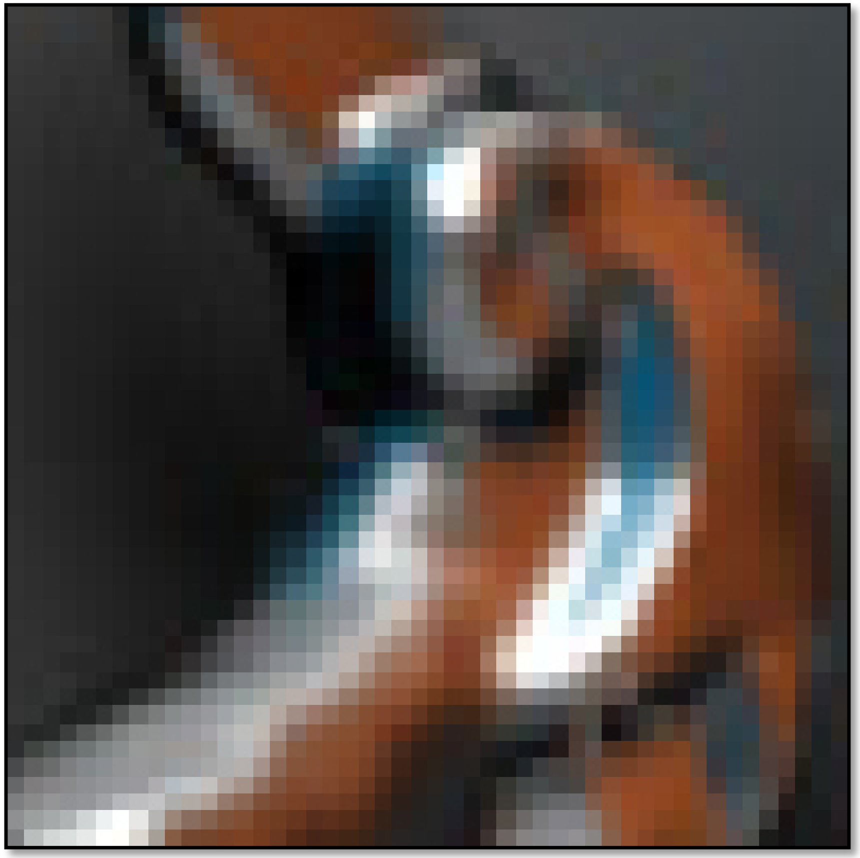
- Images and representation
- Light, vision, and perception
- Color vision, representation of color, and color spaces
- Raster images
  - Sampling
  - Simple operations on images

Salvador Dali
"Gala Contemplating the Mediterranean Sea, which at 20 meters becomes the portrait of Abraham Lincoln", 1976 (Dalí Museum, St. Petersburg, Florida)

Computer Graphics and Color Images

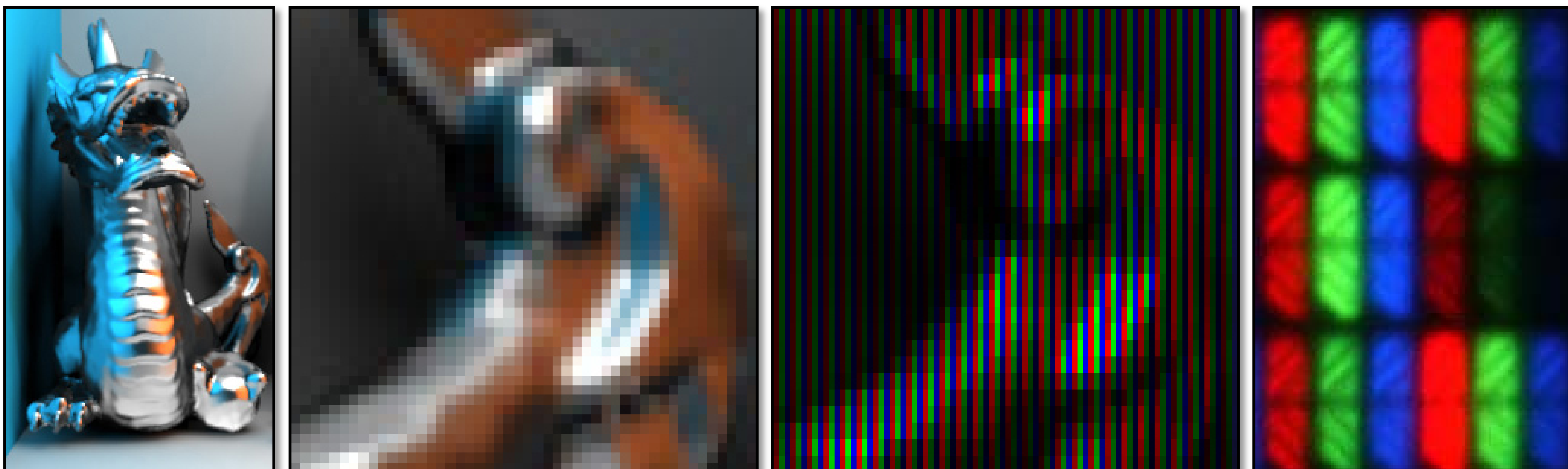
- CG: generation and manipulation of images
  - Images are—in general—raster images: 2D arrays of colored pixels (picture elements)
  - But: what is light? Color?





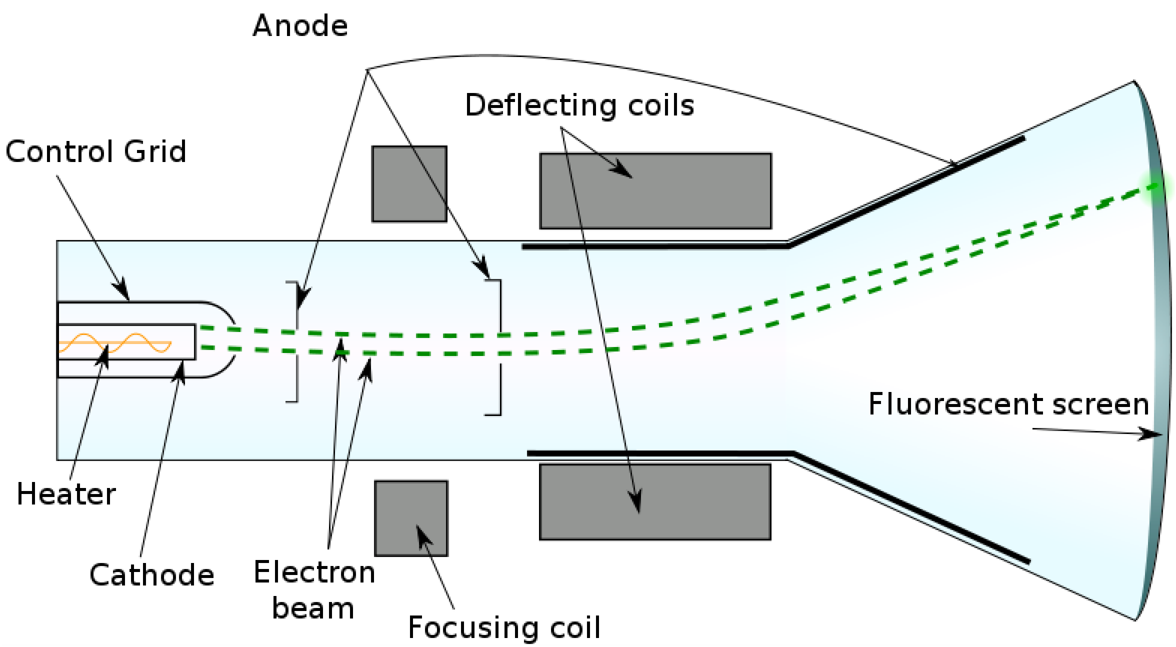
Raster Displays and Images

- Devices for display and acquisition of images, e.g.:
  - Monitor: pixels, colors by mixing of red, green, and blue
  - Digital camera: sensor = 2D grid of light-sensitive pixels

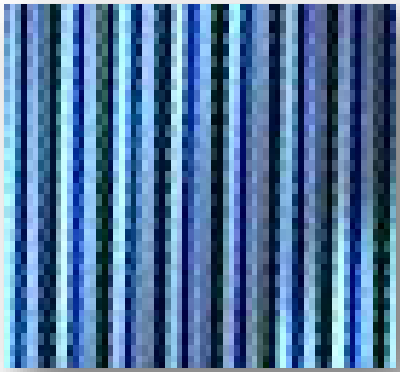


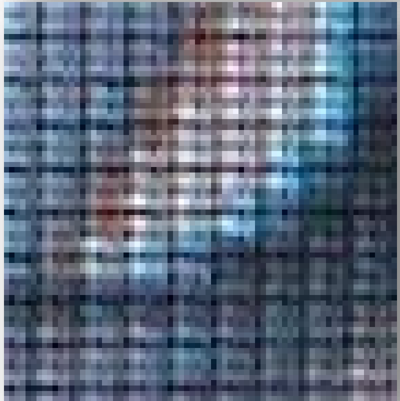
Graphical Output Devices

- **Hardcopy devices**
  - Printer, plotter, video tapes, etc.
- **Display devices**
  - Light-emitting: CRT, LED, plasma
  - Light-propagating: LCD
- **CRT (cathode ray tube)**



CRT aperture grill

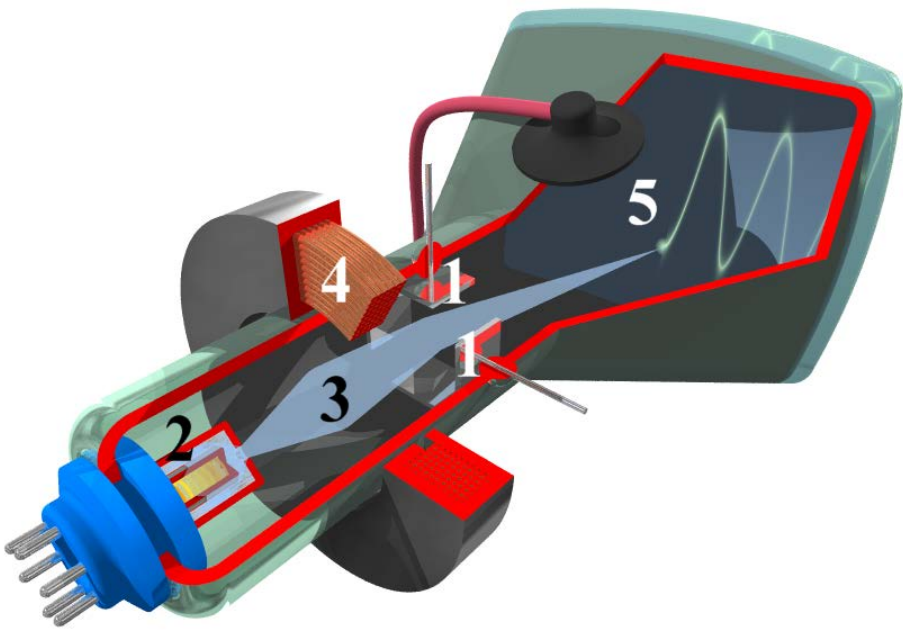




TFT

Vector Monitor

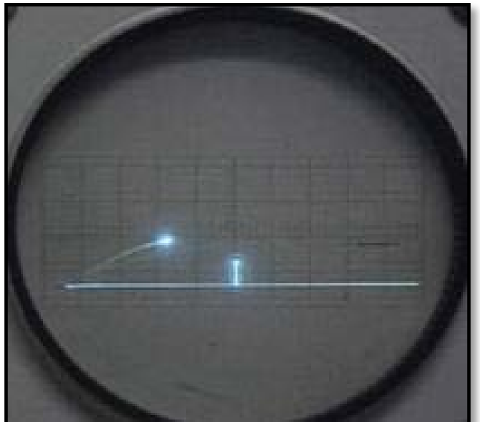
- Basically an oscilloscope
  - Deflecting coils (1), cathode (2), electron beam (3), focusing coil (4), phosphorous coating at inside of screen (5)
  - Electrons on phosphorous coating cause
    - Fluorescence (duration about 22 µs) and
    - Phosphorescence (about 210 ms)
- Representation of an image
  - Draw objects (lines) continuously
  - Represent image repeatedly
  - High refresh rate necessary

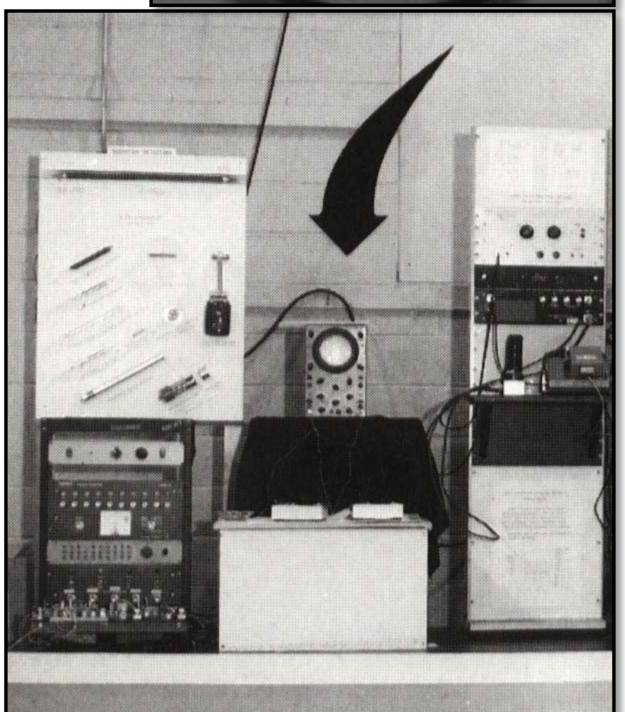


http://en.wikipedia.org/wiki/Oscilloscope

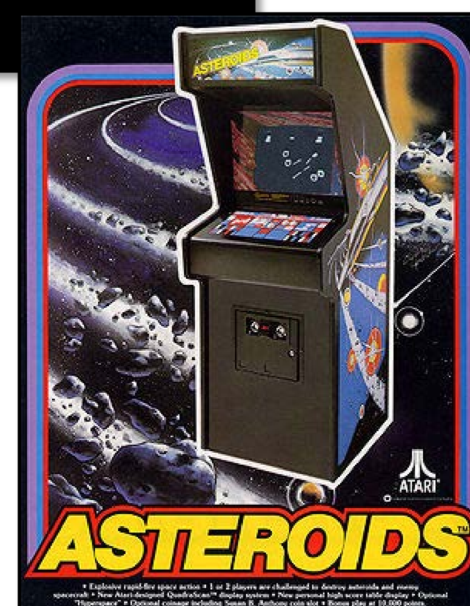
Vector Monitor

- Representation of image: draw objects (lines) continuously





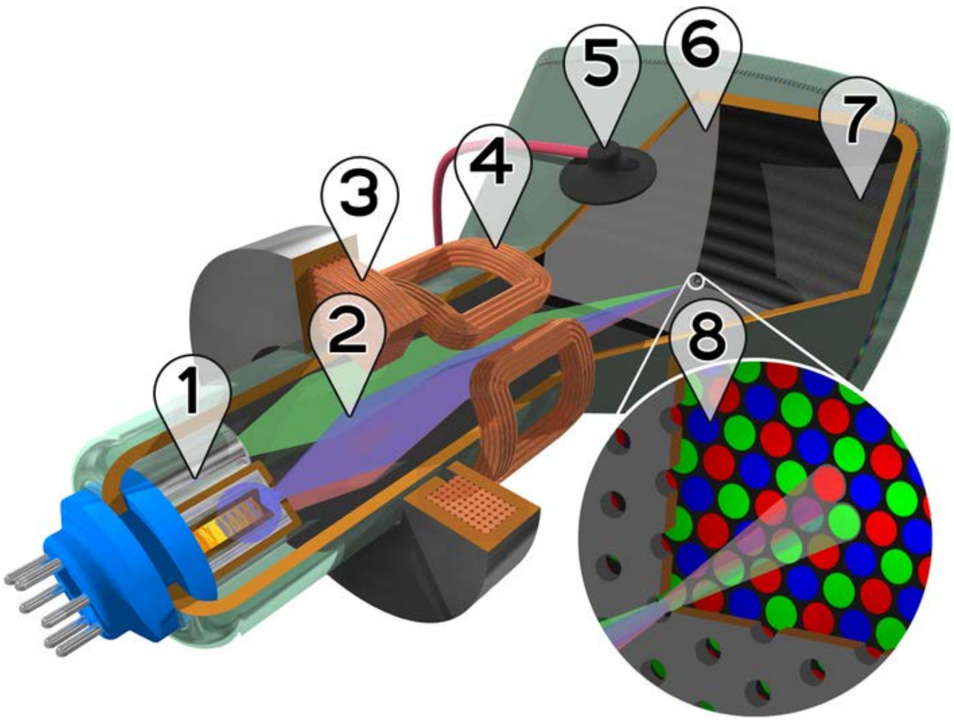
Asteriods, Atari 1979

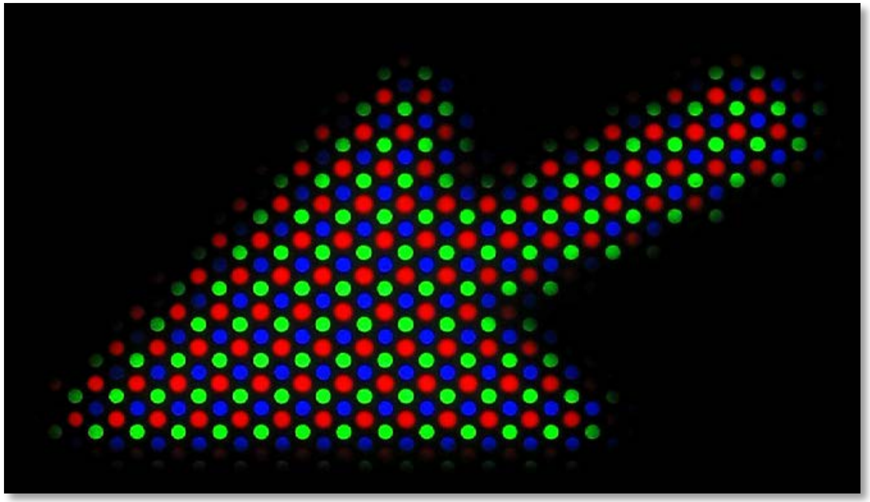


www.gamersquarter.com/tennisfortwo/,
William Higinbotham 1958

Cathode Ray Tube, CRT

- Cathodes (1), electron beams (2), focusing coils (3), deflecting coils (4), anode connection (5), shadow mask (6), fluorescent coating with red, green, and blue pixels (7,8)

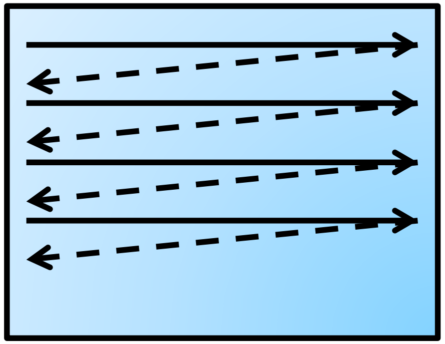




https://en.wikipedia.org/wiki/Shadow_mask

Cathode Ray Tube, CRT

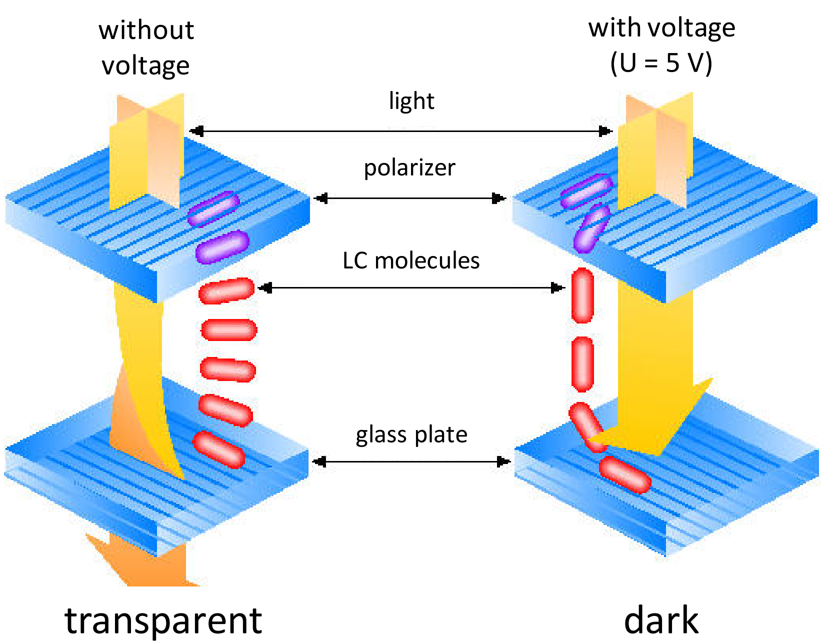
- Principle of representation
  - Image is generated row by row (scan line) (left $\rightarrow$ right, top $\rightarrow$ bottom)
  - Flickerfree representation requires high speed (> 50 Hz)
  - Intensity of electron beams (at the moment at which a pixel is hit) determines brightness of the pixels

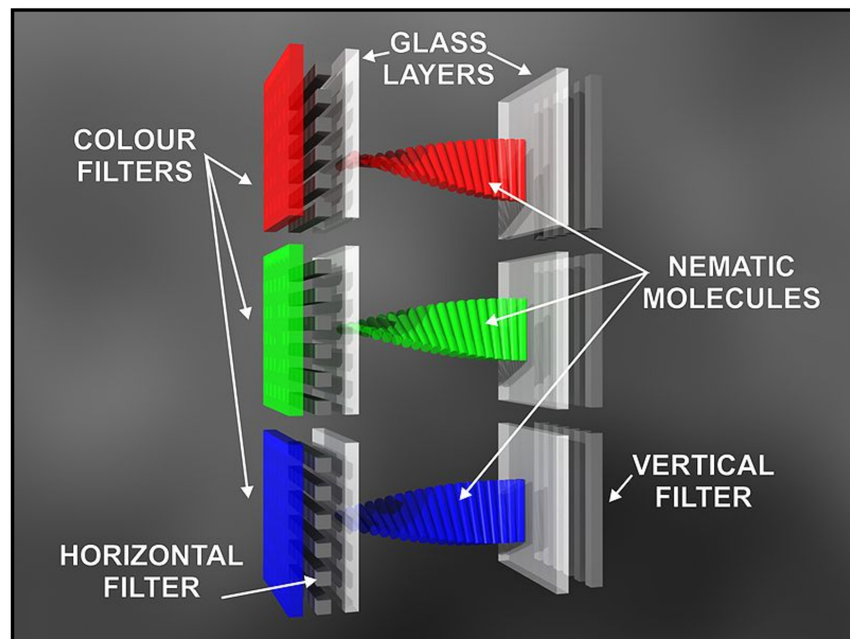


- For this, the image has to be available
  - Obvious: store image as 2D array of pixels
  - For each pixel, store the color with three values for red/green/blue
  - **Frame buffer**: memory in which the image is stored for display
  - Crucial advantage over vector monitors: once image is in frame buffer, display is independent of its computational cost

Liquid Crystal Display, LCD

- Inner surface of two glass plates coated with transparent electrode, liquid crystals in between
- Polarizing filters: 90° orientation to each other
- Orientation of liquid crystals
  - Without voltage: achieved by coating
  - With voltage: according to electric field
  - Liquid crystals modify polarization of light



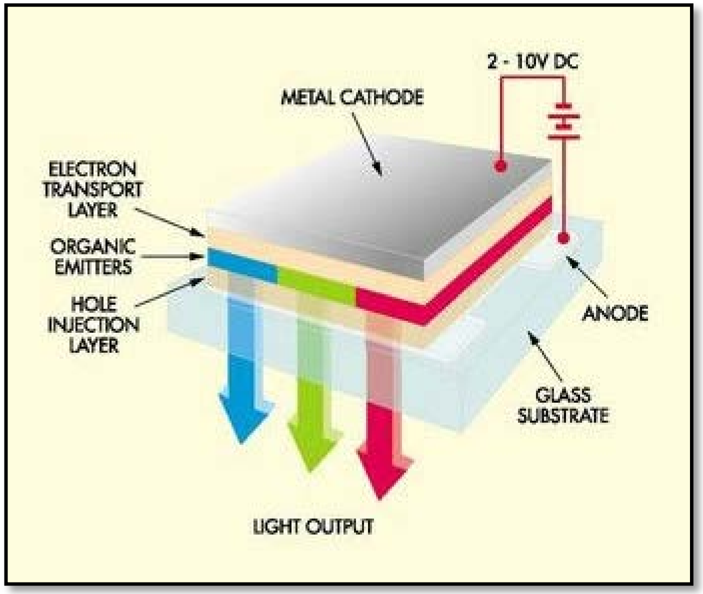


http://en.wikipedia.org/wiki/Liquid_crystal_display

Other Displays

- Further light-emitting screens
  - E.g., LED, Plasma
- Projectors
  - CRT, LCD, DLP/DMD, D-ILA

- LED (light-emitting diode)
  - (Anorganic) semiconductors
- OLED (organic light-emitting diode)
  - Thin-film displays made from organic semiconducting materials
  - Emit light if voltage is applied
- Electronic paper, e-ink, e-paper



Frame Buffer

- Imagine a frame buffer as follows ...

```
#define WIDTH  1024
#define HEIGHT 768

// pointer to rgb frame buffer memory
unsigned char *buffer = ...;

for ( y = 0; y < HEIGHT; y++ ) {
  for ( x = 0; x < WIDTH; x++ ) {
    // values between 0 and 255 (lowest/brightest intensity)
    buffer[ ( x + y * WIDTH ) * 3 + 0 ] = “red value”;
    buffer[ ( x + y * WIDTH ) * 3 + 1 ] = “green value”;
    buffer[ ( x + y * WIDTH ) * 3 + 2 ] = “blue value”;
}  }
```

- Modern systems do not provide direct access to this memory

Frame Buffer

- Imagine a frame buffer as follows …

```
#define WIDTH  1024
#define HEIGHT 768

// pointer to rgb frame buffer memory
unsigned char *buffer = ...;

for ( y = 0; y < HEIGHT; y++ ) {
  for ( x = 0; x < WIDTH; x++ ) {
    // values between 0 and 255 (lowest/brightest intensity)
    buffer[ ( x + y * WIDTH ) * 3 + 0 ] = “red value”;
    buffer[ ( x + y * WIDTH ) * 3 + 1 ] = “green value”;
    buffer[ ( x + y * WIDTH ) * 3 + 2 ] = “blue value”;
}  }

CopyImageToScreen( buffer );
```

- … but there are always API functions that allow you to display some memory range in a window / on the screen

Characteristics

Vector Displays

- Image buildup difficult in case of complex scenes
- Lines and wireframe models
- High resolution

Vector Graphics (e.g., SVG)

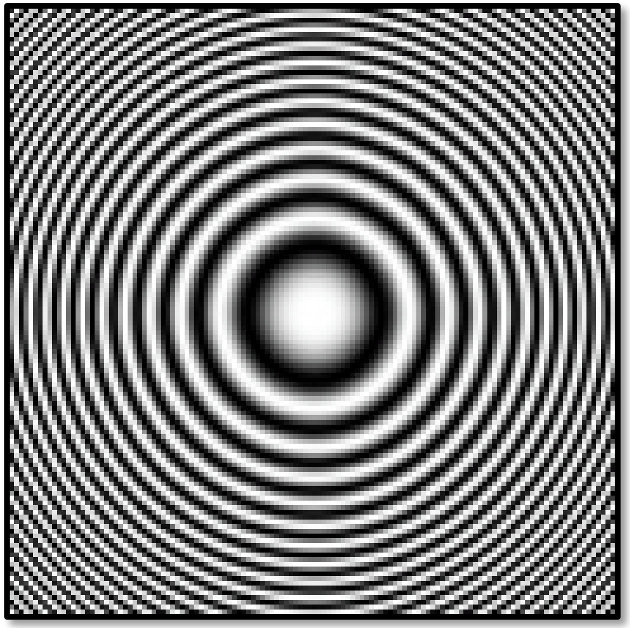
- Important for resolution-independent representation

Raster Graphics

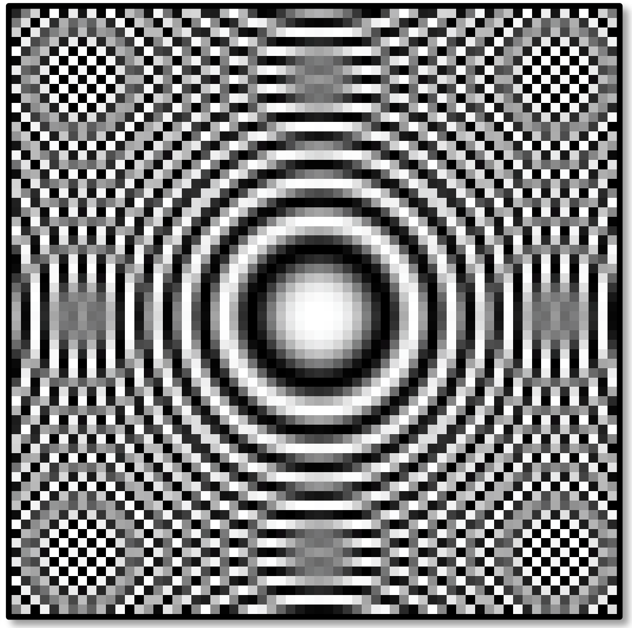
- Repeated image buildup is independent of scene complexity
- Typically filled, shaded surfaces
- Finite number of pixels: aliasing and Moiré effects
- Access to object “lost” (picking)

For the moment, we only talk about raster images and raster graphics

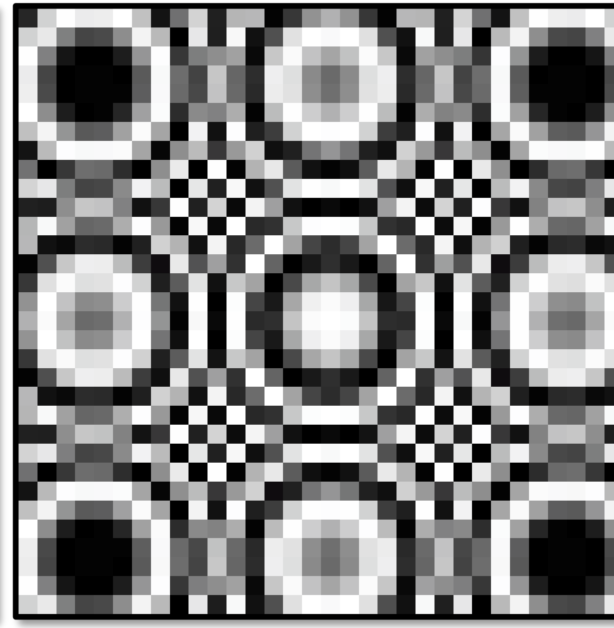
Preview: Aliasing



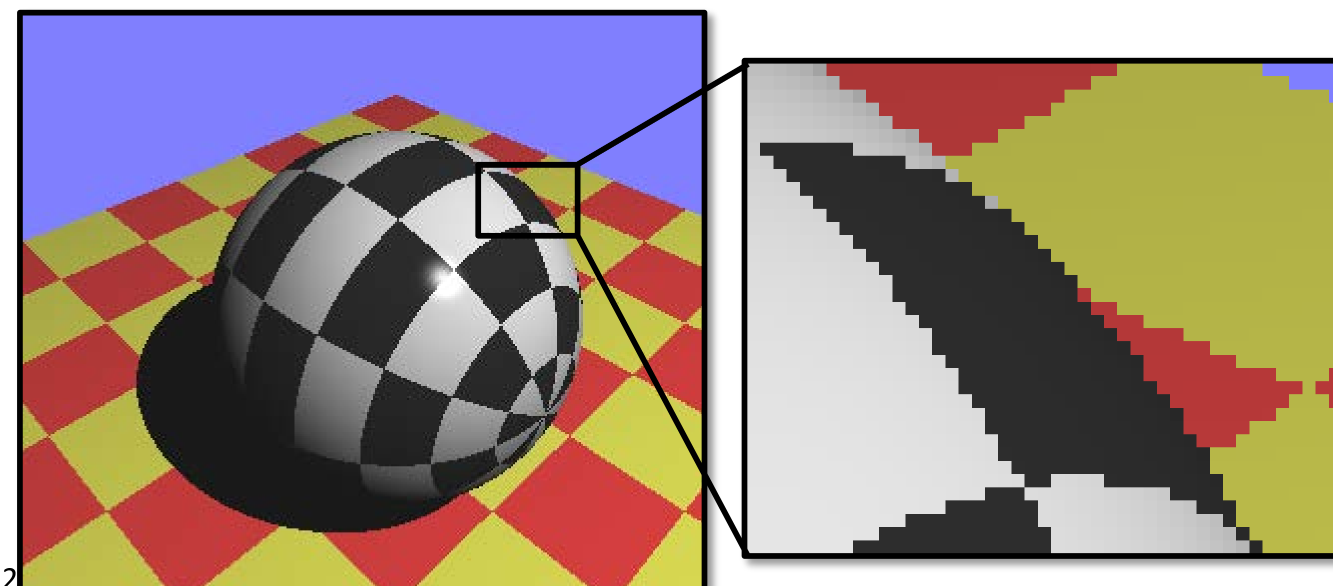
WIDTH=HEIGHT=128



WIDTH=HEIGHT=64

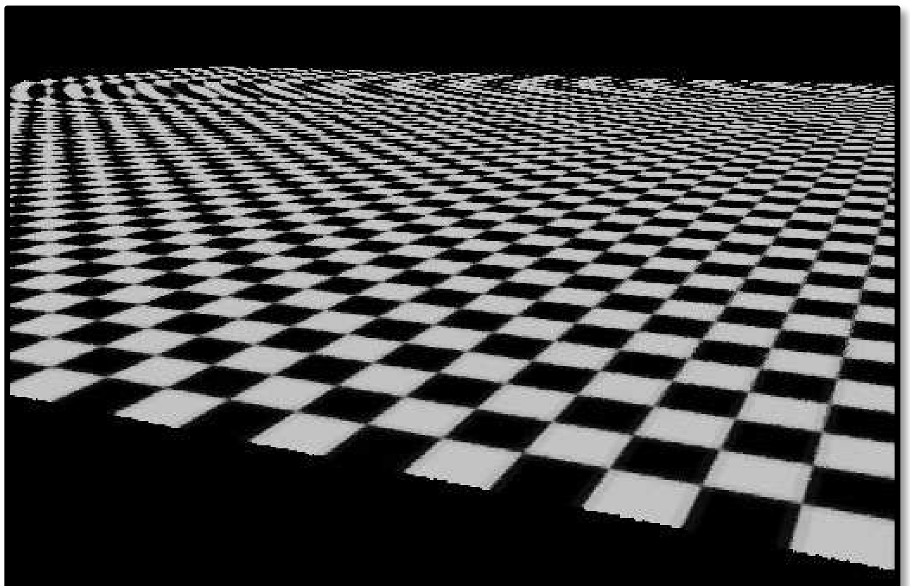


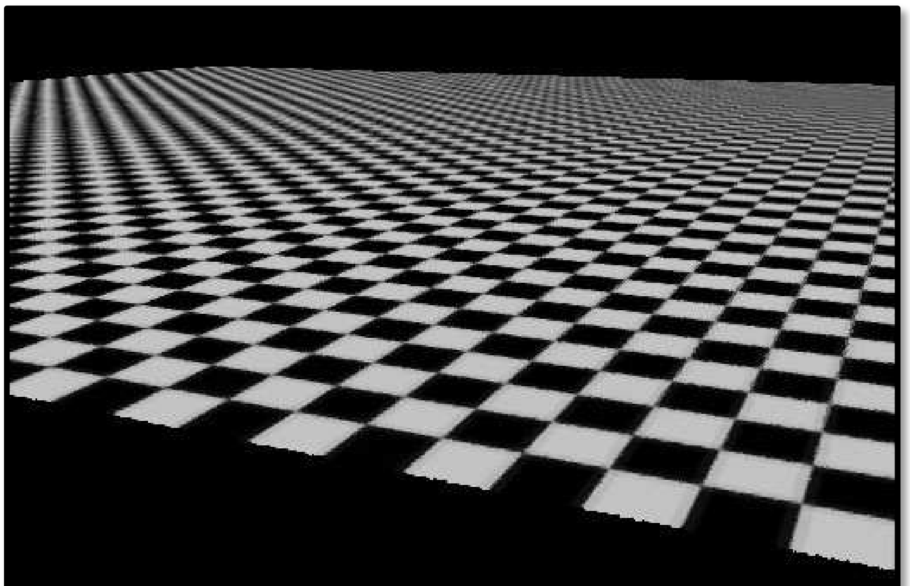
WIDTH=HEIGHT=32



Preview: Aliasing

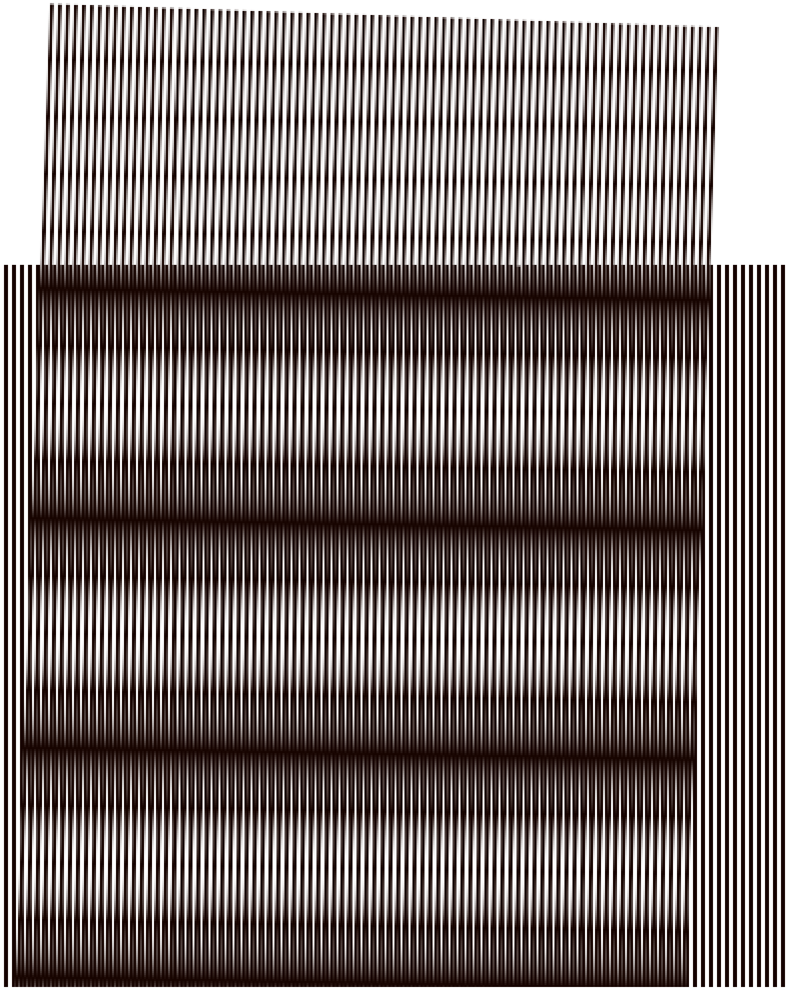
- … the classical example for aliasing in computer graphics





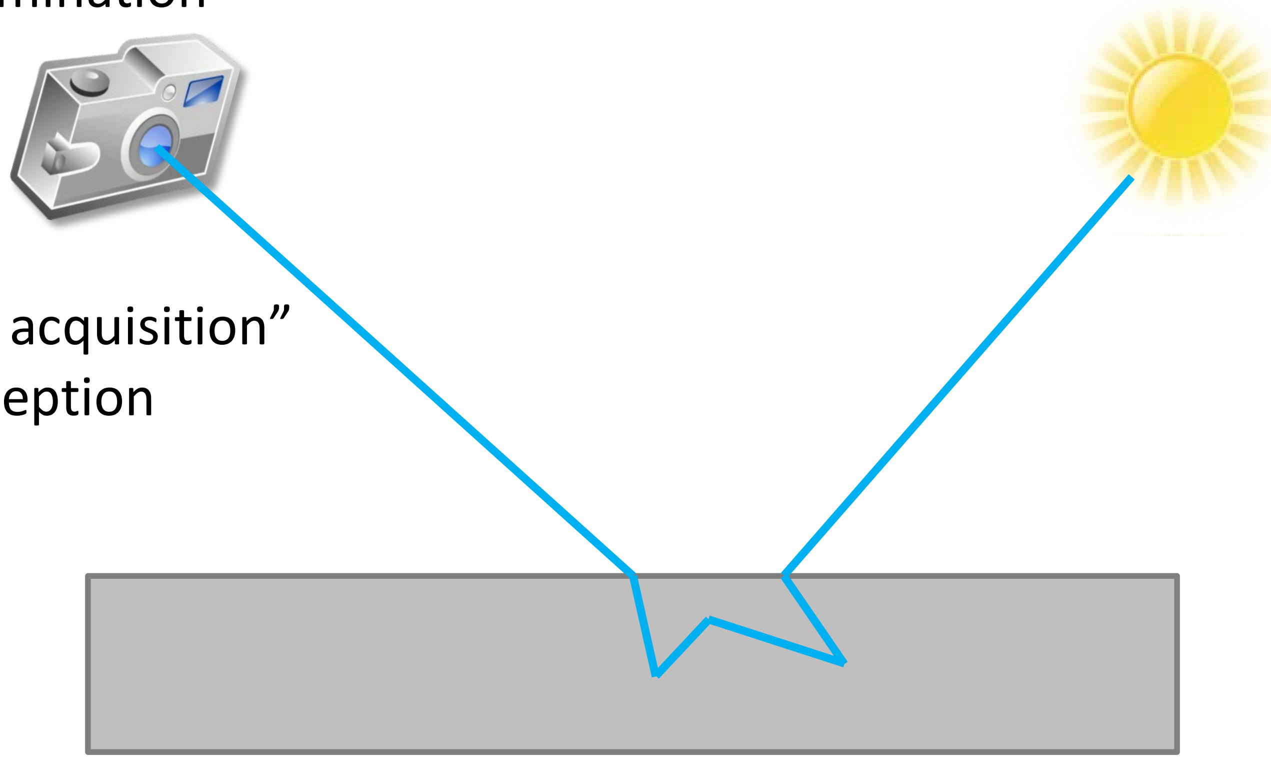
Moiré Effect

- Superposition of regular line rasters



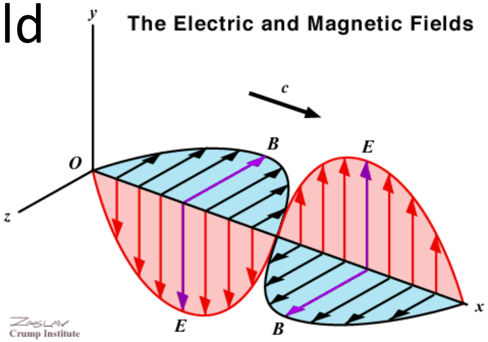
Computer Graphics: Light and Perception

- Understanding **light** and **perception** is important for
  - Acquisition and output on different devices
  - Visualization
  - Realistic image generation
- Process of image generation
  - Light source / illumination
  - Absorption
  - Scattering
  - Reflection
  - “Exposure during acquisition” resp. human perception



What is Light?

- Electromagnetic radiation
  - Propagating waves of an electromagnetic field
  - Wavelength
  - Intensity
  - Polarization

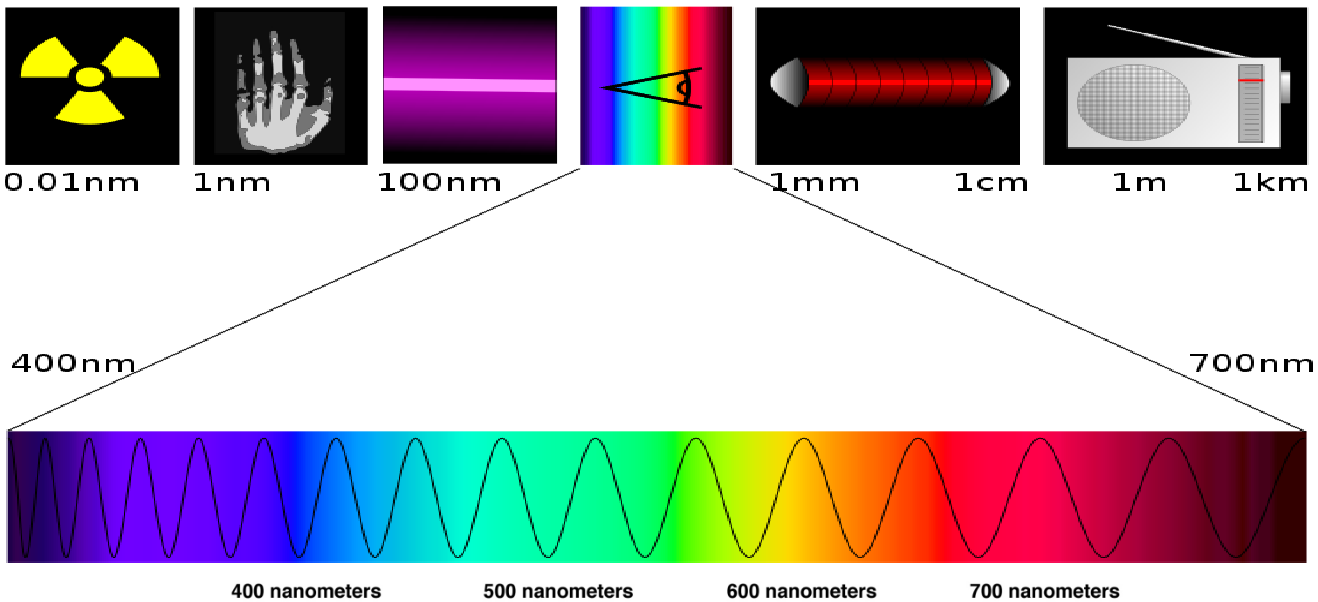


- Wave–particle dualism
  - Wave optics: diffraction, interference (Maxwell, 1865)
  - Particle character, light quantum / photons (Albert Einstein, 1905)
    - Frequency ν, wavelength $\lambda=c/\nu$
    - Energy of a photon: $E=h\nu$ $[eV]$ or $[J]$, with Planck constant h
  - Ray optics: geometrical optics as standard in CG

- Radiometry: radiation resistance, measurement of electromagn. radiation
- Photometry: inclusion of sensitivity of the observer

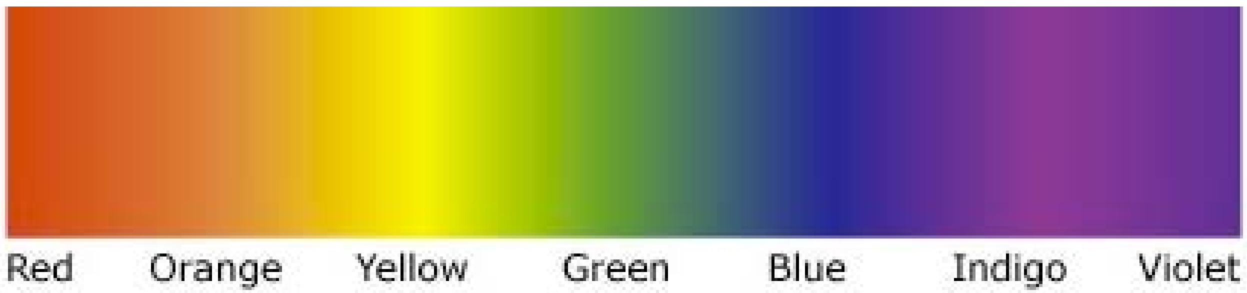
Radiometric Quantities

- Wavelength (λ)
  - Measured in meters (m)
  - Every wavelength represents a spectral color
  - Frequency $\nu = c/\lambda$ $[\mathrm{Hz}]$
- Spectrum of electromagnetic radiation is wide
  - Visible light only a very small range: approx., 380–700 nm

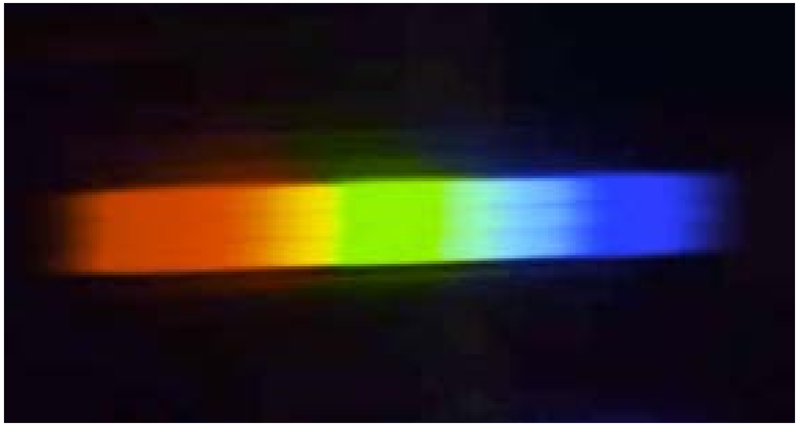


Spectral Colors

- Monochromatic light (only a single wavelength) …



- … appears as bright and clean color
  - Caution: what you see here is only a reproduction, no spectral color!



Visible Spectrum

image: Eric Rolph

Rainbow spectrum:

infrared red orange yellow green blue violet ultraviolet
780 700 600 550 475 370 nm

Radiometric Quantities

- *Radiant energy*
  Q in joule [J]

- *Radiant flux*
  $\Phi = \frac{dQ}{dt}$ in watts $[\mathrm{W{=}J/s}]$

- *Irradiance* (incident), *radiosity* (outgoing)
  $\mathrm{E} = \frac{d\Phi}{dA}$ power per area in $[\mathrm{W/m^2}]$

- *Radiance*
  $\mathrm{L} = \frac{d^2\Phi}{\cos\theta dA d\omega}$ power per solid angle $d\omega$ and per projected unit area $\cos\theta dA$ in $[\mathrm{W/(sr{\cdot}m^2)}]$

- Note
  - These quantities are in general wavelength-dependent
  - They ignore the polarization of light

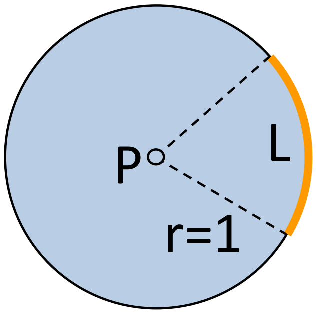
Radiometric Quantities

Intellectual game regarding radiant flux

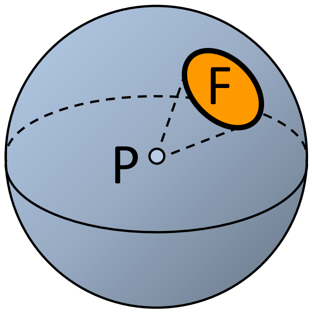
- Describes, e.g., how much energy is emitted by a light source or "how much energy is on the way"
  - 100 watts lightbulb $\rightarrow 1\ W = 1\ \mathrm{Js}^{-1}$
  - Of course less visible light is emitted (thermal loss etc.)
- Example: 100 W light source, emitting photons at 500 nm
  - Frequency of a photon $\nu = \frac{c}{\lambda} = \frac{3 \cdot 10^{8}\ \mathrm{ms}^{-1}}{500 \cdot 10^{-9}\ \mathrm{m}} = 6 \cdot 10^{14}\ \mathrm{s}^{-1}$
  - Energy of such a photon $h\nu \approx 4 \cdot 10^{-19}\ \mathrm{J}$
  - Number of photons $100\ \mathrm{W}/(4 \cdot 10^{-19}\ \mathrm{J}) \approx 2.5 \cdot 10^{20}\ \mathrm{s}^{-1}$

Angle and Solid Angle

- Plane angle (θ)
  - Unit: radian
  - The angle of a curve **K** seen from a point **P** is the arclength L of the projection of **K** onto the unit circle around **P**
  - Circle has 2π (rad) (360 degrees)
  - $\theta = L/r$



- Solid angle (Ω)
  - Unit: steradian (1 sr is the area of 1 m^2 on a sphere with radius 1 m)
  - Solid angle of a surface **F** seen from a point **P** is the area of the projection of **F** onto the unit sphere around **P**
  - Sphere has a solid angle of 4π (sr) (approx 40 000 degrees^2)
  - $\Omega = F/r^2$



Why do I need to know that?

Photorealistic images of virtual scenes are generated by simulating light transport

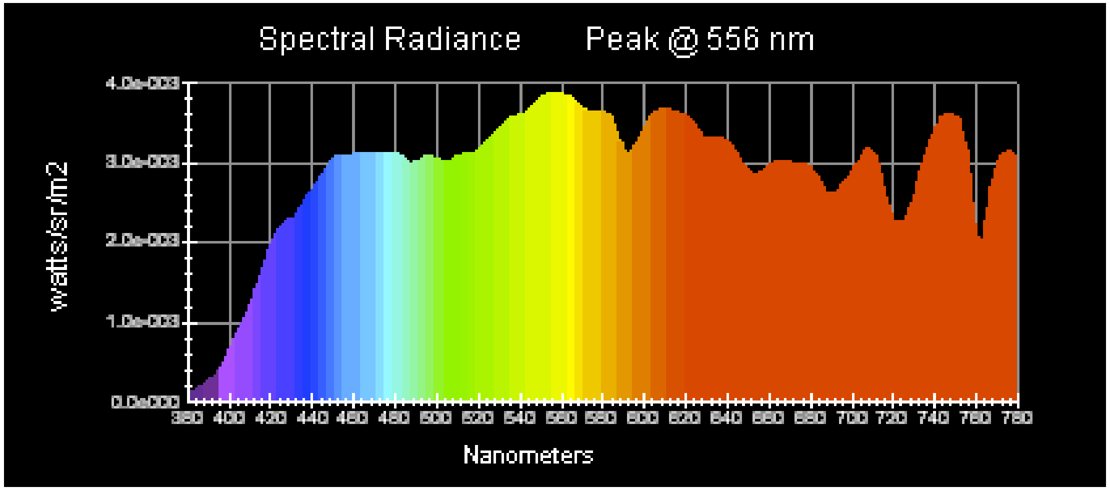
Example: Physically Based Rendering

image: Gianni Melis, computed with “fryrender”

Illumination

- Spectral composition of light
  - Consisting of many wavelengths
  - Each wavelength with a certain intensity
  - $P(\lambda)$ is the radiant power at wavelength λ

  - Example: daylight



- The perception of such distributions is a product of the human visual system (HVS)

Other Colors

- Most colors are a mixture of (many) wavelengths

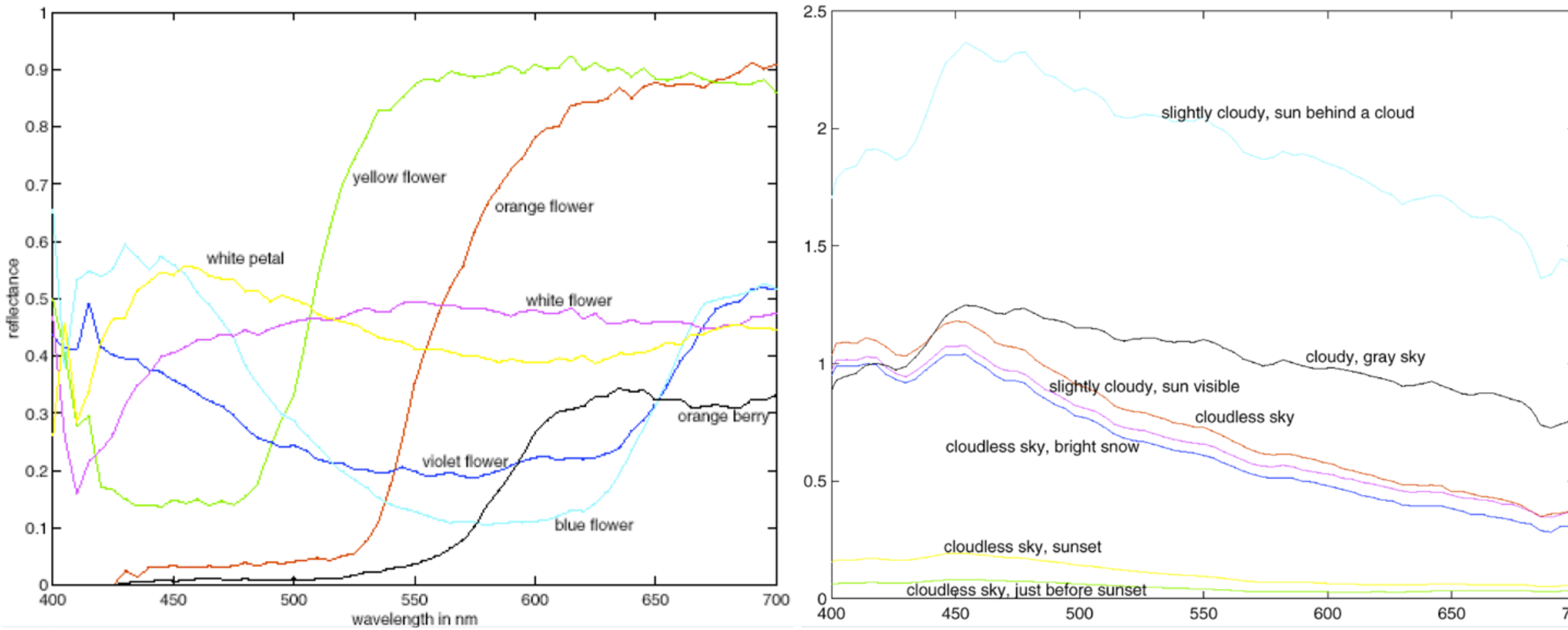


image: David Forsyth

Other Colors

- Most colors are a mixture of (many) wavelengths

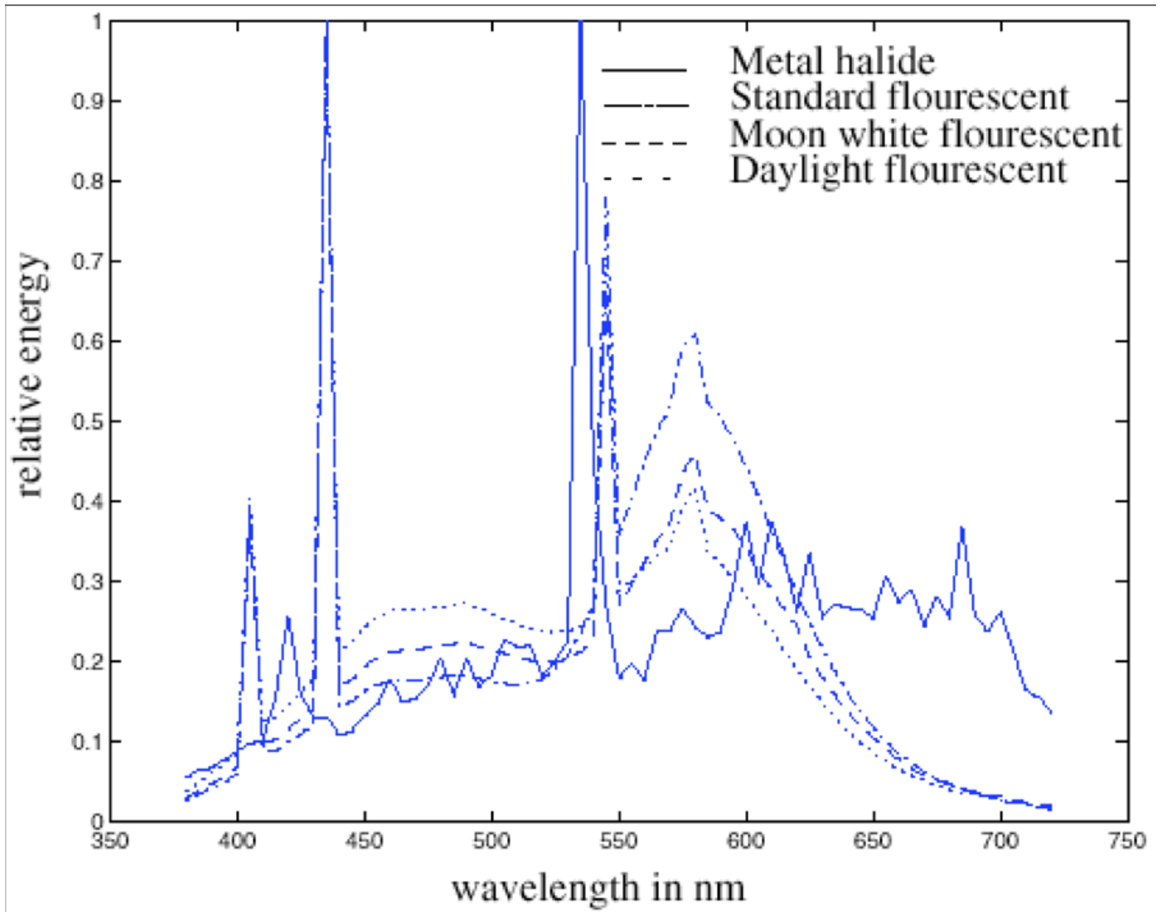


image: David Forsyth

Color Spectra

continuous spectrum of an incandescent light

line spectra of

Na

H

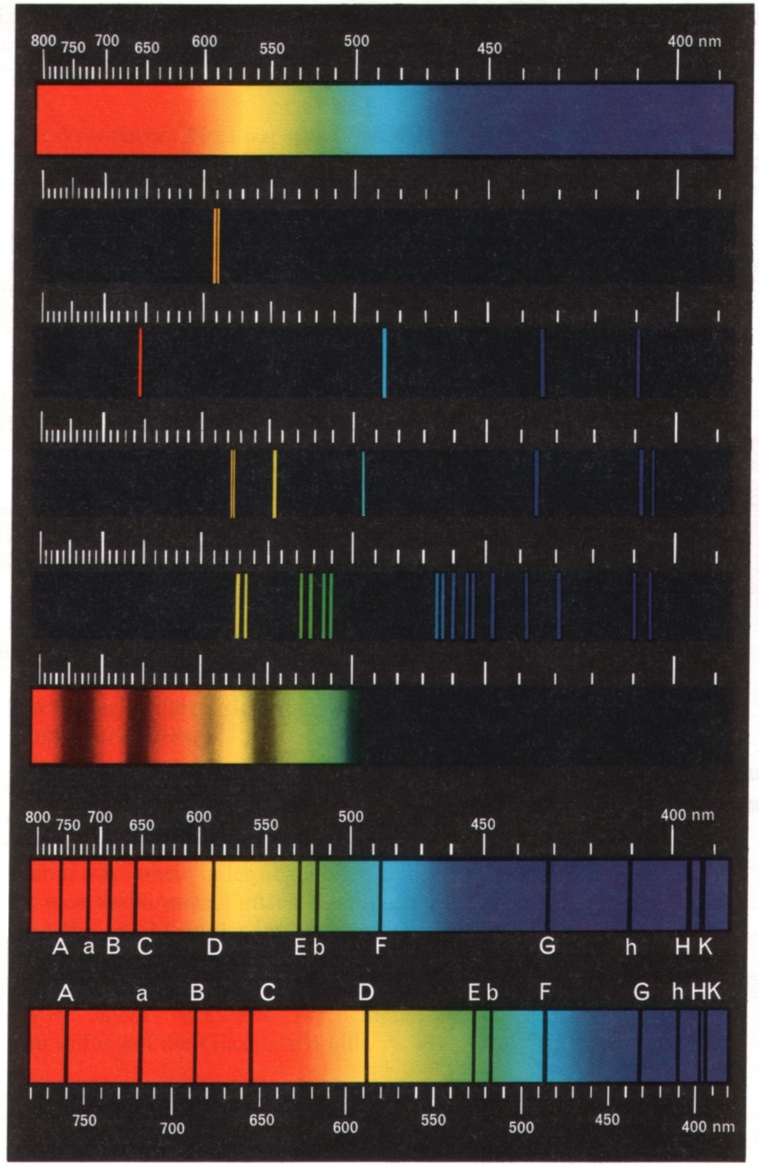
Hg

Cu

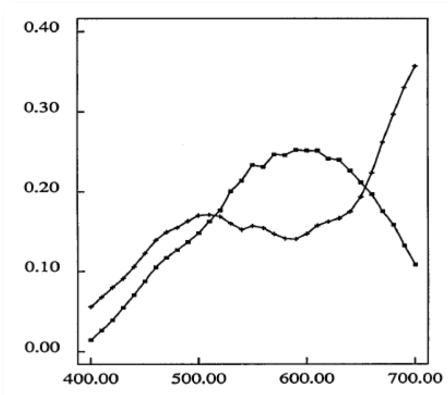
absorption spectrum of chlorophyll

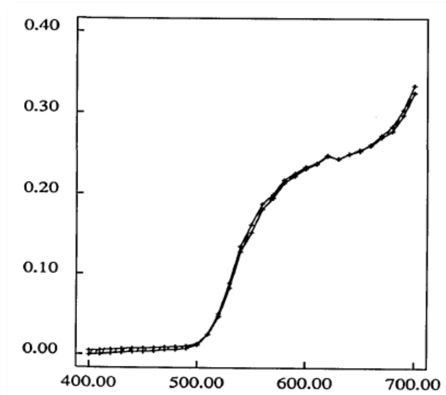
prism spectrum of sunlight

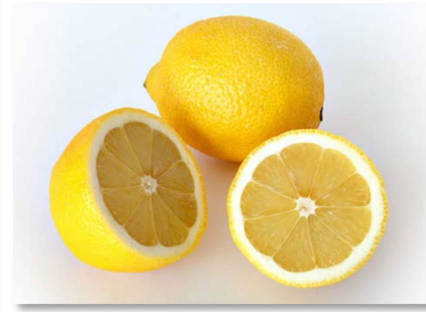
grating spectrum of sunlight

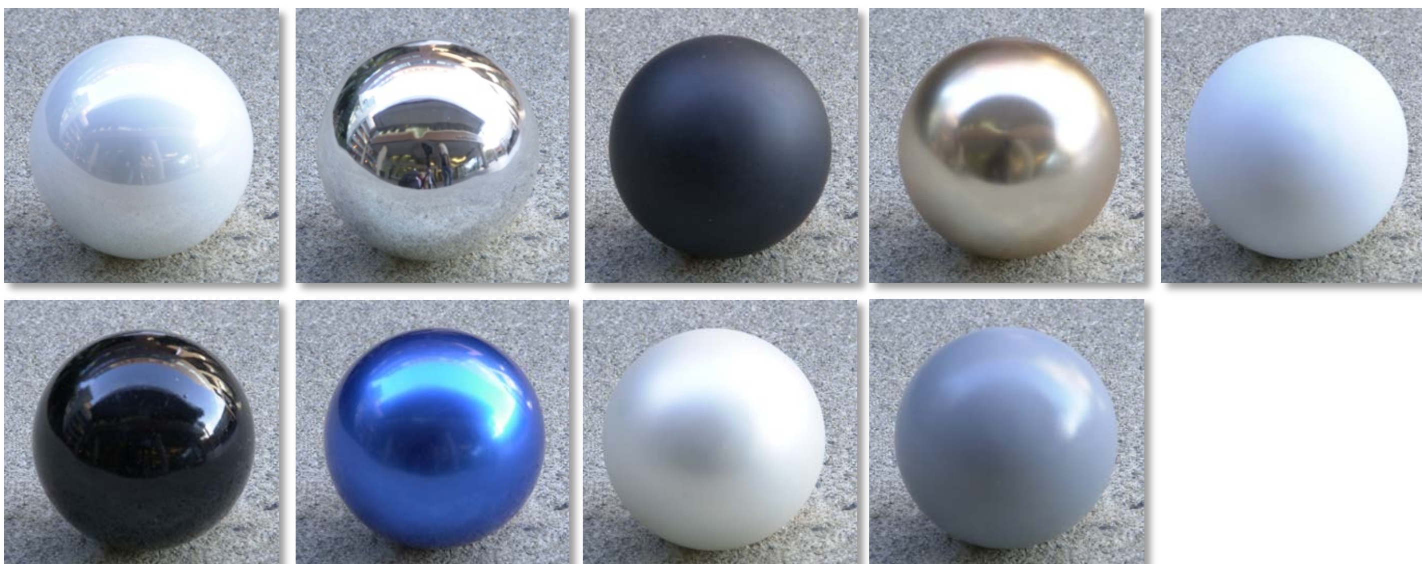


Color and Structure









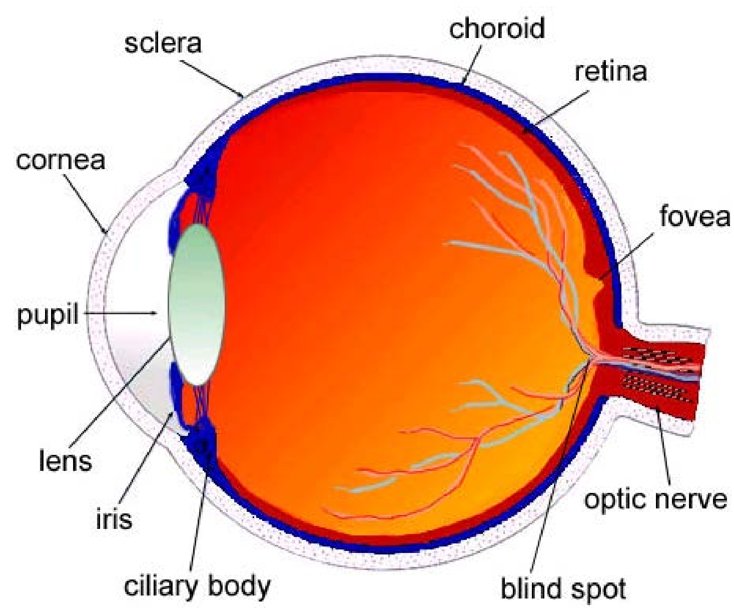
Perception vs. Measurement

- Human visual system
  - The human eye cannot capture the spectral composition of light
  - The eye does limited measurements
  - The eye adapts (physically) to exterior conditions
  - … the brain also does, in various ways …

- **Perceptual psychology**: Examination of the principles of sensory perception
  - Sensory receptors, signaling, … up to the processing of the stimuli in the central nervous system
  - The perception process is not (yet) completely understood!

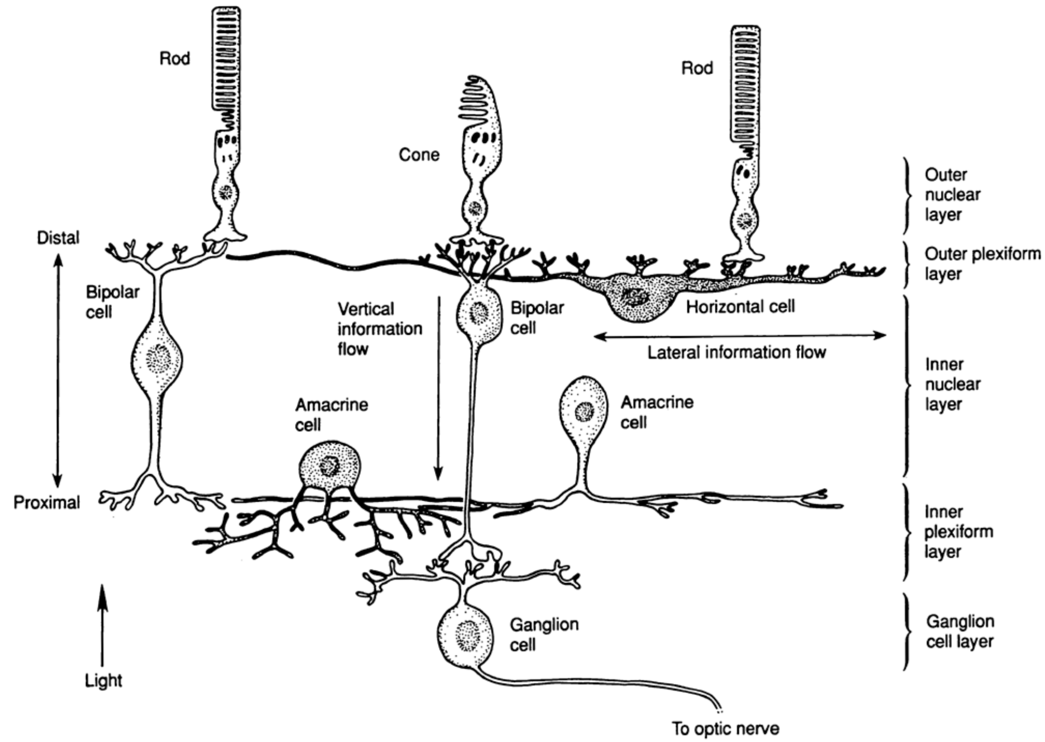
Human Eye

- Cornea and lens
  - Focus incoming light on retina
  - Refraction, focal length approx. 14–17 mm
- Pupil
  - Adjusts incoming light with pigmented iris
- Ciliary muscle: stretches lens, adjusts focal length
- Retina
  - Millions of light-sensitive cells
  - Chemical reaction, signal generation
  - Fovea centralis:
    Region of sharpest vision
    (approx. 1.5 mm diameter,
    50–60 thousand light-sensitive cells)
  - Blind spot

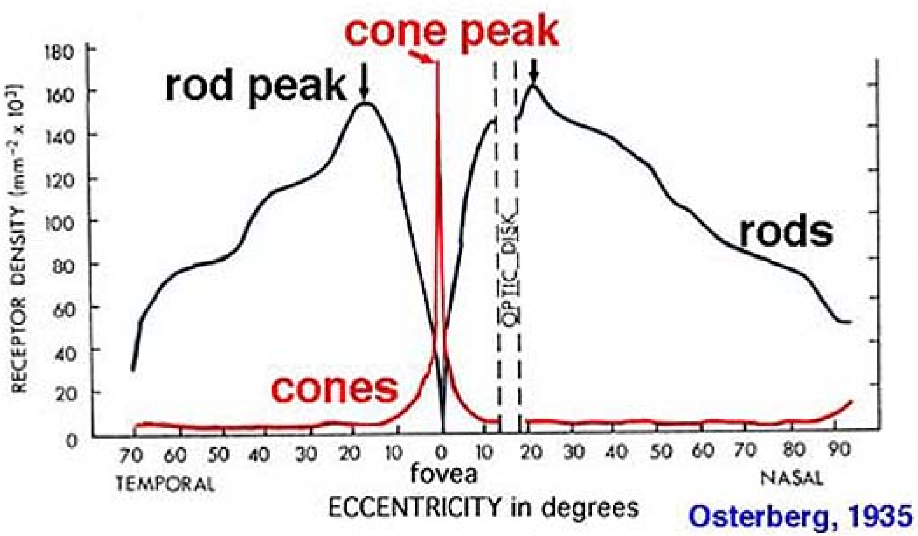


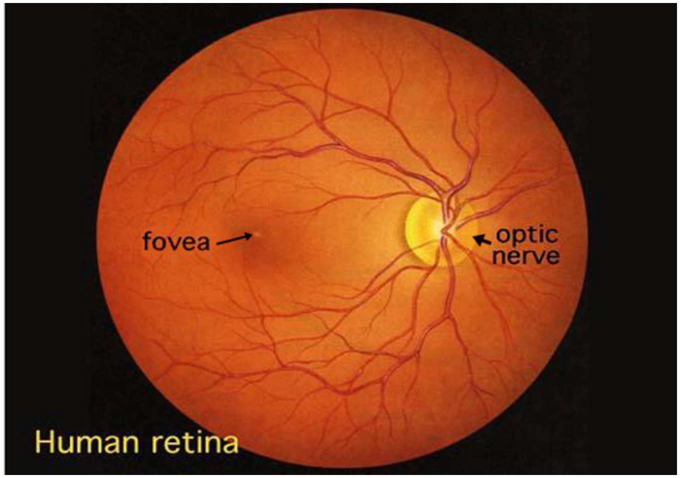
Receptors in Human Eye

- 2 types of receptors in retina
  - Cones
    (central, few,
    3 types for colors)
  - Rods
    (peripheral, many,
    black-and-white vision)



[Kandel et al.: Principles of neural science]

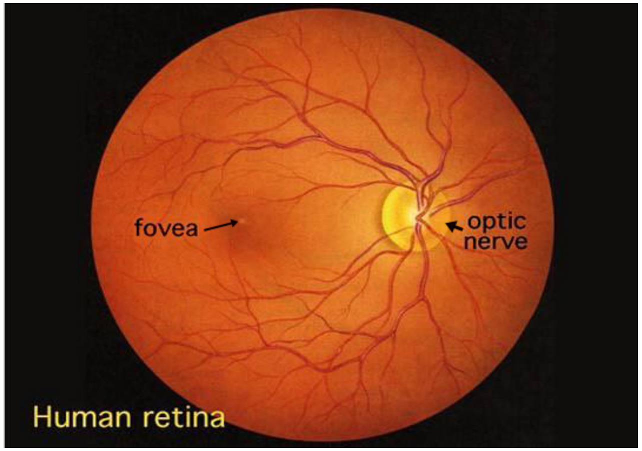




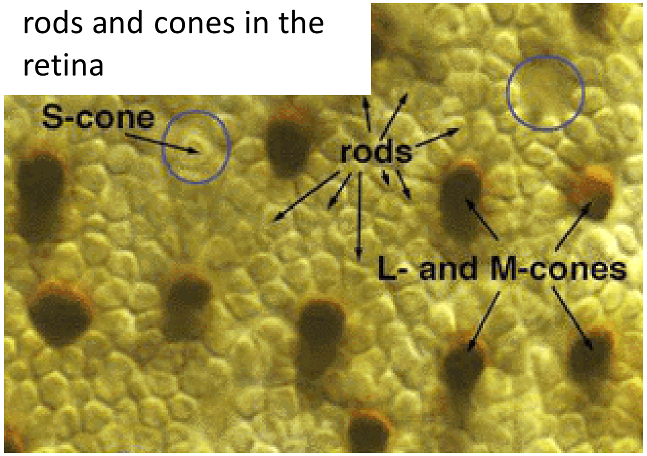
The Eye as a Sensor – Receptors

Light-sensitive receptors in the human eye

- Cones
  - Photopic vision (day) and trichromatic color vision
  - 3 types: different sensitivity (counts) w.r.t. light spectra, approx.:
    - S (**7%**) correspond to blue
    - M (37%) correspond to green
    - L (56%) correspond to red
    - Approx. 6–7 M concentrated around fovea, less sensitive
    - Very few females have fourth “yellow/orange” cone



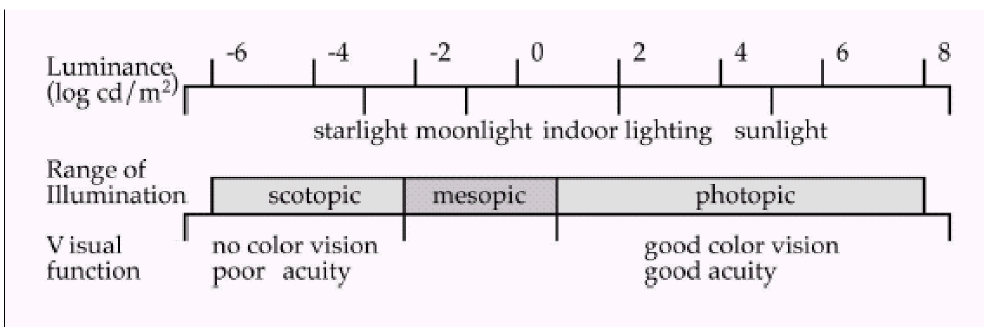
rods and cones in the retina



The Eye as a Sensor – Receptors

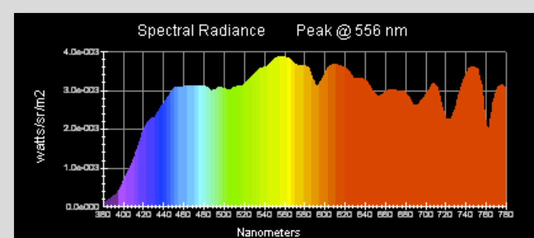
Light-sensitive receptors in the human eye

- Rods
  - Scotopic vision (night), monochromatic
  - Everywhere on retina, dominant in peripheral region (20°)
  - Approx. 120 M
  - 1000× more light-sensitive than cones—maximum at 555 nm (green)



[Ferwerda, James A., IEEE Computer Graphics and Applications, 5, 2001, pp. 22]

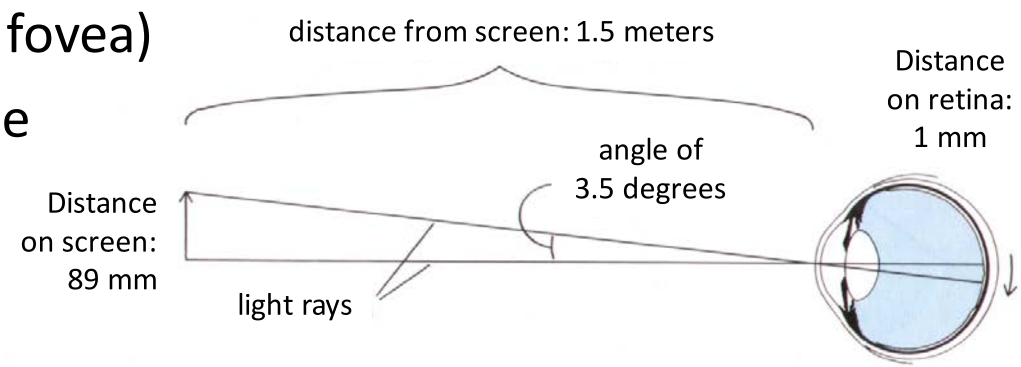
555 nm photopic peak is not a result of evolutionary optimization to sunlight peak at 556 nm
(Some paradoxes, errors, and resolutions …, Soffer and Lynch, 1999)



The Eye as an Optical System

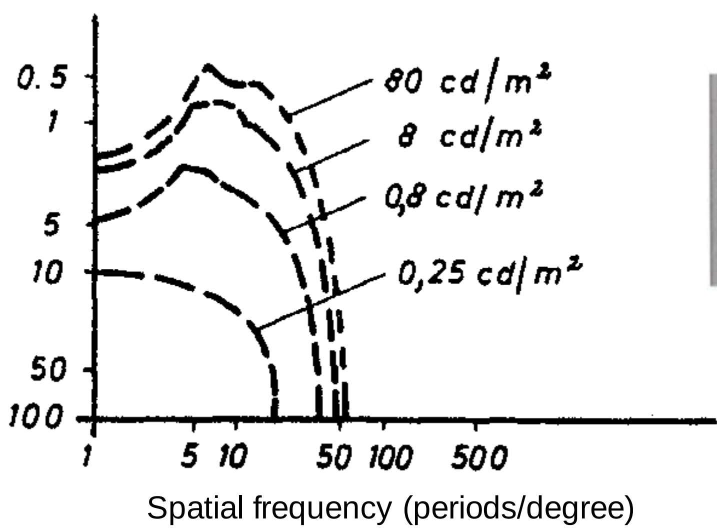
Foveola (approx. 0.35 mm diameter)

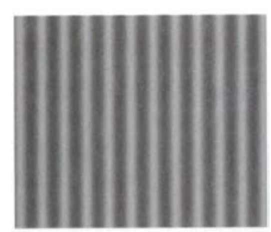
- Area of sharpest vision (within fovea)
- Approx. 1 cm^2 at 57 cm distance (approx. 1 degree)

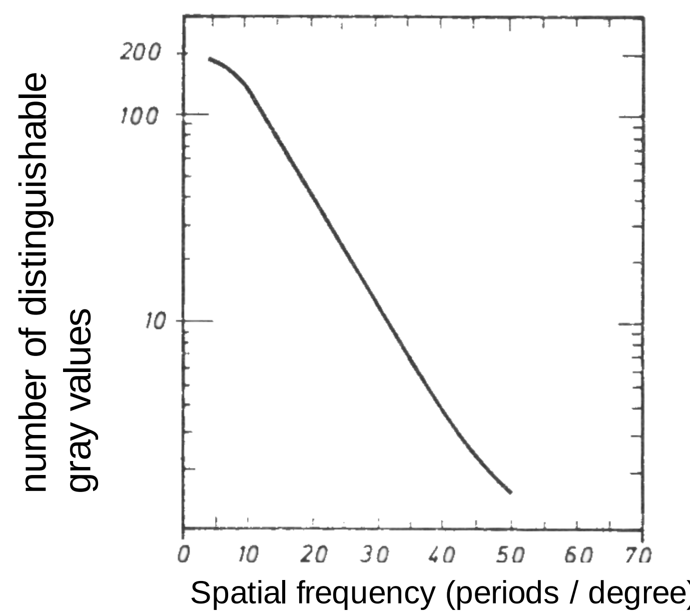


Contrast sensitivity

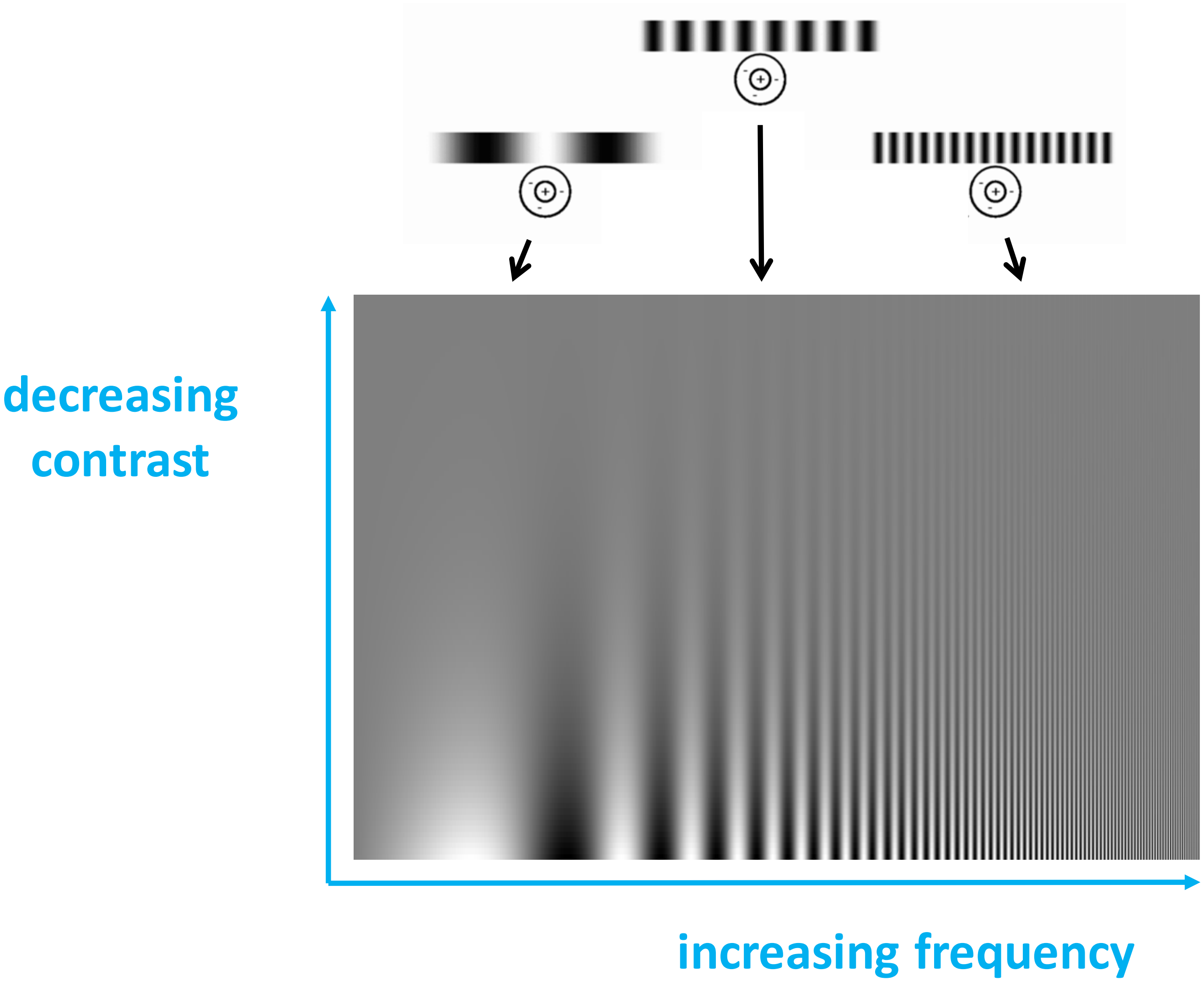
- Best at 5 periods/degree (3–4 mm at 1 m distance)
- Depends on brightness





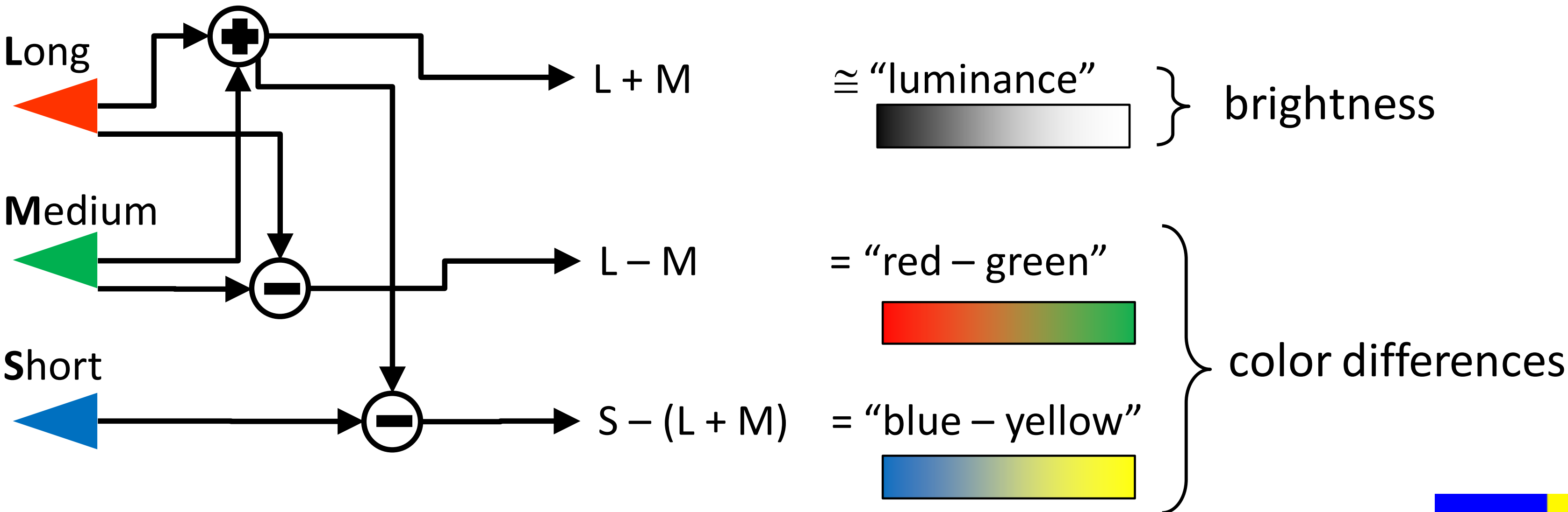


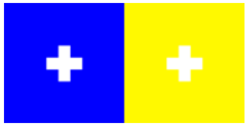
Contrast Sensitivity Function (CSF)



Opponent Process

- Color signals from cones are combined into two pairs of opponent colors and one brightness value (no RBG framebuffer in brain!):
- 2 chromatic and 1 achromatic channel (still 3-dimensional)
- The perceived brightness does not increase linearly with incident luminance

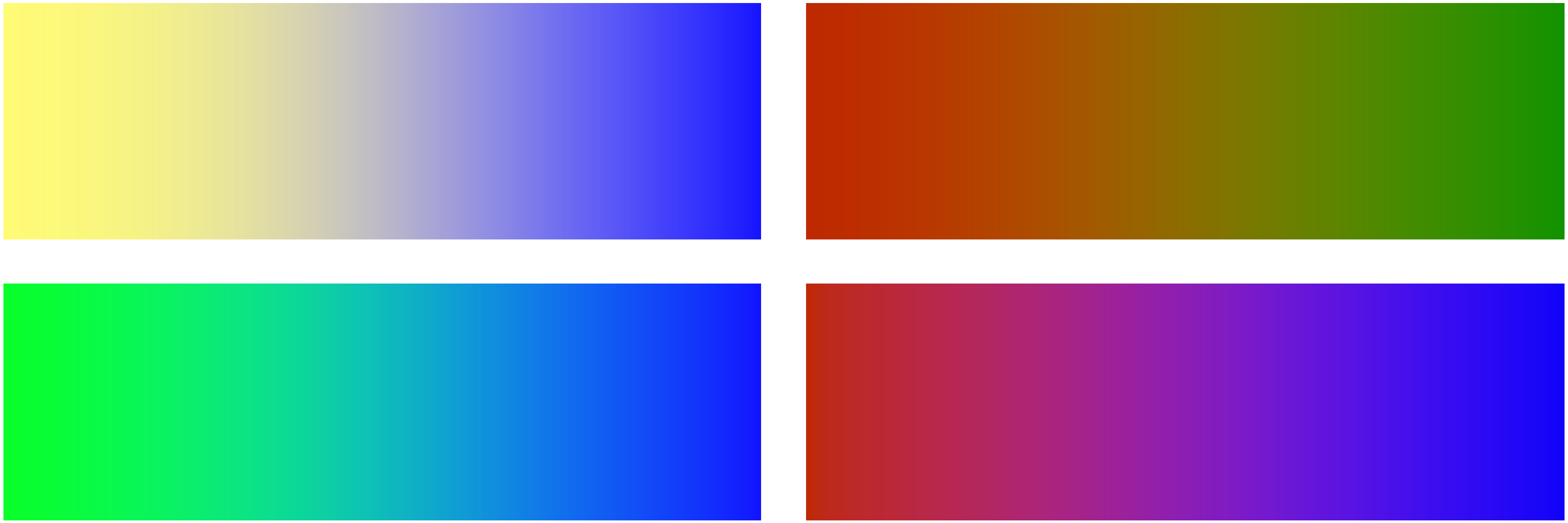




en.wikipedia.org/wiki/Impossible_color

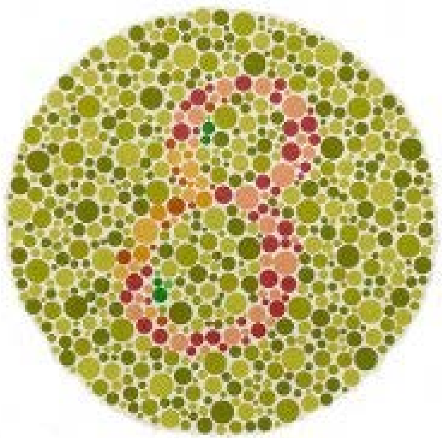
- There is reddish yellow, but no reddish green
- Oversaturation of a channel leads to complementary color
- 10% of males suffer from dyschromatopsia (red–green blindness)

Opponent Process

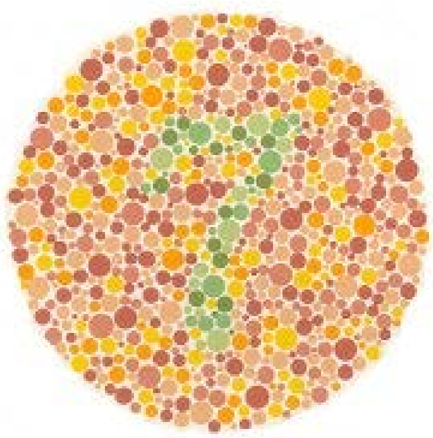


Color Blindness

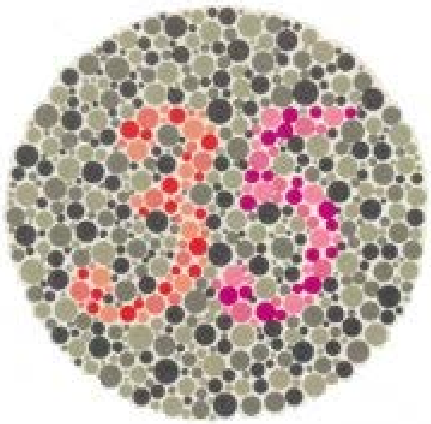
- Color blindness at isoluminance
- Red–green blindness: Hereditary disease
    - Approx. 10 percent of males, 1 percent of females



Normal color viewing: 8
Red–green blind: 3 or nothing



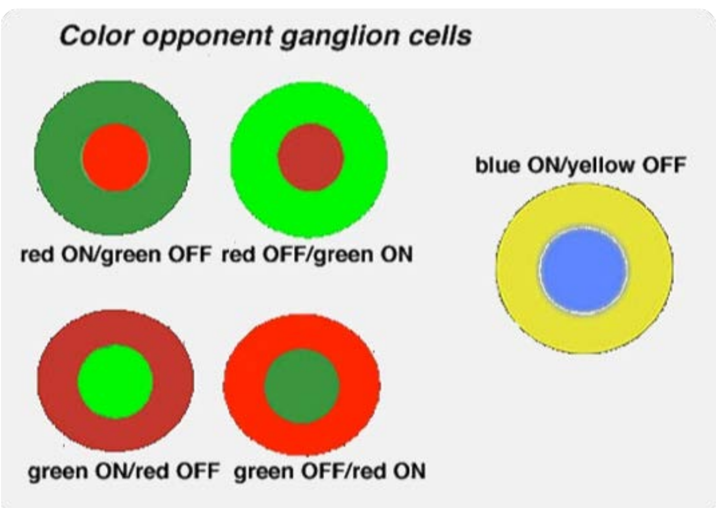
Normal color viewing: 7
Color blind: nothing

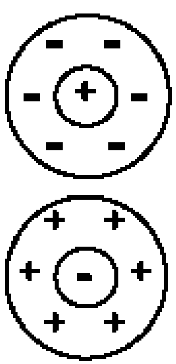


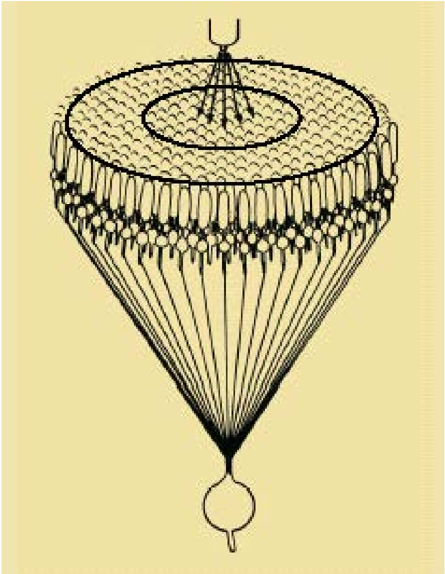
Normal color viewing: 35
Red blind: 5
Green blind: 3

Receptive Fields

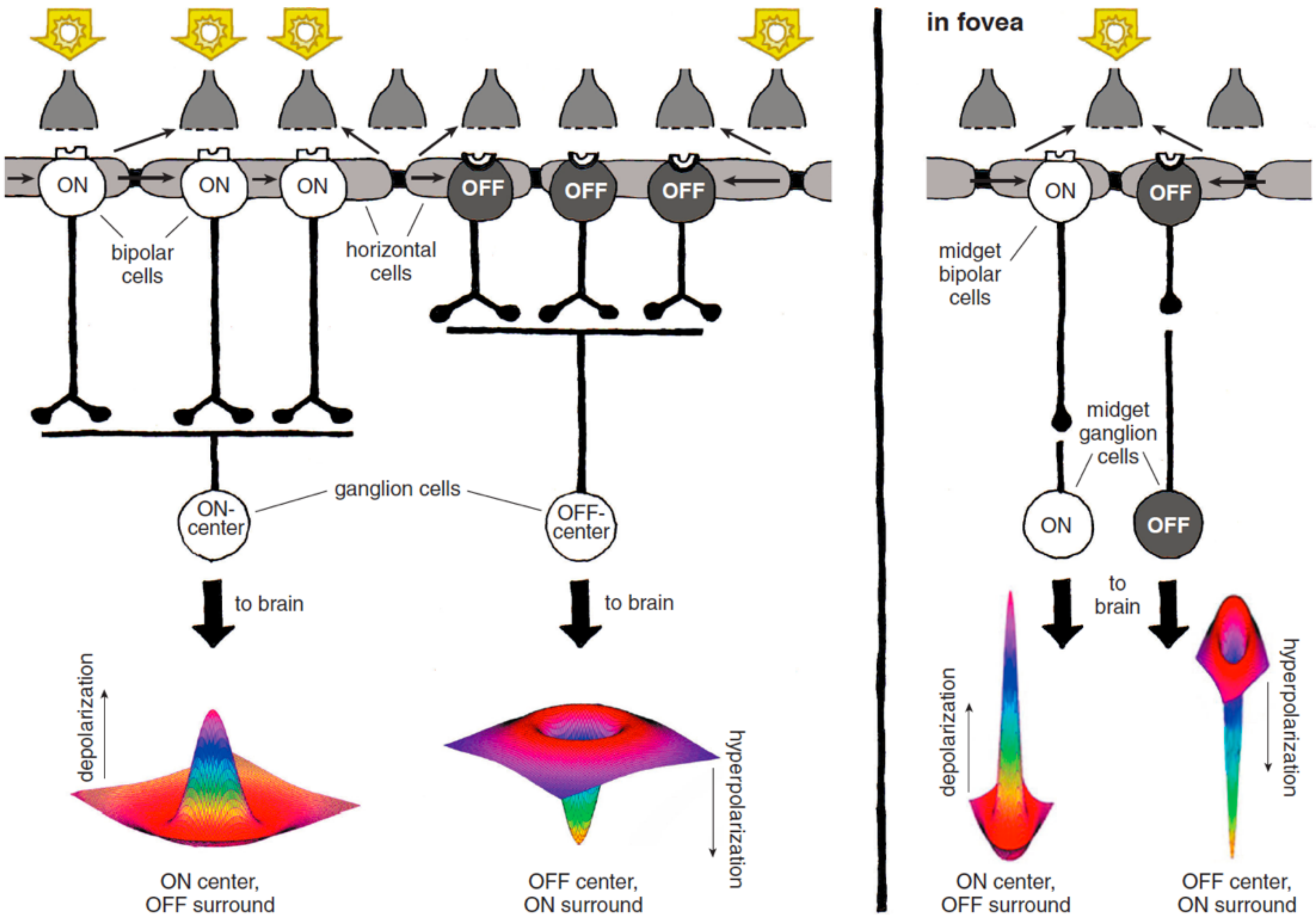
- Approx. 130 million photoreceptors are connected with approx. 1 million ganglion cells
- They form circularly receptive fields of different sizes
- Two types:
  - ON: excitation in the center, inhibition in periphery
  - OFF: excitation in periphery, inhibition in center
- Lateral inhibition
- They react to difference in intensity or color
- Recognition of orientation and texture





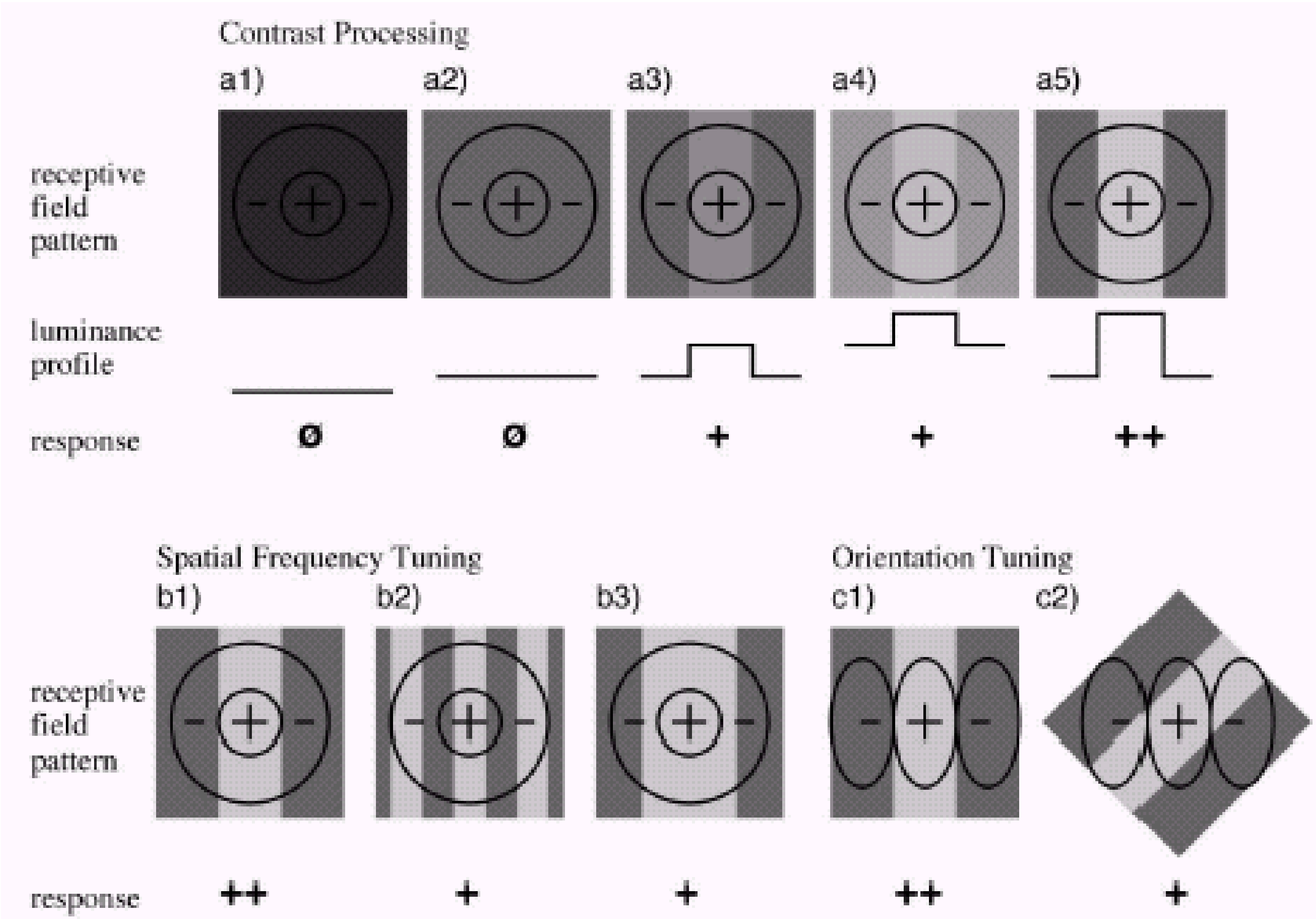


Receptive Fields

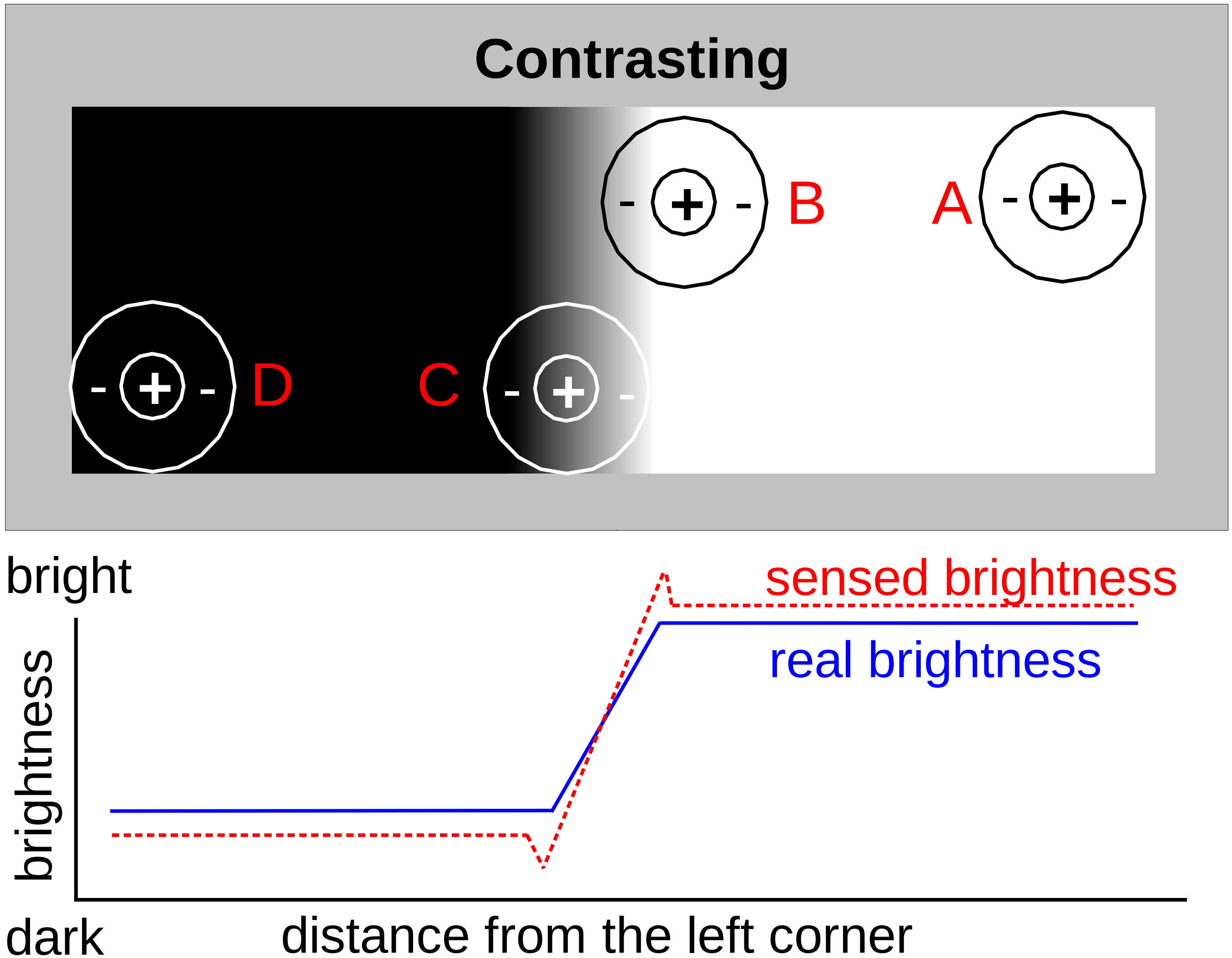


Kolb, 2003

Receptive Fields

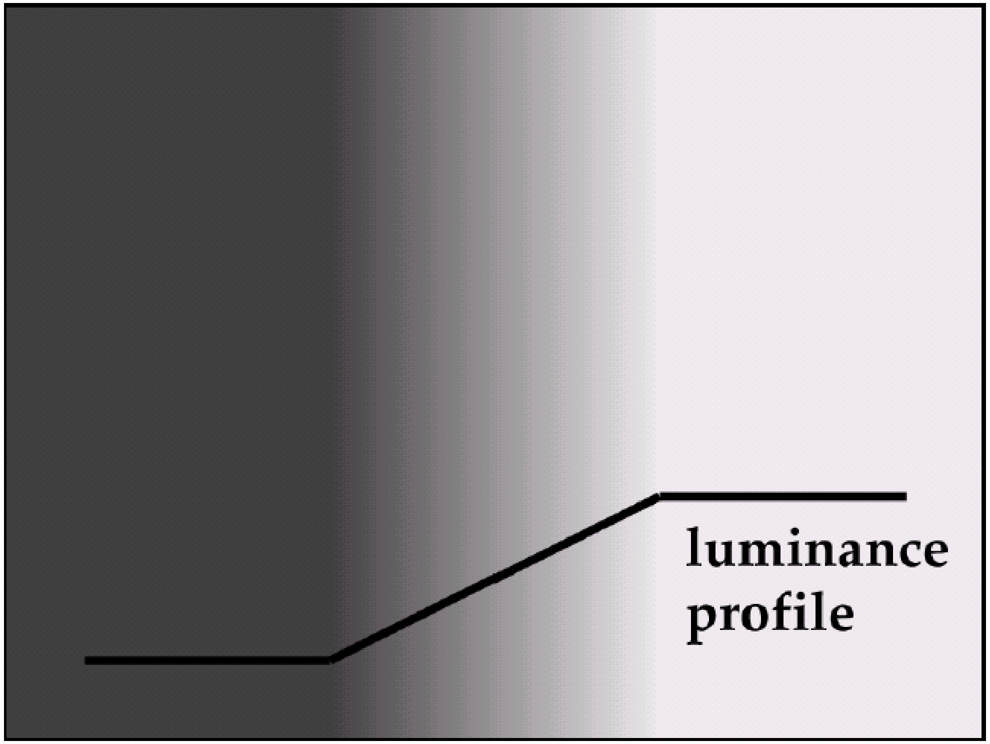


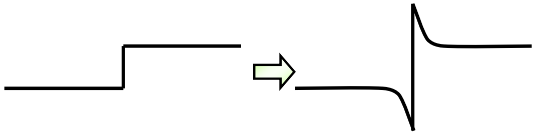
Receptive Fields – Contrast Enhancement



Receptive Fields

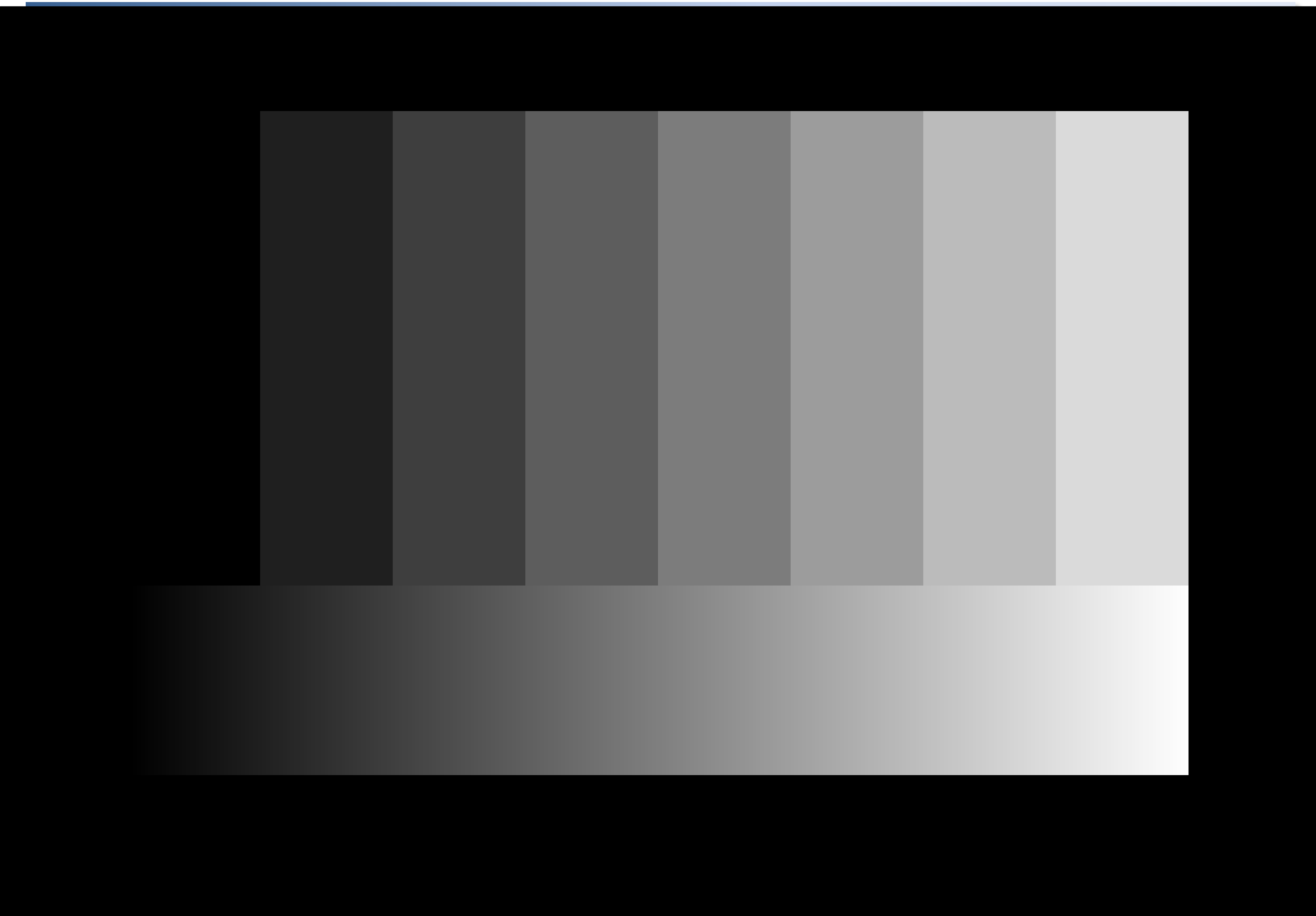
- Mach bands
  - Edge recognition system increases luminance gradient





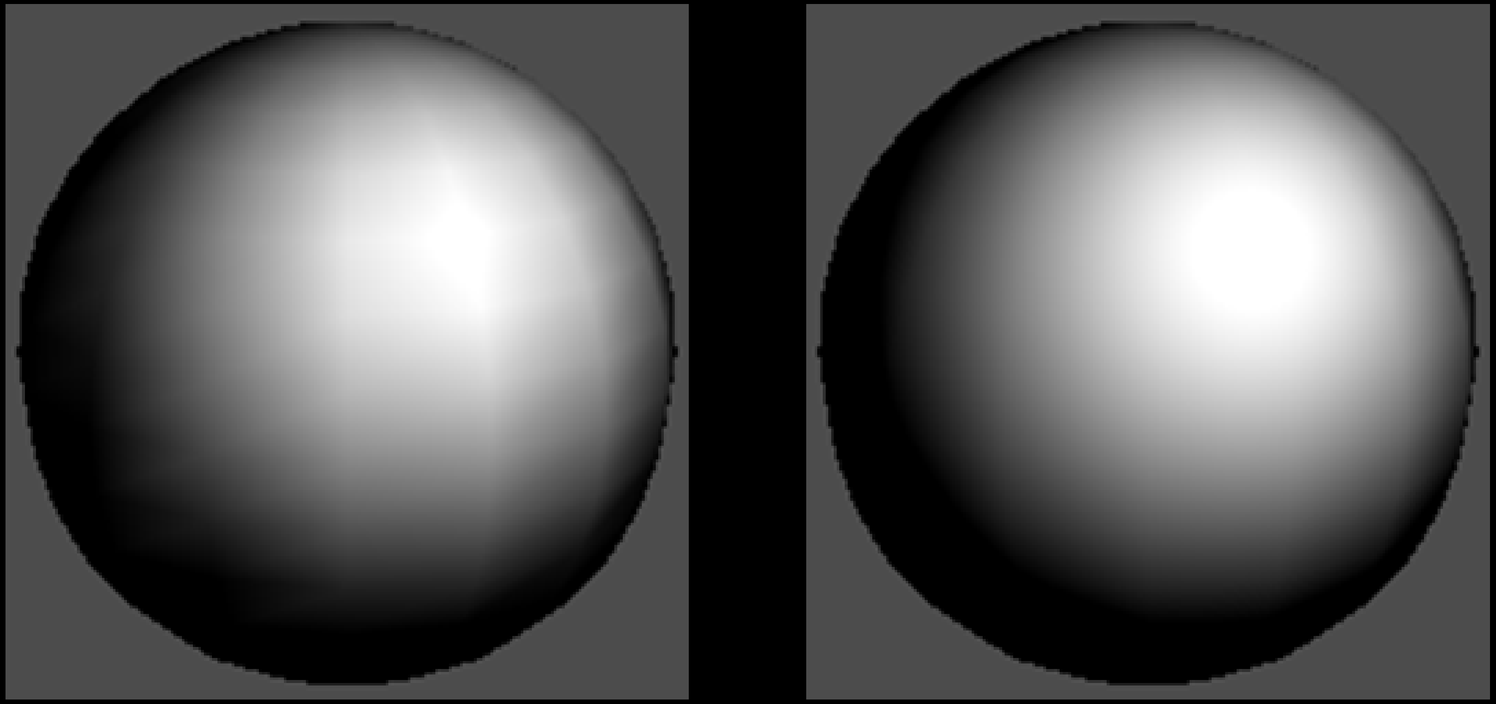
[Ferwerda, James A., IEEE Computer Graphics and Applications, 5, 2001, pp. 22]

Mach Bands



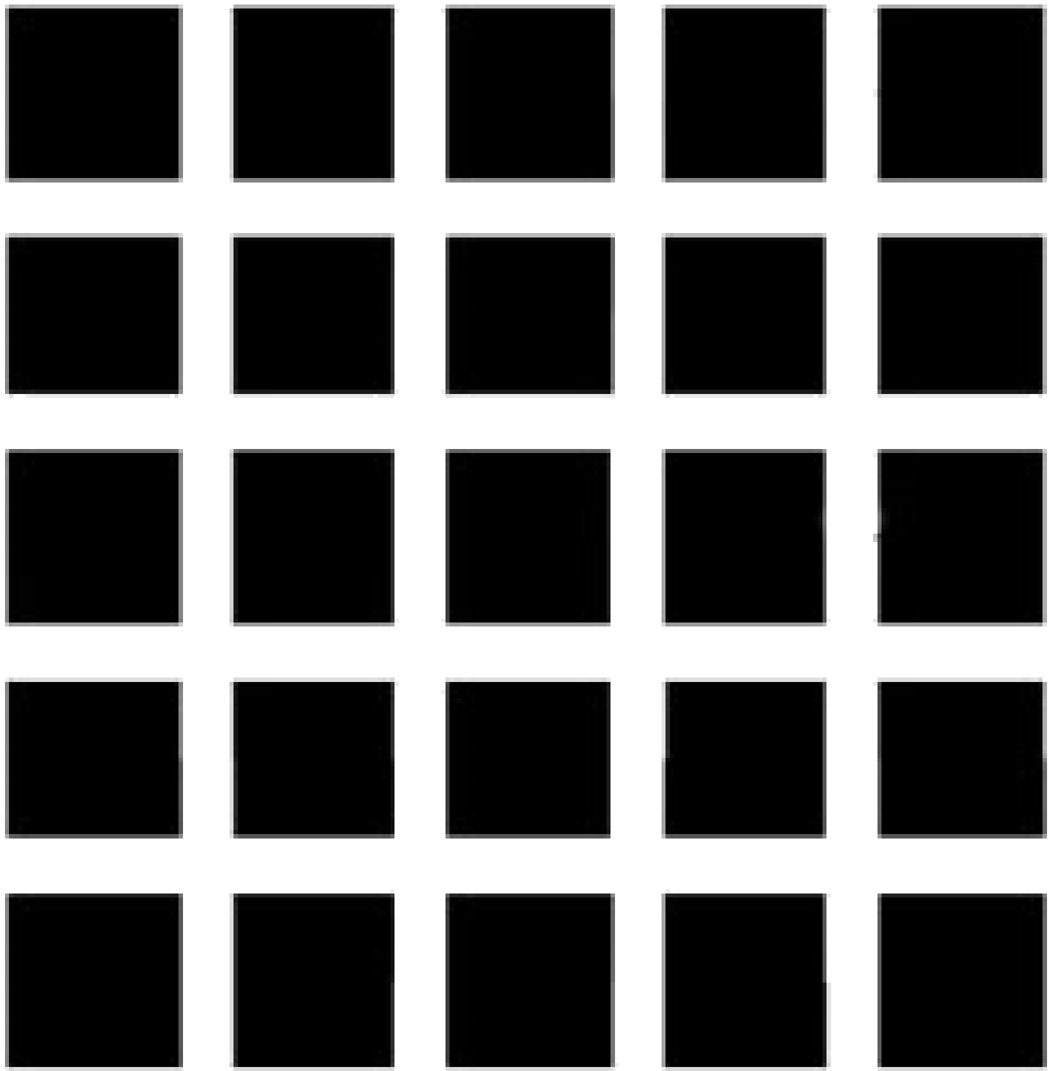
Mach Bands

Mach Bands



Hermann Grid

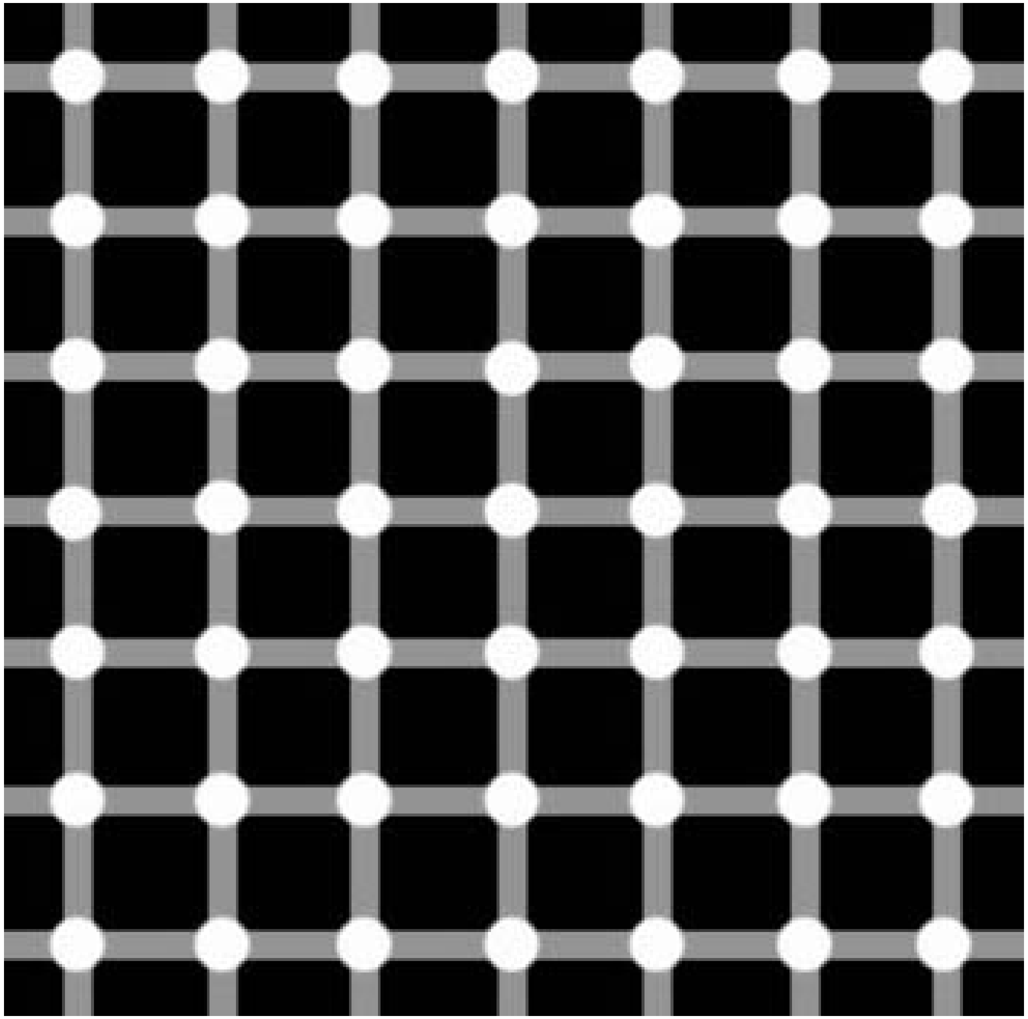
- Due to lateral inhibition

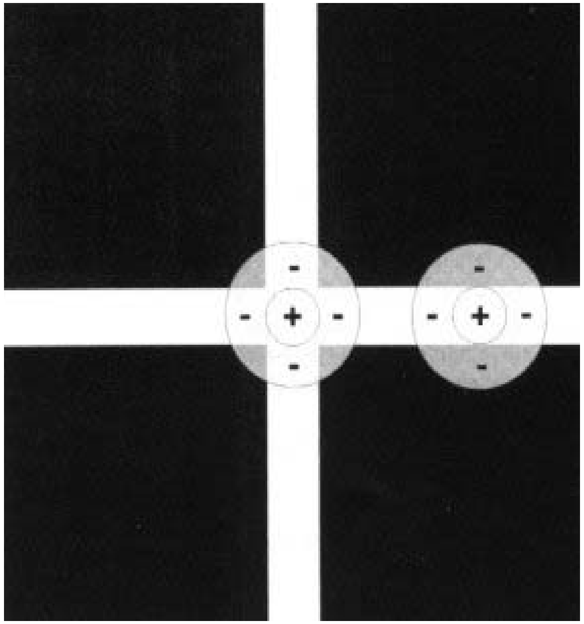


Hermann Grid

- Variant of Hermann grid

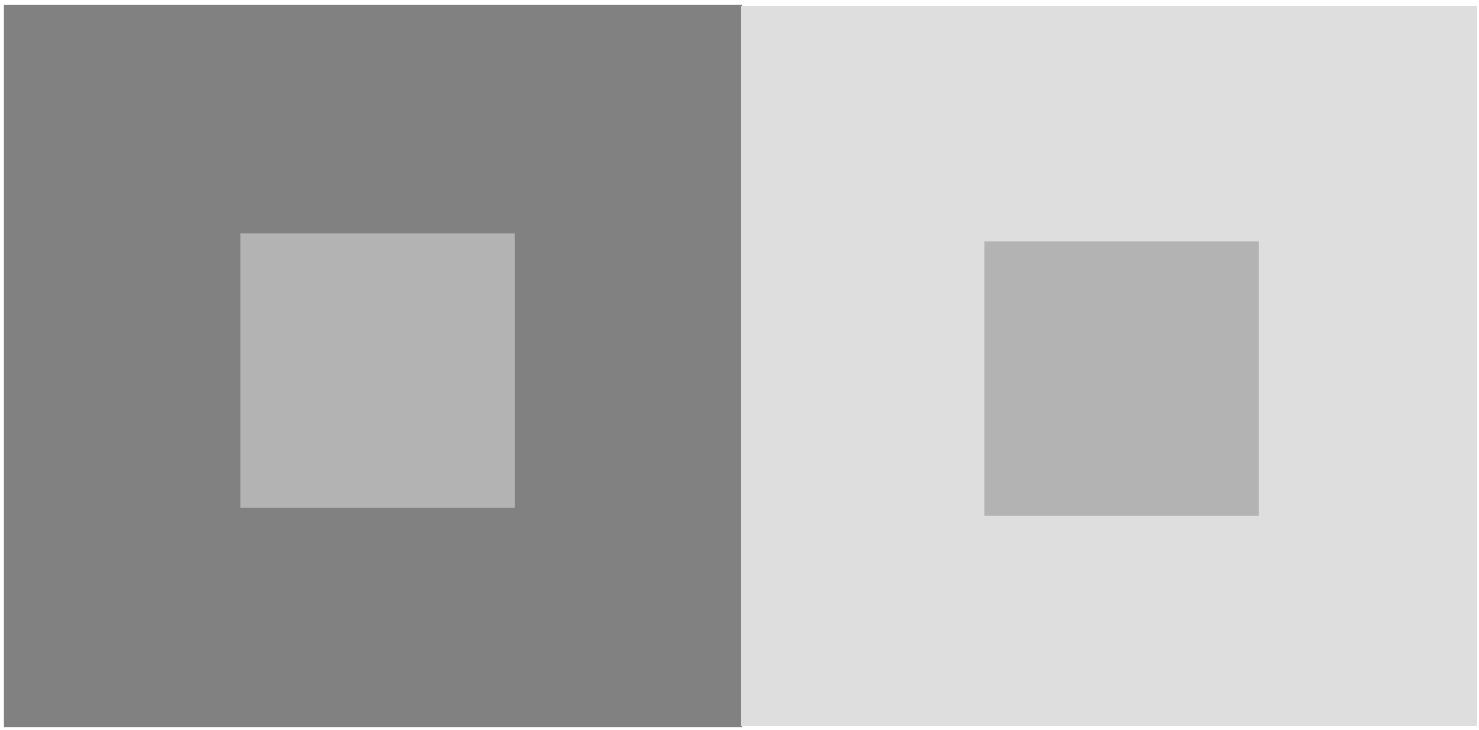
  [http://www.physics.utoledo.edu/~lsa/_color/27_pattern.htm]

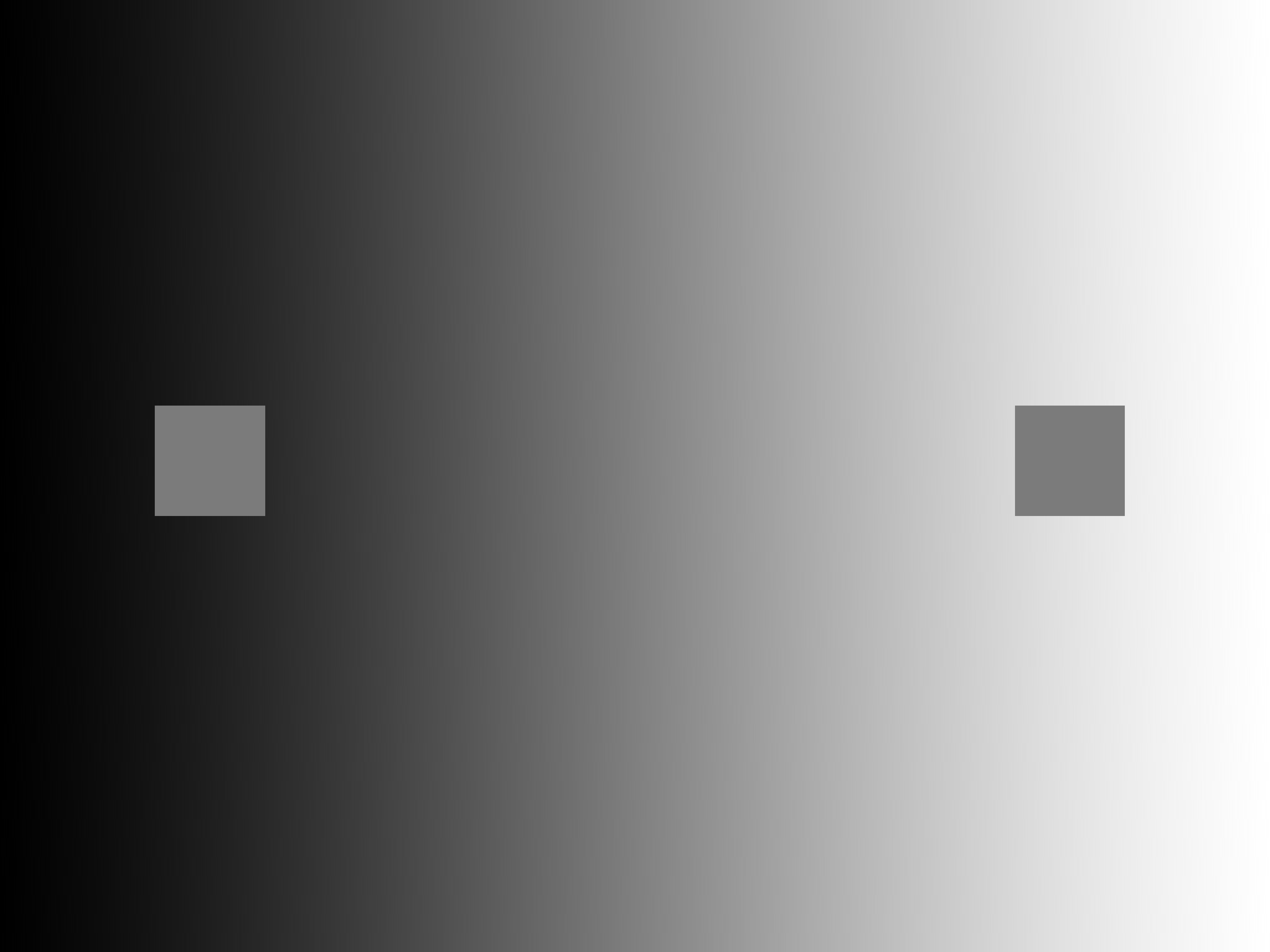


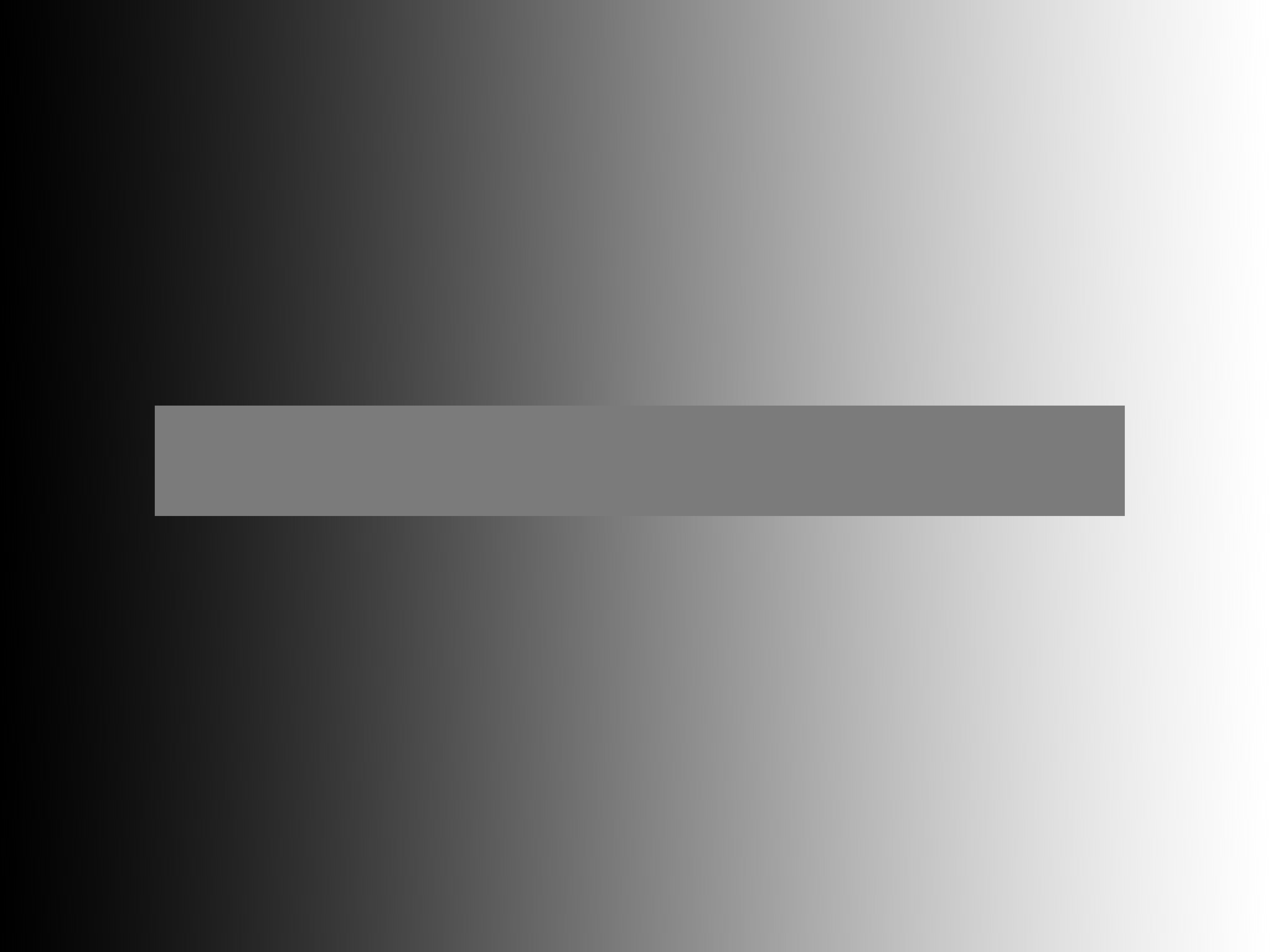


Perception

- Simultaneous contrast



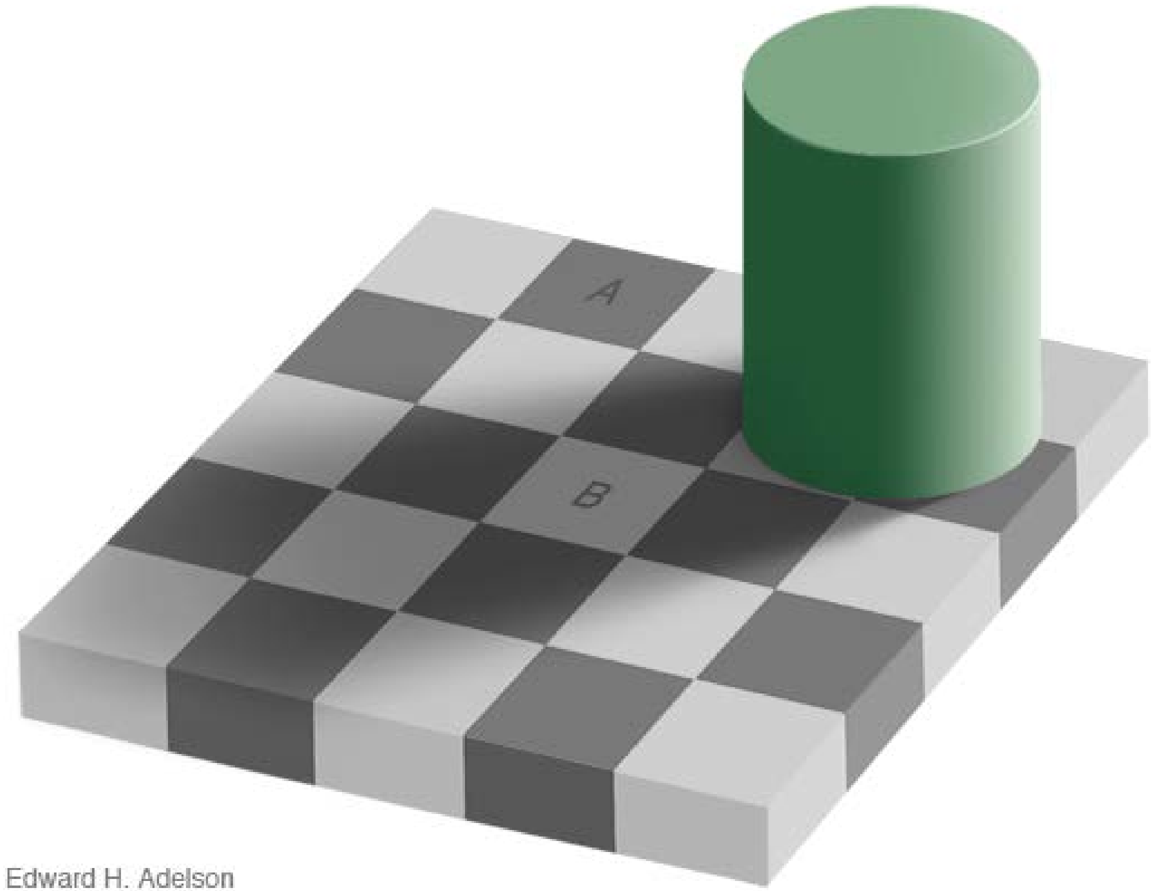




Perception

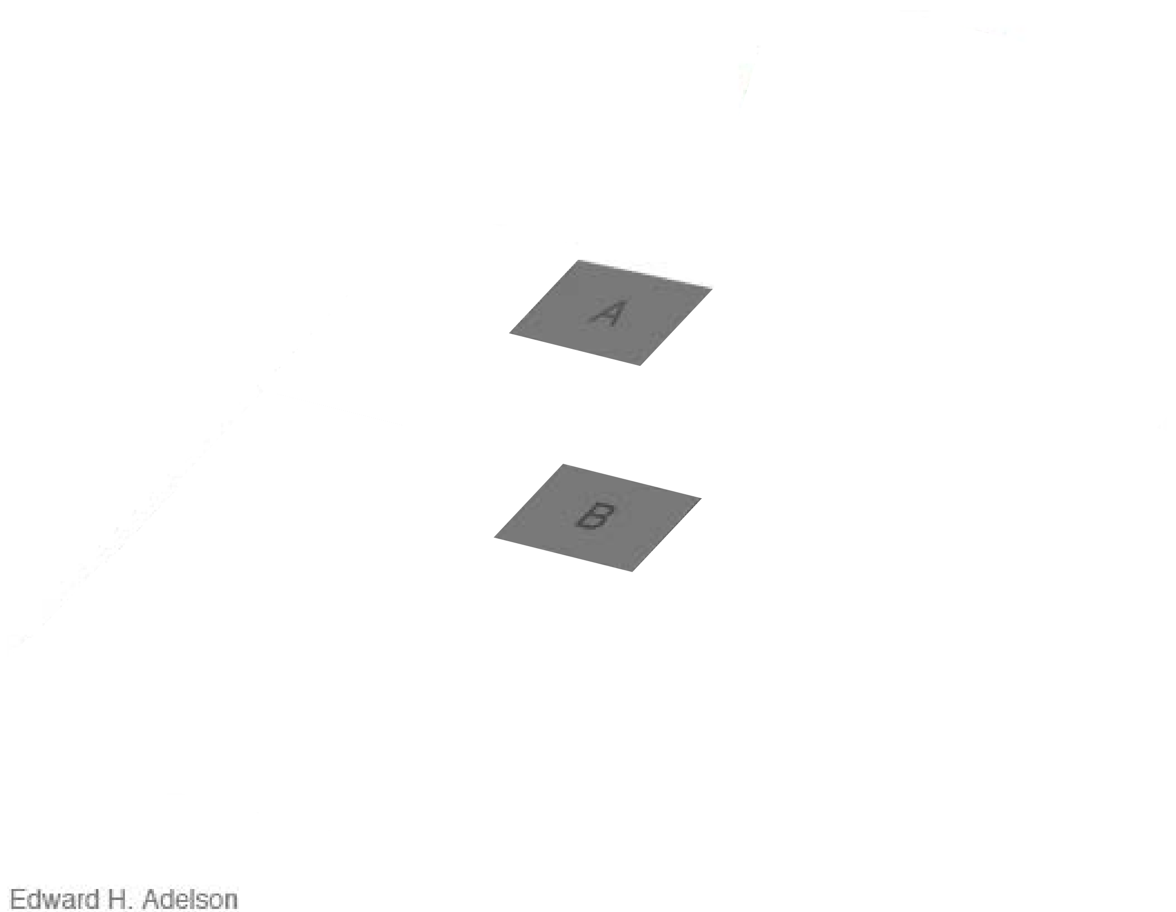
Perception

- Adelson‘s checkerboard

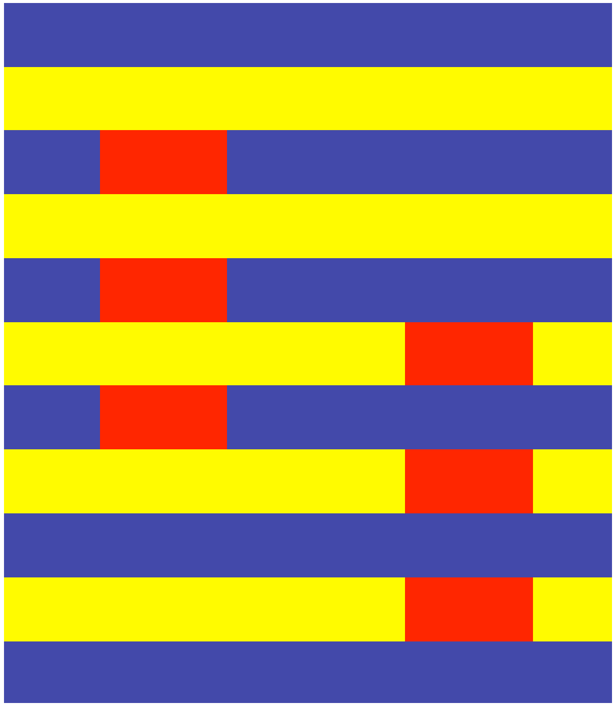


Perception

- Adelson‘s checkerboard:
  A and *B* have identical gray value



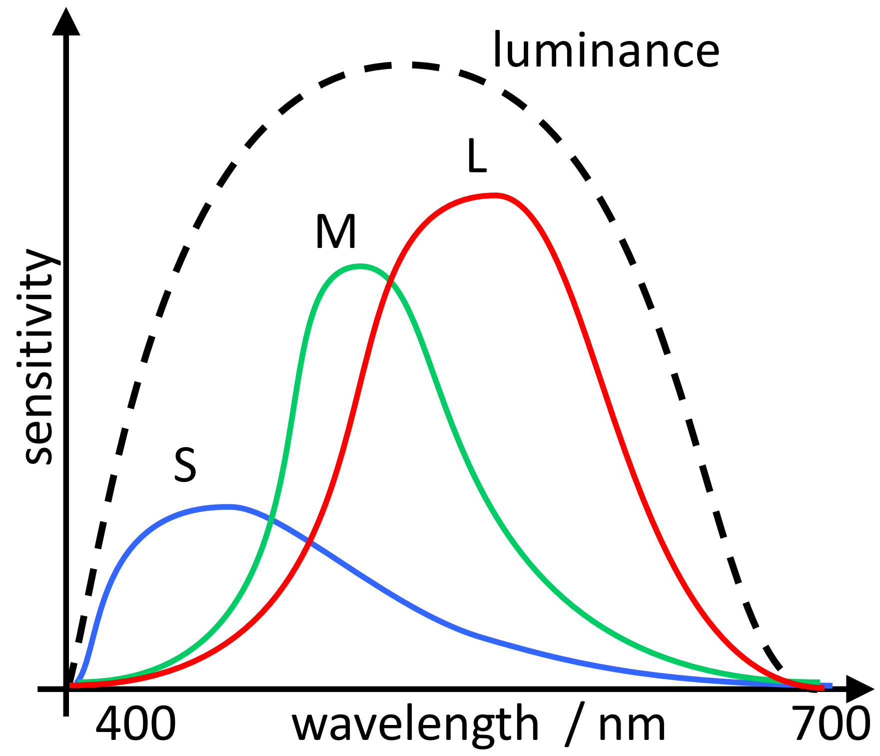
- Question: what color is [255,0,0]?
- Answer: **red**



Trichromatic Color Vision

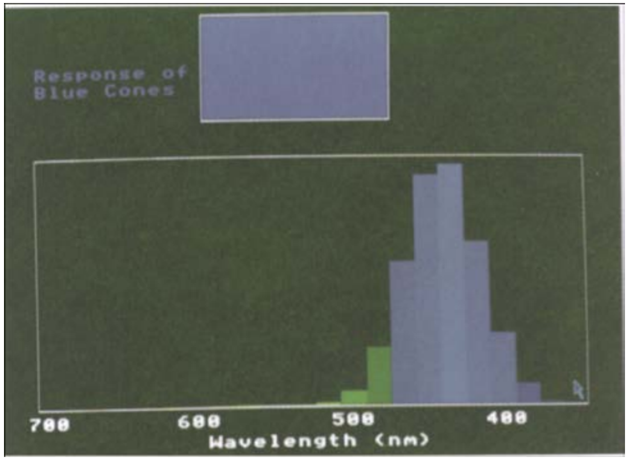
- Perceptual response to light of different wavelength:

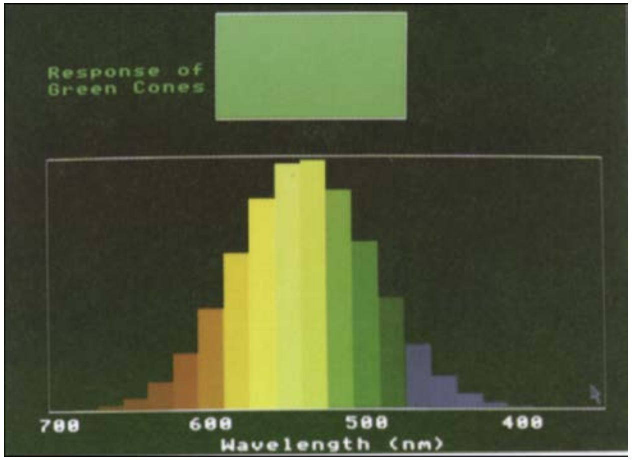
$$\mathrm{s} = \int S(\lambda)P(\lambda)d\lambda \qquad \mathrm{m} = \int M(\lambda)P(\lambda)d\lambda \qquad \mathrm{l} = \int L(\lambda)P(\lambda)d\lambda$$

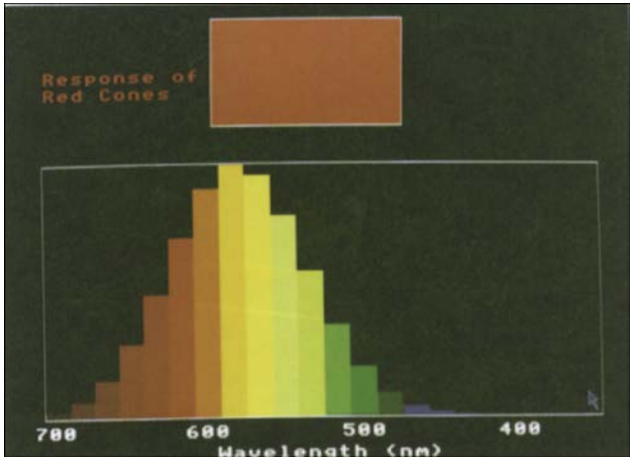


Trichromatic Color Vision

- Response to S-receptor roughly corresponds to blue light
- M- and L-receptors cover larger wavelength ranges
  - I.e., not only “green” and “red”
  - Rather yellow–green and orange–red

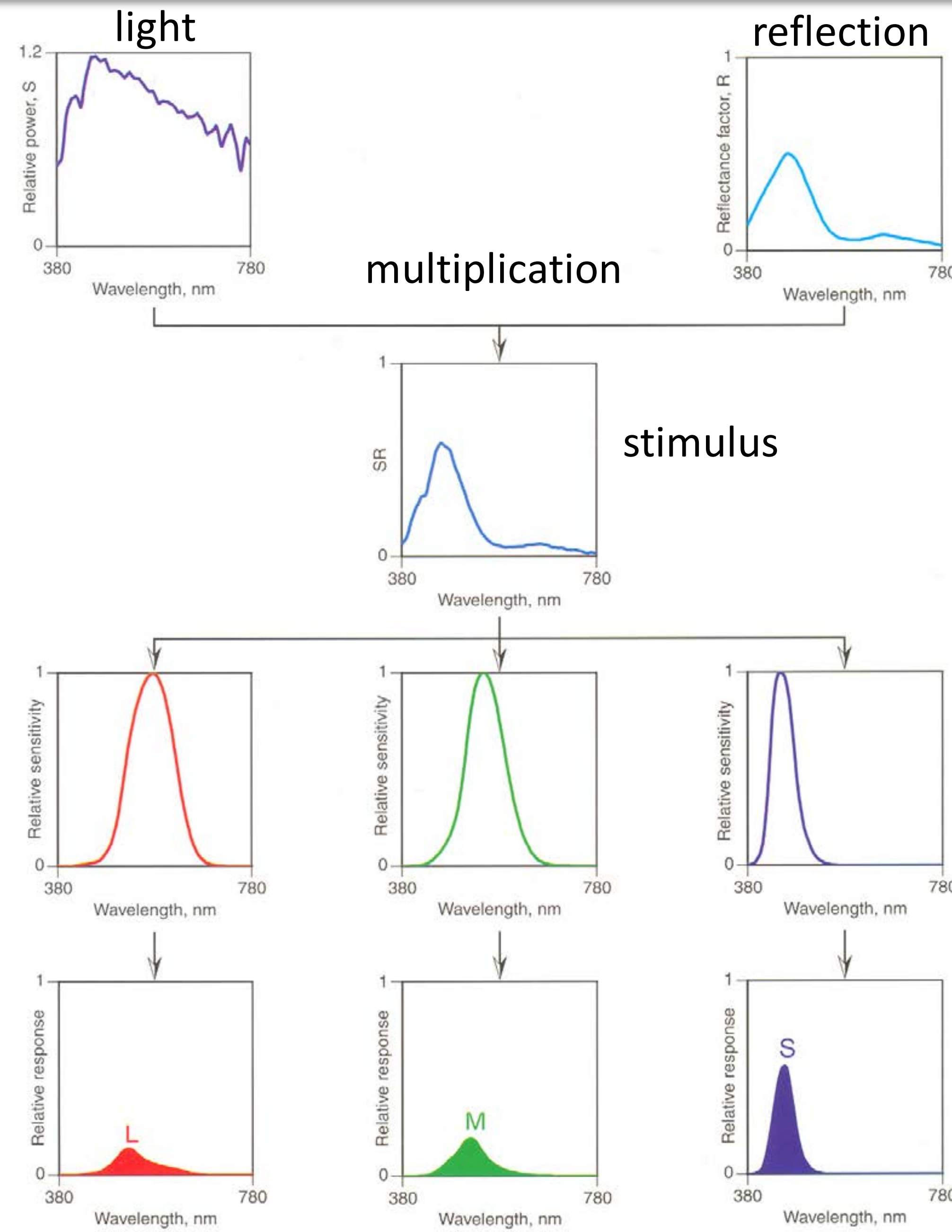






images: David Forsyth

Trichromatic Color Vision



sensitivity curves

multiplication of wavelengths

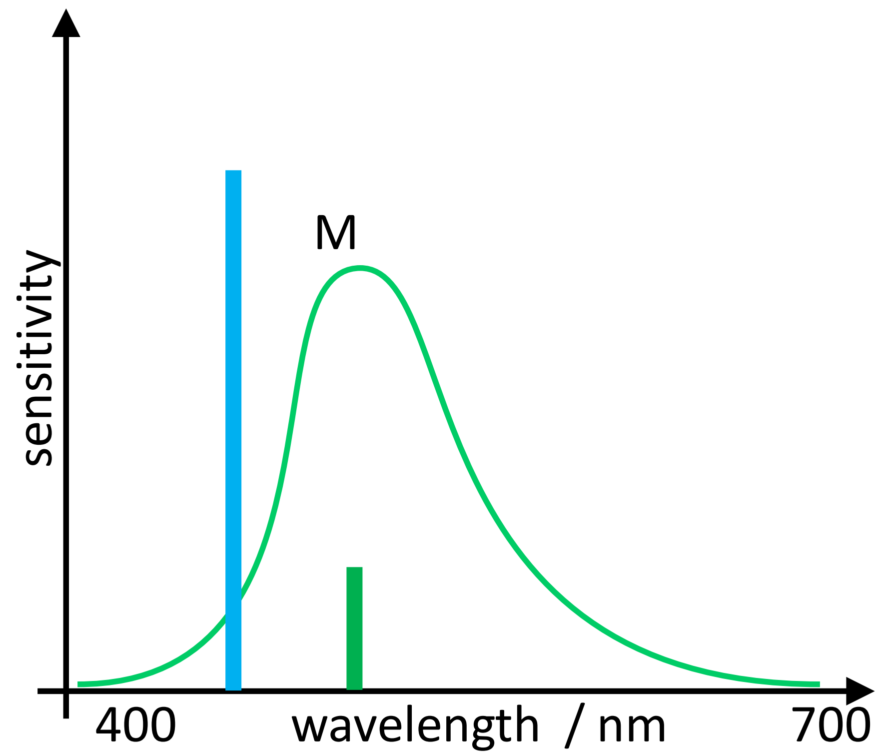
integration

Trichromatic Color Vision

- Perceptual response to light of different wavelength:

$$\mathrm{s} = \int S(\lambda)P(\lambda)d\lambda \qquad \mathrm{m} = \int M(\lambda)P(\lambda)d\lambda \qquad \mathrm{l} = \int L(\lambda)P(\lambda)d\lambda$$

- Different wavelength and intensity, but identical M-response

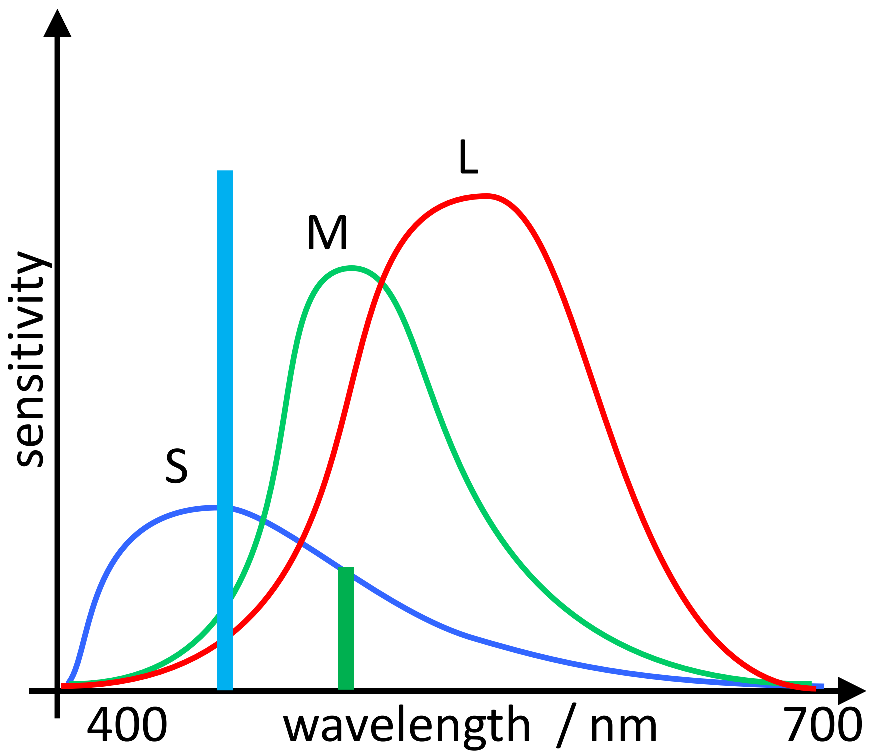


Trichromatic Color Vision

- Perceptual response to light of different wavelength:

$$\mathrm{s} = \int S(\lambda)P(\lambda)d\lambda \qquad \mathrm{m} = \int M(\lambda)P(\lambda)d\lambda \qquad \mathrm{l} = \int L(\lambda)P(\lambda)d\lambda$$

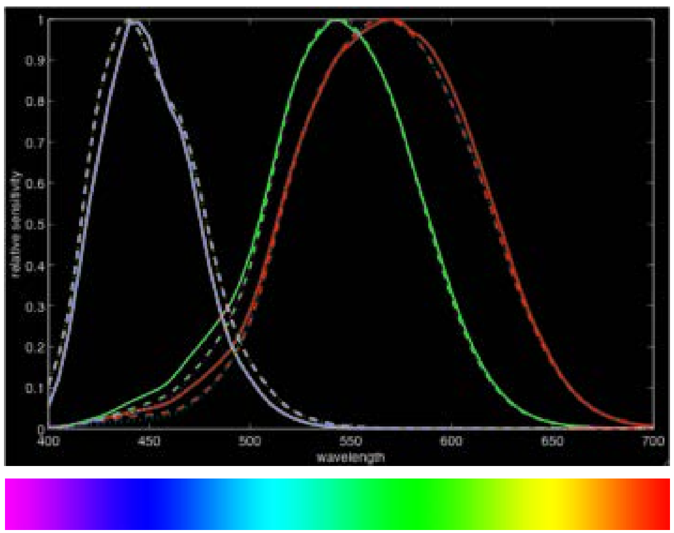
- Different wavelength and intensity, but different response w.r.t. different receptors!



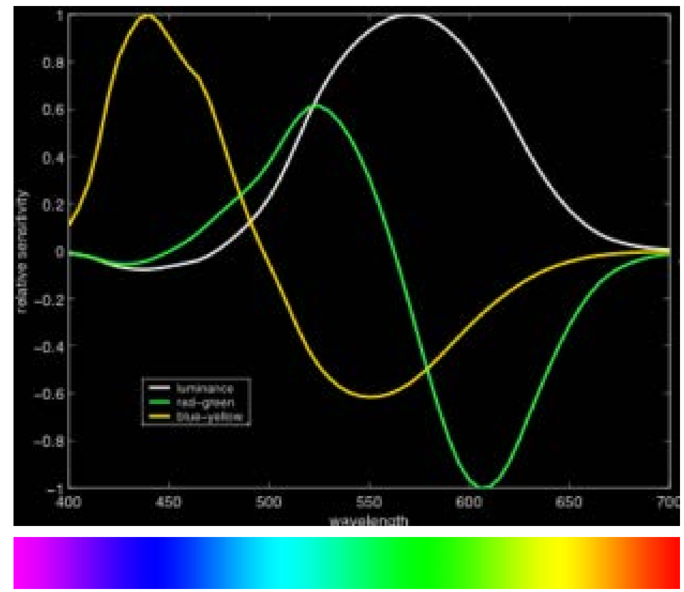
Trichromatic Color Vision

- Efficient encoding of L, M, S cone responses:
  - De-correlation of LMS cone responses in opponent basis functions

Stockman & Sharpe, 2002



PCA



G. Buchsbaum and A. Gottschalk, Trichromacy, Opponent Colours Coding and Optimum Colour Information Transmission in the Retina. Proc. of the Royal Society of London. Series B, Biological Sciences, 1983.

Metamerism

- Perceptual response to light of different wavelength:

$$\mathrm{s} = \int S(\lambda)P(\lambda)d\lambda \qquad \mathrm{m} = \int M(\lambda)P(\lambda)d\lambda \qquad \mathrm{l} = \int L(\lambda)P(\lambda)d\lambda$$

- Different light spectra $\mathrm{P}(\lambda)$ can cause identical response $(\mathrm{s}, \mathrm{m}, \mathrm{l})$!
- Metamerism: different spectra can look identically

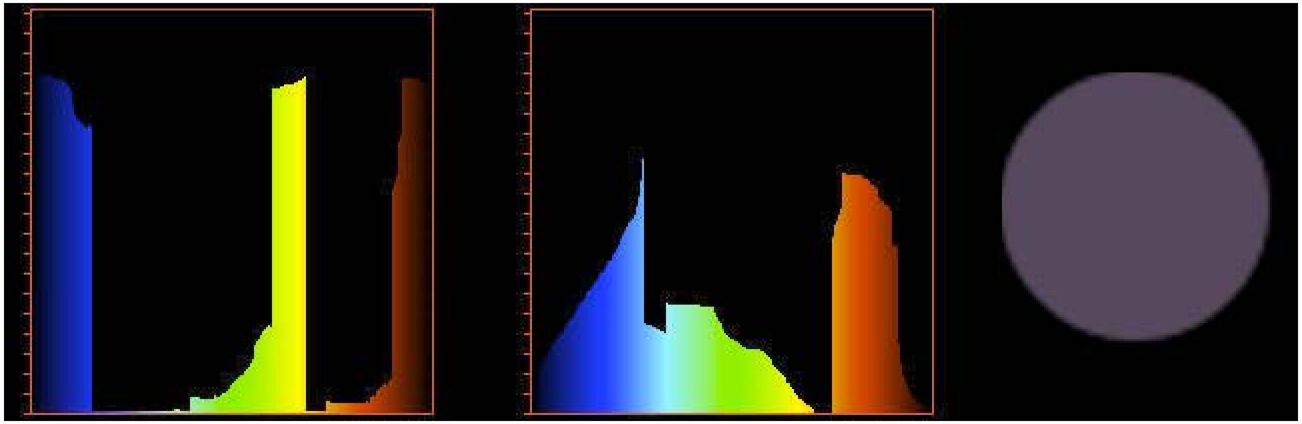
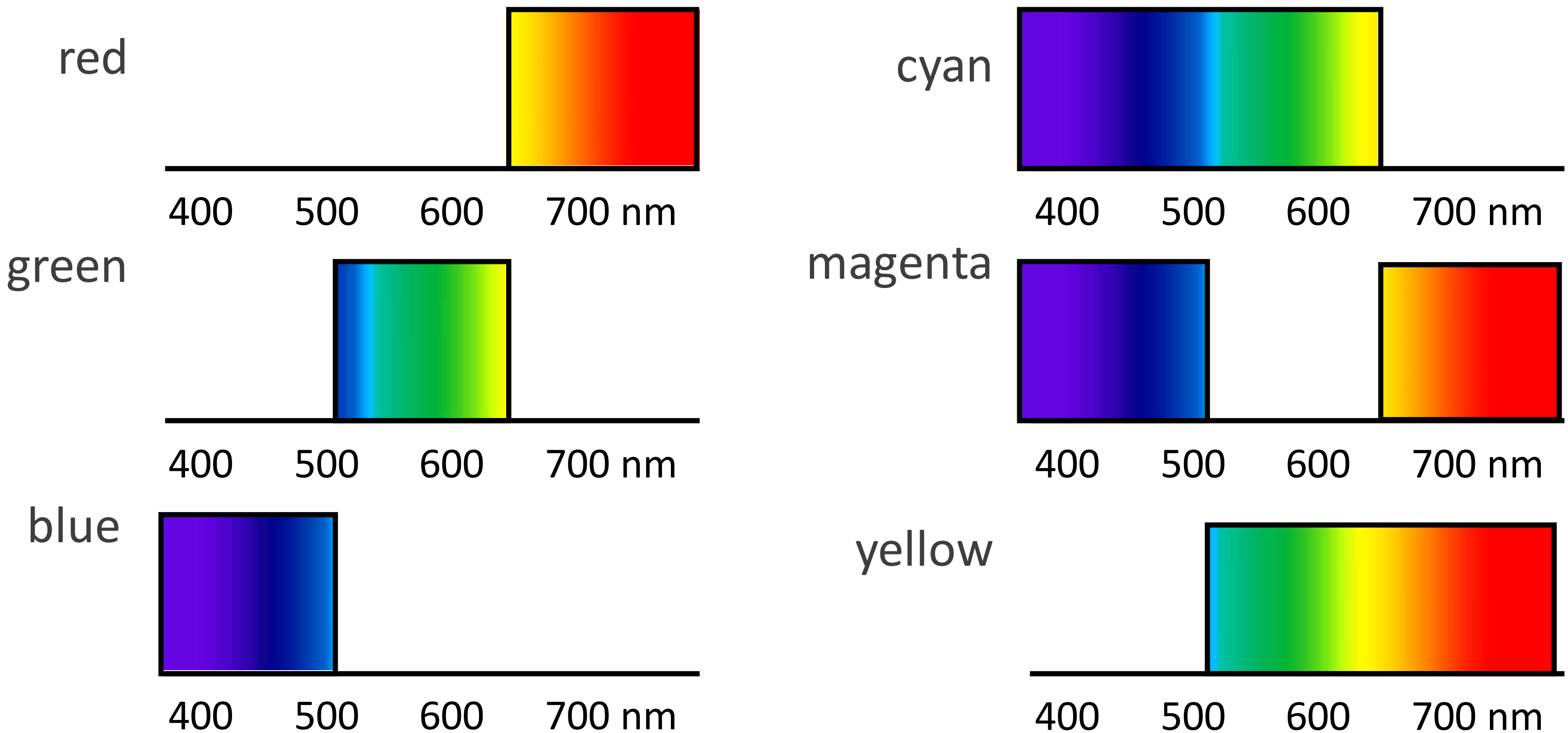


image: Hughes, Bell, Doppelt

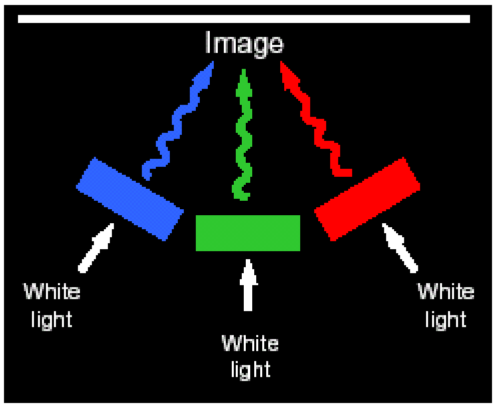
Additive and Subtractive Color Mixing

- Basic colors of
  - additive color mixing: red, green, and blue
  - subtractive color mixing: cyan, magenta, and yellow

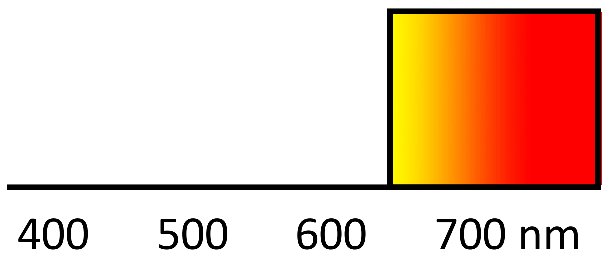


Additive Color Mixing

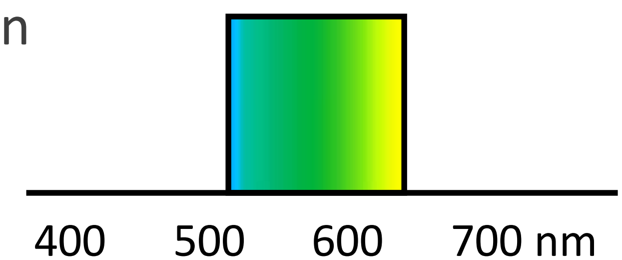
- Combination of colors by **addition of the spectra**
  - E.g. CRT monitors, multiple projectors on one screen



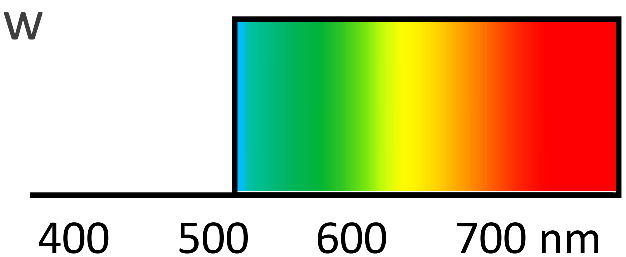
red

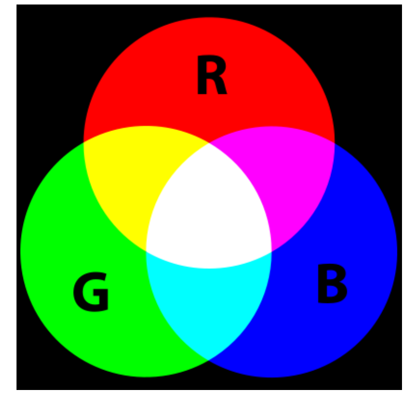


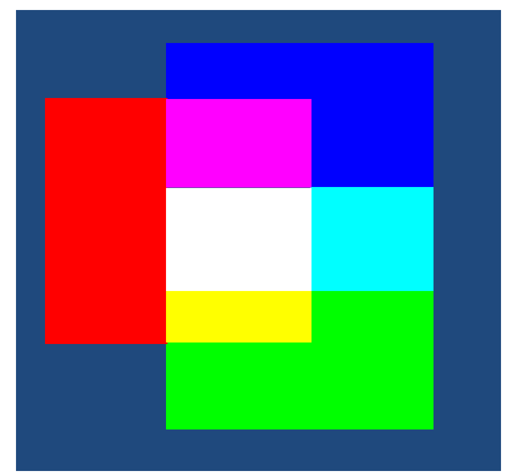
green



yellow



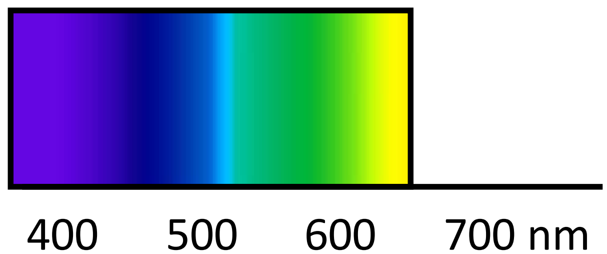


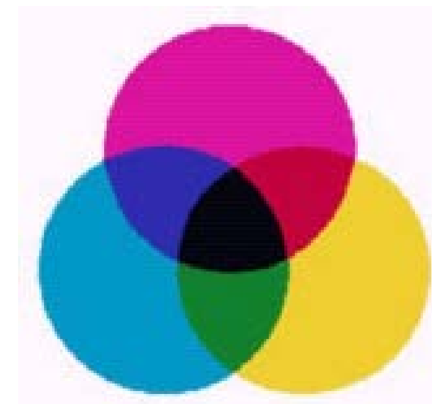


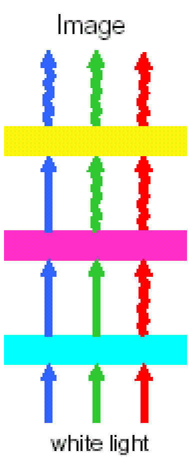
Subtractive Color Mixing

- Combination of colors by **multiplication of the spectra**
  - E.g. photographic film, color pens, color printer

cyan





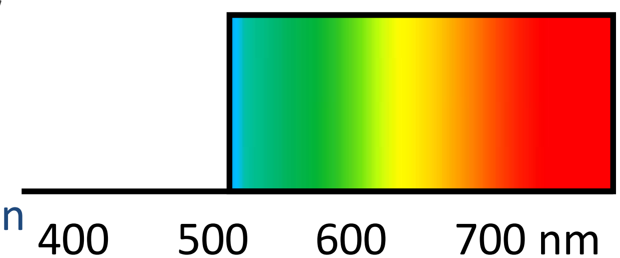


yellow absorbs blue

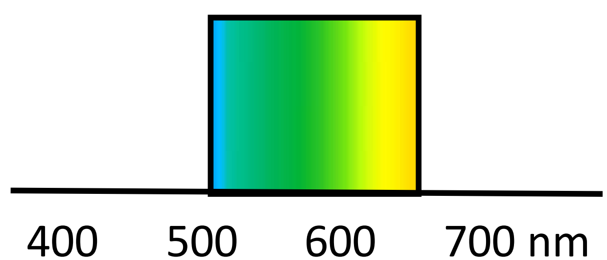
magenta absorbs green

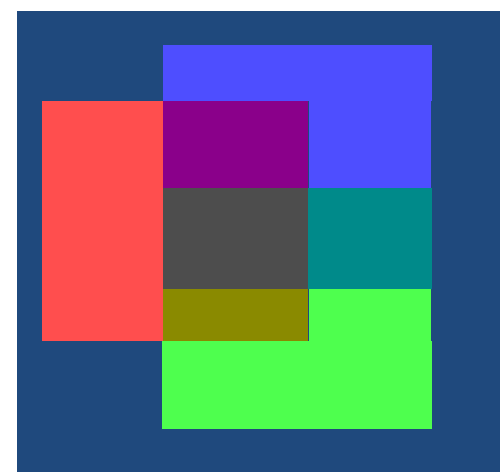
cyan absorbs red

yellow



green



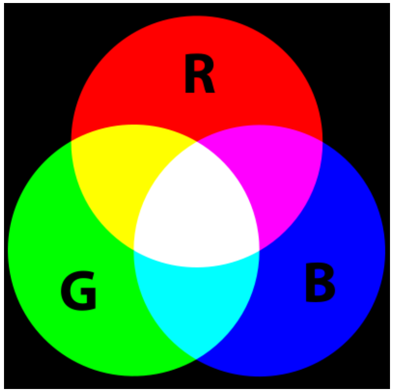


Grassmann's Law

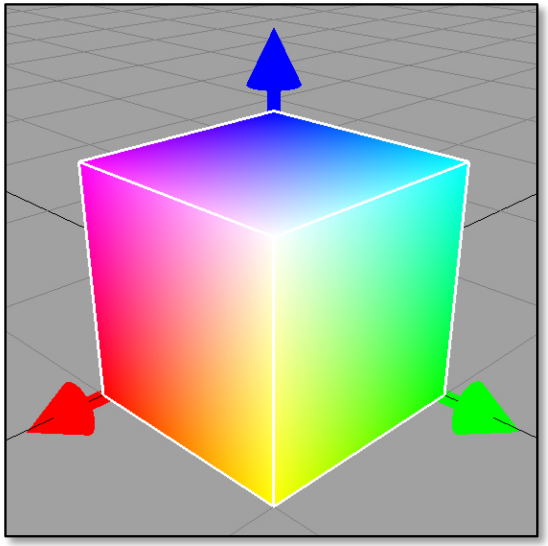
After Hermann Günther Graßmann (1809–1877):

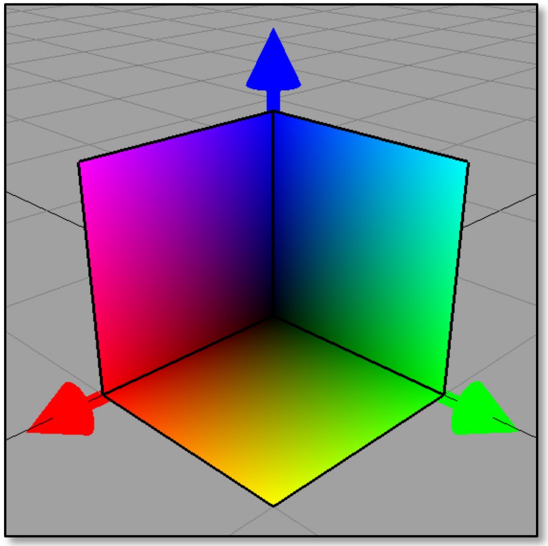
- Every color impression can be described by means of 3 basic quantities
  - Color is a 3-dimensional quantity
  - E.g., color hue, saturation, lightness, or red, green, blue, or …
- The color hue of an additively mixed color depends only on the color impression of the original colors
  - I.e., conclusions about spectral composition not possible
- Intensity of an additively mixed color corresponds to sum of intensities of original colors

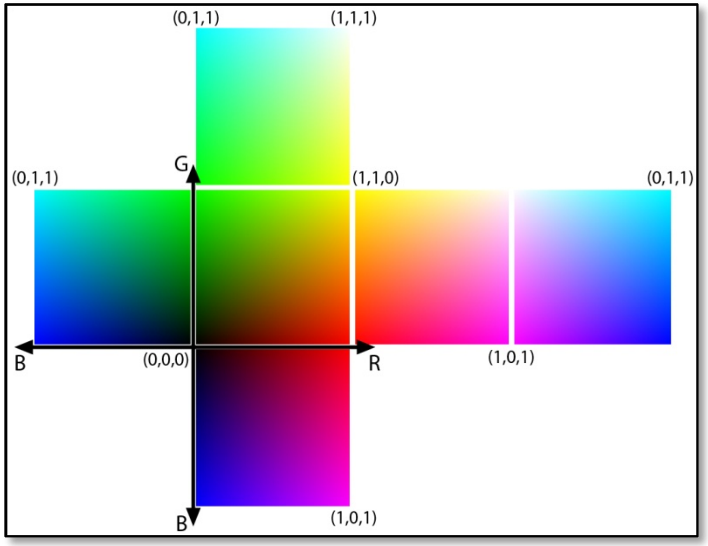
RGB Color Space



- Hardware-oriented color representation
  - 3 primary colors (red, green, blue) represent basis
  - Results in 3-dimensional color space:
    $\mathbf{C} = \mathrm{r}\mathbf{R} + \mathrm{g}\mathbf{G} + \mathrm{b}\mathbf{B}$ with $(\mathrm{r}, \mathrm{g}, \mathrm{b}) \in [0,1]^3$
  - The coefficients r, g, b are called **tristimulus values**
  - The definition of primary colors depends on respective RGB space
  - Additive color mixing suits well for monitors, projectors, …
  - Luminance approximation: $\mathrm{Y} = 0.3\mathrm{r} + 0.59\mathrm{g} + 0.11\mathrm{b}$

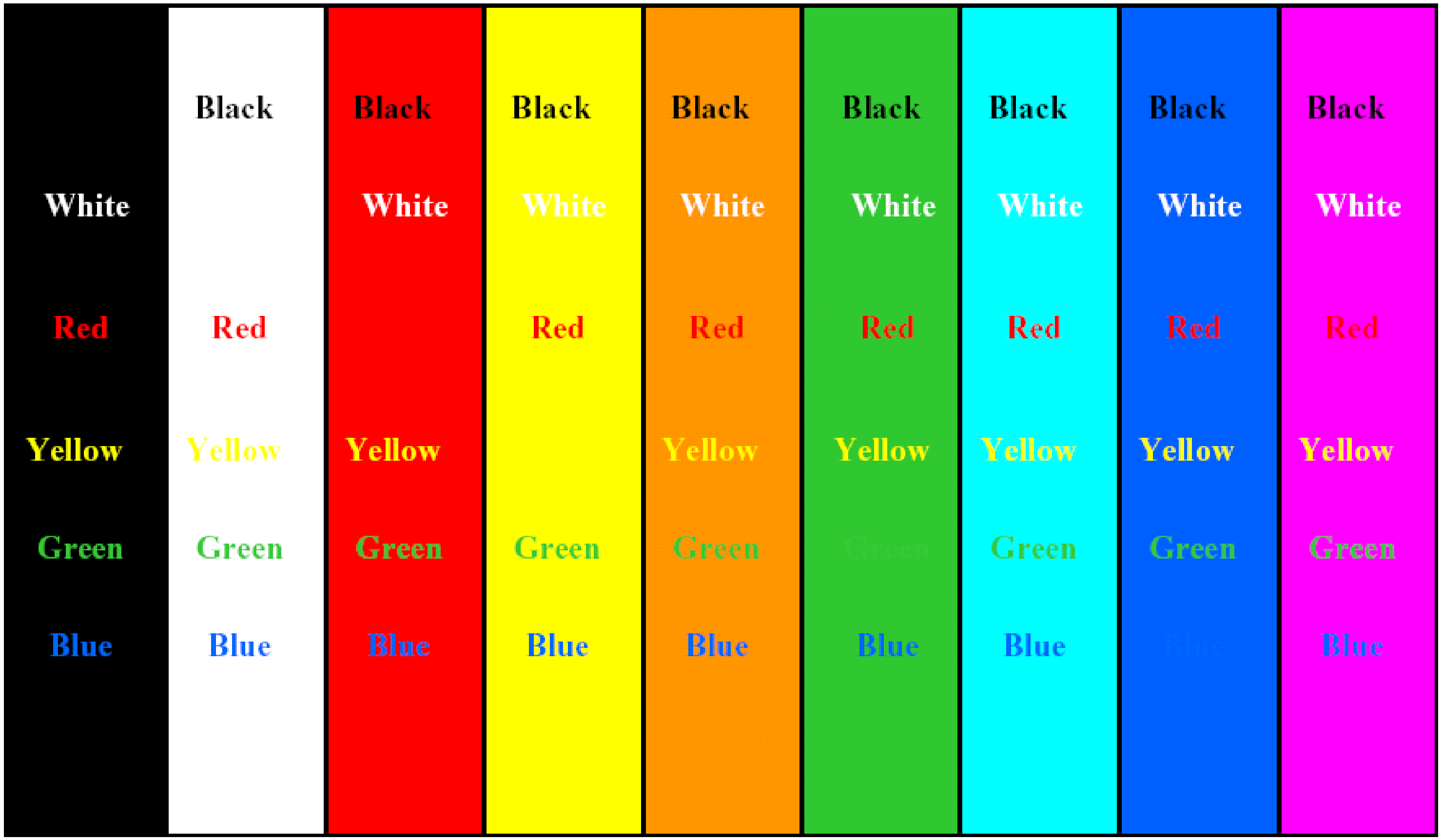






Contrast

Foreground–background



Rule of thumb:
Difference of luminance foreground/background > 0.2

CMY(K) Color Space

- Subtractive color space (cyan, magenta, yellow)
  - Dual space to RGB

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

  - Each primary color absorbs part of the spectrum
  - Ink, dyes, pigments

- CMYK
  - Printers typically use an additional 4. “key” color
  - Key (or blacK) is pure black
  - Practical reasons
    - No covering black
    - Saving of ink

$K = \min(C, M, Y)$ and

$C' = C - K$

$Y' = Y - K$

$M' = M - K$

Color Models in Television

- **YIQ Model** (old US television NTSC)
  - Y = luminance
  - I, Q = color differences

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0{,}3 & 0{,}59 & 0{,}11 \\ 0{,}6 & -0{,}28 & -0{,}32 \\ 0{,}21 & -0{,}52 & 0{,}3 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

- **YUV Model** (in European PAL system and newer NTSC)

$$U = 0.493 \cdot (B - Y)$$
$$V = 0{,}877 \cdot (R - Y)$$

$$\begin{pmatrix} I \\ Q \end{pmatrix} = \begin{pmatrix} -\sin 33° & \cos 33° \\ \cos 33° & \sin 33° \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix}$$

chrominance components

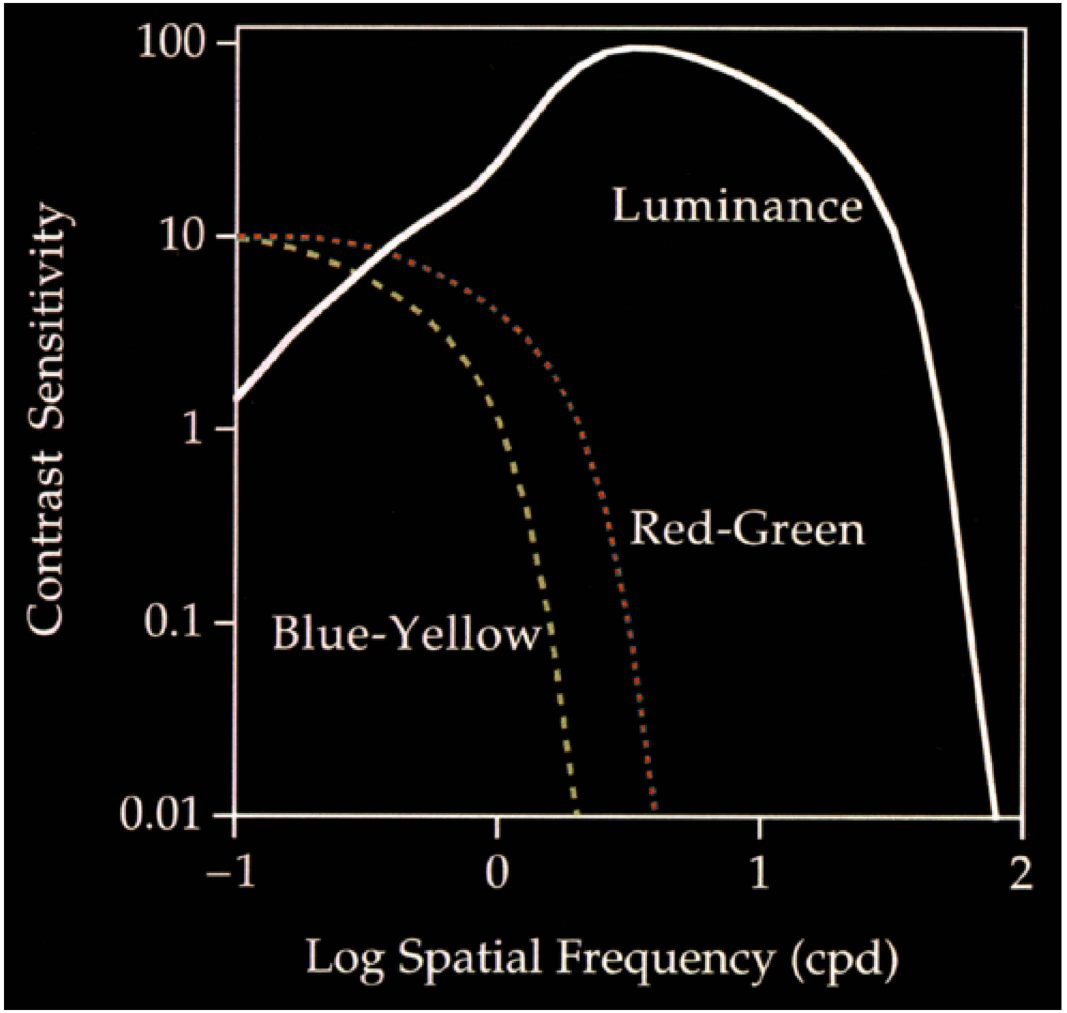
- **YCbCr Model**
  - ***Y*** = luma
  - ***C_b*** = blue–yellow
  - ***C_r*** = red–green

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

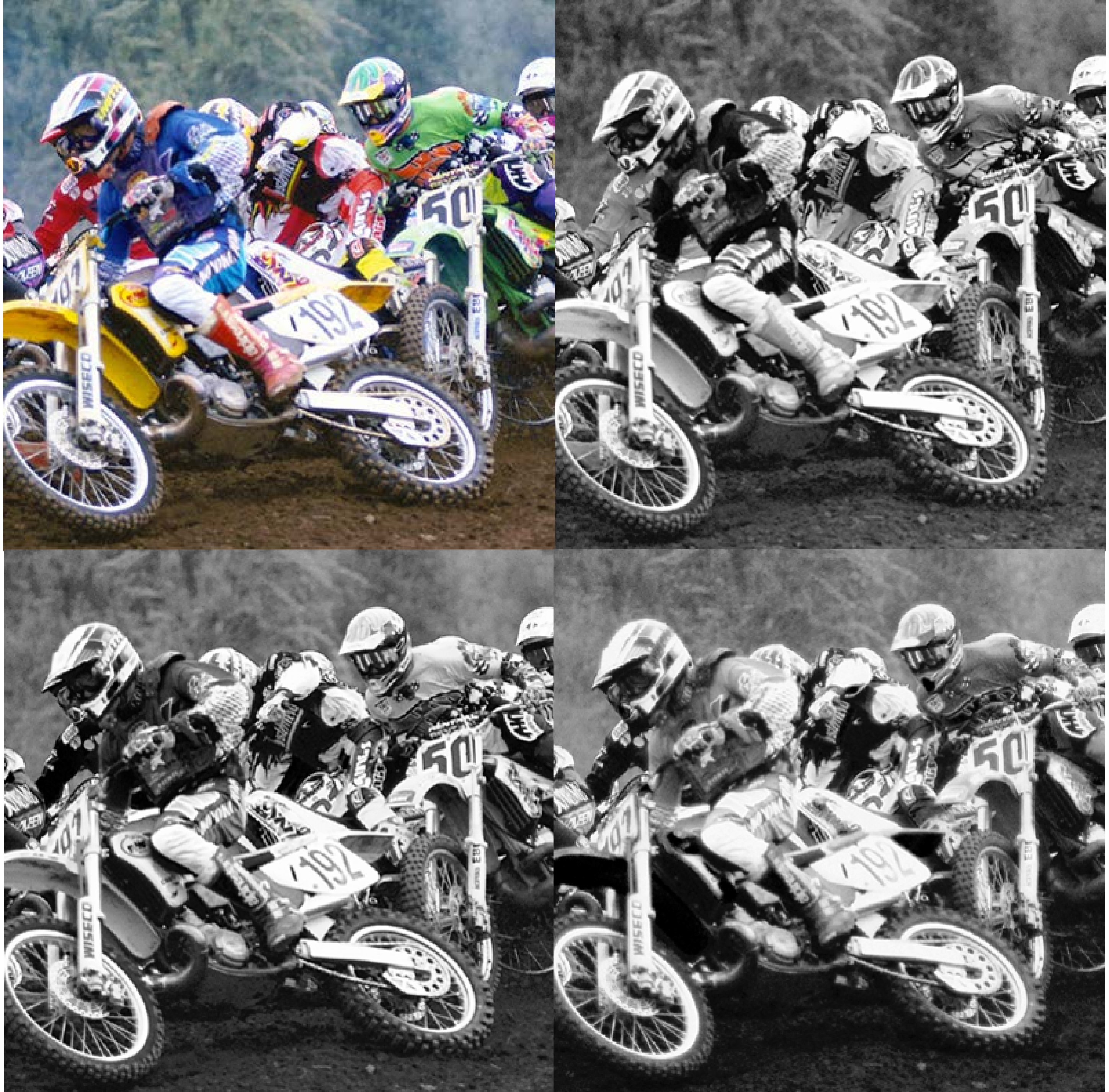
  - R’, G’, B’ : *nonlinear* encoded sRGB values (sRGB: standard RGB space)

Contrast Sensitivity Luminance–Chrominance

- Color differences contribute less to contrast than luminance!
- Application: compression
- Examples
  - RBG vs. YCrCb channels
  - Compression via 64 × subsampling chrominance first, then RGB



Images: Sabine Süsstrunk EPFL



red

green

blue

Y

Cr

Cb

Contrast Sensitivity Luminance–Chrominance

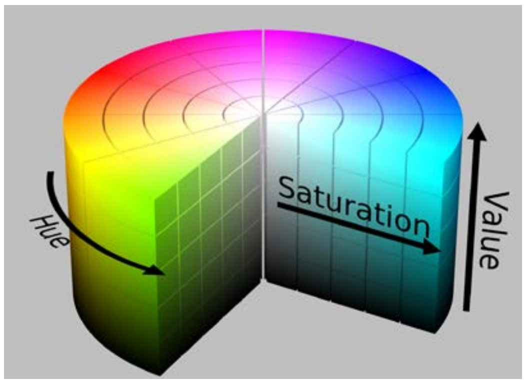
chrominance subsampled by 64

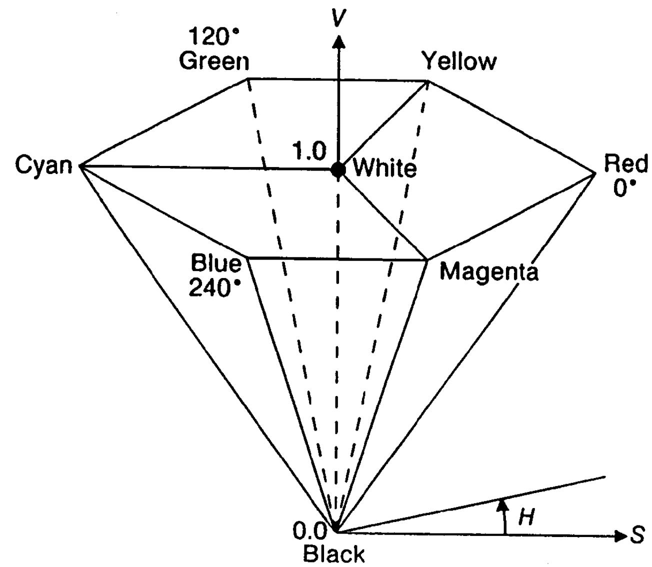
Contrast Sensitivity Luminance–Chrominance

RGB subsampled by 64

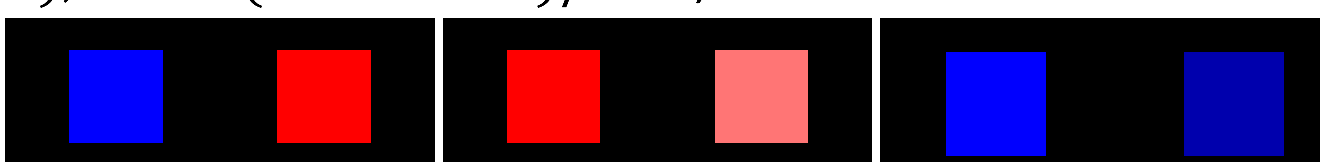
HSV Color Space

- Perceptually based color system
  tone (hue), saturation, lightness (value)
  - Representation as a cylinder or (hexagonal) cone





  - Neither additive nor subtractive
  - But: intuitive and thus often used in user interfaces
  - Conversion RGB ⟷ HSV
    $V = \max(r, g, b),\ S = (\max - \min)/\max,$
    $\mathrm{H} \in [0°, 360°]$
    (case analysis)



Hue Saturation Intensity

Color Model, Color Space, and Stimuli

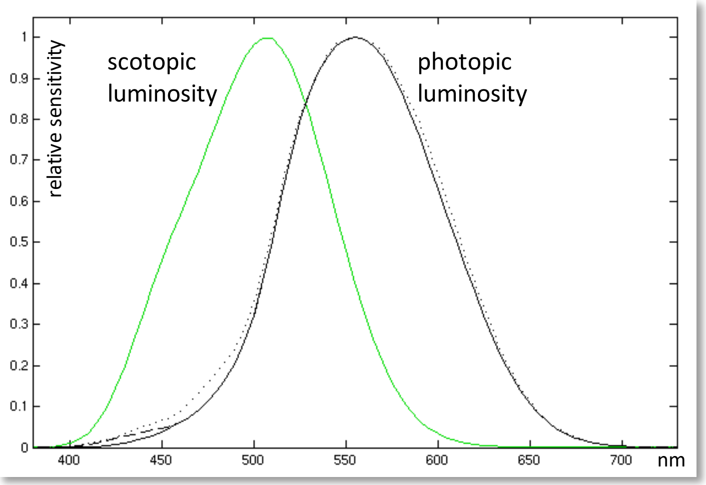
- A **color model** is a mathematical model to describe colors with value tuples
  - Typically 3-tuples (e.g., RGB), sometimes 4 (CMYK)

- A **color space** is the set of all colors that can be described with a *given model*

- The **tristimulus values** describe a color in a given color space (i.e., the values are meaningless if the color model and color space are not provided)

Photometric Quantities

- Photometry: quantities equivalent to radiometry
  - Weighted with brightness sensitivity curve (luminous efficiency function)
- CIE[1] photopic luminous efficiency function
  - E.g., radiant flux to luminous flux
    $\bar{y}(\lambda) = S(\lambda) + M(\lambda) + L(\lambda)$



en.wikipedia.org/wiki/Luminosity_function

- Luminous flux in lumen [lm]
  - Analog to radiant flux

$$F = 683.02 \frac{lm}{W} \int \bar{y}(\lambda) P(\lambda) d\lambda$$

[1] Commission Internationale de l'Eclairage

Photometric Quantities

- *Luminous energy*

  Q_v in lumen second (Talbot) $[\mathrm{lm{\cdot}s}]$

- *Luminous flux*

  $\Phi_v = \frac{dQ_v}{dt}$ in lumen $[\mathrm{lm{=}Talbot/s(=cd{\cdot}sr)}]$

- Illumination intensity: *illuminance*

  $E_v = \frac{d\Phi_v}{dA}$ in lux $[\mathrm{lux{=}lm/m^2}]$

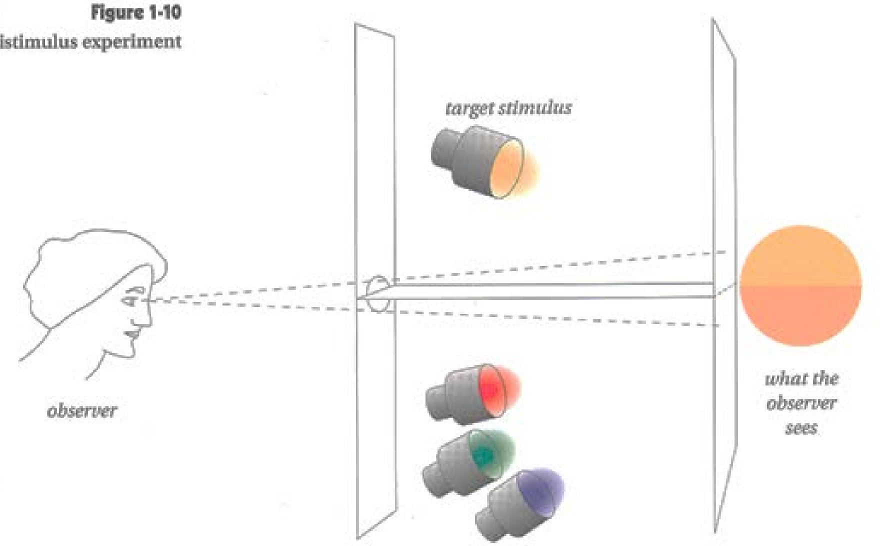
- Light density: *luminance*

  $L_v = \frac{d^2\Phi_v}{\cos\theta dAd\omega}$ luminous flux per solid angle and per projected unit area in $[\mathrm{lm/(sr{\cdot}m^2)}]$

CIE Color Matching Functions

- Perceptual experiments
  - W. David Wright (1928) and John Guild (1931)
  - Two colors projected to screen
    - Reference color given
    - Comparison color by mixing of three primary colors
    - Participant adjusts intensities of primaries until perception match

Figure 1-10
Tristimulus experiment



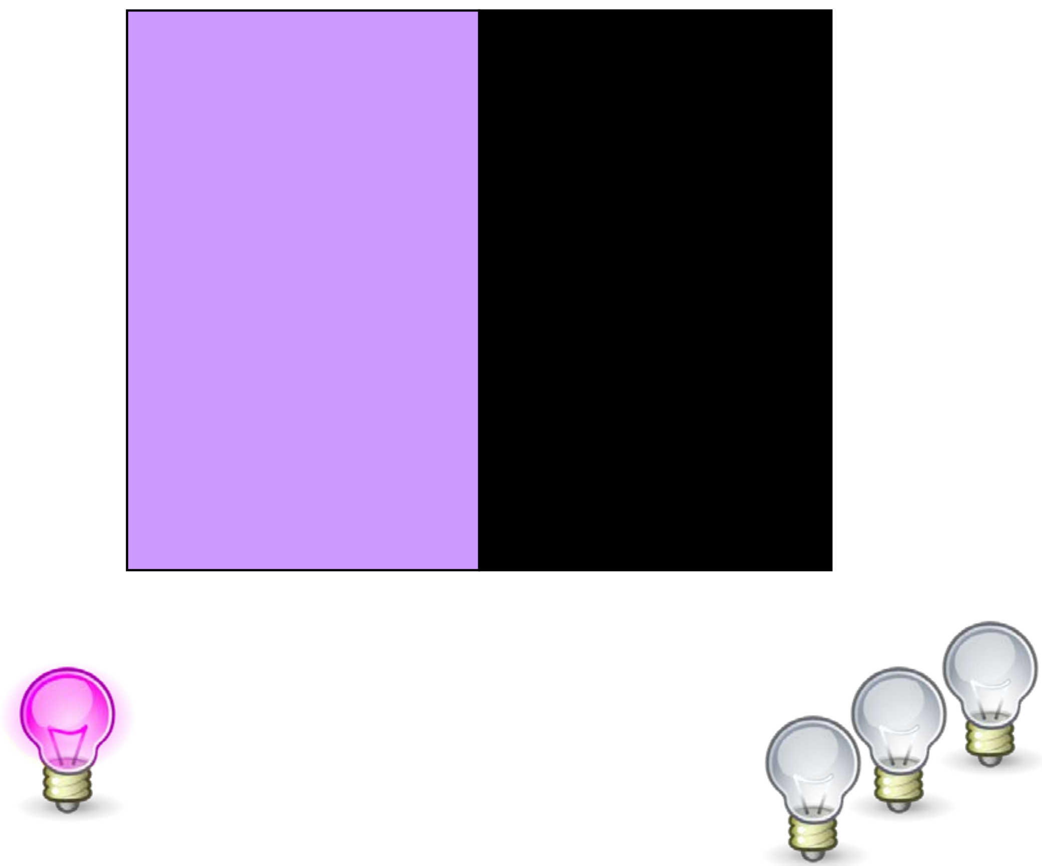
The observer adjusts the intensities of the red, green, and blue lamps until they match the target stimulus on the split screen.

CIE Color Matching Functions

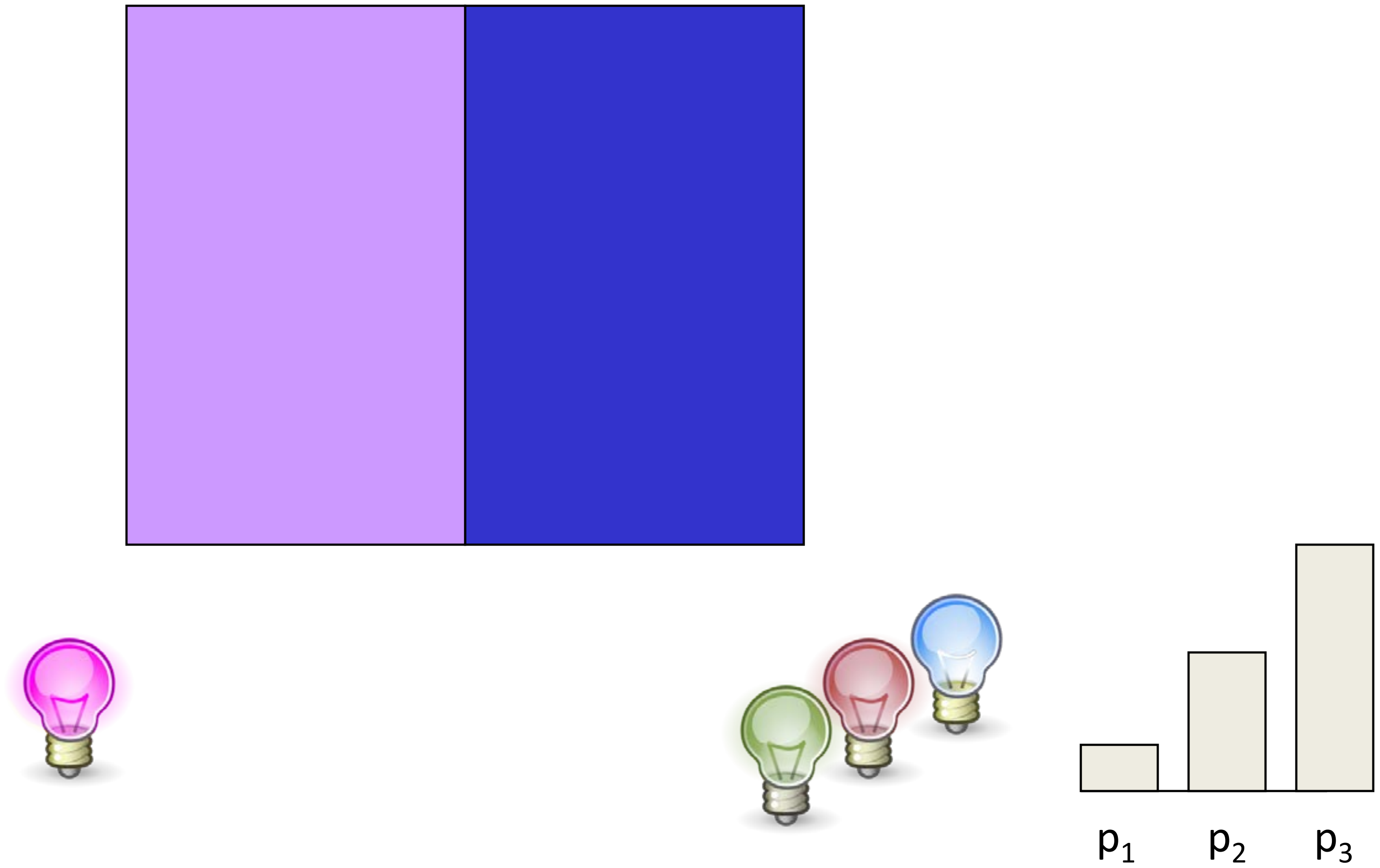
- Two colors projected on one screen
  - Reference color given
  - Comparison color by mixing of three primary colors
  - Participant adjusts intensities of primaries until perception match
- Not all colors could be reproduced!
  - In such cases, one of the primary colors was “added” to the test color
  - Color adjustment with the remaining primary colors
  - Primary color regarded as *negative* comparative value

Color Matching – Experiment 1

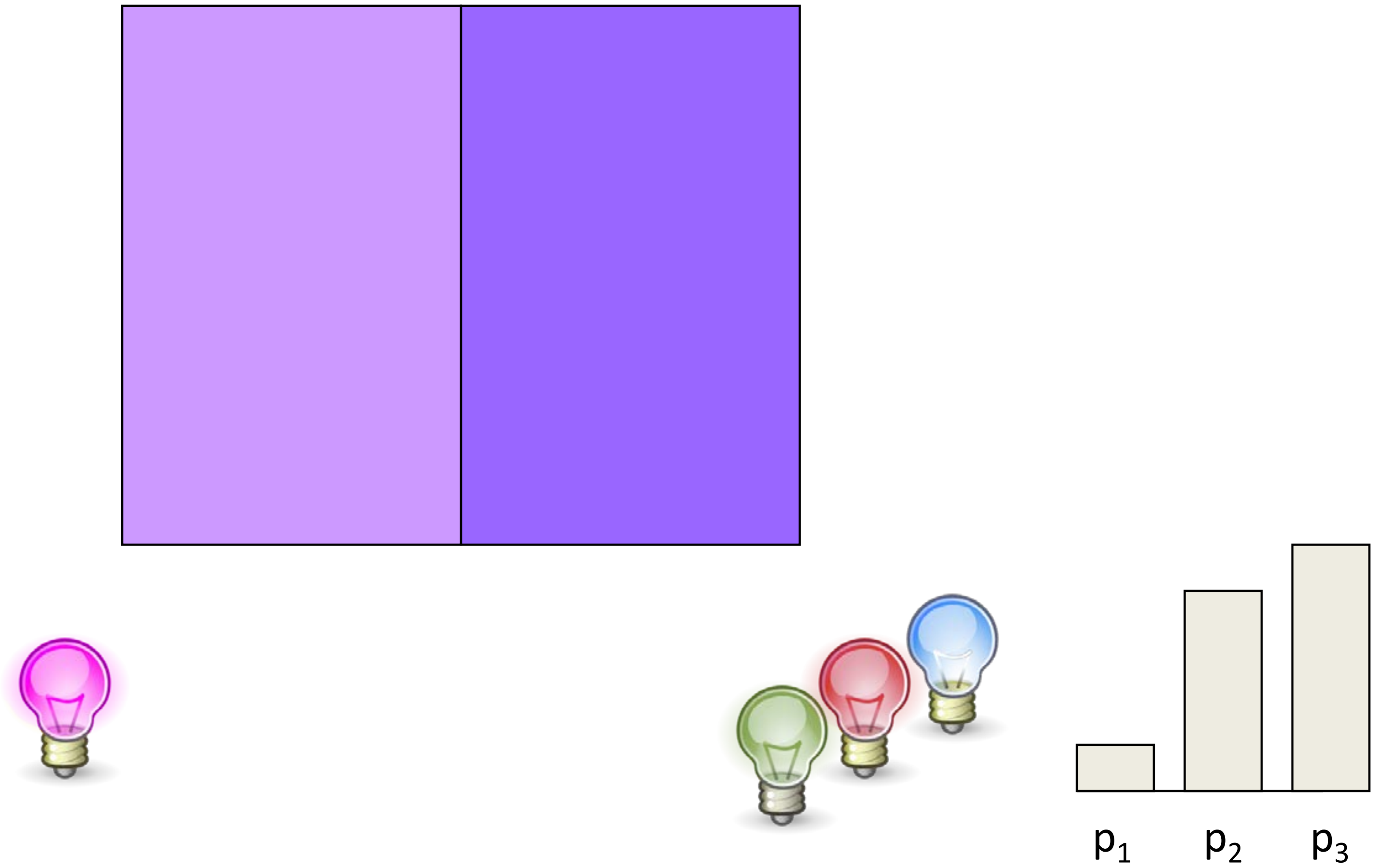
- Primary colors 435.8 nm (blue), 546.1 nm (green), 700 nm (red)



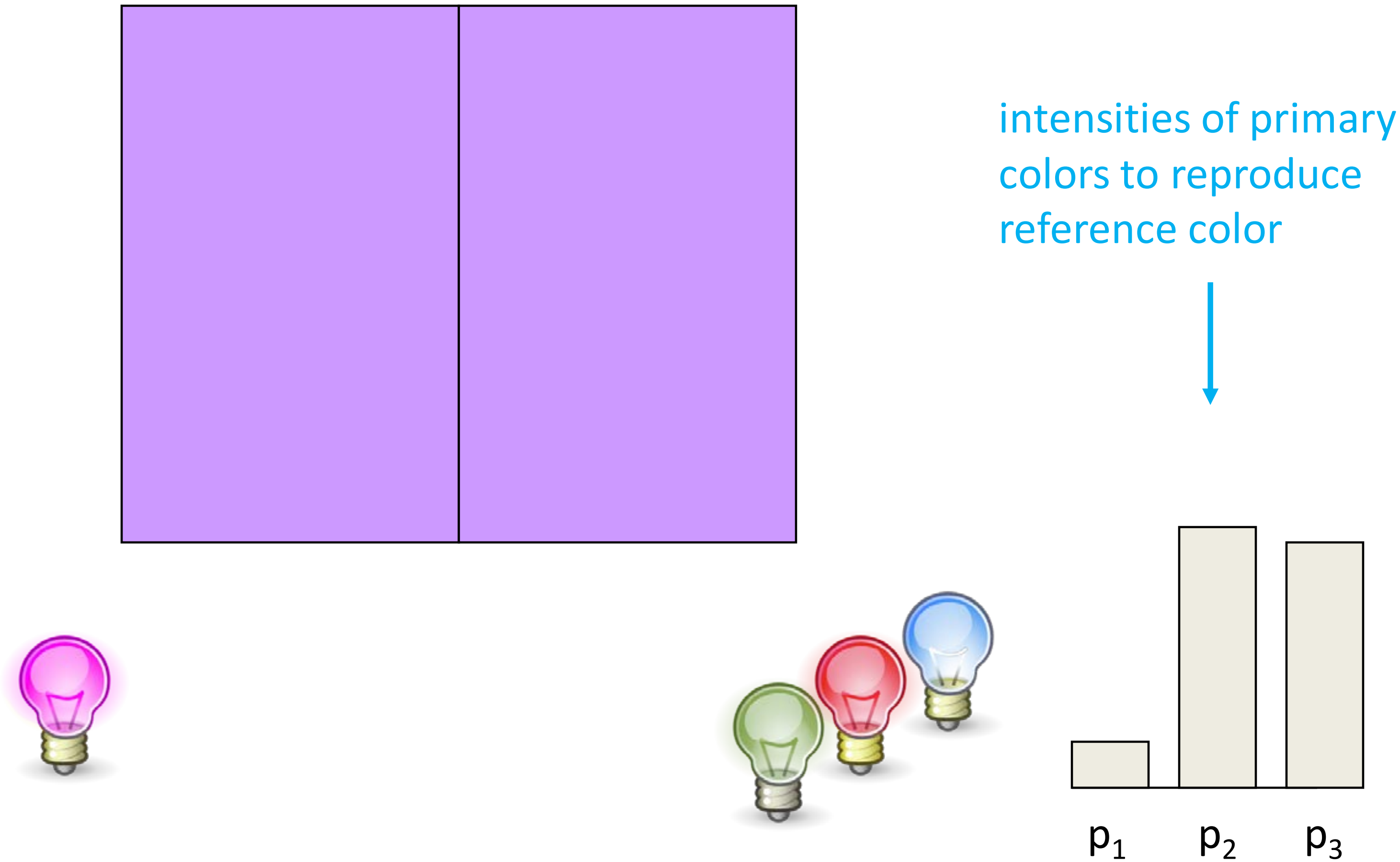
Color Matching – Experiment 1



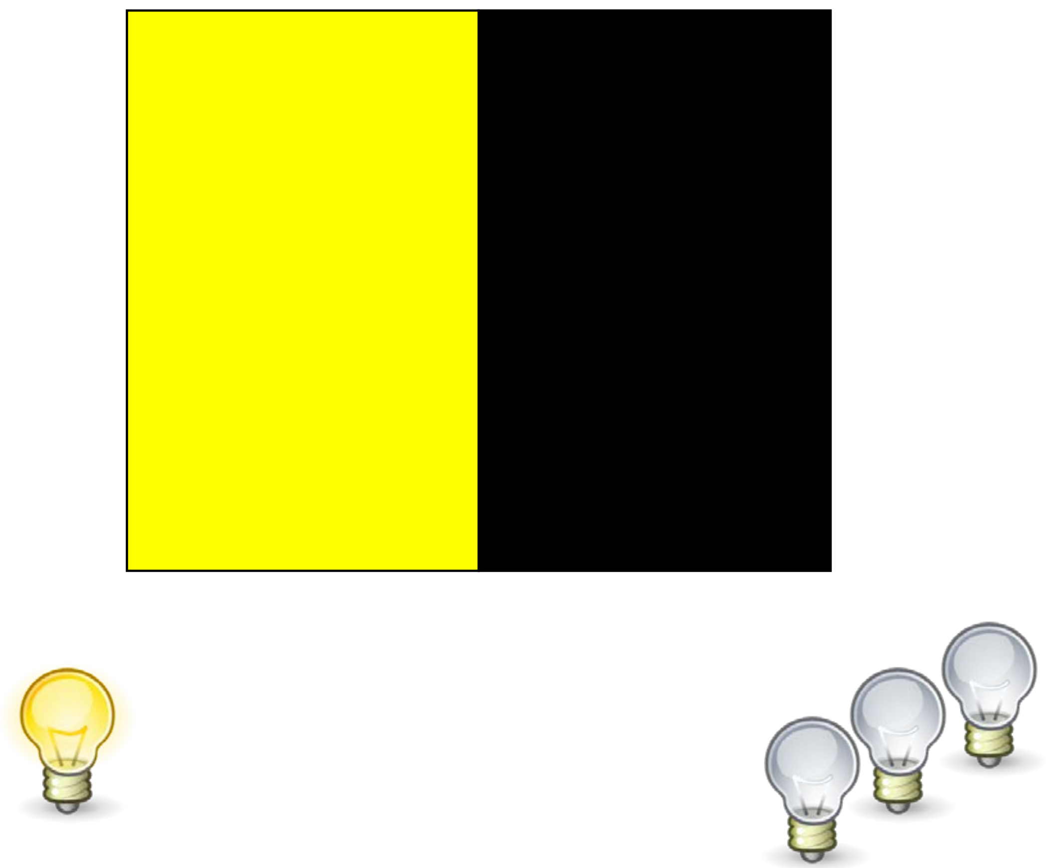
Color Matching – Experiment 1



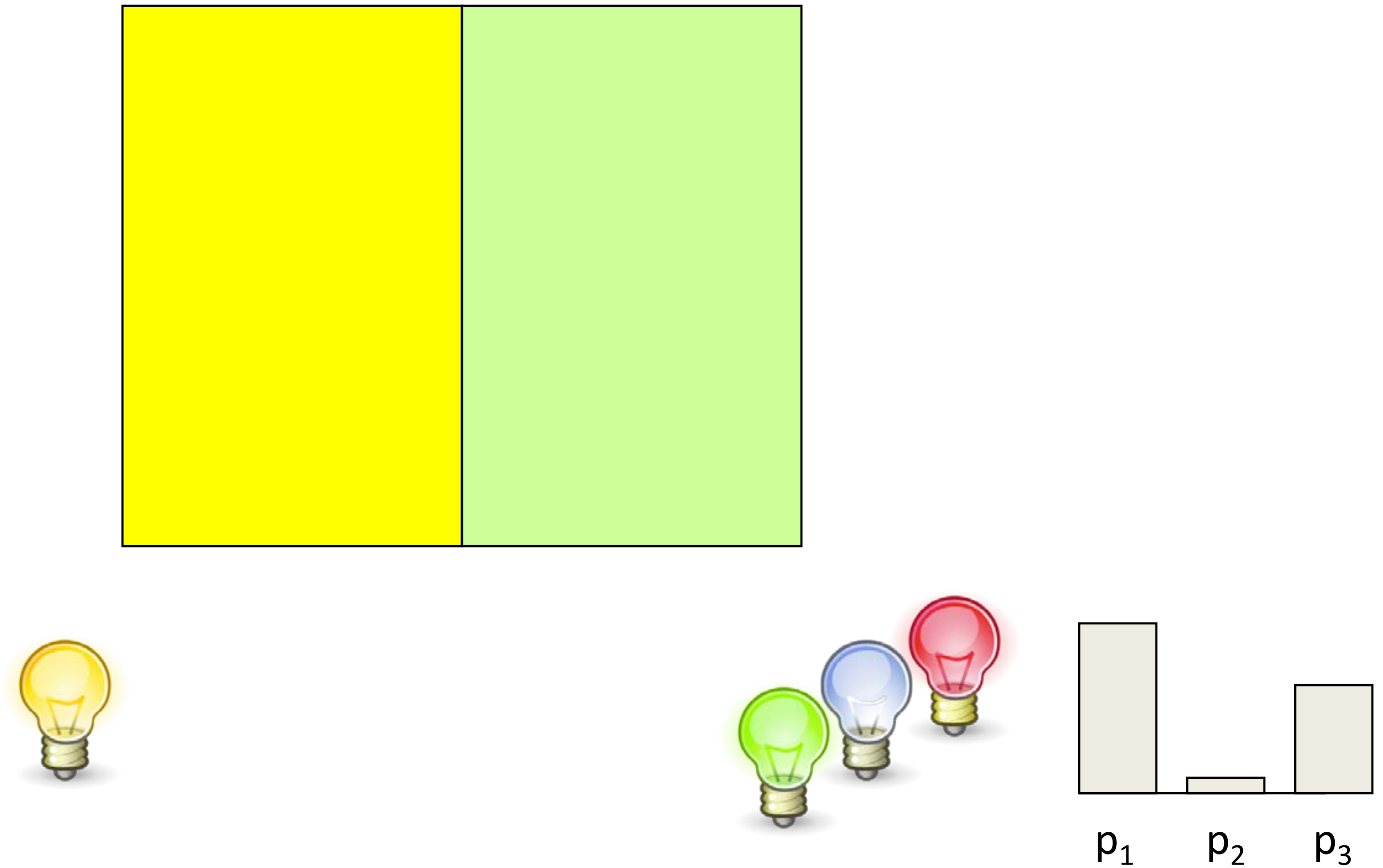
Color Matching – Experiment 1



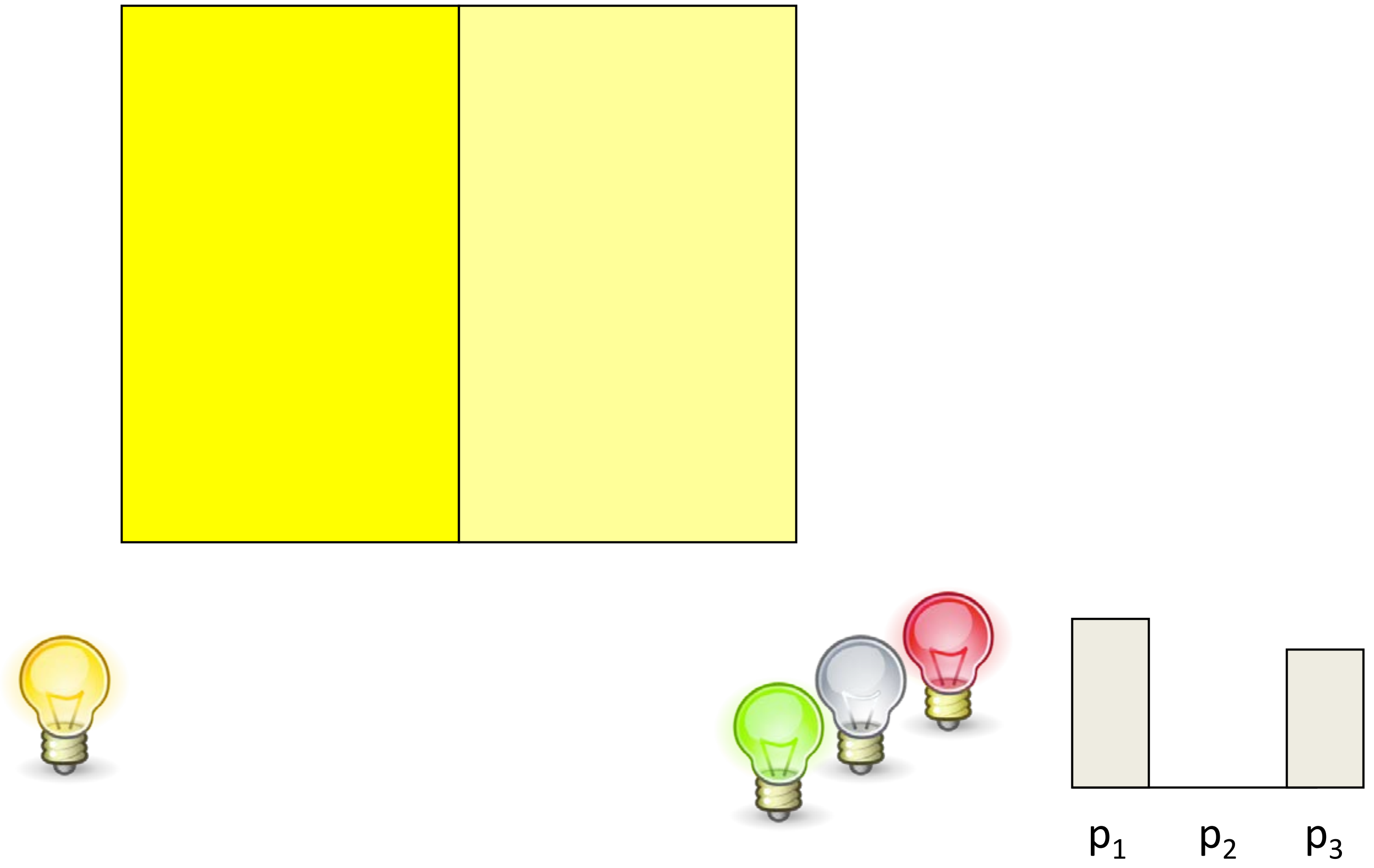
Color Matching – Experiment 2



Color Matching – Experiment 2

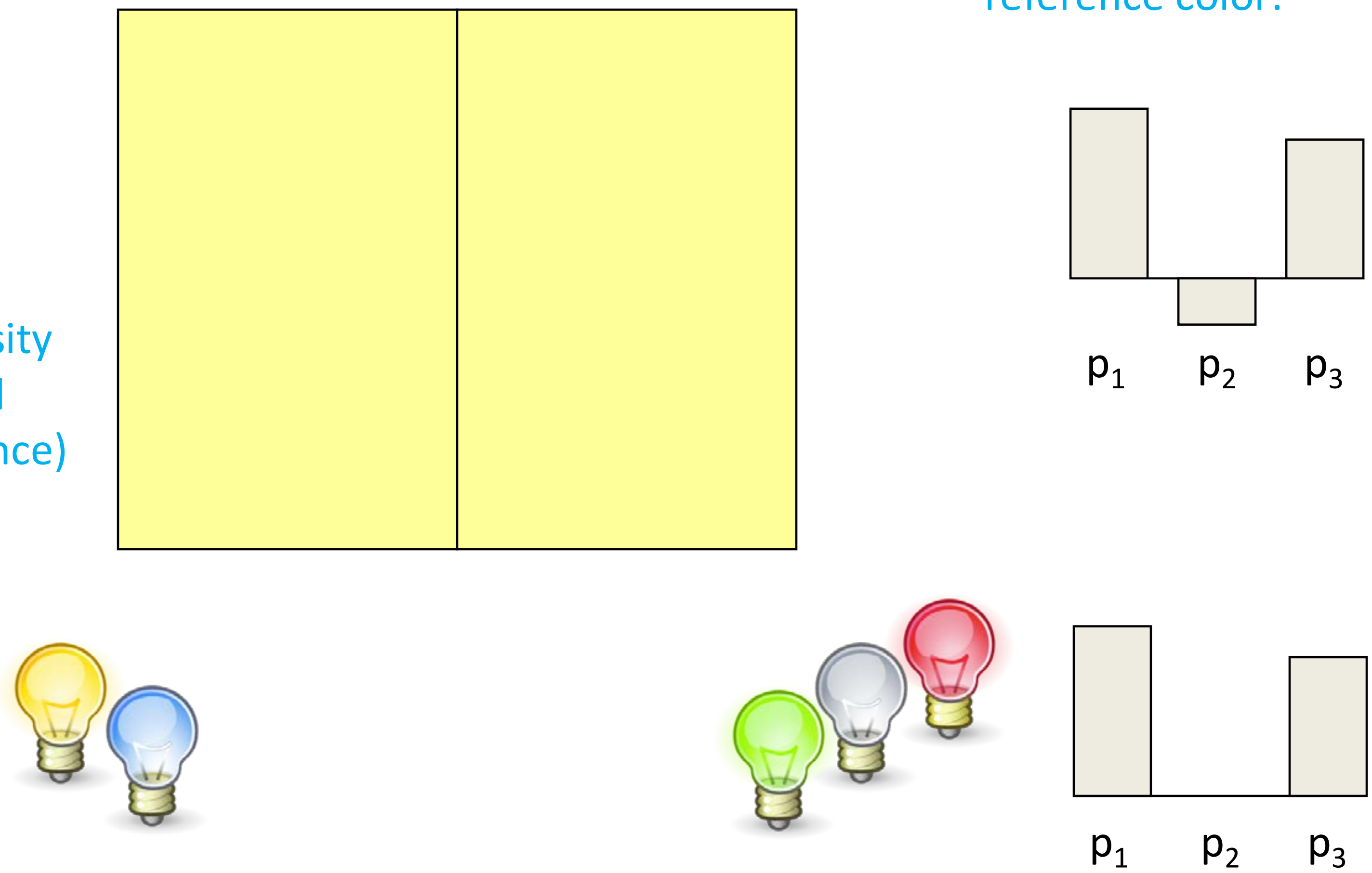


Color Matching – Experiment 2

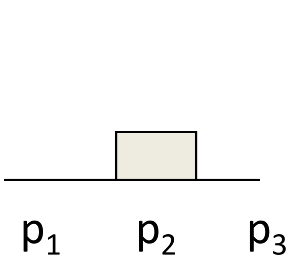


Color Matching – Experiment 2

intensities of primary colors to reproduce reference color:



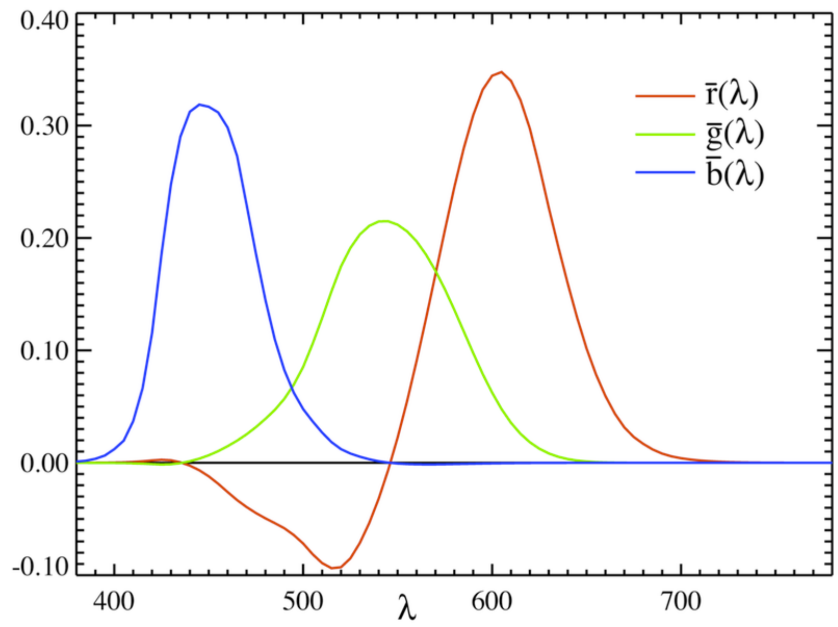
“negative” intensity of p_2 was needed (added to reference)



CIE Color Matching Functions

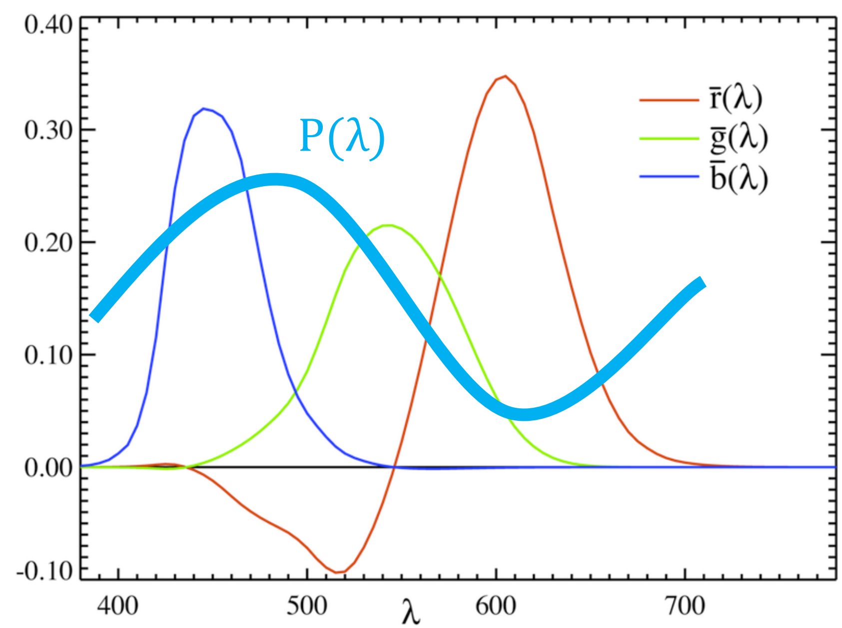
- By reproduction of the spectral colors with the RGB primary colors, one obtains the so-called **color matching functions**
- “How do I need to adjust the lamps to reproduce a color?“
- Tristimulus values for general spectra:

$$r = \int \bar{r}(\lambda)P(\lambda)d\lambda \qquad g = \int \bar{g}(\lambda)P(\lambda)d\lambda \qquad b = \int \bar{b}(\lambda)P(\lambda)d\lambda$$



Color Matching – Meaning

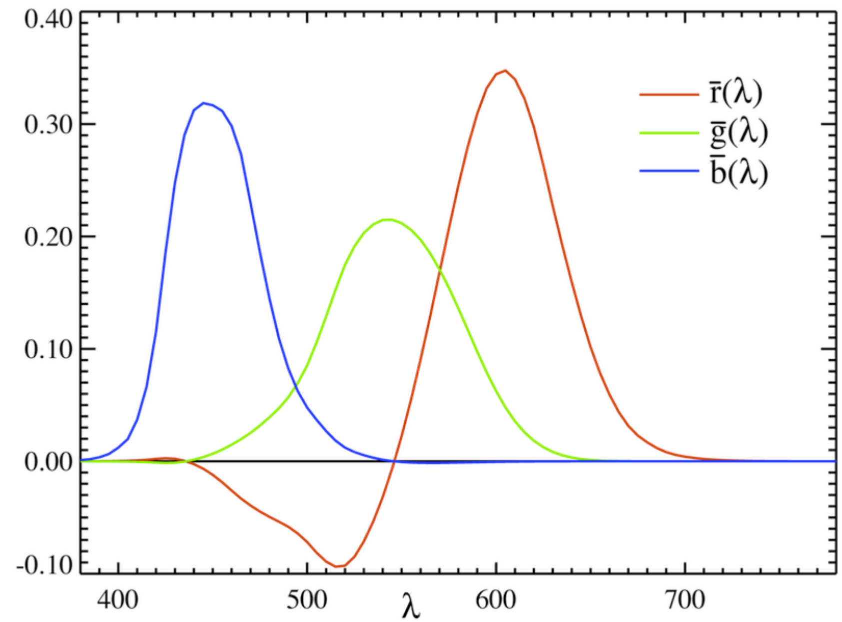
- Given a spectrum $P(\lambda)$
- Compute response to 3 sensitivity curves (tristimulus values)
- Choose intensity of primary colors (435.8 nm, 546.1 nm, 700 nm) according to the 3 responses
- One obtains a metameric color
- But: responses can be negative



CIE Color Matching Functions

- By reproduction of the spectral colors with the RGB primary colors, one obtains the so-called **color matching functions**
- Tristimulus values:

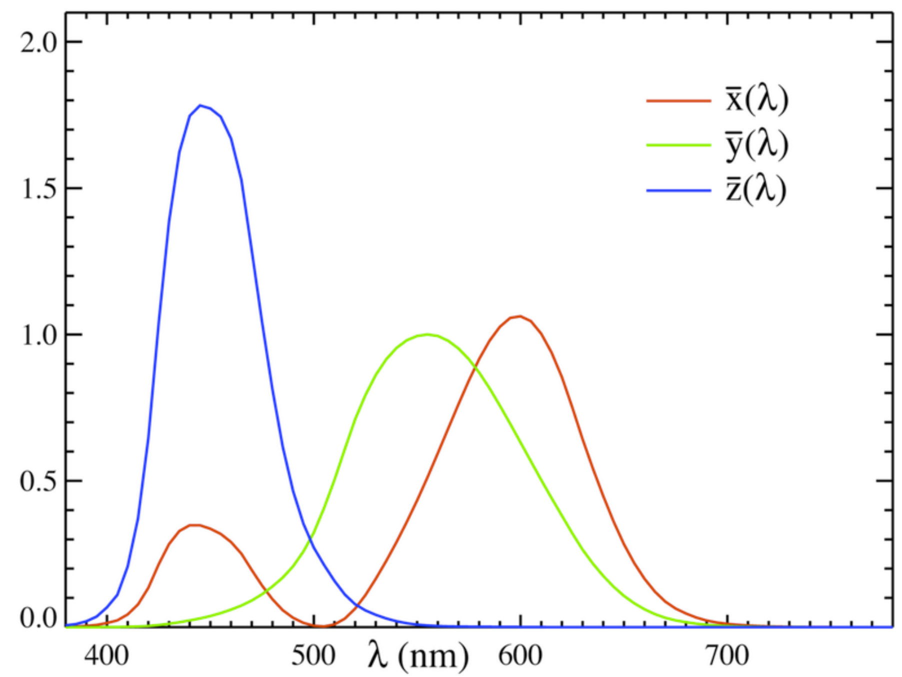
  $r = \int \bar{r}(\lambda)P(\lambda)d\lambda \qquad g = \int \bar{g}(\lambda)P(\lambda)d\lambda \qquad b = \int \bar{b}(\lambda)P(\lambda)d\lambda$



- Problems:
  - Negative comparative values
  - Some spectral colors **cannot be realized** by combination of 3 primaries!
  - RGB is not a perfect color space—but technically feasible!

XYZ Color Space (CIE 1931)

- Goal:
  - Nonnegative color matching functions
  - Description of all perceivable colors
- Imaginary oversaturated primary colors (not physically realizable)
- Y-component chosen such that it corresponds to luminance $\bar{y}(\lambda)$
  - $X = \int \bar{x}(\lambda) P(\lambda) d\lambda$
  - $Y = \int \bar{y}(\lambda) P(\lambda) d\lambda$
  - $Z = \int \bar{z}(\lambda) P(\lambda) d\lambda$
- XYZ are the tristimulus values for the "CIE standard observer"
- CIE 1931 addresses fovea (2°)
- CIE 1964 covers 10° (excluding 2°)



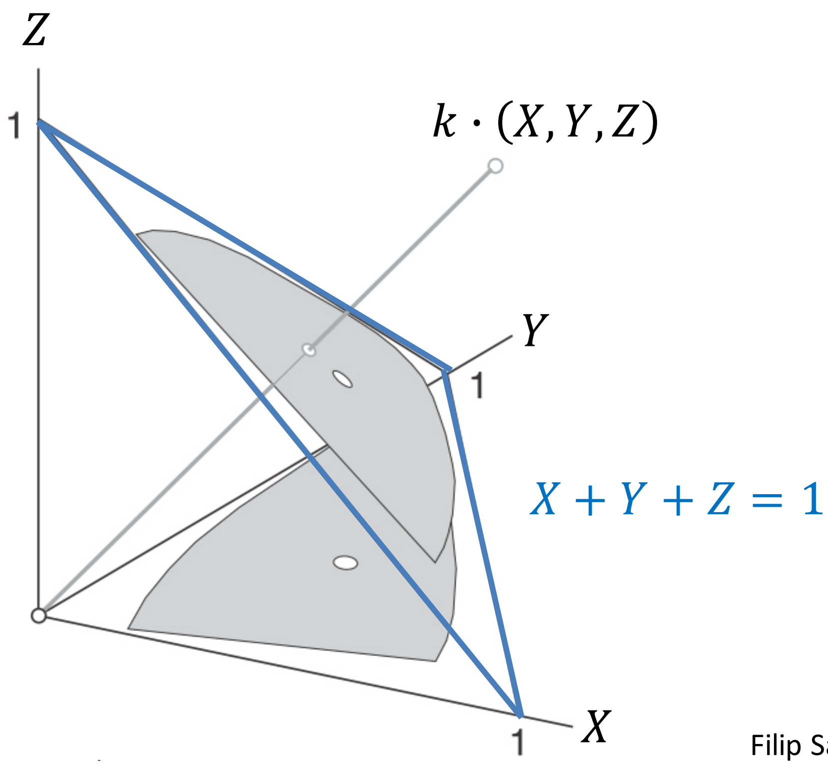
Chromaticity (CIE xyY Color Space)

Representation of all visible colors

- Observation: all $k \cdot (X, Y, Z)$ ($k > 0$) represent same color, only intensity varies

$\rightarrow$ Normalization onto $X + Y + Z = 1$ plane

- Subsequent projection onto the XY-plane (omit z)
- xy-diagram still contains all colors



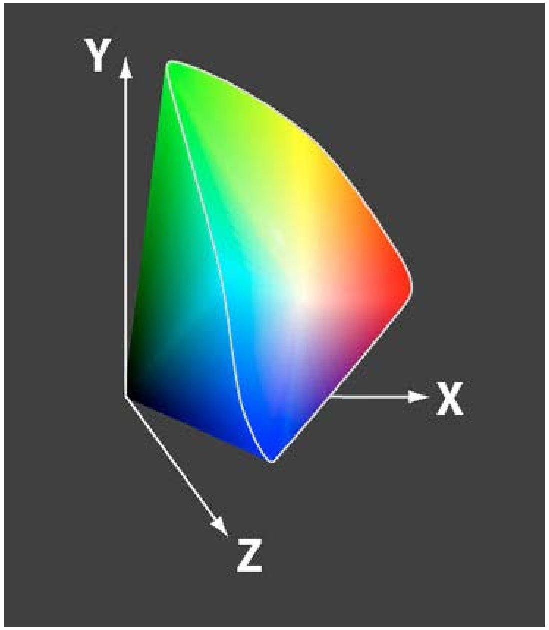
$$x = \frac{X}{X + Y + Z}$$

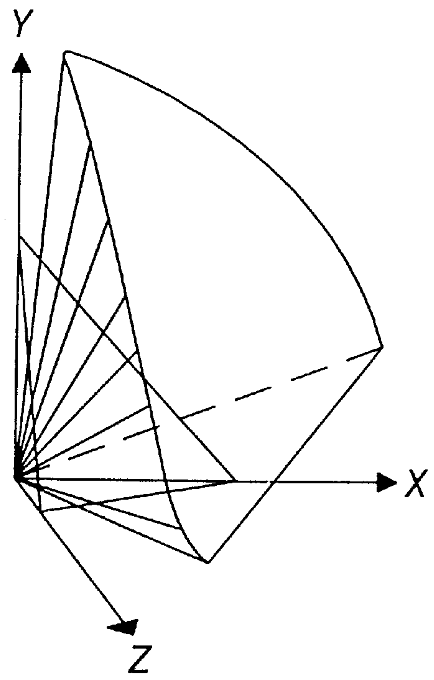
$$y = \frac{Y}{X + Y + Z}$$

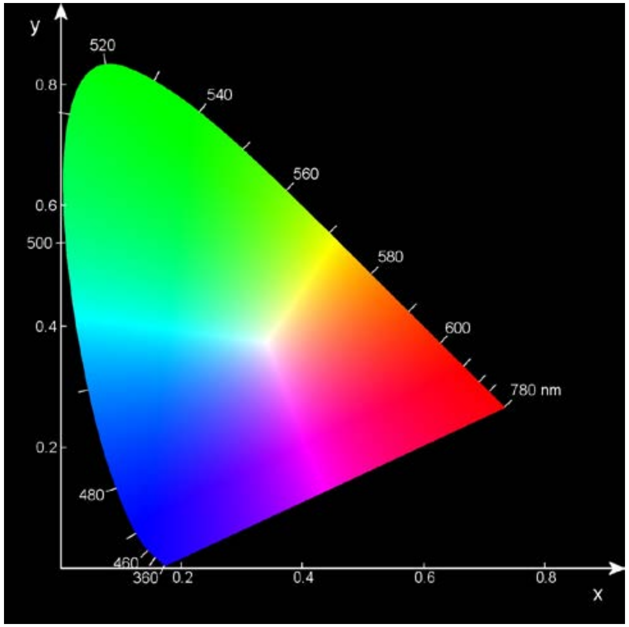
$$z = \frac{Z}{X + Y + Z} = 1 - x - y$$

Chromaticity (CIE xyY Color Space)

- Split information into brightness Y and color (chromaticity) xy
- Normalize $x = \frac{X}{X+Y+Z}$, $y = \frac{Y}{X+Y+Z}$, and $z = \frac{Z}{X+Y+Z} = 1 - x - y$
- If a color is given by x, y, one needs to additionally provide brightness Y
  $\rightarrow$ Tristimulus values can be determined from CIE xyY
  - $X = \frac{Y}{y} x$ and $Z = \frac{Y}{y}(1 - x - y)$



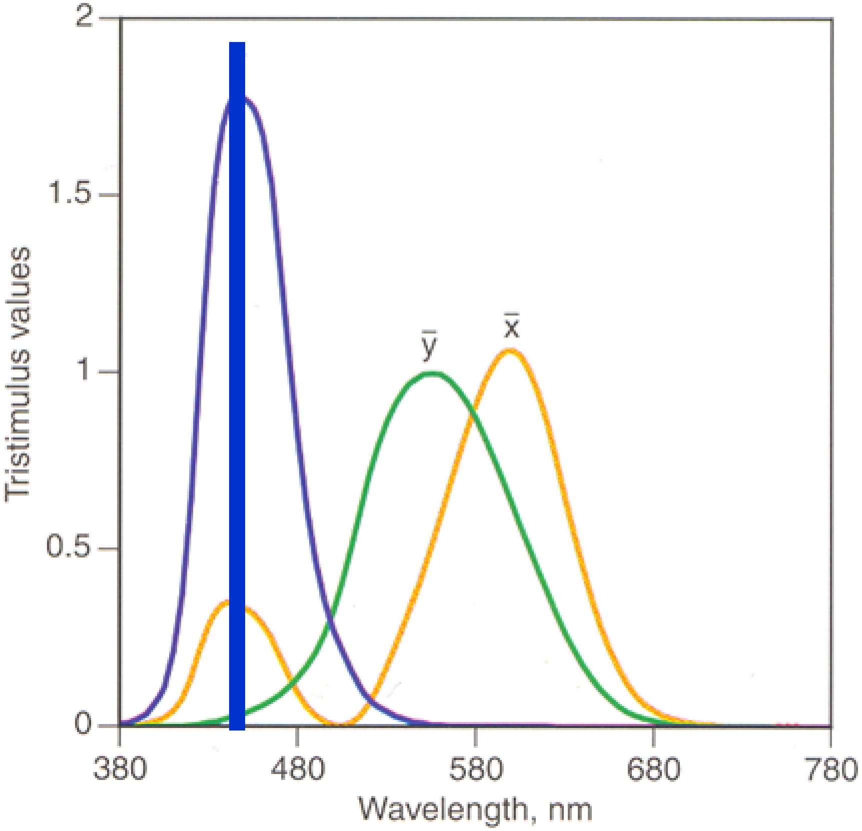




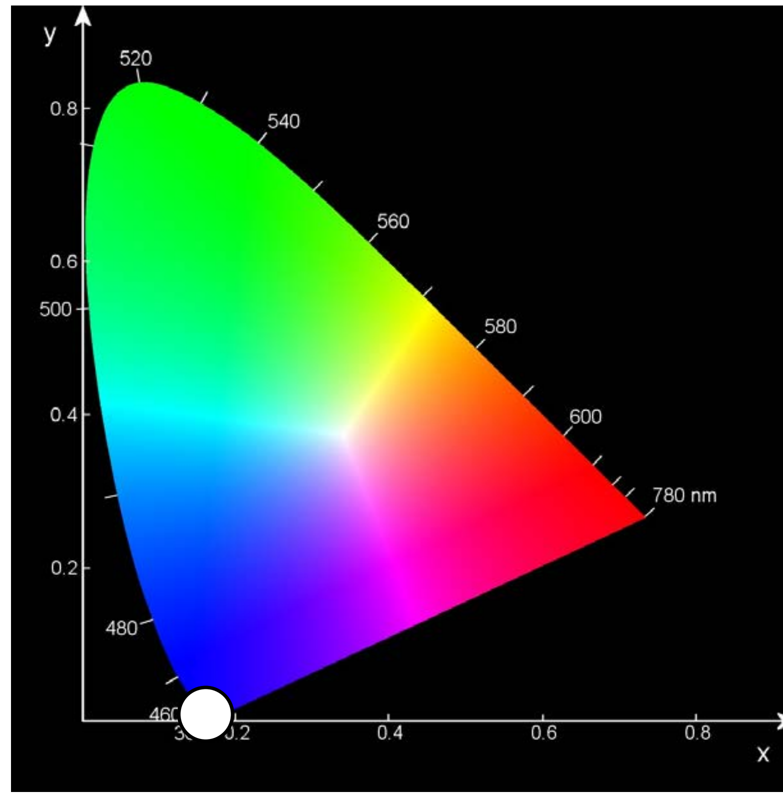
chromaticity values only depend on wavelength and saturation

Spectral Colors in Chromaticity Diagram

- Large Z-value $\rightarrow x$ and y small
- $x = \frac{X}{X+Y+Z}$ and $y = \frac{Y}{X+Y+Z}$

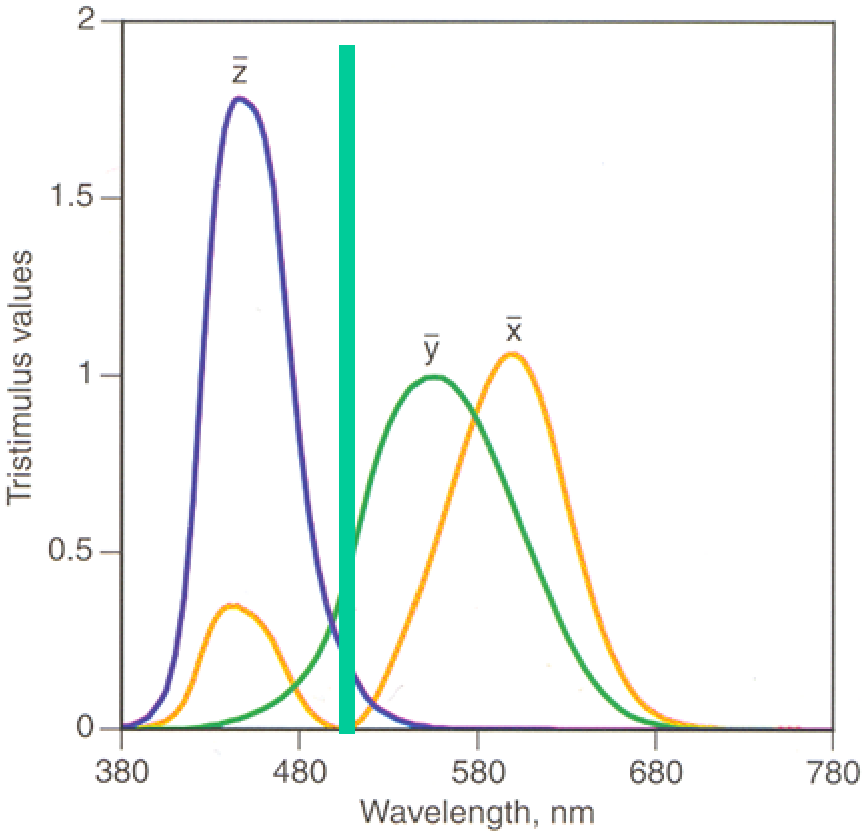


The 1931 standard observer, as it is usually shown.

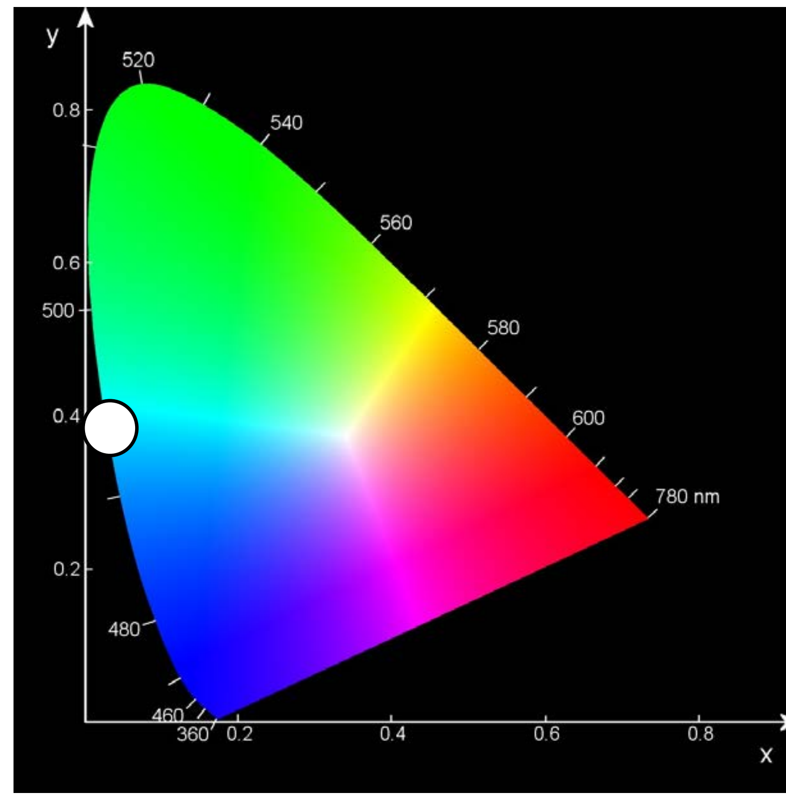


Spectral Colors in Chromaticity Diagram

- y larger
- $x = \frac{X}{X+Y+Z}$ and $y = \frac{Y}{X+Y+Z}$

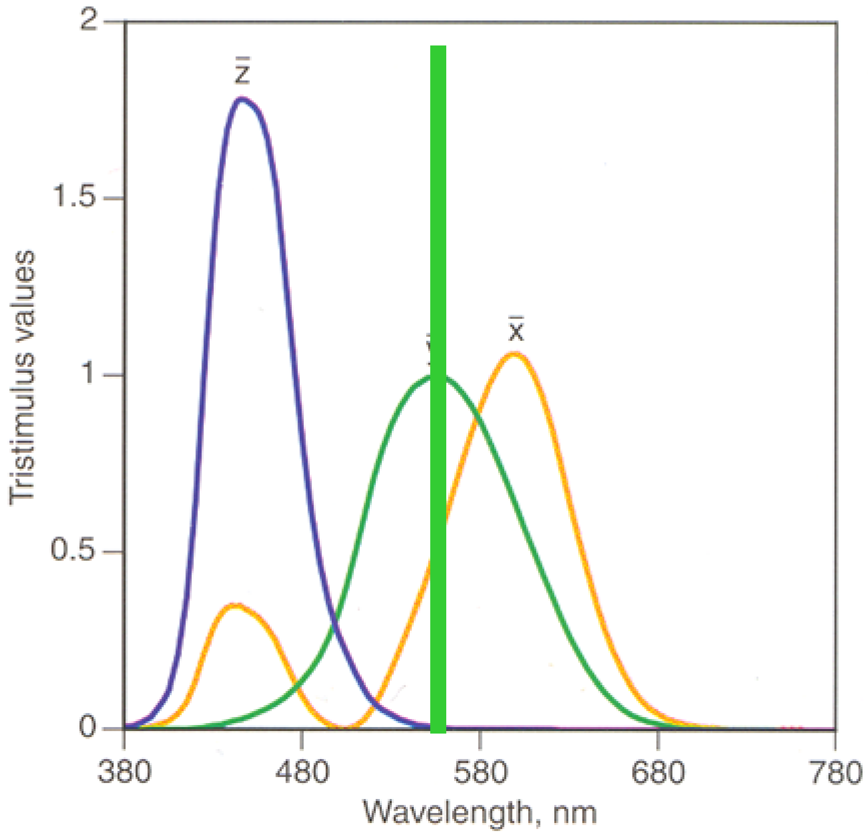


The 1931 standard observer, as it is usually shown.

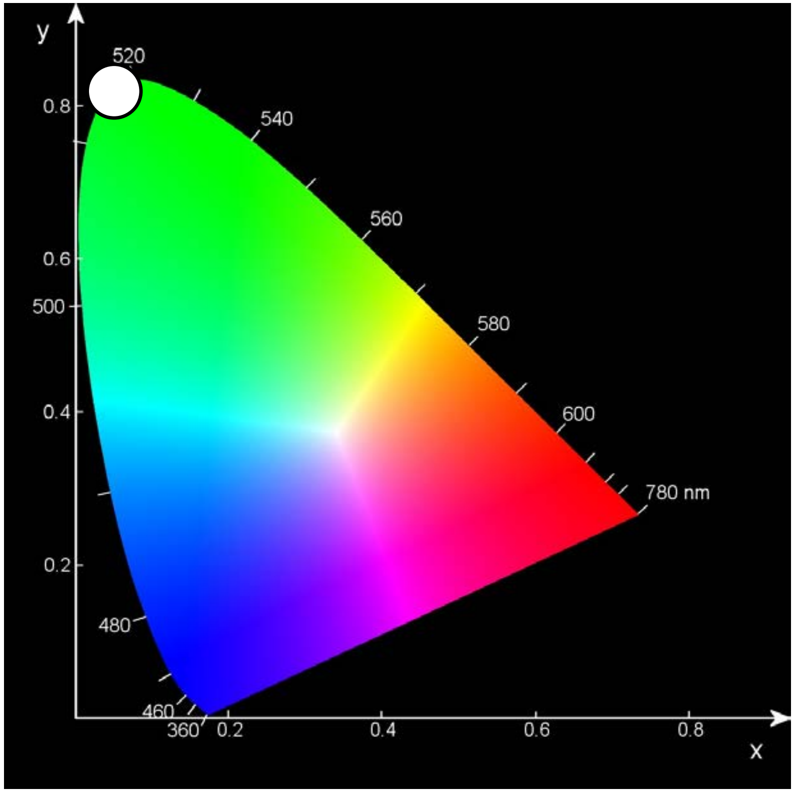


Spectral Colors in Chromaticity Diagram

- y is large
- $x = \frac{X}{X+Y+Z}$ and $y = \frac{Y}{X+Y+Z}$

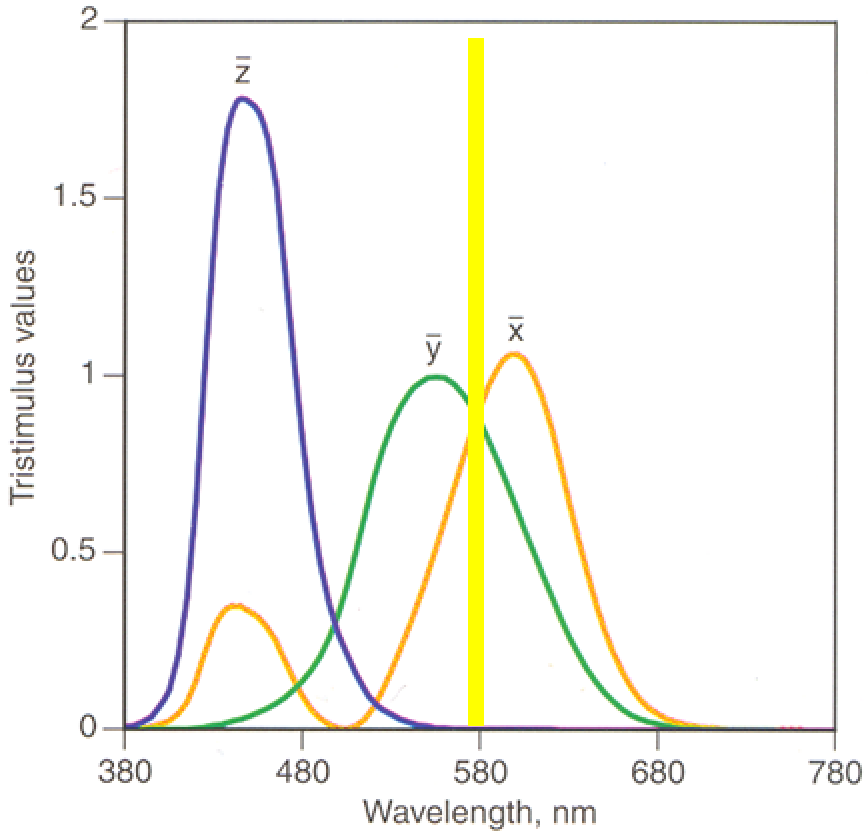


The 1931 standard observer, as it is usually shown.

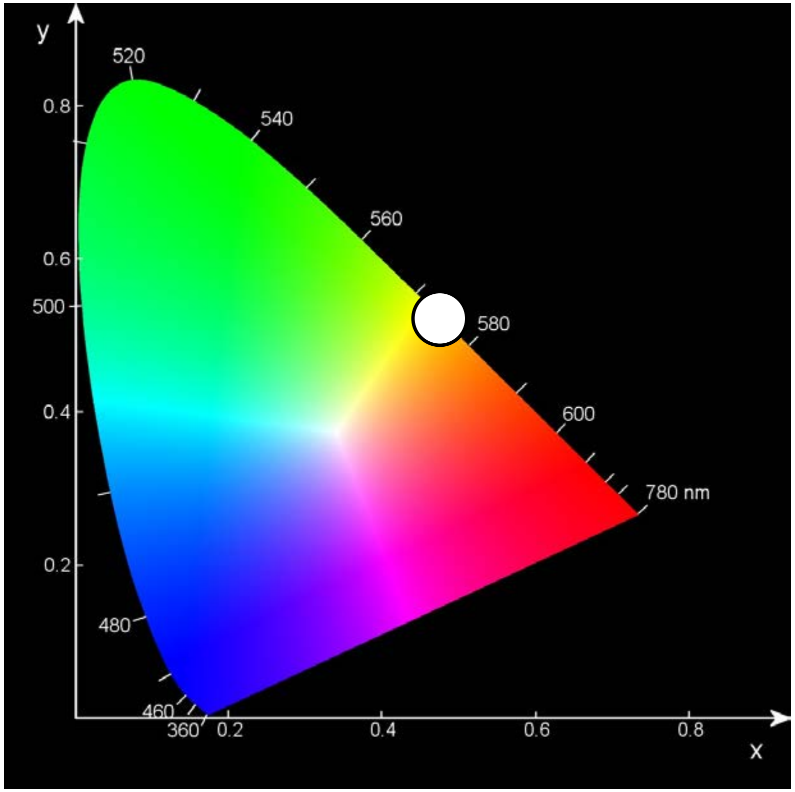


Spectral Colors in Chromaticity Diagram

- y slowly decreases, x increases
- $x = \frac{X}{X+Y+Z}$ and $y = \frac{Y}{X+Y+Z}$

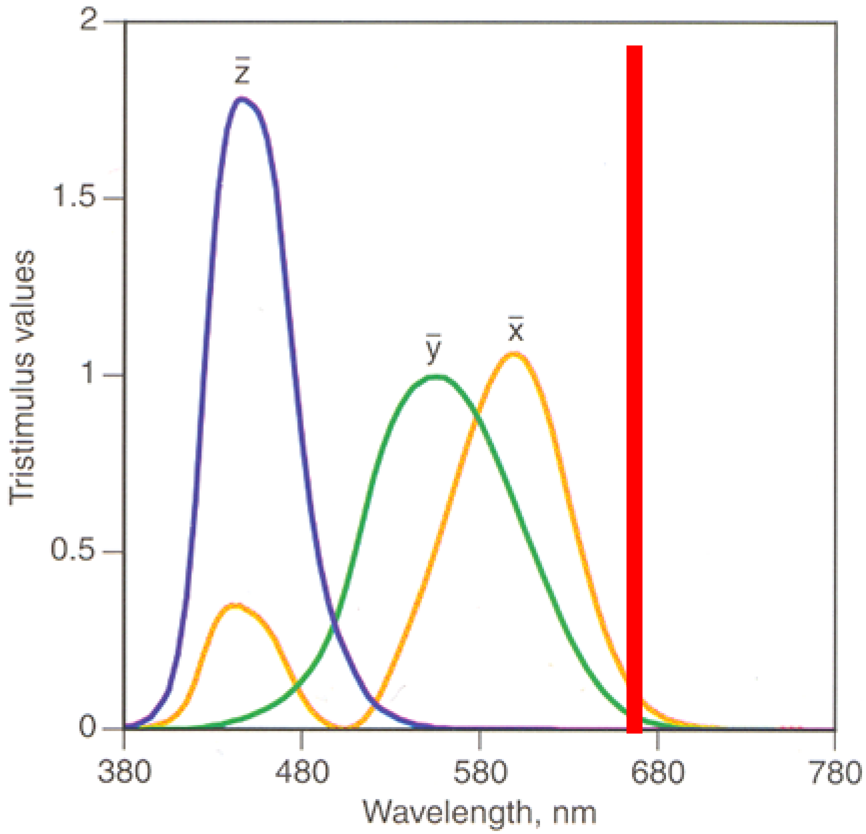


The 1931 standard observer, as it is usually shown.

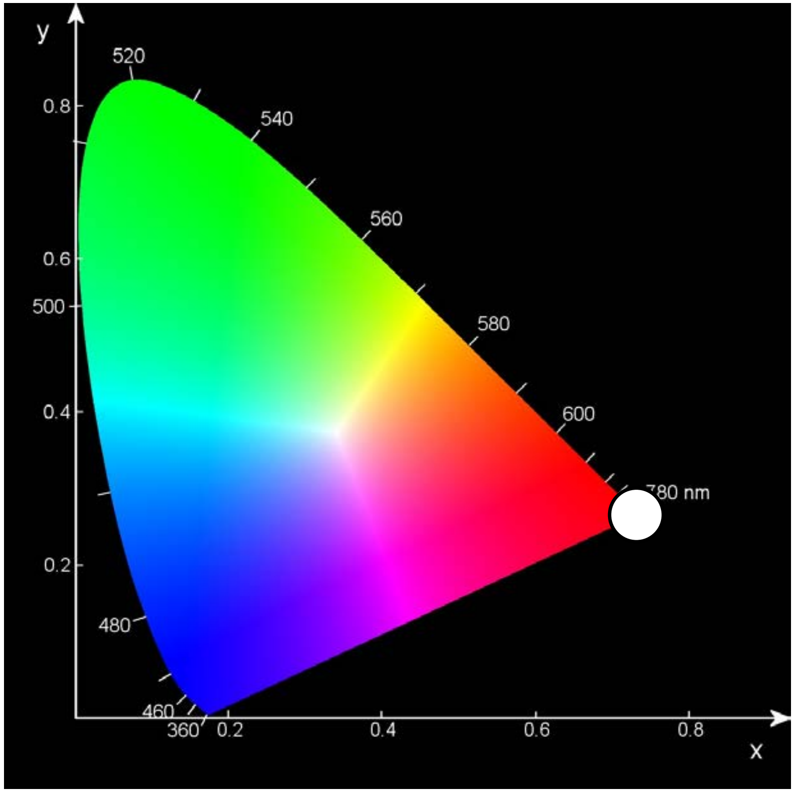


Spectral Colors in Chromaticity Diagram

- y decreases further, x is still large (we divide by $X + Y + Z$)
- $x = \frac{X}{X+Y+Z}$ and $y = \frac{Y}{X+Y+Z}$

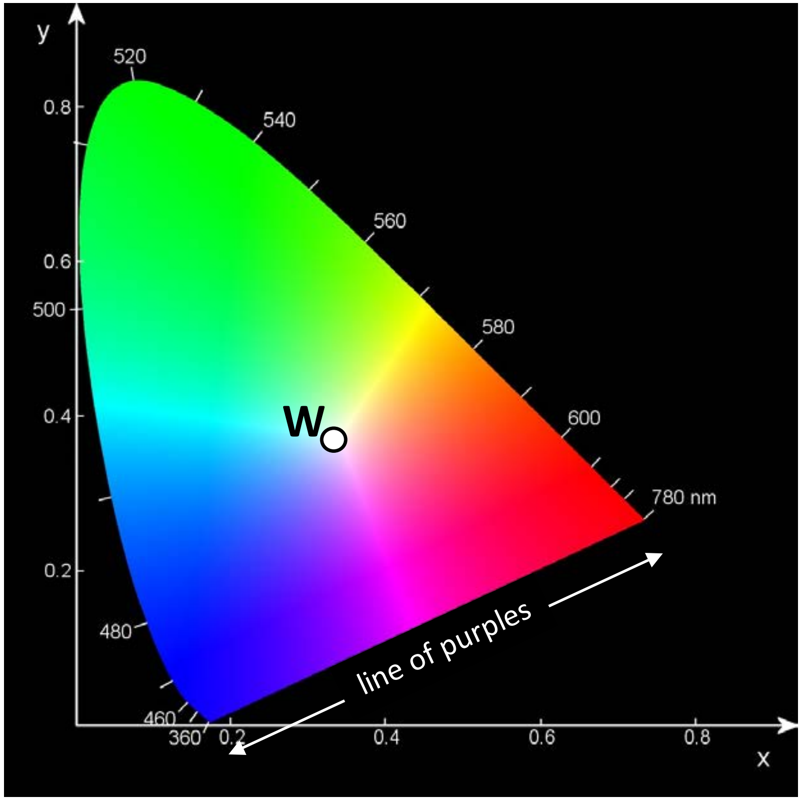


The 1931 standard observer, as it is usually shown.



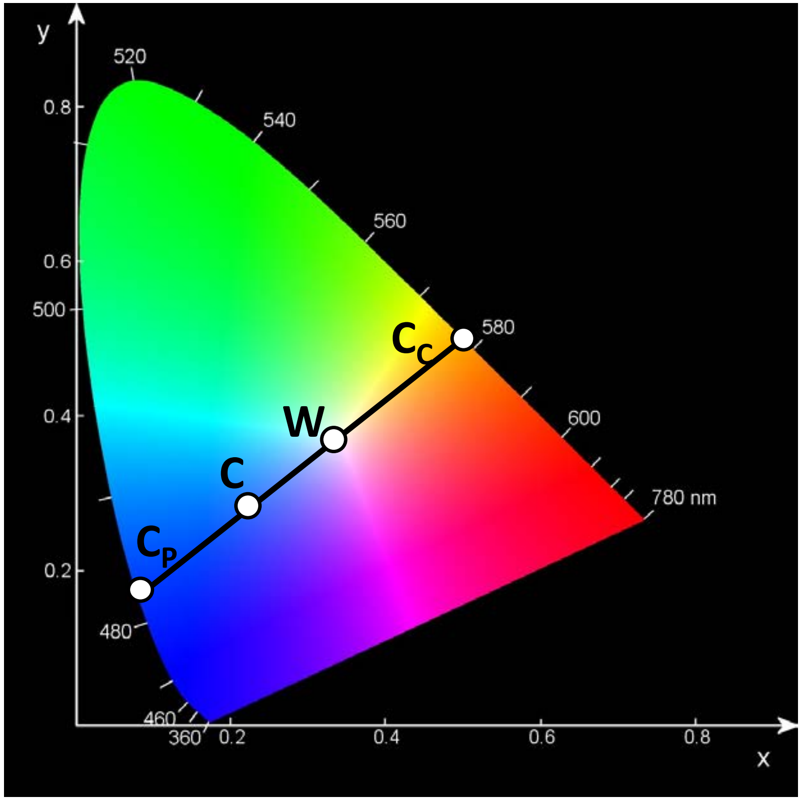
Chromaticity Diagram (CIE xyY Color Space)

- Contains all visible colors (**gamut** of human perception)
- White point W (x=y=z=1/3) (roughly corresponds to sunlight)
- Spectral colors are located along boundary curve and correspond to monochromatic light
- Strange shape results from the sensitivity curves
- Line of purples: set of all fully saturated colors which are *not* spectral colors



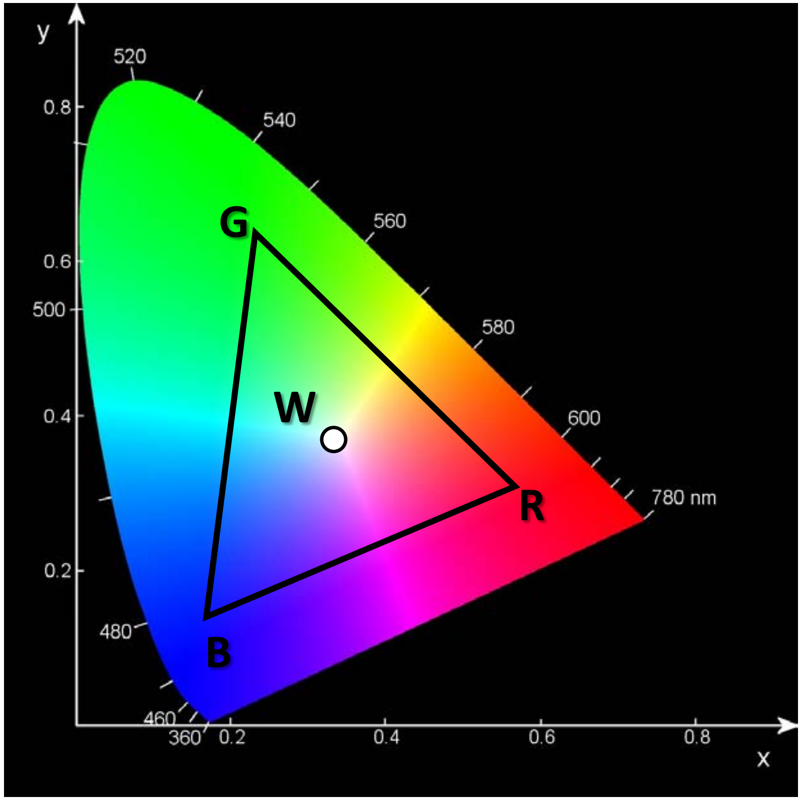
Chromaticity Diagram (CIE xyY Color Space)

- Colors along line between 2 boundary points can be obtained by mixing the respective colors at the endpoints
- Pure color C_P: line from white point through C to pure color
- Complementary color C_c: line through white point to opposite boundary
- All colors within a triangle by mixing colors at its vertices



Chromaticity Diagram (CIE xyY Color Space)

- The representable colors of an output device (resp. recordable colors of an acquisition device) are described by the **gamut**
- Gamut mapping: mapping of gamuts between devices with the goal to avoid (reduce) color shifts
- Example: RGB monitor
  - $\mathbf{C} = \mathrm{r}\mathbf{R} + \mathrm{g}\mathbf{G} + \mathrm{b}\mathbf{B}$ with $(\mathrm{r}, \mathrm{g}, \mathrm{b}) \in [0,1]^3$
  - The 3 primary colors span a triangle within the diagram
  - Colors outside of this gamut would require negative r, g, or b
- If 3 primaries are given, larger gamut can only be achieved by using additional primary colors!



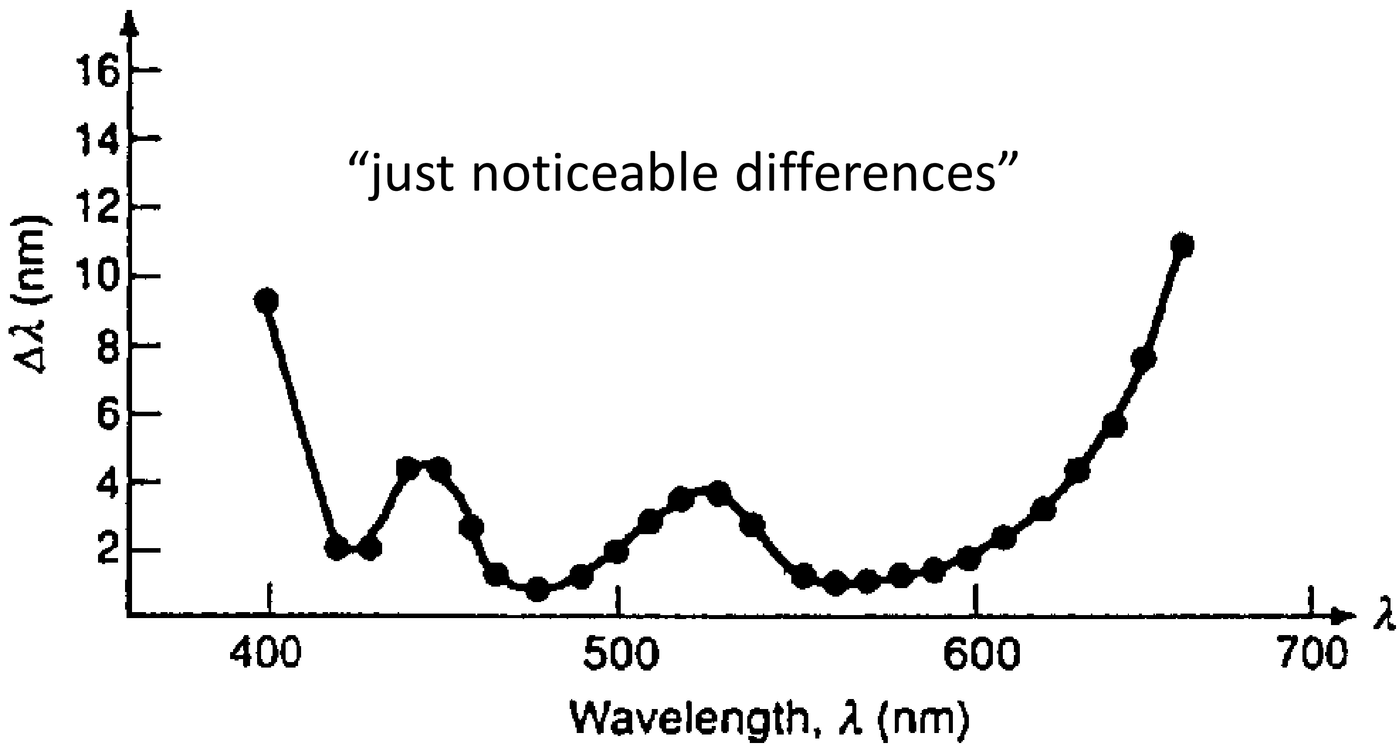
Dynamic Range

- Not only gamut is limited, but also minimum and maximum intensity
  - Monitor: emitted light, print: reflected light
  - “Darkest black” vs. “brightest white”

image: Jack Tumblin

Perceptually Uniform Color Models

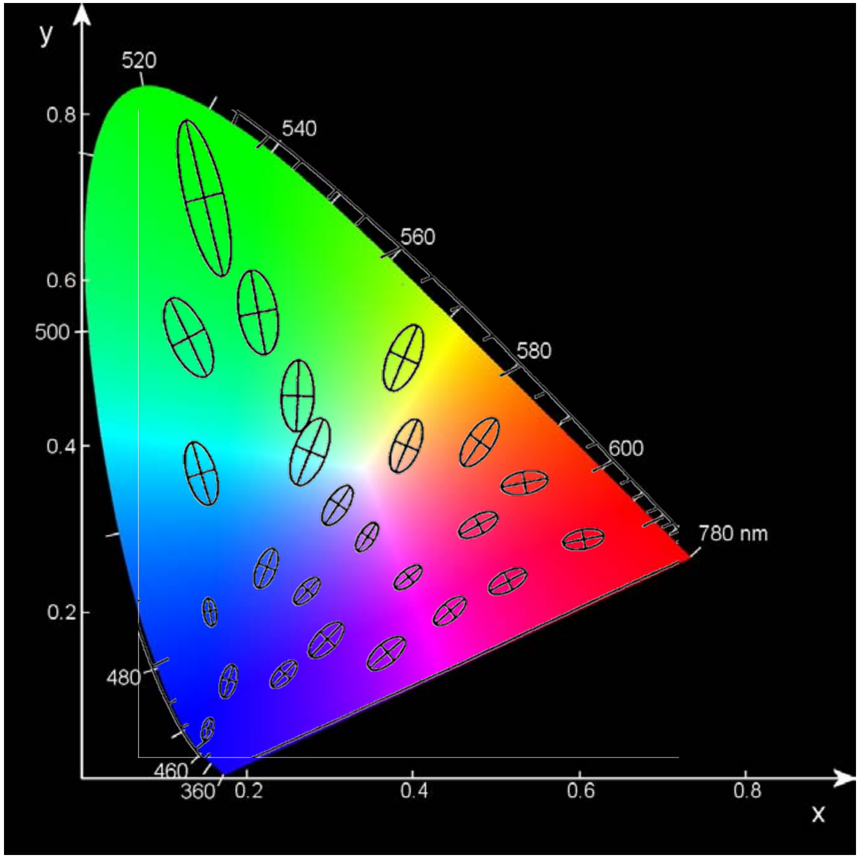
- CIE XYZ is not an intuitive system:
  changes in color values do not result in equally large differences in perceived color
- CIE L*u*v*; CIE L*a*b* are obtained by nonlinear transformations of the CIE XYZ, with the goal of a perceptually uniform color space



Perceptually Uniform Color Models

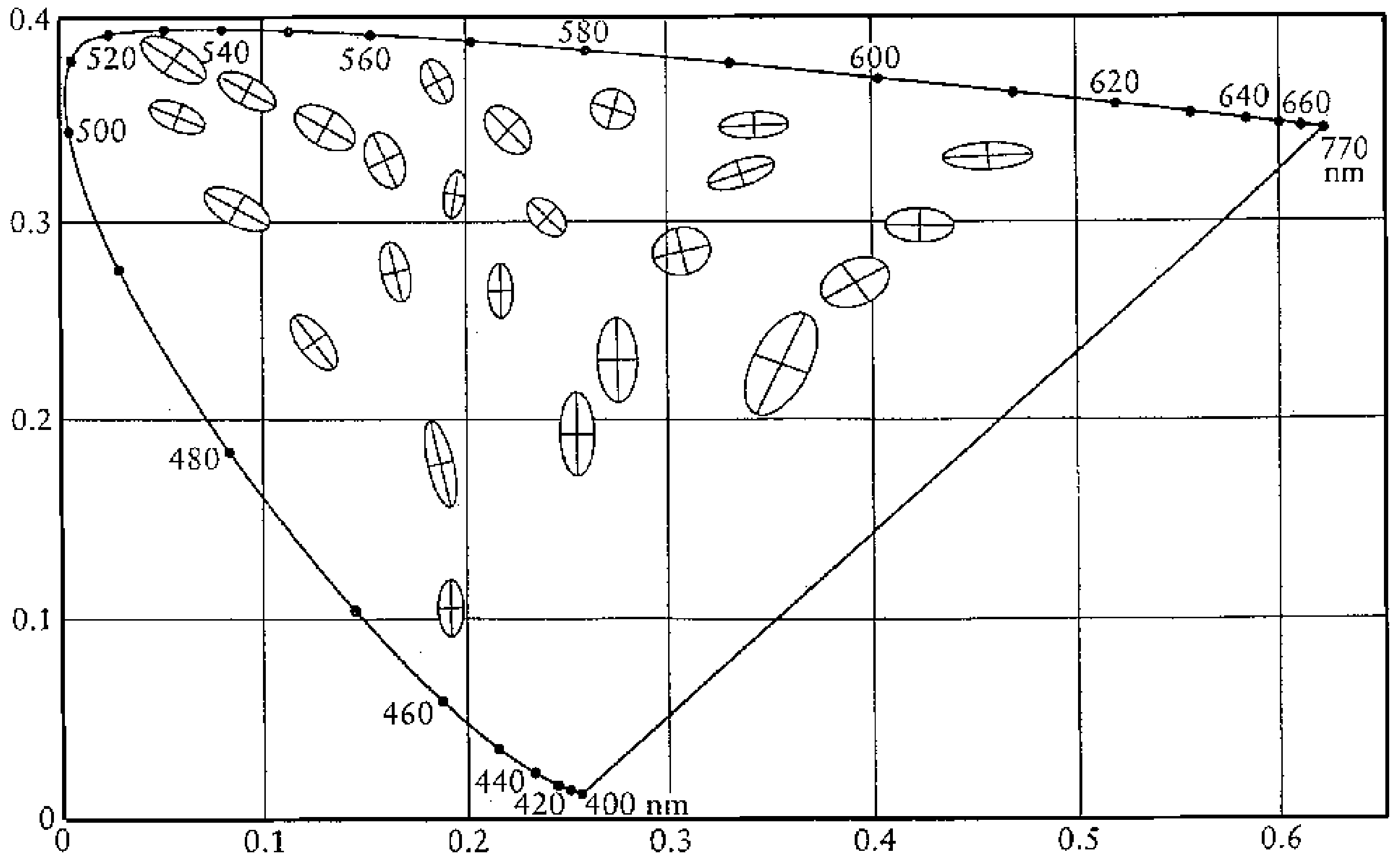
MacAdam ellipses in CIE xy

- Colors within a MacAdam ellipse indistinguishable from color at its center
- Size and orientation of the ellipses vary substantially within color space
- Size of ellipses overemphasized here by factor of 10!

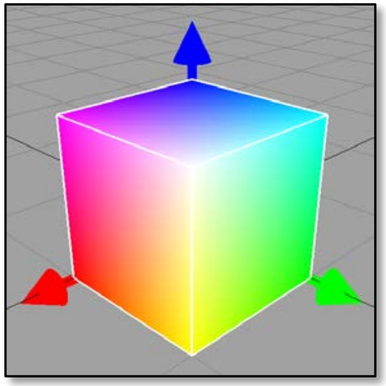


Perceptually Uniform Color Models

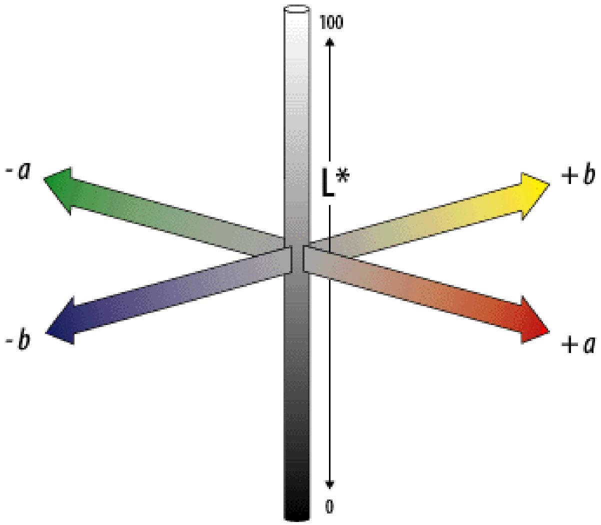
MacAdams ellipses in approx. perceptually uniform color space (CIE L*u*v*)



Psychophysical Color Model



- Most color models are rather guided by technical constraints, such as
  - which colors can be represented? (RGB, CIE)
  - how can choice of colors be simplified? (HSV)
  - …
- But: What is the perceived difference between two colors?
- CIE L*a*b* model
  - Luminance, red–green, blue–yellow
  - Intuitive reproduction of the perception
  - Euclidean distances in color space correspond to perceived color distances



Example: CIELAB, CIE Lab, CIE L*a*b*

- **Linear** RGB to XYZ:

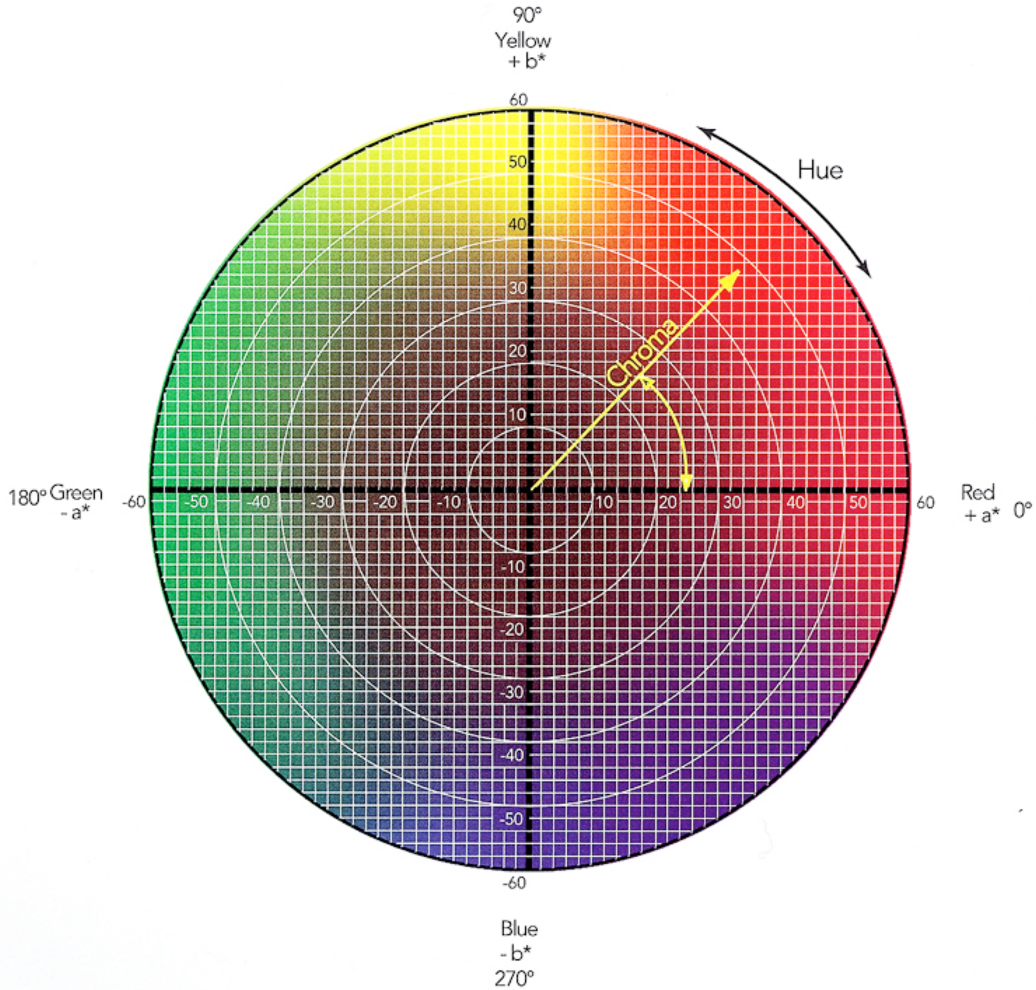
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- XYZ to CIELAB:

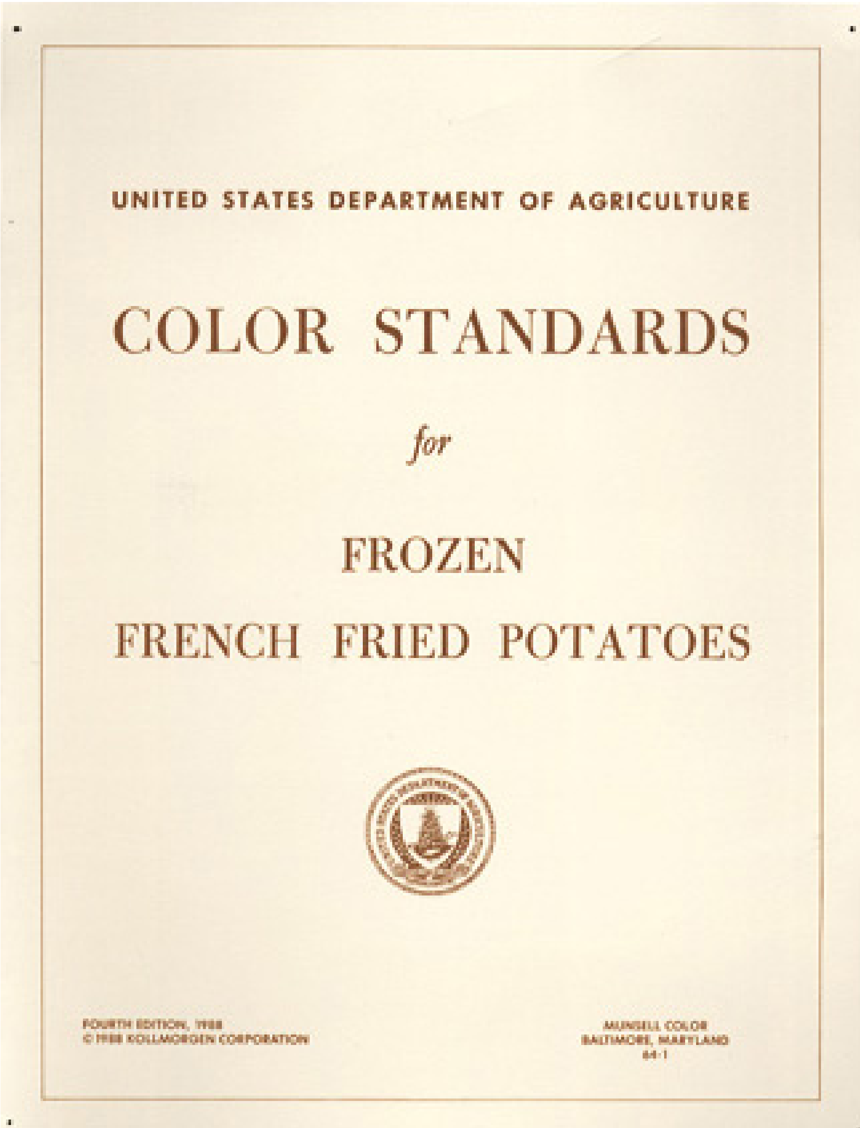
$$\begin{aligned} L^* &= 116\left(Y/Y_n\right)^{1/3} - 16 \qquad \text{for } Y/Y_n > 0.008856 \\ a^* &= 500\left[\left(X/X_n\right)^{1/3} - \left(Y/Y_n\right)^{1/3}\right] \\ b^* &= 200\left[\left(Y/Y_n\right)^{1/3} - \left(Z/Z_n\right)^{1/3}\right] \end{aligned}$$

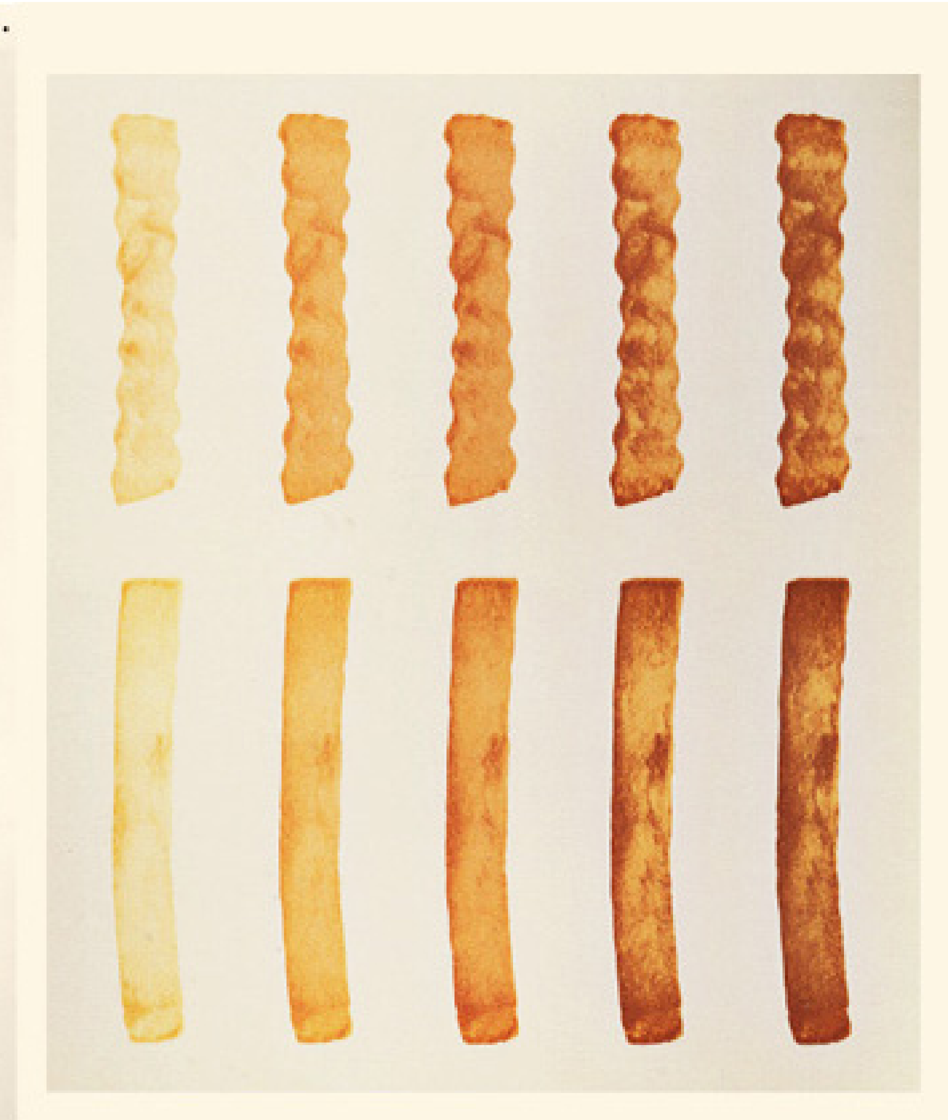
  - *L**: lightness, *a**: red–green, *b**: blue–yellow
  - X_n, Y_n, Z_n: tristimulus value of reference white point

Example: CIELAB, CIE Lab, CIE L*a*b*



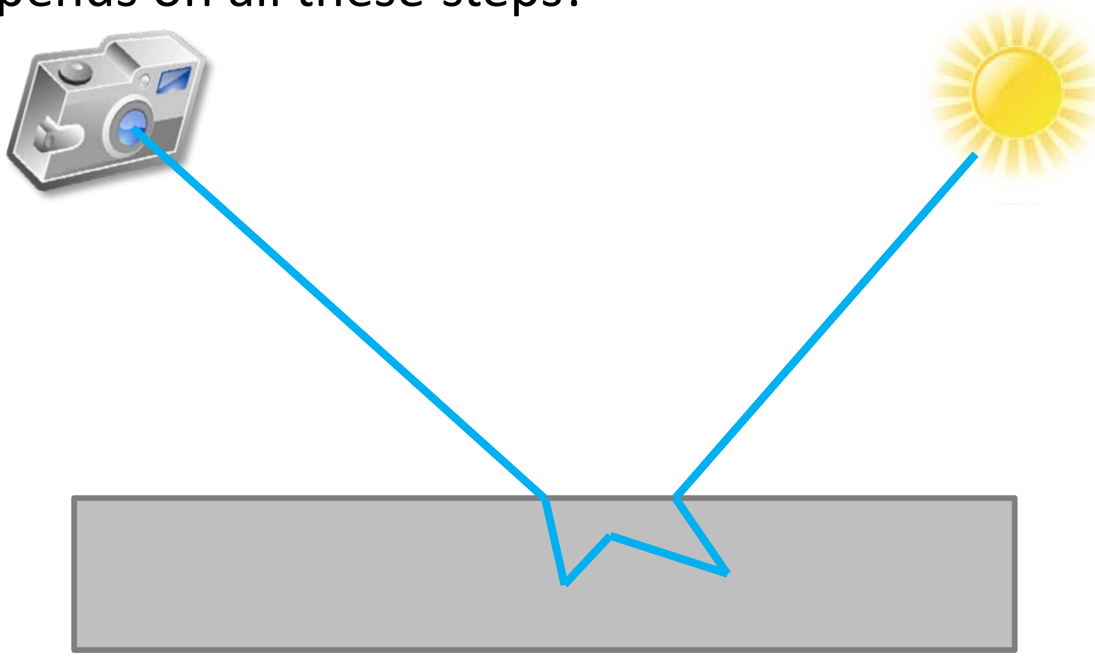
Another Color Model …





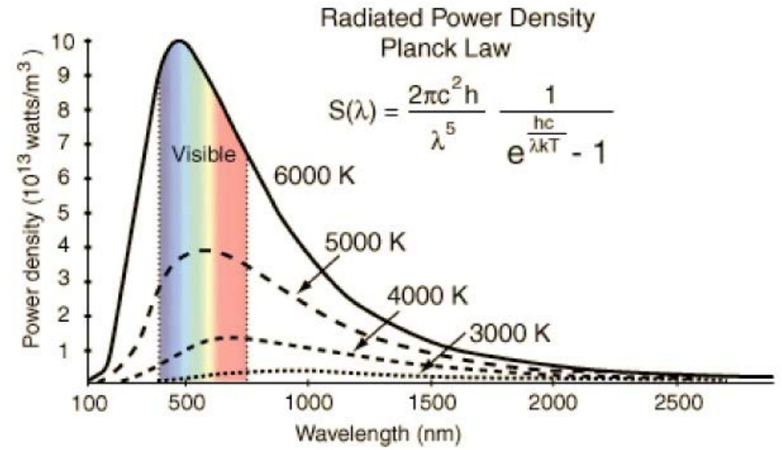
Color Phenomena

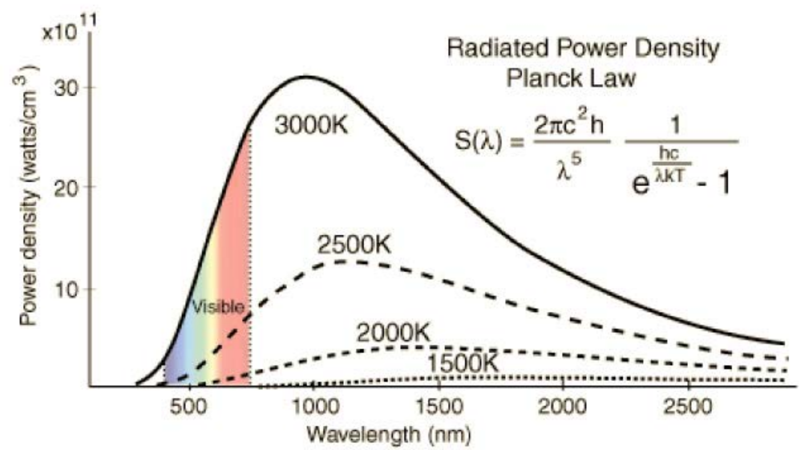
- Generally, light does not shine directly into the eye
- Light follows transport paths, e.g.,
  - from the light source
  - through the atmosphere
  - onto object surfaces (possibly within material)
  - through the atmosphere
- The color depends on all these steps!

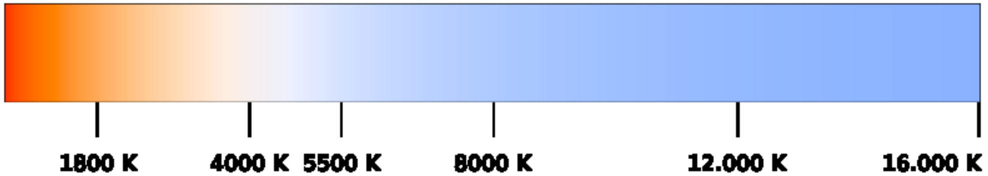


Thermal Radiation Source

- Warm objects emit energy
- Wavelength/frequency is temperature-dependent
- If temperature is high enough, part of emitted energy is within visible spectrum
- Black body radiation
  - Idealized radiation source
  - Color temperature = temperature of black body corresponding to a certain color

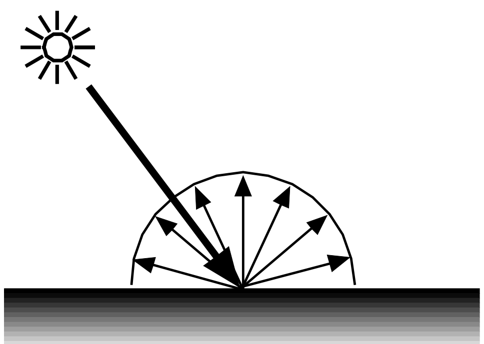




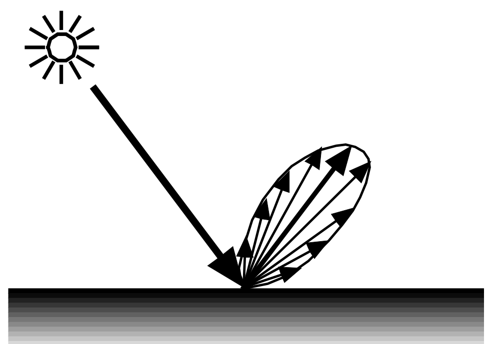


Reflection at Surfaces

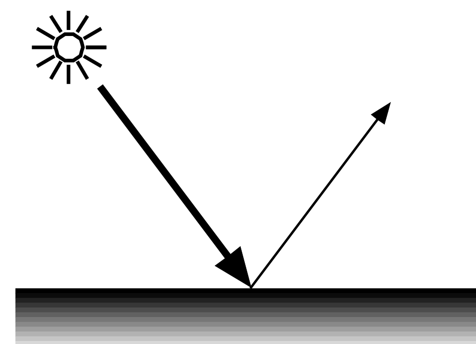
diffuse surface



shiny surface

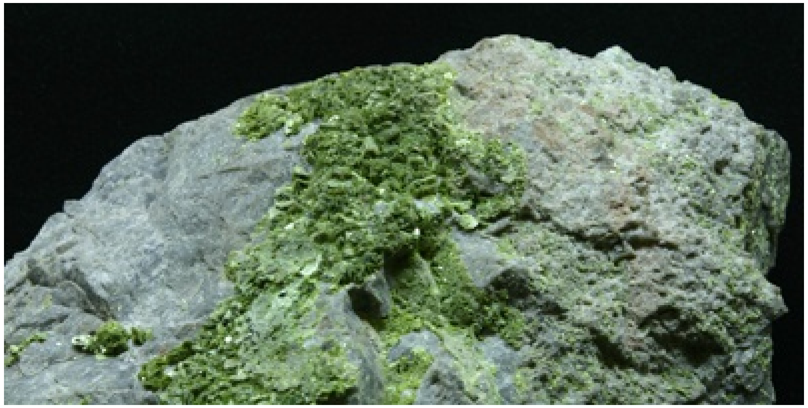


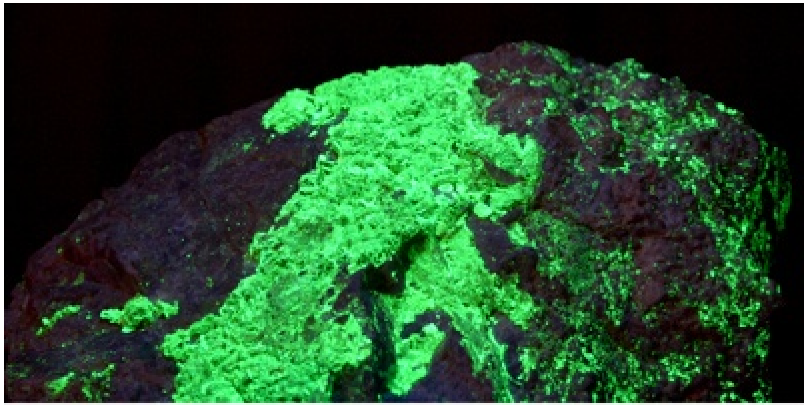
perfect mirror



Luminescence

- Luminescence phenomena (cold light)
  - Absorption of photons, reemission of light at different wavelength
  - Fluorescence: typically short-time (~10^{-6} s), induced by electromagnetic radiation
  - Phosphorescence: longer time scales, up to several hours
  - Time-dependency and spectral effects





Uranium ore: Autunite
(left: normal light, right: fluorescence due to illumination with UV light)
www.geoberg.de

Light Transport

- Transport path
  - Sun → atmosphere → water (refraction) → surface →
    → water (refraction) → atmosphere → eye

Light Transport

- Scattering within atmosphere

Light Transport

- Haze/scattering within atmosphere

Henrik Wann Jensen

Light Transport

- Scattering, emission, …

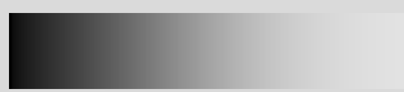
T.A.Rector (NOAO/AURA/NSF) and the Hubble Heritage Team (STScI/AURA/NASA)

Raster Graphics

- Framebuffer and representation
  - Gamma correction, alpha channel
- Signals and sampling
- Image manipulation

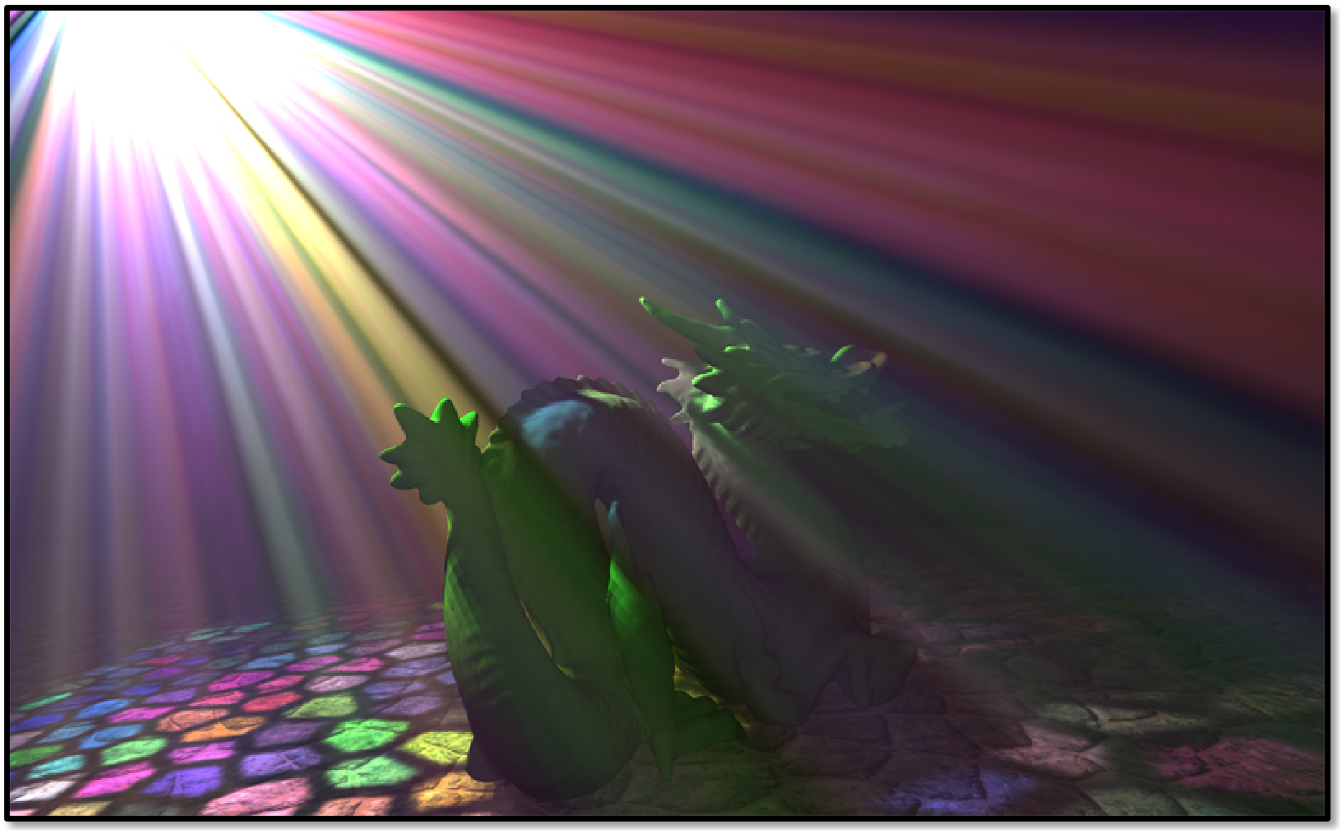
Raster Graphics

- Discrete representation of an image
  - Rectangular grid of pixels
  - Resolution: width × height = number of pixels
    - E.g., 640 × 480, 1280 × 1024, …
  - Color depth: number of bits per pixel
- The image is stored in the framebuffer for the representation on displays (monitors, projectors, …)

Type	Color	Color Depth
binary, black/white		1 bit/pixel
gray level, intensity		8 bit/pixel (12 bit/pixel medical application)
color with lookup table (LUT)		8 bit/pixel → 256 entries 24 bit / LUT entry
true color		24 bit/pixel

Displays, Framebuffer, and Representation

- Typical values for framebuffers
  - Resolution 1920 × 1080 ≈ 2 M pixels, color depth: 24 bit (8 bit RGB)
  - Approx. 6 MB data → approx. 356 MB/s bandwidth at 60 Hz display
  - Formerly: 16 bit framebuffer (5 bit red, 6 bit green, 5 bit blue "HiColor")

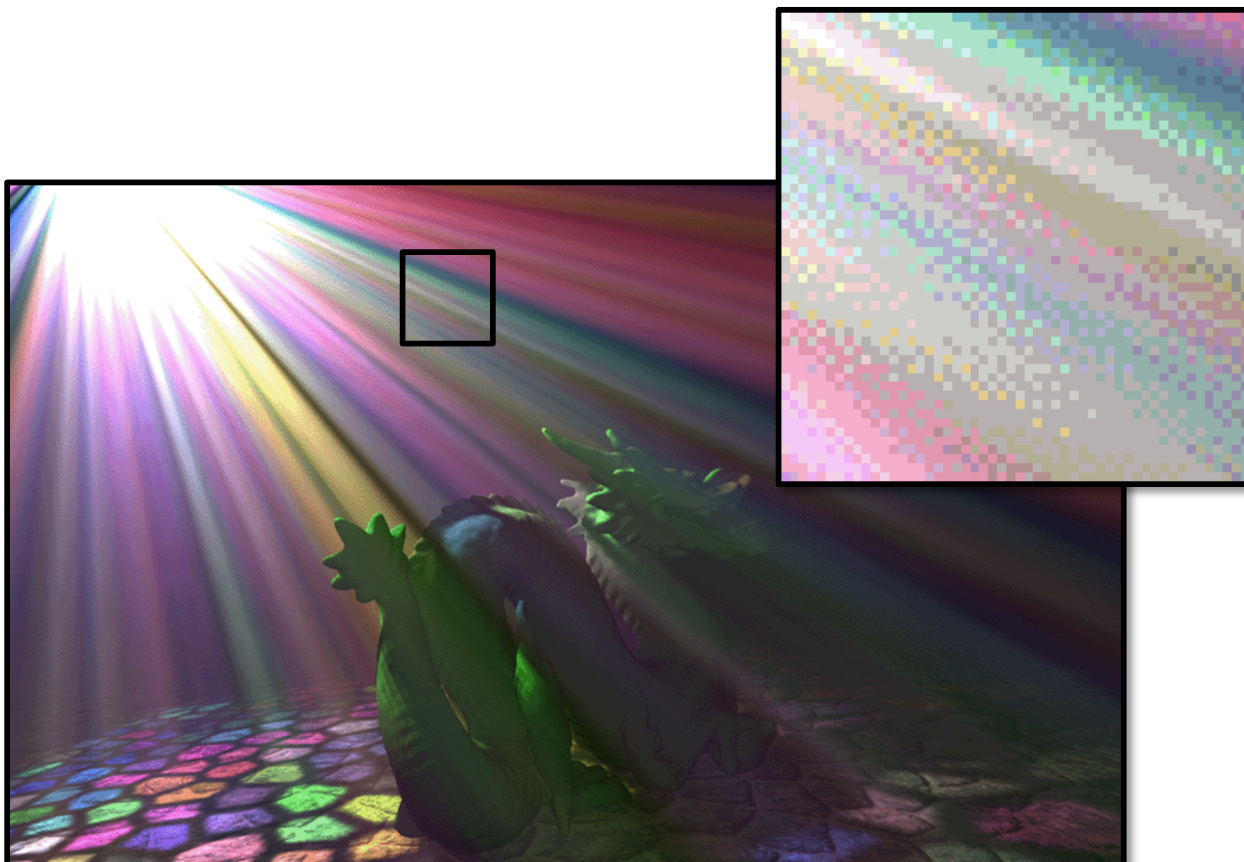


Framebuffer (Historical)

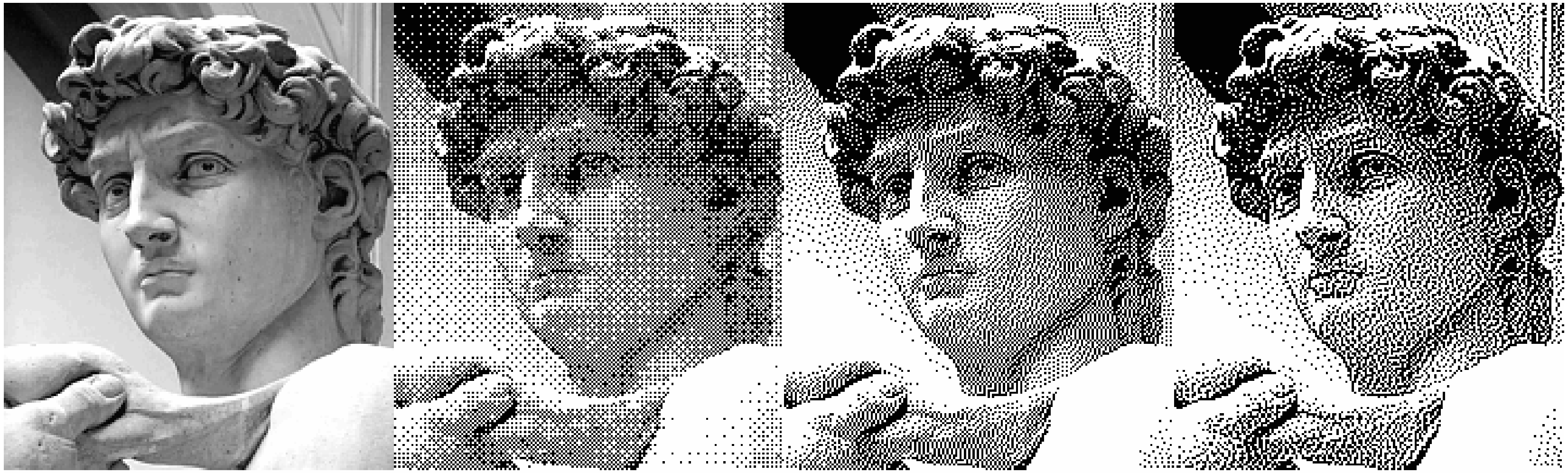
- 8 bit framebuffer
  - Lookup table (256 freely definable colors out of 2^{24})
  - Among others: VGA graphics cards, GIF data format

Framebuffer (Historical)

- 8 bit framebuffer (256 freely definable colors out of 2^{24})
  - Dithering: simulate missing colors by certain arrangement of available colors
  - Perception as mixed colors

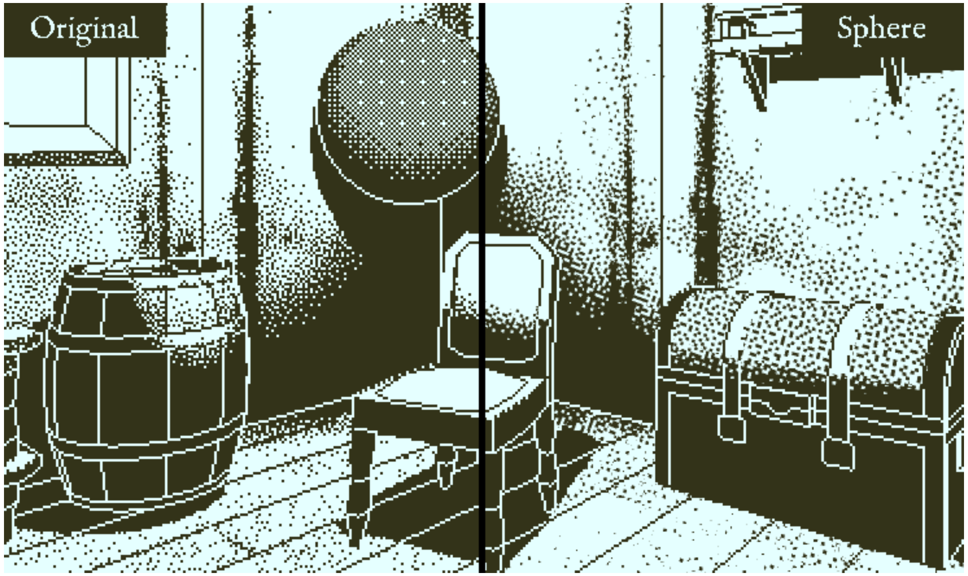


Example: Dithering



original (256 greys) ordered Floyd-Steinberg Jarvis

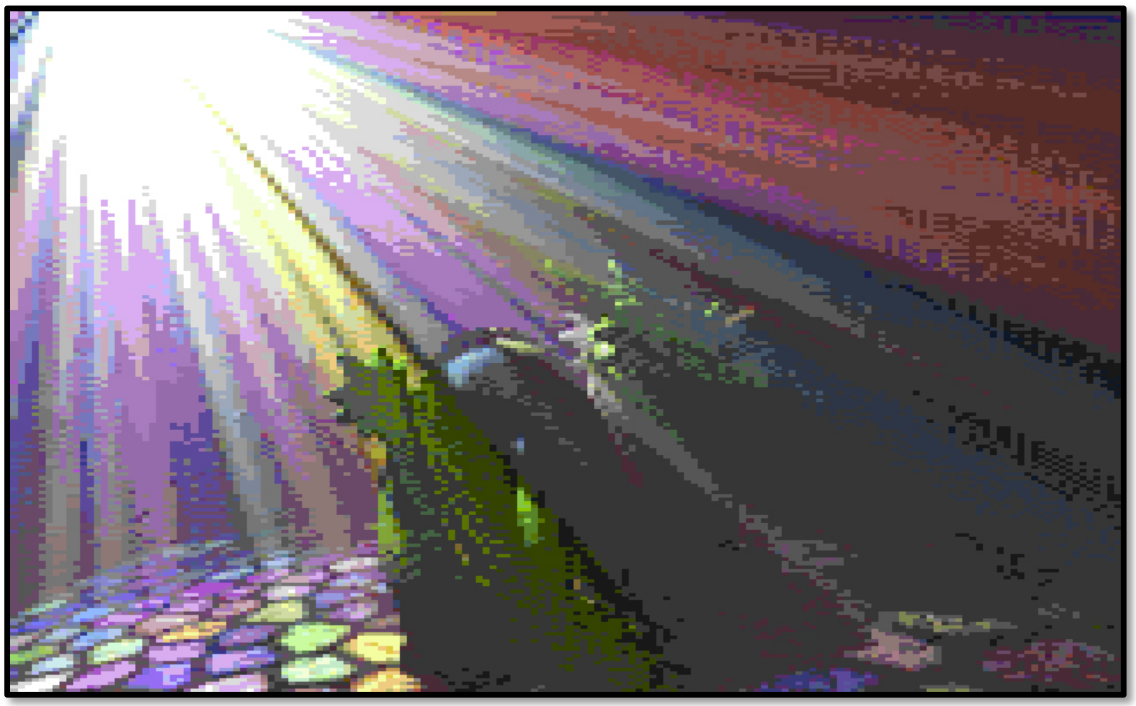
- Recent application: Return of the Obra Dinn (2018)



(https://forums.tigsource.com/index.php?topic=40832.msg1363742#msg1363742)

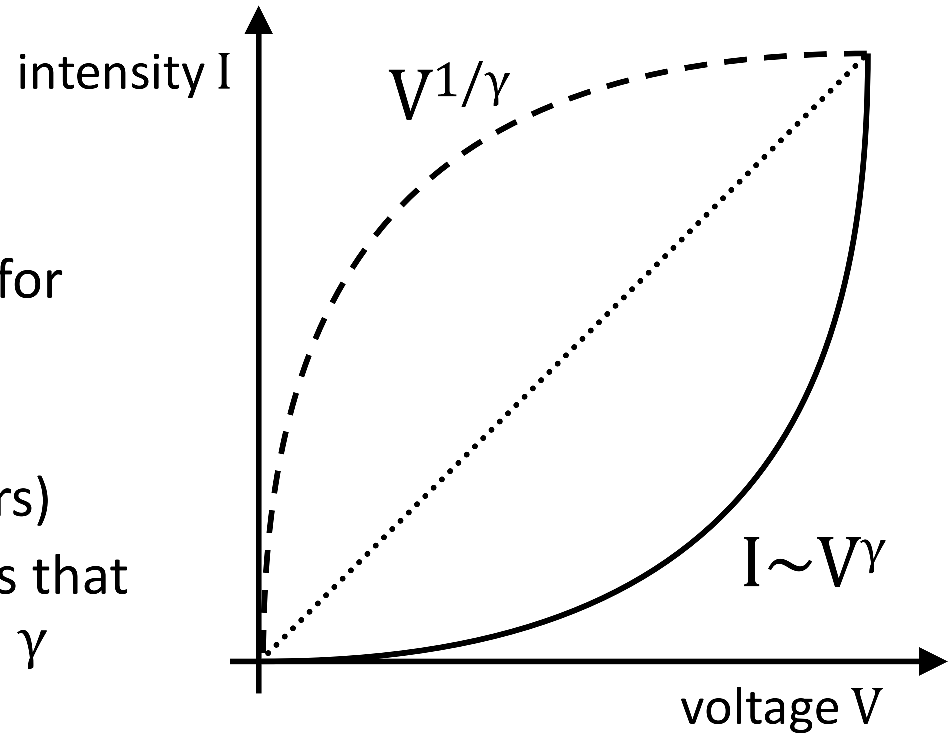
Framebuffer (Nostalgic …)

- Commodore C64 (1982)
  - 160×200 pixels, four colors (out of 16) per 4×8 block



Gamma Correction for Displays

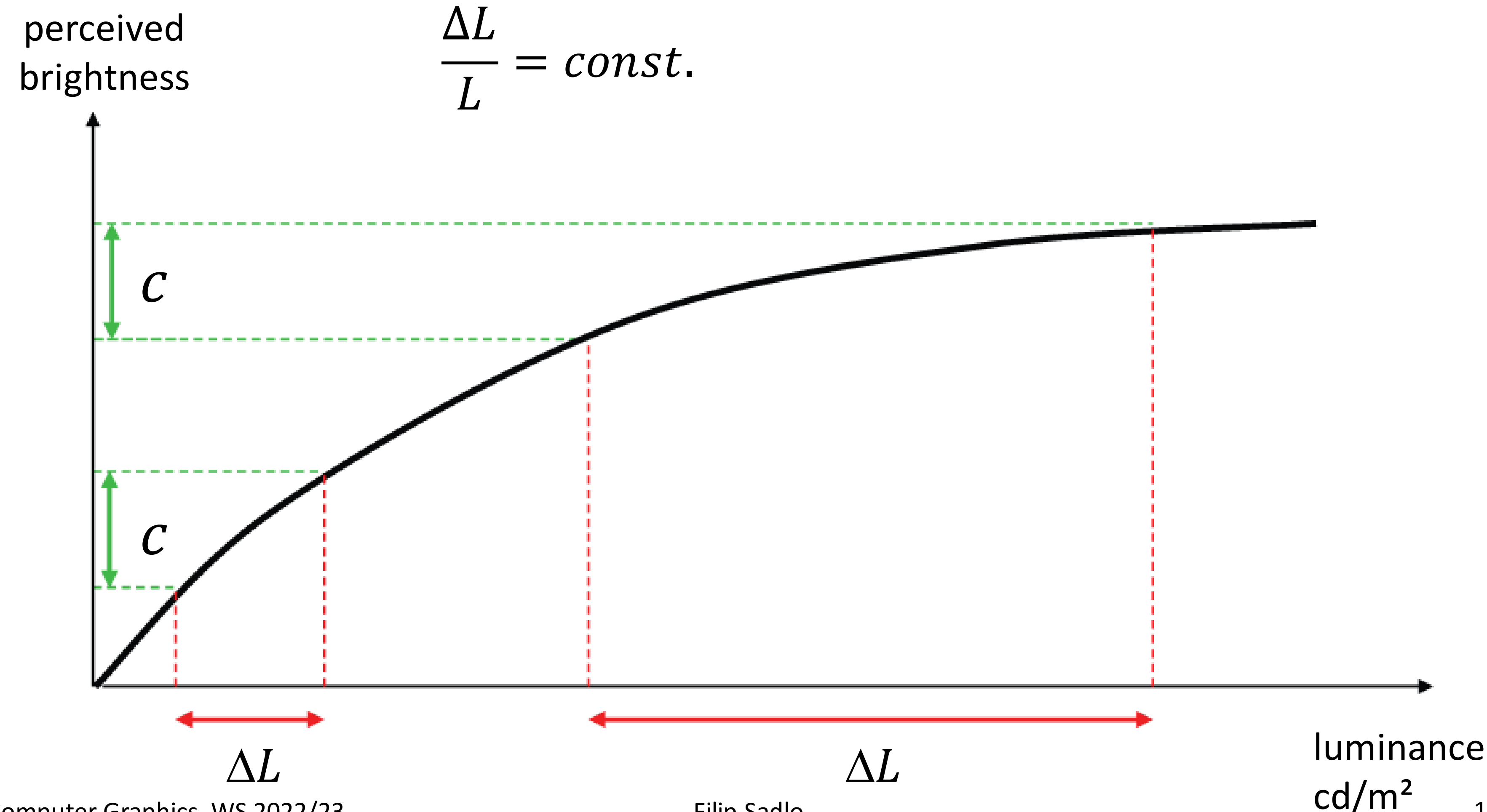
- Root of all “evil”: CRT monitors
  - Number of emitted photons from fluorescent coating increases (nonlinearly) with a power of applied voltage
  - But voltage is linear with the values in framebuffer
  - Today’s monitors, e.g., LCDs, emulate this behavior!
- Relation intensity—voltage resp. pixel values
  - $\mathrm{I} \sim \mathrm{V}^{\gamma} \quad (\gamma \approx 2.3)$
    I = intensity
    V = voltage
    γ = gamma value
- Gamma correction: adapt intensity for driving display
  - In software
  - In hardware (graphics card drivers)
  - Caution: there are image formats that already store values for a certain γ



Weber–Fechner Law

- The *subjectively perceived intensity* of sensations is *proportional to the logarithm of the intensity* of the physical stimulus
  - Proportion 1:2 is perceived as contrast identical to proportion 100:200

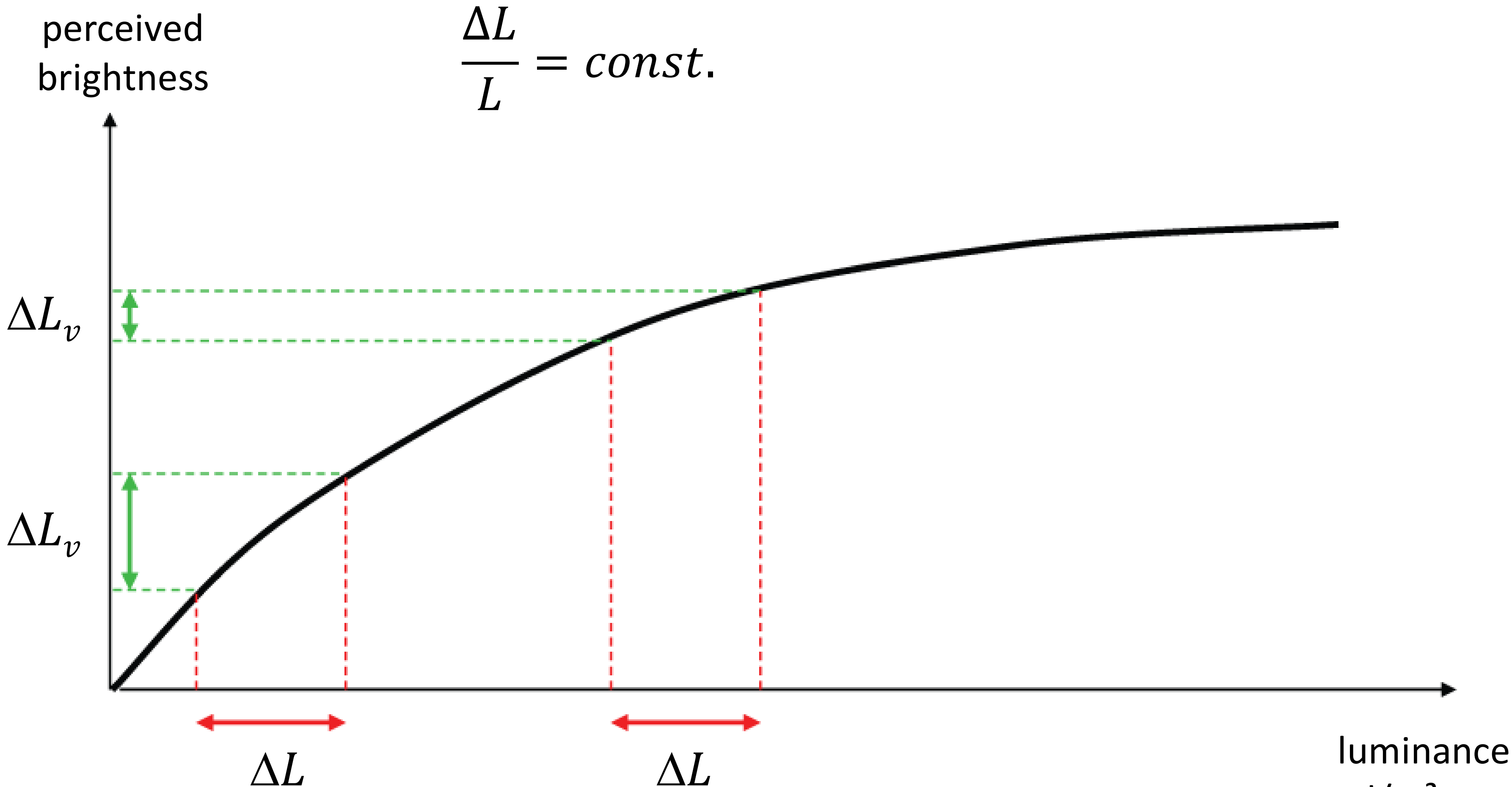
$$\frac{\Delta L}{L} = const.$$



Weber–Fechner Law

- The *subjectively perceived intensity* of sensations is *proportional to the logarithm of the intensity* of the physical stimulus
  - Equal differences in luminance are perceived logarithmically

$$\frac{\Delta L}{L} = const.$$



Weber–Fechner Law

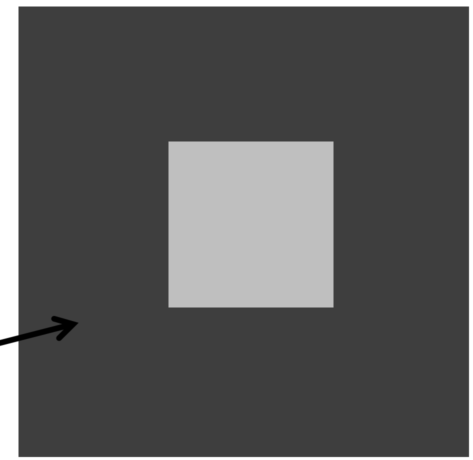
- The *subjectively perceived intensity* of sensations is *proportional to the logarithm of the intensity* of the physical stimulus

- Just noticeable brightness difference

Just noticeable difference (JND)

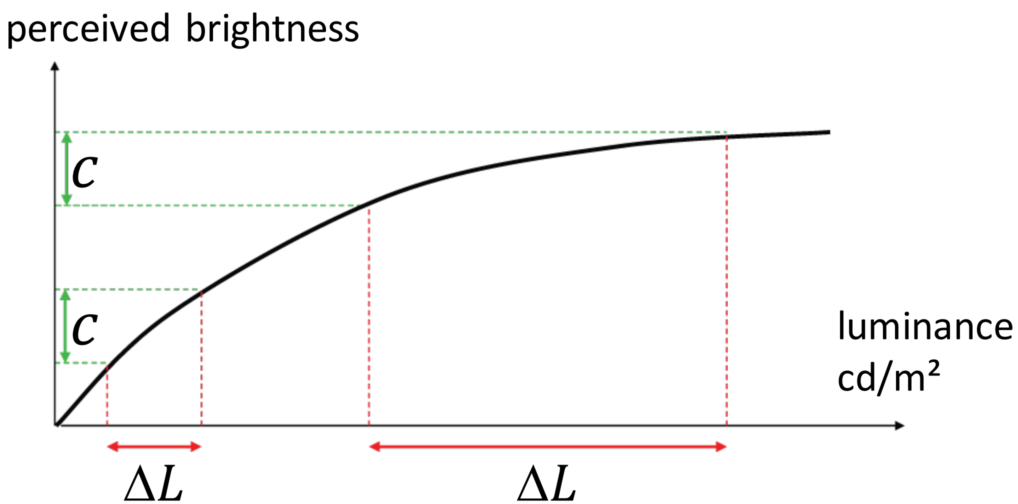
$$\frac{\Delta L}{L} = const. \approx 1\% \text{ to } 2\%$$

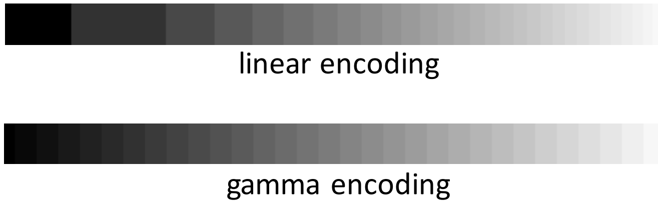
background brightness



Gamma-Compressed Images

- Human perception is based on relations, not on absolute brightness
  - "This gray is twice as bright as the other"
- For linear encoding
  - Many values between 128 and 255
  - Only few values between 1 and 2





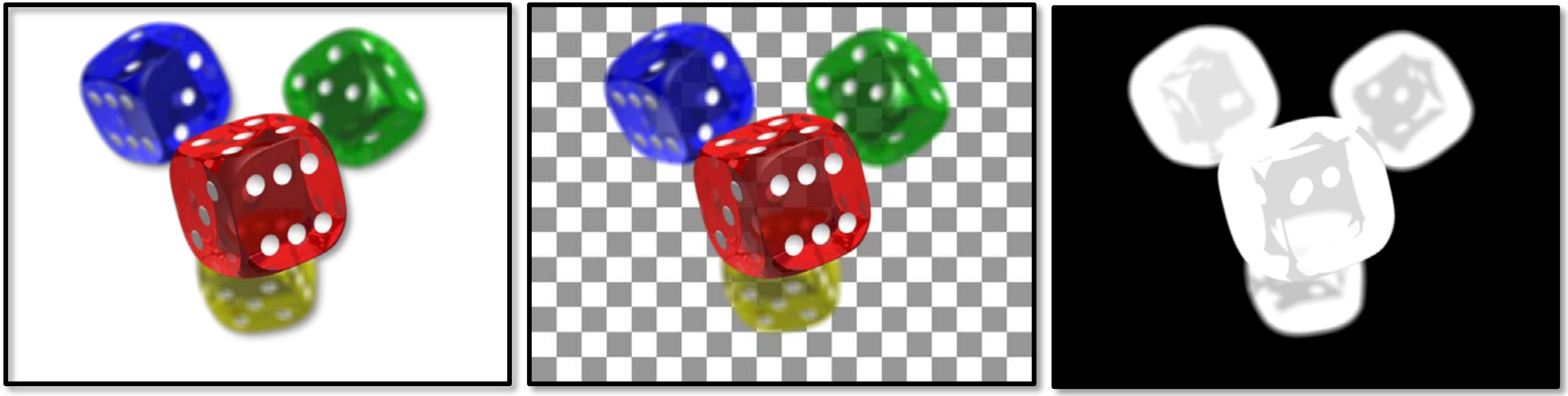
https://www.cambridgeincolour.com/tutorials/gamma-correction.htm

- Therefore, images sometimes stored with gamma compression ($I' \sim I^{1/\gamma}$, with $\gamma \approx 2$)
  - It is only coincidence that this approximately fits monitors

Alpha Channel

- Images are often encoded/stored with 32 bit/pixel
  - 24 bit color information and
  - 8 bit alpha channel (α = opacity, opposite of transparency)
  - So-called RGBA format
  - E.g., in framebuffer of the graphics card, textures, …

- Essential for image editing/manipulation, matting / blue-screen techniques, textures in computer graphics, …



http://de.wikipedia.org/wiki/Alphakanal

Raster Graphics and Image Processing

- Raster graphics (recap)
  - Advantage for display of images: repeated display is independent of scene complexity (framebuffer)
  - Properties
    - Uniform, discrete sampling of a signal
    - A piecewise constant function (here: illustration in 1D)
  - “Aliasing” and Moiré effects, “jaggies” (staircase effects)

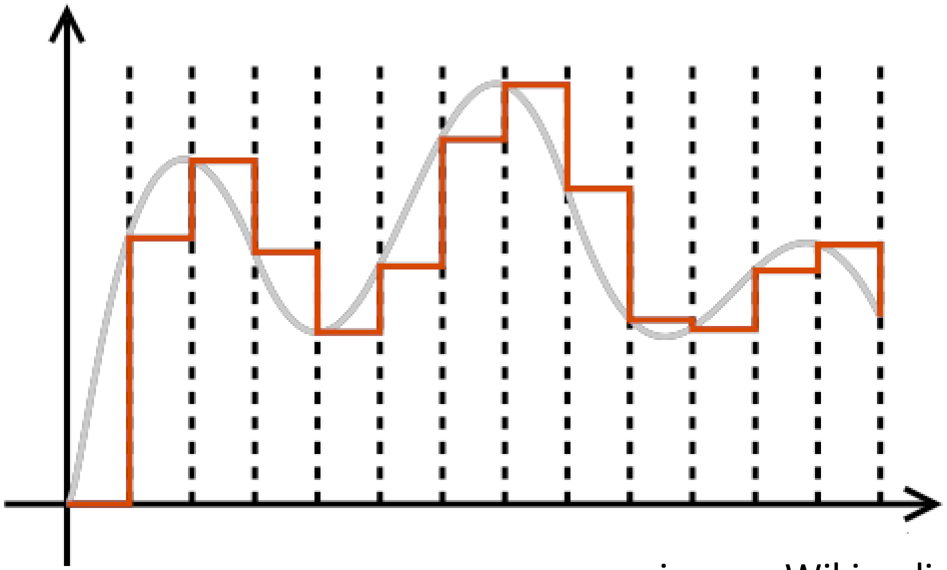
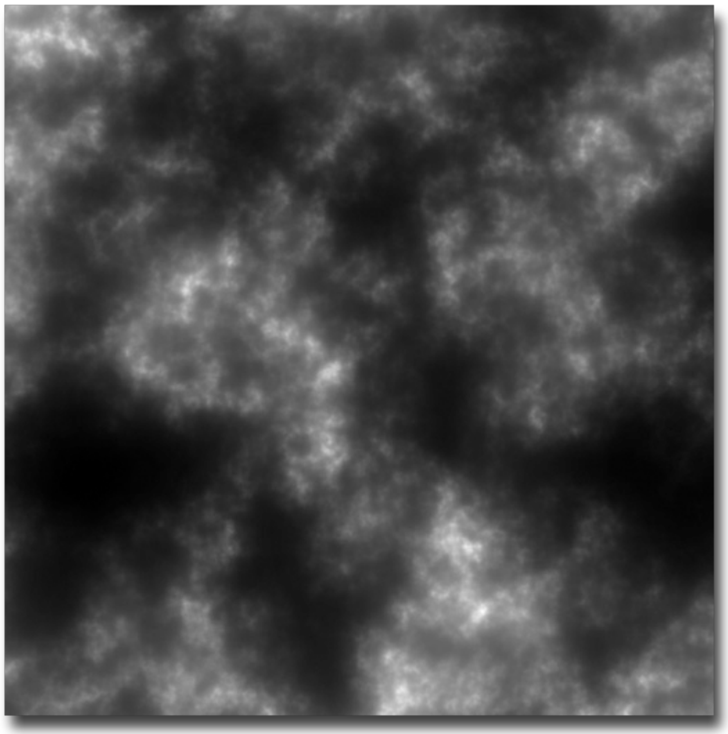
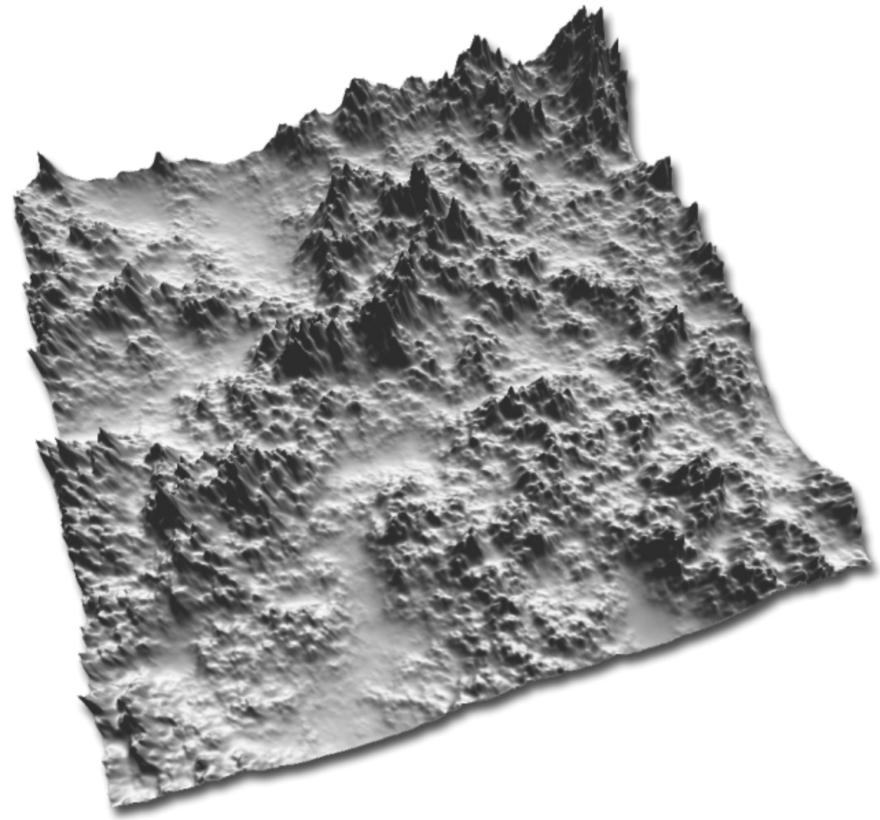


image: Wikipedia

Image ↔ Signal



2D height field

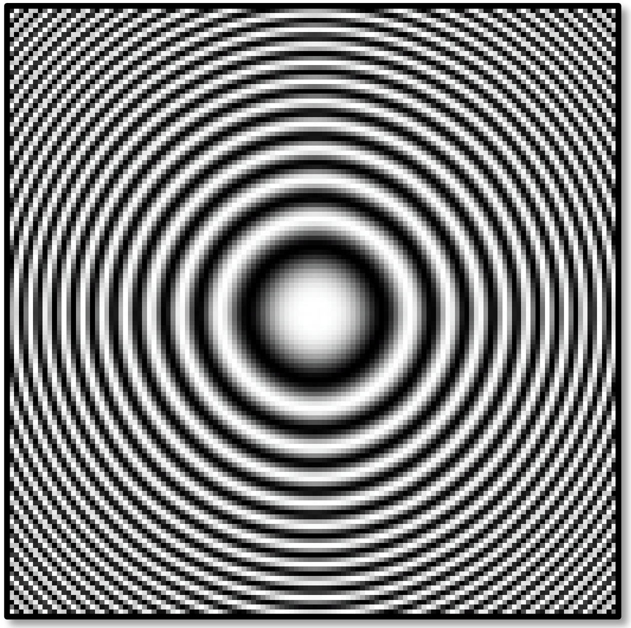


representation as surface in 3D

Image, Signal, and Sampling

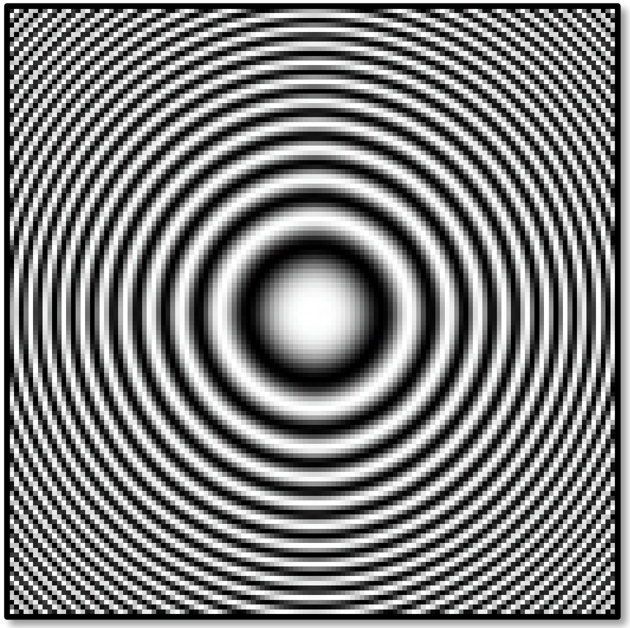
- Gray-level image of $f(d) = \cos ad^2$ with $d = \|\mathbf{x}\|$, $\mathbf{x} \in [-1,1]^2$, $a \in \mathbb{R}$

```
#define WIDTH  128
#define HEIGHT 128

unsigned char img[ WIDTH * HEIGHT ];

for ( y = 0; y < HEIGHT; y++ ) {
  for ( x = 0; x < WIDTH; x++ ) {
    float dx = (float)( 2 * x – WIDTH  ) / (float) WIDTH;
    float dy = (float)( 2 * y – HEIGHT ) / (float) HEIGHT;
    float d2 = dx * dx + dy * dy;

    // f(d) with a = 64
    float f = cosf( 64.0f * d2 );

    // mapping of f(d) to [0..255]
    img[ x + y * WIDTH ] =
        ( f + 1.0f ) * 127.5f;
} }
```

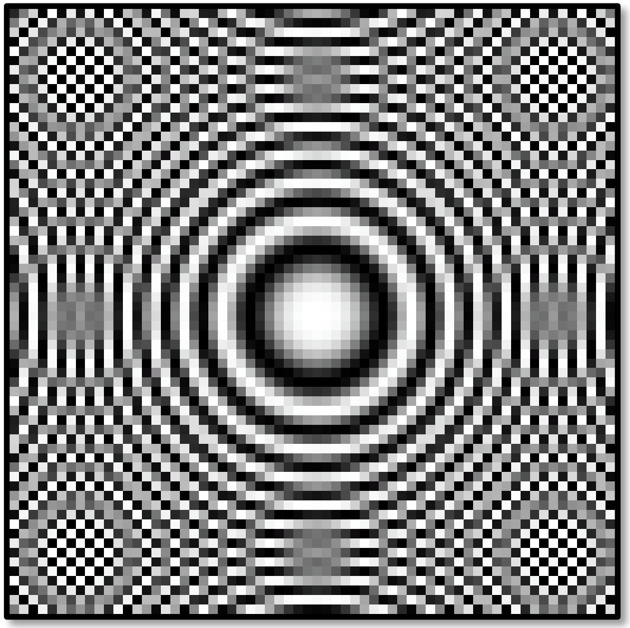


Image, Signal, and Sampling

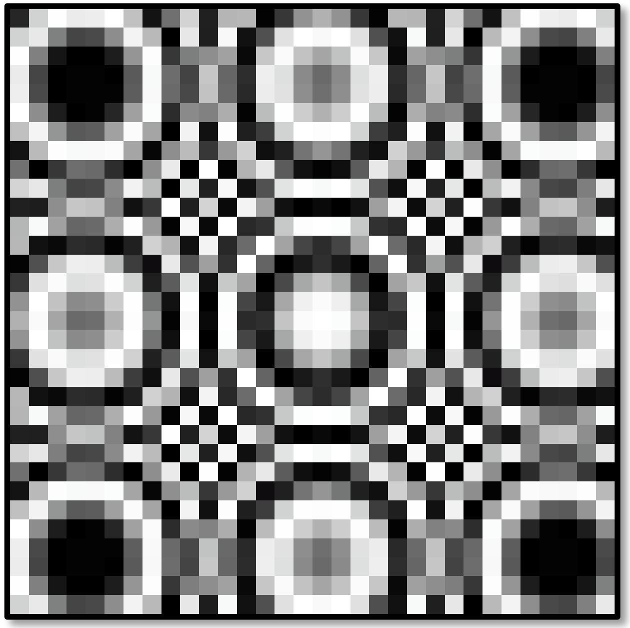
- Gray-level image of $f(d) = \cos ad^2$ with $d = \|\mathbf{x}\|$, $\mathbf{x} \in [-1,1]^2$, $a \in \mathbb{R}$



WIDTH=HEIGHT=128



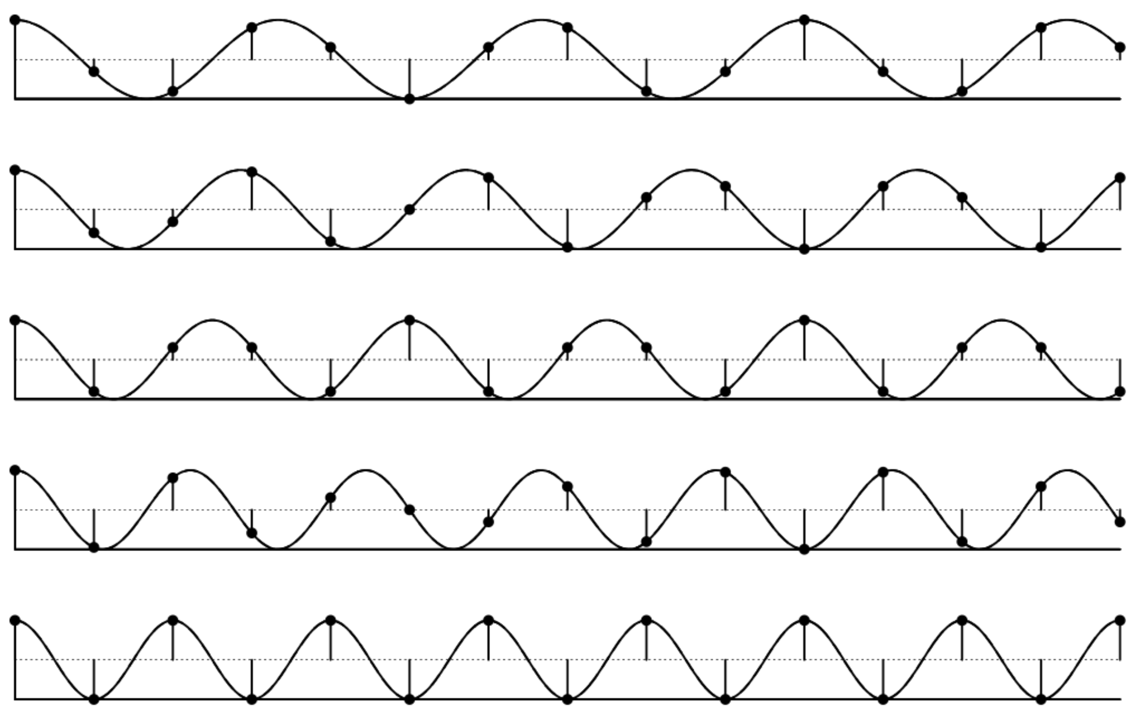
WIDTH=HEIGHT=64



WIDTH=HEIGHT=32

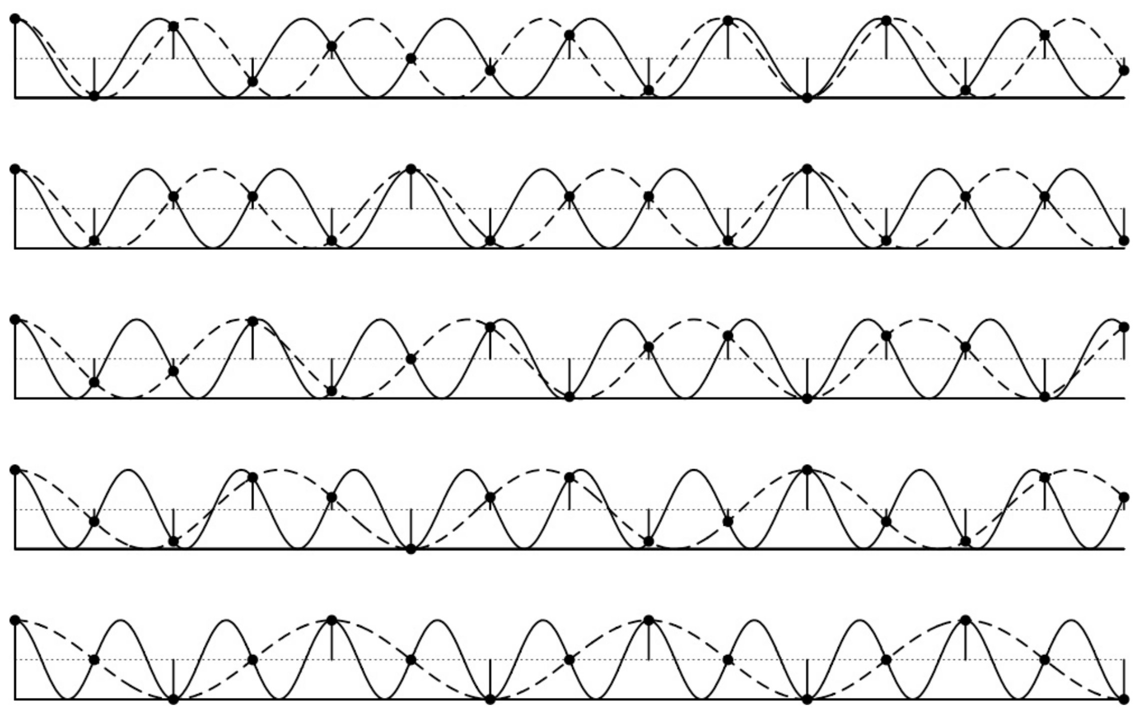
Image, Signal, and Sampling

- Nyquist–Shannon sampling theorem:
  A continuous, band-limited signal with a max. frequency $f_{\max}$ has to be sampled at a frequency larger than $2f_{\max}$ to be able to exactly reconstruct the original signal from the discretized signal
- Example: $f_{\mathrm{sampl}} > 2f_{\max}$, reconstructed signal identical to original



Image, Signal, and Sampling

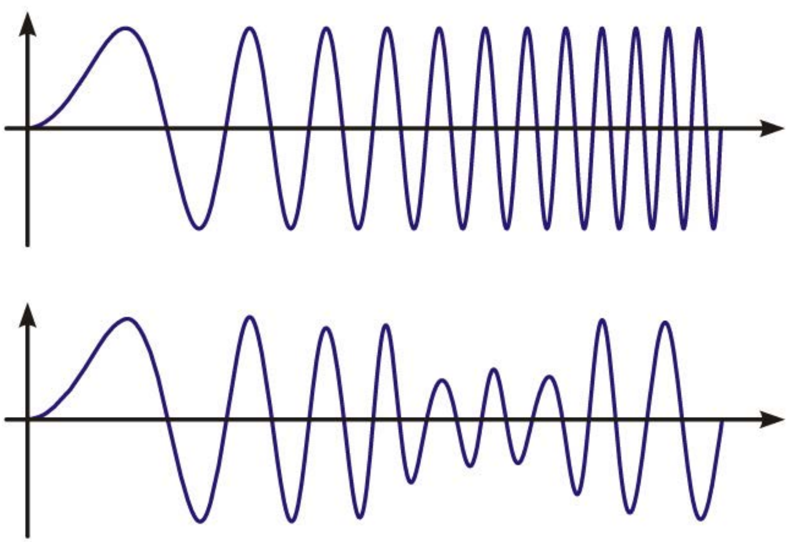
- Nyquist–Shannon sampling theorem:
  A continuous, band-limited signal with a max. frequency $f_{\max}$ has to be sampled at a frequency larger than $2f_{\max}$ to be able to exactly reconstruct the original signal from the discretized signal
- Example: $f_{\mathrm{sampl}} \leq 2f_{\max}$ —— orig. signal -------- reconstr. signal

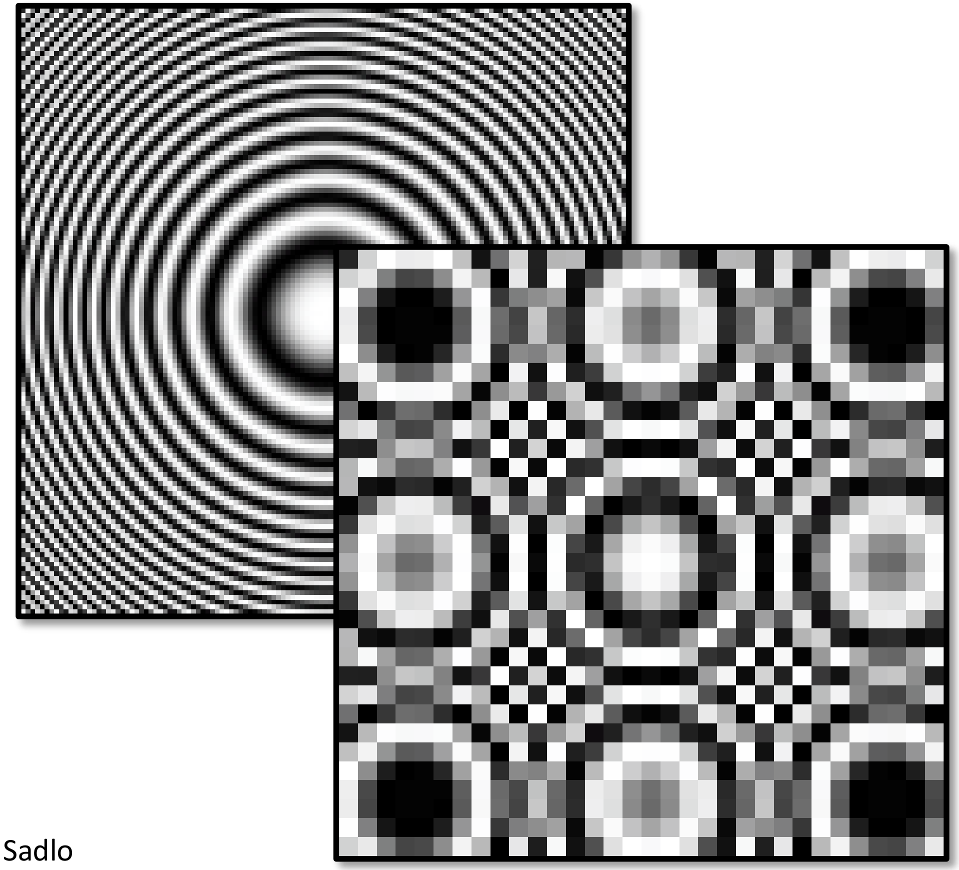


Image, Signal, and Sampling

- Aliasing effect: error introduced by sampling of signals
  - There are frequencies in reconstructed signal (i.e., in representation on monitor) that are not contained in original signal

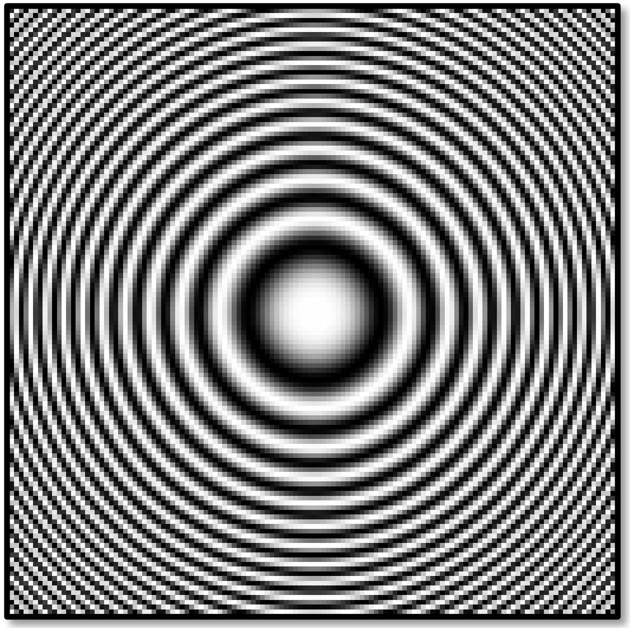
- 1D and 2D version of our example: original and reconstruction



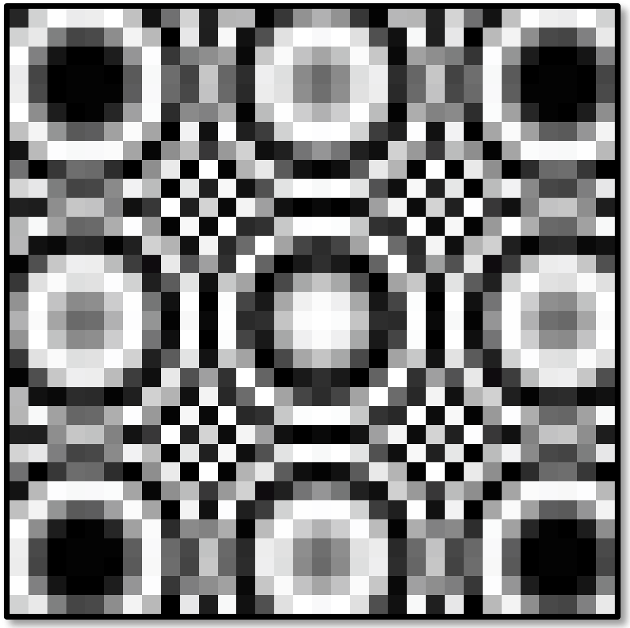


Image, Signal, and Sampling

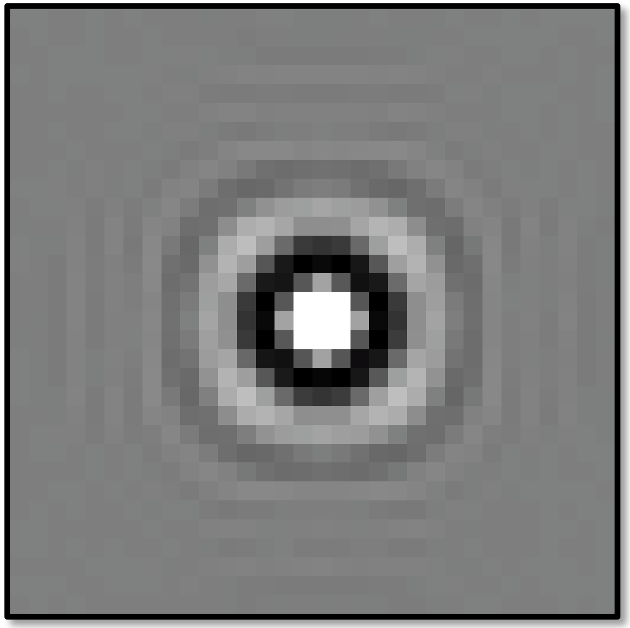
- Aliasing effect: error introduced by sampling of signals
  - There are frequencies in reconstructed signal (i.e., in representation on monitor) that are not contained in original signal
- Solution approaches:
  - Filtering of signal prior to sampling (“removal” of high frequencies):
    - In general not possible!
  - Higher sampling of signal, together with averaging (right image)



sufficient sampling



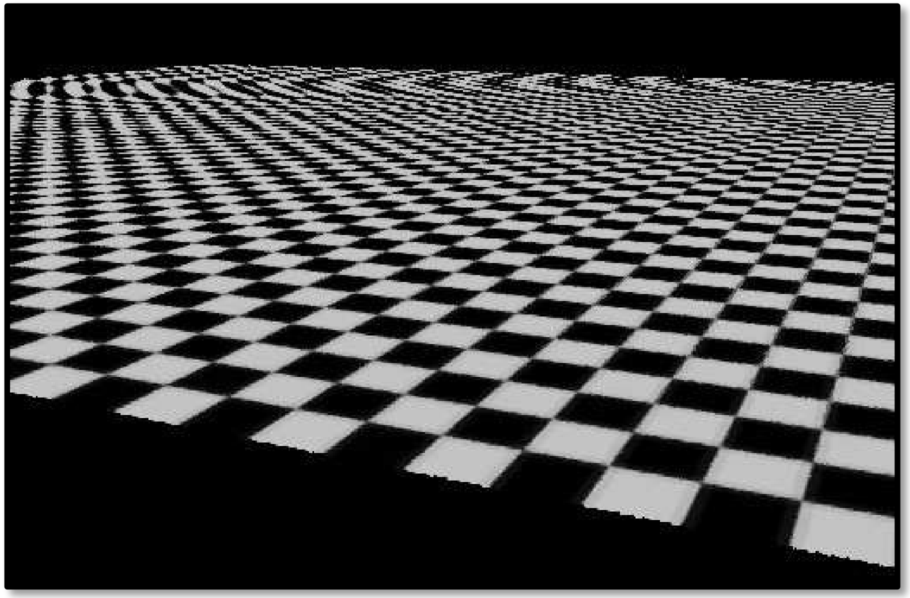
sampling too low → aliasing

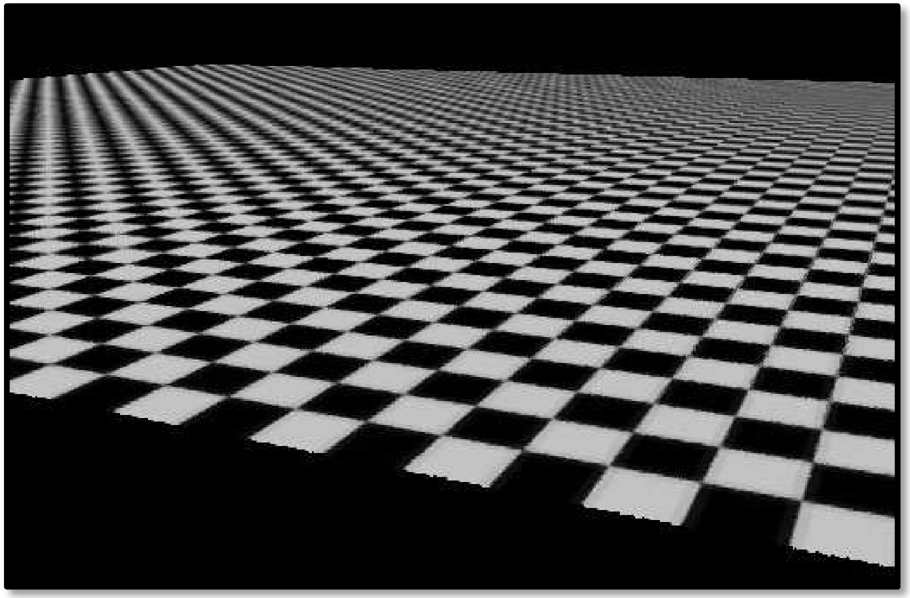


antialiasing by oversampling

Aliasing Example

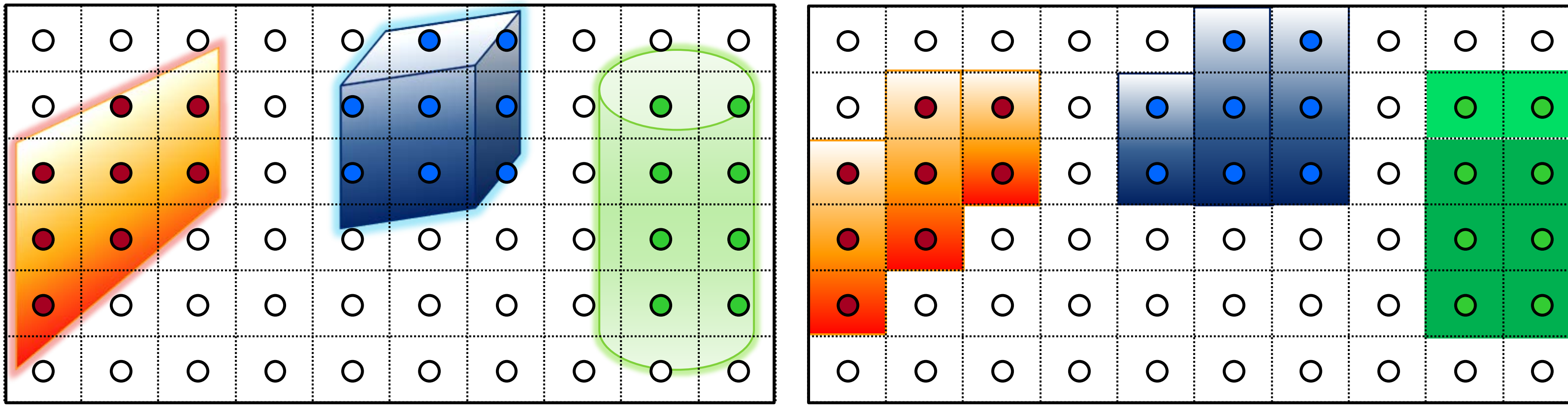
- … the classical example for aliasing in computer graphics

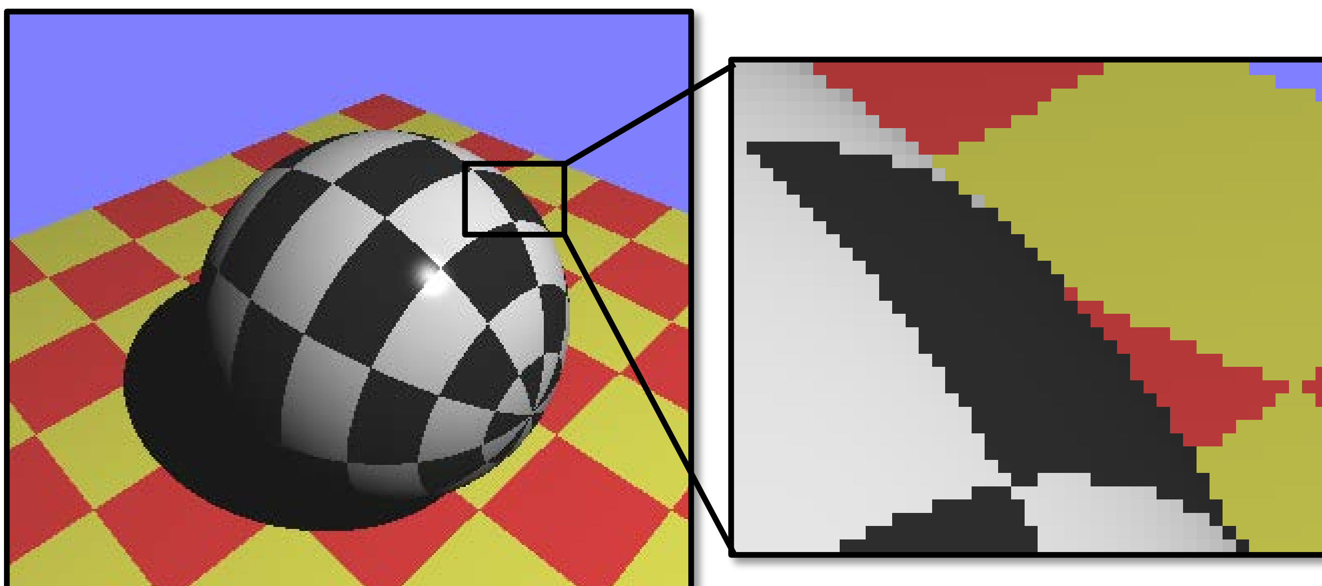




Jaggies

- A type of aliasing that is frequent in computer graphics





Jaggies

- "Edge smoothing"
  - In general by (smart) oversampling (see later)

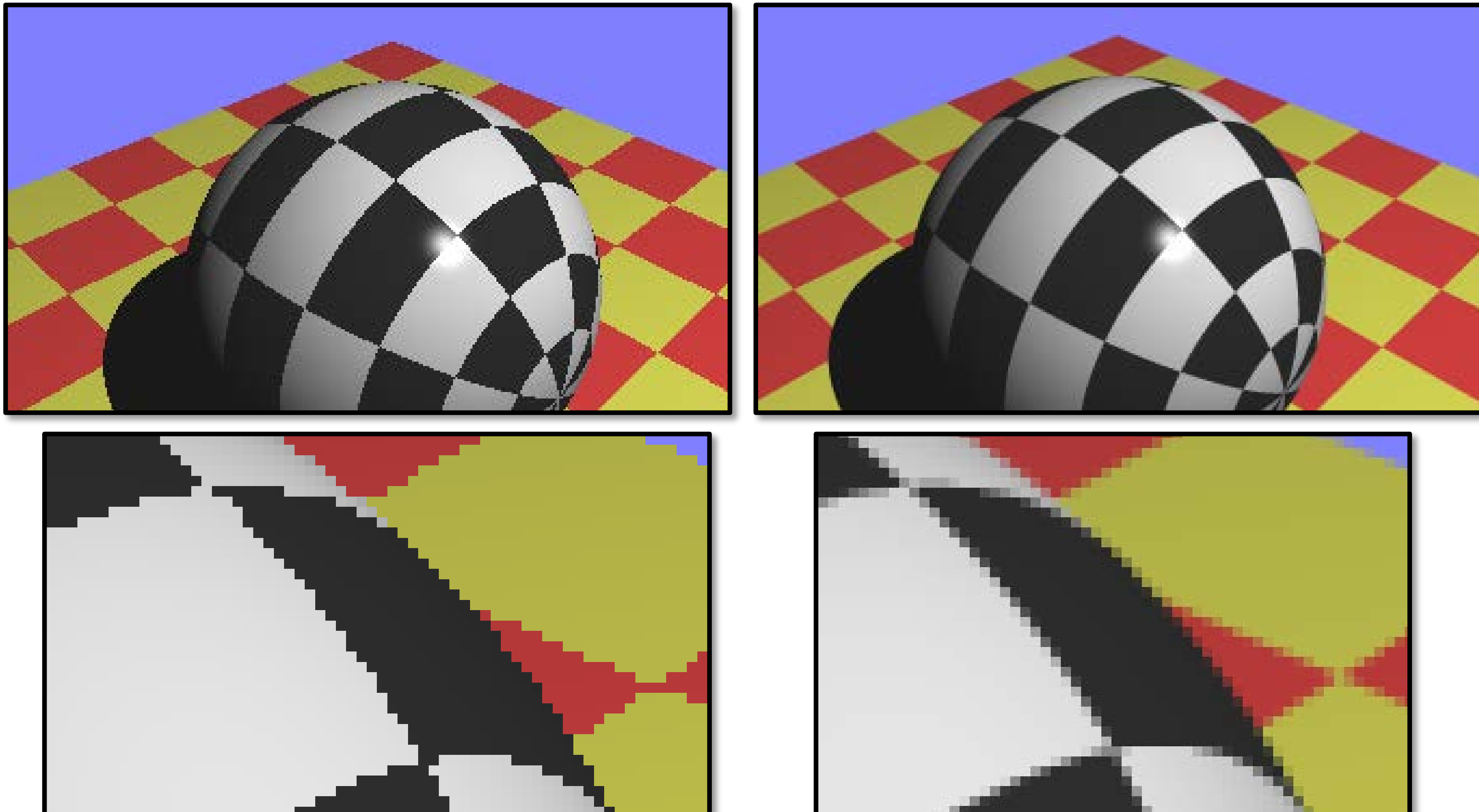


Image operations in color space

- Modification of color of a pixel based on current color at this position
  - No consideration of other (neighboring) pixels

- Composition of functions
  - Image function $f\colon \mathbb{R}^2 \rightarrow \mathbf{C}$ assigns to a position $\mathbf{u} \in \mathbb{R}^2$ in the image a color: $f(\mathbf{u}) \in \mathbf{C}$
  - Image operation is a function $g\colon \mathbf{C} \rightarrow \mathbf{C}'$
    - Example: modification of brightness
    - $\mathbf{C}'$ may, but does not need to, be same color space as $\mathbf{C}$
  - The color at position $\mathbf{u}$ is then $g\big(f(\mathbf{u})\big)$
  - Or: the image is now described by the function $g \circ f$

Example: Modification of Brightness

- Double brightness (here: gray-level image): $g(x) = 2x$

Example: Modification of Contrast

- Double contrast:

$$g(\mathbf{x}) = 2(\mathbf{x} - 0.5) + 0.5$$
$$g\big(f(\mathbf{u})\big) \in [-0.5; 1.5]^3$$

$$f(\mathbf{u}) \in [0; 1]^3$$

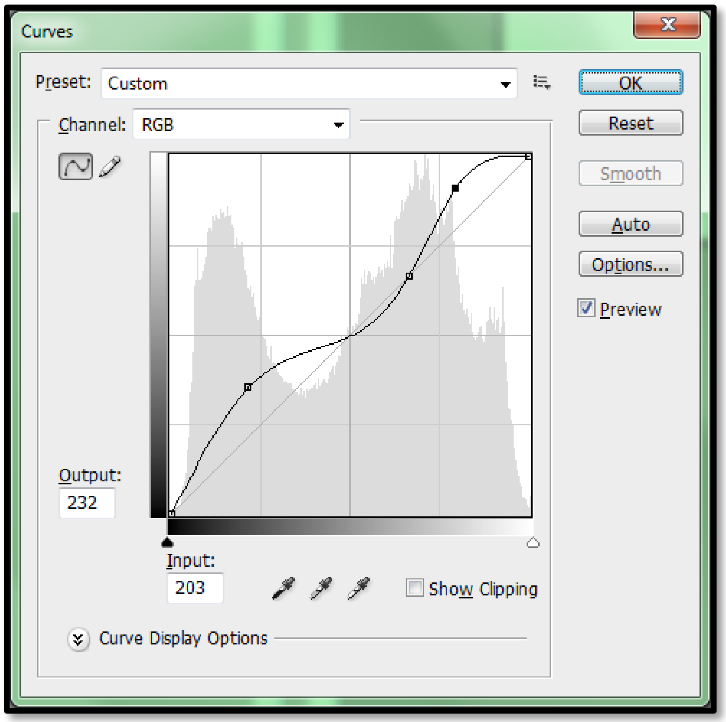
$$g(\mathbf{x}) = 0.5(\mathbf{x} - 0.5) + 0.5$$
$$g\big(f(\mathbf{u})\big) \in [0.25; 0.75]^3$$

Example: Desaturation

- Often-used mapping (recap): $Y = 0.3R + 0.59G + 0.11B$

Gradation Curves

- Visual manipulation of function g
- Precise control



Operations in Color Space and Image Space

- New color of a pixel depends on
  - its current color
  - the colors of its neighboring pixels
- Many image operations (so-called filters) can be expressed by means of convolution
  - Examples: blur, sharpening, edge detection, embossing, …

Convolution – Linear Filters

- Image function f assigns a color to each position
- A convolution computes the "overlap" of a function f with a second function g, which is shifted (mirrored about the y-axis) over f

$$[f * g](t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

- g is the filter function
- The convolution of two functions is again a function

Convolution

- Convolution of two box functions f and g
- Results in a triangle function: the area of the product of f and g for each translation

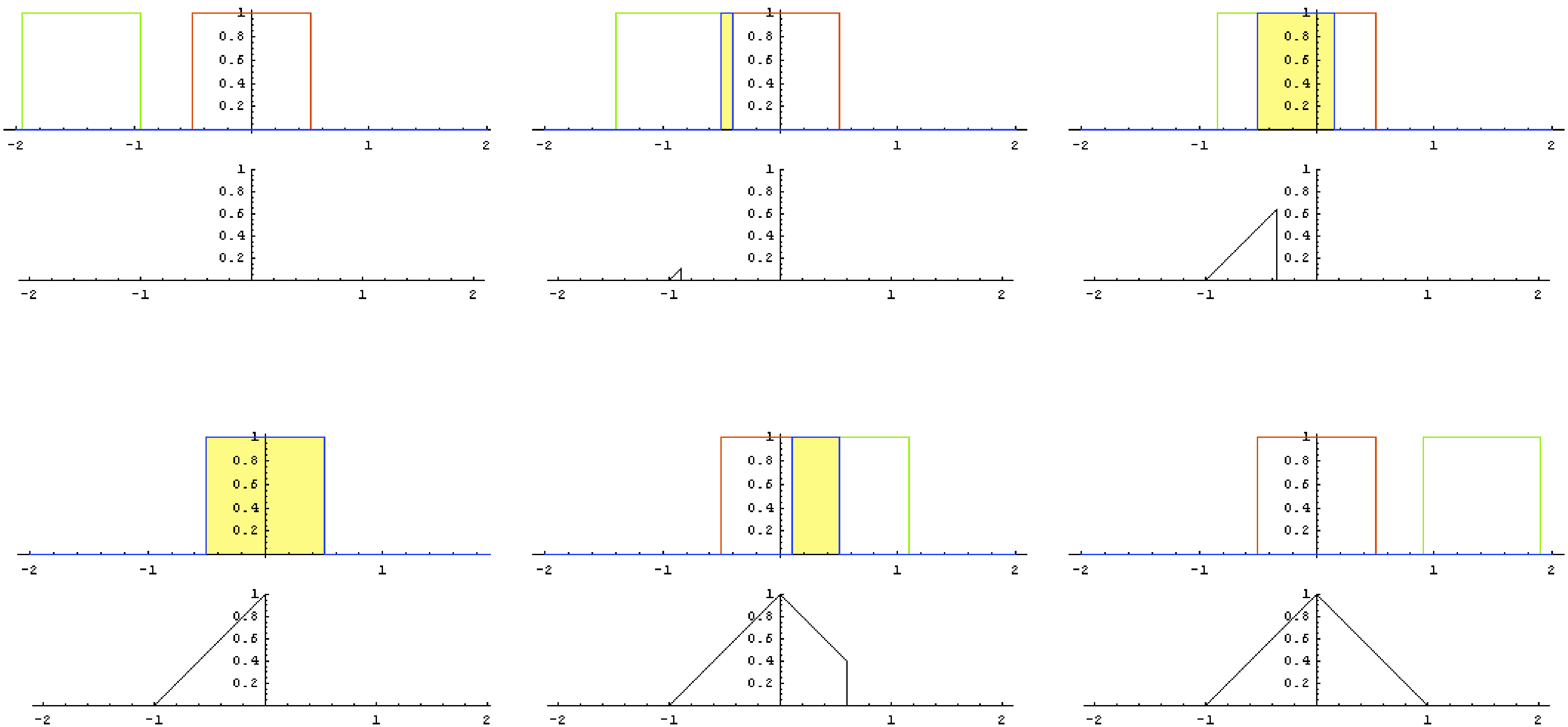


image: Wikipedia

Discrete Convolution

- If f and g are discrete (e.g., only defined at integer positions), then discrete convolution is

$$[f * g](n) = \sum_{i=-\infty}^{\infty} f(i)g(n-i)$$

- Principle
  - Filter function ("kernel") g is centered around n-th pixel
  - Weighting of each pixel according to the value of g at that position
  - Summation of the weighted color values results in color of n-th pixel

Convolution in 2D

- Continuous

$$[f * g](s,t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\sigma,\tau) g(s-\sigma, t-\tau) d\sigma d\tau$$

- Discrete

$$[f * g](m,n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i,j) g(m-i, n-j)$$

- Filters $g(\cdot,\cdot)$ are often nonzero only within a (small) interval

$$[f * g](m,n) = \sum_{i=m-a}^{m+a} \sum_{j=n-b}^{n+b} f(i,j) g(m-i, n-j)$$

with $-a \le m - i \le a$, and $-b \le n - j \le b$

Example: Discrete Convolution in 2D

```
#define WIDTH  128
#define HEIGHT 128

// an atypical floating-point frame buffer
float img[ WIDTH * HEIGHT ];

// gray-level image in floating-point repres. (0.0 = black, 1.0 = white)
float f[ WIDTH * HEIGHT ];

// discrete filter function with a=b=3
float g_table[ 7 * 7 ];
#define g( i, j ) g_table[ ((i)+3) + ((j)+3) * 7 ]
int a = 3, b = 3;

for ( n = 0; n < HEIGHT; n++ ) {
  for ( m = 0; m < WIDTH; m++ ) {

    float res = 0.0f;

    for ( int j = n - b; j <= n + b; j++ )
      for ( int i = m - a; i <= m + a; i++ )
        res += f[ i + j * WIDTH ] * g( m - i, n - j );

    img[ m + n * WIDTH ] = res;
}  }
```

Caution:
this example does not perform range checks!

Example: Discrete Convolution in 2D

- Assumption: all values outside of domain of f are 0 (other possible assumptions: periodic boundaries, repetition, …)
- In this example, the kernel matrix is symmetric, thus one cannot see that it is mirrored
  - Instead of mirroring the kernel, one could, of course, also adjust the indexing …

$$[f * g](m,n) = \sum_{i=m-a}^{m+a} \sum_{j=n-b}^{n+b} f(i,j)g(m-i,n-j)$$

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & 5 & 1 \\ 1 & 0 & 8 \end{bmatrix} * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 7 & 7 & 1 \\ -8 & 21 & -9 \\ 5 & -14 & 39 \end{bmatrix}$$

Example: Discrete Convolution in 2D

Example: Discrete Convolution in 2D

0	-1	0
2 -1	3 5	1 -1
0 0	5 -1	1 0
1	0	8

7	7	?
?	?	?
?	?	?

Example: Discrete Convolution in 2D

Example: Discrete Convolution in 2D

0	2 -1	3 0	1
-1	0 5	5 -1	1
0	1 -1	0 0	8

7	7	1
-8	?	?
?	?	?

Example: Discrete Convolution in 2D

2 / 0	3 / -1	1 / 0
0 / -1	5 / 5	1 / -1
1 / 0	0 / -1	8 / 0

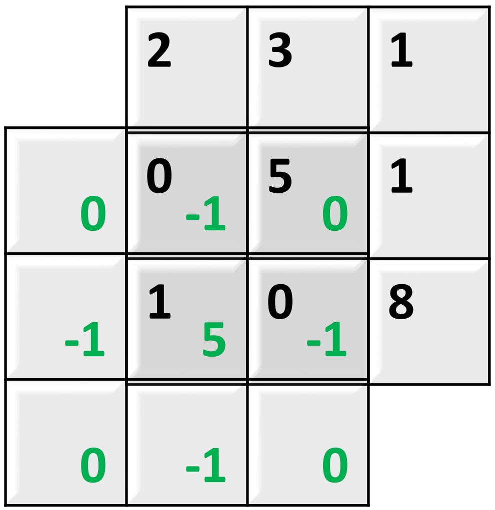
7	7	1
-8	21	?
?	?	?

Example: Discrete Convolution in 2D

2	3 0	1 -1	0
0	5 -1	1 5	-1
1	0 0	8 -1	0

7	7	1
-8	21	-9
?	?	?

Example: Discrete Convolution in 2D



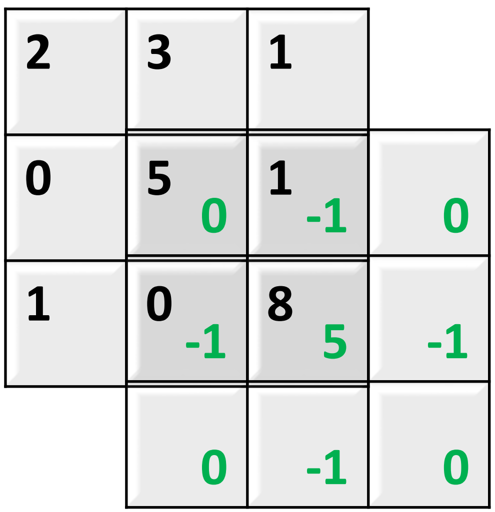
7	7	1
-8	21	-9
5	?	?

Example: Discrete Convolution in 2D

2	3	1
0 0	5 -1	1 0
1 -1	0 5	8 -1
0	-1	0

7	7	1
-8	21	-9
5	-14	?

Example: Discrete Convolution in 2D



7	7	1
-8	21	-9
5	-14	39

Example: Discrete Convolution in 2D

$$\begin{bmatrix} 2 & 3 & 1 \\ 0 & 5 & 1 \\ 1 & 0 & 8 \end{bmatrix} * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 7 & 7 & 1 \\ -8 & 21 & -9 \\ 5 & -14 & 39 \end{bmatrix}$$

Filter: Blur

- Normalized kernel: entries sum up to 1 → total brightness is preserved

$$* \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 9$$

Filter: Edge Detection

- "Negative Laplace operator"

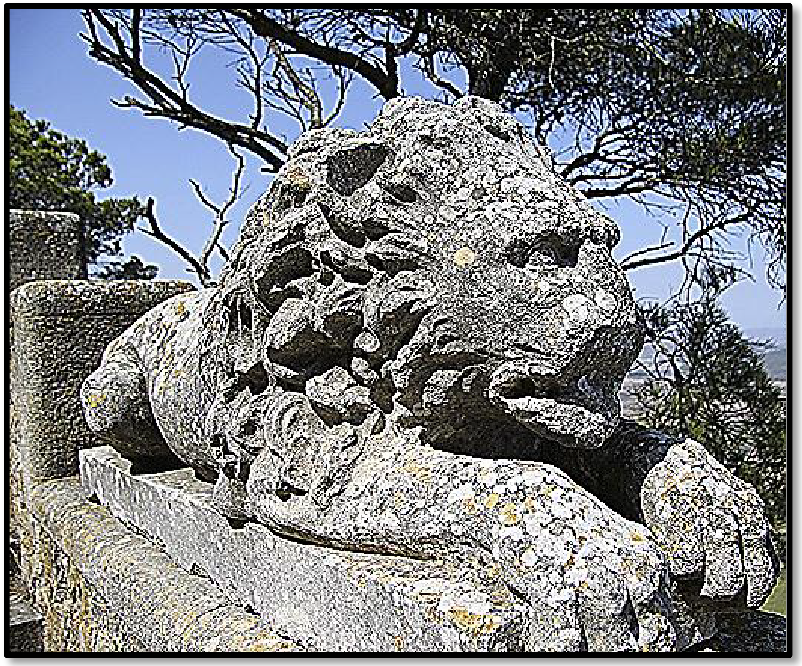
*

0	-1	0
-1	4	-1
0	-1	0

Filter: Sharpen

$*$

0	-1	0
-1	5	-1
0	-1	0



Filter: Emboss

$*$

-1	-1	0
-1	1	1
0	1	1

Morphological Operators

- Structurally changing operation (nonlinear filter)
- Originally defined on binary images A
- Consider set of neighboring pixels
  - Set of neighboring pixels: structural element B
  - Operation: set or delete pixels

B

0	1	0
1	1	1
0	1	0

- $(\hat{B})_z$ represents translation/centering of B around a pixel z
- **Dilation**

$$A \oplus B = \left\{ z \middle| (\hat{B})_z \cap A \neq \emptyset \right\}$$

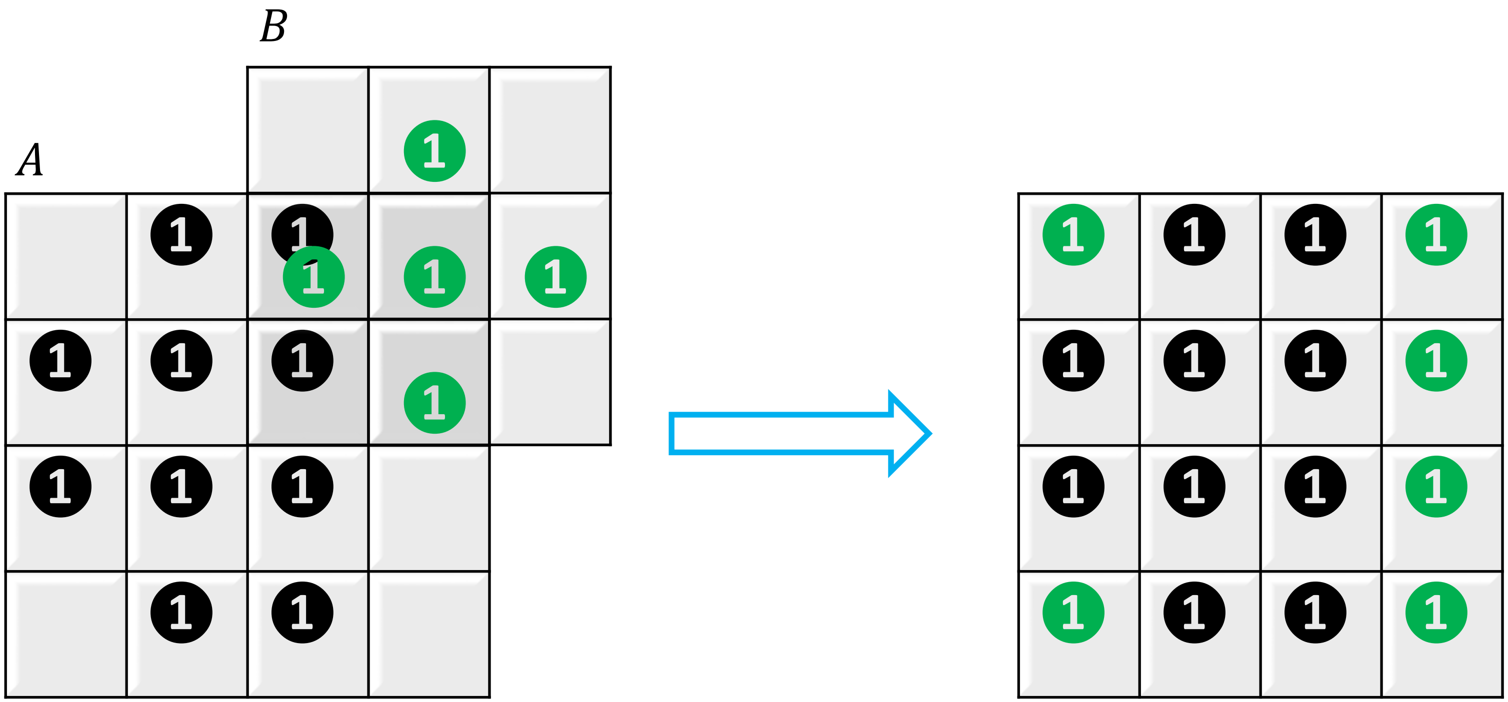
  - Set of all pixels z, at which a “1” in $(\hat{B})_z$ overlaps at least with one “1” in A

- **Erosion**

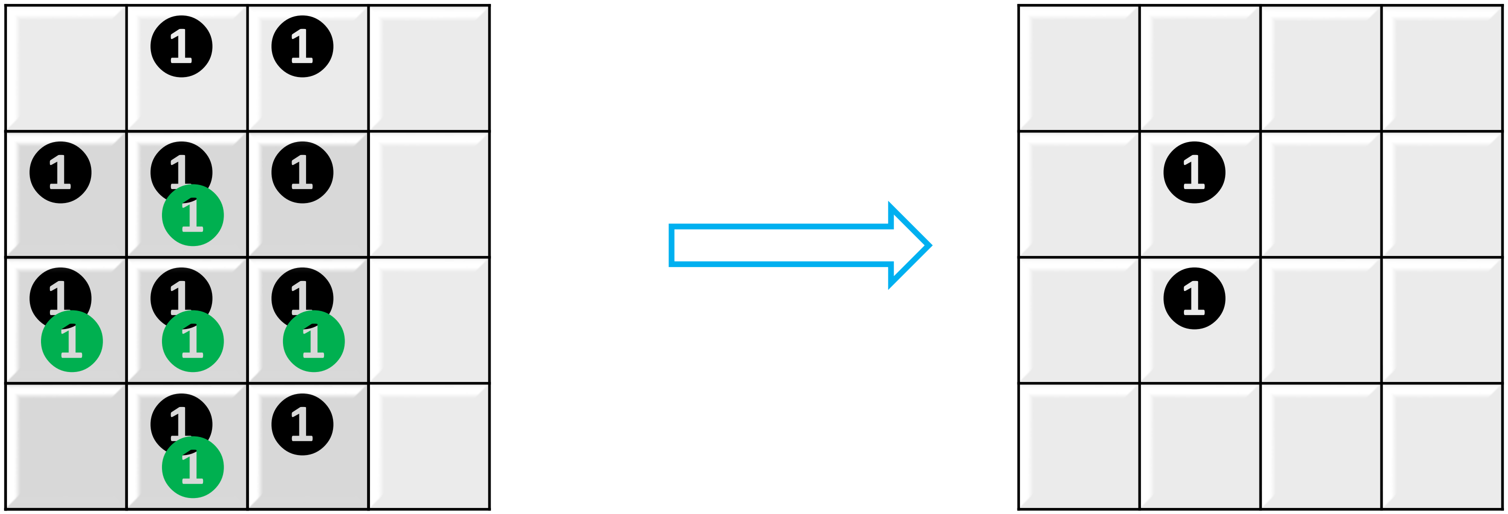
$$A \ominus B = \left\{ z \middle| (\hat{B})_z \subseteq A \right\}$$

  - Set of all pixels z, at which all “1” in $(\hat{B})_z$ overlap with a “1” in A
  - Or: if there is at least one “1” in $(\hat{B})_z$ that overlaps a “0” in A, then respective pixel is not set in resulting image

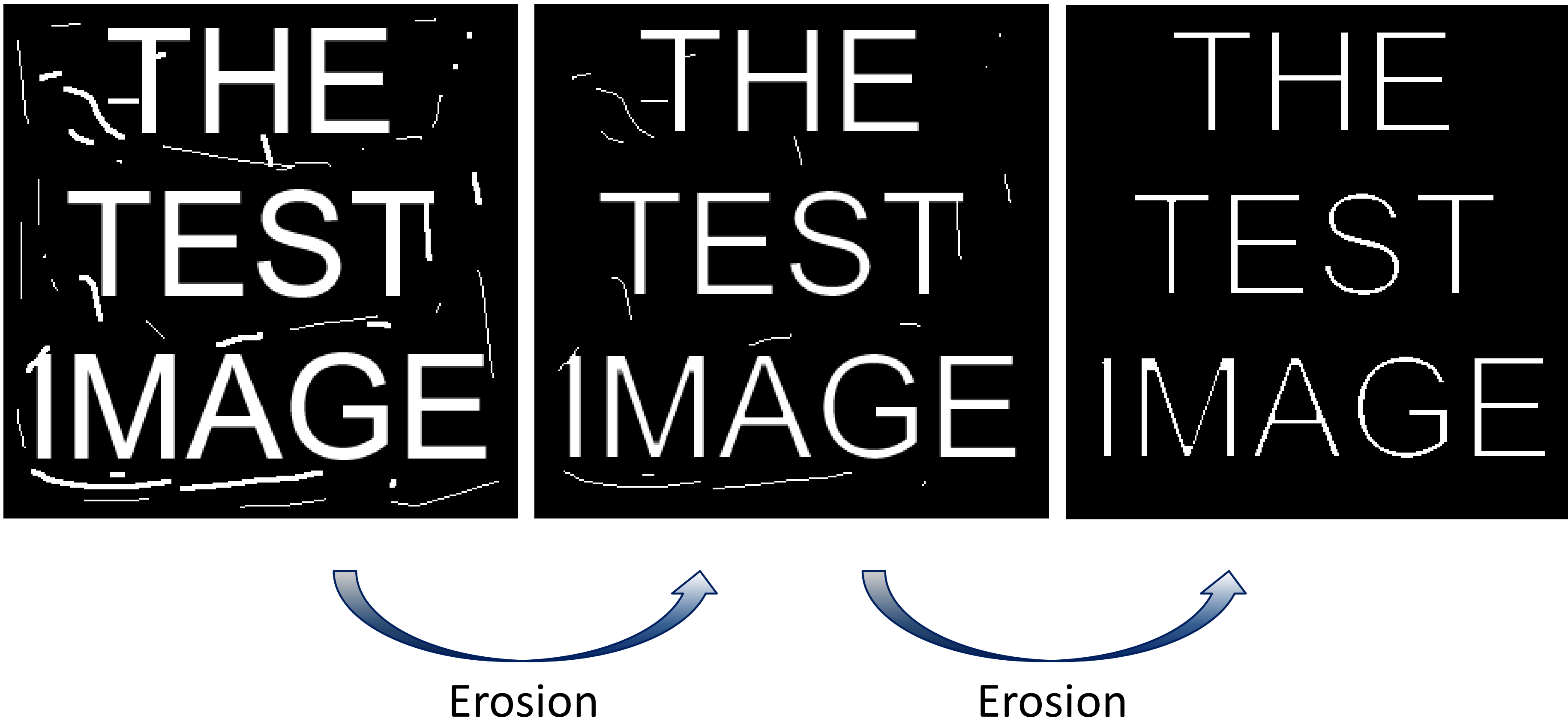
Dilation



Erosion



Erosion

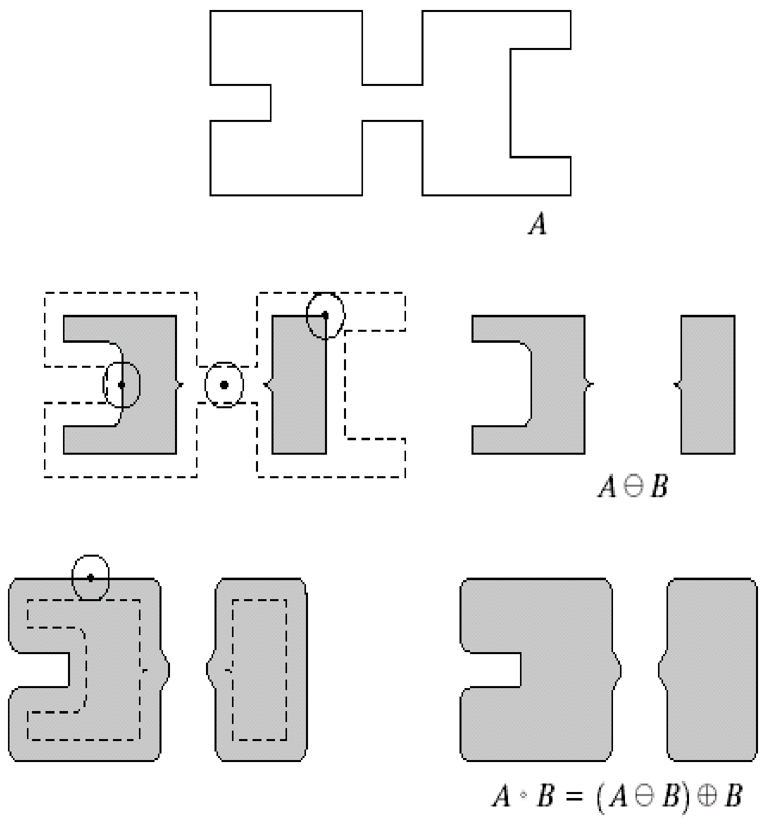

THE TEST IMAGE
THE TEST IMAGE
THE TEST IMAGE
Erosion
Erosion

Opening and Closing

- **Opening**: smooths parts of contour, removes protrusions and bridges

$$A \odot B = (A \ominus B) \oplus B$$

- Erosion, followed by dilation:



Opening and Closing

- **Closing**: smooths parts of contour, closes smaller gaps

$$A \odot B = (A \oplus B) \ominus B$$

- Dilation, followed by erosion: