Mini Lecture on

Image Reproduction

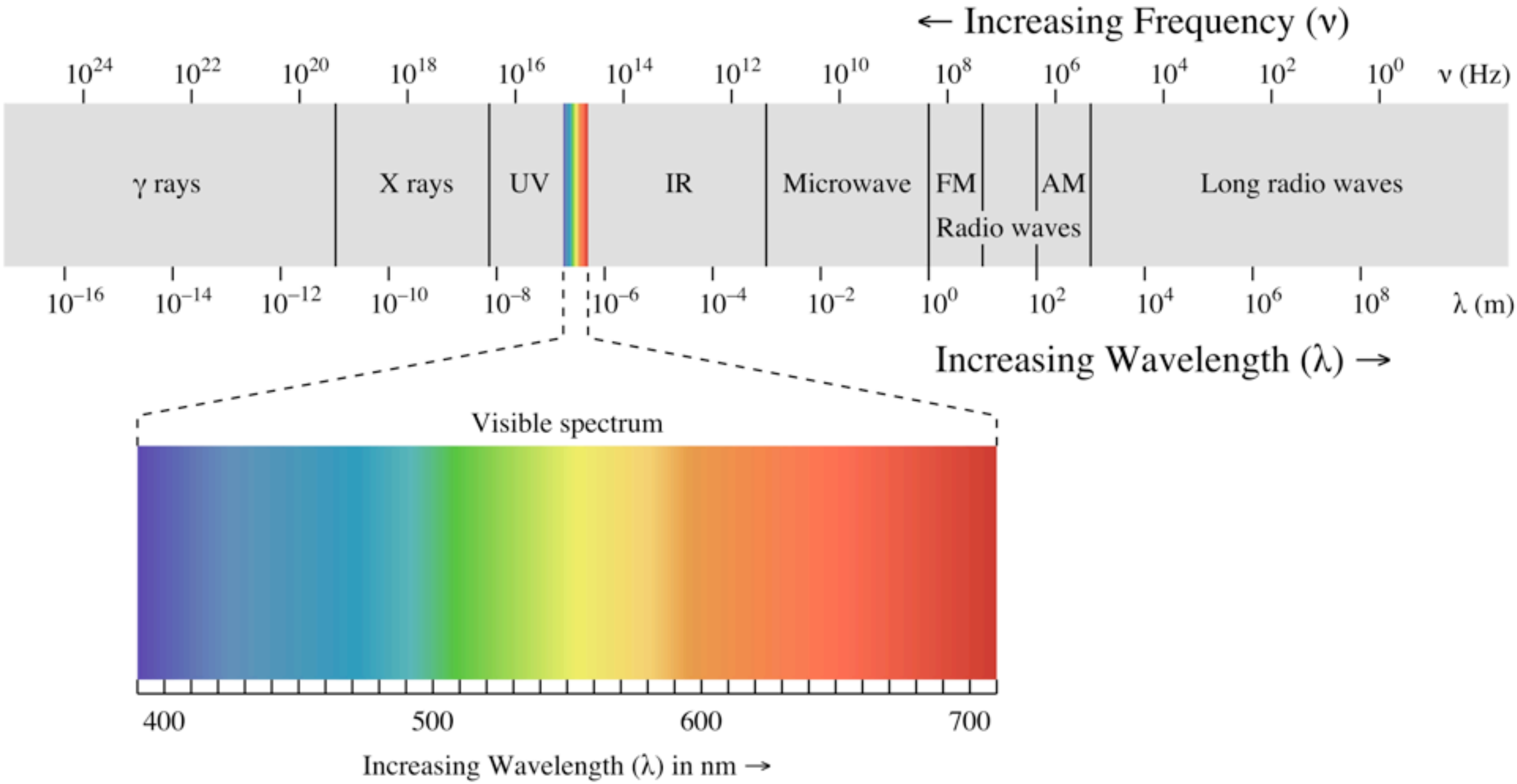
Images everywhere

- Interesting properties of the information age
  - Images are everywhere. Color works. It is all inexpensive.
  - Conversion and extension of image creation and manipulation of images.
    - It is all digital
  - Physical capture --> digital representation --> display
    - Intangible replaces the tangible

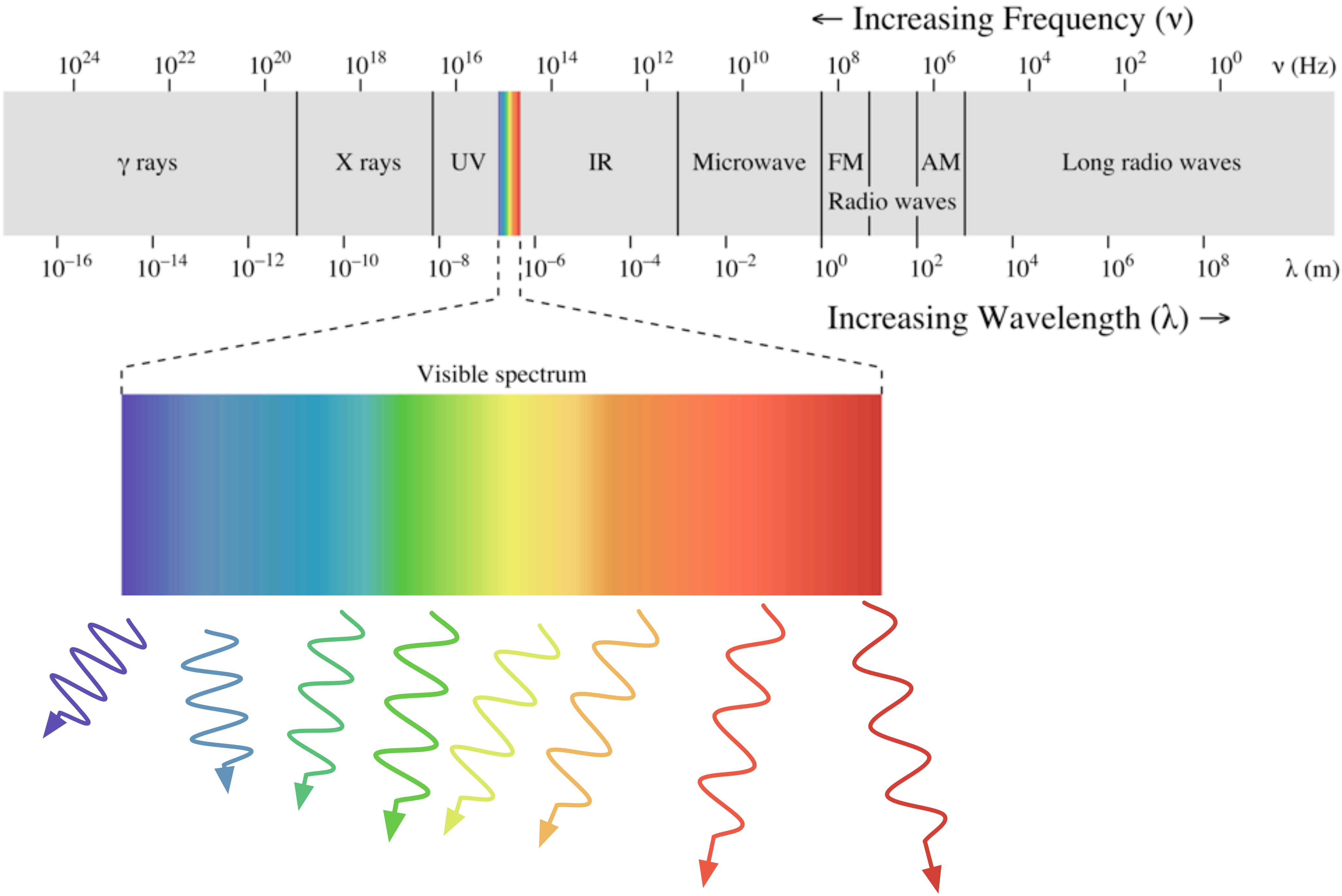
A model for light around us

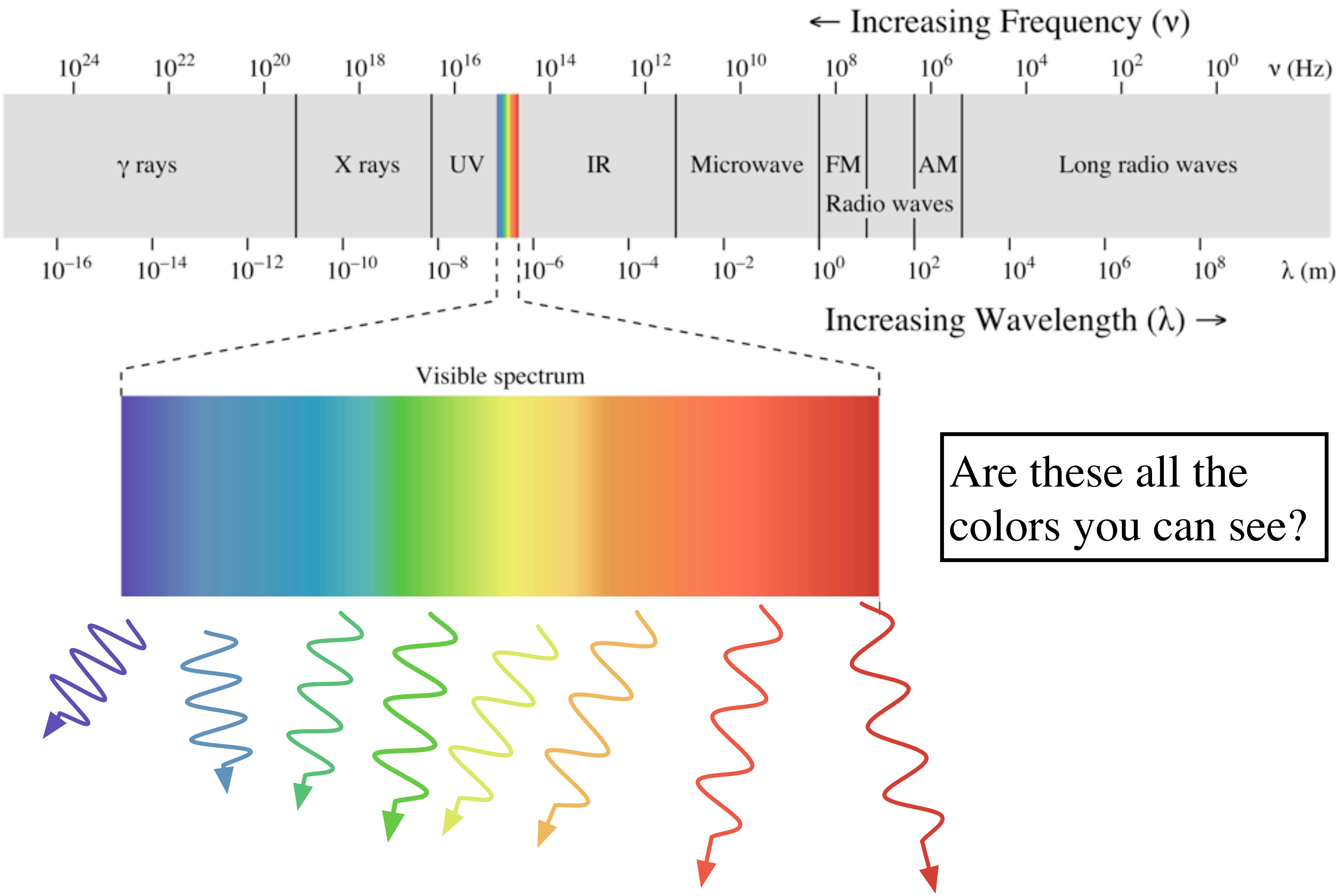
- The light around us is a mixture of **photons** of different **wavelengths**

- Key points of the model for our discussion
  - The signal is made of discrete bits (like rain drops)
    - You can imagine counting them
  - The bits have a characteristic number (wavelength)
    - This distinguishes light from the colors in a rainbow



(from WikiPedia commons)


← Increasing Frequency (ν)
10^24 10^22 10^20 10^18 10^16 10^14 10^12 10^10 10^8 10^6 10^4 10^2 10^0 ν (Hz)
γ rays
X rays
UV
IR
Microwave
FM
AM
Radio waves
Long radio waves
10^−16 10^−14 10^−12 10^−10 10^−8 10^−6 10^−4 10^−2 10^0 10^2 10^4 10^6 10^8 λ (m)
Increasing Wavelength (λ) →
Visible spectrum

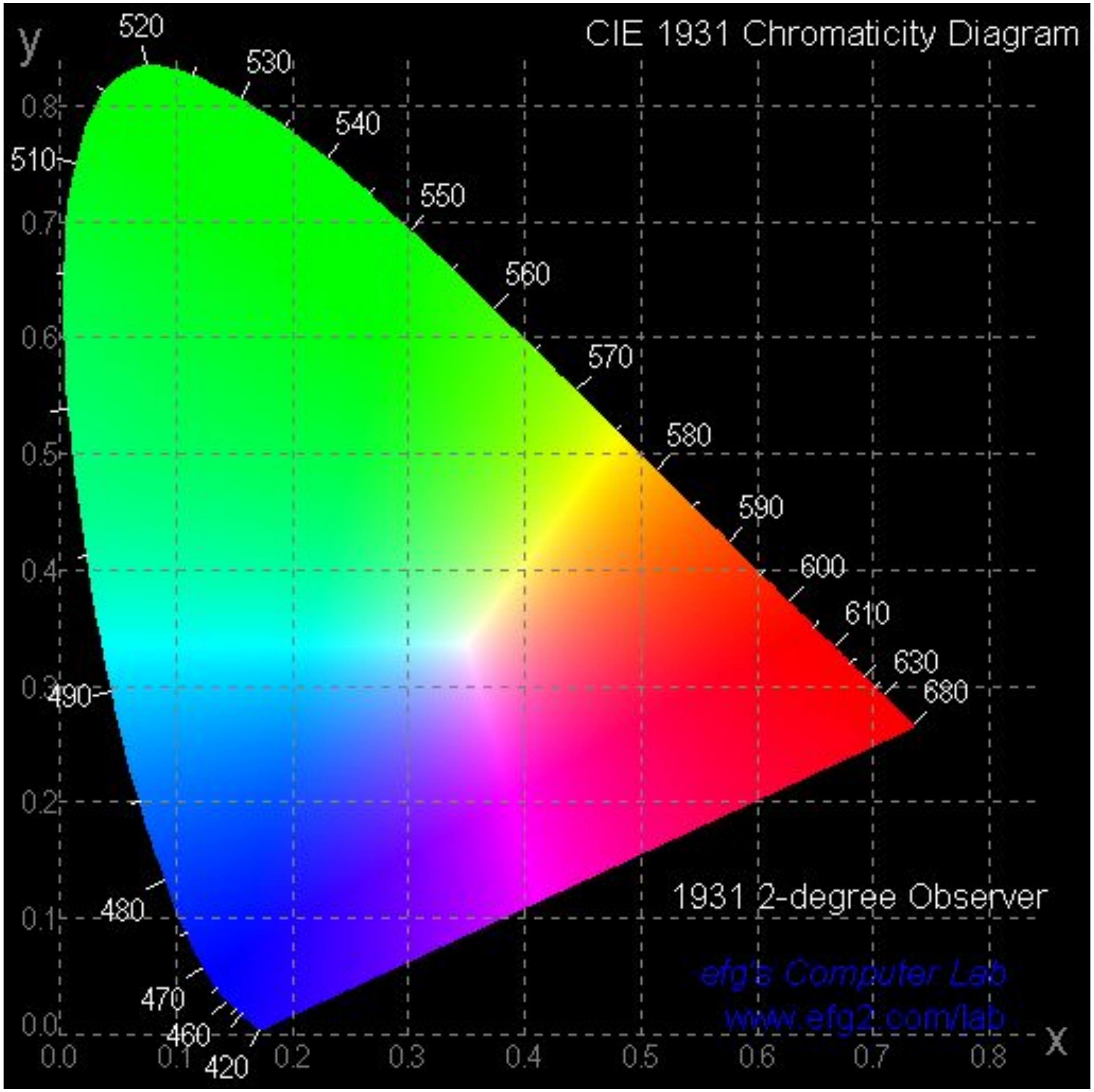


Are these all the colors you can see?

Colors we can see

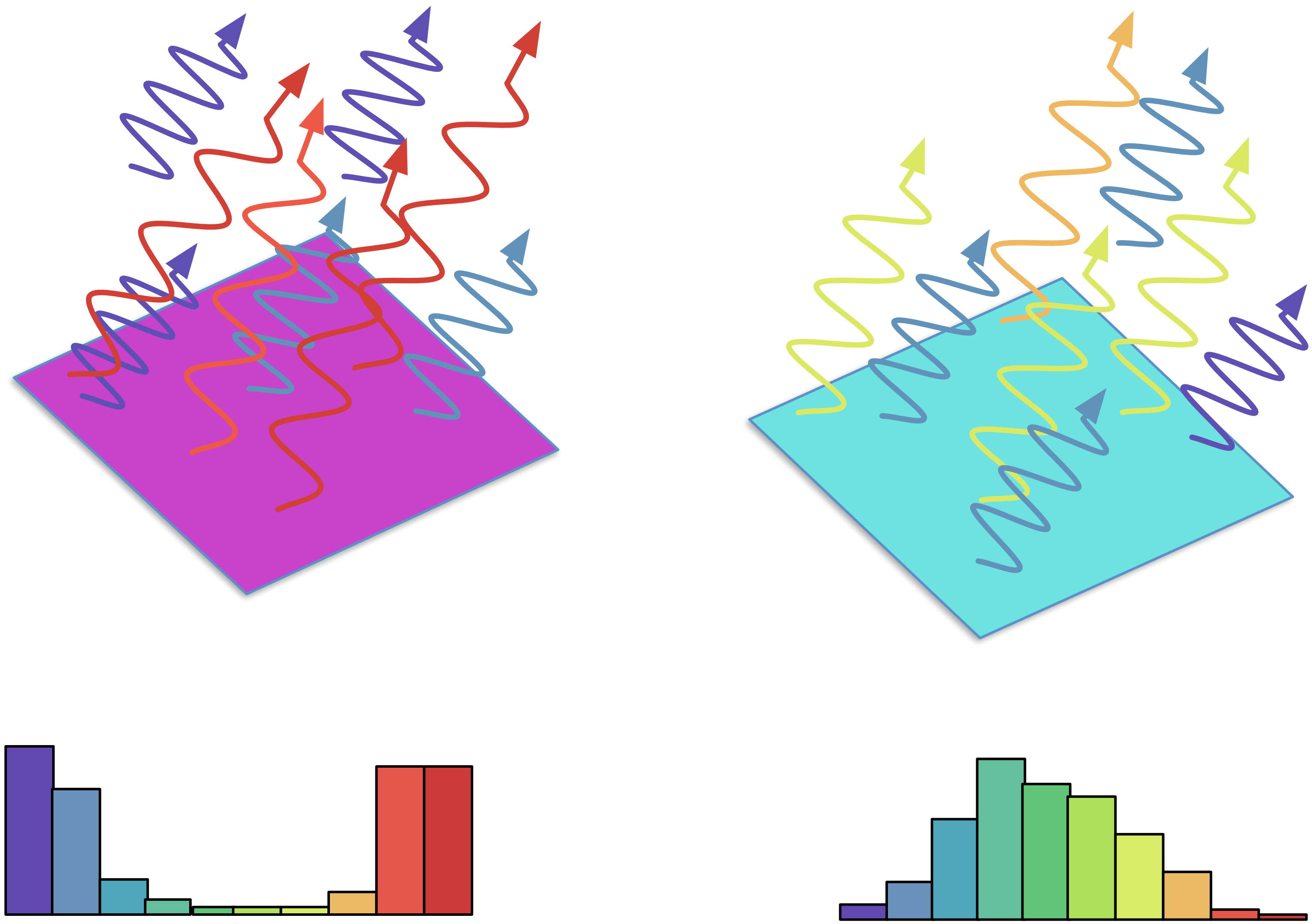
- The rainbow is missing white, pink, purple, ...

- There is more to color than what is in a rainbow!

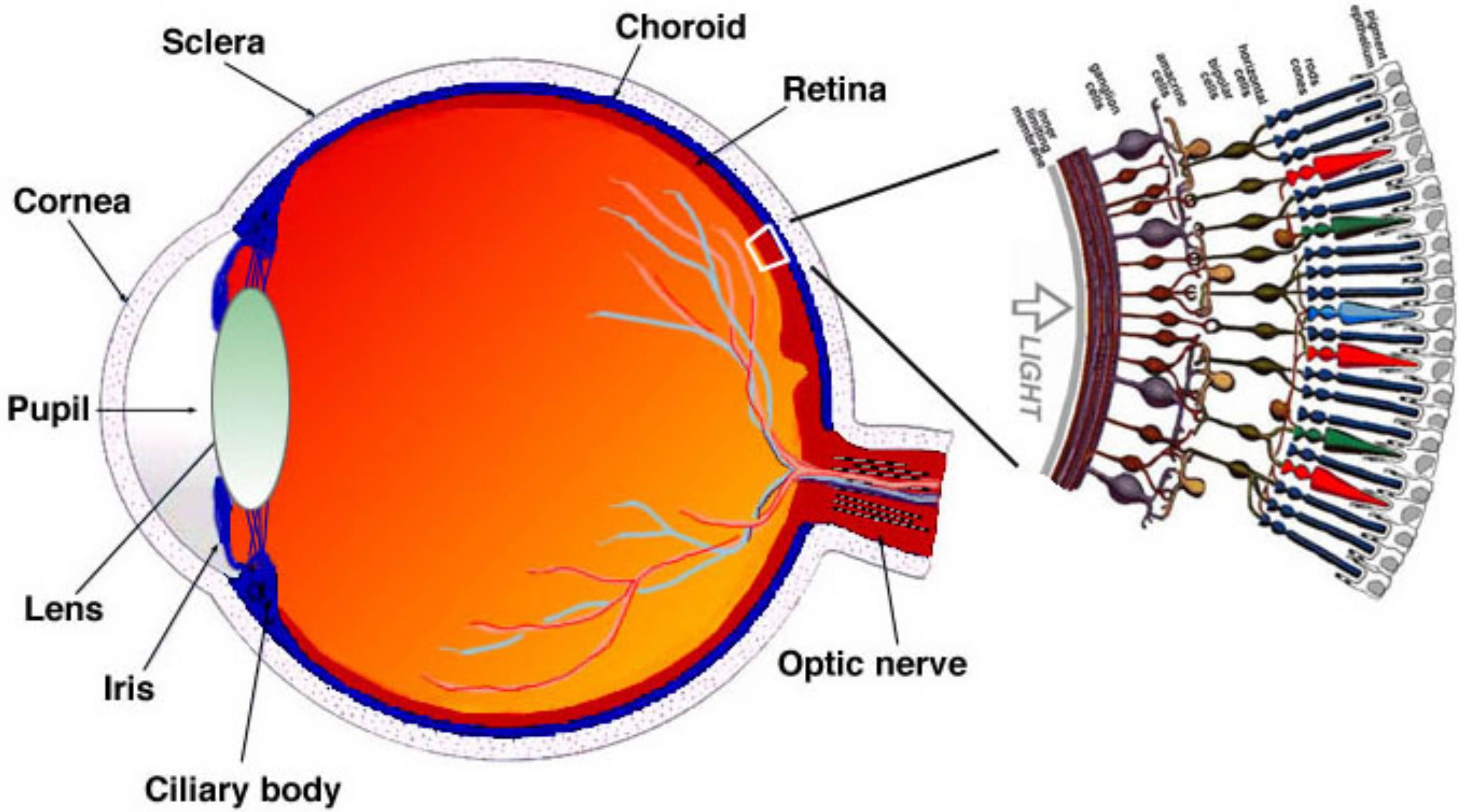


A representation of the colors we see at a given brightness.

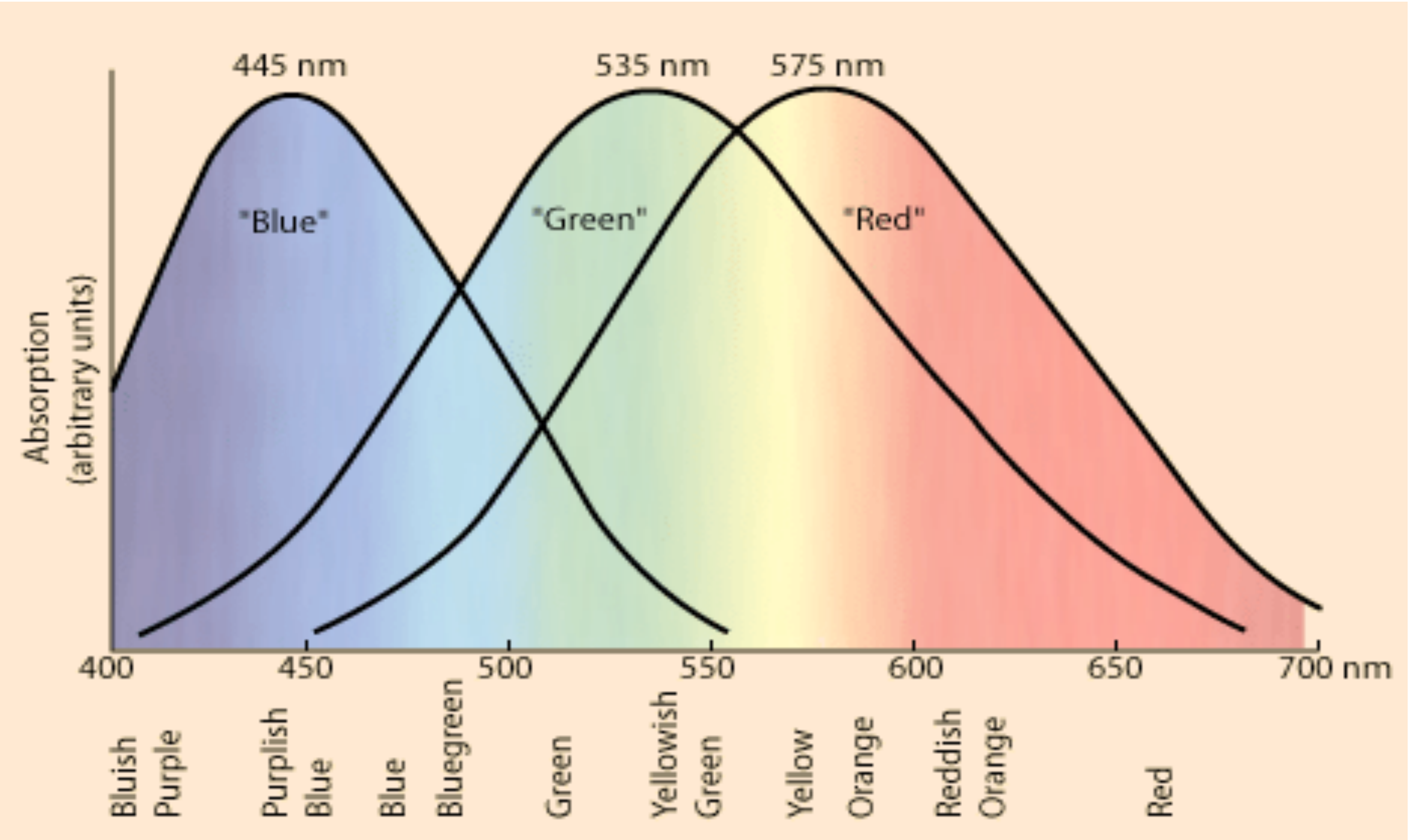
Light from two different surfaces



Color Vision Basics

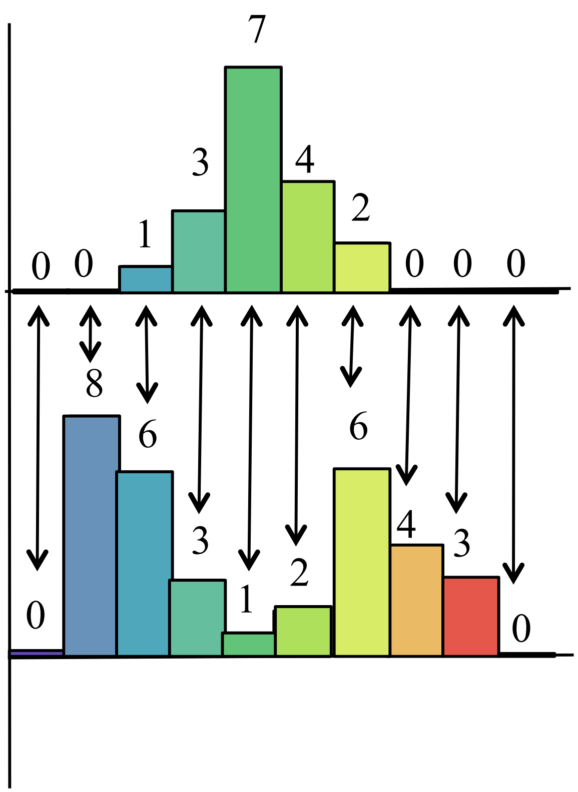

Sclera
Choroid
Retina
Cornea
Pupil
Lens
Iris
Ciliary body
Optic nerve
inner limiting membrane
ganglion cells
amacrine cells
bipolar cells
horizontal cells
rods
cones
pigment epithelium
LIGHT

Approximate spectra sensitivity for the three cone types



Sensor/light interaction example

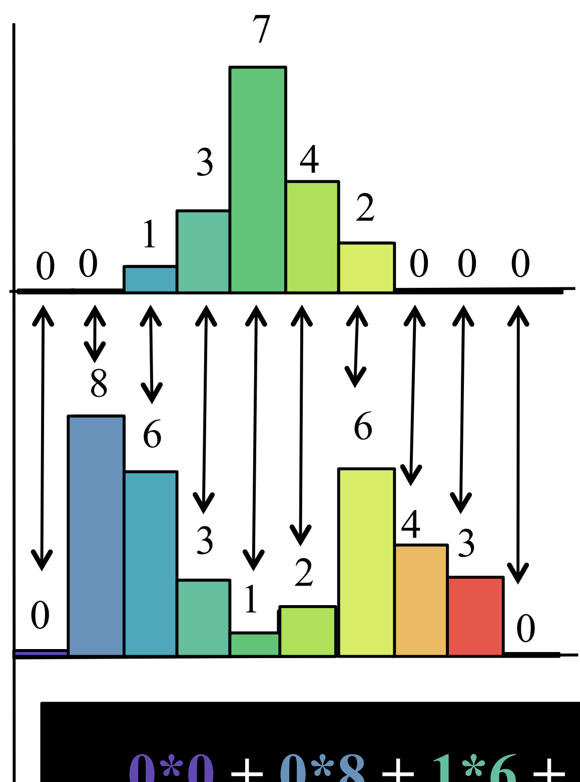
sensor



photon distribution

Multiply lined up pairs of numbers and then sum up

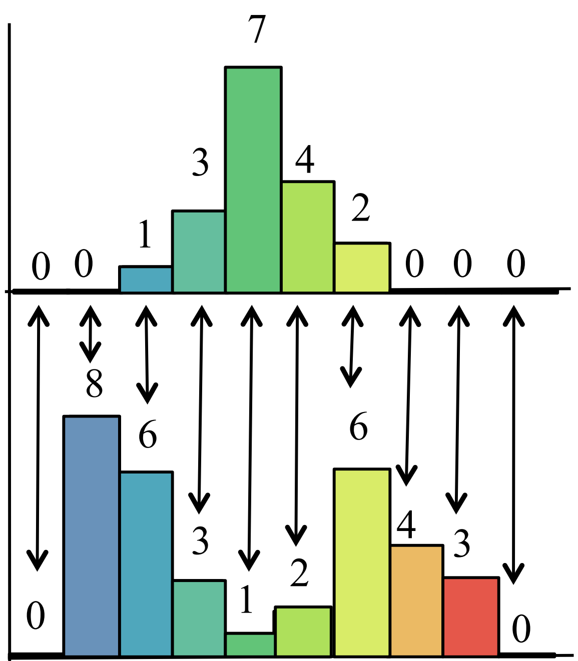
Sensor/light interaction example



Multiply lined up pairs of numbers and then sum up

$$0*0 + 0*8 + 1*6 + 3*3 + 7*1 + 4*2 + 2*6 + 0*4 + 0*3 + 0*0 = 42$$

Sensor/light interaction example



This suggests that sensor/ light interaction is linear

$$0*0 + 0*8 + 1*6 + 3*3 + 7*1 + 4*2 + 2*6 + 0*4 + 0*3 + 0*0 = 42$$

Review of the main points so far

- **Physics**. The light signal is a distribution of photons of different wavelengths

- **Human vision**. There are three cones, each which make a different weighted sum of the input

- Next top --- how to recreate the experience?


How to recreate the experience?

- **Plan A**. Simply duplicate the light signal present when the image was taken
  - This works, but it is **impractical**

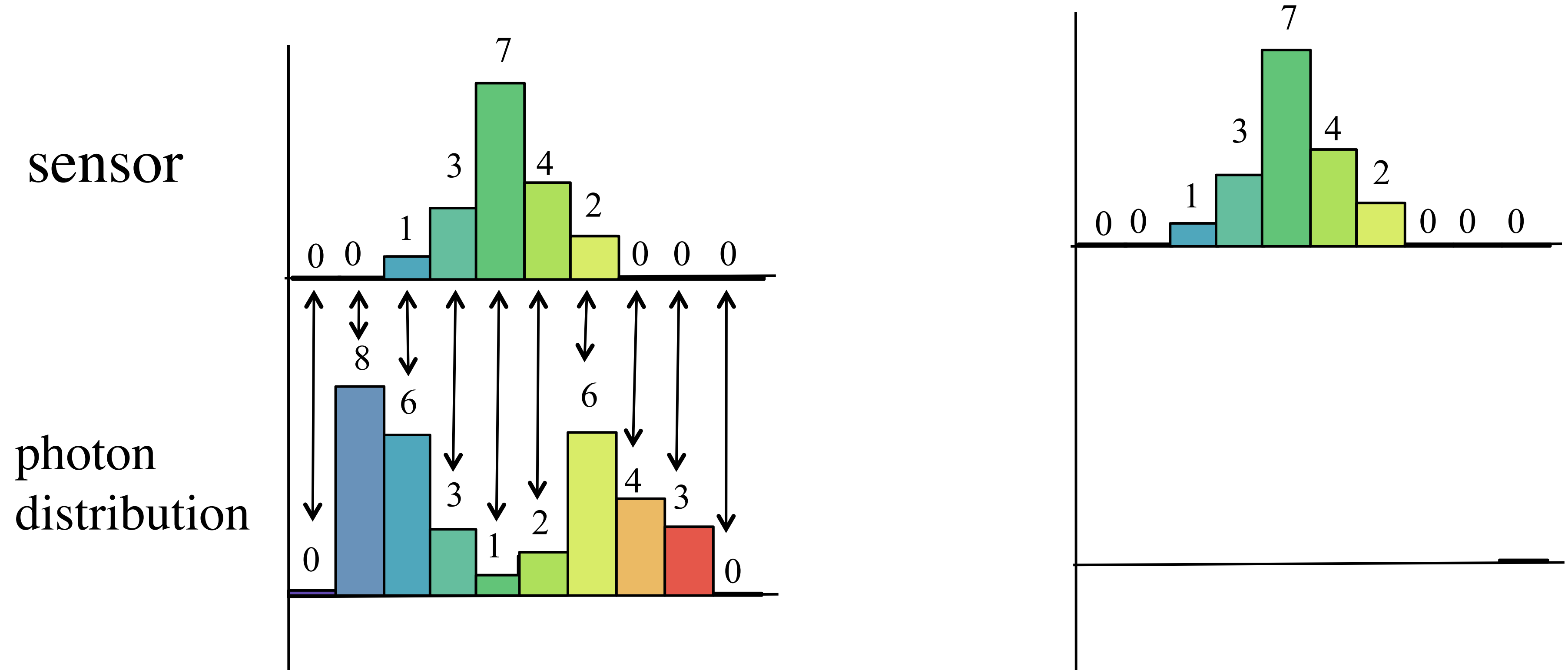
How to recreate the experience?

- **Plan A**. Simply duplicate the light signal present when the image was taken
  - This works, but it is **impractical**

- **Plan B.** Duplicate the sensor responses

Key idea

Do you need the same light to get 42?



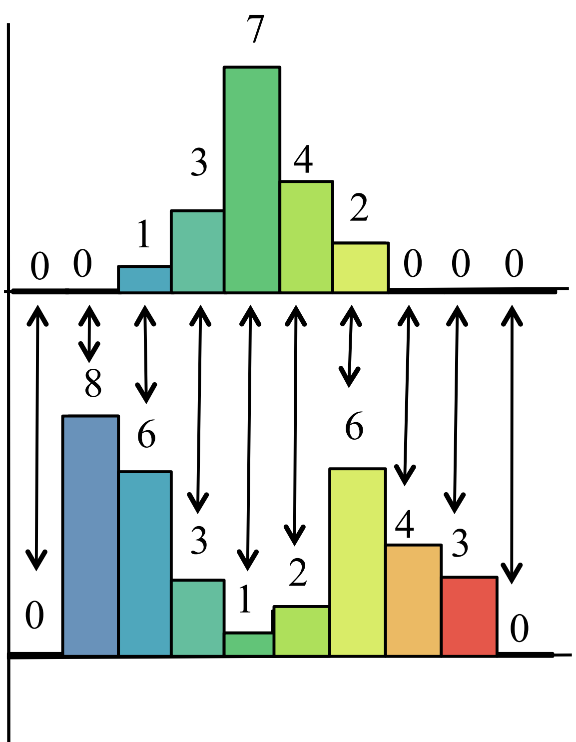
0*0 + 0*8 + 1*6 + 3*3 + 7*1
+ 4*2 + 2*6 + 0*4 + 0*3 + 0*0 = 42

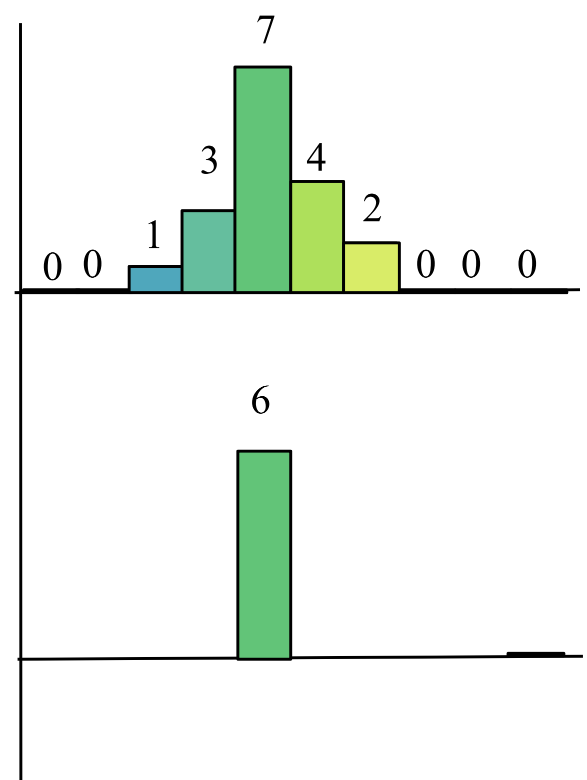
0*? + 0*? + 1*? + 3*? + 7*?
+ 4*? + 2*? + 0*? + 0*? + 0*? = 42

There are many possible ways to get 42!

sensor

photon distribution





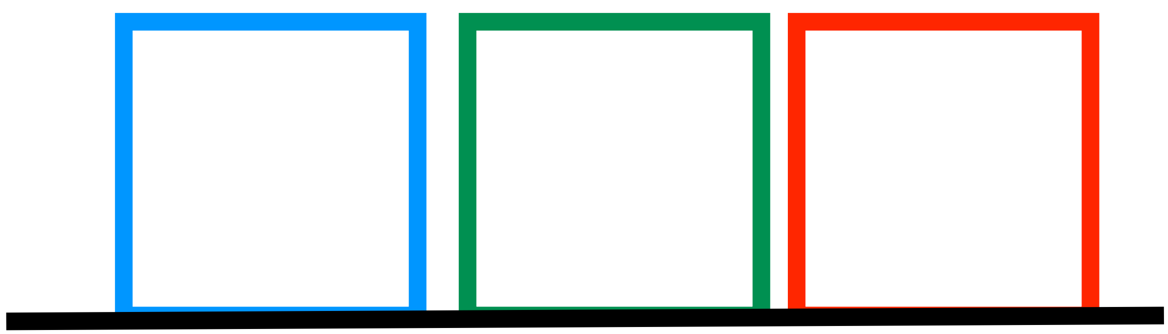
0*0 + 0*8 + 1*6 + 3*3 + 7*1 + 4*2 + 2*6 + 0*4 + 0*3 + 0*0 = 42

0*0 + 0*0 + 1*0 + 3*0 + 7*6 + 4*0 + 2*0 + 0*0 + 0*0 + 0*0 = 42

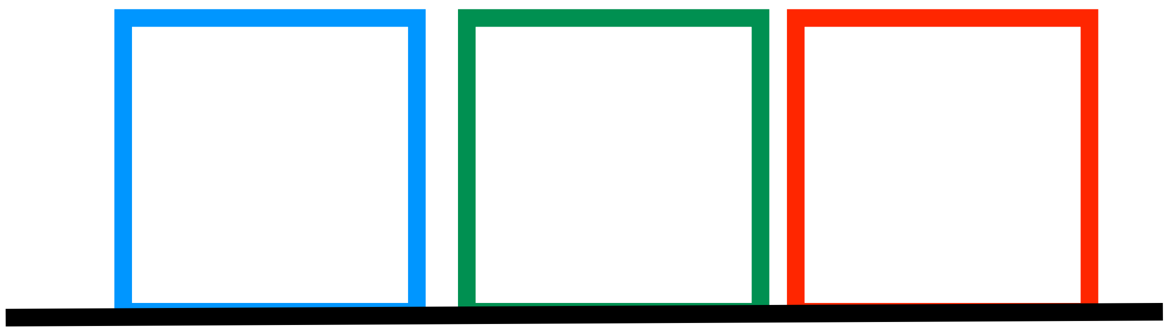
Main point

- To recreate the cone responses, stimulate each one **independently**.

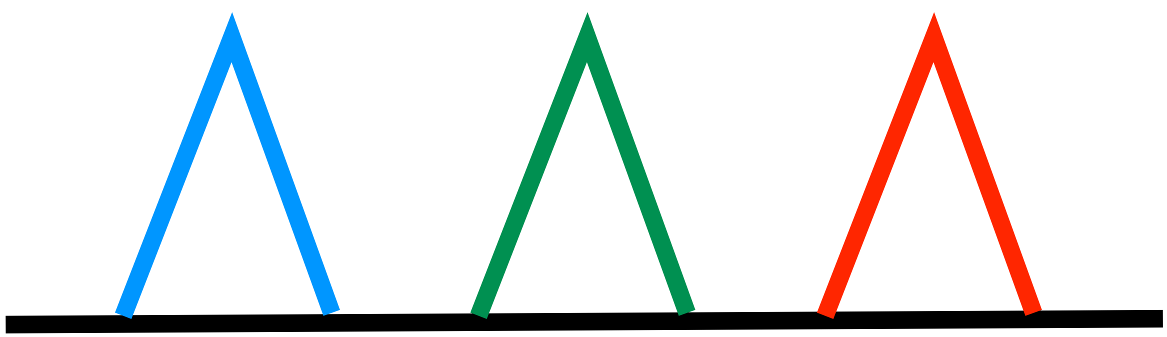
- Suppose that our cone sensors were like these simplified ones. Can this work?



Main point



Simplified sensors



Simplified ipad screen element photon distribution

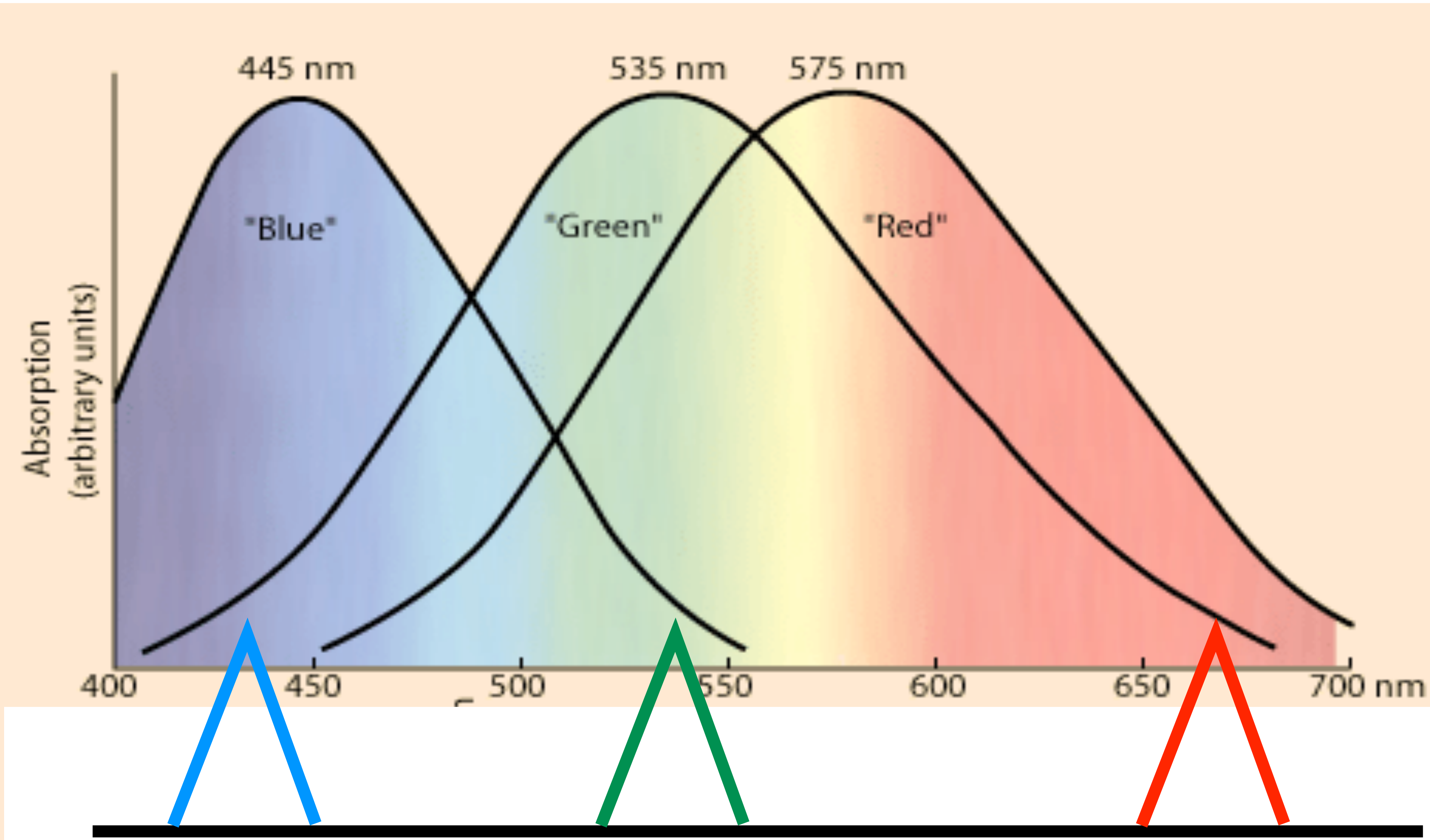
(primaries)

Recreating the sensor responses

- Note that the photon distribution to recreate the sensor response can look completely different from the original!

- We need to compute the amount of each primary needed for each color to display
  - This can be achieved by matrix-vector multiplication
    - (Details beyond the scope of this lecture)

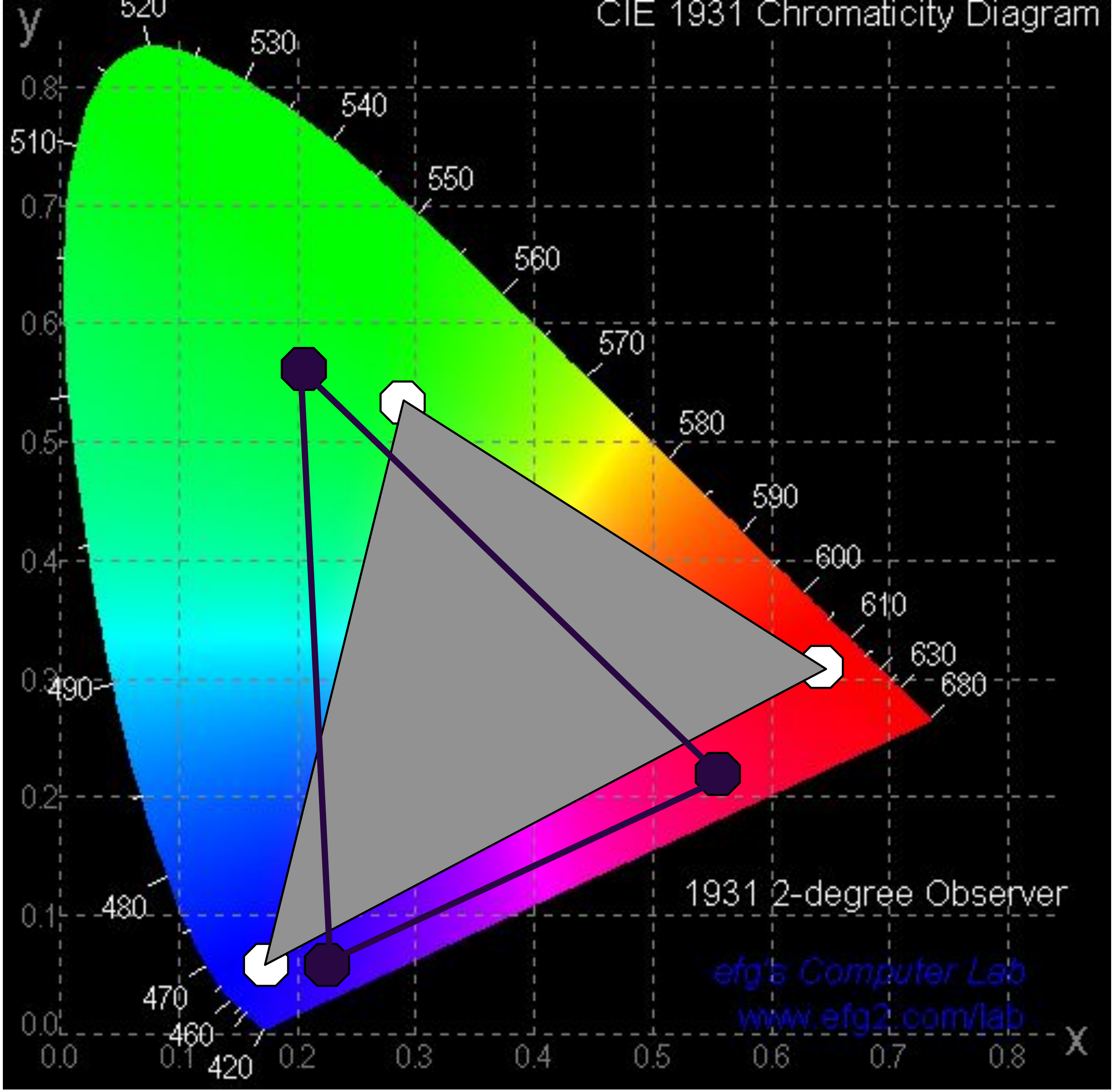
What about actual human sensors?



Second Main Point

- With three numbers, you **cannot** recreate all colors you can see.

- This is not a question of poor engineering. It is a consequence of the significant cone sensor **overlap**.



Available from efg2.com

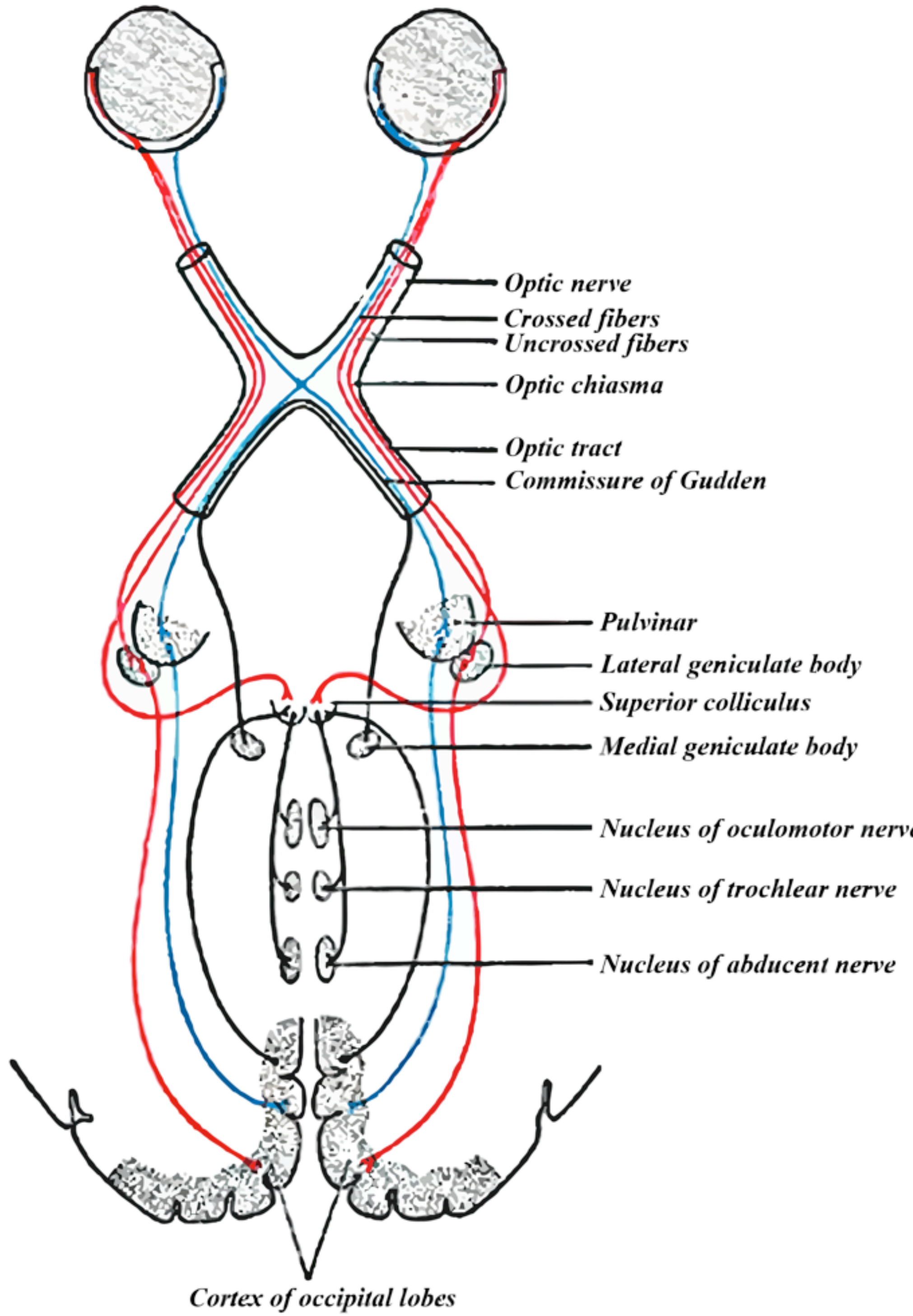
Downstream from the cones

- Color reproduction based on three numbers works relatively well anyway, partly because our brain is so adept at reconstructing a world based on relative **properties**.

- If you (approximately) reproduce the cone responses, you will reproduce the effect.

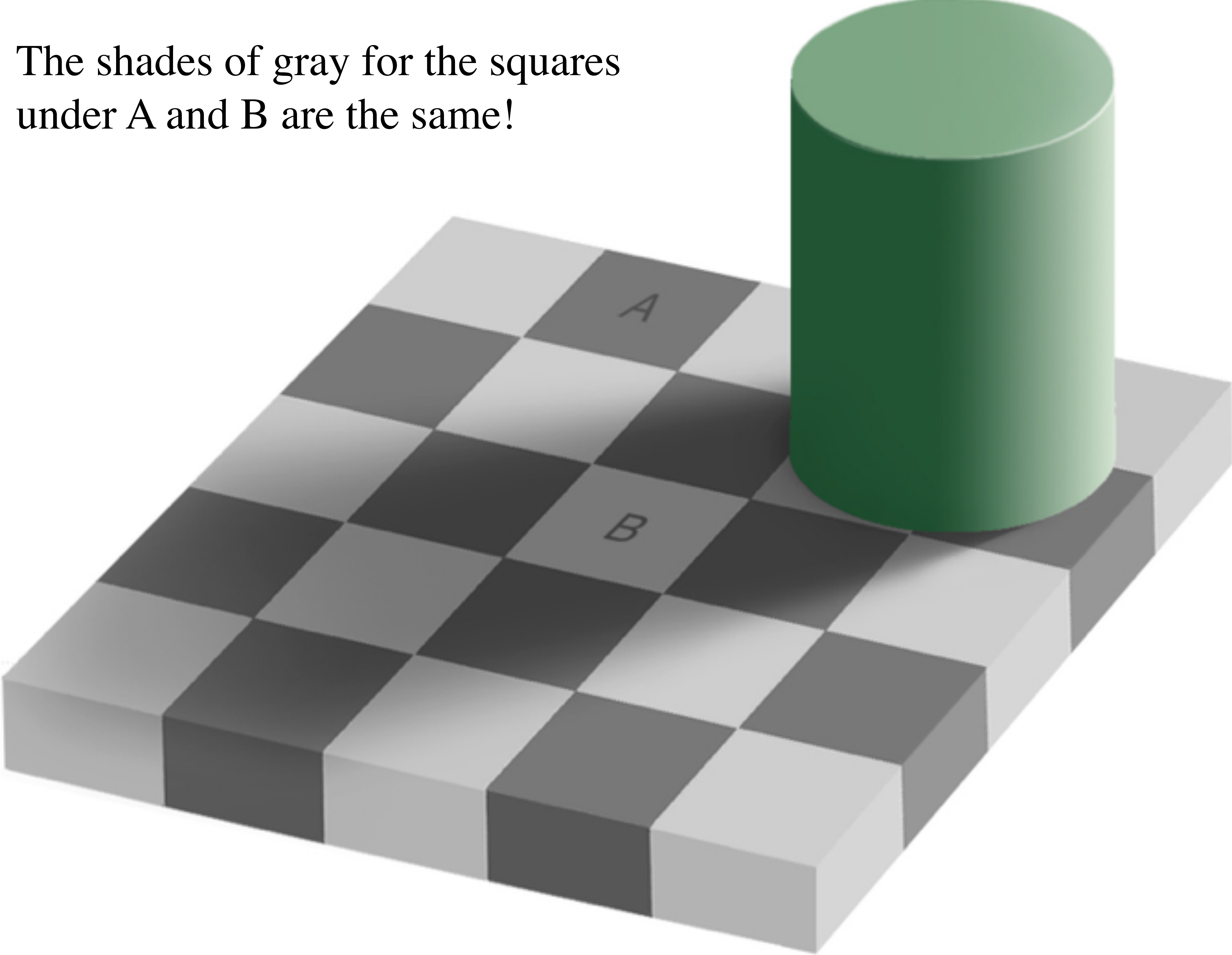
- But what you actually see is complex!

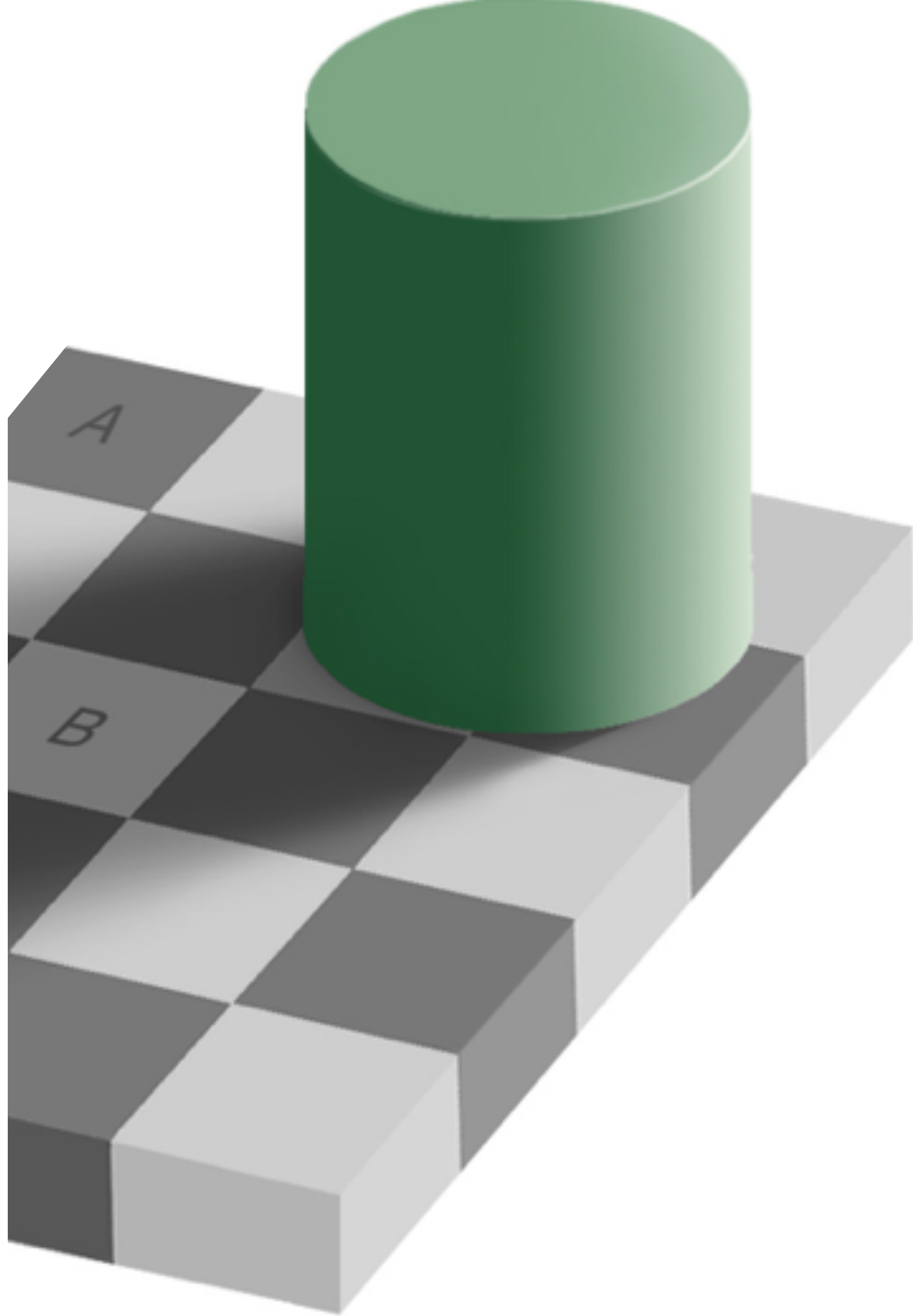
The HVS

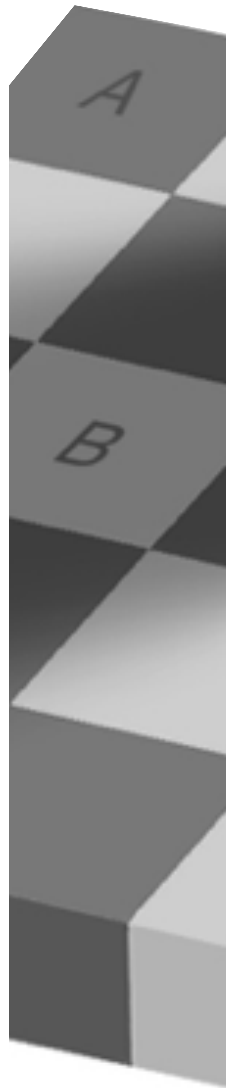


Optic nerve
Crossed fibers
Uncrossed fibers
Optic chiasma
Optic tract
Commissure of Gudden
Pulvinar
Lateral geniculate body
Superior colliculus
Medial geniculate body
Nucleus of oculomotor nerve
Nucleus of trochlear nerve
Nucleus of abducent nerve
Cortex of occipital lobes

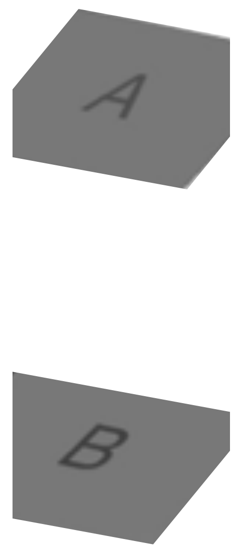
The shades of gray for the squares under A and B are the same!




A
B


A
B

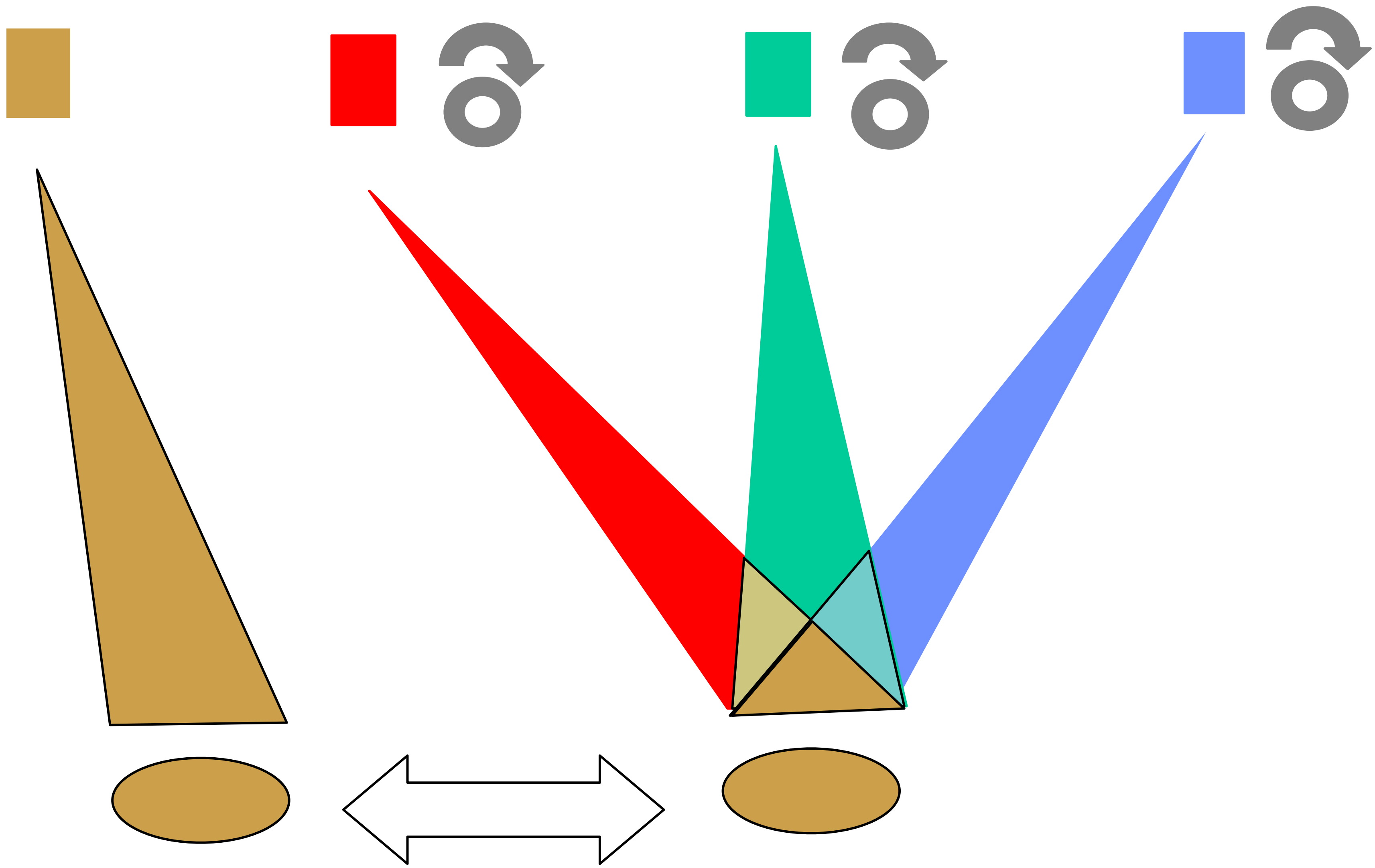

A
B


A
B

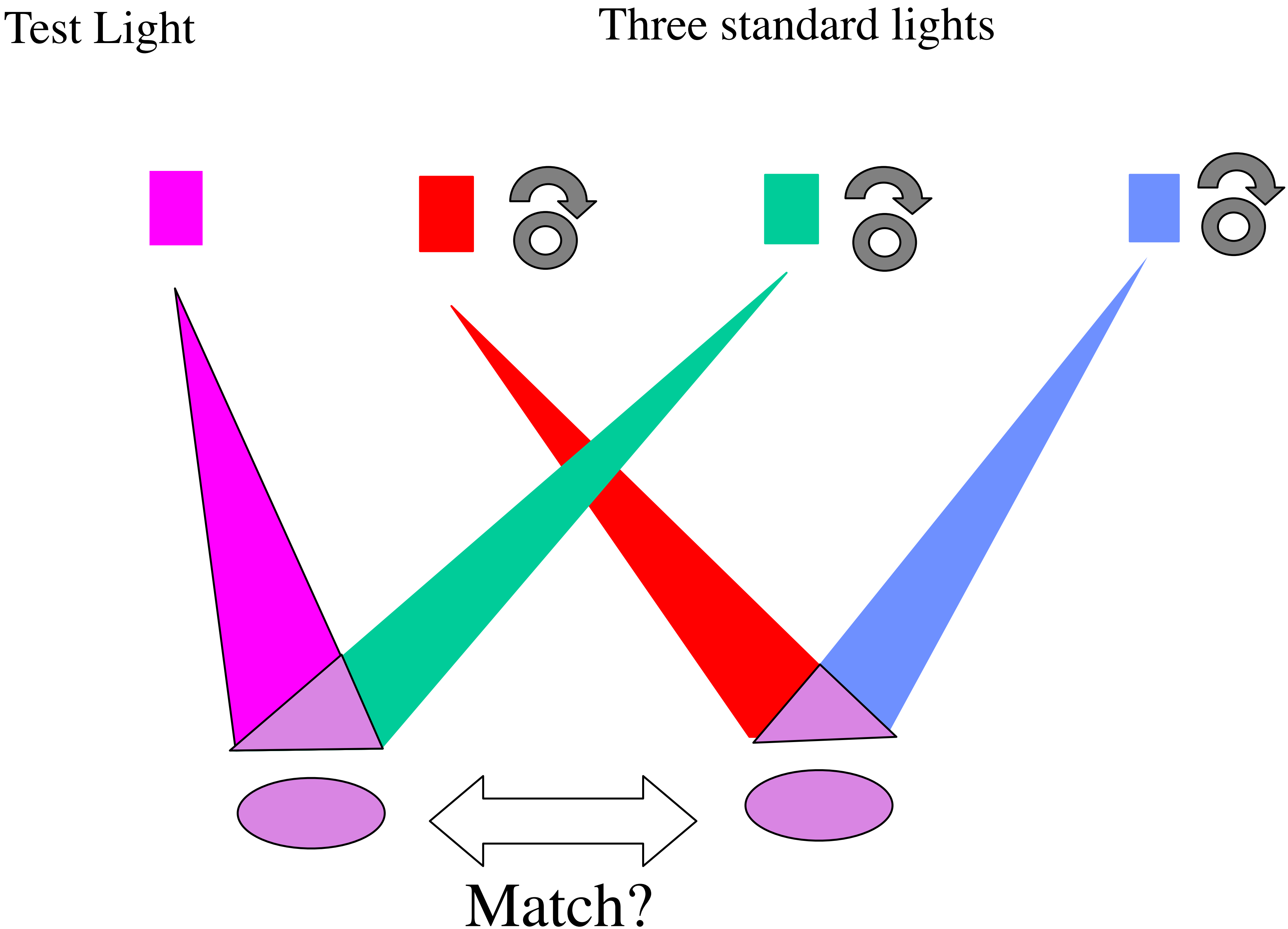
Specifying Colour

Test Light

Three standard lights



Match?

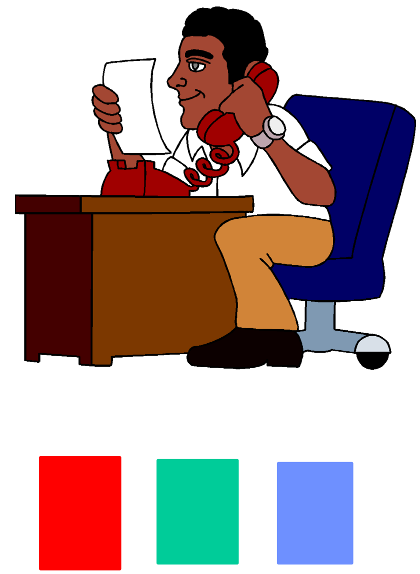

Test Light
Three standard lights
Match?

Trichromacy

Experimental fact about people (with "normal" colour vision)

Specifying Colour

(50,150,75)



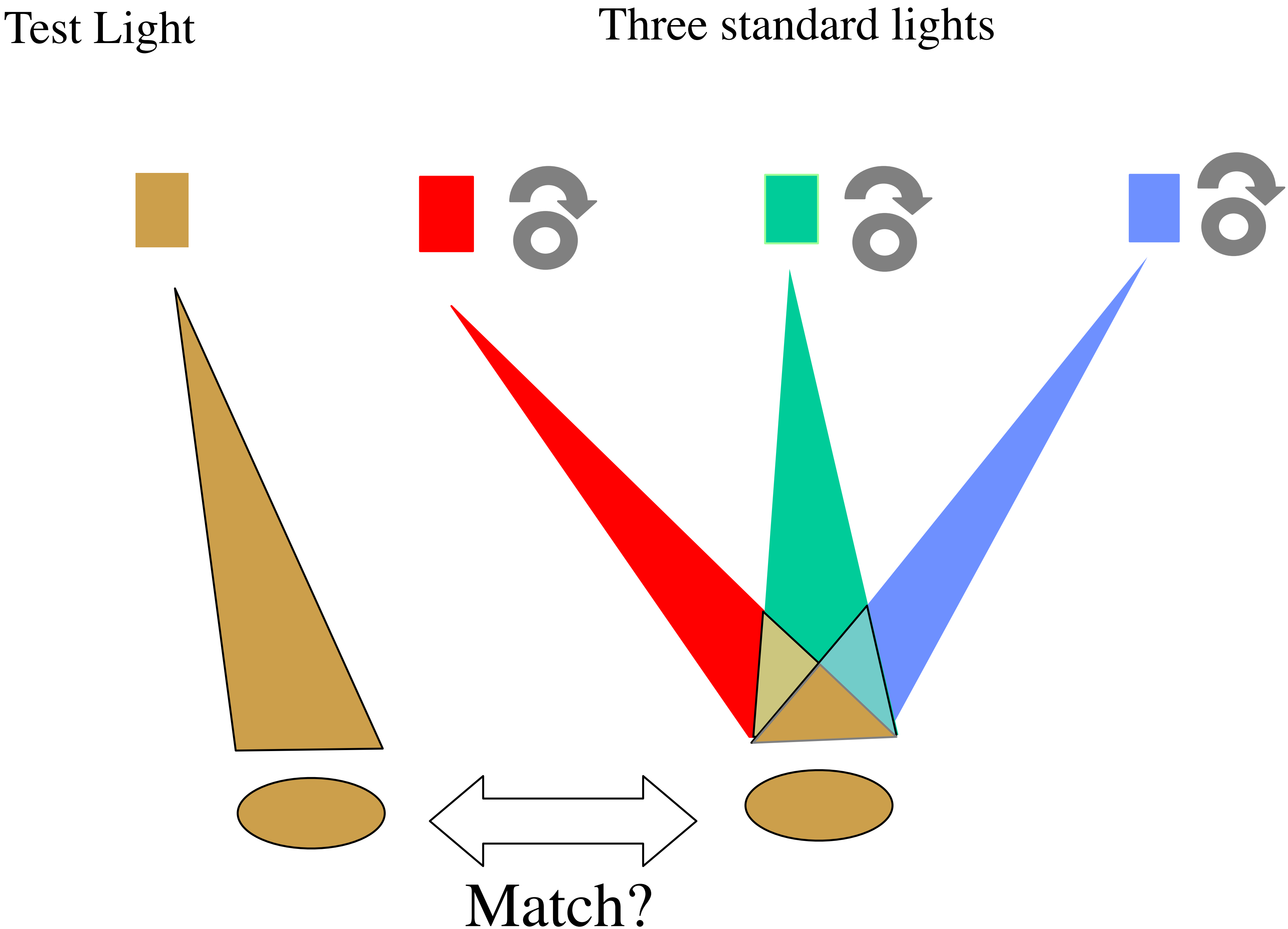
(50,150,75)

Specifying Colour

We don’t want to do a matching experiment every time we want to use a new color!

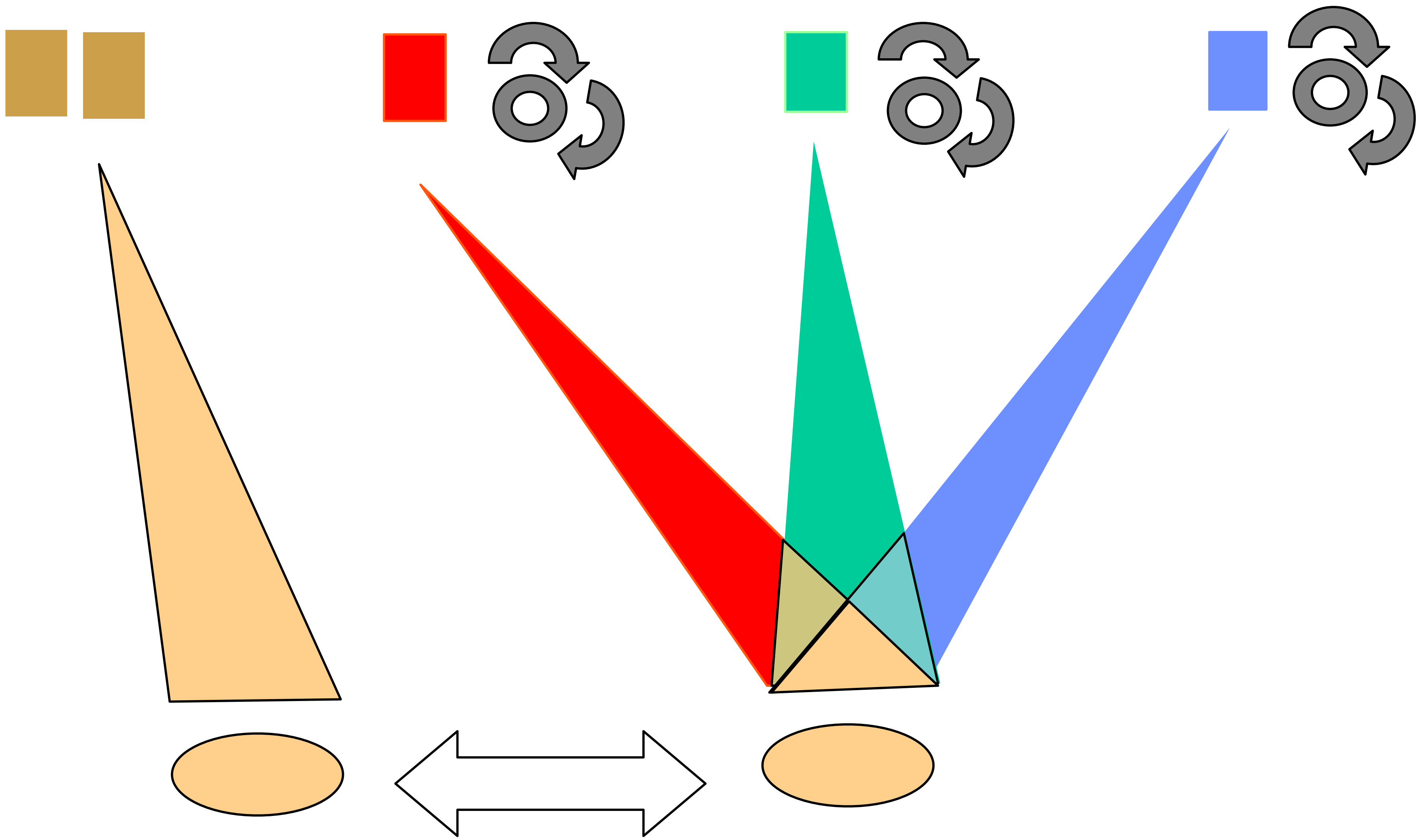
Grassman's Contribution

Colour matching is linear


Test Light
Three standard lights
Match?

Test Light

Three standard lights



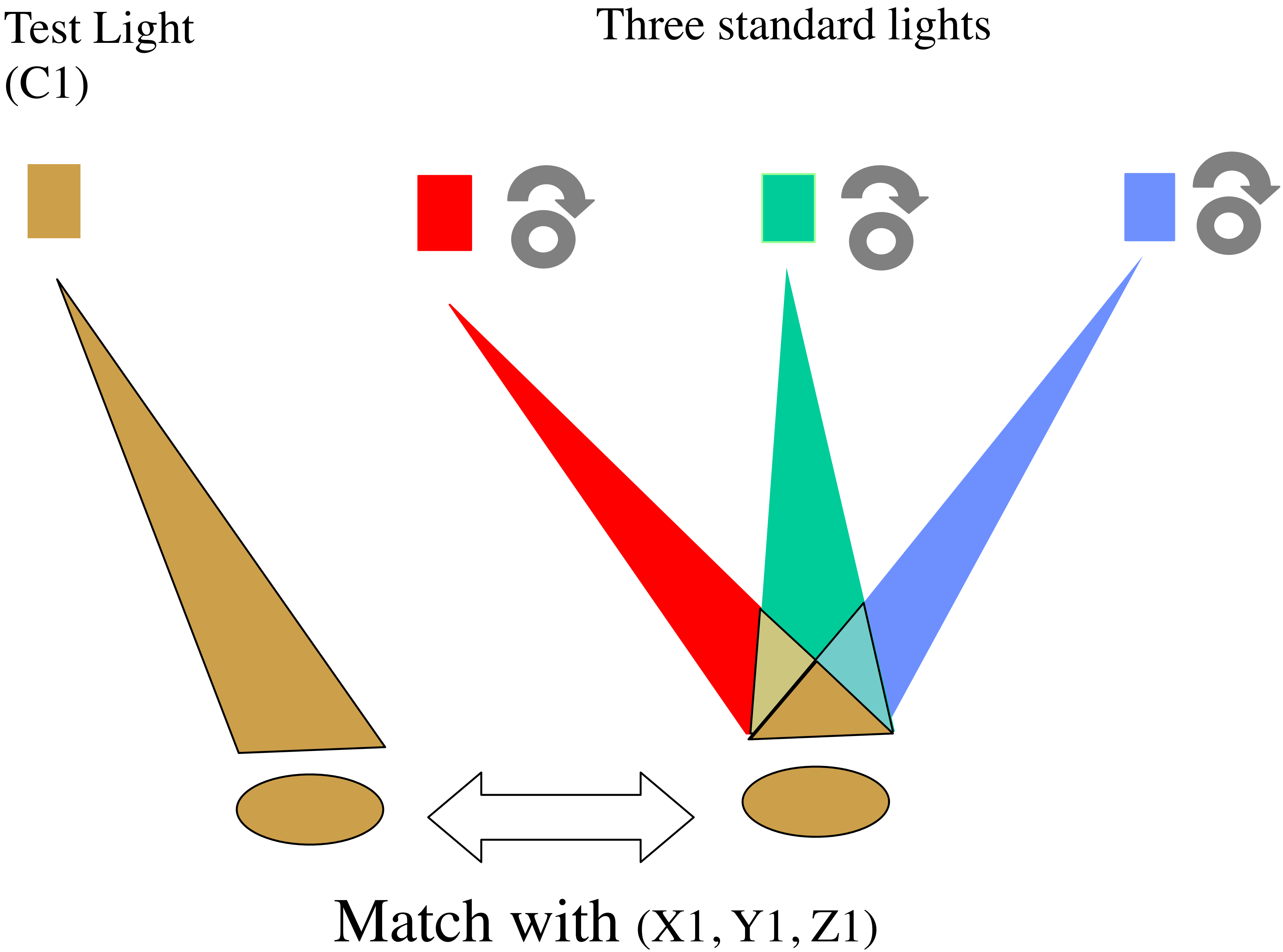
Match (with twice as much)

Matching is Linear (Part 1)

C1 is matched with (X1,Y1,Z1)

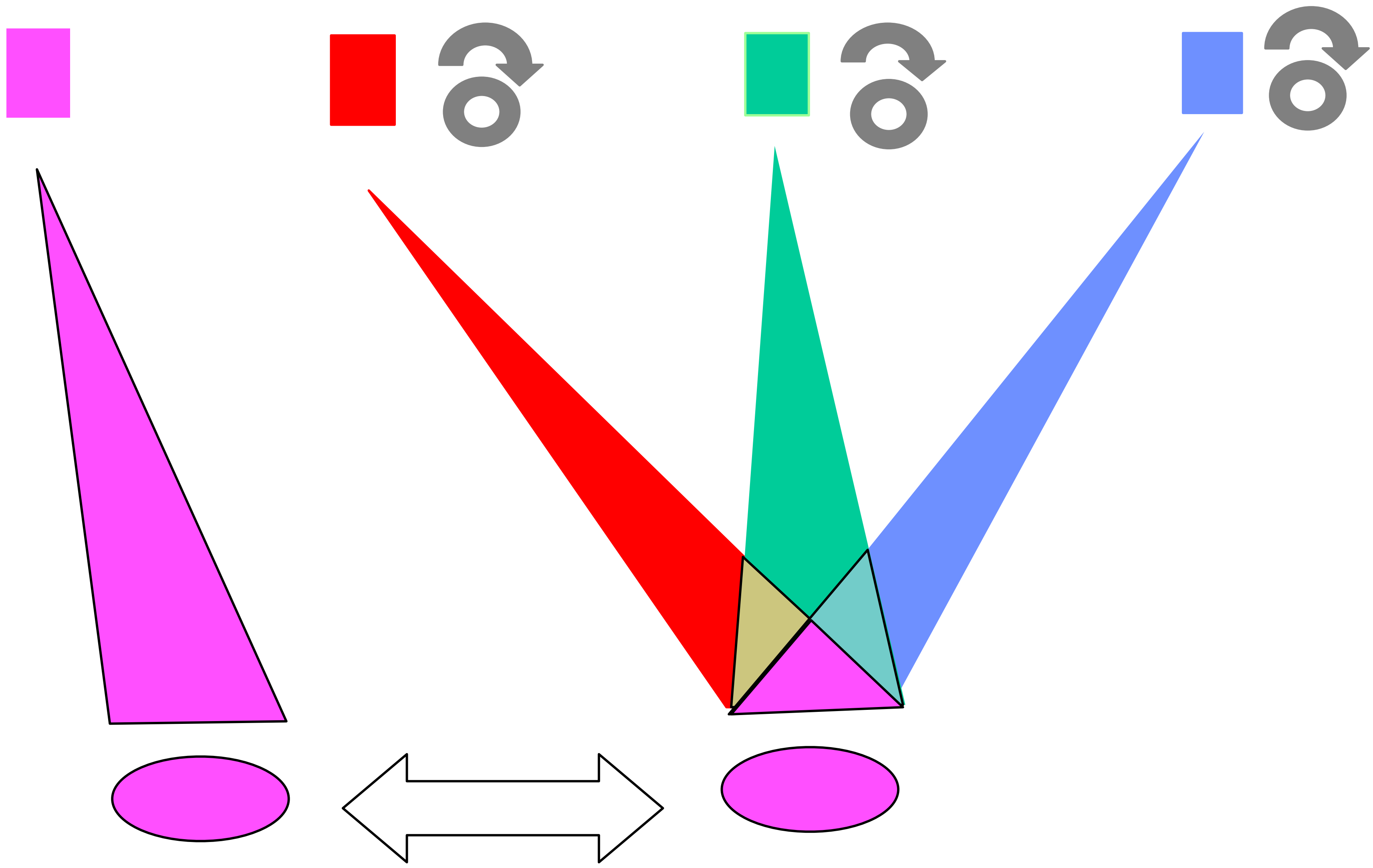
C = a*C1

C is matched with a * (X1, Y1, Z1)

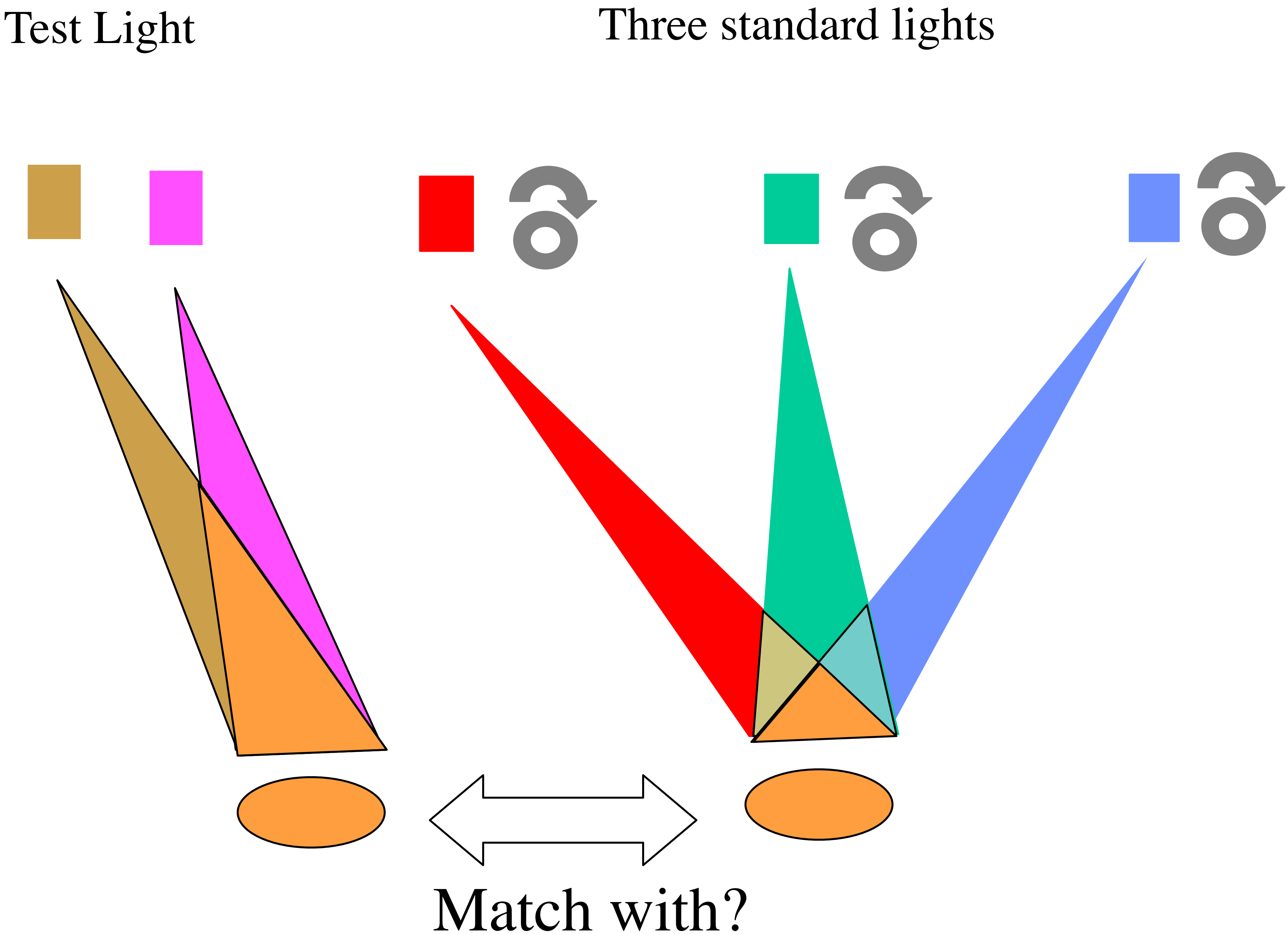

Test Light
(C1)
Three standard lights
Match with (X1, Y1, Z1)

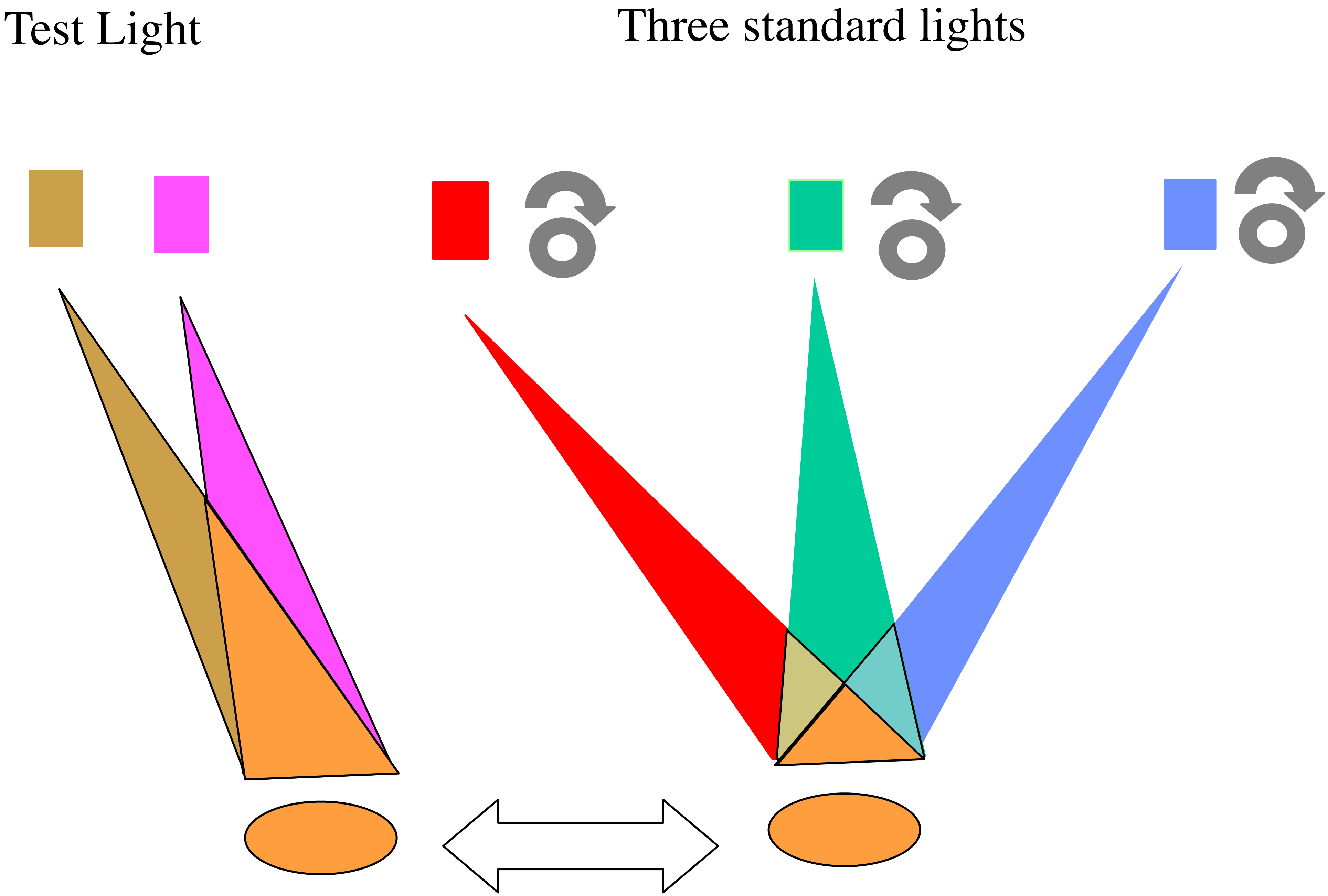
Test Light
(C2)

Three standard lights



Match with (X2, Y2, Z2)


Test Light
Three standard lights
Match with?



Match with (X1+X2, Y1+Y2, Z1+Z2)

Matching is Linear (formal)

C = a*C1 + b*C2

C1 is matched with (X1,Y1,Z1)
C2 is matched with (X2,Y2,Z2)

C is matched by
a*(X1,Y1,Z1) + b*(X2,Y2,Z2)

Specifying Color

On my monitor it's
(R,G,B) = (75,150,100)

Specifying Colour

But what is (R,G,B)?

Specifying Colour

R matches (X_r, Y_r, Z_r)

G matches (X_g, Y_g, Z_g)

B matches (X_b, Y_b, Z_b)

Specifying Colour

Then by (R,G,B)=(75,150,100) you mean (X,Y,Z), where …..

$$X = 75 * X_r + 150 * X_g + 100 * X_b$$

$$Y = 75 * Y_r + 150 * Y_g + 100 * Y_b$$

$$Z = 75 * Z_r + 150 * Z_g + 100 * Z_b$$

(No need to match--just compute!)

Specifying Colour

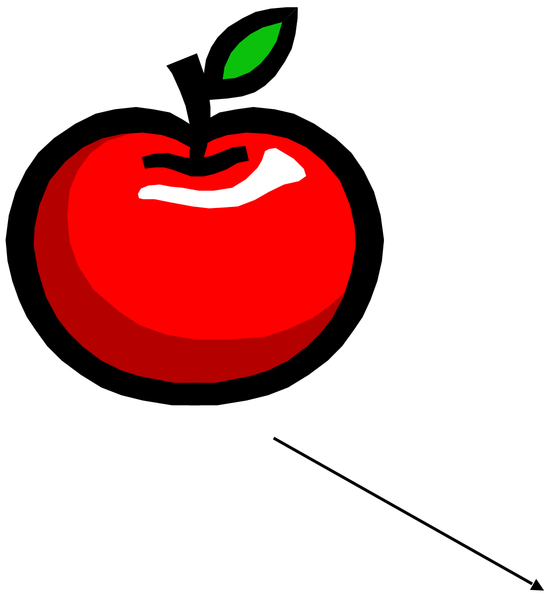
… , now that we have specified the colour, I can print it!

$$\begin{vmatrix} X \\ Y \\ Z \end{vmatrix} = \begin{vmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{vmatrix} \begin{vmatrix} 75 \\ 100 \\ 150 \end{vmatrix}$$

$$
\begin{vmatrix} X \\ Y \\ Z \end{vmatrix} = \begin{vmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{vmatrix} \begin{vmatrix} R \\ G \\ B \end{vmatrix}
$$

$$\begin{vmatrix} X \\ Y \\ Z \end{vmatrix} = M \begin{vmatrix} R \\ G \\ B \end{vmatrix}$$

Colour Reproduction
(Monitors & Projectors)



$$\begin{vmatrix} X \\ Y \\ Z \end{vmatrix}_{apple}$$

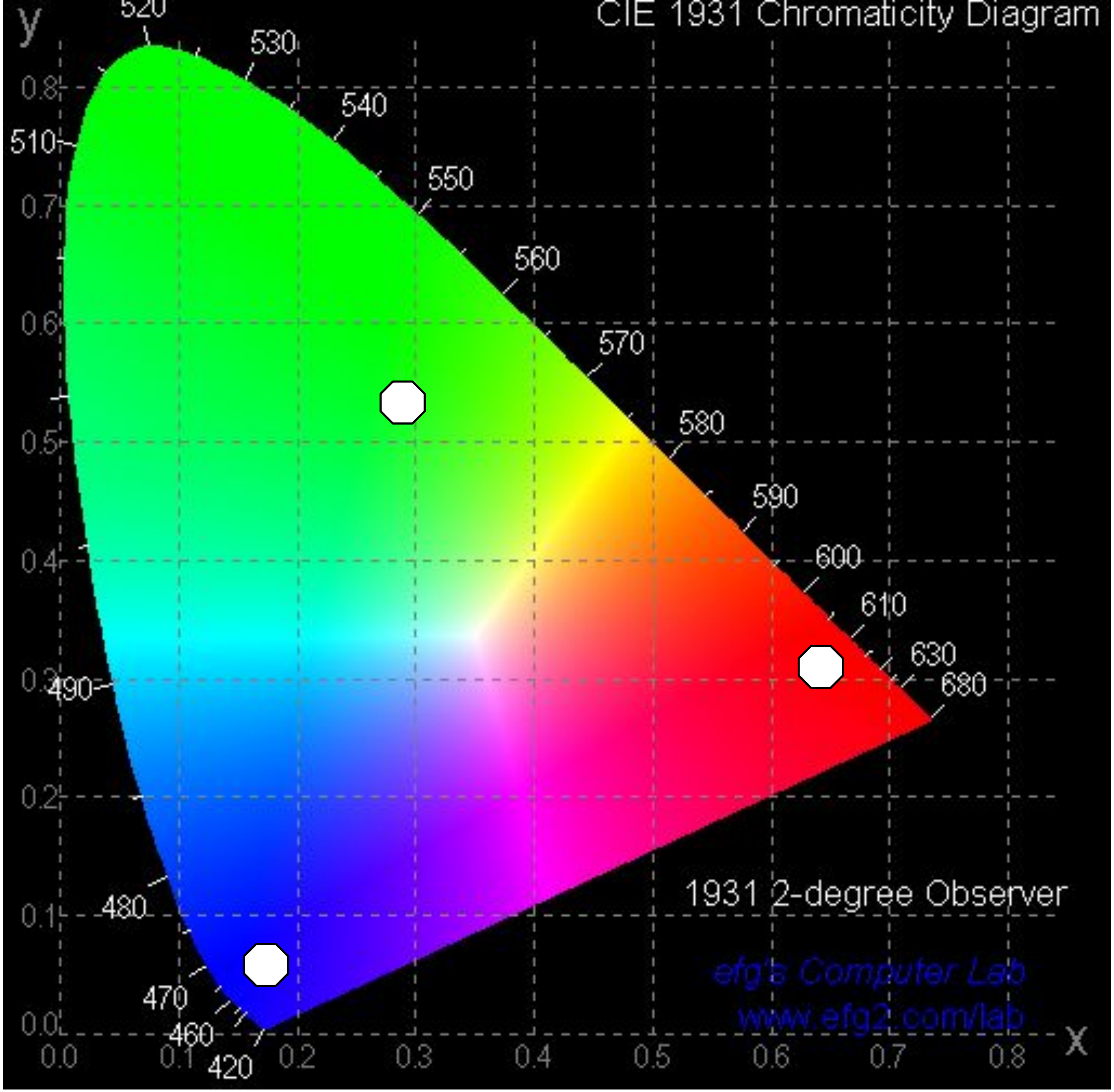
Find (R,G,B)

$$\begin{vmatrix} X \\ Y \\ Z \end{vmatrix}_{apple} = M \begin{vmatrix} R \\ G \\ B \end{vmatrix}_{apple}$$

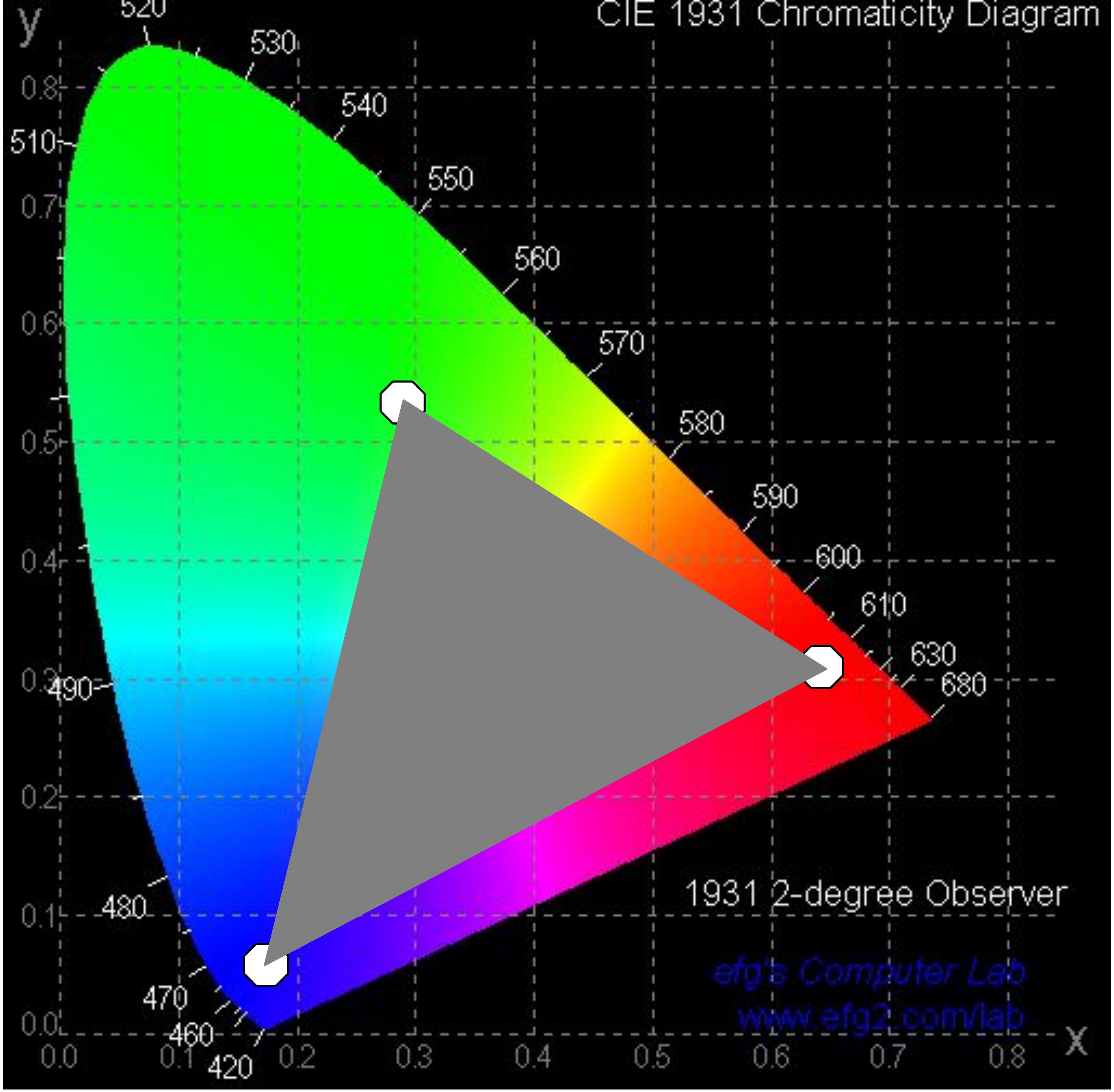
$$\begin{vmatrix} R \\ G \\ B \end{vmatrix}_{\text{apple}} = M^{-1} \begin{vmatrix} X \\ Y \\ Z \end{vmatrix}_{\text{apple}}$$

$$\begin{vmatrix} R \\ G \\ B \end{vmatrix}_{apple} = M^{-1} \begin{vmatrix} X \\ Y \\ Z \end{vmatrix}_{apple}$$

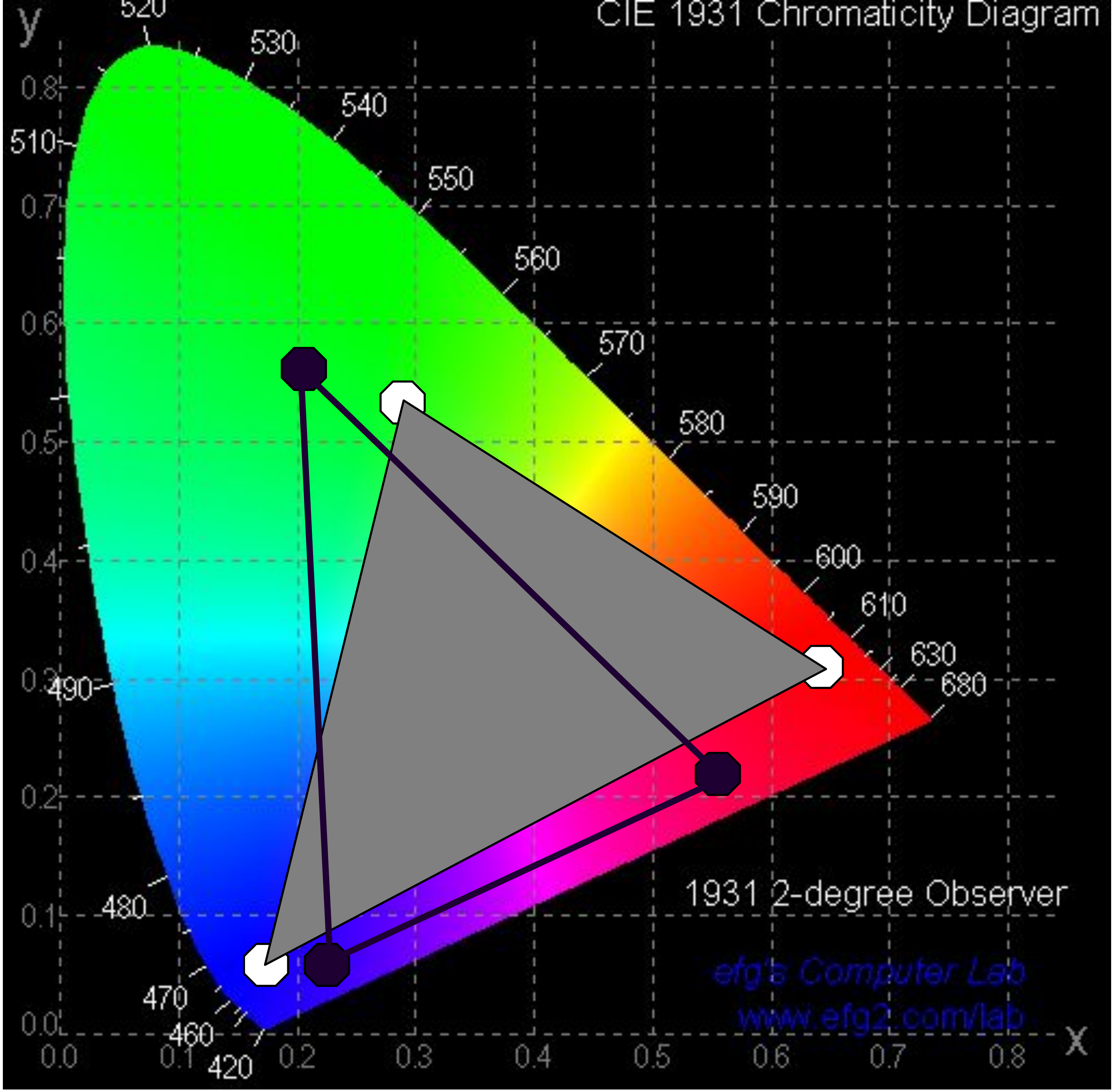
Possible problems?



Avalable from efg2.com



Avalable from efg2.com



Avalable from efg2.com

Luminosity is not linear

Luminosity is not linear

- There is a huge dynamic range of brightness in the world we need to navigate
- Your response to brightness is controlled by various factors such as aperture size
- If one had to put a mathematical function on brightness, log() might be a good choice.

Image encoding is not linear either

Deviations from our nice model

- Camera “black”
- Gamma

Camera Black

- Sensors always produce electrons, even if there is no light
- The effect increases as the temperature increases
- We can improve the model by adding a fixed offset
  - Specifically, the R,G, and B recorded with the lens cap on
- The resulting model is not a linear transformation
  - Technically, it is “affine”

Gamma correction

- For complicated reasons, the final output of a camera is often a non-linear transformation of the RGB described so far.
- Usually the same transformation is used for R, G, and B
- A typical "gamma correction" transformation is approximately

$$F(x) = 255 * \left(\frac{x}{255} \right)^{1/2.2} \qquad \text{(roughly square root)}$$

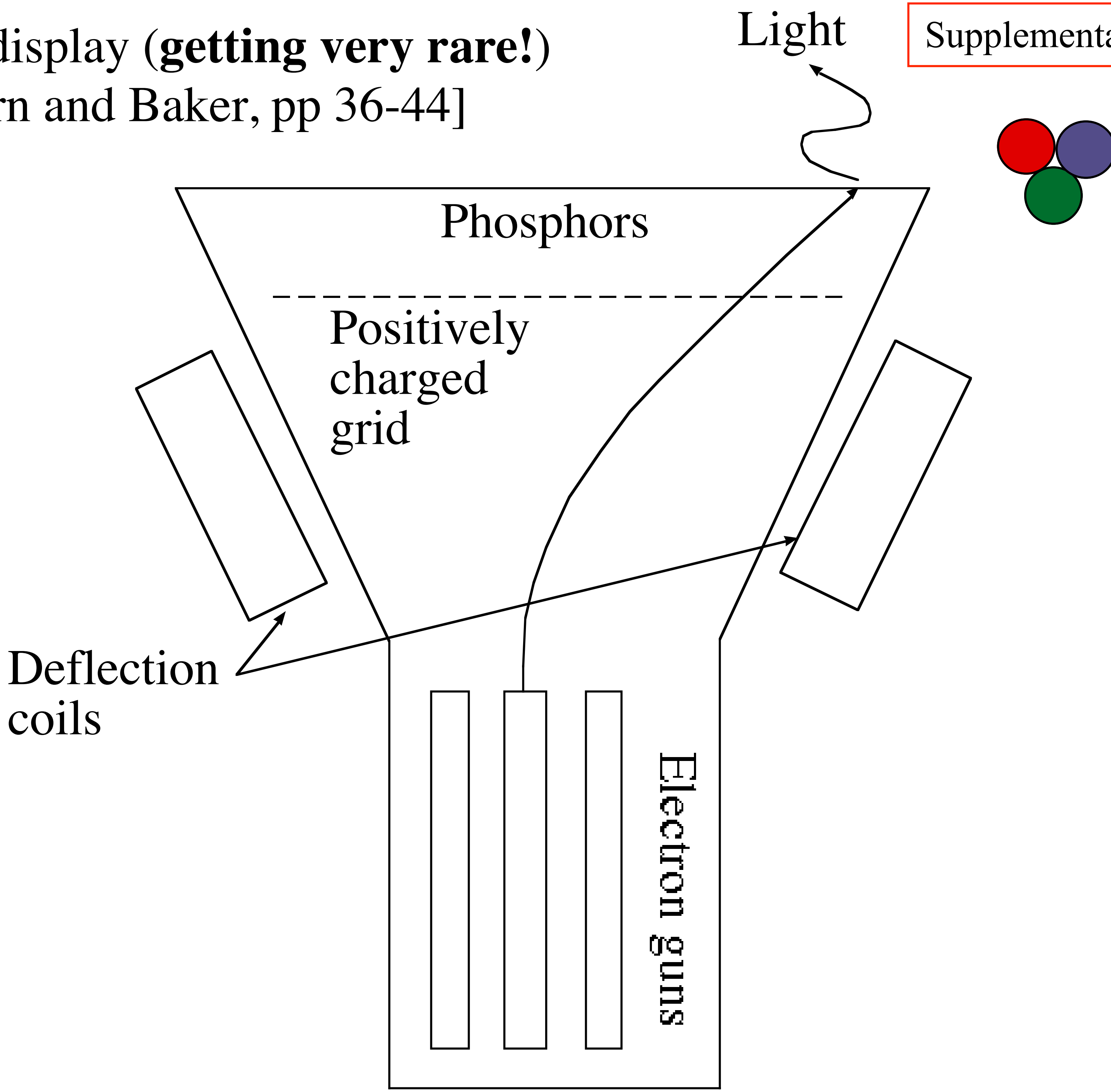
Image Formation (non-linear transform)

Why are images typically encoded in this way?

Historically, images have been gamma corrected on the assumption that their values drive a CRT (cathode ray tube) monitor which are non-linear devices.

CRT display (**getting very rare!**)
[Hearn and Baker, pp 36-44]

Supplemental material



Supplemental material

Gamma encoding

- In the CRT, for a given input voltage, V, electrons hit the phosphors with energy E

$$E \propto V^{\gamma}, \quad \text{where } \gamma \text{ is } \tfrac{5}{2} \quad (\text{i.e., } 2.5)$$

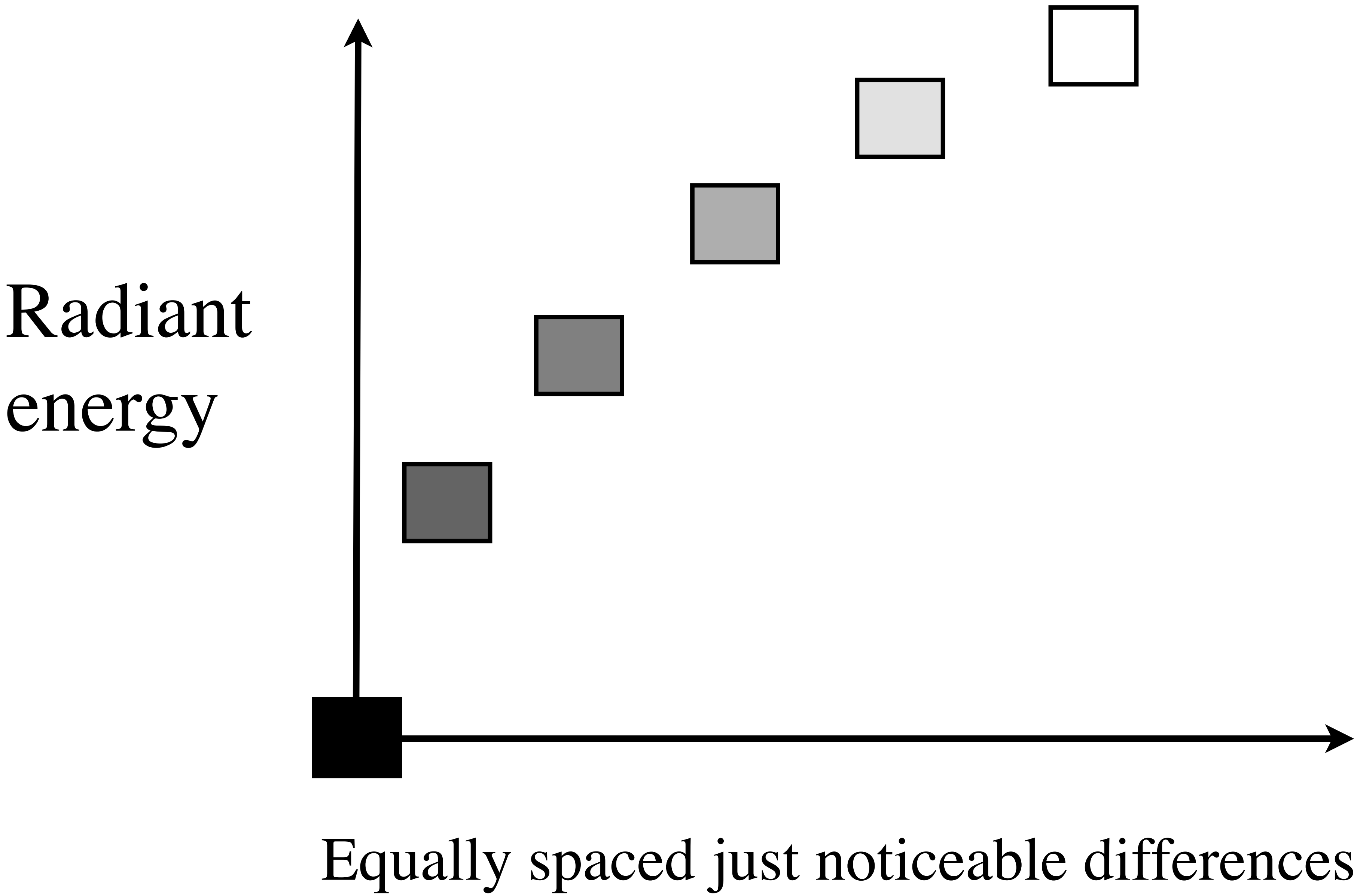
- So, to drive the CRT so that the output energy is linear (recreating its capture) you send it a voltage

$$V \propto E^{1/\gamma}$$

Image Formation (non-linear transform)

Coincidentally, this typically gamma correction is a sensible way to encode image data into a limited number of values (e.g. 256) due to the noise sensitivity of the human vision system.

Hence, while CRT displays are now obsolete, images are still typically non-linear, and the signal to modern displays (which are linear) are typically adjusted assuming typical incoming non-linear in images.


Radiant
energy
Equally spaced just noticeable differences

Gamma encoding

- The non-linear encoding means that linear displays (now common) need to implement the mapping from gamma encoded to linear
  - One way to think about it is that they have to emulate CRT monitor
  - Gamma is also becoming an image tone correction "knob" that either fixes an incorrect value, or simply makes some images look better.

Gamma calibration

- How can your mac robustly emulate a gamma of 2.2 for your monitor?
  - The OS has no idea what you have hooked up to it!
  - But it can make you turn knobs to make an image that should be linear to be linear
    - System Preferences --> Displays --> Color --> Calibrate
    - Select “expert mode”

Gamma calibration

If you have access to a Mac, then you can play with this under System Preferences --> Displays --> Color --> Calibrate
(may need to select “expert”)

You should be able to explain why matching the brightness of the middle gray object compared to the black and white stripes seen from a distance can help adjust a monitor so its output is linear.

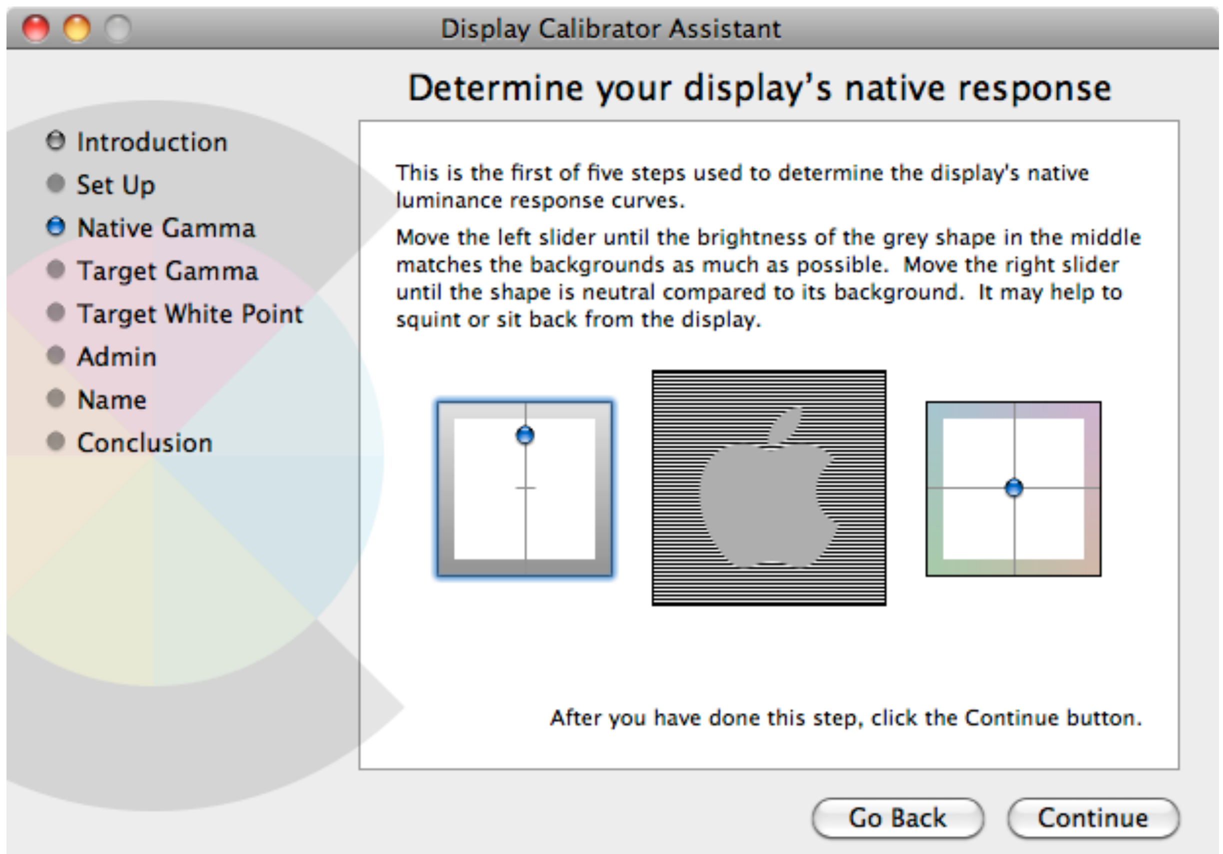


Image Formation (deluxe version)

The response of an image capture system to a light signal $L(\lambda)$ associated with a given pixels is modeled by

$$G^{(k)} = F^{(k)}(C^{(k)}) = F^{(k)}\left(b^{(k)} + \underbrace{\int L(\lambda) \; S^{(k)}(\lambda) \; d\lambda}_{\text{from before}} \right)$$

where $S^{(k)}(\lambda)$ is the sensor response function for the k^{th} channel and $b^{(k)}$ is the k^{th} channel response to black.

$S^{(k)}(\lambda)$ includes the contributions due to the aperture, focal length, sensor position in the focal plane.

$F^{(k)}$ accounts for typical non-linearities such as gamma.