Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2020-2021

Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo, object instance recognition

4. **Object Recognition: high – level vision**

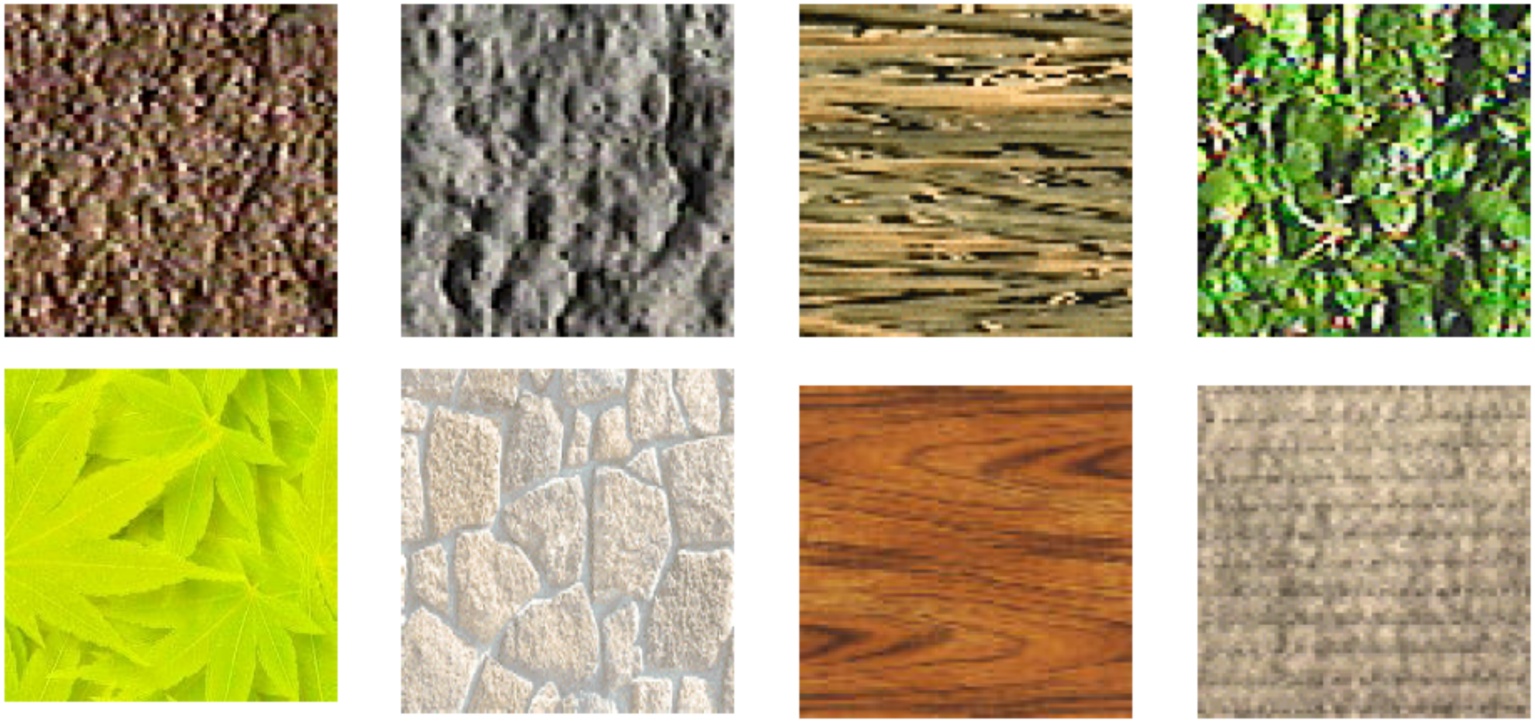
   Object classification, object detection, part based models, bovw models

5. **Video understanding**

   Object tracking, background subtraction, motion descriptors, optical flow

Texture

Texture



- Segmenting out texels (texture elements) is difficult or impossible in real images.
- Numeric quantities or statistics that describe a texture can be computed from the grayscale (or colors) images alone.
- This approach is less intuitive, but is computationally efficient.
- It can be used for both classification and segmentation.

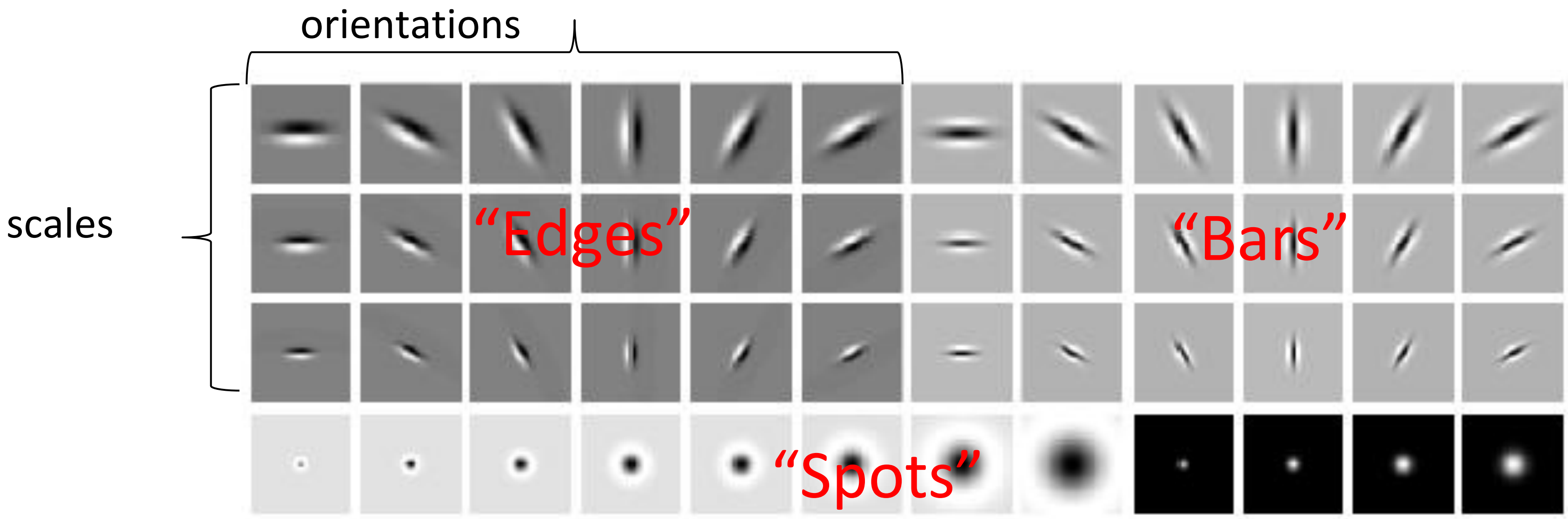
Alternative approach

Signal-processing-based algorithms use texture filters applied to the image to create filtered images from which texture features are computed.

Textures are made up of repeated local patterns, so:

- Find the patterns
  - Use filters that look like patterns (spots, bars, raw patches…)
  - Consider magnitude of response
- Describe their statistics within each local window, e.g.,
  - Mean, standard deviation
  - Histogram
  - Histogram of "prototypical" feature occurrences

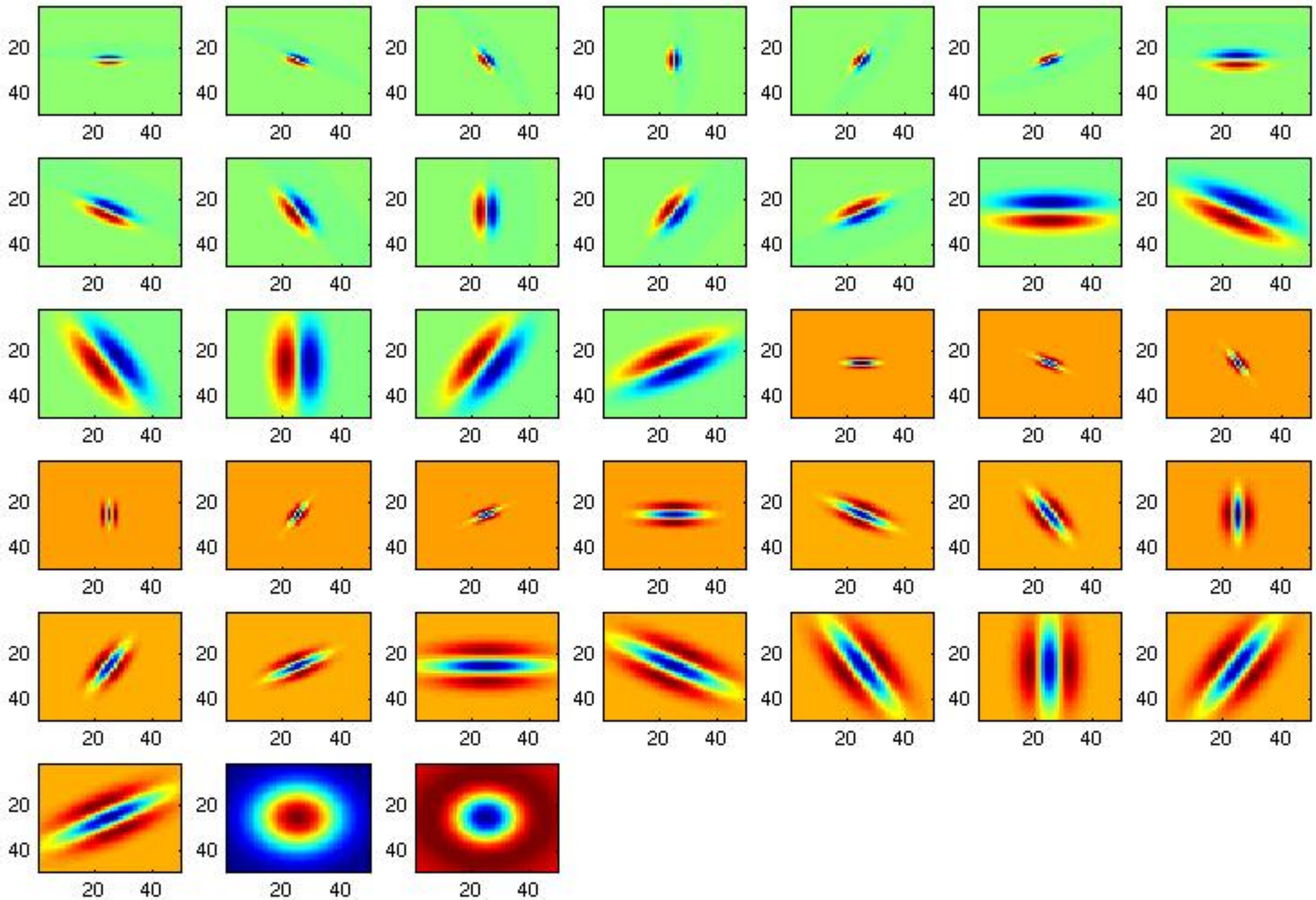
Filter banks



- What filters to put in the bank?
  - Typically we want a combination of scales and orientations, different types of patterns.

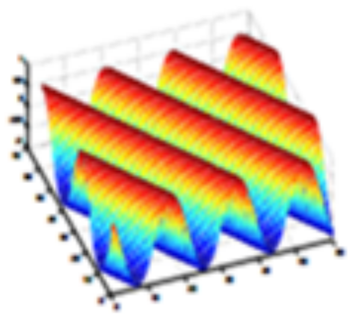
http://www.robots.ox.ac.uk/~vgg/research/texclass/filters.html

Filter bank



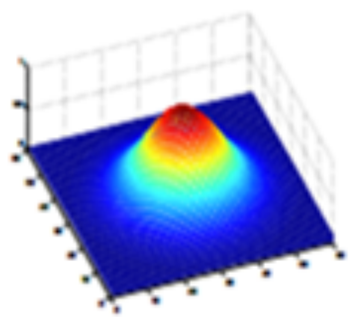
Slide credit: Kristen Grauman

Gabor Filters



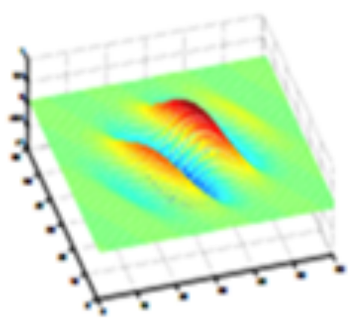
(a)

A Sinusoid oriented 30^0 with X-axis



(b)

A 2-D Gaussian

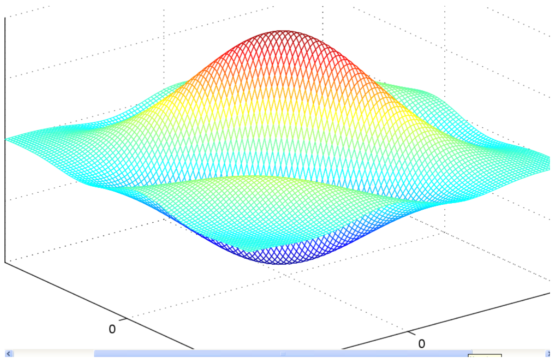


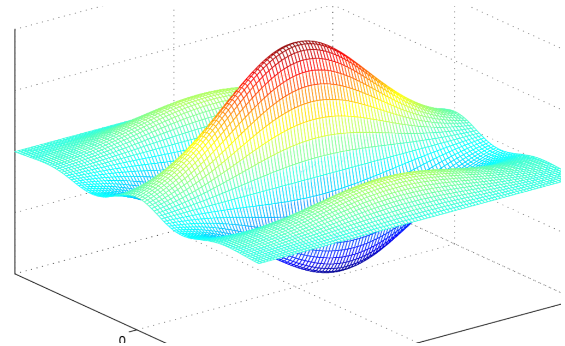
(c)

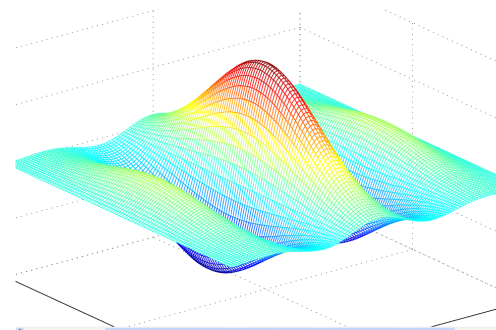
The corresponding 2-D Gabor filter

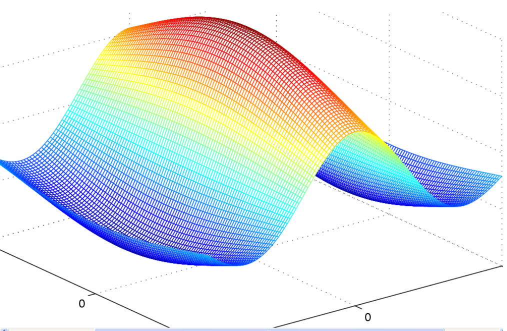
Gabor Filters

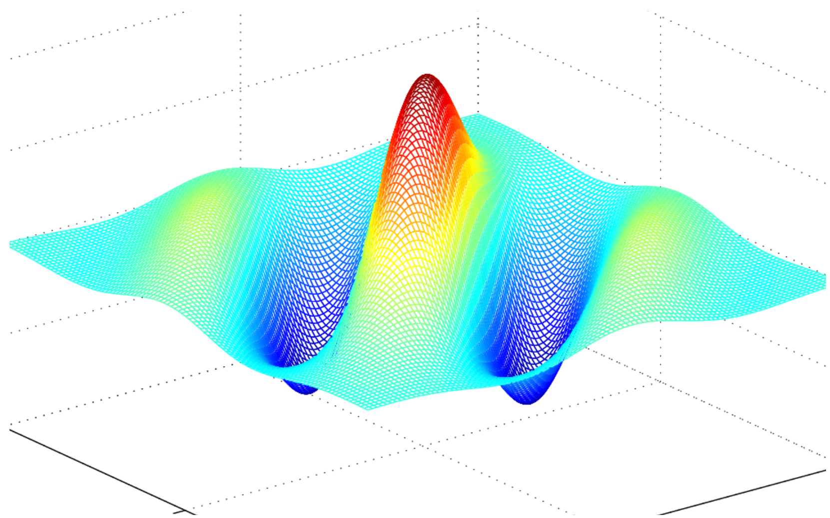
- Wavelets at different frequencies and different orientations











Gabor filter values visualization by Juergen Mueller.

variable naming conventions as used by http://en.wikipedia.org/wiki/Gabor_filter.
Red: positive values. Blue: negative values

1. Wavelength is changed.
1. wavelength (lambda) : 5.00
2. orientation (theta) : 0.79
3. gaussvar (sigma) : 20.00
4. phaseoffset (phi) : 0.00
5. aspectratio (gamma) : 0.50

min gabor val : -0.91
max gabor val : 1.00

Color

Important cue for recognition

Important cue for segmentation

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts

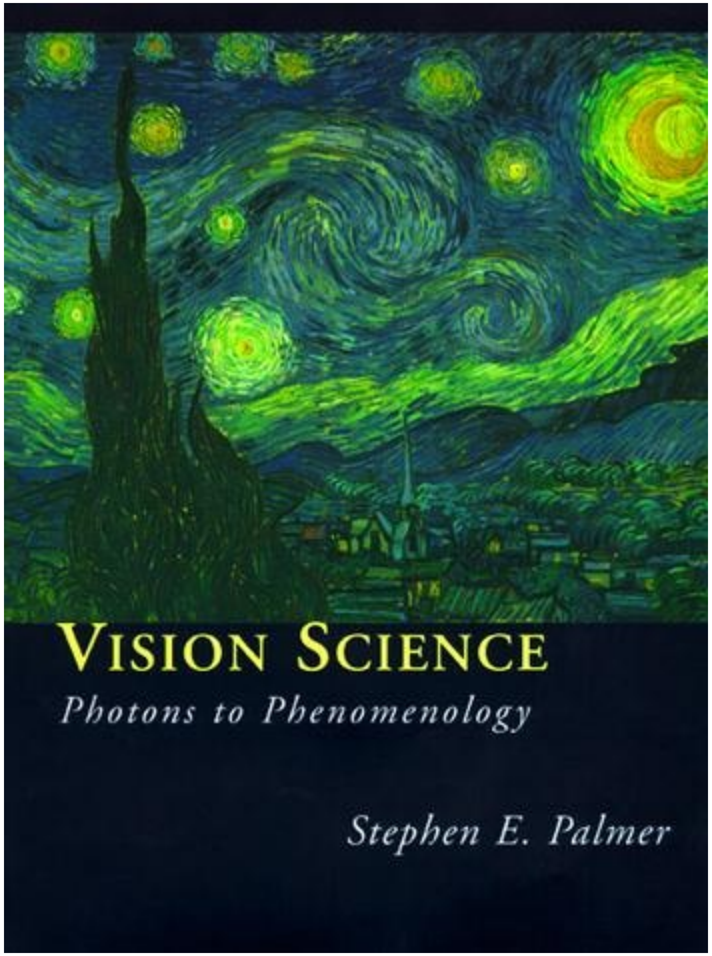
Carsten Rother[*] Vladimir Kolmogorov[†] Andrew Blake[‡]
Microsoft Research Cambridge, UK

Figure 1: **Three examples of GrabCut.** The user drags a rectangle loosely around an object. The object is then extracted automatically.

What is color?

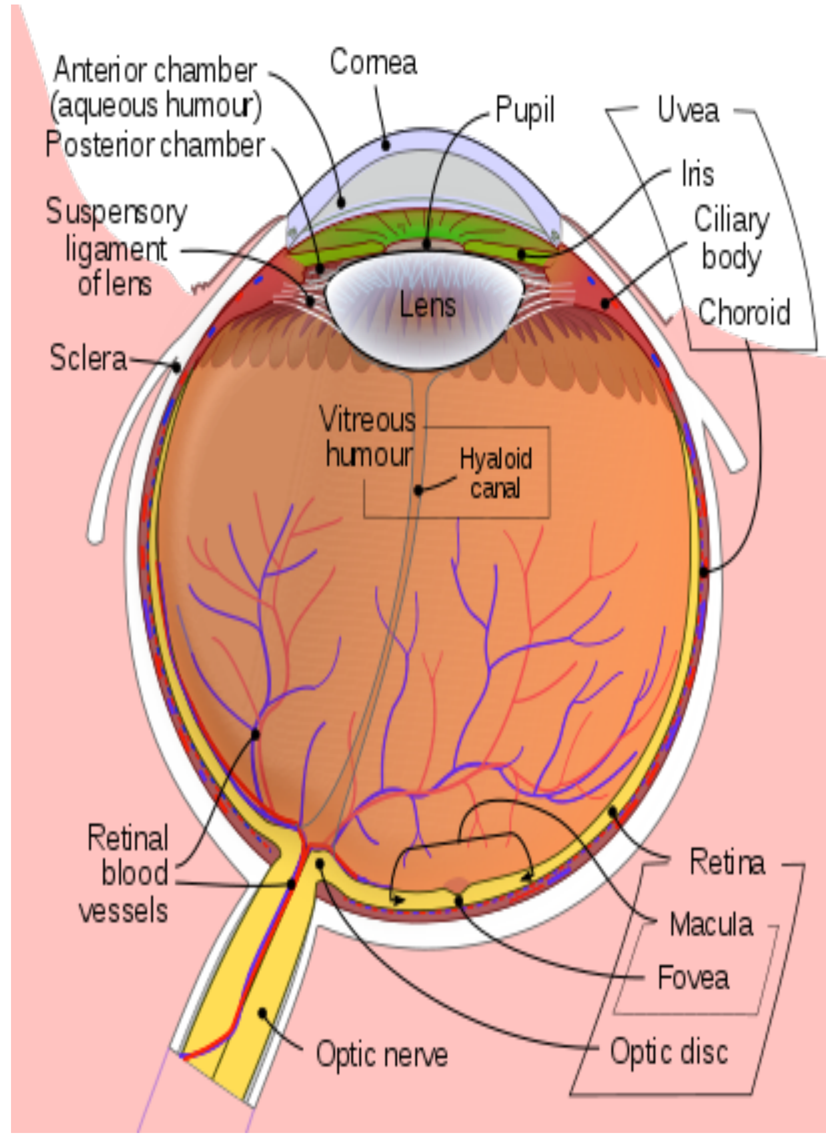
- Color is the result of interaction between physical light in the environment and our visual system

- Color is a psychological property of our visual experiences when we look at objects and lights,
  not a physical property of those objects or lights
  (S. Palmer, *Vision Science: Photons to Phenomenology*)

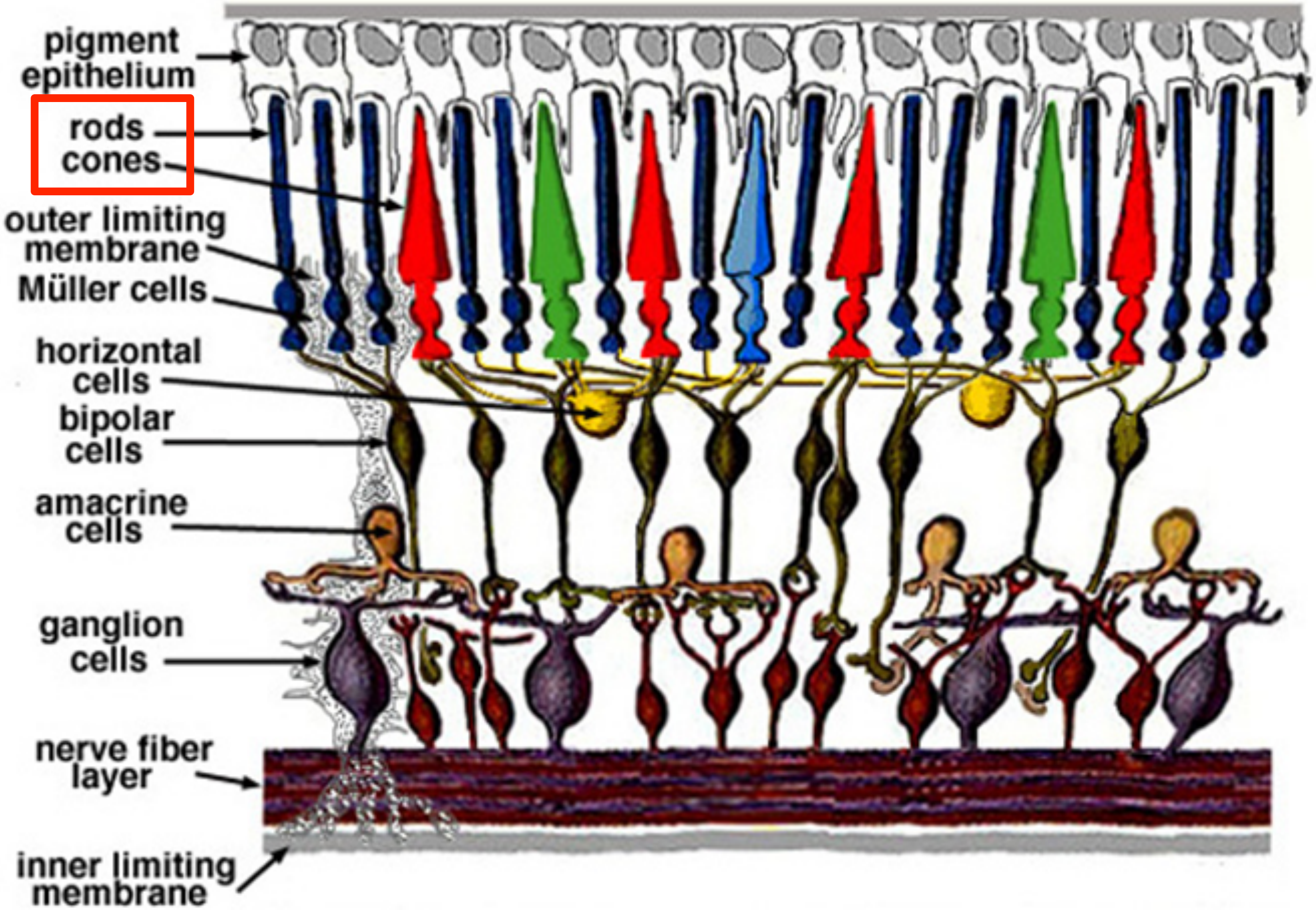


So how do human eyes work?

- Complex!
- Light passes through
  - Cornea, humours, lens refract light to focus
- Hit the retina
- Absorbed by photosensitive cells
- Info transmitted through optic nerve, processed by visual cortex

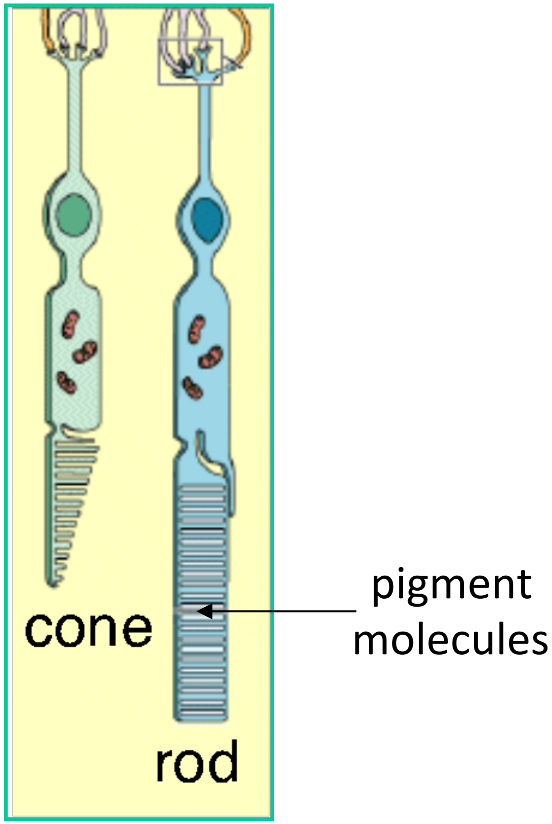


Retina



Simple diagram of the organization of the retina.

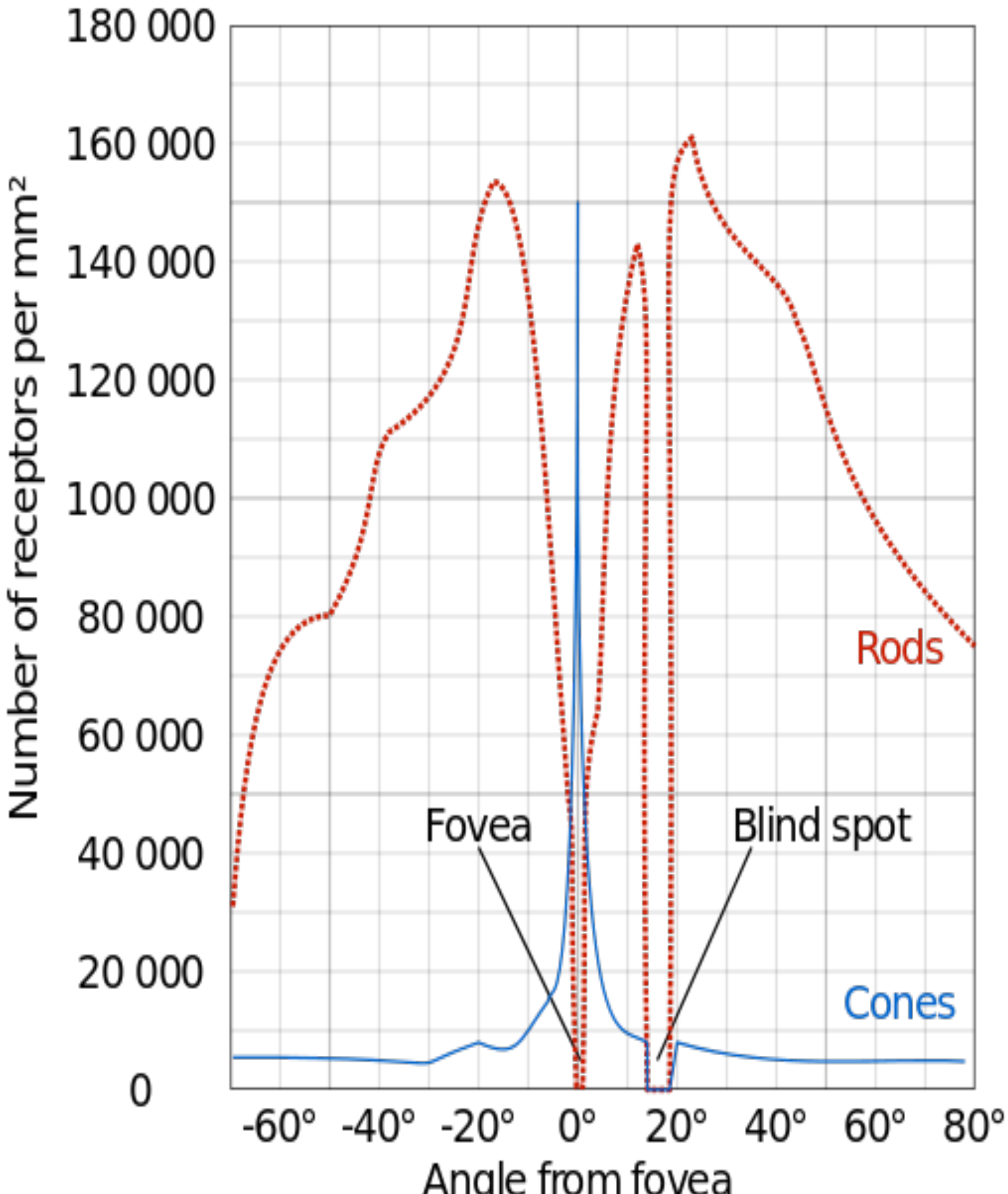
Photoreceptive cells - rods and cones

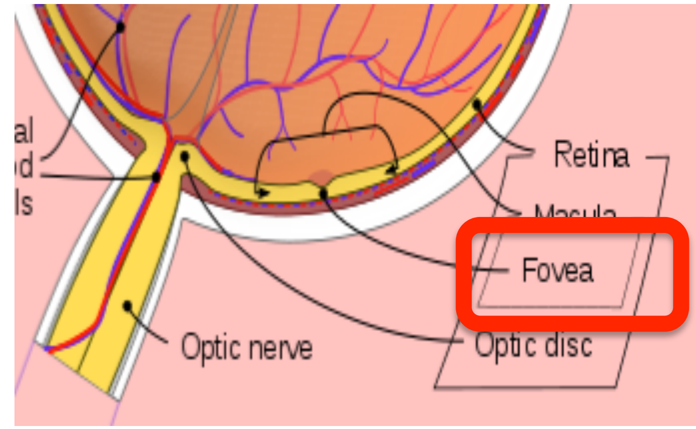


- Cones (RO: conuri)
  - around 6 million
  - need many photons to activate, bright light
  - fast response time
  - fine details
  - fast changes over time
  - responsible for most daytime vision (color vision)
  - mostly packed into one region: Fovea

- Rods (RO: bastonașe)
  - ~120 million
  - sensitive to 1 photon
  - can pool responses
  - slow response time
  - only operate in low-light conditions
  - saturate quickly in lots of light
  - take ~7 minutes to adjust (night vision)

Fovea: where it’s all happening



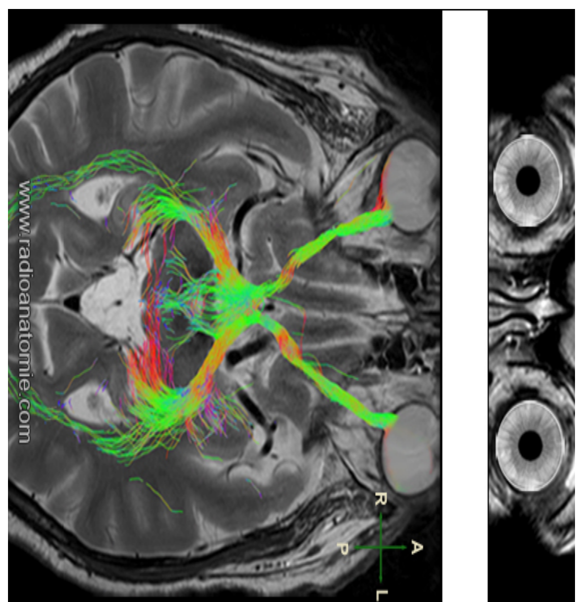


Fovea

- Small circle on the retina, 1.5mm^2
- Densely packed with cones
  - 200,000 cones/mm2
- Highest visual acuity
- Reading: move your eyes so the text is centered in fovea
- Responsible for sharp central vision;

The brain and vision

- Enormous processing power devoted to vision
- Visual cortex is the largest “system” in the brain
  - 30% of the cerebral cortex
  - ⅔ of the electrical activity
- Lots of processing happening “subconsciously”



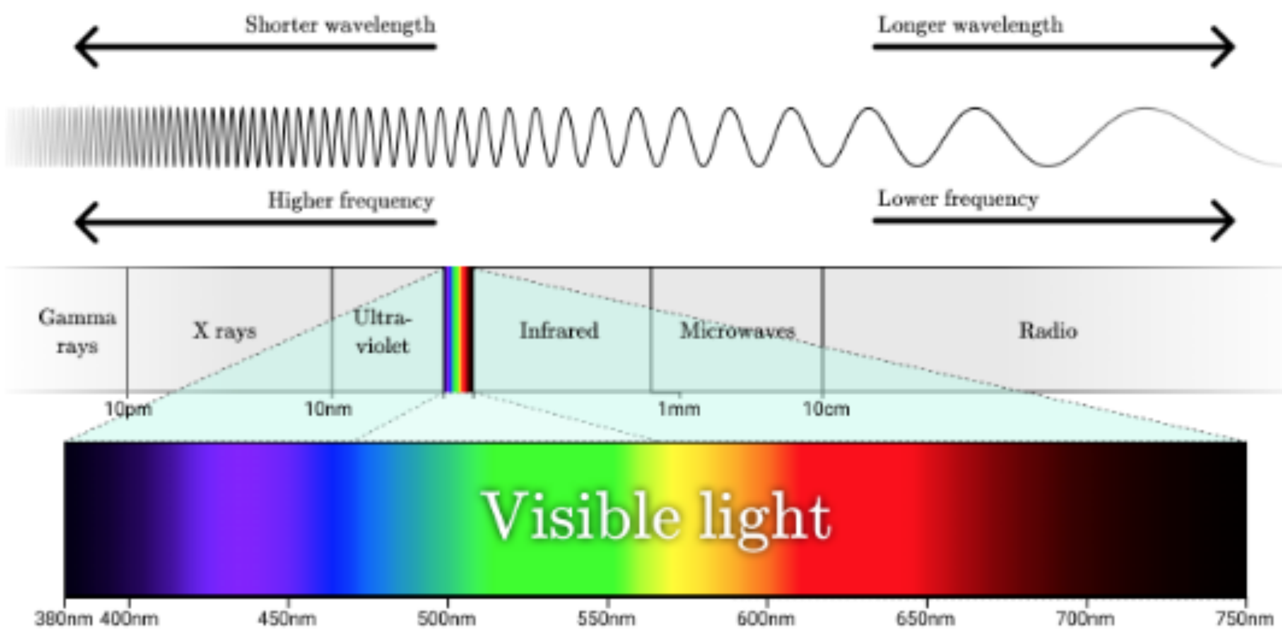
We don’t really understand vision

- Visual cortex - highly studied part of the brain
- Only rough idea of what different components do
- New discoveries in vision all the time
    - Eye uses blinking to reset its rotational orientation
    - Visual cortex can make some “high-level” decisions

What is color?

Color = the result of interaction between physical light in the environment and our visual system

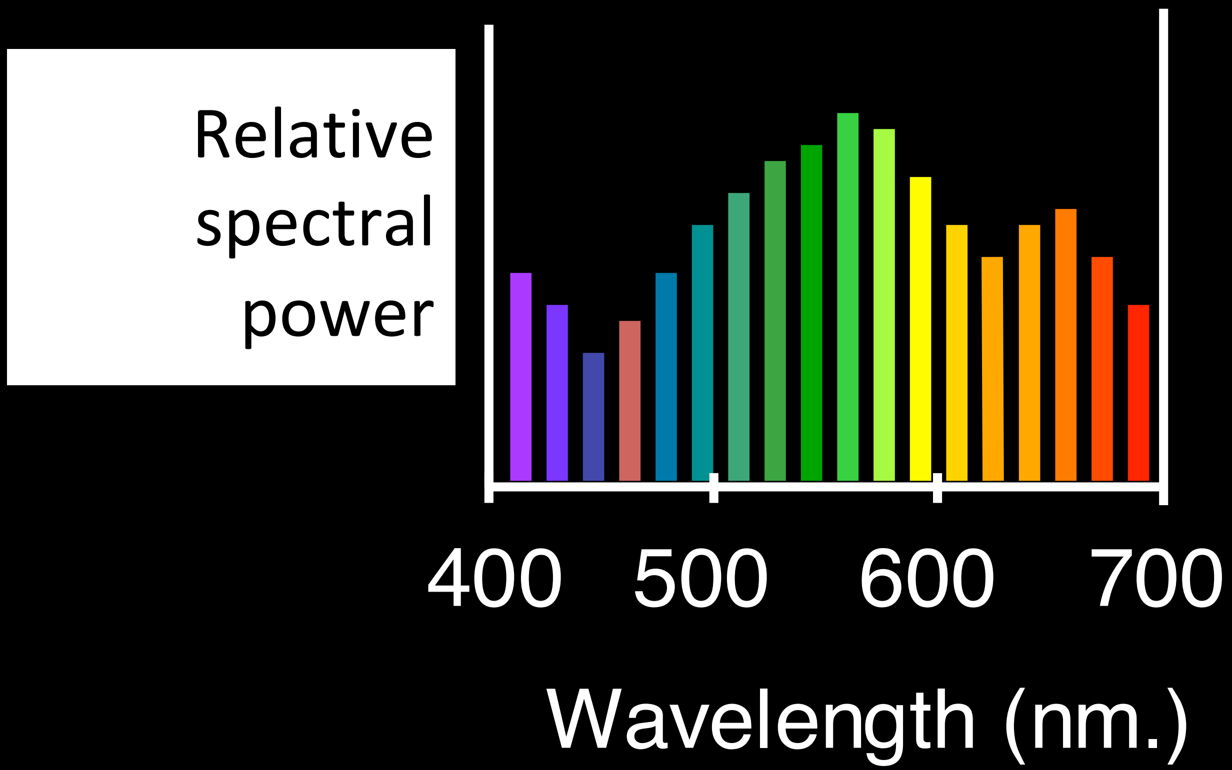
Color = electromagnetic wave detected by the eye and interpreted by the brain. The visible light is composed of a relatively narrow band of the electromagnetic spectrum. The visible spectrum is in the wavelength range: 380nm - 780nm.



The electromagnetic spectrum

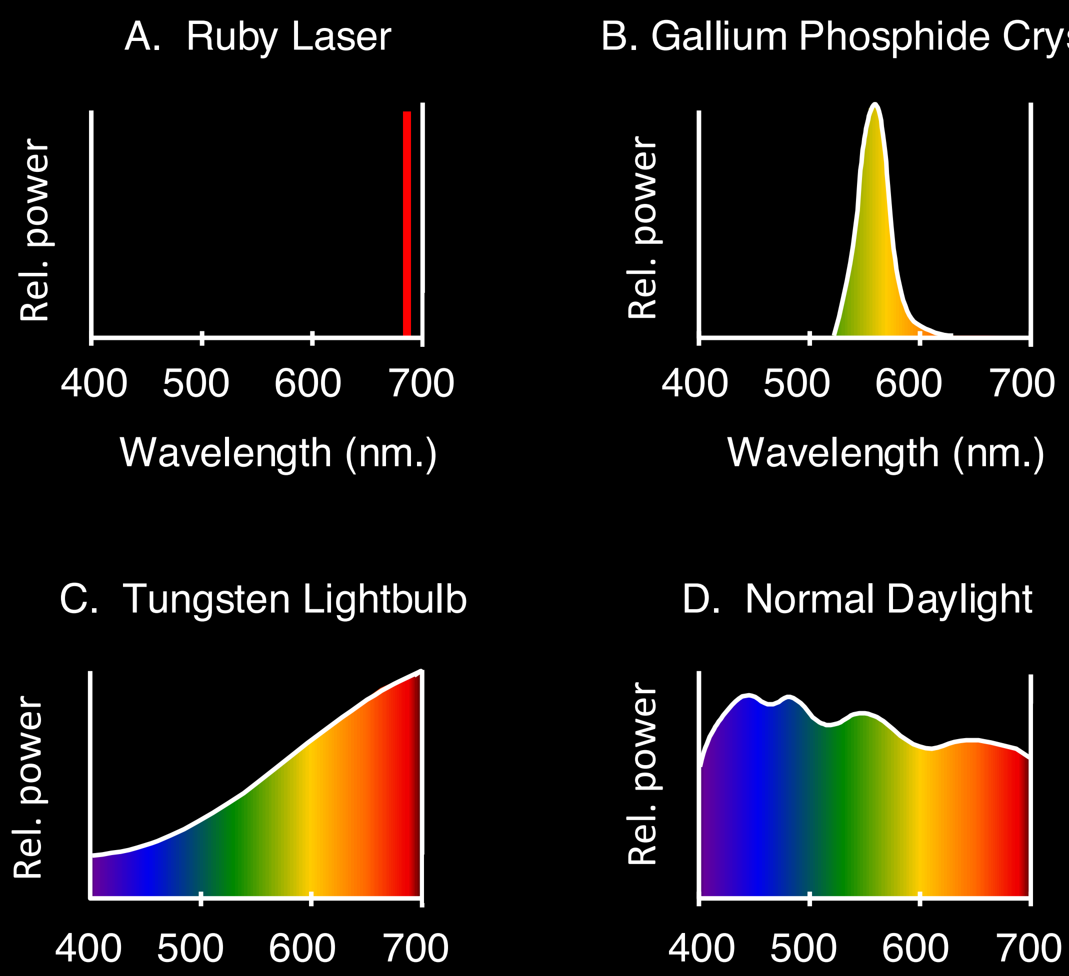
The Physics of Light

Any source of light can be completely described physically by its spectrum: the amount of energy emitted (per time unit) at each wavelength 400 - 700 nm.



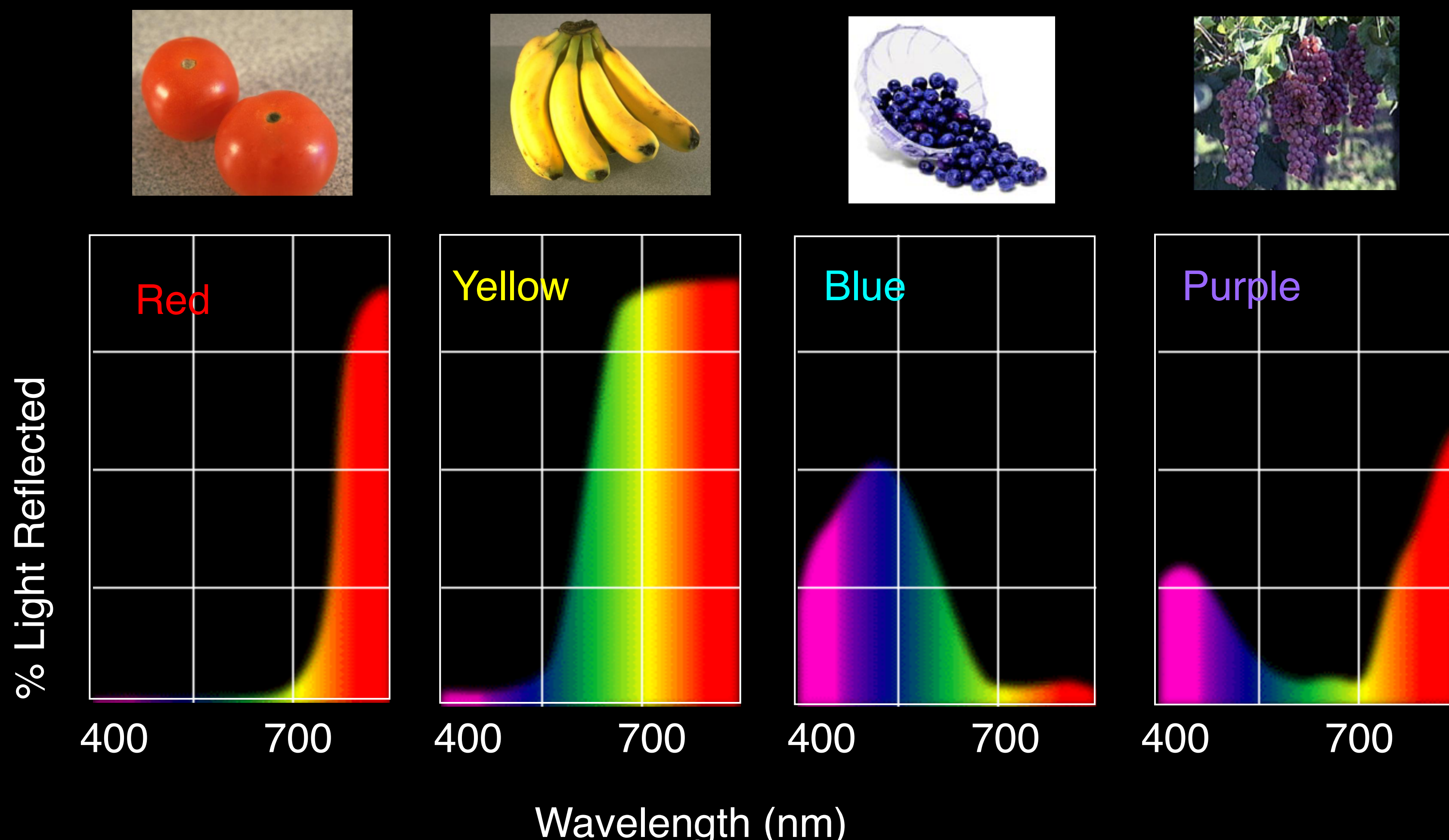
Spectra of Light Sources

Some examples of the spectra of light sources



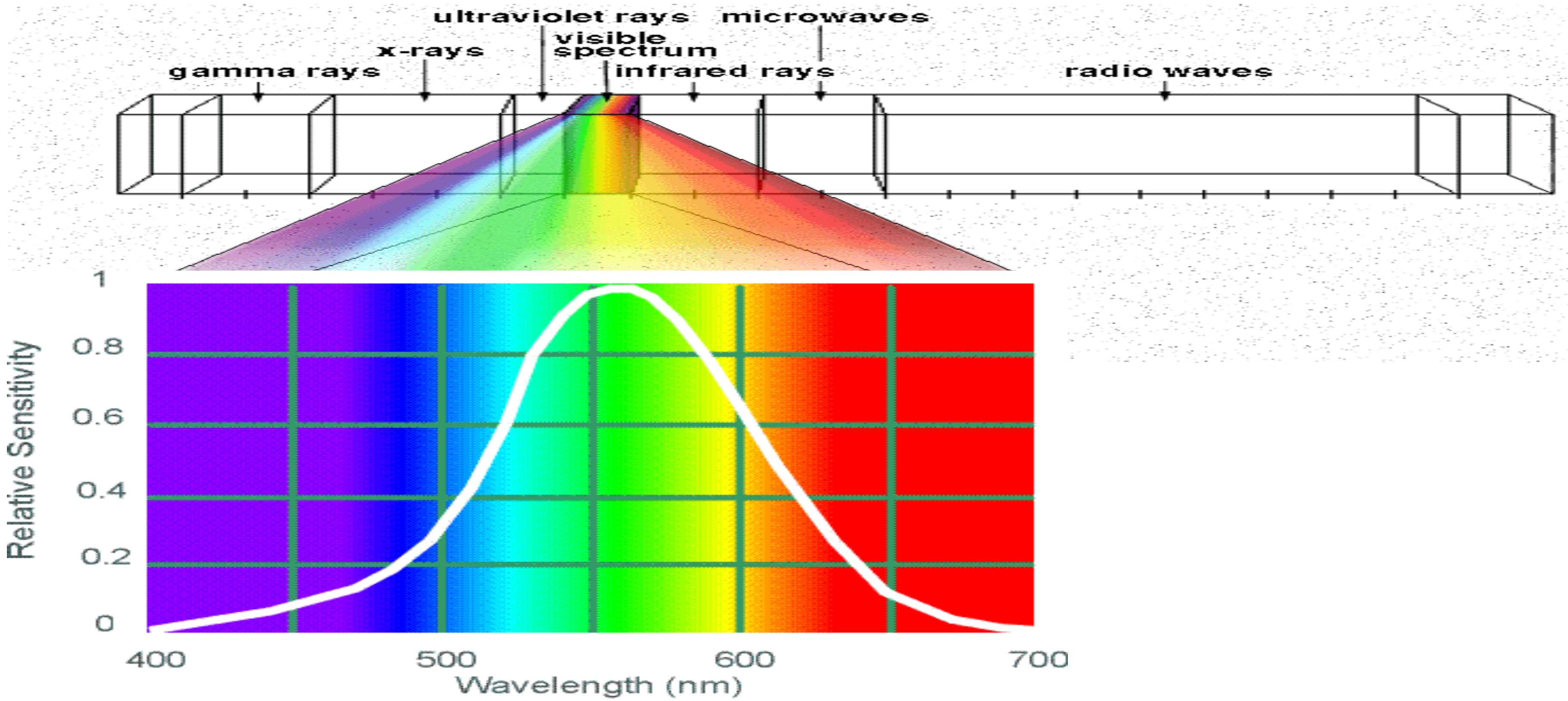
Reflectance Spectra of Surfaces

Some examples of the reflectance spectra of surfaces



Human Luminance Sensitivity Function

Light is composed of a spectrum of wavelengths. If we measure a human's eye sensitivity to every wavelength we get the standard human luminance sensitivity function



Human Luminance Sensitivity Function

Slide Credit: A. Efros

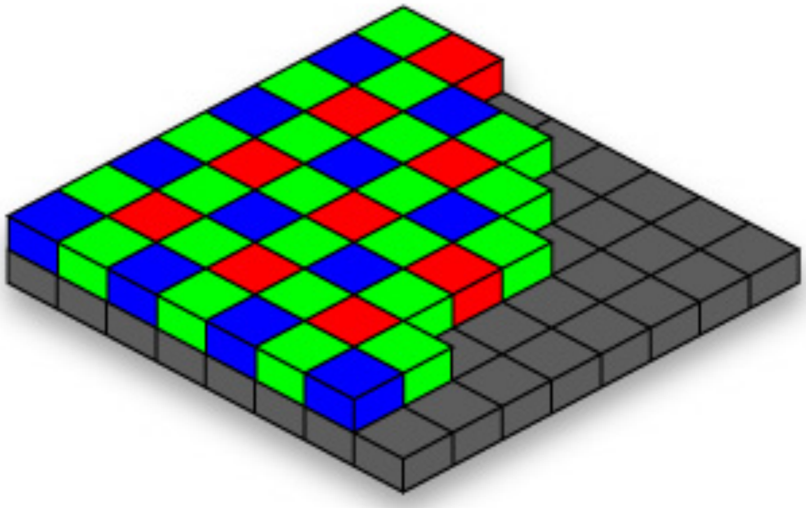
This is hard to read

This is easier to read

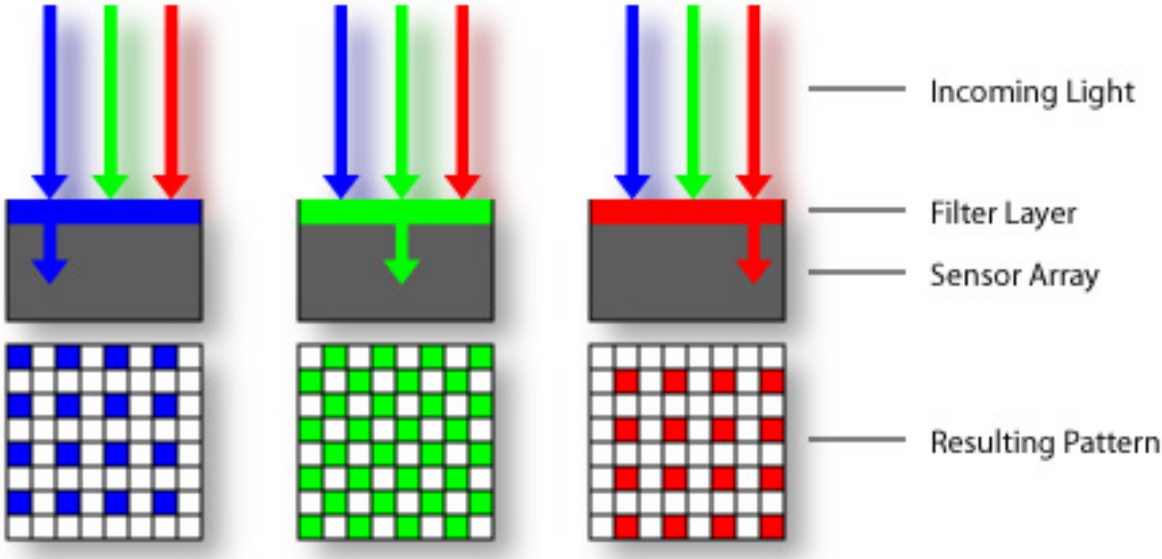
Why RGB?

If light is a spectrum, why are images RGB?

Color Sensing: Bayer Grid



- Estimate RGB at each cell from neighboring values

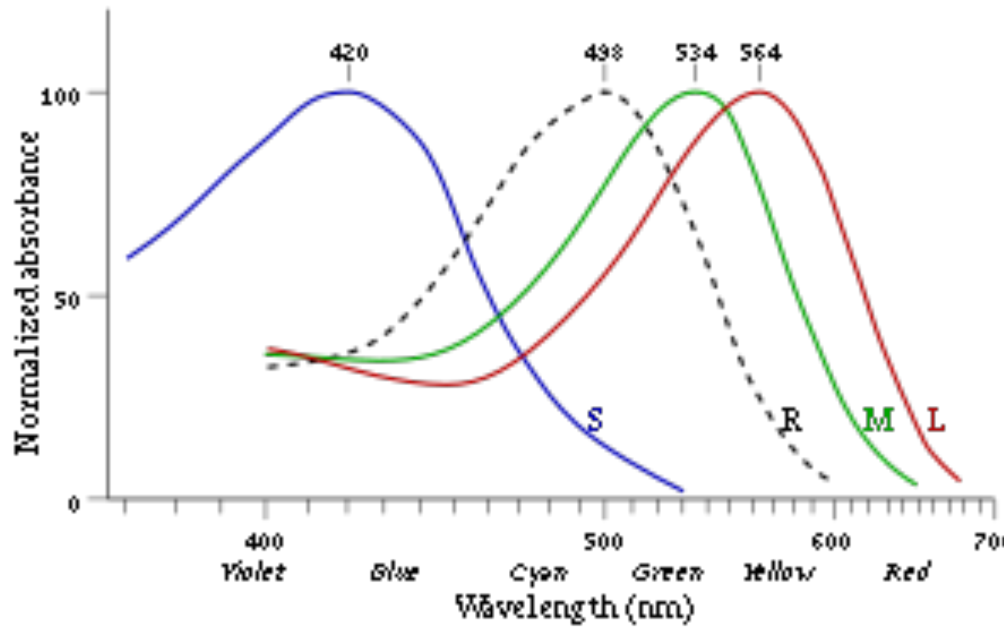


http://en.wikipedia.org/wiki/Bayer_filter

Slide credit: Steve Seitz

Photoreceptors and light

- Each receptor has a responsiveness curve
- Receptors more responsive to some wavelengths, less responsive to others
- Rods: peak around 498 nm
- Cones: 3 kinds
  - Short: peak around 420 nm
  - Medium: peak around 530 nm
  - Long: peak around 560 nm



Cones and color

Our perception of color comes from cones

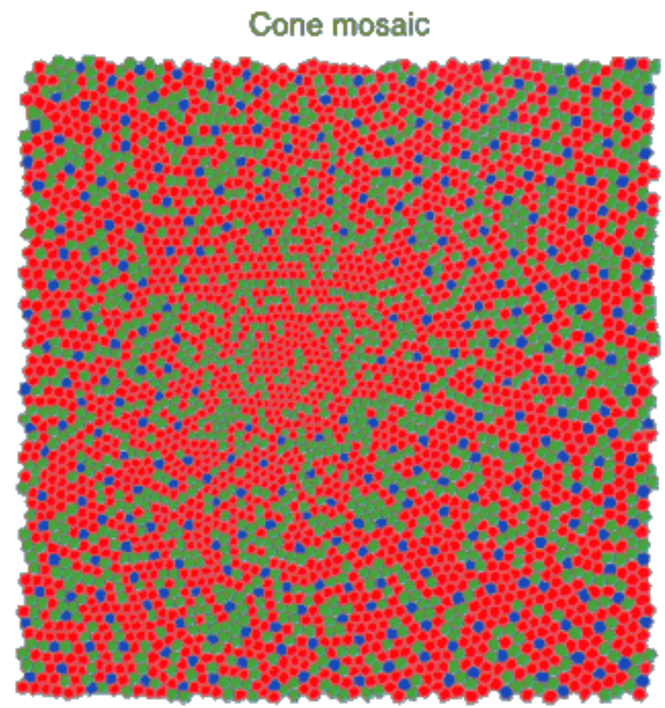
Different waveforms provoke different responses

Each cone has essentially one “output”

To calculate:

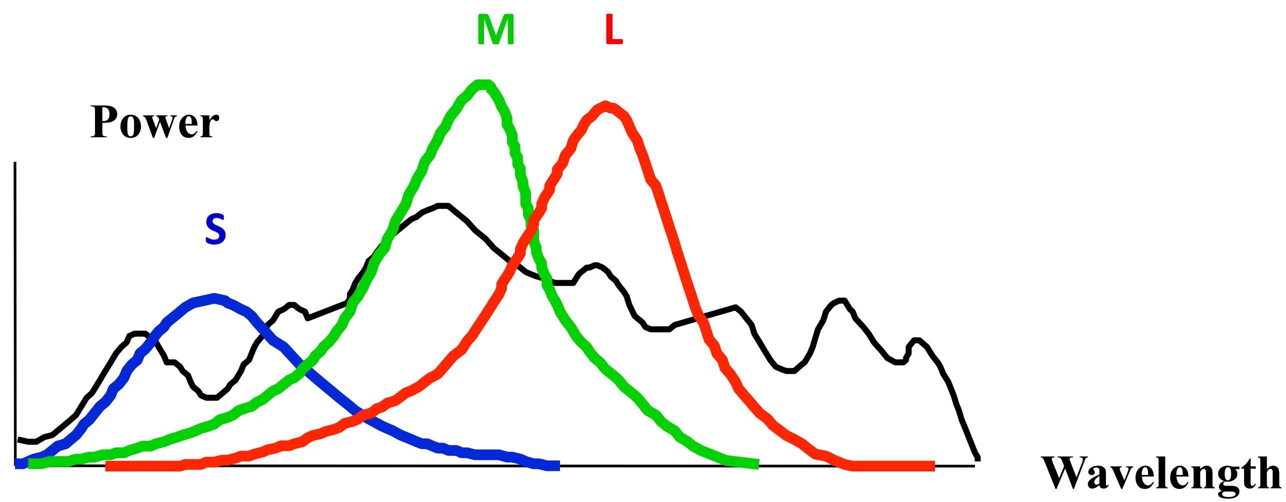
- Multiply input waveform by response curve
- Integrate area under the curve

The “color” we see is the relative activation of the 3 kinds of cones



- Ratio of Long to Medium to Short cones: approx. 10:5:1
- Almost no S cones in the center of the fovea

Color perception

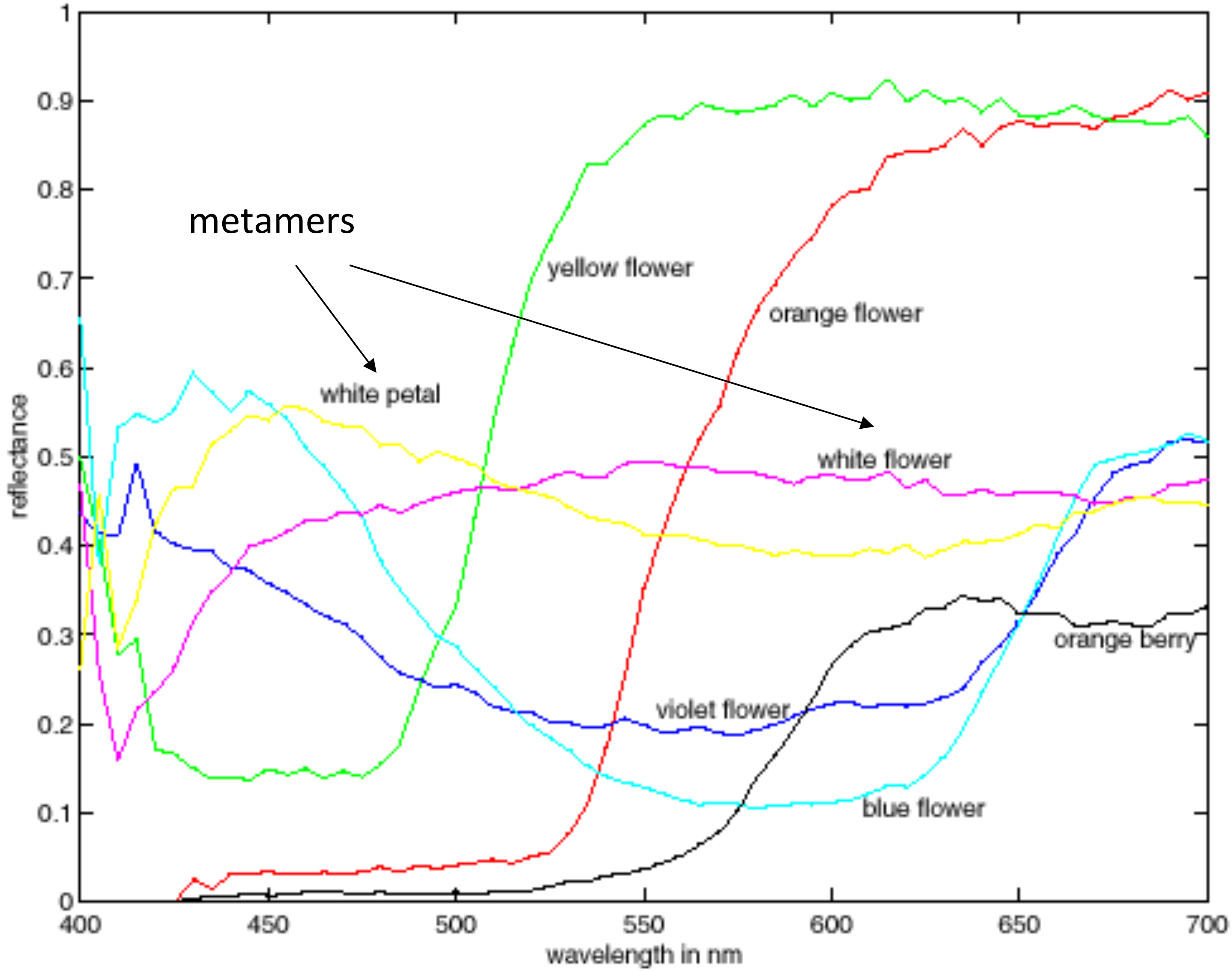


Rods and cones act as *filters* on the spectrum

- To get the output of a filter, multiply its response curve by the spectrum, integrate over all wavelengths
  - Each cone yields one number
- How can we represent an entire spectrum with three numbers? We can't! Most of the information is lost
  - As a result, two different spectra may appear indistinguishable (such spectra are known as **metamers**)

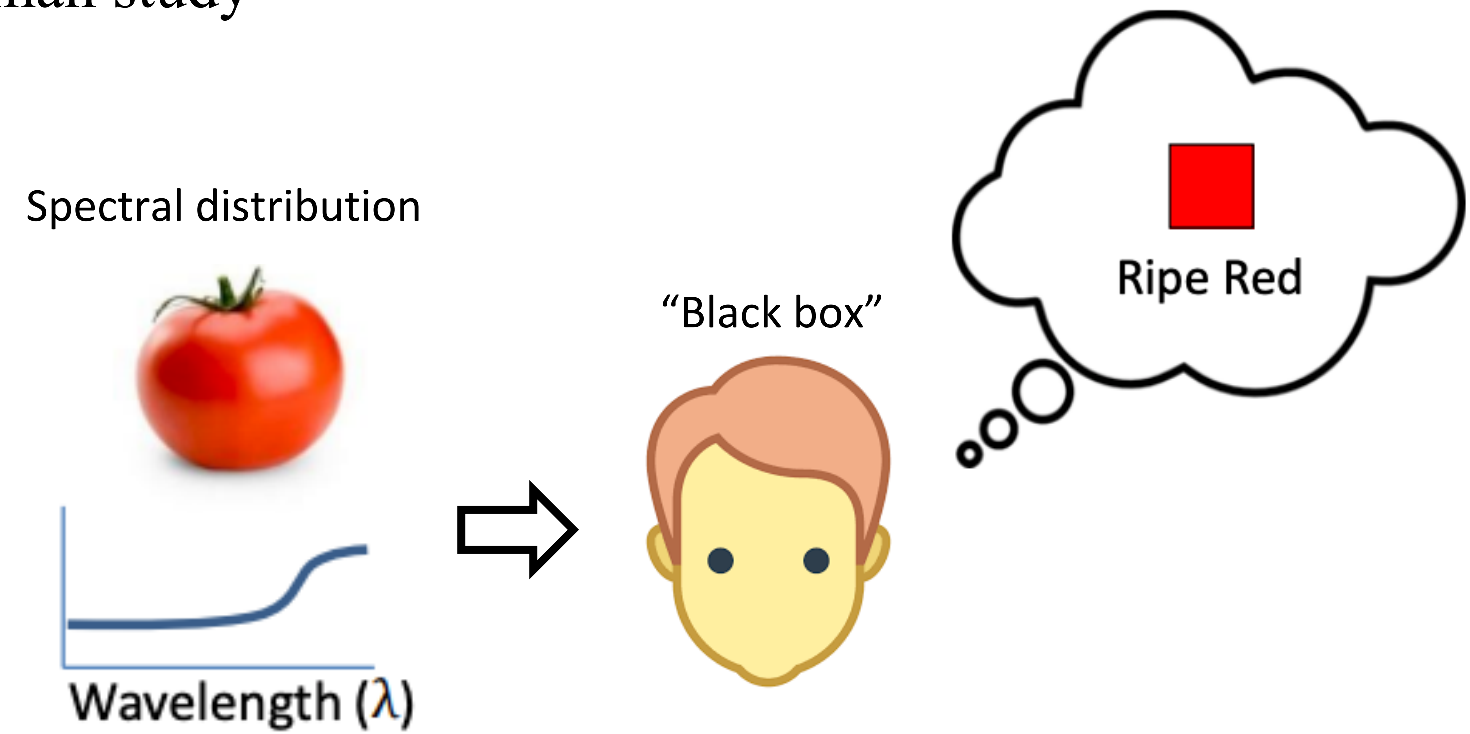
Slide credit: Steve Seitz

Spectra of some real-world surfaces



Quantifying color

- Spectral distributions go through a "black box" (human visual system) and are perceived as color
- The only way to quantify the "black box" is to perform a human study



Color matching experiments

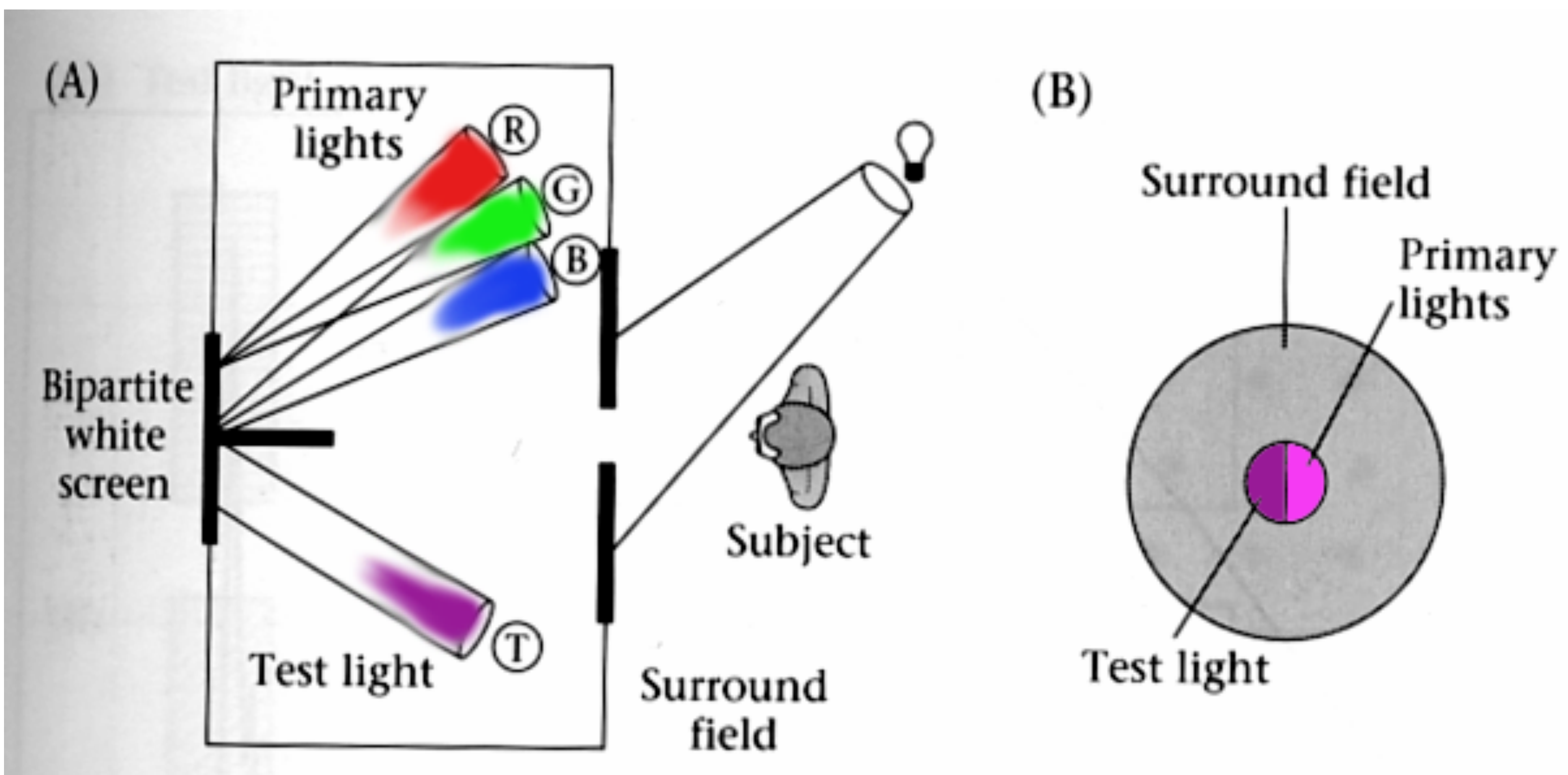
- We would like to understand which spectra produce the same color sensation in people under similar viewing conditions.

- Late 1920s experimented with colors! (and people)
- Subjects get controls to 3 “primary” lights
- Show them a light
- Subject adjusts their lights to match the given light

Color matching experiments

- Late 1920s experimented with colors! (and people)

- Subjects get controls to 3 “primary” lights
- Show them a light
- Subject adjusts their lights to match the given light



Trichromatic color theory

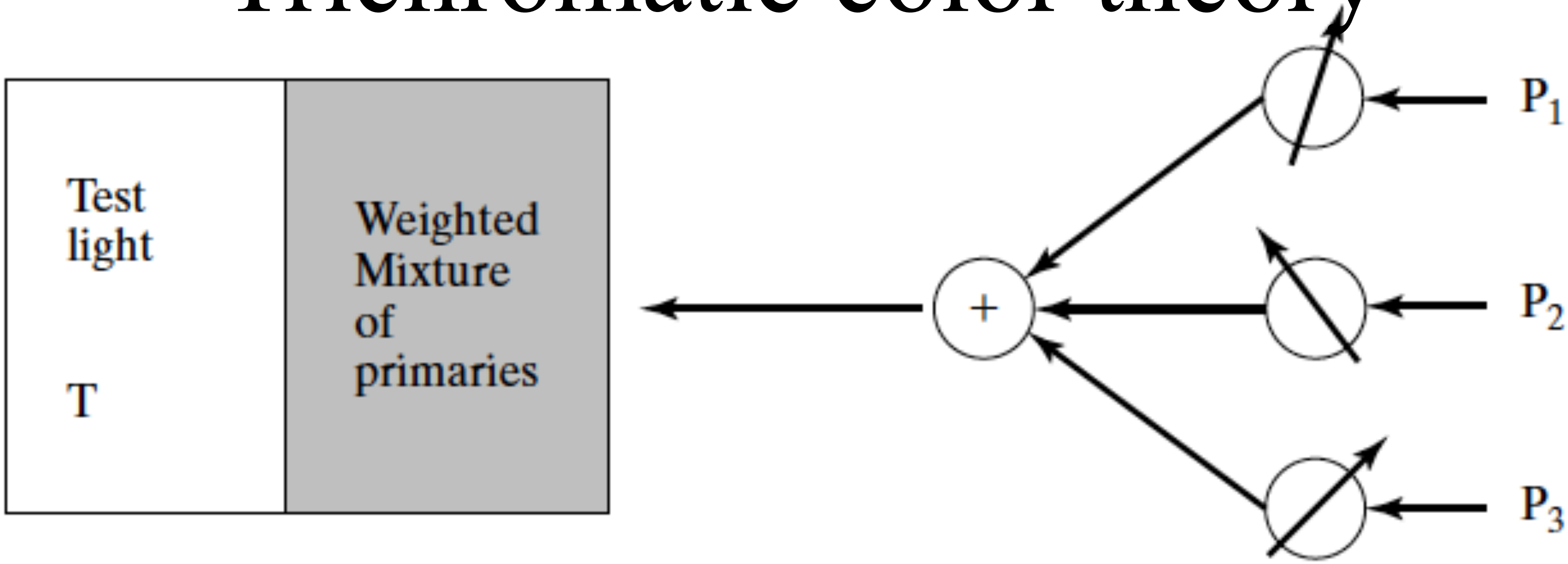
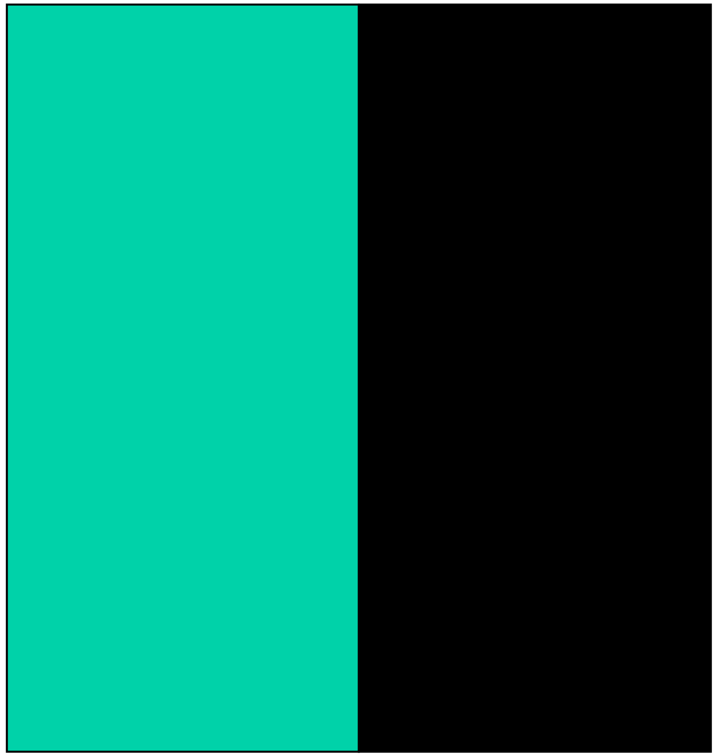
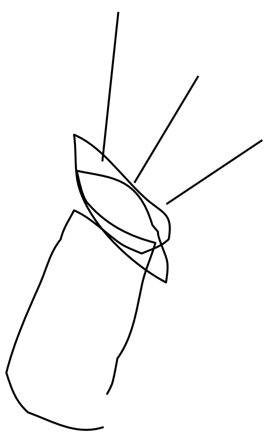


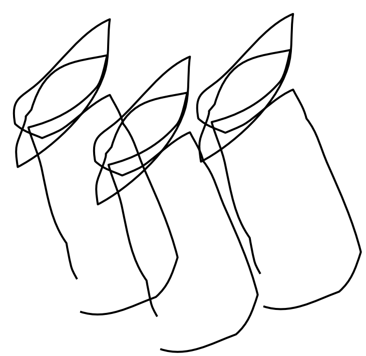
FIGURE 3.2: Human perception of color can be studied by asking observers to mix colored lights to match a test light shown in a split field. The drawing shows the outline of such an experiment. The observer sees a test light T and can adjust the amount of each of three primaries in a mixture displayed next to the test light. The observer is asked to adjust the amounts so that the mixture looks the same as the test light. The mixture of primaries can be written as $w_1 P_1 + w_2 P_2 + w_3 P_3$; if the mixture matches the test light, then we write $T = w_1 P_1 + w_2 P_2 + w_3 P_3$. It is a remarkable fact that for most people three primaries are sufficient to achieve a match for many colors, and three primaries are sufficient for all colors if we allow subtractive matching (i.e., some amount of some of the primaries is mixed with the test light to achieve a match). Some people require fewer primaries. Furthermore, most people choose the same mixture weights to match a given test light.

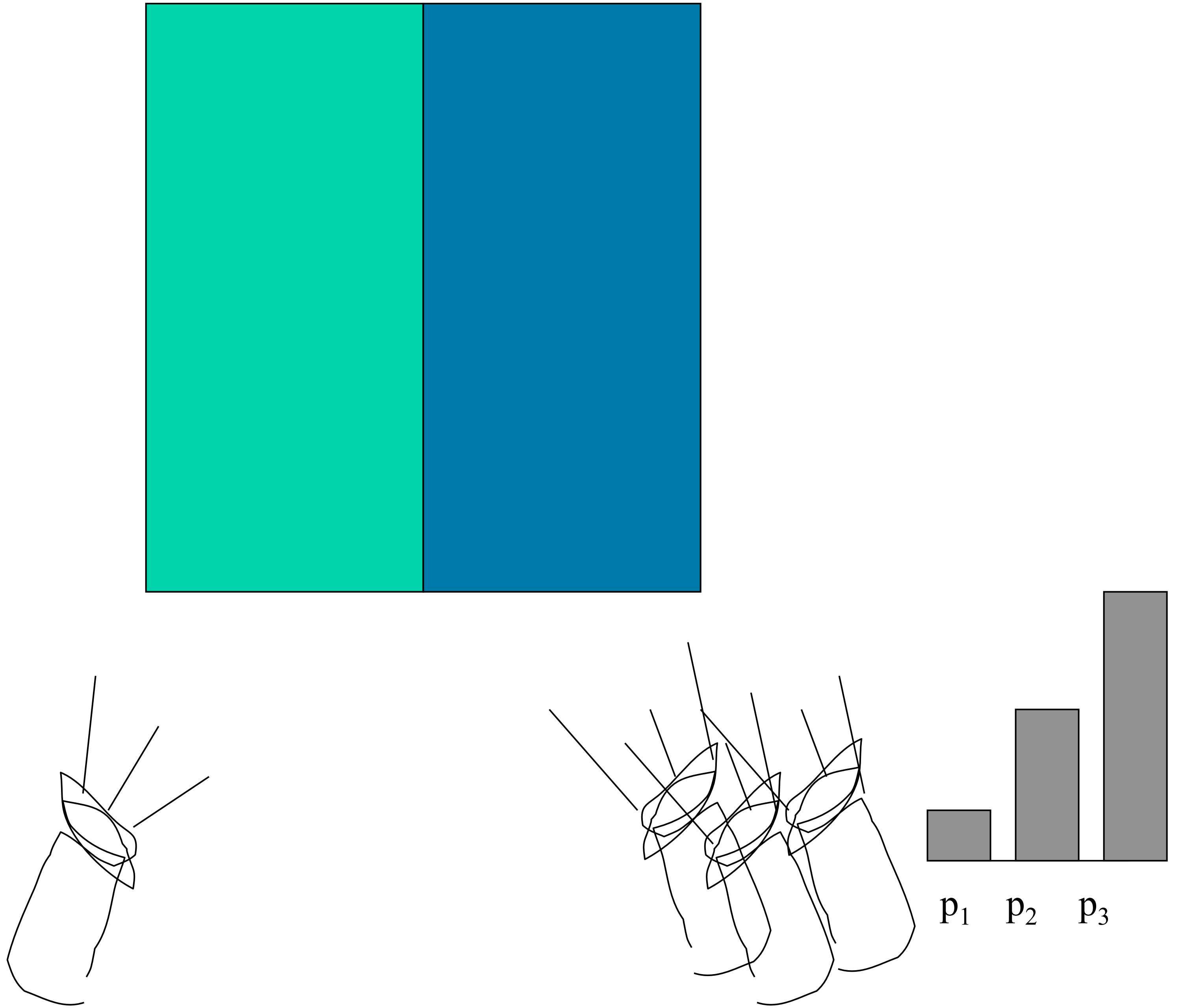
Trichromatic color theory

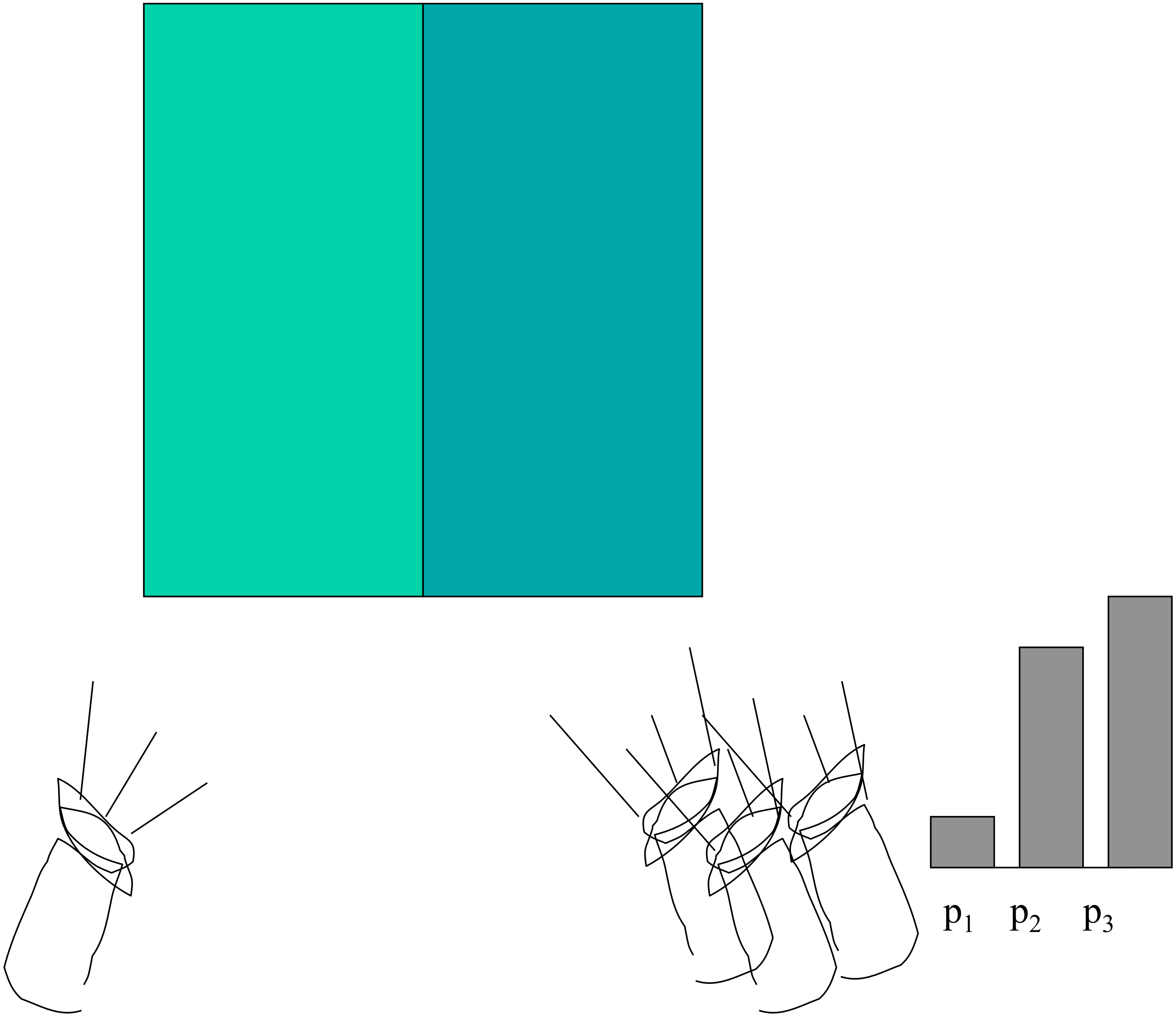
- In color matching experiments, most people can match any given light with three primaries
  - Primaries must be *independent*
- For the same light and same primaries, most people select the same weights
  - Exception: color blindness
- Trichromatic color theory
  - Three numbers seem to be sufficient for encoding color
  - Dates back to 18th century (Thomas Young)



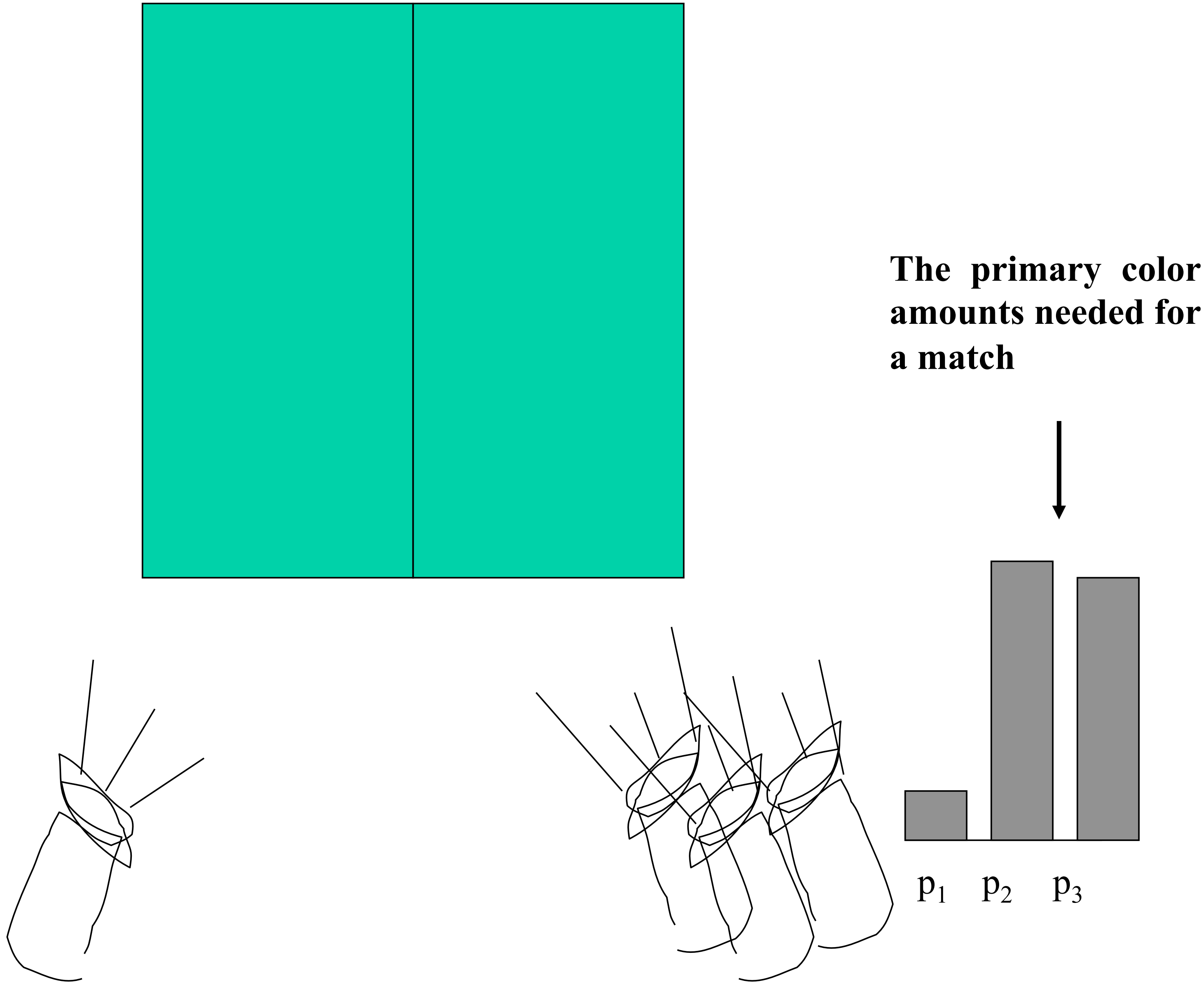




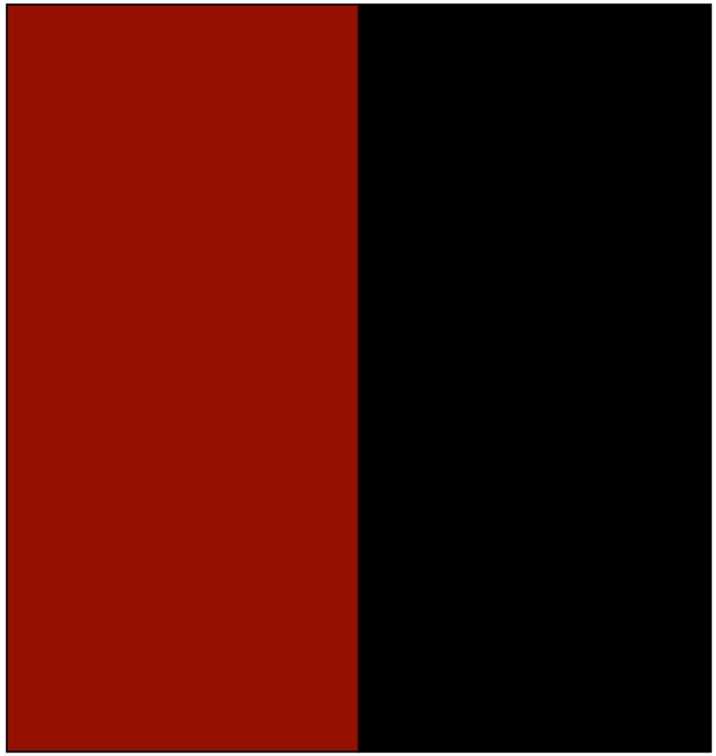

p_1
p_2
p_3

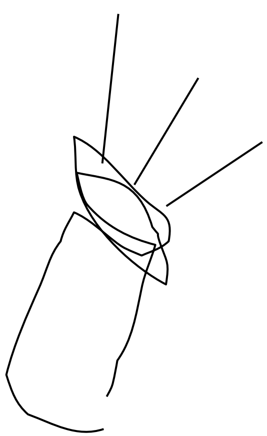


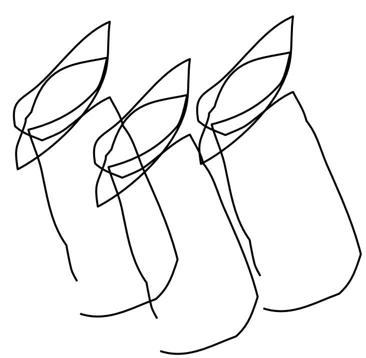
p_1 p_2 p_3

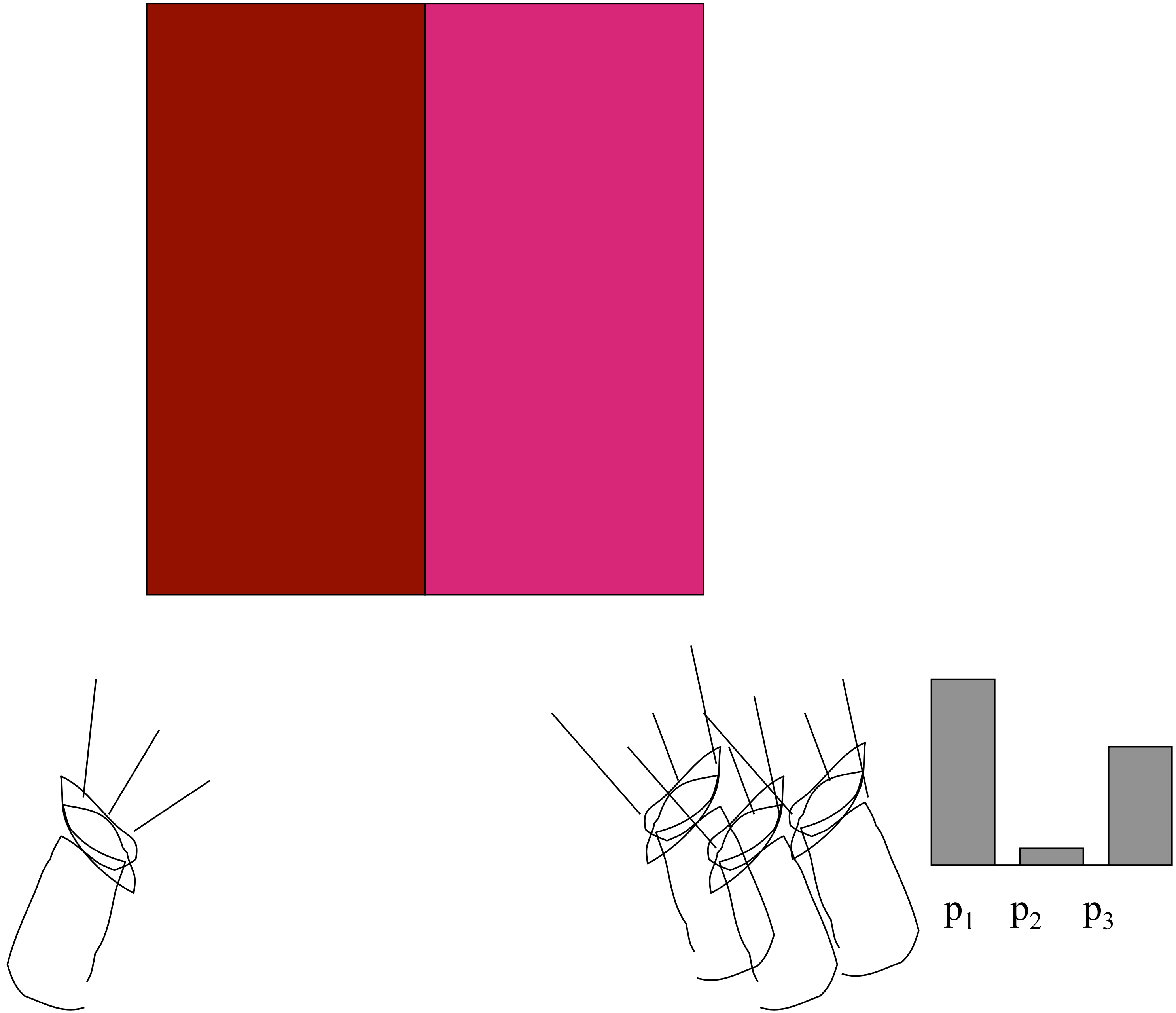


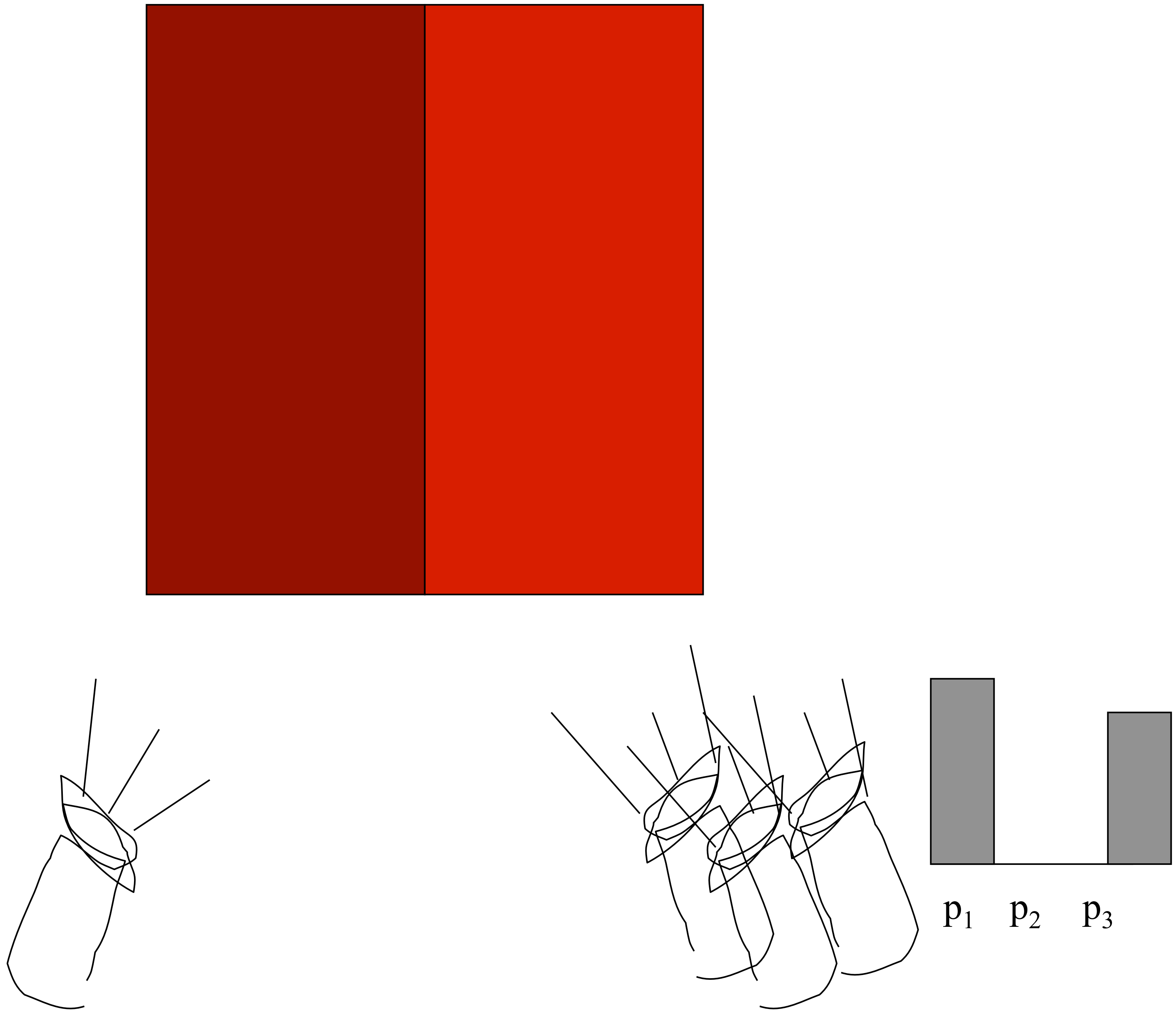
The primary color amounts needed for a match
p_1 p_2 p_3





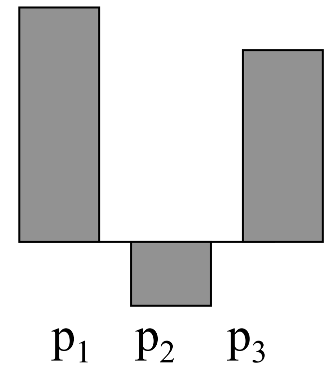



p_1
p_2
p_3

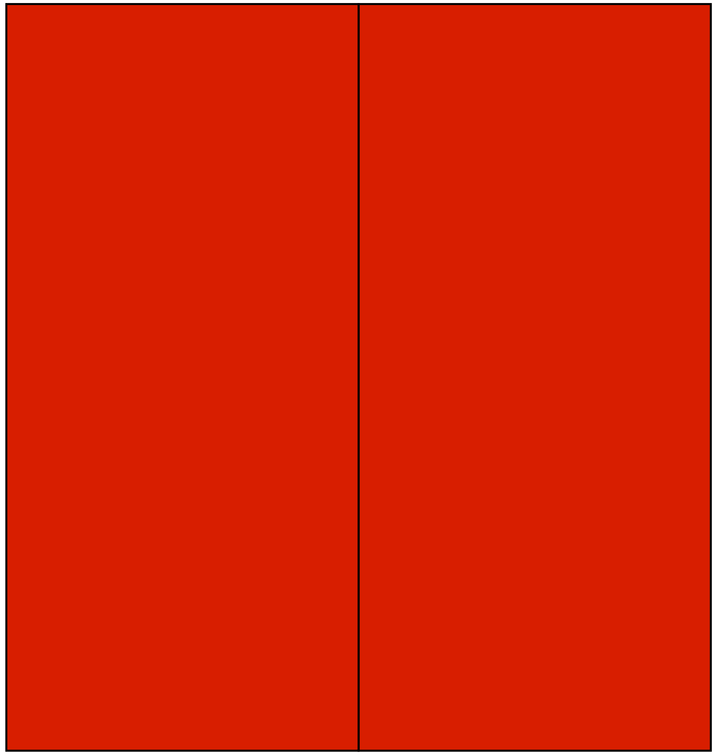


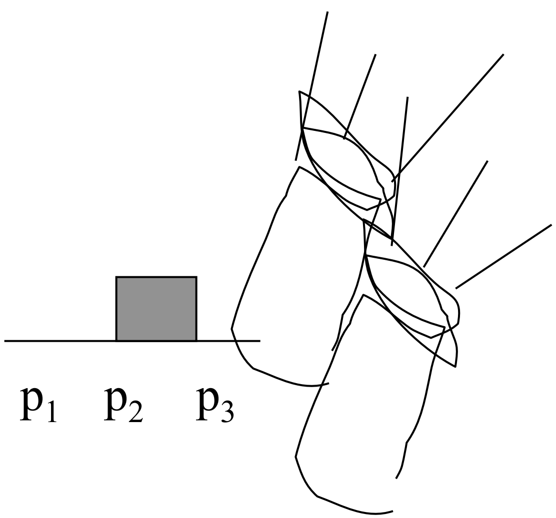
p_1 p_2 p_3

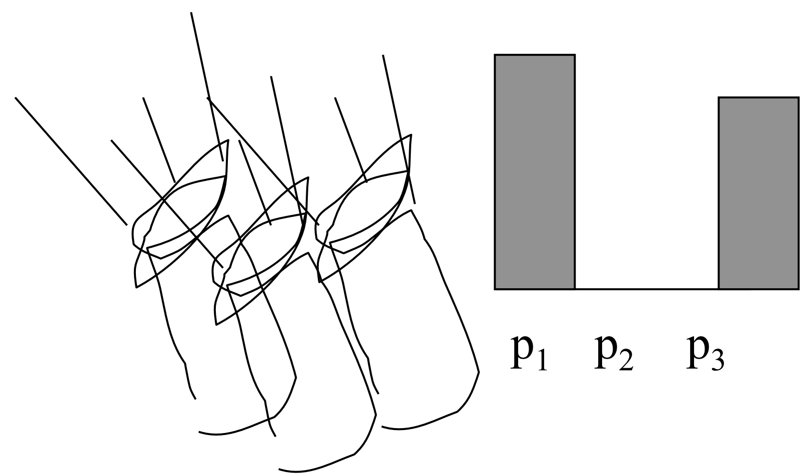
The primary color amounts needed for a match:



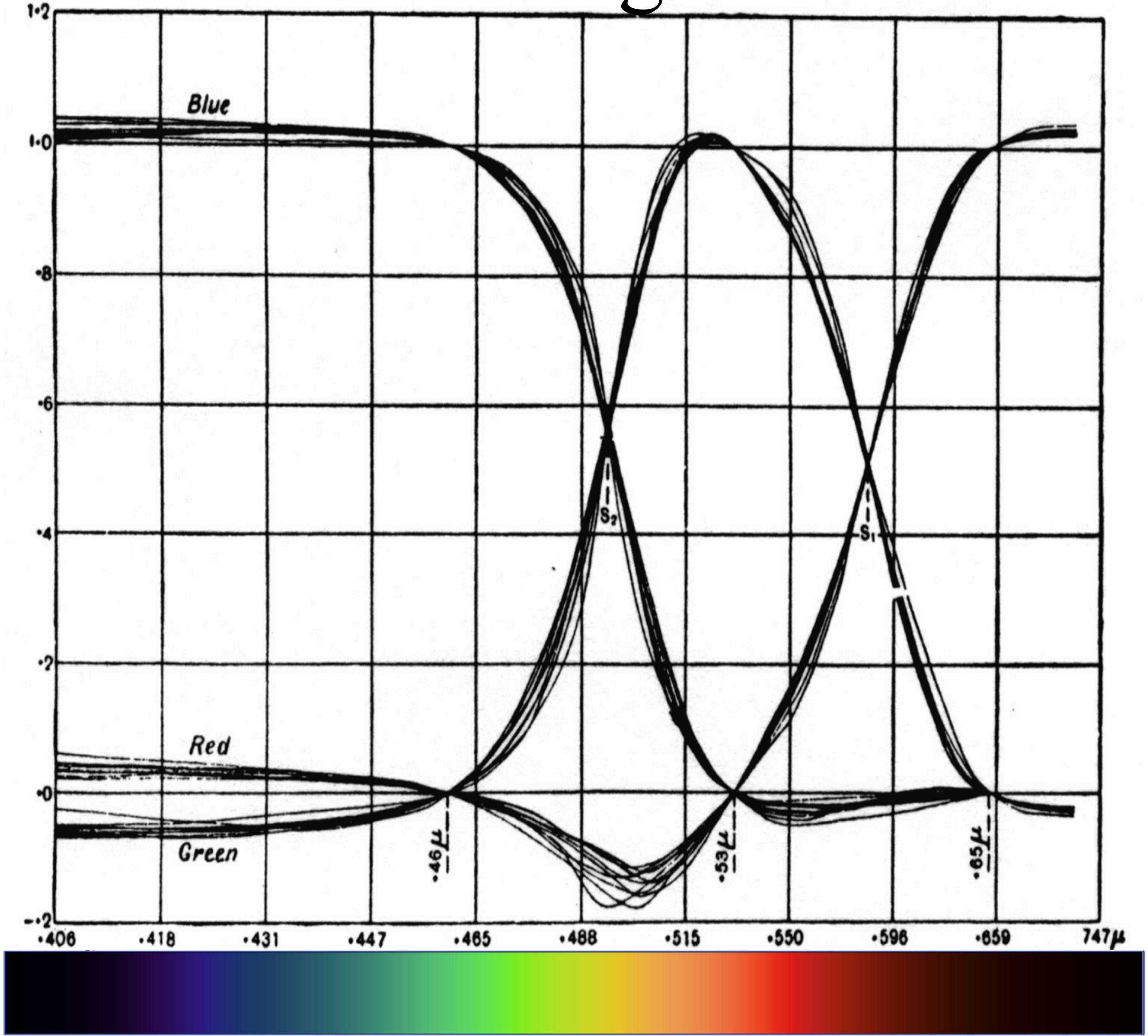
We say a "negative" amount of p_2 was needed to make the match, because we added it to the test color's side.



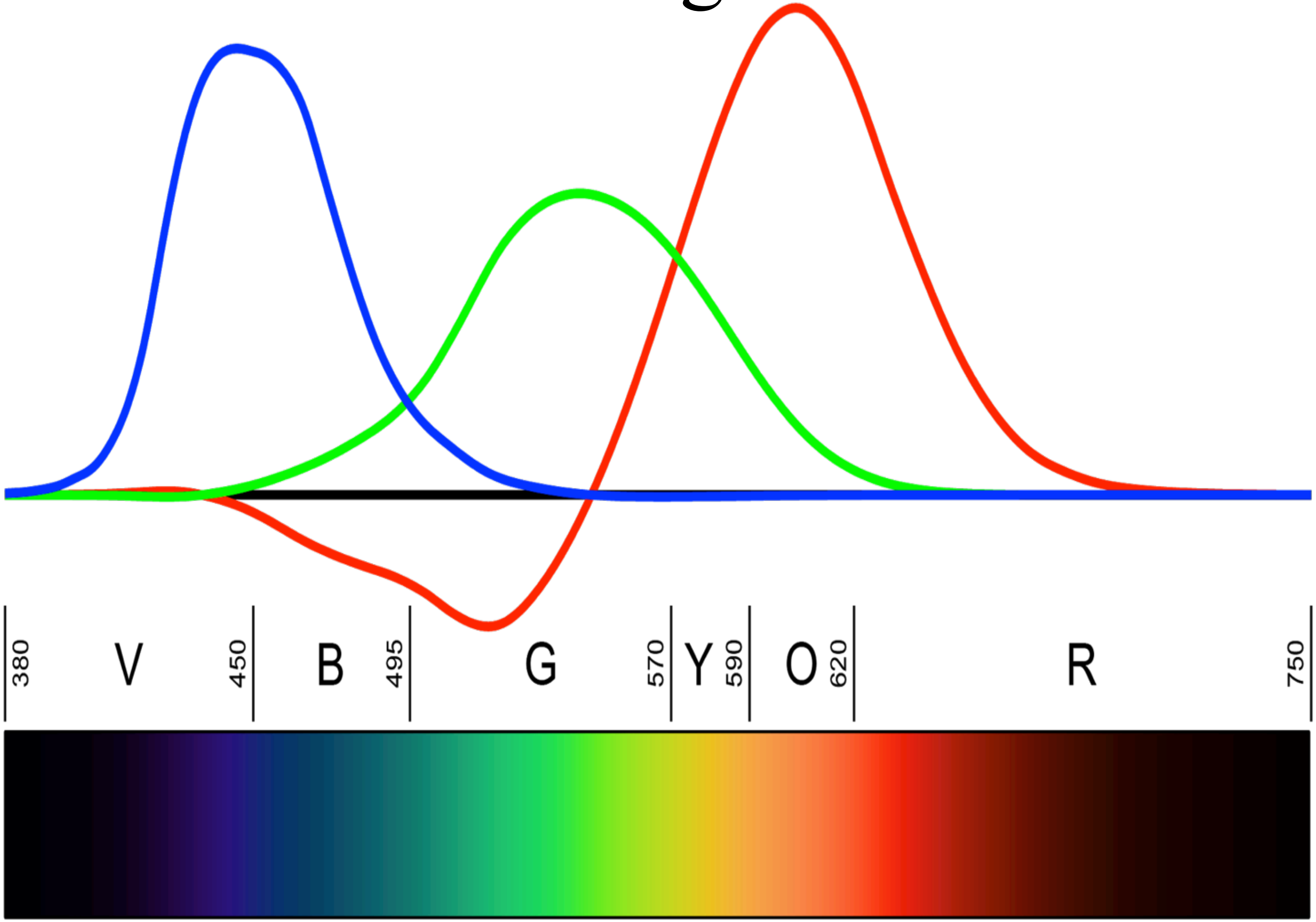




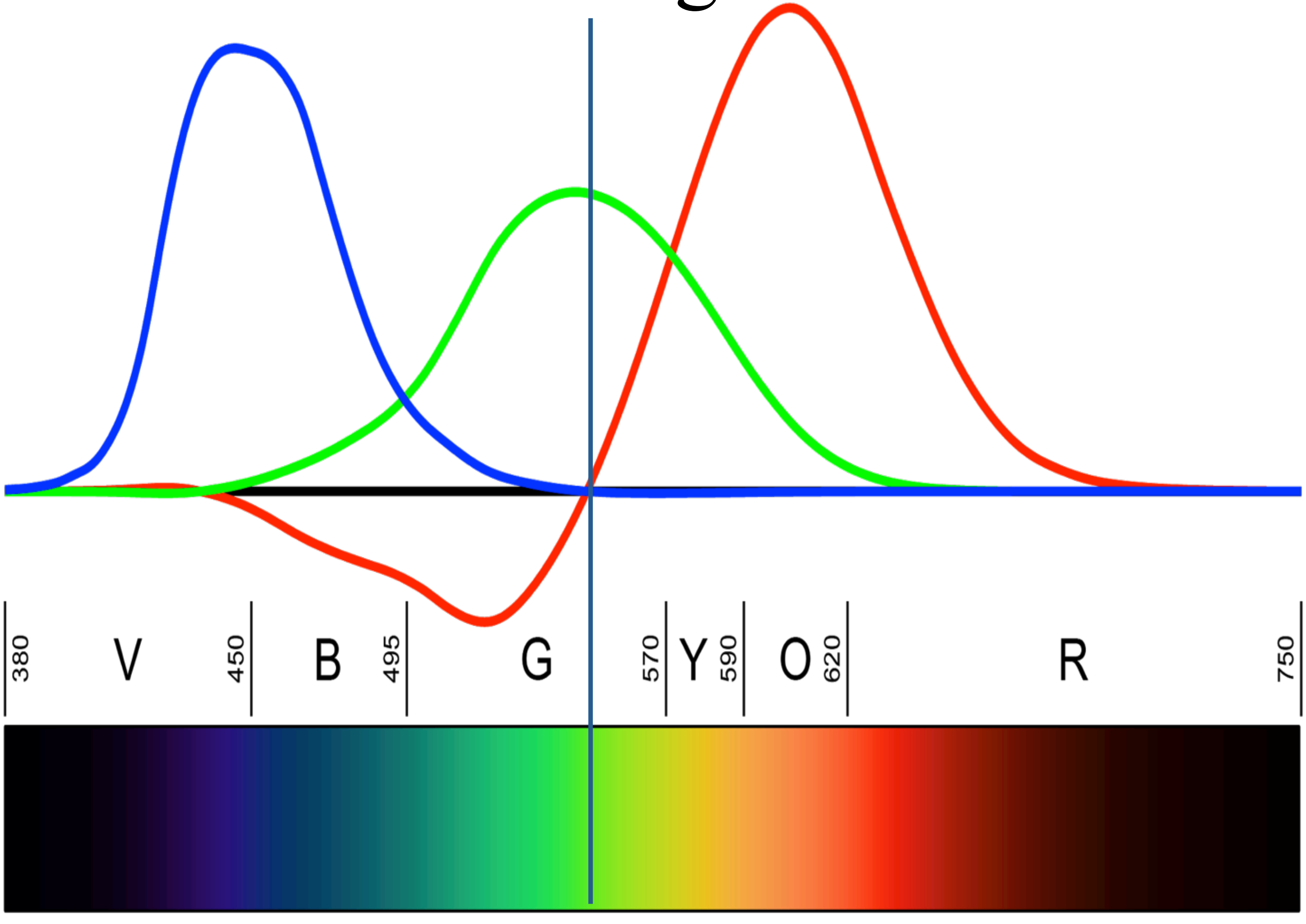
RGB matching functions



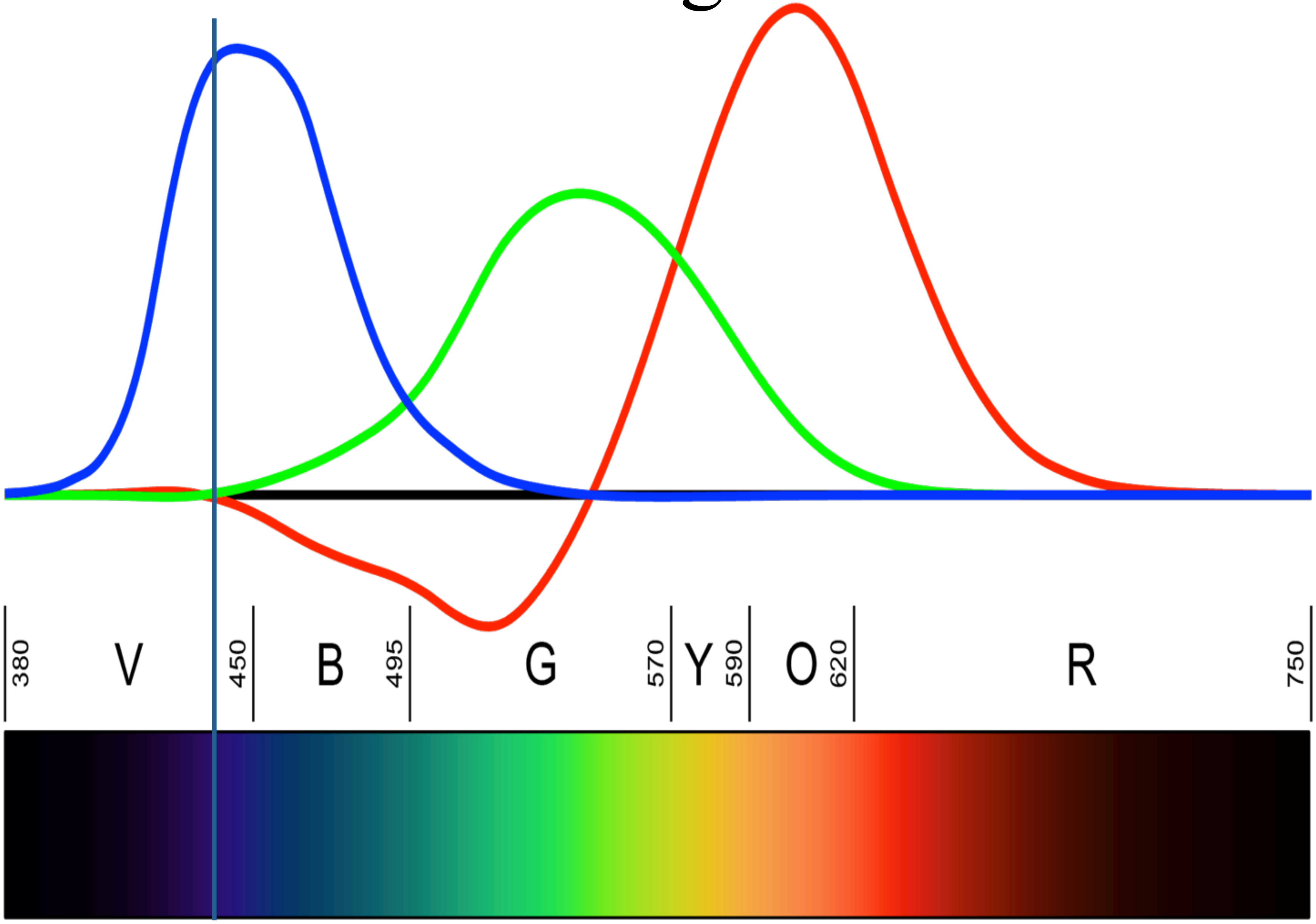
RGB matching functions



RGB matching functions



RGB matching functions

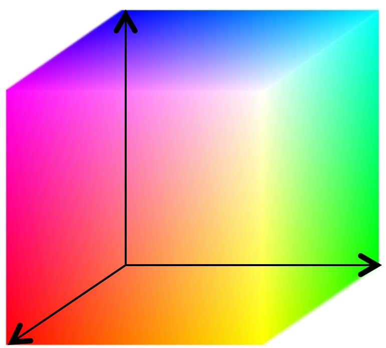


Linear color spaces

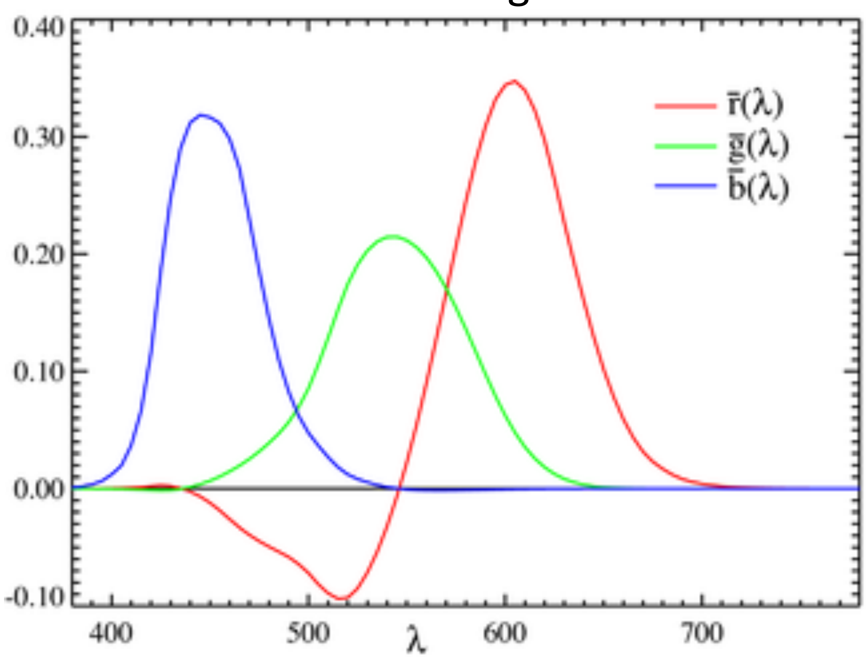
- Defined by a choice of three **primaries**
- The **coordinates** of a color are given by the weights of the primaries used to match it
- In addition to primaries, need to specify **matching functions:** the amount of each primary needed to match a monochromatic light source at each wavelength

Linear color spaces - RGB

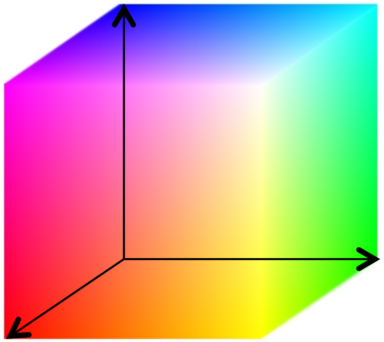
- Defined by a choice of three **primaries**
- The **coordinates** of a color are given by the weights of the primaries used to match it
- In addition to primaries, need to specify **matching functions:** the amount of each primary needed to match a monochromatic light source at each wavelength

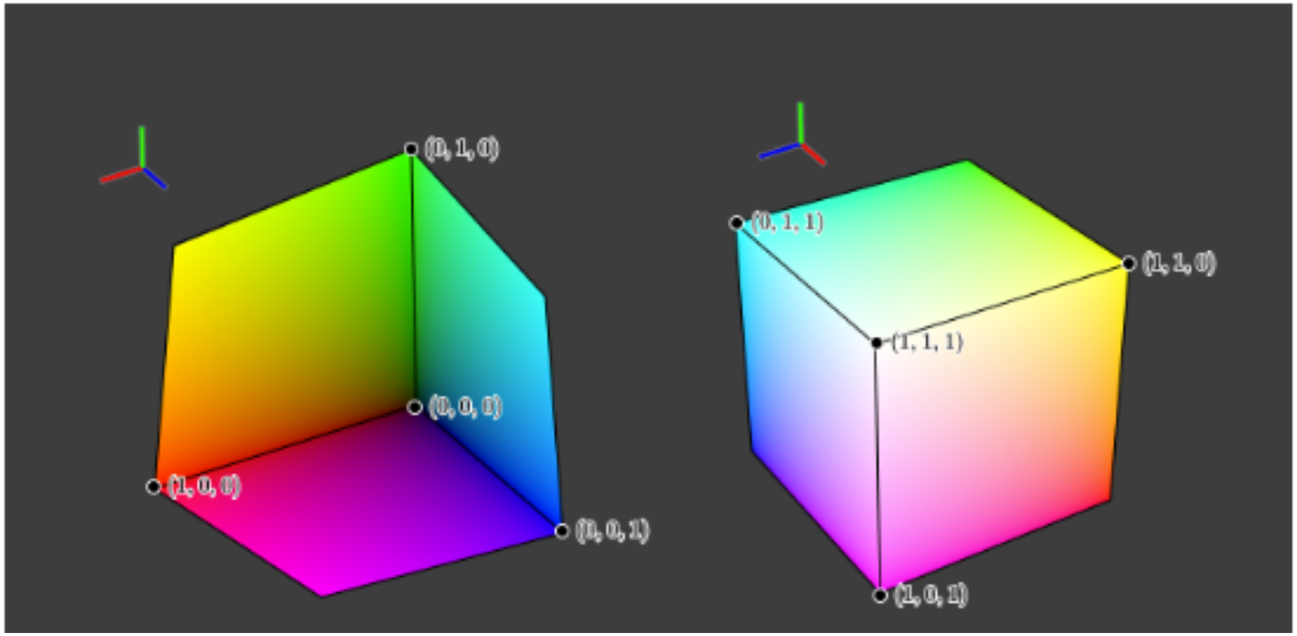


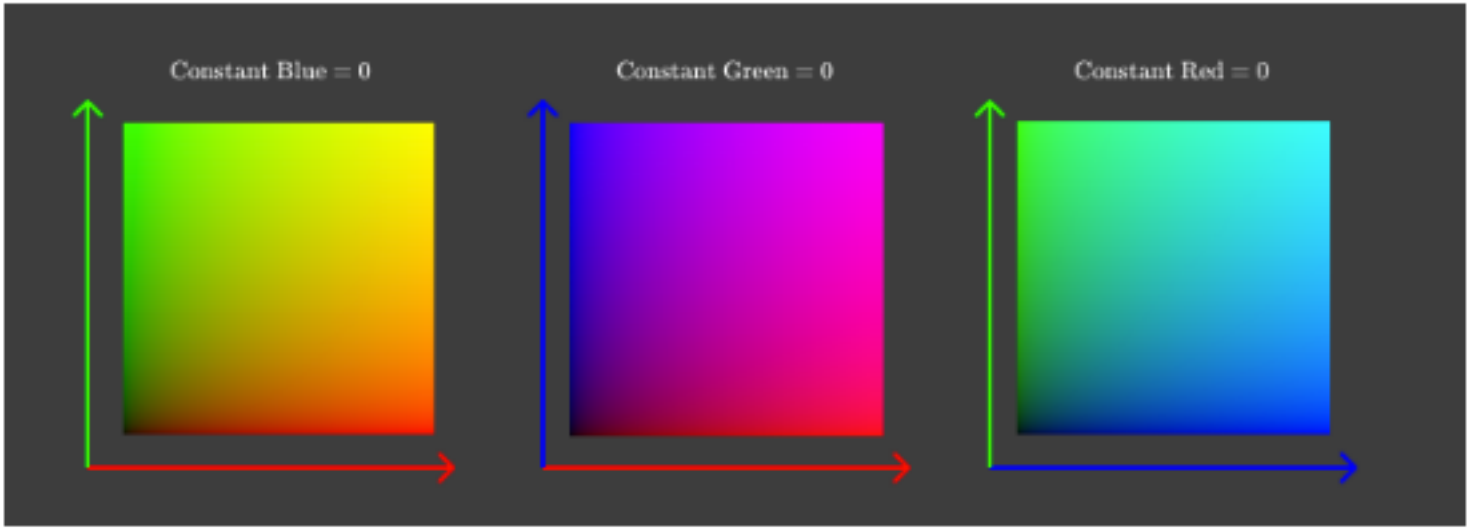
RGB matching functions



Linear color spaces - RGB

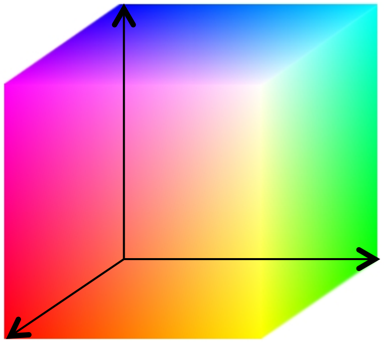


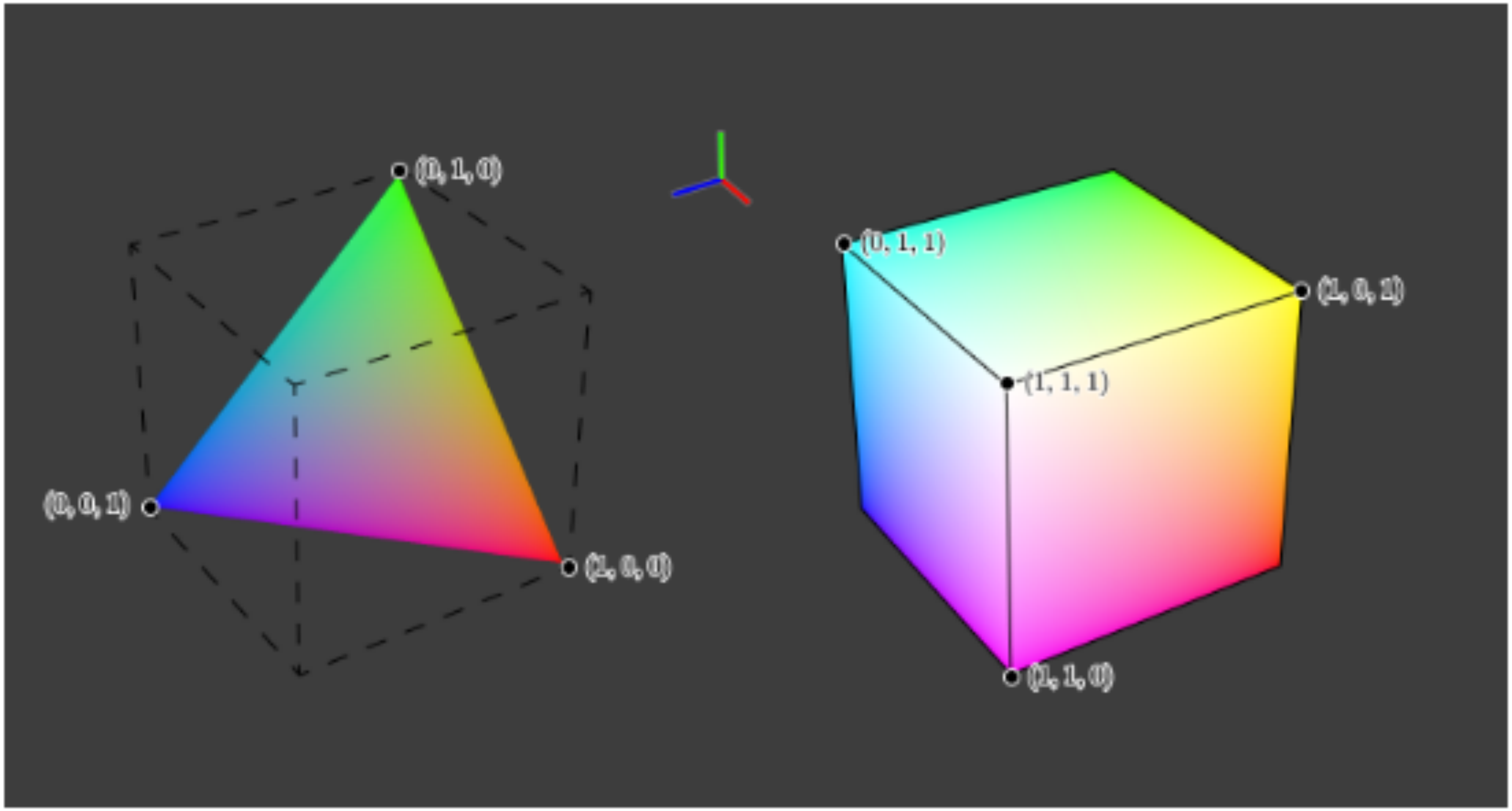




Component pairs plotted, holding the third coordinate constant

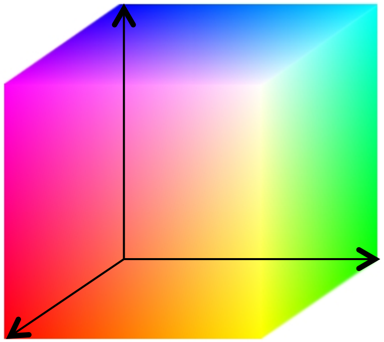
Slicing RGB: R+G+B=1

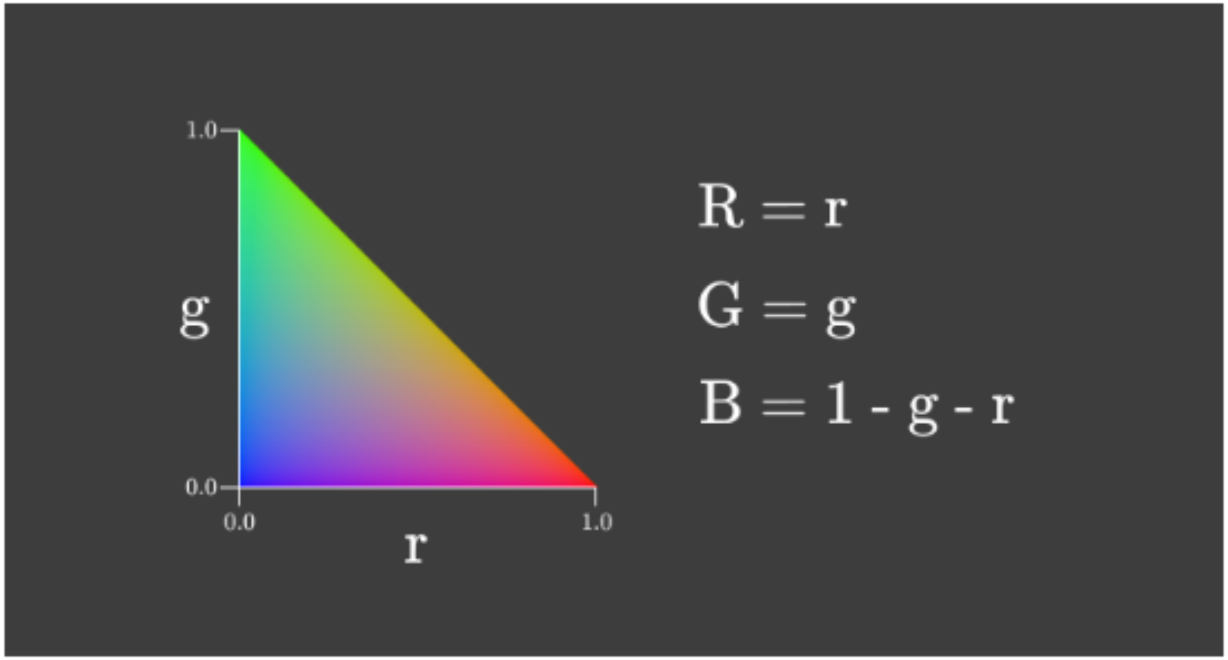




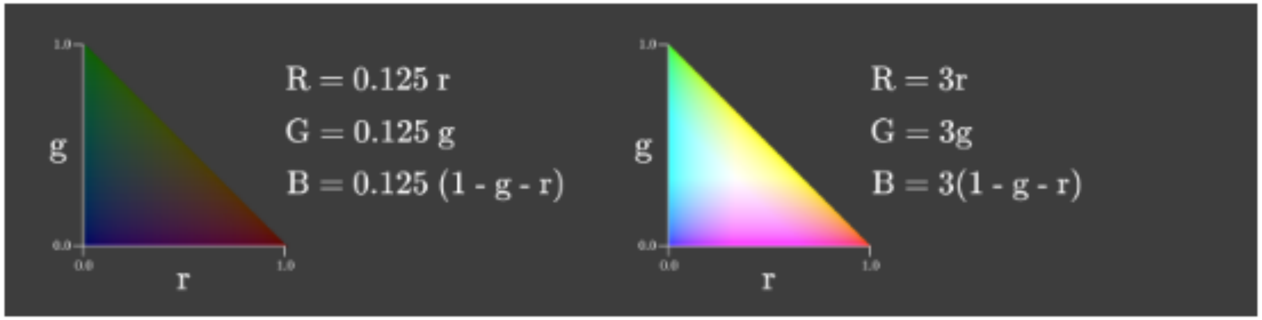
R + G + B = 1 see as an approximation value of brightness
R + G + B = 3 (white)
R + G + B = 0 (black)

rg chromaticity





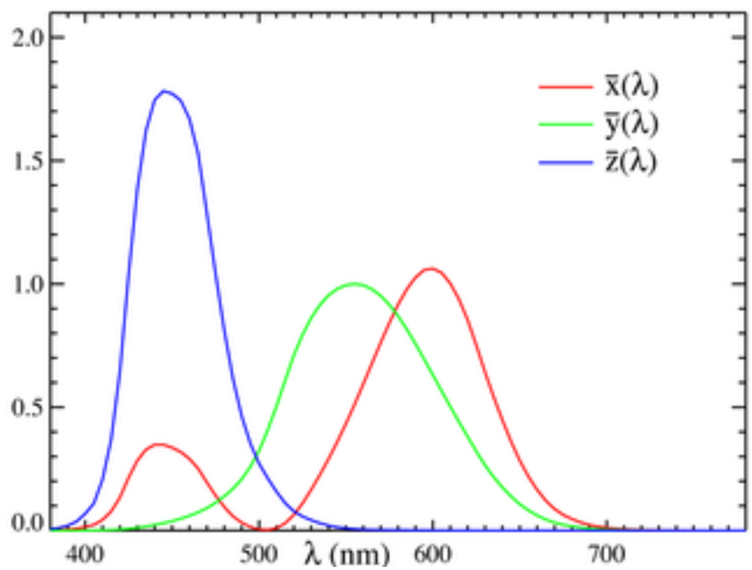
This two dimensional representation of color is called **chromaticity**. This particular kind is called rg chromaticity. Chromaticity gives us information about the ratio of the primary colors independent of the lightness



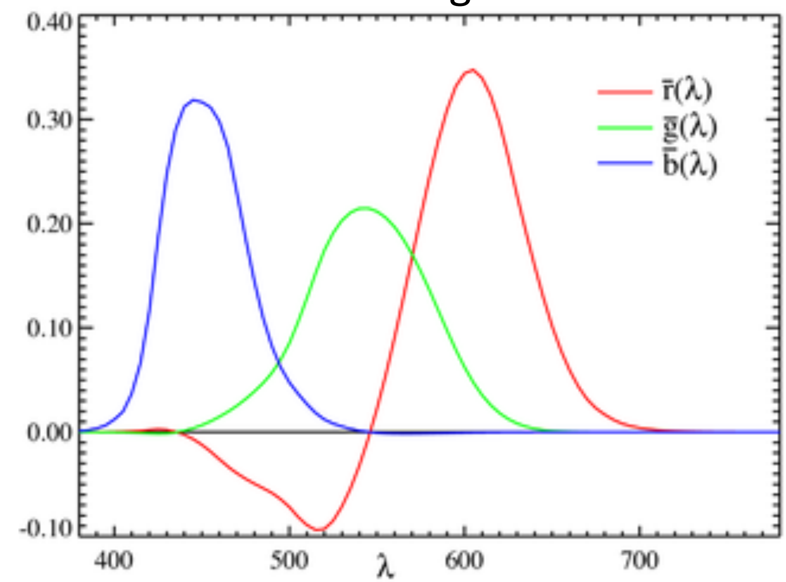
Linear color spaces: CIE XYZ

- Primaries are *imaginary*, but matching functions are everywhere positive
- The Y parameter corresponds to brightness or *luminance* of a color
- Z is blue
- X is a mix of response curves chosen to be nonnegative

XYZ matching functions



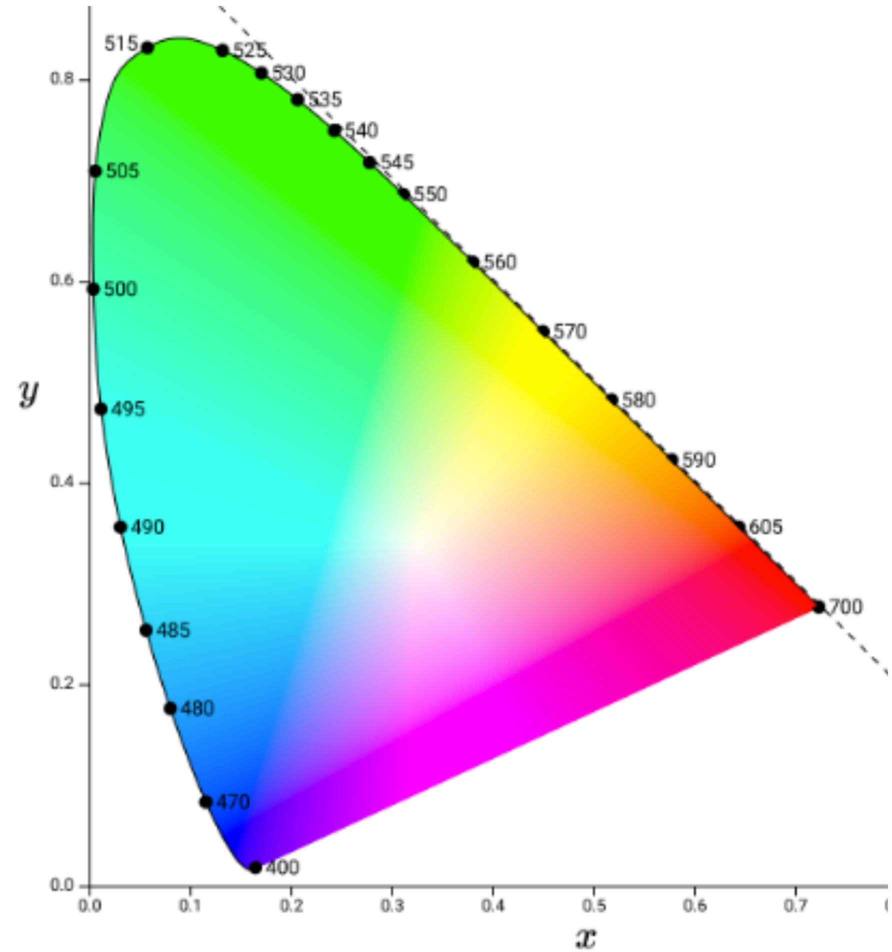
RGB matching functions



http://en.wikipedia.org/wiki/CIE_1931_color_space

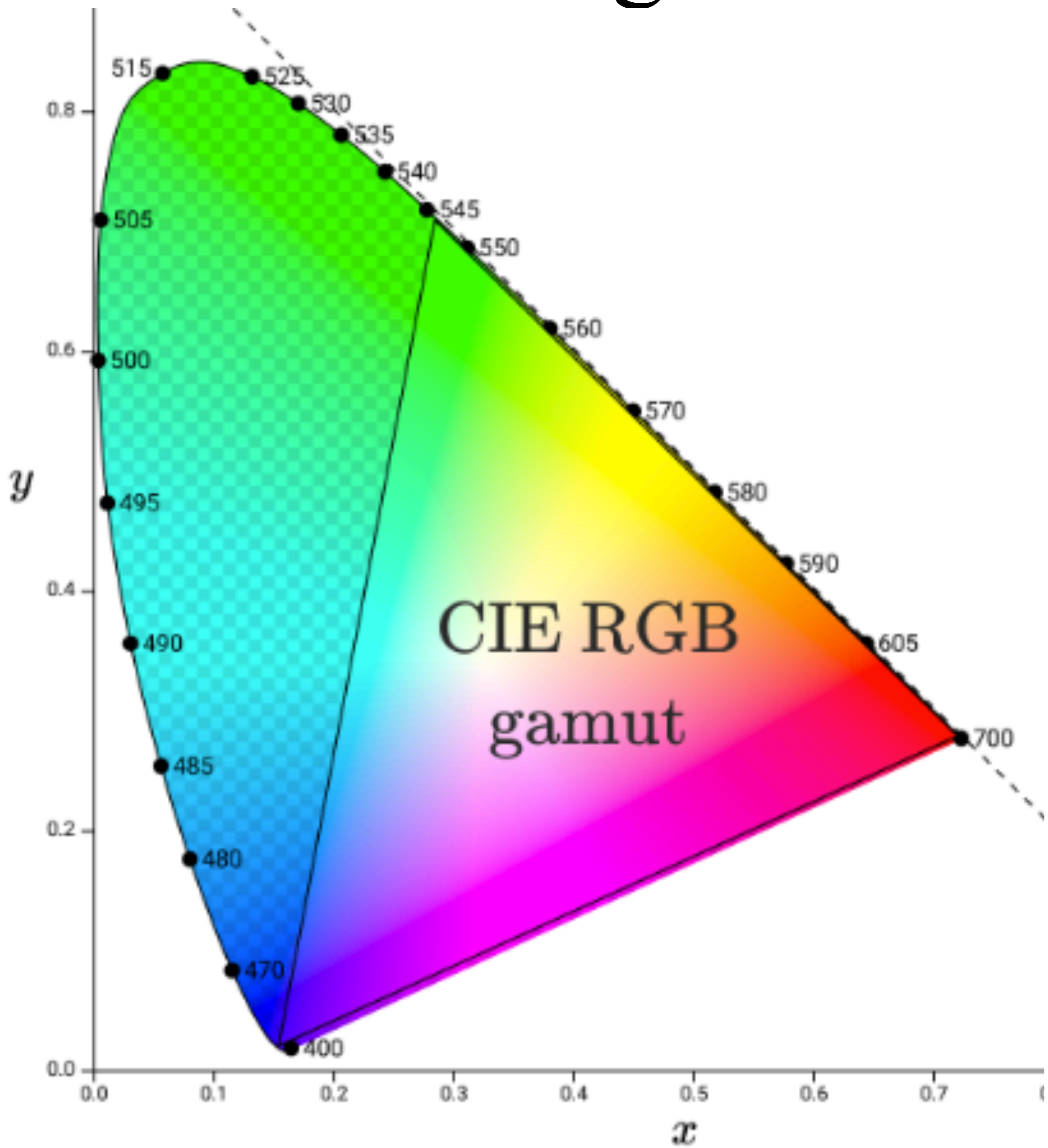
The xy chromaticity diagram

- the concept of color can be divided into two parts: brightness and chromaticity.
- white and grey have the same chromaticity while their brightness differs
- go from 3D (color) to 2D (chromaticity)
- 2D visualization: draw (x,y), where $x = X/(X+Y+Z)$, $y = Y/(X+Y+Z)$

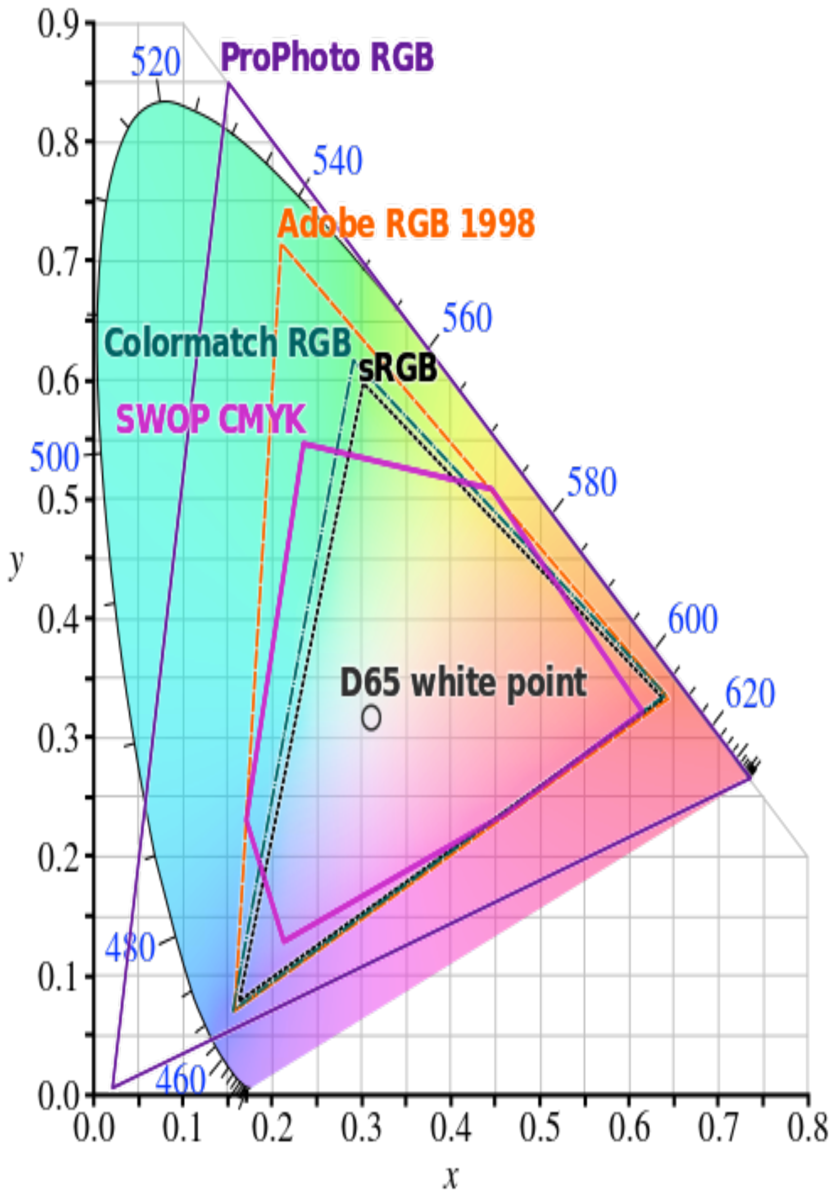


Chromaticity diagram

The RGB gamut

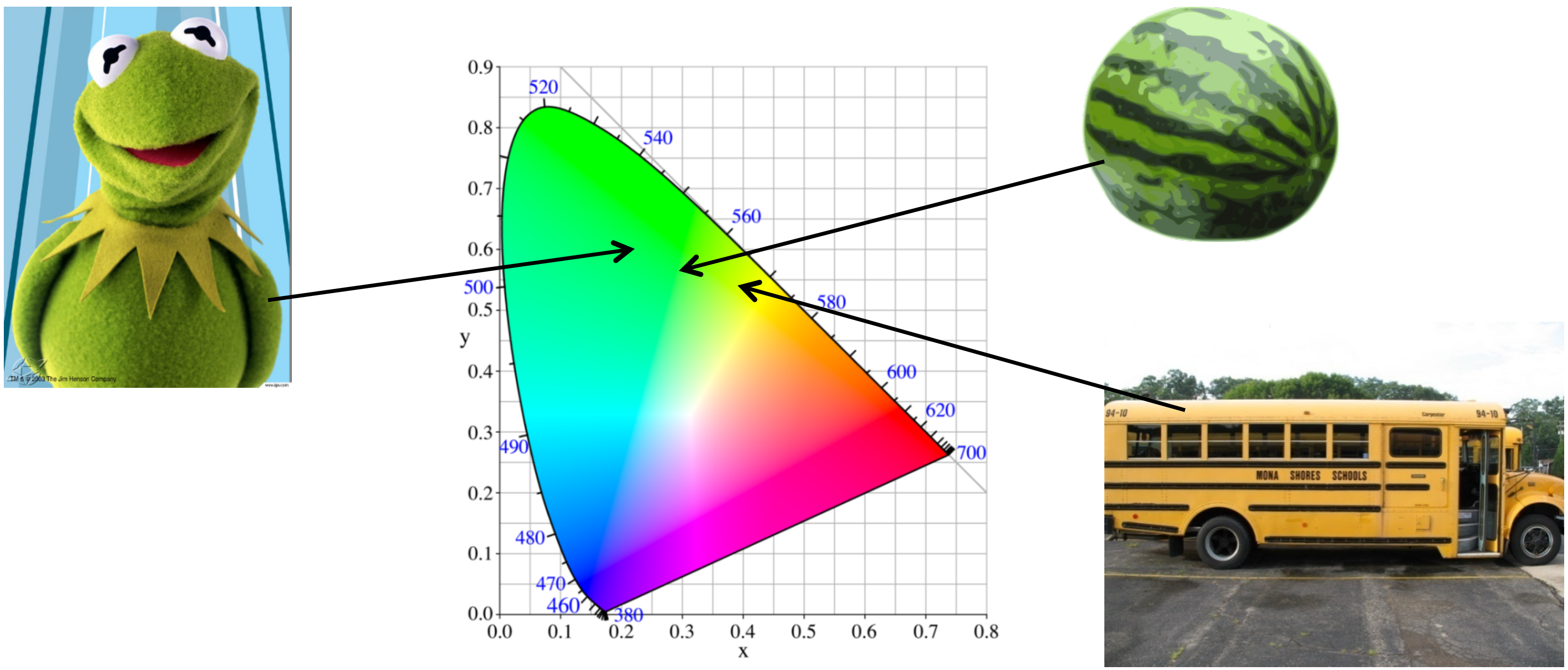


MANY different colorspaces



Distances in color space

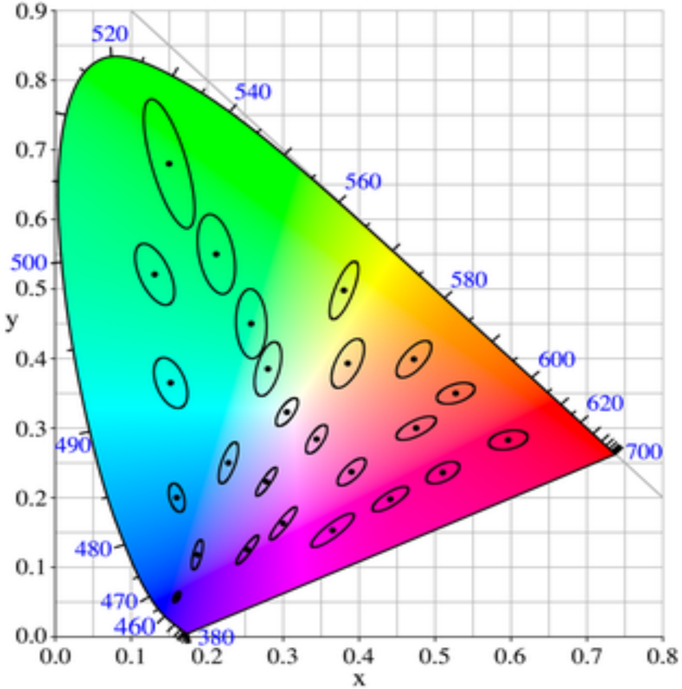
- Are distances between points in a color space perceptually meaningful?



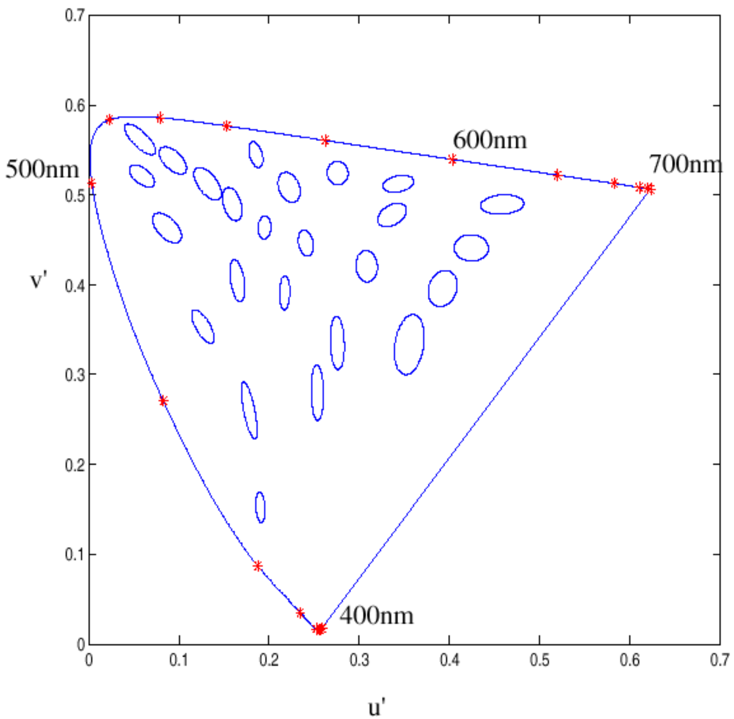
- Not necessarily: CIE XYZ is **not** a *uniform* color space, so magnitude of differences in coordinates are poor indicator of color “distance”.

Uniform color spaces

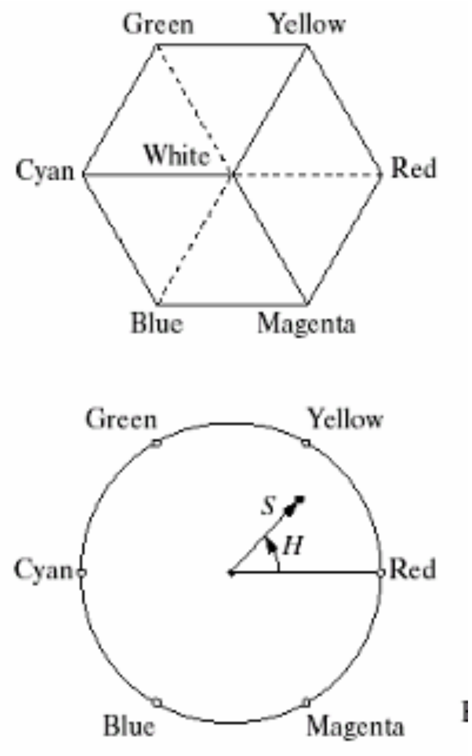
- Unfortunately, differences in x,y coordinates do not reflect perceptual color differences
- CIE u'v' is a projective transform of x,y to make the ellipses more uniform

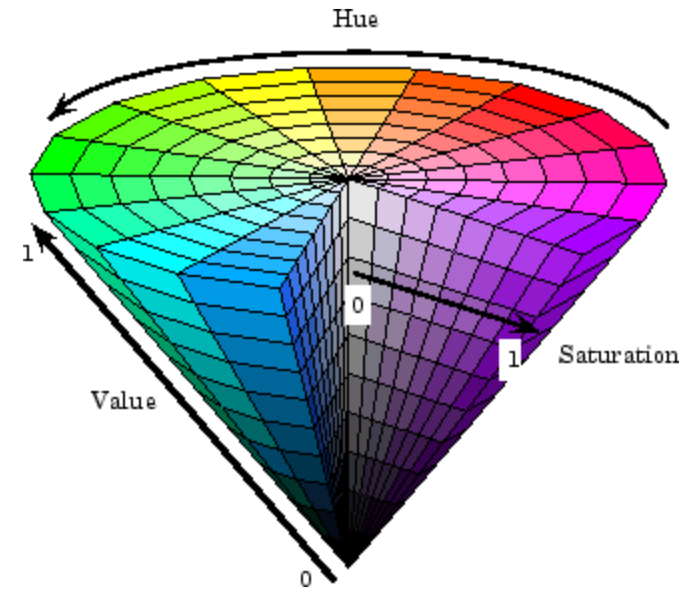


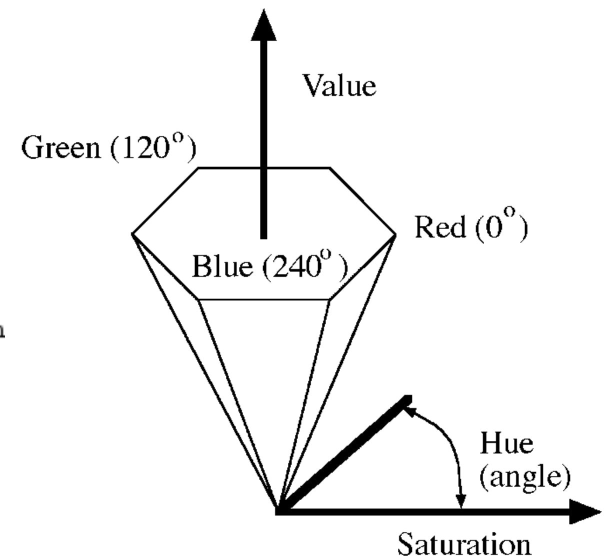
McAdam ellipses: Just noticeable differences in color



Nonlinear color spaces: HSV

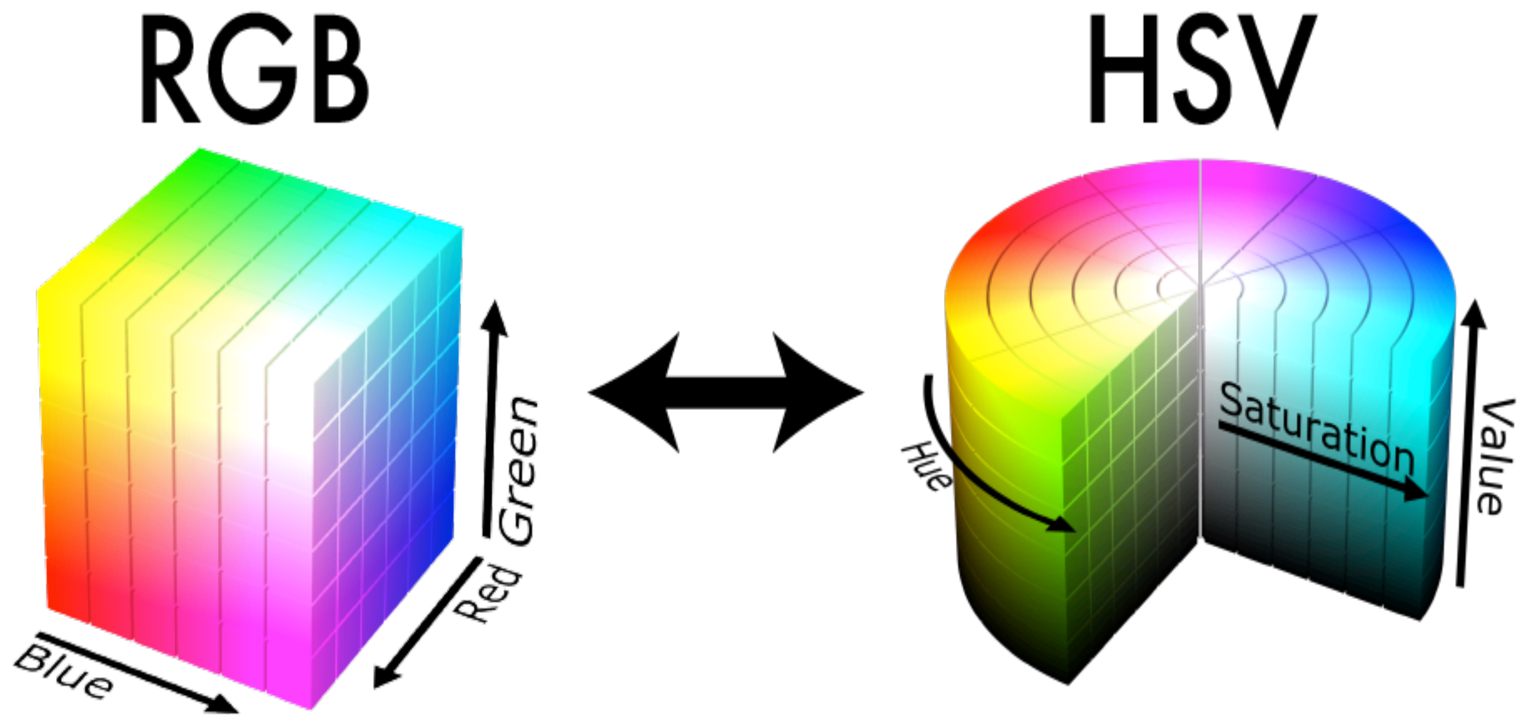






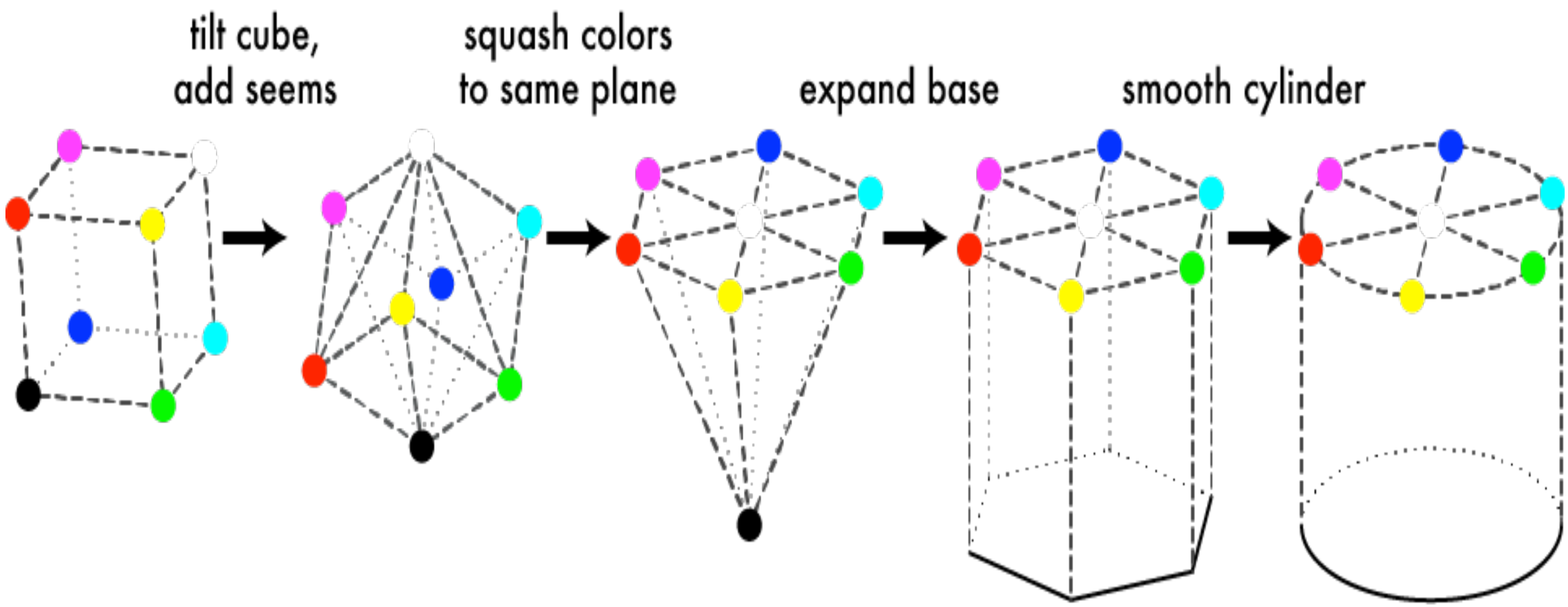
- Perceptually meaningful dimensions: Hue, Saturation, Value (Intensity)
- RGB cube on its vertex

RGB2HSV

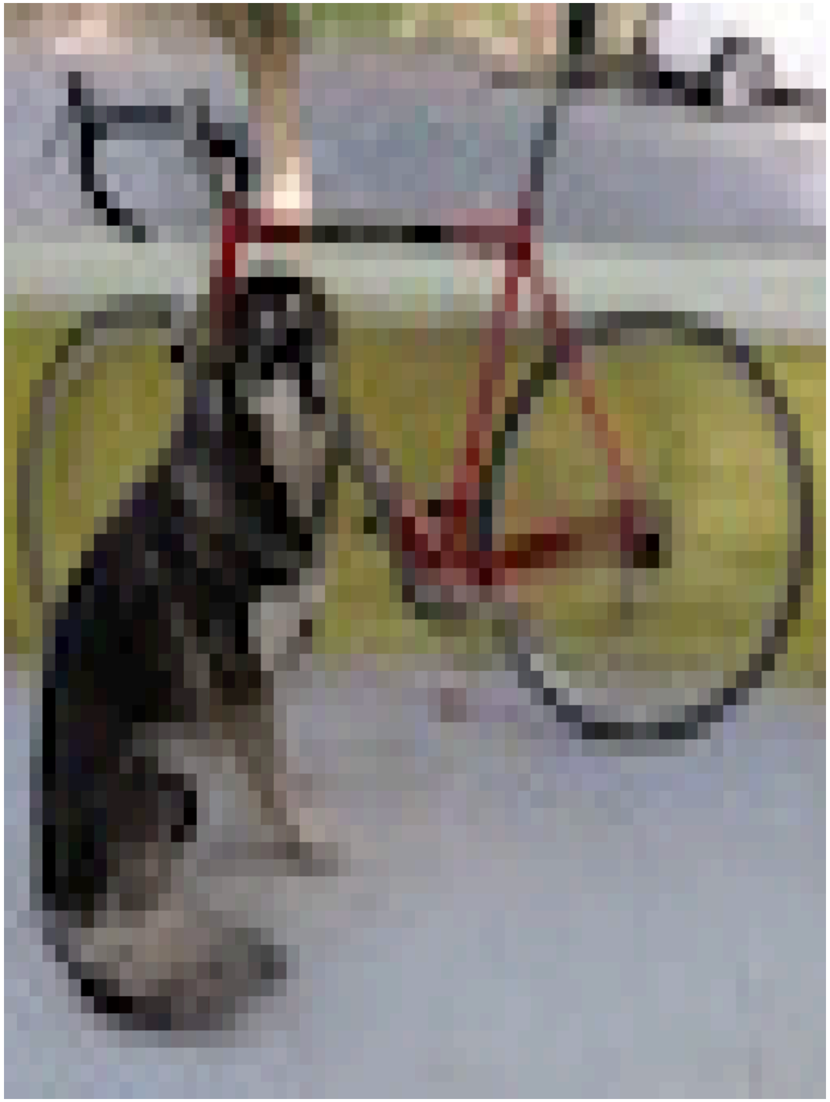


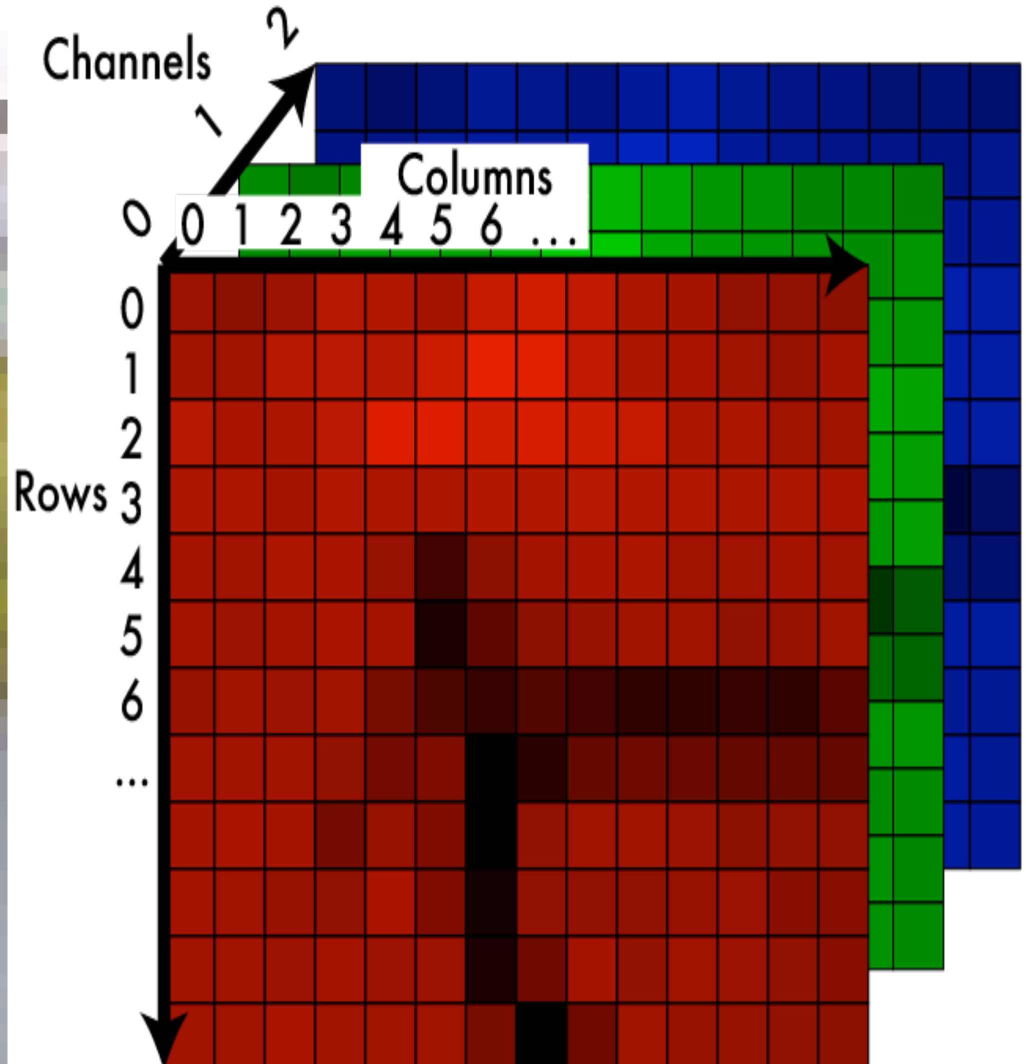
- Different model based on perception of light
- Hue: what color
- Saturation: how much color
- Value: how bright
- Allows easy image transforms: shift the hue, increase saturation

Geometric RGB to HSV

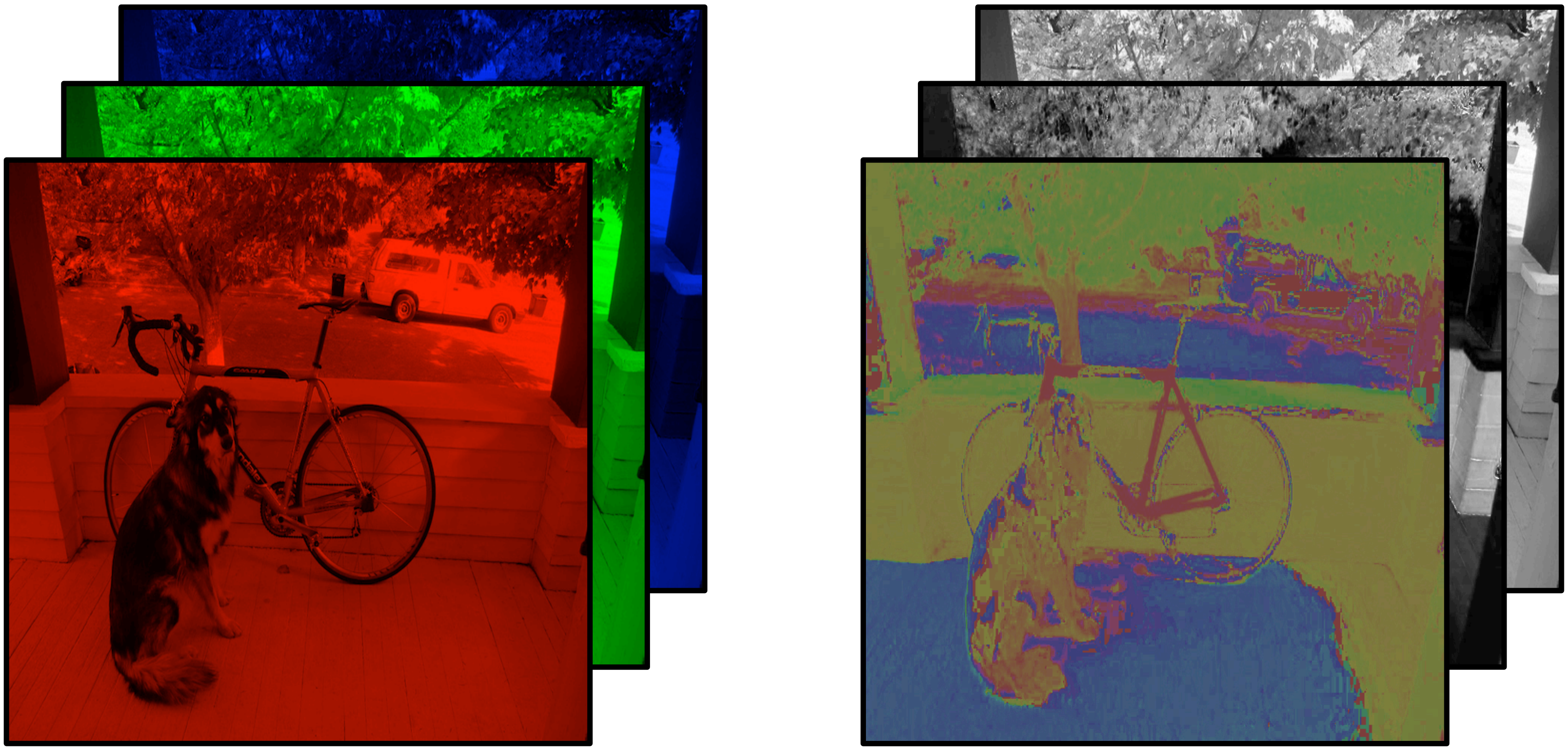


RGB Color image: 3d tensor in colorspace

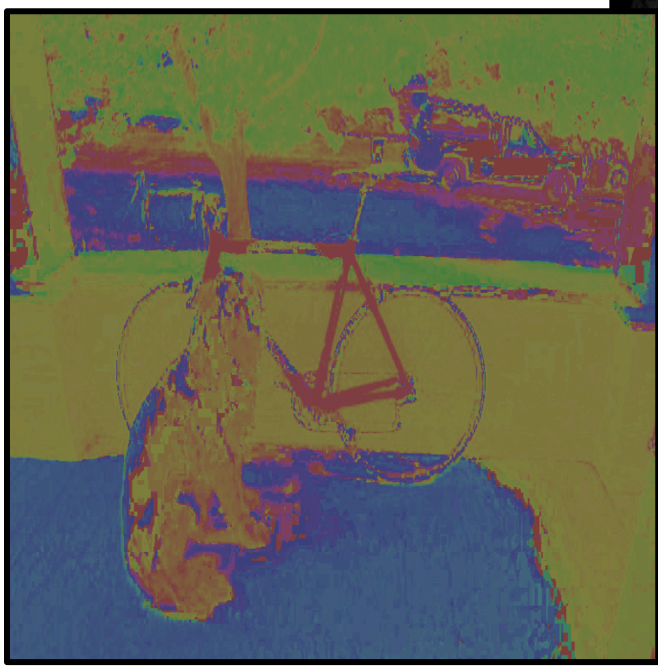




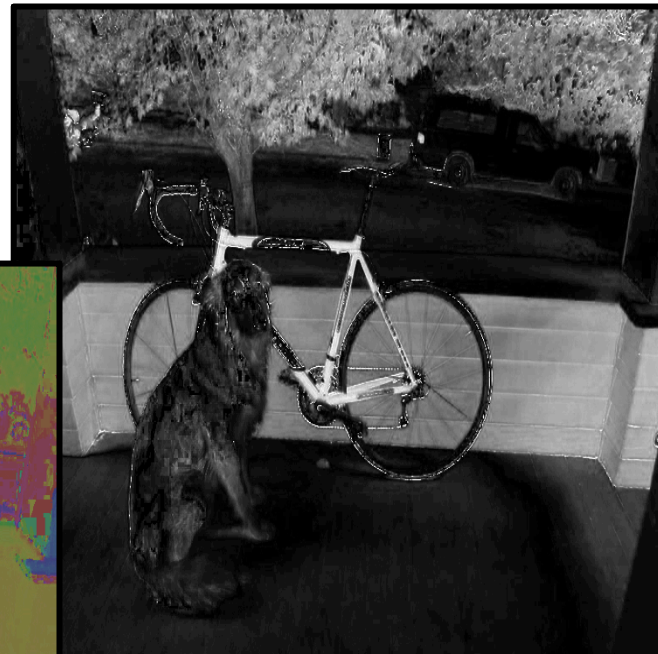
HSV Color image: 3d tensor, different info



Hue

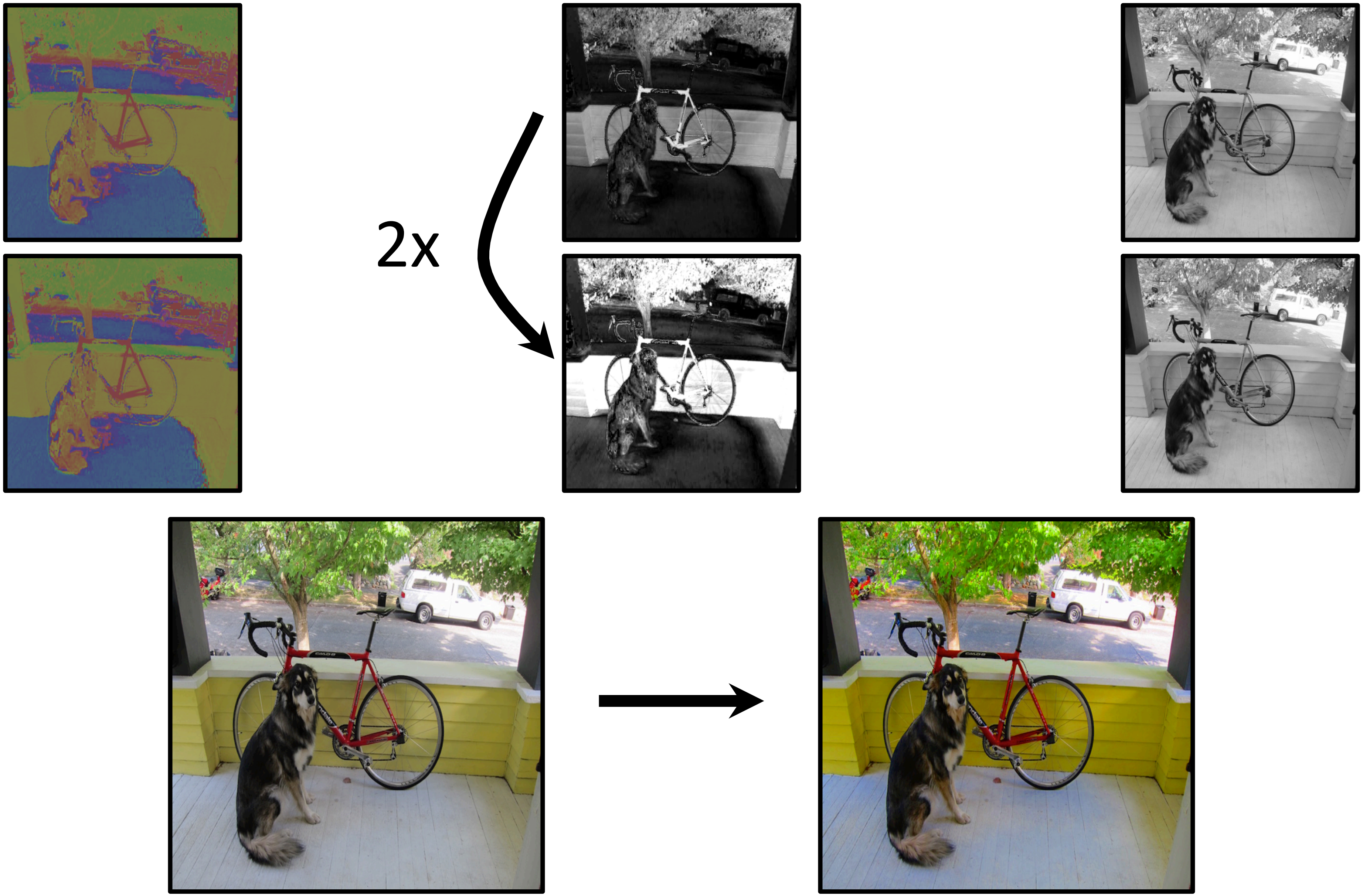


Saturation

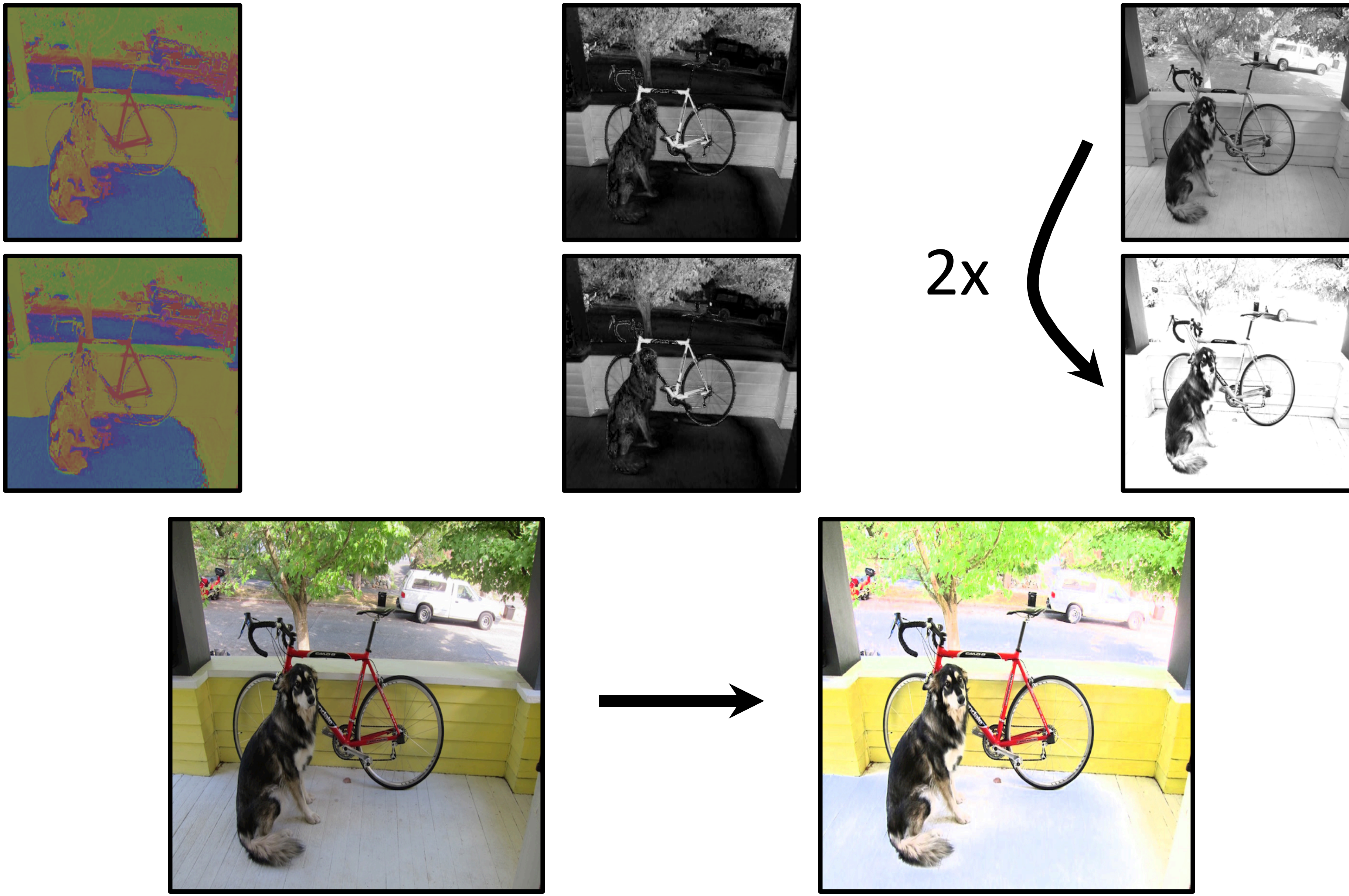


Value

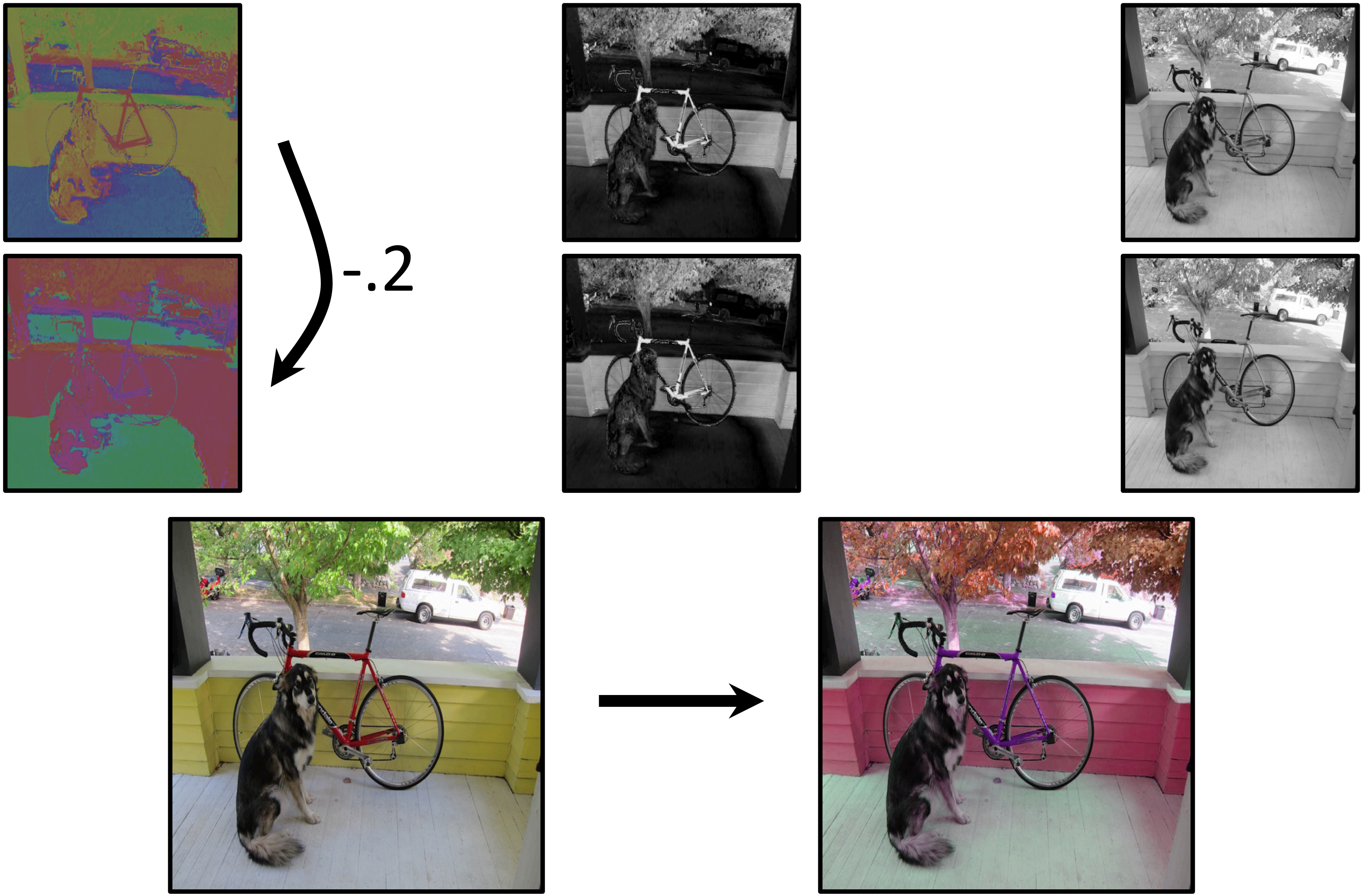
More saturation = intense colors



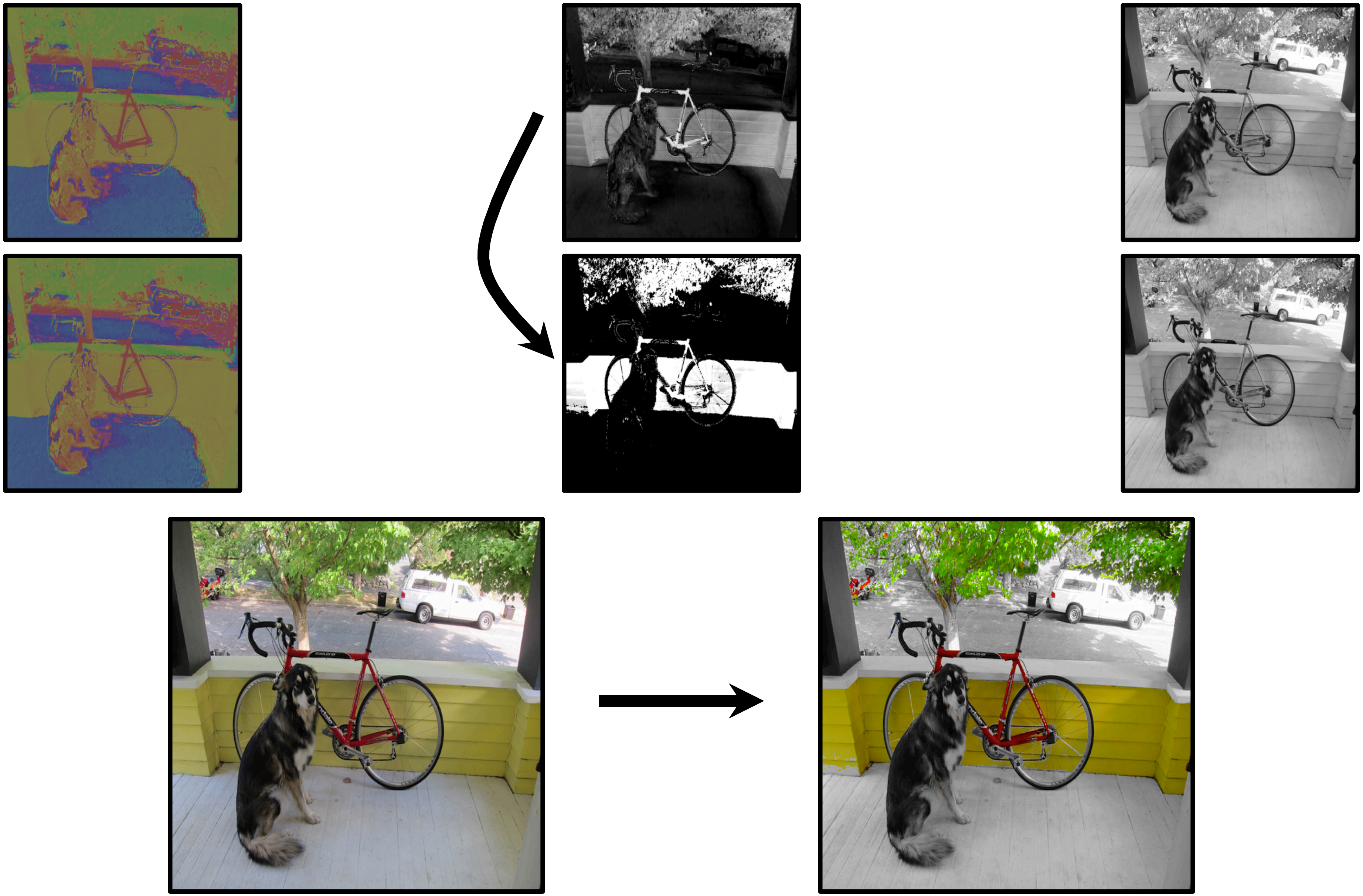
More value = lighter image



Shift hue = shift colors



Increase and threshold saturation



Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo, object instance recognition

4. **Object Recognition: high – level vision**

   Object classification, object detection, part based models, bovw models

5. **Video understanding**

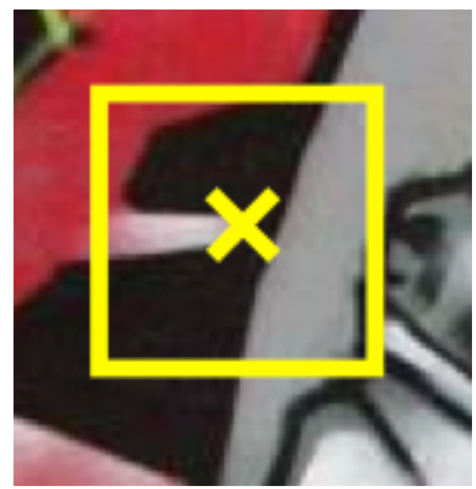
   Object tracking, background subtraction, motion descriptors, optical flow

Local features

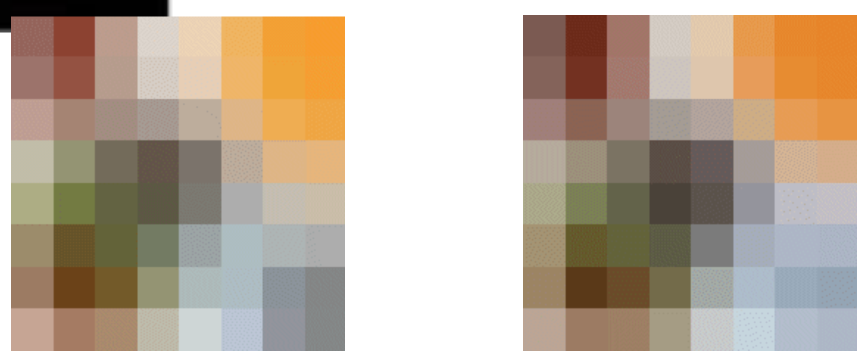
- local feature = interest points = keypoint + region descriptor
- local image feature = a tiny patch in the image that is invariant to image scaling, rotation and change in illumination. It is geometrically (translation, rotation, ...) and photometrically (brightness, exposure, ...) invariant.

Local features

- local feature = interest points = keypoint + region descriptor
- local image feature = a tiny patch in the image that is invariant to image scaling, rotation and change in illumination. It is geometrically (translation, rotation, ...) and photometrically (brightness, exposure, ...) invariant.



Geometric transformations

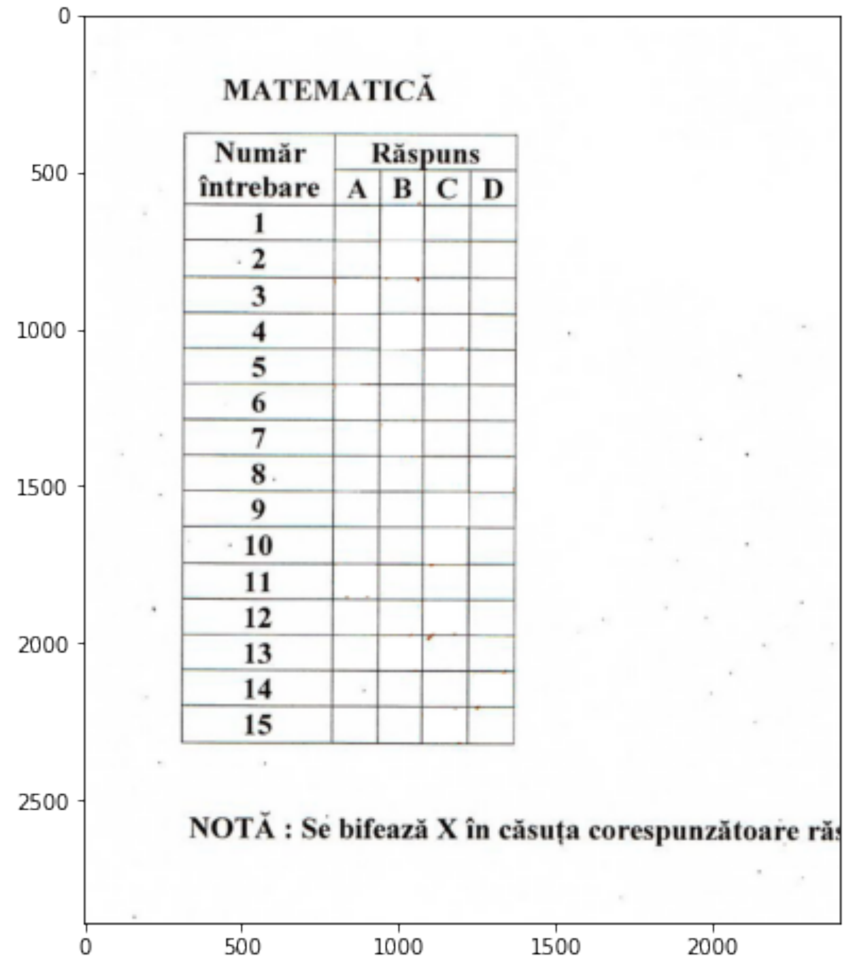


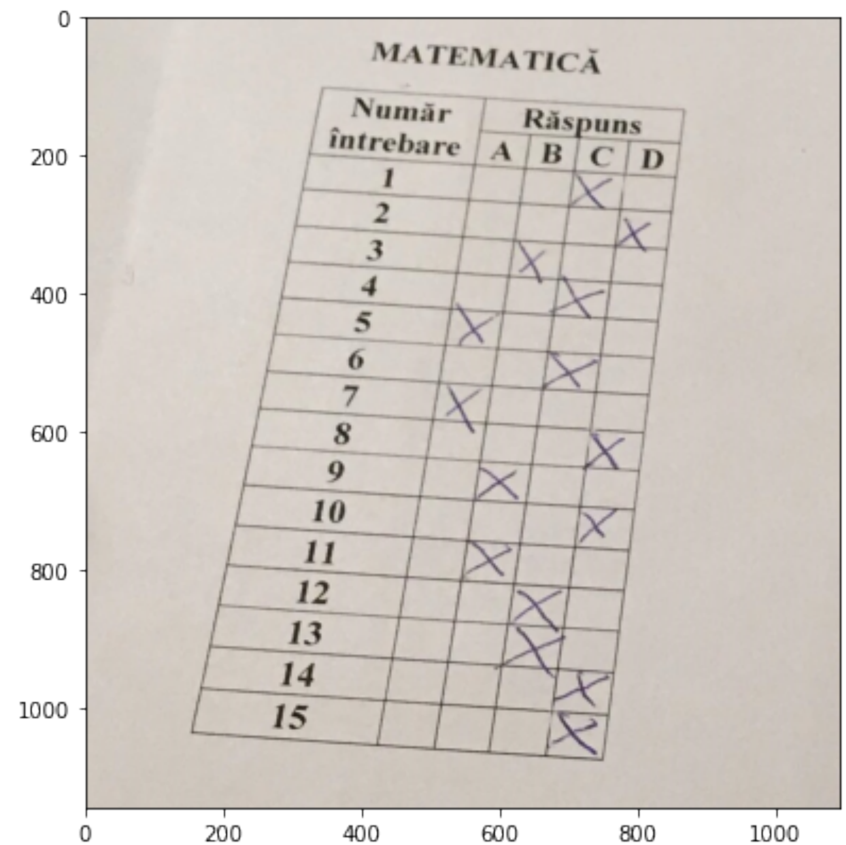
Exemples:
translation,
rotation,
scaling

Photometric transformations

Matching local features – last year P1

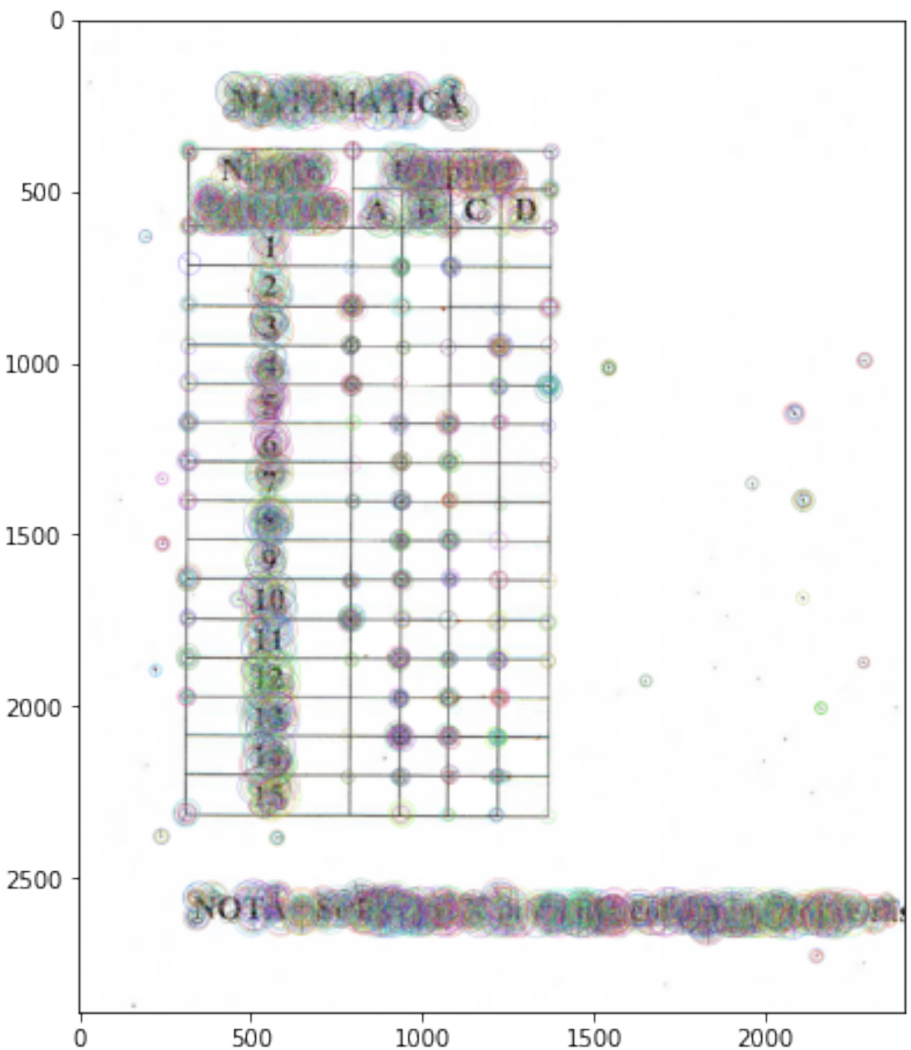
- matching local image feature = find correspondence points between images based on local features that look the same

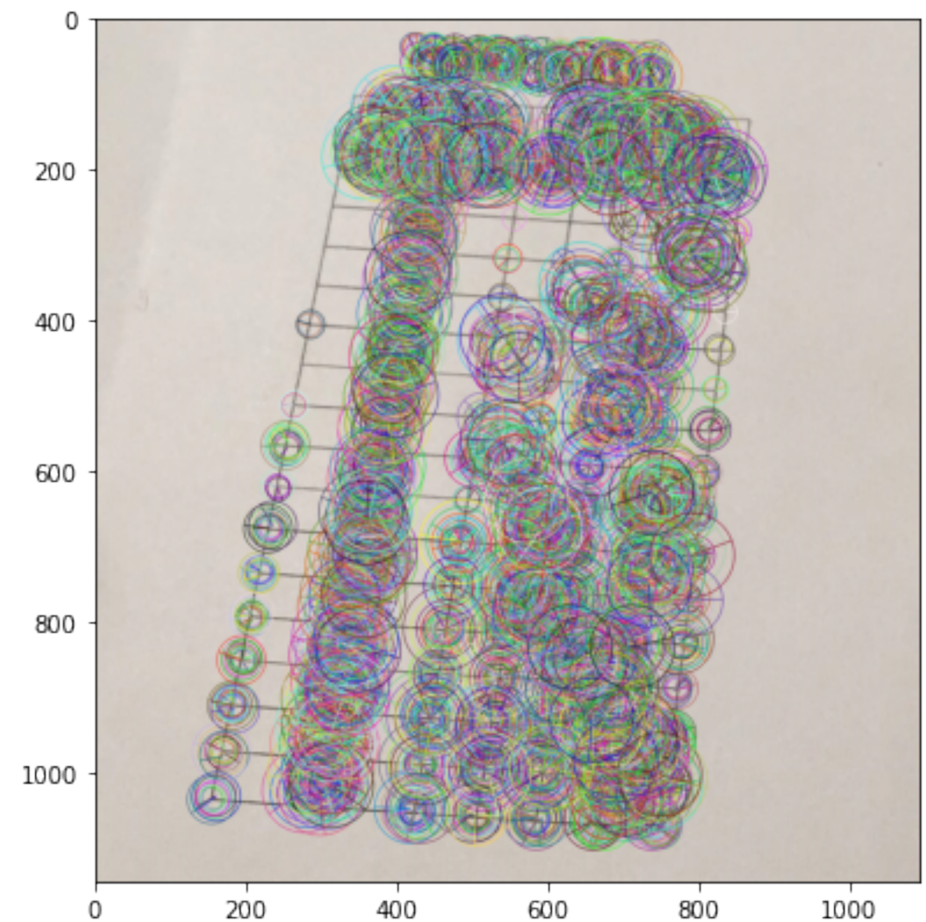




Matching local features – last year P1

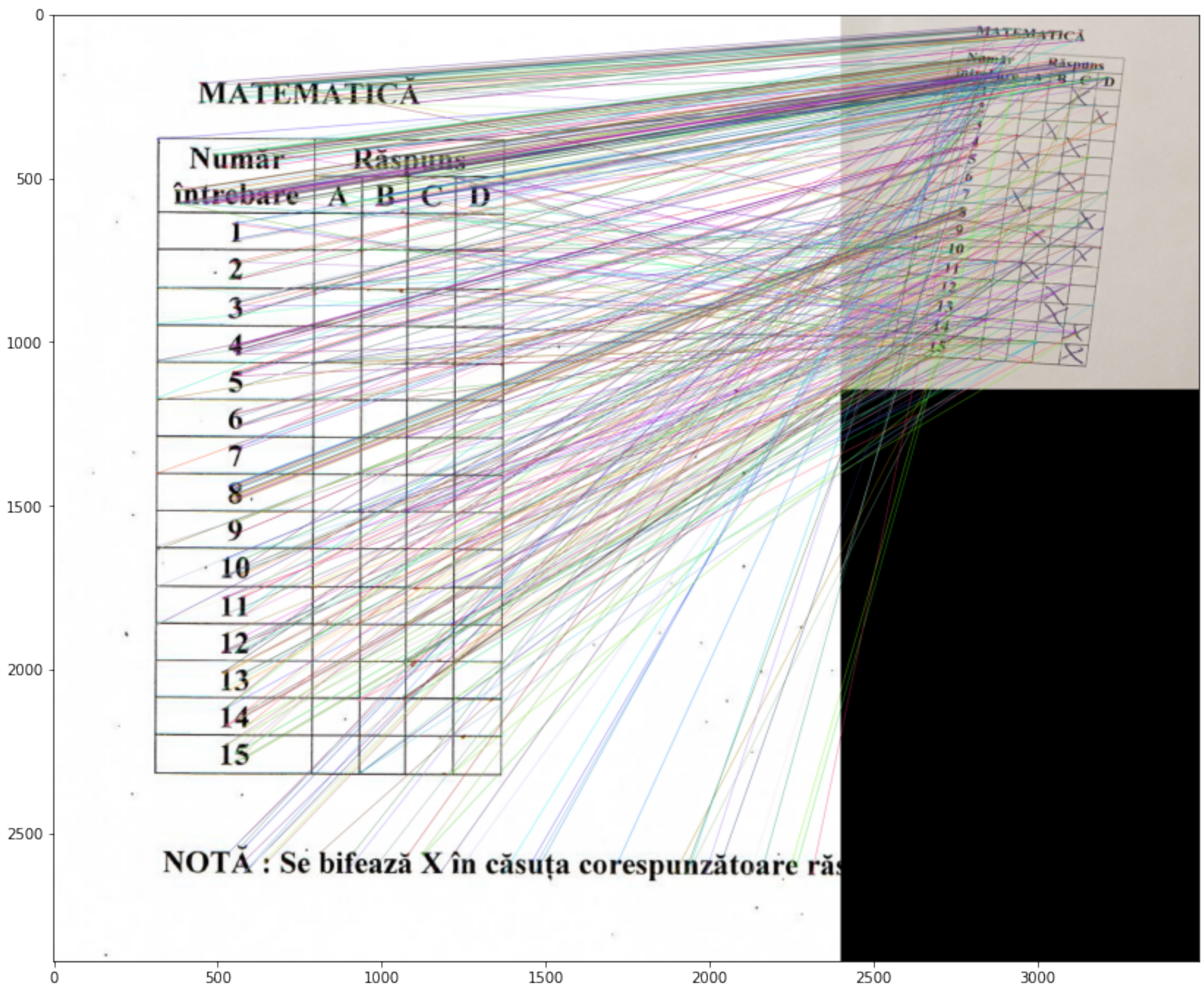
- matching local image feature = find correspondence points between images based on local features that look the same





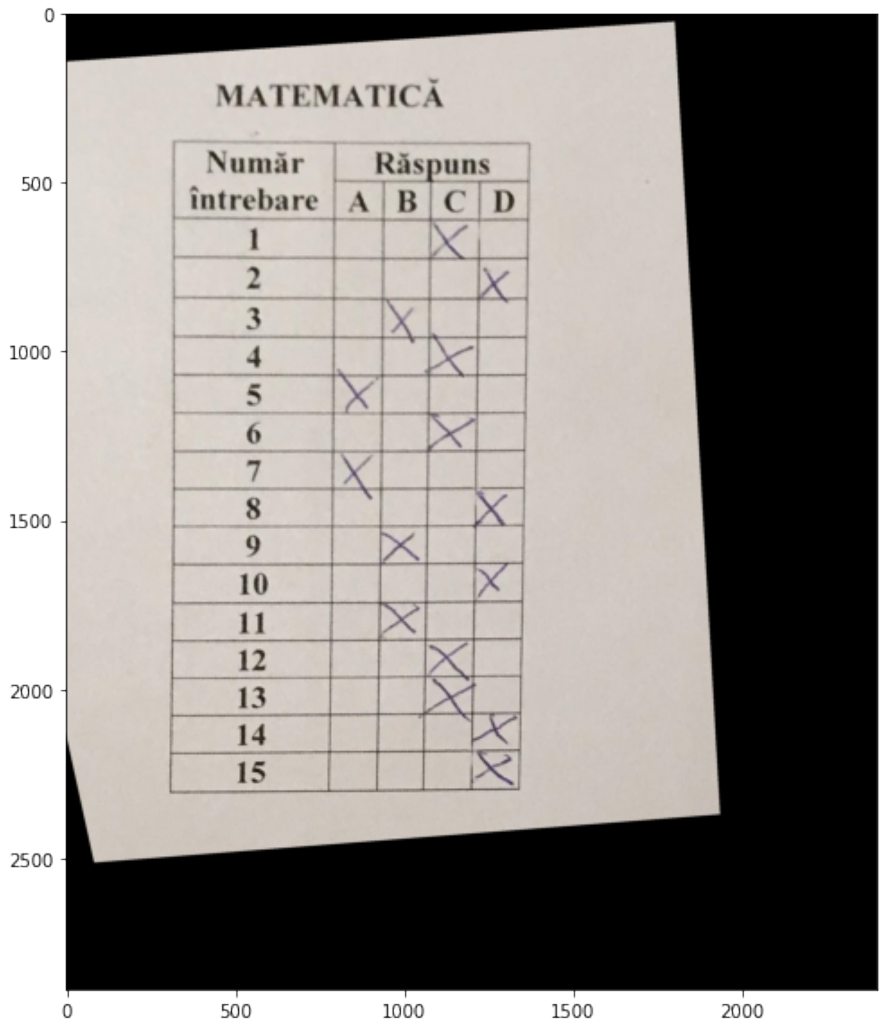
Matching local features – last year P1

- matching local image feature = find correspondence points between images based on local features that look the same



Finding the homography transformation

- Find the matrix of the homography (perspective) transformation and use it to warp one image

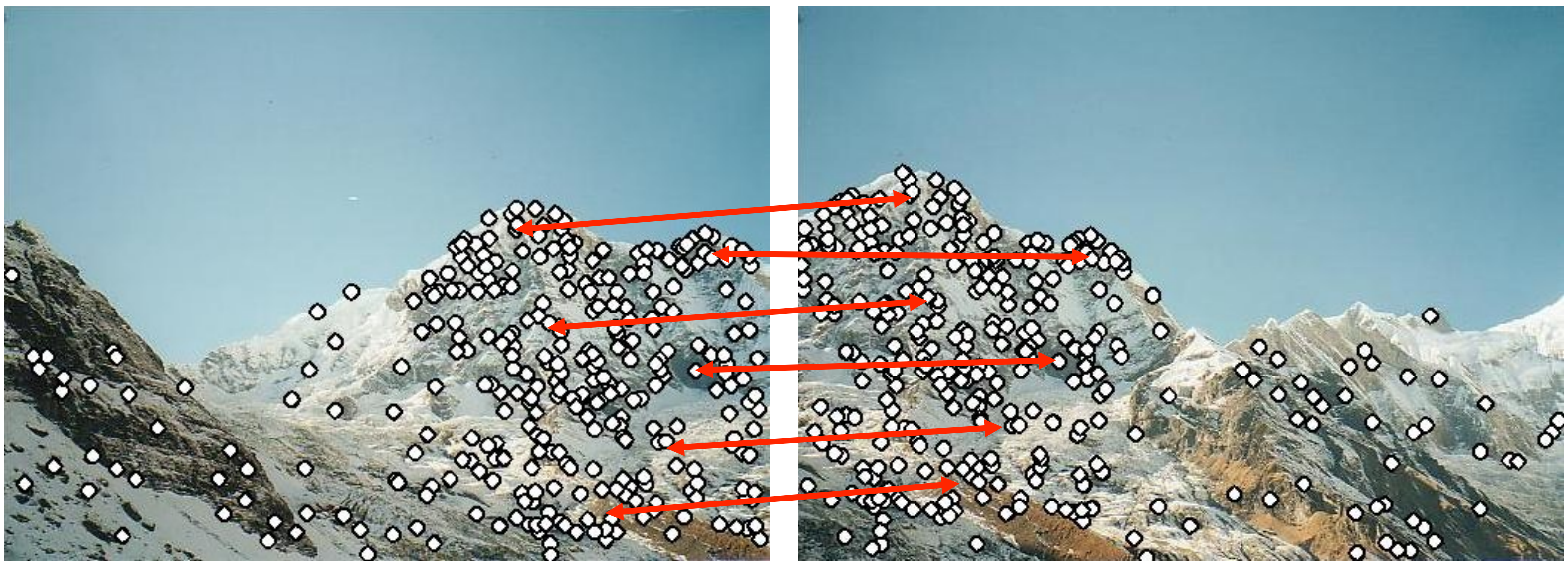


Why extract local features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?


Why extract local features?

- Motivation: panorama stitching
  - We have two images – how do we combine them?



Step 1: extract local features

Step 2: match local features

Why extract local features?

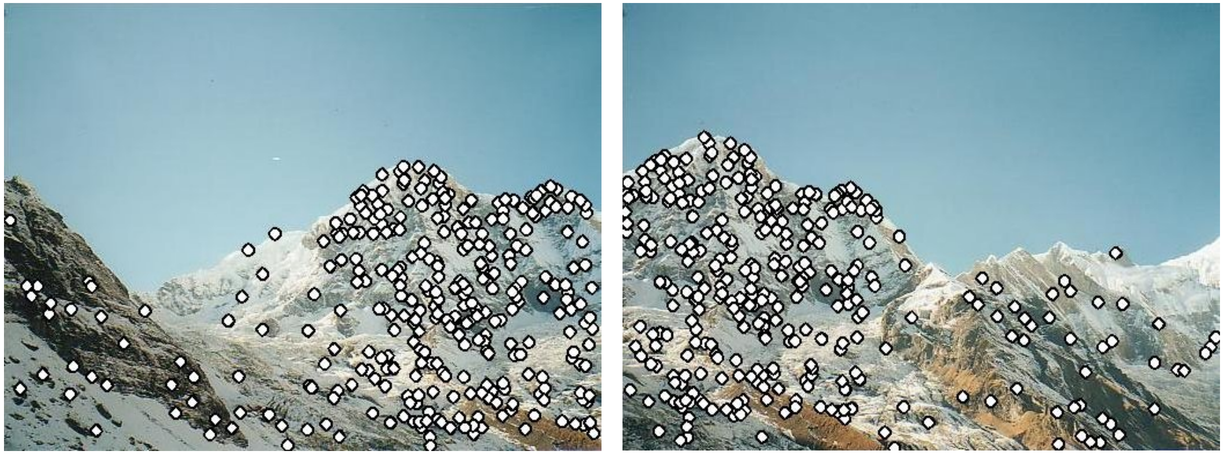
- Motivation: panorama stitching
  - We have two images – how do we combine them?

Step 1: extract local features

Step 2: match local features

Step 3: align images

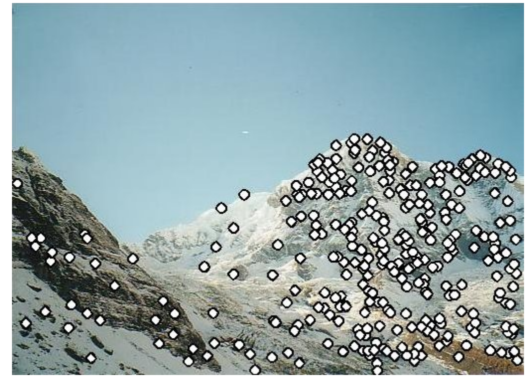
Characteristics of good local features



- Compactness and efficiency
  - Many fewer features than image pixels
- Saliency
  - Each local feature is distinctive
- Locality
  - A local feature occupies a relatively small area of the image; robust to clutter and occlusion
- Repeatability
  - The same local feature can be found in several images despite geometric and photometric transformations

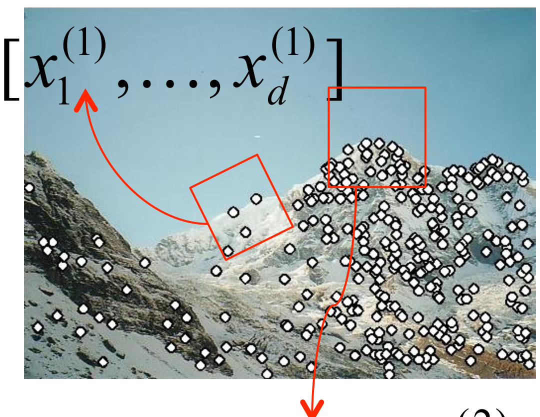
Local features: main components

1) **Detection:** Identify the interest points



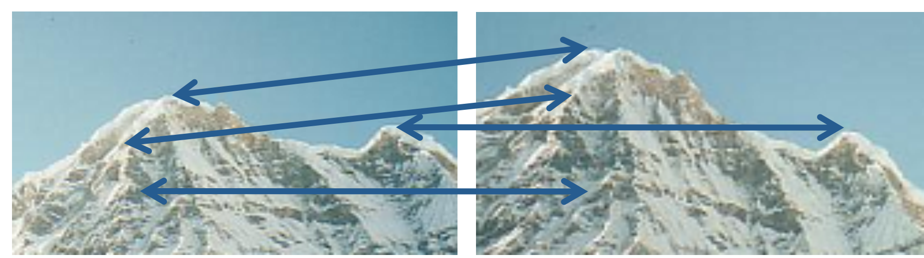
2) **Description:** Extract vector feature descriptor surrounding each interest point.

$$\mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]$$



$$\mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]$$

3) **Matching:** Determine correspondence between descriptors in two views



Goal: interest operator repeatability

- We want to detect (at least some of) the same points in both images.

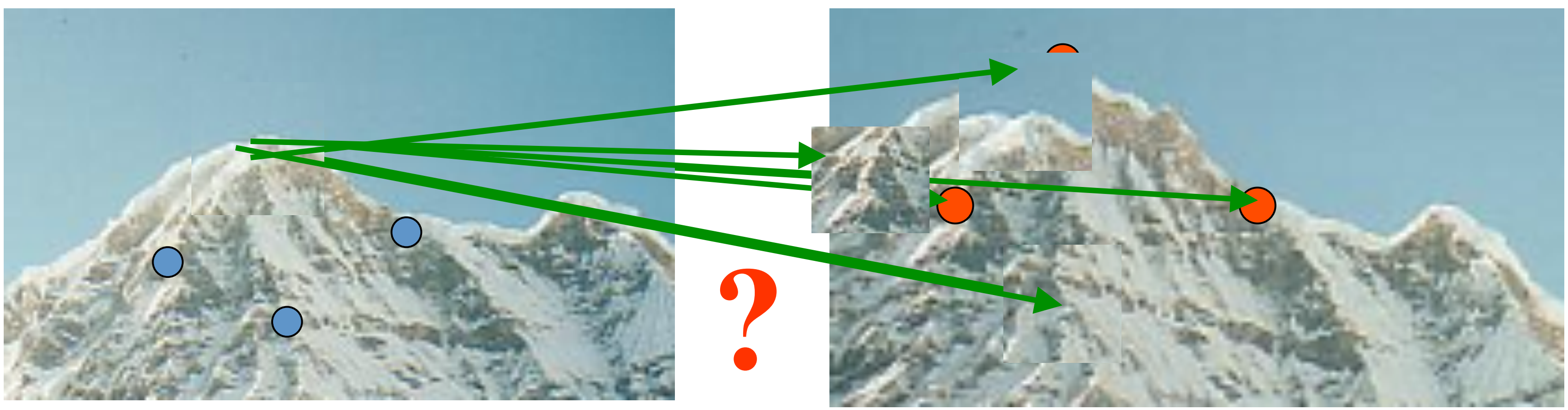
 

No chance to find true matches!

- Yet we have to be able to run the detection procedure *independently* per image.

Goal: descriptor distinctiveness

- We want to be able to reliably determine which point goes with which.



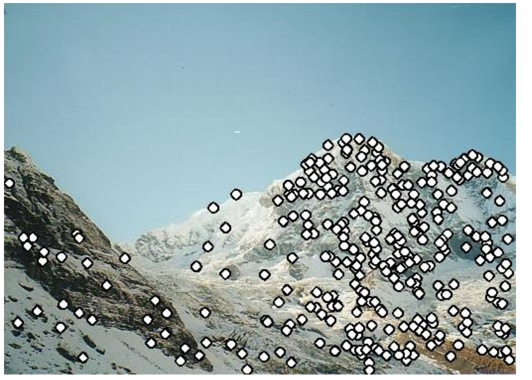
- Must provide some invariance to geometric and photometric differences between the two views.

Applications

- Local features are used for:
  - Image alignment
  - 3D reconstruction
  - Motion tracking
  - Robot navigation
  - Database indexing and retrieval
  - Object recognition (before Deep Learning)

Local features: main components

1) **Detection:** Identify the interest points



2) **Description**:Extract vector feature descriptor surrounding each interest point.

3) **Matching:** Determine correspondence between descriptors in two views

- What points would you choose?


1
5
3
4
6
2

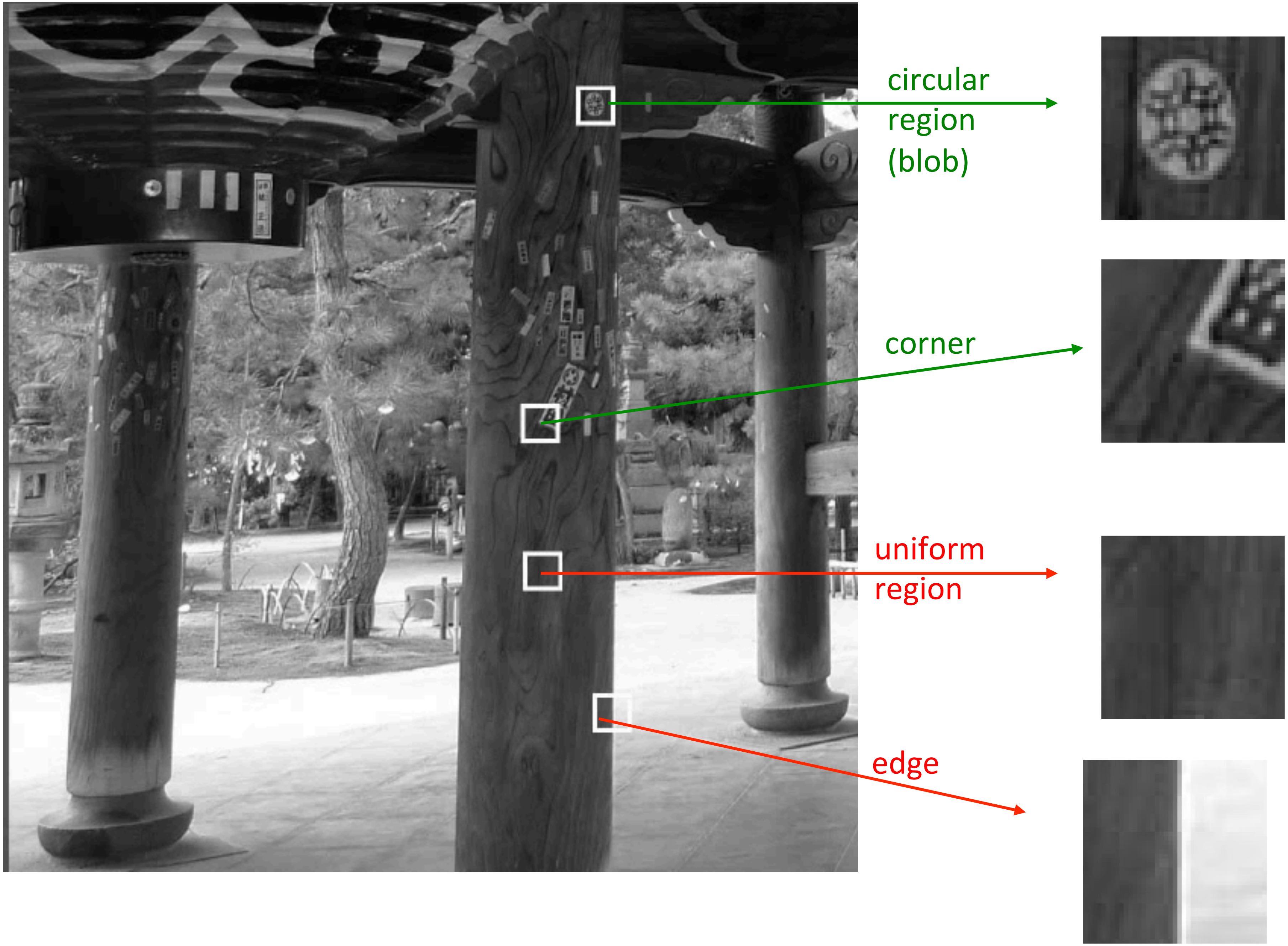
Detecting local invariant features

- Detection of interest points

Uniform regions
Ambiguous for
matching

Edges
Ambiguous for
matching

Corners
Good pentru
matching


circular
region
(blob)
corner
uniform
region
edge