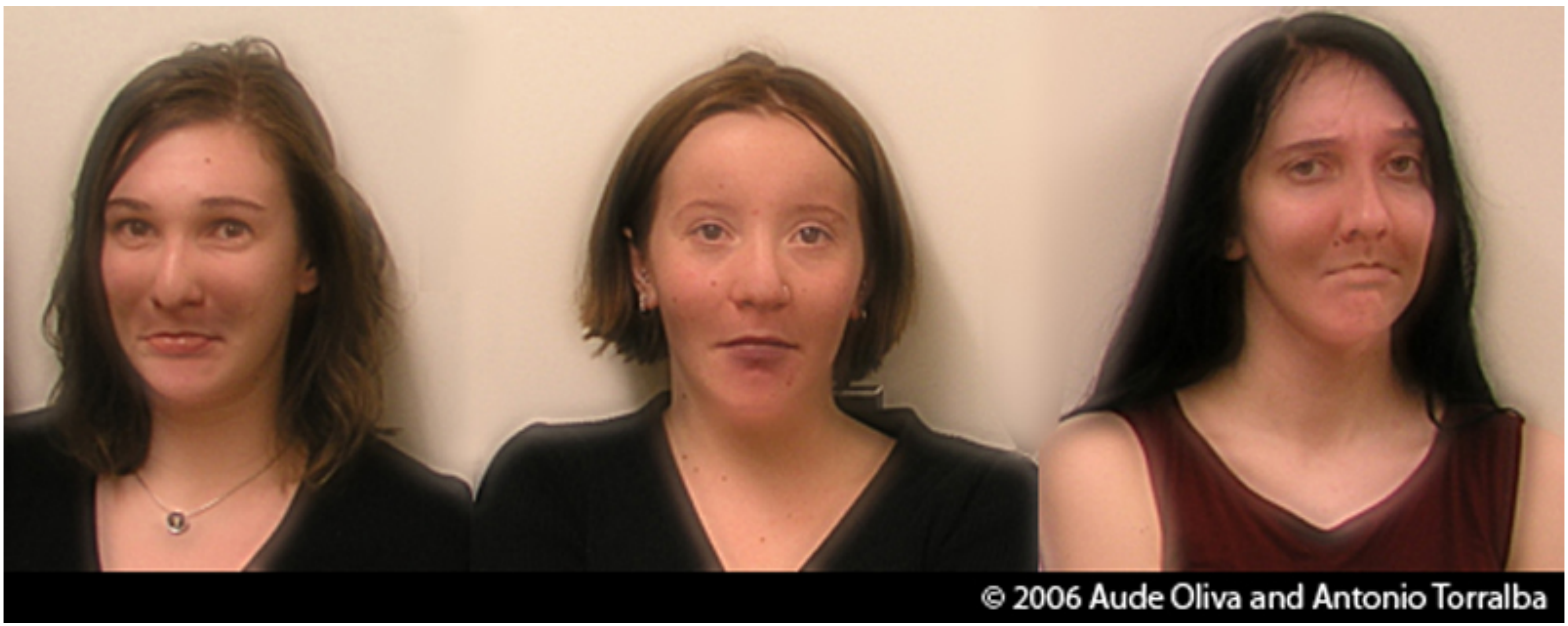
CS4670/5670: Intro to Computer Vision

Kavita Bala

Lecture 2: Images and image filtering

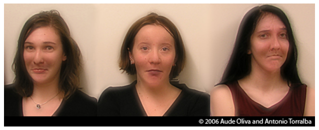


Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm

CS4670: Computer Vision

Kavita Bala

Lecture 2: Images and image filtering

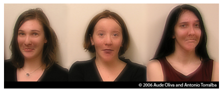


Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm

CS4670: Computer Vision

Kavita Bala

Lecture 2: Images and image filtering

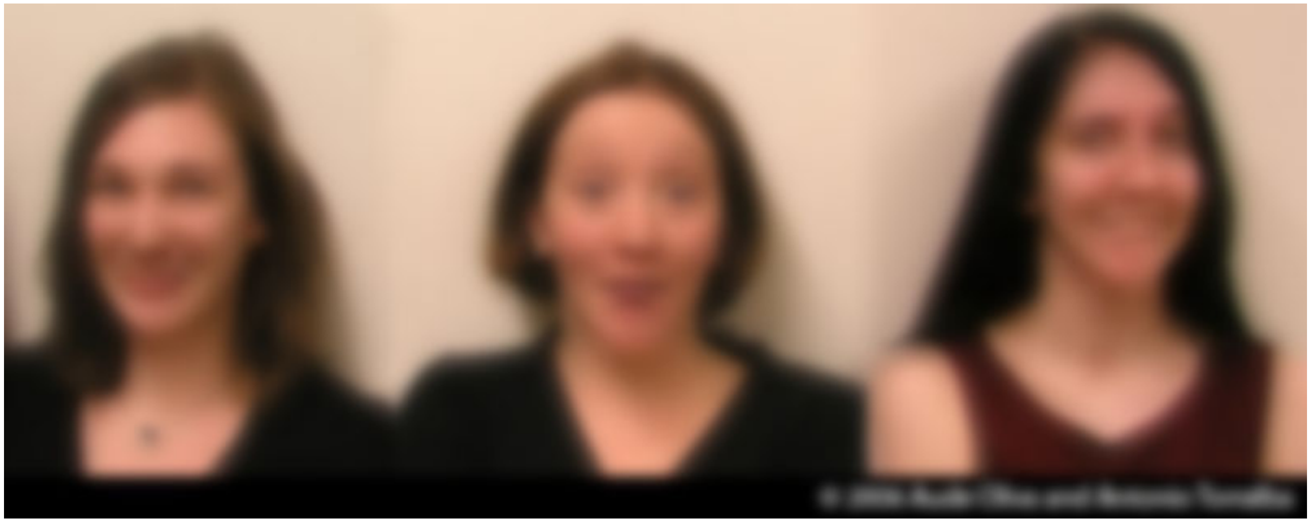


Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm

CS4670: Computer Vision

Kavita Bala

Lecture 2: Images and image filtering

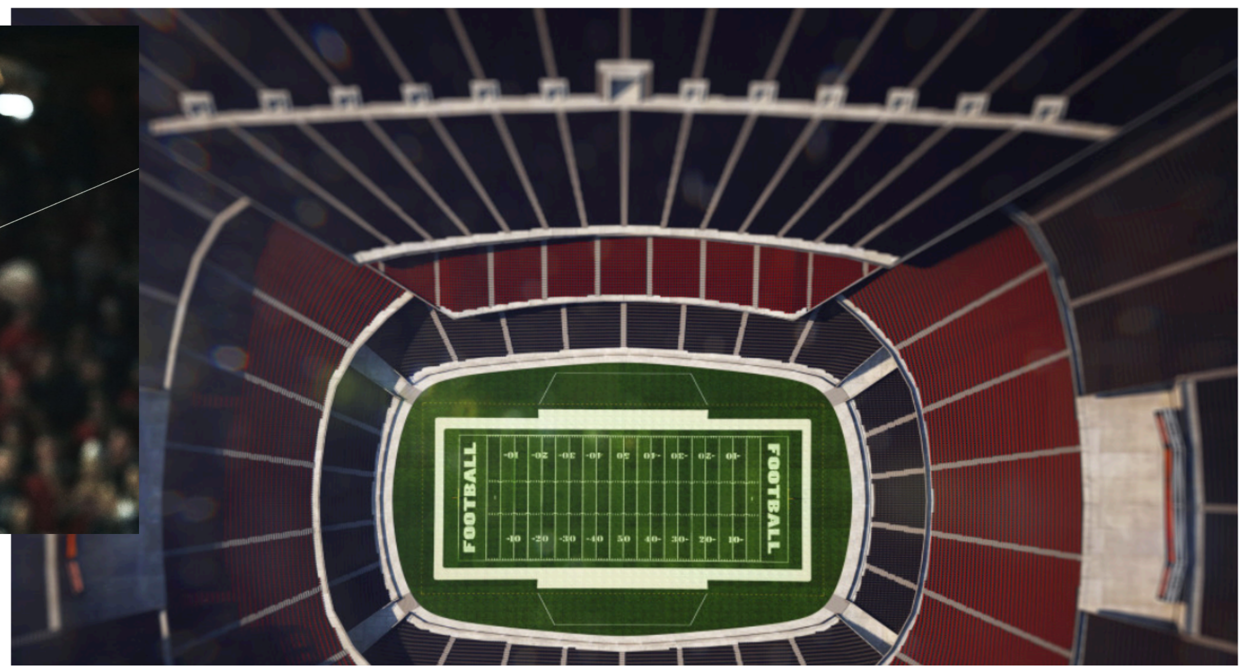


Hybrid Images, Oliva et al., http://cvcl.mit.edu/hybridimage.htm

Reading and Announcements

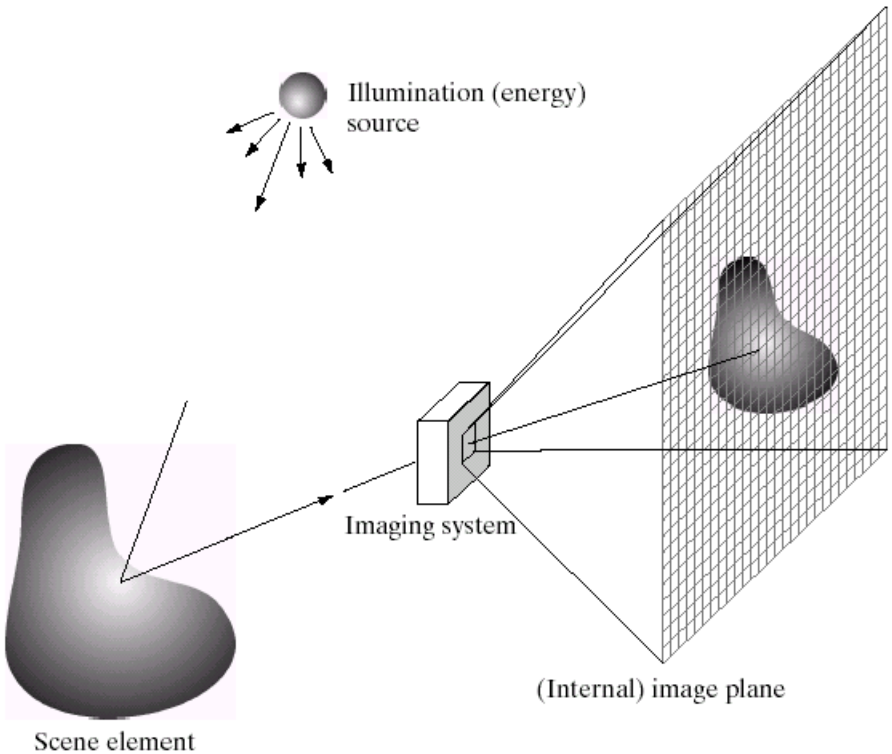
- Szeliski, Chapter 3.1-3.2

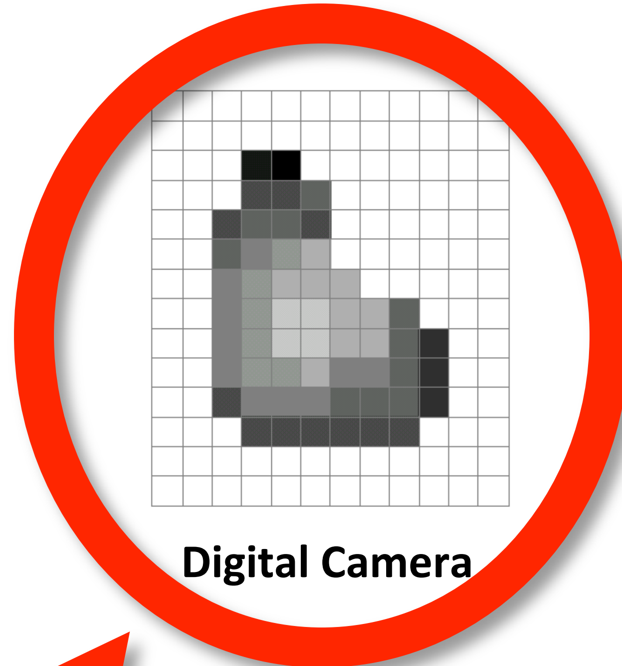
THE CAMERAS THAT'LL MAKE THE SUPER BOWL WAY MORE INTERESTING THIS YEAR

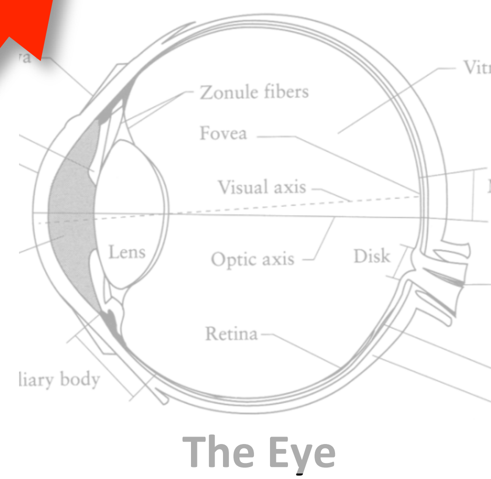


What is an image?

What is an image?



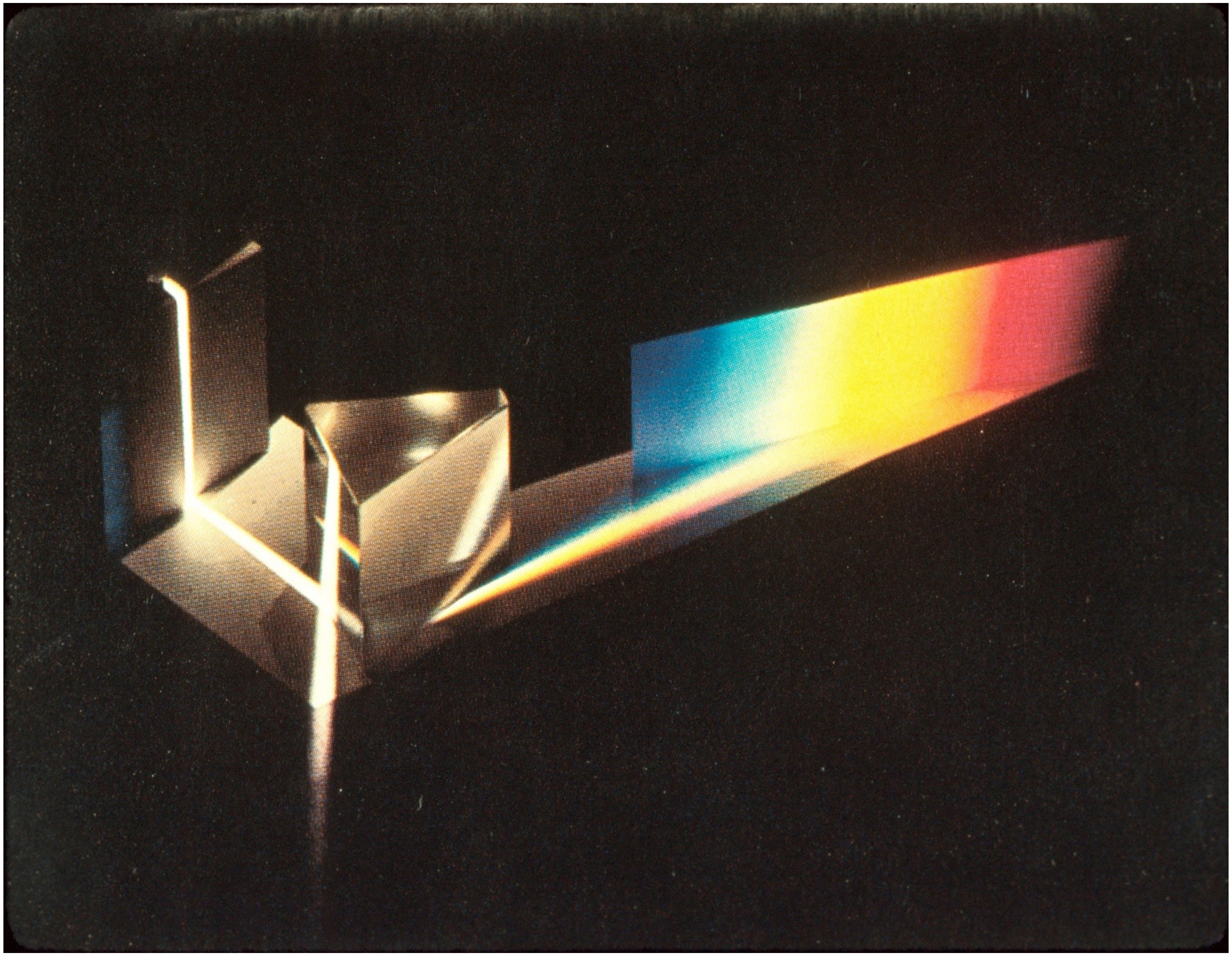




We'll focus on these in this class

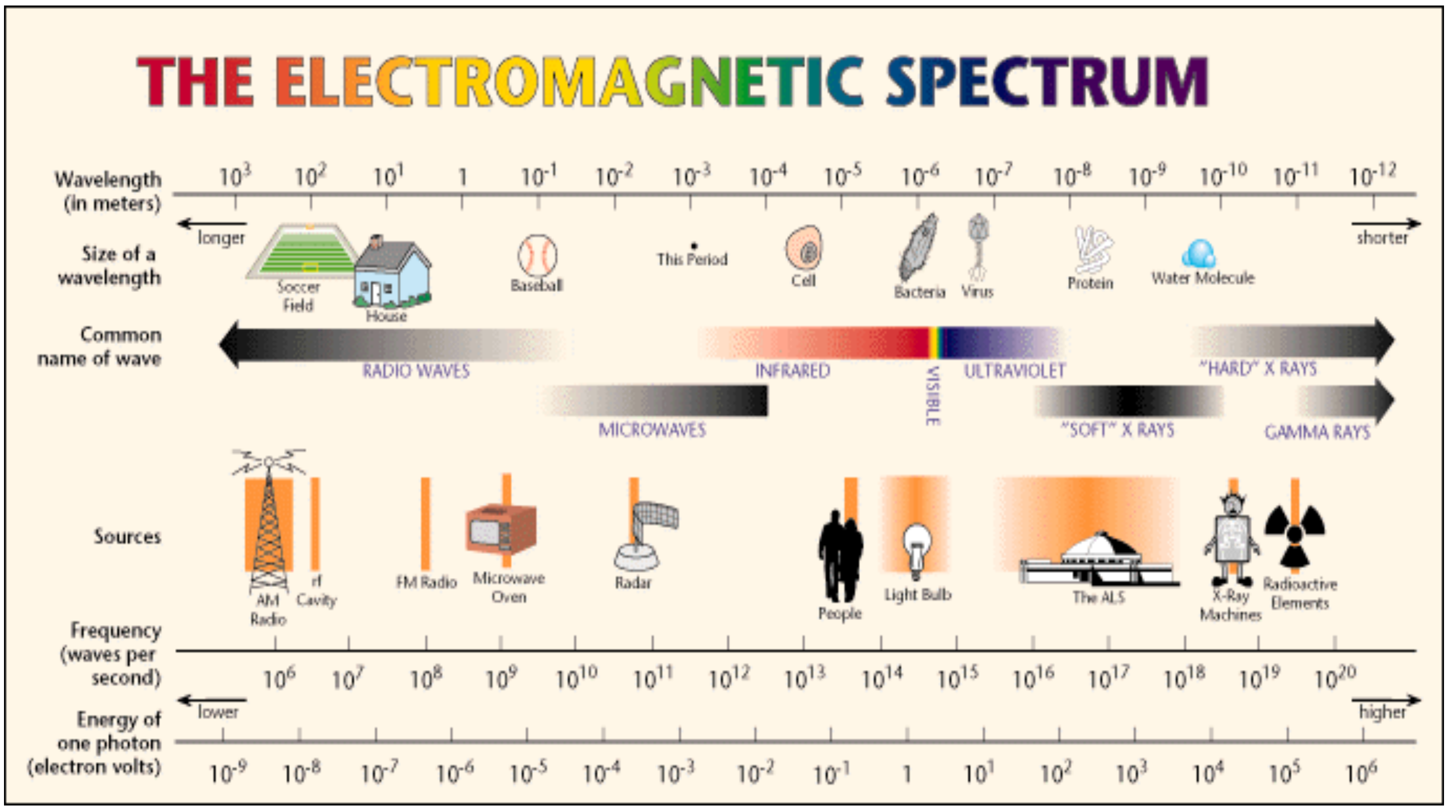
(More on this process later)

Source: A. Efros



What light is

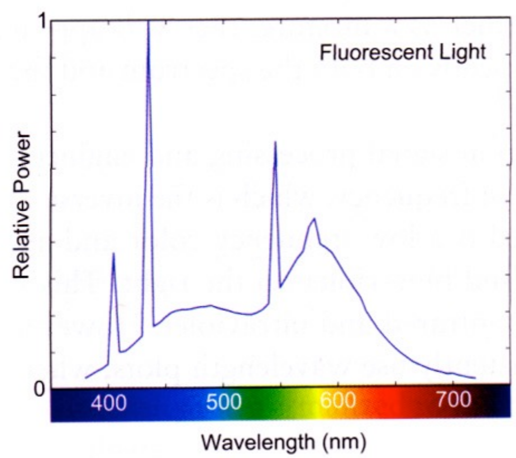
- Light is electromagnetic radiation



[Lawrence Berkeley Lab / MicroWorlds]

Physics to Brain

[Stone 2003]

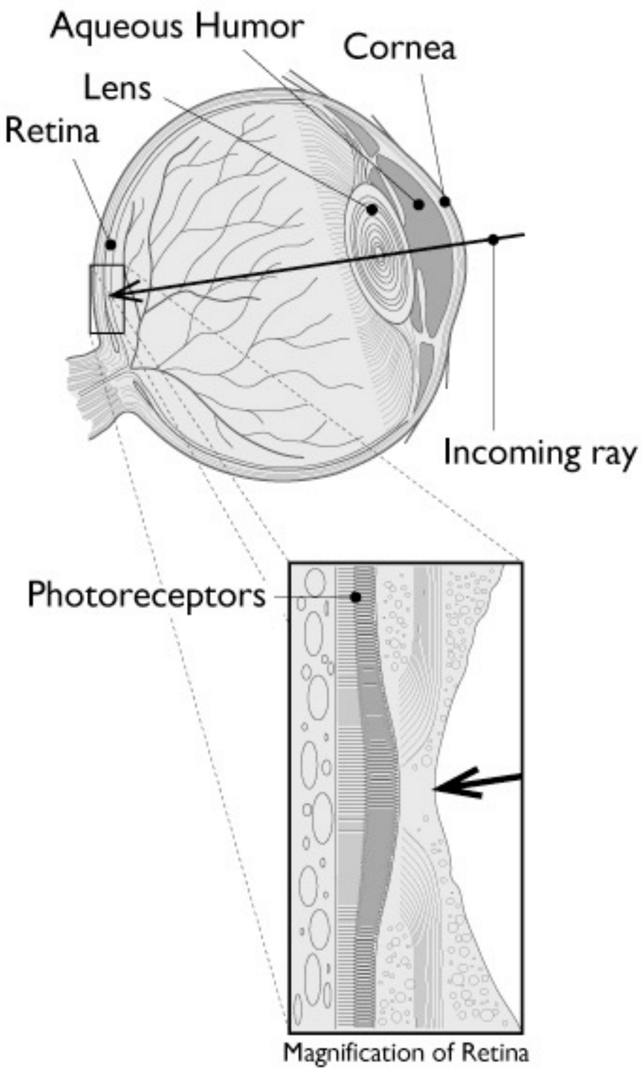


Physical

?

Perceptual

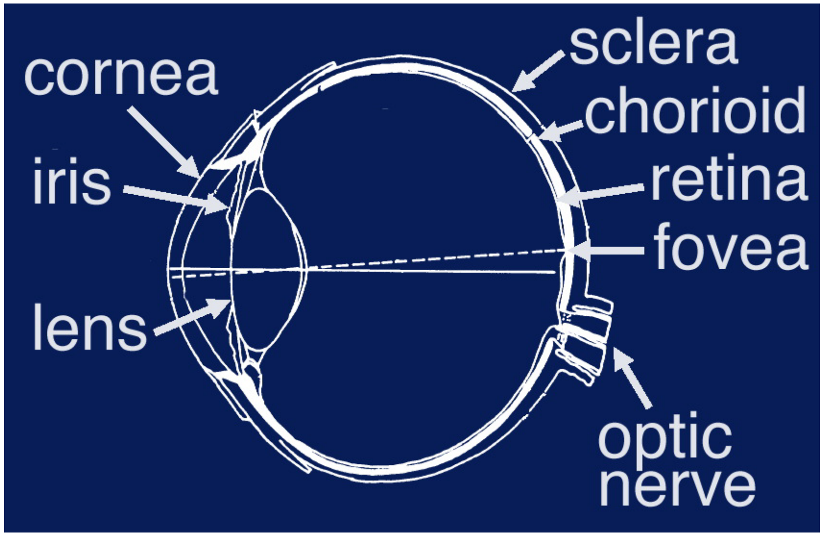
The eye as a measurement device



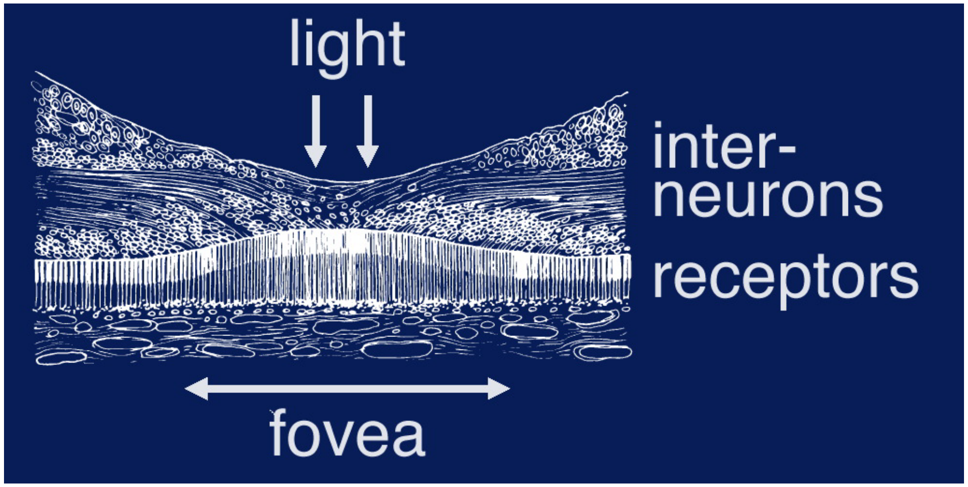
[Greger et al. 1995]

- We can model the low-level behavior of the eye by thinking of it as a light-measuring machine
  - its optics are much like a camera
  - its detection mechanism is also much like a camera
- Light is measured by the *photoreceptors* in the retina
  - they respond to visible light
  - different types respond to different wavelengths

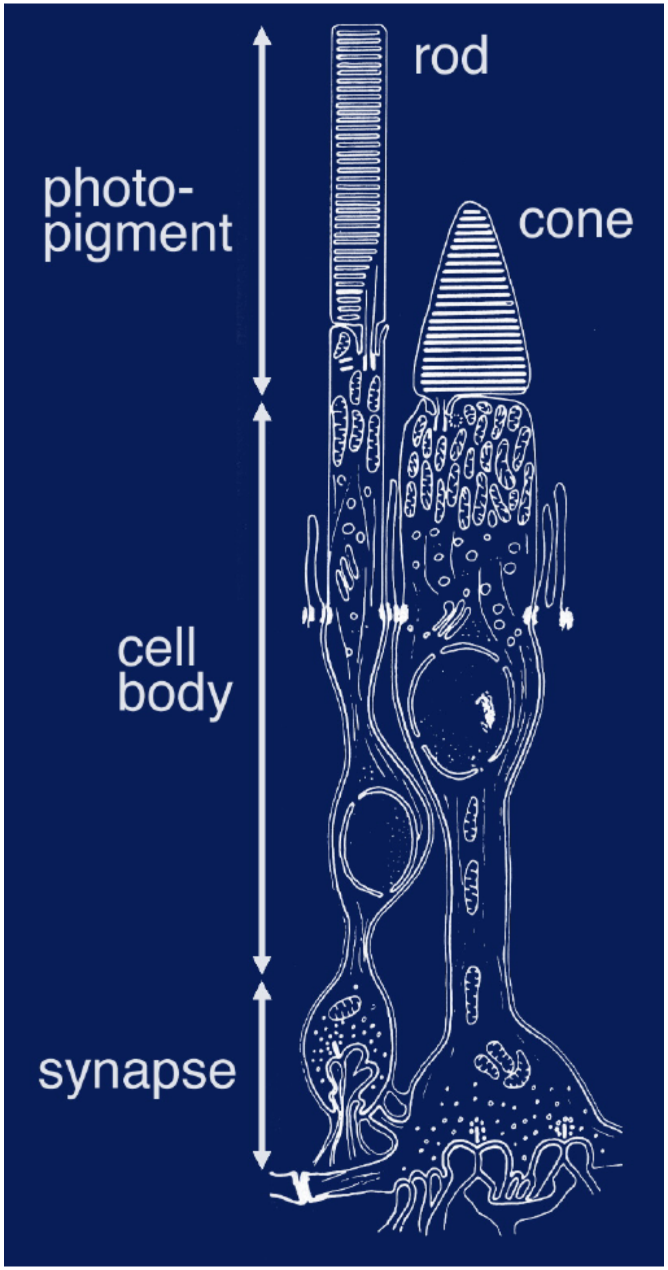
The eye



cornea
iris
lens
sclera
chorioid
retina
fovea
optic nerve



light
inter-neurons
receptors
fovea



Photoreceptors

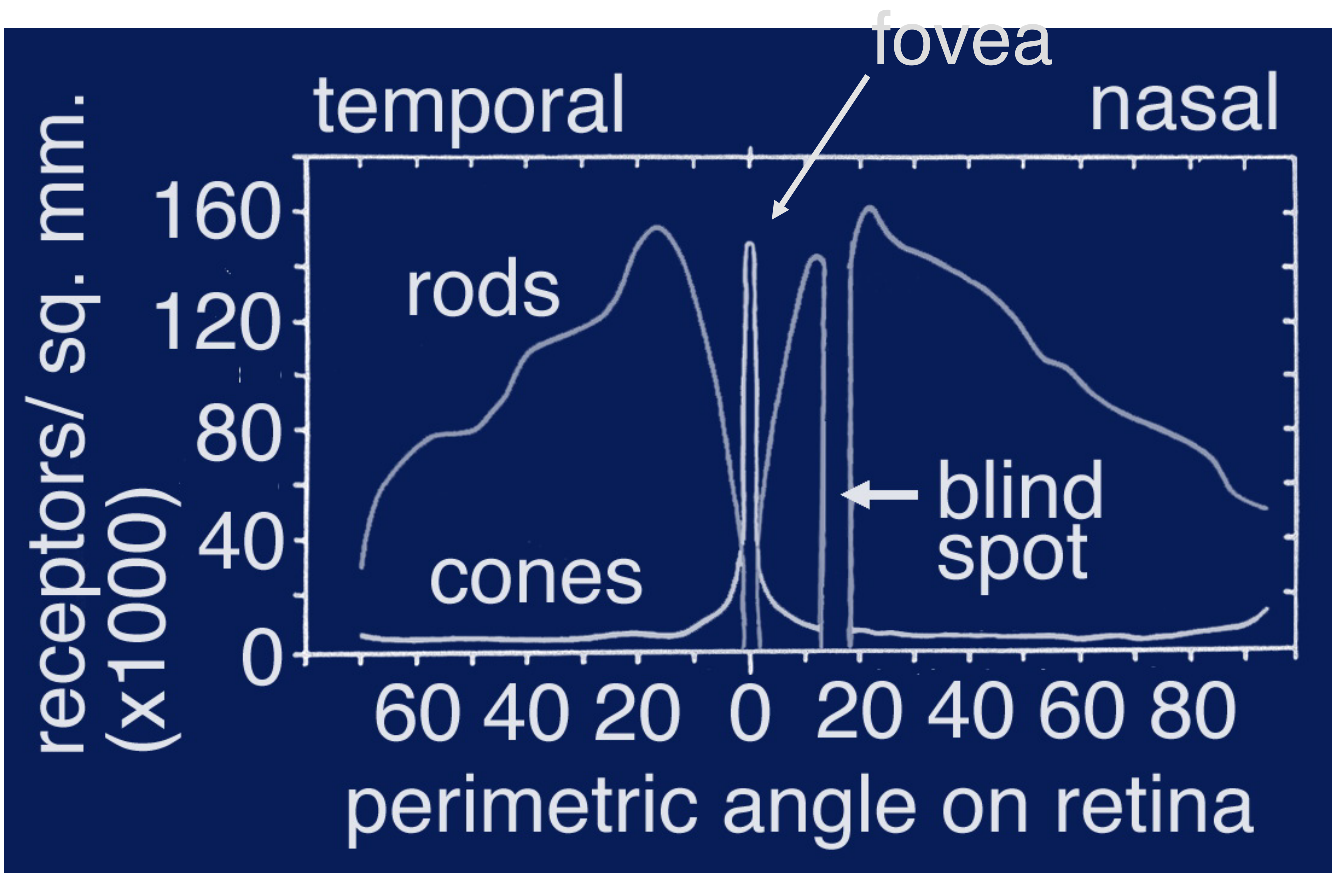
- 120 million rods
- 7-8 million cones

  in each eye

- rods: scotopic
- cones: photopic

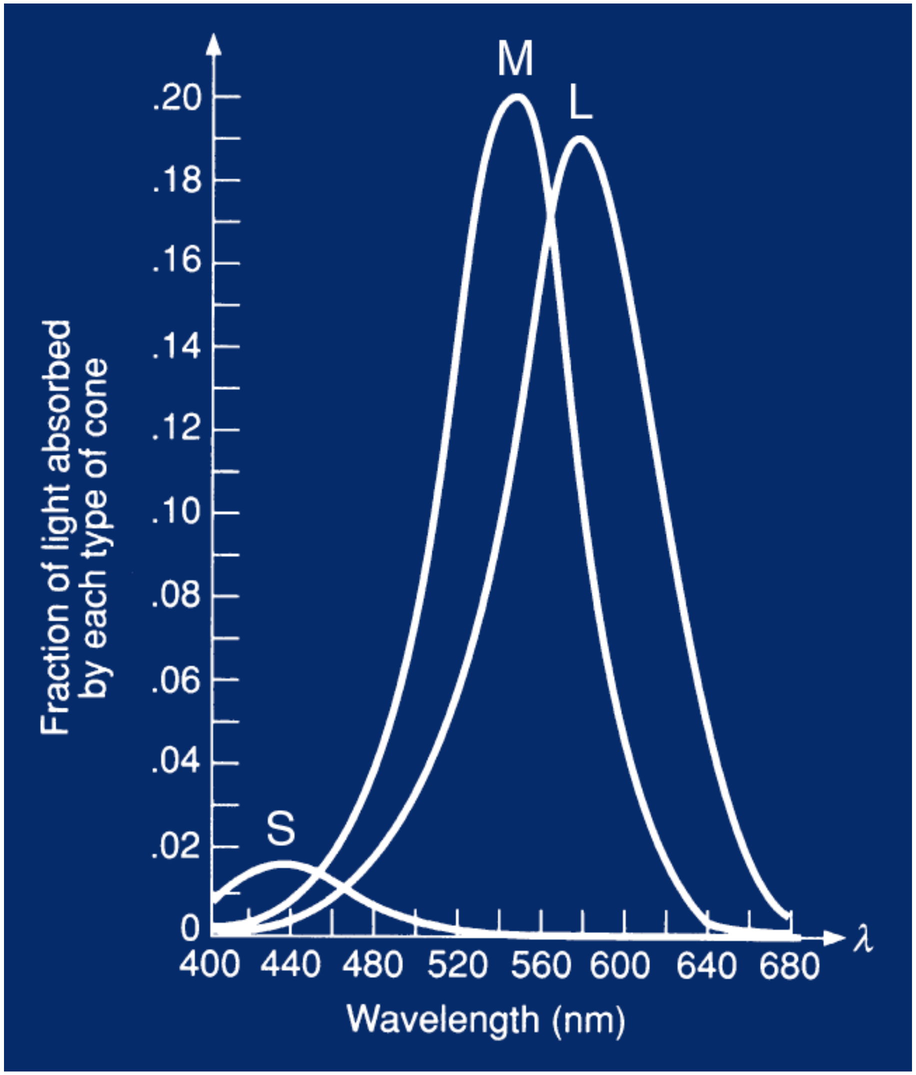
Adapted from Levine, Vision in Man and Machine © McGraw-Hill, 1985.

Receptor distribution



Adapted from Levine, Vision in Man and Machine © McGraw-Hill, 1985.

Cone Responses

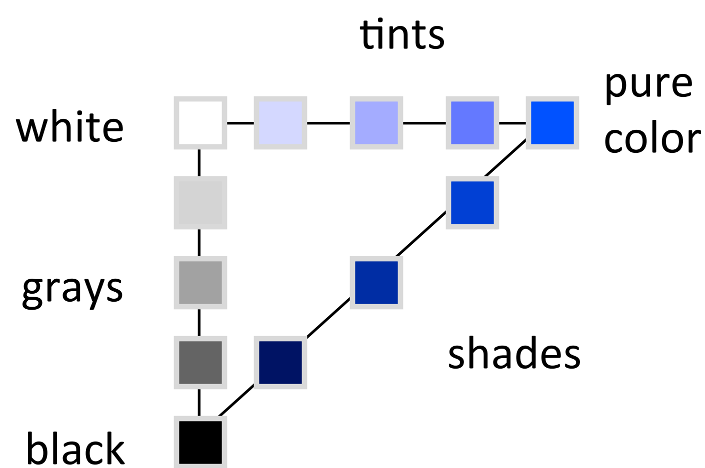


- S,M,L cones have broadband spectral sensitivity
- Results in a trichromatic visual system
- S, M, and L are *tristimulus values*

[source unknown]

Color perception

- Artists often refer to colors as *tints*, *shades*, and *tones* of pure pigments
  - tint: mixture with white
  - shade: mixture with black
  - tones: mixture with black and white
  - gray: no color at all (aka. neutral)



[after FvDFH]

- This seems intuitive
  - tints and shades are inherently related to the pure color
    - "same" color but lighter, darker, paler, etc.

Perceptual dimensions of color

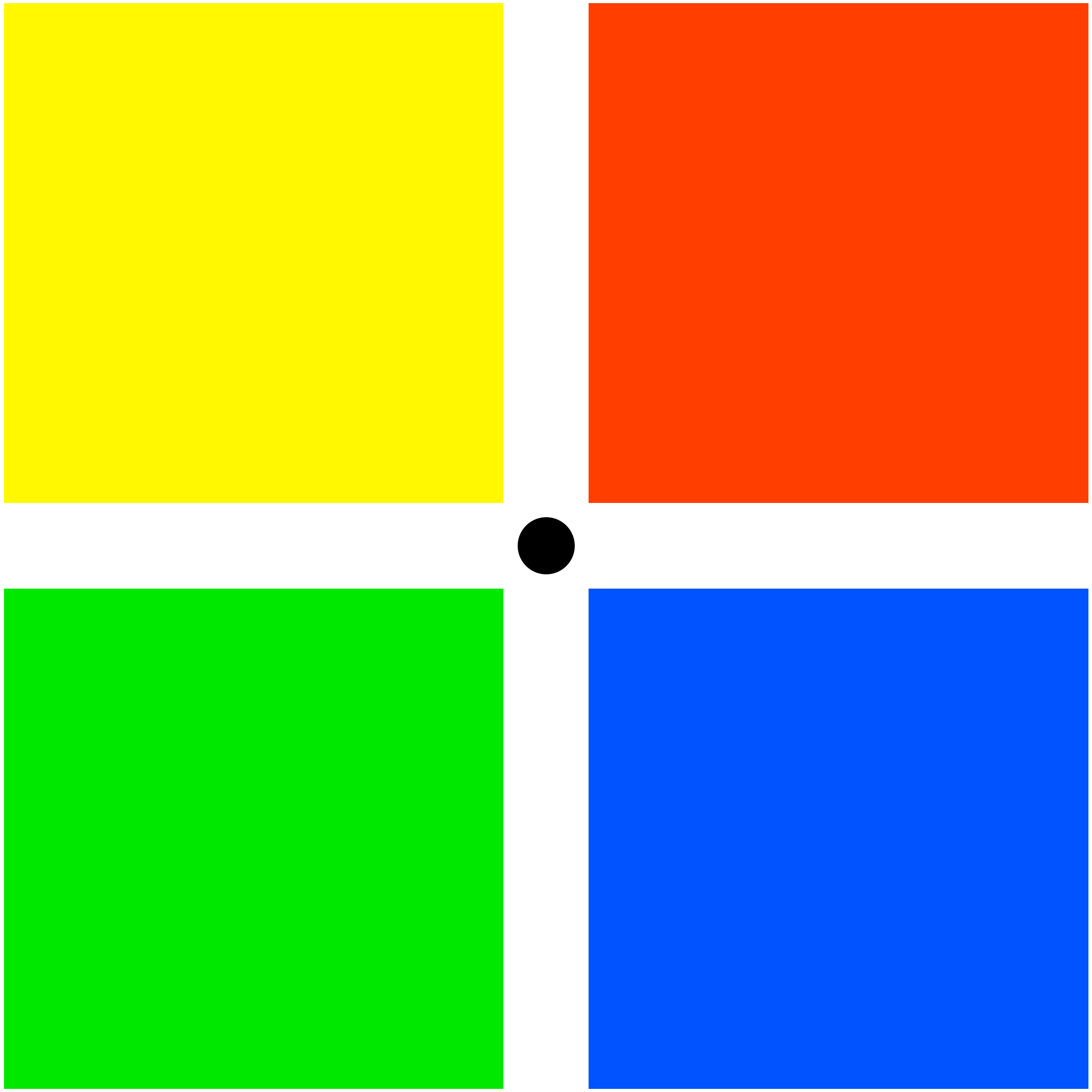
- Hue
  - the “kind” of color, dominant wavelength
  - artist’s correlate: the chosen pigment color
- Saturation
  - the “colorfulness”
  - artist’s correlate: fraction of paint from the colored tube
- Lightness (or value)
  - the overall amount of light
  - artist’s correlate: tints are lighter, shades are darker

Perceptual dimensions of color

- There's good evidence ("opponent color theory") for a neurological basis for these dimensions
  - the brain seems to encode color early on using three axes:

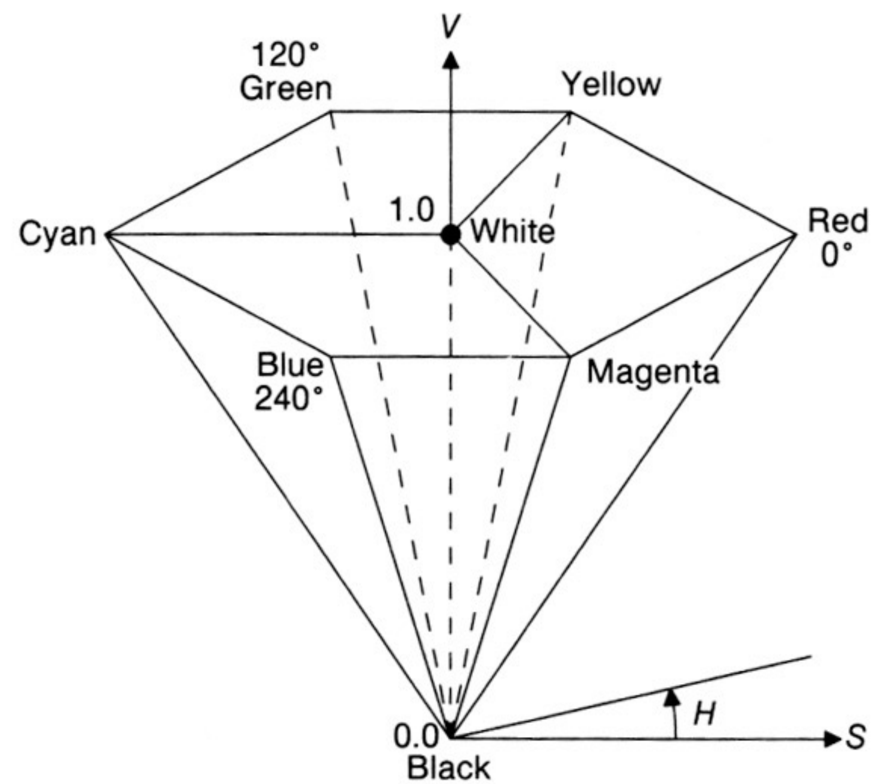
  white — black,   red — green,   yellow — blue

  - the white—black axis is lightness; the others determine hue and saturation
  - one piece of evidence: you can have a light green, a dark green, a yellow-green, or a blue-green, but you can't have a reddish green (just doesn't make sense)
    - thus red is the *opponent* to green
  - another piece of evidence: afterimages (next slide)



Perceptual organization for RGB: HSV

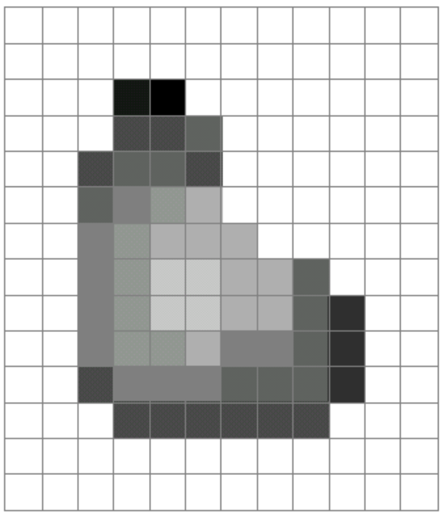
- Uses hue (an angle, 0 to 360), saturation (0 to 1), and value (0 to 1) as the three coordinates for a color
  - –the brightest available RGB colors are those with one of R,G,B equal to 1 (top surface)



[FvDFH]

What is an image?

- A grid (matrix) of intensity values



=

255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

(common to use one byte per value: 0 = black, 255 = white)

Images as functions

- An image contains discrete numbers of pixels

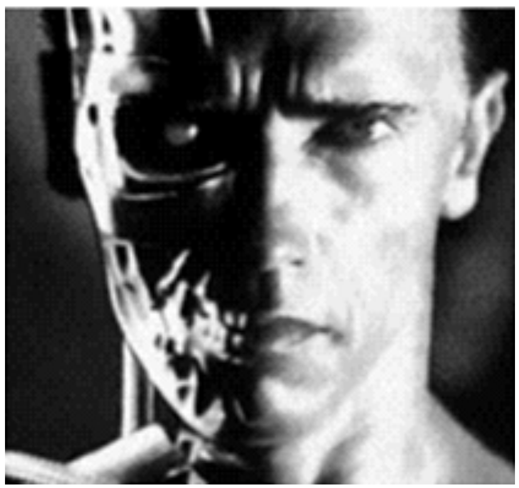
- Pixel value
  - grayscale/intensity
    - [0,255]
  - Color
    - RGB [R, G, B], where [0,255] per channel
    - Lab [L, a, b]: Lightness, a and b are color-opponent dimensions
    - HSV [H, S, V]: Hue, saturation, value

Images as functions

- Can think of image as a **function**, f, from R^2 to R or R^{M}:
  - Grayscale: $f(x,y)$ gives **intensity** at position (x,y)
    - f: [a,b] x [c,d] $\rightarrow$ [0,255]
  - Color: $f(x,y) = [r(x,y), g(x,y), b(x,y)]$

What is an image?

A **digital** image is a discrete (**sampled**, **quantized**) version of this function



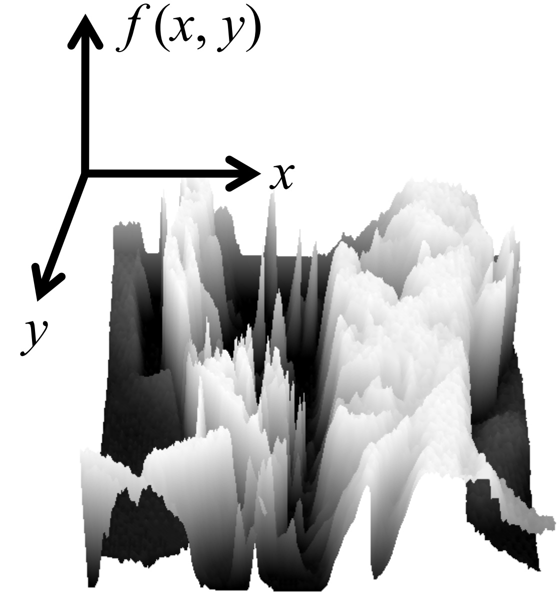
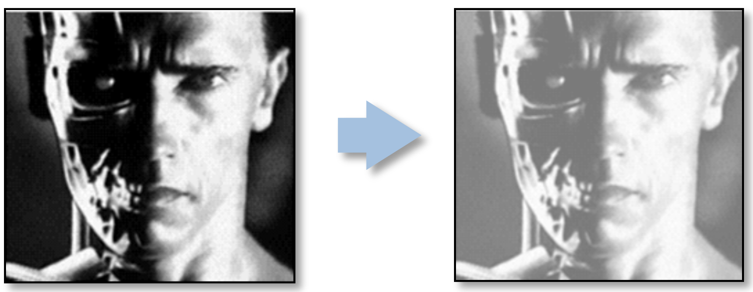
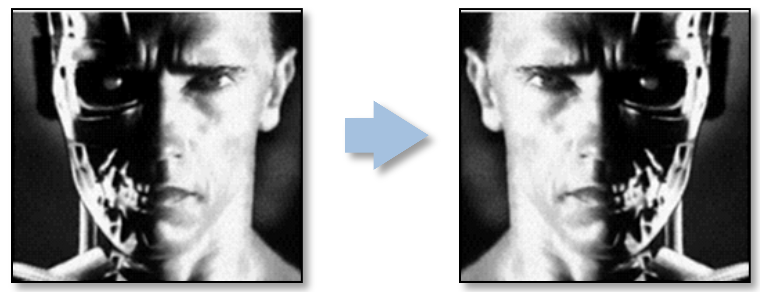


Image transformations

- As with any function, we can apply operators to an image



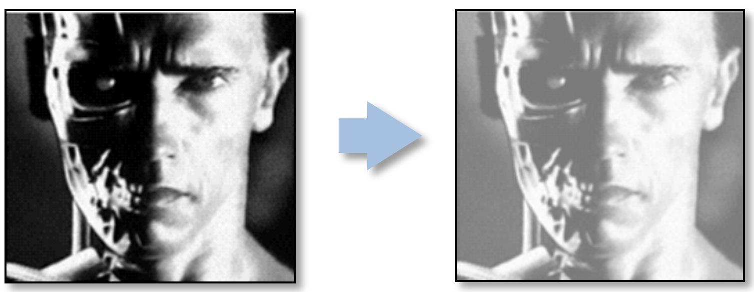
$g(x,y) = f(x,y) + 20$



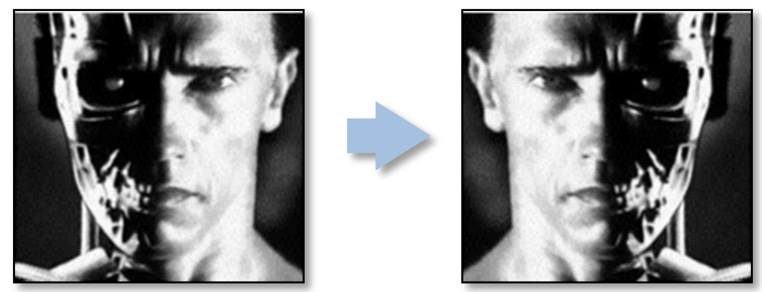
$g(x,y) = f(-x,y)$

Image transformations

- As with any function, we can apply operators to an image



$g(x,y) = f(x,y) + 20$



$g(x,y) = f(-x,y)$

Filters

- Filtering
  - Form a new image whose pixels are a combination of the original pixels
- Why?
  - To get useful information from images
    - E.g., extract edges or contours (to understand shape)
  - To enhance the image
    - E.g., to blur to remove noise
    - E.g., to sharpen to “enhance image” a la CSI


Dave
Brand New

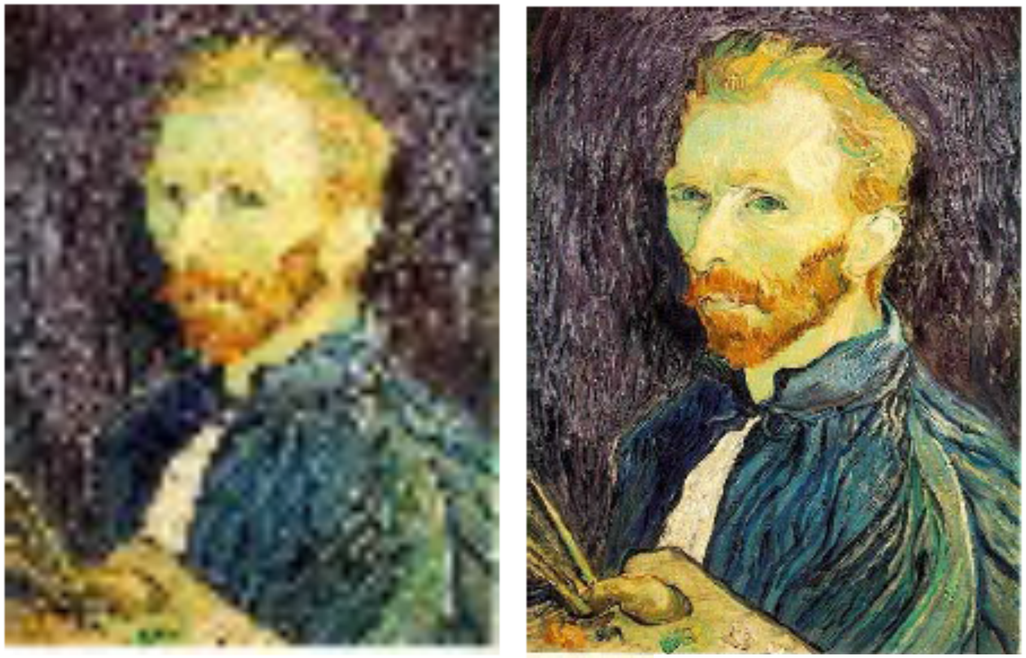
Canonical Image Processing problems

- Image Restoration
  - denoising
  - deblurring
- Image Compression
  - JPEG, JPEG2000, MPEG..
- Computing Field Properties
  - orientation
  - optical flow
  - disparity
- Locating Structural Features
  - corners
  - edges

Computing field properties

these are defined at every pixel (x,y)

- Orientation
  - at every pixel, one can define a local orientation by computing the gradient of the image
- Optical Flow
  - at every pixel, a vector corresponding to the movement from one time frame to the next
- Binocular Disparity
  - at every pixel, a vector corresponding to the displacement of the corresponding point from the left to the right image



Super-resolution

Noise reduction

Noise reduction

- Given a camera and a still scene, how can you reduce noise?

Take lots of images and average them!

How to formulate as filtering?

Source: S. Seitz

Image filtering

- Modify the pixels in an image based on some function of a local neighborhood of each pixel

10	5	3
4	5	1
1	1	7

Local image data

Some function S

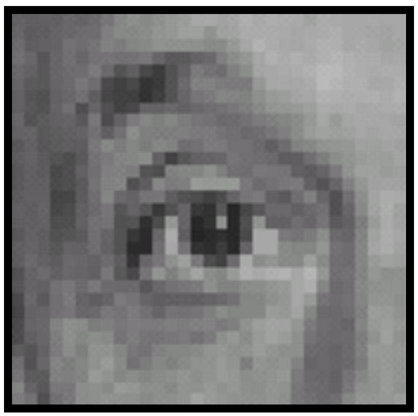
	7	

Modified image data

$$f[m,n] \rightarrow S \rightarrow g[m,n]$$

Source: L. Zhang

Filters: examples



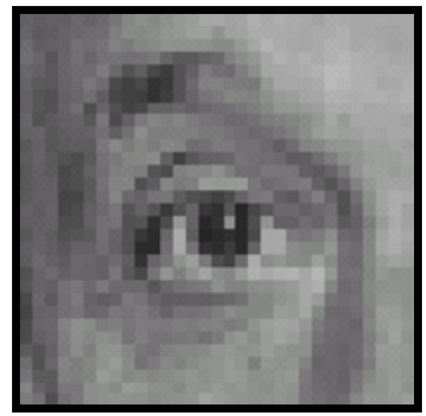
Original (f)

*

0	0	0
0	1	0
0	0	0

Kernel (k)

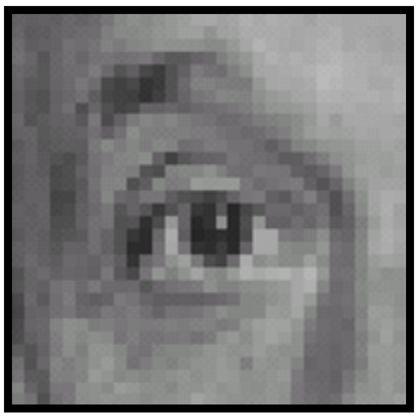
=



Identical image (g)

Source: D. Lowe

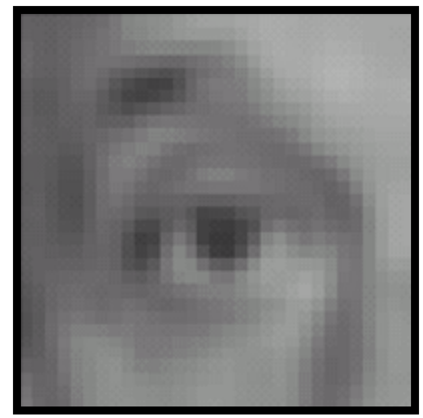
Filters: examples



Original (f)

$$* \ \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} =$$

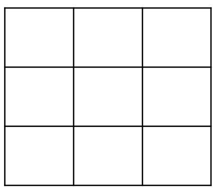
Kernel (k)



Blur (with a mean filter) (g)

Source: D. Lowe

Mean filtering



H

$*$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

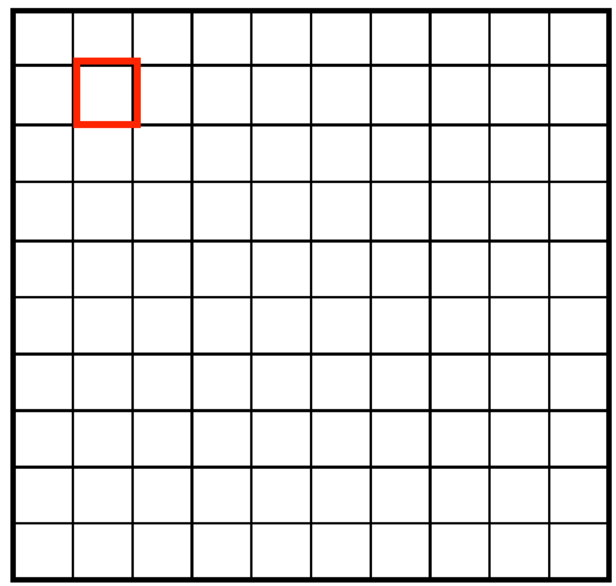
F

Mean filtering/Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$



Mean filtering/Moving average

$F[x,y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x,y]$

	0	10							

Mean filtering/Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20						

Mean filtering/Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30					

Mean filtering/Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30	30				

Mean filtering/Moving average

$F[x, y]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$G[x, y]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Mean filtering

H

$*$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

F

$=$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

G

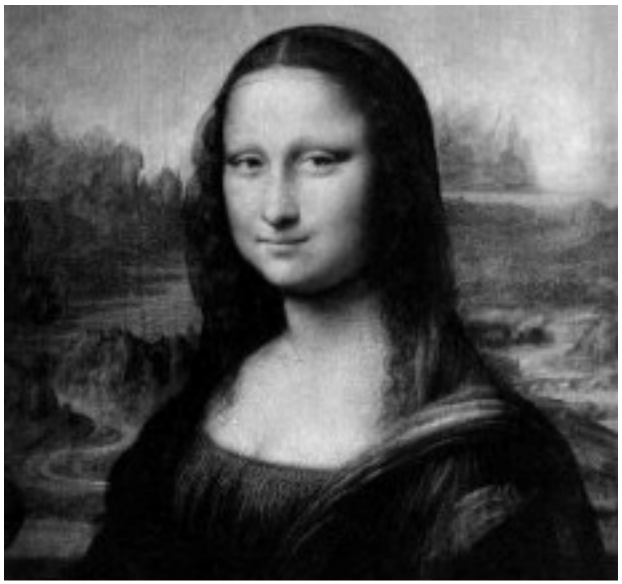
Mean filtering/Moving Average

- Replace each pixel with an average of its neighborhood

- Achieves smoothing effect
  - Removes sharp features

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filters: Thresholding



$$g(m,n) = \begin{cases} 255, & f(m,n) > A \\ 0 & otherwise \end{cases}$$

Linear filtering

- One simple version: linear filtering
  - Replace each pixel by a linear combination (a weighted sum) of its neighbors
  - Simple, but powerful
  - Cross-correlation, convolution
- The prescription for the linear combination is called the “kernel” (or “mask”, “filter”)

10	5	3
4	6	1
1	1	8

Local image data

0	0	0
0	0.5	0
0	1	0.5

kernel

	8	

Modified image data

Source: L. Zhang

Filter Properties

- Linearity
  - Weighted sum of original pixel values
  - Use same set of weights at each point
  - S[f + g] = S[f] + S[g]
  - S[k f + m g] = k S[f] + m S[g]

Linear Systems

- Is mean filtering/moving average linear?

- Is thresholding linear?