Designing Applications that See
Lecture 3: Image Processing

Dan Maynes-Aminzade

15 January 2008

Reminders

- Register on Axess
- Assignment #1 out today, due next Tuesday in lecture
- Newsgroup: su.class.cs377s
- Next lecture is an interactive workshop, so bring your webcam if you have one
- Remember to check the course calendar for the latest readings

Today’s Goals

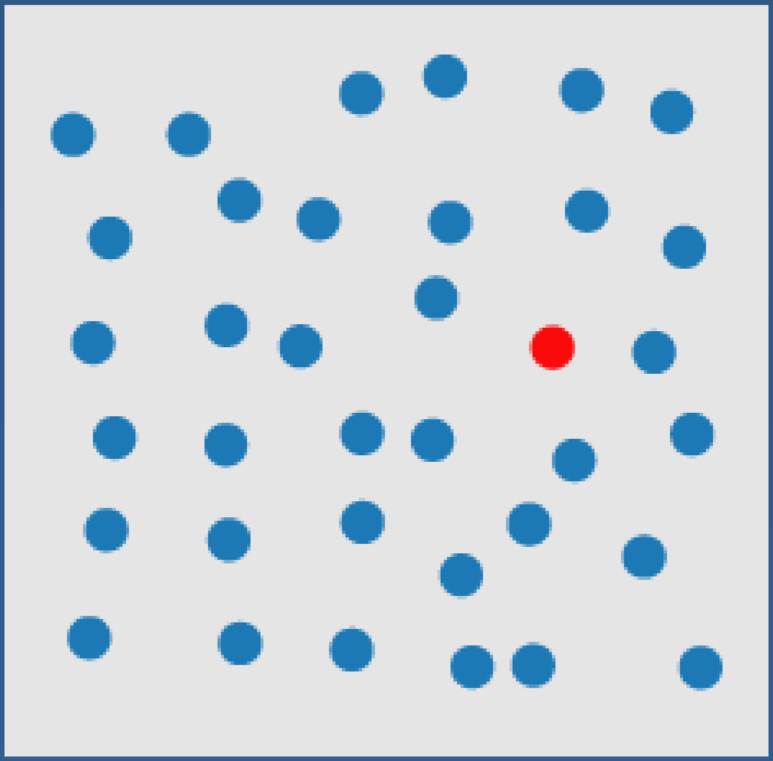
- Get an overview of various low-level image processing techniques
- Understand what they are good for and when to use them
- Next lecture: hands-on demo of these techniques in action

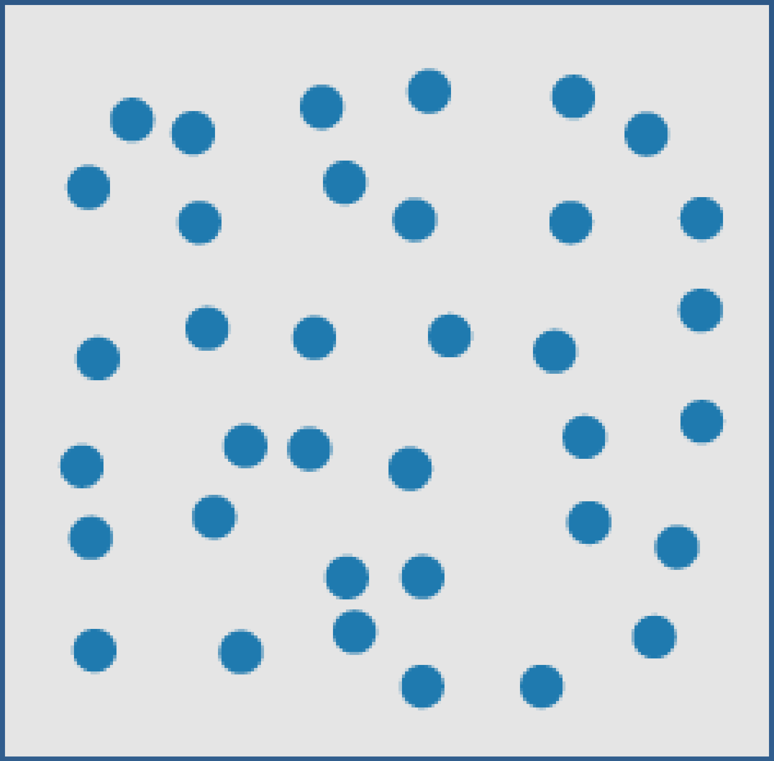
Outline

- Image basics
- Color
- Image filters
- Features
- Shapes

“Image Processing”

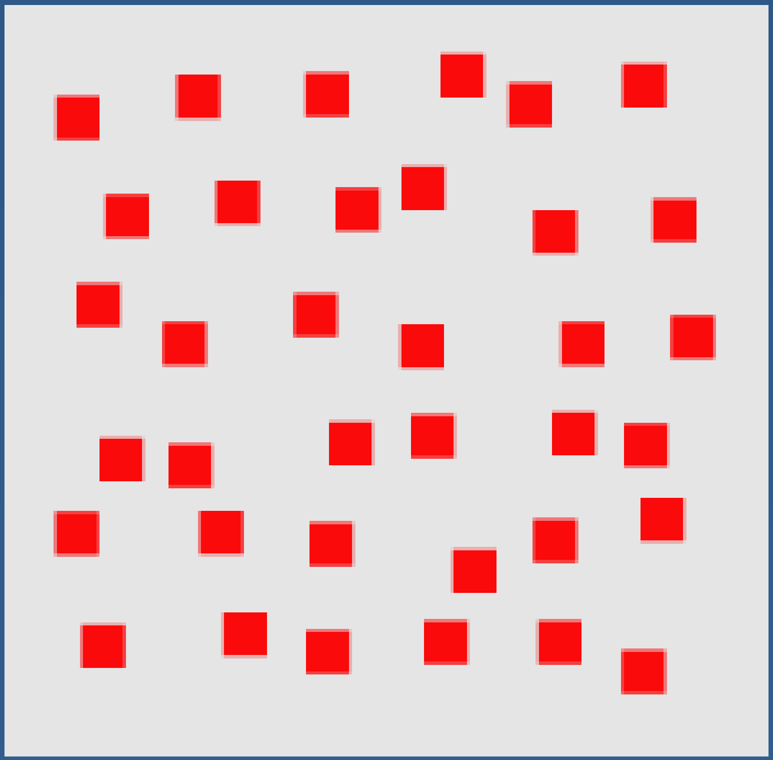
- Which of these two images contains a red circle?

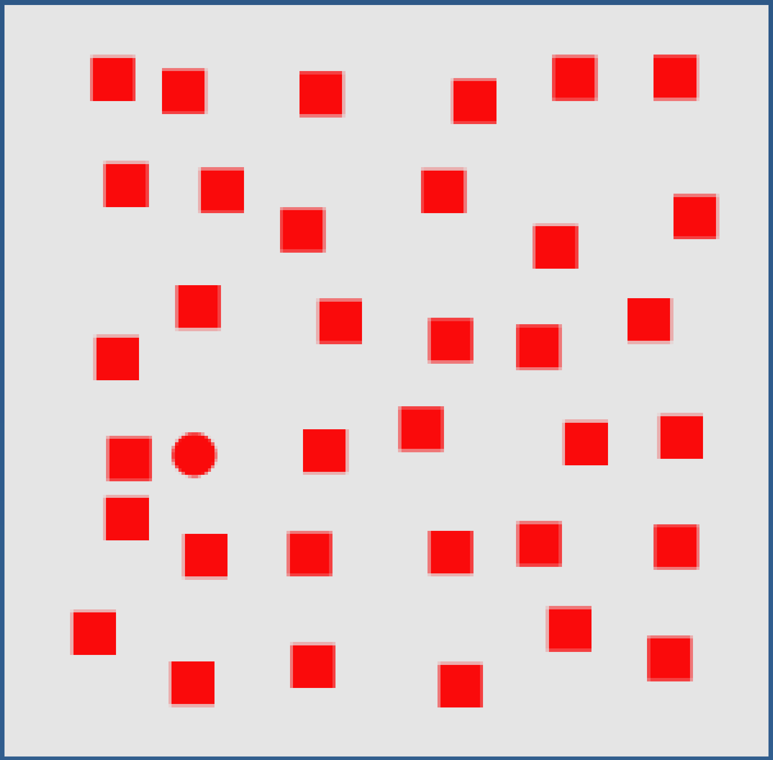




“Image Processing”

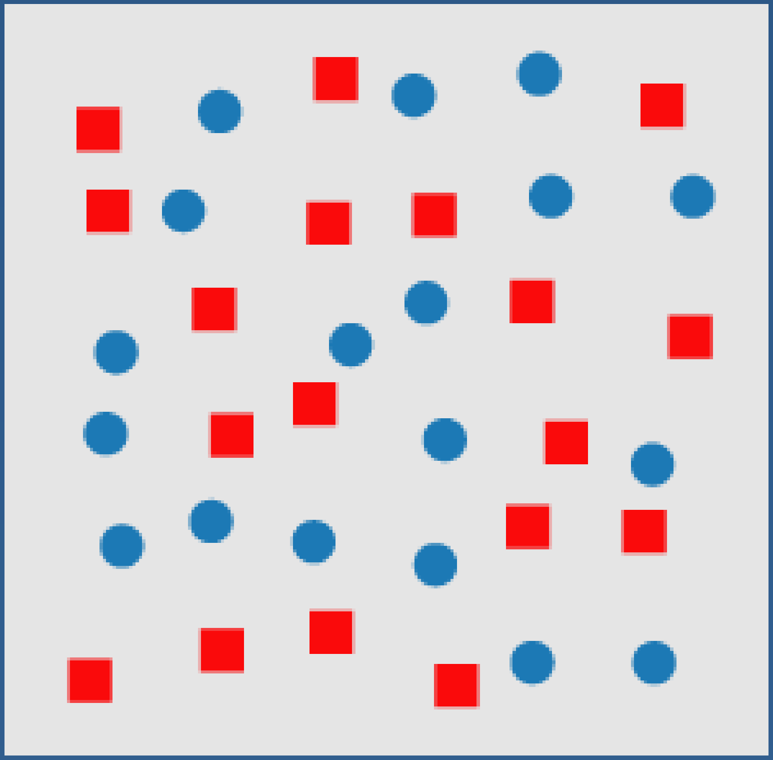
- Which of these two images contains a red circle?

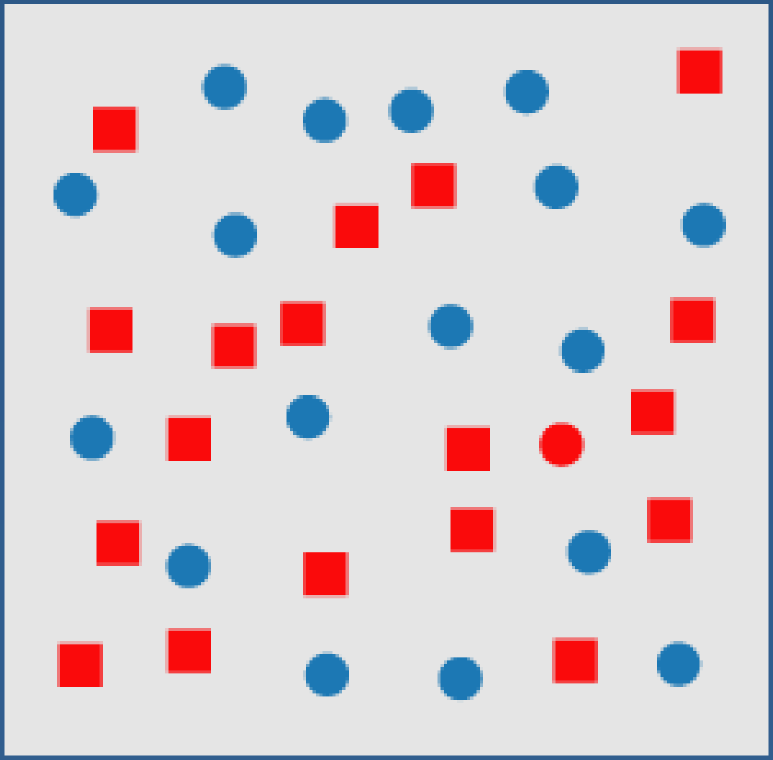




“Image Processing”

- Which of these two images contains a red circle?

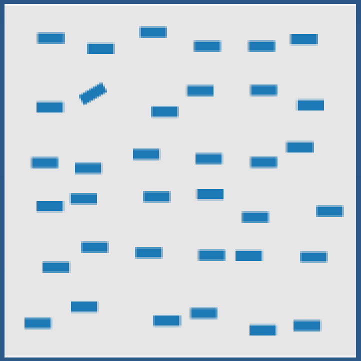


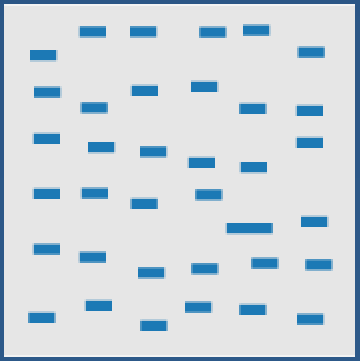


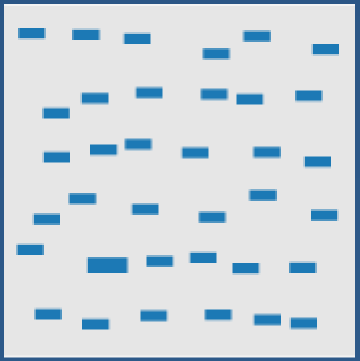
“Preattentive” Processing

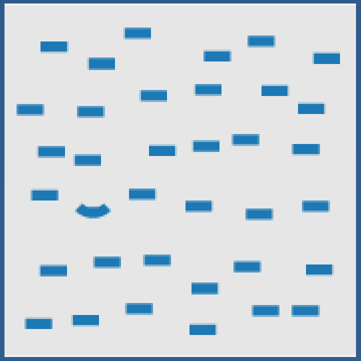
- Certain basic visual properties are detected immediately by low-level visual system
- “Pop-out” vs. serial search
- Tasks that can be performed in less than 200 to 250 milliseconds on a complex display
- Eye movements take at least 200 ms to initiate

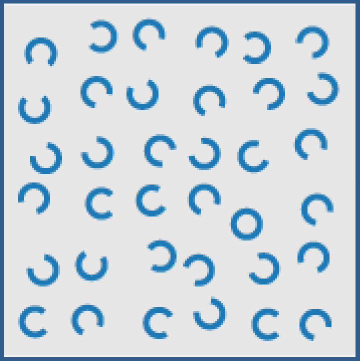
Preattentive Visual Search Tasks

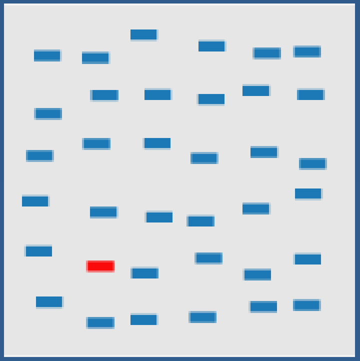

Orientation


Length


Size

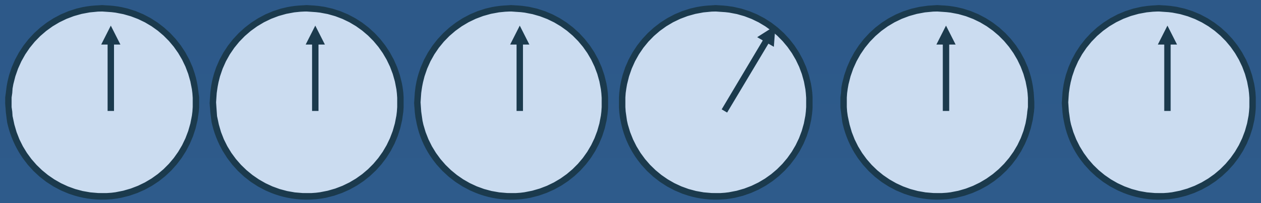

Curvature


Closure

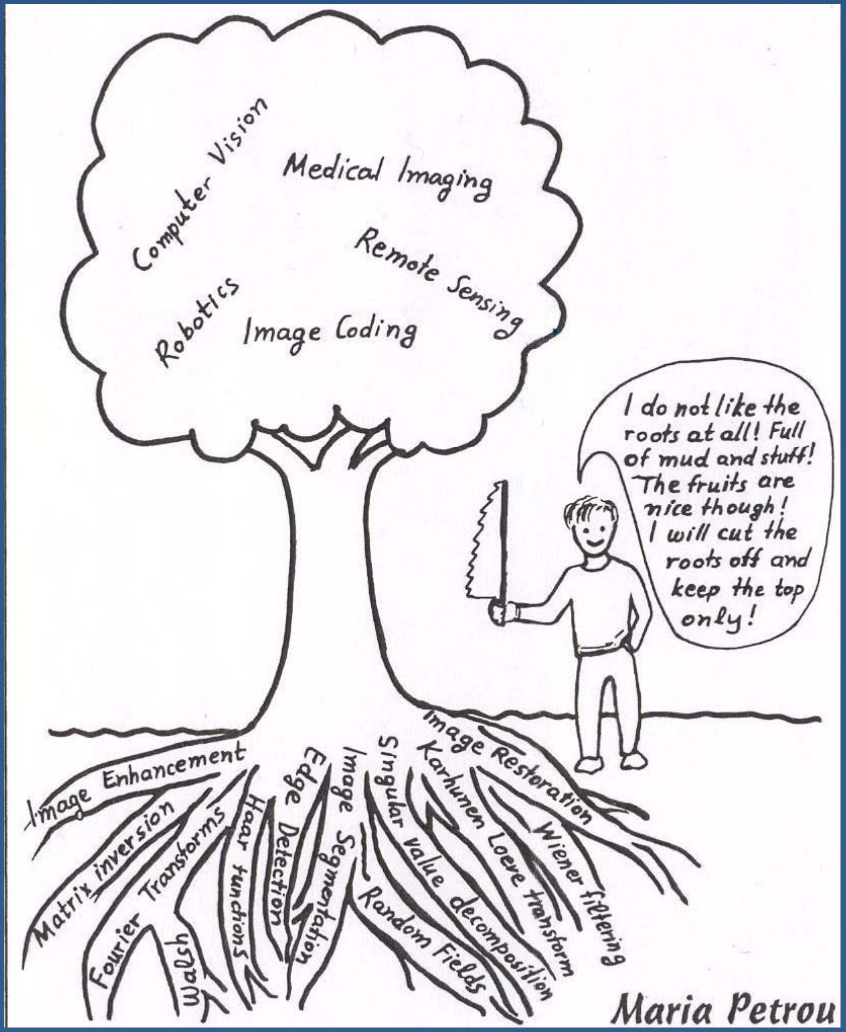

Color

Cockpit Dials

- Detection of a slanted line in a sea of vertical lines is preattentive



Perspective



(courtesy of Maria Petrou)

What is an Image?

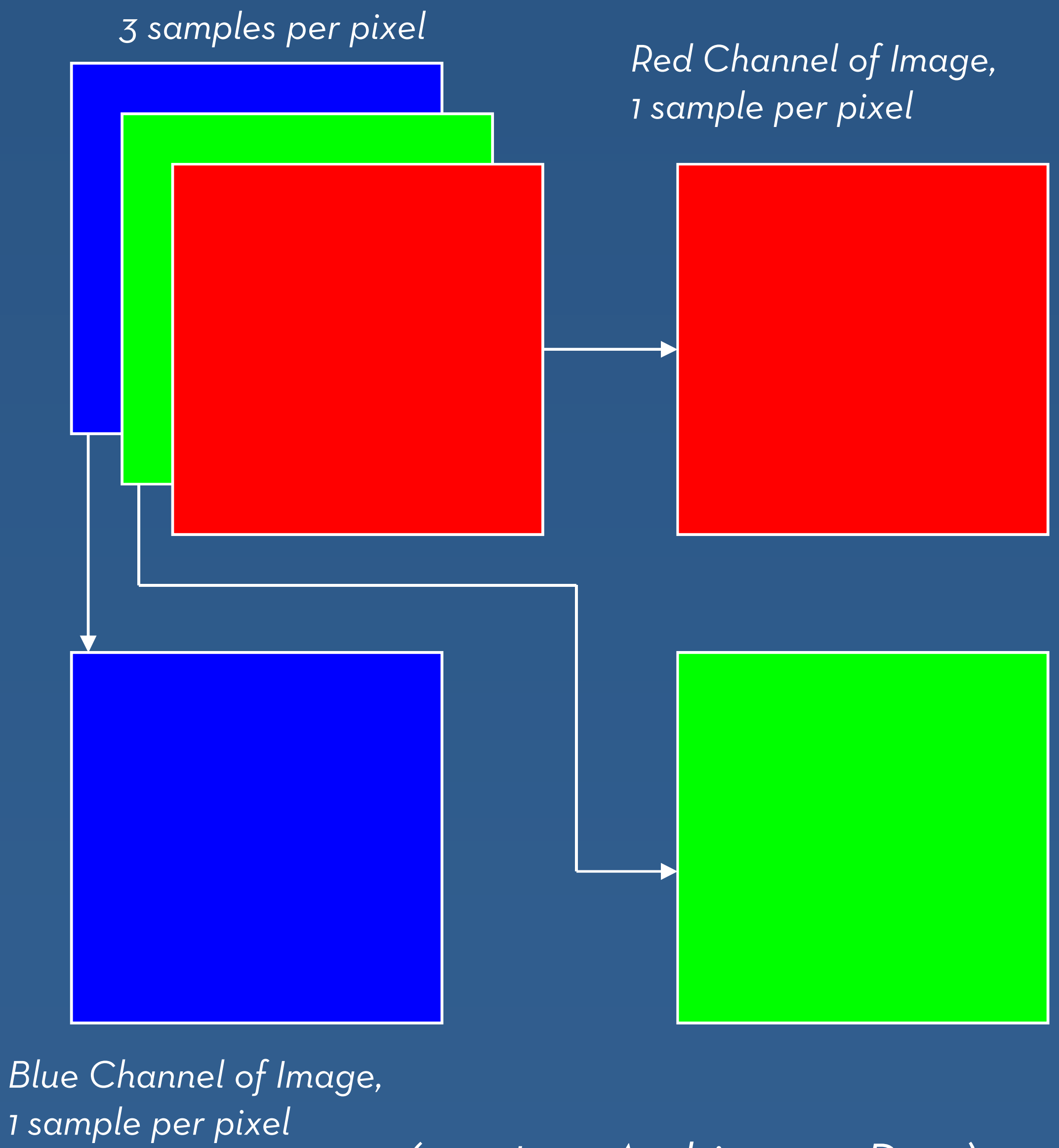
- Digital representation of a real-world scene
- Composed of discrete “picture elements” (pixels)
- Pixels parameterized by
  - Position
  - Intensity
  - Time

What is an Image?

- A 2D domain
  - With samples at regular points (almost always a rectilinear grid)
  - Whose values represent gray levels, colors, or opacities
- Common image types:
  - 1 sample per point (B&W or Grayscale)
  - 3 samples per point (Red, Green, and Blue)
  - 4 samples per point (Red, Green, Blue, and "Alpha", a.k.a. Opacity)

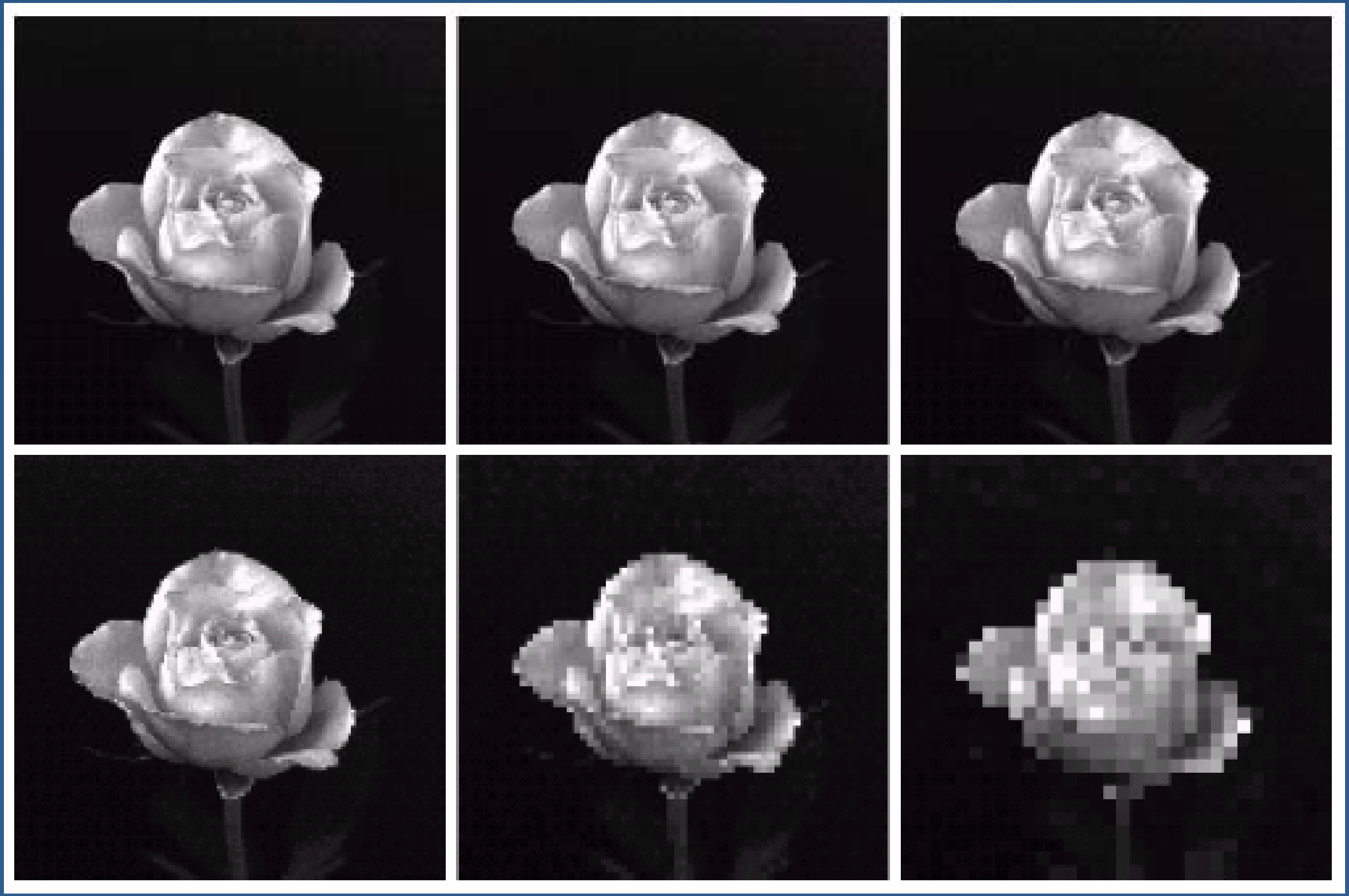
Channels

- In an images with multiple samples per pixel, we refer to each set of one type of sample as a "plane" or "channel."



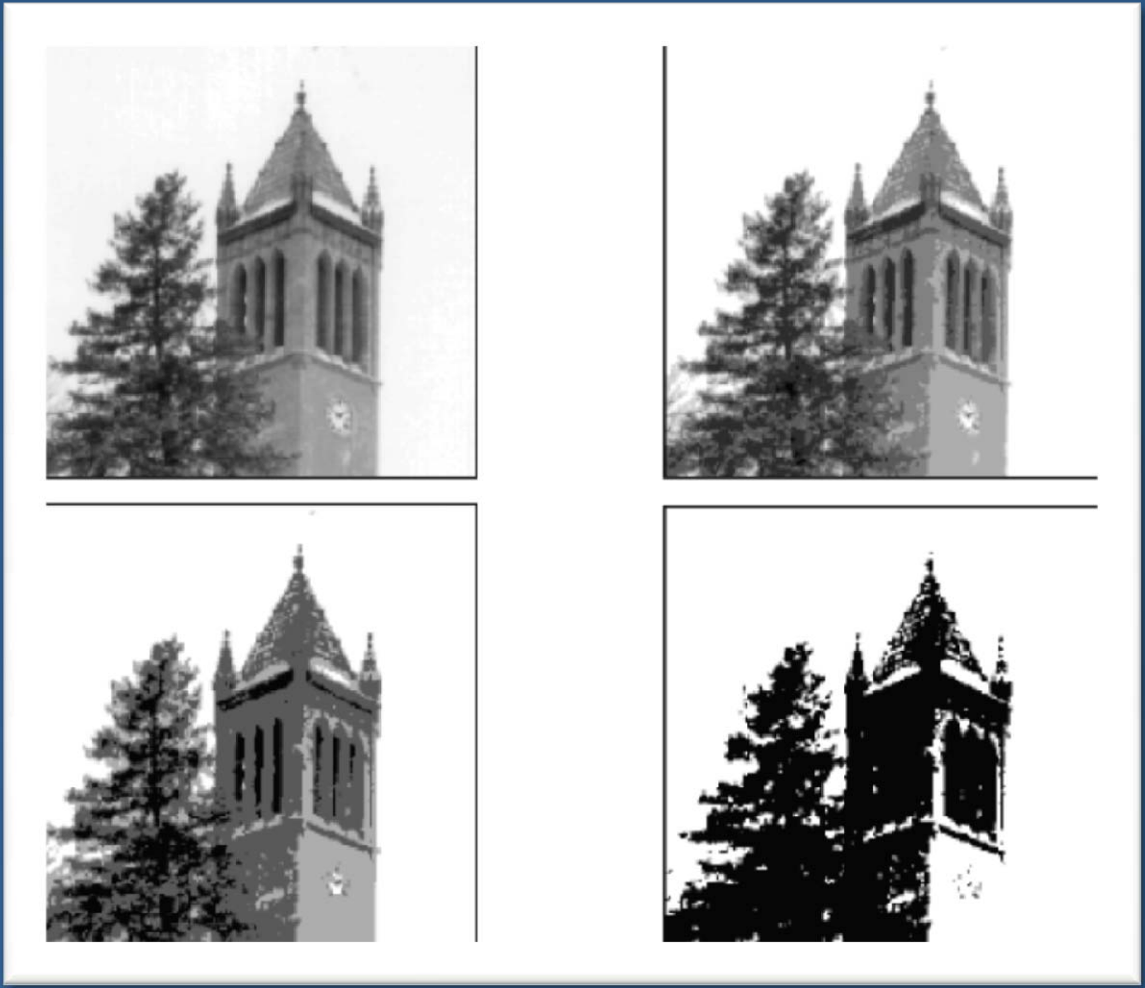
(courtesy Andries van Dam)

Spatial Sampling



(courtesy of Gonzalez and Woods)

Quantization



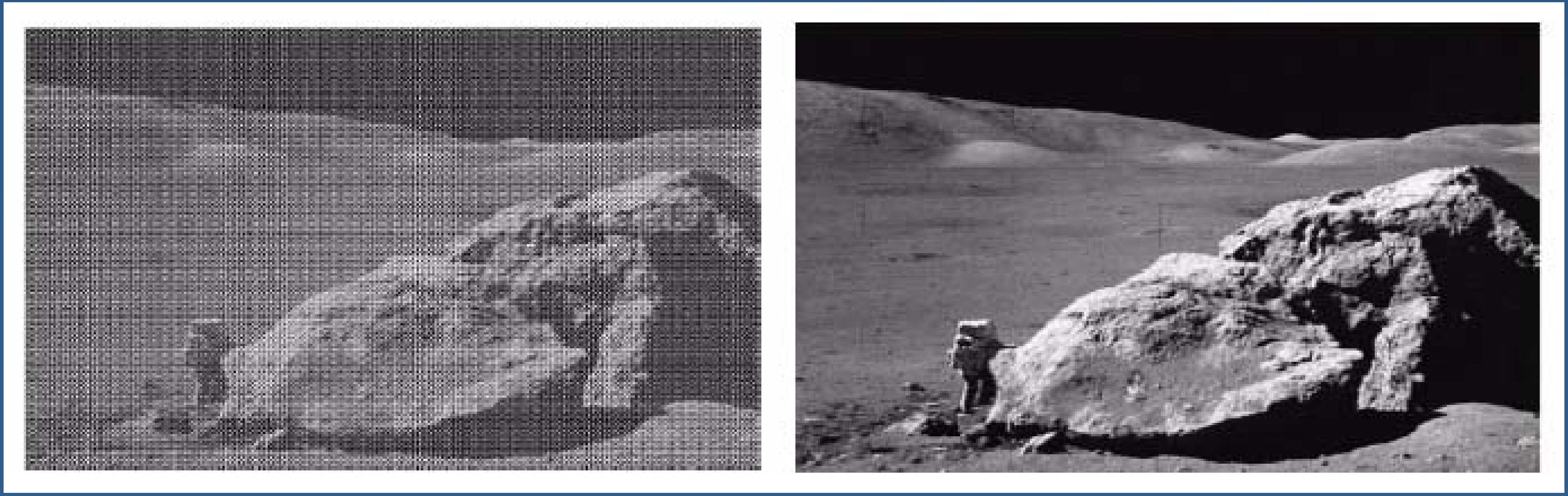
(courtesy of Gonzalez and Woods)

What is Image Processing?

- Generally, an attempt to do one of the following:
  - **Restore an image** (take a corrupted image and recreate a clean original)
  - **Enhance an image** (alter an image to make its meaning clearer to human observers)
  - **Understand an image** (mimic the human visual system in extracting meaning from an image)

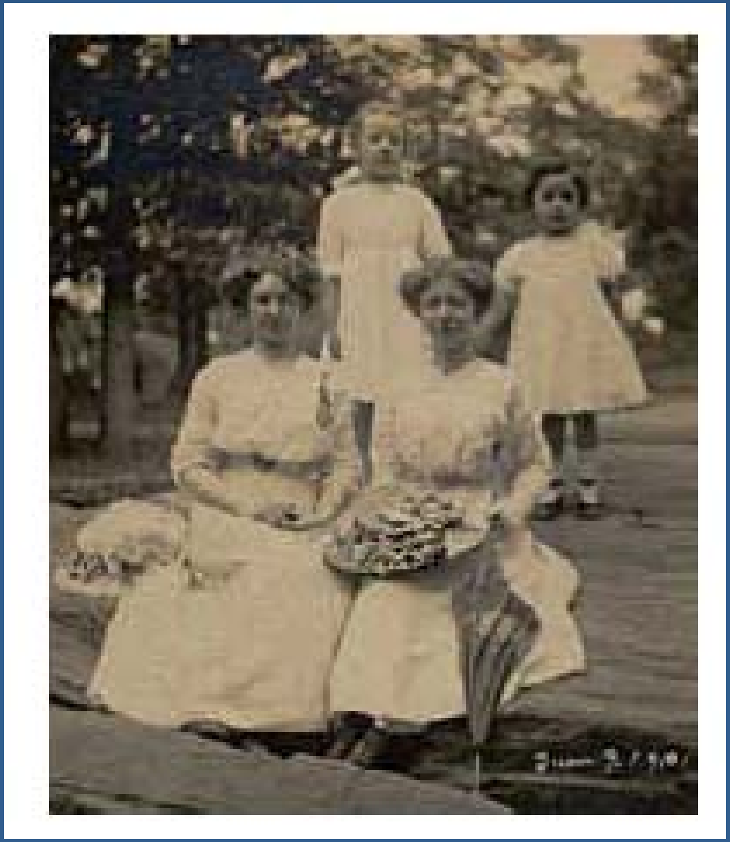
Image Restoration

- Removing sensor noise
- Restoring old, archived film and images that has been damaged



(courtesy of Gonzalez and Woods)

Image Restoration



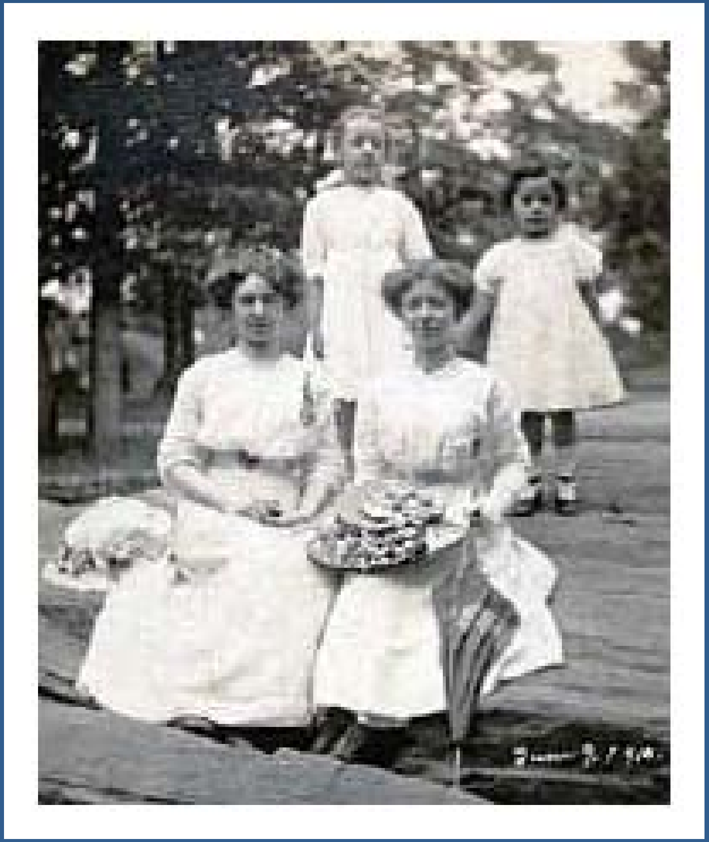
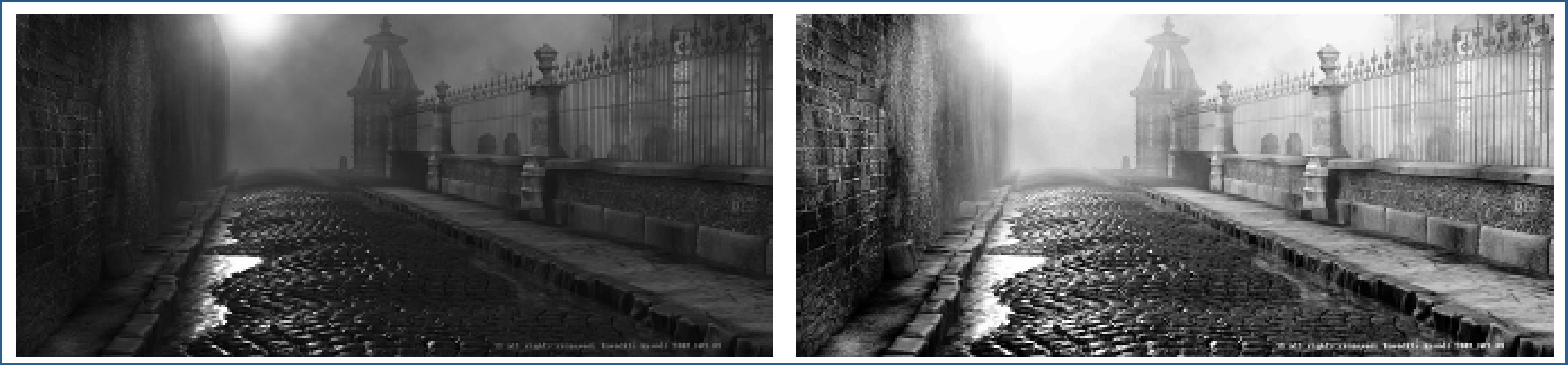


Image Enhancement

- Often used to increase the contrast in images that are overly dark or light
- Enhancement algorithms often play to humans' sensitivity to contrast

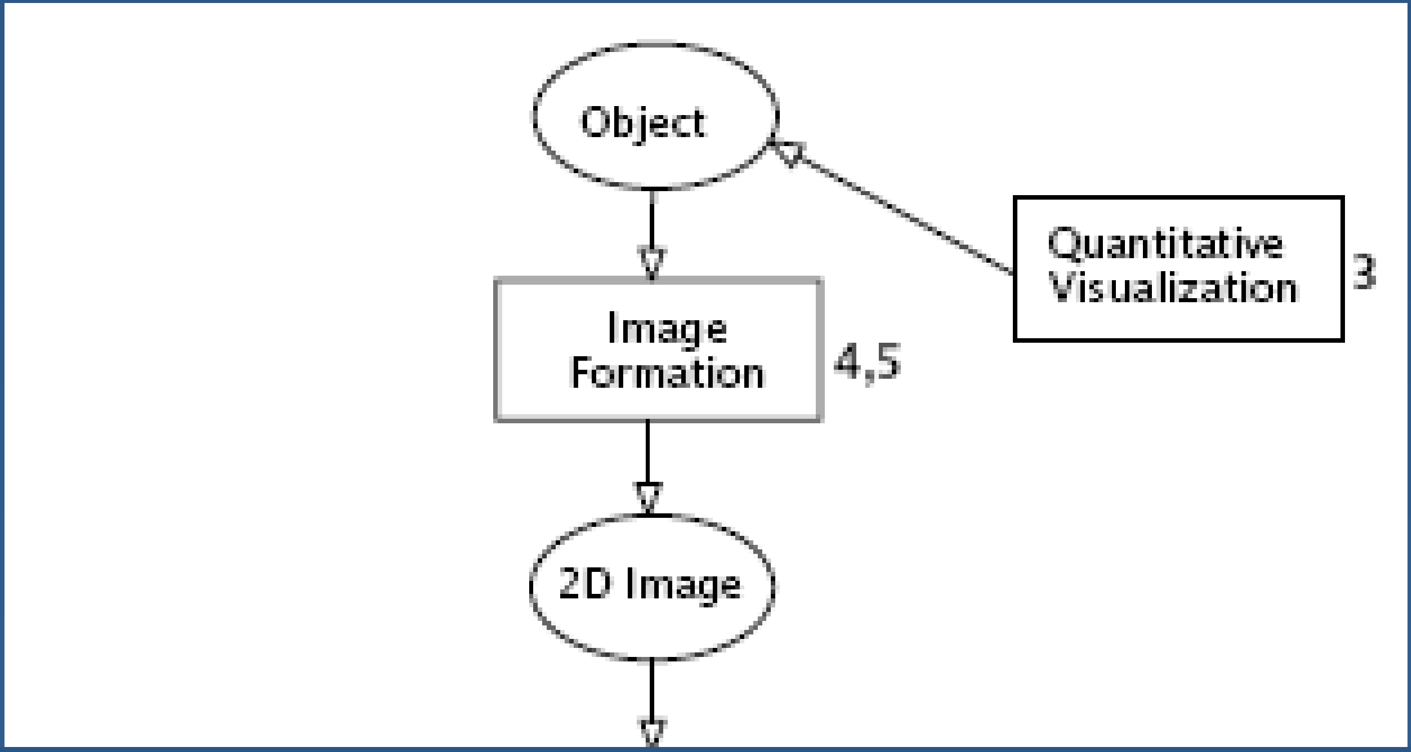


(courtesy of Tobey Thorn)

Image Understanding

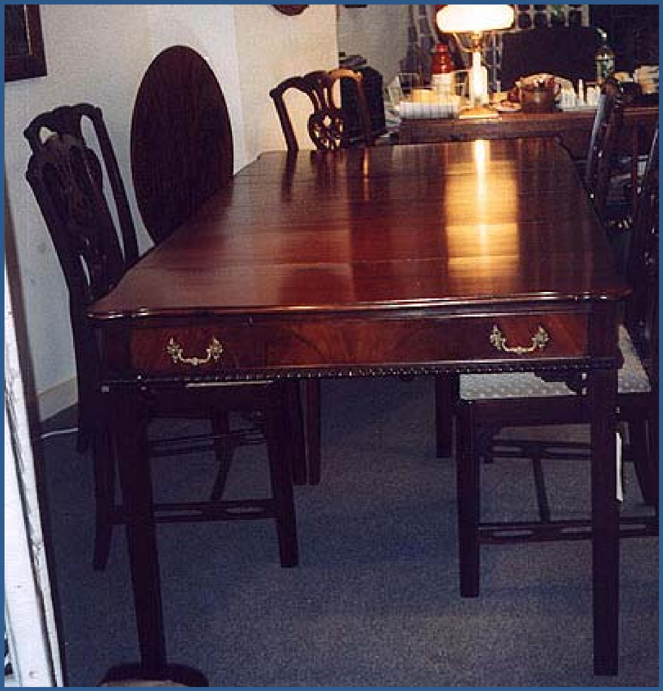
- Image understanding includes many different tasks
  - Segmentation (identifying objects in an image)
  - Classification (assigning labels to individual objects or pixels)
  - Interpretation (extracting meaning from the image as a whole)

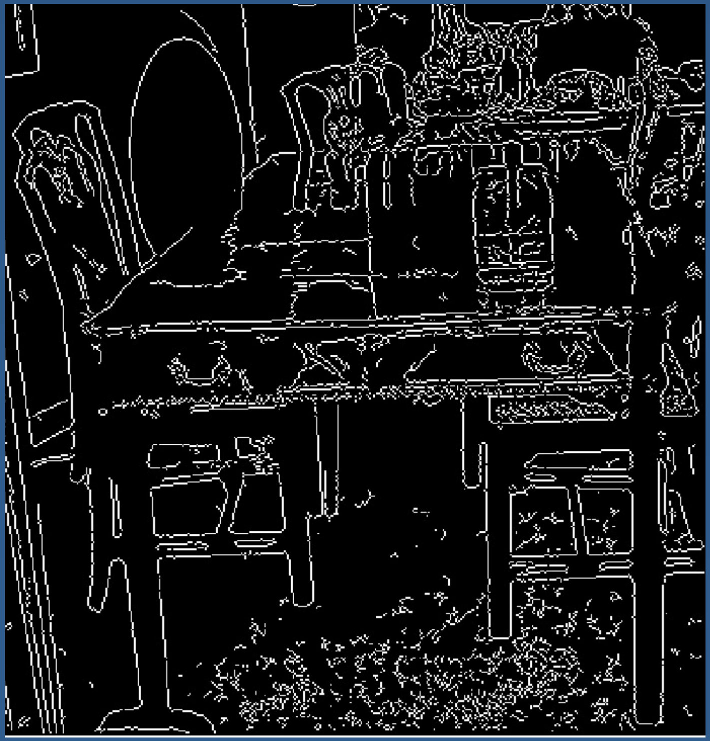
Image Processing: Hierarchy of Tasks



(courtesy of B. Jähne)

Separating Regions with Filters



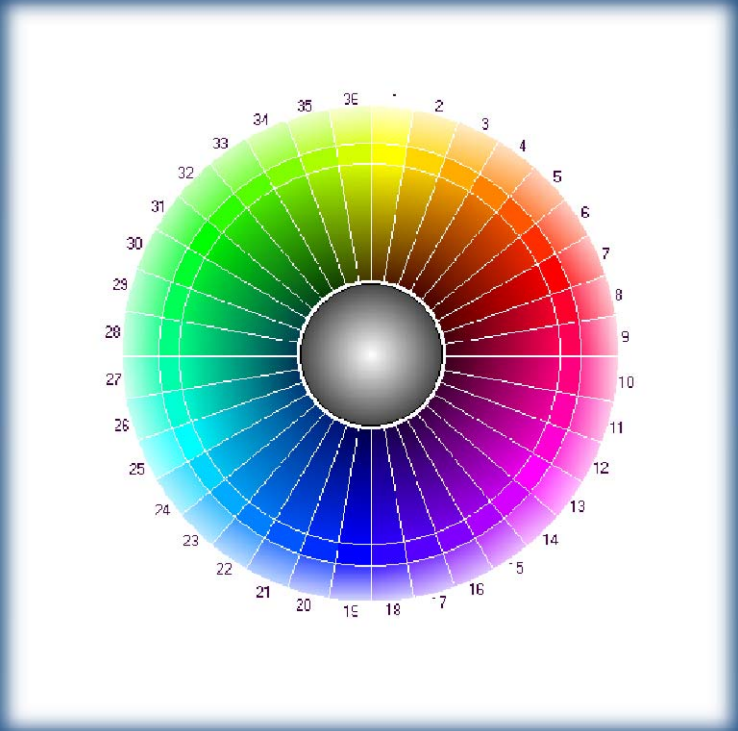


What is Color?

- A box of pencils?

What is Color?

- A quantity related to the wavelength of light in the visible spectrum?



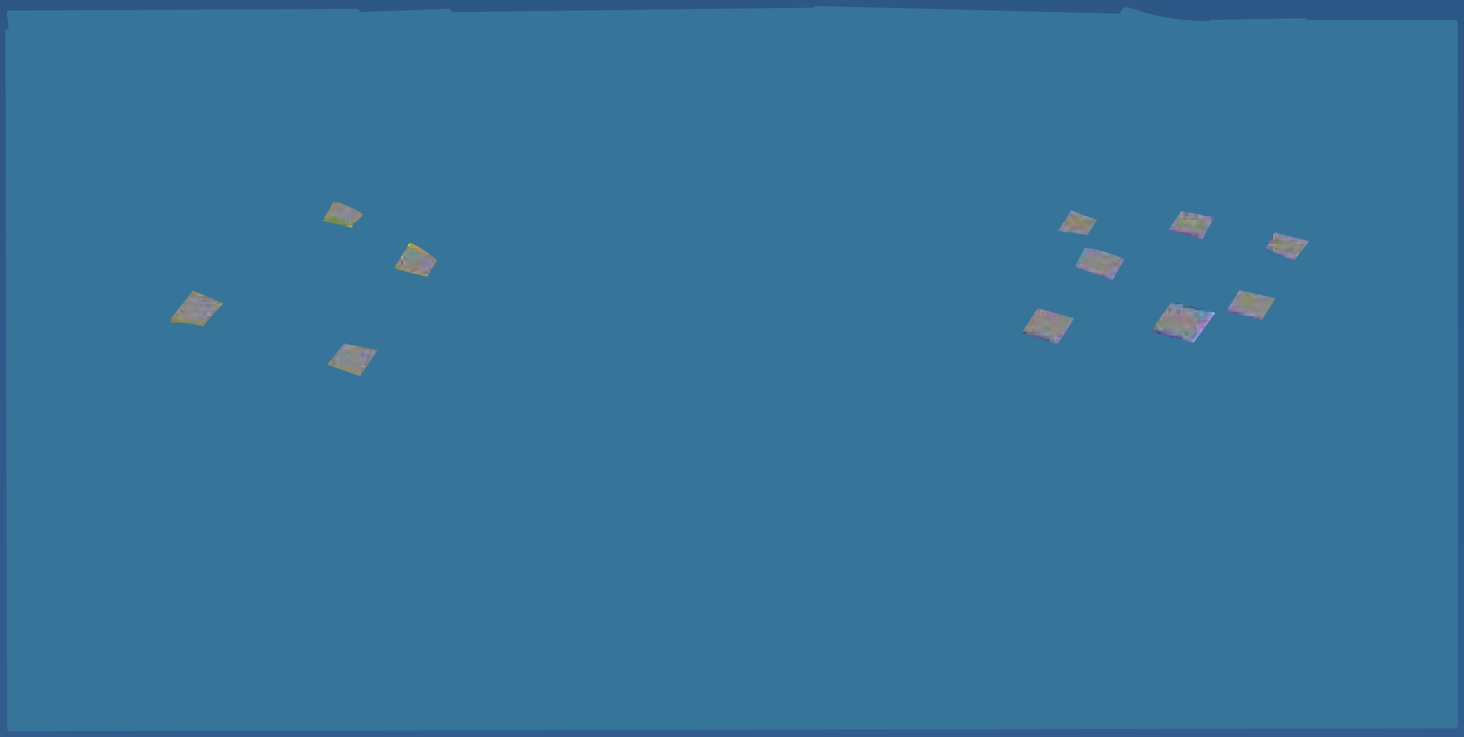
(courtesy of J.M. Rehg)

What is Color?

- A perceptual attribute of objects and scenes constructed by the visual system?

(courtesy R. Beau Lotto)

Color Perception



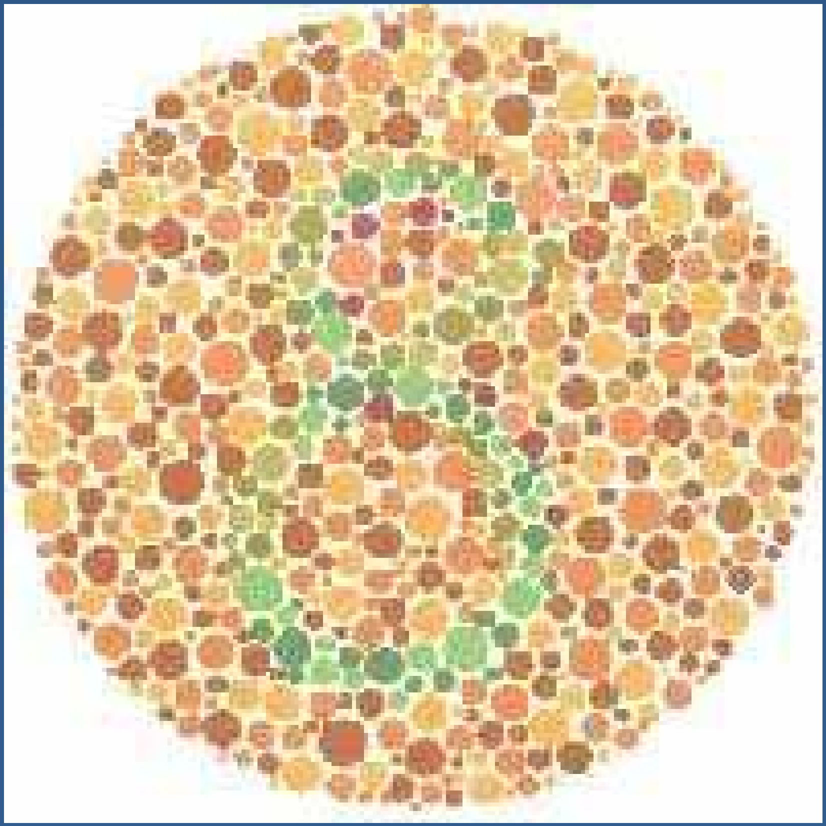
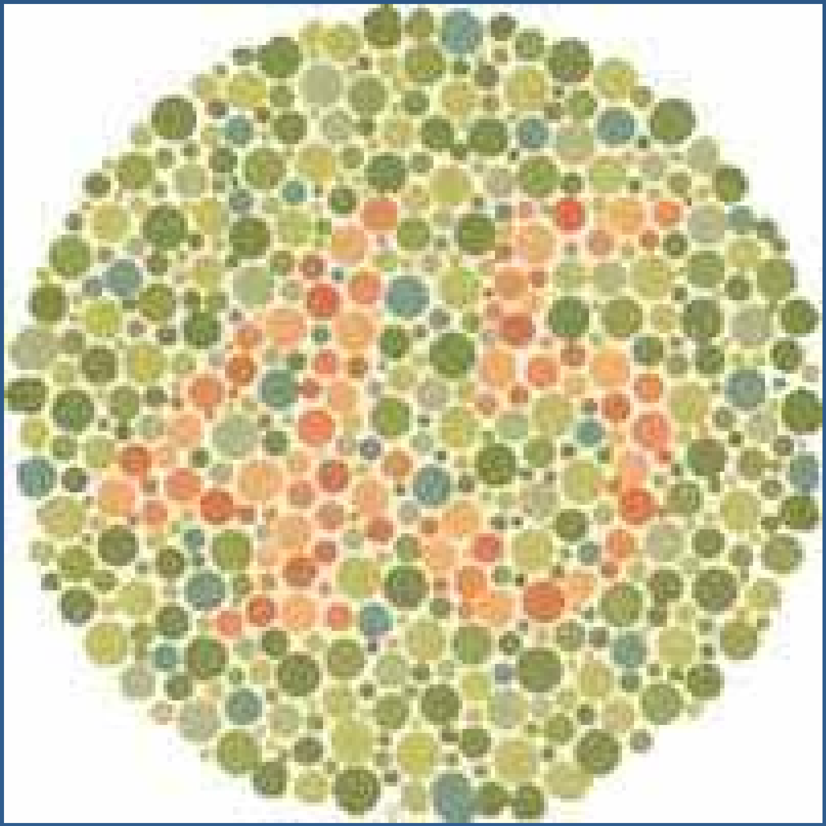
(courtesy R. Beau Lotto)

How Do We Use Color?

- In Biological Vision
  - Distinguish food from nonfood
  - Help identify predators and prey
  - Check health of others
- In Computer Vision
  - Find a person’s skin
  - Segment (group together) pixel regions belonging to the same object

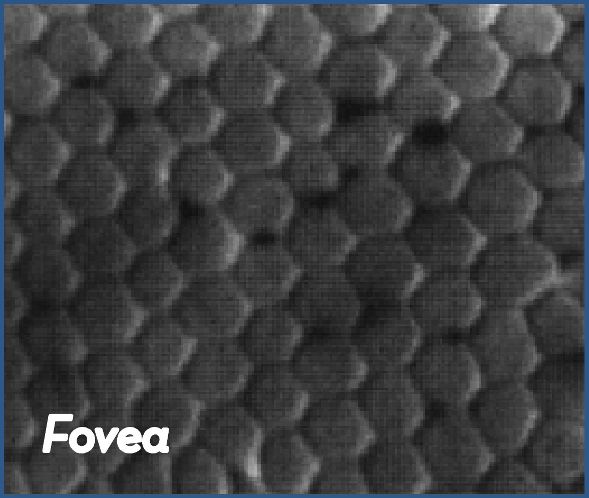
(courtesy of J.M. Rehg)

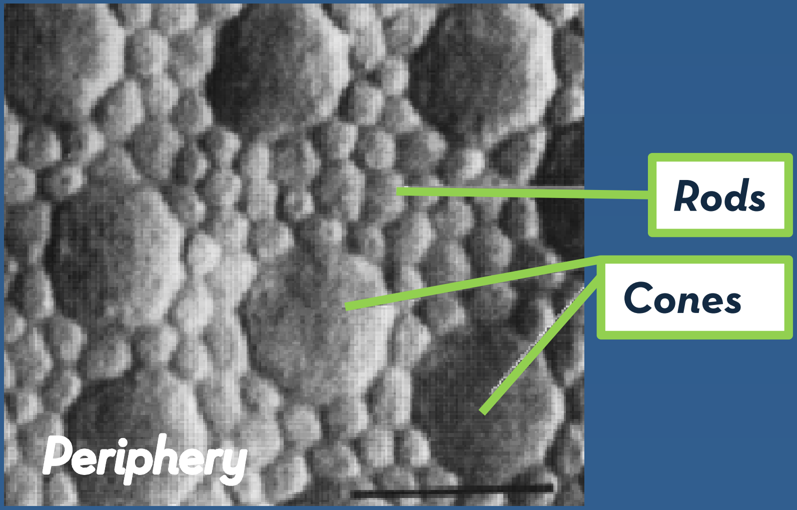
Colorblindness

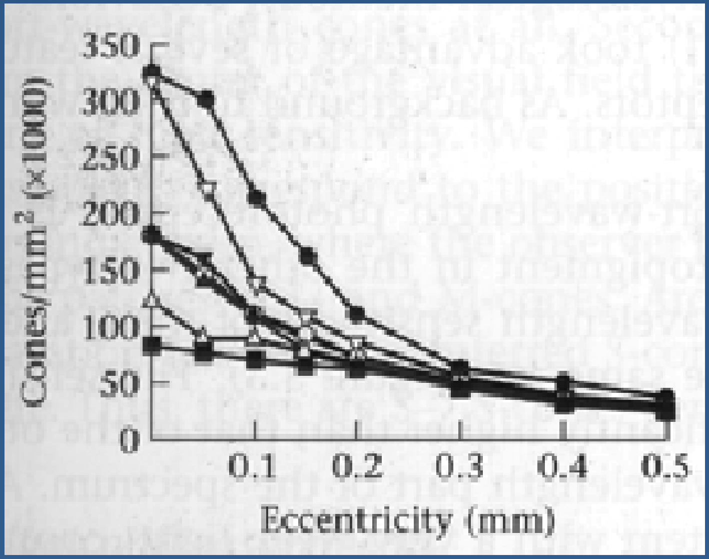
Ishihara Test for Color Blindness

Human Photoreceptors

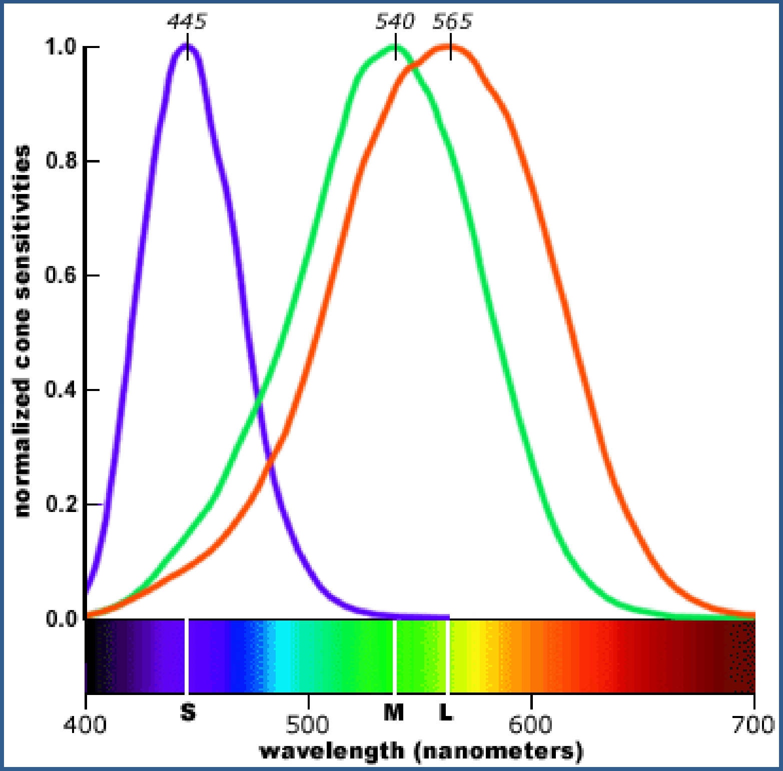


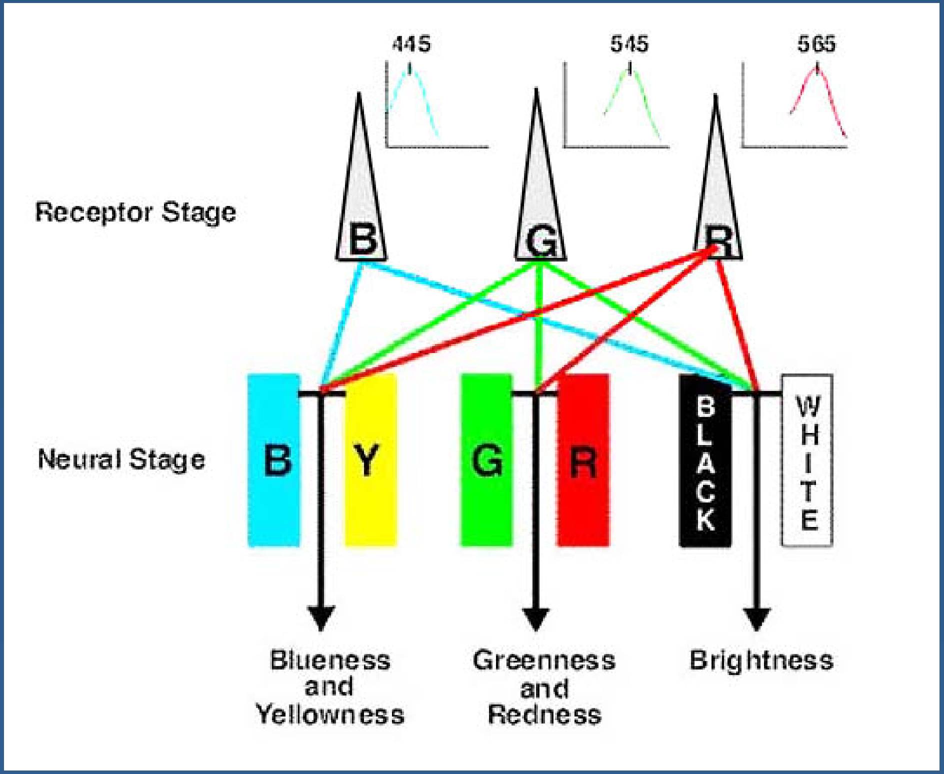


Cone Density

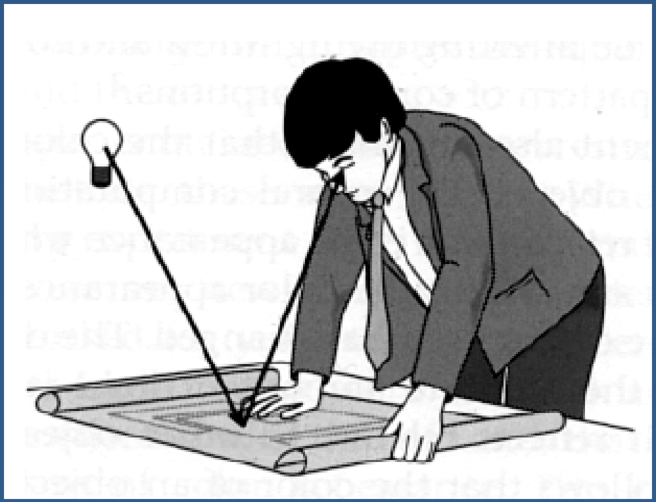


Human Cone Sensitivities

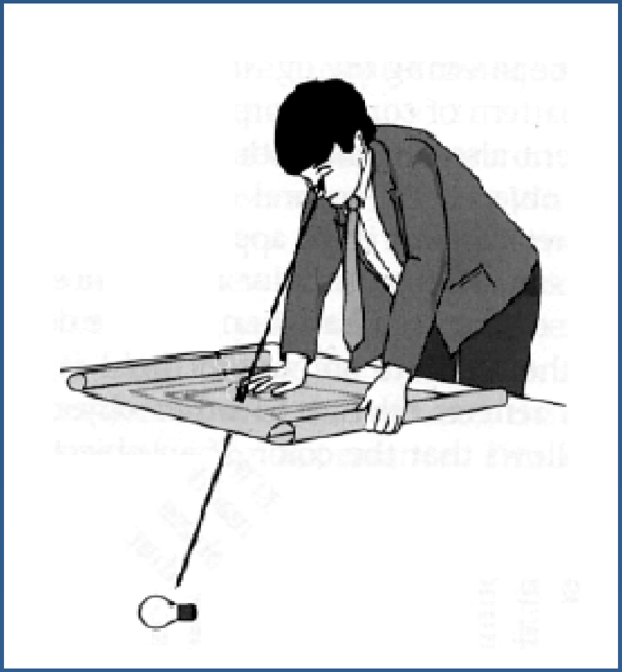




Reflectance and Transmittance



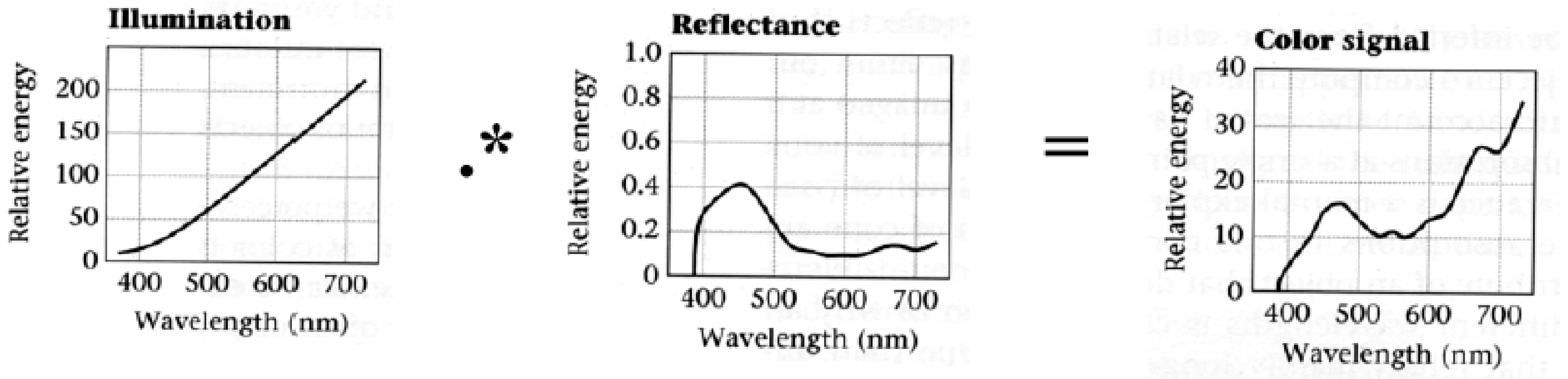
Reflectance



Transmittance

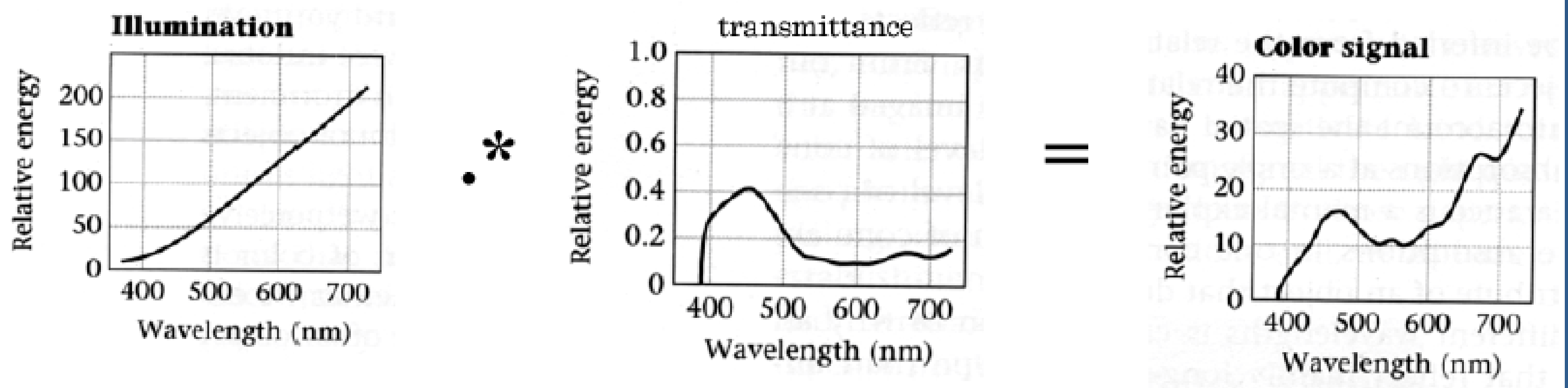
(courtesy of Brian Wandell)

Reflectance



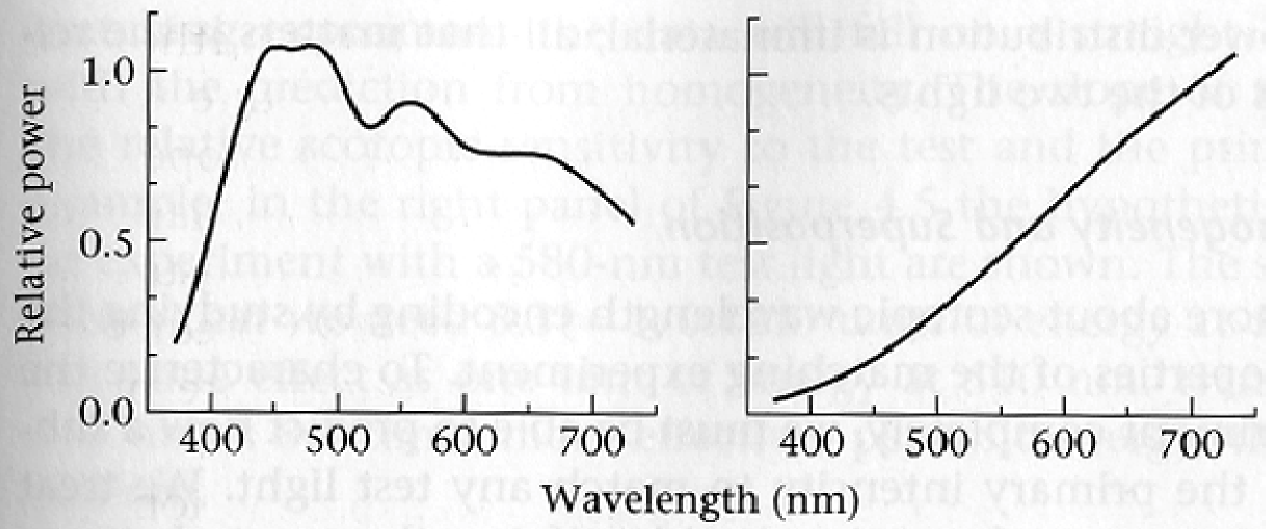
Foundations of Vision, by Brian Wandell, Sinauer Assoc., 1995

Transmittance



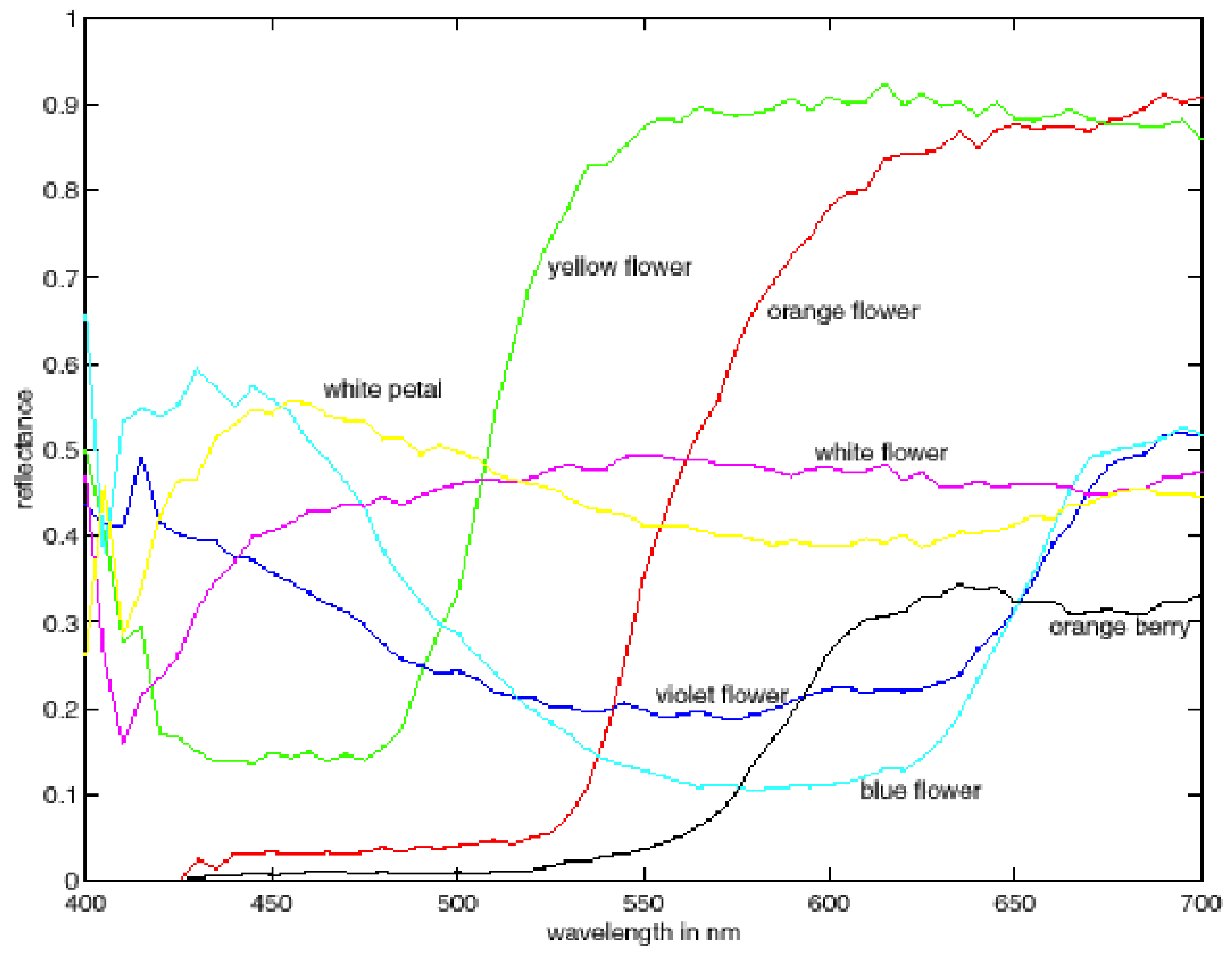
Foundations of Vision, by Brian Wandell, Sinauer Assoc., 1995

Illumination Spectra



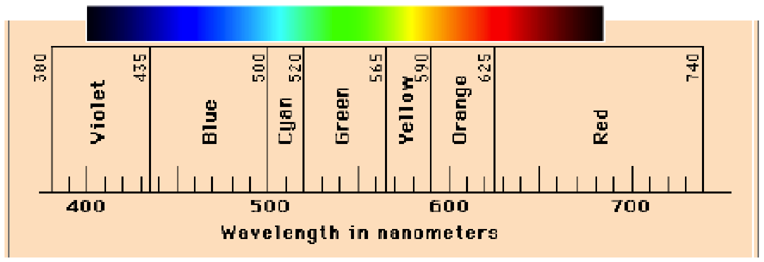
4.4 THE SPECTRAL POWER DISTRIBUTION of two important light sources are shown: (left) blue skylight and (right) a tungsten bulb.

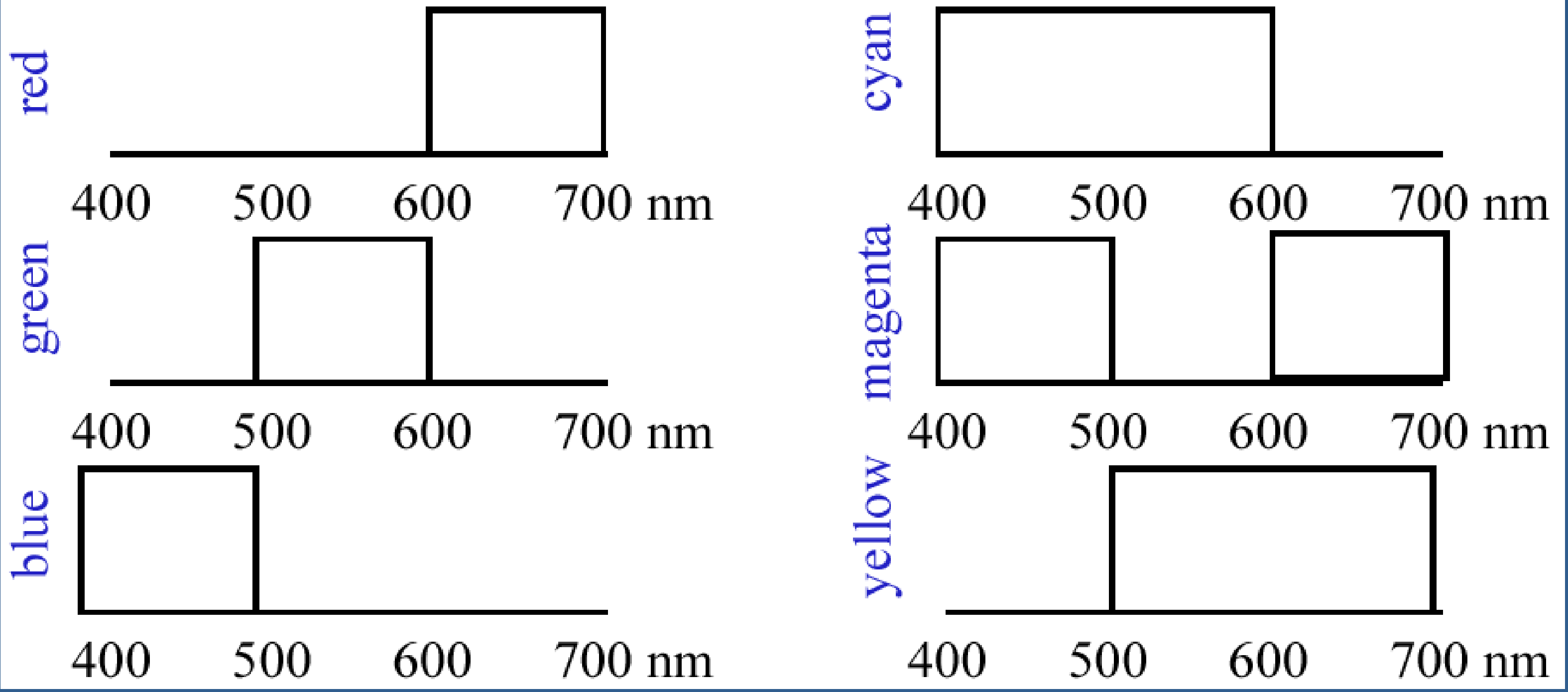
Reflectance Spectra



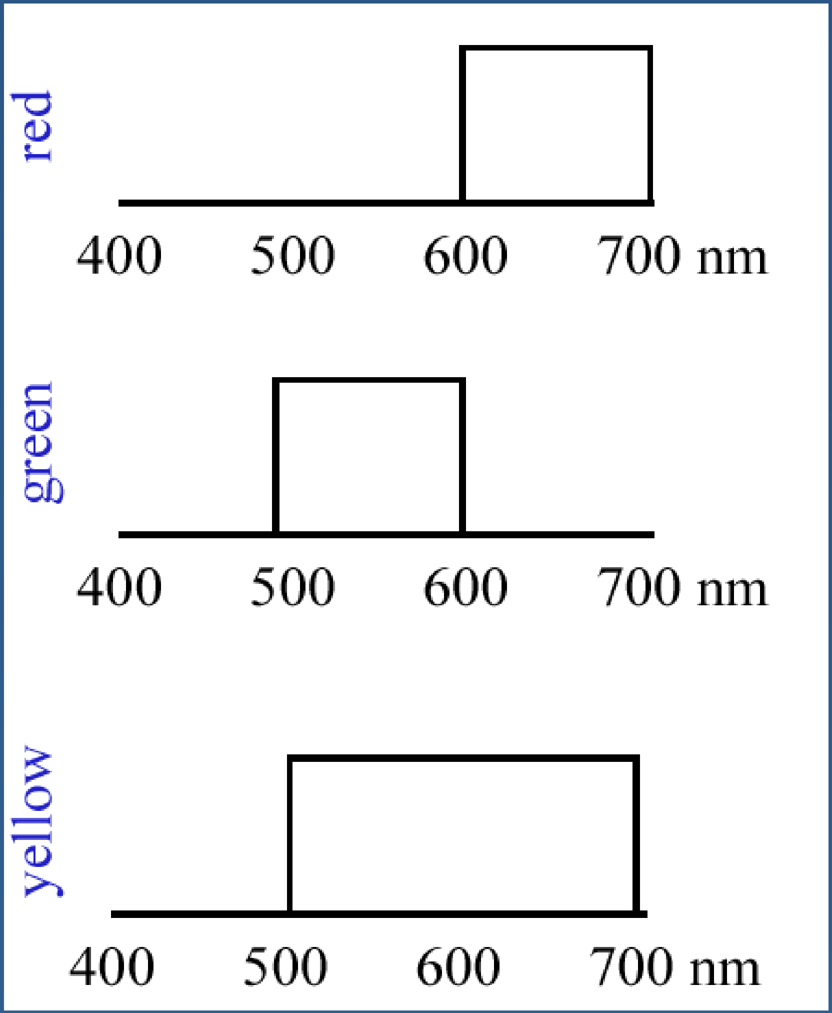
Spectral albedoes for several different leaves, with color names attached. Notice that different colours typically have different spectral albedo, but that different spectral albedoes may result in the same perceived color (compare the two whites). Spectral albedoes are typically quite smooth functions. Measurements by E.Koivisto.

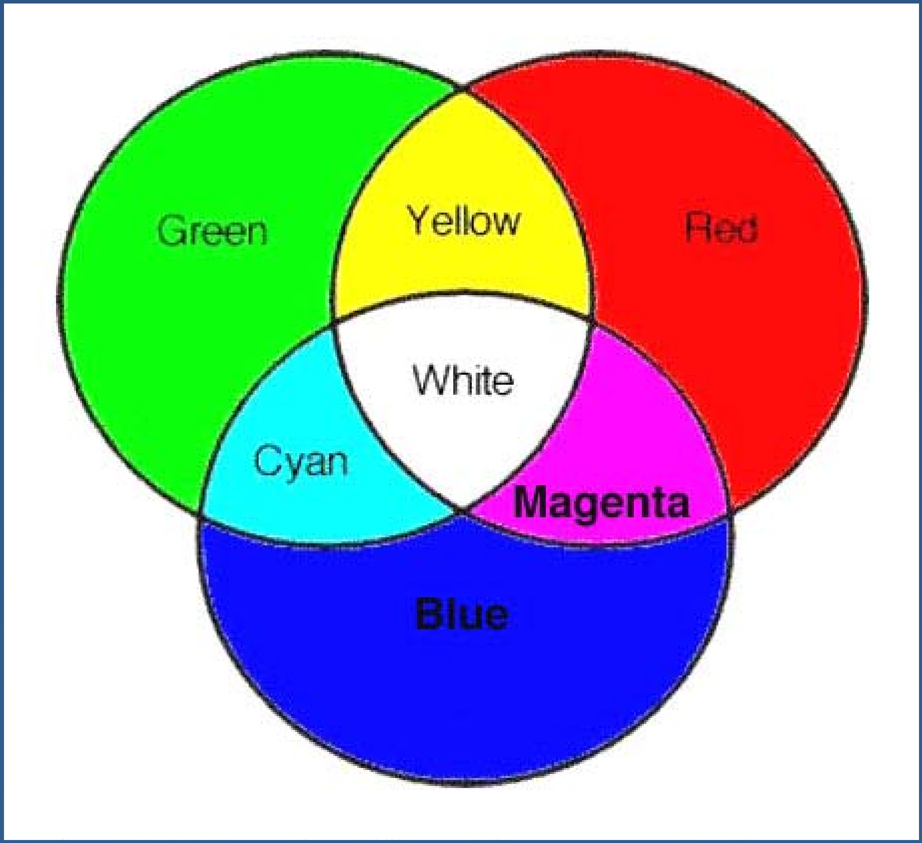
Assigning Names to Spectra



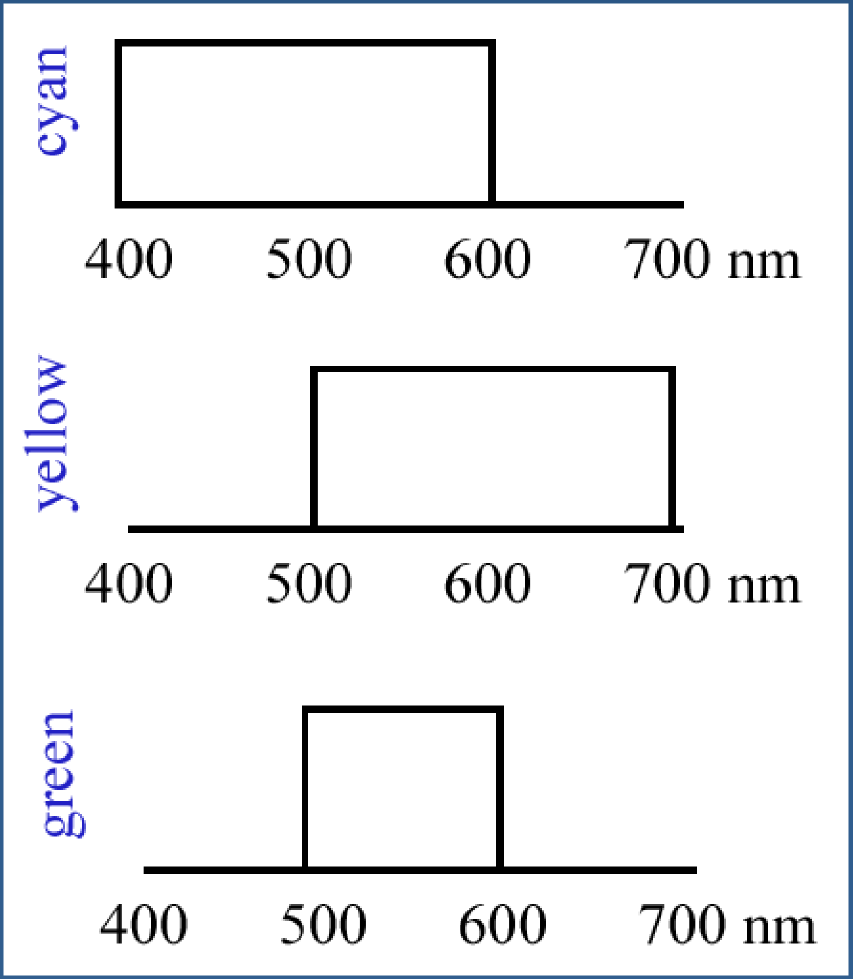


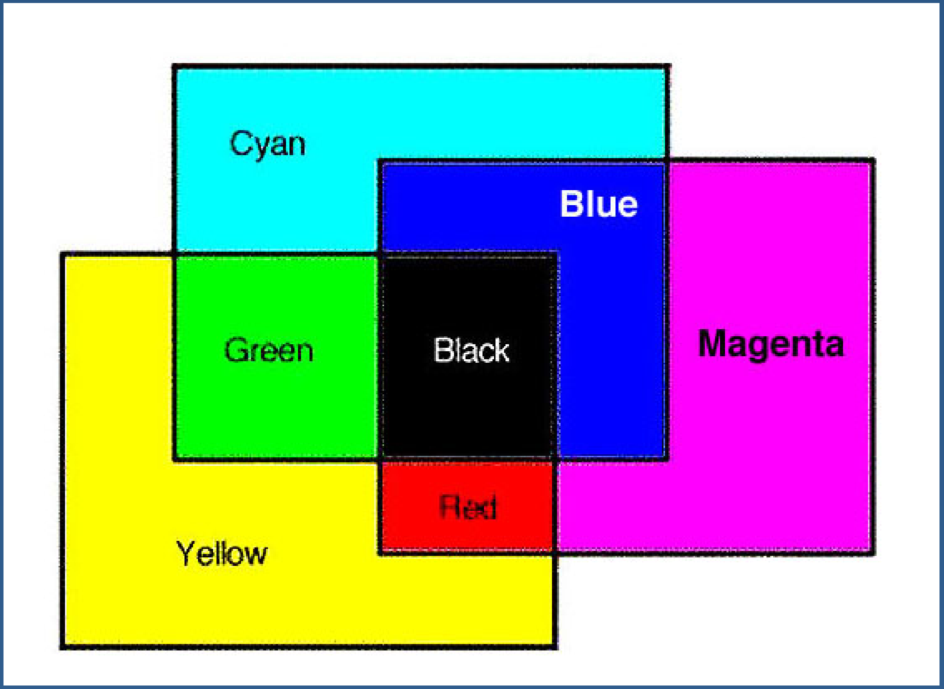
Additive Color Mixing



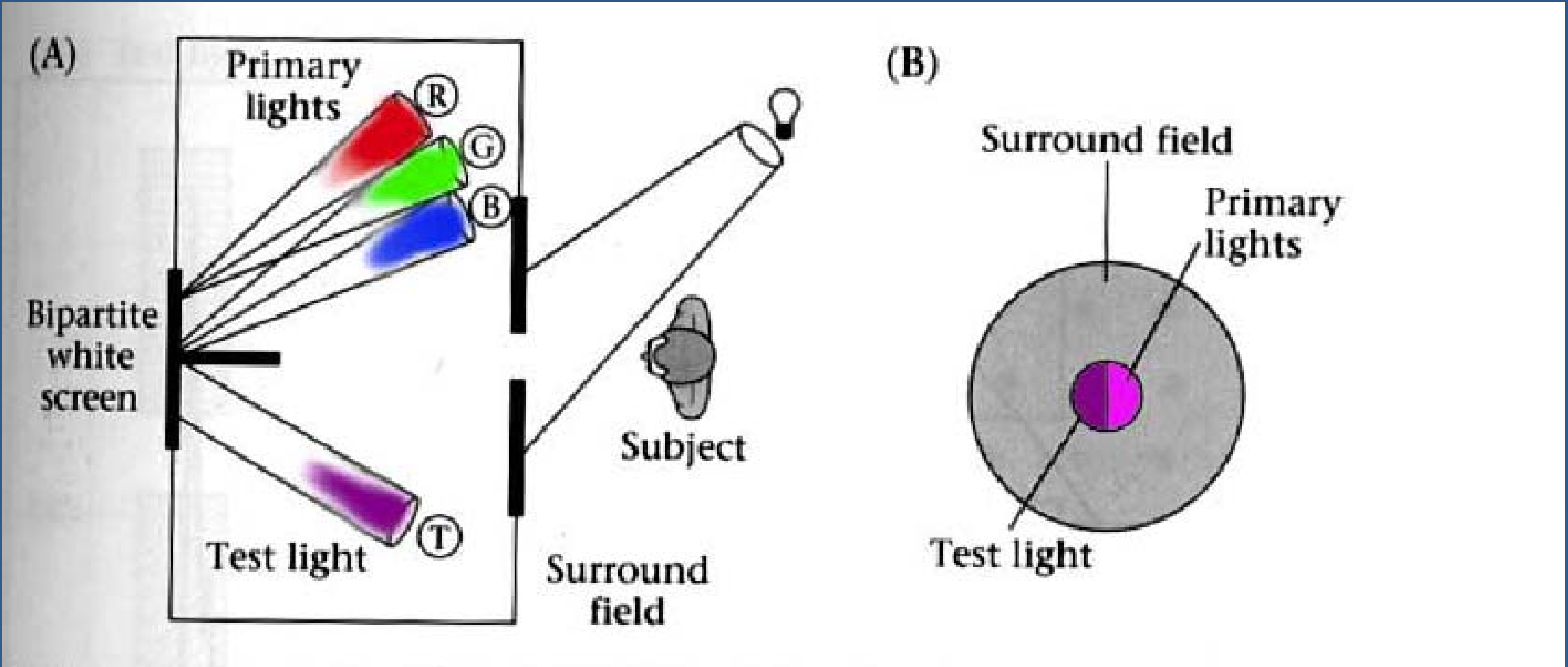


Subtractive Color Mixing



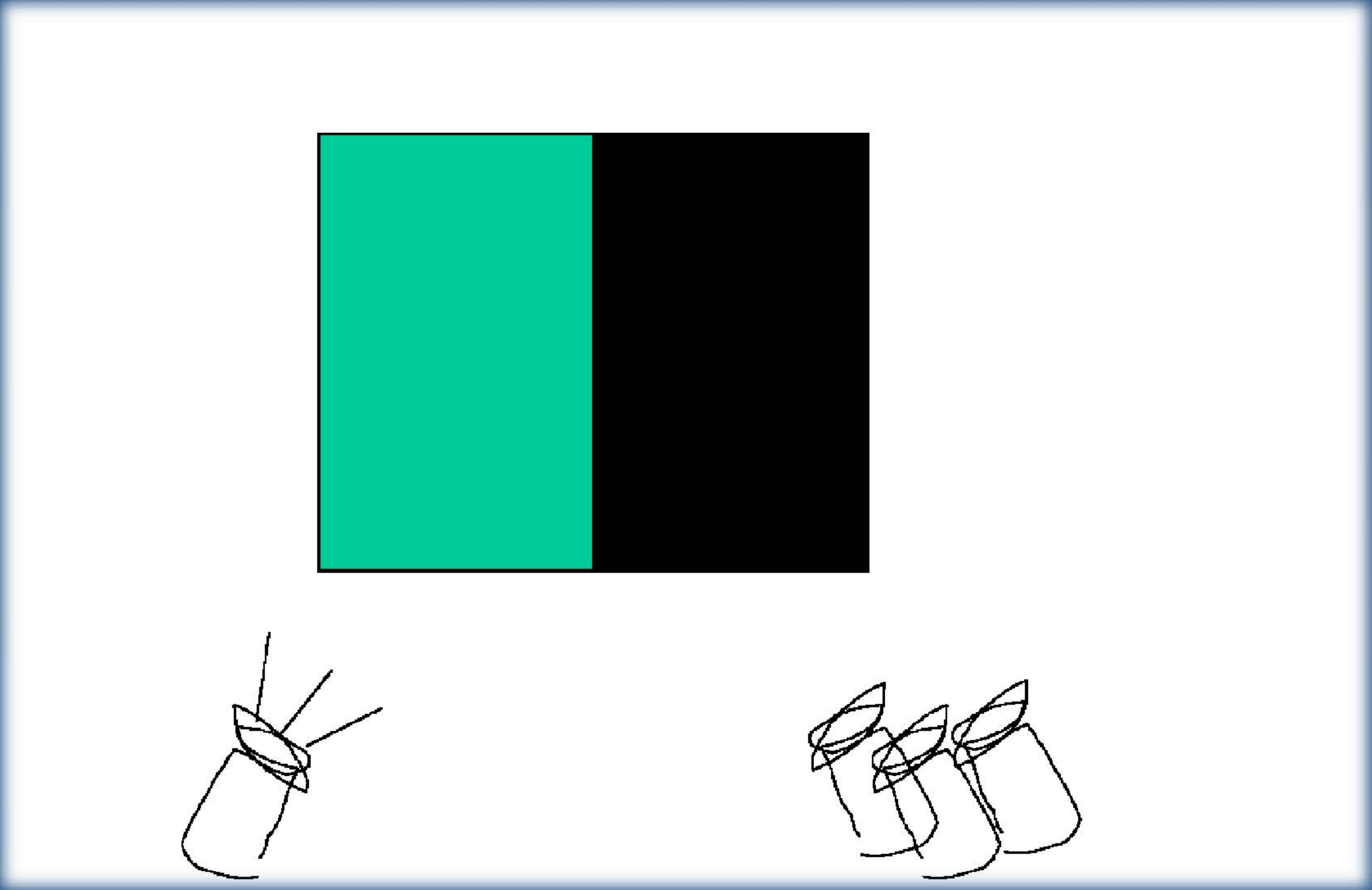


The Color Matching Experiment



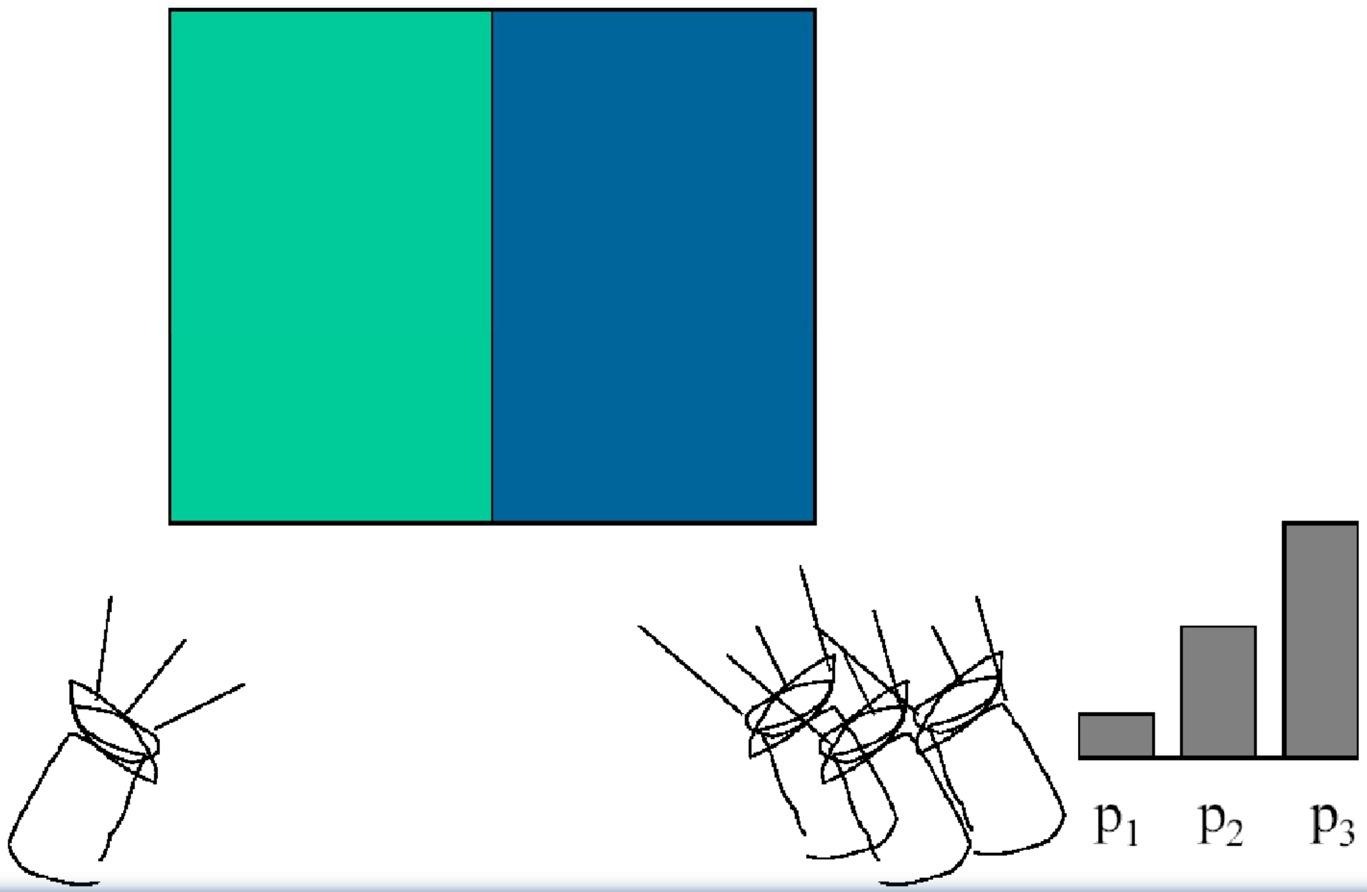
4.10 THE COLOR-MATCHING EXPERIMENT. The observer views a bipartite field and adjusts the intensities of the three primary lights to match the appearance of the test light. (A) A top view of the experimental apparatus. (B) The appearance of the stimuli to the observer. After Judd and Wyszecki, 1975.

Experiment Step 1



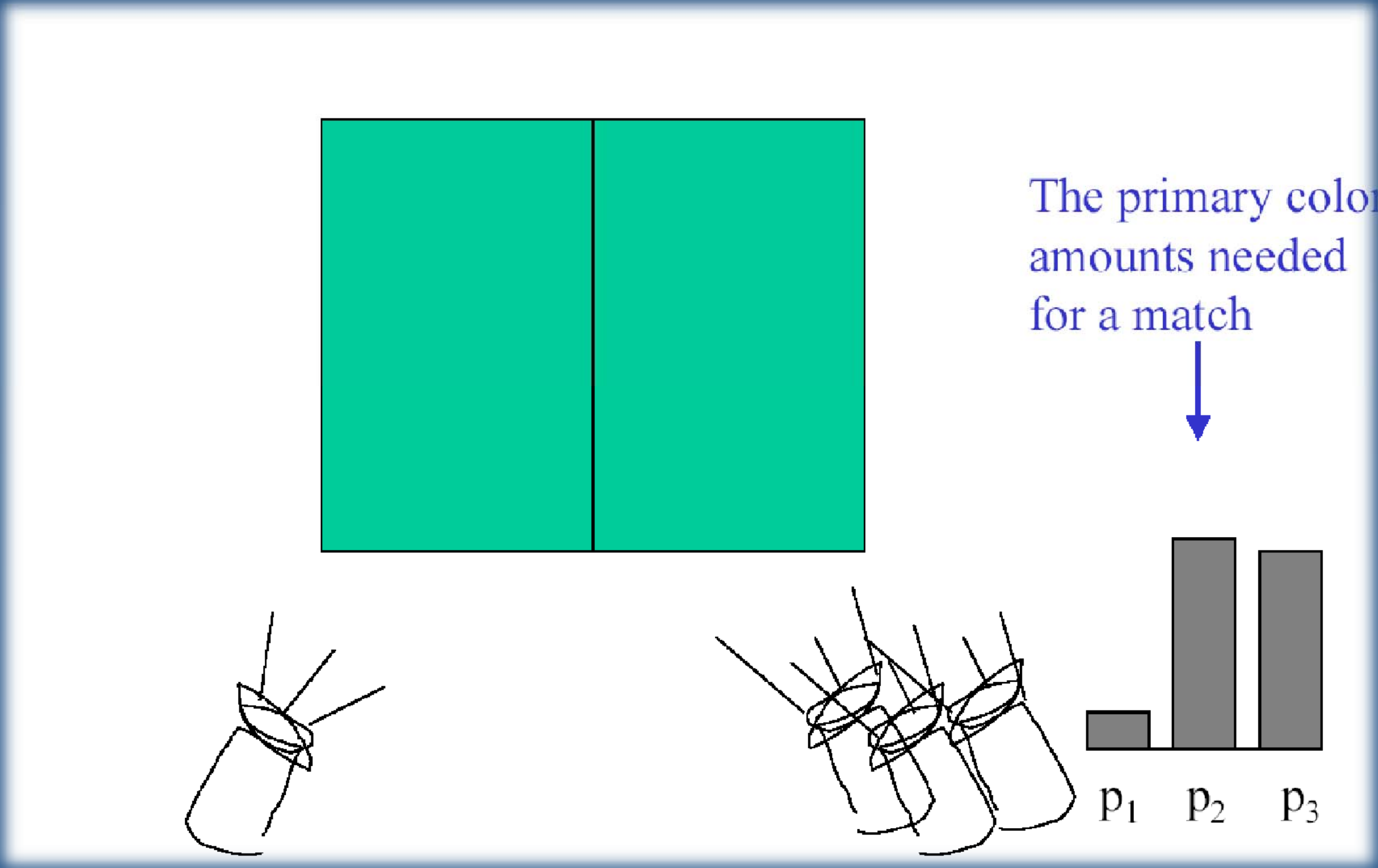
(courtesy of Bill Freeman)

Experiment Step 1



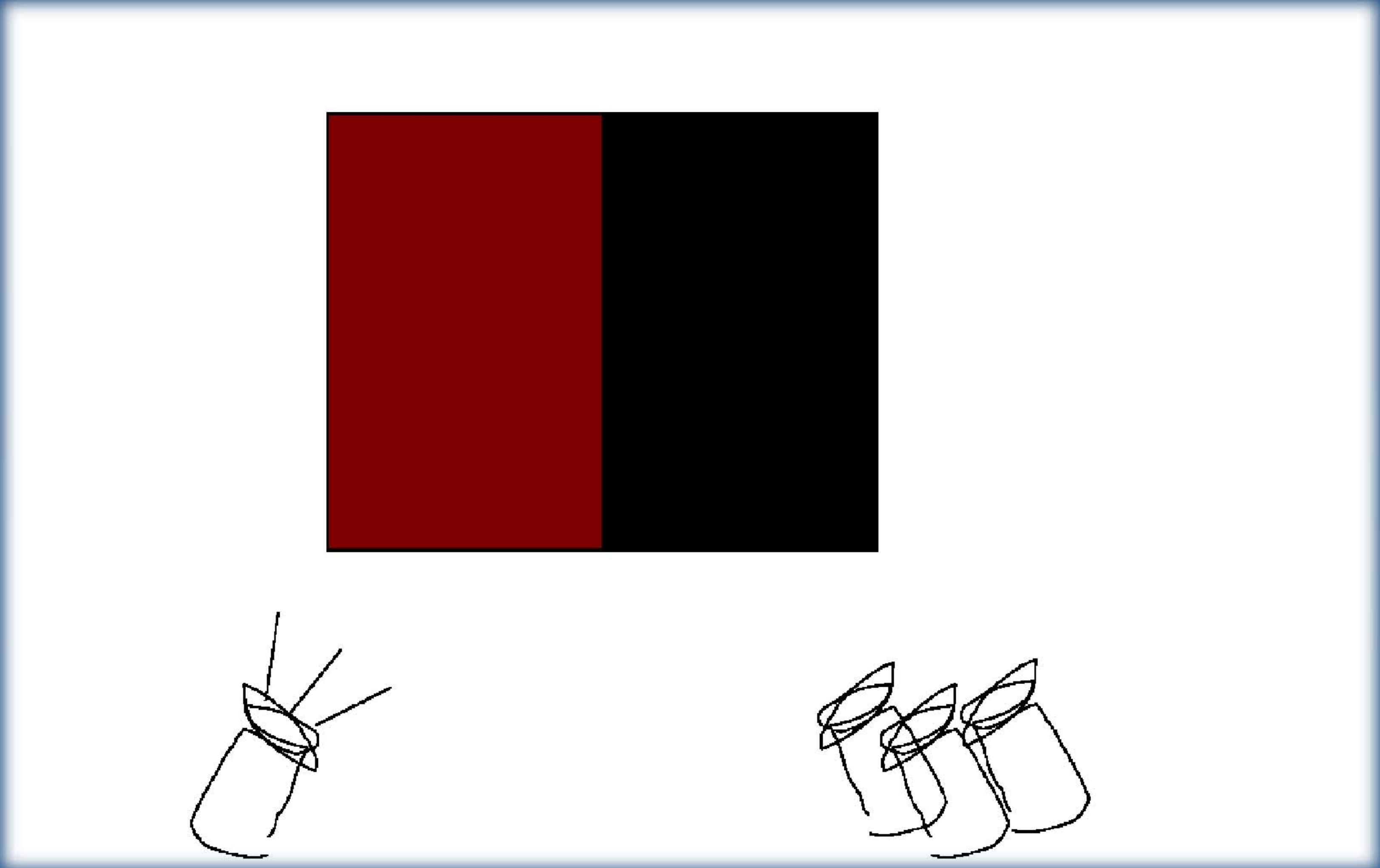
(courtesy of Bill Freeman)

Experiment Step 1



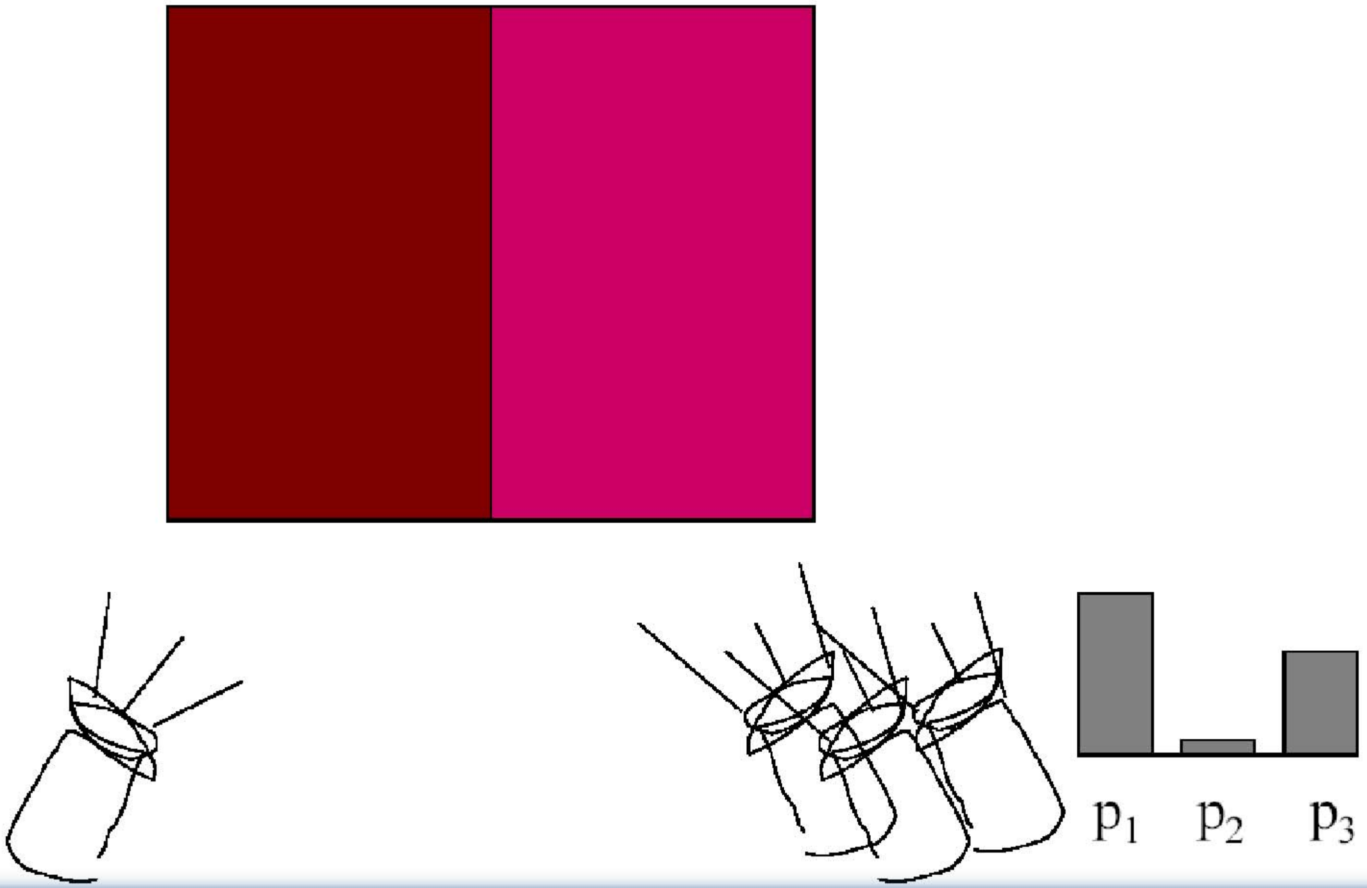
(courtesy of Bill Freeman)

Experiment Step 2



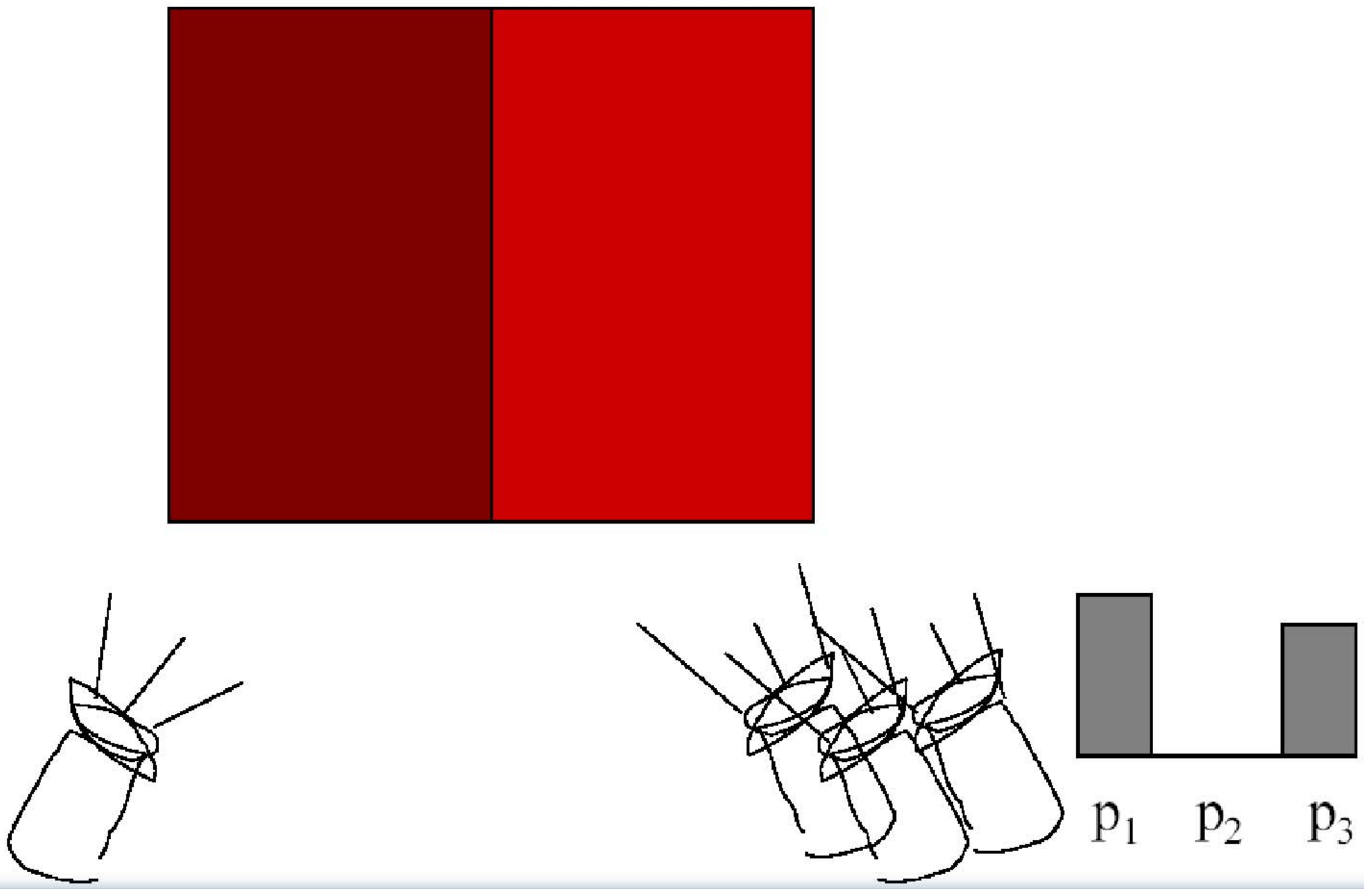
(courtesy of Bill Freeman)

Experiment Step 2



(courtesy of Bill Freeman)

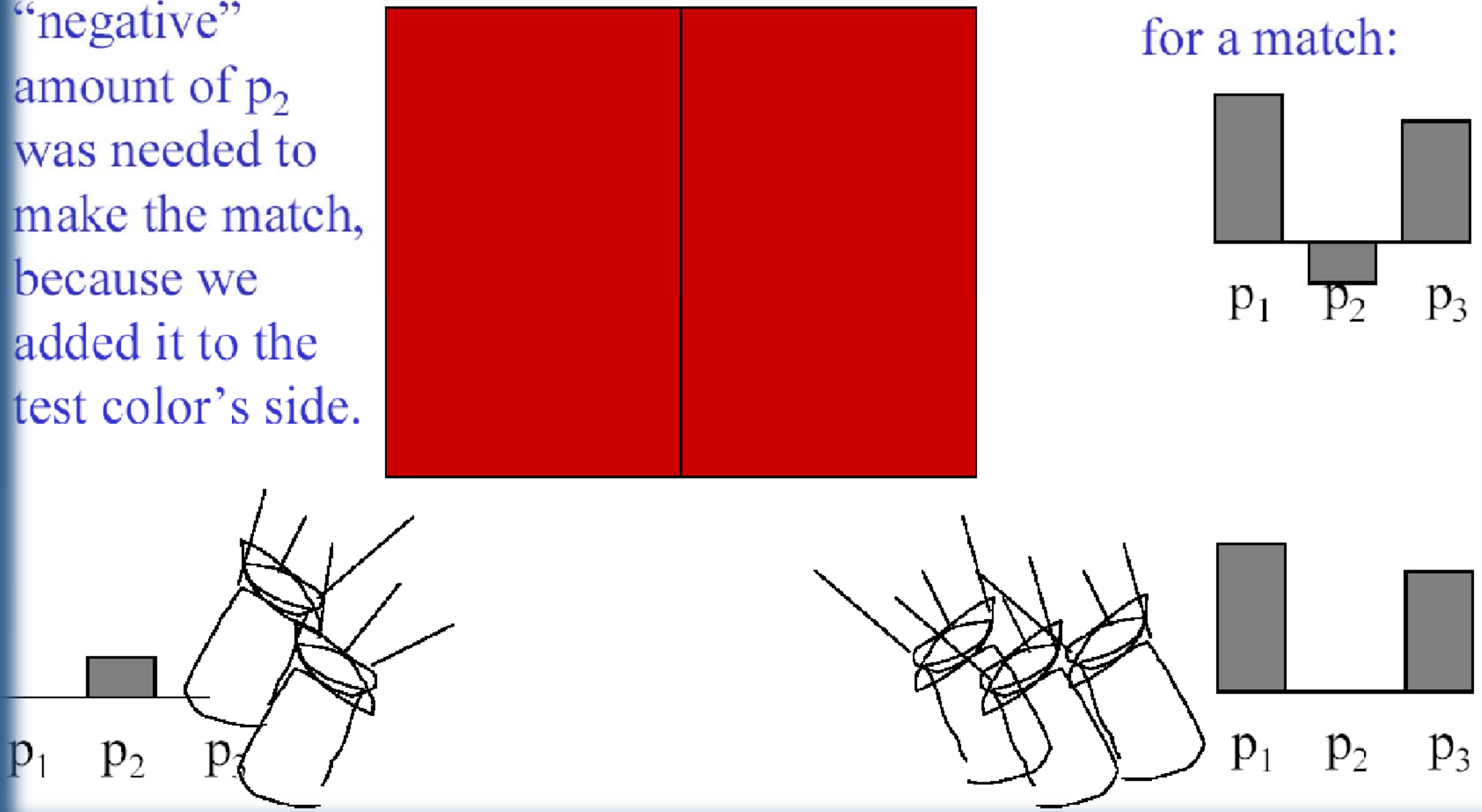
Experiment Step 2



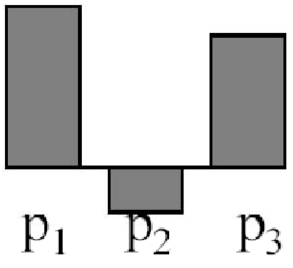
(courtesy of Bill Freeman)

Experiment Step 2

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



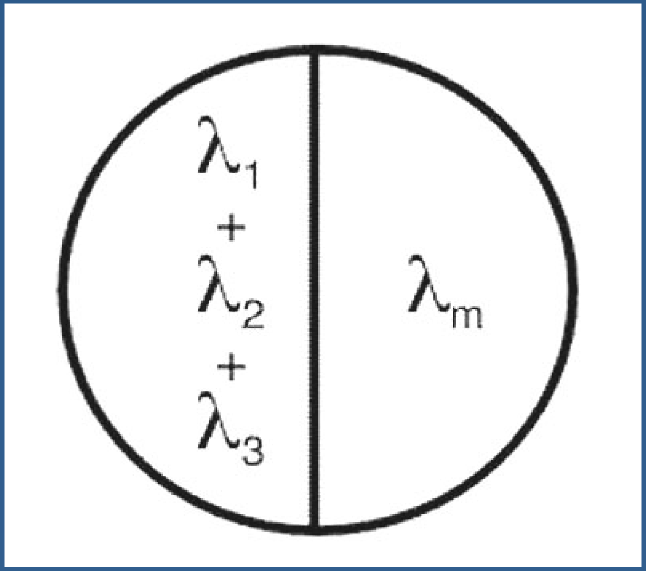
The primary color amounts needed for a match:



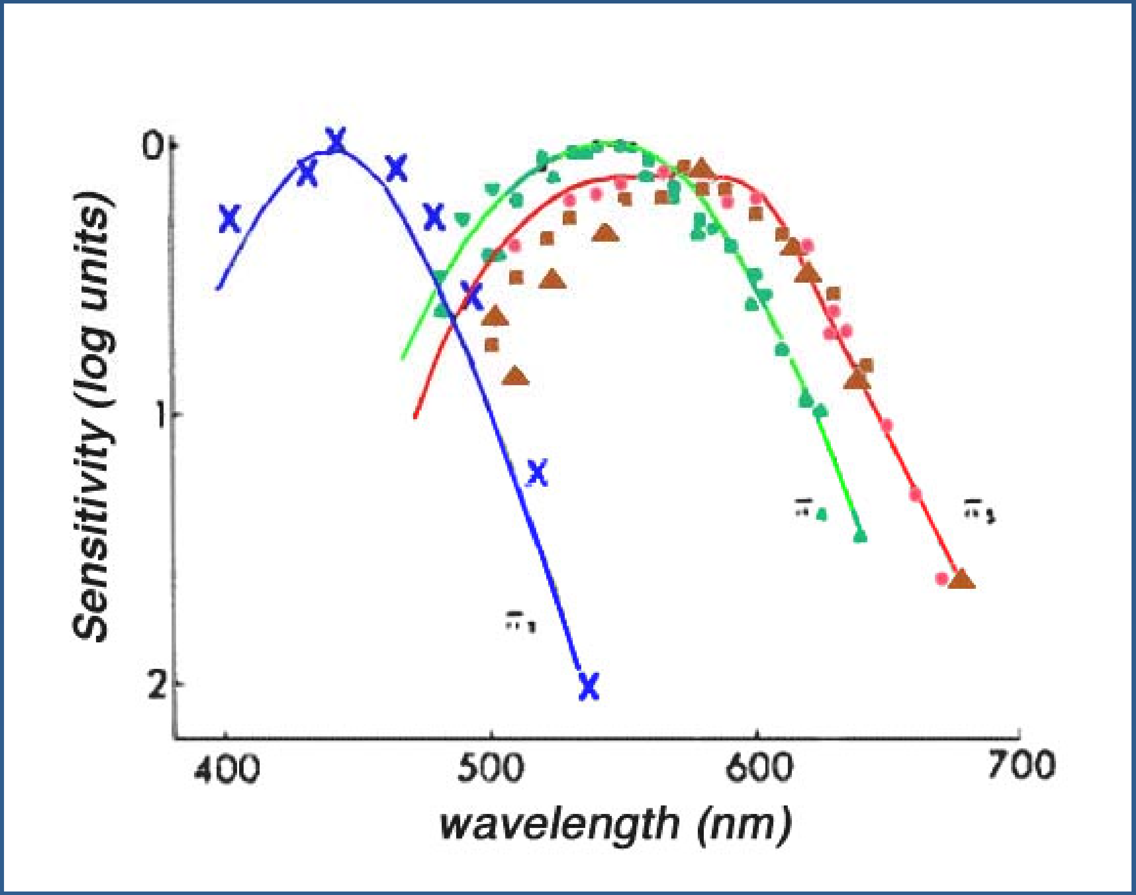
(courtesy of Bill Freeman)

Conclusion from Experiment

- Three primaries are enough for most people to reproduce arbitrary colors
- Why is this the case?



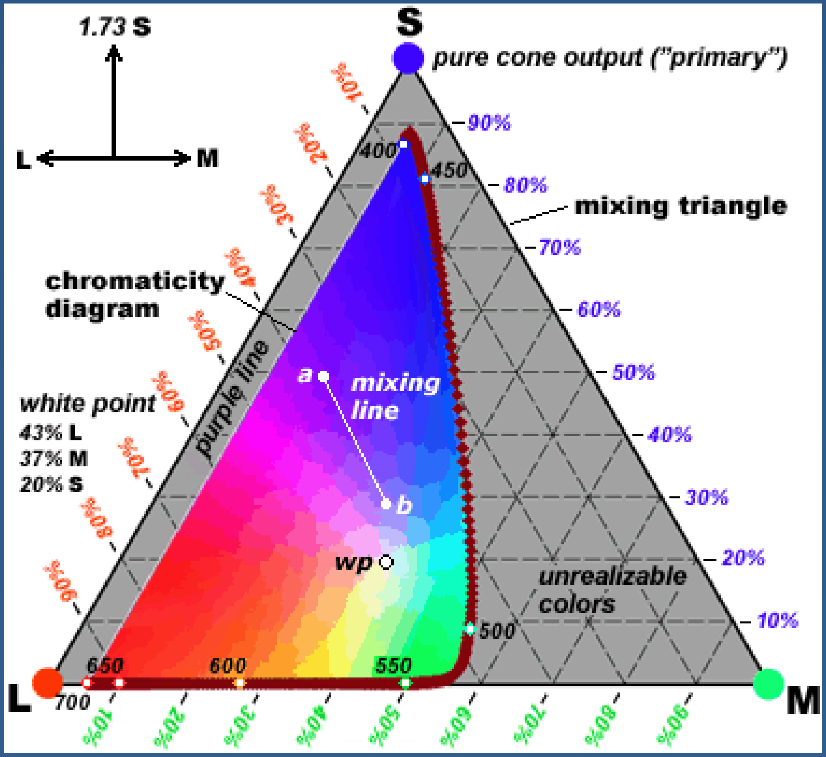
Univariance Principle

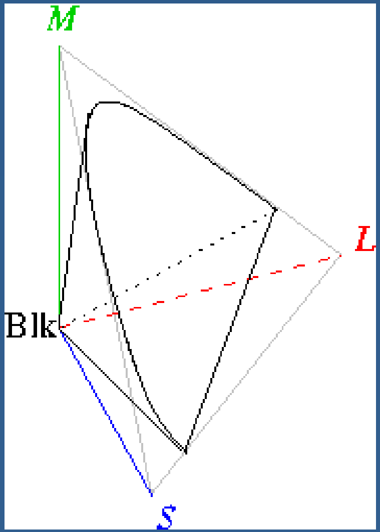


$$L_k = \int \rho_k(\lambda) E(\lambda)\, d\lambda \qquad k = L, M, S \qquad \text{For illuminant } E(\lambda)$$

Tristimulus Color Space

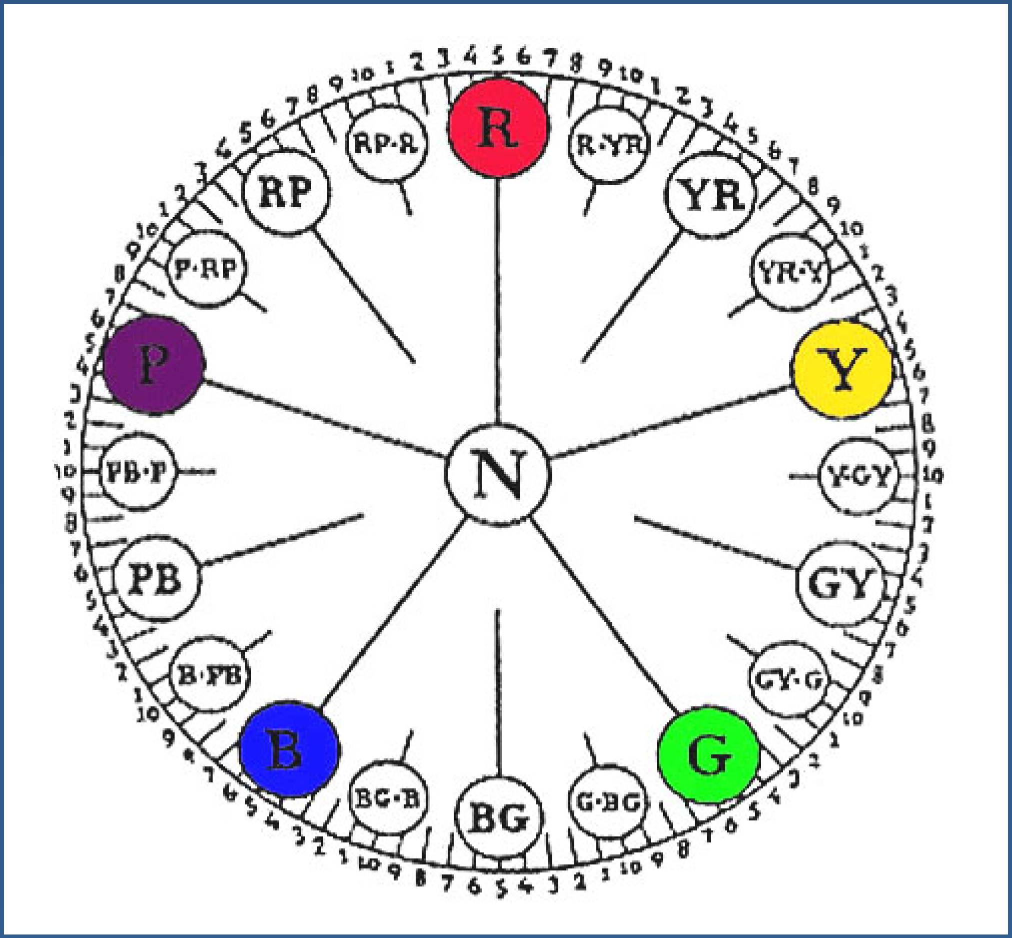
- Three axes represent responses of the three types of receptors
- Horseshoe-shaped cone is the set of all human-perceptible colors



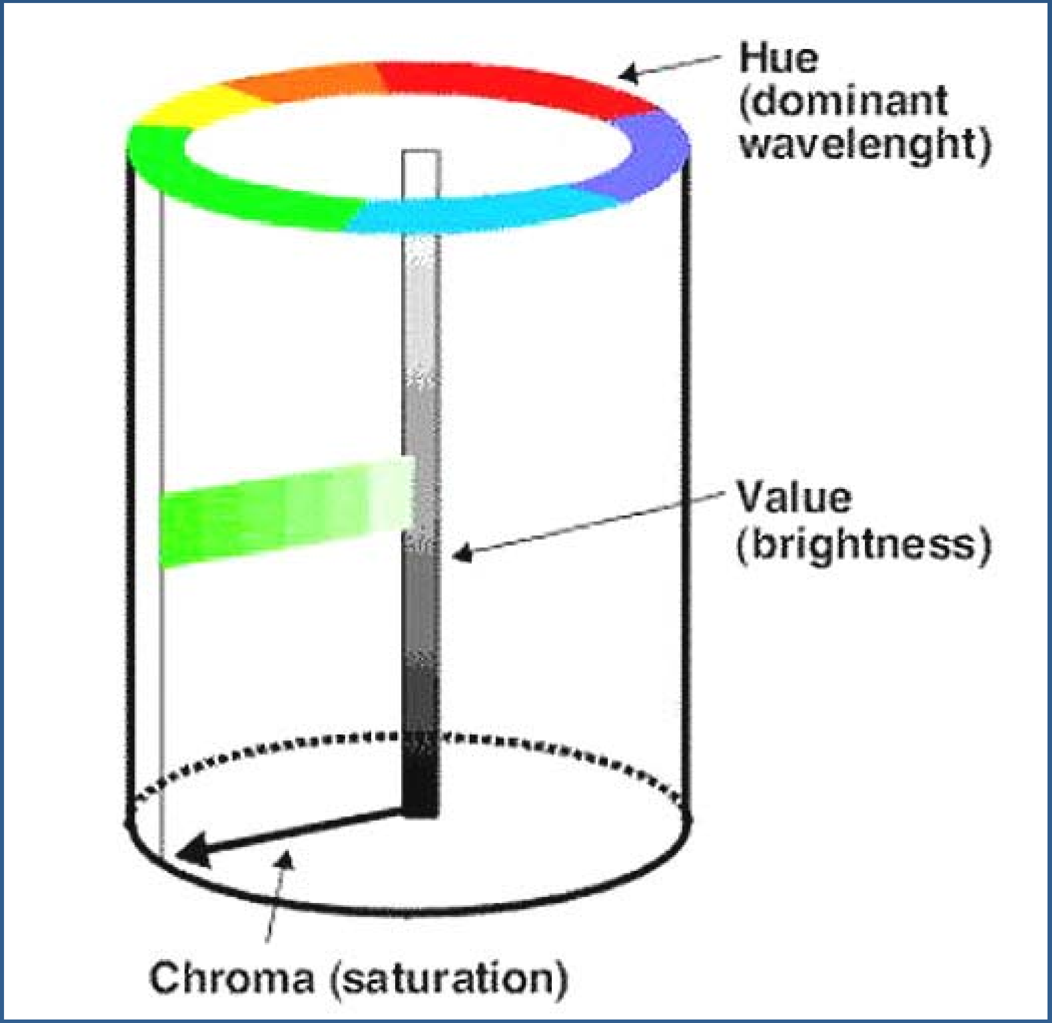


Munsell Color System

- Hue (Wavelength)
- Value (Luminosity)
- Chroma (Saturation)



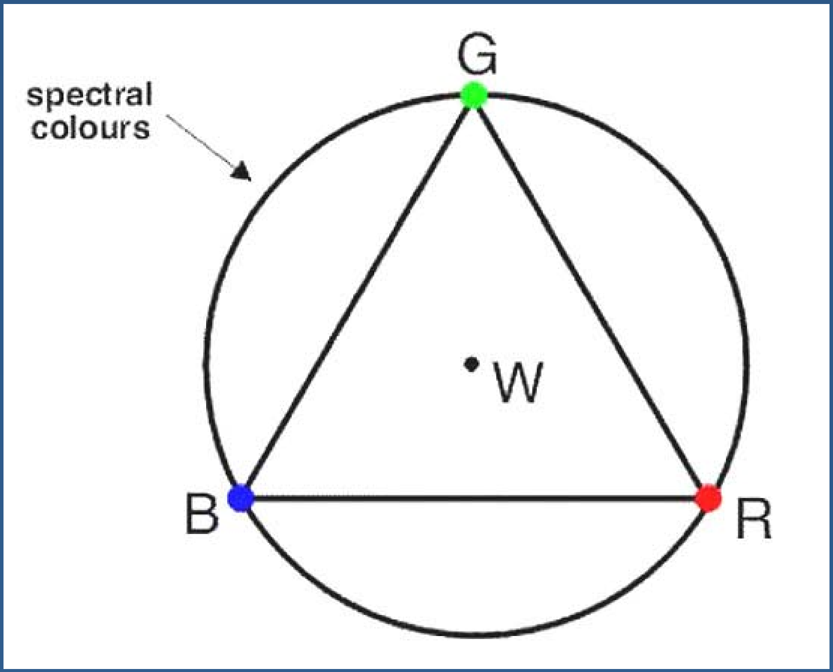
Color Space

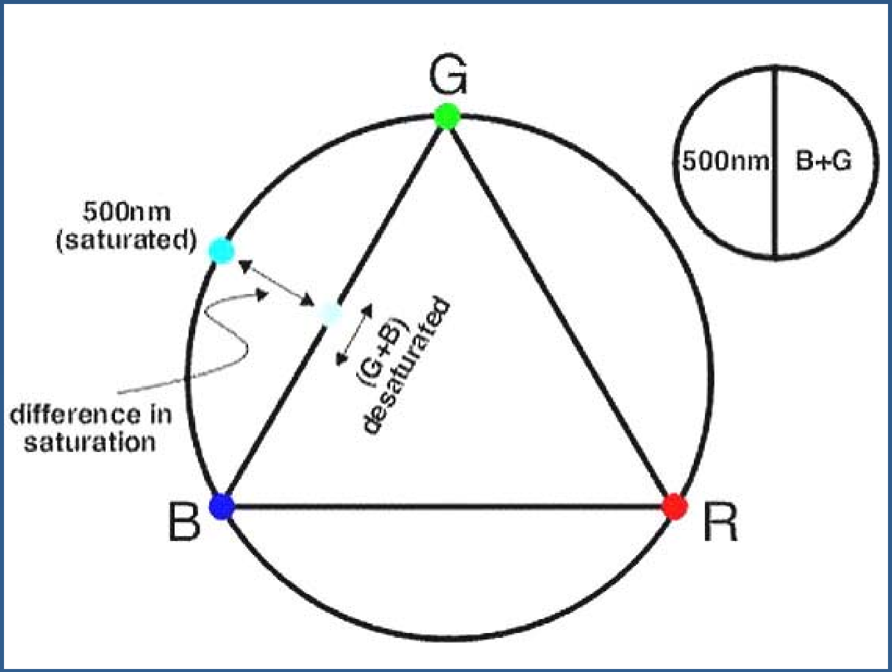

Hue (dominant wavelenght)
Value (brightness)
Chroma (saturation)

How to Match a Color

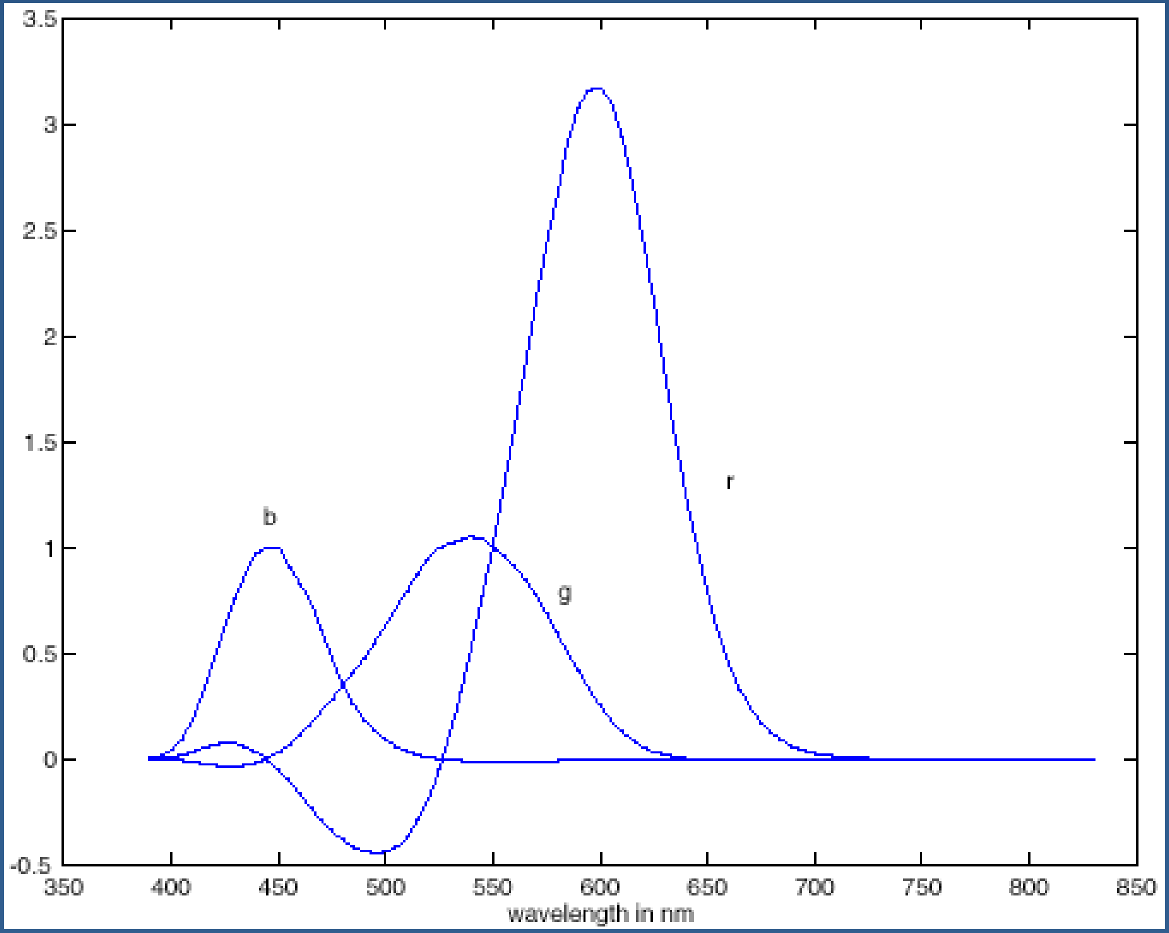
- Choose primaries A, B, C
- Given an energy distribution, which amounts of primaries will match it?
- For each wavelength in the distribution, determine how much of A, B, and C is needed to match light of that wavelength alone
- Add these together to produce a match

Matching on a Color Circle

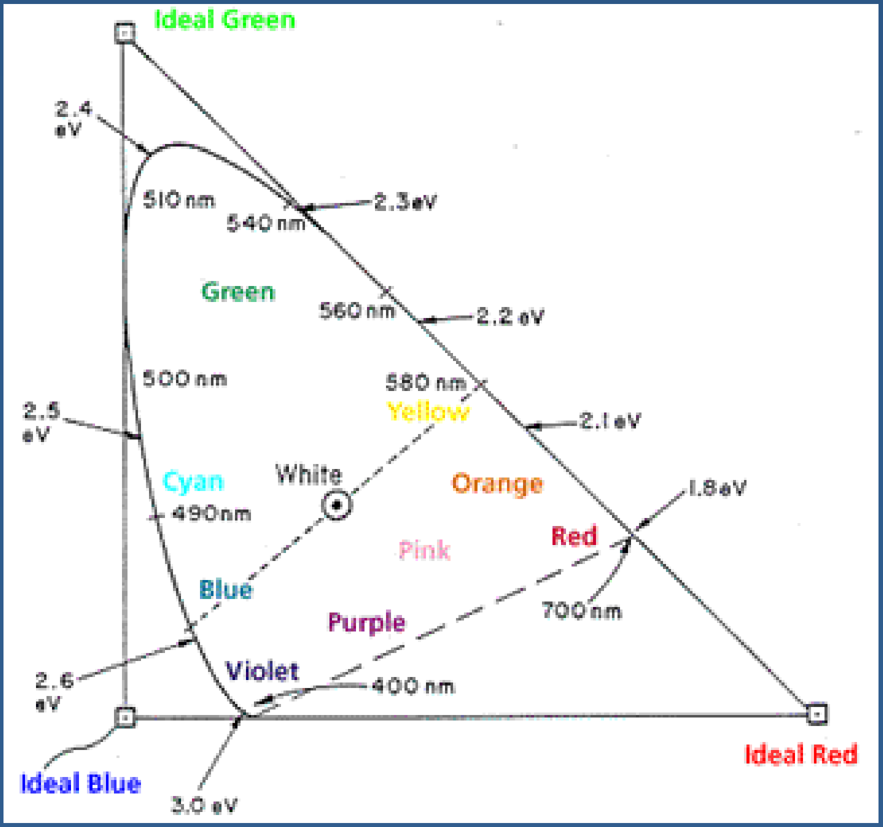


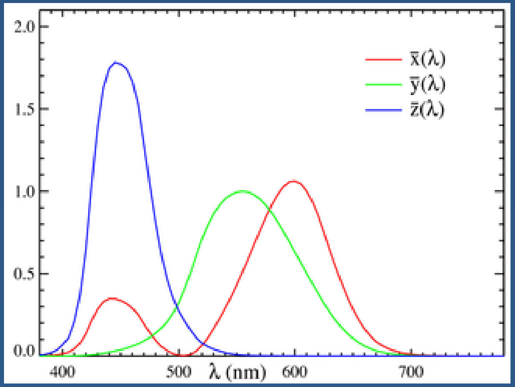


RGB Color Matching



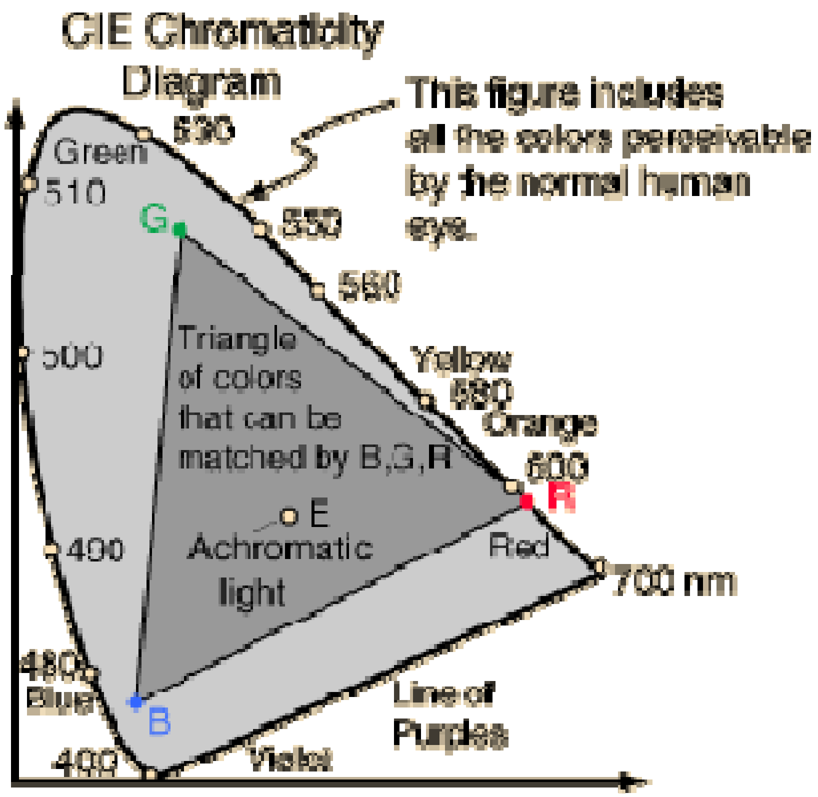
Geometry of Color (CIE)



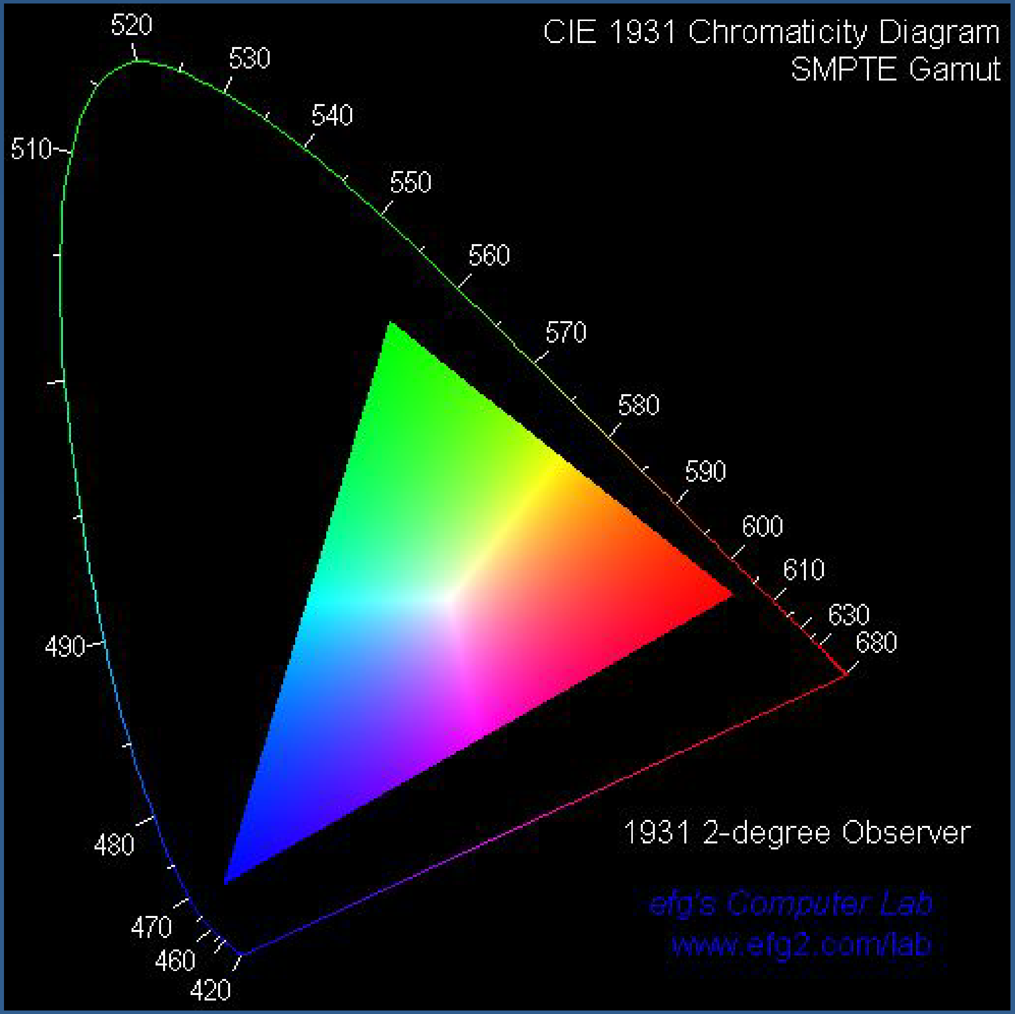


- Perceptual color spaces are non-convex
- Three primaries can span the space, but weights may be negative.
- Curved outer edge consists of single wavelength primaries

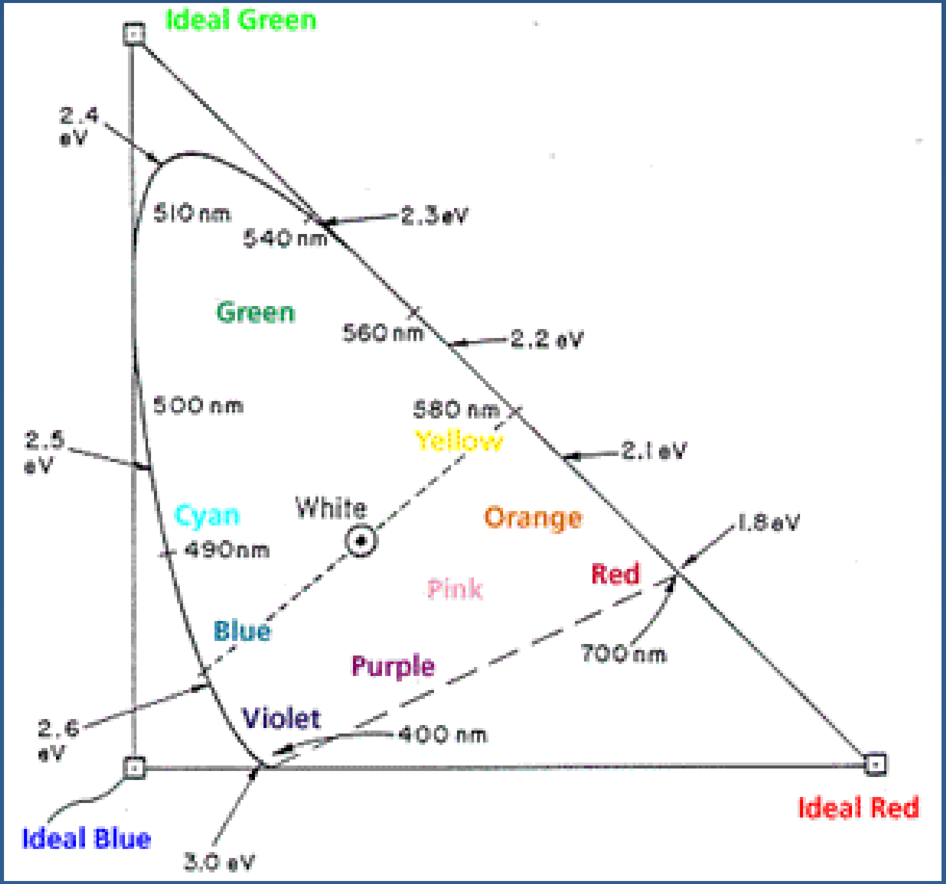
CIE Chromaticity Diagram

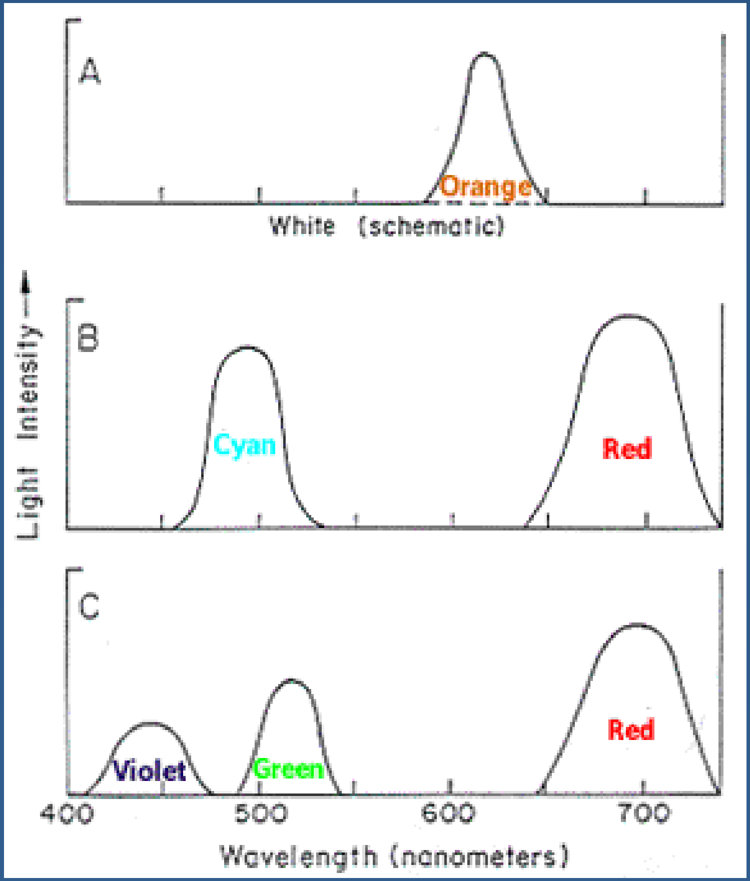


RGB Color Space



Constructing Colors





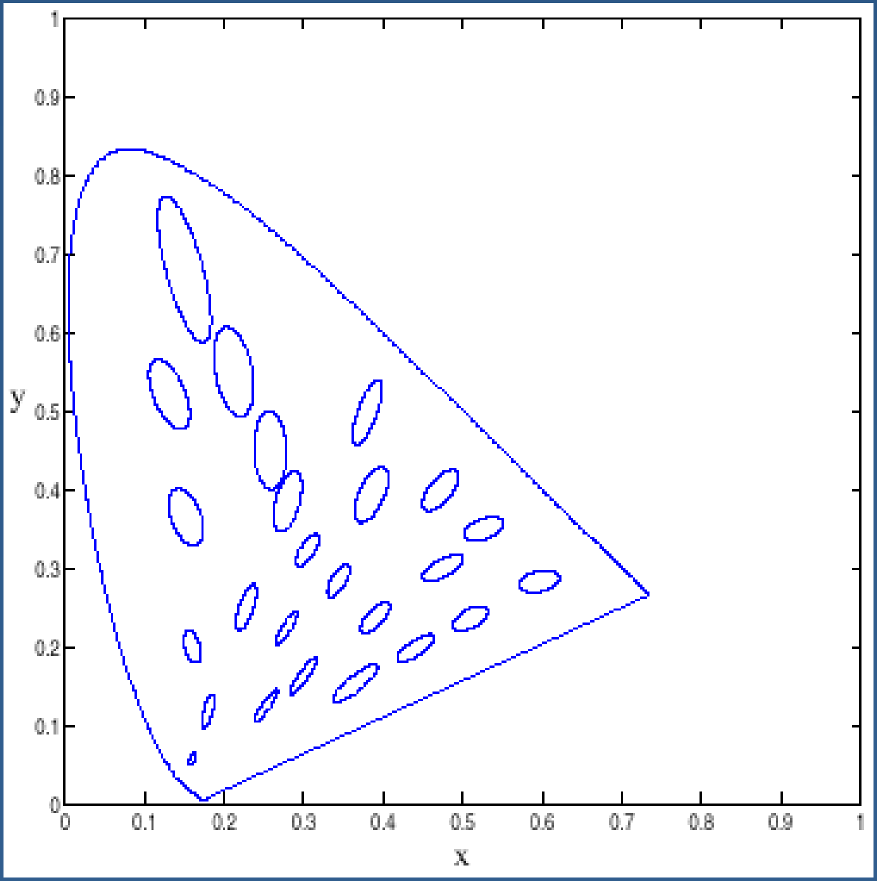
Color can be constructed in many ways

Color Spaces

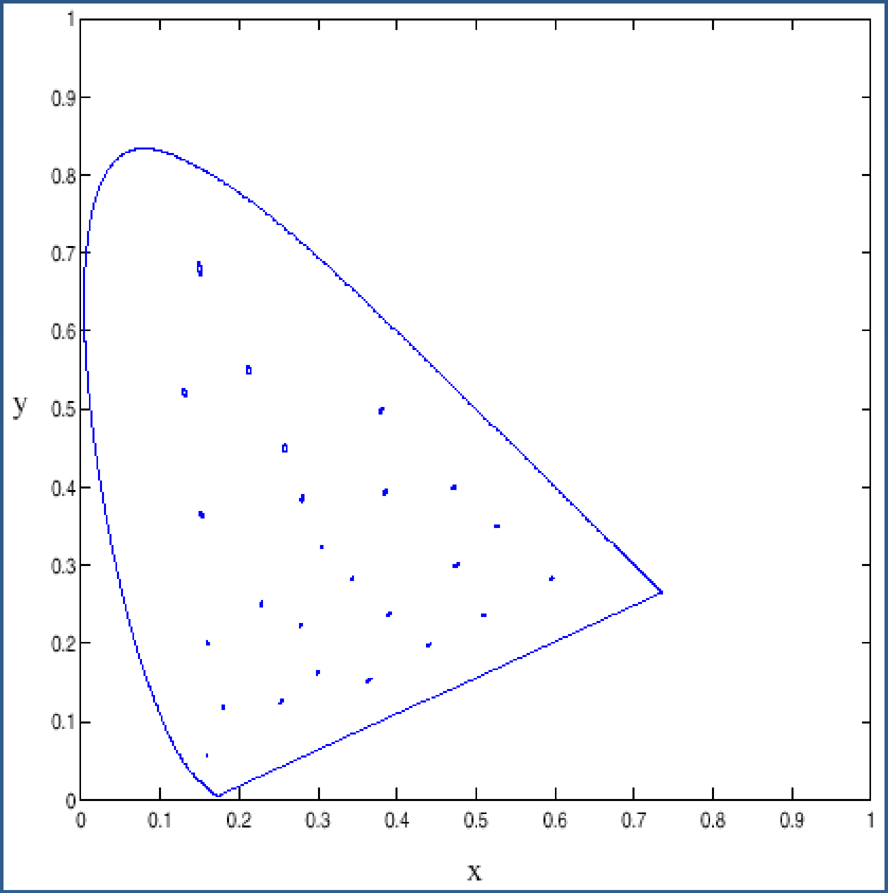
- Use color matching functions to define a coordinate system for color
- Each color can be assigned a triple of coordinates with respect to some color space (e.g. RGB)
- Devices (monitors, printers, projectors) and computers can communicate colors precisely

McAdam Ellipses

(10 times actual size)



(Actual size)



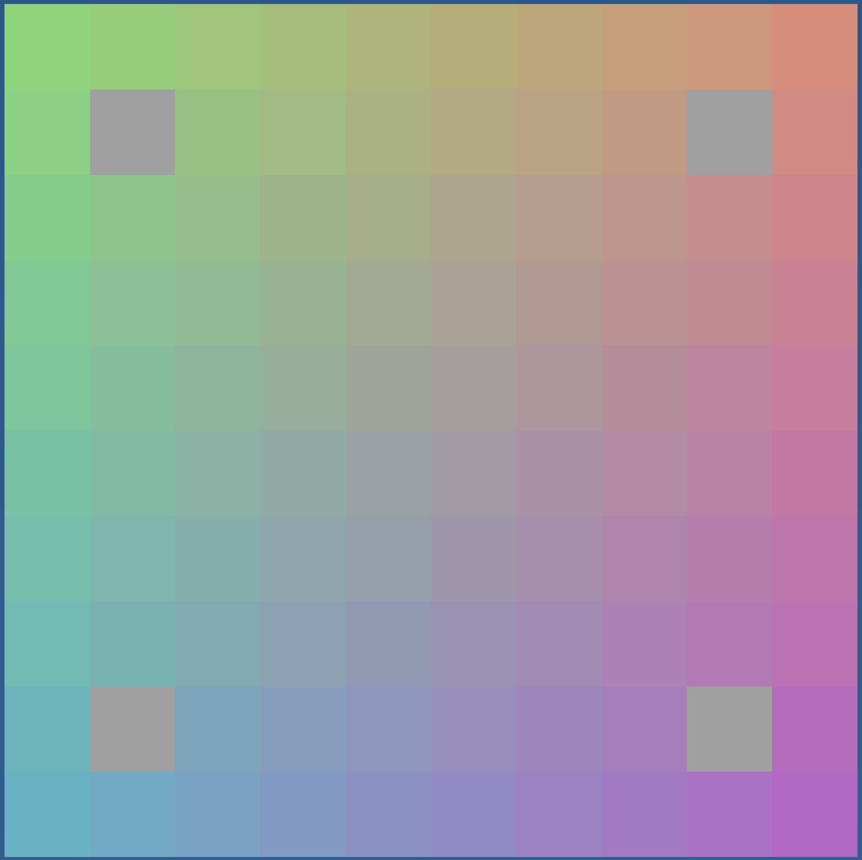
(courtesy of D. Forsyth)

Human Color Constancy

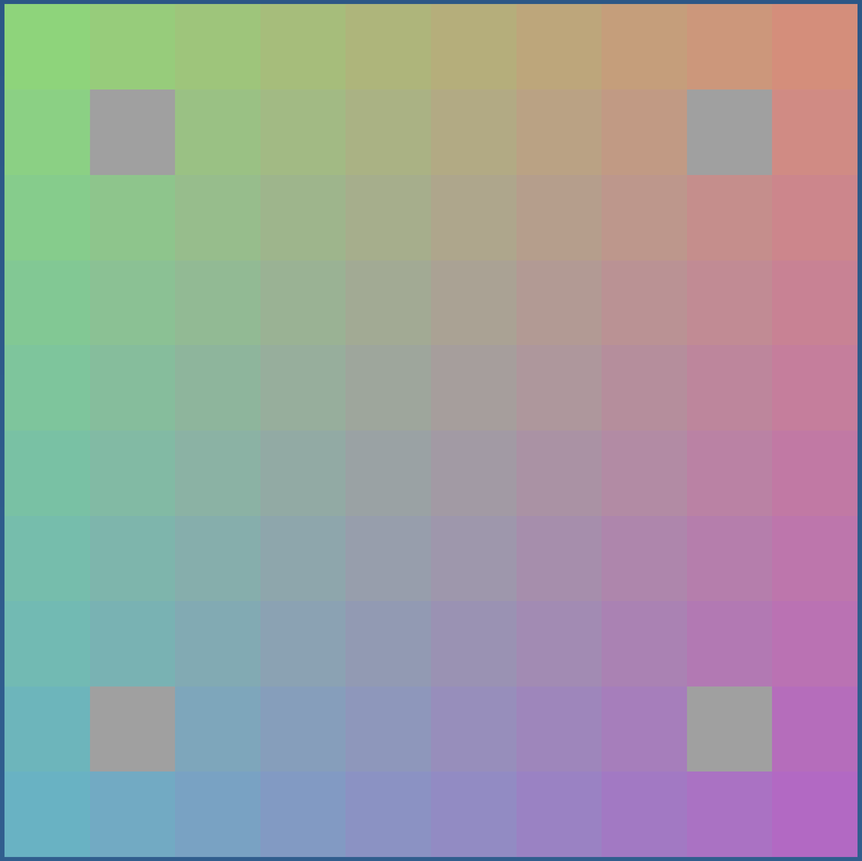
- Distinguish between
  - Color constancy, which refers to hue and saturation
  - Lightness constancy, which refers to gray-level.
- Humans can perceive
  - Color a surface would have under white light (surface color)
  - Color of the reflected light (limited ability to separate surface color from measured color)
  - Color of illuminant (even more limited)

(courtesy of J.M. Rehg)

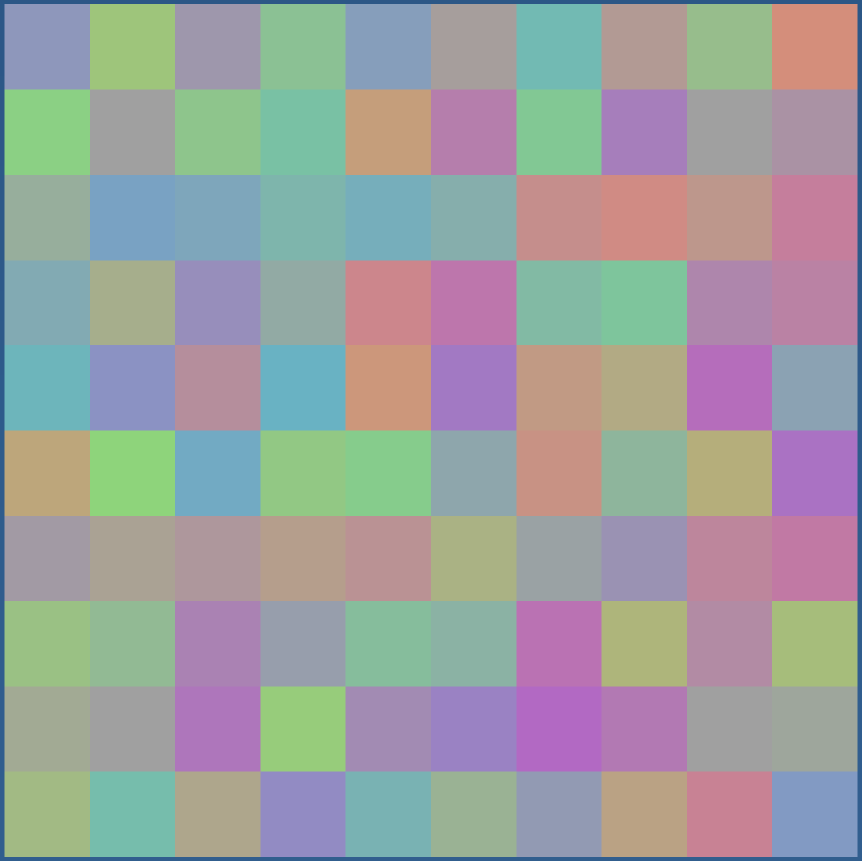
Spatial Arrangement and Color Perception



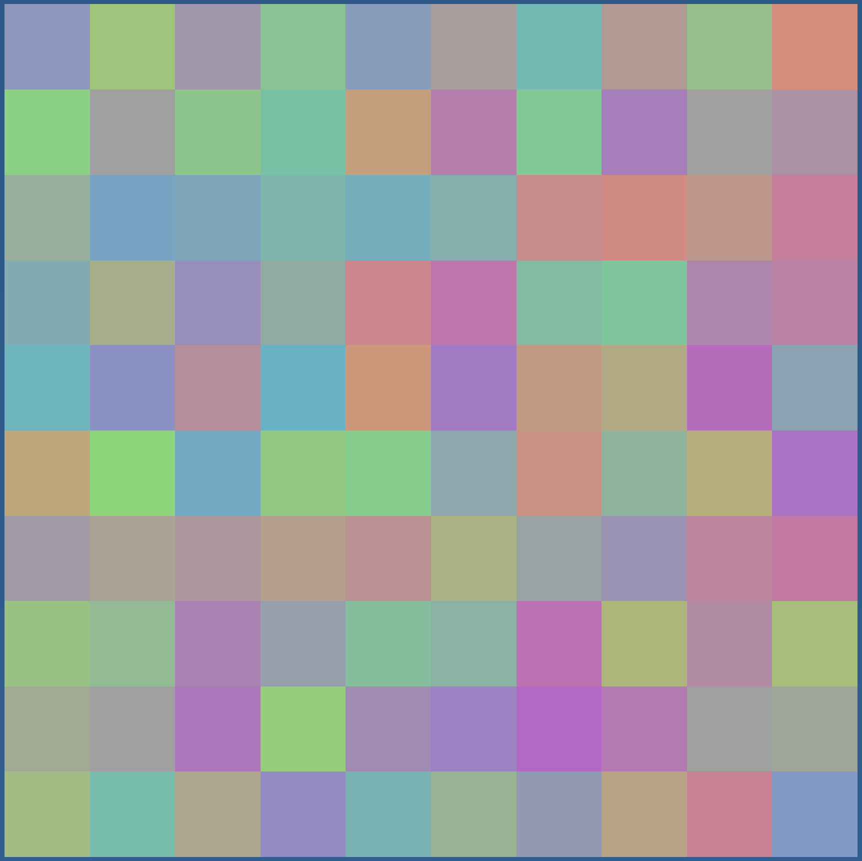
Spatial Arrangement and Color Perception



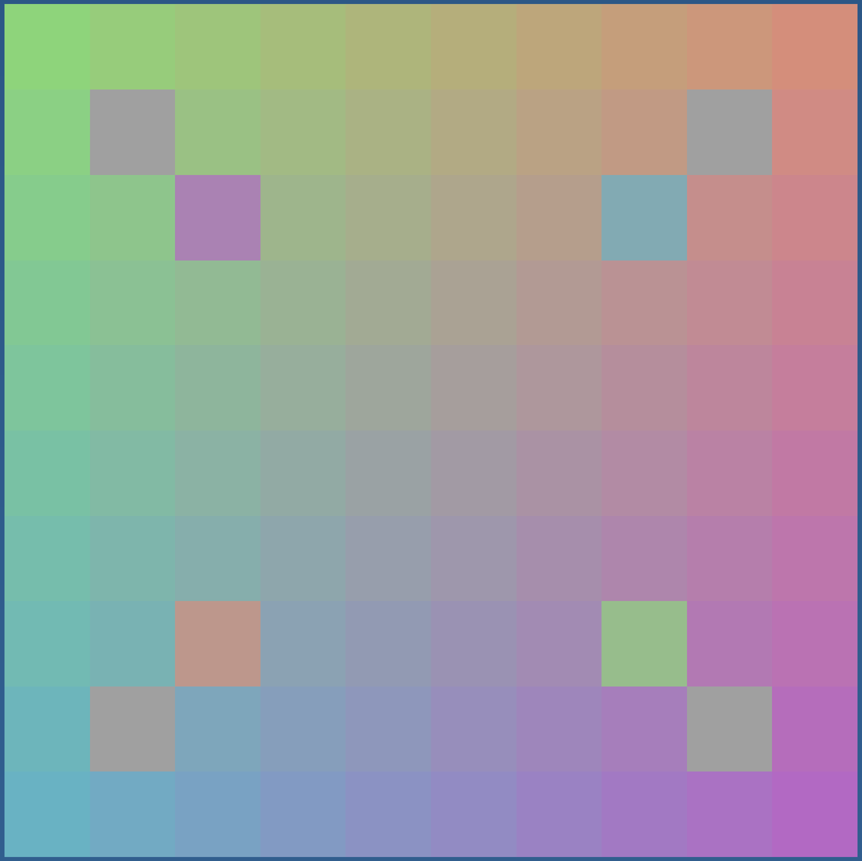
Spatial Arrangement and Color Perception

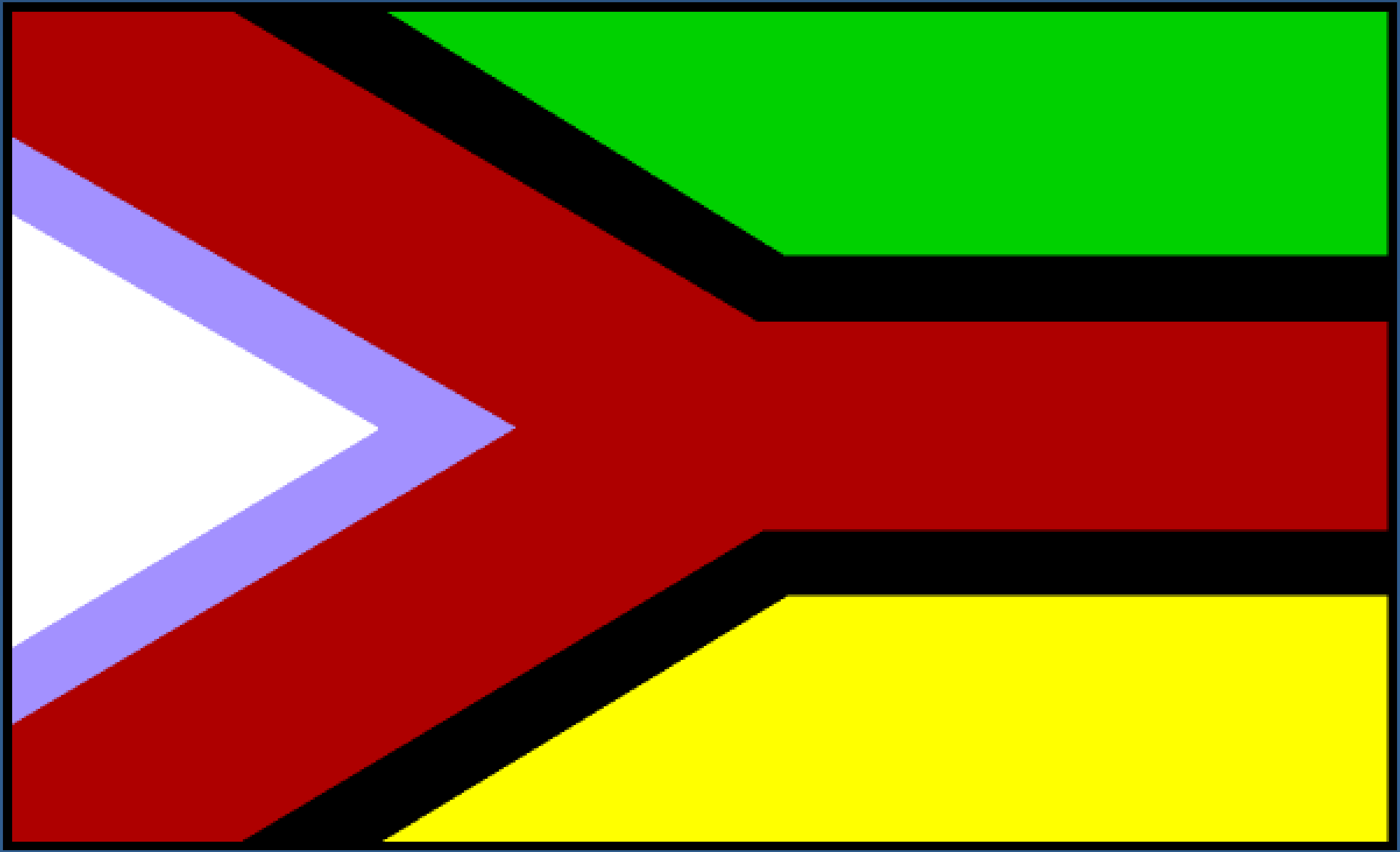


Spatial Arrangement and Color Perception



Spatial Arrangement and Color Perception

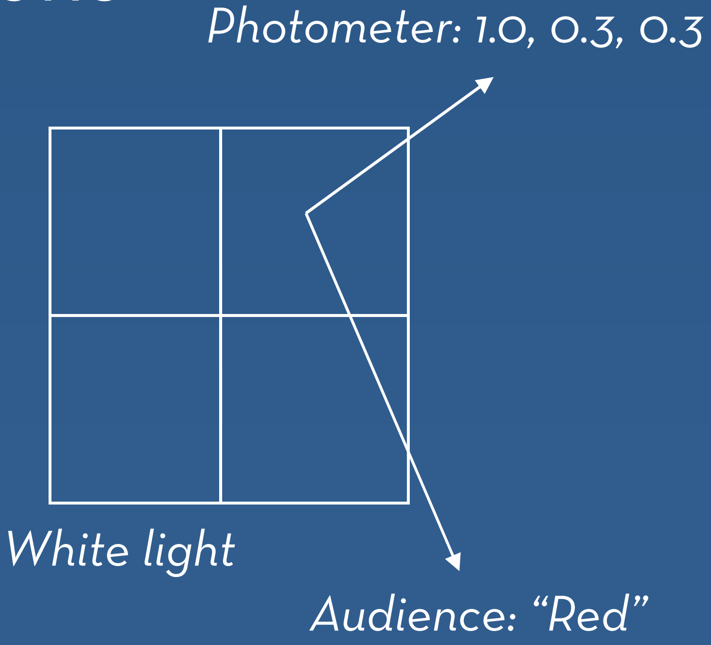


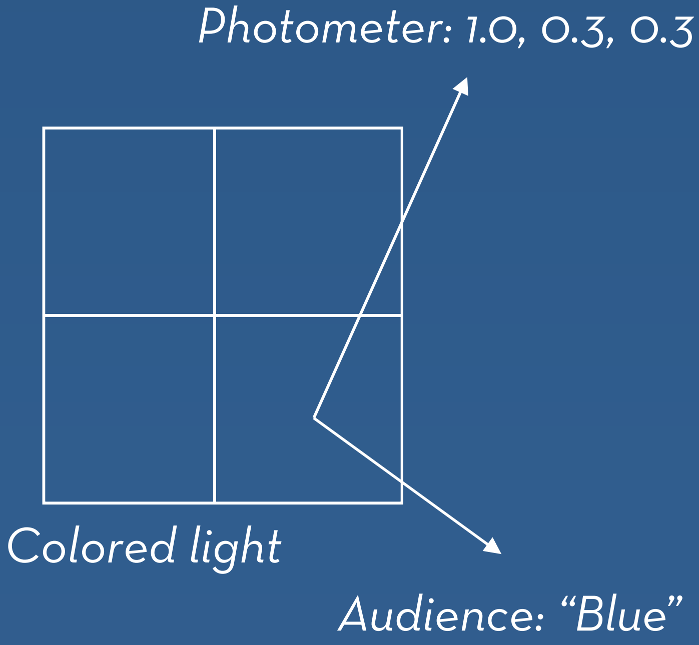


(courtesy of D. Forsyth)

Land’s Mondrian Experiments

- Squares of color with the same color radiance yield very different color perceptions



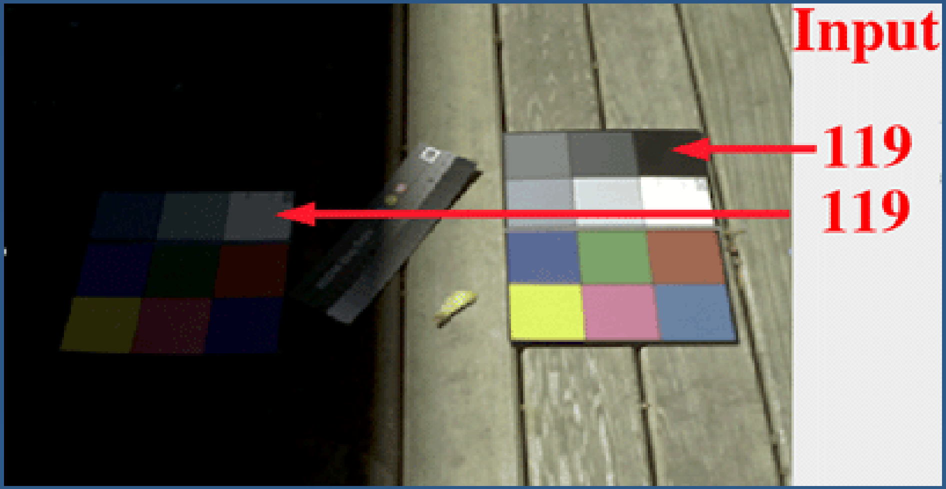


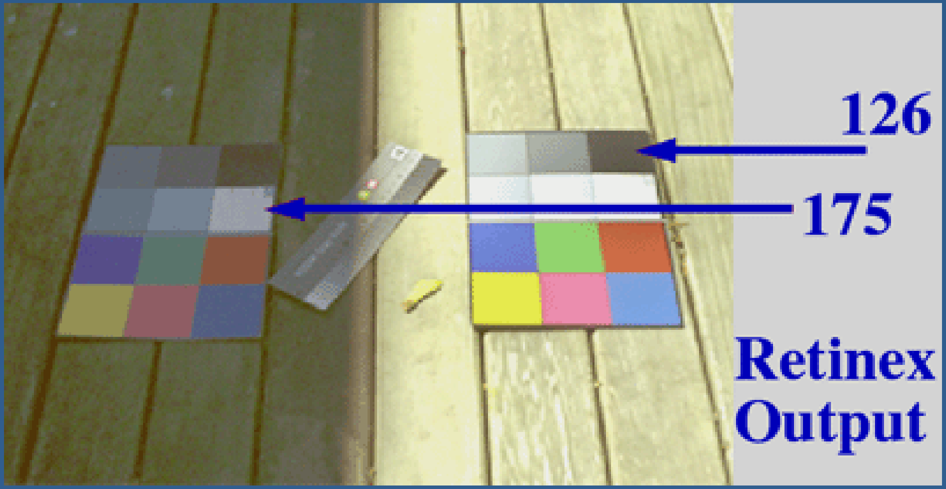
(courtesy of J.M. Rehg)

Lightness Constancy Algorithm

- The goal is to determine what the surfaces in the image would look like under white light.
- Compares the brightness of patches across their common boundaries and computes relative brightness
- Establish an absolute reference for lightness (e.g. brightest point is “white”)

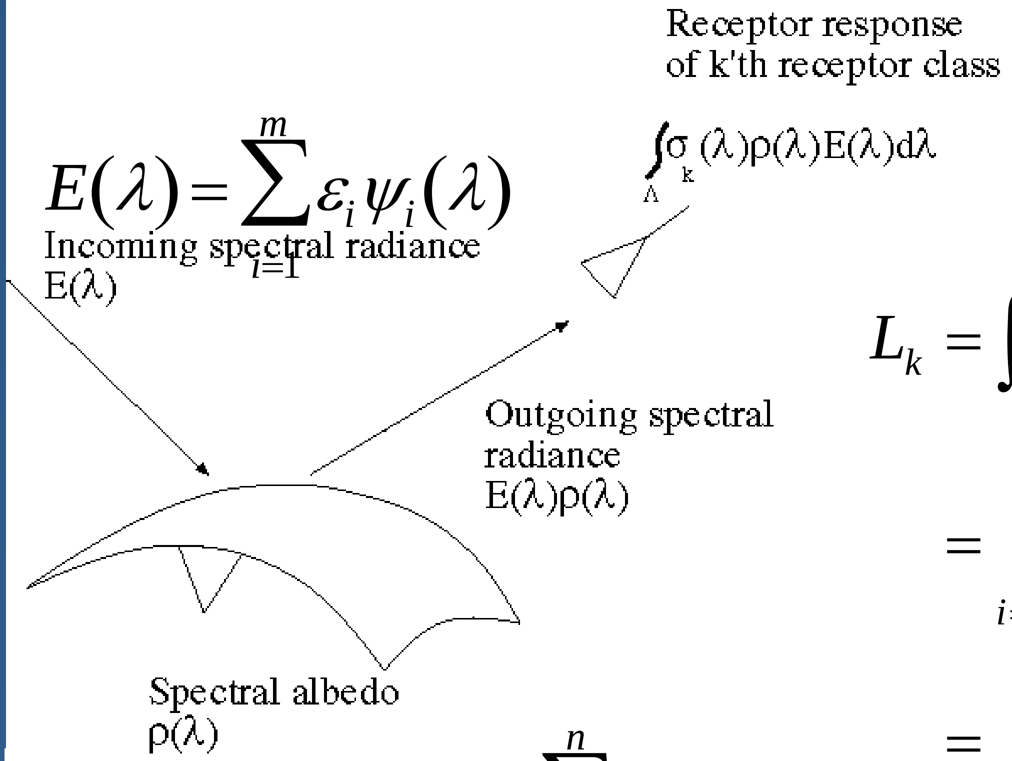
Lightness Constancy Example





(courtesy of John McCann)

Finite Dimensional Linear Models



$$E(\lambda)=\sum_{i=1}^{m}\varepsilon_i\psi_i(\lambda)$$

$$\rho(\lambda)=\sum_{j=1}^{n}r_j\varphi_j(\lambda)$$

$$\begin{aligned}L_k &= \int \sigma_k(\lambda)\left(\sum_{i=1}^{m}\varepsilon_i\psi_i(\lambda)\right)\left(\sum_{j=1}^{n}r_j\varphi_j(\lambda)\right)d\lambda \\ &= \sum_{i=1,\,j=1}^{m,n}\varepsilon_i r_j\int\sigma_k(\lambda)\psi_i(\lambda)\varphi_j(\lambda)d\lambda \\ &= \sum_{i=1,\,j=1}^{m,n}\varepsilon_i r_j g_{ijk}\end{aligned}$$

From Images to Objects and Regions

- Attributes of regions
  - Bounding edges
  - Texture
- The need to compute and reason about spatial aggregations of pixels leads us to filtering
- Key problem is *segmentation*

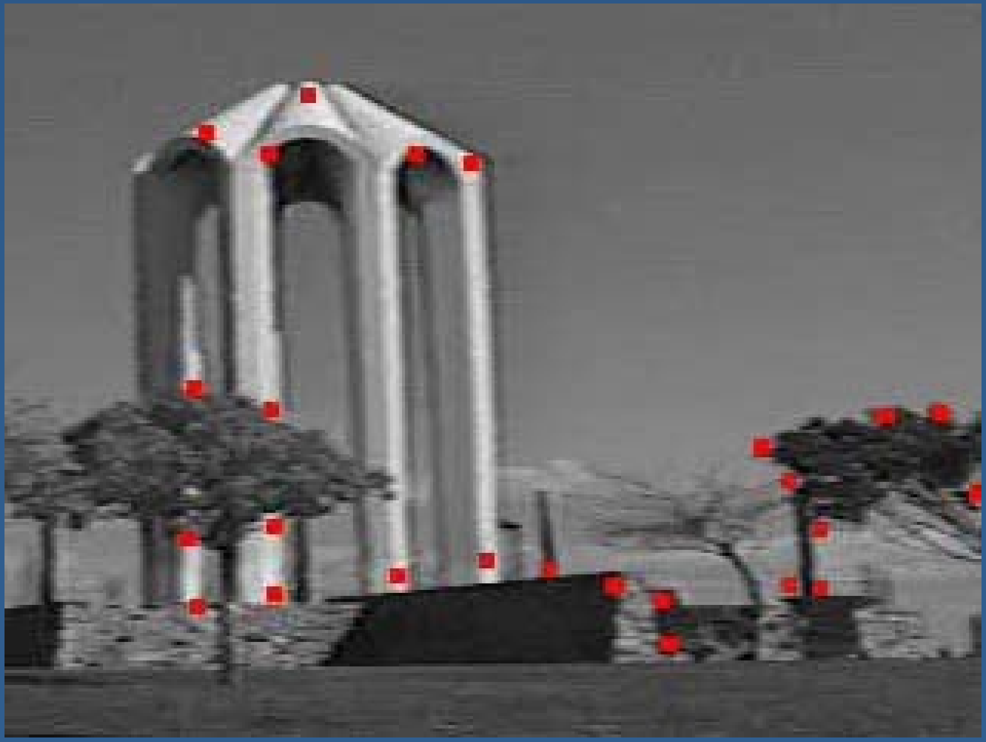
Features and Filters: Questions

- What is a feature?
- What is an image filter? What is it good for?

- How can we find corners?
- How can we find edges?

What is a Feature?

- In computer vision, a *feature* is a local, meaningful, detectable part of an image



Why Use Features?

- High information content
- Invariant to changing viewpoint or changing illumination
- Reduces computational burden

(courtesy of Sebastian Thrun)

One Type of Computer Vision

Image 1

Feature 1
Feature 2
:
Feature N

Computer Vision Algorithm

Image 2

Feature 1
Feature 2
:
Feature N

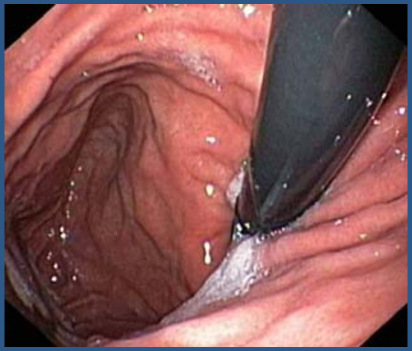
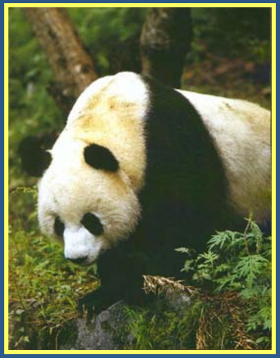
(courtesy of Sebastian Thrun)

Where Features Are Used

- Calibration
- Image Segmentation
- Correspondence in multiple images (stereo, structure from motion)
- Object detection, classification

(courtesy of Sebastian Thrun)

What Makes a Good Feature?

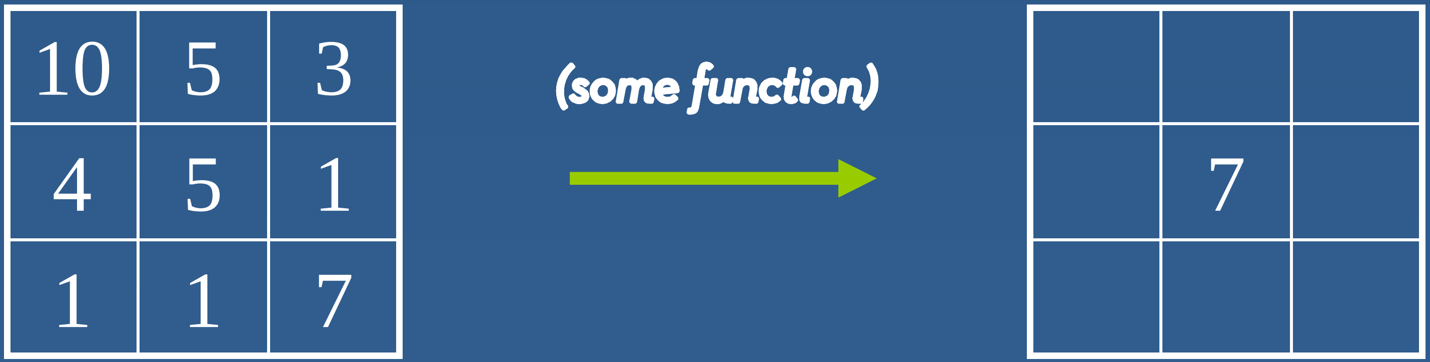



- Invariance
  - View point (scale, orientation, translation)
  - Lighting condition
  - Object deformations
  - Partial occlusion
- Other Characteristics
  - Uniqueness
  - Sufficiently many
  - Tuned to the task

(courtesy of Sebastian Thrun)

What is Image Filtering?

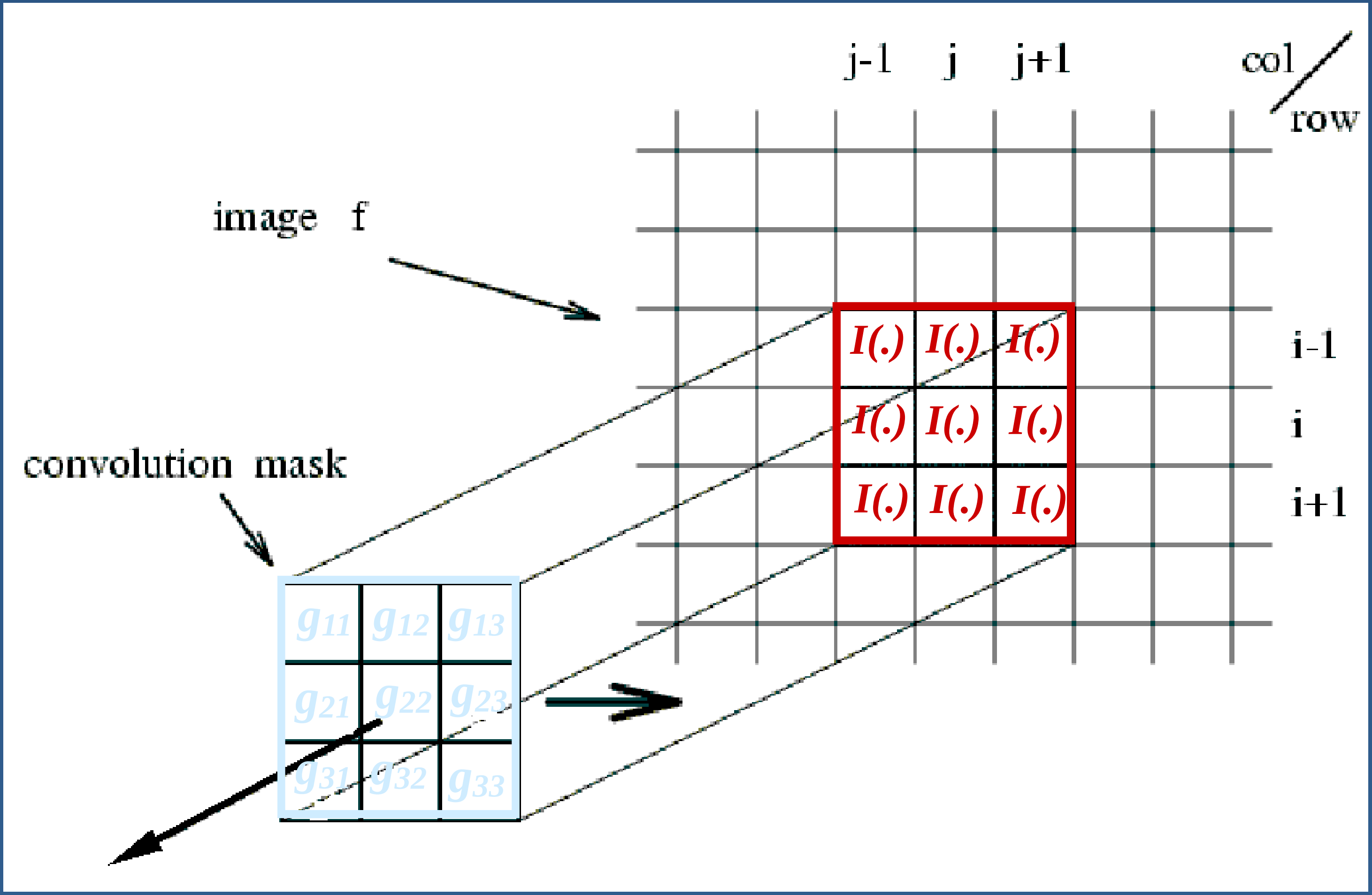
- Modify the pixels in an image based on some function of a local neighborhood of the pixels



Linear Filtering

- Linear case is simplest and most useful
  - Replace each pixel with a linear combination of its neighbors.
- The specification of the linear combination is called the convolution kernel.

$$\begin{array}{|c|c|c|}\hline 10 & 5 & 3 \\ \hline 4 & 5 & 1 \\ \hline 1 & 1 & 7 \\ \hline \end{array} * \underset{\textit{kernel}}{\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0.5 & 0 \\ \hline 0 & 1.0 & 0.5 \\ \hline \end{array}} = \begin{array}{|c|c|c|}\hline & & \\ \hline & 7 & \\ \hline & & \\ \hline \end{array}$$

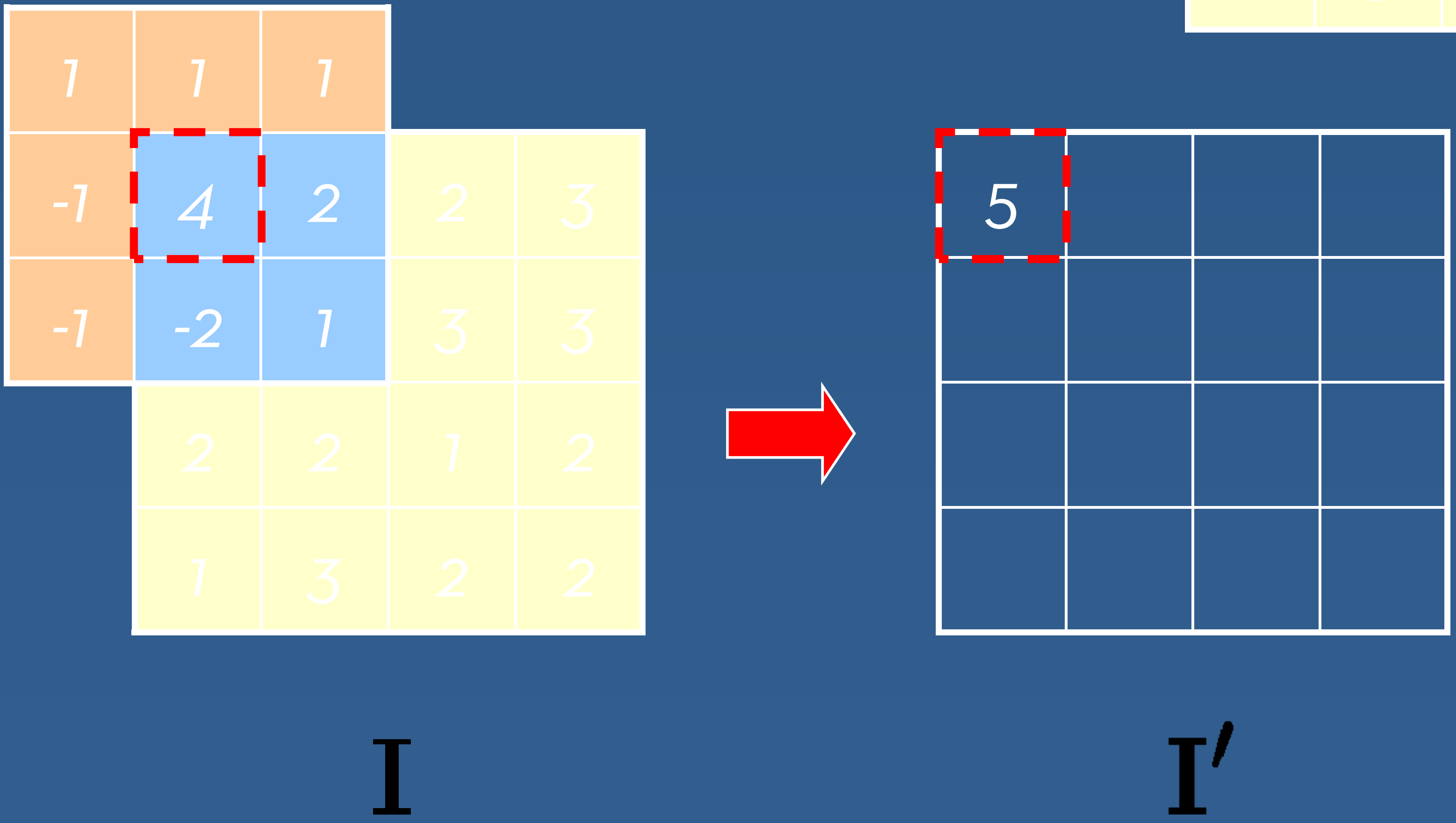

j-1
j
j+1
col
row
image f
i-1
i
i+1
I(.)
convolution mask
g_{11} g_{12} g_{13}
g_{21} g_{22} g_{23}
g_{31} g_{32} g_{33}

$$
\begin{aligned}
f(i,j) = \; & g_{11}\, I(i-1,j-1) + g_{12}\, I(i-1,j) + g_{13}\, I(i-1,j+1) + \\
& g_{21}\, I(i,j-1) + g_{22}\, I(i,j) + g_{23}\, I(i,j+1) + \\
& g_{31}\, I(i+1,j-1) + g_{32}\, I(i+1,j) + g_{33}\, I(i+1,j+1)
\end{aligned}
$$

Step 1

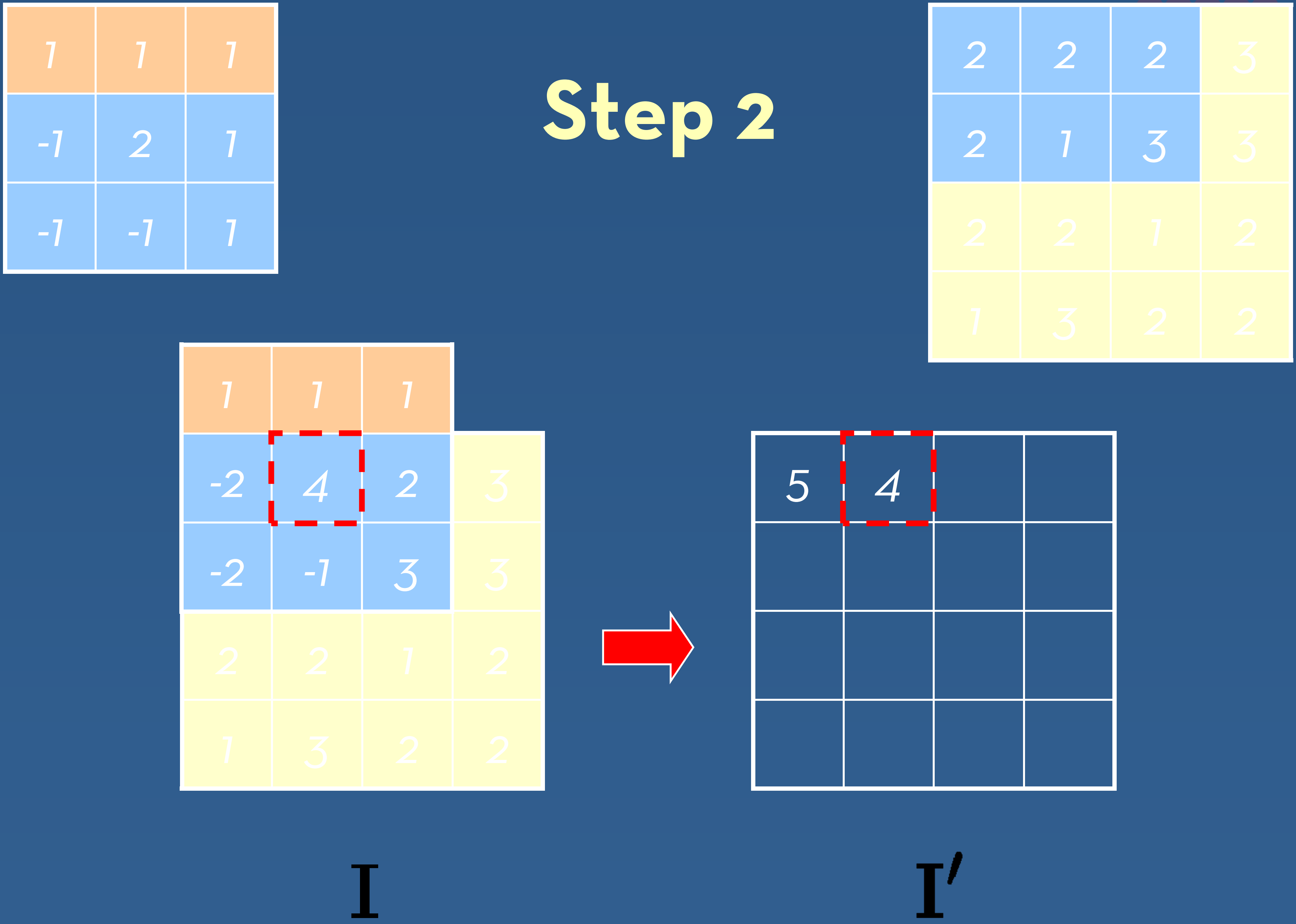
1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2



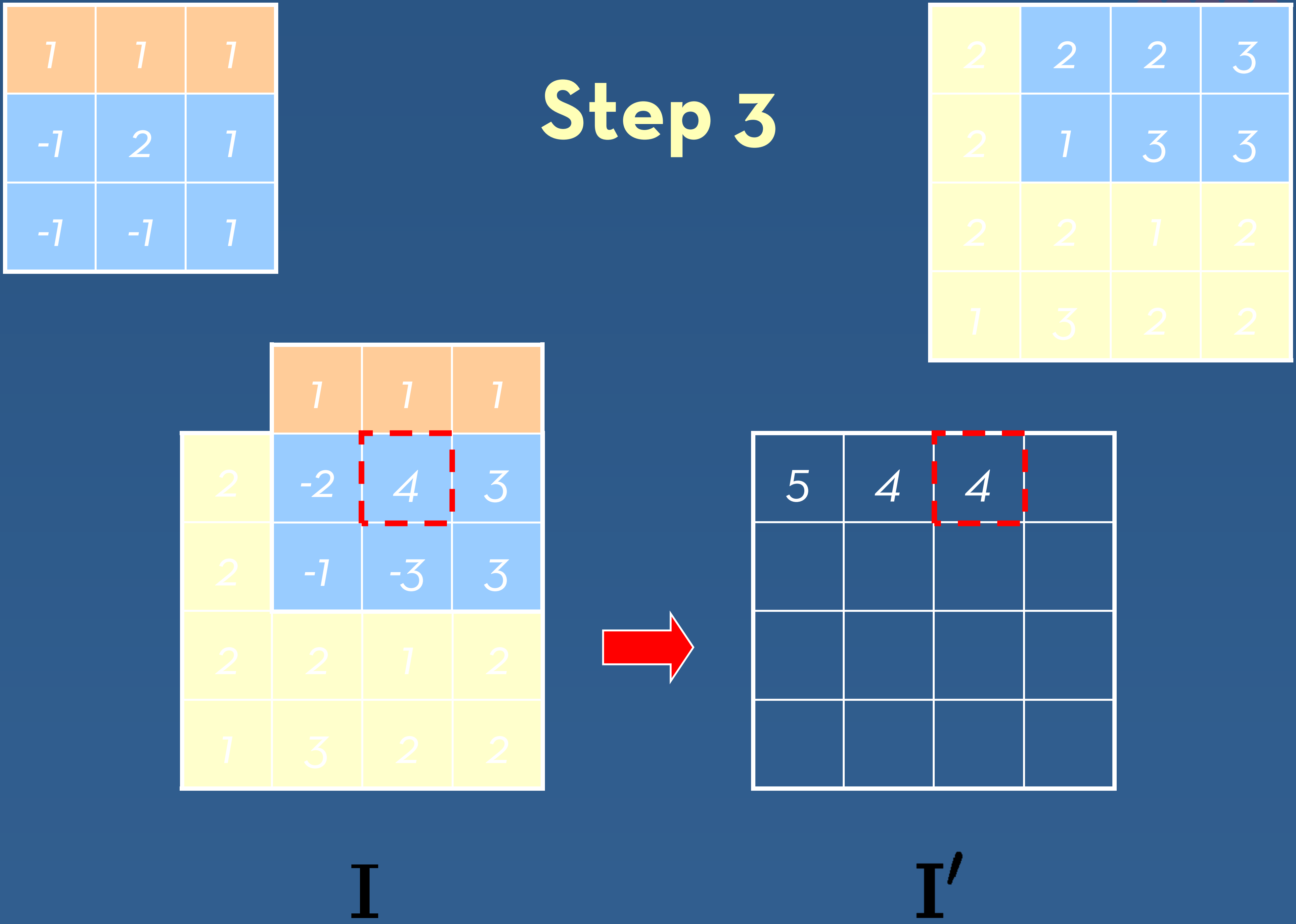
(courtesy of Christopher Rasmussen)

Step 2



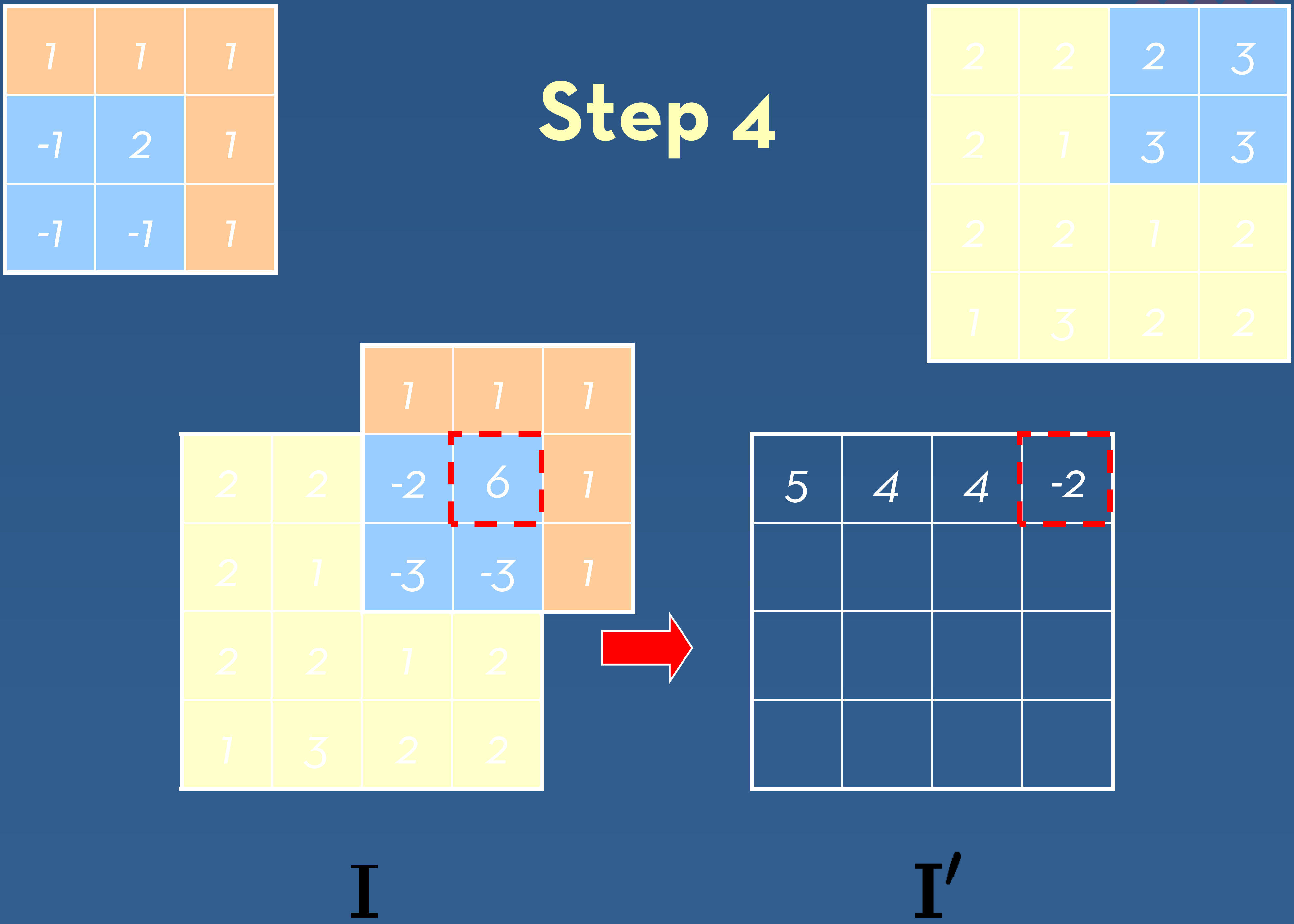
(courtesy of Christopher Rasmussen)

Step 3



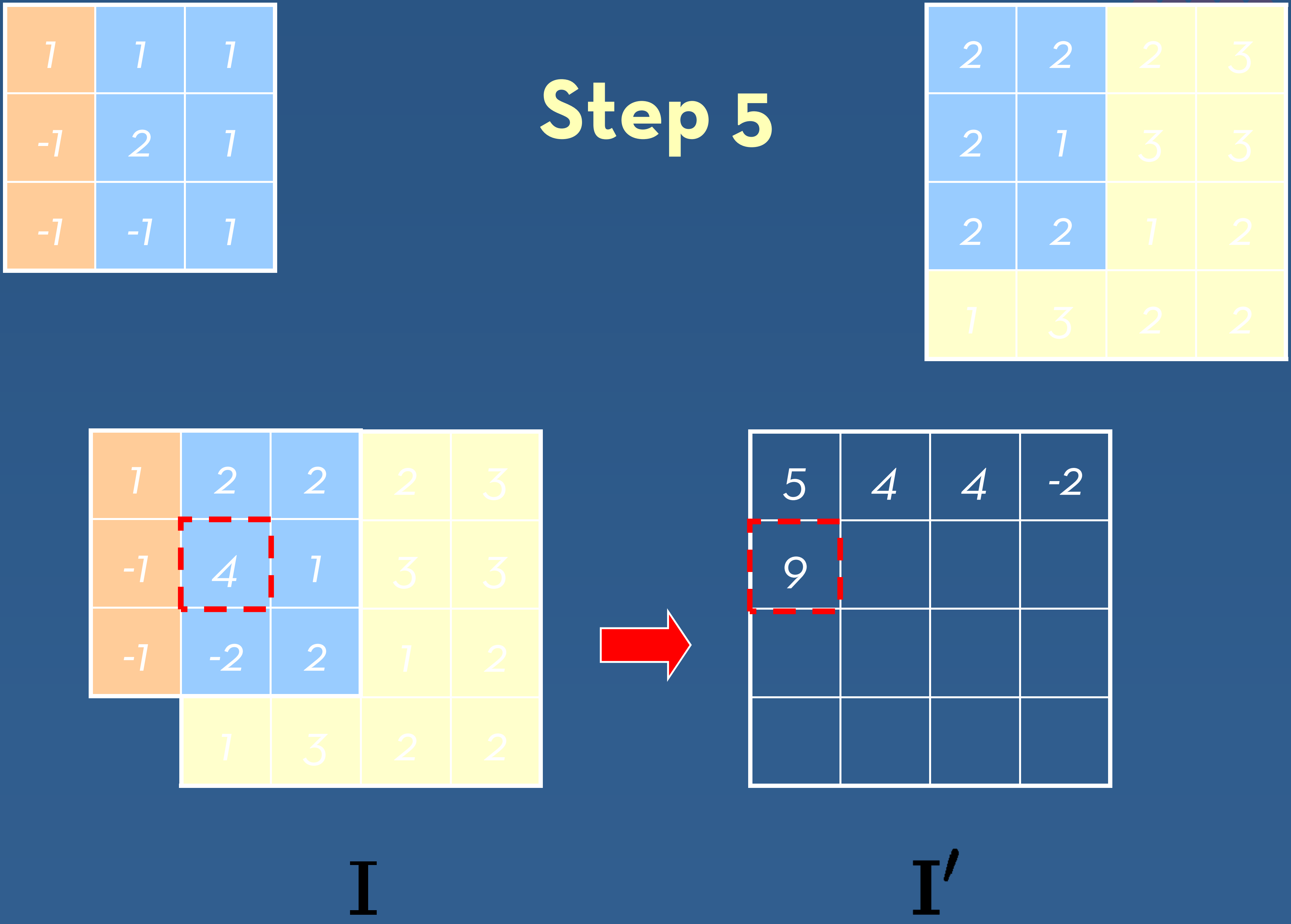
(courtesy of Christopher Rasmussen)

Step 4



(courtesy of Christopher Rasmussen)

Step 5



$$\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline -1 & 2 & 1 \\ \hline -1 & -1 & 1 \\ \hline \end{array} \qquad \begin{array}{|c|c|c|c|}\hline 2 & 2 & 2 & 3 \\ \hline 2 & 1 & 3 & 3 \\ \hline 2 & 2 & 1 & 2 \\ \hline 1 & 3 & 2 & 2 \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|c|}\hline 1 & 2 & 2 & 2 & 3 \\ \hline -1 & \boxed{4} & 1 & 3 & 3 \\ \hline -1 & -2 & 2 & 1 & 2 \\ \hline & 1 & 3 & 2 & 2 \\ \hline \end{array} \xrightarrow{\quad} \begin{array}{|c|c|c|c|}\hline 5 & 4 & 4 & -2 \\ \hline \boxed{9} & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}$$

(courtesy of Christopher Rasmussen)

Step 6

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

2	2	2	3
-2	2	3	3
-2	-2	1	2
1	3	2	2

$\mathbf{I}$

5	4	4	-2
9	6		

$\mathbf{I}'$

Final Result

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

I

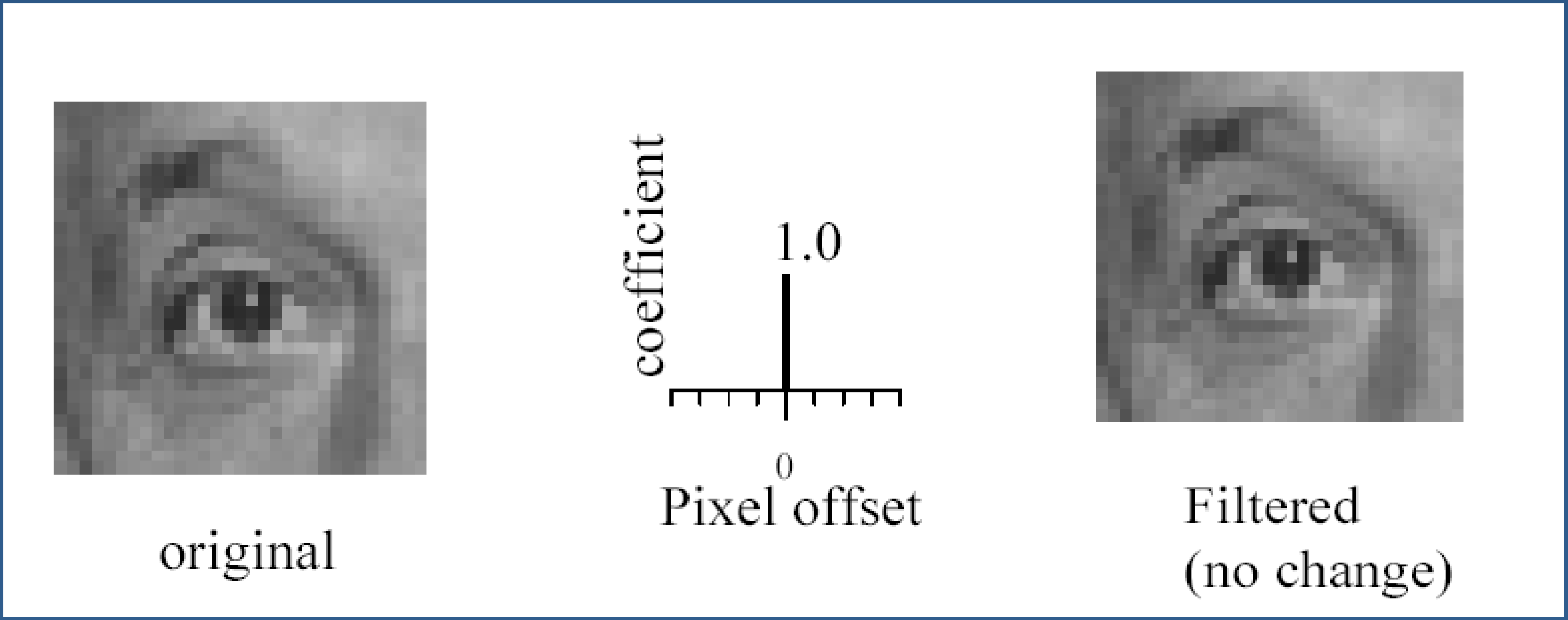
5	4	4	-2
9	6	14	5
11	7	6	5
9	12	8	5

I'

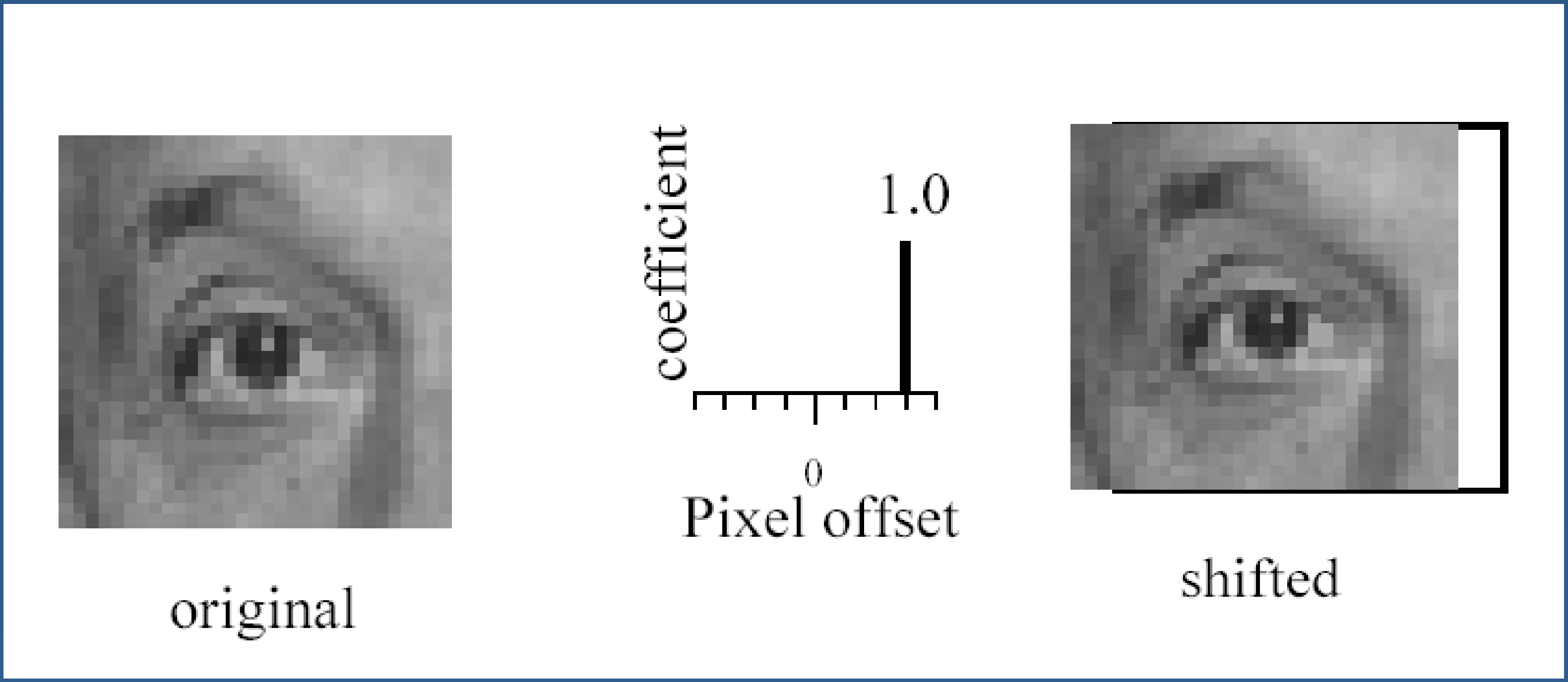
*Why is **I'** large in some places and small in others?*

(courtesy of Christopher Rasmussen)

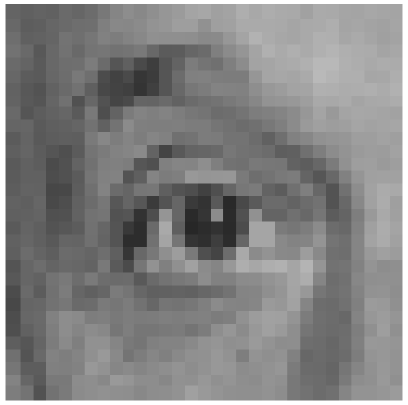
Filtering Example

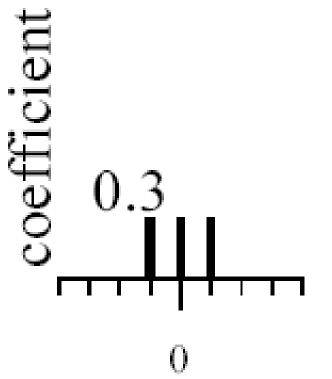


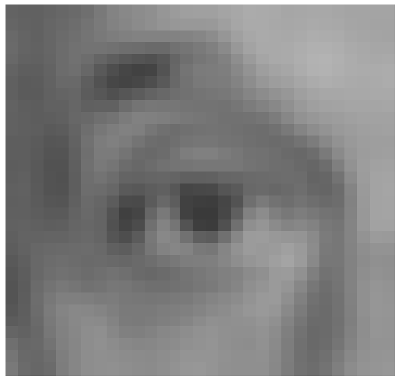
Filtering Example



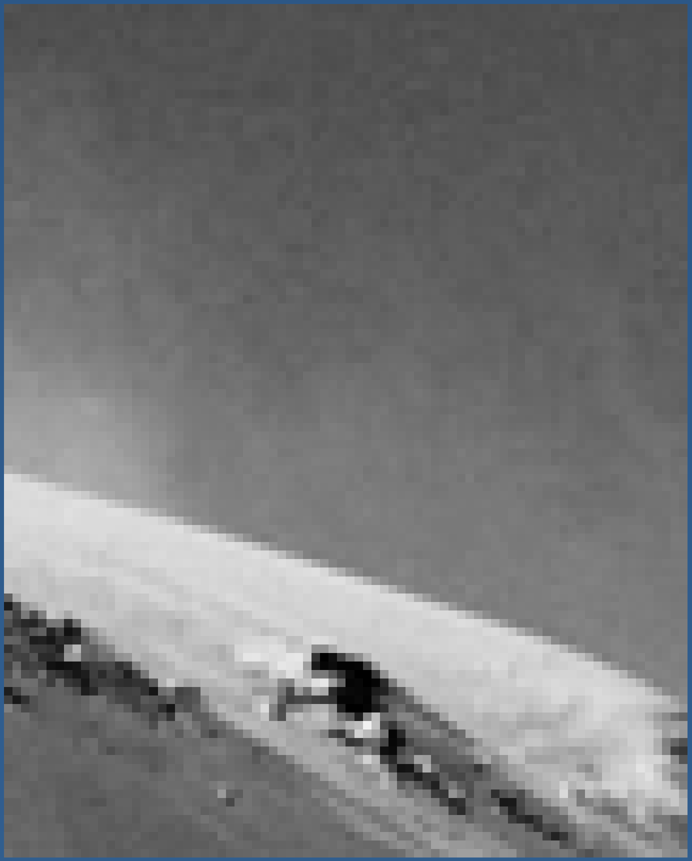
Filtering Example

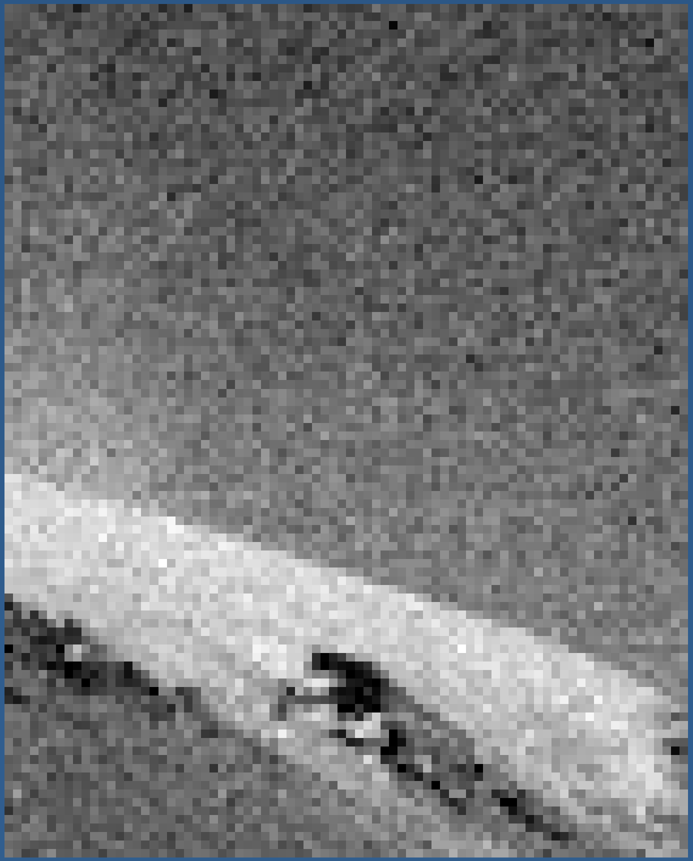

original




Blurred (filter applied in both dimensions).

Problem: Image Noise

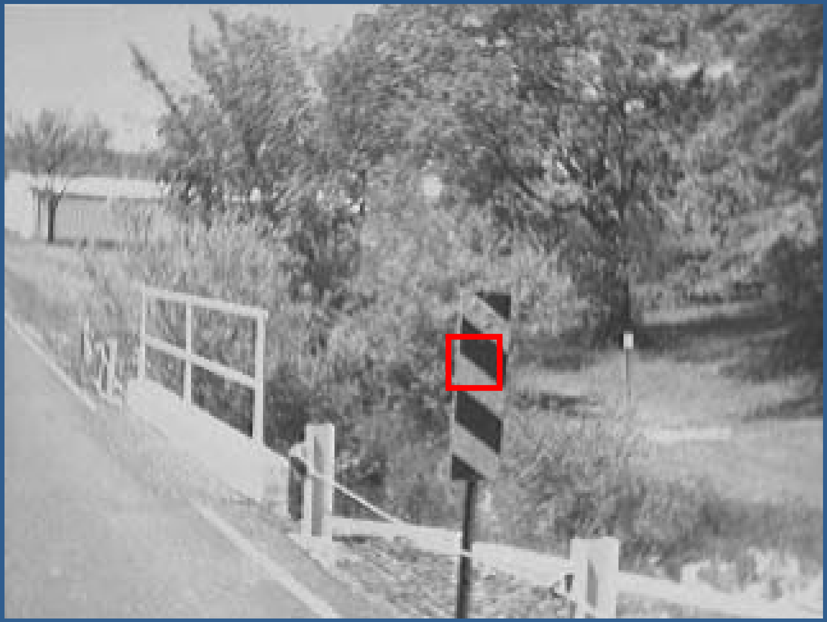


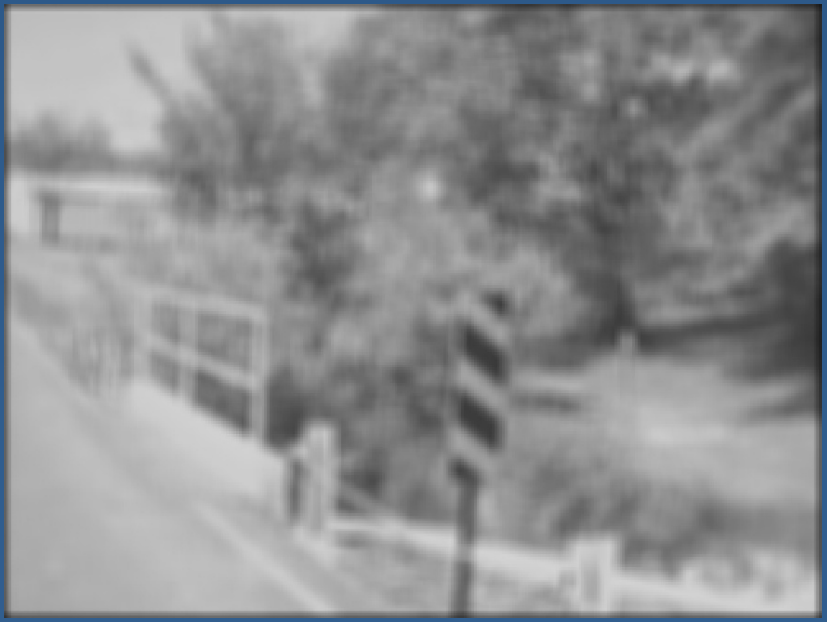


(courtesy of Forsyth & Ponce)

Solution: Smoothing Filter

- If object reflectance changes slowly and noise at each pixel is independent, then we want to replace each pixel with something like the average of neighbors
  Disadvantage: Sharp (high-frequency) features lost


Original image


7 x 7 averaging neighborhood

Smoothing Filter: Details

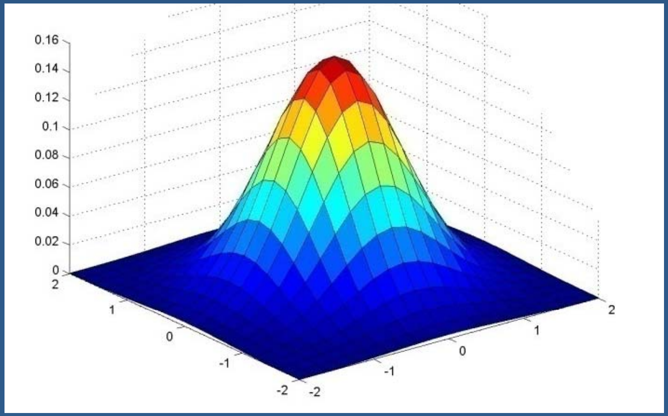
- Filter types
  - Mean filter (box)
  - Median (nonlinear)
  - Gaussian
- Can specify linear operation by shifting kernel over image and taking product

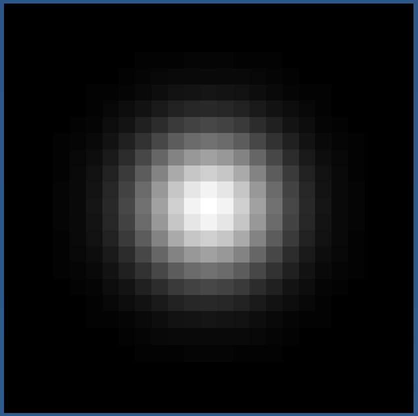
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

3 x 3 box filter kernel

Gaussian Kernel

- Idea: Weight contributions of neighboring pixels by nearness





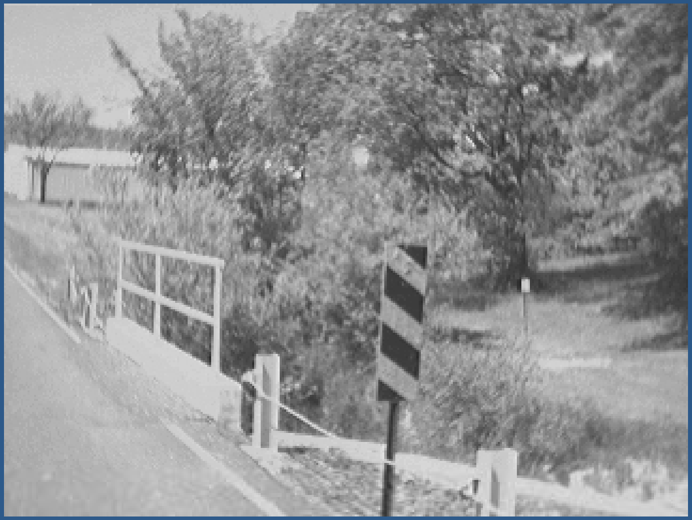
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

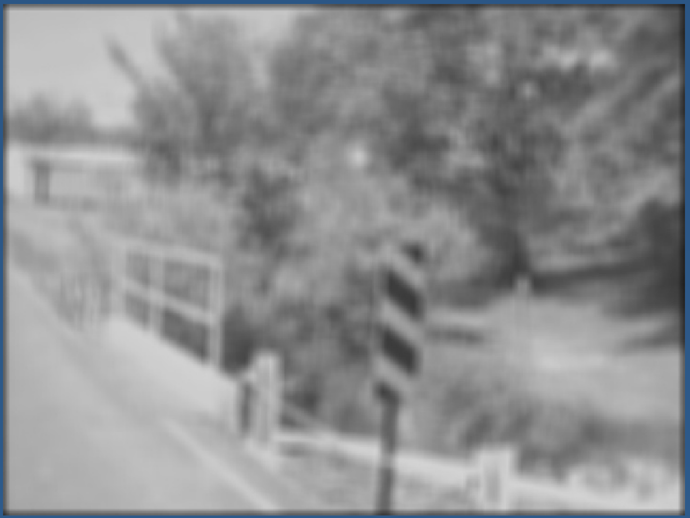
5 x 5, $\sigma = 1$

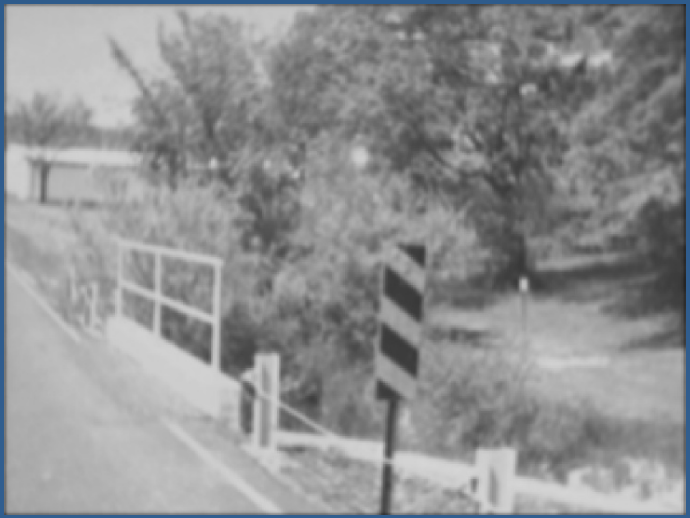
$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

- Smooth roll-off reduces “ringing” seen in box filter

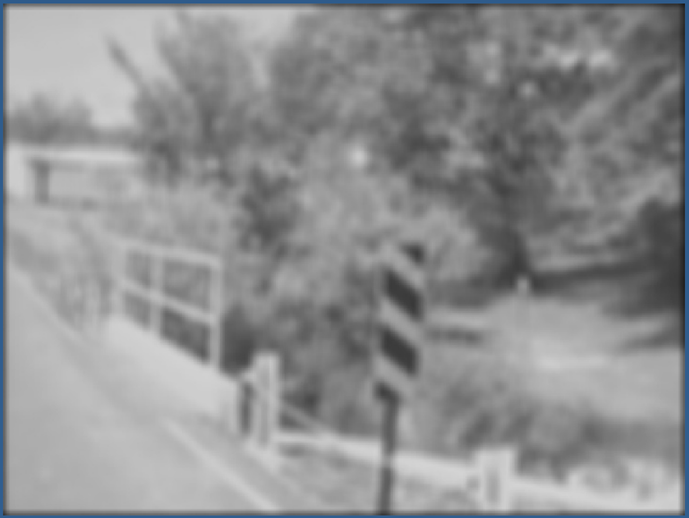
Gaussian Smoothing Example


Original image

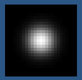

Box filter

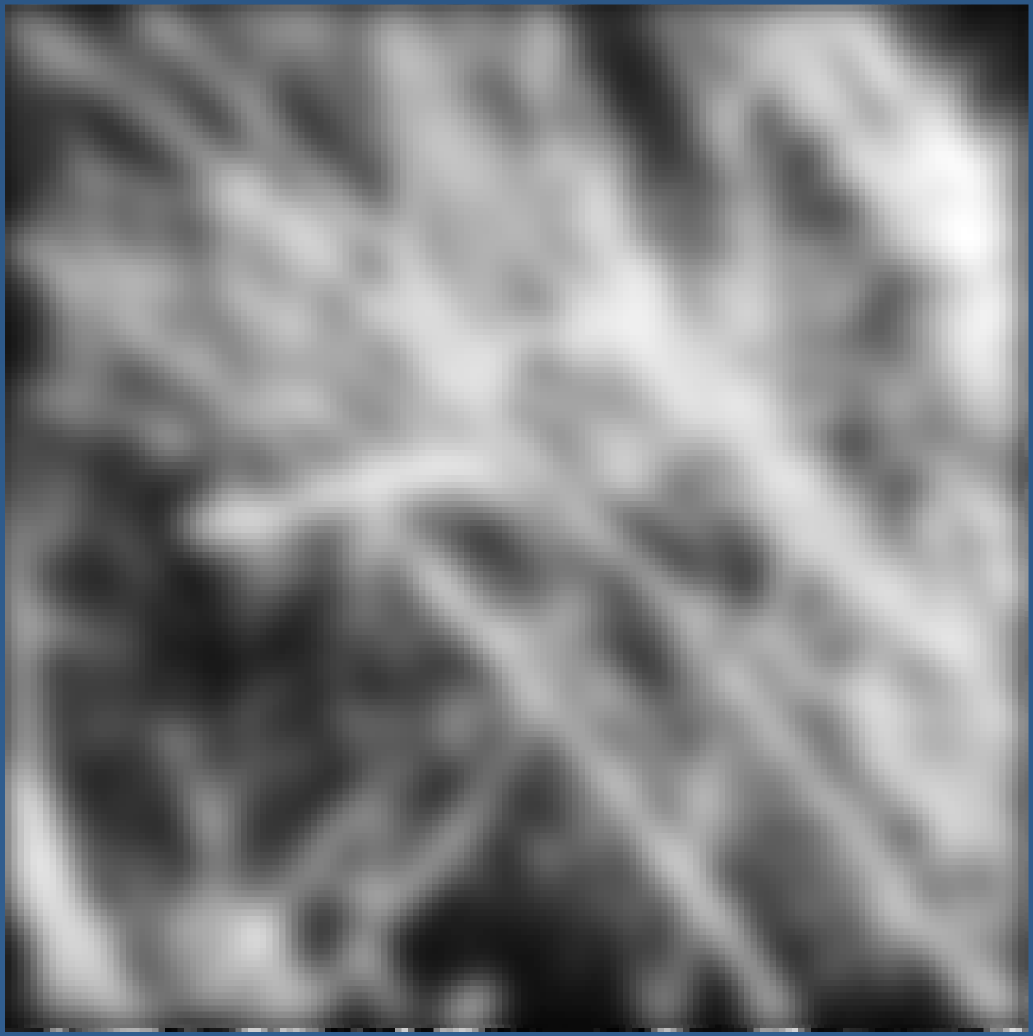

σ = 1

7 x 7 kernel


σ = 3

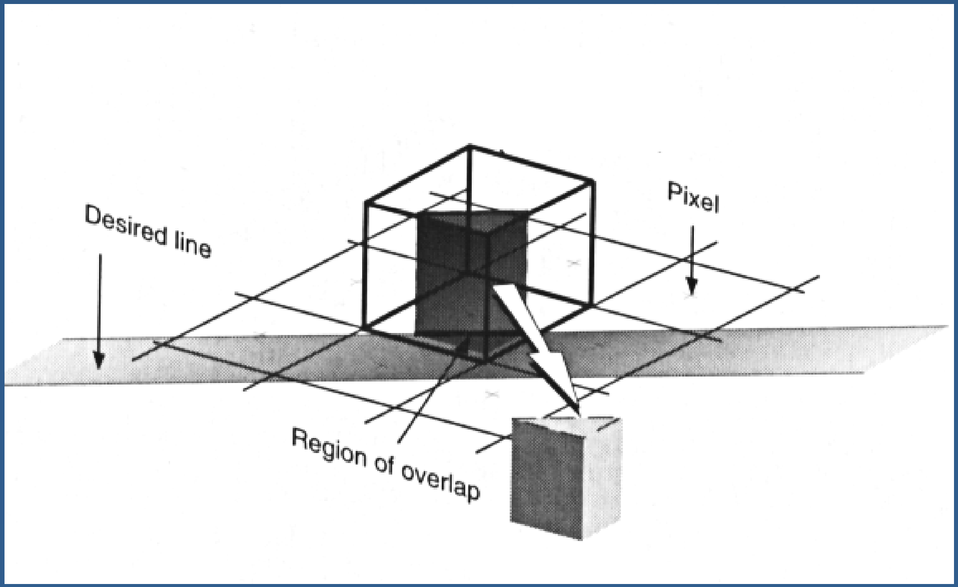
Gaussian Smoothing

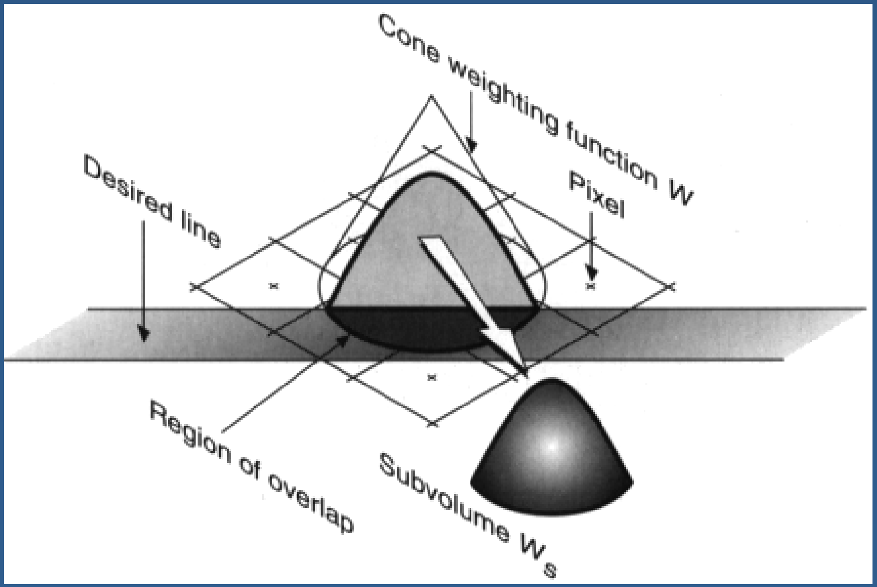
Averaging  *Gaussian*



(courtesy of Marc Pollefeys)

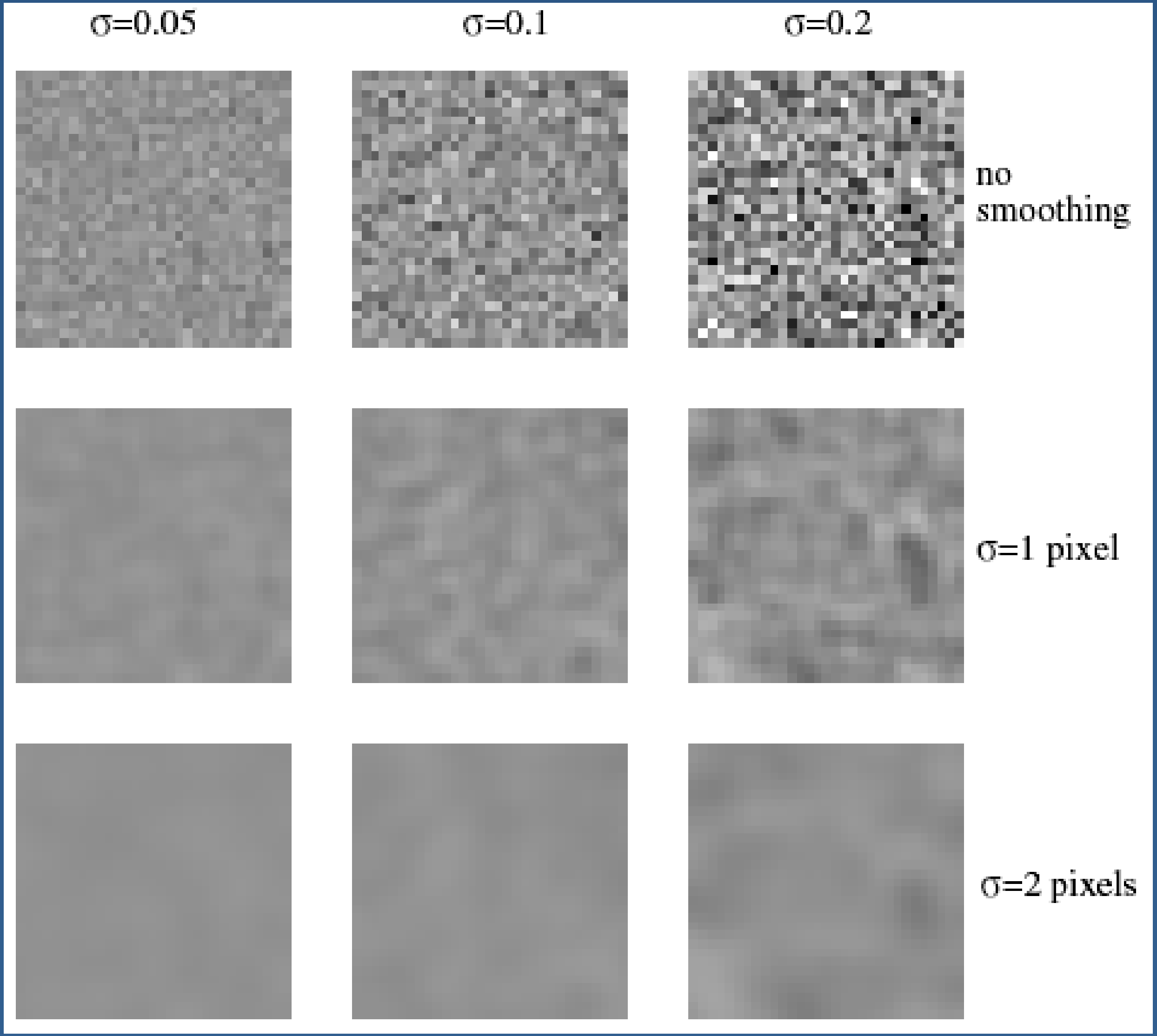
“Box” vs. “Cone” Filters





(courtesy Andries van Dam)

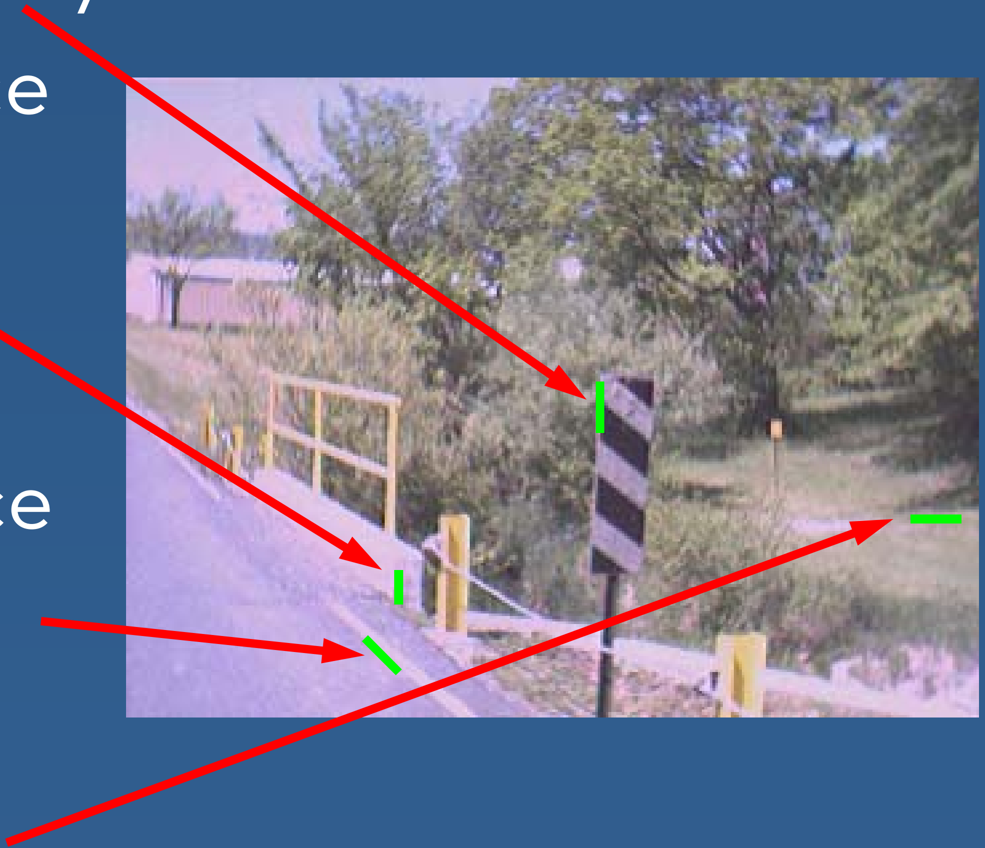
Smoothing Reduces Noise



(courtesy of Marc Pollefeys)

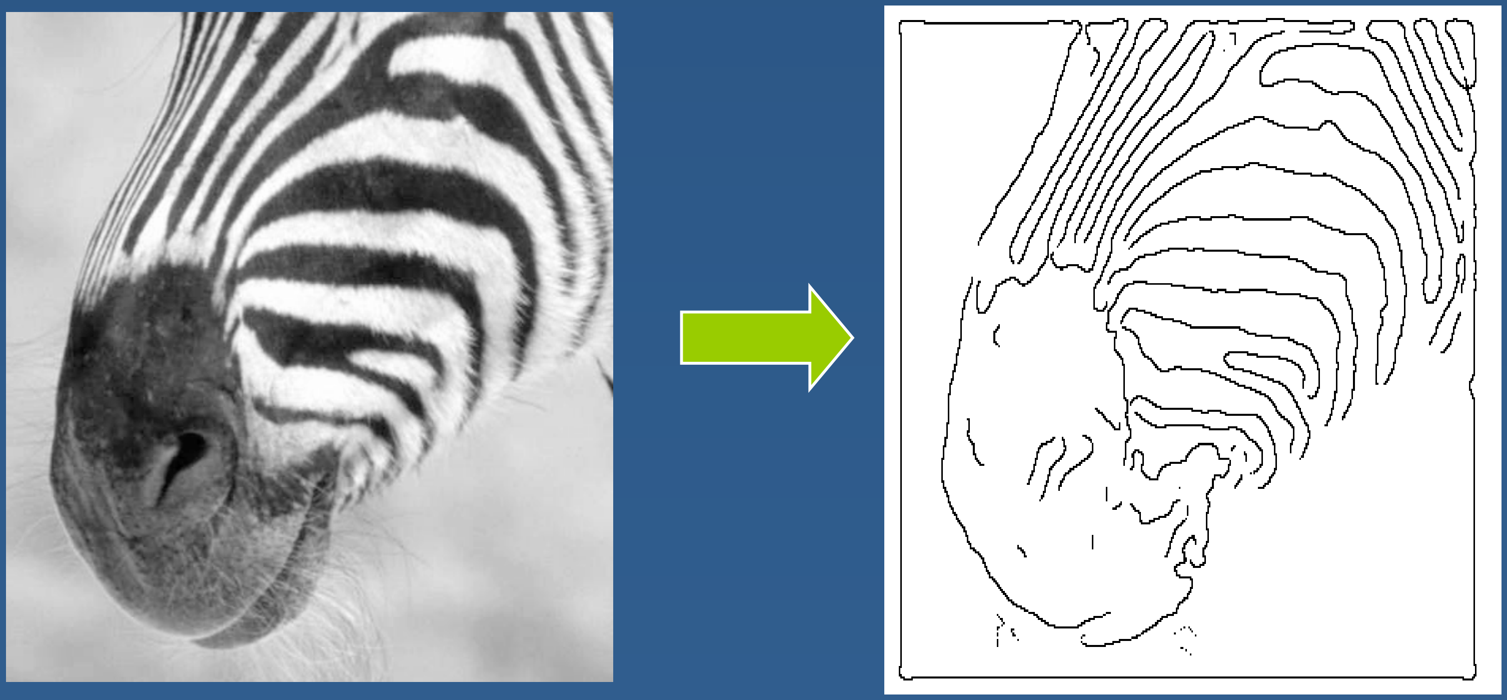
What causes an edge?

- Depth discontinuity
- Change in surface orientation
- Reflectance discontinuity (change in surface properties)
- Illumination discontinuity (light/shadow)

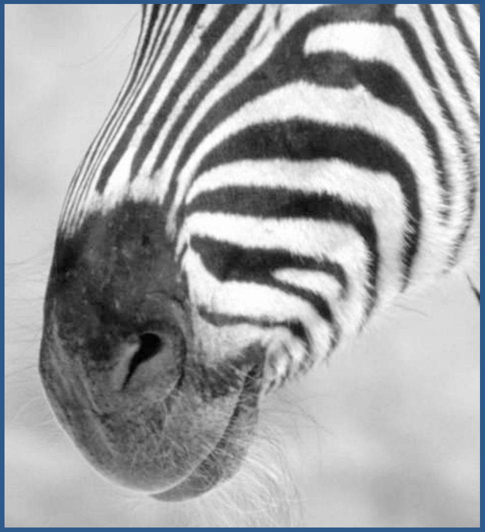


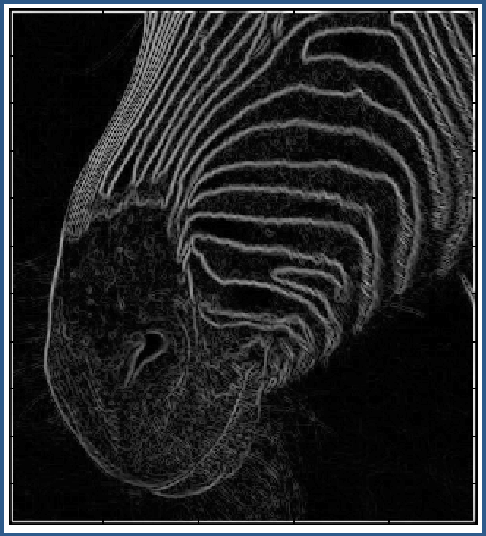
(courtesy of Christopher Rasmussen)

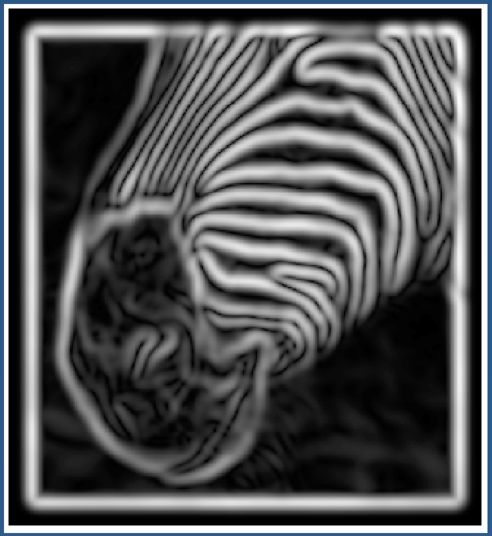
How Can We Find Edges?



Spatial Structure of Edges

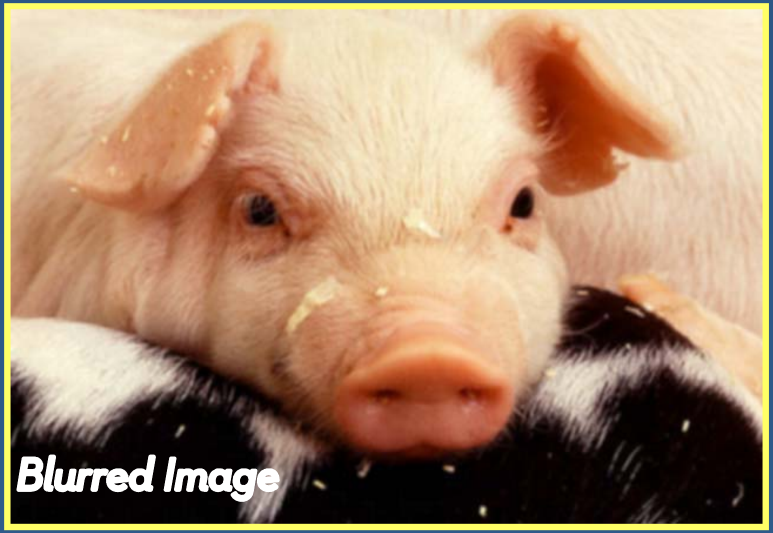


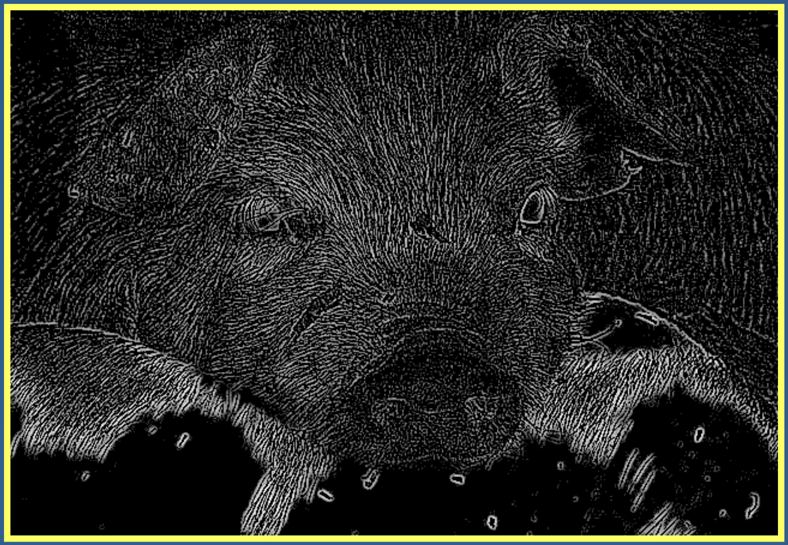




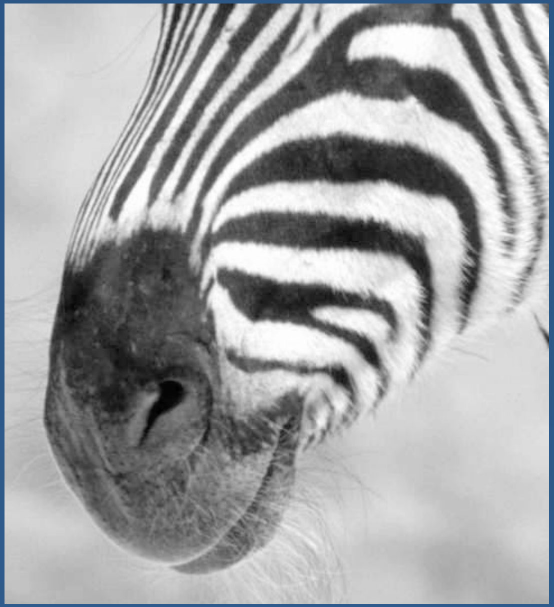
- Edges exist at different scales from fine (whiskers) to coarse (stripes)
- Solution: Gaussian smoothing

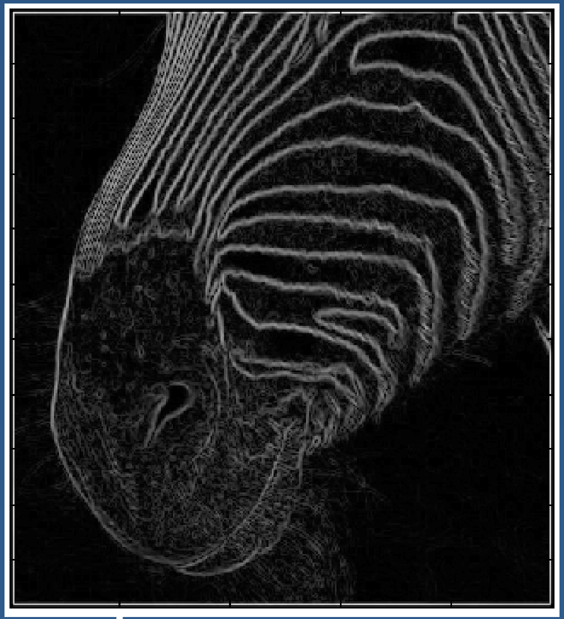
Blurring Example

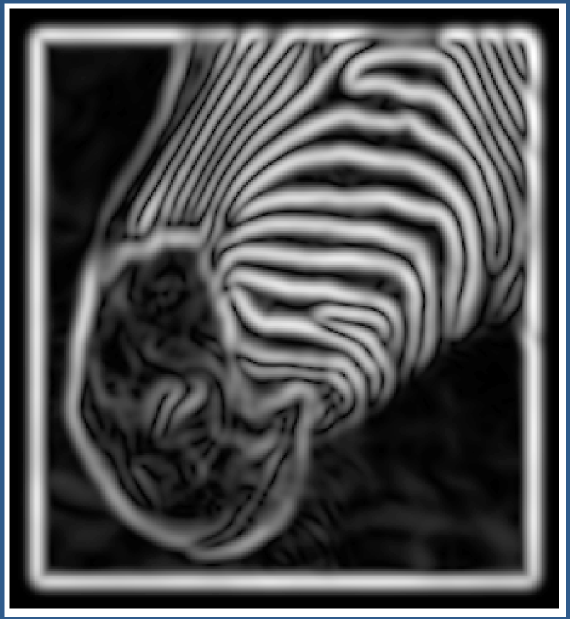




Scale

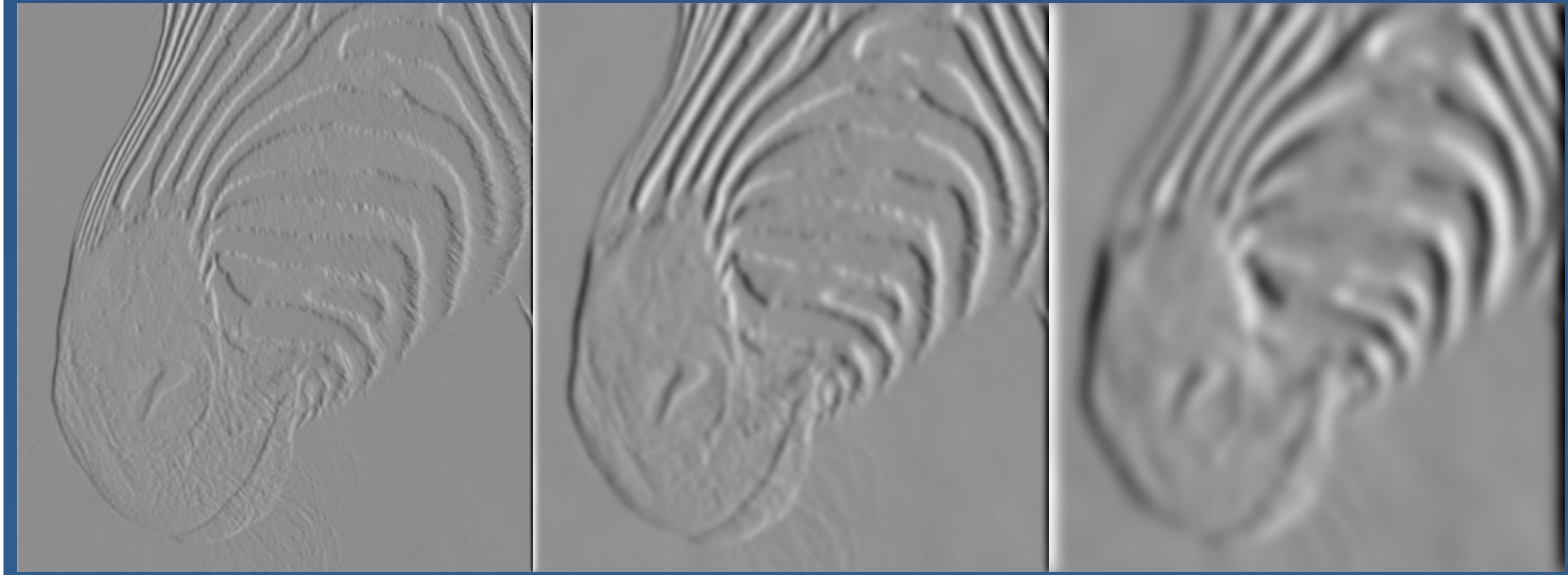






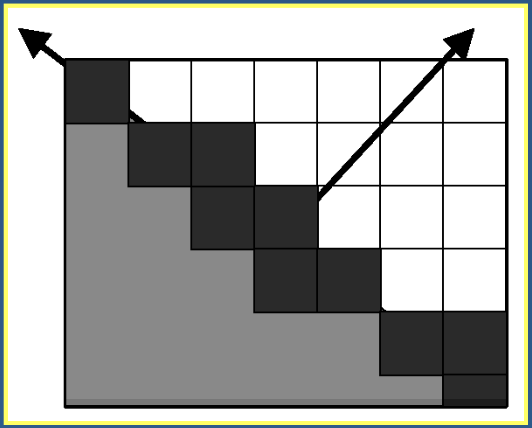
- Increasing smoothing:
  - Eliminates noise edges
  - Makes edges smoother and thicker
  - Removes fine detail

Effect of Smoothing Radius



1 pixel **_3 pixels_** **_7 pixels_**

The Edge Normal



$$S = \sqrt{dx^2 + dy^2}$$

$$\alpha = \arctan \frac{dy}{dx}$$

Sobel Operator

$$S_1 = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \qquad S_2 = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

$$\textit{Edge Magnitude} = \sqrt{S_1^2 + S_1^{\ \ 2}}$$

$$\textit{Edge Direction} = \tan^{-1}\left(\frac{S_1}{S_2}\right)$$

The Sobel Kernel, Explained

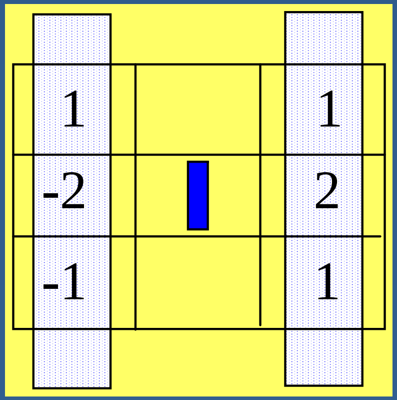
-1	0	1
-2	0	2
-1	0	1

$$= 1/4 * [\ 1\ \ 0\ \ -1] \circledast \begin{bmatrix}1\\2\\1\end{bmatrix}$$

1	2	1
0	0	0
-1	-2	-1

$$= 1/4 * [\ 1\ \ 2\ \ 1] \circledast \begin{bmatrix}1\\0\\-1\end{bmatrix}$$

Sobel kernel is separable!

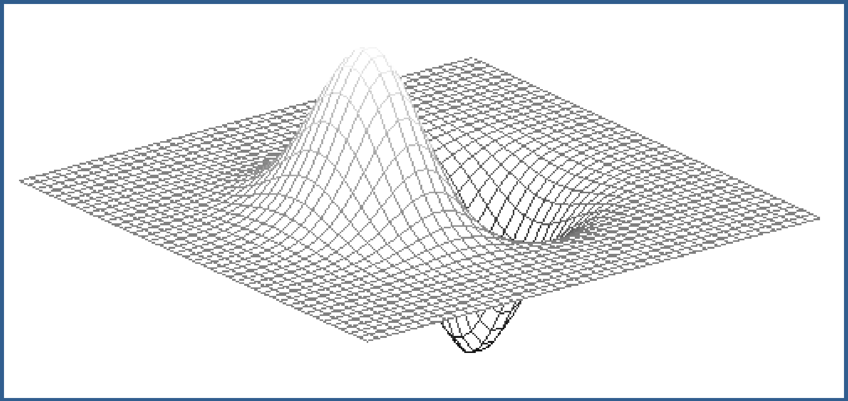


Averaging done parallel to edge

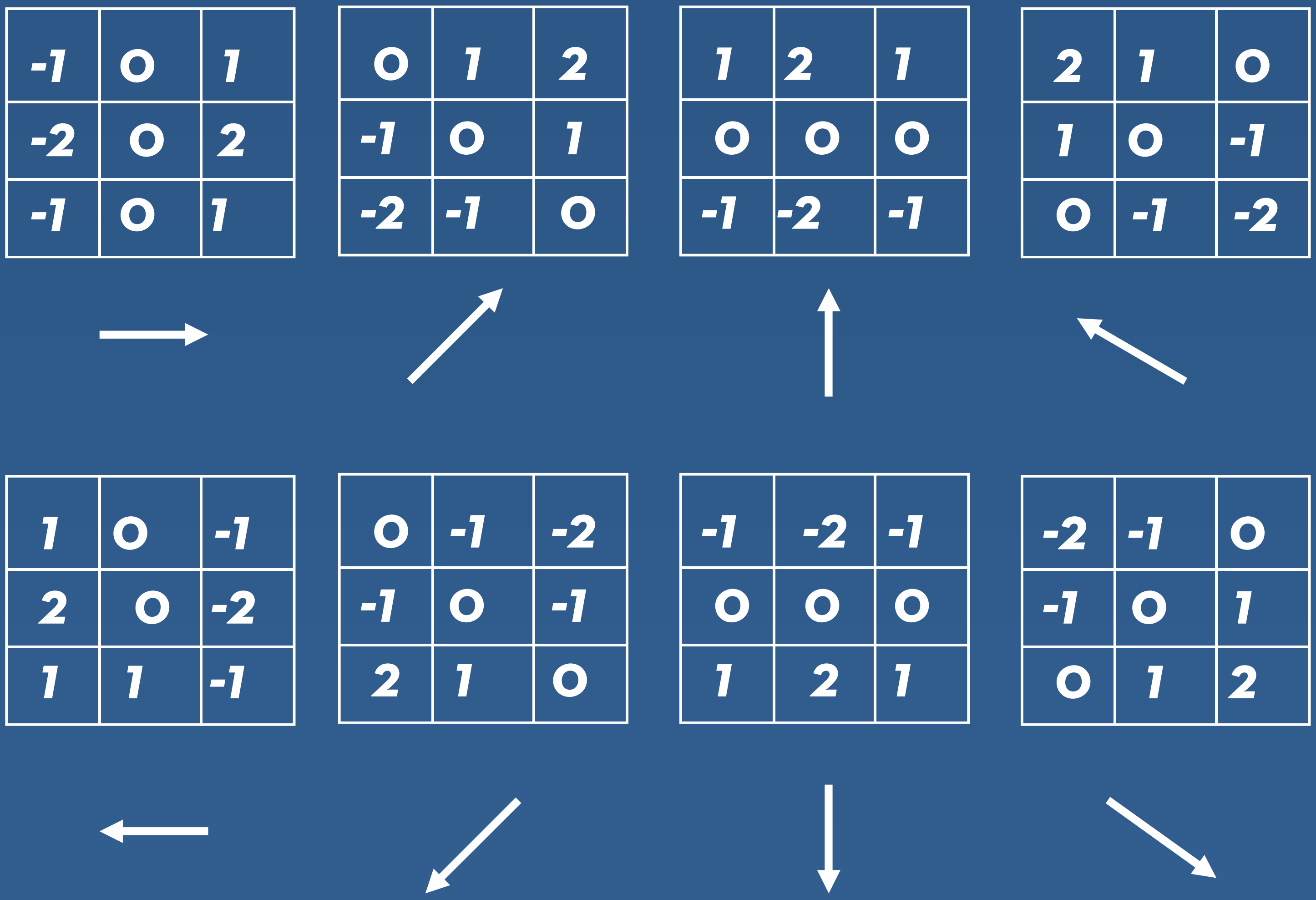
Combining Kernels

$$(I \otimes g) \otimes h = I \otimes (g \otimes h)$$

$$\begin{bmatrix} 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0219 & 0.0983 & 0.1621 & 0.0983 & 0.0219 \\ 0.0133 & 0.0596 & 0.0983 & 0.0596 & 0.0133 \\ 0.0030 & 0.0133 & 0.0219 & 0.0133 & 0.0030 \end{bmatrix} \otimes \begin{bmatrix} 1 & -1 \end{bmatrix}$$



Robinson Compass Masks



-1	0	1
-2	0	2
-1	0	1

$\rightarrow$

0	1	2
-1	0	1
-2	-1	0

$\nearrow$

1	2	1
0	0	0
-1	-2	-1

$\uparrow$

2	1	0
1	0	-1
0	-1	-2

$\nwarrow$

1	0	-1
2	0	-2
1	1	-1

$\leftarrow$

0	-1	-2
-1	0	-1
2	1	0

$\swarrow$

-1	-2	-1
0	0	0
1	2	1

$\downarrow$

-2	-1	0
-1	0	1
0	1	2

$\searrow$

Robert's Cross Operator

$$\left| \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \right| + \left| \begin{array}{|c|c|} \hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array} \right|$$

$$S = \sqrt{[\,I(x, y) - I(x+1, y+1)\,]^2 + [\,I(x, y+1) - I(x+1, y)\,]^2}$$

or

$$S = |\,I(x, y) - I(x+1, y+1)\,| + |\,I(x, y+1) - I(x+1, y)\,|$$

Claim Your Own Kernel!

1	1	1
1	-2	1
-1	-1	-1

Prewitt 1

5	5	5
-3	0	-3
-3	-3	-3

Kirsch

-1	$\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

Frei & Chen

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

Prewitt 2

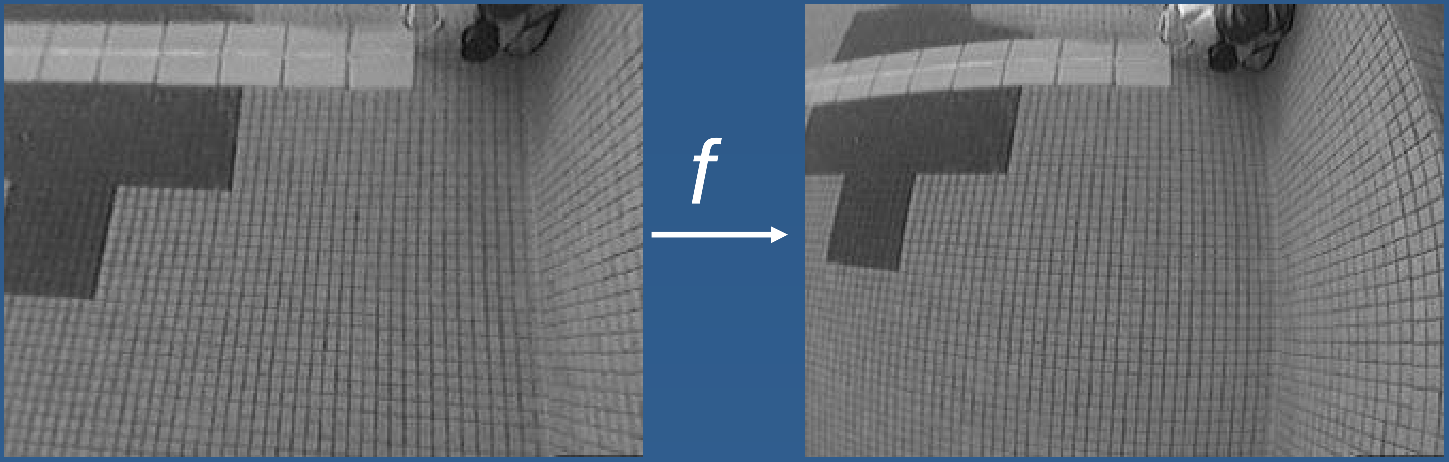
1	2	1
0	0	0
-1	-2	-1

Sobel

(courtesy of Sebastian Thrun)

What’s Not a Convolution?

- Nonlinear systems
  - For example, radial distortion of a fish-eye lens

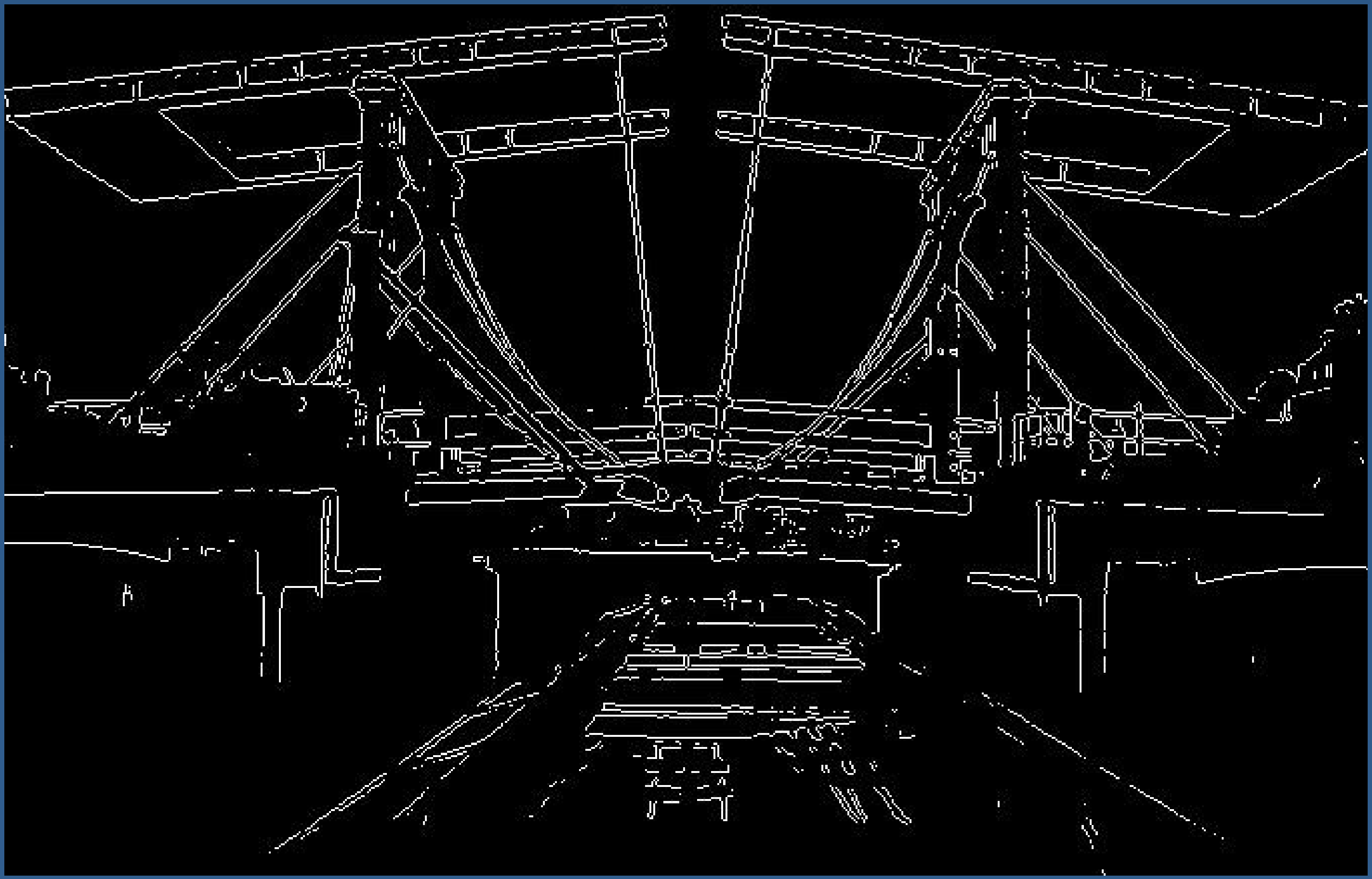


(courtesy of M. Fiala)

John Canny’s Edge Detector

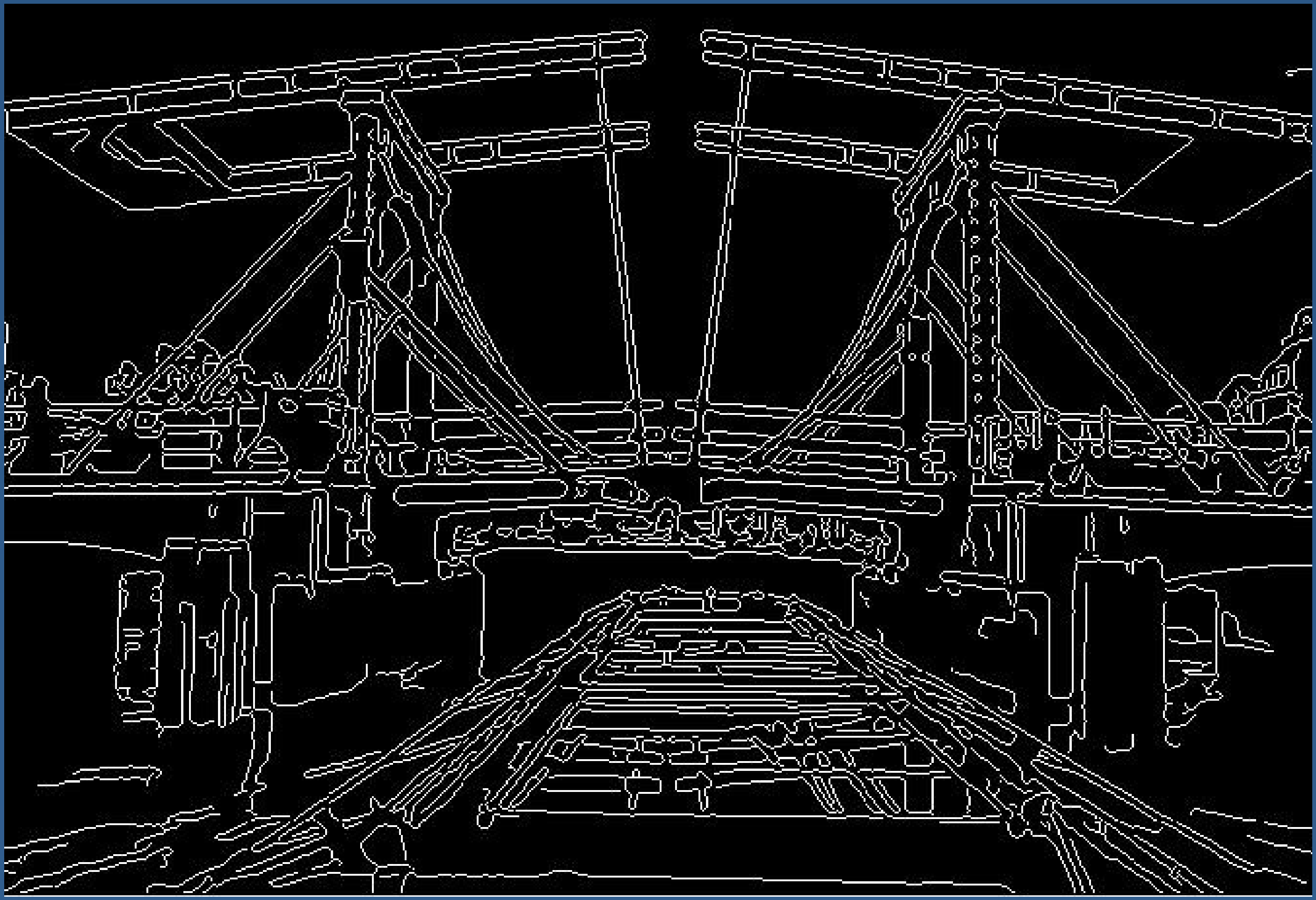
- John Canny, “Finding Edges and Lines in Images,” Master’s Thesis, MIT, June 1983.
- Developed a practical edge detection algorithm with three properties
  - Gaussian smoothing
  - Non-maximum suppression (remove edges orthogonal to a maxima)
  - Hysteresis thresholding – Improved recovery of long image contours

Sobel Example



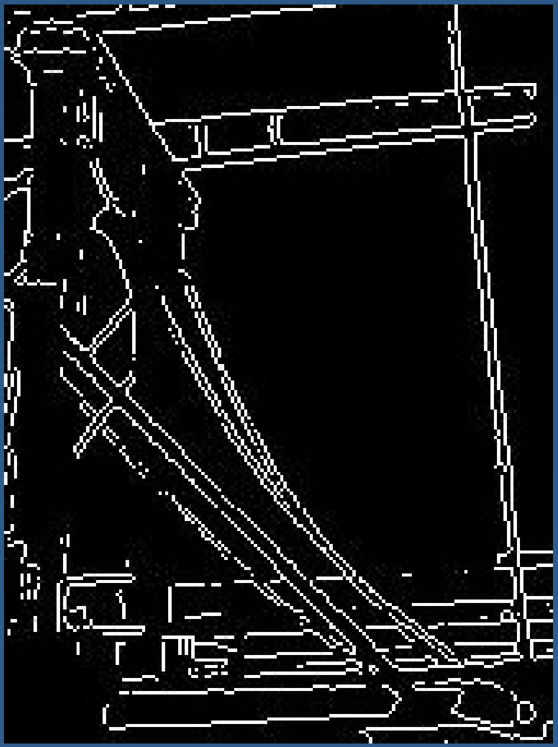
(courtesy of Sebastian Thrun)

Canny Example

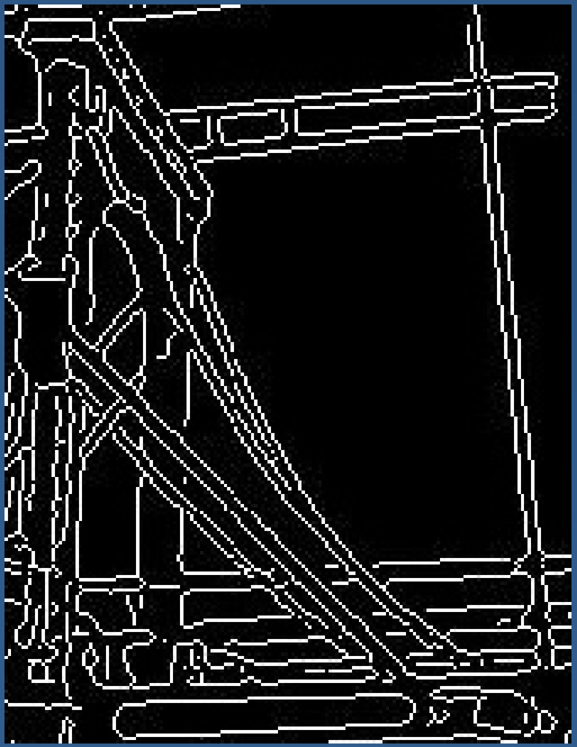


(courtesy of Sebastian Thrun)

Comparison



Sobel



Canny

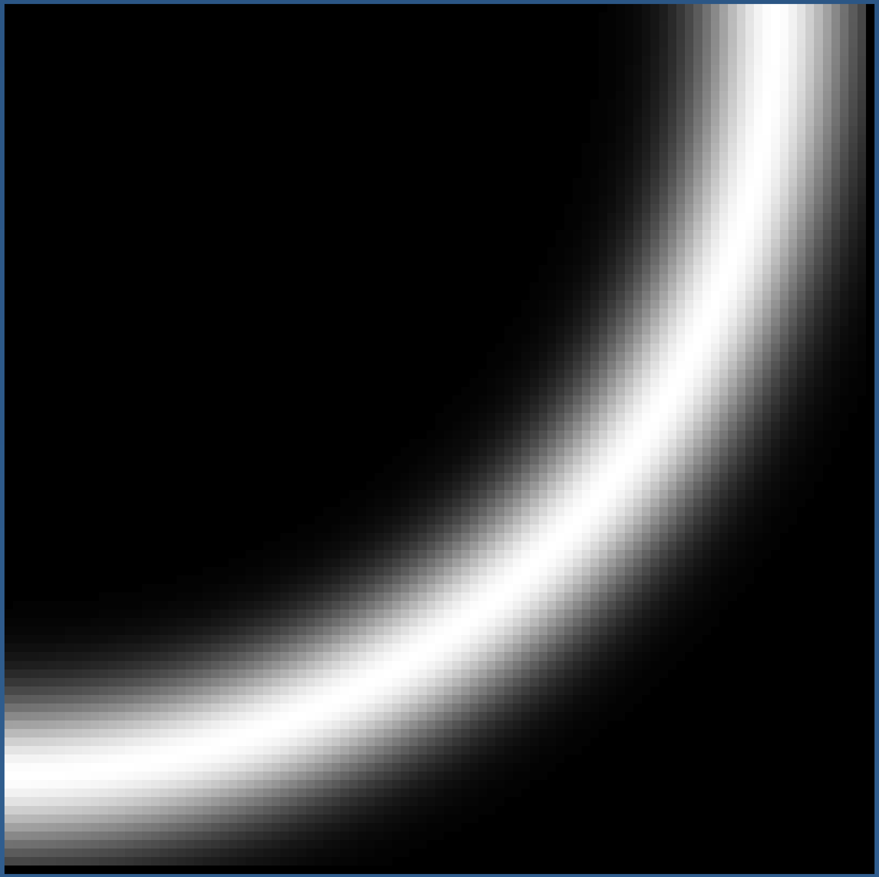
(courtesy of Sebastian Thrun)

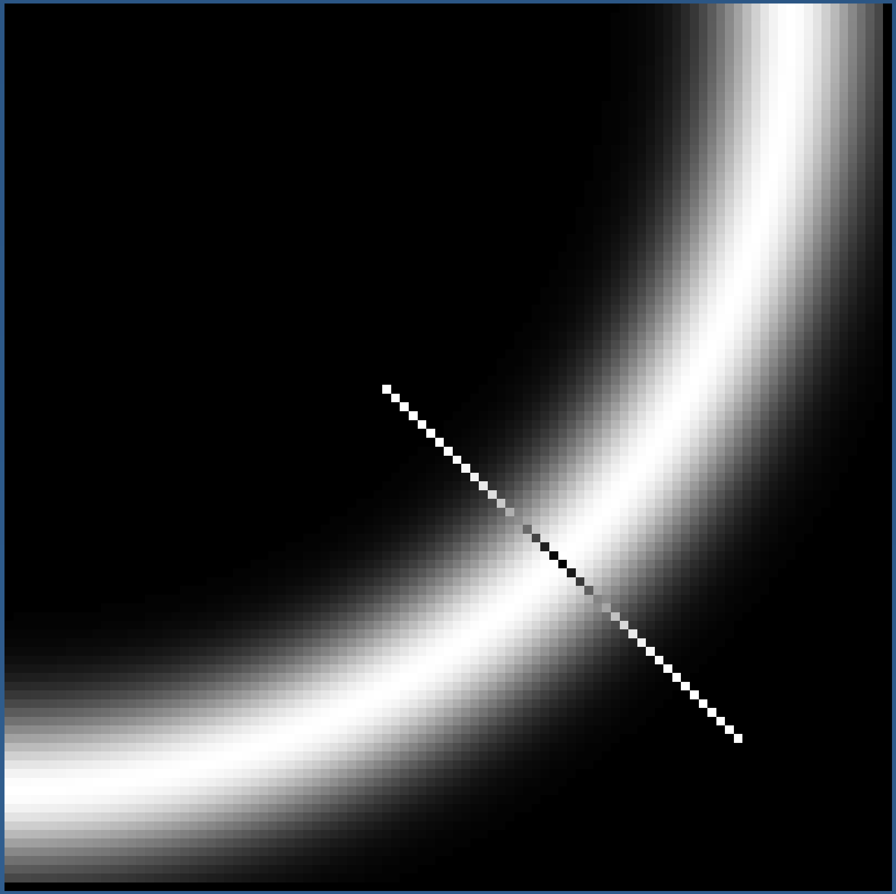
Canny Edge Detection

Steps:

1. Apply derivative of Gaussian
2. Non-maximum suppression
   - Thin multi-pixel wide “ridges” down to single pixel width
3. Linking and thresholding
   - Low, high edge-strength thresholds
   - Accept all edges over low threshold that are connected to edge over high threshold

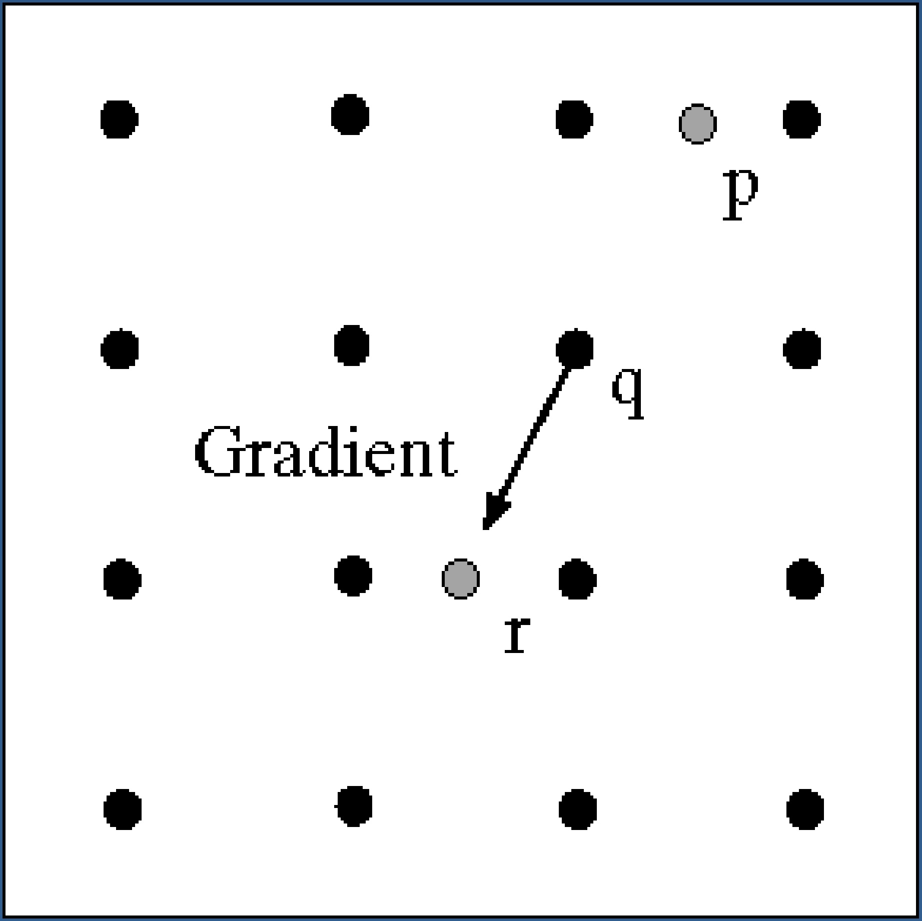
Non-Maximum Suppression





- Select the single maximum point across the width of an edge.

Linking to the Next Edge Point



- Assume the marked point q is an edge point.
- Take the normal to the gradient at that point and use this to predict continuation points (either r or p).

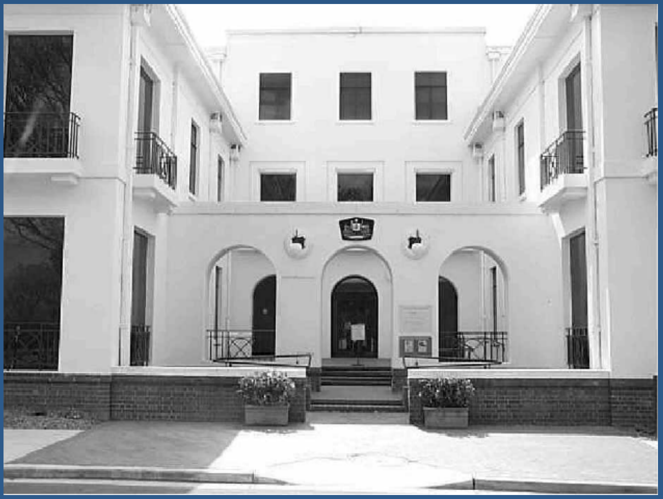
Edge Hysteresis

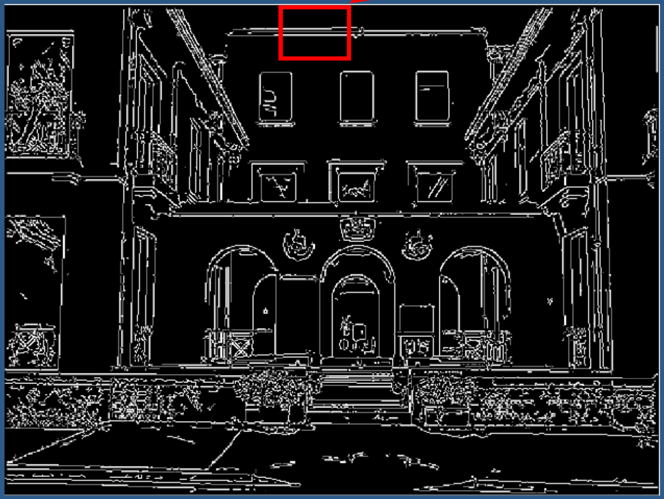
- Hysteresis: A lag or momentum factor
- Idea: Maintain two thresholds: k_{HIGH} and k_{LOW}
- Use k_{HIGH} to find strong edges to start edge chain
- Use k_{LOW} to find weak edges which continue edge chain
- Typical ratio of thresholds is roughly $k_{\mathrm{HIGH}} / k_{\mathrm{LOW}} = 2$

Canny Edge Detection (Example)

gap is gone

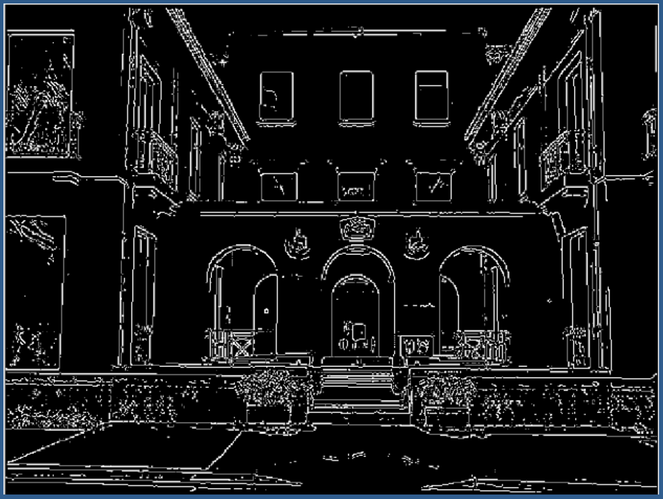
Original image

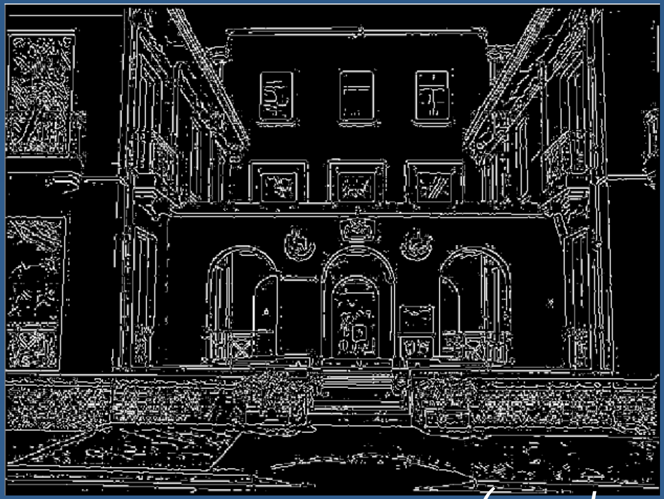




Strong + connected weak edges

Strong edges only

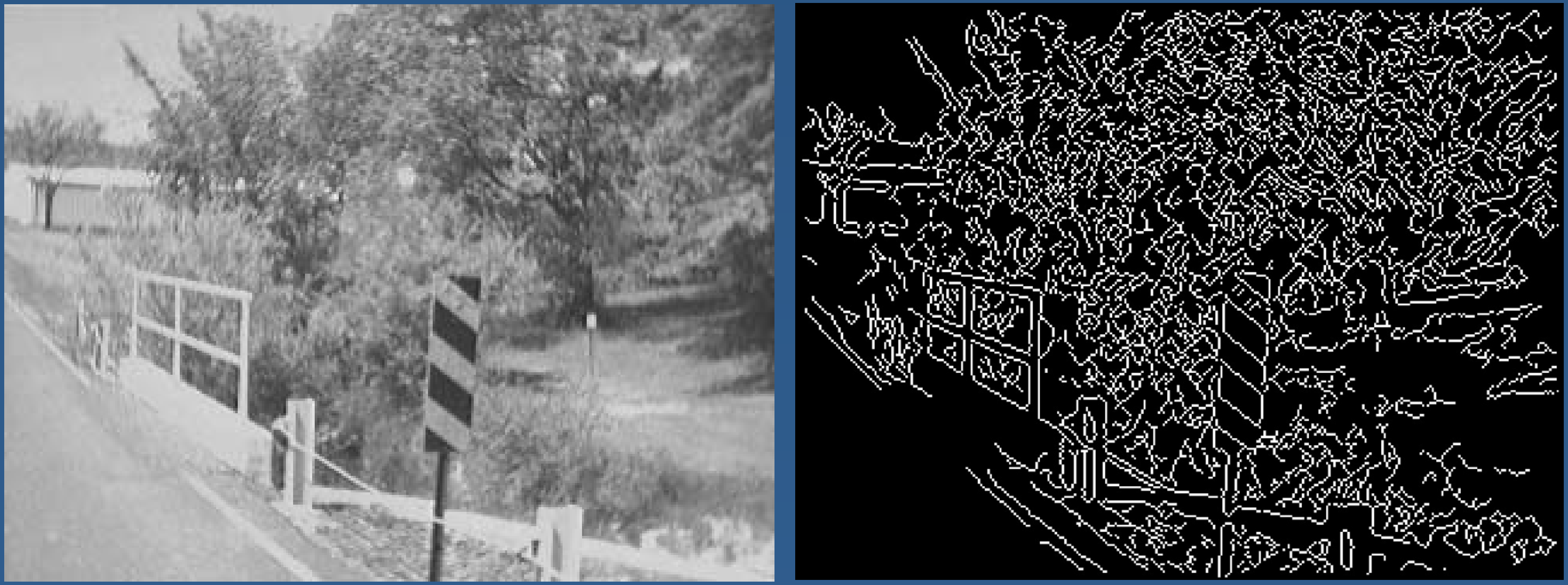




Weak edges

(courtesy of G. Loy)

Canny Edge Detection (Example)

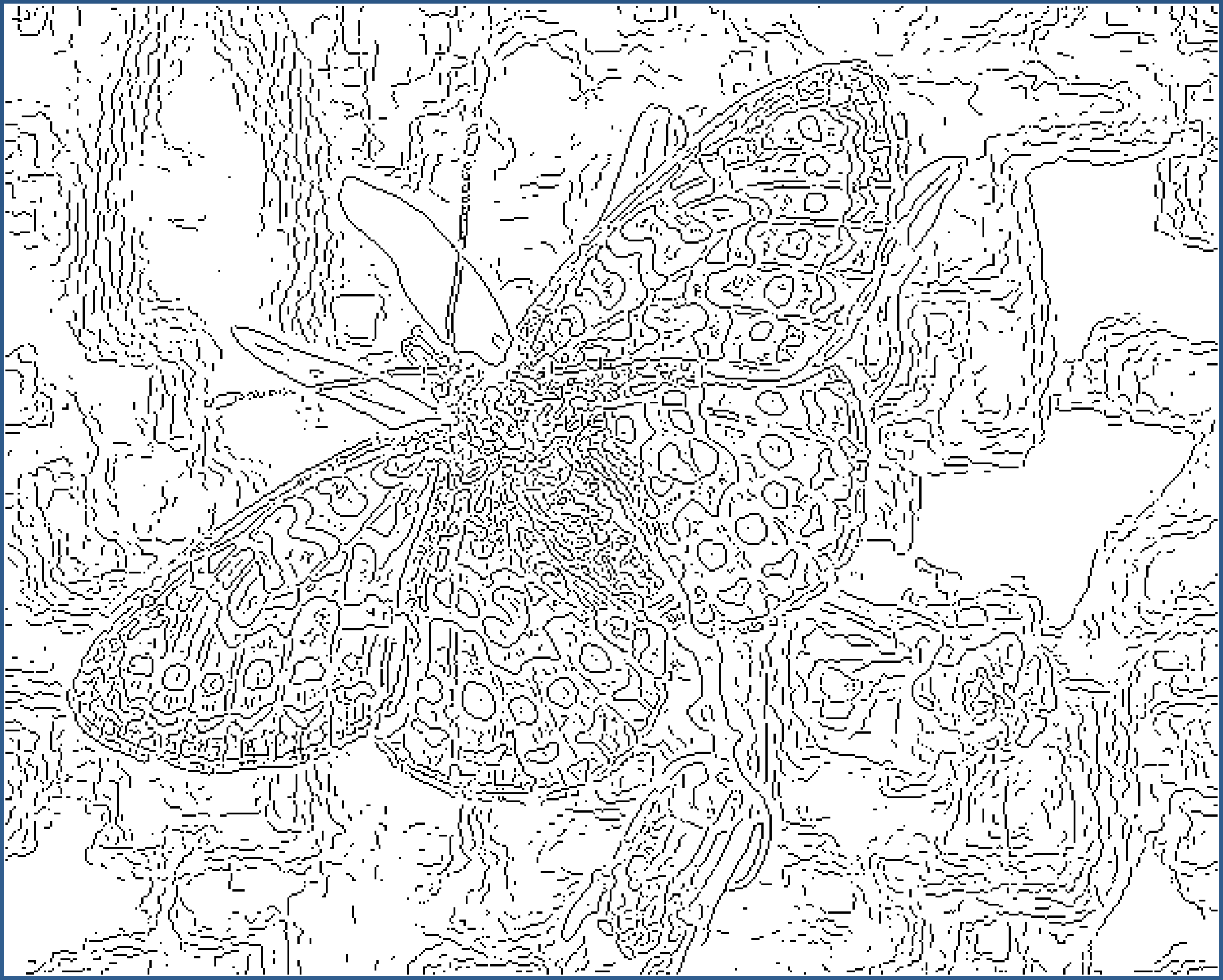


Using default thresholds in Matlab

Choosing Parameters

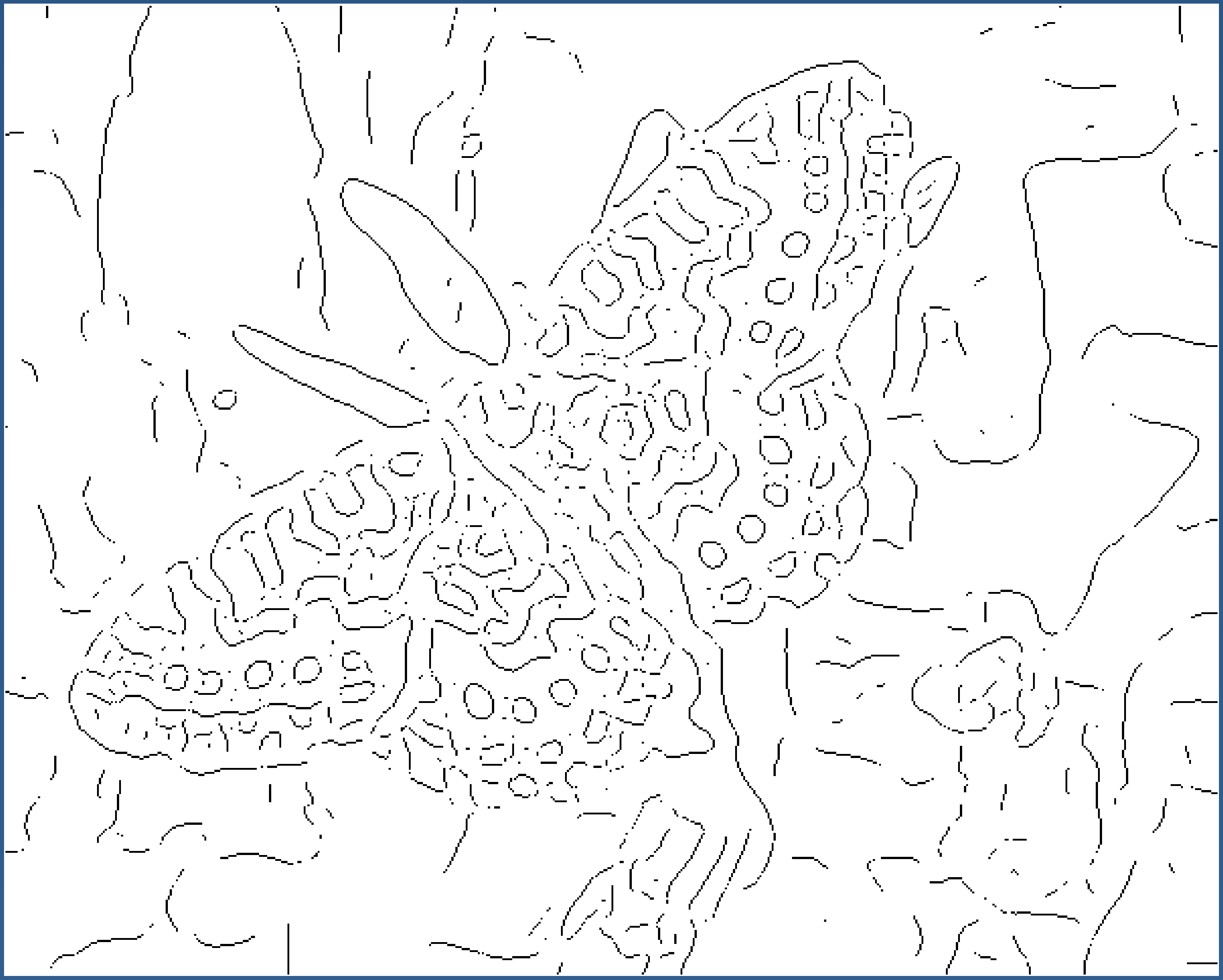
Choosing Parameters

Fine scale,
High threshold



Choosing Parameters

Coarse scale,
High threshold



Choosing Parameters

Coarse scale,
Low threshold

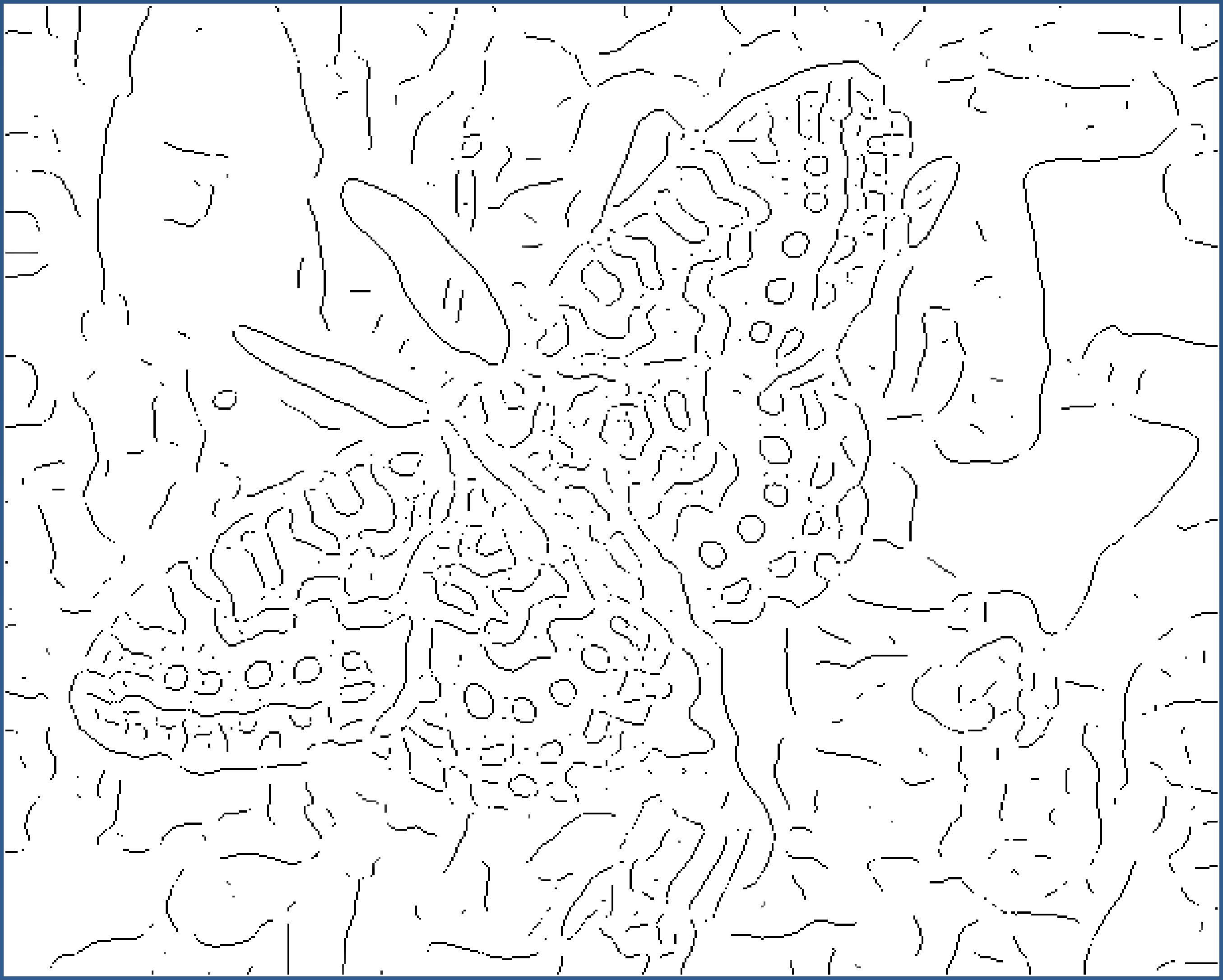
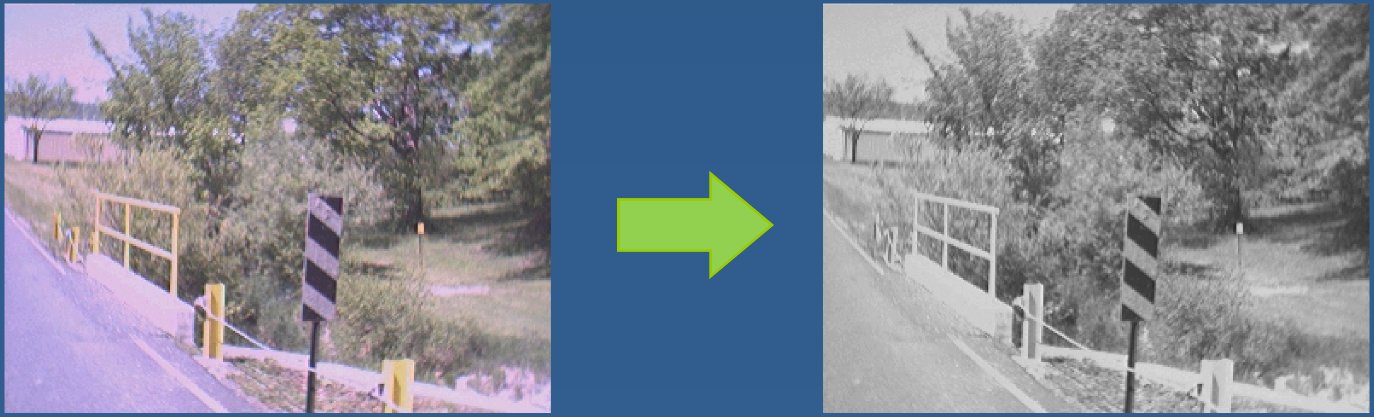


Image Arithmetic

- Just like matrices, we can do pixelwise arithmetic on images
- Some useful operations
  - Differencing: Measure of similarity
  - Averaging: Blend separate images or smooth a single one over time
  - Thresholding: Apply function to each pixel, test value

Image Type Conversion

- Many processing functions work on just one channel, so the options are:
  - Run on each channel independently
  - Convert from color → grayscale weighting each channel by perceptual importance



Thresholding

- Grayscale → Binary: Choose threshold based on histogram of image intensities

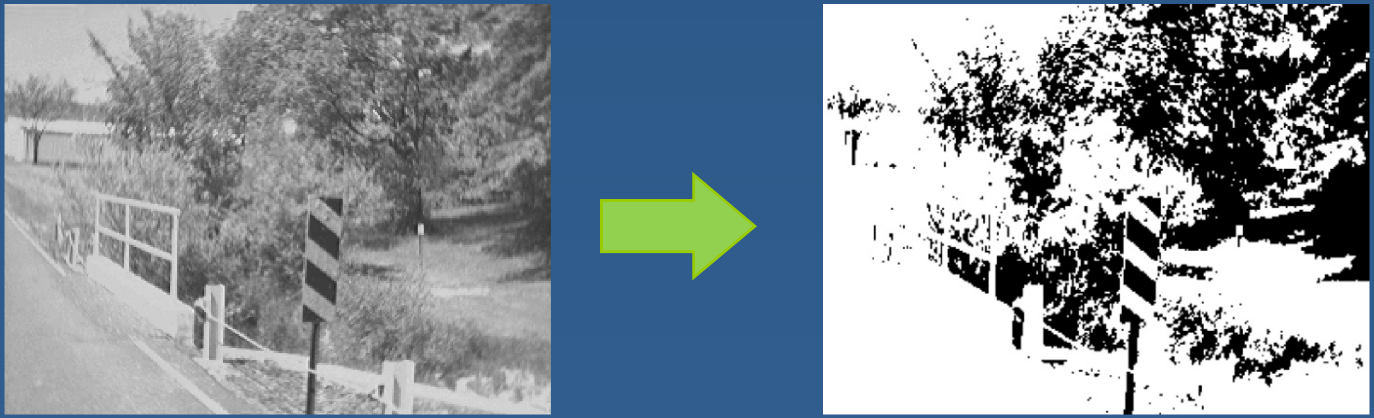
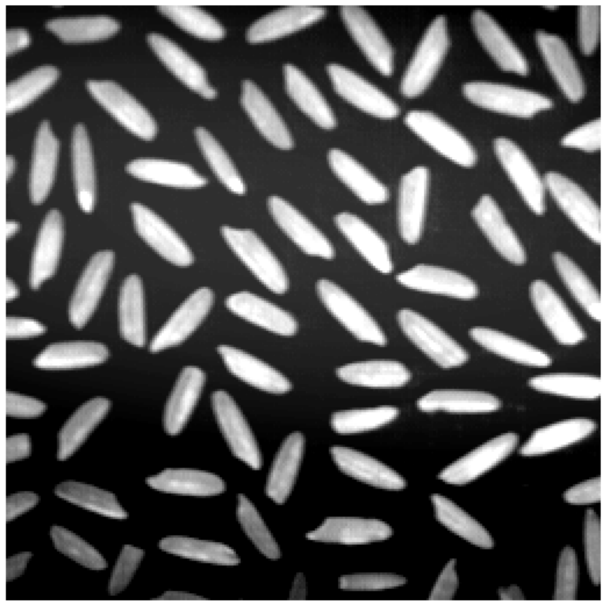
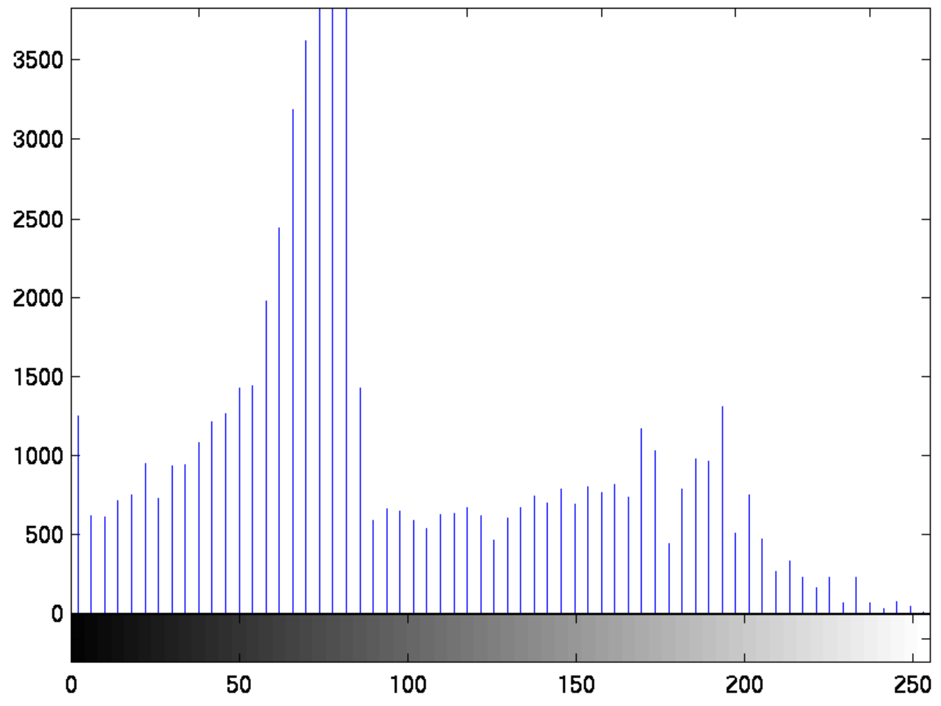


Image Comparison: Histograms





Color Similarity

- One measure is "Chrominance Distance": distance from color to a reference color

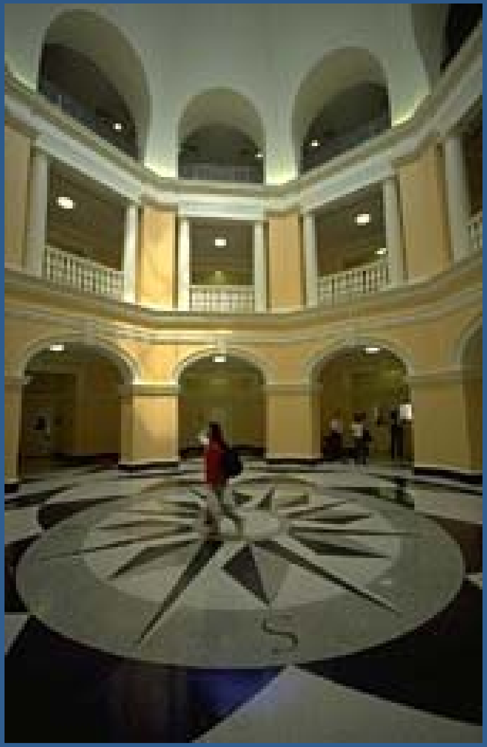
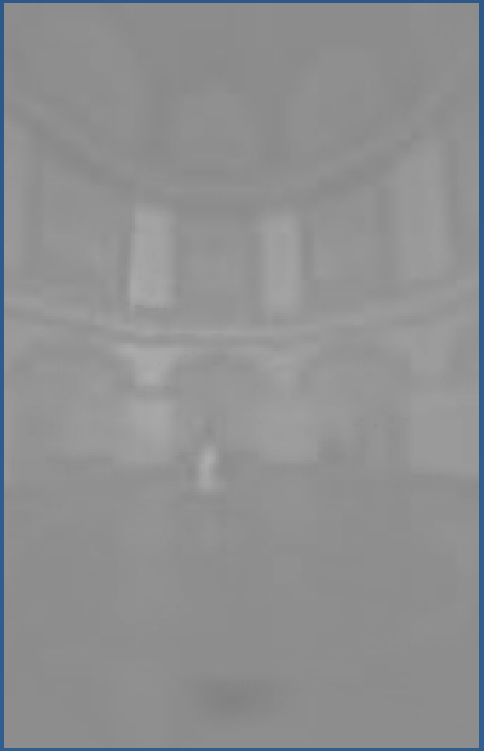
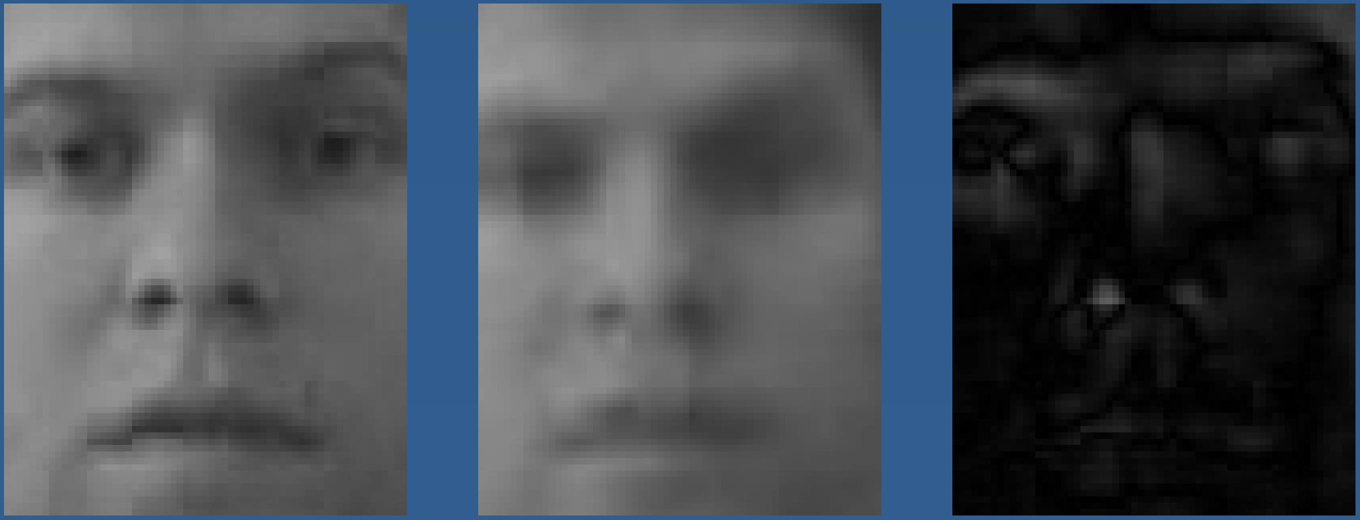



Image Comparison

- One approach to template matching is a sum of squared differences:

$$\sum_{x,y} [\mathbf{I}_T(x,y) - \mathbf{I}(x,y)]^2$$



Correlation for Template Matching

- Note that SSD formula can be written:

$$\sum_{x.y} \mathbf{I}_T^2(x,y) + \mathbf{I}^2(x,y) - 2\mathbf{I}_T(x,y)\mathbf{I}(x,y)$$

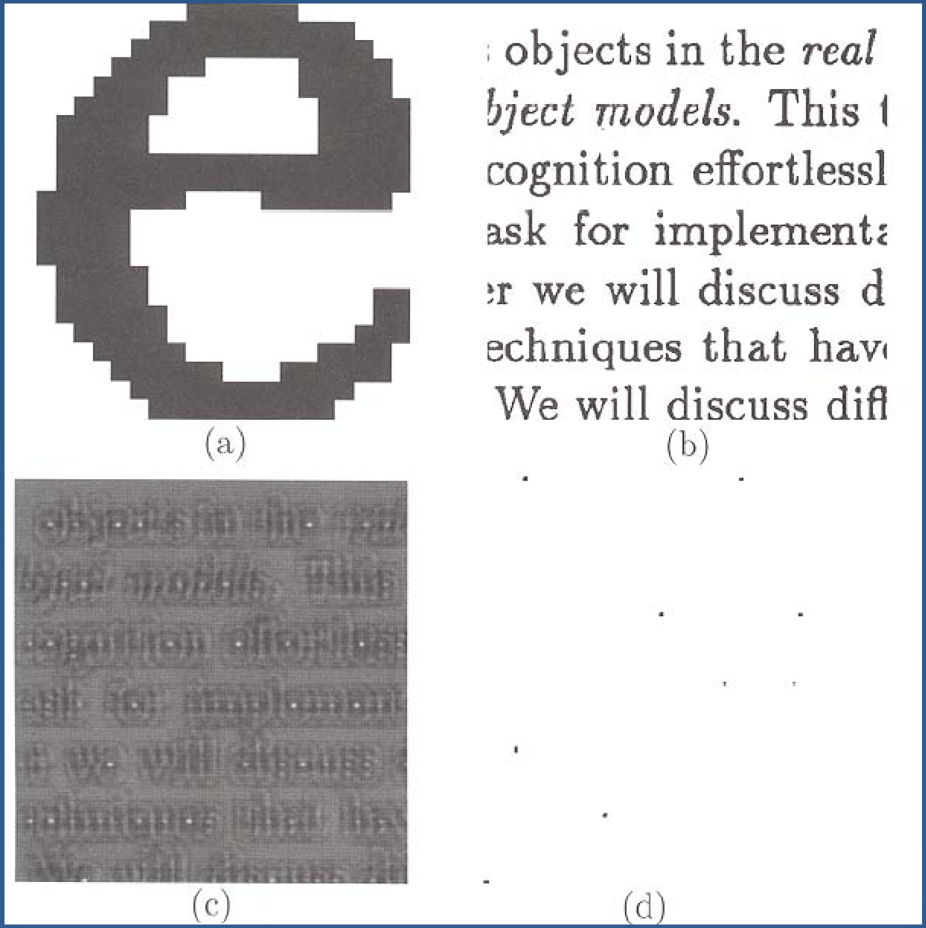
- When the last term is big, the mismatch is small—the dot product measures *correlation:*

$$\sum_{x,y} \mathbf{I}_T(x,y) \cdot \mathbf{I}(x,y)$$

- By normalizing by the vectors' lengths, we are measuring the angle between them

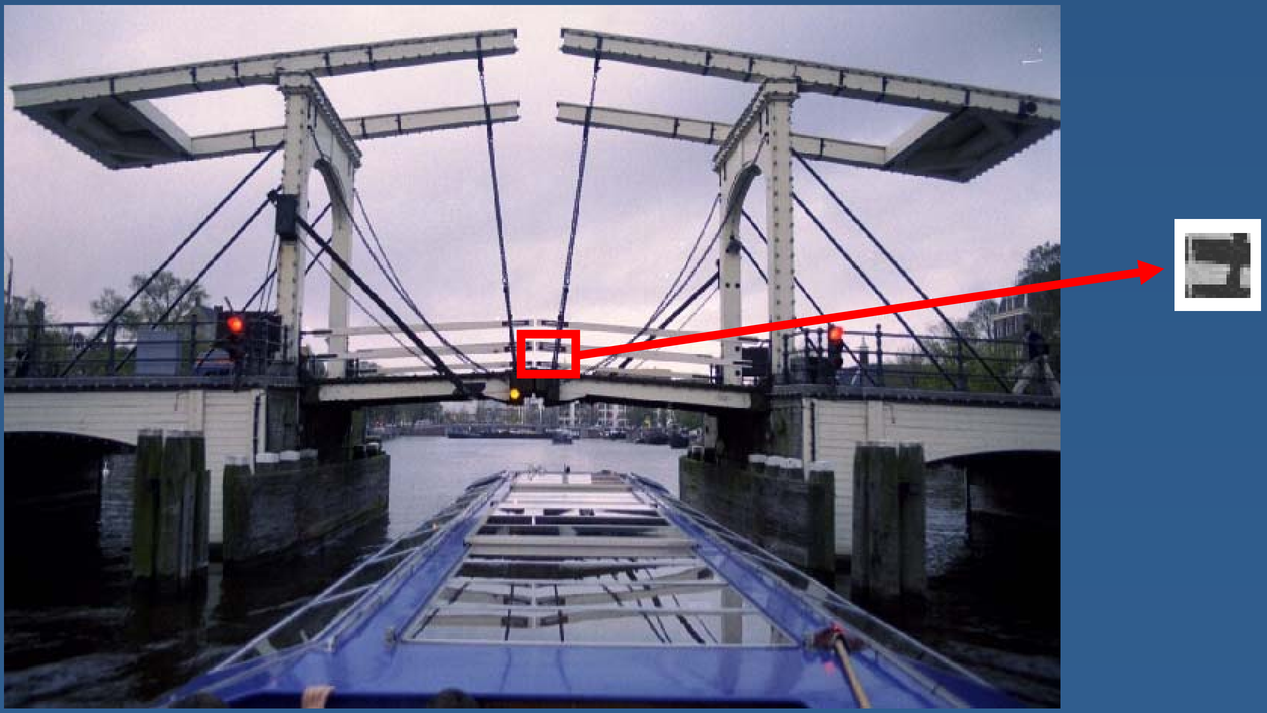
Normalized Cross-Correlation

- Shift *template* image over search image, measuring normalized correlation at each point
- Local maxima indicate template matches



: objects in the *real*
bject *models.* This
cognition effortlessl
ask for implement
er we will discuss d
echniques that hav
We will discuss diff

Template Matching Example



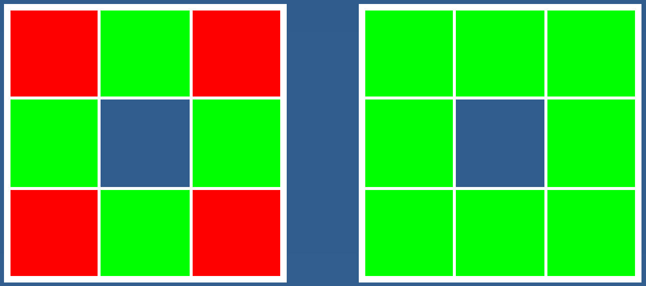
(courtesy of Sebastian Thrun)

Statistical Image Comparison: Color Histograms

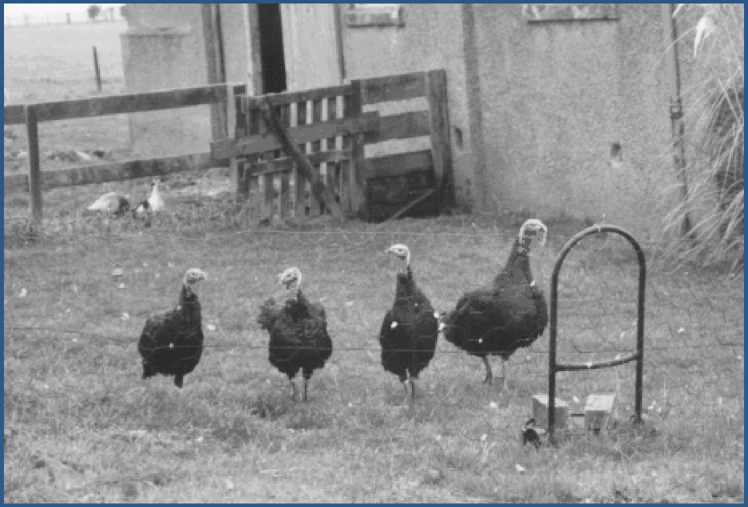
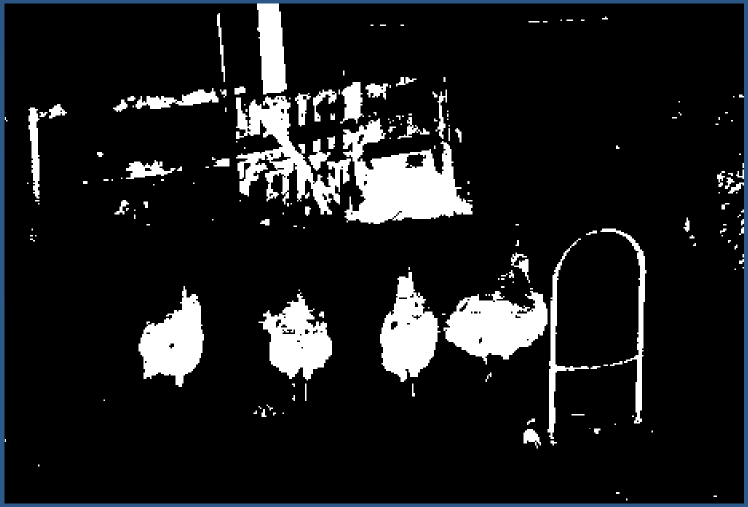
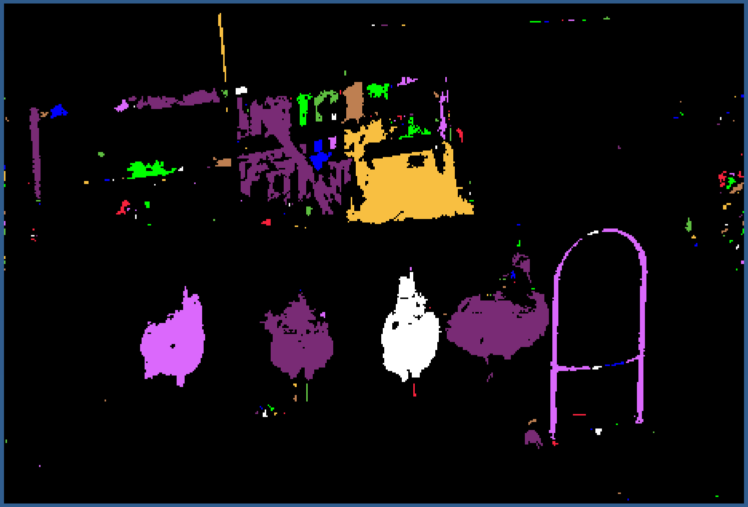
- Steps
  - Histogram RGB/HSI triplets over two images to be compared
  - Normalize each histogram by respective total number of pixels to get frequencies
  - Similarity is Euclidean distance between color frequency vectors
- Insensitive to geometric changes, including different-sized images

Connected Components

- After thresholding, method to identify groups or clusters of like pixels
- Uniquely label each *n*-connected region in binary image
- 4- and 8-connectedness

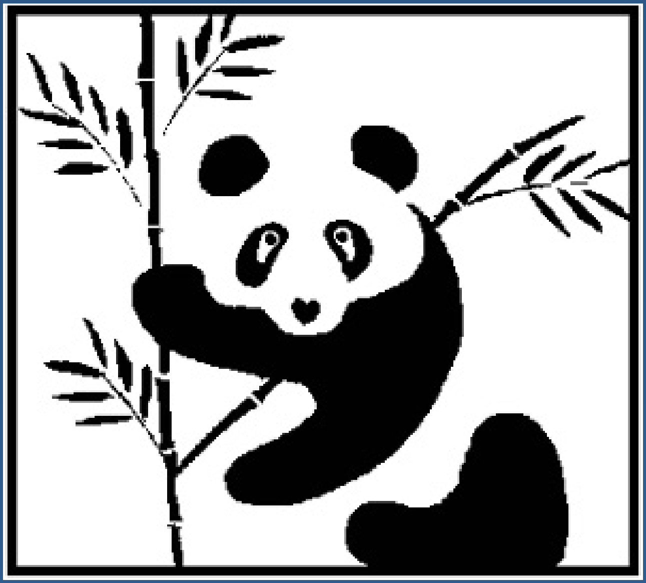


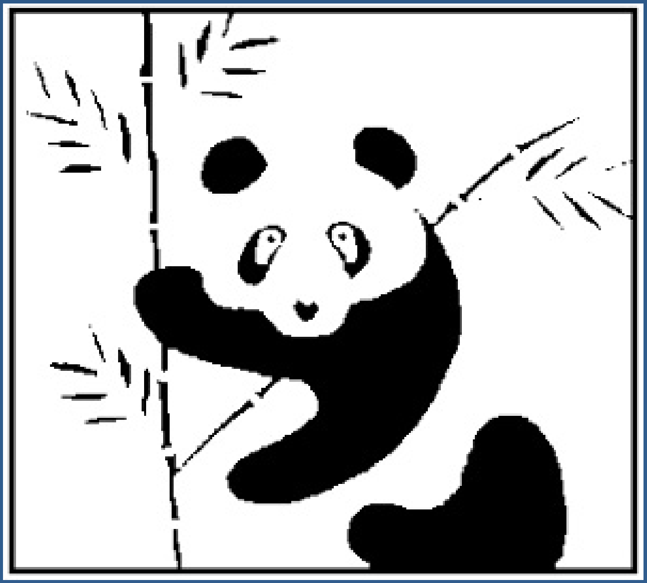
Connected Components Example

Binary Operations

- Dilation, erosion
  - Dilation: All 0's next to a 1 $\rightarrow$ 1 (Enlarge foreground)
  - Erosion: All 1's next to a 0 $\rightarrow$ 0 (Enlarge background)


Original


Eroded

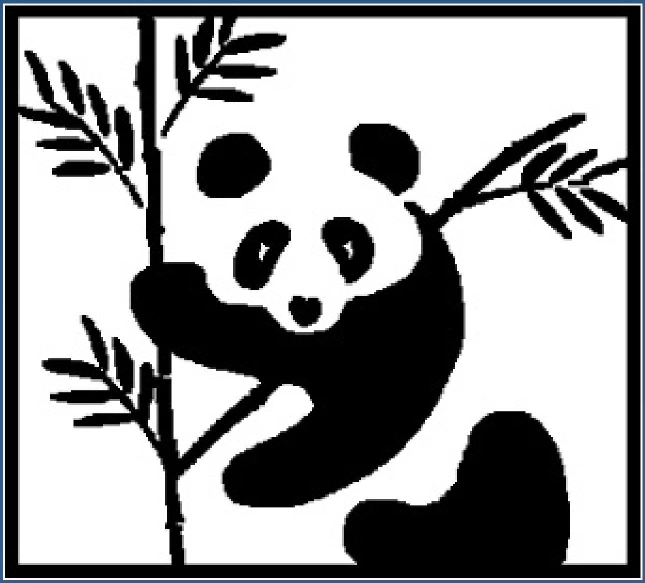
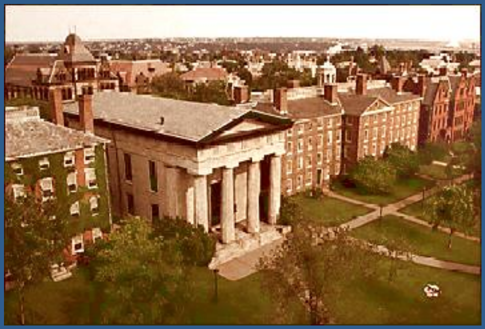

Dilated

Image Moments: Region Statistics

- Zeroth-order: Size/area
- First-order: Position (centroid)
- Second-order: Orientation

Other Effects

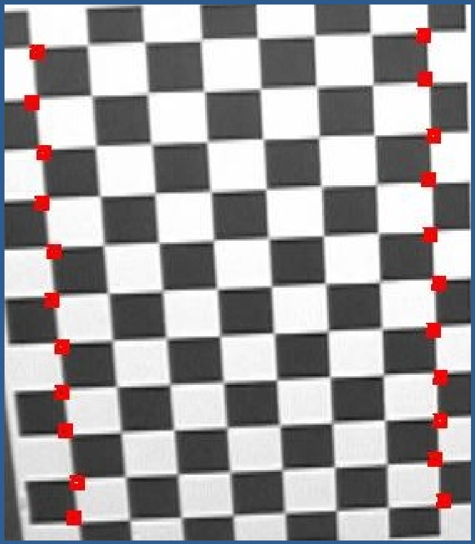
- Art Effects
  - Posterizing
  - Faked "Aging"
  - "Impressionist" pixel remapping
- Technical Effects
  - Color remapping
  - Grayscale conversion
  - Contrast balancing



(courtesy Andries van Dam)

Finding Corners

- Edge detectors perform poorly at corners.
- Corners provide repeatable points for matching, so are worth detecting.
- Problem:
  - Exactly at a corner, gradient is ill-defined.
  - However, in the region around a corner, gradient has two or more different values.



Harris Corner Detector

Sum over a small region around the hypothetical corner

Gradient with respect to x, times gradient with respect to y

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

Matrix is symmetric

Simple Case

- First, consider case where:

$$C = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

- This means dominant gradient directions align with x or y axis
- If either λ is close to 0, then this is **not** a corner, so look for locations where both are large.

General Case

- It can be shown that since C is rotationally symmetric:

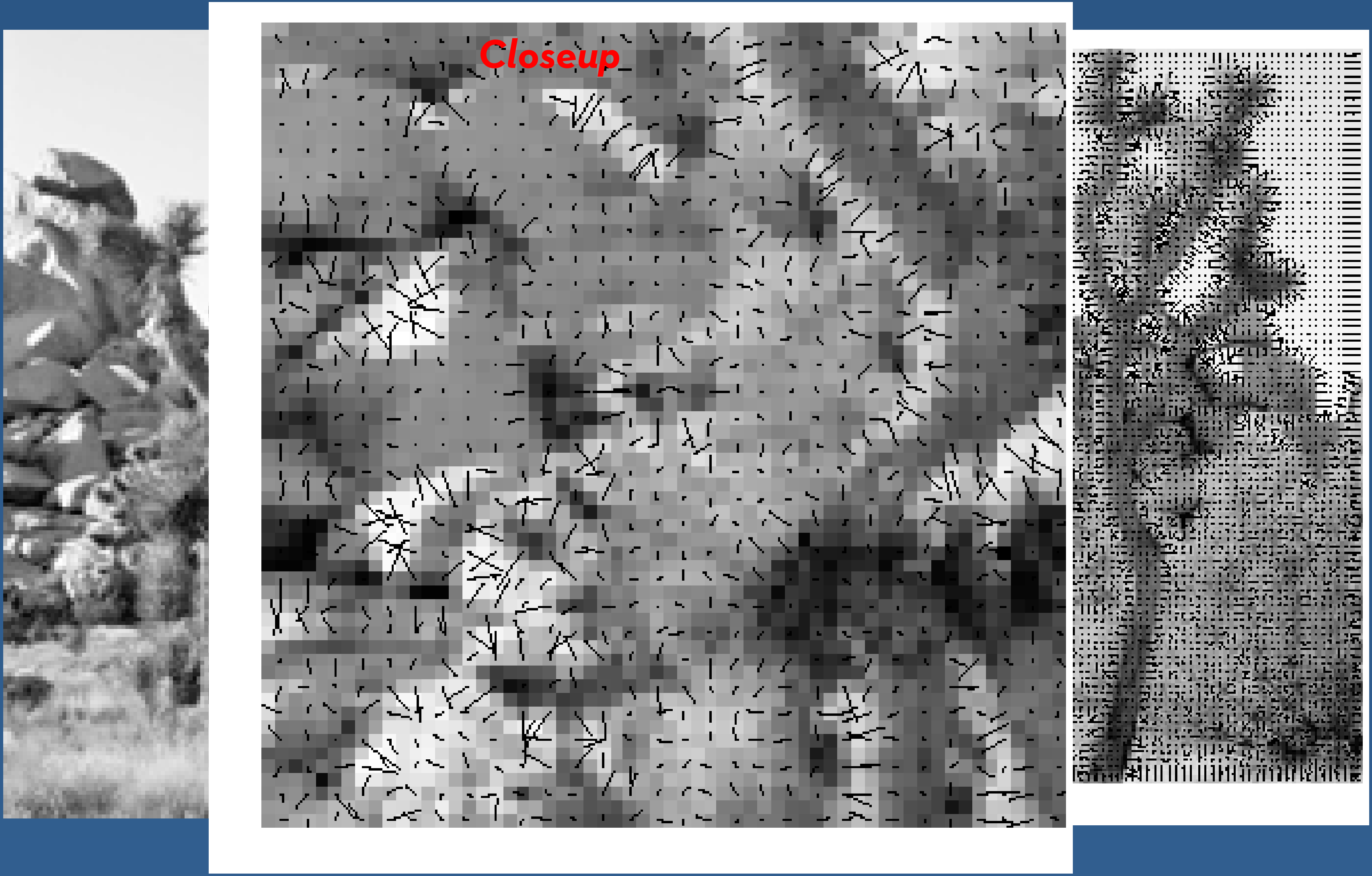
$$C = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

- So every case is like a rotated version of the one on last slide.

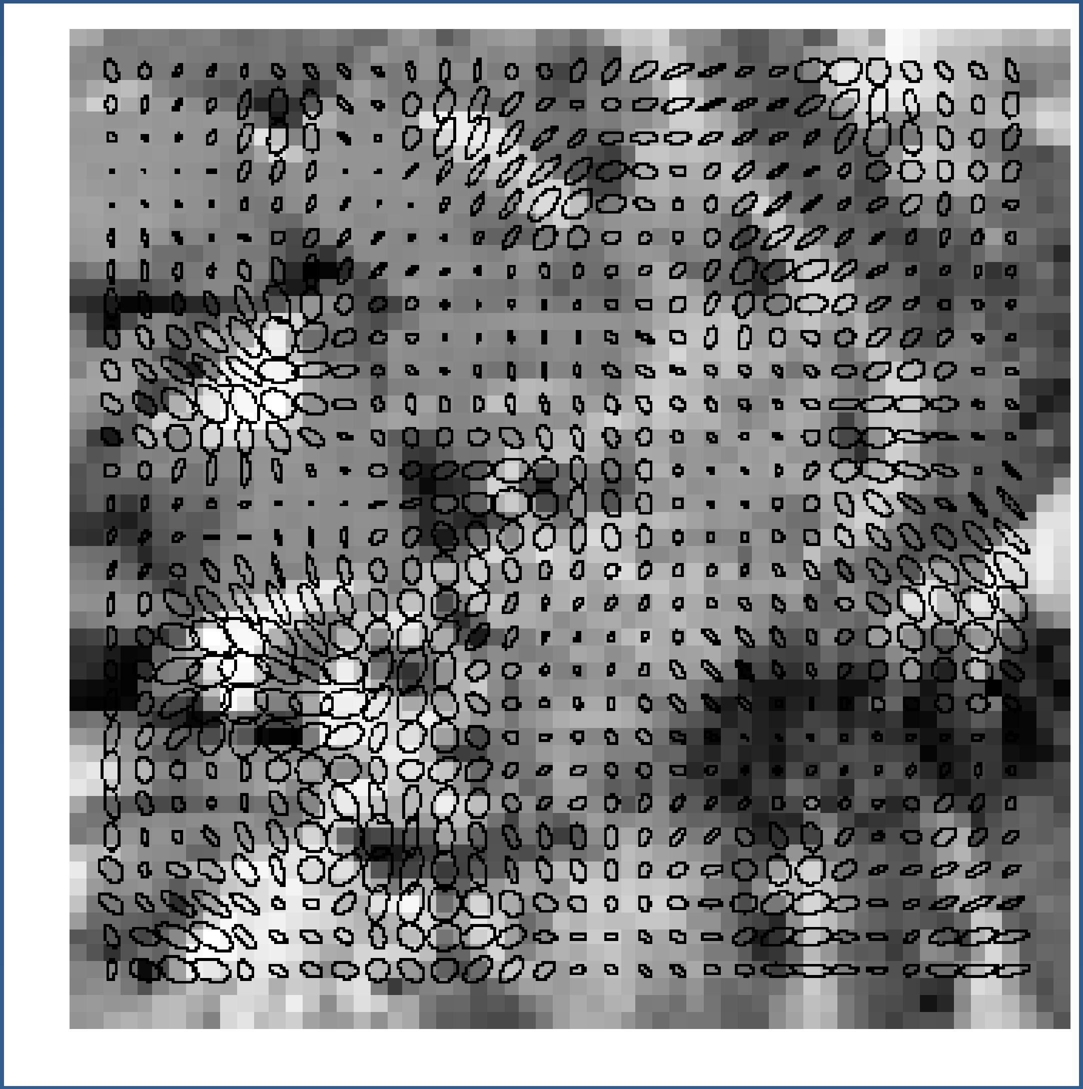
So, To Detect Corners

- Filter image with Gaussian to reduce noise
- Compute magnitude of the *x* and *y* gradients at each pixel
- Construct C in a window around each pixel (Harris uses a Gaussian window – just blur)
- Solve for product of l_S (determinant of C)
- If l_S are both big (product reaches local maximum and is above threshold), we have a corner (Harris also checks that ratio of l_S is not too high)

Gradient Orientation

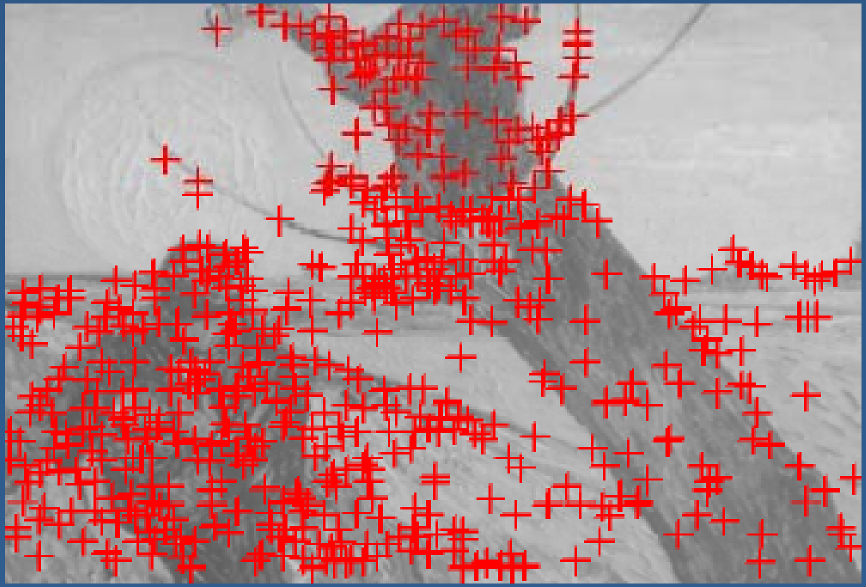


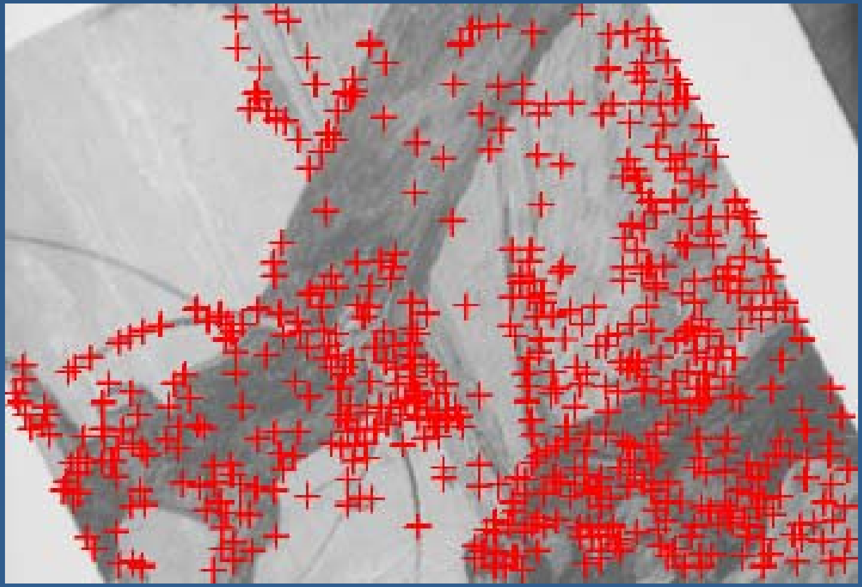
Corner Detection



- Corners are detected where the product of the ellipse axis lengths reaches a local maximum.

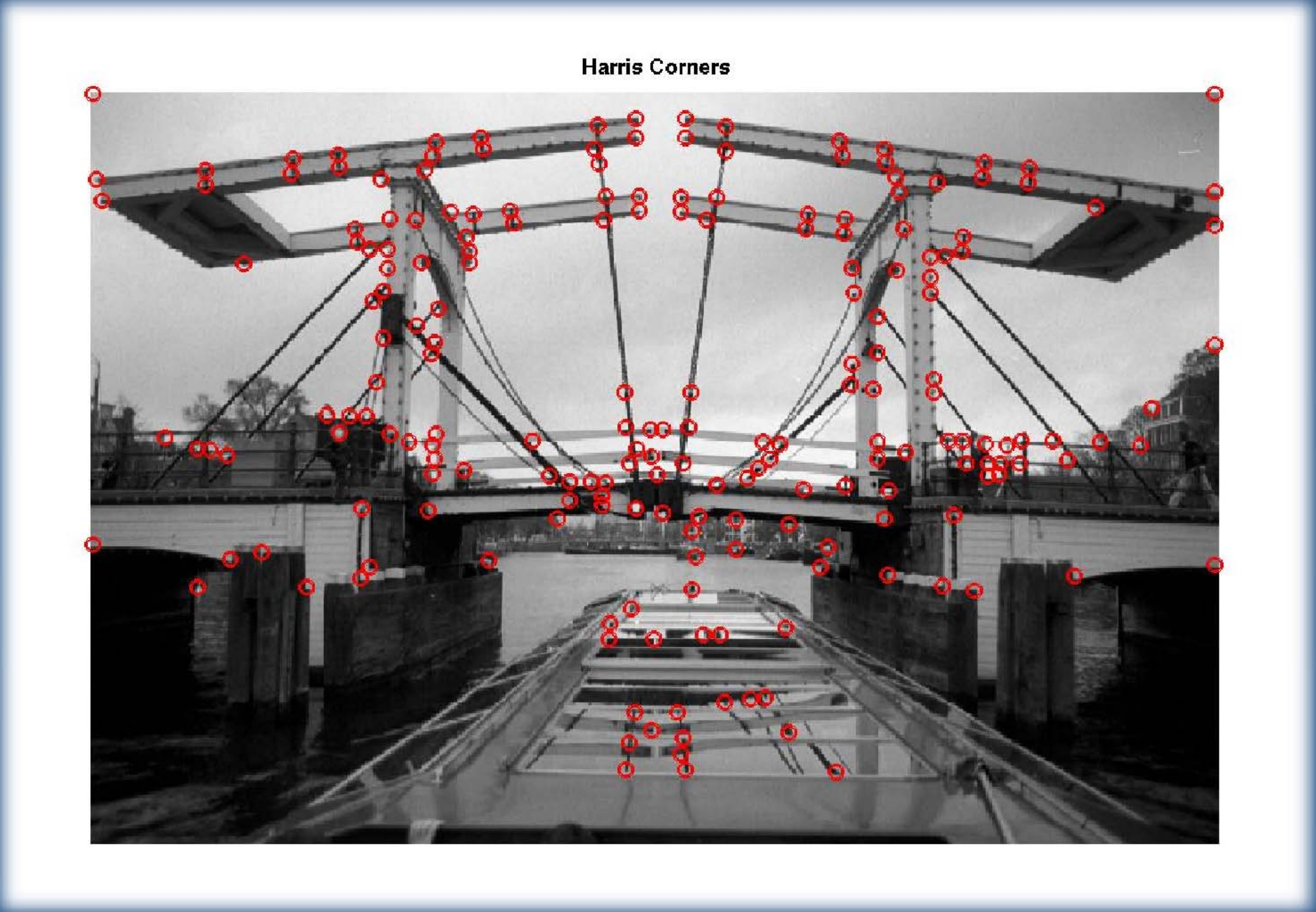
Harris Corners





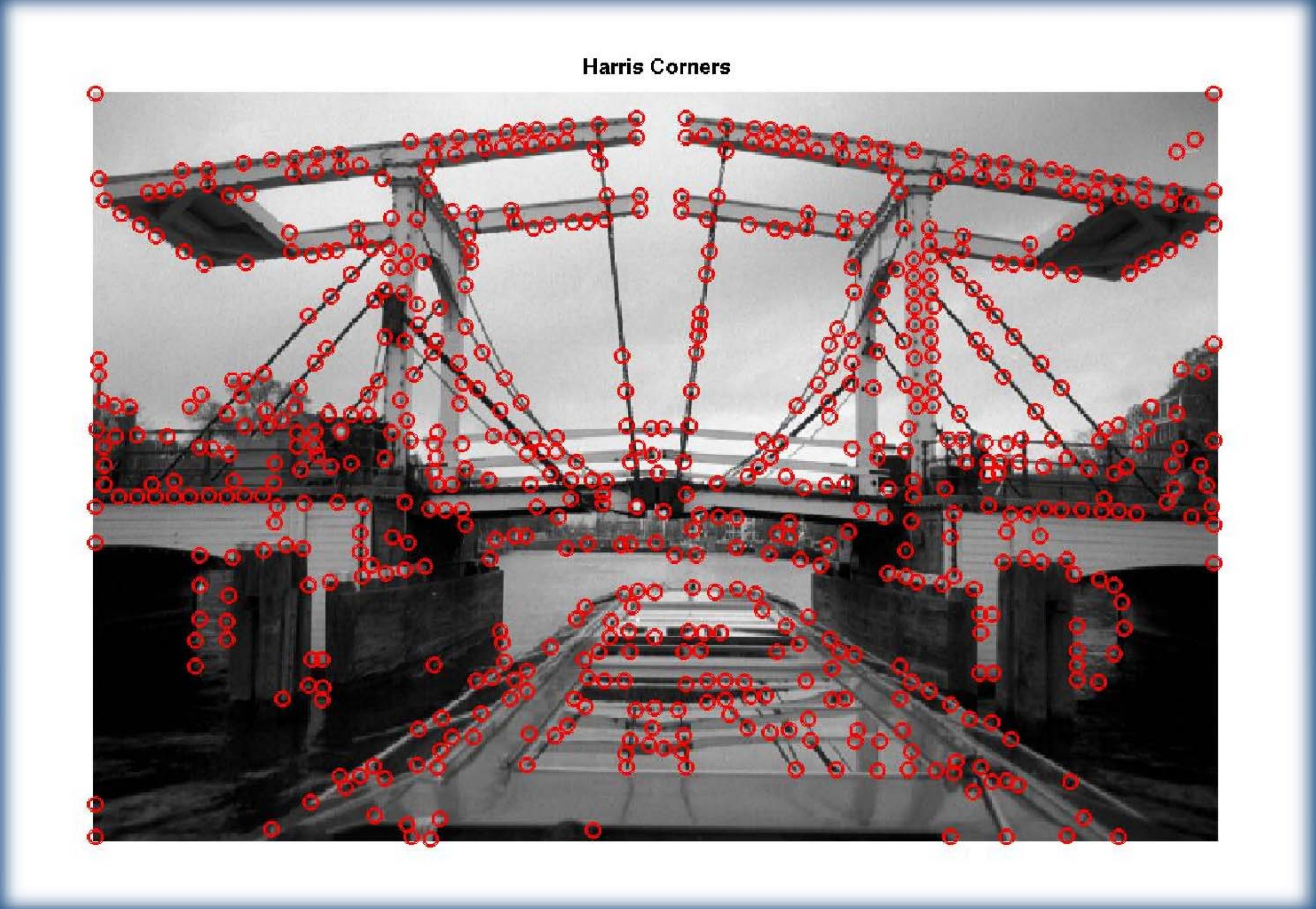
(courtesy of Sebastian Thrun)

Example (σ=0.1)



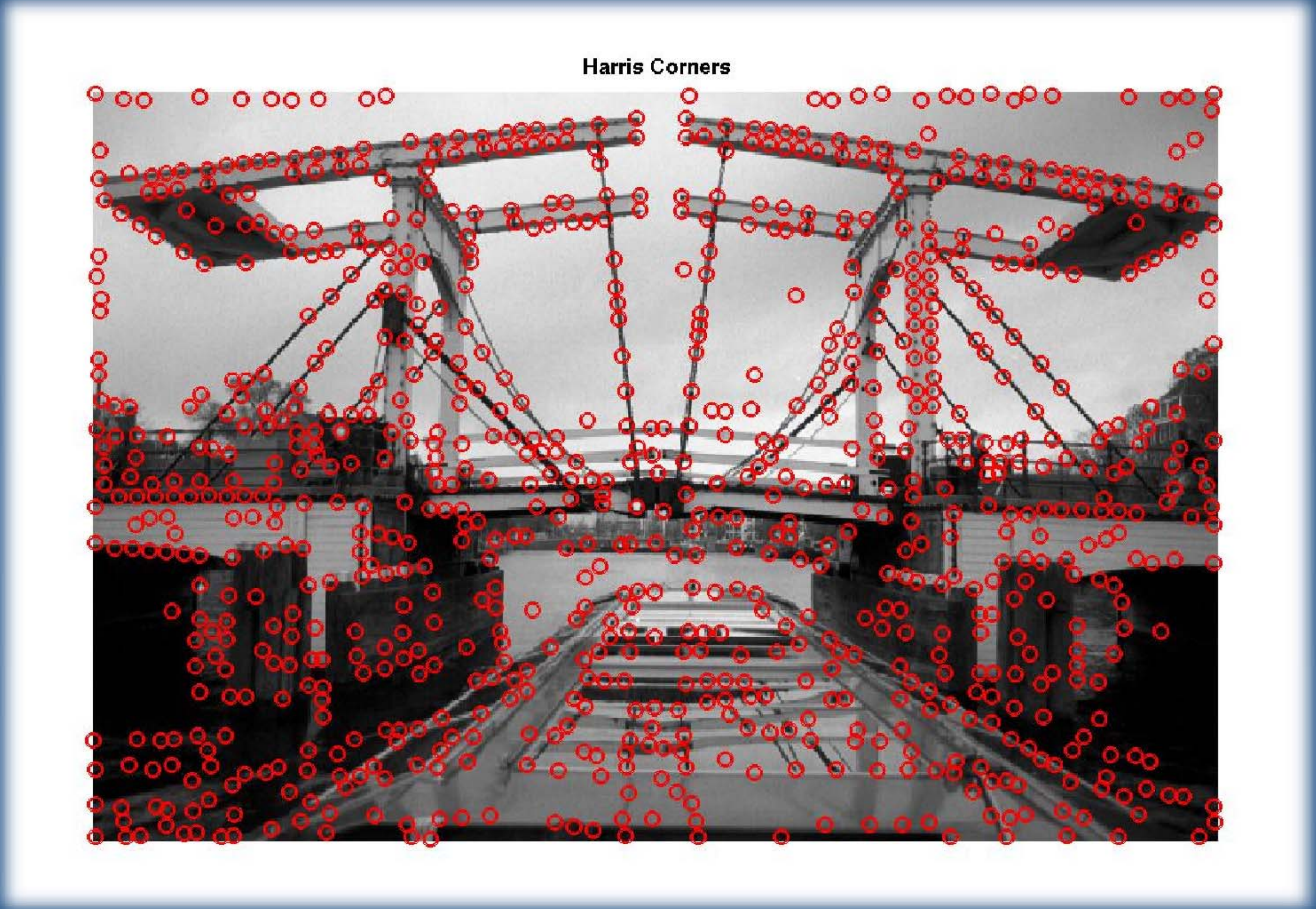
(courtesy of Sebastian Thrun)

Example (σ=0.01)



(courtesy of Sebastian Thrun)

Example (σ=0.01)

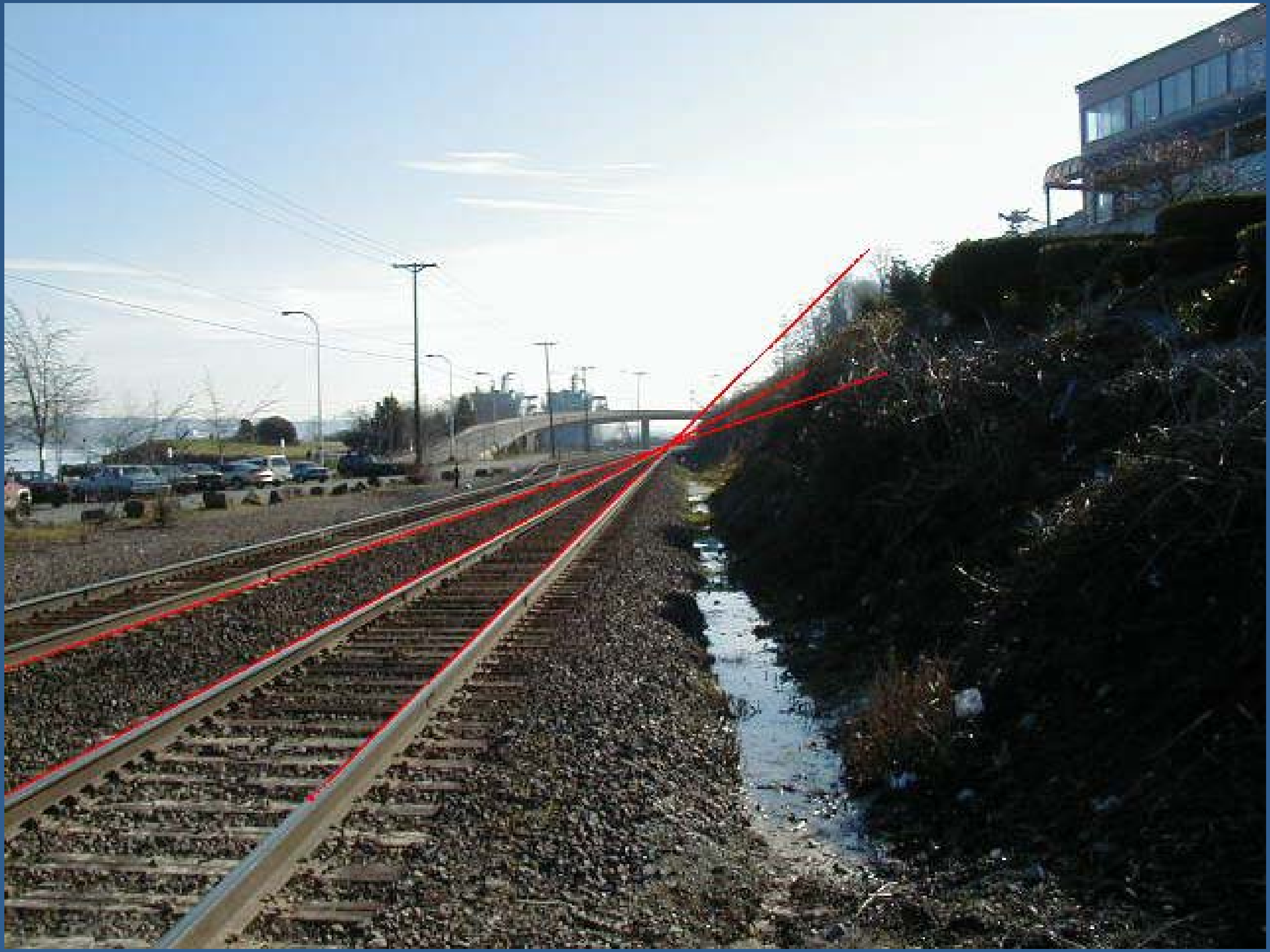


(courtesy of Sebastian Thrun)

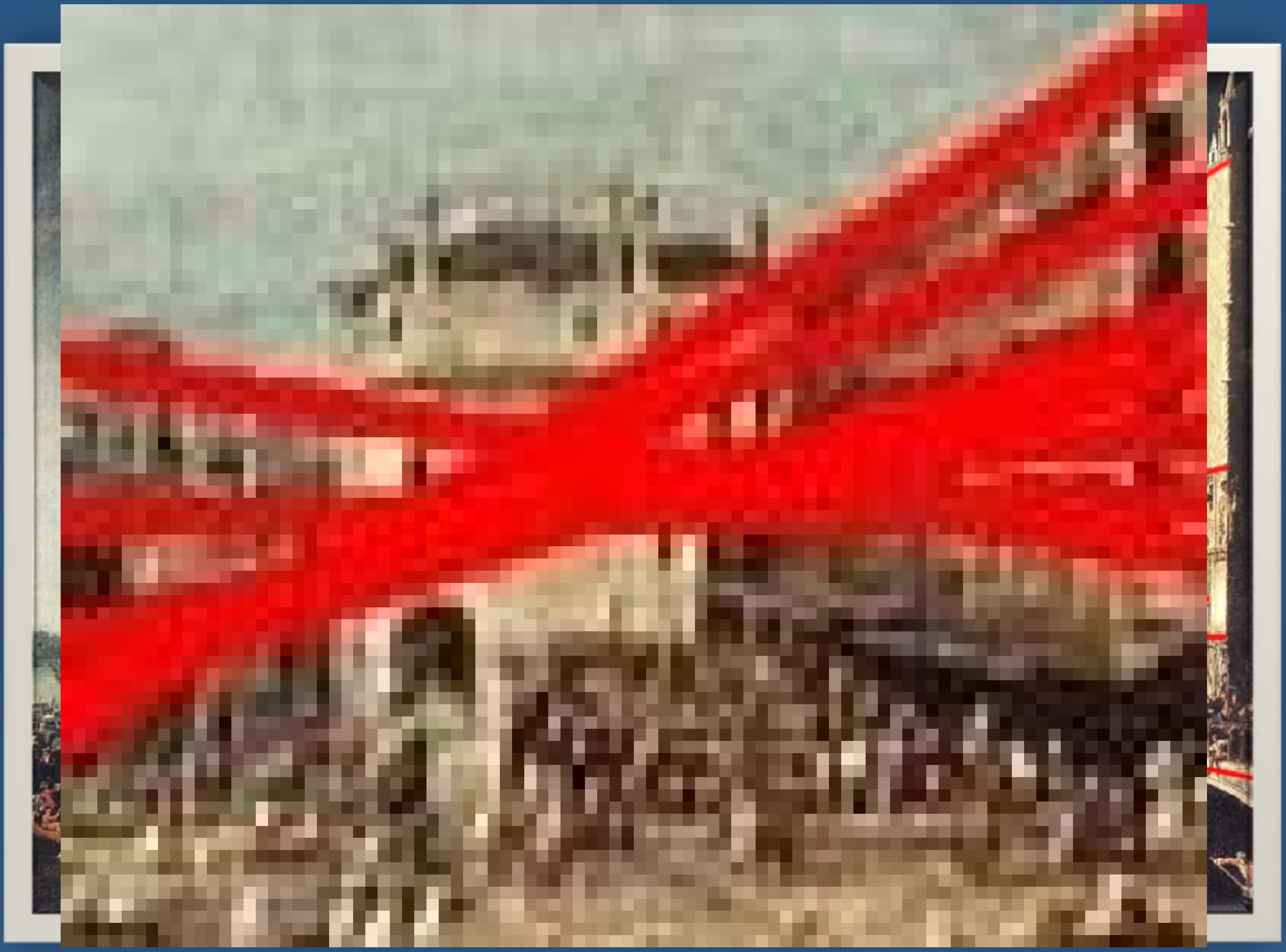
Features?

Global, not Local

Vanishing Points

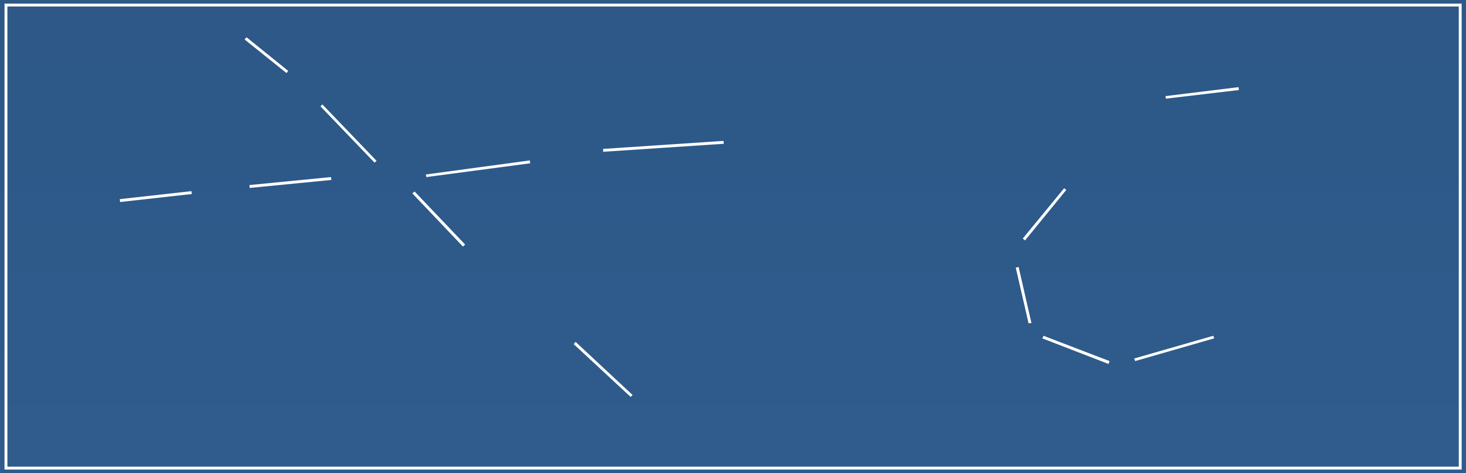


Vanishing Points

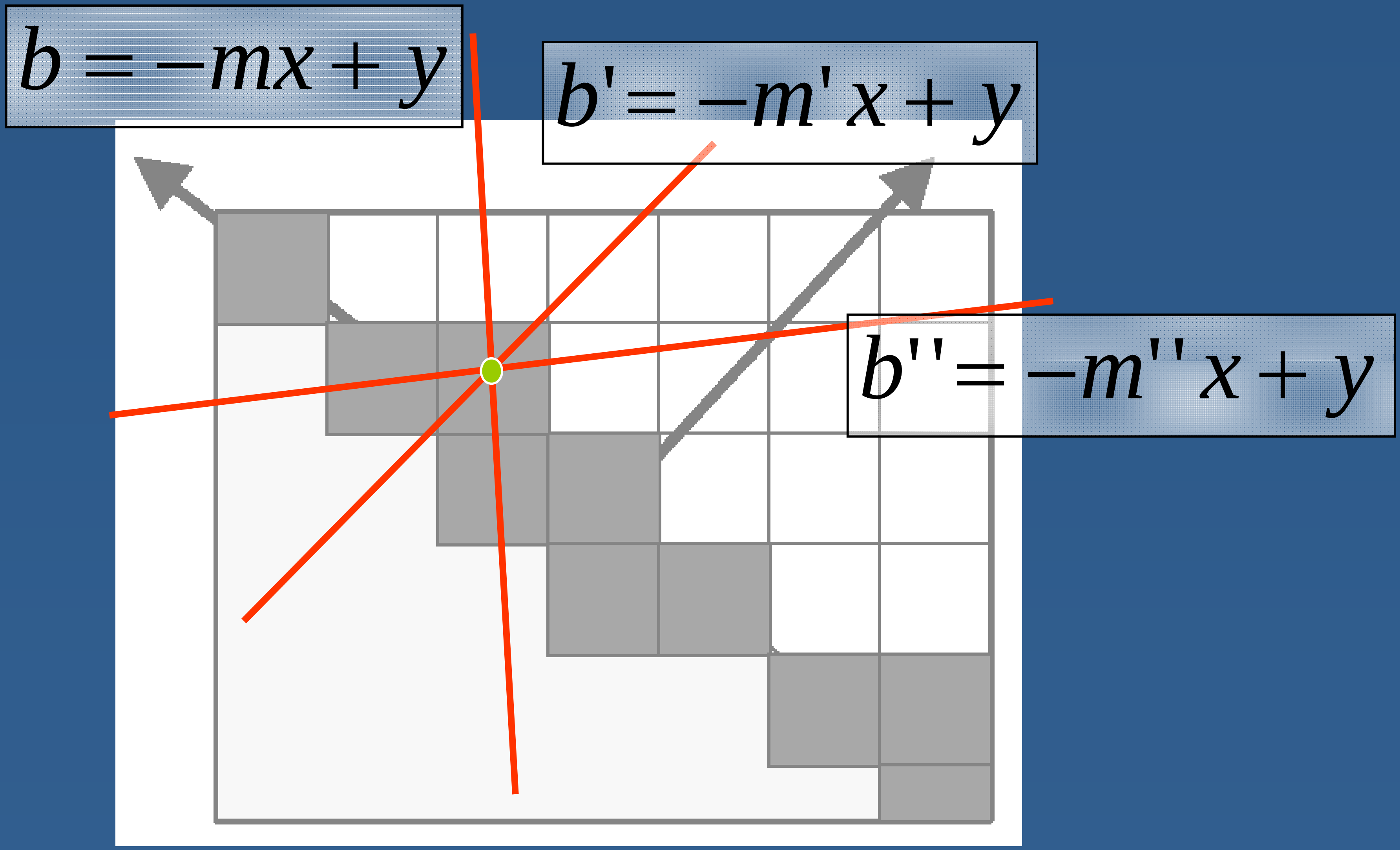


A. Canaletto [1740], Arrival of the French Ambassador in Venice

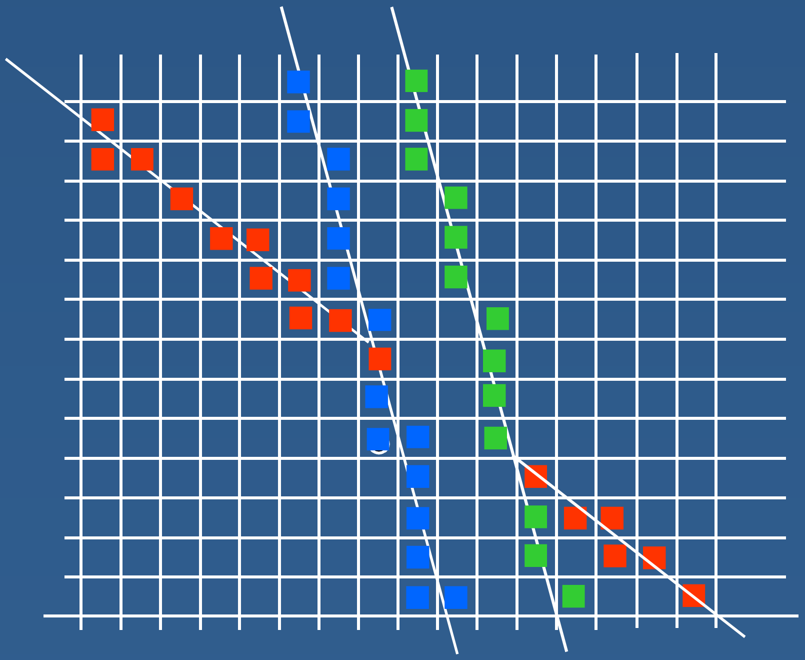
From Edges to Lines

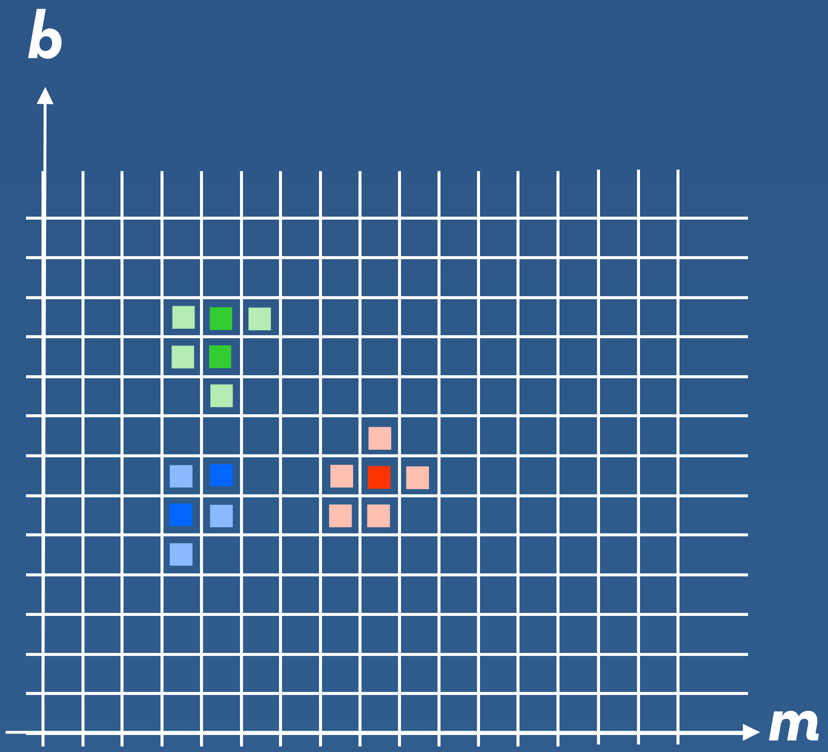


Hough Transform


$$b = -mx + y$$
$$b' = -m'x + y$$
$$b'' = -m''x + y$$

Hough Transform: Quantization



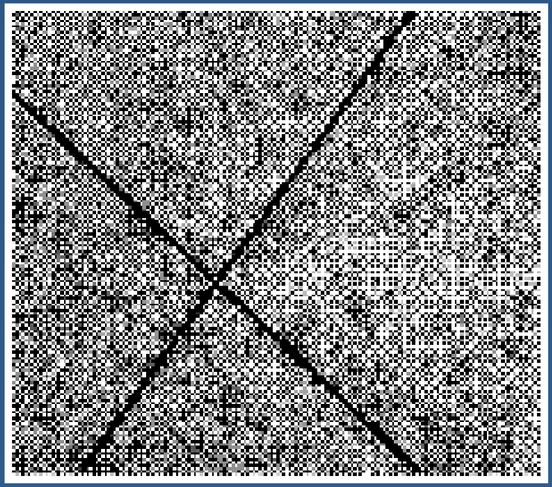


(courtesy of Sebastian Thrun)

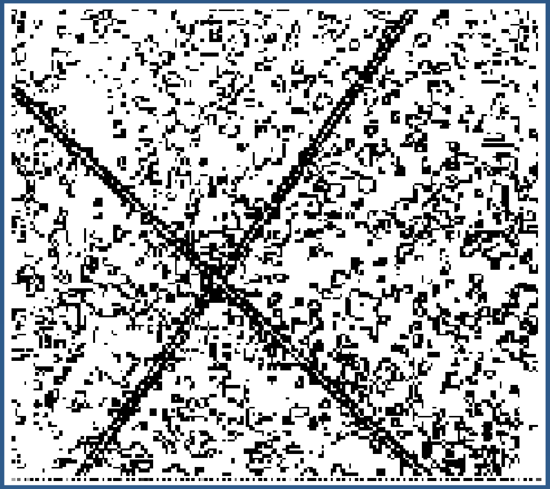
Hough Transform: Algorithm

- For each image point, determine
  - most likely line parameters b,m (direction of gradient)
  - strength (magnitude of gradient)
- Increment parameter counter by strength value
- Cluster in parameter space, pick local maxima

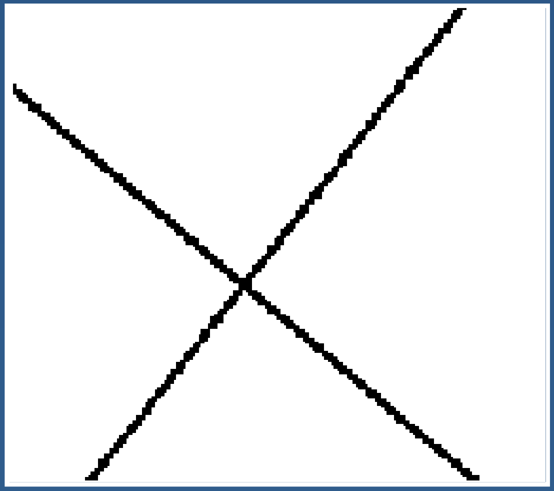
Hough Transform: Results



Image

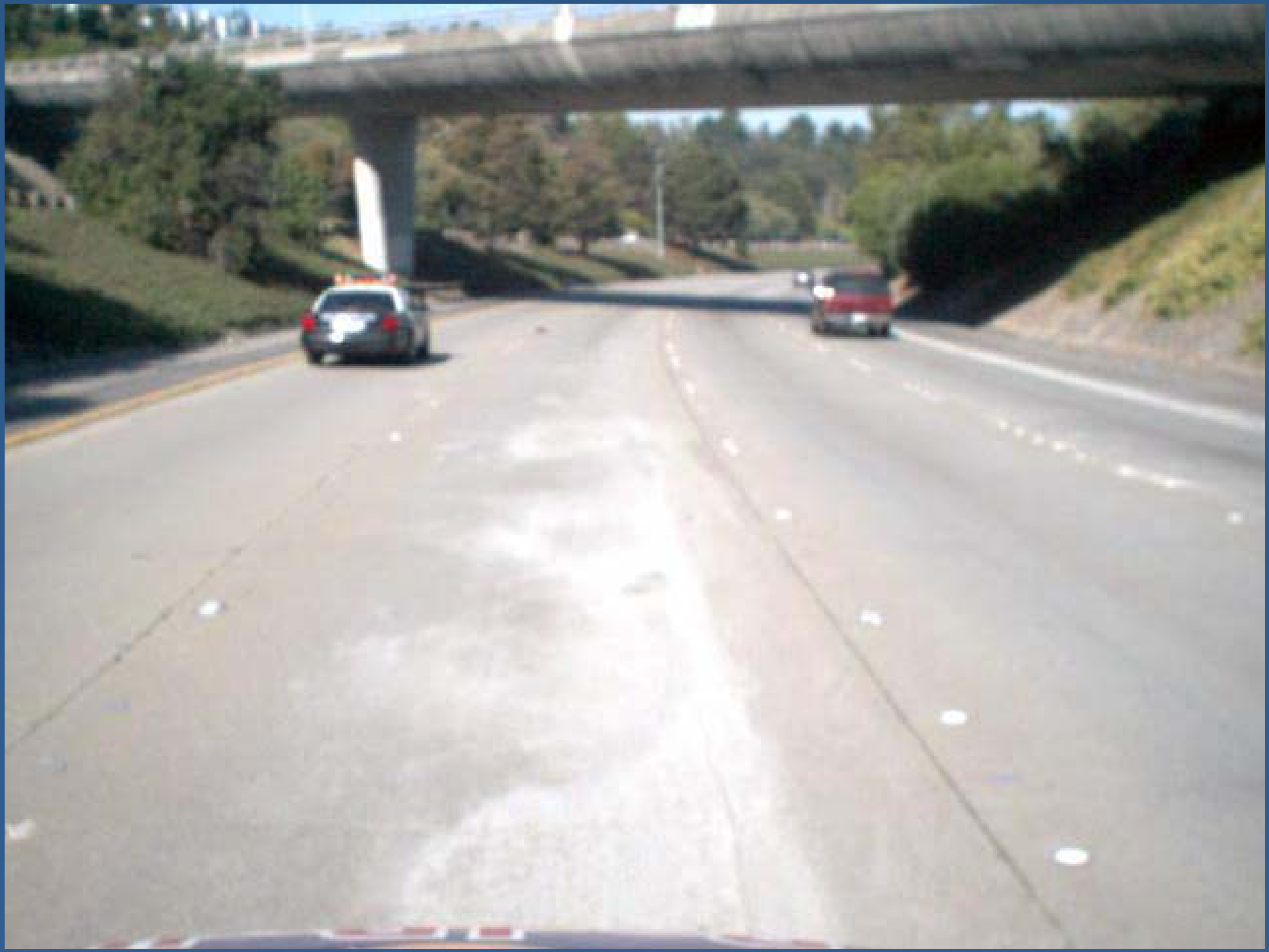


Edge detection



Hough Transform

Hough Transform?



Summary

- Many kinds of mathematical operations can be performed on images to extract basic information about shapes and features
- Image processing does not itself lead to image understanding...
- But it's often a good start

Reading for Next Lecture

- Introduction to the MATLAB Image Processing Toolbox
- Assignment #1 Question #4: MATLAB introduction