OUTLINE

- Shape of boundaries:
  - Hough transform (straight lines)
- Color Fundamentals
  - Color matching experiment
  - Color spaces
    - RGB
    - HSI (HSV)
- Image segmentation based on color.
  - Geometrical properties of regions
- Image segmentation:
  - Seed segmentation algorithm

Shape of boundaries

- There are several techniques for representing the shape of boundaries (edges).

- These representations are the abstraction of edges in a symbolic form (e.g. useful in object “recognition”).

- For instance, the grouping of edge points into a *straight lines*: Hough transform.

Difficulty of Line Fitting

- Extra edge points (clutter), multiple models: Which points go with which line, if any?

- Only some parts of each line detected, and some parts are missing: How to find a line that bridges missing evidence?

- Noise in measured edge points, orientations: How to detect true underlying parameters?

Voting technique

- It’s not feasible to check all combinations of features by fitting a model to each possible subset.
- **Voting** is a general technique where we let the features vote for all models that are compatible with it.
  - Cycle through features, cast votes for model parameters.
  - Look for model parameters that receive a lot of votes.
- Noise & clutter features will cast votes too, but typically their votes should be inconsistent with the majority of “good” features.

Hough Transform

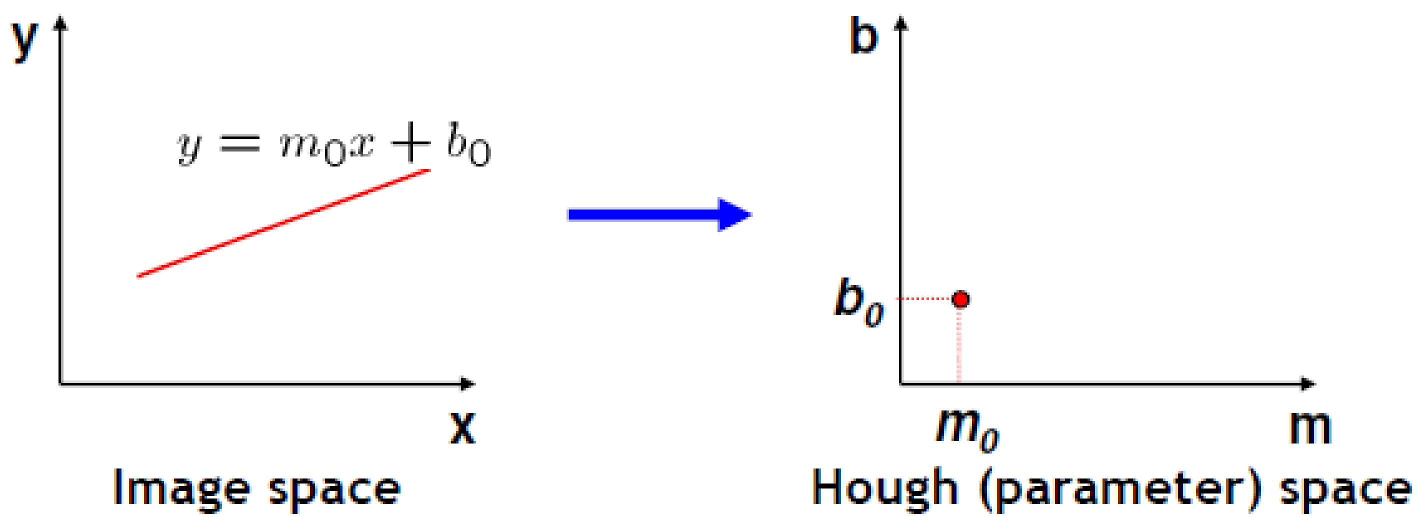
- Hough Transform is a voting technique that can be used to solve all of those difficulties.

- Main idea:
  1. Record all possible lines on which each edge point lies.
  2. Look for lines that get many votes.

Hough transform

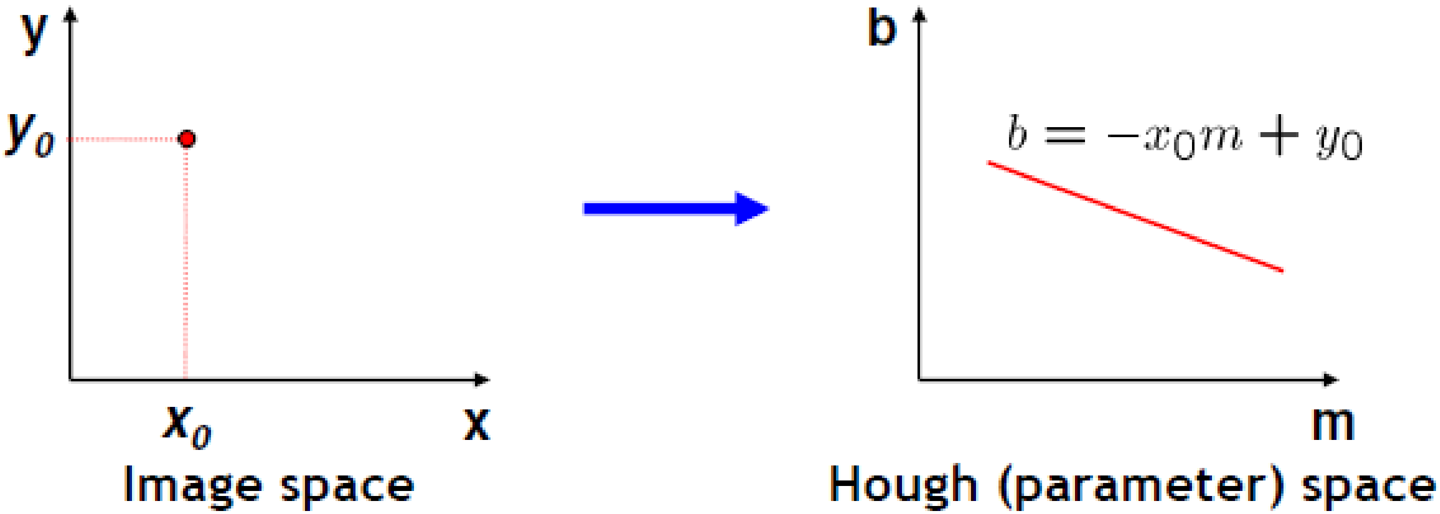
- We will use the Hough transform to fit the equation of a straight line to the edge points.
- The equation of a line $y=mx+b$ can be rewritten as $b=(-x)m+y$, this is a the equation of a line in the m-b parametric space.
- Therefore, a point (x,y) in the x-y space is mapped to a straight line in the m-b space.
- The lines intersect at a single point (m',b') in the m-b space, if the (x,y) points belong to the same straight line in the x-y space.

Hough transform



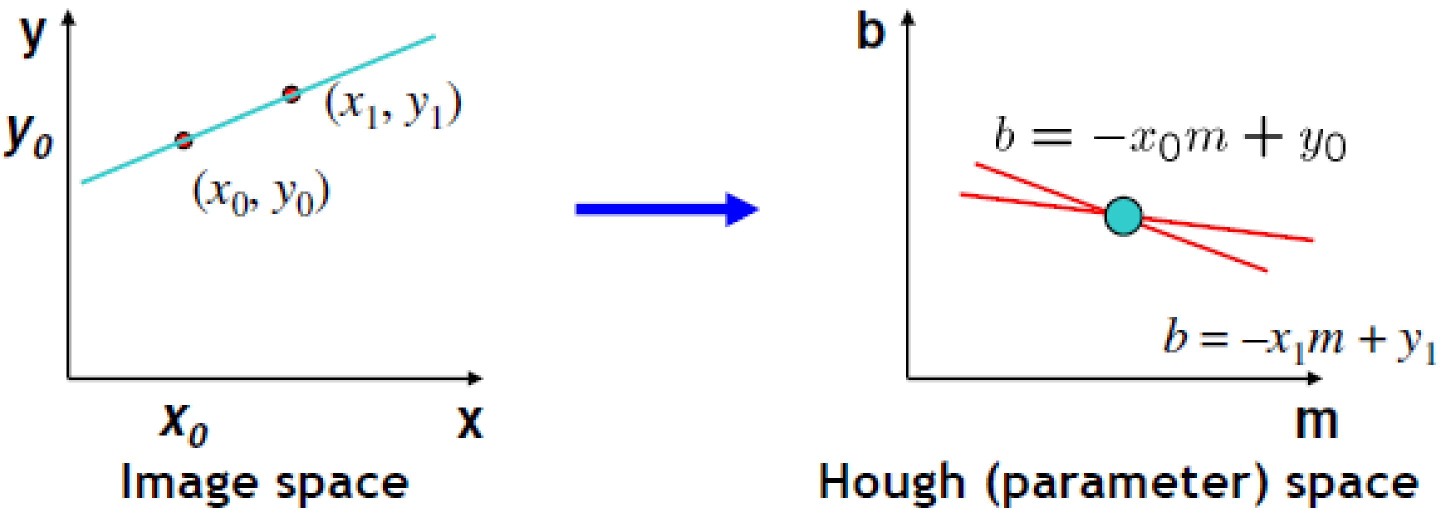
- A line in the image corresponds to a point in Hough space.
- To go from image space to Hough space:
  - Given a set of points *(x,y)*, find all *(m,b)* such that $y = mx + b$

Hough transform



- What does a point (x_0, y_0) in the image space map to?
  - Answer: the solutions of $b = -x_0 m + y_0$
  - This is a line in Hough space.

Hough transform



- What are the line parameters for the line that contains both (x_0, y_0) *and* (x_1, y_1)?
- It is the intersection of the lines $b = -x_0 m + y_0$ and $b = -x_1 m + y_1$

Hough transform

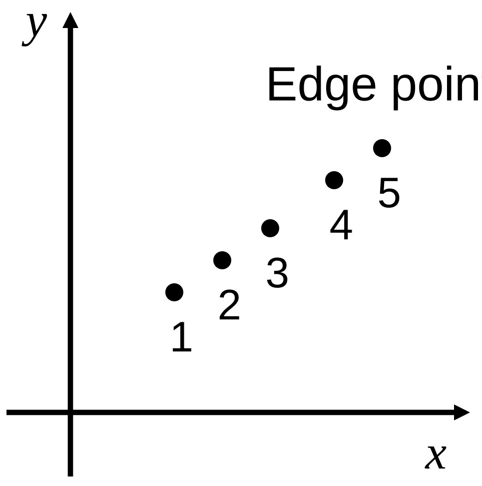
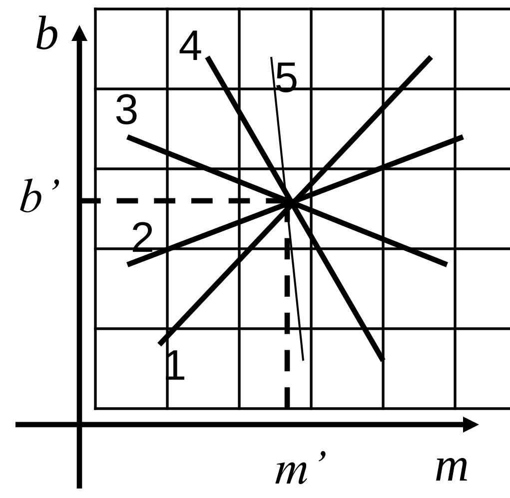


Image space



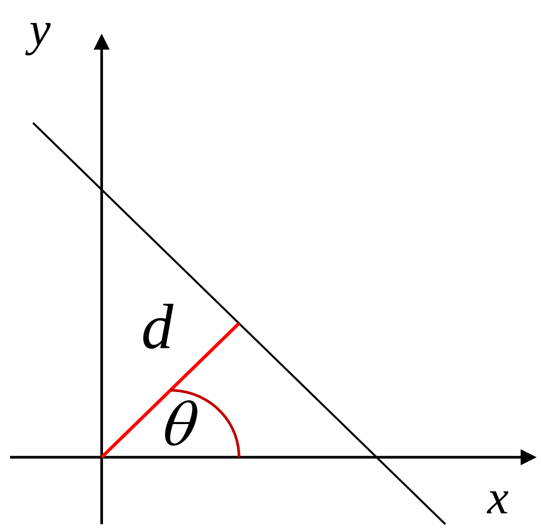
Hough (parameter) space

- How can we use this to find the most likely parameters (*m,b*) for the most prominent line in the image space?
  - Let each edge point in image space <u>vote</u> for a set of possible parameters in Hough space.
  - Accumulate votes in discrete set of bins; parameters with the most votes (*m',b'*) indicate line in image space.

Hough transform

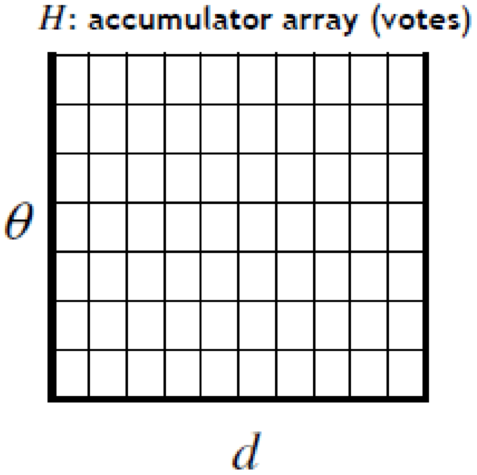
- The parametric model presented has some difficulties in the representation of vertical straight lines, because the parameter m tends to infinity.
- Thus, we chose the polar representation of a straight line.

- The polar form of the equation of a straight line:
  $d = x\cos\theta + y\sin\theta$



Hough transform

- Basic Hough transform algorithm

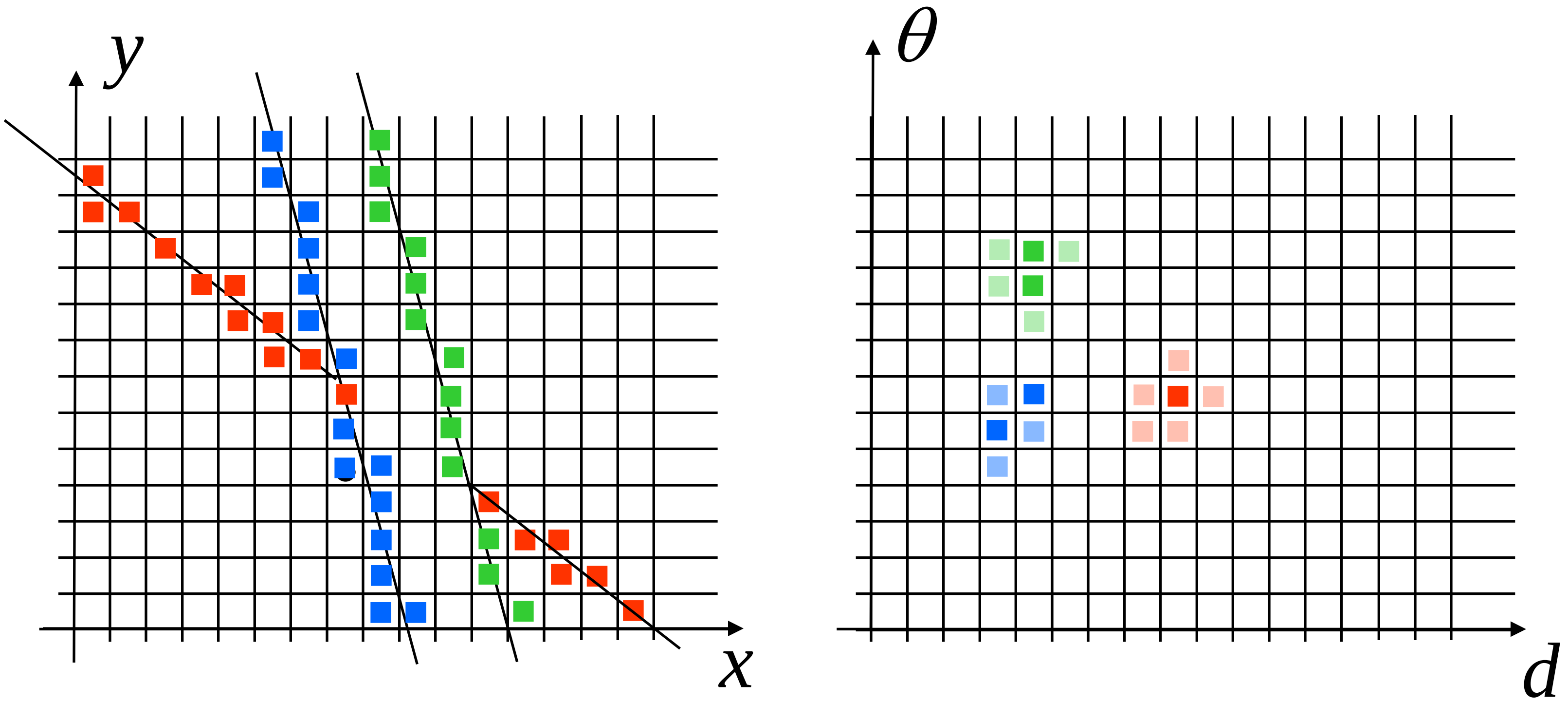


1. Initialize $H[d, \theta] = 0$
2. For each edge point (x,y) in the image
3. for $\theta = 0$ *to 180* *//sampling*

   $d = x \cos \theta - y \sin \theta$ *//quantization*

   $H[d, \theta] = H[d, \theta] + 1$
4. Find the value(s) of (d^*, θ^*) where $H[d, \theta]$ is maximal.
5. The detected line in the image is given by
   $d^* = x \cos\theta^* + y \sin\theta^*$

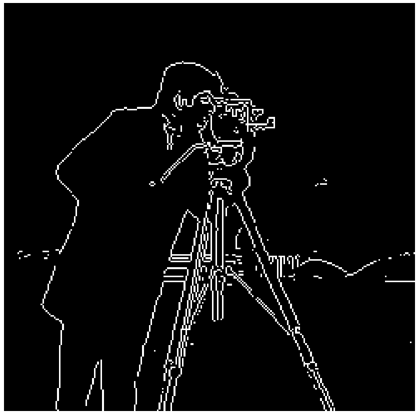
Hough transform



For real edges there is noise, thus the edge points are not strait lines: we have small areas of high values, not points of high values. Detecting lines by finding maxima in the parameter space.

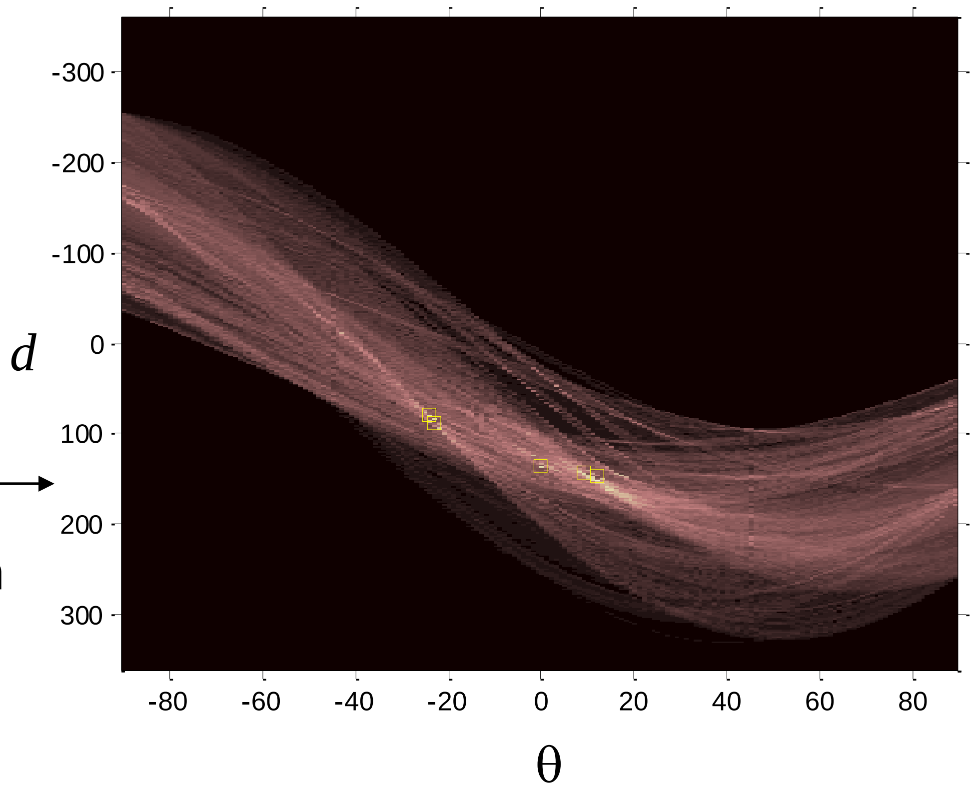
Hough transform

Edge detection
(e.g. Sobel)



Hough
transform

To identify peaks in Hough transform

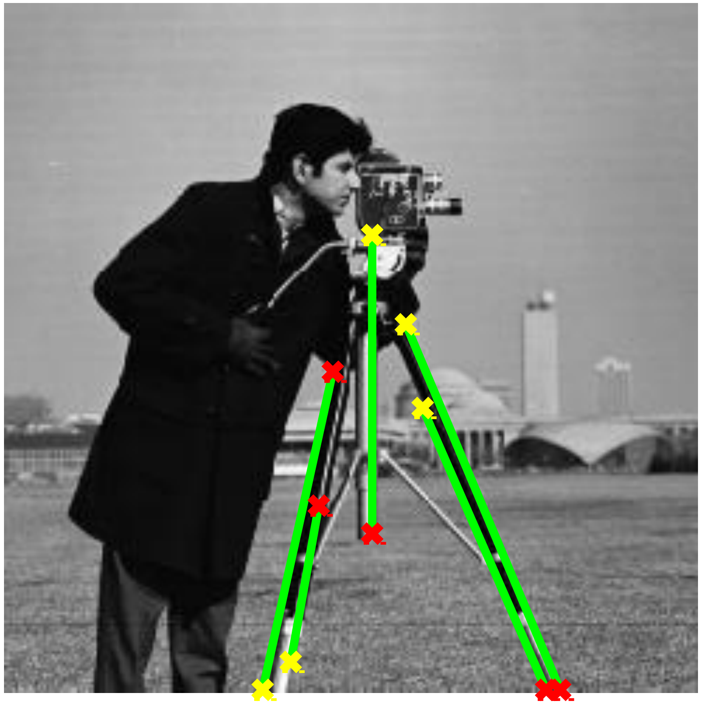


See ex_hough.m

Hough transform

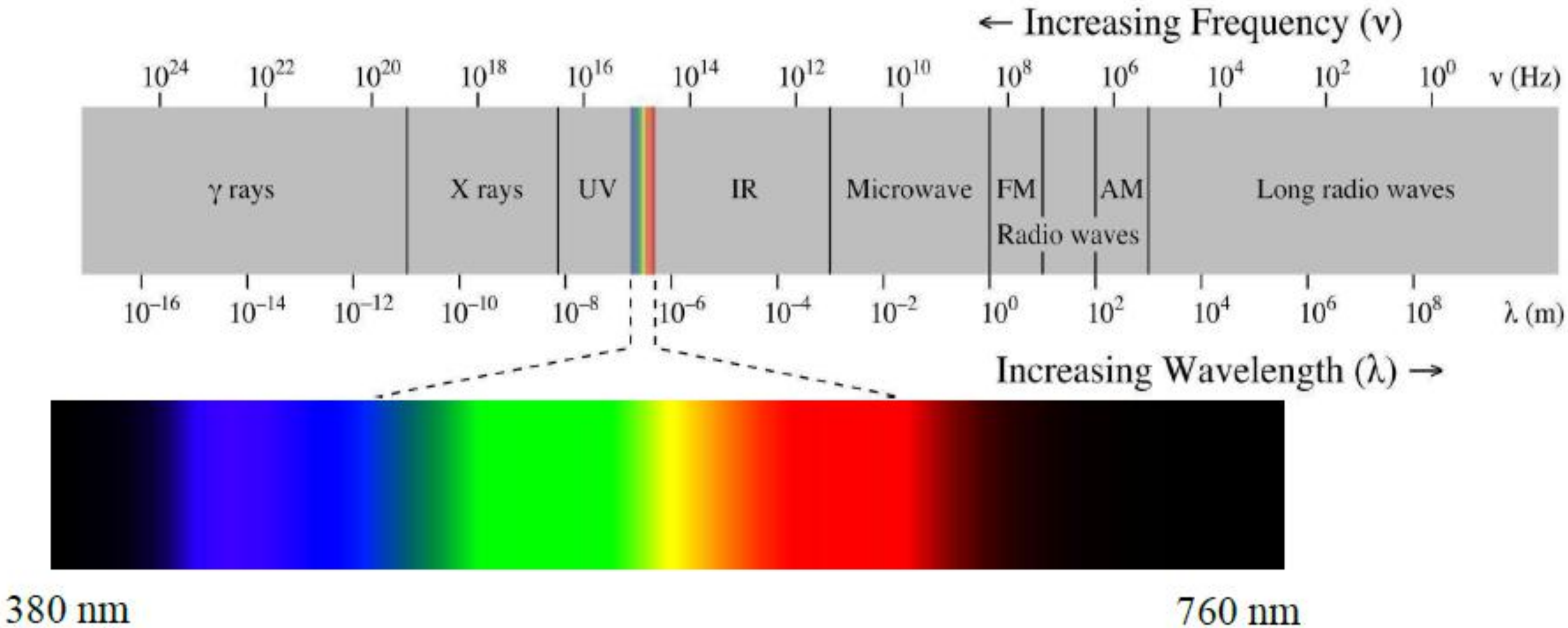
The straight lines $d = x\cos\theta + y\sin\theta$ are computed by using the peak values in Hough transform.

The line segments are computed by using the straight lines and the values in the edge map.



Color Fundamentals

- Color is a rich and complex experience, usually caused by the vision system responding differently to different wavelengths of light (wavelength in nm).



Color Fundamentals

- To be able to describe colors, we need to know how people respond to them.

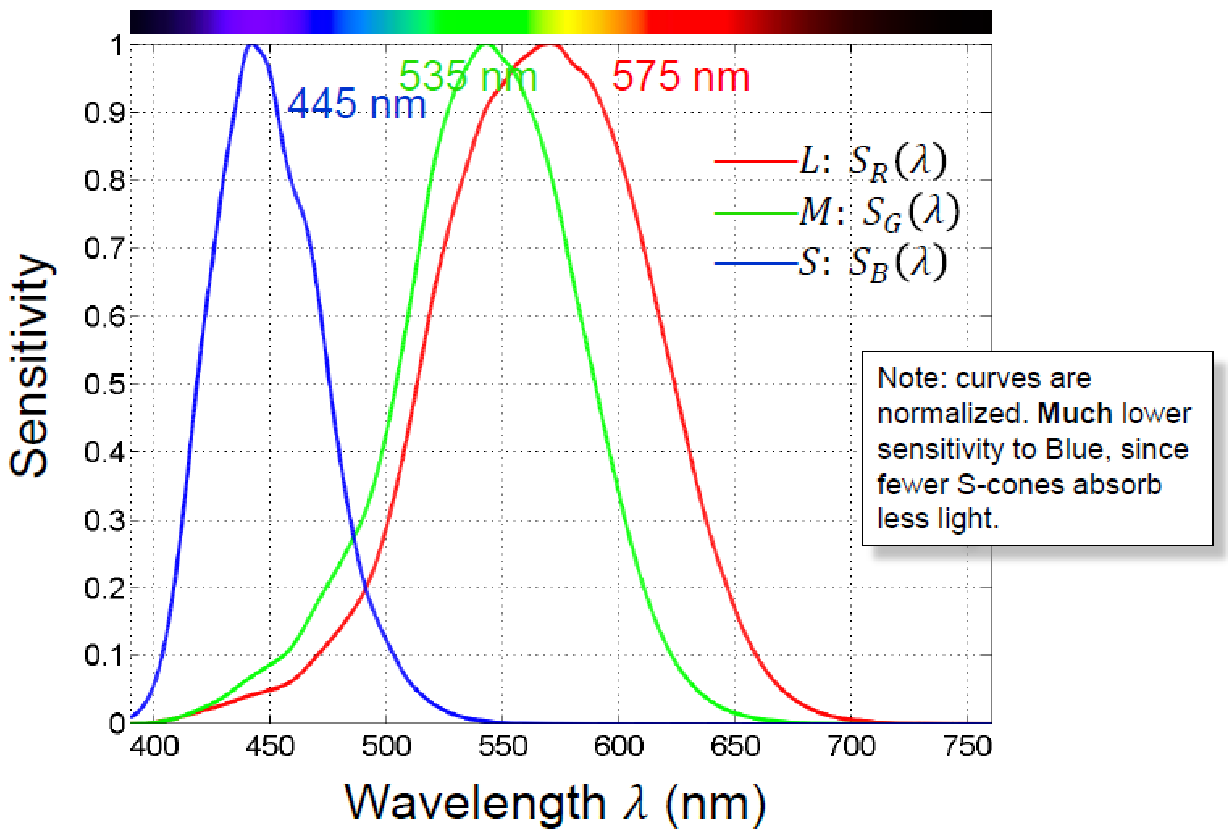
- The simplest question is to understand which spectral radiances produce the same response from people under simple viewing conditions.

- This yields a simple, linear theory of color matching which is accurate and extremely useful for describing colors.

Color Fundamentals

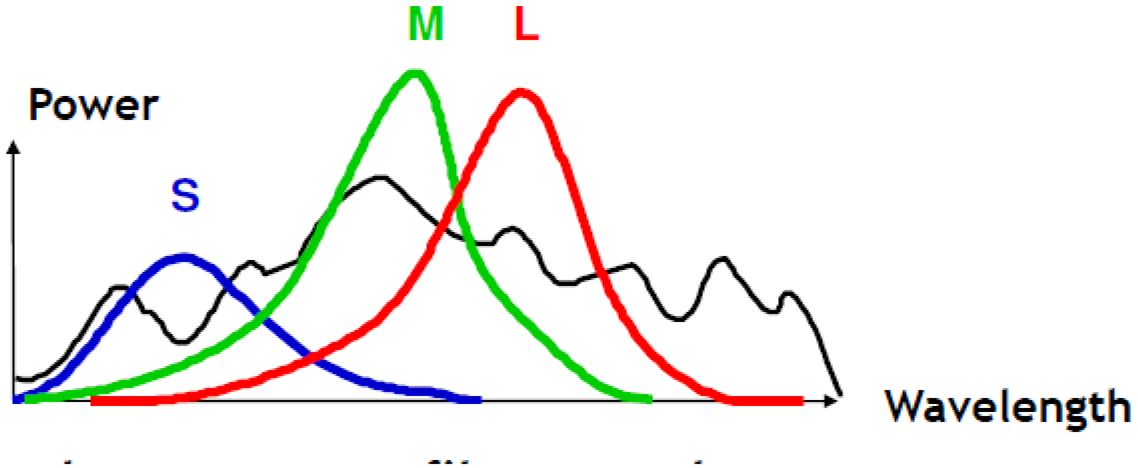
- Human retina:
  - Rods responsible for intensity, cones responsible for color
  - 6 to 7 million cones in the human eye can be divided into three principal sensing categories, corresponding roughly to red, green, and blue. 65%: red 33%: green 2%: blue (blue cones are the most sensitive)

Absorption of light in the cones of the human retina



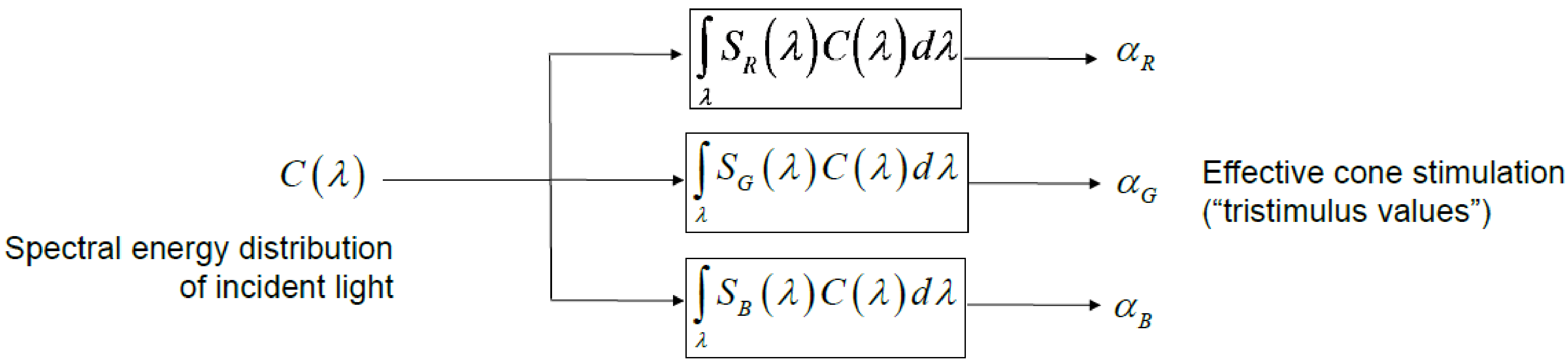
Color Fundamentals

- 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



Color Fundamentals

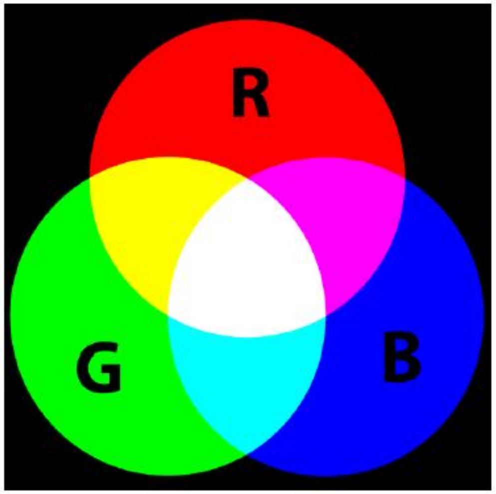
- <u>Three-receptor model of color perception</u>



- Different spectra can map into the same tristimulus values and hence look identical (“metamers”)
- Three numbers suffice to represent any color

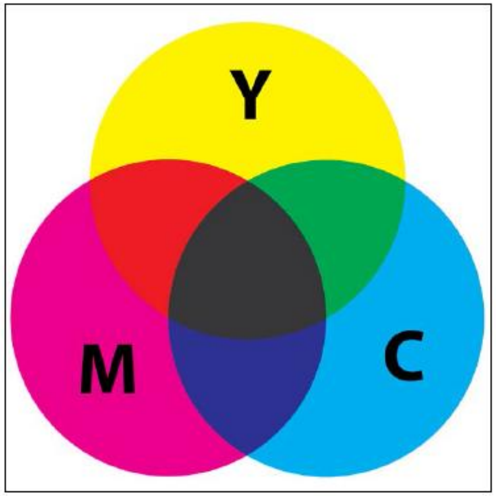
Additive vs. subtractive color mixing

Additive primaries



Mixtures of light

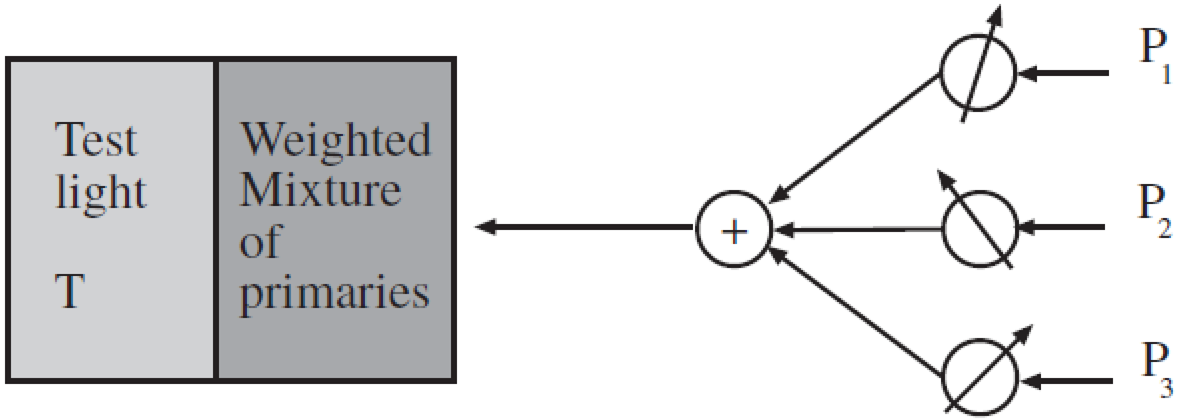
Subtractive primaries



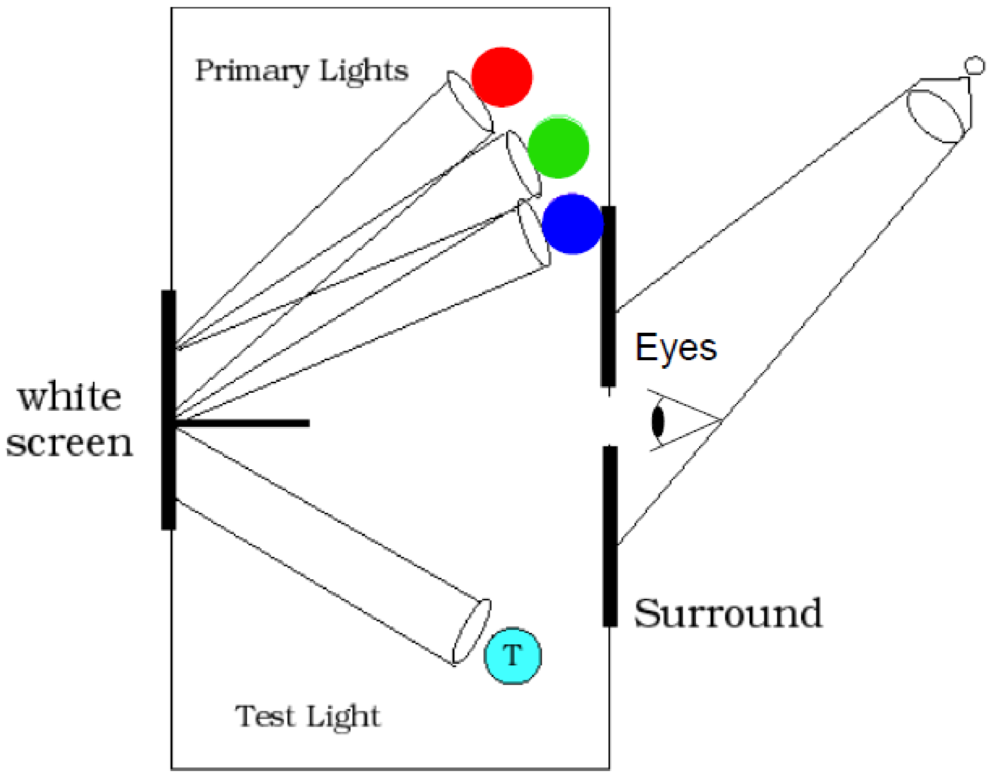
Mixtures of pigments

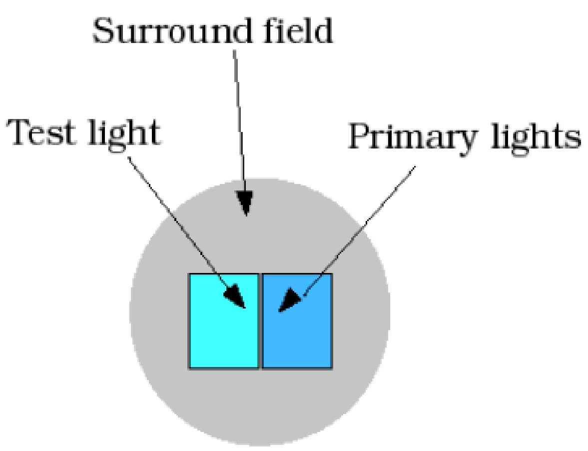
Color matching experiment

- Human perception of color can be studied by asking observers to mix colored lights to match a test light T, shown in a split field.
- The mixture of primaries P_1, P_2 and P_3 can be written as $w_1P_1+w_2P_2+w_3P_3$; if the mixture matches the test light, then we write $T = w_1P_1 + w_2P_2 + w_3P_3$.



Color matching experiment

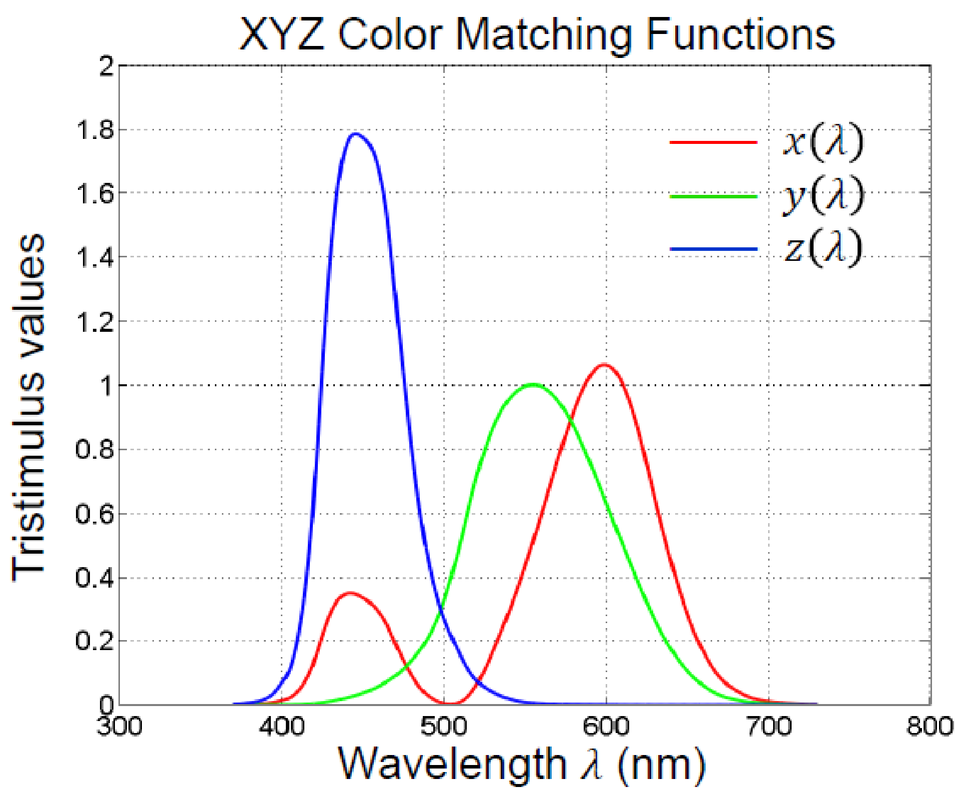




It is a matter of experimental fact that for most observers only three primaries are required to match a test light. This phenomenon is known as the the principle of trichromacy. It is often explained by assuming that there are three distinct types of color transducer in the eye.

Color matching functions

- If we have a record of the weight (w_1,w_2,w_3) of each primary (P_1,P_2,P_3) required to match a single wavelength source (a set of color matching functions) we can obtain the weights used to match



CIE XYZ color system (red, green, and blue are denoted X, Y, and Z, respectively)

Color spaces

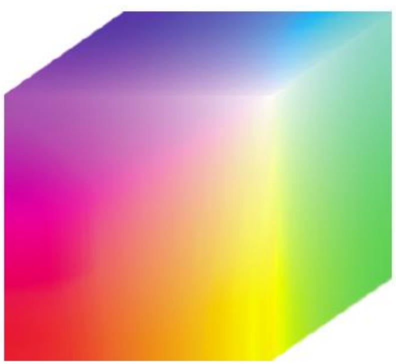
- Color spaces (color model) are normally invented for practical reasons, and so a wide variety exist.

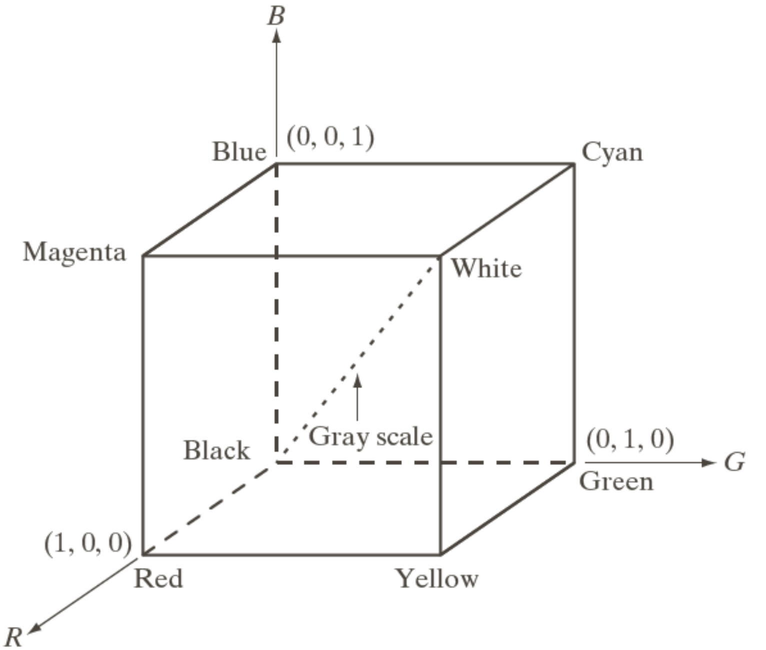
- Color spaces are oriented either towards hardware or towards applications: e.g.
  - RGB model is related to color monitors,
  - CMY model to color printers, and
  - HSI (HSV) model that corresponds closely with the way humans describe and interpret color.

Color space: RGB

- The RGB color space is a linear color space that formally uses single wavelength primaries (645.16 nm for R, 526.32nm for G and 444.44nm for B).
- Available colors are usually represented as a unit cube, usually called the RGB cube, whose edges represent the R, G, and B

The total number of colors in a 24-bit (pixel depth) RGB image is $(2^8)^3 = 16,777,216$





Color space: CMY

- In the CMY space, there are three primaries: cyan (a blue-green color); magenta (a purplish color) and yellow. These primaries should be thought of as subtracting a light primary from white light, whereas RGB are called additive primaries.
- Practical printing devices use at least four inks (cyan, magenta, yellow and black).

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

Color space: NTSC TV

- The YIQ model is used in NTSC TV broadcasting. Its major advantage is that guarantees downward compatibility with monochrome TV.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

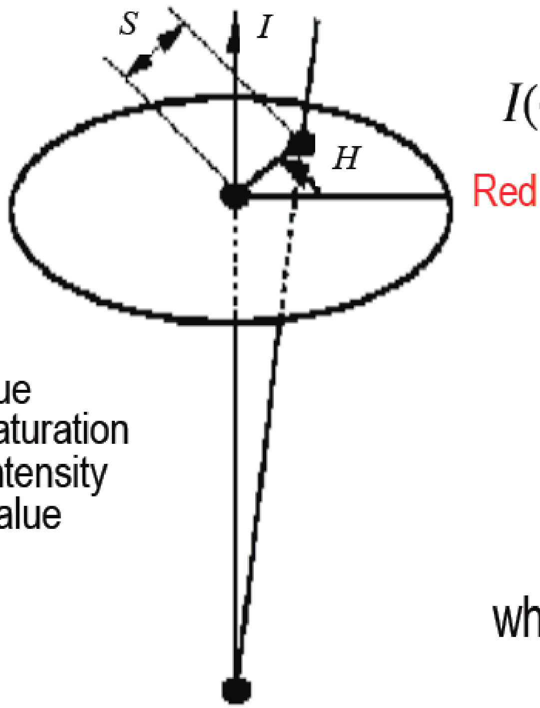
- Luminance is represented by the Y component. The I,Q components encode image chrominance.

Color space: HSI

- The coordinates of a color in a linear space may not necessarily encode properties that are common in language or are important in applications. Useful color terms include:
  - **Hue**, the property of a color that varies in passing from red to green;
  - **Saturation**, the property of a color that varies in passing from red to pink;
  - **Brightness** (sometimes called lightness or **value**), the property that varies in passing from black to white.
- A standard method for dealing with this problem is to construct a color space that reflects these relations by applying a non-linear transformation to the RGB space.

Color space: HSI

- Conversion from RGB to HSI



H - hue
S - saturation
I - intensity
V - value

$$I(\text{or } V) = \frac{1}{3}(R+G+B)$$

$$S = 1 - \frac{\min\{R,G,B\}}{I}$$

$$H = \begin{cases} \delta & \text{if } B < G \\ 360^\circ - \delta & \text{otherwise} \end{cases}$$

$$\text{where } \delta = \cos^{-1}\left(\frac{0.5\big((R-G)+(R-B)\big)}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right)$$

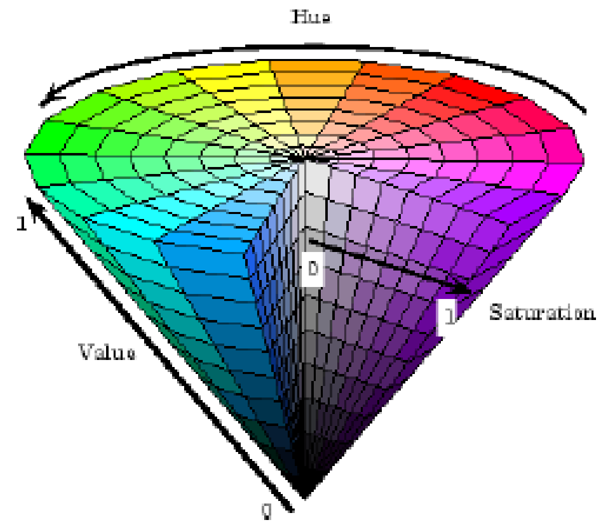


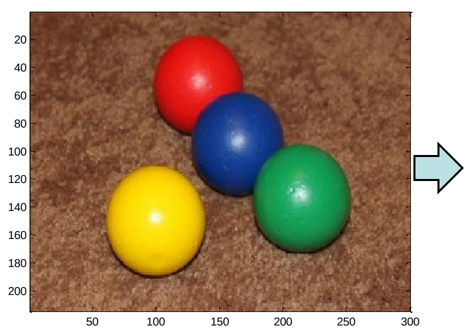
Image regions

- The shape of an object can be described either in terms of its boundary or in terms of the region it occupies.
- Region-based shape representation requires image segmentation in several homogeneous region. Image regions are expected to have homogenous characteristics (e.g. intensity, color) that are different in each region.
- Thus, *edge detection and region segmentation are dual (complementary) approaches in image analysis.*

Image segmentation based on color

- To segment an image based on color, it is natural to think first of the *HSV space*, because color is conveniently represented in the *hue channel*.
- The objective is to classify each pixel in a given image as having a color in the specified range or not.
- Coding this two sets of pixels in the image with black and white produces a binary segmented image.

Image segmentation based on color



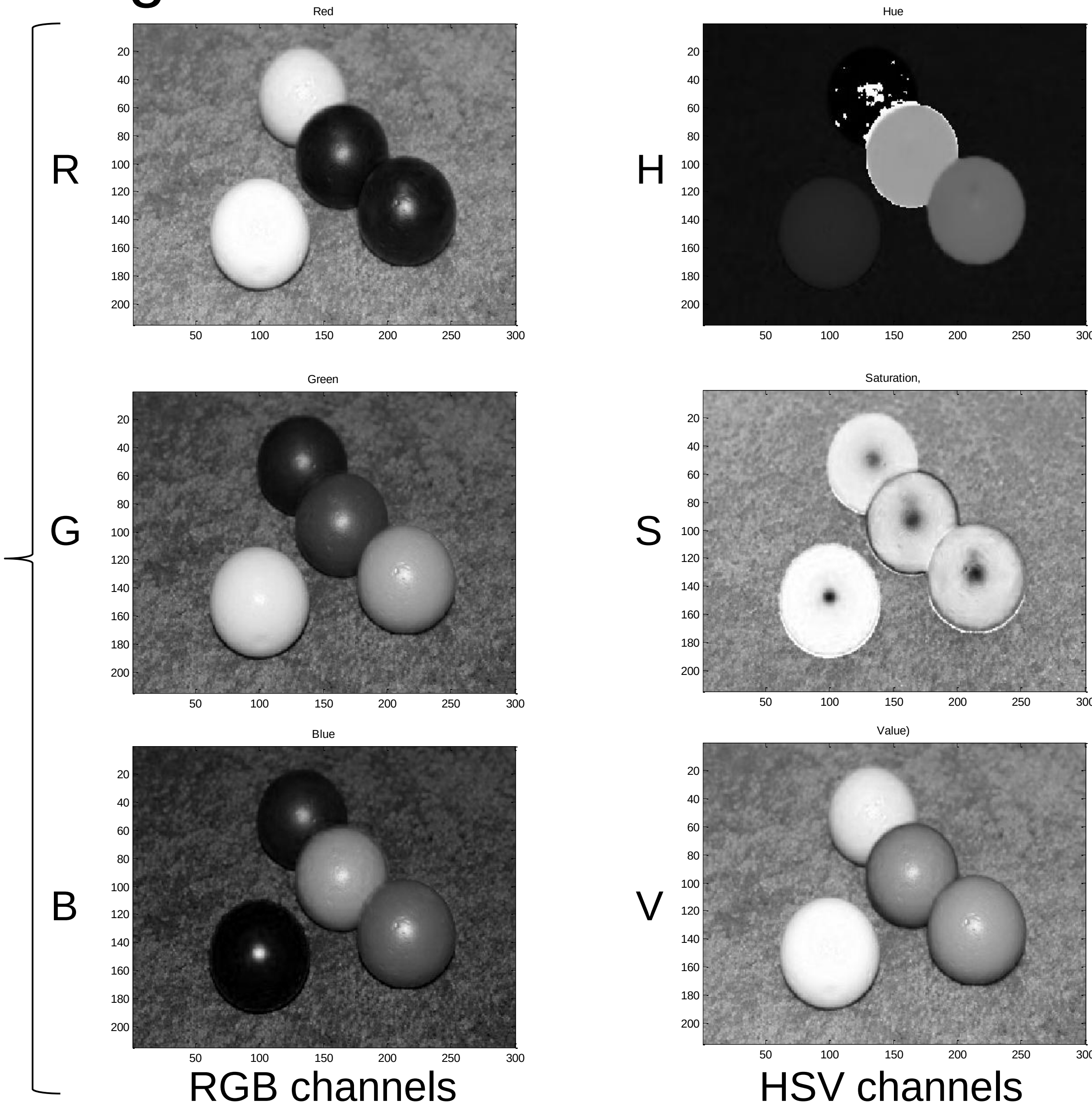
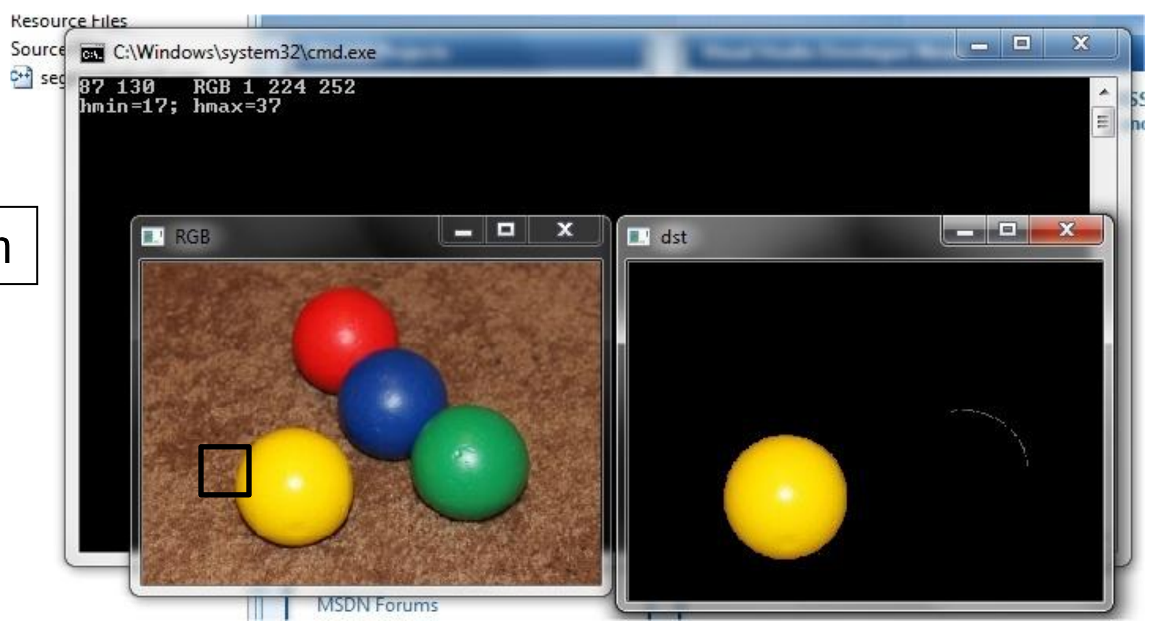


Image segmentation based on color

- We choose a rectangular region that contains samples of color we wish to segment out of the color image.
- We compute the mean h_m and the standard deviation h_{std} of the hue values of pixels contained within the rectangle.
- Then, we code each point as 1, if it is inside the range of hue values (e.g. $h_m \pm h_{std}$), otherwise as 0.

See ex_color_seg_prop.m



Geometrical properties of regions (blob analysis)

- Once the scene is segmented into **regions** (**blobs**), we can determine the geometrical properties of these regions.
- We assume that the blob is represented by a mxn binary sub-image $B(i,j)$.

- ***Area*** or size of the region is a total number of pixels occupied by the region:

$$A = \sum_{i=1}^{m} \sum_{j=1}^{n} B(i, j)$$

Geometrical properties of regions

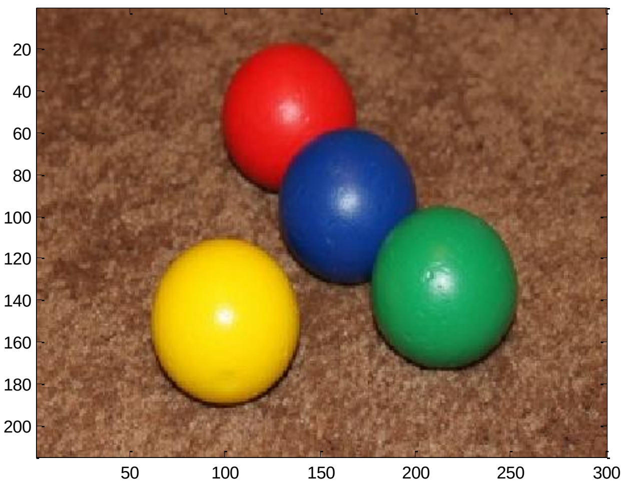
- The *centroid* (center of mass) is like the center of an arbitrary shaped region, and it can be used to represent the location of a region:

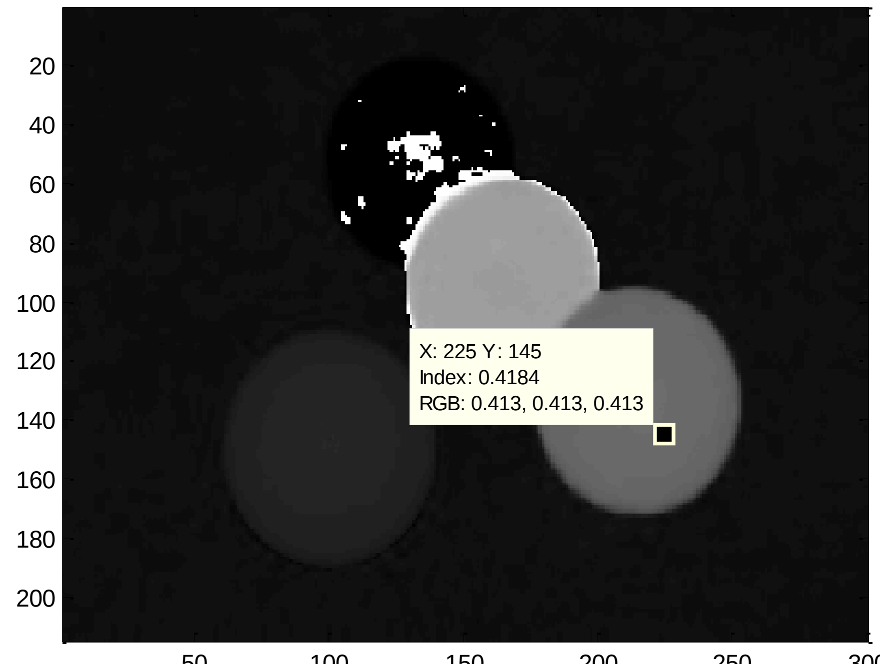
$$\bar{i} = \frac{1}{A} \sum_{i=1}^{m} \sum_{j=1}^{n} iB(i, j)$$

$$\bar{j} = \frac{1}{A} \sum_{i=1}^{m} \sum_{j=1}^{n} jB(i, j)$$

- The *perimeter* of a region is the sum of its border pixels. A pixel which has at least one pixel in its neighborhood from the background is called a border pixel.

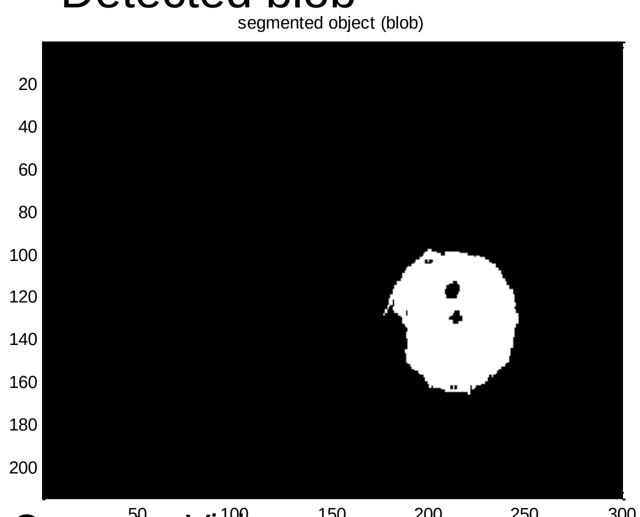
Image segmentation based on color: example

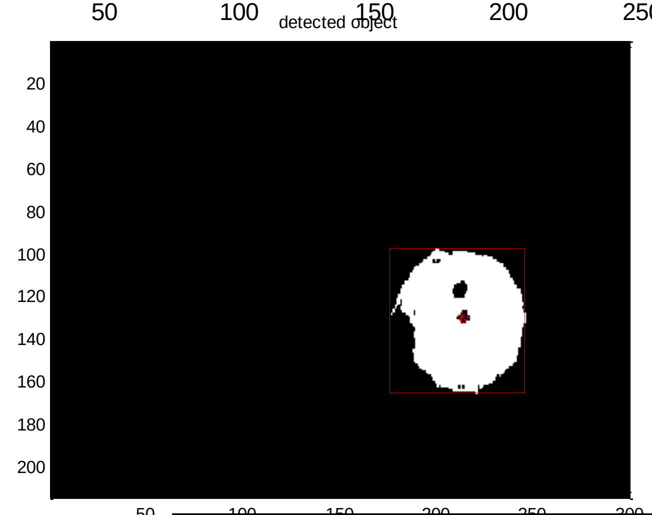




Hue component

Detected blob





Geometrical properties of the blob: centroid and boundingbox

See ex_color_seg_prop.m

Image segmentation: thresholds

- In the simplest case where the image contains one object, the grey level image can be converted into a corresponding binary image.
- The distribution of gray levels can be used to determine the threshold used in binary image.

- If the histogram has two spikes, it is called bimodal:

255	255	255	255	255	255
255	255	255	255	255	255
255	255	100	100	100	255
255	255	100	100	100	255
255	255	100	100	100	255
255	255	255	255	255	255

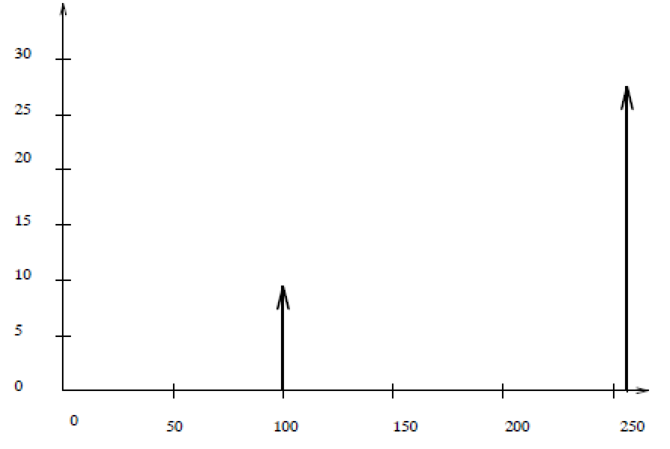
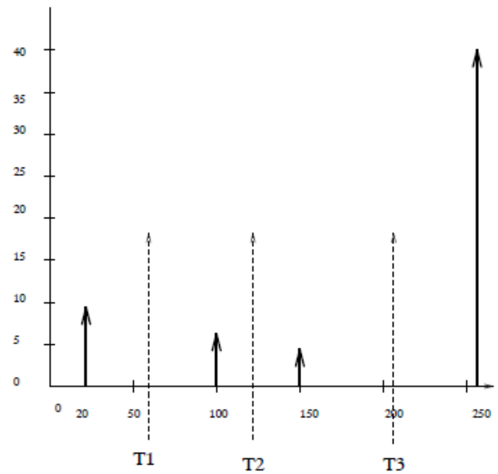


Image segmentation : thresholds

- The histogram of real image may consist of peaks and valleys (and also may contain small peaks due to noise, thus we have to smooth it).

255	255	255	255	255	255	255	20
255	255	255	100	100	255	20	20
255	255	255	100	100	255	20	20
255	255	255	100	100	255	20	20
255	255	255	255	255	255	20	20
255	255	255	255	255	255	255	255
150	150	255	255	255	255	255	255
150	150	255	255	255	255	255	255



- Objects/regions with approximately the same range of gray levels form a <u>class</u>.

Image segmentation : thresholds

- To distinguish between distinct classes of objects/regions, we have to choose some thresholds, each lying between two peaks.
- It is worth noting that the histogram can only distinguish objects by using gray level information, thus we have to use also spatial information of regions.

- In order to find a connected group of pixels in a image we need to apply a connected component algorithm.

Image segmentation: connected components algorithm

	4	
4	*	4
	4	

(a)

8	8	8
8	*	8
8	8	8

(b)

6	6	
6	*	6
	6	6

(c)

Pixel connectedness. (a) 4-connected. (b) 8-connected. (c) 6-connected.

1. Scan the binary image left to right, top to bottom.
2. If an unlabeled pixel has a value of '1', assign a new label to it according to the following rules:

$$\begin{array}{cc} & 0 \\ 0 & 1 \end{array} \rightarrow \begin{array}{cc} & 0 \\ 0 & \mathrm{L} \end{array} \qquad \begin{array}{cc} & 0 \\ \mathrm{L} & 1 \end{array} \rightarrow \begin{array}{cc} & 0 \\ \mathrm{L} & \mathrm{L} \end{array}$$

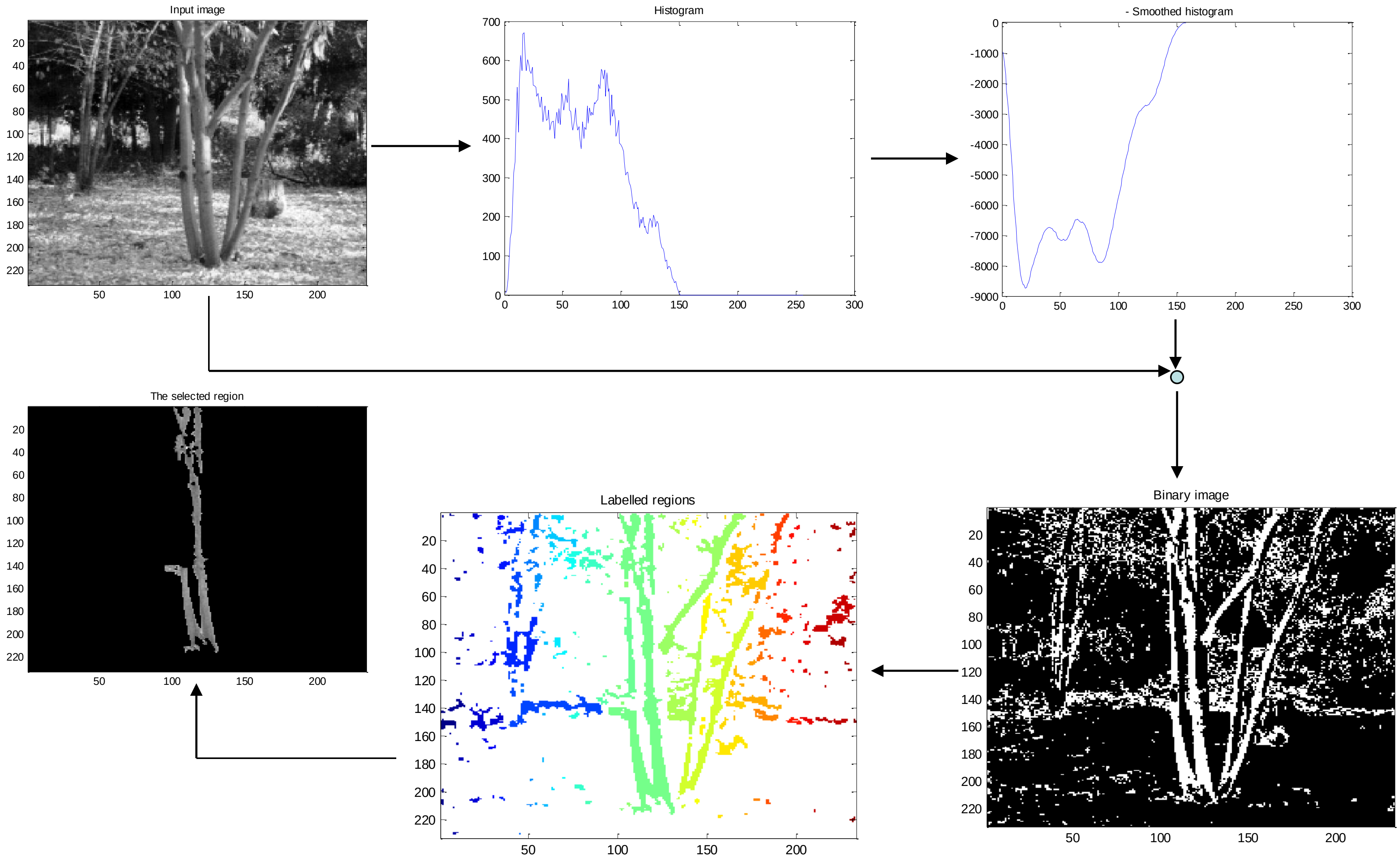
$$\begin{array}{cc} & \mathrm{L} \\ 0 & 1 \end{array} \rightarrow \begin{array}{cc} & \mathrm{L} \\ 0 & \mathrm{L} \end{array} \qquad \begin{array}{cc} & \mathrm{L} \\ \mathrm{M} & 1 \end{array} \rightarrow \begin{array}{cc} & \mathrm{L} \\ \mathrm{M} & \mathrm{L} \end{array} \quad (\text{Set } L = M).$$

3. Determine equivalence classes of labels.
4. In the second pass, assign the same label to all elements in an equivalence class.

Image segmentation: seed algorithm

- We summarize various steps in the seed segmentation algorithm:
  - Compute the histogram of the image;
  - Smooth the histogram to remove small peaks;
  - Identify peaks and valleys in the histogram;
  - Segment the image using thresholds at the valleys;
  - Apply connected component algorithm.

Image segmentation: example



See ex_reg.m

HOMEWORK

- Reading Assignments:
  - Forsyth and Ponce book: sections 6.1, 6.2, 6.3; 14.1; 15.1;
  - Mubarak Shah, “Fundamentals of Computer Vision”: chapter 3, from section 3.1 to section 3.6.1, sections 3.8 and 3.9; chapter 4, from section 4.1 to section 4.2.1.

- Next class: Wednesday, October 19