Computação Visual e Multimédia

10504: Mestrado em Engenharia Informática

Chap. 6 — Image Segmentation

Image Segmentation

Outline

...:

- Image segmentation: a reminder
- Image processing pipeline
- Image segmentation techniques
- Edge-based segmentation:
  - Basic algorithm
  - Hough transform
- Thresholding segmentation:
  - Global thresholding
  - Local thresholding

Image segmentation: a reminder

Definition:

- Segmentation is to subdivide an image into its component regions or objects.
- Segmentation should stop when the objects of interest have been isolated.

Goal:

- Segmentation subdivides an image into its constituent regions or objects that have similar descriptors according to a set of predefined criteria.

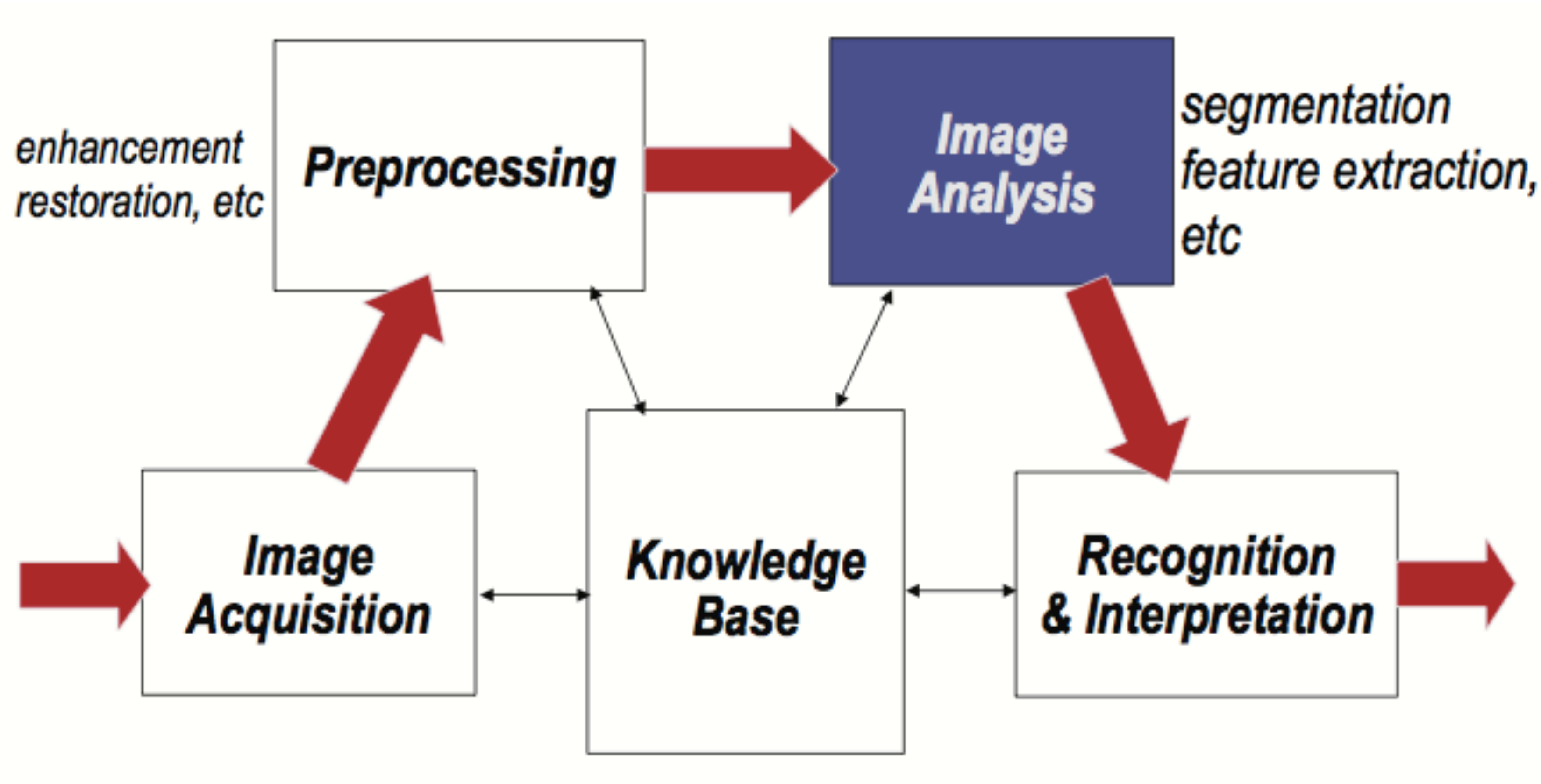
Descriptors?

- Intensity, histogram, mean, variance, energy, texture, *etc.*

Application Dependency:

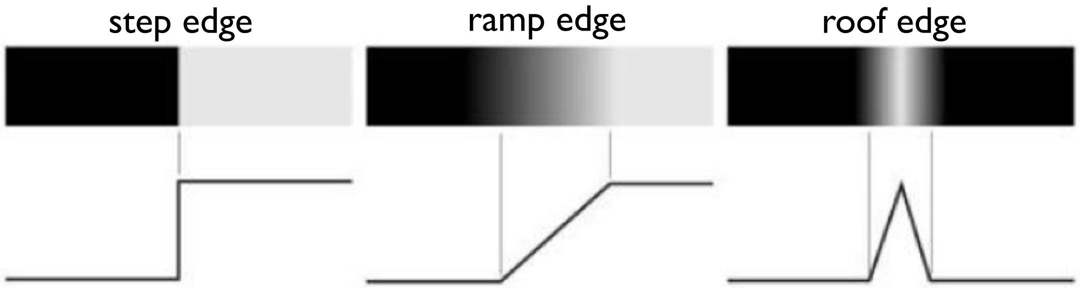
- Segmentation depends on applications or the problems to solve

Image processing pipeline (steps)



EDGE-BASED SEGMENTATION

Edge detection: another reminder



Usefulness:

– Edge detection is the common approach for detecting meaningful discontinuities.

Edge vs boundary:

– An edge is a set of connected pixels that lie on the boundary between two regions.

– An edge is a “local” concept whereas a region boundary is a more global idea.

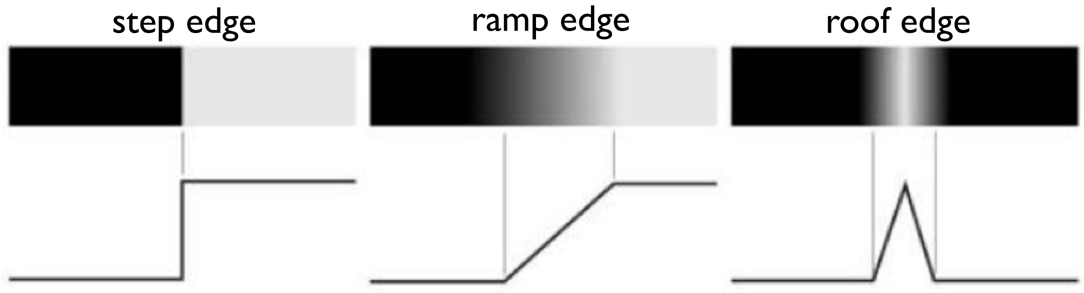
– A reasonable definition of “edge” requires the ability to measure gray level (intensity) transitions in a meaningful way.

Ideal edge model:

– An ideal edge has the properties of the step edge model shown above.

– An ideal edge according to this model is a set of connected pixels, each of which is located at an orthogonal step transition in gray level.

Edge detection (cont'd): another reminder



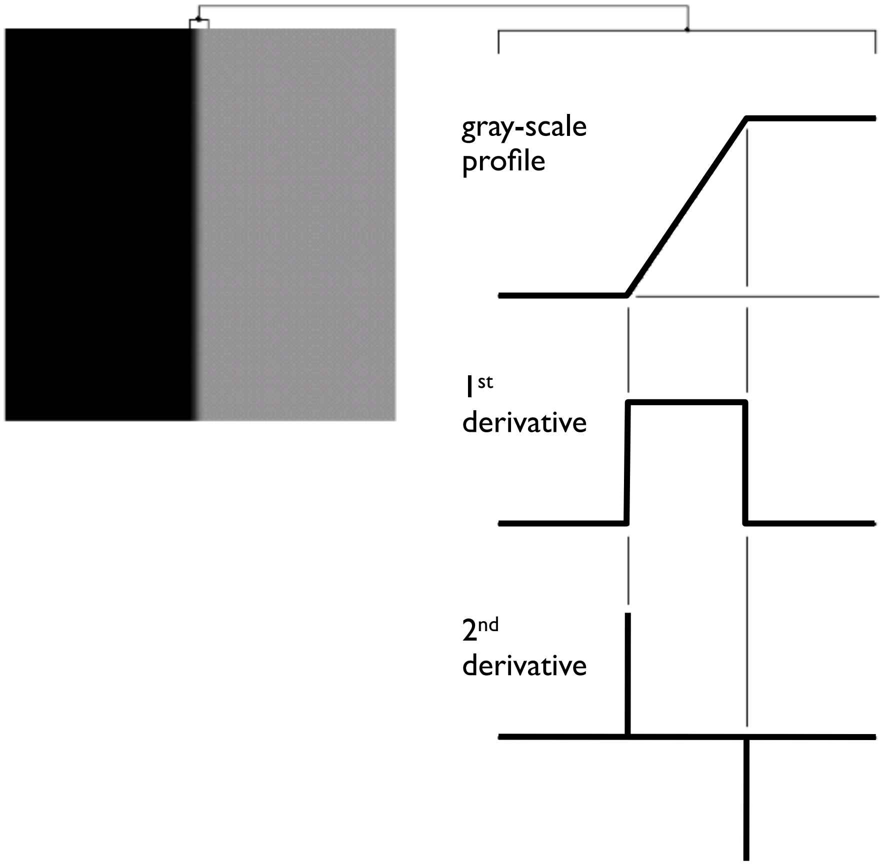
Practical edge model:

- In practice, optics, sampling, and other image acquisition imperfections yield edges that are blurred.
- Consequently, edges are more closely modeled as a "ramp"-like profile.
- The *degree of blurring* is determined by factors such as the quality of the image acquisition system, the sampling rate, and illumination conditions.
- An edge point is any point contained in the ramp, and an edge would be a set of such points that are connected.
- The "thickness" of the edge is determined by the length of the ramp.
- Blurred edges tend to be thick and sharp edges tend to be thin.

Edge detection (cont'd): still another reminder

Edge detection: essentials

- The magnitude of the first derivative can be used to detect the presence of an edge.
- The sign of the second derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge.
- An imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge (*zero-crossing property)*



Edge segments

Edge point:

- We define a point in an image as being an **edge point** if its two-dimensional first-order derivative is greater than a specified threshold.

Edge:

- A set of such points is defined as an **edge** if such pixels are connected according to a predefined criterion of connectedness.

Edge segment:

- The term **edge segment** generically is used if the edge is short in relation to the dimensions of the image.

Segmentation key problem:

How to link edge segments into longer edges?

edge detection → edge linking

Edge linking

linking edge segments into longer edges = *edge linking*

Basic edge linking algorithm:

- One of the simplest approaches for linking edge points is to analyze the characteristics of pixels in a small neighborhood.
- All points that are similar according to a set of predefined criteria are linked.
- Two principal properties (as criteria):
  - The strength of the gradient: $|\nabla f(x,y) - \nabla f(x',y')| \leq \varepsilon$
  - The direction of the gradient vector: $|\theta(x,y) - \theta(x',y')| \leq \tau$

Recall that the direction of the edge at (x,y) is perpendicular to the direction of the gradient vector at that point.

Edge linking: finding straight lines

Problem:

- Given *n* points in an image, suppose that we want to find subsets of these points that lie on straight lines.

A possible solution:

- One possible solution is to first find all lines determined by every pair of points,
- then find all subsets of points that are close to particular lines.
- But, this involves finding n(n-1)/2 lines and then performing *n*x*n*(*n*-1)/2 comparisons!

A better solution:

Hough Transform [*1962*]

Hough transform

Context:

- Performed after edge detection.
- Hough transform (1962): generalized template matching technique

Purpose:

- It is a technique to isolate the curves (e.g., straight lines, circles, parabolas, ellipses, etc) of a given shape / shapes in a given image.
- Requires that the curve be specified in some parametric form.

Generalization:

- Generalized HT when a simple analytic description of feature is not possible.

Advantages:

- The Hough Transform is tolerant of gaps in the edges
- It is relatively unaffected by noise
- It is also unaffected by occlusion in the image

Hough transform for straight lines

Problem:

– Fit a straight line (or curve) to a set of edge pixels.

Explicit form: $y = mx + b$

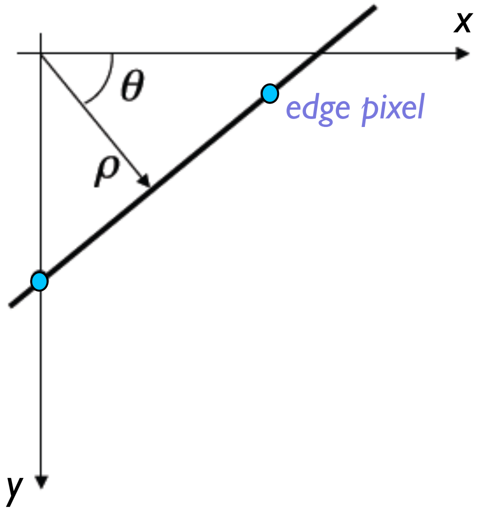
– This representation fails in case of vertical lines.

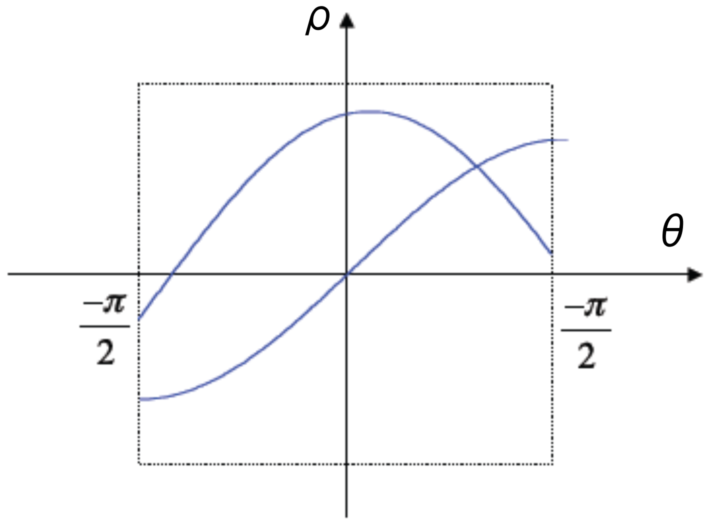
Normal form: $x\cos\theta + y\sin\theta = \rho$

– The *normal* for a given line is defined to be the shortest segment between the line and the origin.

– The normal form of the equation of a straight line is given by the expression above.

– This parameterization avoids infinite-slope problem.





Hough transform for straight lines (cont'd)

How does it work?:

– Points in the picture correspond to sinusoids in parameter space.

– Points in parameter space correspond to lines in the picture.

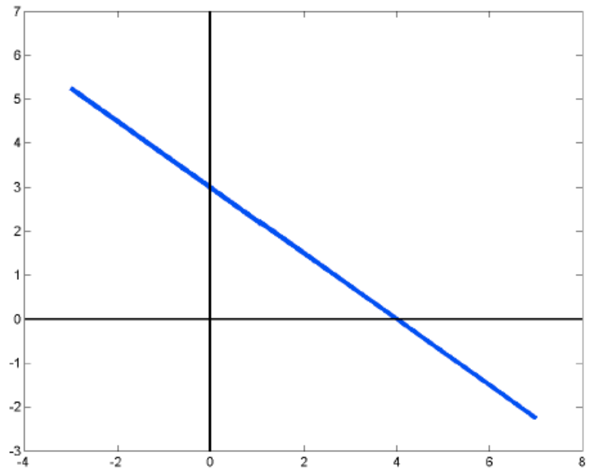
Sinusoids corresponding to co-linear points intersecting at an unique point

Example:

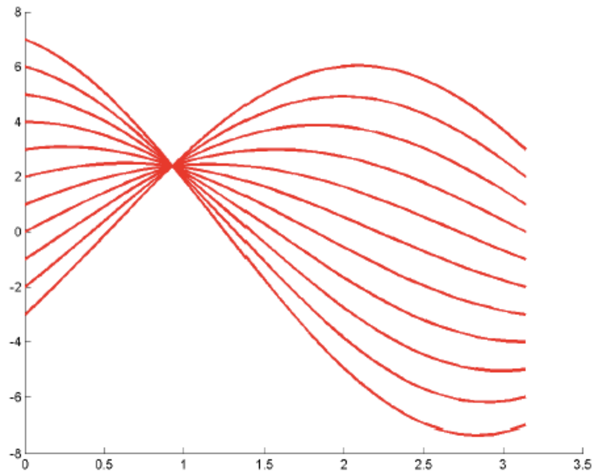
Line: $0.6x + 0.4y = 2.4$

Sinusoids intersect at:

$\rho = 2.4$ and $\theta = 0.9273$

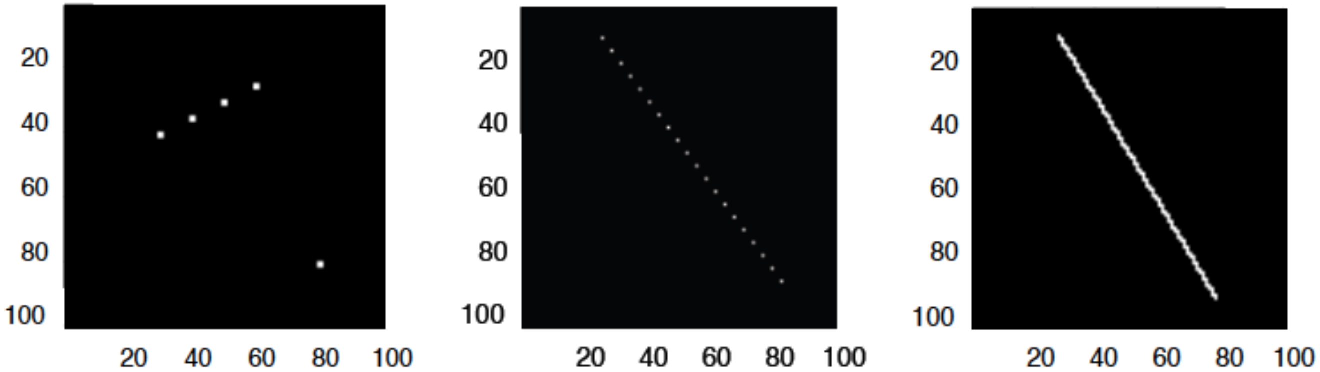


$y = mx + b$



$x\cos\theta + y\sin\theta = \rho$

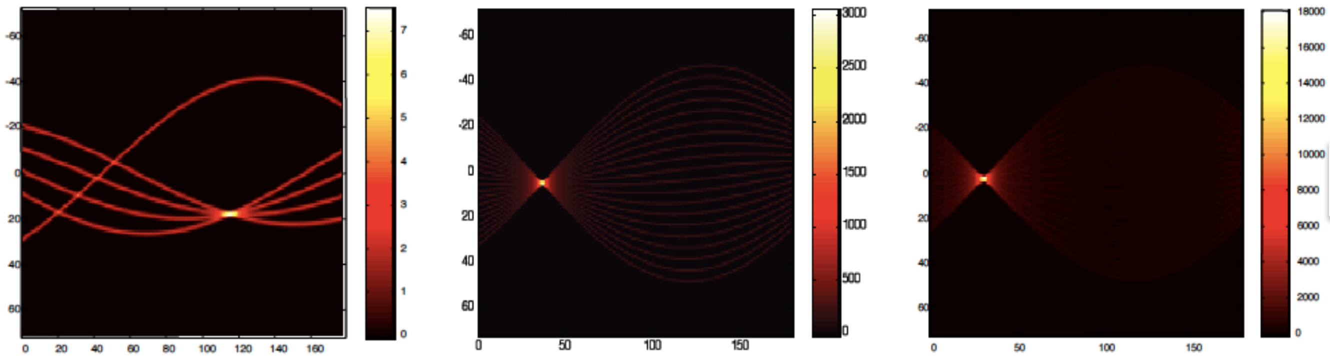
Hough transform for straight lines: *examples*



original image

How does it work (again)?:

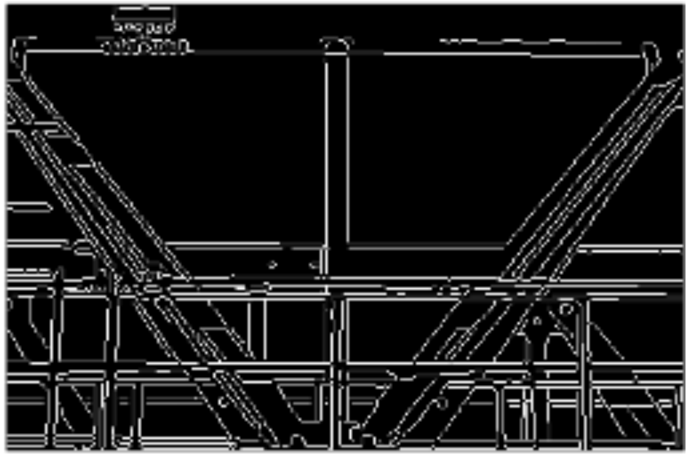
- The HT given by the normal form generates a parameter space matrix whose rows and columns correspond to those ρ and θ values, respectively.
- The value of each matrix element indicates how many points/pixel that lie on the line with the parameters (*rho, theta*).
- Therefore, the matrix element with the highest value tells us which line is most represented in the input image.



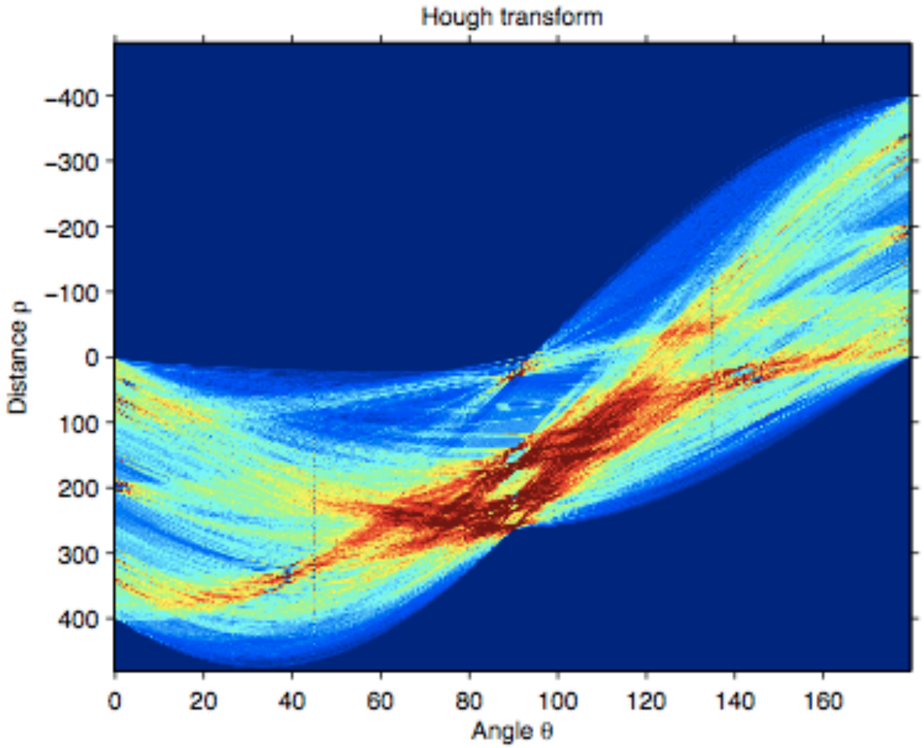
Hough Transform

Hough transform for straight lines: one more *example*

original image

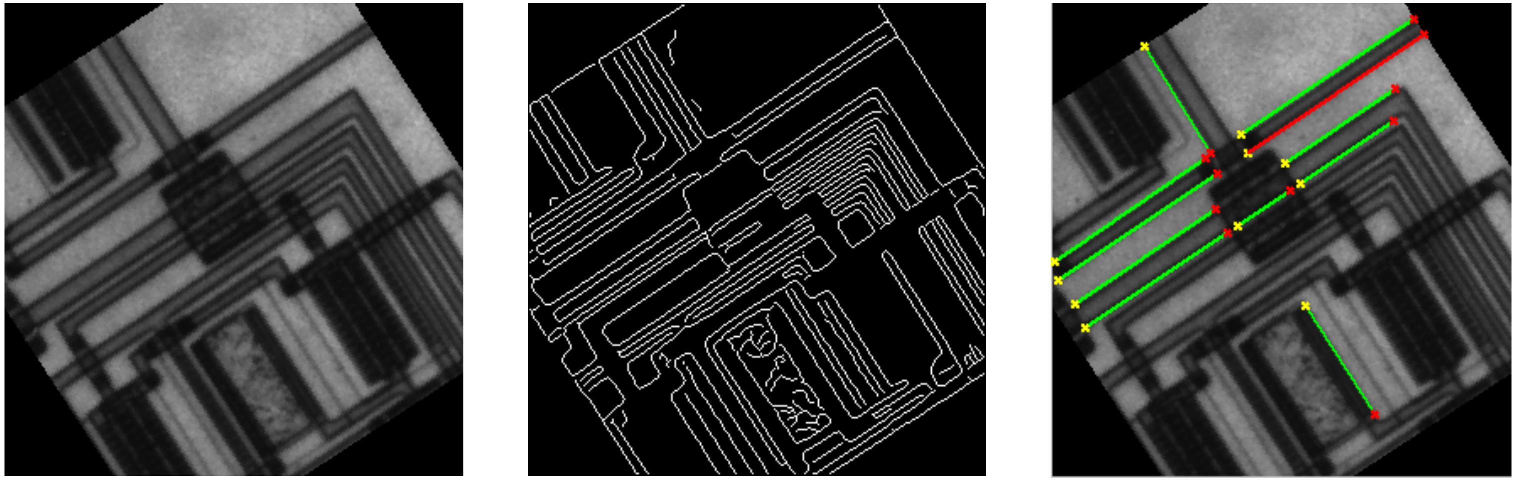


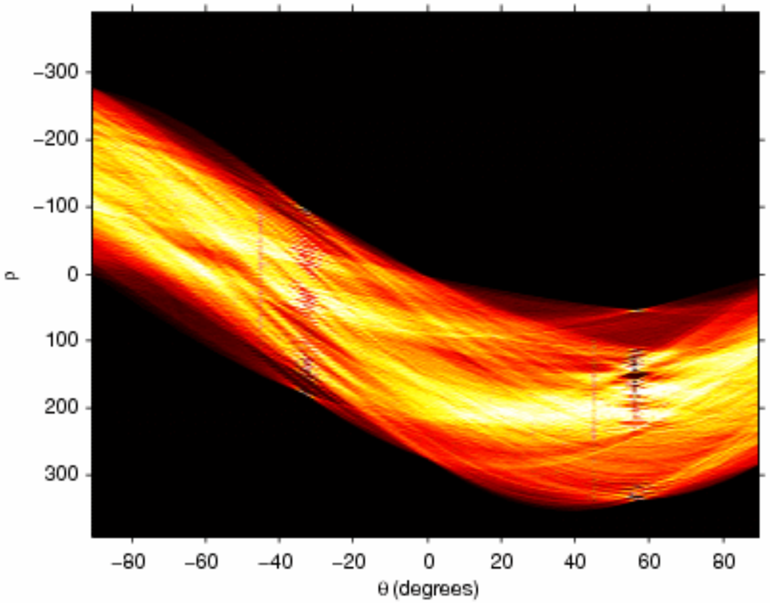
Canny edge detected image



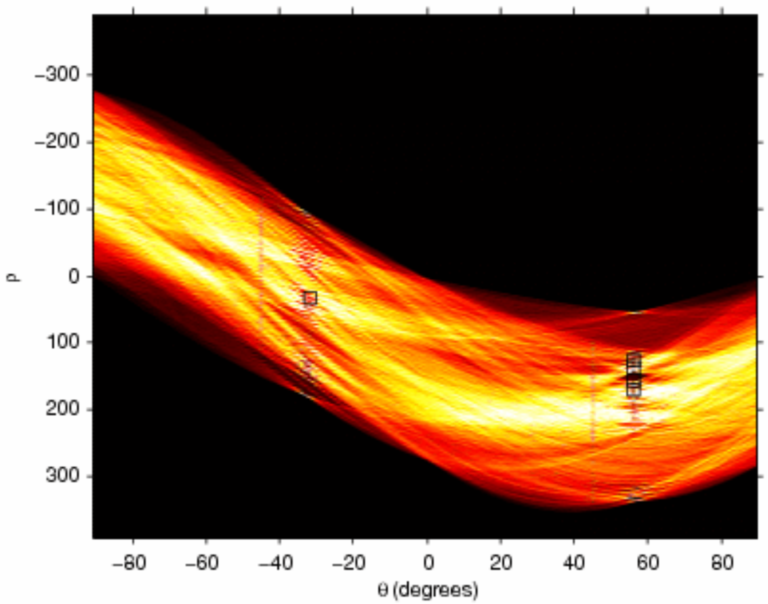
The Hough space of the original image: the dark red points are the points with the highest number of intersections. Many dark red points are around 90º, i.e., that the image has many horizontal lines.

Hough transform for straight lines: another still *example*





Hough transform.



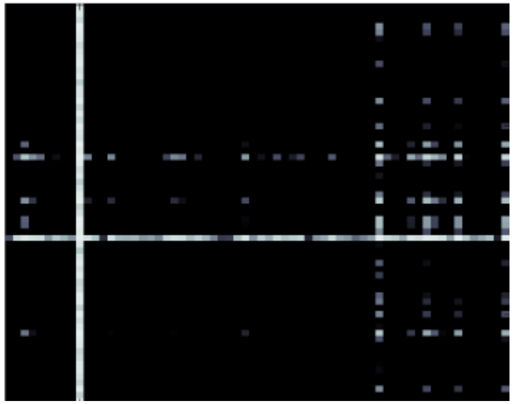
Hough transform with the peaks made visible.

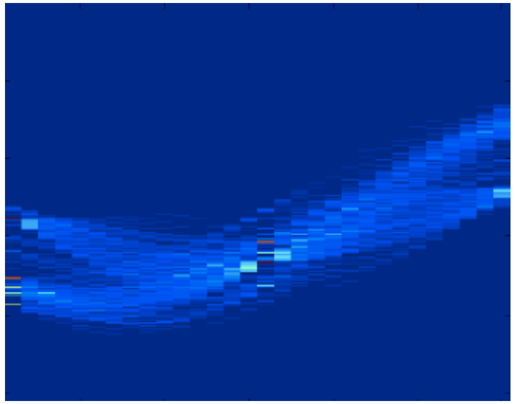
Hough transform algorithm

Outline:

- Voting in the parameter space (Hough space)
- Using directional information
- Error compensation: smoothing

Hough transform algorithm: *quantize parameter space and vote into bins*





Algorithm:

– Let $\rho \in [-D, D]$ and $\theta \in [0, \pi)$

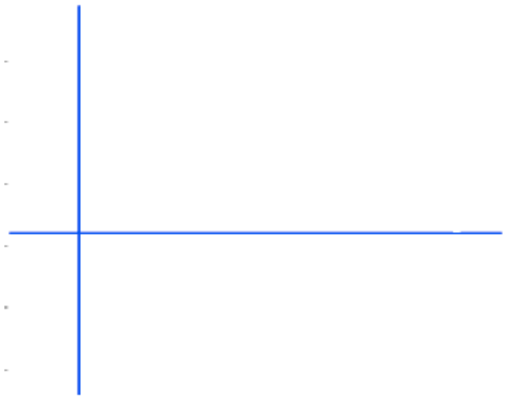
– For each edge point (x_i, y_i), compute:

$$\rho = x_i \cos\theta + y_i \sin\theta \qquad \forall \theta \in [0, \pi)$$

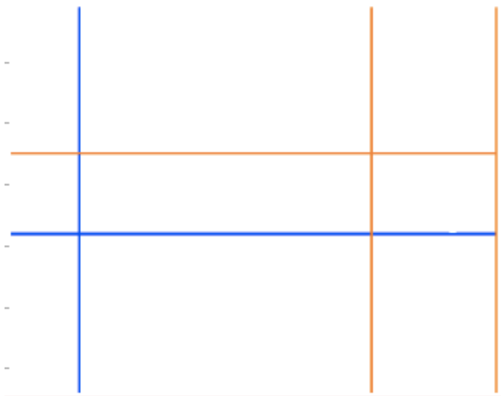
– Accumulator:

$$A(\rho,\theta) = A(\rho,\theta) + 1$$

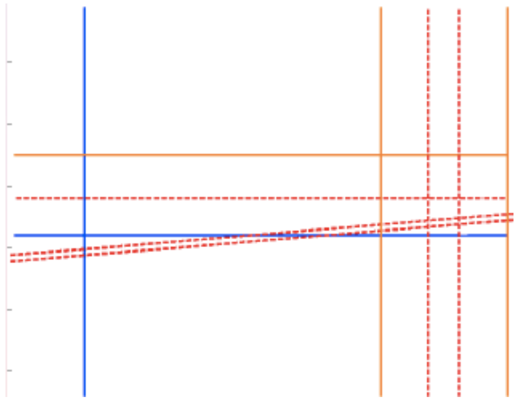
– Threshold the accumulator values to get parameters for detected lines



threshold at $A(\rho,\theta) = 30$



threshold at $A(\rho,\theta) = 20$



threshold at $A(\rho,\theta) = 15$

Hough transform algorithm reformulated: *using gradient direction to reduce computation load*

Let us then reformulate:

- Let $\rho \in [-D, D]$ and $\theta \in [0, \pi)$
- For each edge point (x_i,y_i), compute:

$$\theta = \tan^{-1}\left(\frac{G_Y}{G_X}\right)$$

$$\rho = x_i \cos\theta + y_i \sin\theta$$

$$A(\rho,\theta) = A(\rho,\theta) + 1$$

- For each bin of the accumulator A(ρ, θ) :
  - Search for the maximum value of A(ρ, θ)
  - Calculate the equation of the line
- To reduce the effect of noise more than one element (elements in a neighborhood) in the accumulator array are increased.

Hough transform: recap

In detecting lines:

- The parameters ρ and θ were found relative to the origin (0,0).
- We had *a priori* knowledge of the shape.

Generalization:

- The idea can be extended to shapes like ellipses, parabolas, etc:

Analytic Form	*Parameters*	*Equation*
Line	ρ, θ	$\cos\theta + y\sin\theta = \rho$
Circle	x_0, y_0, ρ	$(x-x_o)^2+(y-y_0)^2=r^2$
Parabola	x_0, y_0, ρ, θ	$(y-y_0)^2=4\rho(x-x_o)$
Ellipse	x_0, y_0, a, b, θ	$(x-x_o)^2/a^2+(y-y_0)^2/b^2=1$

A classification for image segmentation techniques

Taxonomy:

- Intensity-based segmentation: Thresholding
- Edge-based segmentation
- Region-based segmentation

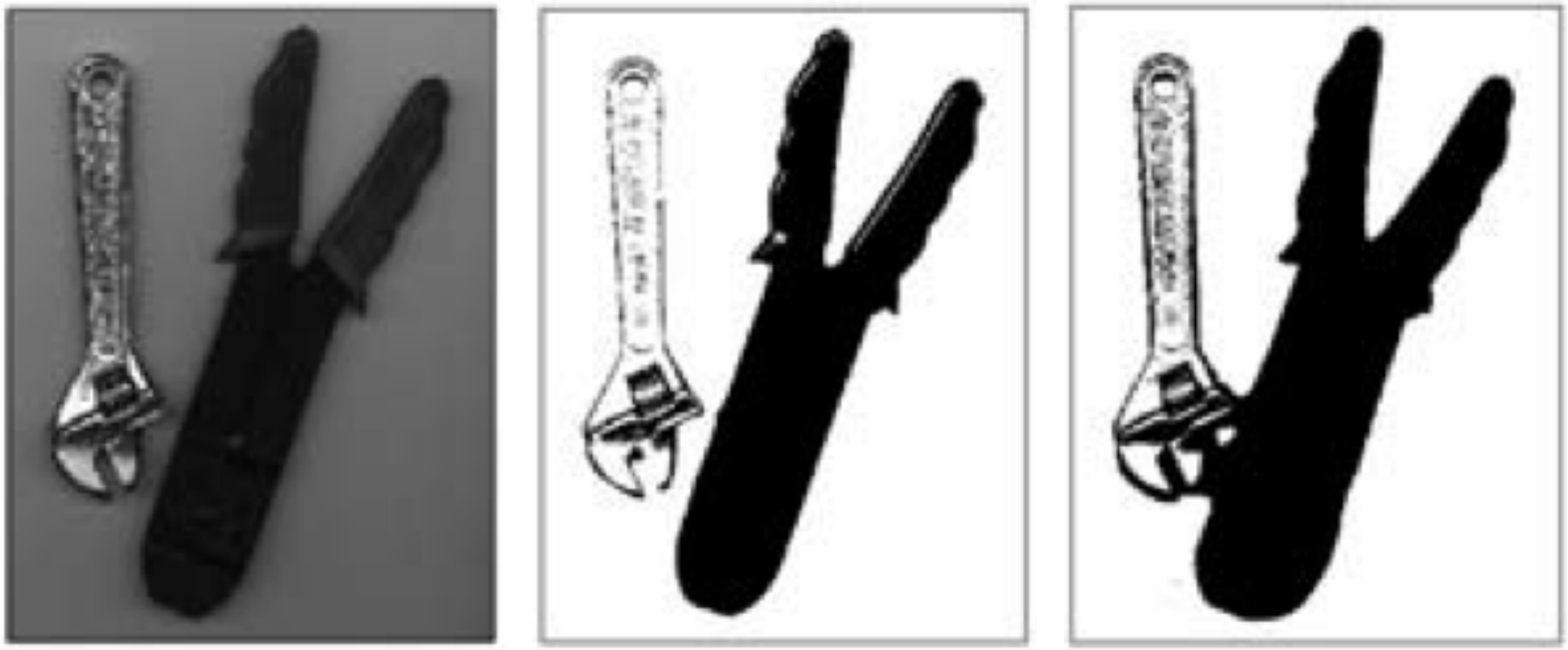
THRESHOLDING

(or intensity-based segmentation)

Assumptions for thresholding

Assumptions:

- The intensity values are different in different regions.
- Within each region, which represents the corresponding object in a scene, the intensity values are similar.



original image *threshold=50* *threshold=75*

Thresholding

How does it work?:

- Thresholding is usually the first step in any segmentation approach.
- Image thresholding classifies pixels into two categories:
  - Those to which some property measured from the image falls below a threshold *T*,
  - and those at which the property equals or exceeds a threshold *T*.
- Thresholding creates a binary image : binarization.

Two big categories:

- **Global** (or fixed) thresholding. The threshold value is held constant throughout the image:

$$g(x,y) = \begin{cases} 0 & f(x,y) < T \\ 1 & f(x,y) \geq T \end{cases}$$

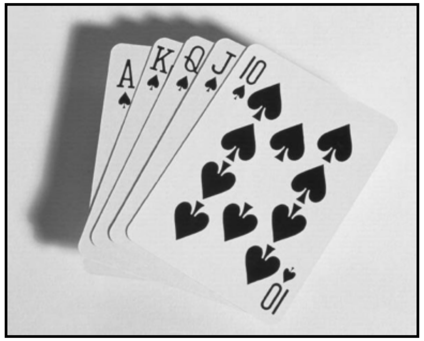
- **Local** (or dynamic or adaptive) thresholding depends on the position in the image. The image is divided into overlapping sections which are thresholded one by one.

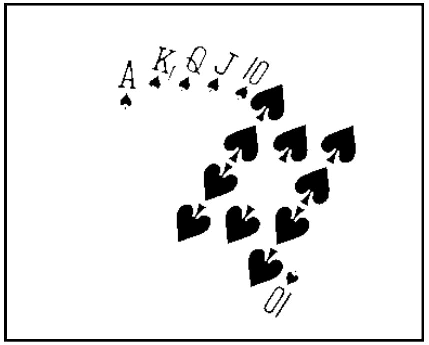
Thresholding: *example*

Example:

- Imagine a poker playing robot that needs to visually interpret the cards in its hand:

original image

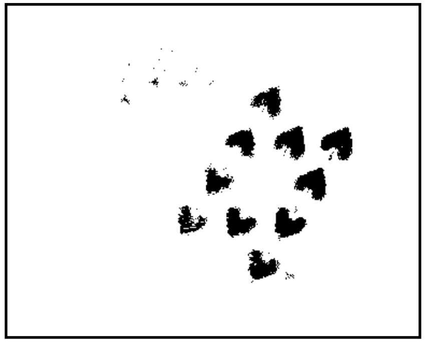


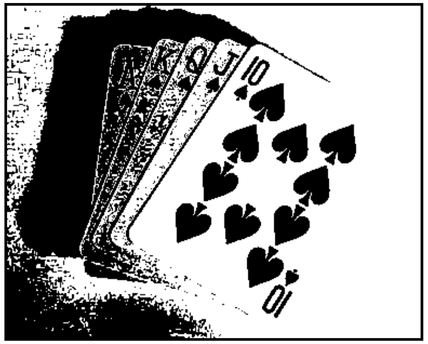


thresholded image

- If you get the threshold wrong the results can be disastrous:

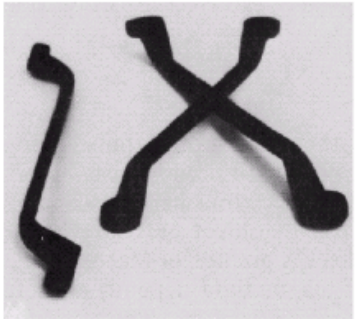
threshold too low

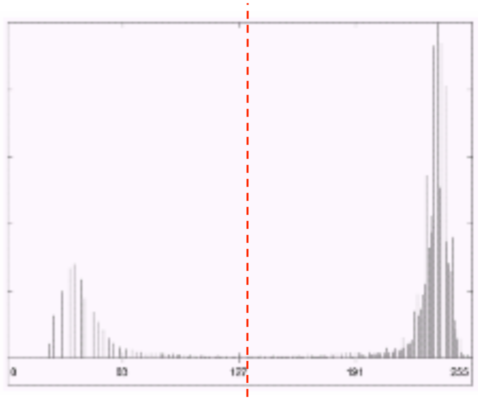


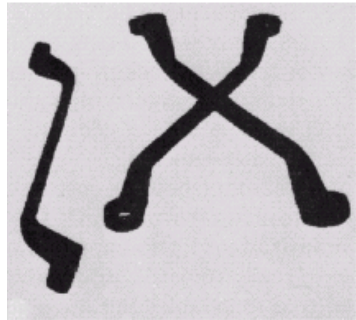


threshold too high

Basic global thresholding







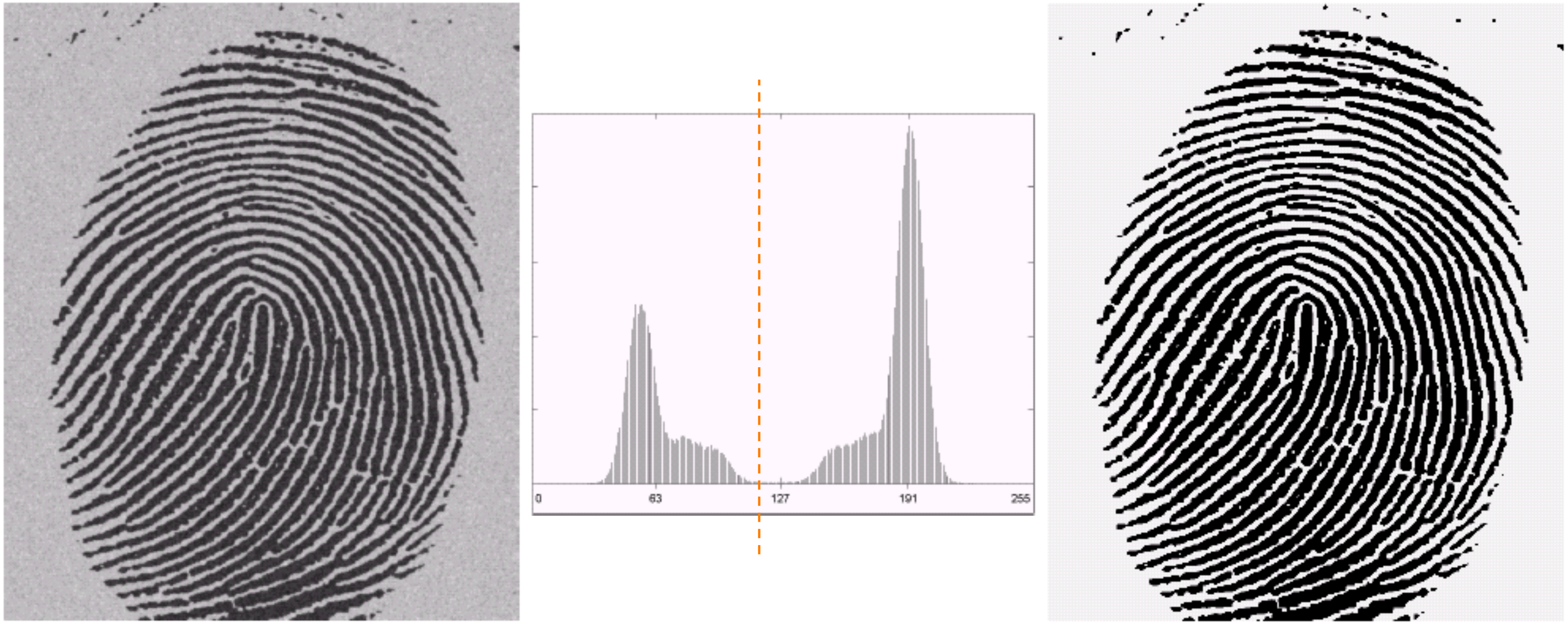
How does it work?:

- It is based on the histogram of an image.
- Partition the image histogram using a single global threshold. The success of this technique very strongly depends on how well the histogram can be partitioned.

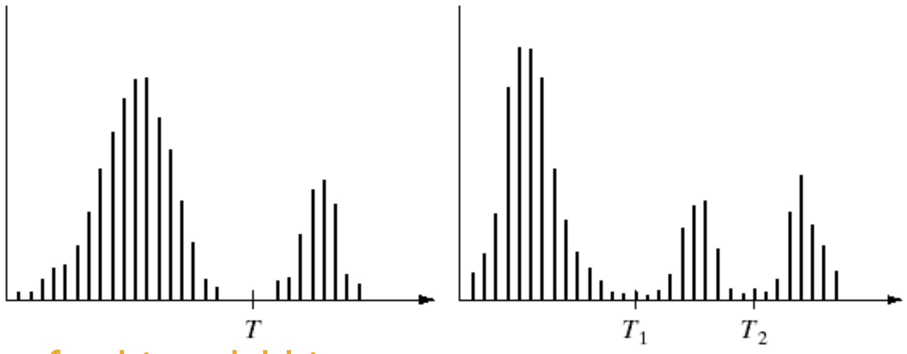
Algorithm (to obtain *T* automatically):

- Step 1: Select an initial estimate for T (typically the average grey level in the image)
- Step 2: Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels ≤ T
- Step 3: Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2
- Step 4: Compute a new threshold value: $$T = \frac{\mu_1 + \mu_2}{2}$$
- Step 5: Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit T_∞.

Basic global thresholding: *example*

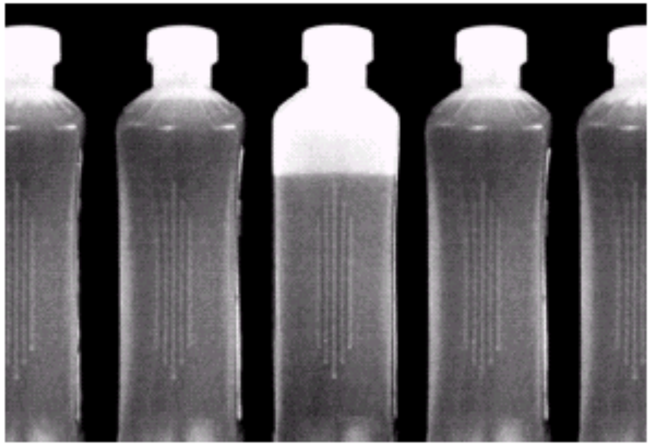


Problems with single value thresholding



Problem 1 (single thresholding):

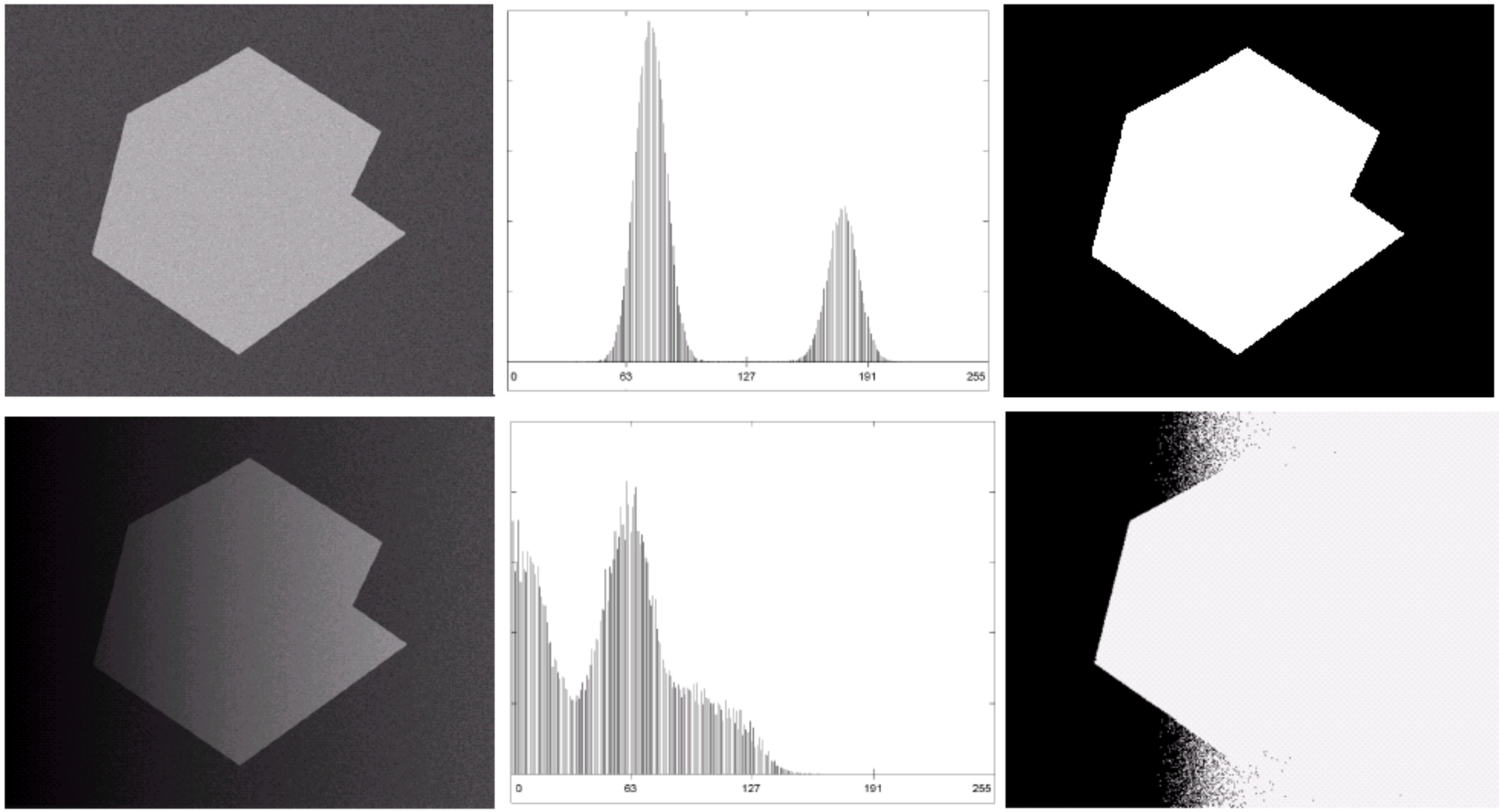
- Single value thresholding only works for bimodal histograms.
- Images with other kinds of histograms need more than a single threshold.
- Example:
  - Let's say we want to isolate the contents of the bottles shown below. Think about what the histogram for this image would look like!?
  - What would happen if we used a single threshold value?



Problems with single value thresholding (cont'd)

Problem 2 (illumination):

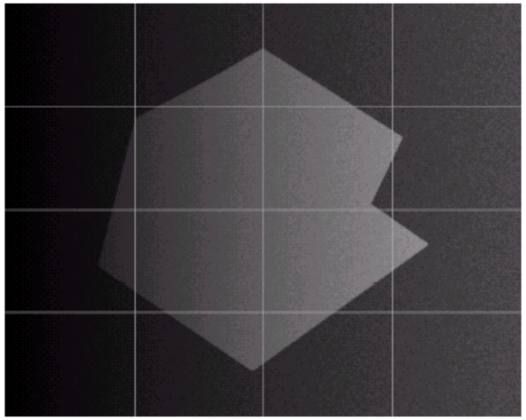
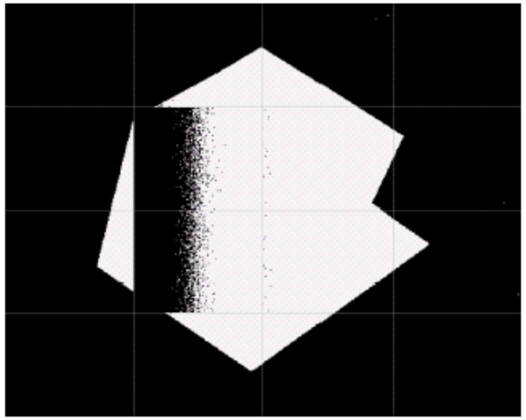
- Uneven illumination can really upset a single valued thresholding scheme.
- Example:



Basic local thresholding (also called dynamic or adaptive thresholding)

Context:

- An approach to handling situations in which single value thresholding will not work is to divide an image into sub images and threshold these individually
- Since the threshold for each pixel depends on its location within an image this technique is said to *adaptive*.
- Example:
  - The image shown in previous overhead now using adaptive thresholding.
  - The success is mixed, but the result can be imporved by further subdividing the troublesome sub images.

Summary:

…:

- In this lecture we overview the image processing pipeline.
- We reminded the edge detection techniques.
- We overview the segmentation techniques.
- We approached the edge-based segmentation technique, with a particular focus on Hough transform.
- We saw the basic global thresholding algorithm and its shortcomings.
- We also saw a simple way to overcome some of these limitations using local (adaptive) thresholding.