Image Segmentation

Preview

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

Principal approaches

- Segmentation algorithms generally are based on one of 2 basis properties of intensity values
  - ***discontinuity***: to partition an image based on sharp changes in intensity (such as edges)
  - ***similarity***: to partition an image into regions that are similar according to a set of predefined criteria.

Detection of Discontinuities

- Detect the three basic types of gray-level discontinuities
  - points , lines , edges
- The common way is to run a mask through the image

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Point Detection

- A point has been detected at the location on which the mask is centered if

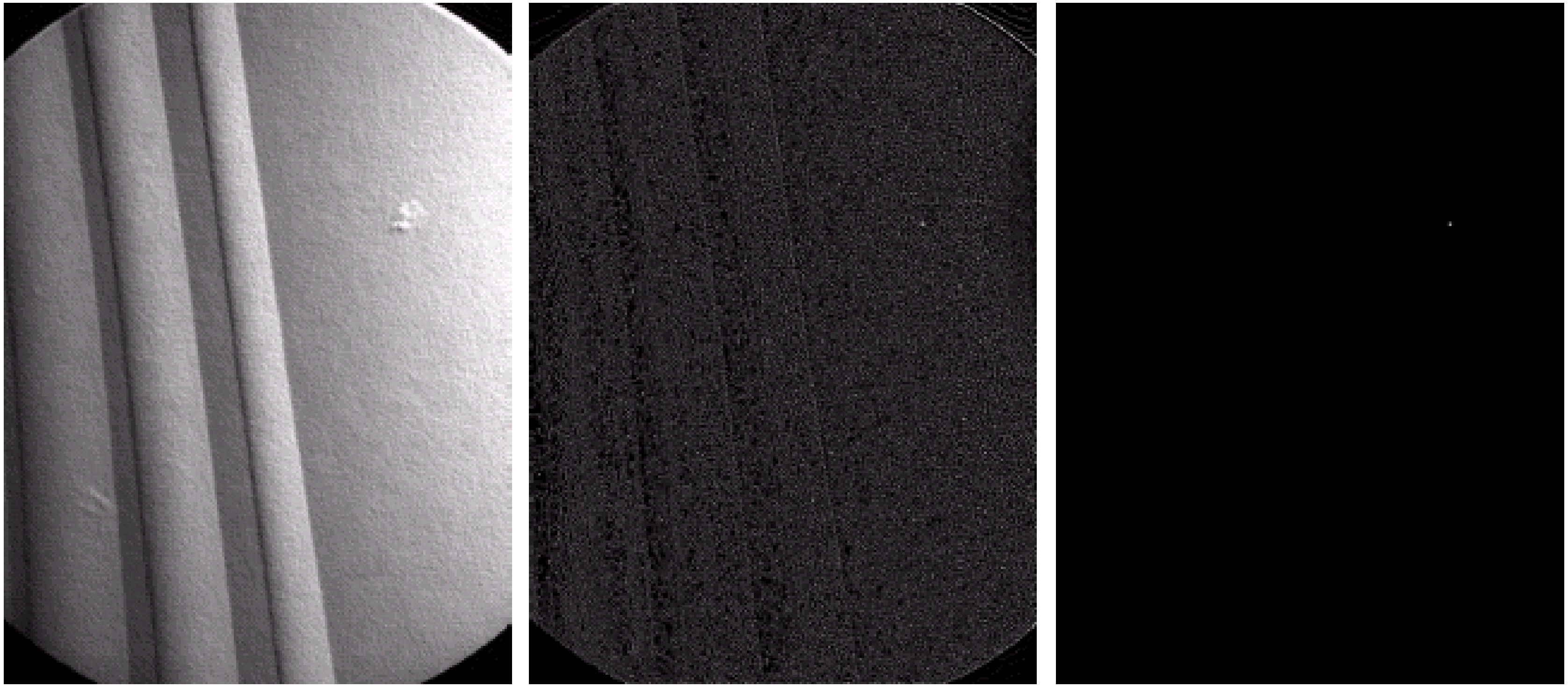
−1	−1	−1
−1	8	−1
−1	−1	−1

$$|R| \geq T$$

- where
  - T is a nonnegative *threshold*
  - R is the sum of products of the coefficients with the gray levels contained in the region encompassed by the mask.

Example

−1	−1	−1
−1	8	−1
−1	−1	−1



Line Detection

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

−45°

- Horizontal mask will result with *maximum* response when a line passes through the middle row of the mask with a constant background
- Similar idea is used with other masks

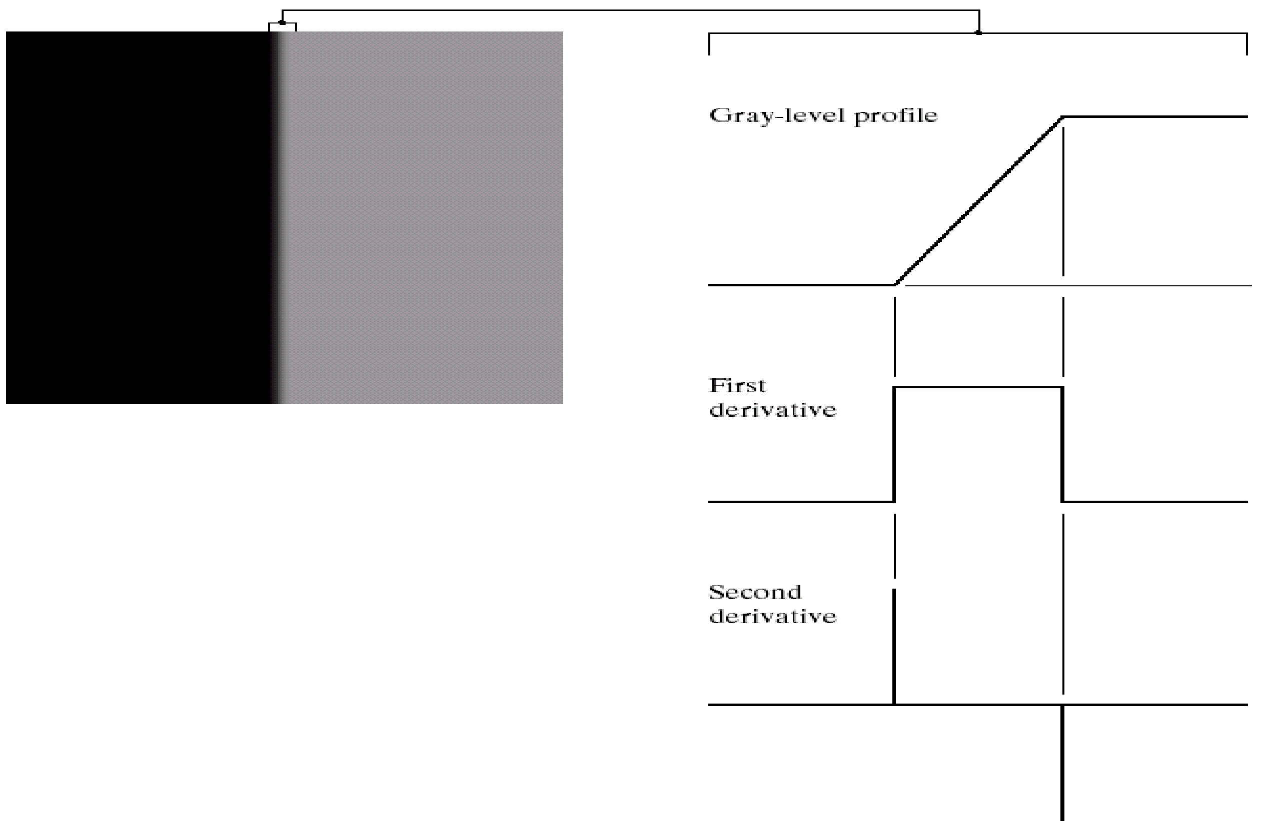
note: the preferred direction of each mask is weighted with a larger coefficient (i.e.,2) than other possible directions

Line Detection Contd.

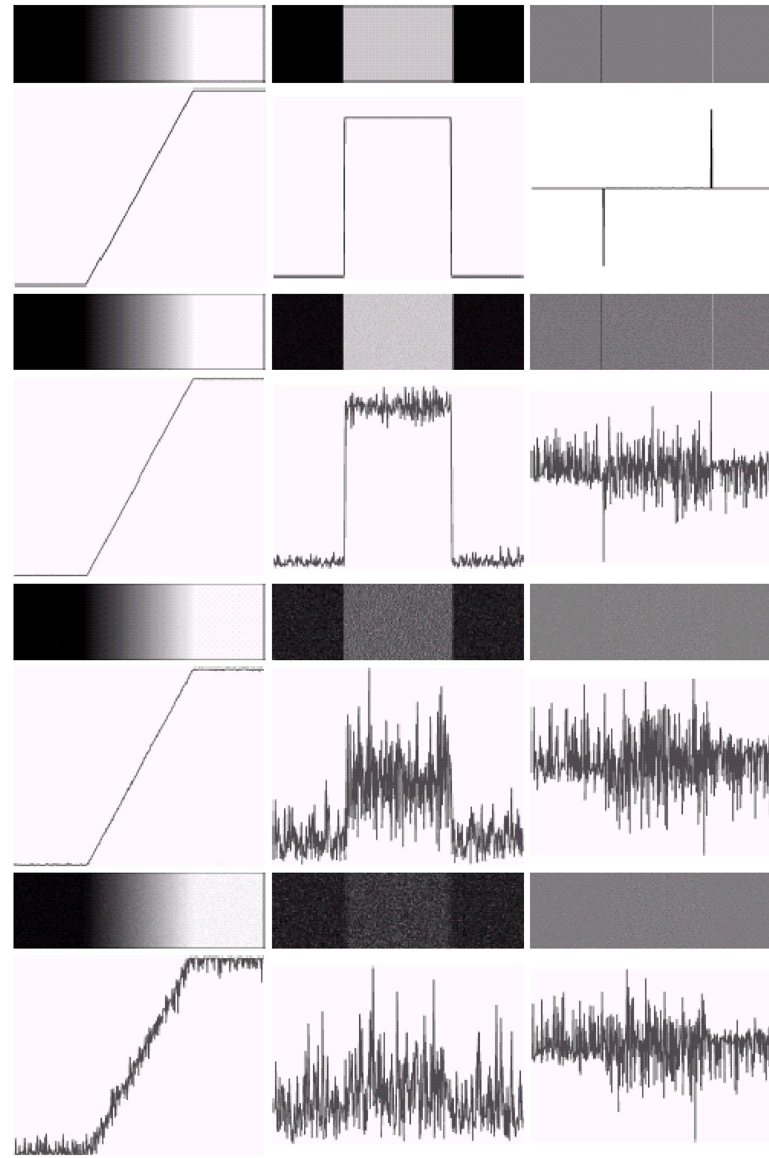
- Apply every mask on the image
- Let R1, R2, R3, R4 denotes the response of the horizontal, +45 degree, vertical and -45 degree masks, respectively
- If, at a certain point in the image $|R_i| > |R_j|$, for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i.
- Alternatively, for detecting all the lines in an image in the direction defined by a given mask, we simply run the mask through the image and threshold the absolute value of the result

Edge Detection

- first-order derivative (Gradient operator)
- second-order derivative (Laplacian operator)



Noisy Images



- First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and σ = 0.0, 0.1, 1.0 and 10.0, respectively.
- Second column: first-derivative images and gray-level profiles.
- Third column : second-derivative images and gray-level profiles.

Observation

- Fairly little noise can have such a significant impact on the two key derivatives used for edge detection in images
- Image smoothing should be serious consideration prior to the use of derivatives in applications where noise is likely to be present.

Gradient Masks

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

Diagonal edges with Prewitt and Sobel masks

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

Laplacian of Gaussian

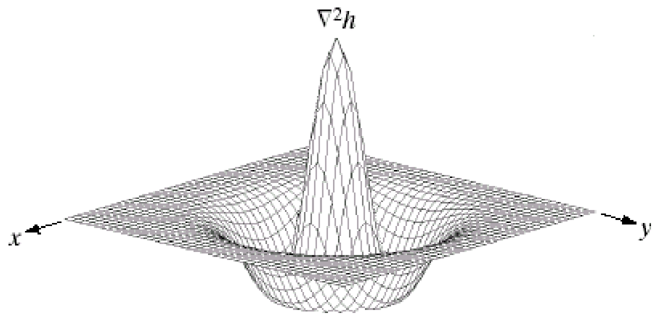
- Laplacian combined with smoothing to find edges via zero-crossing.

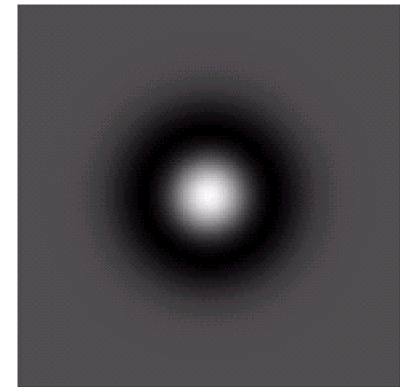
$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}$$

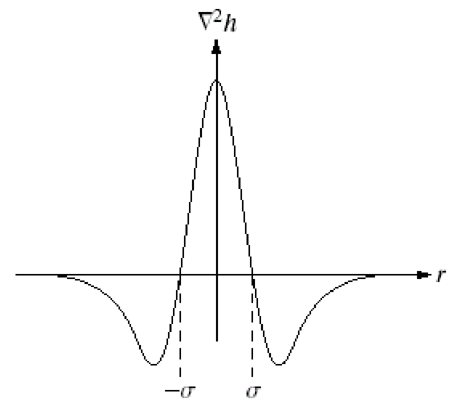
$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

where $r^2 = x^2+y^2$, and
σ is the standard deviation

Mexican hat

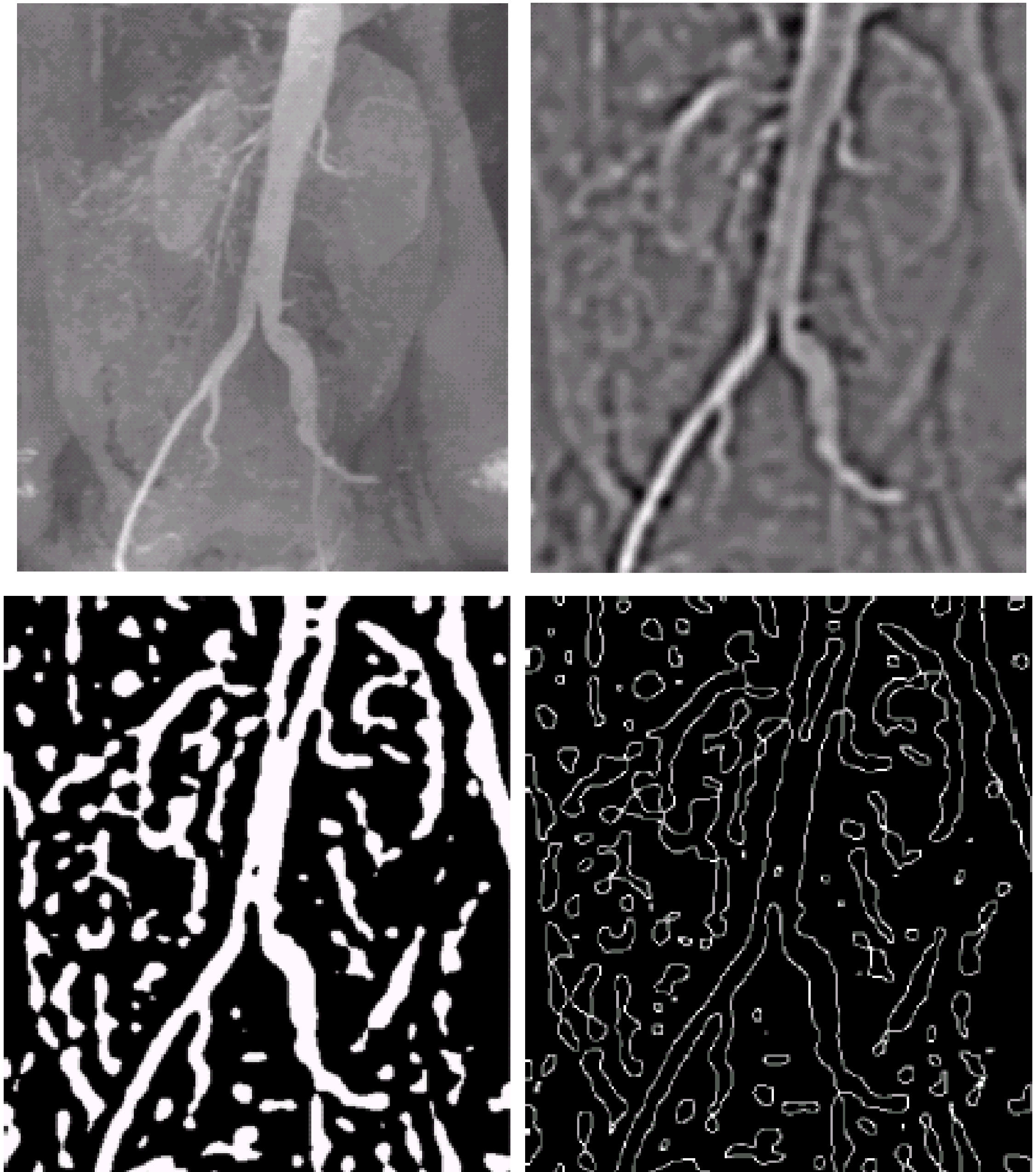






0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Example



a) Original image
b) LoG

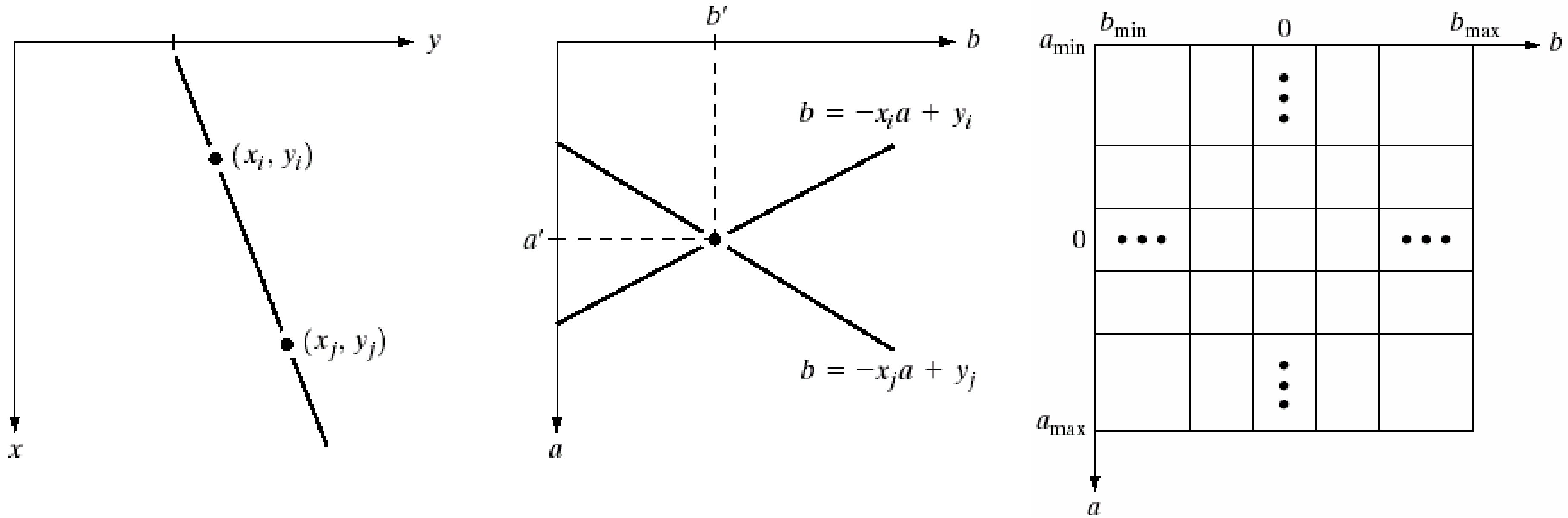
c) Threshold LoG
d) Zero crossing

Edge Linking: Local Processing

- Analyze pixels in small neighborhood of each edge point
- Pixels that are similar are linked
  - Link edges points with similar gradient magnitude and direction

Global Processing: Hough Transform

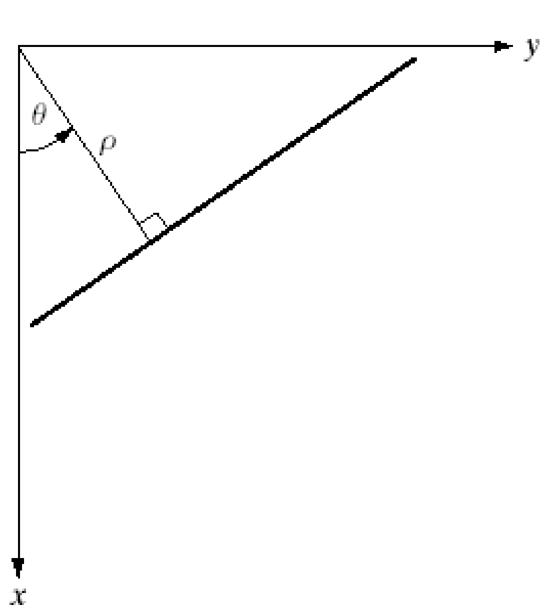
- Attempts to link edge pixels that lie on specified curves
- Representation of lines in parametric space: Cartesian coordinate

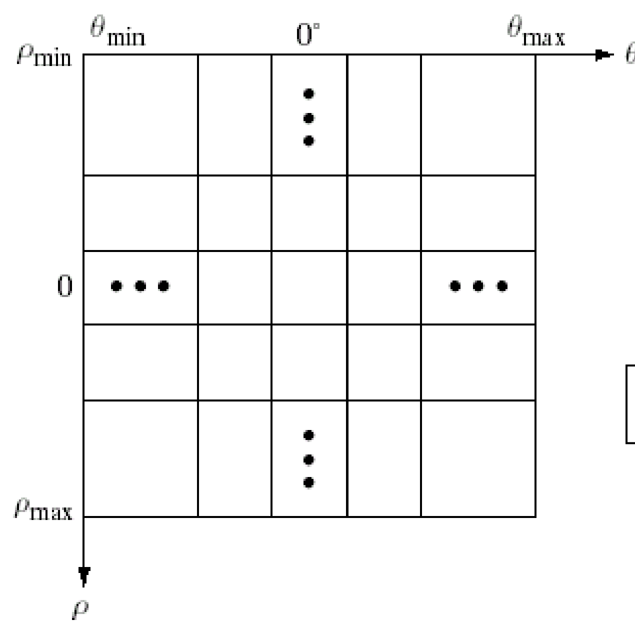


- Suppose that these two lines intersect at the point (a’, b’), then y = a’x+b’ represents the line in the xy-plane on which both (xi, yi) and (xj, yj) lie

Hough Transform Contd.

- Since a computer can only deal with a finite number of straight lines, we subdivide the parameter space *ab* into a finite number of accumulator cells...
- Representation in parametric space: polar coordinate



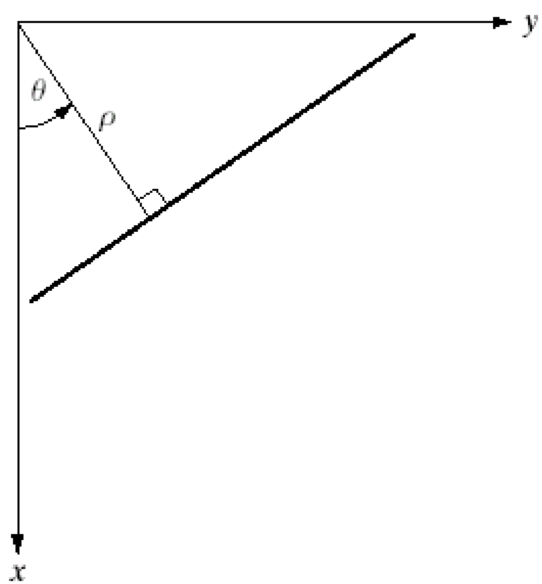


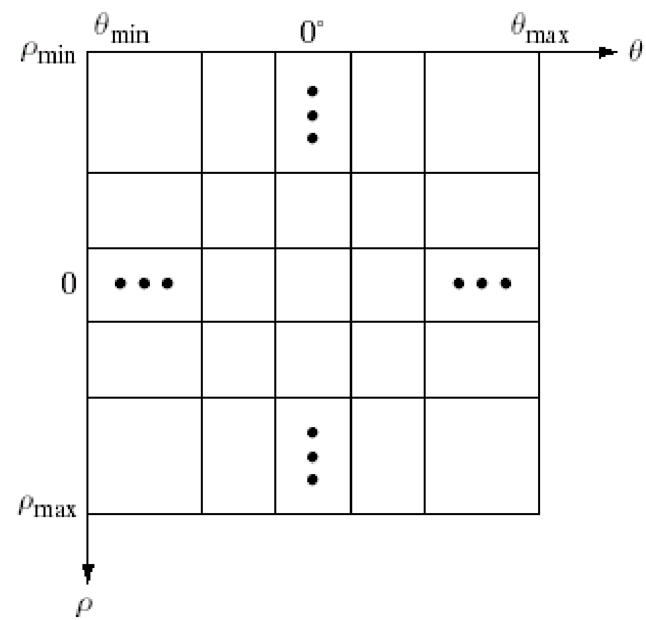
$$x \cos \theta + y \sin \theta = \rho$$

Hough Transform Contd.

- Now that ρ ∈ [−√2D, √2D] and θ ∈ [−90º, 90º],
- where √2D is the diagonal distance between two opposite corners in the image.

Problem solved!





Edge Linking Using Hough Transform

1) Compute $|\nabla f|$ and isolate edge pixels through thresholding
2) Specify subdivisions in the ρθ-plane
3) // Apply Hough transform to edge pixels

   Set all cells equal to zero

   For every (x_k, y_k)

      Let θ = every subdivision on the θ-axis

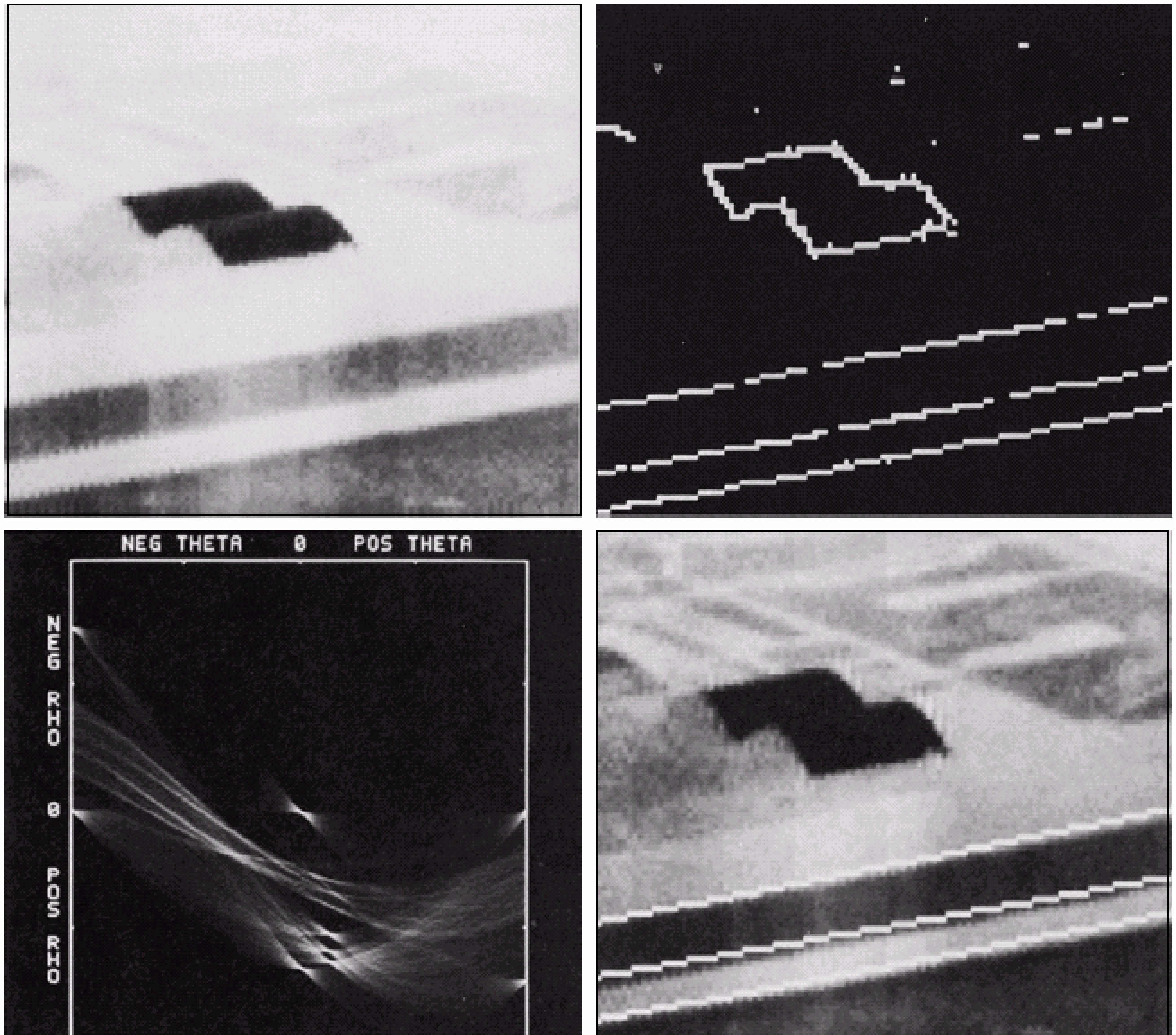
      Calculate $\rho = x_k \cos\theta + y_k \sin\theta$

      Round off ρ to the nearest allotted value on the ρ-axis

      Increment accumulator cell (ρ, θ) with 1

4) Identify accumulator cells with highest values
5) Examine continuity of pixels that constitute cell
6) Link these pixels if gaps are smaller than threshold

Example



Extension to more general Curves

- Hough transform applicable to any graph $g(v, c) = 0$, where *v* is vector of coordinates and *c* is vector of coefficients
- Example: Find the points that lie on a circle
- $(x - c_1)^2 + (y - c_2)^2 = c_3^2$
- The presence of three parameters (c1, c2 and c3) results in a 3-D parameter space with cube-like cells and accumulators of the form *A(i, j, k)*!

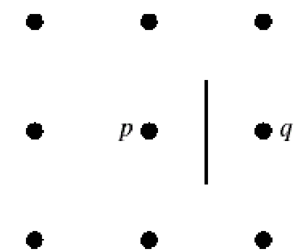
Global processing: Graphic-Theoretic Techniques

- Represent edge segments in the form of a graph
- Search graph for low-cost paths that correspond to significant edges
- Rugged approach that performs well in the presence of noise
- Procedure more complicated; requires more processing time

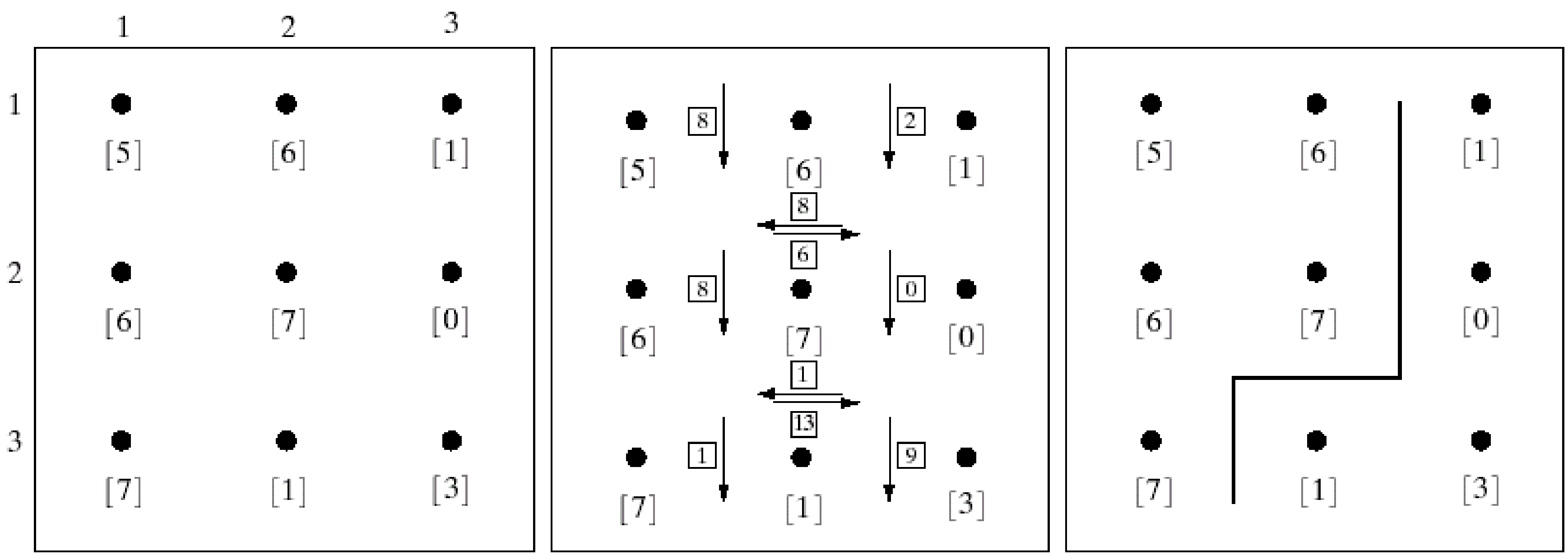
Graphic-Theoretic Techniques

- Minimal-cost path

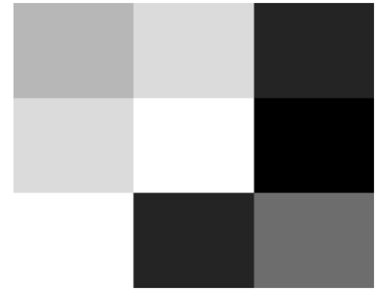
$$c = \sum_{i=2}^{k} c(n_{i-1}, n_i)$$

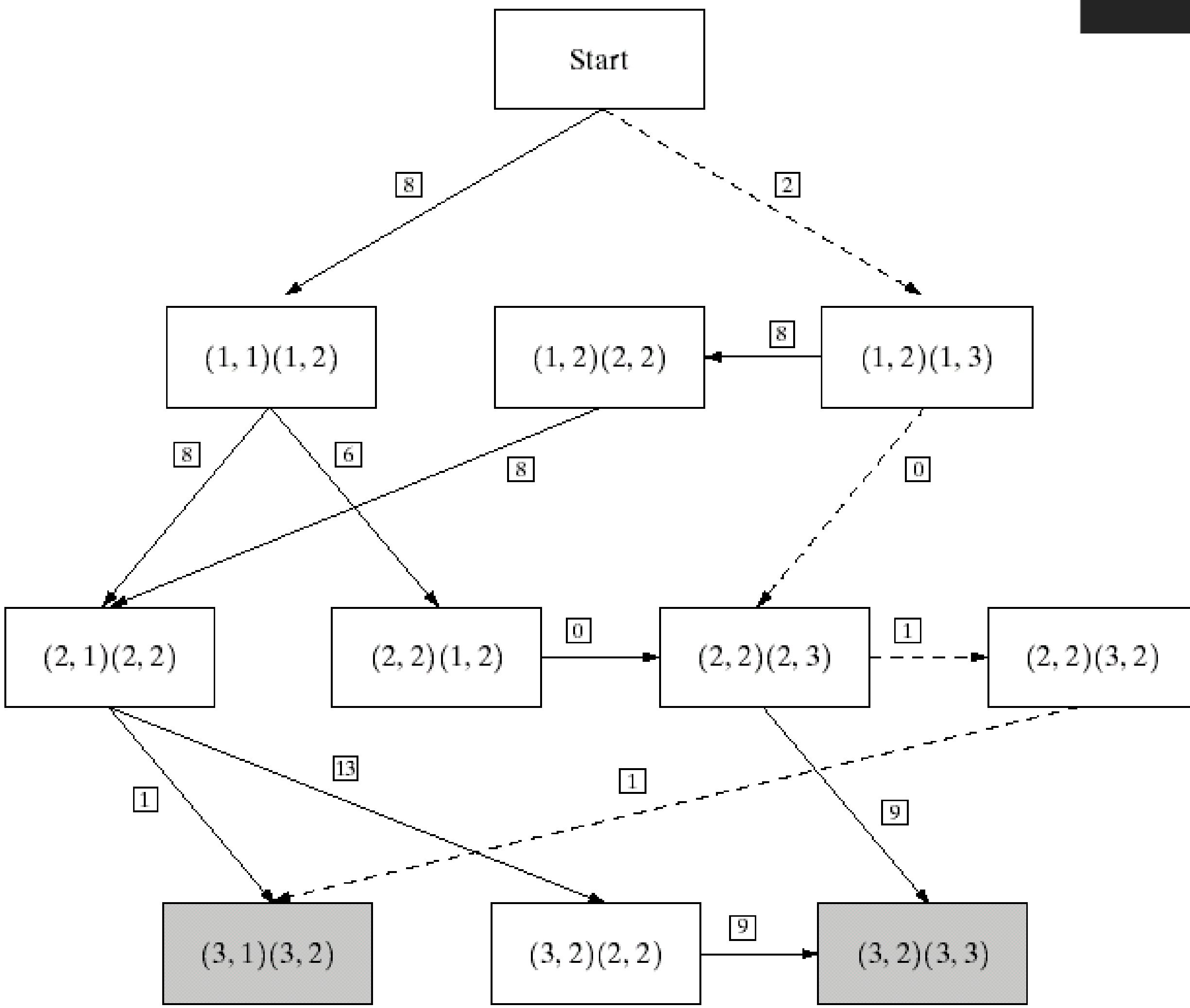


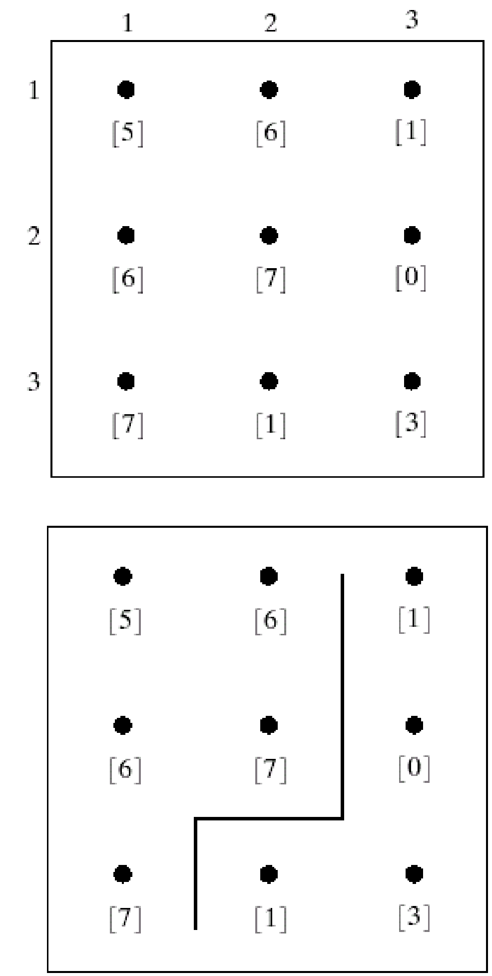
$$c(p,q) = H - [f(p) - f(q)]$$



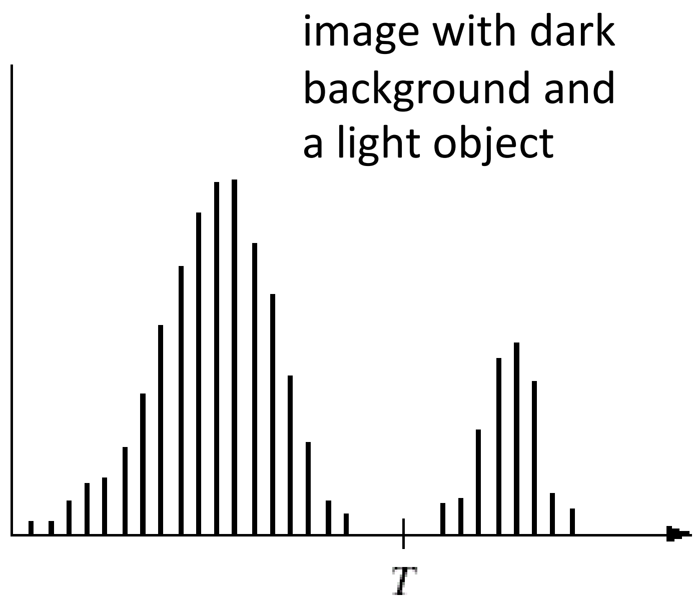
Illustration

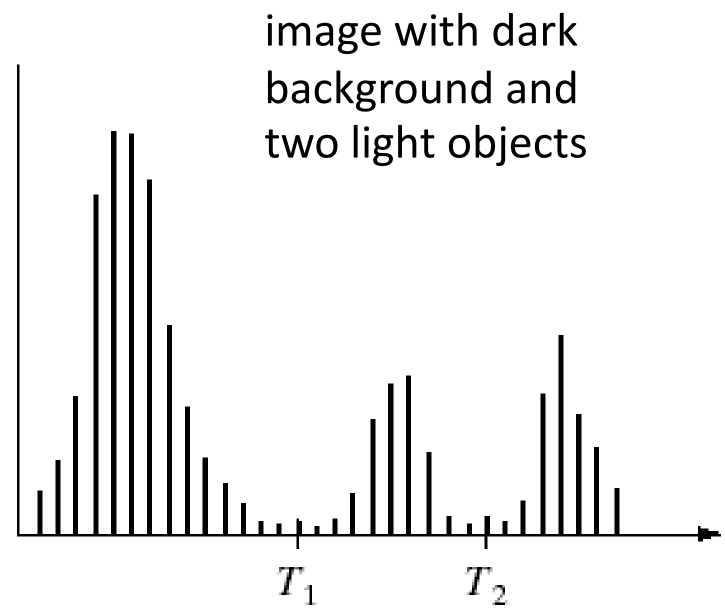






Thresholding





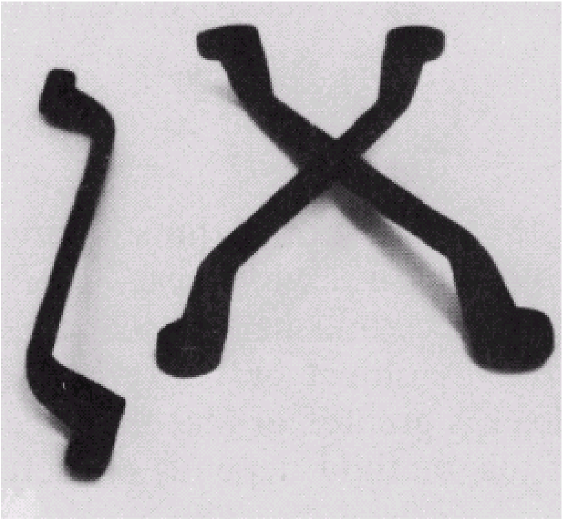
T depends on

only f(x,y) : only on gray-level values ⇨ ***Global threshold***

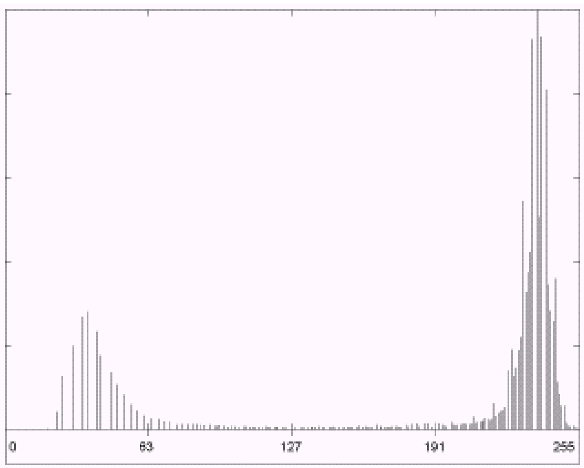
both f(x,y) and p(x,y) : on gray-level values and its neighbors ⇨ ***Local threshold***

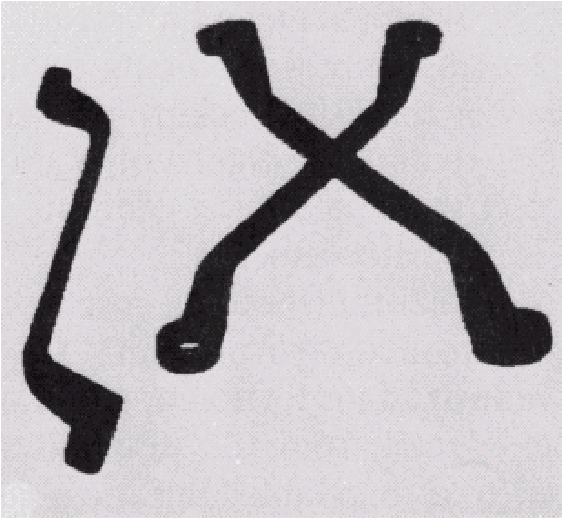
x, y, p(x, y), and f(x, y) ⇨ ***Dynamic or adaptive threshold***

Basic Global Thresholding



use T midway between the max and min gray levels





generate binary image

Basic Global Thresholding Algorithm

1. Select an initial estimate for T

2. Segment the image using T.

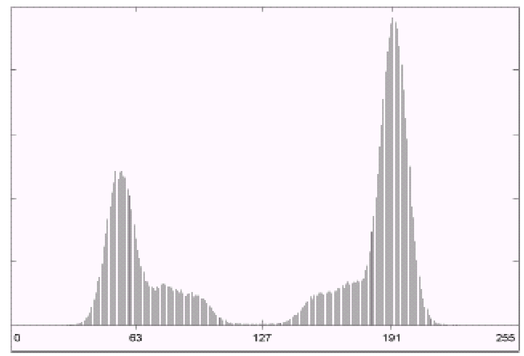
   This will produce two groups of pixels: G_1 consisting of all pixels with gray level values > T and G_2 consisting of pixels with gray level values $\leq$ T

3. Compute the average gray level values $\mu 1$ and $\mu 2$ for the pixels in regions G1 and G2

4. Compute a new threshold value $T = 0.5\ (\mu_1 + \mu_2)$

5. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_o.

Example: Heuristic method

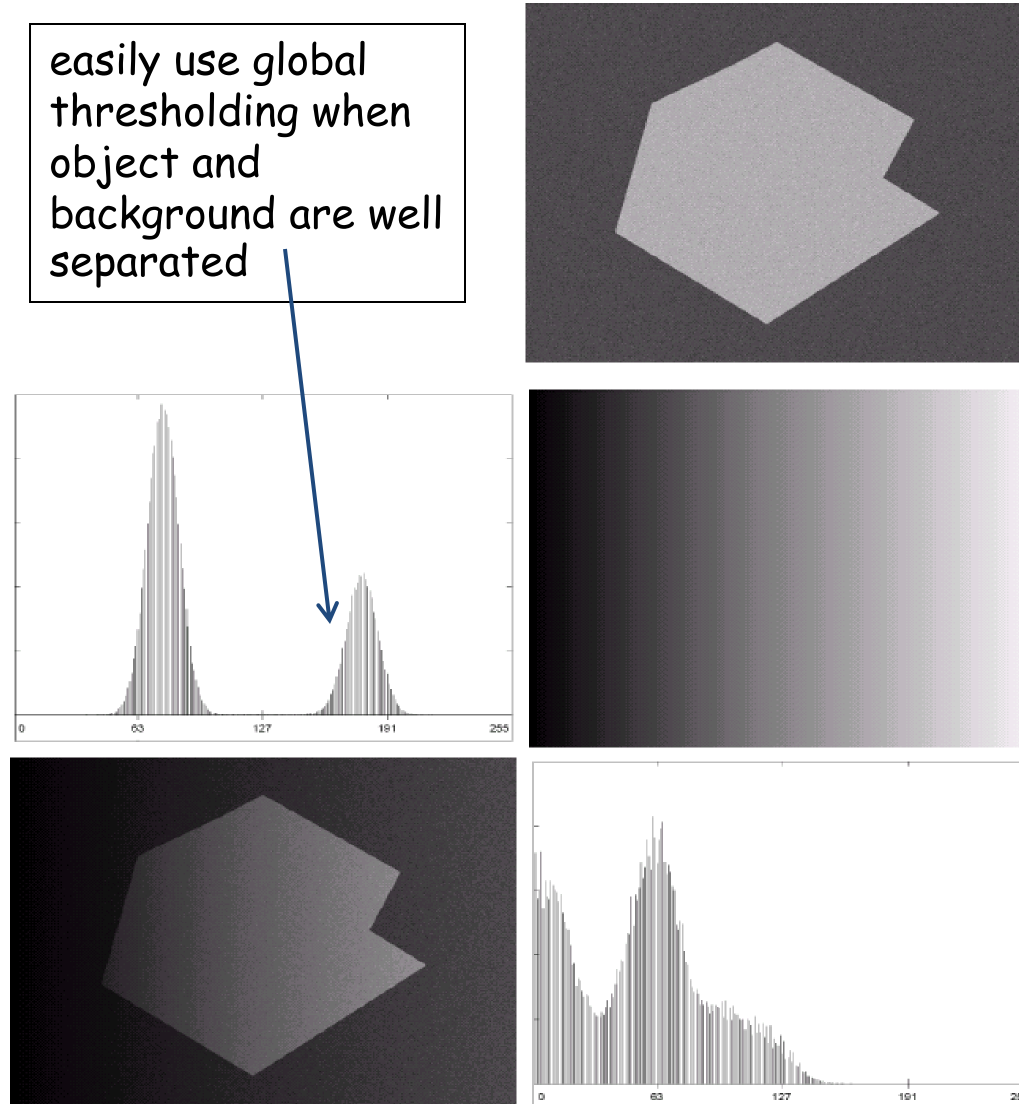


$T_0 = 0$
3 iterations
with result T = 125

The Role of Illumination

$$f(x,y) = i(x,y)\ r(x,y)$$

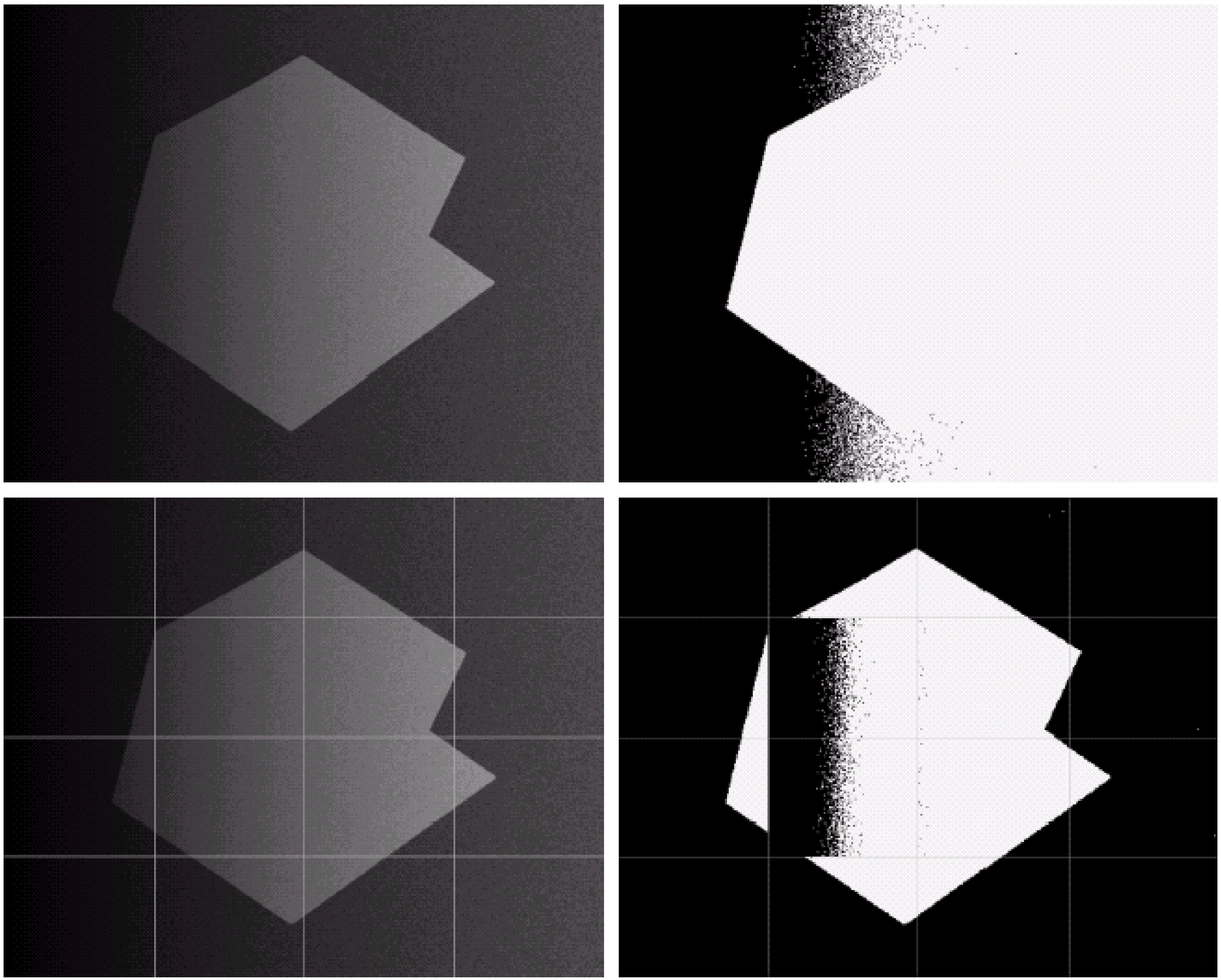
- In general, when only the reflectance component is present, the modes in the histogram can be more easily separated
- When the illumination component is present separation becomes much more difficult...



Local / Basic Adaptive Thresholding

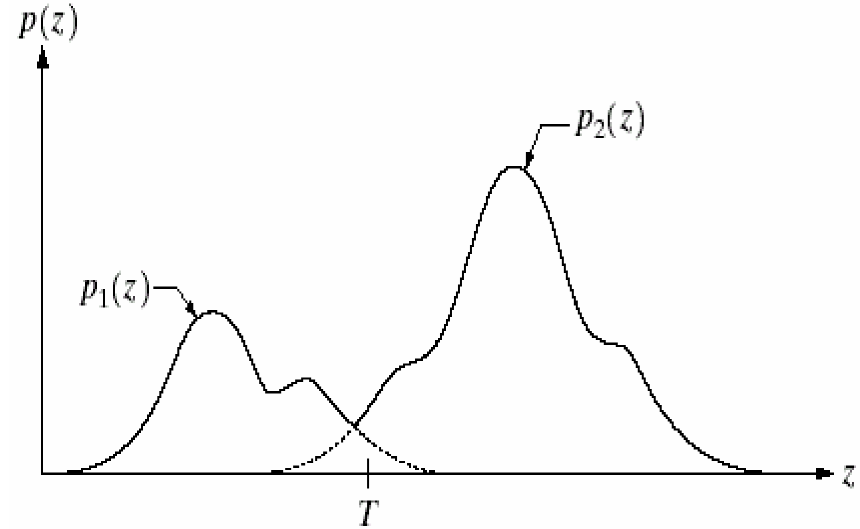
- Subdivide original image into small areas.
- Utilize a different threshold to segment each sub-image
- Since the threshold used for each pixel depends on the location of the pixel in terms of the sub-image, this type of thresholding is adaptive.

Example : Adaptive Thresholding



Optimal Global and Adaptive Thresholding

- Mixture PDF describing overall gray level variation...
- p(z) = P1 p1(z) + P2 p2(z)
- P1: probability that pixel is object pixel
- P2: probability that pixel is background pixel
- P1 + P2 = 1



Select T that minimizes average error in making decision

- Probability in erroneously classifying background as object

$$E_1(T) = \int_{-\infty}^{T} p_2(z)\, dz$$

- Probability in erroneously classifying object as background

$$E_2(T) = \int_{T}^{\infty} p_1(z)\, dz$$

- Overall probability of error is

$$E(T) = P2\ E1(T) + P1\ E2(T)$$

- Threshold value for which the error is minimal

$$P1\ p1(T) = P2\ p2(T)$$

Optimal threshold for Gaussian Densities

- Approximate p1(z) and p2(z) with Gaussian densities

$$p(z) = \frac{P_1}{\sqrt{2\pi}} e^{-(z-\mu_1)^2/2\sigma_1^2} + \frac{P_2}{\sqrt{2\pi}} e^{-(z-\mu_2)^2/2\sigma_2^2}$$

- Using optimality condition results in $AT^2 + BT + C = 0$, where

$$\begin{aligned} A &= \sigma_1^2 - \sigma_2^2 \\ B &= 2\,(\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2) \\ C &= \sigma_1^2 \mu_2^2 - \sigma_2^2 \mu_1^2 + 2\,\sigma_1^2 \sigma_2^2 \ln(\sigma_2 P_1/\sigma_1 P_2) \end{aligned}$$

- if $\sigma^2 = \sigma_1^2 = \sigma_2^2$, one threshold is sufficient

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$

- When P1 = P2 and/or $\sigma = 0$ the optimal threshold is the average of the means

Boundary Characteristic for Histogram Improvement and Local Thresholding

light object with dark background

$$s(x,y)=\begin{cases}0 & \text{if } \nabla f < T \\ + & \text{if } \nabla f \geq T \text{ and } \nabla^2 f \geq 0 \\ - & \text{if } \nabla f \geq T \text{ and } \nabla^2 f < 0\end{cases}$$

- Gradient gives an indication of whether a pixel is on an edge
- Laplacian can yield information regarding whether a given pixel lies on the dark or light side of the edge
- all pixels that are not on an edge are labeled 0
- all pixels that are on the dark side of an edge are labeled +
- all pixels that are on the light side an edge are labeled -

Example

Automatic Thresholding

- Use of one or more of the following:-
  1. Intensity characteristics of objects
  2. Sizes of objects
  3. Fractions of image occupied by objects
  4. Number of different types of objects
- Size and probability of occurrence – most popular
- Intensity distributions estimate by histogram computation.

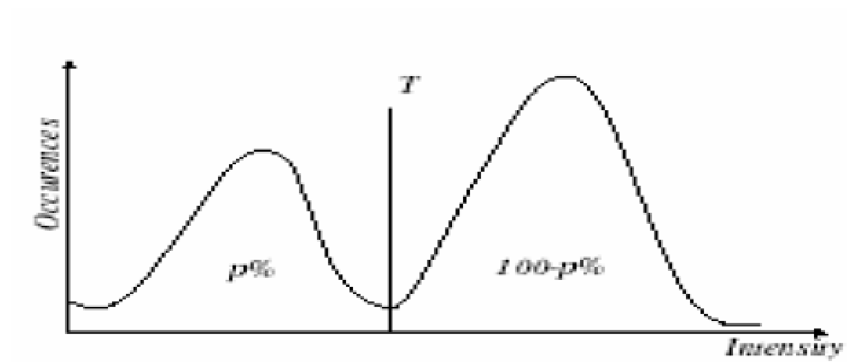
Automatic Thresholding Methods

- Some automatic thresholding schemes:

1. P-tile method
2. Mode method
3. Iterative threshold selection
4. Adaptive thresholding
5. Variable thresholding
6. Double thresholding

Thresholding Methods

- P-tile Method:- If object occupies P% of image pixels the set a threshold T such that P% of pixels have intensity below T.



- Iterative Thresholding:-

Successively refines an approx. threshold to get a new value which partitions the image better.

$$T = \frac{1}{2}\left(\mu 1 + \mu 2\right)$$

Thresholding Methods (Continued)

- Adaptive Thresholding:- Used in scenes with uneven illumination where same threshold value not usable throughout complete image.
- In such case, look at small regions in the image and obtain thresholds for individual sub-images. Final segmentation is the union of the regions of sub-images.

- Variable Thresholding:- Approximates the intensity values by a simple function such as a plane or biquadratic. It is called background normalization.

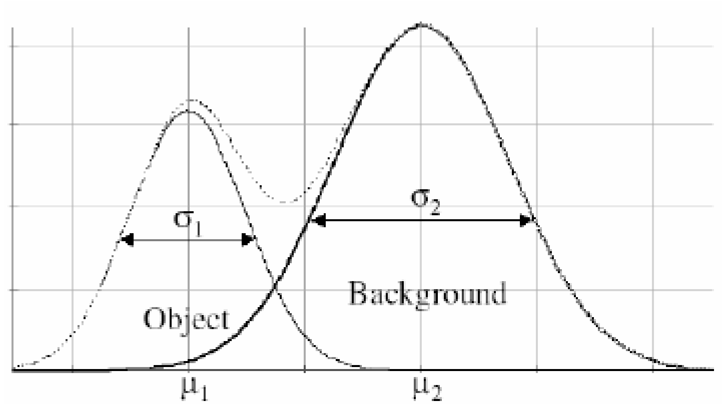
Even more thresholding methods

- Mode method:-

Assume that gray values are drawn from two normal distributions with parameters $(\mu1, \sigma1), (\mu2, \sigma2)$

If the standard deviations are zero, there will be two spikes in the histogram and the threshold can be placed anywhere between them.

For non-ideal cases, there will be peaks and valleys and the threshold can be placed corresponding to the valley.



Region-Based Segmentation - Region Growing

- start with a set of **“seed”** points
- growing by appending to each seed those neighbors that have similar properties such as specific ranges of gray level

Region Growing

select all seed points with gray level 255

criteria:

1. the absolute gray-level difference between any pixel and the seed has to be less than 65
2. the pixel has to be 8-connected to at least one pixel in that region (if more, the regions are merged)