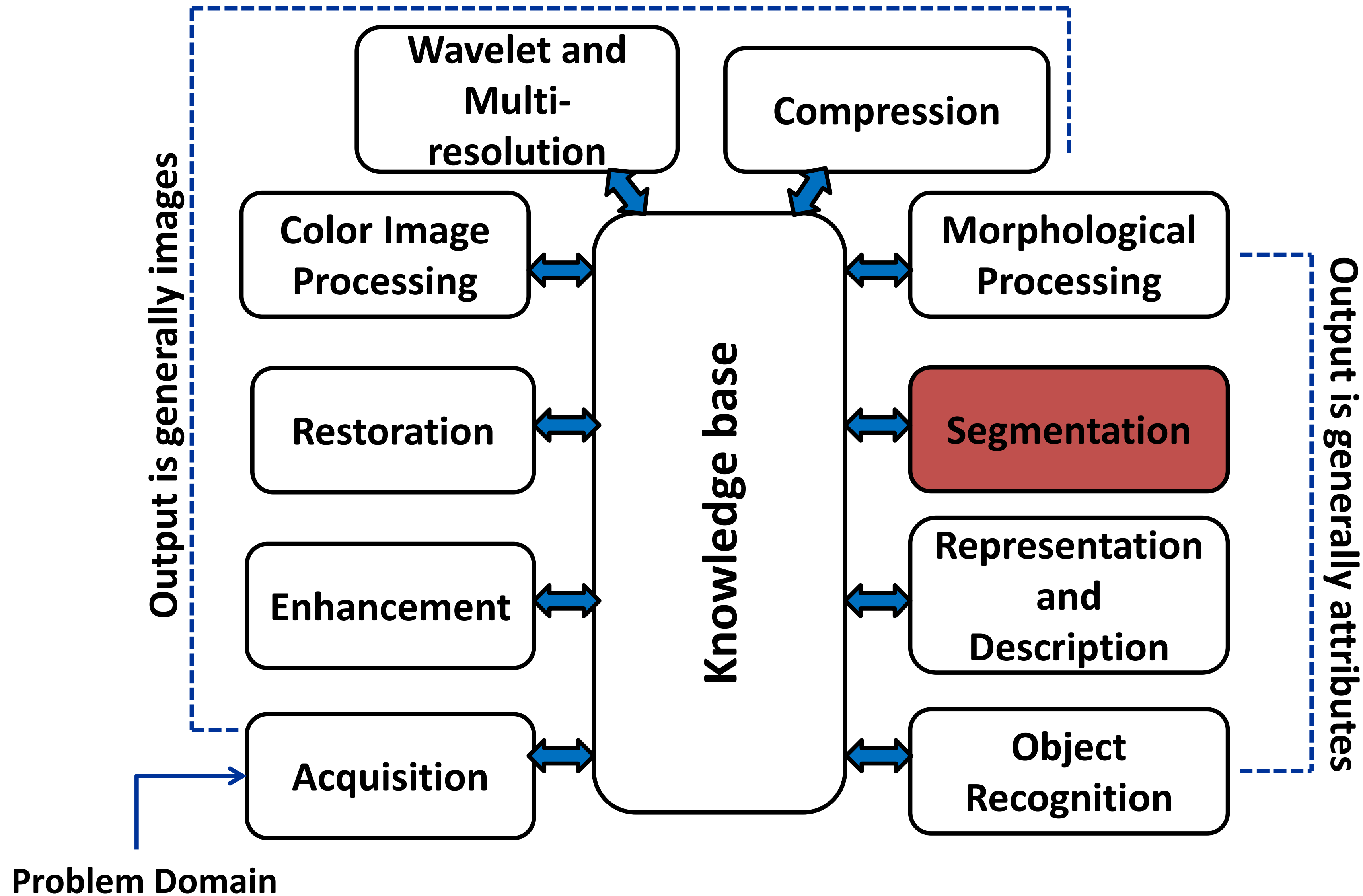
CSC 447 Digital Image Processing

Maryam_nabil@cis.asu.edu.eg

Image Segmentation I

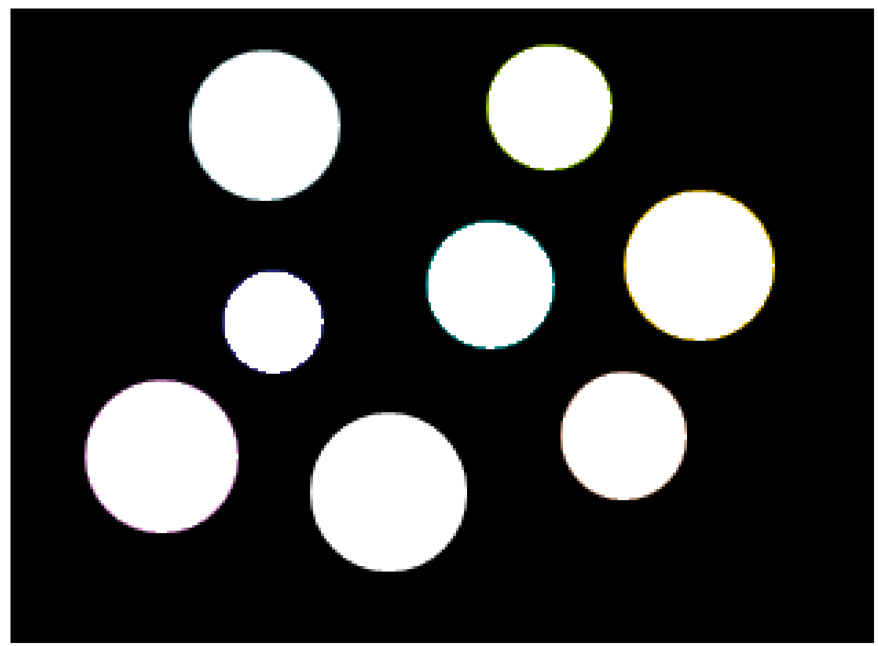
Fundamental Steps of DIP



Contents

Basics of Image Segmentation

- **Segmentation is to subdivide an image into its constituent regions or objects.**
- **Typically the first step in any automated computer vision application.**



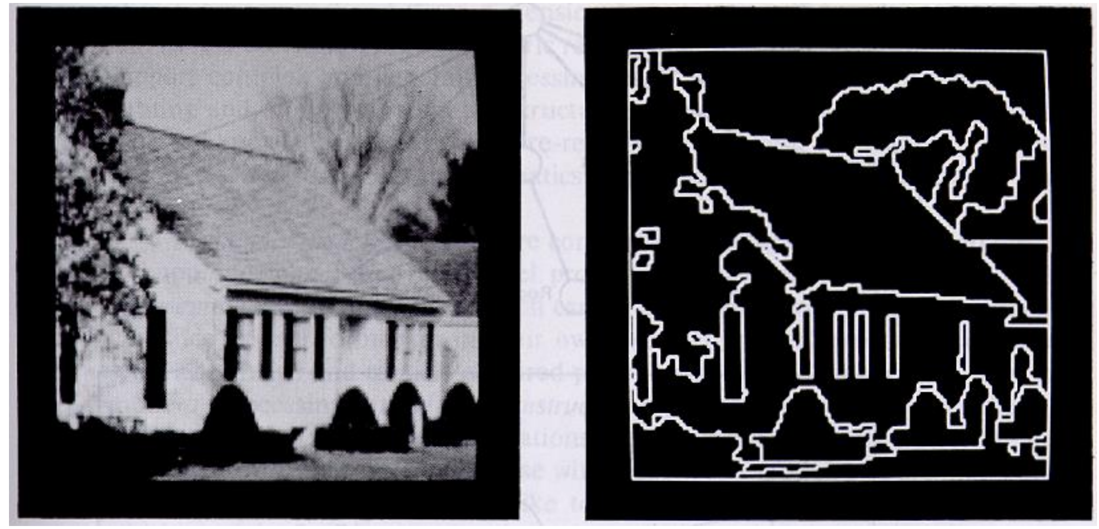
Basics of Image Segmentation – (cont.)

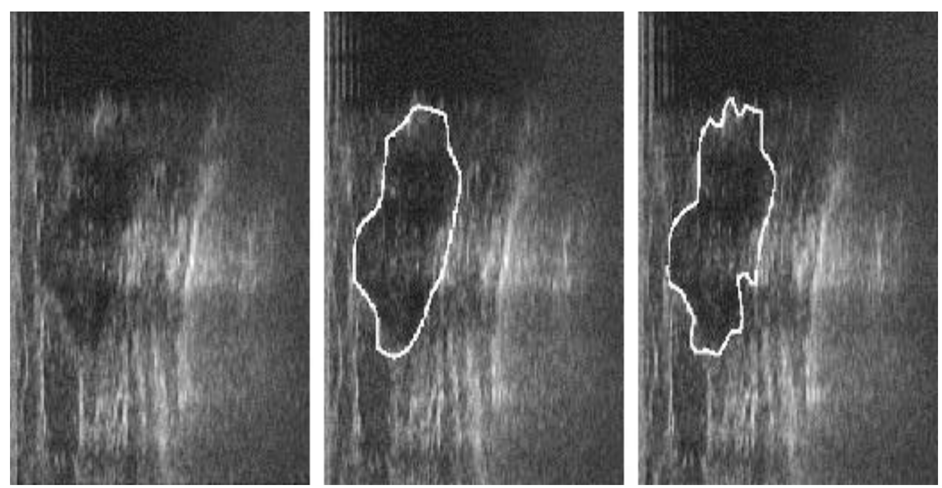
- **Automated segmentation is a very difficult problem in general.**

- **The results of segmentation are very important in determining the eventual success or failure of image analysis.**

Basics of Image Segmentation – (cont.)

Examples

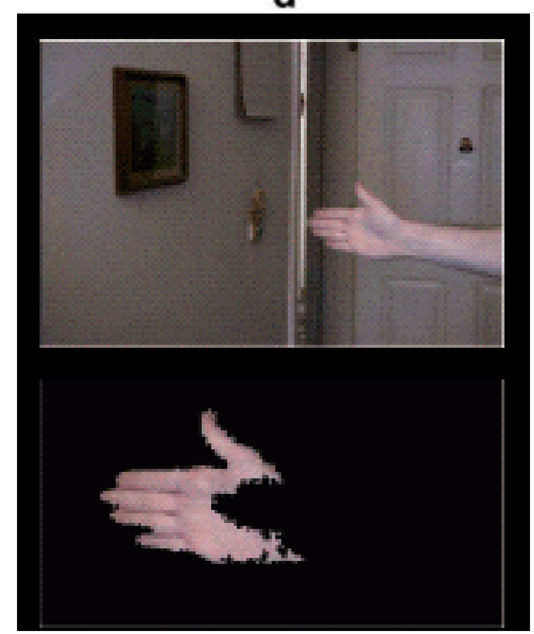




Basics of Image Segmentation – (cont.)

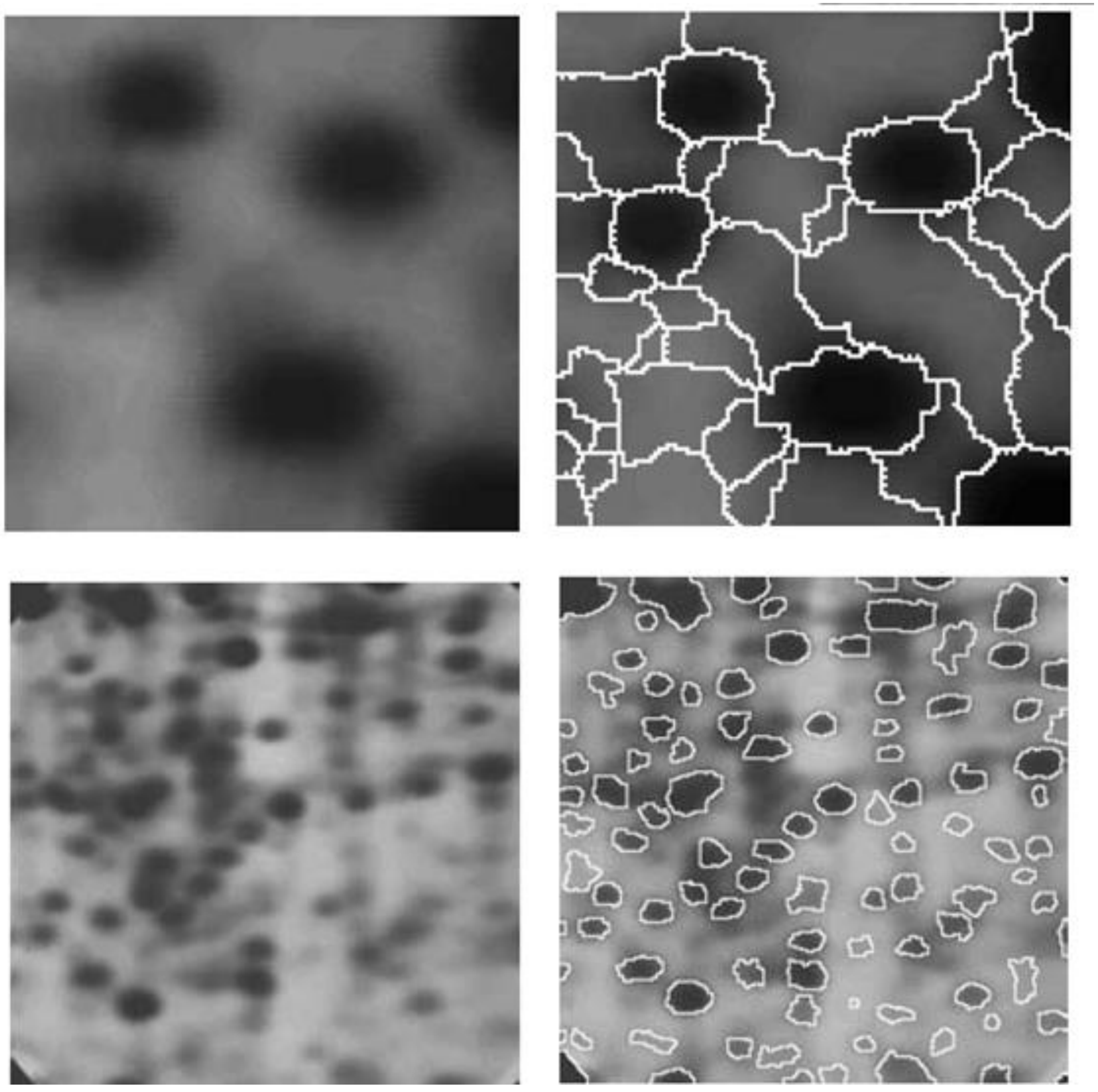
Examples

a b c d



Basics of Image Segmentation – (cont.)

Examples



Basics of Image Segmentation – (cont.)

- **Monochrome image segmentation algorithms are generally based on:**

1- Edge-based Approaches

Use the boundaries of regions to segment the image by detecting abrupt changes in intensity (discontinuities) → edge detection

2- Region-based Approaches

Use similarity among pixels to partition an image into regions according to a set of predefined criteria.
→ thresholding, region growing, merging, splitting…

Basics of Image Segmentation – (cont.)

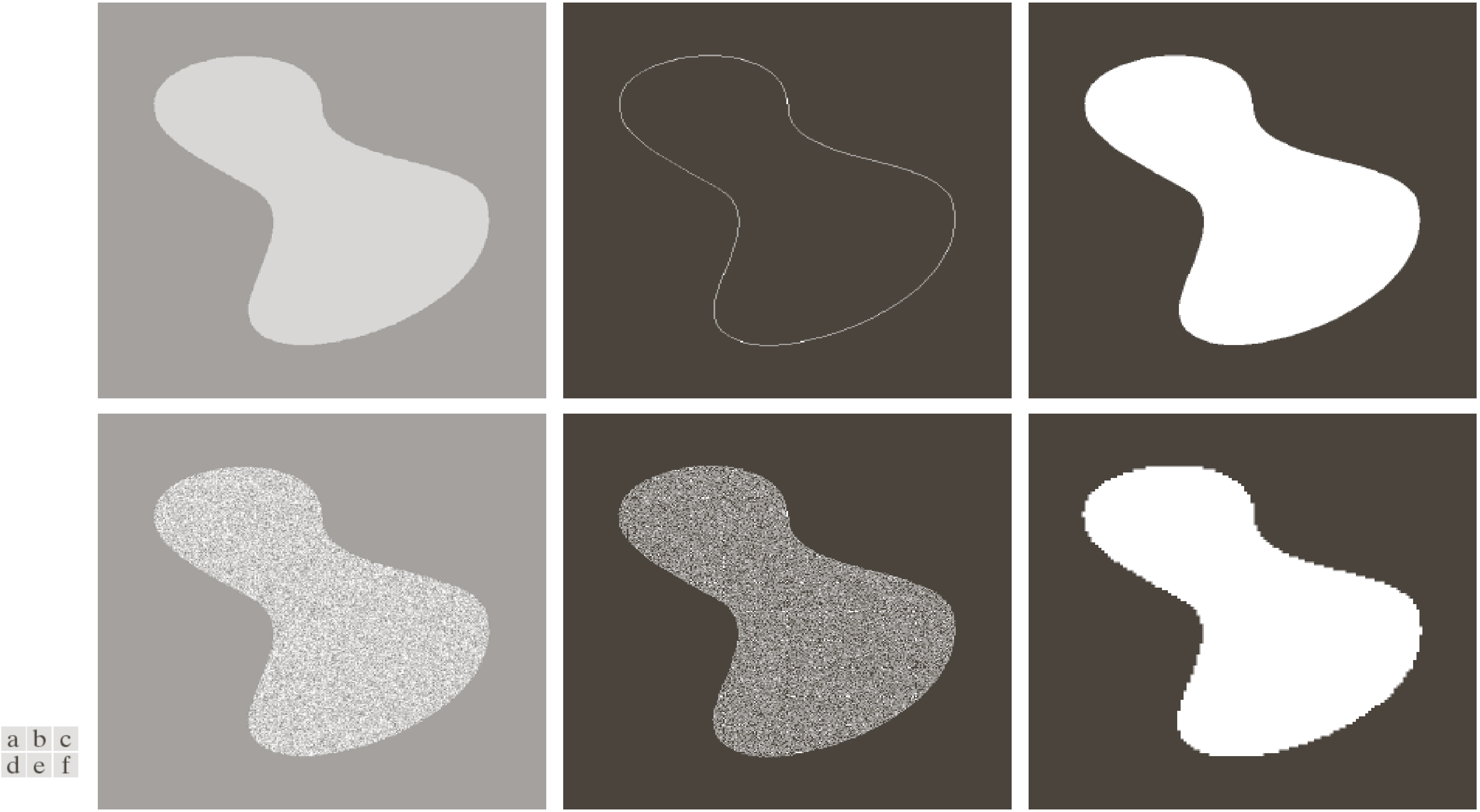


FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Edge-based

Edge-based Approaches

- Discontinuities

Point, line, edge

- Discontinuity Detectors

1st derivative (Gradient).

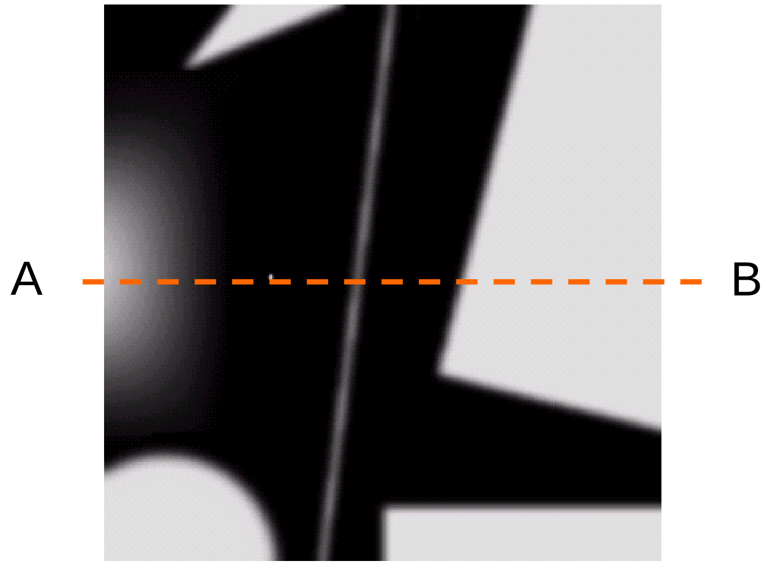
2nd derivative (Laplacian).

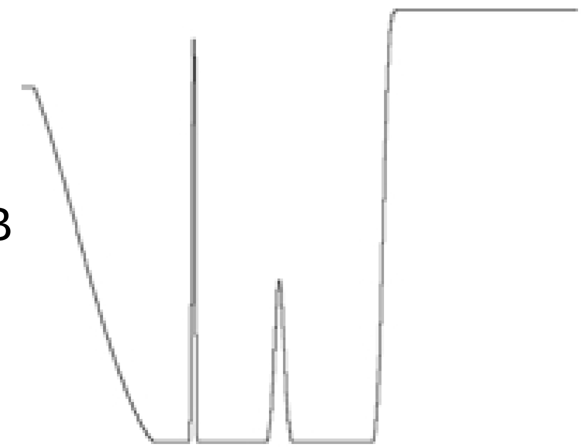
$$\nabla f = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

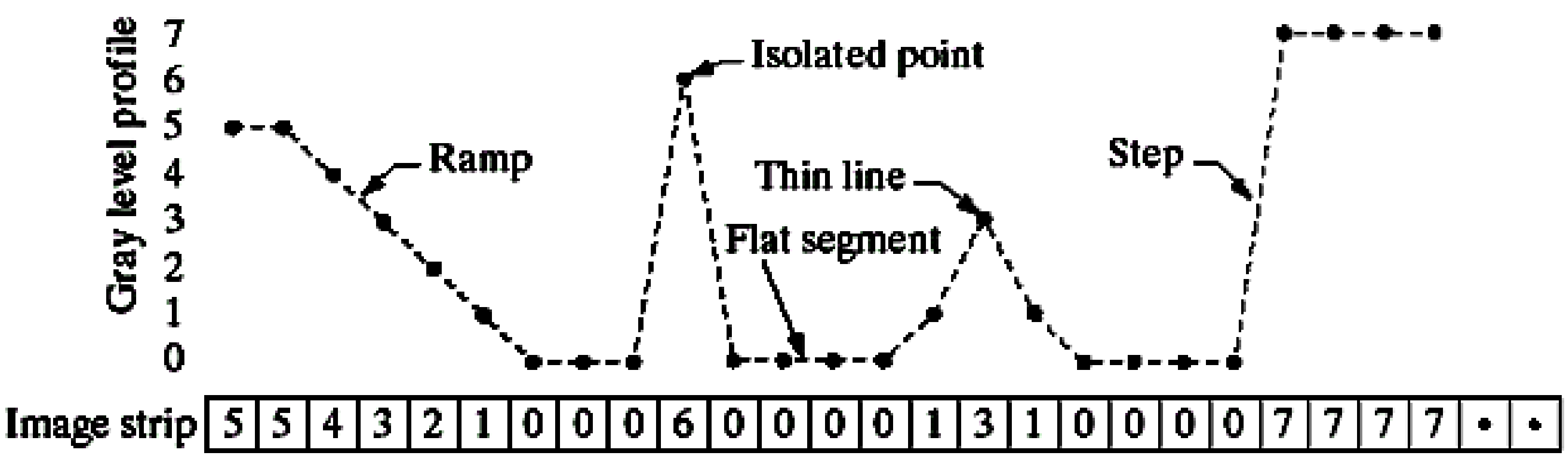
$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$$

Recall - Sharpening Filters

1st Derivative Example in 1D:

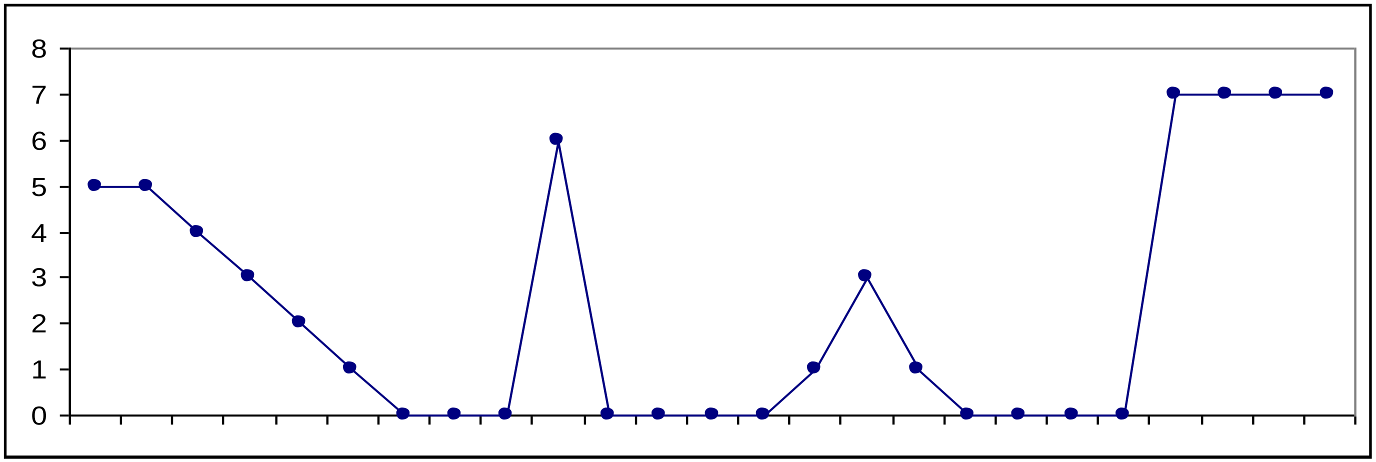






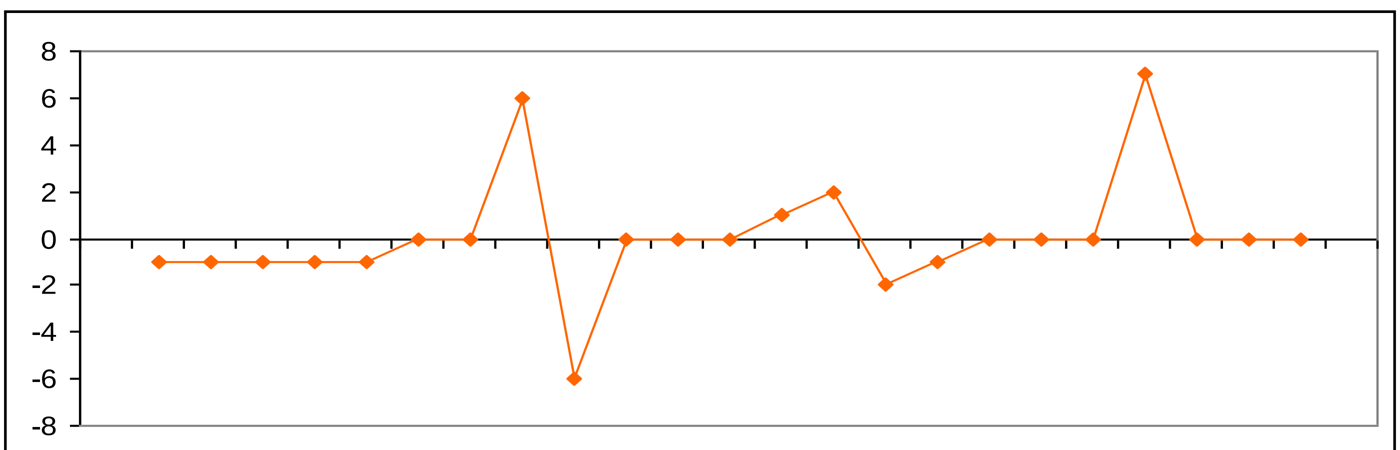
Recall - Sharpening Filters

1st Derivative Example in 1D:



5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

	0	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	



Recall - Sharpening Filters

2nd Derivative:

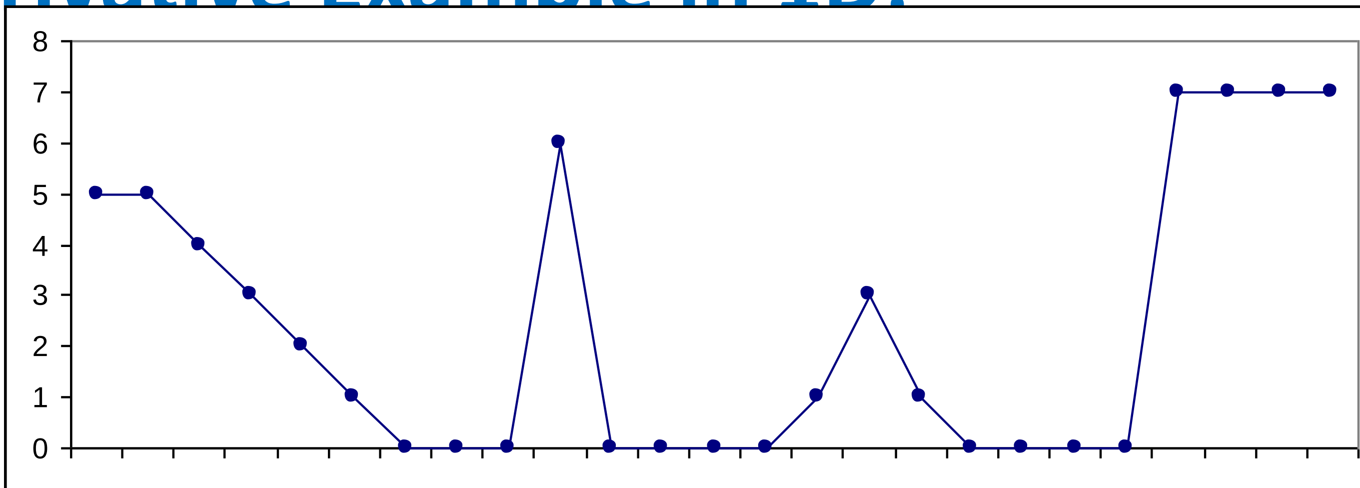
The formula for the 2nd derivative of a function is as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

Simply takes into account the values both before and after the current value.

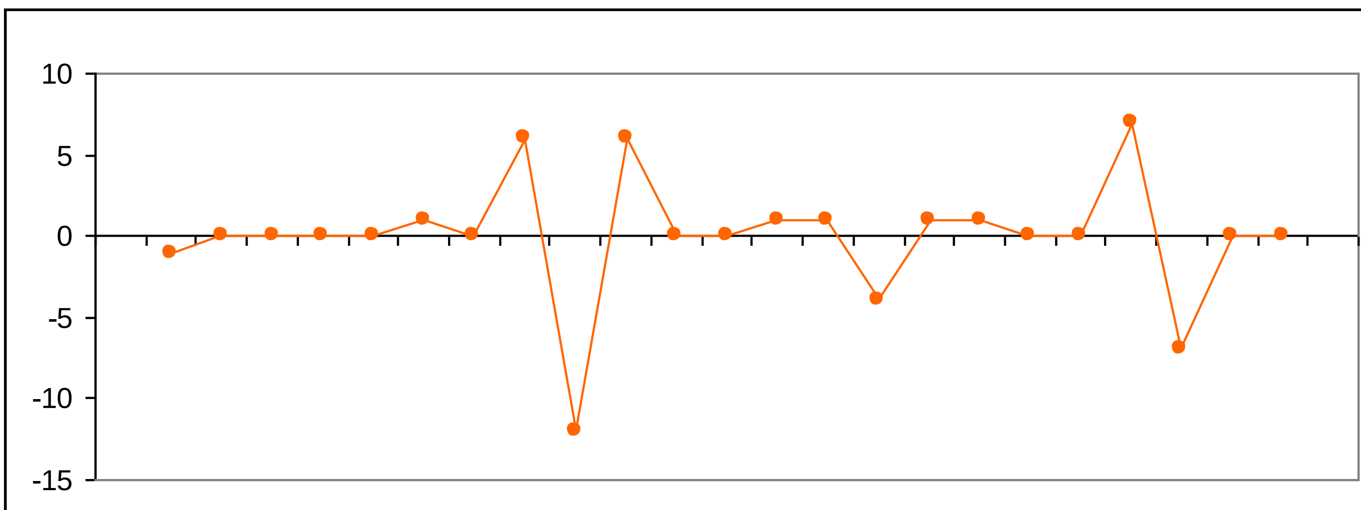
Sharpening Filters – (cont.)

2^{nd} Derivative Example in 1D:

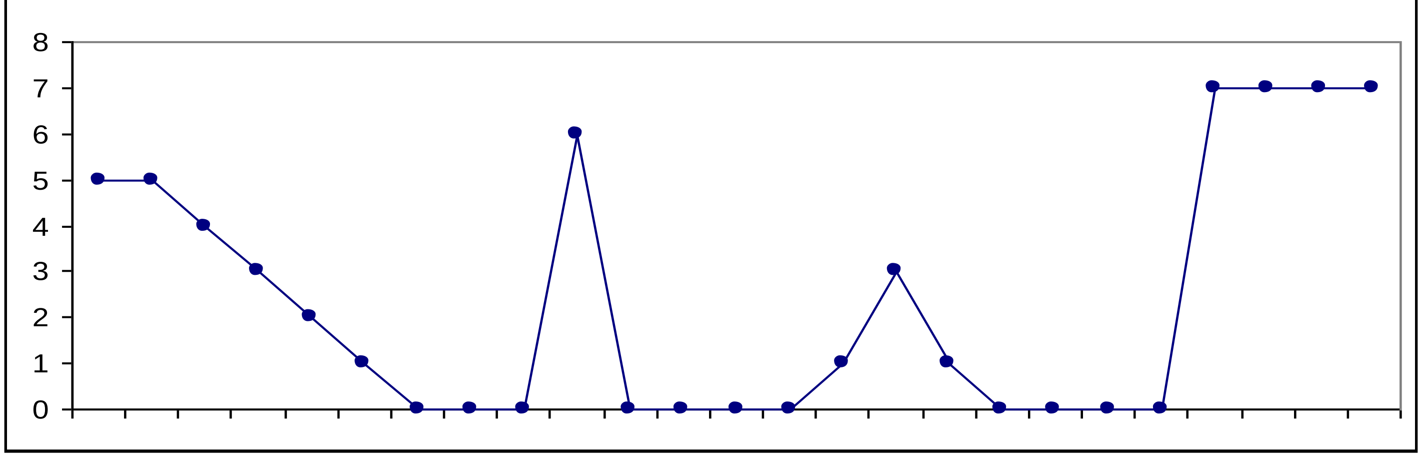


5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

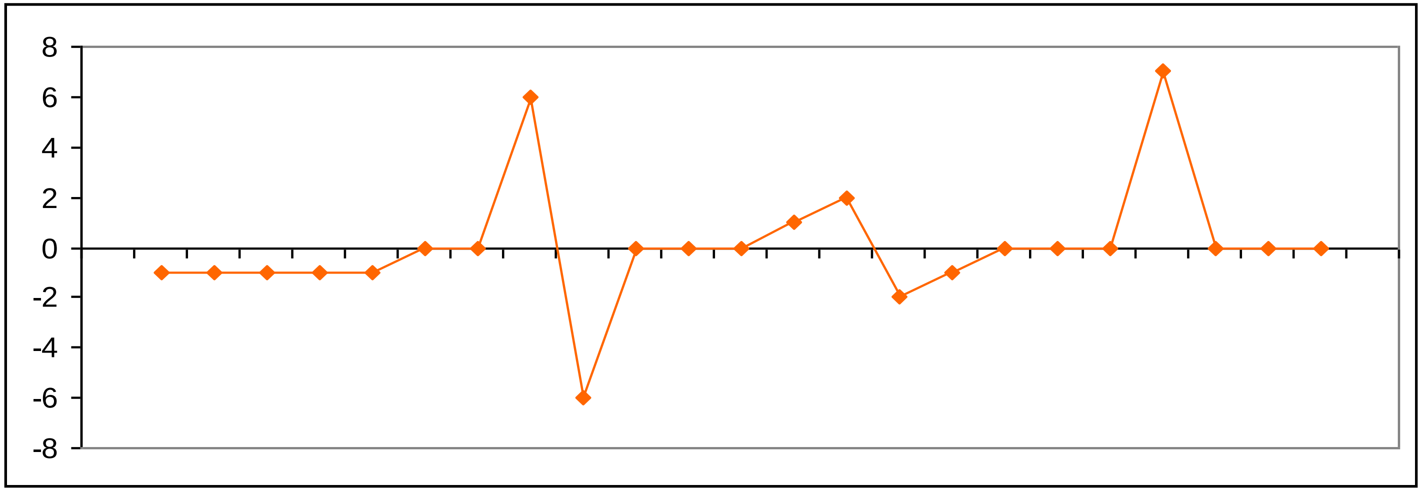
	-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	



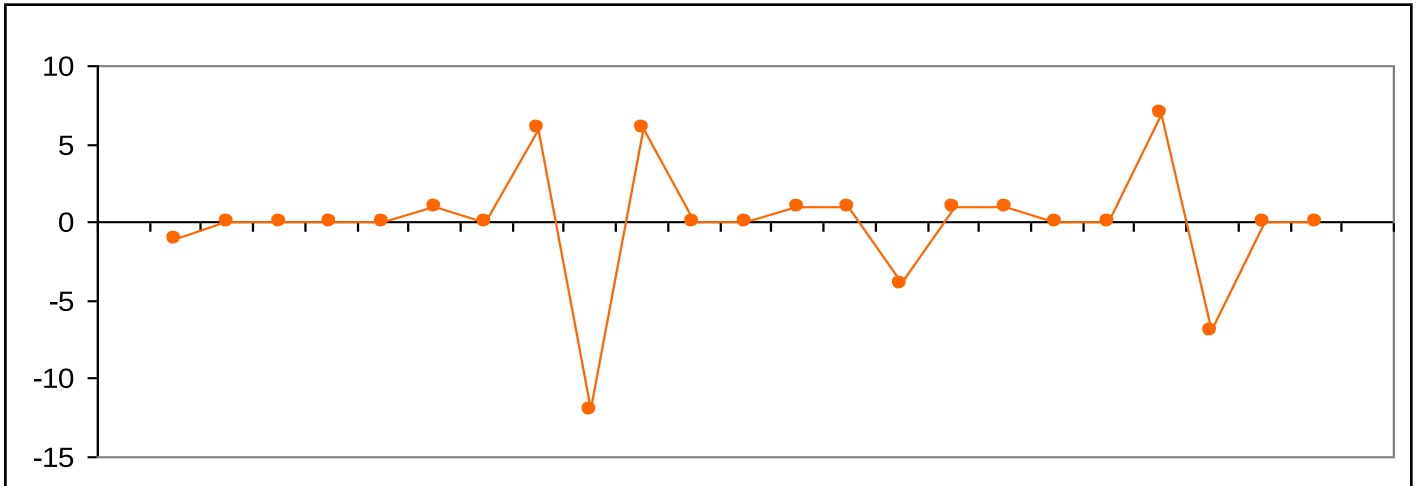
Profile



1st



2nd



Edge-based Approaches – (cont.)

- Isolated Points

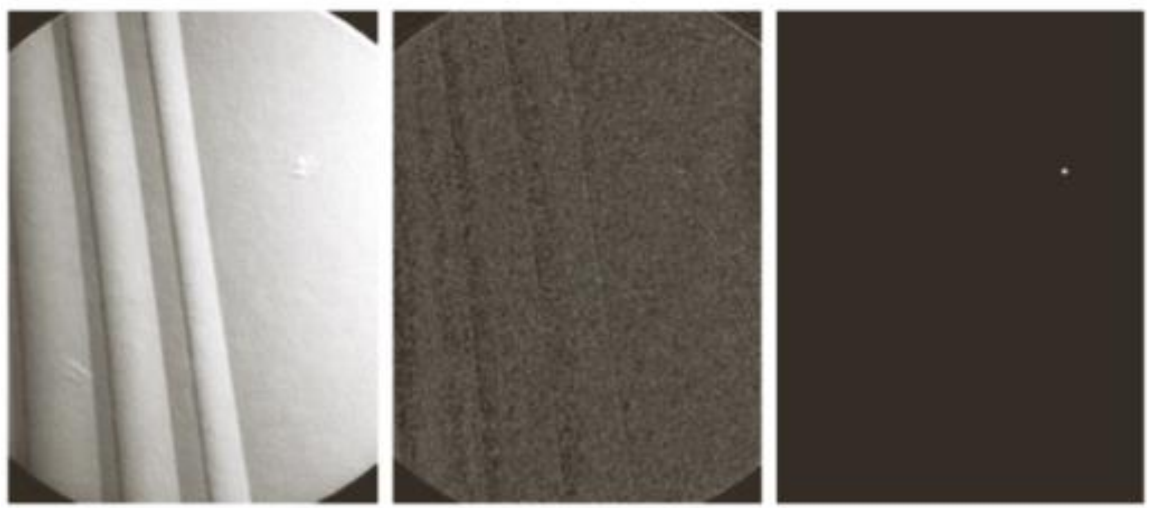
Which detector?

Edge-based Approaches – (cont.)

- Isolated Points

Laplacian + threshold

1	1	1
1	−8	1
1	1	1



Edge-based Approaches – (cont.)

- Lines

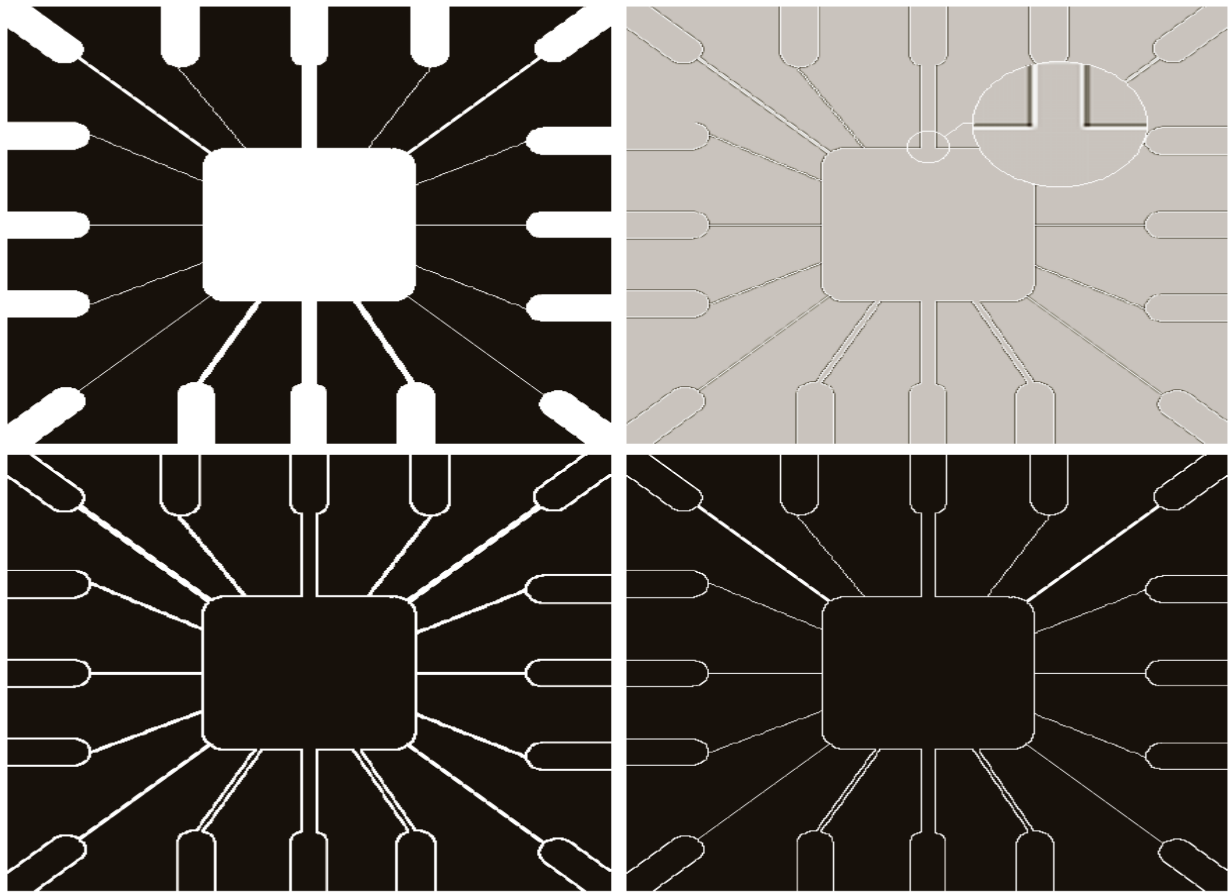
Which detector?

Edge-based Approaches – (cont.)

- Lines

Laplacian + deal with double-line effect.

1	1	1
1	−8	1
1	1	1



Edge-based Approaches – (cont.)

- Lines

Gradient for each direction?

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Edge-based Approaches – (cont.)

- Lines

Gradient for each direction?

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

NO!

Edge-based Approaches – (cont.)

- Lines

Masks for specific direction + threshold the absolute value

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

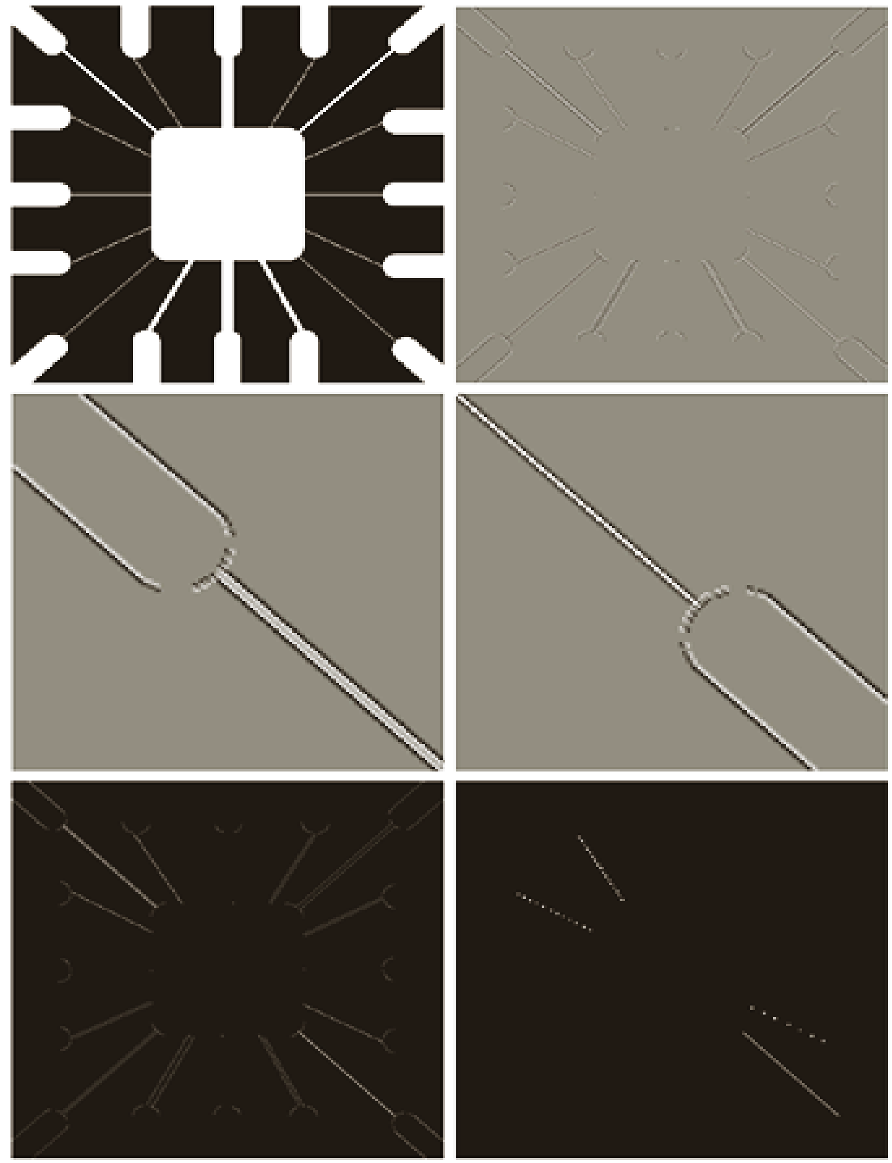
−45°

Edge-based Approaches – (cont.)

- Lines

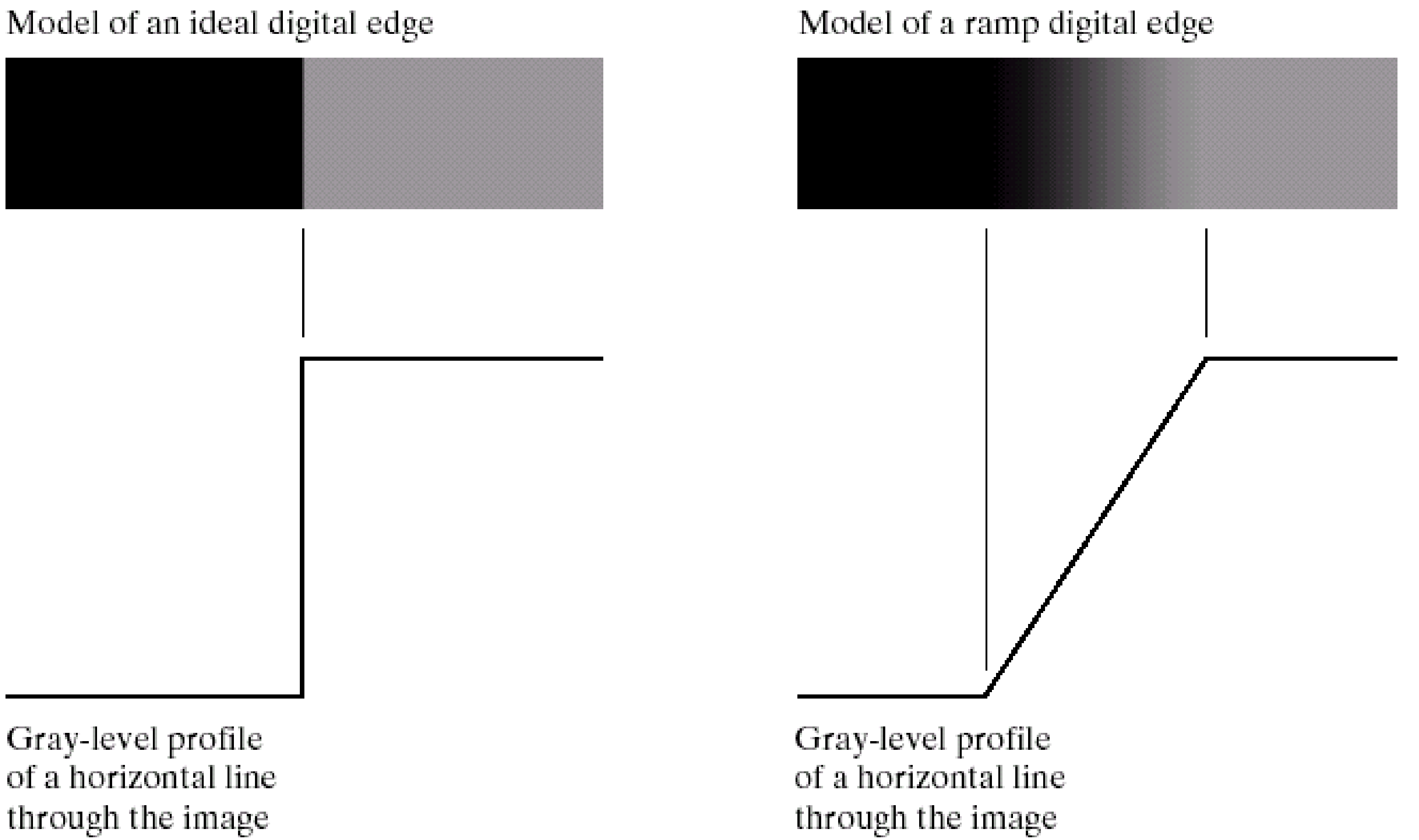
2	−1	−1
−1	2	−1
−1	−1	2

+45°



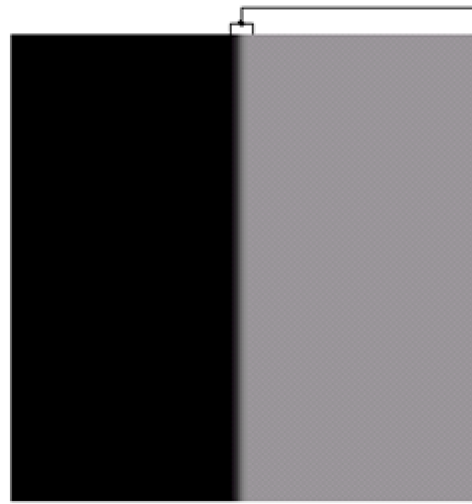
Edge-based Approaches – (cont.)

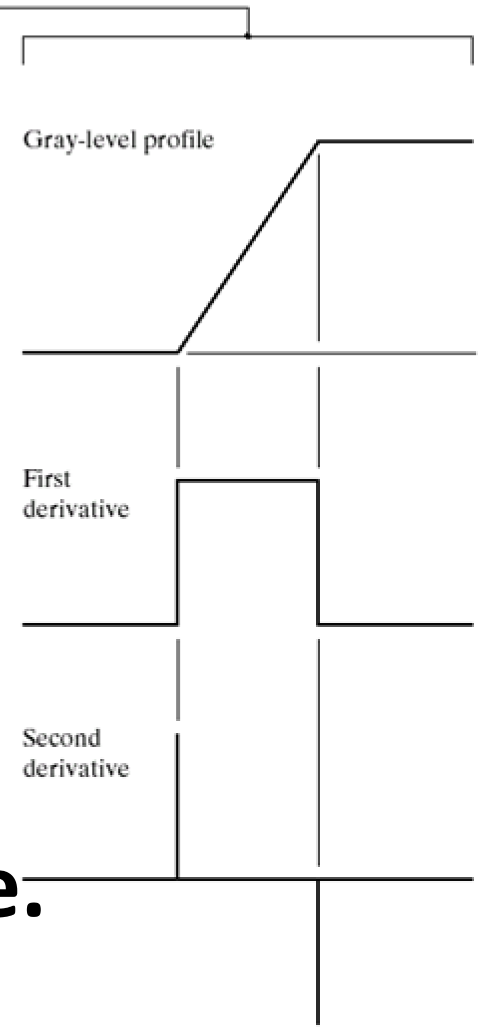
- Edge Models



Edge-based Approaches – (cont.)

- Ramp Edge Model

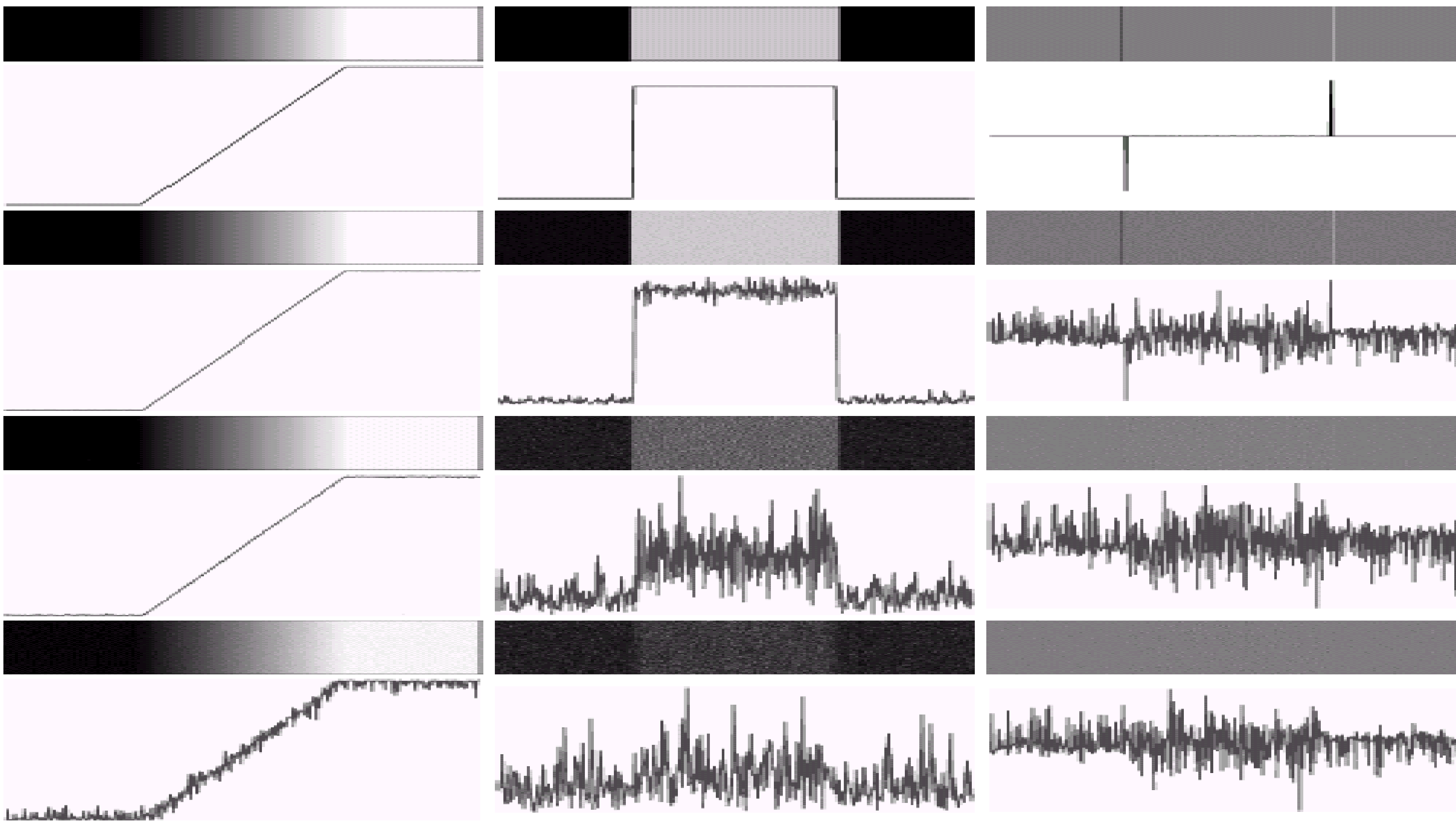




- **Magnitude of 1st derivative.**
- **Zero crossing of 2nd derivative.**

Edge-based Approaches – (cont.)

- **Derivative-based edge detectors are extremely sensitive to noise.**



Edge-based Approaches – (cont.)

- **Image smoothing should be a serious consideration prior to the use of derivatives in applications where noise is likely to be present.**
- **Another approach is to threshold the gradient image.**
- **Sometimes, both smoothing and thresholding of gradient are used.**

Edge-based Approaches – (cont.)

Examples

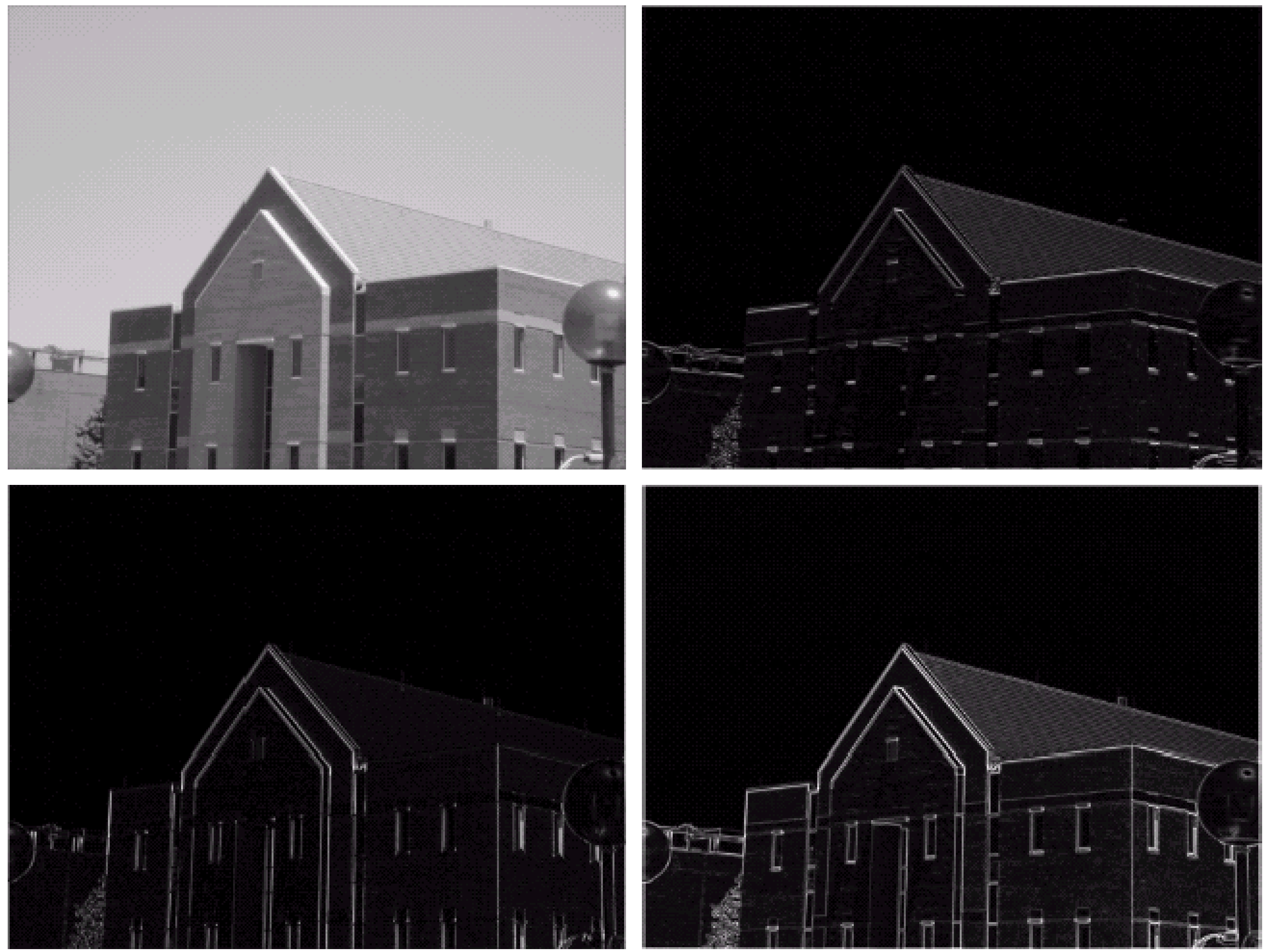
Original Image Horizontal Gradient Component

Vertical Gradient Component Combined Edge Image

Edge-based Approaches – (cont.)

Examples

Smoothed Image   Horizontal Gradient Component



Vertical Gradient Component   Combined Edge Image

Edge-based Approaches – (cont.)

- Marr-Hildreth Edge Detector

• The Laplacian is too sensitive to noise, so usually when used for edge detection, it is combined with a smoothing Gaussian filter

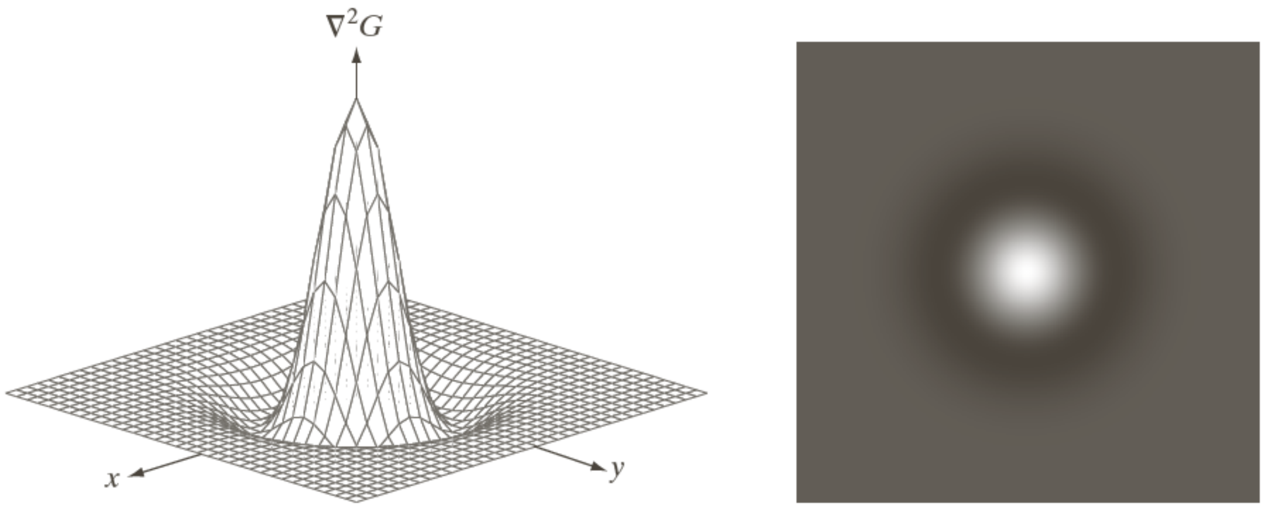
$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

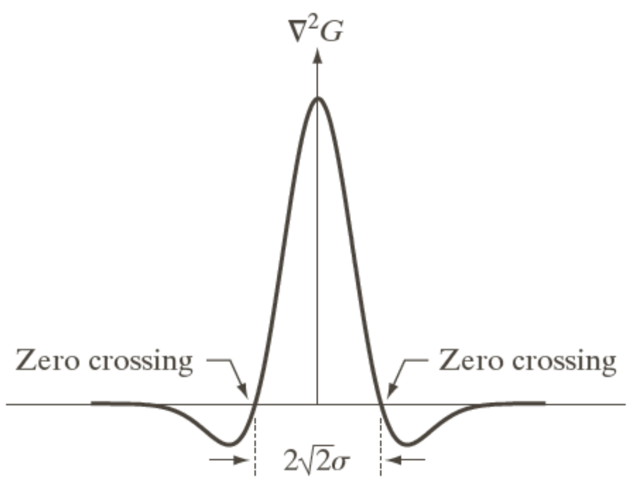
The mask can be of arbitrary size and is called **The Laplacian of a Gaussian (LoG):**

$$\nabla^2 G(x,y) = \left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Edge-based Approaches – (cont.)

- Marr-Hildreth Edge Detector





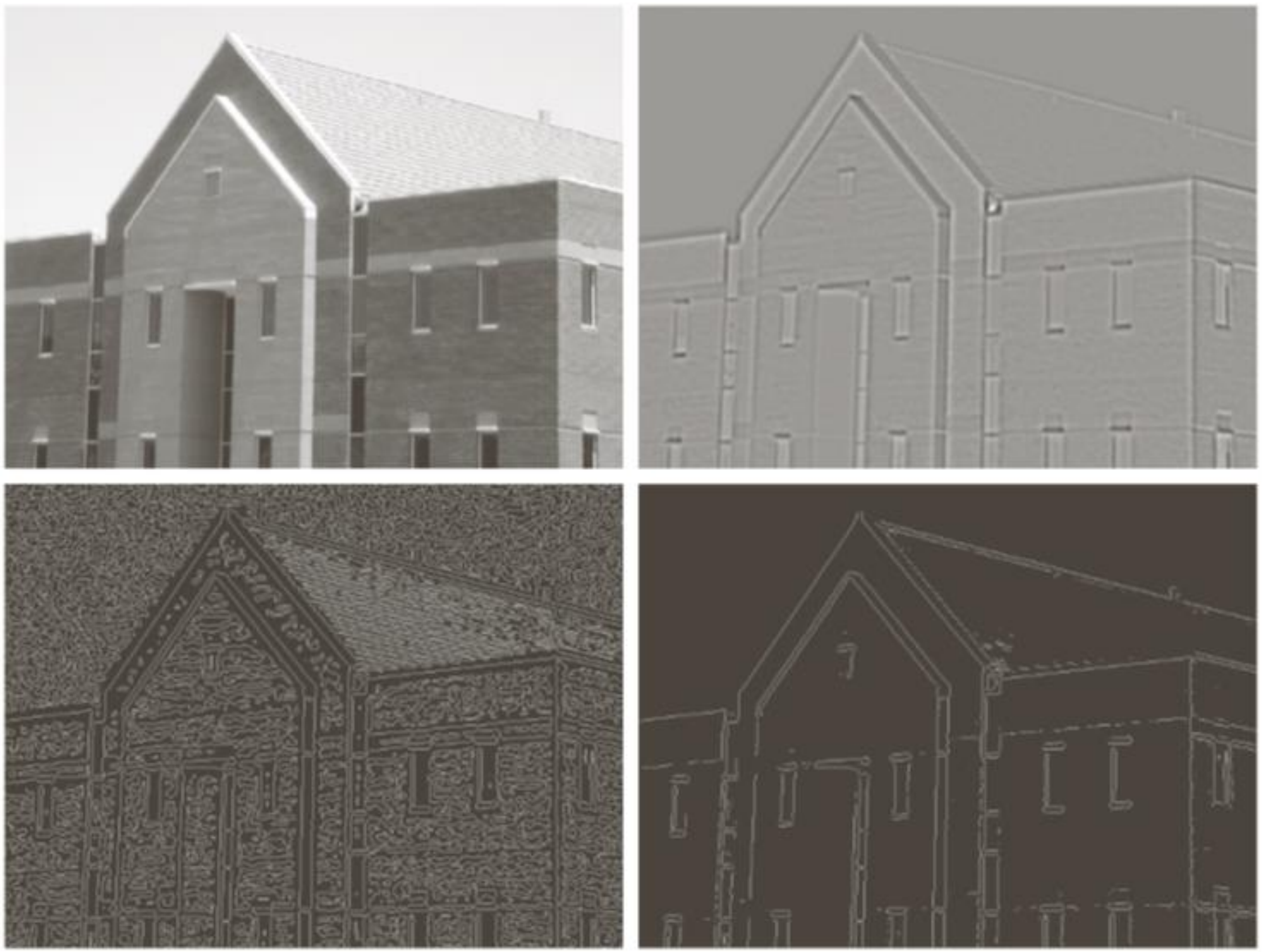
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

a b
c d

FIGURE 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5×5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

Edge-based Approaches – (cont.)

- Marr-Hildreth Edge Detector



a b
c d

FIGURE 10.22 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

Edge-based Approaches – (cont.)

- Canny Edge Detector

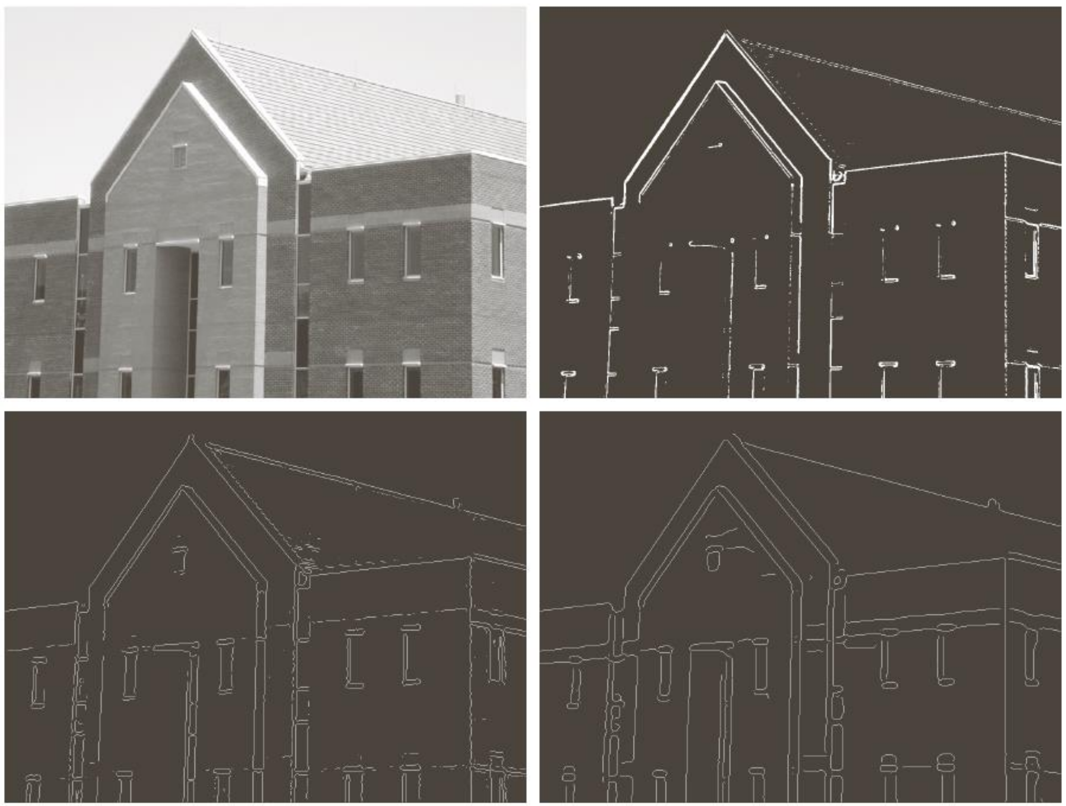
- **A good approximation to the optimal step edge detector and is based on the 1st derivative of a Gaussian**

$$\frac{d}{dx} e^{-\frac{x^2}{2\sigma^2}} = \frac{-x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

- **Involves smoothing, computing the gradient, computing edge angles, and thresholding.**
- **Trade-off between improved performance and speed.**

Edge-based Approaches – (cont.)

- Canny Edge Detector



a b
c d

FIGURE 10.25 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

Edge-based Approaches – (cont.)

- Fundamental steps for edge detection:

1- Image smoothing

2- Edge points detection

3- Edge localization

- Local Processing

- Regional Processing

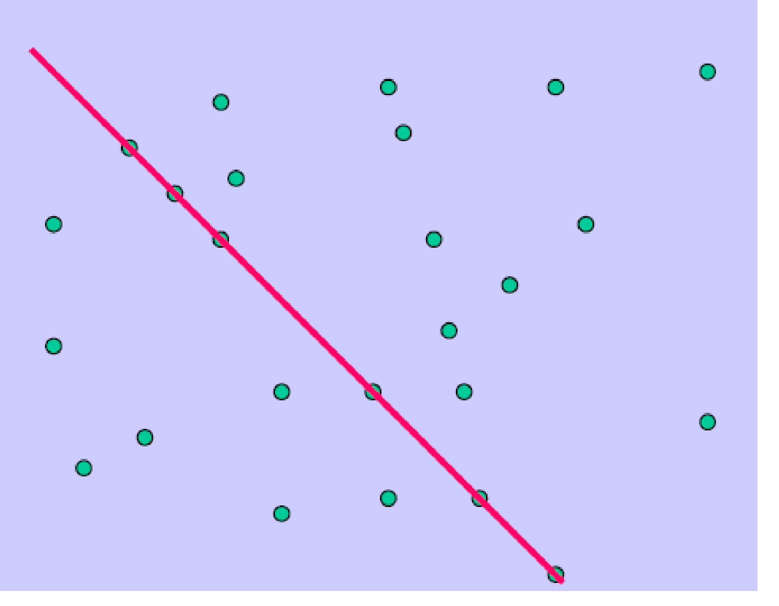
- Global Processing (Hough Transform)

Hough Transform

Edge-based Approaches – (cont.)

- Hough Transform

- **Based on whether a set of candidate pixels lie on curves of a specified shape.**
- **Once detected, these curves form edges or boundaries of regions.**



Edge-based Approaches – (cont.)

- Hough Transform - Line

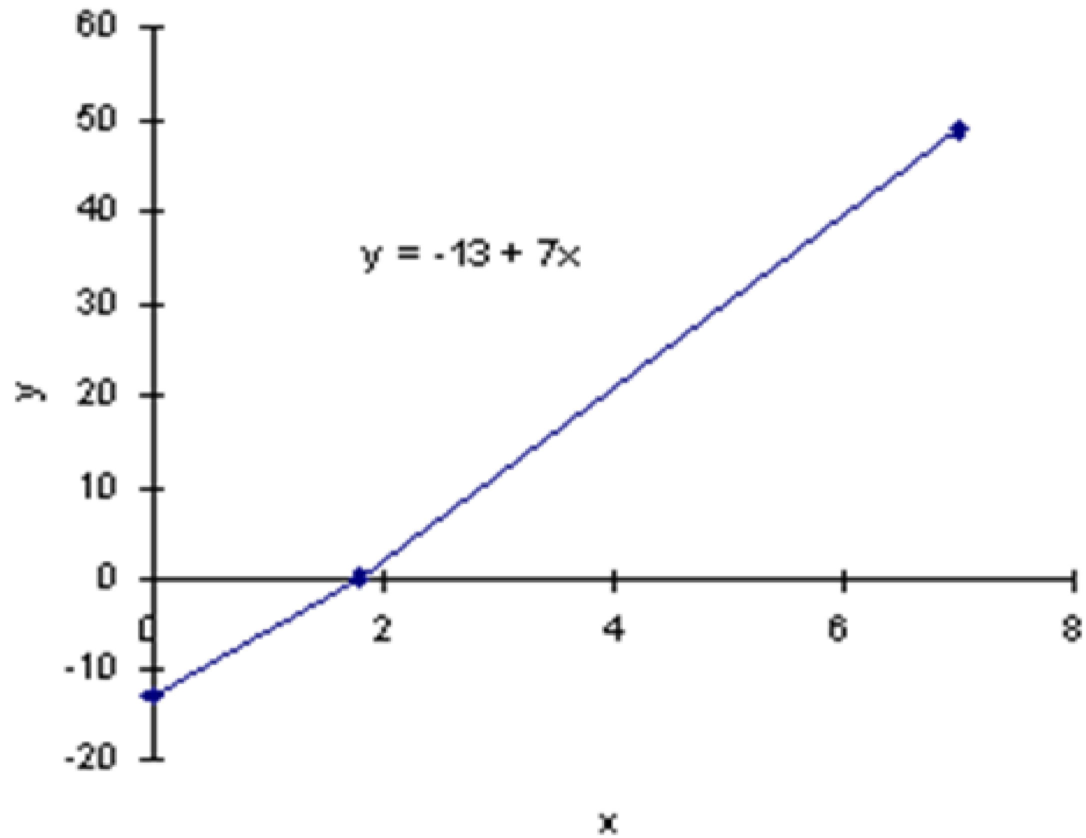
- A line in the xy-plane can be expressed by the slope-intercept form

$$y = ax + b$$

- Where

a = slope

b = y-intercept

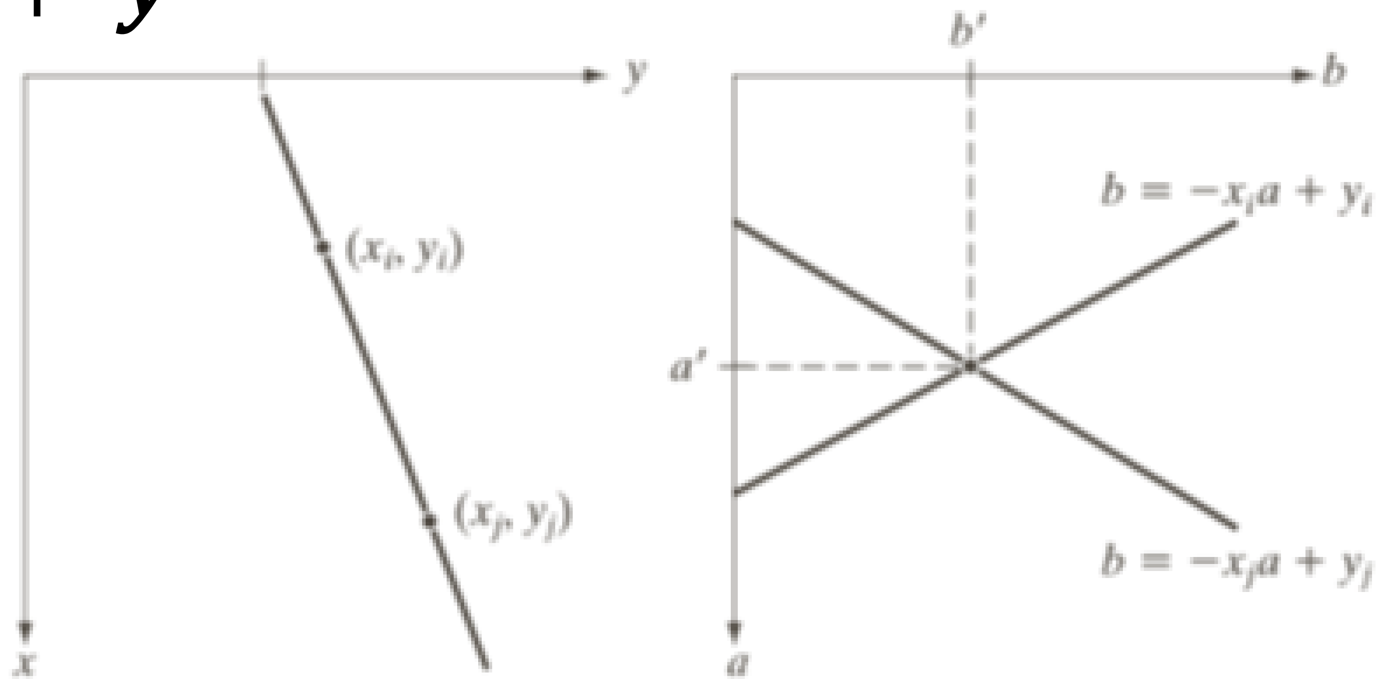


Edge-based Approaches – (cont.)

- Hough Transform - Line

- On the other hand, the line can be represented in the parameter space as:

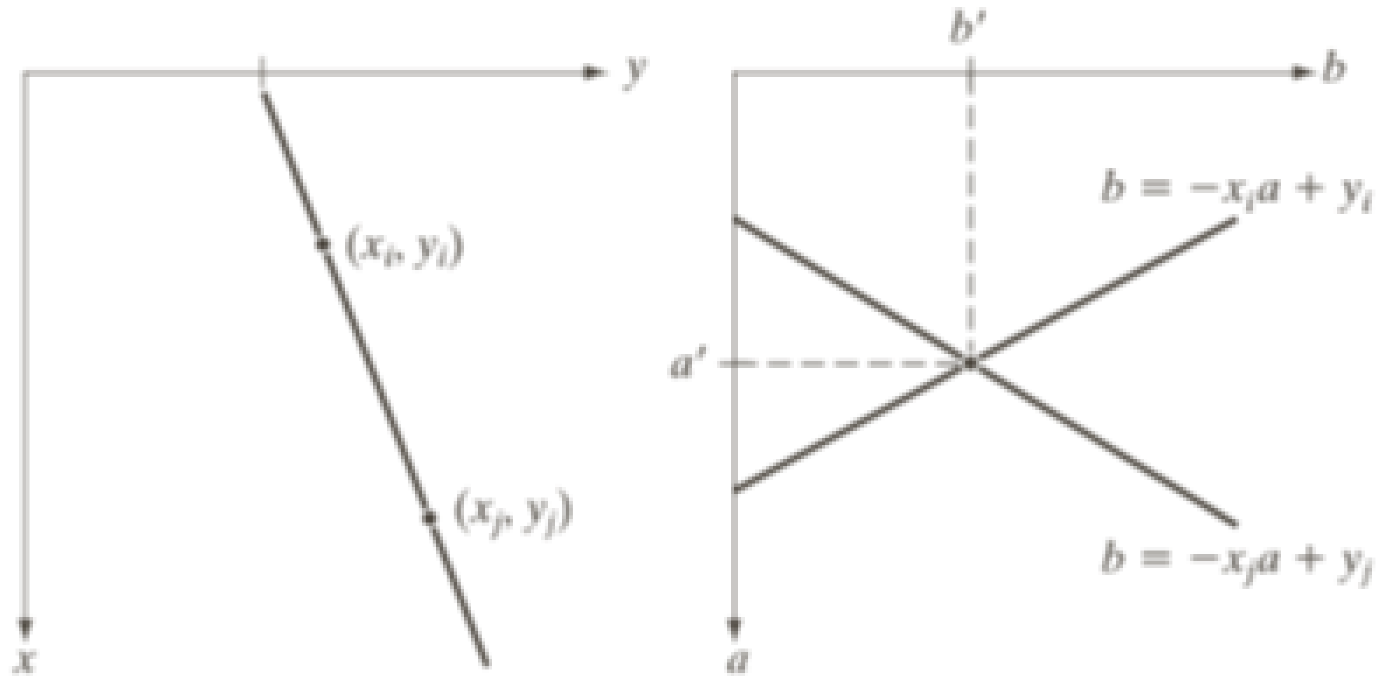
$$b = -ax + y$$



Edge-based Approaches – (cont.)

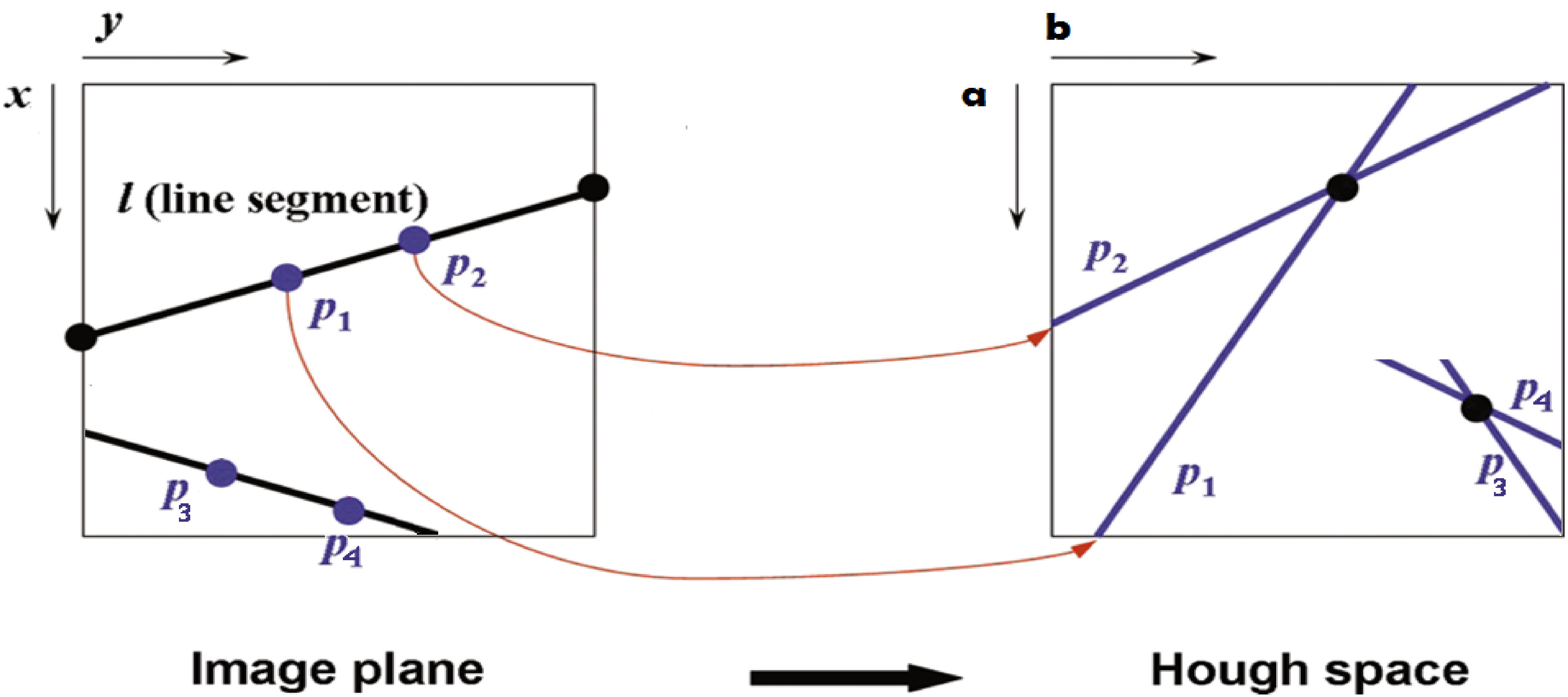
- Hough Transform - Line

- Hence, principal lines in xy-plane can be found by identifying points in ab-plane where large number of lines intersect.



Edge-based Approaches – (cont.)

- Hough Transform - Line

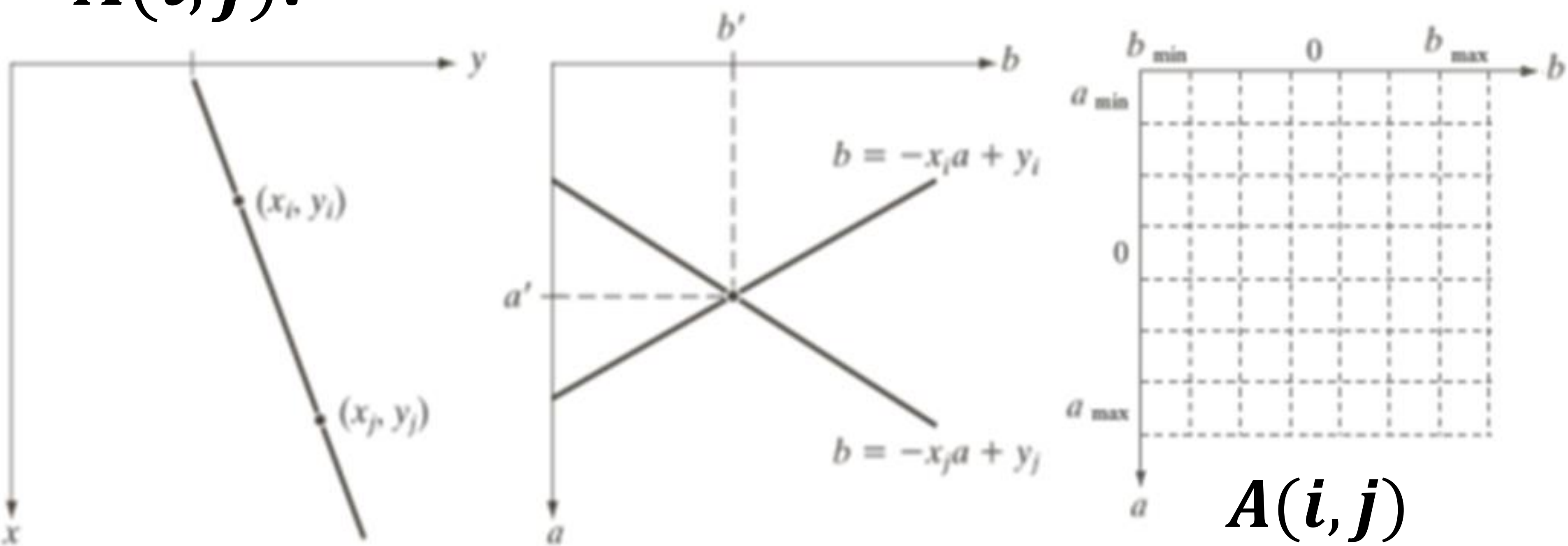


Edge-based Approaches – (cont.)

- Hough Transform - Line

- For each edge point (x_k, y_k), loop every possible a_i, solve for b_j, and accumulate $A(i, j)$.

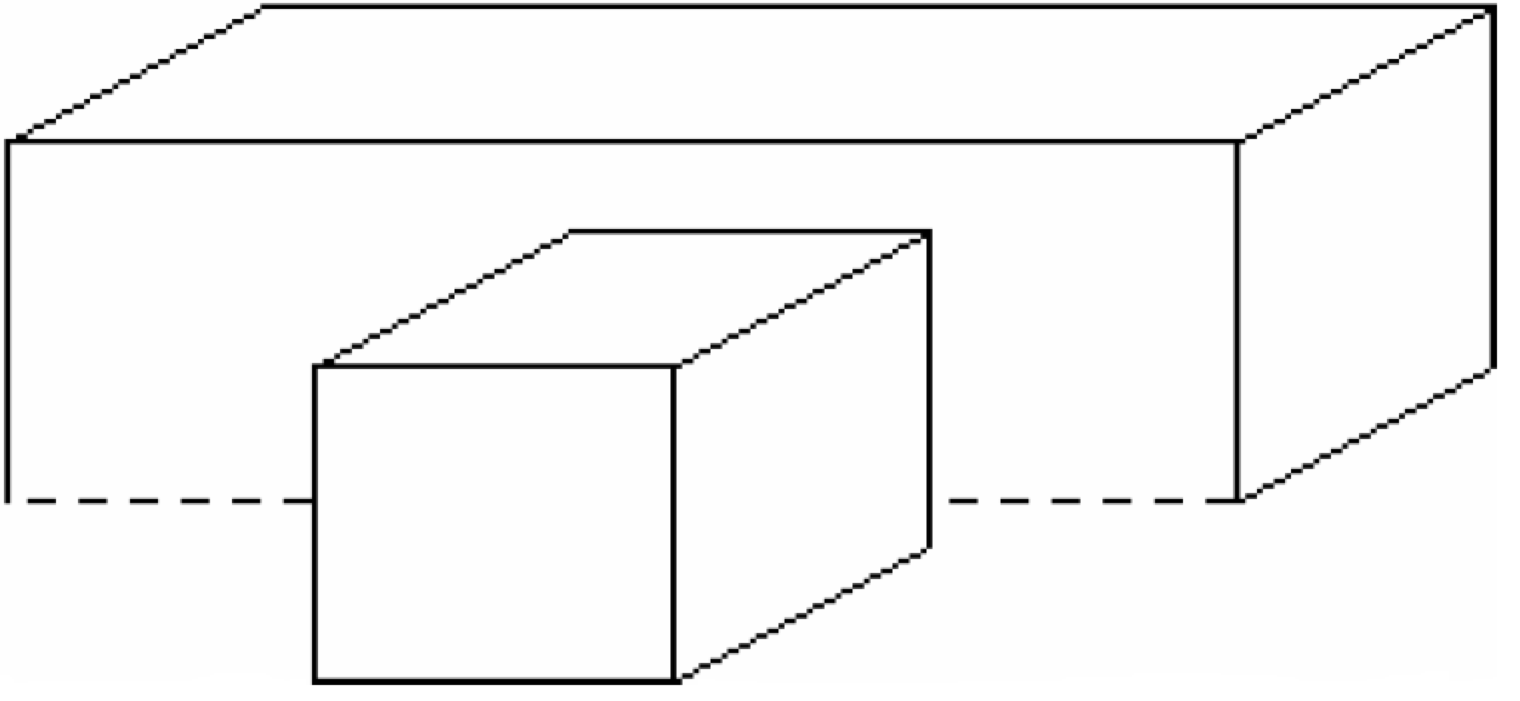
$$b = -ax + y$$



Edge-based Approaches – (cont.)

- Hough Transform - Line

- **Pixels, which lie on the line, need not all be contiguous (HT works even with occlusions).**



Edge-based Approaches – (cont.)

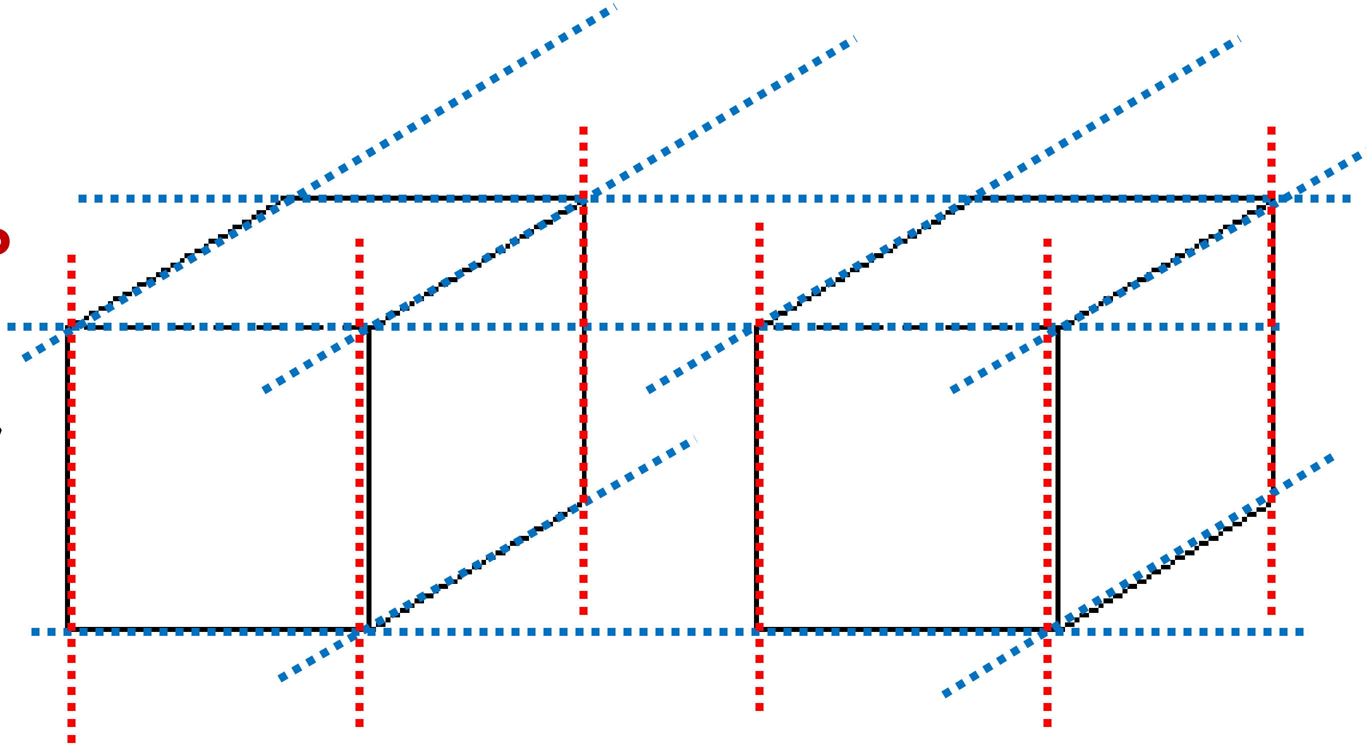
- Hough Transform - Line

- **Problem!**

Vertical lines have slope ~ ∞ and

$$b = -ax + y$$

fails

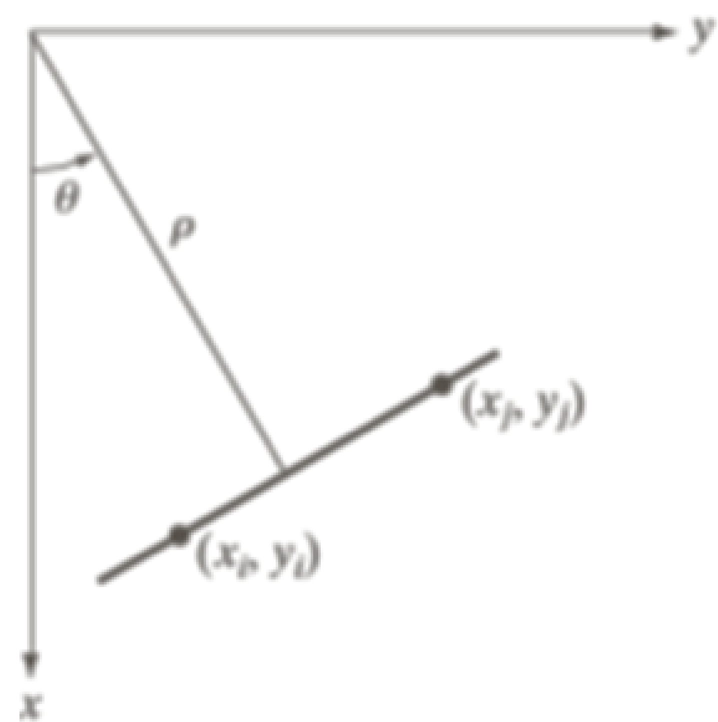


Edge-based Approaches – (cont.)

- Hough Transform - Line

- **Use the normal representation of a line (polar coordinates)**

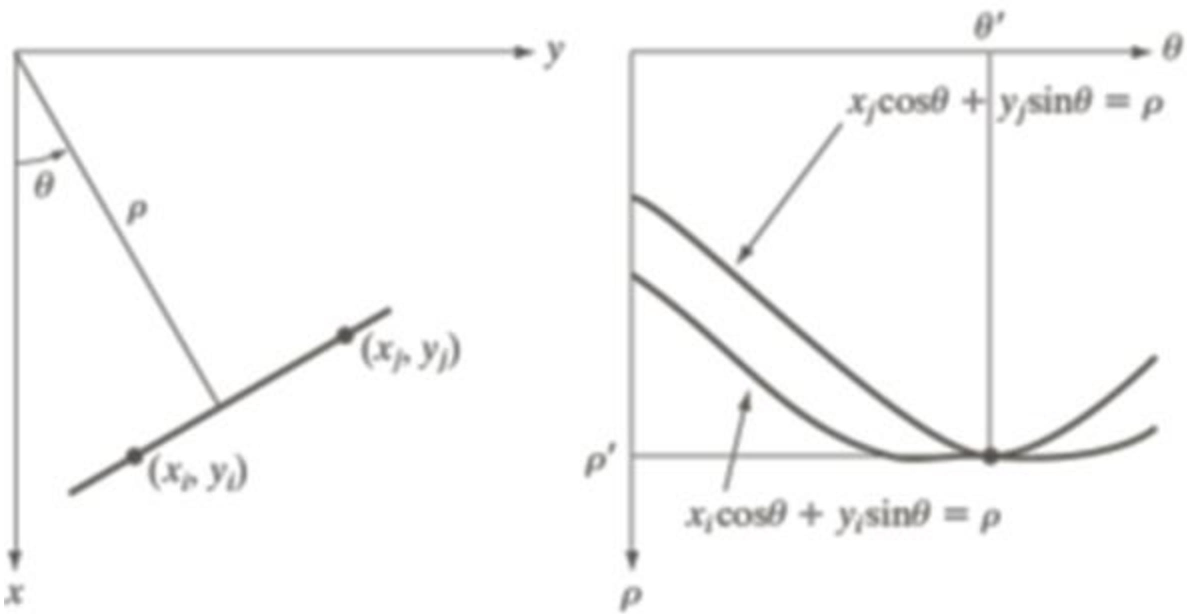
$$\mathbf{y} = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \frac{\rho}{\sin\theta}$$



Edge-based Approaches – (cont.)

- Hough Transform - Line

- **A line can be represented in the parameter space as:** $\rho = x\cos\theta + y\sin\theta$



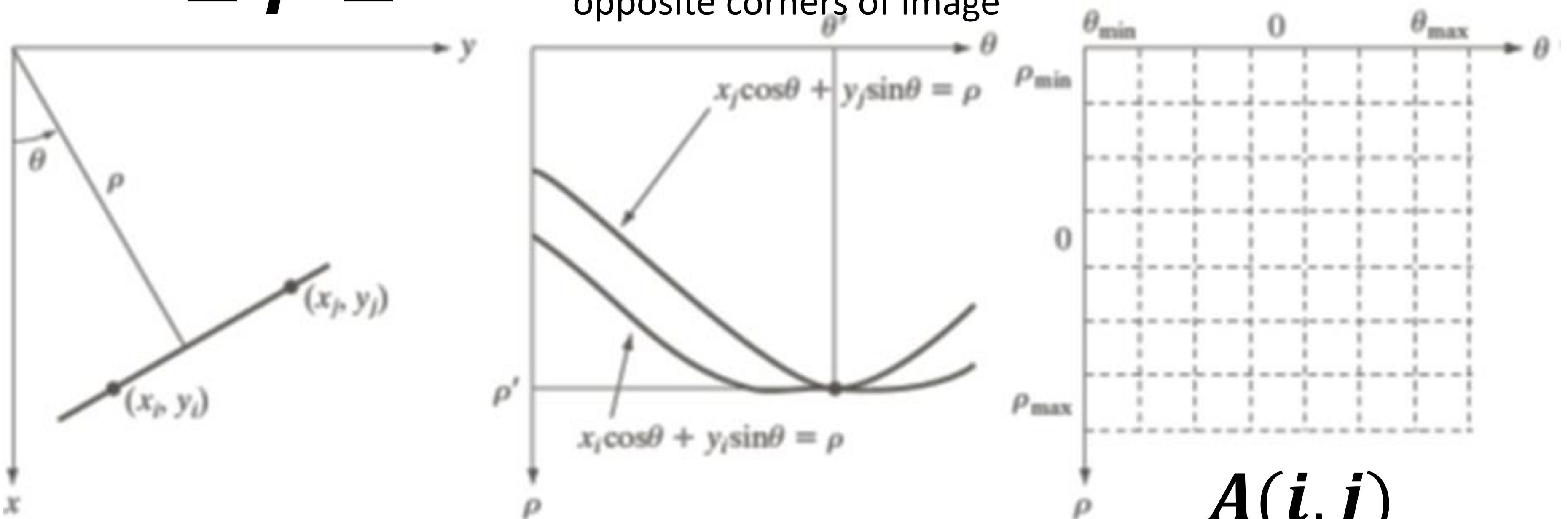
Edge-based Approaches – (cont.)

- Hough Transform - Line

- **Subdivide $\rho\theta$-space into accumulative cells such that $(\theta_{min}, \theta_{max})$: $-90 \le \theta \le 90$**

$$-D \le \rho \le D$$

D=minimum distance between opposite corners of image

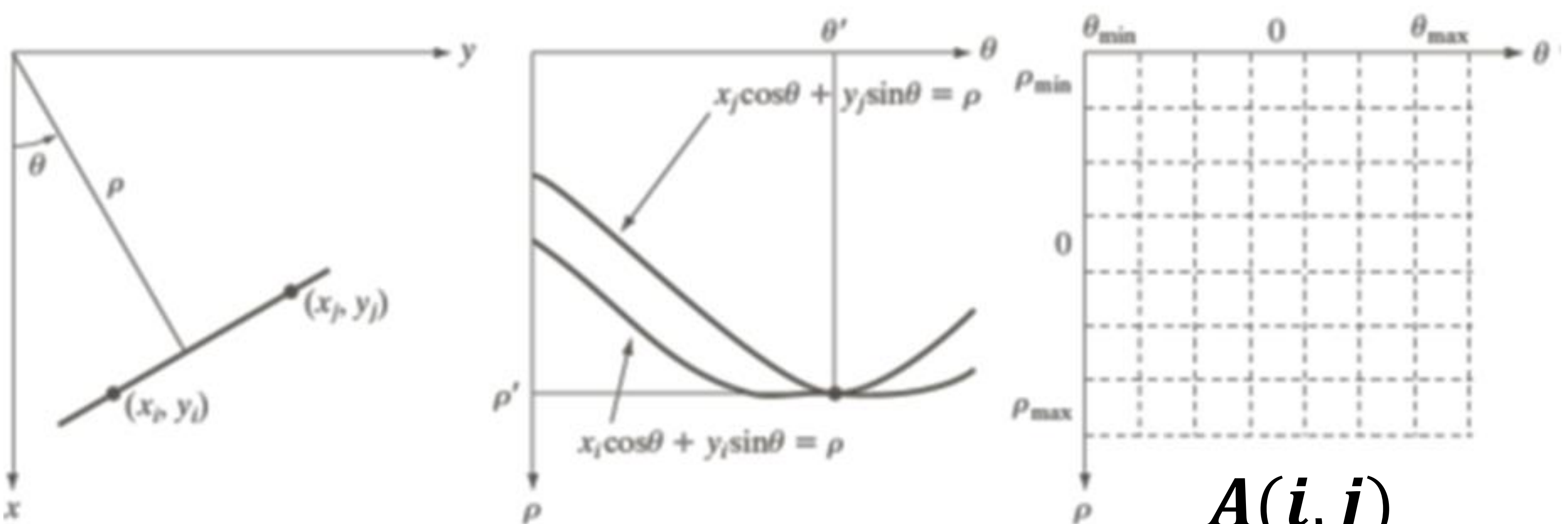


$$A(i, j)$$

Edge-based Approaches – (cont.)

- Hough Transform - Line

- **Number of subdivisions (steps of θ and ρ) determines accuracy of colinearity of points.**

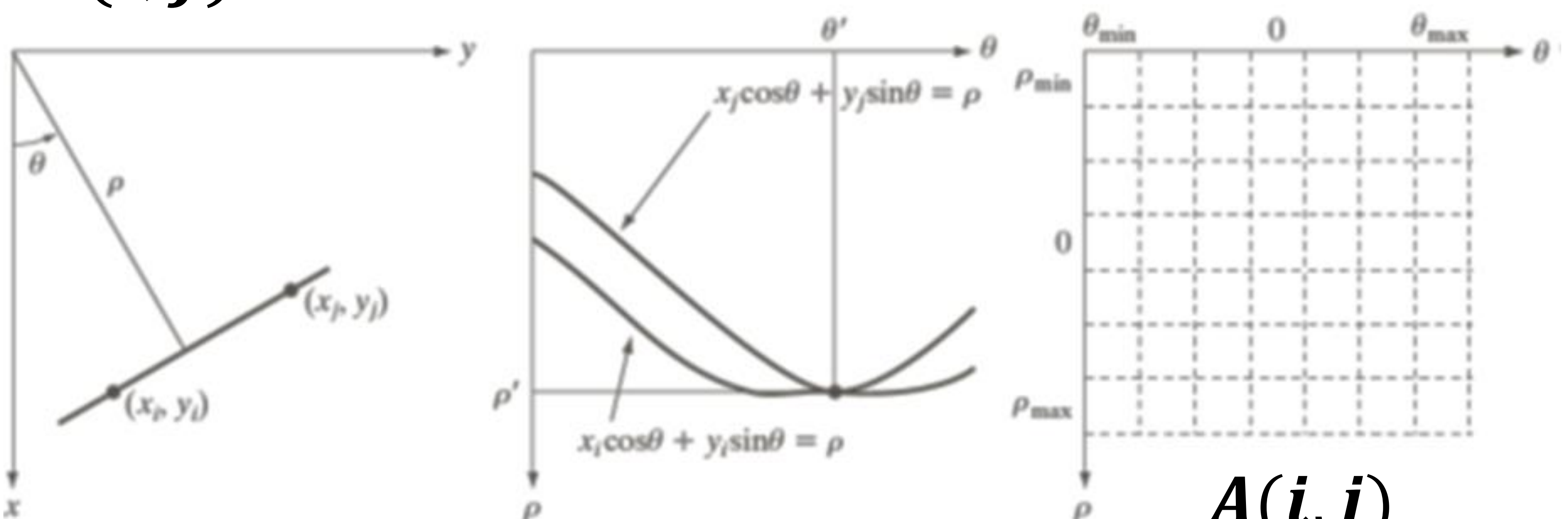


$A(i, j)$

Edge-based Approaches – (cont.)

- Hough Transform - Line

- For each edge point (x_k, y_k), loop every possible θ_i, solve for ρ_j, and accumulate $A(i, j)$.



$A(i, j)$

Edge-based Approaches – (cont.)

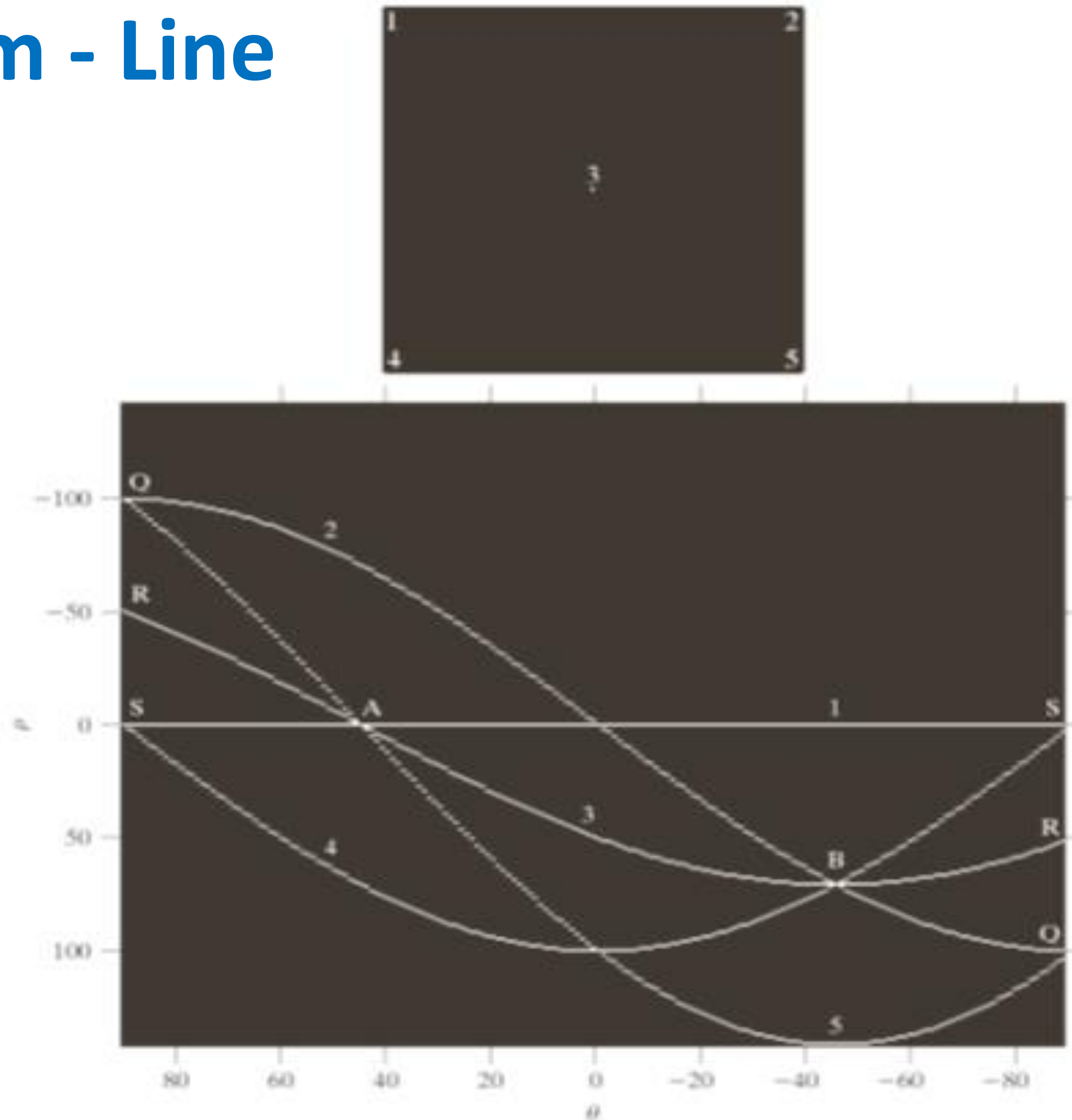
- Hough Transform - Line

- **Finally, examine continuity of pixels in high count accumulator cells; usually by the distance between disconnected pixels.**
- **Gaps are bridged if length of gap is less than a certain value.**

Edge-based Approaches – (cont.)

- Hough Transform - Line

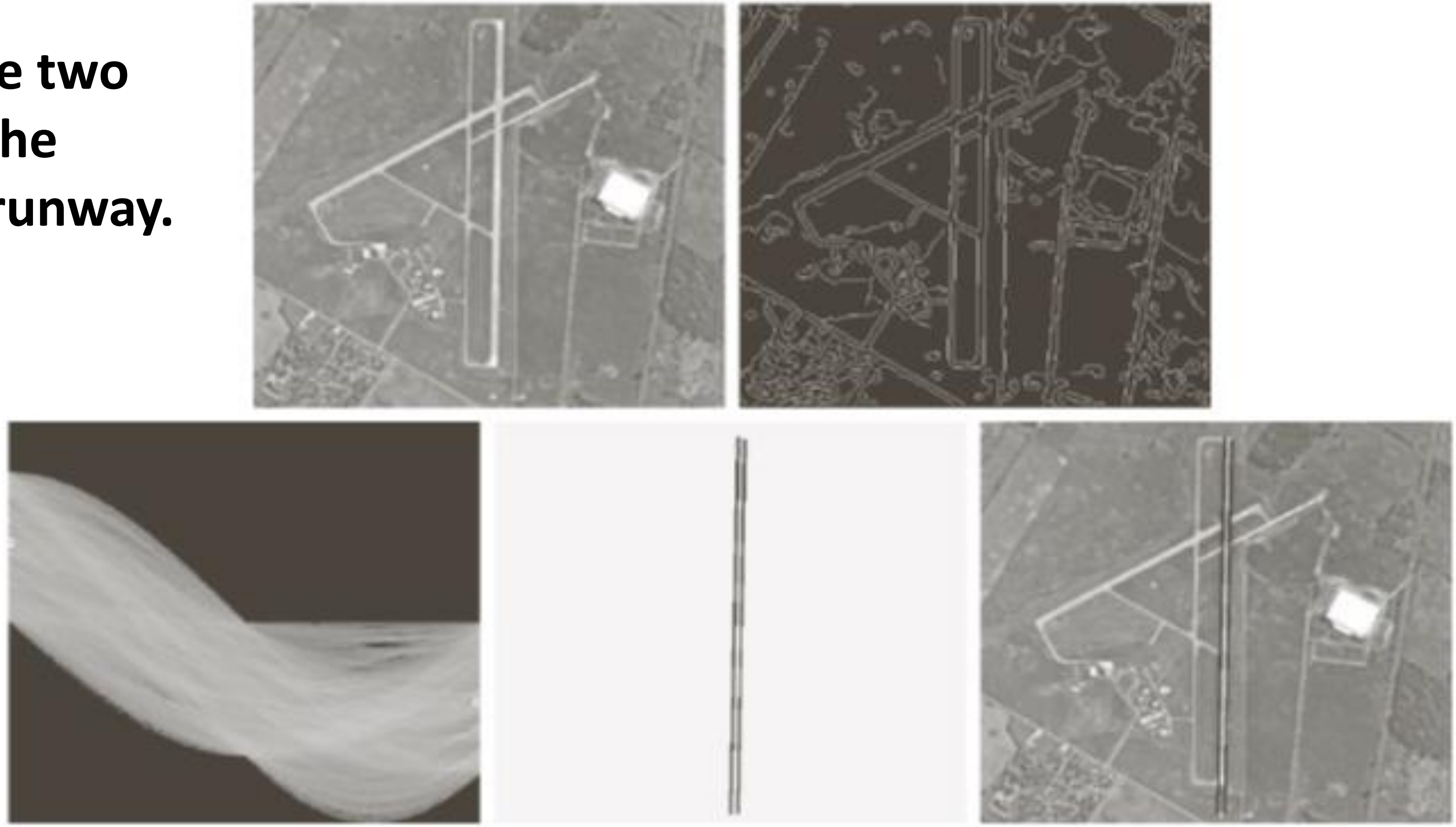
Example



Edge-based Approaches – (cont.)

- Hough Transform - Line

Extract the two edges of the principal runway.

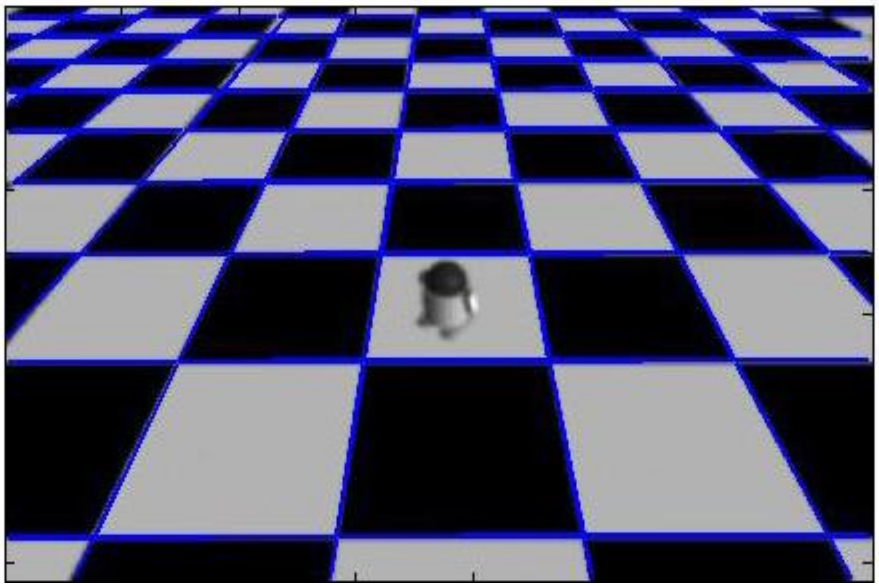


a b
c d e

FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed on the original image.

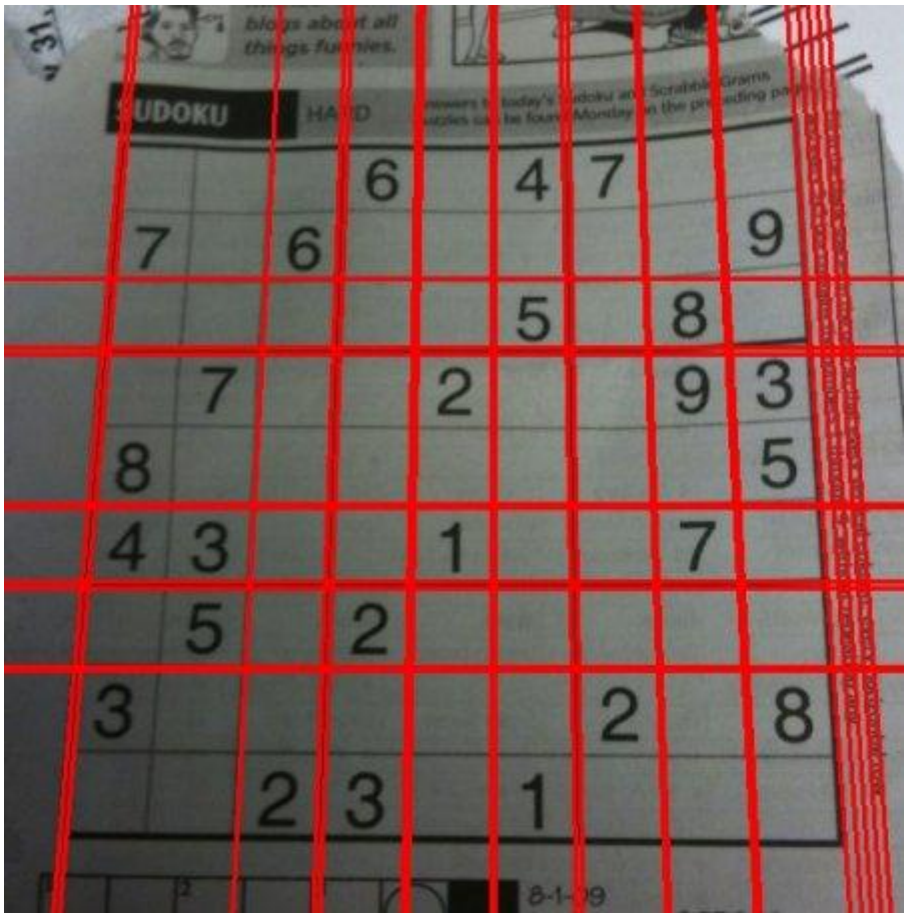
Edge-based Approaches – (cont.)

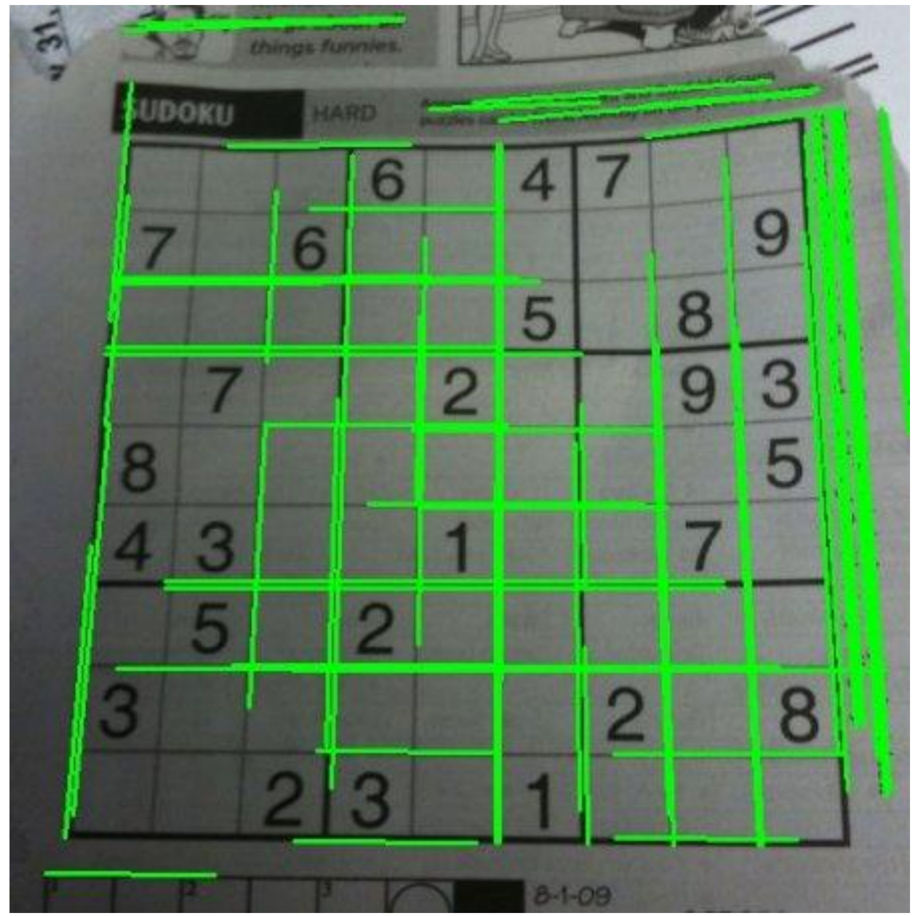
- Hough Transform - Line



Edge-based Approaches – (cont.)

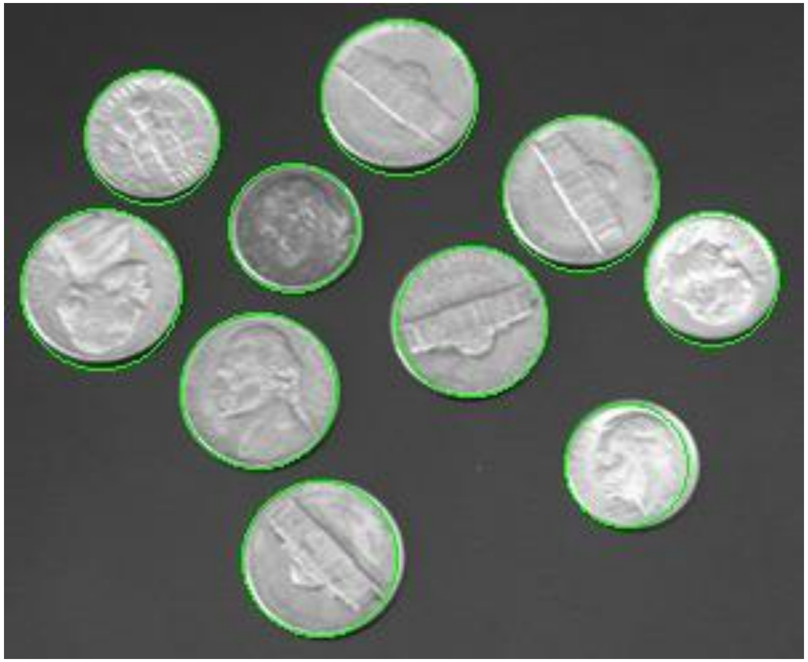
- Hough Transform vs Probabilistic HT





Edge-based Approaches – (cont.)

- Hough Transform - Circular



Other Segmentations Methods

Active Contours

Snakes, Scissors, Level Sets

Split and Merge

Watershed, region split/merge, graph

Mean Shift

Graph Cuts

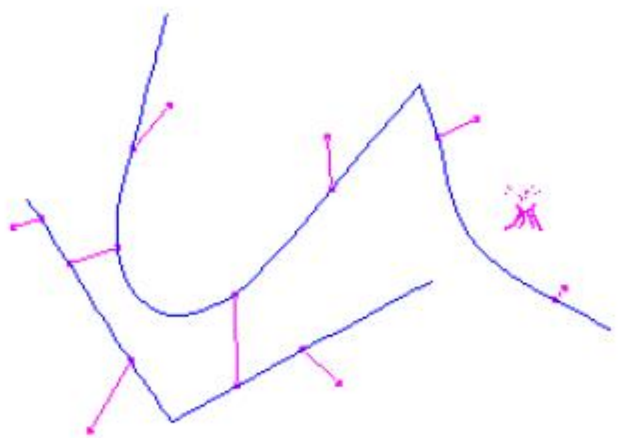
Examples

(a) active contours
(b) level sets
(c) graph-based merging
(d) mean shift (e) texture and intervening contour-based normalized cuts
(f) binary MRF solved using graph cuts

Richard Szeliski, Computer Vision Algorithms and Applications, Springer 2010. Ch. 5

Examples – (cont.)

Active Contours - Snakes



(a)

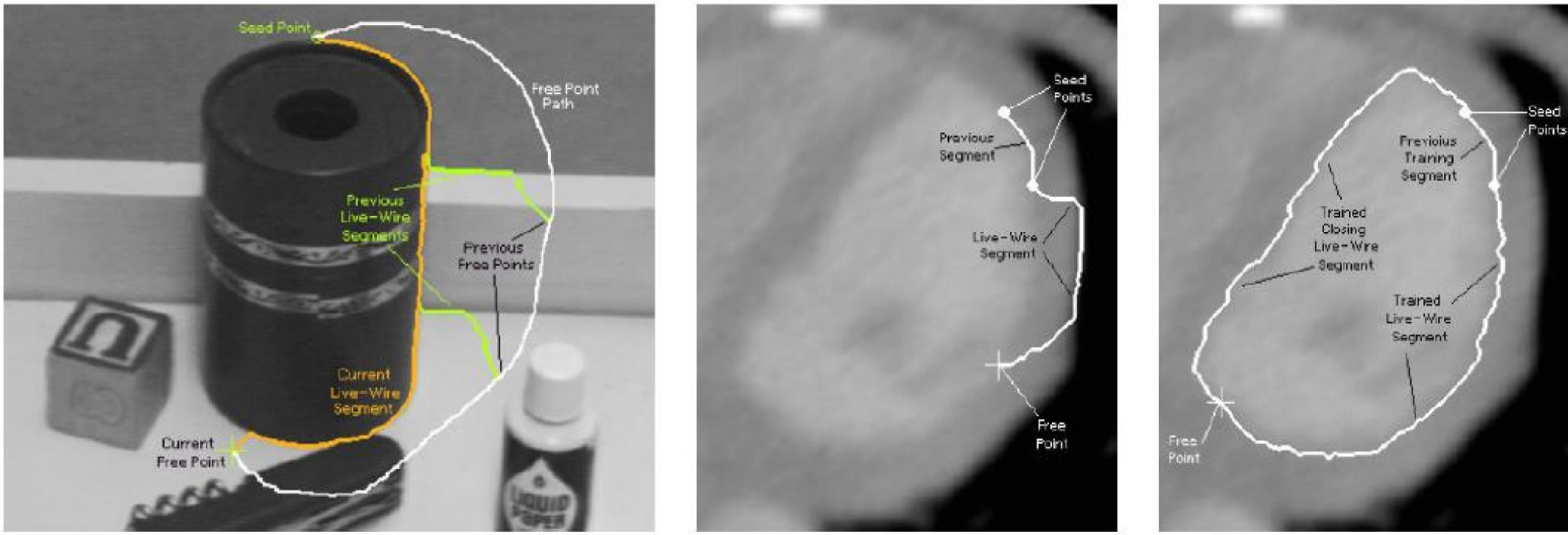
(b)

(a) the "snake pit" for interactively controlling shape; (b) lip tracking.

Richard Szeliski, Computer Vision Algorithms and Applications, Springer 2010. Ch. 5

Examples – (cont.)

Active Contour - Scissors



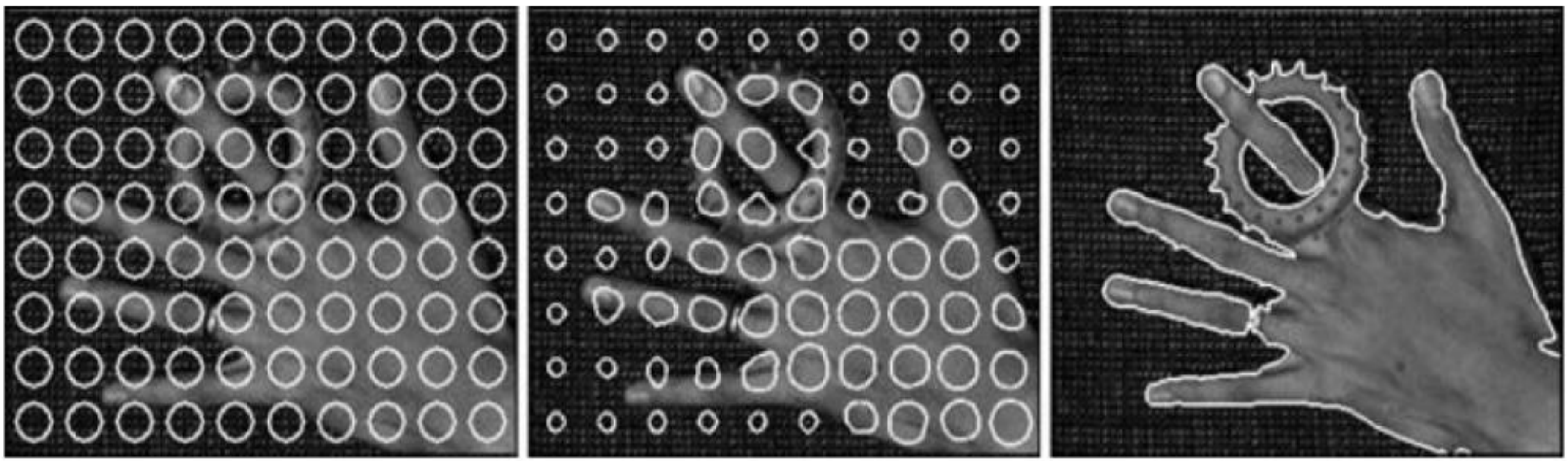
(a) (b) (c)

(a) as the mouse traces the white path, the scissors follow the orange path along the object boundary (the green curves show intermediate positions).
(b) regular scissors can sometimes jump to a stronger (incorrect) boundary.
(c) after training to the previous segment, similar edge profiles are preferred.

Richard Szeliski, Computer Vision Algorithms and Applications, Springer 2010. Ch. 5

Examples – (cont.)

Active Contours - Level Sets

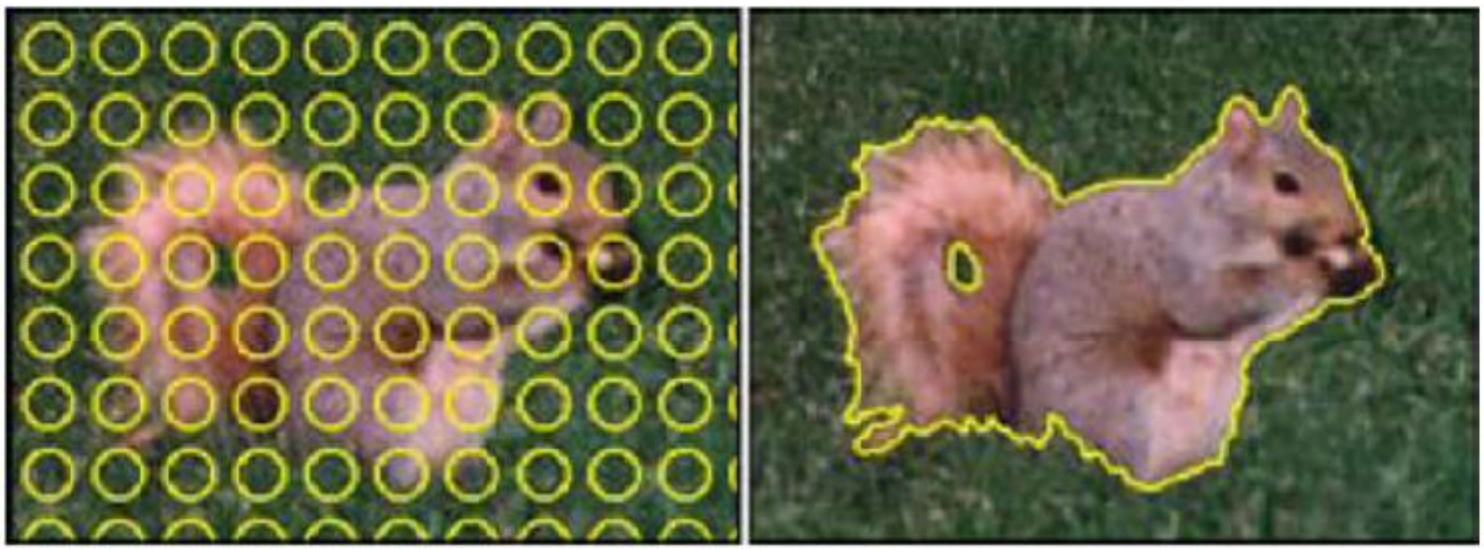


The initial circles evolve towards an accurate segmentation of foreground and background, adapting their topology as they evolve. (Grayscale).

Richard Szeliski, Computer Vision Algorithms and Applications, Springer 2010. Ch. 5

Examples – (cont.)

Active Contours - Level Sets



The initial circles evolve towards an accurate segmentation of foreground and background, adapting their topology as they evolve. (Color).

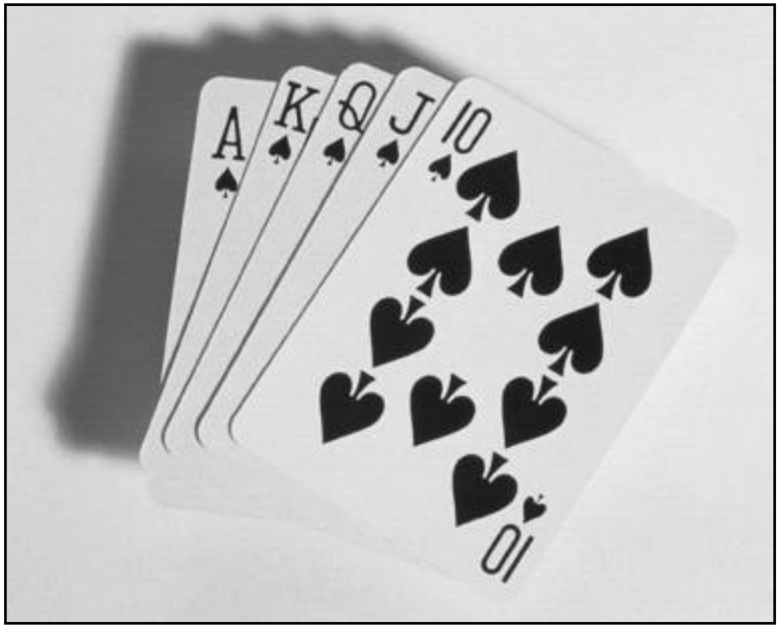
Richard Szeliski, Computer Vision Algorithms and Applications, Springer 2010. Ch. 5

Thresholding

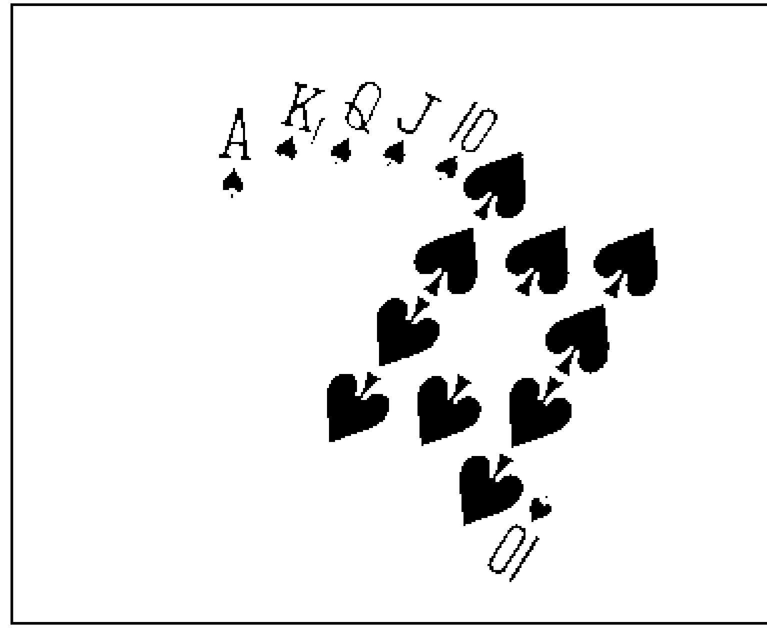
Thresholding

- **Thresholding is usually a first or intermediate step in any segmentation approach.**

- **Segmenting an image is based on direct intensity values or properties of these values.**

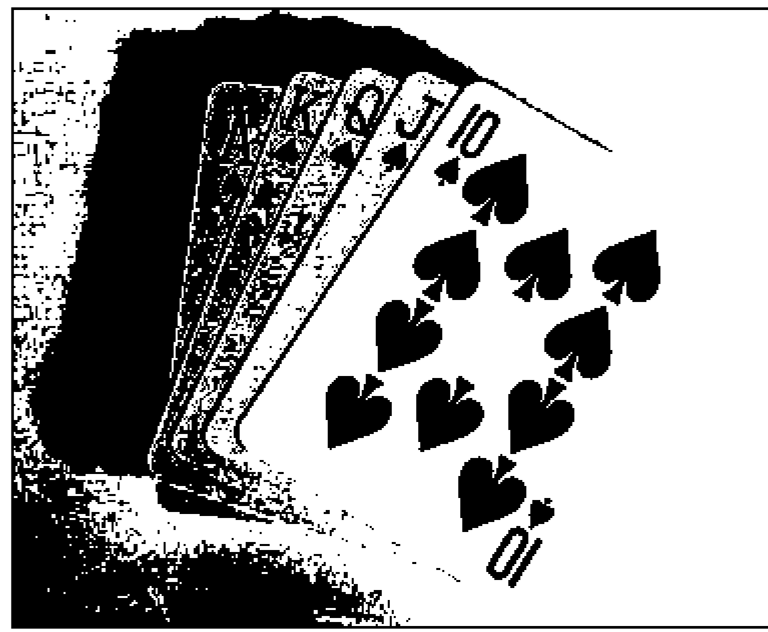


Original Image



Thresholded Image

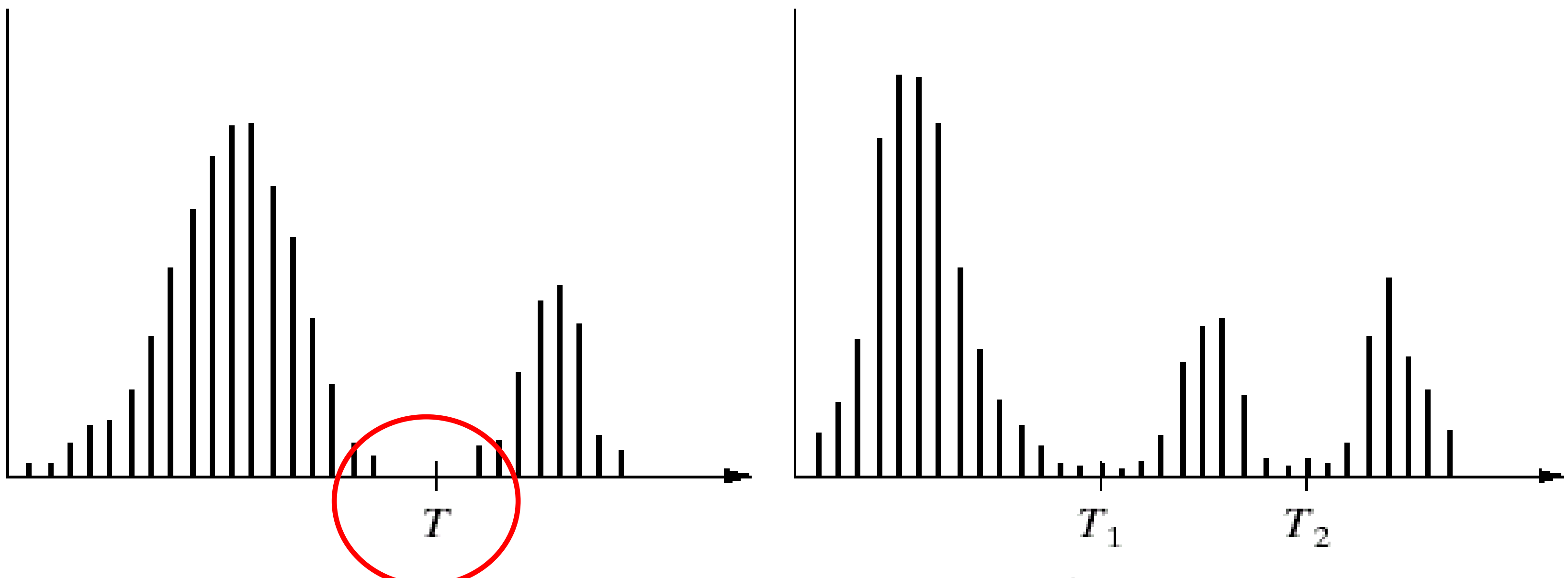
Threshold Too Low



Threshold Too High

Thresholding – (cont.)

- **Thresholding based on the histogram of the image differentiates between modes.**



- **When intensity distribution of objects and background are sufficiently distinct, a single values can be used to segment.**

Thresholding – (cont.)

Thresholding can be viewed as

- $T = T[x, y, p(x, y), f(x, y)]$

$p(x, y)$: some neighborhood property.

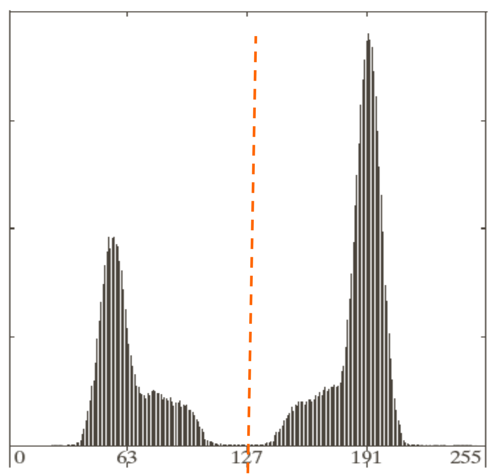
$T = T[f(x, y)]$	T is applied to the entire image → **Global thresholding**	Single constant threshold
$T = T[p(x, y), f(x, y)]$	T depends on properties of neighborhood → **Local thresholding**	Variable thresholds
$T = T[x, y, p(x, y), f(x, y$	T depends on spatial coordinates → **Dynamic/adaptive thresholding**	

Thresholding – (cont.)

Basic Global Thresholding

- **The success of this technique depends very strongly on how well the histogram can be partitioned.**

$$g(x,y)=\begin{cases}1\, if\;\; f(x,y)>T\\ 0\, if\;\; f(x,y)\le T\end{cases}$$

Thresholding – (cont.)

Basic Global Thresholding

- **Automatic procedure:**

Not so auto

  - Select initial T.
  - Segment using $T \rightarrow G1$ and $G2$.
  - Compute means $m1$ and $m2$.
  - Compute new threshold $T = ½(m1 + m2)$.
  - Repeat until the difference between successive T is $< \Delta T$.
- **Choice of T.**

Thresholding – (cont.)

Basic Global Thresholding

- **Thresholding viewed as decision-theory problem aims to minimize average error incurred in assigning pixels to two or more groups (classes).**
- **Bayes decision rule.**
- **Requires**
  - **PDF of intensity levels of each class.**
  - **Probability that each class occurs in an application**

Thresholding – (cont.)

Basic Global Thresholding

- Thresholding viewed as decision-theory problem aims to minimize average error incurred in assigning pixels to two or more groups (classes).
- Bayes decision rule.
- Requires
  - PDF of intensity levels of each class.
  - Probability that each class occurs in an application

Not trivial
→ assume the form of PDF, e.g. Gaussian

Thresholding – (cont.)

Basic Global Thresholding

- Thresholding viewed as decision-theory problem aims to minimize average error incurred in assigning pixels to two or more groups (classes).
- Bayes decision rule.
- Requires
  - PDF of intensity levels of each class.
  - Probability that each class occurs in an application

Not trivial
→ assume the form of PDF, e.g. Gaussian
→ Still not practical

Thresholding – (cont.)

Optimum Global Thresholding (Otsu)

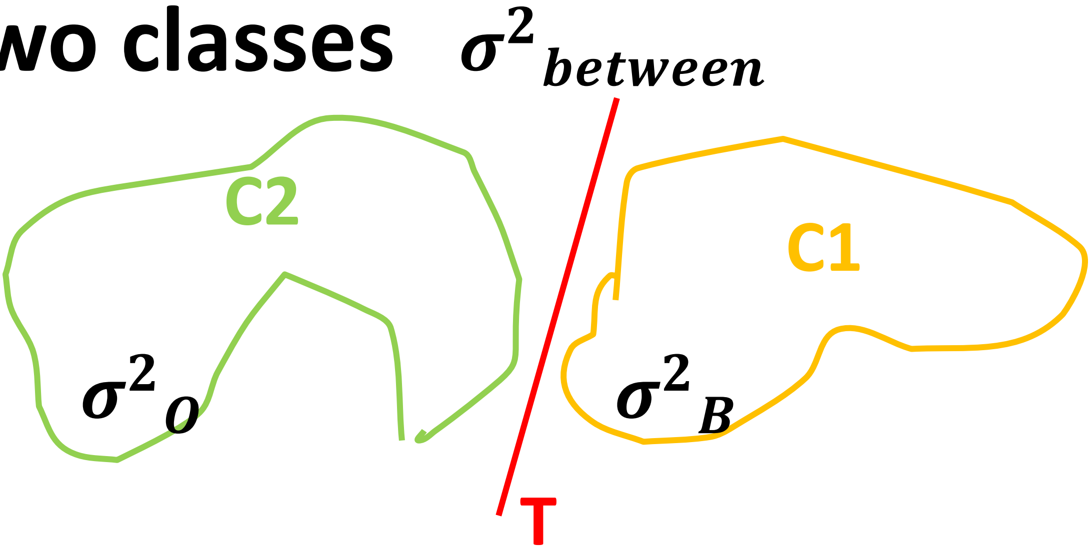
- **A measure of region homogeneity is the variance (i.e., regions with high homogeneity will have low variance).**
- **Otsu's method selects the threshold by minimizing the within-class variance of the two groups of pixels separated by the thresholding operator.**
- **Equivalently, it maximize the between-class variance.**
- **Section 10.3.3 for reference and math. model.**

Thresholding – (cont.)

Optimum Global Thresholding (Otsu)

- $\sigma^2 = \sigma^2{}_{within} + \sigma^2{}_{between}$

- **Within-class variance = weighted sum of the variances of the two classes**



Thresholding – (cont.)

Optimum Global Thresholding (Otsu)

Minimize the within-class variance

$$\boxed{\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)}$$

a lot of computations

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \qquad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Thresholding – (cont.)

Optimum Global Thresholding (Otsu)

Easier to minimize the between-class variance

$$\sigma^2 = \underbrace{\sigma_w^2(t)}_{\text{Within-class}} + \underbrace{q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2}_{\text{Between-class}}$$

- Usually done with a simple iterative approach.

- Other optimization methods might be used.

Check end of slides and textbook Section 10.3.3 for more details

Note

Expected Value for a Discrete Random Variable

$$E(X) = \sum x_i p_i$$

x_i= value of the i$^{\text{th}}$ outcome
p_i = probability of the i$^{\text{th}}$ outcome

Conceptual Formulas

Variance for a Discrete Random Variable

$$\sigma^2 = \sum[(x_i - \mu)^2 p_i]$$

Standard Deviation for a Discrete Random Variable

$$\sigma = \sqrt{\sum[(x_i - \mu)^2 p_i]}$$

x_i= value of the i$^{\text{th}}$ outcome
$\mu = E(X) = \sum x_i p_i$
p_i = probability of the i$^{\text{th}}$ outcome

Thresholding – (cont.)

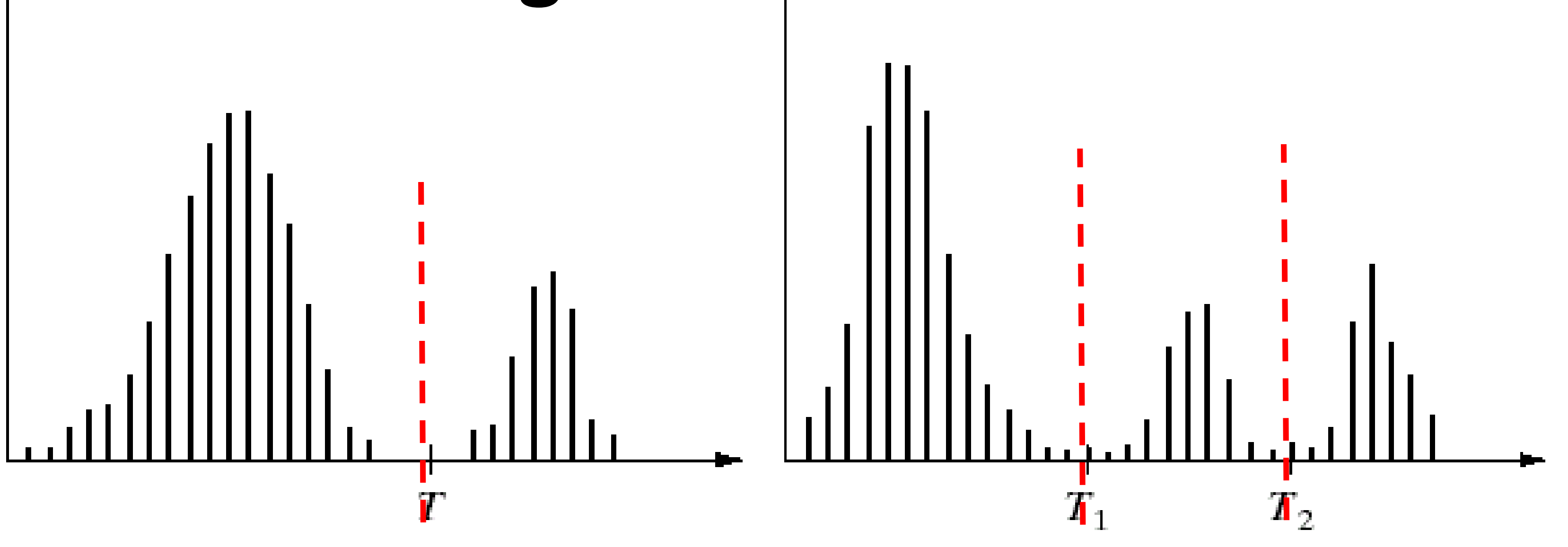
Drawbacks of Otsu's Method

- **Assumes the histogram is bimodal.**
- **Fails when the two classes are very unequal (i.e., have very different sizes).**
  - **In this case sigmaB may have two maxima.**
  - **The correct maximum is not necessary the global one.**
  - **The selected threshold should correspond to a valley of the histogram.**
- **Does not work well with variable illumination.**

Thresholding – (cont.)

Global Thresholding Problems

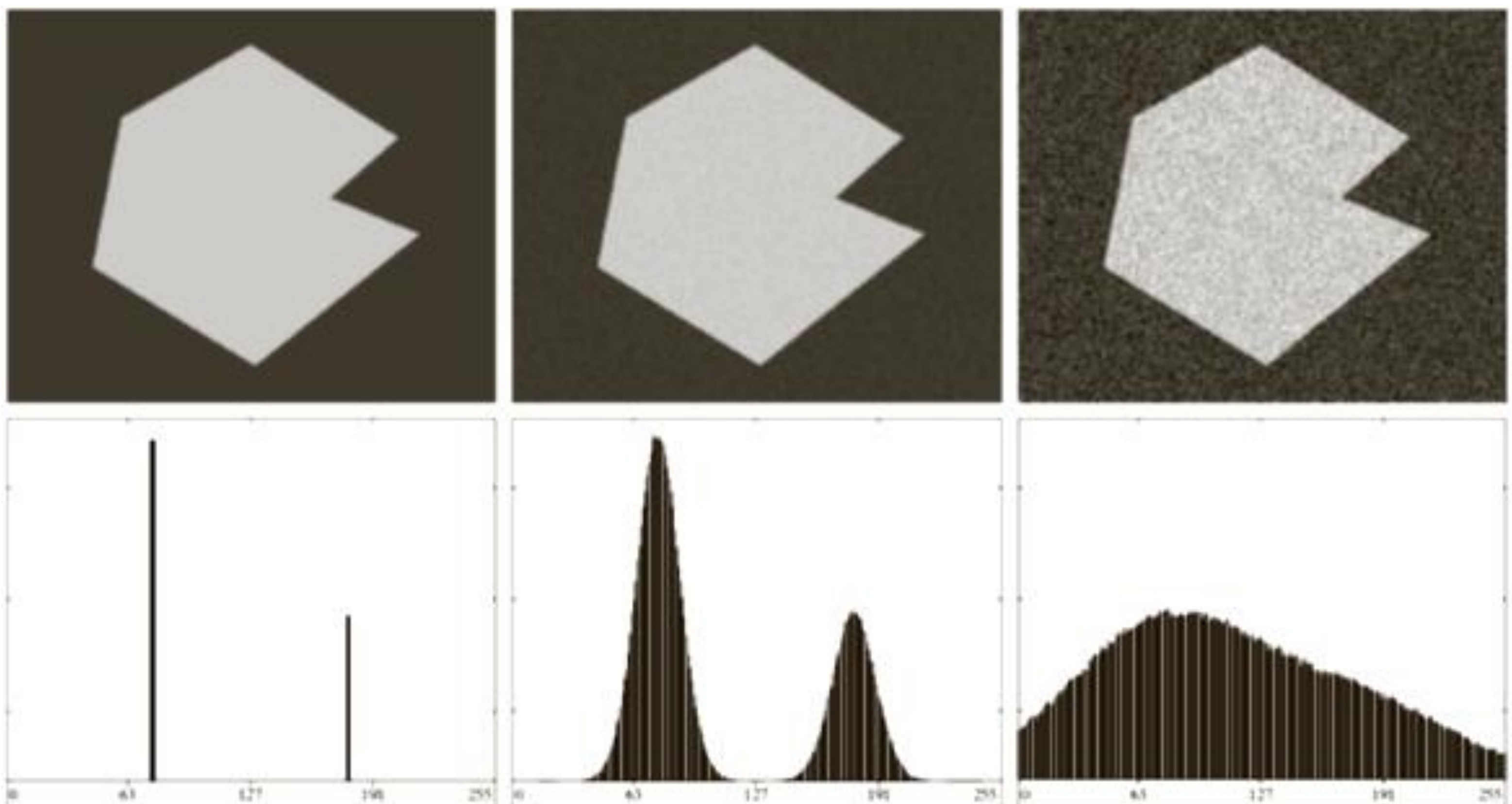
- **Single value thresholding only works for bimodal histograms.**
- **Images with other kinds of histograms need more than a single threshold.**



Thresholding – (cont.)

Global Thresholding Problems

- **Noise**



Gaussian noise with zero mean and different standard deviations

Thresholding – (cont.)

Global Thresholding Problems

- **Noise**
- **Solutions:**
- **Smoothing**

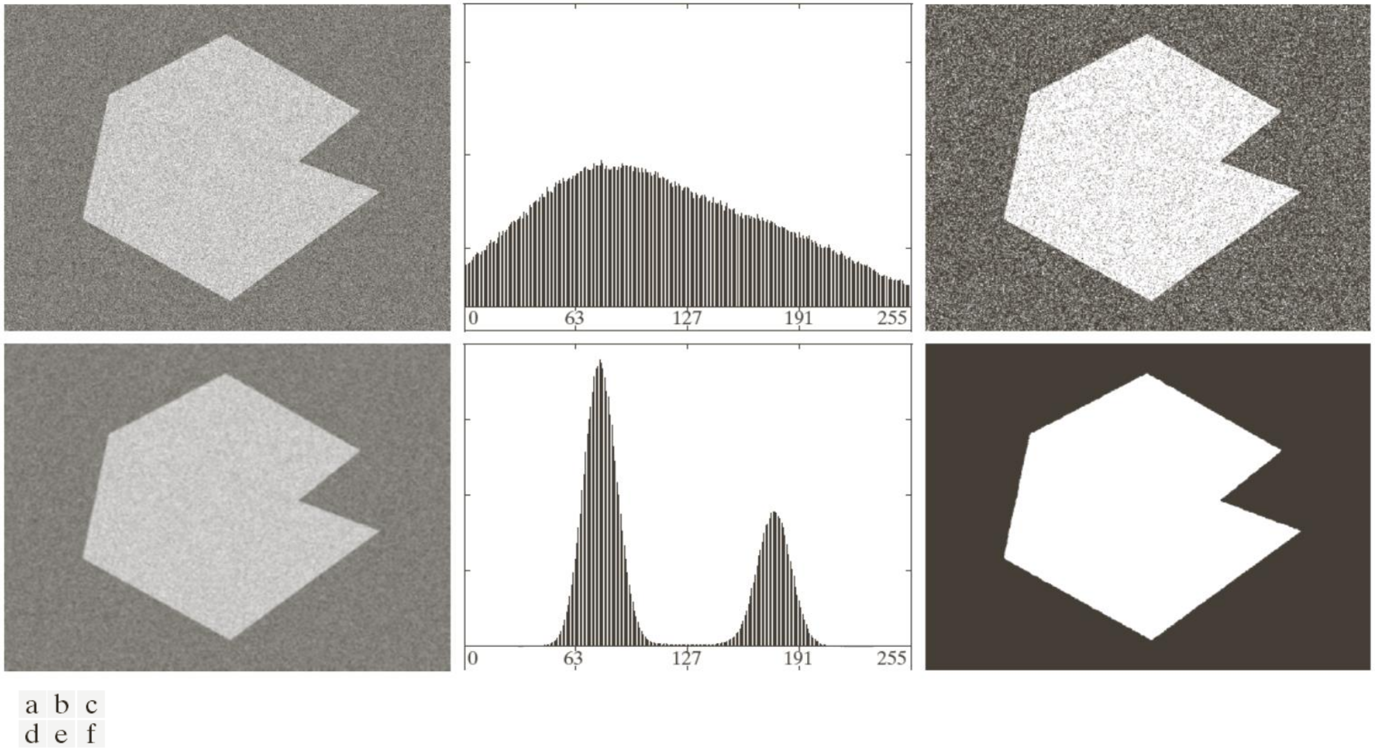
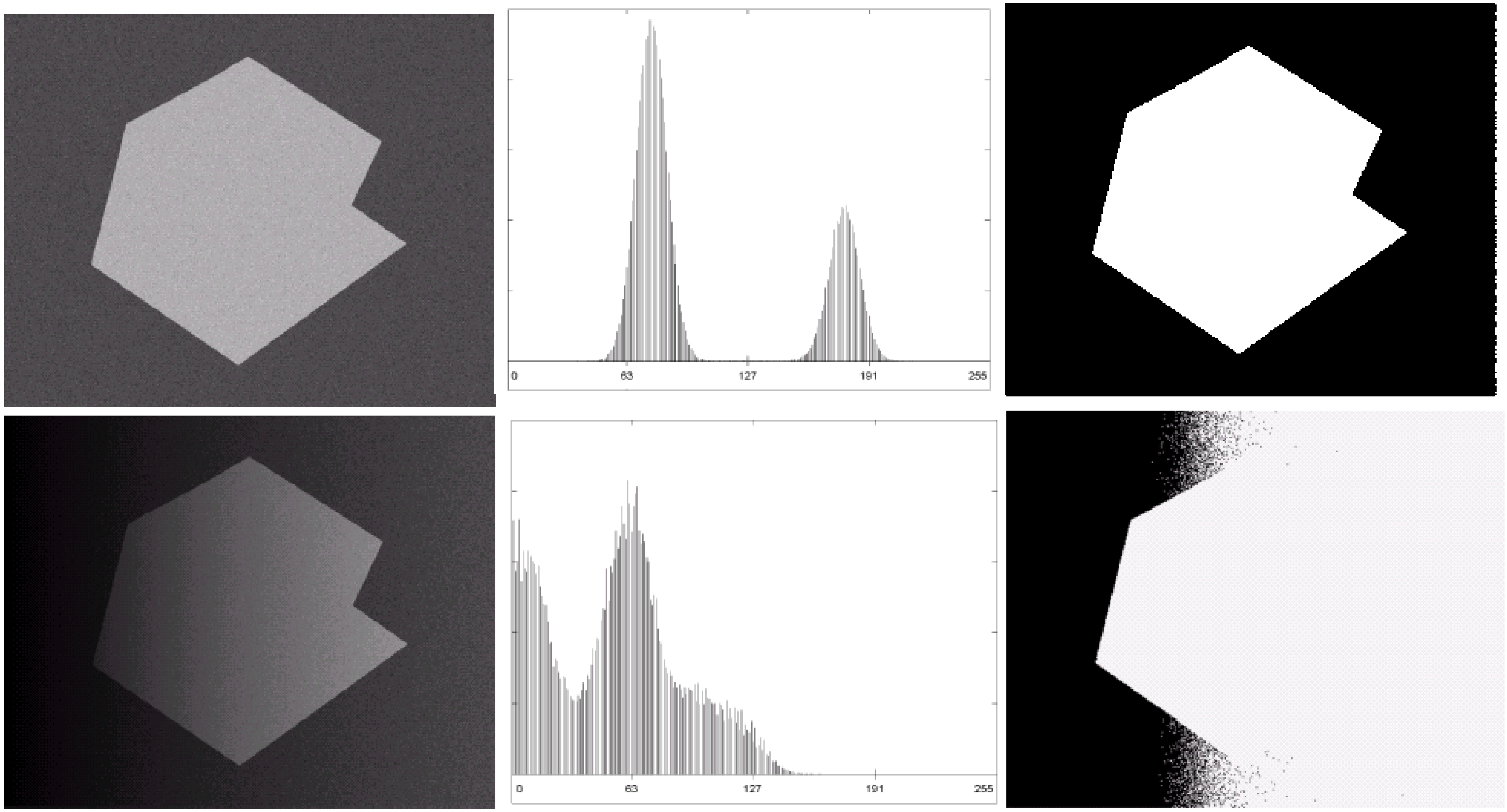


FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Thresholding – (cont.)

Global Thresholding Problems

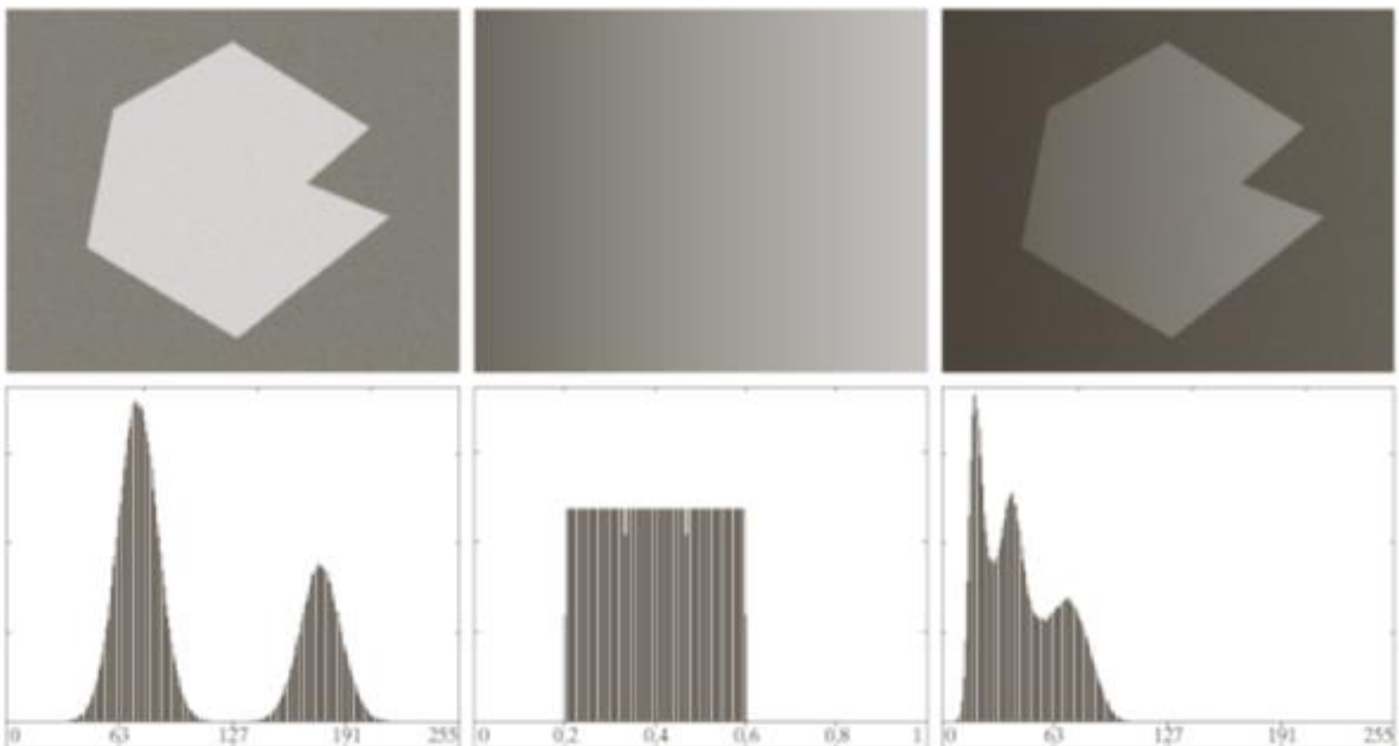
- **Non-uniform illumination or reflectance**



Thresholding – (cont.)

Global Thresholding Problems

- **Non-uniform illumination or reflectance**

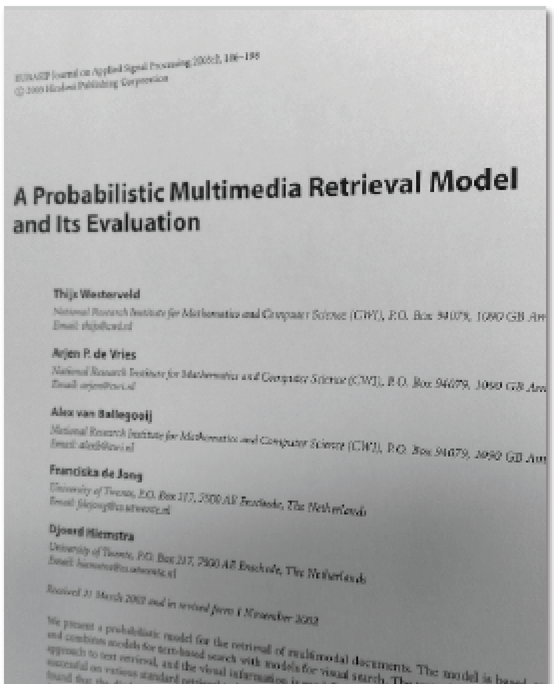


Non-uniform illumination can result from multiplying by ramp pattern.

Thresholding – (cont.)

Global Thresholding Problems

- **Non-uniform illumination or reflectance**



A Probabilistic Multimedia Retrieval Model and Its Evaluation

Thijs Westerveld

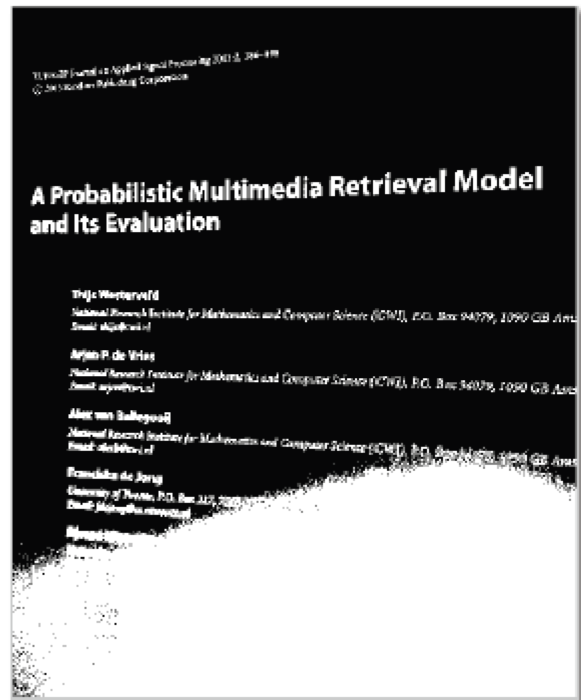
Arjen P. de Vries

Alex van Ballegooij

Franciska de Jong

Djoerd Hiemstra

Original image



A Probabilistic Multimedia Retrieval Model and Its Evaluation

Thijs Westerveld

Arjen P. de Vries

Alex van Ballegooij

Franciska de Jong

Thresholded with Otsu's Method

Thresholding – (cont.)

Global Thresholding Problems

- **Non-uniform illumination or reflectance**
- **Solutions:**

- **Correct the non-uniform shading pattern by its inverse (requires device).**
    - **Correcting non-uniform pattern: background subtraction, top-hat filtering, etc.**
- **Or: Variable thresholding.**

Thresholding – (cont.)

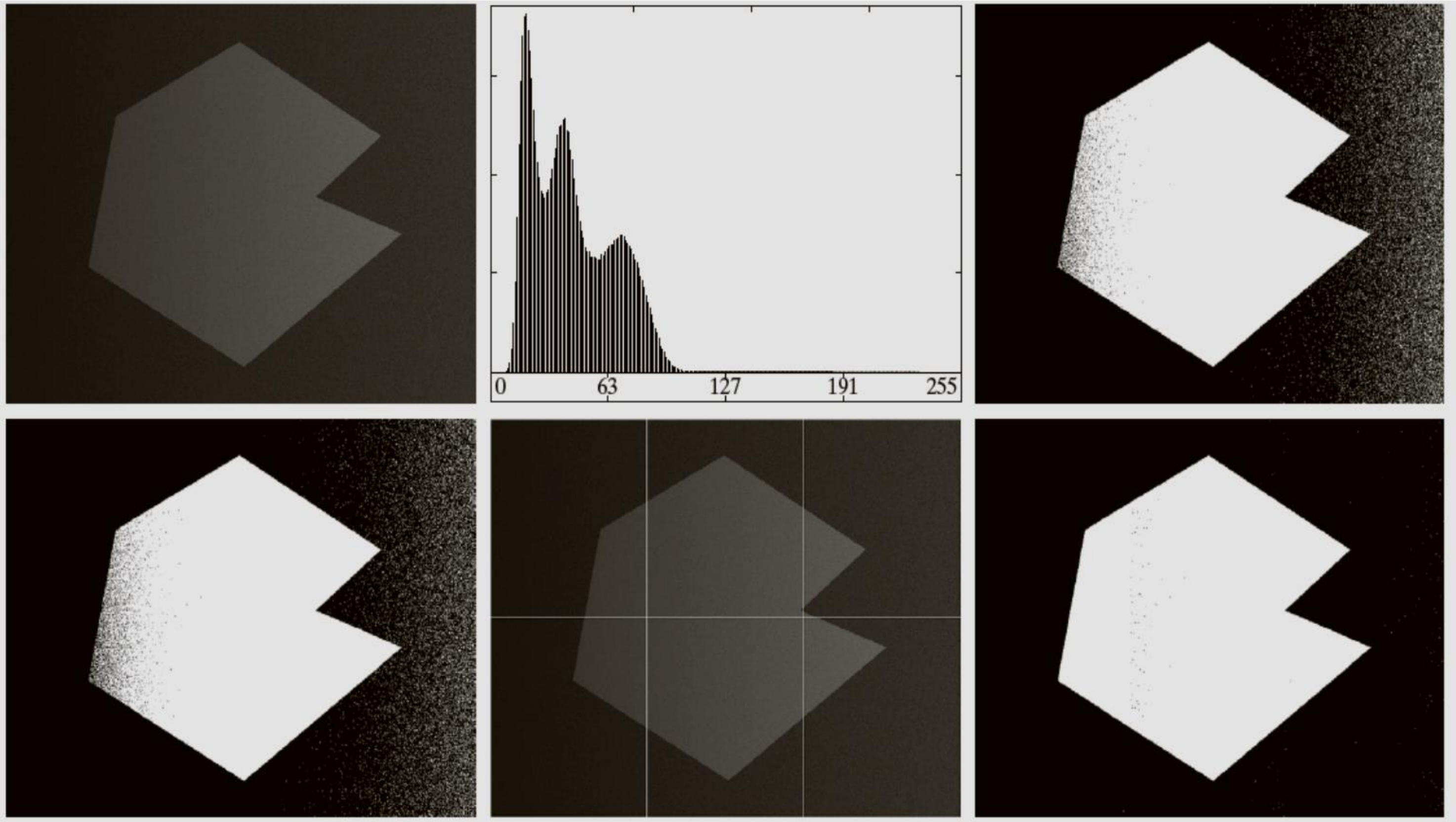
Variable Thresholding

- Image Partitioning
  - Subdivide image into non-overlapping rectangles.
  - Small enough so that illumination of each is approximately uniform.
  - Estimate a different threshold to segment each subimage.

Thresholding – (cont.)

Variable Thresholding

- **Image Partitioning**



Iterative global

Otsu's method

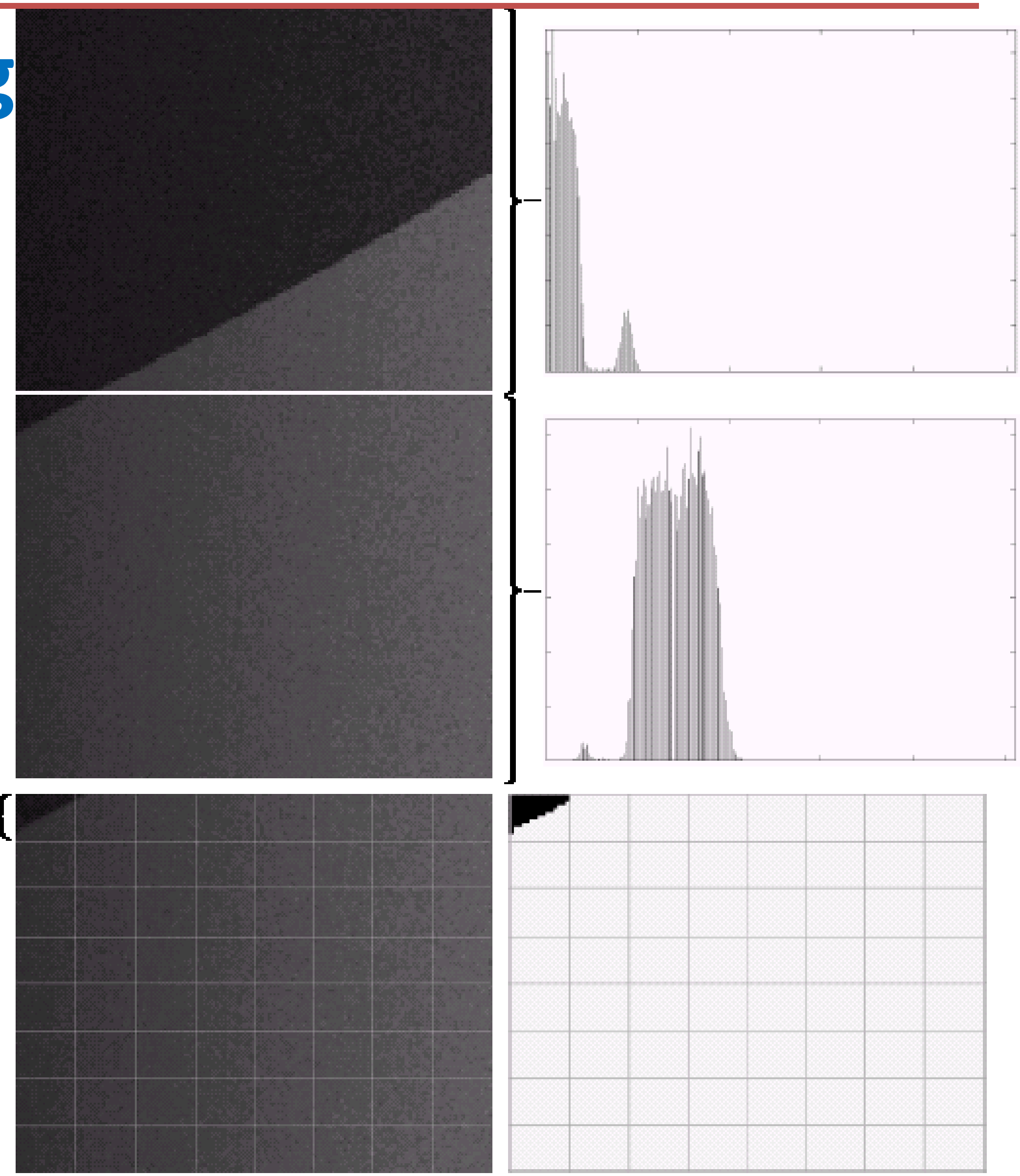
Otsu's method on each partition

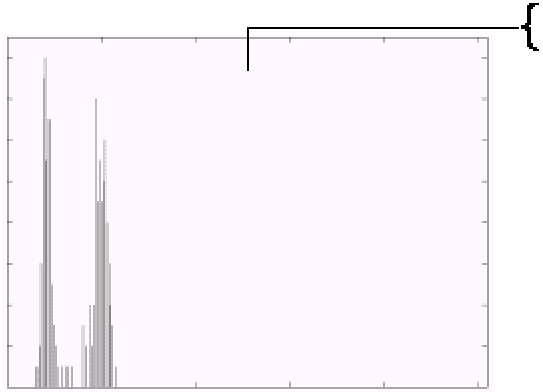
Thresholding – (cont.)

Variable Thresholding

- **Image Partitioning**

- Further subdivisions may be required for better results.

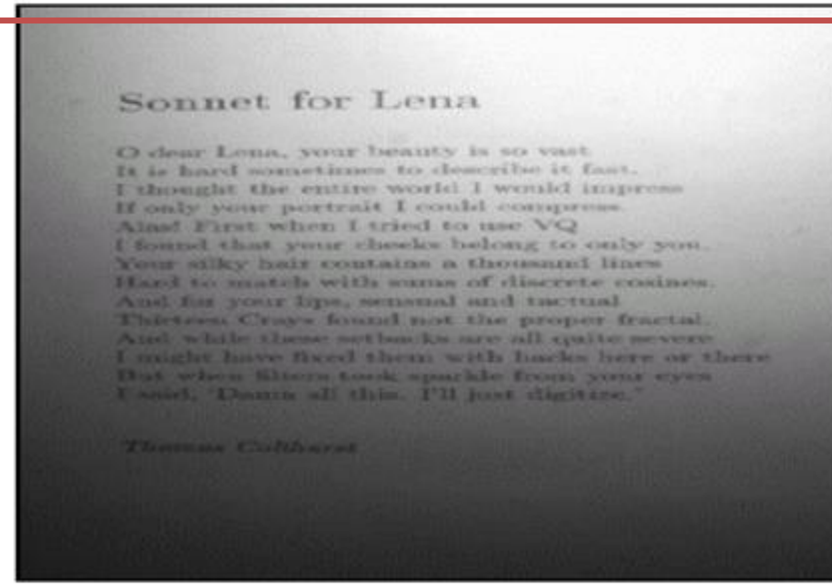




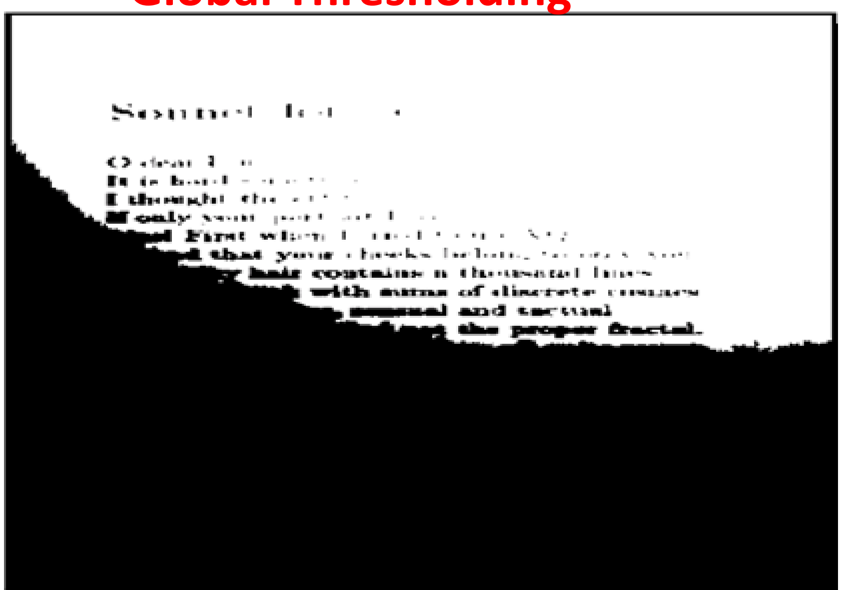
Thresholding – (cont.)

Variable Thresholding

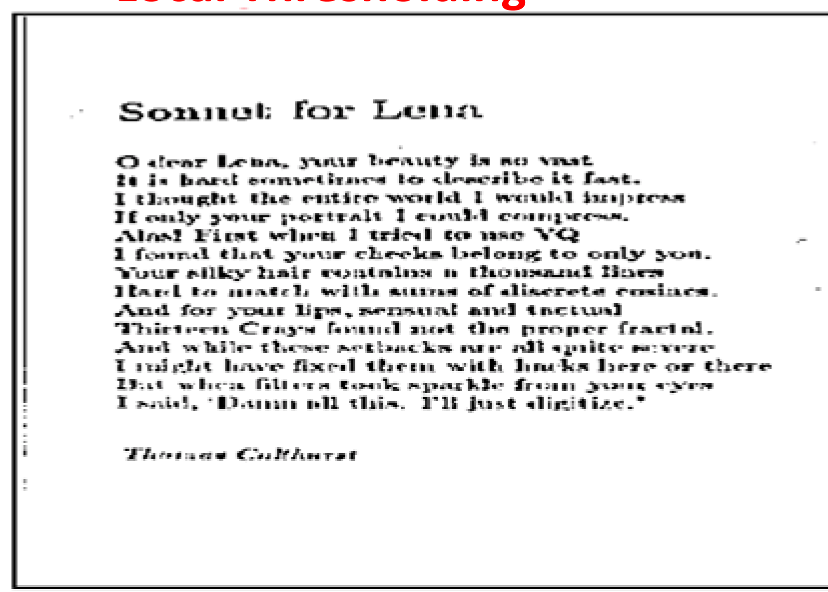
- Image Partitioning



Global Thresholding



Local Thresholding



Thresholding – (cont.)

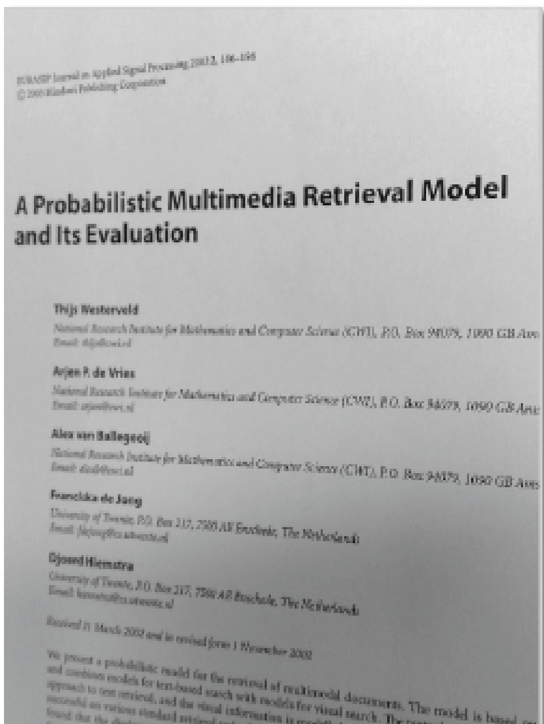
Variable Thresholding

- **Local Image Properties**
  - Threshold according to the mean and standard deviation of a neighborhood.
  - $T_{xy} = a\sigma_{xy} + bm_{xy}$
  - $T_{xy} = a\sigma_{xy} + bm_G$
  - $g(x,y) = \begin{cases} 1 \; if \; f(x,y) > T_{xy} \\ 0 \; if \; f(x,y) \leq T_{xy} \end{cases}$

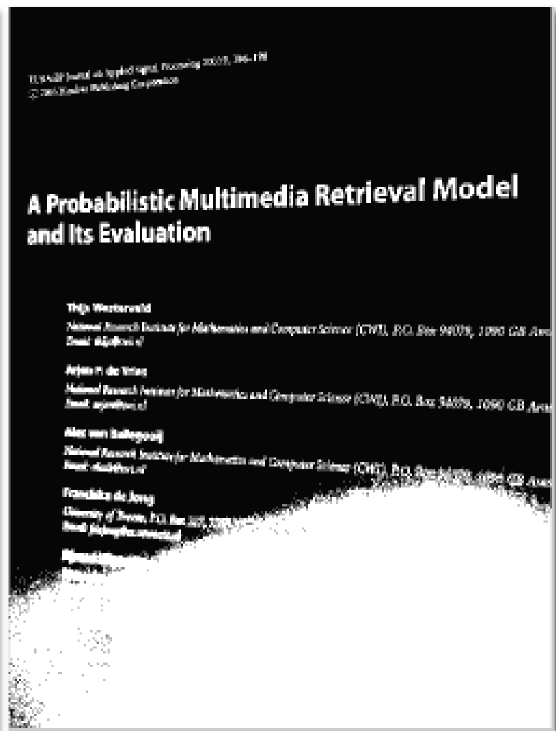
Thresholding – (cont.)

Variable Thresholding

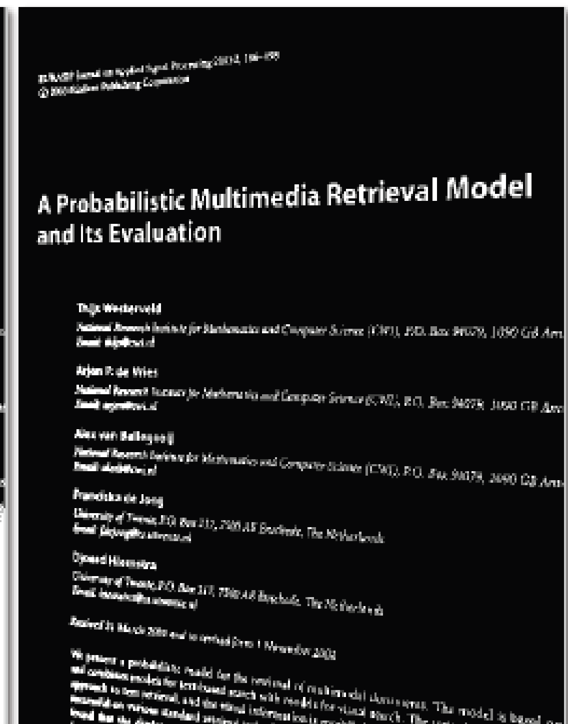
- **Local Image Properties**



Original image



Global threshold



Local threshold

Thresholding – (cont.)

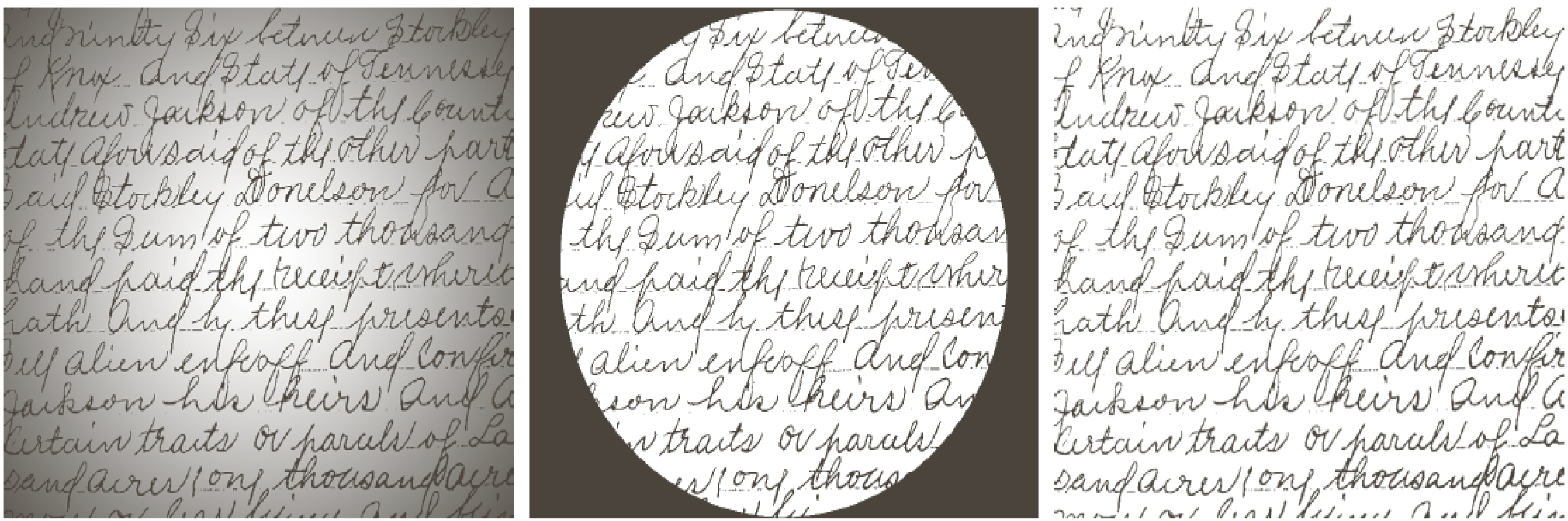
Variable Thresholding

- **Moving Average**
  - **Scanning is carried out line by line in zigzag pattern to reduce illumination bias.**
  - $m(k+1) = \frac{1}{n}\sum_{i=k+2-n}^{k+1} z_i$
  - $T_{xy} = bm_{xy}$
  - $g(x,y) = \begin{cases} 1 \; if \; f(x,y) > T_{xy} \\ 0 \; if \; f(x,y) \leq T_{xy} \end{cases}$
  - **Works well when the objects of interest are small (or thin) with respect to the image size, a condition satisfied in images of handwritten text.**

Thresholding – (cont.)

Variable Thresholding

- **Moving Average**



a b c

FIGURE 10.49 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

Next Lecture

Region-based Segmentation + Morphology

Assignment

Check book sections and associated problems

Chapter 10	1, 2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.7(Hough only)
Associated problems	2, 3, 4, 5, 6, 7, 12, 23, 26, 27, 28, 29, 36, 39

References

- Gonzalez and Woods, Digital Image Processing.

- Richard Szeliski, Computer Vision Algorithms and Applications, Springer 2010. (Chapter 5) http://szeliski.org/Book/

- Peters, Richard Alan, II, "Image Segmentation (PointLineEdge)", Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008, Available on the web at the Internet Archive, http://www.archive.org/details/Lectures_on_Image_Processing.

-http://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html