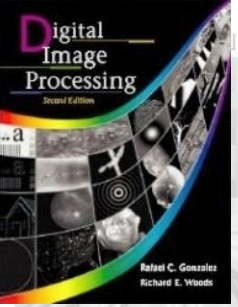


Digital Image Processing

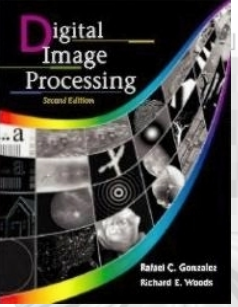
Chapter 10

Image Segmentation

Dr. Kai Shuang

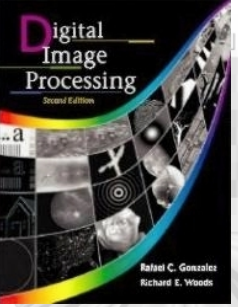
Department of Electronic Engineering
China University of Petroleum
shuangkai@cup.edu.cn

Some slides and illustrations from Dr. Jimin Liang and Dr. Nawapak Eua-Anant



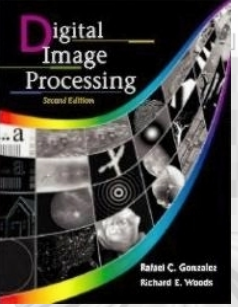
Outline

- Detection of gray level discontinuities
  - Point detection
  - Line detection
  - Edge detection
    - Gradient operators
    - LoG : Laplacian of Gaussian
- Edge linking and boundary detection
  - Hough transform
- Thresholding
- Region-based segmentation
- Segmentation by Morphological watersheds
- The use of motion in segmentation

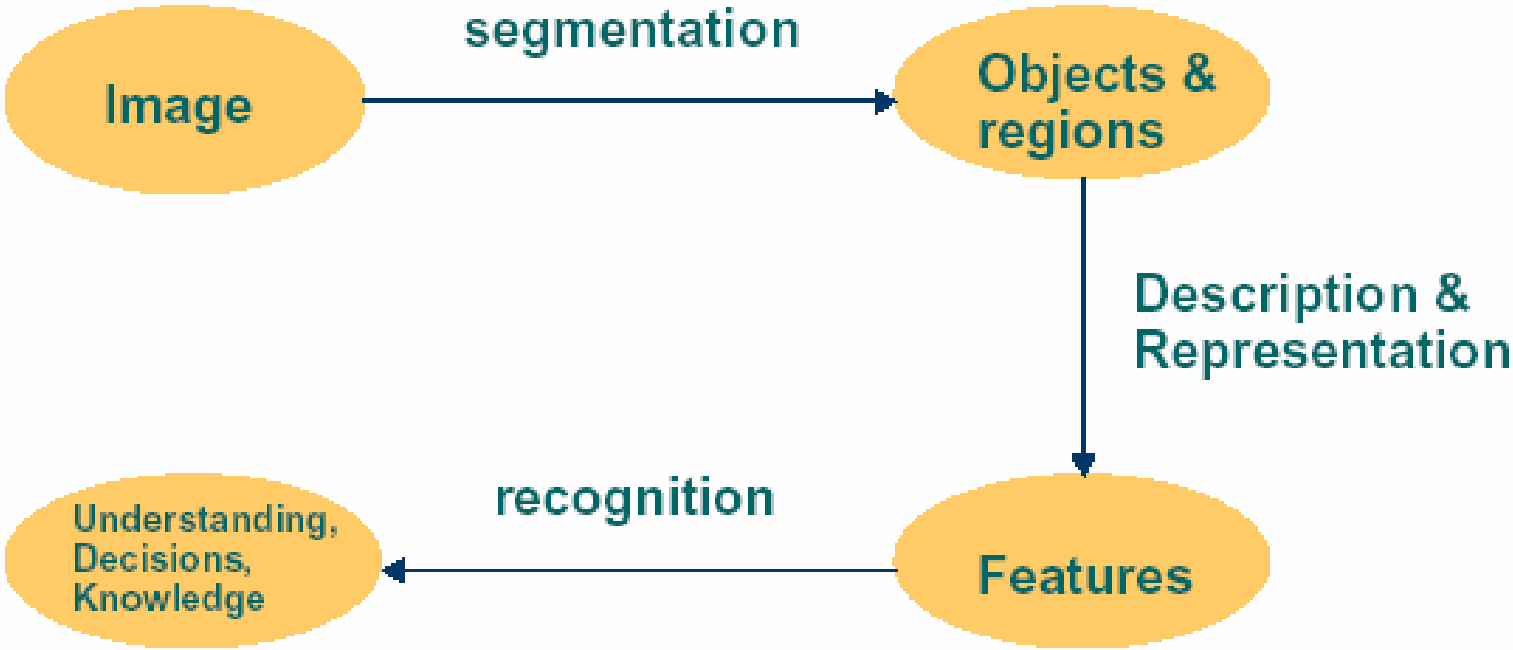


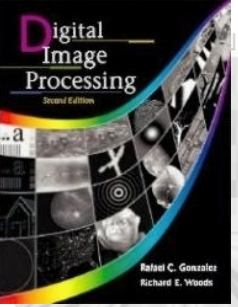
Goal of Image analysis

- Extracting information from an image
  - Step 1 : segment the image ---> objects or regions
  - Step 2 : describe and represent the segmented regions in a form suitable for computer processing
  - Step 3 : image recognition and interpretation



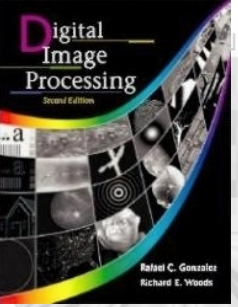
Procedure of Image analysis



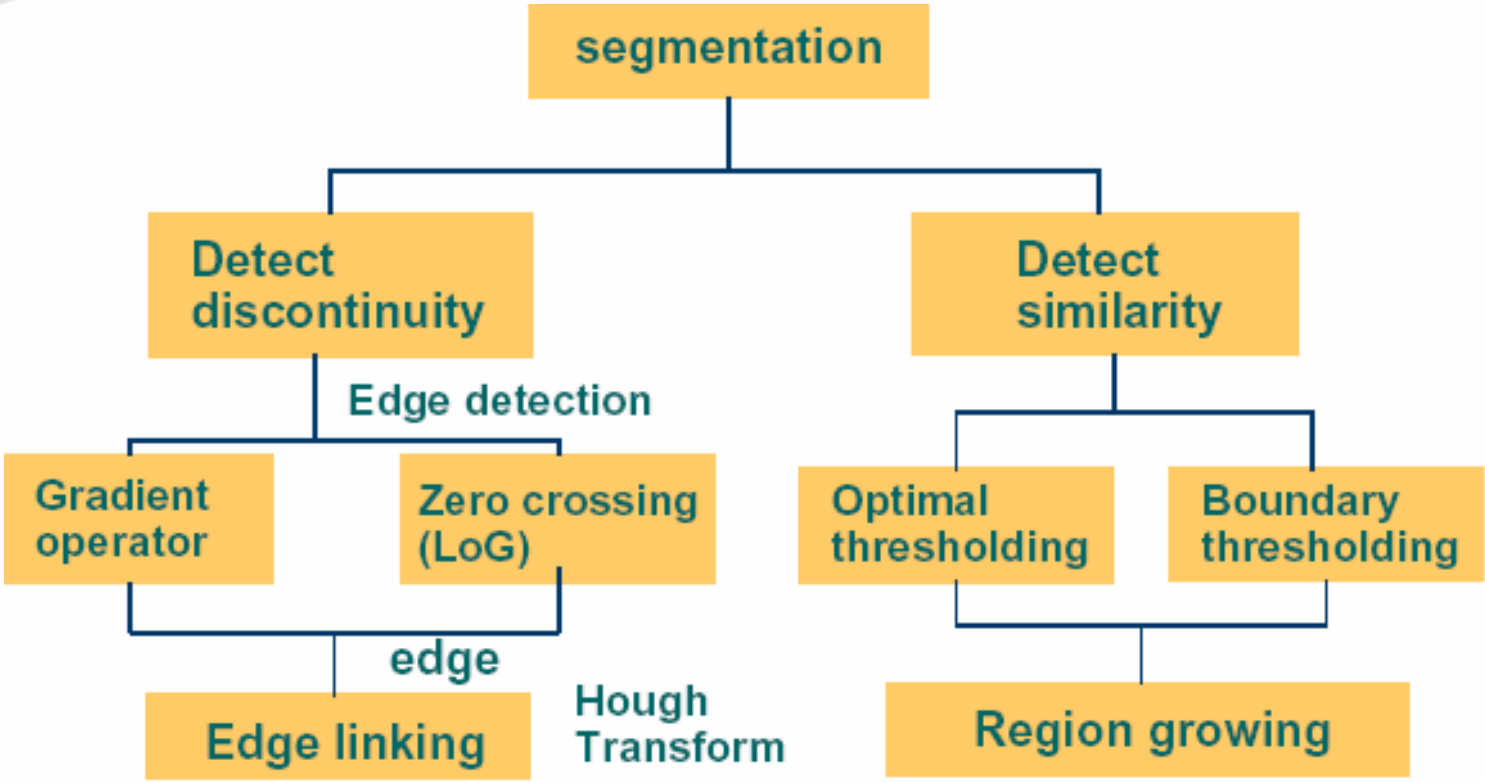


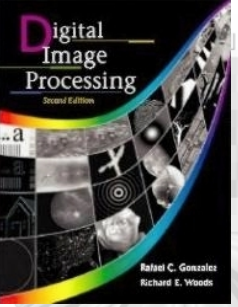
What is segmentation?

- Subdivides an image into its constituent regions or objects
- Separate an image into regions which are called as objects and background
- Represent the result images with binary images, label the objects as “1” and the background as “0” commonly.
- Heavily rely on one of two properties of intensity values:
  - Discontinuity ---- Partition based on abrupt changes in intensity, e.g. edges in an image
    - point / line / edge / corner detection
  - Similarity ---- Partition based on intensity similarity, e.g. thresholding
    - thresholding
    - region growing / splitting / merging



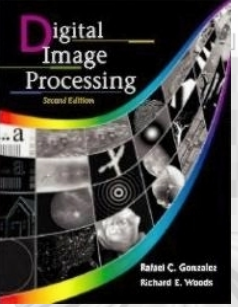
Segmentation





What Should Good Image segmentation be?

- Region interiors
  - Simple
  - Without many small holes
- Adjacent regions
  - Should have significantly different values
- Boundaries
  - Simple
  - Not ragged
  - Spatially accurate



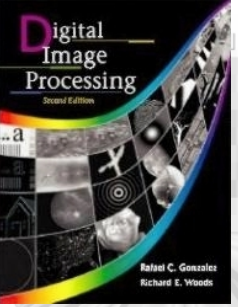
Way to detect discontinuity

Correlation

$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

Grayscale image

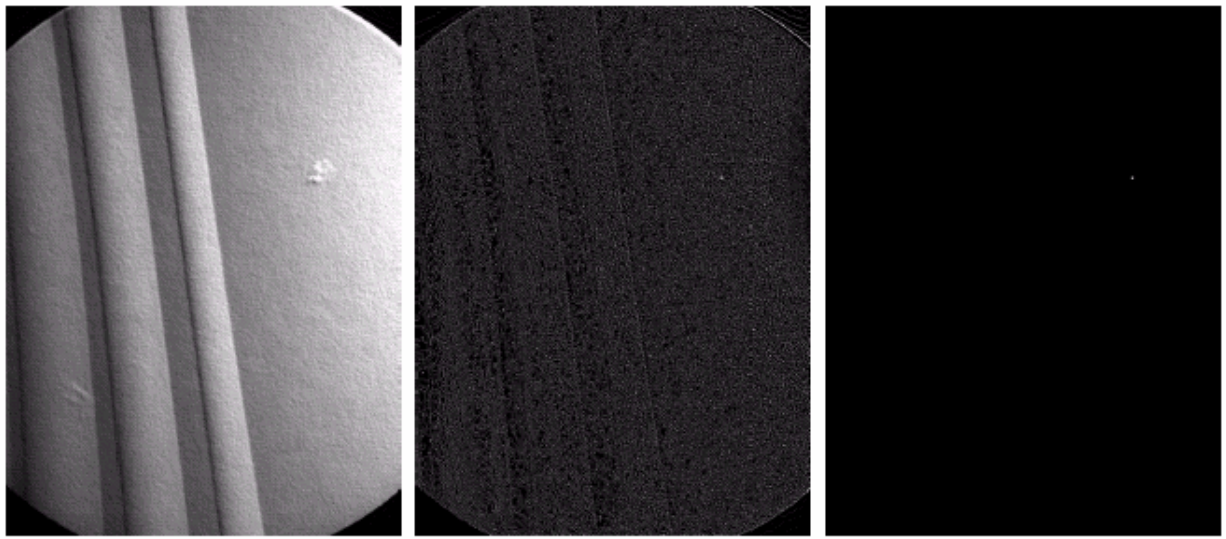
Mask coefficient



Point detection

$$|R| \geq T$$

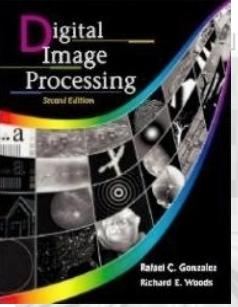
−1	−1	−1
−1	8	−1
−1	−1	−1



a
b c d

FIGURE 10.2
(a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



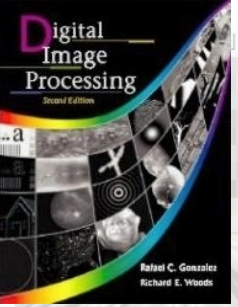
Line detection

- Masks for lines of different directions:

FIGURE 10.3 Line masks.

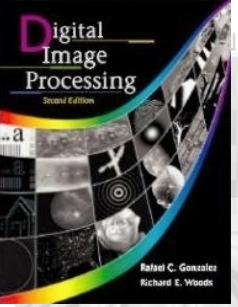
−1	−1	−1	−1	−1	2	−1	2	−1	2	−1	−1
2	2	2	−1	2	−1	−1	2	−1	−1	2	−1
−1	−1	−1	2	−1	−1	−1	2	−1	−1	−1	2
Horizontal			+45°			Vertical			−45°		

- Respond more strongly to lines of one pixel thick of the designated direction.
- High or low pass filters?

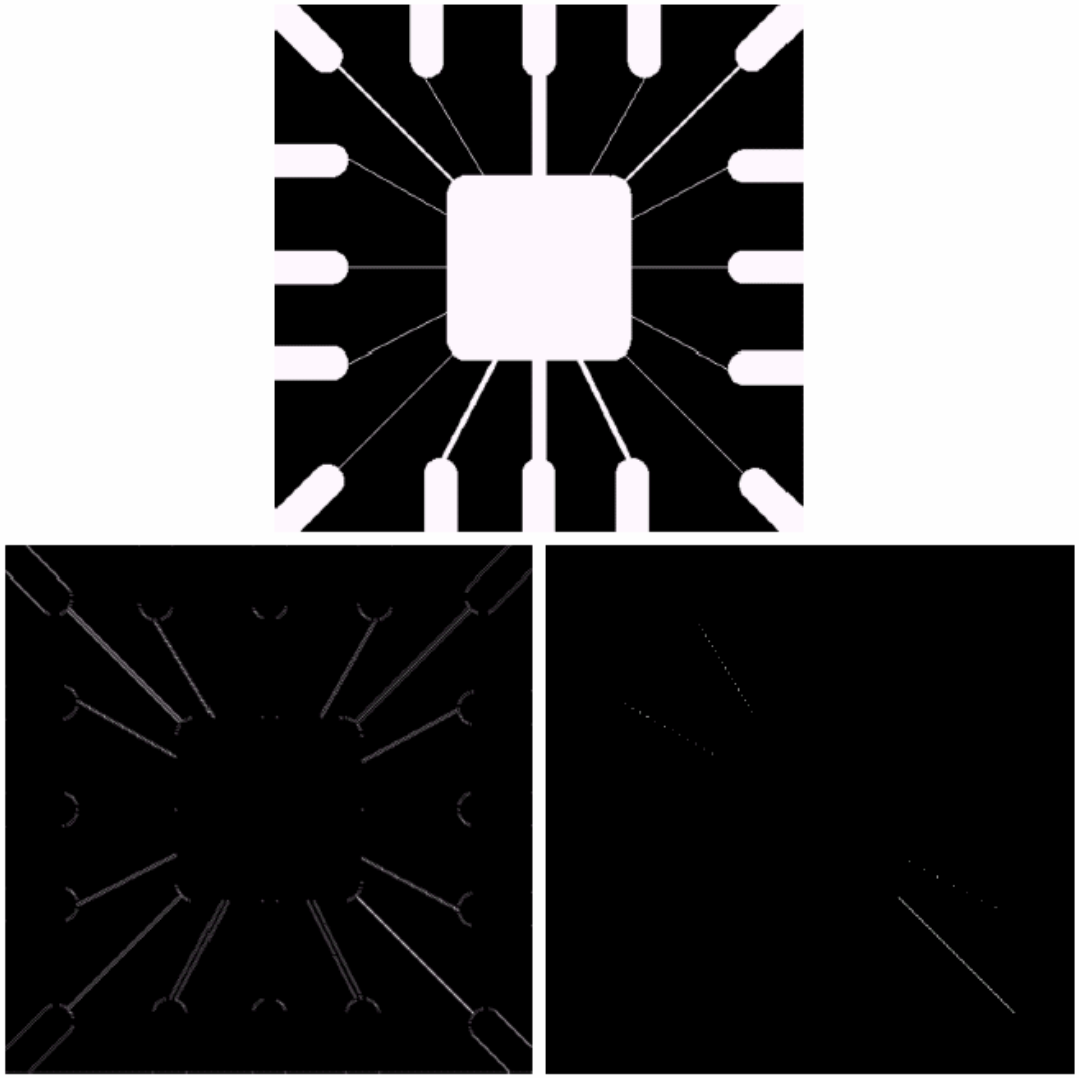


Line detection (cont'd)

- If interested in lines of any directions, run all 4 masks and select the highest response.
- If interested only in lines of a specific direction (e.g. vertical), use only the mask associated with that direction.
- Threshold the output.
- The strongest responses for lines one pixel thick, and correspond closest to the direction defined by the mask.



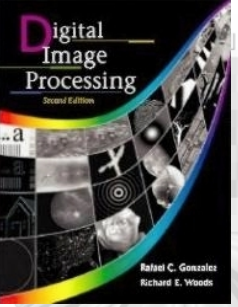
Line detection (cont'd)



a
b c

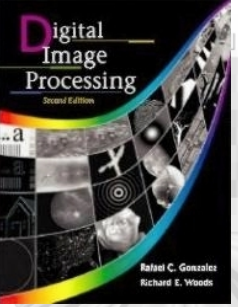
FIGURE 10.4
Illustration of line detection.
(a) Binary wire-bond mask.
(b) Absolute value of result after processing with −45° line detector.
(c) Result of thresholding image (b).

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



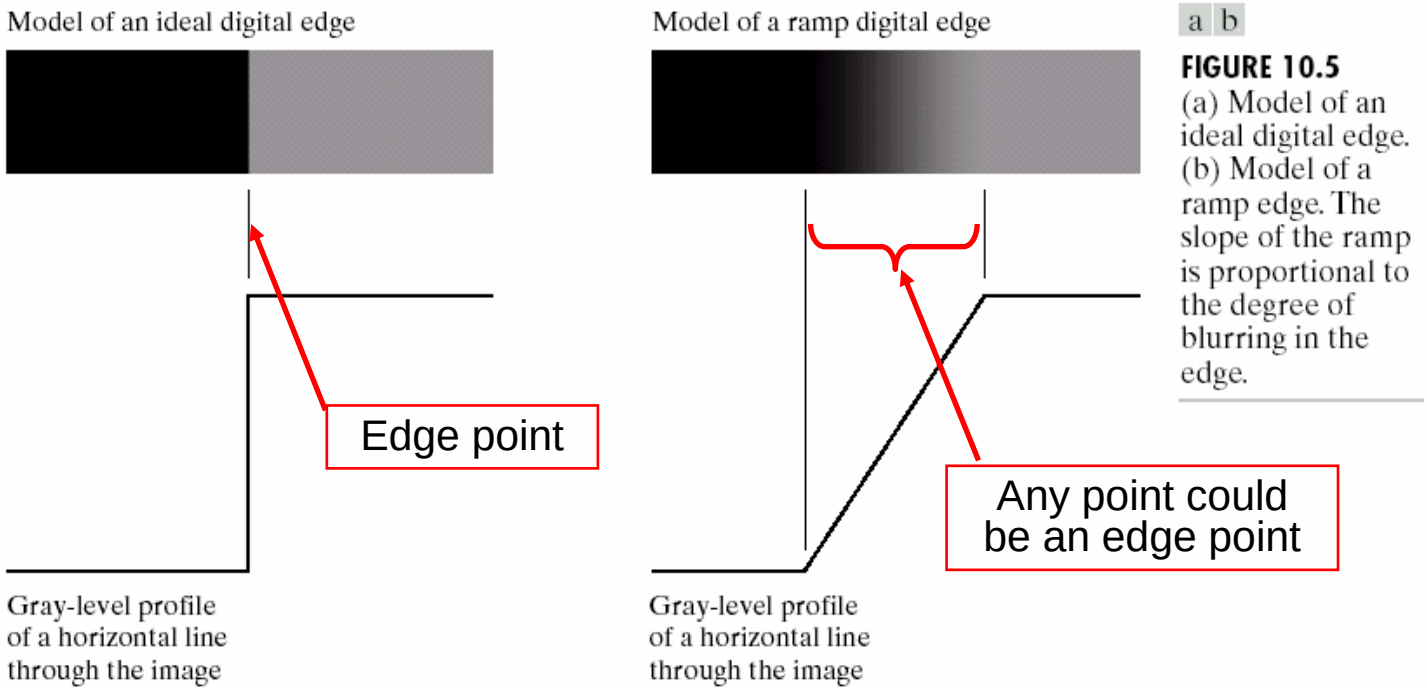
Edge Detection

- Far more practical than line detection.
- Approaches discussed based on
  - 1st-order digital derivative
  - 2nd-order digital derivative



What is an edge?

- A set of connected pixels that lie on the boundary between two regions.

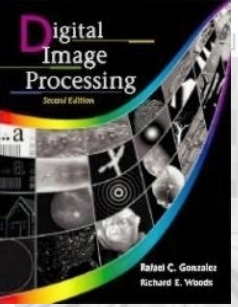


a b

FIGURE 10.5
(a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

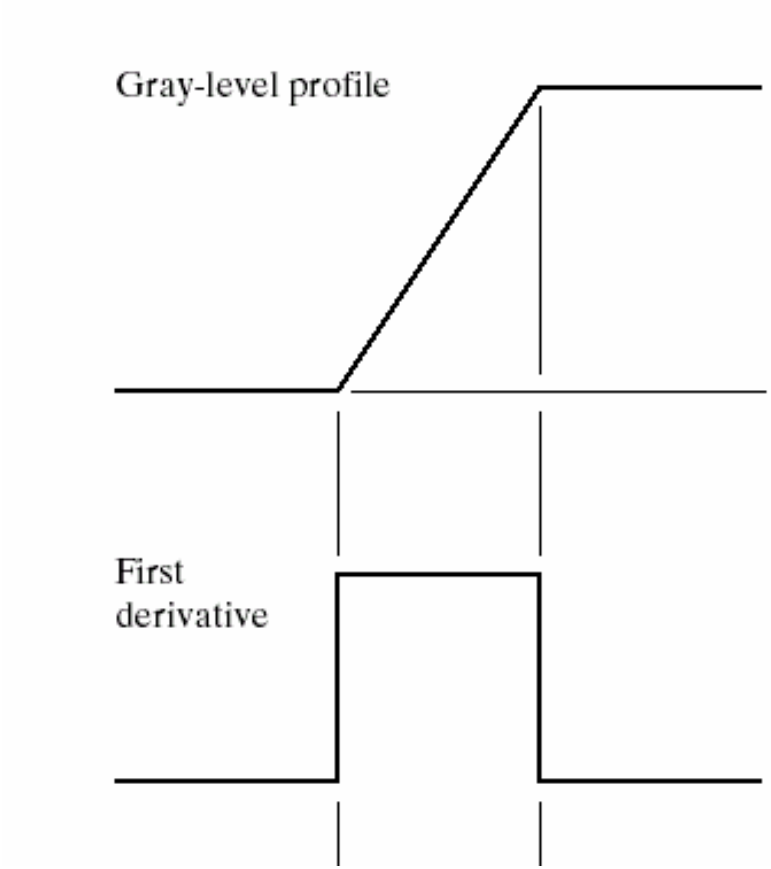
Ideal/step edge

Ramp-like (in real life) edge

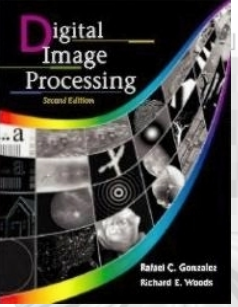


1st Derivative

- Positive at the points of transition into and out of the ramp, moving from left to right along the profile
- Constant for points in the ramp
- Zero in areas of constant gray Level
- Magnitude for presence of an edge at a point in an image (i.e. if a point is on a ramp)

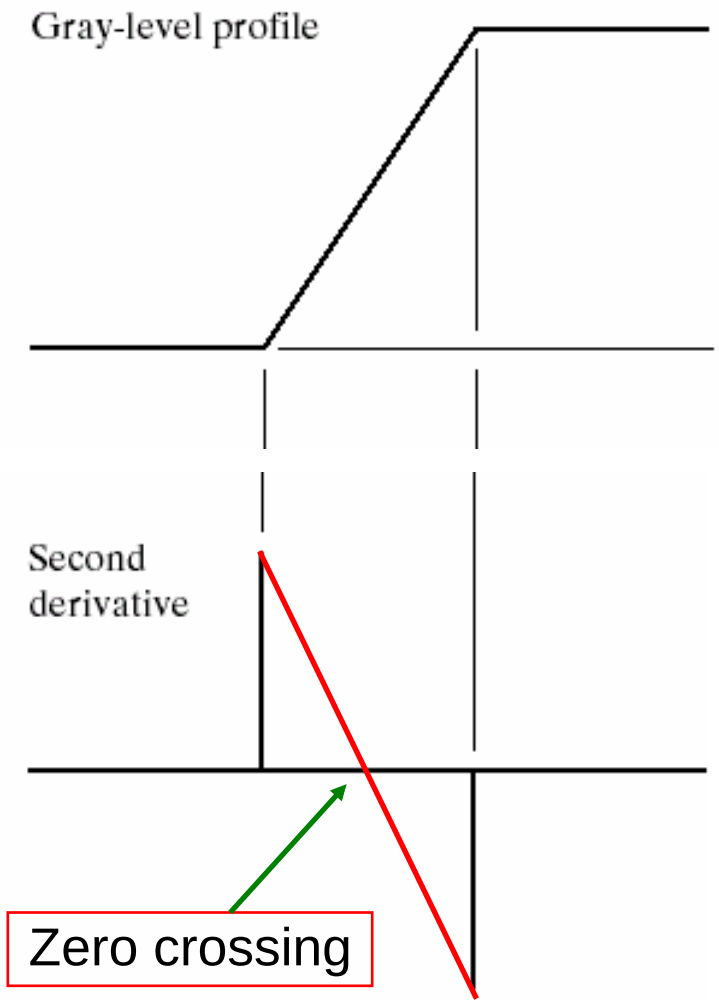


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

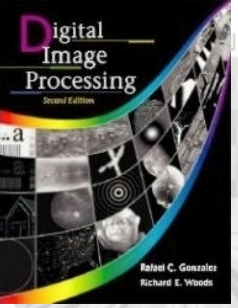


2nd Derivative

- Positive at the transition associated with the dark side of the edge
- Negative at the transition associated with the bright side of the edge
- Zero elsewhere
- Producing 2 values for every edge in an image (an undesirable feature).
- Center of a thick edge is located at the zero crossing

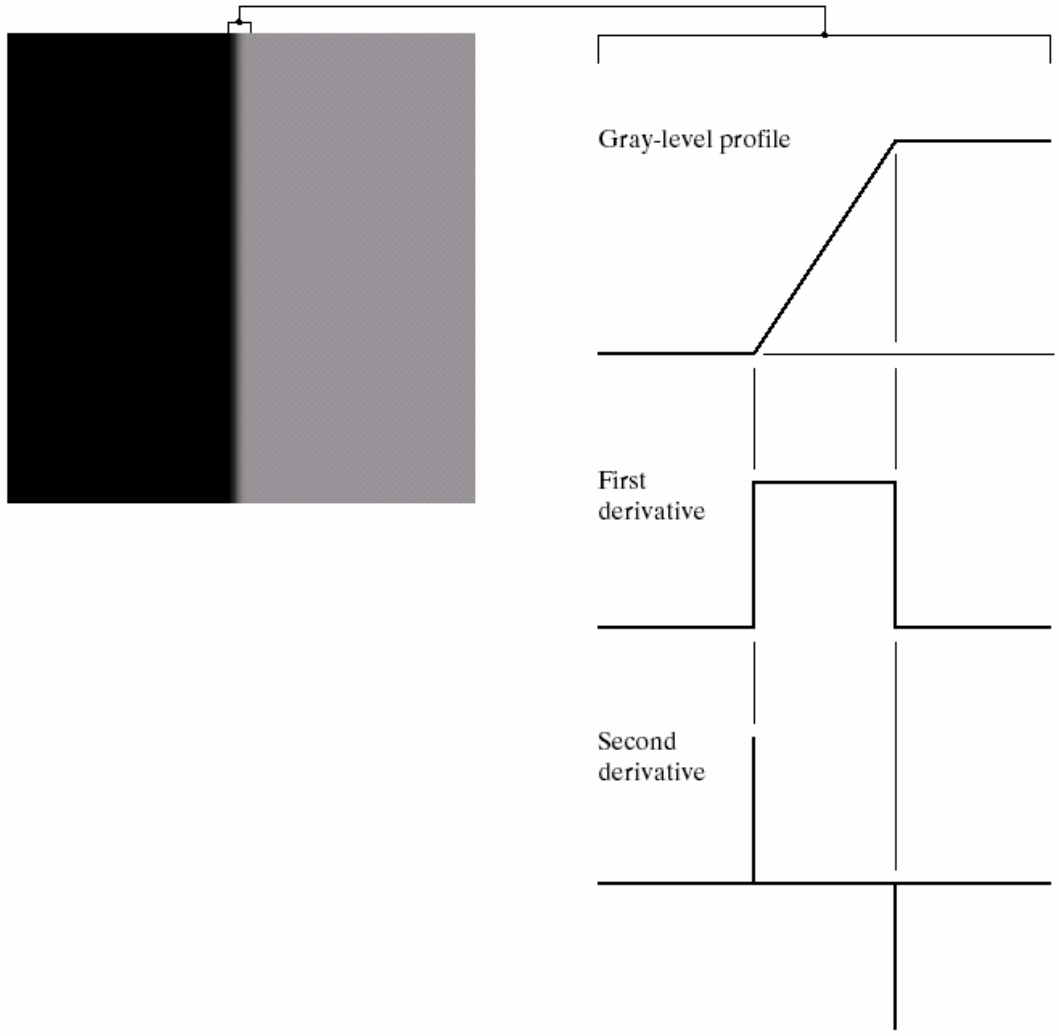


Edge detection (cont'd)

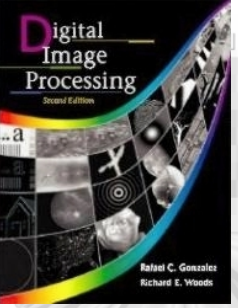


a b

FIGURE 10.6
(a) Two regions separated by a vertical edge. (b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.

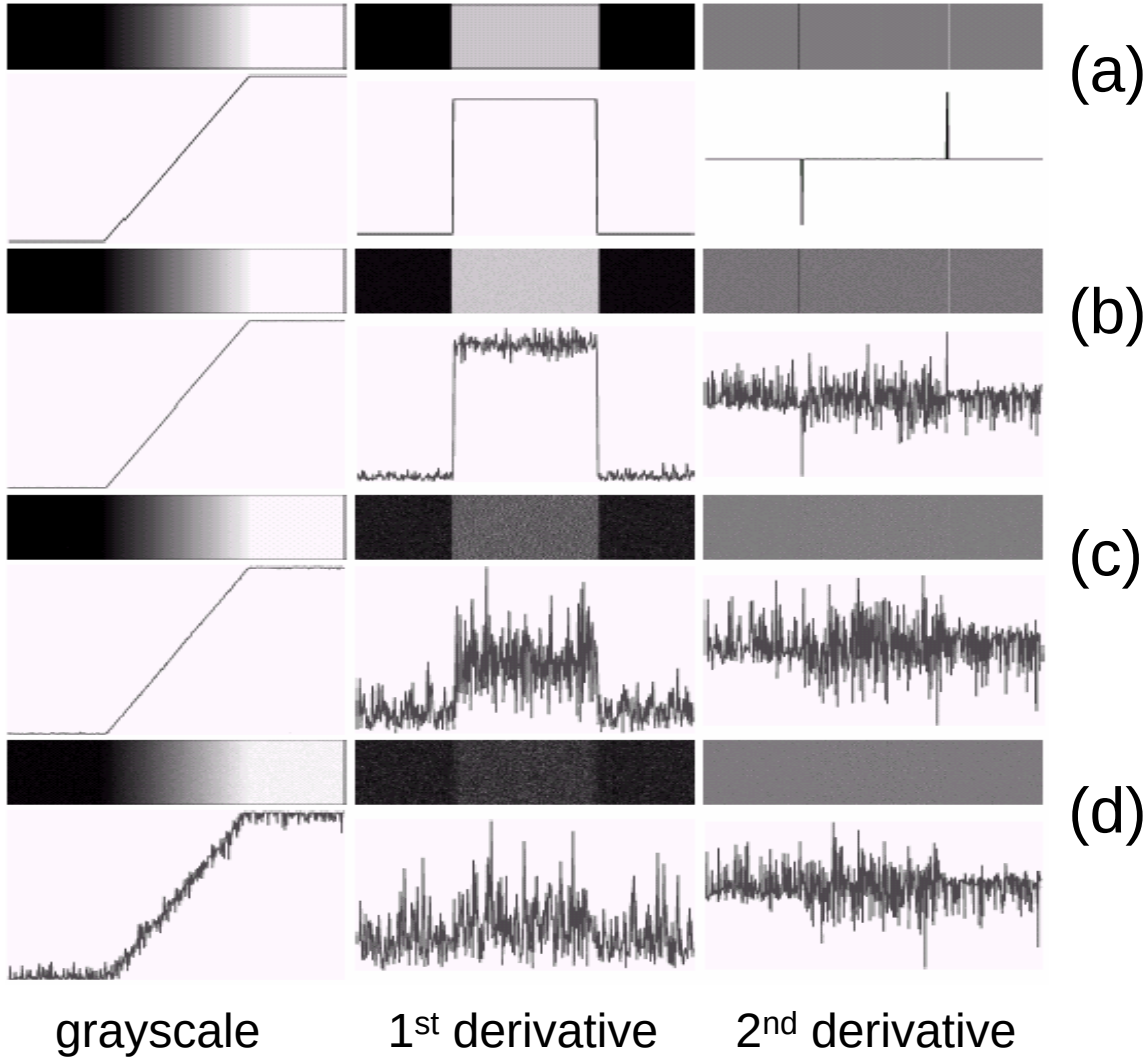


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

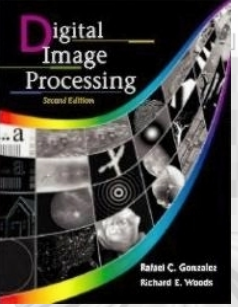


Edge detection (cont') ---- Effect of Noise

- Corrupted by Random Gaussian noise of mean 0 and standard deviation of
  (a) 0
  (b) 0.1
  (c) 1.0
  (d) 10.0
- Conclusion? Sensitivity of derivative to noise

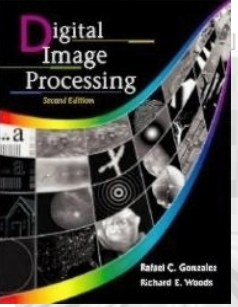


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



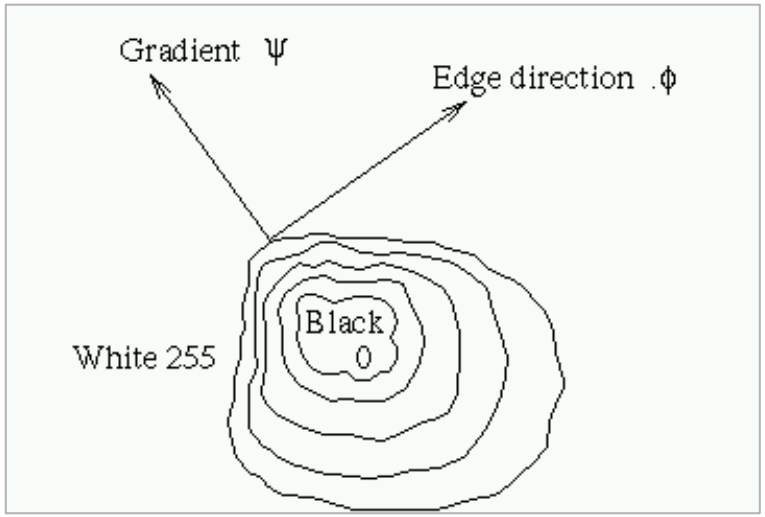
Edge detection (cont'd)

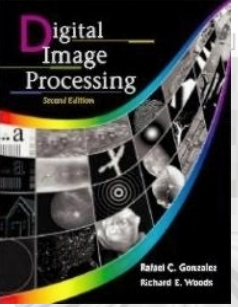
- The difference between edge and boundary
  - Edge: a "local" concept
  - Boundary: a more global idea
- Edge detection steps
  - Compute the local derivative
  - Magnitude of the 1st derivative can be used to detect the presence of an edge
  - The sign of the 2nd derivative can be used to determine whether an edge pixel lies on the dark or light side of an image
  - Zero crossing of the 2nd derivative is at the midpoint of a transition in gray level, which provides a powerful approach for locating the edge.



Some Terminology

- An edge element is associated with 2 components:
  - magnitude of the gradient, and
  - and edge direction ϕ , rotated with respect to the gradient direction ψ by -90° .



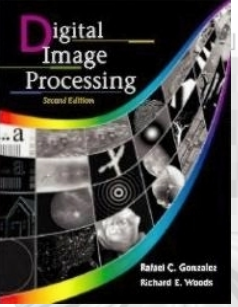


Gradient operators (1st Derivative)

- Use gradient for image differentiation
- The gradient of an image *f(x,y)* at location *(x,y)* is defined as

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T$$

- Some properties about this gradient vector
  - It points in the direction of maximum rate of change of image at *(x,y)*
  - Magnitude $mag(\nabla f) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$
  - angle $\psi(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$



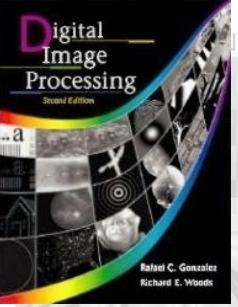
Finite Gradient - Approximation

- Central differences (not usually used because they neglect the impact of the pixel *(x,y)* itself)

$$\Delta_x f(x,y) = \frac{f(x+h,y) - f(x-h,y)}{2h}$$

$$\Delta_y f(x,y) = \frac{f(x,y+h) - f(x,y-h)}{2h}$$

- **h** is a small integer, usually 1.
- **h** should be chosen small enough to provide a good approximation to the derivative, but large enough to neglect unimportant changes in the image.



Edge operator

a
b c
d e
f g

FIGURE 10.8
A 3 × 3 region of an image (the z's are gray-level values) and various masks used to compute the gradient at point labeled z_5.

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

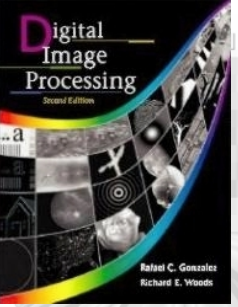
−1	−1	−1	−1	0	1
0	0	0	−1	0	1
1	1	1	−1	0	1

Prewitt

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

Sobel

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Sobel edge operator

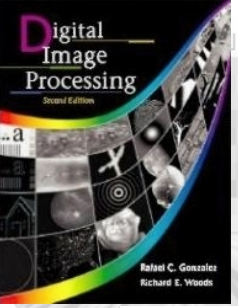
- Advantages : providing both differencing and a smooth effect and slightly superior noise reduction characteristics.

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

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

Better noise-suppression

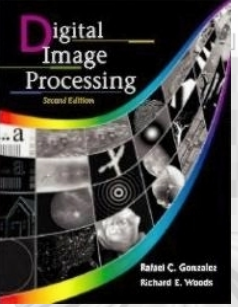
Edge detection example



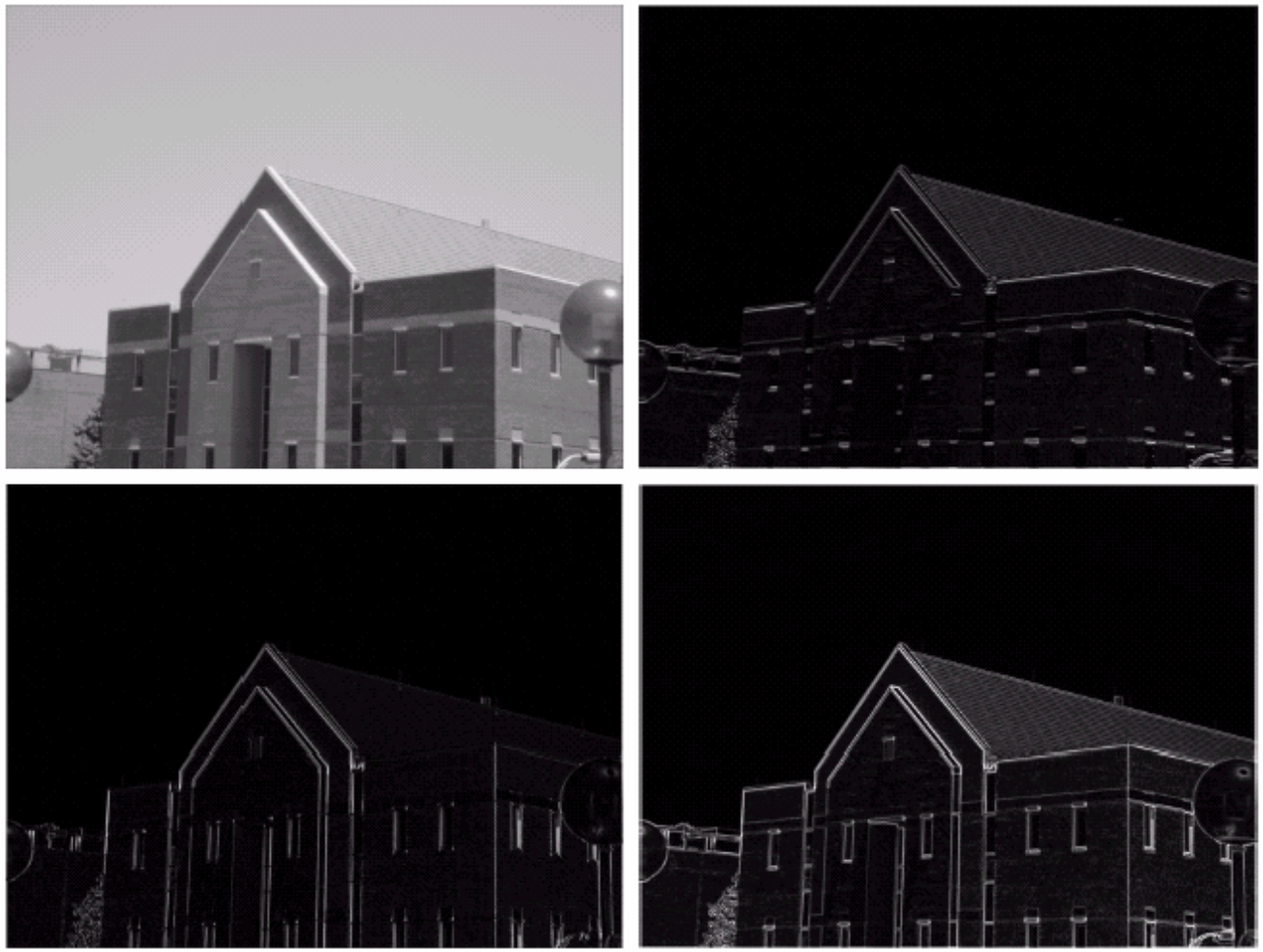
a b
c d

FIGURE 10.10
(a) Original image. (b) $|G_x|$, component of the gradient in the x-direction. (c) $|G_y|$, component in the y-direction. (d) Gradient image, $|G_x| + |G_y|$.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



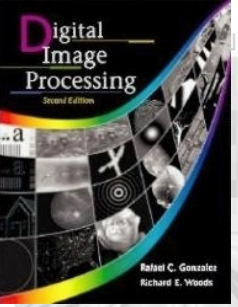
Edge detection example (cont'd)



a b
c d

FIGURE 10.11
Same sequence as in Fig. 10.10, but with the original image smoothed with a 5×5 averaging filter.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Edge detection example (cont'd)

0	1	1	−1	−1	0
−1	0	1	−1	0	1
−1	−1	0	0	1	1

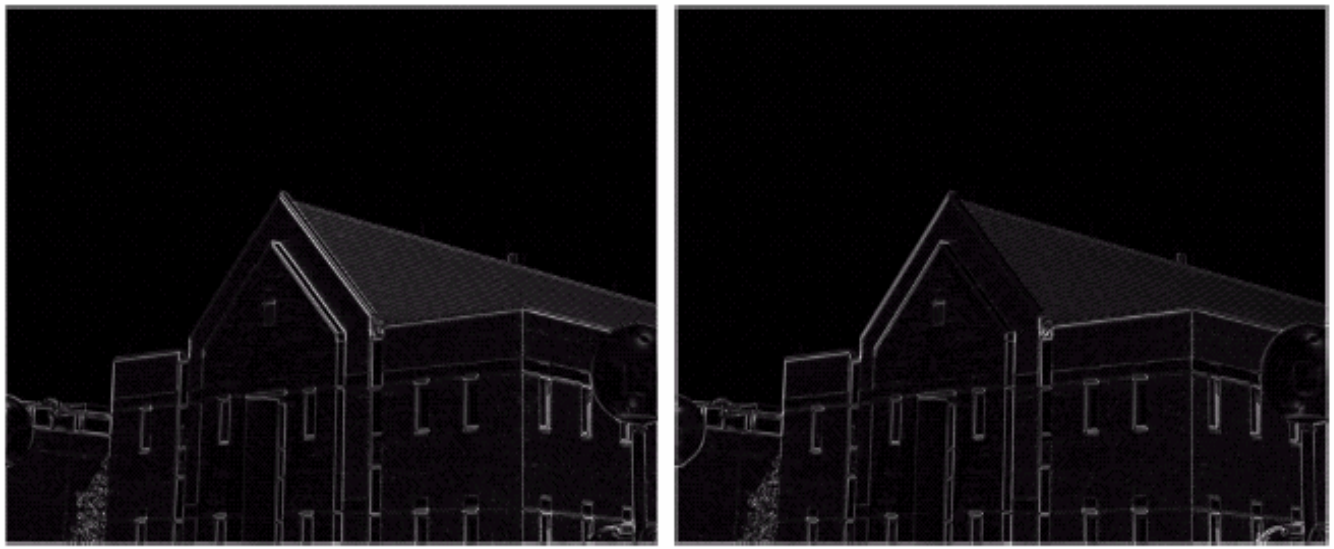
Prewitt

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

Sobel

a b
c d

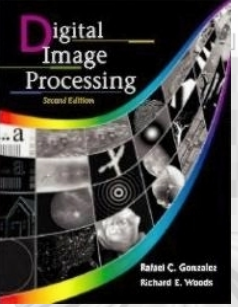
FIGURE 10.9 Prewitt and Sobel masks for detecting diagonal edges.



a b

FIGURE 10.12 Diagonal edge detection. (a) Result of using the mask in Fig. 10.9(c). (b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



2nd Derivative: Laplacian Operator

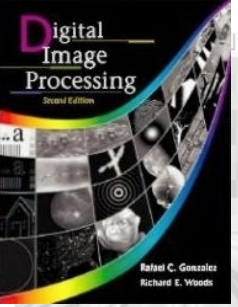
- Review: The Laplacian operator (∇^2) is a very popular operator approximating the second derivative which gives the gradient magnitude only.
- We discussed this operator in spatial filtering $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
- It is isotropic

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

4-neighborhood

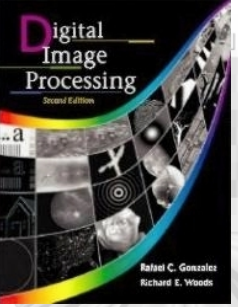
$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

8-neighborhood



Issues with Laplacian

- Problems:
  - Unacceptably sensitive to noise
  - Magnitude of Laplacian results in double edges
  - Does not provide gradient, so can't detect edge direction
- Fixes:
  - Smoothing
  - Using zero-crossing property for edge location
  - Not for gradient direction, but for establishing whether a pixel is on the dark or light side of an edge



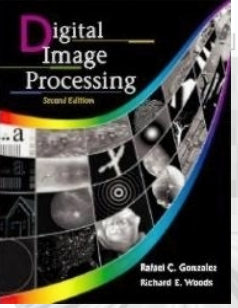
Smoothing for Laplacian

- Our goal is to get a second derivative of a smoothed 2D function $f(x, y)$
- We have seen that the Laplacian operator gives the second derivative, and is non-directional (isotropic).
- Consider then the Laplacian of an image $f(x, y)$ smoothed by a Gaussian.
- This operator is abbreviated as **LoG**, from **Laplacian of Gaussian**:

$$\nabla^2 [h(x, y) * f(x, y)]$$

- The order of differentiation and convolution can be interchanged due to linearity of the operations:

$$[\nabla^2 h(x, y)] * f(x, y)$$



Laplacian of Gaussian (LoG)

- Let's make the substitution $r^2 = x^2 + y^2$ where ***r*** measures distance from the origin.
- Now we have a 1D Gaussian to deal with

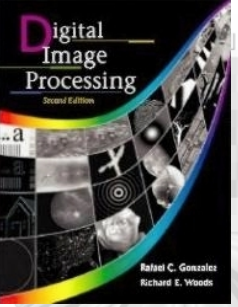
$$h(r) = ce^{\frac{-r^2}{2\sigma^2}}$$

- Laplacian of Gaussian becomes

$$\nabla^2 h(r) = \frac{c}{\sigma^2}\left(\frac{r^2}{\sigma^2} - 1\right)e^{\frac{-r^2}{2\sigma^2}} = c_1\left(\frac{x^2 + y^2 - \sigma^2}{\sigma^2}\right)e^{\frac{-\left(x^2+y^2\right)}{2\sigma^2}}$$

Normalize the sum of the mask elements to 0

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



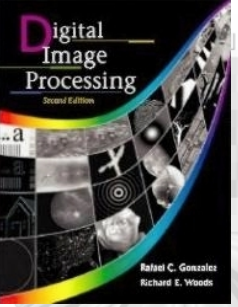
Marr and hildreth's approach

- Smooth the image to reduce noise
- Then calculate the 2nd derivative
- Finally, find the zero-crossing
  - LoG (Laplacian of Gaussian, Mexican hat function)

$$\nabla^2[h(x,y,\delta) * f(x,y)] = \nabla^2[h(x,y,\sigma)] * f(x,y)$$

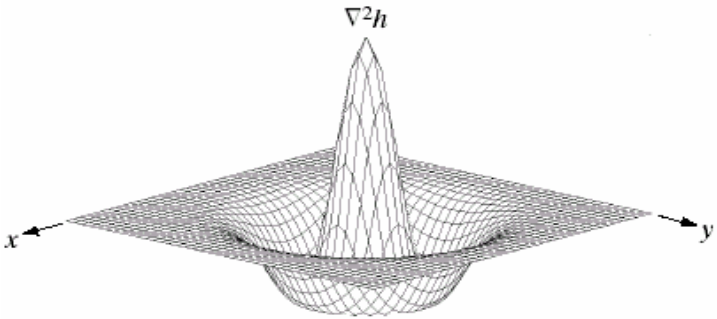
$$h(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \quad r^2 = x^2+y^2$$

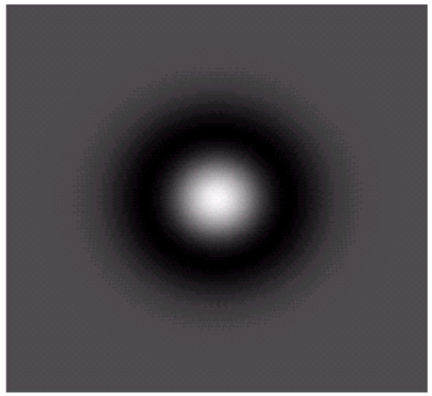
$$\nabla^2 h = \left(\frac{r^2-\sigma^2}{\sigma^4}\right)\exp\left(-\frac{r^2}{2\sigma^2}\right)$$

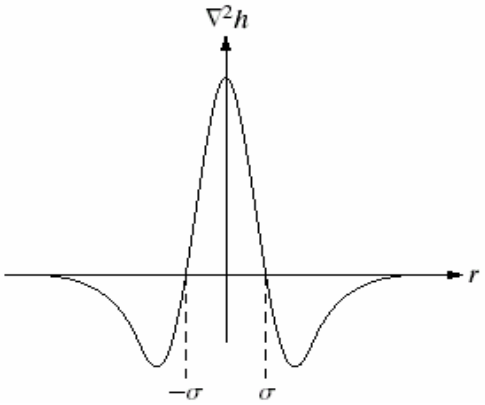


Laplacian of Gaussian (LoG)

- Because of its shape, the LoG operator is commonly called a Mexican hat.





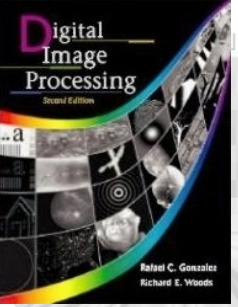


0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

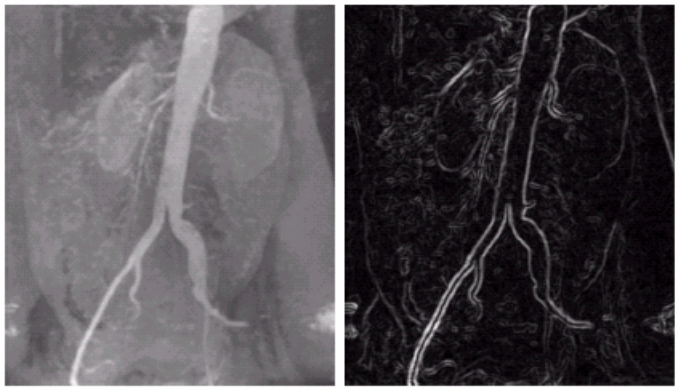
a b
c d

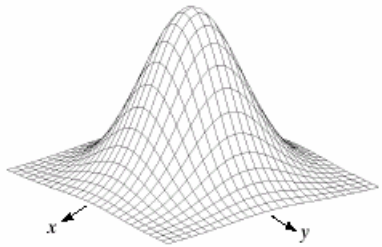
FIGURE 10.14 Laplacian of a Gaussian (LoG). (a) 3-D plot. (b) Image (black is negative, gray is the zero plane, and white is positive). (c) Cross section showing zero crossings. (d) 5 × 5 mask approximation to the shape of (a).

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

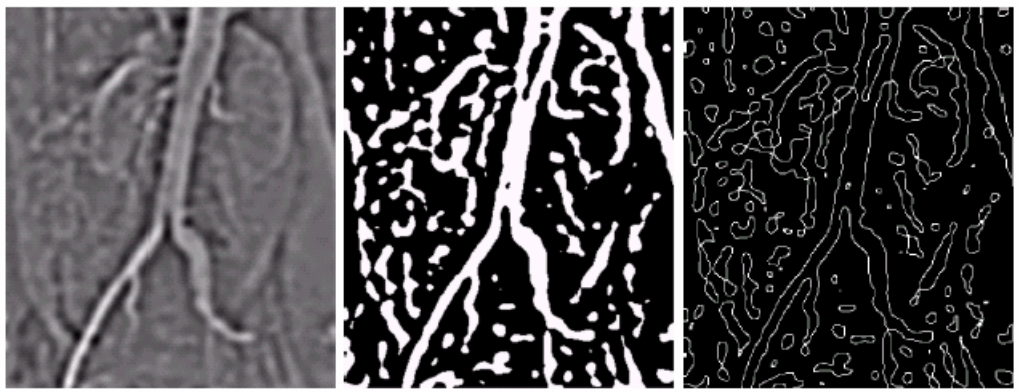


Gradient operators – examples





–1	–1	–1
–1	8	–1
–1	–1	–1



Zero-Crossing:

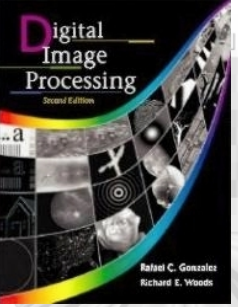
Advantages:

- noise reduction capability;
- edges are thinner.

Drawbacks:

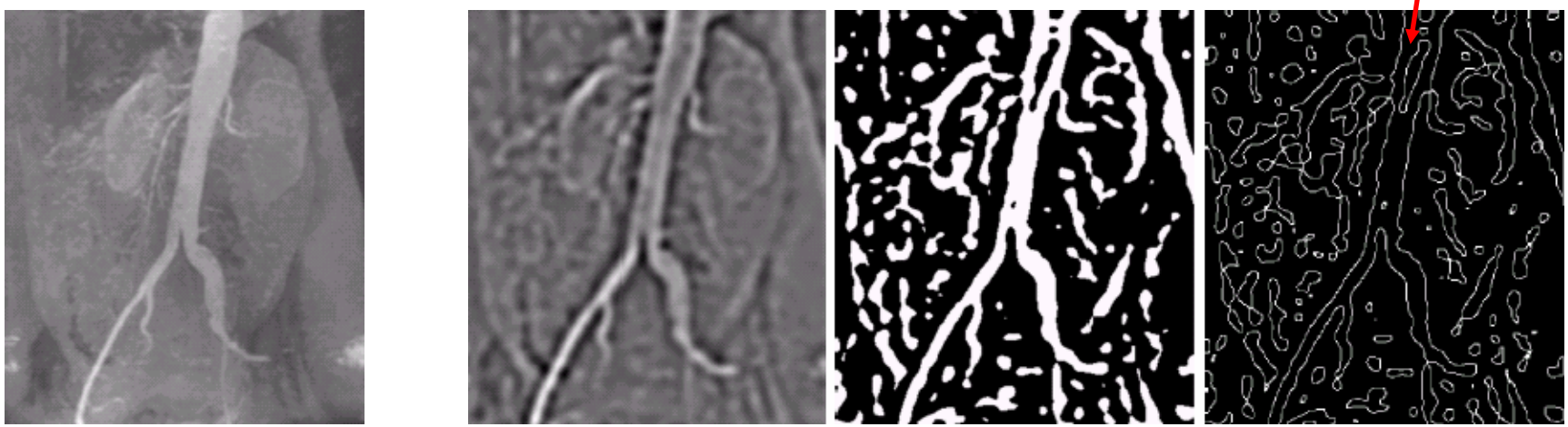
- edges form numerous closed loops (spaghetti effect);
- computation complex.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

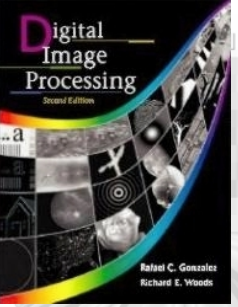


Illustration

- One simple method for approximating zero-crossing:
  - Setting all + values to white, - values to black.
  - Scanning the thresholded image and noting the transition between black and white.

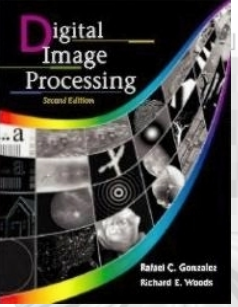


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



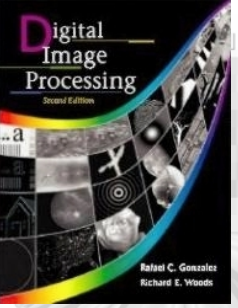
Discussion

- Edge detection by gradient operations tends to work well when
  - Images have sharp intensity transitions
  - Relative low noise
- Zero-crossing approach work well when
  - Edges are blurry
  - High noise content
  - Provide reliable edge detection



Edge Linking and Boundary Detection

- Edge linking
  - How to deal with gaps in edges?
  - How to deal with noise in edges?
  - Linking points by determining whether they lie on a curve of a specific shape



Edge linking – Local Processing

- Analyze the characteristics of the edge pixels in a small neighborhood
  - Its magnitude
  - Its direction

$$\left|\nabla f(x,y) - \nabla f(x_0,y_0)\right| \le E$$

$$\left|\alpha(x,y) - \alpha(x_0,y_0)\right| < A$$

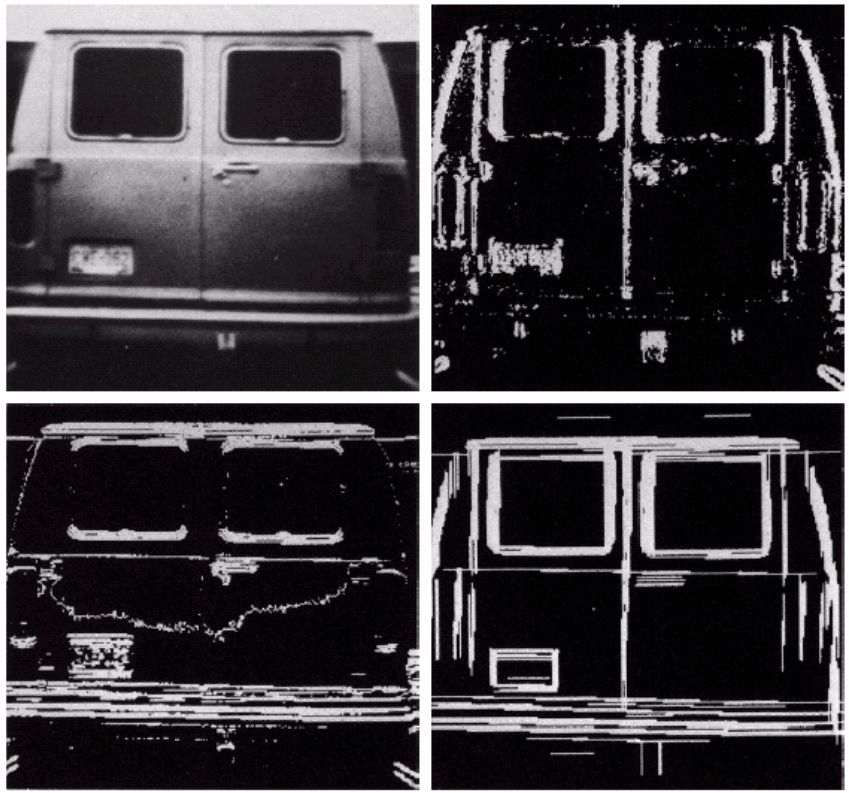
E: nonnegative threshold E=25

A: nonnegative angle threshold A=15°

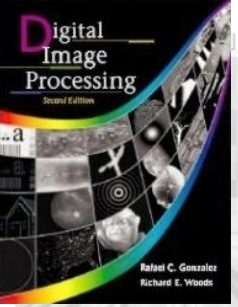
(x_0,y_0): an edge point

a b
c d

FIGURE 10.16
(a) Input image. (b) G_y component of the gradient. (c) G_x component of the gradient. (d) Result of edge linking. (Courtesy of Perceptics Corporation.)

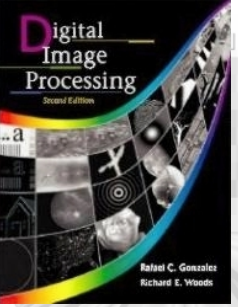


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



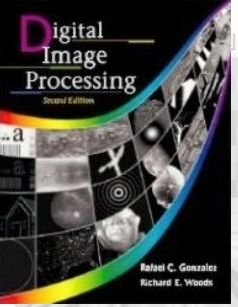
What is missing?

- Q1: Is a pixel an edge element or not?
  - Canny Edge Detection
- Q2: If the edge elements are not connect, how to establish the boundary between regions?
  - Hough Transform



Canny Edge Detector

- Canny edge detector answers the first question.
- Characteristics of the edge detector:
  - Low error rate -- finding all edges and nothing but edges.
  - Localization of edges – distance between edges found and actual edges should be minimized.
  - Single response – no multiple edge pixels when only a single edge exists.



Edge Function in Matlab

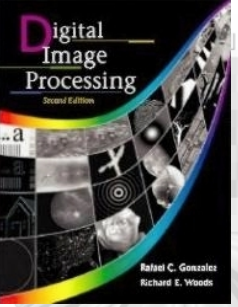
Syntax

```
BW = edge(I,'canny')
BW = edge(I,'canny',thresh)
BW = edge(I,'canny',thresh,sigma)
```

sensitivity thresholds

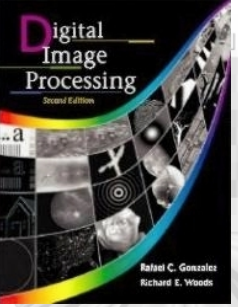
smoothing

- I: intensity image, BW: binary image
- THRESH: is a two-element vector in which the first element is the low threshold, and the second element is the high threshold.
- The Canny method uses 2 thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges.
  - This method is therefore less likely than the others to be "fooled" by noise, and more likely to detect true weak edges.

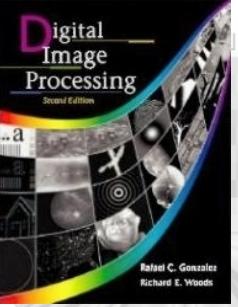


Example

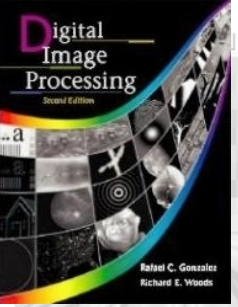
```
I = imread('Lena.tif');
bw =  edge(I,'canny');
```



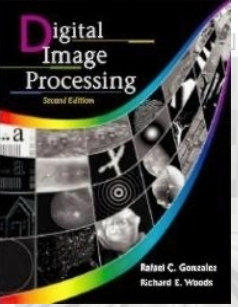
Example: A =edge(lena,'canny',[], 1);



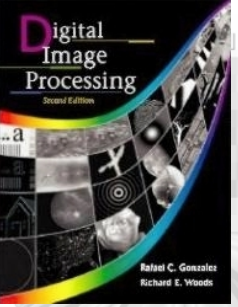
Example: B =edge(lena,'canny',[], 2);



Example: C =edge(lena,'canny',[], 3);

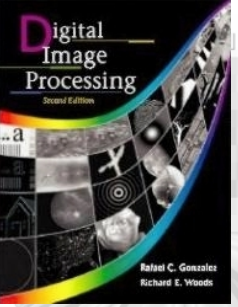


Example: D =edge(lena,'canny',[], 4);



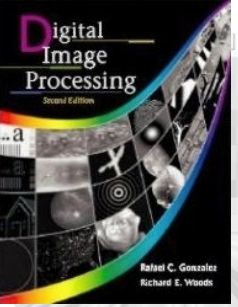
Why is Canny so dominant?

- Still widely used after 20 years.
- Theory is nice (but end result same).
- Details good (magnitude of gradient).
- Hysteresis an important heuristic.
- Code was distributed.
- Perhaps this is about all you can do with linear filtering.



Hough Transform

- Results of edge-detection methods may contain sparse points, instead of straight lines or curves.
- Therefore, need to fit a line to the edge points.
- Efficient solution: Hough Transform
  - Originally for finding lines
  - Easily varied to find other shapes



Simple Example

- Consider a point (x_i,y_i) and the general equation of a straight line in slope-intercept form:

$$y_i = ax_i + b$$

intercept

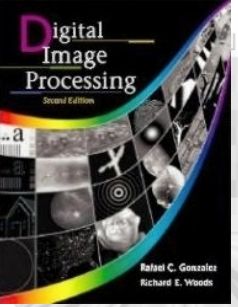
slope

  - Q: How many lines may pass through (x_i,y_i)?

- Re-writing the equation in *ab*-plane

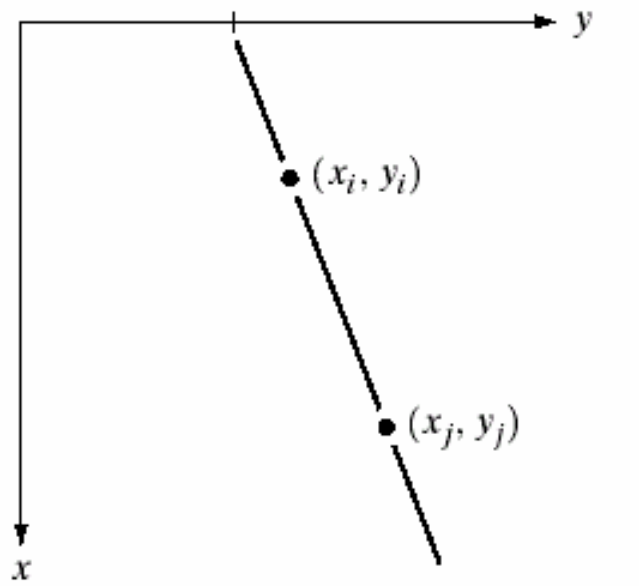
$$b = -x_i a + y_i$$

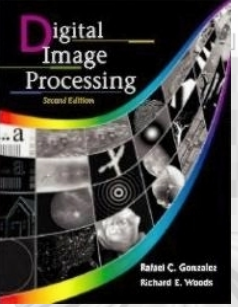
  - How many lines do we get for a fixed (x_i,y_i)?



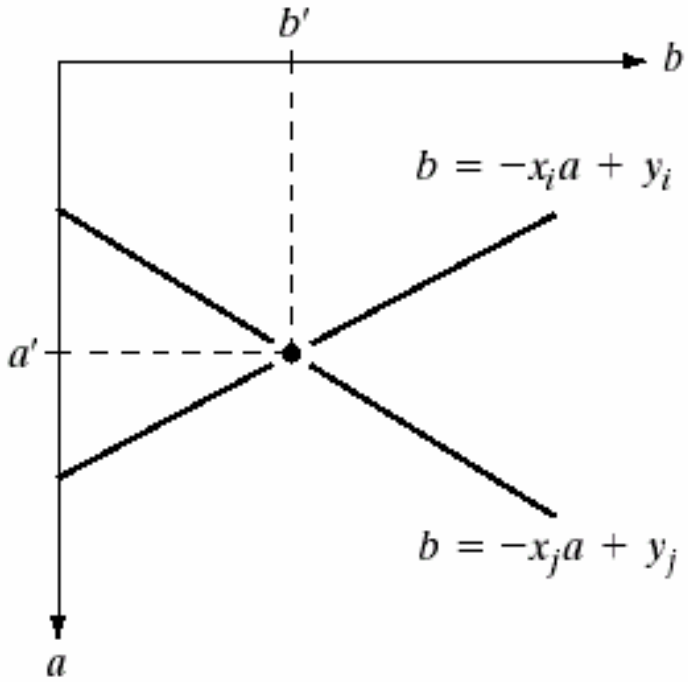
Finding (a,b)

- Now given another point (x_j, y_j), how to find the parameter (a',b') which defines the line that contains both points?

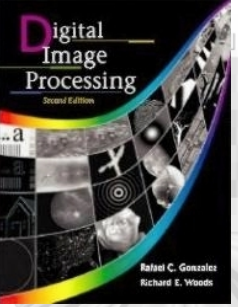




Finding (a,b) (cont'd)

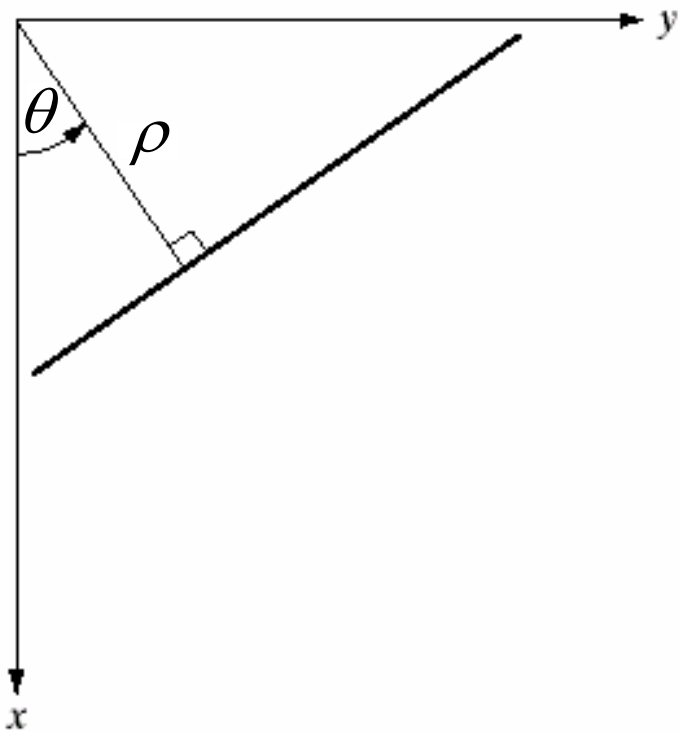


- All points on this line have lines in parameter space that intersect at *(a',b')*.



What is the problem?

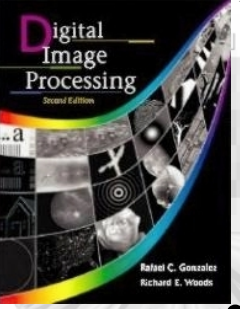
- How about a vertical line?
  - Gradient/slope is infinite
- Need another parameterization scheme. Now consider



ρ is the shortest distance from the line to the origin, and θ is the angle

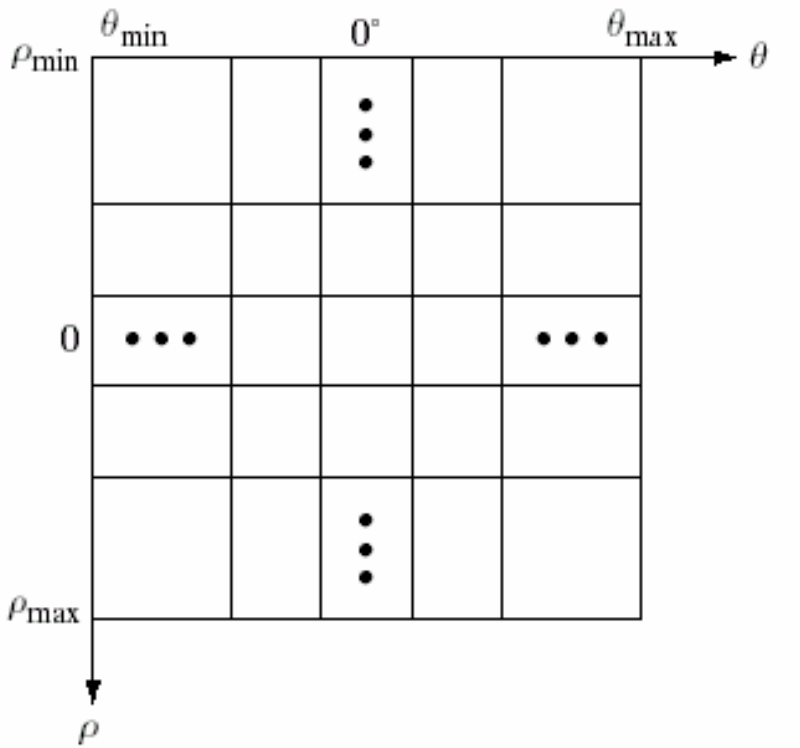
New parameterization:

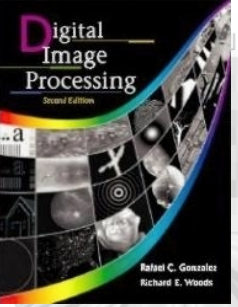
$$x\cos\theta + y\sin\theta = \rho$$



Computation for Hough Transform

- Choosing a discrete set of values of ρ and θ.
- Construct an accumulator array, initialized with 0 for each entry.
- Picture this process as a voting process



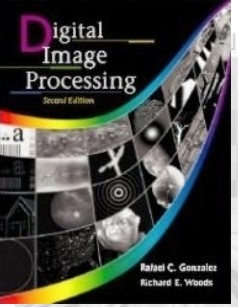


Computation for Hough Transform (cont'd)

- For each edge point (x,y) in the image,
  - we compute

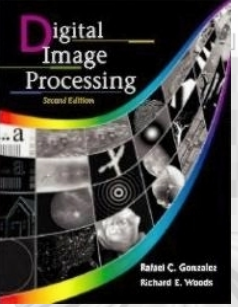
$$\rho = x\cos\theta + y\sin\theta$$

    for each θ
  - Increment the counter for the cell of the resulting (ρ, θ)
- At the end, the (ρ, θ) with highest count correspond to strongest line in the image.



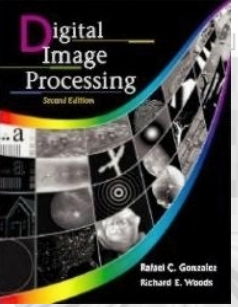
Edge linking - Hough transform

- Can tolerate noise and gaps in edge image
- Look for solutions in a parameter space
- Classical Hough transform
  - Detect simple shape
  - Line detection
  - Circle detection
- Generalized Hough Transform
  - Detect complicated shapes

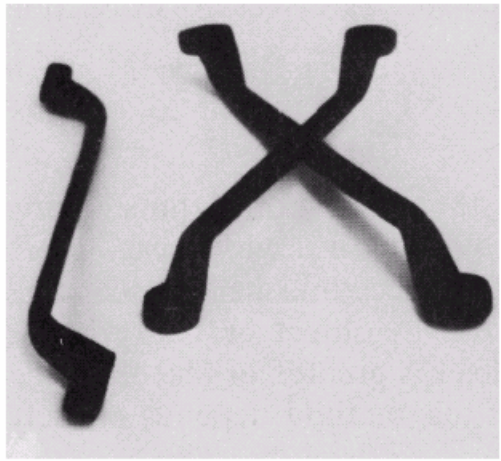


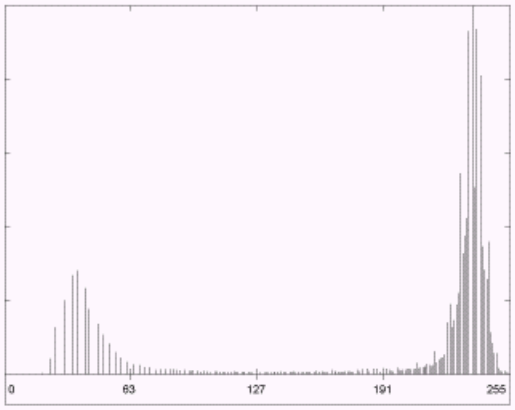
Thresholding

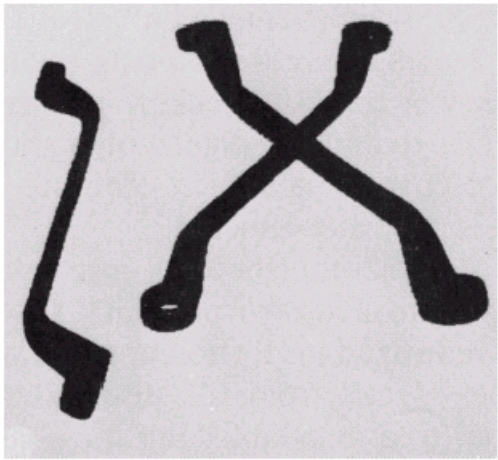
- What is thresholding?
- Any point *(x,y)* for which $f(x,y) > T$ is called an object point; otherwise, the point is called a background point.
- Important for segmentation



Example



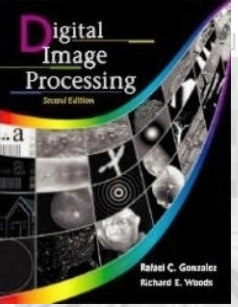




a
b c

FIGURE 10.28
(a) Original image. (b) Image histogram. (c) Result of global thresholding with T midway between the maximum and minimum gray levels.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



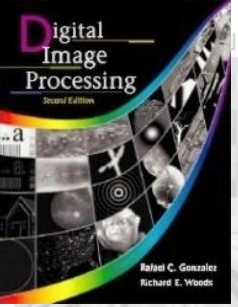
General Formulation

Some local property

$$T = T\left[x, y, p(x, y), f(x, y)\right]$$

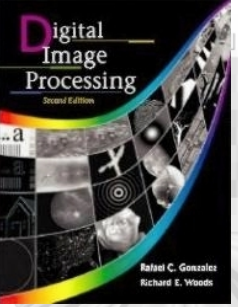
Grayscale value

- When *T* depends only on *f(x,y)*, the threshold is called global.
- If *T* depends on both *f(x,y)* and *p(x,y)*, the threshold is local.
- If *T* depends on the spatial coordinates *x* & *y*, the threshold is called dynamic or adaptive.



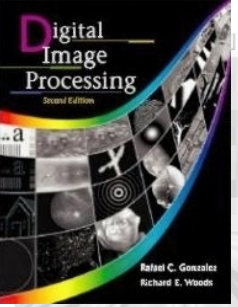
Basic Global Thresholding

1. Select an initial estimate for T
2. Segment the image using T to generate 2 regions, G_1 & G_2.
3. Compute the average gray level values m_1 and m_2 for regions G_1 and G_2, respectively.
4. Compute the new T as $(m_1+m_2)/2$.
5. Repeat the steps 2～4 until the *T* converges.

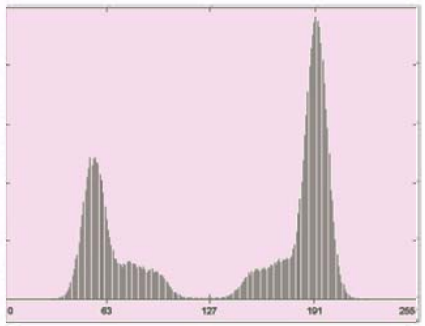


Selection of Initial T

- If foreground and background occupy comparable areas, average graylevel might be good.
- If the areas not comparable, midway between max and min graylevel might be better.



Example

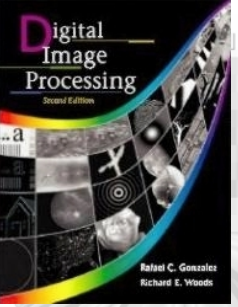


a b
c

FIGURE 10.29
(a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)

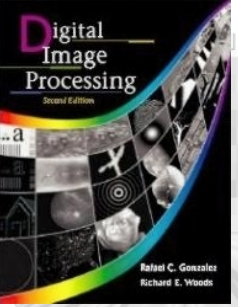
Initial T is the average graylevel
The final T is 125.4 with 3 iterations

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



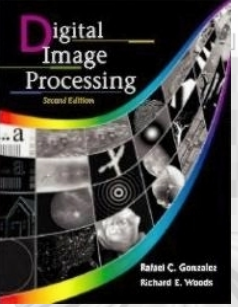
Thresholding method: categories

- Threshold Number
  - Bi-level
  - Multi-level
- Spatial variance
  - Global threshold: thresholds the entire image with a single threshold value.
  - Local threshold: partitions a given image into subimages and determines a threshold for each of these subimages.
- Point dependent or region dependent
- Fixed or adaptive threshold



Thresholding method: categories by Sankur

- According the information exploited
  - Histogram shape based: peaks, valleys and curvatures.
  - Clustering based: pixels are modeled as a mixture of two Gaussians.
  - Entropy base
  - Object attribute based
  - The spatial methods use high-order probability distribution or correlation between pixels
  - Local methods



Otsu's Method (graythresh in Matlab)

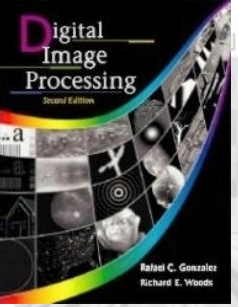
- A popular method that maximizes inter-class variance.
- Let's remind ourselves again that the image histogram is a probability distribution of the gray levels

$$p_i = n_i / N$$

where n_i is the number of pixels of gray level i, and N is the total number of pixels.

- The image average is

$$m = \sum_{i=0}^{L-1} i p_i$$



Otsu's Method (cont'd)

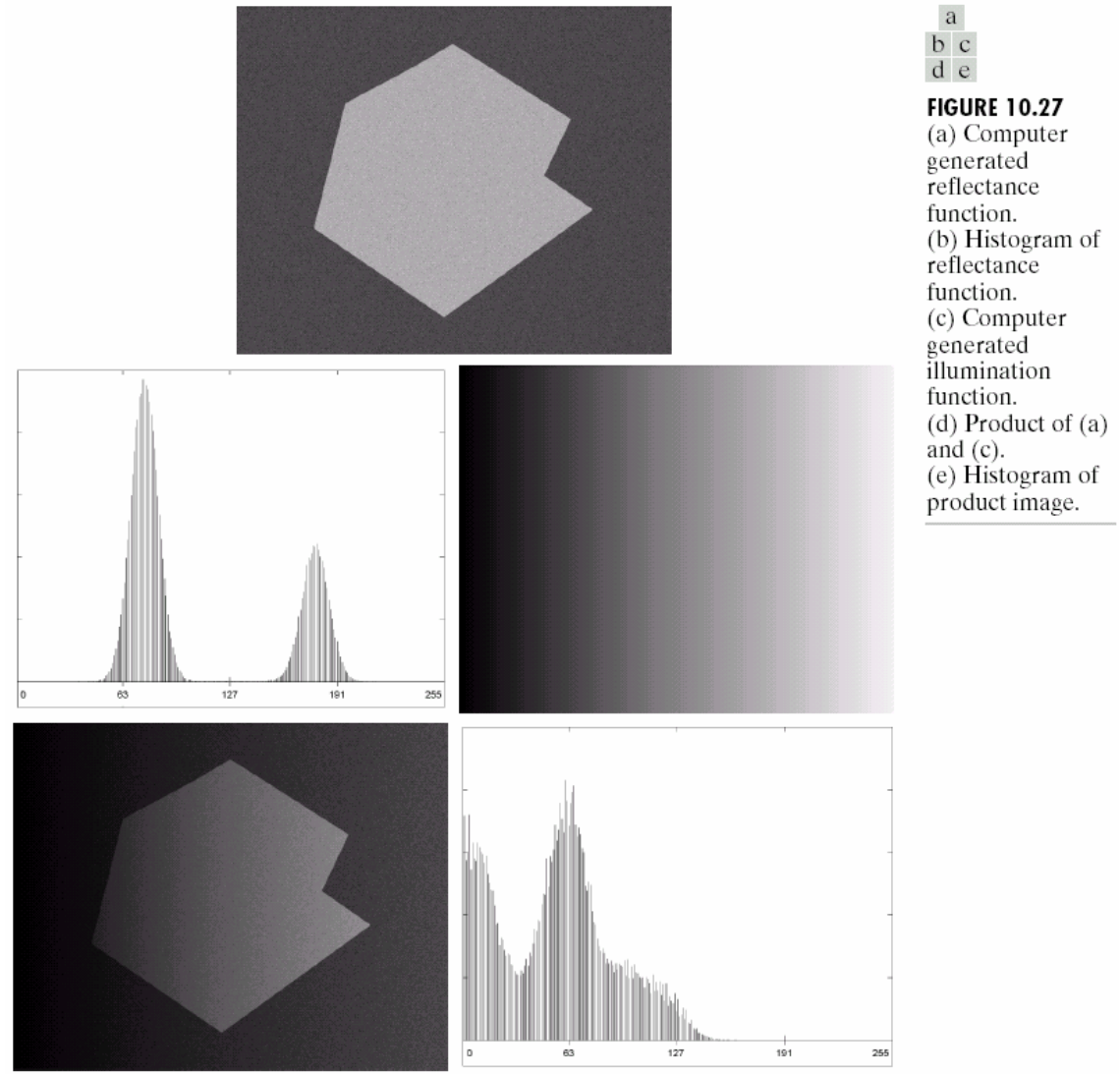
- If we threshold at level k, we get

$$\omega(k)=\sum_{i=0}^{k} p_i,\ \ \mu(k)=\sum_{i=k+1}^{L-1} p_i$$

- We would like to maximize the inter-class variance, i.e. the difference between $\omega(k)\ \&\ \mu(k)$
- k is set to the value that maximize

$$\frac{(m\omega(k)-\mu(k))^2}{\omega(k)\mu(k)}$$

Non-uniform illumination

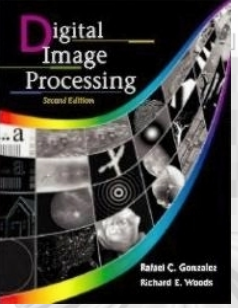


a
b c
d e

FIGURE 10.27
(a) Computer generated reflectance function.
(b) Histogram of reflectance function.
(c) Computer generated illumination function.
(d) Product of (a) and (c).
(e) Histogram of product image.

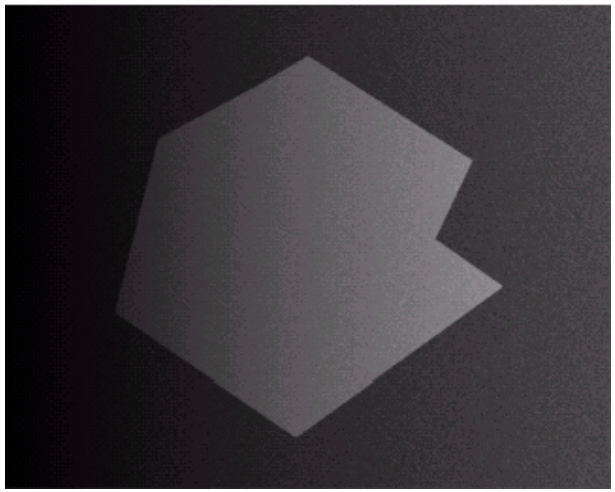
$$f(x,y) = i(x,y)r(x,y)$$

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

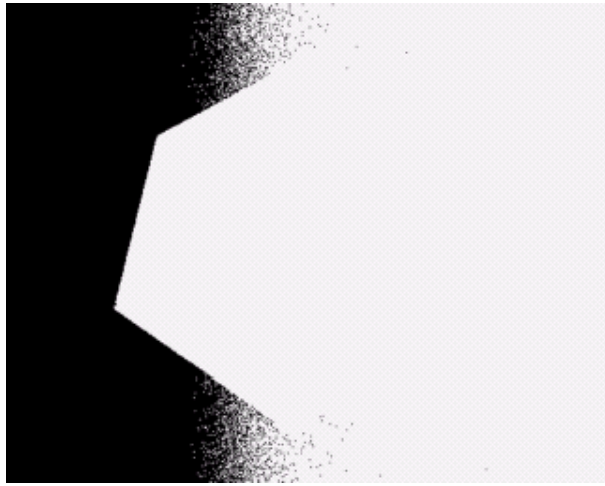


Basic Adaptive Thresholding

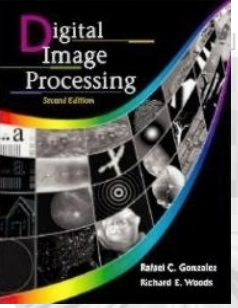
- Global thresholding often fails in the case of uneven illumination.



global thresholding



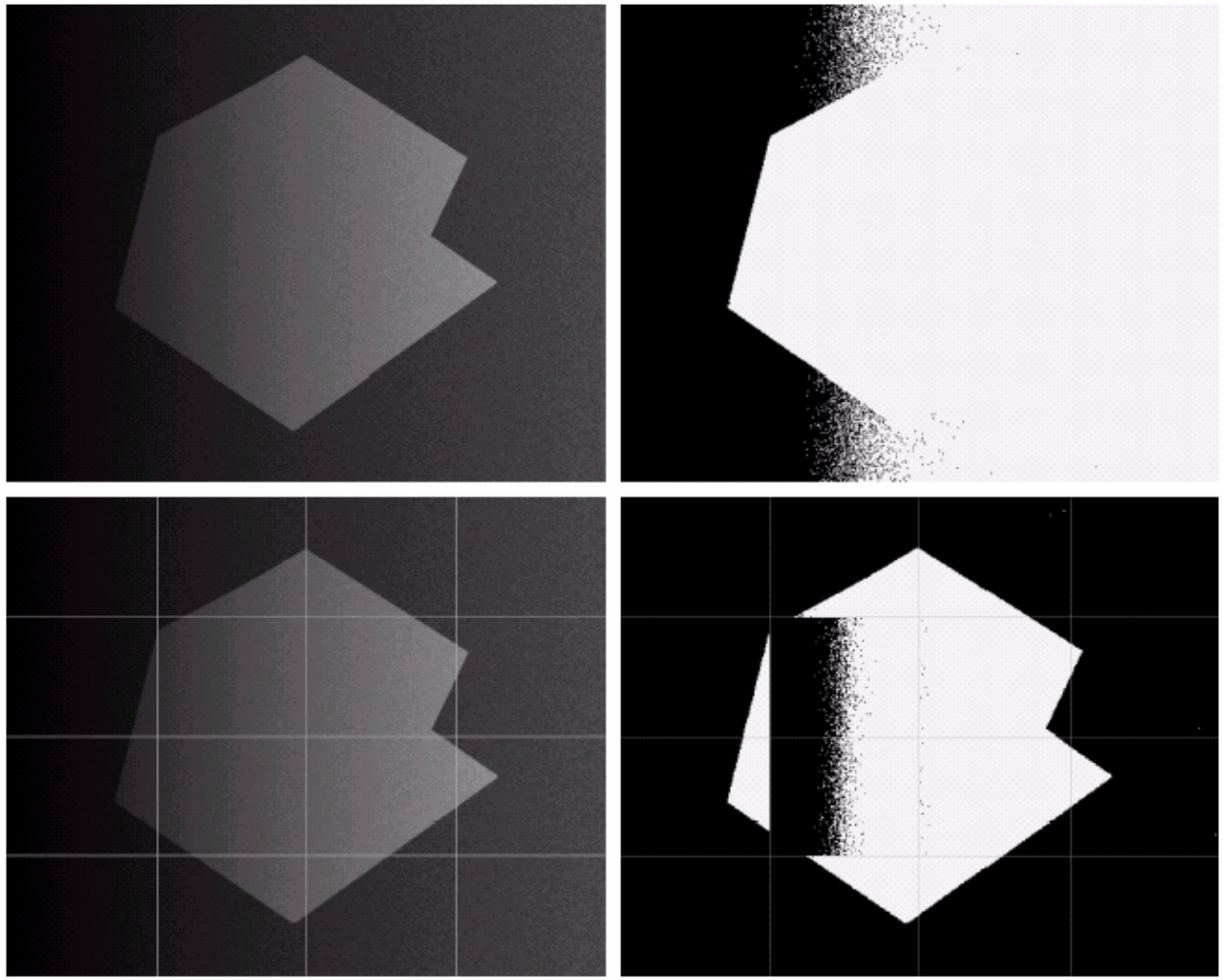
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Adaptive thresholding

a b
c d

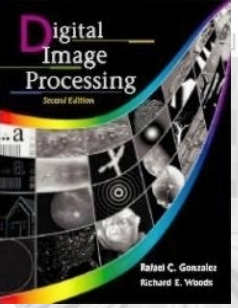
FIGURE 10.30
(a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.



$$T_0 = (f_{\max} + f_{\min})/2$$

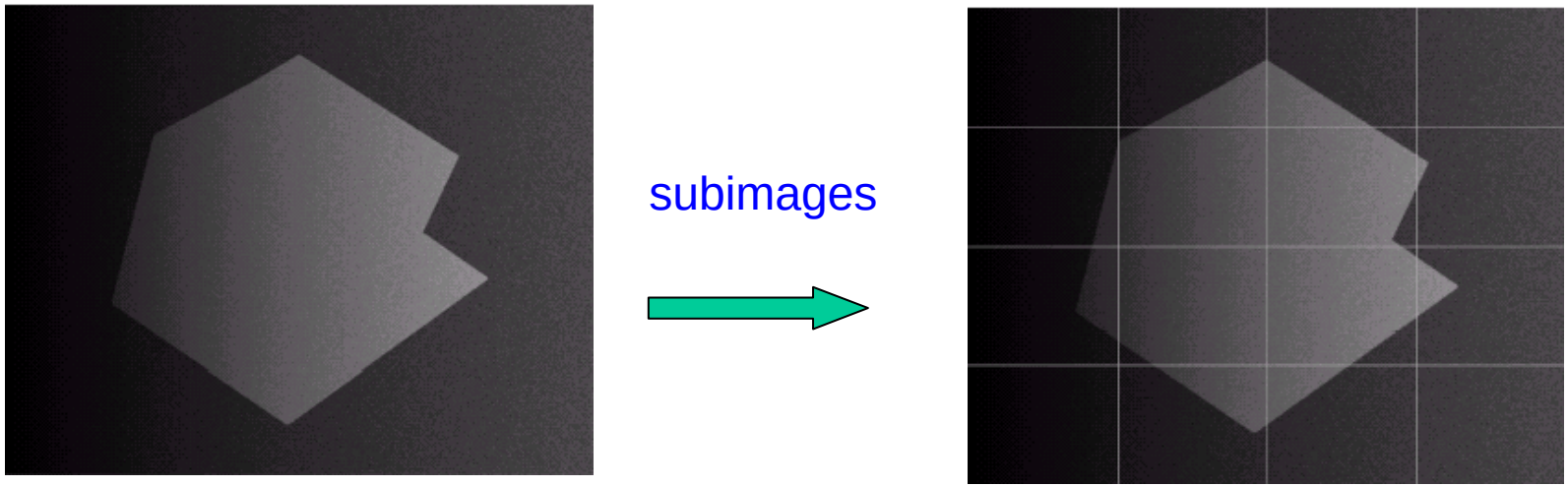
$$T_k = (\mu_1^{k-1} + \mu_2^{k-1})/2$$

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

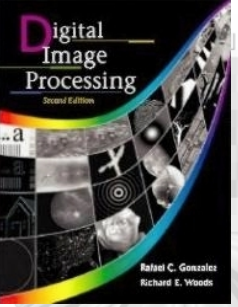


Adaptive thresholding (cont'd)

- The simplest fix is to
  - Divide the image into subimages (but how?)
  - Determine *T* for each subimage (but how?)

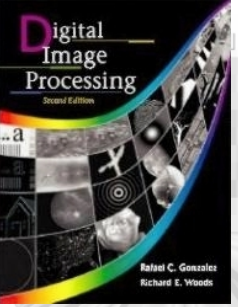


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

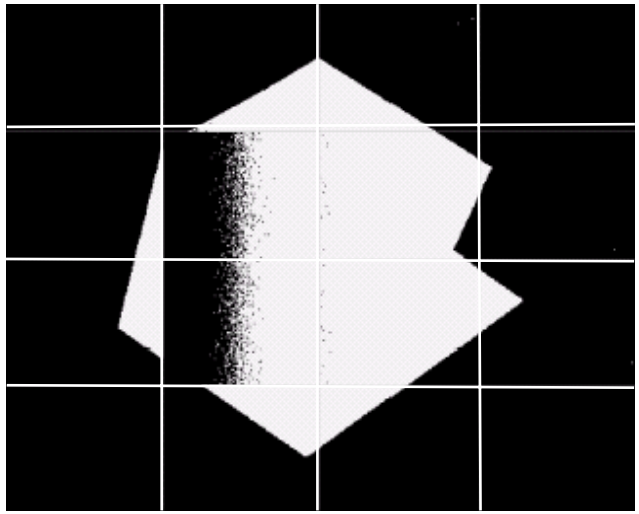


How to deal with the Subimages

- Observation:
  - Type 1: For all subimages, if the graylevel variance is < 75 do not contain object boundary
  - Type 2: For all subimages, if the graylevel variance is > 100 contain object boundary
- All subimages of type 1 are merged and thresholded using a single threshold value
- Each subimage of type 2 is thresholded independently.
- All thresholding is done using basic global method.



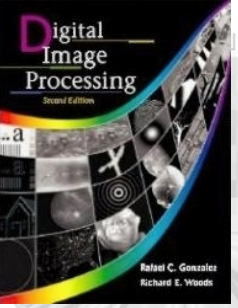
Result



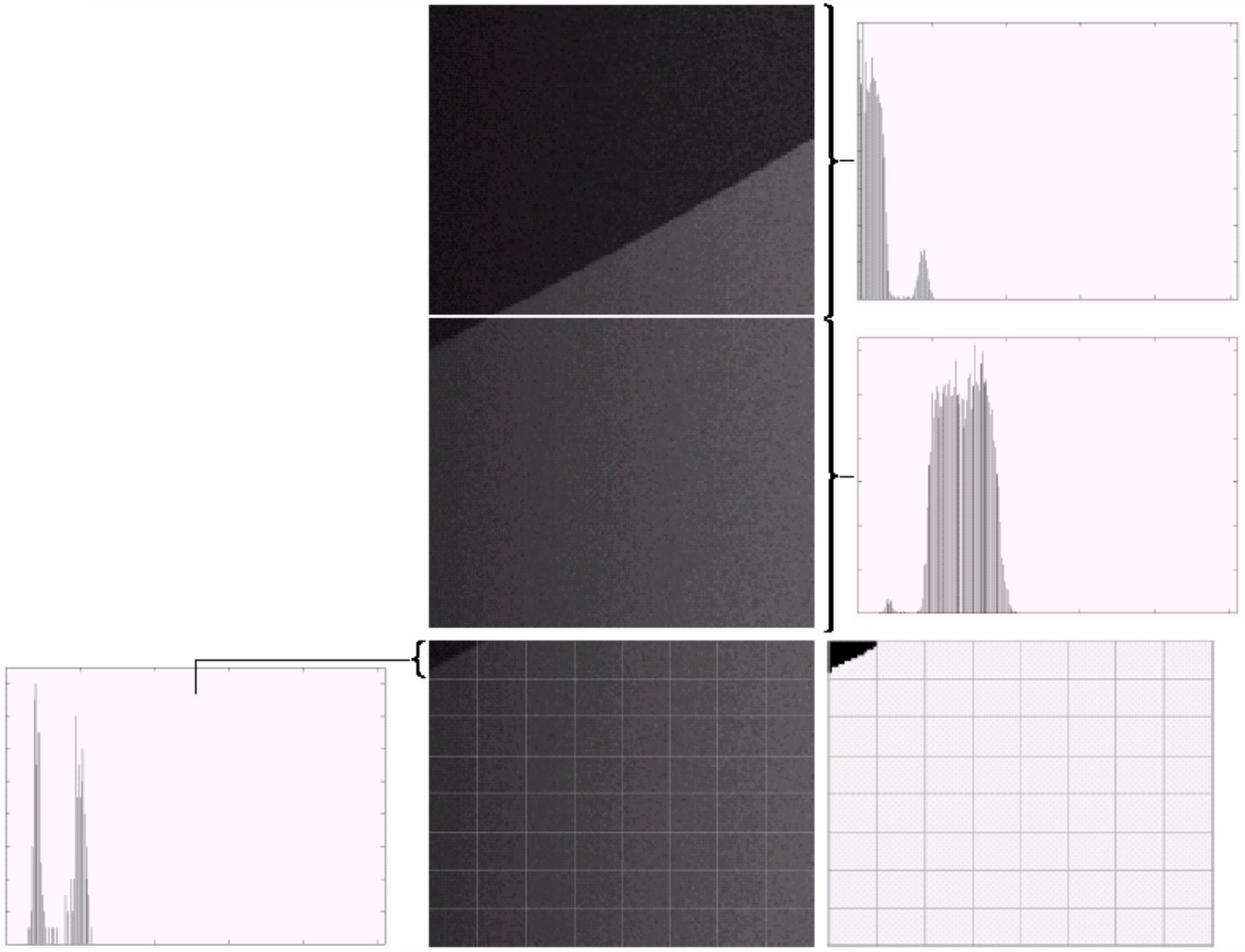
How to be better?

- How can the result be improved?
- There is no the-solution that handles all thresholding problems (hard research topic)

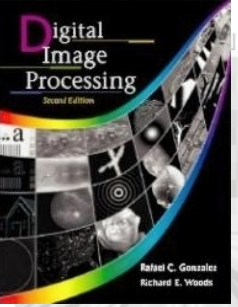
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Adaptive thresholding (cont'd)



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Non-uniform illumination Compensation

- Image formation model

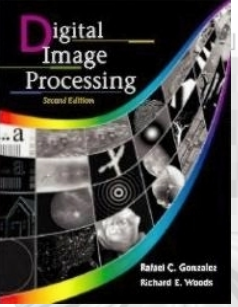
$$f(x,y) = i(x,y)r(x,y)$$

- First take a shot of a constant, white reflective surface

$$g(x,y) = ki(x,y)$$

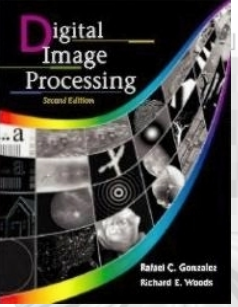
- For any image $f(x,y)$, segment the normalized image $h(x,y)$, instead of $f(x,y)$

$$h(x,y) = f(x,y)/g(x,y) = r(x,y)/k$$



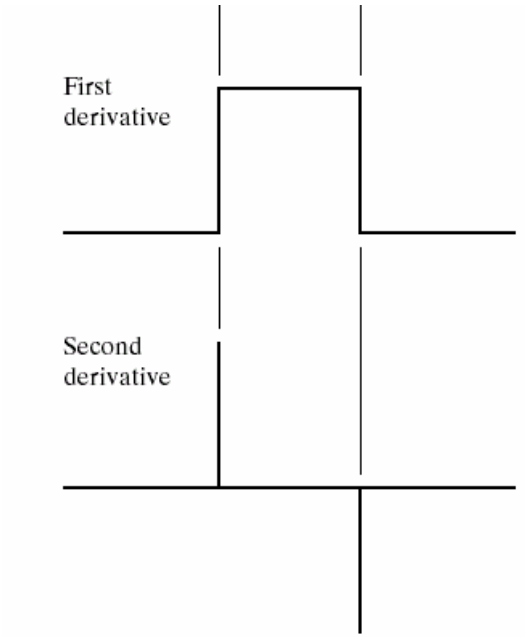
Use of Boundary Characteristics

- It is easier to select T if the histogram peaks are tall, narrow, symmetric and separated by deep valley.
- If relying on only graylevel information, the result is susceptible to relative sizes of the object and background.
- One possible solution: using only pixels on or near the edges.
  - Problem: edges are only know after segmentation, which is what we're trying to achieve now …

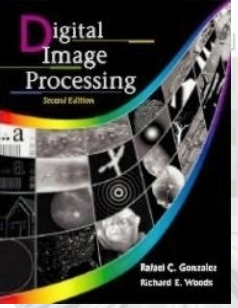


Using Gradient/Laplacian

- There is no need to have the edges extracted first
- Review:
  - The gradient reveals whether a pixel is on edge or not
  - Laplacian indicates the transition of the graylevel on the edge



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Thresholding on Gradient Information

- Threshold is applied to the gradient instead.

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

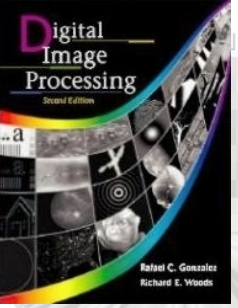
Dark side of edge

Pixels on edge

Bright side of edge

Dark object, light background

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

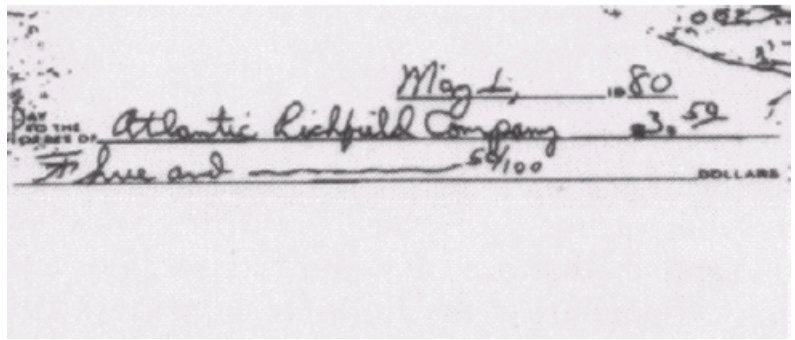


How to turn this into a binary image?

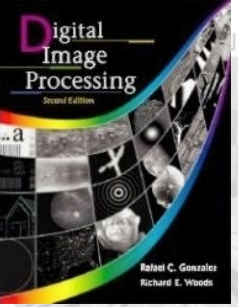
- For dark object, look for the following pattern in the scan line (horizontal or vertical)

(…)(-,+)(0 or +)(+,-)(…)

1 0 1

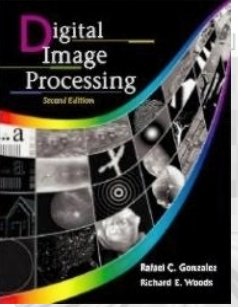


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



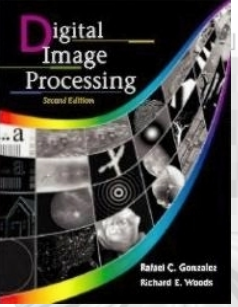
Region-Based Segmentation

- Different from previous techniques, algorithms of this category find regions directly, instead of the boundaries dividing the regions.
- There are few such algorithms
  - Region growing
  - Region splitting and merging



Basic Formulation

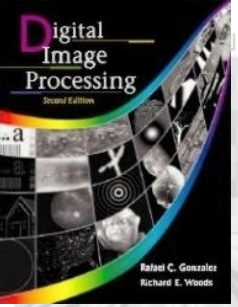
- Let R be the entire region partitioned into n subregions, R_1, R_2, …, R_n such that
  - $\bigcup_{i=1}^{n} R_i = R$
  - R_i is a connected region i=1,2,...,n
  - $R_i \cap R_j = \varnothing$ for $i \neq j$
  - $P(R_i) = true \quad \forall i$
  - $P(R_i \cup R_j) = false$ for any adjacent R_i and R_j
- $P(R_k)$ is a logical predicate defined over the points in R_k.



Region growing

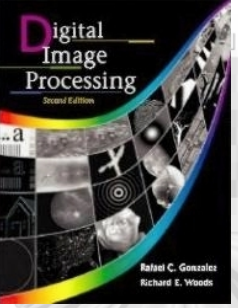
- Groups pixels or subregions into larger regions based on predefined criteria (gray tone or texture).
- Basic method:

  Start with a set of "seed" points and from these grow regions by appending to each seed those neighboring pixels that have properties similar to the seed, such as specific ranges of gray level or color.
- Problems in region growing:
  - Selection of the seeds
  - Criteria of similarity
    - Gray level's similarity / connectivity / texture / moments
  - Formulation of a stopping rule
    - Growing a region should stop when no more pixels satisfy the criteria for inclusion in that region

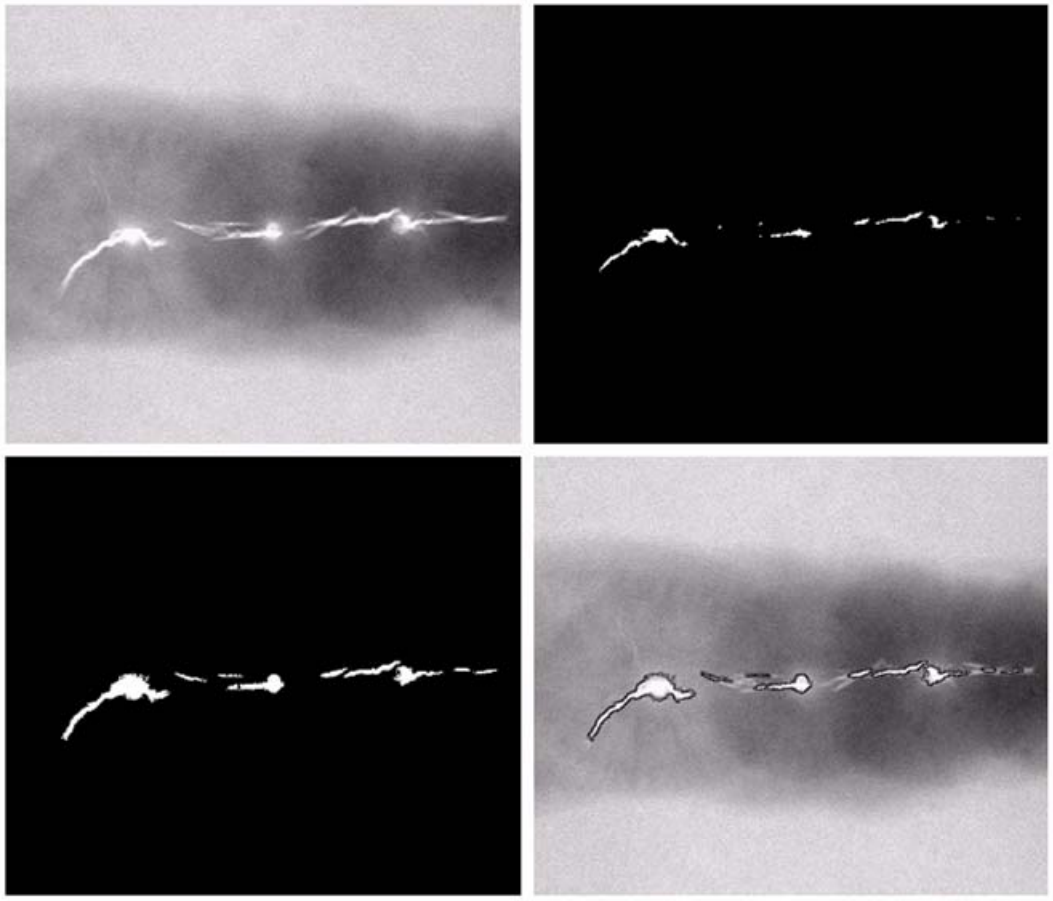


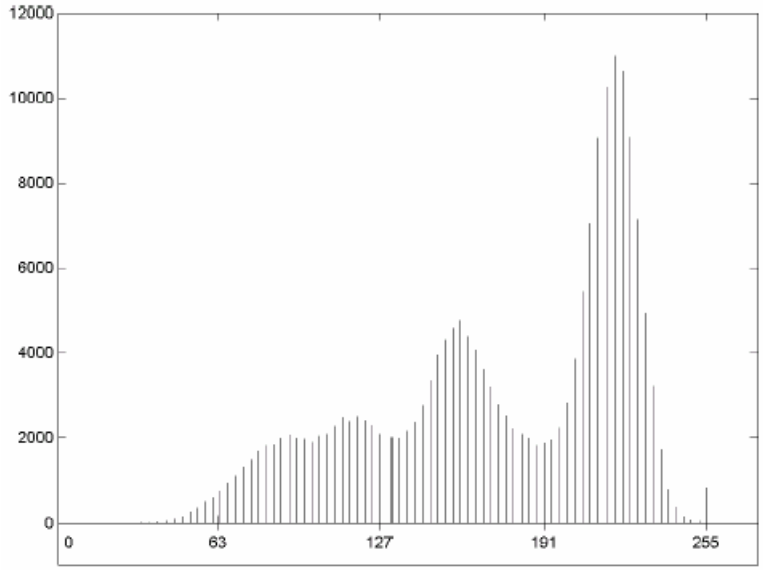
Region growing (cont'd)

- Step 1: Assume we find a good threshold, and use it to partition the regions into pure black and white.
- Step 2: Use different labels to identify different objects
  - Use region growing to connect parts that should have belong to the same region
  - This is called "Connected component analysis"
  - The region with the same label generate one segment

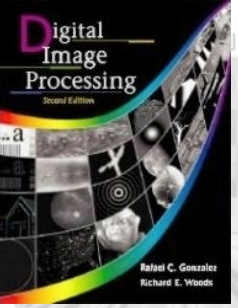


Region growing - example





(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

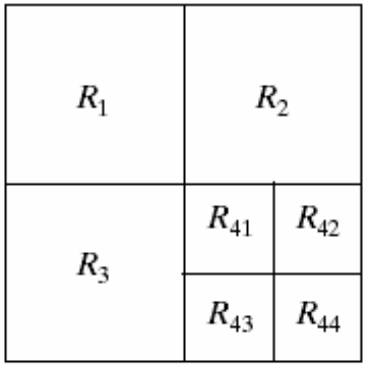


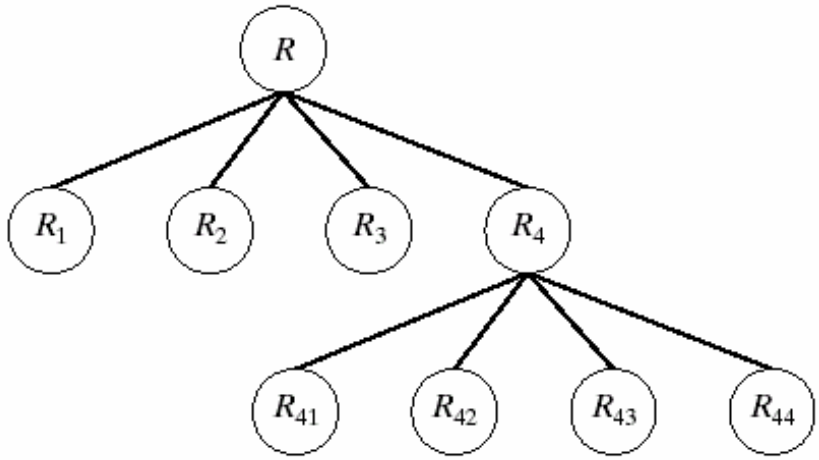
Region Splitting & Merging

- Splitting:
  - Starting with the entire region.
  - If P(R)=*false*, divide the image into quadrant
  - If P is *false* for any quadrant, divide that quadrant into subquadrants.
  - The result is a quadtree

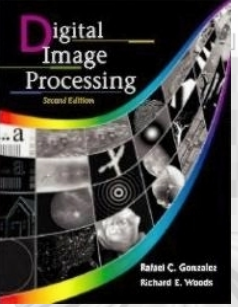
a b

FIGURE 10.42
(a) Partitioned image.
(b) Corresponding quadtree.



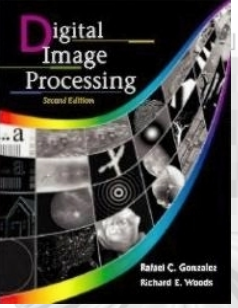


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Region Splitting & Merging (cont'd)

- If only splitting, it is likely that adjacent regions have identical properties.
- Adjacent regions whose combined region satisfies P should be merged.
- Algorithm
  - Splitting into 4 disjoint quadrants if P(R)=*false*
  - Merging any adjacent quadrants for which P is *true*
  - Stop when no further merging and splitting is possible



Example

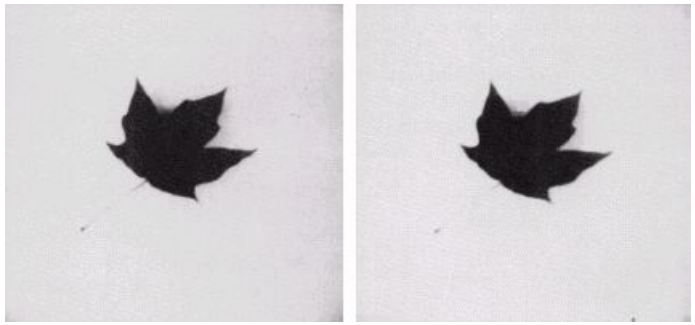
- $P(R_i)$=*true* if at least 80% of the pixels in R_i have the property

$$\left| z_j - m_i \right| \leq 2\sigma_i$$

Pixel intensity value

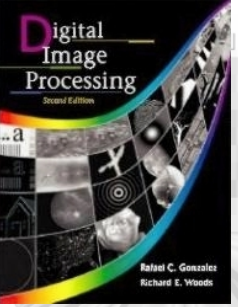
Mean of the region

Std deviation



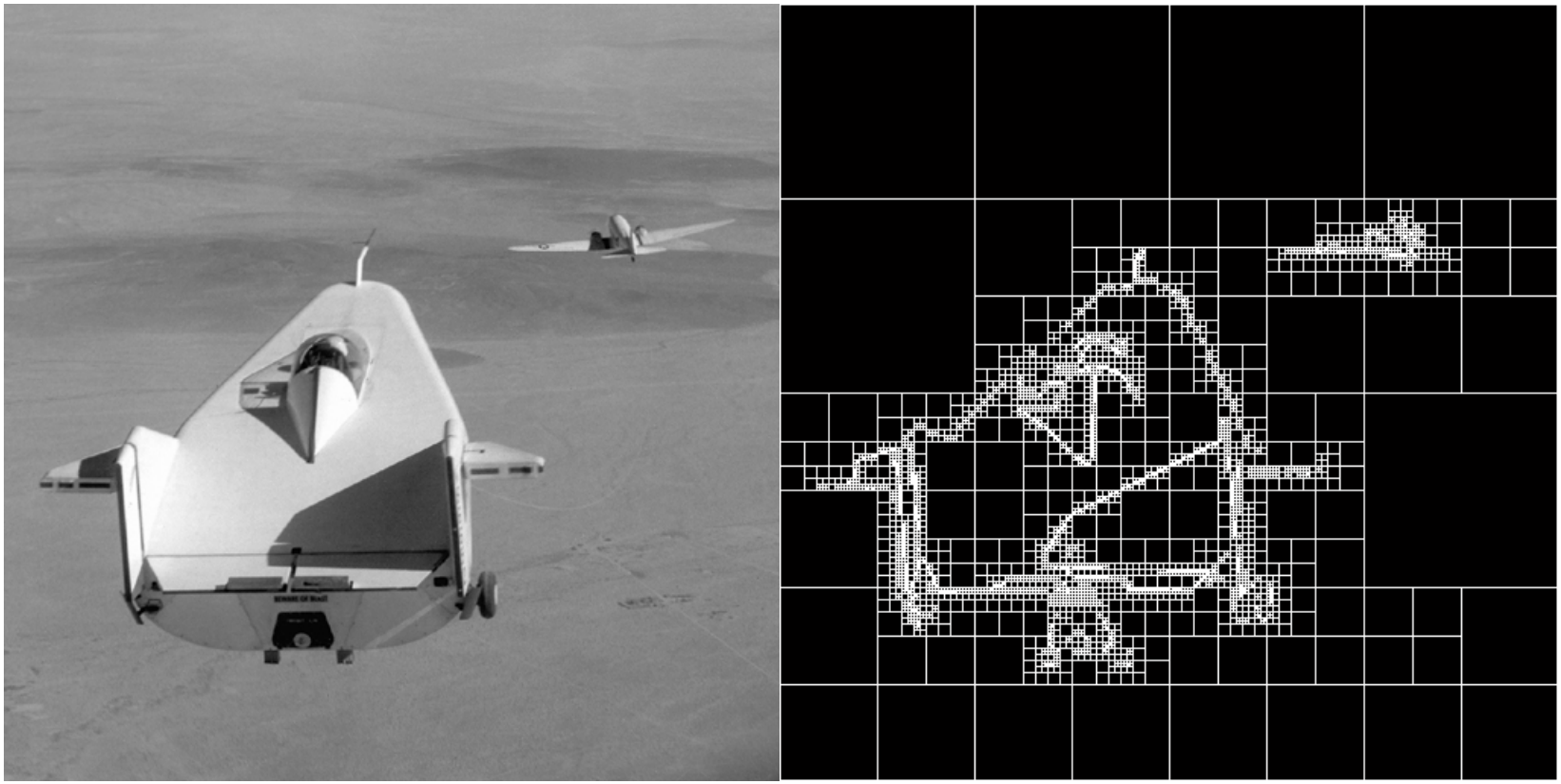
- Can specify P(R) based on texture content for texture segmentation

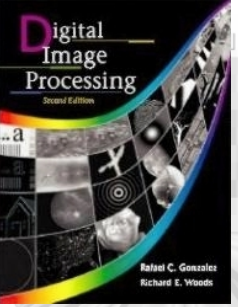
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Region Splitting and Merging

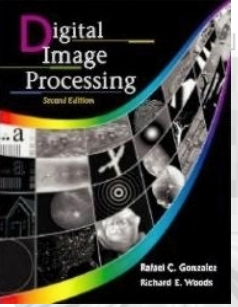
QuadTree Decomposition





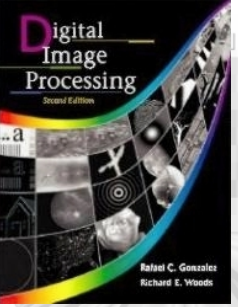
Summary

- Thresholding
  - Global
  - Adaptive
  - Using boundary characteristics
- Region-based segmentation
  - Growing, splitting and merging



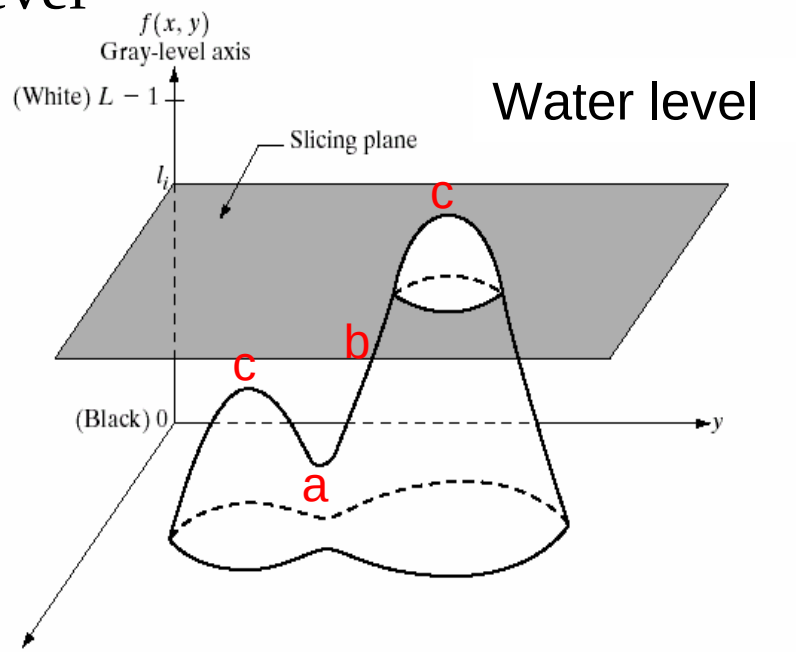
Morphological Watersheds

- Review: We have done
  - Detection of discontinuities
  - Thresholding
  - Region processing
- Morphological watersheds embodies properties from other 3 approaches, therefore more superior.

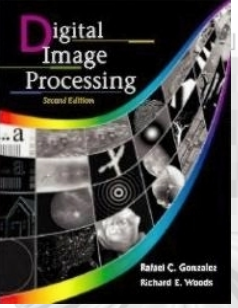


Basic Concept

- Visualize the image in 3D topography (地形)
  - 2 spatial coordinates + graylevel

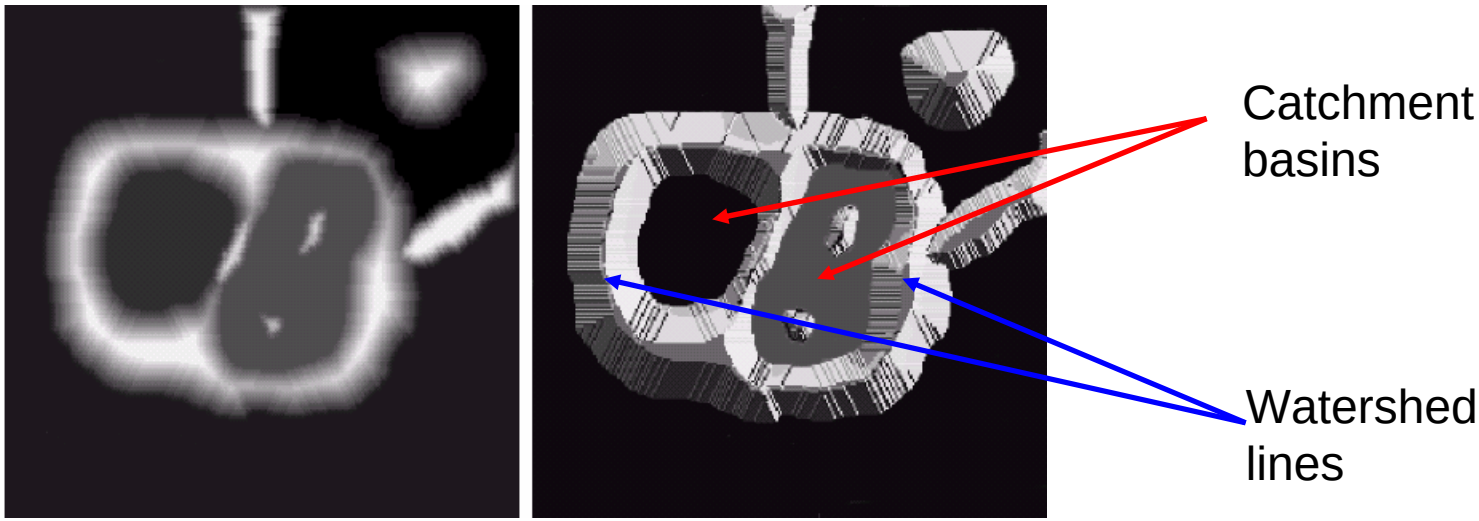


- 3 types of points:
  - (a) Points in a regional minimum
  - (b) Points on hills
  - (c) Points on watershed lines
- How to differentiate (b) & (c)?



Illustration

- *Objective: finding the watershed lines*
  - *Construction of dam to prevent catchment basins from merging*



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

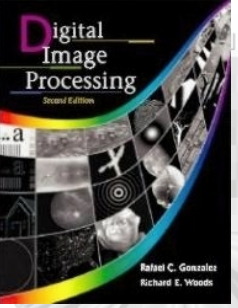
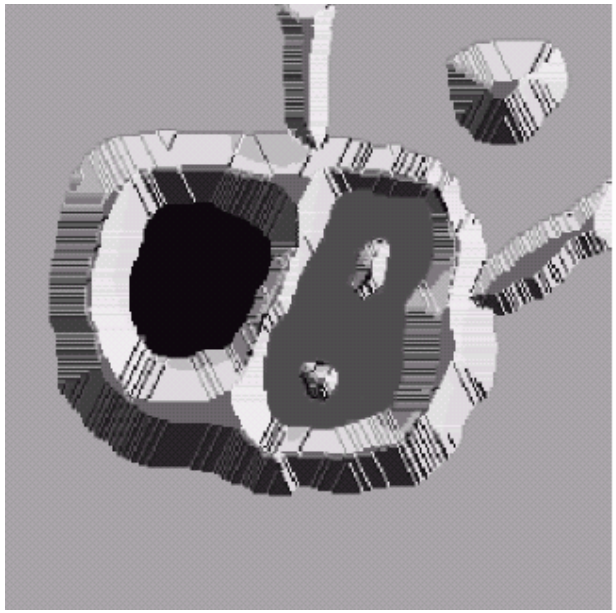
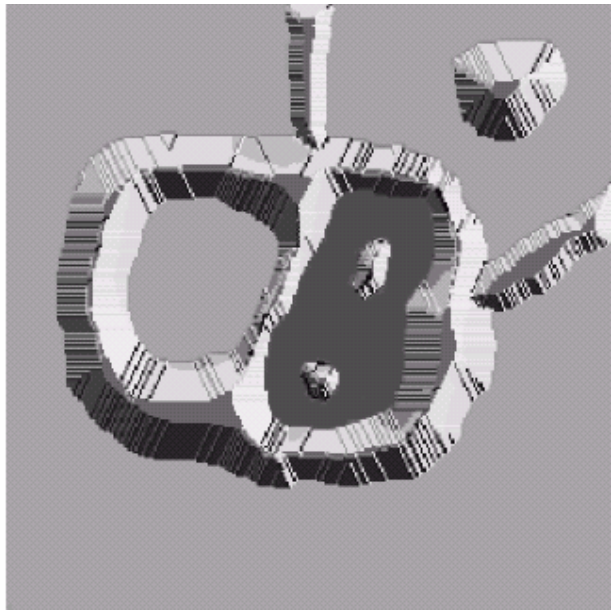


Illustration (cont'd)

- Punch a hole in each regional minimal for water to fill up the catchment basin.





(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

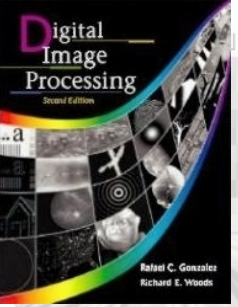
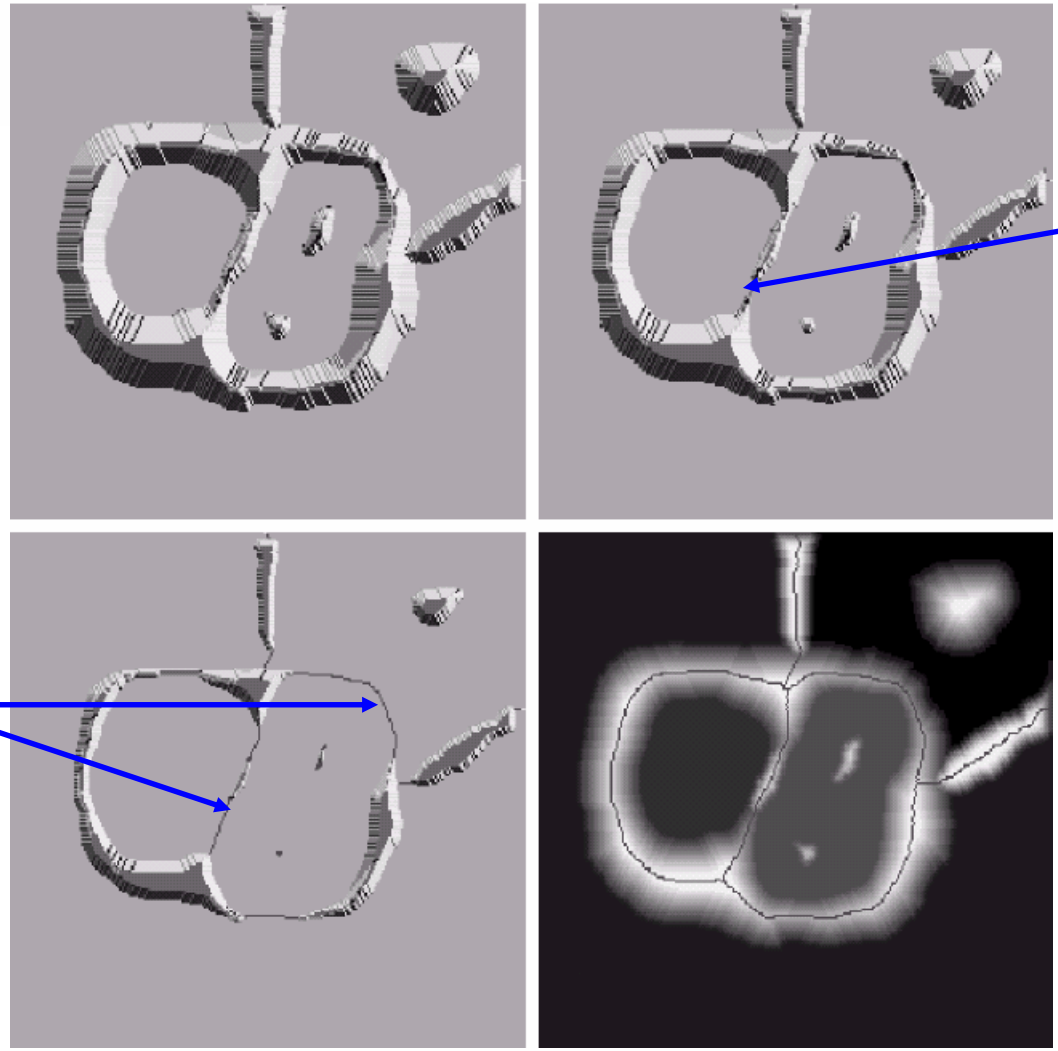


Illustration (cont'd)

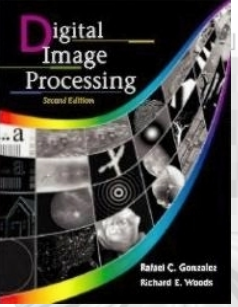


Water start merging, so shorter dam constructed

Longer dam constructed

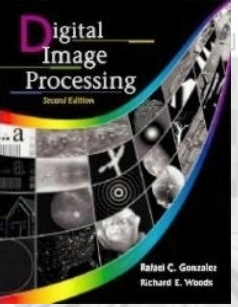
Final result

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



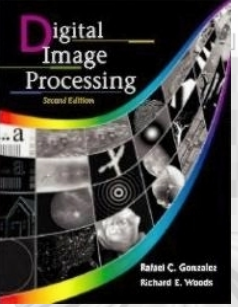
Aside

- Watershed algorithm is often applied to the gradient of an image, rather than to the image itself.
  - Regional minima of catchment basins correlate nicely with the small value of the gradient corresponding the objects of interest
  - Boundaries are highlighted as the watershed lines.
- The important property is that the watershed lines form a connected path, thus giving continuous boundaries between regions.



The Use of Motion in Segmentation

- Motion is a powerful cue used by humans and many animals to extract objects of interest from a background of irrelevant detail.
- Motion arises from a relative displacement between the sensing system and the scene being viewed.
  - Robotic applications
  - Autonomous navigation
  - Dynamic scene analysis



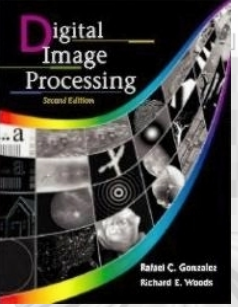
Basic approach

- A difference image between two images taken at times t_i and t_j may be defined as

$$d_{ij}(x,y) = \begin{cases} 1 & \text{if } \left| f(x,y,t_i) - f(x,y,t_j) \right| > T \\ 0 & \text{otherwise} \end{cases}$$

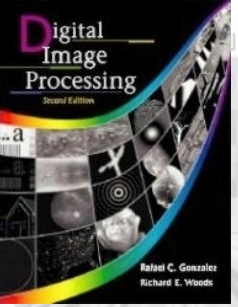
  T is a specified threshold.

- In dynamic image processing, all pixels in $d_{ij}(x,y)$ with value 1 are considered the result of object motion. The approach is applicable only if :
  - The images are registered spatially
  - The illumination is relatively constant within the bounds established by T
- All images in the sequence are of the same size, so as to $d_{ij}(x,y)$.



Accumulative difference image (ADI)

- Why use ADI?
- Isolated entries resulting from noise is not an insignificant problem when trying to extract motion components from a sequence of images. Although the number of these entries can be reduced by a thresholded connectivity analysis, this filtering process can also remove small or slow-moving objects. One way to address this problem is by considering changes at a pixel location over several frames, thus introducing a "memory" into the process. The idea is to ignore changes that occur only sporadically over a frame sequence and can therefore be attributed to random noise.



Accumulative difference image (cont'd)

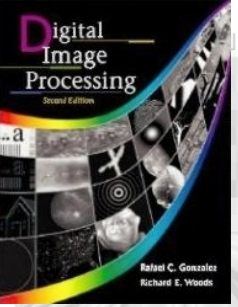
- Three types of ADI:

$$\begin{cases} \textit{absolute} \\ \textit{positive} \\ \textit{negetive} \end{cases}$$

$$A_k(x,y) = \begin{cases} A_{k-1}(x,y)+1 & \textit{if } \left|R(x,y) - f(x,y,k)\right| > T \\ A_{k-1}(x,y) & \textit{otherwise} \end{cases}$$

$$P_k(x,y) = \begin{cases} P_{k-1}(x,y)+1 & \textit{if } \left[R(x,y) - f(x,y,k)\right] > T \\ P_{k-1}(x,y) & \textit{otherwise} \end{cases}$$

$$N_k(x,y) = \begin{cases} N_{k-1}(x,y)+1 & \textit{if } \left[R(x,y) - f(x,y,k)\right] < -T \\ N_{k-1}(x,y) & \textit{otherwise} \end{cases}$$



Accumulative difference image (cont'd)

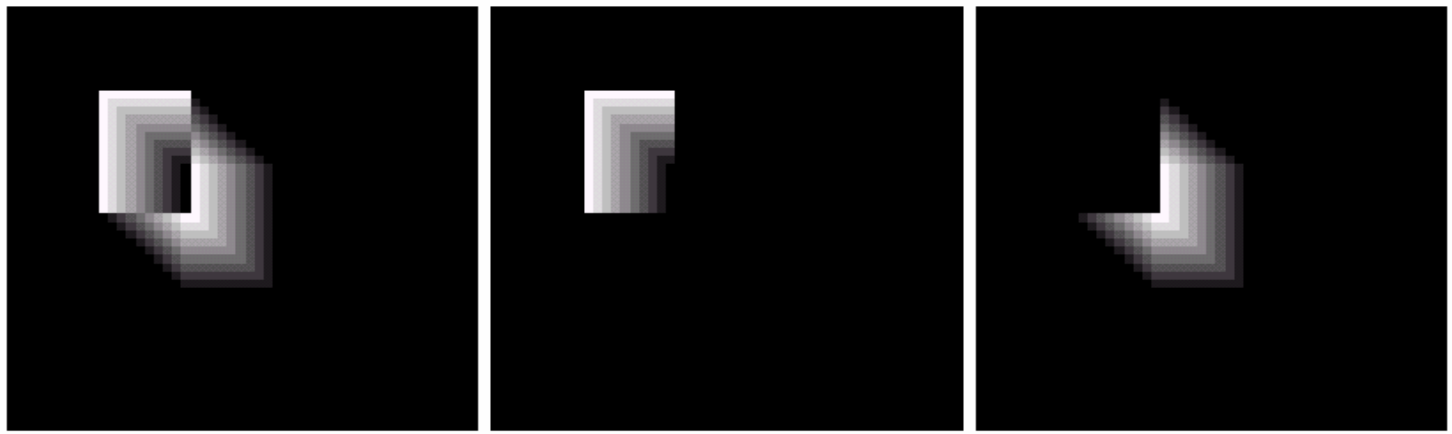
(1) The nonzero area of the positive ADI is equal to the size of the moving object.

(2) The location of the positive ADI corresponds to the location of the moving object in the reference frame.

(3) The number of counts in the positive ADI stops increasing when the moving object is displaced completely with respect to the same object in the reference frame.

(4) The absolute ADI contains the regions of the positive and negative ADI.

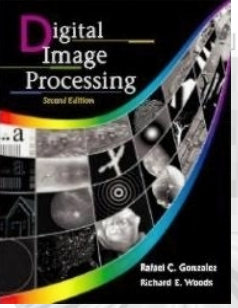
(5) The direction and speed of the moving object can be determined from the entries in the absolute and negative ADIs.



a b c

FIGURE 10.49 ADIs of a rectangular object moving in a southeasterly direction. (a) Absolute ADI. (b) Positive ADI. (c) Negative ADI.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



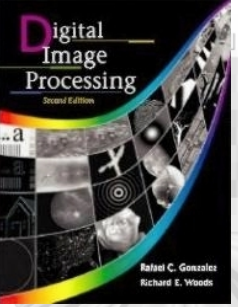
Establishing a reference image

- This necessity applies particularly to situations:
  - Describing busy scenes
  - In cases where frequent updating is required
- Object displacement can be established by monitoring the changes in the positive ADI.

a b c

FIGURE 10.50 Building a static reference image. (a) and (b) Two frames in a sequence. (c) Eastbound automobile subtracted from (a) and the background restored from the corresponding area in (b). (Jain and Jain.)

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.



Summary

- Image segmentation is an essential preliminary step in most automatic pictorial pattern recognition and scene analysis problems.
- The methods discussed in this chapter, although far from exhaustive, are representative of techniques commonly used in practice.