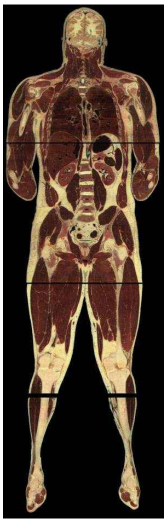
Medical Image Processing

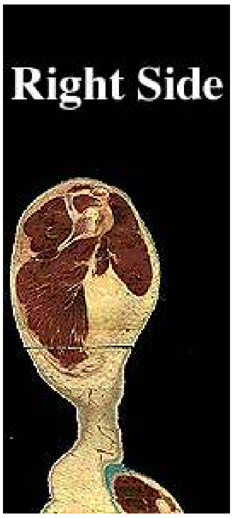
X. Image Segmentation

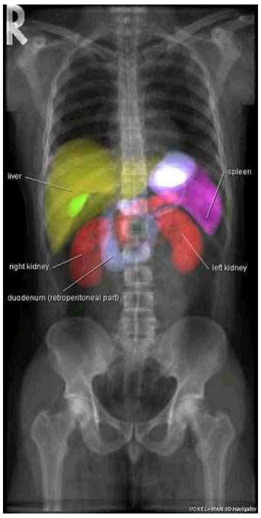
Medical image segmentation is generally difficult

- Noisy images
  - Often noise-to-signal is 10%
  - This is ten times N/S of camera images
- Often textured in complex ways
- Relatively poorly sampled
  - Many pixels contain more than one tissue type. (Partial Volume Effect)
- Objects of interest have complex shapes
- Signs of clinical interest are subtle

Visual Human Project







X. Image Segmentation

1. Preview
2. Point, Line and Edge Detection
3. Thresholding
4. Region-Based Segmentation
5. Segmentation Using Morphological Watersheds

6. Advanced Topics : Active Contours
   - Snake
   - Level Set

T

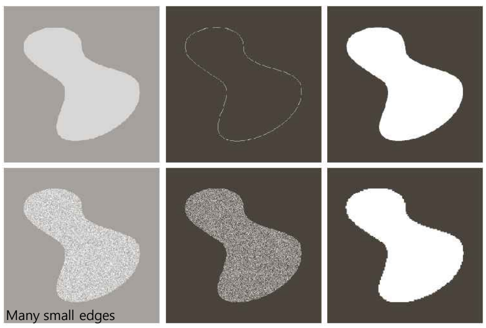
Preview

- Segmentation is to subdivide an image into its constituent regions or objects.
- Segmentation algorithms generally are based on one of two basis properties of intensity values
  - Discontinuity : to partition an image based on abrupt changes in intensity (edges)
  - Similarity : to partition an image into regions that are similar according to a set of predefined criteria. (thresholding, region growing, region splitting/merging)

Two Principal Approaches : Discontinuity and Similarity

Edge-based

Region-based

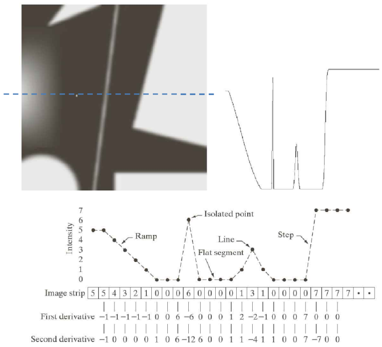


a b c
d e f

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.

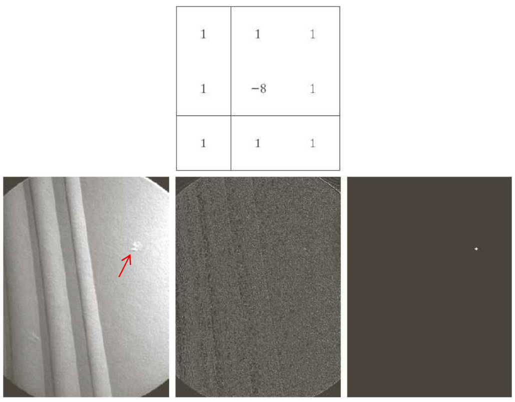
Detection of Discontinuities (Point, Line, and Edge)

- Detecting sharp, local changes in intensities
- Isolated points, lines, and edges.
- First and second-order derivatives
  - First-order (gradient operator)
    - produce thicker edges
  - Second-order (Laplacian operator)
    - Stronger response to fine detail
    - Double-edge response at ramp and step transitions
    - The sign determines whether a transition into an edge is from light to dark or dark to light.



Detection of Isolated Points

- Laplacian + thresholding

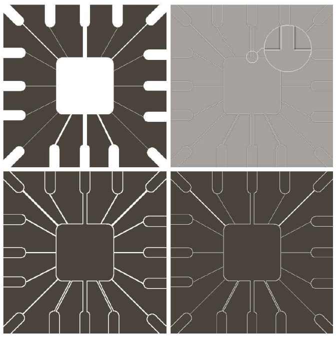


a
b c d

FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Line Detection

- Laplacian → positive and negative lines
  - Take absolute values → thick lines
  - Use only the positive values



a b
c d

FIGURE 10.5 (a) Original image. (b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian. (c) Absolute value of the Laplacian. (d) Positive values of the Laplacian.

Detection of lines in a specified direction

Horizontal

−1	−1	−1
2	2	2
−1	−1	−1

+45°

2	−1	−1
−1	2	−1
−1	−1	2

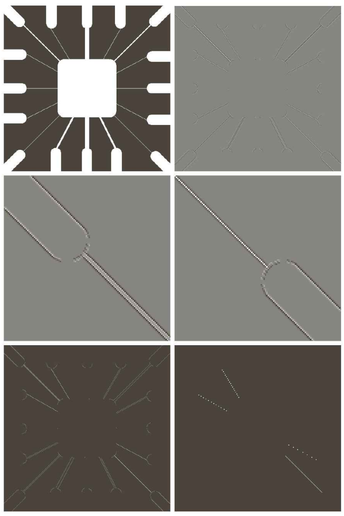
Vertical

−1	2	−1
−1	2	−1
−1	2	−1

−45°

−1	−1	2
−1	2	−1
2	−1	−1

Line detection masks

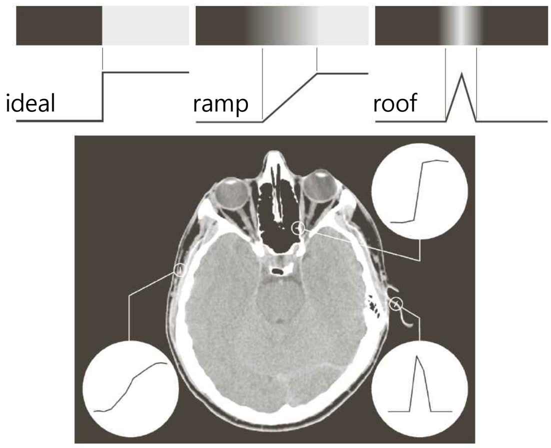


a b
c d
e f

FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the +45° line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

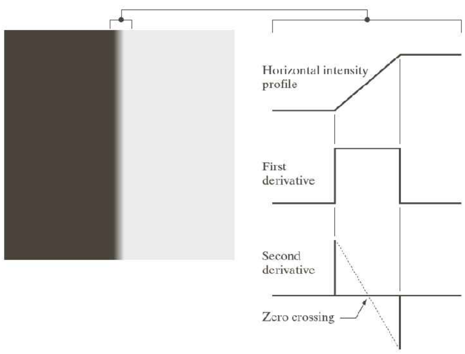
Edge Models

- In practice, edges are blurred and noisy. → ramp edge
- Ramp slope ← degree of blurring ← focusing mechanism (optics, sampling, image acquisition imperfection)



Derivatives and Ramp Edge

- The 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 edge
- The 2nd derivative produces two values → undesirable
- The zero crossings of the 2nd derivative can be used for locating the centers of thick edges



a b

FIGURE 10.10 (a) Two regions of constant intensity separated by an ideal vertical ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

Noise and Edge

- The 2nd derivative is more sensitive to noise.
- Fairly little noise can have a significant impact on the two derivatives.
- Smoothing should be serious consideration prior to the use of derivatives

- Edge detection steps
  - Smoothing for noise reduction
  - Detection of edge points
  - Selection of true edges

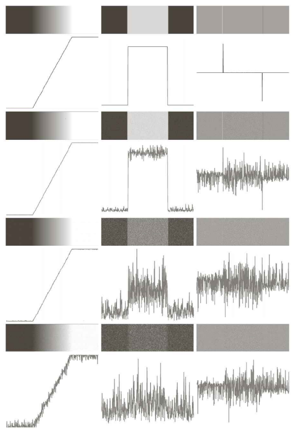


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

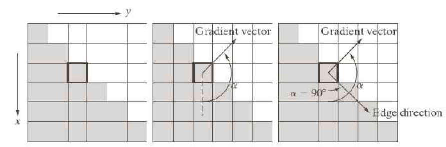
Basic Edge Detection
- The Image Gradient

$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

$$M(x,y) = \mathrm{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

The direction of an edge is perpendicular to $\alpha(x,y)$



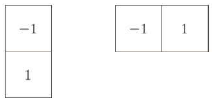
a b c

FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Gradient Operators

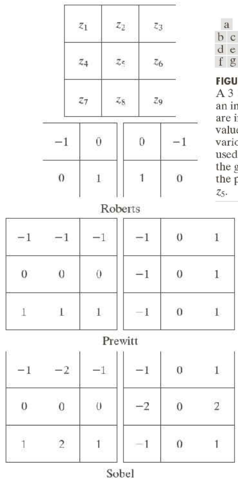
$$g_x = \frac{\partial f}{\partial x} = f(x+1, y) - f(x, y)$$

$$g_y = \frac{\partial f}{\partial y} = f(x, y+1) - f(x, y)$$



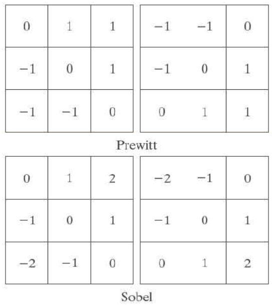
a b

FIGURE 10.13 One-dimensional masks used to implement Eqs. (10.2-12) and (10.2-13).



a
b c
d e
f g

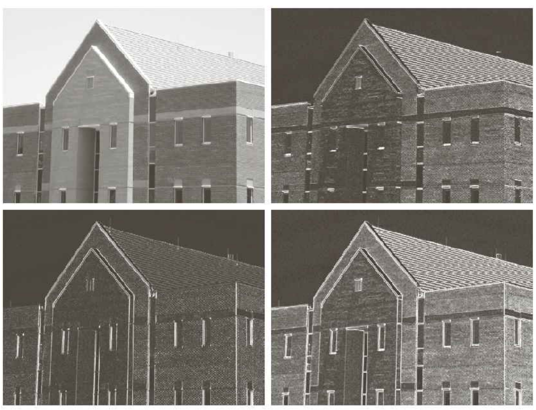
FIGURE 10.14 A 3 × 3 region of an image (the z's are intensity values) and various masks used to compute the gradient at the point labeled z_5.



a b
c d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

Example



a b
c d

FIGURE 10.16 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

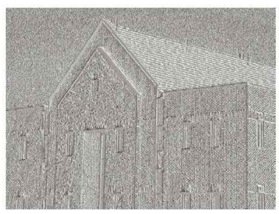
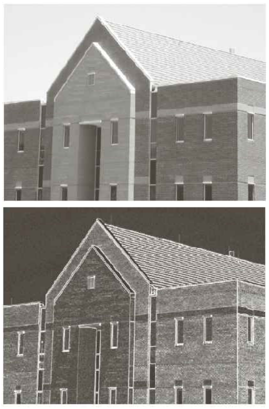
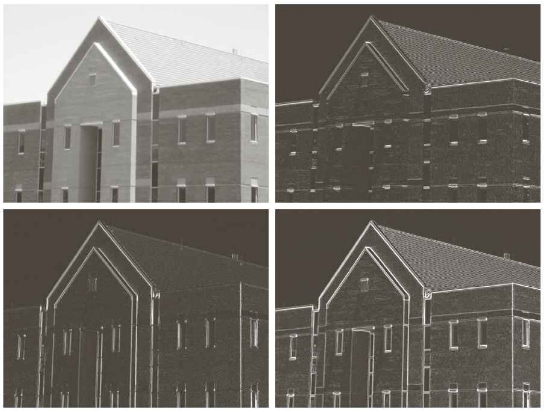


FIGURE 10.17 Gradient angle image computed using Eq. (10.2-11). Areas of constant intensity in this image indicate that the direction of the gradient vector is the same at all the pixel locations in those regions.

Example





a b
c d

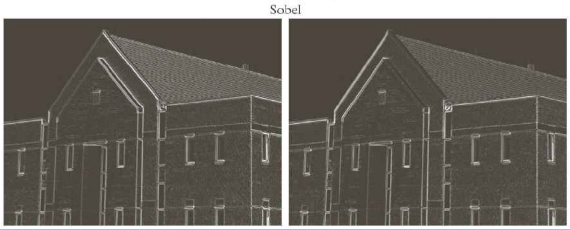
FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5 × 5 averaging filter prior to edge detection.

Example
- Diagonal Edge Detection

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel



a b

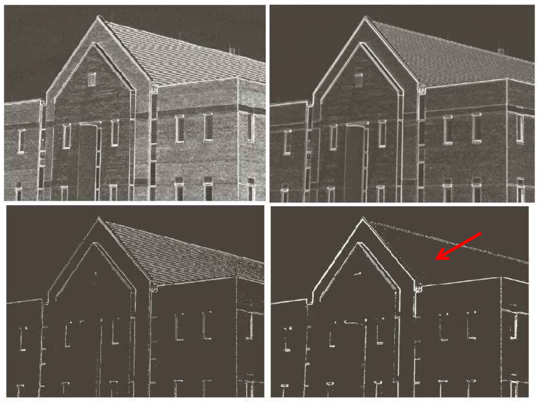
FIGURE 10.19 Diagonal edge detection. (a) Result of using the mask in Fig. 10.15(c). (b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

Combining the Gradient with Thresholding



a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

The Marr-Hildreth Edge Detector

- The intensity changes are not independent of image scale and so their detection requires the use of **operators of different sizes.**
- A sudden intensity change will give rise to a peak or trough in the 1^{st} derivative or, equivalently, to a **zero crossing** in the 2^{nd} derivative.
- An edge detector should be a differential operator capable of being "**tuned" to act at any desired scale**, so that large operators can be used to detect blurry edges and small operators to detect sharply focused fine detail.

LoG (Laplacian of a Gaussian)

- n×n Gaussian LPF+ 3×3 Laplacian :

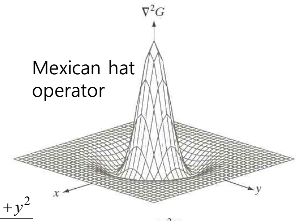
  Convolving the image with Gaussian smoothing function first and then computing the Laplacian of the result
- Find edges via **zero-crossing**.

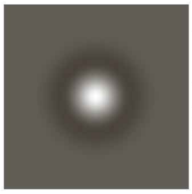
$$G(x,y)=e^{-\frac{x^2+y^2}{2\sigma^2}}$$

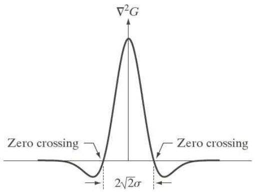
$$\nabla^2 G(x,y)=\frac{\partial^2 G}{\partial x^2}+\frac{\partial^2 G}{\partial y^2}$$

$$=\left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Mexican hat operator







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

the coefficients sum to zero

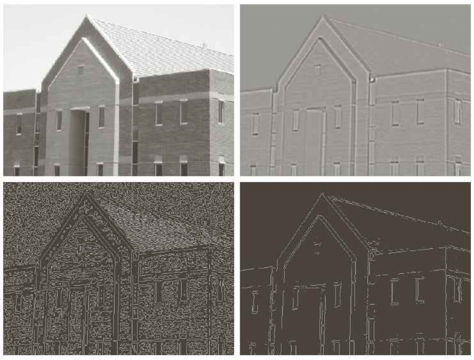
a b
c d

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.

The Marr-Hildreth Edge Detector

- The size of Gaussian filter
  - The smallest odd integer ≥ 6σ
- Zero-crossing
  - The signs of at least two of its opposing neighboring pixels must differ. (left/right, up/down, two diagonals)
  - Their difference ≥ threshold
- Attractive
  - Zero crossing produces thinner edges
  - Noise reduction
- Drawbacks
  - Zero-crossing creates closed loops. (spaghetti effect)
    → a positive threshold

Example



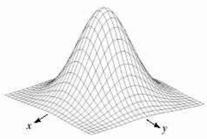
spaghetti effect

a b
c d

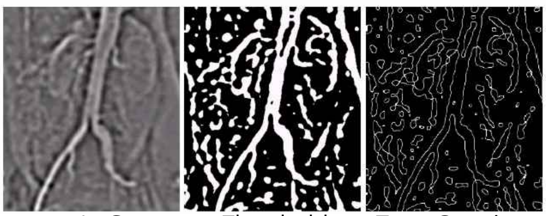
FIGURE 10.22 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found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

Example

Sobel



−1	−1	−1
−1	8	−1
−1	−1	−1



LoG Threshold Zero-Crossings

The Canny Edge Detector

- Objective
  - Low error rate : as close as possible to the true edges (less spurious responses)
  - Edge points should be well localized.
  - Single edge point response (only one point for each true edge point)

Canny Edge Detection Algorithm

1. Smooth the input image with a Gaussian filter
2. Compute the gradient magnitude M(x,y) and angle images α(x,y).
3. Apply nonmaxima suppression to M(x,y). (to thin M(x,y) which contains wide ridges around local maxima)
4. Use double thresholding and connectivity analysis to detect and link edges.

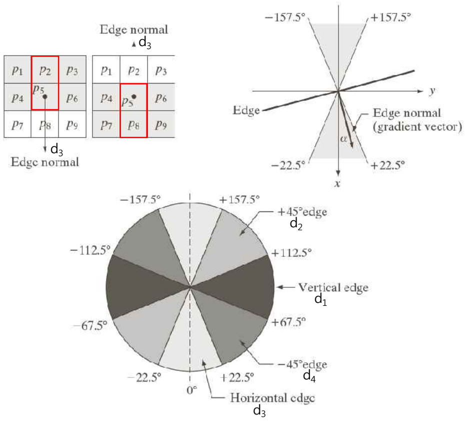

Nonmaxima Suppression

Step 1) Specify a number of discrete orientations of the edge normal (gradient vector).

- Ex) 4 orientations (0˚,45˚,90˚,-45˚) for a 3x3 region

Step 2) Get the nonmaxima-suppressed image $g_N(x,y)$

1) Find the direction d_k that is closest to $\alpha(x,y)$.
2) If $M(x,y)$ is less than at least one of its two neighbors along d_k, $g_N(x,y)=0$ (suppression); otherwise, $g_N(x,y)=M(x,y)$



a b
c

FIGURE 10.24 (a) Two possible orientations of a horizontal edge (in gray) in a 3 × 3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3 × 3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

Double Thresholding and Connectivity Analysis

Step 3) Threshold $g_N(x,y)$ to reduce false edge points.

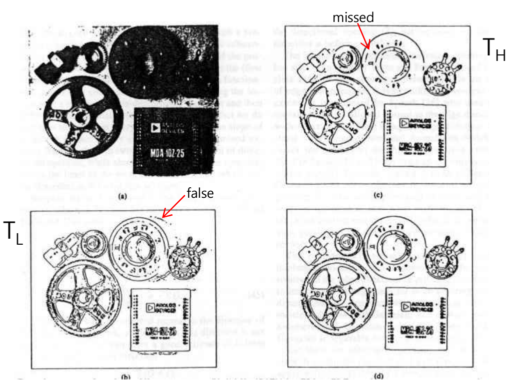
1) Hysteresis thresholding (low and high thresholds)

$g_{NH}(x,y)=g_N(x,y)\geq T_H$ → strong edges → valid edges with gaps

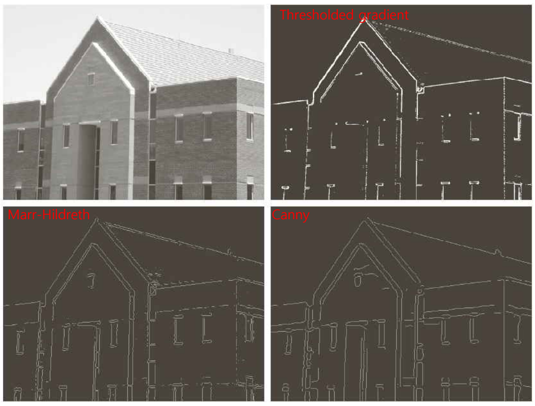
$g_{NL}(x,y)=g_N(x,y)\geq T_L$

$g_{NL}(x,y)=g_{NL}(x,y)- g_{NH}(x,y)$ → weak edges (including false edges)

2) Locate the next unvisited edge pixel, p in $g_{NH}(x,y)$
3) Mark as valid edge pixels all the weak pixels in $g_{NL}(x,y)$ that are connected to p, and repeat
4) Append marked pixels in $g_{NL}(x,y)$ to $g_{NH}(x,y)$



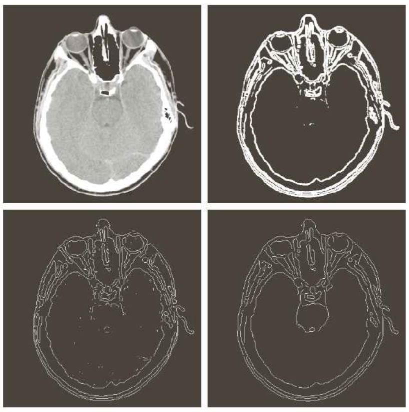
Example



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.

Example



a b
c d

FIGURE 10.26 (a) Original head CT image of size 512 × 512 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. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Edge Linking and Boundary Detection

- Obstacles to edge detection
  - Noise
  - Breaks in the edges due to nonuniform illumination
  - Other effects introducing spurious discontinuities in intensity values
- Require additional linking algorithms designed to assemble edge pixels into meaningful edges and/or region boundaries.
- Basic approaches
  - Local processing
  - Regional processing
  - Global processing via the Hough transform

Local Processing

- Analyze the characteristics of pixels in a small neighborhood (say, 3x3, 5x5) about every edge pixel.
- All points that are similar according to a set of predefined criteria are linked.
- Principal properties used for establishing similarity:

  (1) Magnitude of gradient vector, $M(x,y)=|\nabla f(x,y)|$

  $$|M(s,t)-M(x,y)| \le E$$

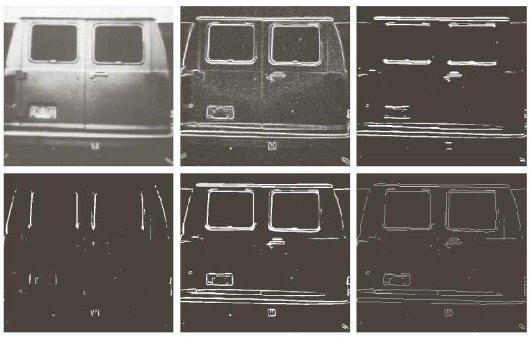
  (2) Direction of gradient vector, $\alpha(x,y)$

  $$|\alpha(s,t)-\alpha(x,y)| \le A$$
- Edge pixel (x,y) is linked with (s,t) if both criteria are satisfied.

Simplified Formulation

1. Compute $M(x,y)$ and $\alpha(x,y)$
2. $g(x,y) =$ 1 if $M(x,y) > T_M$ AND $|\alpha(x,y) - A| < T_A$
   0 otherwise
   A : a specified direction
3. Scan the rows of $g(x,y)$ and fill all gaps in each row that do not exceed a specified length
4. To detect gaps in any direction, rotate $g(x,y)$ by this angle and apply step 3.

Example

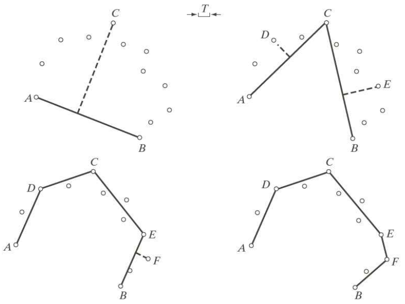


a b c
d e f

FIGURE 10.27 (a) A 534 × 566 image of the rear of a vehicle. (b) Gradient magnitude image. (c) Horizontally connected edge pixels. (d) Vertically connected edge pixels. (e) The logical OR of the two preceding images. (f) Final result obtained using morphological thinning. (Original image courtesy of Perceptics Corporation.)

Regional Processing

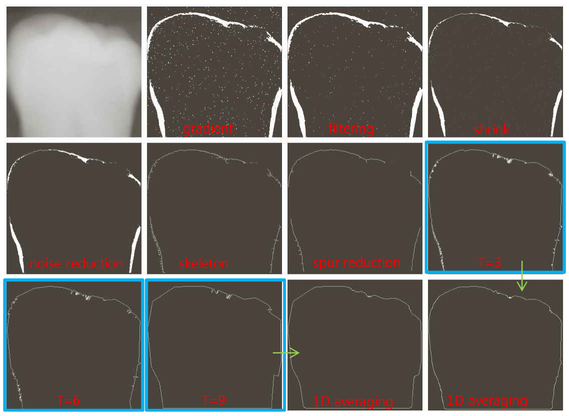
- When knowledge is available regarding the regional membership of pixels
- Polygonal curve fitting



a b
c d

FIGURE 10.28 Illustration of the iterative polygonal fit algorithm.

Example



1D smoothing for jaggedness reduction

a b c d
e f g h
i j k l

FIGURE 10.30 (a) A 550 × 566 X-ray image of a human tooth. (b) Gradient image. (c) Result of majority filtering. (d) Result of morphological shrinking. (e) Result of morphological cleaning. (f) Skeleton. (g) Spur reduction. (h)–(j) Polygonal fit using thresholds of approximately 0.5%, 1%, and 2% of image width (T = 3, 6, and 12). (k) Boundary in (j) smoothed with a 1-D averaging filter of size 1 × 31 (approximately 5% of image width). (l) Boundary in (h) smoothed with the same filter.

Global processing using the Hough transform

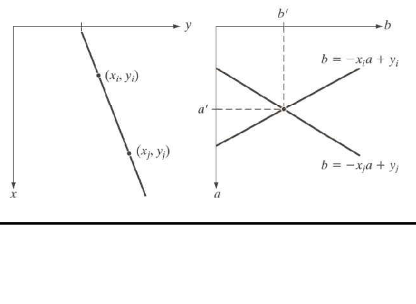
- Situation
  - Unlike regional processing cases, all we have is an edge image and no knowledge about where objects of interest might be.
  - All pixels are candidates for linking and thus have to be accepted or eliminated based on predefined global properties.

- An approach based on whether sets of pixels lie on curves of a specified shape.
- Once detected, these curves form the edges or region boundaries of interest.

The Hough Transform

- Brute force : n edge pixels
  - Number of possible lines = n(n-1)/2
  - Distance per line = n
  - Total number of distances = $n^2(n-2)/2 \approx n^3$ → a computationally prohibitive task

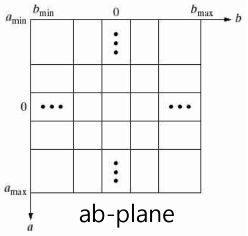
- The Hough transform : Efficient algorithm to find straight lines
  - When different values for a and b are considered, $y_i=ax_i+b$ gives all possible lines through the point (x_i,y_i). The equation $b=-x_ia+y_i$ gives one line in the ab-plane for a specific (x_i,y_i)
  - ab-plane : parametric space yielding the equation of a single line
  - When another point (x_j,y_j) is considered, $b=-x_ja+y_j$ represents another line in the ab-plane.
  - Suppose that these two lines intersect at the point (a', b'), then $y=a'x+b'$ represents the line in the xy-plane on which both (x_i,y_i) and (x_j,y_j) lie



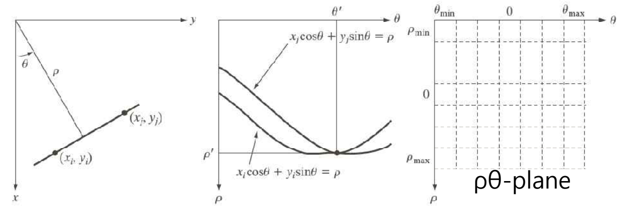
a b
FIGURE 10.31
(a) *xy*-plane.
(b) Parameter space.

Algorithm

1) Set all cells of ab-plane to zero
2) For every (x_k, y_k)
   2.1) Let a=every subdivision on the a-axis
   2.2) Calculate $b=-x_k a+y_k$
   2.3) Round off b to the nearest allotted value on the b-axis
   2.4) Increment accumulator cell (a, b) with 1



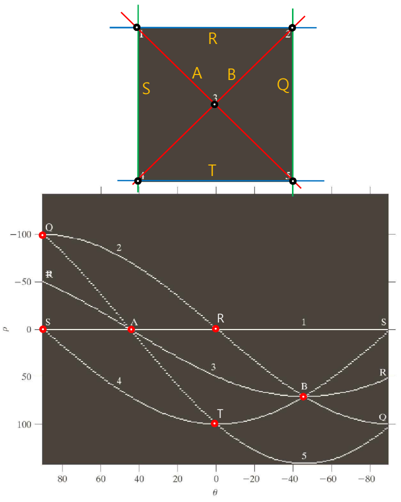
- Problem : $-\infty<a<\infty$, $-\infty<b<\infty$
  → solution : (a,b) → (ρ,θ), $x\cos\theta+y\sin\theta=\rho$,
  $-90^\circ\leq\theta\leq90^\circ$, $-\sqrt{2}D\leq\rho\leq\sqrt{2}D$ (D is the image size)



a b c

FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.

Example



a
b

FIGURE 10.33
(a) Image of size 101 × 101 pixels, containing five points.
(b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)

Generalized Hough Transform

- can be used for any function of the form

$$g(v,c) = 0$$

  - v is a vector of coordinates
  - c is a vector of coefficients
- Ex) Circle
  - equation: $(x-c_1)^2 + (y-c_2)^2 = c_3^2$
  - three parameters (c_1, c_2, c_3) : cube like cells
  - accumulators of the form A(i, j, k)
  - increment c_1 and c_2 , solve of c_3 that satisfies the equation
  - update the accumulator corresponding to the cell associated with triplet (c_1, c_2, c_3)

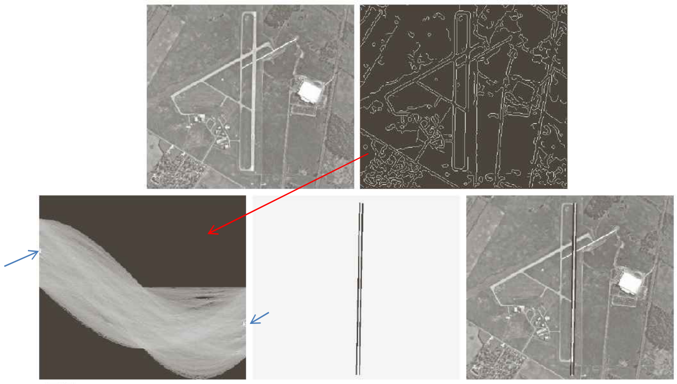
Algorithm for edge linking

1) Compute the gradient of an image and threshold it to obtain a binary image.
2) Specify subdivisions in the $\rho\theta$-plane.
3) Examine the counts of the accumulator cells for high pixel concentrations.
4) Examine the relationship (principally for continuity) between pixels in a chosen cell.
5) Link these pixels if gaps are smaller than threshold

- Require only that we examine pixels associated with specific accumulator cells → a significant advantage of global processing

Example

- Extract the two edges of the principal runway for autonomous navigation of air vehicles

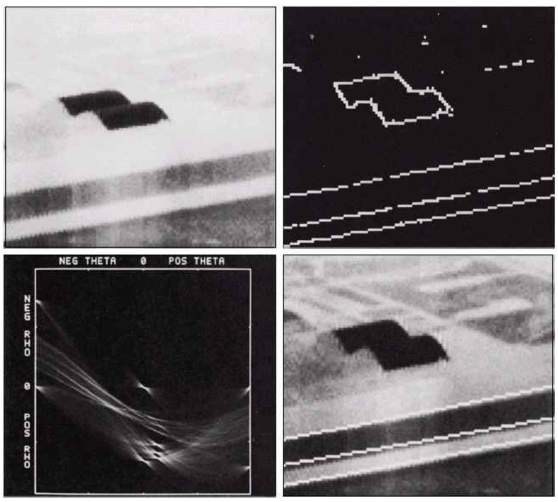


a b
c d e

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.

Example

link criteria:
- the pixels belong to one of the set of pixels linked according to the highest count
- no gaps are longer than 5 pixels



a b
c d

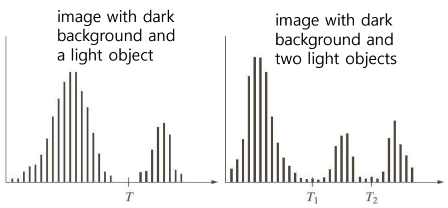
FIGURE 10.21
(a) Infrared image.
(b) Thresholded gradient image.
(c) Hough transform.
(d) Linked pixels.
(Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

Thresholding

- g(x,y)= 1 if f(x,y)>T
  0 if f(x,y)≤T

- Global thresholding
- Variable (regional) thresholding, dynamic (adaptive) thresholding
  - T depends on f(x,y) and p(x,y) (on gray-level values and its neighbors)
- Multiple thresholding

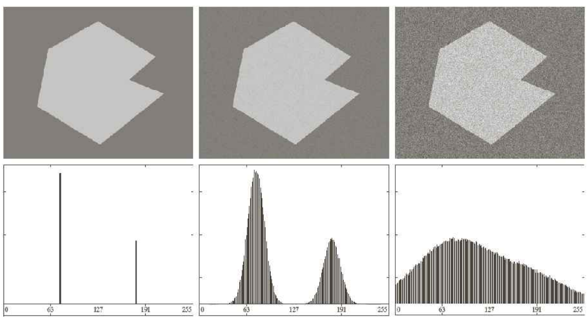
- Key factors
  - the separation between peaks in the histogram
  - the noise content
  - the relative size of objects and background
  - the uniformity of the illumination source
  - the uniformity of the reflectance properties



a b

FIGURE 10.35 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

The Role of Noise

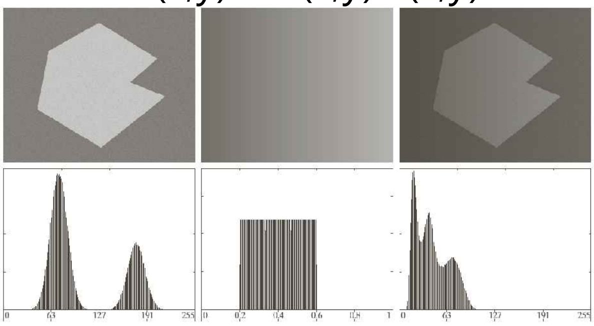


a b c
d e f

FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

The Role of Illumination and Reflectance

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



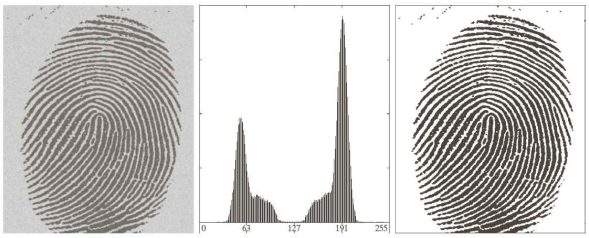
a b c
d e f

FIGURE 10.37 (a) Noisy image. (b) Intensity ramp in the range [0.2, 0.6]. (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

Basic Global Thresholding

- Estimate automatically the threshold value T
  1) Select an initial estimate for T. (ex: average)
  2) Segment the image into regions G_1 and G_2 using T.
  3) Compute the average intensity values m_1 and m_2 for the pixels in G_1 and G_2
  4) Compute a new threshold value :
  $$T=(m_1+m_2)/2$$
  5) Repeat Steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter ΔT.

Example



a b c

FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

$T_{init}=m_G$
$\Delta T=0$
3 iterations with result $T=125$

Otsu's Method for Optimum Global Thresholding

- Objective : Minimize the average error incurred in assigning pixels to two groups (classes)
- Optimum : Maximize the between-class variance
- Basic idea :
  - Well-thresholded classes should be distinct with respect to the intensity values of their pixels
  - The best (optimum) threshold gives the best separation between classes
- Based entirely on the histogram

Otsu's Algorithm

- Normalized histogram $p_i = n_i / MN$ $\quad (\sum_{i=0}^{L-1} p_i = 1)$
- Threshold the image into C_1 and C_2 using $T=k$
- The probability that a pixel is assigned to C_1, $P_1(k) = \sum_{i=0}^{k} p_i$
- The probability of C_2, $P_2(k)=1- P_1(k)$
- Means of C_1 and C_2 :

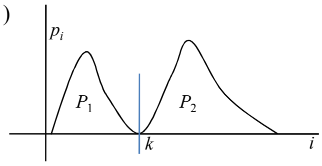
$$m_1(k) = \frac{1}{P_1(k)} \sum_{i=0}^{k} i p_i, \quad m_2(k) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} i p_i \qquad P_1 m_1 + P_2 m_2 = m_G, \quad P_1 + P_2 = 1$$

- Separability measure (goodness of threshold), $\eta = \sigma_B^2(k)/\sigma_G^2$

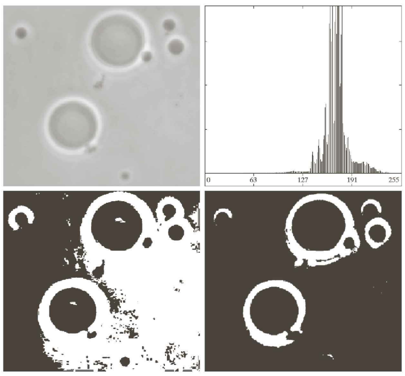
between-class variance $\sigma_B^2(k) = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 = P_1 P_2 (m_1 - m_2)^2$

$$= \frac{(m_G P_1 - m)^2}{P_1(1-P_1)} \qquad (m = \sum_{i=0}^{k} i p_i)$$

- Determine k to maximize $\sigma_B^2(k)$



Example

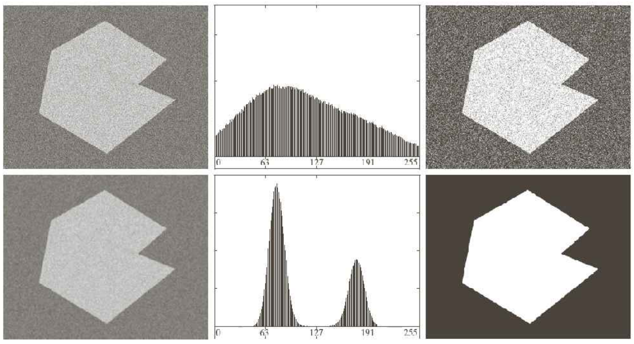


a b
c d

FIGURE 10.39 (a) Original image. (b) Histogram (high peaks were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

k=169 $\qquad$ k=181, η=0.467

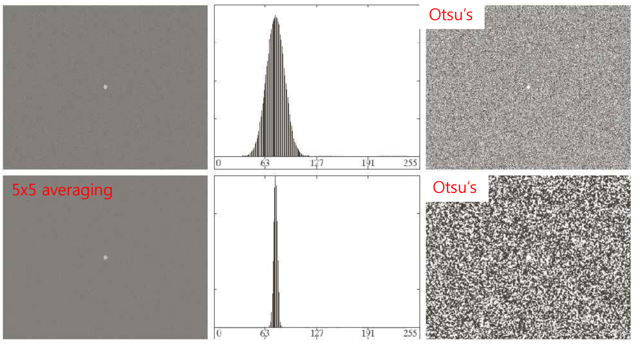
Using Image Smoothing to Improve Global Thresholding



a b c
d e f

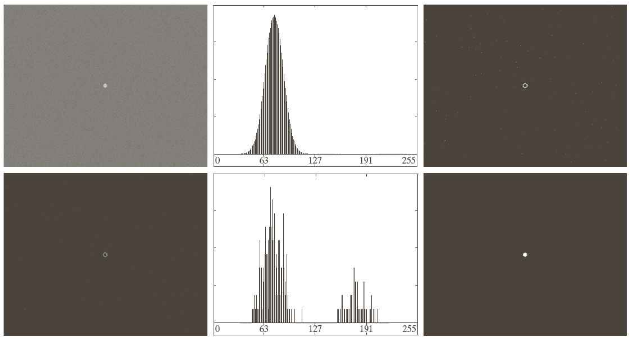
FIGURE 10.40 (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a 5 × 5 averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

Using Edges to Improve Global Thresholding



1) Compute an edge image, g(x,y) from f(x,y)
2) Threshold the edge image using a specifed threshold value, T to make a mask image, $g_T(x,y)$
3) Compute a histogram of f(x,y) corresponding to $g_T(x,y)$
4) Use the histogram to segment f(x,y) globally

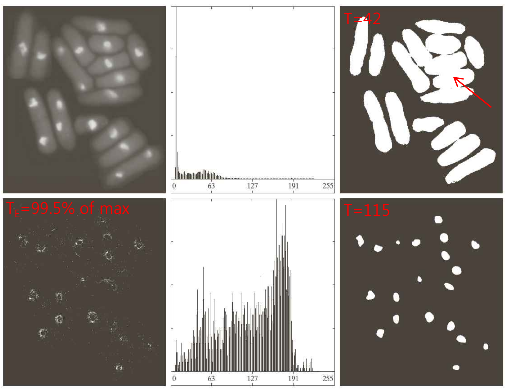
Example



a b c
d e f

FIGURE 10.42 (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based on the histogram in (e). The threshold was 134, which is approximately midway between the peaks in this histogram.

Example



a b c
d e f

FIGURE 10.43 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

T_E=5% of max

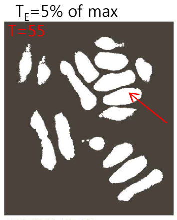
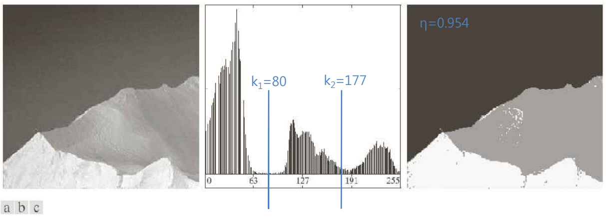


FIGURE 10.44 Image in Fig. 10.43(a) segmented using the same procedure as explained in Figs. 10.43(d)–(f), but using a lower value to threshold the absolute Laplacian image.

Multiple Thresholds

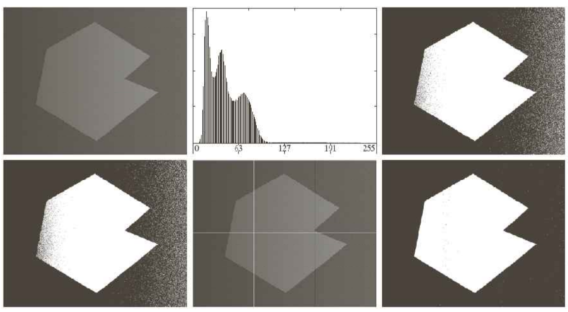
- Extended Otsu's method
- (K-1) threshold values for K classes: $C_1, C_2, C_3, \ldots, C_K$
- Between-class variance $\sigma_B^2 = \sum_{k=1}^{K} P_k (m_k - m_G)^2 \qquad P_k = \sum_{i \in C_k} p_i$



a b c

FIGURE 10.45 (a) Image of iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

Variable Thresholding Using Image Partioning



a b c
d e f

FIGURE 10.46 (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.

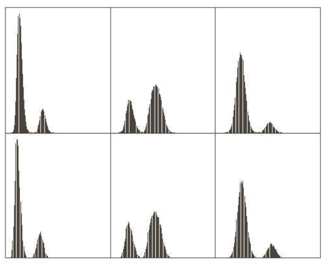


FIGURE 10.47 Histograms of the six subimages in Fig. 10.46(e).

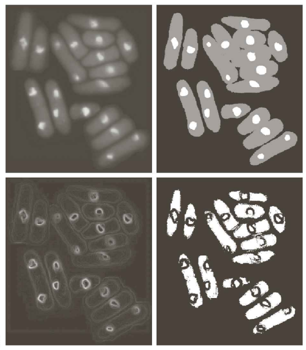
Variable Thresholding
Based on Local Image Properties

- Local threshold, $T_{xy}=a\sigma_{xy}+bm_{xy}$ in a neighborhood, S_{xy} centered at (x,y)
  or $T_{xy}=a\sigma_{xy}+bm_G$

- g(x,y)=1 if the predicate, Q(local parameters) is true
  0 if Q is false
  ex 1) $Q(\sigma_{xy}, m_{xy})$=true if $f(x,y)>a\sigma_{xy}$ AND $f(x,y)>bm_{xy}$
  false otherwise
  ex 2) $Q(\sigma_{xy}, m_{xy})$=true if $f(x,y)>T_{xy}$
  false otherwise

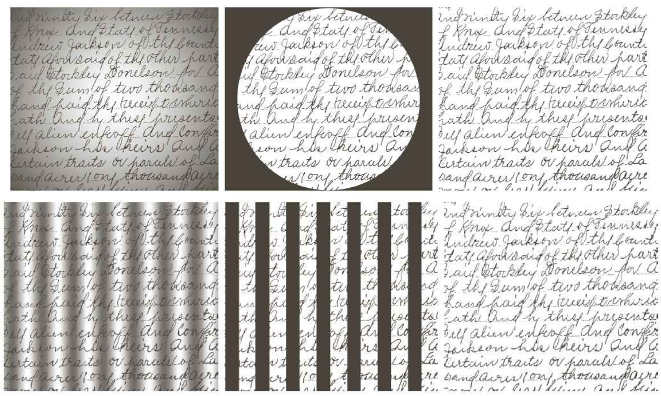
a b
c d

FIGURE 10.48
(a) Image from Fig. 10.43. (b) Image segmented using the dual thresholding approach discussed in Section 10.3.6. (c) Image of local standard deviations. (d) Result obtained using local thresholding.



Variable Thresholding
Based on Moving Averaging

- Moving average along scan lines, m_{xy}
- $T_{xy}=bm_{xy}$

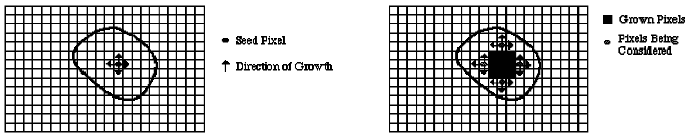


Otsu's

Moving average
n=20, b=0.5

Region Based Segmentation

- Region Growing
  - Procedure grouping pixels or subregions into larger regions based on predefined criteria for growth.
  - Start with a set of “seed” points
  - Grow regions by appending to each seed those neighbors that have predefined properties similar to the seed (such as specific ranges of intensity or color)
- Region growing should stop when no more pixels satisfy the criteria for inclusion in that region
- A basic algorithm is based on 8-connectivity



(a) Start of Growing a Region

(b) Growing Process After a Few Iterations

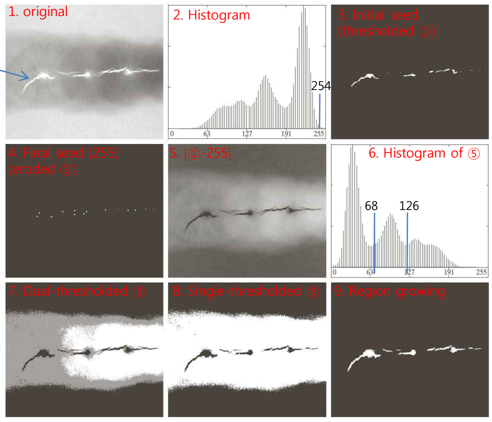
Region Growing Algorithm

- $f(x,y)$: input image, $S(x,y)$: seed image

1) Find all connected components in $S(x,y)$ and erode to one pixel; Label such pixels
2) Form an image f_Q such that, at a pair of coordinates (x,y), let $f_Q(x,y)=1$ if the given predicate, Q=TRUE; otherwise let $f_Q(x,y)=0$
3) Let g be an image formed by appending to each seed point all the 1-valued points in f_Q that are 8-connected to that seed point
4) Label each connected component in g with a different region label (e.g., 1,2,3,…)

The mechanics of the Algorithm

Defective welds



The problem of segmenting the defects cannot be solved using dual thresholds even though the thresholds are in the main valleys.

a b c
d e f
g h i

FIGURE 10.51 (a) X-ray image of a defective weld. (b) Histogram. (c) Initial seed image. (d) Final seed image (the points were enlarged for clarity). (e) Absolute value of the difference between (a) and (c). (f) Histogram of (c). (g) Difference image thresholded using dual thresholds. (h) Difference image thresholded with the smallest of the dual thresholds. (i) Segmentation result obtained by region growing. (Original image courtesy of X-TEK Systems, Ltd.)

Using *intensity* as a measure of similarity

Q=TRUE if diff ≤T

Advantages and Disadvantages

- Advantages :
  1. Region growing methods can correctly separate the regions that have the same properties we define
  2. Region growing methods can provide the original images which have clear edges the good segmentation results.
  3. The concept is simple. We only need a small numbers of seed point to represent the property we want, then grow the region.
  4. We can determine the seed points and the criteria we want to make.
  5. We can choose the multiple criteria at the same time.
  6. It performs well with respect to noise.

- Disadvantage :
  1. The computation is consuming.
  2. Noise or variation of intensity may result in holes or over-segmentation.
  3. This method may not distinguish the shading of the real images.

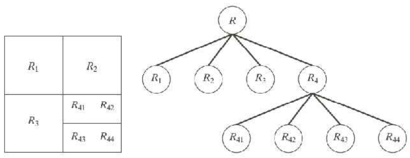
Region Splitting and Merging

- Subdivide an image initially into a set of arbitrary, disjoint regions and then merge and/or split the regions in an attempt to satisfy the necessary conditions
- R: entire image region, Q: predicate

  (1) Split into four disjoint quadrants any region R_i for which $Q(R_i)$ = FALSE

  (2) Merge any adjacent regions R_j and R_k for which $Q(R_j \cup R_k)$ = TRUE

  (3) Stop when no further merging is possible



a b

FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. R represents the entire image region.

Region Splitting and Merging Algorithm

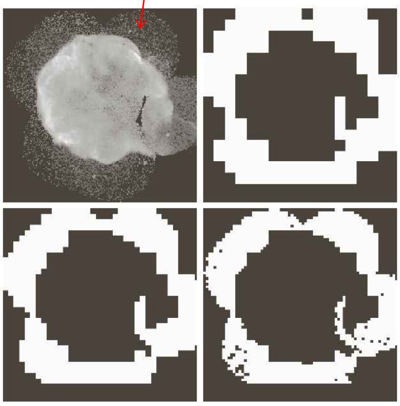
- Initially take the image as a whole to be the area of interest.
- Look at the area of interest and decide if all pixels contained in the region satisfy some similarity constraint.
- If TRUE, then the area of interest corresponds to a region in the image.
- If FALSE, split the area of interest (usually into four equal sub-areas) and consider each of the sub-areas as the area of interest in turn.

- This process continues until no further splitting occurs. In the worst case this happens when the areas are just one pixel in size.
- This is a *divide and conquer* or *top down* method.

Example

Segment out of the image the "ring" of less dense matter surrounding the dense center

If σ>a AND 0<m<b, then Q= TRUE



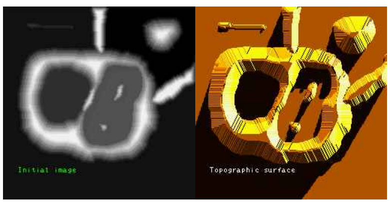
a b
c d

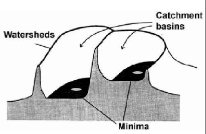
FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

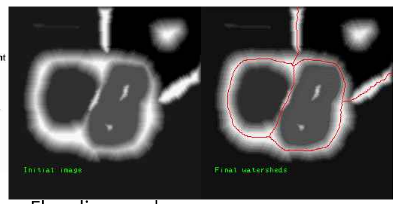
Morphological Watersheds

- To extract nearly uniform objects from the background.
- A closed contour
- Independent of shape and size
- Efficient and accurate

- Any gray-scale image can be considered as a *topographic* surface.
- Often applied to the *gradient* of an image rather than to the image itself
- If we *flood* this surface from its *minima* and, if we prevent the merging of the waters coming from different sources, we partition the image into two different sets: the *catchment basins* and the *watershed lines*.
- When the rising water in distinct basins is about to merge, a *dam* is built to prevent the merging.



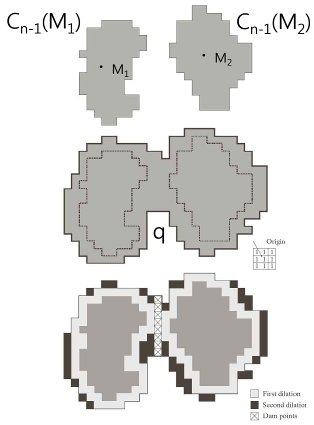




Flooding and building dams

Dam Construction

Dam construction is based on binary images



Two partially flooded catchment basins at stage n-1

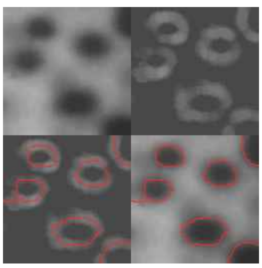
Flooding at stage n, showing that water has spilled between basins and merged; The two connected components have become a single component q.

Result of dilation and dam construction (Dilation is constrained to q)

Building a dam = Setting points of the dam to a value greater than the max

Gradient of The Image

- If we apply this transformation to the image gradient, the catchment basins should theoretically correspond to the homogeneous grey level regions of this image.

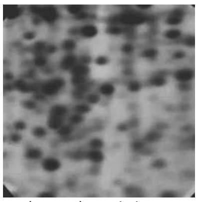


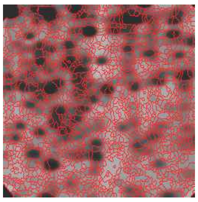
a b
c d

FIGURE 10.56 (a) Image of blobs. (b) Image gradient. (c) Watershed lines. (d) Watershed lines superimposed on original image. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

The Use of Markers

- Over-segmentation problem due to noise and other local irregularities of the gradient
- Limit the number of allowable regions by incorporating a preprocessing stage designed to bring additional knowledge into the segmentation procedure
- **Internal markers** for objects of interest and **external markers** for the background
- Procedure for marker selection
  1) Preprocessing (smoothing) to minimize the effect of small spatial details
  2) Marker criteria
     - A region that is surrounded by points of higher altitude
     - The points in the region form a connected component
     - All the points in the region have the same intensity value
  3) Apply the watershed algorithm to the smoothed image (internal markers are the only allowed regional minima)
  4) The resulting watershed lines are defined as the external markers and effectively partition the image into regions. Each region contains a single internal marker and part of the background.
  5)* Apply the watershed algorithm to each individual region of the gradient of the smoothed image


Electrophoresis image


Result of watershed segmentation (over-segmentation)

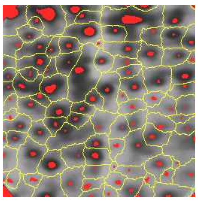
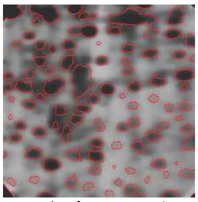
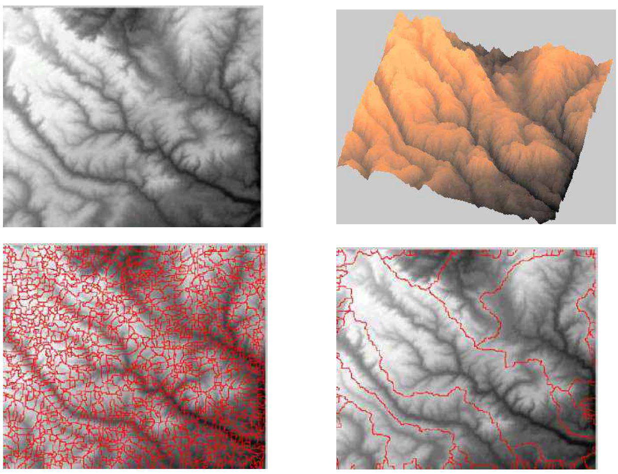


Image showing internal markers (red regions) and external markers (yellow lines)


Result of segmentation

Example



Example

In this case, the criterion used is not the intensity values
but the distance transform of the image