بینایی کامپیوتری

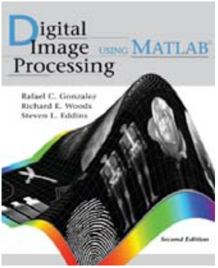
درس ۹ب

بخش‌بندی تصویر در متلب

Image Segmentation in MATLAB®

کاظم فولادی
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

http://courses.fouladi.ir/vision

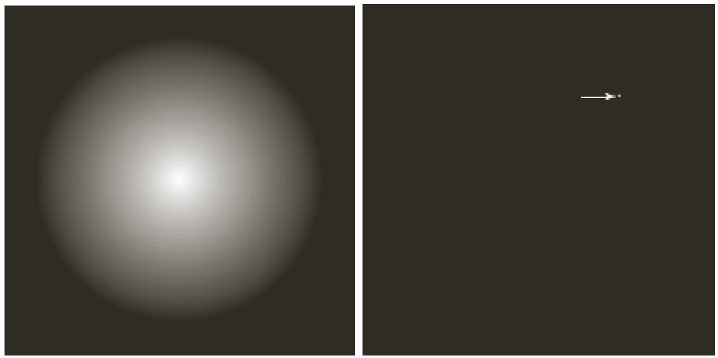


Chapter 10
Image Segmentation

−1	−1	−1
−1	8	−1
−1	−1	−1

FIGURE 10.1 A mask for point detection.

Chapter 10
Image Segmentation



a b

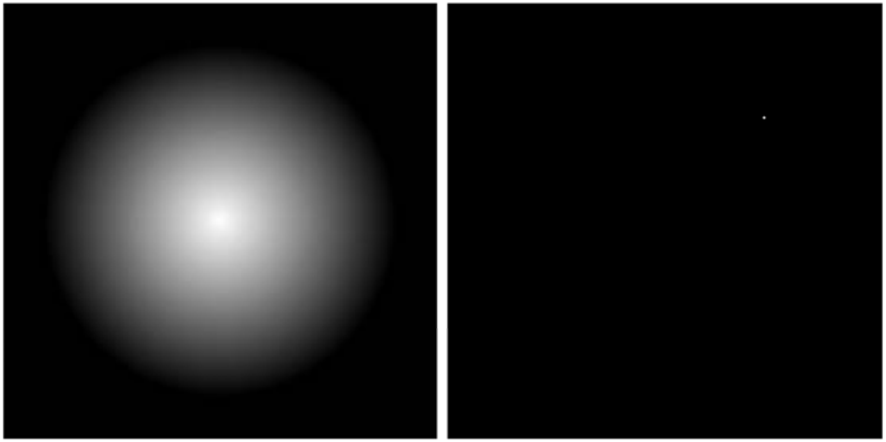
FIGURE 10.2 (a) Gray-scale image with a nearly invisible isolated black point in the northeast quadrant of the sphere. (b) Image showing the detected point. (The point was enlarged to make it easier to see.)

آشکارسازی نقطه

ماسک برای آشکارسازی نقطه

−1	−1	−1
−1	8	−1
−1	−1	−1

A mask for point detection.



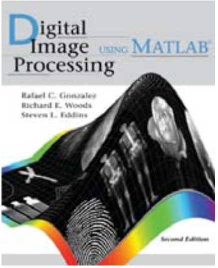
a b

(a) Gray-scale image with a nearly invisible isolated black point in the dark gray area of the northeast quadrant. (b) Image showing the detected point. (The point was enlarged to make it easier to see.)

f

```
w = [-1 -1 -1; -1 8 -1; -1 -1 -1];
g = abs(imfilter(tofloat(f),w));
T = max(g(:));
g = (g >= T);
imshow(g)
```

MATLAB



Chapter 10
Image Segmentation

a b c d

FIGURE 10.3 Line detector masks.

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

آشکارسازی خط

ماسک‌های آشکارساز خط

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

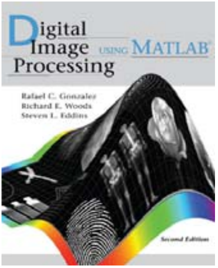
$+45°$

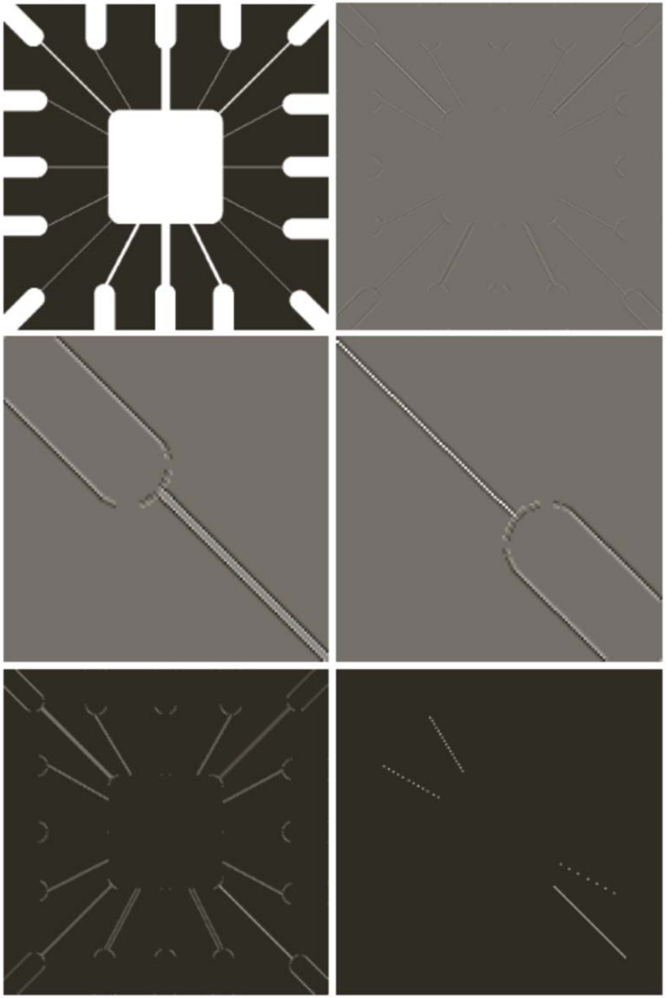
−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

$-45°$



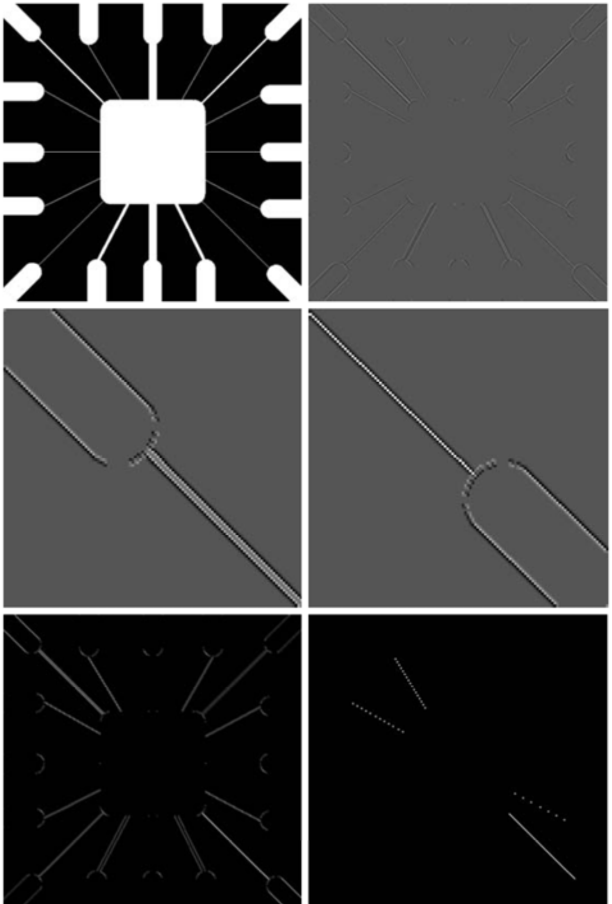


a b
c d
e f

FIGURE 10.4
(a) Image of a wire-bond template.
(b) Result of processing with the +45° detector in Fig. 10.3.
(c) Zoomed view of the top, left region of (b).
(d) Zoomed view of the bottom, right section of (b).
(e) Absolute value of (b).
(f) All points (in white) whose values satisfied the condition `g >= T`, where `g` is the image in (e). (The points in (f) were enlarged to make them easier to see.)

آشکارسازی خط

f

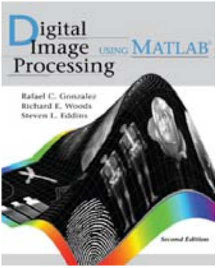


```
w = [2 -1 -1; -1 2-1; -1 -1 2];
g = imfilter(tofloat(f), w);
imshow(g, [ ])
% Fig.(b)
gtop = 9(1:120, 1:120);
% Top, left section.
gtop = pixeldup(gtop, 4);
% Enlarge by pixel duplication.
figure, imshow(gtop, [ ])
% Fig.(c)
gbot = g(end-119:end,end-119:end);
gbot = pixeldup(gbot, 4);
figure, imshow(gbot, [ ])
% Fig.(d)
g = abs(g);
figure, imshow(g, [ ])
% Fig(e)
T = max(g(:));
g = g >= T;
figure, imshow(g)
% Fig.(f)
```

a	b
c	d
e	f

(a) Image of a wire-bond mask. (b) Result of processing with the −45° detector in Fig. 10.3. (c) Zoomed view of the top, left region of (b). (d) Zoomed view of the bottom, right section of (b). (e) Absolute value of (b). (f) All points (in white) whose values satisfied the condition g >= T, where g is the image in (e). (The points in (f) were enlarged slightly to make them easier to see.)

MATLAB



Chapter 10
Image Segmentation

Edge Detector	Description
Sobel	Finds edges using the Sobel approximation to the derivatives in Fig. 10.5(b)
Prewitt	Finds edges using the Prewitt approximation to the derivatives in Fig. 10.5(c).
Roberts	Finds edges using the Roberts approximation to the derivatives in Fig. 10.5(d).
Laplacian of a Gaussian (LoG)	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a Laplacian of a Gaussian filter.
Zero crossings	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a specified filter.
Canny	Finds edges by looking for local maxima of the gradient of $f(x, y)$. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Therefore, this method is more likely to detect true weak edges.

TABLE 10.1 Edge detectors available in function `edge`.

آشکارسازی لبه

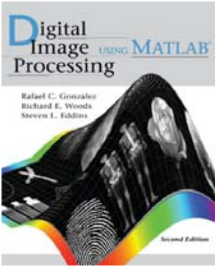
آشکارسازهای لبه

EDGE DETECTION

```
[g,t] = edge(f, method, papremeters)
```

Edge Detector	Basic Properties
Sobel	Finds edges using the Sobel approximation to the derivatives shown in Fig. 10.5(b).
Prewitt	Finds edges using the Prewitt approximation to the derivatives shown in Fig. 10.5(c).
Roberts	Finds edges using the Roberts approximation to the derivatives shown in Fig. 10.5(d).
Laplacian of a Gaussian (LoG)	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a Gaussian filter.
Zero crossings	Finds edges by looking for zero crossings after filtering $f(x, y)$ with a user-specified filter.
Canny	Finds edges by looking for local maxima of the gradient of $f(x, y)$. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. Therefore, this method is more likely to detect true weak edges.

Edge detectors available in function `edge`.



Chapter 10
Image Segmentation

a
b
c
d

FIGURE 10.5 Edge detector masks and the first-order derivatives they implement.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image neighborhood

−1	−2	−1
0	0	0
1	2	1

$g_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$

−1	0	1
−2	0	2
−1	0	1

$g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$

Sobel

−1	−1	−1
0	0	0
1	1	1

$g_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$

−1	0	1
−1	0	1
−1	0	1

$g_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$

Prewitt

−1	0
0	1

$g_x = z_9 - z_5$

0	−1
1	0

$g_y = z_8 - z_6$

Roberts

آشکارسازی لبه

ماسک‌های آشکارساز لبه

Image neighborhood

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sobel

−1	−2	−1
0	0	0
1	2	1

$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$

−1	0	1
−2	0	2
−1	0	1

$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$

Prewitt

−1	−1	−1
0	0	0
1	1	1

$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$

−1	0	1
−1	0	1
−1	0	1

$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$

Roberts

−1	0
0	1

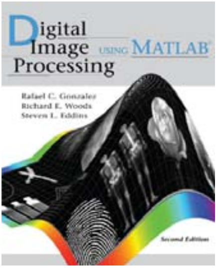
$G_x = z_9 - z_5$

0	−1
1	0

$G_y = z_8 - z_6$

FIGURE 10.5 Some edge detector masks and the first-order derivatives they implement.

Prepared by Kazim Fouladi | Fall 2016 | 2nd Edition



a b
c d
e f

FIGURE 10.6 (a) Original image. (b) Result of function `edge` using a vertical Sobel mask with the threshold determined automatically. (c) Result using a specified threshold. (d) Result of determining both vertical and horizontal edges with a specified threshold. (e) Result of computing edges at −45° with `imfilter` using a specified mask and a specified threshold. (f) Result of computing edges at +45° with `imfilter` using a specified mask and a specified threshold.

آشکارسازی لبه

مثال (سوبل)

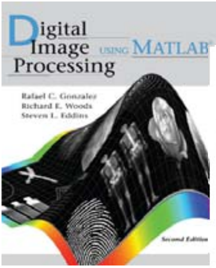
f

```
[gv, t] = edge(f,'sobel', 'vertical');
imshow(gv)
% t = 0.0516
% Fig.(b)
gv = edge(f, 'sobel', 0.15,'vertical');
% Fig.(c)
gboth = edge(f, 'sobel', 0.15);
% Fig.(d)
wneg45 = [-2 -1 0; -1 0 1; 0 1 2];
% weg45 =
%   -2 -1  0
%   -1  0  1
%    0  1  2
gneg45 = imfilter(tofloat(f), ...
                  wneg45, 'replicate');
T = 0.3*max(abs(gneg45(:)));
gneg45 = gneg45 >= T;
figure, imshow(gneg45);
% Fig.(e)
wpos45 = [0 1 2; -1 0 1; -2 -1 0];
gpos45 = imfilter(tofloat(f), ...
                  wpos45, 'replicate');
T = 0.3*max(abs(gpos45(:)));
gpos45 = gpos45 >= T;
figure, imshow(gpos45);
% Fig.(f)
```

a b
c d
e f

(a) Original image. (b) Result of function edge using a vertical Sobel mask with the threshold determined automatically. (c) Result using a specified threshold. (d) Result of determining both vertical and horizontal edges with a specified threshold. (e) Result of computing edges at 45° with imfilter using a specified mask and a specified threshold. (f) Result of computing edges at −45° with imfilter using a specified mask and a specified threshold.

MATLAB



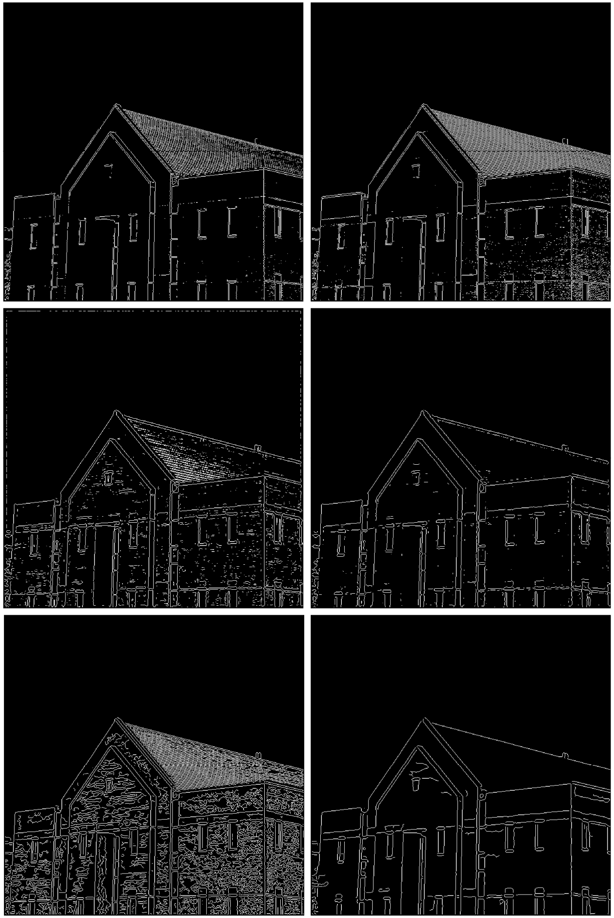
a b
c d
e f

FIGURE 10.7 Left column: Default results for the Sobel, LoG, and Canny edge detectors. Right column: Results obtained interactively to bring out the principal features in the original image of Fig. 10.6(a), while reducing irrelevant detail. The Canny edge detector produced the best result.

آشکارسازی لبه

مثال (مقایسه‌ی کنی، لاپلاسین گاوسی و سوبل)

f



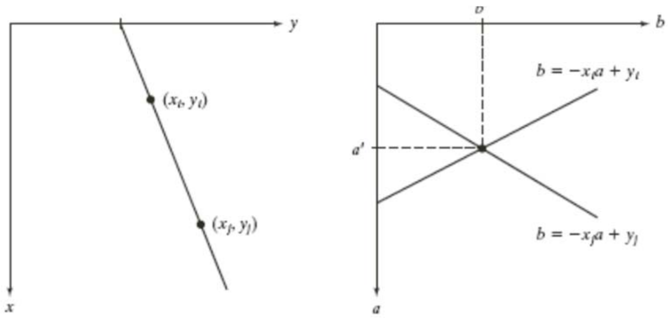
```
f = tofloat(f);
[gSobel_default, ts] = edge(f,'sobel');
% Fig.(a)
[gLOG_default, tlog] = edge(f,'log');
% Fig.(c)
[gCanny_default, tc] = edge(f,'canny');
% Fig.(e)
gSobel_best = edge(f,'sobel', 0.05);
% Fig.(b)
gLOG_best = edge(f,'log',0.003,2.25);
% Fig.(d)
gCanny_best = edge(f,'canny',...
                  [0.04 0.10], 1.5);
% Fig.(f)
```

a	b
c	d
e	f

FIGURE Left column: Default results for the Sobel, LoG, and Canny edge detectors. Right column: Results obtained interactively to bring out the principal features in the original image of Fig. 10.6(a) while reducing irrelevant, fine detail. The Canny edge detector produced the best results by far.

MATLAB

Chapter 10
Image Segmentation

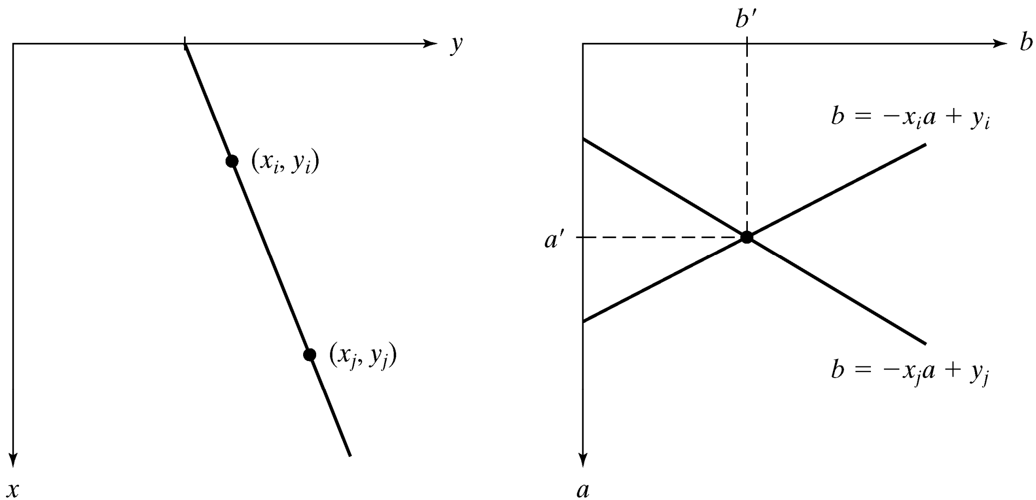


a b
FIGURE 10.8
(a) *xy*-plane.
(b) Parameter space.

تبدیل هاف

برای یافتن خطوط راست

HOUGH TRANSFORM



a b

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

MATLAB

Chapter 10
Image Segmentation

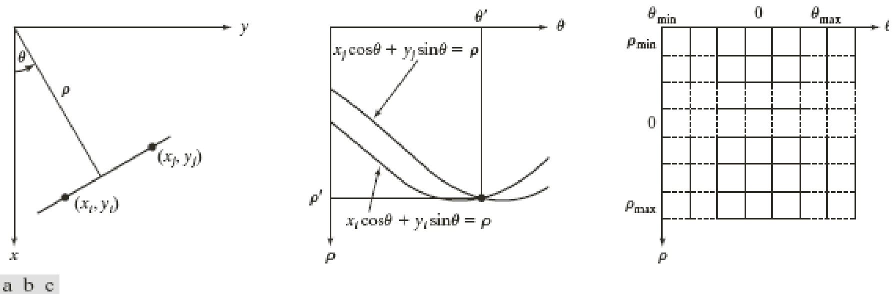
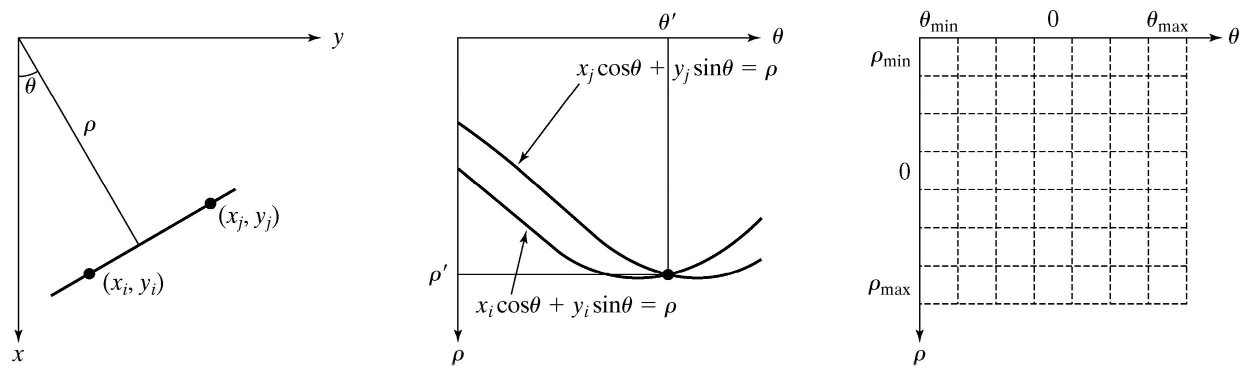


FIGURE 10.9 (a) Parameterization of lines in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection, (ρ', θ'), corresponds to the parameters of the line joining (x_i, y_i) and (x_j, y_j). (c) Division of the $\rho\theta$-plane into accumulator cells.

تبدیل هاف

HOUGH TRANSFORM



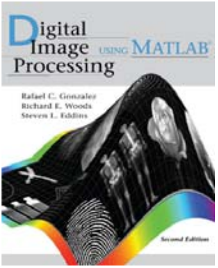
a b c

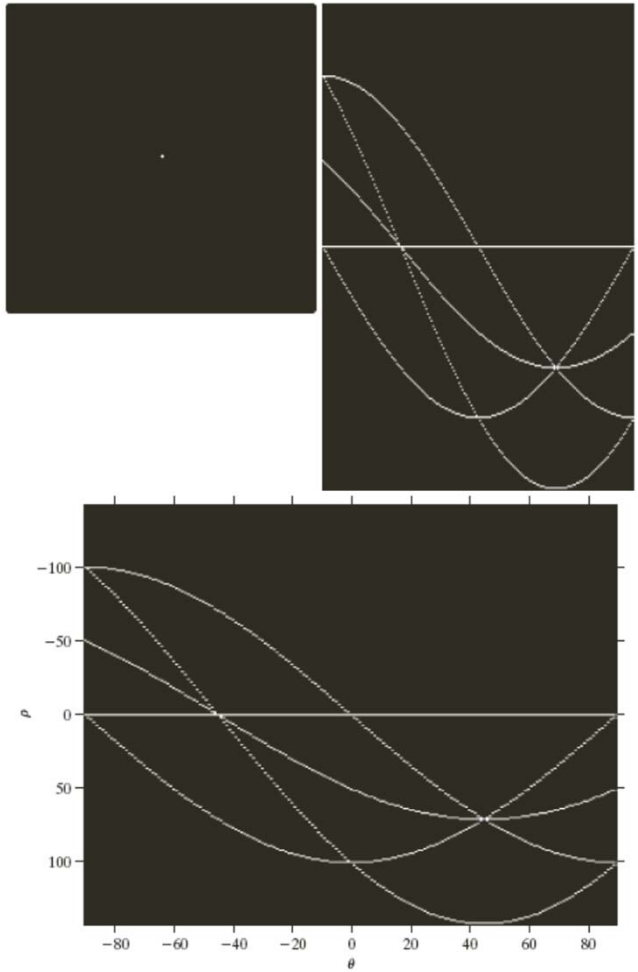
FIGURE (a) (ρ, θ) parameterization of lines in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection, (ρ', θ'), corresponds to the parameters of the line joining (x_i, y_i) and (x_j, y_j). (c) Division of the $\rho\theta$-plane into accumulator cells.

HOUGH TRANSFORM

```
[H, theta, rho] = hough(f)

[H, theta, rho] = hough(f, 'ThetaRes', val1, 'RhoRes', val2)

Peaks = houghpeaks(H, NumPeaks)

Lines = houghlines(f, theta, rho, peaks)
```


MATLAB





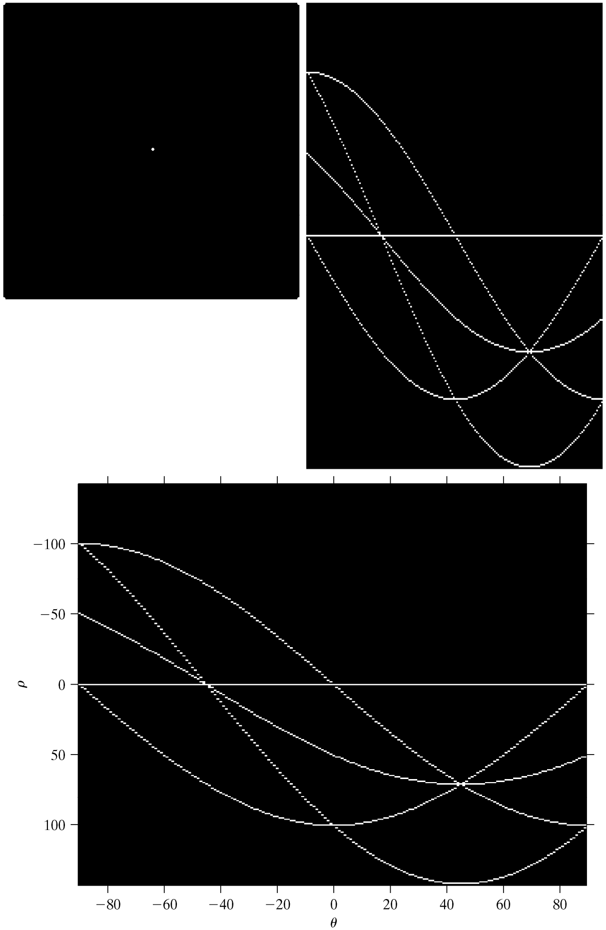
a b
c

FIGURE 10.10 (a) Binary image with five dots (four of the dots are in the corners). (b) Hough transform displayed using `imshow`. (c) Alternative Hough transform display with axis labeling. [The dots in (a) were enlarged to make them easier to see.]

تبدیل هاف

مثال

f



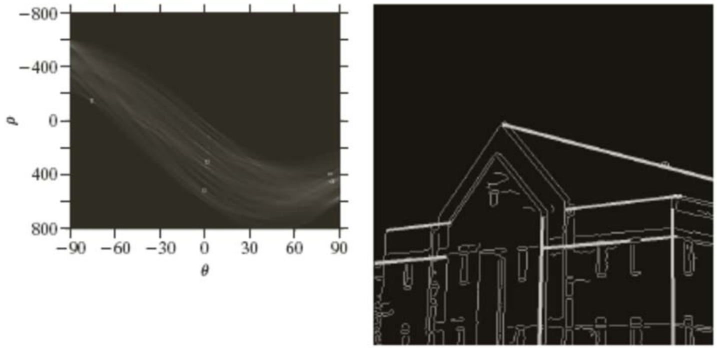

```
f = zeros(101, 101);
f(1, 1) = 1;
f(101, 1) = 1;
f(1, 101) = 1;
f(101, 101) = 1;
f(51, 51) = 1;

H = hough(f);
imshow(H, [ ])

[H, theta, rho] = hough(f);
imshow(H, [], 'XData', theta, 'YData',
rho,'InitialMagnif ication', 'fit')
axis on, axis normal
xlabel('\theta'), ylabel('\rho')
```

a	b
c	

(a) Binary image with five dots (four of the dots are in the corners). (b) Hough transform displayed using imshow. (c) Alternative Hough transform display with axis labeling. (The dots in (a) were enlarged to make them easier to see.)

MATLAB

Chapter 10
Image Segmentation

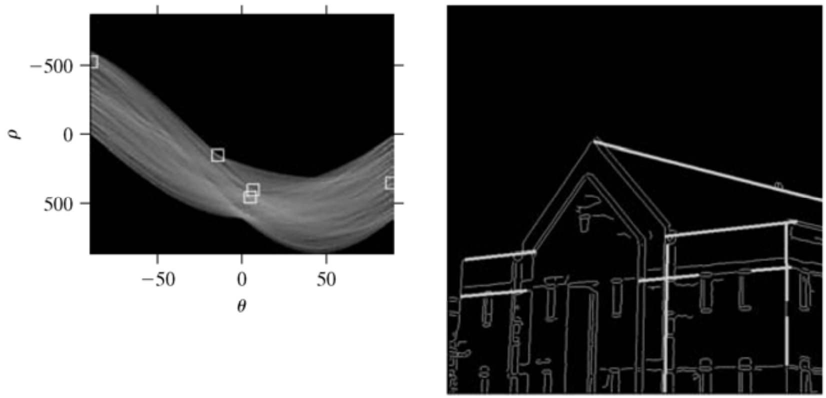


a b

FIGURE 10.11 (a) Hough transform with five peak locations selected. (b) Line segments (in bold) corresponding to the Hough transform peaks.

تبدیل هاف

استفاده از تبدیل هاف برای آشکارسازی خط و پیونددهی



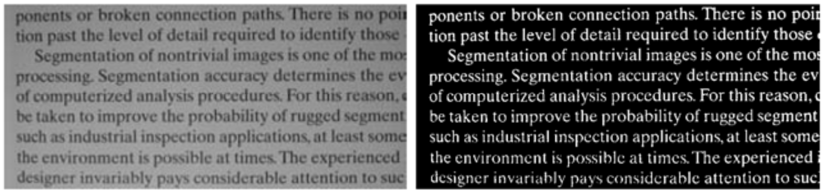
a b

(a) Hough transform with five peak locations selected. (b) Line segments corresponding to the Hough transform peaks.

```
[H, theta, rho] = hough(f, 'ThetaResolution', 0.2);
imshow(H, [], 'XData', theta, 'YData', rho, ' InitialMagnif ication', 'fit')
axis on, axis normal
xlabel('\theta'), ylabel('\rho')

peaks = houghpeaks(H, 5);
hold on
plot(theta(peaks(:, 2)), rho(peaks(:, 1)),'linestyle', 'none', 'marker', 's', 'color', 'w')

lines = houghlines(f, theta, rho, peaks);
figure, imshow(f), hold on
for k = 1:length(lines)
   xy = [lines(k).pointl ; lines(k).point2];
   plot(xy(:,1), xy(:,2), 'LineWidth', 4, 'Color', [.8 .8 .8]);
end
```

آستانه‌گیری

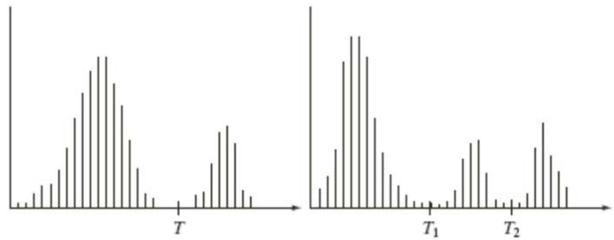
مثال



a b

FIGURE 10.13
(a) Scanned text.
(b) Thresholded text obtained using function `graythresh`.

Chapter 10
Image Segmentation



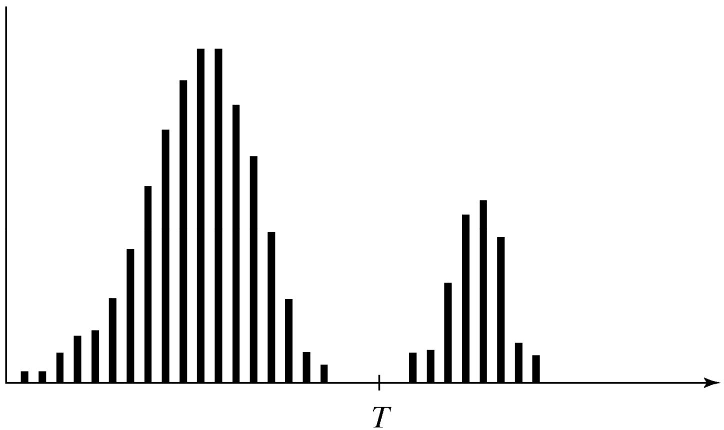
a b

FIGURE 10.12 Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds. These are *unimodal* and *bimodal* histograms, respectively.

آستانه‌گیری

انتخاب آستانه در هیستوگرام دو-مده

THRESHOLDING



Selecting a threshold by visually analyzing a bimodal histogram.

Chapter 10
Image Segmentation

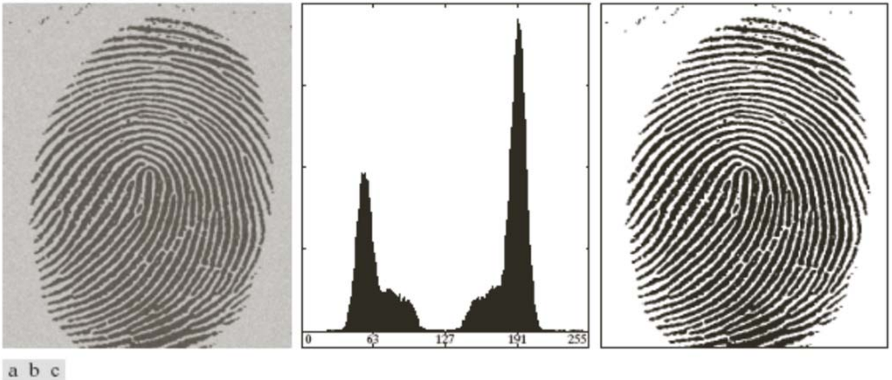
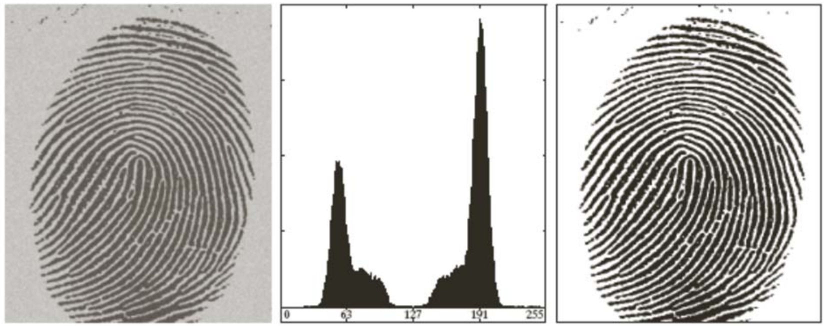


FIGURE 10.13 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added manually for clarity). (Original courtesy of the National Institute of Standards and Technology.)

آستانه‌گیری

انتخاب آستانه در هیستوگرام دو–مده

THRESHOLDING



a b c

```
count = 0;
T = mean2(f);
done = false;
while ~done
  count = count + 1;
  g = f > T;
  Tnext = 0.5*(mean(f(g)) + mean(f(~g)));
  done = abs(T - Tnext) < 0.5;
  T = Tnext;
end
```

```
>> count
% count = 2

>> T
% T = 125.3860

>> g = im2bw(f, T/255);
>> imshow(f) % Fig.(a).
>> figure, imhist(f) % Fig.(b)
>> figure, imshow(g) % Fig.(c)
```


MATLAB

آستانه‌گیری

روش اوتسو

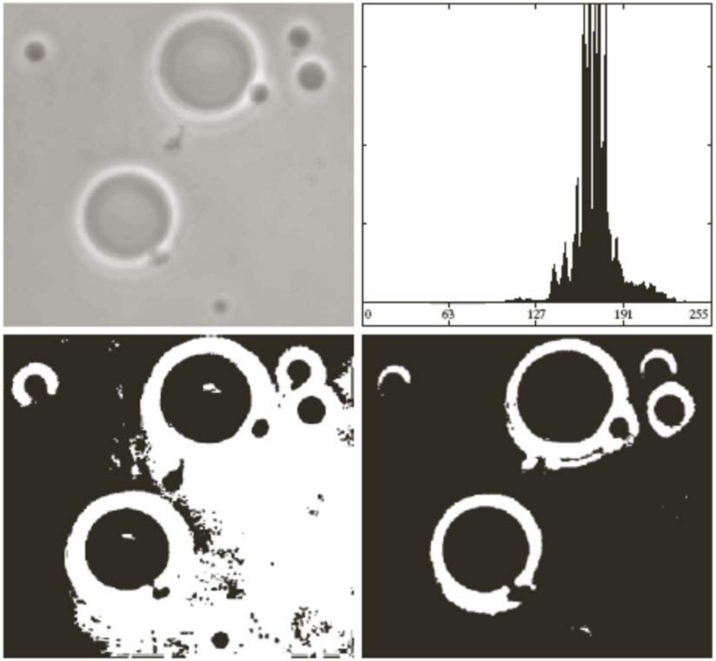
THRESHOLDING

```
[T,SM] = graythresh(f)
```

`f` is the input image,
`T` is the resulting threshold, normalized to the range [0,1],
`SM` is the separability measure.

MATLAB

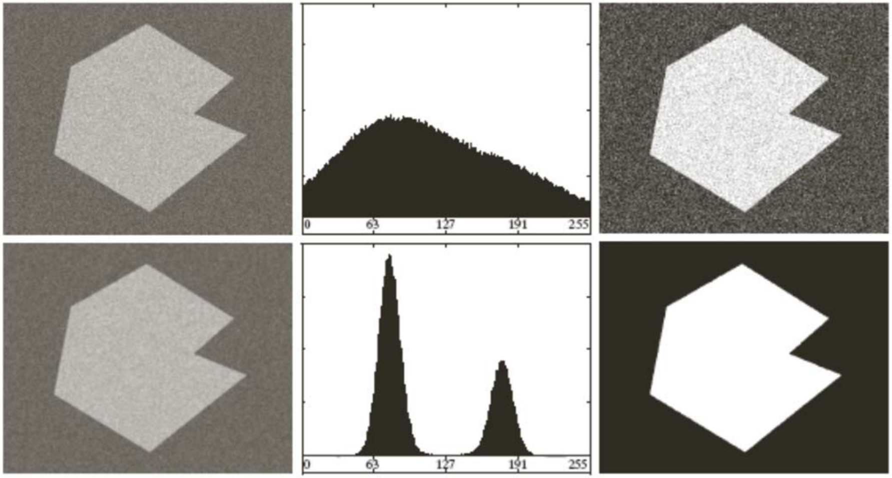
Chapter 10
Image Segmentation



a b
c d

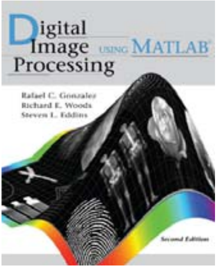
FIGURE 10.14 (a) Original image. (b) Histogram (high values were clipped to highlight details in the lower values). (c) Segmentation result using the basic global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

Chapter 10
Image Segmentation

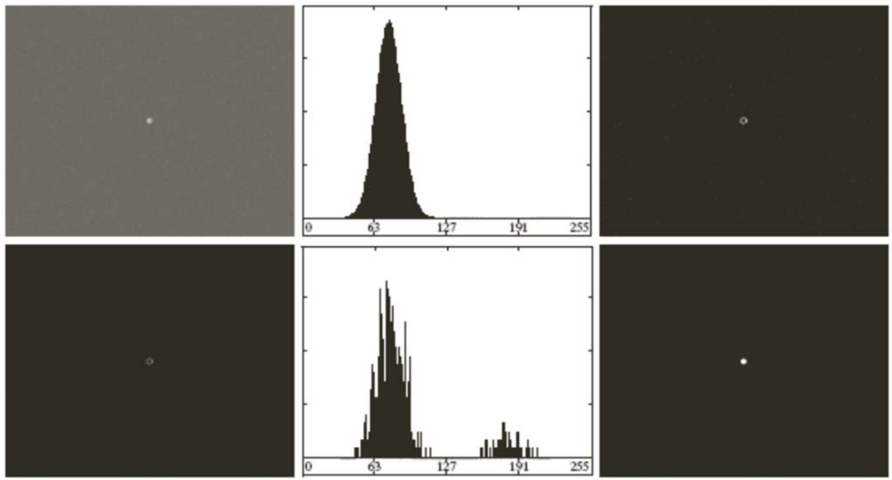


a b c
d e f

FIGURE 10.15 (a) Noisy image, 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.

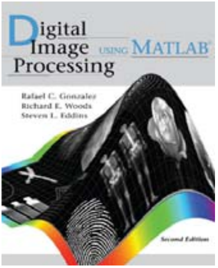


Chapter 10
Image Segmentation

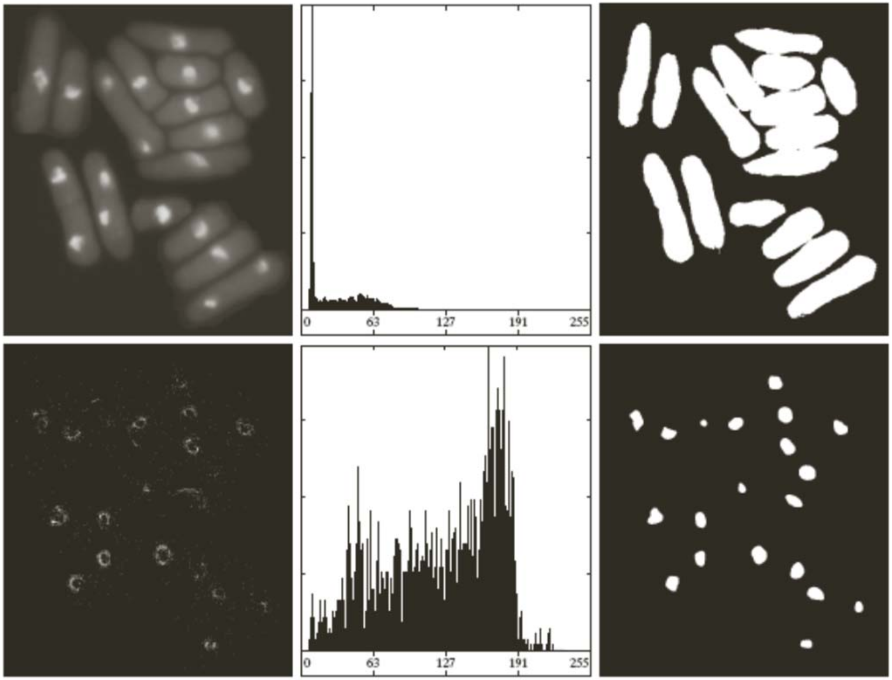


a b c
d e f

FIGURE 10.16 (a) Noisy image of small septagon, and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.9 percentile level. (d) Image formed by 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 found using the histogram in (e). (The threshold found was 133.5, which is approximately midway between the peaks in this histogram.)



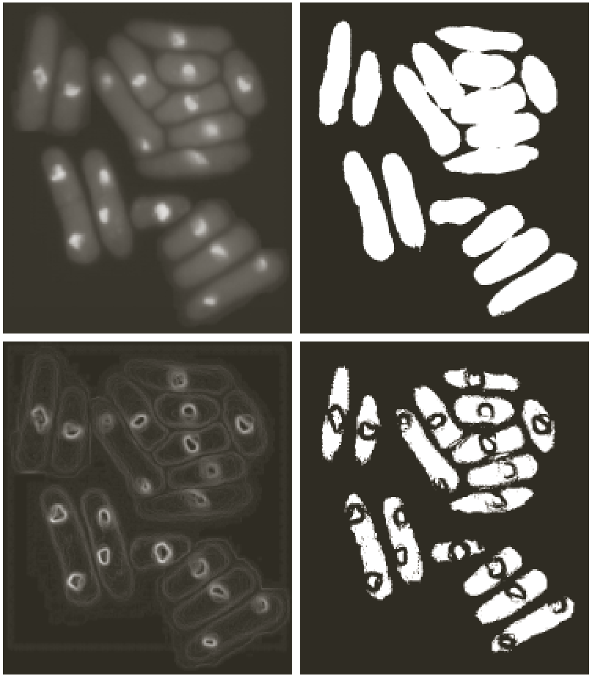
Chapter 10
Image Segmentation



a b c
d e f

FIGURE 10.17 (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) using function `graythresh`. (d) Product of the marker and original images. (e) Histogram of the nonzero pixels in (d). (f) Image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)

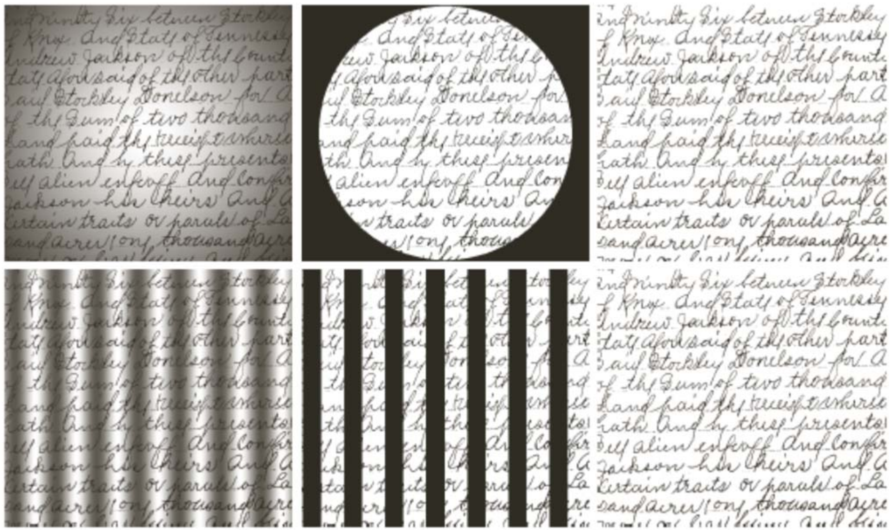
Chapter 10
Image Segmentation



a b
c d

FIGURE 10.18 (a) Yeast cell image. (b) Image segmented using Otsu's method. (c) Image of local standard deviations. (d) Image segmented using local thresholding.

Chapter 10
Image Segmentation



a b c
d e f

FIGURE 10.19 (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages. (d)-(f) Results of using the same sequence of operations on an image corrupted by sinusoidal shading.

بخش‌بندی مبتنی بر ناحیه

روش نمو ناحیه

```
[g, NR, SI, TI] = regiongrow(f, S, T)
```

`f` is an image to be segmented.

parameter `S` can be an array (the same size as f) or a scalar.

If S is an array, it must contain 1s at all the coordinates where seed points are located and 0s elsewhere. Such an array can be determined by inspection, or by an external seed-finding function. If S is a scalar, it defines an intensity value such that all the points in f with that value become seed points.

parameter `T` can be an array (the same size as f) or a scalar.

If T is an array, it contains a threshold value for each location in f. If T is a scalar, it defines a global threshold. The threshold value(s) is (are) used to test if a pixel in the image is sufficiently similar to the seed or seeds to which it is 8-connected.

All values of S and T must be scaled to the range [0,1], independently of the class of the input image.

`g` is the segmented image, with the members of each region being labeled with a different integer value.

Parameter `NR` is the number of regions found.

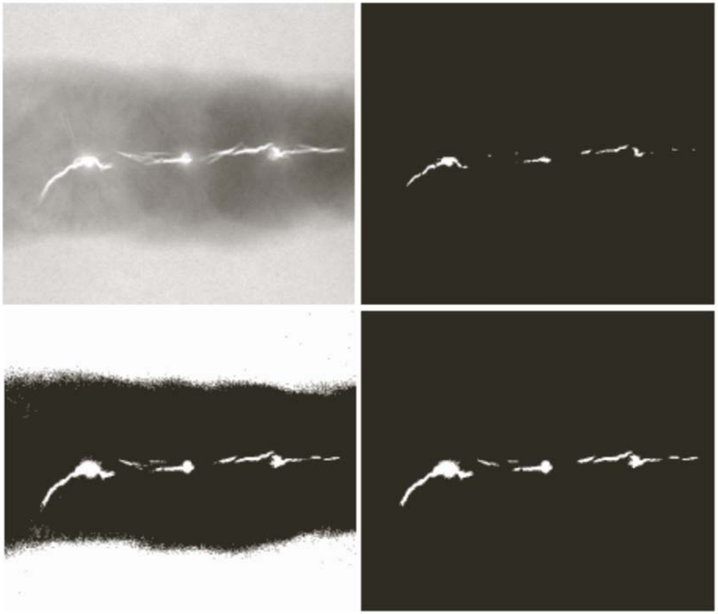
Parameter `SI` is an image containing the seed points.

Parameter `TI` is an image containing the pixels that passed the threshold test before they were processed for connectivity.

Both `SI` and `TI` are of the same size as f.

MATLAB

Chapter 10
Image Segmentation

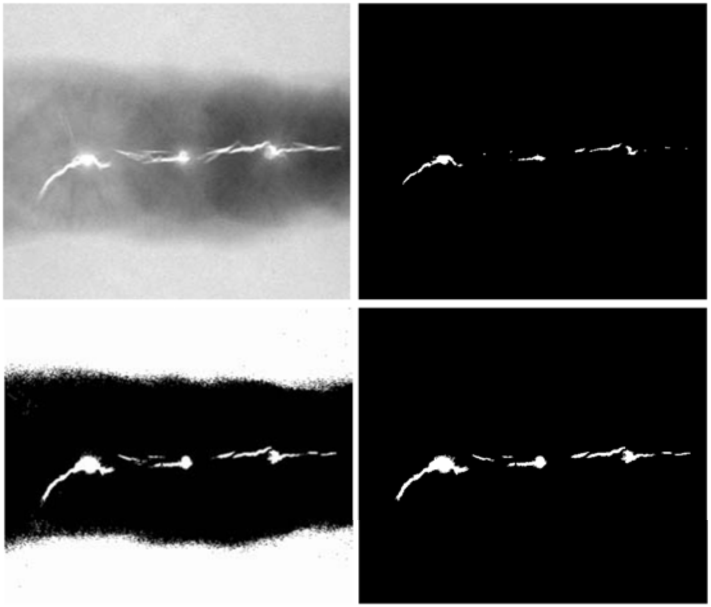


a b
c d

FIGURE 10.20 (a) Image showing defective welds. (b) Seed points. (c) Binary image showing all the pixels (in white) that passed the threshold test. (d) Result after all the pixels in (c) were analyzed for 8-connectivity to the seed points. (Original image courtesy of X-TEK Systems, Ltd.)

آستانه‌گیری

مثال



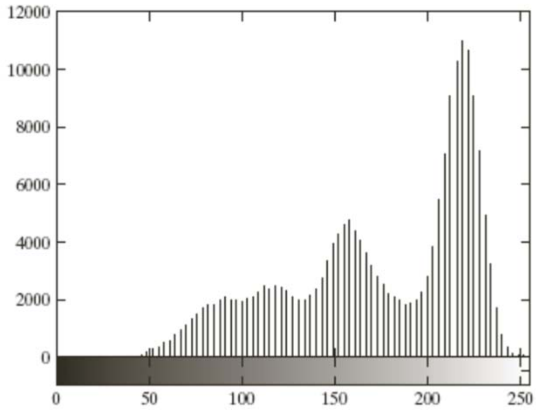
a b
c d

FIGURE 10.14 (a) Image showing defective welds. (b) Seed points. (c) Binary image showing all the pixels (in white) that passed the threshold test. (d) Result after all the pixels in (c) were analyzed for 8-connectivity to the seed points. (Original image courtesy of X-TEK Systems, Ltd.)

MATLAB

Chapter 10
Image Segmentation

FIGURE 10.21
Histogram of Fig. 10.20(a).



آستانه‌گیری

مثال

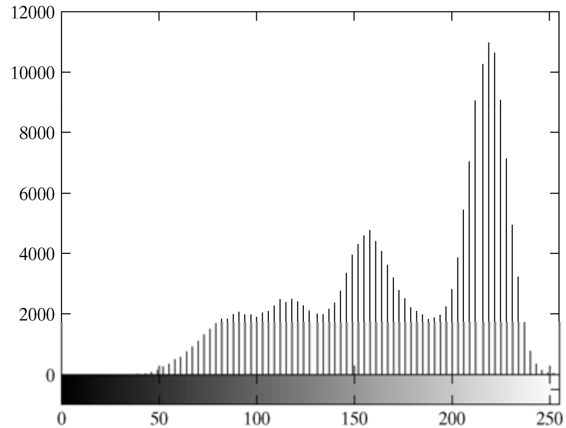
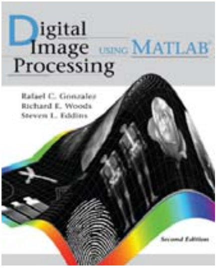
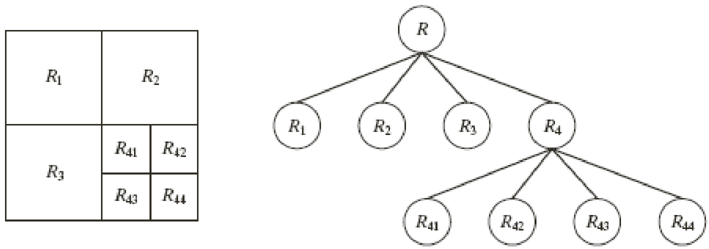


FIGURE 10.15 Histogram of Fig. 10.14(a).

MATLAB



Chapter 10
Image Segmentation

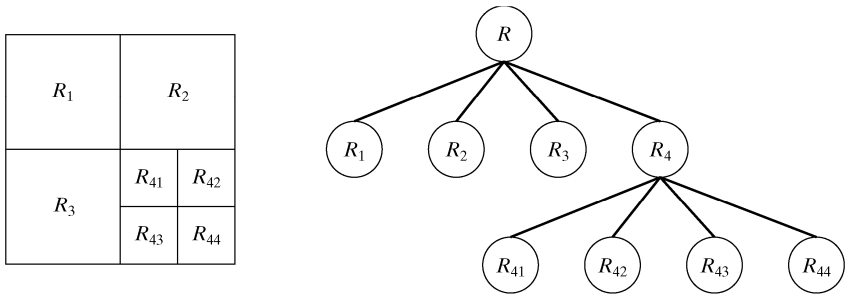


a b
FIGURE 10.22 (a) Partitioned image. (b) Corresponding quadtree.

بخش‌بندی مبتنی بر ناحیه

روش شکافت و ادغام ناحیه

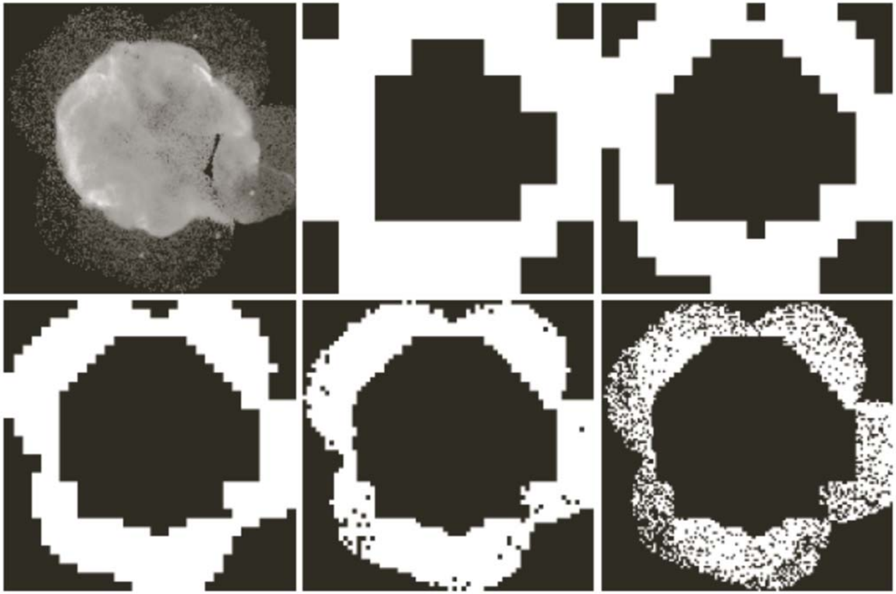
```
g = splitmerge(f, mindim, fun)
```



a b

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

Chapter 10
Image Segmentation

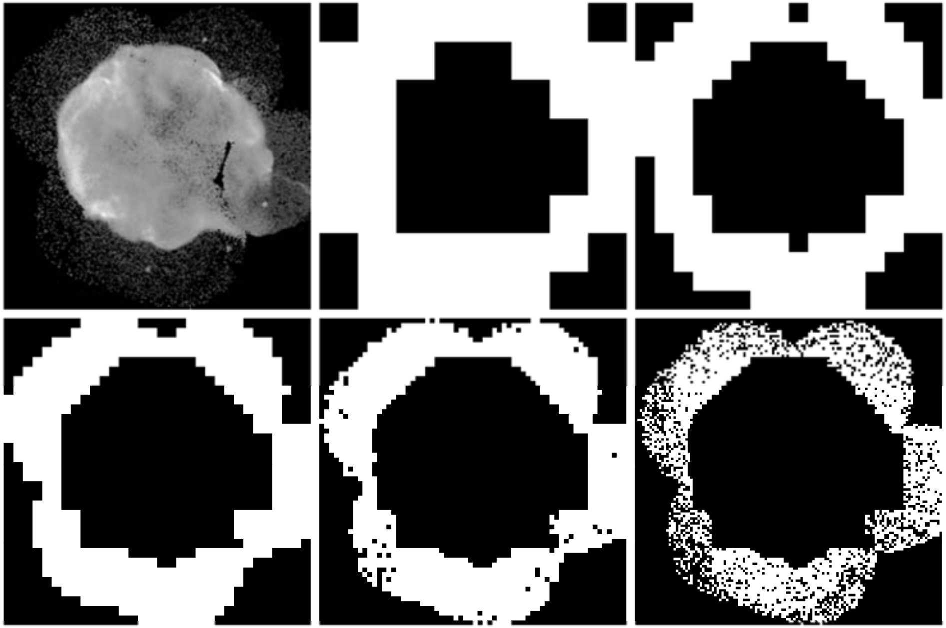


a b c
d e f

FIGURE 10.23 Image segmentation using a split-and-merge algorithm. (a) Original image. (b) through (f) Results of segmentation using function `splitmerge` with values of `mindim` equal to 32, 16, 8, 4, and 2, respectively. (Original image courtesy of NASA.)

بخش‌بندی مبتنی بر ناحیه

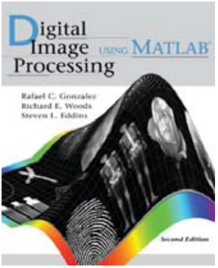
روش شکافت و ادغام ناحیه: مثال



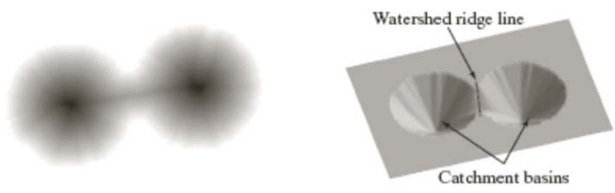
a	b	c
d	e	f

FIGURE Image segmentation by a split-and-merge procedure. (a) Original image. (b) through (f) results of segmentation using function `splitmerge` with values of `mindim` equal to 32, 16, 8, 4, and 2, respectively. (Original image courtesy of NASA.)

MATLAB



Chapter 10
Image Segmentation

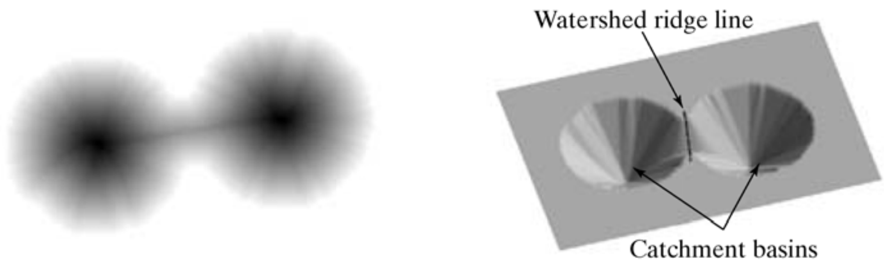


a b

FIGURE 10.24 (a) Gray-scale scale image. (b) Image viewed as a surface, showing a watershed ridge line and catchment basins.

بخش‌بندی با استفاده از تبدیل آب‌پخشان

SEGMENTATION USING THE WATERSHED TRANSFORM



a b

(a) Gray-scale image of dark blobs. (b) Image viewed as a surface, with labeled watershed ridge line and catchment basins.

```
L = watershed(A, conn)
```

بخش‌بندی با استفاده از تبدیل آب‌پخشان

تبدیل فاصله

1	1	0	0	0
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	1	1	1	0

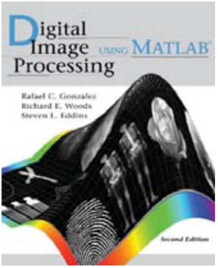
0.00	0.00	1.00	2.00	3.00
0.00	0.00	1.00	2.00	3.00
1.00	1.00	1.41	2.00	2.24
1.41	1.00	1.00	1.00	1.41
1.00	0.00	0.00	0.00	1.00

a b

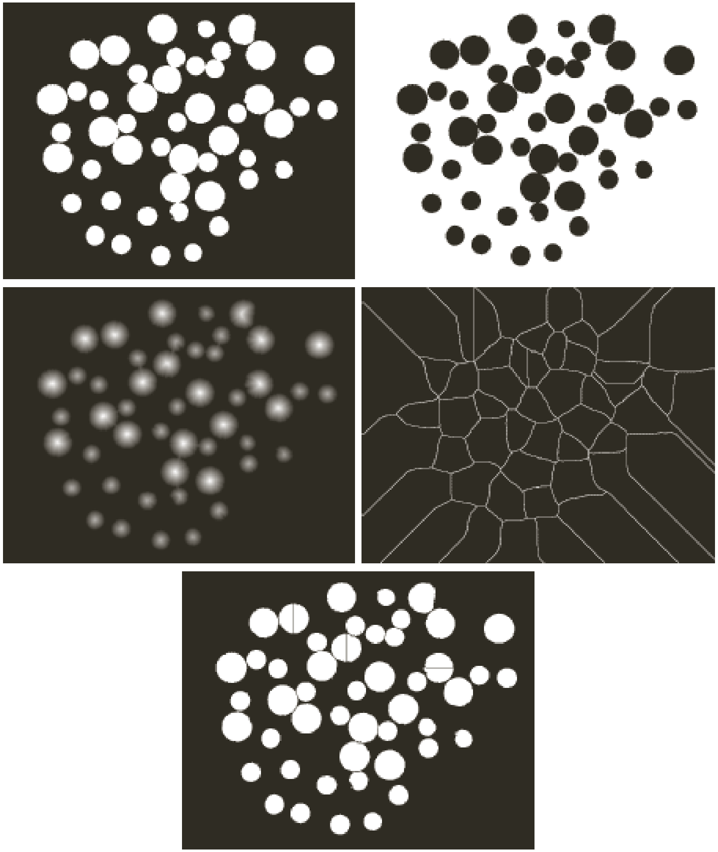
(a) Small binary image.
(b) Distance transform.

```
D = bwdist(f)
```


MATLAB



Chapter 10

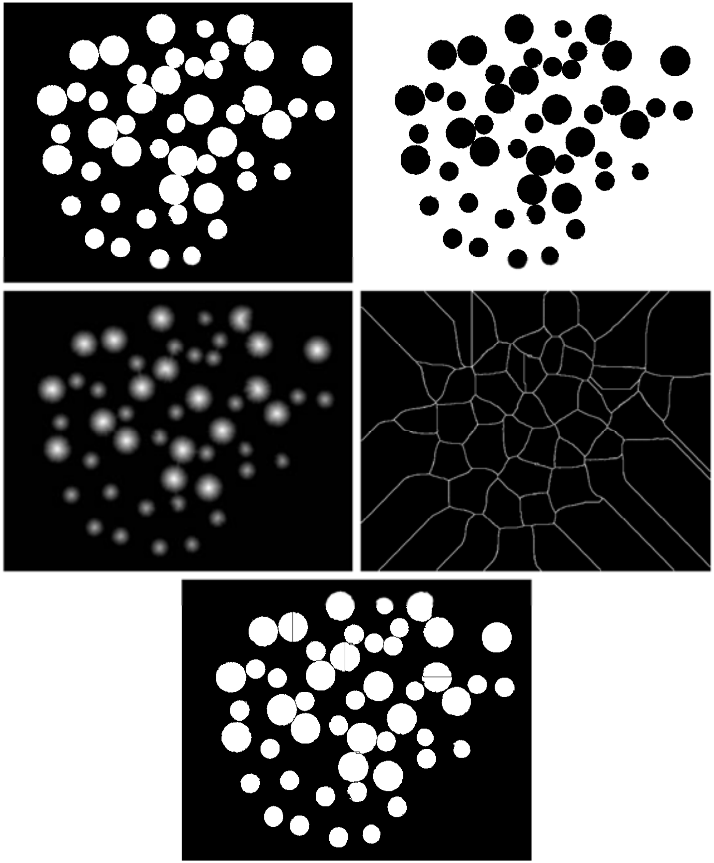


a b
c d
e

FIGURE 10.26 (a) Binary image. (b) Complement of image in (a). (c) Distance transform. (d) Watershed ridge lines of the negative of the distance transform. (e) Watershed ridge lines superimposed in black over original binary image. Some oversegmentation is evident.

بخش‌بندی با استفاده از تبدیل آب‌پخشان

با استفاده از تبدیل فاصله: مثال

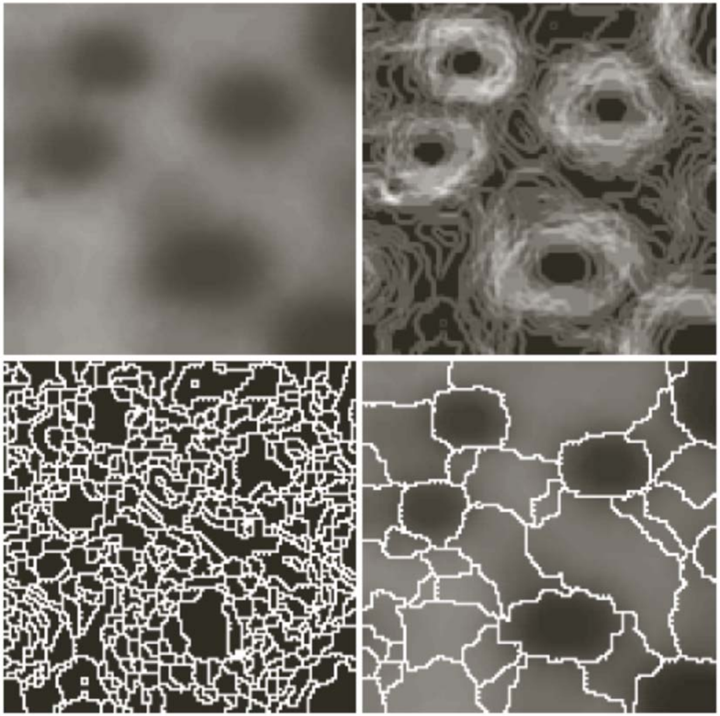


```
g = im2bw(f, graythresh(f));
% Fig.(a)
gc = ~g;
% Fig.(b)
D = bwdist(gc); % Fig.(c)
L = watershed(-D); % Fig.(d)
w = (L == 0);
g2 = g & ~w;
% Fig.(e)
```

a b
c d
e

(a) Binary image. (b) Complement of image in (a). (c) Distance transform. (d) Watershed ridge lines of the negative of the distance transform. (e) Watershed ridge lines superimposed in black over original binary image. Some oversegmentation is evident.

Chapter 10
Image Segmentation

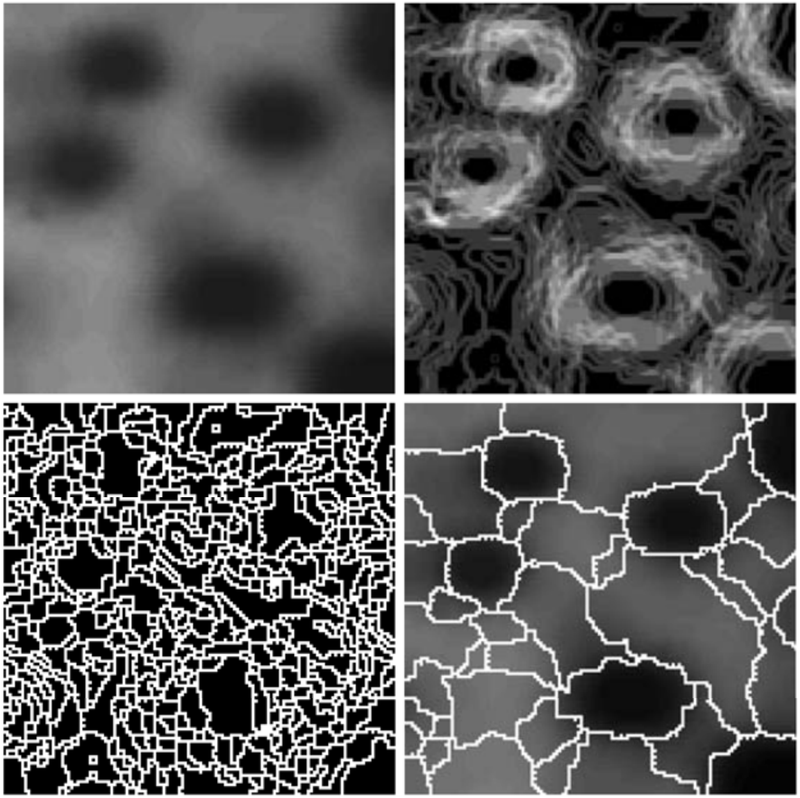


a b
c d

FIGURE 10.27 (a) Gray-scale image of small blobs. (b) Gradient magnitude image. (c) Watershed transform of (b), showing severe oversegmentation. (d) Watershed transform of the smoothed gradient image; some oversegmentation is still evident. (Original image courtesy of Dr. S. Beucher, CMM/ Ecole de Mines de Paris.)

بخش‌بندی با استفاده از تبدیل آب‌پخشان

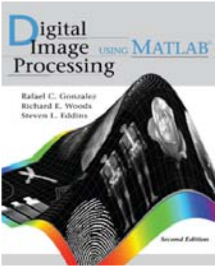
با استفاده از گرادیان‌ها: مثال



a b
c d

(a) Gray-scale image of small blobs. (b) Gradient magnitude image. (c) Watershed transform of (b), showing severe oversegmentation. (d) Watershed transform of the smoothed gradient image; some oversegmentation is still evident. (Original image courtesy of Dr. S. Beucher, CMM/Ecole de Mines de Paris.)

MATLAB



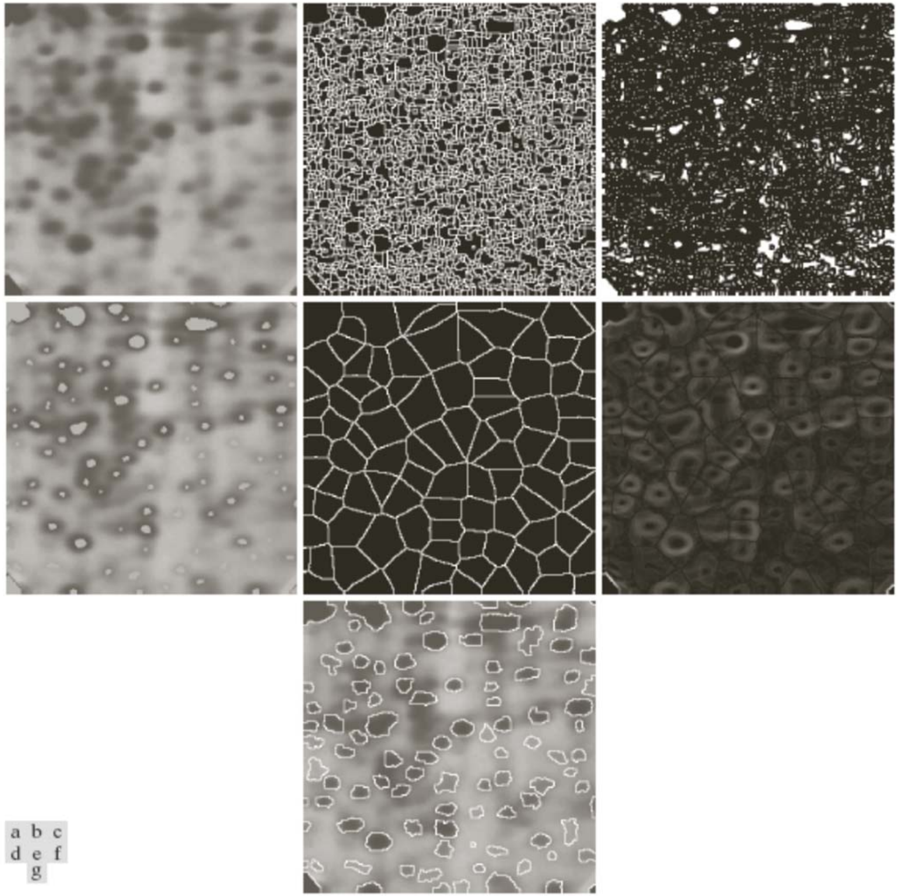


FIGURE 10.28 (a) Gel image. (b) Oversegmentation resulting from applying the watershed transform to the gradient magnitude image. (c) Regional minima of gradient magnitude. (d) Internal markers. (e) External markers. (f) Modified gradient magnitude. (g) Segmentation result. (Original image courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

بخش‌بندی با استفاده از تبدیل آب‌پخشان

با استفاده از علامت‌گذارها

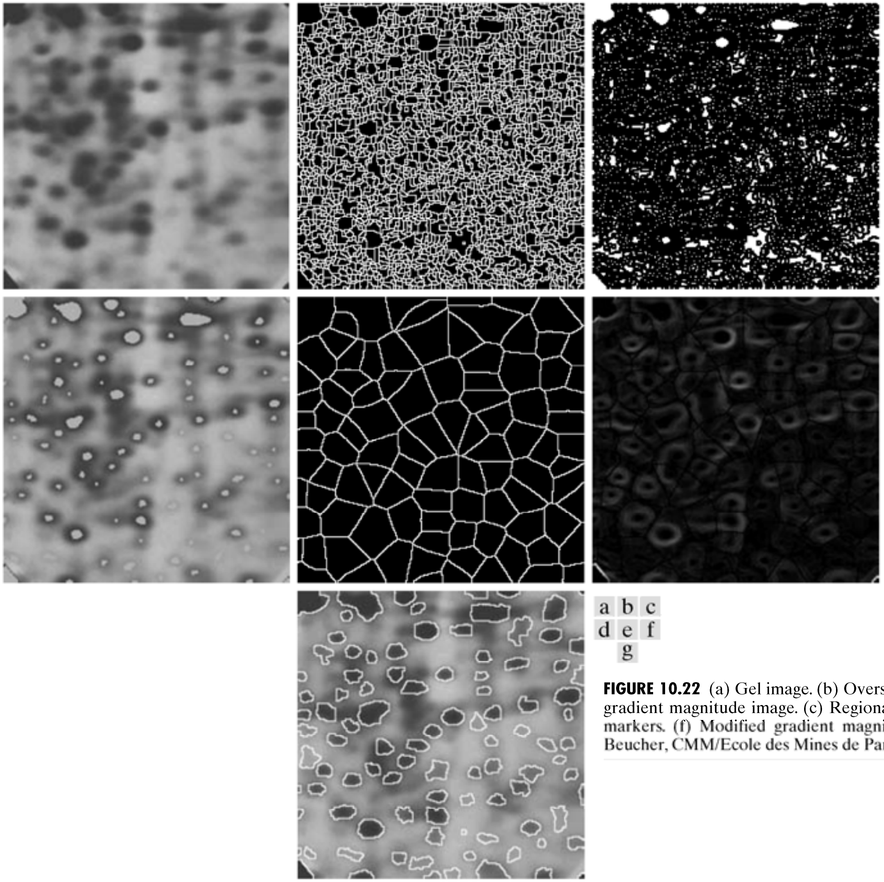
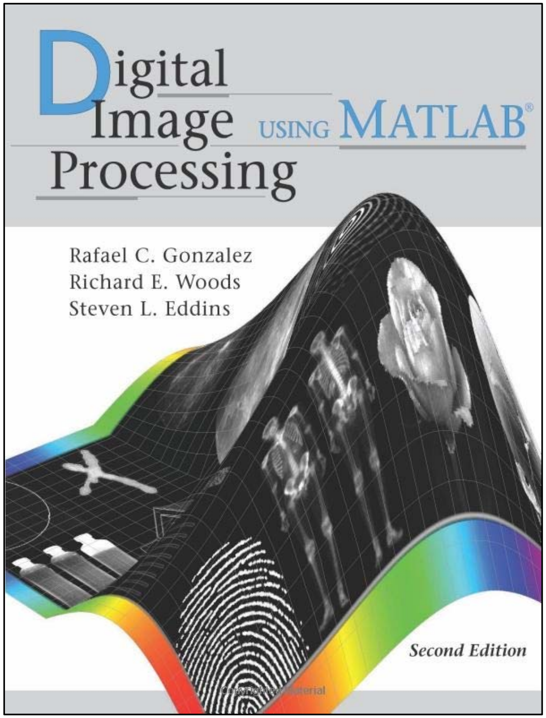


FIGURE 10.22 (a) Gel image. (b) Oversegmentation resulting from applying the watershed transform to the gradient magnitude image. (c) Regional minima of gradient magnitude. (d) Internal markers. (e) External markers. (f) Modified gradient magnitude. (g) Segmentation result. (Original image courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

منبع اصلی



Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins,
Digital Image Processing Using MATLAB®,
Second Edition, Pearson Prentice Hall, 2008.
Chapter 11