

FEATURE EXTRACTION

- 11.1 Background
- 11.2 Boundary Preprocessing
- 11.3 Boundary Feature Descriptors
- 11.4 Region Feature Descriptors
- 11.5 Principle Components as Feature Descriptor
- 11.6 Whole-Image Features
- 11.7 Scale-Invariant Feature Transform (SIFT)

Prof. Ta-Te Lin

Dept. of Biomechatronics Engineering, National Taiwan University



11.1 Background

- There are two principal aspects of image feature extraction: feature detection, and feature description.
- A feature descriptor is ***invariant*** with respect to a set of transformations if its value remains unchanged after the application of any transformation from the family.
- A feature descriptor is ***covariant*** with respect to a set of transformations if applying to the entity any transformation from the set produces the same result in the descriptor.
- Another major classification of features is ***local*** vs. ***global***.
- Numerical features are usually presented in the form of a ***feature vector***.



11.2 Boundary Preprocessing

11.2.1 Boundary Following (Tracing)

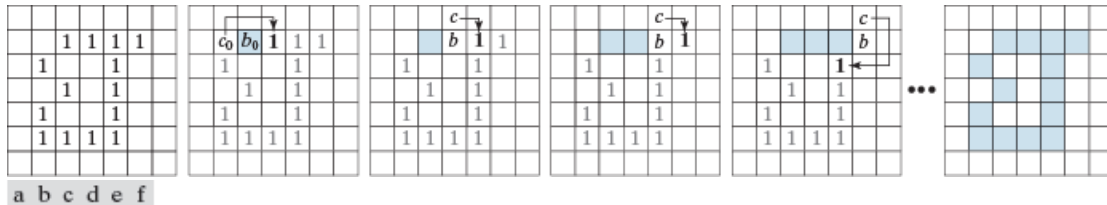


FIGURE 11.1 Illustration of the first few steps in the boundary-following algorithm. The point to be processed next is labeled in bold, black; the points yet to be processed are gray; and the points found by the algorithm are shaded. Squares without labels are considered background (0) values.

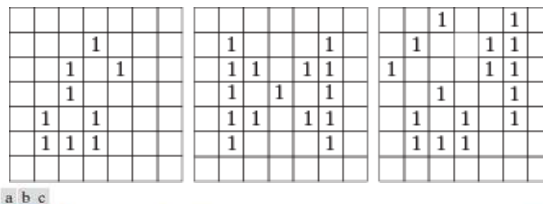


FIGURE 11.2 Examples of boundaries that can be processed by the boundary-following algorithm. (a) Closed boundary with a branch. (b) Self-intersecting boundary. (c) Multiple boundaries (processed one at a time).

- If we start with a binary region instead of a boundary, the algorithm extracts the outer boundary of the region.



11.2 Boundary Preprocessing

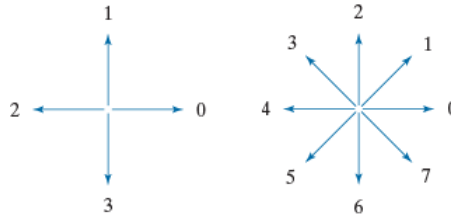
11.2.2 Chain Codes

■ Freeman Chain Codes

a b

FIGURE 11.3

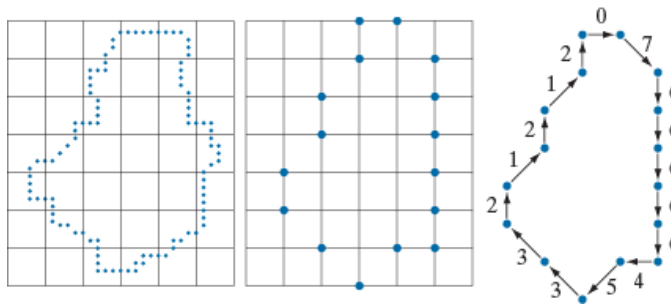
Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.



a b c

FIGURE 11.4

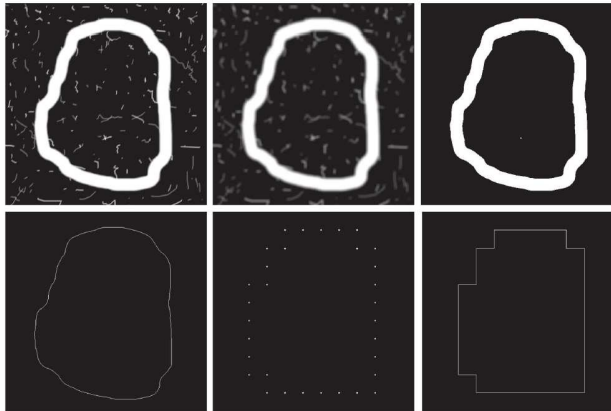
(a) Digital boundary with resampling grid superimposed. (b) Result of resampling. (c) 8-directional chain-coded boundary.



11.2 Boundary Preprocessing

11.2.2 Chain Codes

■ Freeman Chain Codes



a b c
d e f

FIGURE 11.5 (a) Noisy image of size 570×570 pixels. (b) Image smoothed with a 9×9 box kernel. (c) Smoothed image, thresholded using Otsu's method. (d) Longest outer boundary of (c). (e) Subsampled boundary (the points are shown enlarged for clarity). (f) Connected points from (e).

- The 8-directional Freeman chain code

000060666666666444444242222202202

- The integer of minimum magnitude

000060666666666444444242222202202

- The first difference of the code

00062600000006000006260000620626



11.2 Boundary Preprocessing

11.2.2 Chain Codes

■ Slope Chain Codes

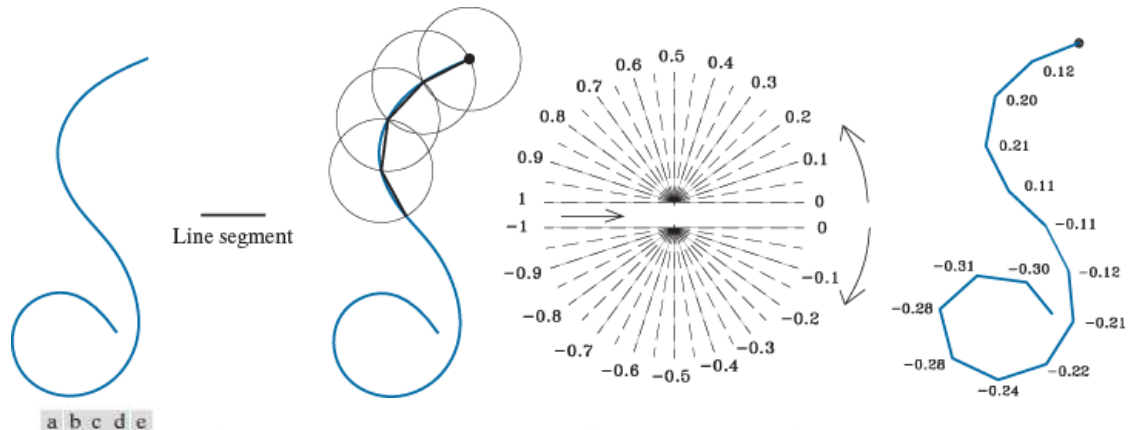


FIGURE 11.6 (a) An open curve. (b) A straight-line segment. (c) Traversing the curve using circumferences to determine slope changes; the dot is the origin (starting point). (d) Range of slope changes in the open interval $(-1, 1)$ (the arrow in the center of the chart indicates direction of travel). There can be ten subintervals between the slope numbers shown. (e) Resulting coded curve showing its corresponding numerical sequence of slope changes. (Courtesy of Professor Ernesto Bribriesca, IIMAS-UNAM, Mexico.)



11.2 Boundary Preprocessing

11.2.3 Boundary Approximations Using Minimum-Perimeter Polygons

■ Foundation

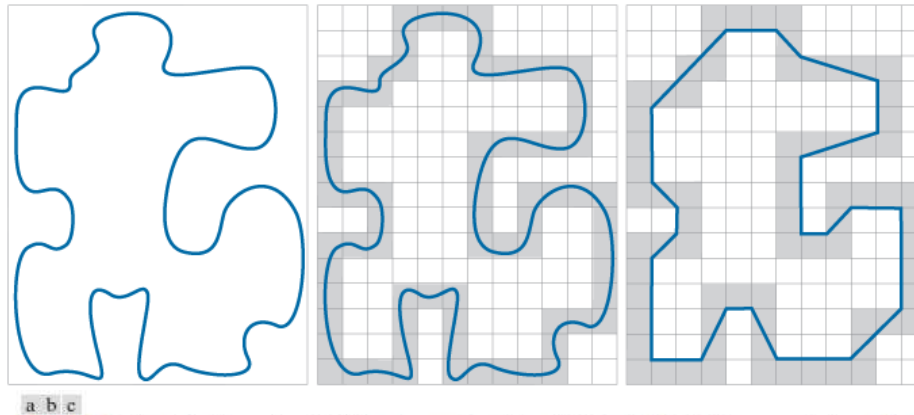


FIGURE 11.7 (a) An object boundary. (b) Boundary enclosed by cells (shaded). (c) Minimum-perimeter polygon obtained by allowing the boundary to shrink. The vertices of the polygon are created by the corners of the inner and outer walls of the gray region.



11.2 Boundary Preprocessing

11.2.3 Boundary Approximations Using Minimum-Perimeter Polygons

■ Foundation

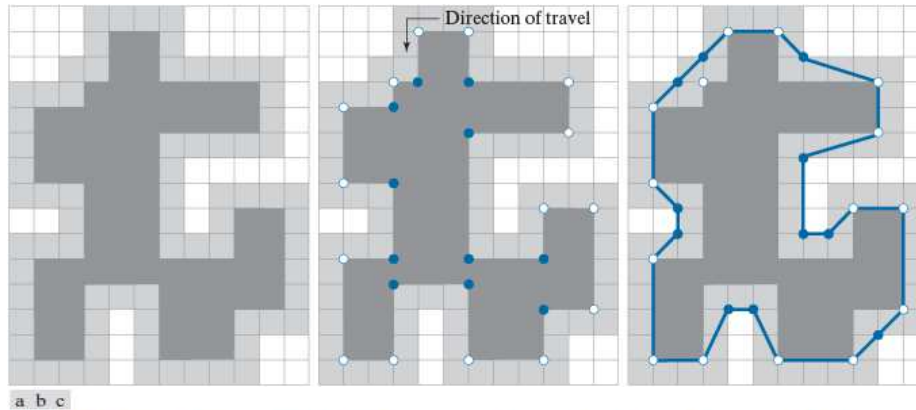


FIGURE 11.8 (a) Region (dark gray) resulting from enclosing the original boundary by cells (see Fig. 11.7). (b) Convex (white dots) and concave (blue dots) vertices obtained by following the boundary of the dark gray region in the counterclockwise direction. (c) Concave vertices (blue dots) displaced to their diagonal mirror locations in the outer wall of the bounding region; the convex vertices are not changed. The MPP (solid boundary) is superimposed for reference.



11.2 Boundary Preprocessing

11.2.3 Boundary Approximations Using Minimum-Perimeter Polygons

- MPP Algorithm

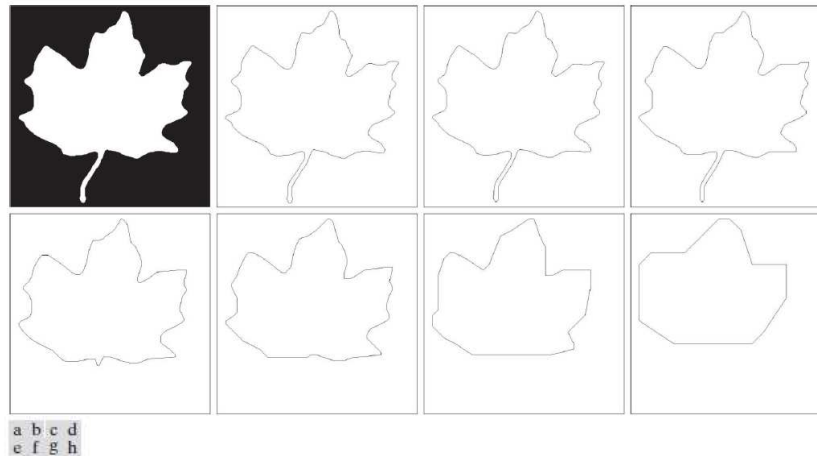
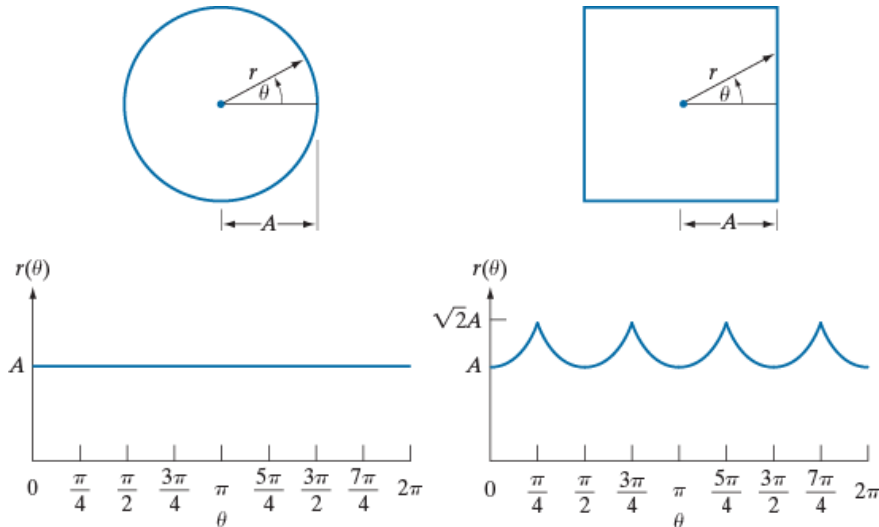


FIGURE 11.9 (a) 566×566 binary image. (b) 8-connected boundary. (c) through (h), MMPs obtained using square cells of sizes 2, 4, 6, 8, 16, and 32, respectively (the vertices were joined by straight-line segments for display). The number of boundary points in (b) is 1900. The numbers of vertices in (c) through (h) are 206, 127, 92, 66, 32, and 13, respectively. Images (b) through (h) are shown as negatives to make the boundaries easier to see.



11.2 Boundary Preprocessing

11.2.4 Signatures



a b

FIGURE 11.10

Distance-versus-angle signatures.

In (a), $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern

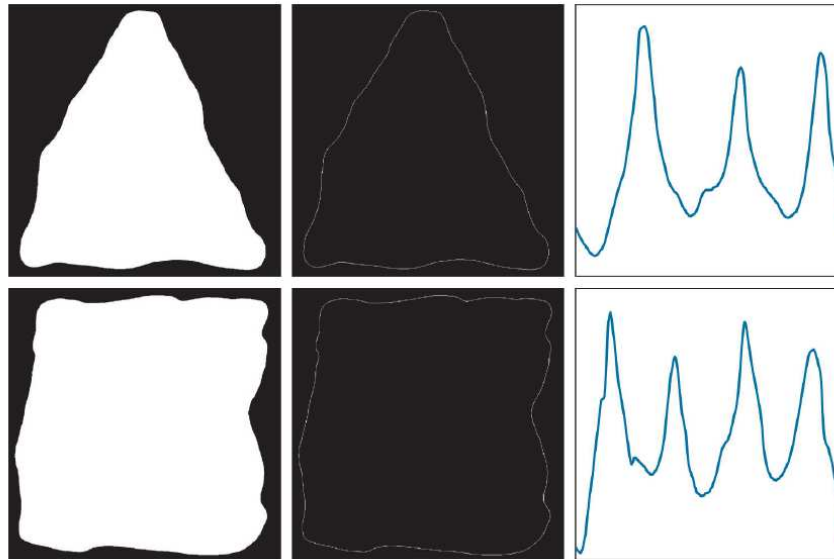
$r(\theta) = A \sec \theta$ for $0 \leq \theta \leq \pi/4$, and $r(\theta) = A \csc \theta$ for $\pi/4 < \theta \leq \pi/2$.

- The basic idea of using signatures is to reduce the boundary representation to a 1-D function that presumably is easier to describe than the original 2-D boundary.



11.2 Boundary Preprocessing

11.2.4 Signatures



a b c
d e f

FIGURE 11.11

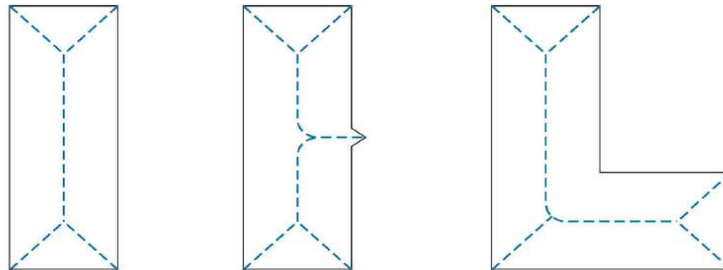
(a) and (d) Two binary regions, (b) and (e) their external boundaries, and (c) and (f) their corresponding $r(\theta)$ signatures. The horizontal axes in (c) and (f) correspond to angles from 0° to 360° , in increments of 1° .



11.2 Boundary Preprocessing

11.2.5 Skeletons, Medial Axes, and Distance Transforms

■ Medial Axes



a b c
FIGURE 11.12
Medial axes
(dashed) of three
simple regions.

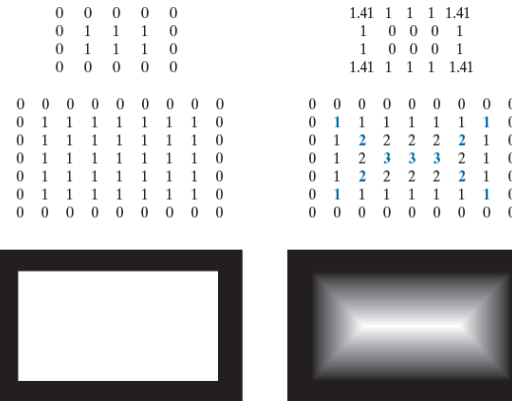
- The skeleton of a region is the set of points in the region that are equidistant from the border of the region.
- The MAT of a region R with border B is as follows: For each point p in R , we find its closest neighbor in B . If p has more than one such neighbor, it is said to belong to the medial axis of R .



11.2 Boundary Preprocessing

11.2.5 Skeletons, Medial Axes, and Distance Transforms

Distance Transform



a b
c d
e f

FIGURE 11.13

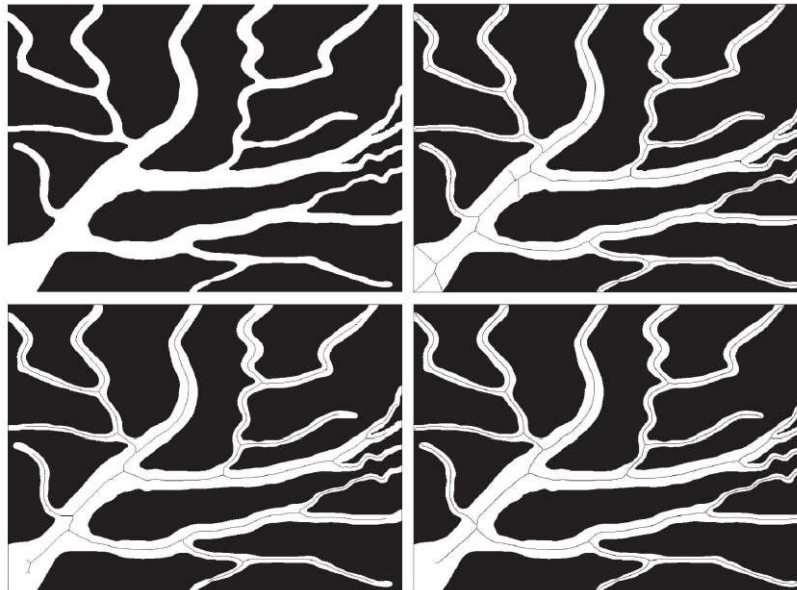
(a) A small image and (b) its distance transform. Note that all 1-valued pixels in (a) have corresponding 0's in (b). (c) A small image, and (d) the distance transform of its complement. (e) A larger image, and (f) the distance transform of its complement. The Euclidian distance was used throughout.

- The distance transform of a region of foreground pixels in a background of zeros is the distance from every pixel to the nearest nonzero valued pixel. For MAT, we compute the distance transform of the *complement* of the image.



11.2 Boundary Preprocessing

11.2.5 Skeletons, Medial Axes, and Distance Transforms



a b
c d

FIGURE 11.14

(a) Thresholded image of blood vessels.

(b) Skeleton obtained by thinning, shown superimposed on the image (note the spurs).

(c) Result of 40 passes of spur removal.

(d) Skeleton obtained using the distance transform.



11.3 Boundary Feature Descriptors

11.3.1 Some Basic Boundary Descriptors

- Diameter

$$\text{diameter}(B) = \max_{i,j} [D(p_i, p_j)] \quad (11 - 4)$$

- Length

$$\text{length}_m = [(x_2 - x_1)^2 + (y_2 - y_1)^2]^{1/2} \quad (11 - 5)$$

$$\text{angle}_m = \tan^{-1} \left[\frac{y_2 - y_1}{x_2 - x_1} \right]$$

- Major and minor axis
- Eccentricity
- Basic rectangle (bounding box)
- Curvature
- Convex and concave
- Tortuosity

$$\tau = \sum_{i=1}^n |a_i| \quad (11 - 6)$$



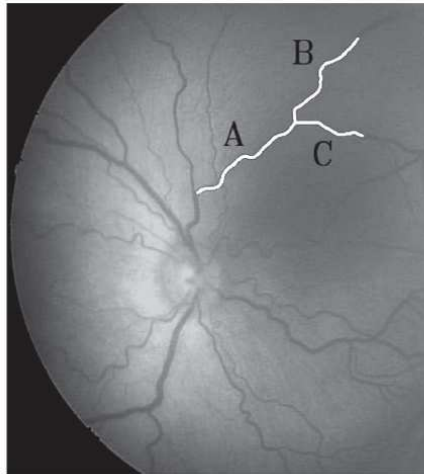
11.3 Boundary Feature Descriptors

11.3.1 Some Basic Boundary Descriptors

a b

FIGURE 11.15

(a) Fundus image from a prematurely born baby with ROP.
(b) Tortuosity of vessels A, B, and C.
(Courtesy of Professor Ernesto Bribiesca, IIMAS-UNAM, Mexico.)



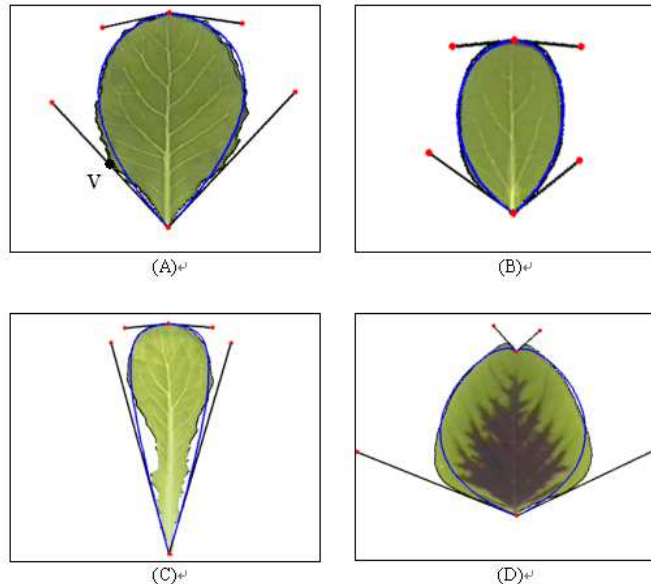
Curve	n	τ
A	50	2.3770
B	50	2.5132
C	50	1.6285



11.3 Boundary Feature Descriptors

11.3.1 Some Basic Boundary Descriptors

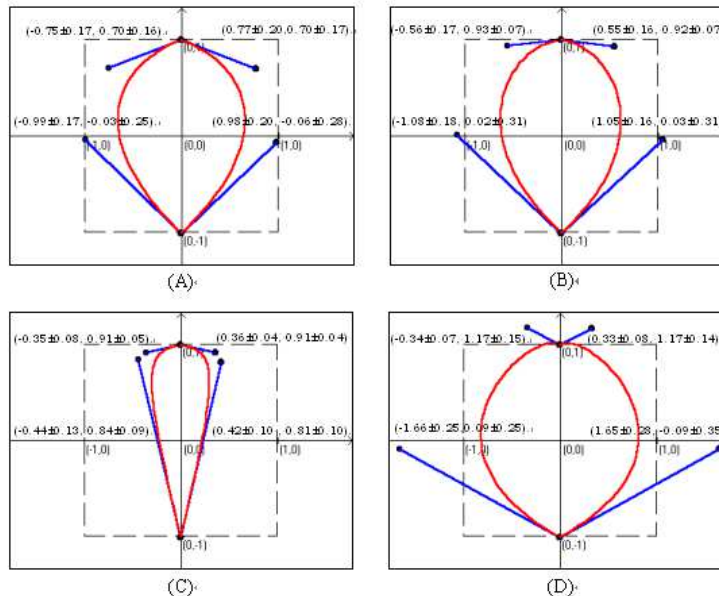
- Boundary descriptors for plant leaves



11.3 Boundary Feature Descriptors

11.3.1 Some Basic Boundary Descriptors

- Boundary descriptors for plant leaves



11.3 Boundary Feature Descriptors

11.3.2 Shape Numbers

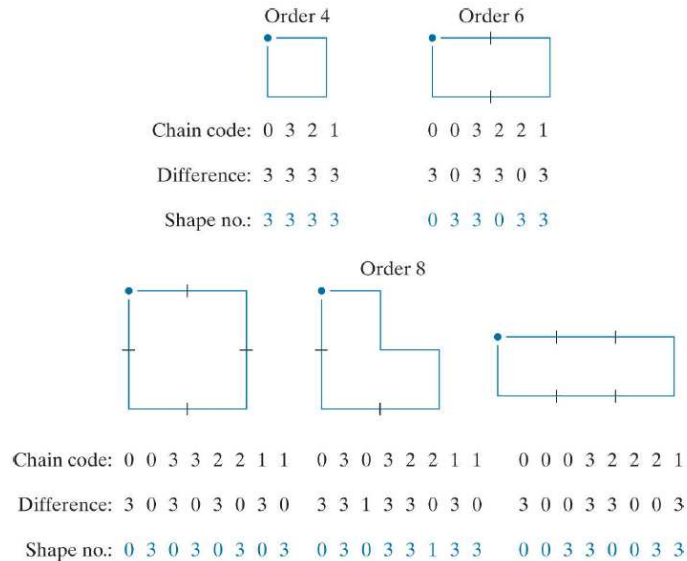
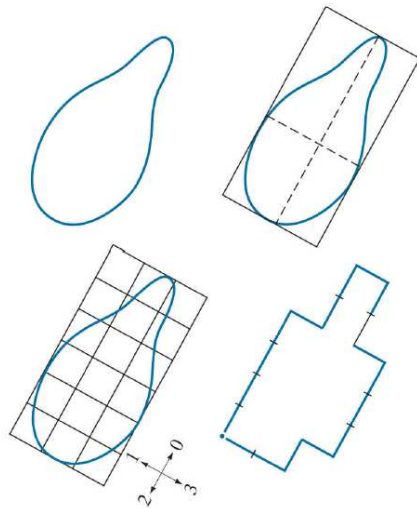


FIGURE 11.16
All shapes of order 4, 6, and 8. The directions are from Fig. 11.3(a), and the dot indicates the starting point.



11.3 Boundary Feature Descriptors

11.3.2 Shape Numbers



Chain code: 0 0 0 0 3 0 0 3 2 2 3 2 2 1 2 1 1

Difference: 3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

Shape no.: 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0 3

a b
c d

FIGURE 11.17
Steps in the
generation of a
shape number.



11.3 Boundary Feature Descriptors

11.3.3 Fourier Descriptors

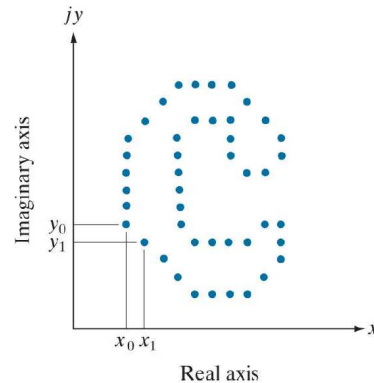
$$s(k) = x(k) + jy(k) \quad (11-7)$$

$$a(u) = \frac{1}{K} \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K} \quad (11-8)$$

$$s(k) = \sum_{u=0}^{K-1} a(u) e^{j2\pi uk/K} \quad (11-9)$$

$$\hat{s}(k) = \sum_{u=0}^{P-1} a(u) e^{j2\pi uk/K} \quad (11-10)$$

FIGURE 11.18
A digital
boundary and its
representation
as sequence of
complex numbers.
The points (x_0, y_0)
and (x_1, y_1) are
(arbitrarily)
the first two points in
the sequence.



11.3 Boundary Feature Descriptors

11.3.3 Fourier Descriptors

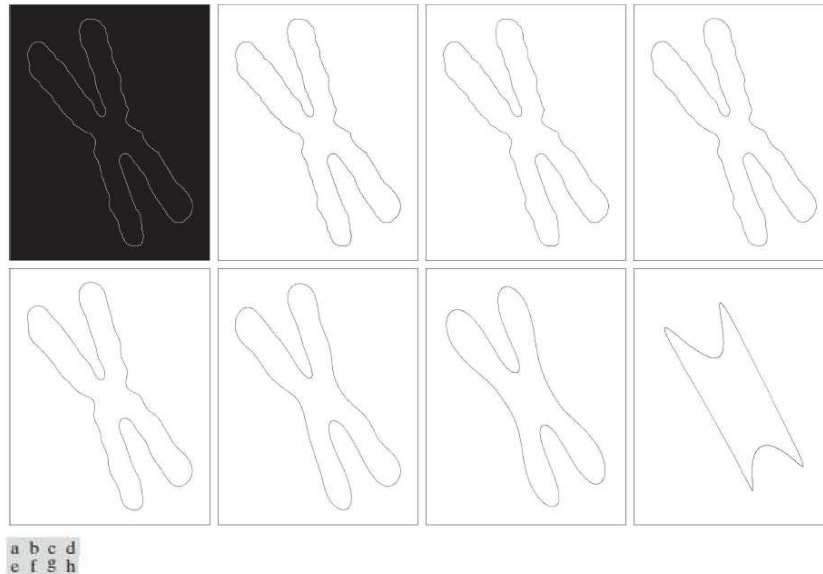
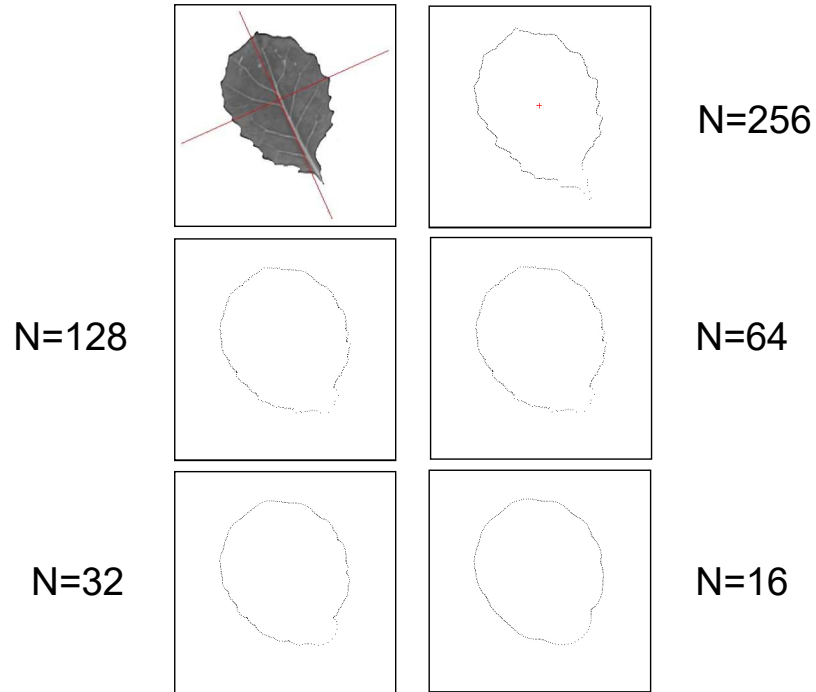


FIGURE 11.19 (a) Boundary of a human chromosome (2868 points). (b)–(h) Boundaries reconstructed using 1434, 286, 144, 72, 36, 18, and 8 Fourier descriptors, respectively. These numbers are approximately 50%, 10%, 5%, 2.5%, 1.25%, 0.63%, and 0.28% of 2868, respectively. Images (b)–(h) are shown as negatives to make the boundaries easier to see.



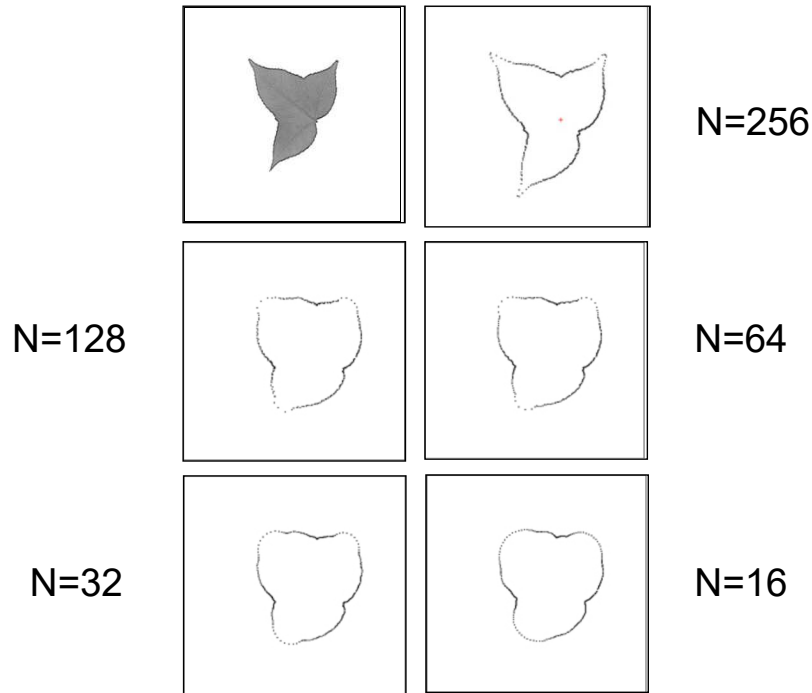
11.3 Boundary Feature Descriptors

11.3.3 Fourier Descriptors



11.3 Boundary Feature Descriptors

11.3.3 Fourier Descriptors



11.3 Boundary Feature Descriptors

11.3.3 Fourier Descriptors

$$a_r(u) = \frac{1}{K} \sum_{k=0}^{K-1} s(k) e^{j\theta} e^{-j2\pi uk/K} = a(u) e^{j\theta} \quad (11-11)$$

$$s_t(k) = [x(k) + \Delta x] + j[y(k) + \Delta y] \quad (11-12)$$

$$s_p(k) = x(k - k_0) + jy(k - k_0) \quad (11-13)$$

TABLE 12.1
Some basic
properties of
Fourier
descriptors.

Transformation	Boundary	Fourier Descriptor
Identity	$s(k)$	$a(u)$
Rotation	$s_r(k) = s(k) e^{j\theta}$	$a_r(u) = a(u) e^{j\theta}$
Translation	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy} \delta(u)$
Scaling	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Starting point	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u) e^{-j2\pi k_0 u/K}$



11.3 Boundary Feature Descriptors

11.3.4 Statistical Moments

$$\mu_n(v) = \sum_{i=1}^{A-1} (z_i - m)^n p(z_i) \quad (11-14)$$

$$m = \sum_{i=1}^{A-1} z_i p(z_i) \quad (11-15)$$

Normalize the area of $g(r)$

$$\mu_n(r) = \sum_{i=1}^{K-1} (r_i - m)^n g(r_i) \quad (11-16)$$

$$m = \sum_{i=1}^{K-1} r_i g(r_i) \quad (11-17)$$

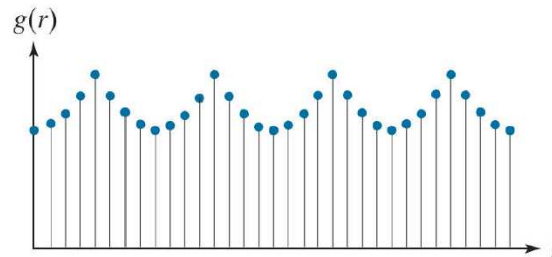


FIGURE 11.20
Sampled
signature from
Fig. 11.10(b) treat-
ed as an ordinary,
discrete function
of one variable.



11.4 Region Feature Descriptors

11.4.1 Some Basic Descriptors

- Compactness

$$compactness = \frac{p^2}{A} \quad (11 - 18)$$

- Circularity (roundness)

$$circularity = \frac{4\pi A}{p^2} \quad (11 - 19)$$

- Effective diameter

$$d_e = 2\sqrt{\frac{A}{\pi}} \quad (11 - 20)$$



11.4 Region Feature Descriptors

11.4.1 Some Basic Descriptors

■ Eccentricity

$$\text{eccentricity} = \frac{c}{a} = \frac{\sqrt{a^2 - b^2}}{a} = \sqrt{1 - \left(\frac{b}{a}\right)^2} \quad a \geq b$$

$$\mathbf{C} = \frac{1}{K-1} \sum_{k=1}^K (\mathbf{z}_k - \bar{\mathbf{z}})(\mathbf{z}_k - \bar{\mathbf{z}})^T \quad (11-21)$$

$$\bar{\mathbf{z}} = \frac{1}{K-1} \sum_{k=1}^K \mathbf{z}_k \quad (11-22)$$

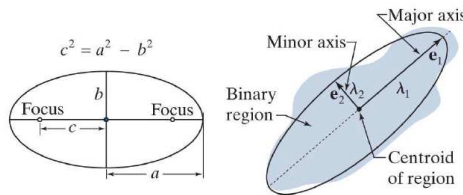
$$\text{eccentricity} = \frac{\sqrt{\lambda_2^2 - \lambda_1^2}}{\lambda_2} = \sqrt{1 - \left(\frac{\lambda_1}{\lambda_2}\right)^2} \quad \lambda_1 \geq \lambda_2 \quad (11-23)$$

a b

FIGURE 11.21

(a) An ellipse in standard form.

(b) An ellipse approximating a region in arbitrary orientation.







\mathbf{e}_1, λ_1 and \mathbf{e}_2, λ_2 are the eigenvectors and corresponding eigenvalues of the covariance matrix of the coordinates of the region



11.4 Region Feature Descriptors

11.4.1 Some Basic Descriptors

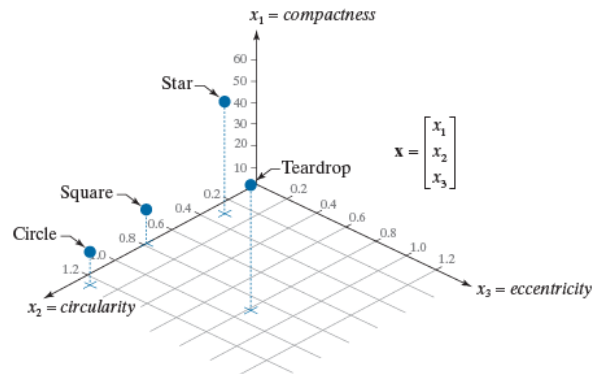
Descriptor				
<i>Compactness</i>	10.1701	42.2442	15.9836	13.2308
<i>Circularity</i>	1.2356	0.2975	0.7862	0.9478
<i>Eccentricity</i>	0.0411	0.0636	0	0.8117

a b c d

FIGURE 11.22
Compactness, circularity, and eccentricity of some simple binary regions.

FIGURE 11.23

The descriptors from Fig. 11.22 in 3-D feature space. Each dot shown corresponds to a feature vector whose components are the three corresponding descriptors in Fig. 11.22.



11.4 Region Feature Descriptors

11.4.1 Some Basic Descriptors

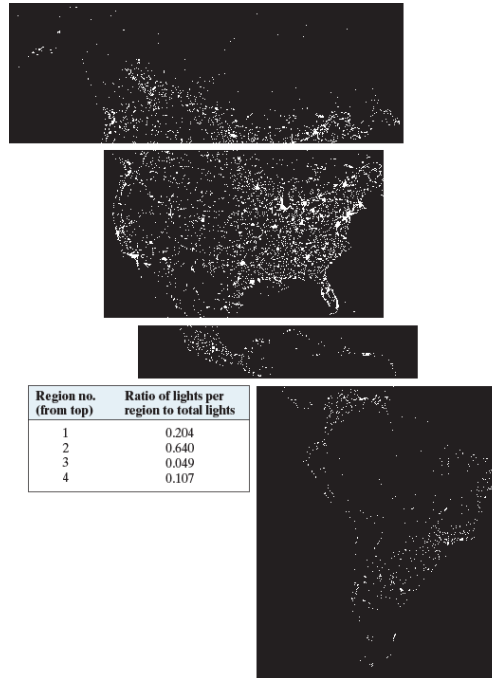


FIGURE 11.24
Infrared images
of the Americas at
night. (Courtesy
of NOAA.)



11.4 Region Feature Descriptors

11.4.2 Topological Descriptors

Euler Number E :

$$E = C - H \quad (11-24)$$

Euler Formula:

$$V - Q + F = C - H \quad (11-25)$$

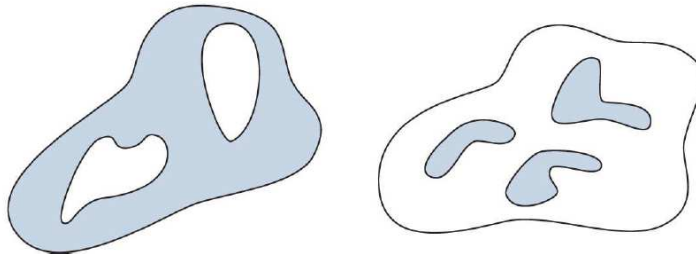
$$V - Q + F = C - H = E \quad (11-26)$$

a b

FIGURE 11.25

(a) A region with two holes.

(b) A region with three connected components.



11.4 Region Feature Descriptors

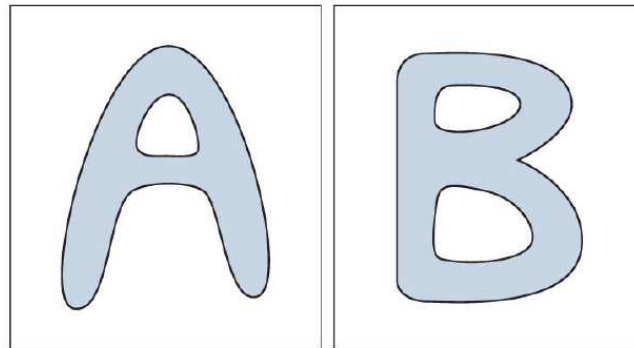
11.4.2 Topological Descriptors

Euler Number E :

$$E = C - H \quad (11-24)$$

a b

FIGURE 11.26
Regions with
Euler numbers
equal to 0 and -1,
respectively.



11.4 Region Feature Descriptors

11.4.2 Topological Descriptors

Euler Formula:

$$V - Q + F = C - H \quad (11-25)$$

$$V - Q + F = C - H = E \quad (11-26)$$

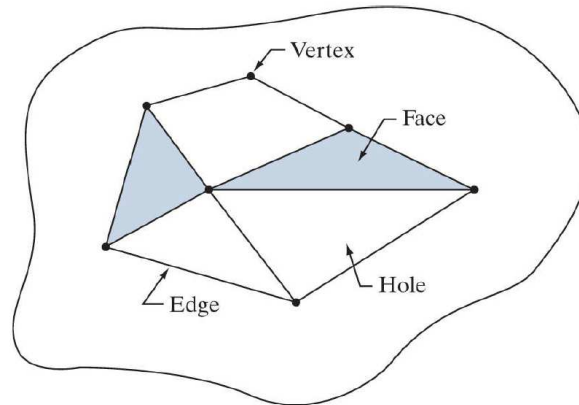
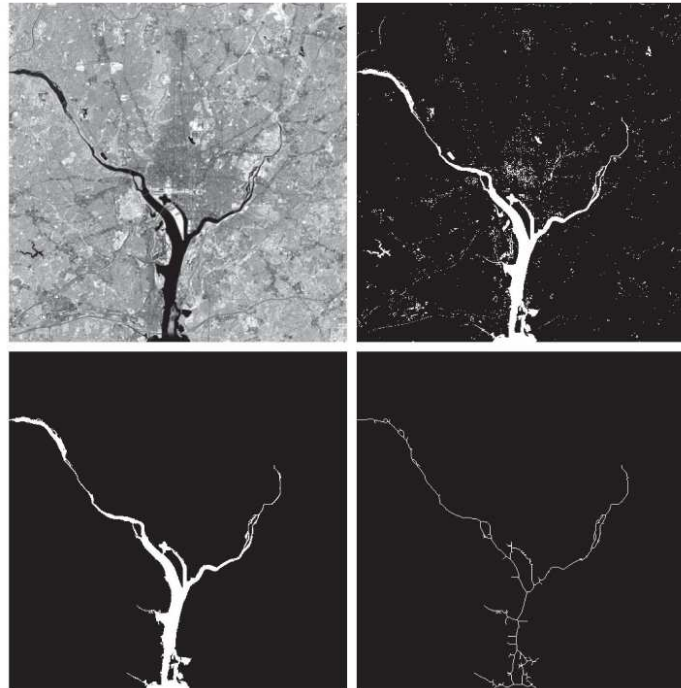


FIGURE 11.27
A region
containing a
polygonal
network.



11.4 Region Feature Descriptors

11.4.2 Topological Descriptors



a b
c d

FIGURE 11.28

(a) Infrared image of the Washington, D.C. area.

(b) Thresholded image.

(c) The largest connected component of (b).

(d) Skeleton of (c). (Original image courtesy of NASA.)



11.4 Region Feature Descriptors

11.4.3 Texture

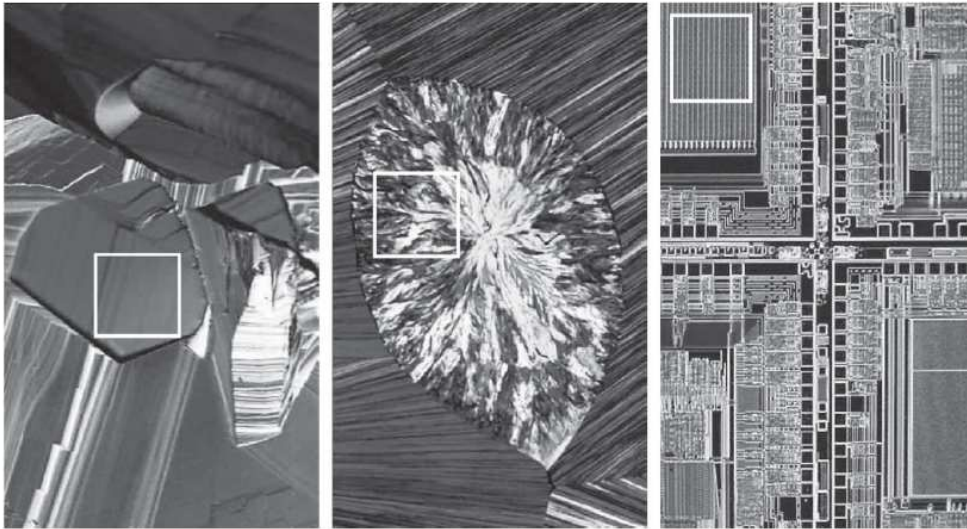


FIGURE 11.29
The white squares mark, from left to right, smooth, coarse, and regular textures. These are optical microscope images of a superconductor, human cholesterol, and a microprocessor. (Courtesy of Dr. Michael W. Davidson, Florida State University.)



11.4 Region Feature Descriptors

11.4.3 Texture

■ Statistical Approaches

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \quad (11-27)$$

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \quad (11-28)$$

$$R = 1 - \frac{1}{1 + \sigma^2(z)} \quad (11-29)$$

$$\mu_0 = 1, \quad \mu_1 = 0, \quad \mu_2(z) = \sigma^2(z),$$

$$\text{skewness} \quad \mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$$

$$\text{uniformity} \quad U = \sum_{i=0}^{L-1} p^2(z_i) \quad (11-30)$$

$$\text{average entropy} \quad e = \sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i) \quad (11-31)$$



11.4 Region Feature Descriptors

11.4.3 Texture

■ Statistical Approaches

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i) \quad (11-27)$$

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \quad (11-28)$$

$$R = 1 - \frac{1}{1 + \sigma^2(z)} \quad (11-29)$$

$$\mu_0 = 1, \quad \mu_1 = 0, \quad \mu_2(z) = \sigma^2(z),$$

$$\text{skewness} \quad \mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$$

$$\text{uniformity} \quad U = \sum_{i=0}^{L-1} p^2(z_i) \quad (11-30)$$

$$\text{average entropy} \quad e = \sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i) \quad (11-31)$$



11.4 Region Feature Descriptors

11.4.3 Texture

■ Statistical Approaches

TABLE 12.2

Statistical texture measures for the subimages in Fig. 12.29.

Texture	Mean	Standard deviation	R (normalized)	3rd moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	-0.105	0.026	5.434
Coarse	143.56	74.63	0.079	-0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

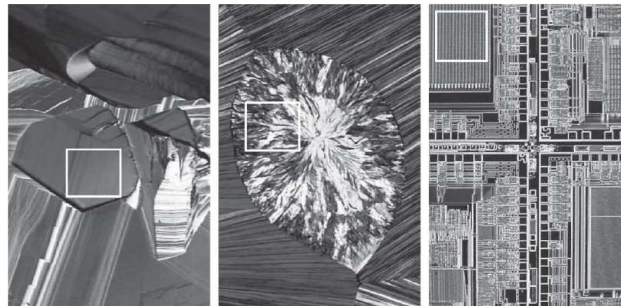


FIGURE 11.29
The white squares mark, from left to right, smooth, coarse, and regular textures. These are optical microscope images of a superconductor, human cholesterol, and a microprocessor. (Courtesy of Dr. Michael W. Davidson, Florida State University.)



11.4 Region Feature Descriptors

11.4.3 Texture

■ Statistical Approaches

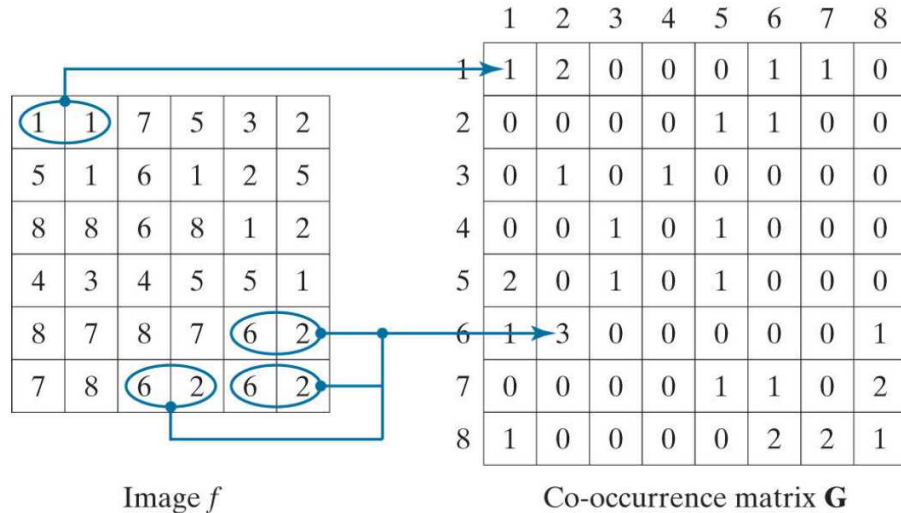


FIGURE 11.30
How to construct
a co-occurrence
matrix.



11.4 Region Feature Descriptors

11.4.3 Texture

■ Statistical Approaches

TABLE 12.3

Descriptors used for characterizing co-occurrence matrices of size $K \times K$. The term p_{ij} is the ij -th term of \mathbf{G} divided by the sum of the elements of \mathbf{G} .

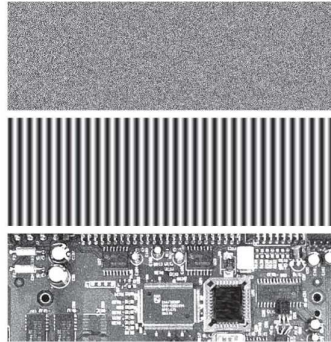
Descriptor	Explanation	Formula
Maximum probability	Measures the strongest response of \mathbf{G} . The range of values is $[0, 1]$.	$\max_{i,j} (p_{ij})$
Correlation	A measure of how correlated a pixel is to its neighbor over the entire image. The range of values is 1 to -1 corresponding to perfect positive and perfect negative correlations. This measure is not defined if either standard deviation is zero.	$\frac{\sum_{i=1}^K \sum_{j=1}^K (i - m_r)(j - m_c) p_{ij}}{\sigma_r \sigma_c}$ $\sigma_r \neq 0; \sigma_c \neq 0$
Contrast	A measure of intensity contrast between a pixel and its neighbor over the entire image. The range of values is 0 (when \mathbf{G} is constant) to $(K-1)^2$.	$\sum_{i=1}^K \sum_{j=1}^K (i - j)^2 p_{ij}$
Uniformity (also called Energy)	A measure of uniformity in the range $[0, 1]$. Uniformity is 1 for a constant image.	$\sum_{i=1}^K \sum_{j=1}^K p_{ij}^2$
Homogeneity	Measures the spatial closeness to the diagonal of the distribution of elements in \mathbf{G} . The range of values is $[0, 1]$, with the maximum being achieved when \mathbf{G} is a diagonal matrix.	$\sum_{i=1}^K \sum_{j=1}^K \frac{p_{ij}}{1 + i - j }$
Entropy	Measures the randomness of the elements of \mathbf{G} . The entropy is 0 when all p_{ij} 's are 0, and is maximum when the p_{ij} 's are uniformly distributed. The maximum value is thus $2\log_2 K$.	$-\sum_{i=1}^K \sum_{j=1}^K p_{ij} \log_2 p_{ij}$



11.4 Region Feature Descriptors

11.4.3 Texture

■ Statistical Approaches

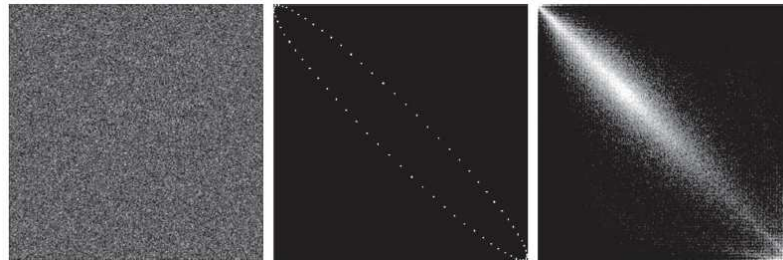


a
b
c

FIGURE 11.31
Images whose pixels have
(a) random,
(b) periodic, and
(c) mixed texture patterns. Each image is of size 263×800 pixels.

a b c

FIGURE 11.32
 256×256
co-occurrence matrices G_1 , G_2 , and G_3 , corresponding from left to right to the images in Fig. 11.31.



11.4 Region Feature Descriptors

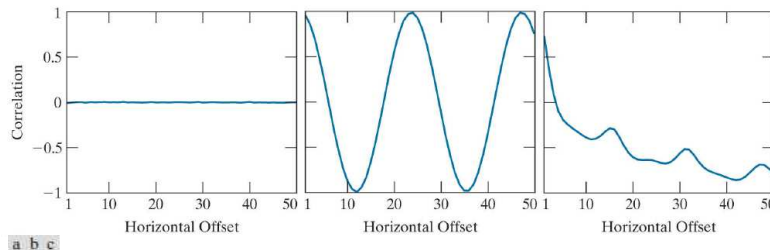
11.4.3 Texture

■ Statistical Approaches

TABLE 12.4

Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 12.32.

Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
G_1/n_1	0.00006	-0.0005	10838	0.00002	0.0366	15.75
G_2/n_2	0.01500	0.9650	00570	0.01230	0.0824	06.43
G_3/n_3	0.06860	0.8798	01356	0.00480	0.2048	13.58



a b c

FIGURE 11.33 Values of the correlation descriptor as a function of offset (distance between “adjacent” pixels) corresponding to the (a) noisy, (b) sinusoidal, and (c) circuit board images in Fig. 11.31.

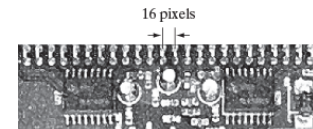


FIGURE 11.34

A zoomed section of the circuit board image showing periodicity of components.



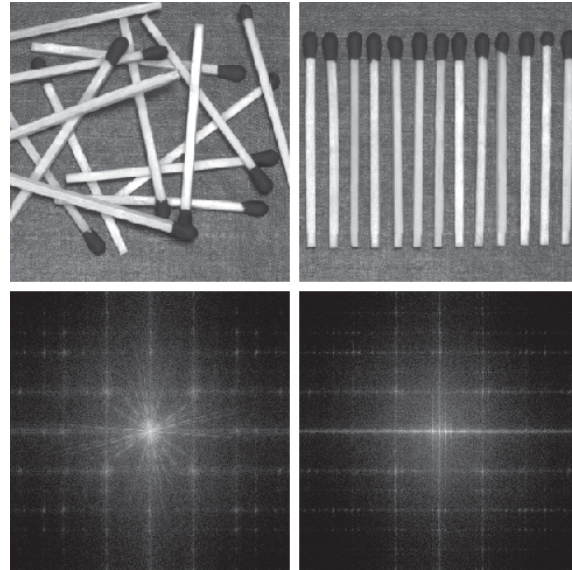
11.4 Region Feature Descriptors

11.4.3 Texture

■ Spectral Approaches

$$S(r) = \sum_{\theta=0}^{\pi} S_{\theta}(r) \quad (11-32)$$

$$S(\theta) = \sum_{r=1}^{R_0} S_r(\theta) \quad (11-33)$$



a b
c d

FIGURE 11.35

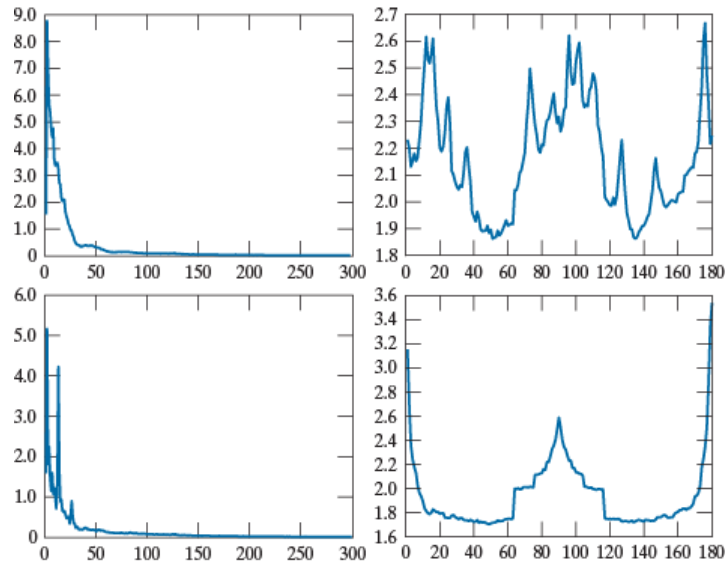
(a) and (b) Images of random and ordered objects. (c) and (d) Corresponding Fourier spectra. All images are of size 600×600 pixels.



11.4 Region Feature Descriptors

11.4.3 Texture

■ Spectral Approaches



a b
c d

FIGURE 11.36
(a) and (b) Plots of $S(r)$ and $S(\theta)$ for Fig. 11.35(a).
(c) and (d) Plots of $S(r)$ and $S(\theta)$ for Fig. 11.35(b).
All vertical axes are $\times 10^5$.



11.4 Region Feature Descriptors

11.4.4 Moment Invariants

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (11-34)$$

The central moment of order $(p+q)$ is defined as

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy \quad (11-35)$$

for $p = 0, 1, 2, \dots$ and $q = 0, 1, 2, \dots$, where

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{and} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (11-36)$$

The normalized central moment is defined as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad (11-37)$$

$$\gamma = \frac{p+q}{2} + 1 \quad (11-38)$$



11.4 Region Feature Descriptors

11.4.4 Moment Invariants

A set of seven, 2-D moment invariants can be derived from the second and third normalized central moments

$$\varphi_1 = \eta_{20} + \eta_{02} \quad (11-39)$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (11-40)$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (11-41)$$

$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (11-42)$$

$$\begin{aligned} \varphi_5 = & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (11-43)$$

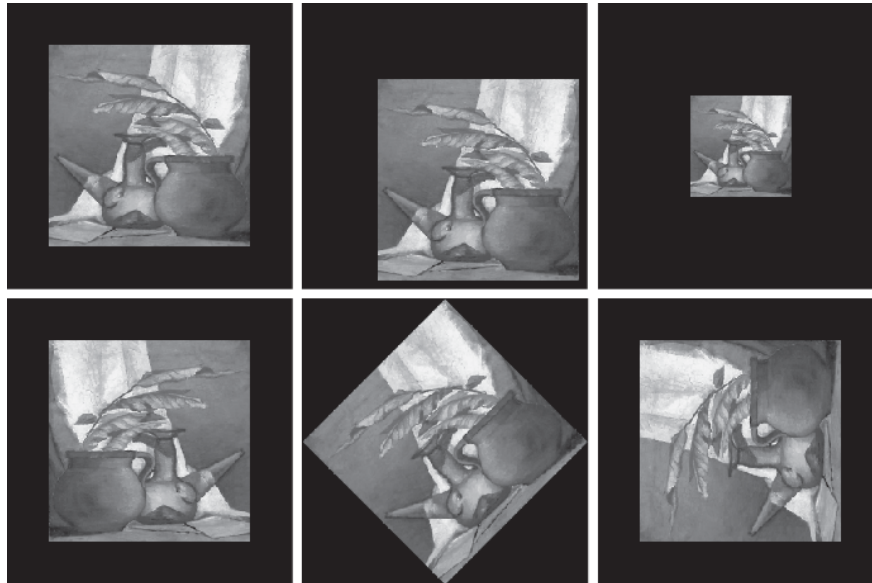
$$\begin{aligned} \varphi_6 = & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \quad (11-44)$$

$$\begin{aligned} \varphi_7 = & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \quad (11-45)$$



11.4 Region Feature Descriptors

11.4.4 Moment Invariants



a b c
d e f

FIGURE 11.37 (a) Original image. (b)–(f) Images translated, scaled by one-half, mirrored, rotated by 45°, and rotated by 90°, respectively.



11.4 Region Feature Descriptors

11.4.4 Moment Invariants

TABLE 12.5

Moment invariants for the images in Fig. 12.37.

Moment Invariant	Original Image	Translated	Half Size	Mirrored	Rotated 45°	Rotated 90°
ϕ_1	2.8662	2.8662	2.8664	2.8662	2.8661	2.8662
ϕ_2	7.1265	7.1265	7.1257	7.1265	7.1266	7.1265
ϕ_3	10.4109	10.4109	10.4047	10.4109	10.4115	10.4109
ϕ_4	10.3742	10.3742	10.3719	10.3742	10.3742	10.3742
ϕ_5	21.3674	21.3674	21.3924	21.3674	21.3663	21.3674
ϕ_6	13.9417	13.9417	13.9383	13.9417	13.9417	13.9417
ϕ_7	-20.7809	-20.7809	-20.7724	20.7809	-20.7813	-20.7809



11.5 Principal Components as Feature Descriptors

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (11-46)$$

Mean Vector

$$\mathbf{m}_x = E\{\mathbf{x}\} \quad (11-47)$$

Covariance Matrix

$$\mathbf{C}_x = E\{[\mathbf{x} - \mathbf{m}_x][\mathbf{x} - \mathbf{m}_x]^T\} \quad (11-48)$$

$$\mathbf{m}_x = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k$$

$$\mathbf{C}_x = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k \mathbf{x}_k^T - \mathbf{m}_x \mathbf{m}_x^T$$



11.5 Principal Components as Feature Descriptors

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x) \quad (11-49)$$

$$\mathbf{m}_y = E\{\mathbf{y}\} = 0 \quad (11-50)$$

$$\mathbf{C}_y = \mathbf{A}\mathbf{C}_x\mathbf{A}^T \quad (11-51)$$

$$\mathbf{C}_y = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix} \quad (11-52)$$

$$\mathbf{x} = \mathbf{A}^T \mathbf{y} + \mathbf{m}_x \quad (11-53)$$

$$\hat{\mathbf{x}} = \mathbf{A}_k^T \mathbf{y} + \mathbf{m}_x \quad (11-54)$$

$$e_{ms} = \sum_{j=1}^n \lambda_j - \sum_{j=1}^k \lambda_j = \sum_{j=k+1}^n \lambda_j \quad (11-55)$$



11.5 Principal Components as Feature Descriptors

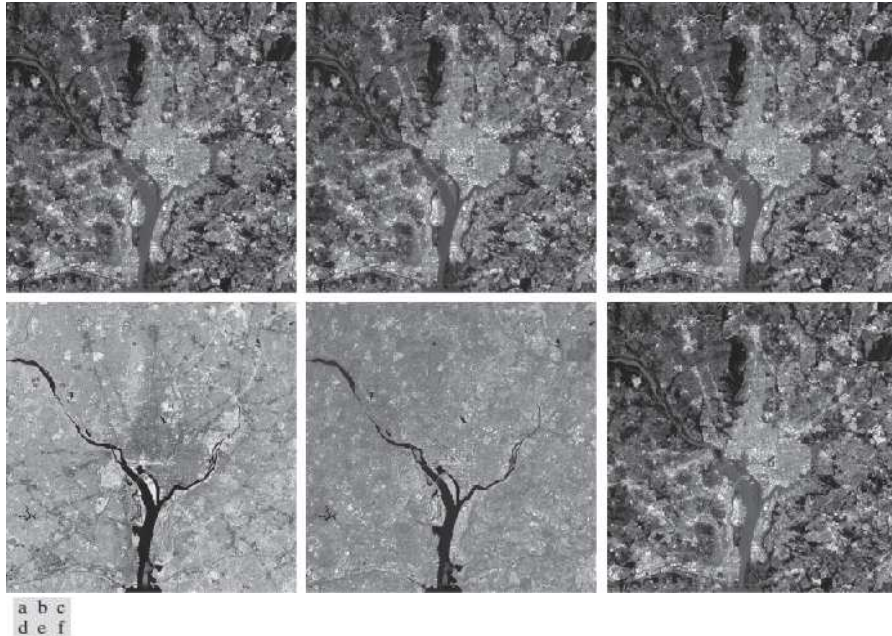


FIGURE 11.38 Multispectral images in the (a) visible blue, (b) visible green, (c) visible red, (d) near infrared, (e) middle infrared, and (f) thermal infrared bands. (Images courtesy of NASA.)



11.5 Principal Components as Feature Descriptors

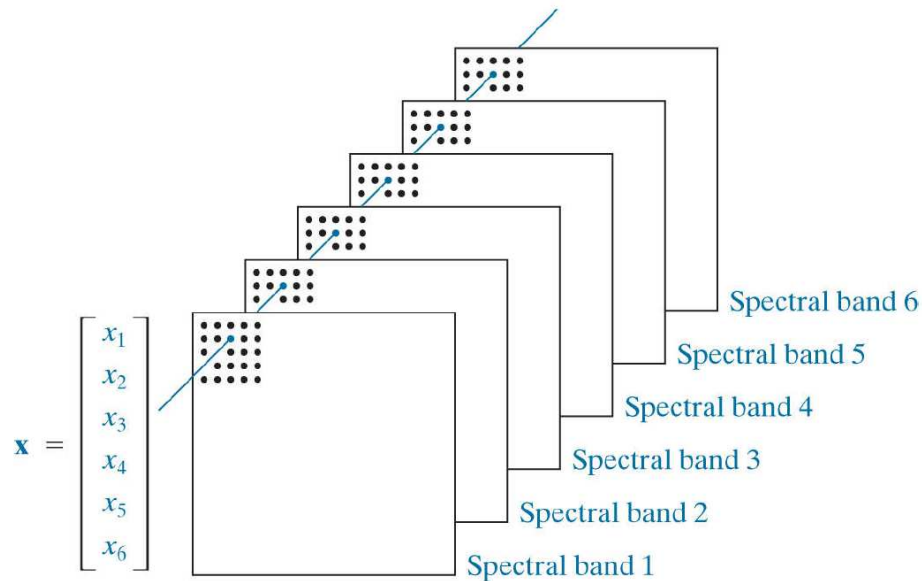


FIGURE 11.39
Forming of a
feature vector from
corresponding
pixels in six images.



11.5 Principal Components as Feature Descriptors

TABLE 12.6

Eigenvalues of C_x obtained from the images in Fig. 12.38.

λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
10344	2966	1401	203	94	31

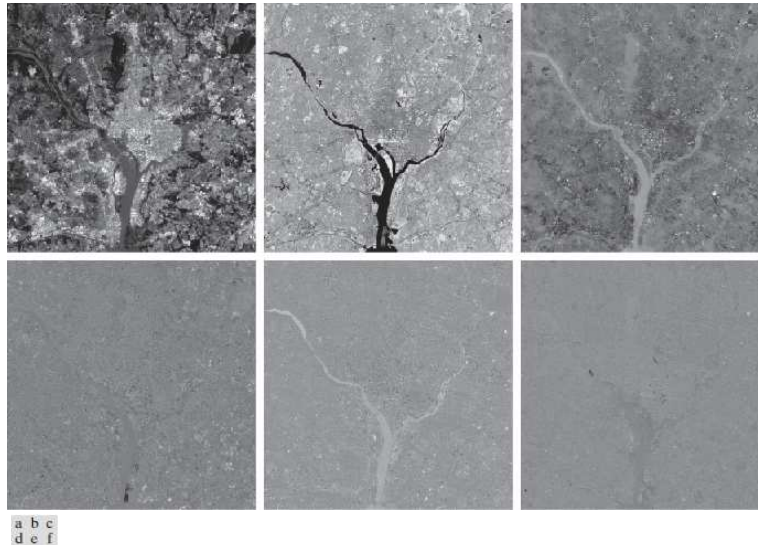


FIGURE 11.40 The six principal component images obtained from vectors computed using Eq. (11-49). Vectors are converted to images by applying Fig. 11.39 in reverse.



11.5 Principal Components as Feature Descriptors

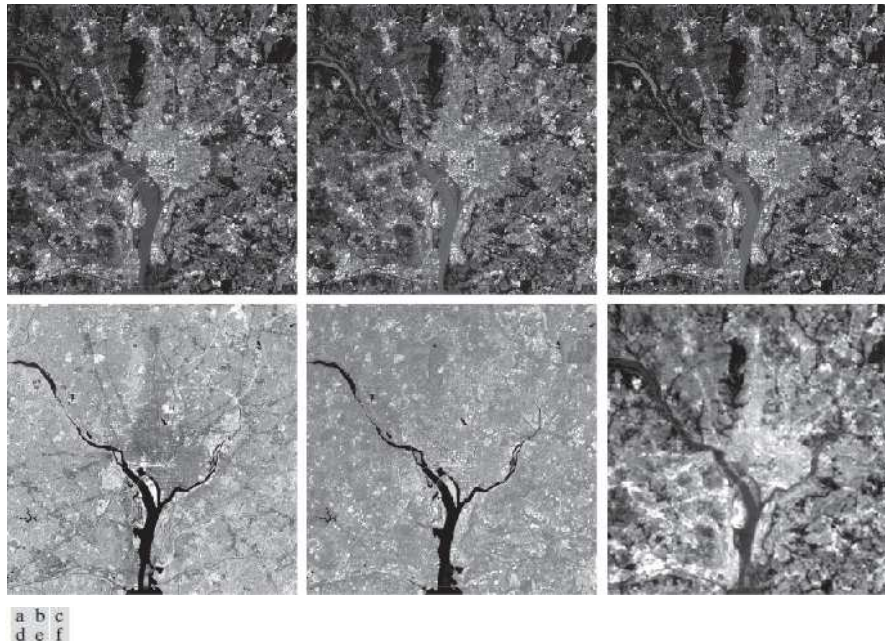
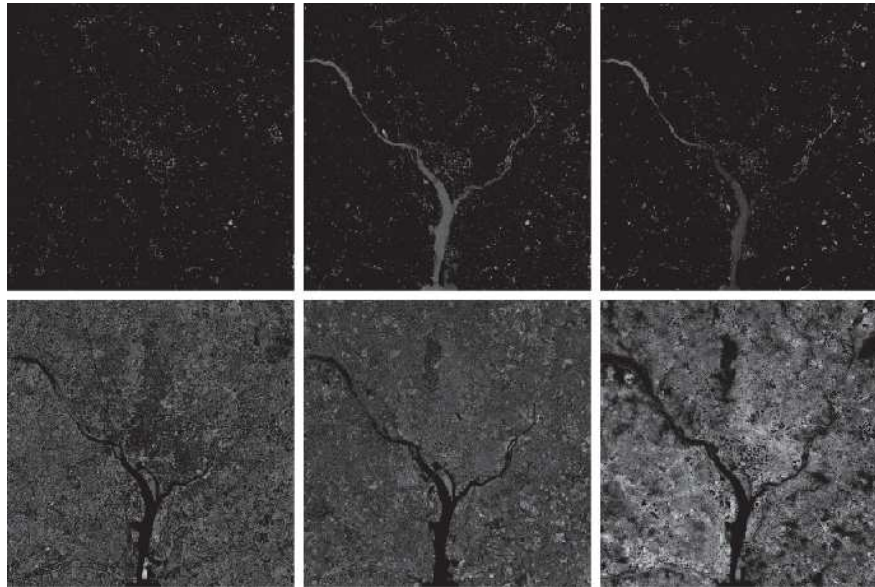


FIGURE 11.41 Multispectral images reconstructed using only the two principal component images corresponding to the two principal component vectors with the largest eigenvalues. Compare these images with the originals in Fig. 11.38.



11.5 Principal Components as Feature Descriptors

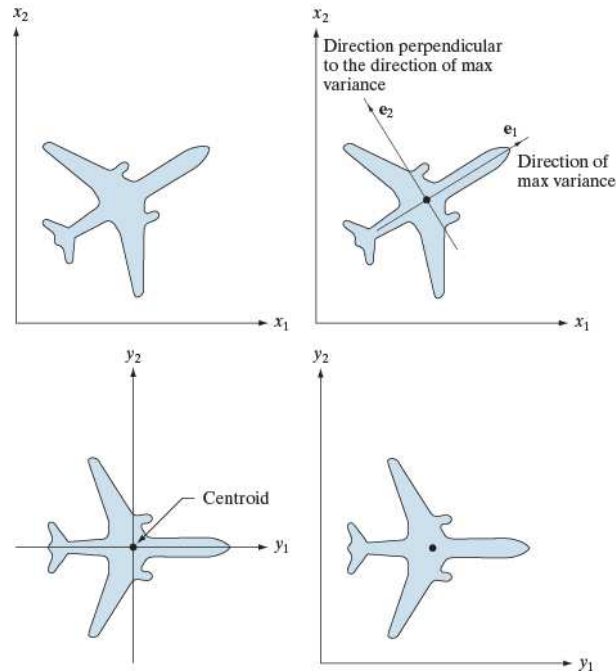


a b c
d e f

FIGURE 11.42 Differences between the original and reconstructed images. All images were enhanced by scaling them to the full $[0, 255]$ range to facilitate visual analysis.



11.5 Principal Components as Feature Descriptors



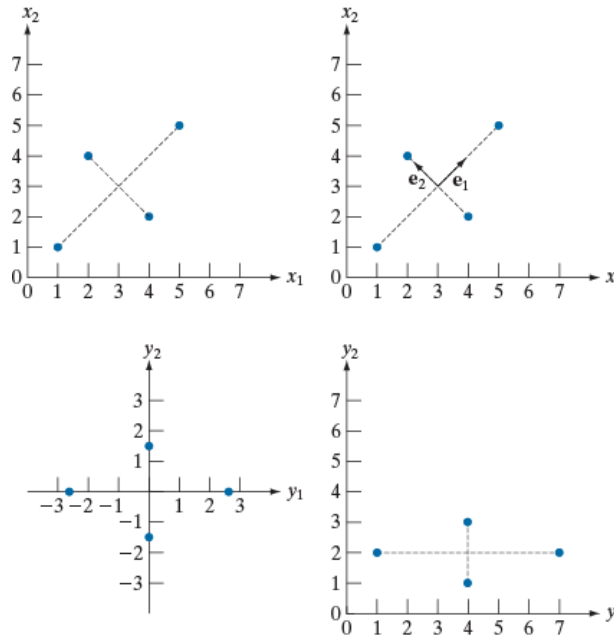
a b
c d

FIGURE 11.43

(a) An object.
(b) Object showing eigenvectors of its covariance matrix.
(c) Transformed object, obtained using Eq. (11-49).
(d) Object translated so that all its coordinate values are greater than 0.



11.5 Principal Components as Feature Descriptors



a b
c d

FIGURE 11.44

A manual example.

(a) Original points.
(b) Eigenvectors of the covariance matrix of the points in (a).

(c) Transformed points obtained using Eq. (11-49).

(d) Points from (c), rounded and translated so that all coordinate values are integers greater than 0. The dashed lines are included to facilitate viewing. They are not part of the data.



11.6 Whole-Image Features

11.6.1 The Harris-Stephens Corner Detector

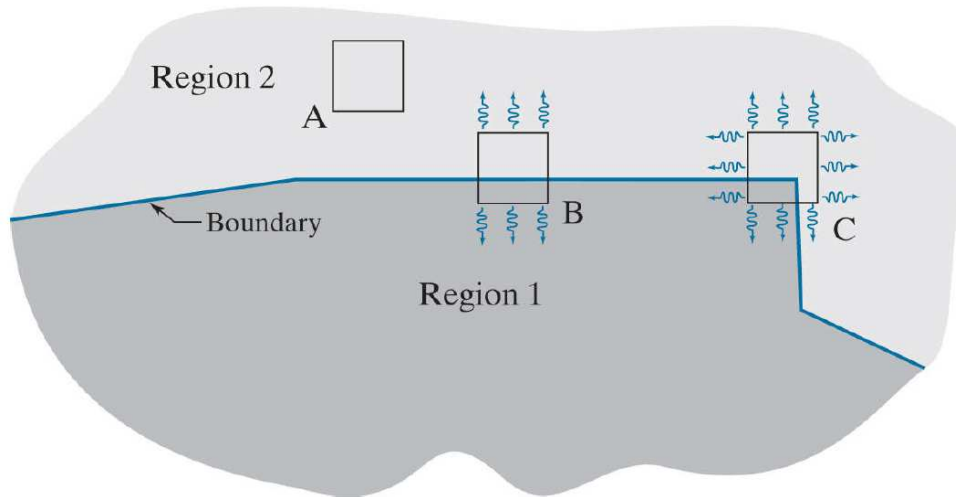


FIGURE 11.45
Illustration of how the Harris-Stephens corner detector operates in the three types of sub-regions indicated by A (flat), B (edge), and C (corner). The wiggly arrows indicate graphically a directional response in the detector as it moves in the three areas shown.



11.6 Whole-Image Features

11.6.1 The Harris-Stephens Corner Detector

$$C(x, y) = \sum_s \sum_t w(s, t) [f(s + x, t + y) - f(s, t)]^2 \quad (11 - 56)$$

$$f(s + x, t + y) \approx f(s, t) + xf_x(s, t) + yf_y(s, t) \quad (11 - 57)$$

$$C(x, y) = \sum_s \sum_t w(s, t) [xf_x(s, t) + yf_y(s, t)]^2 \quad (11 - 58)$$

$$C(x, y) = [x, y] \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix} \quad (11 - 59)$$

$$\mathbf{M} = \sum_s \sum_t w(s, t) \mathbf{A} \quad (11 - 60)$$

$$\mathbf{A} = \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix} \quad (11 - 61)$$

$$w(s, t) = e^{-(s^2 + t^2)/2\sigma^2} \quad (11 - 62)$$

Matrix \mathbf{M} sometimes is called the **Harris matrix**.



11.6 Whole-Image Features

11.6.1 The Harris-Stephens Corner Detector

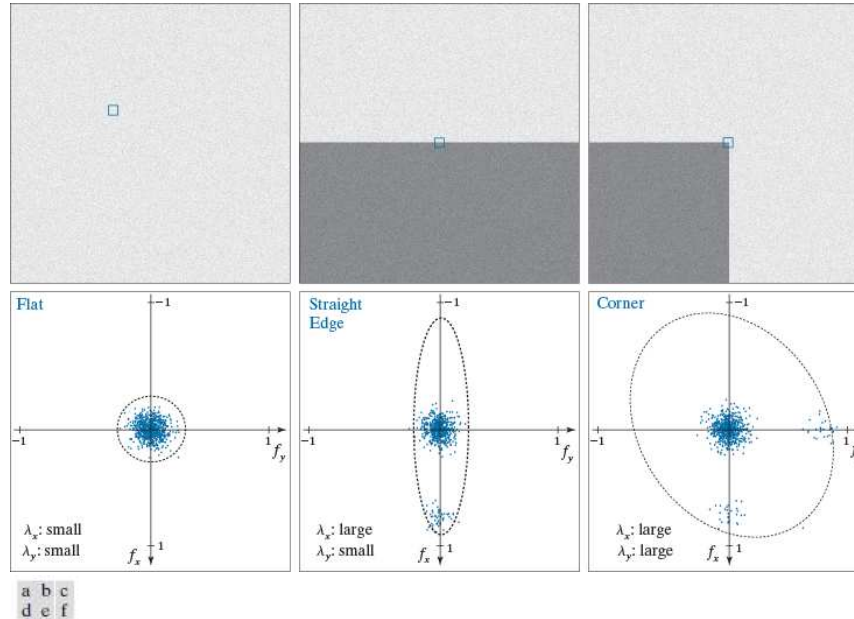


FIGURE 11.46 (a)–(c) Noisy images and image patches (small squares) encompassing image regions similar in content to those in Fig. 11.45. (d)–(f) Plots of value pairs (f_x, f_y) showing the characteristics of the eigenvalues of \mathbf{M} that are useful for detecting the presence of a corner in an image patch.



11.6 Whole-Image Features

11.6.1 The Harris-Stephens Corner Detector

- The HS detector utilizes a measure of corner response based on the fact that the trace of a square matrix is equal to the sum of its eigenvalues, and its determinant is equal to the product of its eigenvalues.

$$\begin{aligned} R &= \lambda_x \lambda_y - k(\lambda_x + \lambda_y)^2 \\ &= \det(\mathbf{M}) - k \text{trace}^2(\mathbf{M}) \end{aligned} \quad (11 - 63)$$

- Measure ***R*** has large positive values when both eigenvalues are large, indicating the presence of a corner.
- Constant *k* is a “sensitivity factor” determined empirically, and its range of values depends on the implementation. For example, the MATLAB Image Processing Toolbox uses $0 < k < 0.25$.



11.6 Whole-Image Features

11.6.1 The Harris-Stephens Corner Detector

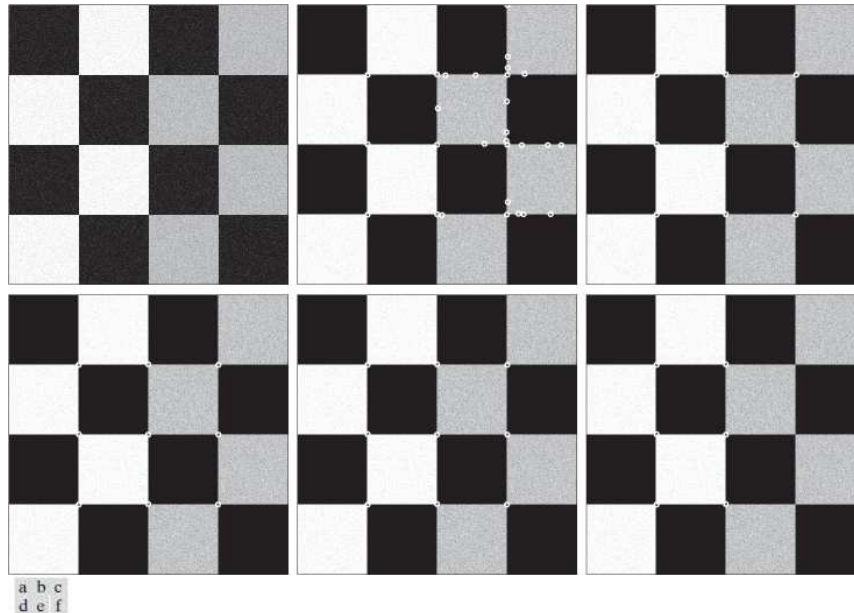
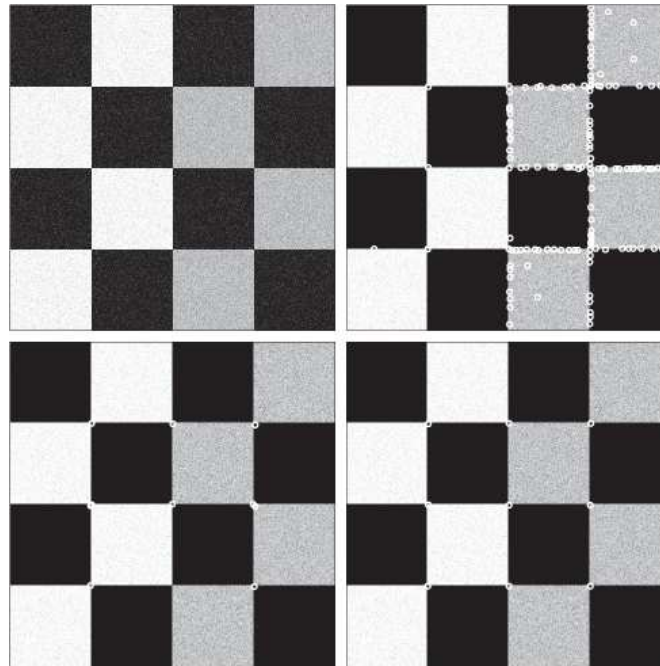


FIGURE 11.47 (a) A 600×600 image with values in the range $[0,1]$, corrupted by additive Gaussian noise with 0 mean and variance of 0.006. (b) Result of applying the HS corner detector with $k = 0.04$ and $T = 0.01$ (the defaults). Several errors are visible. (c) Result using $k = 0.1$ and $T = 0.01$. (d) Result using $k = 0.1$ and $T = 0.1$. (e) Result using $k = 0.04$ and $T = 0.1$. (f) Result using $k = 0.04$ and $T = 0.3$ (only the strongest corners on the left were detected).



11.6 Whole-Image Features

11.6.1 The Harris-Stephens Corner Detector



a b
c d

FIGURE 11.48

(a) Same as Fig. 11.47(a), but corrupted with Gaussian noise of mean 0 and variance 0.01.

(b) Result of using the HS detector with $k = 0.04$ and $T = 0.01$ [compare with Fig. 11.47(b)].

(c) Result with $k = 0.249$, (near the highest value in our implementation), and $T = 0.01$.

(d) Result of using $k = 0.04$ and $T = 0.15$.



11.6 Whole-Image Features

11.6.1 The Harris-Stephens Corner Detector

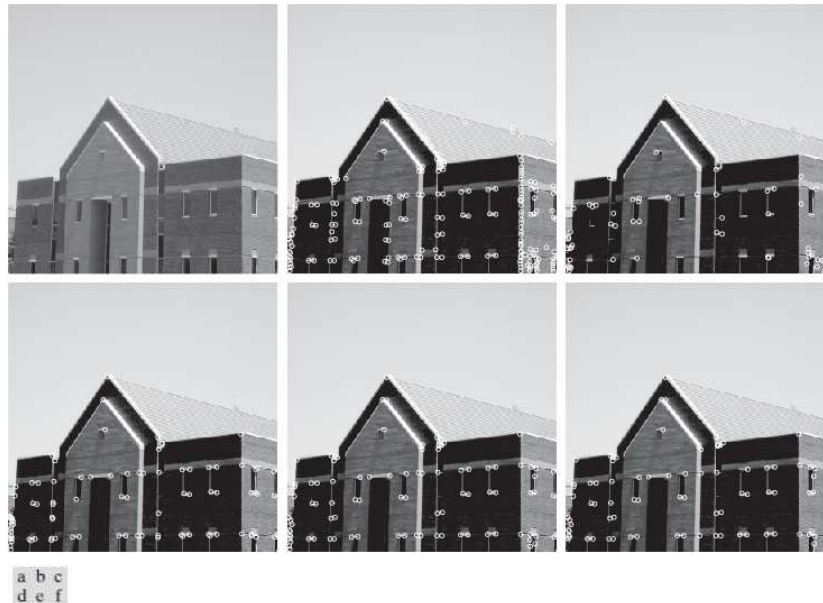
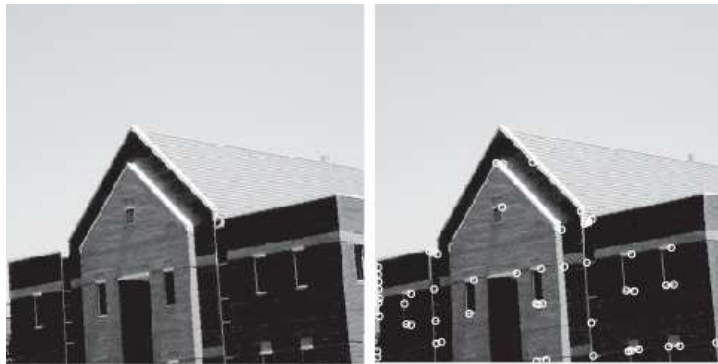


FIGURE 11.49 600 \times 600 image of a building. (b) Result of applying the HS corner detector with $k = 0.04$ and $T = 0.01$ (the default values in our implementation). Numerous irrelevant corners were detected. (c) Result using $k = 0.249$ and the default value for T . (d) Result using $k = 0.17$ and $T = 0.05$. (e) Result using the default value for k and $T = 0.05$. (f) Result using the default value of k and $T = 0.07$.



11.6 Whole-Image Features

11.6.1 The Harris-Stephens Corner Detector



a b

FIGURE 11.50

(a) Image rotated 5°.
(b) Corners detected using the parameters used to obtain Fig. 11.49(f).



11.6 Whole-Image Features

11.6.2 Maximally Stable Extremal Regions (MSERs)

- Extremal regions that do not change size (number of pixels) appreciably over a range of threshold values are called ***maximally stable extremal regions***.
- ***Component tree***

$$\forall p \in R \text{ and } \forall q \in \text{boundary}(R): I(p) > I(q) \quad (11 - 64)$$

- ***Stability measure, ψ***

$$\psi(R_j^{T+n\Delta T}) = \frac{|R_i^{T+(n-1)\Delta T}| - |R_k^{T+(n+1)\Delta T}|}{|R_j^{T+(n)\Delta T}|} \quad (11 - 65)$$

- In practice, the maximally stable regions are regions whose sizes do not change appreciably across two $2\Delta T$, neighboring thresholded images.



11.6 Whole-Image Features

11.6.2 Maximally Stable Extremal Regions (MSERs)

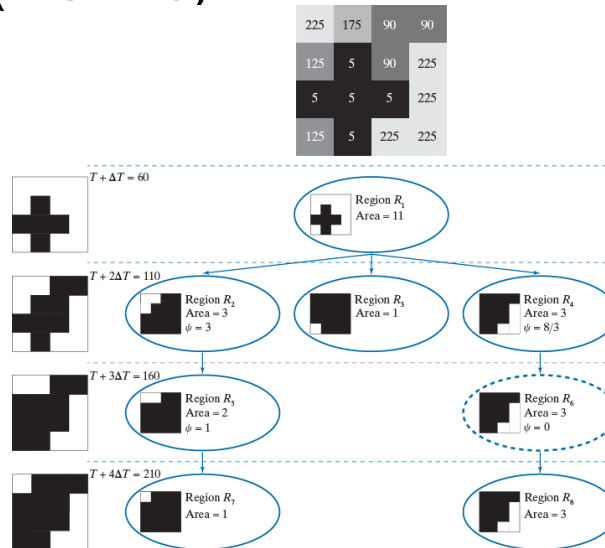
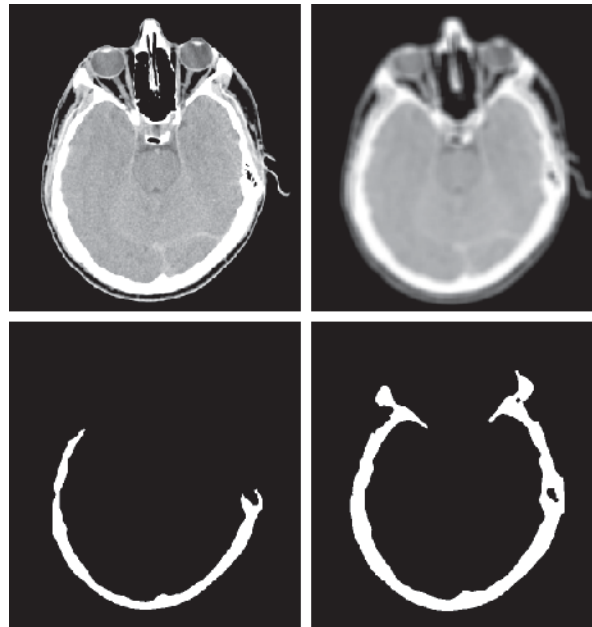


FIGURE 11.51 Detecting MSERs. Top: Grayscale image. Left: Thresholded images using $T = 10$ and $\Delta T = 50$. Right: Component tree, showing the individual regions. Only one MSER was detected (see dashed tree node on the rightmost branch of the tree). Each level of the tree is formed from the thresholded image on the left, at that same level. Each node of the tree contains one extremal region (connected component) shown in white, and denoted by a subscripted R .



11.6 Whole-Image Features

11.6.2 Maximally Stable Extremal Regions (MSERs)



a b
c d

FIGURE 11.52

(a) 600×570 CT slice of a human head. (b) Image smoothed with a box kernel of size 15×15 elements. (c) A extremal region along the path of the tree containing one MSER.

(d) The MSER. (All MSER regions were limited to the range 10,260–34,200 pixels, corresponding to a range between 3% and 10% of image size.)

(Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)



11.6 Whole-Image Features

11.6.2 Maximally Stable Extremal Regions (MSERs)



FIGURE 11.53 (a) Building image of size 600×600 pixels. (b) Image smoothed using a 5×5 box kernel. (c) and (d) MSERs detected using $T = 0$, $\Delta T = 10$, and MSER size range between 10,000 and 30,000 pixels, corresponding approximately to 3% and 8% of the area of the image. (e) Composite image.



11.6 Whole-Image Features

11.6.2 Maximally Stable Extremal Regions (MSERs)



a b c

FIGURE 11.54 (a) Building image rotated 5° counterclockwise. (b) Smoothed image using the same kernel as in Fig. 11.53(b). (c) Composite MSER detected using the same parameters we used to obtain Fig. 11.53(e). The MSERs of the original and rotated images are almost identical.



11.6 Whole-Image Features

11.6.2 Maximally Stable Extremal Regions (MSERs)



FIGURE 11.55 (a) Building image reduced to half-size. (b) Image smoothed with a 3×3 box kernel. (c) Composite MSER obtained with the same parameters as Fig. 11.53(e), but using a valid MSER region size range of 2,500—7,500 pixels.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.1 Scale Space

- The first stage of the **SIFT** algorithm is to find image locations that are invariant to scale change by searching for stable features across all possible scales, using a function of scale known as **scale space**.

$$L(x, y, \sigma) = G(x, y, \sigma) \star f(x, y) \quad (11 - 66)$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (11 - 67)$$

- SIFT subdivides scale space into octaves, with each octave corresponding to a doubling of σ . SIFT further subdivides each octave into an integer number, s , of intervals, so that an interval of 1 consists of two images, an interval of 2 consists of three images, and so forth.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.1 Scale Space

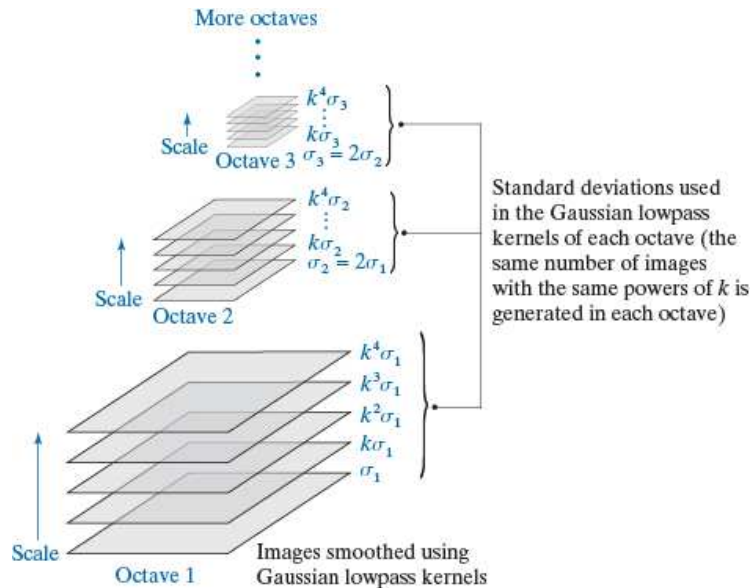


FIGURE 11.56

Scale space, showing three octaves. Because $s = 2$ in this case, each octave has five smoothed images. A Gaussian kernel was used for smoothing, so the space parameter is σ .



11.7.1 Scale Space

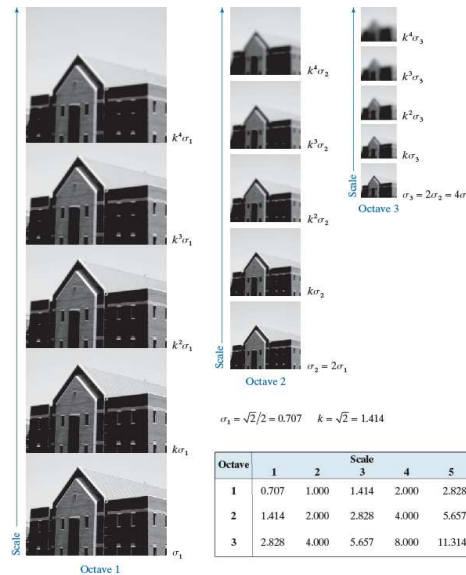


FIGURE 11.57
Illustration using images of the first three octaves of scale space in SIFT. The entries in the table are values of standard deviation used at each scale of each octave. For example the standard deviation used in scale 2 of octave 1 is σ_{k1} , which is equal to 1.0. (The images of octave 1 are shown slightly overlapped to fit in the figure space.)



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.2 Detecting Local Extrema

■ Finding the Initial Keypoints

$$D(x, y, \sigma) = [G(x, y, k\sigma) - G(x, y, \sigma)] \star f(x, y) \quad (11 - 68)$$

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (11 - 69)$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G \quad (11 - 70)$$

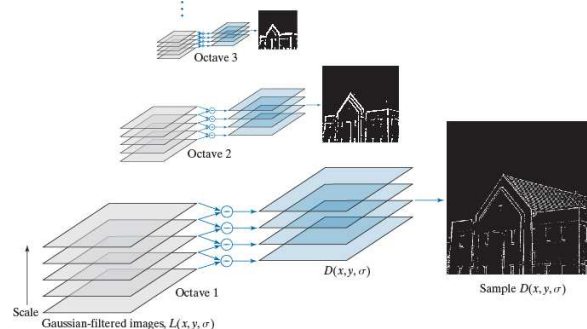


FIGURE 11.58 How Eq. (11-69) is implemented in scale space. There are $s + 3$ $L(x, y, \sigma)$ images and $s + 2$ corresponding $D(x, y, \sigma)$ images in each octave.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.2 Detecting Local Extrema

- Finding the Initial Keypoints

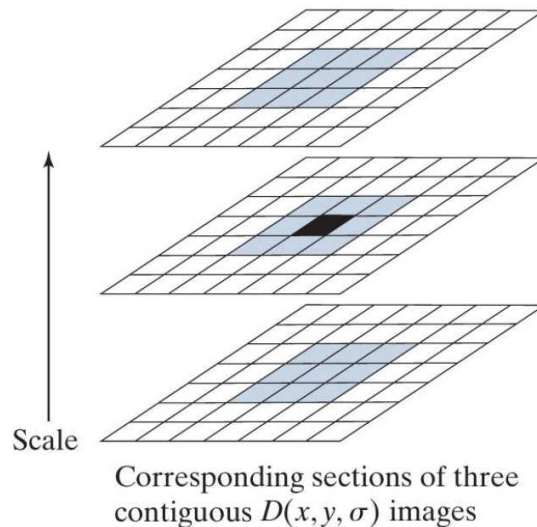


FIGURE 11.59

Extrema (maxima or minima) of the $D(x, y, \sigma)$ images in an octave are detected by comparing a pixel (shown in black) to its 26 neighbors (shown shaded) in 3×3 regions at the current and adjacent scale images.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.2 Detecting Local Extrema

- Improving the Accuracy of Keypoint Locations

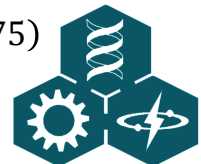
$$\begin{aligned} D(\mathbf{x}) &= D + \left(\frac{\partial D}{\partial \mathbf{x}}\right)^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial}{\partial \mathbf{x}} \left(\frac{\partial D}{\partial \mathbf{x}}\right) \mathbf{x} \\ &= D + (\nabla D)^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} \end{aligned} \quad (11 - 71)$$

$$\nabla D = \frac{\partial D}{\partial \mathbf{x}} = \begin{bmatrix} \partial D / \partial x \\ \partial D / \partial y \\ \partial D / \partial z \end{bmatrix} \quad (11 - 72)$$

$$\mathbf{H} = \begin{bmatrix} \partial^2 D / \partial x^2 & \partial^2 D / \partial x \partial y & \partial^2 D / \partial x \partial \sigma \\ \partial^2 D / \partial y \partial x & \partial^2 D / \partial y^2 & \partial^2 D / \partial y \partial \sigma \\ \partial^2 D / \partial \sigma \partial x & \partial^2 D / \partial \sigma \partial y & \partial^2 D / \partial \sigma^2 \end{bmatrix} \quad (11 - 73)$$

$$\hat{\mathbf{x}} = -\mathbf{H}^{-1}(\nabla D) \quad (11 - 74)$$

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} (\nabla D)^T \hat{\mathbf{x}} \quad (11 - 75)$$



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.2 Detecting Local Extrema

■ Eliminating Edge Responses

$$\mathbf{H} = \begin{bmatrix} \partial^2 D / \partial x^2 & \partial^2 D / \partial x \partial y \\ \partial^2 D / \partial y \partial x & \partial^2 D / \partial y^2 \end{bmatrix} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \quad (11 - 76)$$

$$\text{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta \quad (11 - 77)$$

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$

$$\text{Tr}(\mathbf{H}) = \frac{[\text{Tr}(\mathbf{H})]^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(\gamma\beta + \beta)^2}{\gamma\beta^2} = \frac{(\gamma + 1)^2}{\gamma} \quad (11 - 78)$$

$$\frac{[\text{Tr}(\mathbf{H})]^2}{\text{Det}(\mathbf{H})} < \frac{(\gamma + 1)^2}{\gamma} \quad (11 - 79)$$



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.2 Detecting Local Extrema

- Eliminating Edge Responses



FIGURE 11.60

SIFT keypoints detected in the building image. The points were enlarged slightly to make them easier to see.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.3 Keypoint Orientation

$$M(x, y) = \left[(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2 \right]^{\frac{1}{2}} \quad (11-80)$$

$$\theta(x, y) = \tan^{-1}[(L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))] \quad (11-81)$$



FIGURE 11.61

The keypoints from Fig. 11.60 superimposed on the original image. The arrows indicate keypoint orientations.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.4 Keypoint Descriptors

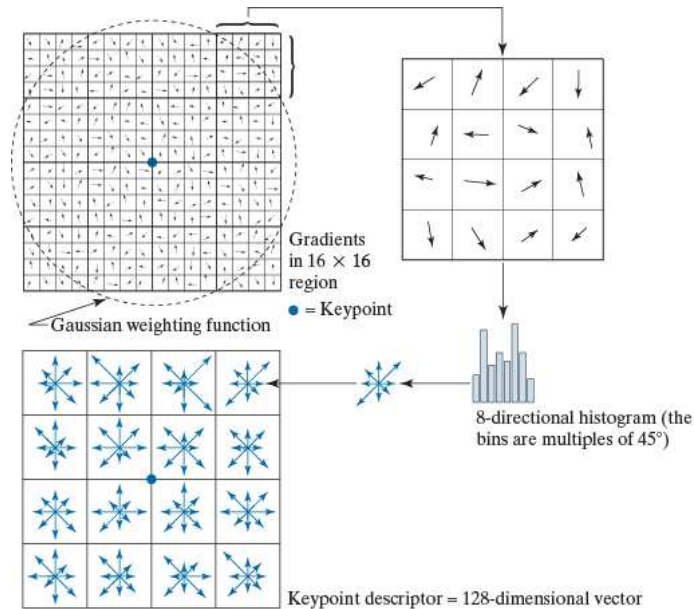


FIGURE 11.62
Approach used to
compute a
keypoint
descriptor.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.5 Summary of the Sift Algorithm

1. **Construct the scale space.** This is done using the procedure outlined in Figs. 12.56 and 12.57 . The parameters that need to be specified are (k is computed from s), and the number of octaves. Suggested values are and three octaves.
2. **Obtain the initial keypoints.** Compute the difference of Gaussians, from the smoothed images in scale space, as explained in Fig. 12.58 and Eq. (12-69) . Find the extrema in each image using the method explained in Fig. 12.59 . These are the initial keypoints.
3. **Improve the accuracy of the location of the keypoints.** Interpolate the values of a via a Taylor expansion. The improved key point locations are given by Eq. (12-74) .
4. **Delete unsuitable keypoints.** Eliminate keypoints that have low contrast and/or are poorly localized. This is done by evaluating D from Step 3 at the improved locations, using Eq. (12-75) . All keypoints whose values of D are lower than a threshold are deleted. A suggested threshold value is 0.03. Keypoints associated with edges are deleted also, using Eq. (12-79) . A value of 10 is suggested for r .
5. **Compute keypoint orientations.** Use Eqs. (12-80) and (12-81) to compute the magnitude and orientation of each keypoint using the histogram-based procedure discussed in connection with these equations.
6. **Compute keypoint descriptors.** Use the method summarized in Fig. 12.62 to compute a feature (descriptor) vector for each keypoint. If a region of size around each keypoint is used, the result will be a 128-dimensional feature vector for each keypoint.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.5 Summary of the Sift Algorithm



FIGURE 11.63 (a) Keypoints and their directions (shown as gray arrows) for the building image and for a section of the right corner of the building. The subimage is a separate image and was processed as such. (b) Corresponding key points between the building and the subimage (the straight lines shown connect pairs of matching points). Only three of the 36 matches found are incorrect.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.5 Summary of the Sift Algorithm

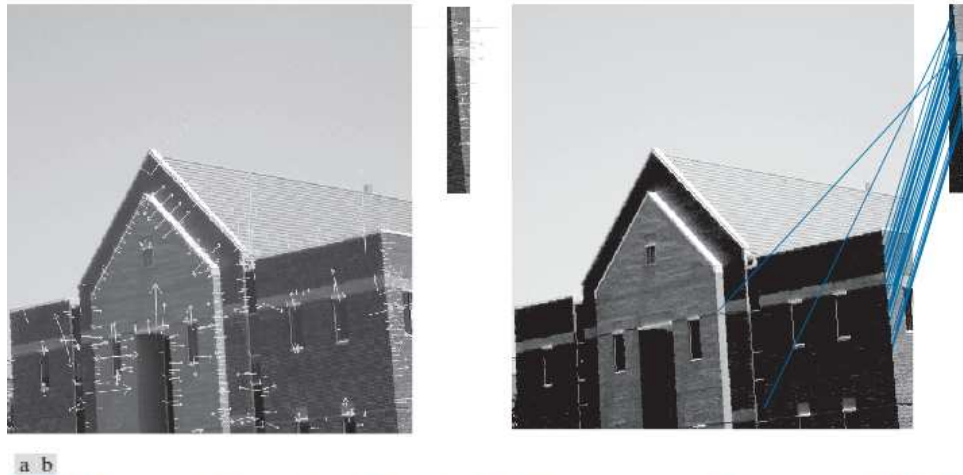


FIGURE 11.64 (a) Keypoints for the rotated (by 5°) building image and for a section of the right corner of the building. The subimage is a separate image and was processed as such. (b) Corresponding keypoints between the corner and the building. Of the 26 matches found, only two are in error.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.5 Summary of the Sift Algorithm

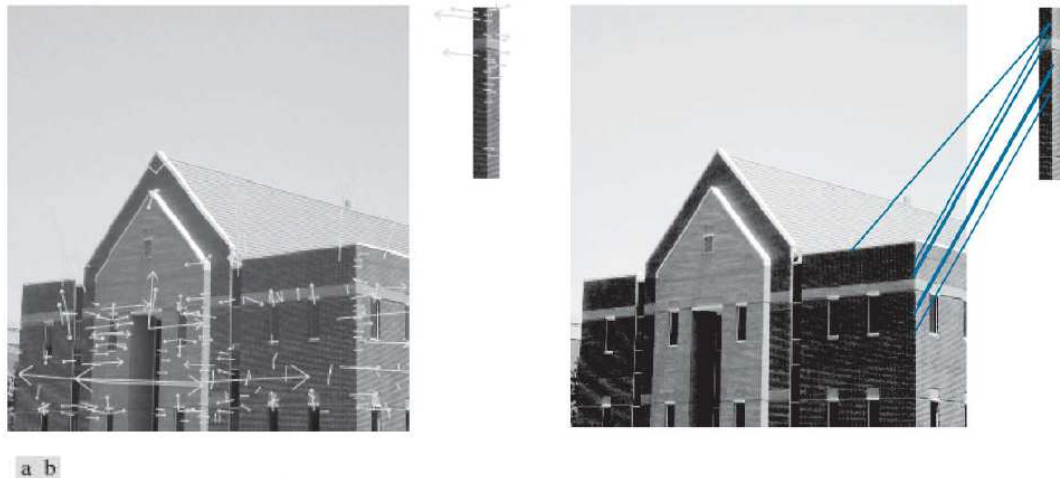


FIGURE 11.65 (a) Keypoints for the half-sized building and a section of the right corner. (b) Corresponding keypoints between the corner and the building. Of the seven matches found, only one is in error.



11.7 Scale-Invariant Feature Transform (SIFT)

11.7.5 Summary of the Sift Algorithm

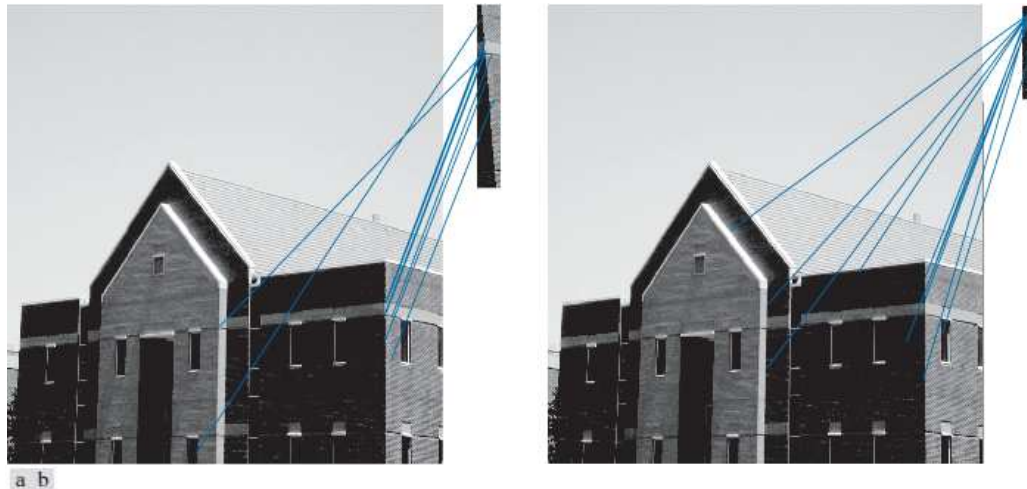


FIGURE 11.66 (a) Matches between the original building image and a rotated version of a segment of its right corner. Ten matches were found, of which two are incorrect. (b) Matches between the original image and a half-scaled version of a segment of its right corner. Here, 11 matches were found, of which four were incorrect.

