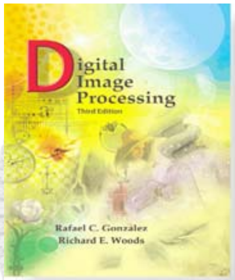


Lecture #23

- Motion segmentation & motion tracking
- Boundary tracking
- Chain codes
- Minimum perimeter polygons
- Signatures

Motion Segmentation

Accumulative Difference Image (ADI) for a moving rectangle.

Positive ADI

Negative ADI

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

The accumulator is incremented if the absolute difference between the reference frame and each successive frame is above a threshold. We can also have positive and negative accumulator images.

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

a b c

FIGURE 10.59 ADIs of a rectangular object moving in a southeasterly direction. (a) Absolute ADI. (b) Positive ADI. (c) Negative ADI.

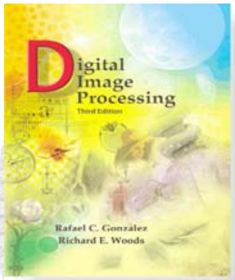
Image Segmentation

a b c

FIGURE 10.60 Building a static reference image. (a) and (b) Two frames in a sequence. (c) Eastbound automobile subtracted from (a) and the background restored from the corresponding area in (b). (Jain and Jain.)

How do you construct a static reference image if something is always is motion?

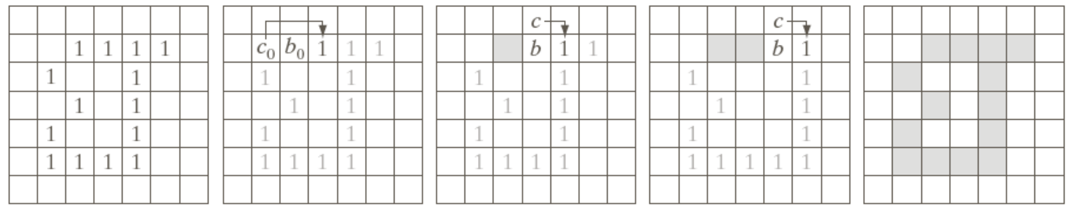
- People Counter



Boundary Tracking

Assumptions:

1. The image is binary where 1=foreground and 0=background
2. The image is padded with a border of 0's so an object cannot merge with the border.
3. There is only one object in this example.



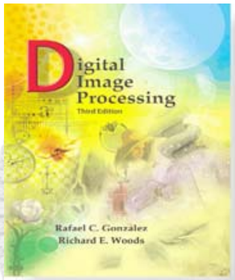
a b c d e

FIGURE 11.1 Illustration of the first few steps in the boundary-following algorithm. The point to be processed next is labeled in black, the points yet to be processed are gray, and the points found by the algorithm are labeled as gray squares.

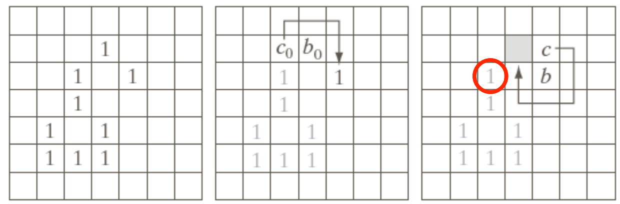
Moore Boundary Tracking Algorithm [1968]:

1. Start with the uppermost, leftmost foreground point b_0 in the image. Let c_0 be the west neighbor of b_0. Going clockwise from c_0 identify the first non-zero neighbor b_1 of b_0. Let c_1 be the background point immediately preceding b_1 in the sequence.
2. Let $b=b_1$ and $c=c_1$.
3. Let the 8-neighbors of b starting at c and proceeding clockwise be denoted as n_1, n_2, …, n_8. Find the first n_k which is foreground (i.e., a "1").
4. Let $b=n_k$ and $c=n_{k-1}$.
5. Repeat steps 3 and 4 until $b=b_0$ and the next boundary point found is b_1.

The sequence of b points found when the algorithm stops is the set of ordered boundary points.



Boundary Tracking



a b c

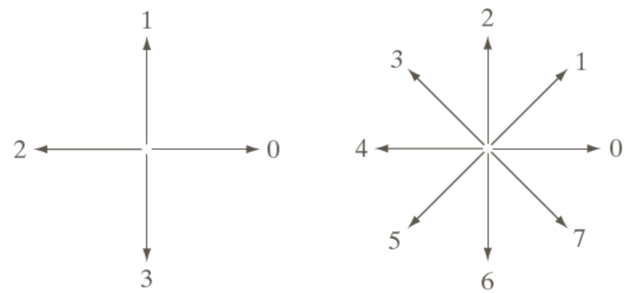
FIGURE 11.2 Illustration of an erroneous result when the stopping rule is such that boundary-following stops when the starting point, b_0, is encountered again.

The Moore boundary following algorithm repeats until $b=b_0$ <u>and</u> the next boundary point found is b_1.

This prevents errors encountered in spurs. Simply finding b_0 again is not enough. For example, in this figure the algorithm would start at b_0, find b, and then come back to b_0 stopping prematurely with only $b=b_0$. However, the next boundary point found should be the circled point which is NOT b_1. Adding "<u>and</u> the next boundary point found is b_1" will cause the algorithm to traverse the lower part of the object.

Chain Codes

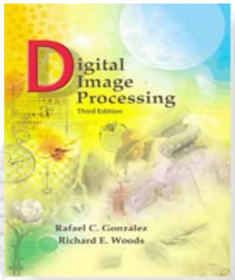
Examples of chain code direction numbers.



a b

FIGURE 11.3 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.

A Freeman chain code represents a boundary as a connected sequence of straight line segments of specified direction and length.

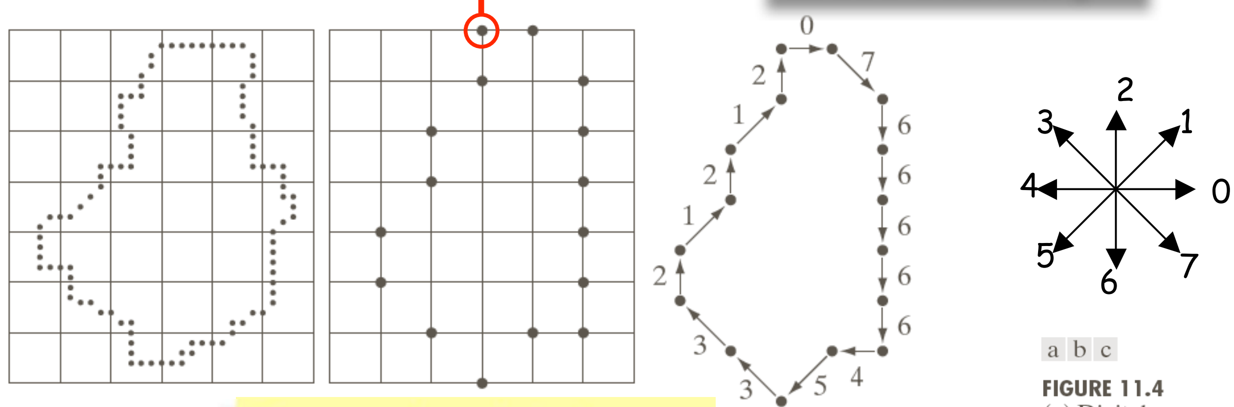


Chain Codes

Arbitrarily start chain code here

8-direction chain coded boundary

Using original pixels usually results in a code which is too long and subject to noise.

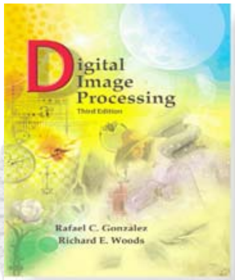


a b c

FIGURE 11.4 (a) Digital boundary with resampling grid superimposed. (b) Result of resampling. (c) 8-directional chain-coded boundary.

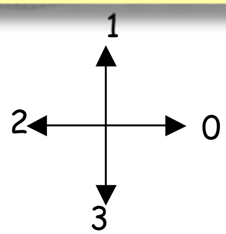
Resample original image on a larger grid to reduce code size.

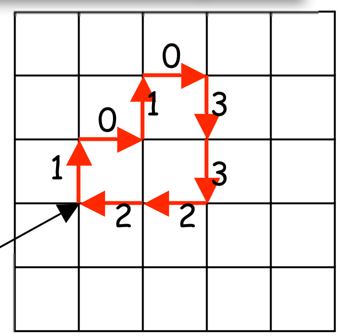
8-connected code:
0766666453321212



Chain Codes

4-direction chain coded boundary





Arbitrarily start chain code here

4-connected code: 10103322

One practice is to rotate (pick start) to get smallest integer code.

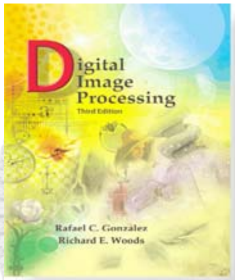
(START CODE HERE)
Between end and start of boundary

Chain codes can be normalized for rotation by using first differences.

To compute the first difference go in counter-clockwise direction and find number of 90° rotations of direction.

1->0 3-90° rotations ccw
0->1 1-90° rotation ccw
1->0 3-90° rotations ccw
0->3 3-90° rotations ccw
3->3 0-90° rotations ccw
3->2 3-90° rotations ccw
2->2 0-90° rotations ccw
2->1 3-90° rotations ccw

4-connected difference code: 33133030



Chain Codes

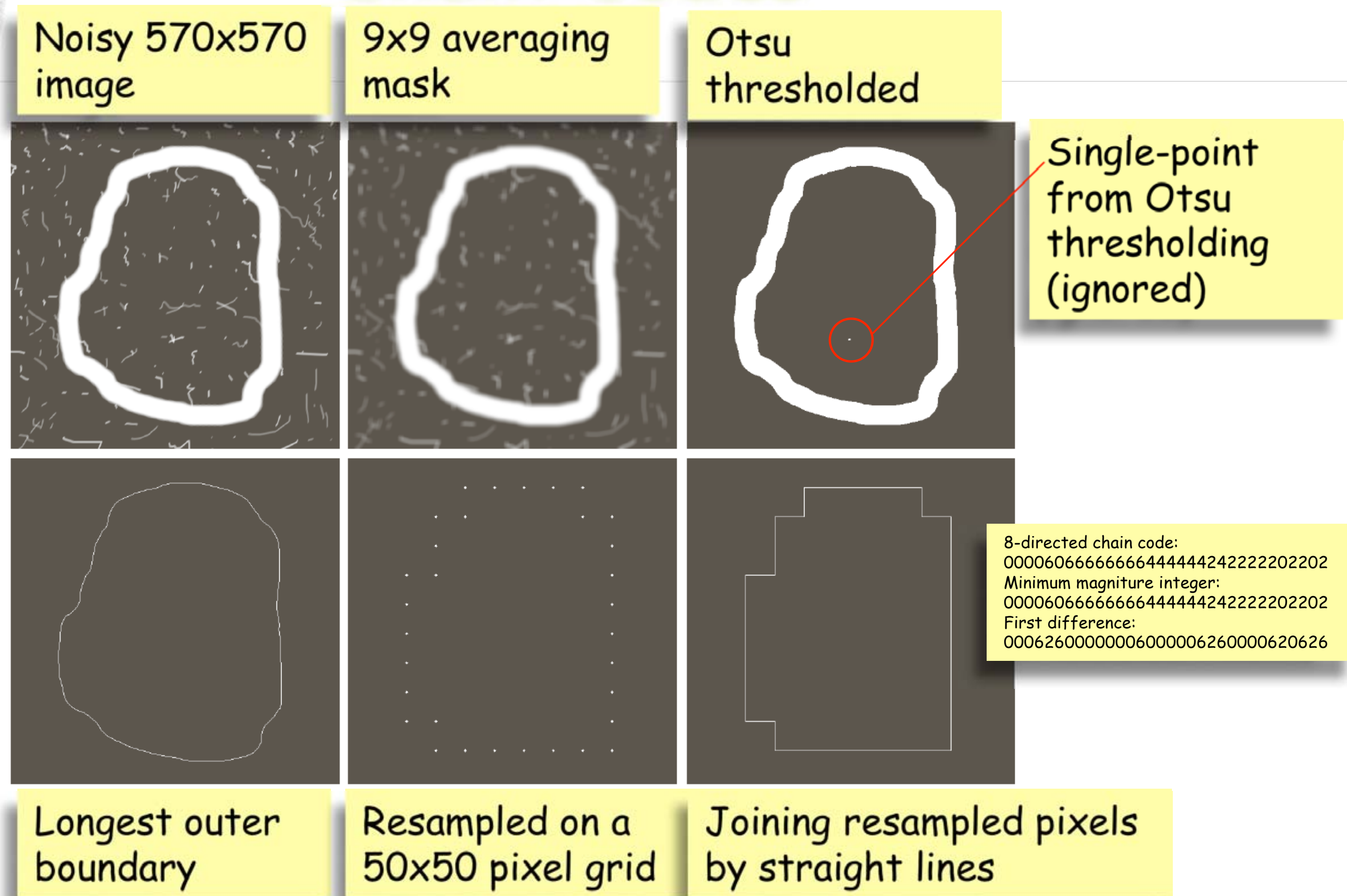
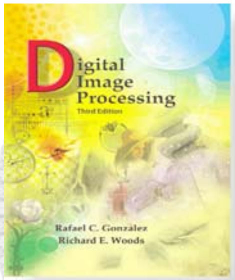
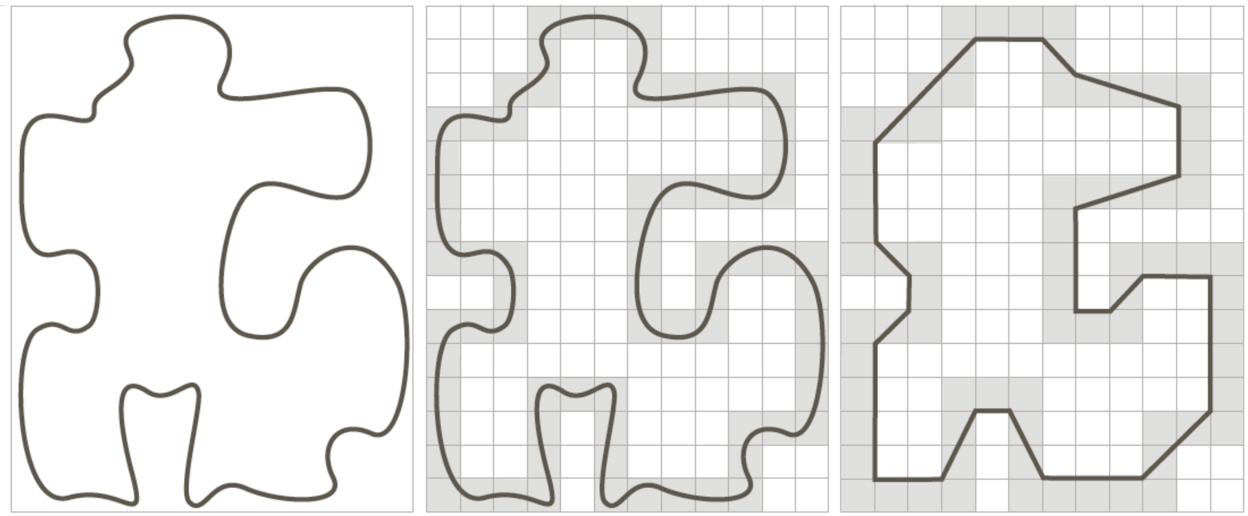


FIGURE 11.5 (a) Noisy image. (b) Image smoothed with a 9 × 9 averaging mask. (c) Smoothed image, thresholded using Otsu's method. (d) Longest outer boundary of (c). (e) Subsampled boundary (the points are shown enlarged for clarity). (f) Connected points from (e).



Minimum-Perimeter Polygon

The goal is to represent the shape in a given boundary using the fewest possible number of sequences.



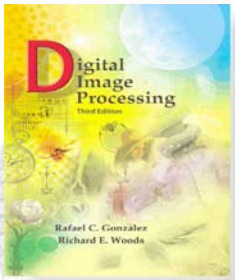
Object boundary (binary image)

Enclose boundary in a grid

Allow boundary to shrink. The vertices of the polygon are all inner or outer corners of the grid.

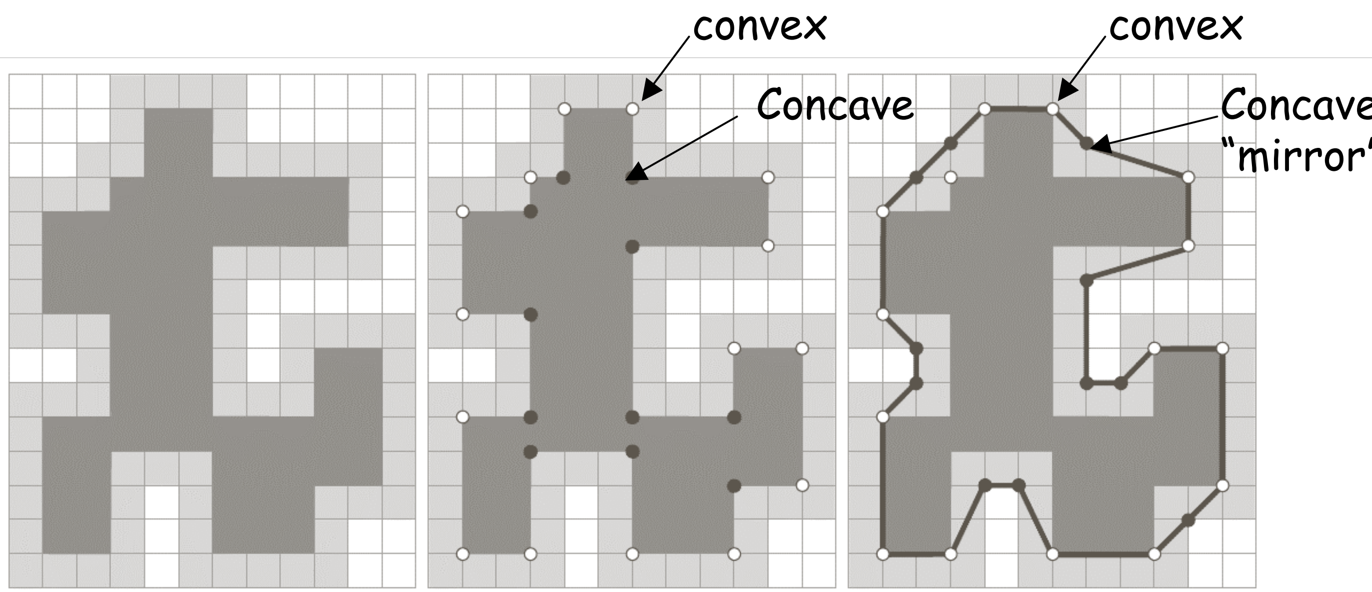
a b c

FIGURE 11.6 (a) An object boundary (black curve). (b) Boundary enclosed by cells (in gray). (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.



Minimum-Perimeter Polygon

The boundary cells from the previous slide enclose the circumscribed shape.

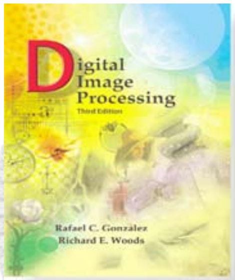


Traverse the 4-connected boundary of the circumscribed shape.

Concave vertices on this boundary have "mirrors" on the outer boundary. The boundary is described by inner convex and outer concave vertices.

a b c

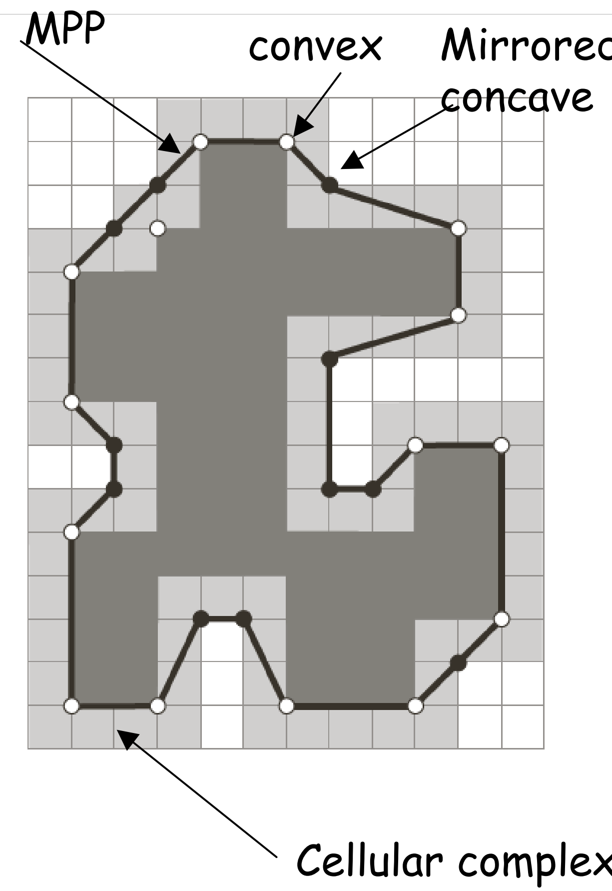
FIGURE 11.7 (a) Region (dark gray) resulting from enclosing the original boundary by cells (see Fig. 11.6). (b) Convex (white dots) and concave (black dots) vertices obtained by following the boundary of the dark gray region in the counterclockwise direction. (c) Concave vertices (black dots) displaced to their diagonal mirror locations in the outer wall of the bounding region; the convex vertices are not changed. The MPP (black boundary) is superimposed for reference.



Minimum-Perimeter Polygon

<u>MPP Observations:</u>

1. The MPP bounded by a simply connected cellular complex is not self-intersecting.
2. Every *convex* vertex of the MPP is a W vertex, but not every W vertex of a boundary is a vertex of the MPP.
3. Every *mirrored concave* vertex of the MPP is a B vertex, but not every B vertex of a boundary is a vertex of the MPP.
4. All B vertices are on or outside the MPP, and all W vertices are on or inside the MPP.
5. The uppermost, leftmost vertex in a sequence of vertices contained in a cellular complex is always a W vertex of the MPP.



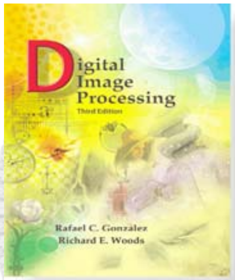
Not all vertices in the MPP become vertices of the MPP

Minimum-Perimeter Polygon

Let a=(x_1,y_1), b=(x_2,y_2), and c=(x_3,y_3)

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$$

$$\operatorname{sgn}(a,b,c) = \det(\boldsymbol{A}) = \begin{cases} >0 & \textit{if } (a,b,c) \textit{ is a counterclockwise sequence} \\ =0 & \textit{if } (a,b,c) \textit{ are collinear} \\ <0 & \textit{if } (a,b,c) \textit{ is a clockwise sequence} \end{cases}$$



Minimum-Perimeter Polygon

Definitions:

Form a list whose rows are the coordinates of each vertex and whether that vertex is W or B. The concave verttices must be mirrored, the vertices must be in sequential order, and the first uppermost, leftmost vertex V_O is a W vertex. There is a white crawler (W_C) and a black crawler (B_C). The Wc crawls along the convex W vertices, and the B_C crawls along the mirrored concave B vertices.

Minimum-Perimeter Polygon

<u>MPP Algorithm:</u>

1. Set $W_C=B_C=V_O$

2.

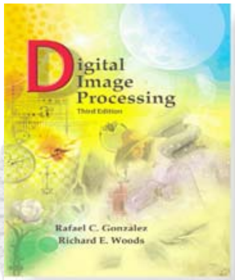
(a) V_K is on the positive side of the line (V_L,W_C) $[sgn(V_L,W_C,V_K)>0$

(b) V_K is on the negative side of the line (V_L,W_C) or is collinear with it $[sgn(V_L,W_C,V_K)\leq 0$; V_K is on the positive side of the line (V_L,B_C) or is collinear with it $[sgn(V_L,B_C,V_K)\geq 0$

(c) V_K is on the negative side of the line (V_L,B_C) $[sgn(V_L,B_C,V_K)<0$

If condition (a) holds the next MPP vertex is W_C and $V_L=W_C$; set $W_C=B_C=V_L$ and continue with the next vertex.
If condition (b) holds V_K becomes a candidate MPP vertex. Set $W_C=V_K$ if V_K is convex otherwise set $B_C=V_K$. Continue with next vertex.
If condition (c) holds the next vertex is B_C and $V_L=B_C$.
Re-initialize the algorithm by setting $W_C=B_C=V_L$ and continue with the next vertex after V_L.

3. Continue until the first vertex is reached again.



Minimum-Perimeter Polygon

V_0	(1,4)	W
V_1	(2,3)	B
V_2	(3,3)	W
V_3	(3,2)	B
V_4	(4,1)	W
V_5	(7,1)	W
V_6	(8,2)	B
V_7	(9,2)	B

The fundamental concept is to move the crawlers along the perimeter, calculate the curvatures, and determine if the vertex is a vertex of the MPP.

	W_C	B_C	V_L	W_C curvature	B_C curvature
	V_0	V_0	V_0		
V_1	V_0	V_0	V_0	$V_L,W_C,V_1=0$	$V_L,W_B,V_1=0$
V_2	V_0	V_1	V_0	$V_L,W_C,V_2=0$	$V_L,W_B,V_2>1$
V_3	V_2	V_1	V_0	$V_L,W_C,V_3<0$	$V_L,W_B,V_3=0$
V_4	V_2	V_3	V_0	$V_L,W_C,V_4<0$	$V_L,W_B,V_4=0$
V_5	V_4	V_3	V_4	$V_L,W_C,V_5>0$	
V_5	V_4	V_4	V_4	$V_L,W_C,V_5=0$	$V_L,W_B,V_5=0$
V_6	V_5	V_4	V_4	$V_L,W_C,V_6>0$	
V_7	V_5	V_6	V_5	$V_L,W_C,V_7=0$	$V_L,W_B,V_7=0$

Minimum Perimeter Polygon

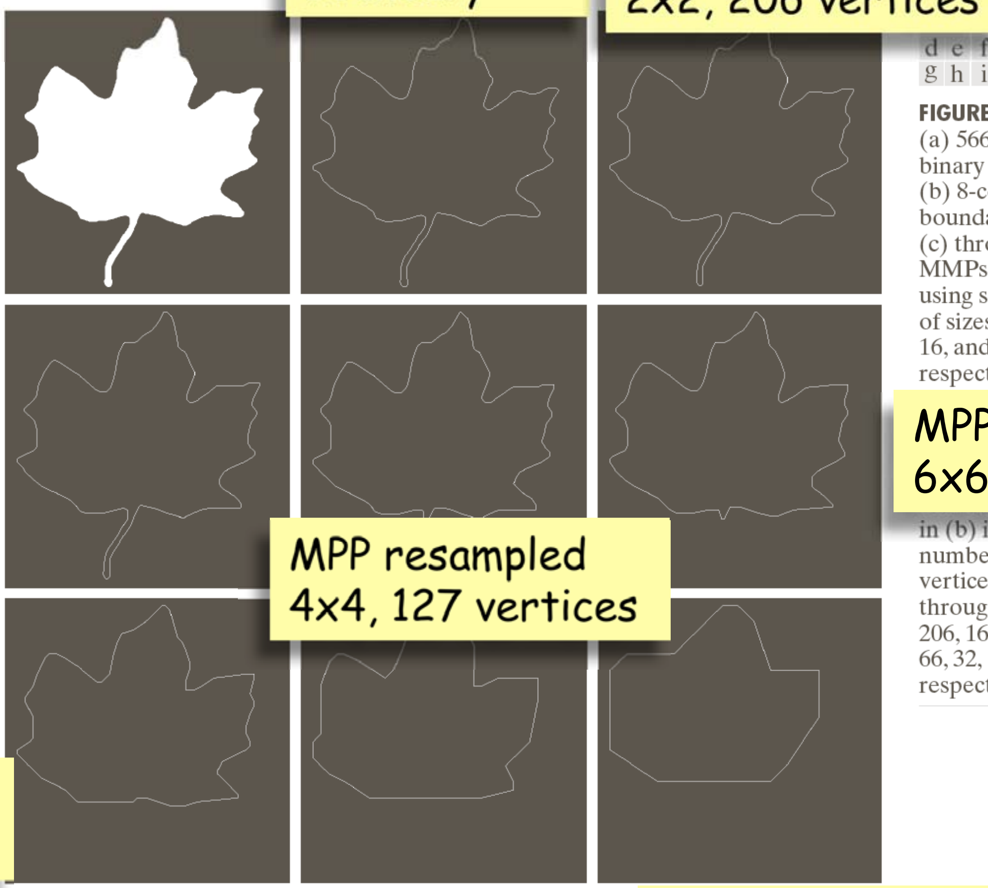
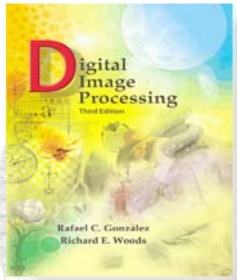


FIGURE 11.8 (a) 566 × 566 binary image. (b) 8-connected boundary. (c) through (i), MMPs obtained using square cells of sizes 2, 3, 4, 6, 8, 16, and 32, respectively (the

in (b) is 1900. The numbers of vertices in (c) through (i) are 206, 160, 127, 92, 66, 32, and 13, respectively.



Boundary Splitting

Subdivide a segment successively until a specified criterion is satisfied

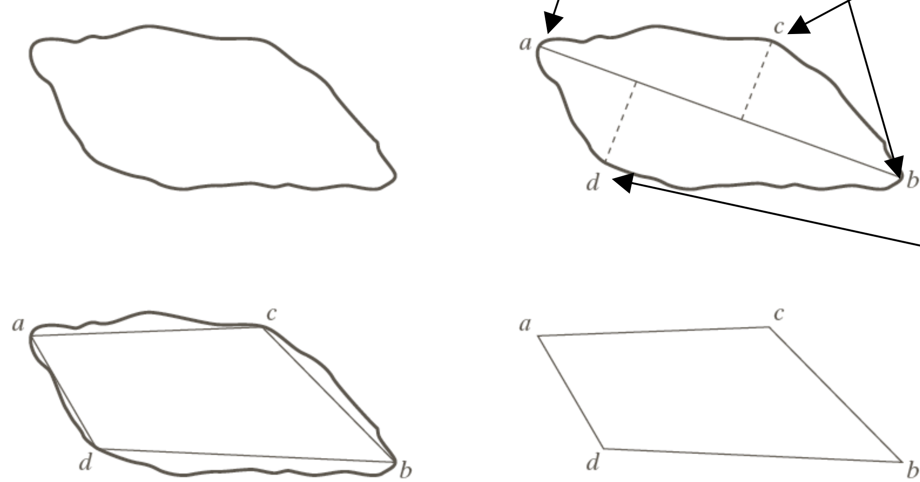
1. Identify the two most distant points joined by ab

2. Identify the most ⊥ distant point up from ab

3. Identify the most ⊥ distant point down from ab

a b
c d

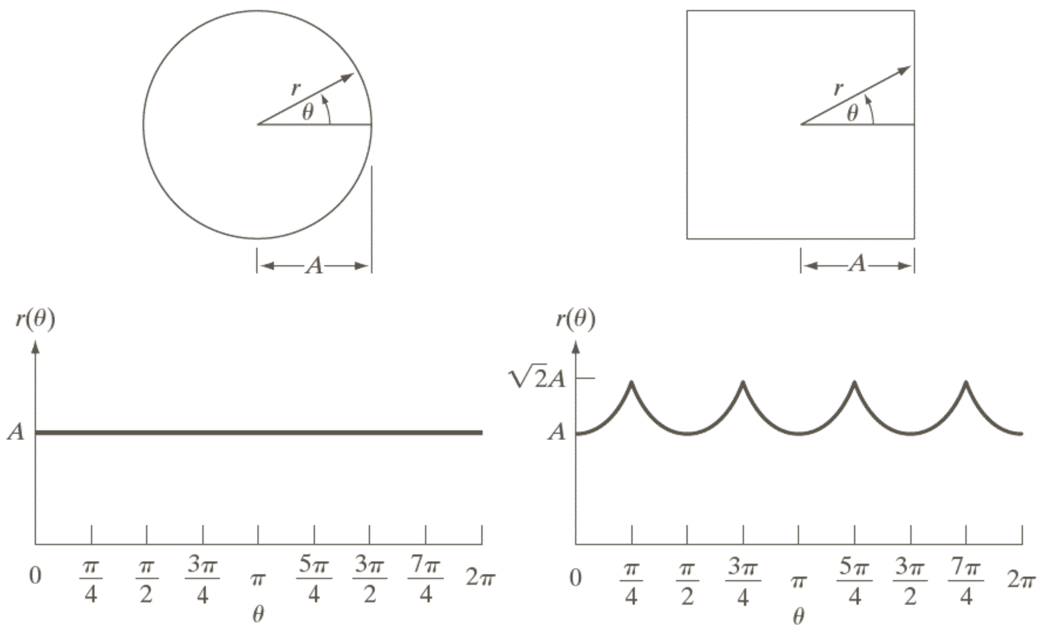
FIGURE 11.9
(a) Original boundary. (b) Boundary divided into segments based on extreme points. (c) Joining of vertices. (d) Resulting polygon.



Signatures

A signature is a 1-D representation of a boundary

Signatures can be made invariant to translation but may be sensitive to rotation and scale.



a b

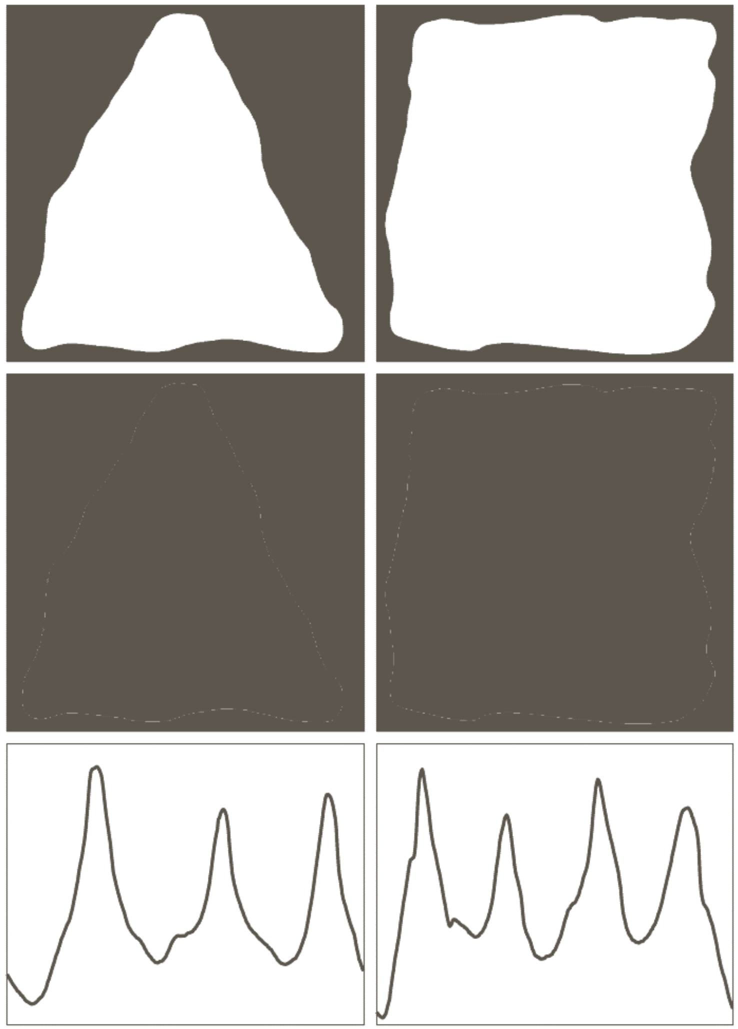
FIGURE 11.10 Distance-versus-angle signatures. In (a) $r(\theta)$ is constant. In (b), the signature consists of repetitions of the pattern $r(\theta) = A \sec\theta$ for $0 \le \theta \le \pi/4$ and $r(\theta) = A \csc\theta$ for $\pi/4 < \theta \le \pi/2$.

These signatures are r-θ but there are others such as ψ(tangent)-s.

Methods of selecting starting point can make signature independent of rotation

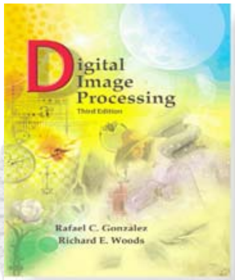
1. Select point farthest from centroid
2. Select point farthest from centroid along eigenaxis
3. Use a chain code

Signatures



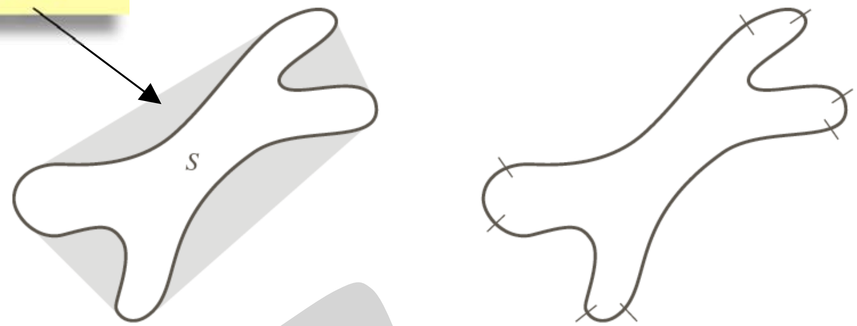
a b
c d
e f

FIGURE 11.11 Two binary regions, their external boundaries, and their corresponding $r(\theta)$ signatures. The horizontal axes in (e) and (f) correspond to angles from 0° to 360°, in increments of 1°.



Representation and Description

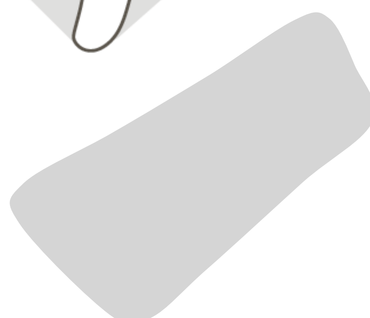
The convex deficiency H-S



a b

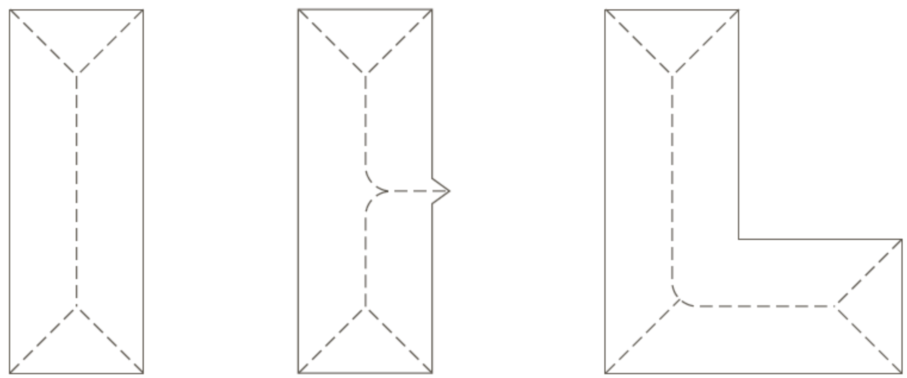
FIGURE 11.12 (a) A region, S, and its convex deficiency (shaded). (b) Partitioned boundary.

The convex hull H



The boundary of the object can be coded by the points where the boundary passes in and out of a convex deficiency

Median Axis Transformation

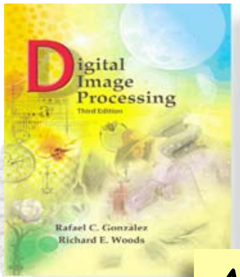


a b c

FIGURE 11.13 Medial axes (dashed) of three simple regions.

Reduce the structure of a shape to a skeleton. However, as seen in our previous examples a morphological skeleton will not necessary be connected.

Median Axis Transformation (MAT) of a region R with border B (guarantees connectivity of the skeleton)
1. For each point p in R find its closest* neighbor on B
2. If p has more than one "closest"* neighbor it belongs to the medial axis (skeleton) of B
* Closest is defined using Euclidian distance



Thinning

A suitable thinning algorithm is computationally MUCH faster that the MAT.

Definitions
A border point is any pixel with value 1 and at least one 8-connected neighbor with value 0.
$N(p_1)$ is the number of non-zero neighbors of p_1, i.e., $N(p_1)=p_2+p_3+\ldots+p_8+p_9$
$T(p_1)$ is the number of 0->1 transitions in the ordered sequence $p_1p_2p_3\ldots p_9p_2$

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

FIGURE 11.14 Neighborhood arrangement used by the thinning algorithm.

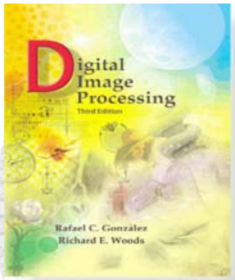
FIGURE 11.15 Illustration of conditions (a) and (b) in Eq. (11.1-4). In this case $N(p_1) = 4$ and $T(p_1) = 3$.

0	0	1
1	p_1	0
1	0	1

p_1 is a boundary point

$N(p_1) = 4$

$T(p_1) = 3$



Representation and Description

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

FIGURE 11.14 Neighborhood arrangement used by the thinning algorithm.

Algorithm for thinning binary images

Step 1. Mark for deletion any <u>border</u> point which has all of the following:

- (a) $2 \le N(p_1) \le 6$ % don't delete if p_1 is an end point or inside region
- (b) $T(p_1)=1$ % prevents breaking lines
- (c) $p_2p_4p_6=0$ % (c) and (d) say that either p_2 AND p_8 are 0
- (d) $p_4p_6p_8=0$ % or p_4 or p_6 are 0, i.e., not part of the skeleton

Step 2. Mark for deletion any <u>border</u> point which has all of the following:

- (a) $2 \le N(p_1) \le 6$ % don't delete if p_1 is an end point or inside region
- (b) $T(p_1)=1$ % prevents breaking lines
- (c) $p_2p_4p_8=0$ % (c) and (d) say that either p_4 AND p_6 are 0
- (d) $p_4p_6p_8=0$ % or p_2 OR p_8 are 0, i.e., not part of the skeleton

Iterate by applying Step 1 to all border points and deleting marked points. Then apply Step 2 to all remaining border points and delete marked points. Continue until no further points are deleted.

Representation and Description

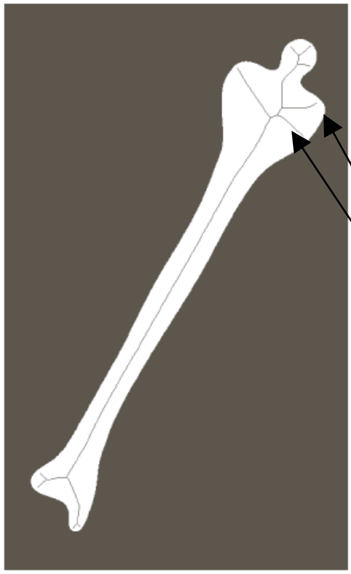
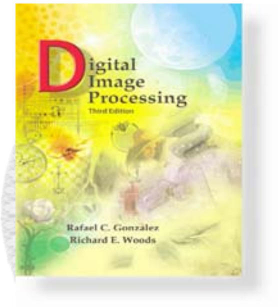


FIGURE 11.16 Human leg bone and skeleton of the region shown superimposed.

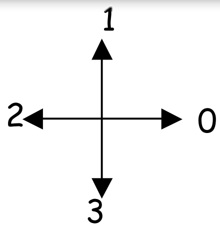
Double branch since wider than left side

Skeletonized ("thinned") image.

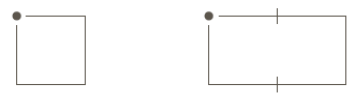


Representation and Description

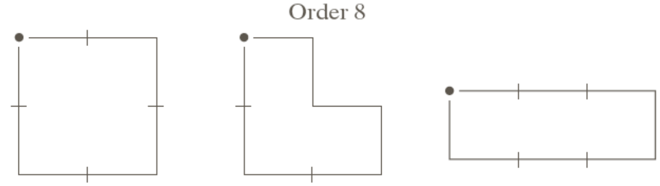
4-direction chain code



To compute the first difference go in counter-clockwise direction and find number of 90° rotations of direction.



	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3



	Order 8		
Chain code:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape no.:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

FIGURE 11.17 All shapes of order 4, 6, and 8. The directions are from Fig. 11.3(a), and the dot indicates the starting point.

The shape number n is the smallest magnitude first difference chain code.
n is even for closed boundaries

Representation and Description

Definitions:

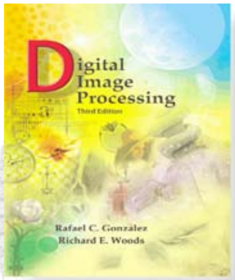
Diameter = $\max[D(p_i), D(p_j)]$ where p_i and p_j are points on the boundary.

Major axis is the line segment of length equal to the diameter and connecting two points on the boundary.

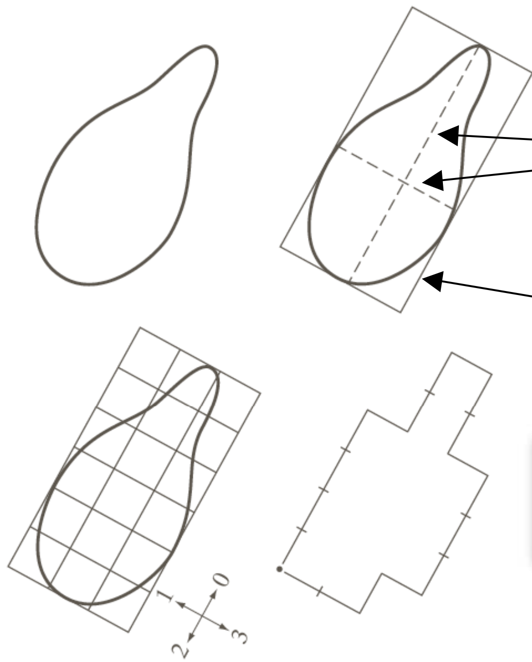
Minor axis is the line perpendicular to the major axis and of such length that a box passing through the outer four points of intersection of the boundary and the major/minor axes completely enclose the boundary

This box enclosing the boundary is called the basic rectangle.

Eccentricity is the ratio of major to minor axis.



Representation and Description



1. The major and minor axes

2. The basic rectangle

3. Use the rectangle with n=18 (given) which best approximates the shape of basic rectangle, i.e., 6x3=18.

4. Resample boundary or use polygonal approximation

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

5. Computer chain code, its first difference, and the corresponding shape number.

a b
c d

FIGURE 11.18 Steps in the generation of a shape number.

Fourier Boundary Descriptors

1. Represent each point on a digital boundary as $s(k)=x(k)+jy(x)$

2. Compute the DFT of the set of boundary points

$$a(u)=\frac{1}{K}\sum_{k=0}^{K-1}s(k)e^{-j\frac{2\pi uk}{K}} \quad ,u=0,1,2,\ldots,K-1$$

3. The coefficients a(u) are the Fourier descriptors of the boundary

$$s(k)=\sum_{u=0}^{K-1}a(u)e^{+j\frac{2\pi uk}{K}}$$

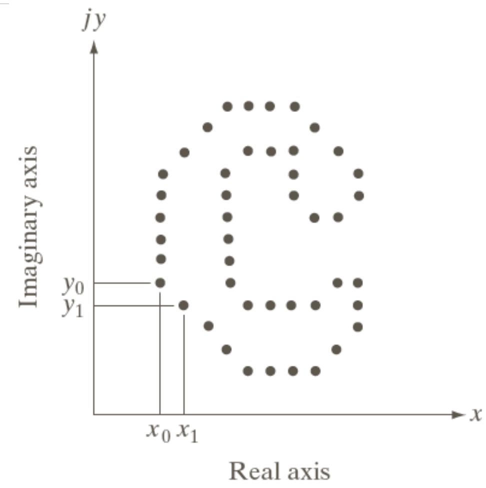
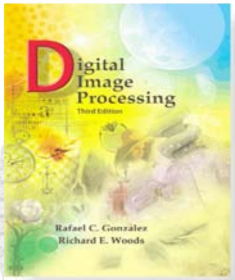


FIGURE 11.19 A digital boundary and its representation as a complex sequence. The points (x_0, y_0) and (x_1, y_1) shown are (arbitrarily) the first two points in the sequence.

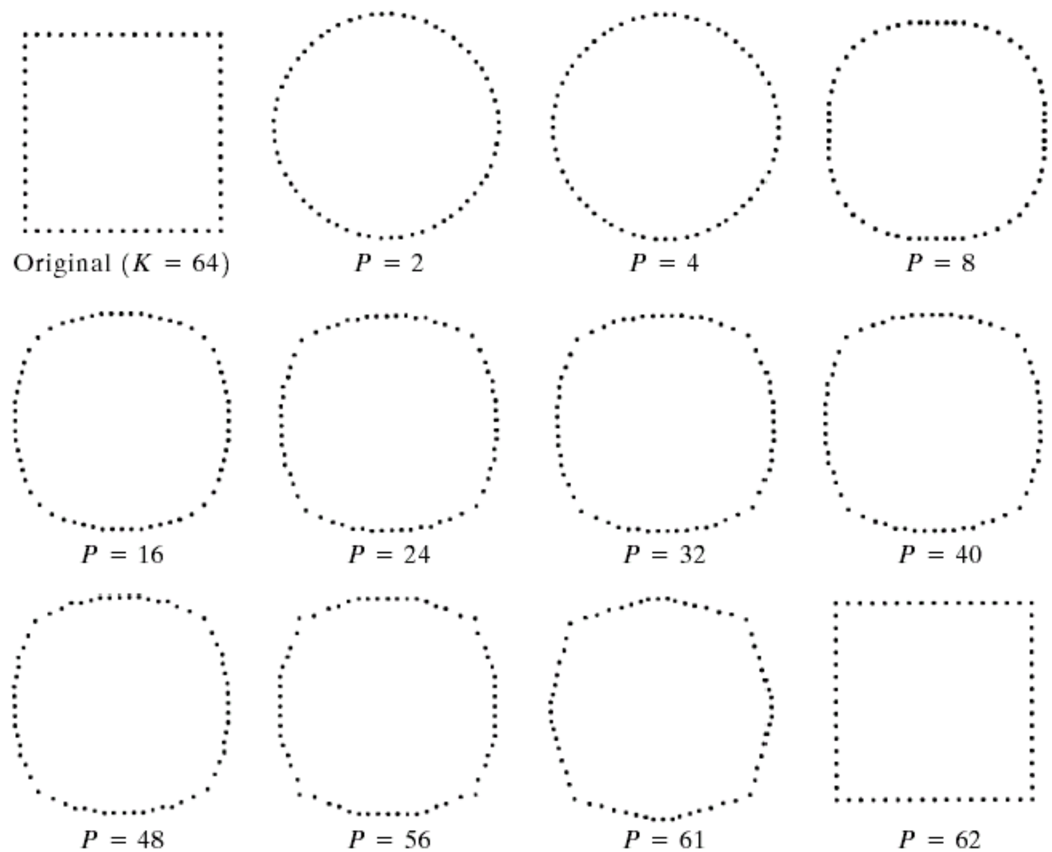
Since K can be large we usually approximate the boundary by a smaller set of points, i.e., P, so that

$$s(k)\cong\hat{s}(k)=\sum_{u=0}^{P-1}a(u)e^{+j\frac{2\pi uk}{K}}$$



Fourier Boundary Descriptors

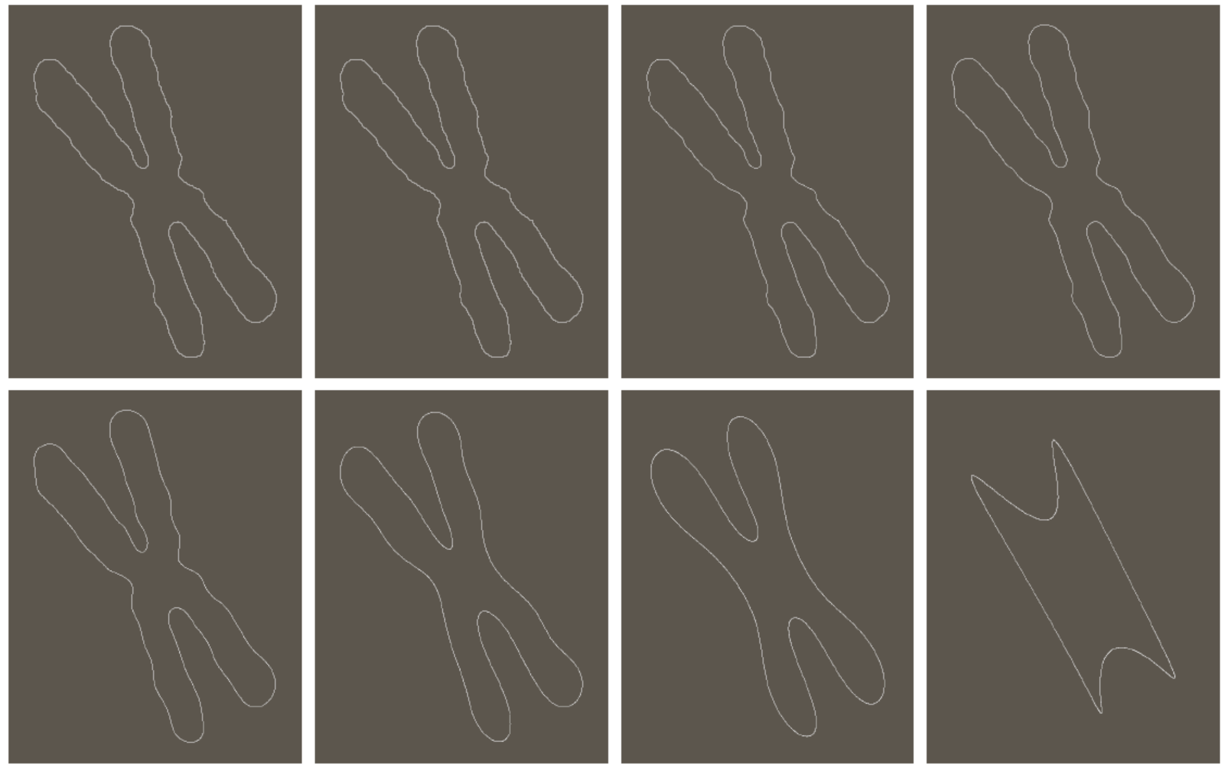
FIGURE 11.14 Examples of reconstruction from Fourier descriptors. *P* is the number of Fourier coefficients used in the reconstruction of the boundary.



There are the same number of points in each reconstructed boundary but only the first P terms of the Fourier boundary descriptor were used to reconstruct the boundary. Basically, this is low-pass filtering of the shape.

Fourier Boundary Descriptors

Fourier shape reconstruction using the first P=1434, 286, 144, 72, 36, 18 and 8 terms respectively.



a b c d
e f g h

FIGURE 11.20 (a) Boundary of 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.

Fourier Boundary 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	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

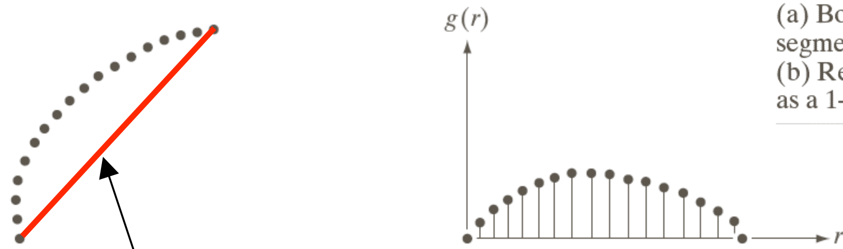
TABLE 11.1 Some basic properties of Fourier descriptors.

Shifts the DC (k=0) term

Multiplies each term in a known way

Rotation, scale, and translation of a boundary have simple effects on the Fourier description of that boundary.

Moments



a b

FIGURE 11.21
(a) Boundary segment.
(b) Representation as a 1-D function.

Connect start and stop points and compute perpendicular displacement from this line

Can compute the mean displacement m and higher order moments μ_n

$$m = \sum_{i=0}^{K-1} r_i g(r_i)$$

$$\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i)$$

Other Representations

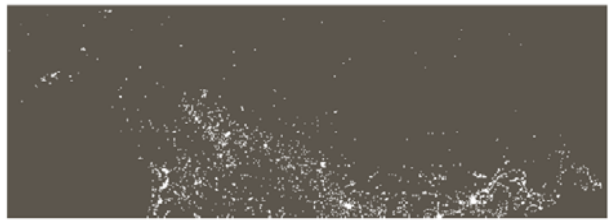


FIGURE 11.22 Infrared images of the Americas at night. (Courtesy of NOAA.)

Region no. (from top)	Ratio of lights per region to total lights
1	0.204
2	0.640
3	0.049
4	0.107

Use ratio of white pixels to total area to estimate electrical energy consumption.

Topology

Topology - properties that are unaffected by "rubber" sheet deformations

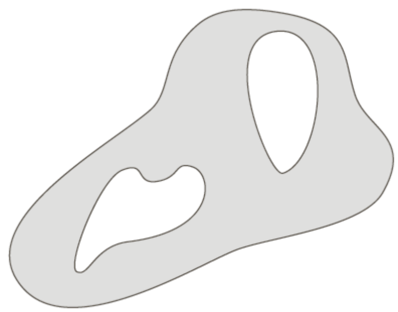


FIGURE 11.23 A region with two holes.

The set of pixels that are connected to any pixel in S is called a connected component of S

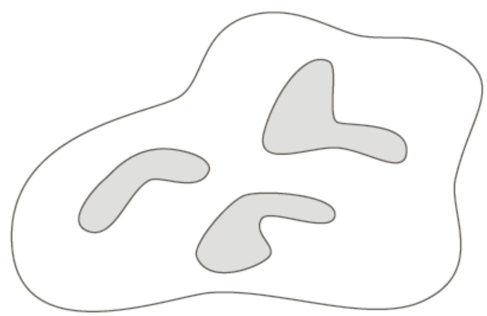
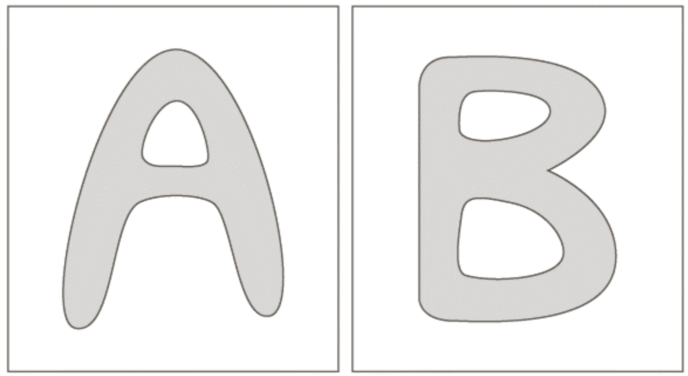


FIGURE 11.24 A region with three connected components.

Representation and Description



a b

FIGURE 11.25 Regions with Euler numbers equal to 0 and −1, respectively.

E=C-H=1-1=0

E=C-H=1-2=-1

Euler number E=C-H

C is the number of connected components; H is the number of holes

Representation and Description

This polygonal network has:
7 vertices
11 edges
2 faces
3 holes
1 connected region

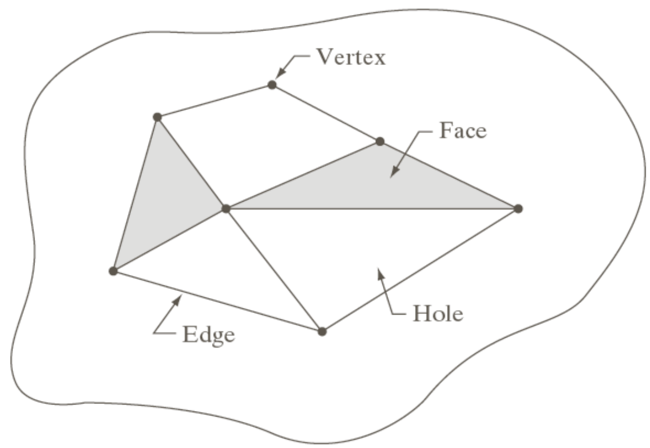


FIGURE 11.26 A region containing a polygonal network.

E=V-Q+F=7-11+2=-2

Euler number E=V-Q+F

V is the number of vertices
Q is the number of edges
F is the number of faces

Representation and Description

Single infrared image

Single component with largest number of pixels (8479)

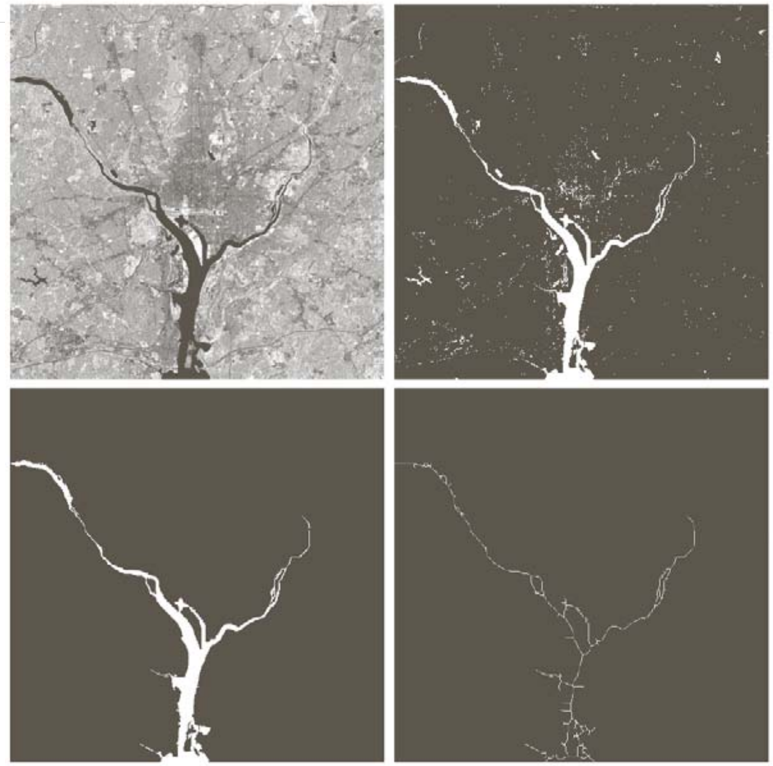


FIGURE 11.27 (a) Infrared image of the Washington, D.C. area. (b) Thresholded image. (c) The largest connected component of (b). Skeleton of (c).

Result of thresholding (largest T before river becomes disconnected) gives 1591connected components

Computed skeleton of this component useful for computing length of river branches, etc.