WRIGHT STATE UNIVERSITY

Chapter-12
Feature Extraction

Beyond Segmentation

- Segmentation results in one or more objects in an image.
- Potential next steps depend on the application:
  - **Biomedical Image Analysis**: Detect or classify disease
    - Alzheimer's disease: Size of ventricles, Gray matter atrophy
  - **Object Recognition**:
    - Distinguish between a car, an airplane or a human
  - **Satellite Imagery**: Identify regions of interest
    - Is this a forest or a building complex hidden in the forest?

Image Representation and Description

Objective:

To represent and describe information embedded in an image in other forms that are more suitable than the image itself.

Benefits:

- Easier to understand
- Require fewer memory, faster to be processed
- More "ready to be used"

What kind of information we can use?

- Boundary, shape
- Region
- Texture
- Relation between regions

Representation

- Need a set of numbers (**Feature Vector**):
  - ➢ Represent a segmented region
  - ➢ Compare against feature vectors from other regions, e.g., prototypes of different classes
- Two main possible types of feature vectors:
  - ➢ **External Characteristics**: Boundary based
    - → Useful for representing Shape
  - ➢ **Internal Characteristics**: Pixel based
    - → Computed from the interior statistics of the region
    - → Useful for representing object color, texture, etc.

Boundary Representation

- We need an ordered set of points

Why focus on a boundary?

➢ The boundary is a good representation of an object shape and also requires limited memory

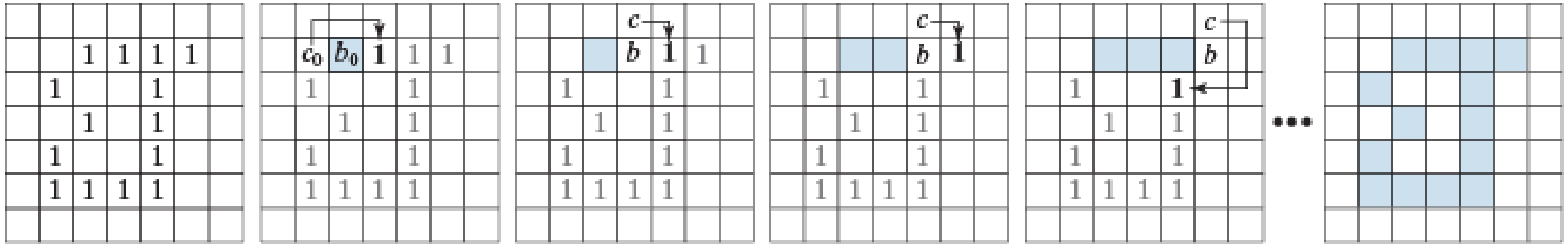
➢ If starting directly from a binary image (1's for edge pixels and 0's for background), we can use a boundary-following algorithm

Boundary (Border) Following

- Boundary regions can be clockwise or counterclockwise. Need:
- Ordered sequence of points
- Two Assumptions:
  - ➢ Binary images:
    - ▪ Objects → 1
    - ▪ Background → 0
  - ➢ Images are padded with borders of zeros

Boundary Following

1. Let b_0 be the starting point – uppermost, leftmost point labeled 1 in the image
   - Let c_0 be the west neighbor of b_0 (always background)
   - Starting from c_0 examine 8-neighbors of b_0 in clockwise direction.
     - Let b_1 be the first pixel with label 1 that is encountered.
     - Let c_1 be the background pixel 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 going clockwise be $n_1, n_2,\ldots, n_8$. Find the first n_k labeled 1
4. Let $b=b_k$ and $c=c_k$
5. Repeat steps 3 and 4 until $b=b_0$ and the next point is b_1



a b c d e f

FIGURE 12.1 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.

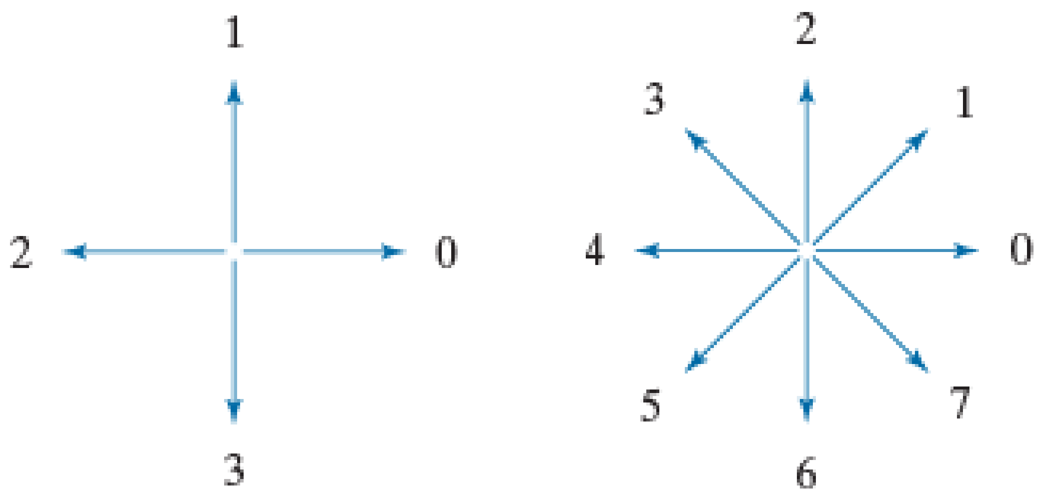
WRIGHT STATE UNIVERSITY

Chain Codes

- Represent an object boundary by a connected sequence of straight line segments of specified length and direction
- Direction of each segment is coded by a numbering scheme
- Known as Freeman Chain Code
- Invariant to Object Rotation

a b

FIGURE 12.3 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.



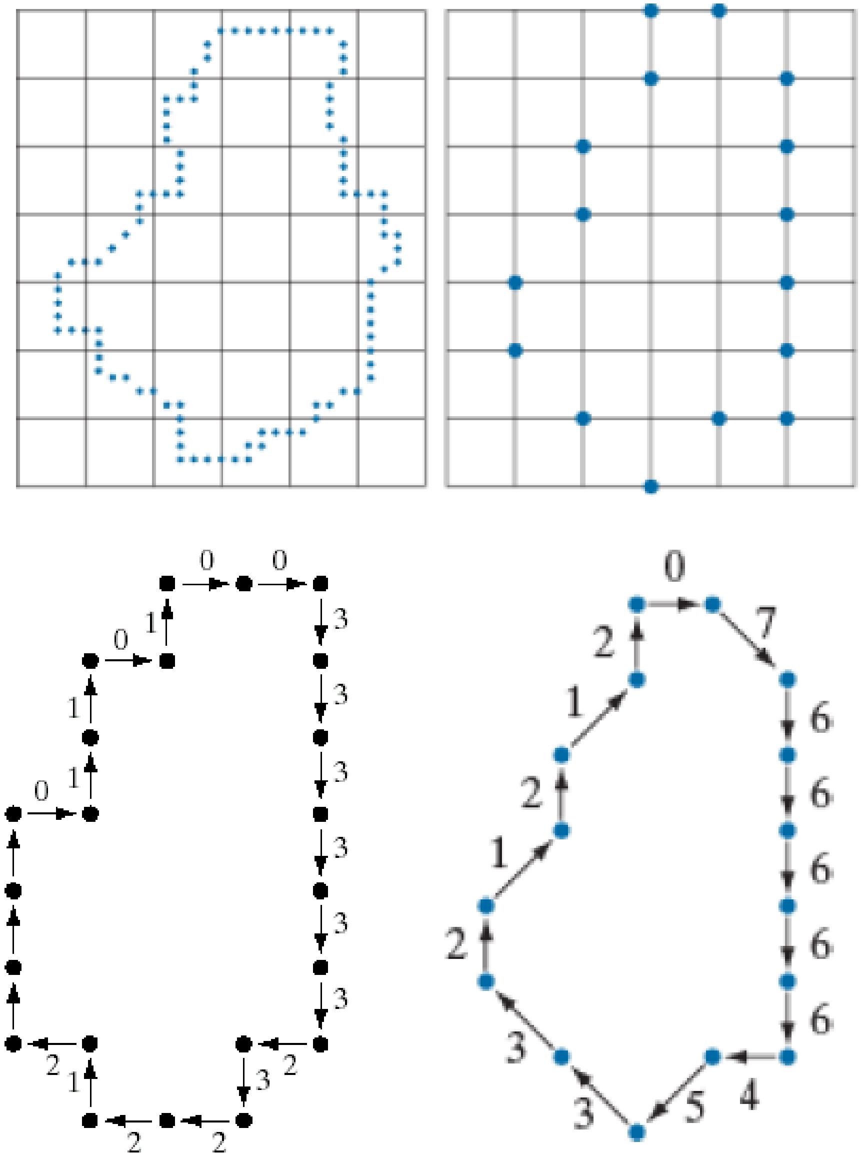
4-directional chain code

8-directional chain code

12.1 Examples of Chain Codes

Object boundary (resampling)

Boundary vertices



a b c

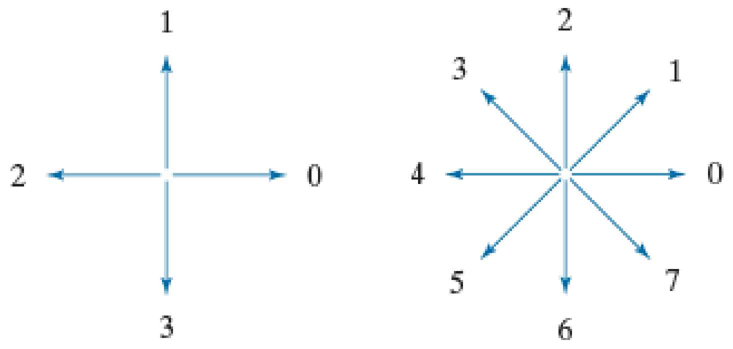
FIGURE 12.4
(a) Digital boundary with resampling grid superimposed.
(b) Result of resampling.
(c) 8-directional chain-coded boundary.

4-directional chain code

8-directional chain code

Chain Code (cont.)

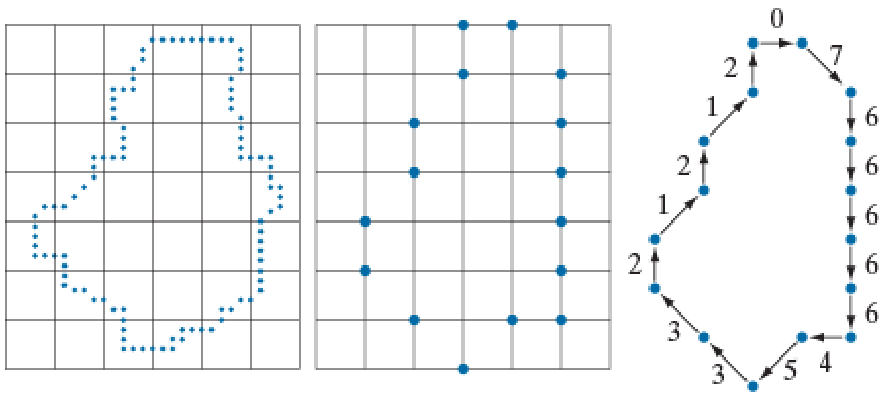
- Represent a boundary by stepping directions based either on 4- or 8-connectivity



a b

FIGURE 12.3 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.

- How does it work?
  - ➢ Start from any point → Go Clockwise → 076666645…
  - ➢ Circular sequence of numbers.
  - ➢ Needs a starting point
  - ➢ Redefine starting point as the one that forms the smallest integer



a b c

FIGURE 12.4 (a) Digital boundary with resampling grid superimposed. (b) Result of resampling. (c) 8-directional chain-coded boundary.

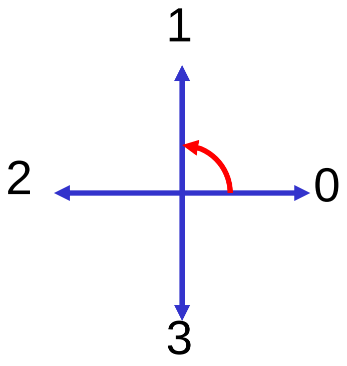
The First Difference of a Chain Code

Problem: A chain code sequence depends on a starting point

Solution: Treat a chain code as a circular sequence and redefine the starting point so that the resulting sequence of numbers forms an integer of minimum magnitude → 076666645…

The first difference of a chain code: Count the number of direction changes (counterclockwise) between two adjacent elements of the code.

Example:



Chain code :	The first difference
0 → 1	1
0 → 2	2
0 → 3	3
2 → 3	1
2 → 0	2
2 → 1	3

and so on..

Example:

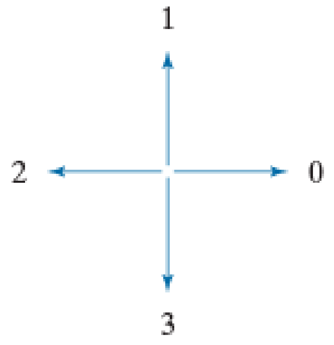
- A Chain code: 10103322
- The first difference = 3133030
- Treating a chain code as a circular sequence, we get the first difference = 33133030
- The first-3 → Transition from last element to first element

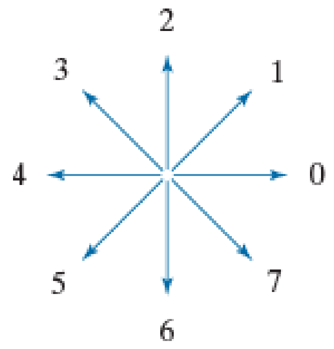
Important: The first difference is rotation invariant.

WRIGHT STATE UNIVERSITY

Chain Code (contd.)

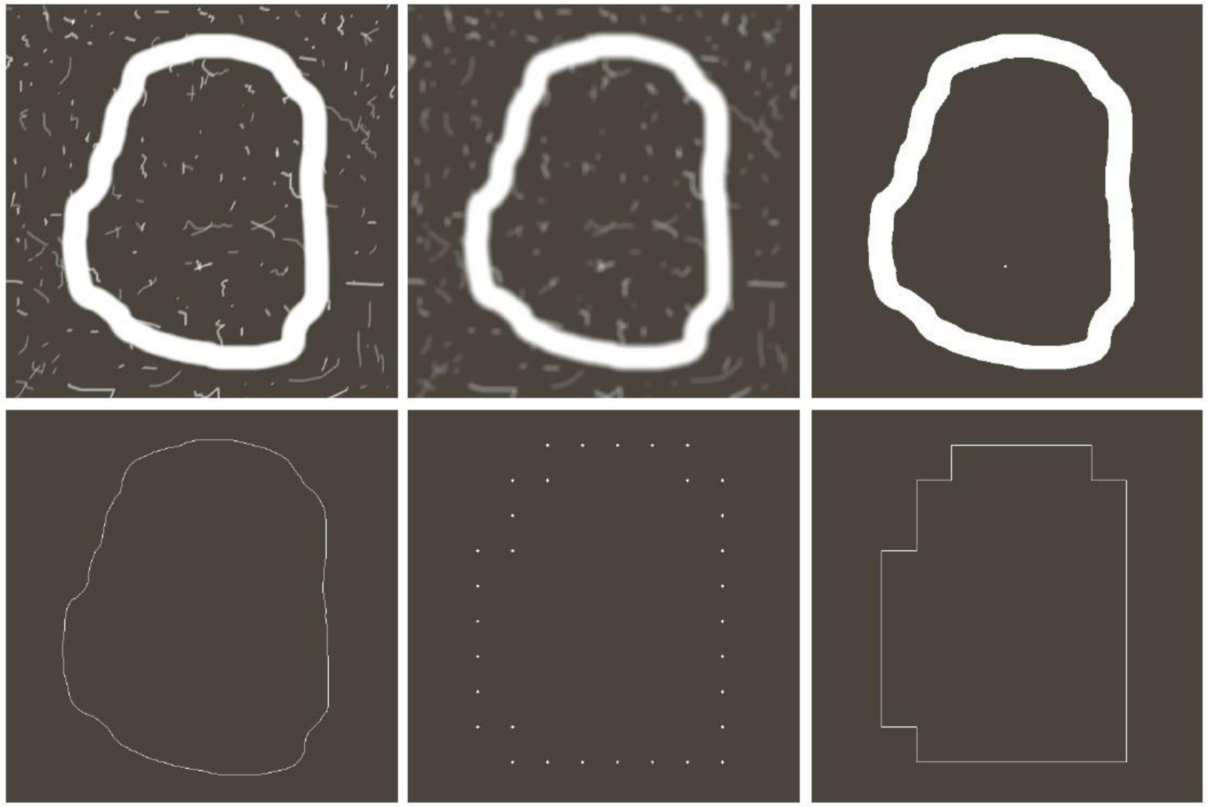
- It is easy to make the chain code invariant to rotations of the object
  - Use first difference of the chain code rather than directly using the chain code
  - Example:
    - Chain code: 10103322
    - First difference: 33133030





- Robustness to noise
  - Noise can change the chain code a lot
  - Possible solutions
    - Smooth the image
    - Apply Threshold
    - Subsample

Chain Code Example



a b c
d e f

FIGURE 12.5 (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).

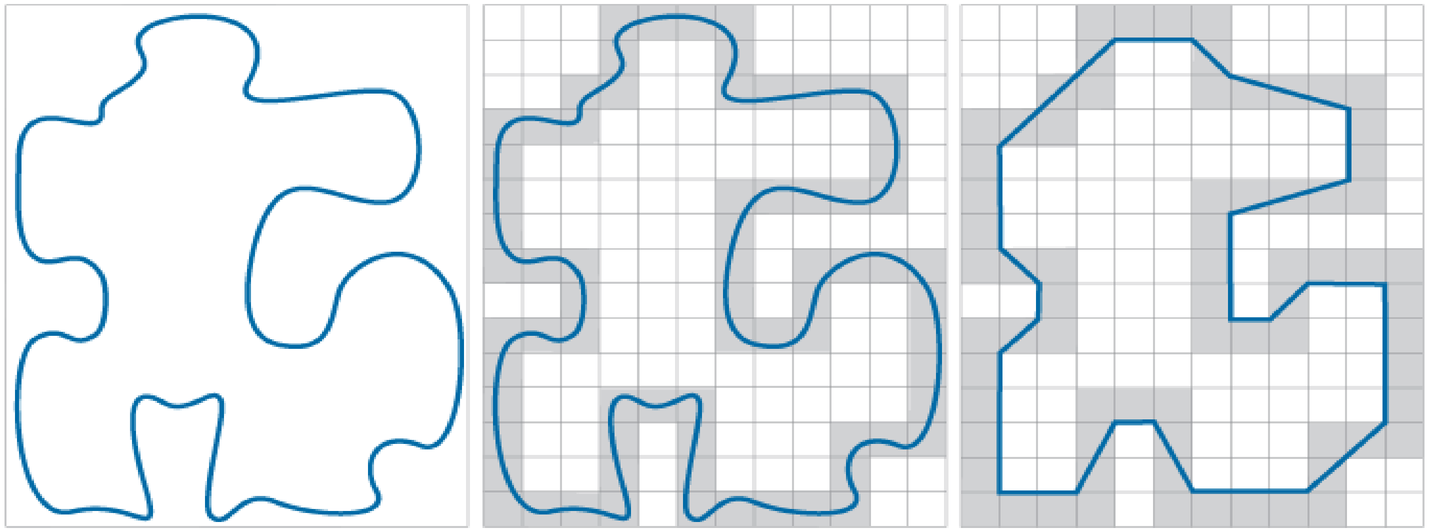
WRIGHT STATE UNIVERSITY

12.1 Polygon Approximation

- Represent an object boundary by a polygon

a b c

FIGURE 12.7 (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.

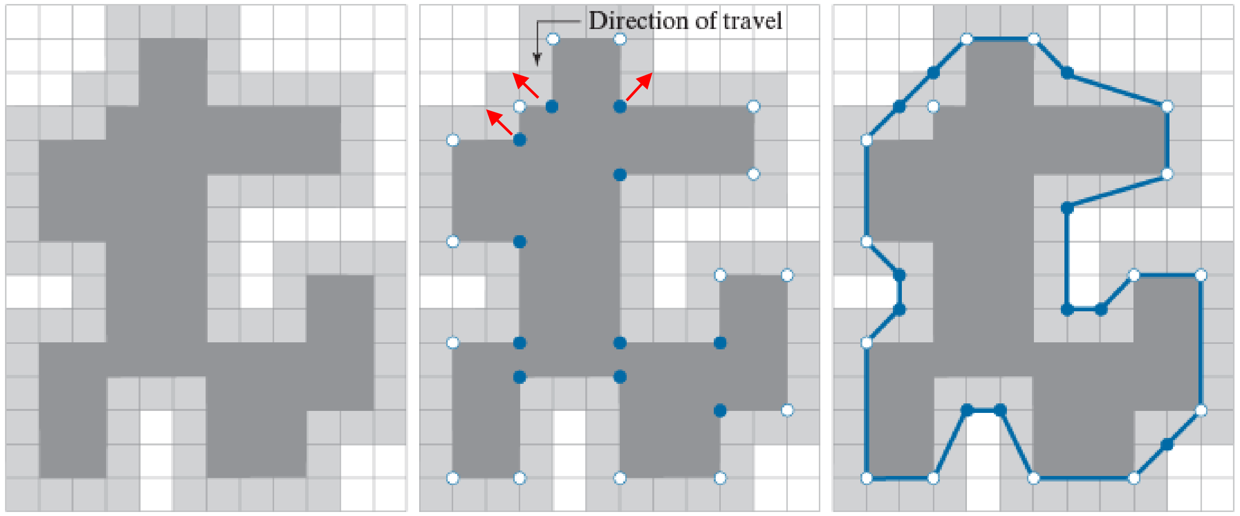


Object boundary Minimum perimeter polygon

- Minimum perimeter polygon (MPP): Consists of line segments that minimize distances between boundary pixels.

12.1 Polygon Approximation

- Represent an object boundary by a Convex polygon



Object boundary

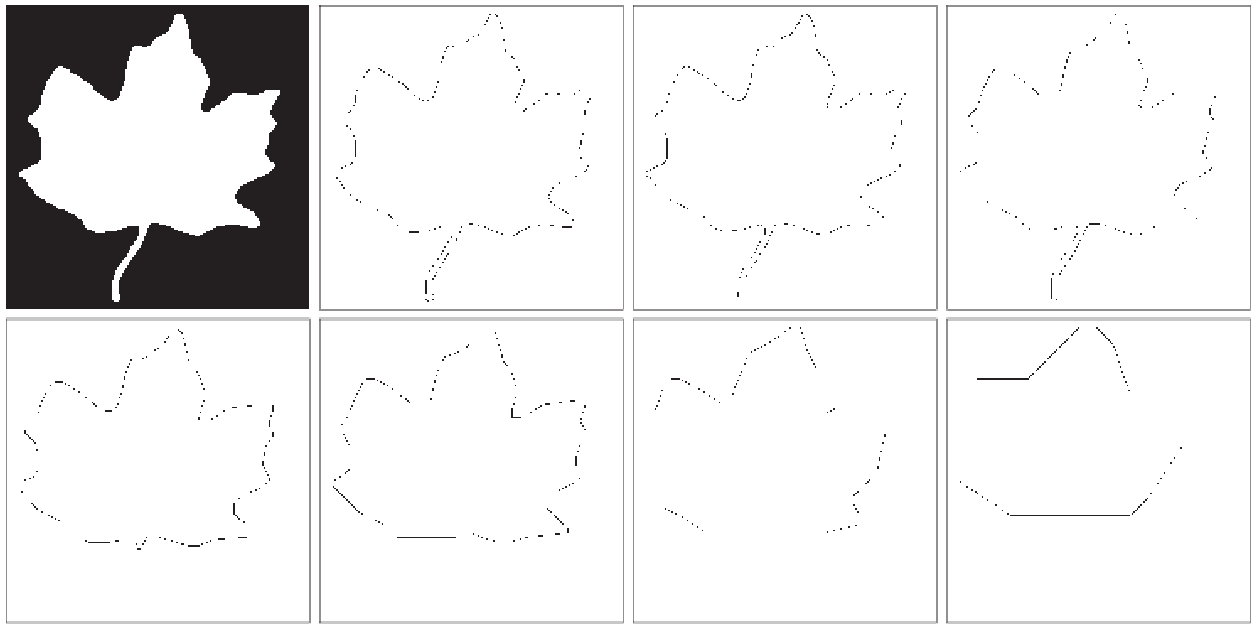
Minimum perimeter polygon (MPP)

a b c

FIGURE 12.8 (a) Region (dark gray) resulting from enclosing the original boundary by cells (see Fig. 12.7). (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.

12.1 Polygon Approximation

- Represent an object boundary by a polygon

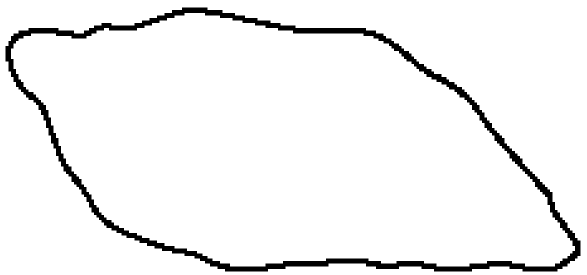


MPP

FIGURE 12.9 (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.

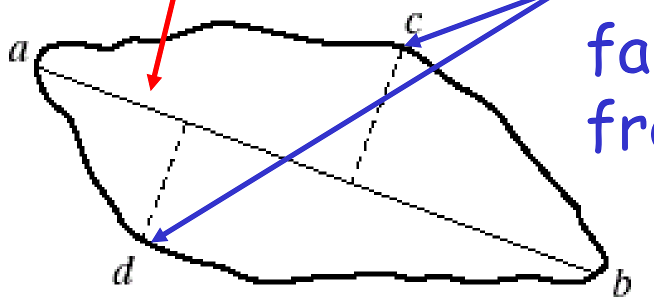
Polygon Approximation: Splitting Techniques

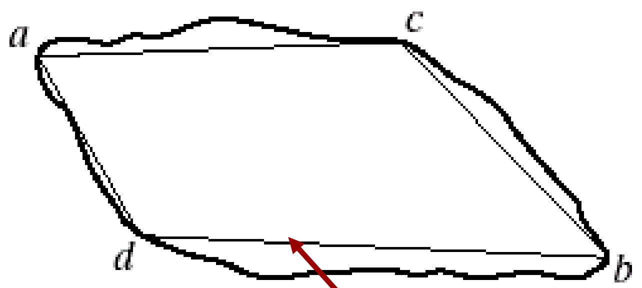
Object boundary

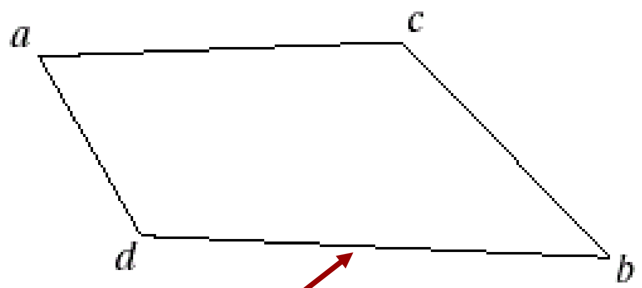


1. Find the line joining two extreme points

2. Find the farthest points from the line







3. Draw a polygon

WRIGHT STATE UNIVERSITY

Boundary Descriptors

- **Perimeter**
  - Have to be careful about spacing in horizontal/vertical vs. diagonal neighbors
- **Diameter**

$$Diam(B) = \max_{i,j} \left[D(p_i, p_j) \right]$$

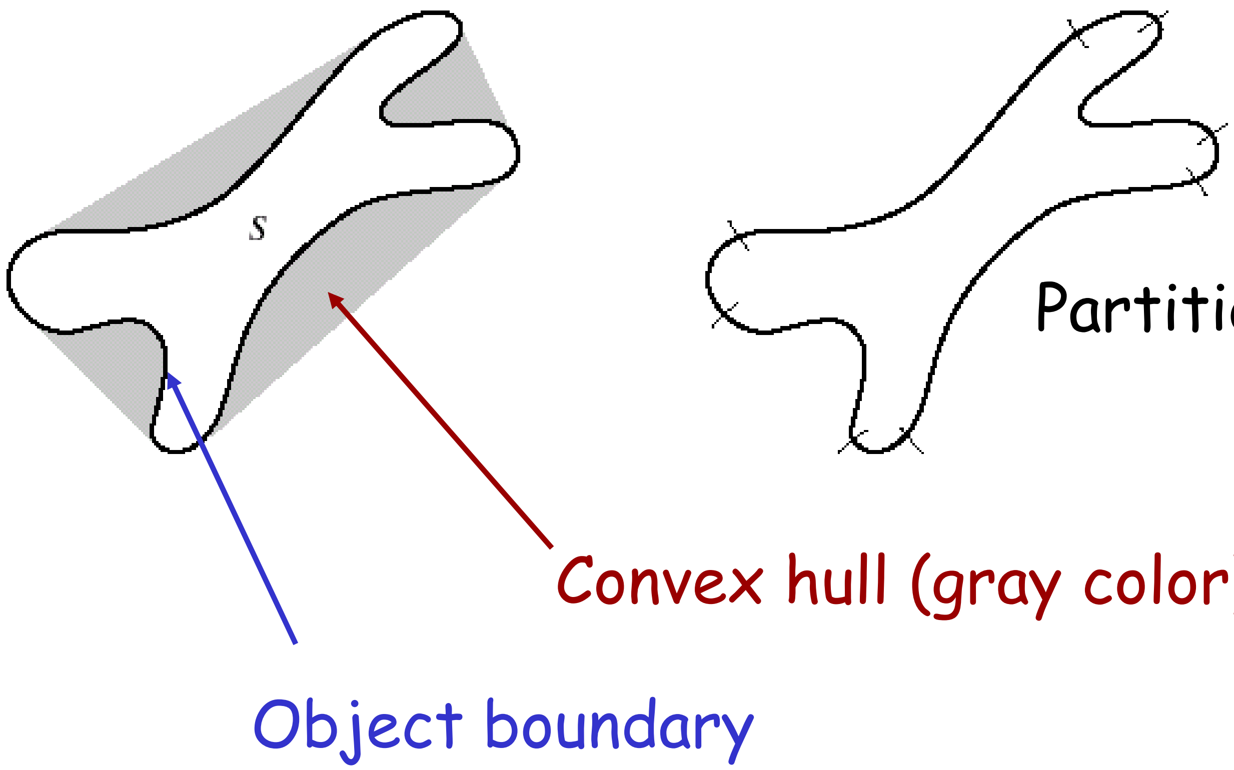
- p_i and p_j are points on the boundary (D: Distance)

  - ➢ **Major Axis:** Orientation of the line connecting the two extreme points
  - ➢ **Minor Axis:** Perpendicular to the major axis
- Eccentricity
  - ➢ Ratio of major to minor axis lengths

WRIGHT STATE
UNIVERSITY

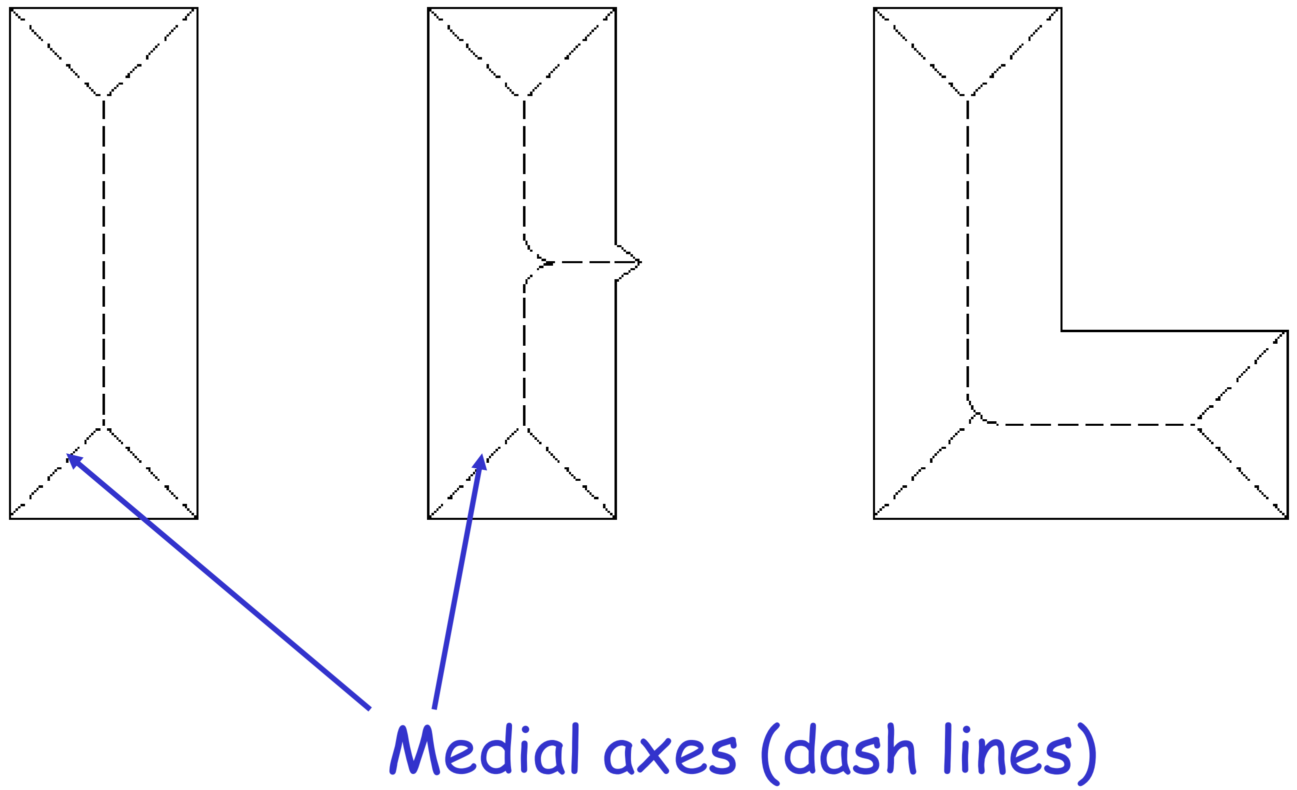
Boundary Segments

Concept: Partition object boundary by using vertices (transition of breaks) of a convex hull (H: Smallest convex set containing S)



12.1 Skeletons

Obtained from thinning or skeletonizing processes



a b c

FIGURE 12.12 Medial axes (dashed) of three simple regions.

Example: Skeletons Obtained from the Thinning Algorithm

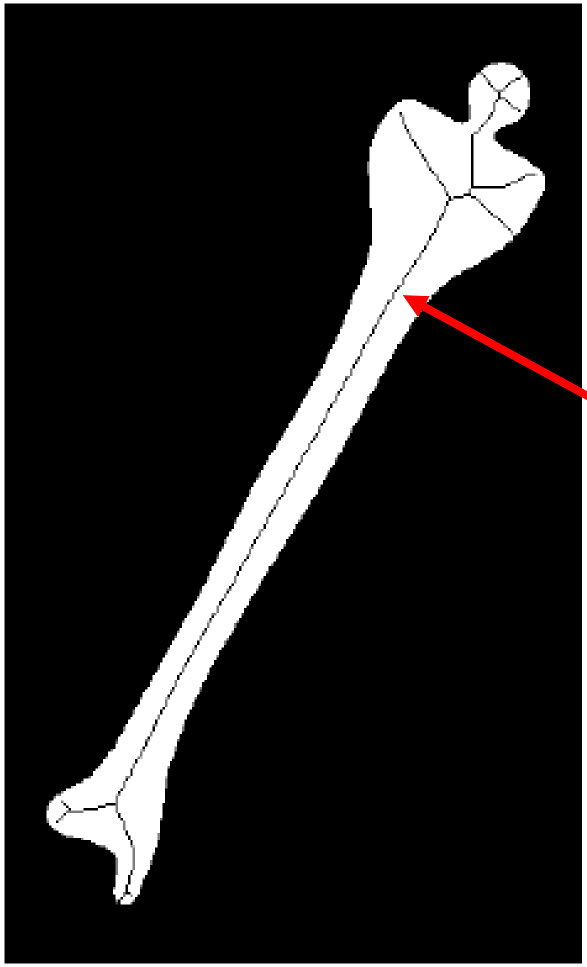
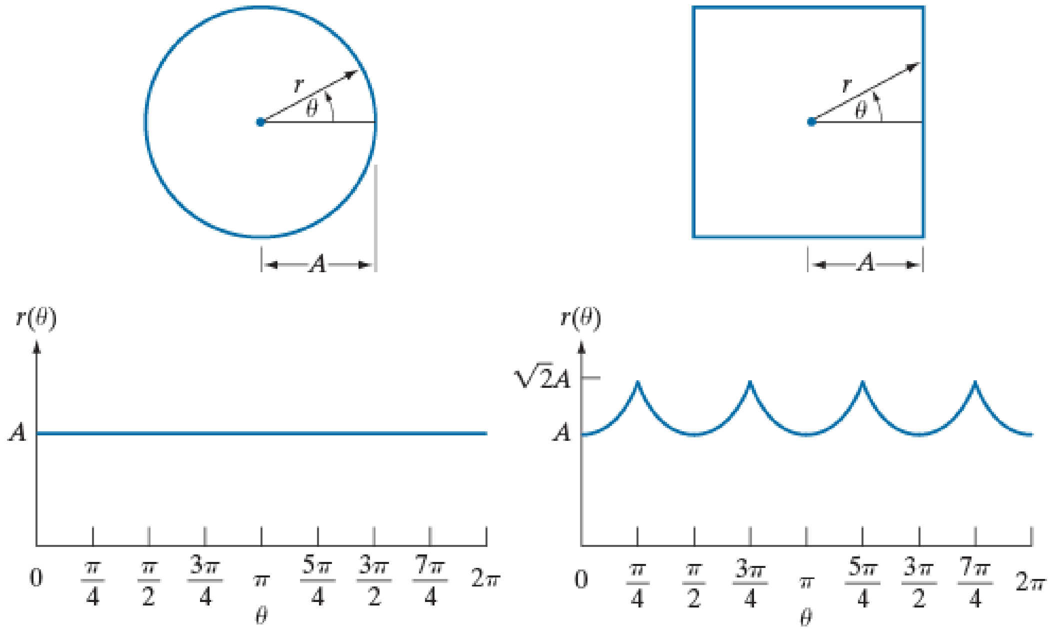


FIGURE 11.10 Human leg bone and skeleton of the region shown superimposed.

Signatures

Represent a 2-D object boundary in terms of a 1-D function of radial distance with respect to Θ.

- Distance from center of mass vs. angle
- Invariant to translation
- Can be made invariant to rotation by choosing starting point



a b

FIGURE 12.10 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 $\pi/4 < \theta \le \pi/2$.

Signatures Example

- How to compare two objects?
- What is the feature vector?
- Advantages vs. chain code? → Simplicity

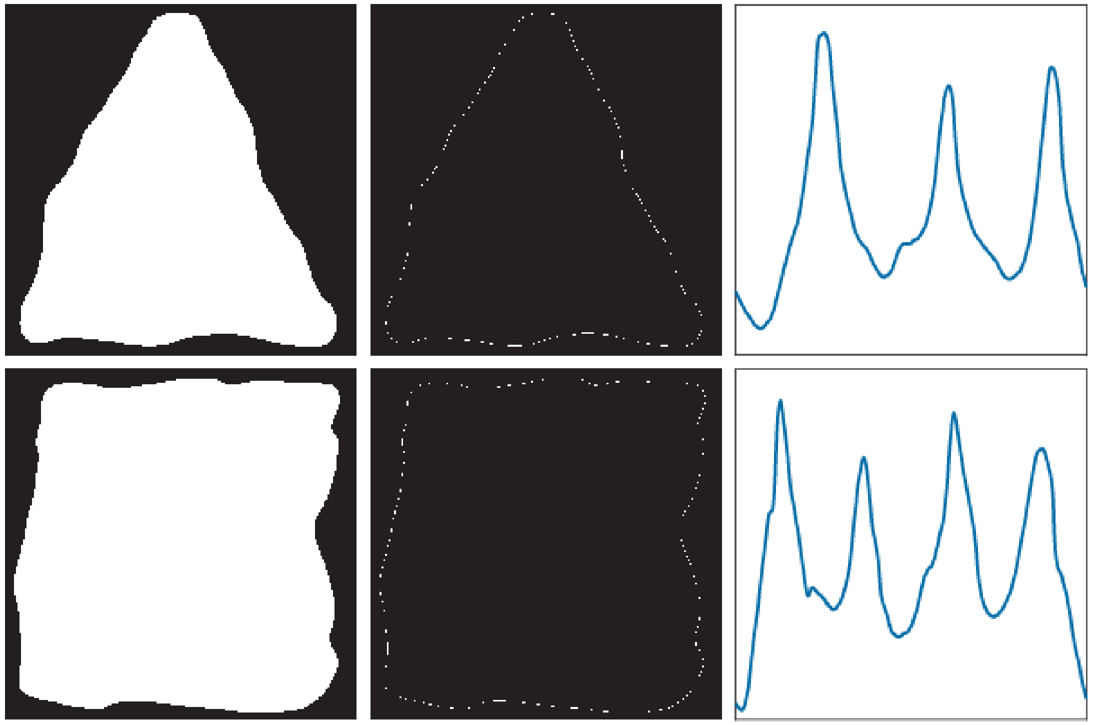


FIGURE 12.11 (a) and (d) Two binary regions, (b) and (e) their external boundaries, and (c) and (f) their corresponding $r(\theta)$ signatures. The horizontal axes in (c) and (f) correspond to angles from 0° to 360°, in increments of 1°.

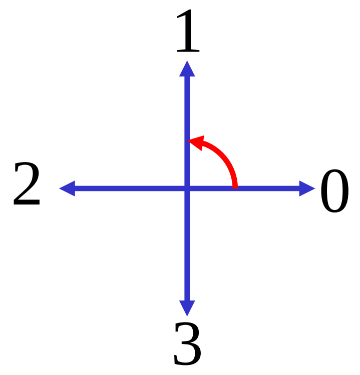
12.2 Shape Number (From Chain Codes)

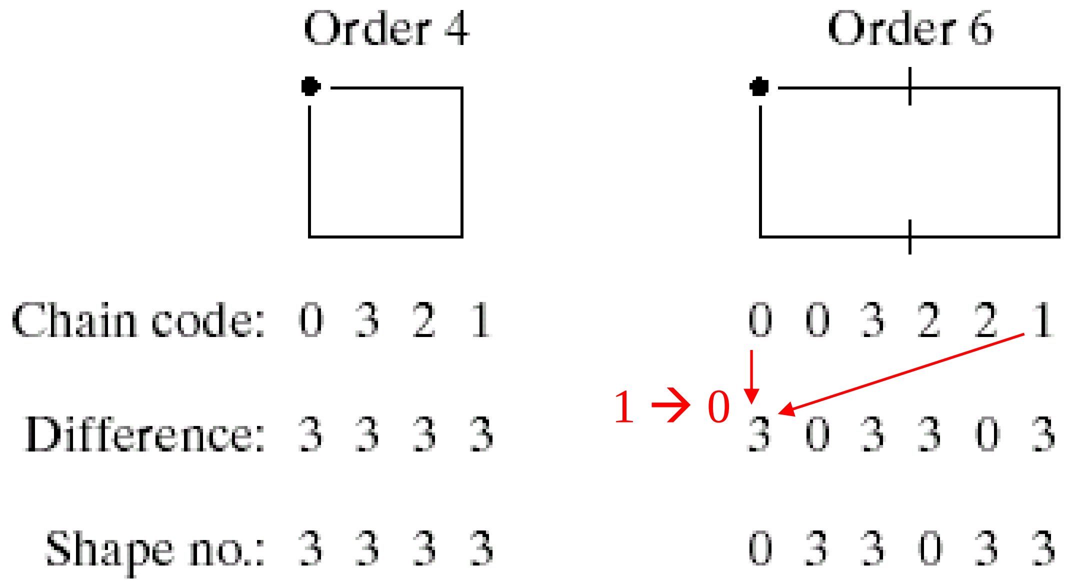
Shape number of the Boundary Definition:

- The first difference of smallest magnitude

The order n of the Shape Number:

- The number of digits in the sequence



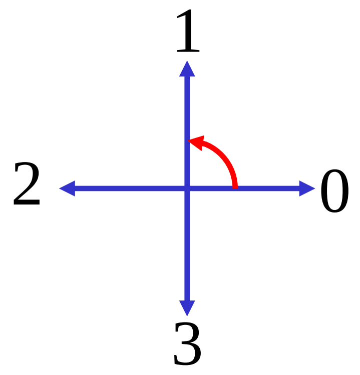


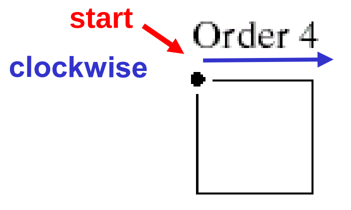
Chain code :	The first difference
0 → 1	1
0 → 2	2
0 → 3	3
2 → 3	1
2 → 0	2
2 → 1	3

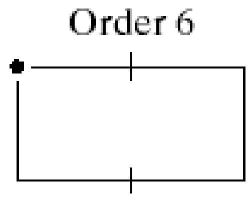
and so on..

Shape Number (cont.)

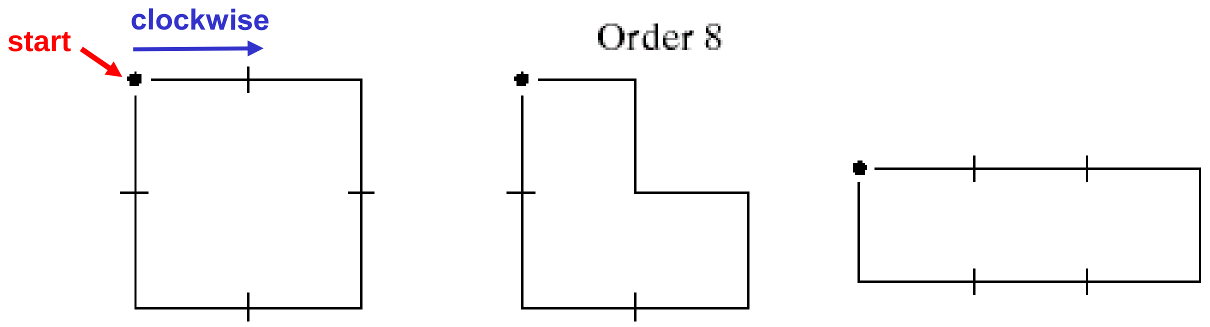
Shape numbers of orders 4, 6 and 8





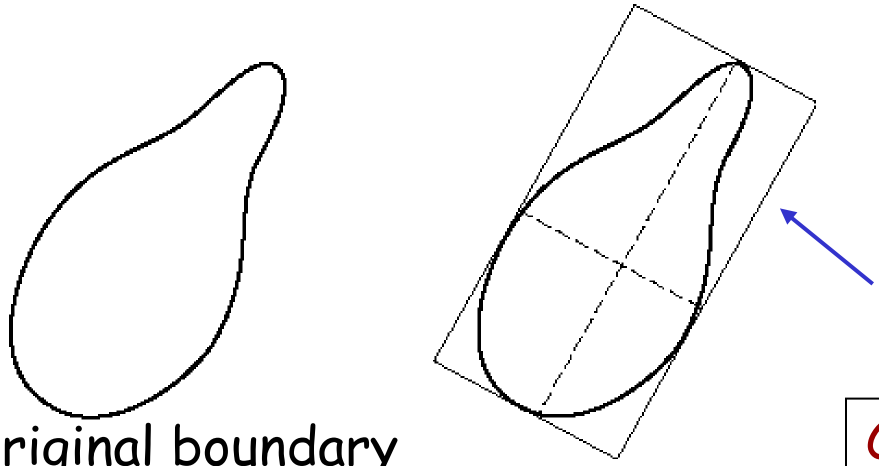


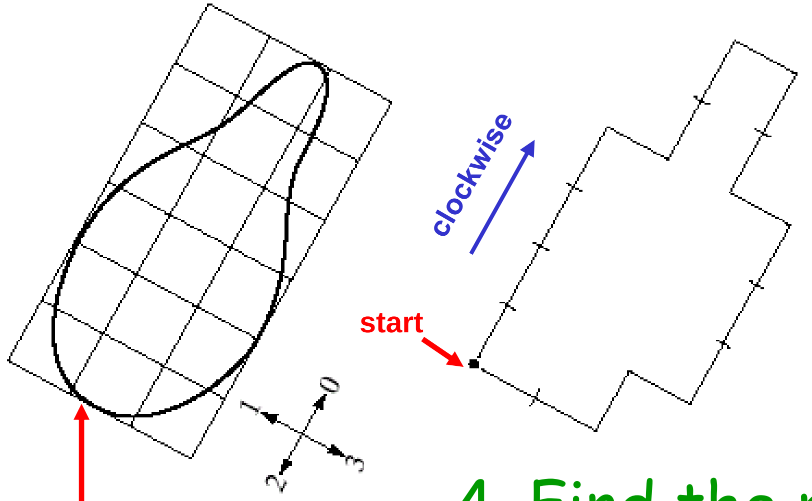
	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3



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

Example: Shape Number





Chain code:

0 0 0 0 3 0 0 3 2 2 3 2 2 2 1 2 1 1

First difference:

3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

Shape No.

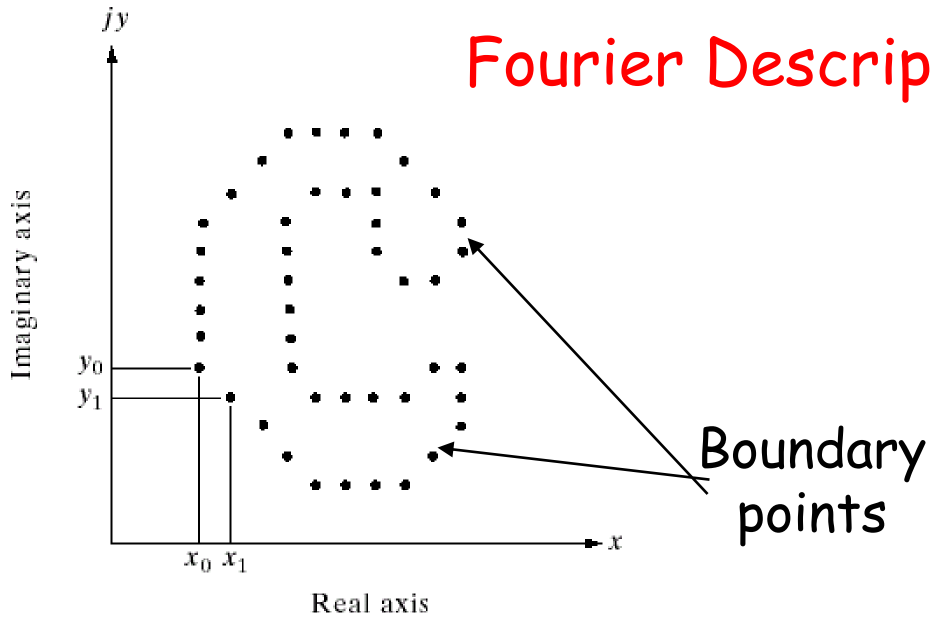
0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0 3

12.2 Fourier Descriptor

Fourier descriptor: View a coordinate (x, y) as a complex number (x = real part and y = imaginary part) then apply the Fourier transform to a sequence of boundary points.

Let $s(k)$ be a coordinate of a boundary point k:

$$s(k) = x(k) + jy(k)$$



Fourier Descriptor :

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

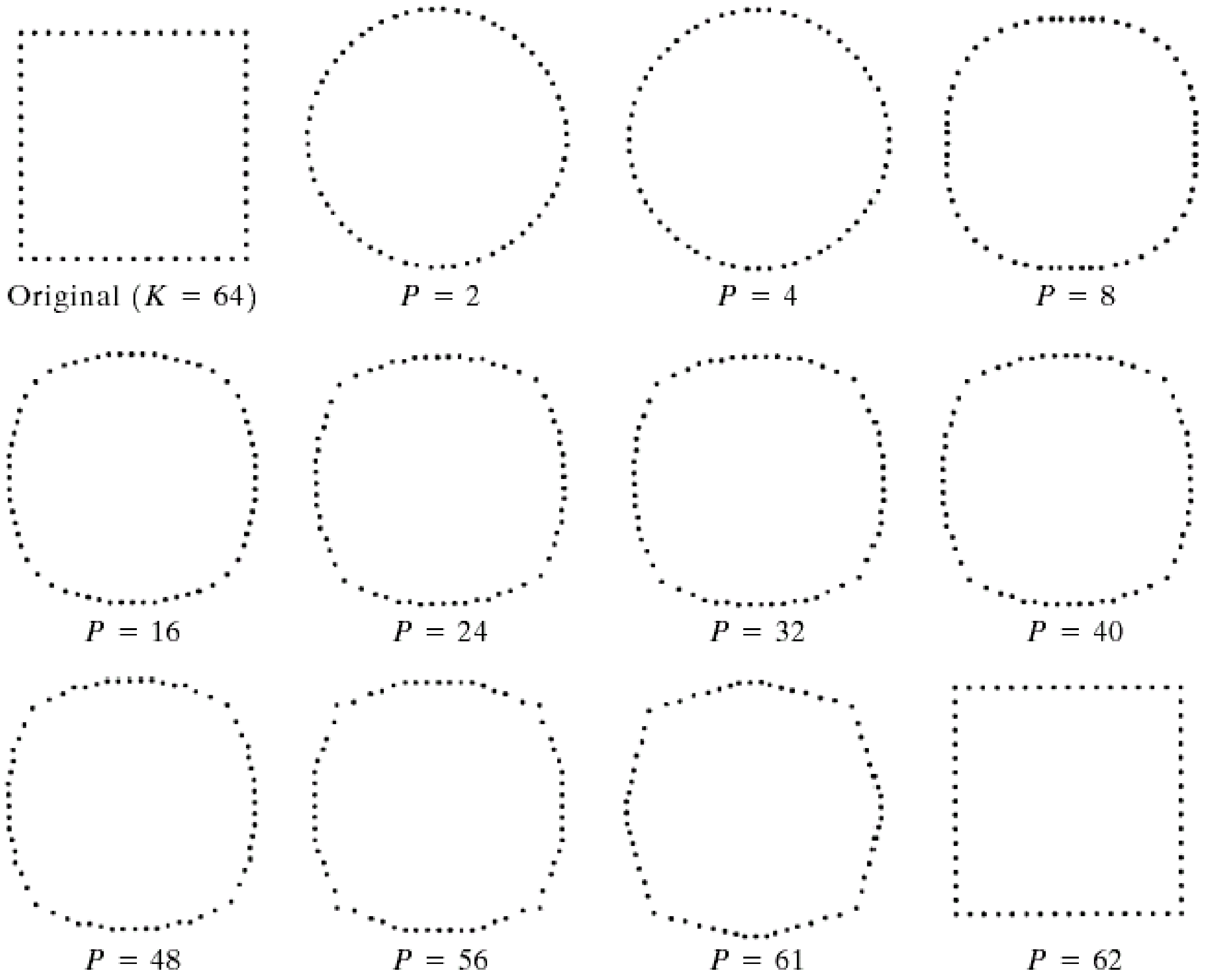
Reconstruction formula

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

This is now a 1D problem!

Example: Fourier Descriptor

Examples of reconstruction from Fourier descriptors

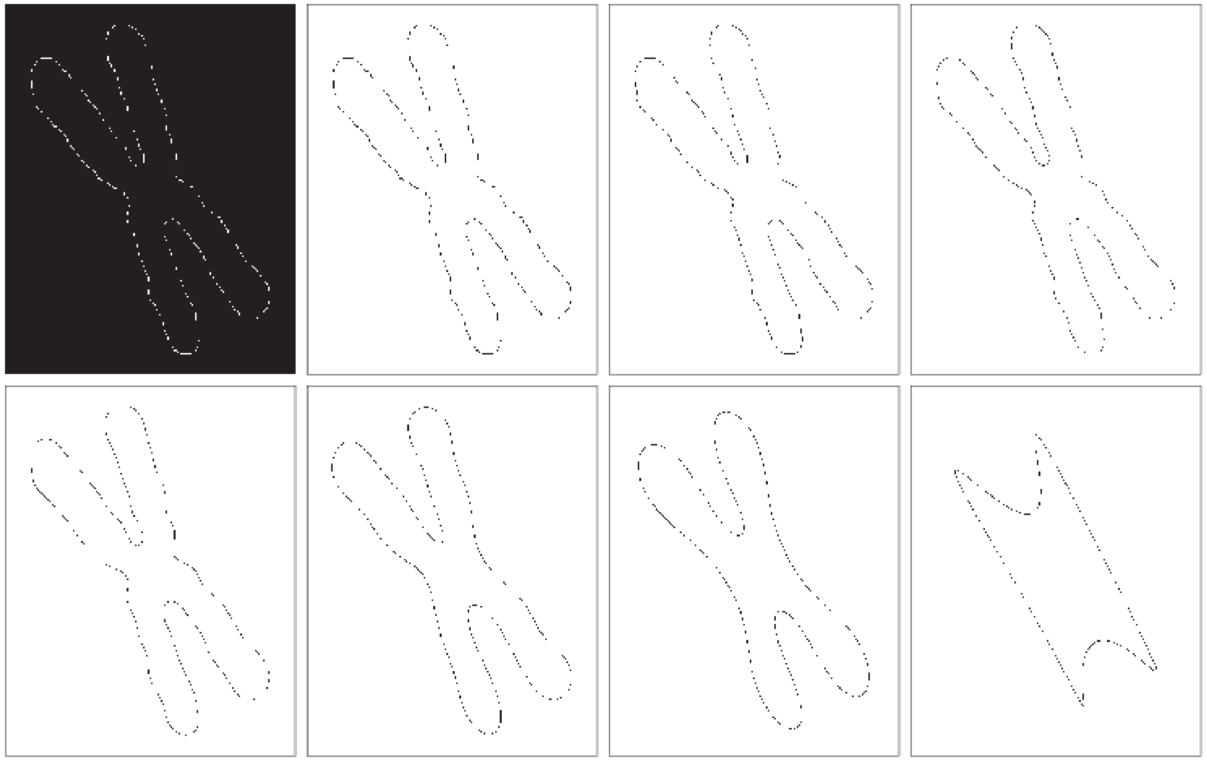


$$\hat{s}(k) = \frac{1}{K}\sum_{k=0}^{P-1} a(u)e^{2\pi uk/K}$$

P: Number of Fourier coefficients used to reconstruct the boundary

Shape Approximation with Fourier

- Use first P coefficients to reconstruct the shape
  - Retains most important shape properties
  - Gets rid of noise and details



a b c d
e f g h

FIGURE 12.19 (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.

WRIGHT STATE UNIVERSITY

Comparing Objects with Fourier

- Being able to represent any shape with only a few numbers is an advantage
- How about invariance of Fourier Descriptors?
  - They are not directly invariant but can be made so

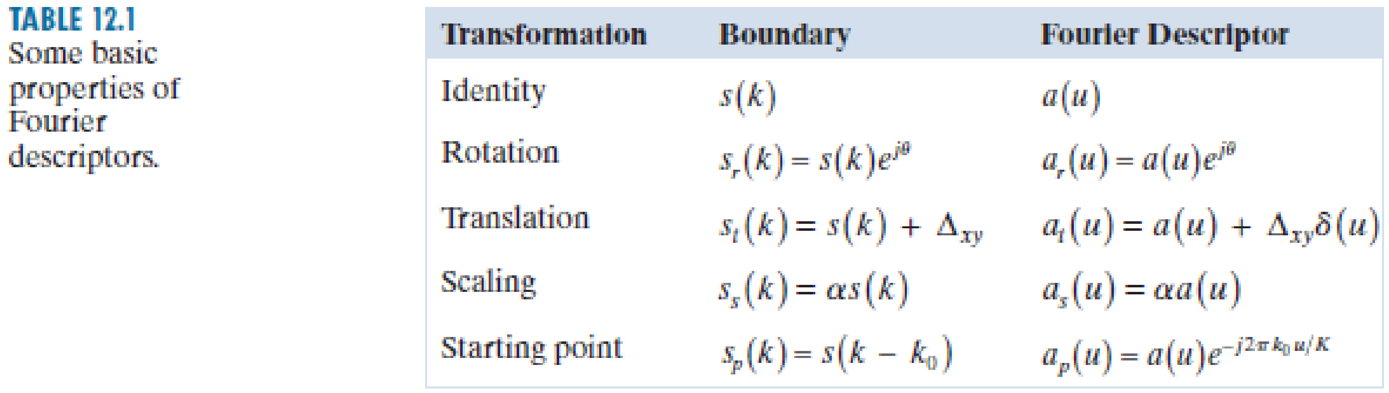


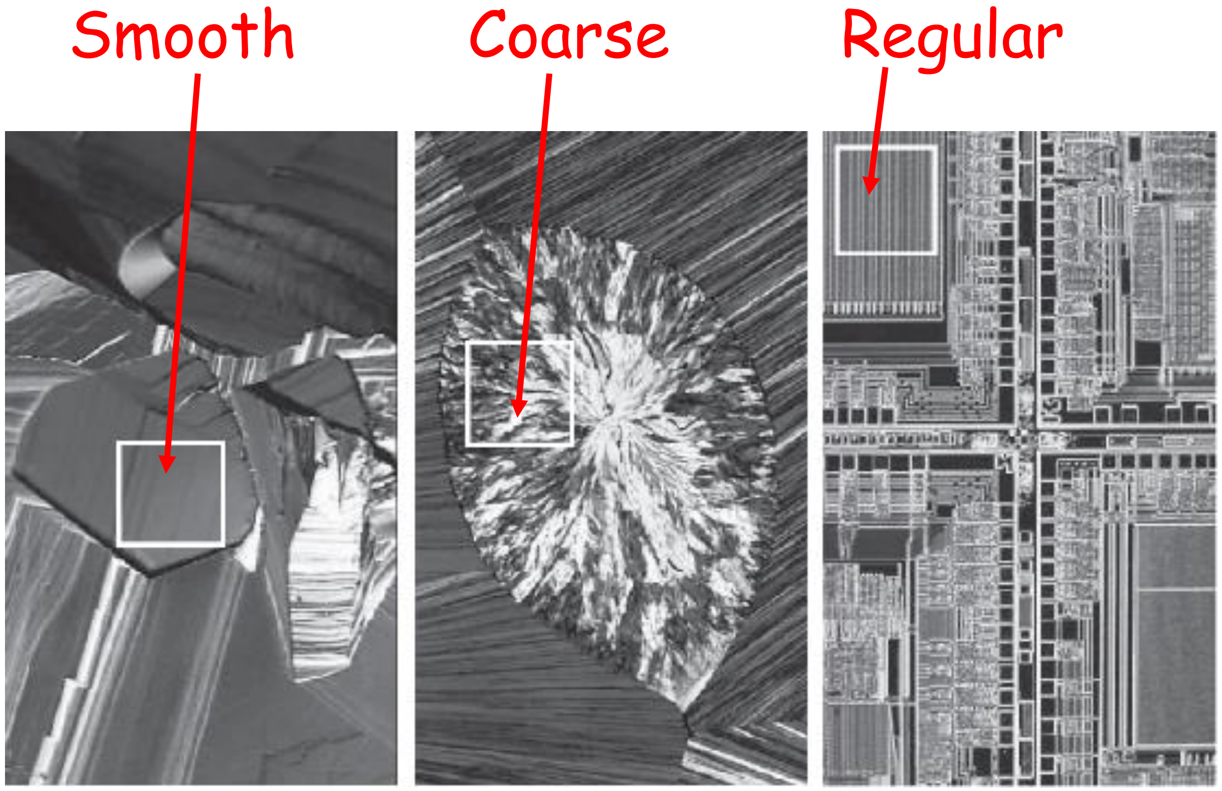
TABLE 12.1 Some basic properties of Fourier descriptors.

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

- Ignoring the 0th coefficient and using only magnitudes gets invariance to translation/starting point and rotation

12.3 Texture

- What is the definition of texture?
  - Smoothness, roughness, regularity



a b c

FIGURE 12.29 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.)

Characterizing Texture

- Measuring texture properties:
  - Use statistics
  - Let z be a random variable with possible values 0,1,2,…, L-1.
  - Its n'th moment about its mean

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

- Note $\mu_0=1$ and $\mu_1=0$. μ_2 is the variance
- μ_3 measures skewness of the pdf.
- μ_4 measures flatness of the pdf.

- Entropy:

$$e(z) = -\sum_{i=0}^{L-1} p(z_i)\log_2 p(z_i)$$

Example

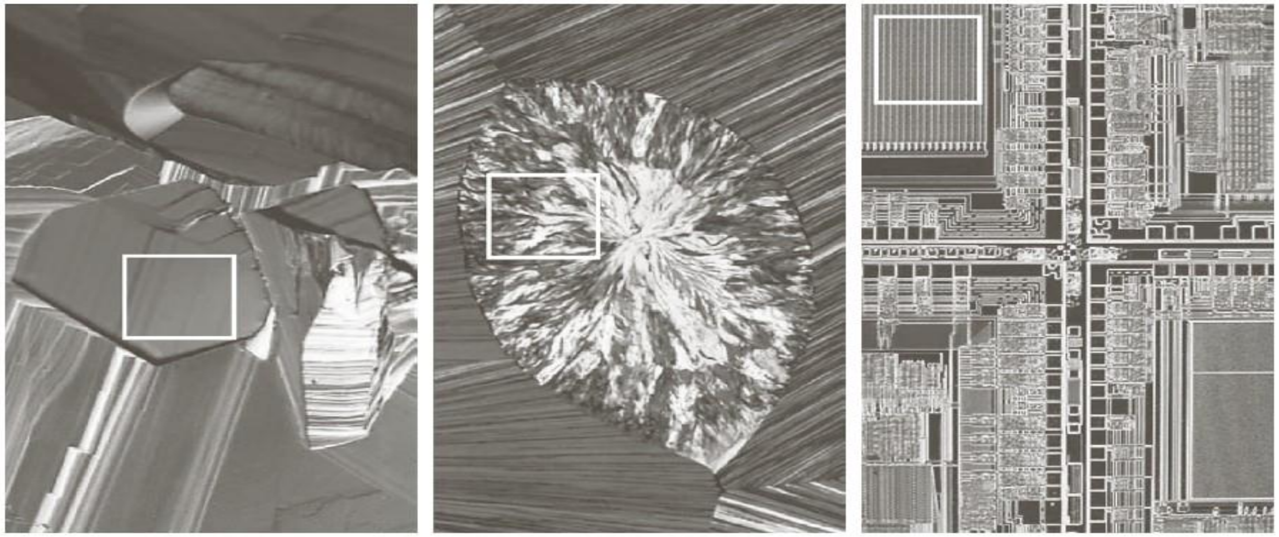

a b c

FIGURE 12.29 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.)

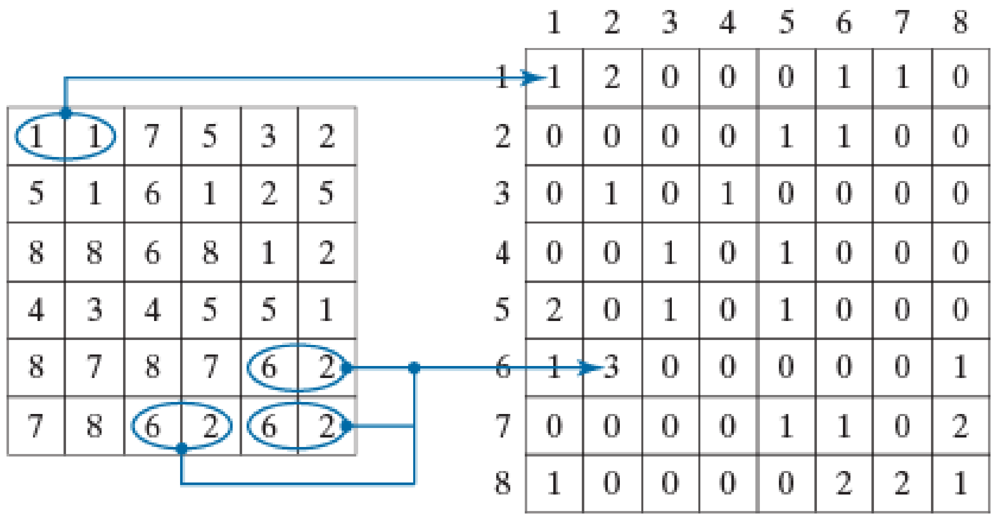
TABLE 12.2
Statistical texture measures for the subimages in Fig. 12.29.

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

Co-Occurrence Matrix

- How often does the **intensity level** i occurs at a specified position **from intensity level** j ?

- Example: One pixel immediately to the right



1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

Image *f*

	1	2	3	4	5	6	7	8
1	1	2	0	0	0	1	1	0
2	0	0	0	0	1	1	0	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	2	0	1	0	1	0	0	0
6	1	3	0	0	0	0	0	1
7	0	0	0	0	1	1	0	2
8	1	0	0	0	0	2	2	1

Co-occurrence matrix **G**

FIGURE 12.30 How to construct a co-occurrence matrix.

Axis: xy

Axis: intensity level
Intensity range 1-8

Co-Occurrence Matrix Examples

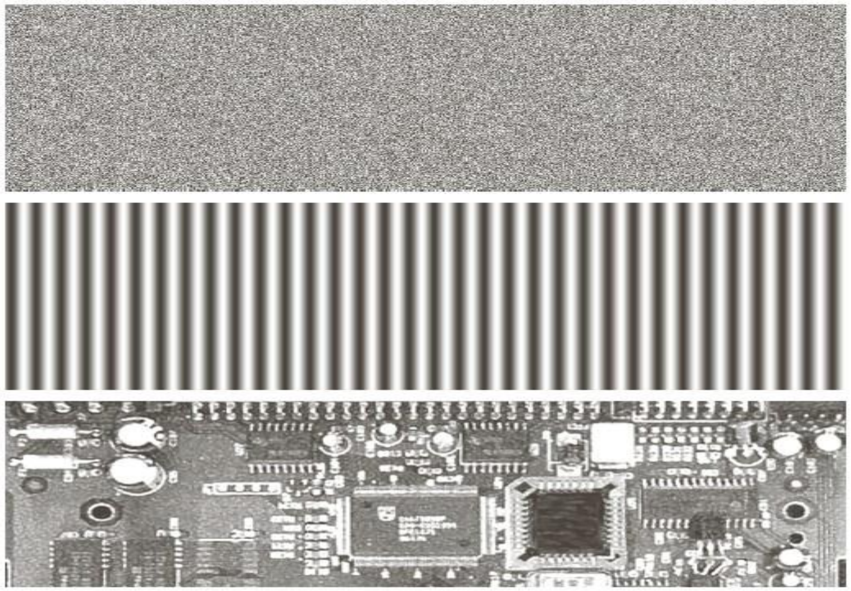

a
b
c

FIGURE 12.31 Images whose pixels have (a) random, (b) periodic, and (c) mixed texture patterns. Each image is of size 263×800 pixels.

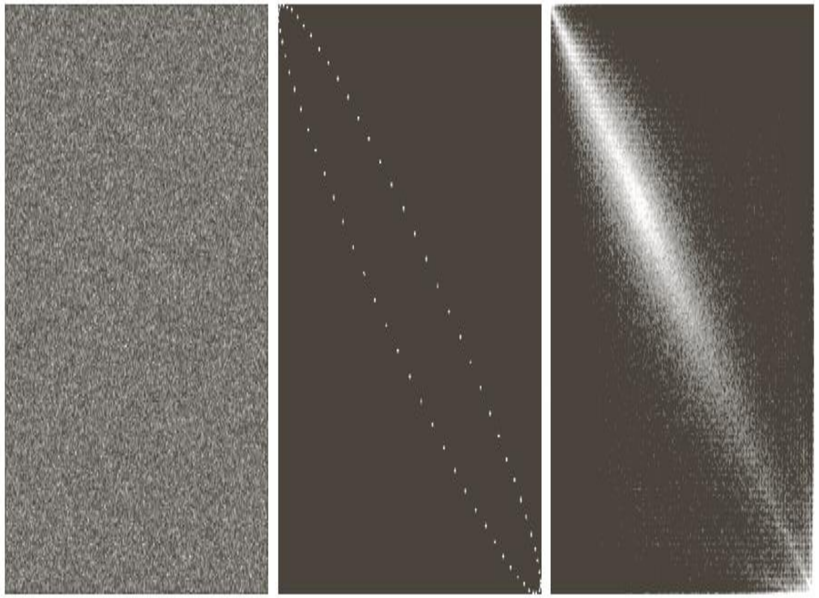

a b c

FIGURE 12.32 256×256 co-occurrence matrices $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, corresponding from left to right to the images in Fig. 12.31.

WRIGHT STATE UNIVERSITY

Acknowlegements

The slides are primarily based on the figures and images in the Digital Image Processing textbook by Gonzalez and Woods:

- http://www.imageprocessingplace.com/DIP-3E/dip3e_book_images_downloads.htm

In addition, slides have been adopted and modified from the following sources:

- http://gear.kku.ac.th/~nawapak/178353.html
- https://cs.nmt.edu/~ip/index.html