Representation and Description for image Processing

Segmentation

↓

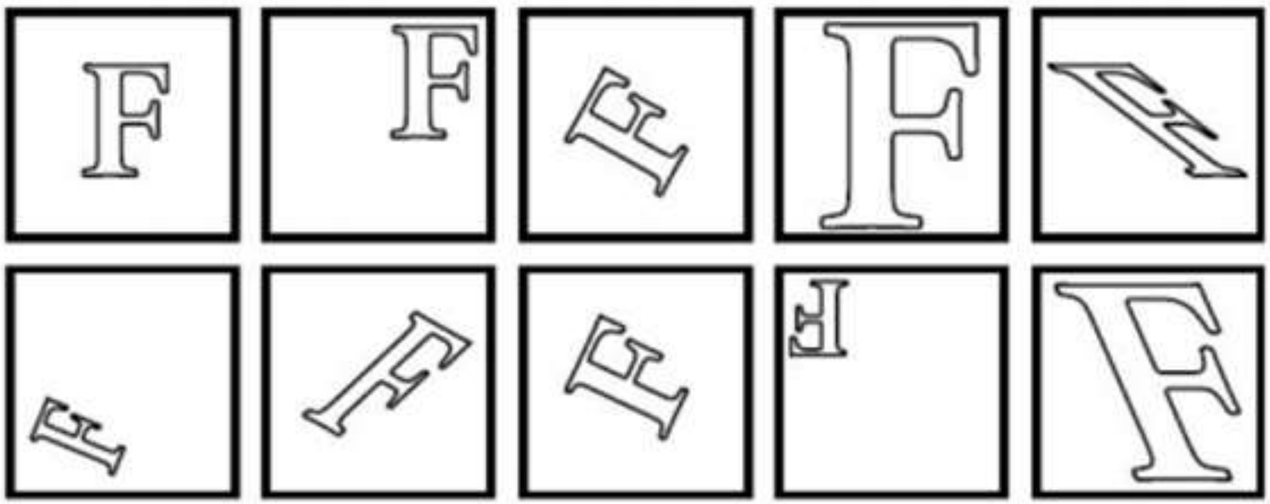
Raw data (pixels) → Representations

- → external characteristics (boundary)
  - shape characteristics
- → internal characteristics (pixels comprising the region)
  - regional properties such as color & texture

↓

Computation of descriptors

Is invariance needed?



- Translation invariance
- Scale invariance
- Rotation invariance

Technique of representation

- **Chain Codes**
- **Polygonal Approximations**
  - Minimum perimeter polygons
  - Merging techniques
  - Splitting techniques
- **Signatures**
- **Boundary Segments**
- **Skeletons**

Chain Codes

<u>Problem1</u>:

- Long chains of codes
- Easily disturbed by noise, and sidetracked

Solution:

- Resampling using larger grid spacing

<u>Problem2</u>:

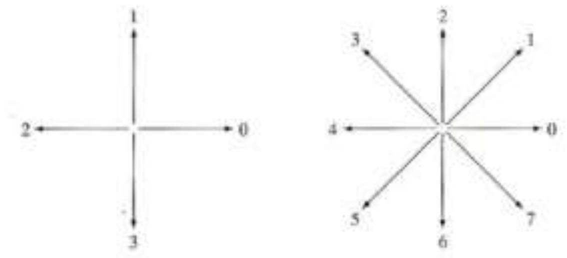
- start point

Solution:

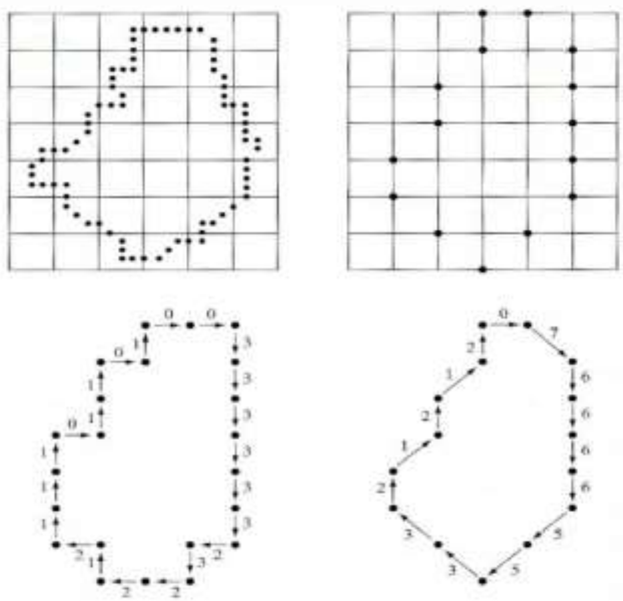
- Normalizations

Before resample: 000000766…1111

After resampling: 076666…12

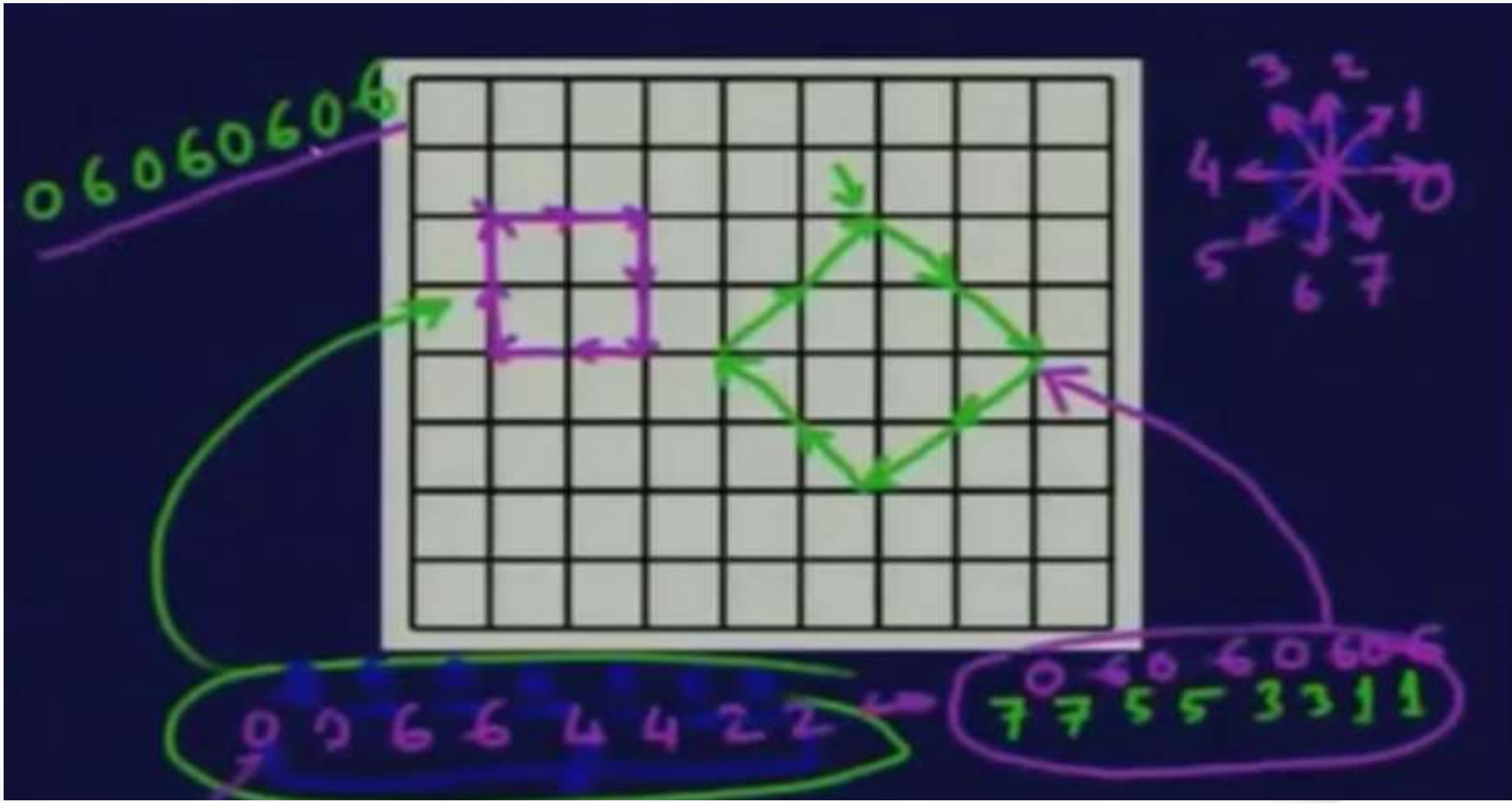


a b
FIGURE 11.1 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.



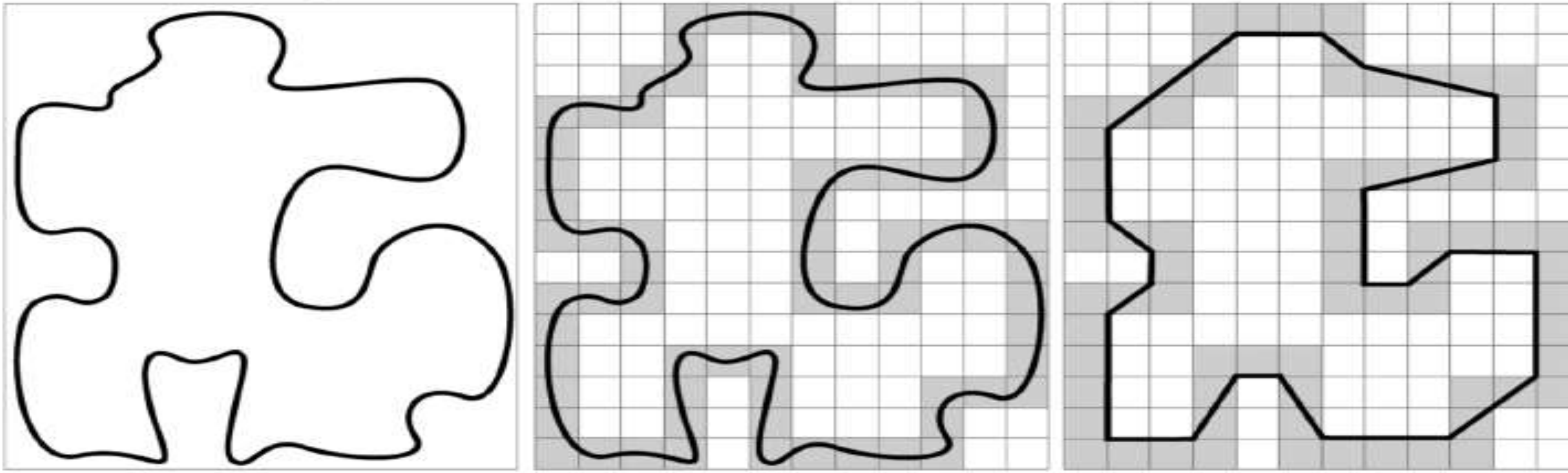
a b
c d
FIGURE 11.2 (a) Digital boundary with resampling grid superimposed. (b) Result of resampling. (c) 4-directional chain code. (d) 8-directional chain code.

- normalized
  - **circular sequence**
  - Ex. First difference of 4-direction **chain code** 10103322 is 3133030.
  - 2 10103322
  - 33133030


06060606
3 2
1
4 0
5
6 7
0 0 6 6 4 4 2 2
7 7 5 5 3 3 1 1

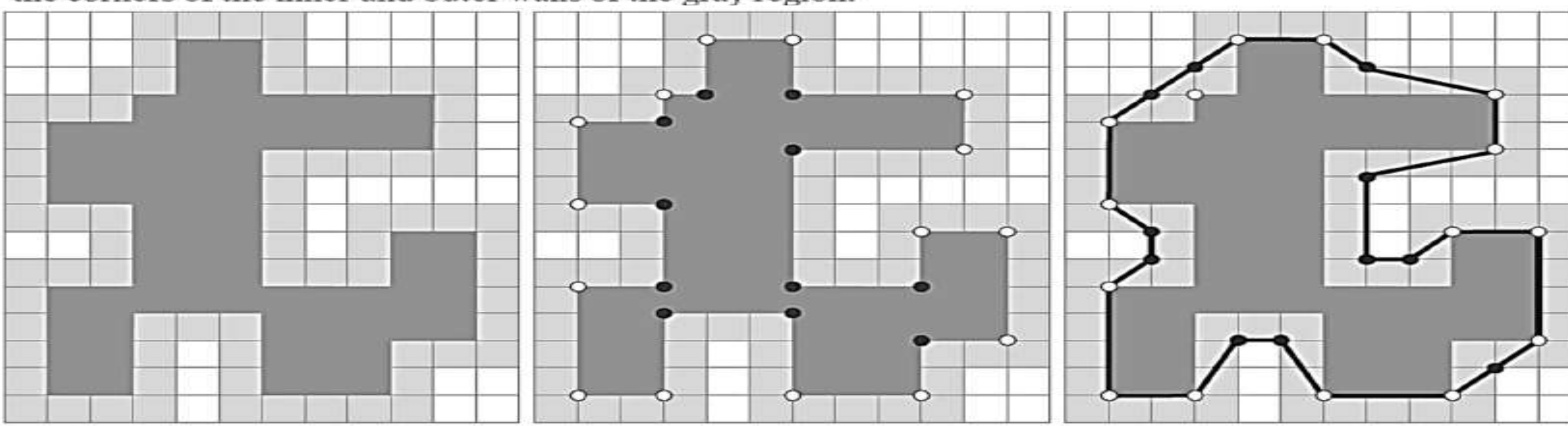
Polygonal Approximations

- Determine which points on the boundary to use
- Minimum perimeter polygons
  - Choose an appropriate grid
    - The boundary is enclosed by a set of concatenated cells
  - Allow the boundary to shrink as a rubber band
    - The maximum error per grid cell is $\sqrt{2}d$, where d is the dimension of a grid cell



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.



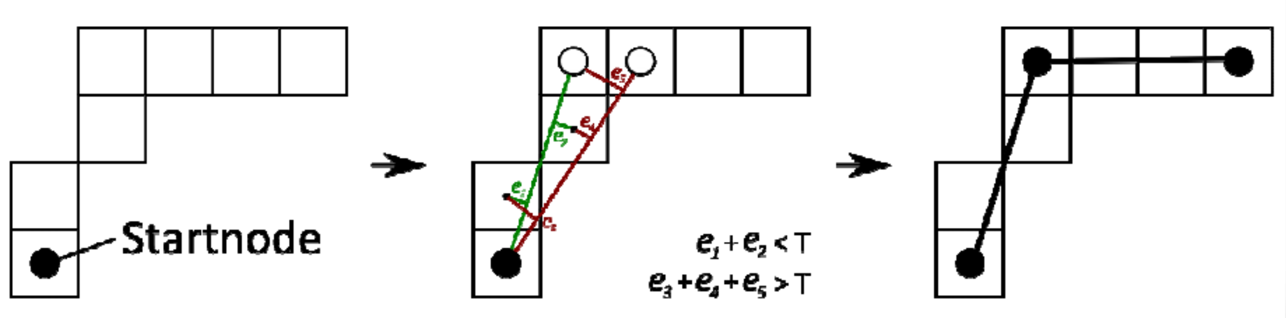
a b c

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.

Merging Techniques

- least square error line fit

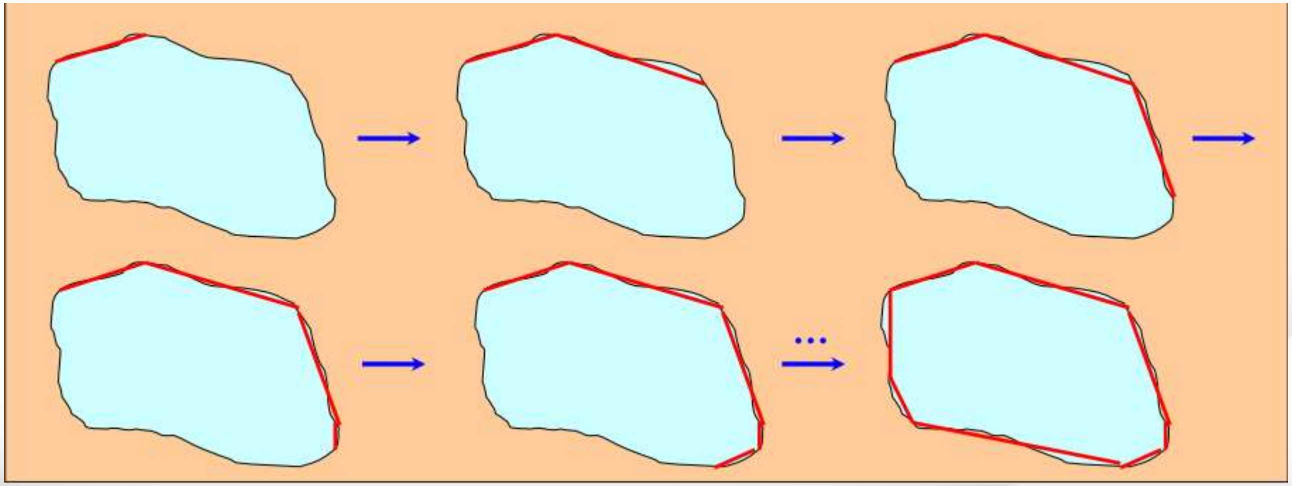
1. Merge points along a boundary until the least square error line fit of the points merged so far exceeds a threshold
2. Record the the two end point of the line
3. Repeat Steps 1 and 2 until all boundary points are processed .



Merging Techniques

- least square error line fit

1. Merge points along a boundary until the least square error line fit of the points merged so far exceeds a threshold
2. Record the the two end point of the line
3. Repeat Steps 1 and 2 until all boundary points are processed .

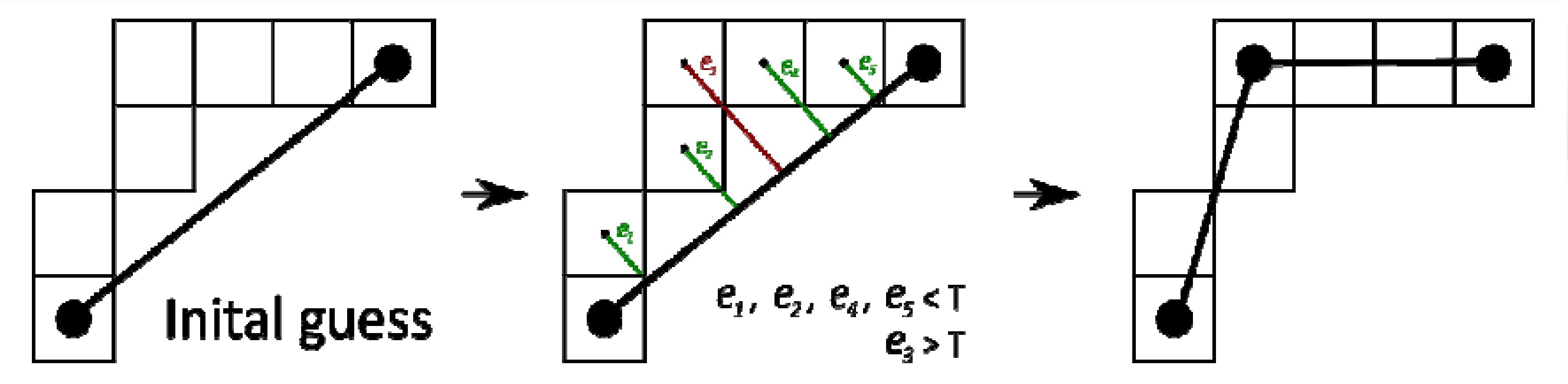


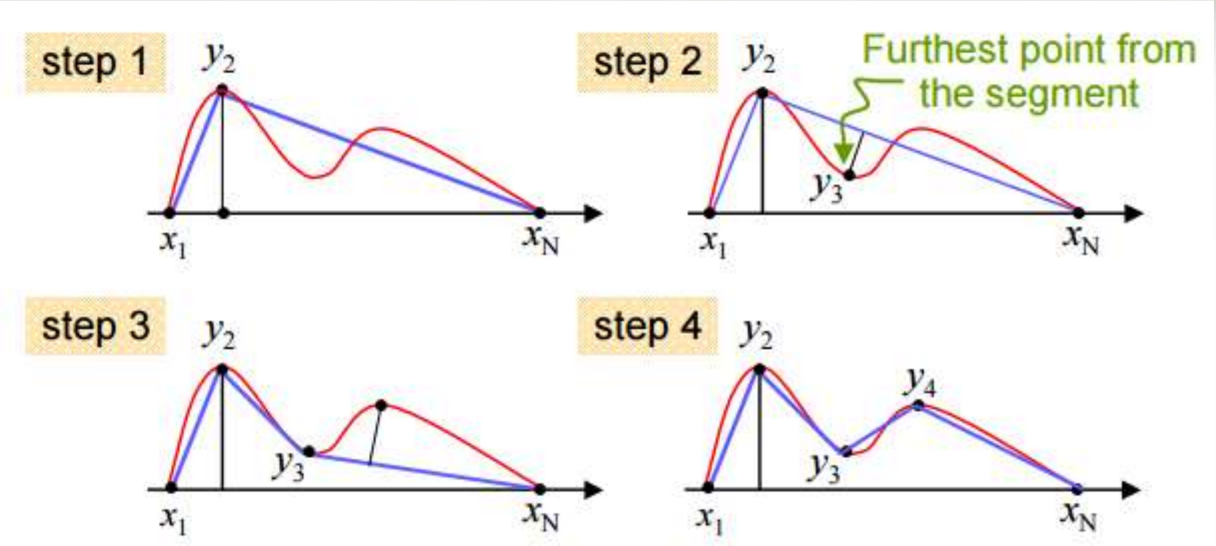
problem

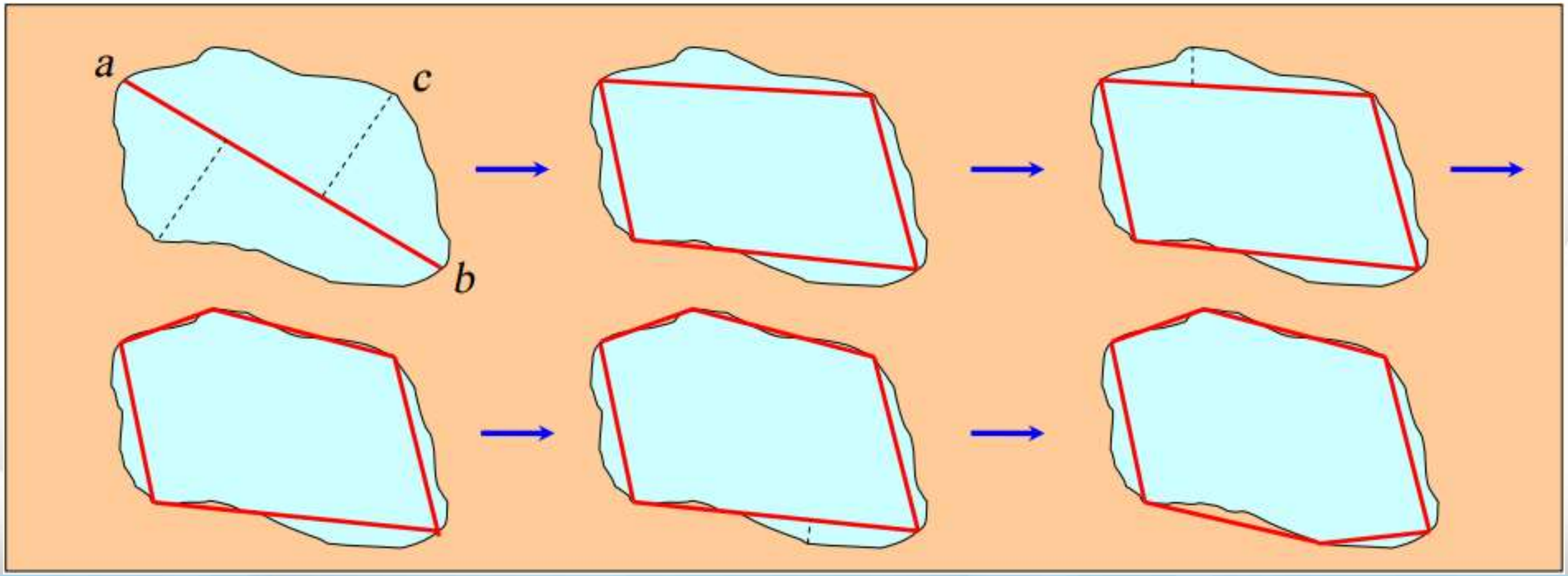
- Merging technique problem:
  - No guarantee for corner detection
- Solution:
  - Splitting: to subdivide a segment successively into two parts until a given criterion is satisfied.
  - Objective: seeking prominent inflection points.

Splitting techniques:

1. Start with an initial guess, e.g., based on majority axes
2. Calculate the orthogonal distance from lines to all points
3. If maximum distance > threshold, create new vertex there
4. Repeat until no points exceed criterion




step 1
y_2
x_1
x_N
step 2
y_2
Furthest point from the segment
y_3
x_1
x_N
step 3
y_2
y_3
x_1
x_N
step 4
y_2
y_4
y_3
x_1
x_N

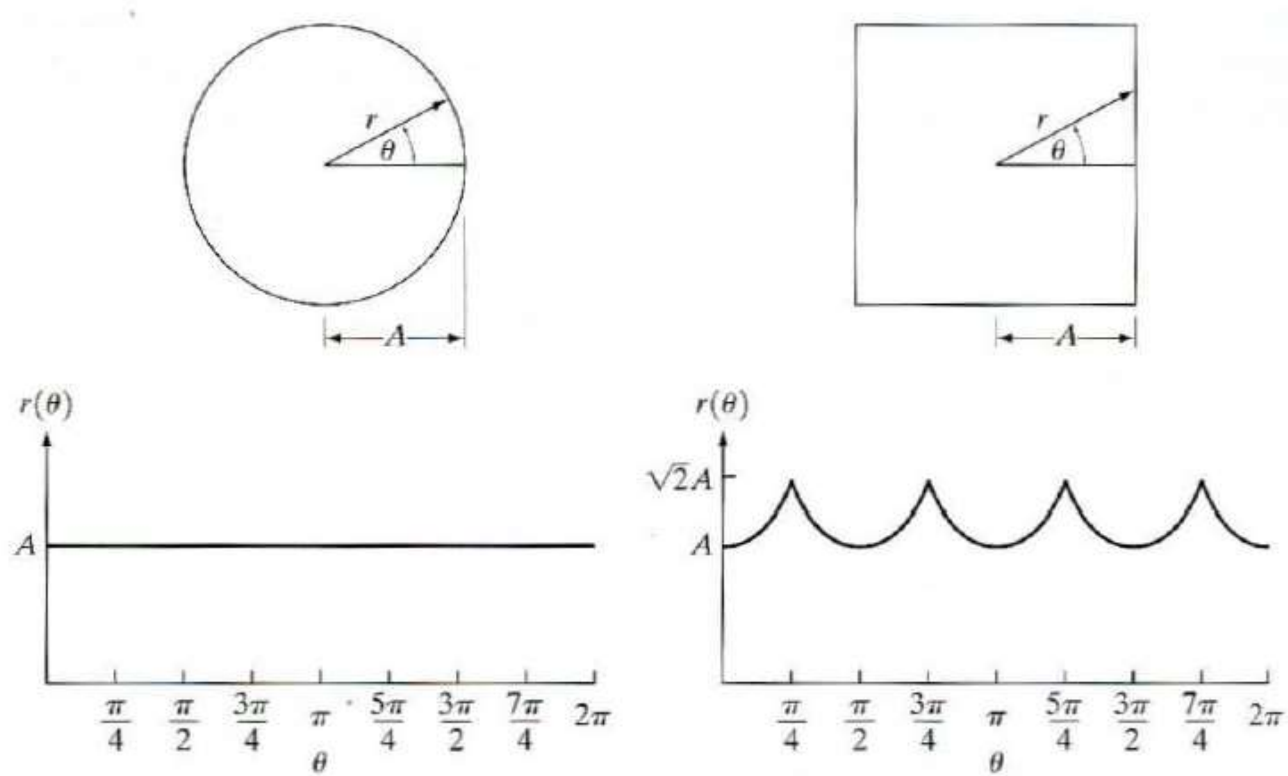

a
c
b

Signatures

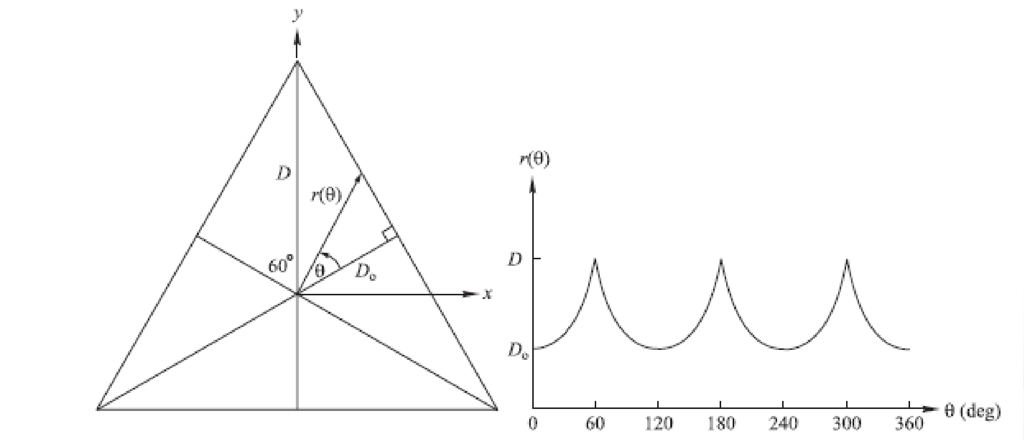
- Signature: a 1D functional representation of a boundary
- To generate:
  - Plot the distance from the centroid to the boundary as a function of angles.
- The signature is often unique for a region
  - We can distinguish the region by its signature
- Independent of translation, but not rotation & scaling .

a b

FIGURE 11.5 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 $\pi/4 < \theta \le \pi/2$.



- $r(\Theta) = D/\cos\Theta \quad 0 \le \Theta < 60$
- $= D/\cos(120-\Theta) \quad 60 \le \Theta < 120$
- $= D/\cos(180-\Theta) \quad 120 \le \Theta < 180$
- $= D/\cos(240-\Theta) \quad 180 \le \Theta < 240$
- $= D/\cos(300-\Theta) \quad 240 \le \Theta < 300$
- $= D/\cos(360-\Theta) \quad 300 \le \Theta < 360$



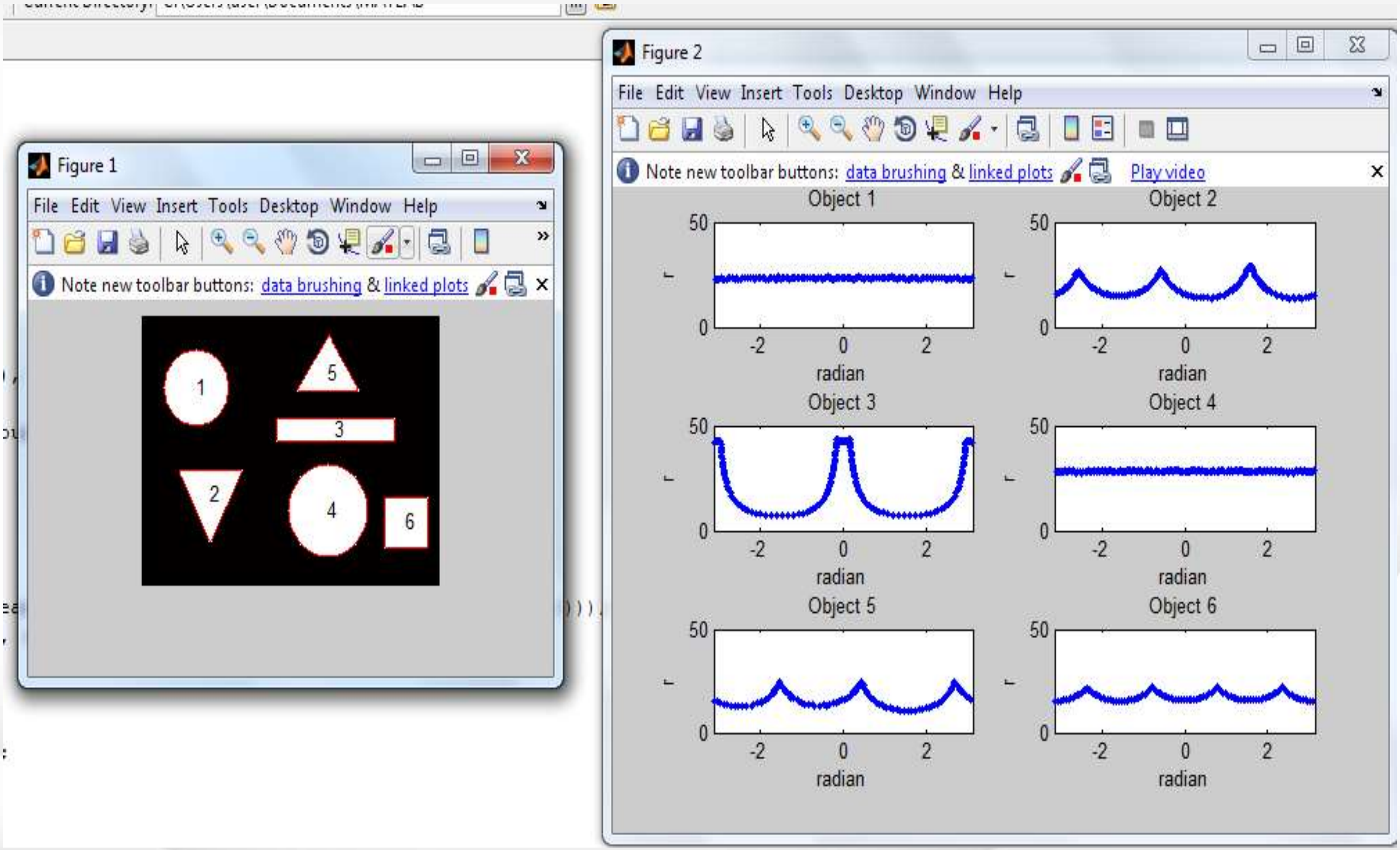
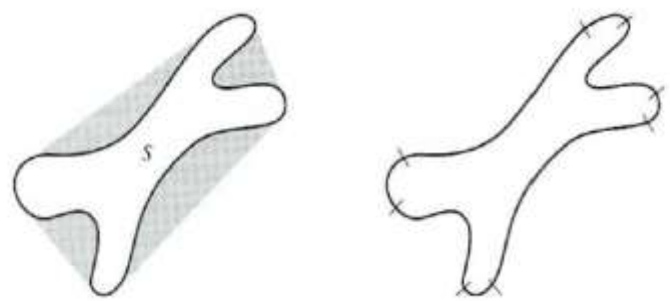


Figure 1
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots
1
5
3
2
4
6
Figure 2
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots
Play video
Object 1
Object 2
Object 3
Object 4
Object 5
Object 6
50
0
-2
0
2
radian

Boundary segments

- The boundary can be decomposed into segments.
  - – Useful to extract information from concave تقعر parts of the objects.
- A good way to achieve this is to calculate the convex Hull of the region enclosed by the boundary Hull
- Can be a bit noise sensitive
  1. Smooth prior to Convex hull calculation
  2. Calculate Convex Hull on polygon approximation

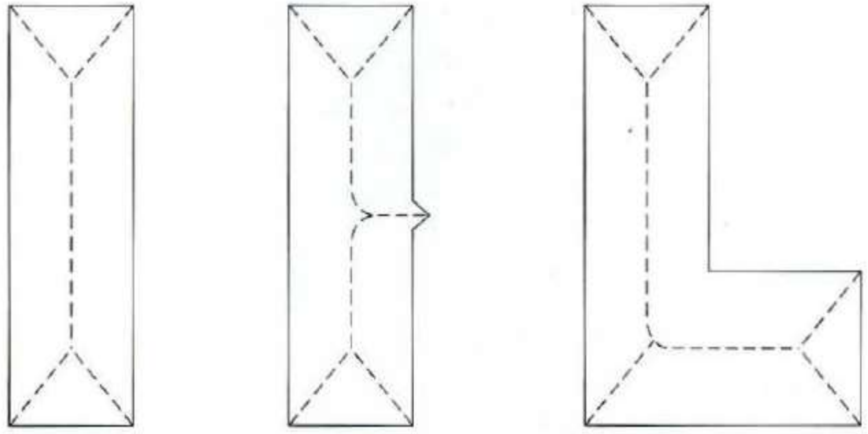


a b

FIGURE 11.6 (a) A region, S, and its convex deficiency (shaded). (b) Partitioned boundary.

Skeletons

- To reduce a plane region to a graph
  - by e.g. obtaining the skeleton of the region via thinning.
- Find medial axis transformation (MAT):
  - ✓ The MAT of region R with border B is found as:
    - For each point p in R, we find its closest neighbor in B.
    - If p has more than one such neighbor, it belongs to the Medial Axis.



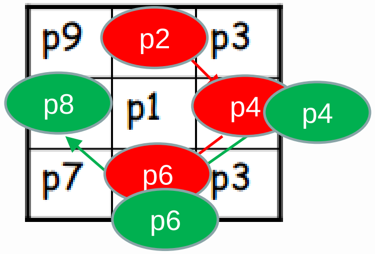
a b c

FIGURE 11.7 Medial axes (dashed) of three simple regions.

Thinning Algorithm

- Assume foreground pixels = “1” , Background = “0”
- **<u>1st pass</u>**
  - ❖ Flag a contour point p for deletion if the following conditions are satisfied:
    - (a) 2 <= N(p1) <= 6
    - (b) T(p1) = 1;
    - (c) p2 * p4 * p6 = 0
    - (d) p4 * p6 * p8 = 0

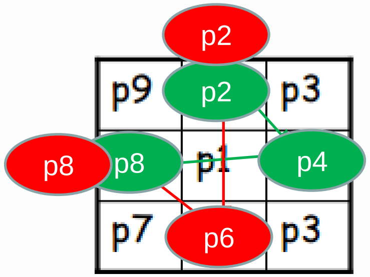
11.1.1



0	**0**	**1**
1	p1	0
1	0	1

  - ❖ N(p1) is the number of nonzero neighbors of p1
  - ❖ T(p1) is the number of 0-1 transitions in the ordered sequence of p2, p3, . . . , p8, p9,p2.
  - ❖ If (a) – (d) are not violated, the point is marked for deletion
    - Points are not deleted until the end of the pass
    - This way the data stays intact until the pass is complete

- **<u>2nd pass</u>**
- Conditions (a) and (b) are the same as the 1st pass
- (c) and (d) are different [call these (c′) and (d′)]:
  - (c′) p2 * p4 * p8 = 0
  - (d′) p2 * p6 * p8 = 0



- Delete all points that are flagged from the 2nd pass

✓ **Repeat 1st pass and 2nd pass until no contour point is deleted during an iteration**

Examples

(a) $N(p_1) = 1 : \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$: end point will be deleted!

NO

$N(p_1) = 7 : \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 0 & 1 \\ \hline \end{array}$: erosion will occur!

NO

(b) $T(p_1) = 2 : \begin{array}{|c|c|c|}\hline 0 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline 1 & 0 & 0 \\ \hline \end{array}$: connectivity will be broken!

Note that $N(p_1) = 2$ here

- *In the first case*
- N(p) = 5, S(p) = 1, $p2 \cdot p4 \cdot p6 = 0$, and $p4 \cdot p6 \cdot p8 = 0$, then p is flagged for deletion.
- *In the second case*
- N(p) = 1, so(2 :5 N(p,) <0 6) is violated and p is left unchanged.
- *In the third case*
- $p2 \cdot p4 \cdot p6 = 1$
  and $p4 \cdot p6 \cdot p8 = 1$, so conditions ($p2 \cdot p4 \cdot p6 = 0$) and ($p4 \cdot p6 \cdot p8 = 0$) violated and p is left unchanged.
- *In the forth case*
- S(p) = 2, so condition (T (p l) = 1) is violated and p is left unchange

THANK YOU