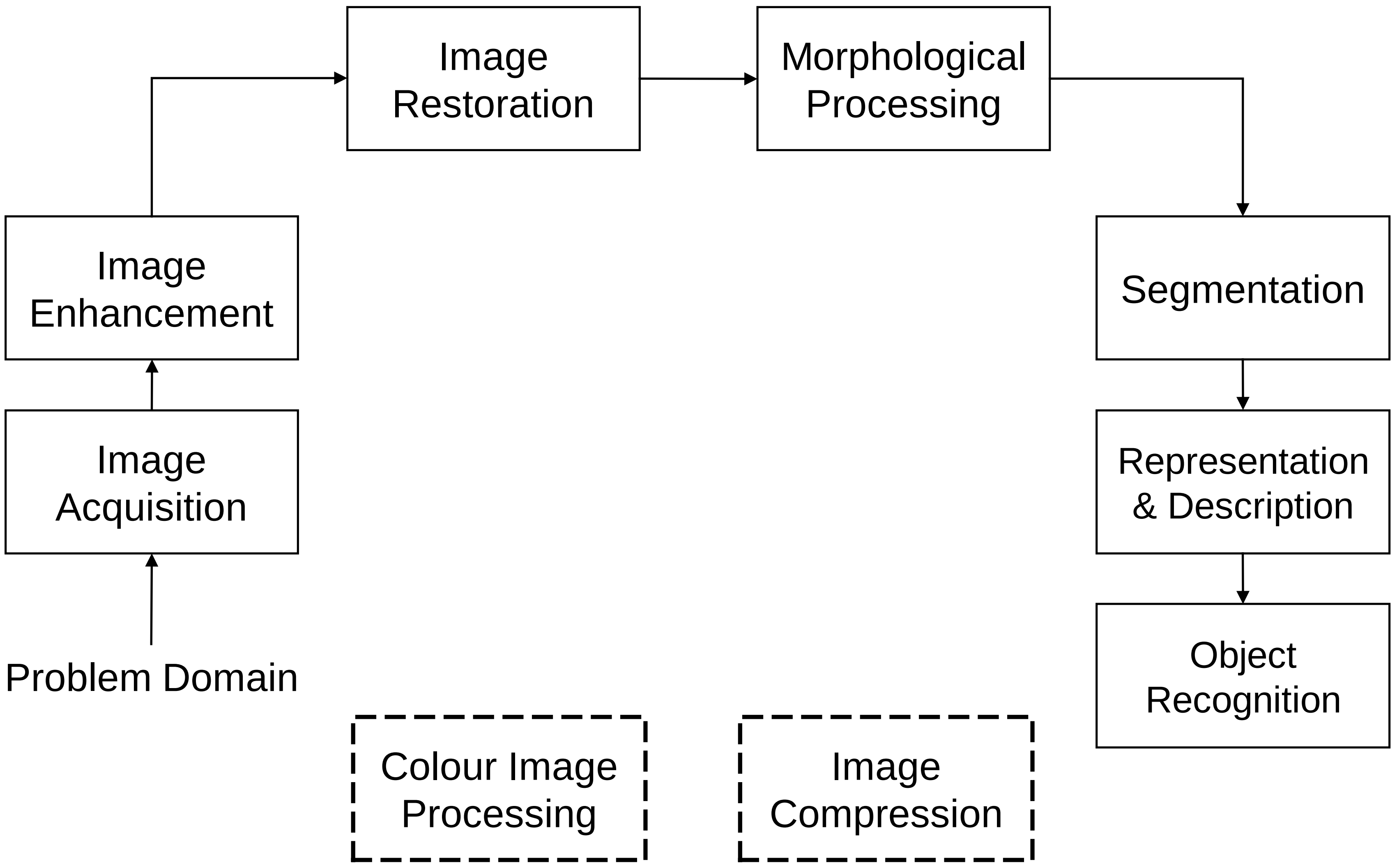
Image Representation and Description

Contents

Phases of Digital Image Processing



Phases of Digital Image Processing : Representation and Description

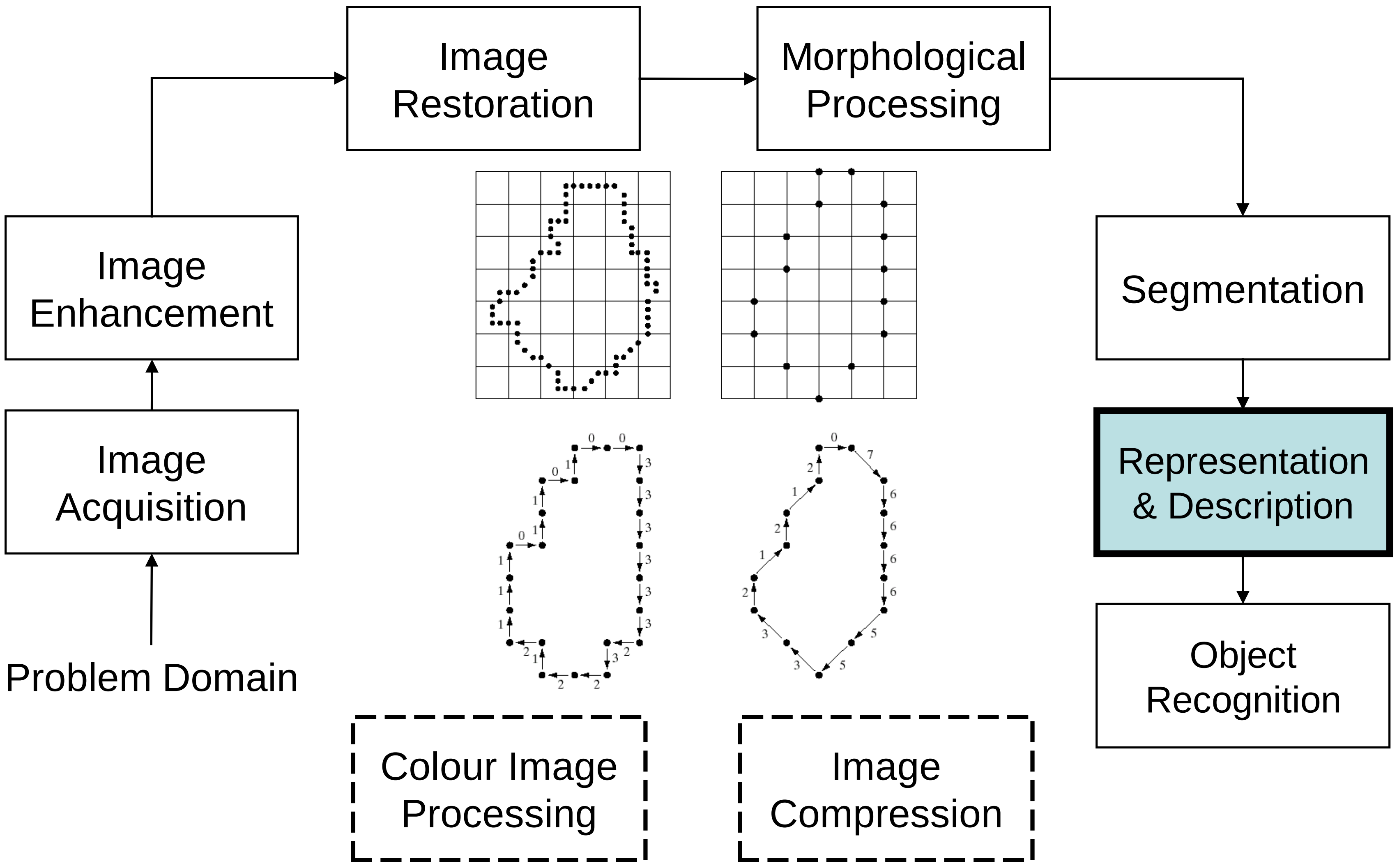


Image Representation

Representation means that we make the object information more accessible for computer interpretation.

Representation of a segmented region:

- In terms of its external characteristics (boundary).
- In terms of its internal characteristics (pixels comprising the region).

Representation Schemes

$$\begin{tikzcd}
\text{Segmentation} \arrow[r] & \text{Raw data (pixels)} \arrow[d] \\
& \text{Representations} \arrow[d] \\
& \text{Computation of descriptors}
\end{tikzcd}$$

Representation Schemes

Run Length Coding

Binary Tree/Quad Tree Coding

Chain Coding

Polygonal Approximations

Boundary Segments

The Skeleton/Thinning of a Region

…

Run Length Coding

Replace runs of equal brightness values by

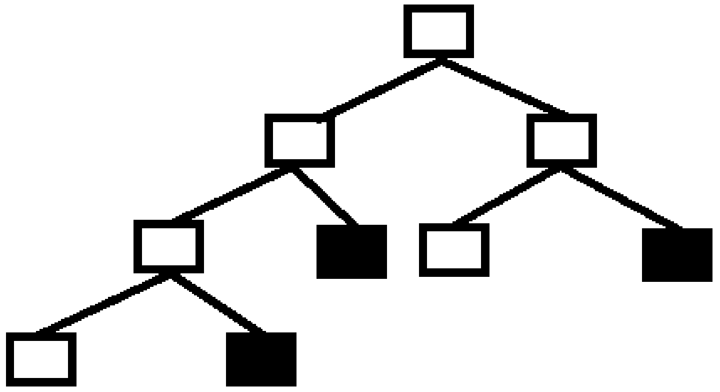
(length of run, value) [Wilkins and Wintz, 1970]

1 2 2 3 3 4 4 4 5 6 5

(1 1) (2 2) (2 3) (3 4) (1 5) (1 6) (1 5)

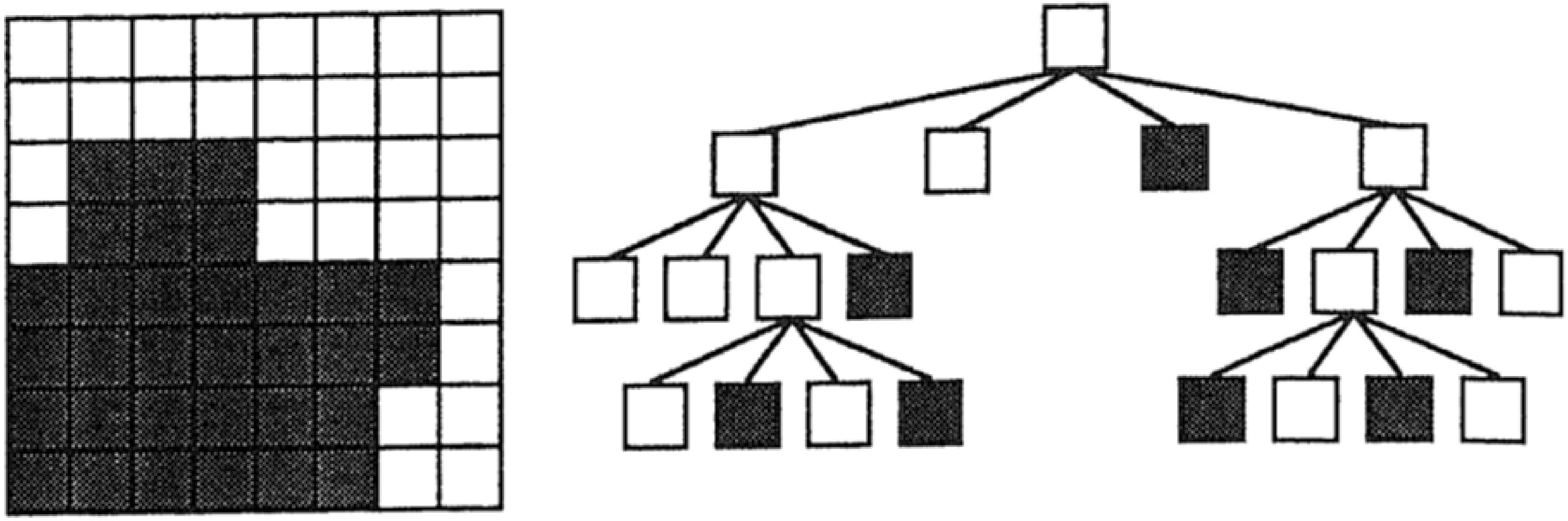
Binary Tree Coding

0011111100001111



Binary Tree Representation of a row of a Binary Image

Quad Tree Coding

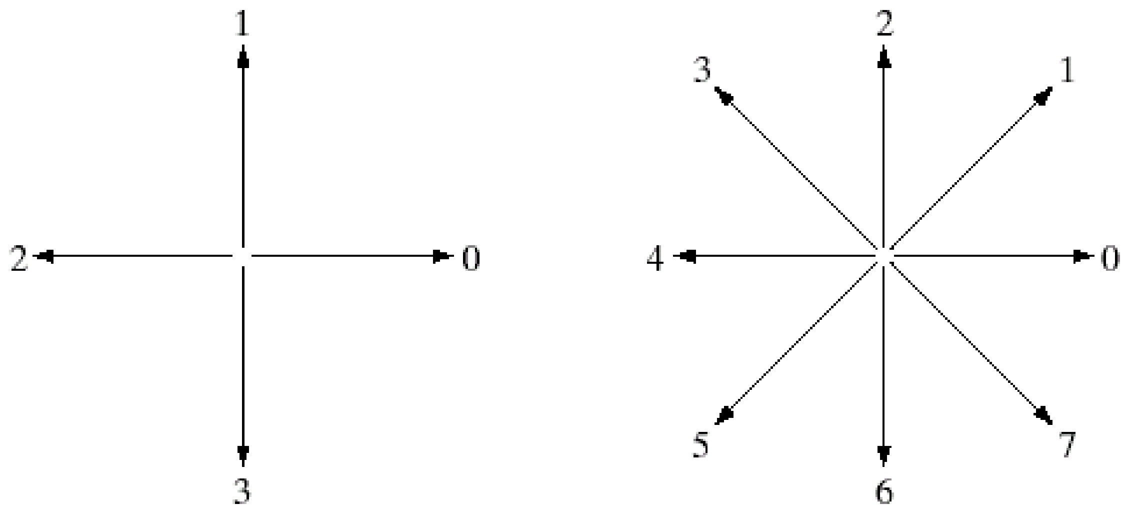

A quad-tree representation of an 8×8 binary image.

Chain Codes

Represent boundaries by a connected sequence of straight-line segments of specified length and direction [Freeman, 1961].

- Based on 4- or 8-connectivity.
- Direction of each segment coded by a numbering scheme.

Chain Codes



a b

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

Chain Codes

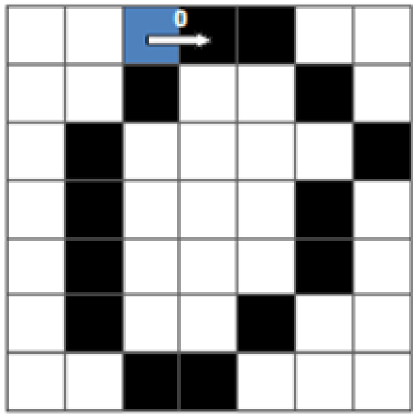
Steps for construction chain codes:

1. Select some starting point of the boundary and represent it by its absolute coordinates in the image.

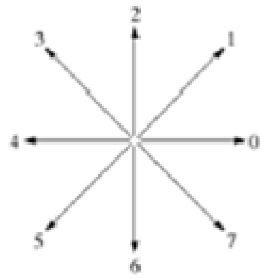
2. Represent every consecutive point by a chain code showing transition needed to go from current point to next point on the boundary.

3. Stop if the next point is the initial point or the end of the boundary.

Chain Codes

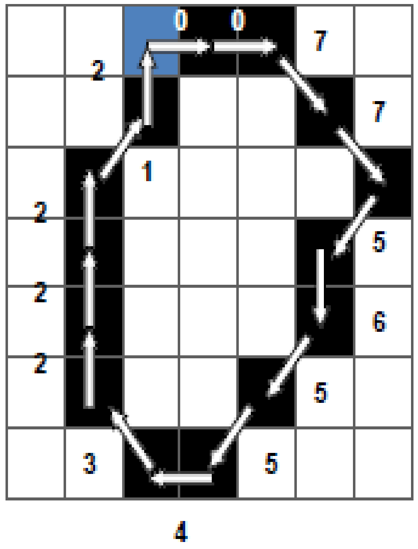


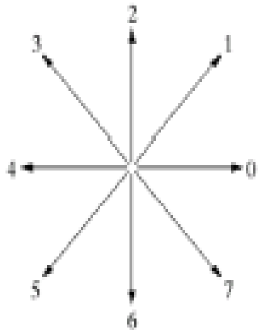
BOUNDARY



Chain Code:

0





Chain Code:

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

Chain Codes

Advantages:

- Preserves the information of interest
- Provides good compression of boundary description
- They are translation invariant

Chain Codes

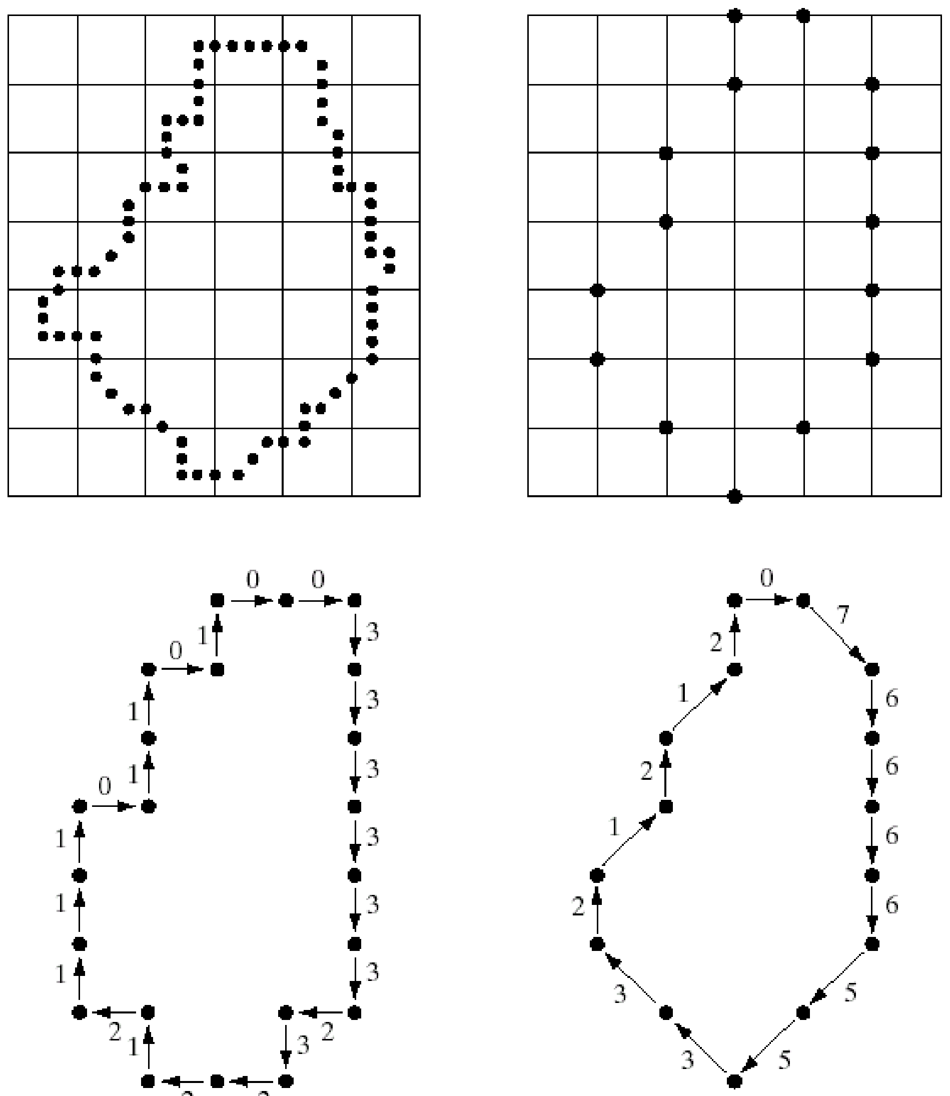
Problems:

- Long chains of codes.
- Easily disturbed by noise.

Solution:

- Resampling using larger grid spacing.
- Normalizations.

Chain Codes



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.

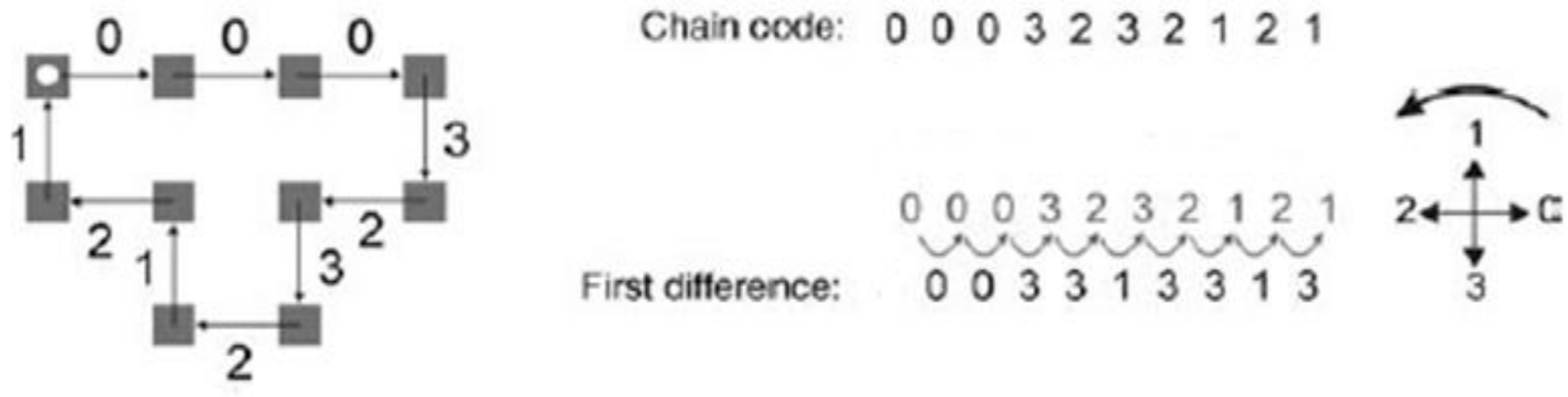
As the boundary is traversed, a boundary point as assigned to each node of the large grid, depending on the proximity of the original boundary of that node.

Chain Codes

Once the chain code for a boundary has been computed, it is possible to convert the resulting array into a <u>Rotation-Invariant</u> Equivalent, known as the first difference.

It is obtained by <u>counting the number of direction changes</u> (in a <u>counter-clockwise direction</u>) that separate two adjacent elements of the Freeman code.

Chain Codes



0
0
0
1
3
2
2
1
3
2
Chain code: 0 0 0 3 2 3 2 1 2 1
0 0 0 3 2 3 2 1 2 1
First difference: 0 0 3 3 1 3 3 1 3
1
2
0
3

Chain Codes

Problem:

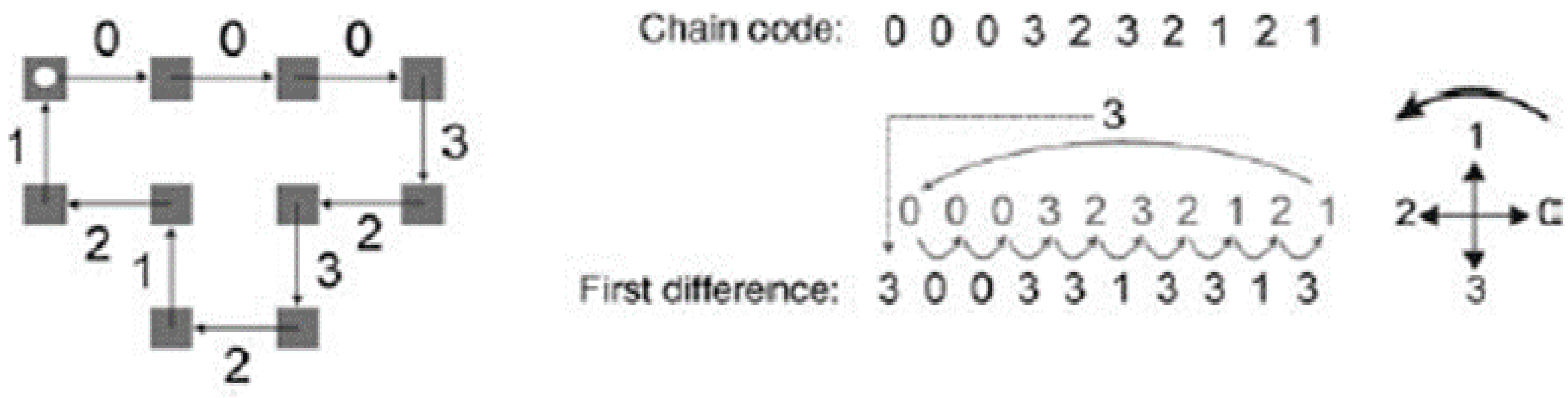
A chain code sequence depends on a starting point.

Solution:

- Treat a chain code as a circular sequence of direction numbers and redefine the starting point so that the resulting sequence of numbers forms an integer of minimum magnitude.

- Perform a circular shift until you get a configuration that would form the smallest integer if you interpreted the numbers in the chain as digits in a integer.

Chain Codes

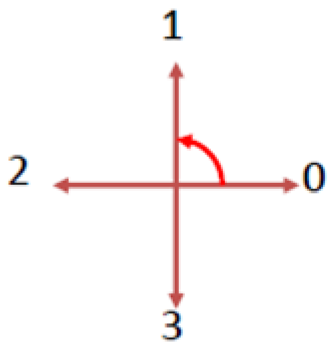

0
0
0
1
3
2
2
1
3
2
Chain code: 0 0 0 3 2 3 2 1 2 1
3
0 0 0 3 2 3 2 1 2 1
First difference: 3 0 0 3 3 1 3 3 1 3
1
2
0
3

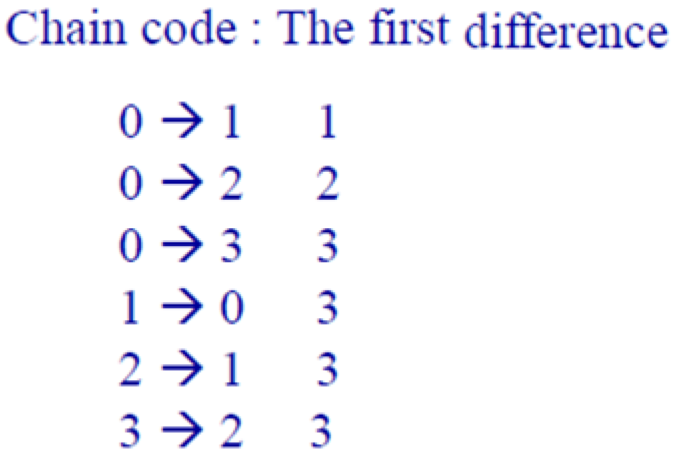
Chain Codes

A chain code: 10103322

The first difference: 3133030

Treating a chain code as a circular sequence, we get the first difference: 33133030




Chain code : The first difference

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

Polygonal Approximations

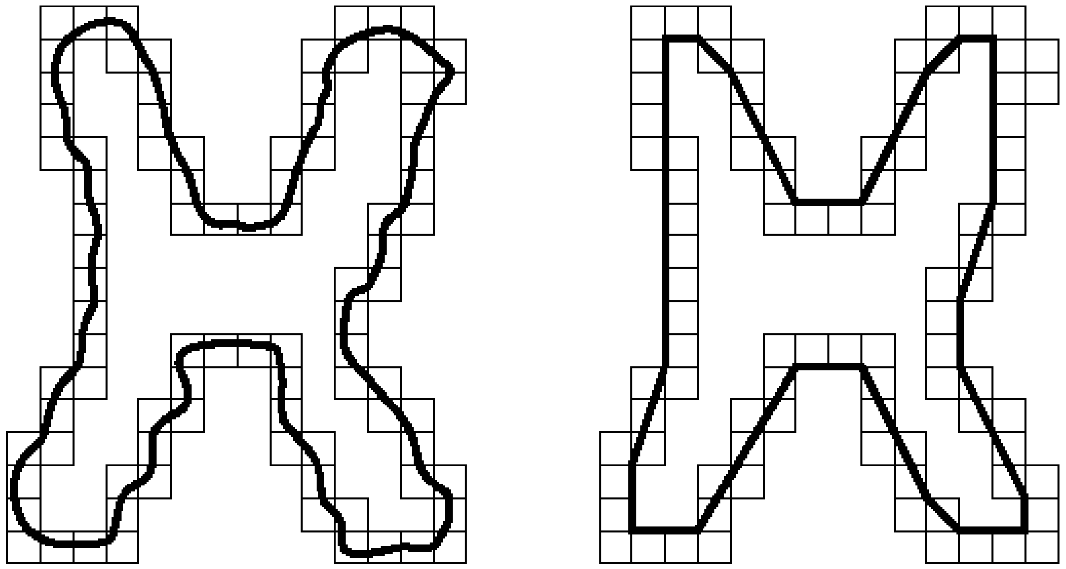
To capture the “essence” of the boundary shape with the fewest possible polygonal segments.

Various methods:

- Minimum perimeter polygons
- Merging techniques (least square error line fit)
- Splitting techniques

Polygonal Approximations

Minimum perimeter polygons



a b

FIGURE 11.3 (a) Object boundary enclosed by cells. (b) Minimum perimeter polygon.

- Enclose a boundary by a set of concatenated cells.
- Visualize this enclosure as two walls corresponding to the outside and inside boundaries of the strip of cells.
- Object boundary as a rubber band contained within the wall.
- If the rubber band is allowed to shrink, it produces a polygon of minimum perimeter that fits the geometry established by the cell strip.

Polygonal Approximations

Merging techniques

- Merge points along a boundary until the least square error line fit of the points merged so far exceeds a preset threshold.
- When this condition occurs, the parameters of the line are stored, the error is set to 0, and the procedure is repeated, merging new points along the boundary until the error again exceeds the threshold.
- At the end, the intersections of adjacent line segments form the vertices of the polygon.

Polygonal Approximations

Splitting techniques

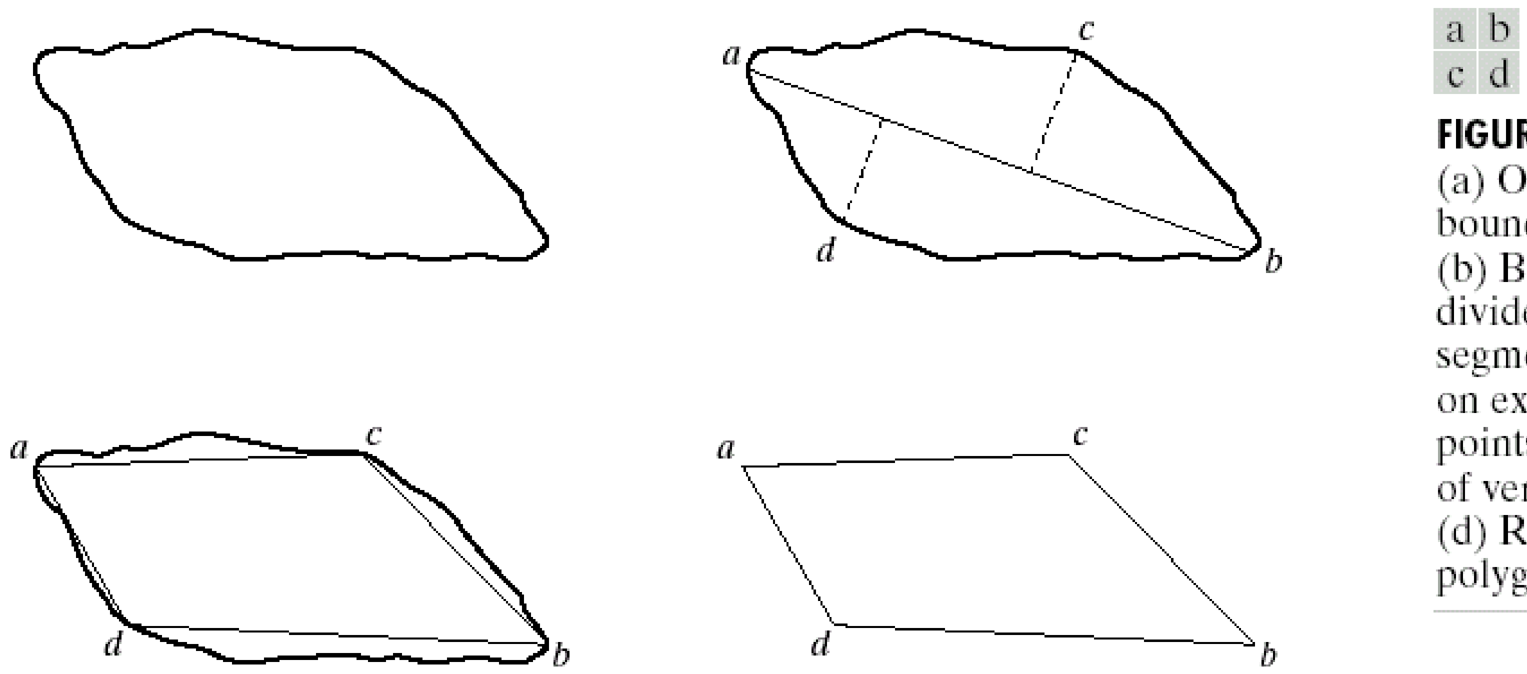


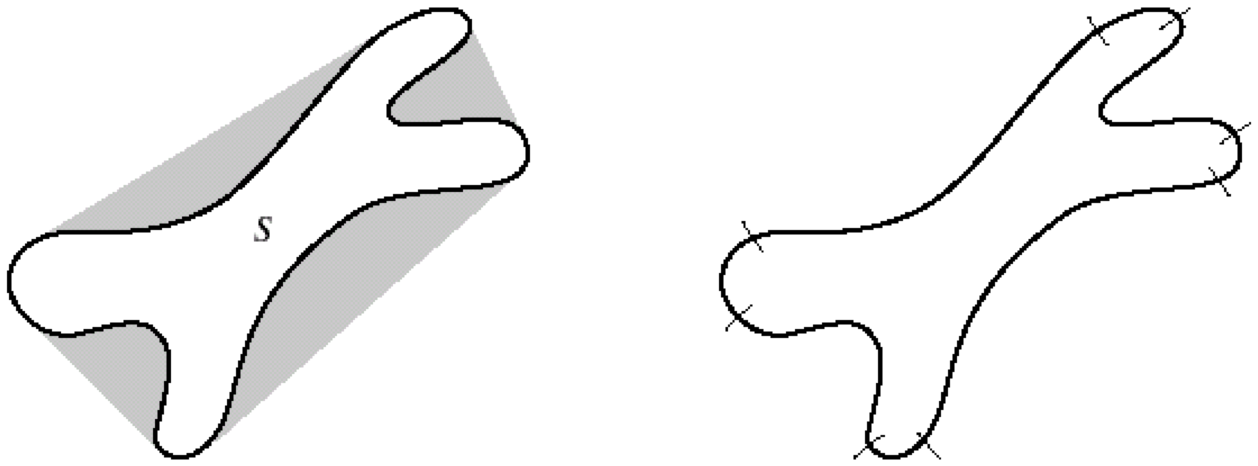
FIGURE 11.4
(a) Original boundary.
(b) Boundary divided into segments based on extreme points. (c) Joining of vertices.
(d) Resulting polygon.

Boundary Segments

Decomposition of a boundary into segments reduces the boundary's complexity and simplifies the description process.

- Convex hull H of a set S is the smallest convex containing S.
- H-S = convex deficiency D of S.
- Scheme: independent of size and orientation.

Boundary Segments



a b

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

The region boundary can be partitioned by following the contour of S and marking the points at which a transition is made into or out of a component of the convex deficiency.

Boundary Segments

Prior to partitioning, smooth the boundary:

- e.g. by replacing each pixel by the average coordinates of its neighbors along the boundary.
- or use a polygonal approximation prior to finding the convex deficiency.

The Skeleton of a Region

To reduce a plane region to a graph

- by e.g. obtaining the skeleton of the region via thinning (also called *skeletonization*) algorithm.
- Thinning procedures play a central role in a broad range of problems in image processing.

The Skeleton of a Region

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* (skeleton) of R.

The Skeleton of a Region

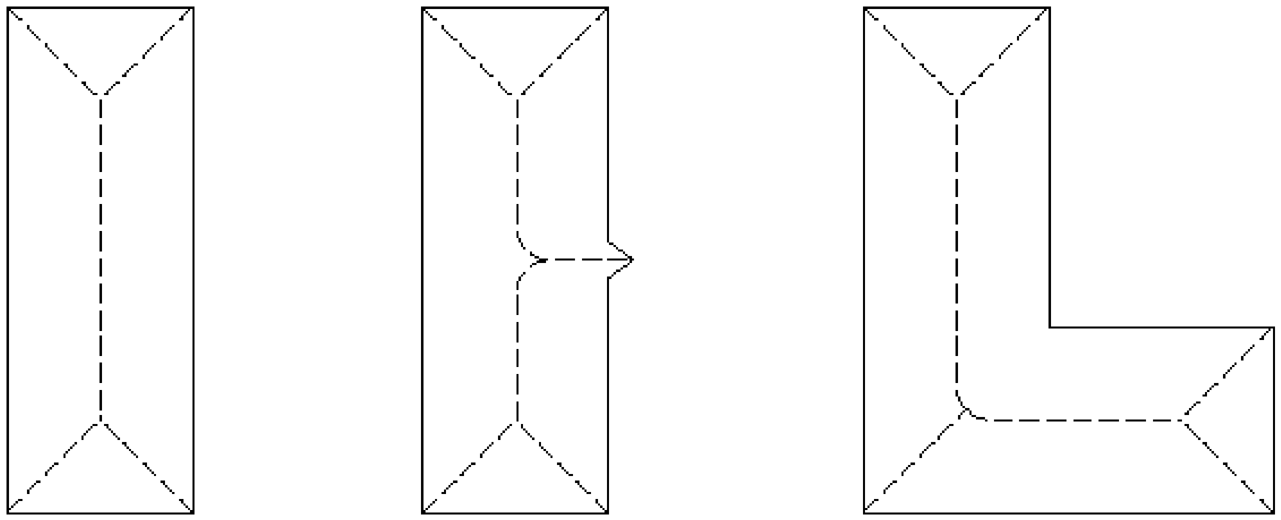

a b c

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

The Skeleton of a Region

To improve computational efficiency, in essence we perform thinning:

Edge points of a region are iteratively deleted if:

- End points are not deleted.
- Connectedness is not broken.
- No excessive erosion is caused.

The Skeleton of a Region

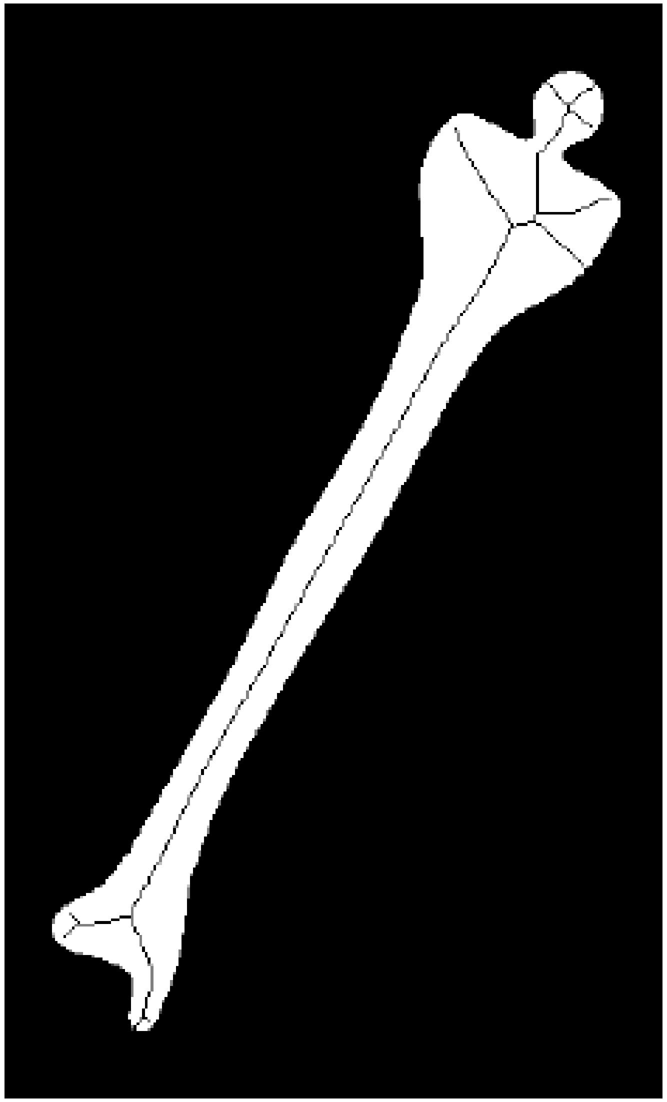


FIGURE 11.10 Human leg bone and skeleton of the region shown superimposed.

Image Description

Description of the region based on the chosen representation:

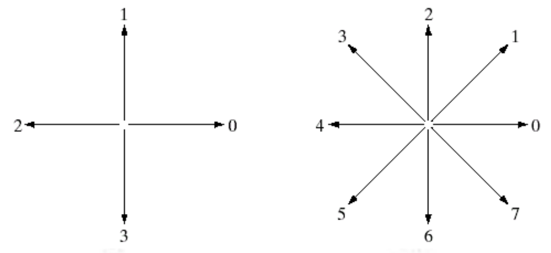
- e.g., boundary: length, orientation of straight line, joining the extreme points, number of concavities.

Descriptors should be insensitive to changes in size, translation, rotation, …

Boundary Descriptors

Length / Perimeter:

For 4-connected objects:



a b

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

- Number of pixels
- Number of vertical and horizontal components + $(\sqrt{2}-2)\times n_d$, where n_d is the number of occurrences of code pair **(1, 0), (2, 1), (3, 2), and (0, 3).**

Boundary Descriptors

Diameter:

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

- D: distance measure; p_i,p_j: boundary points
- Major axis (connecting the two extreme points)

Boundary Descriptors

Curvature:

- Rate of change of slope
- i.e. using the difference between the slopes of adjacent boundary segments, which have been represented as straight lines, as a descriptor of curvature at the point of intersection of the segments.

Boundary Descriptors

Curvature (cont.):

Convex segment: change in slope at p is nonnegative.

Concave segment: change in slope at p is negative.

Ranges in the change of slope:

- Less than 10° → line
- More than 90° → corner

Regional Descriptors

Area: Number of pixels within the boundary

Perimeter: length of boundary

- Can be used with area to measure compactness (perimeter^2/area)

- Compactness is:

  Dimensionless, and thus insensitive to scale changes.

  Insensitive to orientation.

Regional Descriptors

Other descriptors:

- Mean and Median of gray levels
- Minimum and Maximum gray-level values
- Number of pixels with values above and below the mean

Topological Descriptors

Topology:

Topological properties of a set of objects involve adjacency and connectivity.

Properties of figures that are unaffected by deformations.

Looks like rubber sheet distortions where shape and size of the objects may change but not their connectivity or count.

Topological Descriptors

<u>Examples:</u>

Number of holes (H)

Number of connected components (C)

A subset of maximal size such that any two points can be joined by a connected curve lying entirely within the subset.

Euler number E: E = C-H

- Also a topological property.

Topological Descriptors

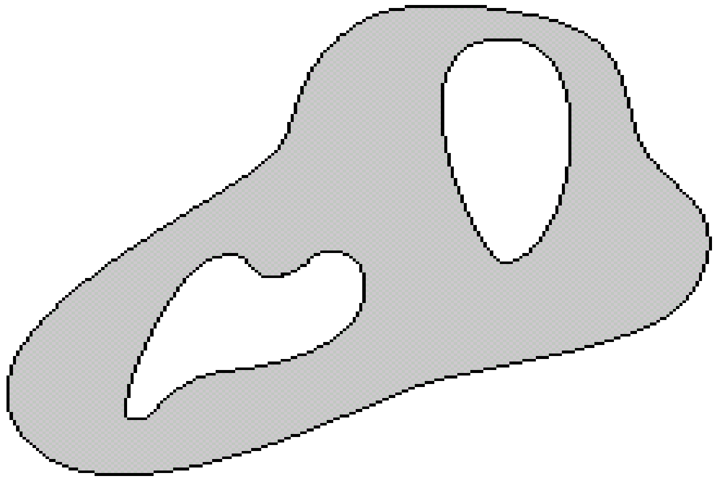


FIGURE 11.17 A region with two holes.

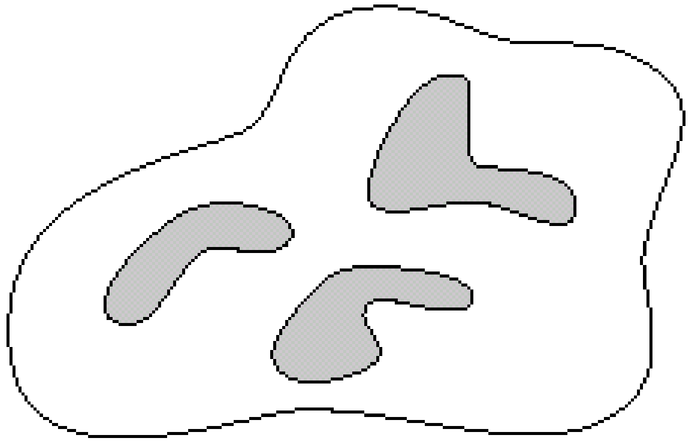


FIGURE 11.18 A region with three connected components.

Topological Descriptors

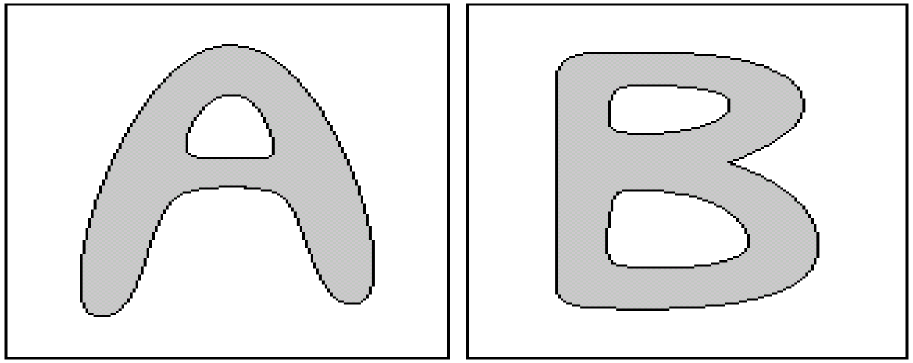
Regions represented by straight line segments (polygonal network) have a particular simple interpretation in terms of the Euler number.

$$\left.\begin{array}{l}\text{- V: Number of vertices}\\ \text{- Q: Number of edges}\\ \text{- F: Number of faces}\end{array}\right\} \quad \text{Euler formula:}$$

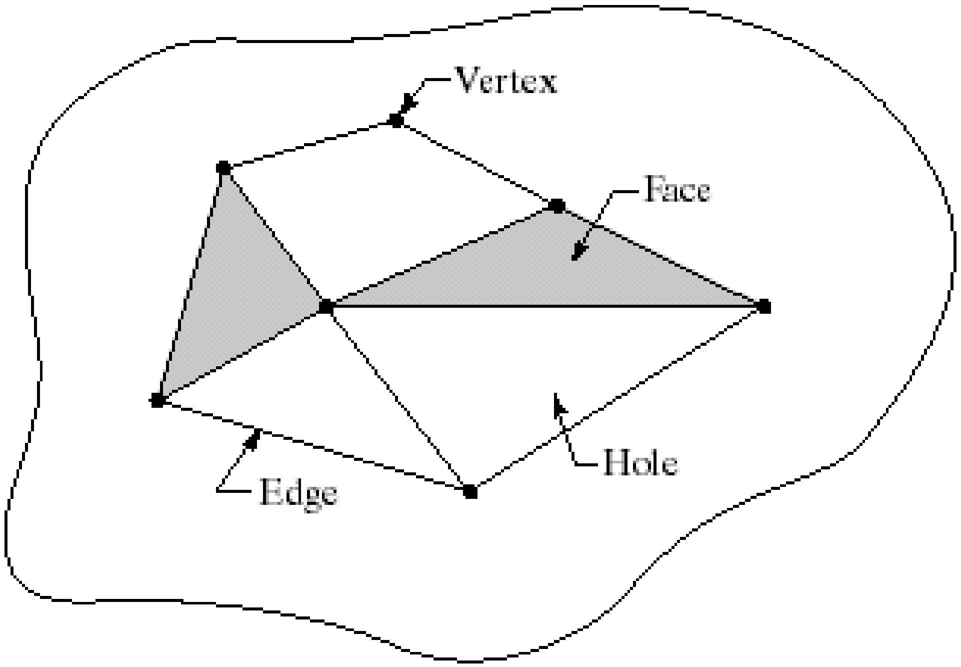
$$V\text{-}Q+F = C\text{-}H = E$$

Topological Descriptors

E=1-1=0



E=1-2=-1



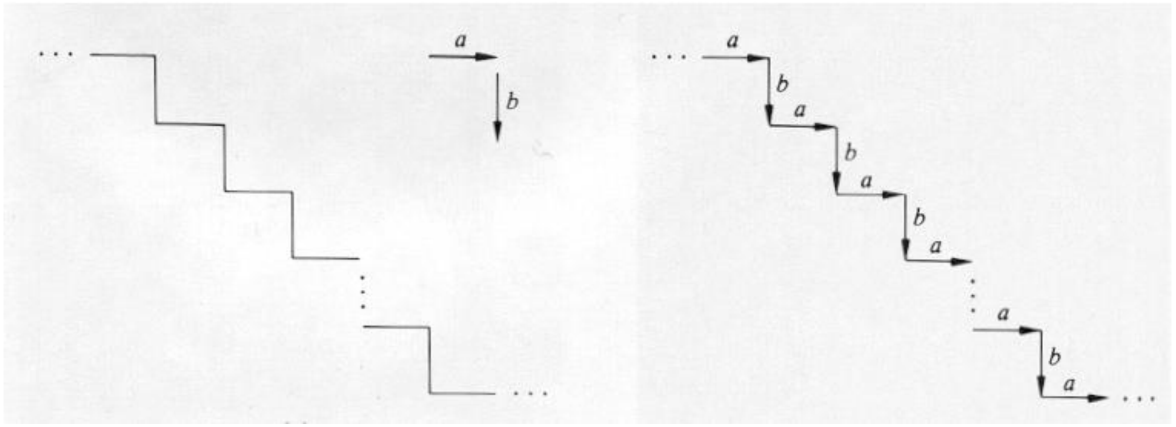
V= 7 (vertices)
Q = 11 (edges)
F = 2 (faces)
C = 1
H = 3
E=7-11+2=1-3=-2

FIGURE 11.20 A region containing a polygonal network.

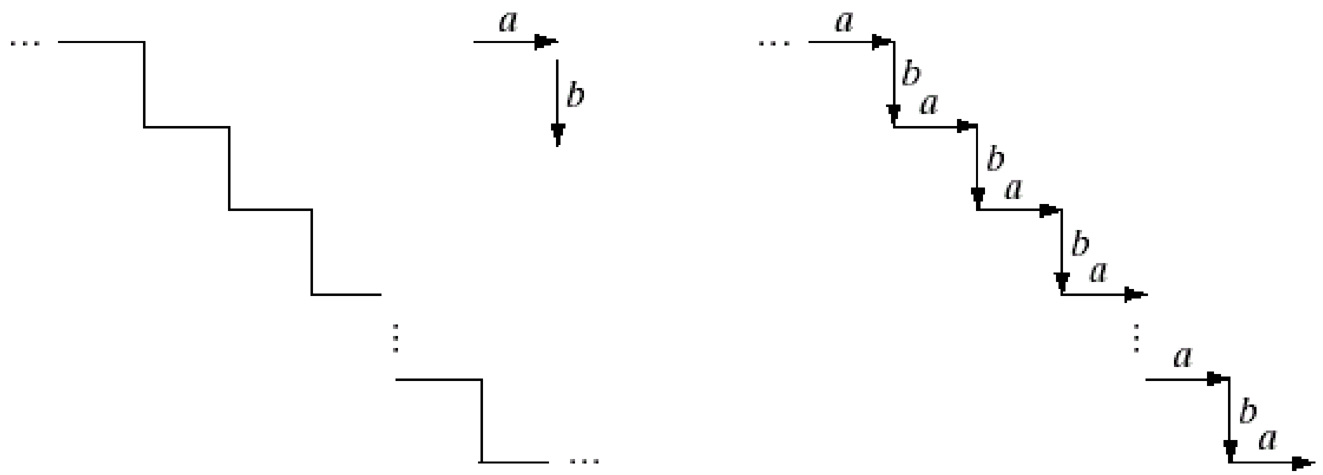
Relational Descriptors

To organize boundaries and regions to exploit any structural relationships that may exist between them.

Example:



Relational Descriptors



a b

FIGURE 11.30
(a) A simple staircase structure.
(b) Coded structure.

Relational Descriptors

In the previous example:

- Recursive relationship involving the primitive elements ‘a’ and ‘b’.

- Rewriting rules:

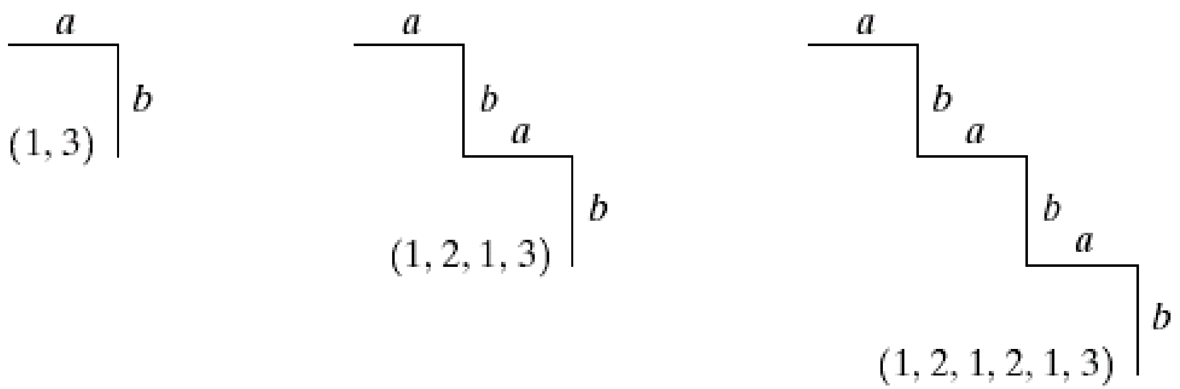
  $S \rightarrow aA$

  $A \rightarrow bS$, and

  $A \rightarrow b$

Relational Descriptors

FIGURE 11.31 Sample derivations for the rules $S \rightarrow aA$, $A \rightarrow bS$, and $A \rightarrow b$.



Relational Descriptors

When dealing with disjoint structures, tree descriptors are used:

A tree T is a finite set of one or more nodes for which:

- There is a unique node $ designated the root.
- The remaining nodes are partitioned into m disjointed sets T_1, …, T_m, each of which in turn is a tree called a subtree of T.

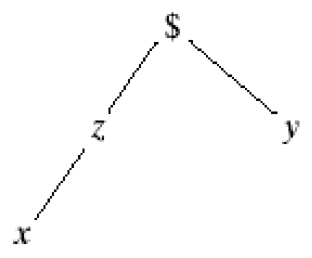
Relational Descriptors

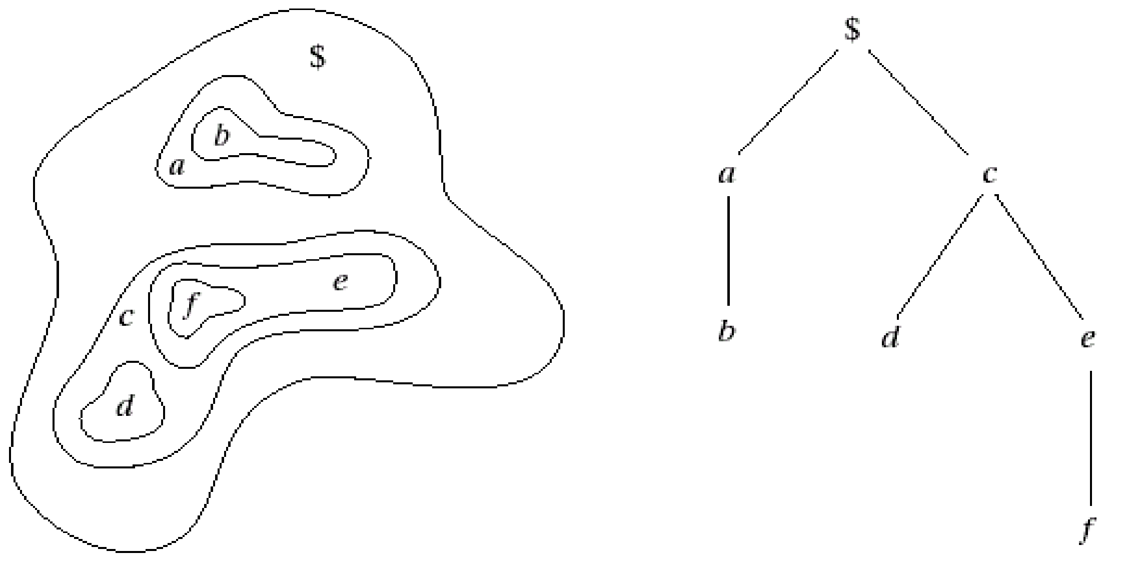
The tree frontier is the set of nodes at the bottom of the tree (leaves), taken in order from left to right.

Two types of information are important (in a tree):

- Information about a node stored as a set of words describing the node.
- Information relating a node to its neighbors stored as a set of pointers to those neighbors.

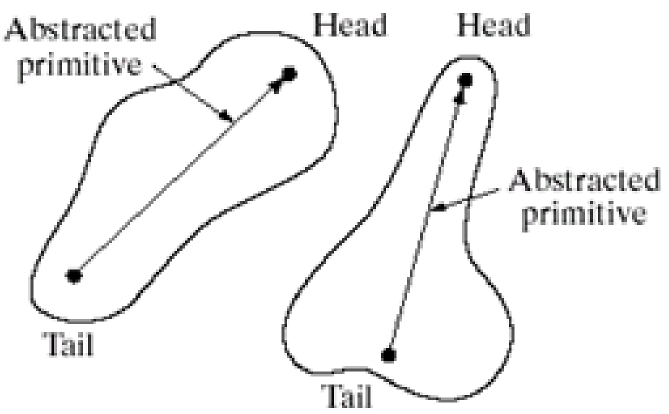
Relational Descriptors

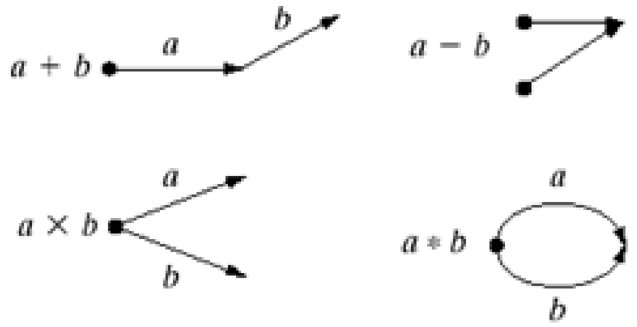


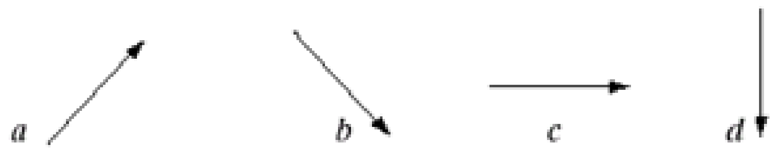


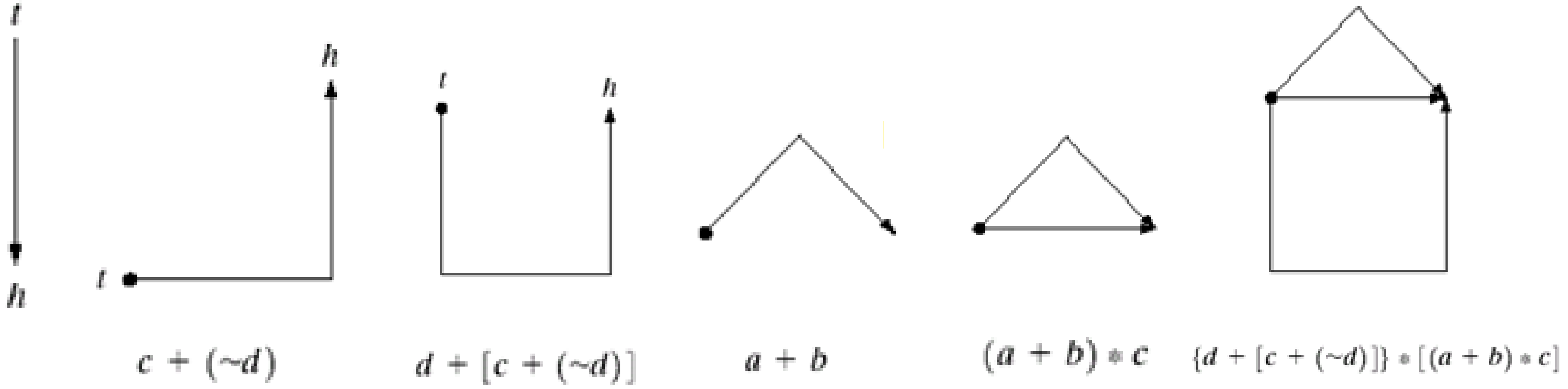
a b

FIGURE 11.35 (a) A simple composite region. (b) Tree representation obtained by using the relationship "inside of."


Abstracted primitive
Head
Head
Abstracted primitive
Tail
Tail


a + b
a
b
a − b
a × b
a
b
a * b
a
b


a
b
c
d


t
h
t
h
c + (~d)
t
h
d + [c + (~d)]
a + b
(a + b) * c
{d + [c + (~d)]} * [(a + b) * c]

Morphology

Morphology deals with form and structure.

Mathematical morphology is a tool for extracting image components useful in:

- Representation and description of region shape.
- Preprocessing (filtering, thinning, etc.).

Basic Morphological Algorithms

Purpose:

- To extract image components that are useful in the representation and description of shape.

Boundary Extraction:

$$\beta(A) = A - (A \ominus B)$$

Basic Morphological Algorithms

Extraction of Connected Components:

$$X_k = (X_{k-1} \oplus B) \cap A \qquad \text{k=1,2,3,...}$$

Where X_0=p →
when X_k=X_{k-1} the algorithm has converged.

Topological Descriptors

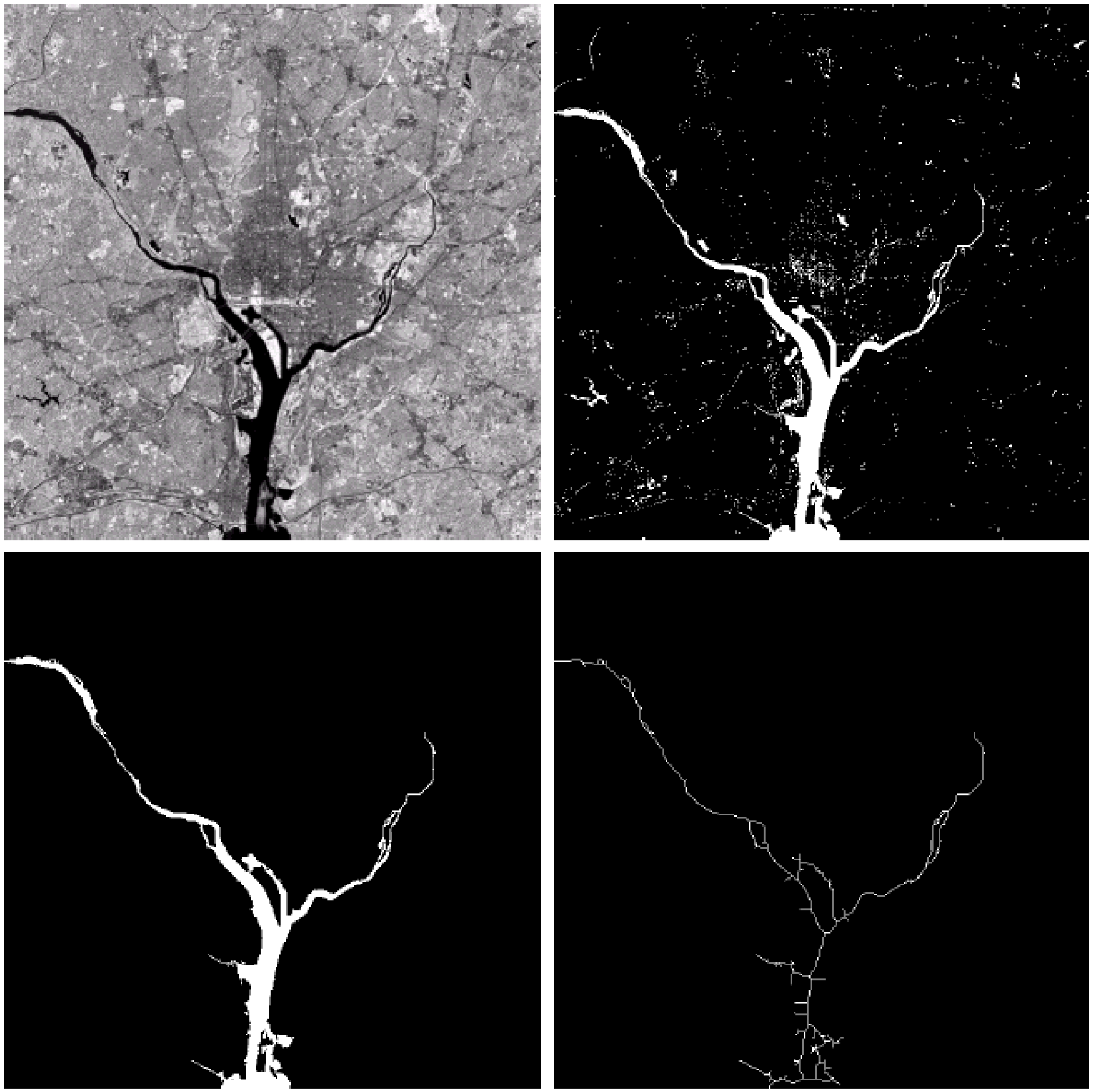


FIGURE 11.21 (a) Infrared image of the Washington, D.C. area. (b) Thresholded image. (c) The largest connected component of (b). Skeleton of (c).