Region and Shape Representation and Description

Presented by
Dr. Subarna Chatterjee
subarna.cs.et@msruas.ac.in

CONTENTS

- After performing image segmentation, a region may be represented in terms of
  - external characteristics (boundaries).
  - internal characteristics (texture).

- A shape may be considered as a filled region with a unique value,
  - e.g. f (x,y)=1

Boundary Processing

- The segmentation techniques discussed earlier, yield raw data in the form of pixels along a boundary or pixels contained in a region.

- It is standard practice to use schemes that compact the segmented data into representations that facilitate the computation of descriptors.

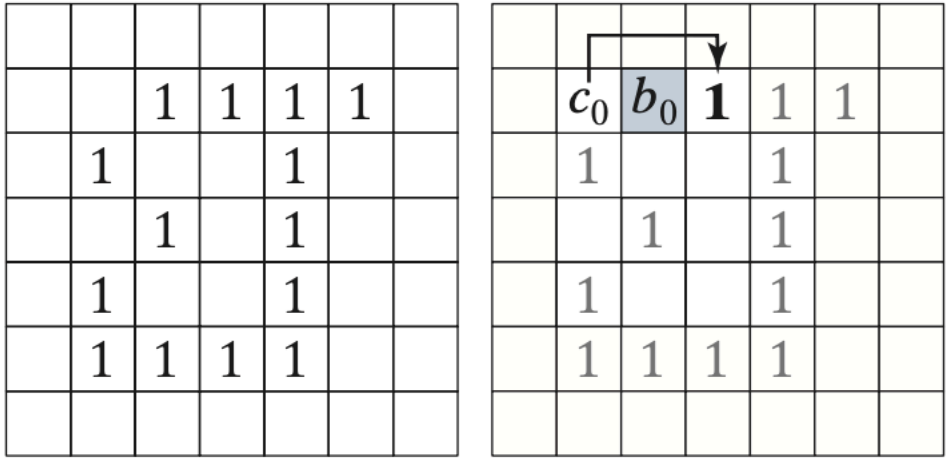
- In this section, we discuss various boundary pre-processing approaches suitable for this purpose.

Boundary Following (Tracing)

- The points in the boundary of a region be ordered in a clockwise or counter-clockwise direction.
- Consequently, we will discuss boundary-following algorithm whose output is an *ordered* sequence of points.
- We assume
  1. Binary Images (object and background points are labelled 1 and 0);
  2. Images are padded with a border of 0's to eliminate the possibility of an object merging with the image borders.
  3. We limit the discussion to single regions. This extends to multiple, disjoint regions by processing the regions individually.

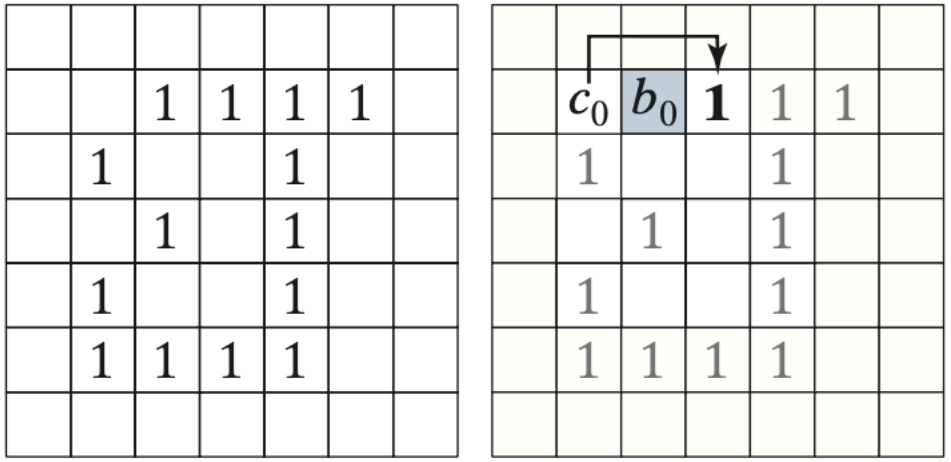
Cont..

- Given a binary region *R* or its boundary, the algorithm for following the border of *R*:
  1. Let the starting point b_0 , be the uppermost, leftmost point in the image labelled 1.
  2. Denote by c_0 the west neighbour of b_0 . c_0 is always a background point.



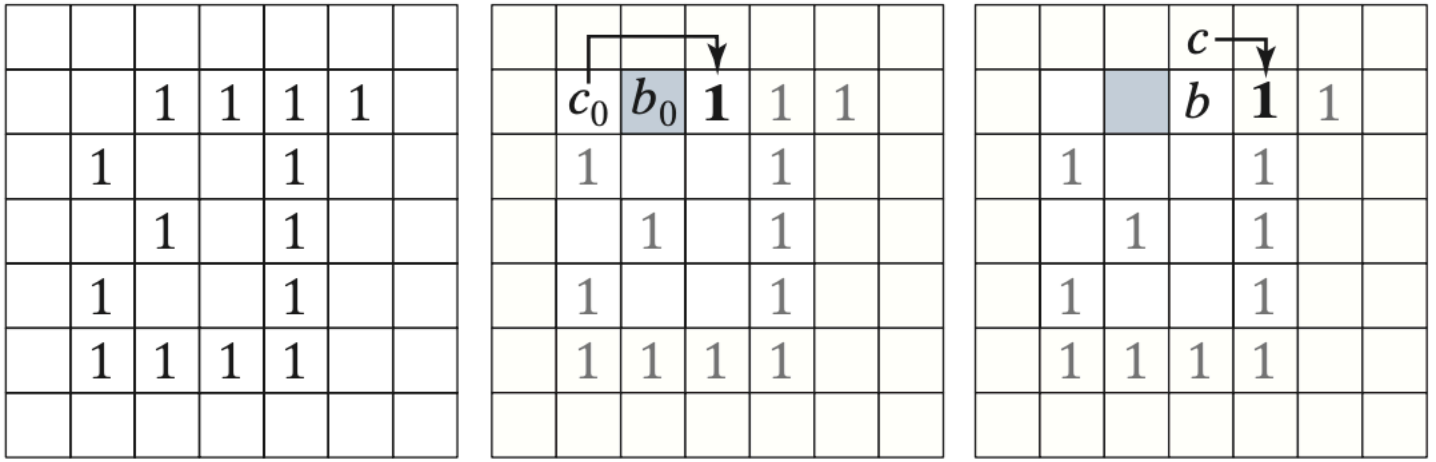
Cont..

3. Examine the 8-neighbors of b_0 , starting at c_0 and proceeding in a clockwise direction.
4. Let b_1 denote the first neighbour encountered whose value is 1.
5. Let c_1 denote the background point immediately preceding b_1 in the sequence.



Cont..

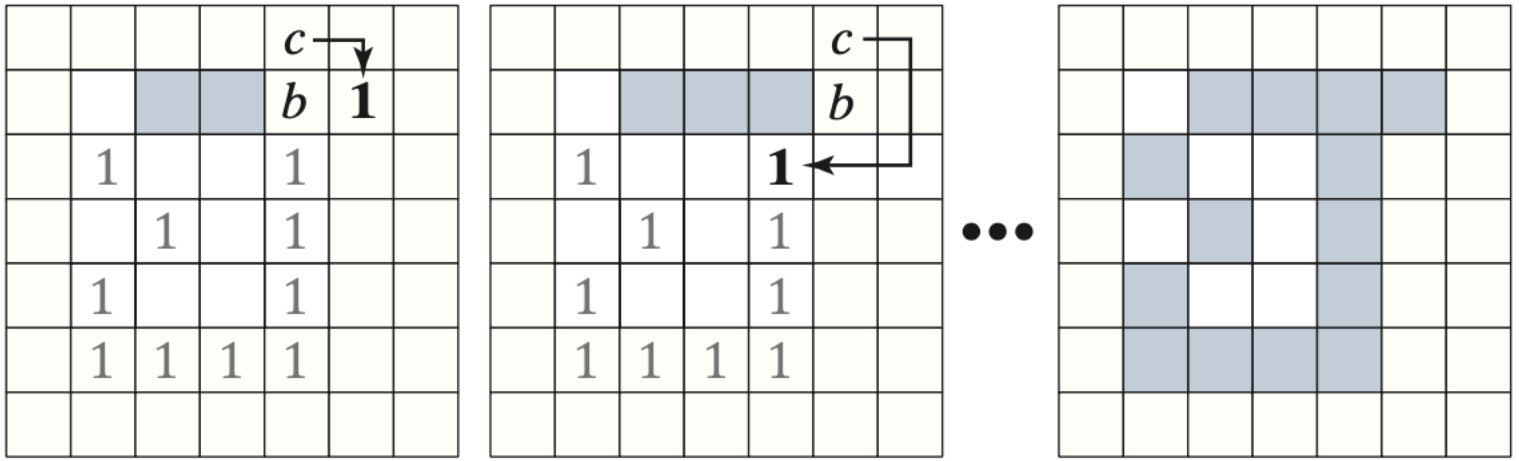
6. Store the locations of b_0 and b_1 for use in Step 10.
7. Let $b=b_1$ and $c=c_1$.
8. Let the 8-neighbors of b, starting at c and proceeding in a clockwise direction, be denoted by n_1 , n_2 ,…, n_8. Find the first n_k labeled 1.



Cont..

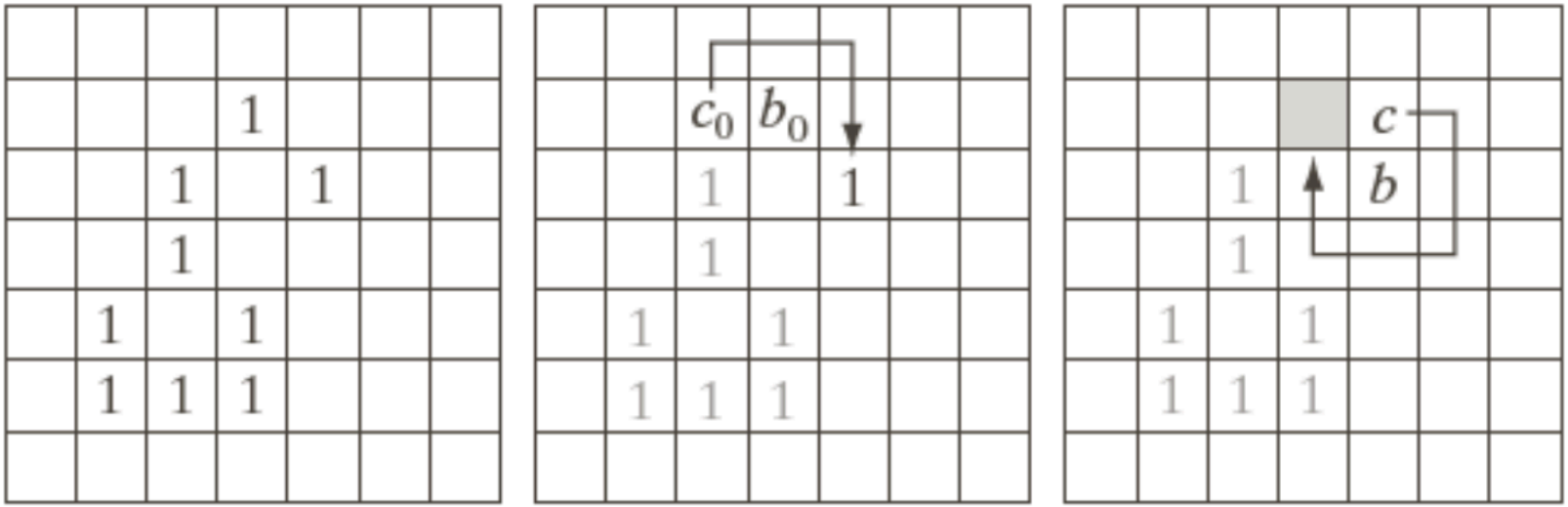
9. Let $b=n_k$ and $c=n_{k-1}$.
10. Repeat steps 8 and 9 until $b=b_0$, that is, we have reached the first point and the next boundary point found is b_1 .

- The algorithm is due to G. Moore [1968]

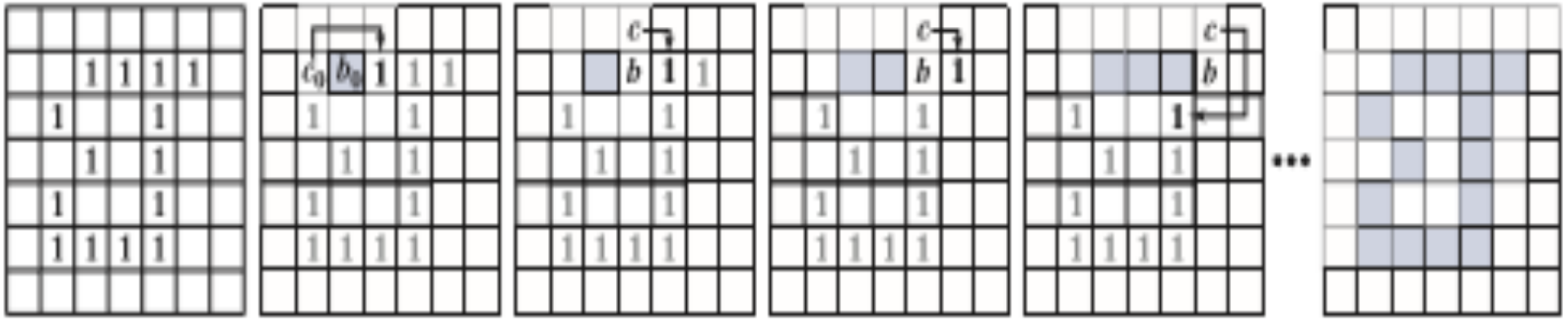


Cont..

- The need for the stopping rule "… and the next boundary point found is b_1 " is shown below.
- We would only include the spur at the right if we stop when we reach the initial point without checking the next point.



Cont..



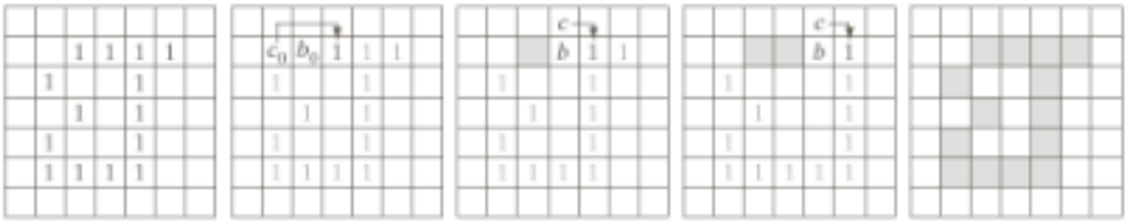
a b c d e f

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.

Cont..

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

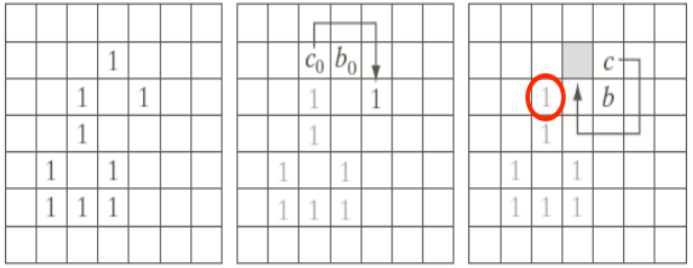
FIGURE 11.1 Illustration of the first few steps in the boundary-following algorithm. The point to be processed next is labeled in black, the points yet to be processed are gray, and the points found by the algorithm are labeled as gray squares.

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 $n_1, n_2, \ldots, n_8$. 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 $b=b_0$ and the next boundary point found is b_1.

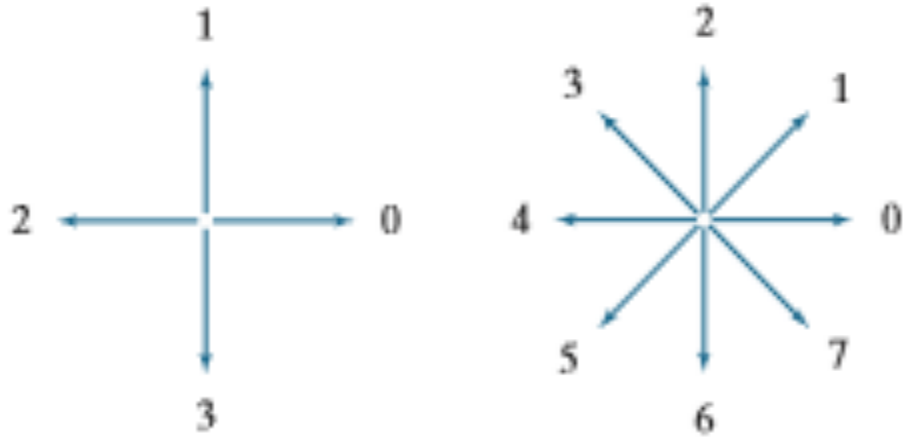
This prevents errors encountered in spurs. Simply finding b_0 again is not enough. For example, in this figure the algorithm would start at b_0, find b, and then come back to b_0 stopping prematurely with only $b=b_0$. However, the next boundary point found should be the circled point which is NOT b_1.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

Boundary Representation: Chain code

- We describe the shape of the boundary of an object in an image, known as *chain code* although name is *Freeman chain code*.
- The idea is to traverse the boundary of the object, and for every new pixel, transcript the direction we travelled to reach this object. i.e. describe the sequence of steps generated when walking around the boundary
- The direction is coded by a numbering scheme (4 or 8-connectivity)

a b

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

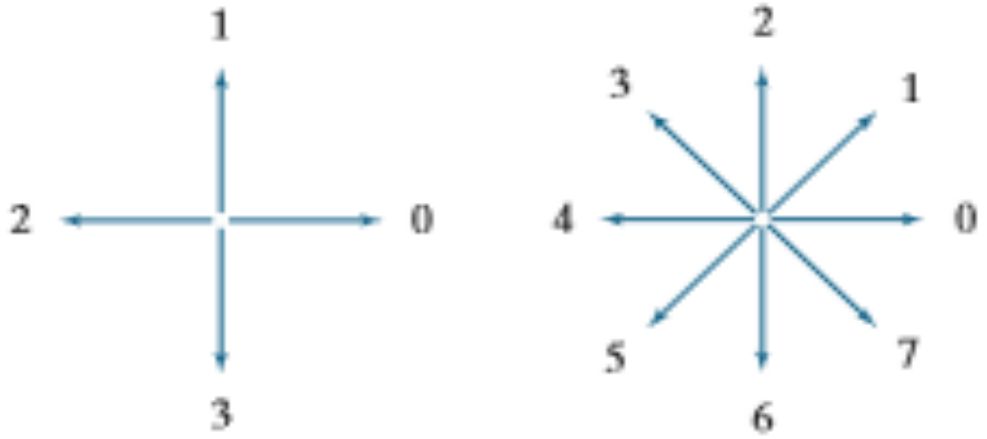


Chain code

- Freeman codes [1961] represent a boundary by the sequence of straight line segments of specified length and direction.
- It is common to denote the code relative to either a N_4 or a N_8, and we have two major classes: *absolute chain code* and *relative chain code*. In both cases, the code consists of directions (referring to the enumerated directions in Figure that you need to follow in order to traverse the boundary

a b

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

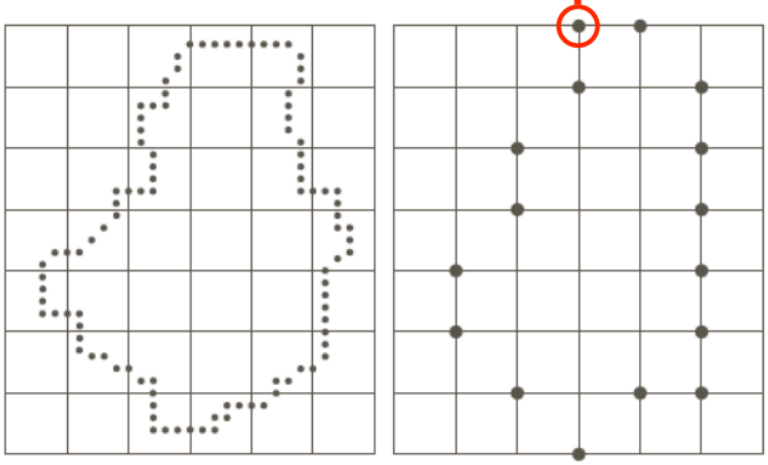


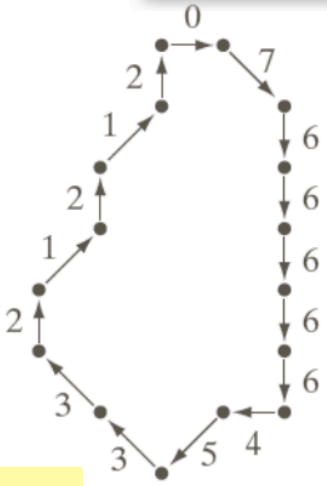
Cont..

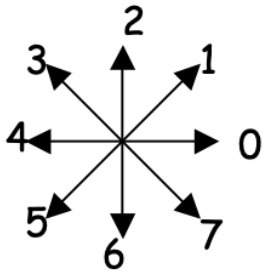
Arbitrarily start chain code here

8-direction chain coded boundary

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







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

a b c

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

8-connected code: 076666645332121 2

Cont..

- Chain codes can be based on either 4-connectedness or 8-connectedness.
- The first difference of the chain code:
  - This difference is obtained by counting the number of direction changes (in a counterclockwise direction)
  - For example, the 1st difference of the 4-direction chain code 10103322 is 3133030.
- Assuming the first difference code represent a closed path, rotation normalization can be achieved by circularly shifting the number of the code so that the list of numbers forms the smallest possible integer.
- Size normalization can be achieved by adjusting the size of the resampling grid.

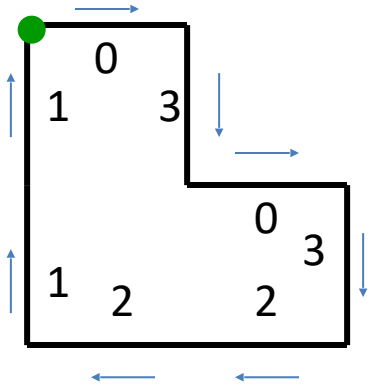
Cont..

- To avoid noise degradation and long chains a resampling of the image grid is commonly used to describe the boundary at a coarser level.
- The chain code depends on the starting point.
- To normalize it, we treat the code as a circular sequence of direction numbers and redefine the starting point so that the resulting sequence forms an integer of minimum magnitude.
- To account for rotation, we use the first differences of the chain code instead of the code itself.
- The 1st difference is obtained by counting the number of direction changes that separate two adjacent elements of the code.

Cont..

- Given an image WD the 4 chain and 8 chain and found the shape number.
- Ans:-
- Steps to find Normalized Chain Code :-
  1. Calculate the Chain Code.
  2. Calculate Circulate 1st Difference (CFD)
  3. Calculate Normalize Chain Code or shape number
     [Normalized Circular 1st Difference (NCFD)]

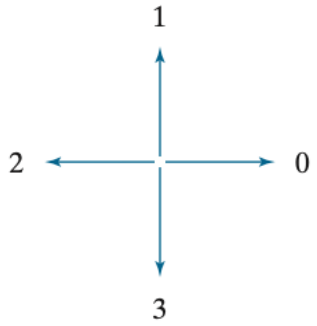
Calculate Normalize Chain Code or Shape Number



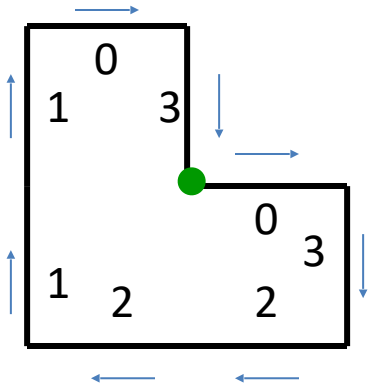
Chain Code: 03032211

CFD : 31330303

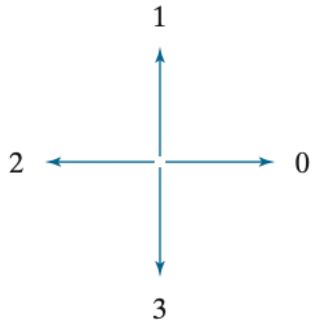
NCFD: 03033133



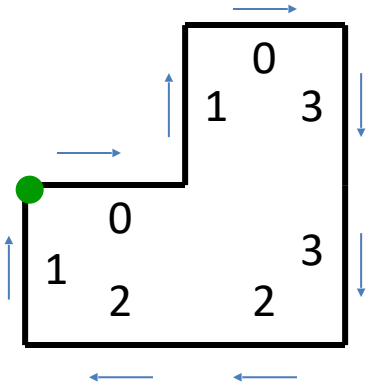
Change Starting Point



Chain Code: 03221103
CFD : 33030331
NCFD: 03033133



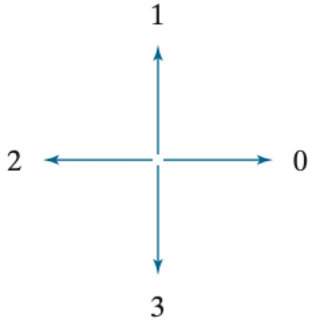
Reflex



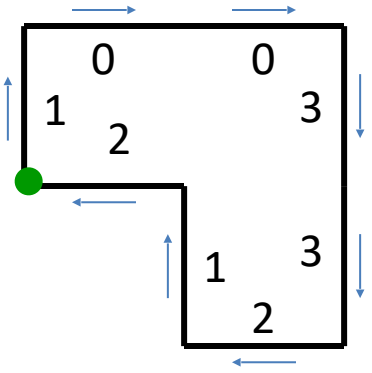
Chain Code: 01033221

CFD : 13303033

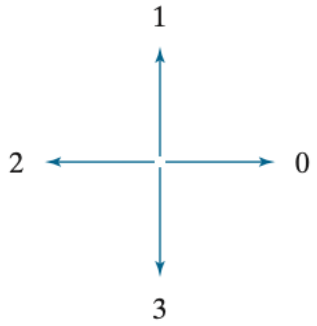
NCFD: 03033133



Rotation

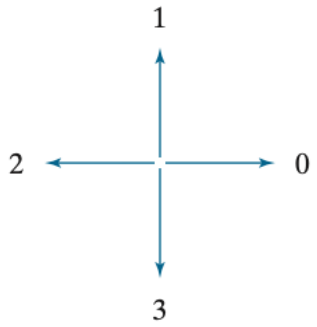


Chain Code: 10033212
CFD : 30303313
NCFD: 03033133



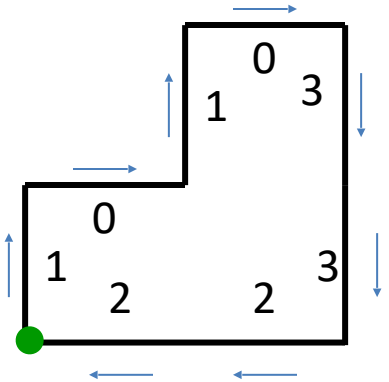
Cont..

- **Example:-** Chain Code 10103322 (in a counter-clockwise direction)

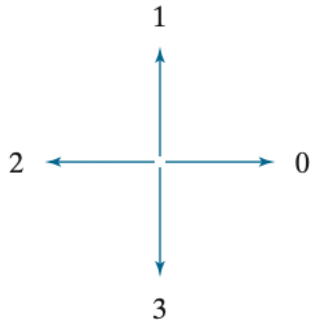


Cont..

- **Example:-** Chain Code 10103322 (in a counter-clockwise direction)

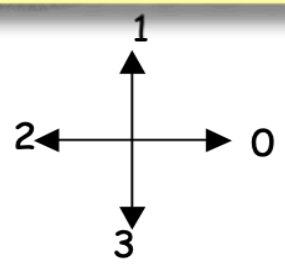


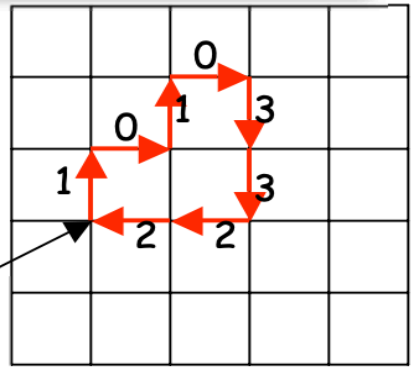
Chain Code: 10103322
CFD : 3133030
NCFD: 2201313



Cont..

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

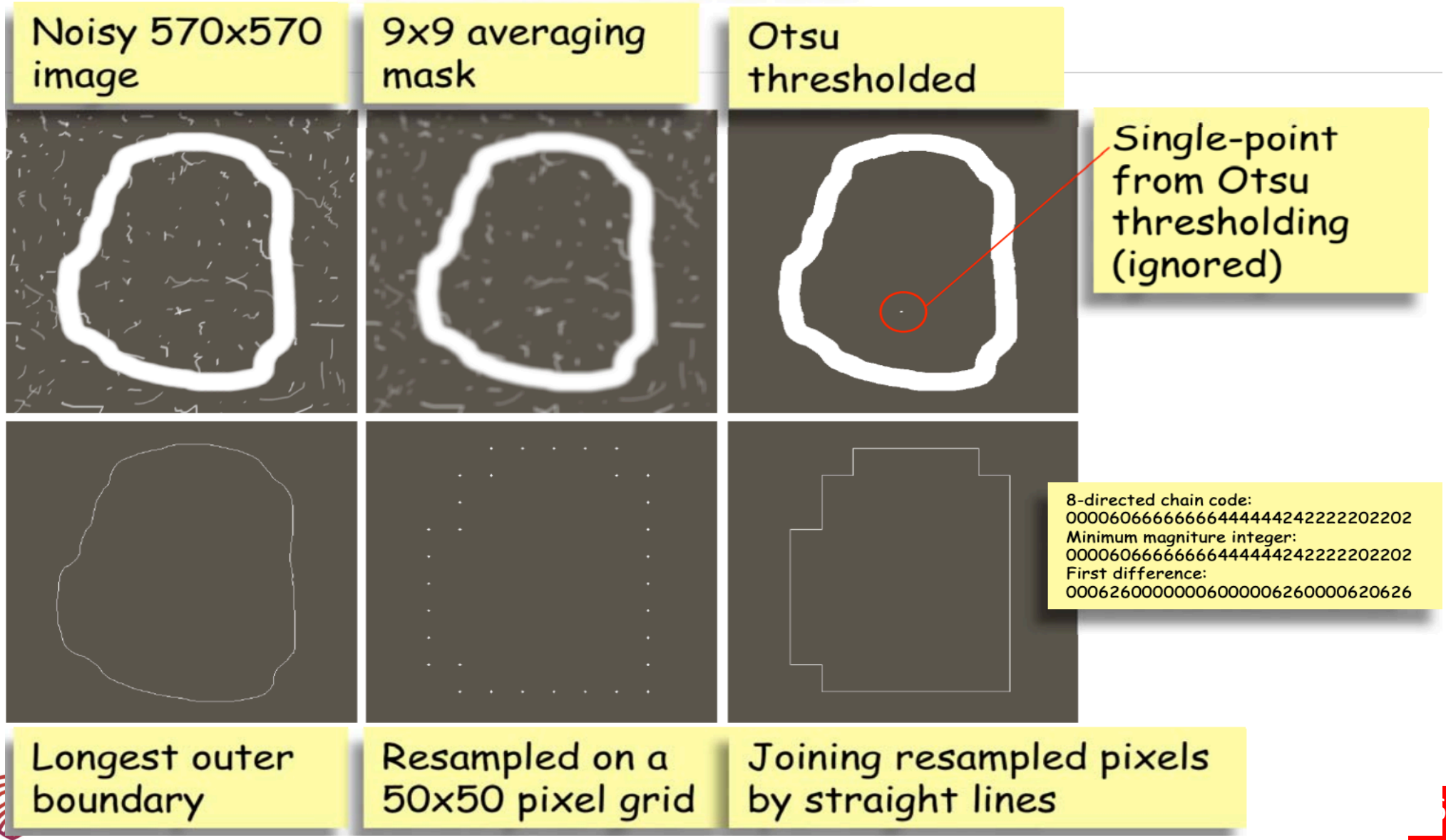
Chain codes can be normalized for rotation by using first differences.

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

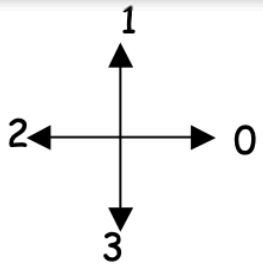
4-connected difference code:
33133030

Cont..



Shape Number – A boundary descriptor

4-direction chain code



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

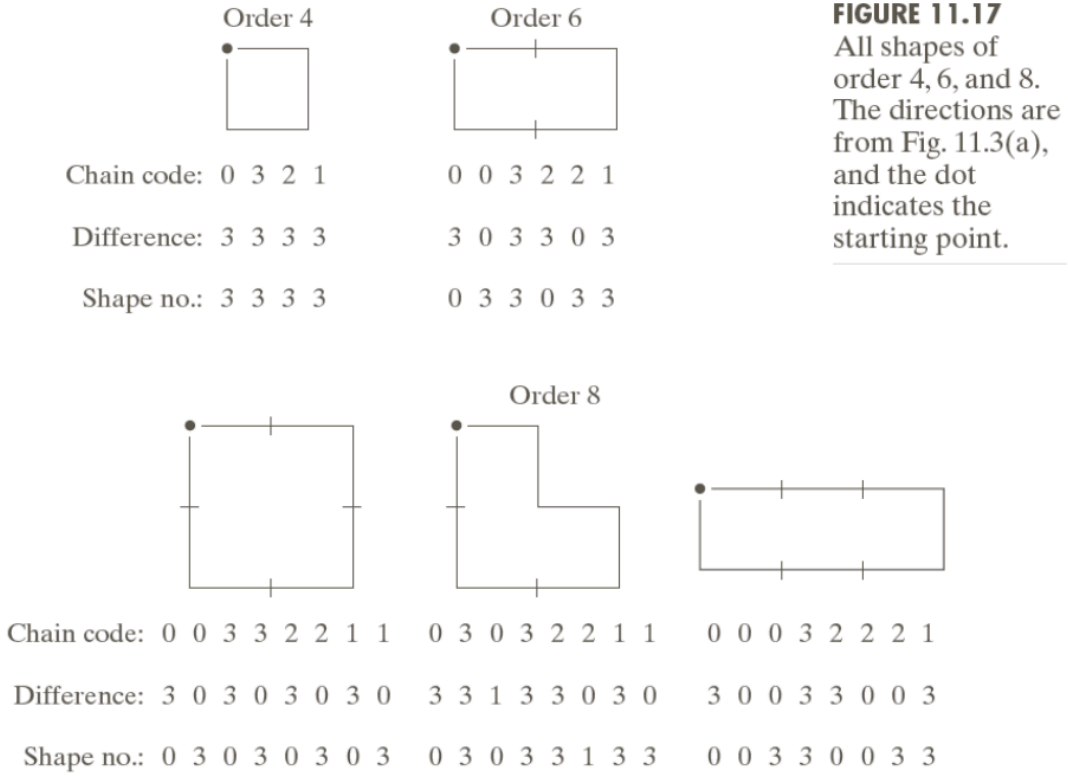


FIGURE 11.17 All shapes of order 4, 6, and 8. The directions are from Fig. 11.3(a), and the dot indicates the starting point.

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

Cont..

Definitions:

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

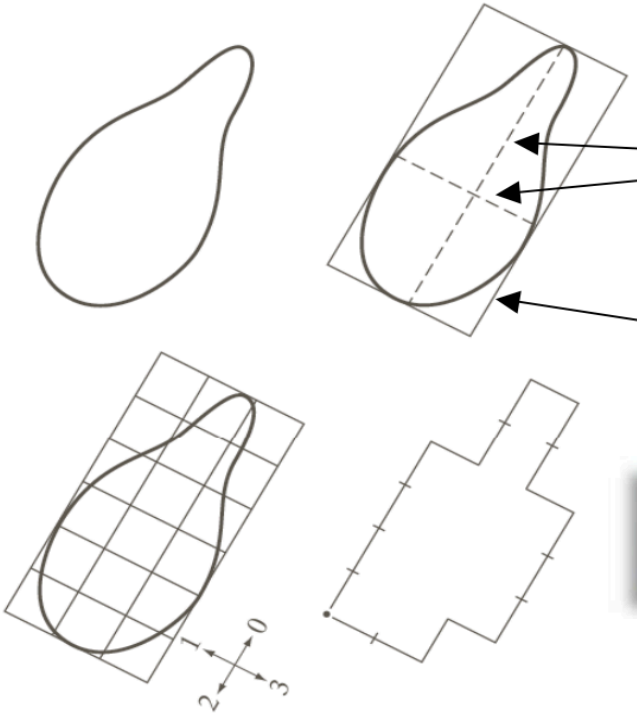
Major axis is the line segment of length equal to the diameter and connecting two points on the boundary.

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.

Cont..



a b
c d

FIGURE 11.18 Steps in the generation of a shape number.

Chain Coding issues/drawbacks

- Code becomes very long and noise sensitive
  - Use larger grid spacing, smooth/edit the code
- Scale dependent
  - Choose appropriate grid spacing
- Start point determines result
  - Treat code as circular (minimum magnitude integer) 754310 -> 075431
- Depends on rotation
  - Calculate difference code (counter-clockwise) 075431 -> 767767

Cont..

- Chain codes are generated by following a boundary in a clockwise or counter-clockwise direction and assigning a direction to the segments connecting every pair of pixels.

- Disadvantage: Can be unacceptably long.
  - Solution: Re-sampling (down sample) the boundary

- Disadvantage: Is starting point dependent
  - Solution: Normalize the representation string to the smallest integer.

Thank You