Computer Vision - CS452

Introduced by

Dr. Ebtsam Adel

Associate professor at IS Dept.
Faculty of computers and information
Damanhour university

QUIZ!

Quiz

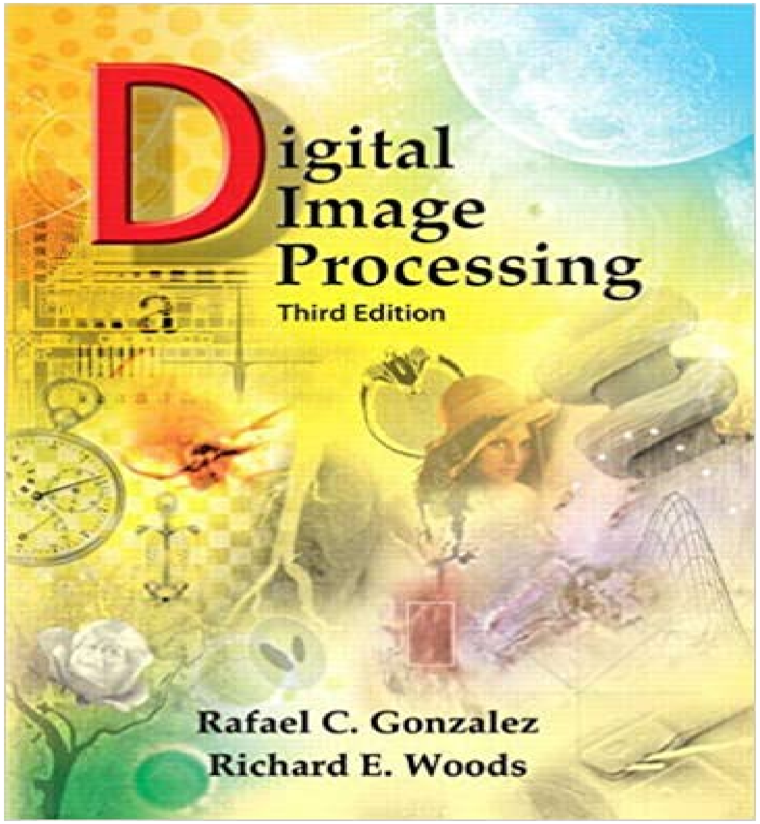
- A 4x4 image is given by: **(6 G)**

100	120	160	120
80	90	100	100
90	90	50	120
40	100	19	150

- Filter the image using a **median filter** (padding with zeros), use 3x3 filter mask.

- Apply the fitter to all pixels in the image.

Chapter 11: Representation and Description



Digital Image Processing

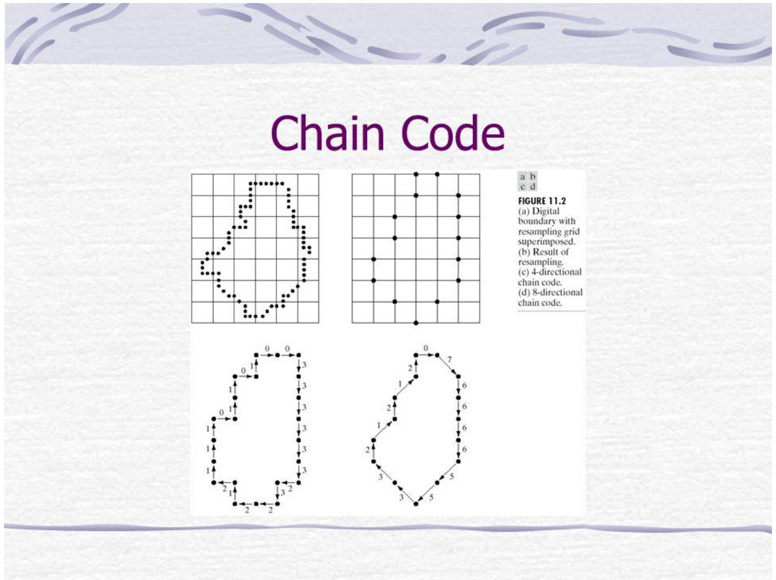
Third Edition

Rafael C. Gonzalez

Richard E. Woods

Introduced by

Dr. Ebtsam Adel



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.

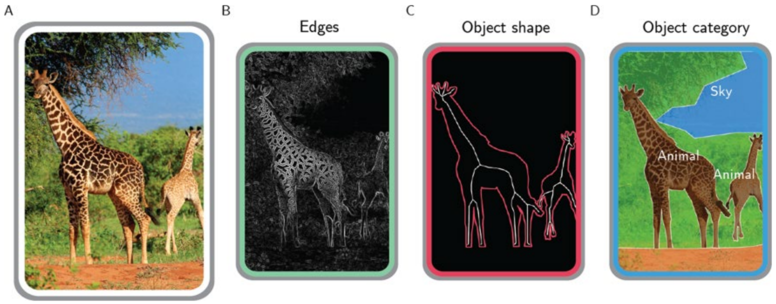


Image Representation and Description

- After an image has been segmented into regions, the resulting aggregate of segmented pixels usually is **represented and described** in a form suitable for further computer processing.

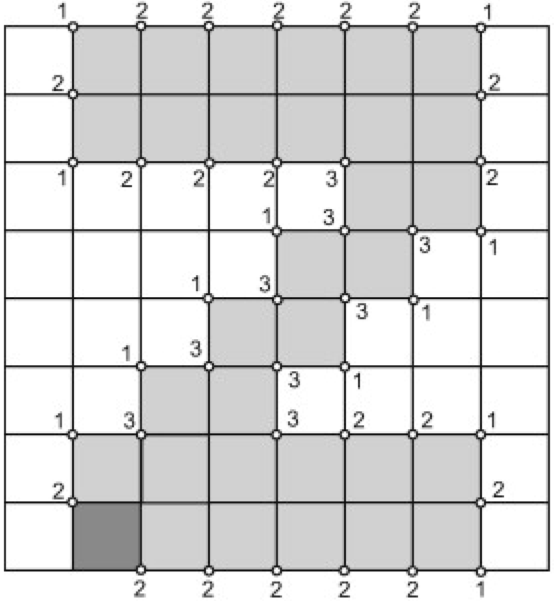


Image Representation and Description

- **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
  - Relation between regions

Image Representation & Description

- Basically, **representing** a region involves **two choices**:

1. We can represent the region in terms of its **external characteristics** (its boundary) [such as numbers & letters],
2. we can represent it in terms of its **internal characteristics** (the pixels comprising the region).

- The next task is to **describe** the region based on the chosen representation. For example, a region may be represented by its boundary, and the boundary described by features such as its length, the orientation of the straight line joining its extreme points, and other features.

Image Representation and Description

- ❑ An **external representation** is chosen when the primary focus is on shape characteristics.
- ❑ An **internal representation** is selected when the primary focus is on regional properties, such as color and texture.

- ❑ Sometimes it may be necessary to use both types of representation.

Common Representation

Common external representation methods are:

- **Chain codes**
- **Polygonal Approximations**
- **Boundary Segments**
- **Skeletons**

Chain Code

Shape Representation by Using Chain Codes

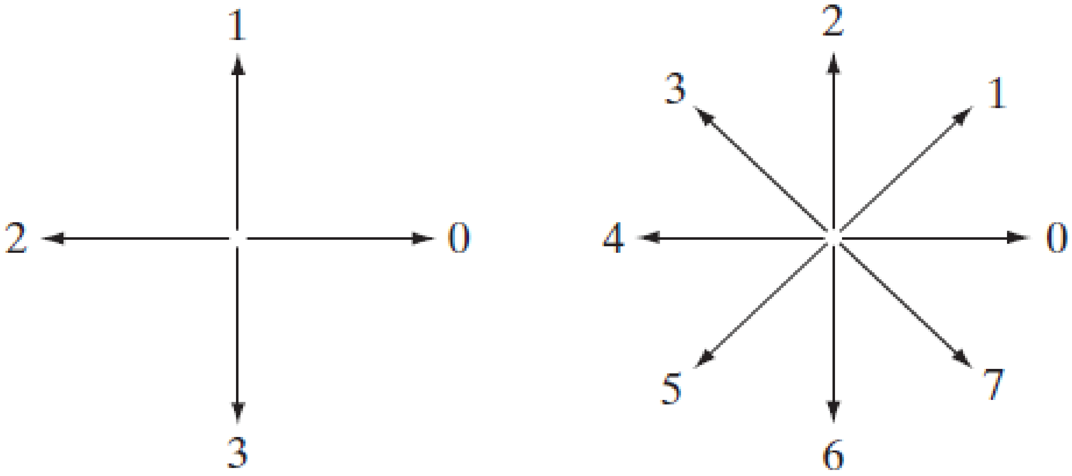
- **Chain code represents boundary.**
- **Why we focus on a boundary?**

The boundary is a good representation of an object shape and also requires a few memory.

- **Chain codes:** represent an object boundary by a connected sequence of straight line segments of specified length and direction.

4-connectivity

8-connectivity

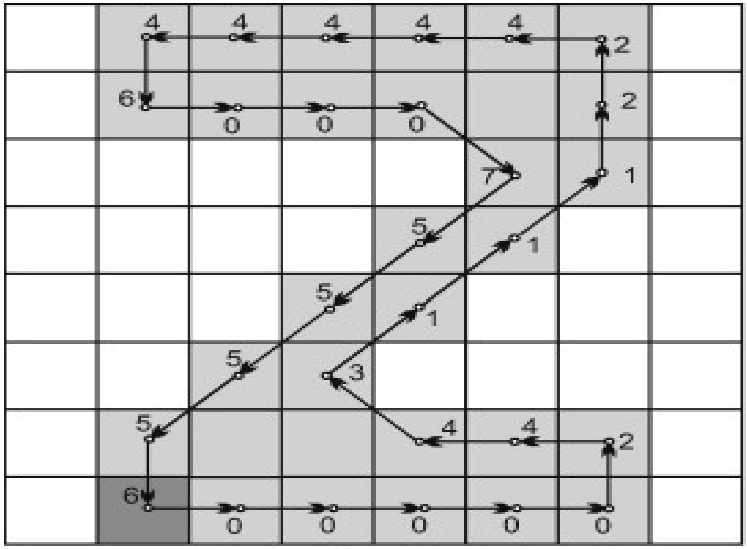


a b

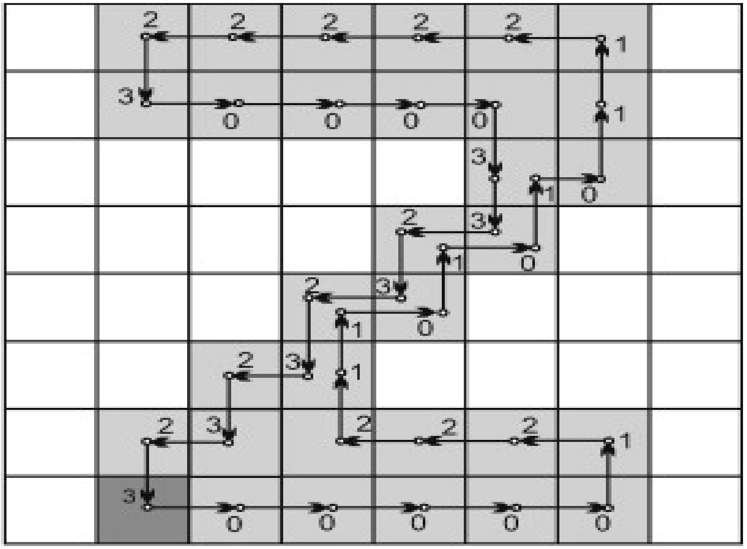
FIGURE 11.3 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.

Counter clockwise

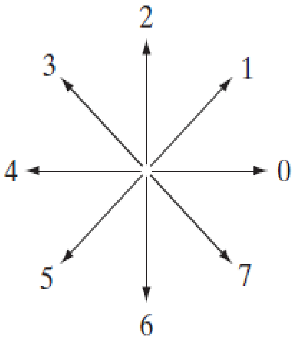
Shape Representation by Using Chain Codes

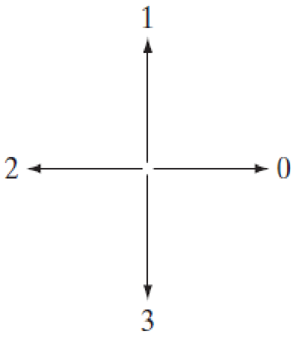


(a)

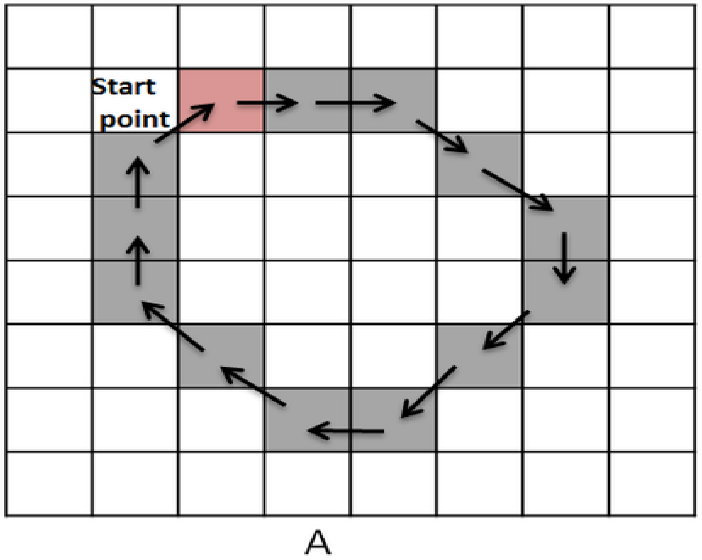


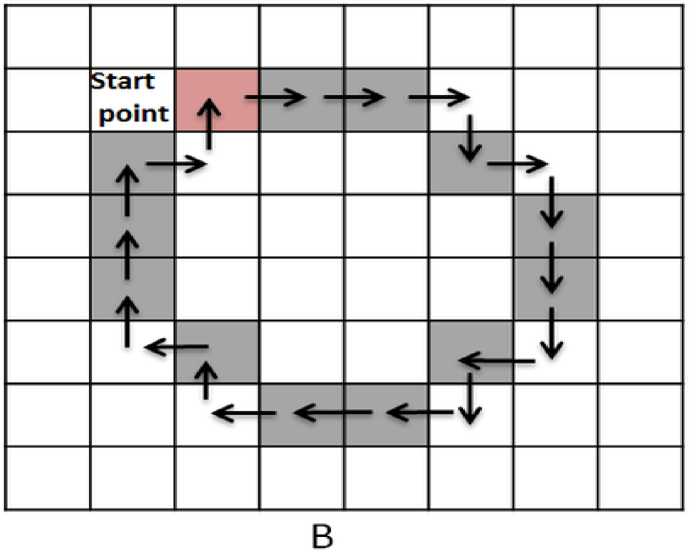
(b)





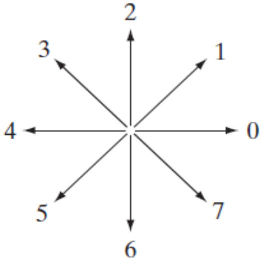
Shape Representation by Using Chain Codes

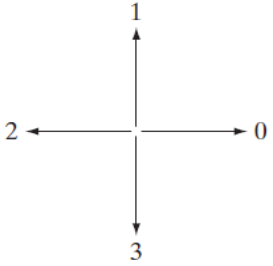




0077655433221

000303332322221211101



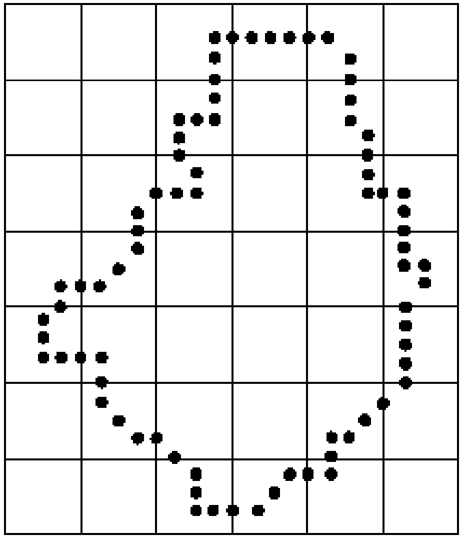


- A boundary code formed as a sequence of such directional numbers is referred to as a **Freeman chain code**.

Shape Representation by Using Chain Codes

This method generally is unacceptable for two principal reasons:

1. The resulting chain tends to be quite long "**long chain code**".
2. any small trouble along the boundary due to **noise** or imperfect segmentation cause changes in the code that may not be related to the principal shape features of the boundary.



Shape Representation by Using Chain Codes

solution

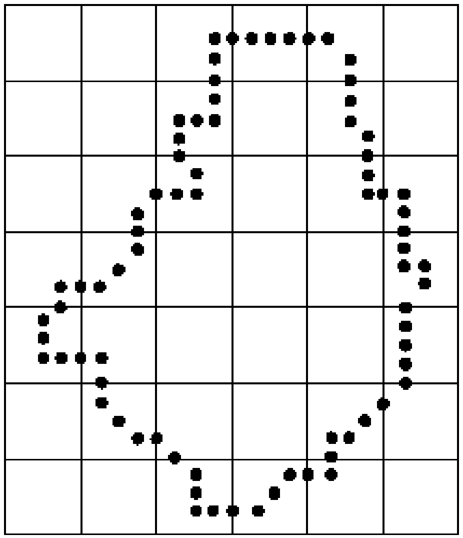
- ❑ An approach frequently used to overcome these problems is to resample the boundary by selecting **a larger grid spacing**.

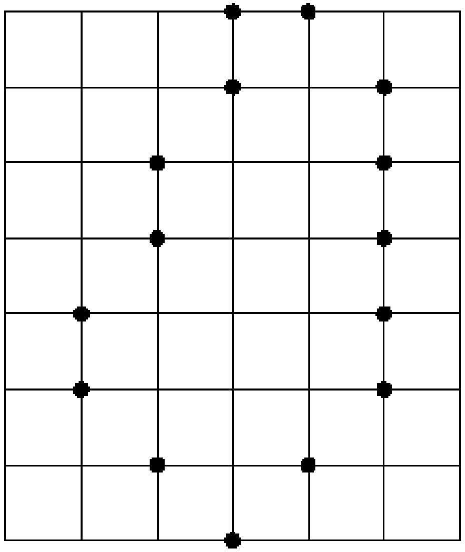
- ❑ Then, as the boundary is traversed, a boundary point is assigned to each node of the large grid, depending on the proximity of the original boundary to that node.

- ○ The nearest neighbor regarding to the grid.
- ○ Less noise.
- ○ Less chain code long.
- ○ Trade off: loss of some data.

Resampling for Chain Codes

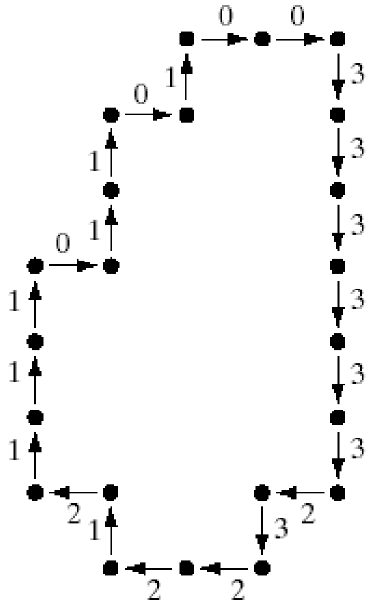
Object boundary (resampling)

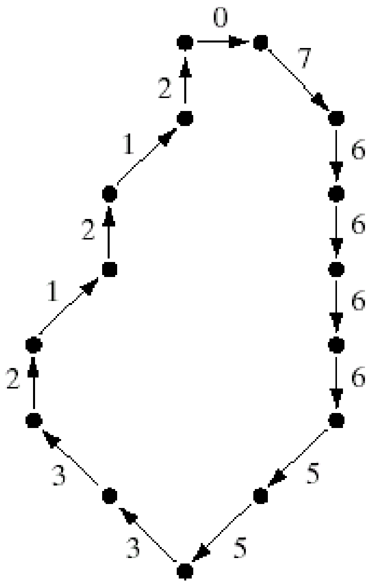




Boundary vertices

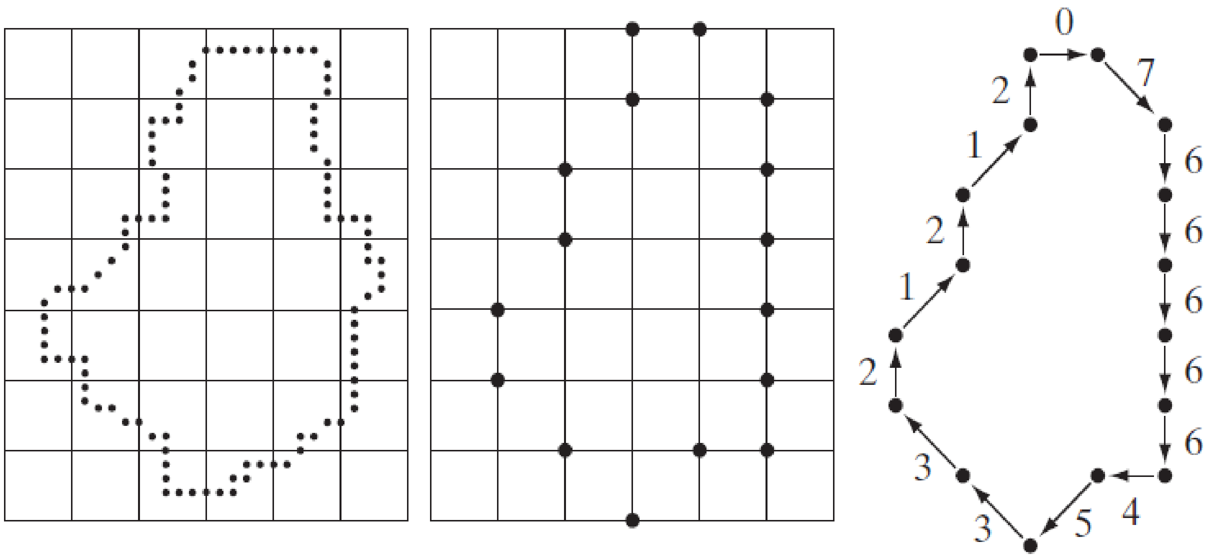
4-directional chain code





8-directional chain code

Shape Representation by Using Chain Codes - Grid spacing



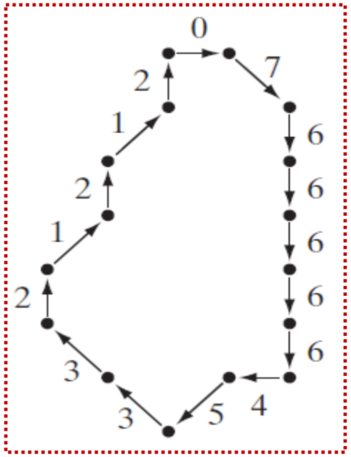
a b c

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

- ➢ As might be expected, the accuracy of the resulting code representation depends on the spacing of the sampling grid.
- ➢ Depend on The starting point.
- ➢ Depend on The rotation “i.e. **rotation variant**”.

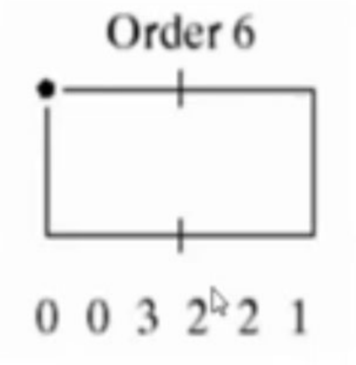
Chain Codes- Normalization for starting point

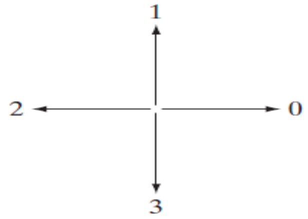
- ❑ The chain code of a boundary depends on the starting point. However, the code can be **normalized with respect to the starting point** by a straightforward procedure.
- ❑ We simply treat the chain code as a circular sequence of direction numbers and redefine the starting point so that the resulting sequence of numbers forms an integer of **minimum magnitude**.
- ❑ E.g. 101003333222 is **normalized** to 003333222101.



Chain Codes- normalization for starting point

Treat the 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**.





3 2 2 1 0 0 $\xrightarrow{\text{Normalization}}$ 0 0 3 2 2 1

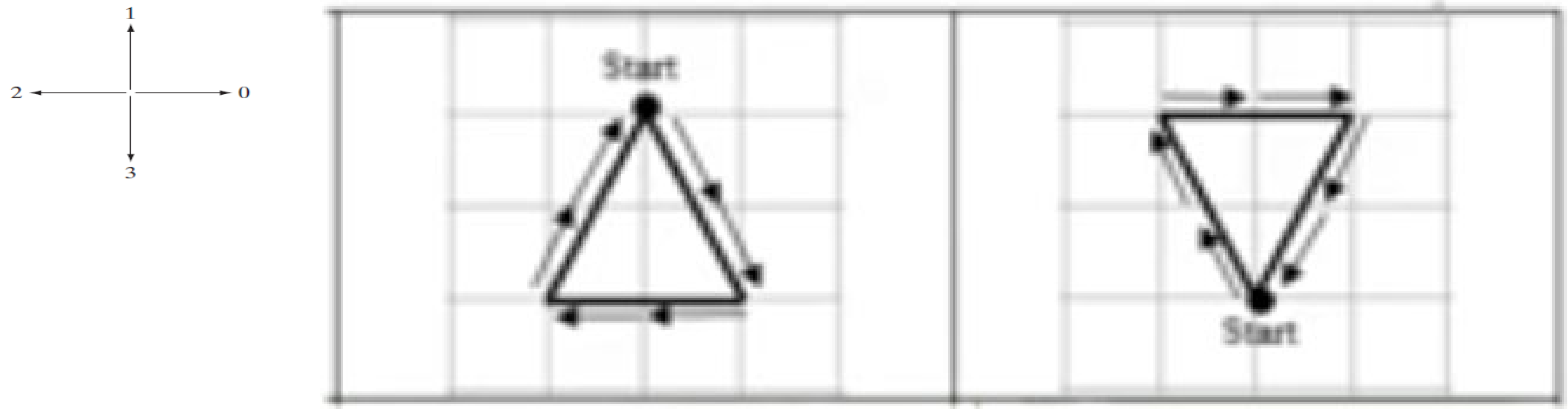
2 2 1 0 0 3 $\longrightarrow$ 0 0 3 2 2 1

1 0 0 3 2 2 $\longrightarrow$ 0 0 3 2 2 1

Normalization for Chain Codes- normalization for rotation

- We can normalize also for **rotation** (in angles that are integer multiples of the directions) by using the **First Difference (FD)** of the chain code instead of the code.
- This difference is obtained by counting the number of direction changes (in a **counterclockwise** direction).

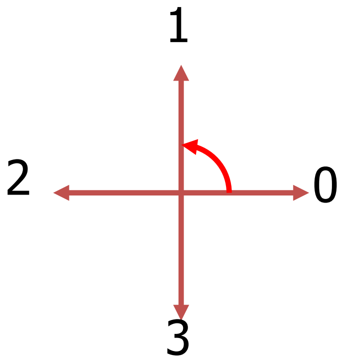
- For instance, the first difference of the 4-direction chain code 10103322 is 3133030.



The First Difference of a Chain Codes

- **The first difference of a chain code:** counting the number of direction change (in counterclockwise) between 2 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

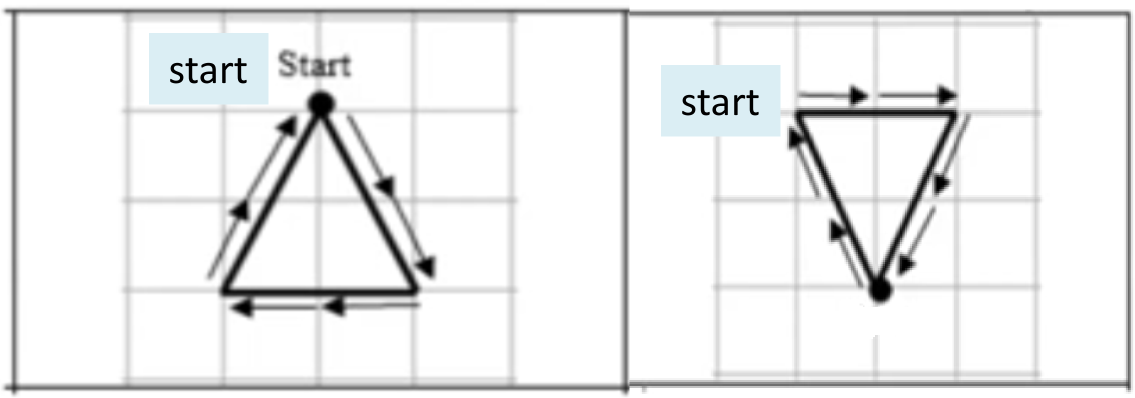
Example:

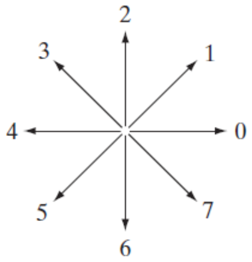
- a chain code: 10103322
- The first difference = 3133030
- Treating a chain code as a circular sequence, we get the first difference = **3**3133030

Counter clockwise

The first difference is rotational invariant.

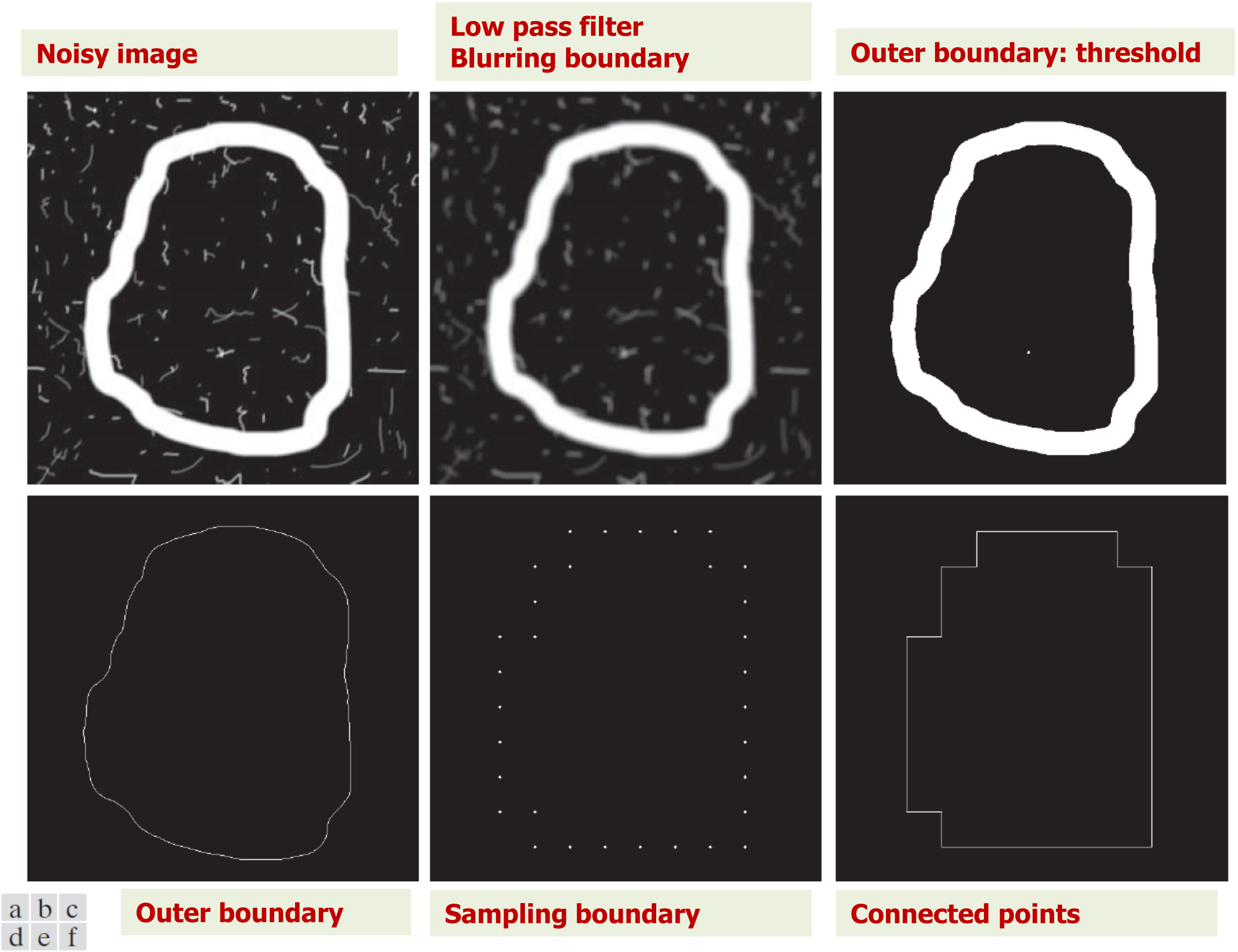
Example on Chain Code





Chain code	**774411**	**005533**
Normalization for **rotation** (**FD**)	050506	050605
Normalization for **starting point**	050506	050506

THE SAME OBJECT



Noisy image

Low pass filter
Blurring boundary

Outer boundary: threshold

Outer boundary

Sampling boundary

Connected points

FIGURE 11.5 (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).

Chain Code

- The 8-directional Freeman chain code of the simplified boundary is:

0 0 0 0 6 0 6 6 6 6 6 6 6 6 4 4 4 4 4 4 2 4 2 2 2 2 2 0 2 2 0 2

- The integer of **minimum magnitude** of the code happens in this case to be the same as the chain code:

0 0 0 0 6 0 6 6 6 6 6 6 6 6 4 4 4 4 4 4 2 4 2 2 2 2 2 0 2 2 0 2

PRACTICAL PART

Practical part

Generate image matrix from freeman chain code.

- import numpy as np
- import matplotlib.pyplot as plt

Thank You !
Any questions?