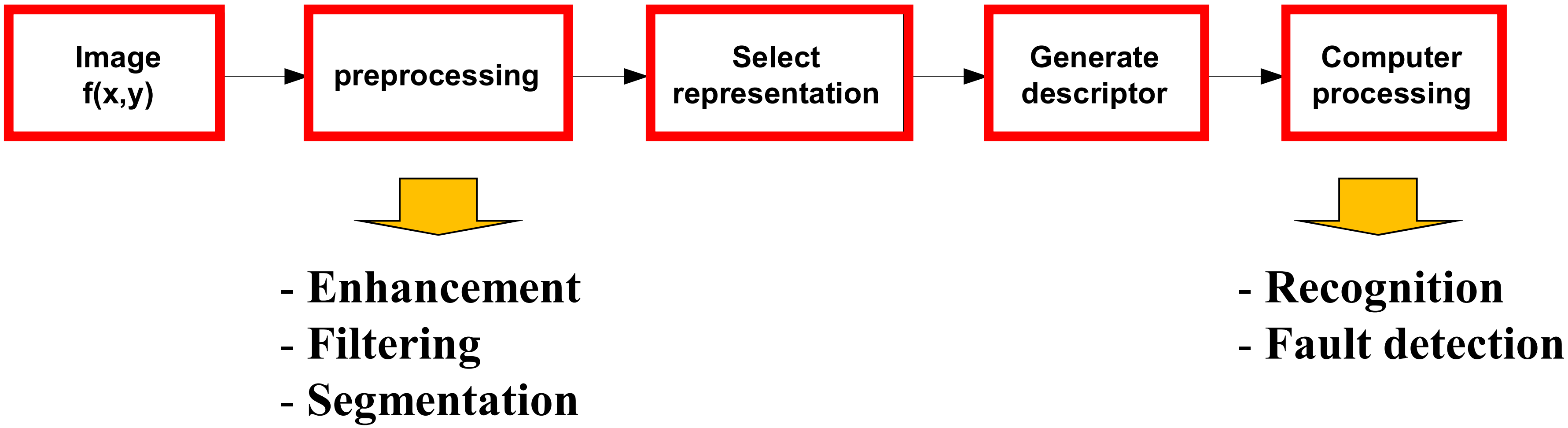
Mahidol University *Wisdom of the Land*

Chapter 12

Representation and Description

Representation and Description



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 and Description

- After segmentation, the image needs to be described and interpreted.
- Representation : an object may be represented by its boundary.
- Description : the object boundary may be described by its length, orientation, or number of concavities.

Image Representation

- **It is a representative of the image**
  - Global representation
    - Global color or texture

- **Local representation**
  - Represents just parts of the image
    - Block representation
    - Region representation

Image Descriptors

- **Generate an abstract of the image**
  - Perform feature extraction
    - Use the extracted feature to generate descriptions

- **Type of descriptors**
  - Represents just parts of the image
    - Global descriptors: color, texture
    - Local descriptors: color, texture, shape

Color Description

- **Type of color description**
  - Histogram (RGB, CMY, YCbCr, HSI, and HSV)
    - Separate-color histogram (H_1,H_2,H_3)
    - Combined-color histogram (H_{RGB})
    - Number of elements in histogram: bin

- **Color Statistics**
  - Color mean
  - Color variance

Statistics

$$Mean = \overline{X} = \frac{\Sigma \mathrm{x}}{\mathrm{n}}$$

$$Variance = \sigma^2 = \frac{\sum (X - u)^2}{N}$$

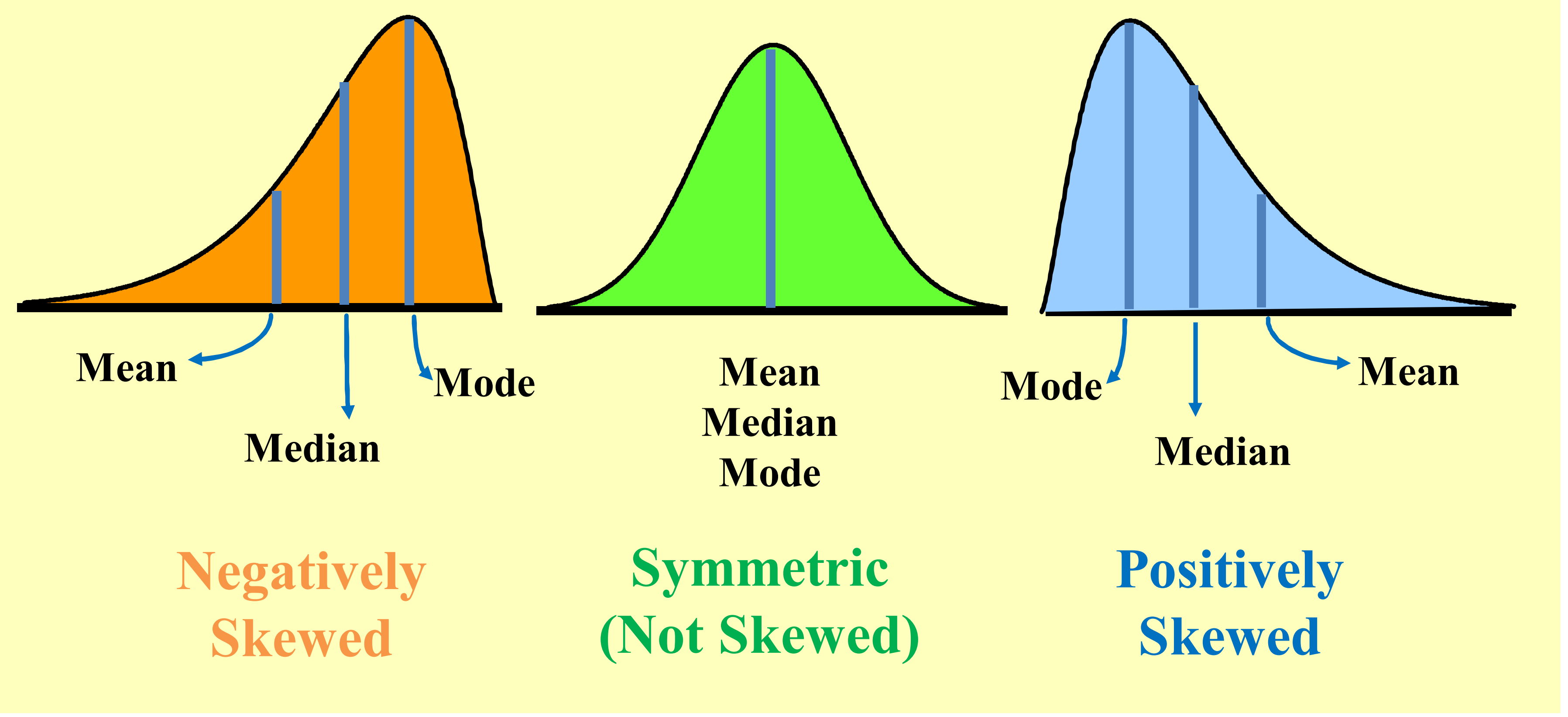
$$\text{Skewness} = \frac{\sqrt{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{3/2}}$$

$$\text{Kurtosis} = \frac{n\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{2}} - 3$$

Color Description

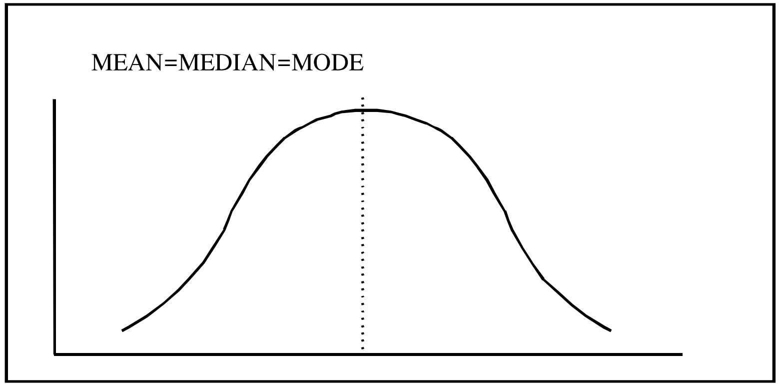
$$Mean = \overline{X} = \frac{\Sigma x}{n}$$

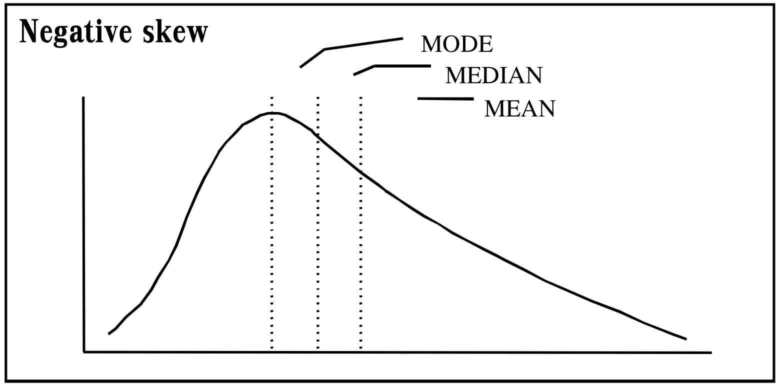
$$Variance = \sigma^2 = \frac{\sum (X-u)^2}{N}$$



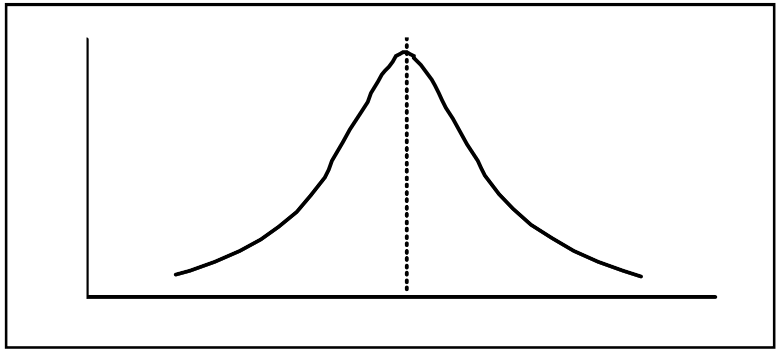
Symmetry vs. Asymmetry

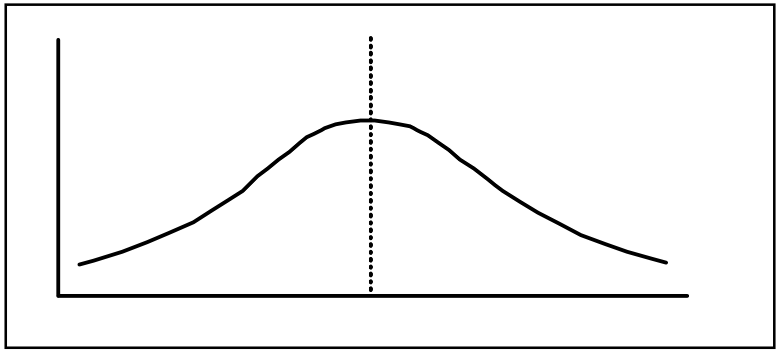
Skew - asymmetry





Kurtosis - peakedness or flatness



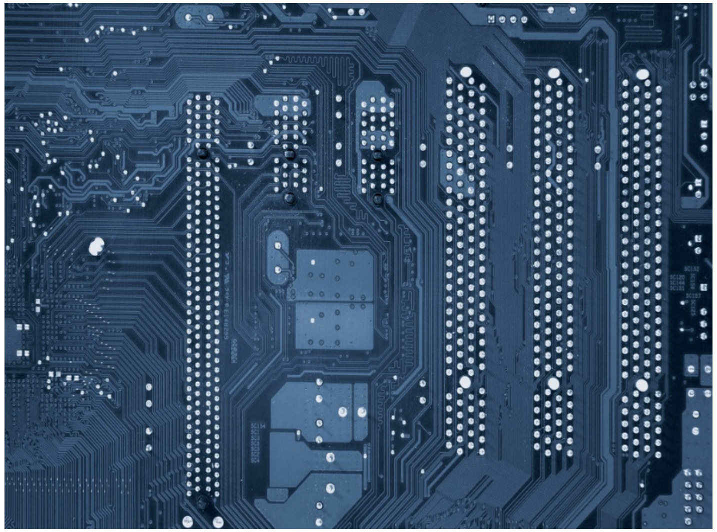


Texture Descriptors

Grass texture

Brick ground texture

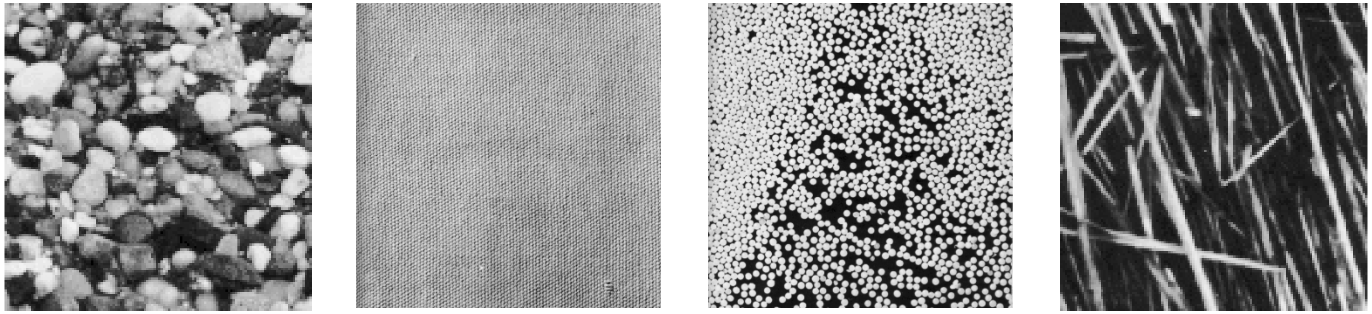
Wood texture



Circuit board texture

Texture Descriptors

- **Texture is one of the important characteristics used in identifying objects or regions of interest in an image**
  - Human describe texture as



Texture Descriptors

- **Describes surface of the image**
  - Usually obtain from Grayscale image
  - Local texture gives better description of the surface

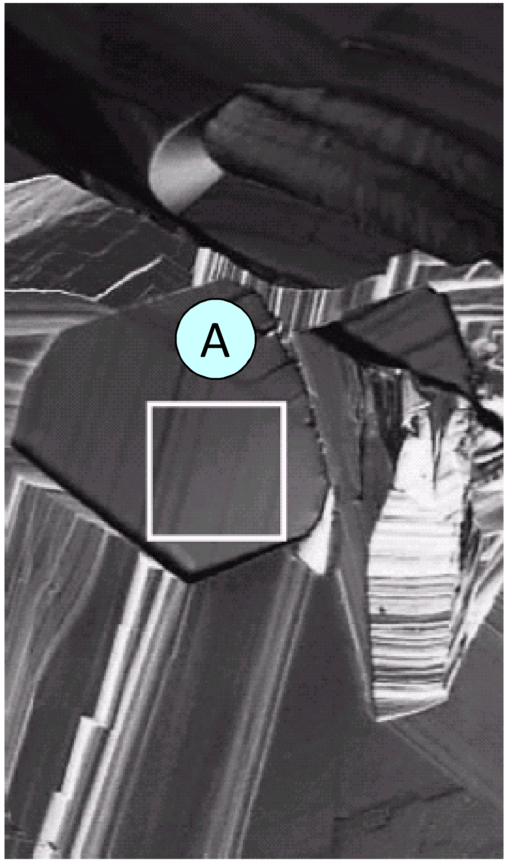
- **Type of texture description**
  - Histogram-based texture description (Smoothness)
    - Standard deviation
    - Contrast
    - Energy
    - Entropy

Texture Descriptors

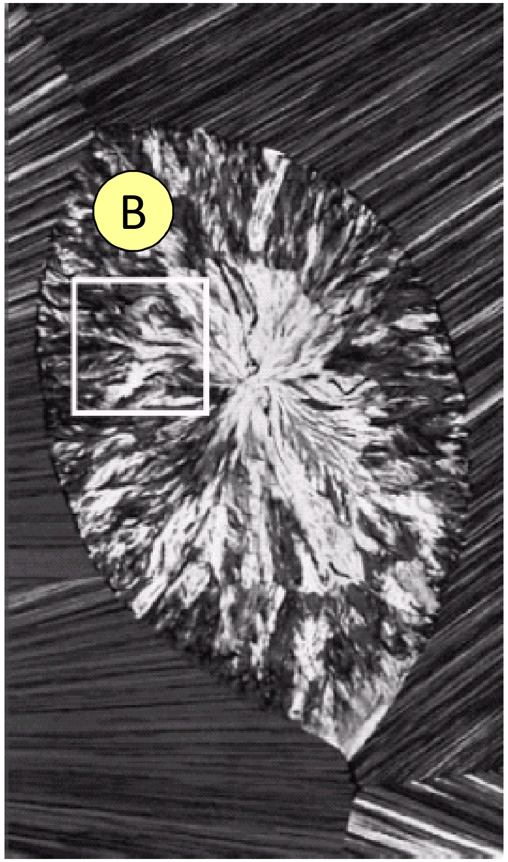
- **Texture is usually defined as the smoothness or roughness of a surface.**
  - In computer vision, it is the visual appearance of the uniformity or lack of uniformity of brightness and color.
- **There are two types of texture: random and regular.**
  - **Random texture** cannot be exactly described by words or equations; it must be described statistically. The surface of a pile of dirt or rocks of many sizes would be random.
  - **Regular texture** can be described by words or equations or repeating pattern primitives. Clothes are frequently made with regularly repeating patterns.

Example : Texture

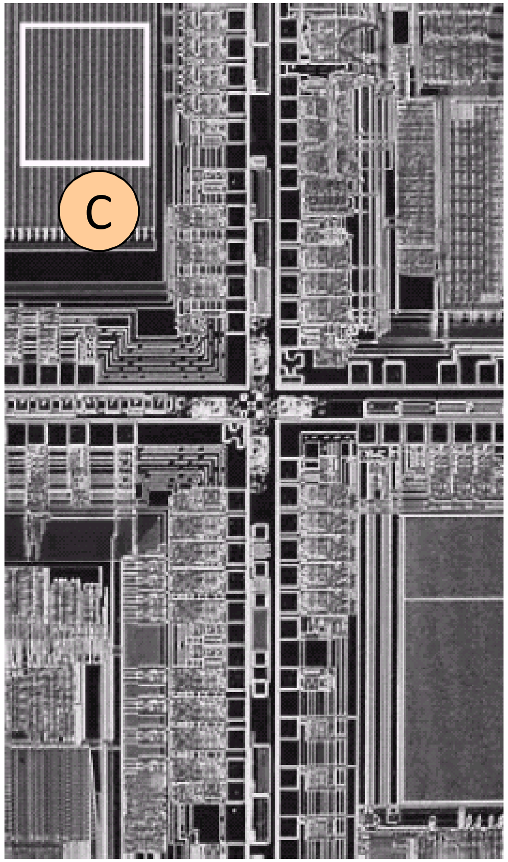
- optical microscope images



Superconductor
(smooth texture)

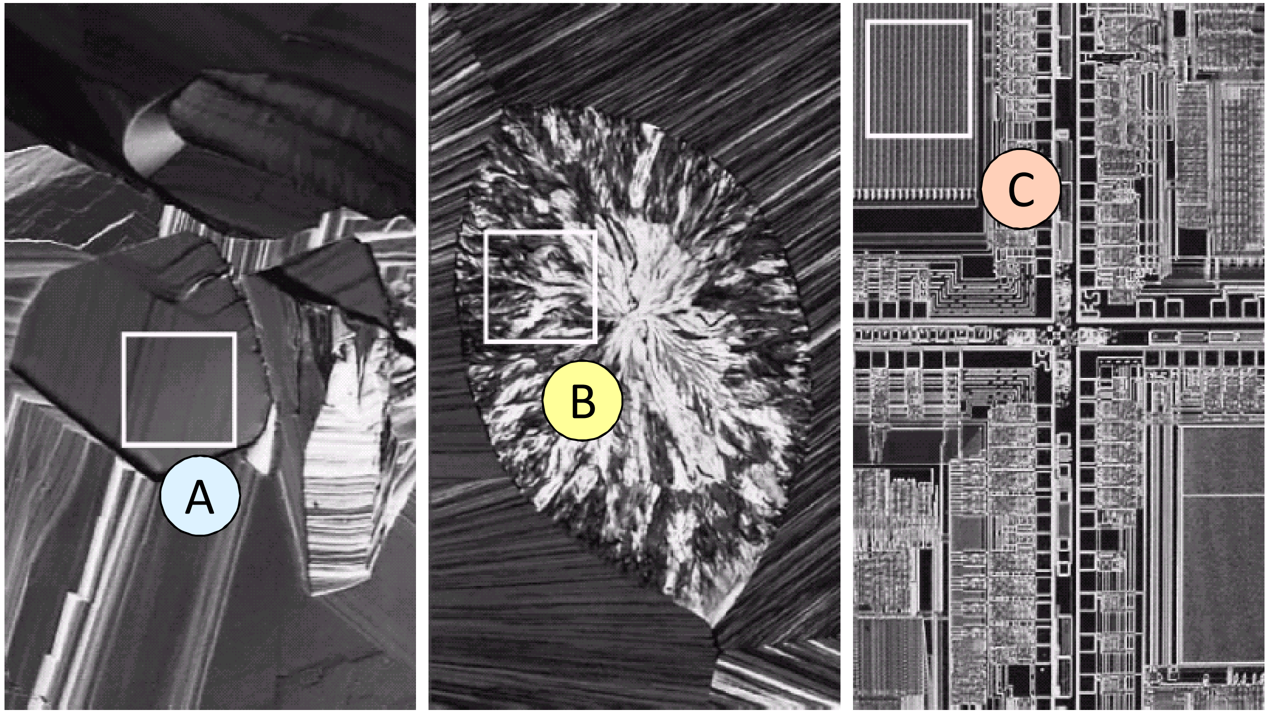


Cholesterol
(coarse texture)



Microprocessor
(regular texture)

Example : Texture



$$R = 1 - \frac{1}{1+\sigma^2(z)}$$

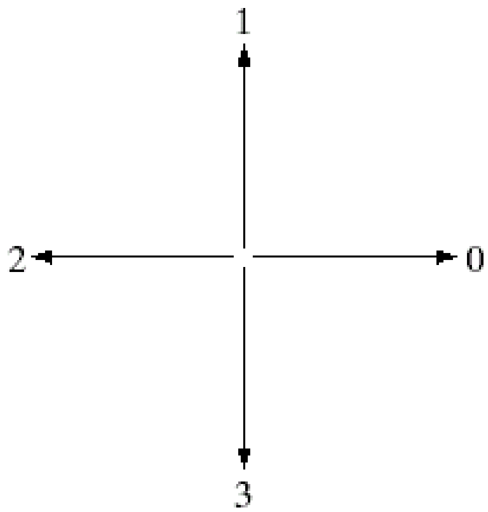
$$e = -\sum_{i=0}^{L-1} p(z_i)\log_2 p(z_i)$$

$$U = \sum_{i=0}^{L-1} p^2(z_i)$$

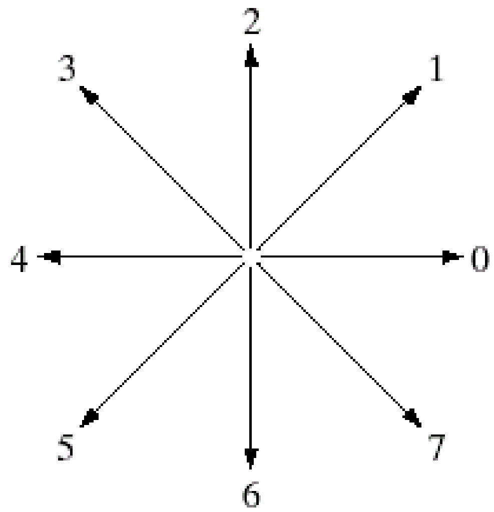
A	B	C
$\text{Mean} = 82.64$	$\text{Mean} = 143.56$	$\text{Mean} = 99.72$
$\sigma = 11.79$	$\sigma = 74.63$	$\sigma = 33.73$
$R = 0.002$	$R = 0.079$	$R = 0.017$
$U = 0.026$	$U = 0.005$	$U = 0.013$
$Entropy = 5.434$	$Entropy = 7.783$	$Entropy = 6.674$

Chain Codes

- 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.

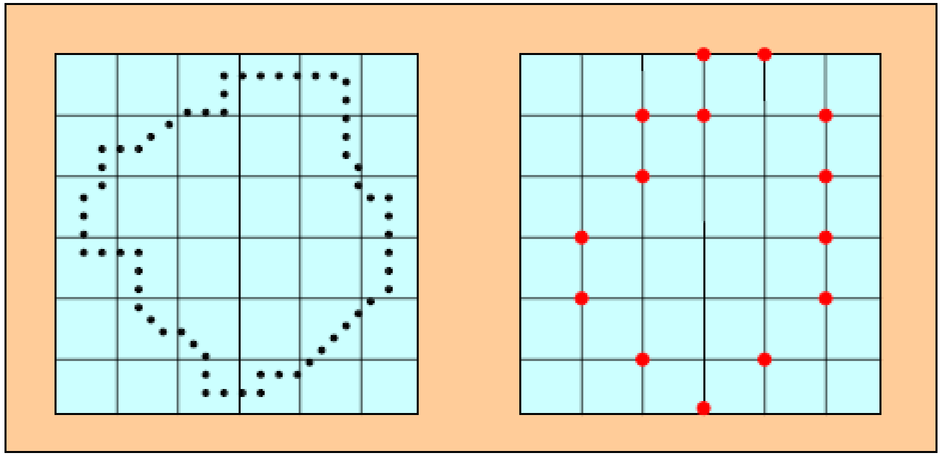


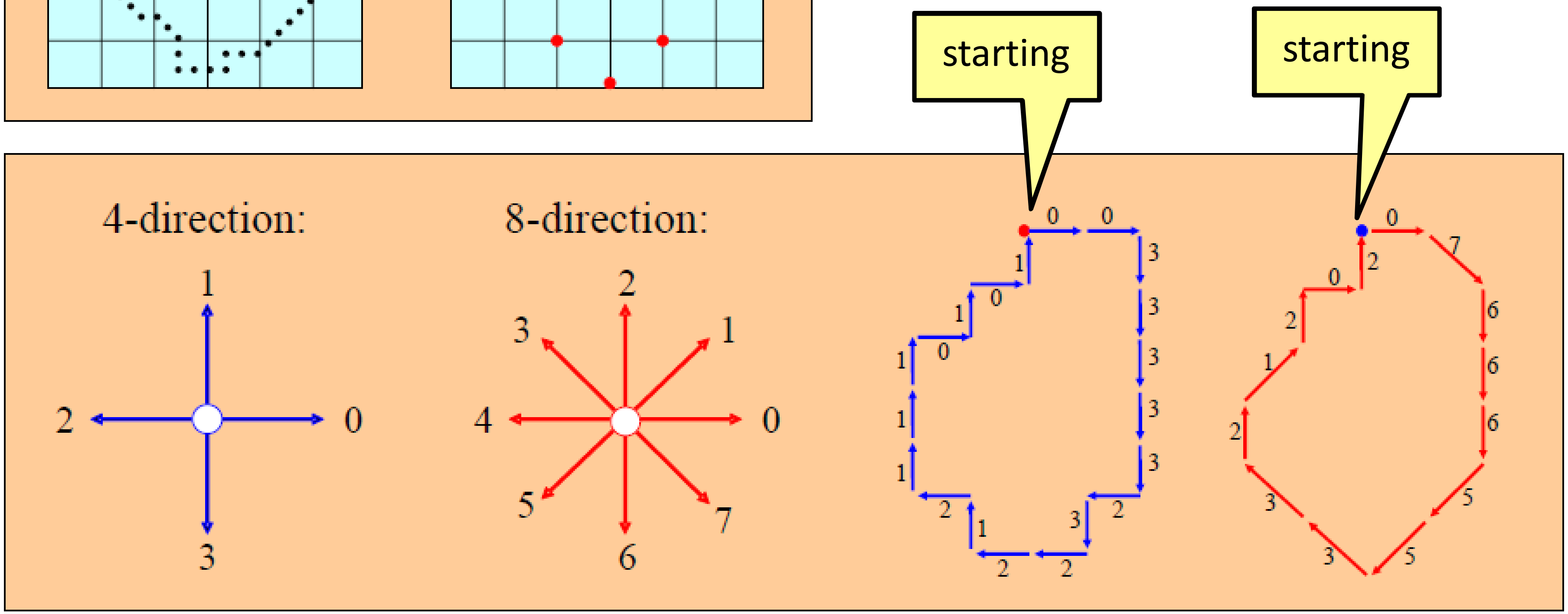
4-directional chain code



8-directional chain code

Example : Chain Codes





Chain code (clockwise):

→ 4-direction: 00333332322121110101, 8-direction: 07666553321202

Problem of a chain code

- Problem of a chain code:

  1. Different starting points result in different chain codes.

     - Solution : normalization → redefine the starting point such that the chain code forms a smallest number.

     * E.g.: 6553320000 → 0000655332

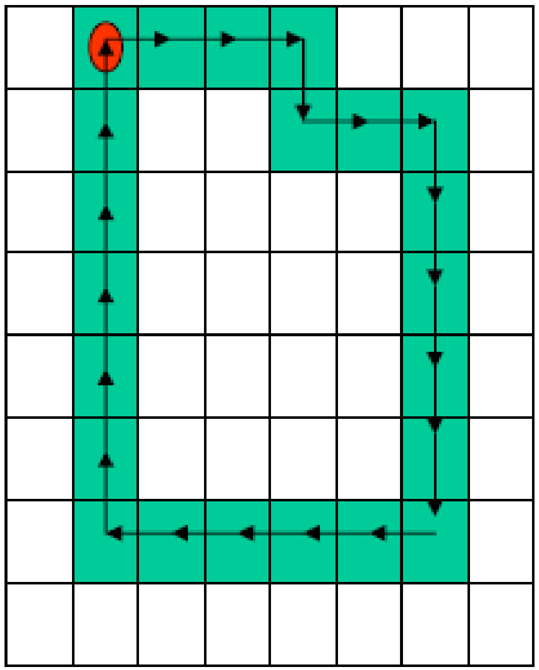
  2. Object rotation results in different chain codes.

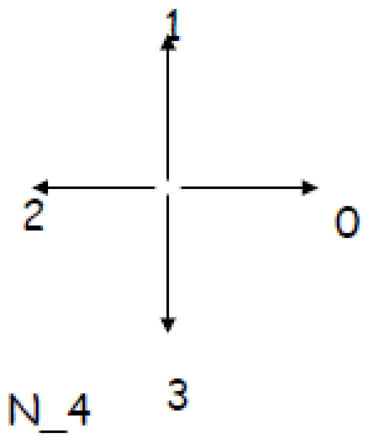
     - Solution : difference code → coding with the difference of directions (counter-clockwise).

     * E.g. 0000655332 → 0006706076 → 0006706076 (normalization)

Example : Chain Codes

Start



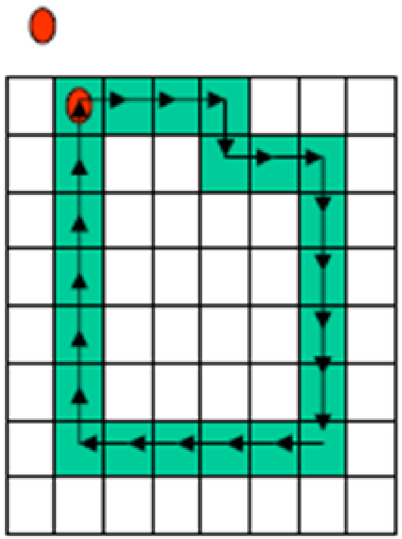


Chain Code

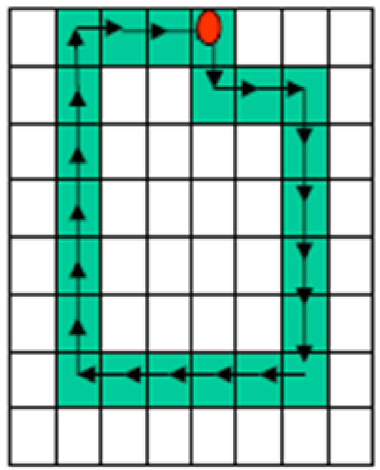
0, 0, 0, 3, 0, 0, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1

Example : Chain Codes

Start



Start



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

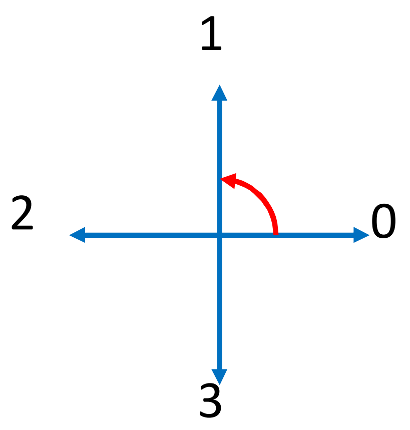
Chain Code 1

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

Chain Code 2

Normalized Code 0, 0, 0, 3, 0, 0, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1

Example : The difference of a Chain Codes



Example:

- 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

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

The first difference is rotational invariant.

Boundary Descriptors

- There are several simple geometric measures that can be useful for describing a boundary.
- The length of a boundary : the number of pixels along a boundary gives a rough approximation of its length.
- Curvature : the rate of change of slope
  - To measure a curvature accurately at a point in a digital boundary is difficult.
  - The difference between the slops of adjacent boundary segments is used as a descriptor of curvature at the point of intersection of segments.

Boundary Descriptors

Some Simple Descriptors :

- Length : sum of distances between adjacent pixels along the boundary.
- Diameter : maximum distance between pairs of pixels.

$$\mathrm{Diam}(B) = \max[D(pi, pj)]$$

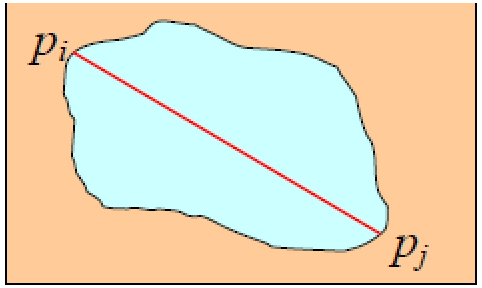
- Where :

  p_i and p_j = Point on Boundary

  B = Boundary

  D = Distance between pairs of pixels

Diameter: $\max_{i,j}(D(p_i, p_j))$



- The line connecting the two most distant boundary pixels is called the major axis of the boundary.

Example : Boundary Descriptors

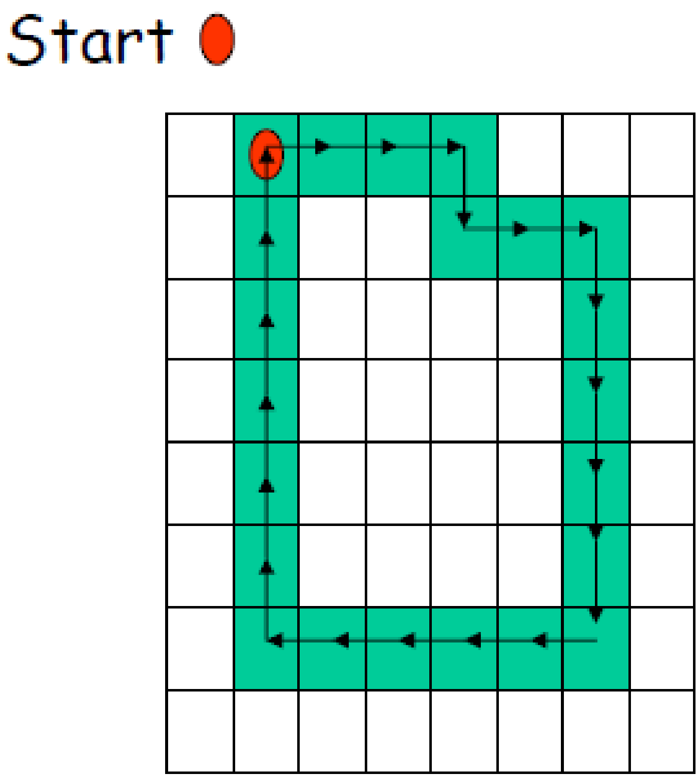
- MATLAB D = regionprops(L, properties)
  - L: labeled image
  - Properties: 'Area', 'BoundingBox', 'Centroid', 'ConvexArea', 'ConvexHull', 'ConvexImage', 'Eccentricity', EquivDiameter', 'EulerNumber', 'Extent', 'Extreme', FilledArea', 'FilledImage', 'Image', 'MajorAxisLength', 'MinorAxisLength', 'Orientation', 'PixelList', 'Solidity', 'all'
  - D: a structure with the fields specified when invoking regionprops( )

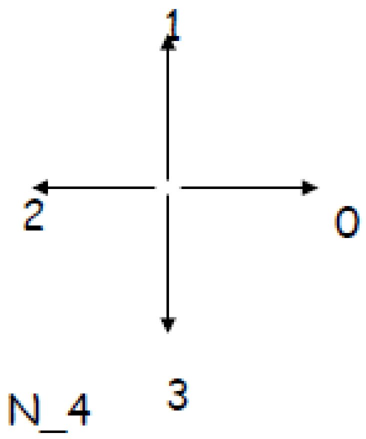
    * E.g., D. MajorAxisLength

Shape Numbers

- The first difference of the chain-coded boundary, having the smallest magnitude (normalized), is called the shape number of the boundary.
- The number of digits of the shape number is called the order of the shape number.
- The order of a shape number limits the number of different shapes it can encode.
- The shape number can also be defined using 8-directional-chain code.

Example : Shape Numbers

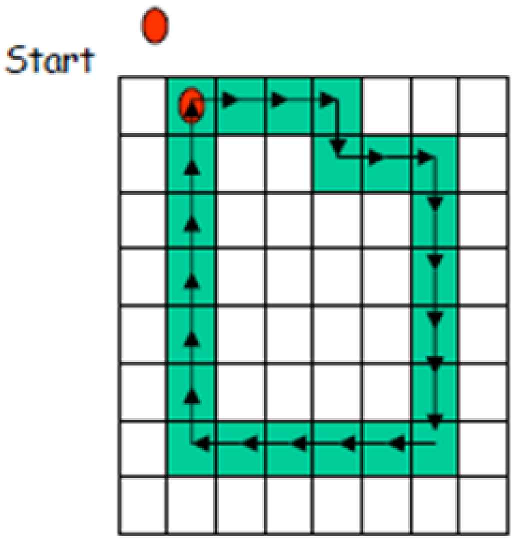


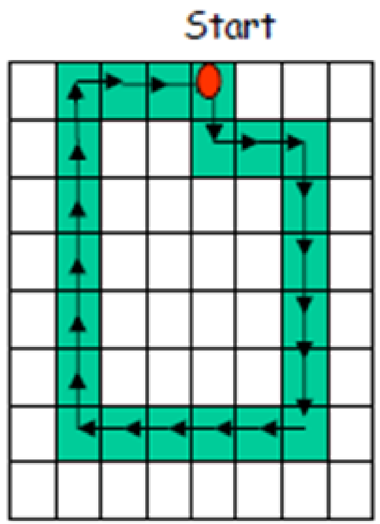


Shape number

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

Example : Shape Numbers





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

Chain Code 1

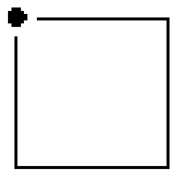
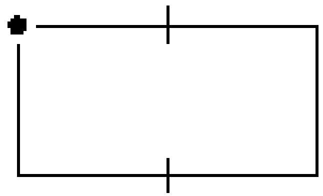
3, 0, 0, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0

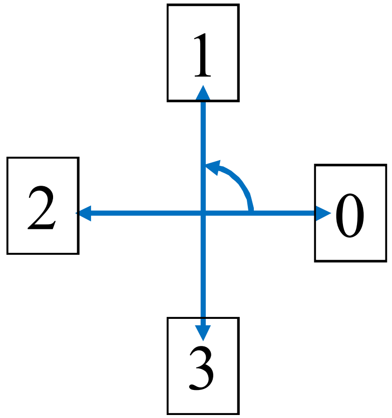
Chain Code 2

Shape number 0, 0, 0, 3, 0, 0, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1

Example : Shape Numbers

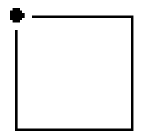
- Shape number of the boundary definition : the first difference of smallest magnitude.
- The order n of the shape number : the number of digits in the sequence.

	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

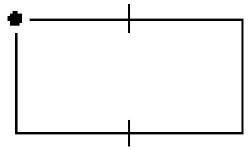


Example : Shape Numbers

Order 4



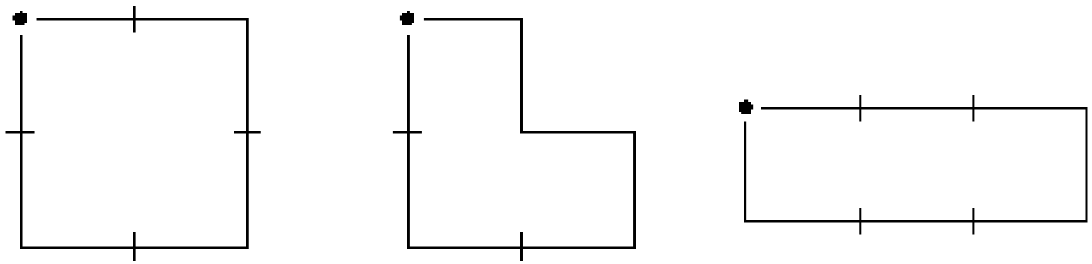
Order 6



	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

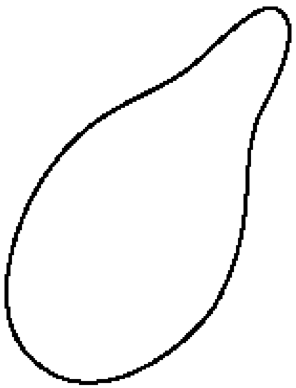
Shape numbers of order 4, 6 and 8

Order 8

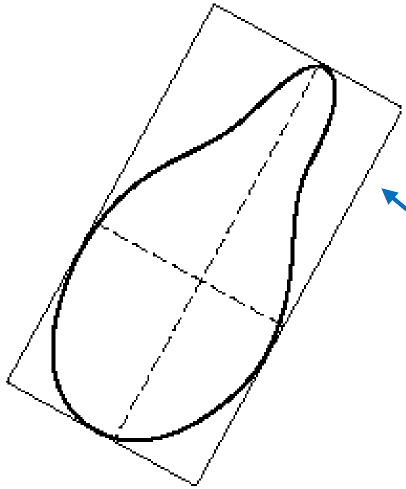


Chain code:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape no.:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

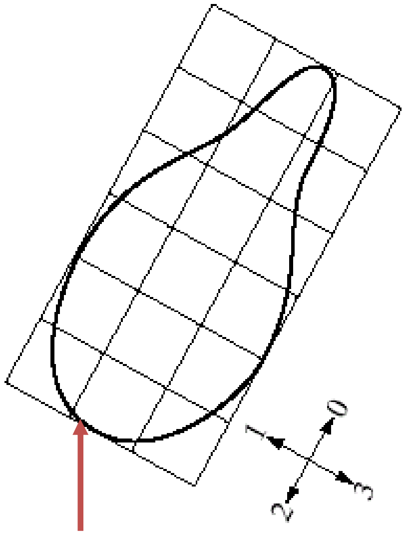
Example : Shape Numbers



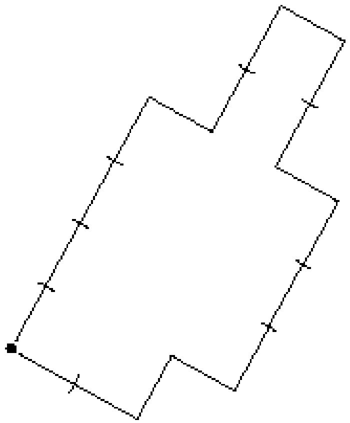
1. Original boundary



2. Find the smallest rectangle that fits the shape



3. Create grid



4. Find the nearest Grid.

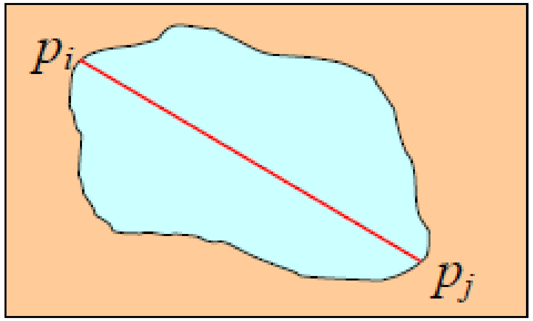
Chain code :
0 0 0 0 3 0 0 3 2 2 3 2 2 2 1 2 1 1

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

Example : Shape Numbers

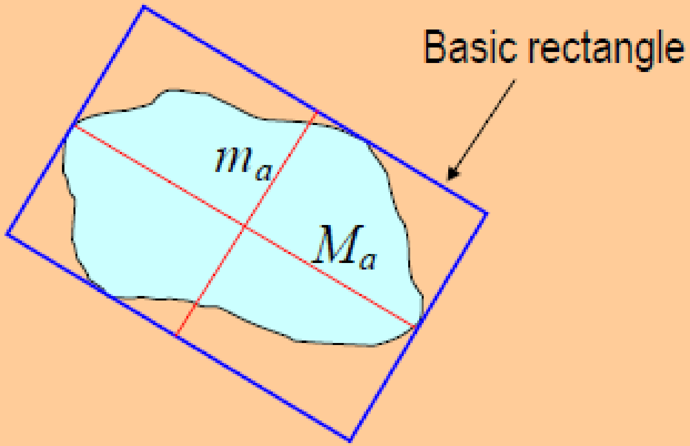
Diameter: $\max_{i,j}(D(p_i, p_j))$



Major axis (M_a) and minor axis (m_a)

$M_a = \max_{i,j}(D(p_i, p_j))$

$m_a \perp M_a$ and forms a *basic rectangle*



Eccentricity: M_a/m_a

Example : Shape Numbers

- MATLAB D = regionprops(L, properties)
  - L: labeled image
  - Properties: 'Area', 'BoundingBox', 'Centroid', 'ConvexArea', 'ConvexHull', 'ConvexImage', 'Eccentricity', ‘EquivDiameter', 'EulerNumber', 'Extent', 'Extreme', FilledArea', 'FilledImage', 'Image', 'MajorAxisLength', 'MinorAxisLength', 'Orientation', 'PixelList', 'Solidity', 'all'
  - D: a structure with the fields specified when invoking regionprops( )

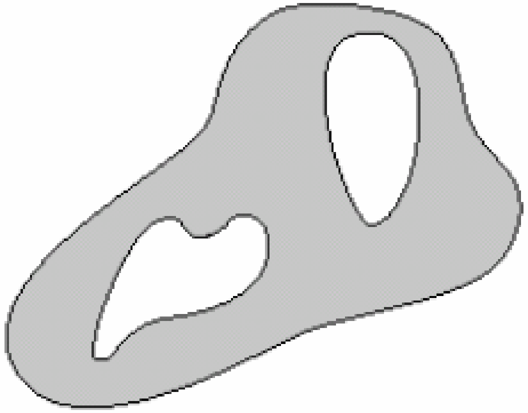
    *E.g., D.MajorAxisLength, D. MinorAxisLength, D.BoundingBox

Topological Descriptors

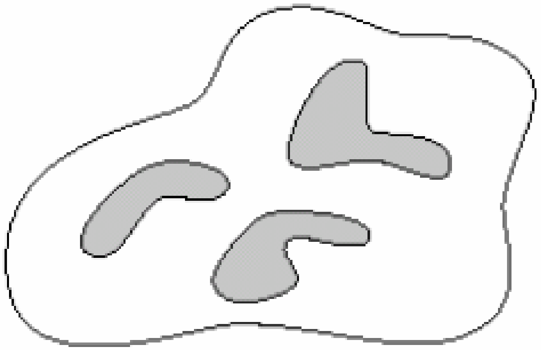
- If a topological descriptors is defined by the number of holes in the region, this property obviously will not be affected by a stretching or rotation transformation.
- Another topological property useful for region description is the number of connected components.

Example : Topological Descriptors

- Use to describe holes and connected components of the region



A region with two holes



A region with three connected components

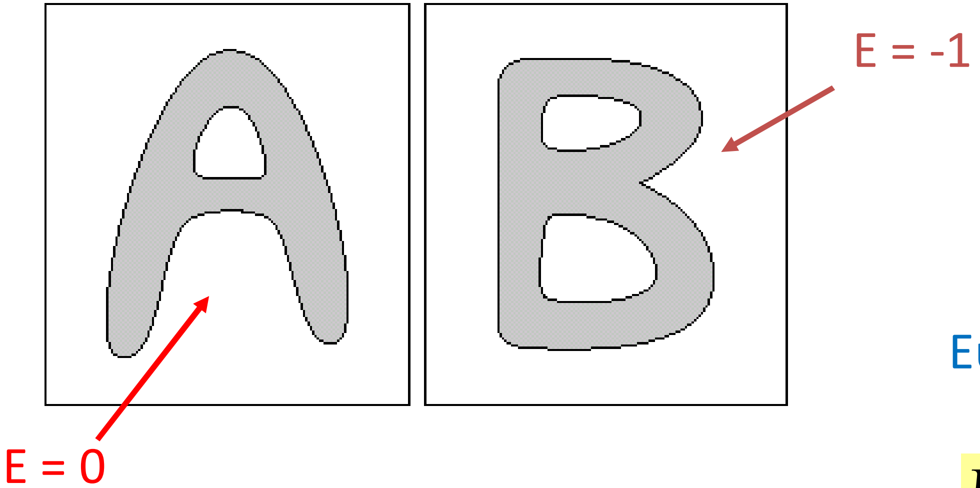
Euler number (E) :

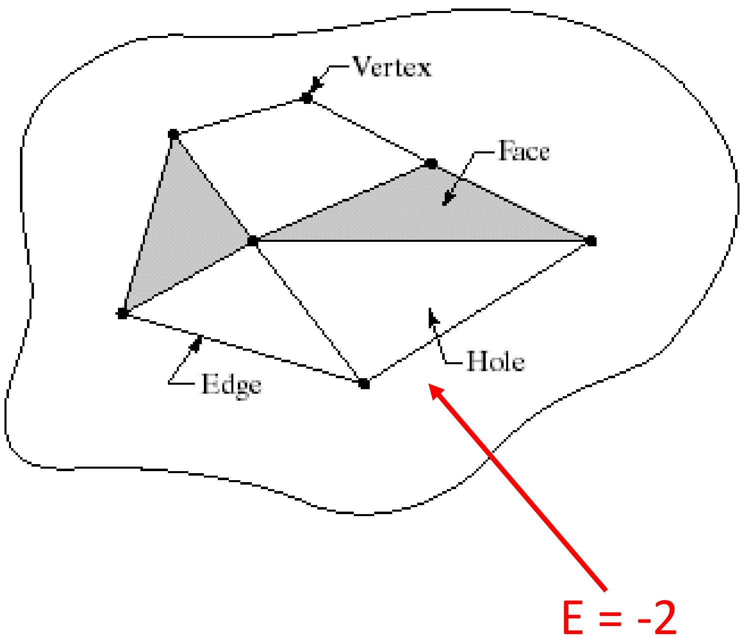
$$E = C - H$$

C = the number of connected components

H = the number of holes

Example : Topological Descriptors





Euler Formula :

$$V - Q + F = C - H = E$$

V = the number of vertices
Q = the number of edges
F = the number of faces

Example : Topological Descriptors

- MATLAB D = regionprops(L, properties)
  - L: labeled image
  - Properties: 'Area', 'BoundingBox', 'Centroid', 'ConvexArea', 'ConvexHull', 'ConvexImage', 'Eccentricity', EquivDiameter', 'EulerNumber', 'Extent', 'Extreme', FilledArea', 'FilledImage', 'Image', 'MajorAxisLength', 'MinorAxisLength', 'Orientation', 'PixelList', 'Solidity', 'all'
  - D: a structure with the fields specified when invoking regionprops( )

    * E.g., D.Area

    * E.g., D.EulerNumber

Thanks for your attention