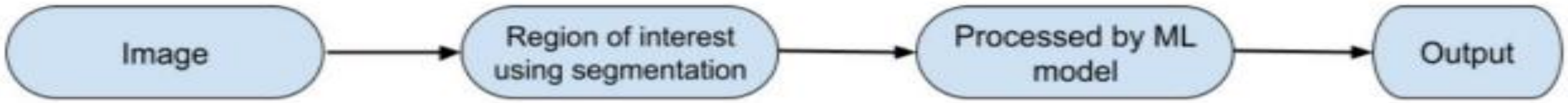
Unit 4

Image segmentation

Introduction

- Image segmentation is a method in which a digital image is broken down into various subgroups called Image segments which helps in reducing the complexity of the image to make further processing or analysis of the image simpler.
- Segmentation refers to the process of partitioning on image into multiple regions. It is typically used to locate objects and boundaries in images.
- Segmentation in easy words is assigning labels to pixels. All picture elements or pixels belonging to the same category have a common label assigned to them.
- For example: Let’s take a problem where the picture has to be provided as input for object detection. Rather than processing the whole image, the detector can be inputted with a region selected by a segmentation algorithm. This will prevent the detector from processing the whole image thereby reducing inference time.



Similarity and discontinuity approach

Similarity approach: This approach is based on detecting similarity between image pixels to form a segment, based on a threshold. ML algorithms like clustering are based on this type of approach to segment an image.

Discontinuity approach: This approach relies on the discontinuity of pixel intensity values of the image. Line, Point, and Edge Detection techniques use this type of approach for obtaining intermediate segmentation results which can be later processed to obtain the final segmented image.

Discontinuity Based Technique:

In discontinuity-based approach, the partitions or sub-division of an image is based on some abrupt changes in the intensity level of images. Here, we mainly interest in identification of isolated points, lined and edges in an image. To identify these, we use 3X3 Mask operation.

The discontinuity-based segmentation can be classified into three approaches:

- Point detection
- Line detection
- Edge detection

Point Detection:

A point is the most basic type of discontinuity in a digital image. The most common approach to finding discontinuities is to run an (nxn) mask over each point in the image.

The Mask is as shown below:

-1	-1	-1
-1	8	-1
-1	-1	-1

g(x,y) = f(x,y)*Mask

The point is detected at a location (x, y) in an image, where the convolution operation is done using this mask. If the absolute corresponding value of **Z** is greater than threshold value **T**, put label **1** for that point and put level **0** for others.

$$g(x, y) = \begin{cases} 1, & if\ |Z(x, y)| > T \\ 0, & Otherwise \end{cases}$$

Where Z is the response of the mask at any point in the image and T is non-negative threshold value.

For Example: Detect a point in a given image. Use threshold T = 8

1	2	3
4	5	6
7	8	9

Solution:

1	2	3
4	5	6
7	8	9

*

-1	-1	-1
-1	8	-1
-1	-1	-1

= -40+40

= 0

1	2	3
4	0	6
7	8	9

Use T=8

0	0	0
0	0	0
0	0	1

The point is detected at position where intensity is 1.

Line Detection

Line detection is the next level of complexity in the direction of image discontinuity. For any point in the image, a response can be calculated that will show which direction the point of a line is most associated with. The mask for different direction is given bellow

-1	-1	-1
2	2	2
-1	-1	-1

Horizontal direction

-1	2	-1
-1	2	-1
-1	2	-1

Vertical direction

-1	-1	2
-1	2	-1
2	-1	-1

45^0 direction

2	-1	-1
-1	2	-1
-1	-1	2

-45^0 direction

Perform convolution operation in given image using these masks.

Edge detection

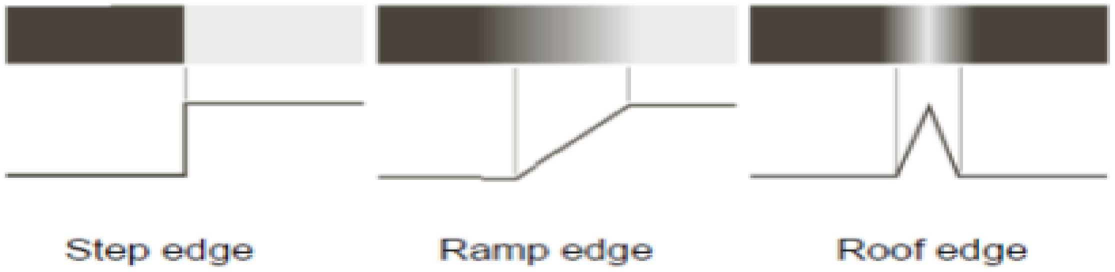
Since isolated points and lines of unitary pixel thickness are infrequent in most practical application, edge detection is the most common approach in gray level discontinuity segmentation. An edge is a boundary between two regions having distinct intensity level. It is very useful in detecting of discontinuity in an image, when the image changes from dark to white or vice-versa.

Three different edge types are observed:

Step edge: Transition of intensity level over 1 pixel only in ideal, or few pixels on a more practical use

Ramp edge: A slow and graduate transition

Roof edge: A transition to a different intensity and back



The first order derivative or gradient based filter such as ***Robert-cross, Prewitt***, and ***Sobel*** operators are prefers for detecting ***thicker lines***.

The second order derivative such as ***Laplacian*** is prefers for detecting ***thinner lines.***

Robert (Cress Gradient) operator

This operator find the gradient difference in cross or diagonal pixel position.

The filter mask of Robert operator is:

g(x) =

-1	0
0	1

g (y) =

0	-1
1	0

By using one of this filter mask, we can perform convolution operation on input image to calculate g(x) and g(y).

Prewitt Operator

This method takes the central difference of the neighboring pixels.

The filter mask of Prewitt Operator is:

g(x) =

-1	-1	-1
0	0	0
1	1	1

g(y) =

-1	0	1
-1	0	1
-1	0	1

By using one of this filter mask, we can perform convolution operation on input image to calculate g(x) and g(y).

Sobel Operator

This method also takes the central difference of the neighboring pixels. It provides both a differentiating and a smoothing effect.

The filter mask of Sobel Operator is:

g(x) =

-1	-2	-1
0	0	0
1	2	1

g(y) =

-1	0	1
-2	0	2
-1	0	1

By using one of this filter mask, we can perform convolution operation on input image to calculate g(x) and g(y).

Edge Linking and Boundary Detection

- **Edge Linking:** Ideally, edge detection should yield sets of pixels lying only on edges. In practice, these pixels rarely characterize edges completely because of non-uniform illumination, noise and breaks in the edges. Therefore, edge detection typically is followed by linking algorithms designed to assemble edge pixels into meaningful edges and/or region boundaries.
- Edge linking may be:

Local:

- requiring knowledge of edge points in a small neighborhood.

Regional:

- requiring knowledge of edge points on the boundary of a region.

Global:

- the Hough transform, involving the entire edge image.
- Edge Linking by Local Processing
- All points that are similar according to predefined criteria are linked, forming an edge of pixels that share common properties.

- Often, the location of regions of interest is known and pixel membership to regions is available. Approximation of the region boundary by fitting a polygon. Polygons are attractive because: They capture the essential shape They keep the representation simple

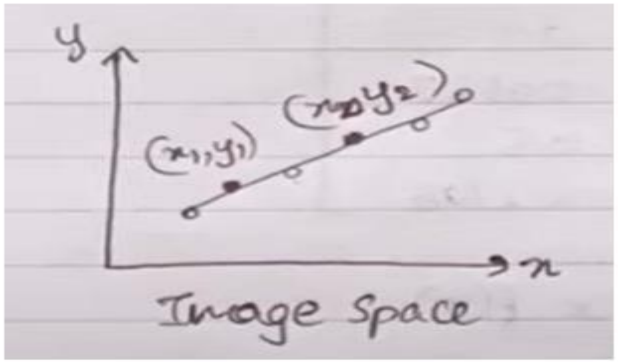
Edge Linking by Global Processing

Hough Transform The Hough Transform is an algorithm patented by Paul V. C. Hough and was originally invented to recognize complex lines in photographs (Hough, 1962). Since its inception, the algorithm has been modified and enhanced to be able to recognize other shapes such as circles and quadrilaterals of specific types. It is mainly used to connect disjoint edge points

Equation of line is

$y = mx + c$

Where, m =slope and c = intercept of the line



A single point can be a part of infinite line. Therefore we transform that point in the x-y plane, into a line in the m-c plane.

Example:

Q1. Using Hough transform, show that the following points are collinear. Also find the equation of the line: (1,2), (2,3) and (3,4).

Ans Equation of the line:

$y = mx + c$

In order to perform Hough transform, we need to convert the line from (x,y) plane to (m,c) plane.

$\therefore$ $\boxed{c = -mx + y}$

(i) For $(x,y) = (1,2)$, $c = -m + 2$

if $c = 0$, $0 = -m + 2$

$\boxed{m = 2}$

if $m = 0$, $\boxed{c = 2}$

Thus, $(m,c) = (2,2)$.

(ii) For $(x,y) = (2,3)$, $c = -2m + 3$.

if $c = 0$, $0 = -2m + 3$

$2m = 3$

$m = 3/2 = 1.5$. $\boxed{m = 1.5}$

if $m = 0$, $\boxed{c = 3}$.

Thus, $(m,c) = (1.5, 3)$.

(iii) For $(x,y) = (3,4)$, $c = -3m + 4$.

if $c = 0$, $0 = -3m + 4$

$3m = 4$

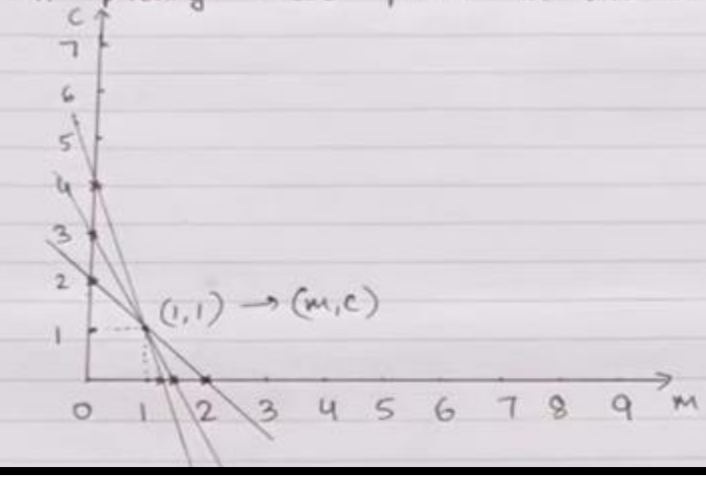
$m = 4/3 = 1.33$. $\boxed{m = 1.33}$

if $m = 0$, $\boxed{c = 4}$.

Thus $(m,c) = (1.33, 4)$.

$(m,c) = (2,2), (1.5,3), (1.33,4)$.

On plotting these points in the m-c plane:



All three lines are intercept in single point (1,1) that means all points (1,2) (2,3) (3,4) are collinear and they are part of the same line.

Original equation:

$y = mx + c$

Substituting (1,1):

$\boxed{y = x + 1}$ → Final equation

Edge Linking by Regional Processing

Often, the location of regions of interest is known and pixel membership to regions is available. Approximation of the region boundary by fitting a polygon.

Polygons are attractive because:

- They capture the essential shape
- They keep the representation simple

Requirements

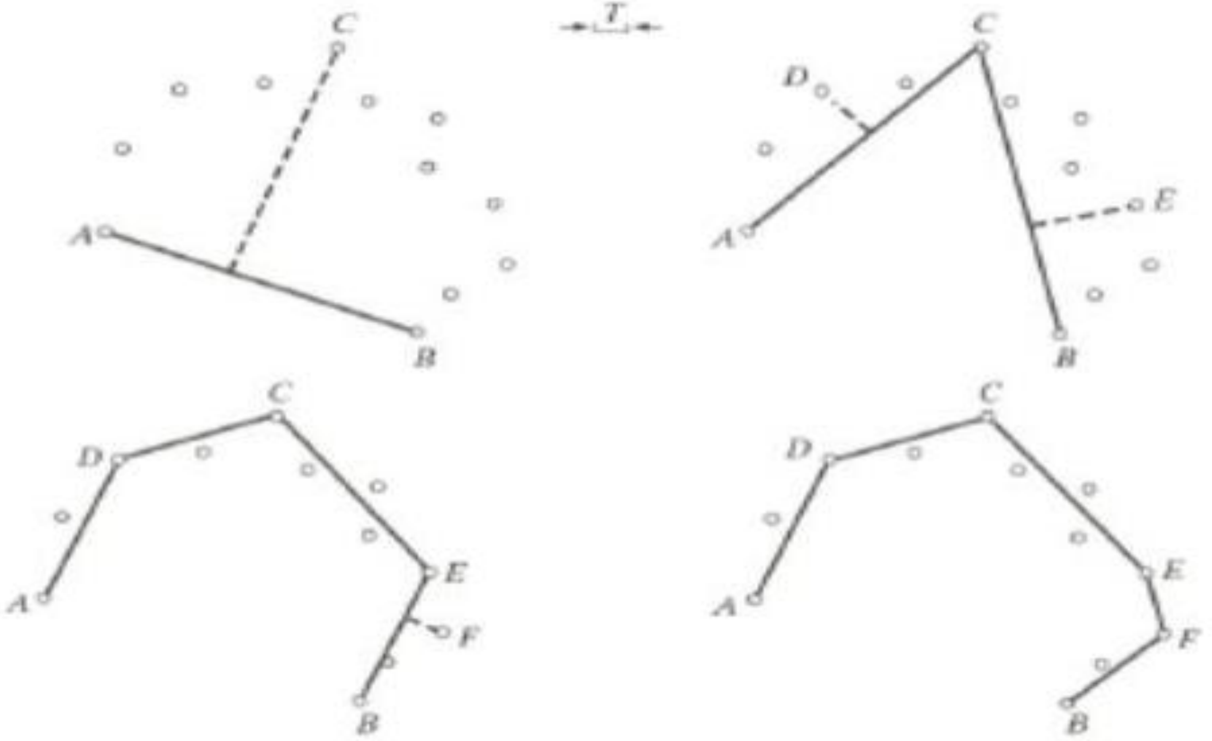
- Two starting points must be specified (e.g. rightmost and leftmost points).
- The points must be ordered (e.g. clockwise).
- Variations of the algorithm handle both open and closed curves.

If this is not provided, it may be determined by distance criteria:

- Uniform separation between points indicate a closed curve
- A relatively large distance between consecutive points with respect to the distances between other points indicate an open curve

We present here the basic mechanism for polygon fitting.

Given the end points A and B, compute the straight line AB. Compute the perpendicular distance from all other points to this line. If this distance exceeds a threshold, the corresponding point C having the maximum distance from AB is declared a vertex. Compute lines AC and CB and continue.



Thresholding

The simplest method for segmentation in image processing is the threshold method. It divides the pixels in an image by comparing the pixel's intensity with a specified value (threshold). It is useful when the required object has a higher intensity than the background (unnecessary parts).

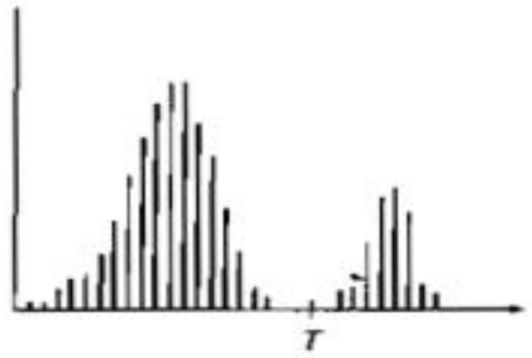
You can consider the threshold value (T) to be a constant but it would only work if the image has very little noise (unnecessary information and data). You can keep the threshold value constant or dynamic according to your requirements.

The thresholding method converts a grey-scale image into a binary image by dividing it into two segments (required and not required sections).

Steps to apply threshold

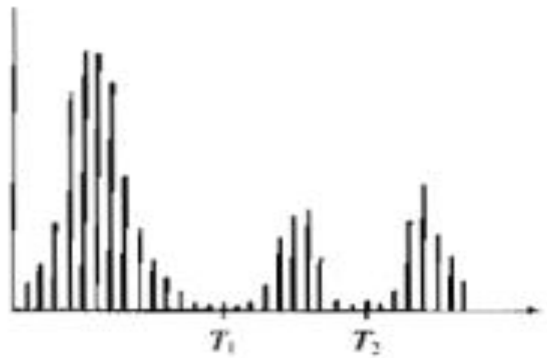
1. Select a threshold value ***T***
2. Any point ***(x,y)*** in the image at which ***f(x,y) > T***, this point is called an object point and ***f(x,y) <= T*** is called background point.
3. The segmented image ***g(x,y)*** is denoted by

$$g(x,y) = \begin{cases} 1, & if\ f(x,y) > T \\ 0, & if\ (x,y) \leq T \end{cases}$$



Here in histogram, the region greater than T is object and less than T is background.

The histogram bellow shows the thresholding problem involving three dominant modes. For example two objects on a dark background.



Here in histogram multiple thresholding classifies a point ***(x,y)*** as belonging to the background if ***f(x,y) <= T₁*** , one object classes if ***T₁ < f(x,y) <= T₂***, and to the other object classes ***f(x,y) > T₂***

$$g(x,y) = \begin{cases} a, & if\ f(x,y) > T_2 \\ b, & if\ T_1 < f(x,y) \leq T_2 \\ c, & if\ f(x,y) \leq T_1 \end{cases}$$

Global Thresholding:

A constant threshold value is apply for both object and background is called global thresholding. In this method, you replace the image's pixels with either white or black.

If the intensity of a pixel at a particular position is less than the threshold value, you'd replace it with black. On the other hand, if it's higher than the threshold, you'd replace it with white.

Procedure for Global thresholding

1. Select initial threshold value T (*choose average value of intensity*)
2. Segment the image using T. this will produce two groups G1 and G2 which are G1 contains values > T and G2 contains values <= T
3. Compute the average gray level values $\mu 1$ and $\mu 2$ for the pixel in region G1and G2
4. Compute the new threshold value

   $$T=\frac{1}{2}(\mu 1+\mu 2)$$

5. Repeat step 2 to step 4 until the T in successive iterations is same.

Example: Find the Global Threshold value of given image

5	3	9
2	1	7
8	4	2

Solution:

Calculate the threshold value T_0 by taking average of all the pixel value

$T_0 = (5+3+9+2+1+7+8+4+2)/9$

$T_0 = 4.55 = 5$

Segment the image using T = 5, we get

G1 = {9,7,8}

Calculate the mean value of G1

$\mu 1 = (9+7+8)/3$

$\mu 1 = 8$

G2 = {5,3,2,1,4,2}

Calculate the mean value of G2

$\mu 2 = \frac{5+3+2+1+4+2}{6}$

$= 2.83$

$\mu 2 = 3$

Now, the new value of T say T_1

$T_1 = \frac{1}{2}(\mu 1 + \mu 2)$

$T_1 = \frac{1}{2}(8+3)$

$T_1 = 5.5 = 6$

Here the threshold value for successive iteration is different,

So, again segment the image using new threshild value of T_1 i.e. 6,

G1 = {9,7,8}

$\mu 1 = 8$

G2 = {5,3,2,1,4,2}

$\mu 2 = 3$

Now, the new value of T say T_2

$T_2 = \frac{1}{2}(8+3)$

$T_2 = 5.5 = 6$

Here the threshold value for successive iteration is same, so final value of T is 6, which is the global threshold value.

Local or Regional Thrisholding

When the value of T changes over an image is called Variable thresholding. The term local or Regional thresholding is used sometimes to denote variable thresholding in which the value of T at any point (x,y) in an image depends on properties of a neighborhood of (x,y).

For example: The average intensity of the pixels in the neighborhood.

Adaptive Thresholding

If T depends on special coordinates (x,y) themselves, then variable thresholding is called dynamic or adaptive thresholding.

Having one constant threshold value might not be a suitable approach to take with every image. Different images have different backgrounds and conditions which affect their properties.

Thus, instead of using one constant threshold value for performing segmentation on the entire image, you can keep the threshold value variable. In this technique, you'll keep different threshold values for different sections of an image.

This method works well with images that have varying lighting conditions. You'll need to use an algorithm that segments the image into smaller sections and calculates the threshold value for each of them.

Procedure for Adaptive thresholding

1. Divide original image in to different regions
2. Apply global thrsholding method in each region separately
3. Merge the resulted region based image

Region-Based Segmentation

Region-based segmentation algorithms divide the image into sections with similar features. These regions are only a group of pixels and the algorithm find these groups by first locating a seed point which could be a small section or a large portion of the input image.

After finding the seed points, a region-based segmentation algorithm would either add more pixels to them or shrink them, so it can merge them with other seed points.

Based on these two methods, we can classify region-based segmentation into the following categories:

Region Growing

In this method, you start with a small set of pixels and then start iteratively merging more pixels according to particular similarity conditions. A region growing algorithm would pick an arbitrary seed pixel in the image, compare it with the neighbor pixels and start increasing the region by finding matches to the seed point.

When a particular region can't grow further, the algorithm will pick another seed pixel which might not belong to any existing region. One region can have too many attributes causing it to take over most of the image. To avoid such an error, region growing algorithms grow multiple regions at the same time.

You should use region growing algorithms for images that have a lot of noise as the noise would make it difficult to find edges or use thresholding algorithms.

Algorithm for Region Growing

1. Choose a seed point
2. Check the condition

    If |seed point – pixel value| <=T

        Add pixel to seed point region

    Else

        Leave as it is

3. Repeat step 2 for all pixel

Example: Apply region growing on following image with seed point at (2, 2) and threshold value as 2.

0	1	2	0
2	5	6	1
1	4	7	8
0	9	5	1

Solution:

Here the seed point is 7 which is at location (2,2)

0	1	2	0
2	5	6	1
1	4	7	8
0	9	5	1

Threshold value (T) =2

Therefore the condition is

|seed point – pixel value| <=2

The possible pixel values which satisfy the condition are {5,6,7,8,9}

Now, let's say region A which is denoted by 1 for condition satisfied values and region B which is denoted by 0 for other values

0	0	0	0
0	1	1	0
0	0	1	1
0	1	1	0

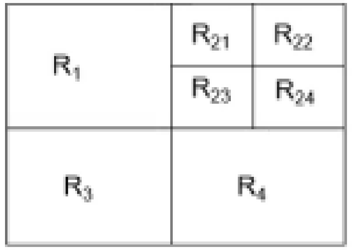
Region Splitting and Merging

As the name suggests, a region splitting and merging focused method would perform two actions together – splitting and merging portions of the image.

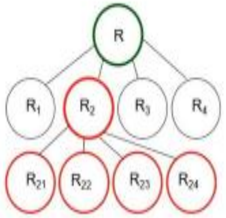
This method is also called divide and conquer.

It would first split the image into regions that have similar attributes and merge the adjacent portions which are similar to one another. In region splitting, the algorithm considers the entire image, while in region growth, the algorithm would focus on a particular point.

The region splitting and merging method follows a divide and conquer methodology. It divides the image into different portions and then matches them according to its predetermined conditions. Another name for the algorithms that perform this task is split-merge algorithms.



Quad tree of splitting



Algorithm for Region Splitting

1. Select max and min intensity value from given image
2. Check the condition

   if ((max – min) > T)

   split the image in to equal 4 regions

   else

   leave as it is
3. Apply step 2 in all regions

Algorithm for Region Merging

1. Select max and min intensity value from neighbor regions
2. Check the condition

   if $((max_1 - min_2) <= T$ && $(max_2 - min_1) <= T)$

   merge those regions

   else

   leave as it is
3. Apply step 2 in all neighbor regions

Example: Apply split and merge in given image. The threshold value is 3.

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

Solution:

Here, given T = 3

Max pixel value = 7

Min pixel value = 0

Now, check the condition (max – min) > T

(7 – 0) > 3

Then split the image in to 4 regions

	6	5	6	6	7	7	6	6	
A	6	7	6	7	5	5	4	7	B
	6	6	4	4	3	2	5	6	
	5	4	5	4	2	3	4	6	
	0	3	2	3	3	2	4	7	
C	0	0	0	0	2	2	5	6	D
	1	1	0	1	0	3	4	4	
	1	0	1	0	2	3	5	4	

Now, apply rule for all the regions A, B, C and D

	6	5	6	6	7	7	6	6	
A	6	7	6	7	5	5	4	7	B
	6	6	4	4	3	2	5	6	
	5	4	5	4	2	3	4	6	
	0	3	2	3	3	2	4	7	
C	0	0	0	0	2	2	5	6	D
	1	1	0	1	0	3	4	4	
	1	0	1	0	2	3	5	4	

Now, merging

Check the condition $((max_1 - min_2) <= T \ \&\& \ (max_2 - min_1) <= T)$

Here, for region A and B1

((7 – 5) <= 3 && (7 – 4) <= 3)

Both the condition is satisfied so merge these two regions

Apply same rule for all the regions

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

In the resulted image we can see two regions one is shaded region and another is un-shaded region.

Image representation

- The objective is to represent and describe the resulting aggregate of segmented pixels in a form suitable for further computer processing after segmenting an image into regions.
- Two choices for representing a region:
  - External characteristics: its boundary
  - Internal characteristics: the pixels comprising the region.

For example, a region may be represented by (a) its boundary with the boundary described by features such as its length, (b) the orientation of the straight line joining the extreme points, and (c) the number of concavities in the boundary.

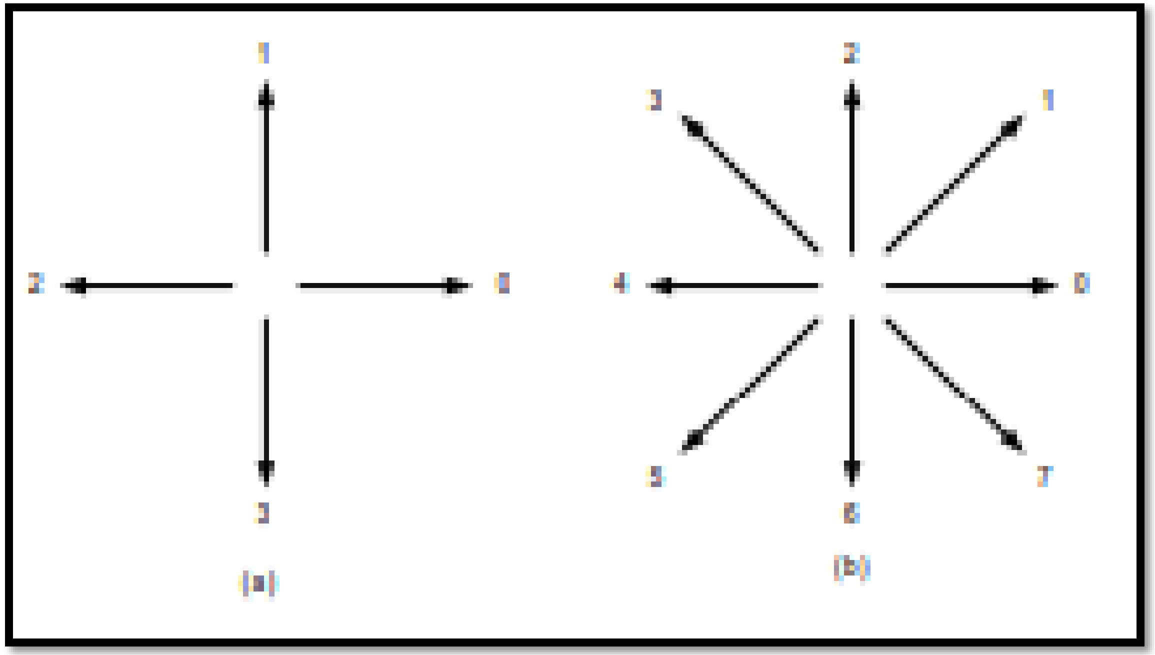
- An external representation is chosen when the primary focus is on shape characteristics.
- An internal representation is selected when the primary focus is on reflectivity properties, such as color and texture.
- In either case, the features selected as descriptors should be as insensitive as possible to variations such as change in size, translation and rotation.

Representation schemes

- The segmentation techniques yield raw data in the form of pixels along a boundary or pixels contained in a region.
- Although these data are sometimes used directly to obtain descriptors (as in determining the texture of a region), standard practice is to use schemes that compact the data into representations that are considerably more useful in the computation of descriptors.

Chain codes

- To represent a boundary by a connected sequence of straight line segments of specified length and direction. • The direction of each segment is coded by using a numbering scheme such as the ones shown below.



- This method generally is unacceptable to apply for the chain codes to pixels:

  (a) The resulting chain of codes usually is quite long;

  (b) Sensitive to noise: any small disturbances along the boundary owing to noise or imperfect segmentation cause changes in the code that may not necessarily be related to the shape of the boundary.

- A frequently used method to solve the problem is to resample the boundary by selecting a larger grid spacing.

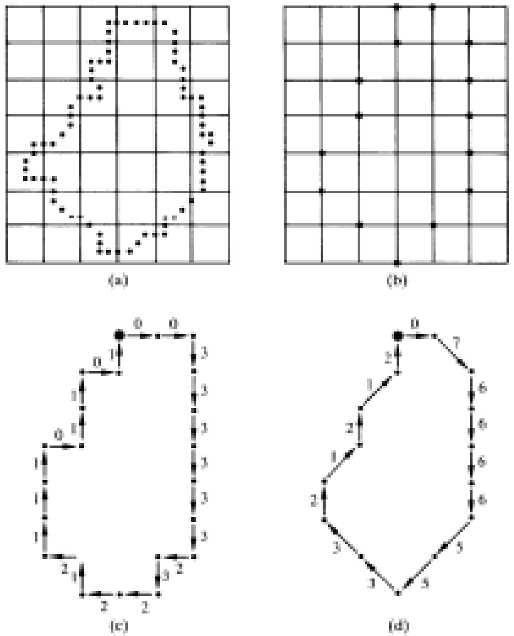


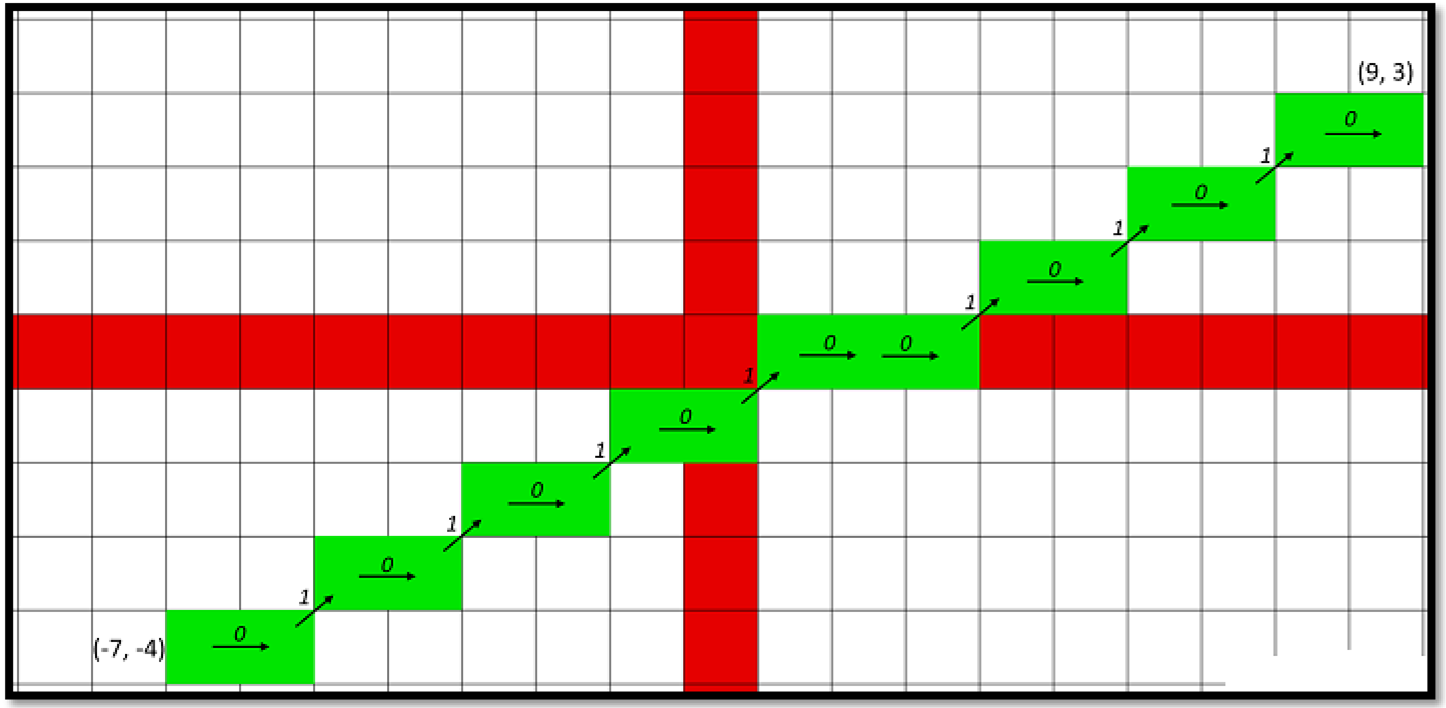
Fig 2. (a) Digital boundary with resampling grid; (b) result of resampling; (c)4-directional chain-code; (d) 8-directional chain code.

- Depending on the proximity of the original boundary to the nodes, a point is assigned to those nodes.
- The accuracy of the resulting code representation depends on the spacing of the sampling grid.

- *Normalization for starting point*:
  Treat the code as a circular sequence and redefine the starting point s that the resulting sequence of numbers forms an integer of minimum magnitude.
- *Normalization for rotation*:
  Use the first difference of the chain code instead of the code itself. The difference is simply by counting (counter-clockwise) the number of directions that separate two adjacent elements of the code.

  Example: The first difference of the 4-direction chain code 10103322 is 33133030.
- Normalization for size:
  Alter the size of the resampling grid.

Input : (-7, -4), (9, 3) Output : Chain code for the straight line from (-7, -4) to (9, 3) is 010101010010101 0

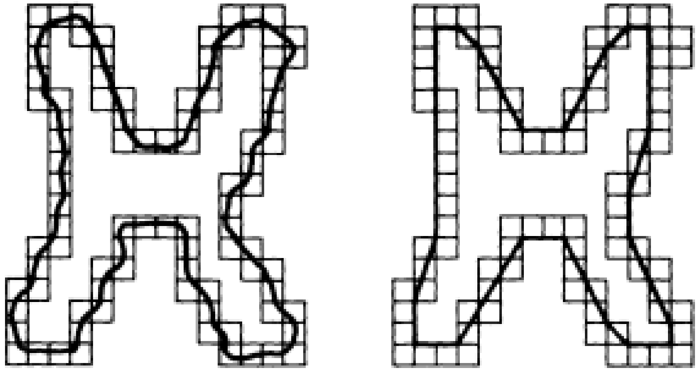


Polygonal approximation

- The objective is to capture the essence of the boundary shape with the fewest possible polygonal segments.
- This problem in general is not trivial and can quickly turn into a time-consuming iterative search.

(a) *Minimum-perimeter Polygons*

- A given boundary is enclosed by cells. We can visualize this enclosure as consisting of two walls corresponding to the inside and outside boundaries of the cells. If the boundary is a rubber band, it will shrink and take the shape as in (b).
- The error in each cell would be at most $\sqrt{2}d$, where d is the pixel distance.



(a) (b)

Fig 3. (a) Object boundary enclosed by cells (b) Minimum-perimeter polygon

(b) *Merging Technique*

- It is based on error or other criteria have been applied to the problem of polygonal approximation.
- One approach is to merge points along a boundary until the least square error line fit of the points merged so far exceeds a preset threshold.
- Vertices do not corresponding to corners in the boundary.

(c) *Splitting Techniques*

- To subdivide a segment successively into two parts until a given criterion is satisfied.
- Example: a line a drawn between two end points of a boundary. The perpendicular distance from the line to the boundary must not exceed a preset threshold. If it does, the farthest points becomes a vertex.

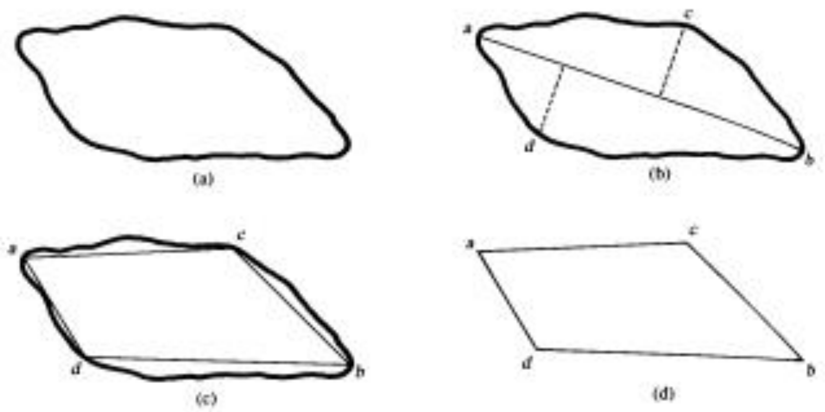
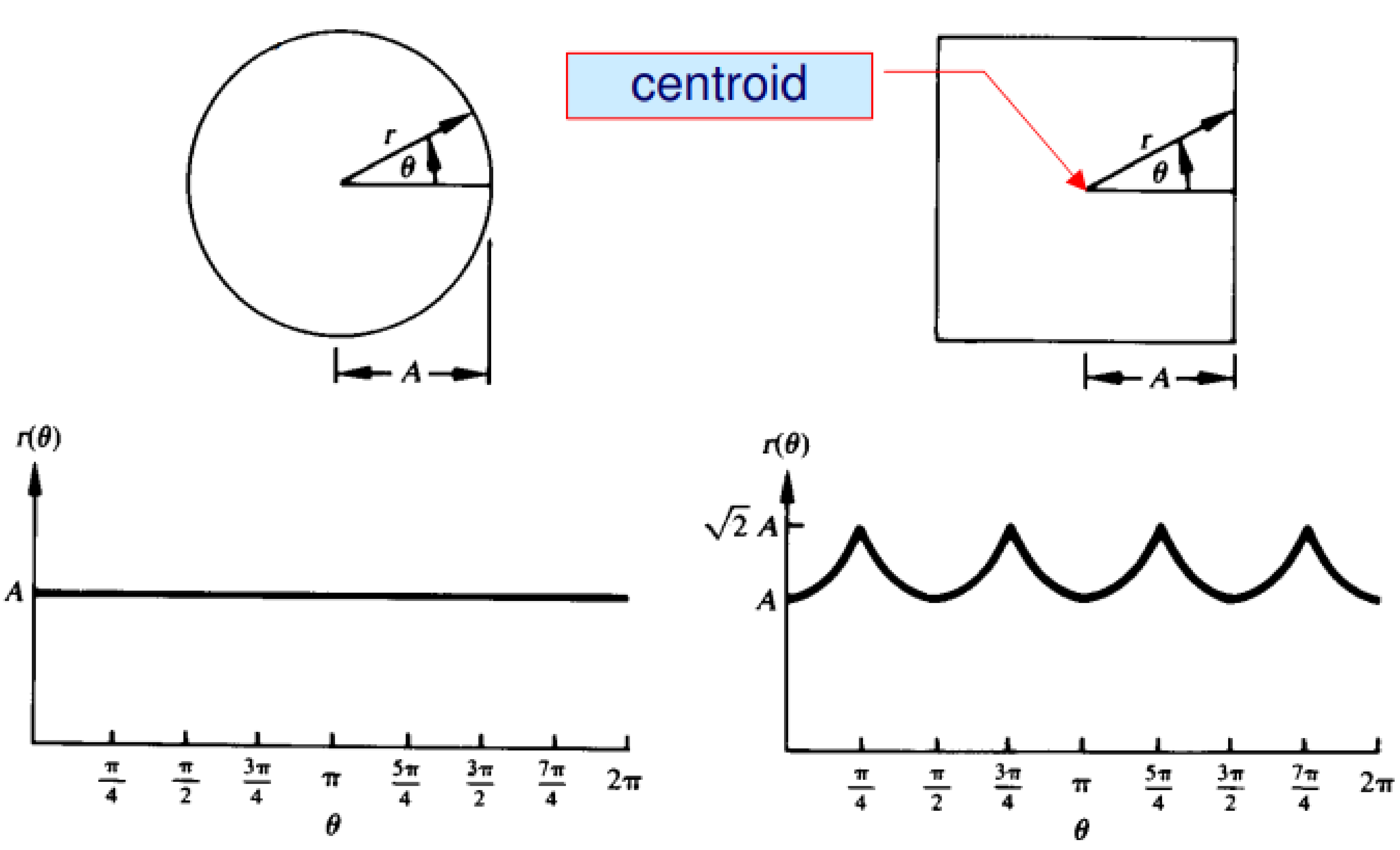


Fig 4. (a) Original boundary (b) boundary divided into segments based on distance computations (c) joining of vertices (d) resulting polygon

Signatures

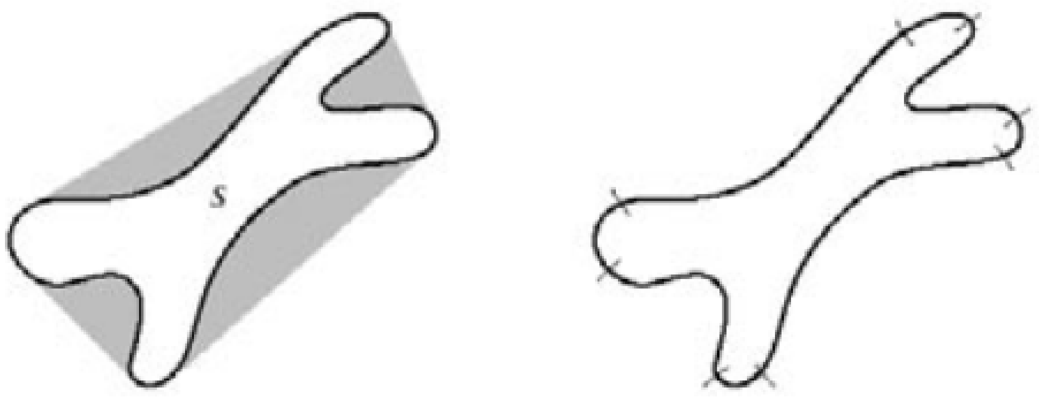
- A signature is a 1-D functional representation of a boundary. The simplest representation is to plot the distance from the centroid to the boundary as a function of angle.
- Making signatures invariant:
- dependence on rotation can be rejected by choosing the starting point as the point farthest from the centroid;
- normalization for scaling can be implemented by removing the mean value of the signature function.

Geometrical figures and their Signatures



Boundary segments

- **convex hull** technique
  - convex hull H of the arbitrary set S is the smallest convex set containing S; the set H-S is called convex deficiency D of the set S
  - 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



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

  - in practice, the boundary is usually smoothed prior to partitioning (for example, by using a polygonal approximation)

The skeleton of a region

- The structural shape of a plane region can be reduced to a graph.
- This reduction can be accomplished by obtaining the skeleton of the region via a thinning algorithm.

(a) *Medial axis transformation*

- The skeleton of a region may be defined via the medial axis transformatin (MAT) proposed by Blum in 1967.

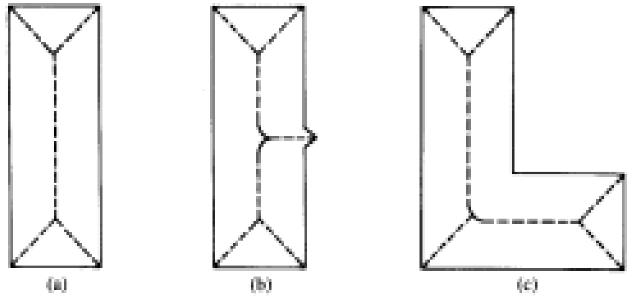


Fig 5. Medial axes of 3 simple regions

- Given a region R and a border B:

  For each point p in R, we find its closest neighbor in B. If p has more that one such neighbor, it belongs to the medial axis (skeleton of R).

- The concept of "closest" depends on the definition of a distance.
- Although the MAT of a region yields an intuitively pleasing skeleton, direct implementation of that definition is typically prohibitive computationally.
- Implementation potentially involves calculating the distance from every interior point to every point on the boundary of a region.

(b) *Thinning algorithm for binary regions*

- Assume region points have value 1 and background points 0.
- A contour point is any pixel with value 1 and having at least one 8-neighbor valued 0.

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

Fig 6. Neighborhood arrangement used by a thinning algorithm

- The thinning method consists of successive passes of two steps applied to the contour points.
  - ❑ Step 1

    Flag a contour point p for deletion if the following conditions are satisfied:

    (a) $2 \leq N(p_1) \leq 6$, where $N(p_1) = \sum_{i=2}^{9} p_i$

    (b) S(p_1)=1, where S(p_1) is the 0-1 transitions in the ordered sequence of $p_2, p_3, ..p_8, p_9$.

    (c) $p_2 \cdot p_4 \cdot p_6 = 0$

    (d) $p_4 \cdot p_6 \cdot p_8 = 0$

    To keep the structure during this step, points are not deleted until all border points have been processed

  - ❑ Step 2

    In the second step, condition (a) and (b) remain the same, and,

    (c') $p_2 \cdot p_4 \cdot p_8 = 0$

    (d') $p_2 \cdot p_6 \cdot p_8 = 0$

0	0	1
1	p	0
1	0	1

Fig 7. Illustration of conditions (a) & (b): M(p)=4 and S(P)=3

- The whole procedure for one iteration
  - ❑ Applying step 1 to flag border points for deletion
  - ❑ Deleting the flagged points
  - ❑ Applying step 2 to flag the remaining border points for deletion
  - ❑ Deleting the flagged points.
- The basic procedure is applied iteratively until no further points are deleted, at which time the algorithm terminates; yielding the skeleton of the region.

- Physical meaning of the conditions:
  - Condition (a) is violated when contour point p1 has only one or seven 8-neighbors valued 1, which implies that p1 is the end point of a skeleton strobe and should not be deleted.

    e.g. Deleting p in $\begin{matrix}0 & 0 & 0\\ 0 & p & 1\\ 0 & 0 & 0\end{matrix}$ shortens the skeleton.

  - Condition (b) is violated when it is applied to points on a stroke 1 pixel thick. Hence this condition prevents disconnection of segments of a skeleton during the thinning operation.

    e.g. Deleting p in $\begin{matrix}0 & 0 & 1\\ 1 & p & 0\\ 0 & 0 & 1\end{matrix}$ disconnects the skeleton.

  - Conditions (c) or (d) is violated when at least 3 of the 4-neighbors of p1 are connected to p1. In such cases, p1 is so critical that it can't be deleted.

    e.g. Deleting p in $\begin{matrix}0 & 1 & 0\\ 0 & p & 1\\ 0 & 1 & 0\end{matrix}$ disconnects the skeleton.

- A point that satisfied conditions (a)-(d) is an east or south boundary point or a northwest corner point in the boundary.
- Similarly, a point that satisfied conditions (a), (b), (c') and (d') is a north or west boundary points, or a southeast corner point in the boundary.
- In either case, p1 is not part of the skeleton and should be removed.

*

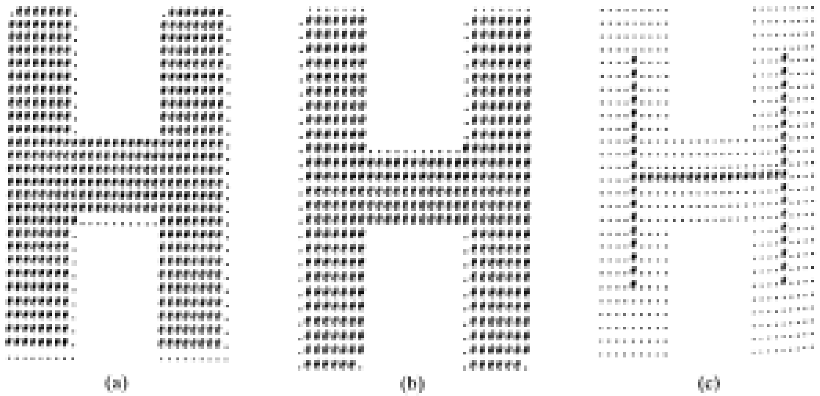
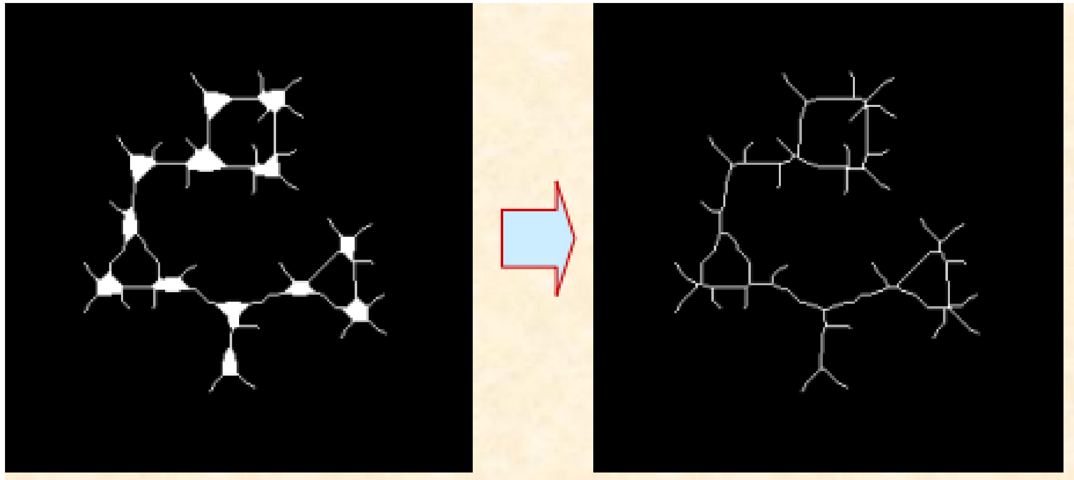
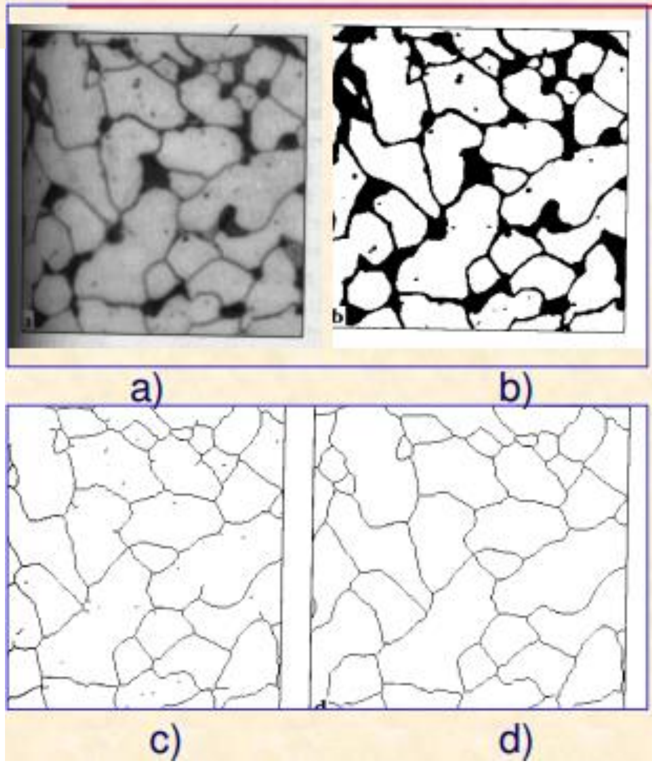


Fig 8. (a) Result of step 1 of the thinning algorithm during the first iteration through 1 region (b) result of step 2 and (c) final result.





Microscopic images of steel (with carbon particles):

a) source image

b) thresholded image

c) after skeletonizing

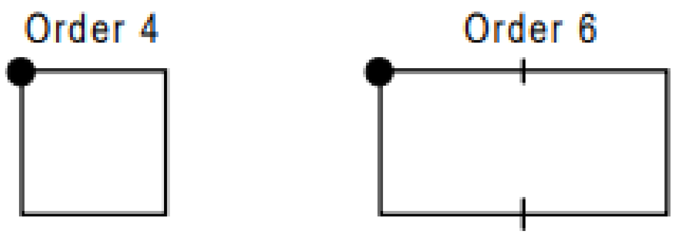
d) after pruning

BOUNDARY DESCRIPTORS

Length	- number of pixels along the contour
Xmax, Xmin, Ymax, Ymin	- maximum and minimum coordinates of the contour
Xcentroid, Ycentroid	- X and Y coordinates of the center of the region
Boundary diameter	- maximum distance between two points on the boundary
Curvature	- rate of change of slope (example: the difference between the slopes of adjacent boundary segments)
Shape order (chain count)	- number of elements in the chain
Shape number	- first difference of smallest magnitude

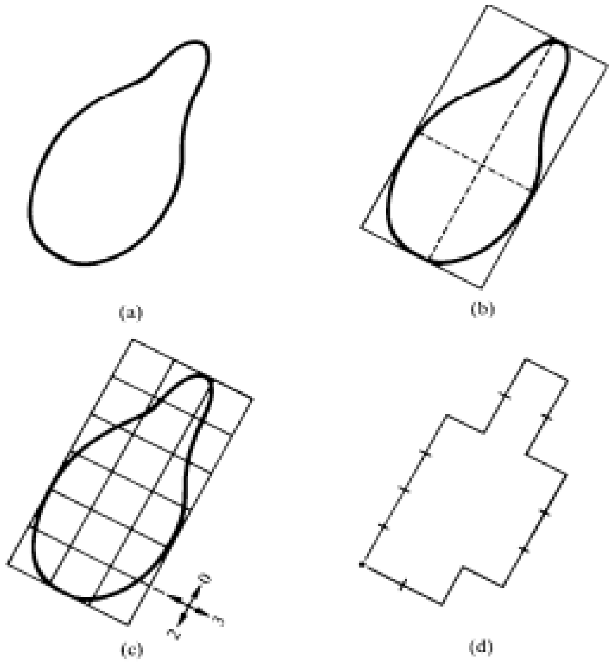
Shape numbers

- Shape number of a boundary is defined as the first difference of a chain code of the smallest magnitude.
- The order n of a shape number is the number of digits in its representation.
- The following figures shows all shapes of order 4 and 6 in a 4-directional chain code:



	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

Fig 9. *All shapes of order 4 and 6. The dot indicates the starting point.*



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

Difference: 3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

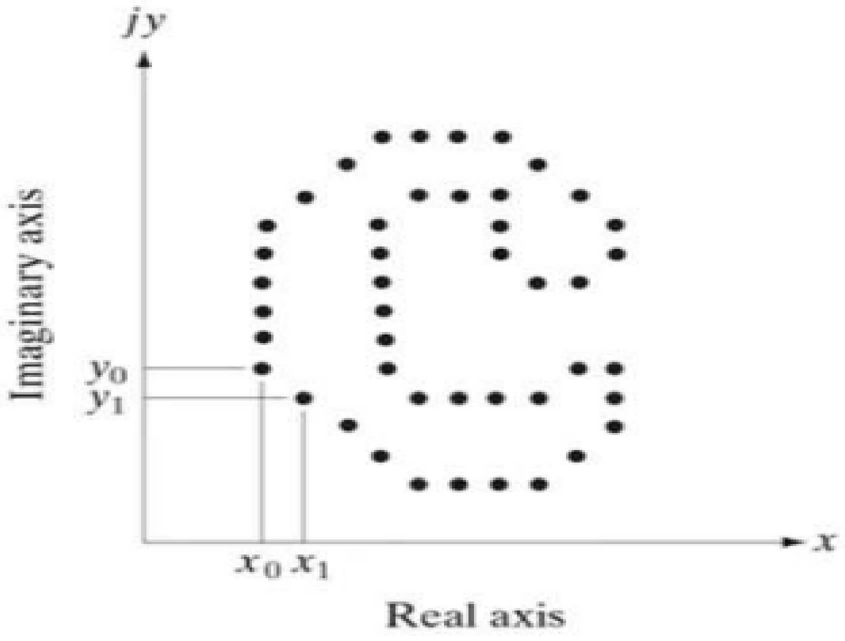
Shape no.: 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0 3

Fig 10. *Steps in the generation of a shape number*

- Note that the first difference is calculated by treating the chain codes as a circular sequence.

Fourier Descriptors

A common method of describing the contour (Outline) of an object is by using 1-Dimesnional Fourier Transform.



The figure shows a N-point digital boundary in the spatial Domain. Each of these edges pixels can be defined by its **x** and **y** coordinates. Starting at an arbitrary point (x0,y0),(x1,y1),(x2,y2)…….(x_{k-1},y_{k-1}) points are encountered as we move in the counter clockwise direction.

The coordinates can be expressed in the form $\mathbf{x(k) = x_k}$ and $\mathbf{y(k) = y_k}$ .With this notation, the boundary itself can be represented as the sequence of coordinates

$$\mathbf{S(k)= [x(k), y(k)]}$$

For, k=0,1, 2,..........,K-1

These coordinates values can be used to generate a complex function of the form,

$$\mathbf{S(k) = x(k) + jy(k)}$$

For, k=0,1, 2,..........,K-1

Hence the x-axis is treated as real axis and y-axis as the imaginary axis of the sequence of complex number.

The Fourier transform of this function s(k) yields the frequency components that describe the given edge.

The discrete Fourier transform (DFT) of s(k) is

$$a(u) = \sum_{k=0}^{K-1} s(k)e^{-j2\pi uk/K}$$

Where, u = 0,1,2………..K-1

The complex coefficients of a(u) from 0 to K-1 are called Fourier Descriptors of the boundaries.

The inverse Fourier Transform of a(u) restore the coefficients of s(k)

$$s(k) = \frac{1}{K}\sum_{u=0}^{K-1} a(u)e^{j2\pi uk/K}$$

Where, k = 0,1,2……….K-1

However instead of using all the Fourier coefficients, we only use few of them say first P in the range 0 to P-1, while remaining terms are made zero, a(u) = 0 if u > P-1

The result of the approximation of s(k) is

$$\hat{s}(\mathrm{k}) = \frac{1}{P}\sum_{u=0}^{P-1} a(u)e^{j2\pi uk/P}$$

Where, k = 0,1,2……….K-1

Although only P terms are used to obtain each component of $\hat{s}(k)$, k still ranges from 0 to K-1. That is the same number of points exits in the approximate boundary, but not as many terms are used in the reconstruction of each point.

Although only P terms are used for a(u), a(k) still has 0 to N-1 values. That is the same number of points exist in the new approximated boundary but not as many terms are used in the reconstruction of each point.

Following example shows the variation of boundary points of chromosomes which varies the shape of chromosome.

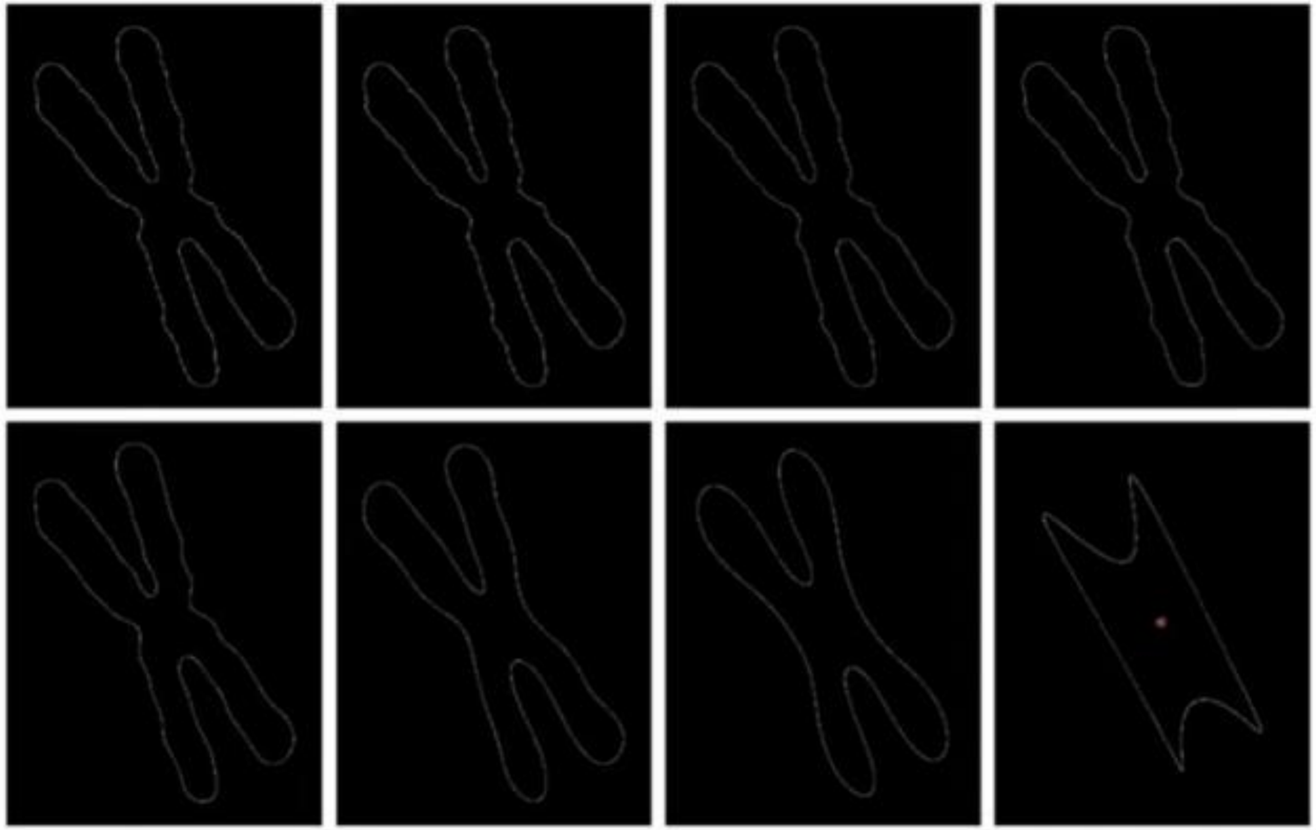


Fig: (a): Boundary of human chromosome (2868) points (b-h): Boundary reconstructed using 1434, 286, 144, 72, 36, 18, and 8 Fourier descriptors respectively. These numbers are approximately 50%, 10%, 5%, 2.5%, 1.25%, 0.63%, and 0.28% of 2868 points respectively.

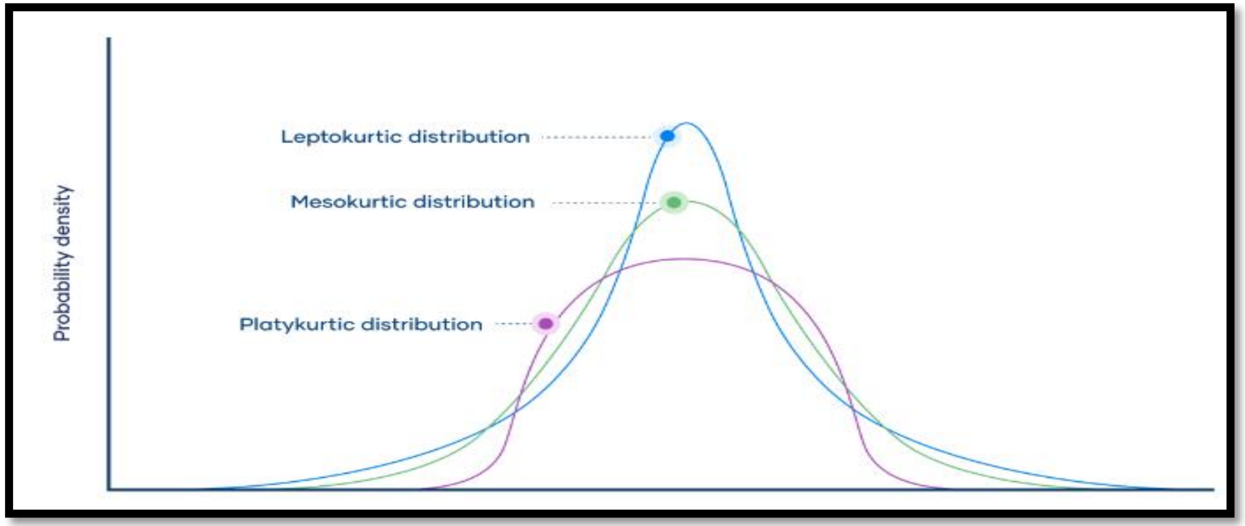
REGIONAL DESCRIPTORS

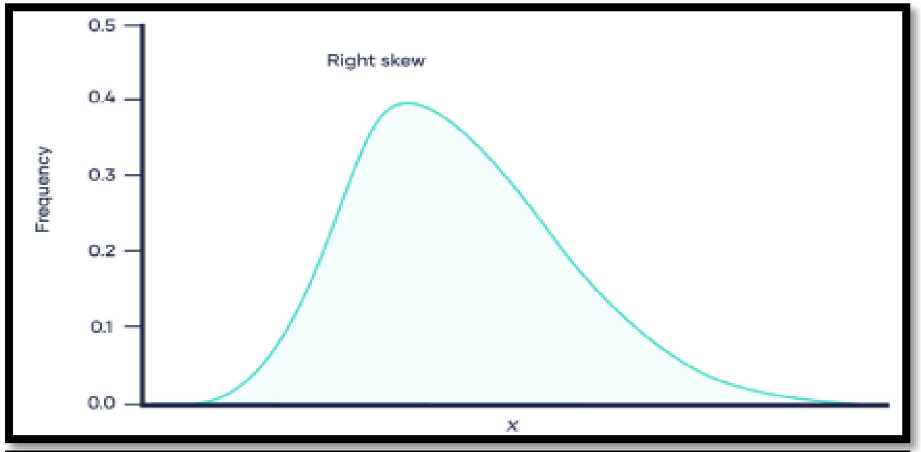
Skewness - the third moment about the mean; an indicator of asymmetry in a pixel distribution

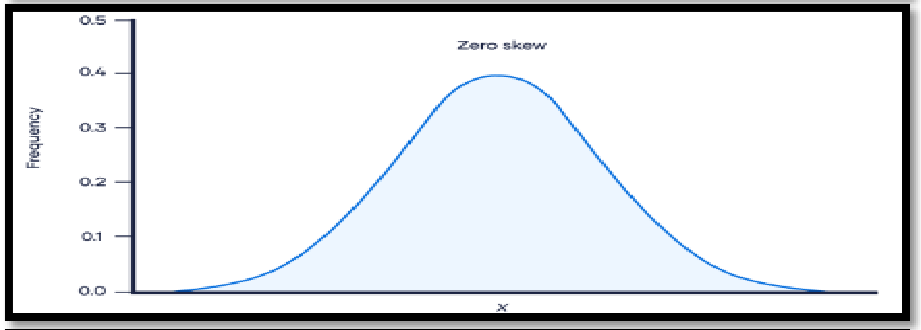
$$sk = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^3}{(N-1)\sigma^3}$$

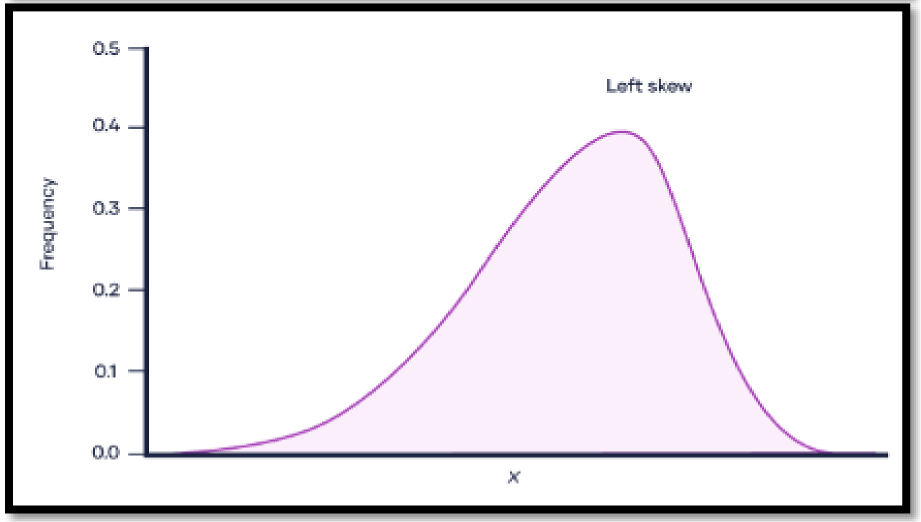
Kurtosis - the fourth moment about the mean; relates to the degree of peakedness or flatness of a distribution (for the normal distribution kurtosis =3)

$$kurt = \frac{\sum_{i=1}^{N}(x_i - \overline{x})^4}{(N-1)\sigma^4}$$









Perimeter	- length of region's boundary
Area	- number of pixels contained within its boundary
Shape factor	- Perimeter^2/Area
Range	- the difference between the largest and the smallest pixel values in a region
Median	- median of pixels' intensity values within the region

Mean - average of pixels' intensity values within the region

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} = \bar{x}$$

Variance - the second moment about the mean

$$\sigma^2 = \frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N-1}$$

Standard deviation - square root of a variance

$$SD = \sqrt{\sigma^2}$$

Coef. of variation - a ratio of standard deviation to the mean

$$C = \frac{\sigma}{\mu}$$

Topological descriptors

- topology - the study of properties of a figure that are unaffected by any deformation except tearing or folding

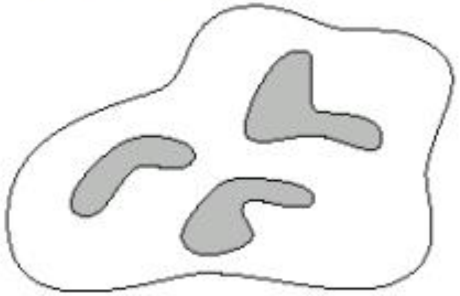


FIGURE A region with three connected components.

FIGURE A region with two holes.

- topological features give a **global** description of a region

  examples:

  number of holes

  number of connected components

  number of object edges

  number of faces

  number of vertices

  Euler number (the difference between the number of connected components and number of holes)

a b

FIGURE Regions with Euler number equal to 0 and −1, respectively.