Representation and Description

Presented by

Group - 7

19CSE 004- Md. Mainul Islam Munna
19CSE019 - Sanjida Haque Ria
19CSE039 - Md Najmus Saadat
19CSE041- Jarin Tasnim
19CSE043 - Moheuddin Ahmed Mubin

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”

Representation and Description(cont.)

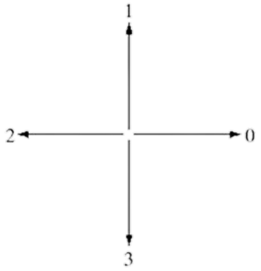
What kind of information we can use?

- Boundary, shape
- Region
- Texture
- Relation between regions

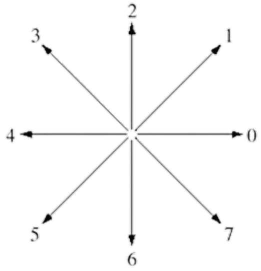
Chain Code

- Chain code is a method for representing object boundary in digital images.
- provides a compact and efficient way to describe shapes.
- Start from a point on the boundary of an object.
- Encode the direction of each successive boundary pixel relative to the current position.

➔ 4- Directional Chain Code
➔ 8- Directional Chain Code

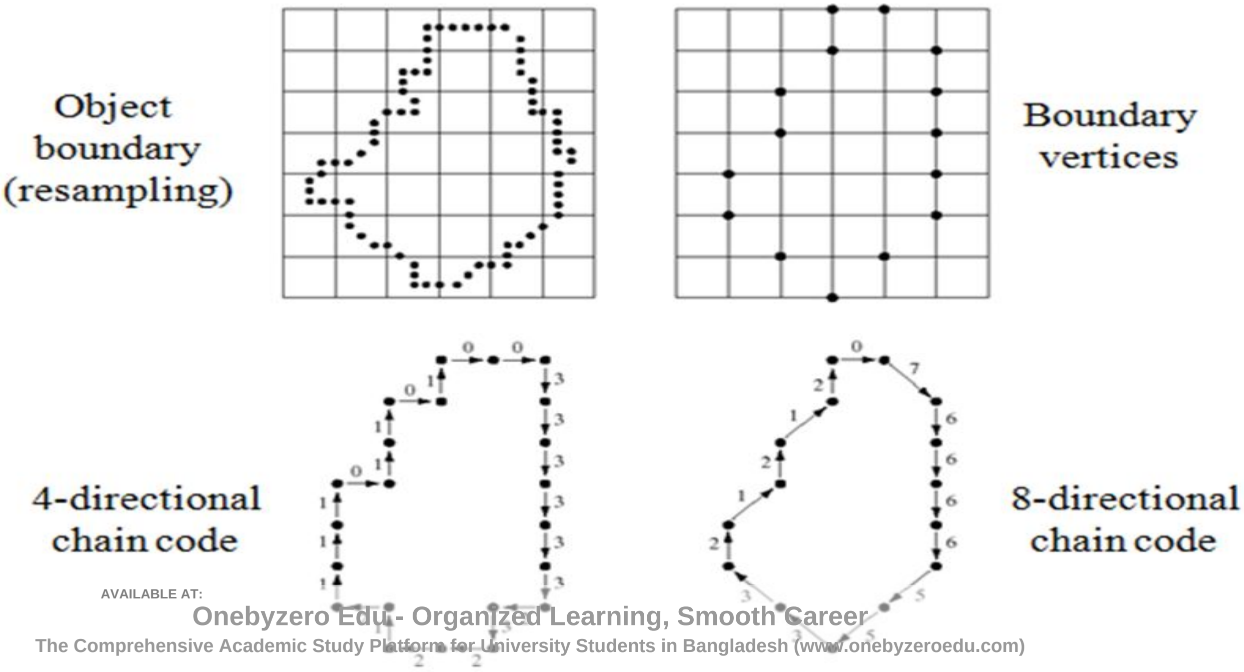


4-directional chain code



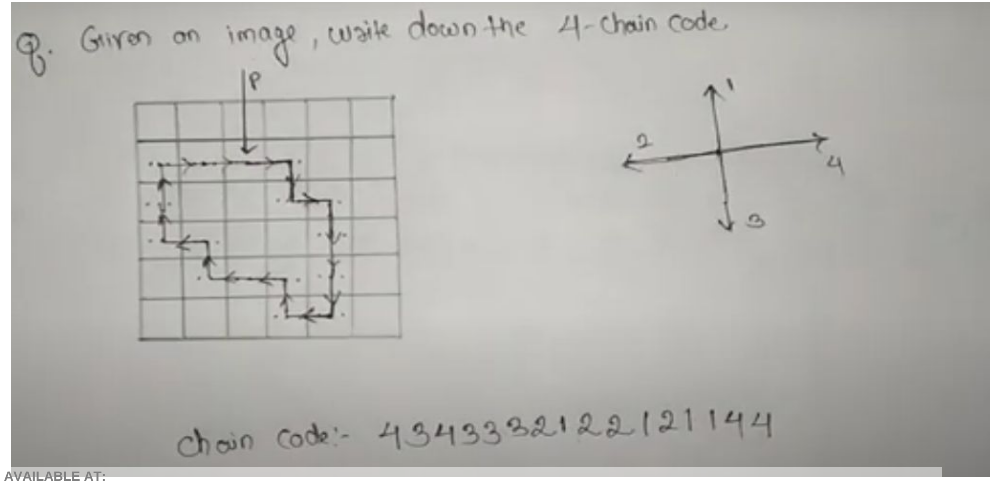
8-directional chain code

Chain Code(Cont..)



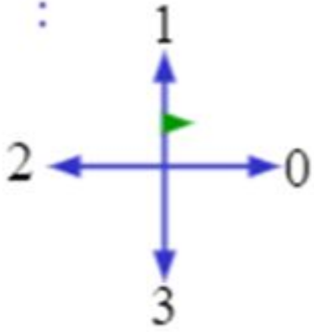
Chain Code(Cont..)

4- Directions chain code



The first difference of a chain code

Example :



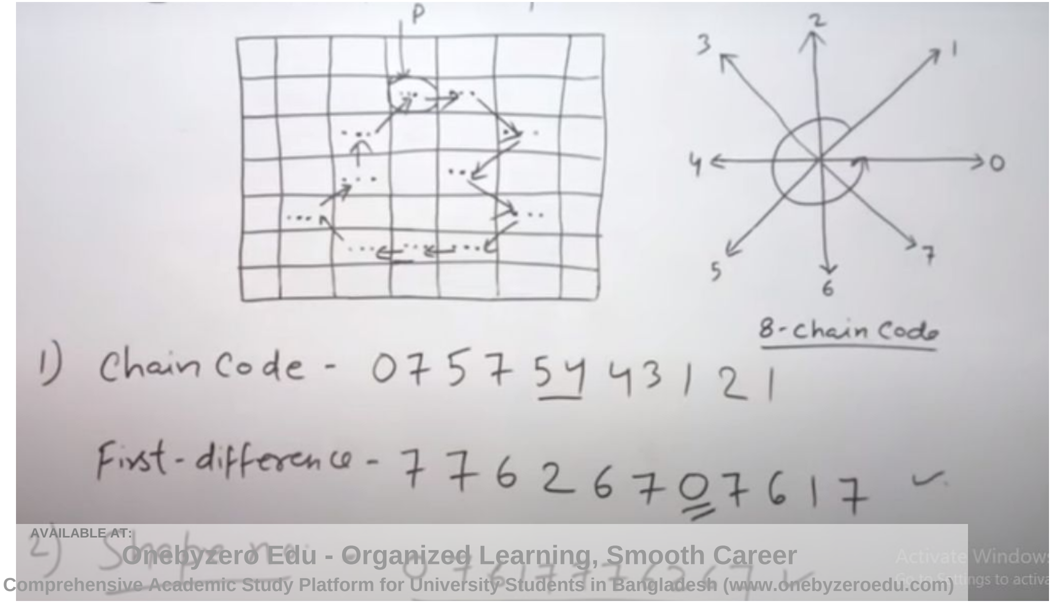
Chain code	The first difference
0 → 1	1
0 → 2	2
0 → 3	3
2 → 3	1
2 → 0	2

Example:

- a chain code: 10103322
- The first difference = 3133030

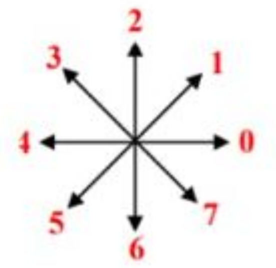
Chain Code (Cont..)

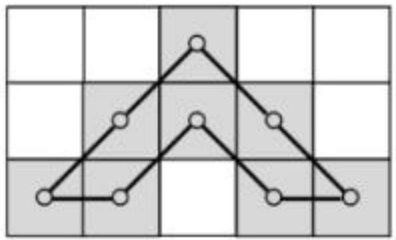
8-Direction Chain Code



Question on Chain code

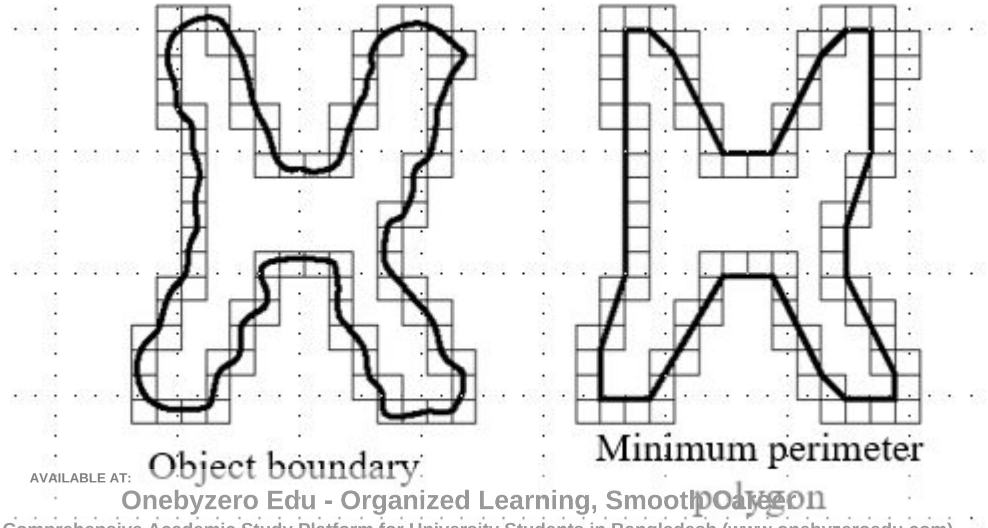
Q. What will be the 8-directions chain code of the given object and also find out the first difference and also find out the shape of the object.





Polygon Approximation

- Represent an object boundary by a polygon



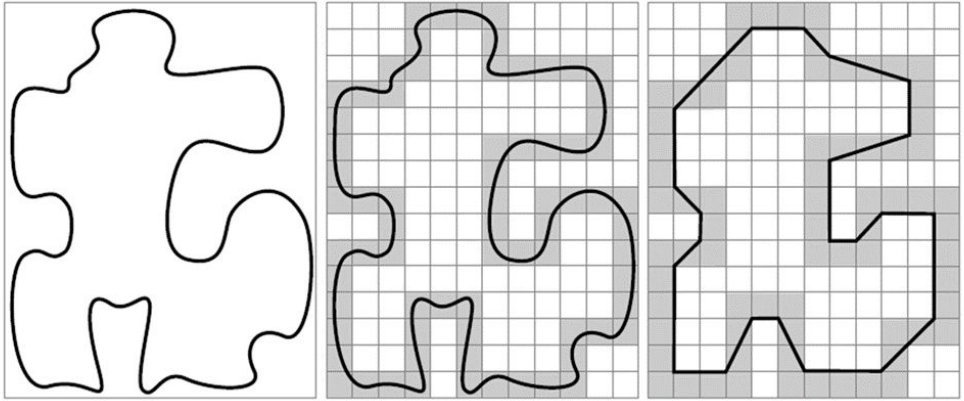
Polygon Approximation(MPP Foundation)

Minimum perimeter polygon consists of line segments that minimize distances between boundary pixels.

enclose a boundary [Fig. a] by a set of concatenated cells, as in Fig.(b).

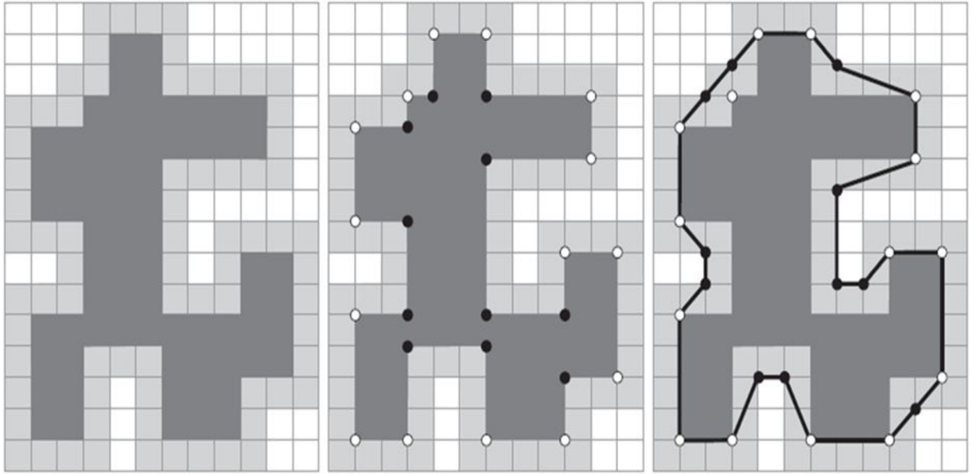
-Think of the boundary as a rubber band. It will be constrained by the inner and outer walls of the bounding region defined by the cells.

-Ultimately, this shrinking produces the shape of a polygon of minimum perimeter that circumscribes the region enclosed by the cell strip, as Fig.(c) shows.



Figure(a) An object boundary (black curve). (b) Boundary enclosed by cells (in gray). (c) Minimum-perimeter polygon obtained by allowing the boundary to shrink. The vertices of the polygon are created by the corners of the inner and outer walls of the gray region.

Polygon Approximation(MPP Foundation)



Figure(a) Region (dark gray) resulting from enclosing the original boundary by cells. (b) Convex (white dots) and concave (black dots) vertices obtained by following the boundary of the dark gray region in the counterclockwise direction. (c) Concave vertices (black dots) displaced to their diagonal mirror locations in the outer wall of the bounding region; the convex vertices are not changed.

Polygon Approximation(MPP Algorithm)

1.The MPP bounded by a simply connected cellular complex is not self-intersecting.

2.Every convex vertex of the MPP is a W vertex, but not every W vertex of a boundary is a vertex of the MPP.

3.Every mirrored concave vertex of the MPP is a B vertex, but not every B vertex of a boundary is a vertex of the MPP.

4.All B vertices are on or outside the MPP, and all W vertices are on or inside the MPP.

5.The uppermost, leftmost vertex in a sequence of vertices contained in a cellular complex is always a W vertex of the MPP.

Polygon Approximation(Splitting Techniques)

One approach to boundary segment splitting is to subdivide a segment successively into two parts until a specified criterion is satisfied.

-For a closed boundary, the best starting points usually are the two farthest points in the boundary.

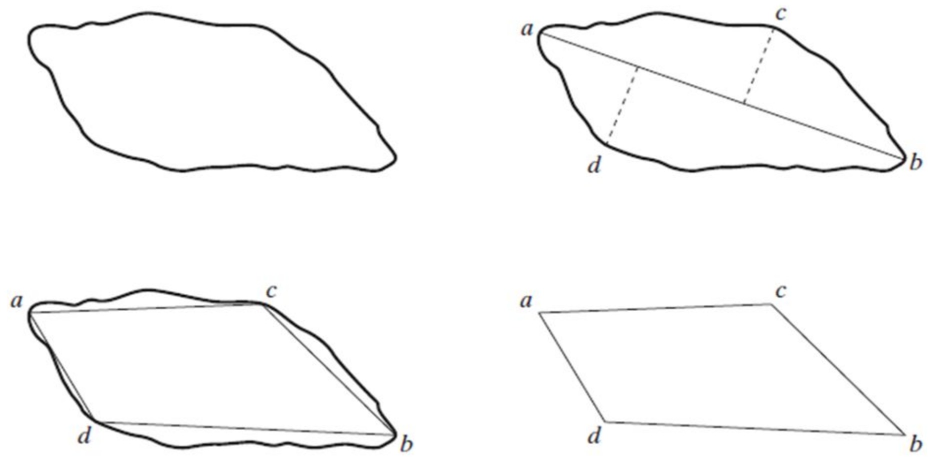


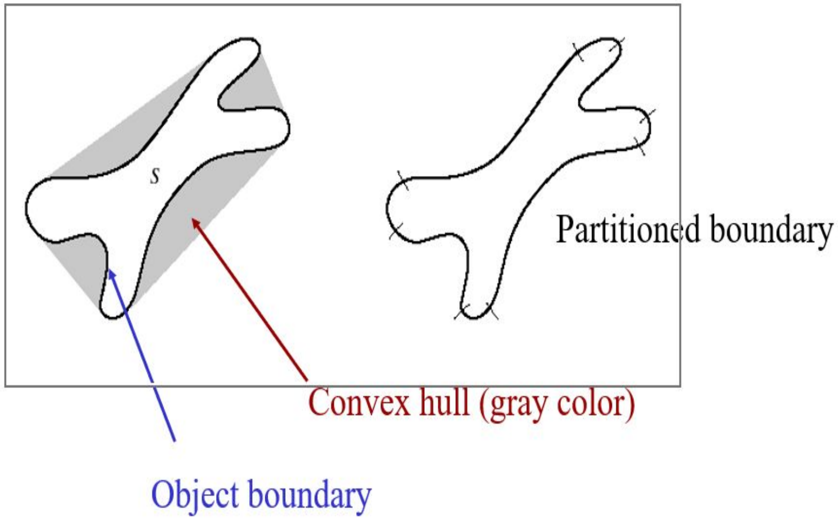
Figure 11.9 (a) Original boundary. (b) Boundary divided into segments based on extreme points. (c) Joining of vertices. (d) Resulting polygon.

Questions on Polygon Approximation

Q: Write Down the Minimum perimeter polygon Algorithm with proper diagram.

Boundary Segments

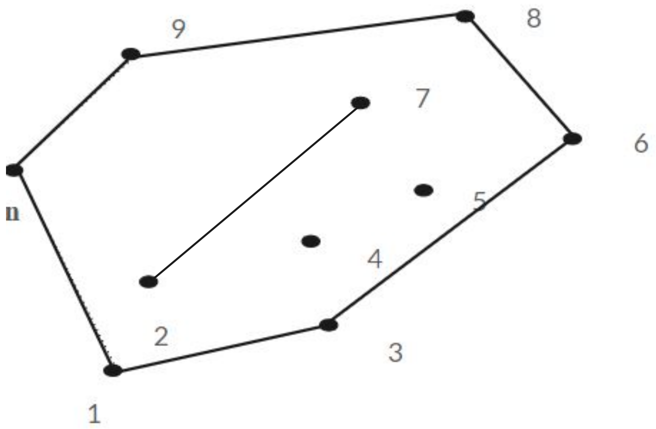
Concept: Partitioning an object boundary by using vertices of a convex hull.



- Decomposing a boundary into segments simplifies representation.
- Convex Hull can be used for decomposition.
- A new segment can be started whenever a Convex Hull deficiency is entered or exited.

Boundary Segments (Cont..)

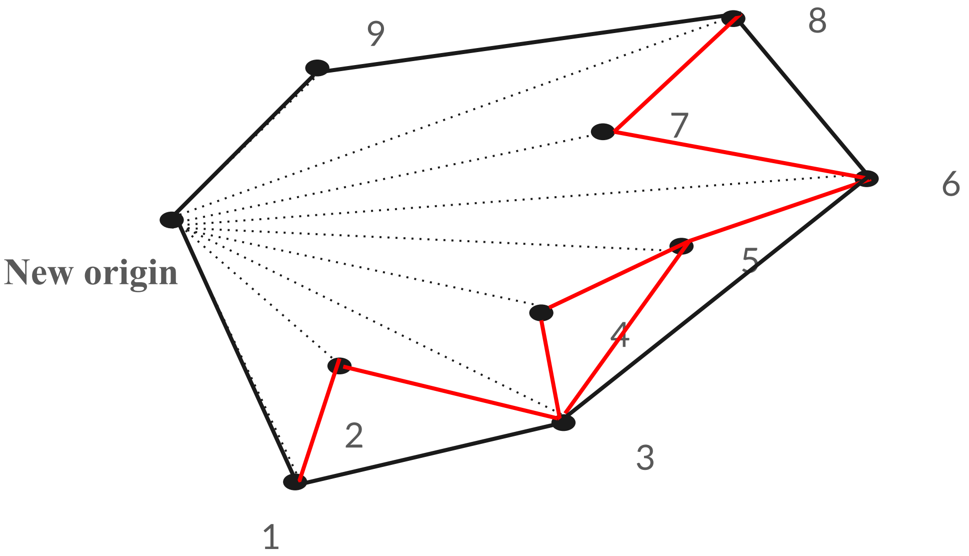
What is Convexity?



Convex Hull:

The convex hull is a way of characterizing the shape of an image by determining which pixels are adjacent to other pixels of the same intensity. This is a good way to find holes and convex features in an image.

Convex Hull Algorithm

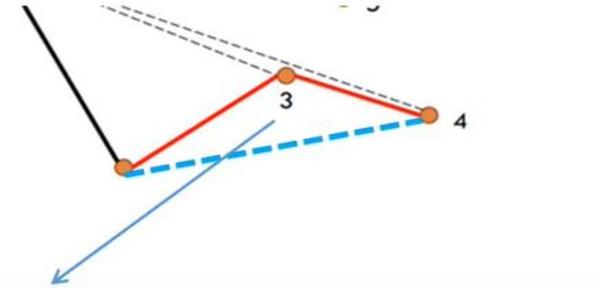


Input : A set of points on a cornea boundary

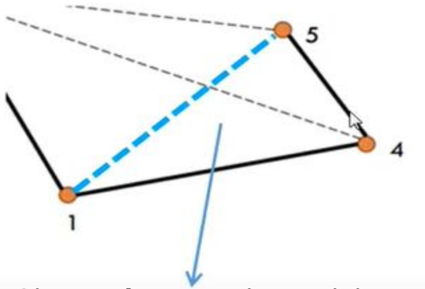
Output: A set of points on a boundary of a convex hull of a cornea

Convex Hull Algorithm (Cont..)

How can Understand the point is inside or outside?



Since The area is negative so point 3 will be eliminated



Since The area is positive so point 4 won't be eliminated

Given the coordinates of the three vertices of a triangle ABC, the area is given

$$area = \left| \frac{Ax(By - Cy) + Bx(Cy - Ay) + Cx(Ay - By)}{2} \right|$$

where A_x and A_y are the x and y coordinates of the point A etc..

Convex Hull Algorithm (Cont..)

Algorithm:

1. Sort the points by x-coordinate to get a sequence $p_1, p_2, \dots, p_n$
2. Put the points p_1 and p_2 in a list L_{upper} with p_1 as the first point
3. For i = 3 to n
4. Do append p_i to L_{upper}
5. While L_{upper} contains more than 2 points and the last 3 points in L_{upper} do not make a right turn
6. Do delete the middle point of the last 3 points from L_{upper}

Convex Hull Algorithm (Cont..)

Convex Hull Algorithm Implementation in C

```
#include <bits/stdc++.h>
#define MAX 100009
#define LL long long
#define sq(x) ((x)*(x))
using namespace std ;
struct point {
    LL x, y;
} P[MAX], C[MAX], P0;

int main(void){
    freopen("in.txt","r",stdin) ;
    int np,nc ;
    scanf("%lld",&np) ;
    for(int i=0;i<np;i++){
        scanf("%lld %lld",&P[i].x,&P[i].y) ;
    }
    ConvexHull(np,nc) ;
    for(int i=0;i<nc;i++){
        printf("%lld %lld\n",C[i].x,C[i].y) ;
```

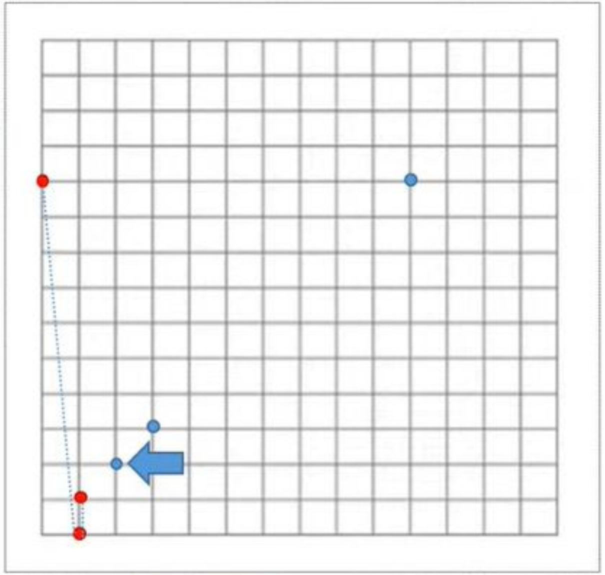
Convex Hull Algorithm Implementation in C (Cont..)

```
void ConvexHull(int np, int &nc) {

    int i, j, pos = 0;
    for(i=1; i<np; i++){
        if(P[i].x<P[pos].x || (P[i].x==P[pos].x && P[i].y>P[pos].y))
        }

    swap(P[0], P[pos]);
    P0 = P[0];

    sort(&P[1], P+np,comp);
    C[0] = P[0] ;
    C[1] = P[1] ;
    C[2] = P[2];

    for(i=j=3; i<np; i++) {
        while(TriArea2(C[j-2], C[j-1], P[i]) <= 0) j--;
        C[j++] = P[i];
    }

    nc = j;
```

```
inline LL TriArea2(point a, point b, point c) {
    return (a.x*(b.y-c.y) + b.x*(c.y-a.y) + c.x*(a.y-b.y));
}

inline LL sqDist(point a, point b) {
    return (sq(a.x-b.x) + sq(a.y-b.y));
}

bool comp(point a, point b) {
    if(TriArea2(P0,a,b)==0) // co linear
        return sqDist(P0,a)<sqDist(P0,b) ;

    double d1x = a.x-P0.x ; double d1y = a.y-P0.y ;
    double d2x = b.x-P0.x ; double d2y = b.y-P0.y ;
    return (atan2(d1y,d1x)-atan2(d2y,d2x))<0  ;
}
```

Convex Hull Algorithm Implementation in C (Cont..)

```
void ConvexHull(int np, int &nc) {

    int i, j, pos = 0;
    for(i=1; i<np; i++){
        if(P[i].x<P[pos].x || (P[i].x==P[pos].x && P[i].y>P[pos].y))
        }

    swap(P[0], P[pos]);
    P0 = P[0];

    sort(&P[1], P+np,comp);
    C[0] = P[0] ;
    C[1] = P[1] ;
    C[2] = P[2];

    for(i=j=3; i<np; i++) {
        while(TriArea2(C[j-2], C[j-1], P[i]) <= 0) j--;
        C[j++] = P[i];
    }

    nc = j;
}
```



P[]	(0,10)	(1,0)	(1,1)	(2,2)	(3,3)	(10,10)
C[]	(0,10)	(1,0)	(1,1)			

Image Description

- Refers to extracting meaningful information or features from the image for various tasks like image recognition, classifications.
- Broadly divided into two types:

1. Boundary descriptors
2. Regional descriptors

Boundary descriptors

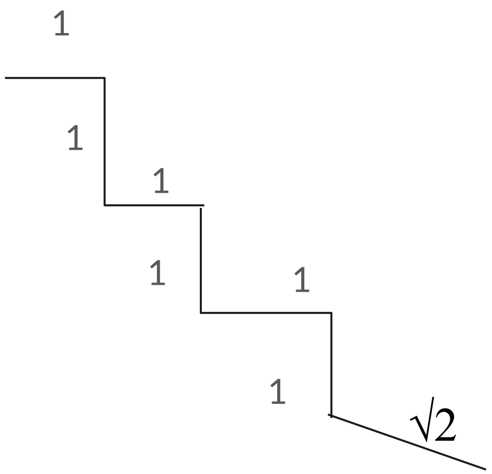
- Boundary descriptor are numerical representation used to characterize the shape of an object's boundary
- Allow to calculate the properties (i.e. length, diameter, curvature) of any object
- Allow us to compare between two boundaries to detect whether they are the same or not.
- Four types of boundary descriptors:
  1. Simple descriptors
  2. Shape Number
  3. Fourier descriptors
  4. Statistical moments

Simple Descriptors

- **Length(perimeter):**
- ➔ Number of pixels along a boundary gives its length.
- ➔ Formula:

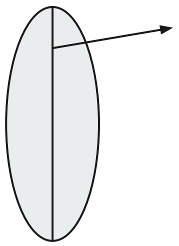
  Length = Vertical + Horizontal + (√2 * Diagonal)

Length = 1+1+1+1+1+1+√2 =6+√2



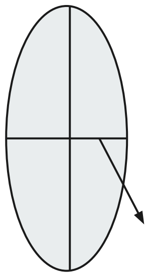
Simple descriptors(Cont.)

- **Major axis:** Line segment connecting the two extreme points that comprise the diameter.



- **Minor axis:** The line perpendicular to major axis is said minor axis.

- **Eccentricity:** The ratio of major axis to the minor axis is called the eccentricity.



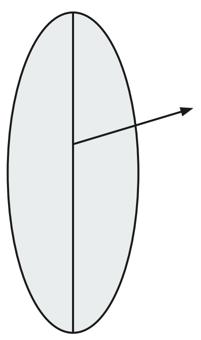
Simple descriptors(Cont.)

- **Diameter**

Diameter = $max_{i,j}[D(P_i, P_j)]$

= major axis

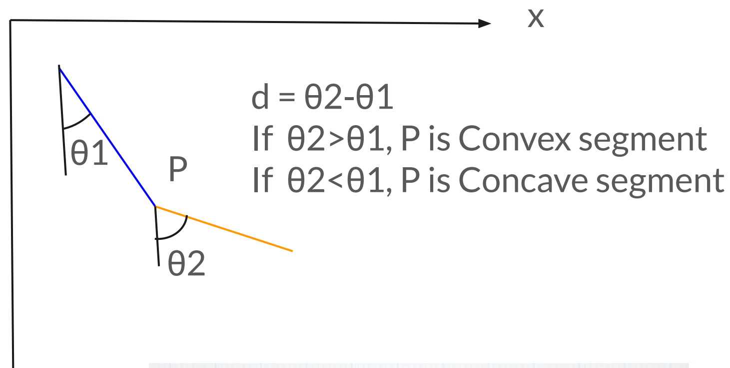
Here, D is the distance measurement method, for e.g., euclidean.

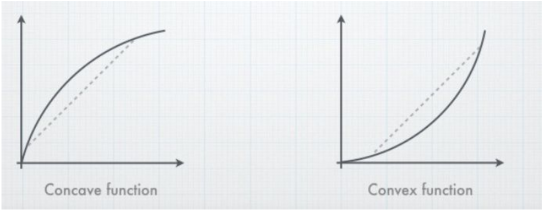


Simple descriptors(Cont.)

❑ Curvature:

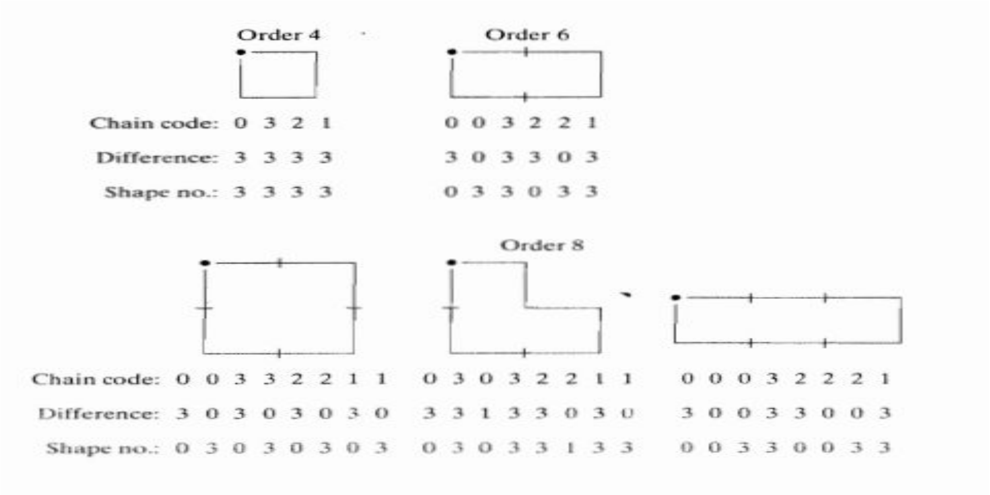
- ➔ Defined as the rate of change of slope.
- ➔ To calculate the curvature, the boundary is traversed in the clockwise direction and a vertex point P belongs to a convex segment, if the change in slope at P is non-negative (i.e. +ve), otherwise P is said to belong to a concave segment.

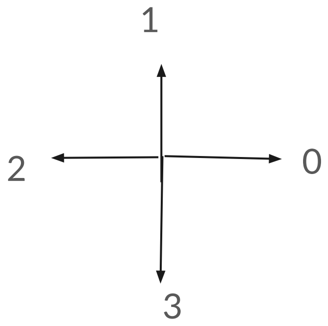




Shape Number

- Defined as first difference of smallest number.
- The order of n shape number is defined as the number of digits in its representation.
- Circularly shift the differential chain code to find the smallest numeric sequence.





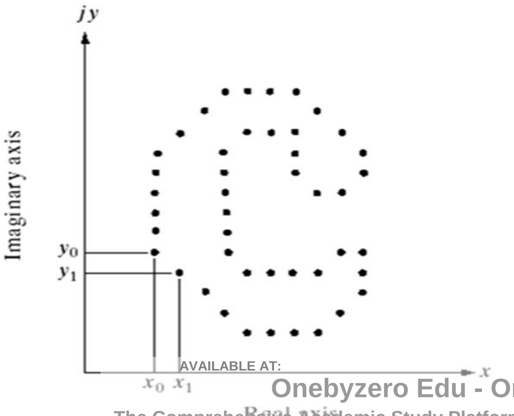
Assignment

Find out the shape number of these boundary on order n=18.

Answer: See page no -839 in textbook

Fourier Descriptors

- Represent the boundary as a sequence of coordinates. They convert the object's boundary into a set of numbers in the frequency domain, capturing its essential shape features.
- Treat each coordinate pair as a complex number.
- Advantage: Reduces a 2-D to a 1-D problem.



$$s(k) = [x(k), y(k)];\ \ k = 0, 1, 2, \ldots, K-1$$

$$s(k) = x(k) + jy(k)$$

Fourier descriptors(according to DFT):

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

Fourier Descriptors(Cont.)

- **Reconstruction formula (according to IDFT):**

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

- Consider only first P, as it is equivalent to a(u)=0 for u > P-1 where P is the number of fourier coefficient used to reconstruct the boundary.
- **Approximate reconstruction using only first P Fourier coefficients:**

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

Fourier Descriptors(Cont.)

Fig: Boundary of human chromosome (2868 points). b-h boundary constructed using 1434, 286, 142, 72, 36, 18 and 8 Fourier descriptors respectively. The numbers are approximately 50%, 10%, 5%, 2.5% 1.25%, 0.63%,0.28% of 2868, respectively.

Fourier Descriptors(Cont.)

❖ Basic properties of fourier descriptors:

Transformation	Boundary	Fourier Descriptor
Identity	$s(k)$	$a(u)$
Rotation	$s_r(k) = s(k)e^{j\theta}$	$a_r(u) = a(u)e^{j\theta}$
Translation	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
Scaling	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Starting point	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

Here,

$$\Delta_{xy} = \Delta x + j\Delta y$$

Statistical moments

- The shape of a boundary segment can be represented by using statistical moments, such as mean, variance and higher order moments.
- Formula:

$$\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i)$$

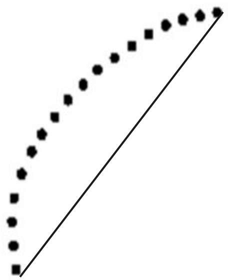
Where,

n=order

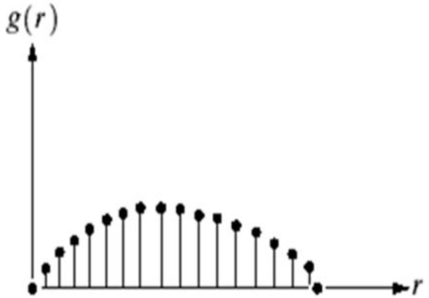
r= random variable

K= number of discrete amplitude

$$m = \sum_{i=0}^{K-1} r_i g(r_i)$$



a. Boundary Segment



b. Representation as a 1-D function

Statistical moments(cont.)

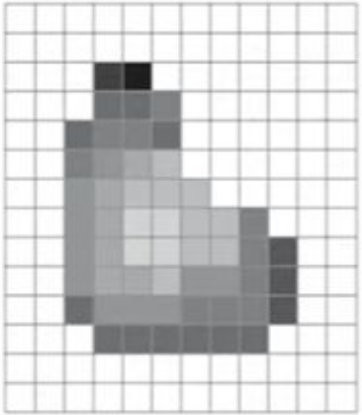
❑ Advantages:

➔ Straightforward techniques

➔ Represent as 1-D function

➔ Also carry physical interpretation

Regional Descriptors



Purpose: to describe regions or "areas"

1. Some simple regional descriptors

- area of the region
- length of the boundary (perimeter) of the region
- Compactness

$$\text{Compactness} = \frac{4\pi \times \text{Area}}{\text{Perimeter}^2}$$

Compactness is a shape descriptor that measures how closely a shape resembles a circle

- **Area**: The number of pixels within the region.
- **Perimeter**: The length of the boundary of the region.
- 4π: A normalization factor to ensure the compactness value for a perfect circle is 1.

2. Topological Descriptors

3. Texture

Topological Descriptors

Use to describe holes and connected components of the region

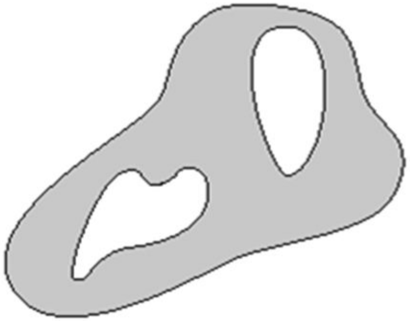


FIGURE 11.17 A region with two holes.

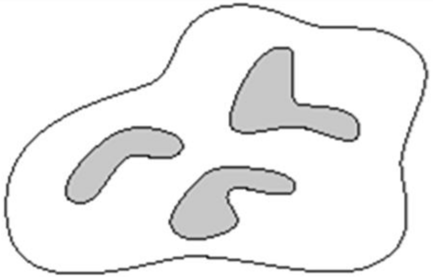


FIGURE 11.18 A region with three connected components.

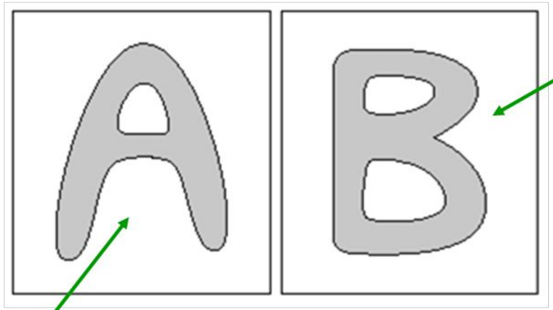
Euler number (E):

$$E = C - H$$

C = the number of connected components

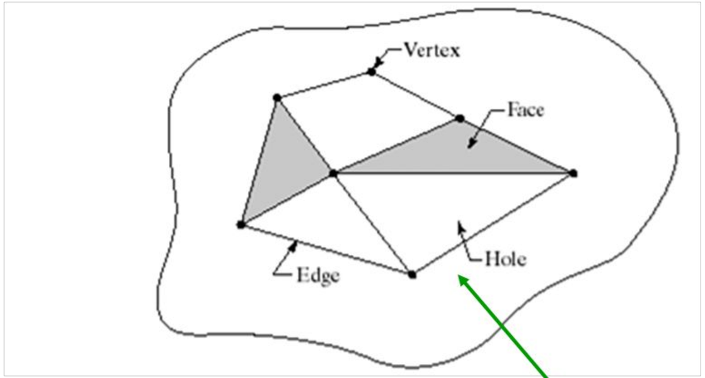
H = the number of holes

Topological Descriptors (cont.)



E = -1

E = 0



E = -2

Euler Formula

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

V = the number of vertices

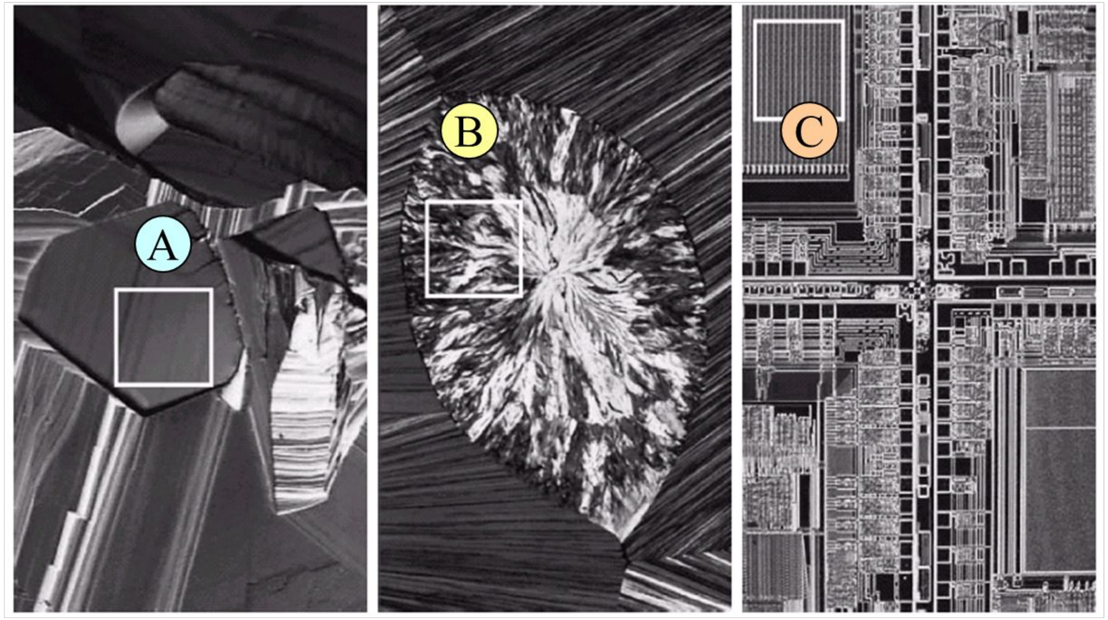
Q = the number of edges

F = the number of faces

Texture Descriptors

Purpose: to describe "texture" of the region.

Examples: optical microscope images:



Superconductor (smooth texture) | Cholesterol (coarse texture) | Microprocessor (regular texture)

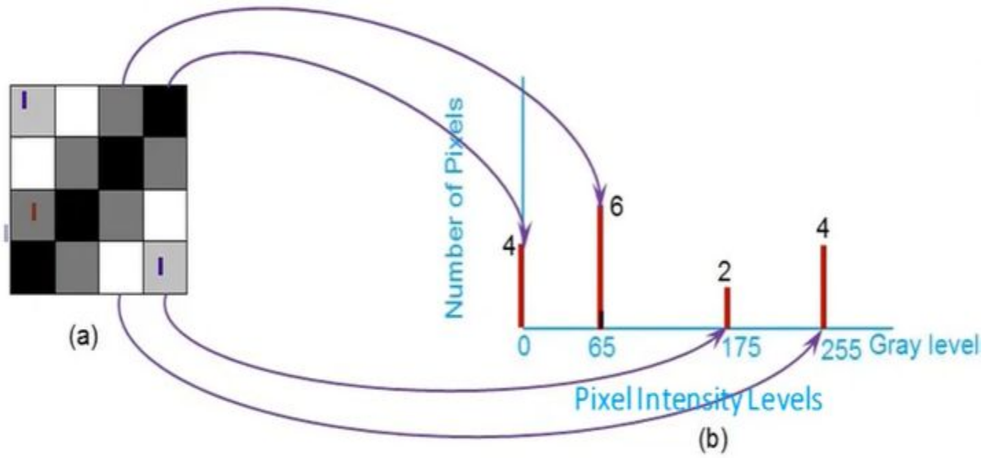
Texture Descriptors

- Texture descriptors are features used in image processing and computer vision to characterize the texture of an image or a region within an image.
- These descriptors capture information about the **spatial** arrangement of pixel intensities, helping to differentiate between different types of textures such as smooth, rough, patterned, or random.

Ways to approach [Common Type of Texture Descriptors**]-**

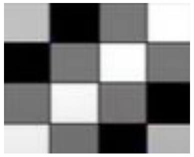
1. Statistical
2. Structural
3. Spectral

Image Histogram

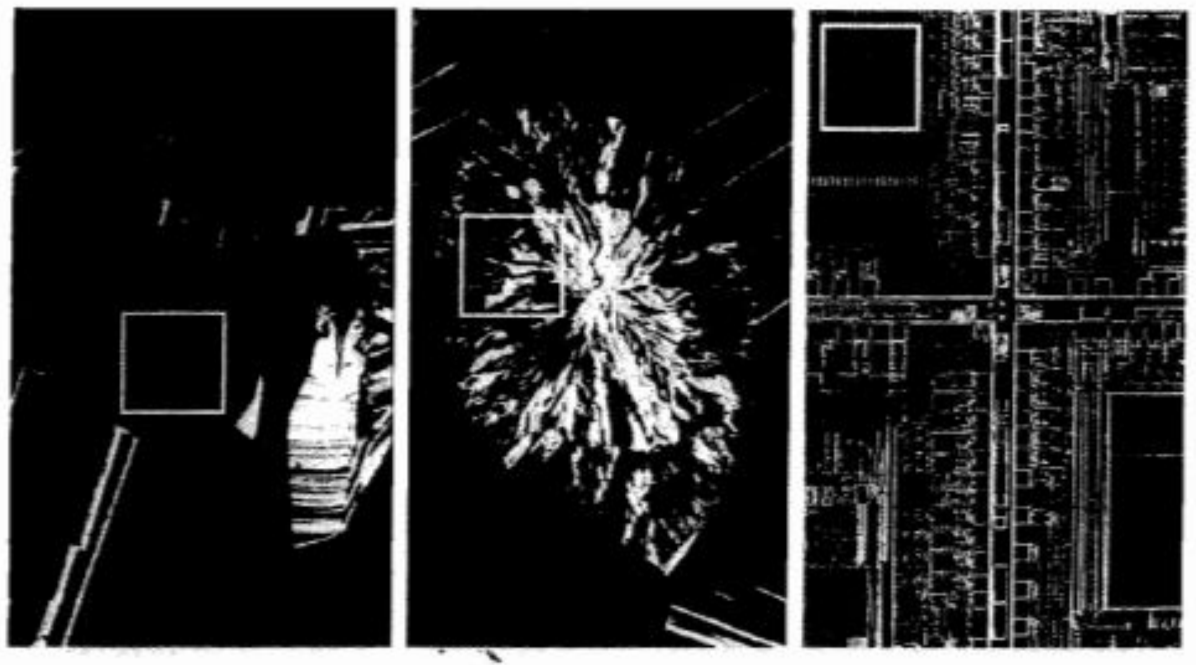


Descriptors

- Mean
- Variance
- Skewness
- Kurtosis



Anomalies



Texture	Mean	Standard deviation	R (normalized)	Third moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

Co-occurrence Matrix

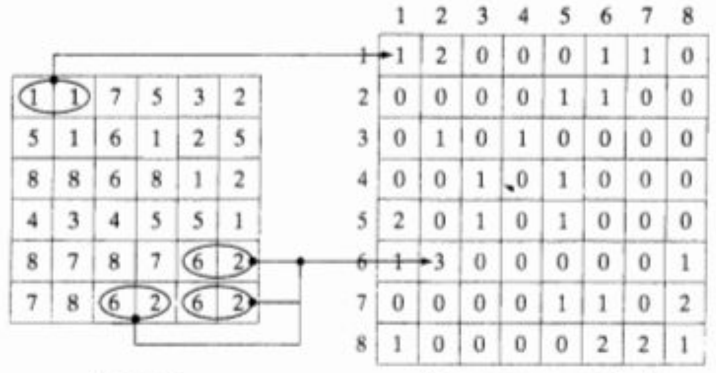


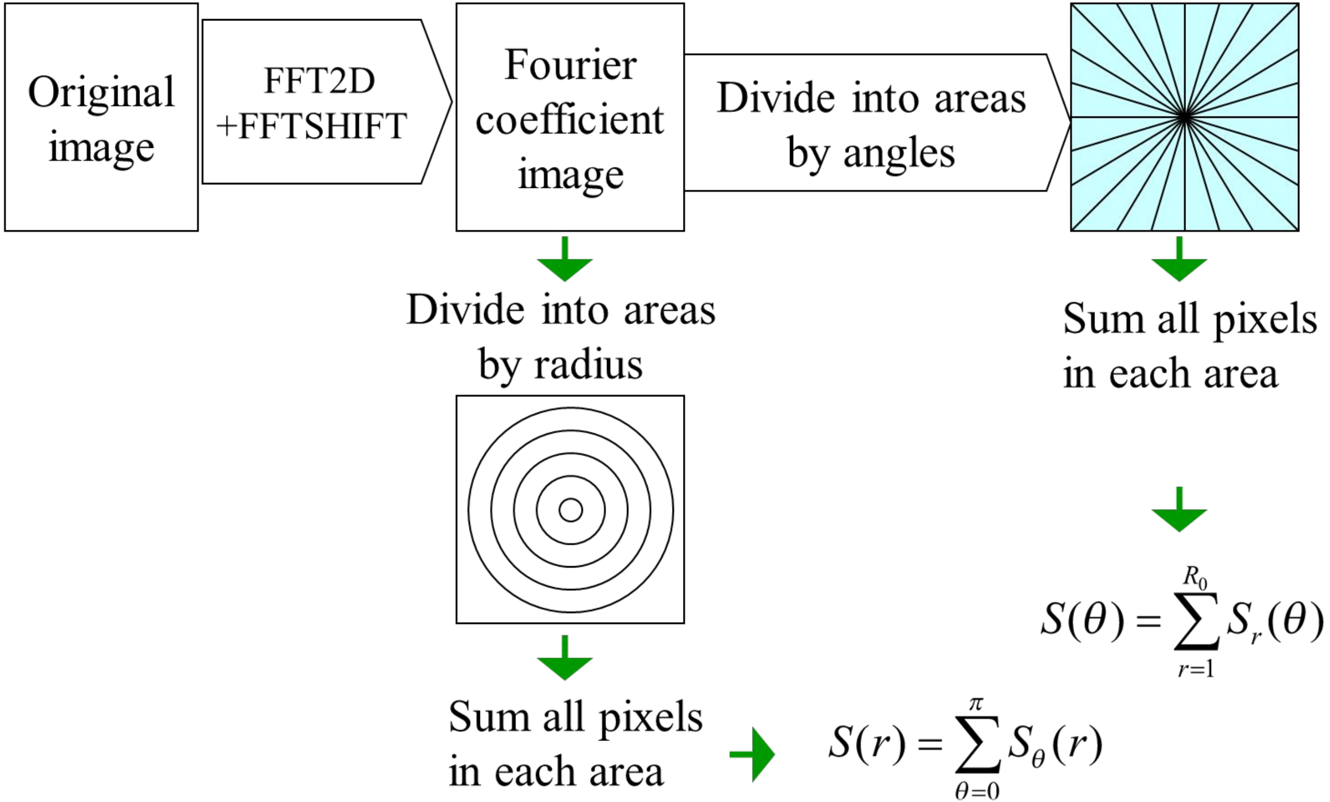
Image f Co-occurrence matrix **G**

Descriptors Used for characterizing Co-occurrence Matrices:

1. Maximum Probability
2. Correlation
3. Contrast
4. Uniformity
5. Homogeneity
6. Entropy

Fourier Approach for Texture Descriptor

Concept: convert 2D spectrum into 1D graphs



$$S(\theta) = \sum_{r=1}^{R_0} S_r(\theta)$$

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r)$$

Fourier Approach for Texture Descriptor

Original image

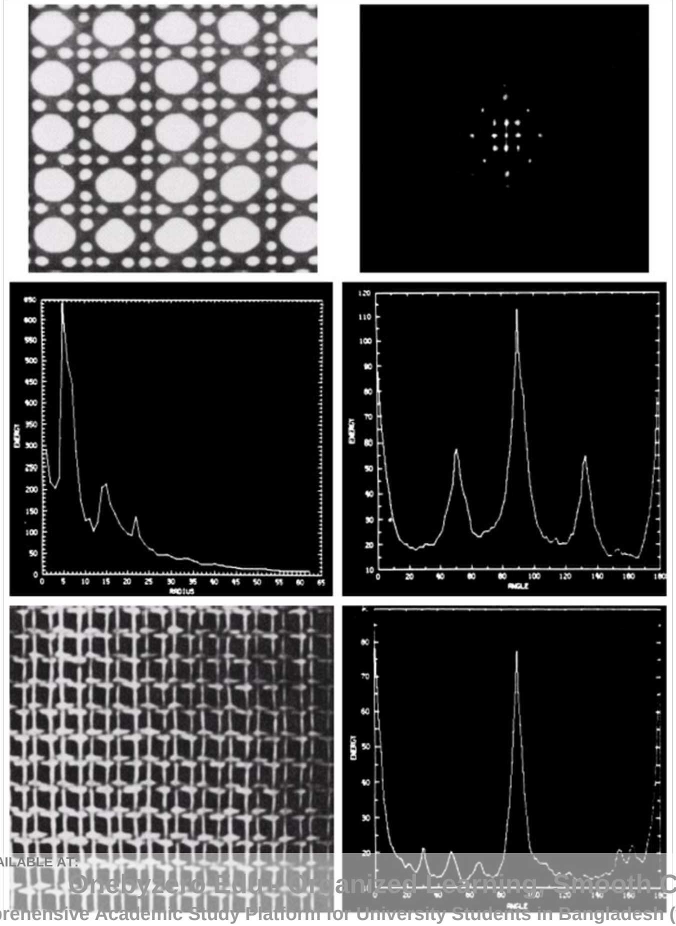
2D Spectrum (Fourier Tr.)

$S(r)$

$S(\theta)$

Another image

Another $S(\theta)$

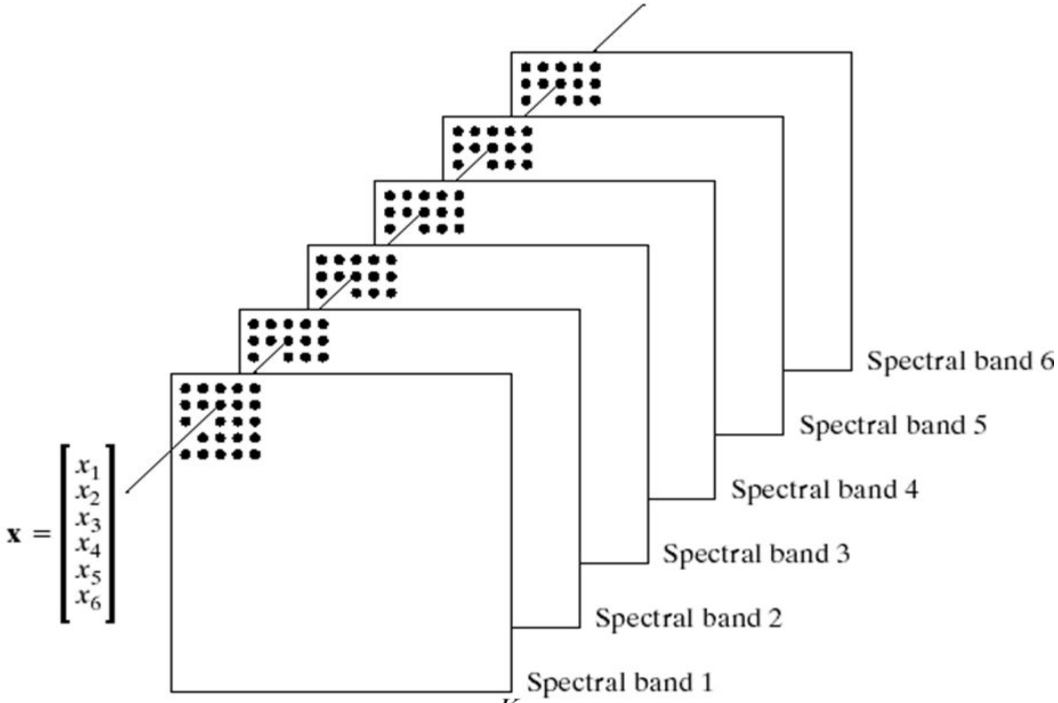


Principal Components for Description

Purpose: to reduce dimensionality of a vector image while maintaining information as much as possible.

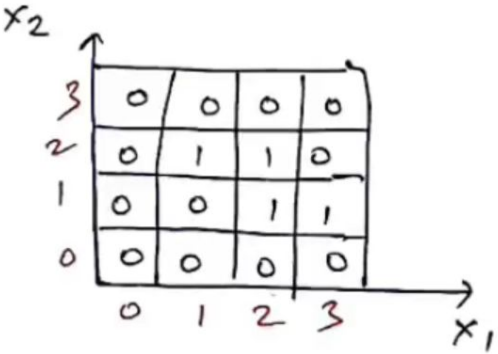
Let $\mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_n]^T$

Mean: $\mathbf{m}_x = E\{\mathbf{x}\} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k$



Covariance matrix $\mathbf{C}_x = E\{(\mathbf{x}-\mathbf{m}_x)(\mathbf{x}-\mathbf{m}_x)^T\} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k\mathbf{x}_k^T - \mathbf{m}_x\mathbf{m}_x^T$

EXAMPLE :



1] **Find x :**

$$x = \left\{ \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix} \right\}$$

2] **Find mean** : (μ_x)

$$\mu_x = \left\{ \begin{pmatrix} 2 \\ 1.5 \end{pmatrix} \right\}$$

3] $C_x = E\left[(x_i - \mu_x)\,(x_i - \mu_x)^T \right]$

Hotelling transformation

Let $\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x)$

Where A is created from eigenvectors of C_x as follows
Row 1 contain the 1st eigenvector with the largest eigenvalue.
Row 2 contain the 2nd eigenvector with the 2nd largest eigenvalue.

....

Then we get

$$\mathbf{m}_y = E\{\mathbf{y}\} = 0$$

$$\mathbf{C}_y = \mathbf{A}\mathbf{C}_x\mathbf{A}^T$$

and

$$\mathbf{C}_y = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \lambda_1 \end{bmatrix}$$

Then elements of $\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x)$ are uncorrelated. The component of y with the largest λ is called the principal component.

output

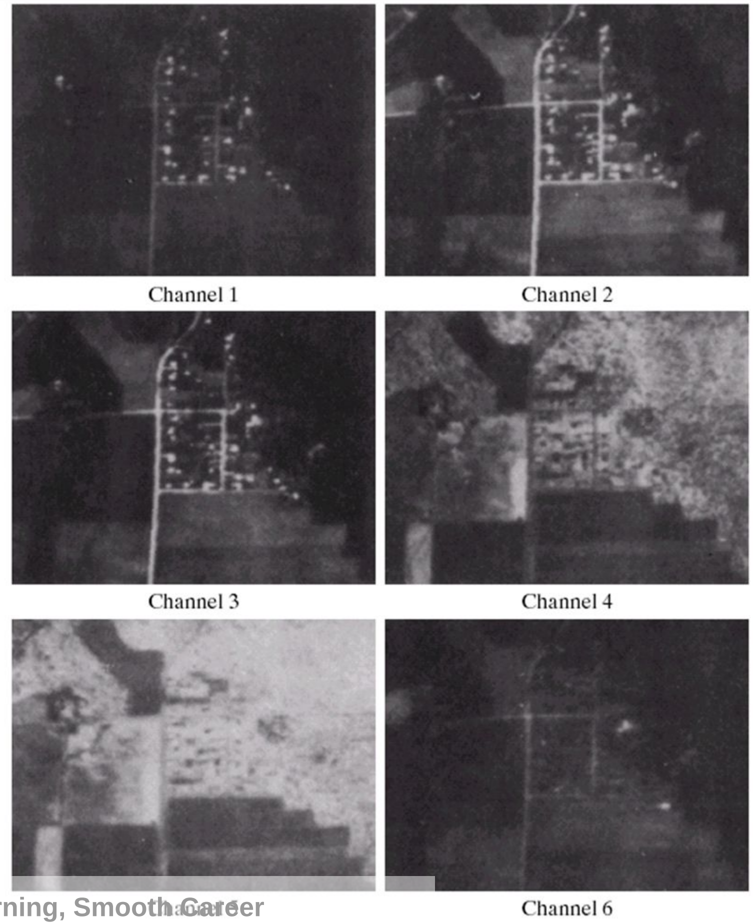
then: $y = A(x - \mu_x) = [\ \ ]$

$2 \times 2 \quad 2 \times 4 = 2 \times 4$

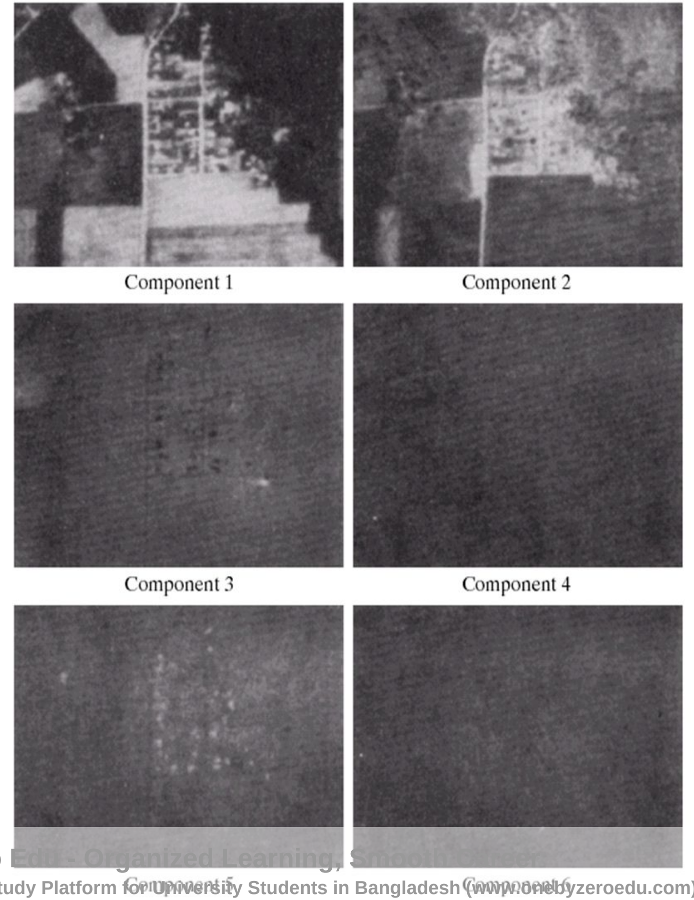
Example: Principal Components

6 spectral images from an airborne Scanner.

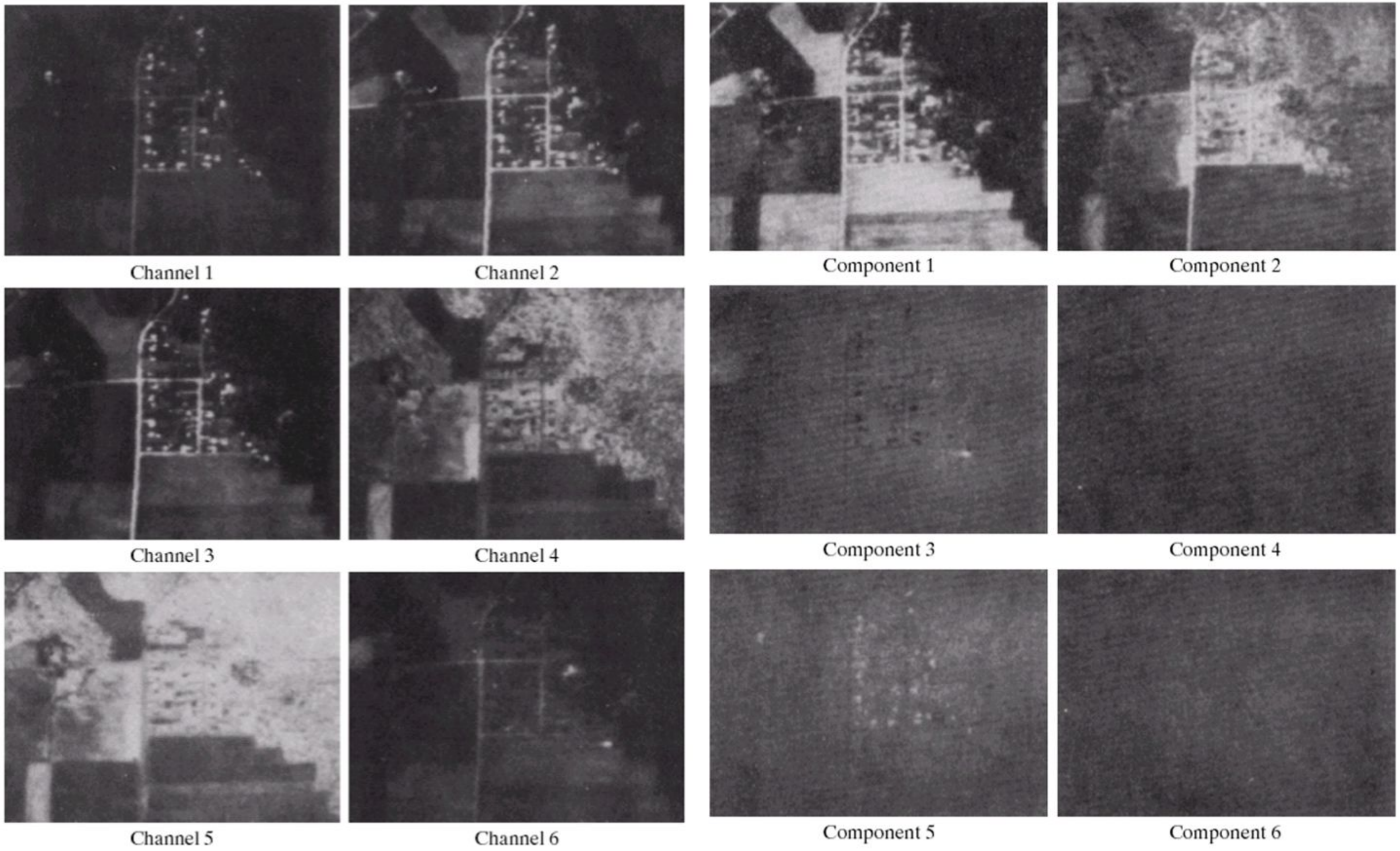
Channel	Wavelength band (microns)
1	0.40–0.44
2	0.62–0.66
3	0.66–0.72
4	0.80–1.00
5	1.00–1.40
6	2.00–2.60



Example: Principal Components (cont.)



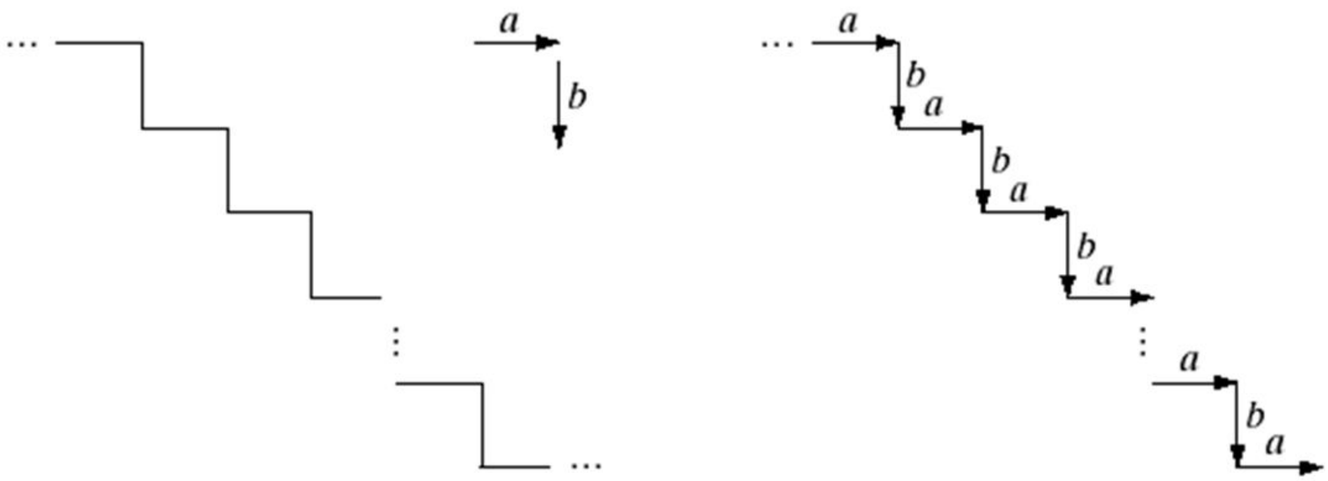
Example: Principal Components (cont.)



Relational Descriptors

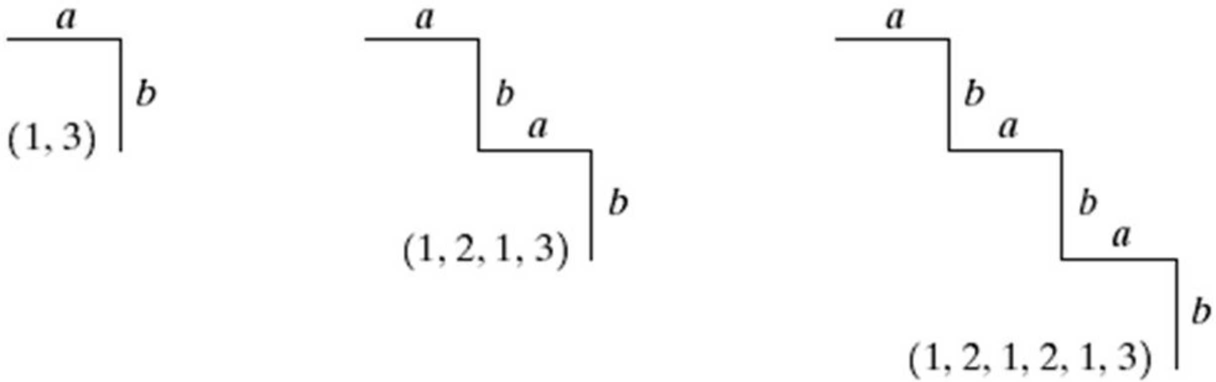
a b

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



Relational Descriptors

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



Relational Descriptors

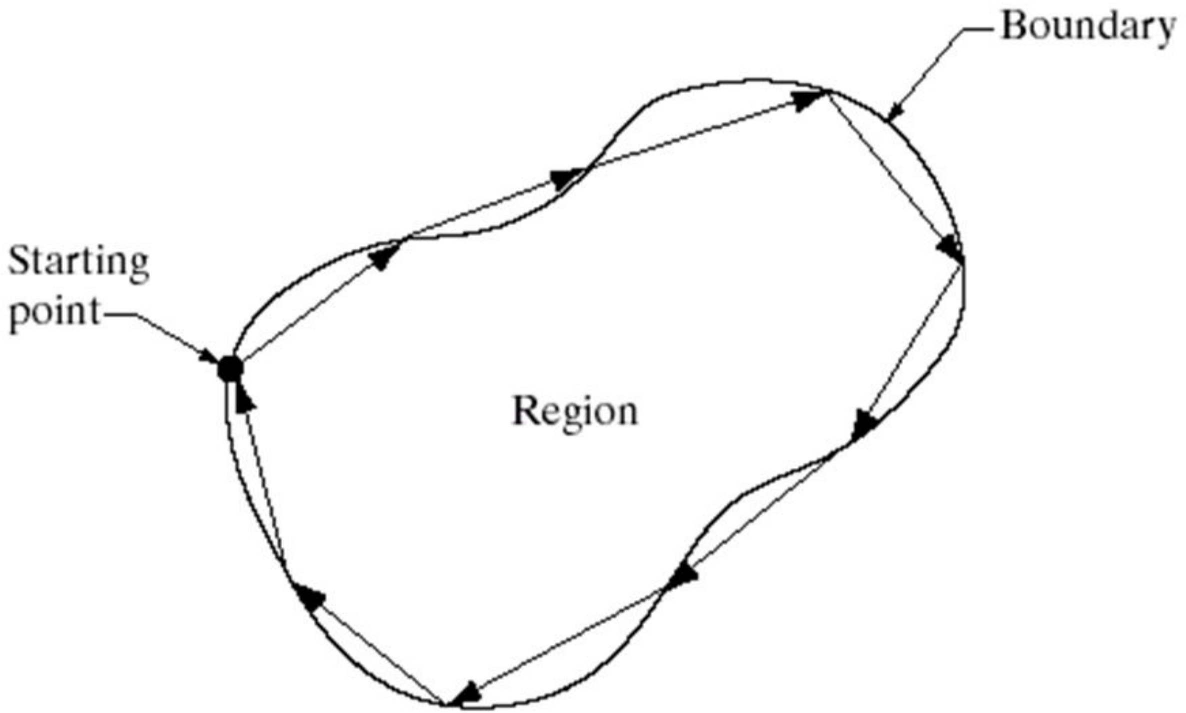
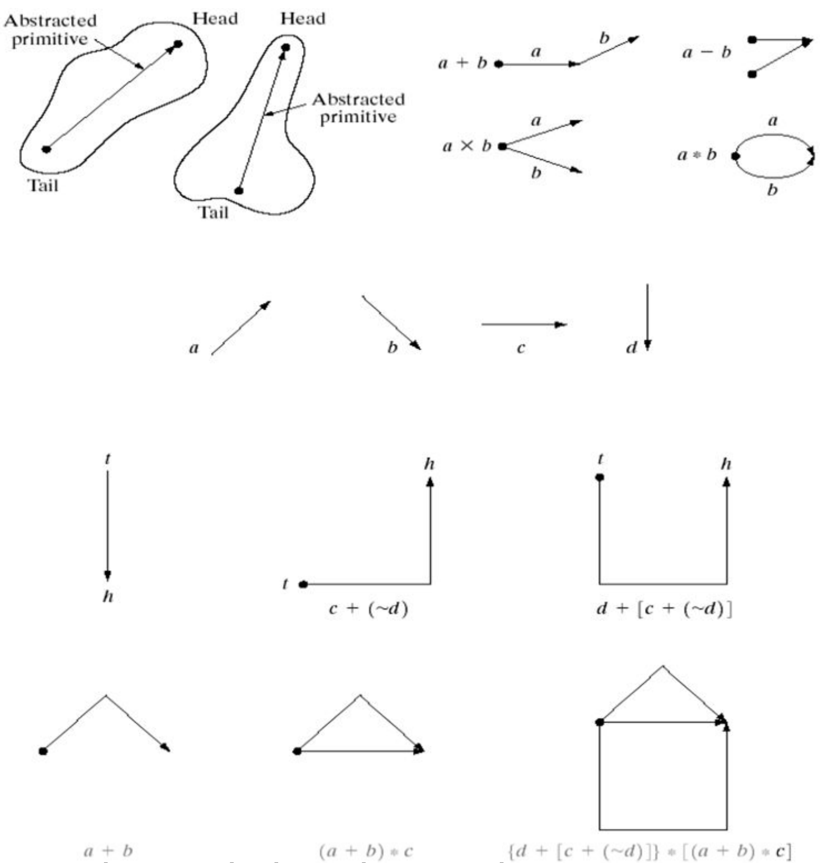


FIGURE 11.32 Coding a region boundary with directed line segments.

Relational Descriptors



Relational Descriptors

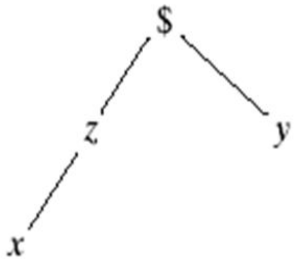
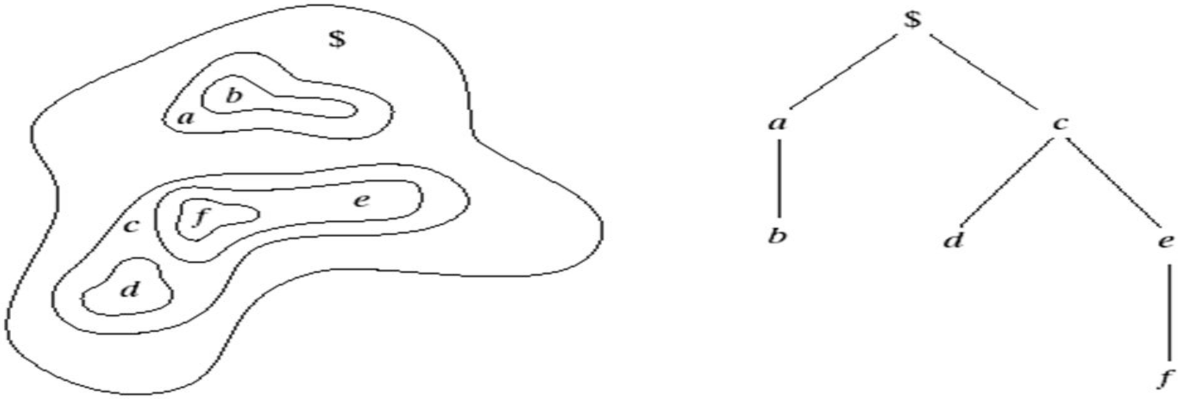


FIGURE 11.34 A simple tree with root $ and frontier *xy*.



a b

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