MODULE VI

MORPHOLOGICAL OPERATIONS

Basics of Set Theory; Dilation and Erosion - Dilation, Erosion; Structuring Element; Opening & Closing; Hit or Miss Transformation.

Representation and Description Representation - Boundary, chain codes, Polygonal approximation approaches, Boundary Segments.

- Mathematical morphology is a collection of non linear processes which can be applied to an image to remove details smaller than a certain reference shape.
- Morphological operations can be used to extract the edges of an image, filter an image, skelationise an image etc.
- Tool for extracting image components that are useful in the representation & descriptn of region shape such as boundaries, skeletons & convex hull.
- Sets in mathematical morphology represent objects in an image.
- Mathematical morphology can be used as the basis for developing image-segmentation procedures with a wide range of applications.

STRUCTURING ELEMENTS

↳ Mathematical morphology is a collection of non-linear processes which can be applied to an image, to remove details smaller than a certain reference shape, which is called structuring Element

↳ S.E. is a morphological operation plays an imp role with its different shape and size.

↳ Shape & size are defined by a no. of 0's & 1's in S.E's.

↳ Morphological operations are defined by moving a S.E over the binary image to be modified, in such a way that it is centered over every image pixel at some pt.

BASIC SET THEORY

↳ Morphology is based on set theory, deals with necessary operations in set theory.

1. Union.

↳ The Union of images A & B as $A \cup B$

$$A \cup B = \text{def} \{x \mid x \in A \text{ or } x \in B\}$$

2. Intersection

$$A \cap B = \text{def} \{x \mid x \in A \text{ and } x \in B\}$$

3. Difference.

$$A - B = \text{def} \{x \mid x \in A \text{ and } x \notin B\}$$

↳ A-B is also called the relative complement of B relative to A.

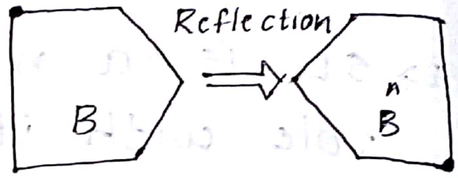
4. Complement

$$A^c = \text{def}\ \{x \mid x \notin A\}$$

5. Reflection

↳ The reflection of a set B, denoted $\hat{B}$ is defined as:

$$\hat{B} = \{w \mid w = -b \text{ for } b \in B\}.$$



6. Translation

↳ Translation operation of Image A is defined as:

↳ The translation of a set B by point $z = (z_1, z_2)$ denoted $(B)_z$ is defined as:

$$(B)_z = \{c \mid c = b + z \text{ for } b \in B\}$$

↳ If B is the set of pixels representing an object in an image, the $(B)_z$ is the set of points in B whose (x, y) coordinates have replaced by $(x + z_1, y + z_2)$

7. Logical Operations : AND, OR, NOT, XOR.

STANDARD BINARY MORPHOLOGICAL OPERATIONS

↳ The basic morphological operations are dilation & erosion.

↳ Expressed by a kernel operating on an i/p binary image X, where white pixels denote uniform regions and black pixels denote region boundaries.

DILATION $\oplus$ grow Does the S.E hit the set?

↳ Dilation is a process in which the binary image is expanded from its original shape.

↳ The way of the binary image is expanded is determined by the structuring element. This S.E is smaller in size compared to the image itself and normally the size used for the S.E is 3x3.

↳ The dilation process is ||r to the convolution process, ie, the S.E is reflected & shifted from Left to Right & from top to bottom, at each shift; the process will look for any overlapping ||r pixels b/w the S.E & that of the binary image.

↳ If there exists an overlapping then the pixels under the centre position of the S.E will be turned to 1 or Black.

[x,y plane (2D)]

↳ With A & B as sets in Z^2, the dilation of A by B denoted $A \oplus B$ is defined as:

$\hat{B}$: Reflection.

z : each pt in S.E B.

$$A \oplus B = \{ z \mid (\hat{B})_z \cap A \neq \phi \}$$

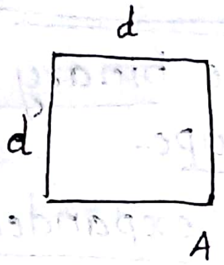
This eqn is based on reflecting B about its origin and shifting this reflection by z.

↳ The dilation of A by B then is the set of all displacements, z, such that $\hat{B}$ and A overlap by atleast one element.

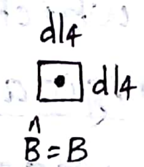
Reflectn

$$A \oplus B = \{ z \mid [(\hat{B})_z \cap A] \subseteq A \}$$

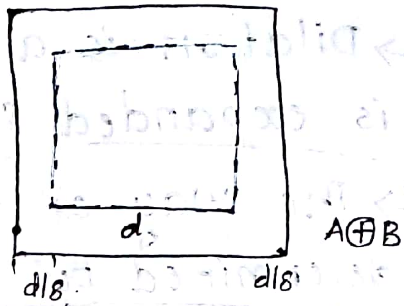
B is a structuring element & A is the set (image objects) to be dilated.



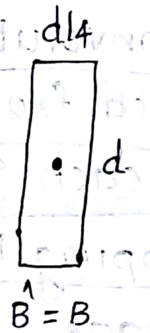
a) Set A



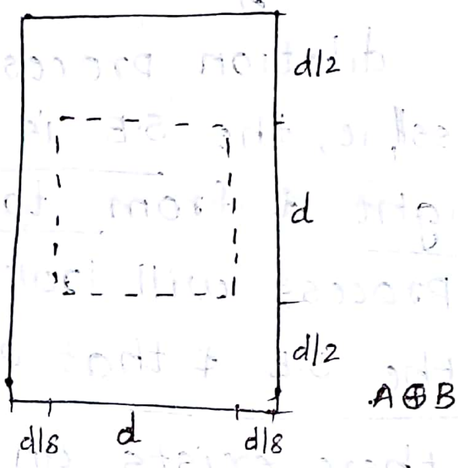
b) Square S.E (the dot denotes the origin)



c) Dilation of A by B



d) Elongated S.E



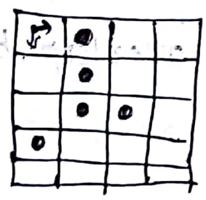
e) Dilation of A using this element.

Fig. An illustration of Dilation.

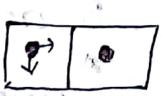
↳ The dilation is an <u>expansion operator</u> that <u>enlarges binary objects.</u>

↳ Dilation has many uses, but the major one is <u>bridging gaps in an image</u>, due to the fact that B is expanding the features of X.

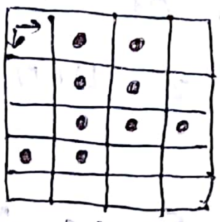
Eg: 1)



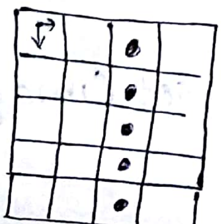
A



B

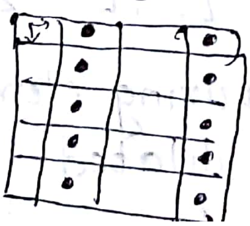


A⊕B



A

B



A⊕B

EROSION $\ominus$ Shrink the object :- Does the S·E fit the set?

- Erosion is the counter-process of dilation.
- If dilation enlarges an image then erosion shrinks the image.
- The way of the image is shrunk is determined by S·E.
- The S·E is normally smaller than the image with a 3x3 size.
- Ensure faster computation time when compared to larger S·E size.
- Almost ||r to dilation process, the erosion process will move the S·E from left to right & top to bottom.
- At the centre position, indicated by the centre of the S·E, the process will look for whether there is a complete overlap with the S·E or not.
- If there is no complete overlapping then the centre pixel indicated by the centre of the S·E will be set white or 0.
- With A and B as sets in Z^2 (2D), the erosion of A by B, denoted $A \ominus B$ is defined as:

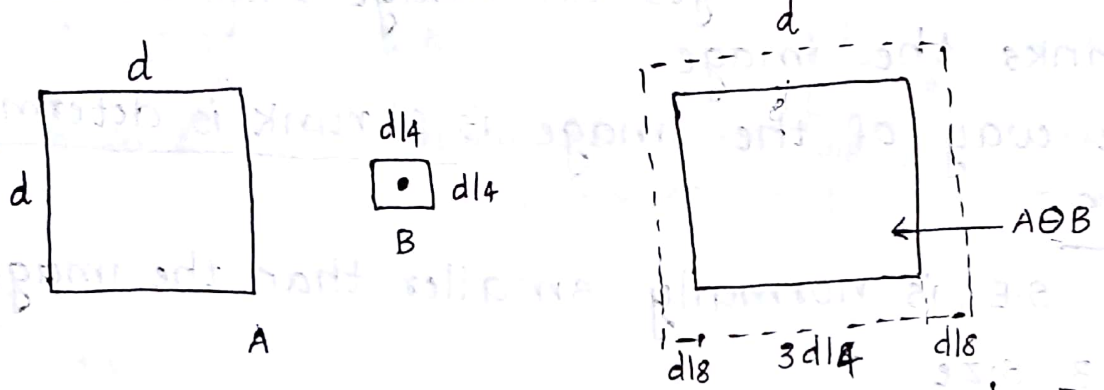
$(B)_z$: Translation

$$A \ominus B = \{ z \mid (B)_z \subseteq A \}$$

this eqn indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

- The stmt B has to be contained in A is equivalent to B not sharing any common elemen with the background

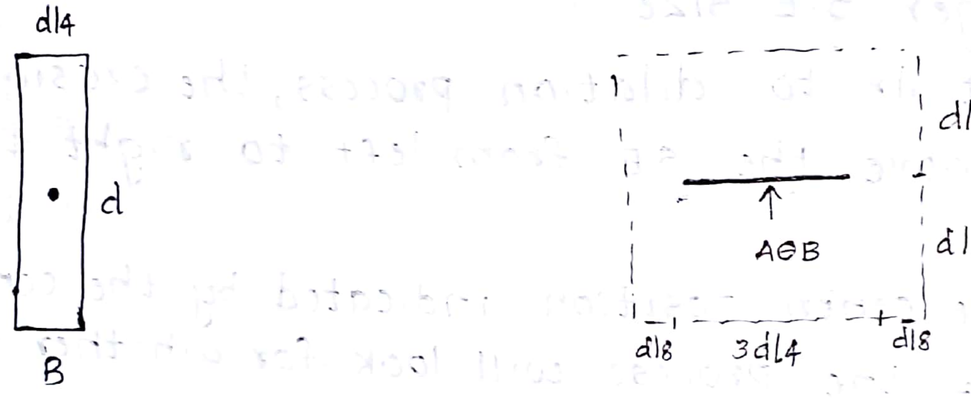
$$A \ominus B = \{ z \mid (B)_z \cap A^c = \phi \}$$



a) Set A

b) Square S.E B

c) Erosion of A by B



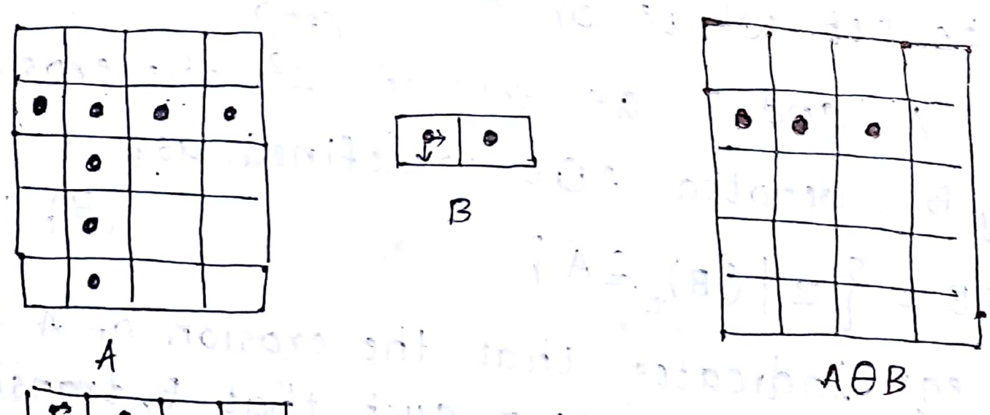
d) Elongated S.E

e) Erosion of A by B using this element.

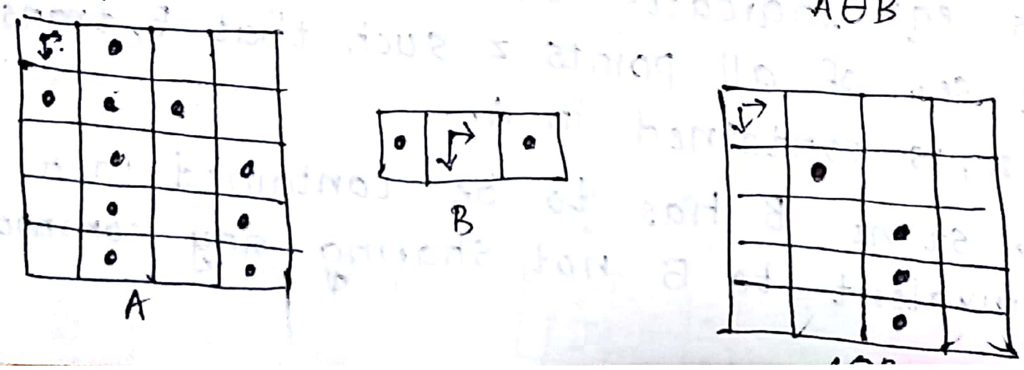
Fig: An illustration of Erosion.

Eg:

1)



2)



DILATION AND EROSION BASED OPERATIONS.

↳ Erosion & dilation can be combined to solve specific filtering tasks.

OPENING AND CLOSING.

◎ Opening $\circ$

↳ Opening is based on the morphological operations, erosion & dilation.

↳ Opening smoothes the inside of the object contour, breaks narrow strips and eliminates thin portions of the image.

↳ It is done by first applying erosion ① & then dilation ② operations on the image.

ED

↳ The opening of set A by structuring element B, denoted $A \circ B$, is defined as:

A : Input image

$$A \circ B = (A \ominus B) \oplus B$$

B : S.E

Properties: a) $A \circ B$ is a subset (subimage) of A.

b) If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$.

c) $(A \circ B) \circ B = A \circ B$.

◎ Closing $(\bullet)$

↳ The closing operation is the opposite of the opening opern.

↳ It is a dilation opern followed by an erosion operation.

↳ Closing operation fills the small holes and gaps in a single pixel object.

↳ It has the same effect of an opening opern, in that it smoothes contours and maintains shapes and sizes of objects.

↳ The closing of set A by S.E B, denoted $A \bullet B$, is defined as:

$$A \bullet B = (A \oplus B) \ominus B$$

A : i/p image

B : S.E

↳ Closing protects coarse structures, closes small gaps & rounds off concave corners.

~~Properties of Morphological Operation~~

~~↳ Let X and Y be two images. Let the image X be smaller than the image Y~~

~~ie., $X(m,n) \leq Y(m,n)$, $\forall$ the values of m & n.~~

↳ **Properties of closing operation.**

a) A is a subset (subimage) of $A \bullet B$

b) If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$

c) $(A \bullet B) \bullet B = A \bullet B$

PROPERTIES OF MORPHOLOGICAL OPERATION

1. Increasing.

↳ Let X & Y be two images. Let the image X be smaller than the image Y, ie, $X(m,n) \leq Y(m,n)$, for all the values of m & n.

$$X \ominus B \leq Y \ominus B$$

$$X \oplus B \leq Y \oplus B$$

$$X \circ B \leq Y \circ B$$

$$X \bullet B \leq Y \bullet B$$

B → S.E

↳ Erosion & dilation are said to be increasing operations. Since erosion and dilation are increasing operations, opening & closing are also increasing operations.

2. Expansivity

↳ The expansivity of the morphological operns:

$$X \ominus B \leq X$$

$$X \oplus B \geq X$$

$$X \circ B \leq X$$

$$X \bullet B \geq X$$

↳ Erosion is anti-expansive, becoz the o/p of the eroded image is not expansive.

↳ No. of ones present in the o/p eroded image is less compared to the original image.

↳ Dilation is expansive, the resultant image of dilation is expanded when compared to the i/p image.

↳ The opening opern is anti expansive and closing is expansive.

3. Duality

Property of duality are:

$$X \ominus B \leq (X^c \oplus \dot{B})^c$$

$$X \oplus B \leq (X^c \ominus B)^c$$

$$X \circ B \leq (X^c \bullet B)^c$$

$$X \bullet B \leq (X^c \circ B)^c$$

4. <u>Chain Rule</u>.

$$(X \ominus B_1) \ominus B_2 = X \ominus (B_1 \oplus B_2)$$

$$(X \oplus B_1) \oplus B_2 = X \oplus (B_1 \oplus B_2)$$

5. <u>Idempotency</u>

↳ Idempotency property is defined as follows:

$$(X \circ B) \circ B = X \circ B$$

$$(X \bullet B) \bullet B = X \bullet B$$

HIT-OR-MISS TRANSFORMATION $\otimes$

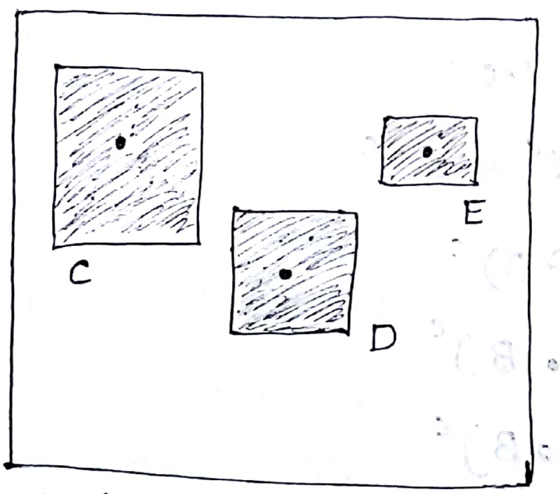
↳ The morphological hit-or-miss transform is a basic tool for shape detection.

↳ The objective is to find the location of one of the shapes.

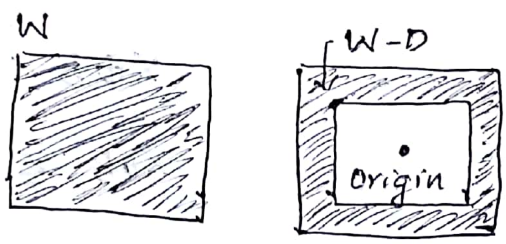
↳ Let the origin of each shape be located at its center of gravity.

↳ Let $\underset{S.E}{D}$ be enclosed by a small window W.

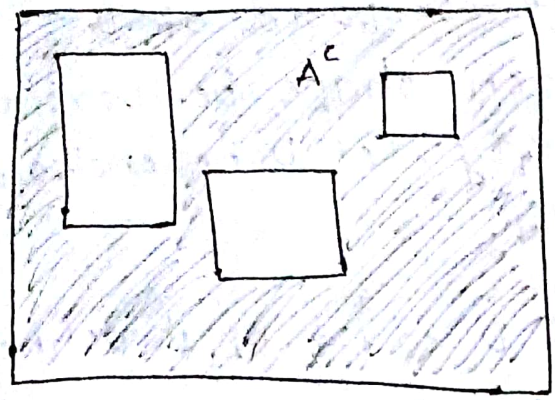
↳ The local bkgd of D w.r.t W is defined as the set difference (W − D)



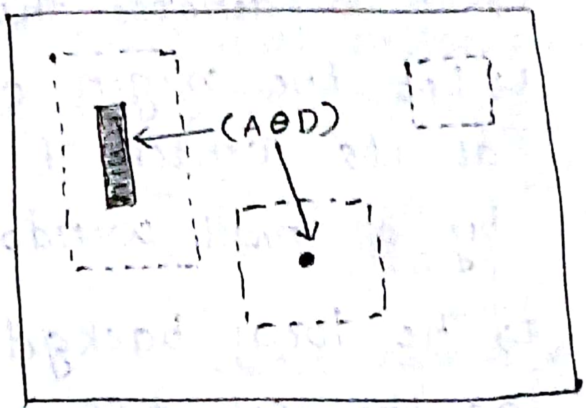
a) Set A



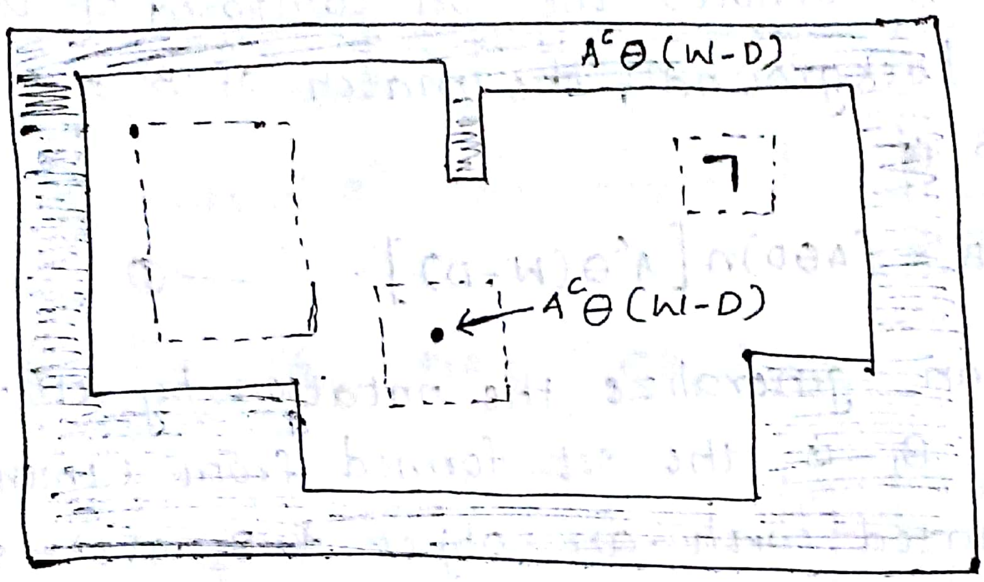
b) A Window W & local bkgd of D w.r.t W, (W−D)



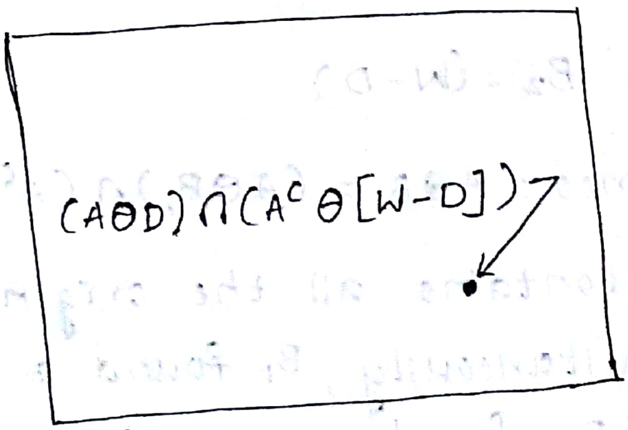
c) Complement of A.



d) Erosion of A by D.



e) Erosion of A^c by $(W-D)$



f) Intersection of (d) & (e).

↳ ~~If B denotes the set composed of D~~

↳ Let the origin of each shape be located at its center of gravity. Let D be enclosed by a small window W.

↳ The local backgd of D w.r.t W is defined as the set difference $(W-D)$

↳ If B denotes the set composed of D and its background, the match of B in A, denoted $A \circledast B$, is

$$A \circledast B = (A \ominus D) \cap [A^c \ominus (W-D)] \quad — ①$$

We can generalize the notation by letting $B = (B_1, B_2)$ where B_1 is the set formed from elements of B associated with an object & B_2 is the set of elements of B associated with the corresponding backgd.

$B_1 = D$ & $B_2 = (W-D)$

Eqn ① becomes: $A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$.

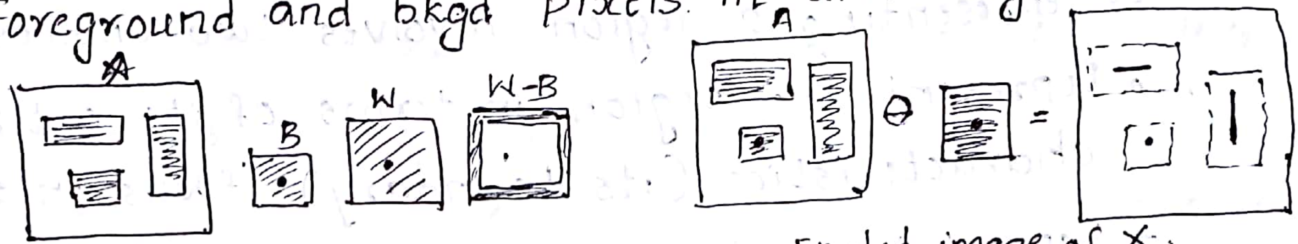
↳ Set $A \circledast B$ contains all the origin points at which simultaneously, B_1 found a match ("hit") in A and B_2 found a match in A^c.

↳ By using the definition of set differences and the dual relationship b/w erosion & dilation

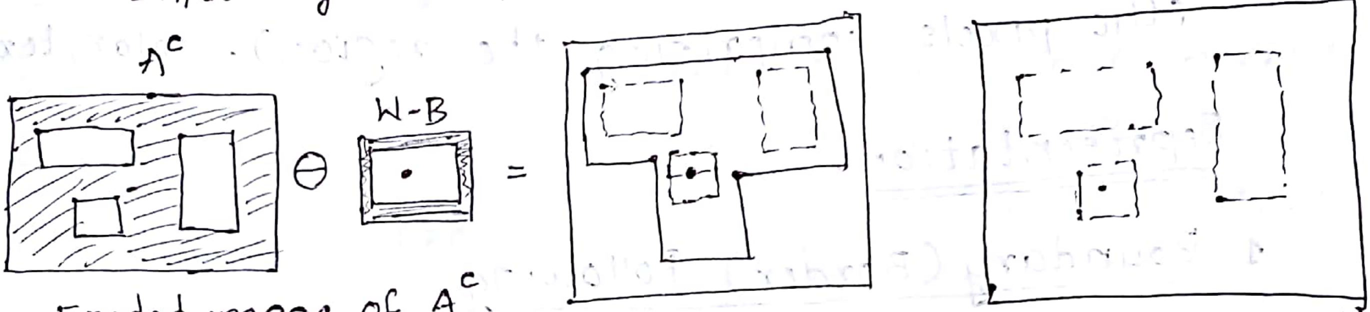
$$A \circledast B = (A \ominus B_1) - (A \oplus \hat{B}_2)$$

These three eqns as the <u>morphological hit or miss transform</u>.

↳ Hit or Miss tfrm is a general binary M.O that can be used in searching of particular patterns of foreground and bkgd pixels in an image.



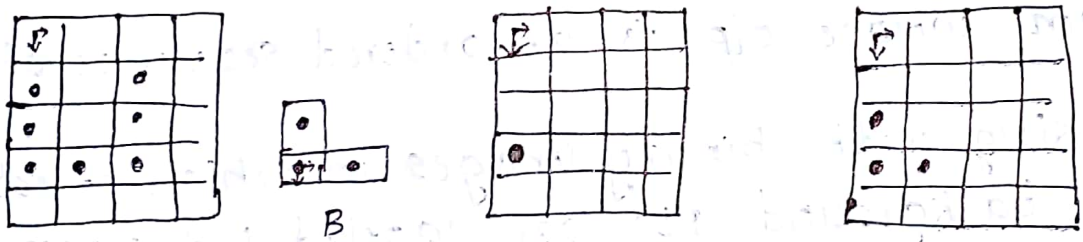
Input image A, S.E B, W, W-B. Eroded image of X.



Eroded image of A^c. Intersection result of above two results

---

Example : OPENING $A \circ B = (A \ominus B) \oplus B$.



A B $A \ominus B$. $(A \ominus B) \oplus B$.

REPRESENTATION AND DESCRIPTION

↳ Representing a region involves two choices:

- Represent the region in terms of its external characteristics (its boundary): focus on shape characteristics
- Represent it in terms of its internal characteristics (the pixels comprising the region). color, texture

Representation.

1. Boundary (Border) Following.

↳ Several of the algorithms require that the pts in the boundary of a region be ordered in a clockwise (or counterclockwise) direction.

↳ Algm whose o/p is an ordered sequence of pts.

(1) Working with binary images in which object and background pts are labeled 1 & 0 respectively.

(2) Images are padded with a border of 0s to eliminate the possibility of an object merging with the image border.

↳ The approach is extended to multiple, disjoint regions by processing the regions individually.

Given a binary region R or its boundary, an algm for following the border of R, or the given boundary, consists of following steps:

Boundary-following algorithm

1. Let the starting point b_0, be the uppermost, leftmost point in the image that is labelled 1. Denote by c_0 the west neighbor of b_0. Clearly, c_0 always is a bkgd pt.

   Examine the 8-neighbors of b_0, starting at c_0 and proceeding in a clockwise direction.

   Let b_1 denote the first neighbor encountered whose value is 1 and let c_1 be the point immediately preceding b_1 in the sequence. Store the locations of b_0 and b_1 for use in step5.

2. Let $b = b_1$ and $c = c_1$

3. Let the 8-neighbors of b, starting at c and proceeding in a clockwise direction, be denoted by $n_1, n_2, \cdots, n_8$. Find the first n_k labeled 1.

4. Let $b = n_k$ and $c = n_{k-1}$.

5. Repeat Step3 & 4 until $b = b_0$ and the next boundary point found is b_1.

   The Sequence of b points found when the algm stops constitutes the set of ordered boundary points.

↳ c in step4 always is a backgd pt becoz n_k is first 1-valued pt found in the clockwise scan.

↳ This algm sometimes is referred to as the Moore boundary tracking algorithm.

Scanned by CamScanner

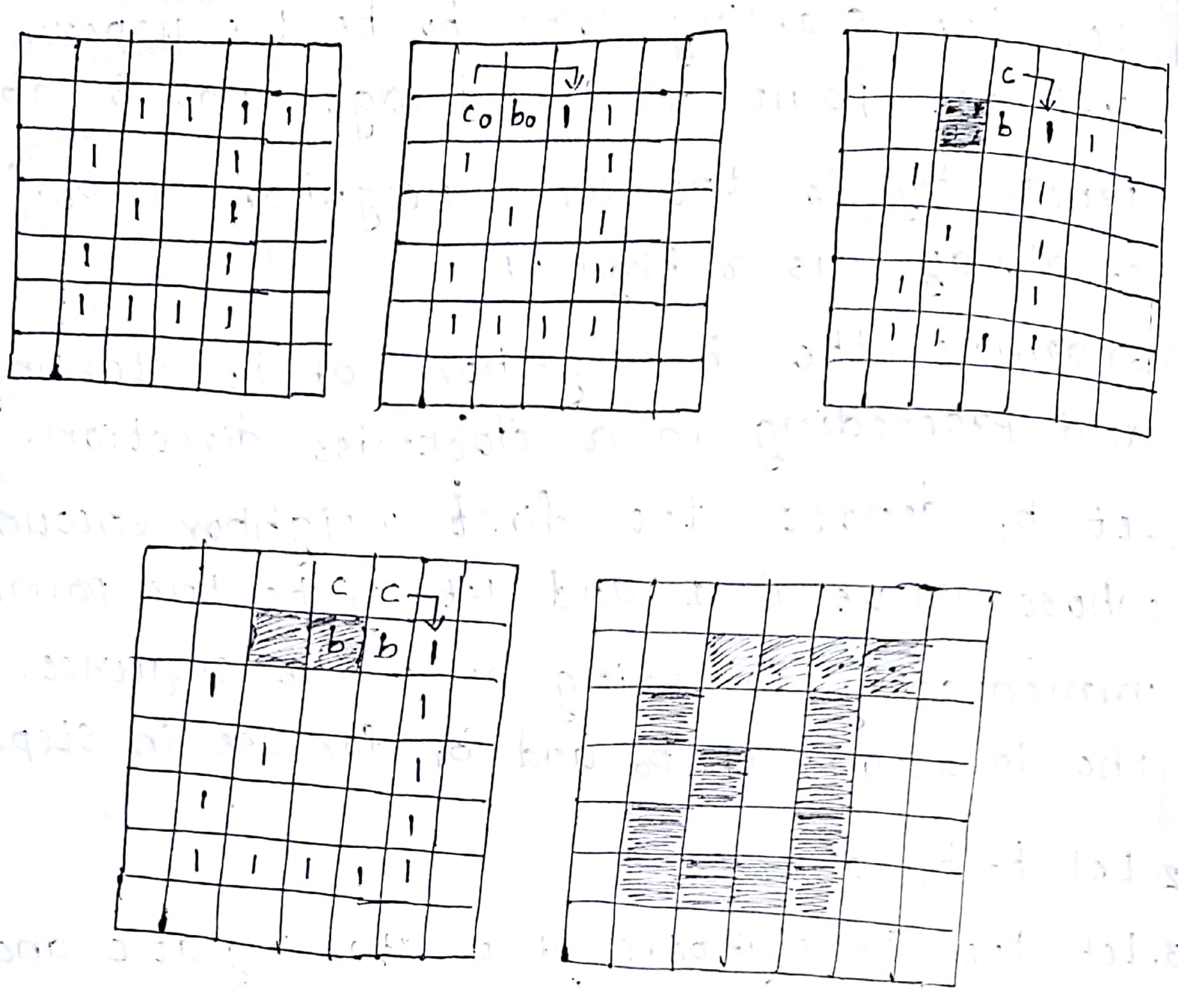


Illustration of the first few steps in the boundary-following algorithm.

↳ The boundary following algm works equally well if a region, rather than its boundary ie, the procedure extracts the outer boundary of a binary region.

↳ If the objective is to find the boundaries of holes in a region (these are called the inner boundaries of the region), a simple approach is to extract the holes and treat them as 1-valued regions on bkgd of 0s.

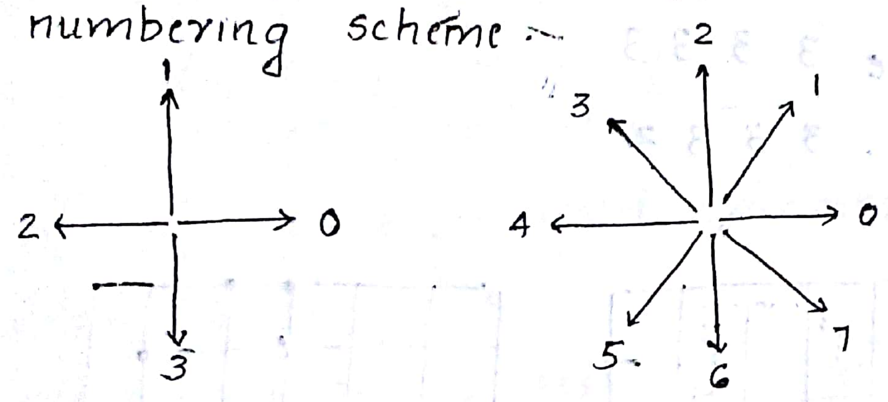
↳ Applying the boundary following algm to these regions will yield the inner boundaries of the original region.

2. CHAIN CODES

↳ Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction.

↳ This representation is based on 4-or-8 connectivity of the segments.

↳ The direction of each segment is coded by using a numbering scheme :-



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

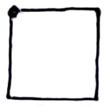
↳ Digital images usually are acquired and processed in a grid format with equal spacing in the x and y directions.

↳ Chain code can be generated by following a boundary in a clockwise direction and assigning a direction to the segments connecting every pair of pixels.

Disadv
↳ 1. The resulting chain tends to be quite long
2. Any small disturbances 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 boundary.

↳ Chain code of a boundary depends on the starting point.

↳ The code can be normalized w.r.t to the starting point by a straightforward procedure.

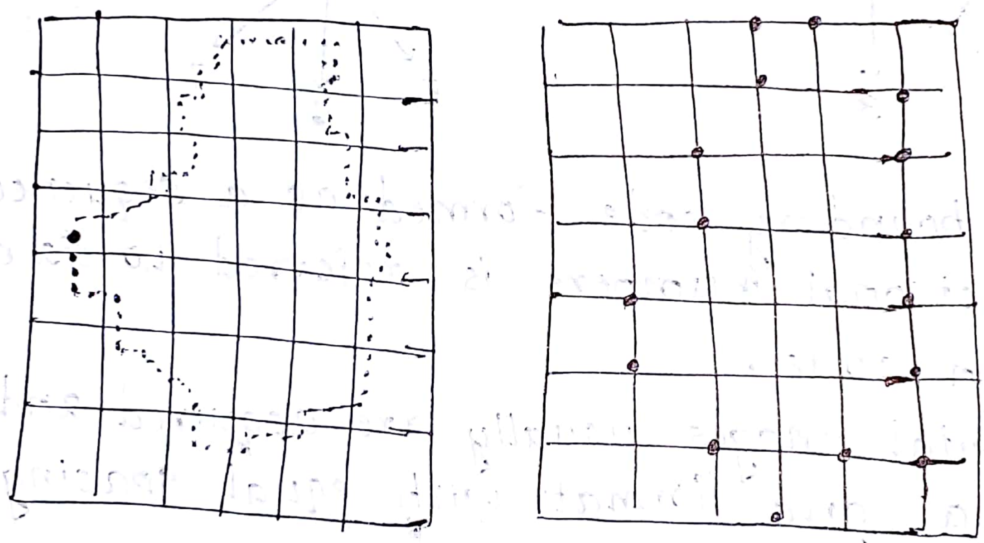


order 4

chain code 0 3 2 1

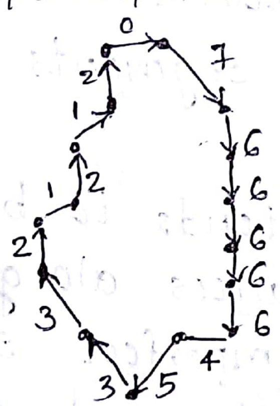
difference 3 3 3 3

shape no : 3 3 3 3



a) Digital boundary with resampling grid superimposed

b) Result of resampling.



8 directional chain coded boundary.

3. POLYGONAL APPROXIMATIONS

↳ A digital boundary can be approximated with arbitrary accuracy by a polygon.

↳ For a closed boundary, approxi^n becomes exact when no. of points in the boundary segments of the polygon is equal to no. of points in the boundary so that each pair of adj. pts defines segment of polygon.

↳ The goal of polygonal approxi^n is to capture the essence of shape in a given boundary using fewest no. of segments.

↳ Approx^n techqs of modest complexity are well suited for IP tasks.

↳ Most powerful is representing a boundary by a Minimum-perimeter polygon (MPP).

An obj. boundary → Boundary enclosed by cells → Min. perimeter polygon obtd by allowing the boundary to shrink → The vertices of polygon are created by the corners of the inner & outer walls of gray region.

↳ The size of the cells determines the accuracy of the polygonal approximation.

↳ If the size of each square cell corresponds to a pixel in the boundary, the error in each cell b/w the boundary & MPP approx^n atmost $\sqrt{2}d$, where d is the minimum possible distance b/w pixels.

↳ The cellular approach reduces the shape of the object enclosed by the original boundary.

↳ Boundary consists of 4-connected straight line segments.

↳ Suppose we traverse this boundary in counter (anti) clockwise direction. Every turn encountered in the traversal will be either a convex or concave vertex, with the angle of a vertex being an interior angle of the 4-connected boundary.

MPP Algorithm

↳ The set of cells enclosing a digital boundary is called a cellular complex.

↳ The boundaries under consideration are not self intersecting, which leads to simply connected cellular complexes.

↳ White (W) and Black (B) denote convex and mirrored concave vertices.

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.

↳ Consider the triplet of points, (a, b, c) and let the coordinates of these points be $a = (x_1, y_1)$ $b = (x_2, y_2)$ and $c = (x_3, y_3)$. If we arrange these pts as the rows of the matrix.

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$$

then it follows from elementary matrix analysis that:

$$\det(A) = \begin{cases} > 0 & \text{if } (a, b, c) \text{ is a counterclockwise sequence.} \\ = 0 & \text{if the points are collinear.} \\ < 0 & \text{if } (a, b, c) \text{ is a clockwise sequence} \end{cases}$$

ADVANTAGE

- Adv of MMPs for representing a boundary.
- MPPs perform boundary smoothing.

OTHER POLYGONAL APPROXIMATION APPROACHES

1. Merging techniques.
2. Splitting techq.

1. <u>Merging techq.</u>

↳ it is based on avg. error or other criteria have been applied to the pblm of polygonal approxn.

↳ To Merge points along a boundary until the least square error line fit of the points merged so far exceeds a preset threshold.

↳ When this condition occurs, the parameters of the line are stored, the error is set to 0, & the procedure is repeated, merging new points along the boundary until the error again exceeds the threshold.

↳ At the end of the procedure the intersect[ns] of adj. line segments form the vertices of polygon.

2. <u>Splitting techq.</u>

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

↳ A requirement that the maximum ⊥r distance from a boundary segment to the line joining its two end pts not exceed a preset threshold.

↳ If it does, the pt having the greatest distance from the line becomes a vertex, thus sub dividing the initial segment into subsegments.

<u>Adv</u>

- Seeking prominent inflection points.

BOUNDARY SEGMENTS

↳ Decomposing a boundary into segments is often useful.

↳ Decomposition reduces the boundary's complexity and thus simplifies the description process.
(length, boundary, eccentricity, curvature)

↳ Convex hull of the region enclosed by the boundary is a powerful tool for robust decomposition of the boundary.

↳ The convex hull H of an arbitrary set S is the smallest convex set ~~containing~~ S.

↳ The set difference H - S is called the convex deficiency D of the set S.

↳ Digital boundaries tend to be irregular becoz of digitization, noise and variations in segmentⁿ. These effects results in convex deficiencies that have small components scattered randomly throughout the boundary.

↳ There are no. of ways to do so :

a) To traverse the boundary and replace the coordinates of each pixel by the avg coordinates of K of its neighbors along the boundary.
Limitation of this approach : Time consuming & difficult to control.

b) Use a polygonal approxⁿ prior to finding the convex deficiency of a region.