GLA UNIVERSITY

DIGITAL IMAGE PROCESSING

By:
Dr. Ankush Agarwal

Outline

- Morphological Image Processing
  - Introduction, Logical Operations involving Binary Images,
  - Dilation and Erosion, Opening and Closing, The Hit-or-Miss Transformation,
  - Morphological Algorithms – Boundary Extraction, Region Filling, Extraction of Connected Components, Convex Hull, Thinning, Thickening
- Image Segmentation
  - Point, Line & Edge detection, Thresholding, Region-based Segmentation,
  - Region Extraction - Pixel Based Approach & Region Based Approach,
  - Edge and Line Detection - Basic Edge Detection, Canny Edge Detection,
  - Edge Linking - Hough Transform.
- Representation & Description
  - Representation - Boundary Following, Chain Codes,
  - Boundary Descriptors - Shape Numbers

MORPHOLOGY

Morphology

- Mathematical tool for processing shapes in image, including boundaries, skeletons, convex hulls, etc
- Morphological operations are typically applied to remove imperfections introduced during segmentation, and so typically operate on binary images

Morphology

- Mathematical tool for processing shapes in image, including boundaries, skeletons, convex hulls, etc
- Morphological operations are typically applied to remove imperfections introduced during segmentation, and so typically operate on binary images

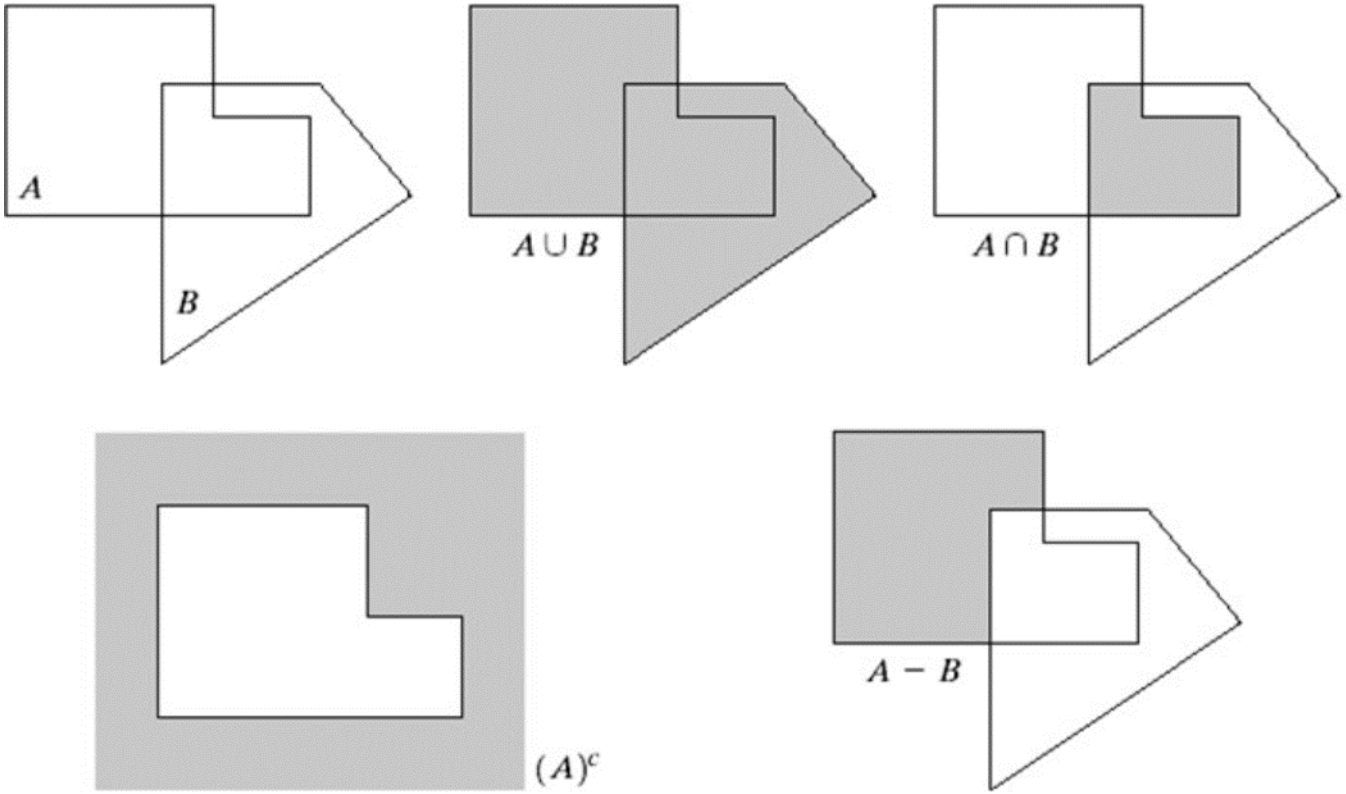
Set Theory

- Set (Ω): A collection of objects (elements)
- Membership ($\in$): If ω is an element of a set Ω, we can write $\omega \in \Omega$
- Subset ($\subset$): Let A, and B are two sets., If for every $a \in A$, we also have $a \in B$, then the set A is a subset of B, that is, $A \subset B$
  - If $A \subset B$ and $B \subset A$, then $A = B$
- Empty set ($\varnothing$)
- Complement: If $A \subset \Omega$, then its complement set $A^c = \{\omega \mid \omega \in \Omega, \text{ and } \omega \notin A\}$

Set Theory

- Union ($\cup$): $A \cup B = \{\omega | \ \omega \in A \text{ or } B\}$
- Intersection ($\cap$): $A \cap B = \{\omega | \ \omega \in A \text{ and } B\}$
- Set difference (-): $B \backslash A = B \cap A^c$
  - Note that $B\text{-}A \neq A\text{-}B$
- Disjoint sets: A and B are disjoint (mutually exclusive) if $A \cap B = \varnothing$

Example sets operations



Reflection and Translation

- Reflection
  - The reflection of a set B, denoted by $\hat{B}$, is defined as

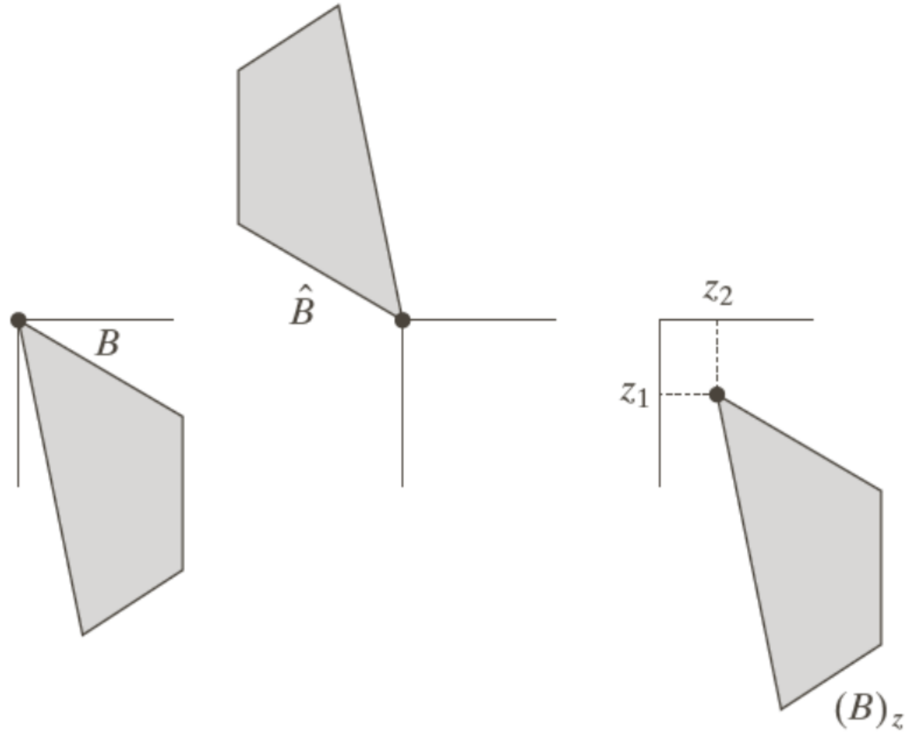
$$\hat{B} = \{w | w = -b, for\ b \in B\}$$

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

$$(B)_z = \{c | c = b + z, for\ b \in B\}$$

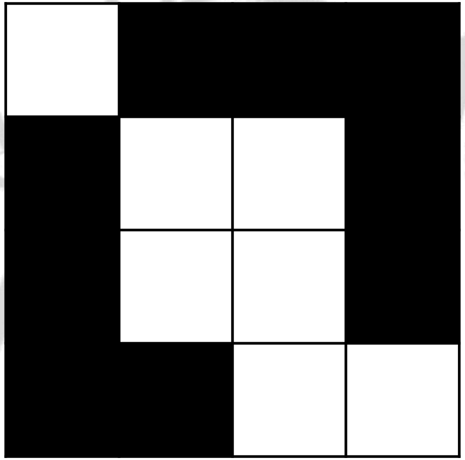
Reflection and Translation

- Eg:

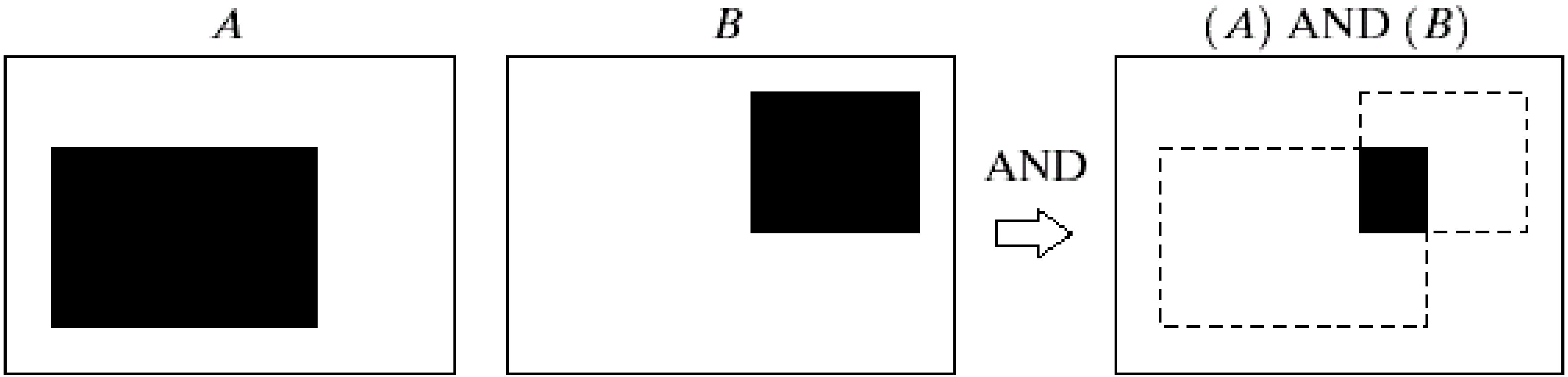


Binary Image

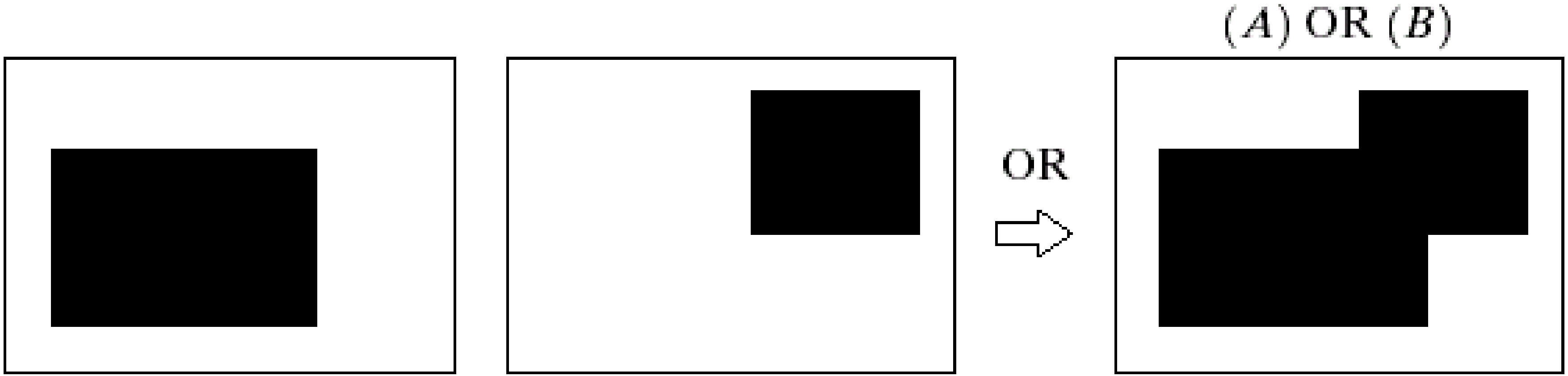
- Binary image
  - bi-valued function of x and y
- Morphological theory views
  - binary image as a set of its foreground (1-valued) pixels



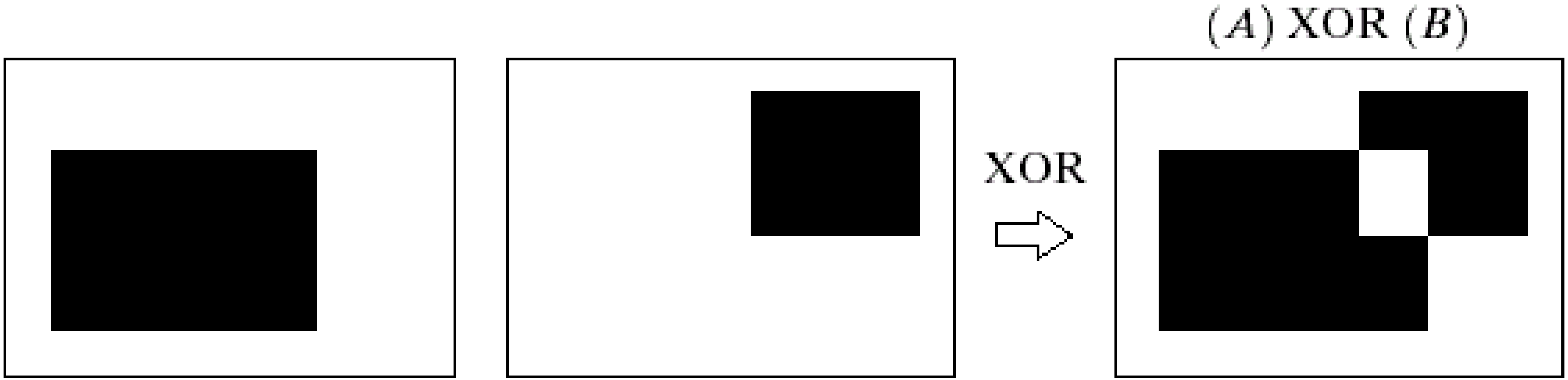
Logic operations between binary images



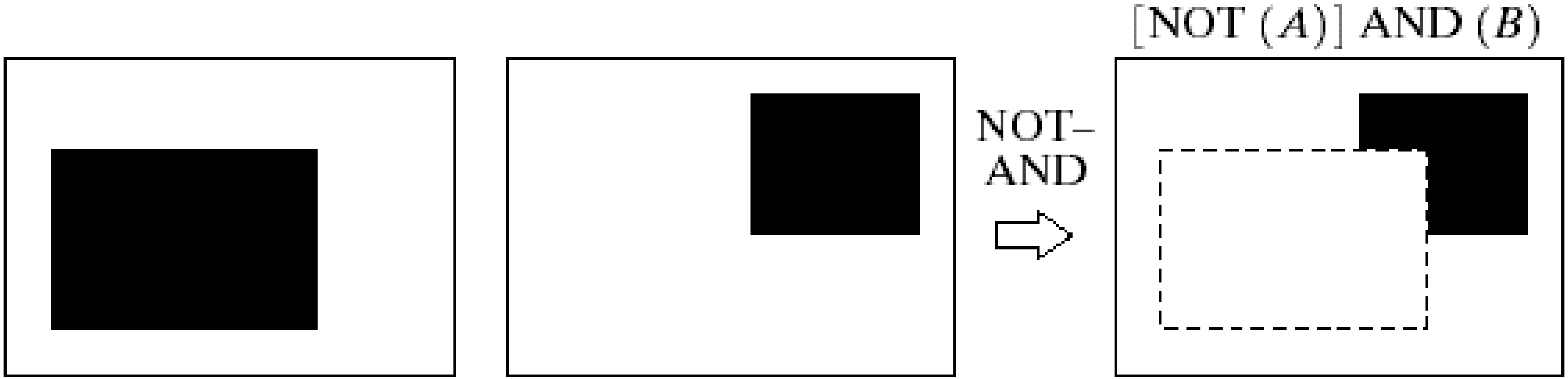
Logic operations between binary images



Logic operations between binary images

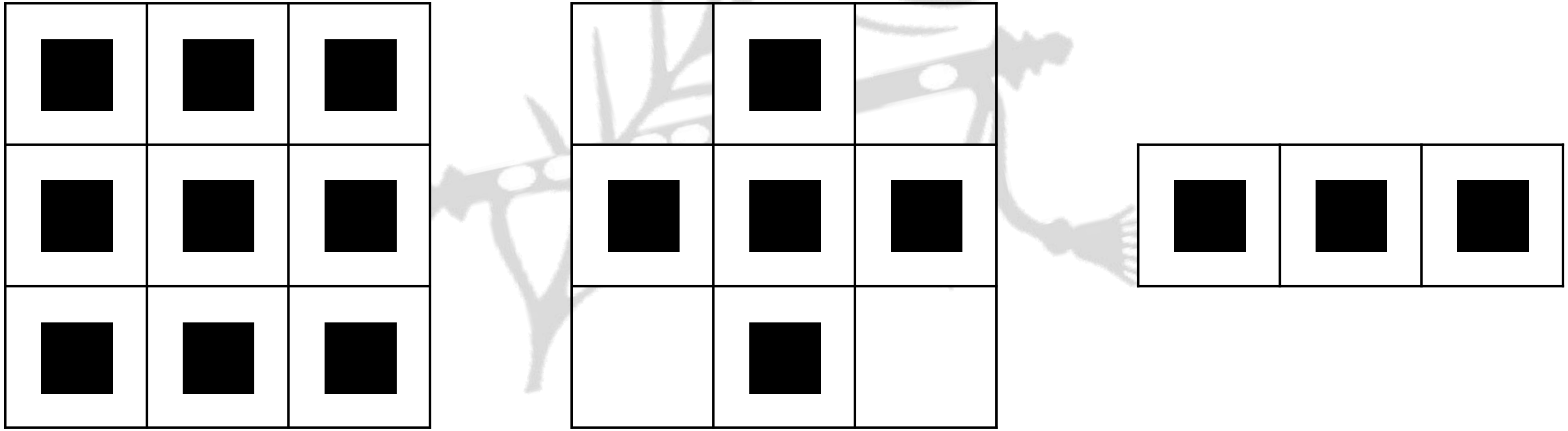


Logic operations between binary images



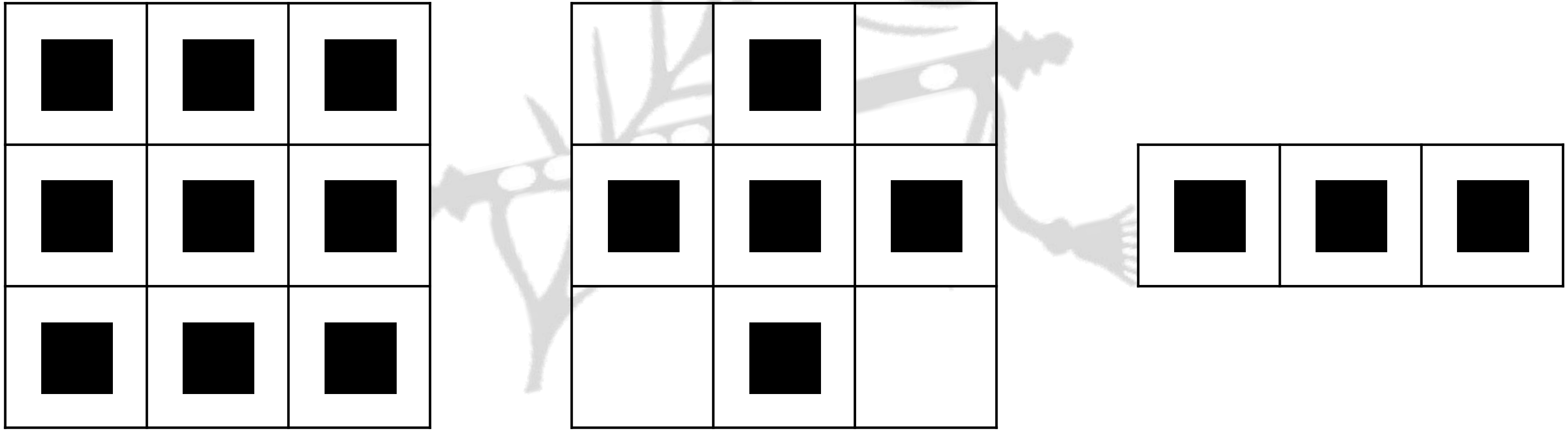
Basic components in Morphology

- Every operation has two elements
  - Input Image
  - Structuring element
- The results of the operation mainly depends upon the structuring element chosen



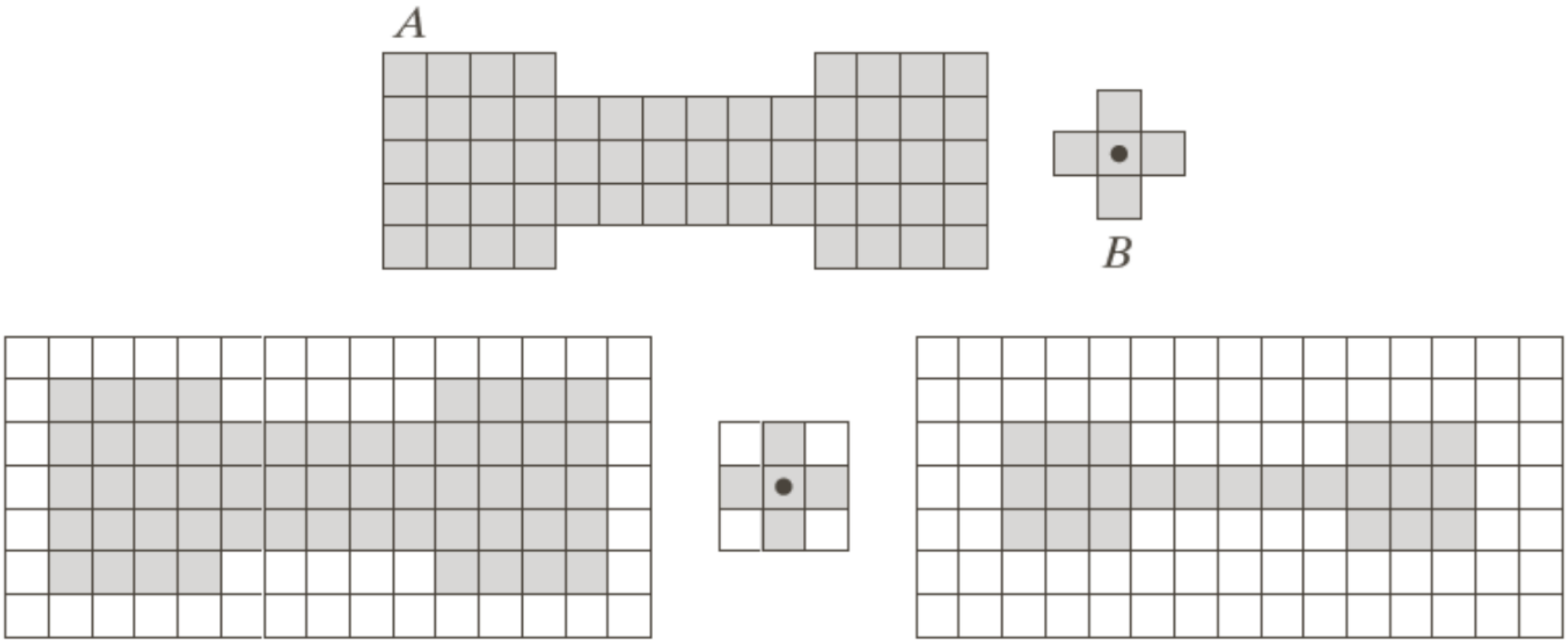
Structuring Elements

- Small sets or sub-images used to analyze an image for properties of interest
- Structuring elements can be any size and any shape



Structuring Elements

- Eg:



Fundamental Operations

- Fundamentally, morphological image processing is like spatial filtering
- The structuring element is moved across every pixel in the original image to give a new value of a pixel in processed image
- The value of this pixel depends on the operation performed
- There are two basic morphological operations
  - Dilation
  - Erosion

Dilation and Erosion

Dilation

- Dilation is an operation that grows or thickens objects in a binary image
- The specific manner of this thickening is controlled by a shape referred to as a structuring element
- The structuring element is translated throughout the domain of the image to see where it overlaps with 1-valued pixels
- The output image is 1 at each location of the origin such that the structuring element overlaps at least one 1-valued pixel in the input image

Dilation

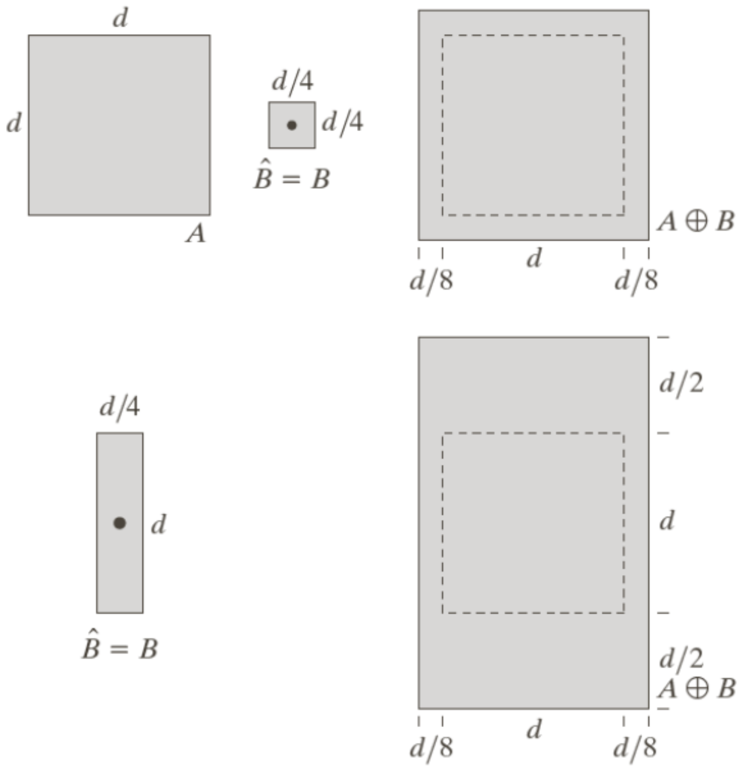
- The dilation of I and S is denoted by I $\oplus$ S

$$I \oplus S = \{z \mid (\hat{S})_z \cap I \neq \phi\}$$

- Theoretical way of generation:
  - Obtain the reflection of S about its origin
  - Shift this reflection by z
  - Dilation of I by S is the set of all structuring element origin locations where the reflected and translated S overlaps at least some portion of I

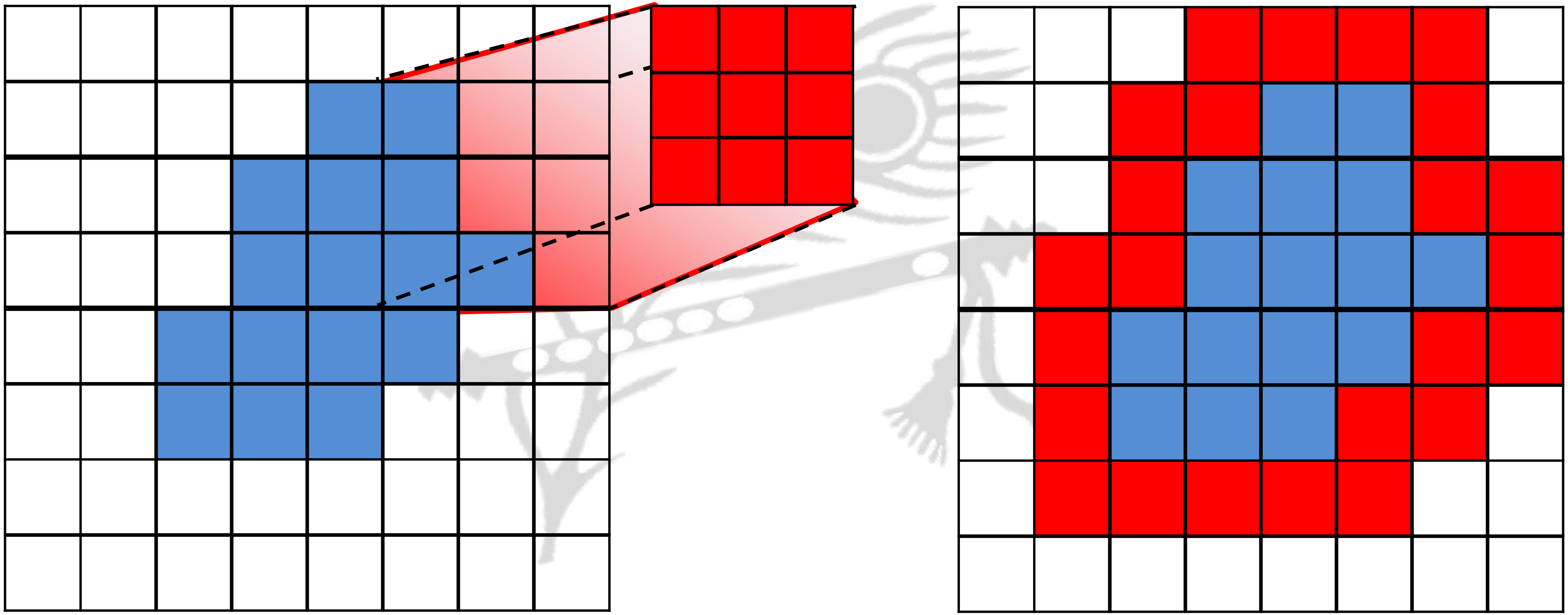
Dilation

- Eg:



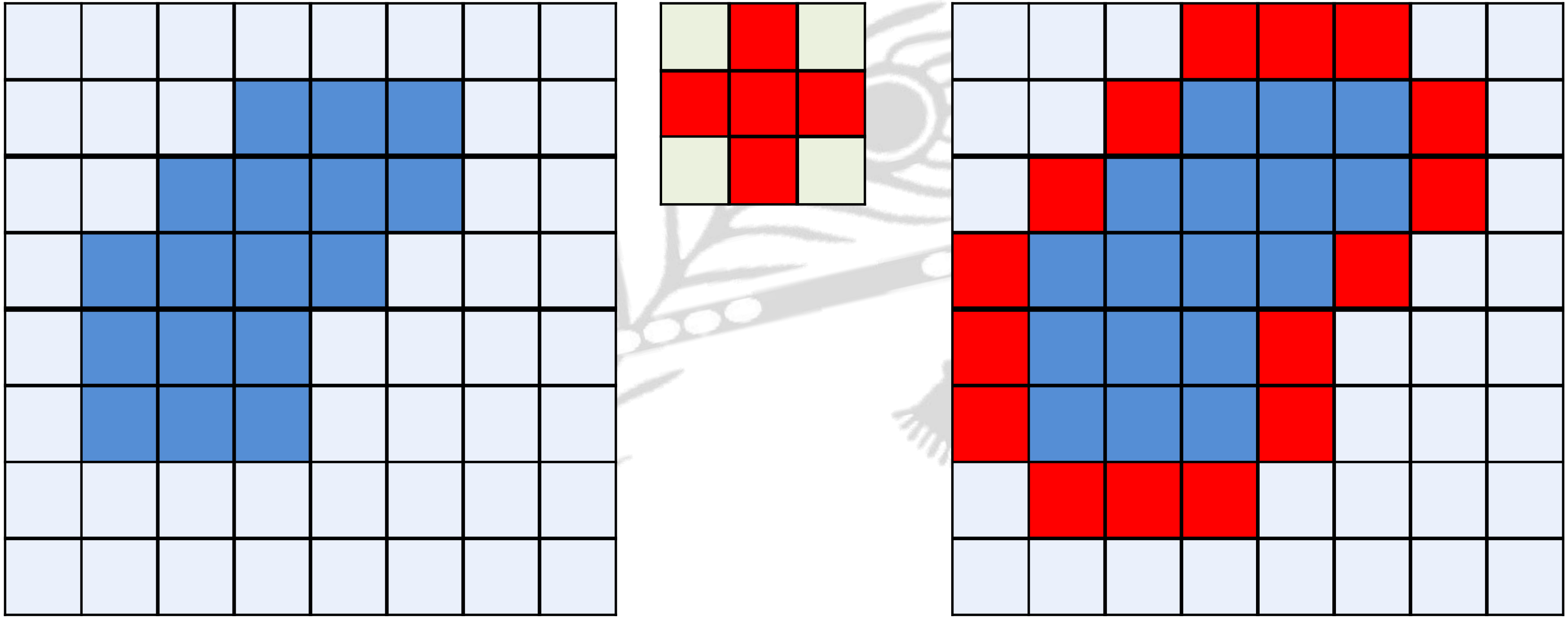
Dilation

- Eg:



Dilation

- Eg:



What is Dilation for…?

- Dilation can repair breaks

- Dilation can repair intrusions

Properties of Dilation

- Dilation is commutative

$$A \oplus B = B \oplus A$$

- Dilation is associative

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

- Dilation is invariant to translation

$$A_h \oplus B = (A \oplus B)_h$$

Erosion

- The erosion of I by S, denoted $I \ominus S$

$$I \ominus S = \{z \mid (S)_Z \subseteq I\}$$

  – The set of all points z such that, S translated by z, is contained by I

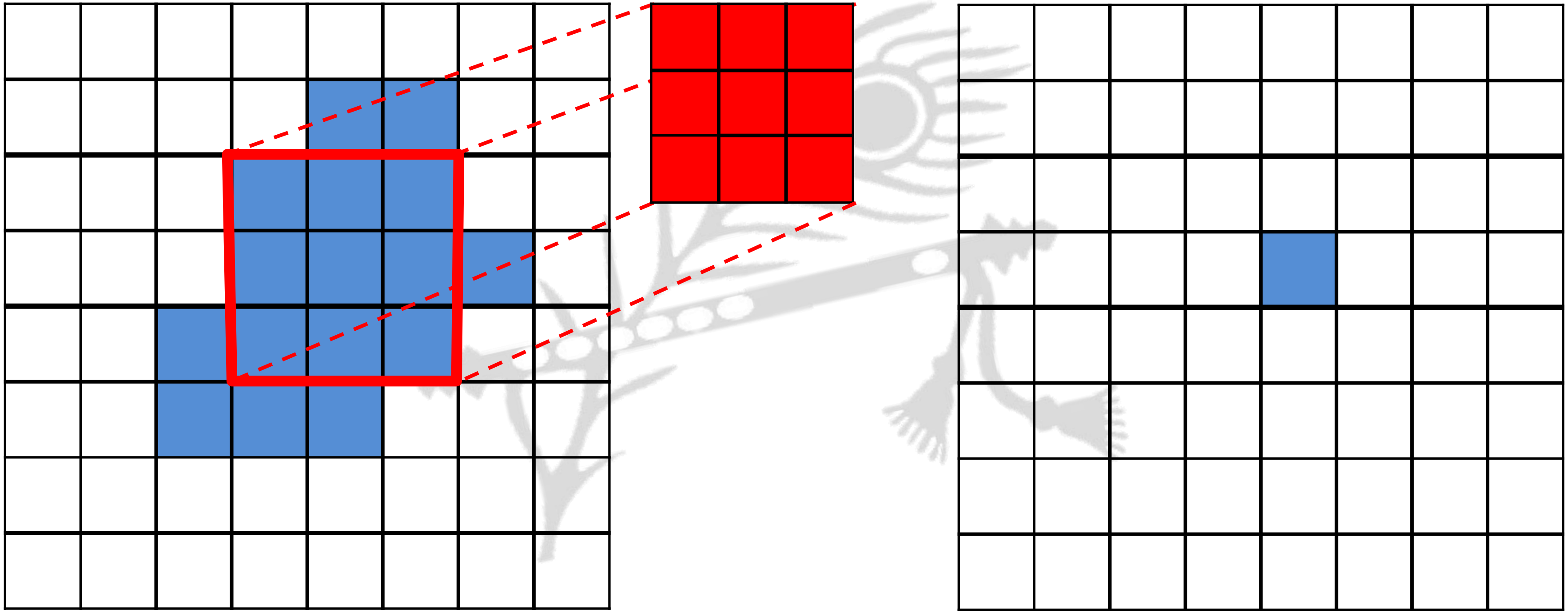
$$I \ominus S = \{z \mid (S)_Z \cap I^c = \emptyset\}$$

  – In other words, erosion of I by S is the set of all structuring element origin locations where the translated S has no overlap with the background of I

- Erosion “shrinks” or “thins” objects in a binary image

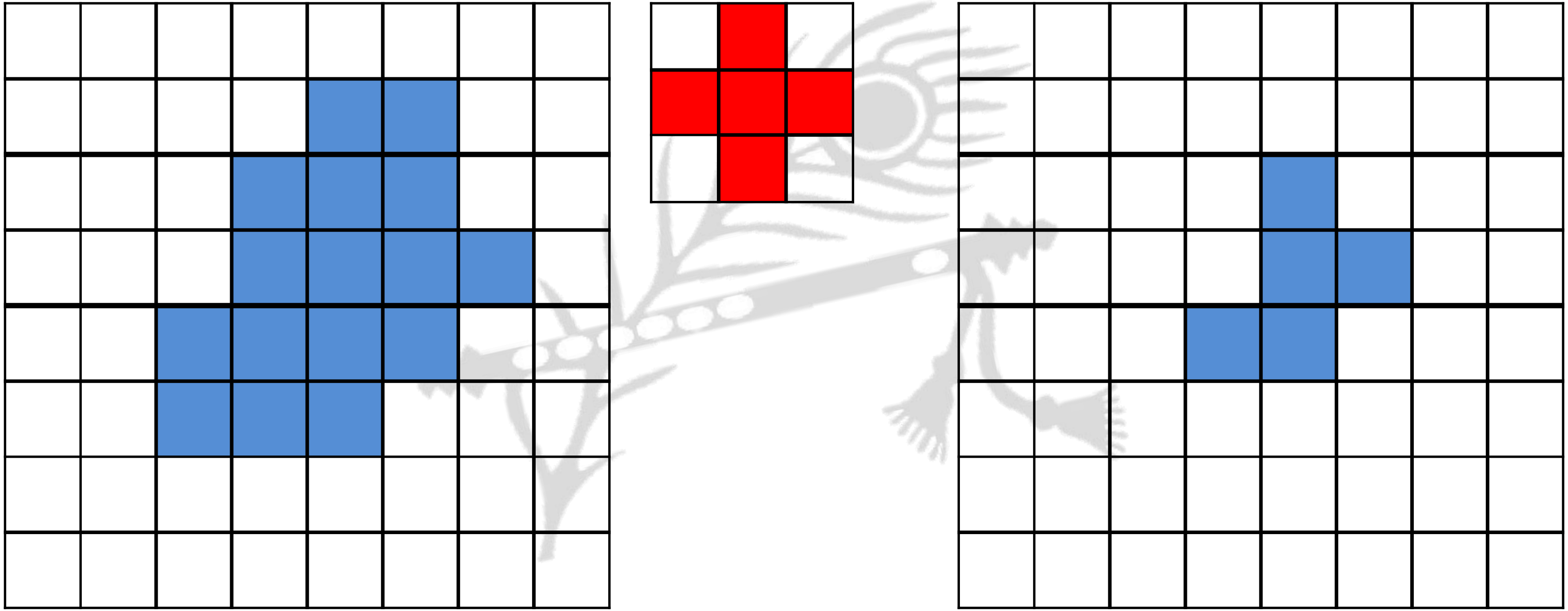
Erosion

- Eg:



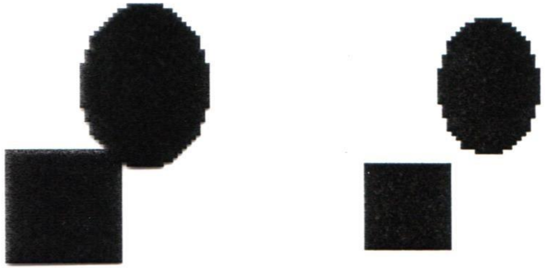
Erosion

- Eg:

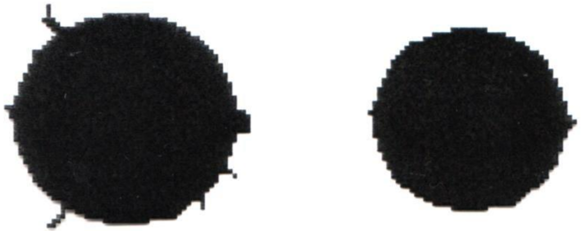


What is Erosion for…?

- Erosion can split apart joined objects



- Erosion shrinks objects and removes random outer edges



Dilation

- Binary image

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

← Image (I)

Structure Element (S) →

	-1	0	1
-1			
0		X	
1			

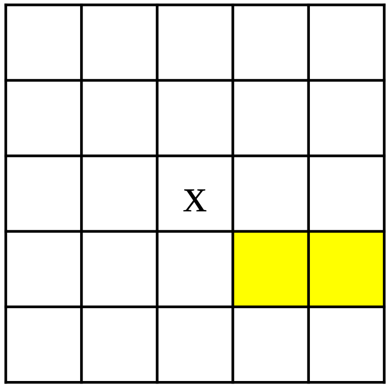
I = {(2,2), (3,2), (3,3), (4,3), (4,4), (5,4)}

U = {(0,0), ..., (7,7)}

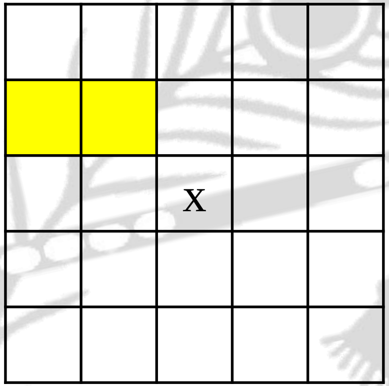
S = {(-1,-1), (0,-1)}

Dilation

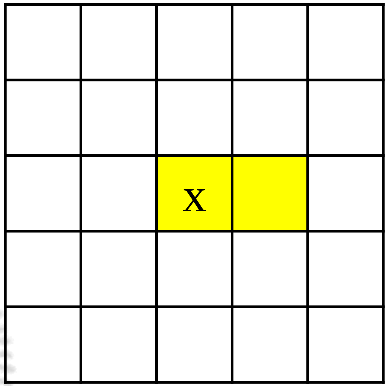
- Reflection and Translation operations



$I = \{(1,1), (1,2)\}$



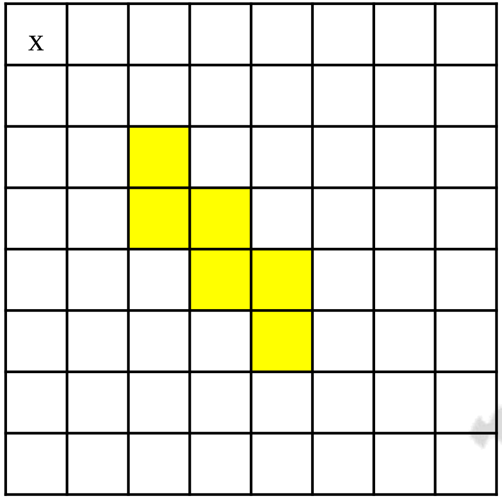
$\hat{I} = \{(-1,-1), (-1,-2)\}$

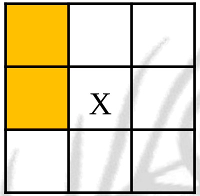


$I_{(-1,-1)} = \{(0,0), (0,1)\}$

Dilation (by coordinate system)

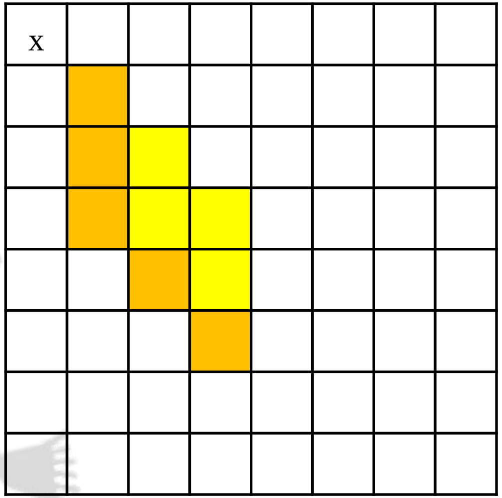
- Eg:





$S = \{(-1,-1), (0,-1)\}$

$$I \oplus S = \{p \mid p = i + s, i \in I, s \in S\}$$



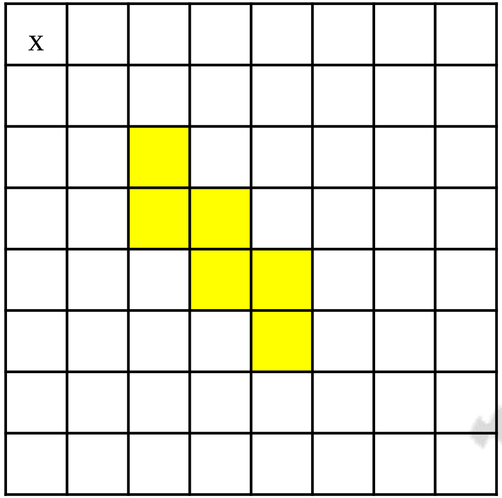
$I = \{(2,2), (3,2), (3,3), (4,3), (4,4), (5,4)\}$

$$I \oplus S = \left\{ \begin{matrix} (1,1),(2,1),(2,2),(3,2),(3,3),(4,3) \\ (2,1),(3,1),(3,2),(4,2),(4,3),(5,3) \end{matrix} \right\}$$

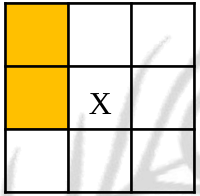
$$= \{(1,1),(2,1),(2,2),(3,1),(3,2),(3,3),(4,2),(4,3),(5,3)\}$$

Dilation (another definition)

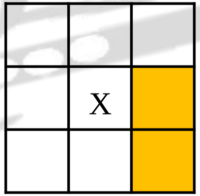
- Eg



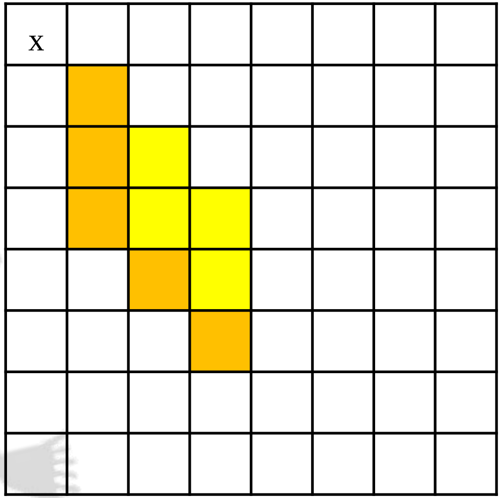
I



S = {(-1,-1), (0,-1)}



S = {(1,1), (0,1)}

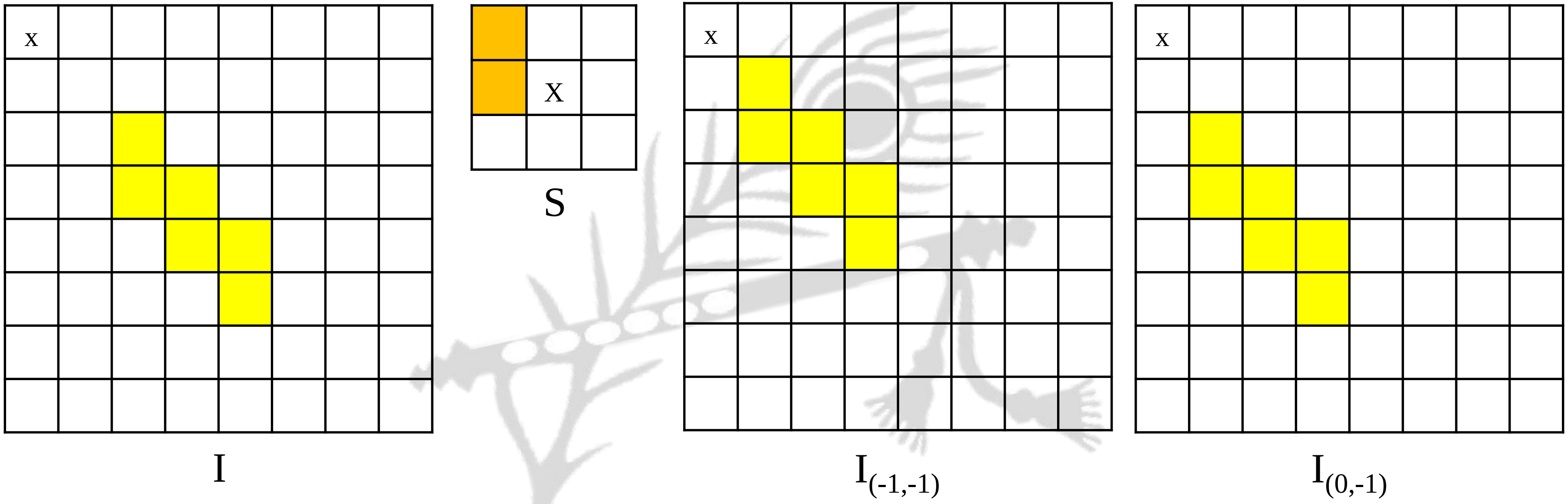


$$I \oplus S = \{p \mid [(\hat{S})_p \cap I] \neq \phi\}$$

$$= \{p \mid [(\hat{S})_p \cap I] \subseteq I\}$$

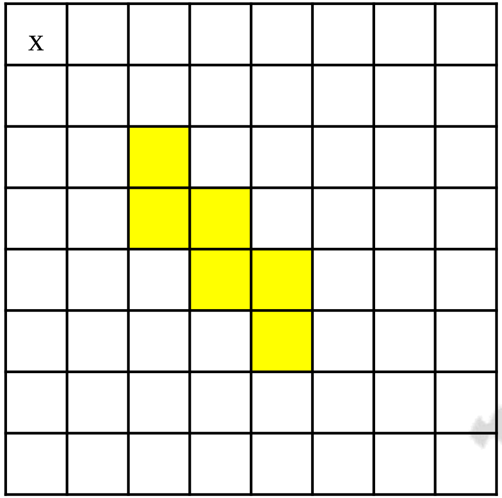
Dilation (as Union of object translation)

- Eg

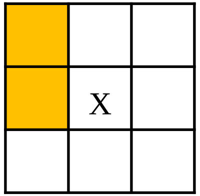


Dilation (as Union of object translation)

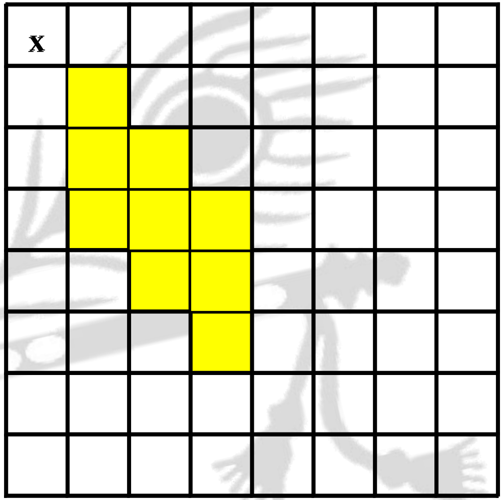
- Eg



I



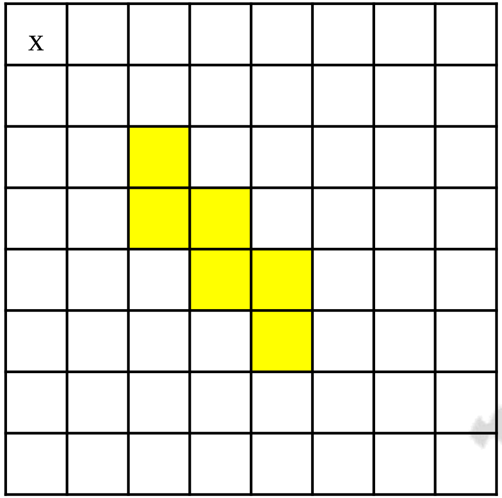
S

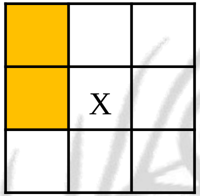


$$I \oplus S = \bigcup_{s \in S} I_s$$

Erosion (by coordinate system)

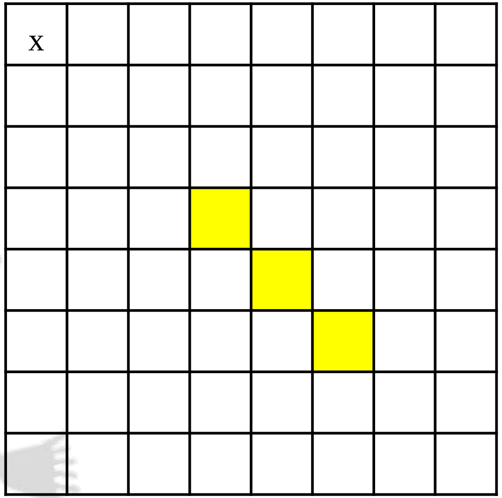
- Eg:





$S = \{(-1,-1), (0,-1)\}$

$$I \Theta S = \{p \mid p + s \in I, s \in S\}$$



$I = \{(2,2), (3,2), (3,3), (4,3), (4,4), (5,4)\}$

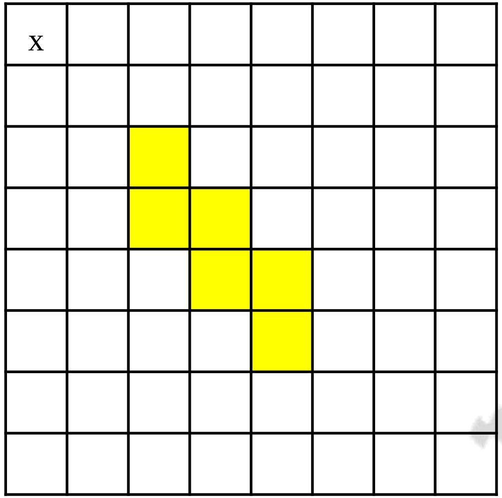
$$(3,3) + \{(-1,-1),(0,-1)\} = \{(2,2),(3,2)\} \in I$$

$$(4,4) + \{(-1,-1),(0,-1)\} = \{(3,3),(4,3)\} \in I$$

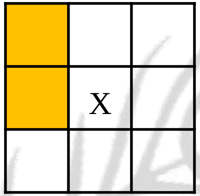
$$(5,5) + \{(-1,-1),(0,-1)\} = \{(4,4),(5,4)\} \in I$$

Erosion (another definition)

- Eg

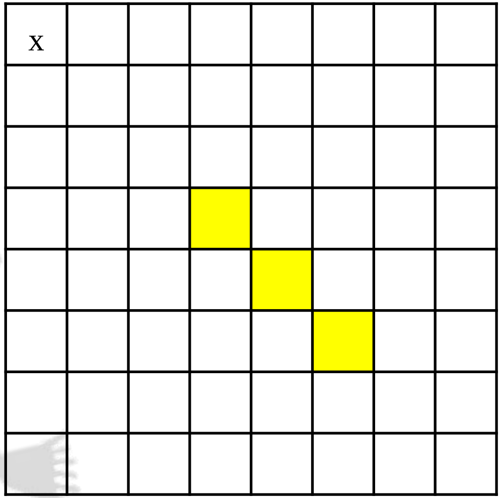


I



S = {(-1,-1), (0,-1)}

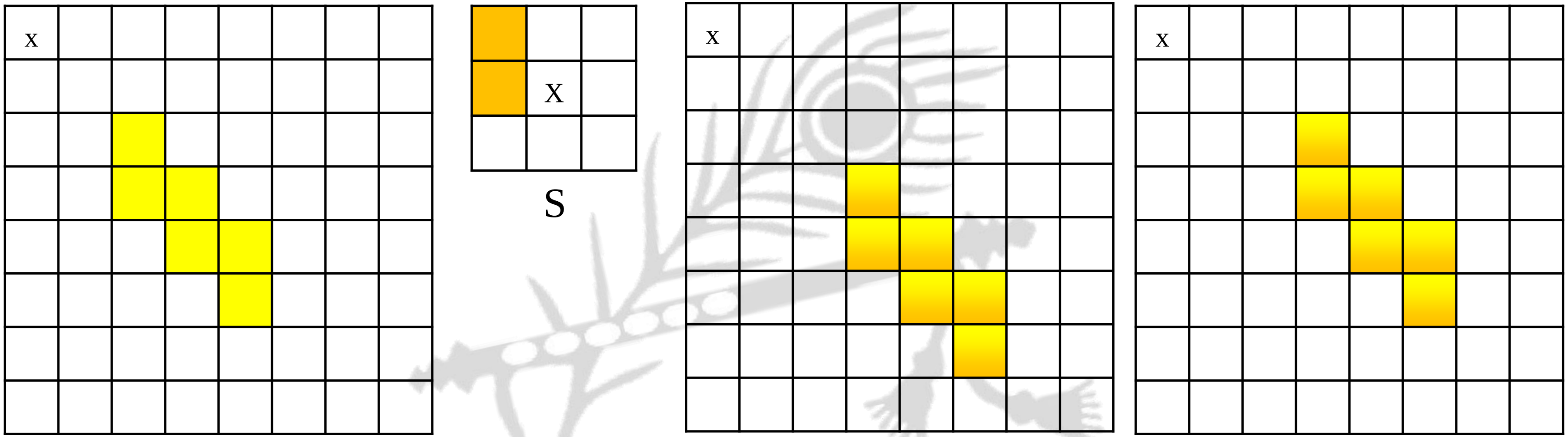
$$I \Theta S = \{p \mid (S)_p \cap I^c = \phi\}$$
$$= \{p \mid [(S)_p] \subseteq I\}$$



$(3,3),(4,4),(5,5)$

Erosion (as Intersection of object translation)

- Eg



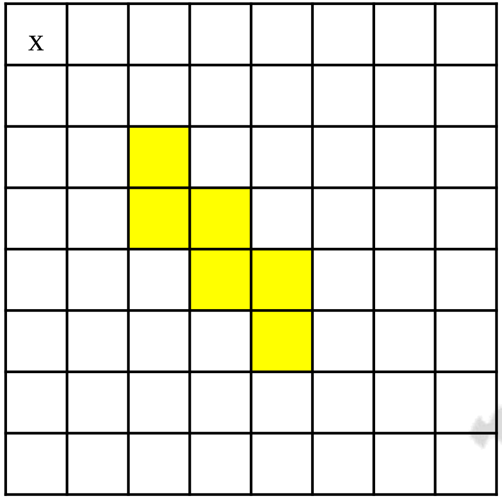
S

I $\quad$ $I_{(1,1)}$ $\quad$ $I_{(0,1)}$

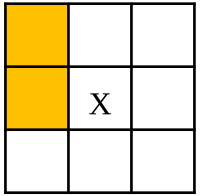
$$I \Theta S = \bigcap_{s \in S} I_{-s}$$

Erosion (as Intersection of object translation)

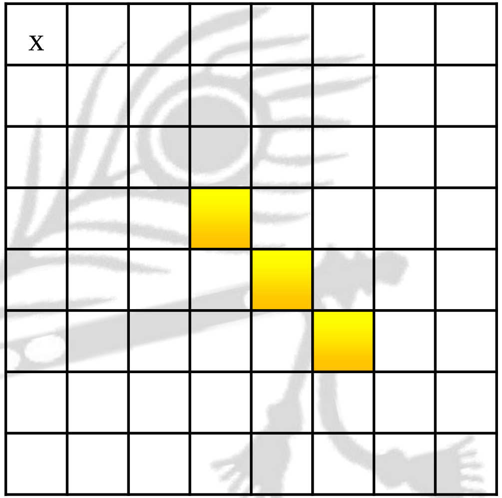
- Eg



I



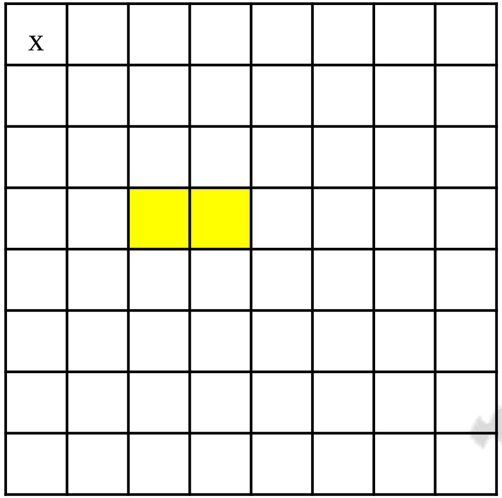
S



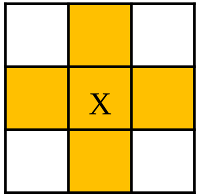
$$I \Theta S = \bigcap_{s \in S} I_{-s}$$

Eg:

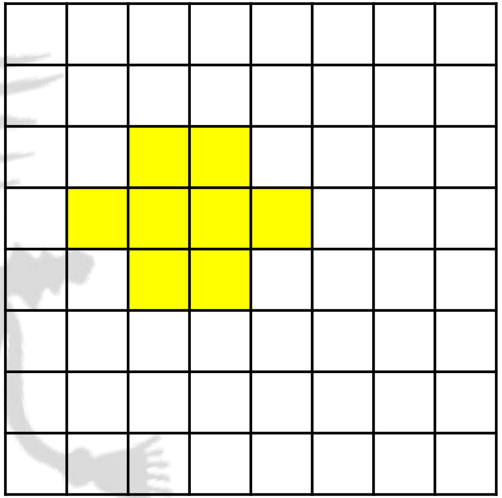
- Find $I \oplus S$



I

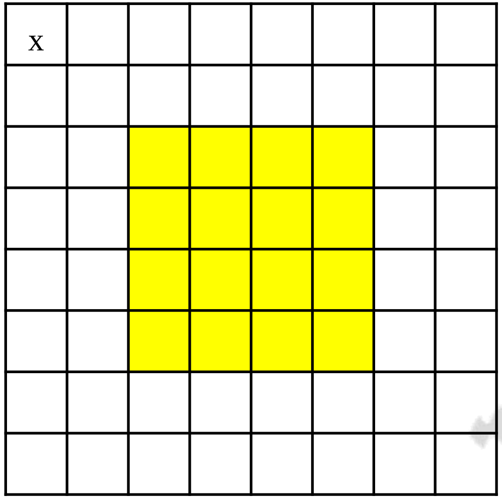


S

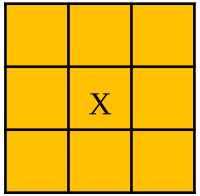


Eg:

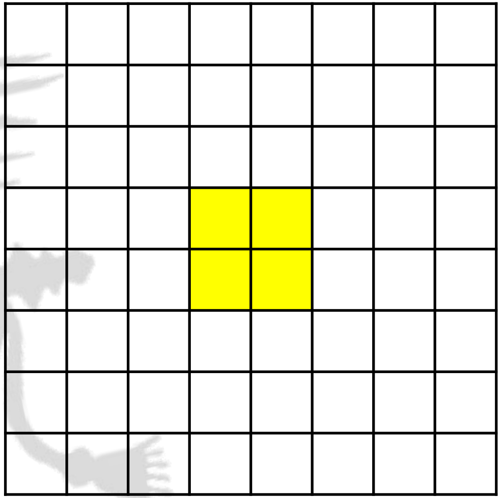
- Find $I \Theta S$



I

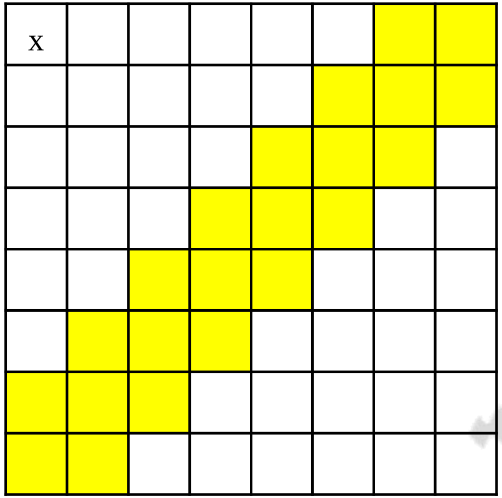


S

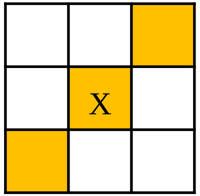


Eg:

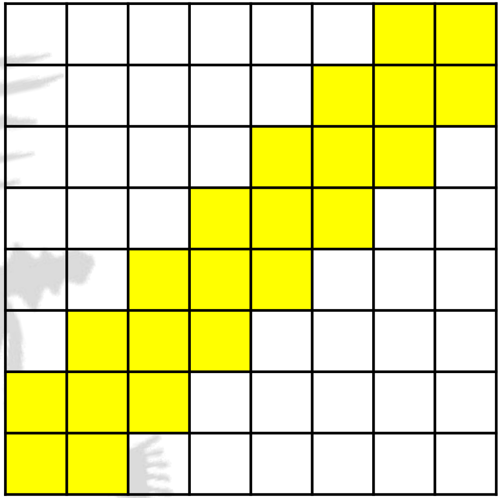
- Find $I \oplus S$



I

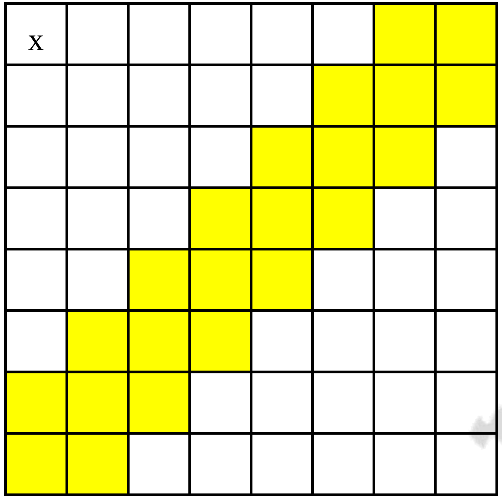


S

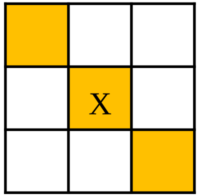


Eg:

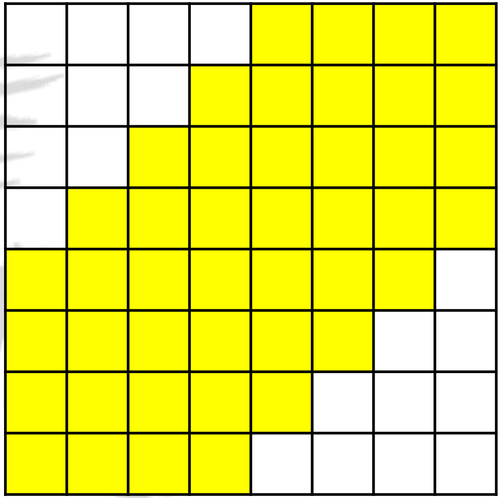
- Find $I \oplus S$



I

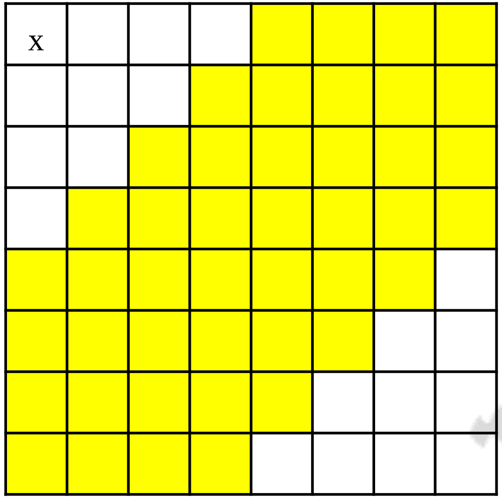


S

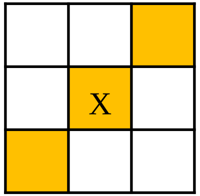


Eg:

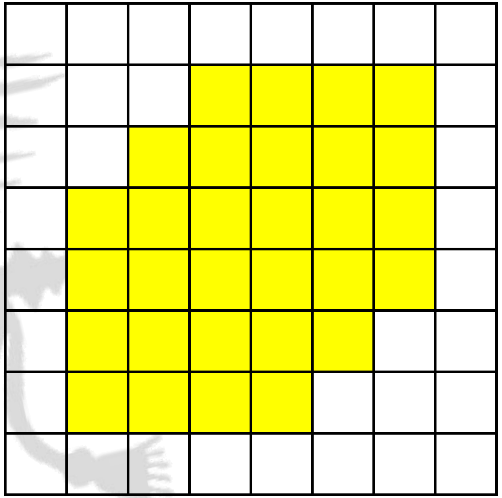
- Find $I \Theta S$



I

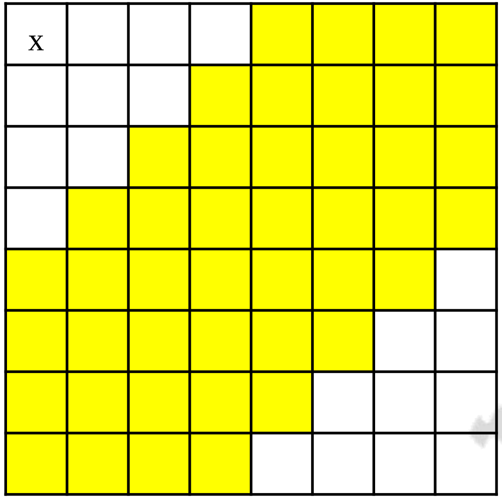


S

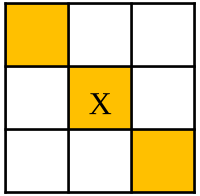


Eg:

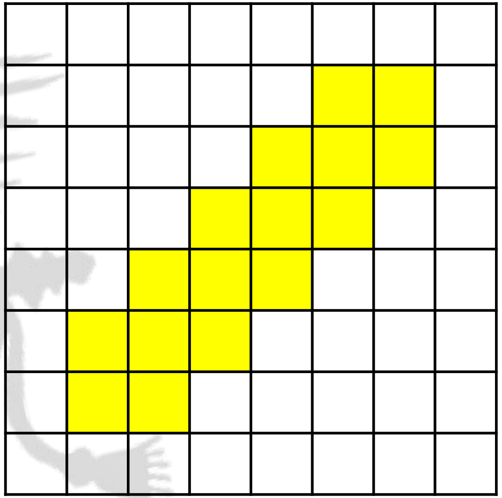
- Find $I \Theta S$



I



S



Duality

- Erosion and dilation are duals of each other with respect to set complementation and reflection

$$\left(A \ominus B\right)^c = \left\{z \mid \left(B\right)_Z \subseteq A\right\}^c$$

$$= \left\{z \mid \left(B\right)_Z \cap A^c = \varnothing\right\}^c$$

$$= \left\{z \mid \left(B\right)_Z \cap A^c \neq \varnothing\right\}$$

$$= A^c \oplus B$$

Combining Dilation and Erosion

- Dilation and Erosion are not inverse transformations
- If an image is eroded & then dilated (or vice-versa), the original image can not be obtained
- In practical applications, dilation and erosion are used most often in various combinations
- Three of the most common combinations of dilation and erosion are
  - Opening
  - Closing
  - Hit or miss transformation

OPENING AND CLOSING

Opening and Closing

- Opening is erosion followed by dilation
- The opening is given as

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

- Closing is dilation followed by erosion
- The closing is given as

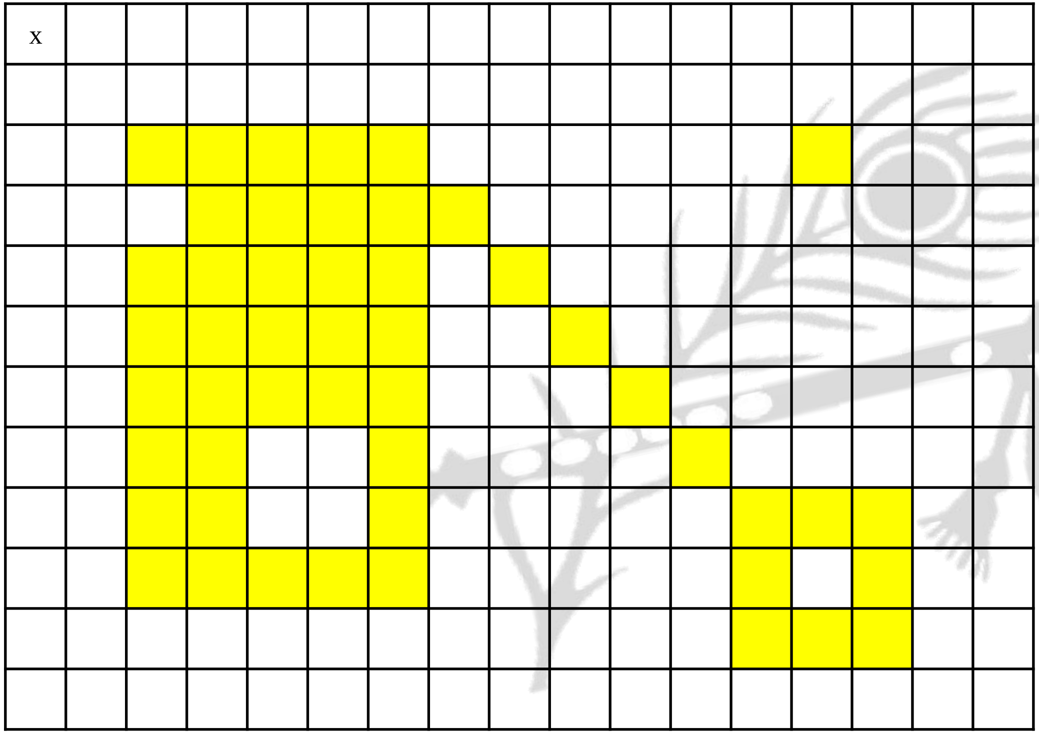
$$A \bullet B = (A \oplus B) \Theta B$$

Opening and Closing

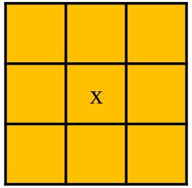
- Opening
  - smooth the contours of an object
  - breaks narrow strips
  - eliminates thin edges
  - it is less destructive than the Erosion

- Closing
  - smooth sections of the contours
  - fuses narrow breaks & long thin gulfs
  - eliminates small holes & fills gaps in the contour

Closing

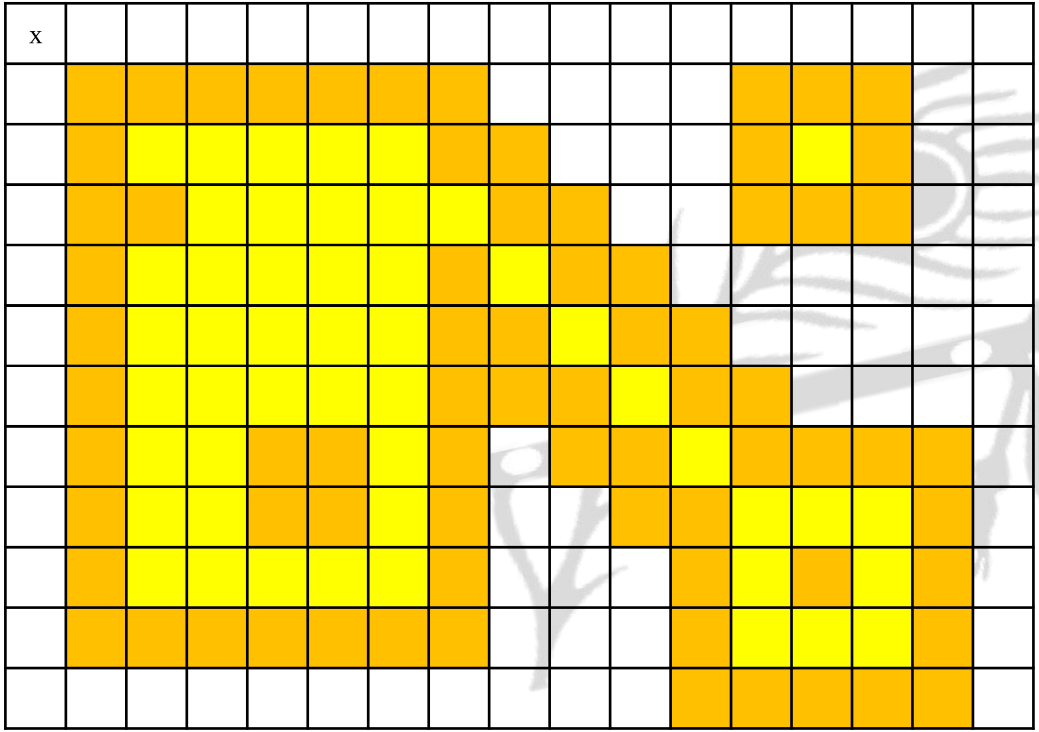


I

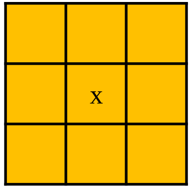


S

Closing

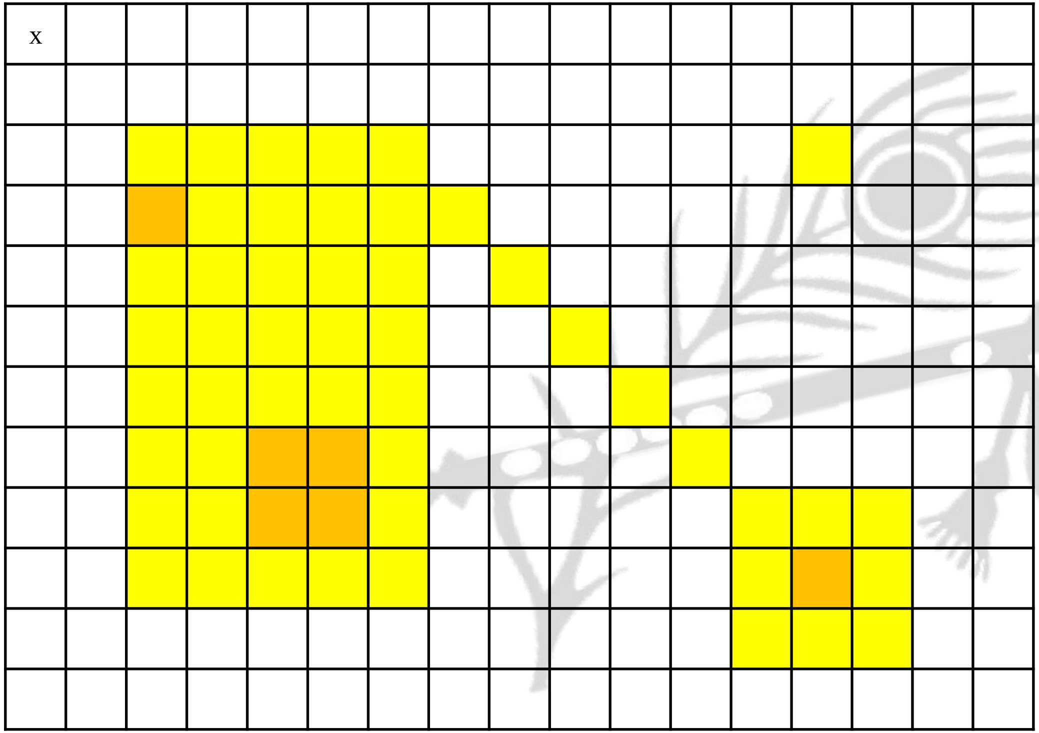


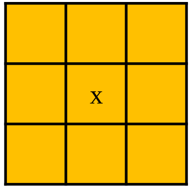
$I \oplus S$



S

Closing

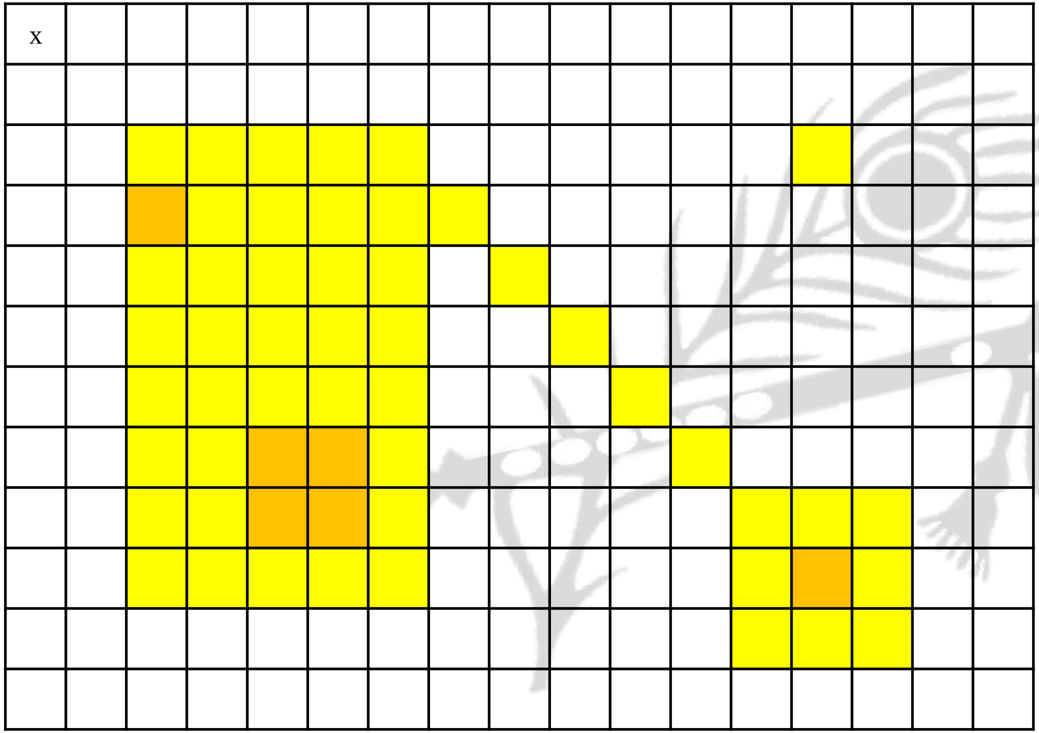




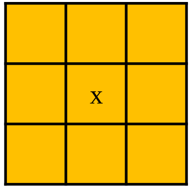
S

$$I \bullet S = (I \oplus S) \ominus S$$

Opening

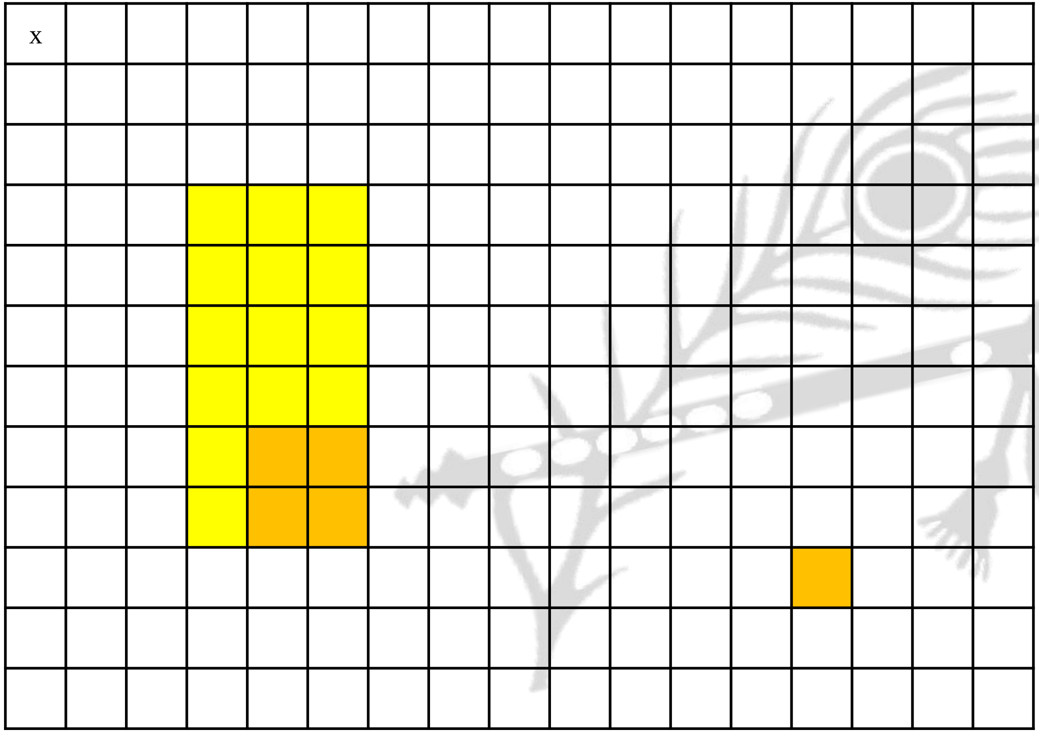


I

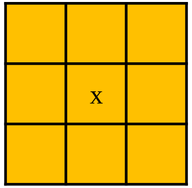


S

Opening

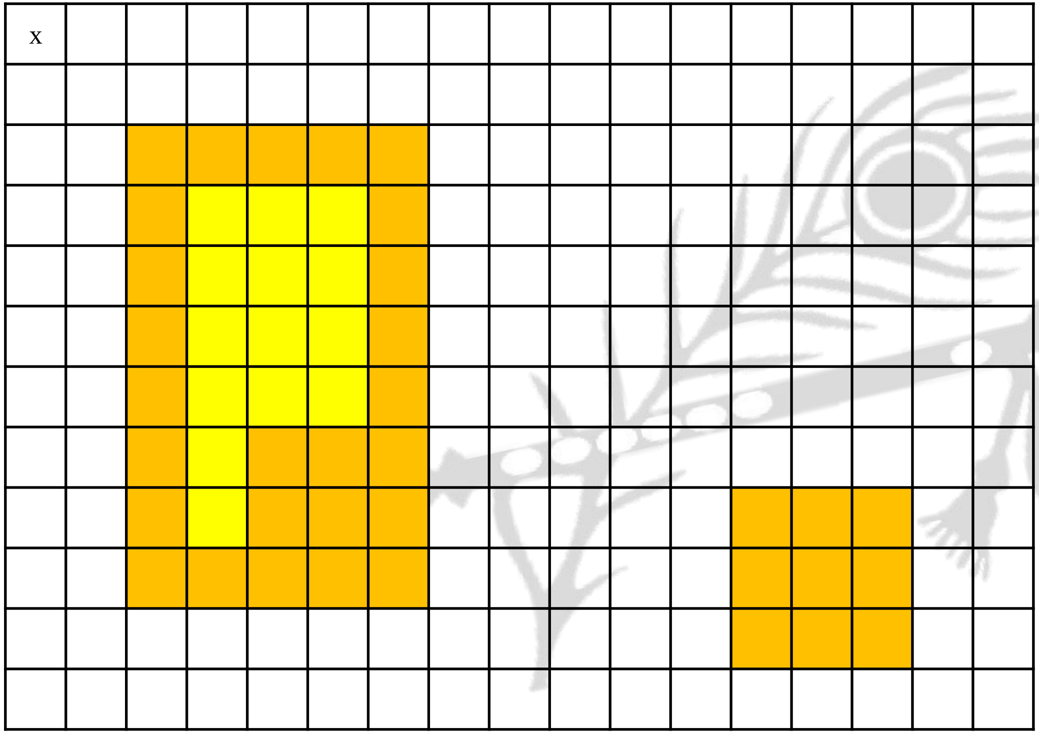


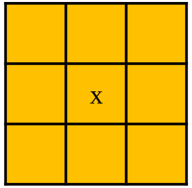
$I \Theta S$



S

Opening





S

$$I \circ S = (I \ominus S) \oplus S$$

HIT OR MISS TRANSFORM

Hit or Miss Transform

- A basic tool for shape detection
- It is a morphological operator for finding local patterns of pixels
- The hit-or-miss transform is a general binary morphological operation that can be used to look for particular patterns of foreground and background pixels in an image

- Concept:
  - Hit object
  - Miss background

Hit or Miss Transform

- It is given as

$$I \circledast S = (I \Theta S) \bigcap (I^c \Theta (W - S))$$

- It can be written as

$$I \circledast S = (I \Theta S_1) \bigcap (I^c \Theta S_2)$$

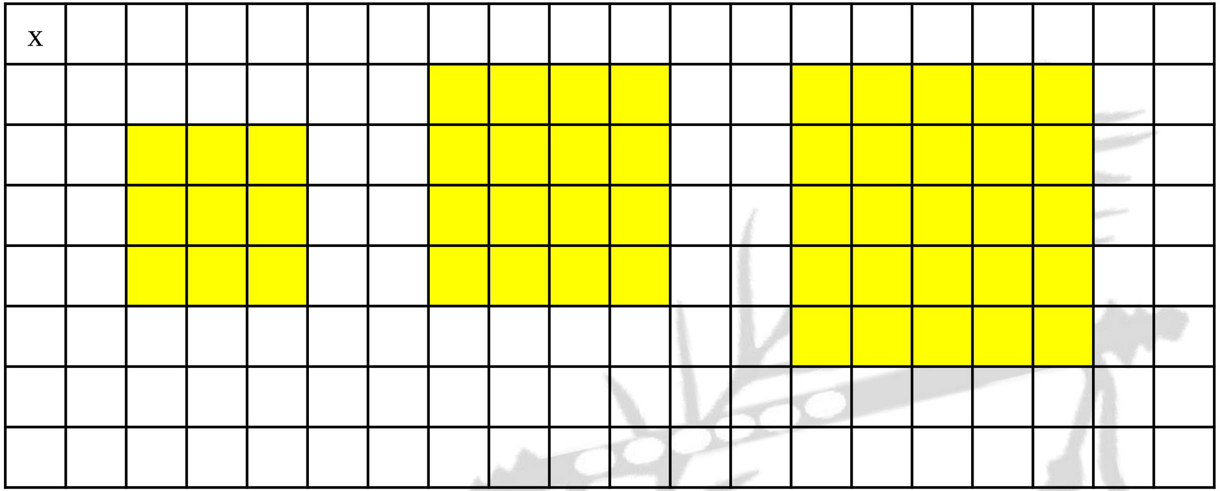
  - where,
    - S_1 is the set formed from elements of S associated with an object (S in this case)
    - S_2 is the set of elements of S associated with the corresponding background (W – S)
- The set contains all the points at which, S_1 found a match (hit) in I and S_2 found a match in I^c

Hit or Miss Transform

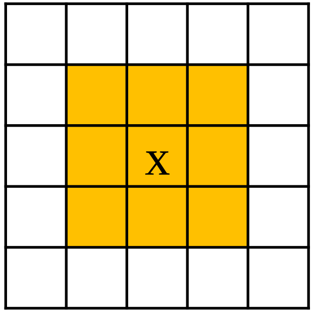
- Using the definition of set difference & the dual relationship between erosion & dilation, the equation can be rewritten as

$$I \circledast S = (I \Theta S_1) - (I \oplus \hat{S}_2)$$

Hit or Miss Transform

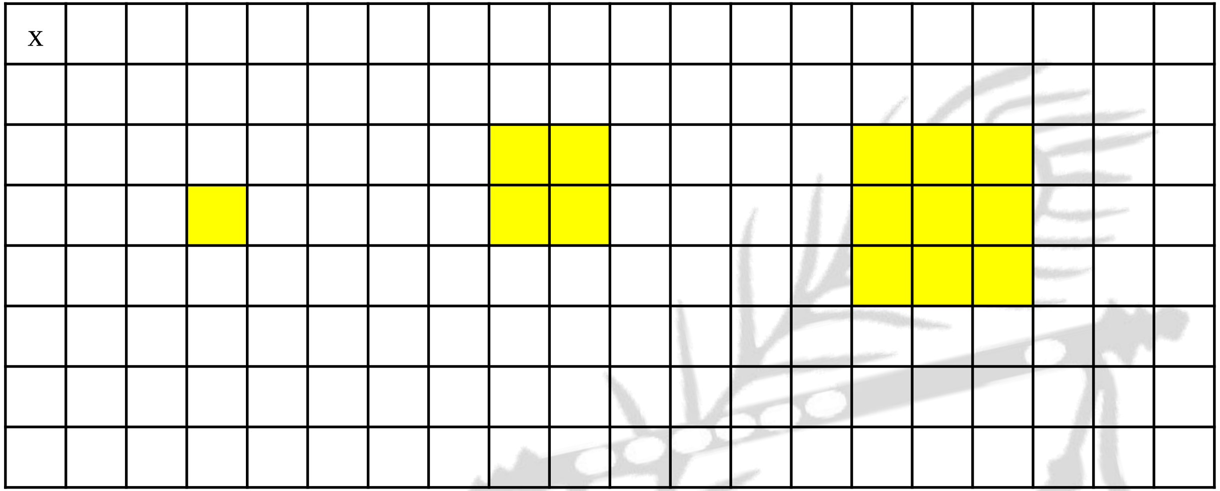


I

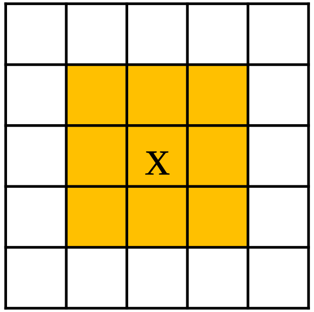


S

Hit or Miss Transform

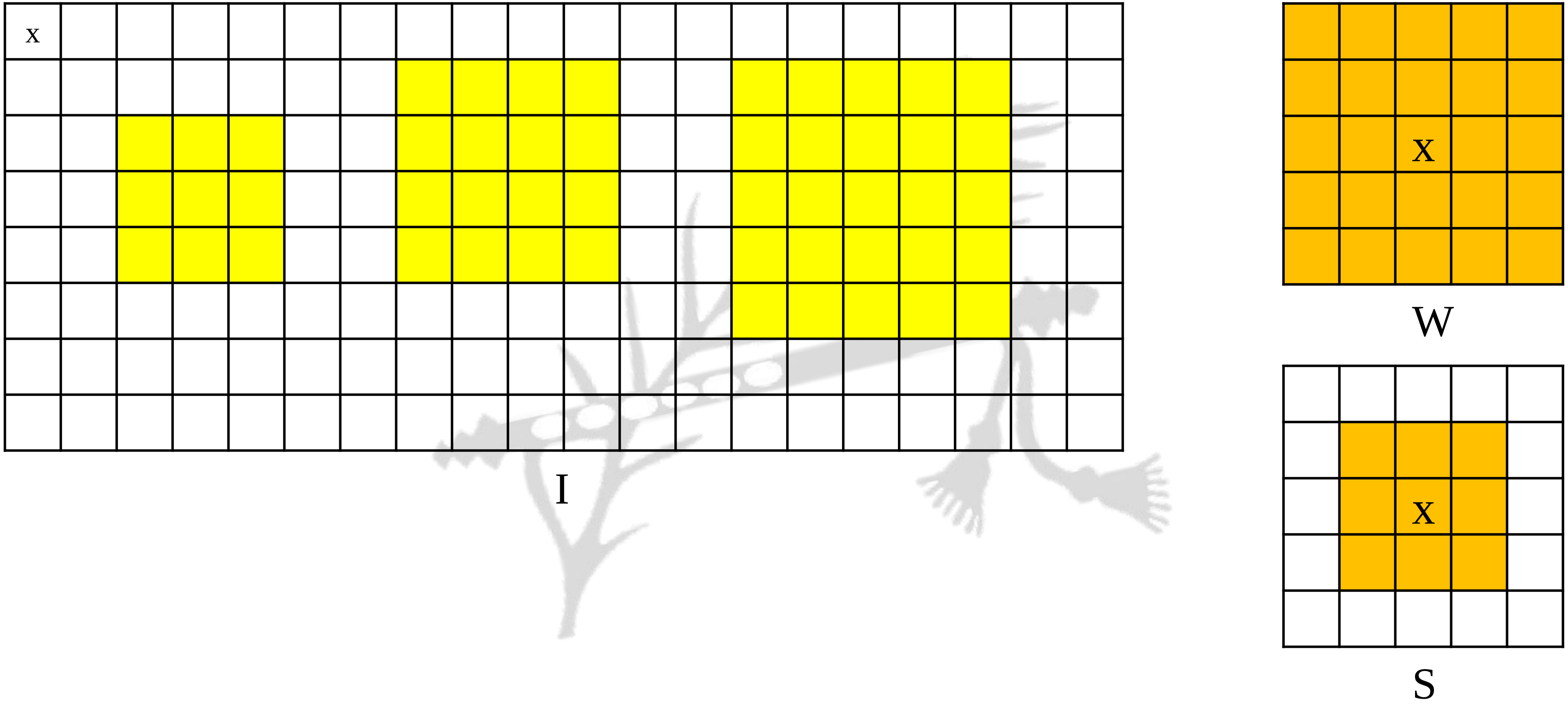


$I \Theta S$

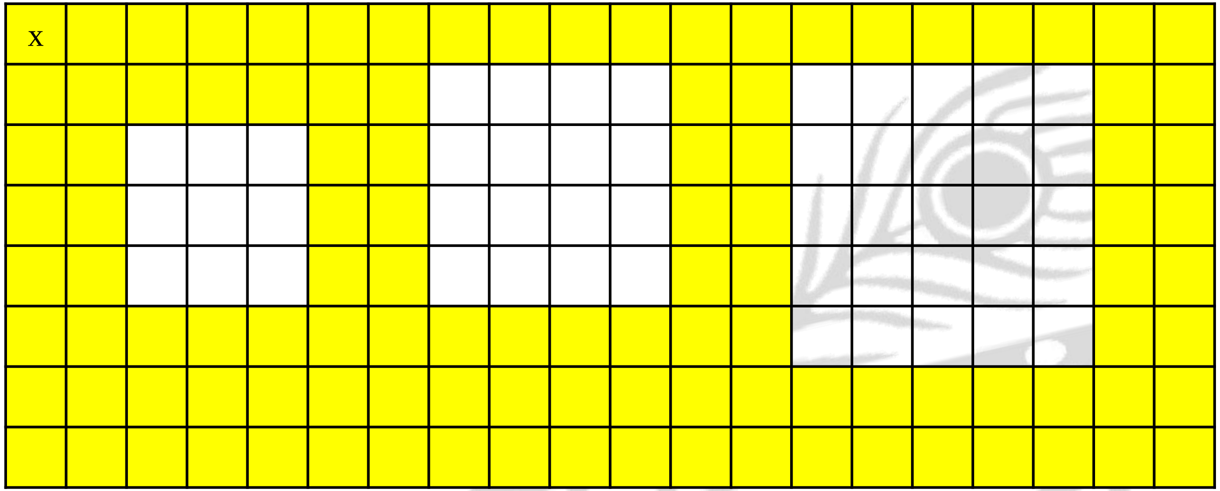


S

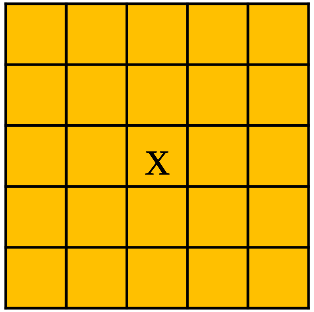
Hit or Miss Transform



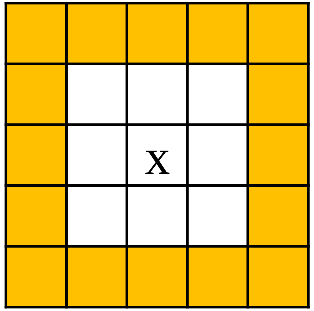
Hit or Miss Transform



I^c

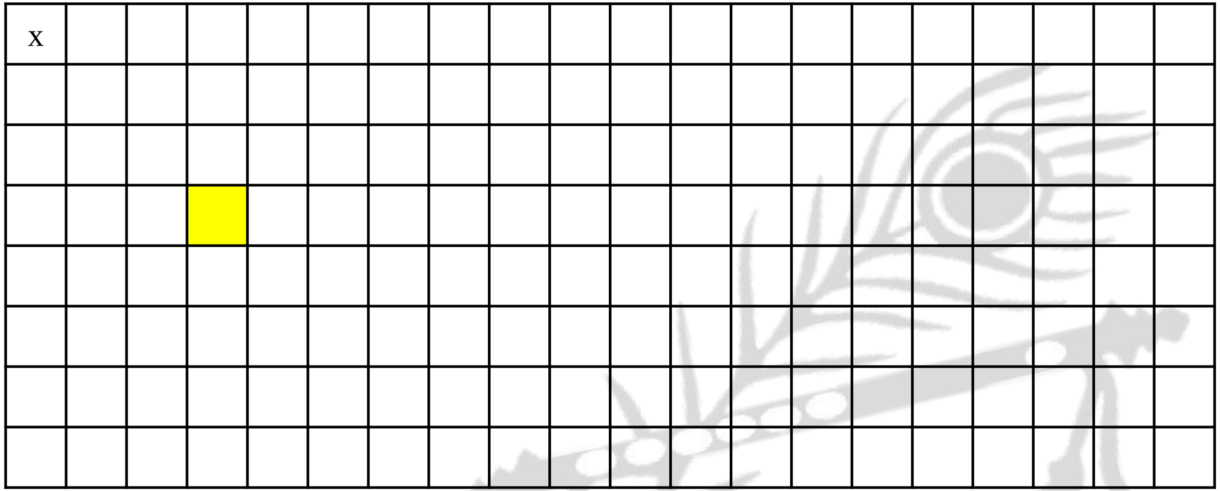


W

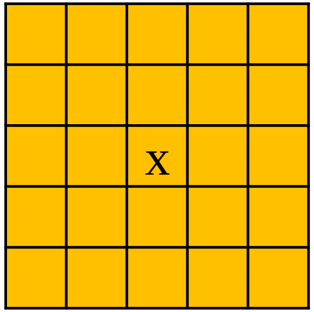


W-S

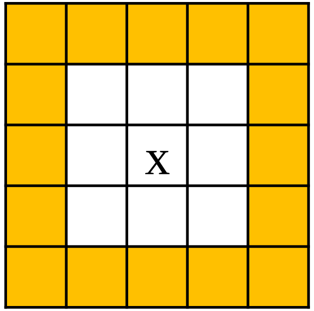
Hit or Miss Transform



$$I^c \Theta (W - S)$$

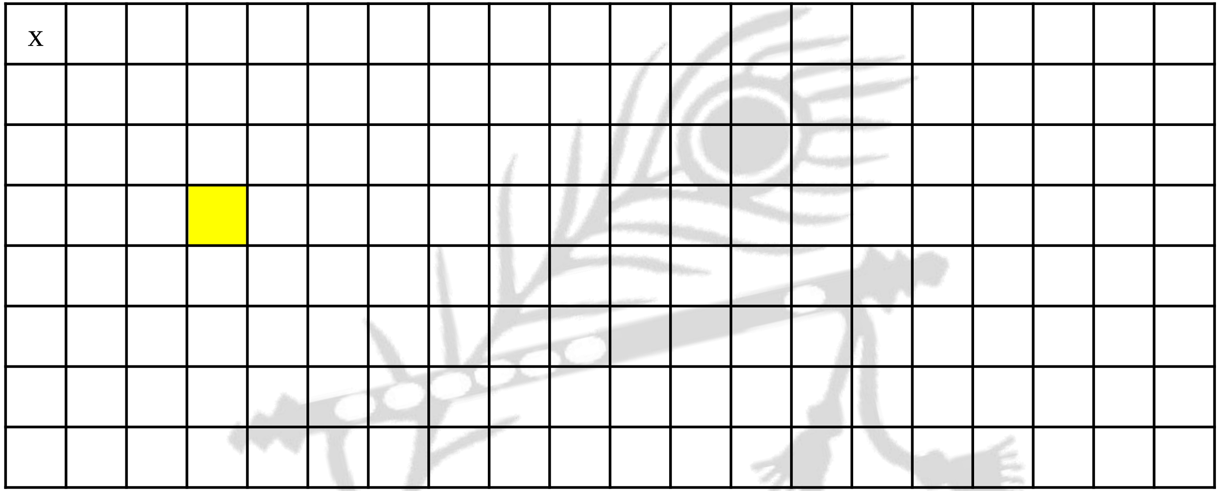


W



W-S

Hit or Miss Transform



$$I \circledast S = (I \Theta S) \bigcap (I^c \Theta (W - S))$$

Eg

- Find $I \circledast S$

←I

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

S→

x	1	x
0	<u>1</u>	1
0	0	x

Eg:

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

$I \Theta S$

x	1	x
0	<u>1</u>	1
0	0	x

S

Eg:

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

I^c

x	1	x
0	<u>1</u>	1
0	0	x

S

Eg:

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

I^c

x	1	x
0	<u>1</u>	1
0	0	x

S

x	0	x
1	<u>0</u>	0
1	1	x

W-S

Eg:

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

$$I^c \Theta (W-S)$$

x	1	x
0	1	1
0	0	x

S

x	0	x
1	0	0
1	1	x

W-S

Eg:

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

Eg

- Find I⊛S

←I

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

S→

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

Eg

- Find I⊛S

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

$I \Theta S$

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

$I^c \Theta (\mathrm{W} - \mathrm{S})$

Boundary Extraction

Boundary Extraction

- The boundary of a region R is the set of pixels in the region that have one or more neighbours that are not in R
- The boundary of a set I can be obtained by first eroding I by S and then performing the set difference between S and its erosion
- It is given by

$$\beta(I) = I - (I \Theta S)$$

Boundary Extraction

- Eg:
  - find $\beta(I)$

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

←I

S→

1	1	1
1	1	1
1	1	1

Boundary Extraction

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

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

$I \Theta S$

Boundary Extraction

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

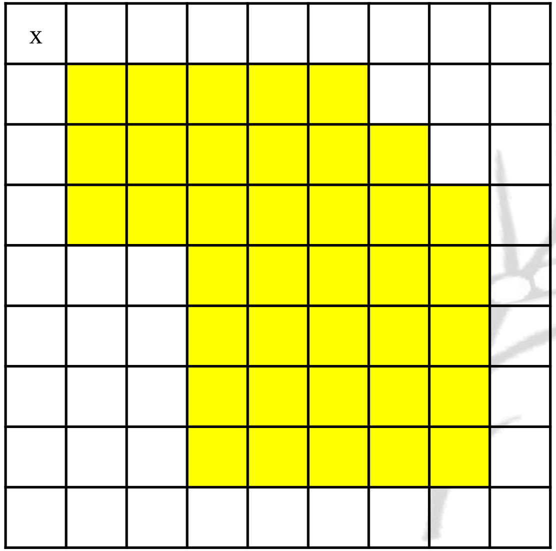
$$I-(I\Theta S)$$

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

$$I\Theta S$$

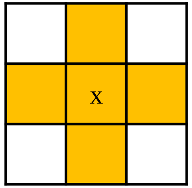
Boundary Extraction

- Eg:
  - find $\beta(I)$

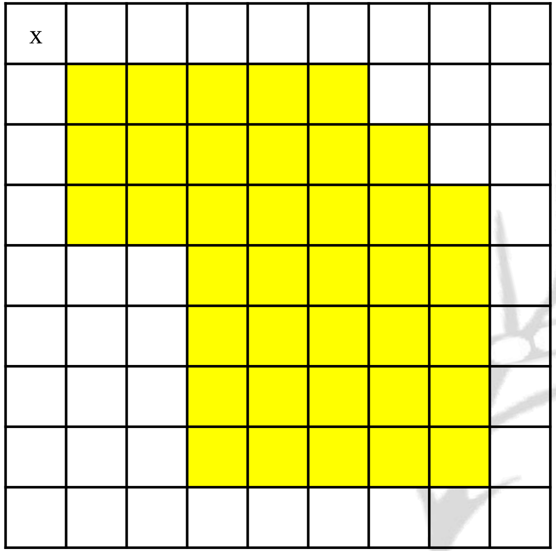


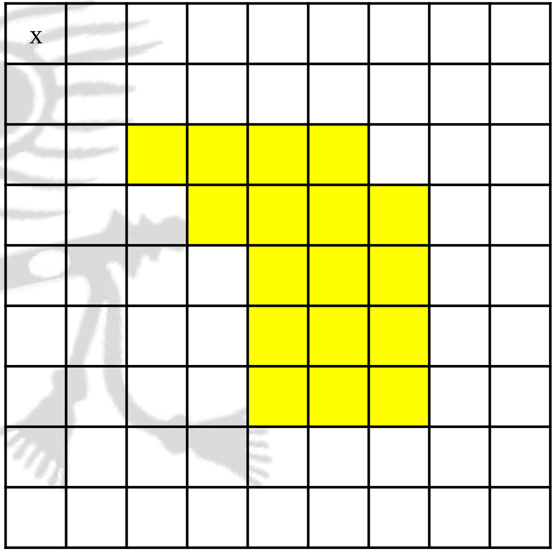
←I

S→



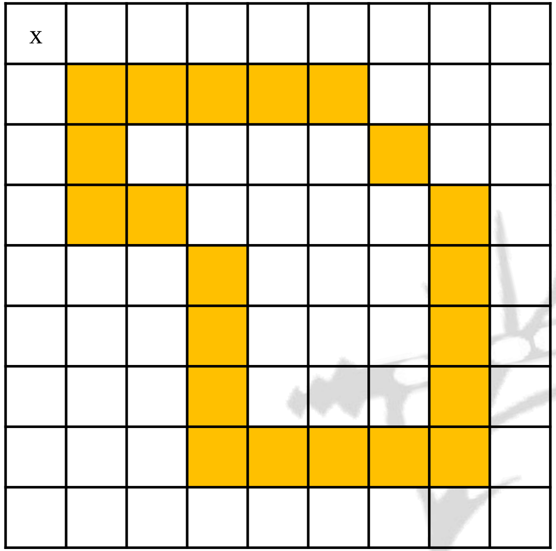
Boundary Extraction





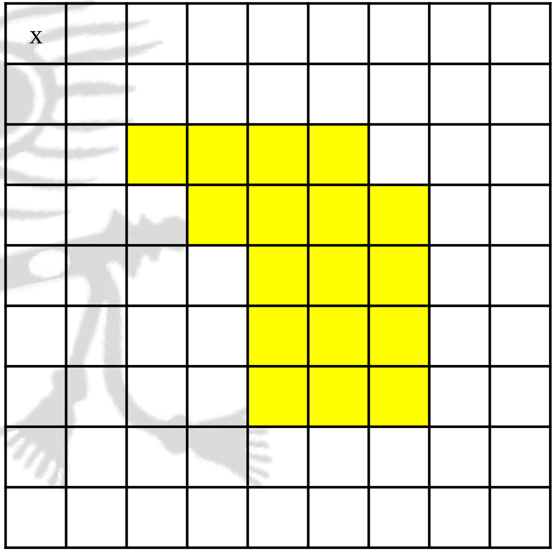
$I \Theta S$

Boundary Extraction



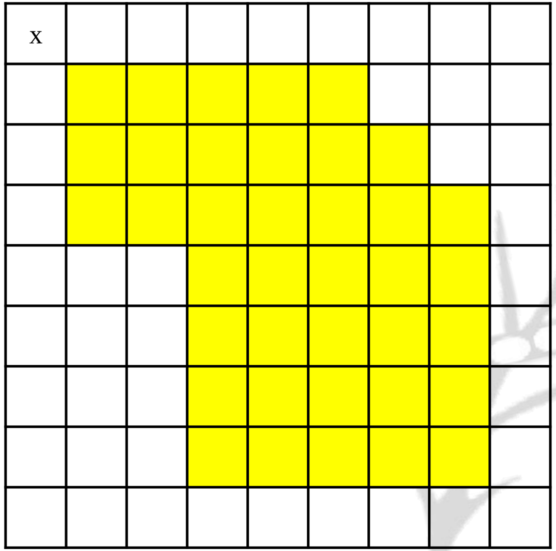
$I-(I\Theta S)$

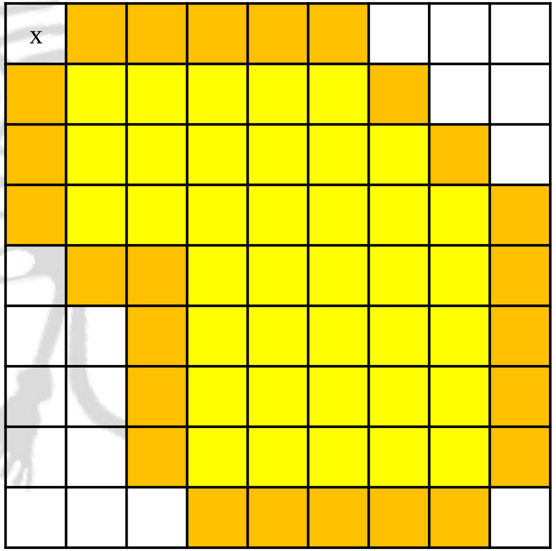
Internal Boundary



$I\Theta S$

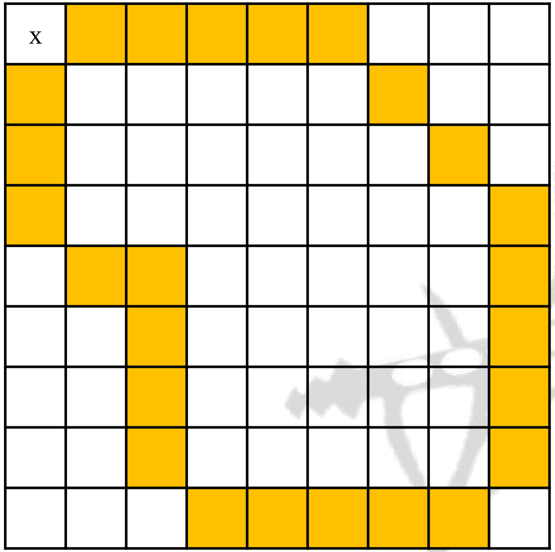
Boundary Extraction





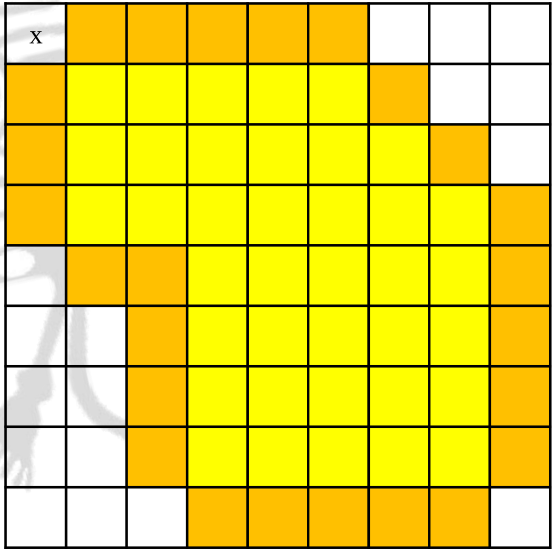
$I \oplus S$

Boundary Extraction



$(I \oplus S) - \mathrm{I}$

External Boundary



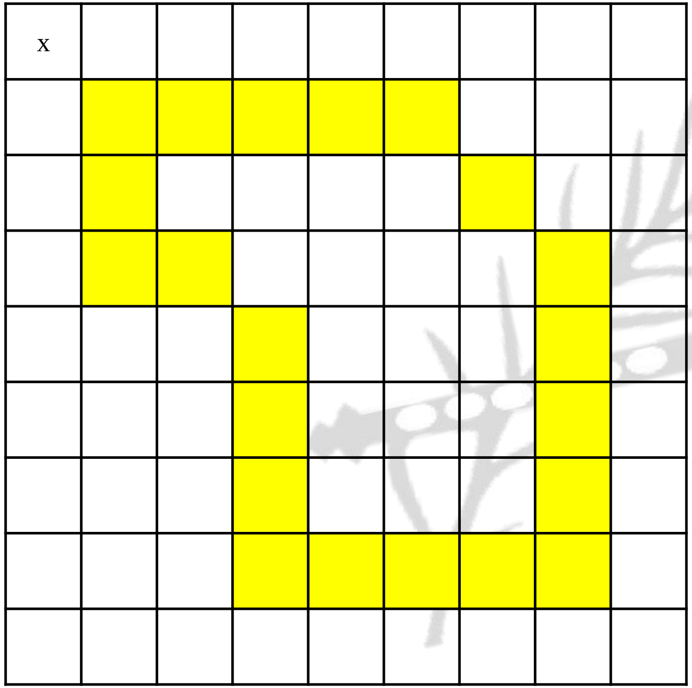
$I \oplus S$

REGION FILLING

Region Filling

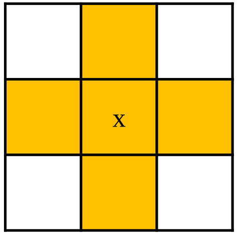
- Region filling is used to fill the selected region of the object
- Steps includes
  - Choose a seed point X_0
  - Iterate $X_k = (X_{k-1} \oplus S) \cap I^c$ until convergence

Region Filling

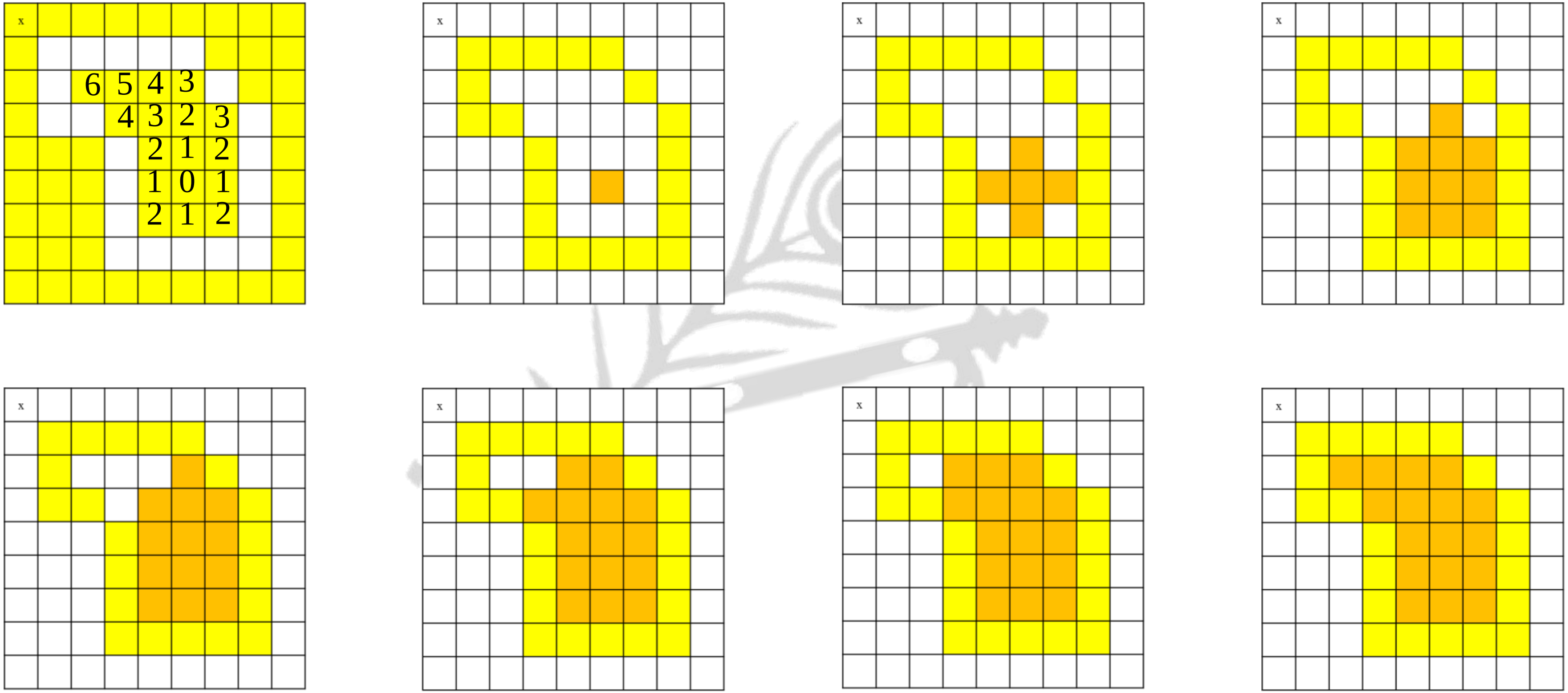


←I

S→

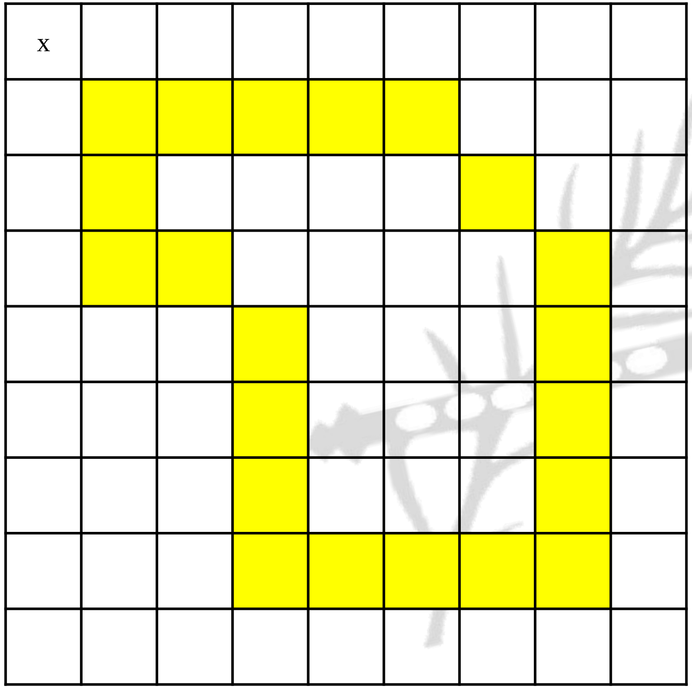


Region Filling



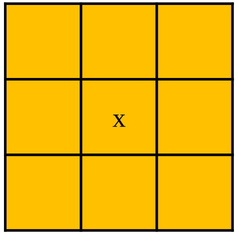
Region Filling

- Eg:



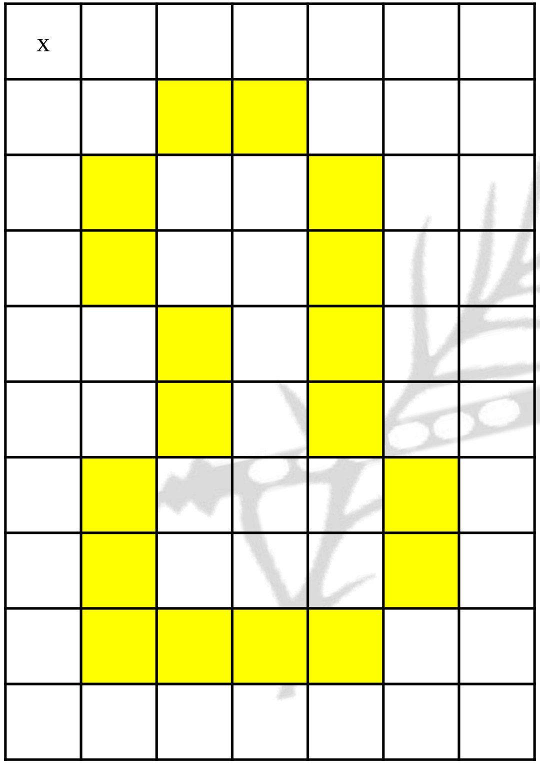
←I

S→



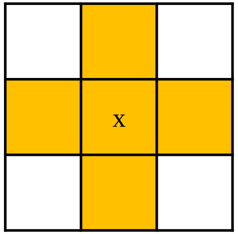
Region Filling

- Eg:



←I

S→

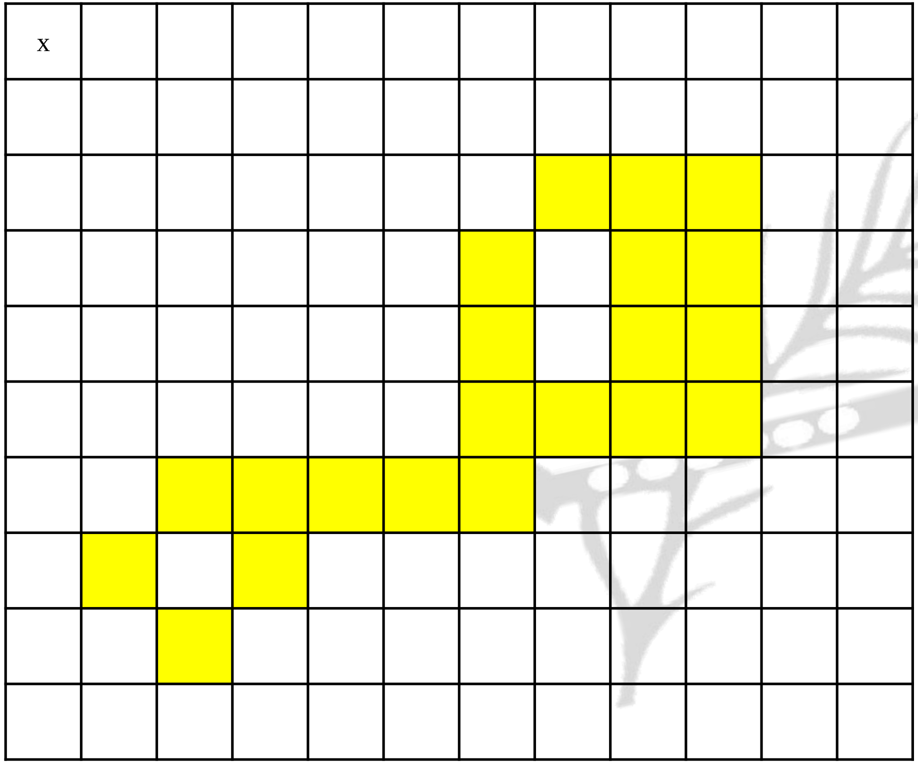


Extraction of Connected Components

Extraction of Connected Components

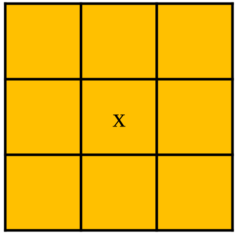
- Connected component labeling is used in computer vision to detect connected regions in the images
- It groups the pixels into components based on the pixel connectivity
- Steps includes
  - Choose a seed point X_0
  - Iterate $X_k = (X_{k-1} \oplus S) \bigcap I$ until convergence

Extraction of Connected Components

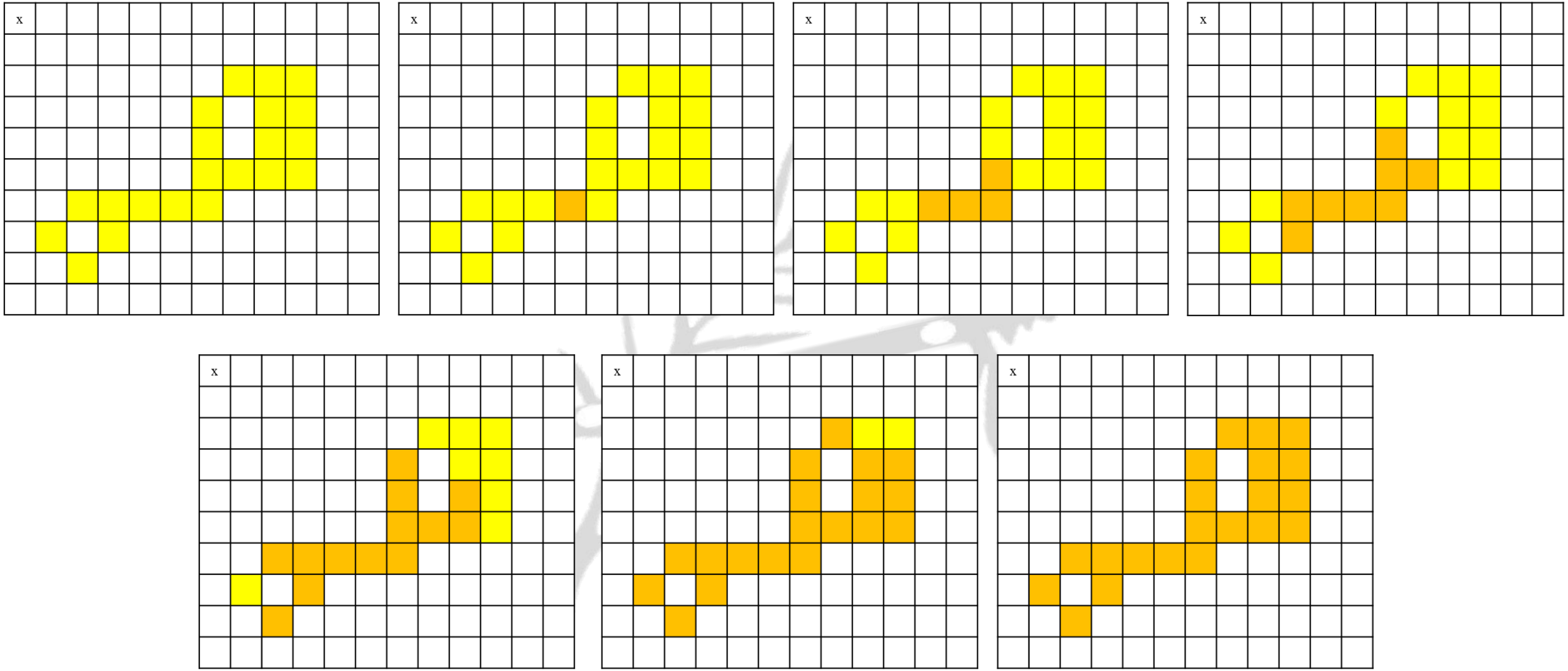


←I

S→

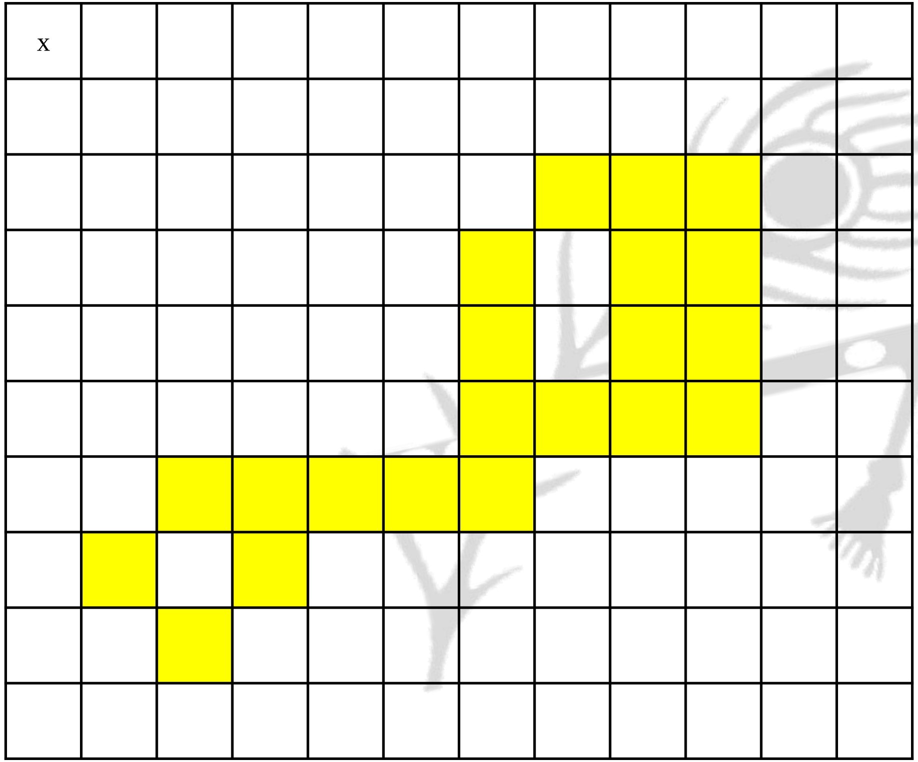


Extraction of Connected Components



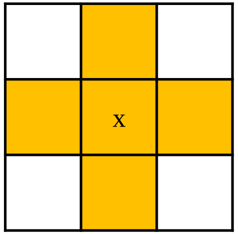
Extraction of Connected Components

- Eg:



←I

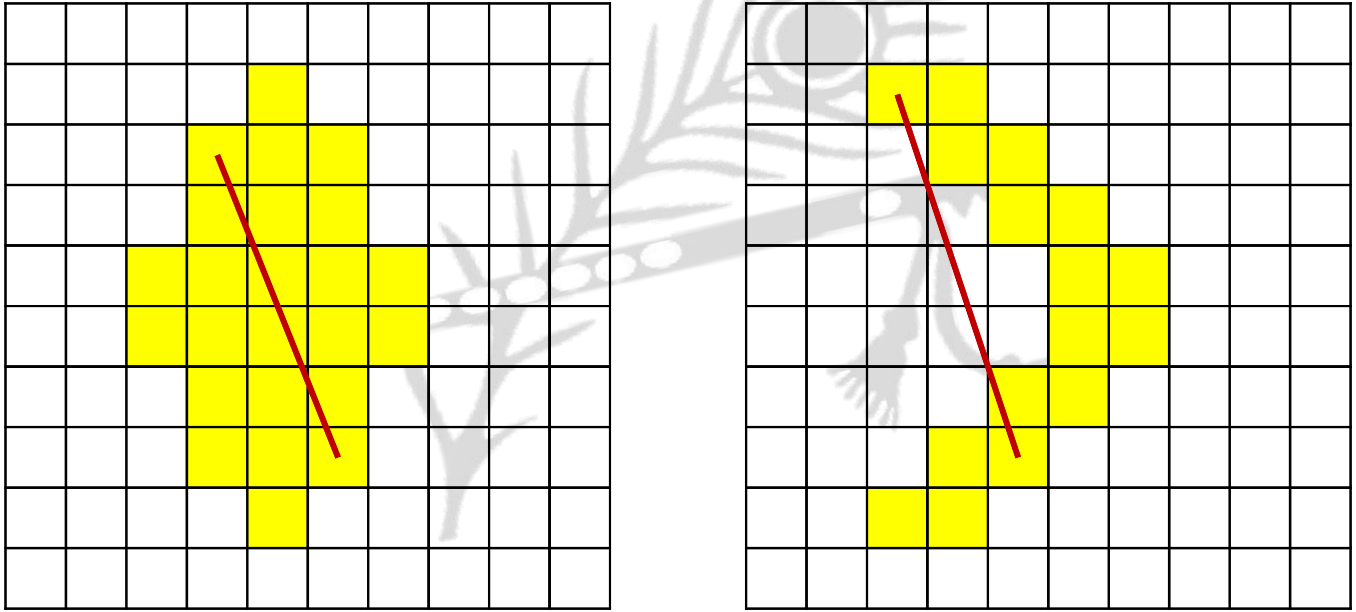
S→



CONVEX HULL

Convex Hull

- A set I is said to be convex if the straight line segment joining any two points in I lies entirely within I

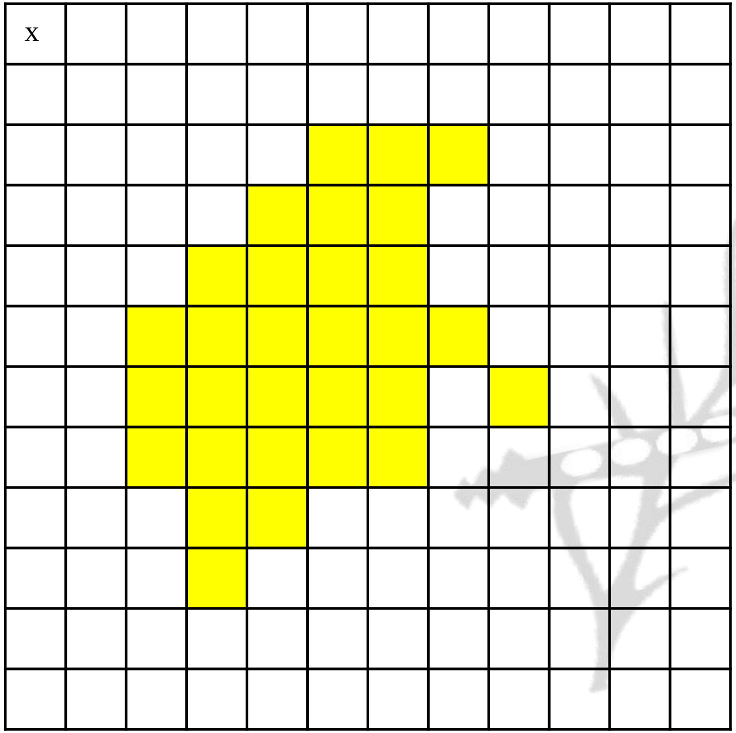


Convex Hull

- Convex Hull (H) = Minimum convex set containing set I
- Hull Deficiency (D) = H – I

- Steps include
  - Choose a seed point X_0
  - do i = 0 to 3
    - Iterate $X_k = (X_{k-1} \circledast S_i) \bigcup I$ until convergence
  - Minimize convex set using bounding box of I

Convex Hull



←I

S→

1	x	x
1	0	x
1	x	x

S_0

1	1	1
x	0	x
x	x	x

S_1

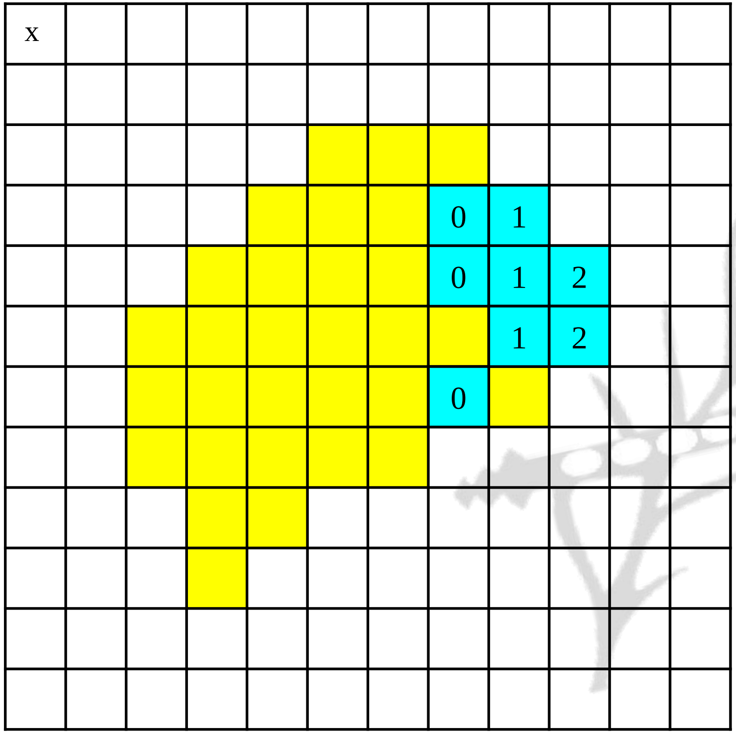
x	x	1
x	0	1
x	x	1

S_2

x	x	x
x	0	x
1	1	1

S_3

Convex Hull



1	x	x
1	0	x
1	x	x

S_0

Convex Hull

x											
							0	1			
							0	1	2		
								1	2		
							0				
							3				
					3	4					
				4	5						
				6							

1	1	1
x	0	x
x	x	x

S_1

Convex Hull

x											
							0	1			
							0	1	2		
								1	2		
	7						0				
	8						3				
		7			3	4					
				4	5						
				6							

x	x	1
x	0	1
x	x	1

S_2

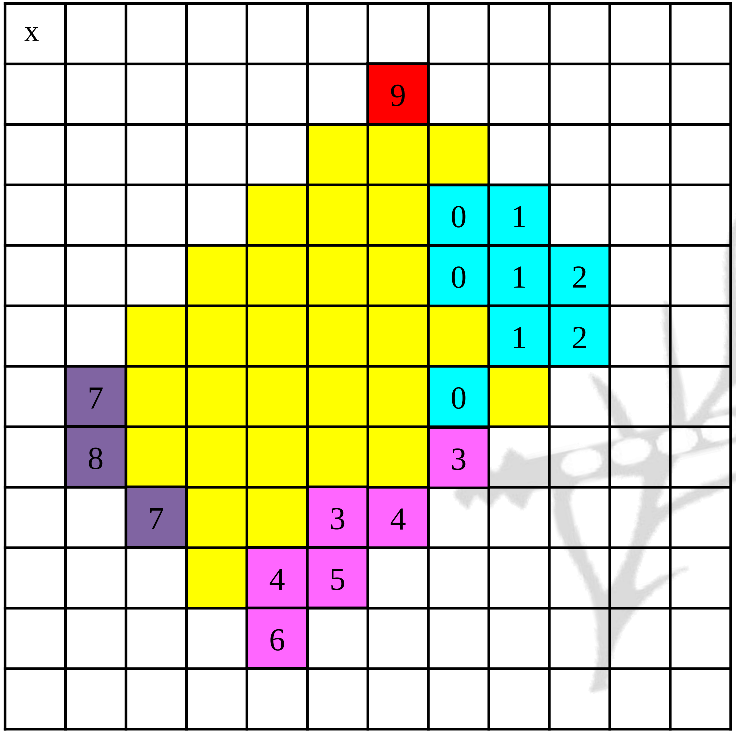
Convex Hull

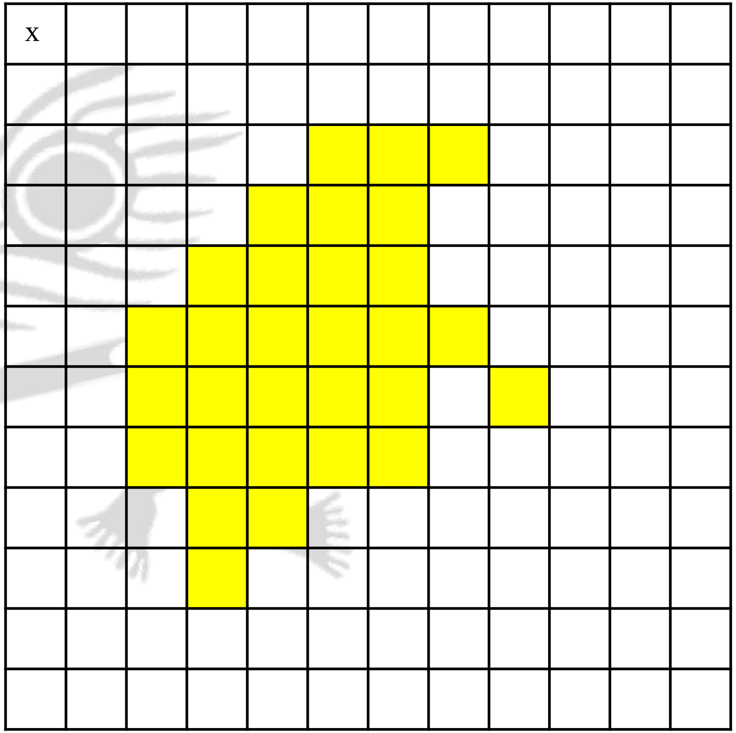
x											
						9					
							0	1			
							0	1	2		
								1	2		
	7						0				
	8						3				
		7			3	4					
				4	5						
				6							

x	x	x
x	0	x
1	1	1

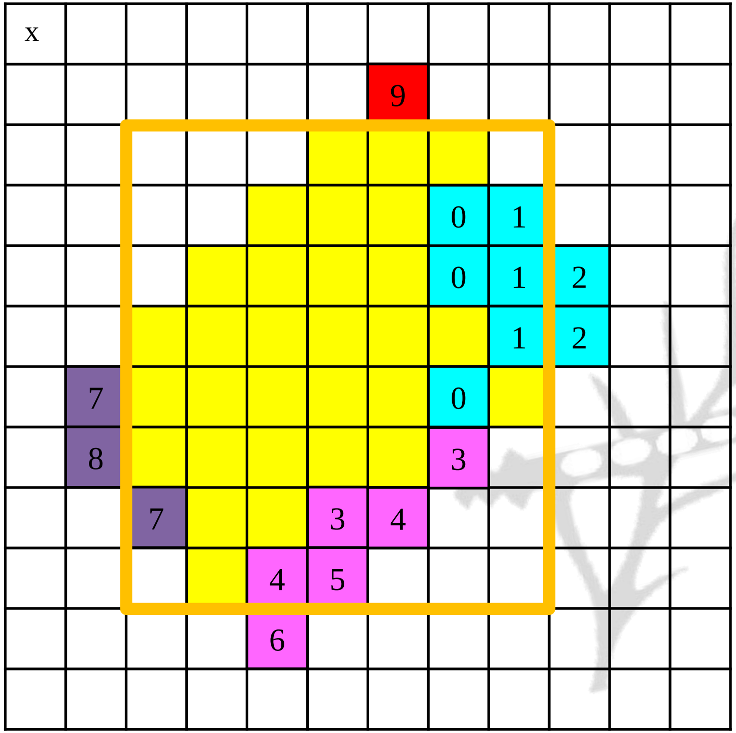
S_3

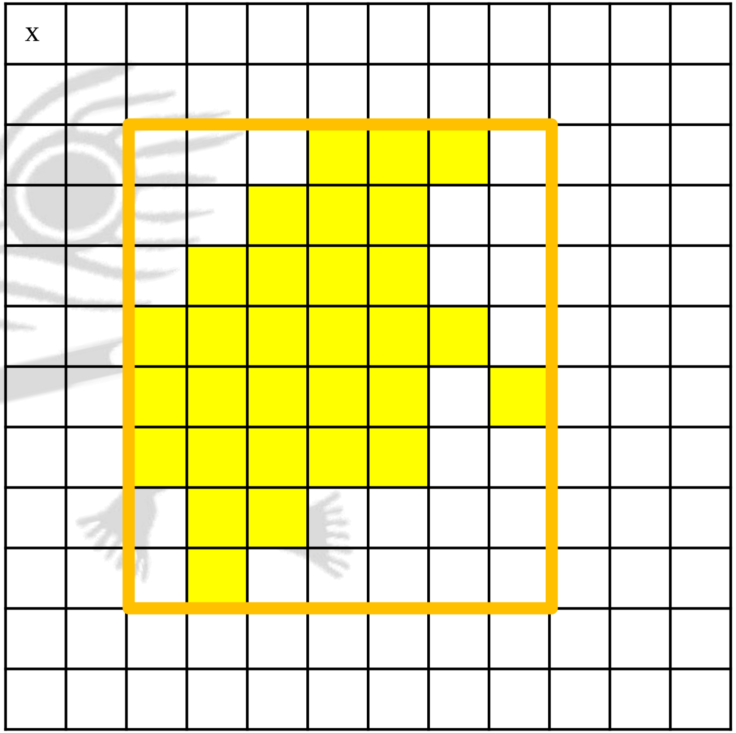
Convex Hull



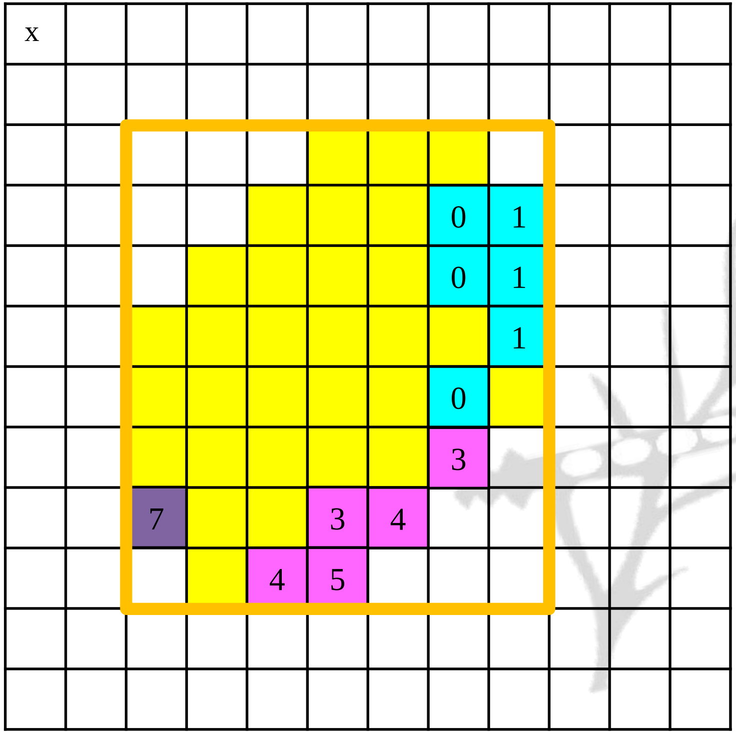


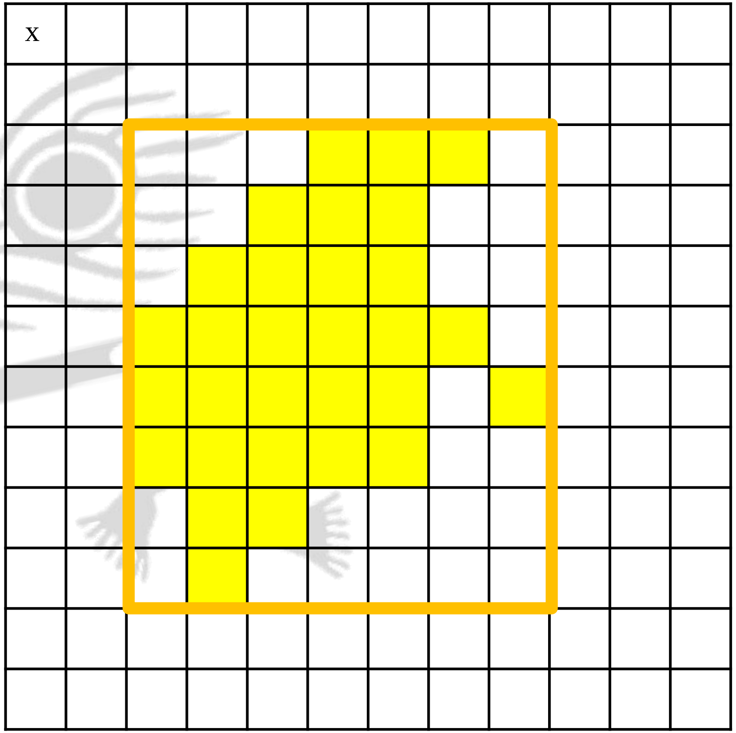
Convex Hull



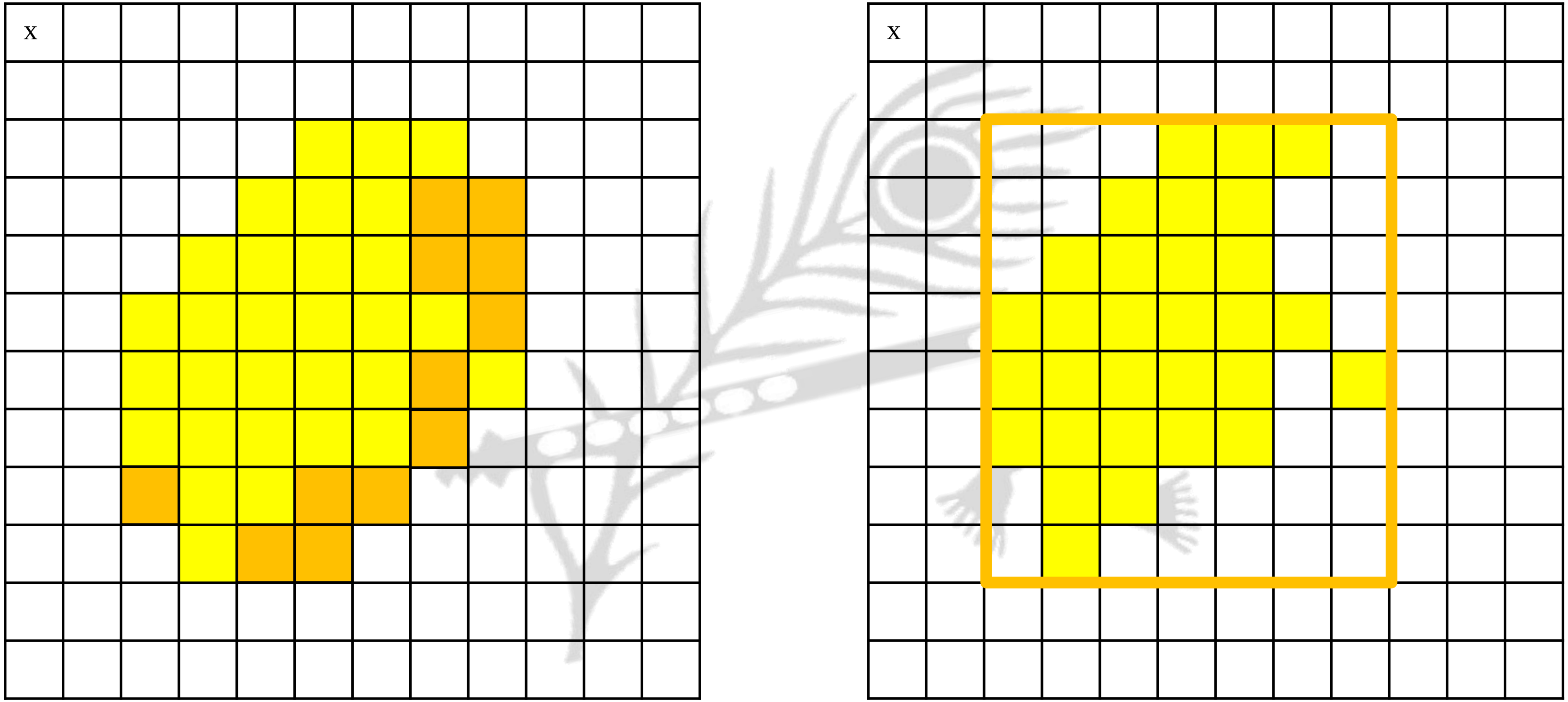


Convex Hull

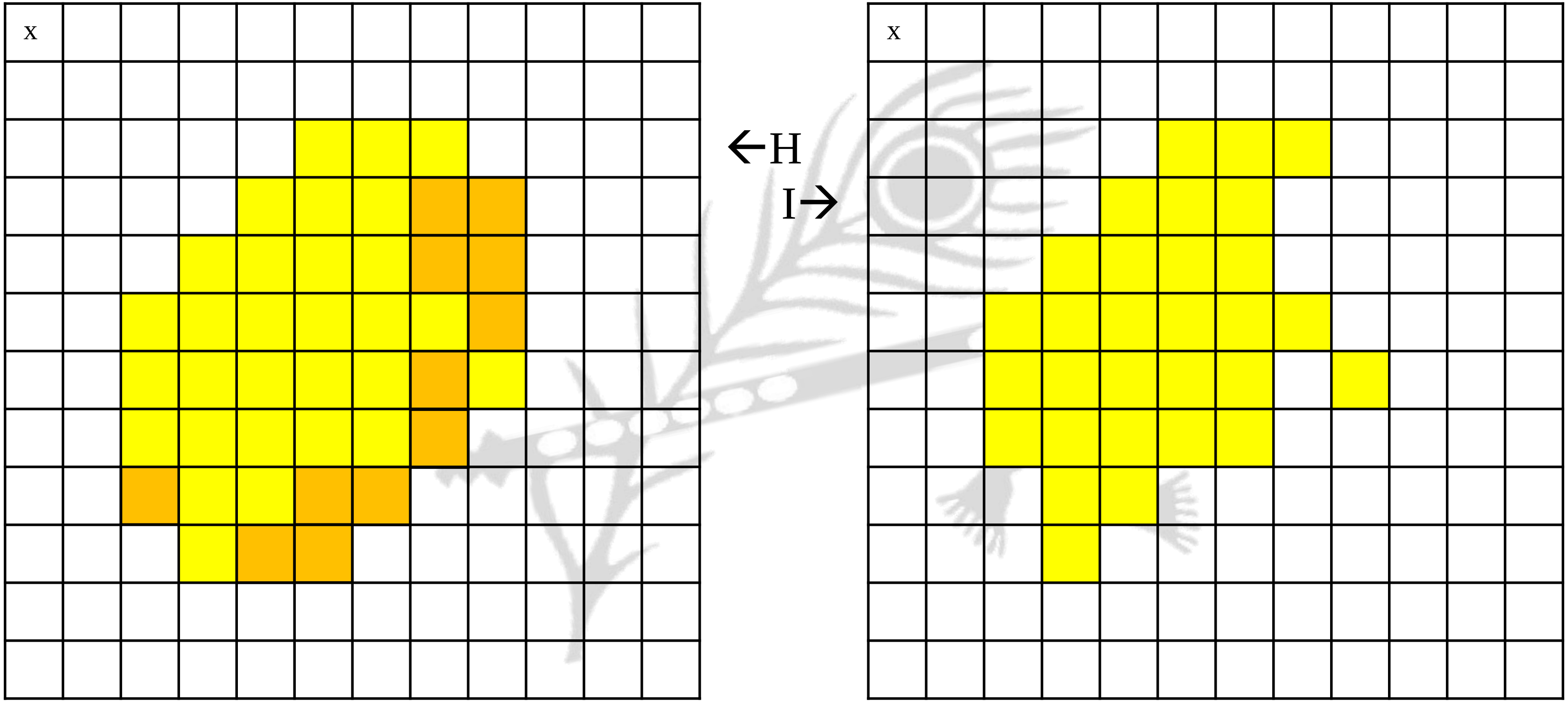




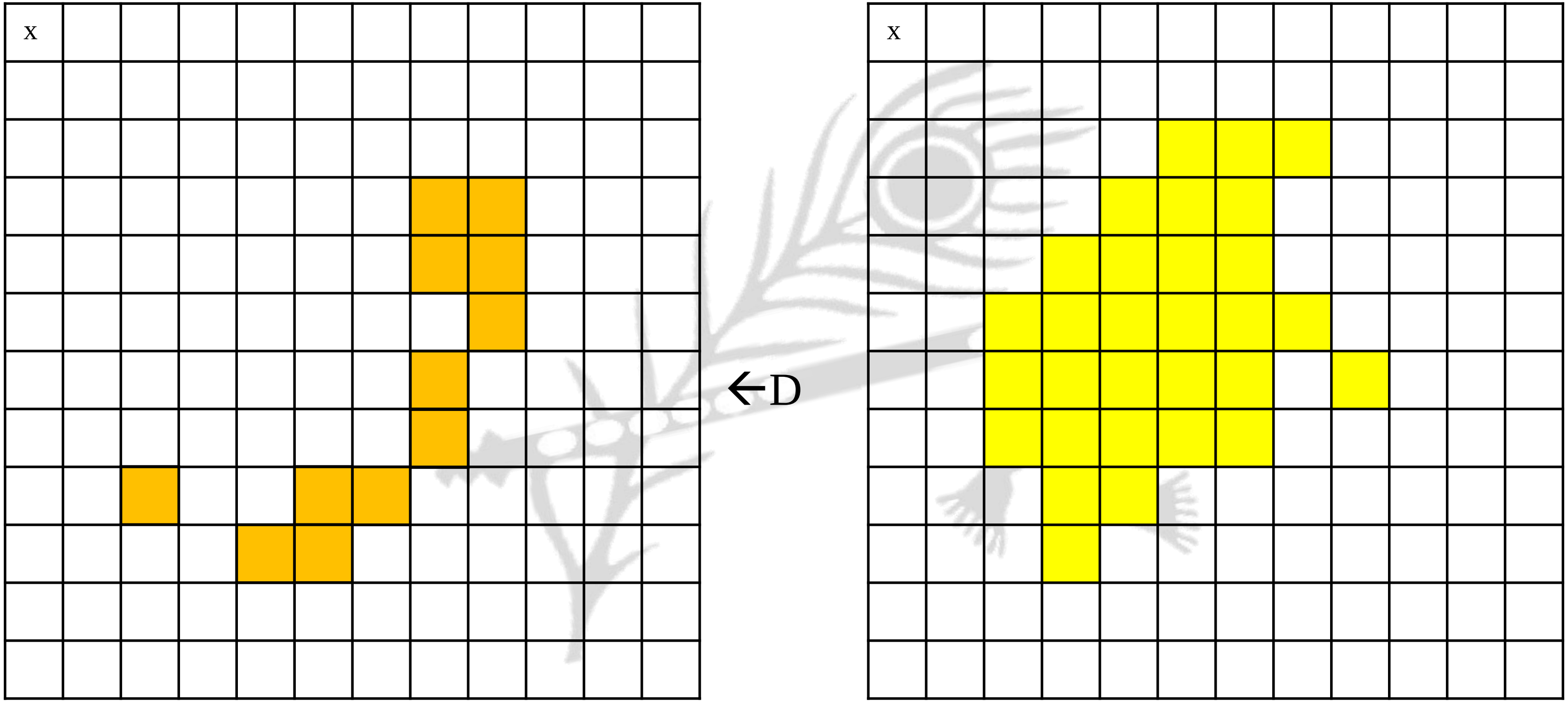
Convex Hull



Convex Hull



Convex Hull



MORPHOLOGICAL THINNING AND THICKENING

Thinning

- It is an operation that is used to remove selected foreground pixels from binary images
- It is the process of reducing an object in a digital image to the minimum size
- It is given by

$$I \otimes S = I - (I \circledast S_i)$$

$$I \otimes S = I \bigcap (I \circledast S_i)$$

$$I \otimes \{S\} = ((((I \circledast S_0) \circledast S_1) \circledast S_2) \ldots \circledast S_7)$$

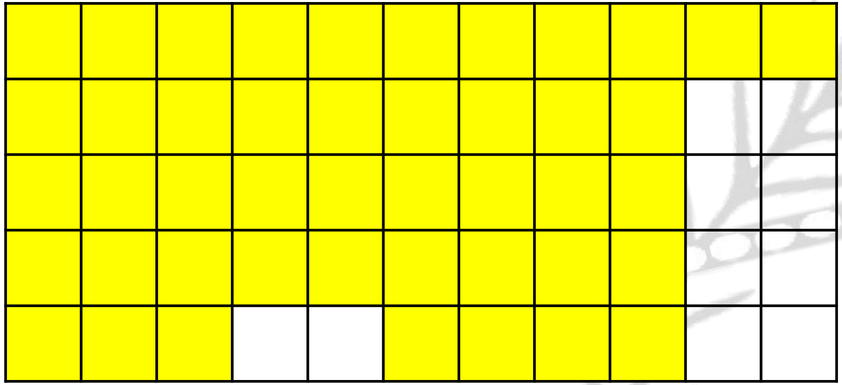
Thinning

- Thinning is basically a search & delete process
- It removes only those boundary pixels from the image whose deletion
  - Does not connectivity of their neighbours locally
  - Does not reduce the length of the already thinned curve

- Critical pixel
  - Its deletion changes the connectivity of its neighbourhood locally

- End pixel
  - Its deletion reduces the length of an already thinned curve

Thinning



I

0	0	0
X	1	X
1	1	1

S_0

X	0	0
1	1	0
1	1	X

S_1

1	X	0
1	1	0
1	X	0

S_2

1	1	X
1	1	0
X	0	0

S_3

1	1	1
X	1	X
0	0	0

S_4

X	1	1
0	1	1
0	0	X

S_5

0	X	1
0	1	1
0	X	1

S_6

0	0	X
0	1	1
X	1	1

S_7

Thinning

0	0	0
X	1	X
1	1	1

S_0

	0	0	0	0	0	0	0			

Thinning

X	0	0
1	1	0
1	1	X

S_1

	0	0	0	0	0	0	0			

Thinning

1	X	0
1	1	0
1	X	0

S_2

	0	0	0	0	0	0	0			
								0		
								0		

Thinning

	0	0	0	0	0	0	0			
								0		
								0		
		0								

1	1	X
1	1	0
X	0	0

S_3

Thinning

	0	0	0	0	0	0	0			
								0		
			0					0		
	0	0			0	0				

1	1	1
X	1	X
0	0	0

S_4

Thinning

	0	0	0	0	0	0	0			
								0		
			0	0				0		
	0	0			0	0				

X	1	1
0	1	1
0	0	X

S_5

Thinning

	0	0	0	0	0	0	0			
0								0		
			0	0				0		
	0	0			0	0				

0	X	1
0	1	1
0	X	1

S_6

Thinning

	0	0	0	0	0	0	0			
0								0		
			0	0				0		
	0	0			0	0				

0	0	X
0	1	1
X	1	1

S_7

Thinning

0	0	0
X	1	X
1	1	1

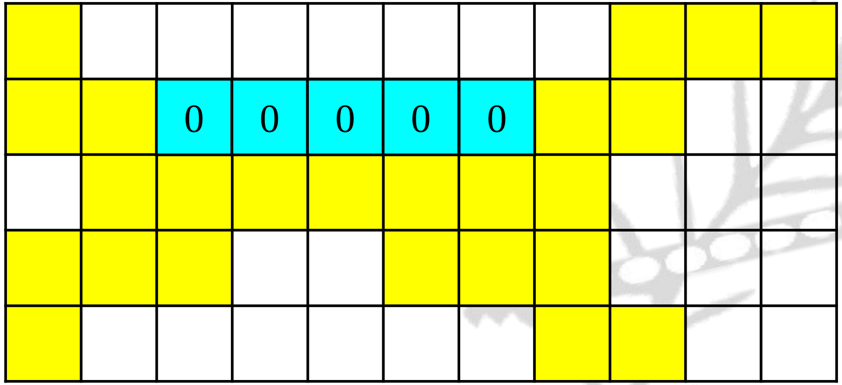
S_0

		0	0	0	0	0				

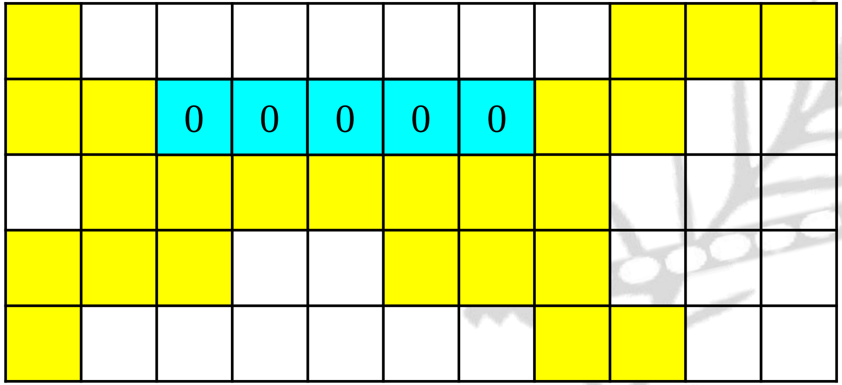
Thinning

X	0	0
1	1	0
1	1	X

S_1



Thinning



1	X	0
1	1	0
1	X	0

S_2

Thinning

		0	0	0	0	0				

1	1	X
1	1	0
X	0	0

S_3

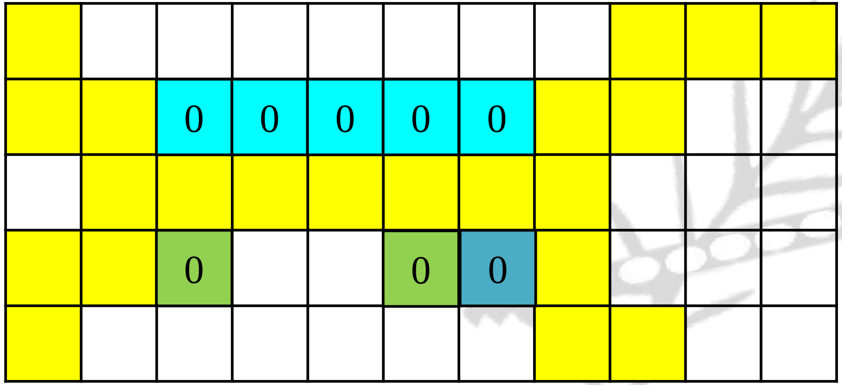
Thinning

		0	0	0	0	0				
		0			0					

1	1	1
X	1	X
0	0	0

S_4

Thinning



X	1	1
0	1	1
0	0	X

S_5

Thinning

		0	0	0	0	0				
		0			0	0				

0	X	1
0	1	1
0	X	1

S_6

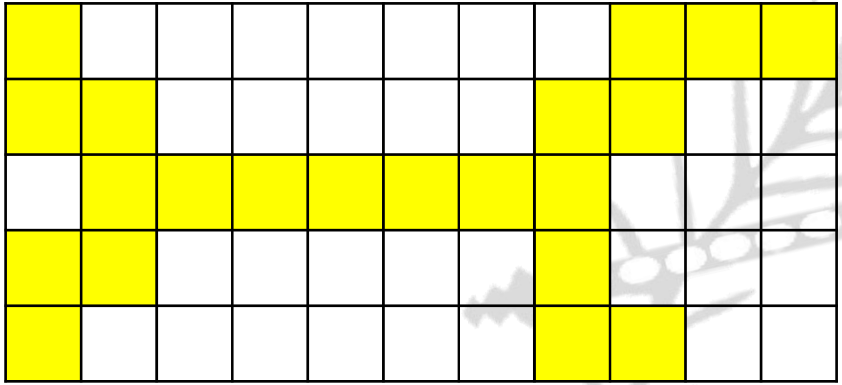
Thinning

		0	0	0	0	0				
		0			0	0				

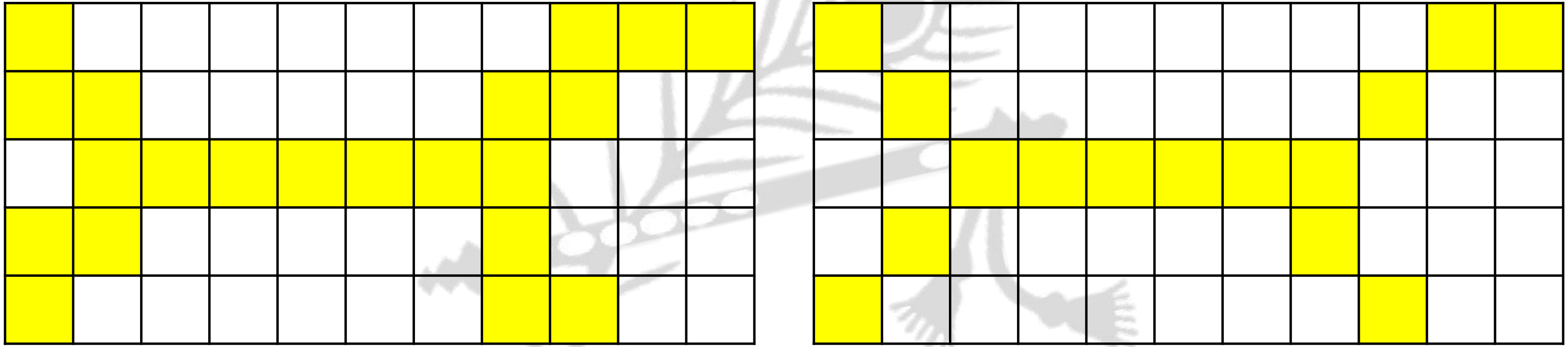
0	0	X
0	1	1
X	1	1

S_7

Thinning

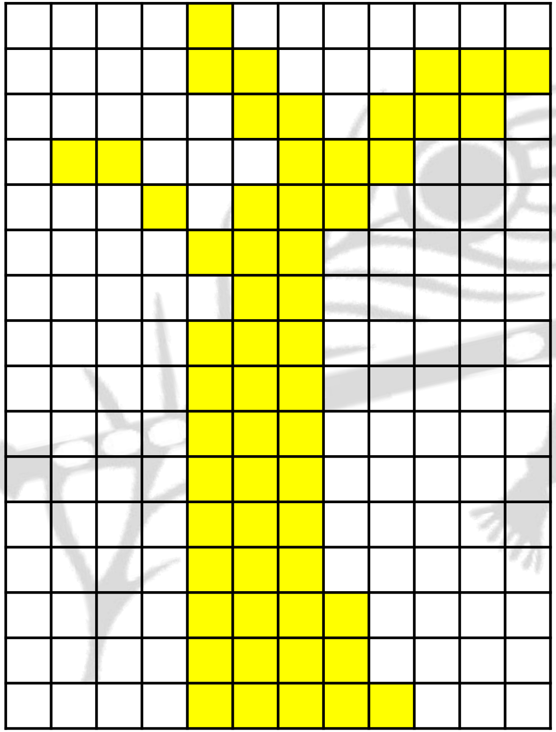


Thinning



Thinning

- Eg:



Thickening

- Thickening is the morphological dual of thinning
- It is defined as

$$I \odot S = I \bigcup (I \circledast S_i)$$

$$I \odot \{S\} = ((((I \circledast S_0) \circledast S_1) \circledast S_2) \ ...\circledast S_7)$$

- Approach
  - Take I^c
  - res = Apply thinning on I^c
  - Take res^c

$$I \odot \{S\} = |(I^c \otimes S)\,\text{followed by isolated pixel removal}\,|^c$$

RECAP

- Dilation $$I \oplus S = \{z \mid (\hat{S})_z \cap I \neq \phi\}$$
- Erosion $$I \Theta S = \{z \mid (S)_z \cap I^c = \phi\}$$
- Opening $$I \circ S = (I \Theta S) \oplus S$$
- Closing $$I \bullet S = (I \oplus S) \Theta S$$
- Hit or Miss $$I \circledast S = (I \Theta S) \cap (I^c \Theta (W - S))$$
- Boundary (Internal) $$\beta(I) = I - (I \Theta S)$$
- Boundary (External) $$\beta(I) = (I \oplus S) - \mathrm{I}$$

RECAP

- Region Filling $$X_k = (X_{k-1} \oplus S) \bigcap I^c$$
- Connected Components $$X_k = (X_{k-1} \oplus S) \bigcap I$$
- Convex Hull $$X_k = (X_{k-1} \circledast S_i) \bigcup I$$
- Thinning $$I \otimes \{S\} = ((((I \circledast S_0) \circledast S_1) \circledast S_2) \ldots \circledast S_7)$$
- Thickening $$I \odot \{S\} = |(I^c \otimes S)\,\text{isolated pixel removal}|^c$$