Introduction to Morphological Operators

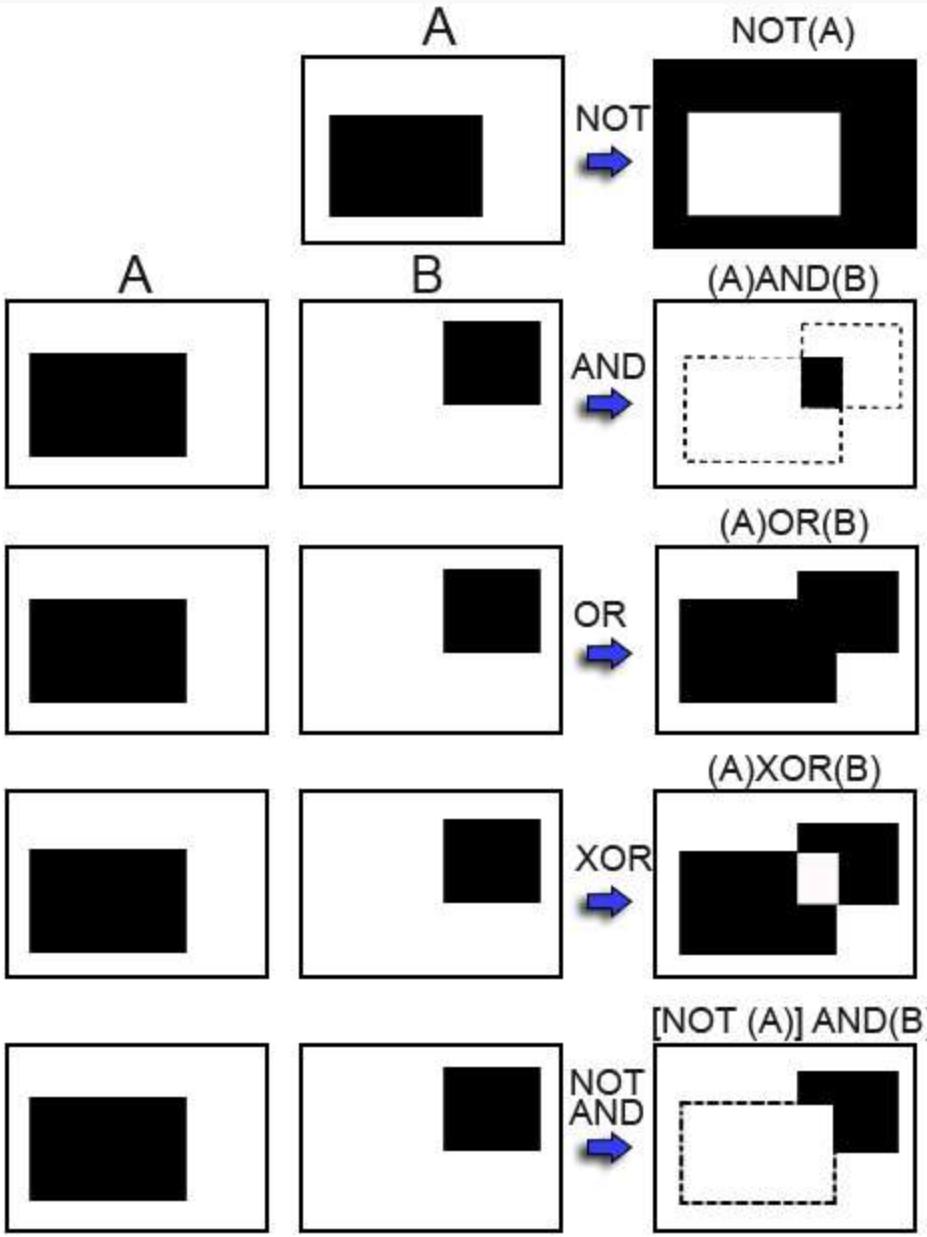
9.1 Basic Concepts in Set Theory

- Subset
  $A \subseteq B$
- Union
  $A \cup B$
- Intersection
  $A \cap B$
  disjoint / mutually exclusive $A \cap B = \emptyset$
- Complement $A^c \equiv \{ w \mid w \notin A \}$
- Difference $A - B \equiv \{ w \mid w \in A, w \notin B \} = A \cap B^c$
- Reflection $B \equiv \{ w \mid w = -b, \quad \forall b \in B \}$
- Translation $(A)z \equiv \{ c \mid c = a + z, \quad \forall a \in A \}$

Logic Operations Involving Binary Pixels and Images

- The principal logic operations used in image processing are: **AND, OR, NOT (COMPLEMENT)**.
- These operations are *functionally complete*.
- Logic operations are preformed on a pixel by pixel basis between corresponding pixels (bitwise).
- Other important logic operations :

  XOR (exclusive OR), NAND (NOT-AND)
- Logic operations are just a private case for a **binary set operations**, such : AND – Intersection , OR – Union, NOT-Complement.


A
NOT(A)
NOT
A
B
(A)AND(B)
AND
(A)OR(B)
OR
(A)XOR(B)
XOR
[NOT (A)] AND(B)
NOT
AND

Preview

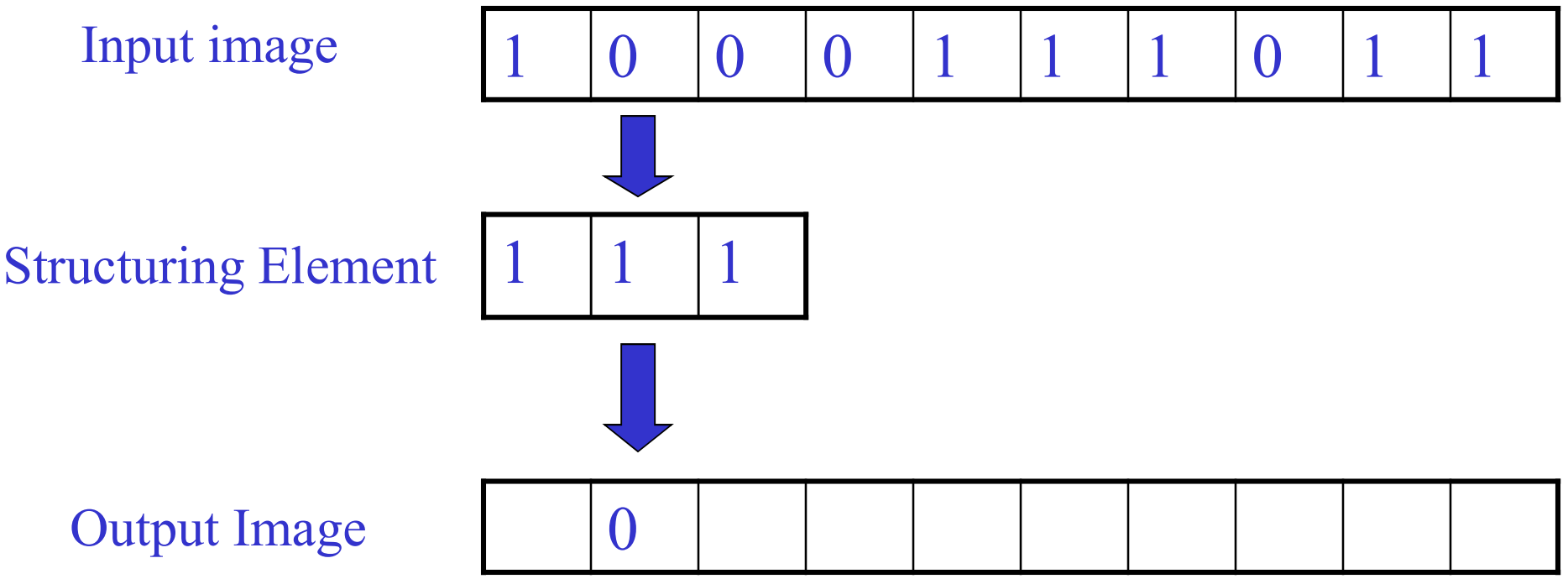
- ***"Morphology "*** – a branch in biology that deals with the form and structure of animals and plants.
- **"*Mathematical Morphology*"** – as a tool for extracting image components, that are useful in the representation and description of region shape.
- The language of mathematical morphology is – **Set theory**.
- Unified and powerful approach to numerous image processing problems.
- In binary images , the set elements are members of the 2-D integer space – Z^2. where each element *(x,y)* is a coordinate of a black (or white) pixel in the image.

Learning Materials

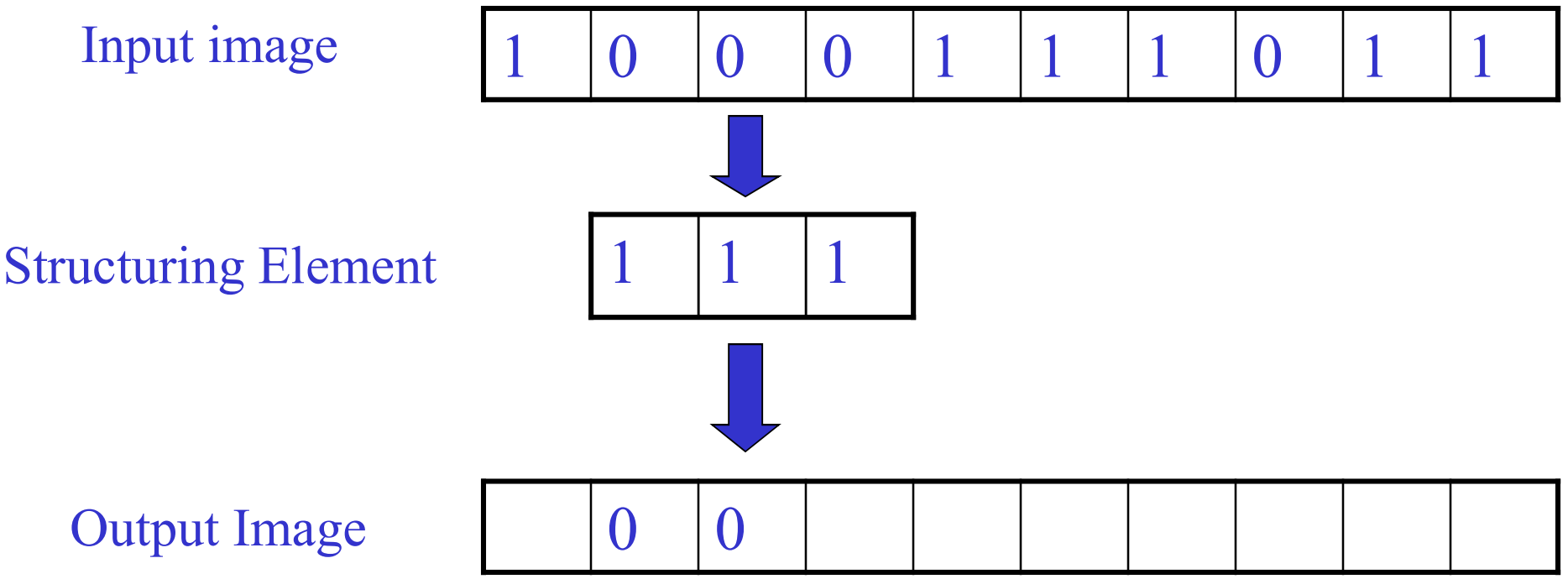
- Structuring Element
- Erosion
- Dilation
- Opening
- Closing
- Hit-and-miss
- Thinning
- Thickening

1D Morphological Operations

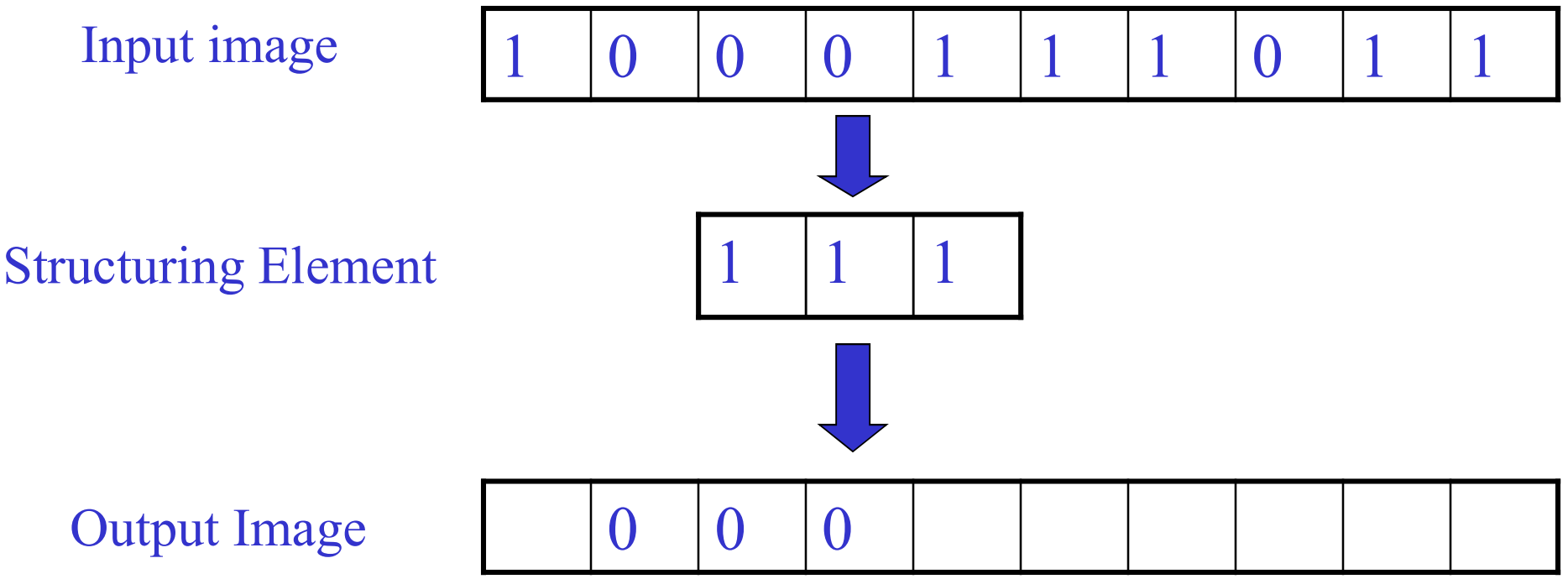
Example for 1D Erosion



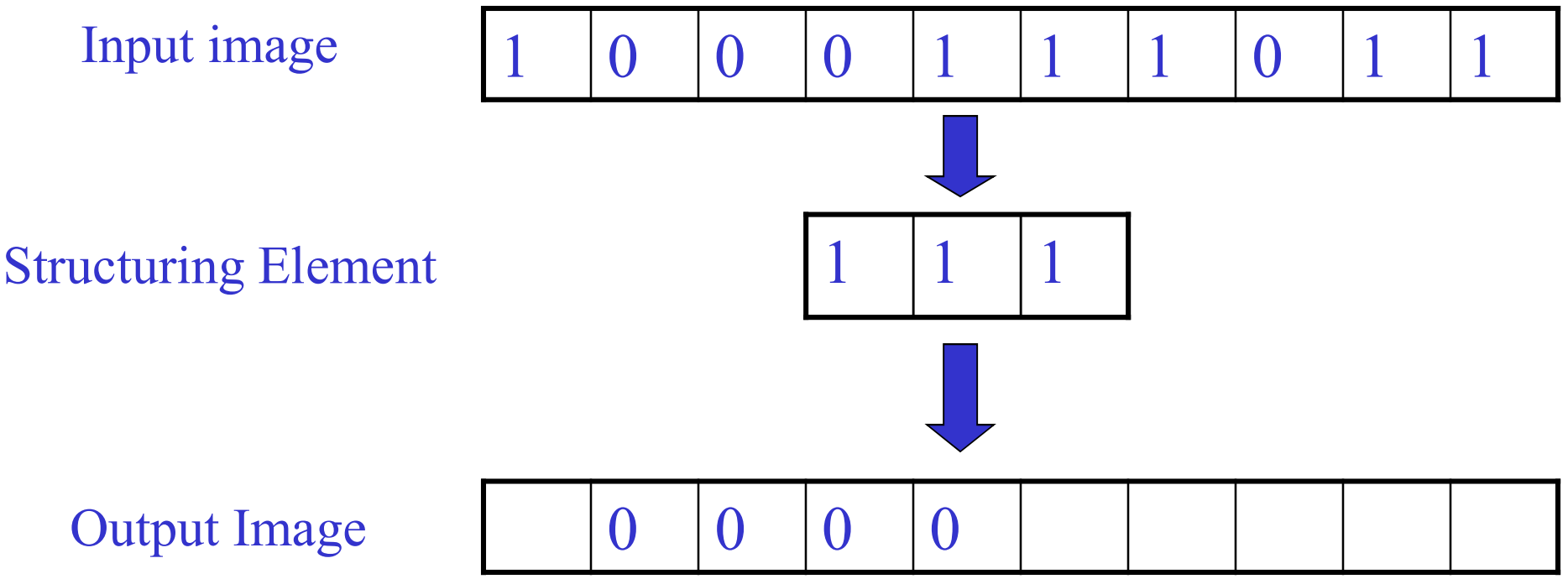
Example for Erosion



Example for Erosion

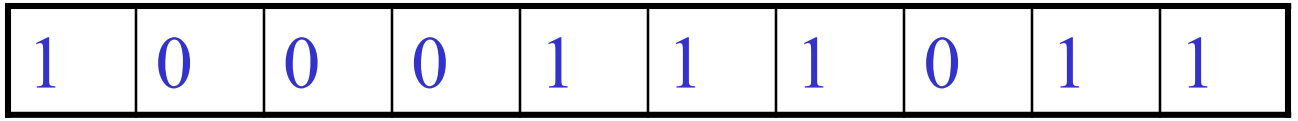


Example for Erosion

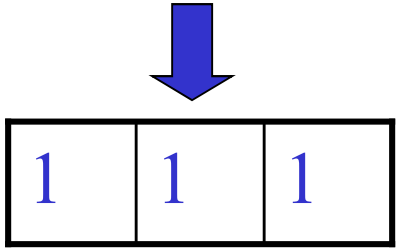


Example for Erosion

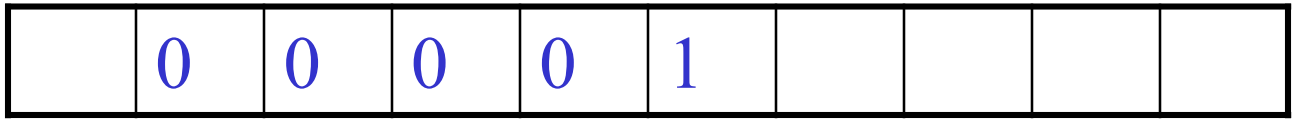
Input image



Structuring Element

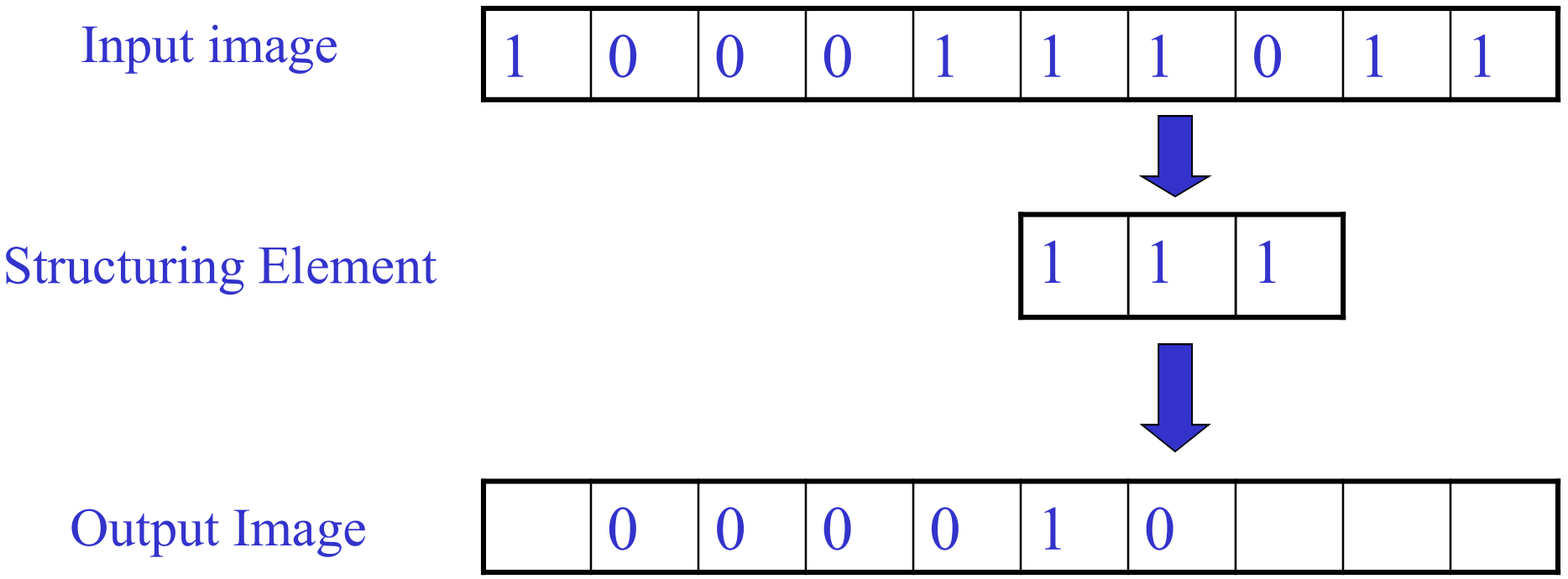


Output Image

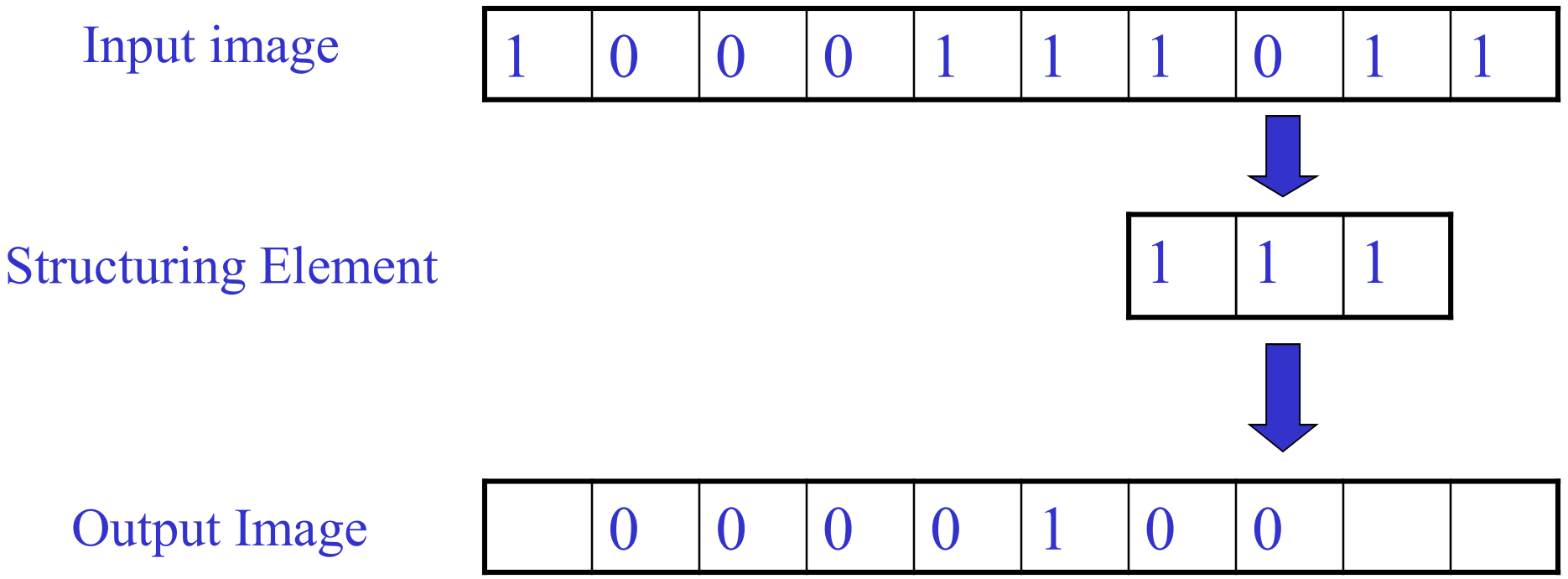


Structured element is completely included in set of ones

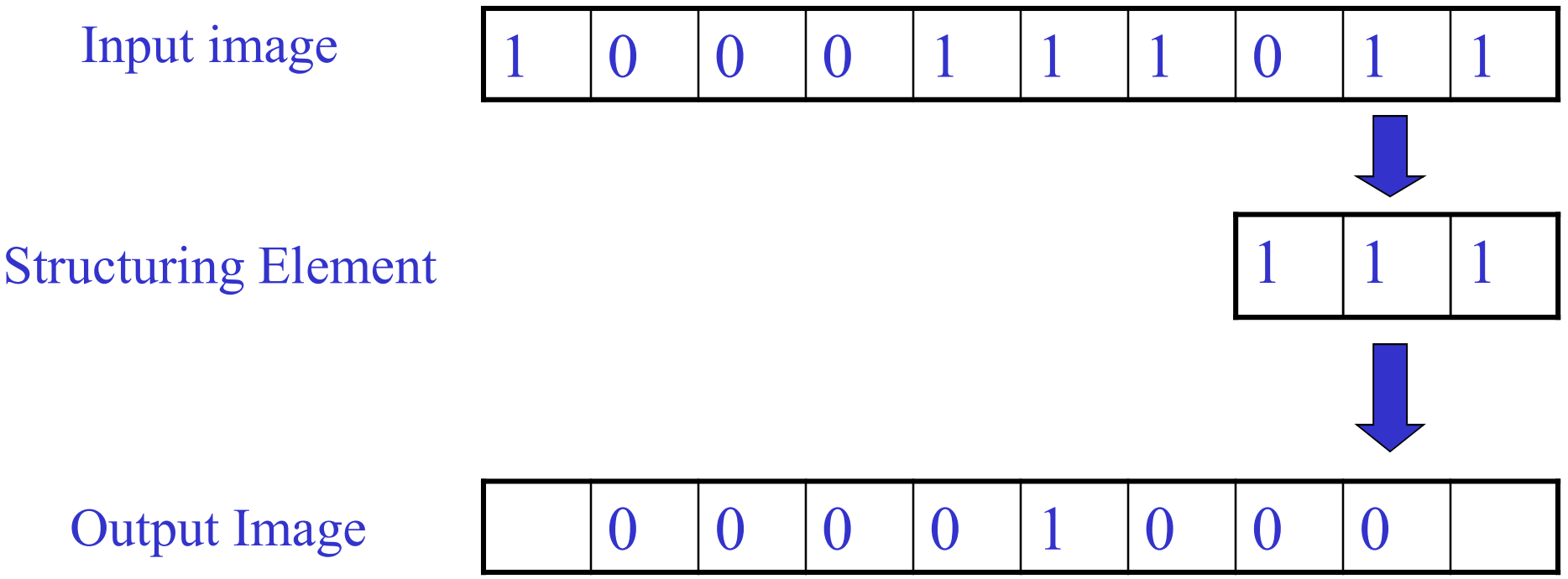
Example for Erosion



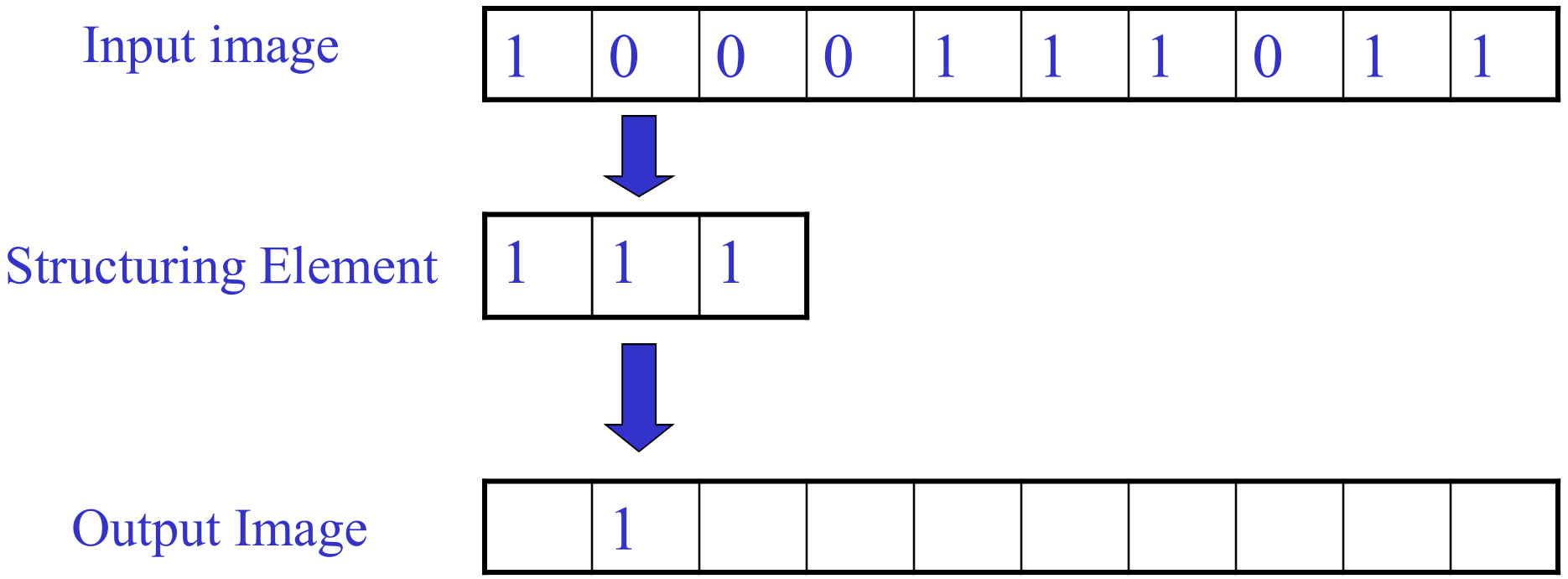
Example for Erosion



Example for Erosion

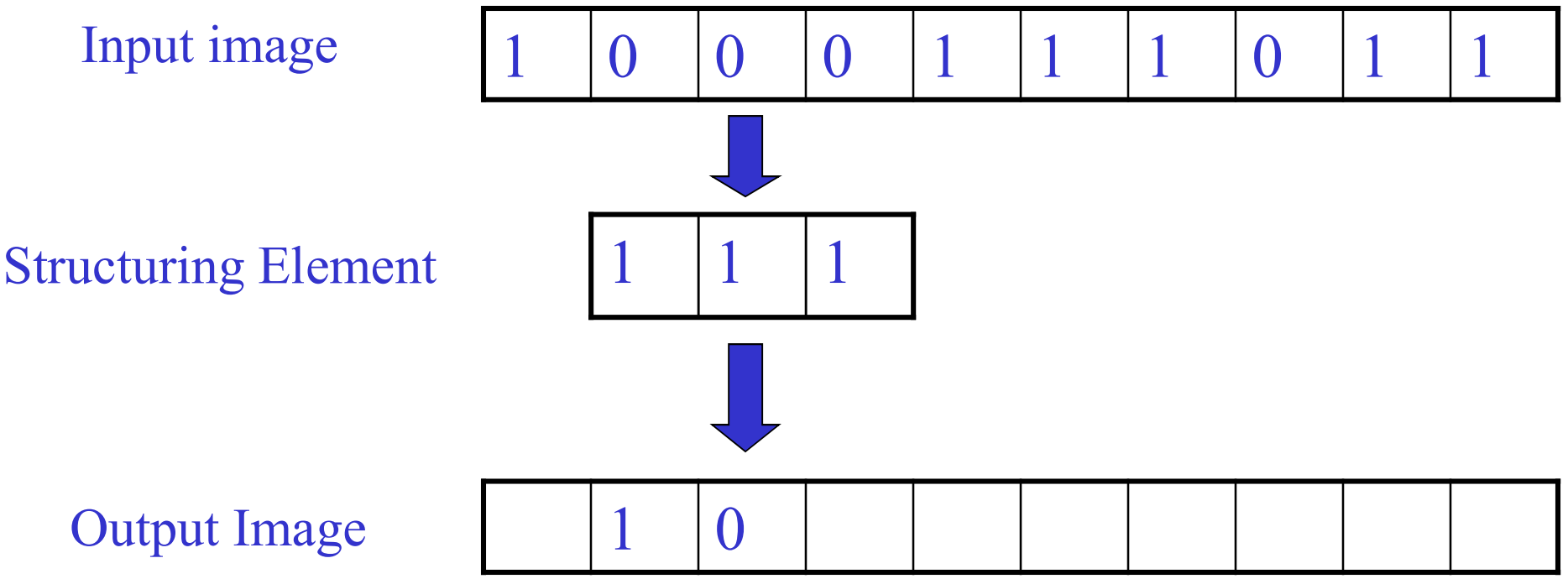


Example for 1D Dilation

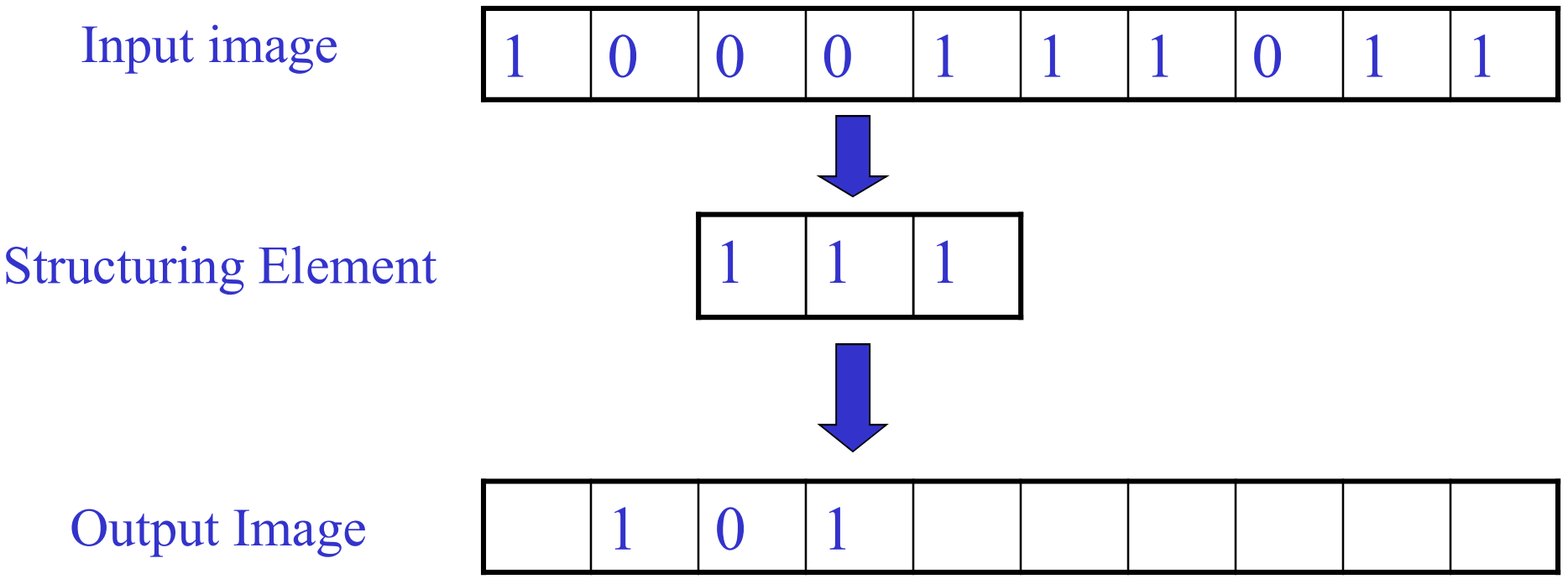


Structuring element overlaps with input image

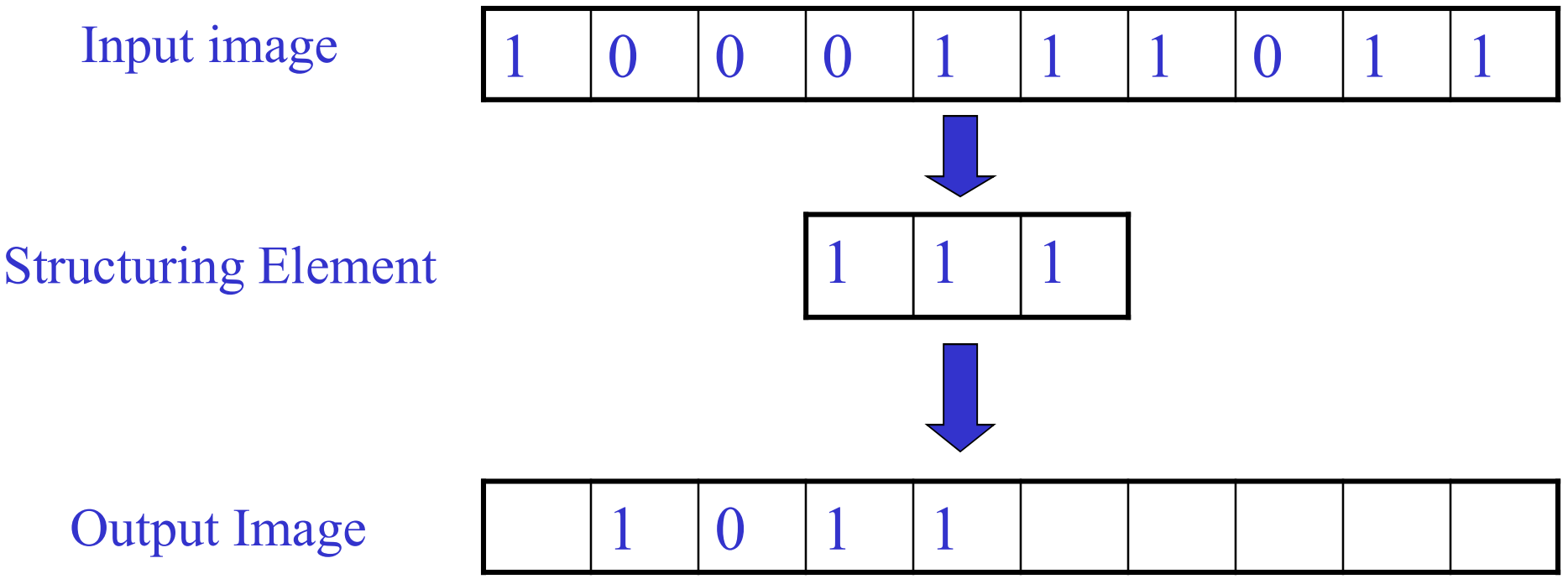
Example for Dilation



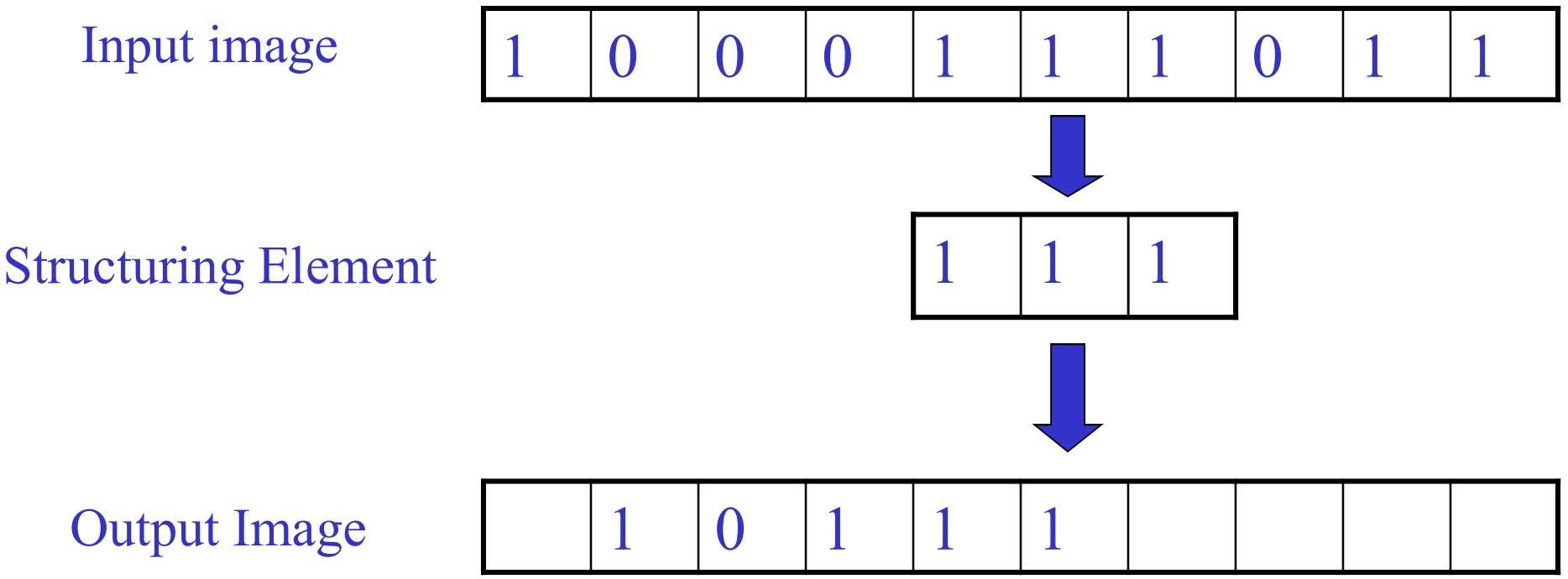
Example for Dilation



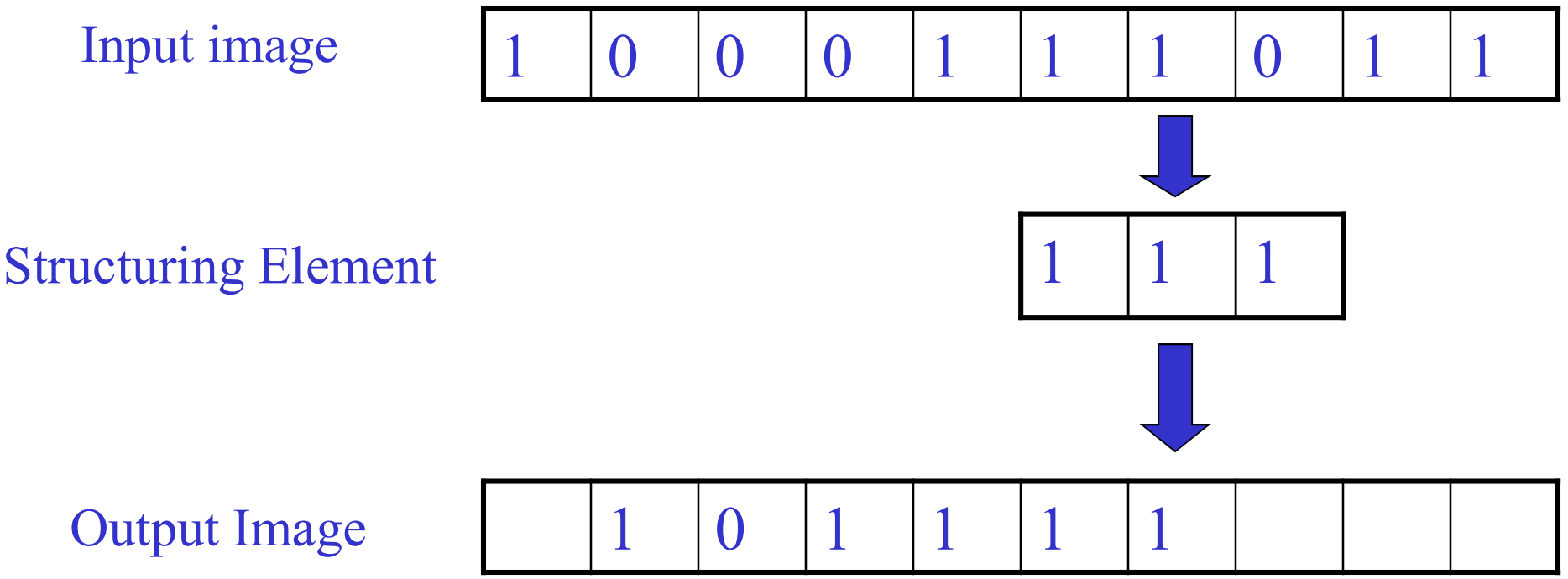
Example for Dilation



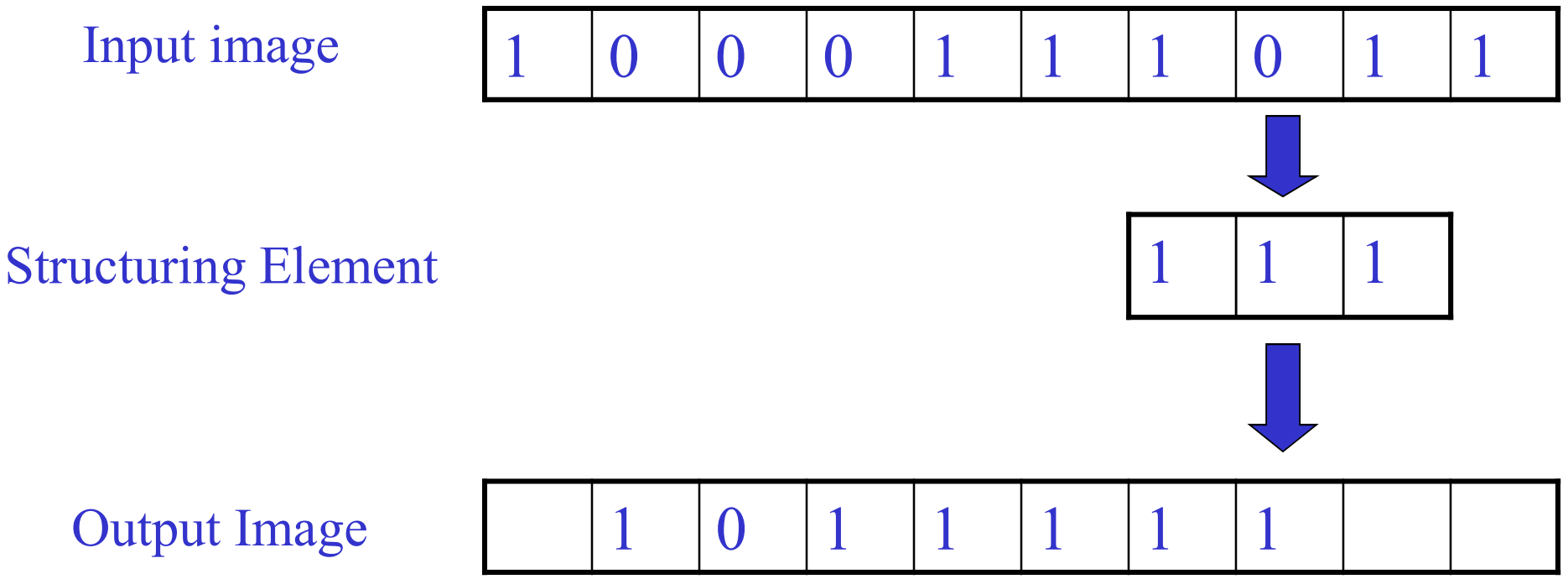
Example for Dilation



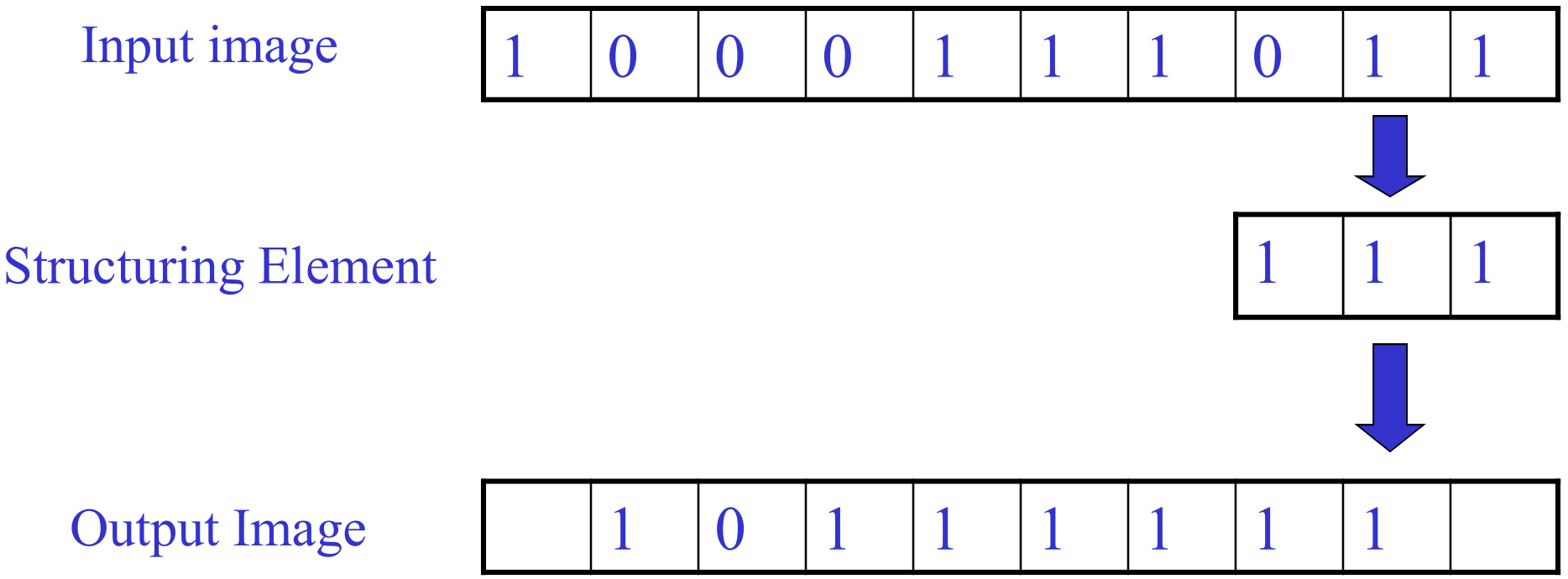
Example for Dilation



Example for Dilation



Example for Dilation



2D Morphological Operations

Image – Set of Pixels

- Basic idea is to treat an object within an image as a set of pixels (or coordinates of pixels)

- In binary images, pixels that are ‘off”, set to 0, are background and appear black.

- Foreground pixels (objects) are 1 and appear white

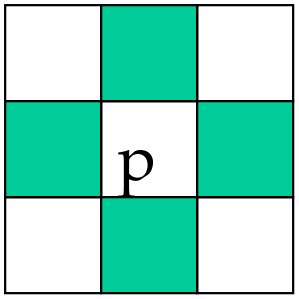
Neighborhood

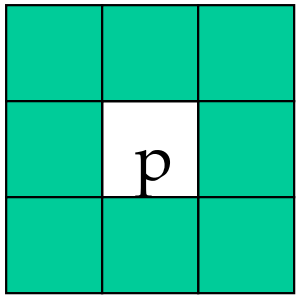
- Set of pixels defined by their location relation to the pixel of interest
  - Defined by structuring element
  - Specified by connectivity

- Connectivity-
  - ‘4-connected’
  - ‘8-connected’

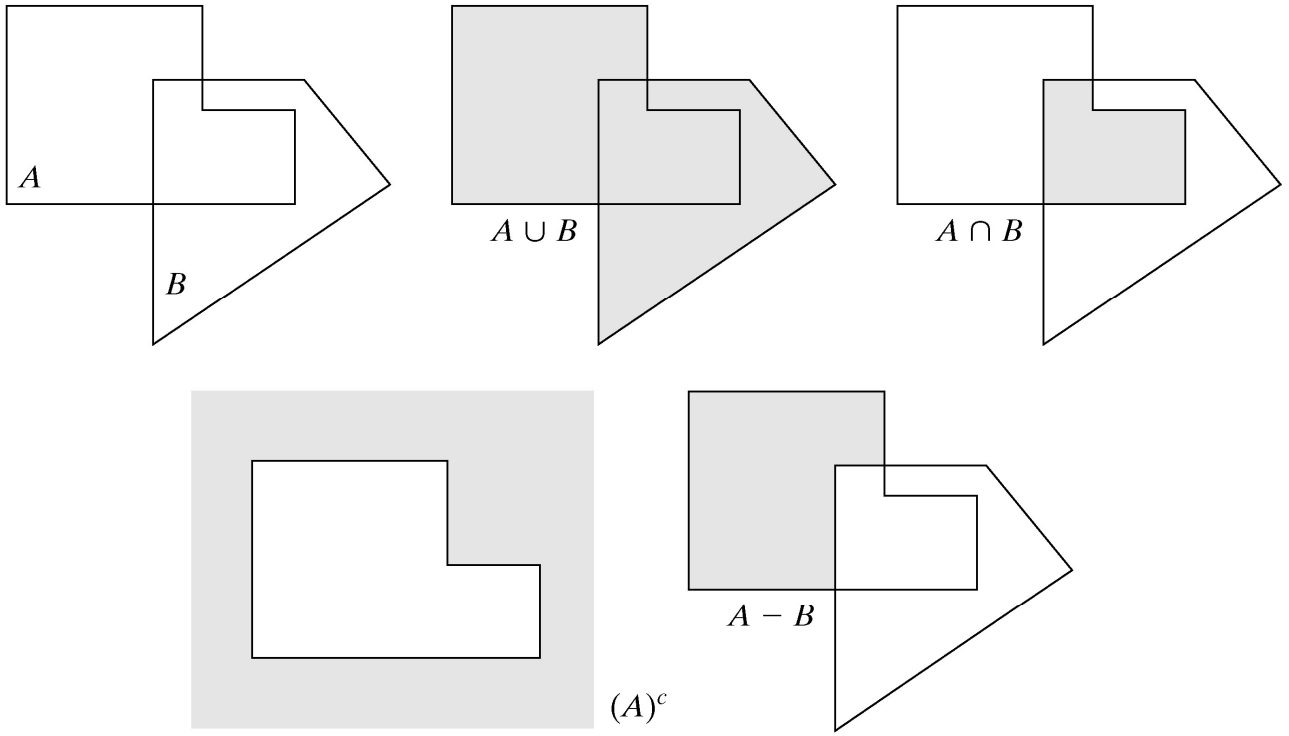
Labeling Connected Components

- Label objects in an image
- 4-Neighbors
- 8-Neighbors





Morphological Image Processing



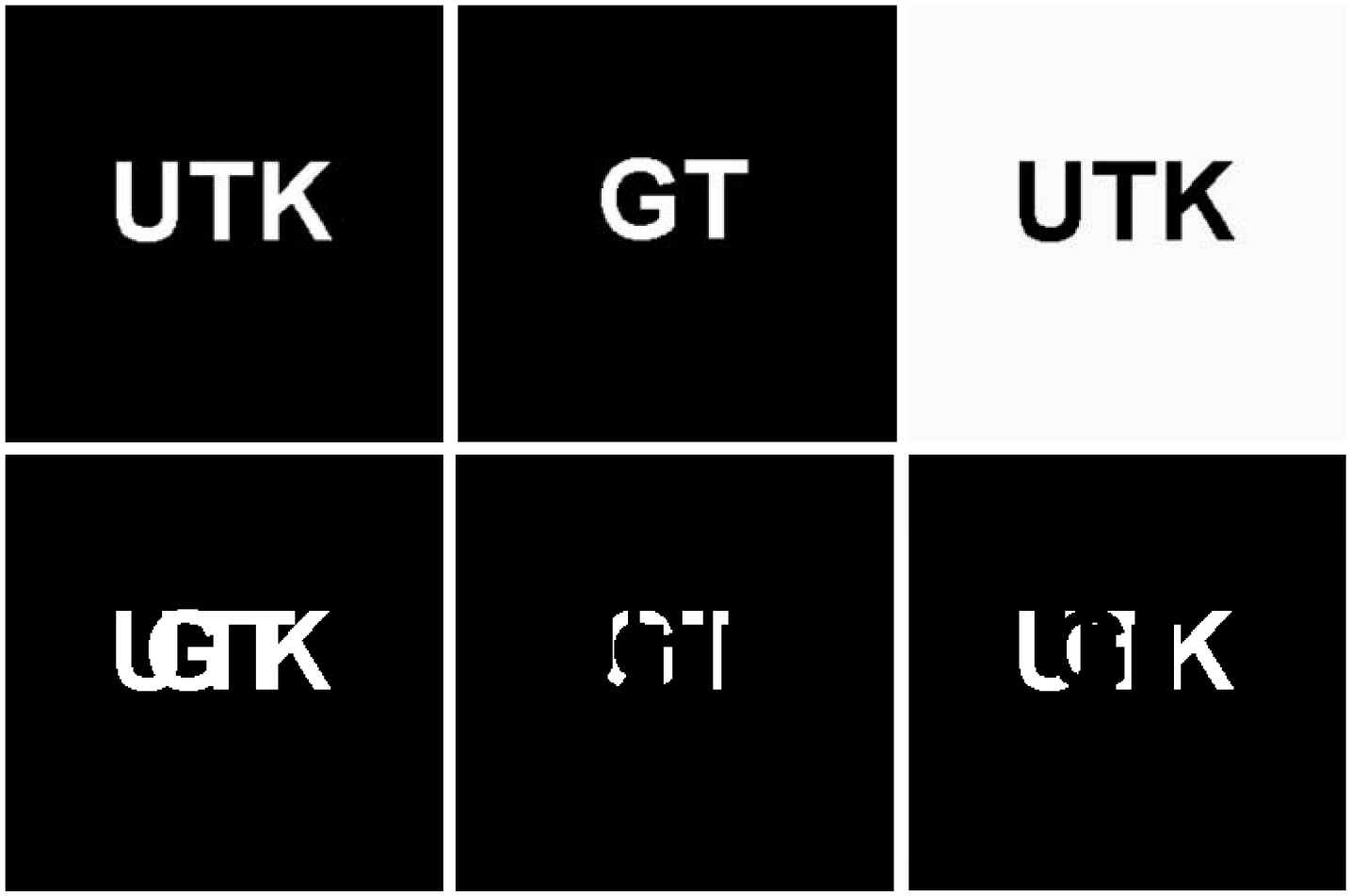
a b c
d e

FIGURE 9.1
(a) Two sets A and B. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B.

Basic operations on shapes

From: Digital Image Processing, Gonzalez,Woods And Eddins

Morphological Image Processing



a b c
d e f

FIGURE 9.3 (a) Binary image A. (b) Binary image B. (c) Complement ~A. (d) Union A | B. (e) Intersection A & B. (f) Set difference A & ~B.

From: Digital Image Processing, Gonzalez,Woods
2nd Edition

Operations on binary images in MATLAB

Set Operation	MATLAB Expression for Binary Images	Name
$A \cap B$	A & B	AND
$A \cup B$	A | B	OR
A^c	~A	NOT
$A - B$	A & ~B	DIFFERENCE

TABLE 9.1 Using logical expressions in MATLAB to perform set operations on binary images.

From: Digital Image Processing, Gonzalez,Woods And Eddins

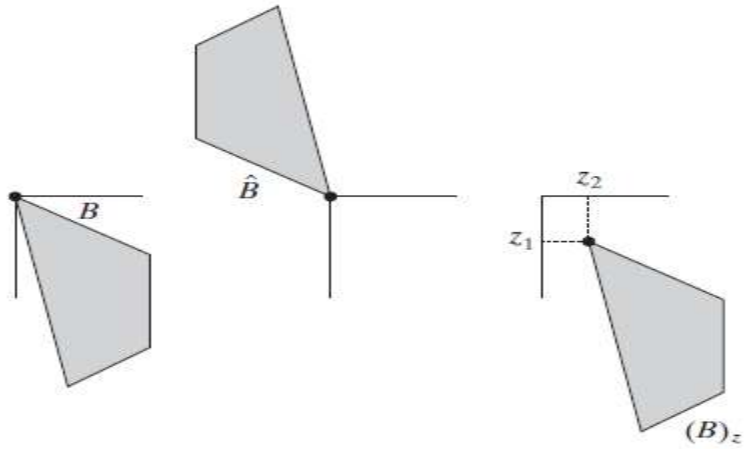
Reflection

- *The reflection of* set B is defined as

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

a b c

FIGURE 9.1
(a) A set, (b) its reflection, and (c) its translation by z.



Translation

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

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

If B is the set of pixels representing an object in an image, then $(B)_z$ is the set of points in B whose (x, y) coordinates have been replaced by $(x + z_1, y + z_2)$. Figure 9.1(c) illustrates this concept using the set B from Fig. 9.1(a).

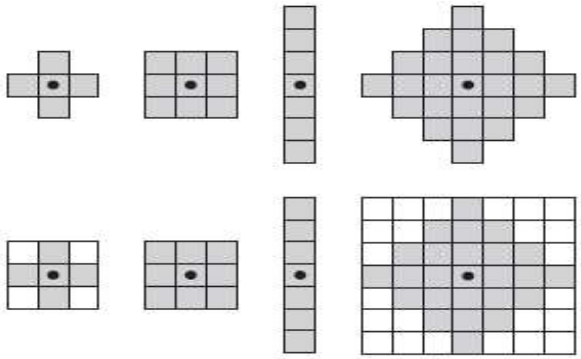
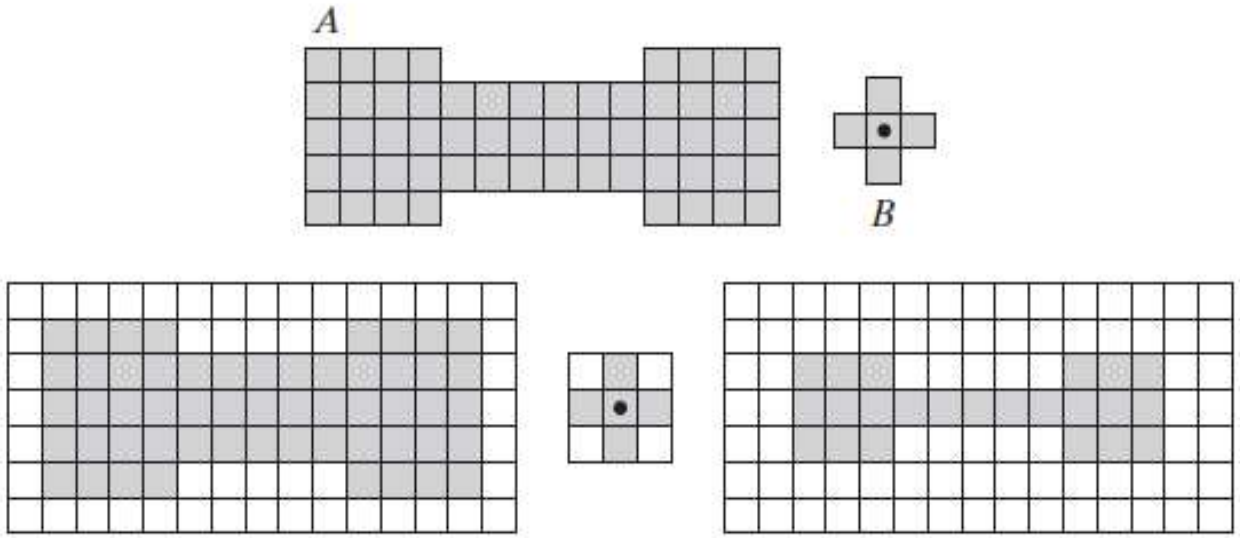


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Translation



a b
c d e

FIGURE 9.3 (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

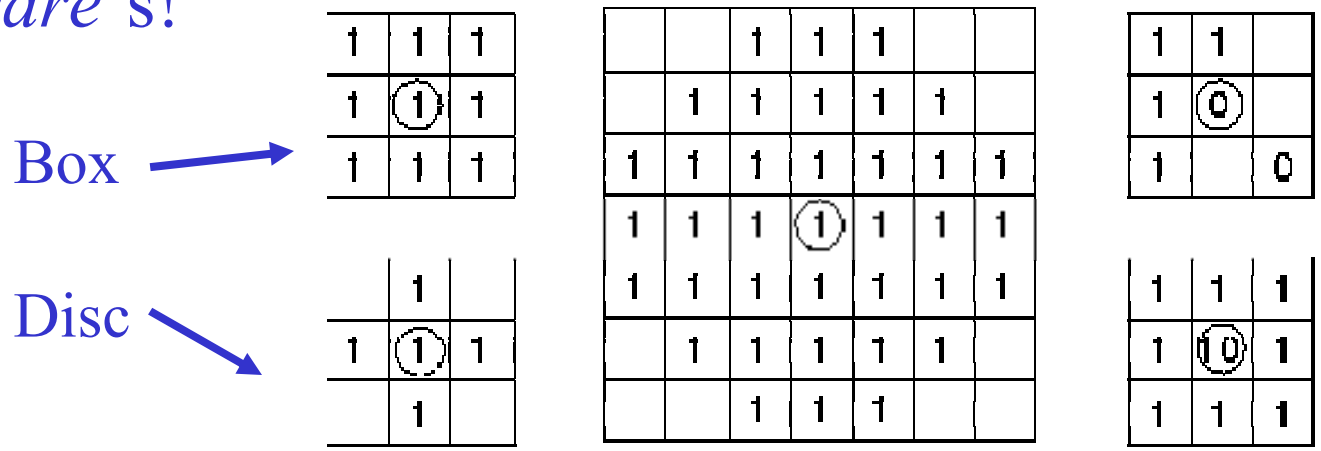
DILATION

- Define Definition of DILATION is the UNION of all the translations:

$A \oplus B = \cup \{ t \in I^2 : t = a+b, a \in A \}$ for all b's in B

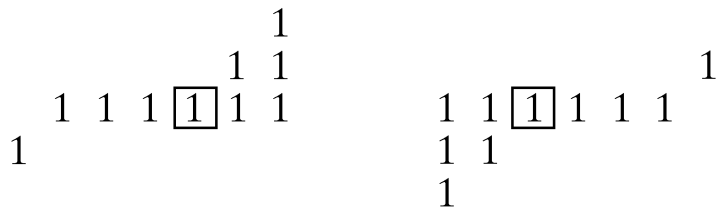
Structuring Element (Kernel)

- Structuring Elements can have varying sizes
- Usually, element values are 0,1 and none(!)
- Structural Elements have an origin
- For thinning, other values are possible
- Empty spots in the Structuring Elements are *don't care*'s!



Examples of stucturing elements

Reflection of the structuring element



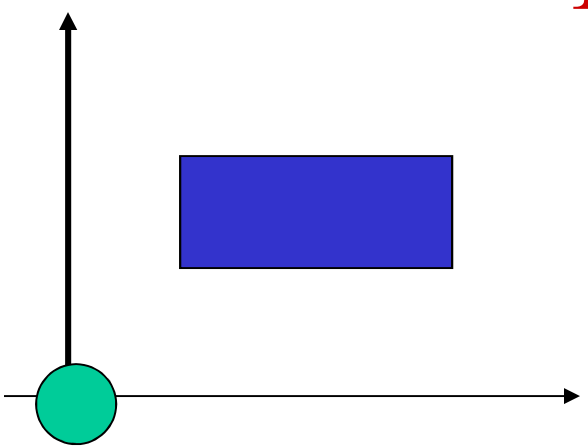
a b

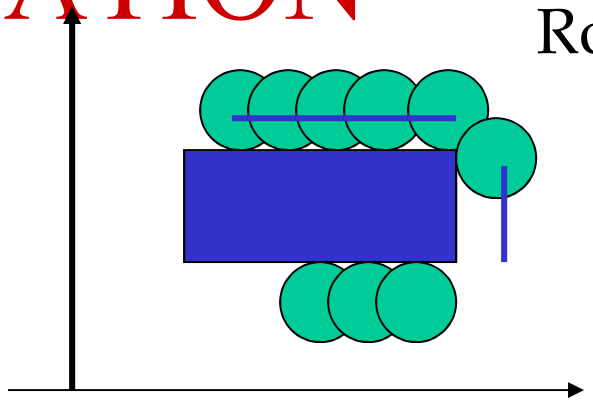
FIGURE 9.5 Structuring element reflection. (a) Nonsymmetric structuring element. (b) Structuring element reflected about its origin.

From: Digital Image Processing, Gonzalez,Woods And Eddins

2D DILATION

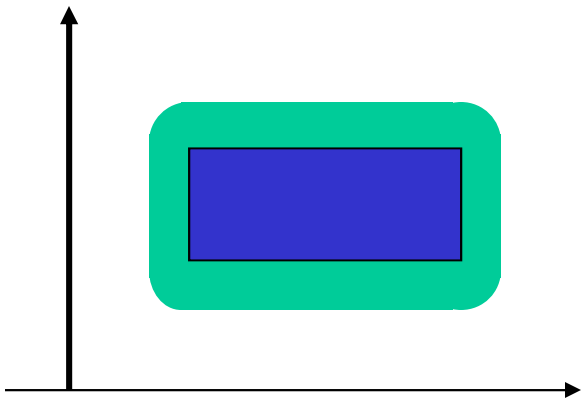
DILATION



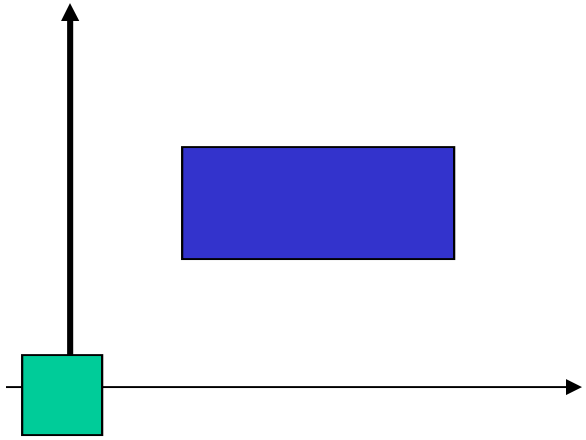


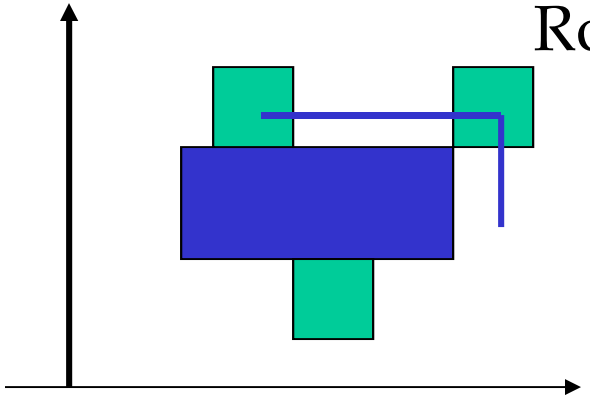
Rounded corners

Round Structuring Element (SE) can be interpreted
as rolling the SE around the contour of the object.
New object has rounded corners and is larger *in every direction* by
½ width of the SE

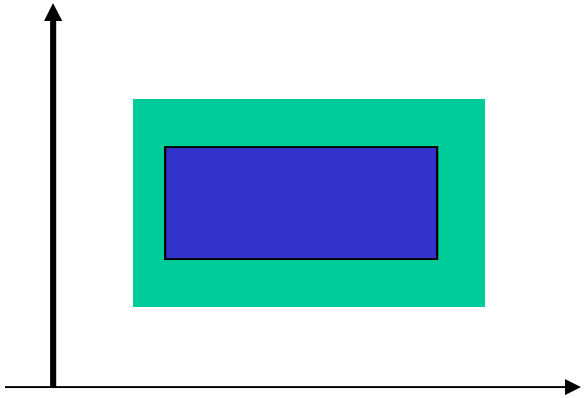


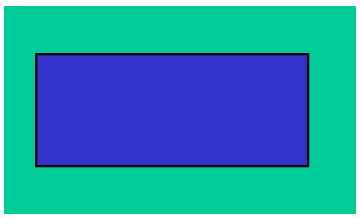
DILATION





Rounded corners





Square Structuring Element (SE)
can be interpreted
as moving the SE around the
contour of the object.
New object has square corners and
is larger *in every direction* by
½ width of the SE

DILATION

- The shape of B determines the final shape of the dilated object.

- B acts as a geometric filter that changes the geometric structure of A

Another important operator

- Introduction to **Morphological Operators**
  - Used generally on binary images, e.g., background subtraction results!
  - Used on gray value images, if viewed as a stack to binary images.
- Good for, e.g.,
  - Noise removal in background
  - Removal of holes in foreground / background
- Check: **www.cee.hw.ac.uk/hipr**

9.2.1 Dilation

Dilation is used for expanding an element A by using structuring element B

- Dilation of A by B and is defined by the following equation:

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} \quad (9.2-1)$$

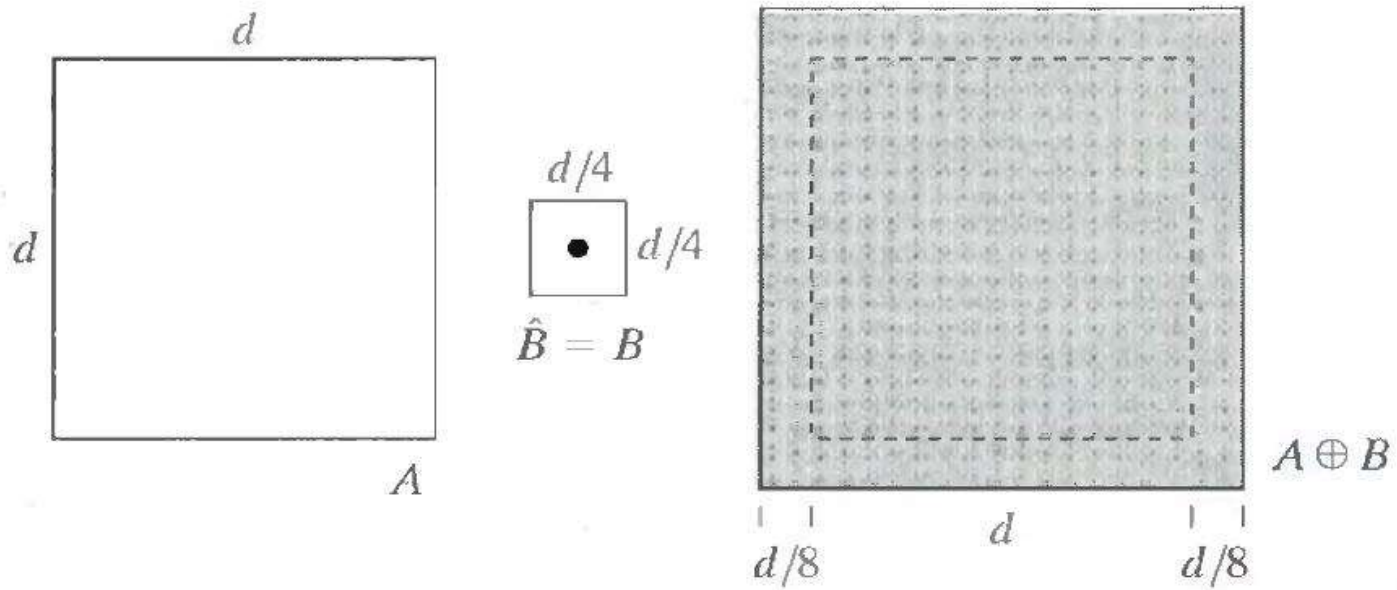
- This equation is based 0n obtaining the reflection 0f B about its origin and shifting this reflection by z.
- The dilation of A by B is the set of all displacements z, such that $\hat{B}$ and A overlap by at least one element. Based On this interpretation the equation of (9.2-1) can be rewritten as:

$$A \oplus B = \{z | [(\hat{B})z \cap A] \subset A\} \quad (9.2-2)$$

9.2.1 Dilation – Example 1

a b c

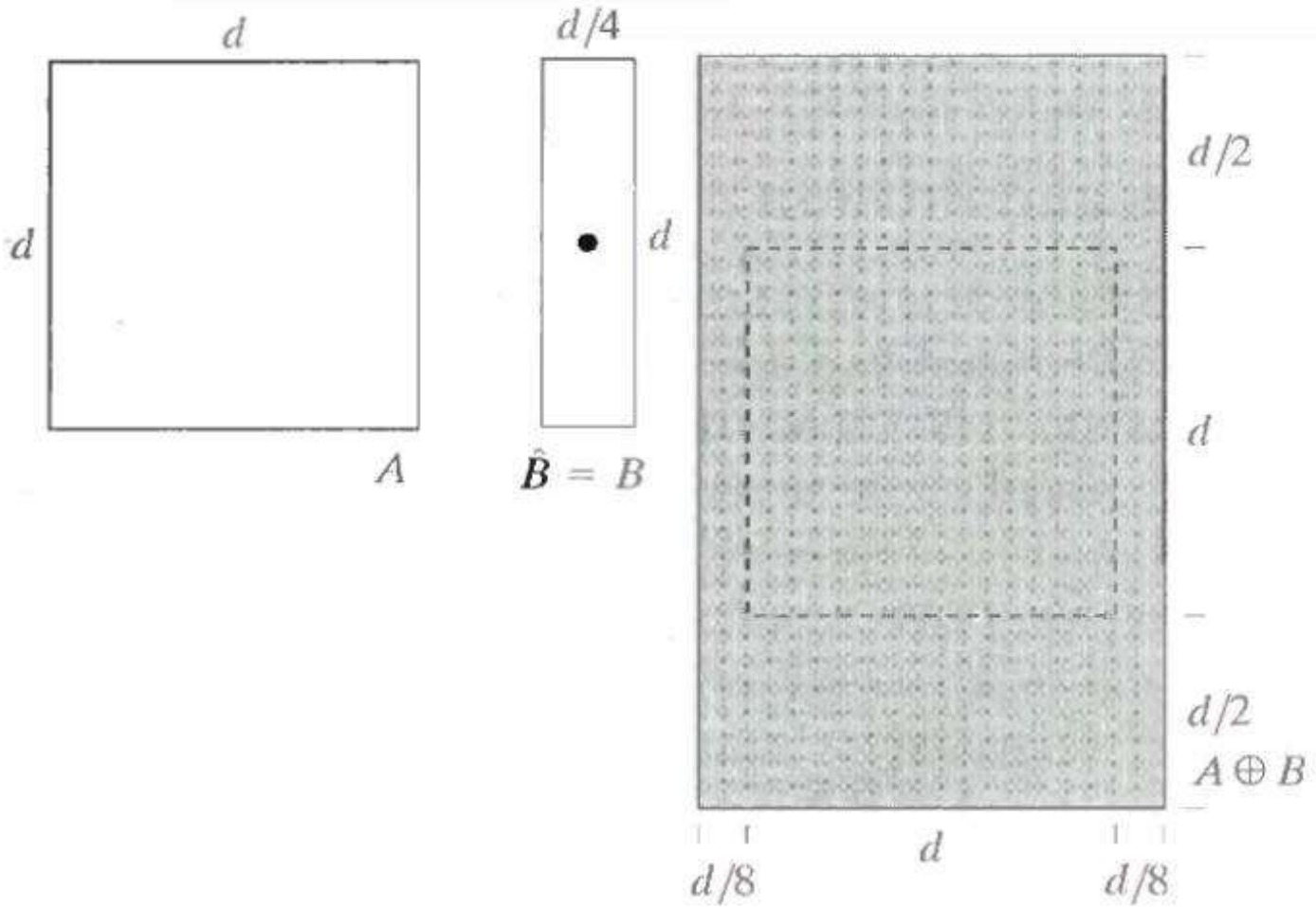
FIGURE 9.4
(a) Set A.
(b) Square structuring element (dot is the center).
(c) Dilation of A by B, shown shaded.



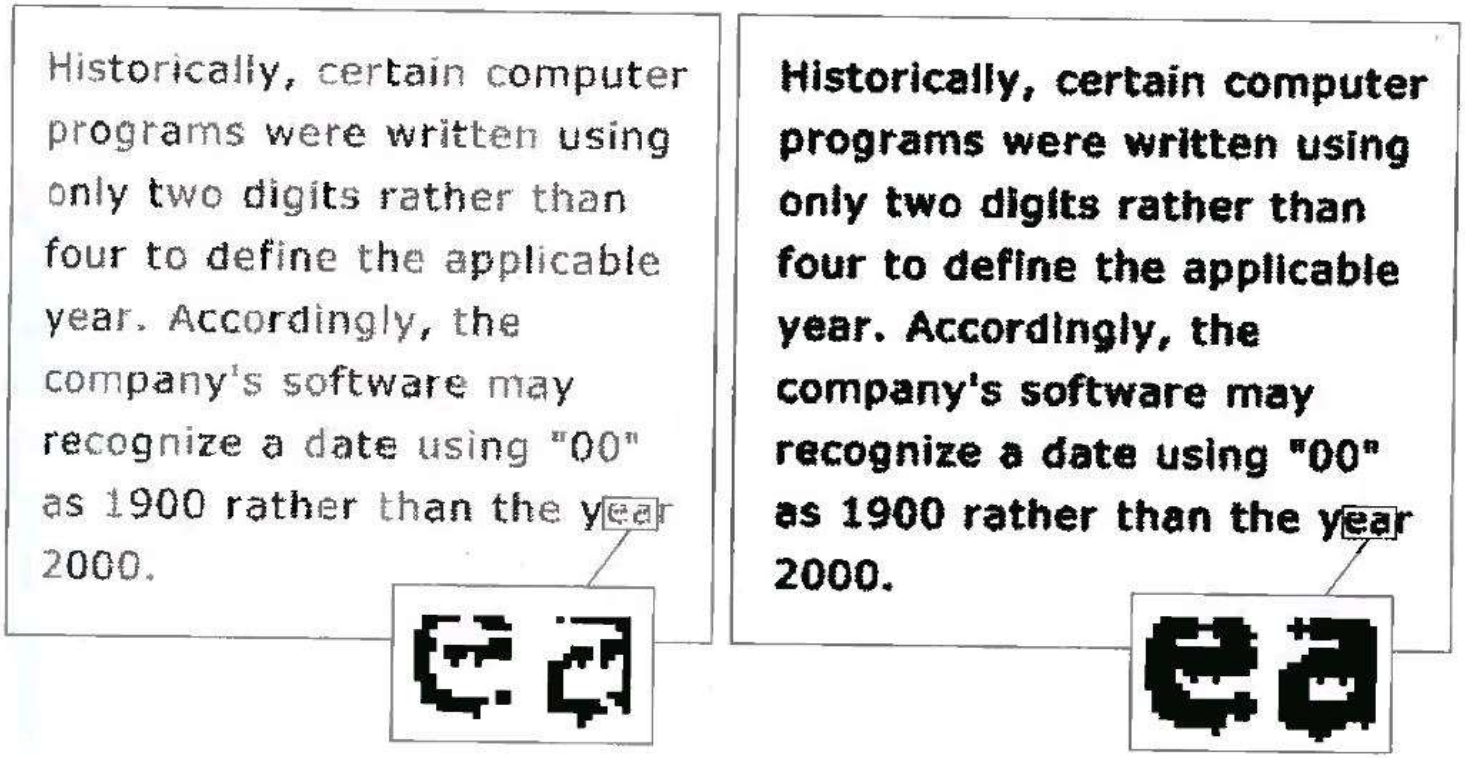
9.2.1 Dilation – Example 2

a d e

(d) Elongated structuring element.
(e) Dilation of A using this element.



9.2.1 Dilation – A More interesting Example



a c
b

FIGURE 9.5
(a) Sample text of poor resolution with broken characters (magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

0	1	0
1	1	1
0	1	0

9.2.2 Erosion

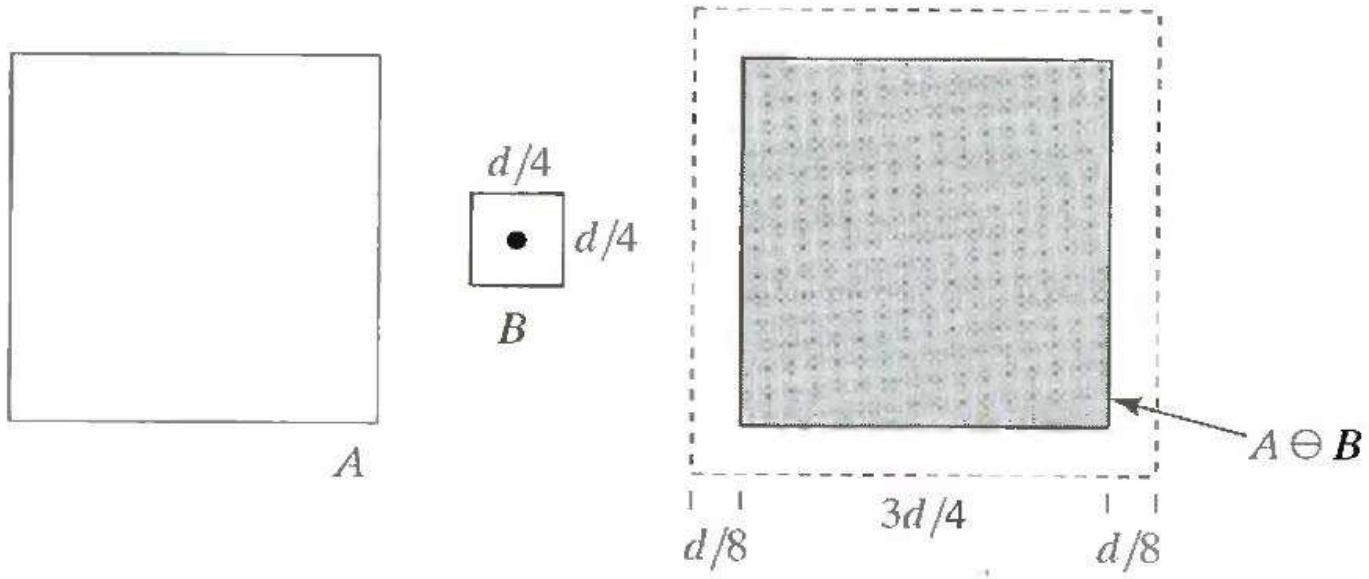
Erosion is used for shrinking of element A by using element B

- Erosion for Sets A and B in Z^2, is defined by the following equation:

$$A \ominus B = \{z|[(B)z \subseteq A\} \quad (9.2-3)$$

- This equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is combined in A.

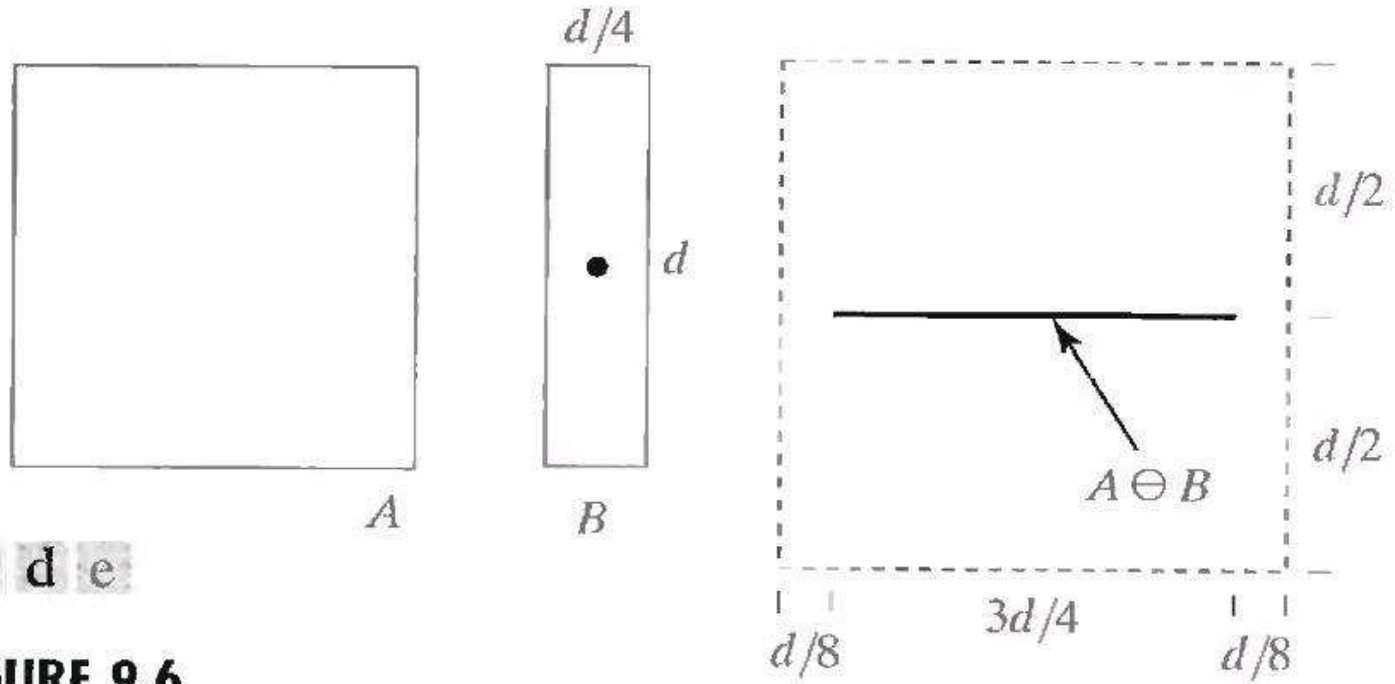
9.2.2 Erosion – Example 1



a b c

FIGURE 9.6 (a) Set A. (b) Square structuring element. (c) Erosion of A by B, shown shaded

9.2.2 Erosion–Example 2



a d e

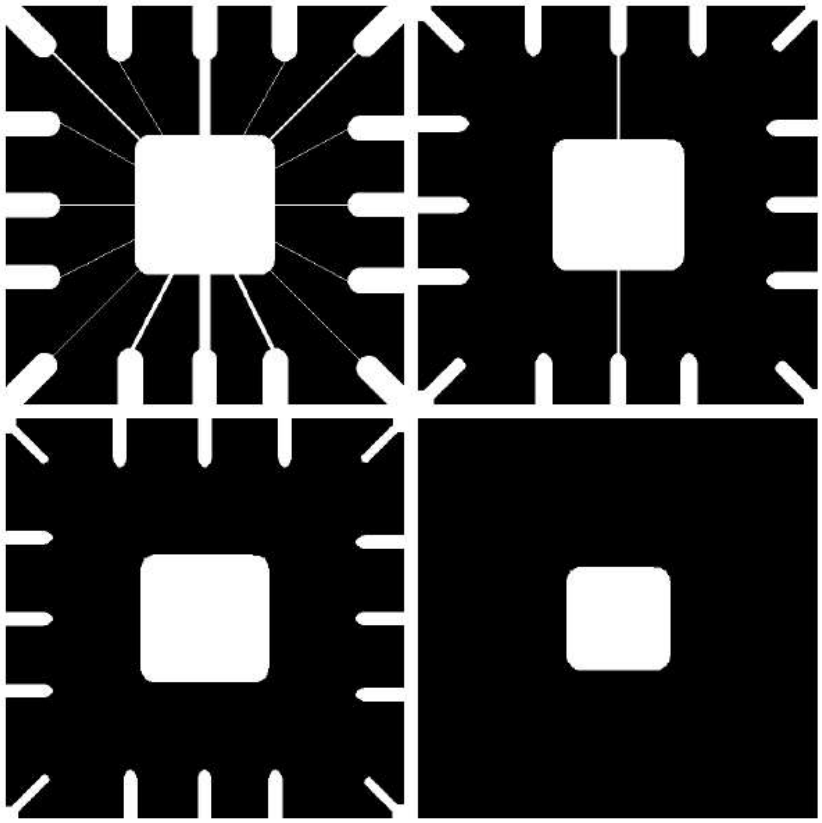
FIGURE 9.6

(a) Set A. (d) Elongated structuring element. (e) Erosion of A using this element.

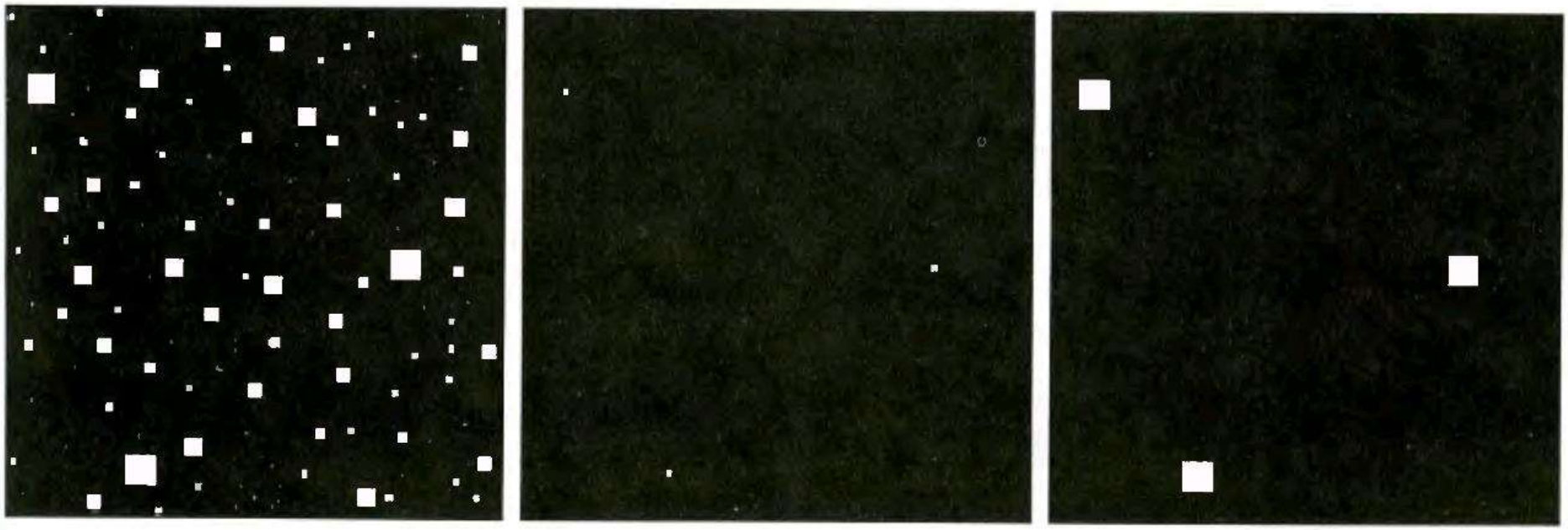
9.2.2 Erosion–Example 3

a b
c d

FIGURE 9.5 Using erosion to remove image components. (a) A 486 × 486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11 × 11, 15 × 15, and 45 × 45, respectively. The elements of the SEs were all 1s.



Erosion and Dilation summary



a b c

FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

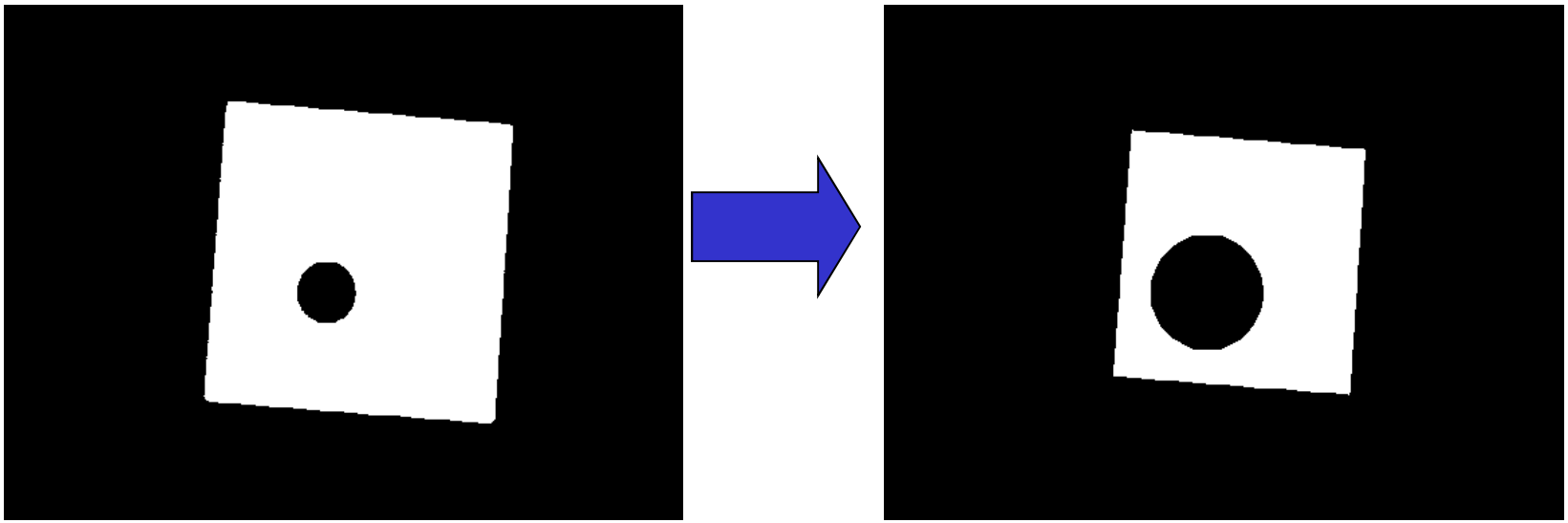
Dilation versus Erosion

- Basic operations
- Are dual to each other:
  - Erosion shrinks foreground, enlarges Background
  - Dilation enlarges foreground, shrinks background

Erosion

- **Erosion** is the set of all points in the image, where the structuring element “fits into”.
- Consider each foreground pixel in the input image
  - If the structuring element fits in, write a “1” at the origin of the structuring element!
- Simple application of **pattern matching**
- **Input:**
  - **Binary Image (Gray value)**
  - **Structuring Element, containing only 1s!**

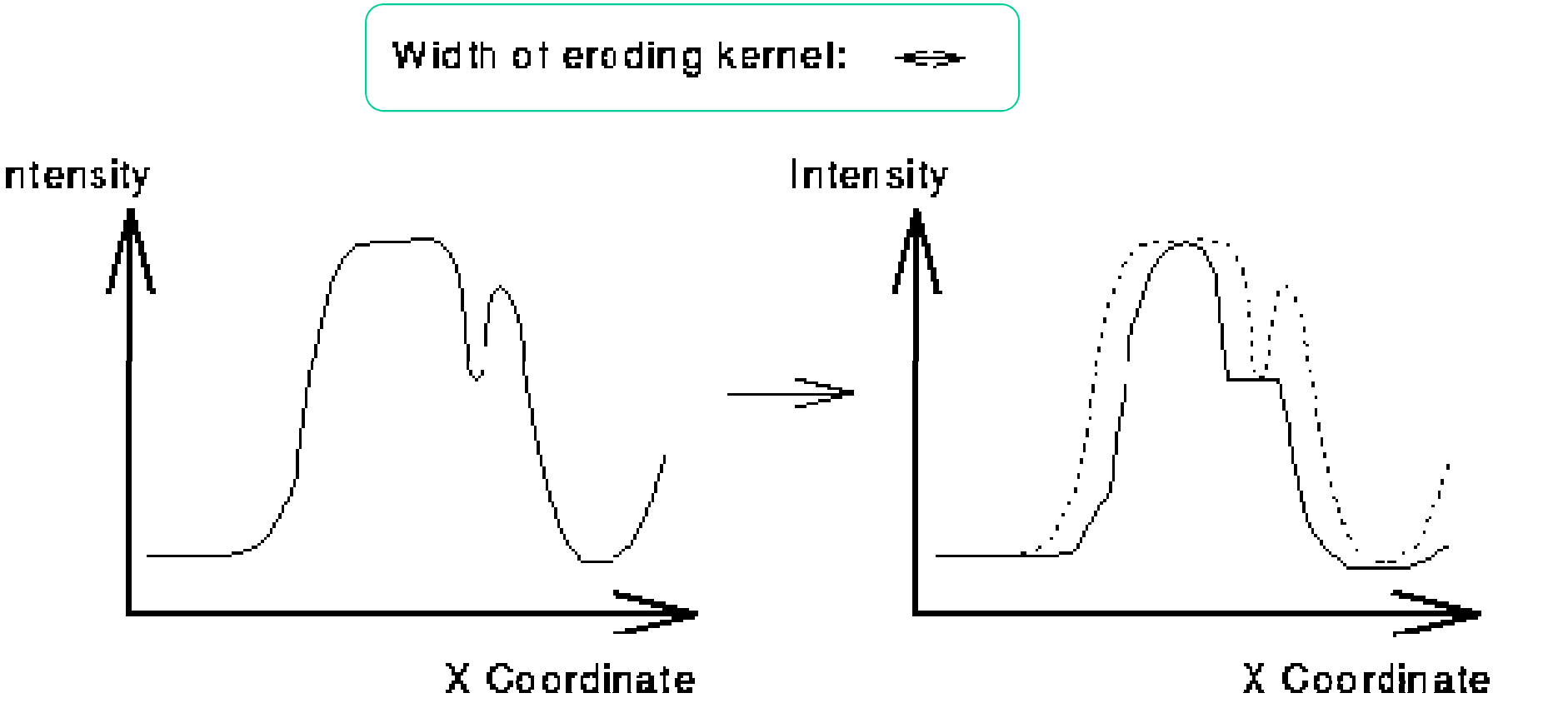
Another example of erosion



- White = 0, black = 1, dual property, image as a result of erosion gets darker

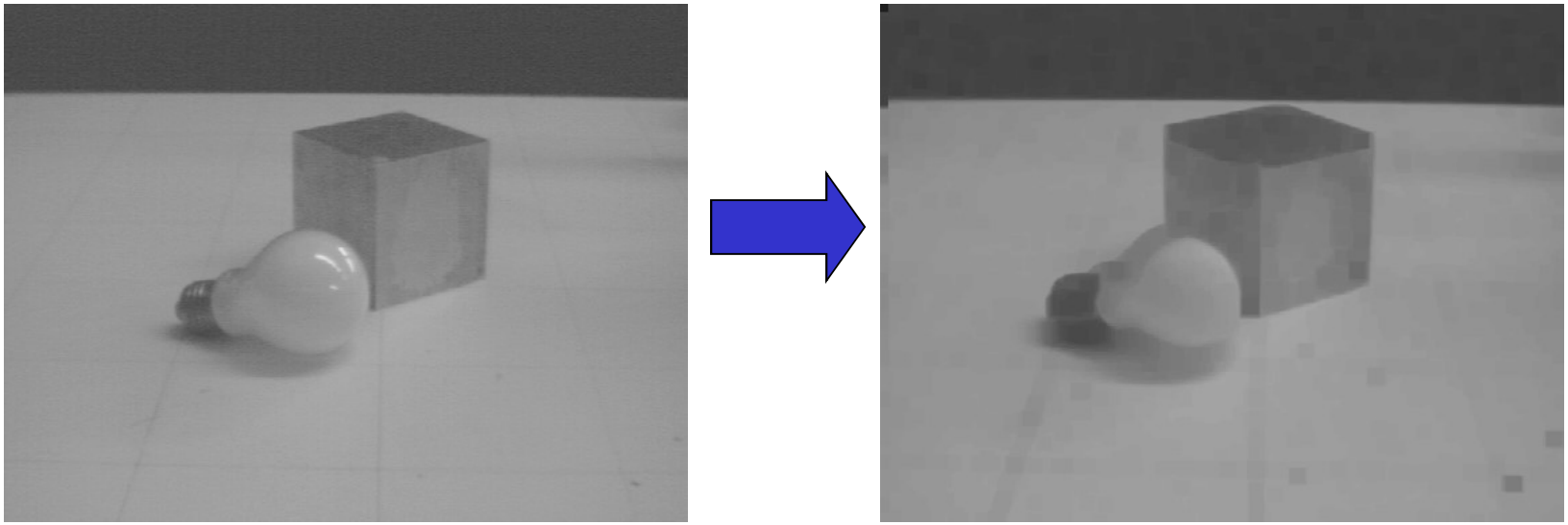
Introduction to Erosion on Gray Value Images

- View gray value images as a **stack of binary images!**



Intensity is lower so the image is darker

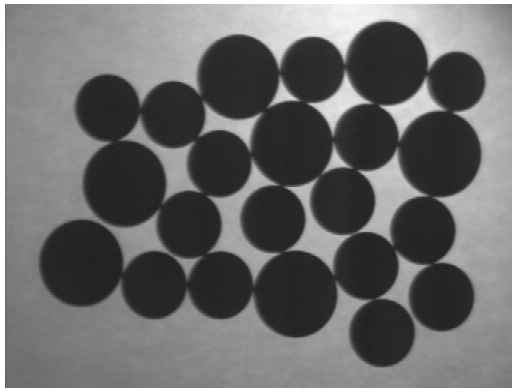
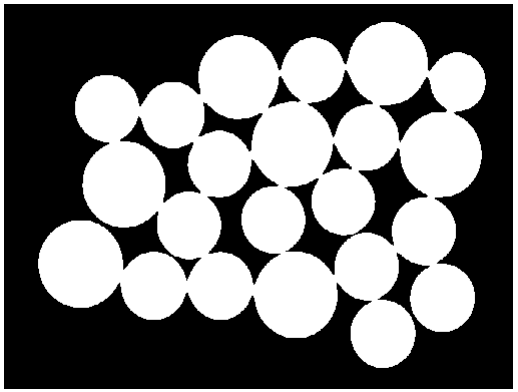
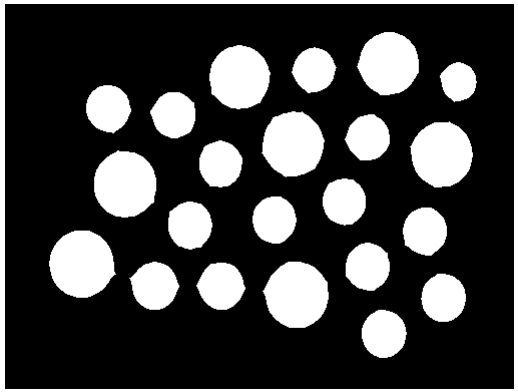
Erosion on Gray Value Images



- Images get darker!

Example: Counting Coins

- Counting coins is difficult because they touch each other!
- Solution: Binarization and Erosion separates them!

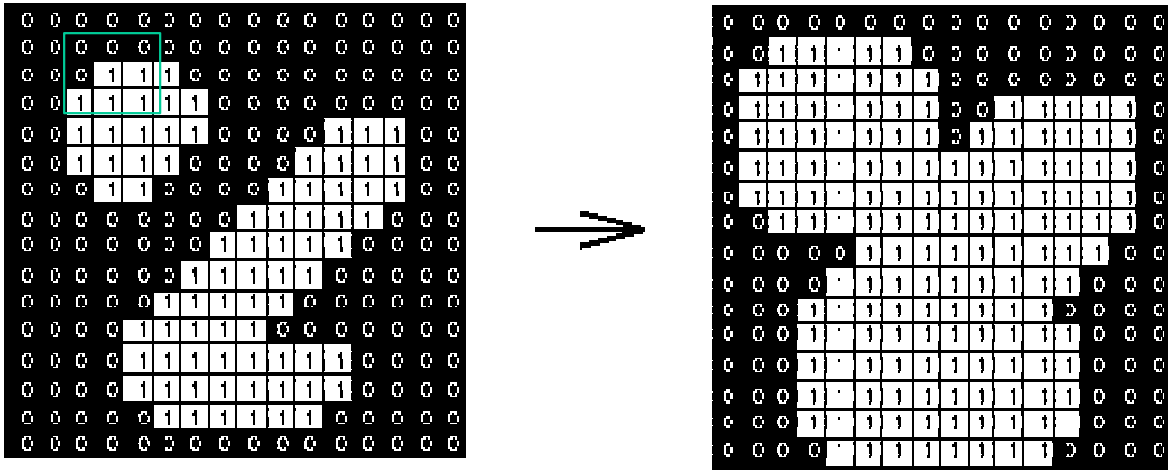




DILATION

more details

Example: Dilation

- **Dilation** is an important morphological operation



- Applied **Structuring Element**:

1	1	1
1	1	1
1	1	1

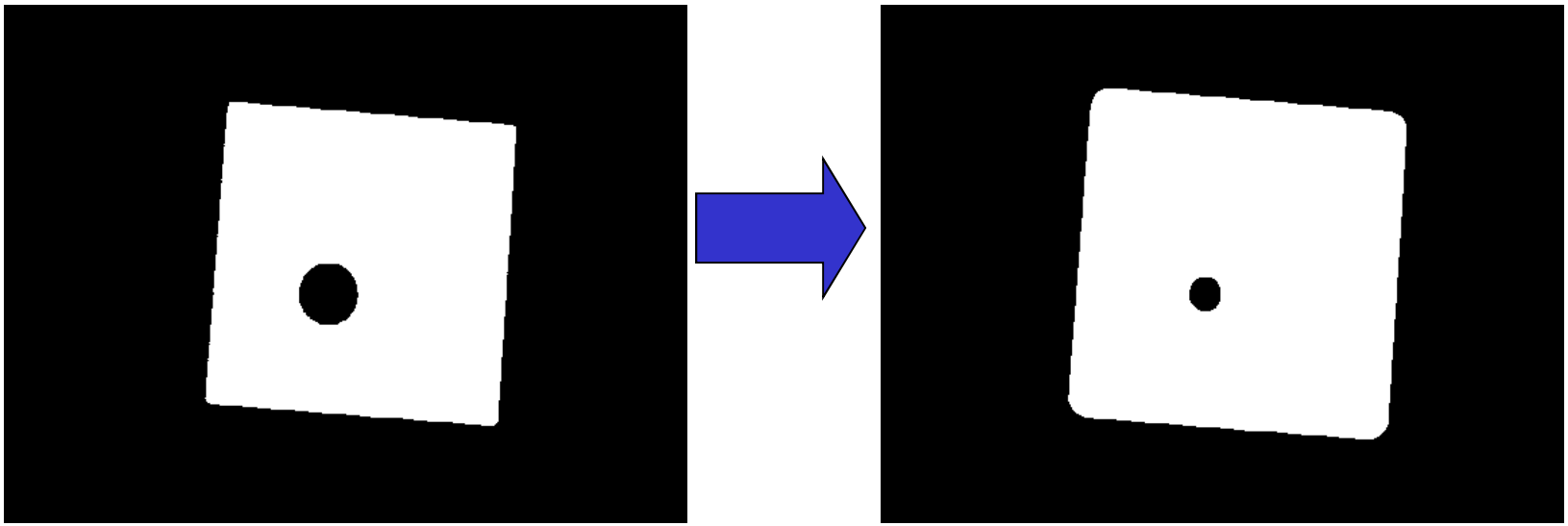
Set of coordinate points =

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

Dilation

- **Dilation** is the set of all points in the image, where the structuring element "touches" the foreground.
- Consider each pixel in the input image
  - If the structuring element touches the foreground image, write a "1" at the origin of the structuring element!
- **Input:**
  - **Binary Image**
  - **Structuring Element, containing only 1s!!**

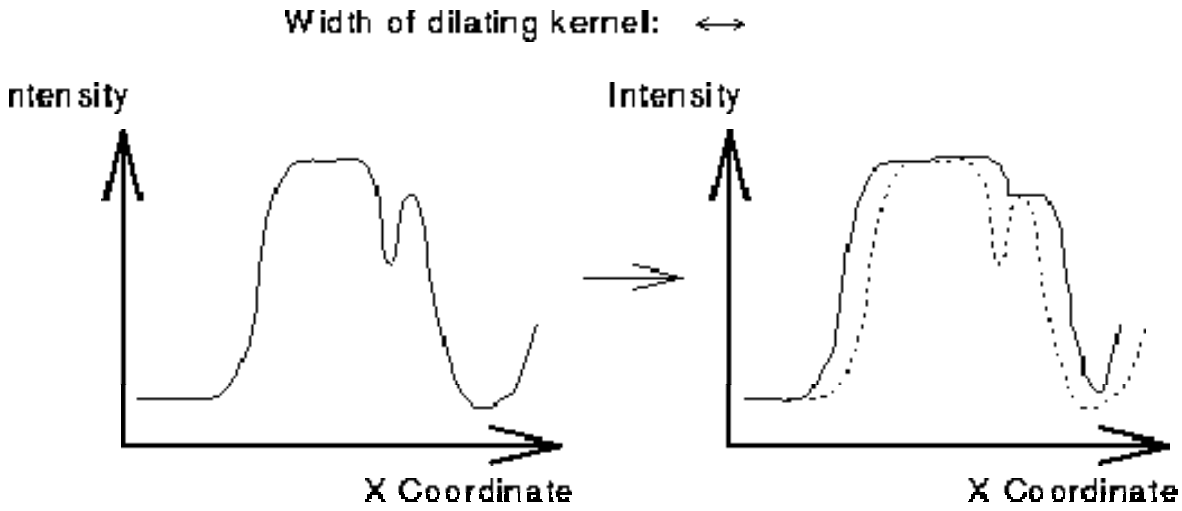
Another Dilation Example



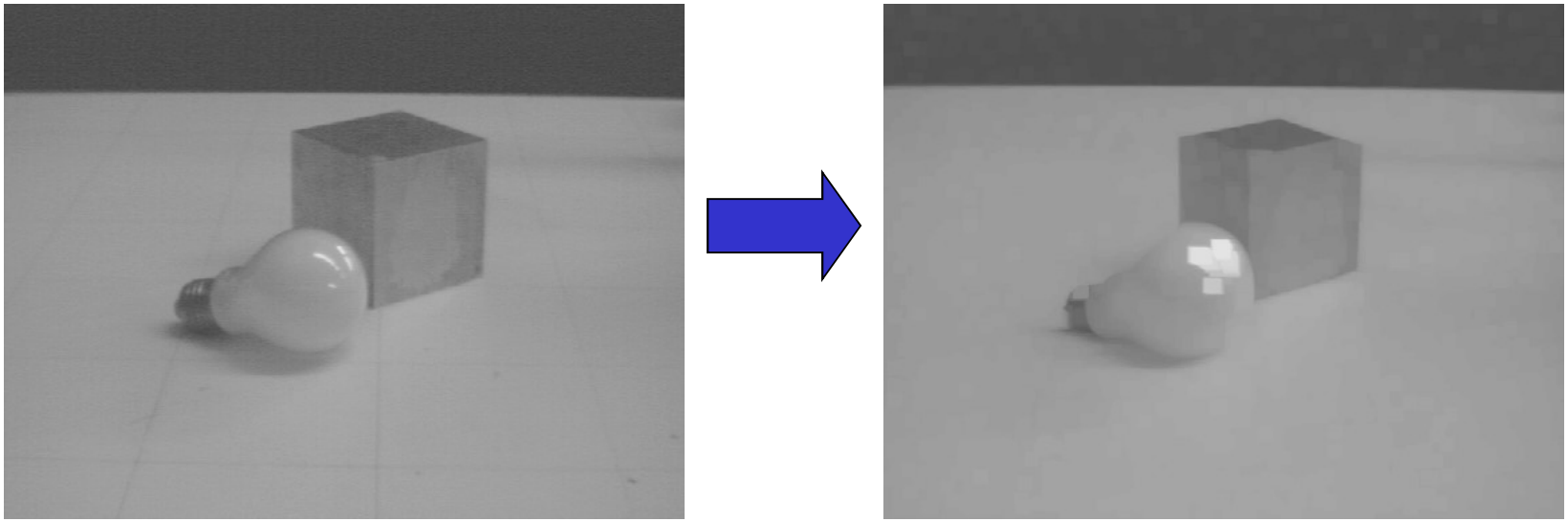
- Image get lighter, more uniform intensity

Dilation on Gray Value Images

- View gray value images as a stack of binary images!



Dilation on Gray Value Images

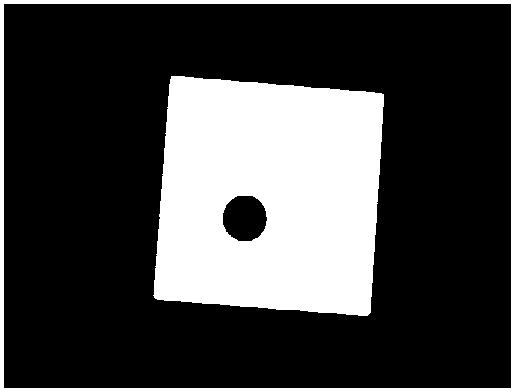
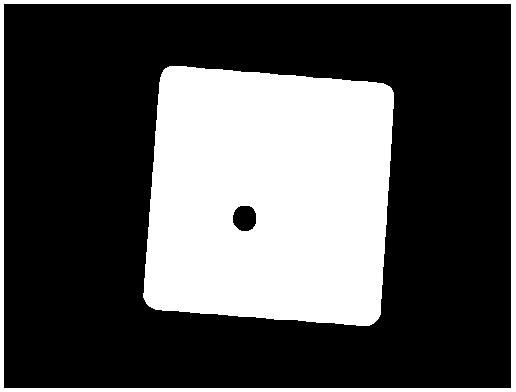
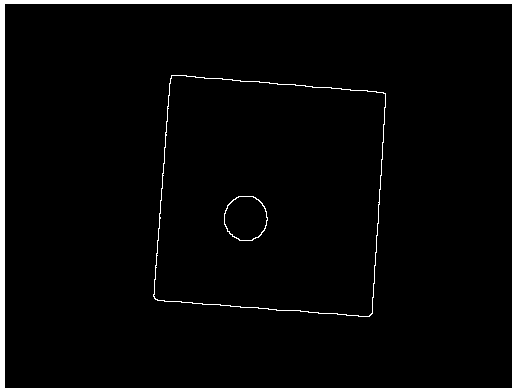


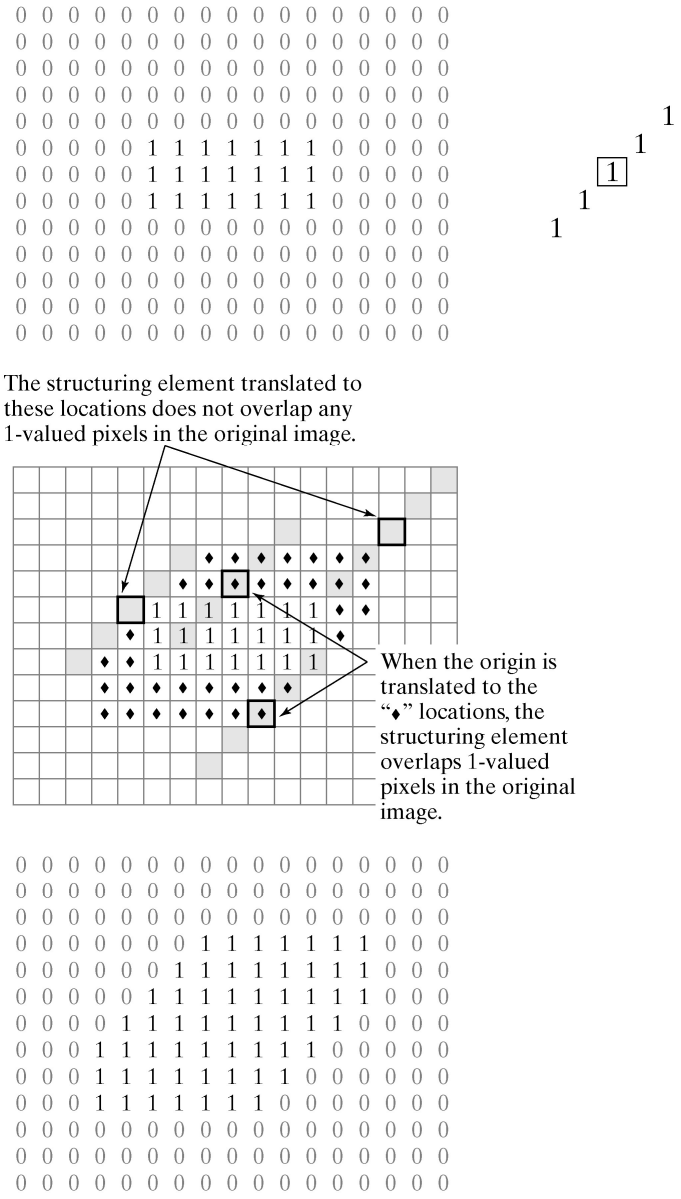
- More uniform intensity

Edge Detection

- Edge Detection

1. Dilate input image
2. Subtract input image from dilated image
3. Edges remain!



a b
c
d

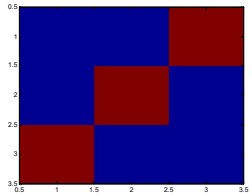
FIGURE 9.4
Illustration of dilation.
(a) Original image with rectangular object.
(b) Structuring element with five pixels arranged in a diagonal line. The origin of the structuring element is shown with a dark border.
(c) Structuring element translated to several locations on the image.
(d) Output image.

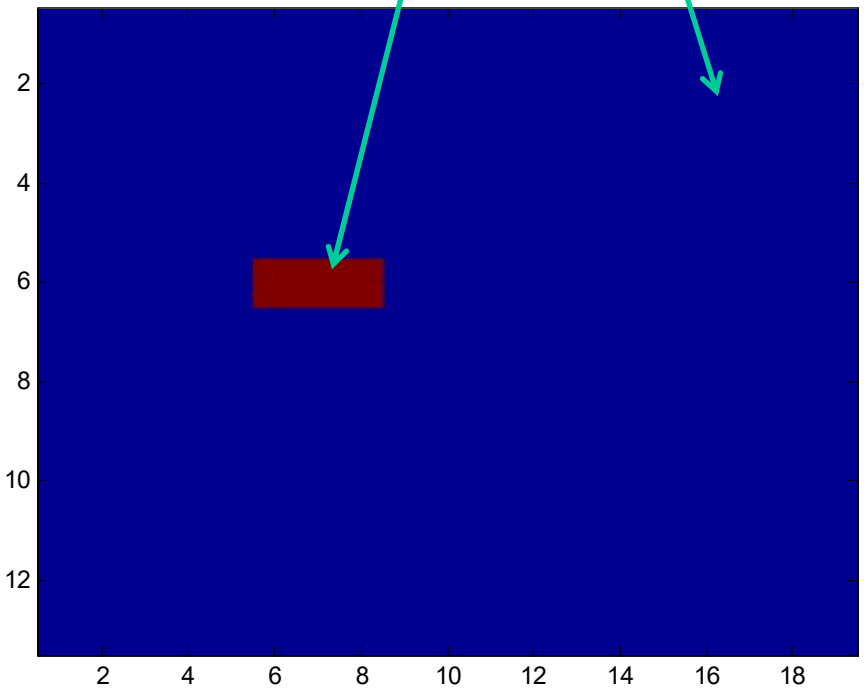
From: Digital Image Processing, Gonzalez,Woods And Eddins

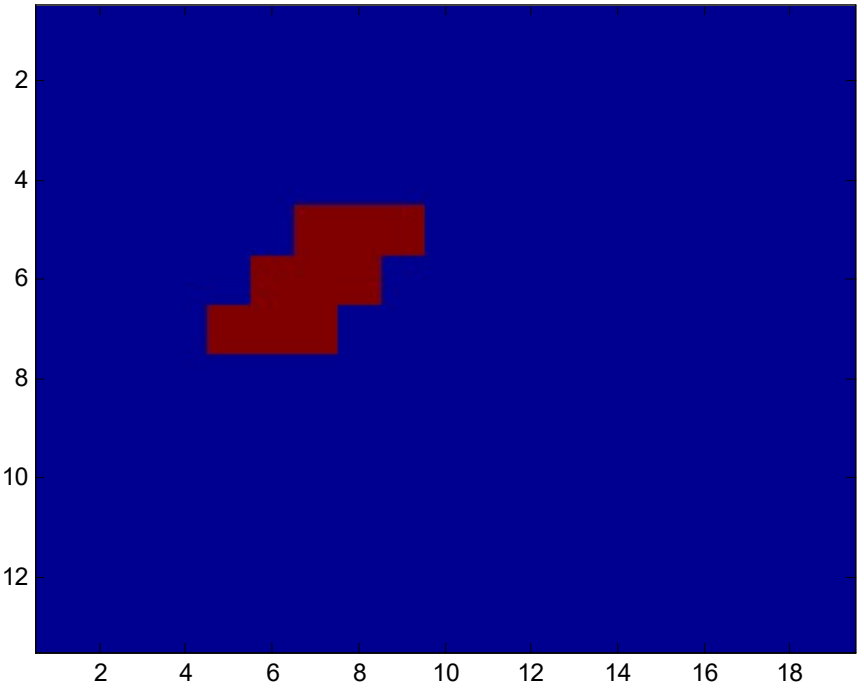
Illustration of dilation

Example of Dilation in Matlab

```
>> I = zeros([13 19]);
>> I(6,6:8)=1;
>> I2 = imdilate(I,se);
```







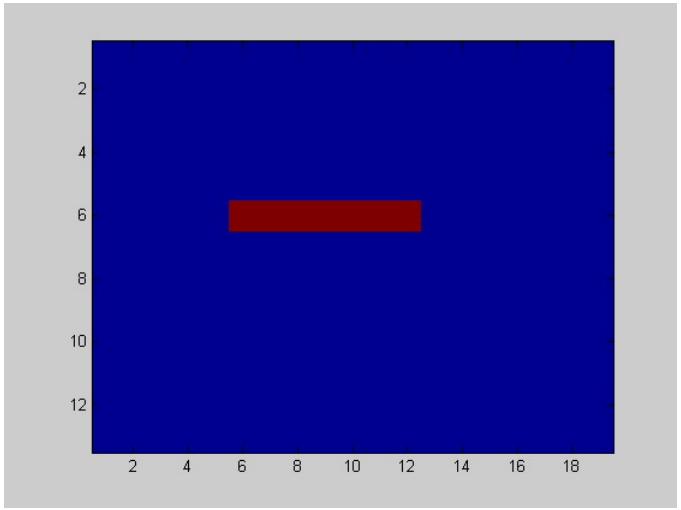
Imdilate function in MATLAB

IM2 = IMDILATE(IM,NHOOD) dilates the image IM, where NHOOD is a matrix of 0s and 1s that specifies the structuring element neighborhood. This is equivalent to the syntax IIMDILATE(IM, STREL(NHOOD)). IMDILATE determines the center element of the neighborhood by FLOOR((SIZE(NHOOD) + 1)/2).

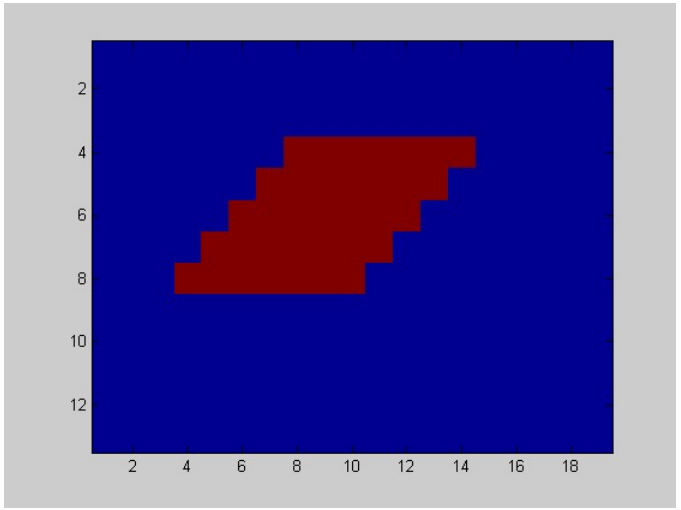
```
>> se = imrotate(eye(3),90)
se =

     0     0     1
     0     1     0
     1     0     0

>> ctr=floor(size(se)+1)/2
ctr =
     2     2
```

MATLAB Dilation Example

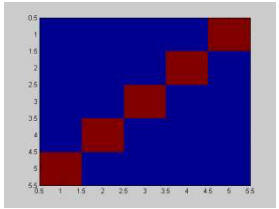
```
>> I = zeros([13 19]);
>> I(6, 6:12)=1;
>> SE = imrotate(eye(5),90);
>> I2=imdilate(I,SE);
>> figure, imagesc(I)
>> figure, imagesc(SE)
>> figure, imagesc(I2)
```



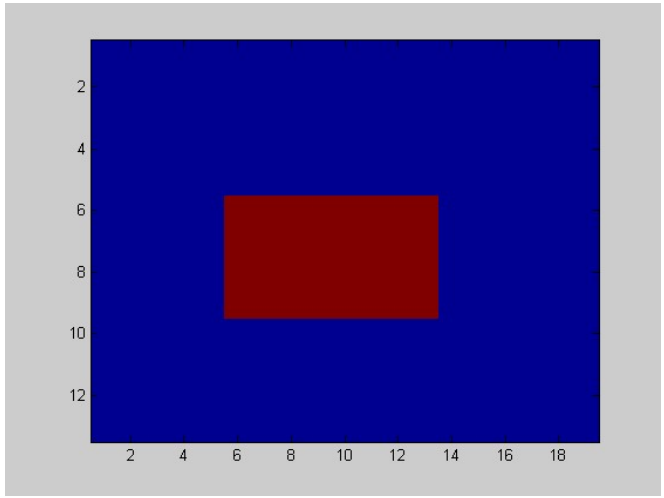
INPUT IMAGE

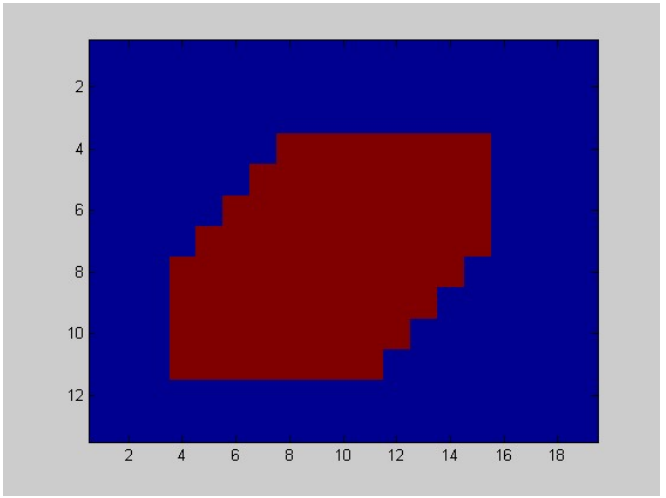


DILATED IMAGE



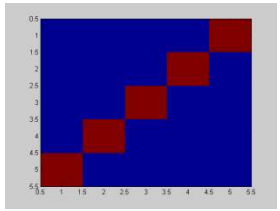
SE





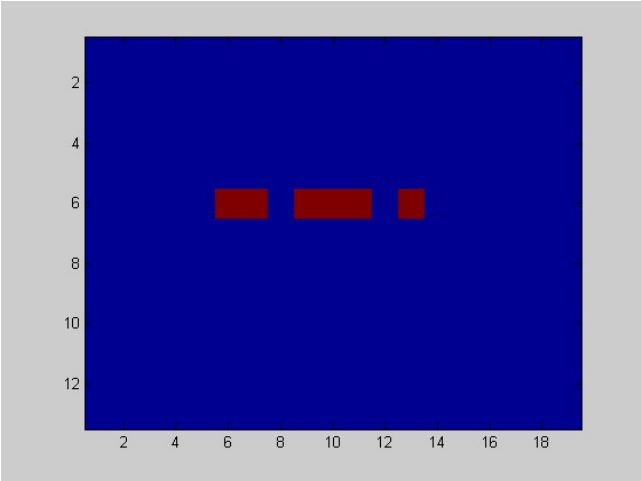
I

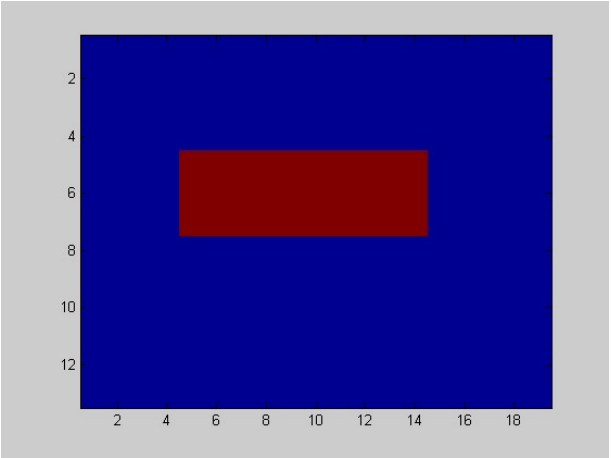
I2



SE

```
>> I(6:9,6:13)=1;
>> figure, imagesc(I)
>> I2=imdilate(I,SE);
>> figure, imagesc(I2)
```





SE = I

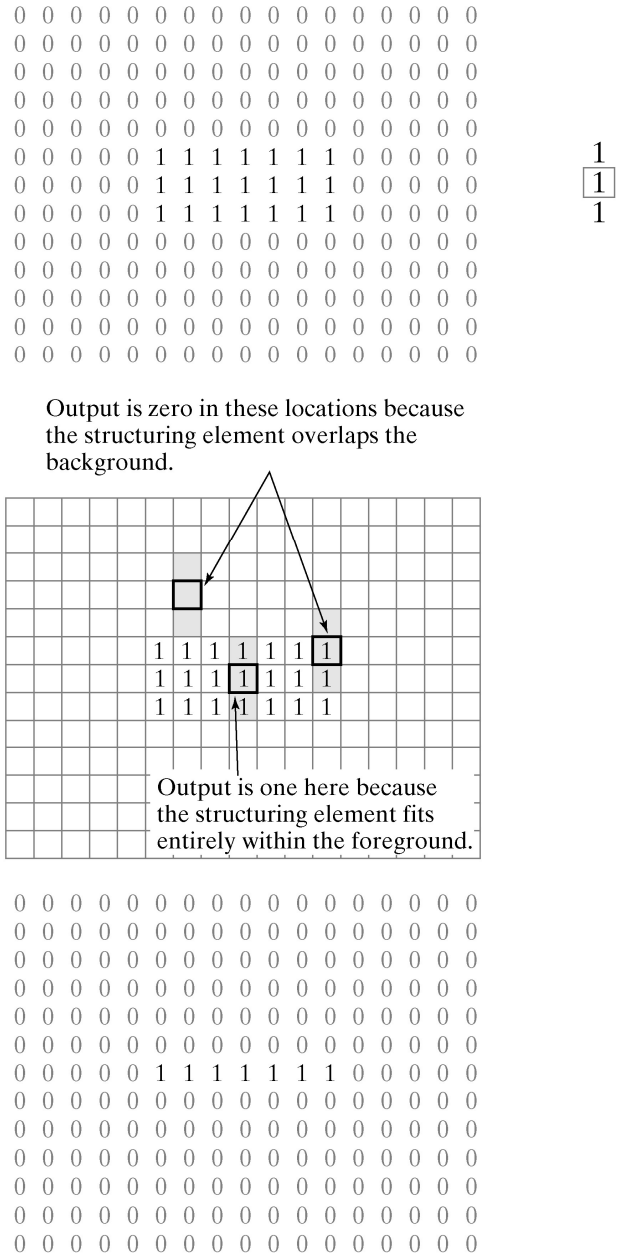
I2

1 1 1
1 1 1
1 1 1

Dilation and Erosion

Dilation and Erosion

- DILATION: Adds pixels to the boundary of an object
- EROSIN: Removes pixels from the boundary of an object
- Number of pixels added or removed depends on size and shape of structuring element



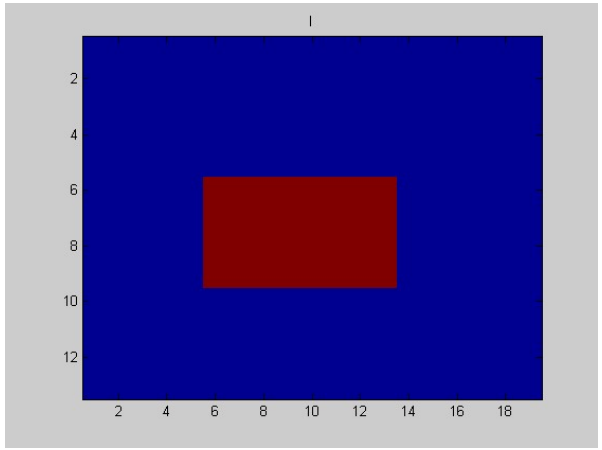
a b
c
d

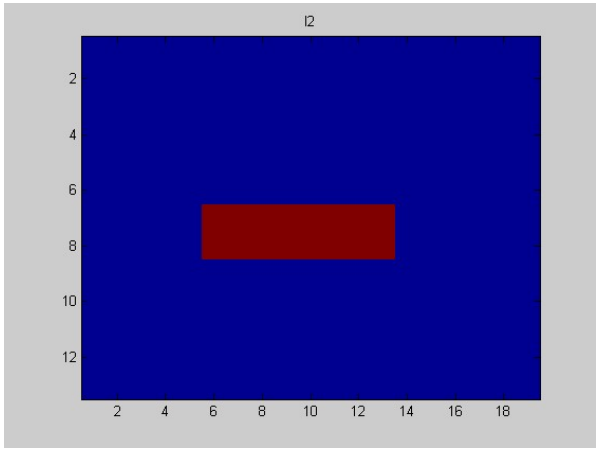
FIGURE 9.7
Illustration of erosion.
(a) Original image with rectangular object.
(b) Structuring element with three pixels arranged in a vertical line. The origin of the structuring element is shown with a dark border.
(c) Structuring element translated to several locations on the image.
(d) Output image.

Illustration of Erosion

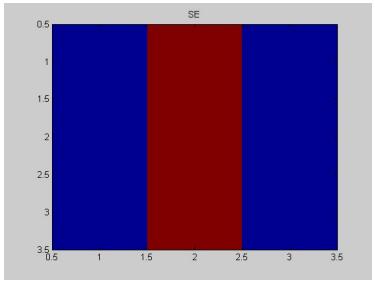
From: Digital Image Processing, Gonzalez,Woods And Eddins

MATLAB Erosion Example





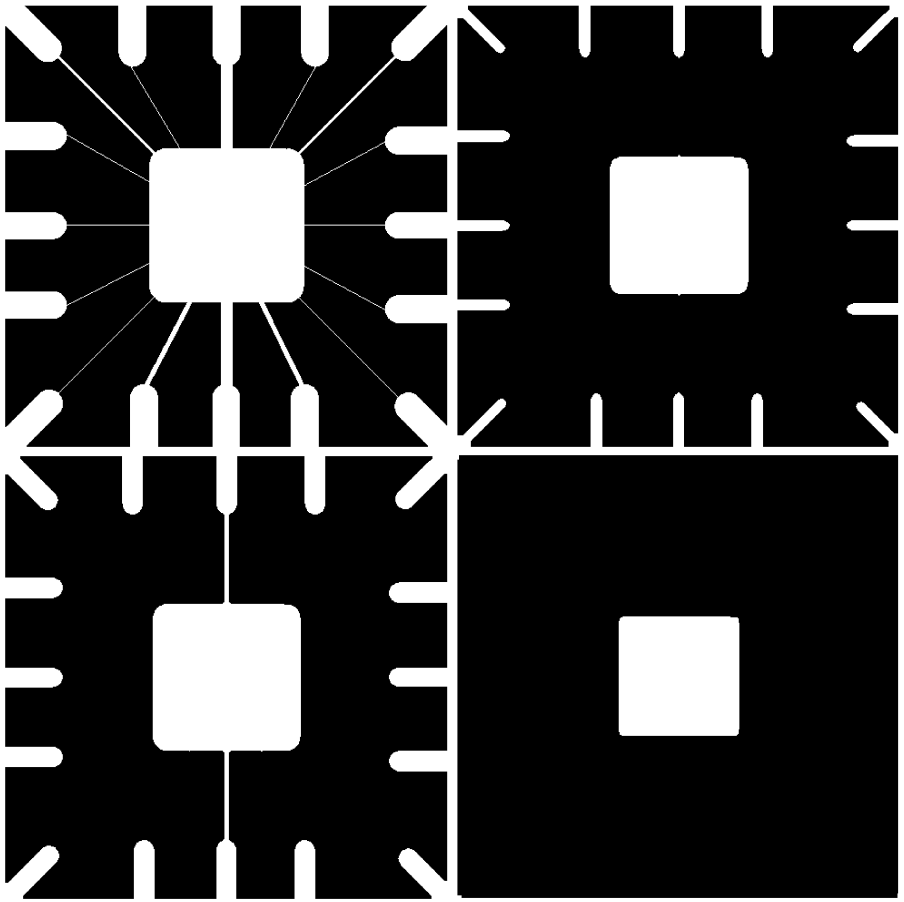
2 pixel wide



SE = 3x3

I3=imerode(I2,SE);

Illustration of erosion



a b
c d

FIGURE 9.8 An illustration of erosion. (a) Original image. (b) Erosion with a disk of radius 10. (c) Erosion with a disk of radius 5. (d) Erosion with a disk of radius 20.

From: Digital Image Processing, Gonzalez,Woods And Eddins

Combinations

- In most morphological applications dilation and erosion are used in combination
- May use same or different structuring elements

Opening & Closing

- Important operations
- Derived from the fundamental operations
  - Dilatation
  - Erosion
- Usually applied to binary images, but gray value images are also possible
- Opening and closing are dual operations

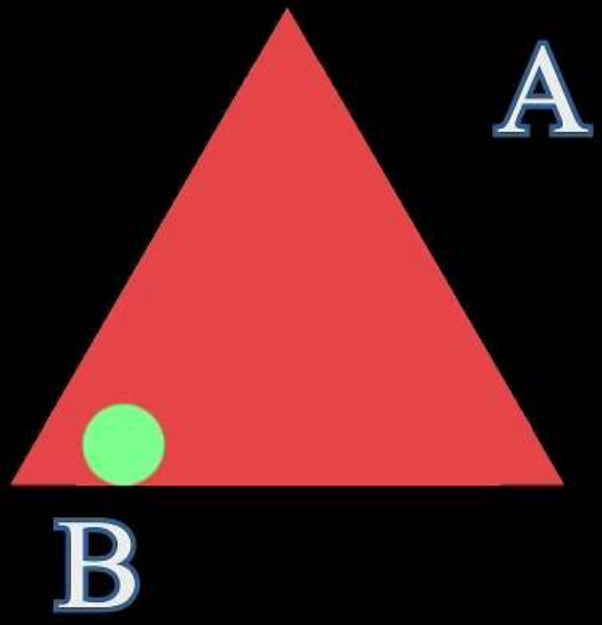
9.3 Opening And Closing

- Opening – smoothes contours , eliminates protrusions
- Closing – smoothes sections of contours, fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps in contours
- These operations are dual to each other
- These operations are can be applied few times, but has effect only once

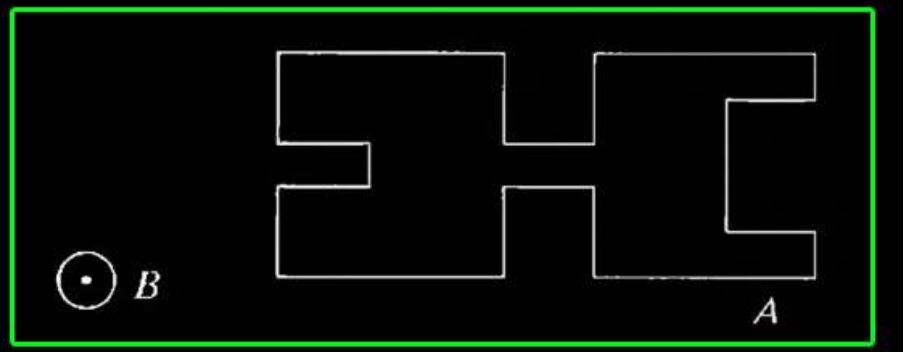
9.3 Opening And Closing

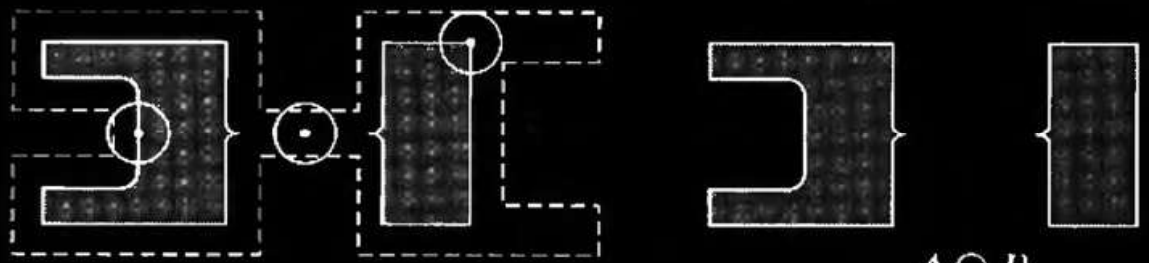
- Opening –
  - First – erode A by B, and then dilate the result by B
  - In other words, opening is the unification of all B objects Entirely Contained in A

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



Erosion





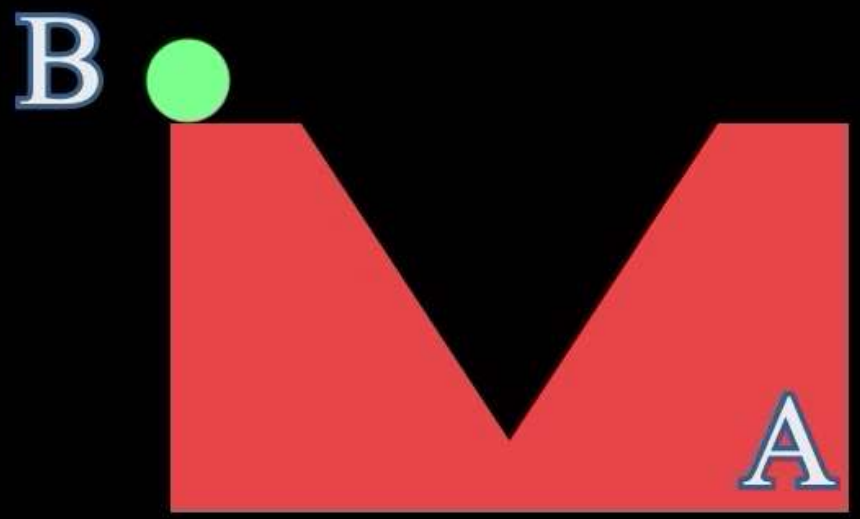
$A \ominus B$

Opening

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

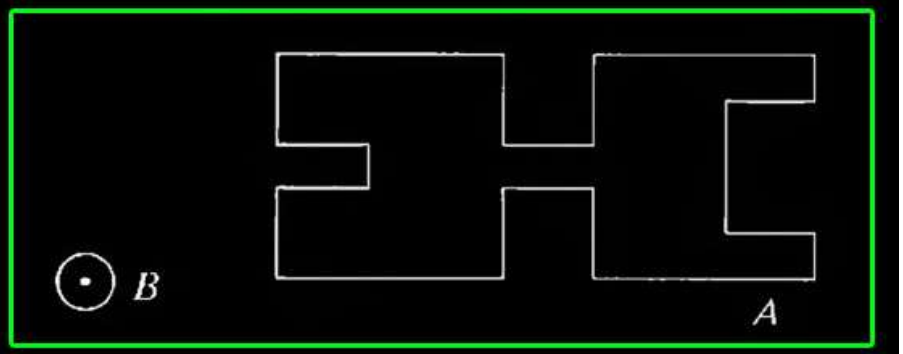
9.3 Opening And Closing

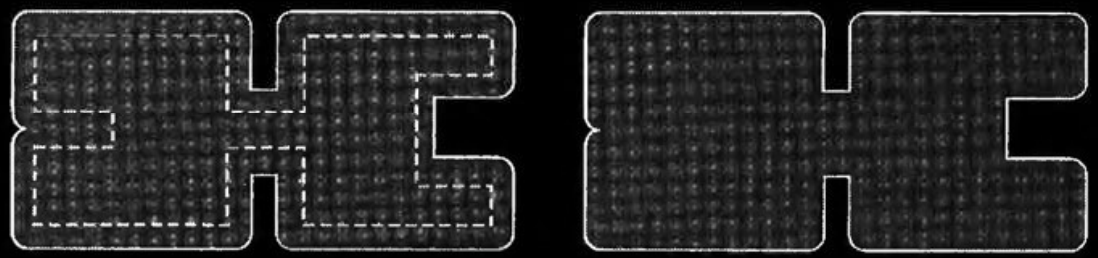
- Closing –
  - First – dilate A by B, and then erode the result by B
  - In other words, closing is the group of points, which the intersection of object B around them with object A – is not empty



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

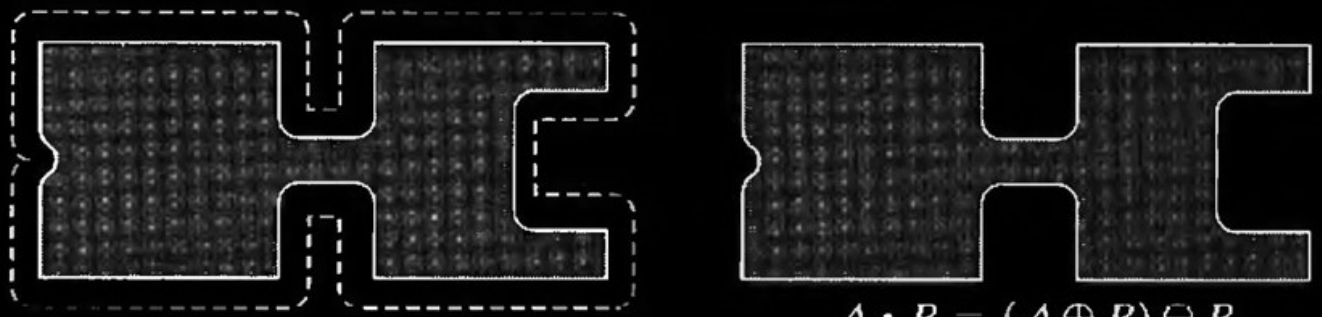
Dilation





$A \oplus B$

Closing



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

Use of opening and closing for morphological filtering

original image

erosion

opening of A

dilation of the opening

closing of the opening

1	1	1
1	1	1
1	1	1

B

OPENING

Opening

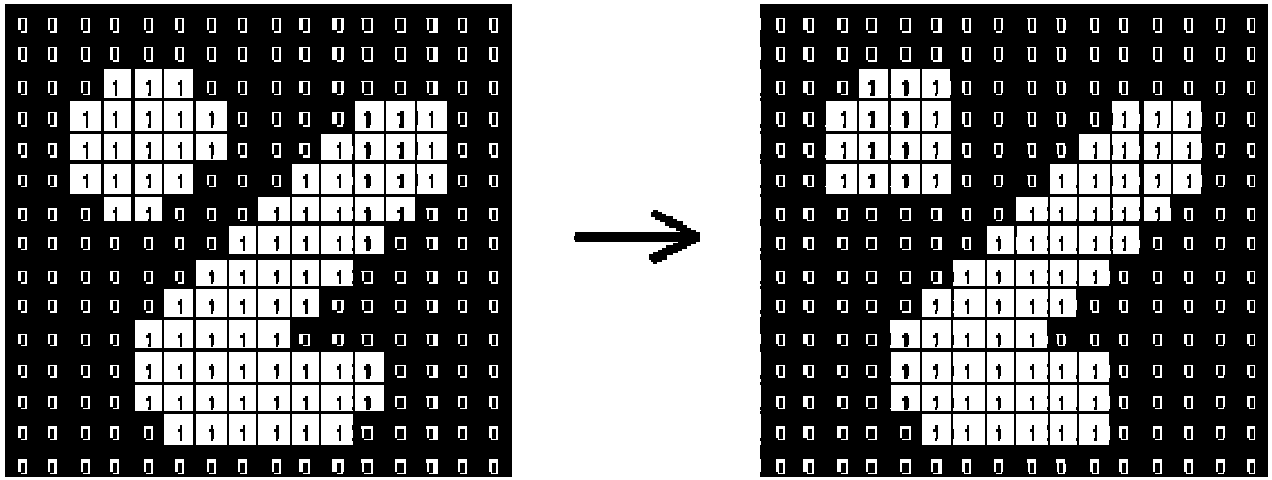
- Similar to Erosion
  - Spot and noise removal
  - Less destructive
- Erosion next dilation
- the s*ame structuring element for both operations.*
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!

Opening

- Take the structuring element (SE) and slide it around *inside* each foreground region.
  - All pixels which can be covered by the SE with the SE being entirely within the foreground region will be preserved.
  - All foreground pixels which can *not* be reached by the structuring element without lapping over the edge of the foreground object will be eroded away!
- Opening is **idempotent:** Repeated application has no further effects!

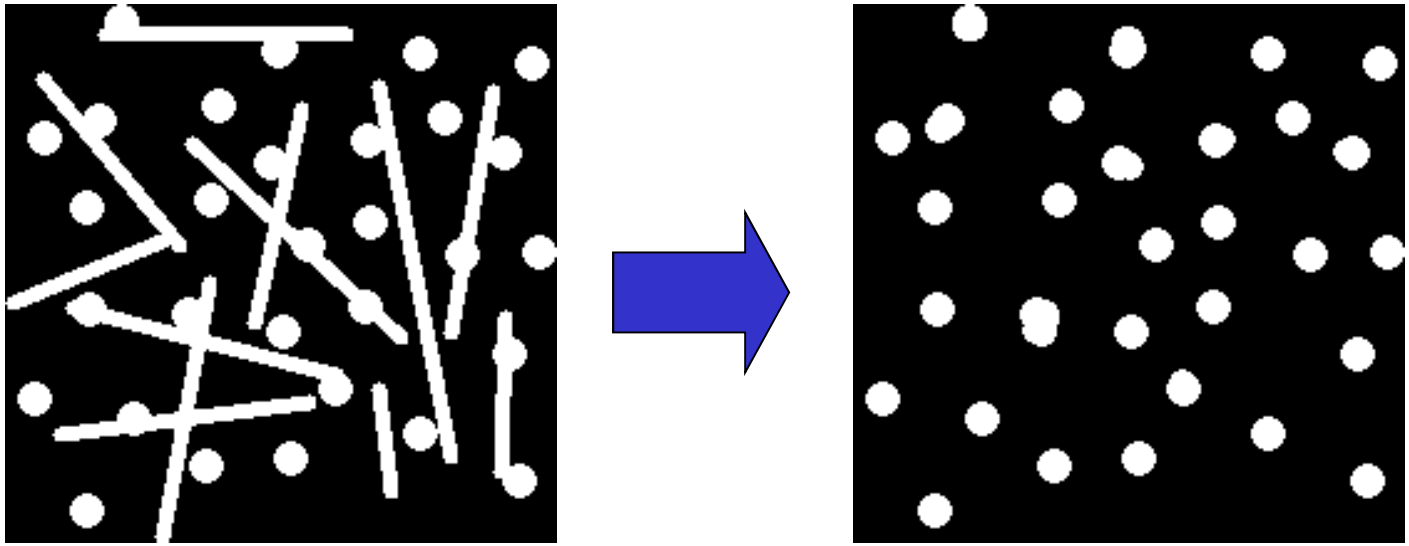
Opening

- Structuring element: 3x3 square



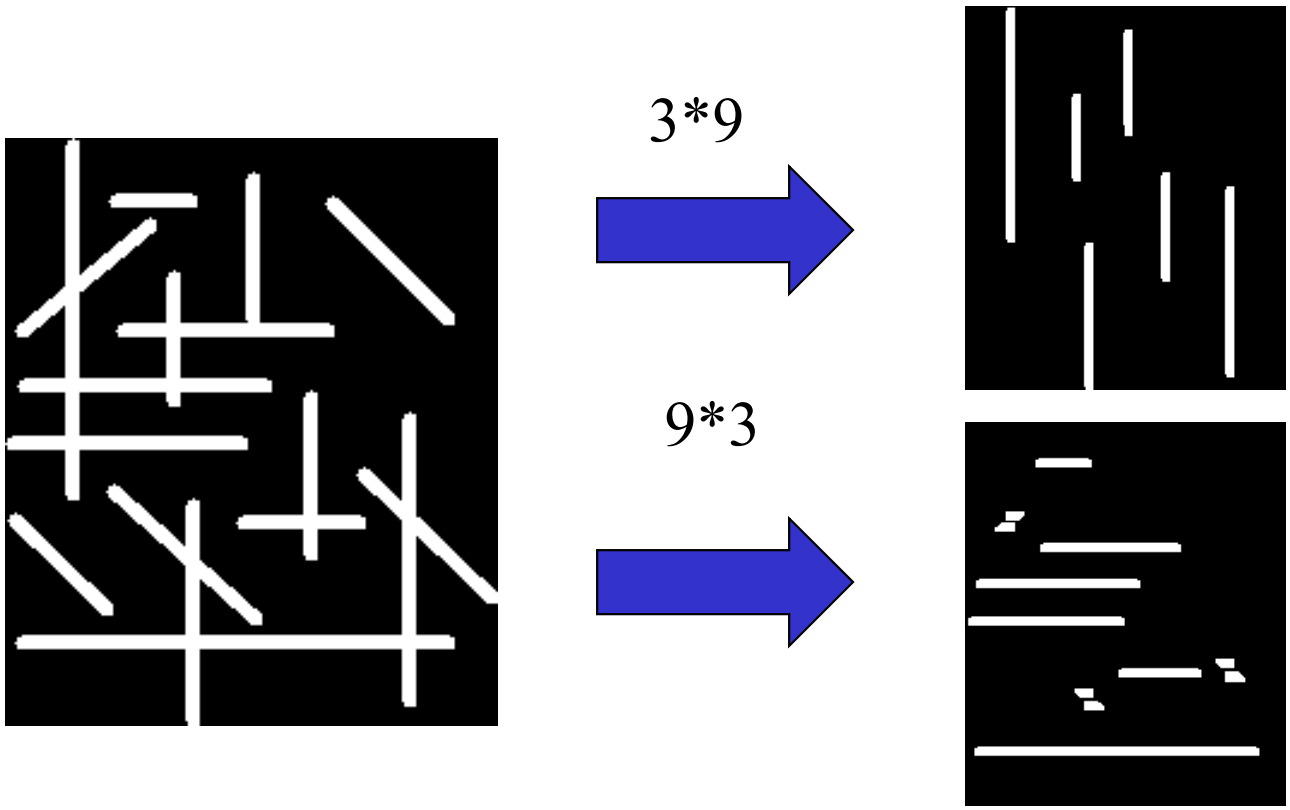
Opening Example

- Opening with a 11 pixel diameter disc



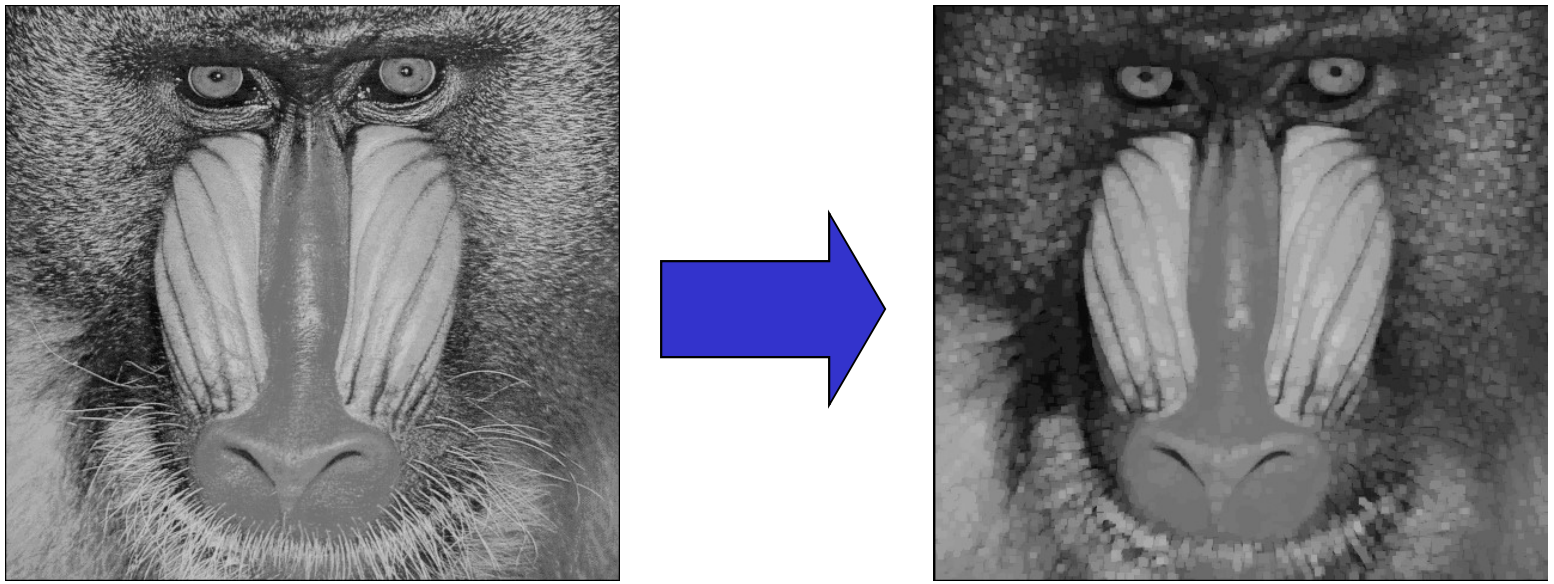
Opening Example

- 3x9 and 9x3 Structuring Element



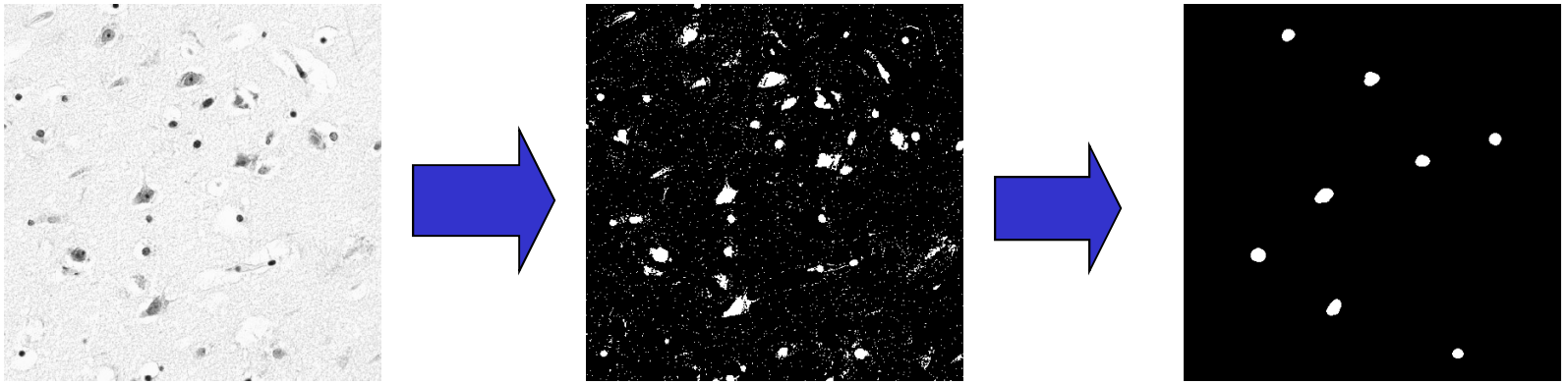
Opening on Gray Value Images

- 5x5 square structuring element



Use Opening for Separating Blobs

- Use large structuring element that fits into the big blobs
- Structuring Element: 11 pixel disc



CLOSING

Closing

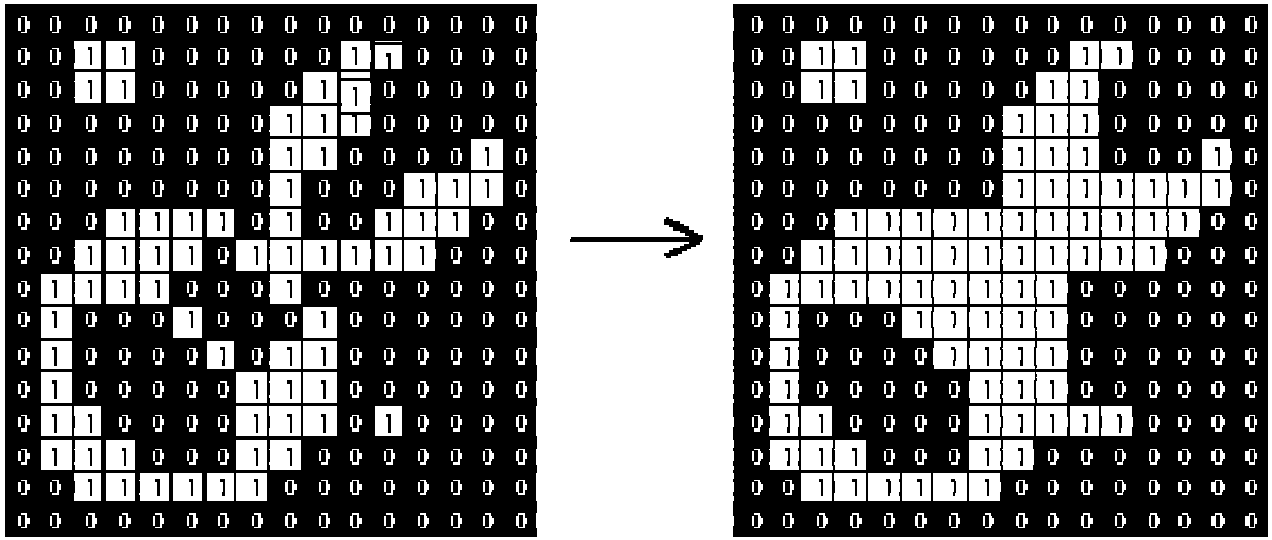
- Similar to Dilation
  - Removal of holes
  - Tends to enlarge regions, shrink background
- Closing is defined as a Dilatation, followed by an Erosion *using the same structuring element for both operations.*
- Dilation next erosion!
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!

Closing

- Take the structuring element (SE) and slide it around *outside* each foreground region.
  - All background pixels which can be covered by the SE with the SE being entirely within the background region will be preserved.
  - All background pixels which can *not* be reached by the structuring element without lapping over the edge of the foreground object will be turned into a foreground.
- Opening is **idempotent:** Repeated application has no further effects!

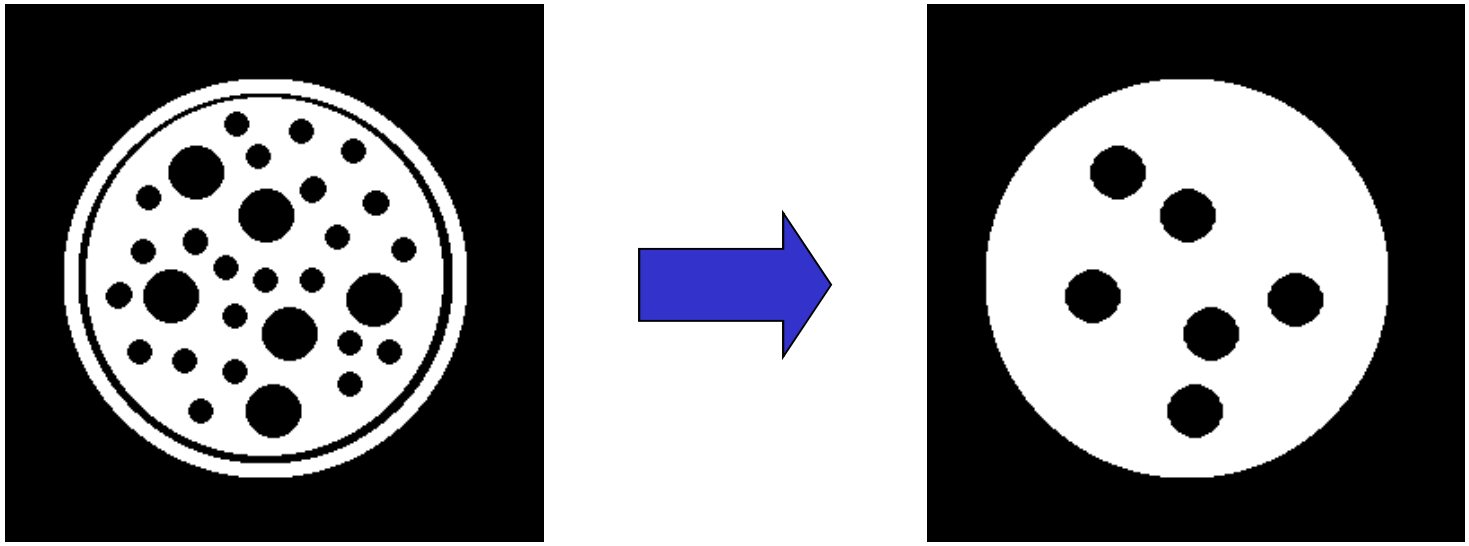
Closing

- Structuring element: 3x3 square



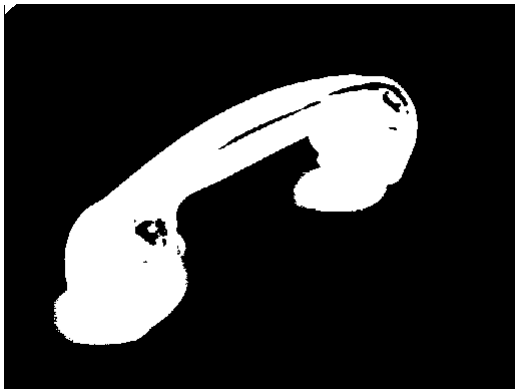
Closing Example

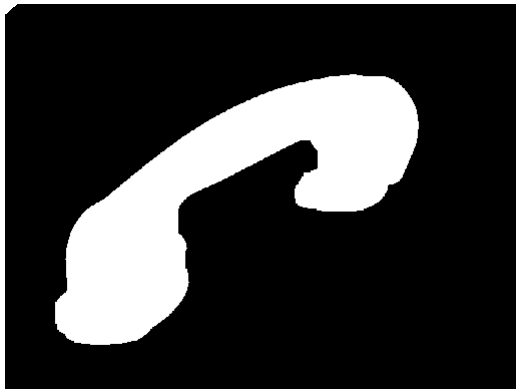
- Closing operation with a 22 pixel disc
- Closes small holes in the foreground



Closing Example 1

1. Threshold
2. Closing with disc of size 20

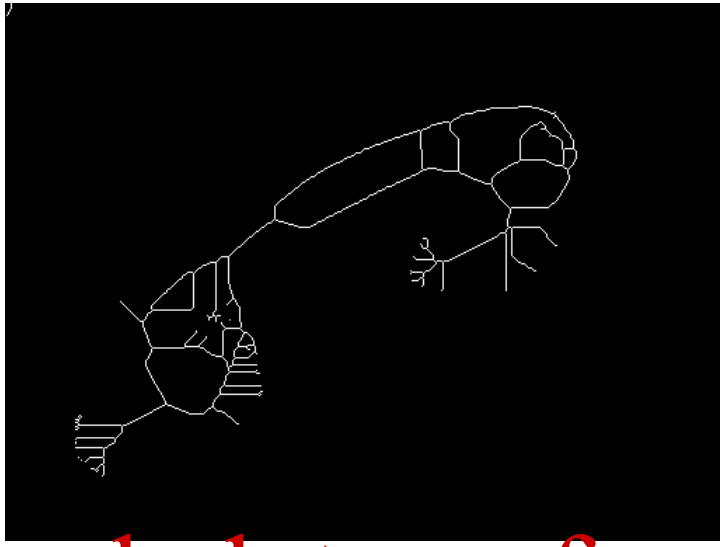




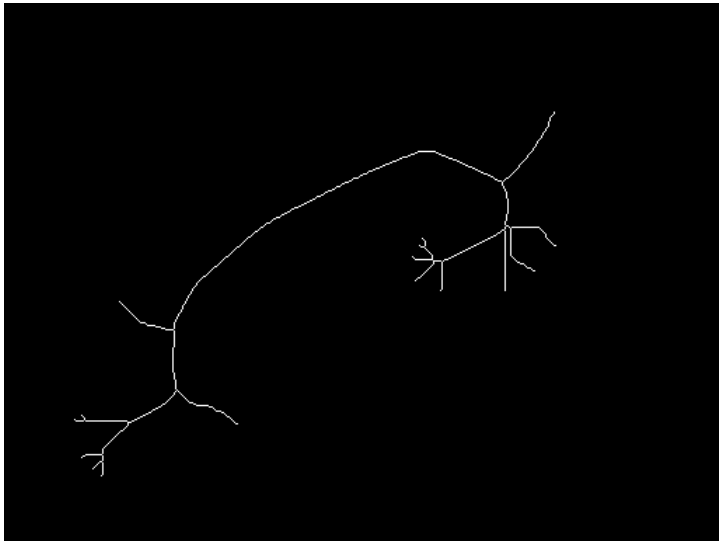
Thresholded closed

Closing Example 2

- Good for further processing: E.g. Skeleton operation looks better for closed image!



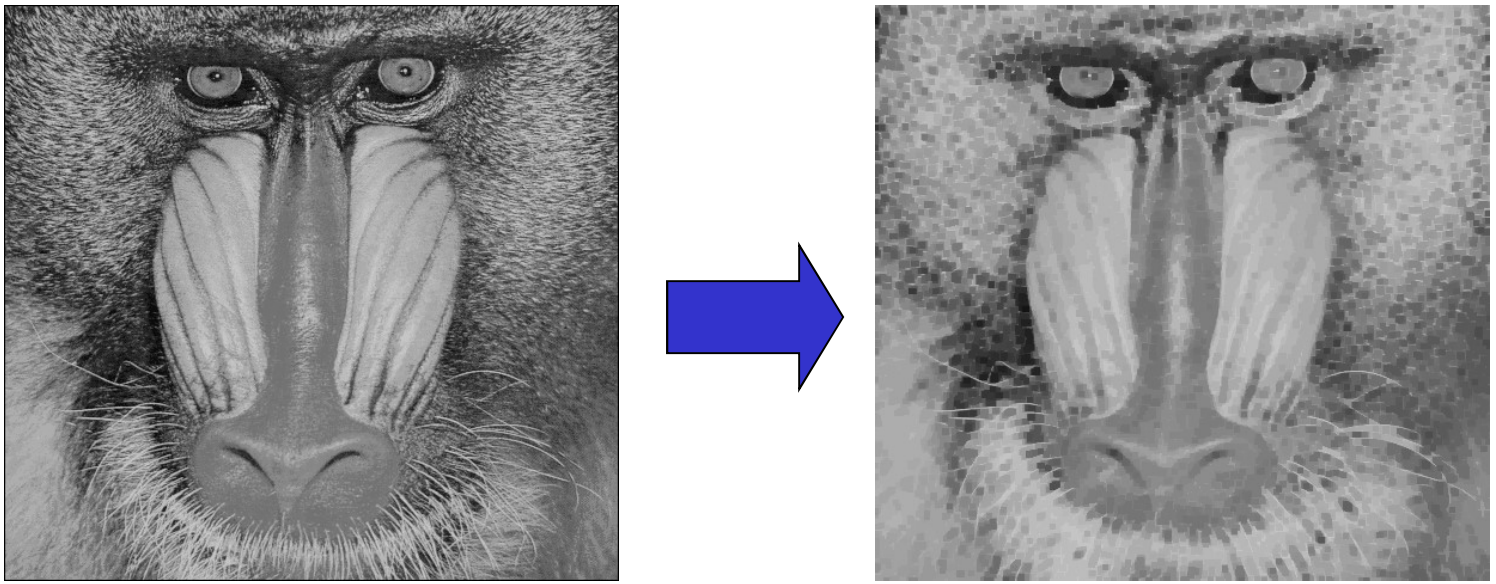
skeleton of Thresholded



skeleton of Thresholded and next closed

Closing Gray Value Images

- 5x5 square structuring element



Opening & Closing

- Opening is the *dual* of closing
- *i.e.* opening the foreground pixels with a particular structuring element
- is equivalent to closing the background pixels with the same element.

Morphological Opening and Closing

- Opening of A by B → $A \circ B$

  Erosion of A by B, followed by

  the dilation of the result by B

  Closing of A by B → $A \bullet B$

  Dilation of A by B, followed by

  the erosion of the result by B

  MATLAB: imopen(A, B)

  imclose(A,B)

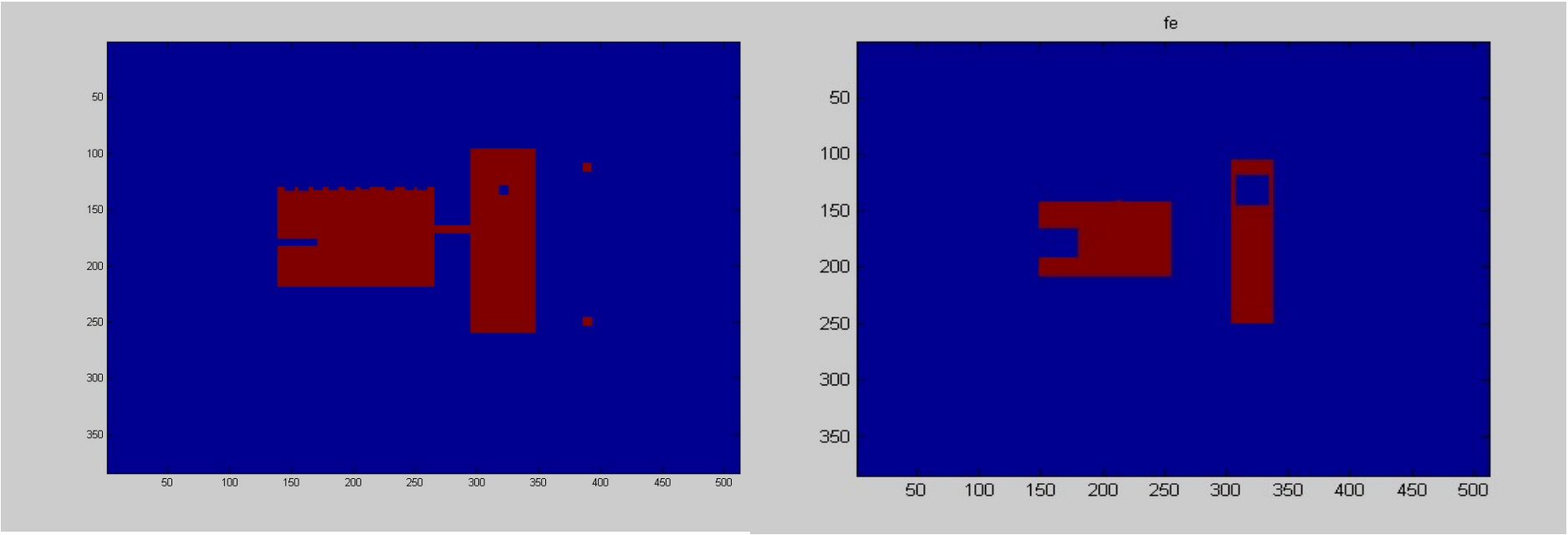
MATLAB Function strel

- strel constructs structuring elements with various shapes and sizes
- Syntax: se = strel(shape, parameters)
- Example:
  - se = strel('octagon', R);
  - R is the dimension – see help function

- # Opening of A by B → A∘ B

Erosion of A by B, followed by the dilation of the result by B

Erosion- if any element of structuring element overlaps with background output is zero

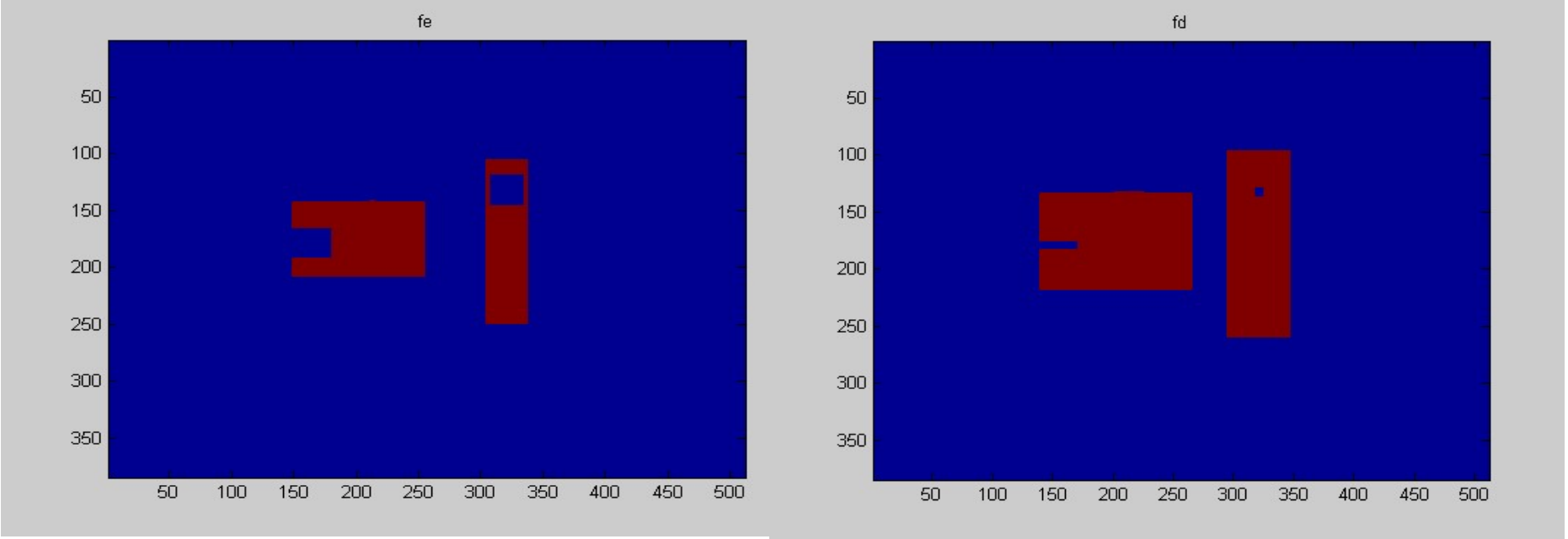


FIRST - EROSION

>> se = strel('square', 20);fe = imerode(f,se);figure, imagesc(fe),title('fe')

Dilation of Previous Result

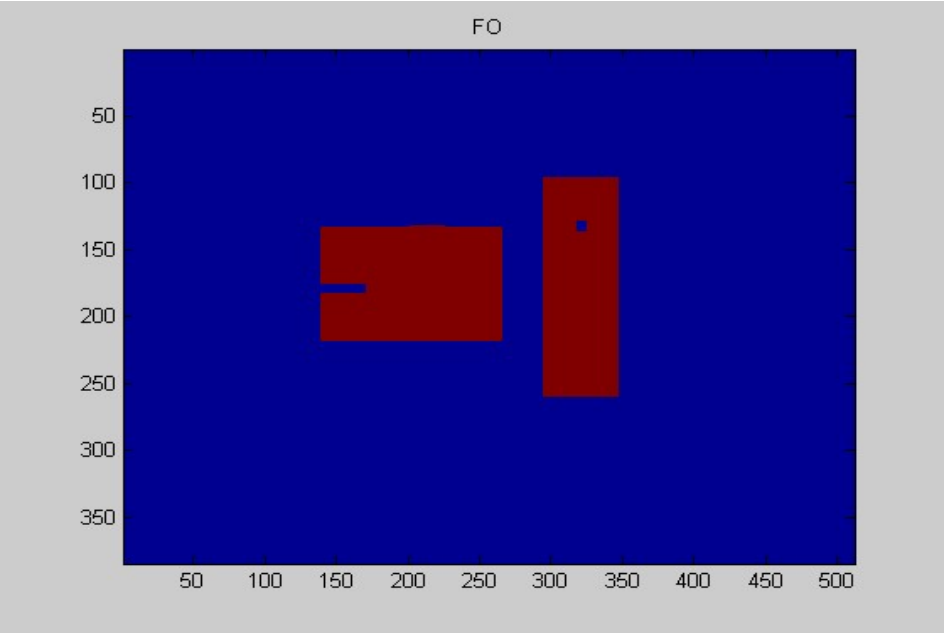
Outputs 1 at center of SE when at least one element of SE overlaps object



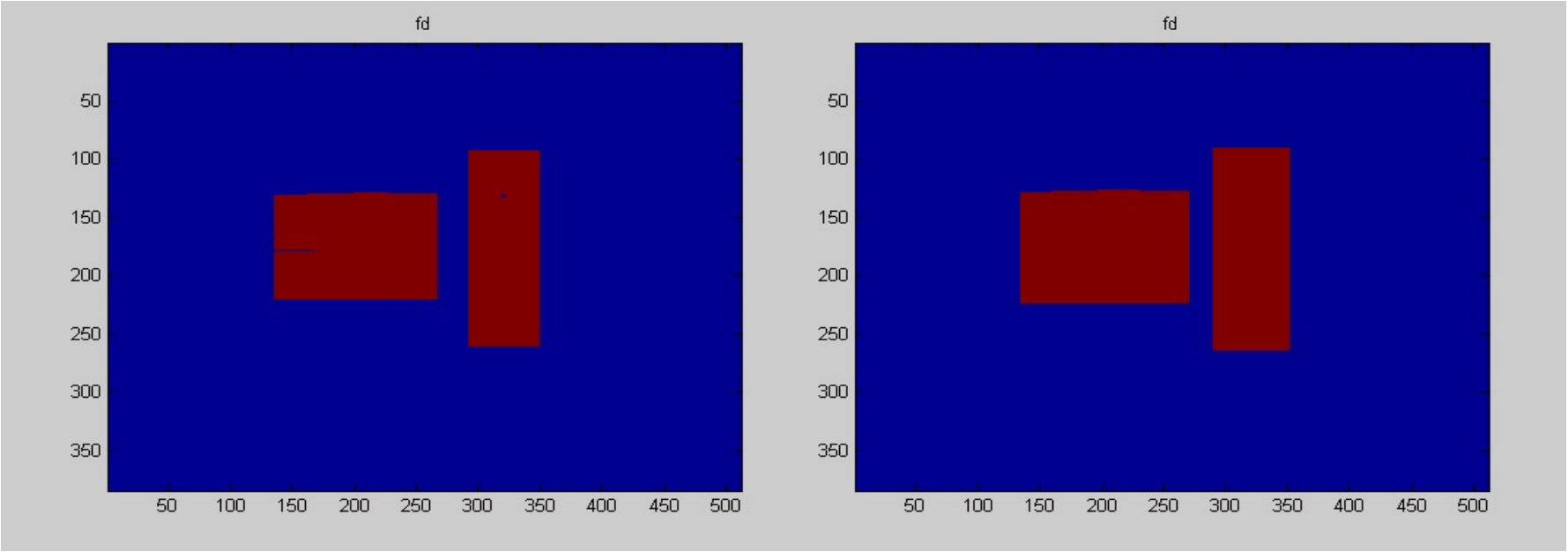
SECOND - DILATION

>> se = strel('square', 20);fd = imdilate(fe,se);figure, imagesc(fd),title('fd')

FO=imopen(f,se); figure, imagesc(FO),title('FO')



What if we increased size of SE for DILATION operation??



se = 25

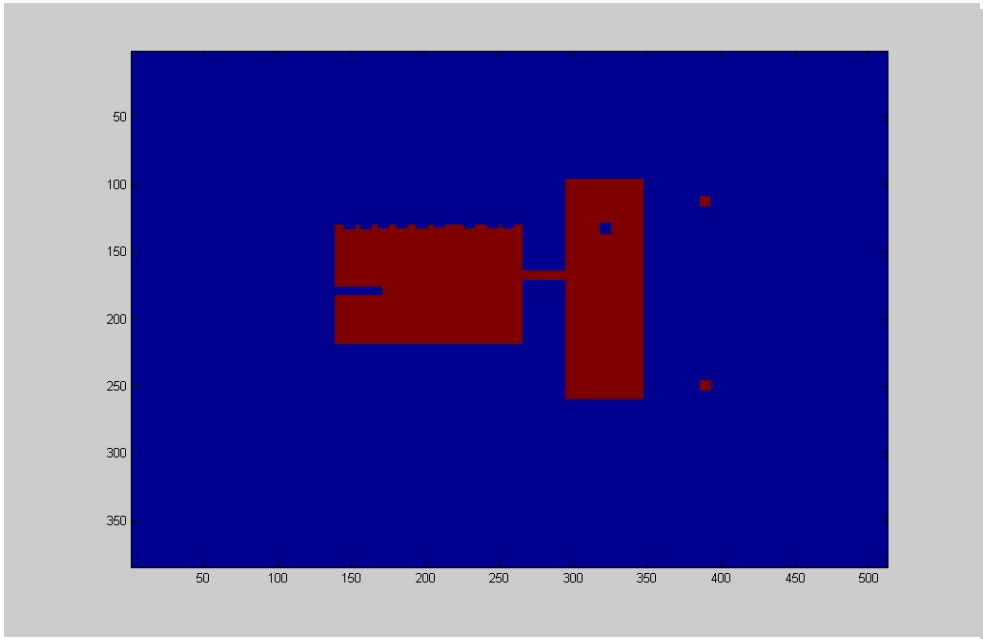
se = 30

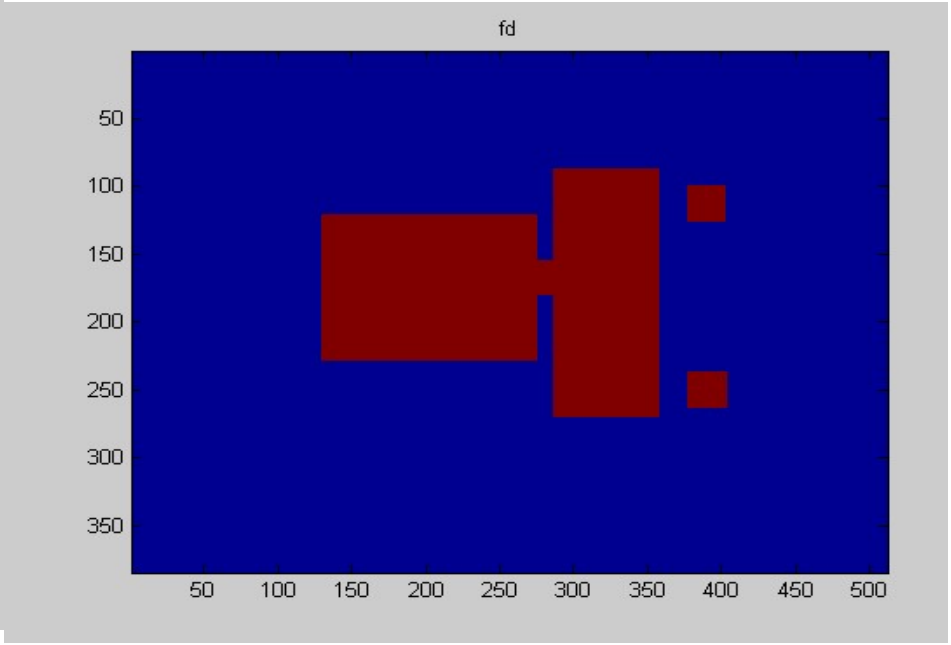
```
se = strel('square', 25);fd = imdilate(fe,se);figure, imagesc(fd),title('fd')
se = strel('square', 30);fd = imdilate(fe,se);figure, imagesc(fd),title('fd')
```

Closing of A by B → A•B

Dilation of A by B

Outputs 1 at center of SE when at least one element of SE overlaps object

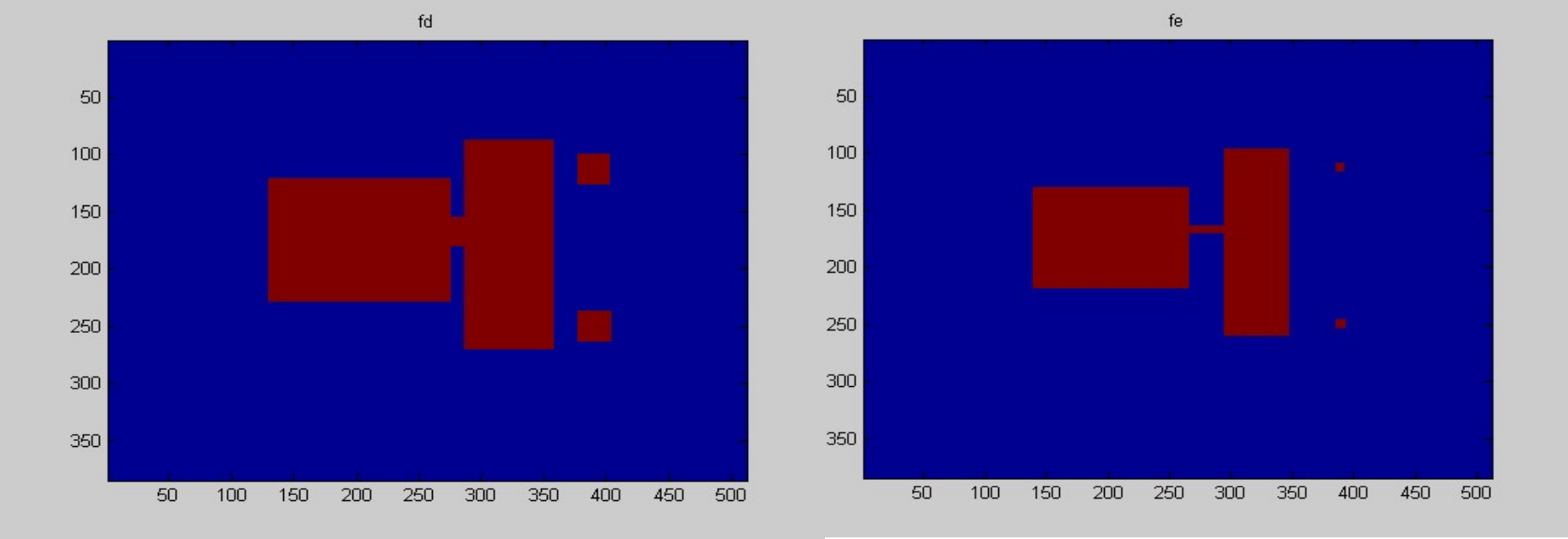


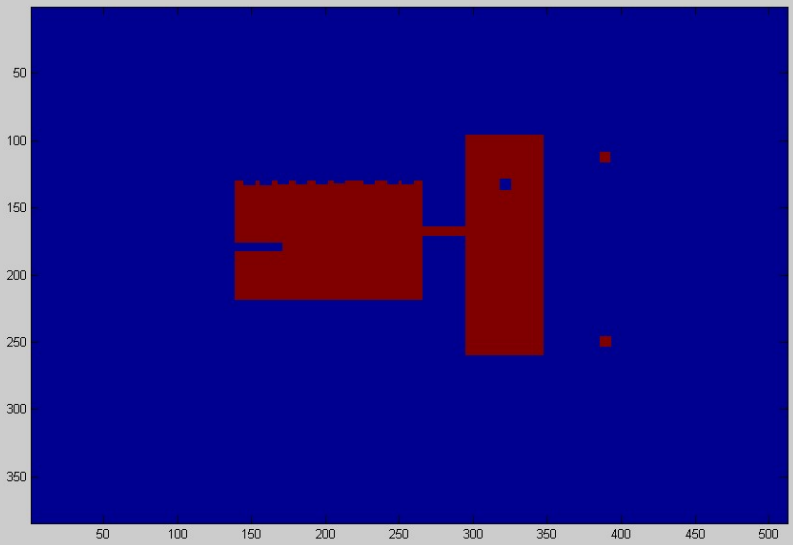


se = strel('square', 20);fd = imdilate(f,se);figure, imagesc(fd),title('fd')

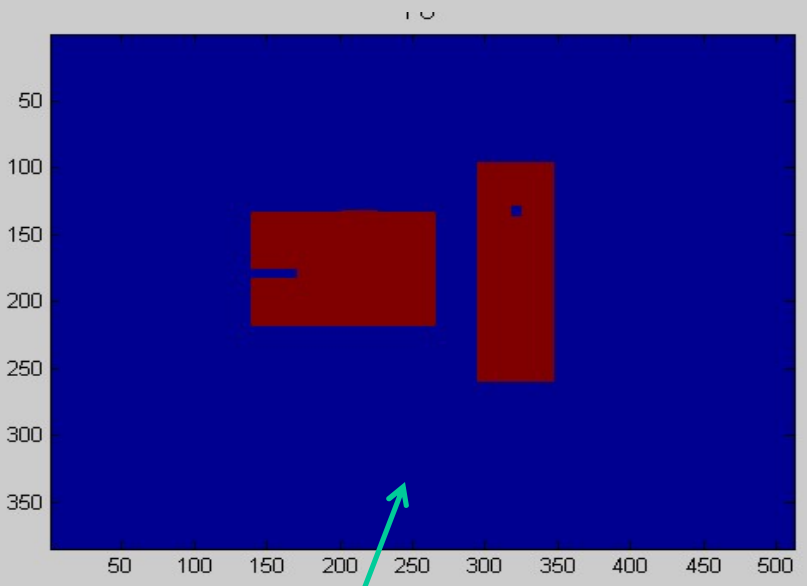
Erosion of the result by B

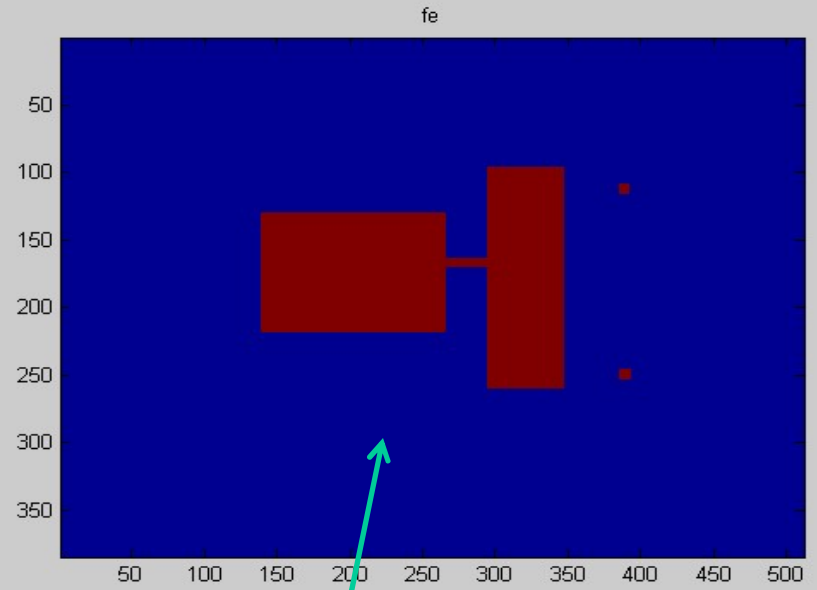
Erosion- if any element of structuring element overlaps with background output is zero





ORIGINAL





OPENING

CLOSING

Fingerprint problem

a b c

FIGURE 9.11 (a) Noisy fingerprint image. (b) Opening of image. (c) Opening followed by closing. (Original image courtesy of the National Institute of Standards and Technology.)

From: Digital Image Processing, Gonzalez,Woods And Eddins

HIT and MISS

Hit or Miss Transformation

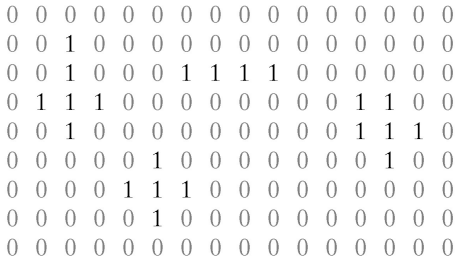
- "Hit or Miss" Called also "Hit and Miss" is useful to identify specified configuration of pixels.
- For instance, such combinations as:
  - isolated foreground pixels
  - or pixels at end of lines (end points)

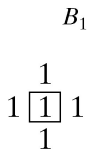
- $A \otimes B = (A \circ B1) \cap (A^c \circ B2)$
  - A eroded by B1, intersection A complement eroded by B2 (two different structuring elements)

Hit or Miss Example: Find cross shape pixel configuration

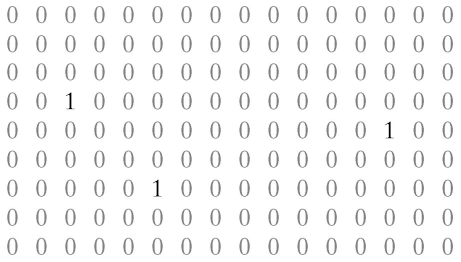
0	1	0
1	1	1
0	1	0

MATLAB Function: C = **bwhitmiss**(A, B1, B2)

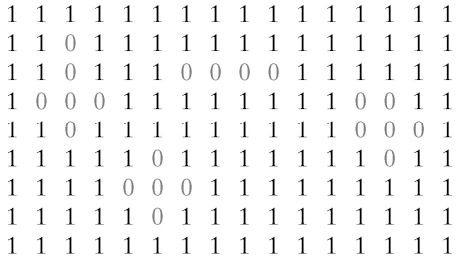


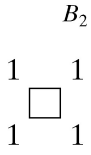


Original Image A and B1

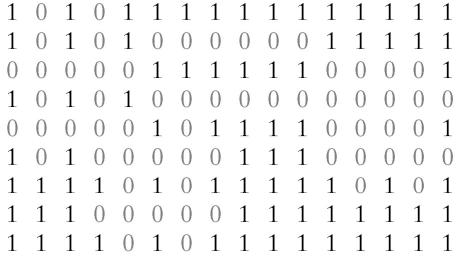


A eroded by B1

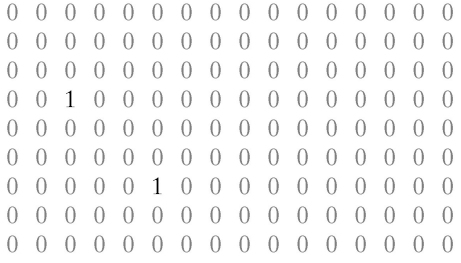




Complement of Original Image and B2



Erosion of A complement And B2



Intersection of eroded images

From: Digital Image Processing, Gonzalez,Woods And Eddins

Hit or Miss

- Have all the pixels in B1, but none of the pixels in B2

Hit or Miss Example #2

- Locate upper left hand corner pixels of objects in an image
- Pixels that have east and south neighbors (Hits) and no NE, N, NW, W, SW Pixels (Misses)

B1 =

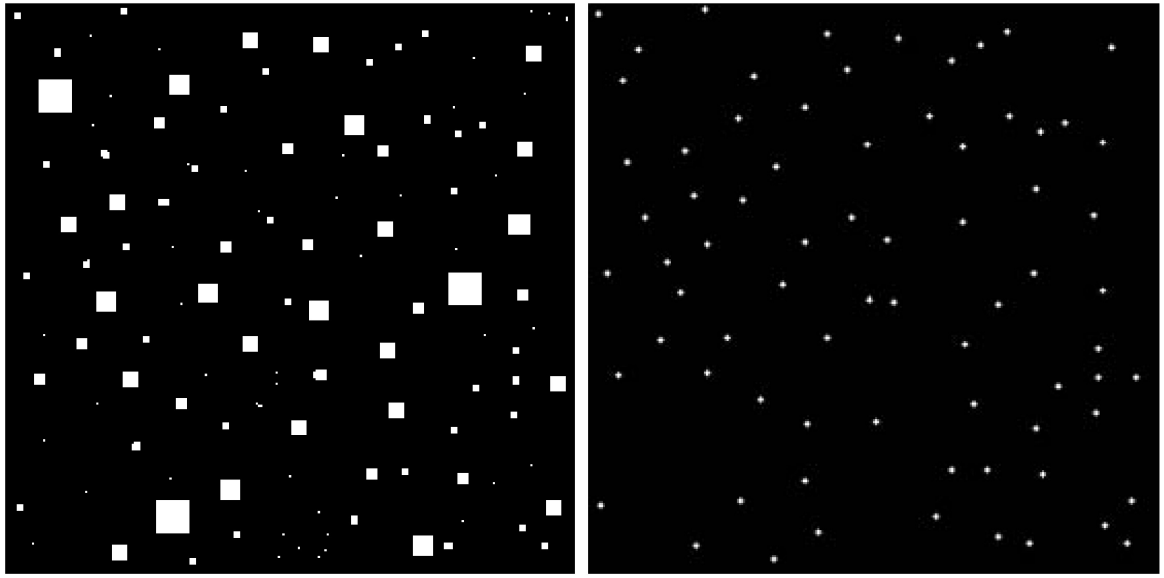
0	0	0
0	1	1
0	1	0

B2 =

1	1	1
1	0	0
1	0	0

Don't Care about SE

Hit or Miss in Matlab



a b

FIGURE 9.13
(a) Original image. (b) Result of applying the hit-or-miss transformation (the dots shown were enlarged to facilitate viewing).

G = bwhitmiss(f, B1, B2);
Figure, imshow(g)

From: Digital Image Processing, Gonzalez,Woods And Eddins

bwmorph(f, operation, n)

- Implements various morphological operations based on combinations of dilations, erosions and look up table operations.
- Example: Thinning

```
>> f = imread('fingerprint_cleaned.tif');
>> g = bwmorph(f, 'thin', 1);
>> g2 = bwmorph(f, 'thin', 2);
>> g3 = bwmorph(f, 'thin', Inf);
```

a b c

FIGURE 9.11 (a) Noisy fingerprint image. (b) Opening of image. (c) Opening followed by closing. (Original image courtesy of the National Institute of Standards and Technology.)

Input

From: Digital Image Processing, Gonzalez,Woods And Eddins

a b c

FIGURE 9.15 (a) Fingerprint image from Fig. 9.11(c) thinned once. (b) Image thinned twice. (c) Image thinned until stability.

```
>> f = imread('fingerprint_cleaned.tif');
>> g = bwmorph(f, 'thin', 1);
>> g2 = bwmorph(f, 'thin', 2);
>> g3 = bwmorph(f, 'thin', Inf);
```

From: Digital Image Processing, Gonzalez,Woods And Eddins

TABLE 9.3
Operations supported by function `bwmorph`.

Operation	Description
`bothat`	"Bottom-hat" operation using a 3×3 structuring element; use `imbothat` (see Section 9.6.2) for other structuring elements.
`bridge`	Connect pixels separated by single-pixel gaps.
`clean`	Remove isolated foreground pixels.
`close`	Closing using a 3×3 structuring element; use `imclose` for other structuring elements.
`diag`	Fill in around diagonally connected foreground pixels.
`dilate`	Dilation using a 3×3 structuring element; use `imdilate` for other structuring elements.
`erode`	Erosion using a 3×3 structuring element; use `imerode` for other structuring elements.
`fill`	Fill in single-pixel "holes" (background pixels surrounded by foreground pixels); use `imfill` (see Section 11.1.2) to fill in larger holes.
`hbreak`	Remove H-connected foreground pixels.
`majority`	Make pixel p a foreground pixel if at least five pixels in $N_8(p)$ (see Section 9.4) are foreground pixels; otherwise make p a background pixel.
`open`	Opening using a 3×3 structuring element; use function `imopen` for other structuring elements.
`remove`	Remove "interior" pixels (foreground pixels that have no background neighbors).
`shrink`	Shrink objects with no holes to points; shrink objects with holes to rings.
`skel`	Skeletonize an image.
`spur`	Remove spur pixels.
`thicken`	Thicken objects without joining disconnected 1s.
`thin`	Thin objects without holes to minimally connected strokes; thin objects with holes to rings.
`tophat`	"Top-hat" operation using a 3×3 structuring element; use `imtophat` (see Section 9.6.2) for other structuring elements.

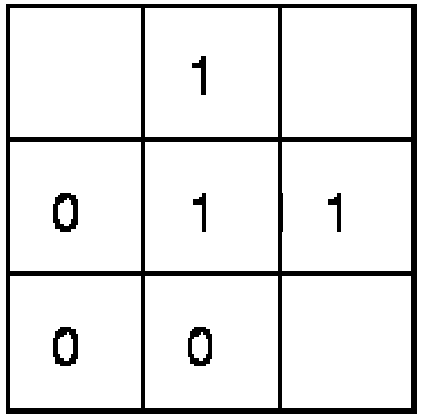
Hit-and-miss Transform is used for "Pattern Matching"

- Hit and Miss is used to look for particular patterns of foreground and background pixels

- It allows a recognition of very simple objects
- All other morphological operations can be derived from it!!
- Input:
  - Binary Image
  - Structuring Element, containing 0s and 1s!!

Example for a Hit-and-miss Structuring Element

- Contains 0s, 1s and *don't care*'s.
- Usually a "1" at the origin!

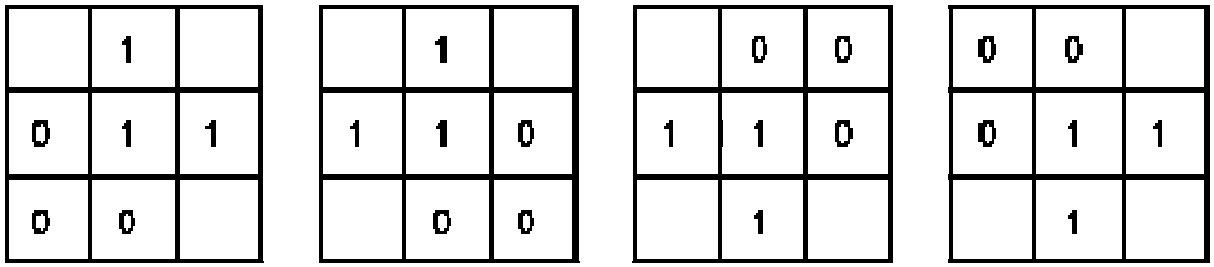


Hit-and-miss Transform as pattern matching

- It is one variant of a general to Pattern Matching approach:
  - If foreground and background pixels in the structuring element *exactly match* foreground and background pixels in the image,
  - then the pixel underneath the origin of the structuring element is set to the foreground color.

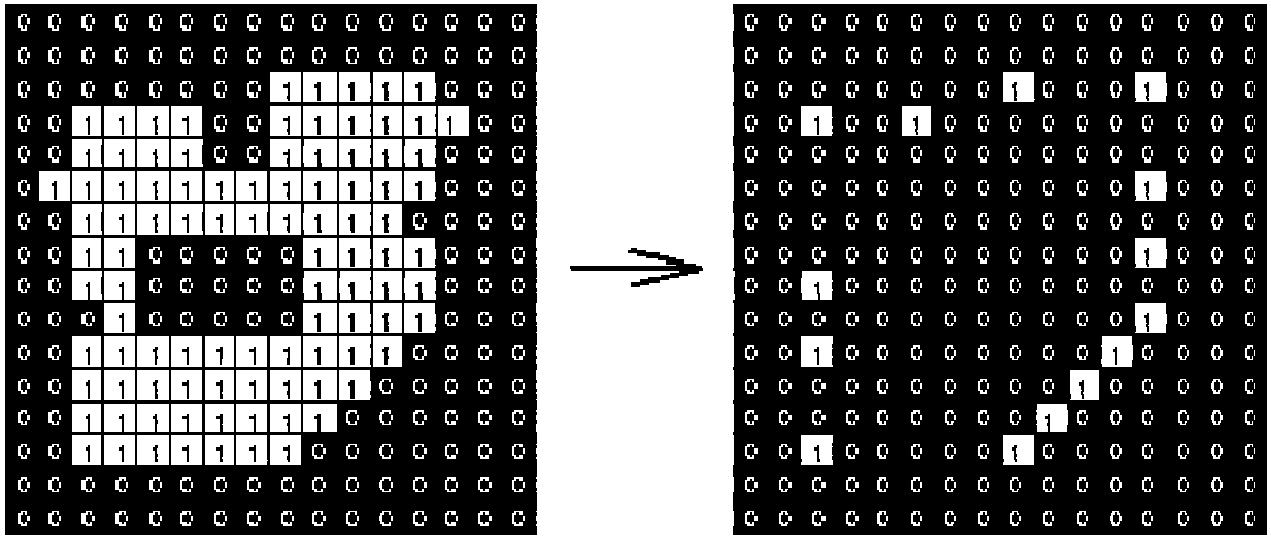
EXAMPLE: Corner Detection with Hit-and-miss Transform

- Structuring Elements representing four corners



Corner Detection with Hit-and-miss Transform

1. Apply Hit-&-Miss with each Structuring Element
2. Use OR operation to combine the four results



Allows to describe approximate shape of an object

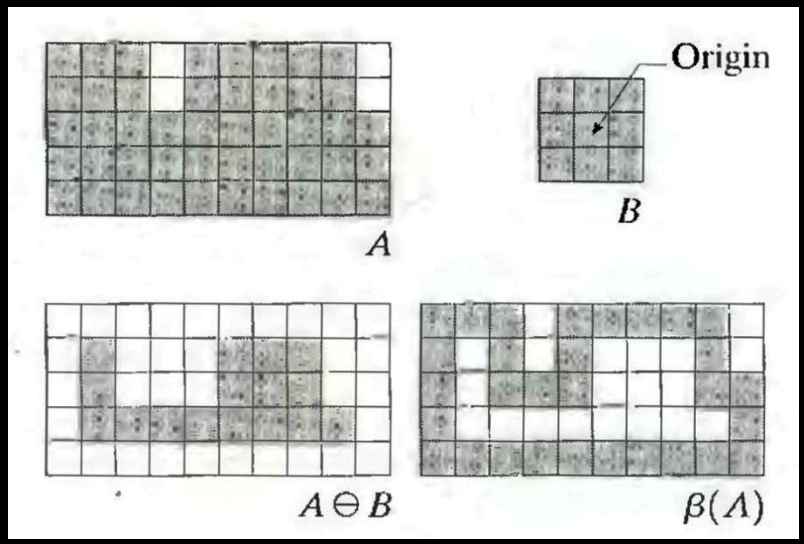
9.5 Basic Morphological Algorithms

- 1 – Boundary Extraction
- 2 – Region Filling
- 3 – Extraction of Connected Components
- 4 – Convex Hull
- 5 – Thinning
- 6 – Thickening
- 7 – Skeletons

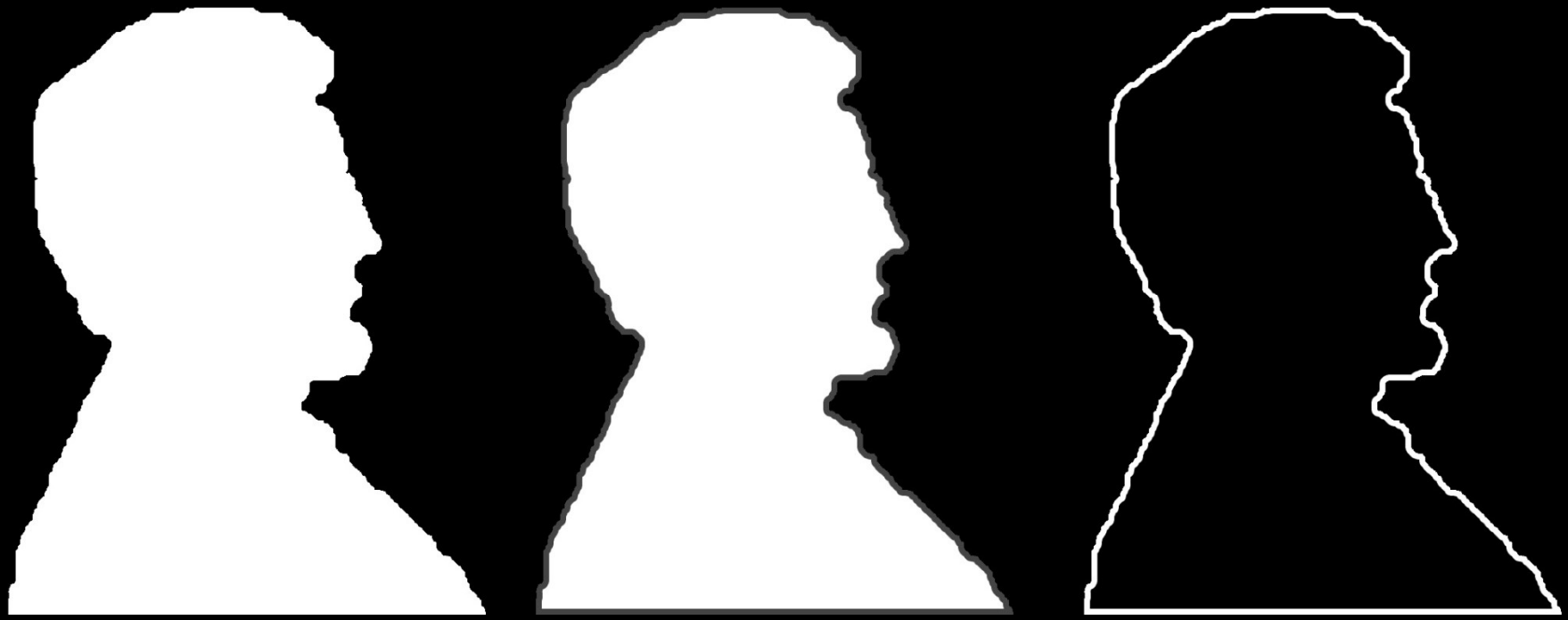
9.5.1 Boundary Extraction

- First, erode A by B, then make set difference between A and the erosion
- The thickness of the contour depends on the size of constructing object – B

$$\beta(A) = A - (A \ominus B)$$

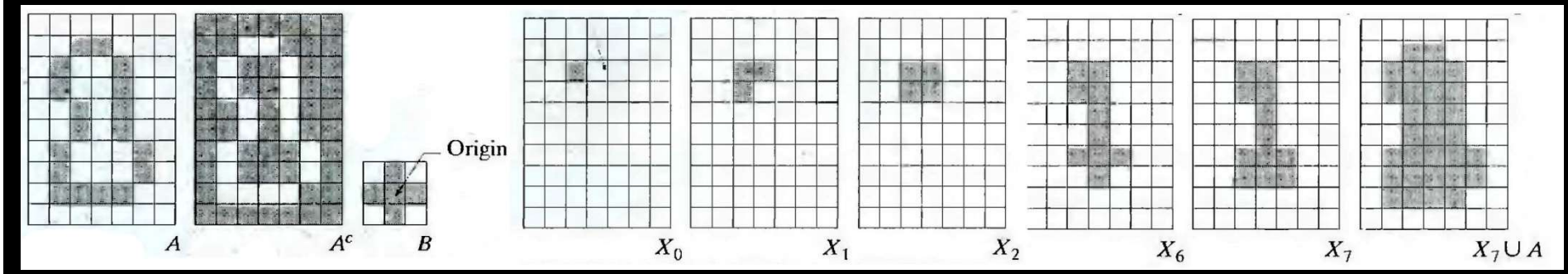


9.5.1 Boundary Extraction

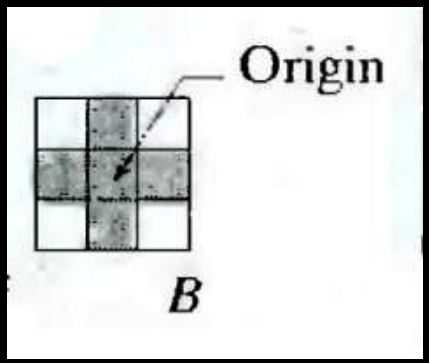


9.5.2 Region Filling

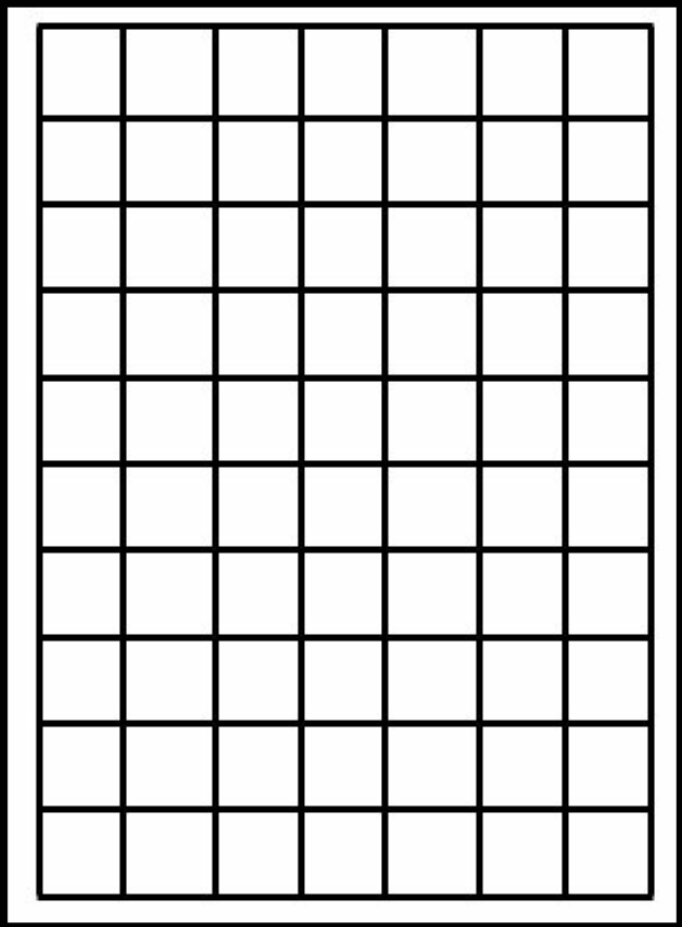
- This algorithm is based on a set of dilations, complementation and intersections
- p is the point inside the boundary, with the value of 1
- $X_{(k)} = (X_{(k-1)}$ xor B) conjunction with complemented A
- The process stops when $X_{(k)} = X_{(k-1)}$
- The result that given by union of A and $X_{(k)}$, is a set contains the filled set and the boundary



9.5.2 Region Filling



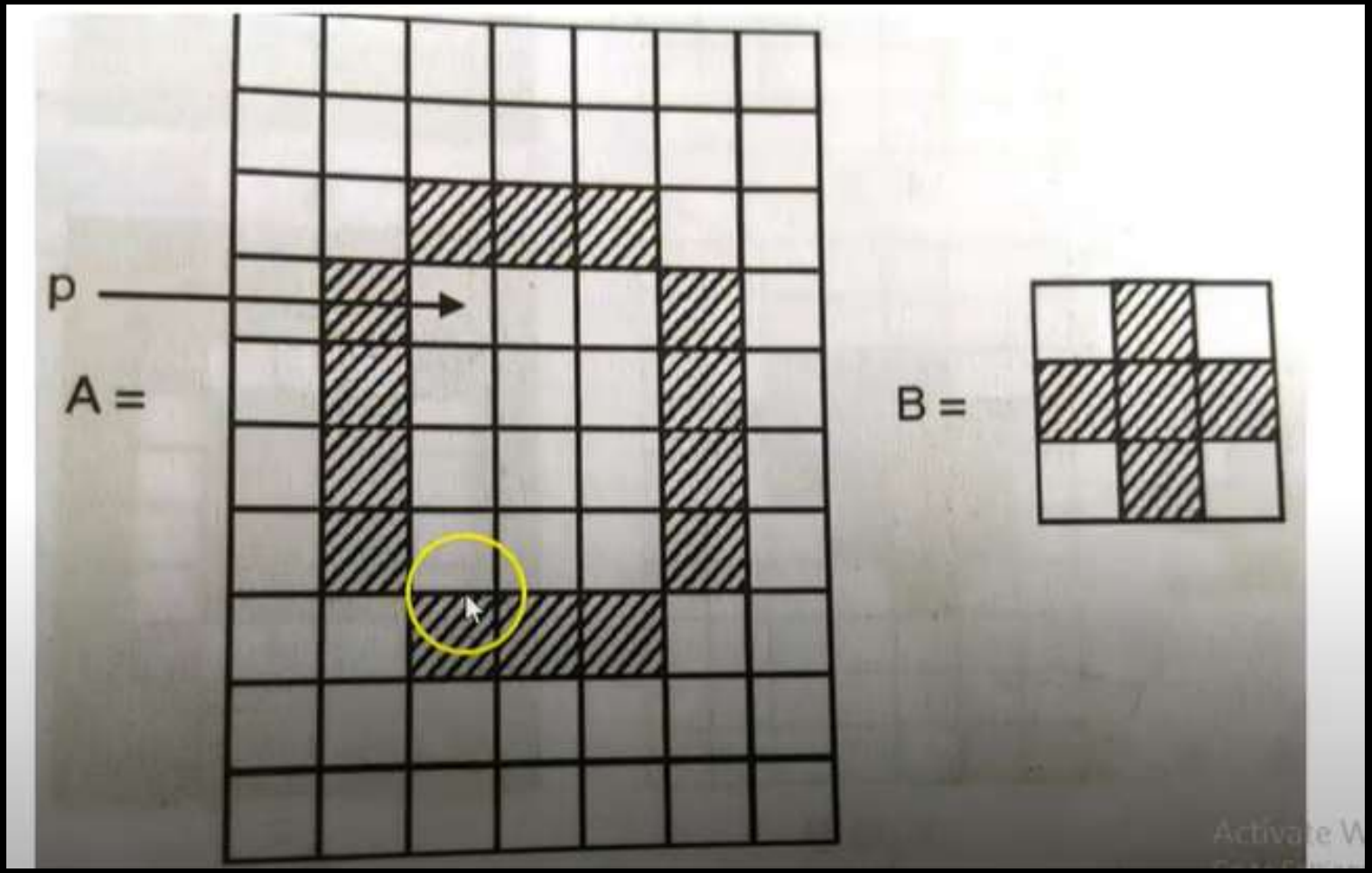
$$X_k = (X_{k-1} \oplus B) \cap A^c$$



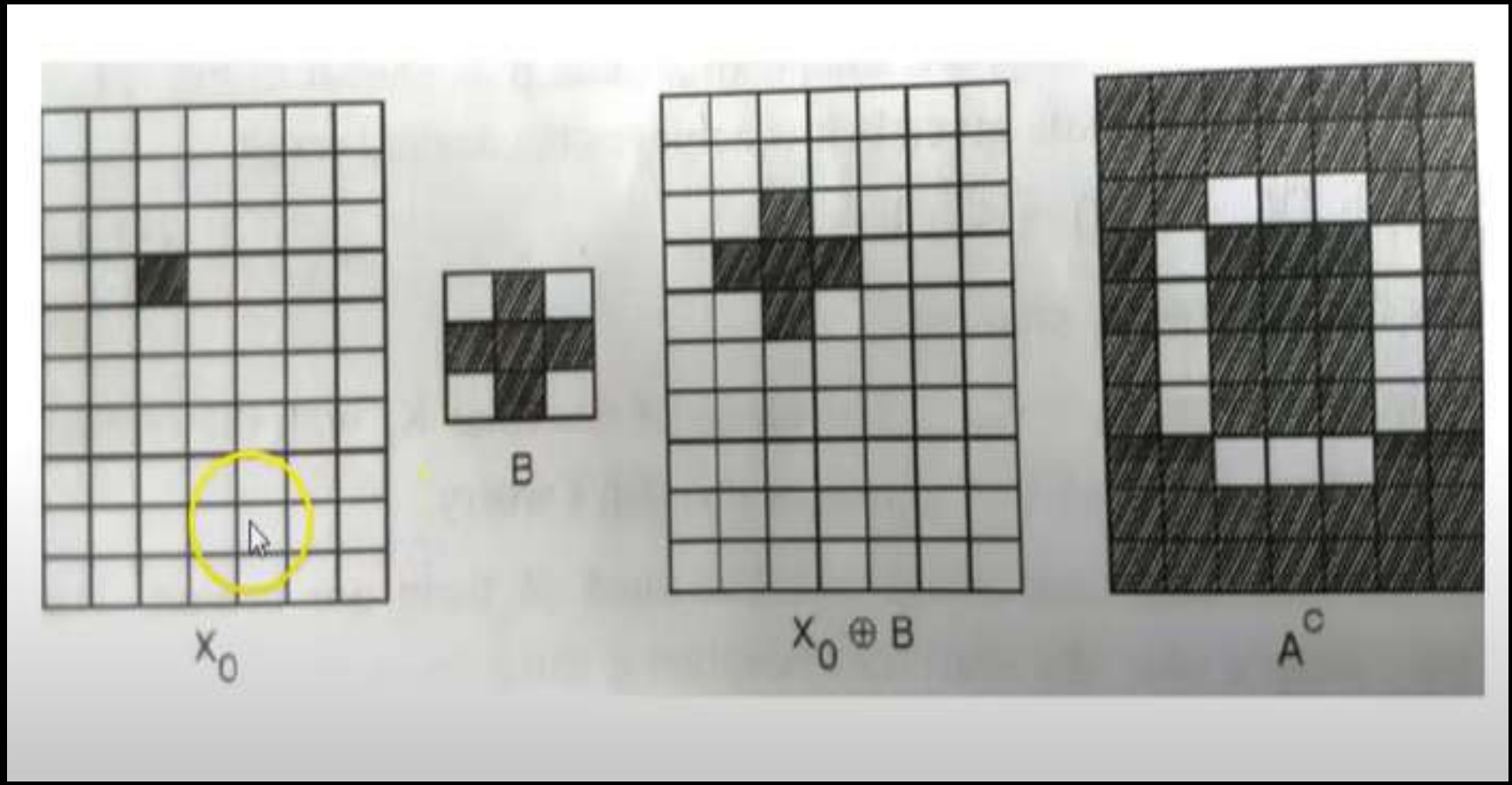
9.5.3 Extraction of Connected Components

- This algorithm extracts a component by selecting a point on a binary object A
- Works similar to region filling, but this time we use in the conjunction the object A, instead of it's complement

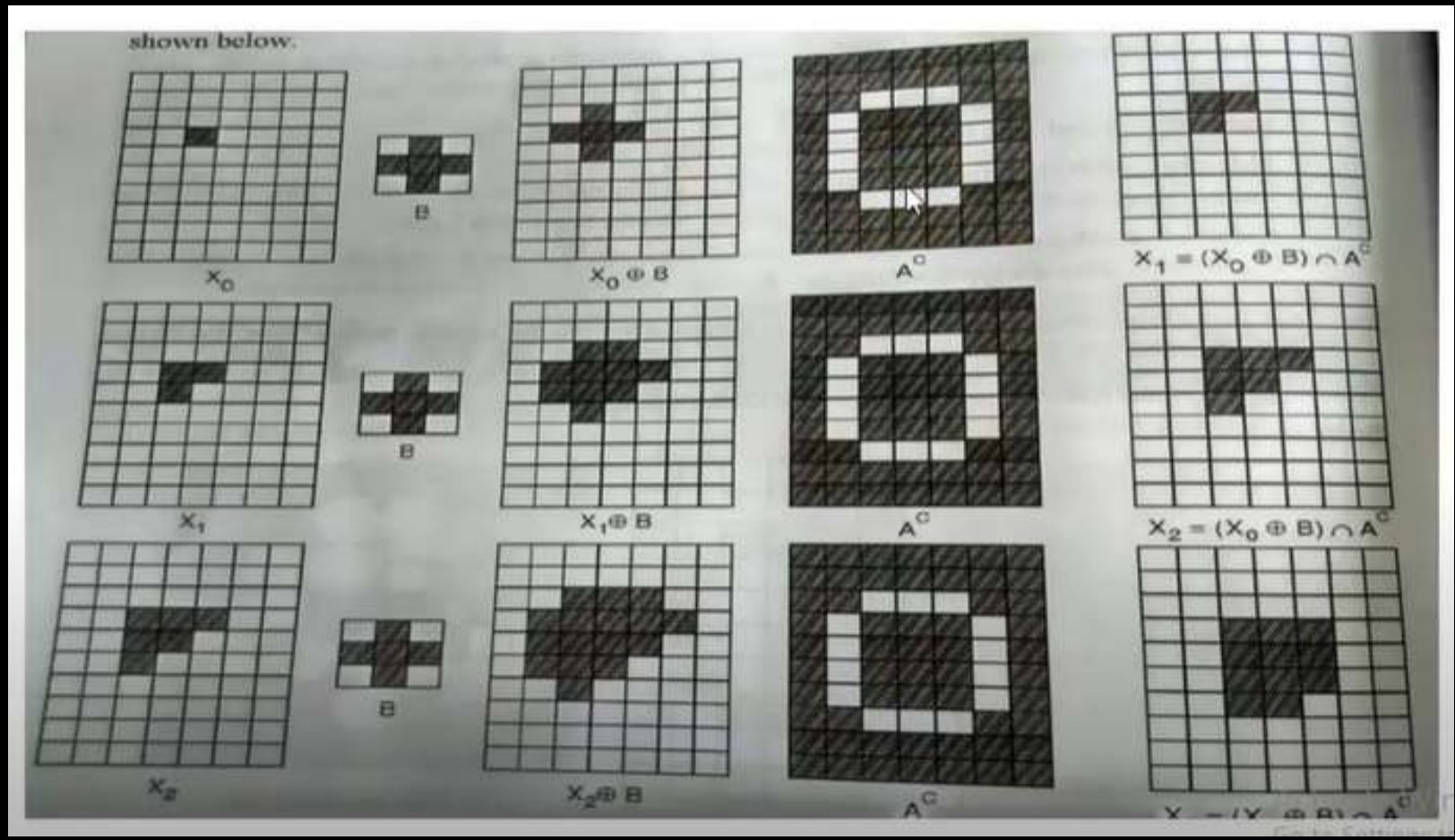
Region Filling Example



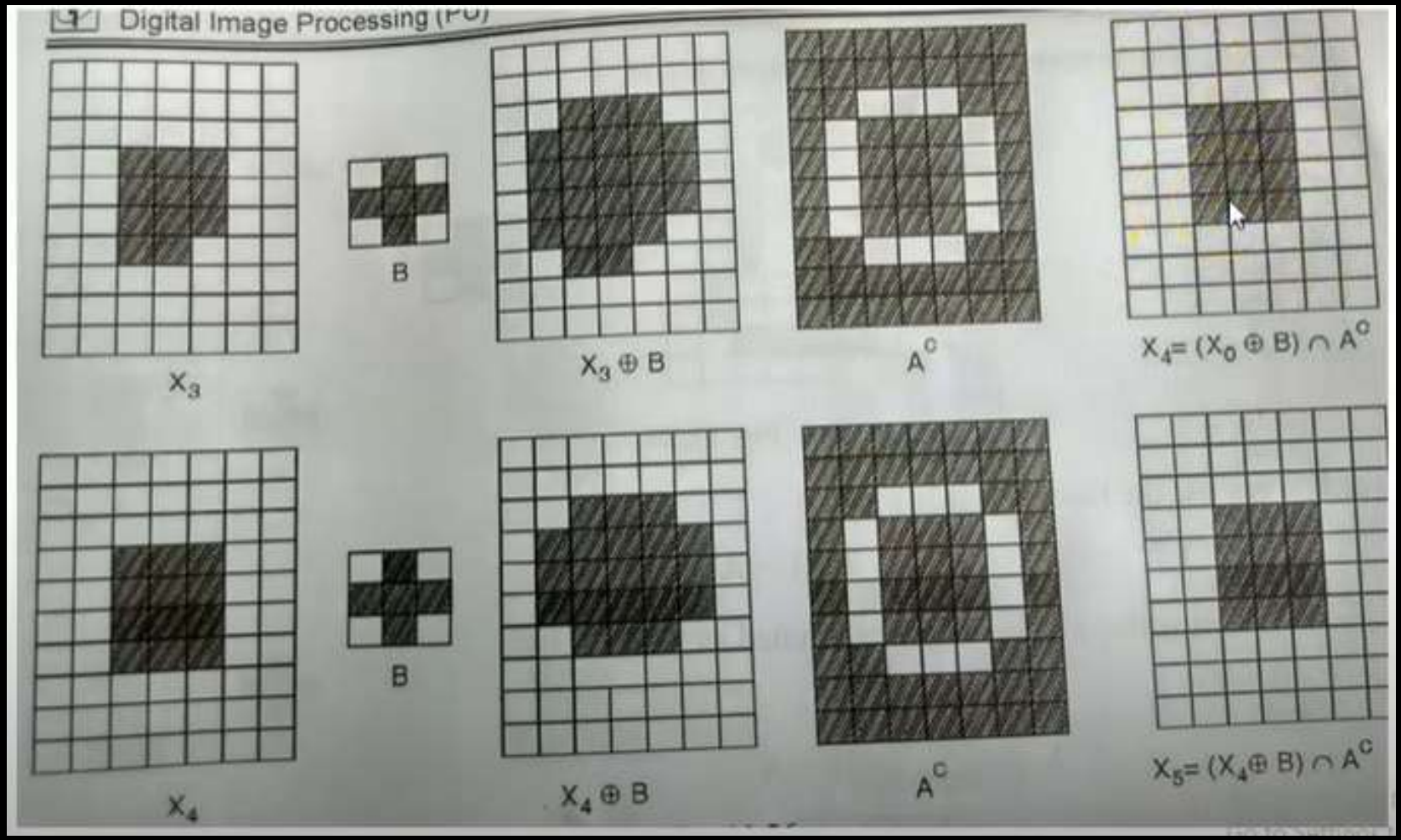
Region Filling Example (cond..)



Region Filling Example (cond..)



Region Filling Example (cond..)

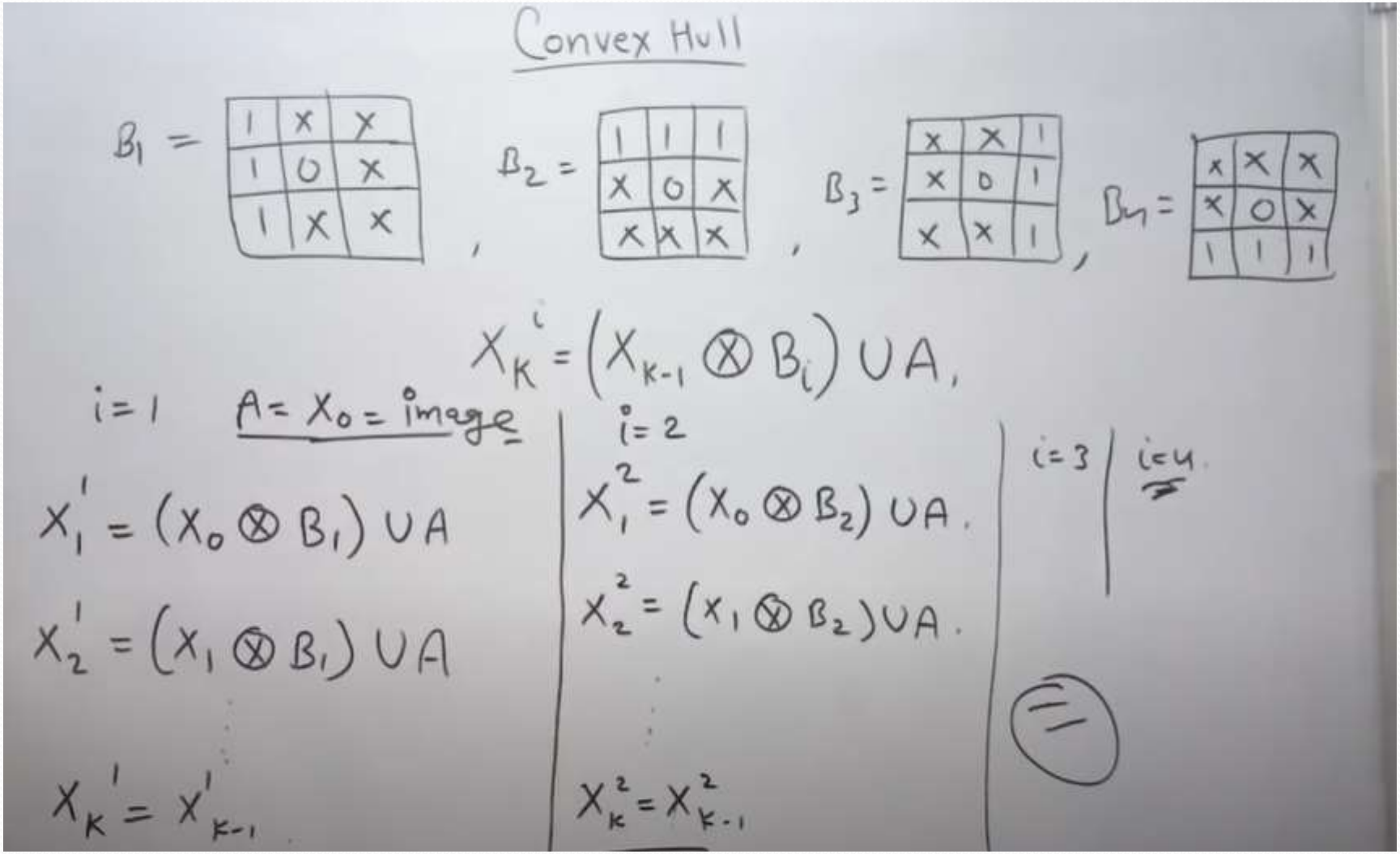


9.5.4 Convex Hull

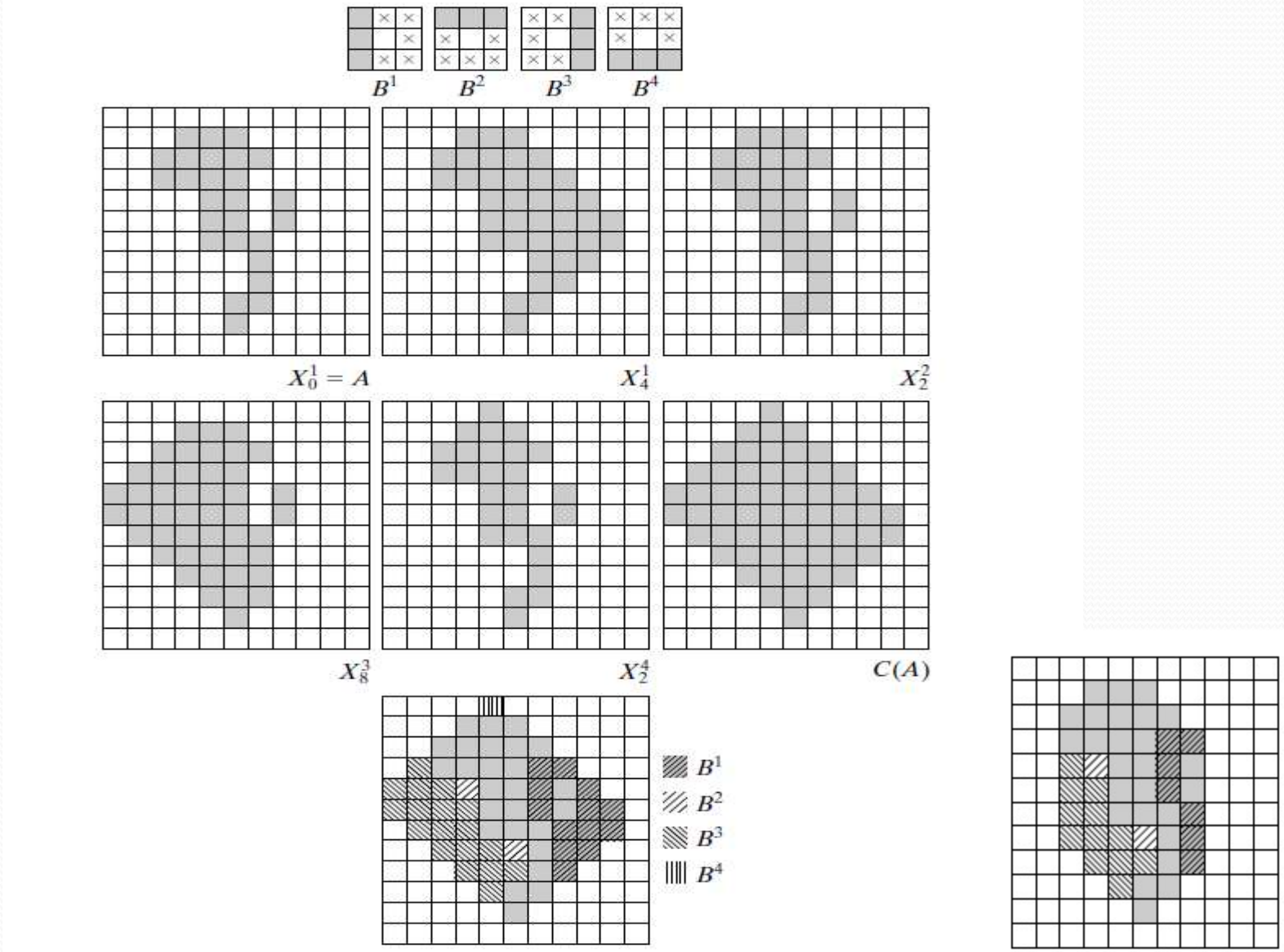
- A is said to be convex if a straight line segment joining any two points in A lies entirely within A
- The convex hull H of set S is the smallest convex set containing S
- Convex deficiency is the set difference H-S
- Useful for object description
- This algorithm iteratively applying the hit-or-miss transform to A with the first of B element, unions it with A, and repeated with second element of B

$$X_k^i = (X_{k-1} \circledast B^i) \cup A$$

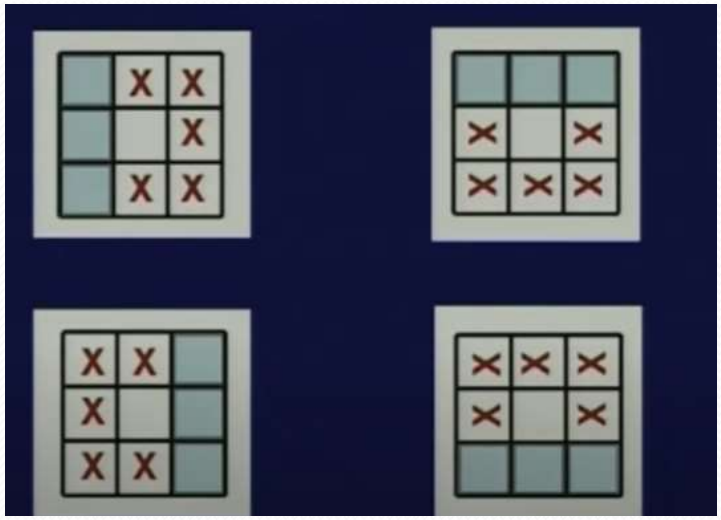
Convex hull example

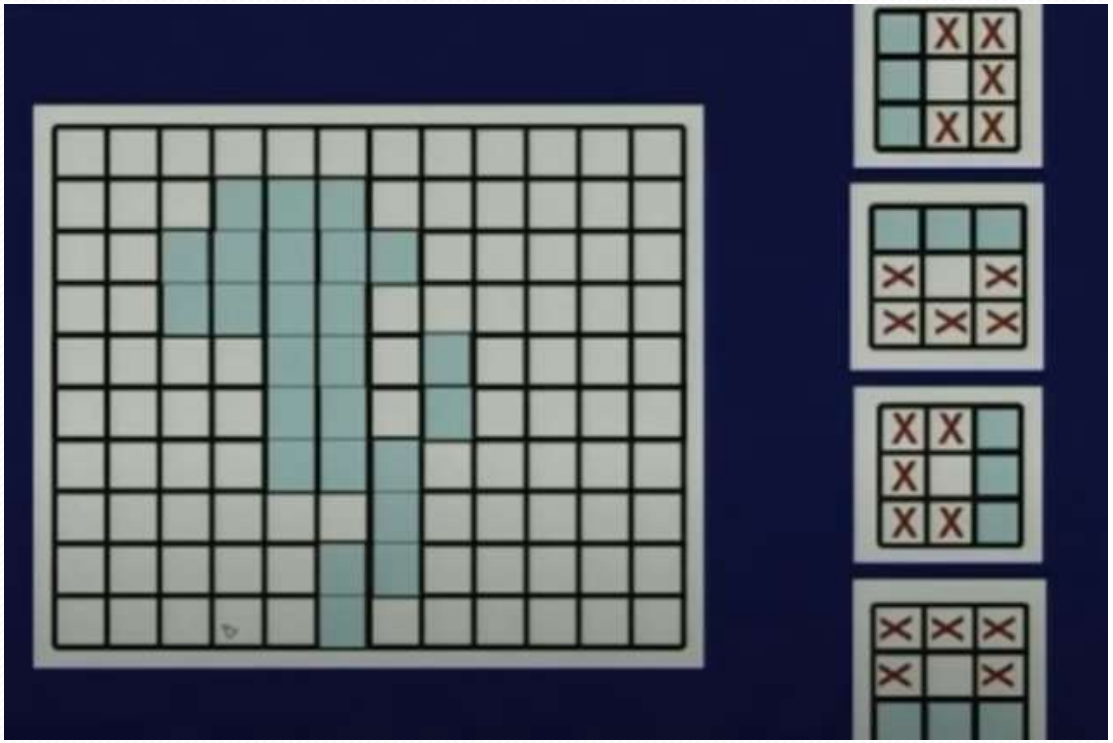


Convex hull example

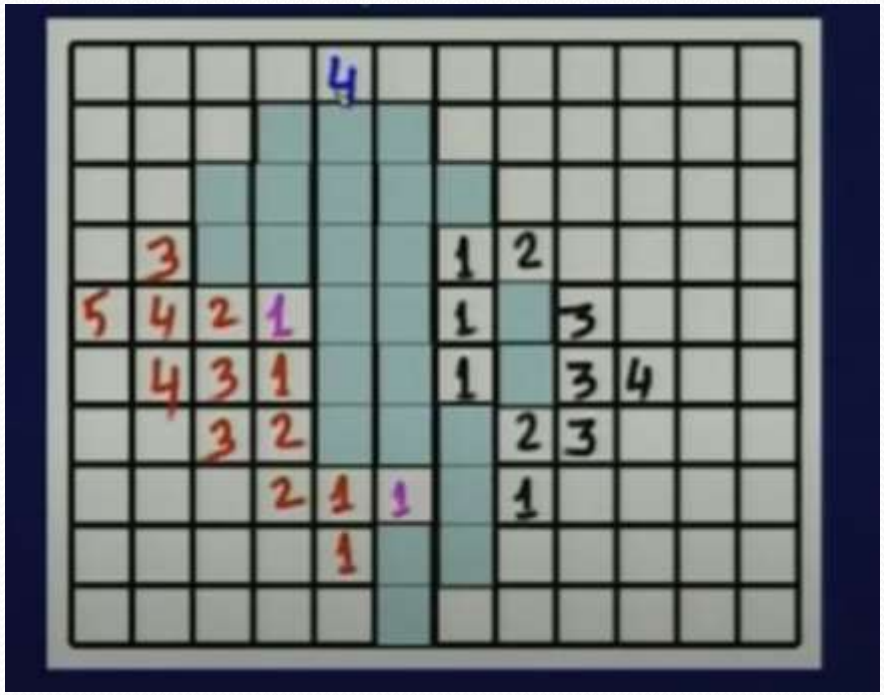


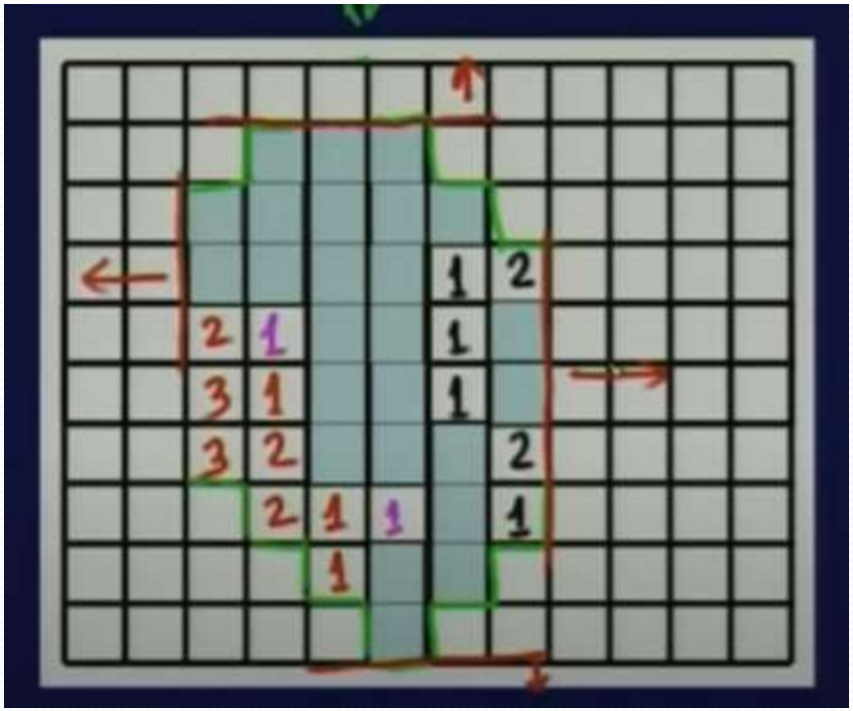
Convex hull example (cond..)





Convex hull example (cond..)





9.5.5 Thinning

- The thinning of a set A by a structuring element B, can be defined by terms of the hit-and-miss transform:

$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

- A more useful expression for thinning A symmetrically is based on a sequence of structuring elements:

  $\{B\}=\{B^1, B^2, B^3, \ldots, B^n\}$

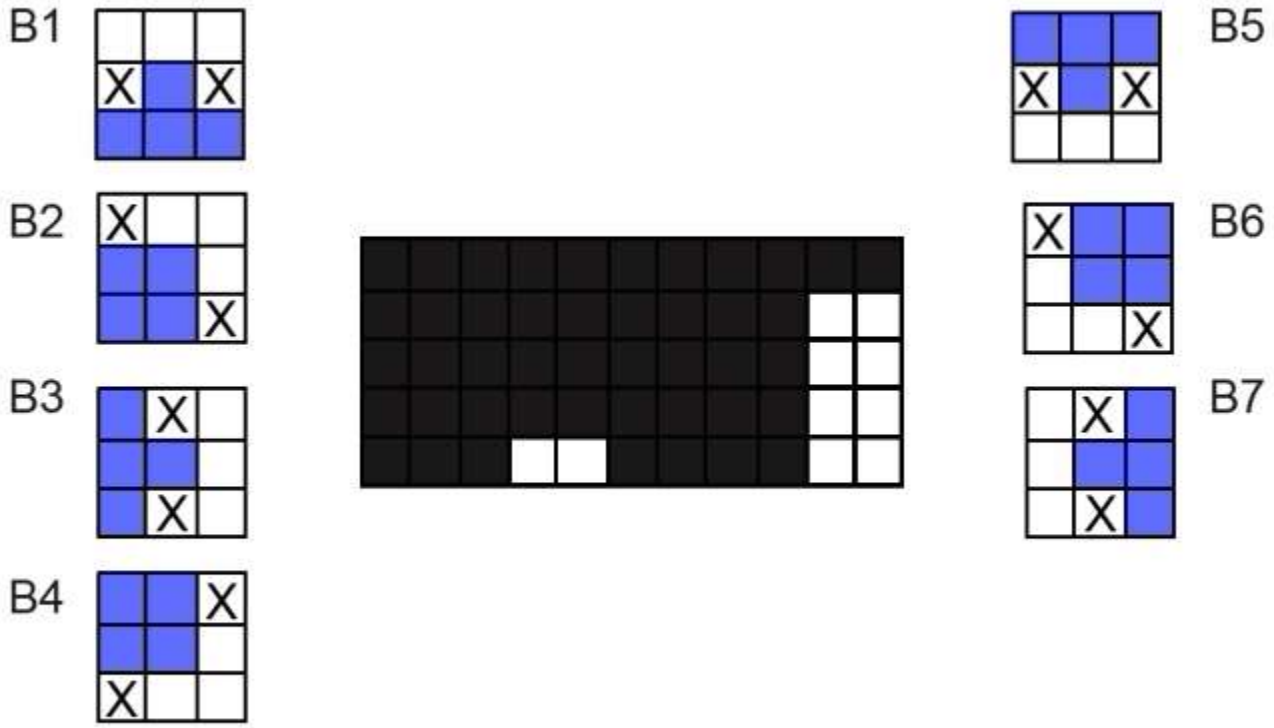
- Where B^i is a rotated version of B^{i-1}. Using this concept we define thinning by a sequence of structuring elements: $A \otimes \{B\} = ((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$

9.5.5 Thinning cont

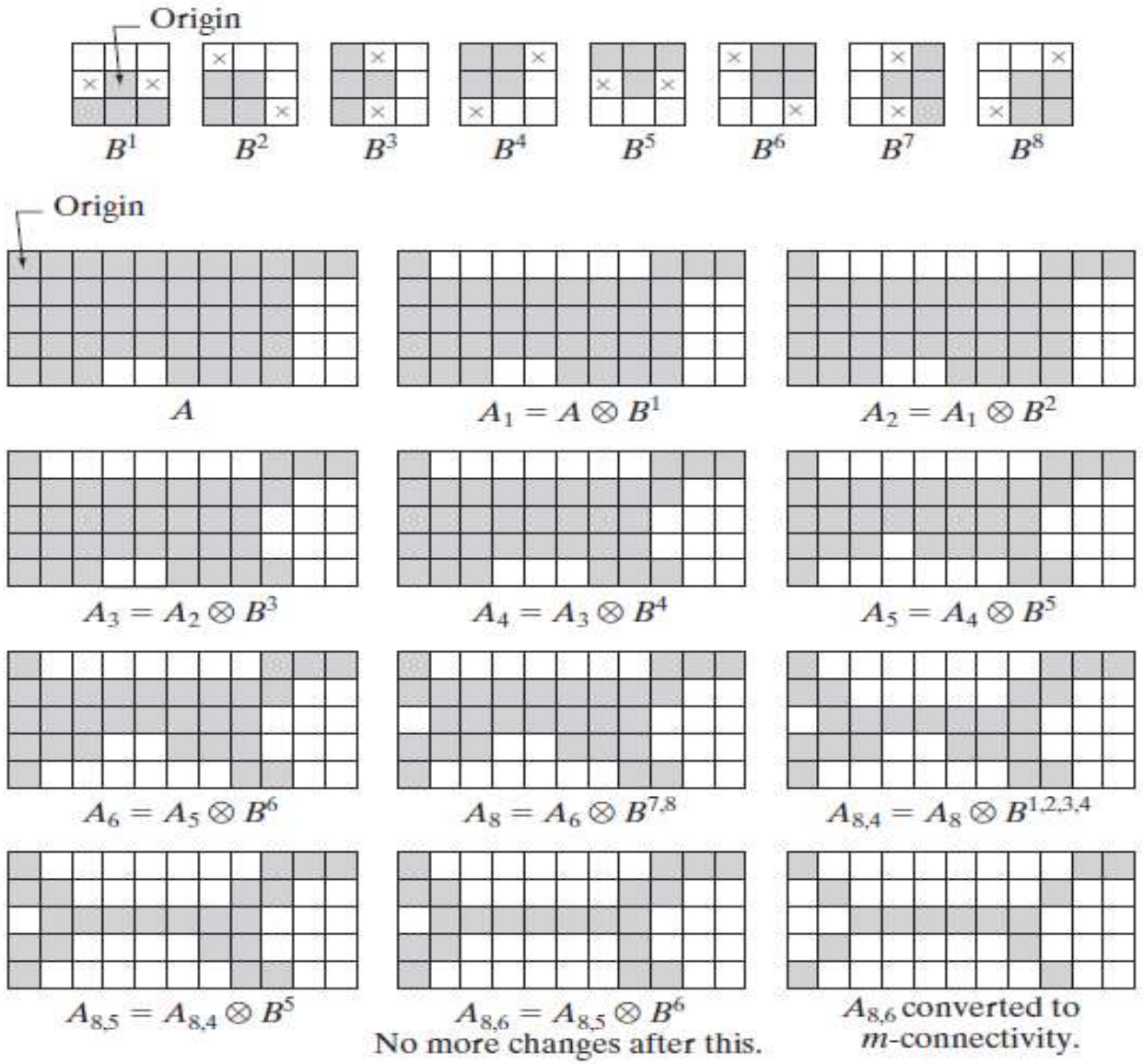
- The process is to thin by one pass with B^1 , then thin the result with one pass with B^2, and so on until A is thinned with one pass with B^n.
- The entire process is repeated until no further changes occur.
- Each pass is preformed using the equation:

$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

9.5.5 Thinning example



9.5.5 Thinning example



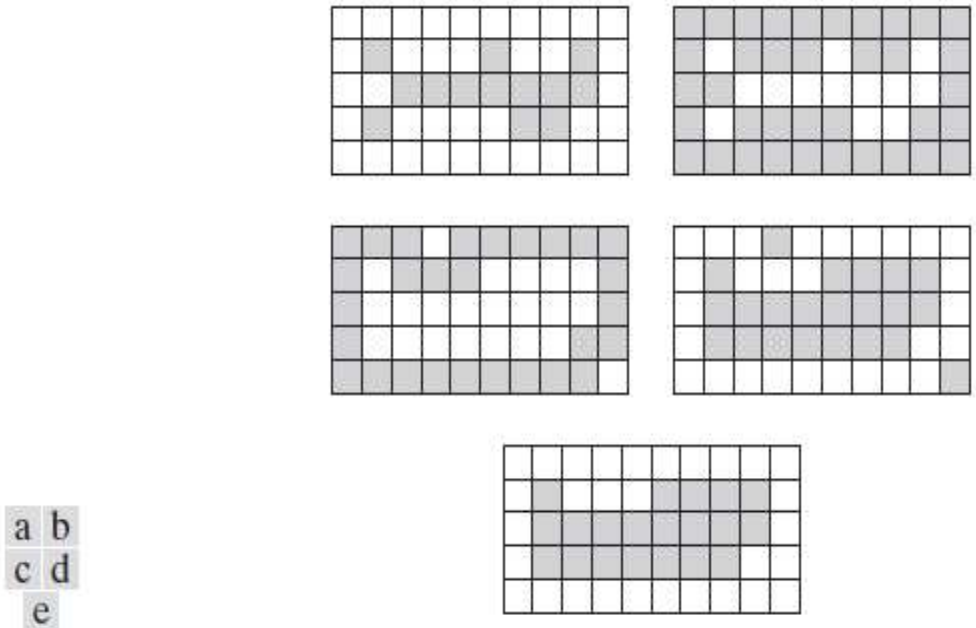
9.5.6 Thickening

- Thickening is a morphological dual of thinning.
- Definition of thickening $A \odot B = A \cup (A \circledast B)$.
- As in thinning, thickening can be defined as a sequential operation:

$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n)$$

- the structuring elements used for thickening have the same form as in thinning, but with all 1's and 0's interchanged.

9.5.6 Thickening



a b
c d
e

FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

9.5.6 Thickening - cont

- A separate algorithm for thickening is often used in practice, Instead the usual procedure is to thin the background of the set in question and then complement the result.

- In other words, to thicken a set A, we form $C=A^c$, thin C and than form C^c.

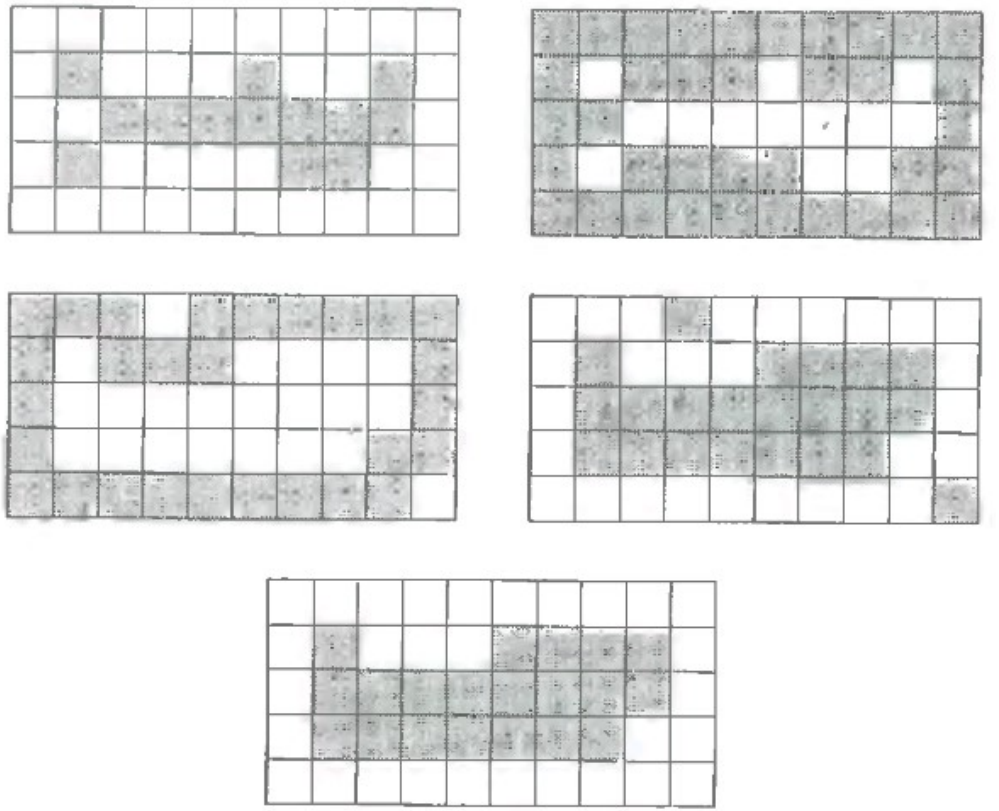
- depending on the nature of A, this procedure may result in some disconnected points. Therefore thickening by this procedure usually require a simple post-processing step to remove disconnected points.

9.5.6 Thickening example preview

- We will notice in the next example 9.22(c) that the thinned background forms a boundary for the thickening process, this feature does not occur in the direct implementation of thickening

- This is one of the reasons for using background thinning to accomplish thickening.

9.5.6 Thickening example



a b
c d
e

FIGURE 9.22 (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.

9.5.7 Skeleton

- The notion of a skeleton S(A) of a set A is intuitively defined, we deduce from this figure that:

a) If z is a point of S(A) and (D)z is the largest disk centered in z and contained in A (one cannot find a larger disk that fulfils this terms) – this disk is called "maximum disk".

b) The disk (D)z touches the boundary of A at two or more different places.

9.5.7 Skeleton

- The skeleton of A is defined by terms of erosions and openings:

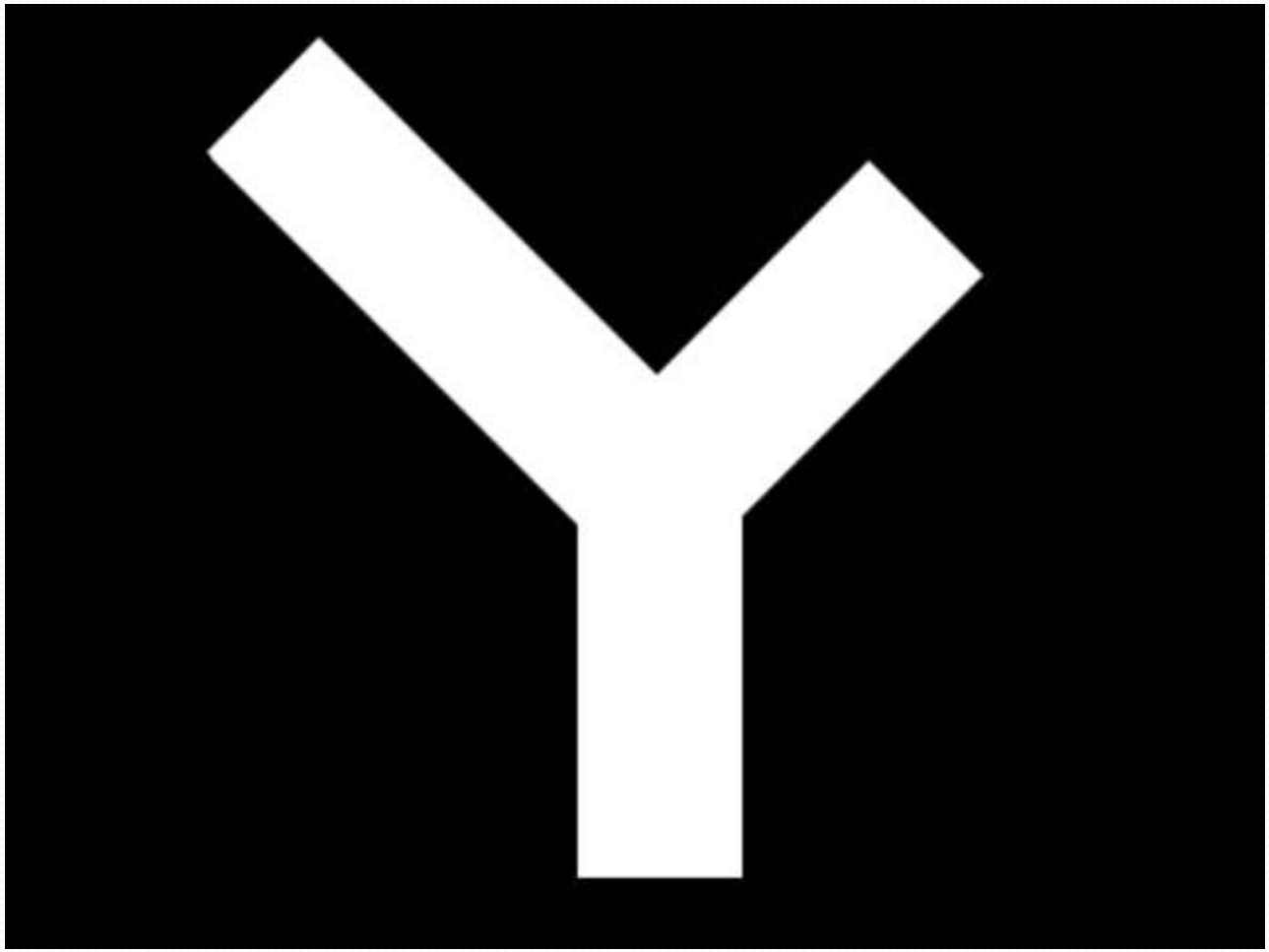
$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

- with $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$
- Where B is the structuring element and $(A \ominus kB)$ indicates k successive erosions of A:

$$(A \ominus kB) = (\dots((A \ominus B) \ominus B) \ominus \dots) \ominus B$$

  - k times, and K is the last iterative step before A erodes to an empty set, in other words: $K = \max\{k | (A \ominus kB) \neq \emptyset\}$
- in conclusion S(A) can be obtained as the union of skeleton subsets Sk(A).

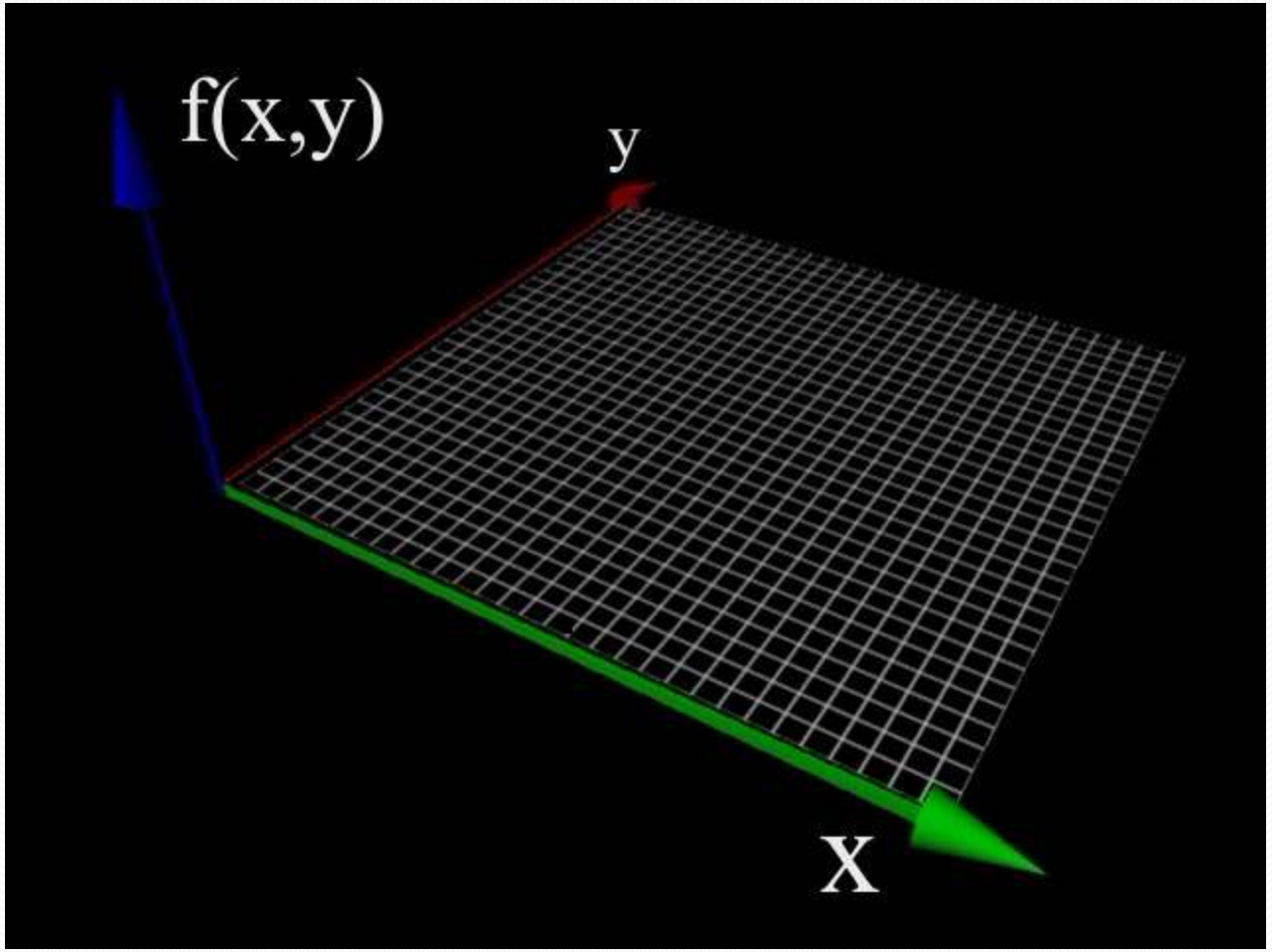
9.5.7 Skeleton Example



9.6 Gray-Scale Images

- In gray scale images on the contrary to binary images we deal with digital image functions of the form f(x,y) as an input image and b(x,y) as a structuring element.
- (x,y) are integers from Z*Z that represent a coordinates in the image.
- f(x,y) and b(x,y) are functions that assign gray level value to each distinct pair of coordinates.
- For example the domain of gray values can be 0-255, whereas 0 – is black, 255- is white.

9.6 Gray-Scale Images



Thickening

- Used to grow selected regions of foreground pixels
- E.g. applications like approximation of *convex hull*

Sources Used

1. Volker Krüger
2. Rune Andersen
3. . Roger S. Gaborski