# CSC 425
# Digital Image Processing

**Dr. Maryam Al-Berry**
maryam_nabil@cis.asu.edu.eg

# Region-based Segmentation

# Region-based Segmentation

## Basic Formulation

- **Partition image region $R$ into subregions such that:**

a) $\bigcup_{i=1}^{n} R_i = R$

b) $R_i$ is a connected region.

c) $R_i \cap R_j = \phi$

d) $P(R_i) = True$

e) $P(R_i \cup R_j) = False$

- **$P(R_i)$: logical predicate property defined over $R_i$.**

# Region-based Segmentation – (cont.)

## 1- Region Growing

- **Group based a predefined criteria of growth.**
- **Start with a set of seed points.**
- **Append to each seed neighboring pixels that satisfy the criteria (have properties similar to the seed).**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 5 | 6 | 7 |
| 1 | 1 | 5 | 8 | 7 |
| 0 | 1 | 6 | 7 | 7 |
| 2 | 0 | 7 | 6 | 6 |
| 0 | 1 | 5 | 6 | 5 |

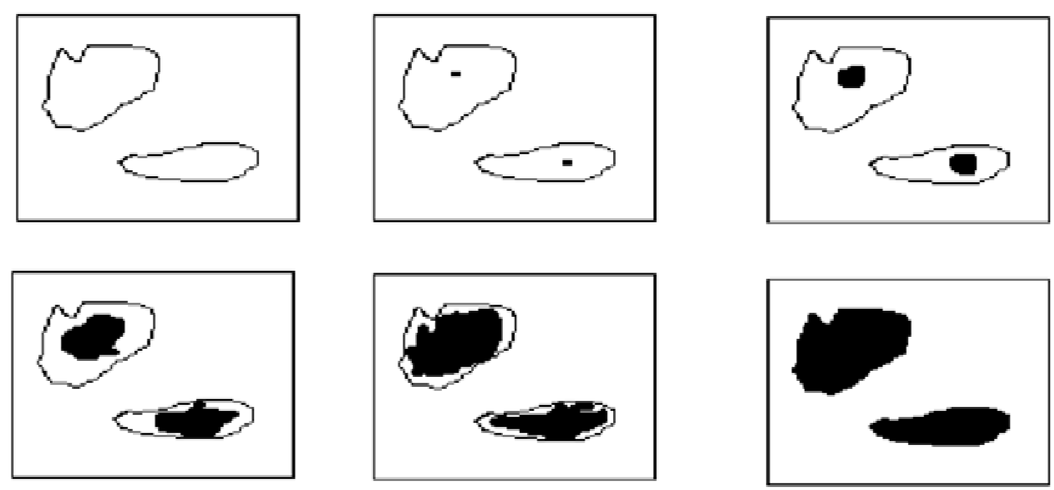
T=3

| | | | | |
|---|---|---|---|---|
| A | A | B | B | B |
| A | A | B | B | B |
| A | A | B | B | B |
| A | A | B | B | B |
| A | A | B | B | B |

# Region-based Segmentation – (cont.)

## 1- Region Growing

- **Choice of seed**
- **Choice of criteria**
- **Connectivity properties**
- **Stopping rule**
- **Additional criteria**



# Region-based Segmentation – (cont.)

## 1- Region Growing

- **Choice of seed**
  - It depends on the nature of the problem.
  - If targets need to be detected using infrared images for example, choose the brightest pixel(s).
  - Without apriori knowledge, compute the histogram and choose the gray-level values corresponding to the strongest peaks.

# Region-based Segmentation – (cont.)

## 1- Region Growing

- **Choice of criteria**
  - **The homogeneity predicate can be based on any characteristic of the regions in the image such as**
    - **average intensity**
    - **variance**
    - **color**
    - **texture**
    - **motion**
    - **shape**
    - **size**

# Region-based Segmentation – (cont.)

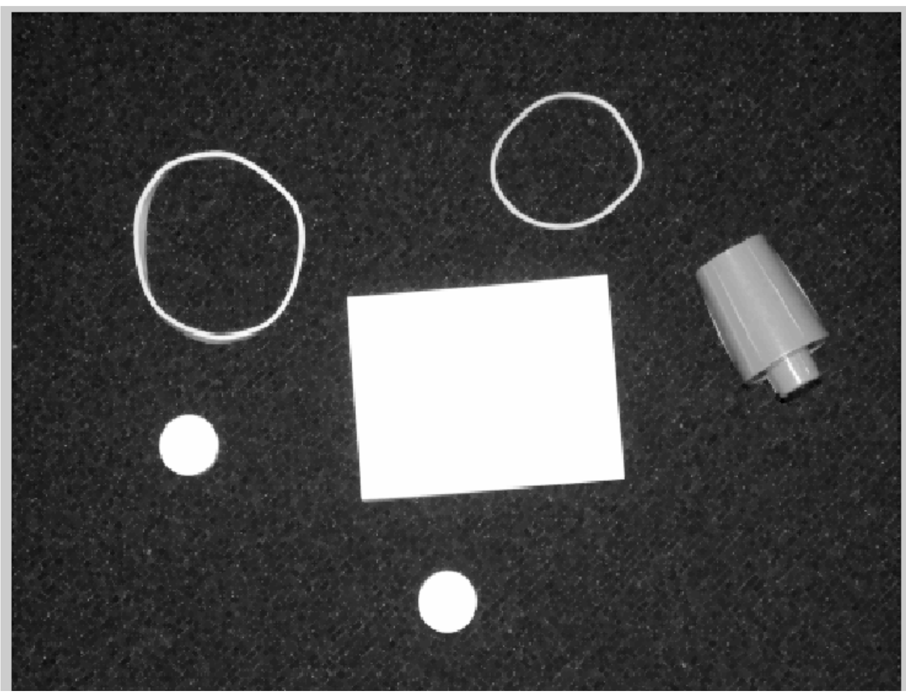
## 1- Region Growing

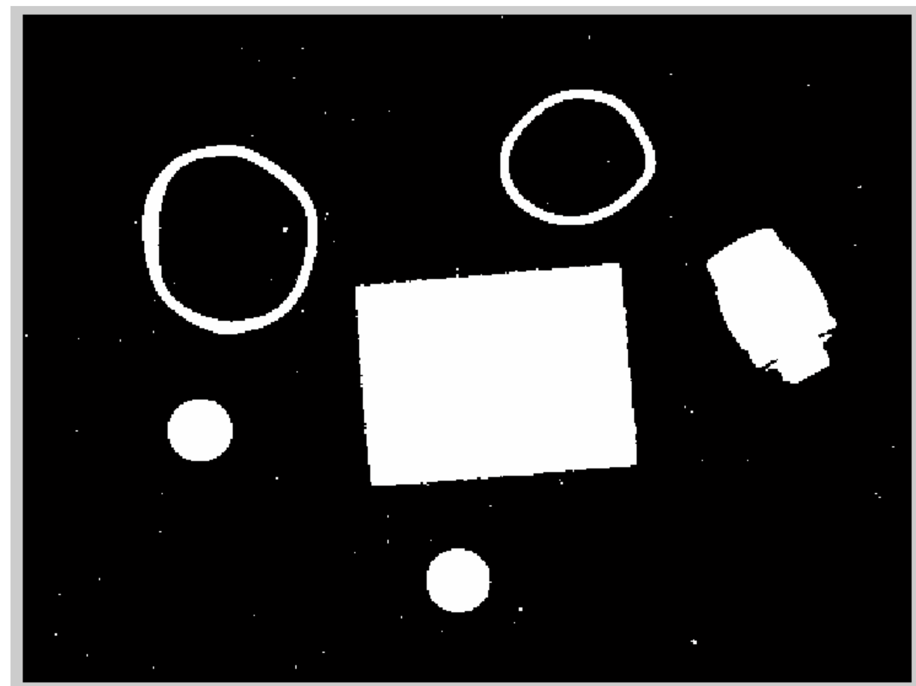
- **Choice of seed**
- **Choice of criteria**
- **Connectivity properties**
  - Four connectivity vs eight connectivity.
- **Stopping rule**
  - When criteria is not met.

# Region-based Segmentation – (cont.)

## 1- Region Growing

- **Example**

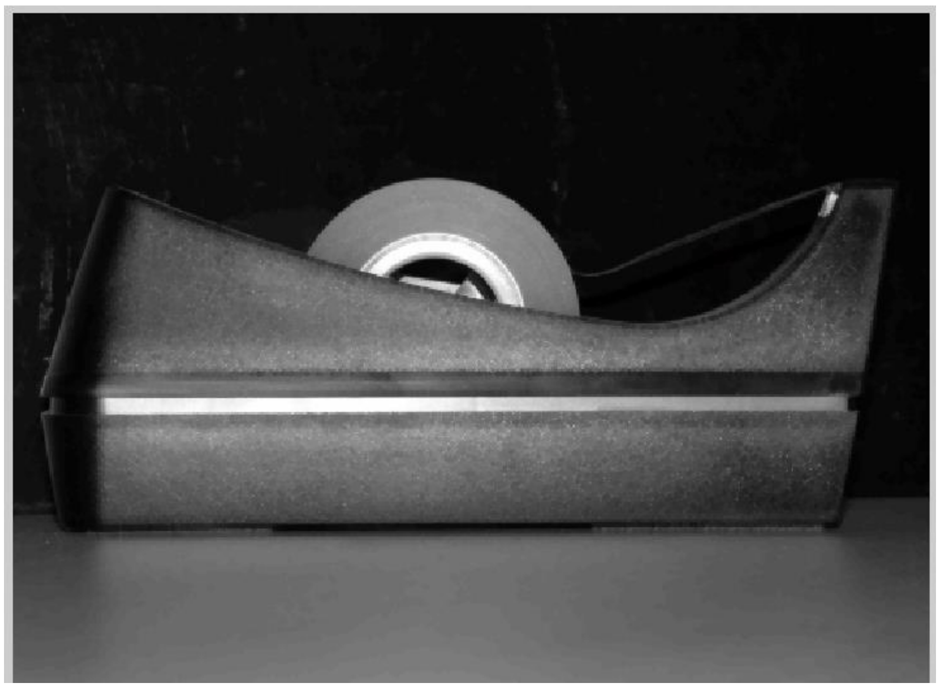


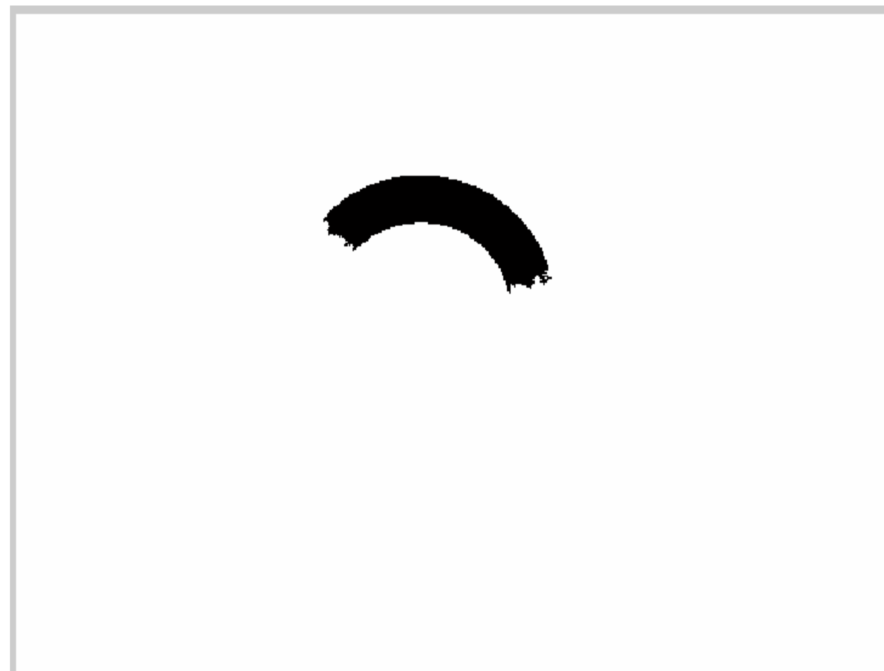


# Region-based Segmentation – (cont.)

## 1- Region Growing

- **Example**





# Region-based Segmentation – (cont.)

## 1- Region Growing

- **Advantages:**

1. Region growing methods can correctly separate the regions that have the same properties we define.
2. Region growing methods can provide the original images which have clear edges with good segmentation results.
3. The concept is simple. We only need a small number of seed points to represent the property we want, then grow the region.
4. We can determine the seed points and the criteria we want to make.
5. We can choose the multiple criteria at the same time.
6. It performs well with respect to noise.

# Region-based Segmentation – (cont.)

## 1- Region Growing

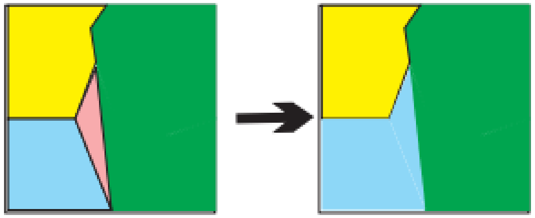
- **Disadvantage:**

1. The computation is consuming.
2. Noise or variation of intensity may result in holes or over segmentation.
3. This method may not distinguish the shading of the real object.

# Region-based Segmentation – (cont.)

## 2- Region Merging

– Region Merging - recursively merge regions that are similar.



(1) Form initial regions in the image.

(2) Build a regions adjacency graph (RAG).

(3) For each region do:

(3.1) Consider its adjacent region and test to see if they are similar.

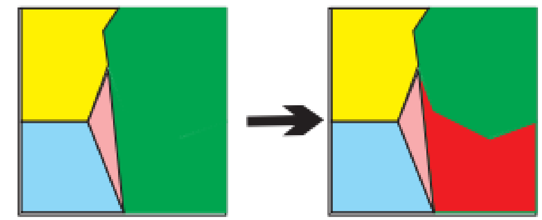
(3.2) For regions that are similar (i.e., $P(R_i \cup R_j)$=True), merge them and modify the RAG.

(4) Repeat step 3 until no regions are merged.

# Region-based Segmentation – (cont.)

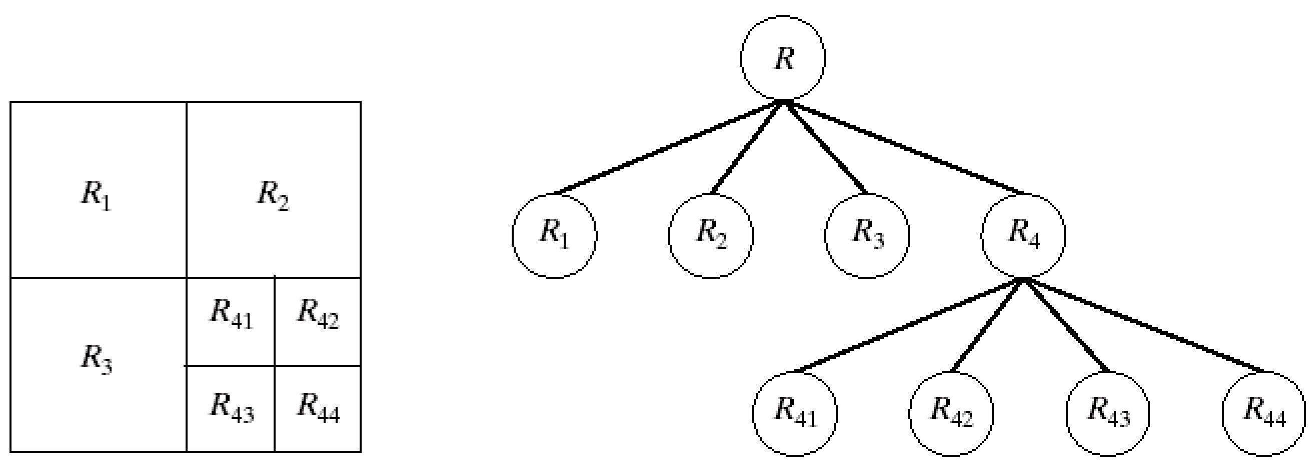
## 3- Region Splitting

– Region Splitting - recursively divide regions that are heterogeneous.



(1) If $P(R)$=False, split $R$ into four quadrants

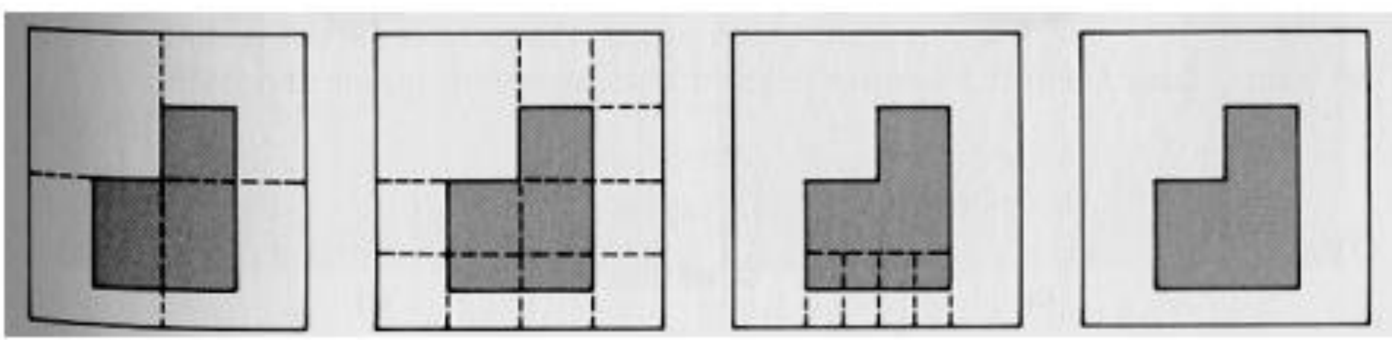
(2) If $P$ is false on any quadrant, subsplit



# Region-based Segmentation – (cont.)

## 4- Region Splitting and Merging

- Subdivide image initially into a set of arbitrary disjoint regions.
- Then merge and/or split the regions to satisfy the segmentation conditions.



# Region-based Segmentation – (cont.)

## 4- Region Splitting and Merging

- **Quadtree:**

- Split into 4 disjoint quadrants any region $R_i$ for which $P(R_i)$ = FALSE.

- Merge any adjacent region $R_j$ and $R_k$ for which $P(R_i \cup R_k)$ = TRUE

- Stop when no further merging or splitting is possible.

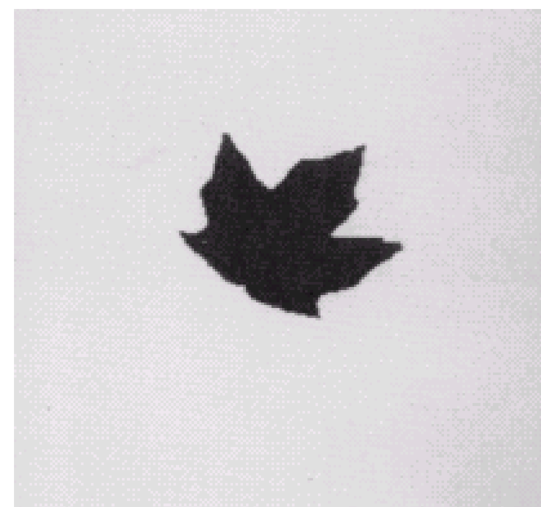
# Region-based Segmentation – (cont.)

## 4- Region Splitting and Merging

**Example:** $P(R_i)$ = TRUE if at least 80% of the pixels in $R_i$ have the property $|z_j - m_i| \leq 2\sigma_i$

Original Image

Result of split/merge



Result of thresholding

# Region-based Segmentation – (cont.)

## 4- Region Splitting and Merging

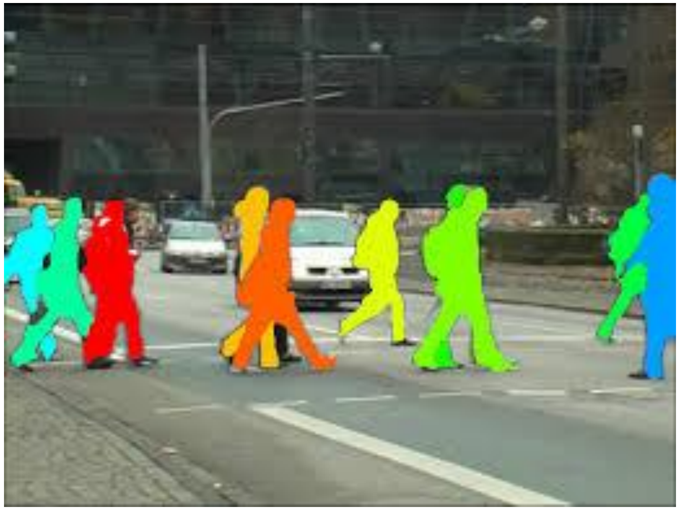
- **Advantages:**
  - The image could be split progressively according to our demanded resolution because the number of splitting level is determined by us.
  - We could split the image using the criteria we decide, such as mean or variance of segment pixel value.
  - In addition, the merging criteria could be different to the splitting criteria.

# Region-based Segmentation – (cont.)

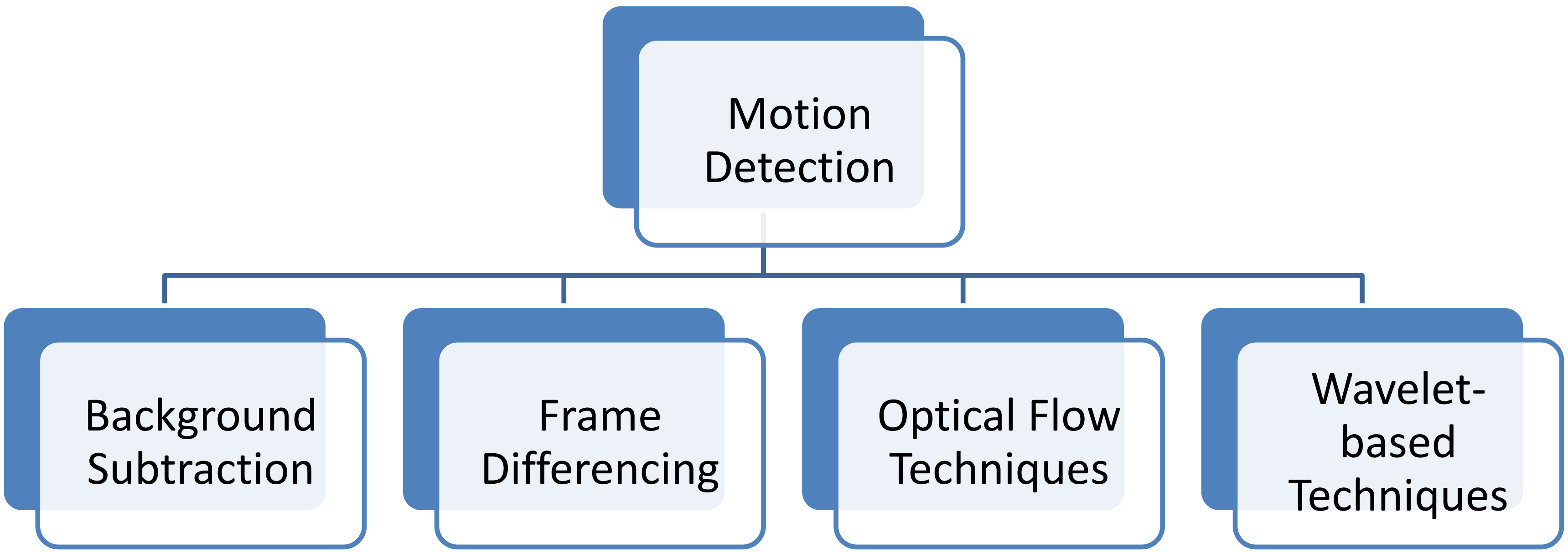
## 4- Region Splitting and Merging

- **Disadvantages:**
  - It may produce the blocky segments.
  - The blocky segment problem could be reduced by splitting in higher level, but the trade off is that the computation time will arise.

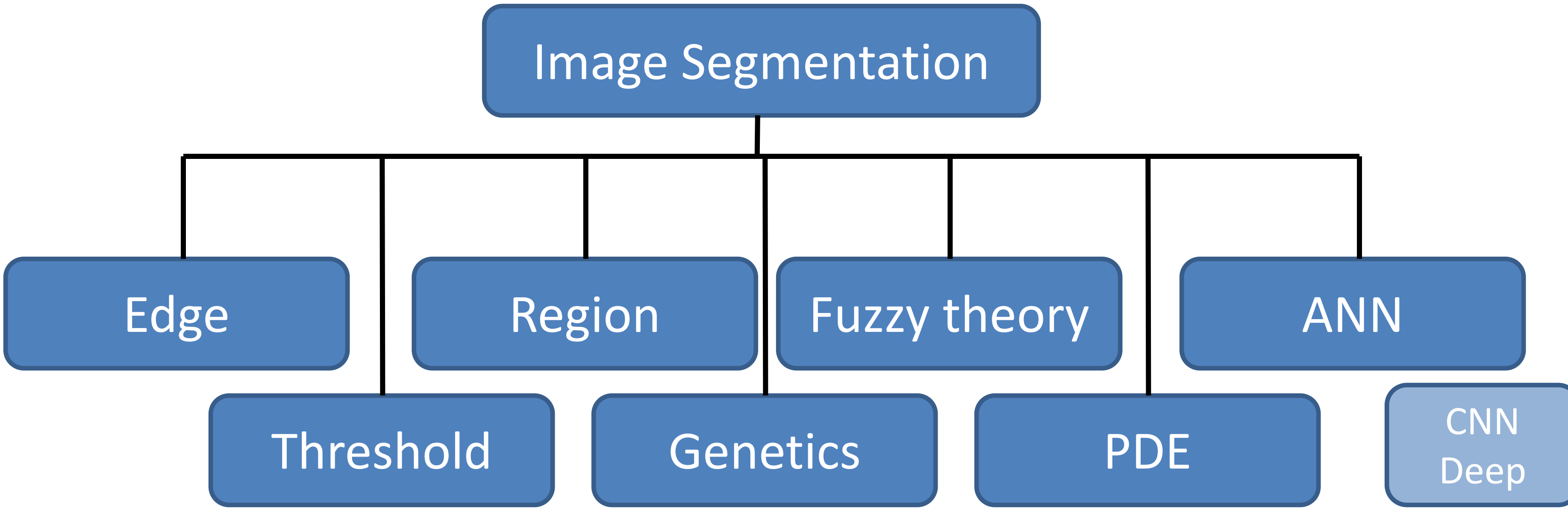
# The Use of Motion in Segmentation



# The Use of Motion in Segmentation



# Segmentation Methods (some)

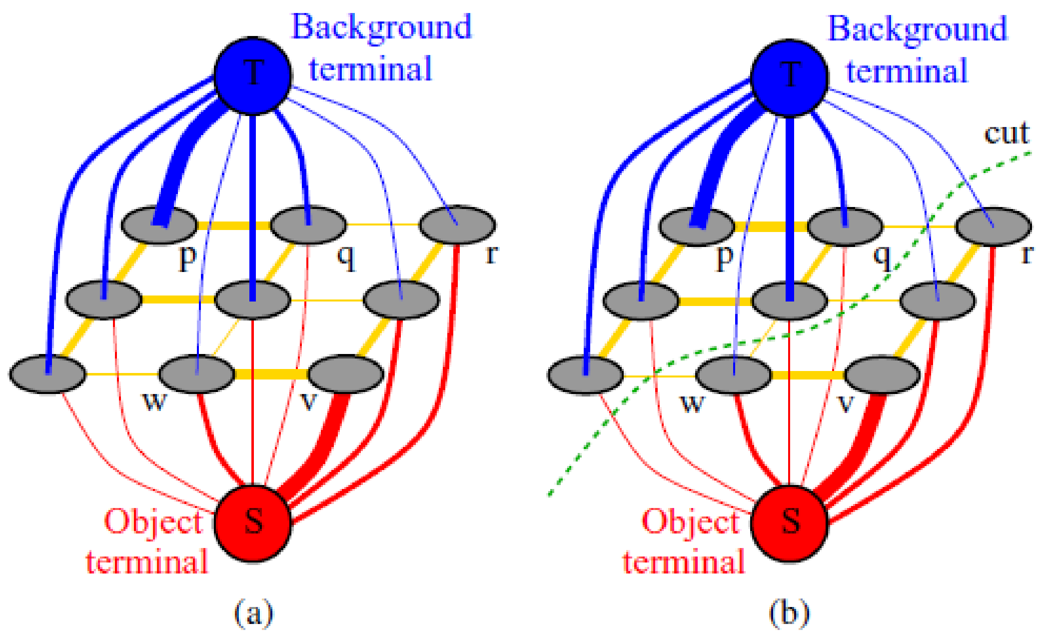



- Graph cuts
- Graph-based
- Color
- Active Contours
- Mean shift
- Matting
- Machine learning
- etc.

- Interactive
- Sports
- Medical images
- Surveillance
- Videos
- Infrared
- etc.

# Examples

Graph cuts for region segmentation

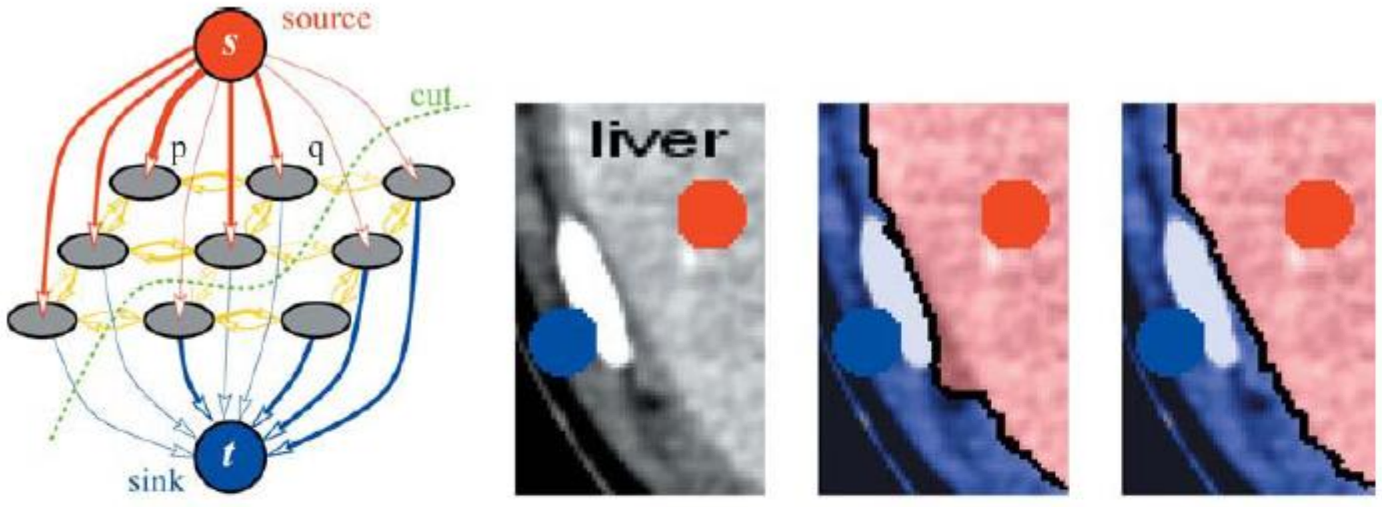



(a) the energy function is encoded as a maximum flow problem; (b) the minimum cut determines the region boundary

Richard Szeliski, Computer Vision Algorithms and Applications, Springer 2010. (Chapter 5)

# Examples

Segmentation with a directed graph cut



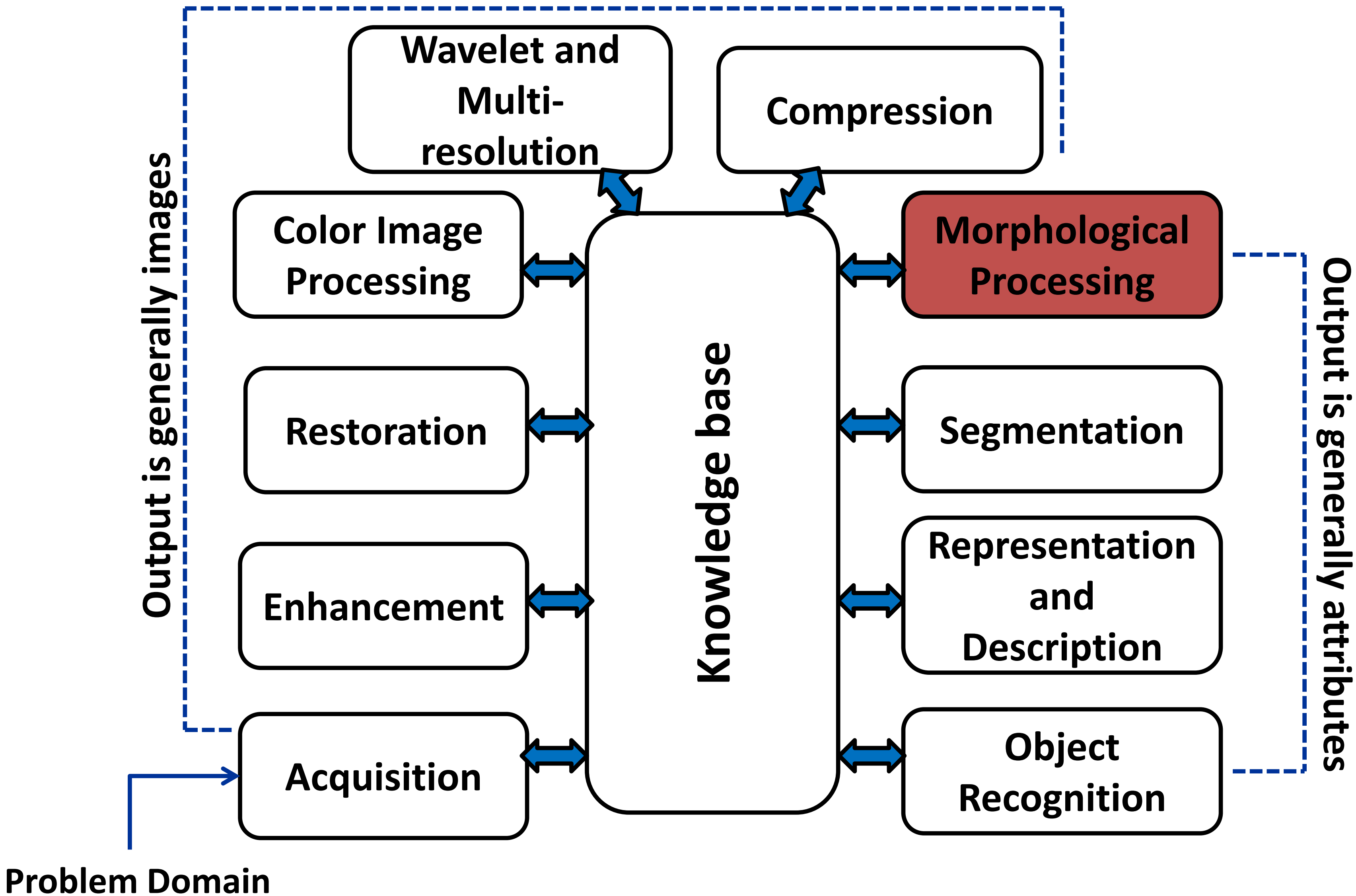

(a) directed graph (b) image (c) undir. result (d) dir. result

(a) directed graph; (b) image with seed points; (c) the undirected graph incorrectly continues the boundary along the bright object; (d) the directed graph correctly segments the light gray region from its darker surround.

Richard Szeliski, Computer Vision Algorithms and Applications, Springer 2010. (Chapter 5)

# Morphological Image Processing

# Fundamental Steps of DIP



# Contents

1. What is Morphology?
2. Fundamentals
3. Basic Morphological Operations
4. Compound Operations
5. Basic Morphological Algorithms

# What is Morphology?

***Mathematical morphology***

- **Morphological image processing describes a range of image processing techniques that deal with the shape of features in an image.**
- **Hence, used in image analysis based on shape.**
- **Nonlinear operations that are based on Minkowski's set theory.**

# What is Morphology? – (cont.)

- **Used to extract image components that are useful in the representation and description of regions shapes.**
- **Can be used to remove imperfections in the segmented image and provide information on the form and structure of the image.**
- **Pre- or post-processing.**

# What is Morphology? – (cont.)

## Uses of Image Morphology

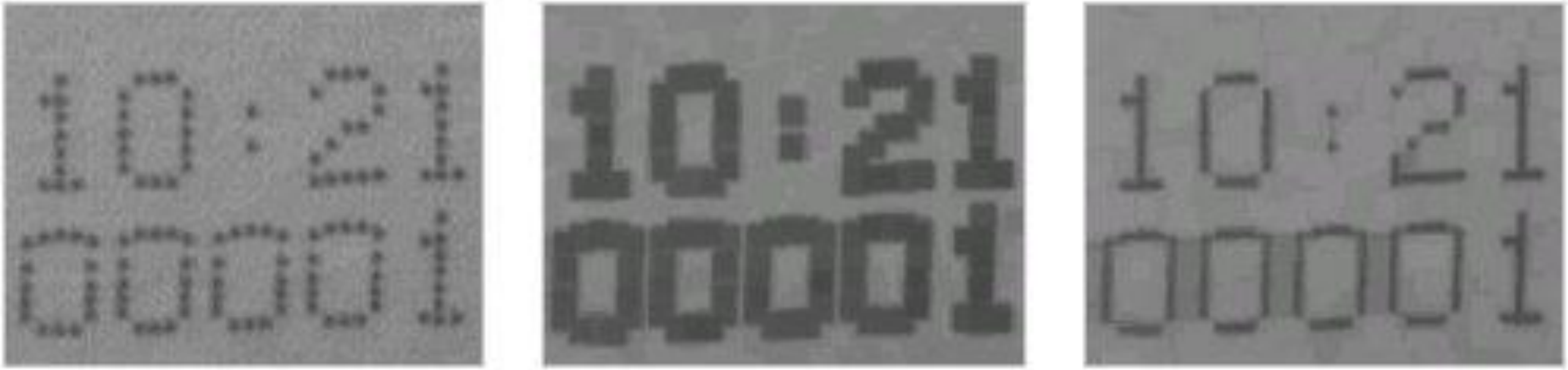
- image enhancement
- image segmentation
- image restoration
- edge detection
- texture analysis
- particle analysis
- feature generation
- skeletonization
- shape analysis
- image compression
- component analysis
- curve filling
- general thinning
- feature detection
- noise reduction

# What is Morphology? – (cont.)

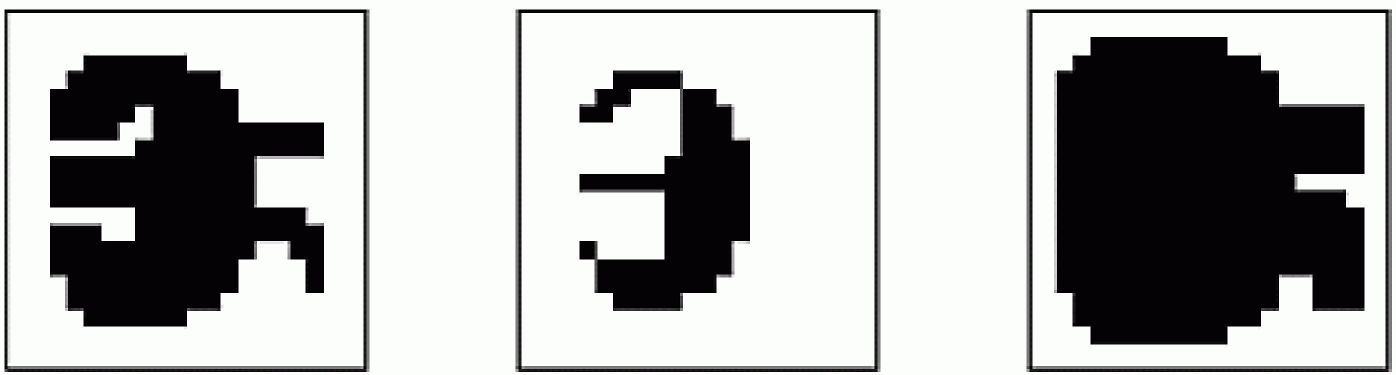
## Examples

Image after segmentation

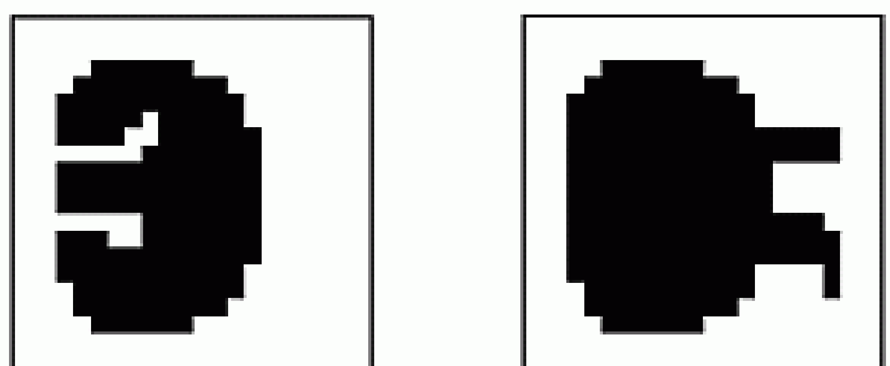
Image after segmentation and morphological processing

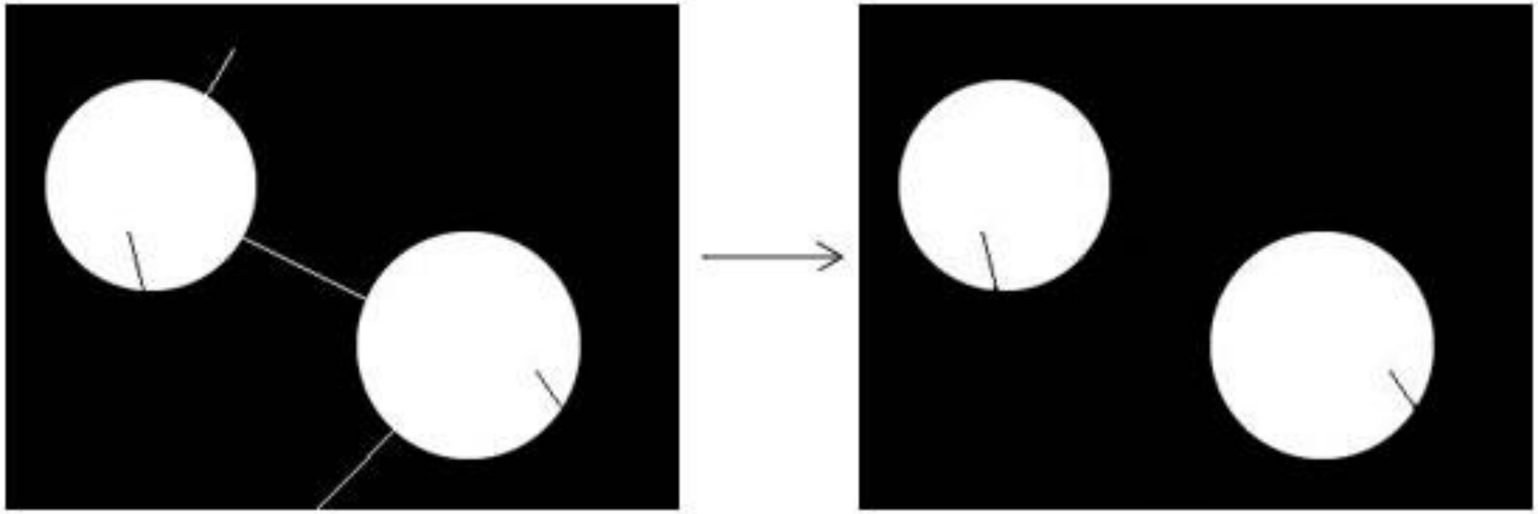



a. Original b. Erosion c. Dilation

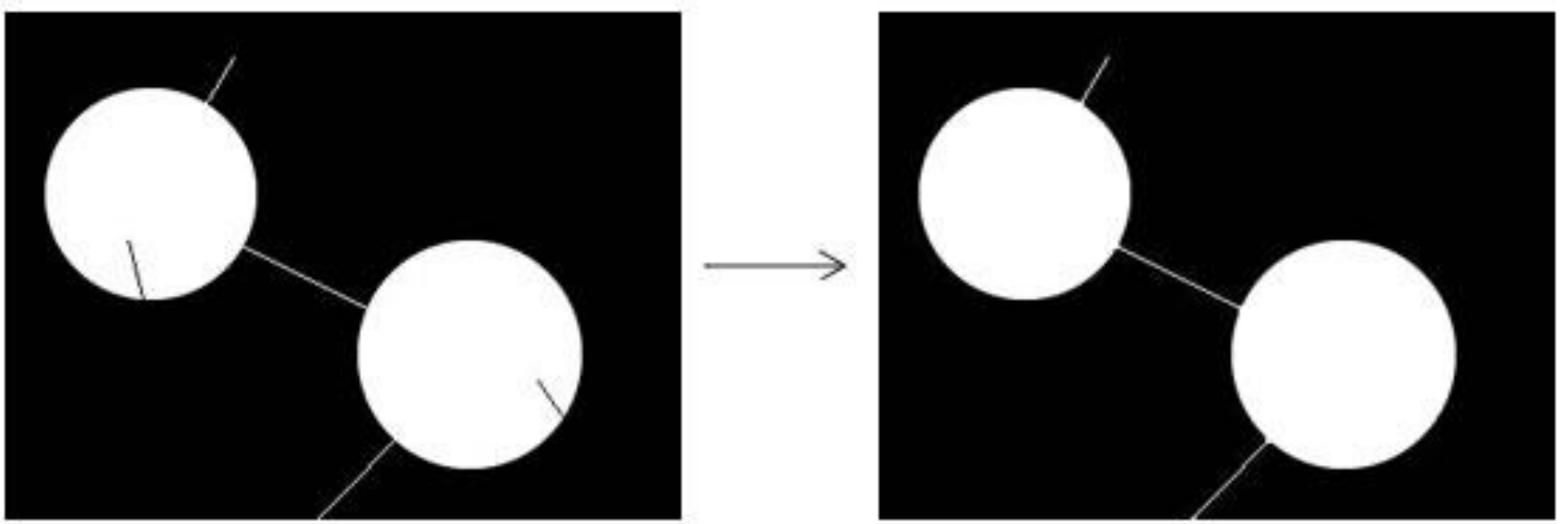


d. Opening e. Closing

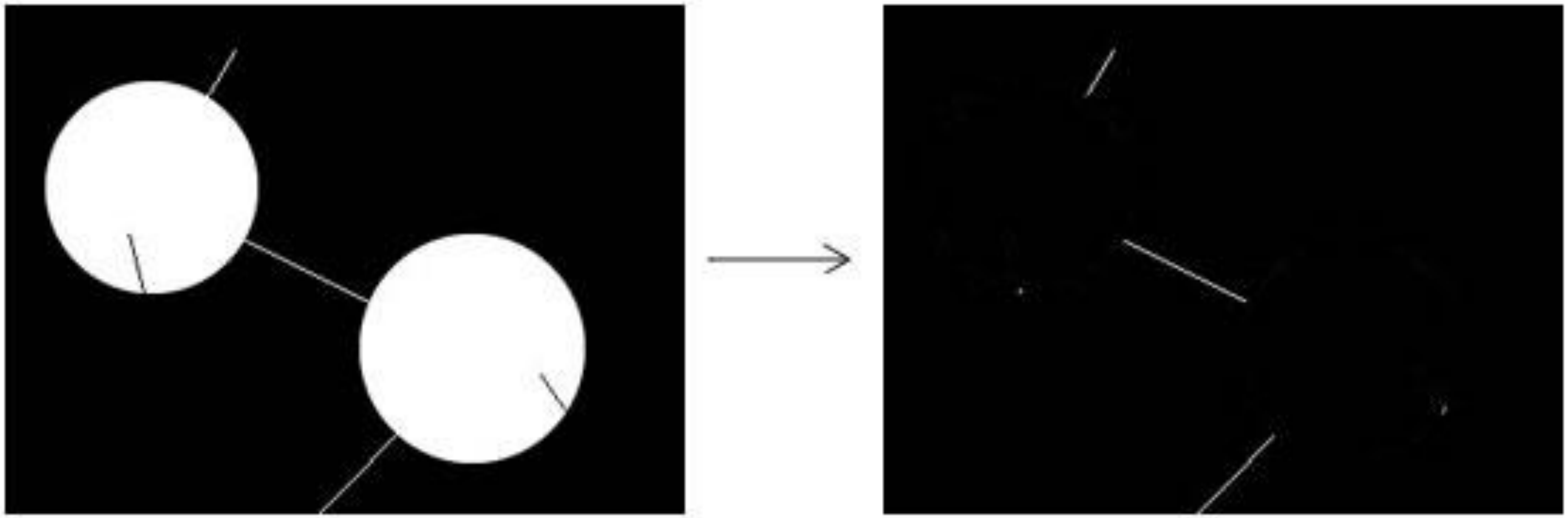




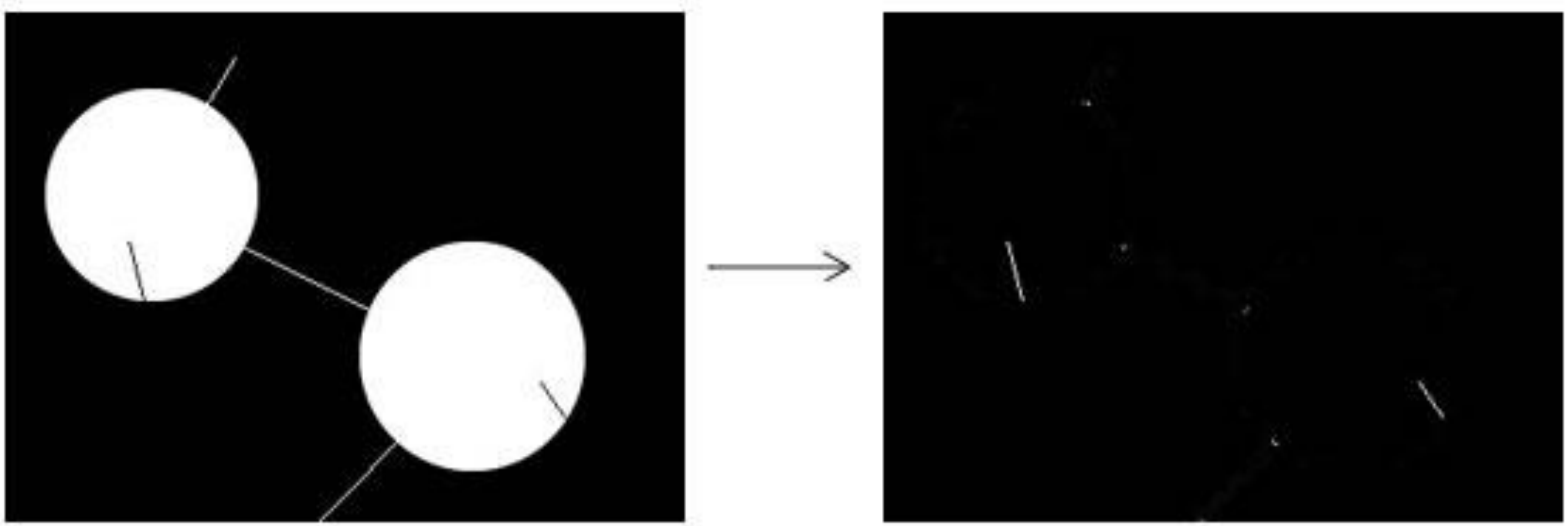
Morphological Opening



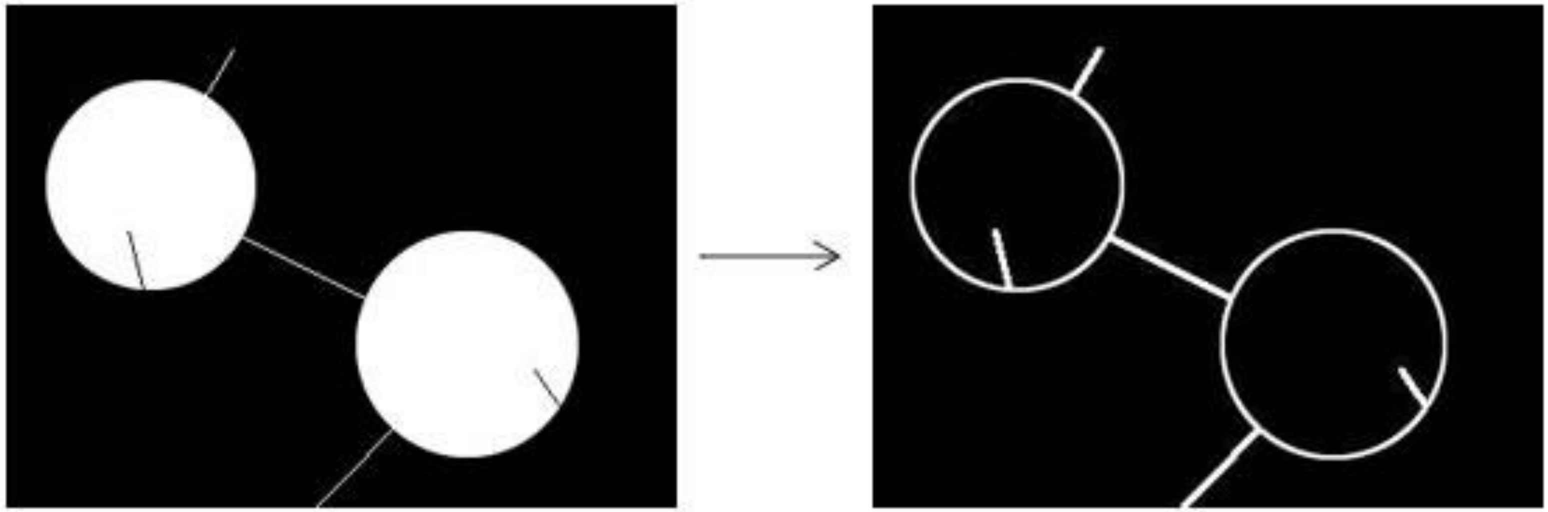
Morphological Closing



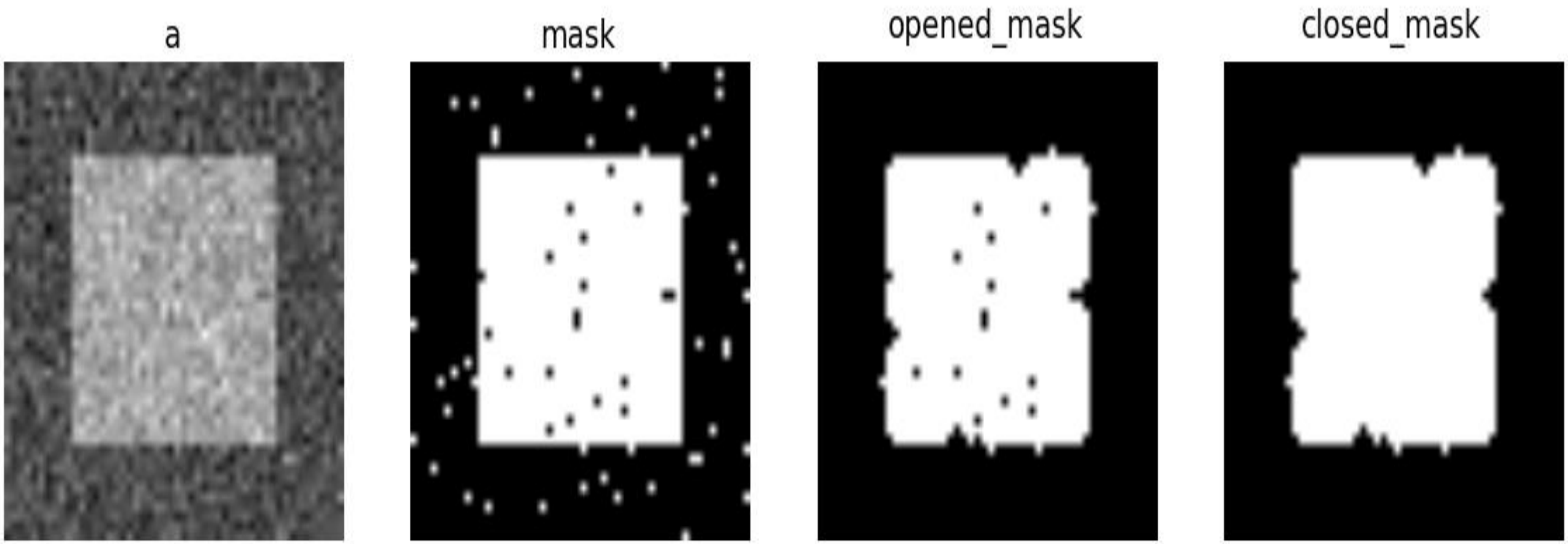
Morphological Top hat



Morphological Black hat



Morphological Gradient



# Fundamentals

## Mathematical Morphology

- **In binary images, sets represents groups of pixels at specific locations.**
- **Each pixel is represented in the 2D integer space $Z^2$ by vector $(x,y)$ of white or black.**
- **Morphological operations change the "shape" of the objects of interest according to set theory concepts, e.g. union, intersection, and complement.**

# Fundamentals – (cont.)

## Bi-level versus Grey-scale Morphology

- **Binary morphological operations are *typically* applied to bi-level images (foreground vs background pixels).**

- **Grey-scale operations are applied in the same manner but to grey-scale images with each pixel represented in the 3D integer space $Z^2$ by vector $(x,y,g)$, where g is the gray level of pixel coordinate $(x,y)$.**

# Fundamentals – (cont.)

- Fundamentally, morphological image processing is very much like spatial filtering.

- A structuring element is moved across every pixel in the original image to give a pixel in a new processed image.

- The value (black or white) of this new pixel depends on the operation performed.

# Fundamentals – (cont.)

## Structuring Element

- **A SE can be any size and any shape.**
- **For simplicity we use rectangular structuring elements with the origin at the middle pixel.**

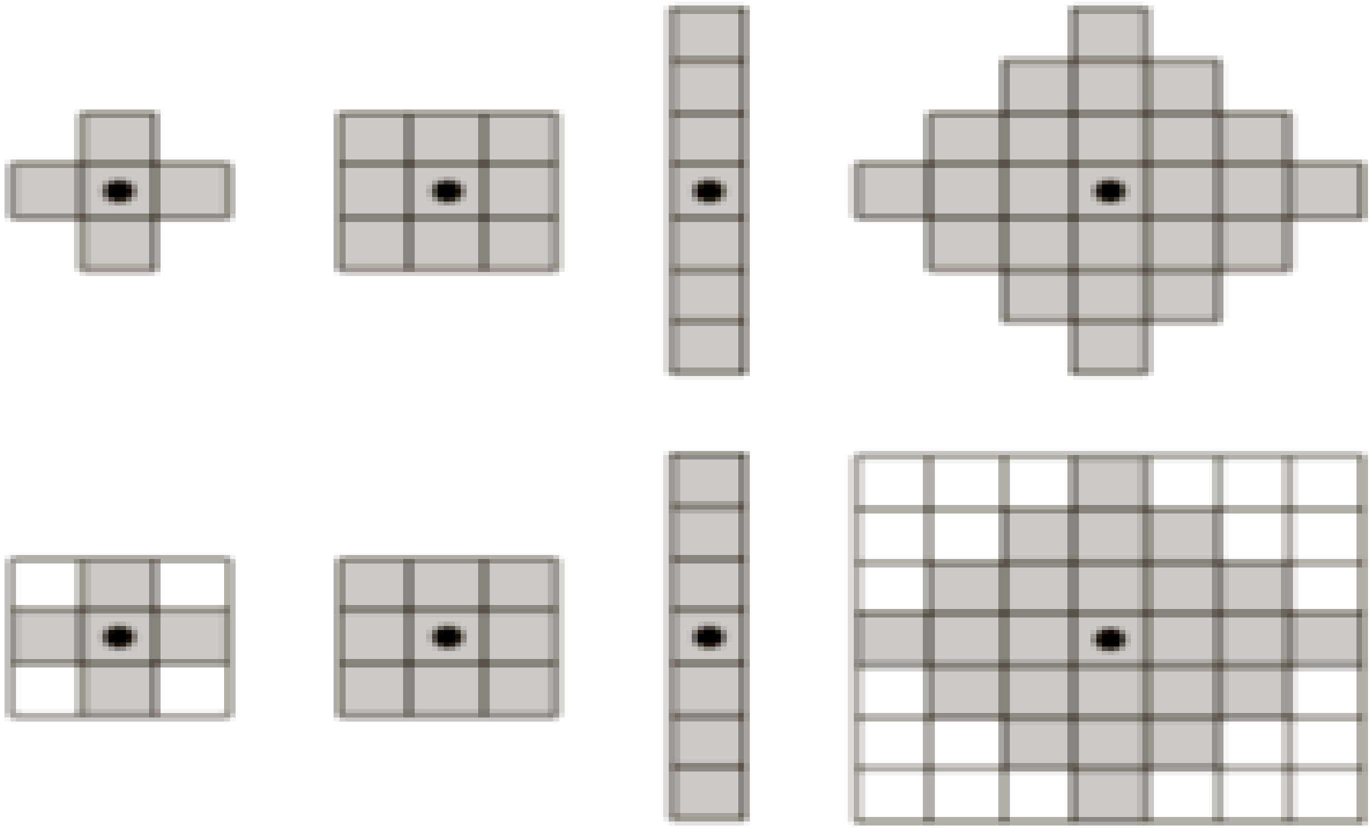
| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | **1** | 1 |
| 1 | 1 | 1 |

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | **1** | 1 |
| 0 | 1 | 0 |

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | **1** | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |

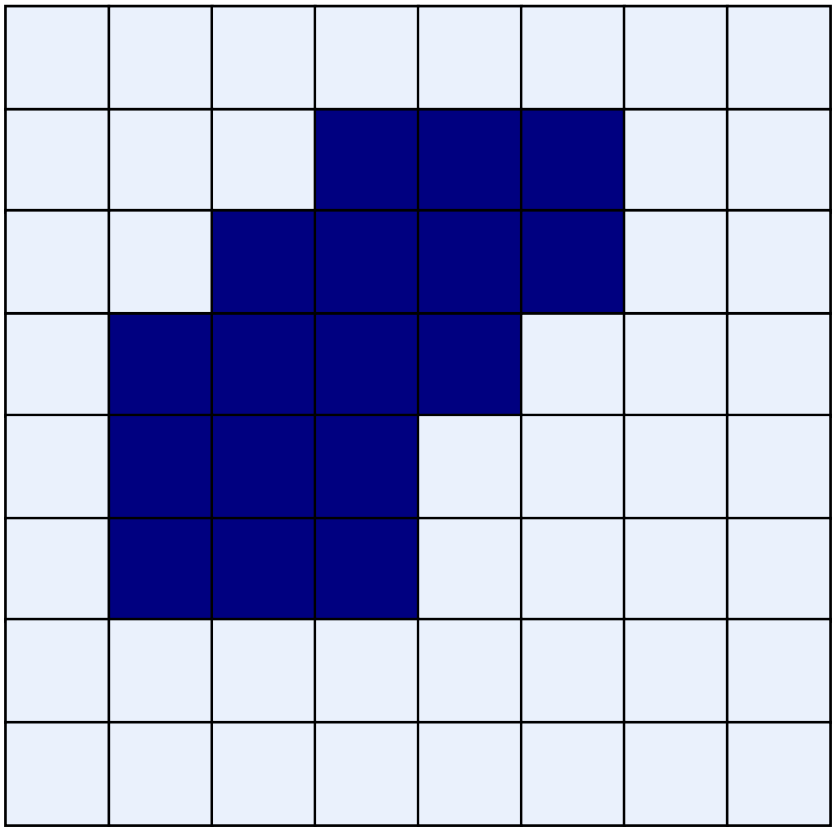
# Fundamentals – (cont.)

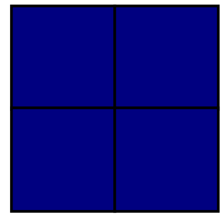
## Structuring Element



# Fundamentals – (cont.)

## Structuring Element – Hits and Fits





SE

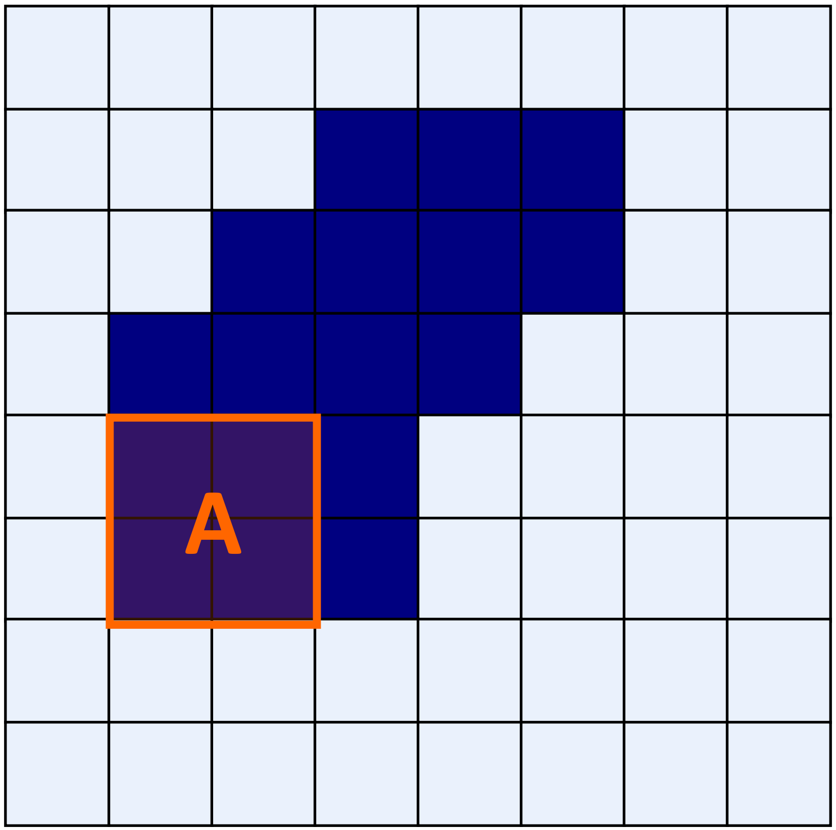
**<u>Fit:</u> All *on pixels* in the structuring element cover *on pixels* in the image**

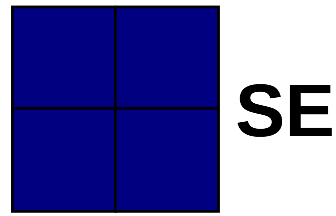
**<u>Hit:</u> Any *on pixel* in the structuring element covers an *on pixel* in the image**

**All morphological processing operations are based on these simple ideas.**

# Fundamentals – (cont.)

## Structuring Element – Hits and Fits



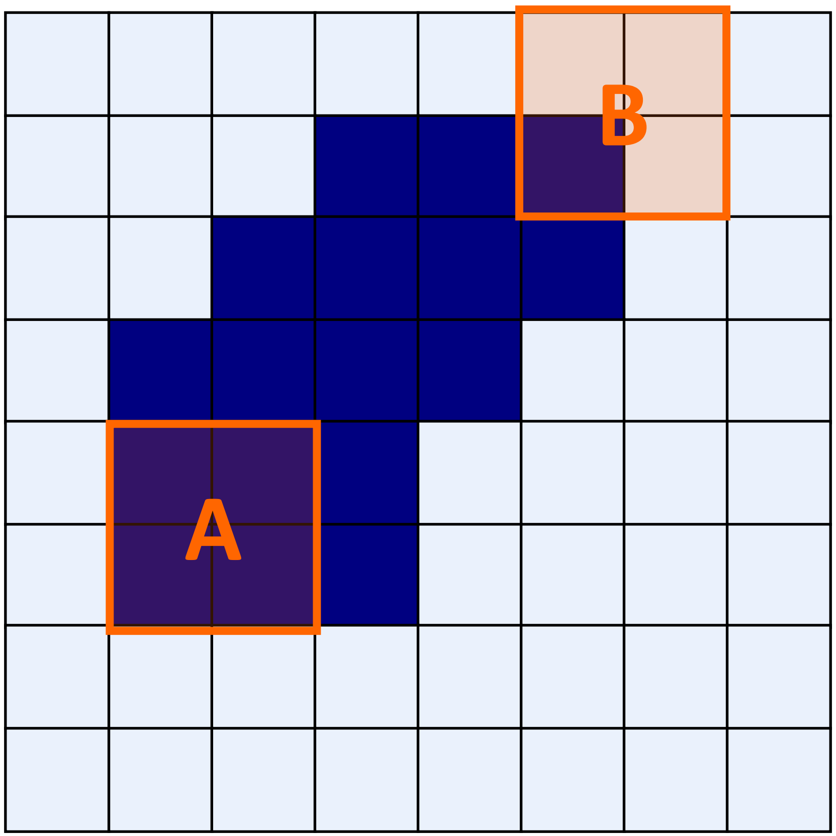

**<u>Fit:</u> All *on pixels* in the structuring element cover *on pixels* in the image**

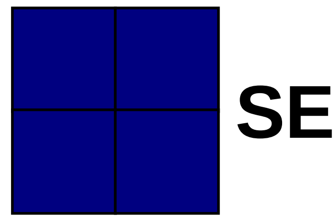
**<u>Hit:</u> Any *on pixel* in the structuring element covers an *on pixel* in the image**

**All morphological processing operations are based on these simple ideas.**

# Fundamentals – (cont.)

## Structuring Element – Hits and Fits



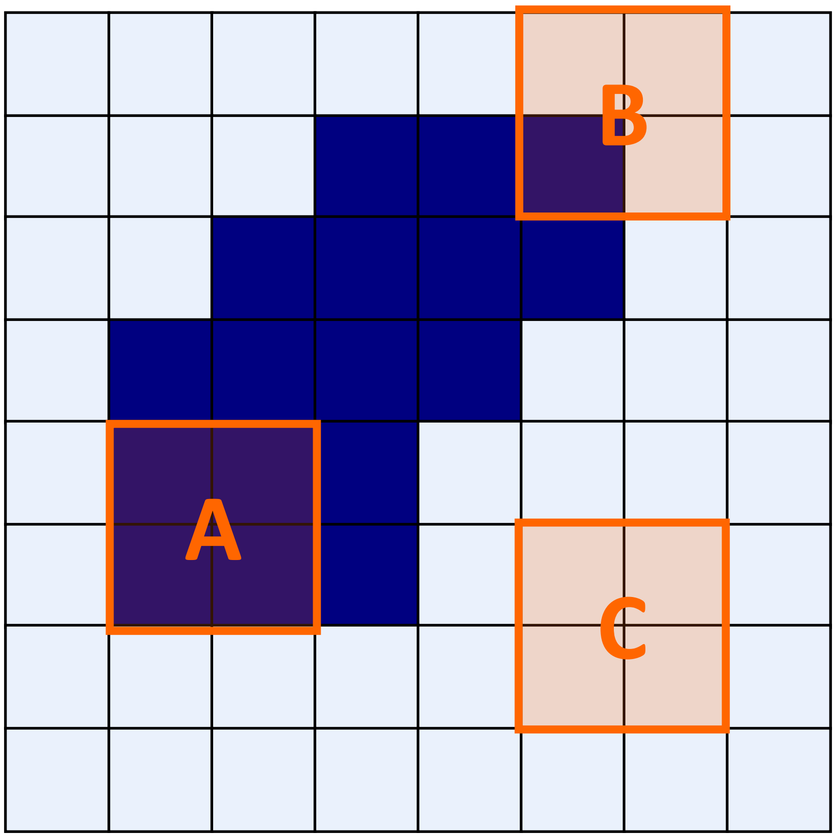

**Fit:** All *on pixels* in the structuring element cover *on pixels* in the image

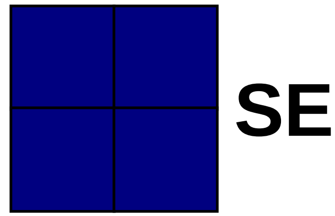
**Hit:** Any *on pixel* in the structuring element covers an *on pixel* in the image

**All morphological processing operations are based on these simple ideas.**

# Fundamentals – (cont.)

## Structuring Element – Hits and Fits




**Fit:** All *on pixels* in the structuring element cover *on pixels* in the image

**Hit:** Any *on pixel* in the structuring element covers an *on pixel* in the image

**All morphological processing operations are based on these simple ideas.**

# Fundamentals – (cont.)

## Structuring Element – Hits and Fits

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

**Structuring Element 1**

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

**Structuring Element 2**

# Fundamentals – (cont.)

## Structuring Element – Hits and Fits

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 A | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

**Structuring Element 1**

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

**Structuring Element 2**

# Fundamentals – (cont.)

## Structuring Element – Hits and Fits

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 B | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 A | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

**Structuring Element 1**

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

**Structuring Element 2**

# Fundamentals – (cont.)

## Structuring Element – Hits and Fits

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

B

C

A

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

**Structuring Element 1**

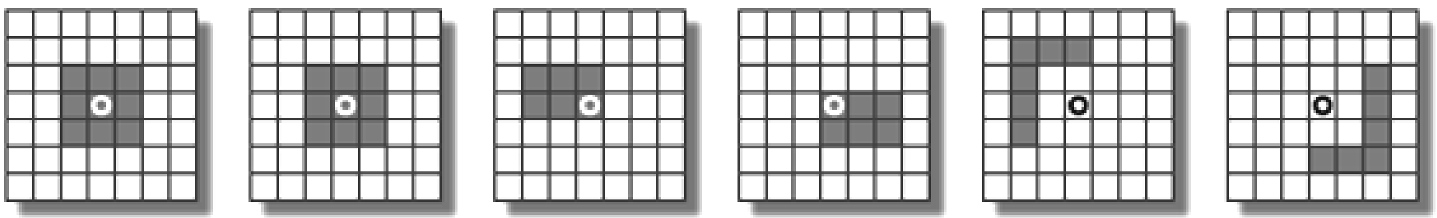
| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

**Structuring Element 2**

# Fundamentals – (cont.)

## Structuring Element – Reflected/Translated

- **The reflection of set $B$, denoted by $\hat{B}$, is defined as:** $\hat{B} = \{z \mid z = -b, \ \text{for } b \in B\}$



$\mathrm{SE}_1 \quad \hat{\mathrm{S}}\mathrm{E}_1 \quad \mathrm{SE}_2 \quad \hat{\mathrm{S}}\mathrm{E}_2 \quad \mathrm{SE}_3 \quad \hat{\mathrm{S}}\mathrm{E}_3$

**The translation of set $B$, by point $z$, denoted by $(B)_z$, is defined as:**

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

# Basic Morphological Operations

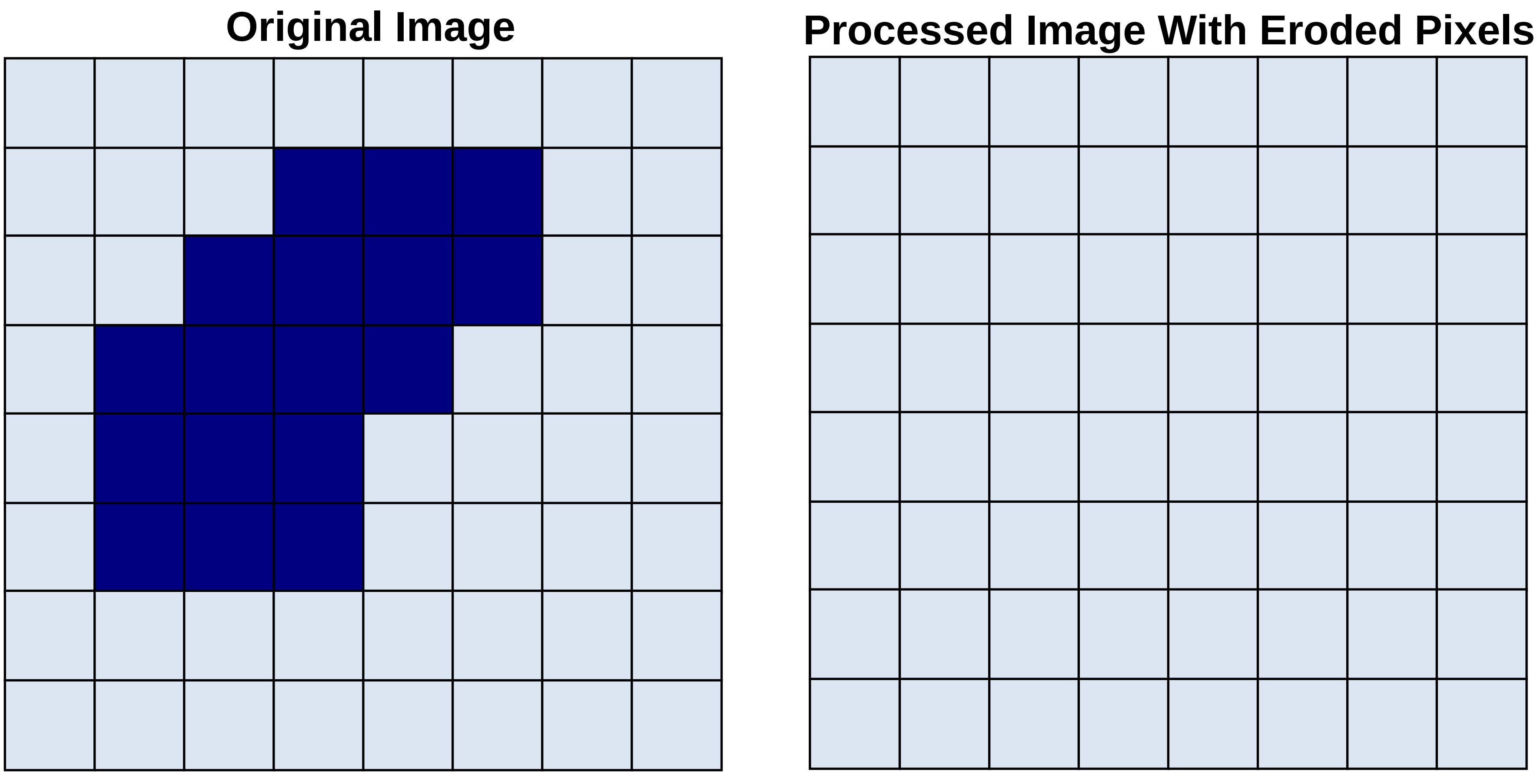
## 1- Erosion

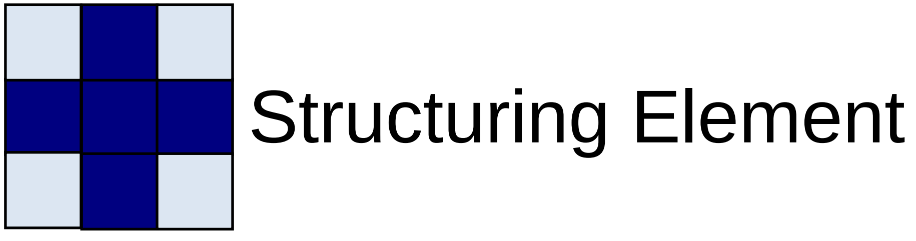
- **Erosion of image $X$ by structuring element $SE$ is given by $X \ominus SE$.**
- **The structuring element $SE$ is positioned with its origin at $(x, y)$ and the new pixel value is determined using the rule:**

$$g(x,y) = \begin{cases} 1 \text{ if } SE \text{ fits } X \\ 0 \text{ otherwise} \end{cases}$$

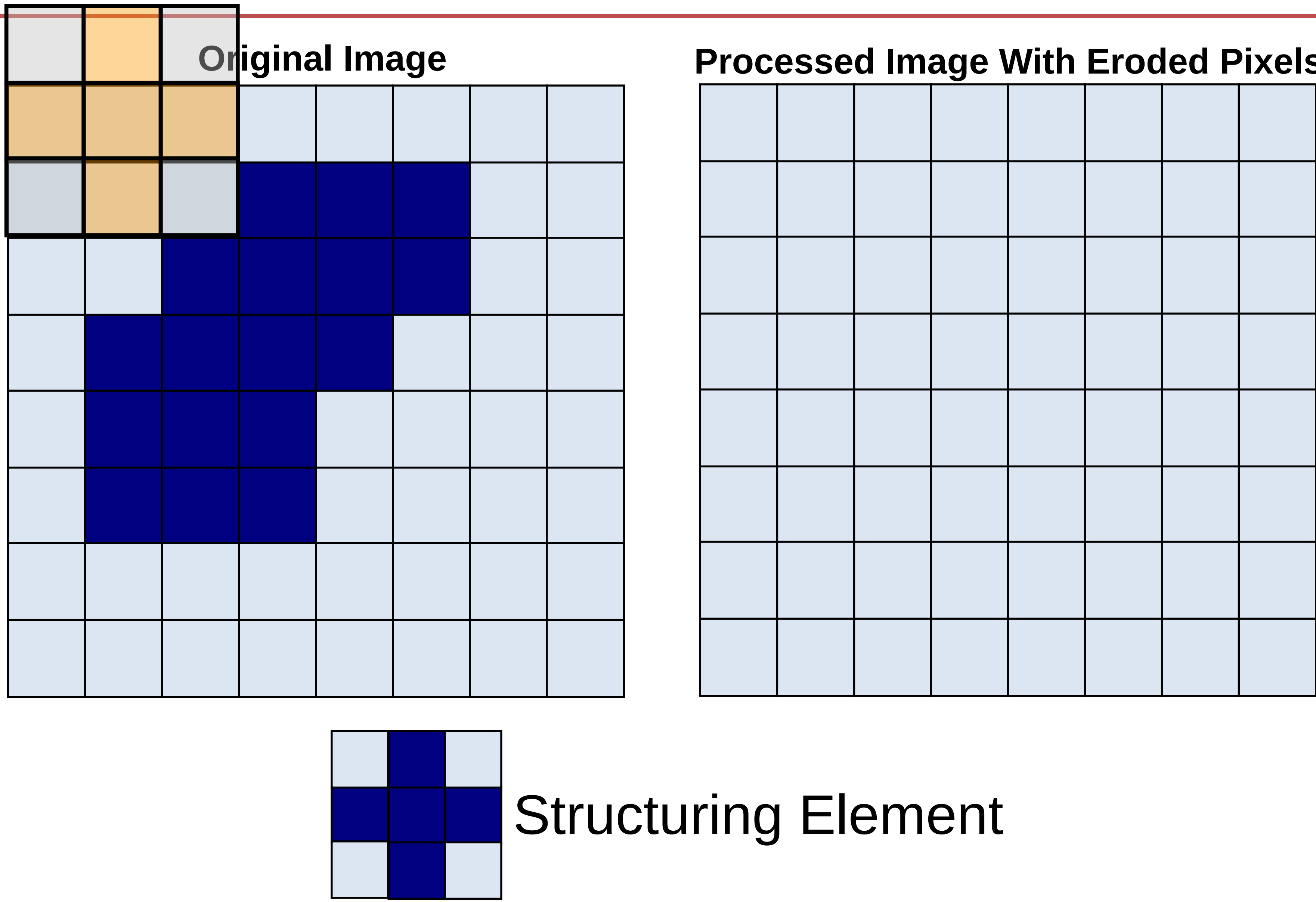
- **Or in set language:** $X \ominus SE = \{z \mid (SE)_z \subseteq X\}$

# Erosion

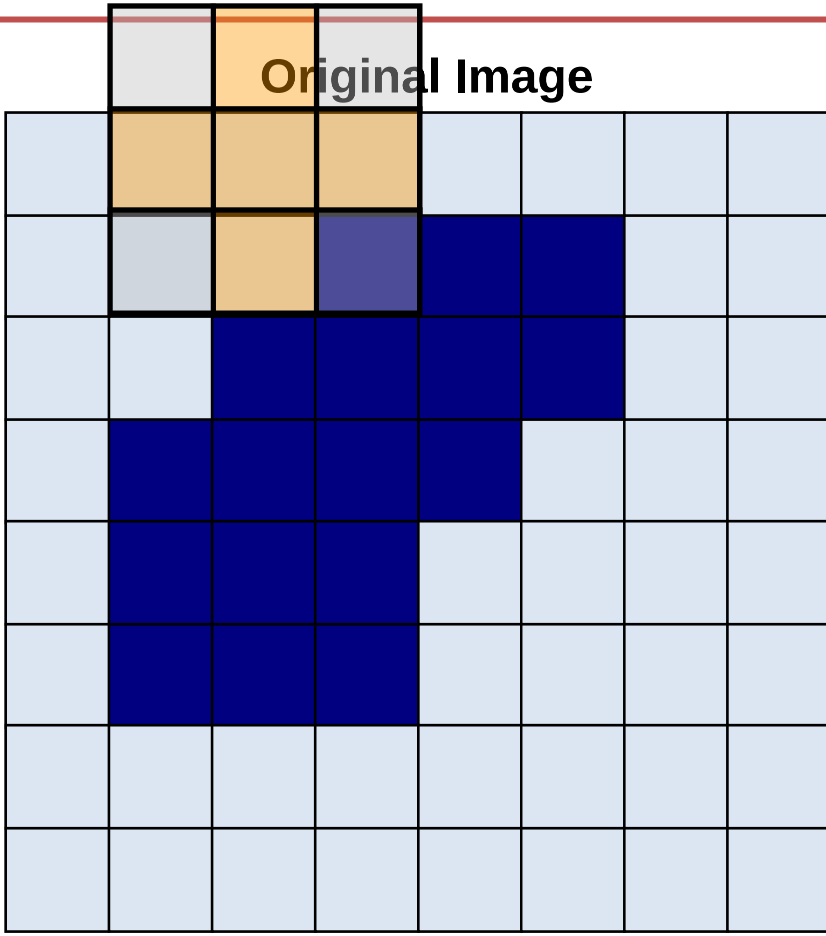

Original Image
Processed Image With Eroded Pixels

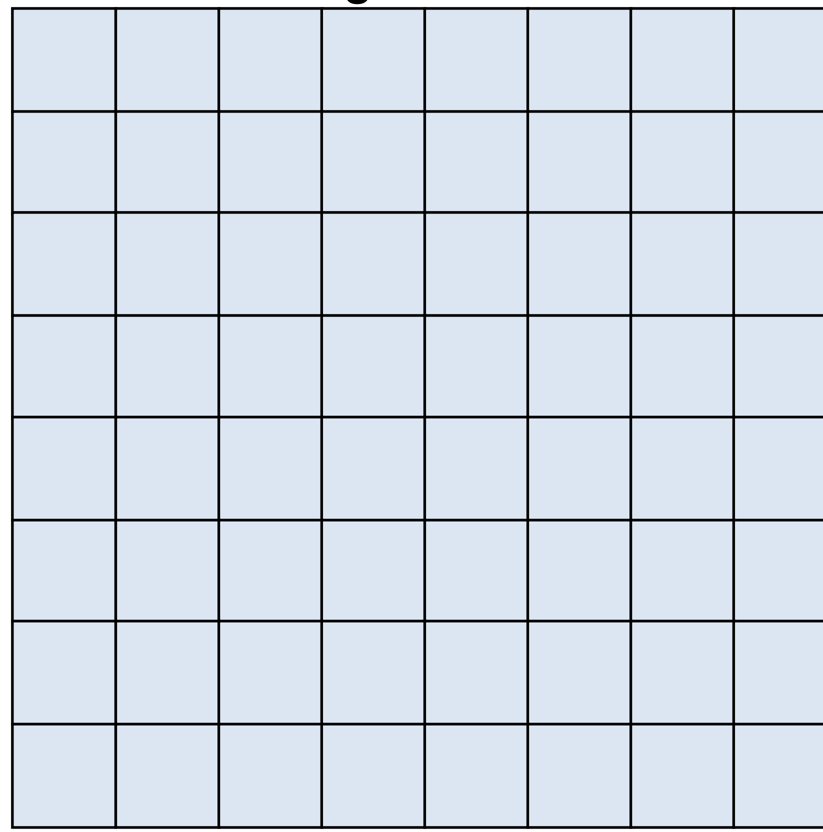

Structuring Element

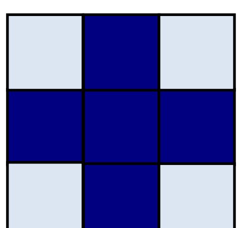
# Erosion – (cont.)



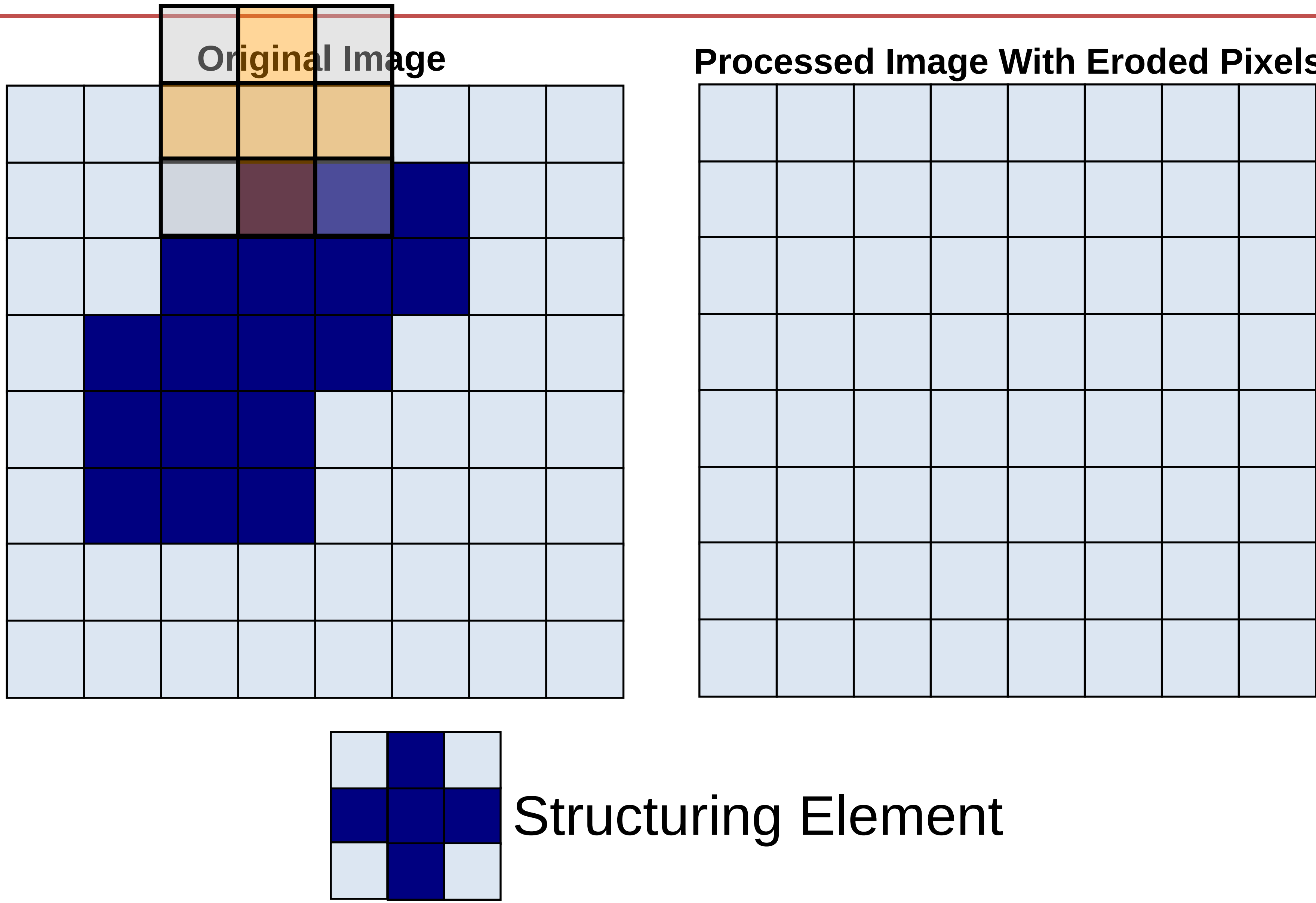
# Erosion – (cont.)



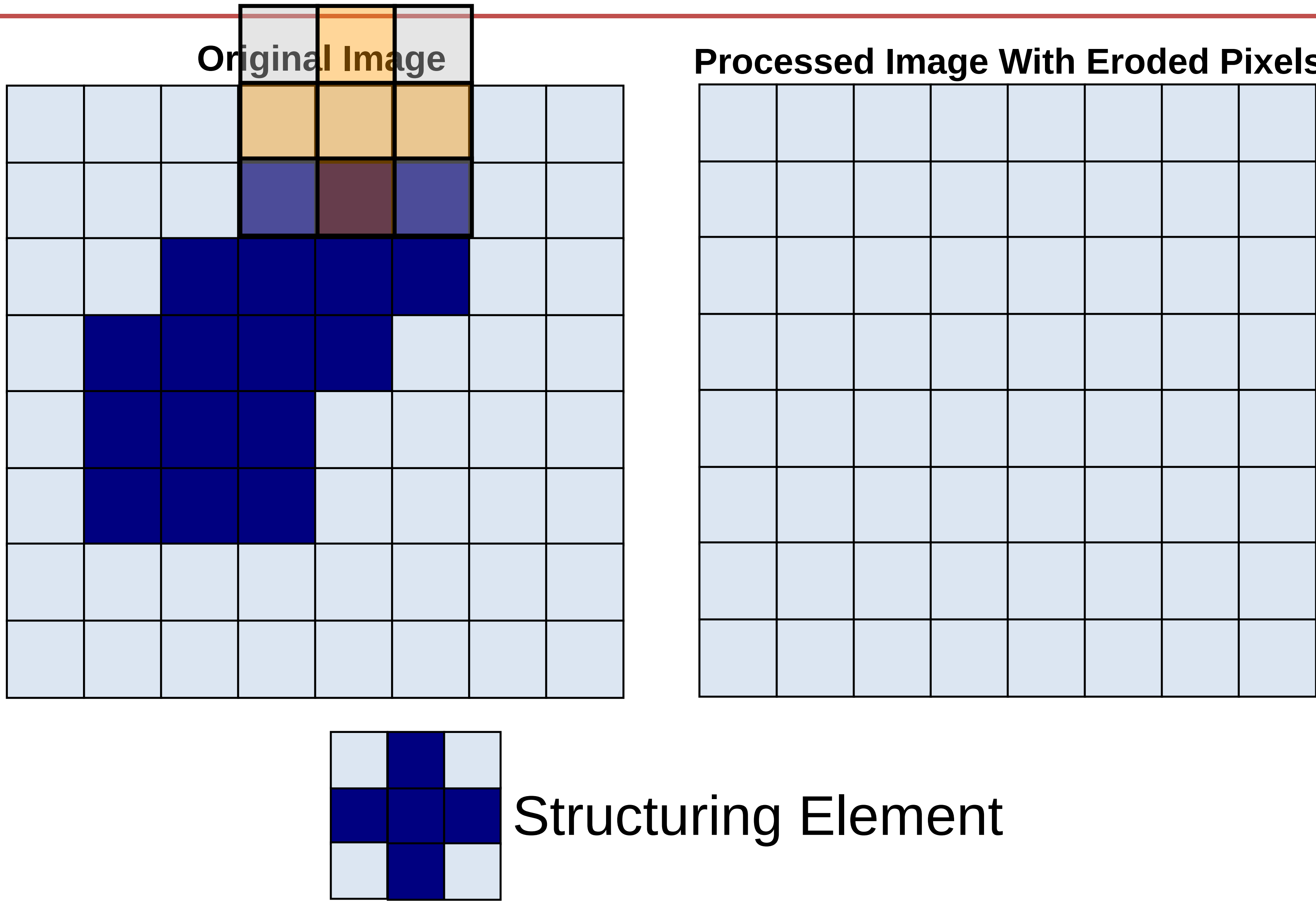



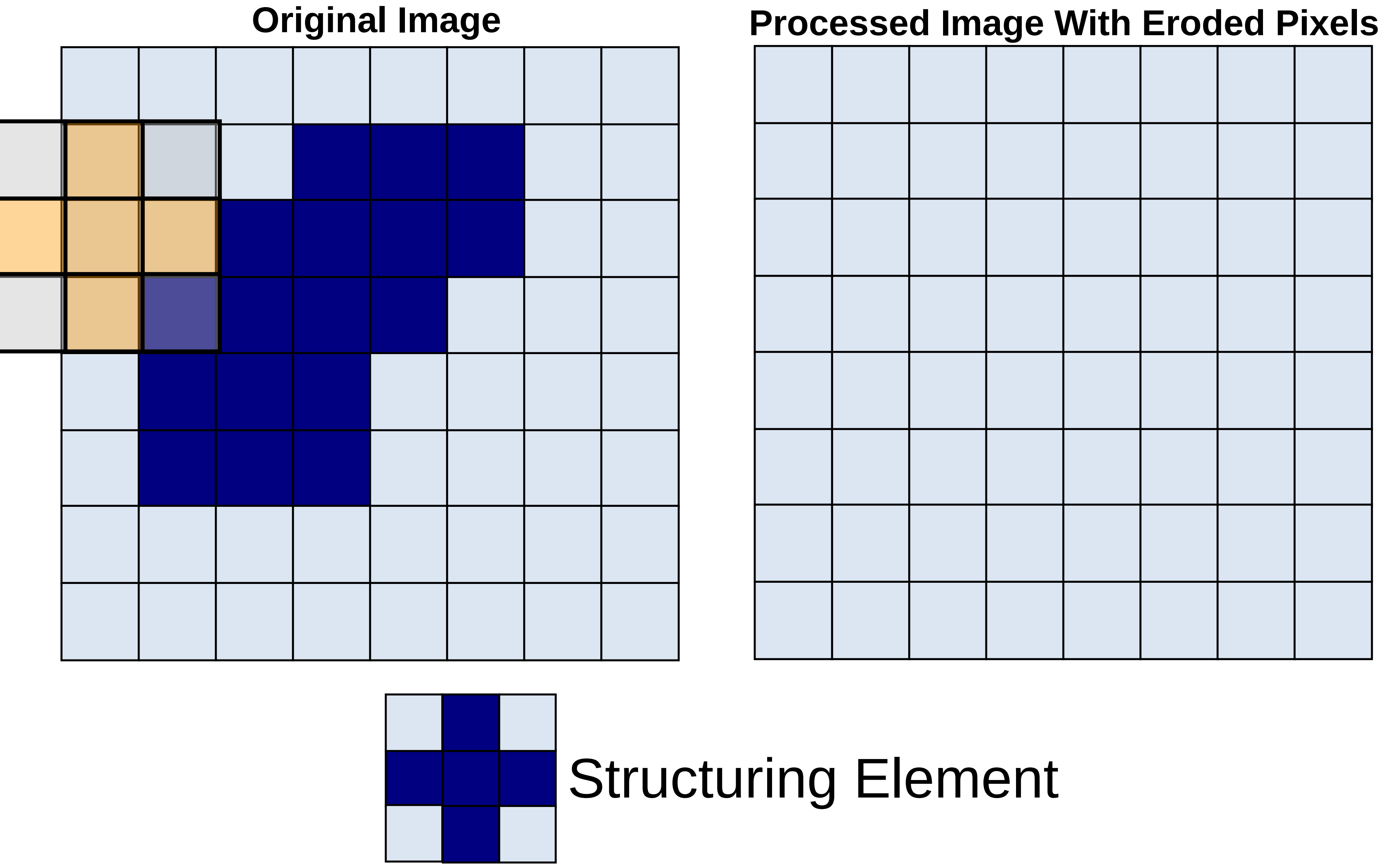
# Erosion – (cont.)



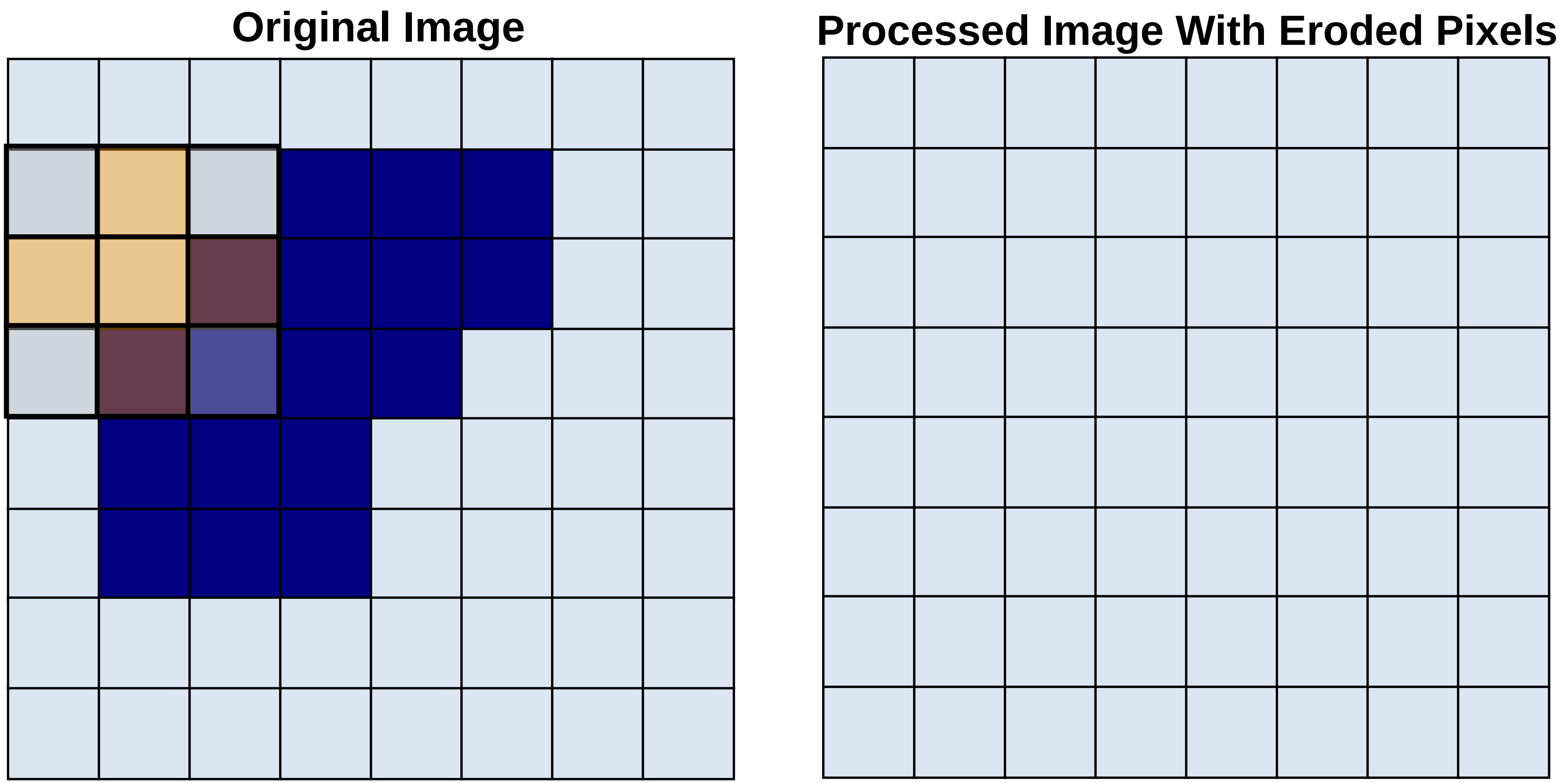
# Erosion – (cont.)

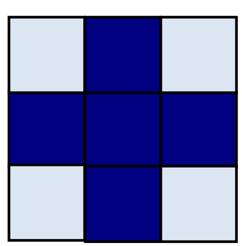


# Erosion – (cont.)



# Erosion – (cont.)

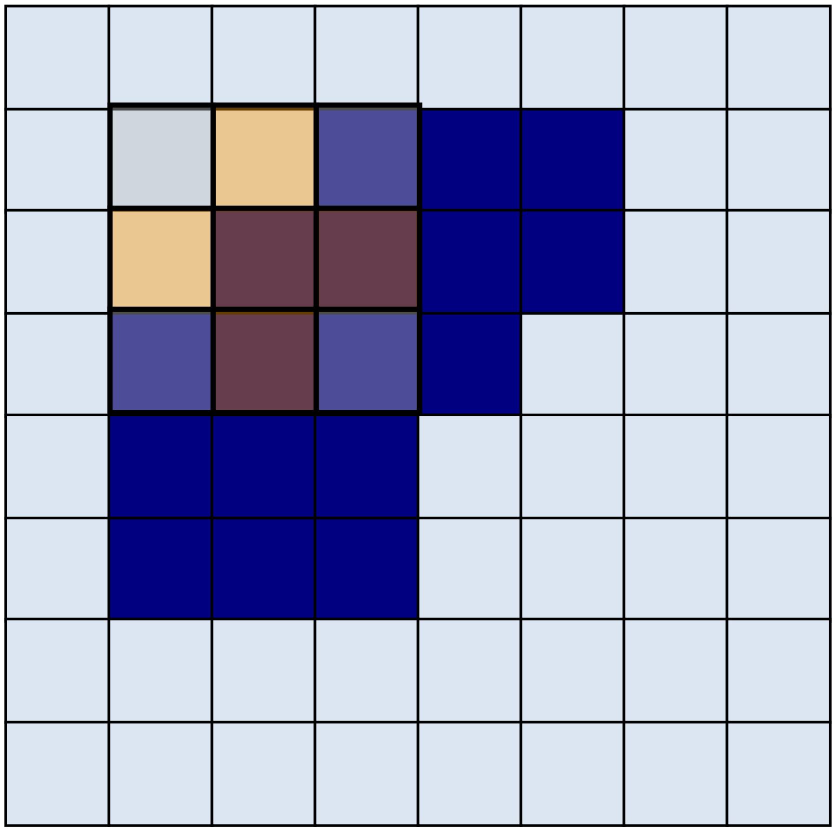


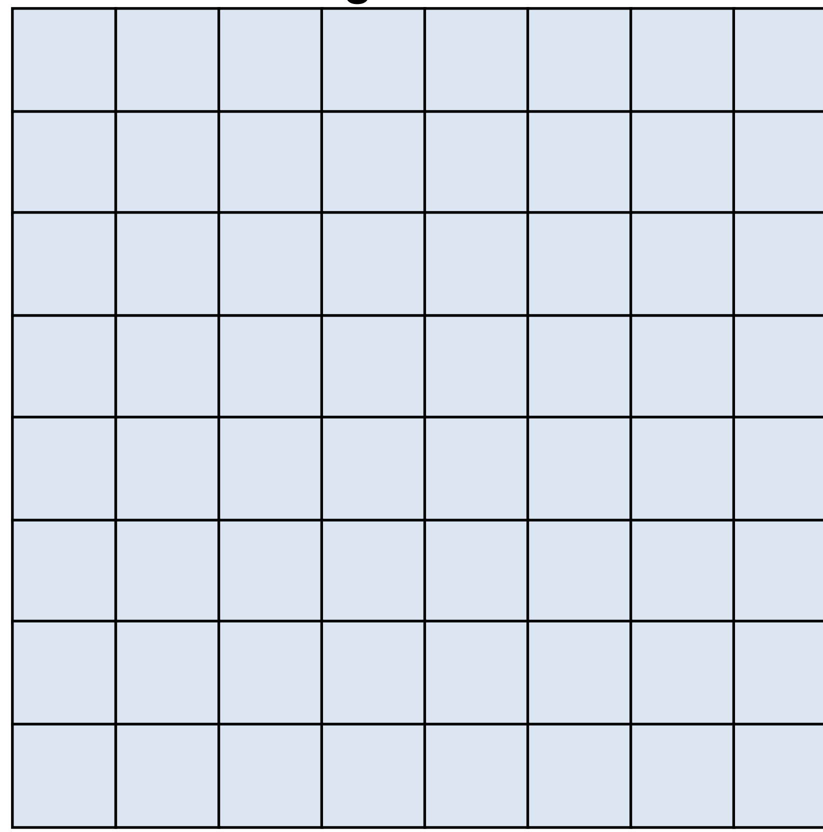
Structuring Element

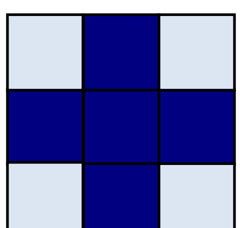
# Erosion – (cont.)

**Original Image**



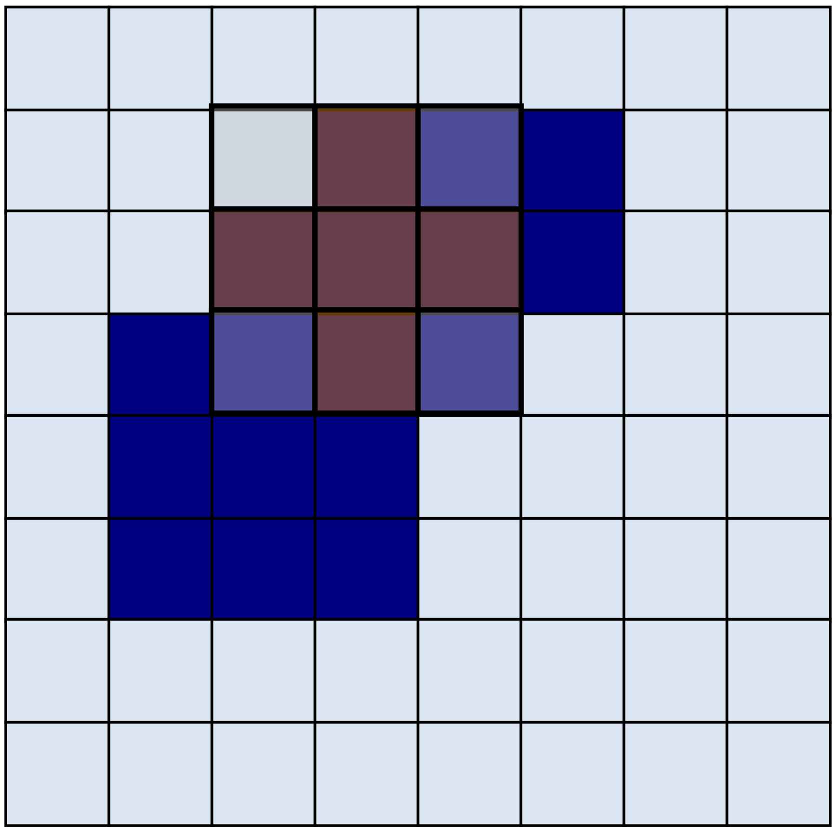
**Processed Image With Eroded Pixels**

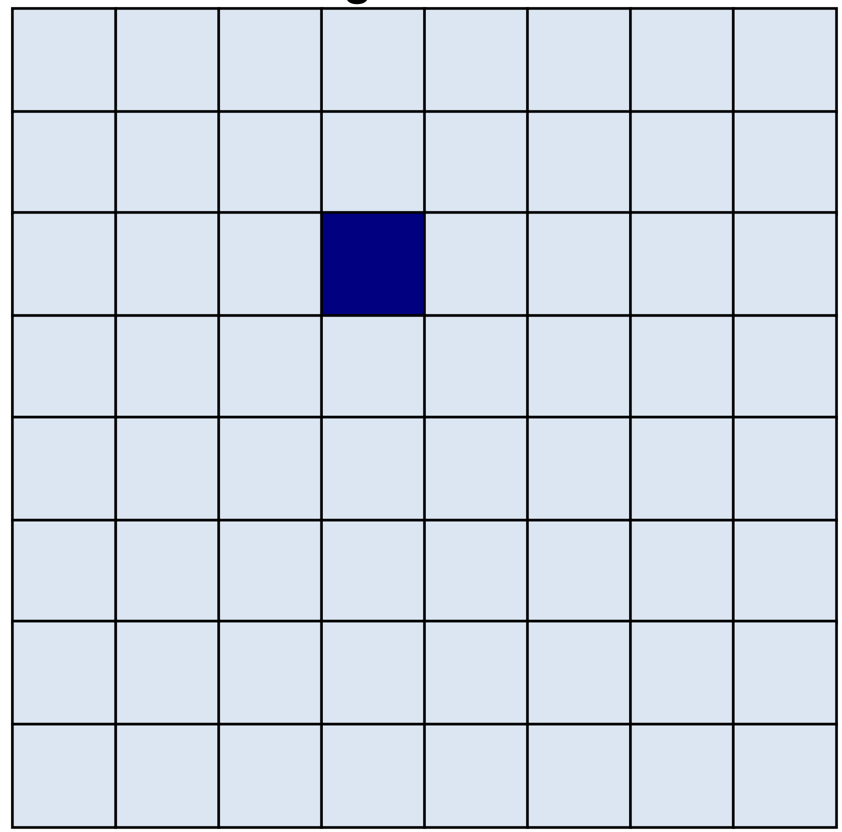


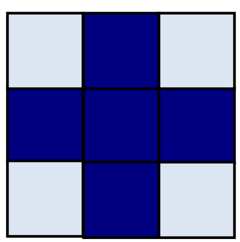


Structuring Element

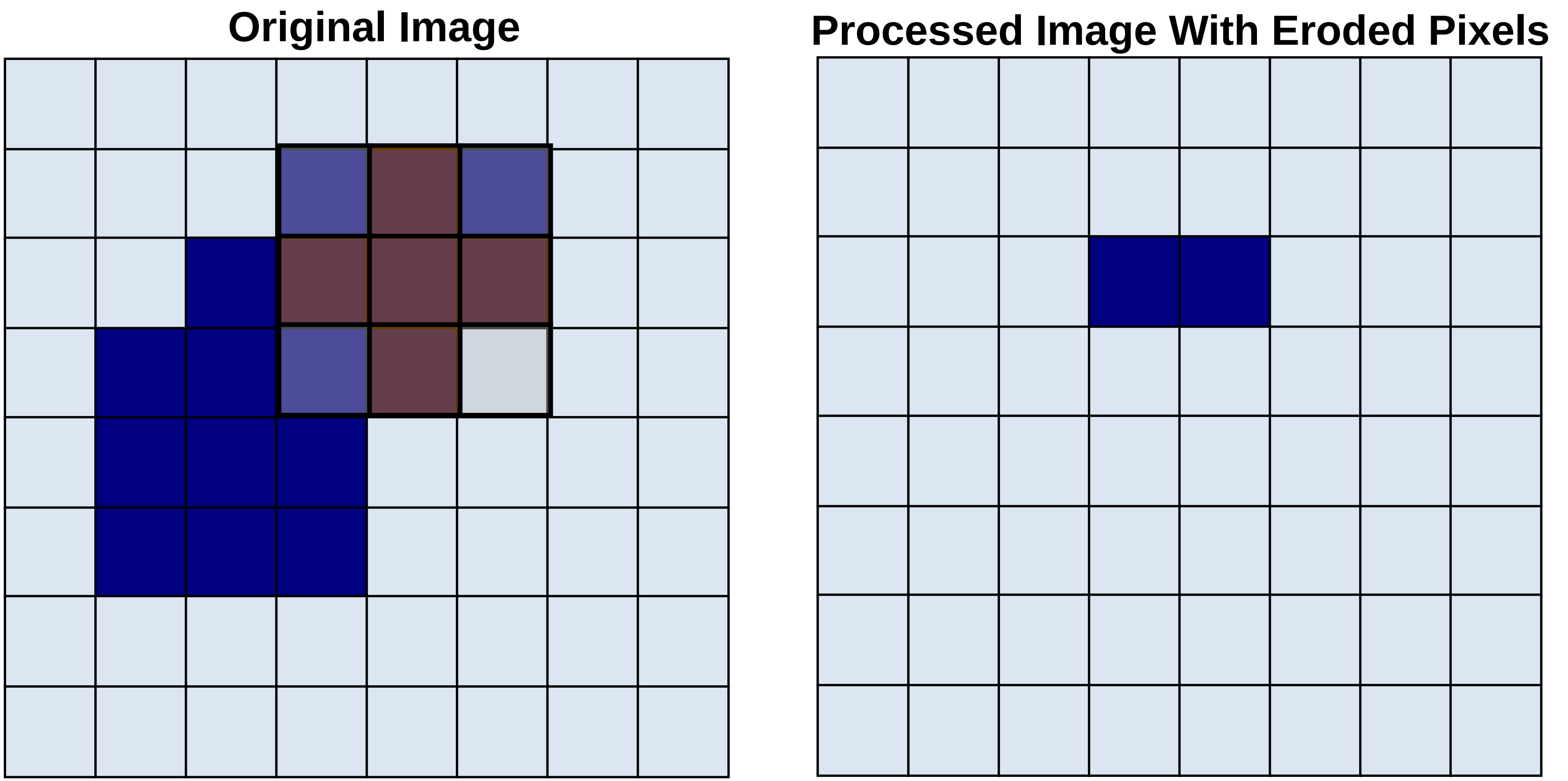
# Erosion – (cont.)

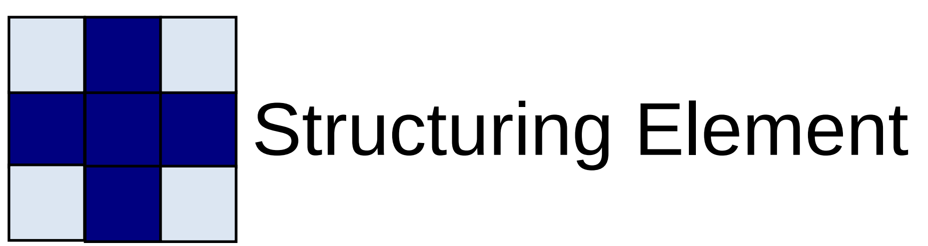


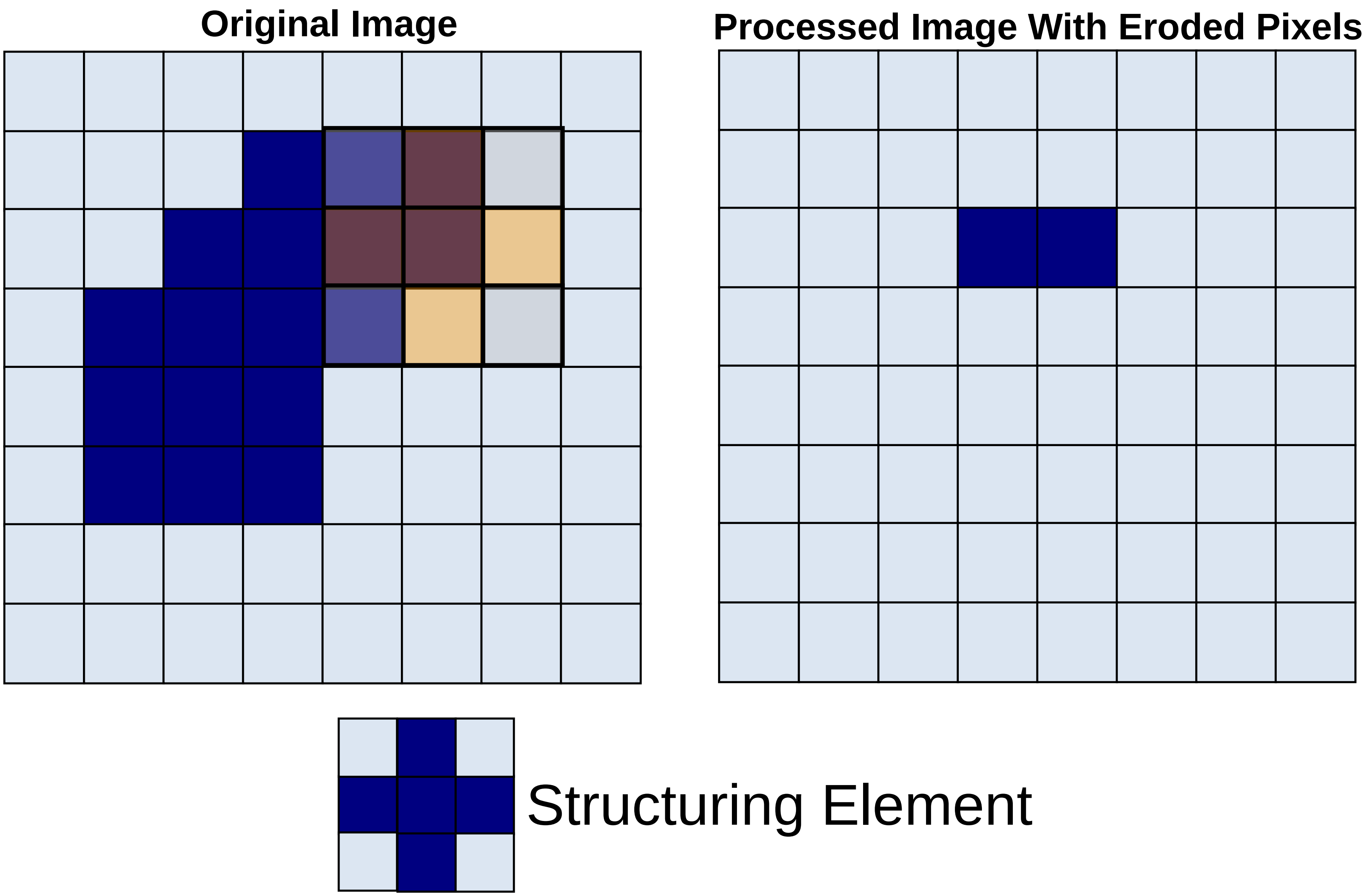



# Erosion – (cont.)

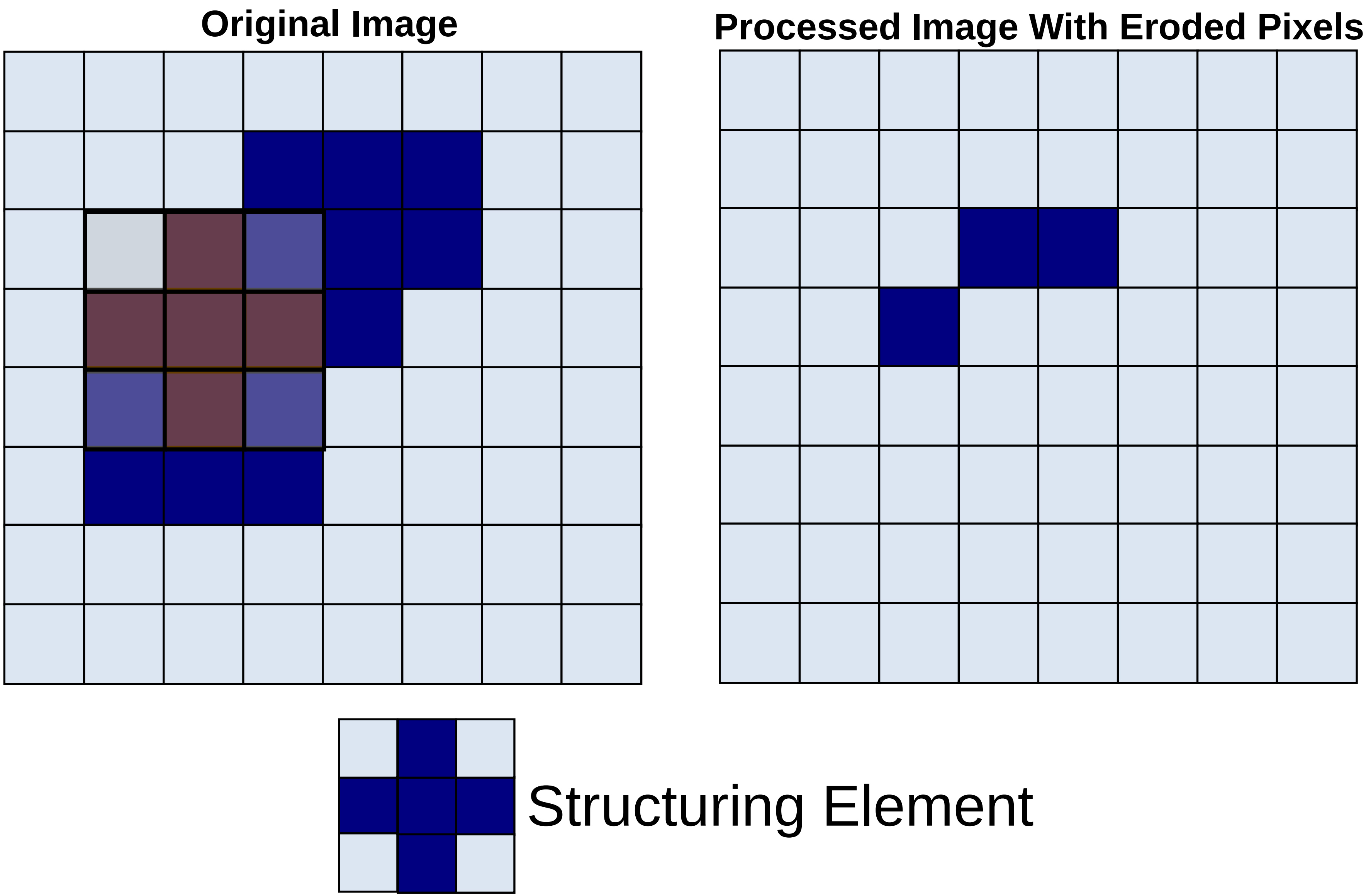


# Erosion – (cont.)

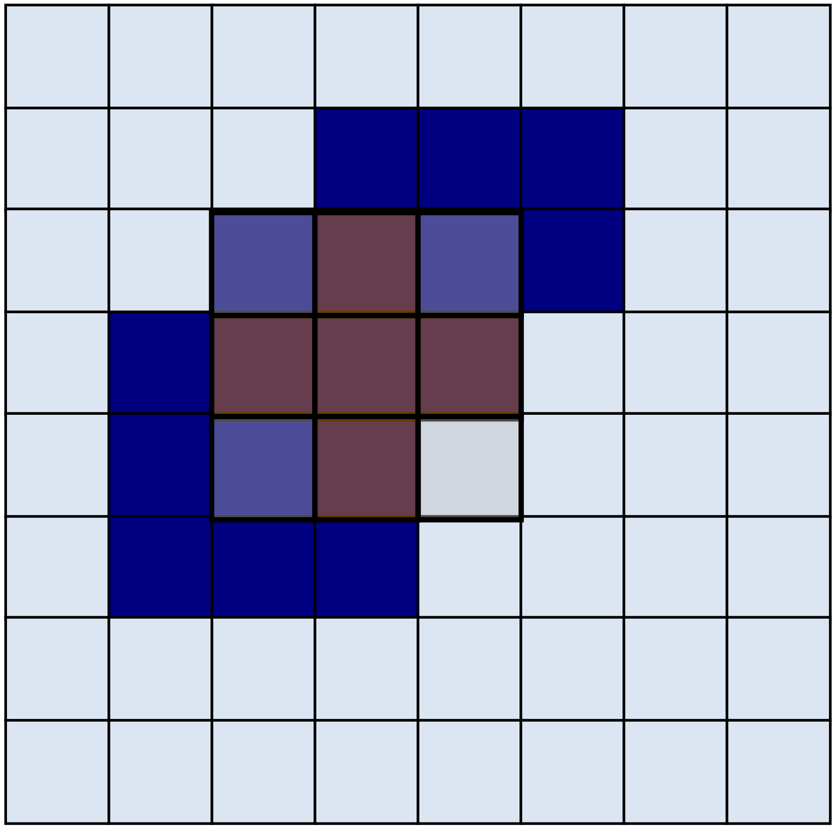


# Erosion – (cont.)

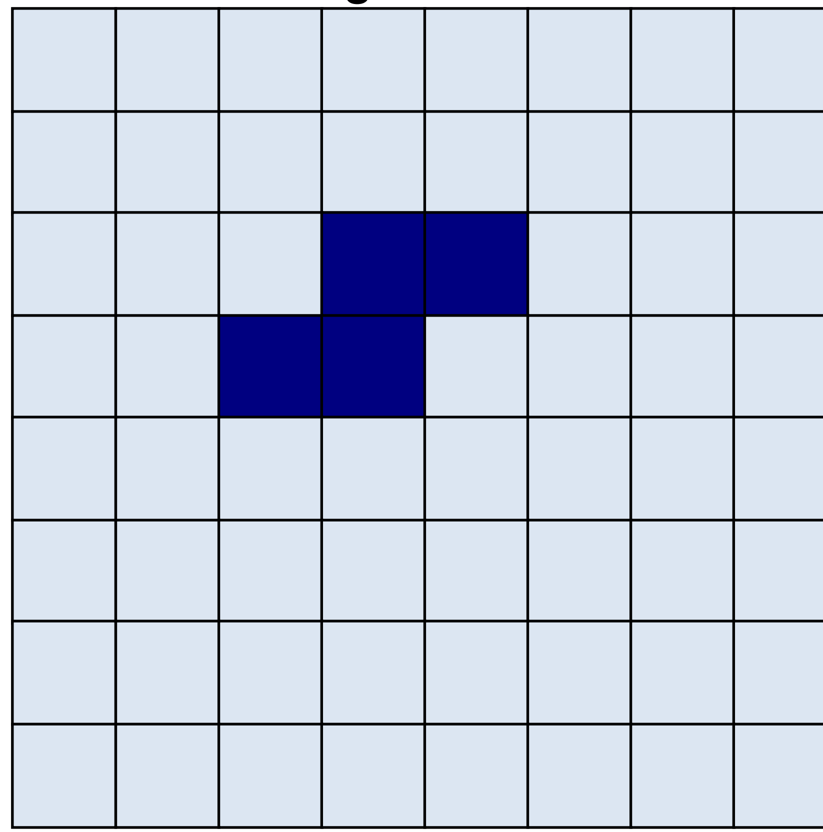


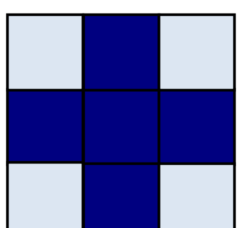
# Erosion – (cont.)

**Original Image**



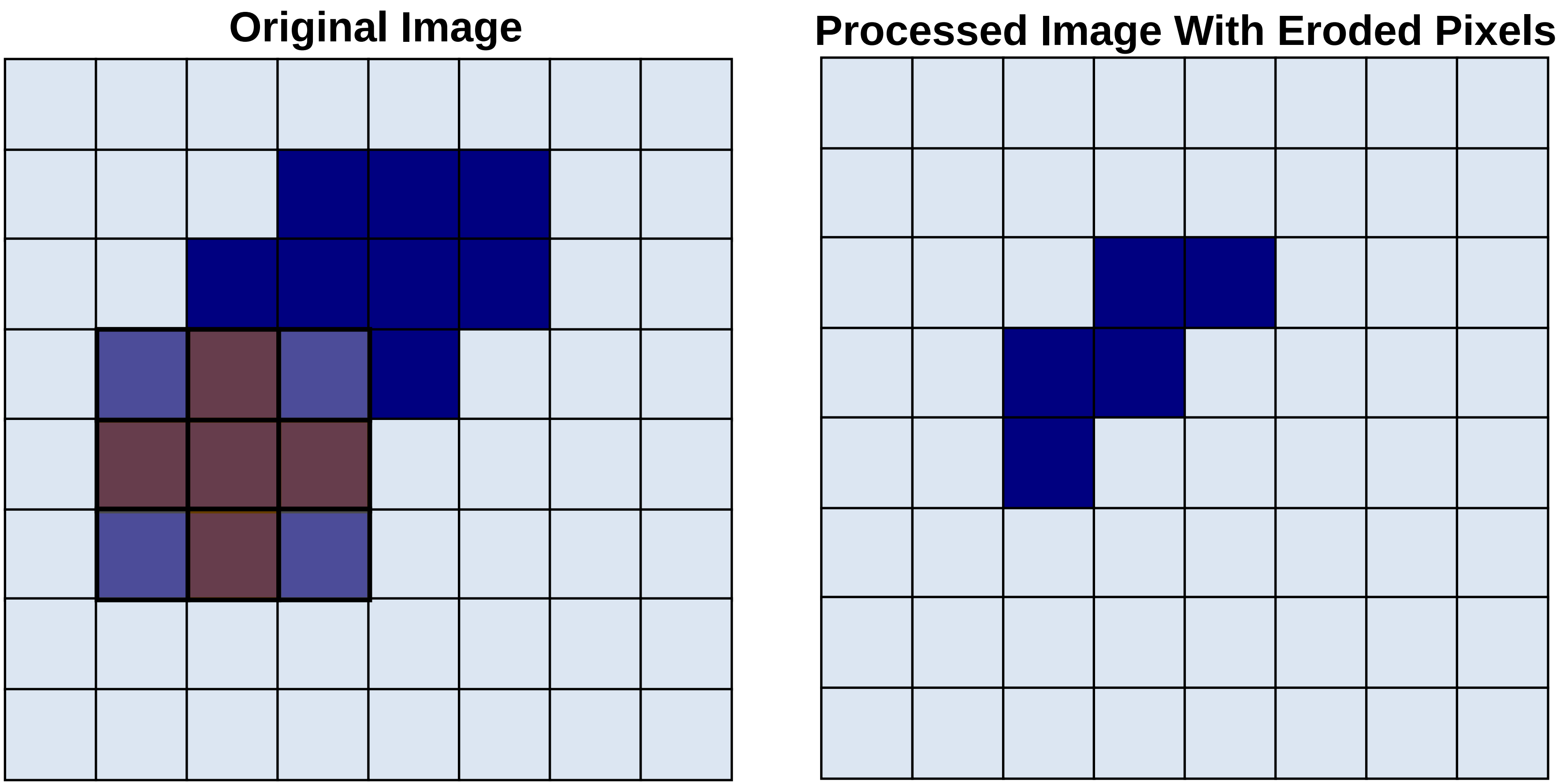
**Processed Image With Eroded Pixels**

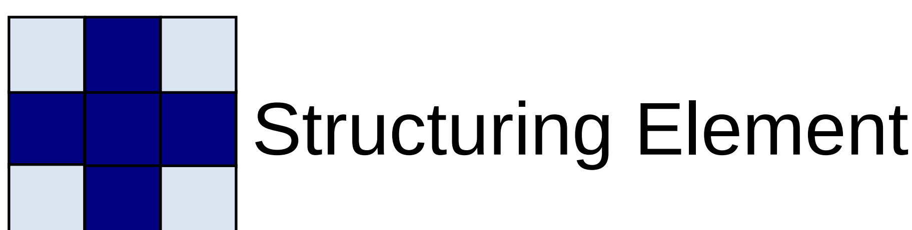




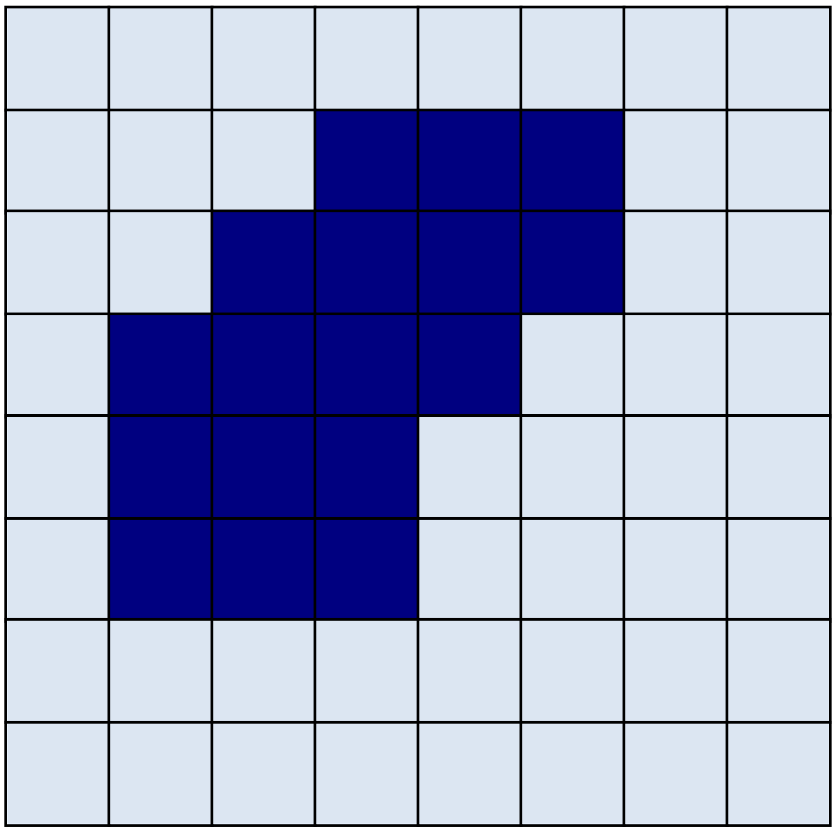
Structuring Element

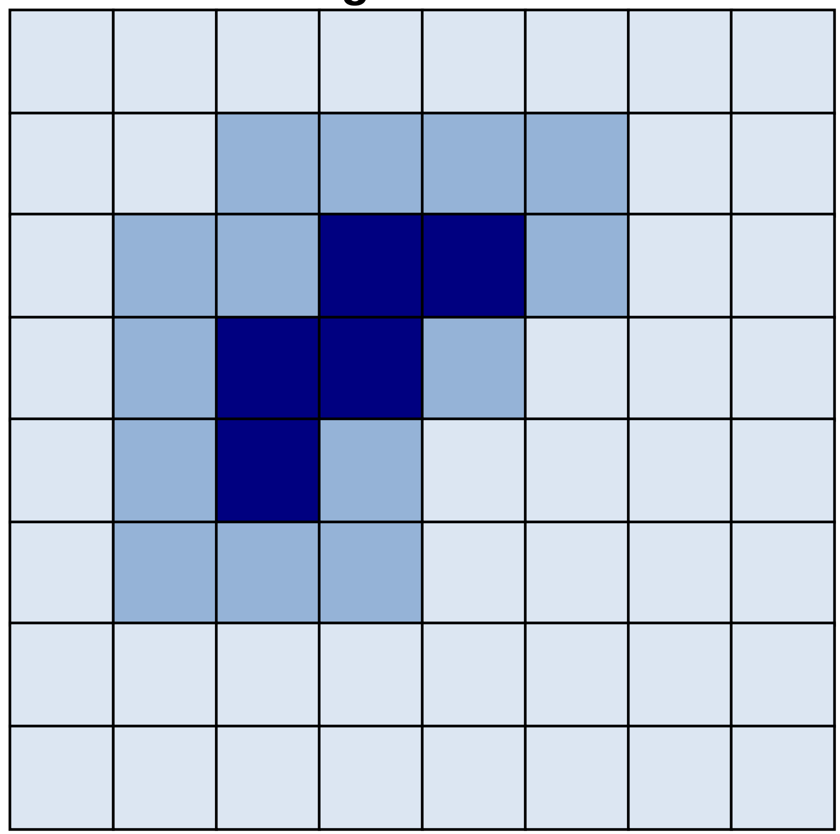
# Erosion – (cont.)

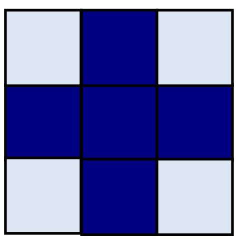


# Erosion – (cont.)

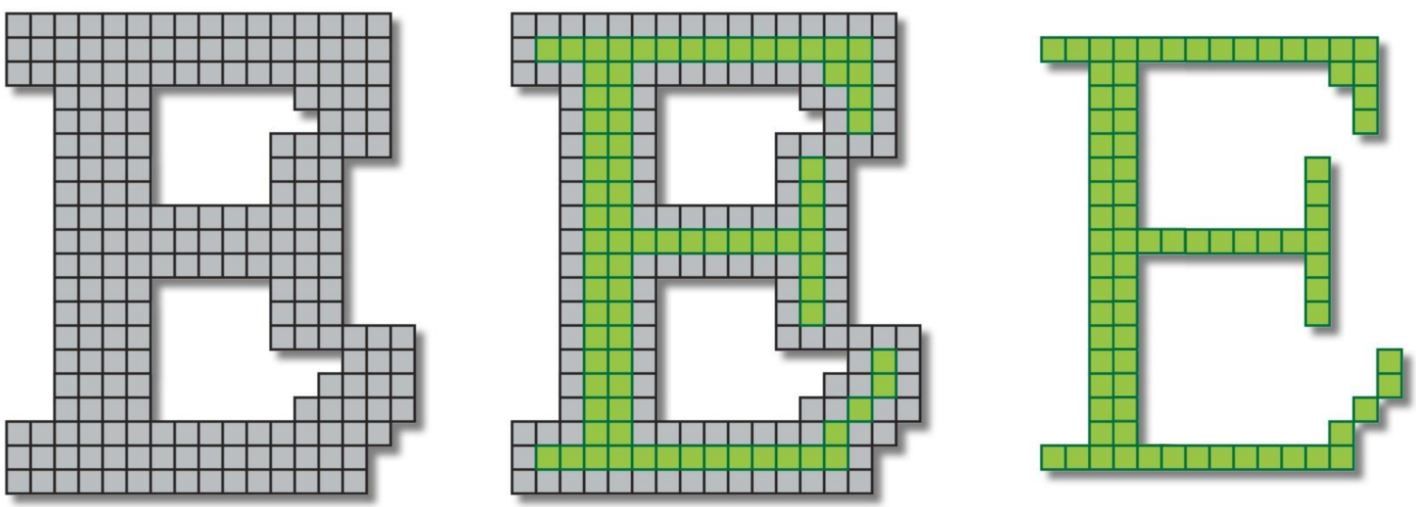


# Erosion – (cont.)

**Original image** | **Erosion by 3*3 square SE** | **Erosion by 5*5 square SE**



**Watch out:** In these examples, a 1 refers to a black pixel!

# Erosion – (cont.)

eroded image | erosion / original | original image

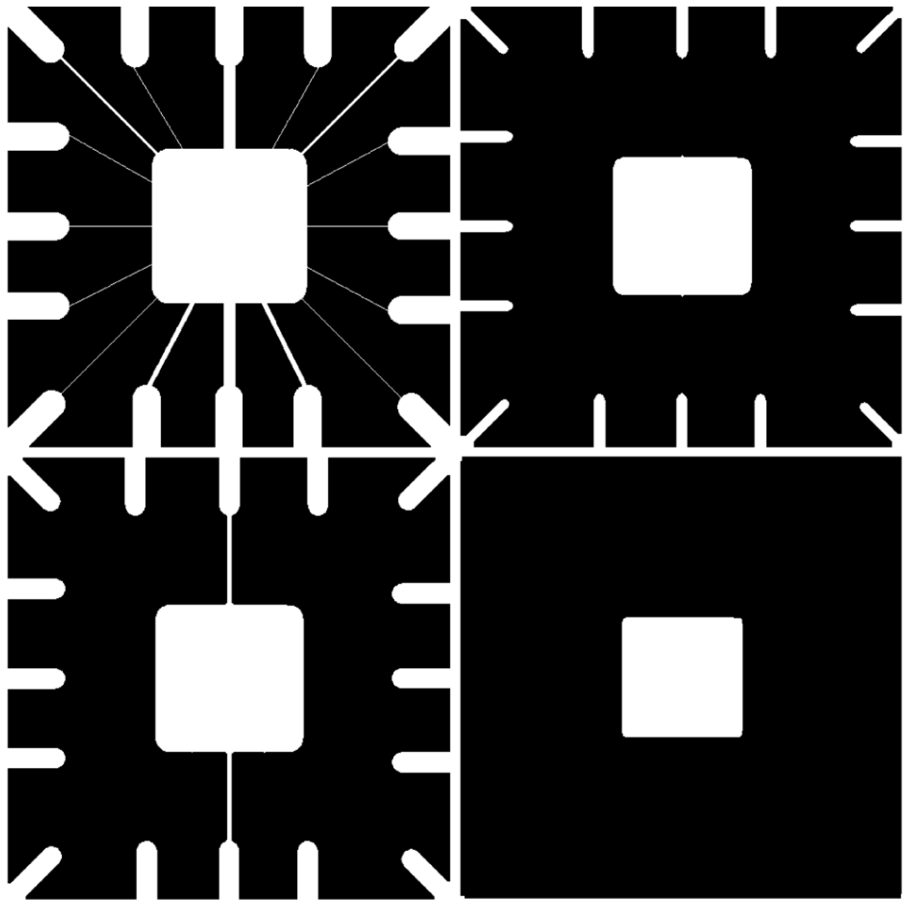
SE = $\mathbf{z}_8$

This is a piece of a larger image. Boundary effects are not apparent

# Erosion – (cont.)

Original image

After erosion with a disc of radius 10

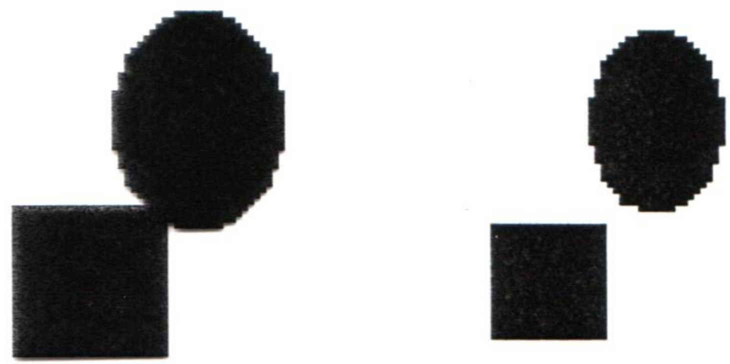


After erosion with a disc of radius 5

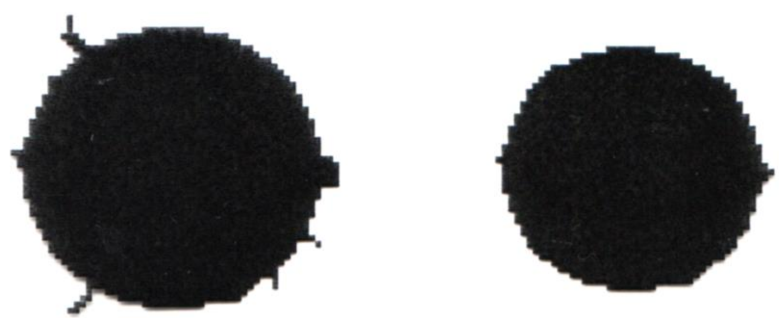
After erosion with a disc of radius 20

# Erosion – (cont.)

**Erosion can split apart joined objects**



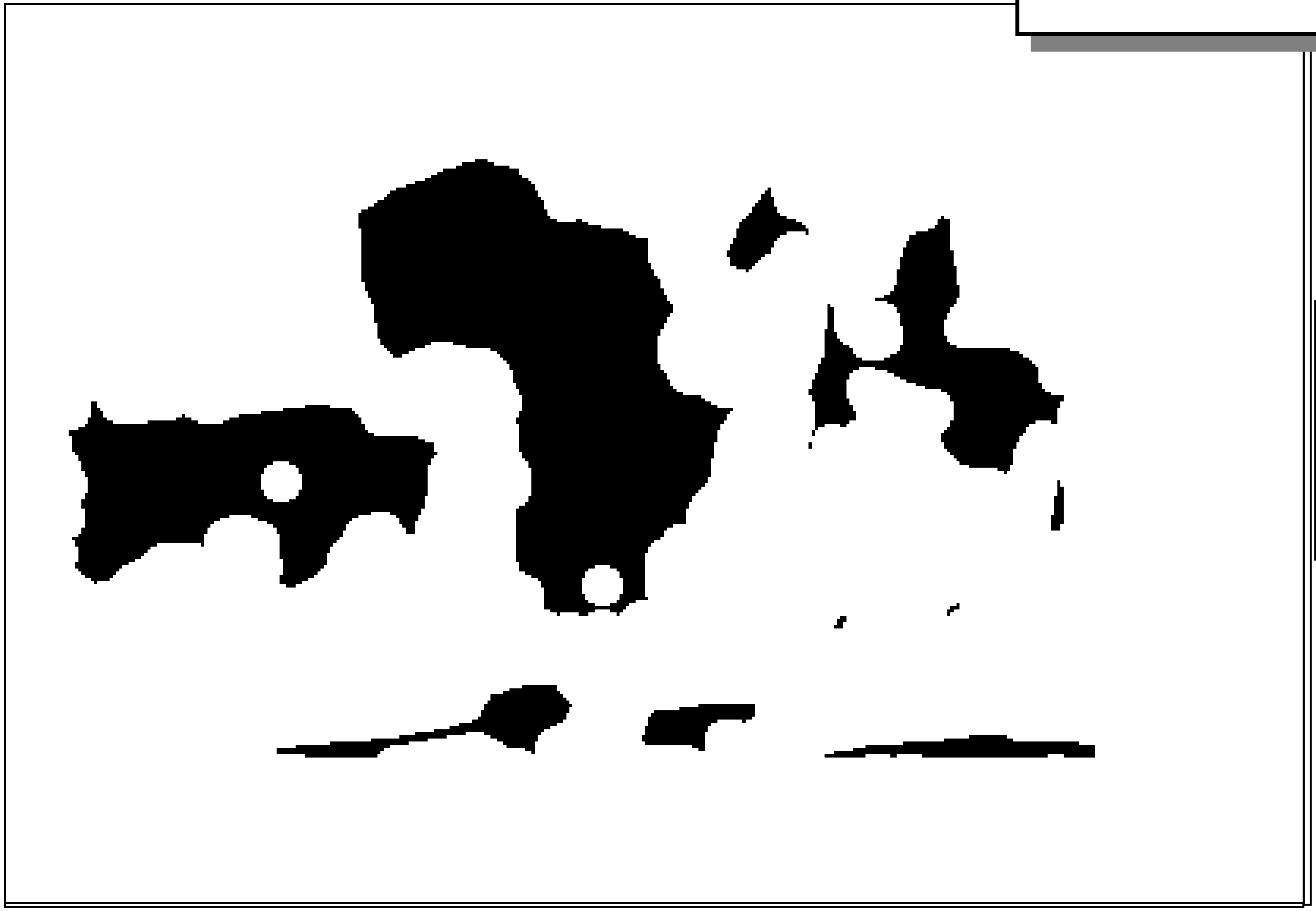
**Erosion can strip away extrusions**



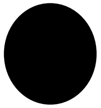
**Erosion shrinks objects!**

# Erosion – (cont.)

Examples of varying SE size

Structuring Element



Foreground is BLACK

# Basic Morphological Operations

## 2- Dilation

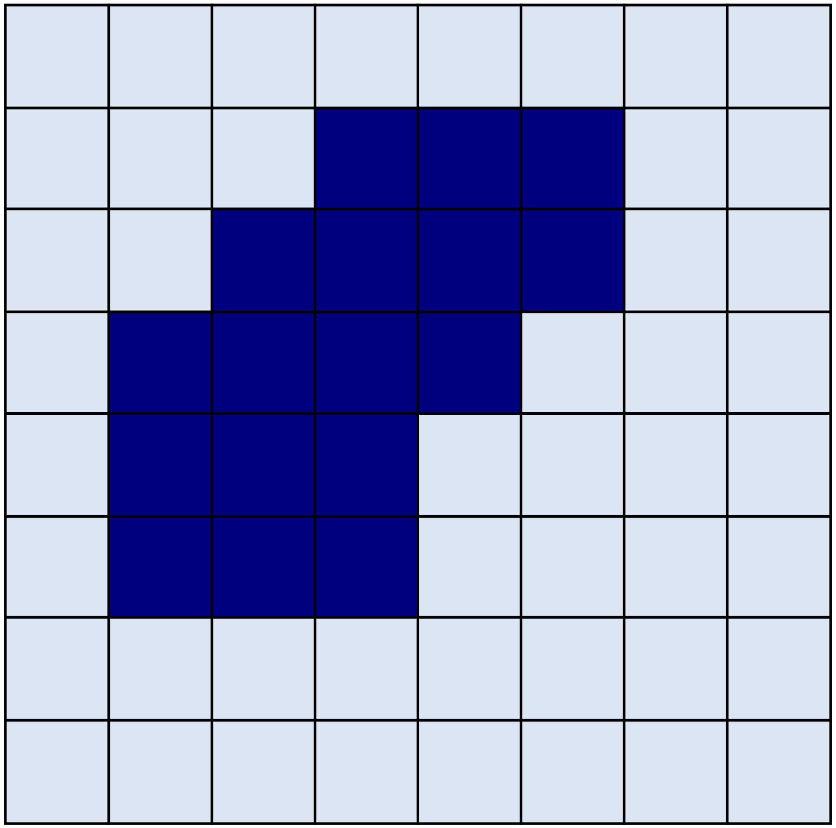
- **Dilation of image $X$ by structuring element $SE$ is given by $X \oplus SE$**
- **The structuring element $SE$ is positioned with its origin at $(x, y)$ and the new pixel value is determined using the rule:**

$$g(x,y) = \begin{cases} 1 \text{ if } SE \text{ hits } X \\ 0 \text{ otherwise} \end{cases}$$

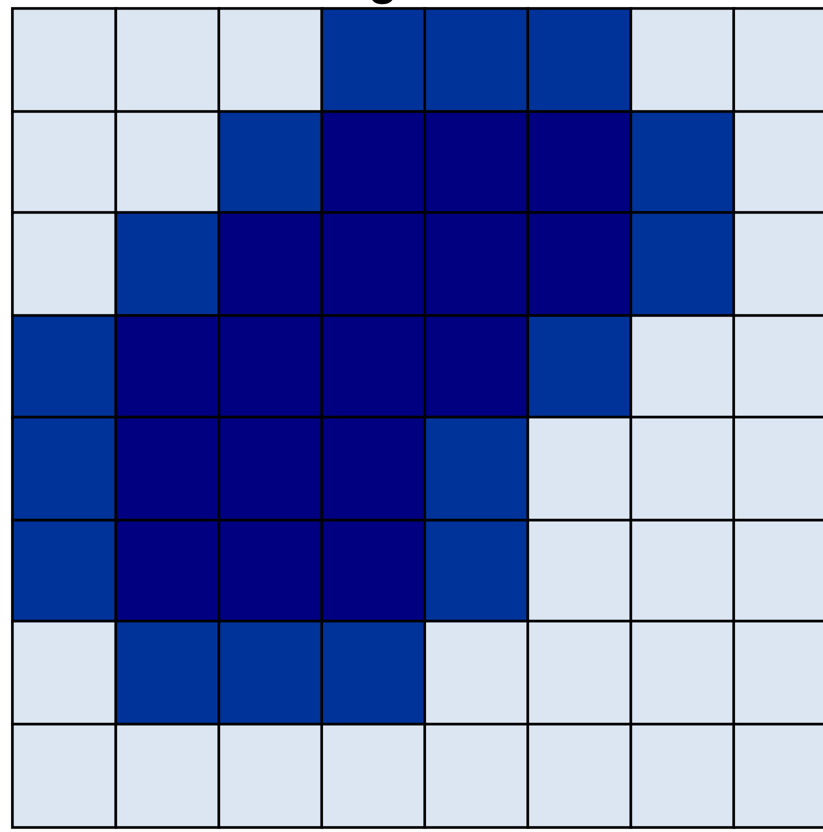
**Or in set language:** $X \oplus SE = \{z \mid (\hat{SE})_z \cap X \neq \phi\}$

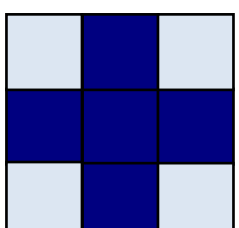
# Dilation

**Original Image**



**Processed Image With Dilated Pixels**





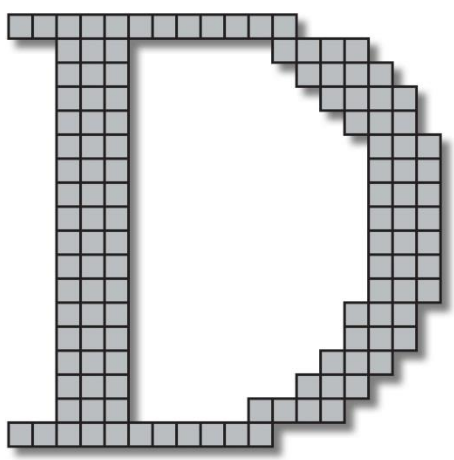
Structuring Element

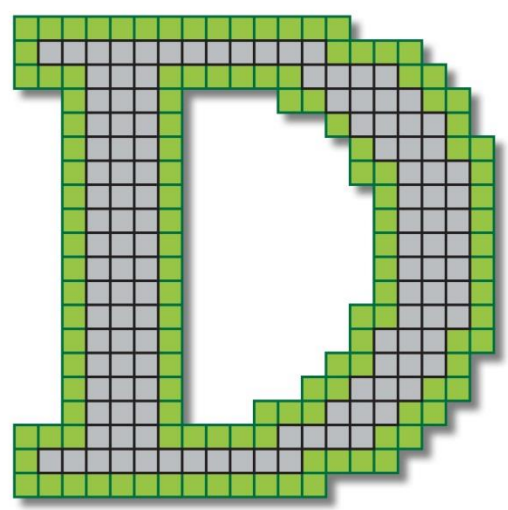
# Dilation – (cont.)

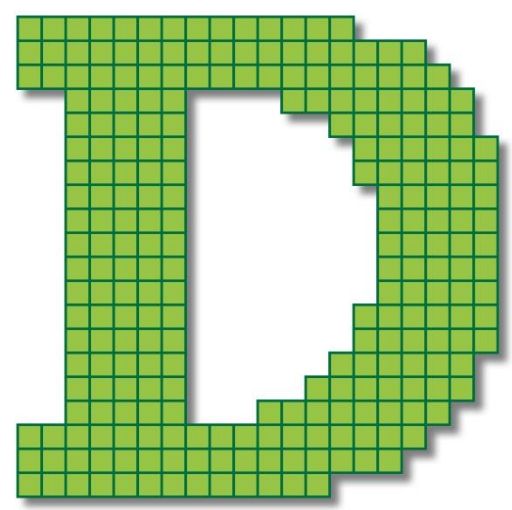
**Original image**

**Dilation by 3*3 square structuring element**

**Dilation by 5*5 square structuring element**

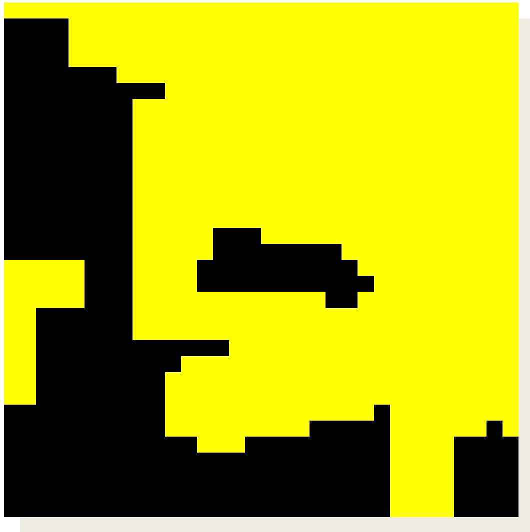




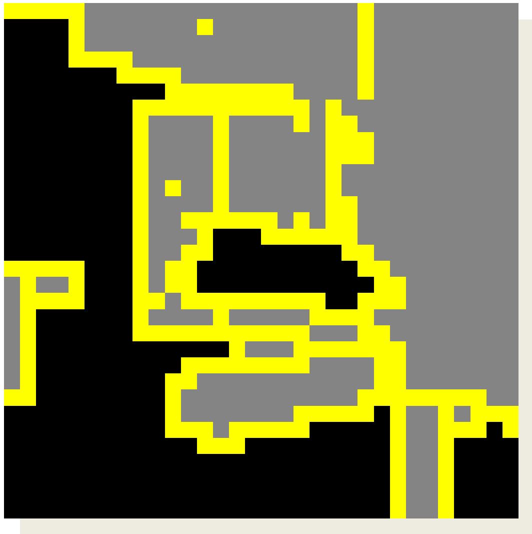


**Watch out:** In these examples, a 1 refers to a black pixel!

# Dilation – (cont.)



dilated image



original / dilation

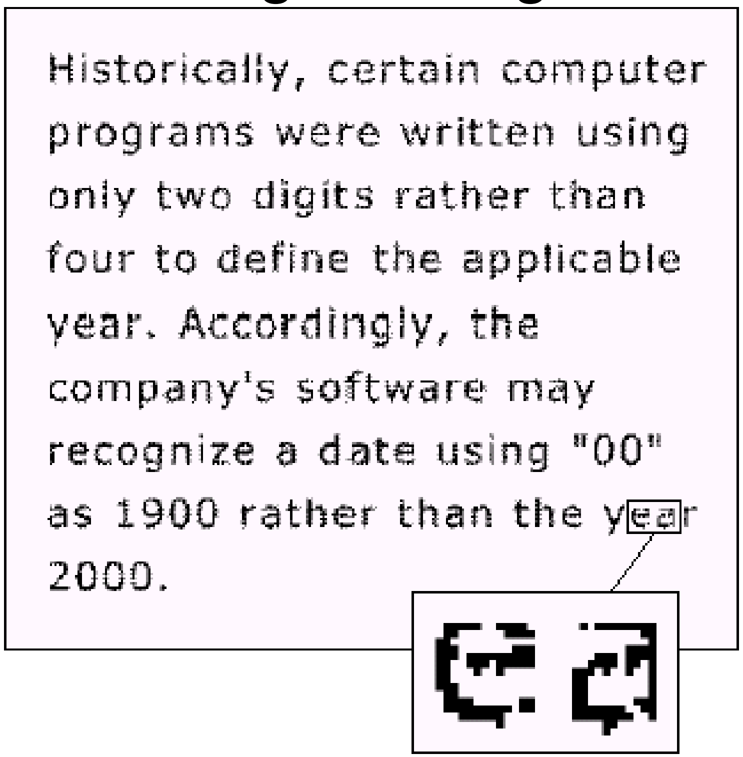
original image

SE = $\mathbf{z}_8$

This is a piece of a larger image.  Boundary effects are not apparent

# Dilation – (cont.)

**Original image**




**After dilation**


| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 0 |

Structuring element

# Dilation – (cont.)

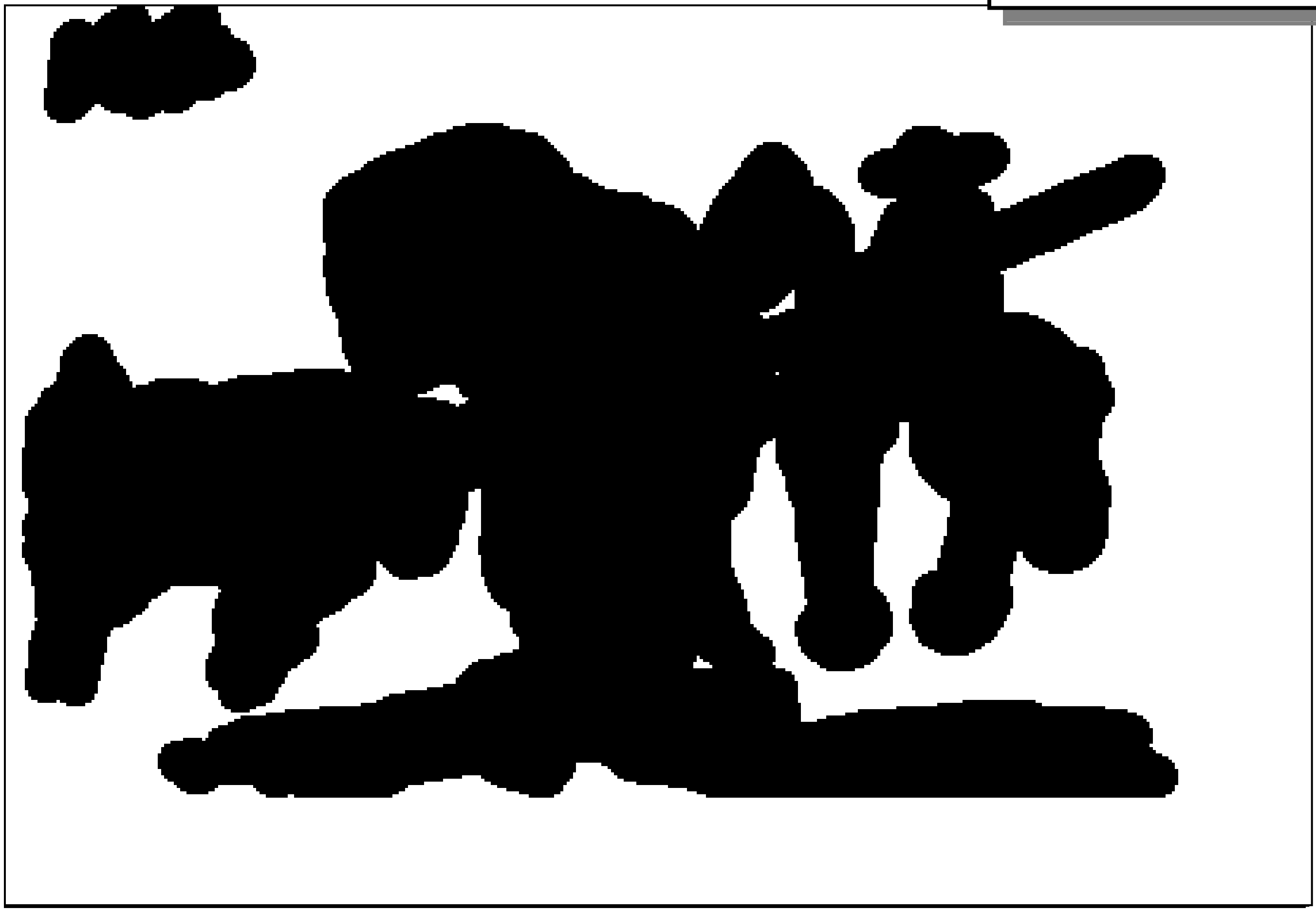
Dilation can repair breaks

Dilation can repair intrusions

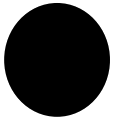
Dilation enlarges objects!

# Dilation – (cont.)

Examples of varying SE size



**Structuring Element**



Foreground is BLACK

# Compound Operations

- More interesting morphological operations can be performed by combinations of erosions and dilations.
- The most widely used compound operations are:

    - Opening.

    - Closing.

# Opening

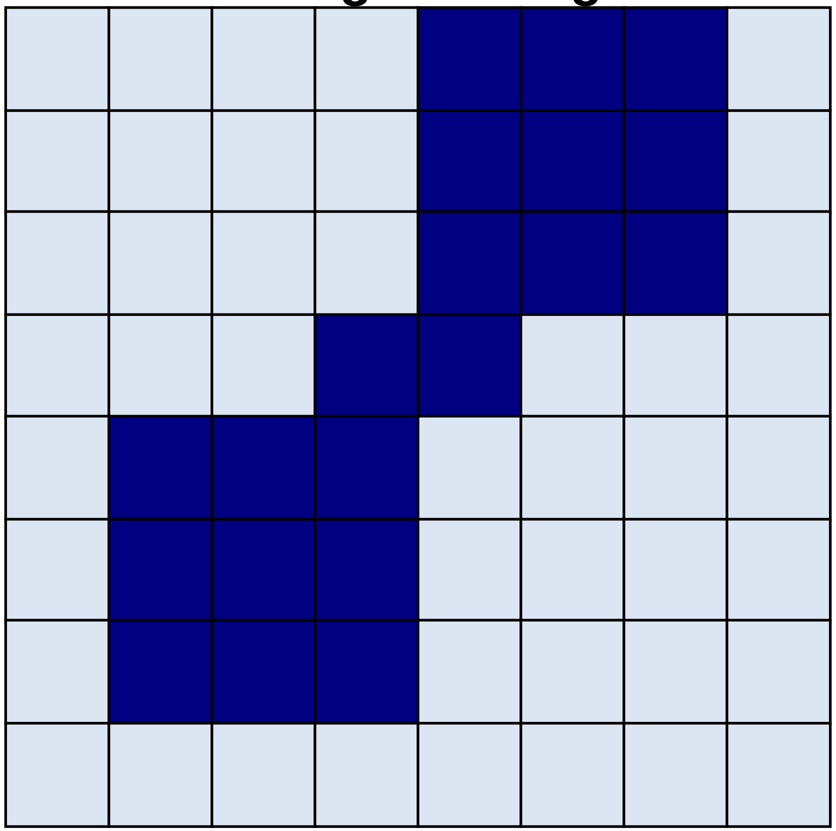
- **The opening of image $AX$ by structuring element $SE$, denoted $X \circ SE$ is simply an erosion followed by a dilation:**

$$X \circ SE = (X \ominus SE) \oplus SE$$

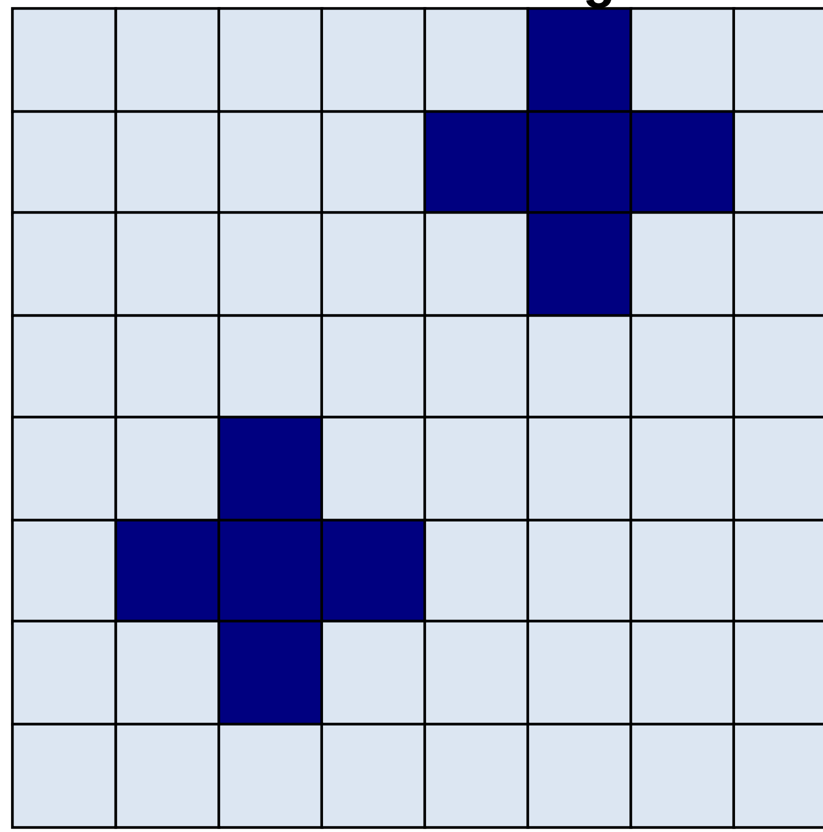
- **Opening smoothes the contour of objects, breaks narrow bridges, and eliminates thin protrusions.**

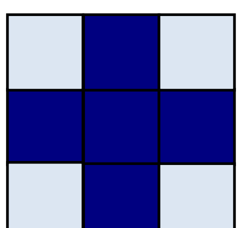
# Opening – (cont.)

**Original Image**



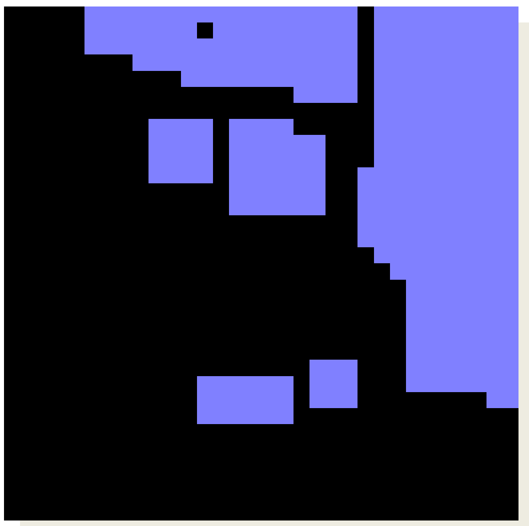
**Processed Image**





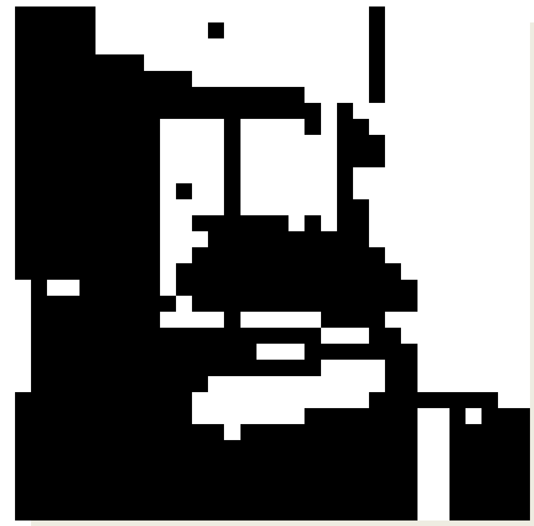
Structuring Element

# Opening – (cont.)



opening

erosion

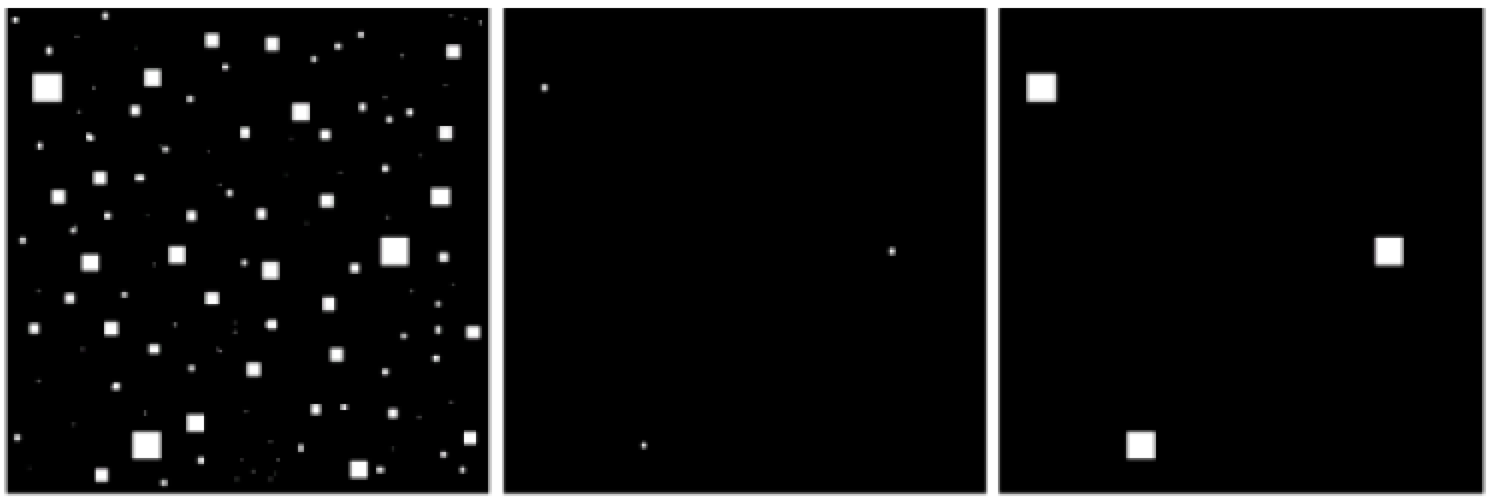


original

SE = $\mathbf{Z}_8$

This is a piece of a larger image. Boundary effects are not apparent

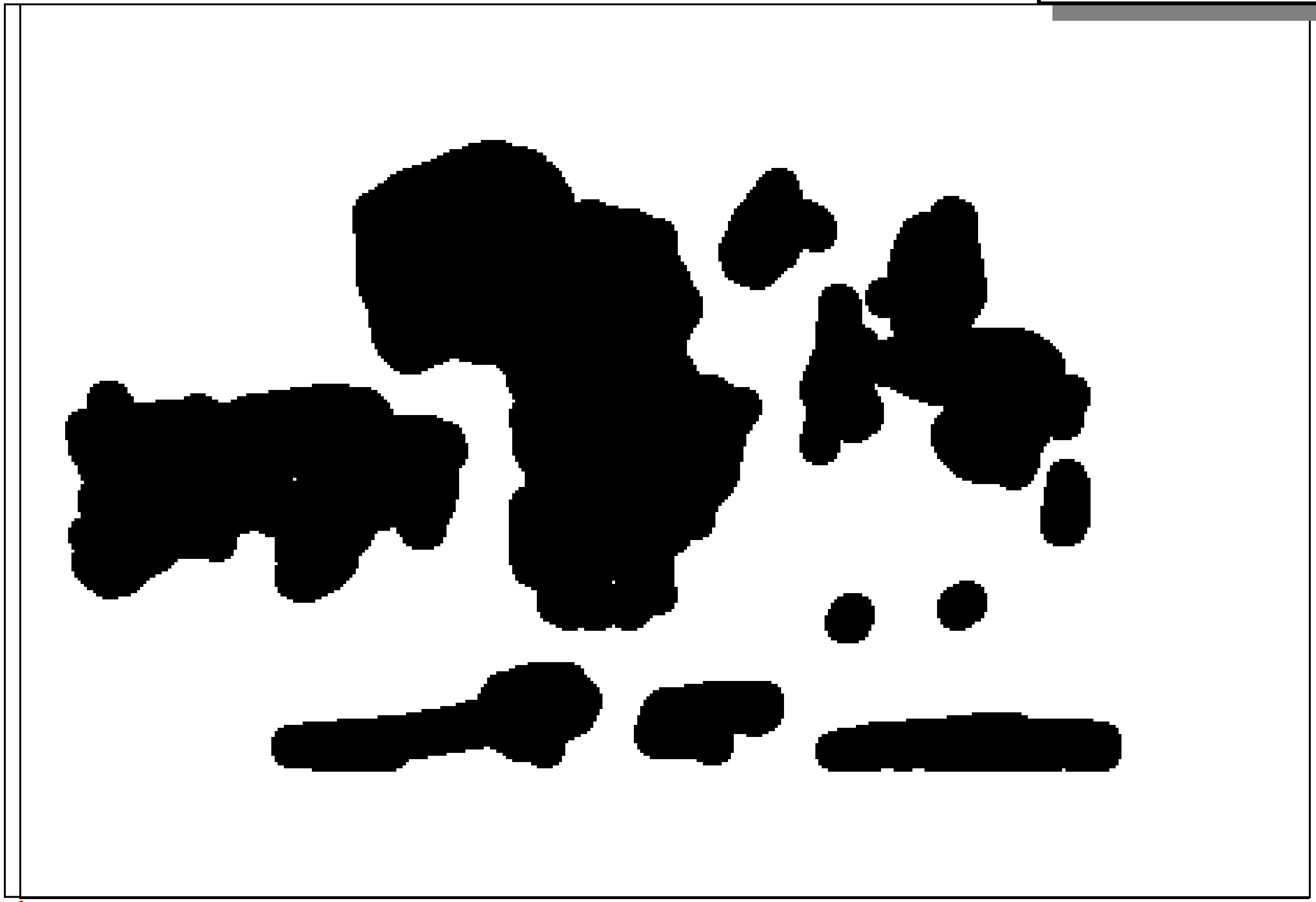
# Opening – (cont.)

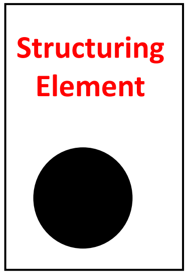


original erosion opening

# Opening – (cont.)

Examples of varying SE size






Foreground is BLACK

# Closing

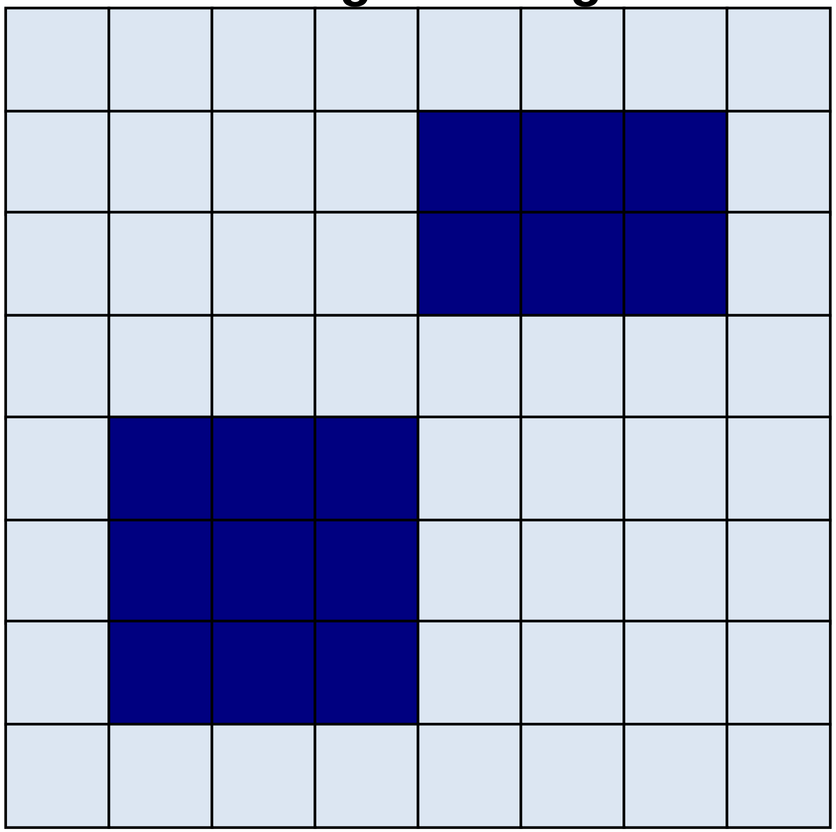
- **The closing of image $X$ by structuring element $SE$, denoted $X \bullet SE$ is simply a dilation followed by an erosion:**

$$X \bullet SE = (X \oplus SE) \ominus SE$$

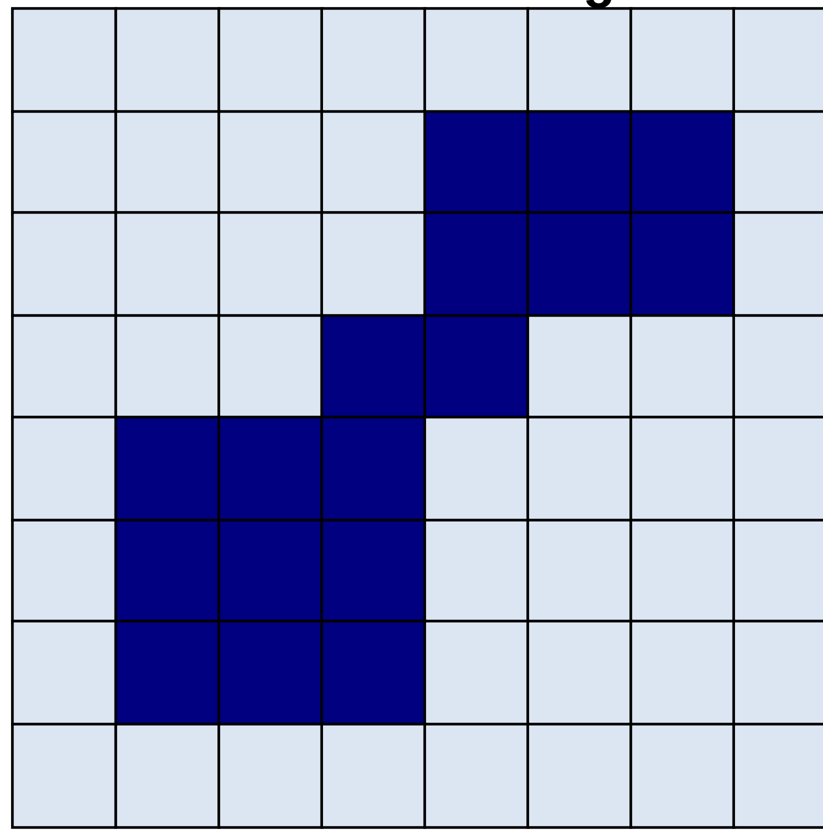
- **Closing smoothes sections of contours as well, but fuses narrow breaks, eliminate small holes, and fill gaps.**

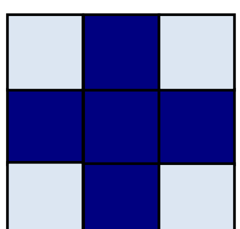
# Closing – (cont.)

**Original Image**



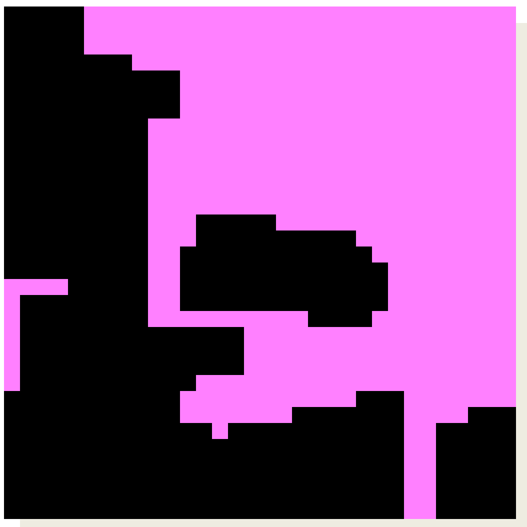
**Processed Image**



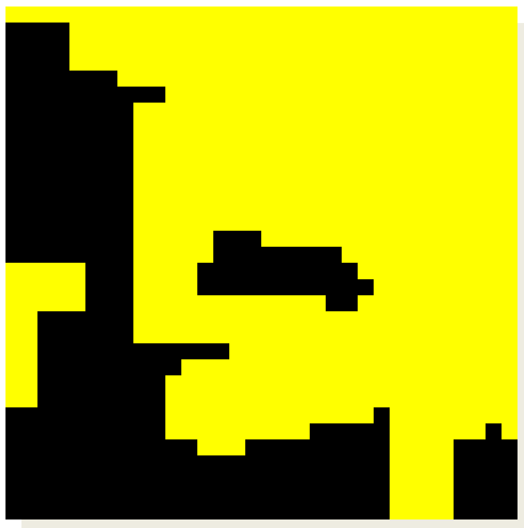


Structuring Element

# Closing – (cont.)



closing



dilation

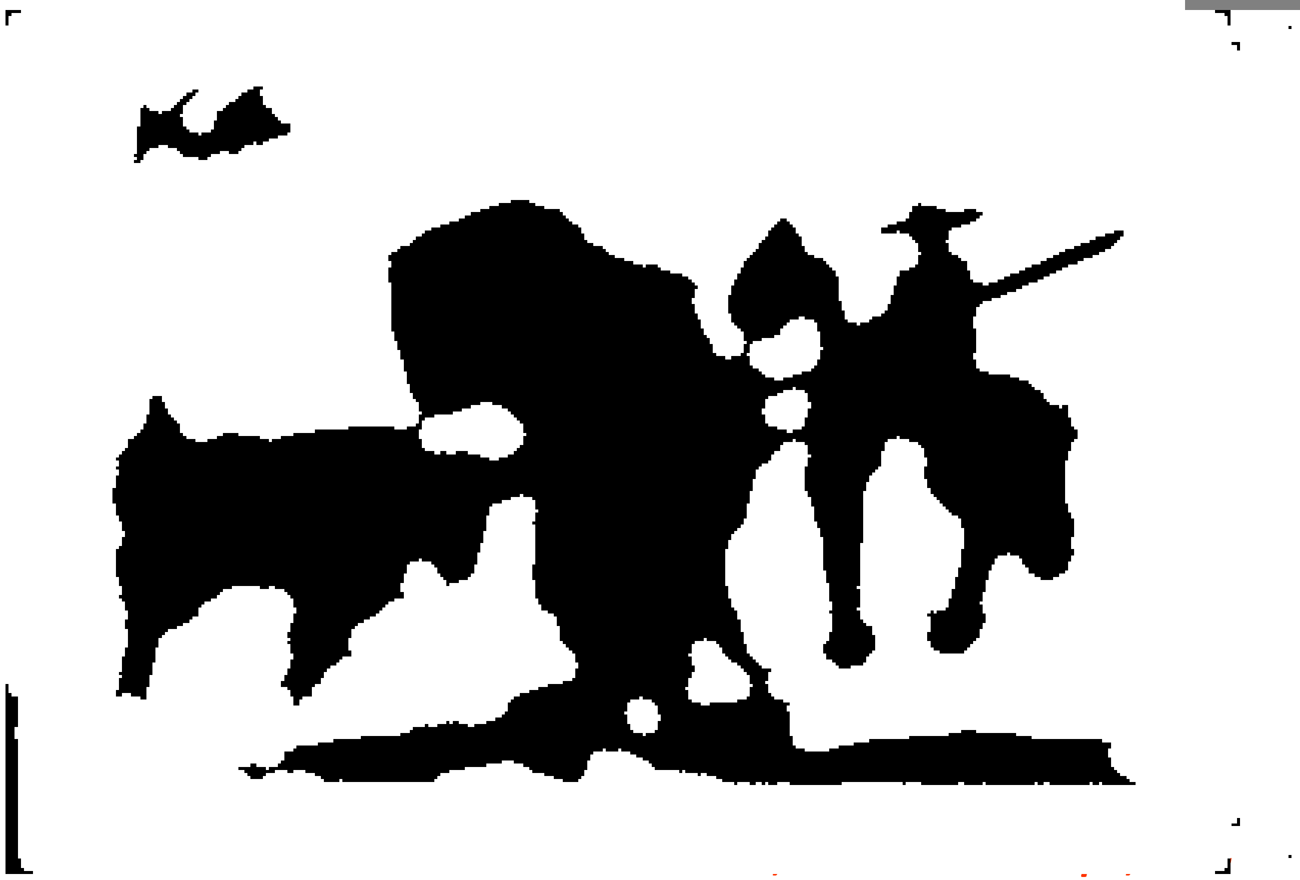
original

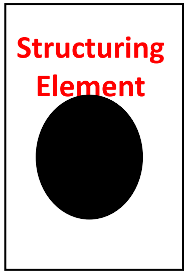
SE = $\mathbf{z}_8$

[1]using the reflected SE, $\check{\mathbf{z}}$

# Closing – (cont.)

Examples of varying SE size

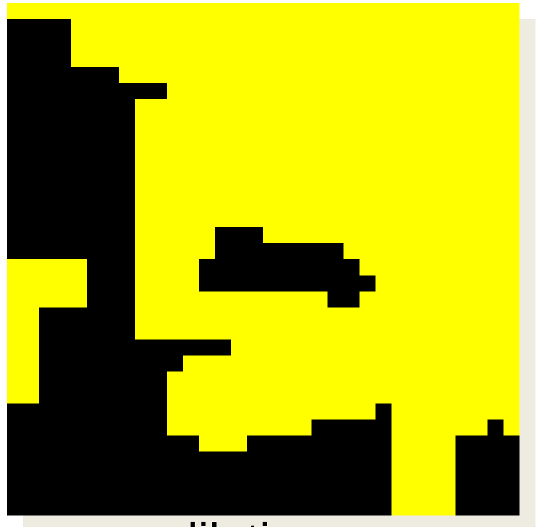




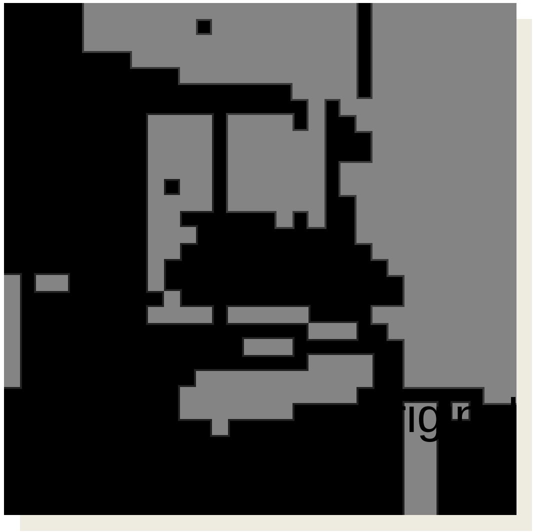

Foreground is BLACK

# Compare

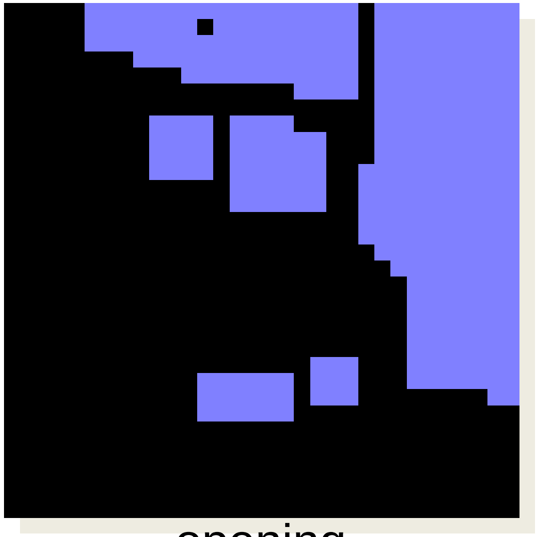
closing



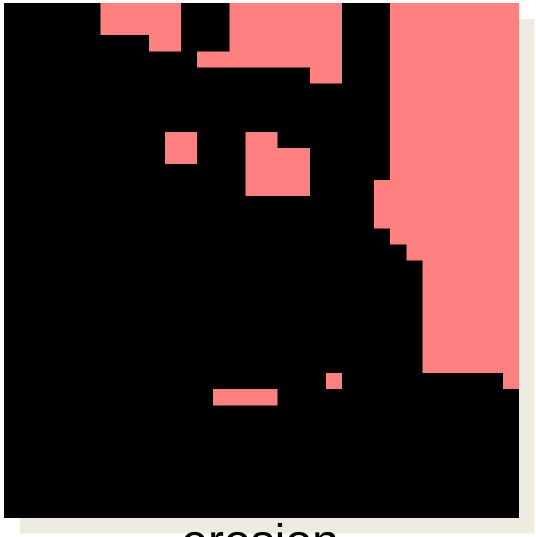
dilation



original



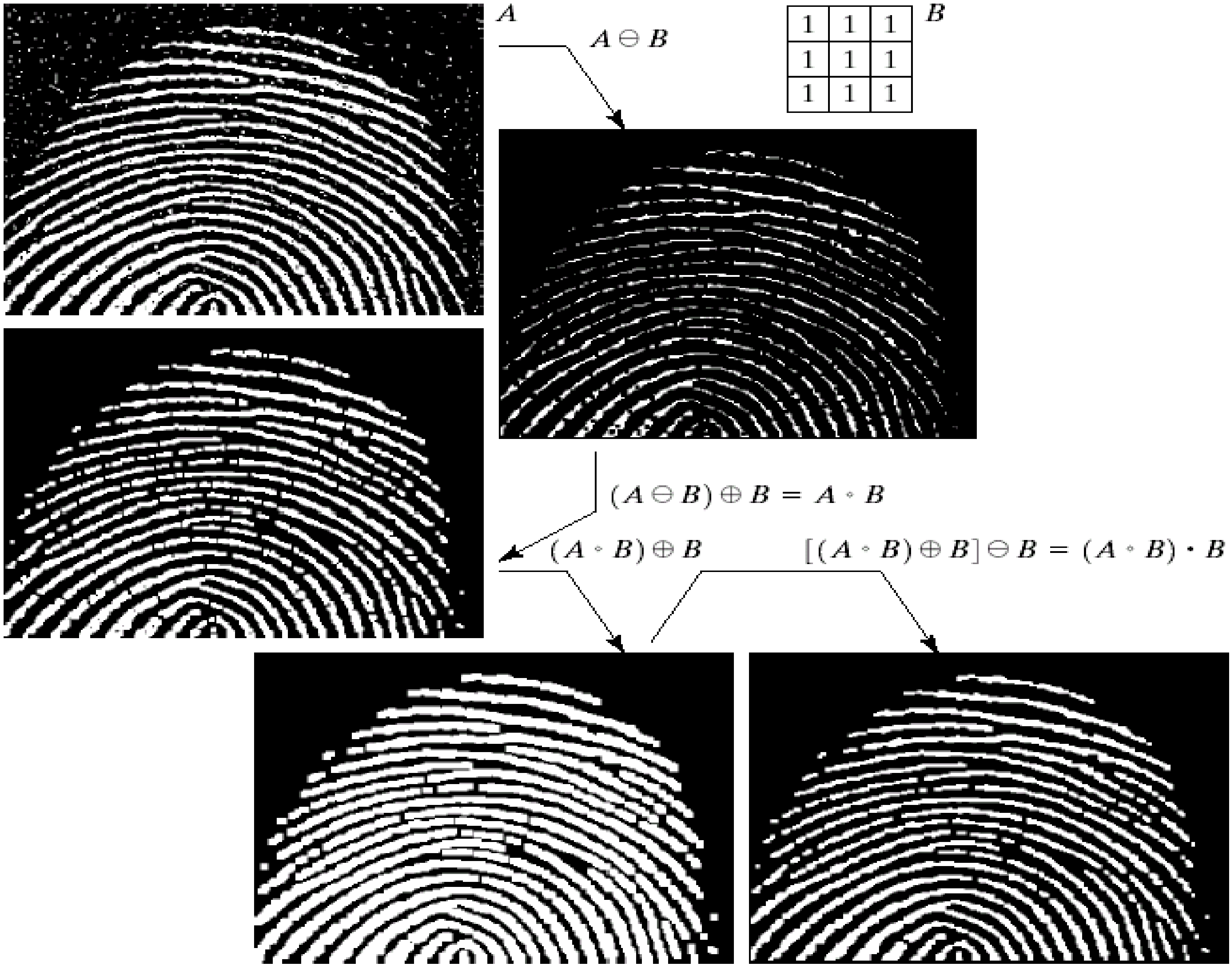
opening



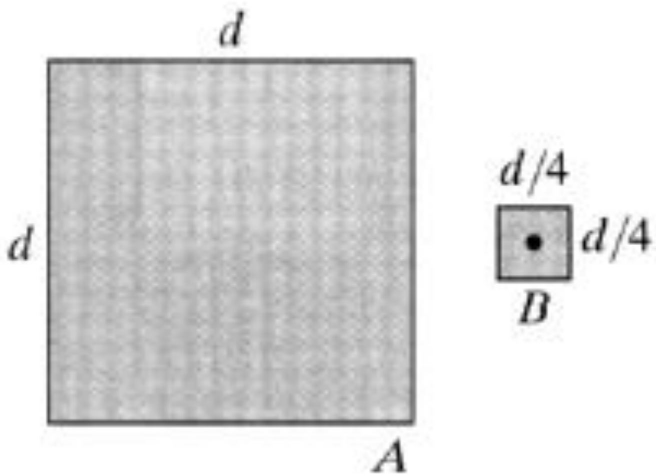
erosion

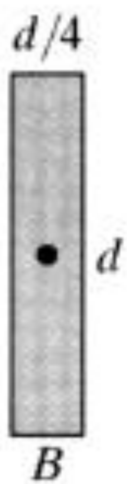
SE = $\mathbf{Z}_8$

[1]using the reflected SE, $\check{\mathbf{Z}}$


$A$
$A \ominus B$
$B$
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |
$(A \ominus B) \oplus B = A \circ B$
$(A \circ B) \oplus B$
$[(A \circ B) \oplus B] \ominus B = (A \circ B) \bullet B$

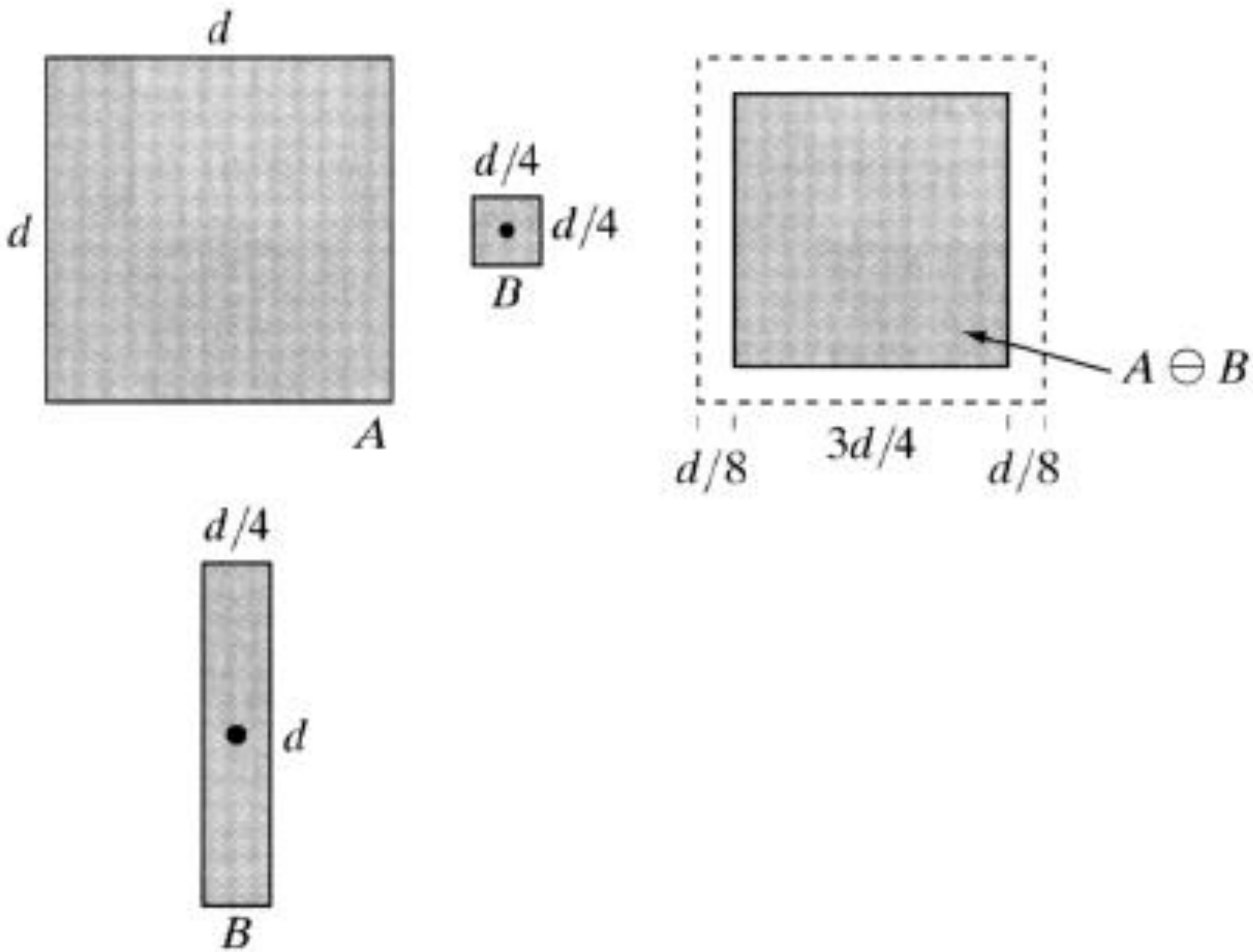
# Examples



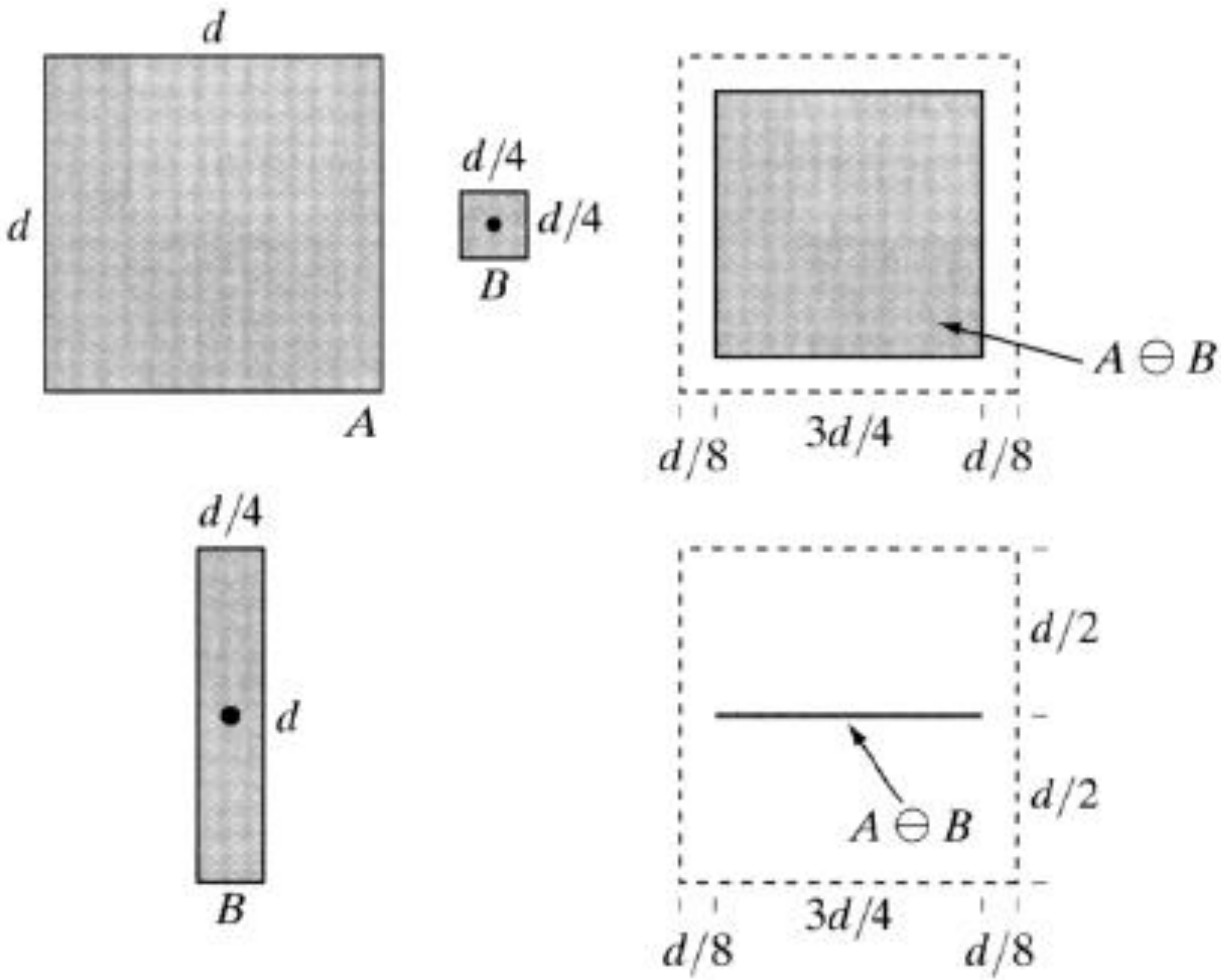

**Erosion**

# Examples – (cont.)



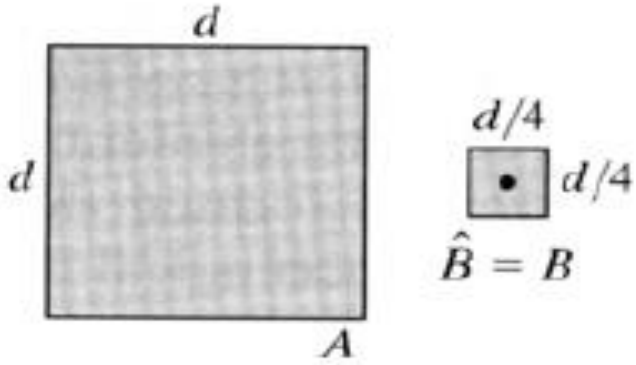

Erosion

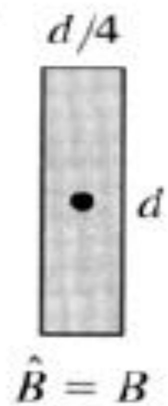
# Examples – (cont.)




**Erosion**

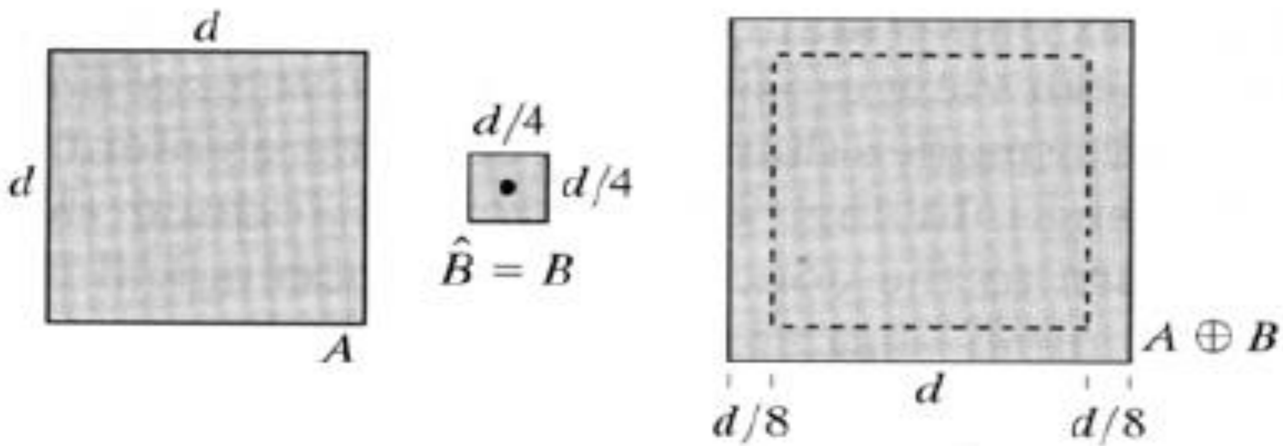
# Examples – (cont.)

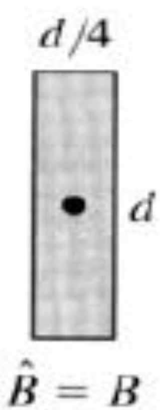



**Dilation**

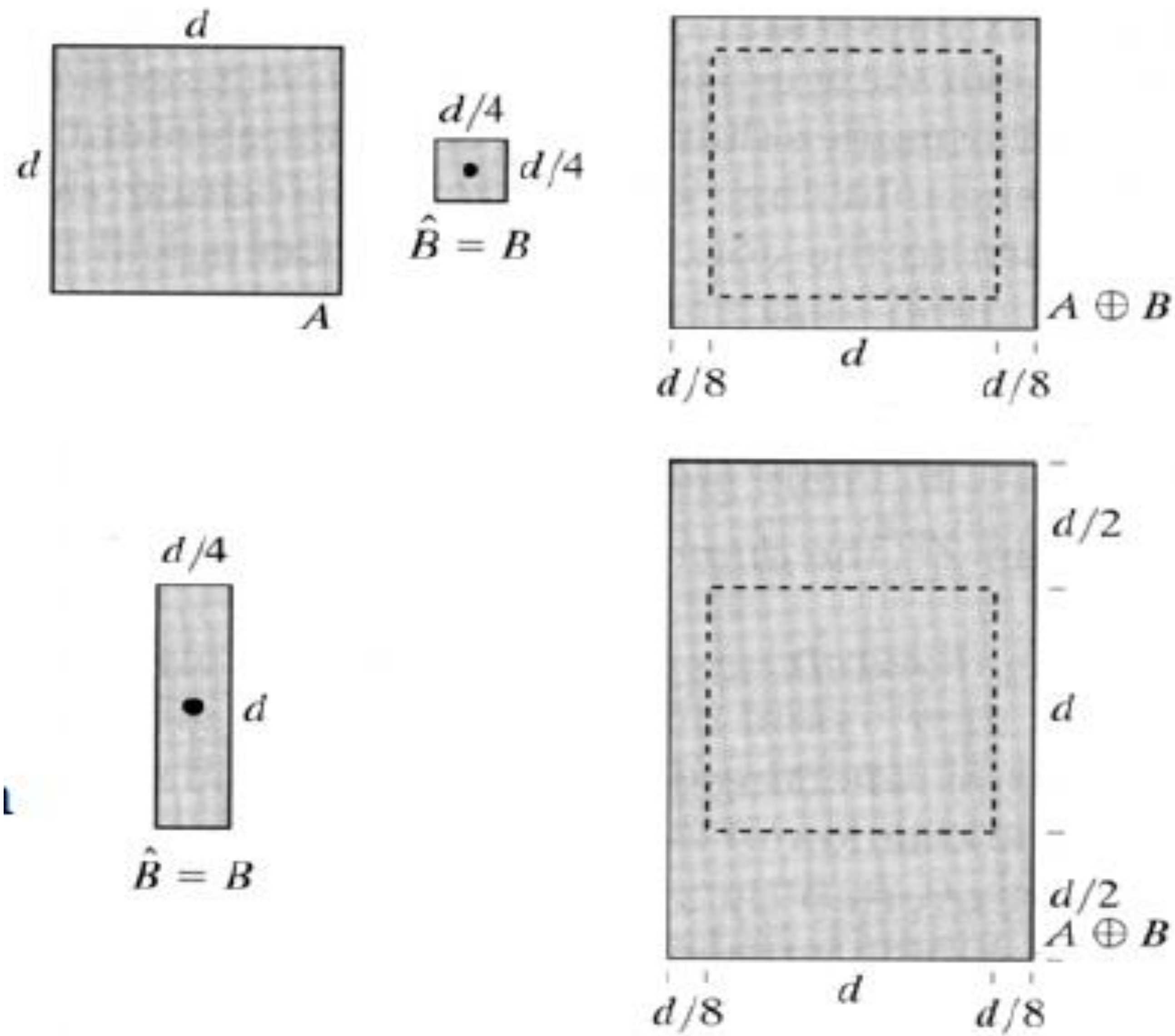
# Examples – (cont.)



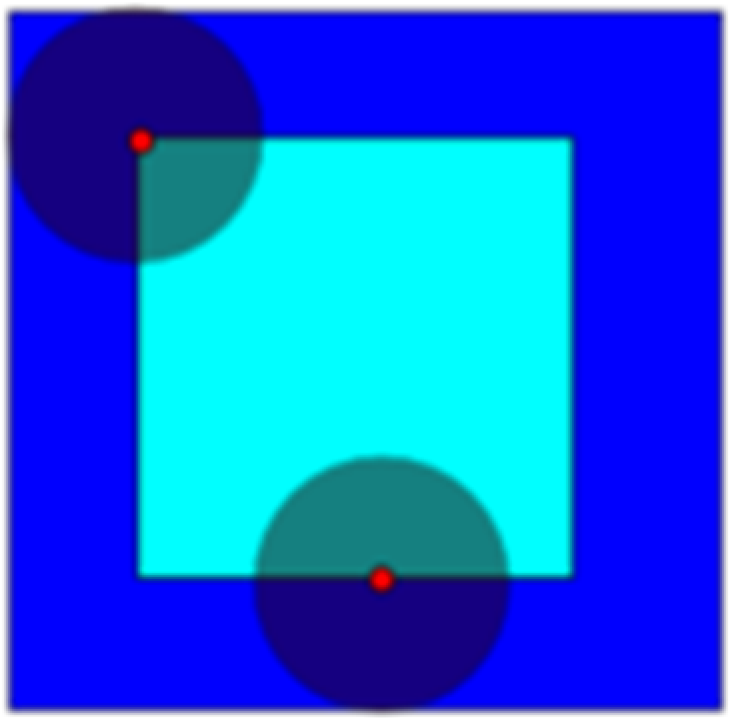

**Dilation**

# Examples – (cont.)

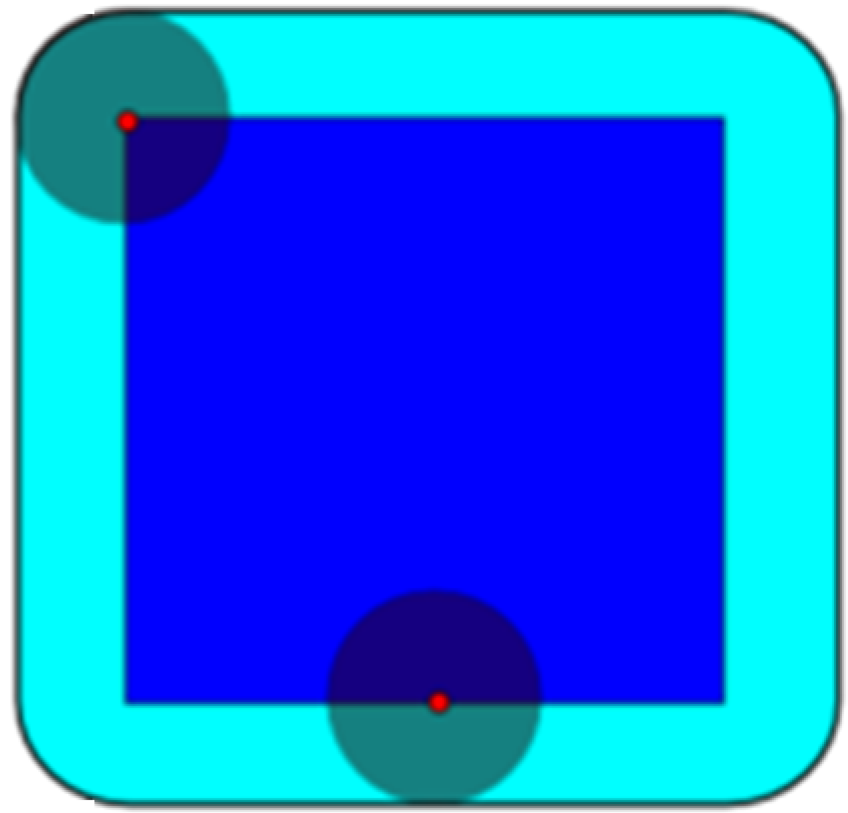



**Dilation**

# Examples – (cont.)

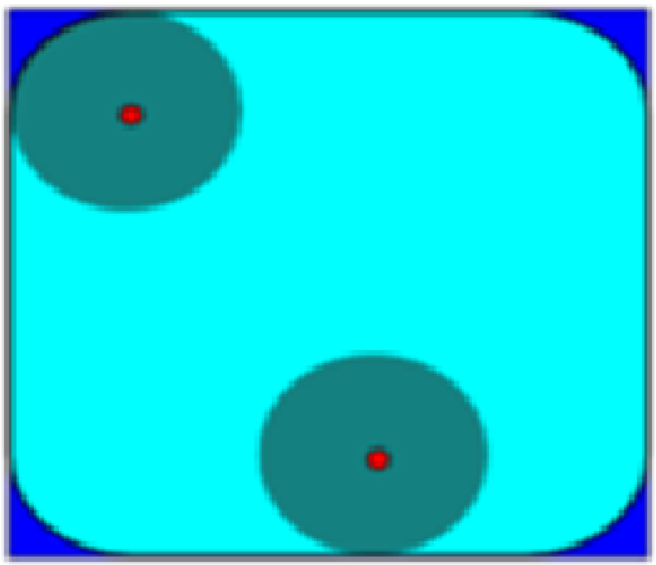


The erosion of the dark-blue square by a disk, resulting in the light-blue square

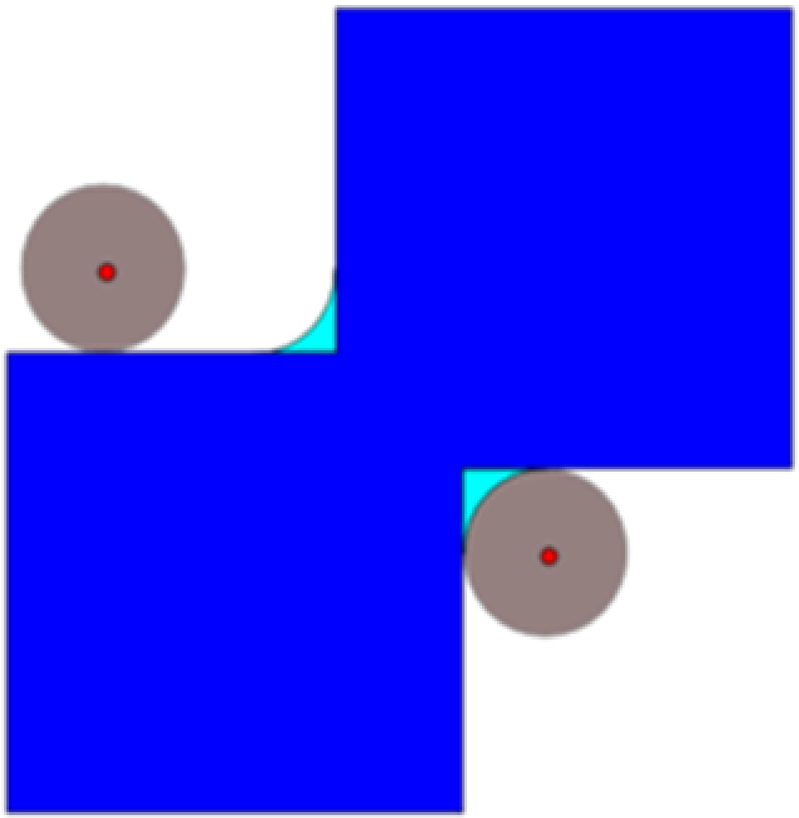


The dilation of the dark-blue square by a disk, resulting in the light-blue square with rounded corners

# Examples – (cont.)



The opening of the dark-blue square by a disk, resulting in the light-blue square with round corners.



The closing of the dark-blue shape (union of two squares) by a disk, resulting in the union of the dark-blue shape and the light-blue areas.

# Examples – (cont.) Opening

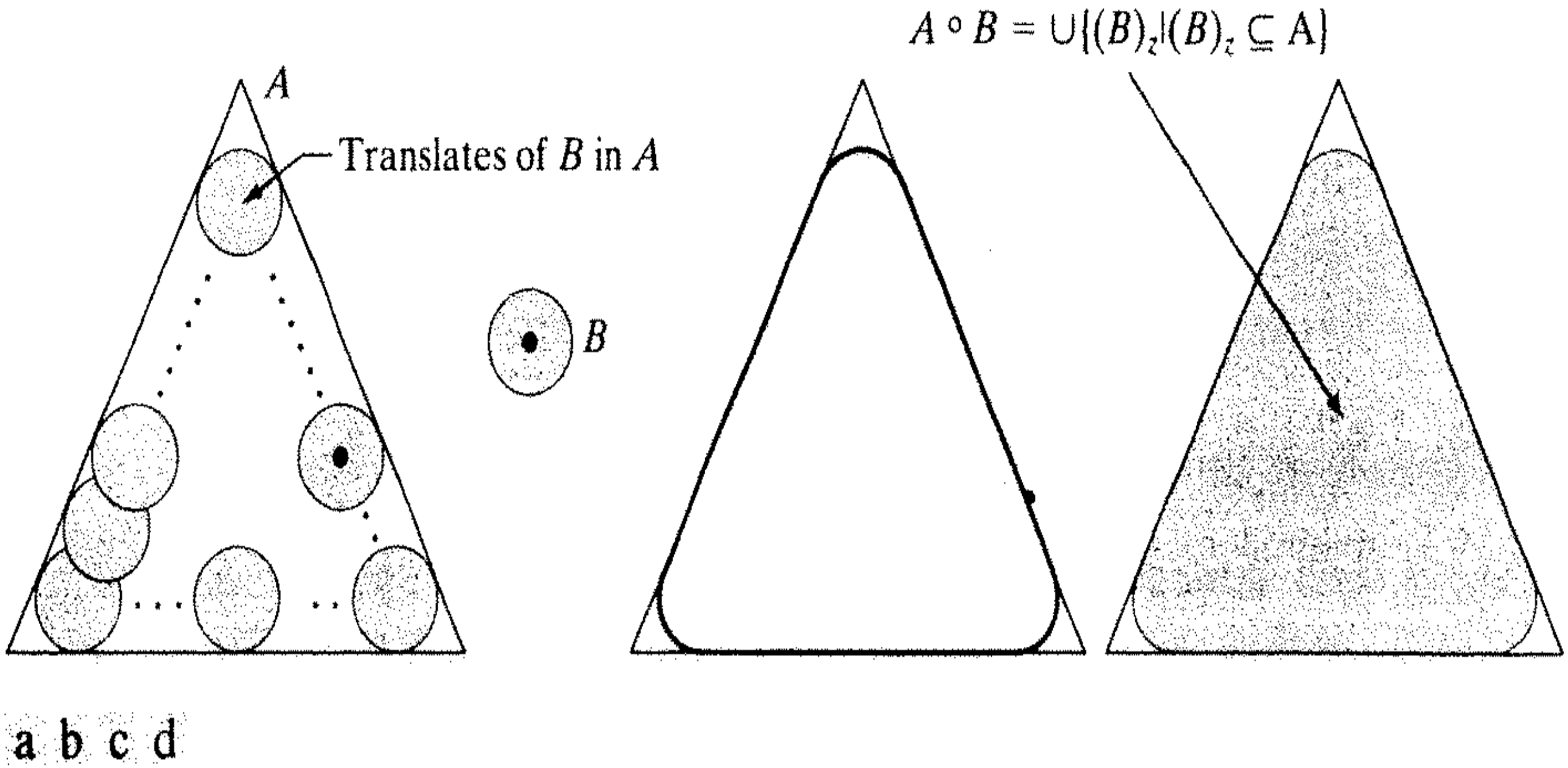
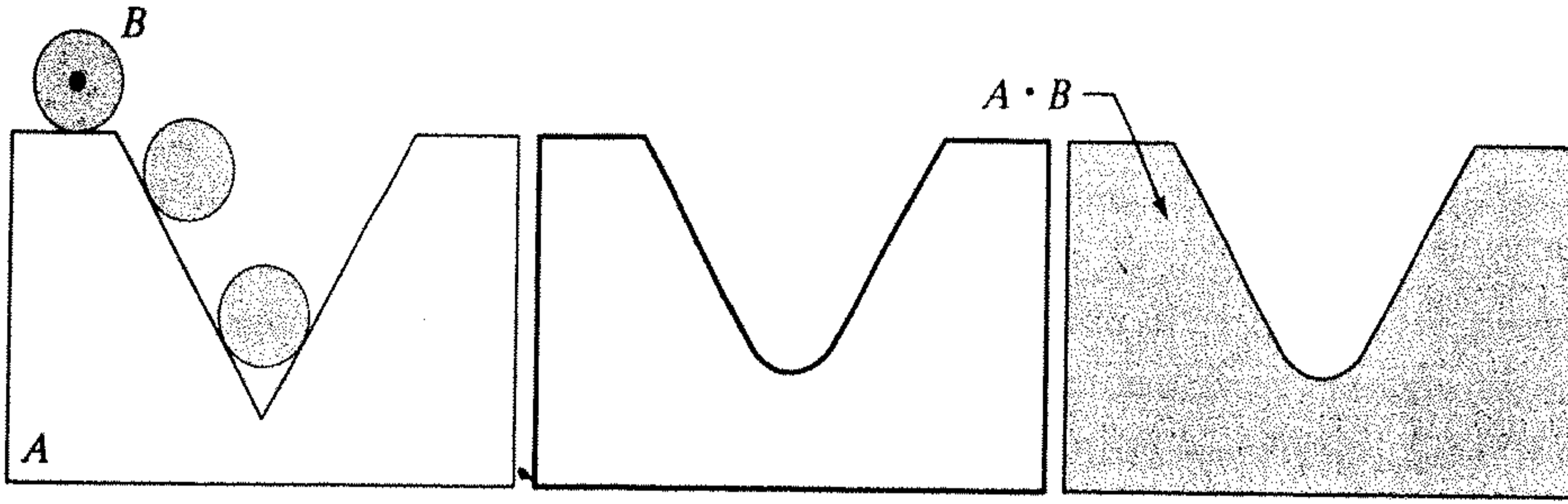



**FIGURE 9.8** (a) Structuring element $B$ "rolling" along the inner boundary of $A$ (the dot indicates the origin of $B$). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade $A$ in (a) for clarity.

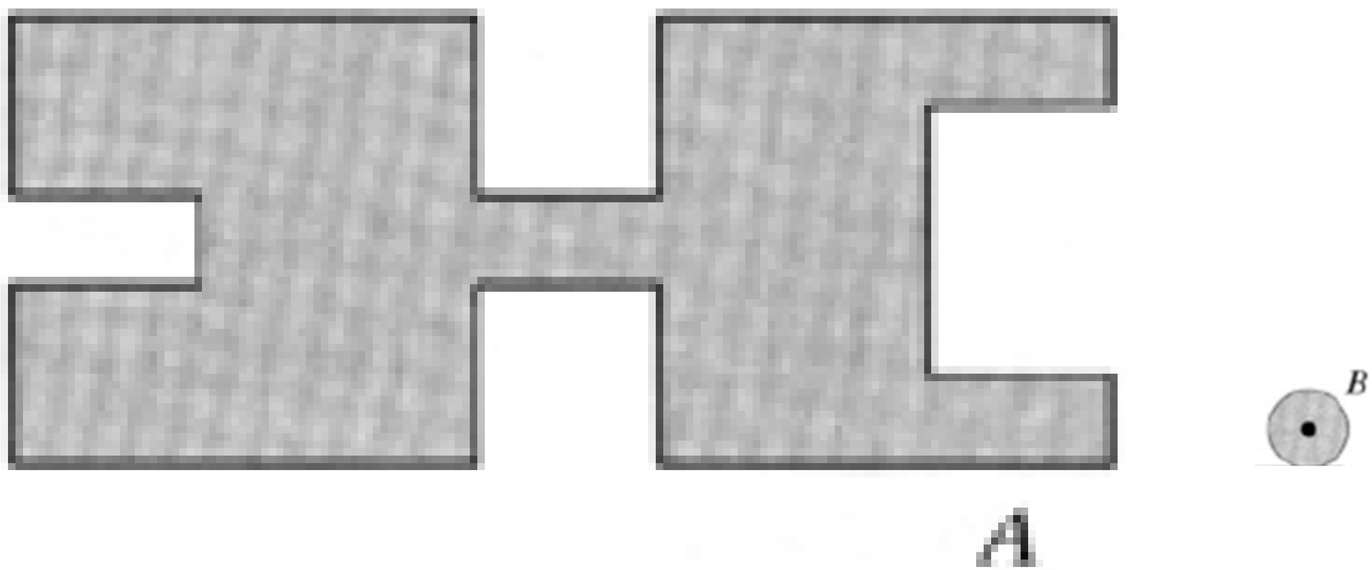
# Examples – (cont.) Closing



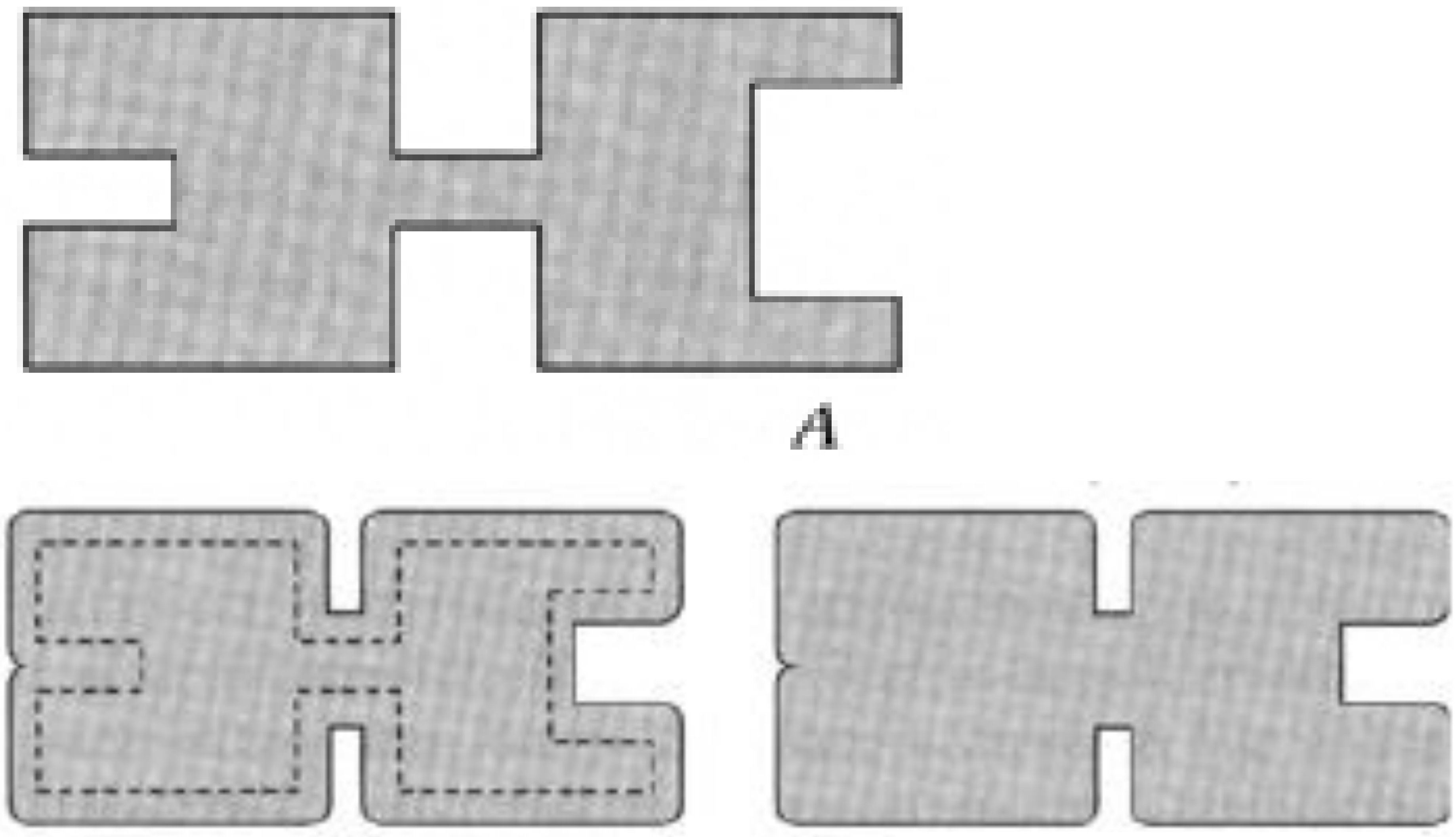

a b c

**FIGURE 9.9** (a) Structuring element $B$ "rolling" on the outer boundary of set $A$. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade $A$ in (a) for clarity.

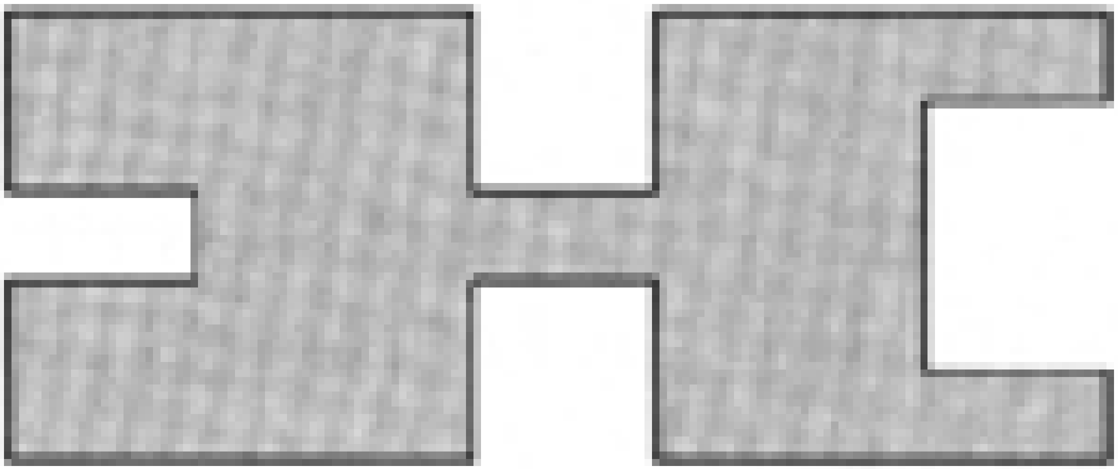
# Examples – (cont.)



# Examples – (cont.)

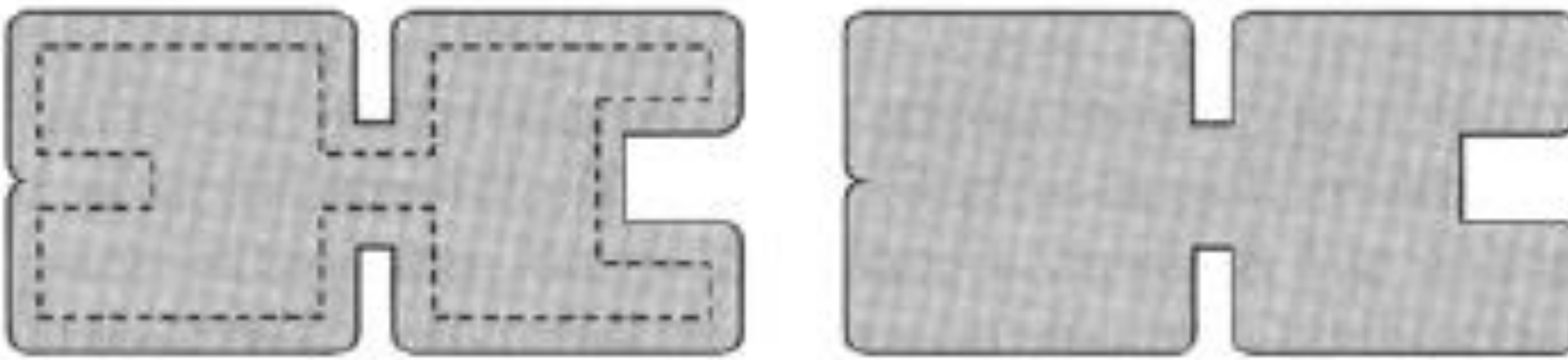


# Examples – (cont.)



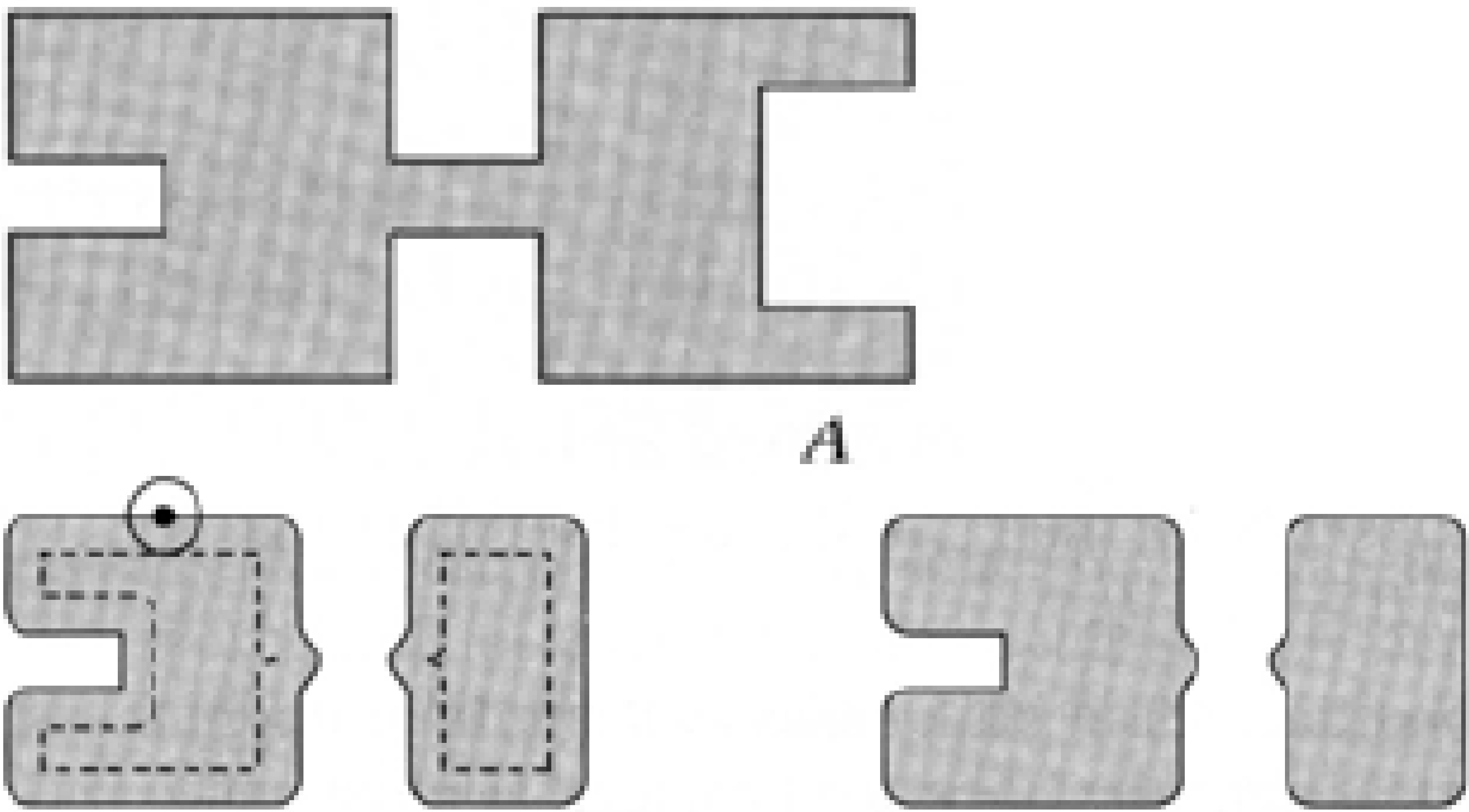
$A$

**Dilation:**
**- Object thickens**
**- Intrusions reduce in size if survived**
**- Protrusions increase in size if survived**
**- Outer corners round (Disk SE).**

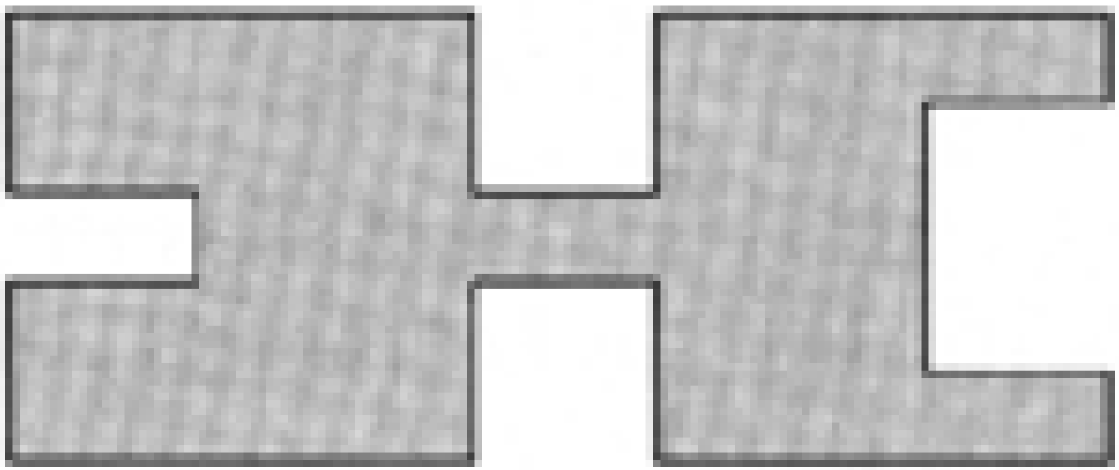


$A \oplus B$

# Examples – (cont.)

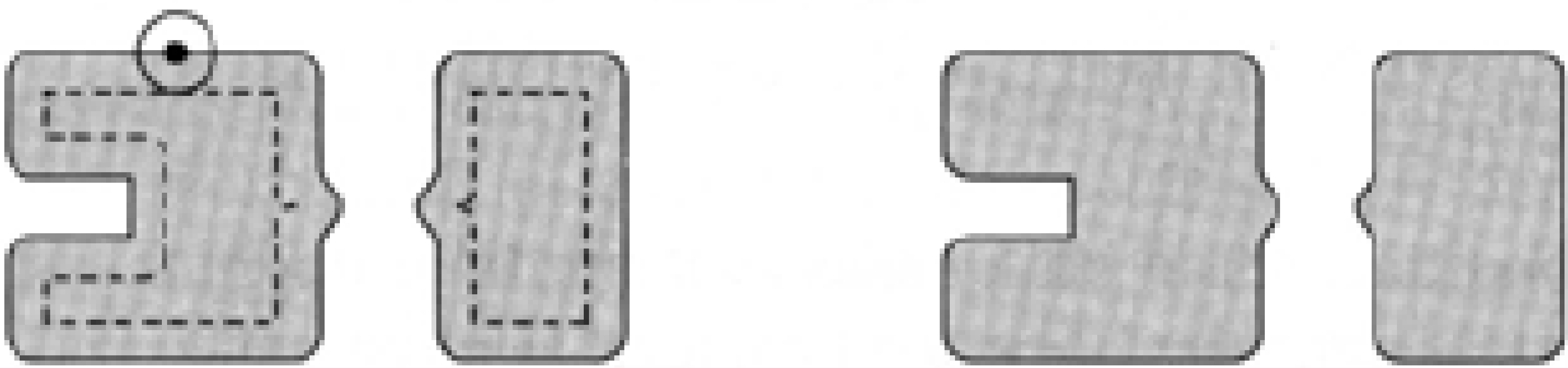


# Examples – (cont.)



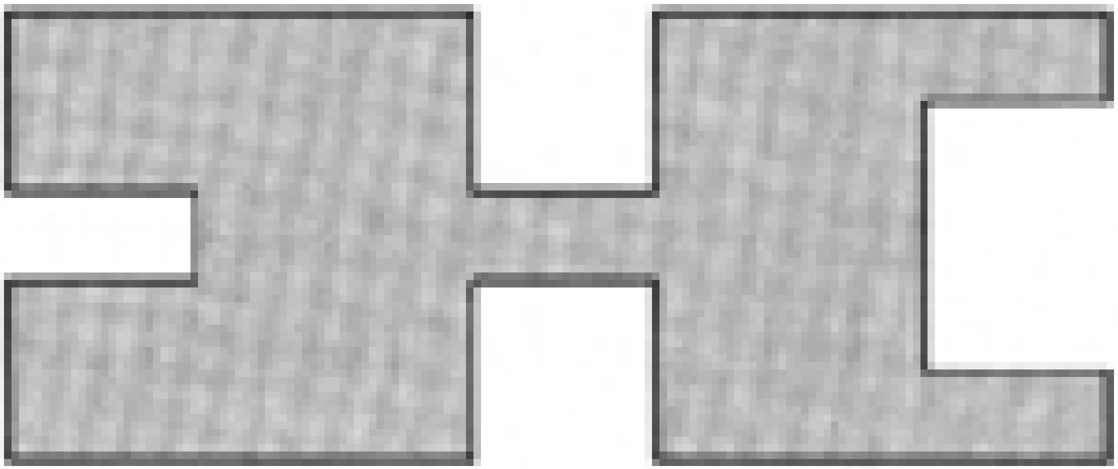
$A$

**Opening:**
- Outward pointing corners → rounded
- Inward pointing corners → unchanged if survived
- Smoothing effect

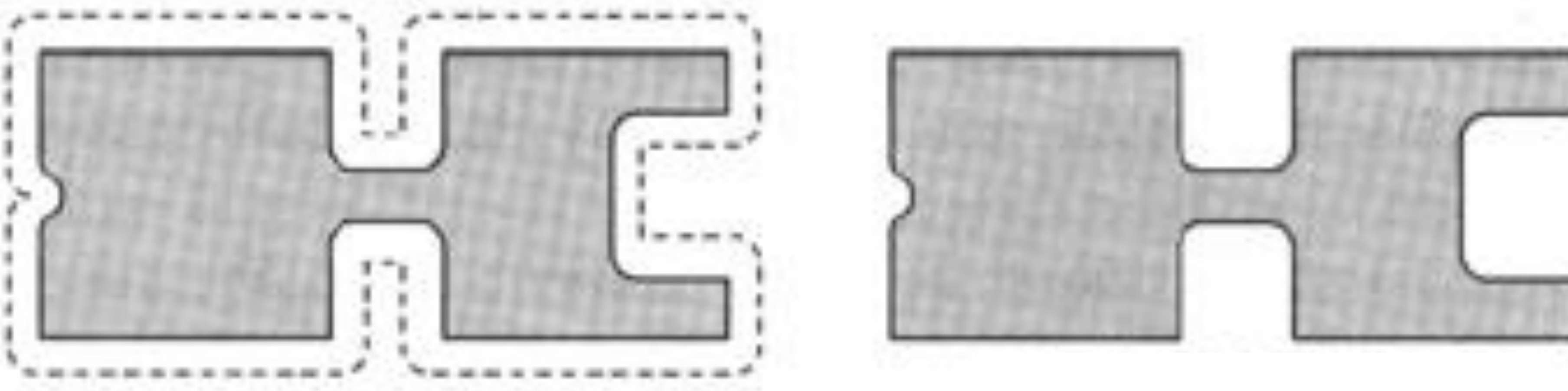


$$A \circ B = (A \ominus B) \oplus B$$

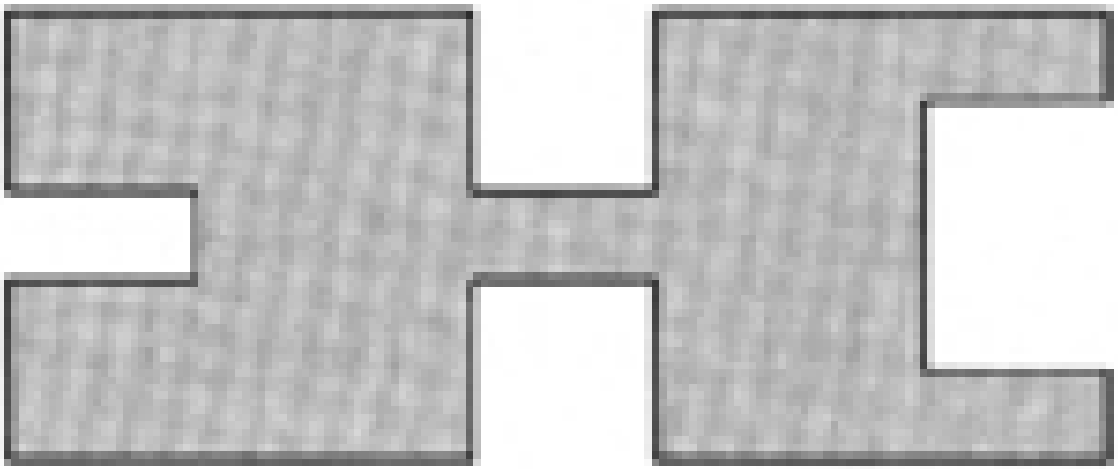
# Examples – (cont.)



$A$



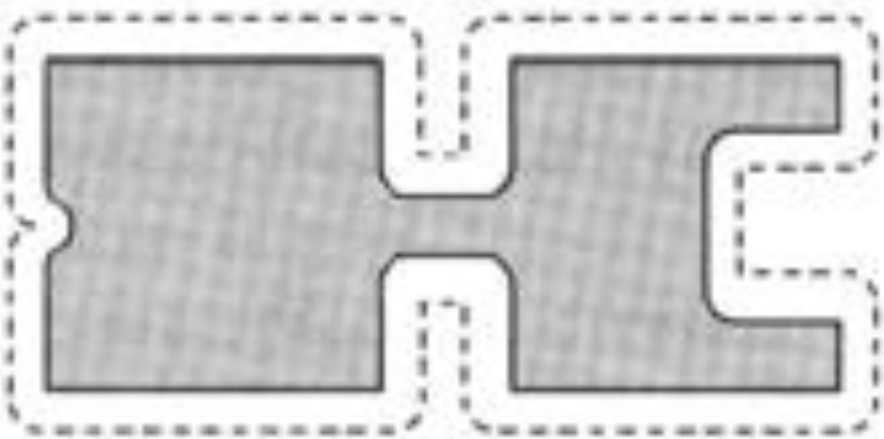
# Examples – (cont.)

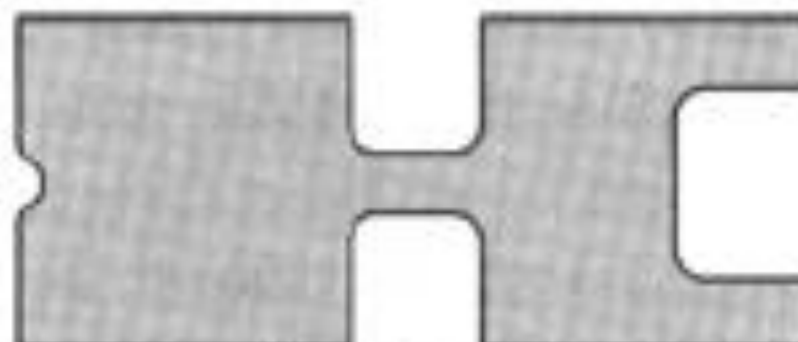


$A$

**Closing:**

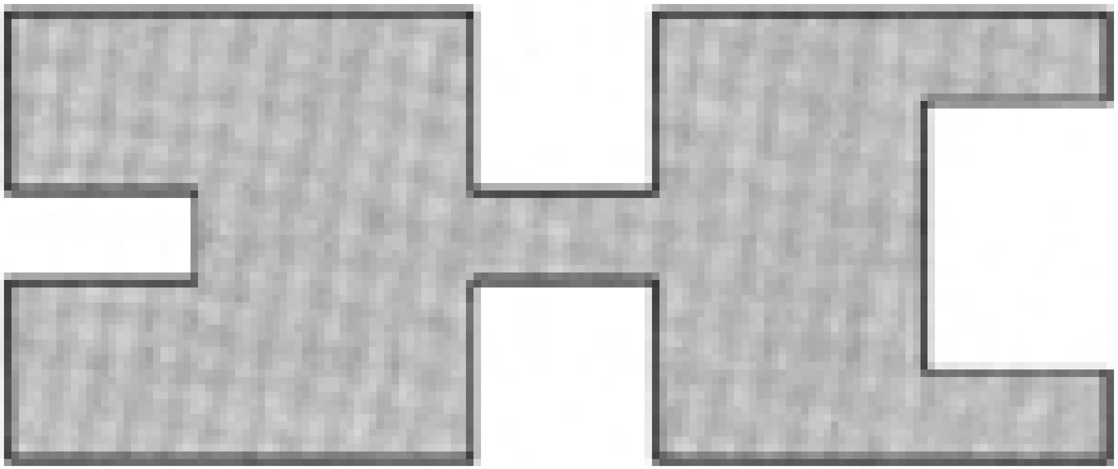
- **Inward pointing corners → rounded**
- **Outward pointing corners → unchanged**
- **Intrusions reduces/breaks if B doesn't fit**
- **Smoothing effect**



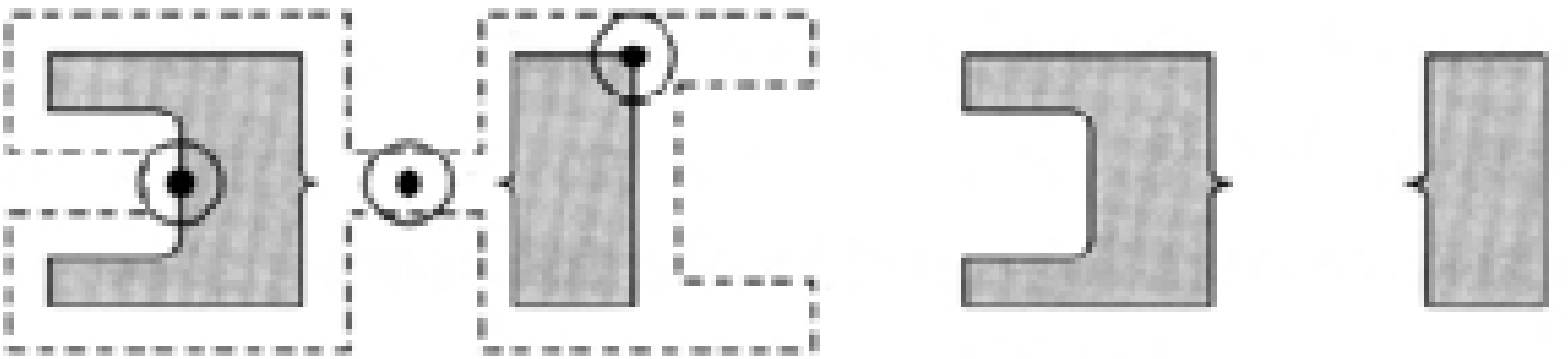


$A \bullet B = (A \oplus B) \ominus B$

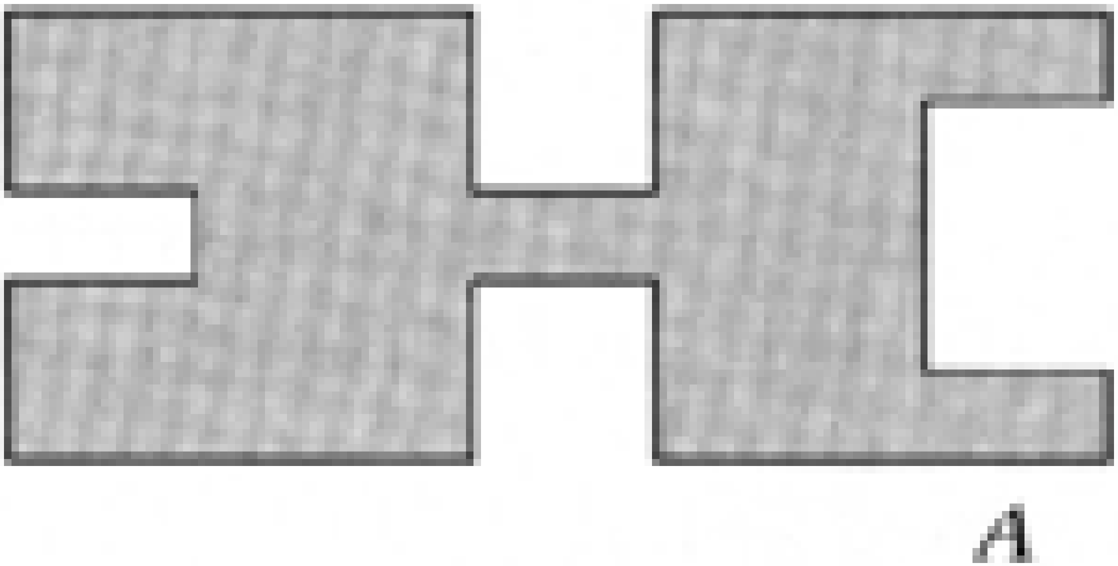
# Examples – (cont.)



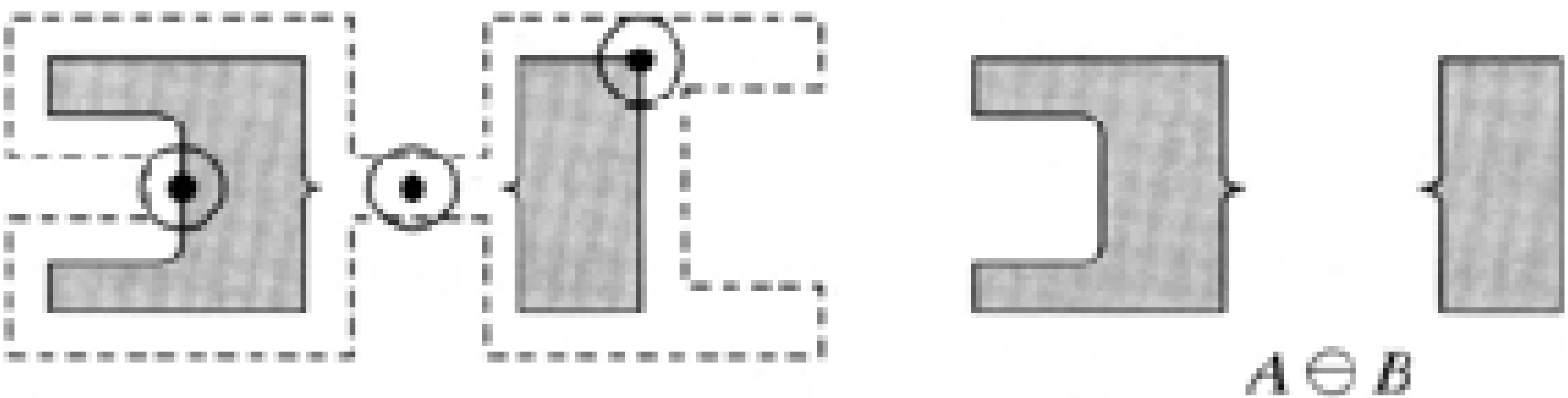
A



# Examples – (cont.)




**Erosion:**
- **Object shrinks by SE size**
- **Protrusions where B didn't fit were eliminated**



# Some Basic Morphological Algorithms

- Using the simple techniques we have looked at so far we can begin to consider some more interesting morphological algorithms:
  - Hit-or-miss transform
  - Boundary Extraction
  - Connected components
  - Thinning/thickening
  - Pruning
  - Top-Hat
  - Hole Filling
  - Convex Hull
  - Skeletonziation

# Top-Hat Transform

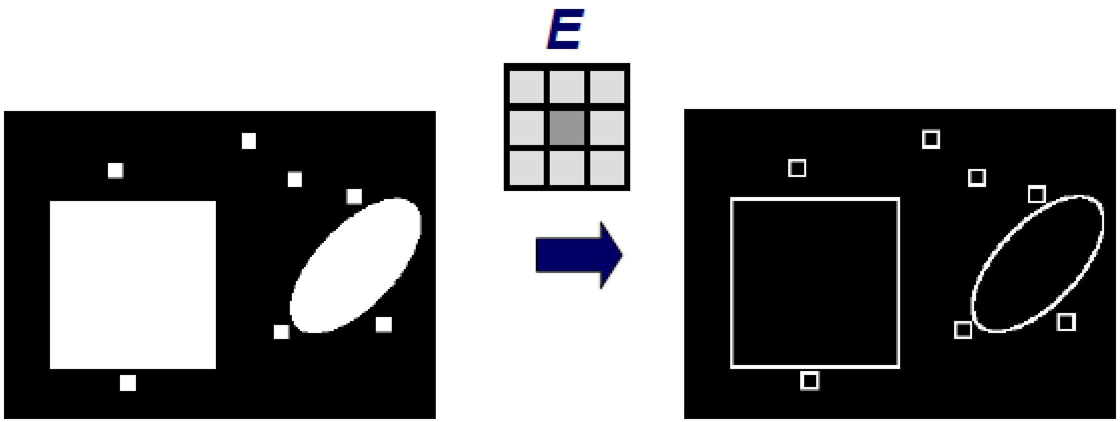
- **Top-hat transformation of an image X is defined as the difference between the original image X and its opening**
- $$TH = X - (X \circ SE)$$

$X$ $\quad$ $X \circ SE$ $\quad$ *Top Hat*

Y. Tarabalka, J. A. Benediktsson, J. Chanussot

# Morphological Gradient

- The basic morphological gradient of an image X is defined as the arithmetic difference between the dilation and the erosion of X by the elementary SE
- $$TH = (X \oplus SE) - (X \ominus SE)$$




Y. Tarabalka, J. A. Benediktsson, J. Chanussot

# Boundary Extraction

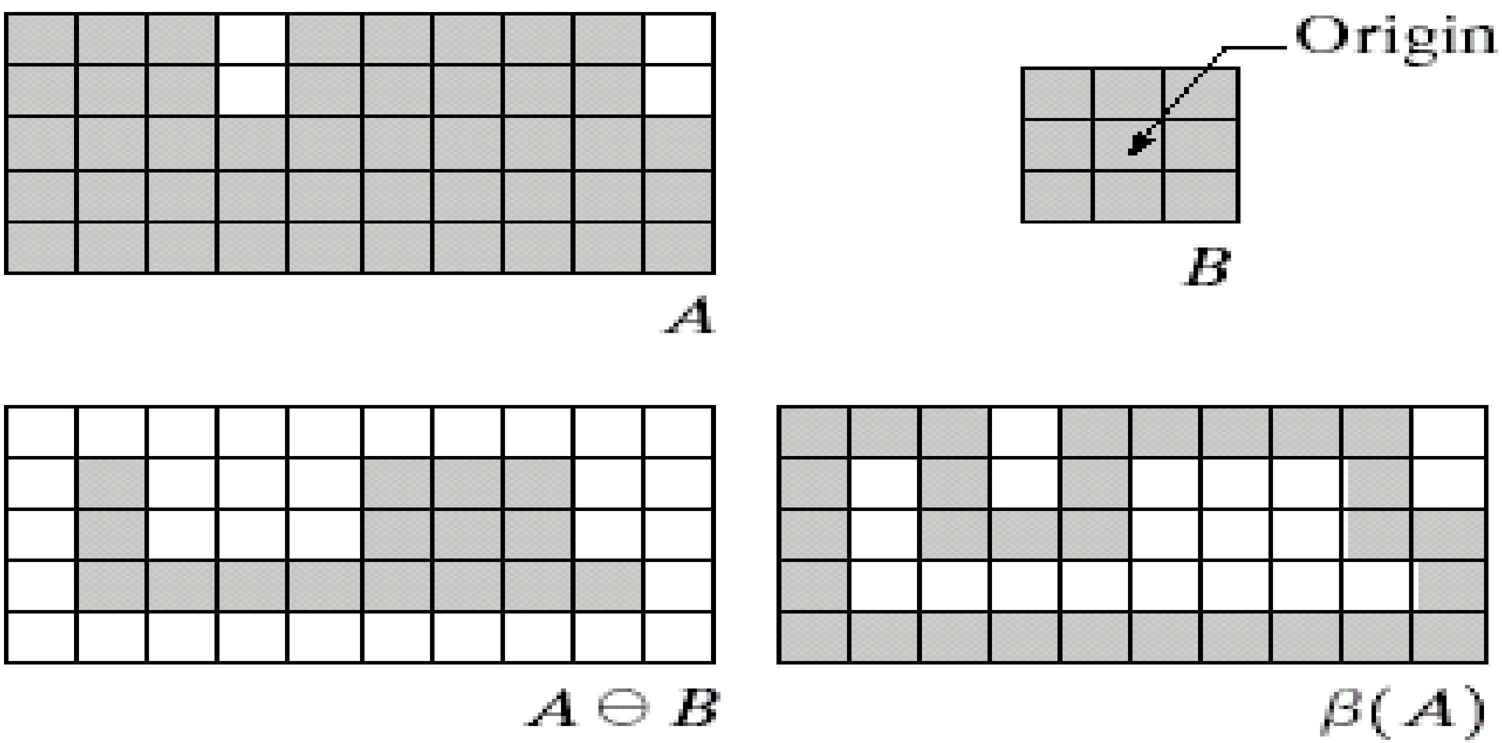
Original Image

Extracted Boundary

# Boundary Extraction – (cont.)

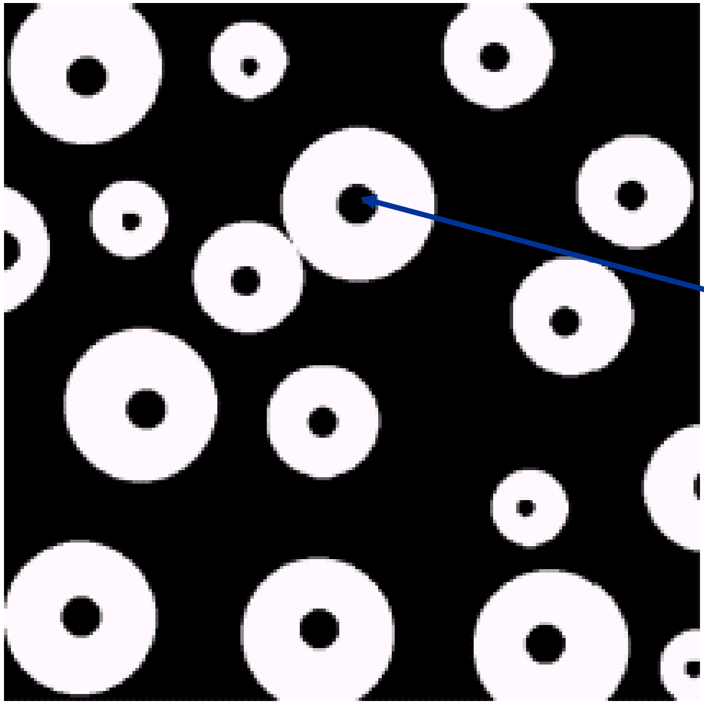
- **The boundary can be given simply as**

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



# Hole Filling

- **What is a hole?**
- **Given a pixel inside a boundary, *region filling* attempts to fill that boundary with object pixels (1s).**



Given a point inside here, can we fill the whole circle?

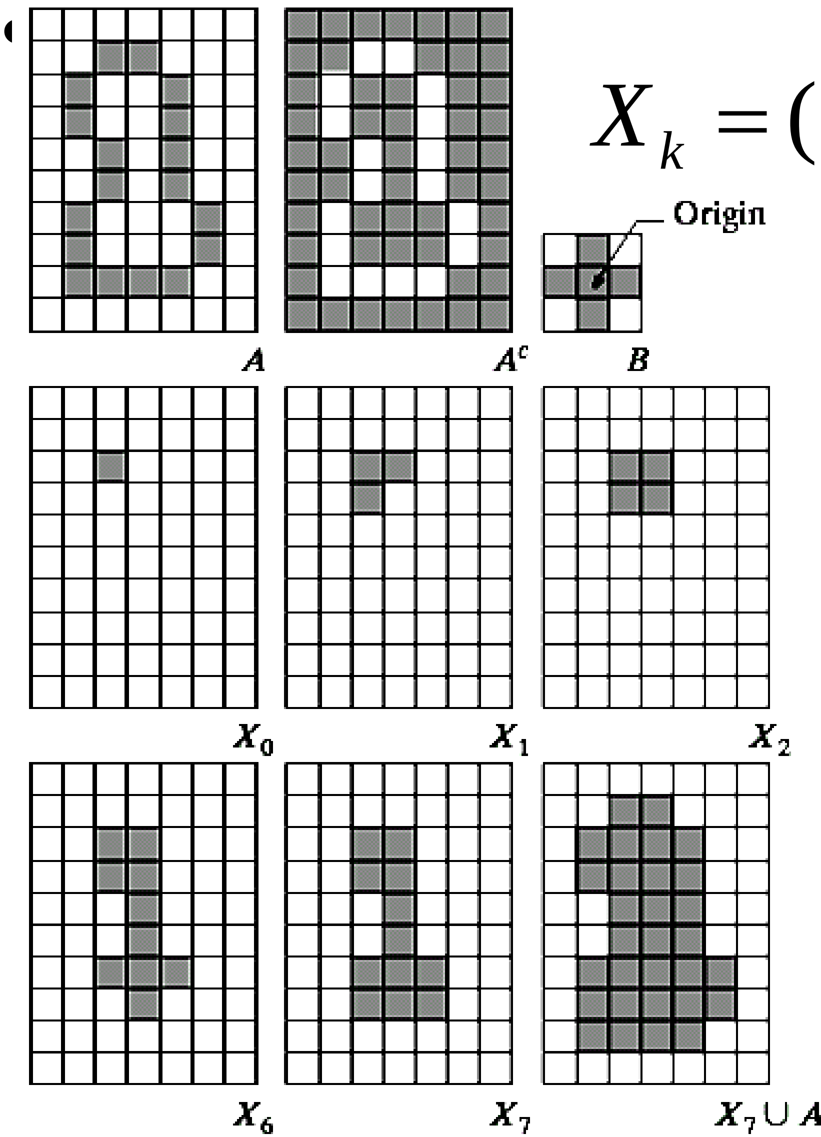
# Hole Filling – (cont.)

- **The key equation for region filling is:**

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3.....$$

- **Where $X_0$ is a zero image except the starting point inside the holes, $B$ is a simple SE, and $A^c$ is the complement of $A$.**
- **This equation is applied repeatedly until $X_k = X_{k-1}$.**
- **Finally the result is unioned with the original boundary.**

# Hole Filling – (cont.)

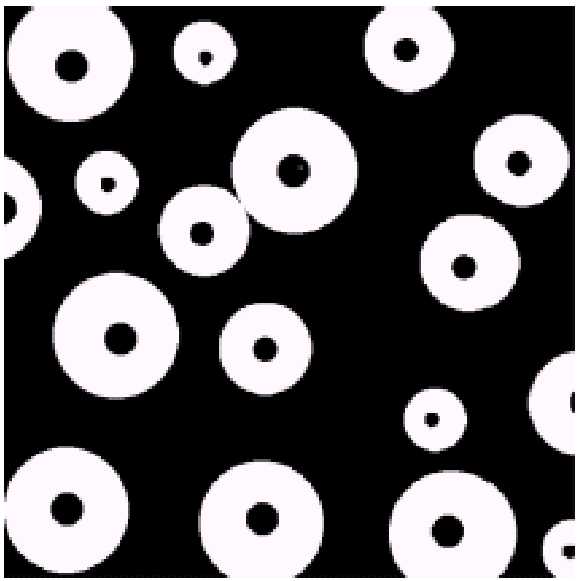



$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3\ldots..$$

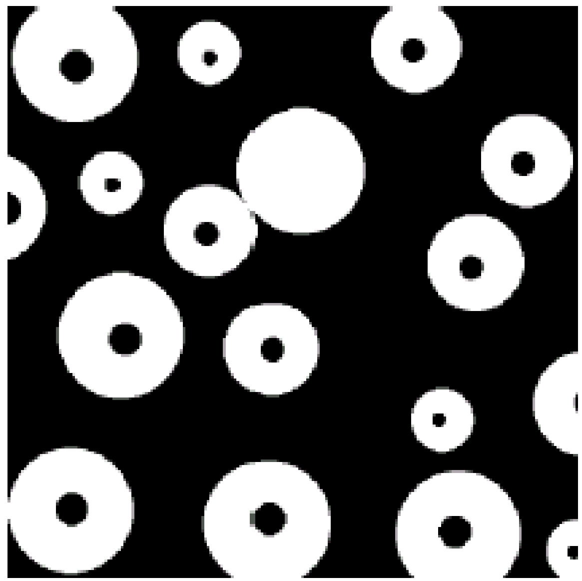
**The intersection with $A^c$ limits the result inside the region of interest.**

**Note here that the foreground is dark and the background is white.**

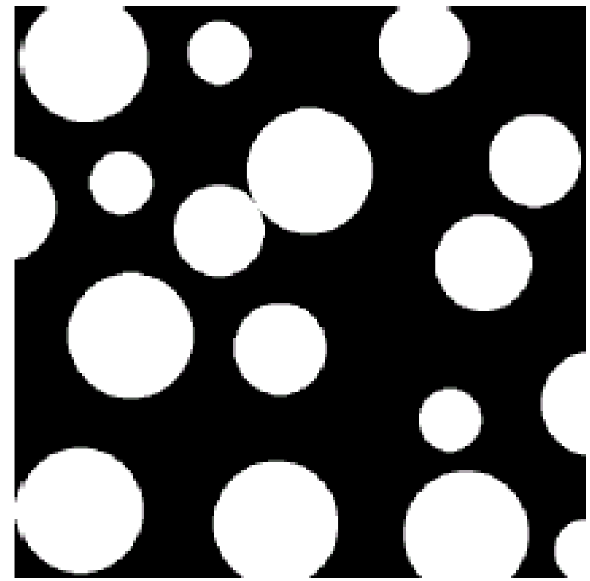
# Hole Filling – (cont.)



Original Image



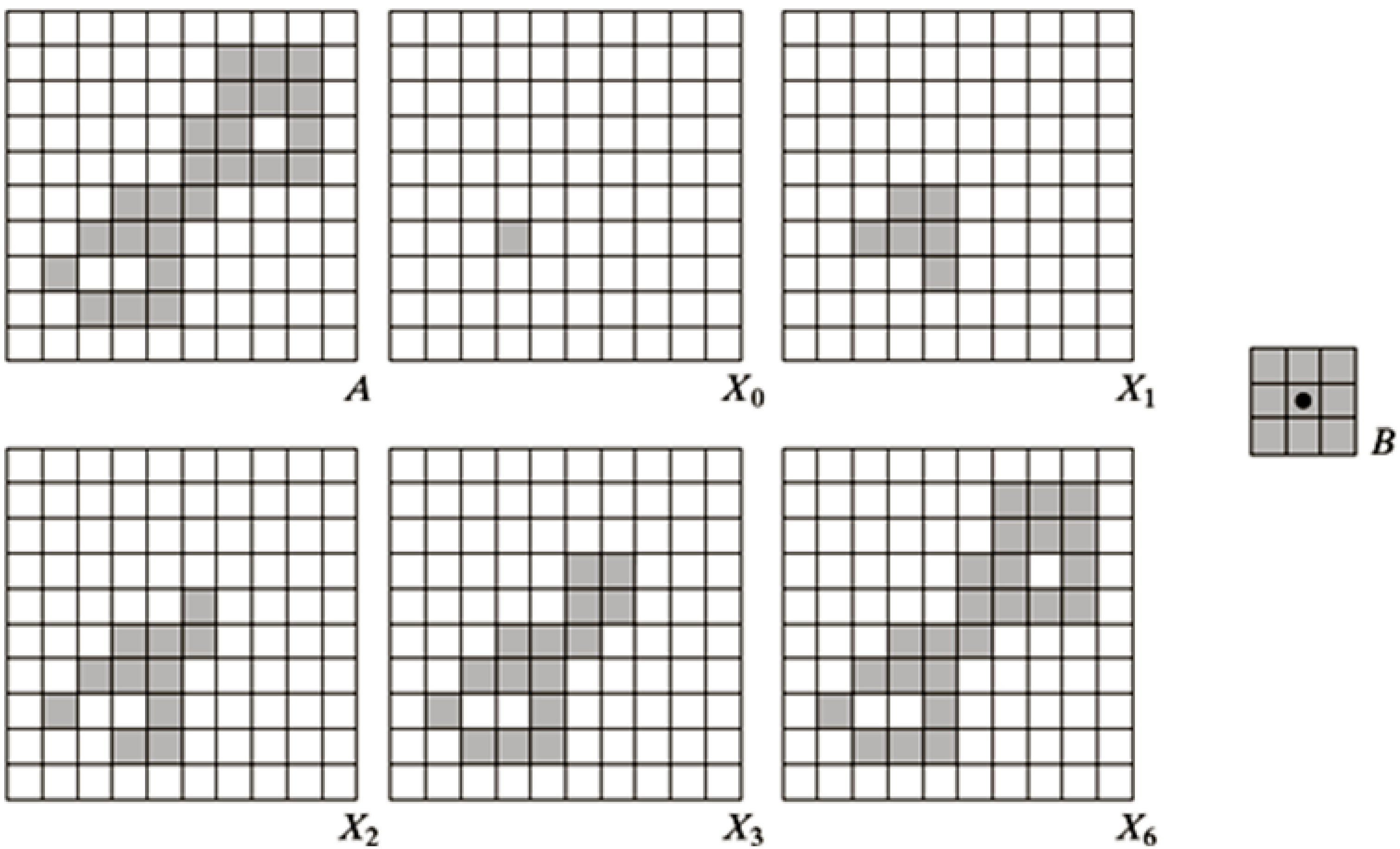
One Region Filled



All Regions Filled

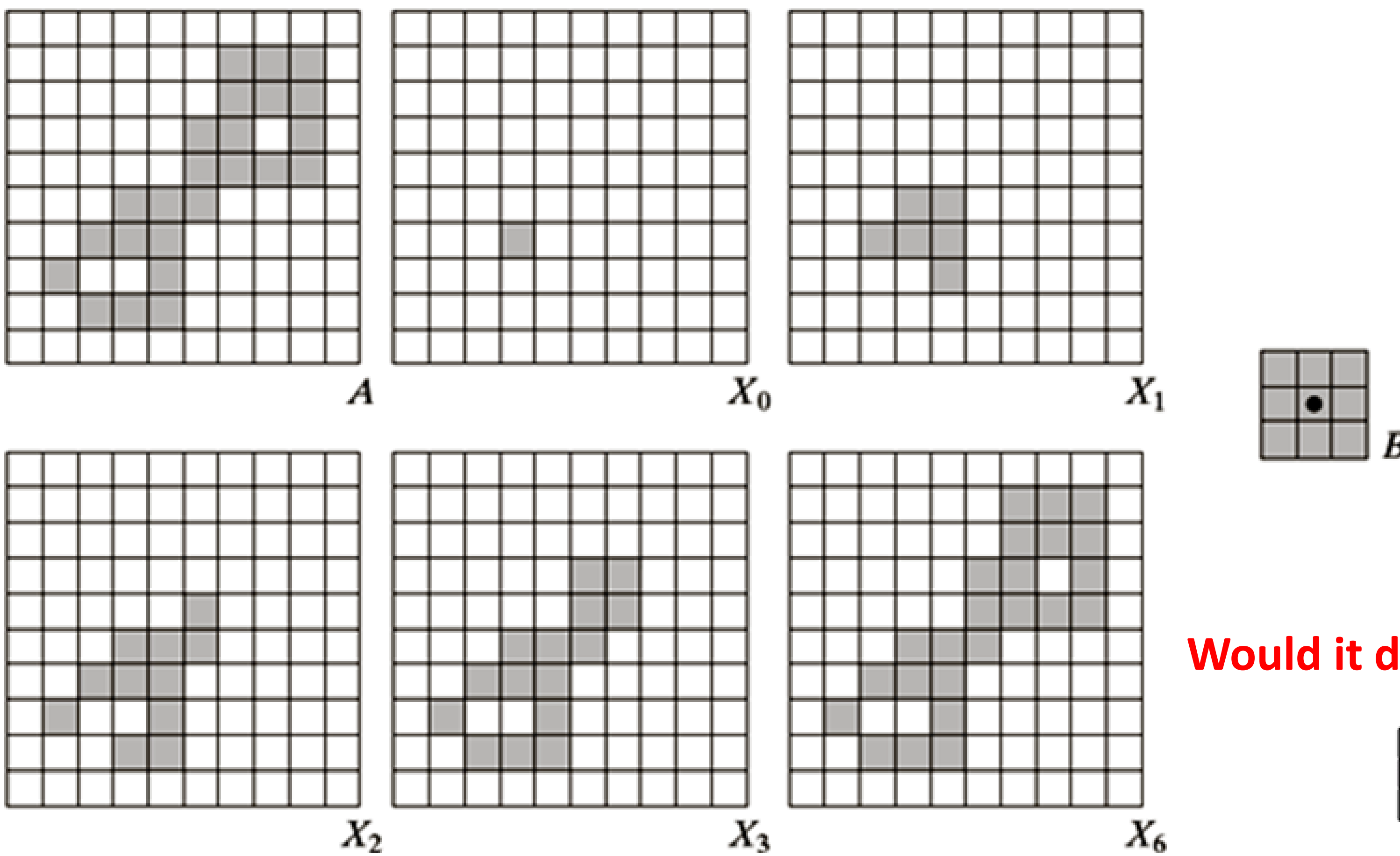
# Connected Components Extraction

$$X_k = (X_{k-1} \oplus B) \cap A \quad k = 1,2,3.....$$

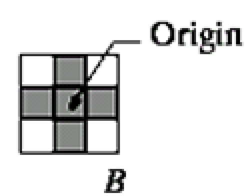


# Connected Components Extraction

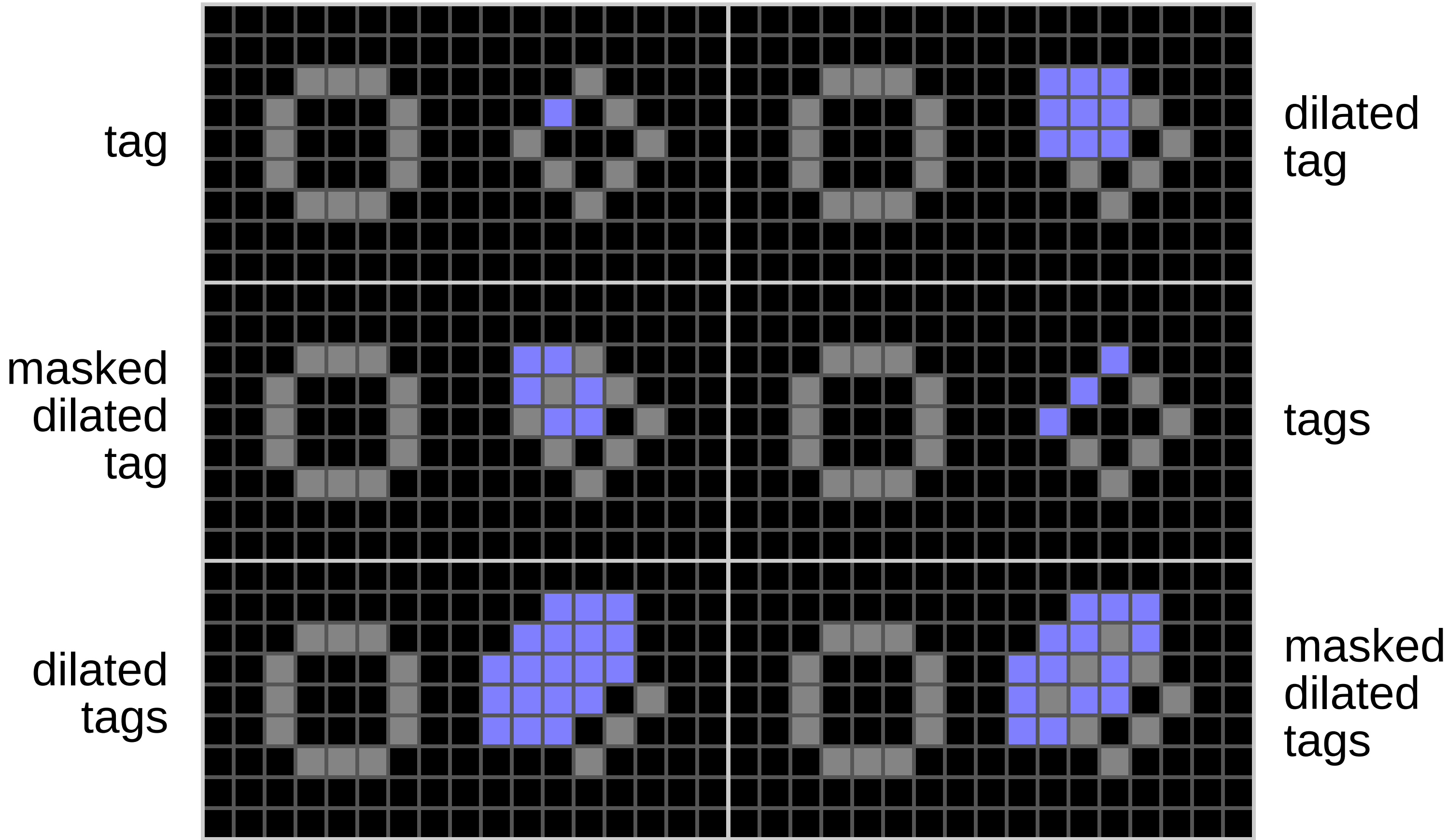
$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1,2,3.....$$



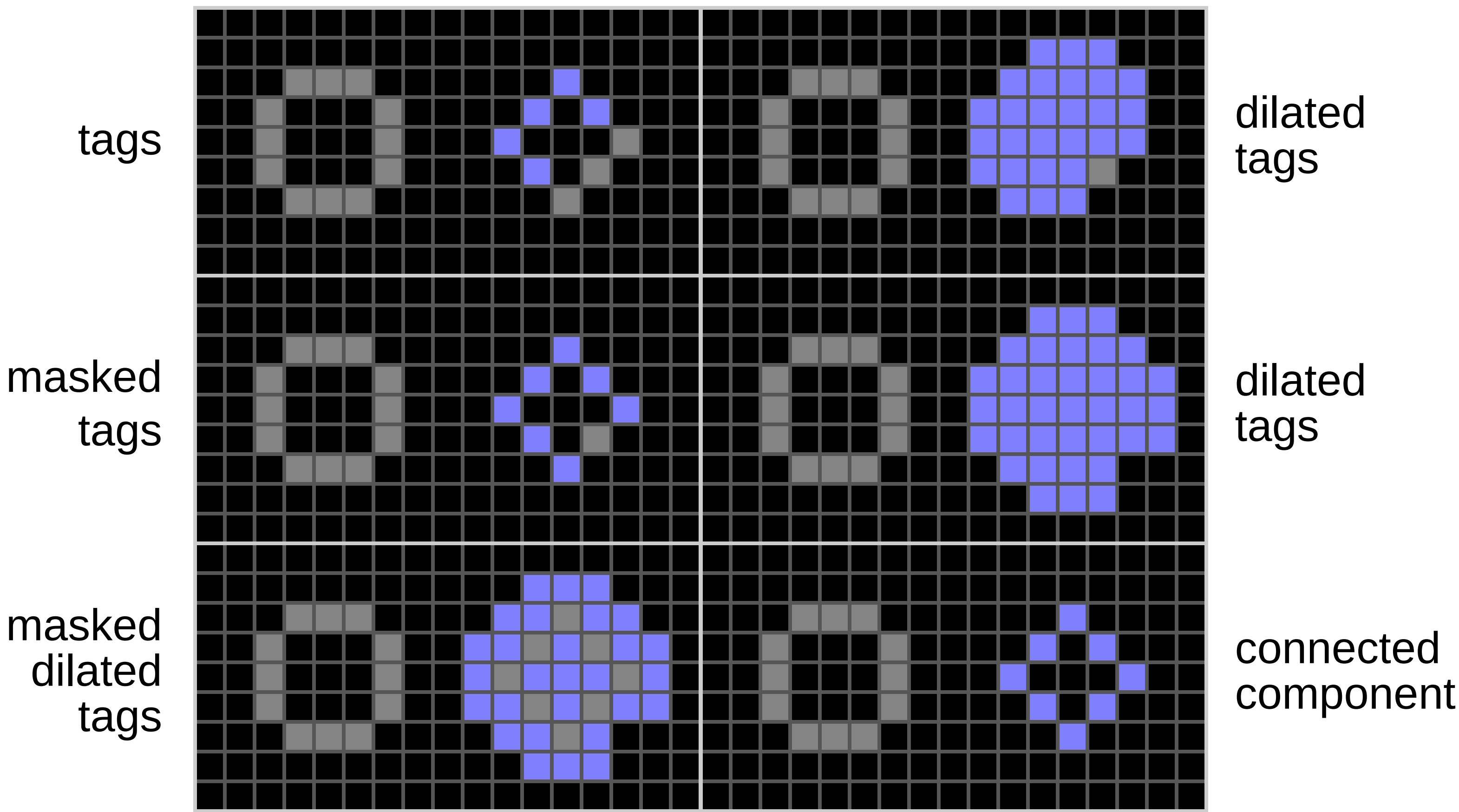

**Would it differ if we used**



# Connected Components Extraction

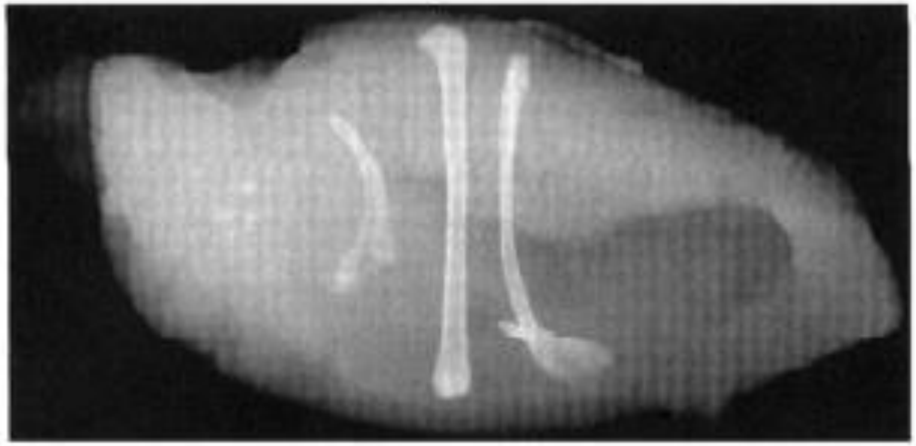


# Connected Components Extraction



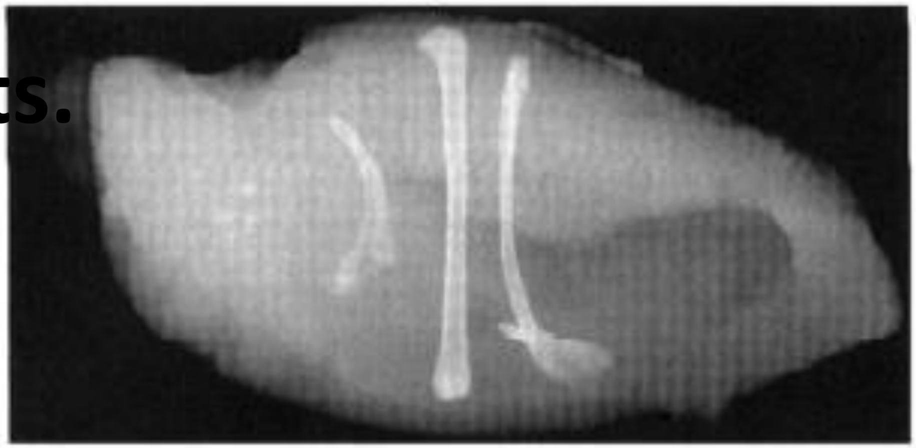
# Connected Components Extraction

- Automated inspection of processed food
- X-ray image of chicken filet.
- Identify bone fragments.

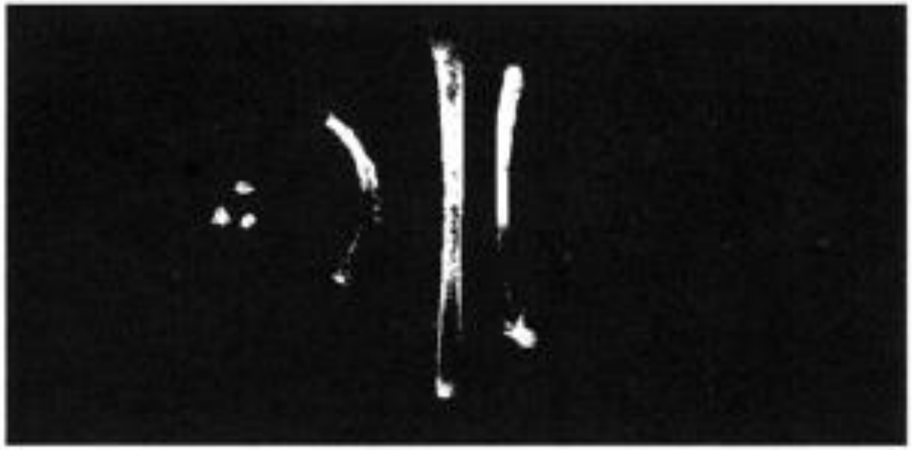


# Connected Components Extraction

- **Automated inspection of processed food**
- **X-ray image of chicken filet.**
- **Identify bone fragments.**

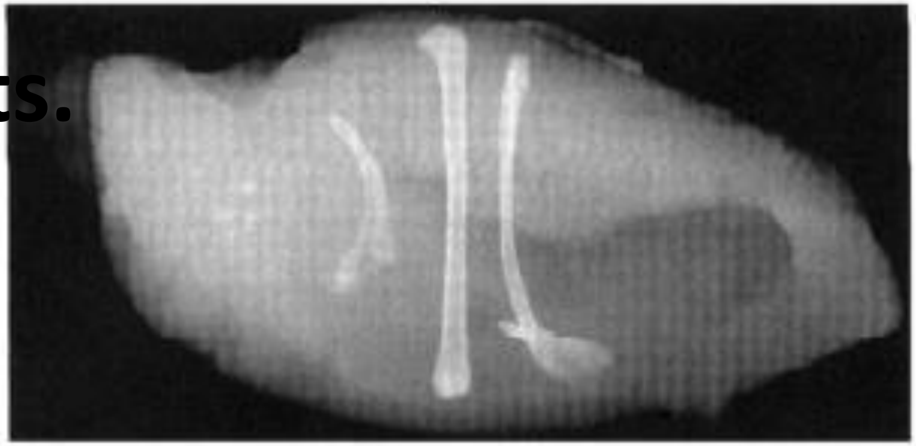


1. Thresholded image

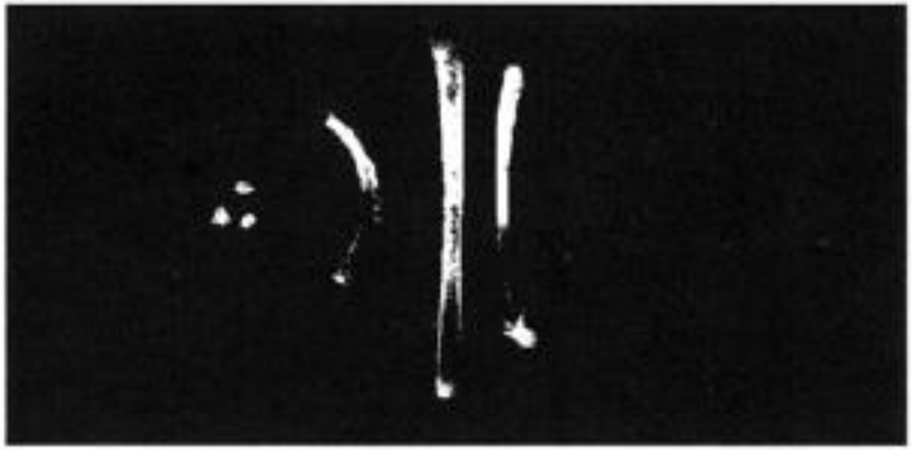


# Connected Components Extraction

- Automated inspection of processed food
- X-ray image of chicken filet.
- Identify bone fragments.

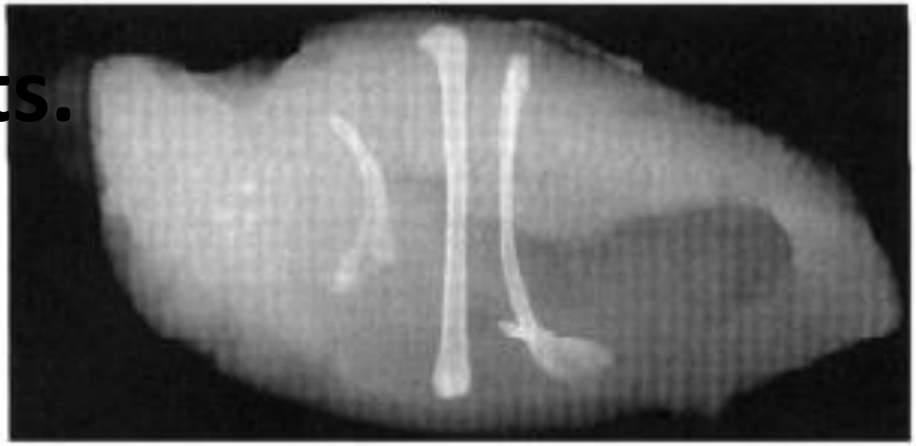


1. Thresholded image
2. Eroded image (size?)

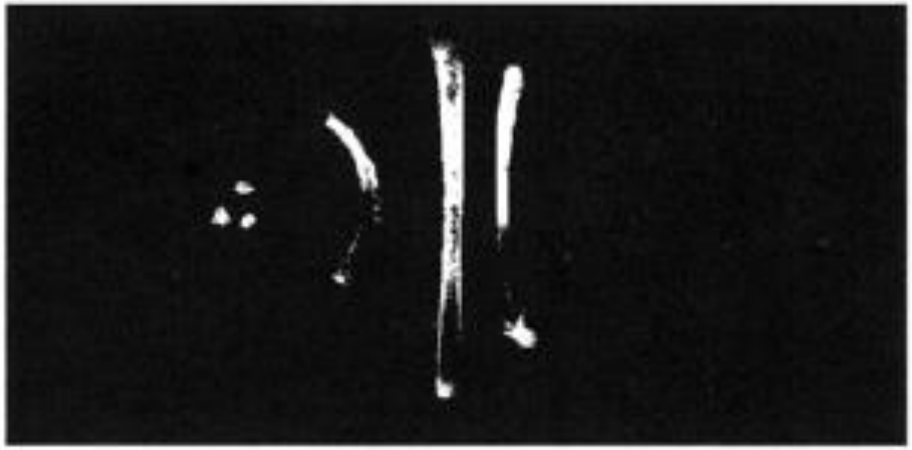


# Connected Components Extraction

- **Automated inspection of processed food**
- **X-ray image of chicken filet.**
- **Identify bone fragments.**

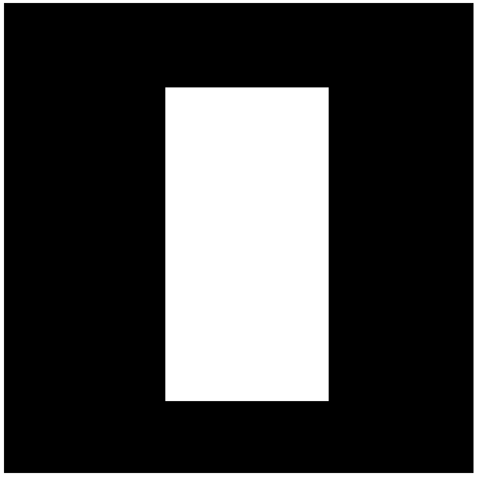


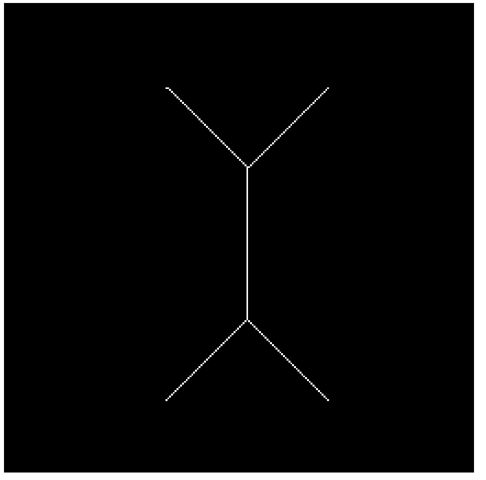
1. Thresholded image
2. Eroded image (size?)
3. Size of connected component



# Helpful Reading

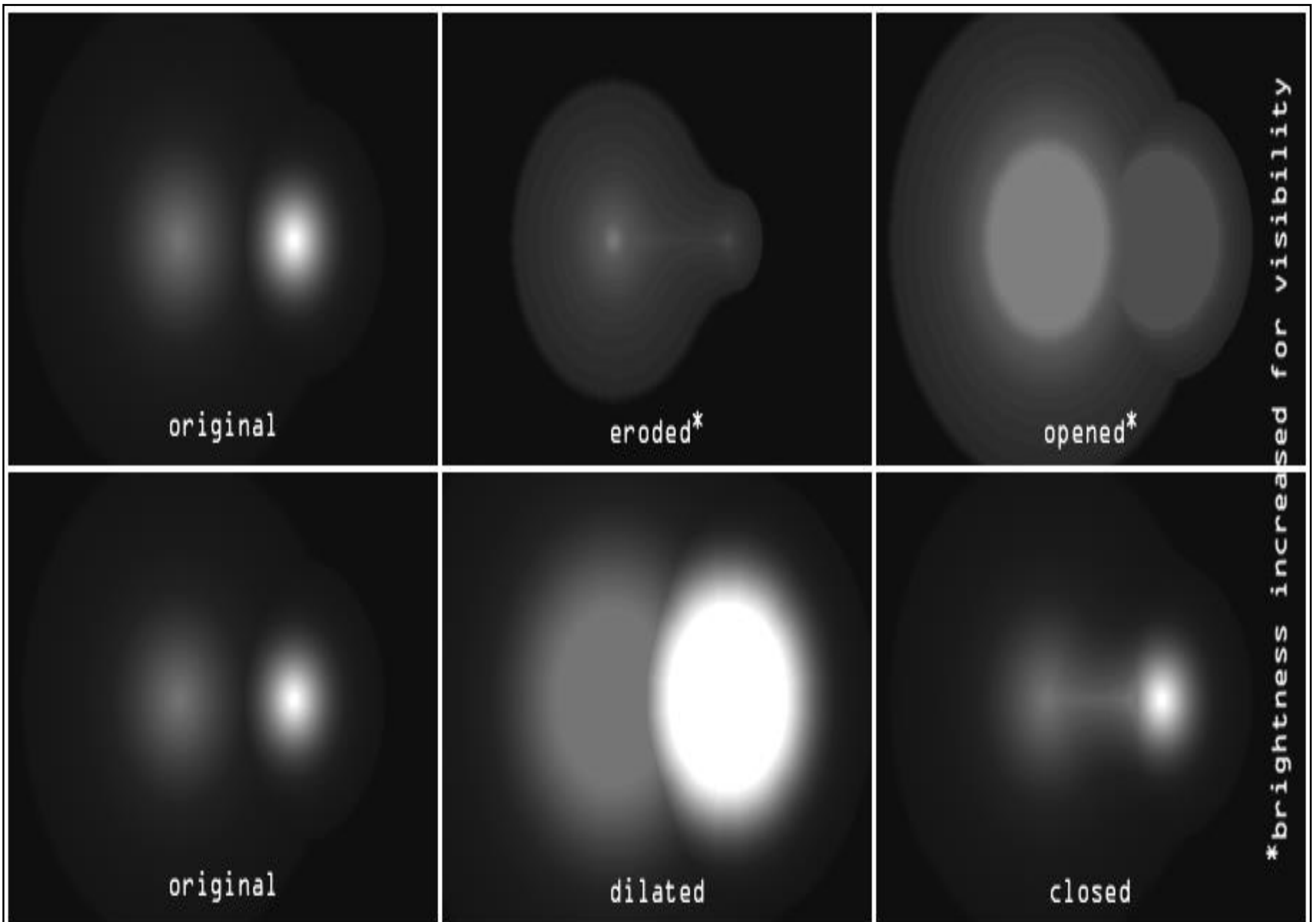
- Convex Hull
- Thinning/Thickening
- Sekeletonization
- Pruning
- MR: morphological reconstruction





# Gray Scale Morphology

## Basic Operations



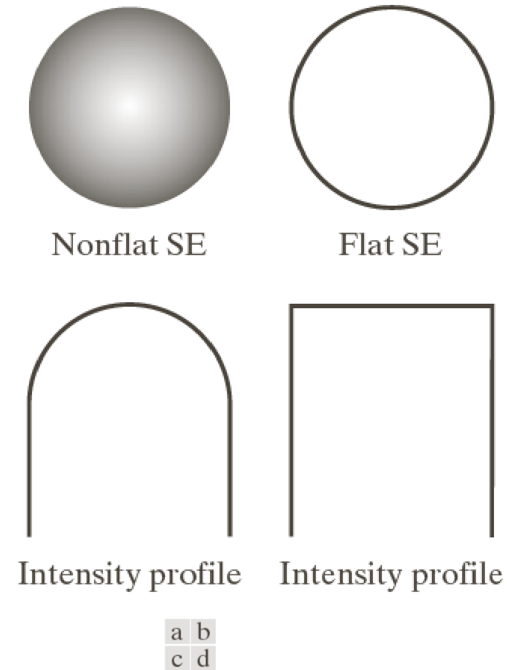




**FIGURE 9.34** Nonflat and flat structuring elements, and corresponding horizontal intensity profiles through their center. All examples in this section are based on flat SEs.

# Gray Scale Morphology – (cont.)

## Dilation

SE, **Z**, is a flat disk.

# Gray Scale Morphology – (cont.)

## Erosion

SE, **Z**, is a flat disk.

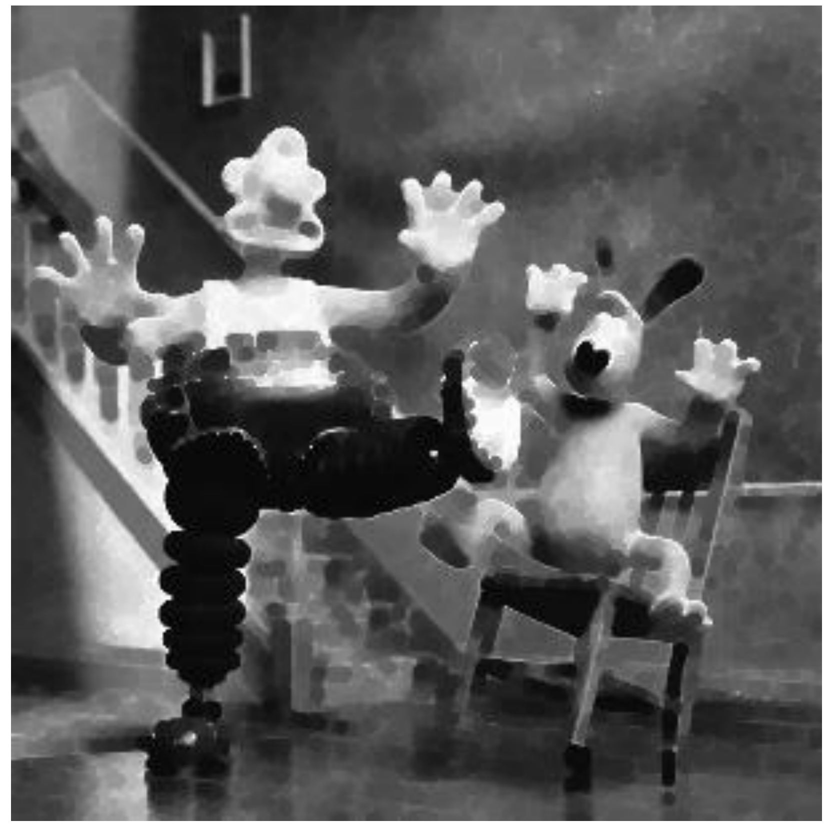
# Gray Scale Morphology – (cont.)

## Opening

SE, **Z**, is a flat disk.

# Gray Scale Morphology – (cont.)

## Closing



SE, **Z**, is a flat disk.

# Next Lecture

Revision + Quiz 1

# Assignment

Check book sections and associated problems

| | |
|---|---|
| Chapter 9 | 1, 2, 3, 5(1,2,3) |
| Associated problems | 1, 5, 6, 7, 8, 17, 18, 20, 21, 22, 24, 36, 37 |

# References

- Gonzalez and Woods, Digital Image Processing.

- Peters, Richard Alan, II, “Image Morphology", Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008, Available on the web at the Internet Archive, http://www.archive.org/details/Lectures_on_Image_Processing.

-Yuliya Tarabalka, Jón Atli Benediktsson, Jocelyn Chanussot, “Mathematical morphology”, presentation, University of Iceland/Grenoble Institute of Technology, France.

- Qigong Zheng, “Mathematical Morphology”, University of Maryland College Park.