# *Morphological Image Processing*

李东晓

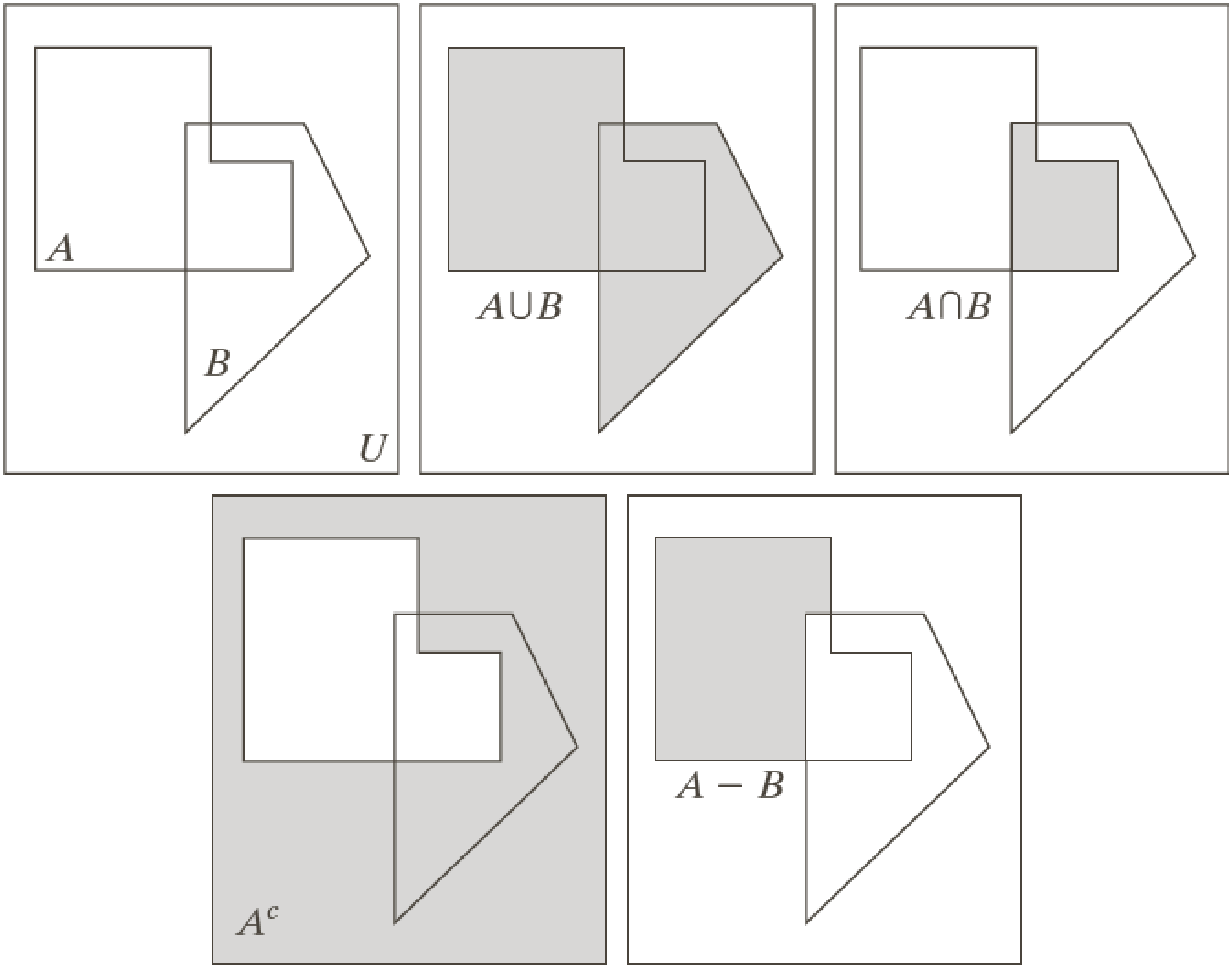
lidx@zju.edu.cn



浙江大学信息与电子工程学院
College of Information Science & Electronic Engineering, Zhejiang University

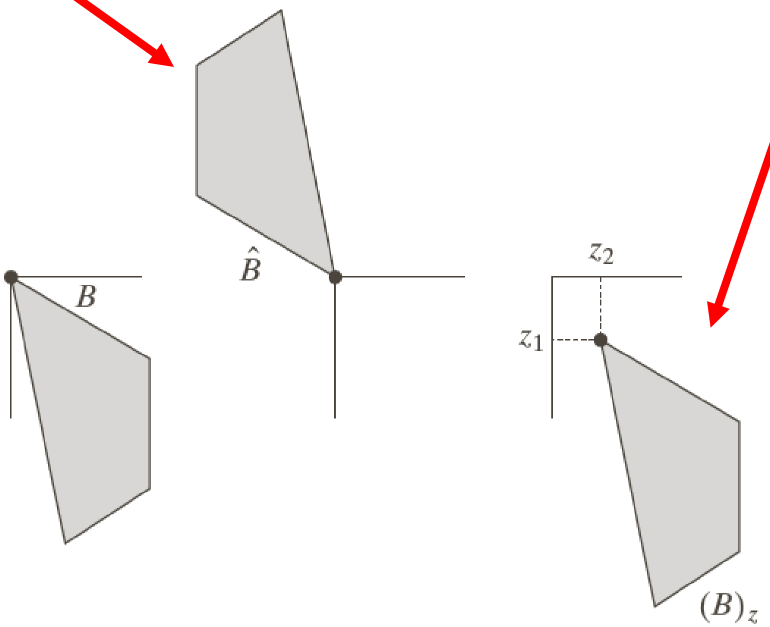
# Contents

# Preliminaries – Set Theory



# Preliminaries – Set Theory

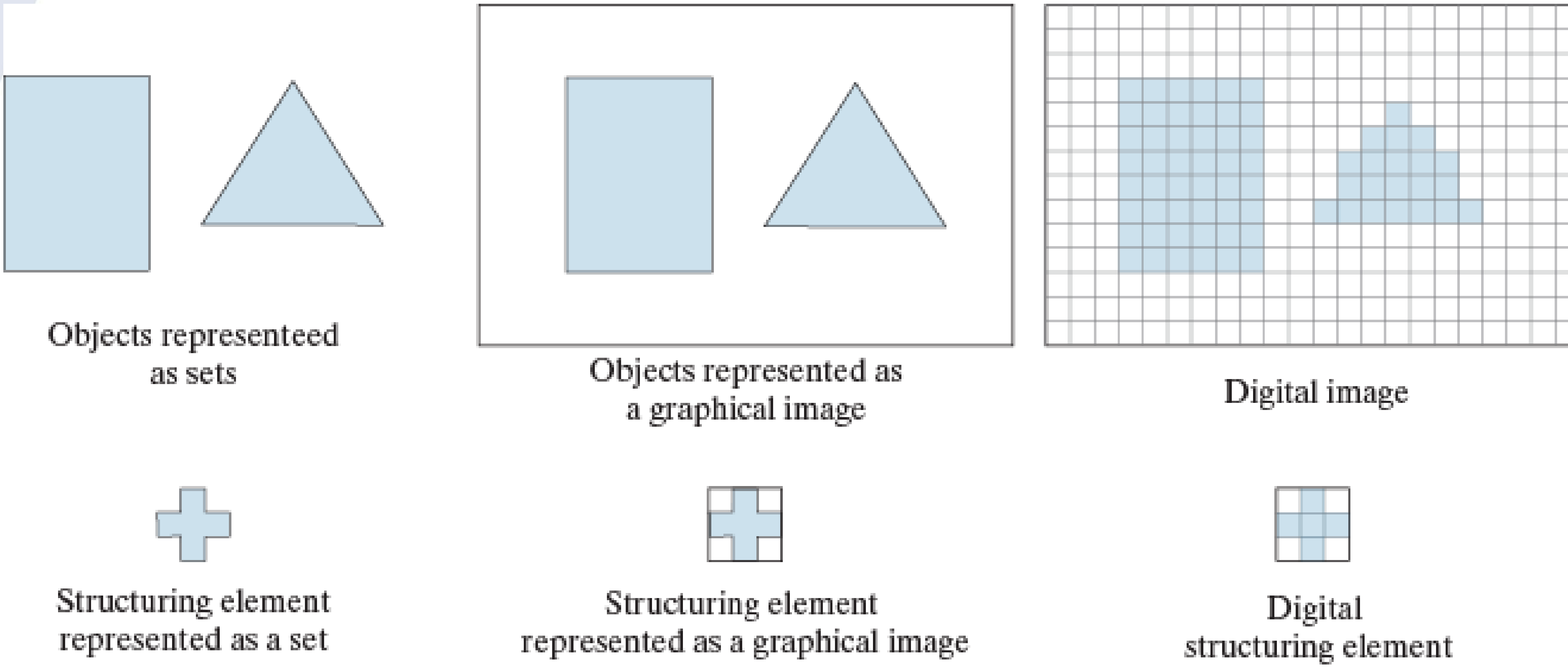
- Binary image: $Z^2$, set of (x, y) coordinates
- Gray-scale image: $Z^3$, set of (x, y, f(x, y))
- Set reflection $\hat{B} = \{w \mid w = -b, \quad \text{for} \quad b \in B\}$
- Set translation $(B)_z = \{c \mid c = b + z, \quad \text{for} \quad b \in B\}$



# What Is Binary Morphology?

- Morphological image processing describes a range of image processing techniques that deal with the shape (or morphology) of features in an image.
- Morphological operations are typically applied to remove imperfections introduced during segmentation, and so typically operate on binary images.
- Whether 0 and 1 refer to white or black is a little interchangeable

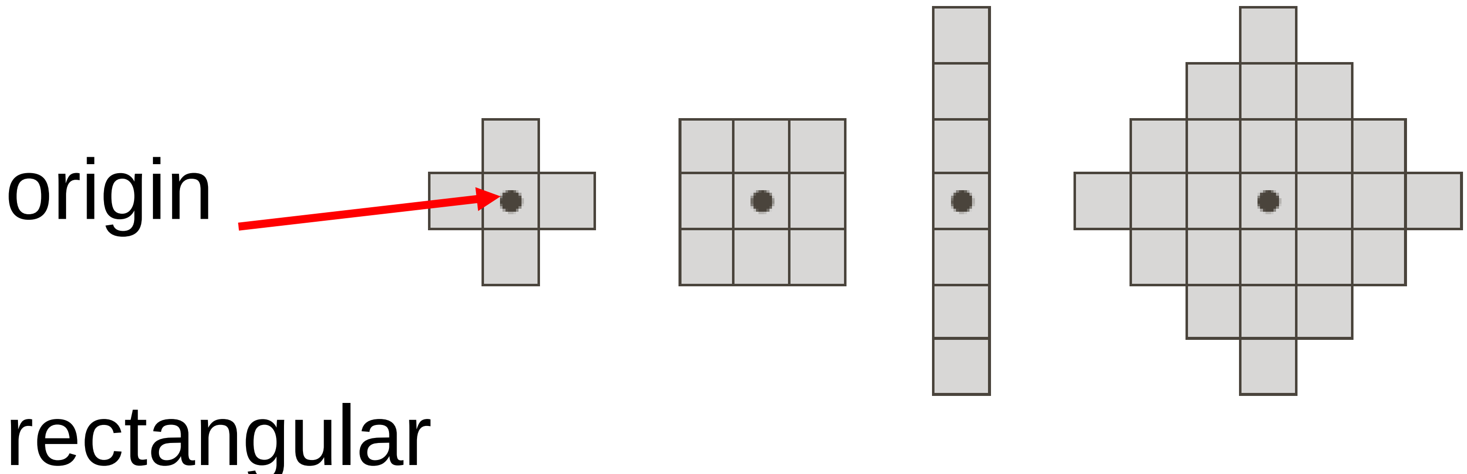
# Objects & Structure Elements

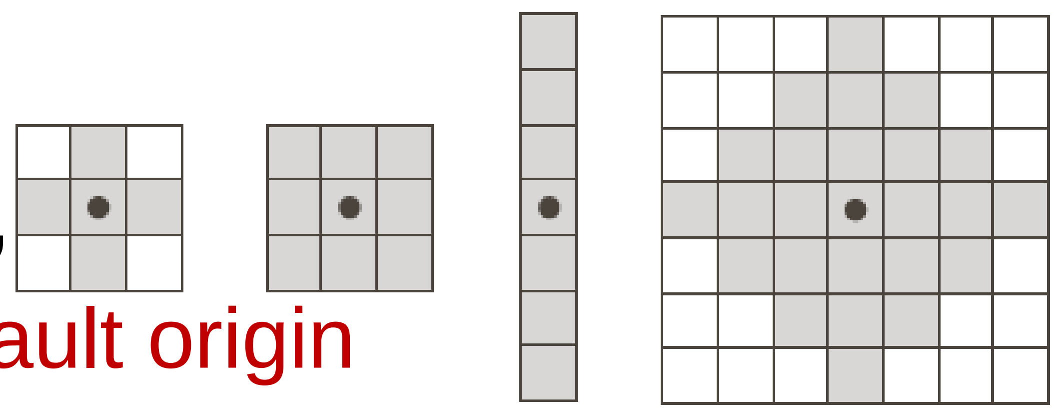


# Structuring Elements

- Structuring Element (SE):

small set or sub-image, any size and any shape

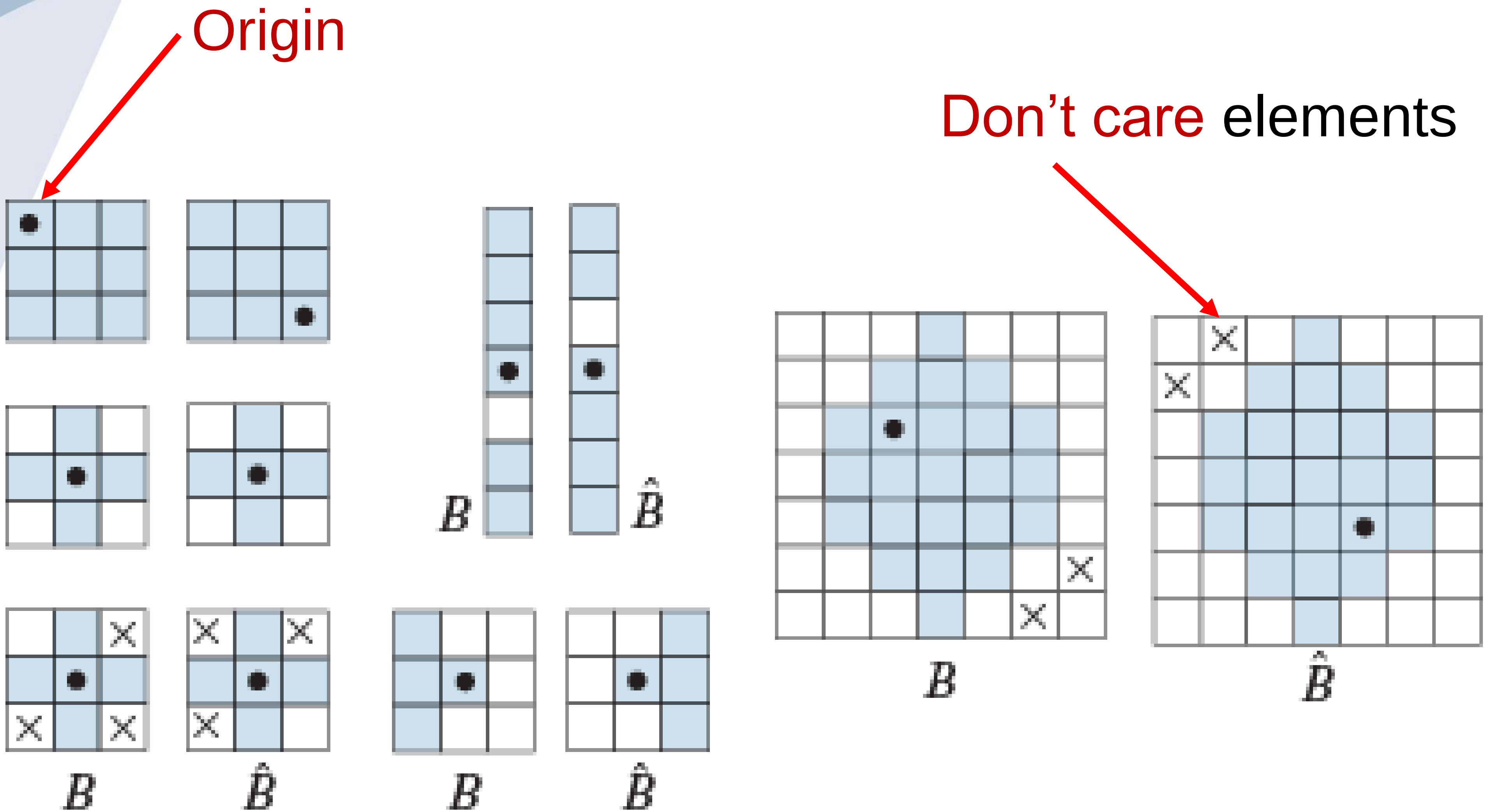


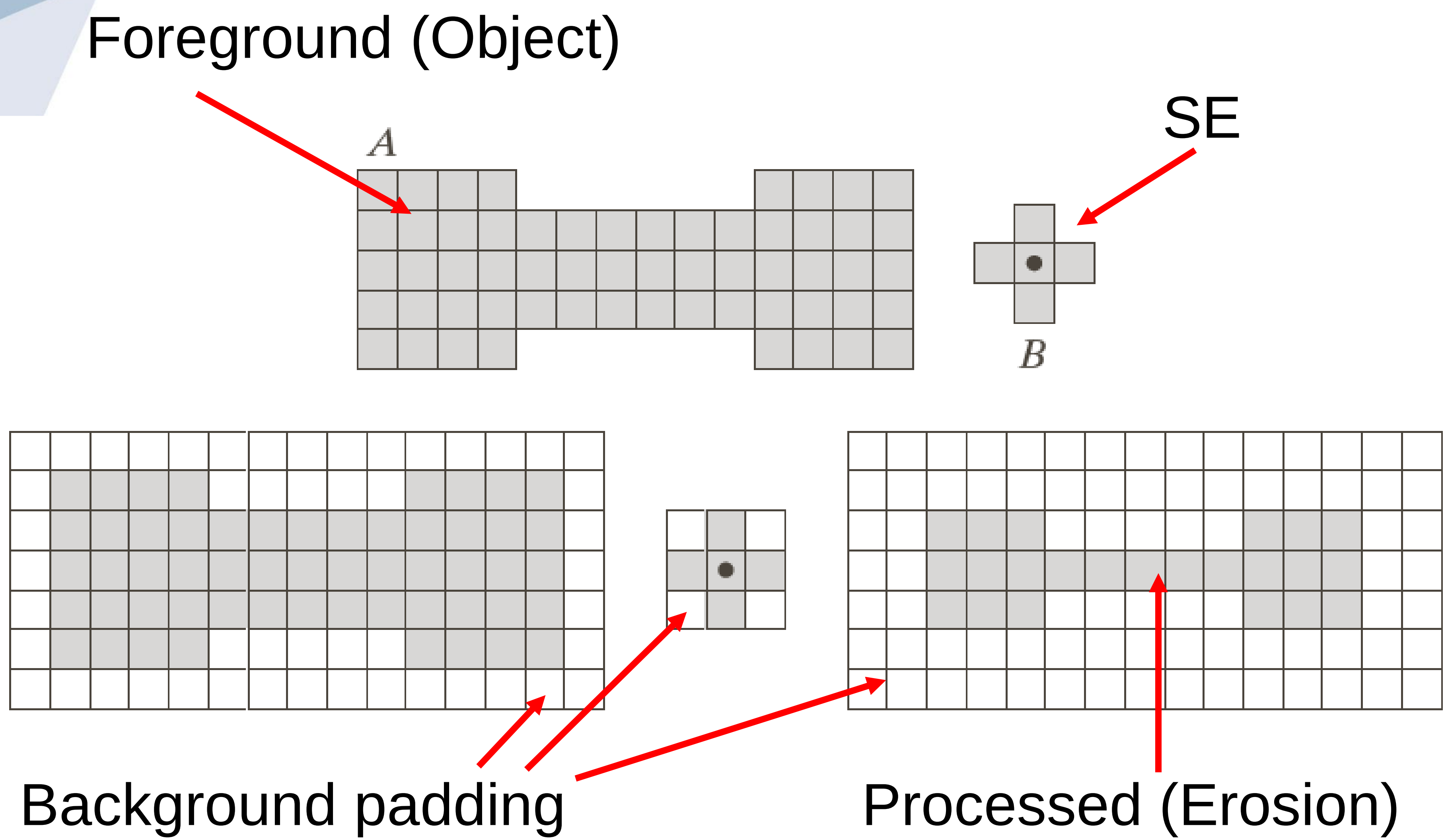
- For symmetric SE,

the center is the default origin

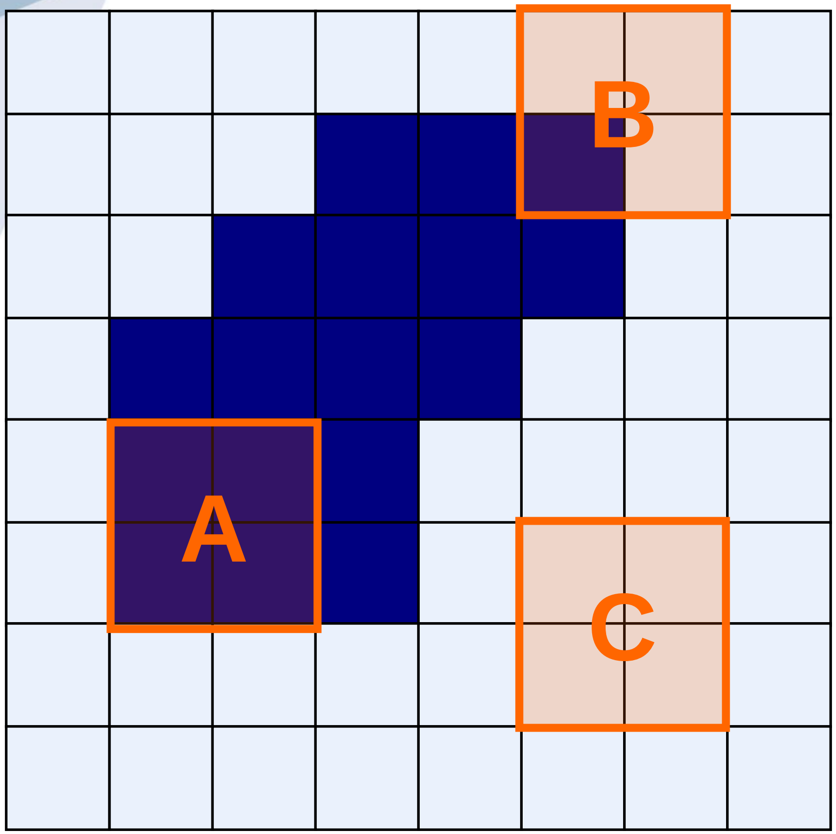
# Structuring Elements & Reflections

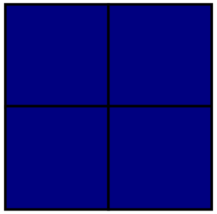


# Foreground and Background



# Structuring Elements: Hit, Fit, Miss



Structuring Element

**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

**Miss:** otherwise

All morphological processing operations are based on these simple ideas

# Fitting & Hitting

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | **B** 1 | 1 | 1 | 1 | 0 | **C** 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 | **A** 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 |

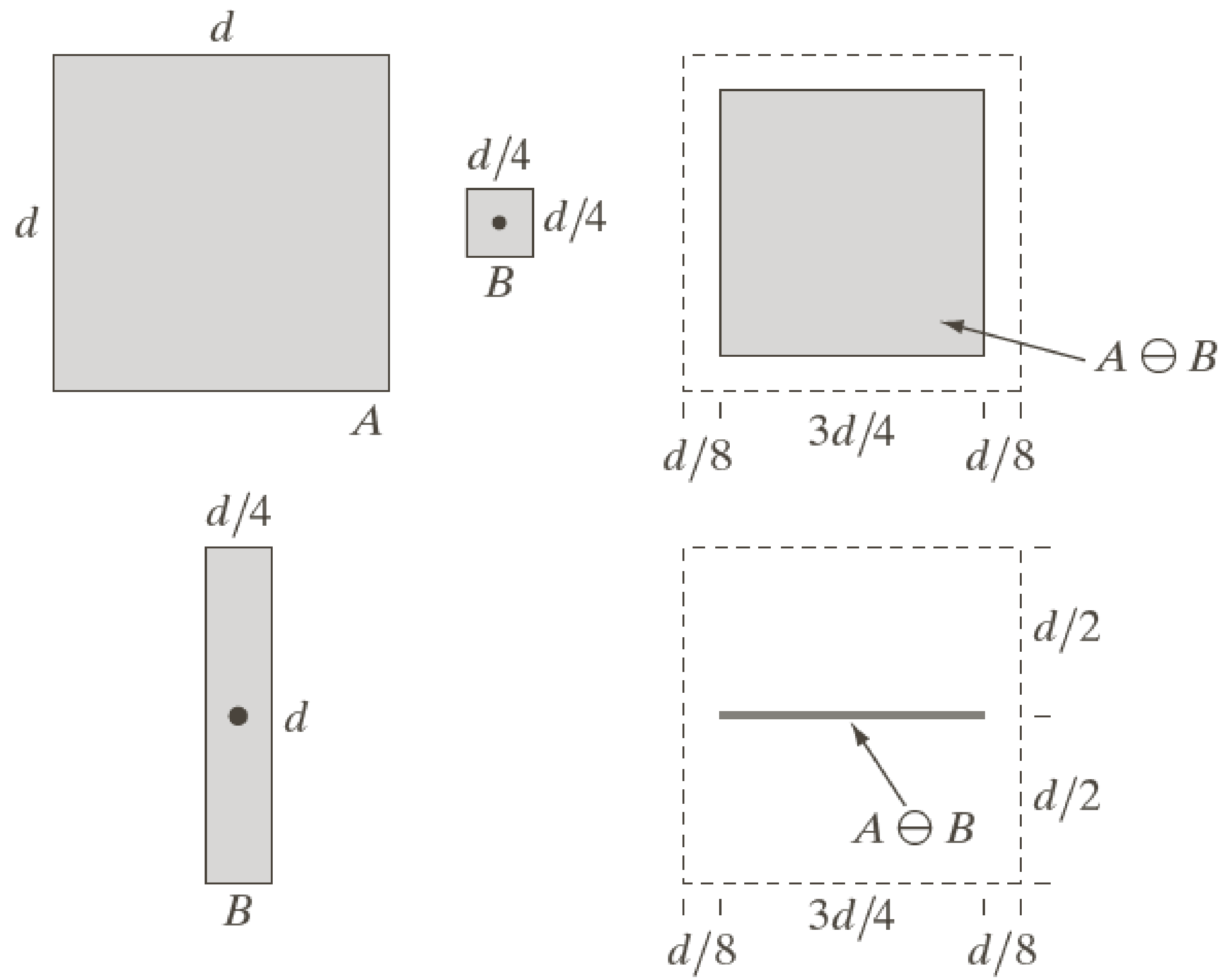
Structuring Element 1

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

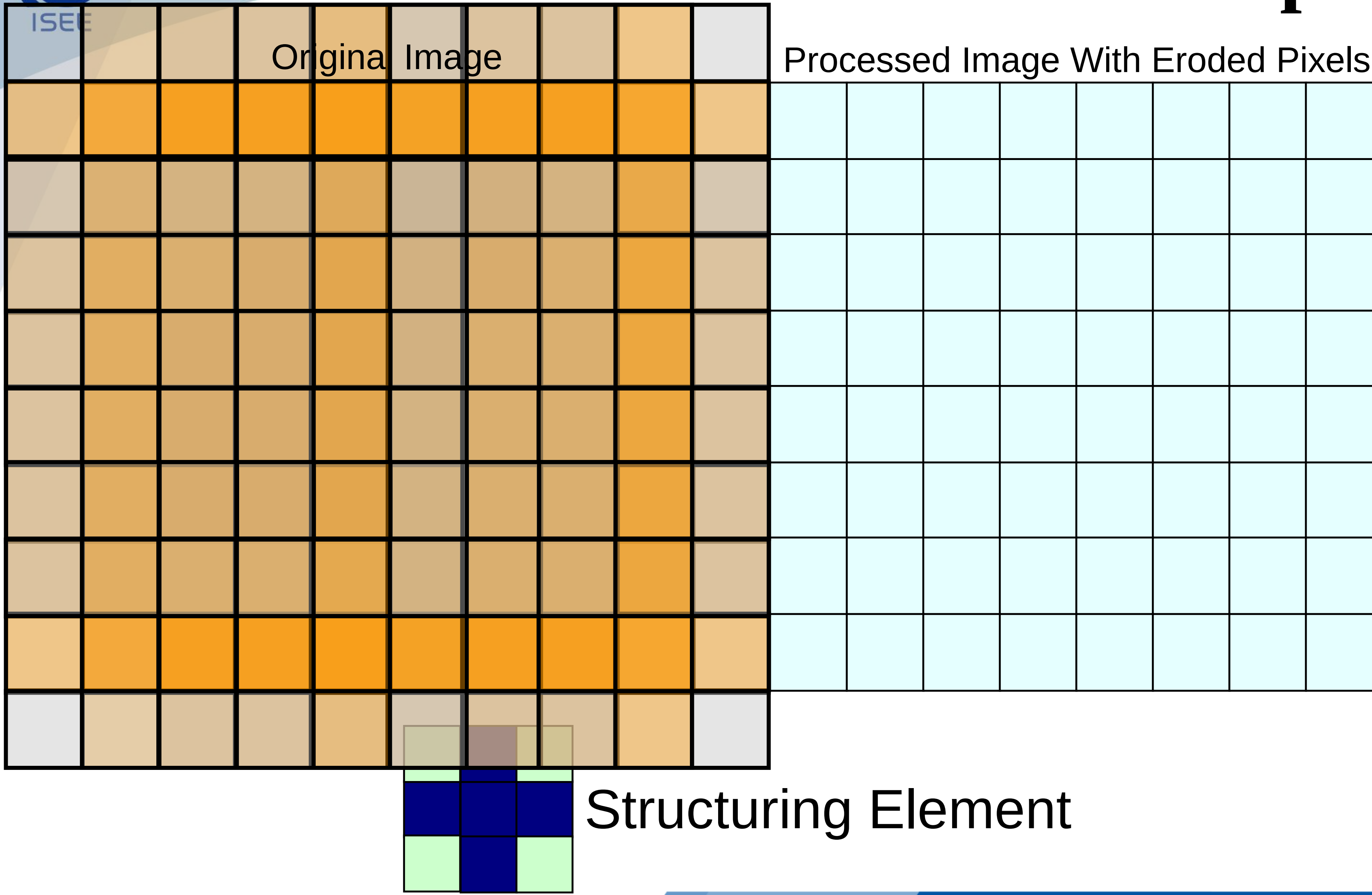
Structuring Element 2

# Fundamental Operation - Erosion

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

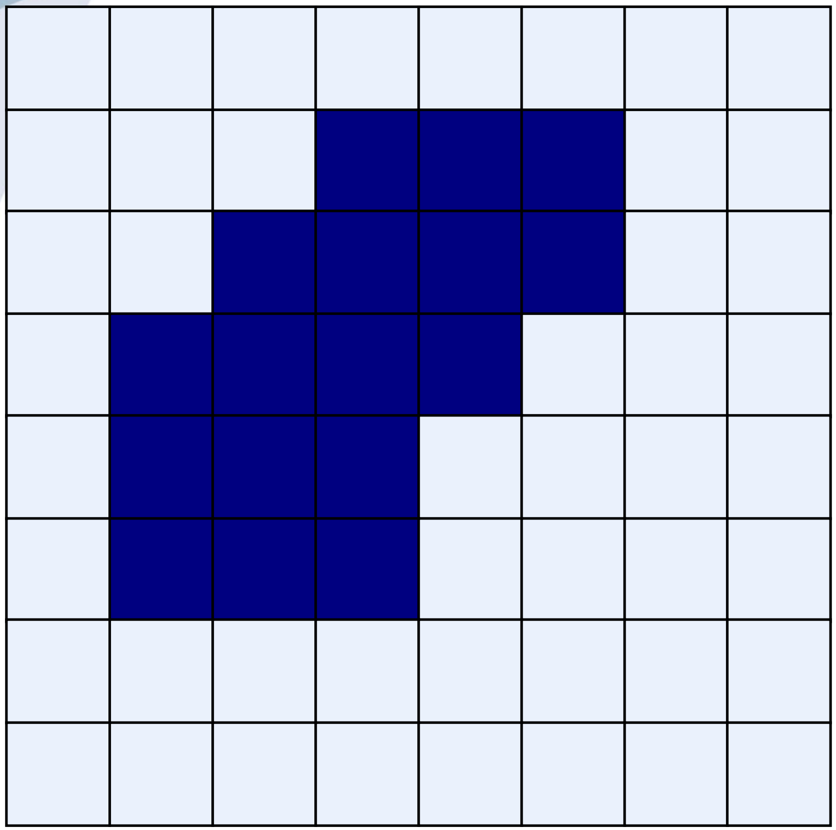


# Erosion Example

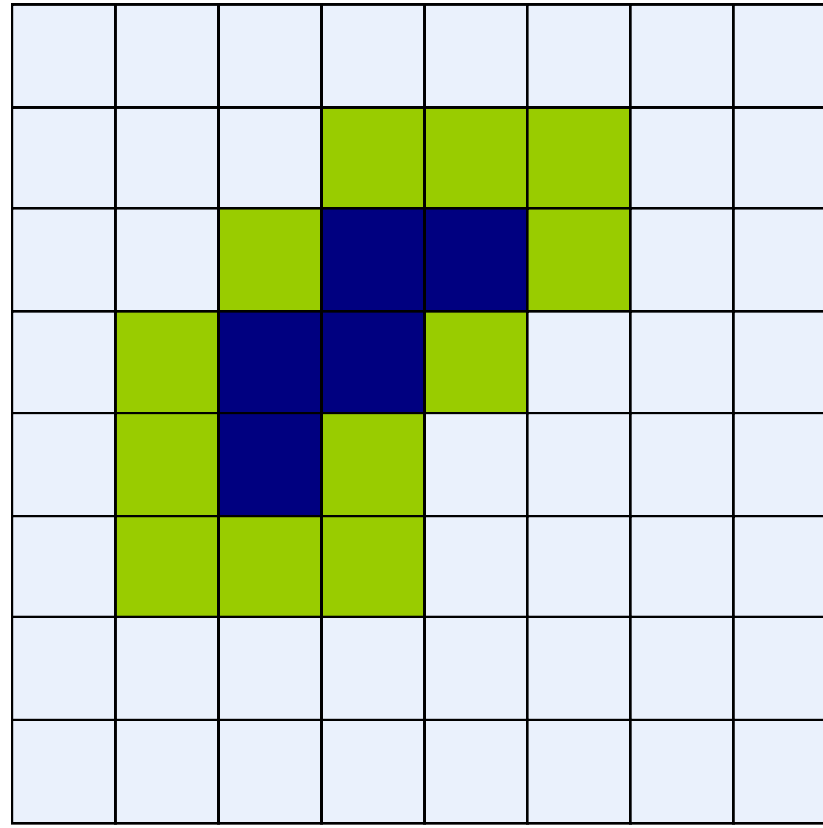


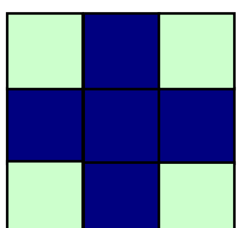
# Erosion Example

Original Image



Processed Image

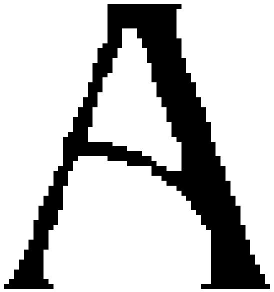




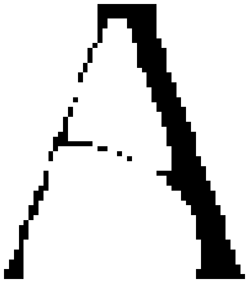
Structuring Element

# Erosion Example 1

Original image



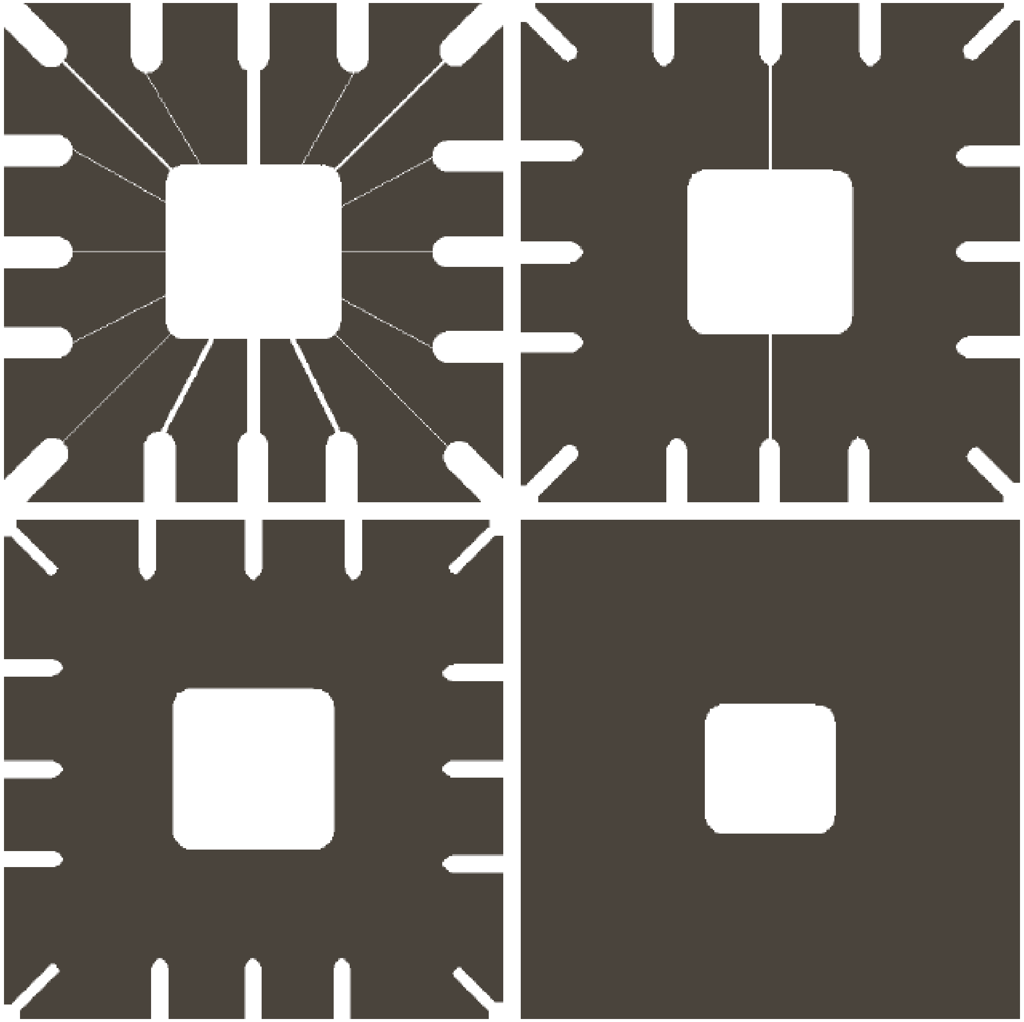
Erosion by 3*3 square structuring element



Erosion by 5*5 square structuring element

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

# Erosion Example 2



Original image (486x486 binary)

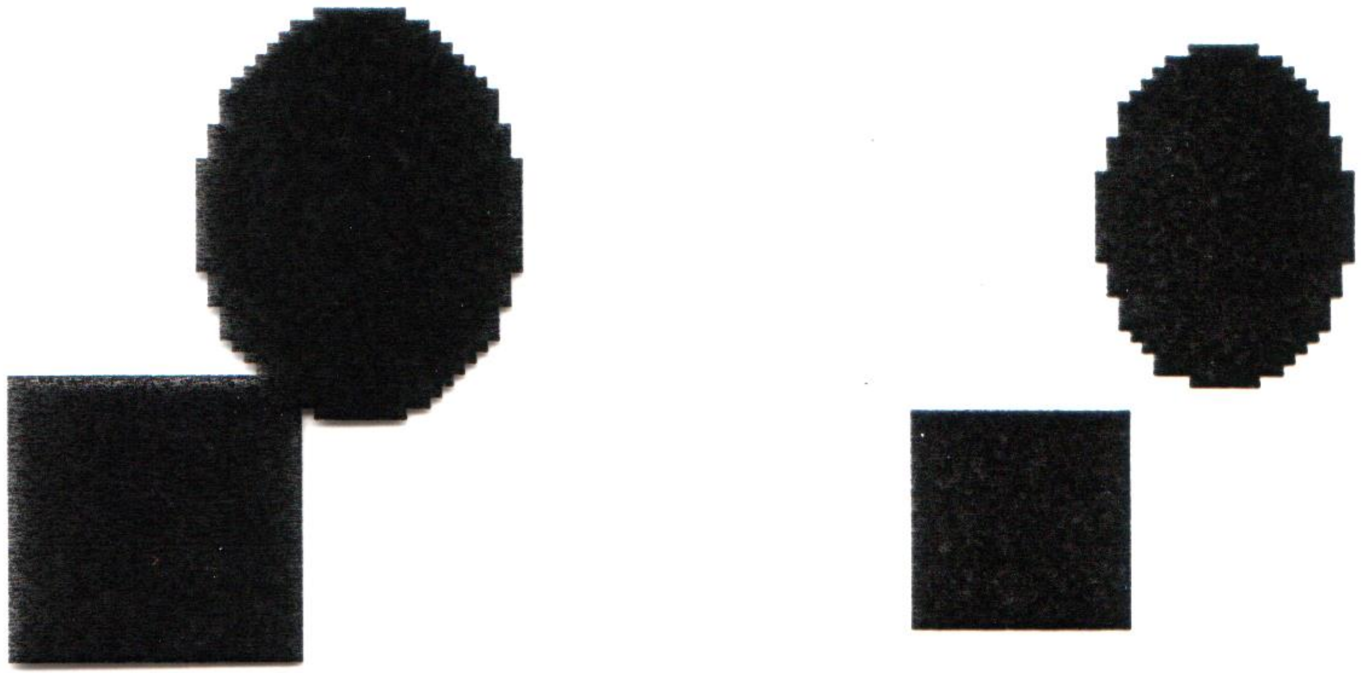
After erosion with SE of size 11x11

After erosion with SE of size 15x15

After erosion with SE of size 45x45

# What Is Erosion For?

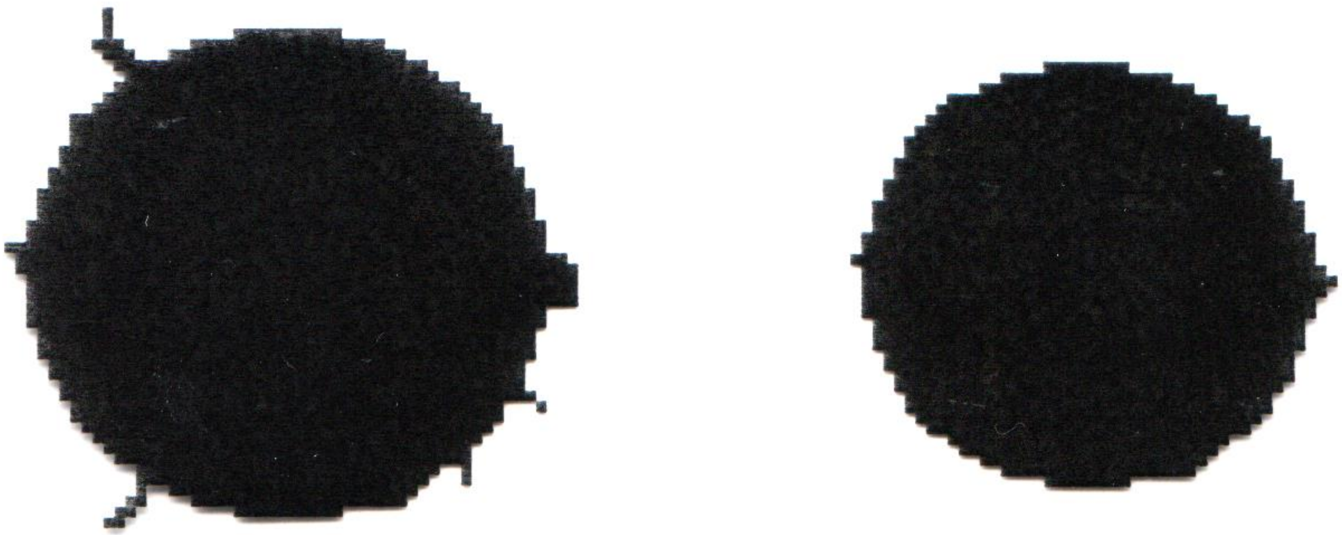
Erosion can split apart joined objects



# What Is Erosion For?

- *morphological filtering* operation

image details smaller than the structuring element are filtered (removed) from the image



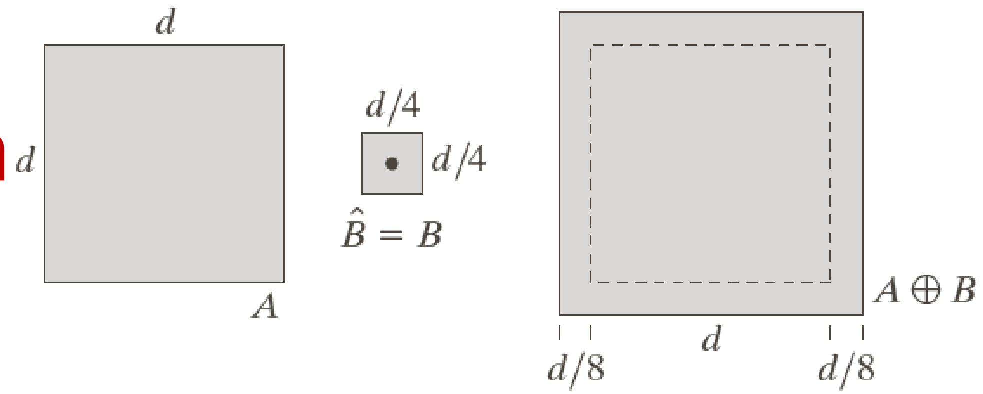
- **Watch out:** Erosion shrinks objects

# Dilation

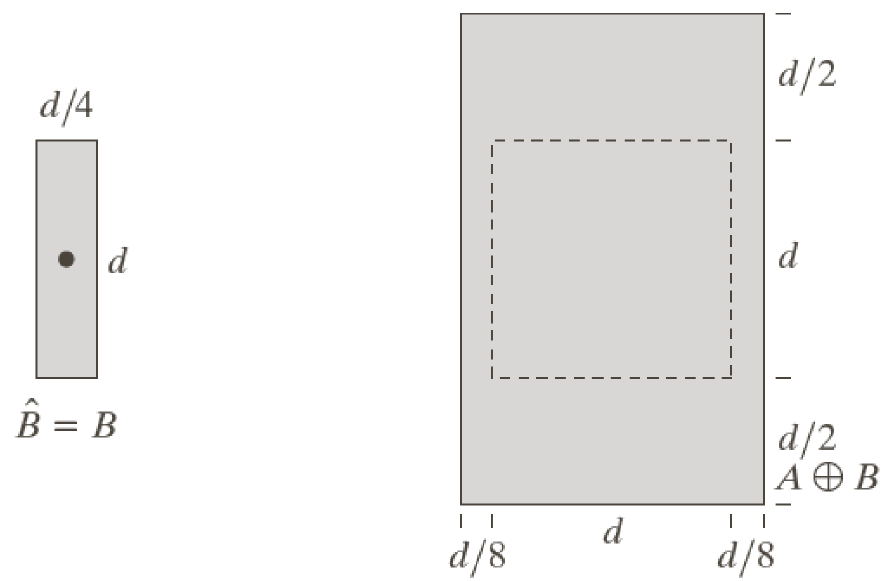
$$A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \varnothing \right\}$$

- Why reflection?

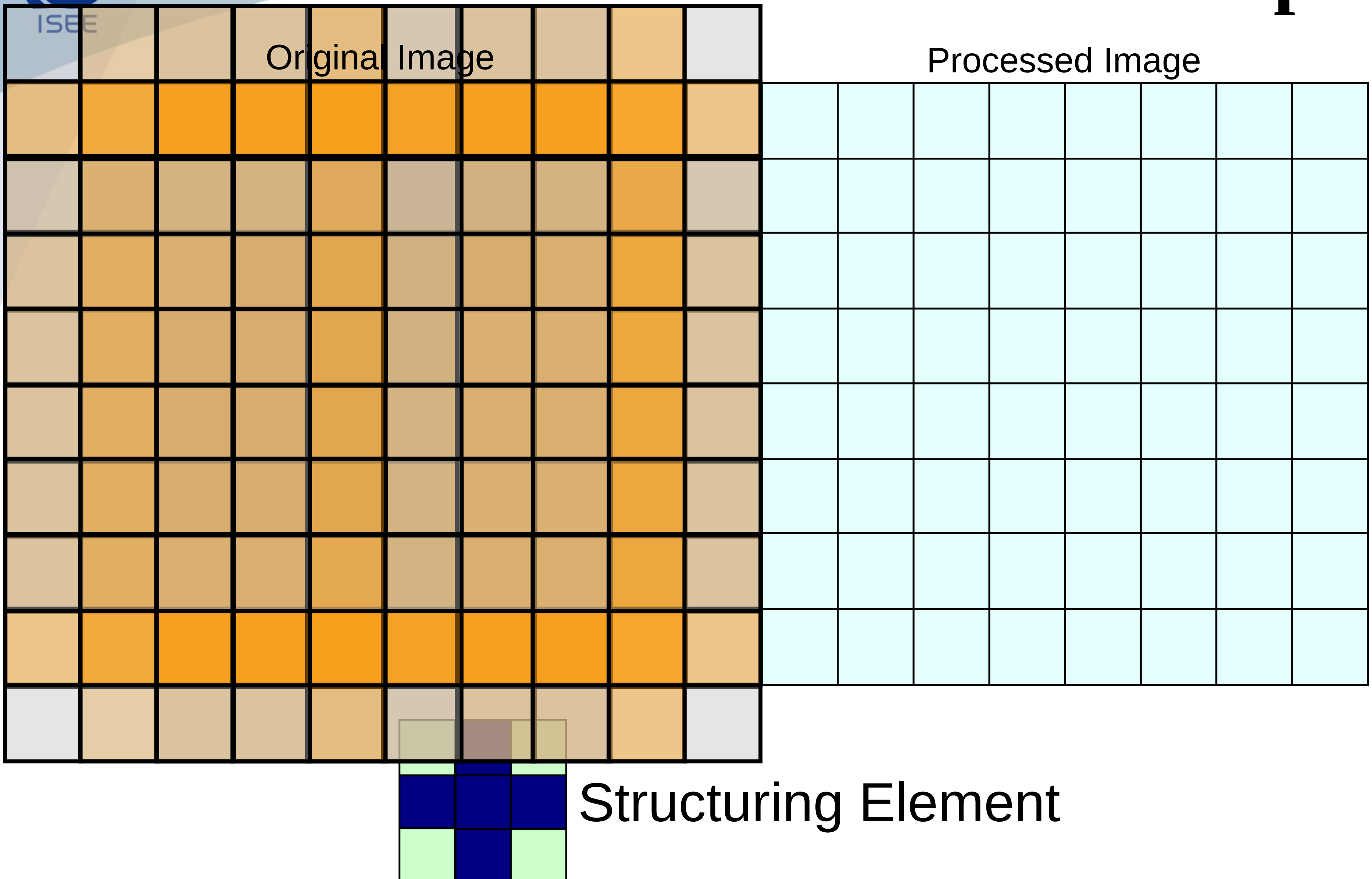
Make Duality between Erosion & Dilation




- The reflection and shifting of B is analogous to spatial convolution

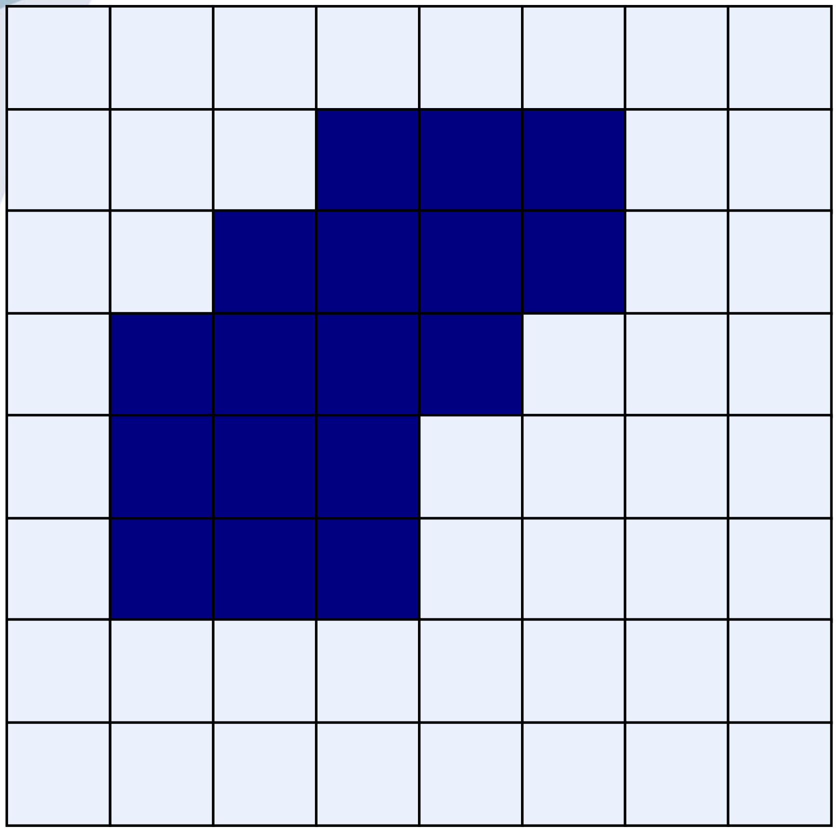


# Dilation Example

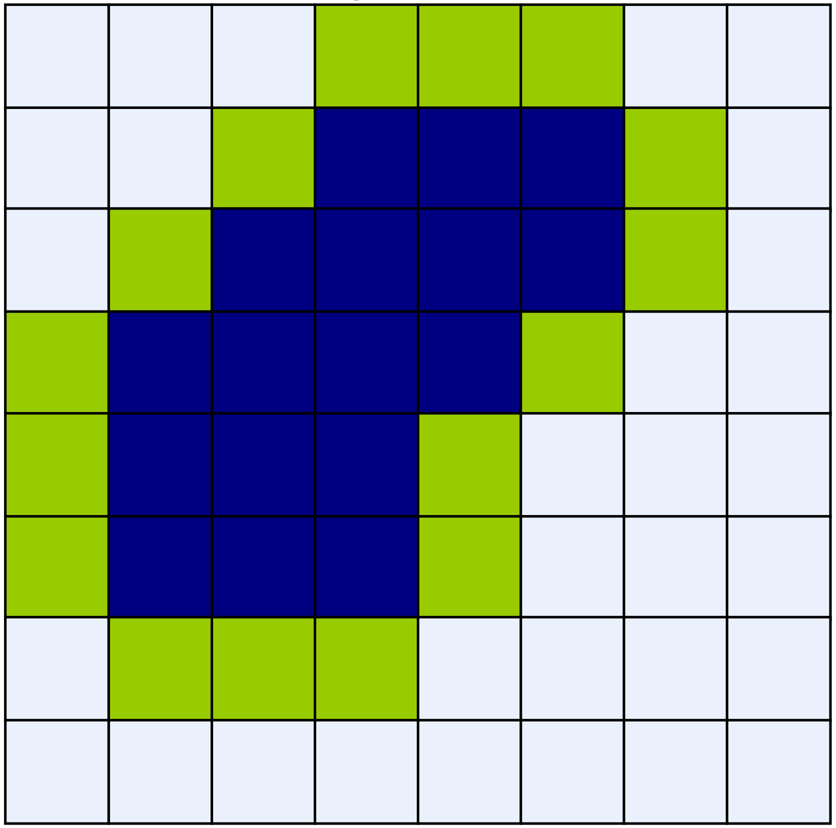


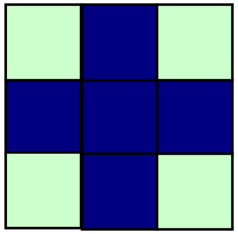
# Dilation Example

Original Image



Processed Image With Dilated Pixels





Structuring Element

# Dilation Example 1

Original image

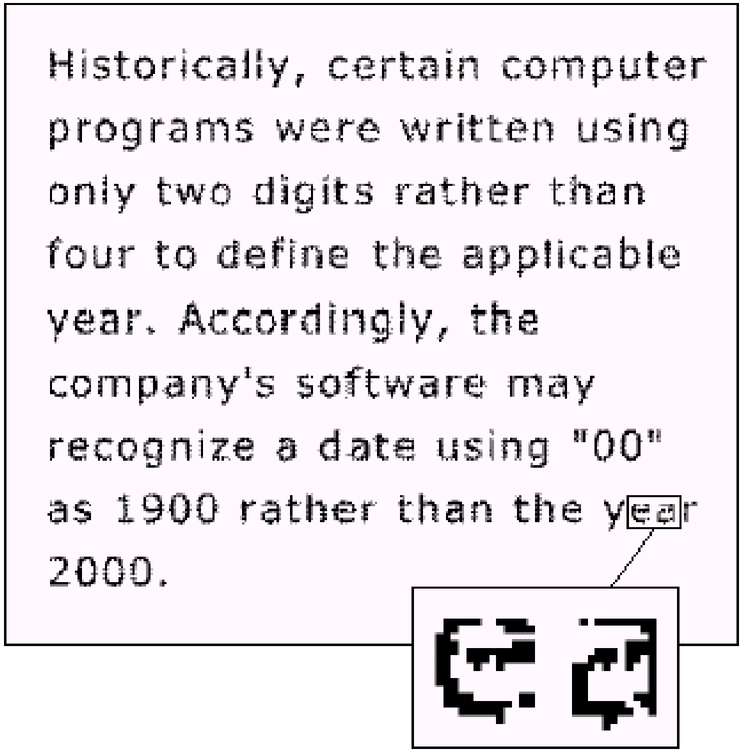
Dilation by 3*3 square structuring element

Dilation by 5*5 square structuring element

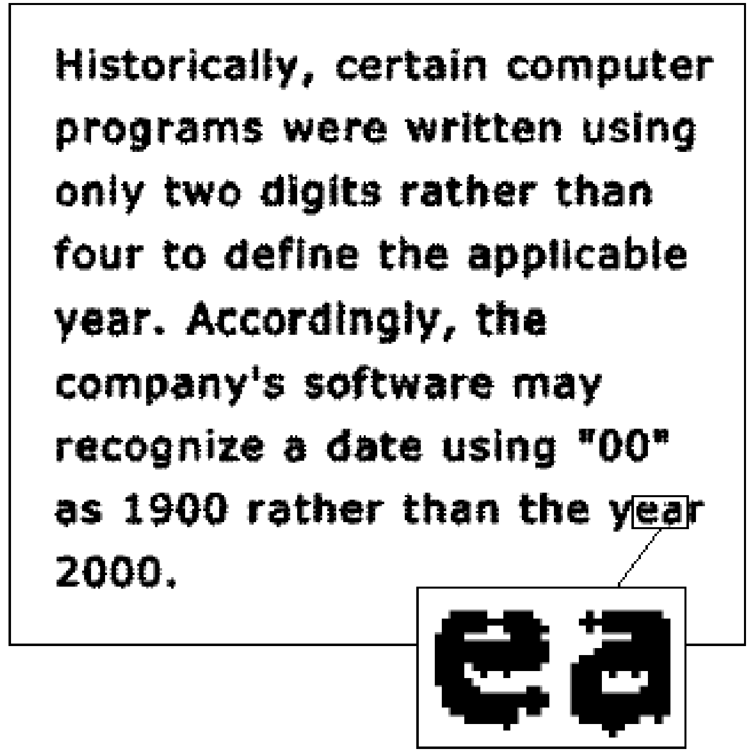
**Watch out:** In these examples a 1 refers to a black pixel!

# Dilation Example 2

Original image




After dilation




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

Structuring element

# What Is Dilation For?

Dilation can repair breaks

Dilation can repair intrusions

**Watch out:** Dilation enlarges objects

- Duality

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

$$(A \oplus B)^c = A^c \ominus \hat{B}$$

Proof:

$$(A \ominus B)^c = \left\{z \,|\, (B)_z \subseteq A\right\}^c$$

$$= \left\{z \,|\, (B)_z \cap A^c = \varnothing\right\}^c$$

$$= \left\{z \,|\, (B)_z \cap A^c \neq \varnothing\right\}$$

$$\Downarrow \qquad A \oplus B = \left\{z \,|\, (\hat{B})_z \cap A \neq \varnothing\right\}$$

$$= A^c \oplus \hat{B}$$

- Definition: $A \oplus B = \left\{ z \mid (\hat{B})_z \cap A \neq \varnothing \right\}$

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

- Dilation = Minkowski Union

$$A \oplus B = \bigcup_{z \in A} B_z = \left\{ (z+b) \,\middle|\, z \in A, b \in B \right\} = \bigcup_{b \in B} A_b$$

- Erosion = Minkowski Intersection

$$A \Theta B = \bigcap_{b \in B} A_{-b}$$

proof:

$$A \Theta B = (A^C \oplus \hat{B})^C = \Big(\bigcup_{b \in \hat{B}} (A^C)_b\Big)^C = \bigcap_{b \in \hat{B}} A_b = \bigcap_{b \in B} A_{-b}$$

# Compound Operations

More interesting morphological operations can be performed by performing combinations of erosions and dilations

The most widely used of these *compound operations* are:

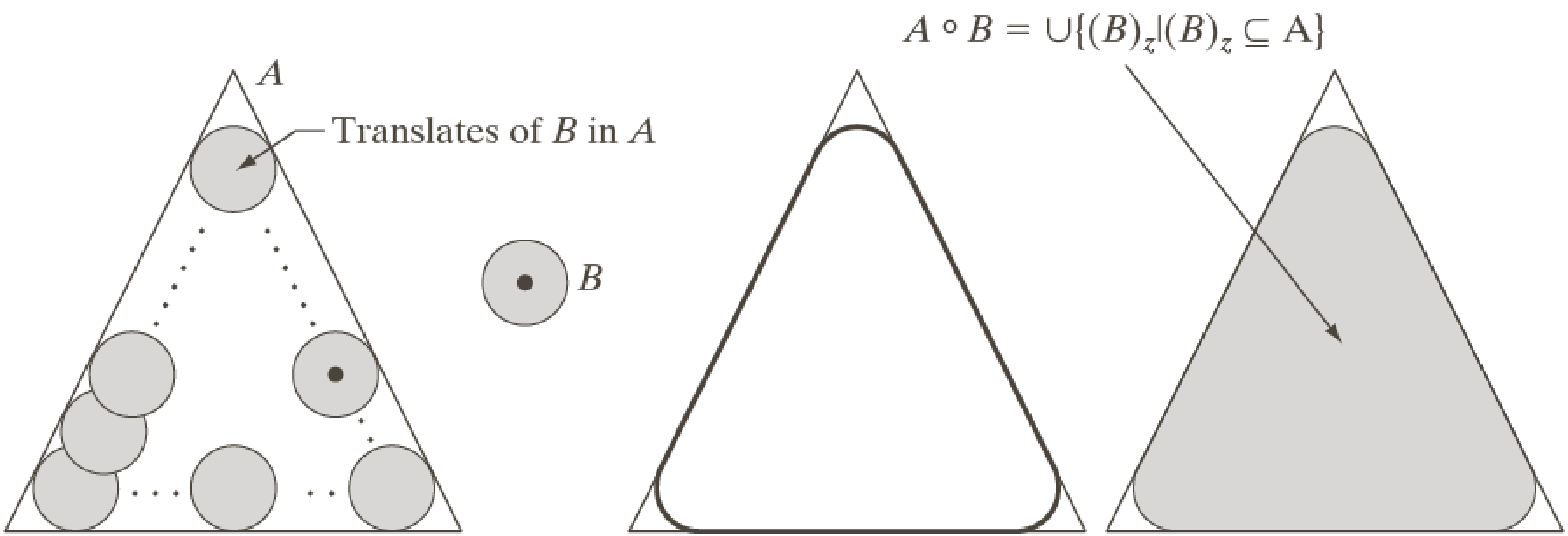
- Opening
- Closing

# Opening

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

$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$

Structuring element B rolling along the inner boundary of A

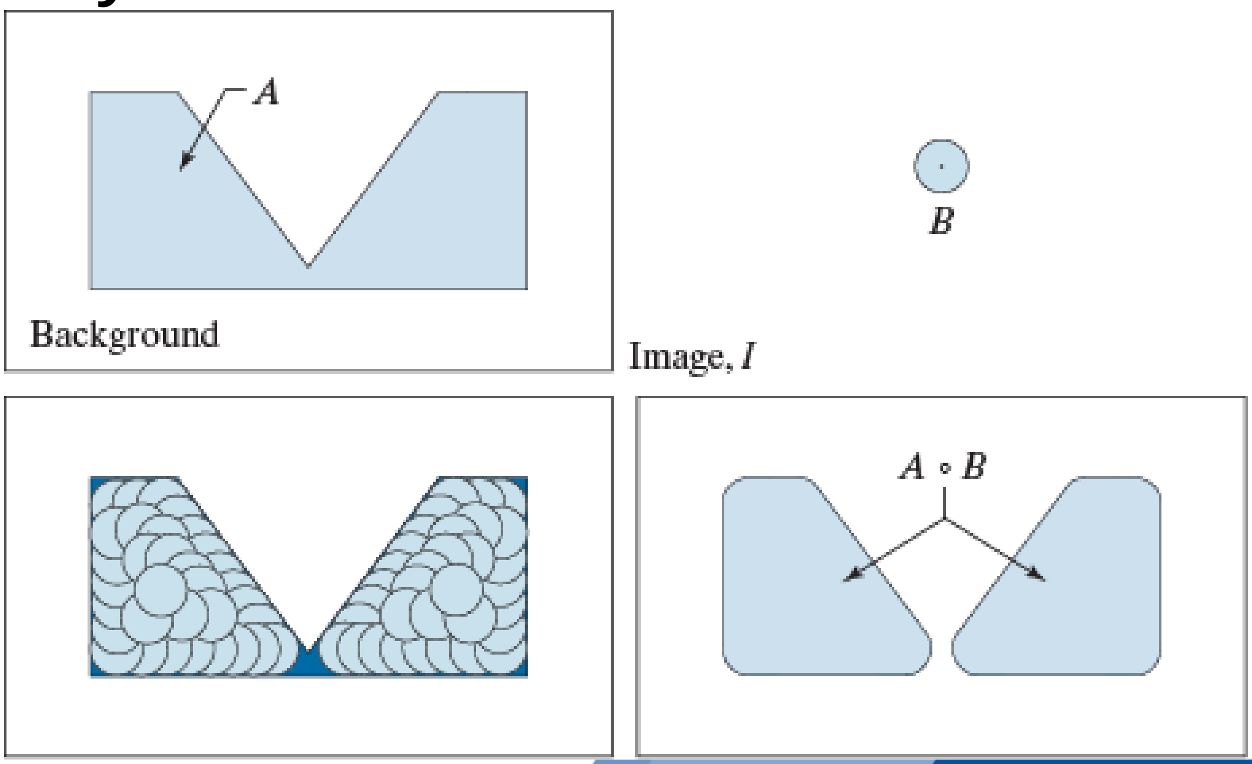


# Opening

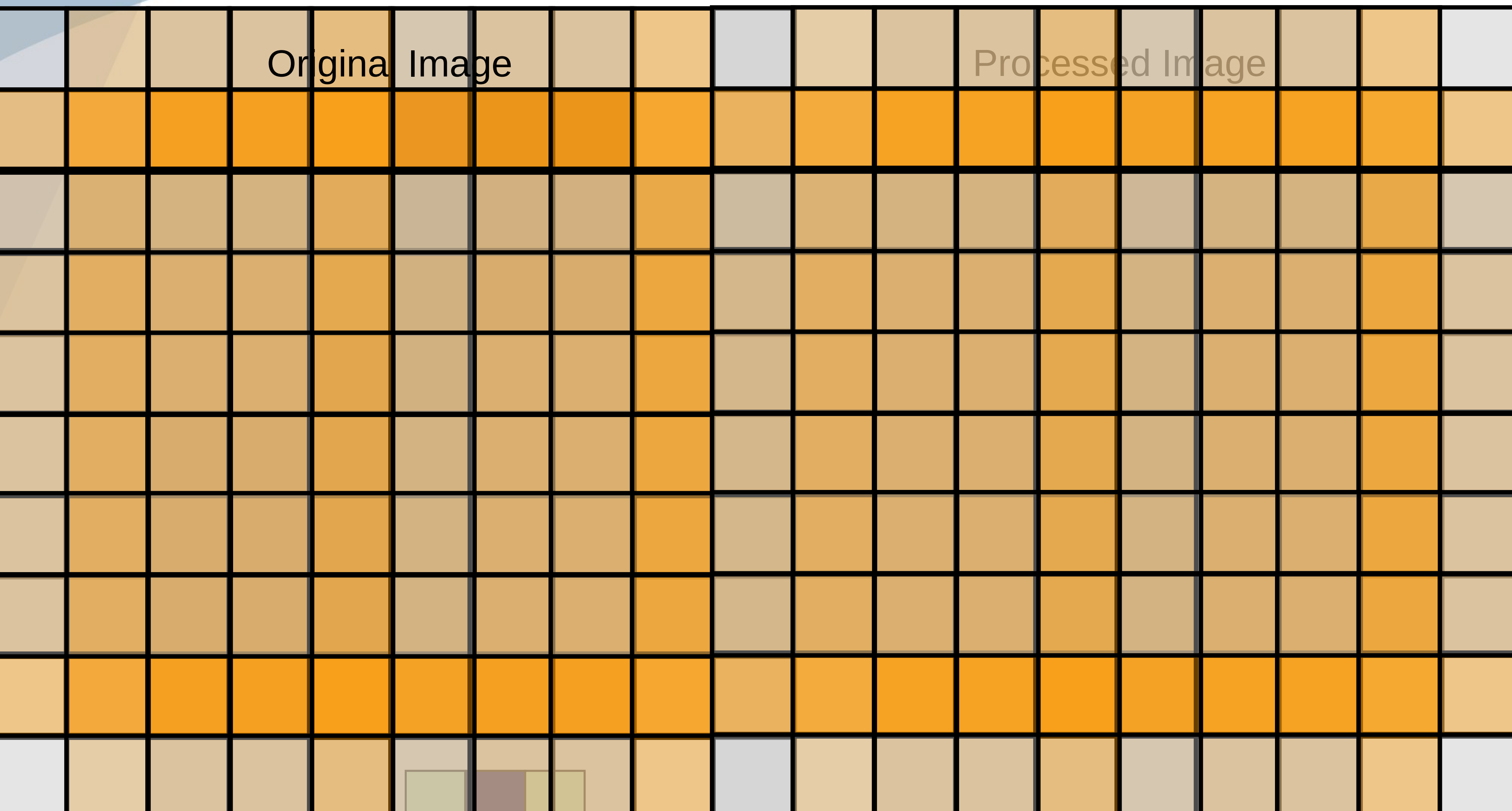
$$A \circ B = (A \ominus B) \oplus B$$

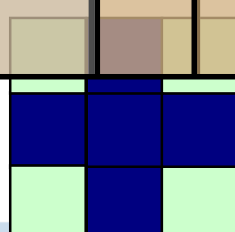
$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$

Structuring element B rolling along the inner boundary of A



# Opening Example

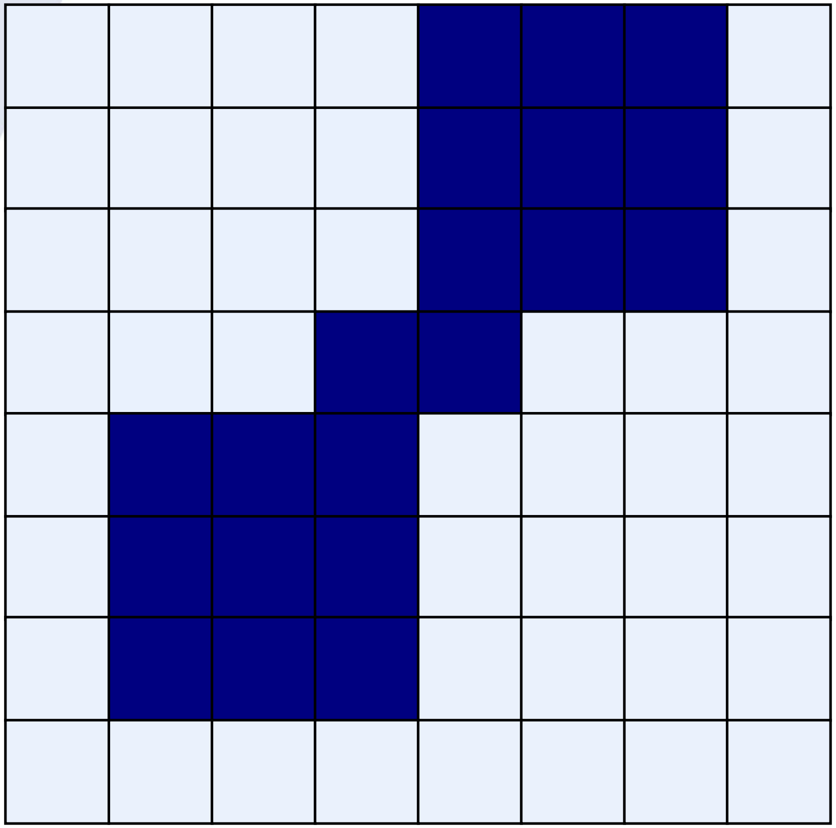


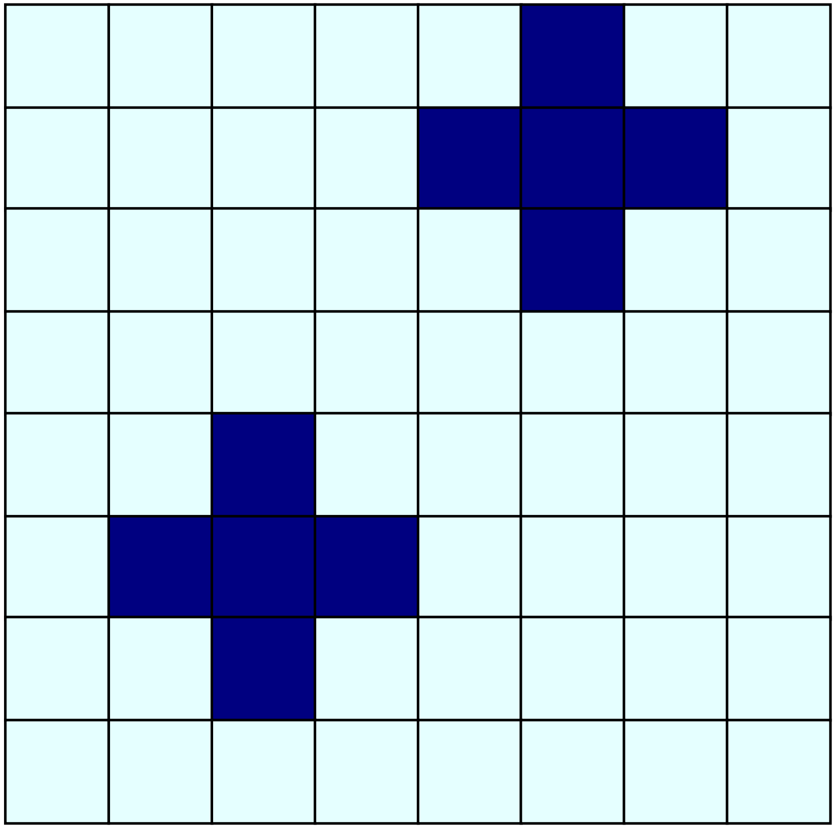
Structuring Element

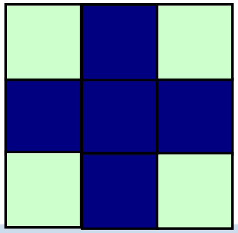
# Opening Example

Original Image



Processed Image



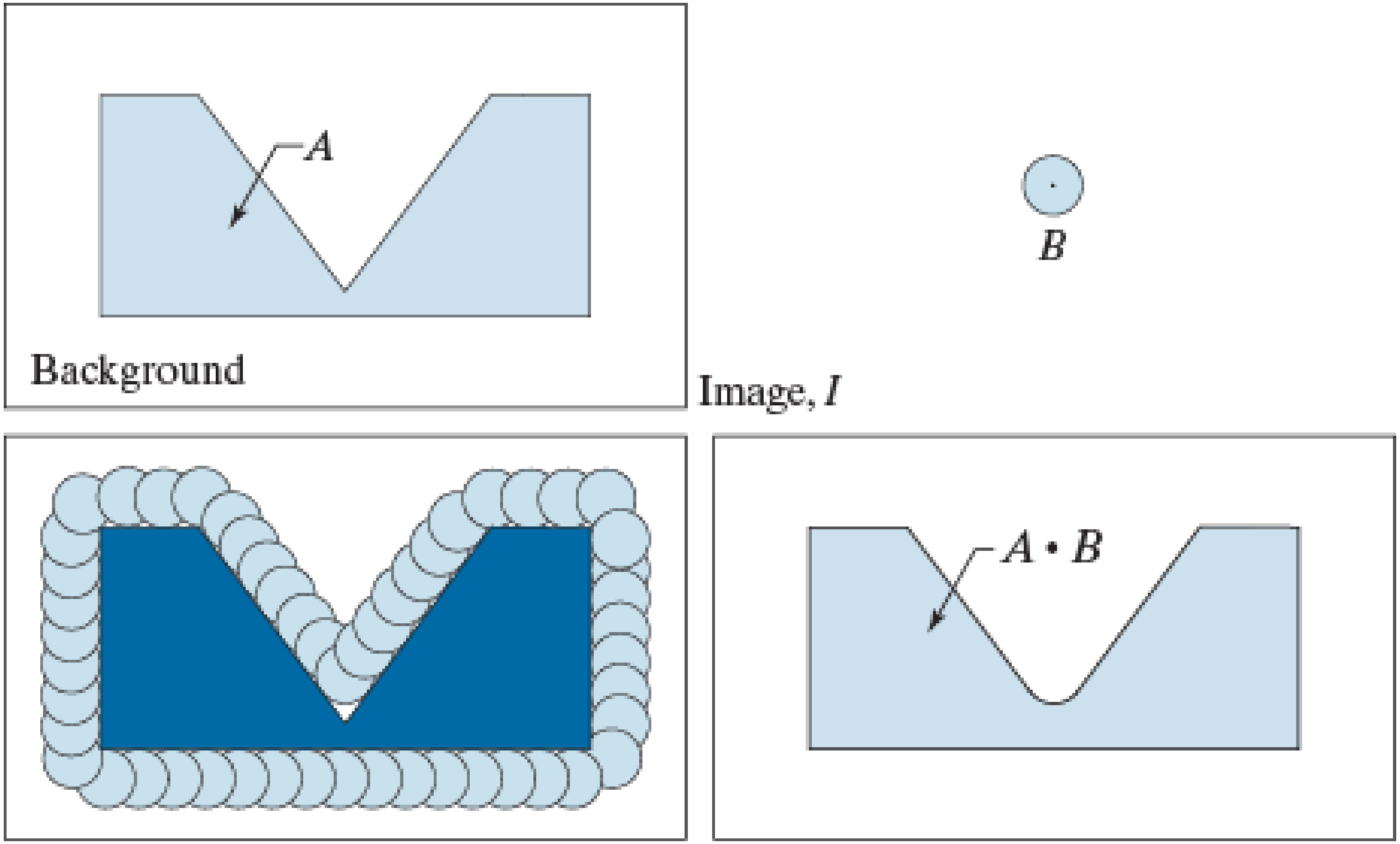


Structuring Element

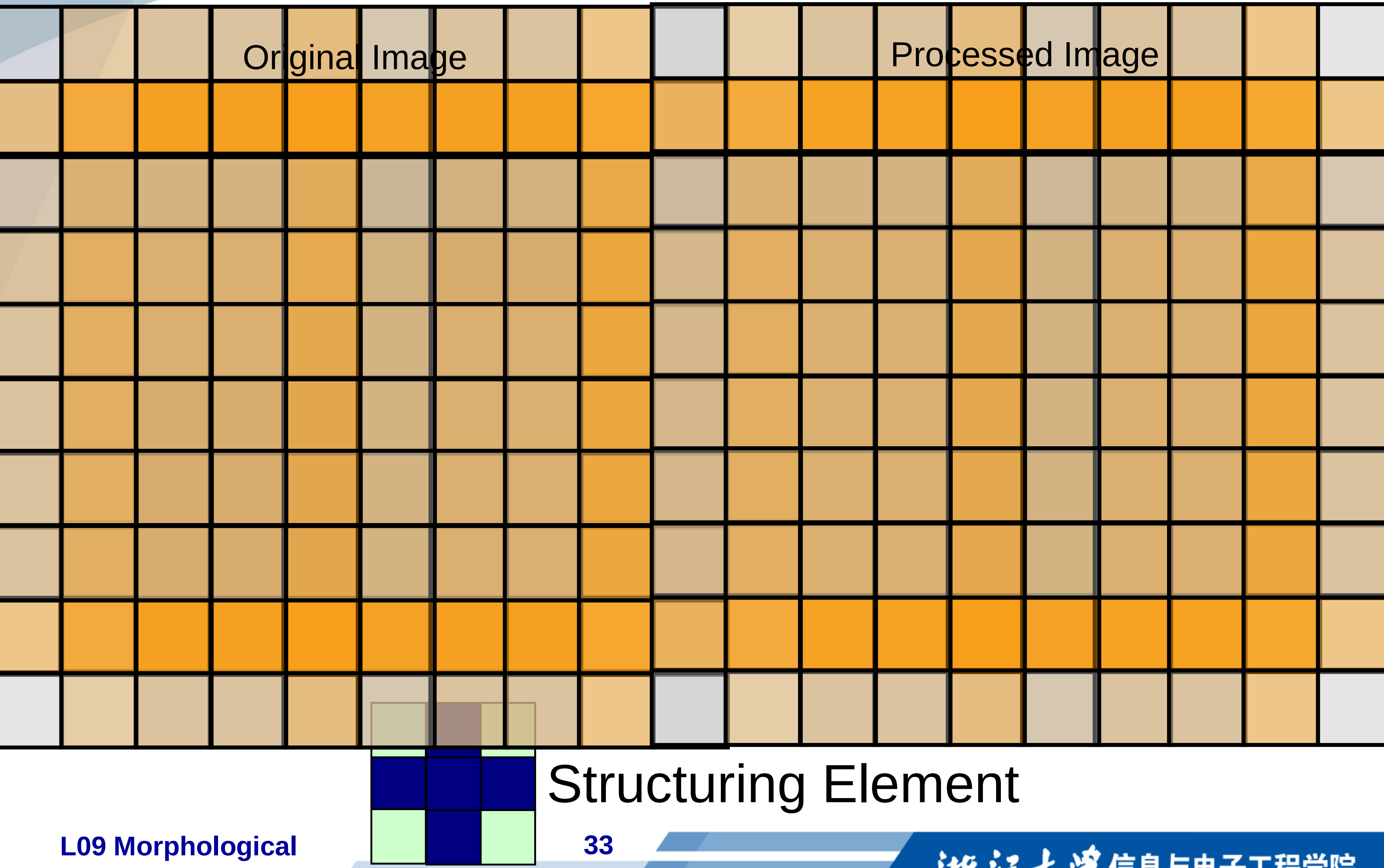
# Closing

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

Structuring element B rolling along the outer boundary of A

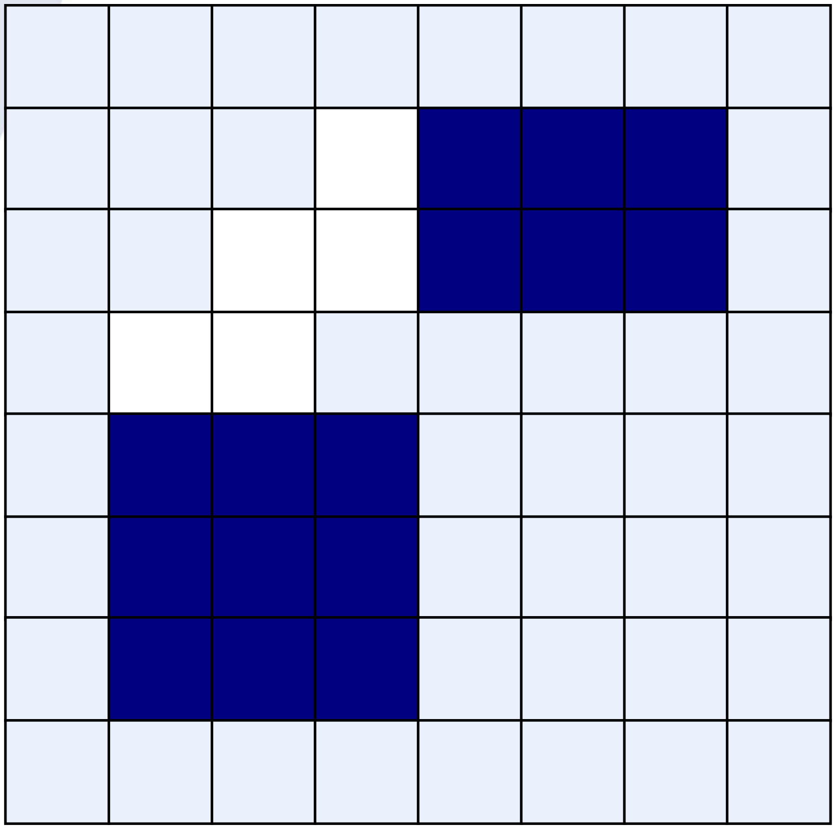


# Closing Example

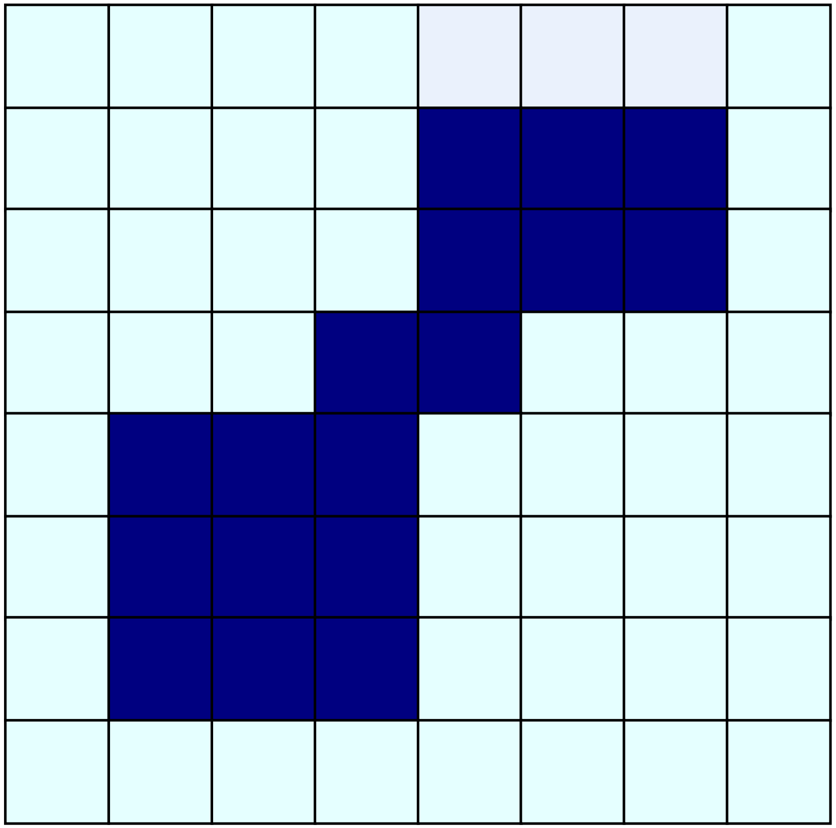


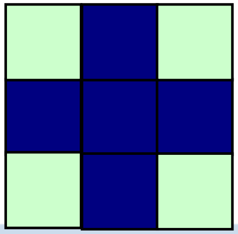
# Closing Example

Original Image



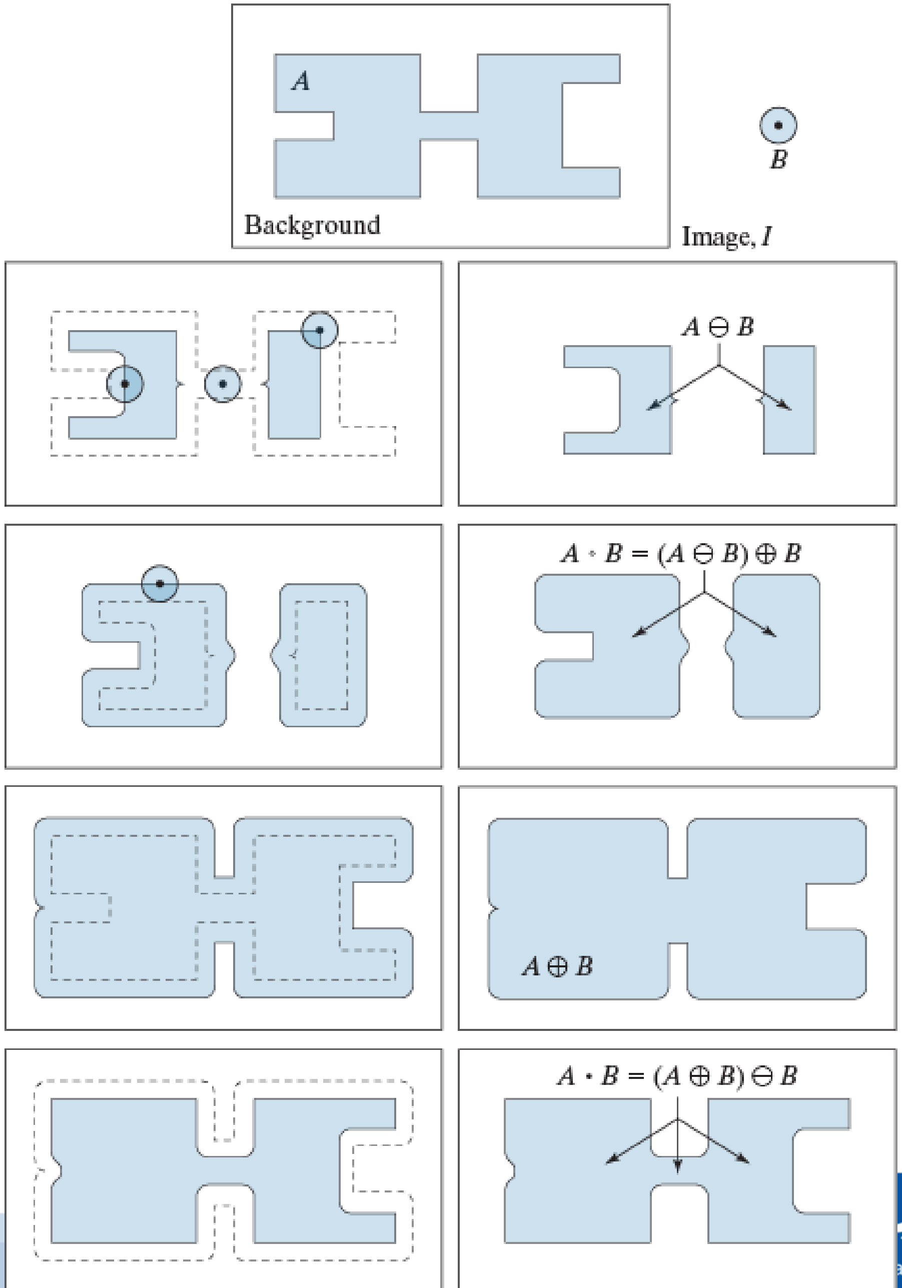
Processed Image





Structuring Element

# Opening and Closing Example



# Opening and Closing Properties

- Duality

$$(A \bullet B)^c = (A^c \circ \hat{B})$$
$$(A \circ B)^c = (A^c \bullet \hat{B})$$

- Opening

**(a)** $A \circ B$ is a subset (subimage) of $A$.
**(b)** If $C$ is a subset of $D$, then $C \circ B$ is a subset of $D \circ B$.
**(c)** $(A \circ B) \circ B = A \circ B$.

- Closing

**(a)** $A$ is a subset (subimage) of $A \bullet B$.
**(b)** If $C$ is a subset of $D$, then $C \bullet B$ is a subset of $D \bullet B$.
**(c)** $(A \bullet B) \bullet B = A \bullet B$.

# Opening, Closing & Set Operations

- Set Union

$$\left(\bigcup_{i=1}^{n} A_i\right) \circ B \supseteq \bigcup_{i=1}^{n} \left(A_i \circ B\right)$$

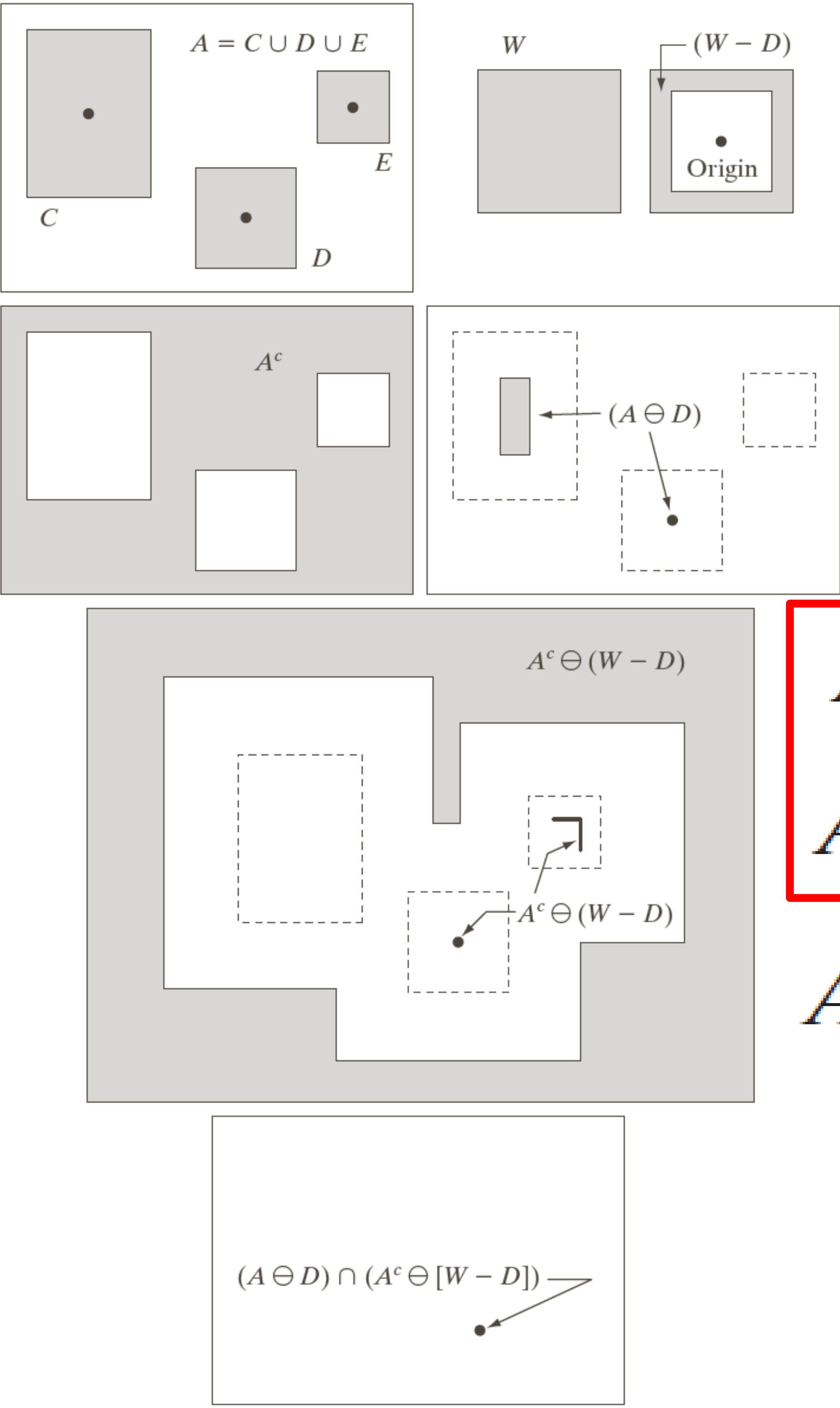
$$\left(\bigcup_{i=1}^{n} A_i\right) \bullet B \supseteq \bigcup_{i=1}^{n} \left(A_i \bullet B\right)$$

- Set Intersection

$$\left(\bigcap_{i=1}^{n} A_i\right) \circ B \subseteq \bigcap_{i=1}^{n} \left(A_i \circ B\right)$$

$$\left(\bigcap_{i=1}^{n} A_i\right) \bullet B \subseteq \bigcap_{i=1}^{n} \left(A_i \bullet B\right)$$

# Morphological Processing Example




# Hit-or-Miss Transformation

Find the location of a shape

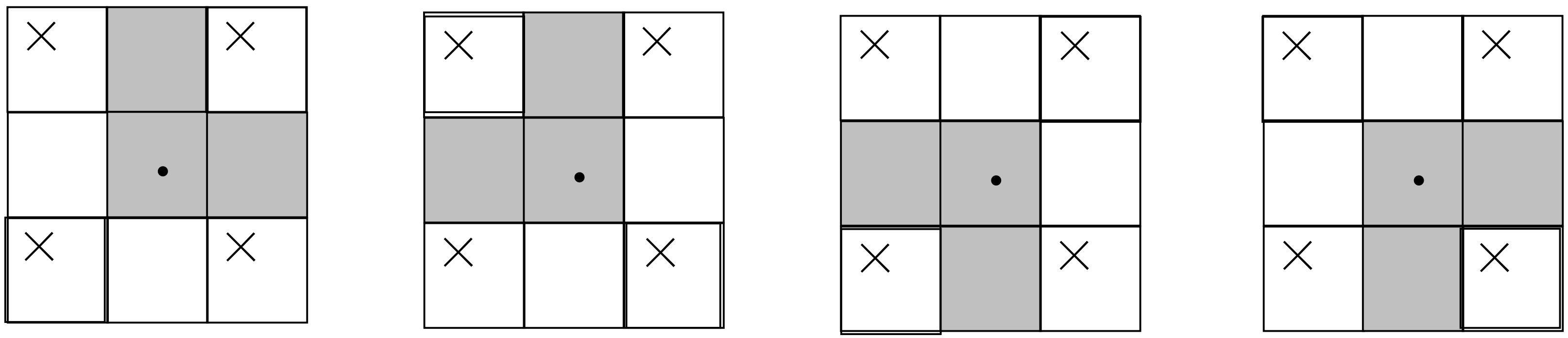
$$A \circledast B = (A \ominus D) \cap \left[A^c \ominus (W - D)\right]$$

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

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

# Hit-or-Miss Transformation

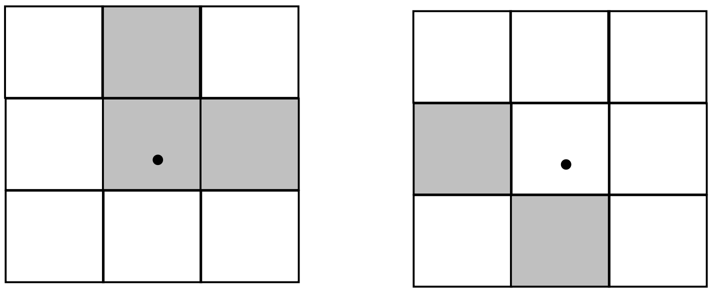
× : don't care pixels



(a) (b) (c) (d)

Structuring elements for corner detection

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$



(a) $B_1$ (b) $B_2$

# Properties

| 性质 \ 运算 | 膨胀 | 腐蚀 | 开 | 闭 |
|---|---|---|---|---|
| 位移不变性 | $(A)_x \oplus B = (A \oplus B)_x$ | $(A)_x \Theta B = (A \Theta B)_x$ | $A \circ (B)_x = A \circ B$ | $A \bullet (B)_x = A \bullet B$ |
| 互换性 | $A \oplus B = B \oplus A$ | | | |
| 组合性 | $(A \oplus B) \oplus C = A \oplus (B \oplus C)$ | $(A \Theta B) \Theta C = A \Theta (B \oplus C)$ | | |
| 增长性 | $A \subseteq B \Rightarrow A \oplus C \subseteq B \oplus C$ | $A \subseteq B \Rightarrow A \Theta C \subseteq B \Theta C$ | $A \subseteq B \Rightarrow A \circ C \subseteq B \circ C$ | $A \subseteq B \Rightarrow A \bullet C \subseteq B \bullet C$ |
| 同前性 | | | $(A \circ B) \circ B = A \circ B$ | $(A \bullet B) \bullet B = A \bullet B$ |
| 外延性 | $A \subseteq A \oplus B$ | $A \Theta B \subseteq A$ | $A \circ B \subseteq A$ | $A \subseteq A \bullet B$ |

上表中膨胀和腐蚀的外延性只当结构元原点在内部时成立

# Basic Morphological Algorithms

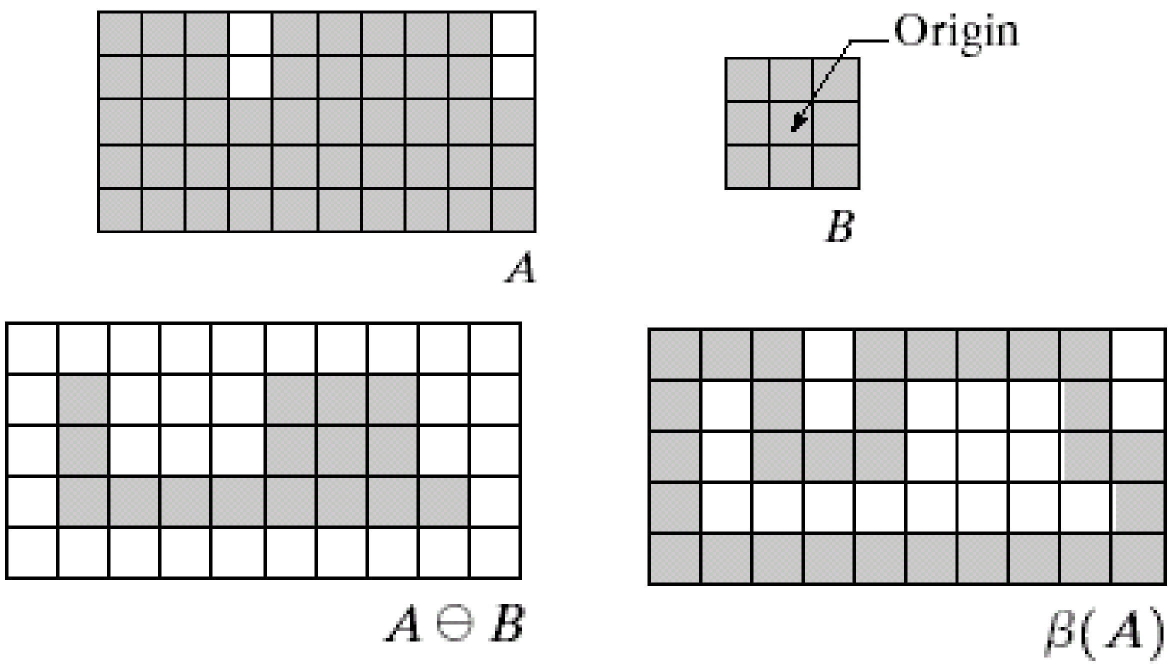
- Boundary Extraction
- Region Filling
- Extraction of Connected Components
- Convex Hull
- Thinning
- Thickening
- Skeletons
- Pruning（修剪）

# Boundary Extraction

Extracting the boundary (or outline) of an object is often extremely useful

The boundary can be given simply as

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

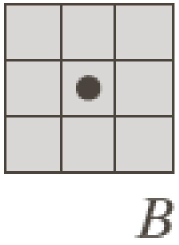


# Boundary Extraction Example

A simple image and the result of performing boundary extraction using a square 3*3 structuring element

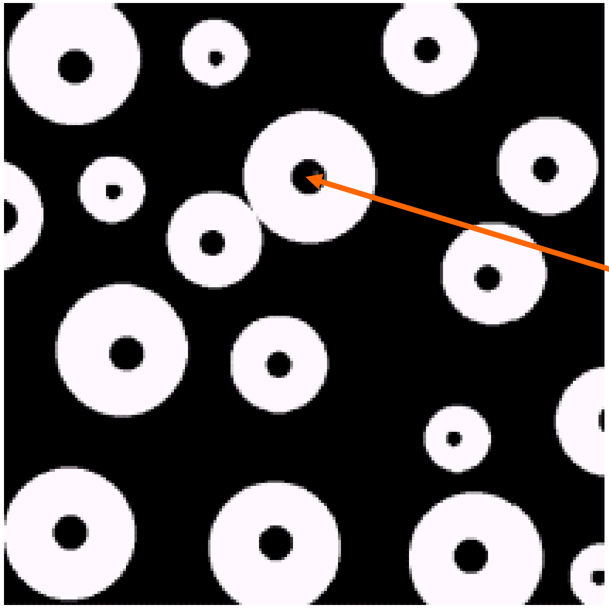
Original Image

Extracted Boundary



# Region Filling

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?

# Region Filling (cont…)

The key equation for region filling is

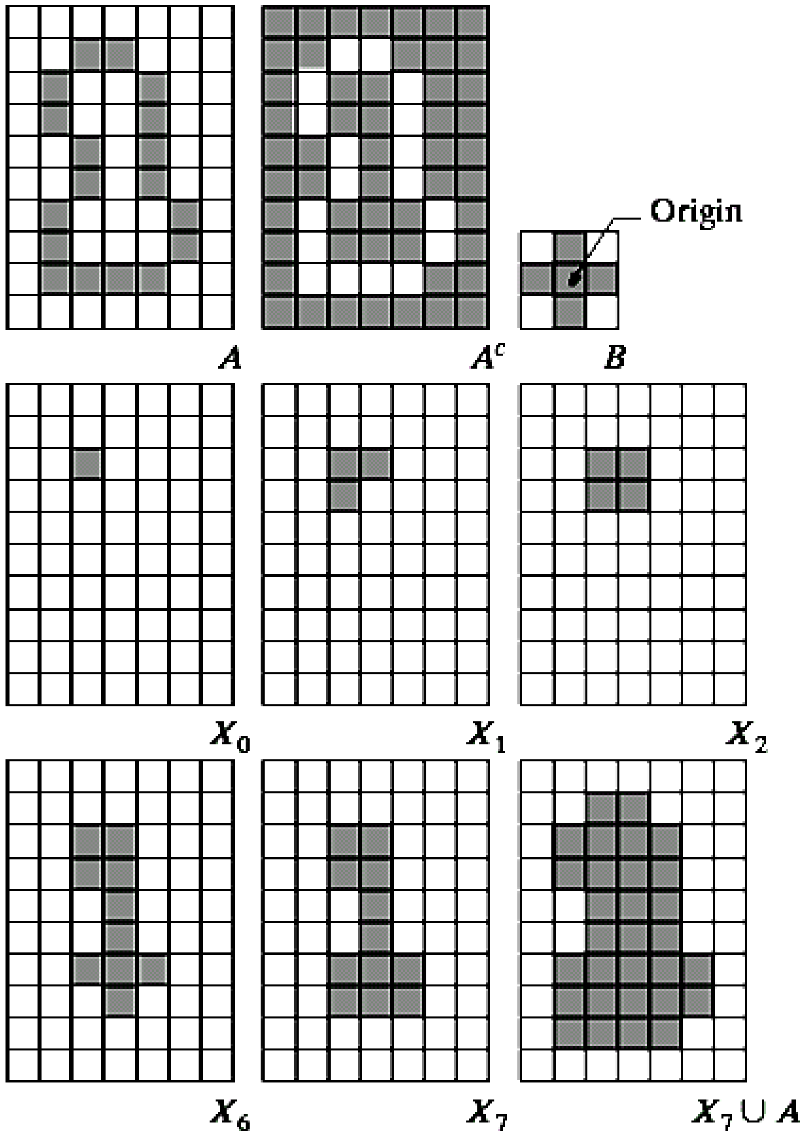
$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3.....$$

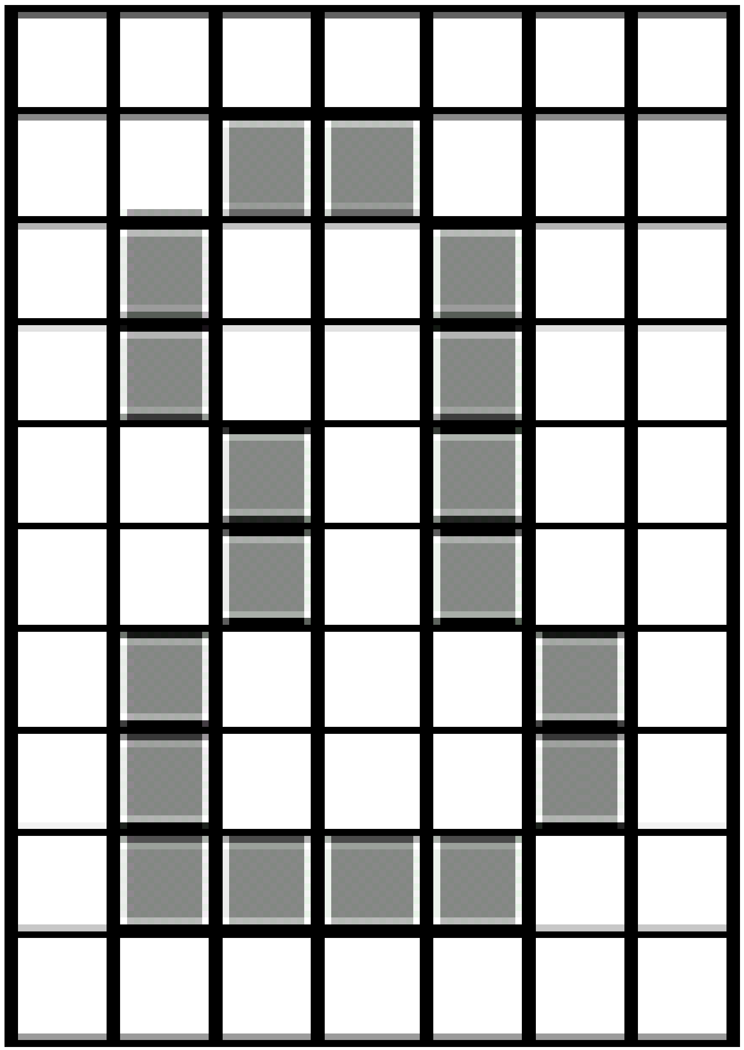
Where $X_0$ is simply the starting point inside the boundary, B is a simple structuring element and $A^c$ is the complement of A

This equation is applied repeatedly until $X_k$ is equal to $X_{k-1}$

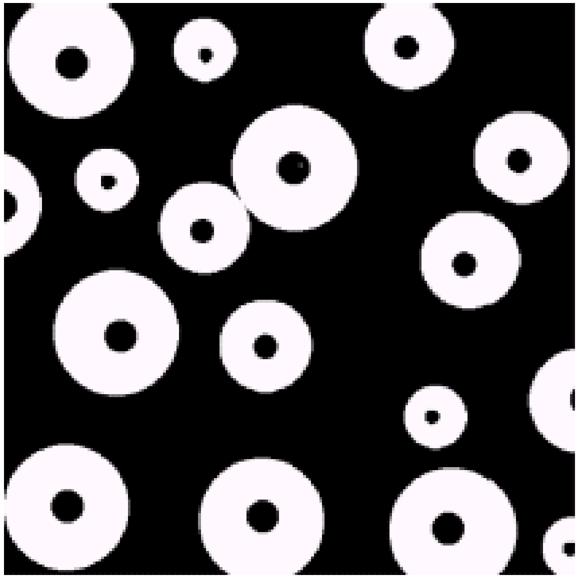
Finally the result is unioned with the original boundary

# Region Filling Step By Step

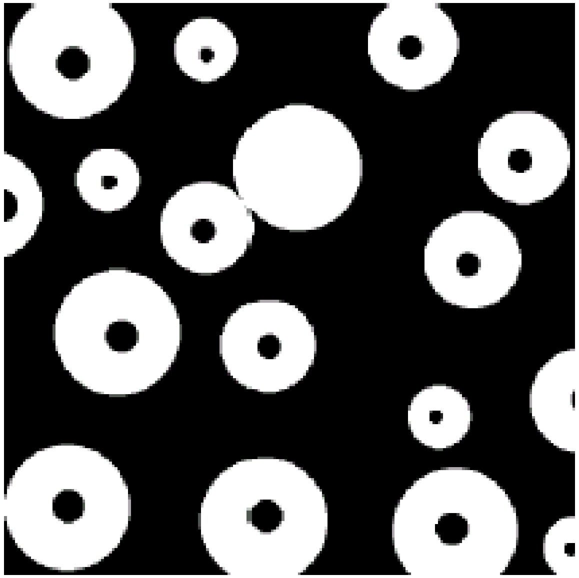


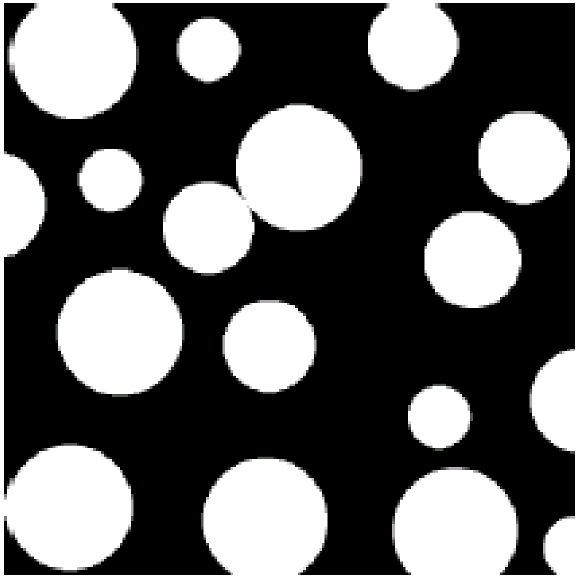
# Region Filling Example



Original Image



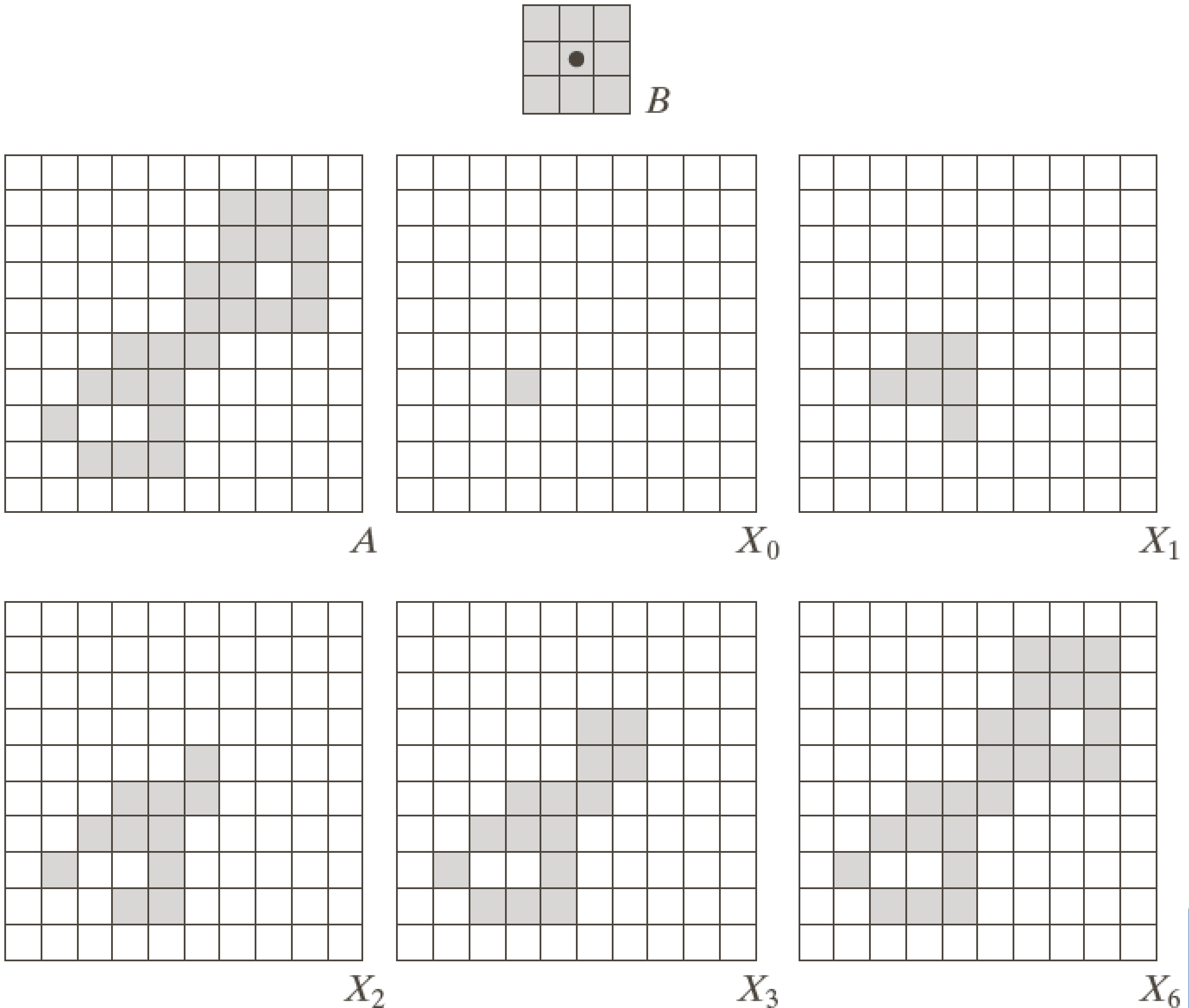
One Region Filled



All Regions Filled

# Extraction of Connected Components

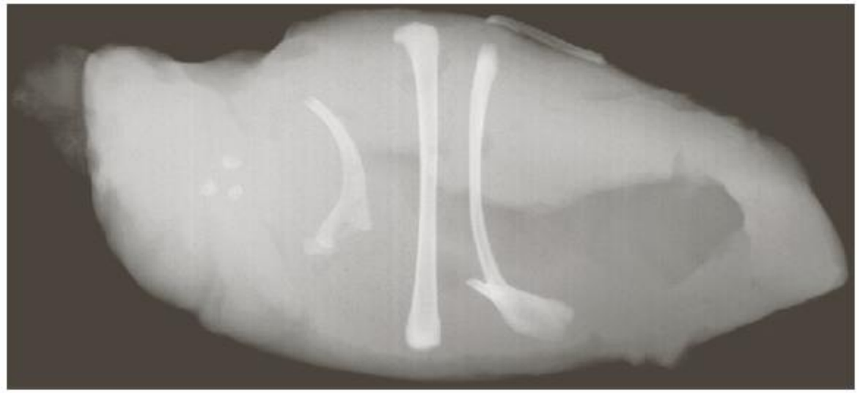
$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3, \ldots$$



# Extraction of Connected Components

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

X-ray Image of Chicken breast



Thresholded

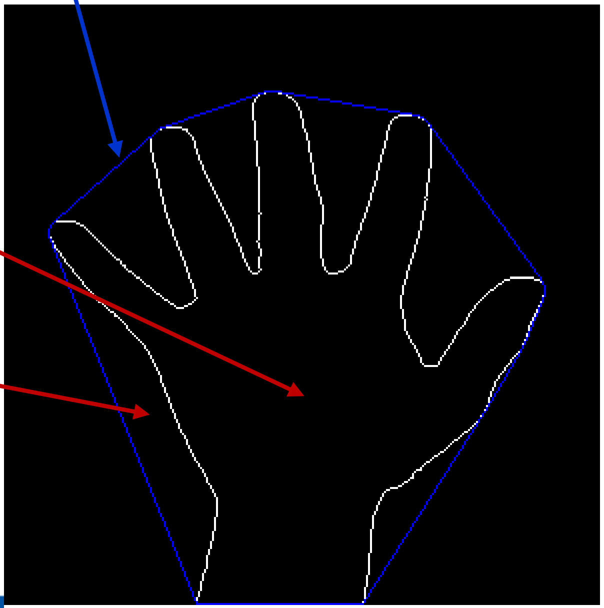
Eroded with a 5x5 SE of 1s

| Connected component | No. of pixels in connected comp |
|---|---|
| 01 | 11 |
| 02 | 9 |
| 03 | 9 |
| 04 | 39 |
| 05 | 133 |
| 06 | 1 |
| 07 | 1 |
| 08 | 743 |
| 09 | 7 |
| 10 | 11 |
| 11 | 11 |
| 12 | 9 |
| 13 | 9 |
| 14 | 674 |
| 15 | 85 |

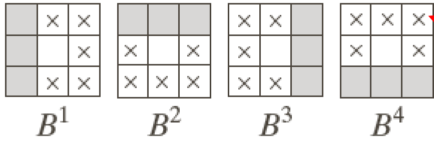
# Convex Hull

- Convex (凸): set *A* is said to be *convex* if the straight line segment joining any two points in *A* lies entirely within *A*.
- Convex Hull (凸包): *convex hull H* of an arbitrary set *S* is the smallest convex set containing *S*
- Convex Deficiency (凸缺):

set difference *H - S* is called the *convex deficiency* of *S*



# Convex Hull

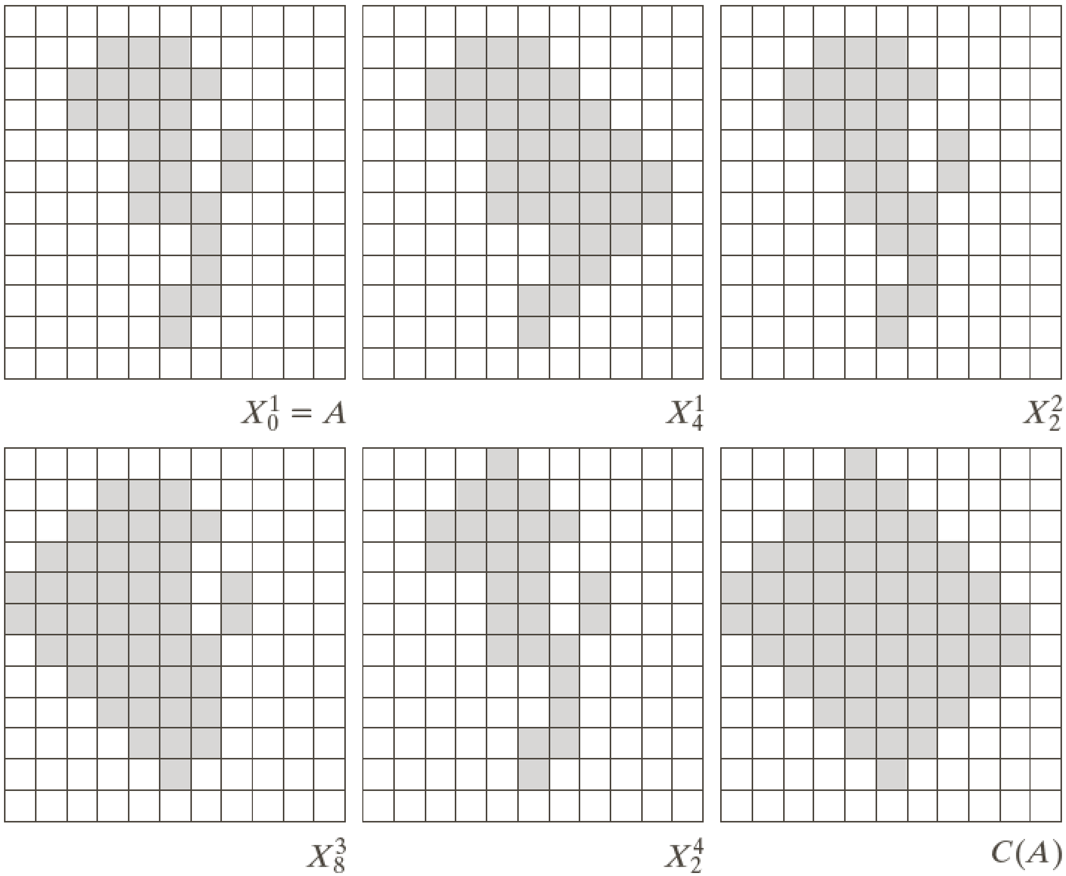



Don't care

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

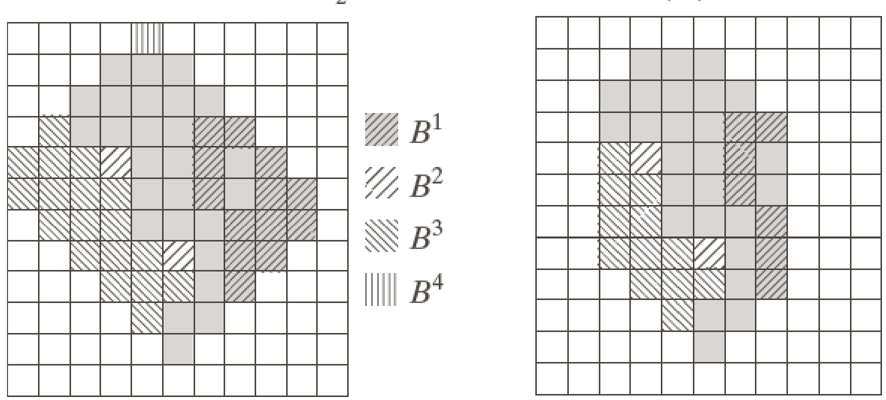
$$i = 1, 2, 3, 4$$

$$k = 1, 2, 3, \ldots$$




$$X_0^i = A \qquad D^i = X_k^i$$

$$C(A) = \bigcup_{i=1}^{4} D^i$$




Limiting growth of the convex hull

# Thinning

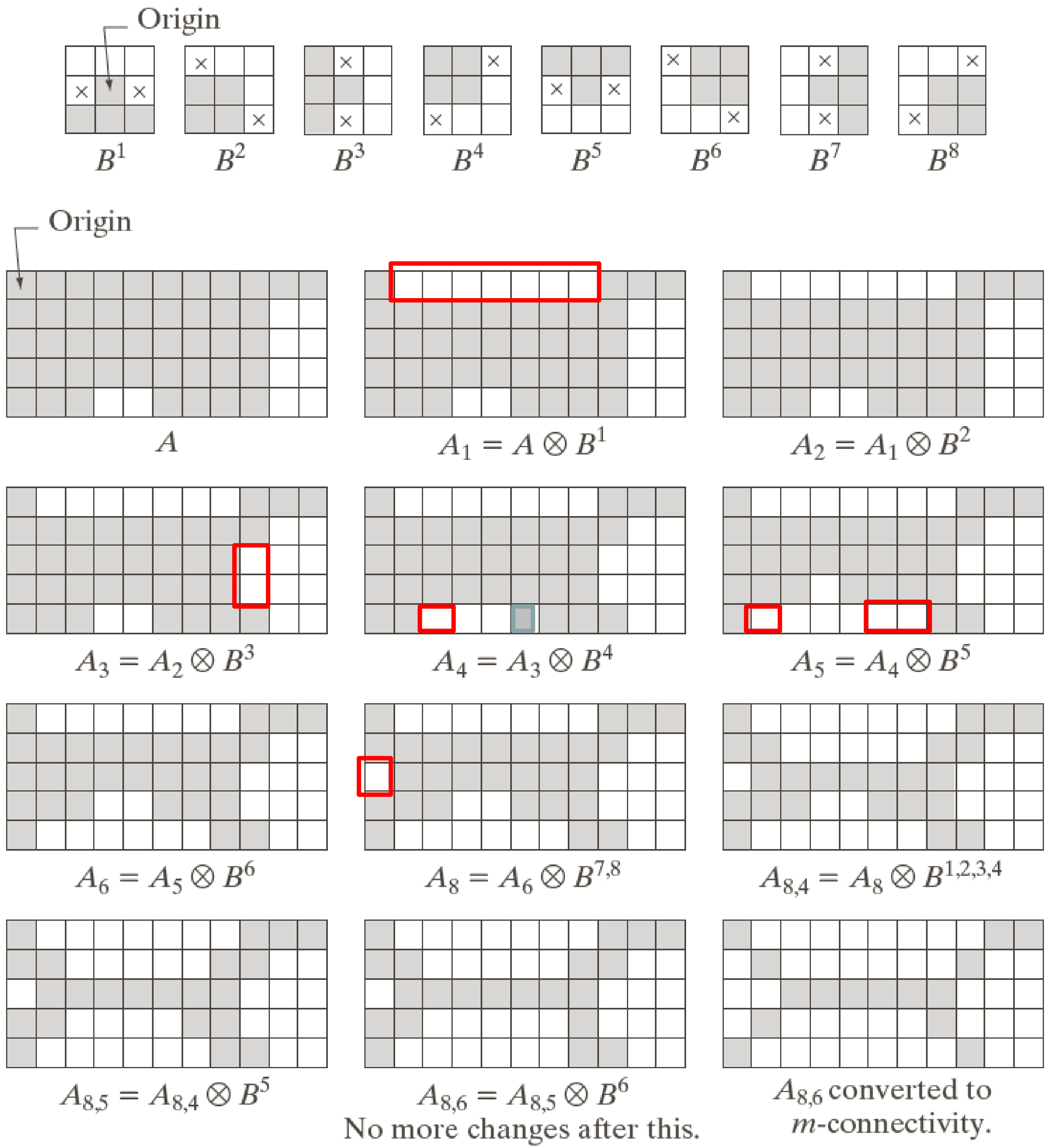
$$A \otimes B = A - (A \circledast B) = A \cap (A \circledast B)^c$$

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

$B^i$ is a rotated version of $B^{i-1}$

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

# Thinning Example



# Thickening

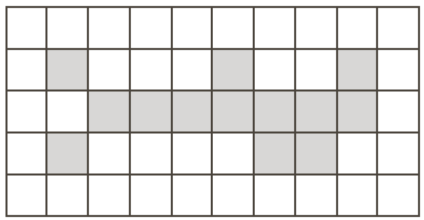
$$A \odot B = A \cup (A \circledast B)$$

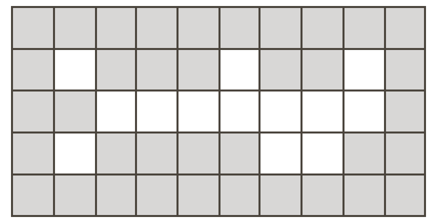
$$A \odot \{B\} = ((\ldots((A \odot B^1) \odot B^2) \ldots) \odot B^n)$$

Duality:

Thickening the foreground = Thinning the background

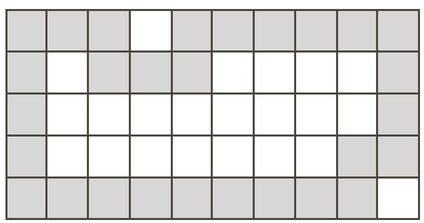
$A$

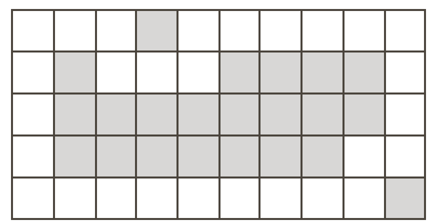




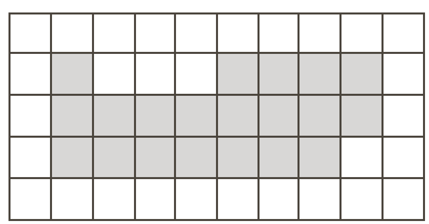
$A^c$

Thinning of $A^c$





Thickening of $A$



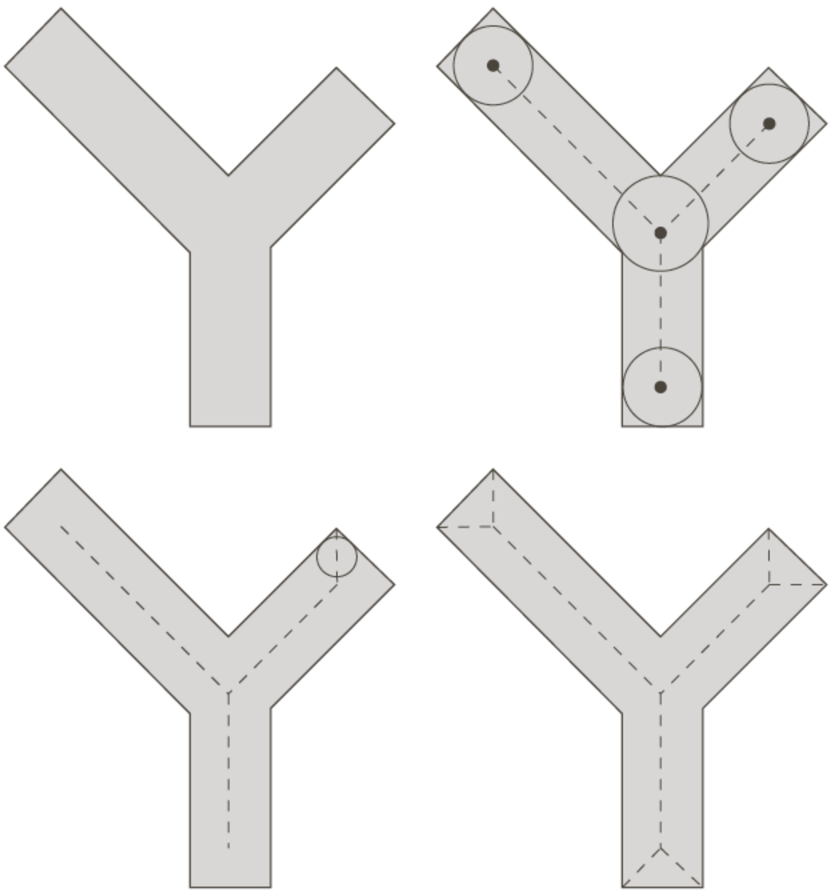
Remove disconnected points

# Skeletons（骨架、中轴）

**(a)** If $z$ is a point of $S(A)$ and $(D)_z$ is the largest disk centered at $z$ and contained in $A$, one cannot find a larger disk (not necessarily centered at $z$) containing $(D)_z$ and included in $A$. The disk $(D)_z$ is called a *maximum disk*.

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

- 区域边界内切圆的圆心的集合
- 火烧草地：边界上同时点火，假设火蔓延的速度处处相同，火线相遇的地方构成中轴



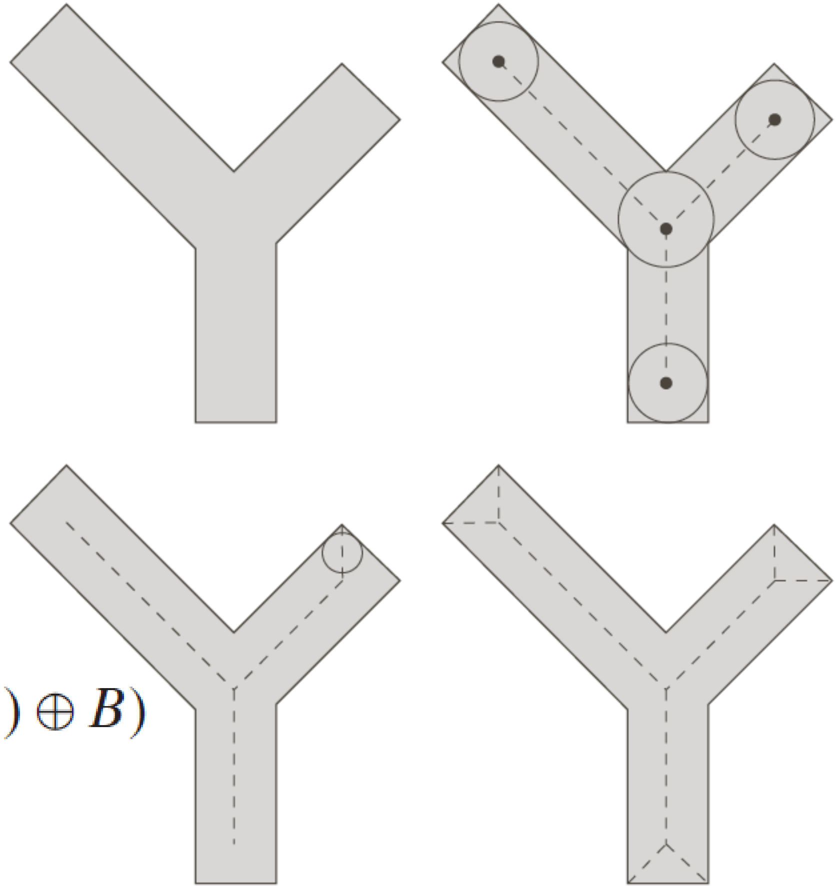
# Skeletons（骨架、中轴）

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

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

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

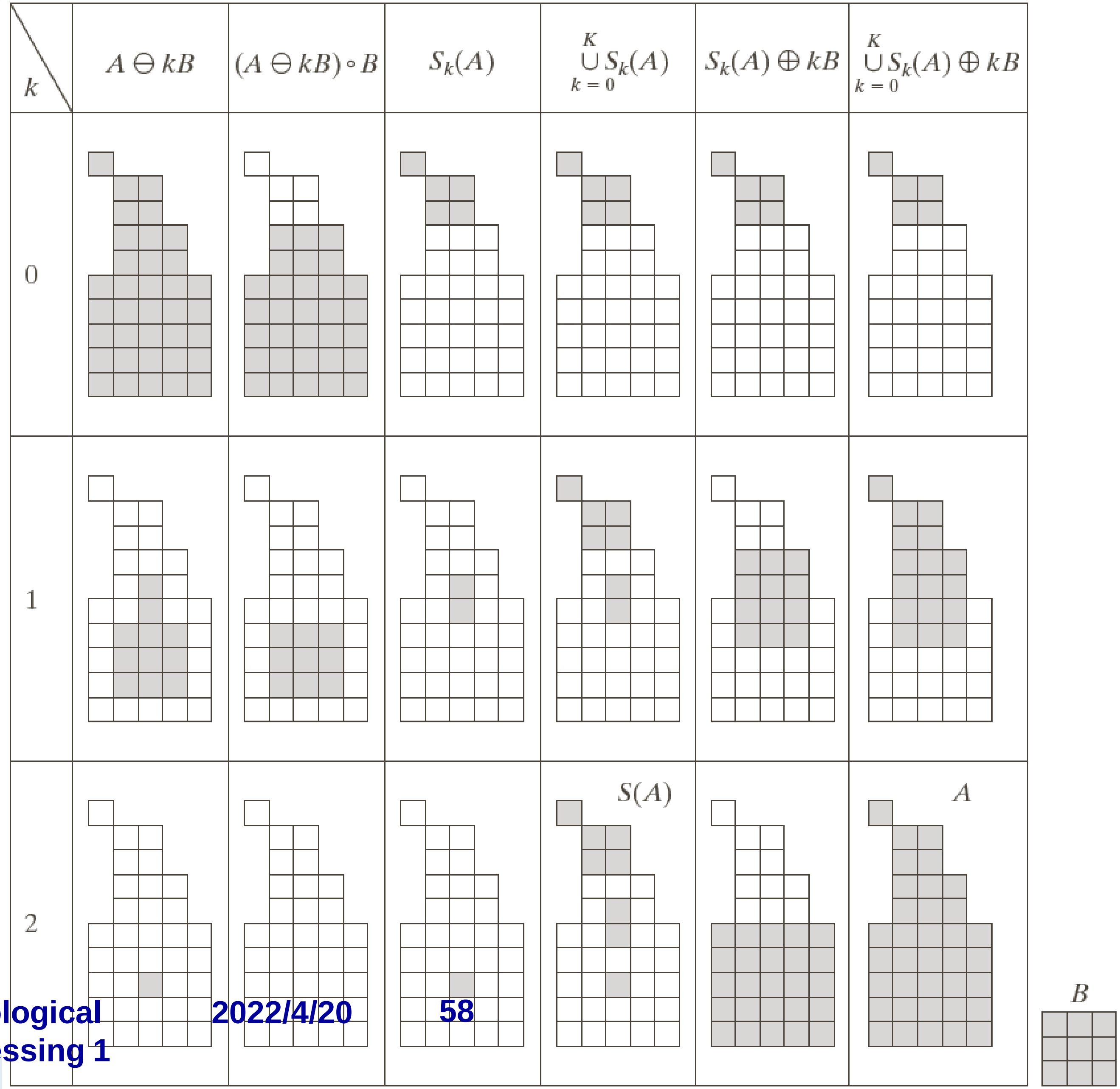
$$K = \max\{k \mid (A \ominus kB) \neq \varnothing\}$$



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

$$(S_k(A) \oplus kB) = ((\ldots((S_k(A) \oplus B) \oplus B) \oplus \ldots) \oplus B)$$

# Skeletons Example



# Pruning

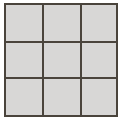
- Problem: “spurs” (parasitic components) after thinning and skeletonizing

$$X_1 = A \otimes \{B\}$$

3 times

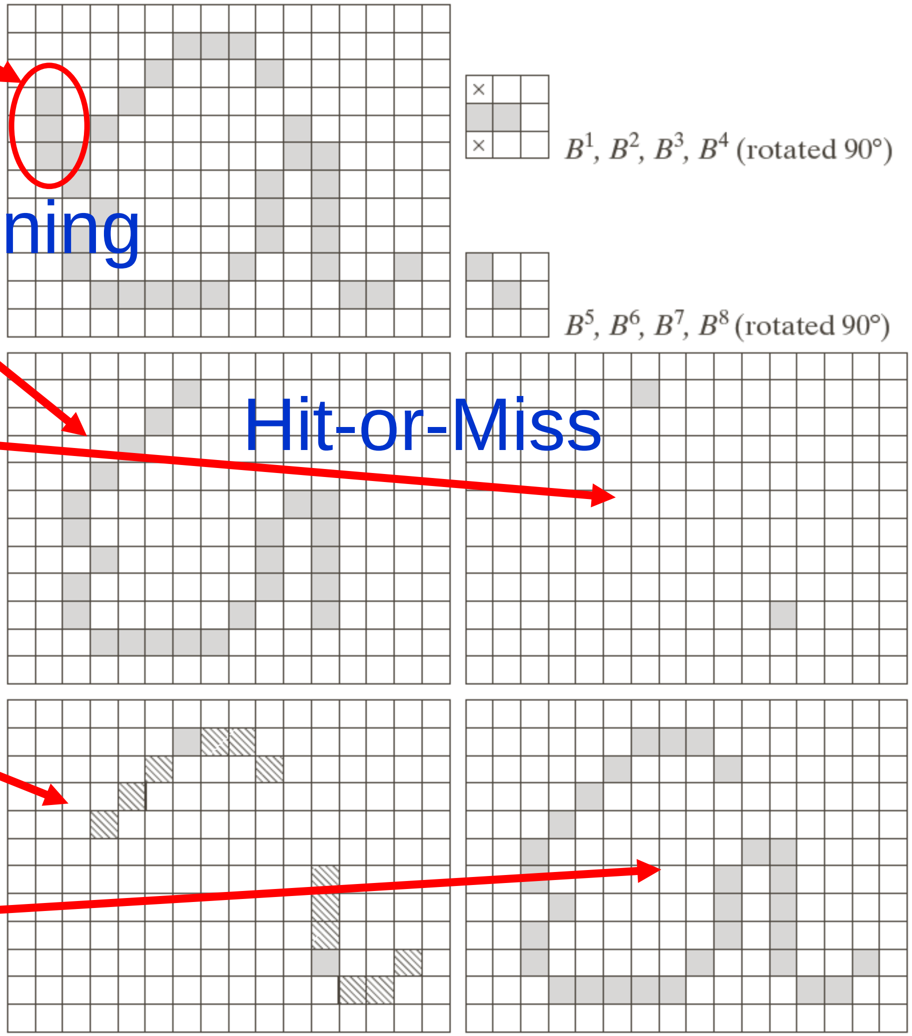
$$X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$$

$$X_3 = (X_2 \oplus H) \cap A$$



3 times

$$X_4 = X_1 \cup X_3$$



# Assignments

- 9.8, 9.9, 9.23, 9.26

课后作业题目请对照参考第4版英文原版

- 第4次编程作业

从Laboratory Projects_DIP3E.pdf的Proj09-xx中选做1个题目。也可针对DIP4E Chapter 9内容，自拟任务。

# Assignments

每个编程作业要求递交1份实验报告，命名“学号姓名_prjX.pdf”，内容提纲包括：

- 实验任务： 描述本次实验的任务，即所选择的ProjXX-xx题目，或自拟题目。
- 算法设计：理论上描述所设计的算法。
- 代码实现：描述编程环境，给出自己编写的核心代码。
- 实验结果：描述具体的实验过程，给出每个小实验的输入数据、算法参数和实验结果，并对结果做简要的讨论。
- 总结：简要总结本次实验的技术内容，以及心得体会