上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

IVM
Image, Video, and Multimedia Communications Laboratory

Digital Image Processing

Hongkai Xiong
熊红凯
http://ivm.sjtu.edu.cn

电子工程系
上海交通大学

Today

- Morphological Image Processing

Image Processing —Image Understanding —Computer Vision

- Low-level Processing: both input and output are images
  Noise Reduction;
  Image Enhancement;
  Image Sharpening
- Mid-level Processing: input images, output attributes of those images
  Image Segmentation
  Image Indexing (Feature Extraction)
- High-level Processing: related to computer vision
  Image Analysis and Understanding

Preview

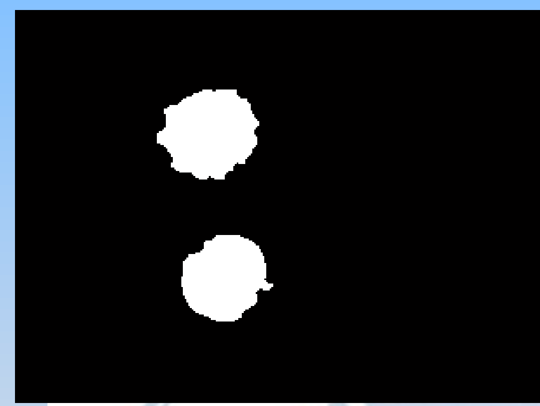
- Morphology
  - About the **form** and **structure** of animals and plants
- Mathematical morphology
  - Using set theory
  - Extract image component
  - Representation and description of region shape

Mathematic Morphology

- Used to extract image components that are useful in the representation and description of region shape, such as
  - boundaries extraction
  - skeletons
  - convex hull
  - morphological filtering
  - thinning
  - pruning

Z^2 and Z^3

- set in mathematic morphology represent objects in an image



  - binary image (0 = white, 1 = black) : the element of the set is the coordinates (x,y) of pixel belong to the object ⇨ Z^2
  - gray-scaled image : the element of the set is the coordinates (x,y) of pixel belong to the object and the gray levels ⇨ Z^3

Outline

Binary images:

- Preliminaries – set theory
- Dilation and erosion
- Opening and closing
- Hit-or-miss transformation
- Some basic morphological algorithms
- Morphological operations on gray images

Preliminaries – Set Theory

- Our interest in this part is sets in Z^2, where each element denotes the coordinates of an object pixel.
  - If $a=(a_1, a_2)$, we write $a \in A$ if a is an element in A.
  - $a \notin A$ if a is *not* an element in A.
- The *null* or *empty* set is denoted by $\varnothing$.
- We use braces, { ·}, to specify the content of a set. For example, $C=\{w|w=-d$, for $d \in D\}$.

Set Operations

- A is a subset of B: every element of A is an element of another set B

$$A \subseteq B$$

- Union

$$C = A \cup B$$

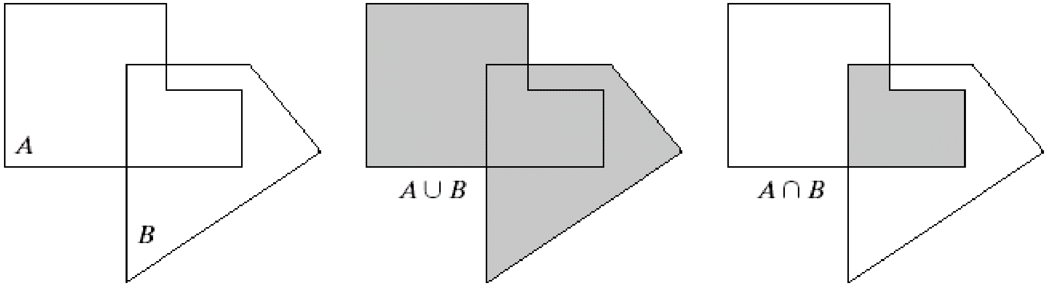
- Intersection

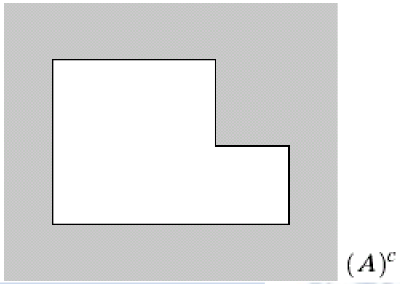
$$C = A \cap B$$

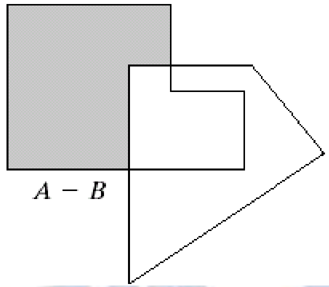
- Mutually exclusive

$$A \cap B = \varnothing$$

Graphical Examples







$$A \bigcup B = \{w \mid w \in A \text{ or } w \in B\}$$

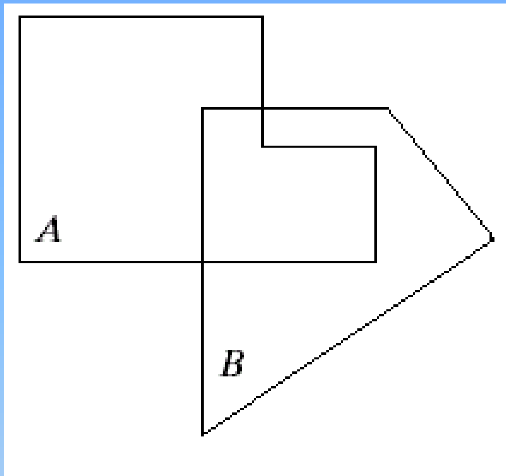
$$A \bigcap B = \{w \mid w \in A \text{ and } w \in B\}$$

(A and B are said to be disjoint or mutually exclusive if $A \bigcap B = \varnothing$)

$$A^c = \{w \mid w \notin A\}$$

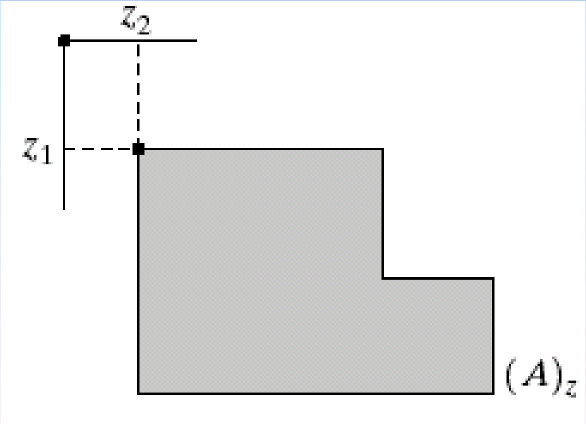
$$A - B = \{w \mid w \in A, w \notin B\} = A \bigcap B^c$$

Additional Definitions



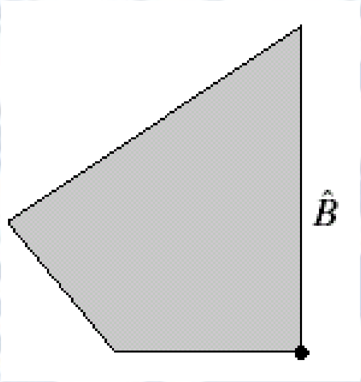
translation

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



reflection

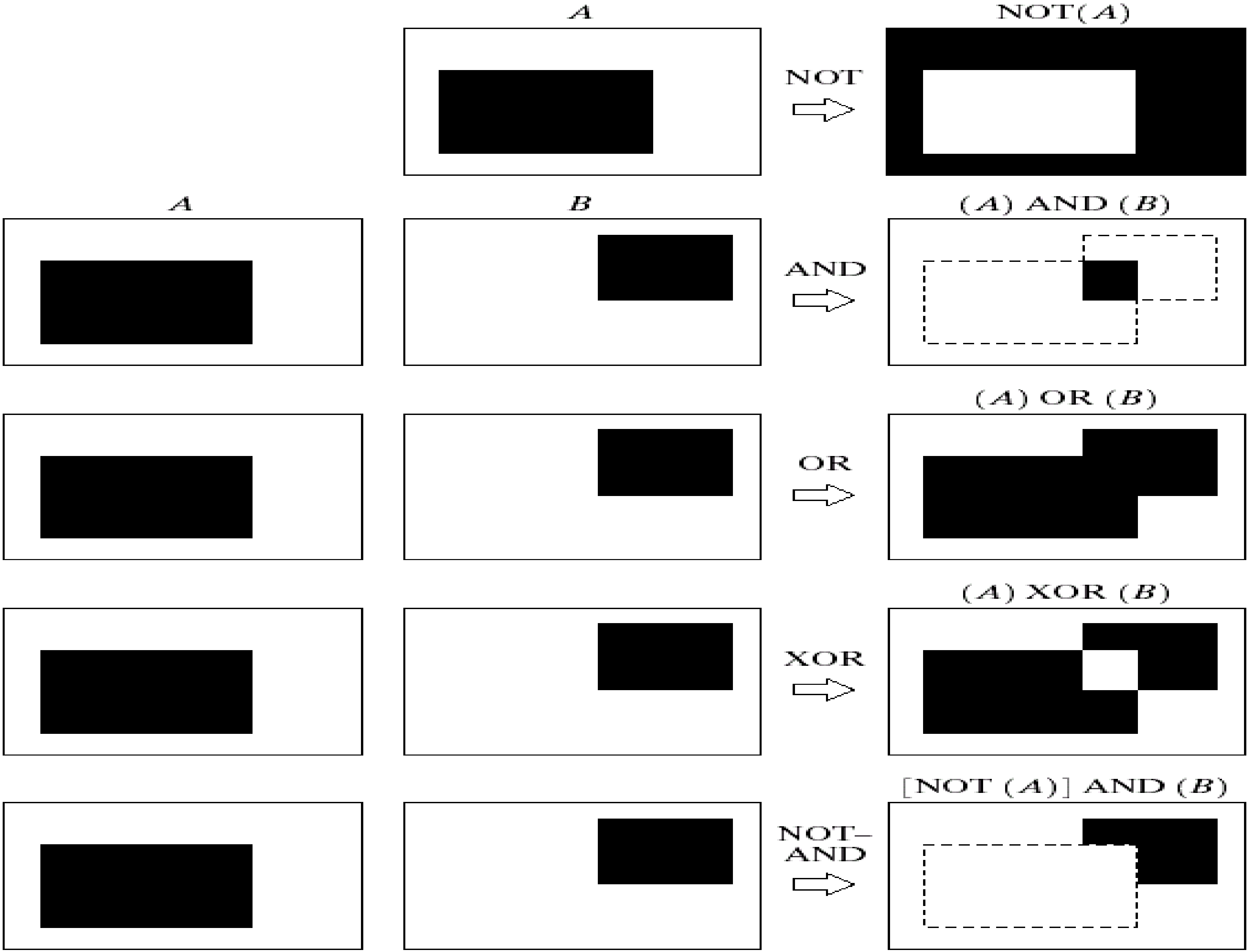
$$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$$



Logic Operations on Binary Images

- Functionally complete operations
  - AND, OR, NOT

p	q	p AND q (also $p \cdot q$)	p OR q (also $p + q$)	NOT (p) (also $\bar{p}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0


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

Basic Concepts of Set Theory

- A is a set in Z^2 , a=(a_1,a_2) an element of A, a$\in$A
  If not, then a$\notin$A
  - $\varnothing$: null (empty) set
  - A subset of B: A$\subseteq$B
  - Union of A and B: C=A$\cup$B
  - Intersection of A and B: D=A$\cap$B
  - Disjoint sets: A$\cap$B= $\varnothing$
  - Complement of A: $A^c = \{w \mid w \notin A\}$
  - Difference of A and B: A-B={w|w $\in$ A, w $\notin$ B}= $A \cap B^c$
  - Reflection of B: $\hat{B} = \{w \mid w = -b, b \in B\}$
  - Translation of A by z=(z_1,z_2): $(A)_z = \{c \mid c = a + z, a \in A\}$

Outline

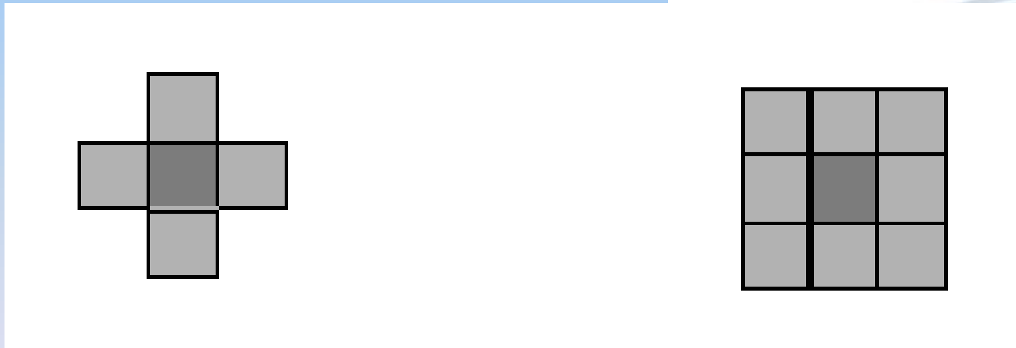
- Preliminaries
- Dilation and erosion
- Opening and closing
- Hit-or-miss transformation
- Some basic morphological algorithms
- Morphological operations on gray images

Dilation and Erosion

- These two operations are fundamental to morphological processing.
- Dilation: To expand an image or a region A by a template or a structuring element B.
  - enlarge an object along its boundary.
  - The dilation process consists of obtaining the reflection of B about its origin and then shifting this reflection by some displacement x.
- Erosion: To shrink an image or a region A by a template or a structuring element B.

Structuring Element (SE)

- Small set to probe the image under study
- Shape and size must be adapted to geometric properties for the objects



Dilation

- With A and B are sets in Z^2, the dilation of A by B, denoted $A \oplus B$, is defined as

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

  - Other interpretation:

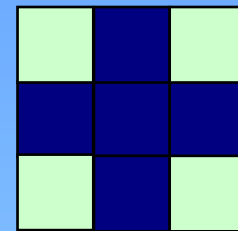
$$A \oplus B = \left\{ z \,\middle|\, \left[(\hat{B})_z \cap A\right] \subseteq A \right\}$$

    - B is the *structuring element*.
    - The dilation of A by B is the set of all displacements, z, such that the reflection of B and A overlap by at least one element.

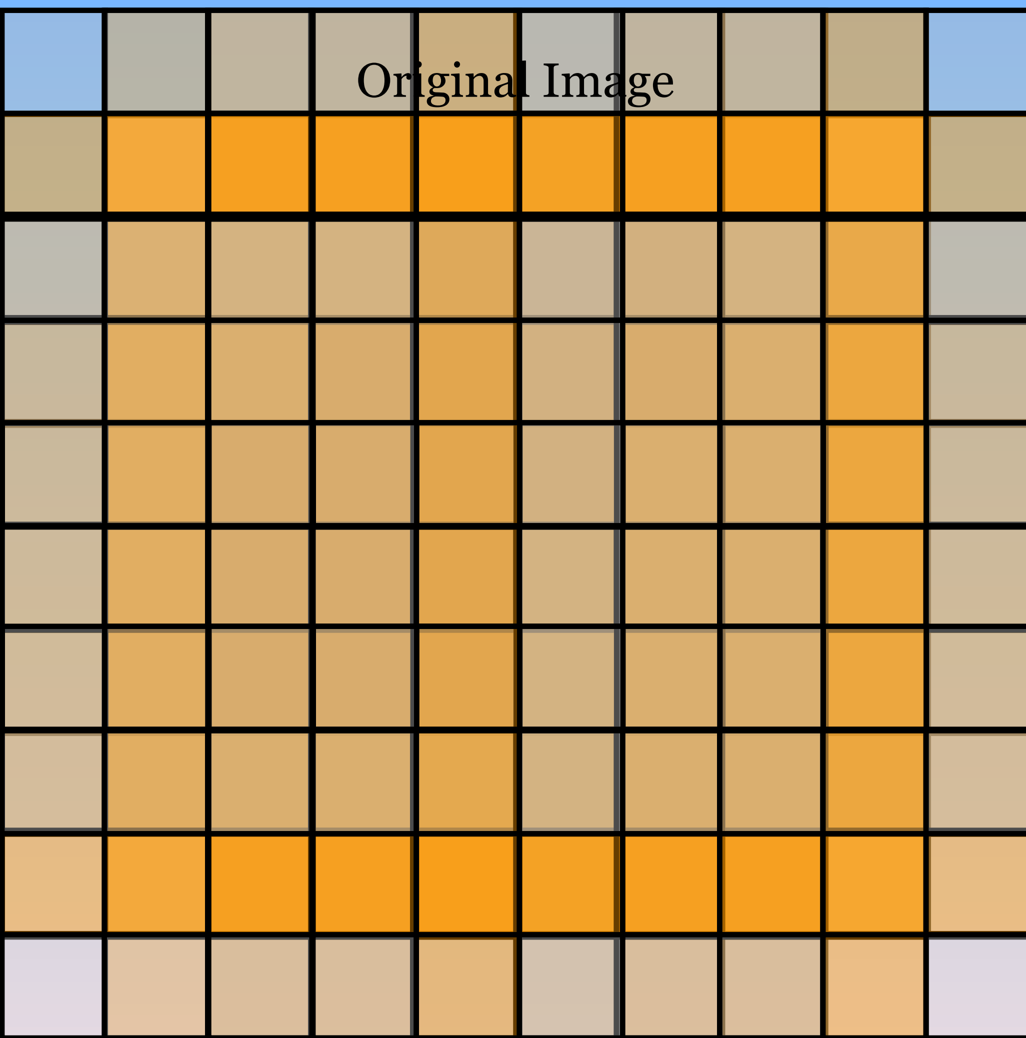
The Implementation of Dilation

- Given a binary image A and the structuring element B, construct a duplicate of A, denoted by g.
- For each pixel $p = A(x, y)$, do the following:
  - If p is black:
    - If p is at the boundary (any of the 4-adjacent neighbors is white) of the object, center the origin of B at (x, y) in g, and fill black in the pixels that B covers.
- Return g.

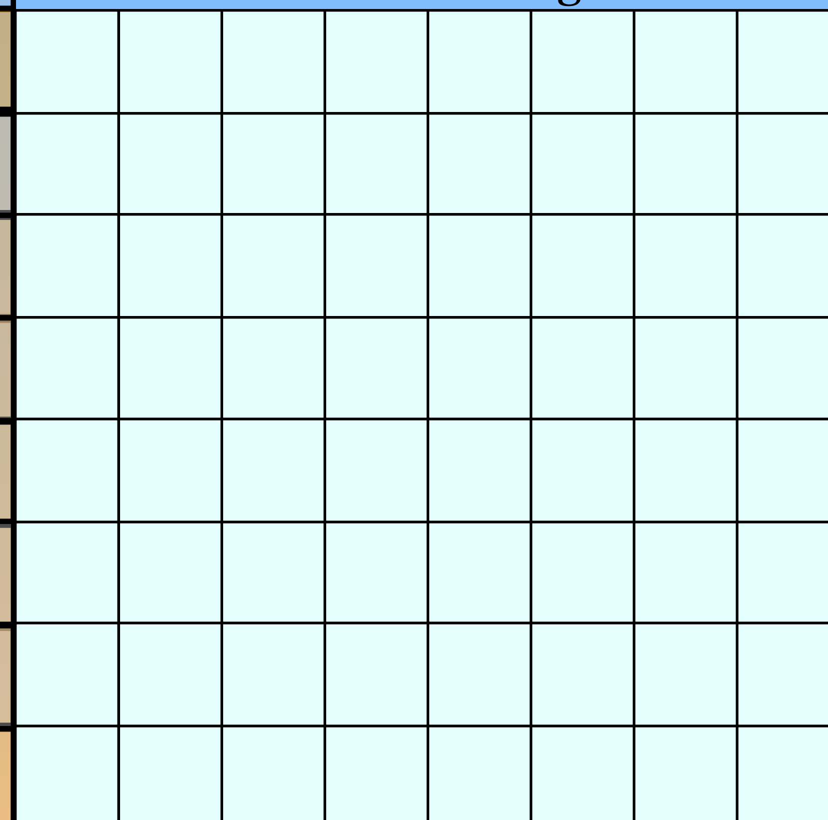
Dilation Example



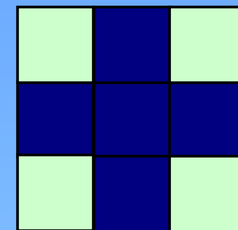
Structuring Element



Processed Image

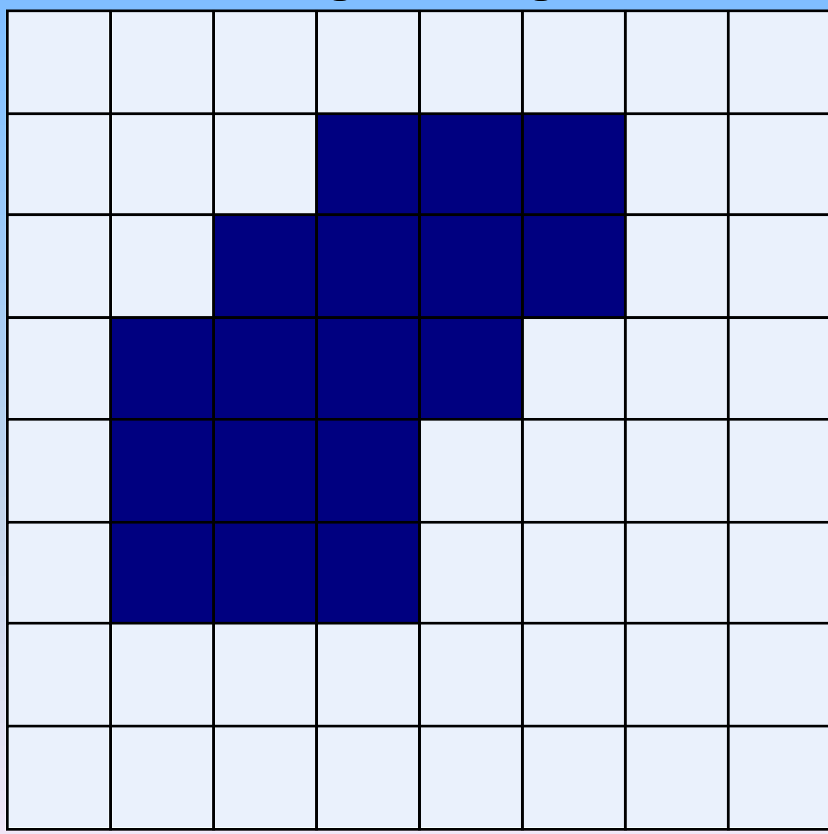


Dilation Example

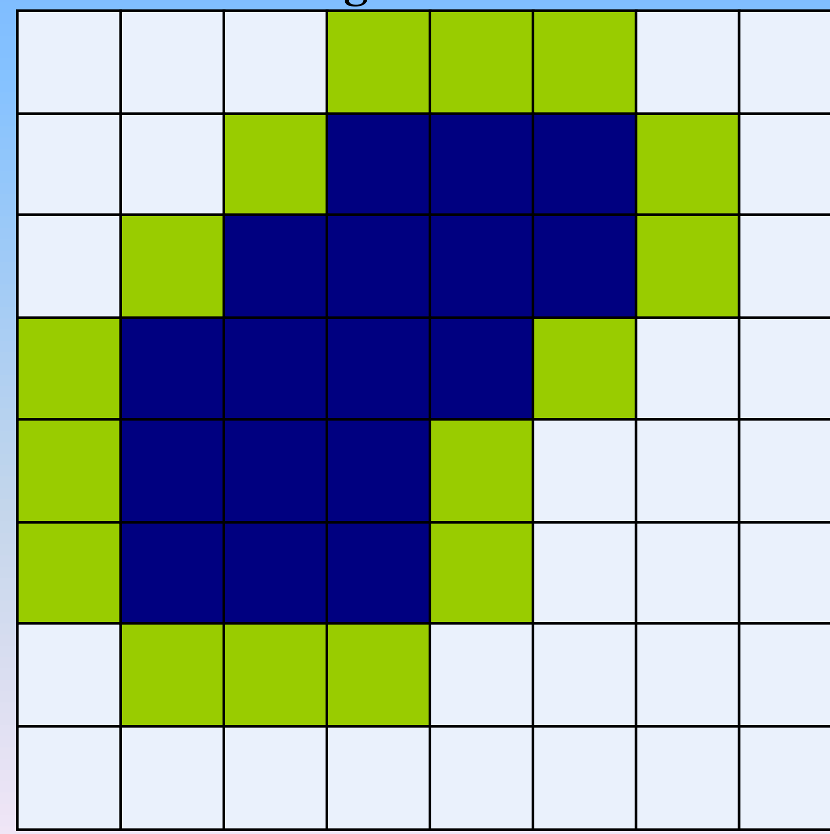


Structuring Element

Original Image



Processed Image With Dilated Pixels

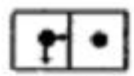


The Illustration of Dilation

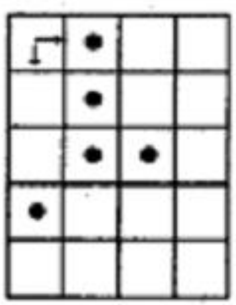
$$A_t = \{c \in E^N \mid c = a + t \text{ for some } a \in A\}$$

$$A = \{(0,1),(1,1),(2,1),(2,2),(3,0)\}$$

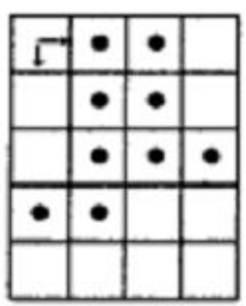
$$B = \{(0,0),(0,1)\}$$



B

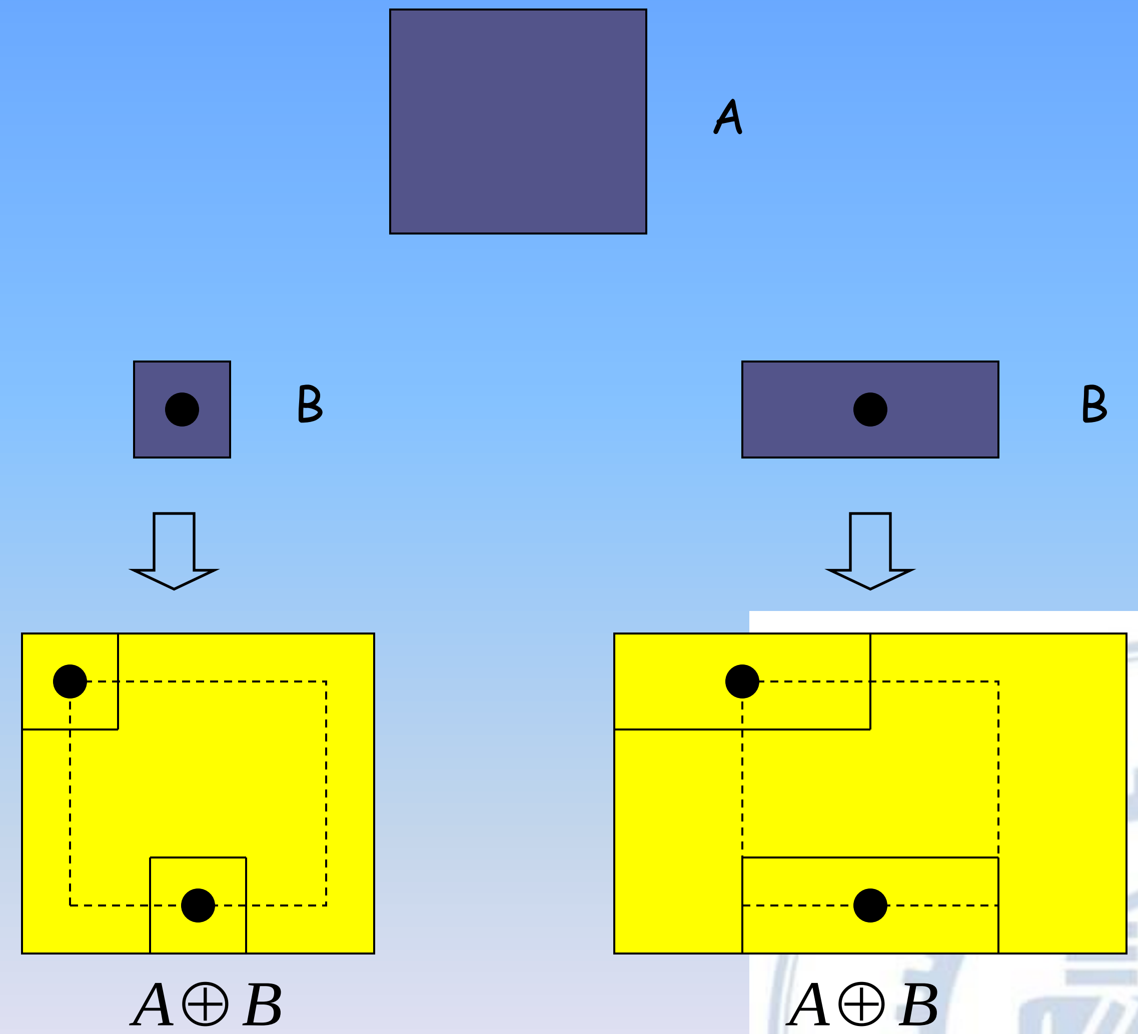


A



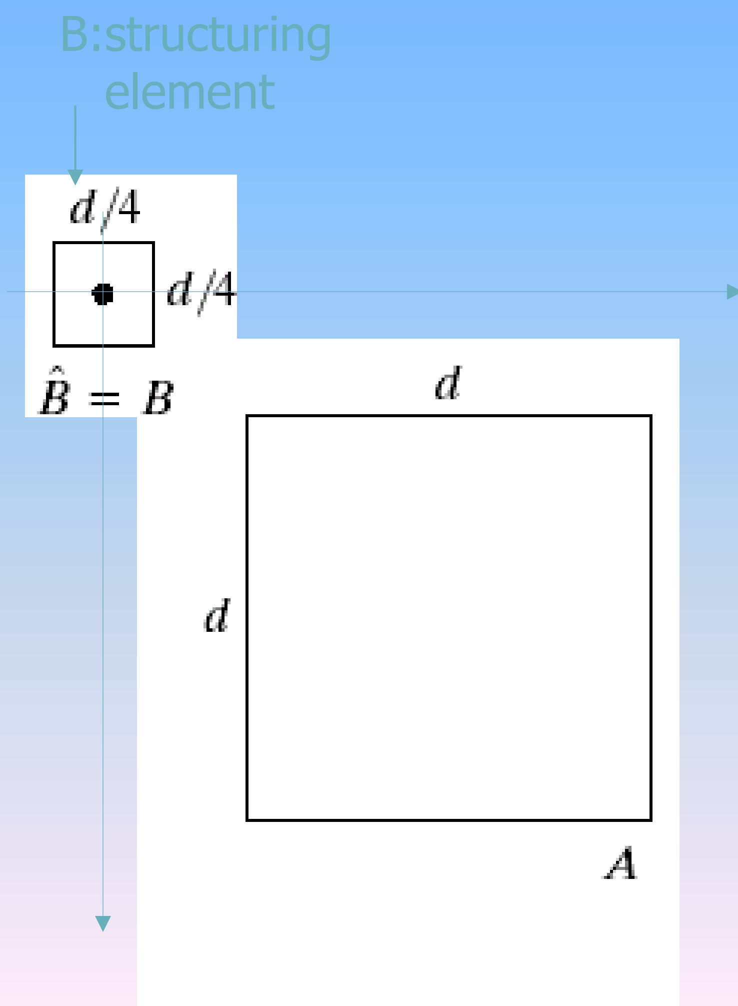
A ⊕ B

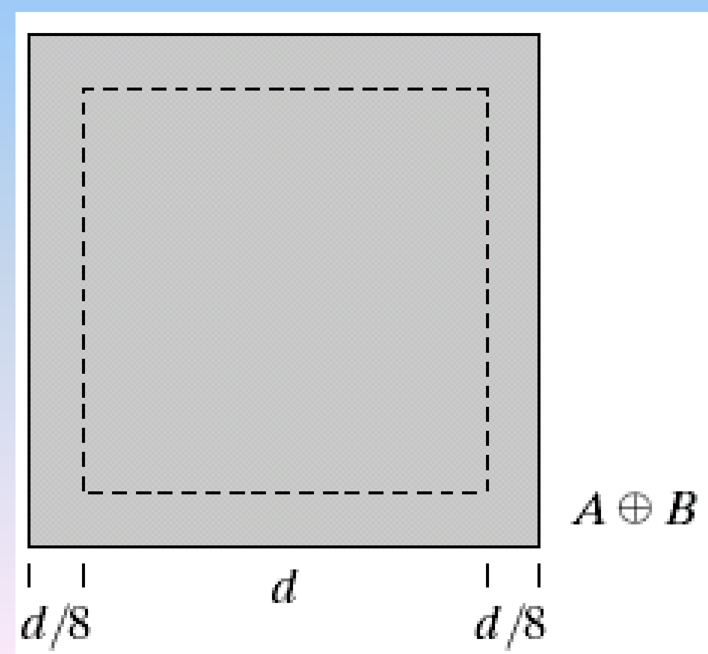
$$A \oplus B = \{(0,1),\ (1,1),\ (2,1),\ (3,0)$$
$$(0,2),\ (1,2),\ (2,2),\ (2,3),\ (3,1)\}$$



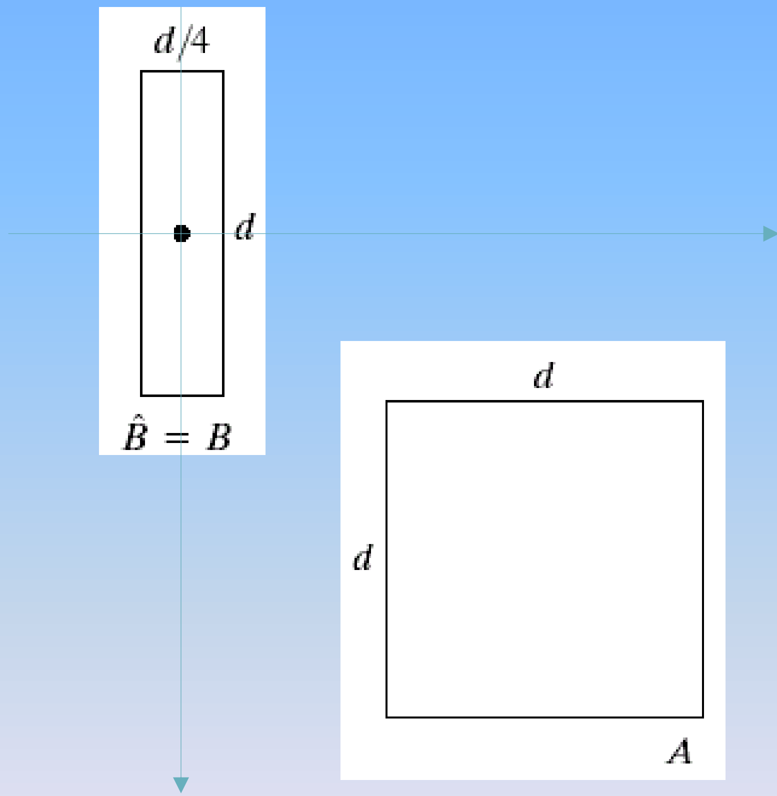
Dilation of a region A by a *structuring element* B.

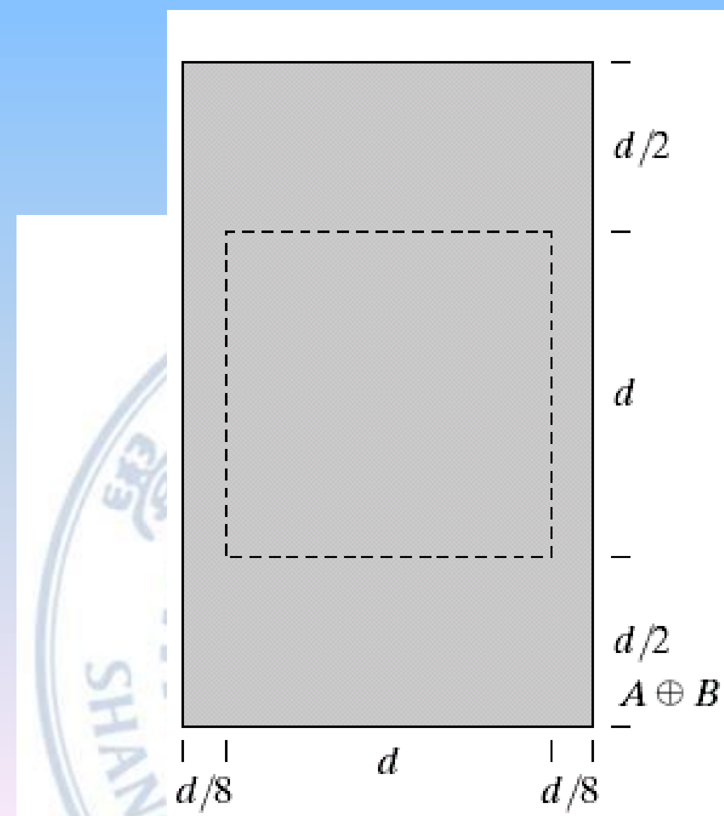
The Illustration of Dilation





The Illustration of Dilation



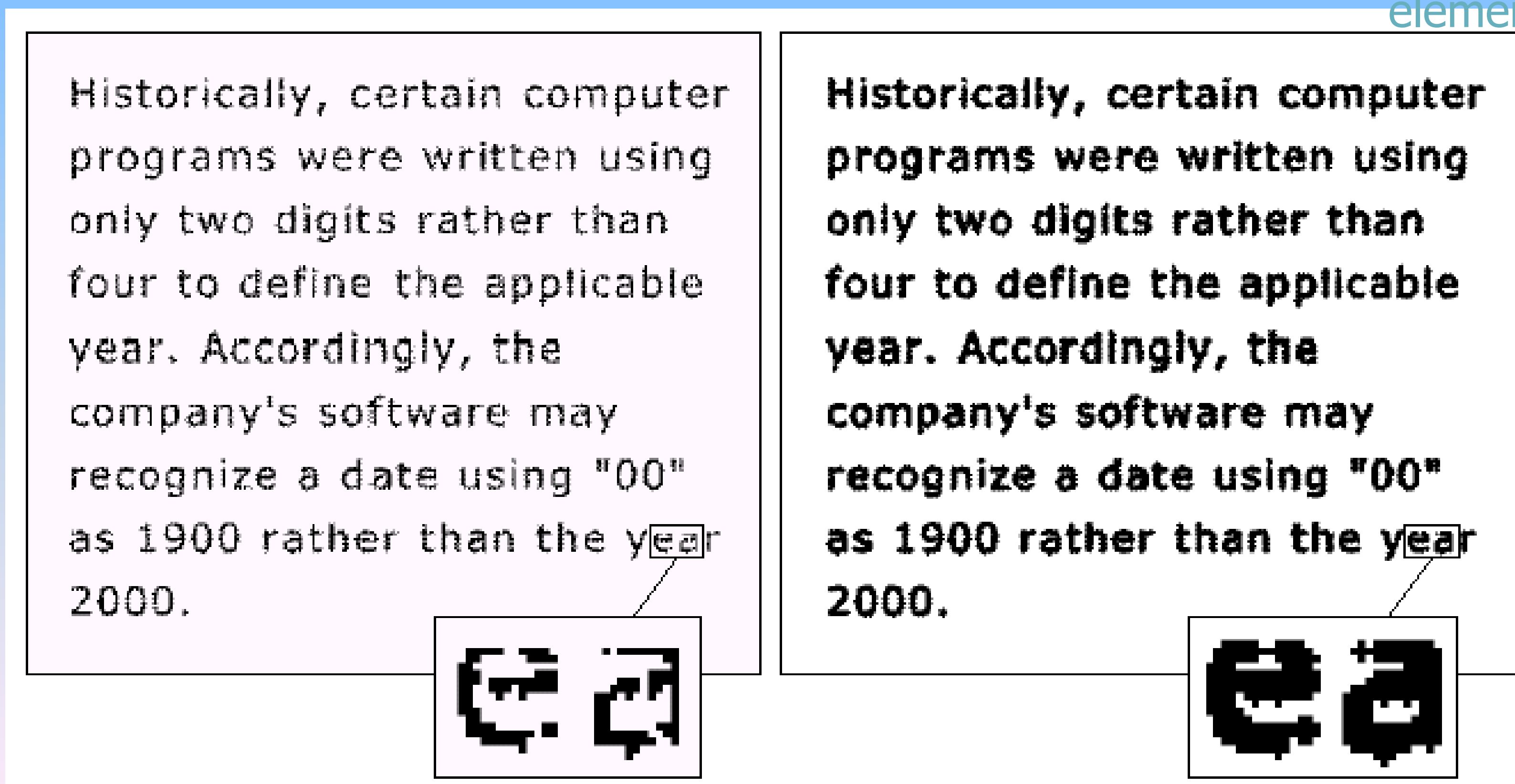


Application of Dilation

- Bridging gaps in images: increase size, fill gap

0	1	0
1	1	1
0	1	0

Structuring element

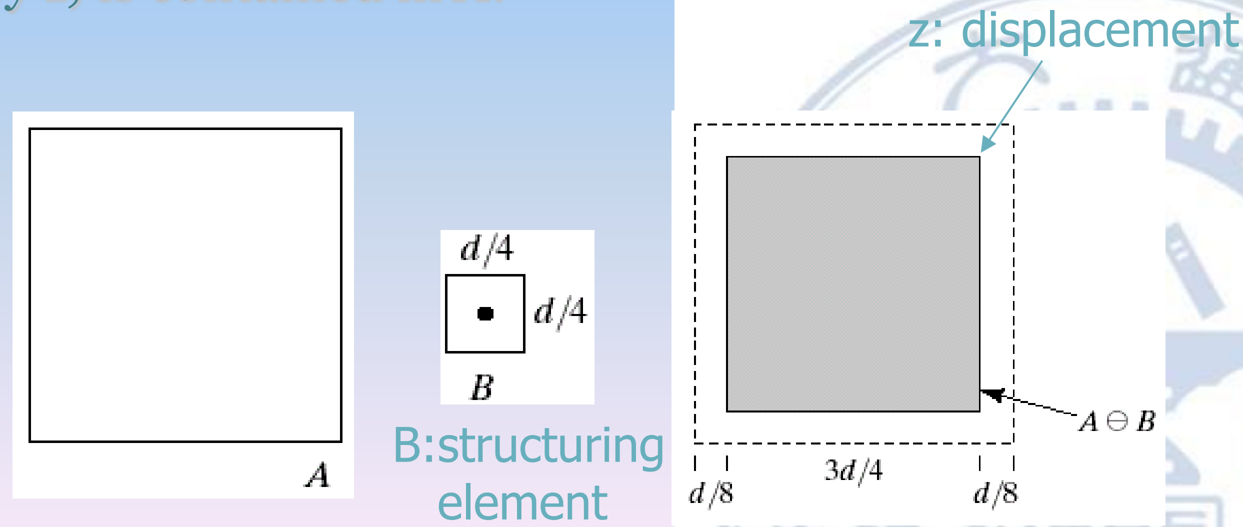


Erosion

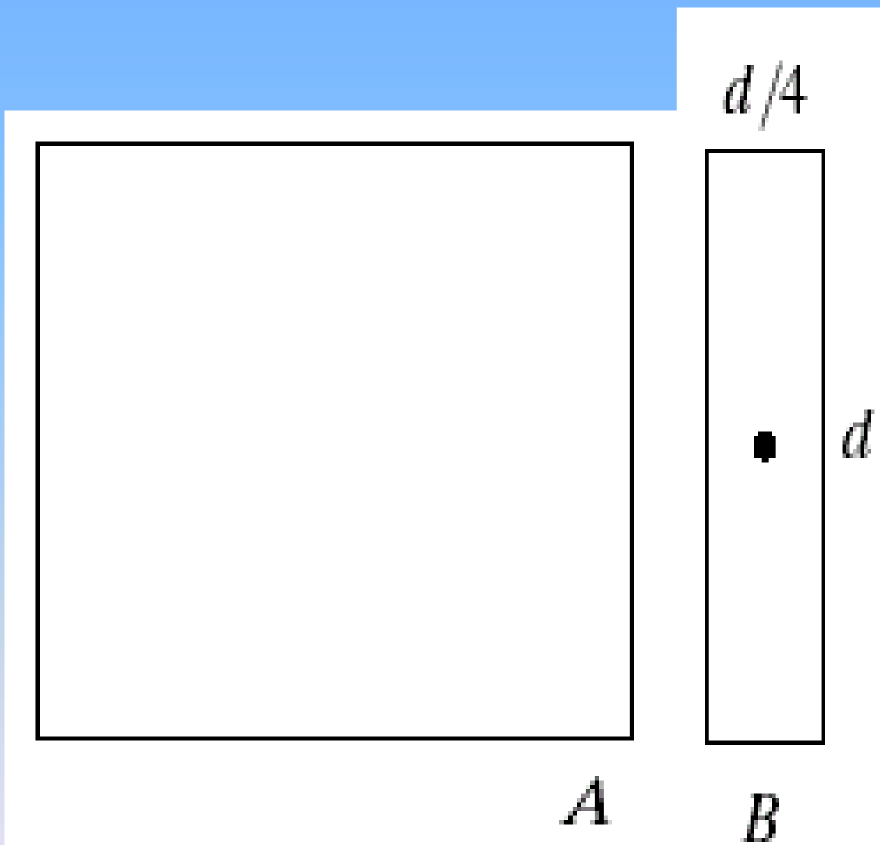
- With *A* and *B* are sets in Z^2, the erosion of *A* by *B*, is defined as

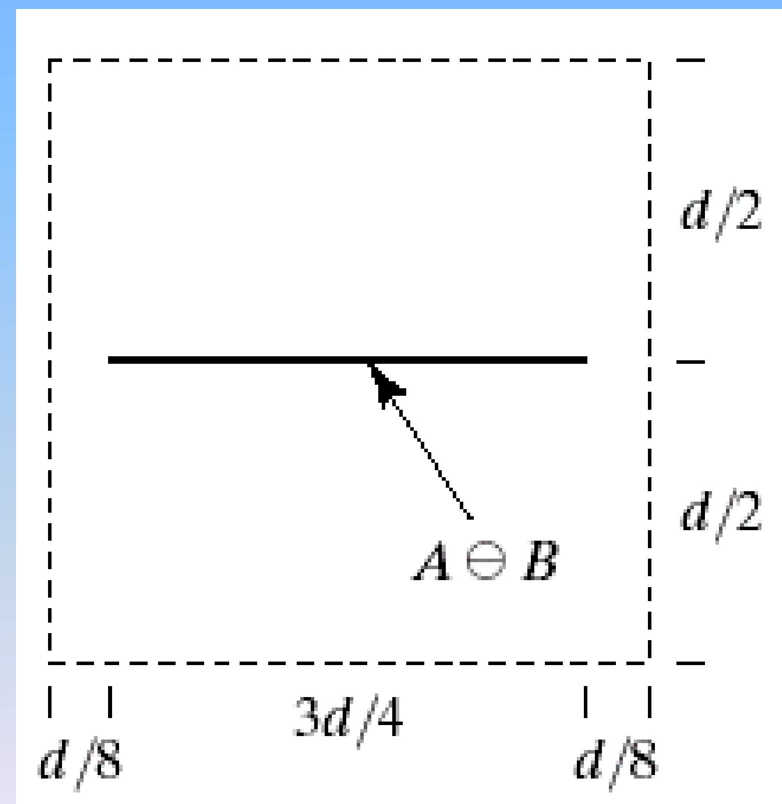
$$A \ominus B = \left\{ z \middle| (B)_z \subseteq A \right\}$$

  - The erosion of *A* by *B* is the set of all points *z* such that *B*, translated by *z*, is contained in *A*.



Erosion

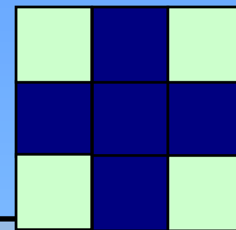




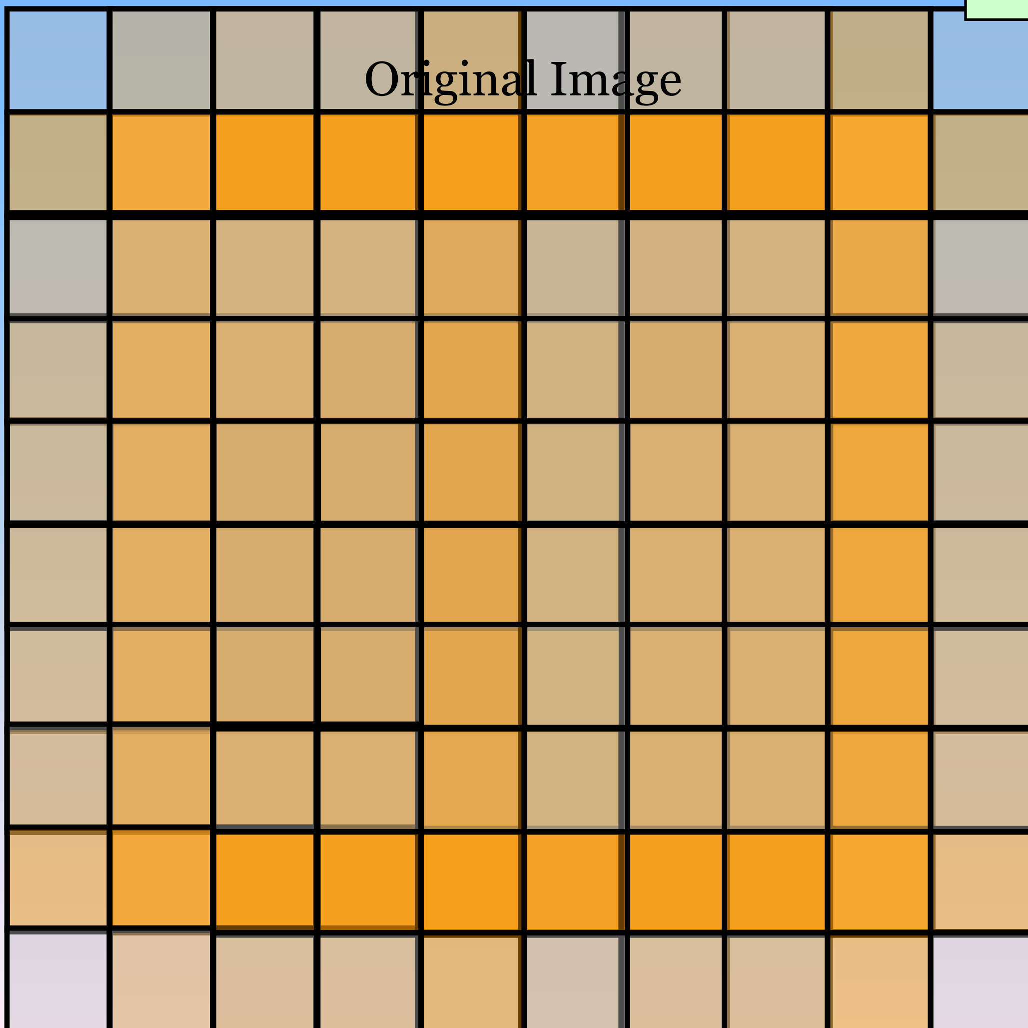
The Implementation of Erosion

- Given a binary image A and the structuring element B, construct a duplicate of A, denoted by g.
- For each pixel $p = A(x, y)$, do the following:
  - If p is white:
    - If p is adjacent to the boundary of the object, center the origin of B at (x, y) in g, and fill white in the pixels that B covers.
- Return g.

Erosion Example

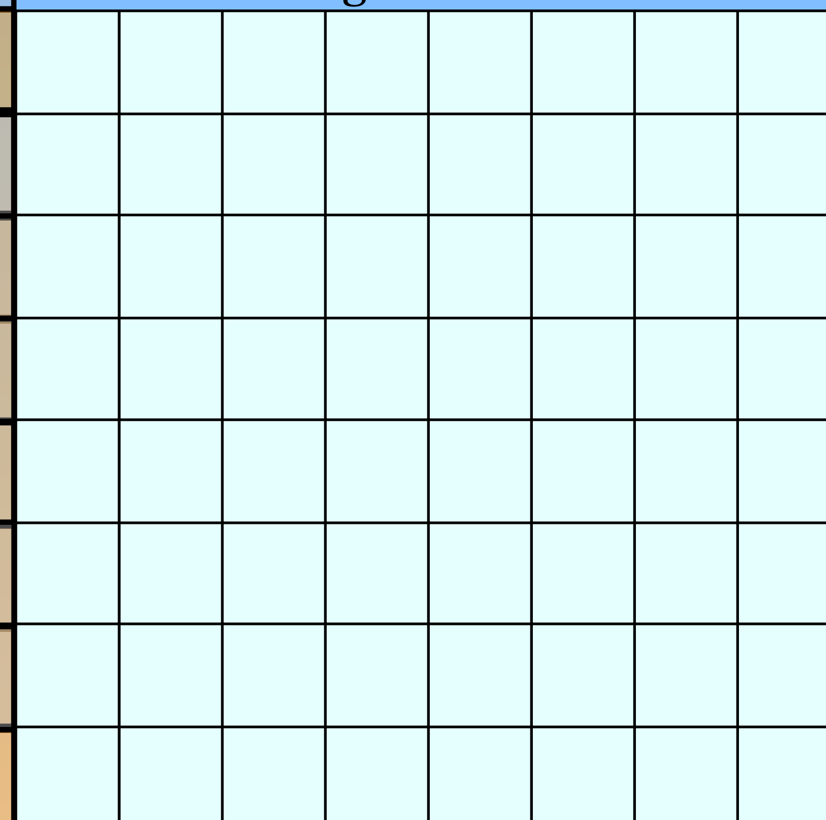


Structuring Element

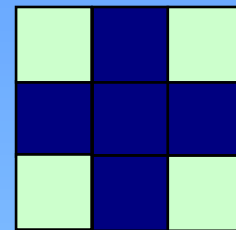


Original Image

Processed Image With Eroded Pixels

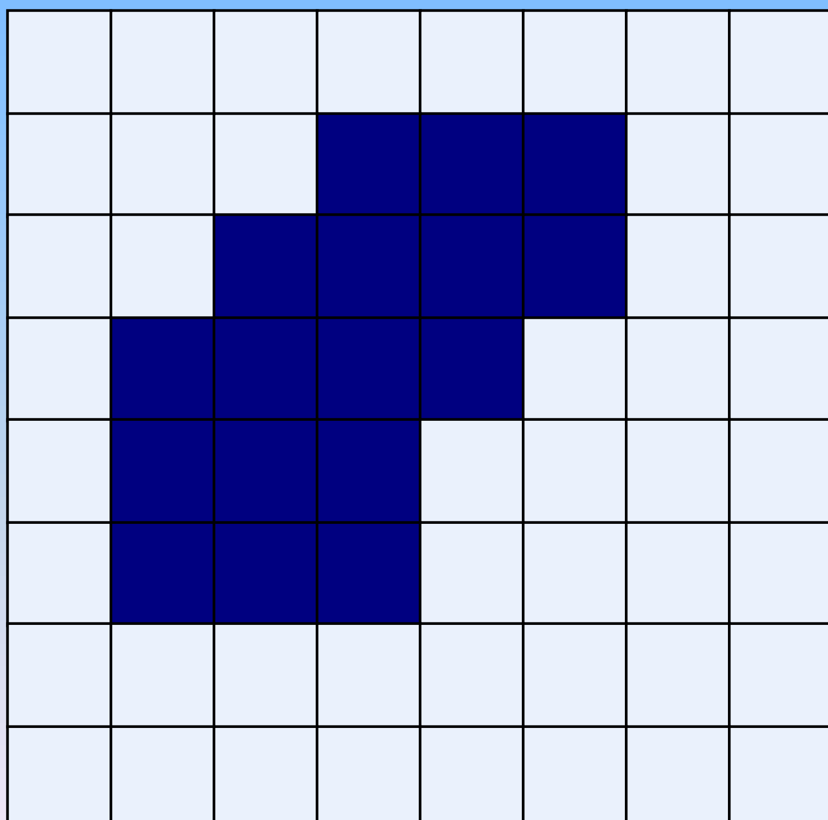


Erosion Example

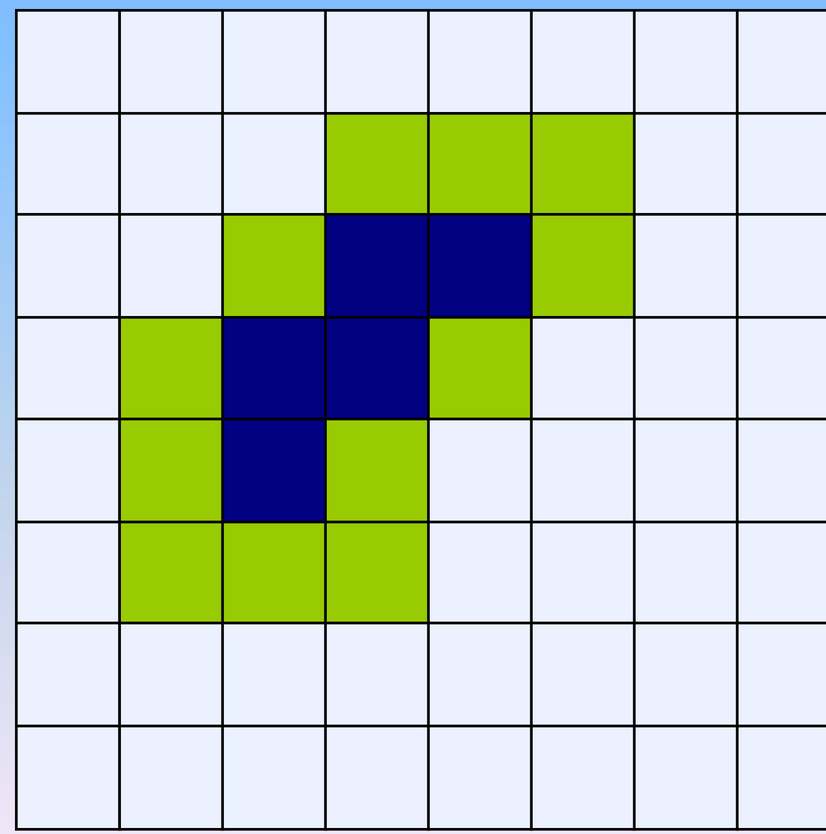


Structuring Element

Original Image



Processed Image

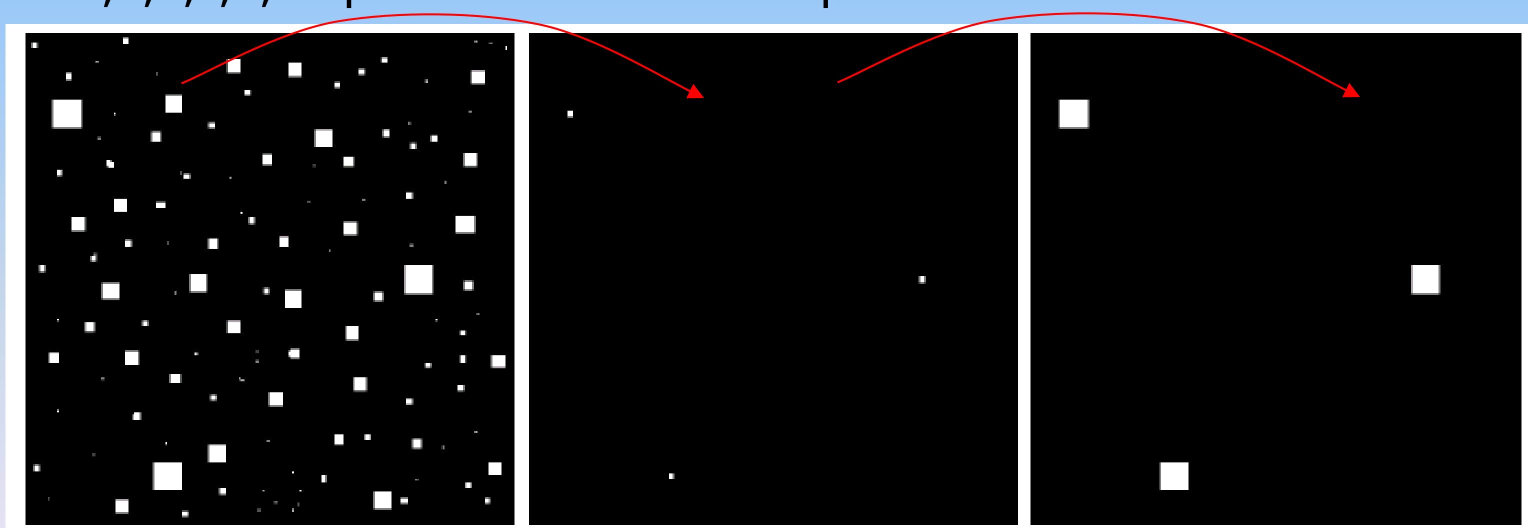


Application of Erosion

- One of the simplest uses of erosion is for eliminating irrelevant detail (in terms of size) from a binary image.

Squares of size 1,3,5,7,9,15 pels

Erode with 13x13 square



original image → erosion → dilation

Note that objects are represented by white pixels, rather than by black pixels.

Dilation and Erosion are Duals

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

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

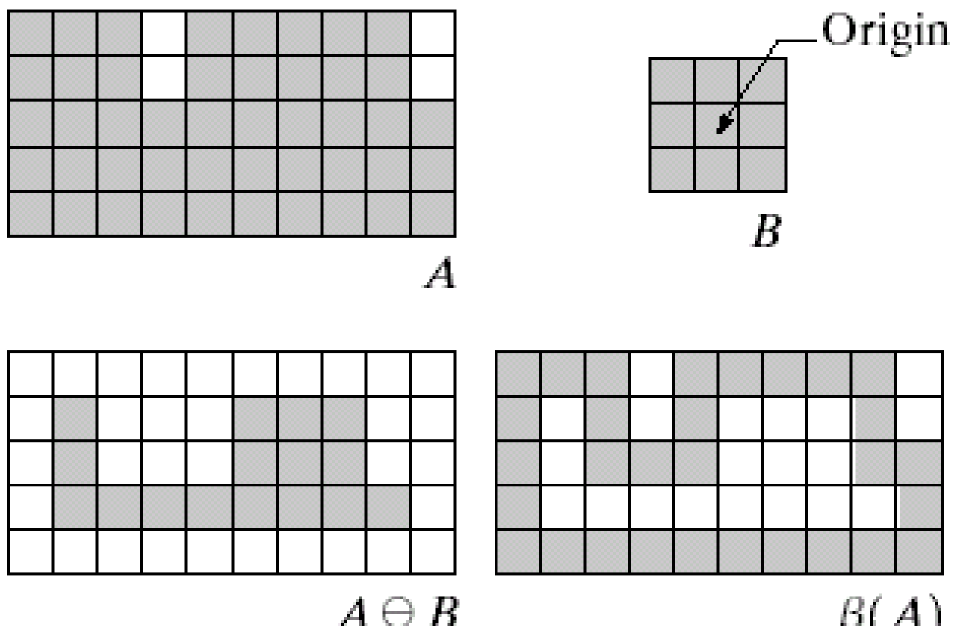
$$= \{ z | (B)_z \cap A^c \neq \varnothing \}$$

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

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

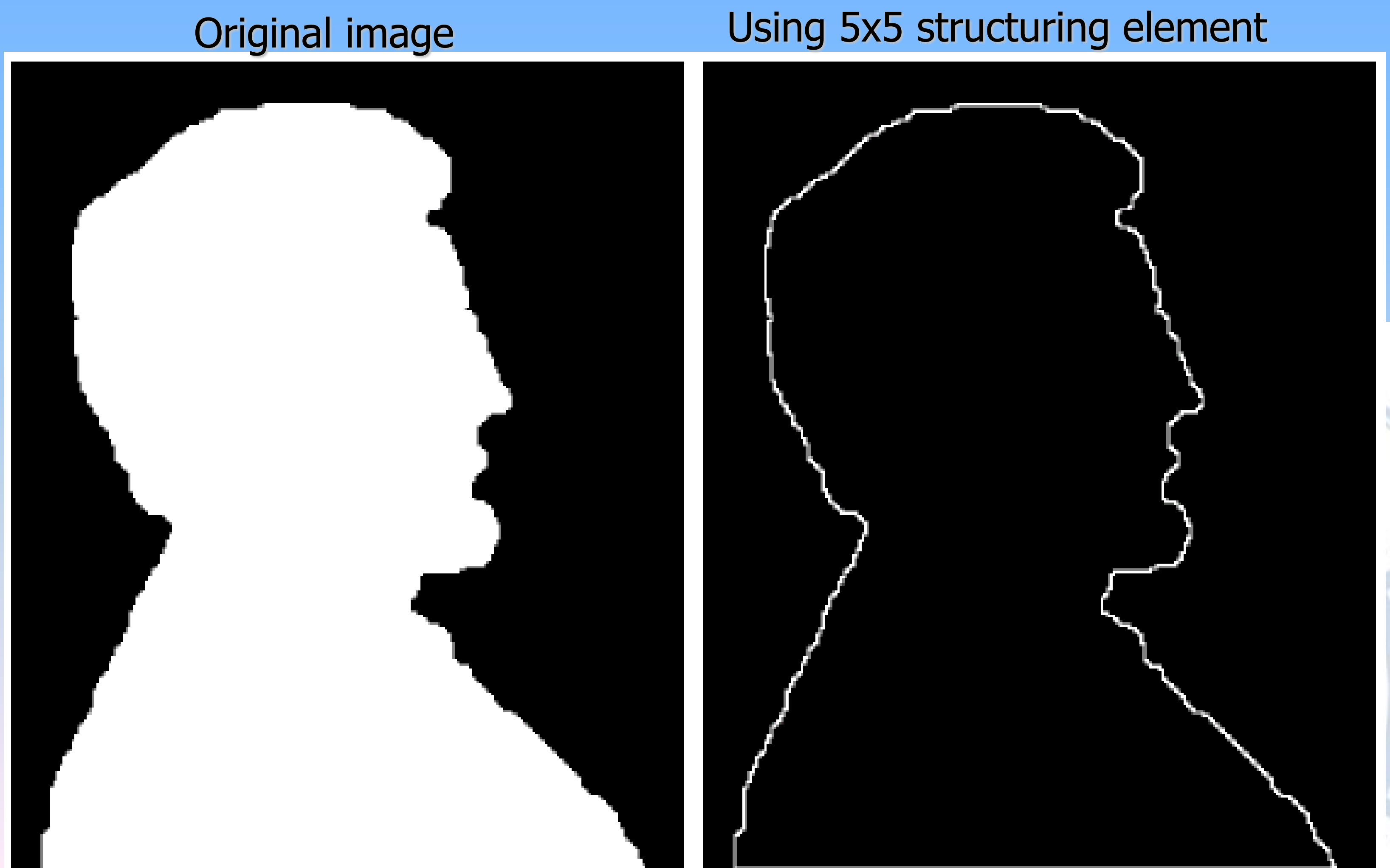
Application: Boundary Extraction

- Extract boundary of a set A:
  - First erode A (make A smaller)
  - A – erode(A)



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

Application: Boundary Extraction



Outline

- Preliminaries
- Dilation and erosion
- Opening and closing
- Hit-or-miss transformation
- Some basic morphological algorithms
- Morphological operations on gray images

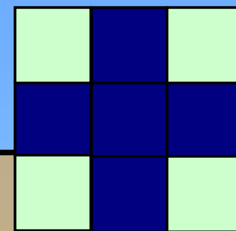
Opening

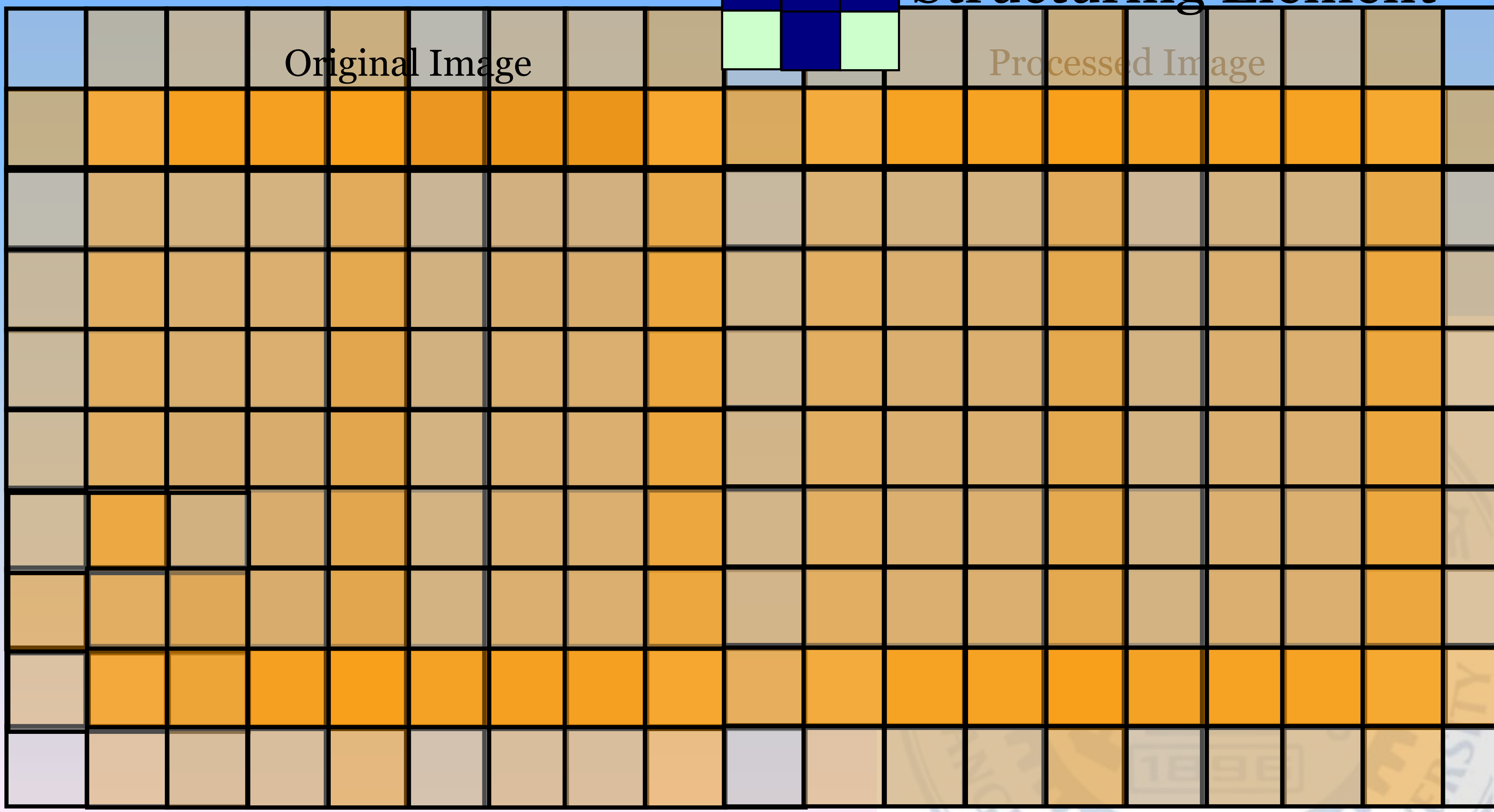
- Dilation: expands image w.r.t structuring elements
- Erosion: shrink image
- erosion+dilation = original image ?
- Opening= erosion + dilation

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

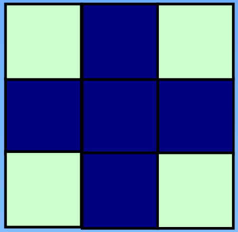
- Opening: to break narrow isthmuses and to eliminate thin protrusions.

Opening



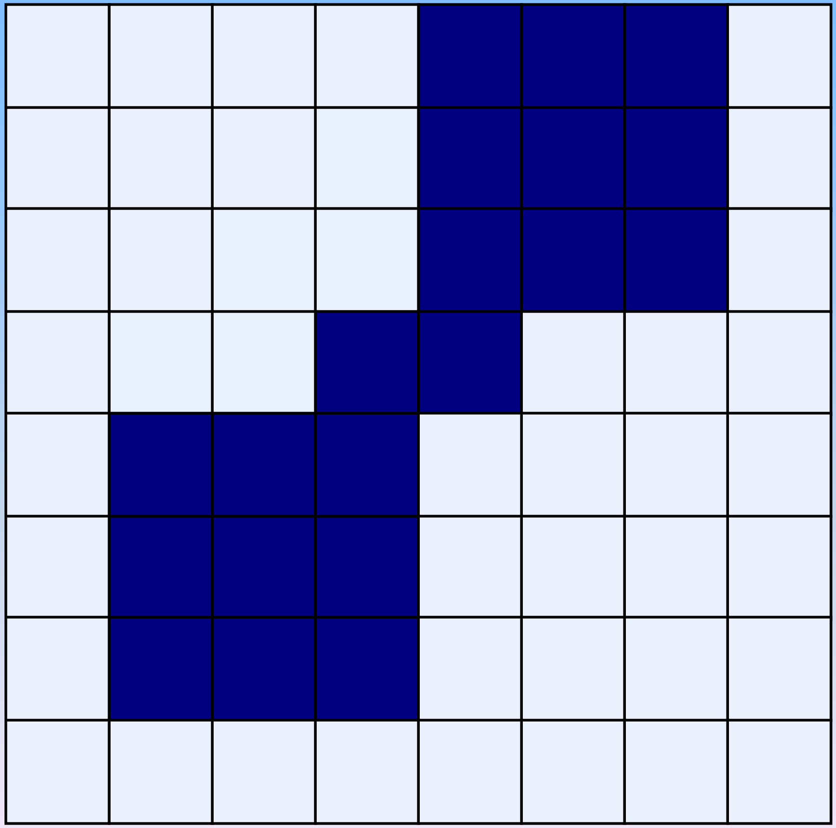


Opening

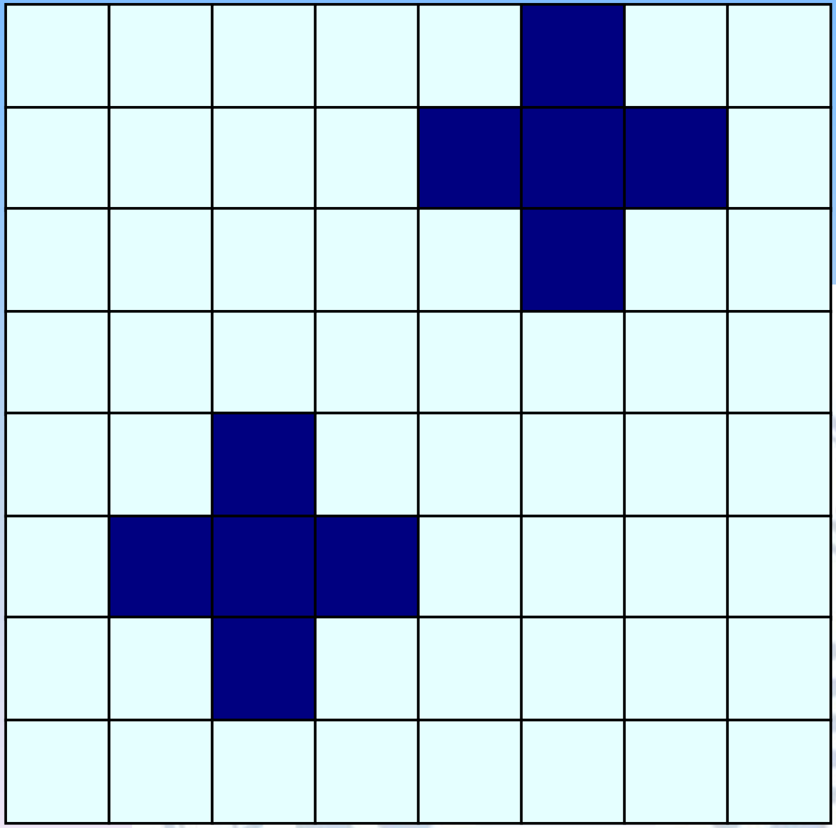


Structuring Element

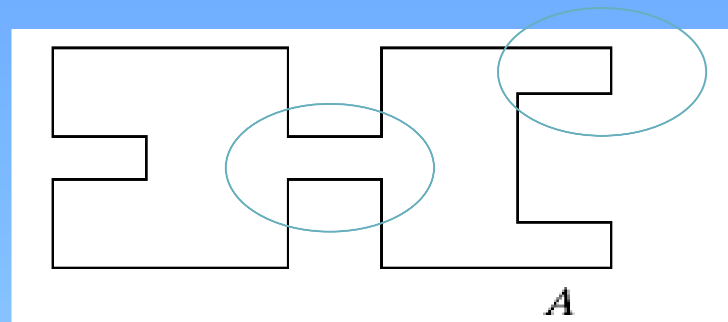
Original Image

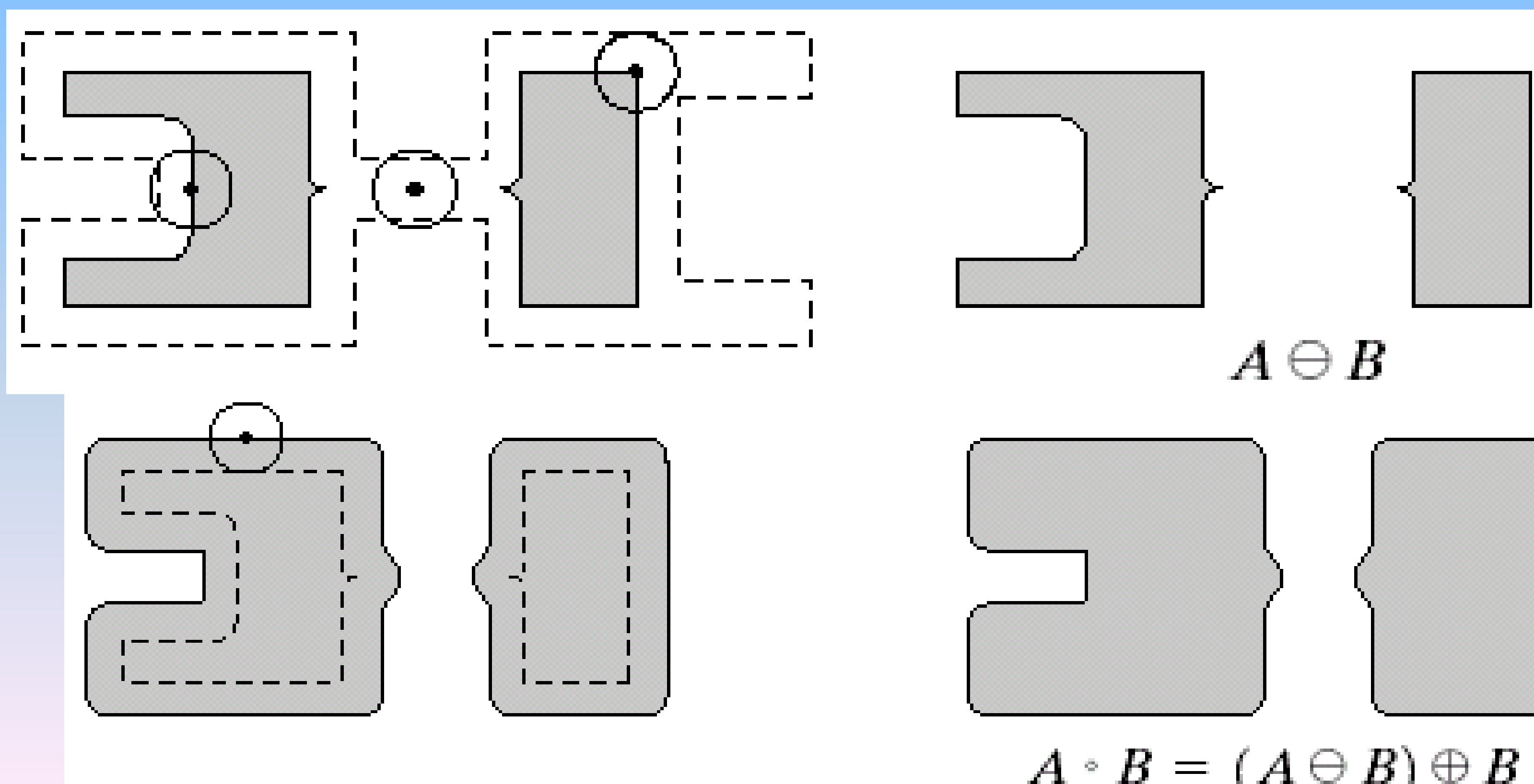


Processed Image

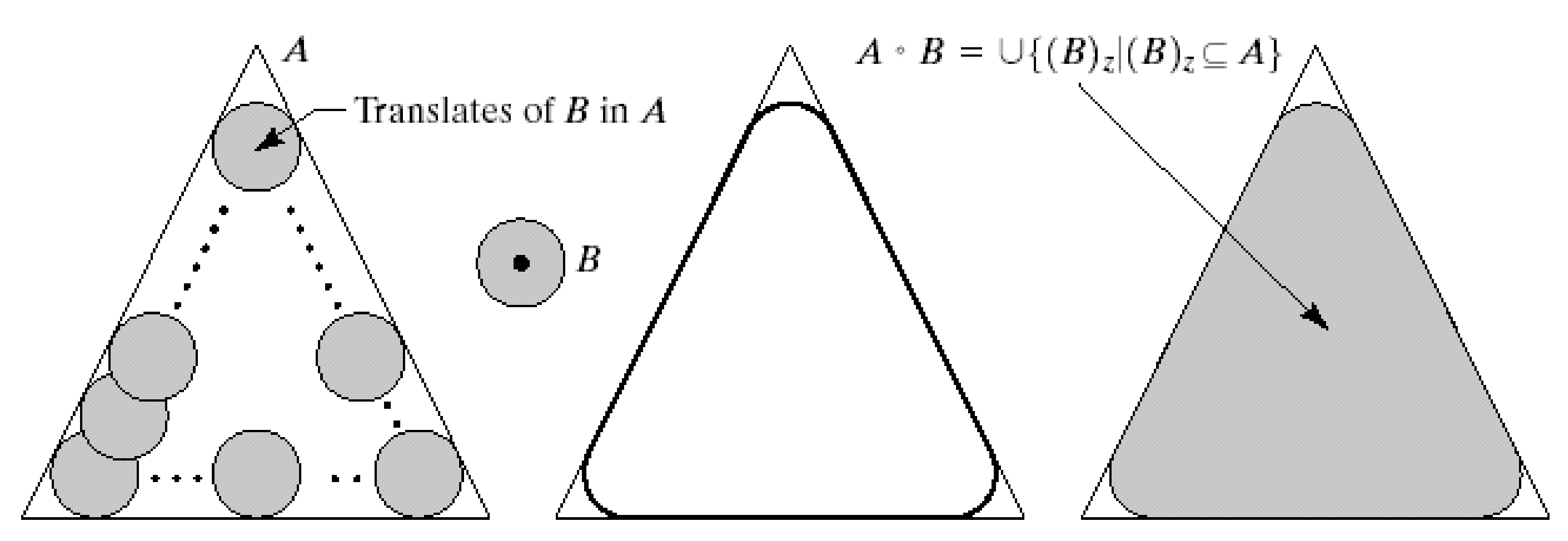


Opening





Opening



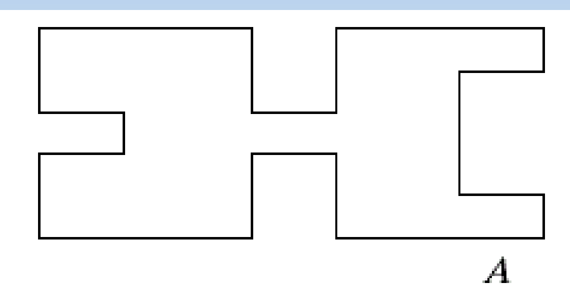
Find contour

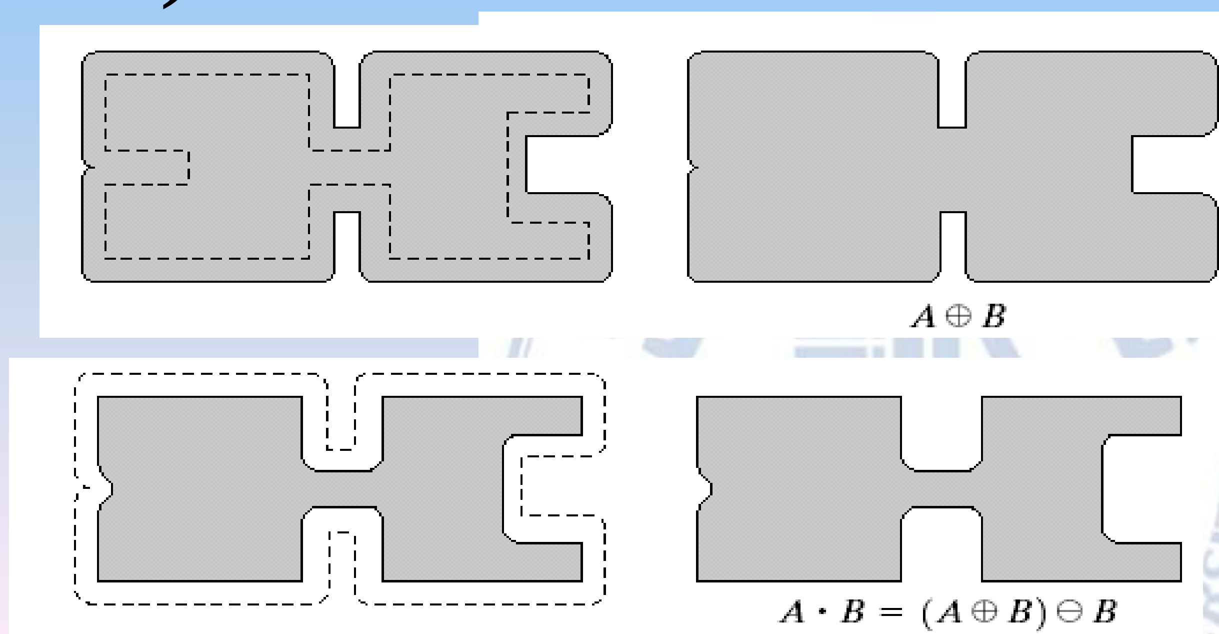
Fill in contour

Closing

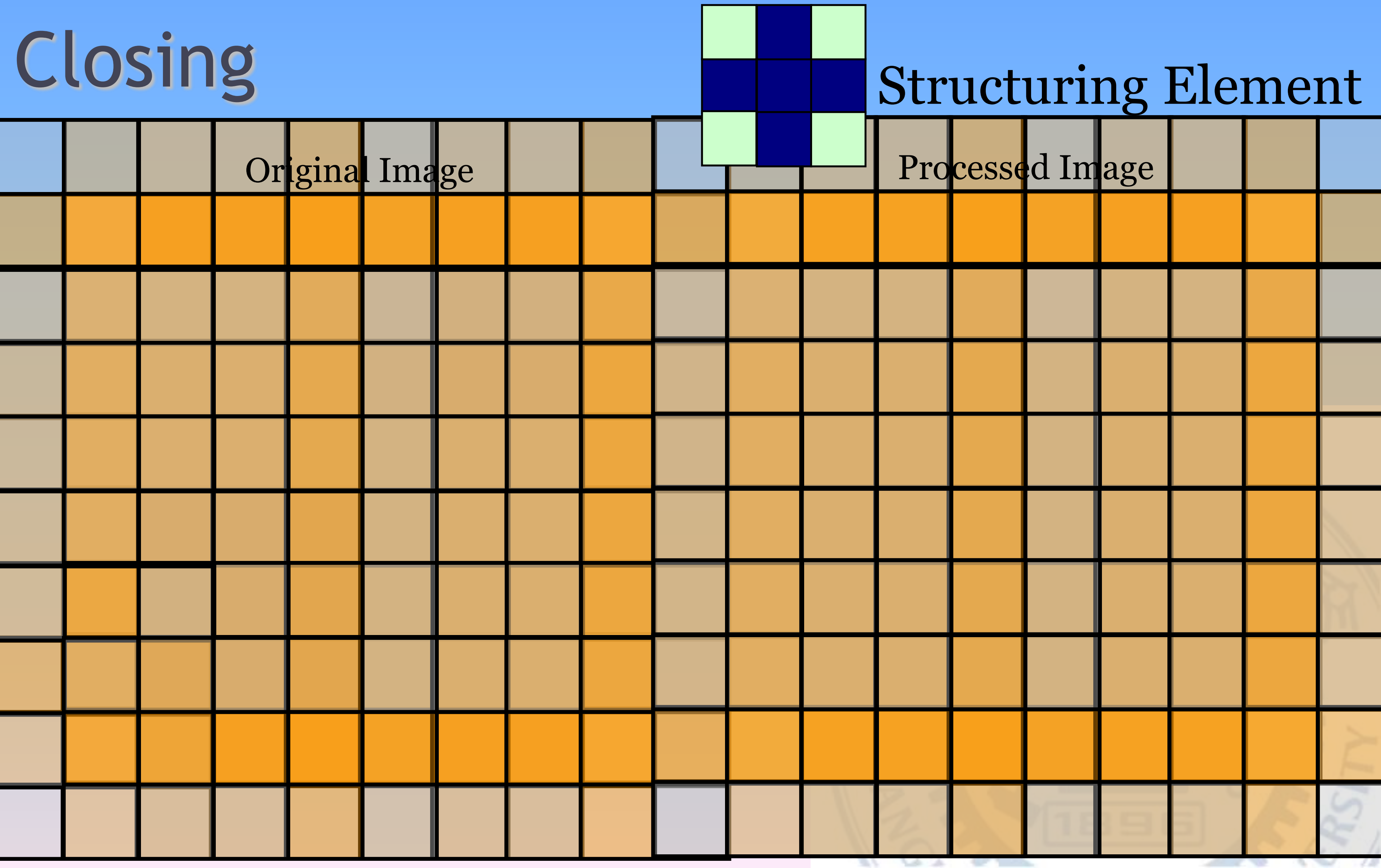
- Dilation+erosion = erosion + dilation ?
- Closing = dilation + erosion

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

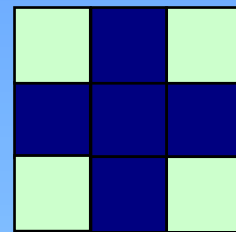




Closing

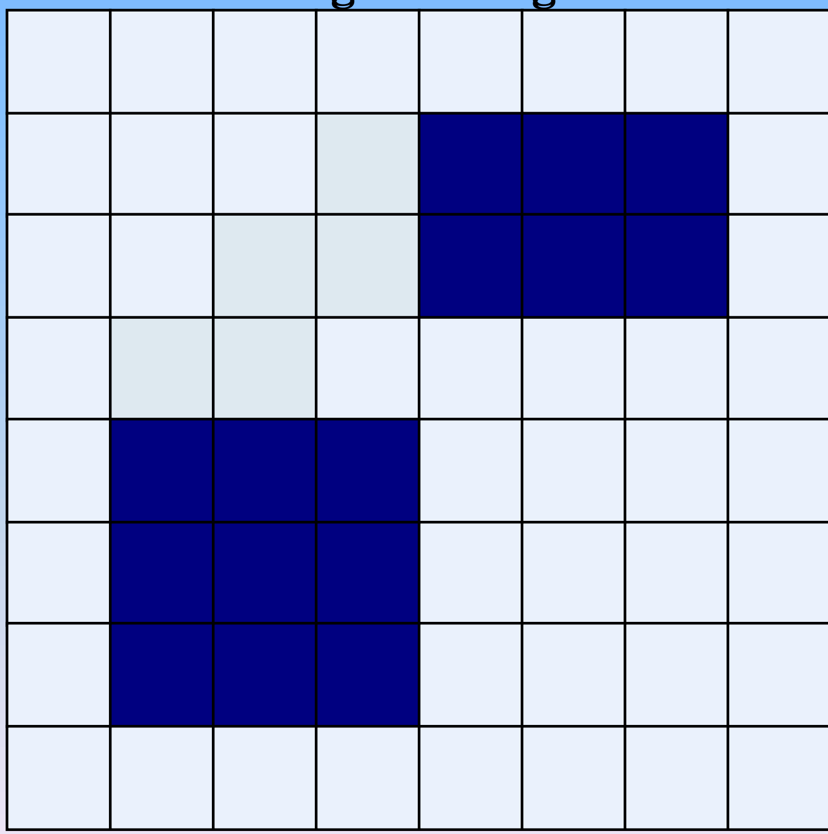


Closing

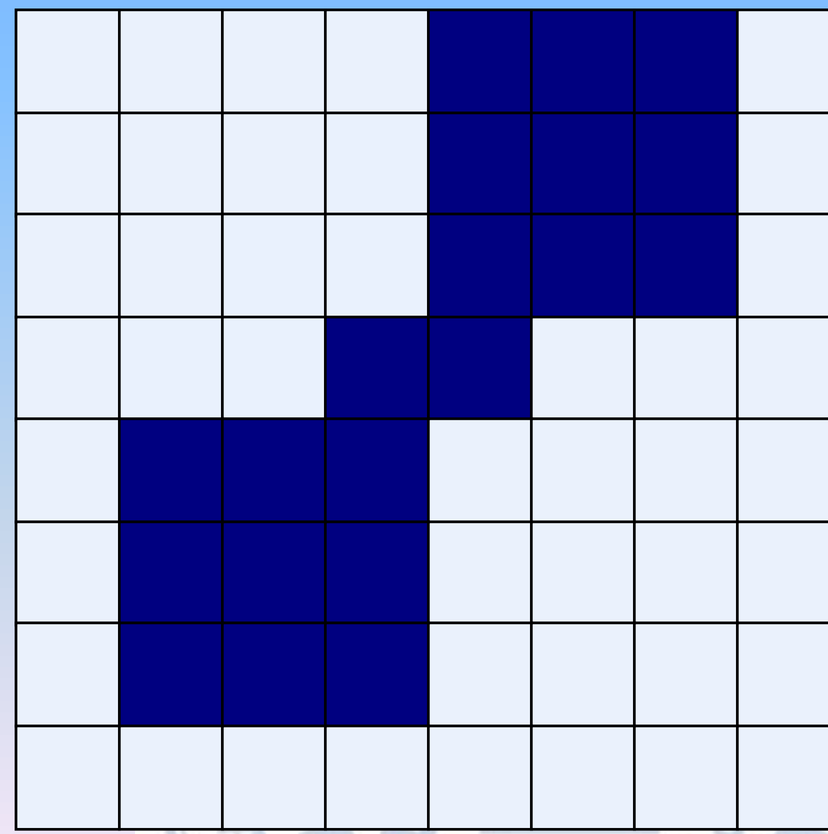


Structuring Element

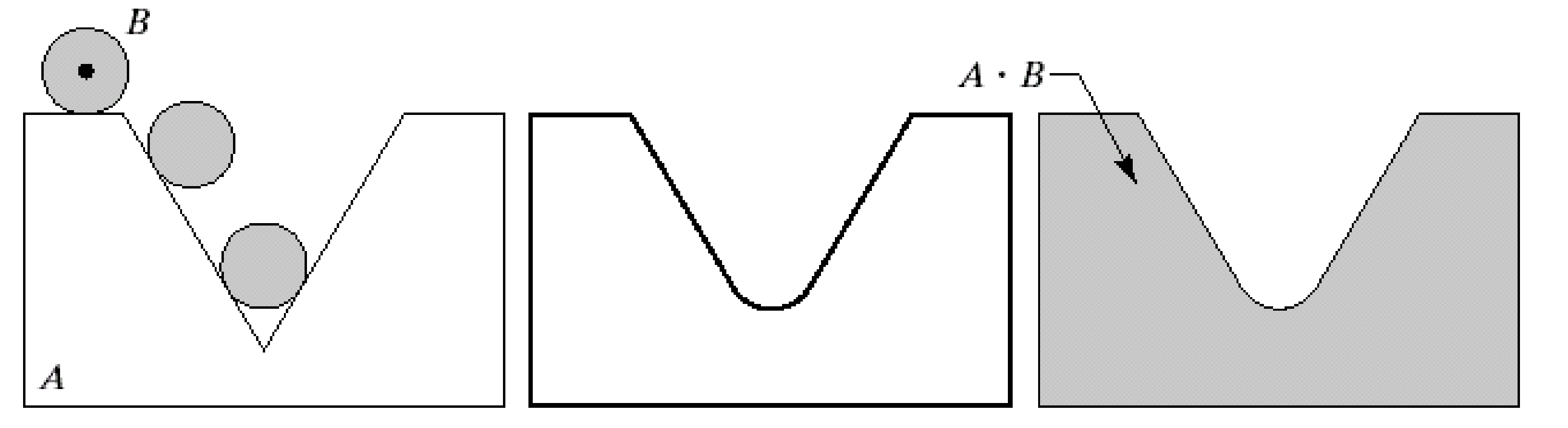
Original Image



Processed Image



Closing



Find contour Fill in contour

Closing: to fuse narrow breaks and long thin gulfs, to eliminate small holes, and to fill gaps in the contour.

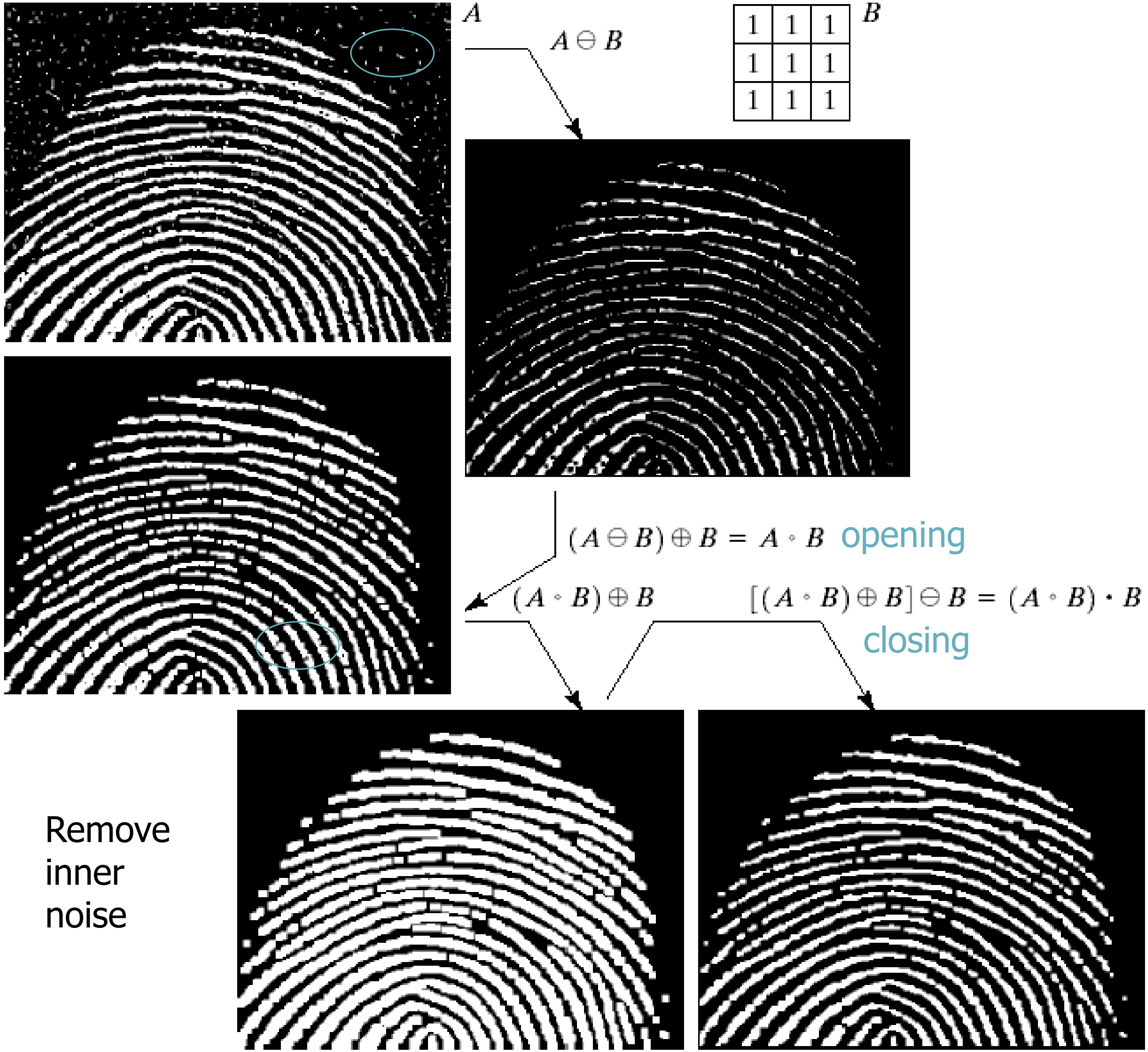
Properties of Opening and Closing

- Opening
  (i) $A \circ B$ is a subset (subimage) of A
  (ii) If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$
  (iii) $(A \circ B) \circ B = A \circ B$
- Closing
  (i) A is a subset (subimage) of $A \bullet B$
  (ii) If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$
  (iii) $(A \bullet B) \bullet B = A \bullet B$

Noisy image

Remove outer noise

Remove inner noise



Outline

- Preliminaries
- Dilation and erosion
- Opening and closing
- Hit-or-miss transformation
- Some basic morphological algorithms
- Morphological operations on gray images

Hit-or-miss Transformation

- The morphological hit-or-miss transform is a basic tool for *shape detection* or *pattern matching*.
- Let B denote the set composed of X and its background.
  - $B = (B_1, B_2)$, where $B_1=X$, $B_2=W-X$.

  The match of B in A, denoted by $A \circledast B$, is

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

Hit-or-miss Transformation

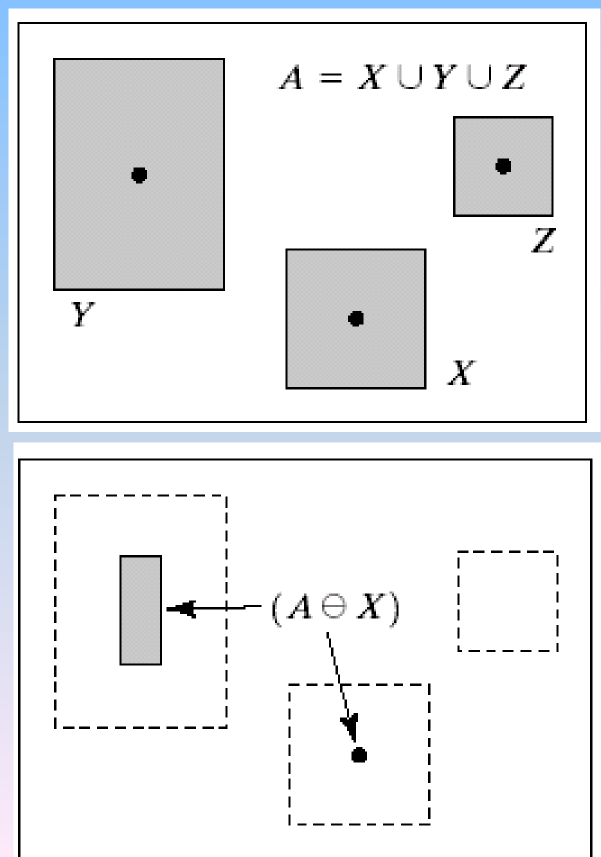
- Other interpretation:

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

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

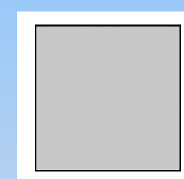
Hit-or-miss Transformation

- Find the location of certain shape



erosion

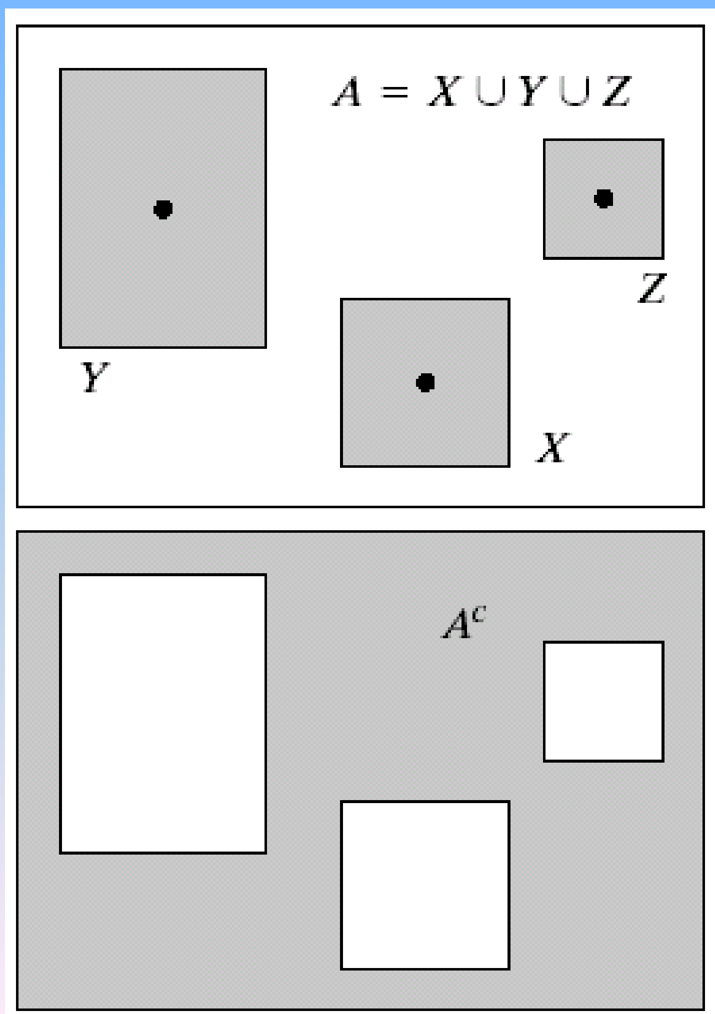
X

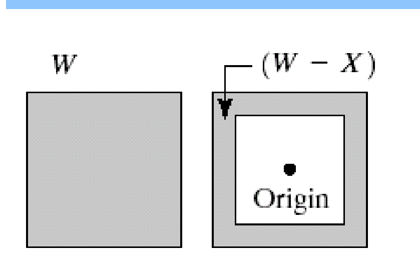


Find the set of pixels that contain shape *X*

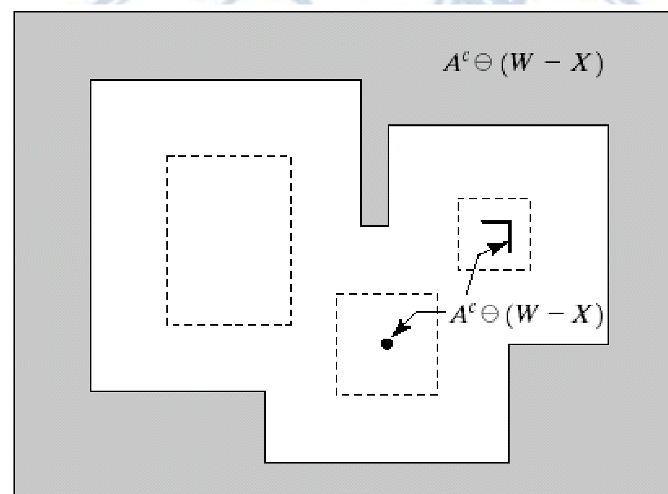
Hit-or-miss Transformation

Detect object via background



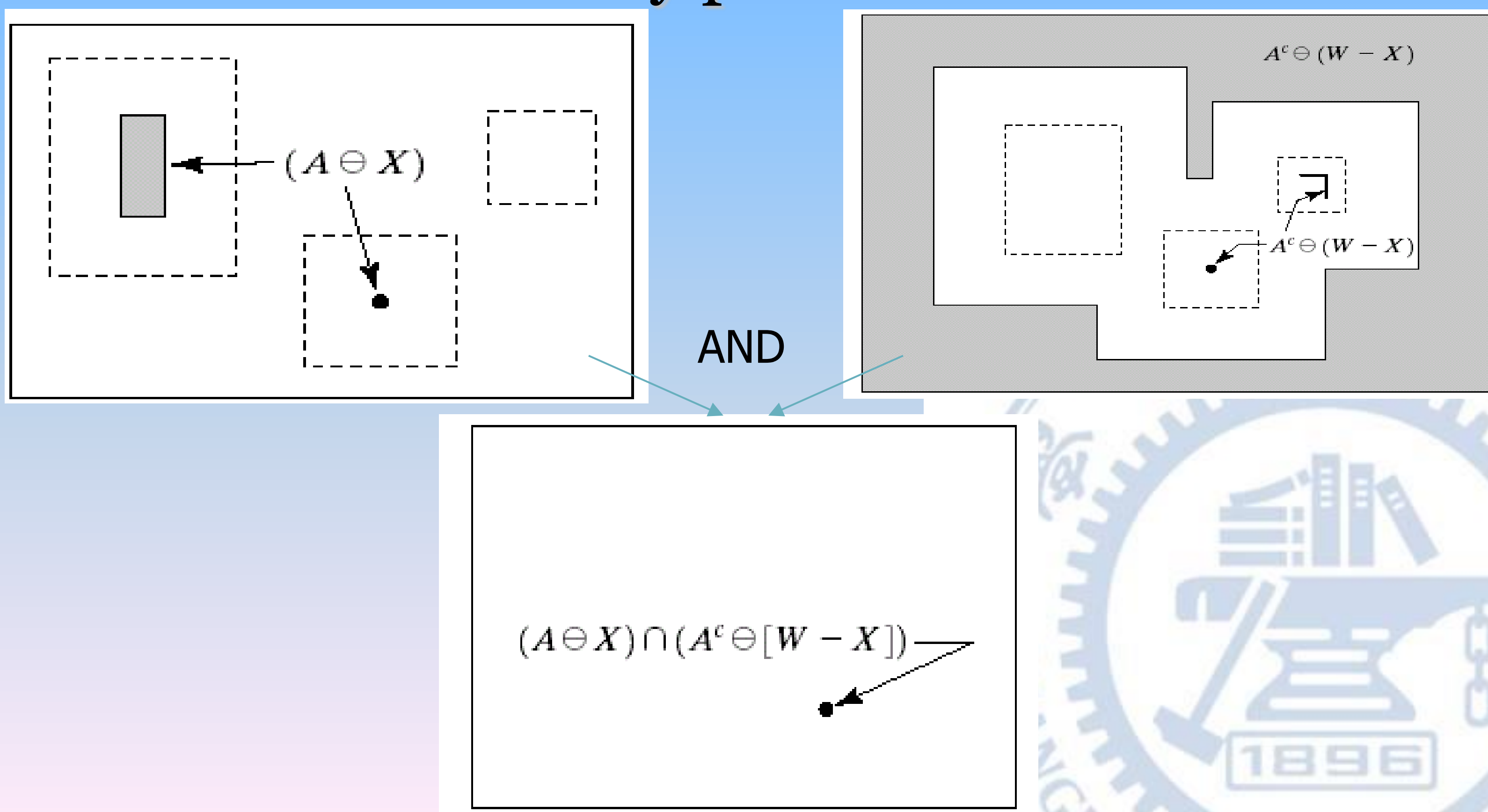


Erosion with (W-X)



Hit-or-miss Transformation

- Eliminate un-necessary parts



Outline

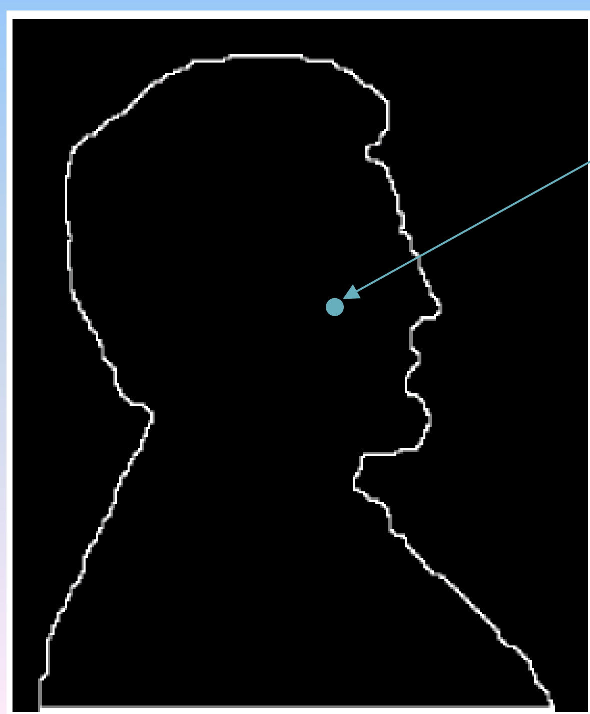
- Preliminaries
- Dilation and erosion
- Opening and closing
- Hit-or-miss transformation
- Some basic morphological algorithms
- Morphological operations on gray images

Basic Morphological Algorithms

- Extract image components that are useful in the representation and description of shape
- Boundary extraction
- Region filling
- Extract of connected components
- Convex hull
- Thinning
- Thickening
- Skeleton
- Pruning

Region Filling

- How?
- Idea: place a point inside the region, then dilate that point iteratively



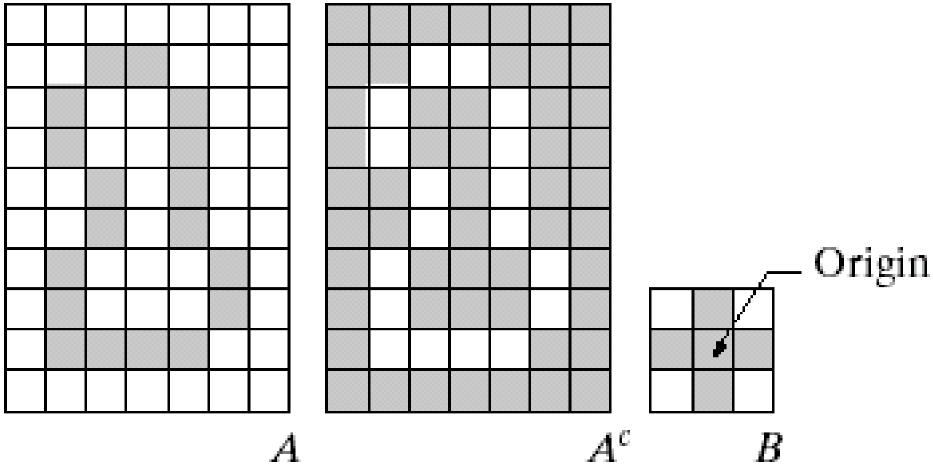
$$X_0 = p$$

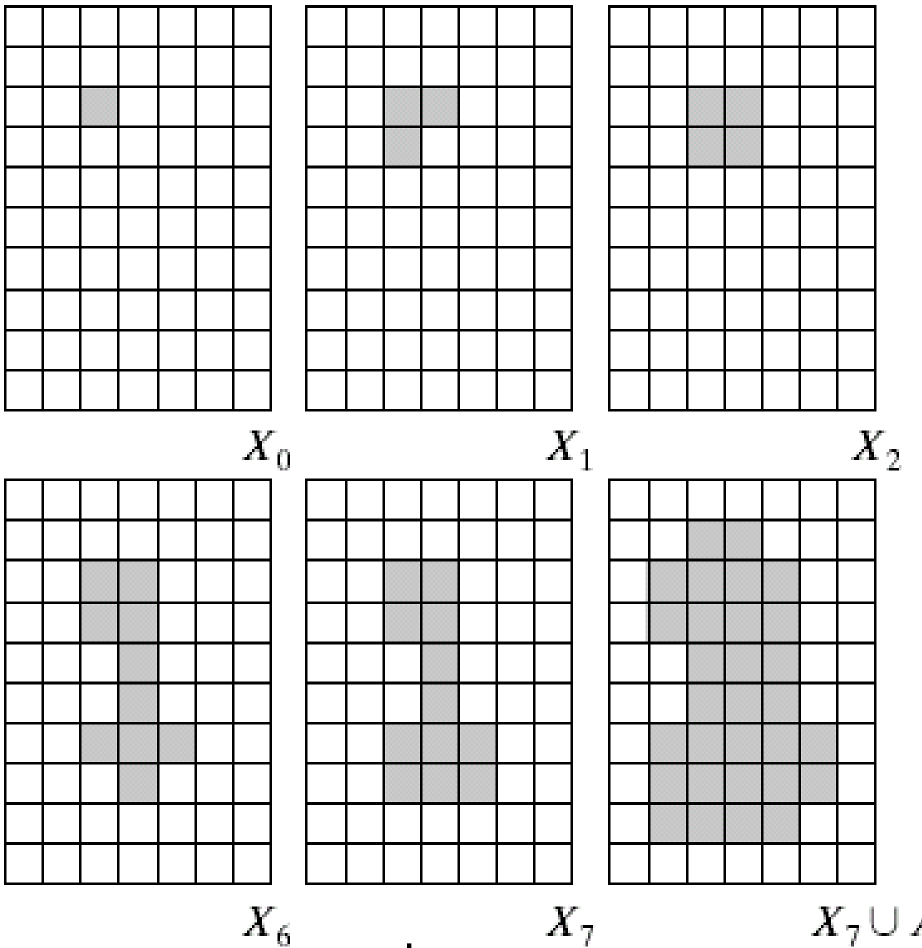
$$X_k = (X_{k-1} \oplus B) \cap A^c, k = 1,2,3,\ldots$$

Until $X_k = X_{k-1}$

Bound the growth

Region Filling

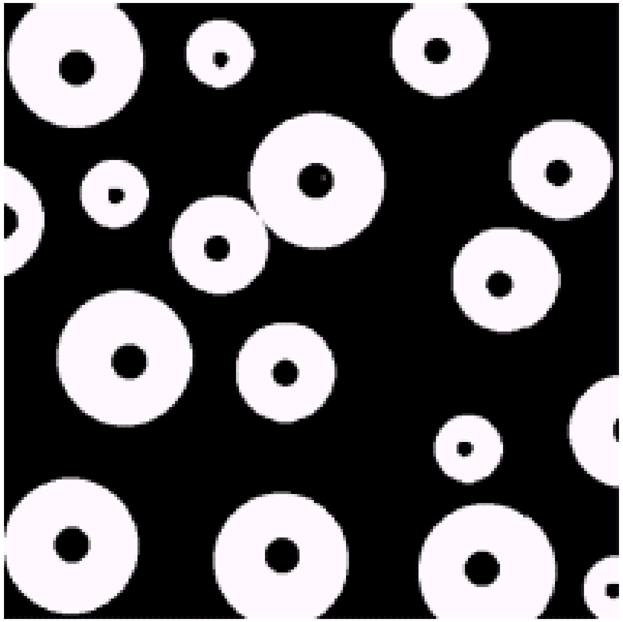




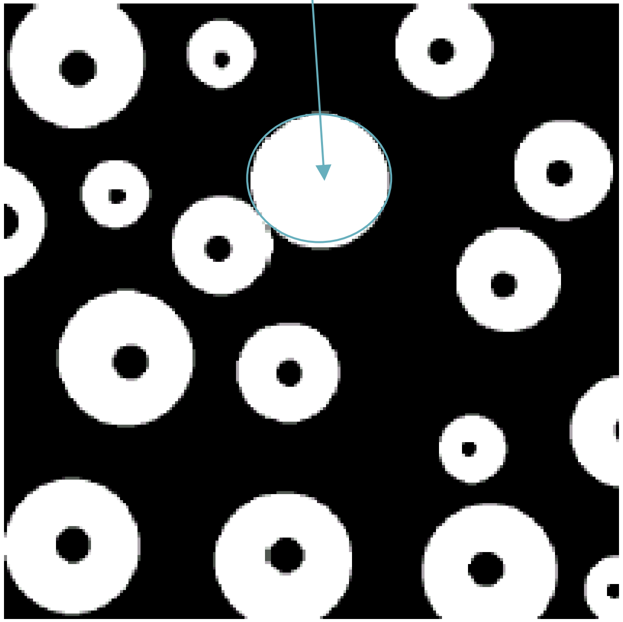
stop

Application: region filling

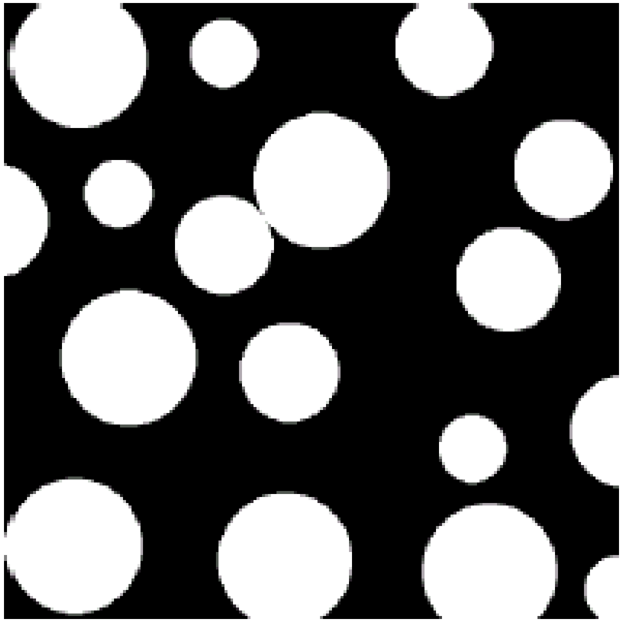
Original image



The first filled region



Fill all regions



Extraction of connected components

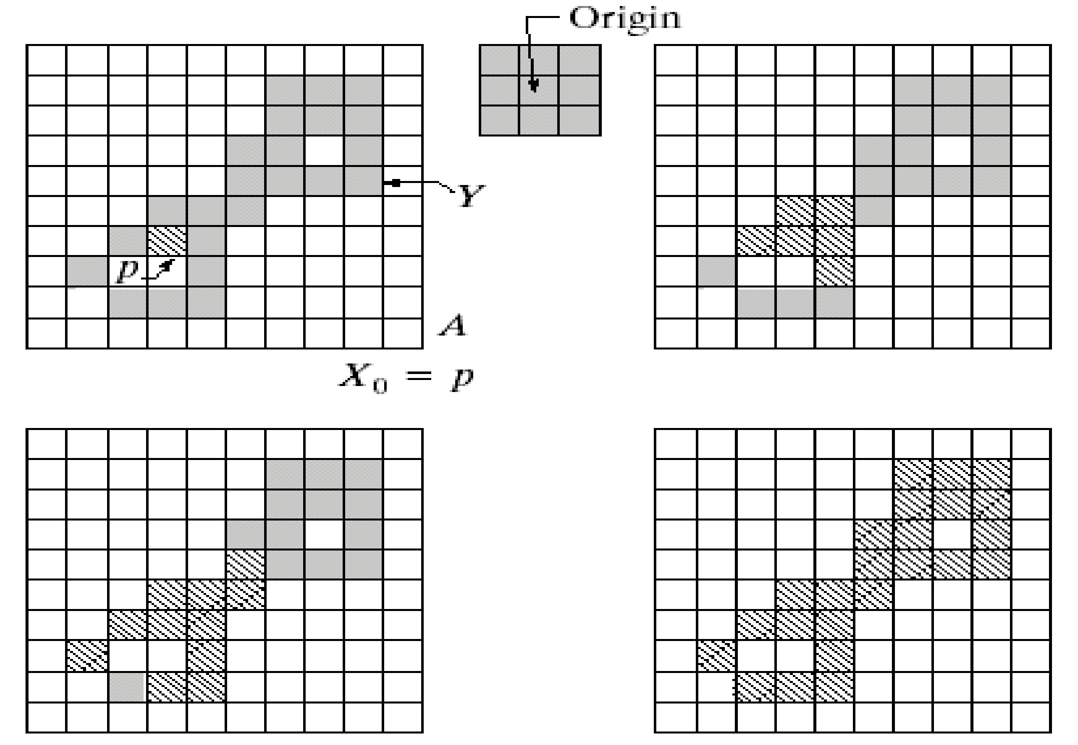
- Idea: start from a point in the connected component, and dilate it iteratively

$$X_0 = p$$

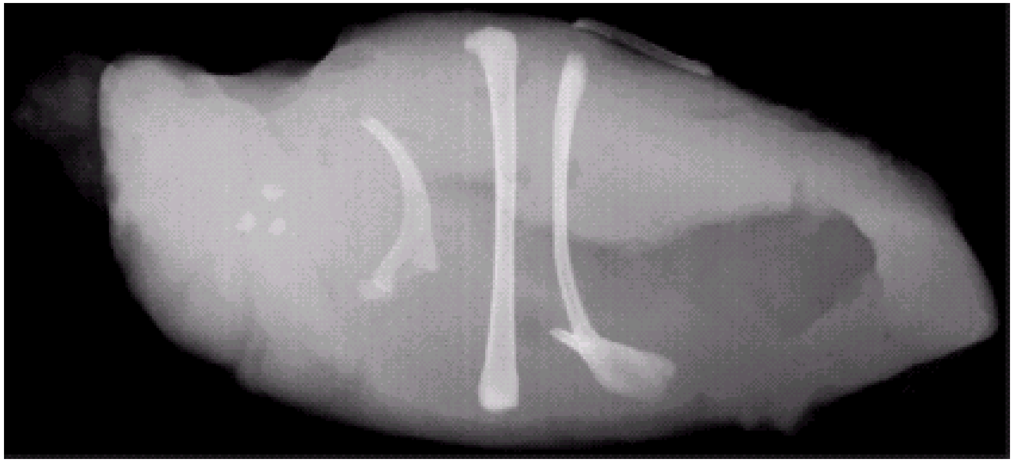
$$X_k = (X_{k-1} \oplus B) \cap A, \; k = 1,2,3,\ldots$$

Until $X_k = X_{k-1}$

Extraction of connected components



original



thresholding

erosion

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

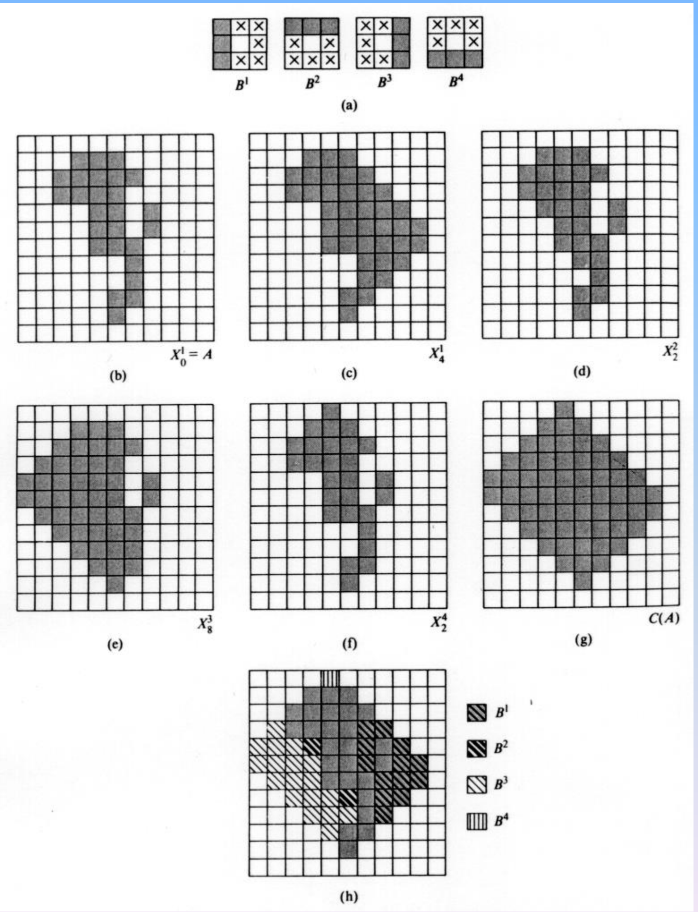
Convex Hull

- A set *A* is said to be *convex*.
  - If the straight line joining any two points in *A* lies entirely within *A*.
- The *convex hull H* of a set *S* is the smallest convex set containing *S*.
  - The set *H-S* is called the convex difference, which is useful for object description.
- The procedure is to implement the equation:

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1,2,3,4 \text{ and } k = 1,2,3\ldots$$

  - With $X^i_0 = A$. Let $D^i = X^i_{conv}$, where "conv" indicates that $X^i_k = X^i_{k-1}$. The convex hull of *A* is $C(A) = \bigcup_{i=1}^{4} D^i$

Convex Hull



Convex hull:
the procedure for obtaining convex hull consists of iteratively applying the hit-or-miss transform to A with B^1 (B^i, i = 1, 2, 3, 4, represent four structuring elements.); when no further changes occur, we perform the union with A and call the result D^1. The procedure is repeated with B^2 until no further changes occur, and so on. The union of the four resulting D's constitutesthe convex hull of A.

Thinning

- The thinning of a set A by a structuring element B, denoted $A \otimes B$, is defined by

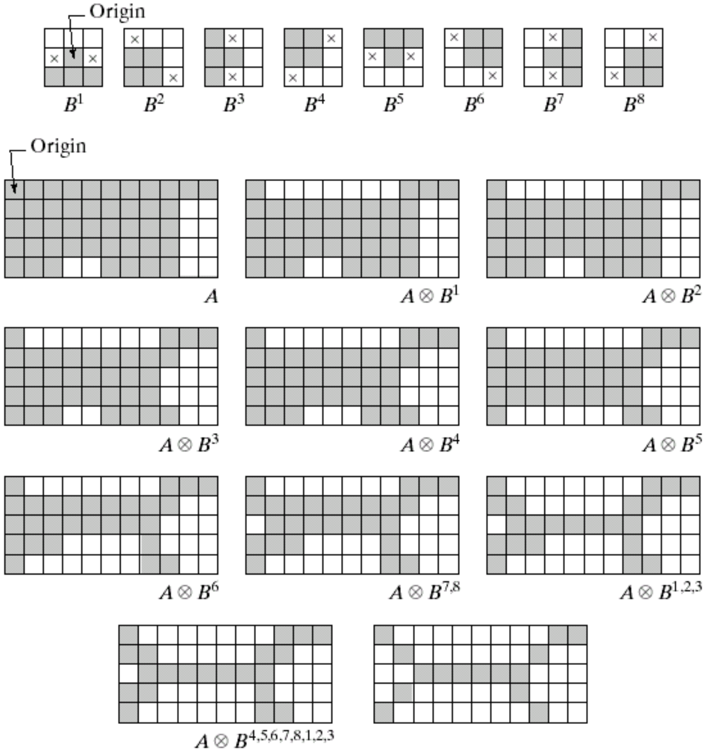
$$A \otimes B = A - (A \circledast B)$$

- Each B is usually a sequence of structuring elements:

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

  - B^1, B^2,...are different rotated versions of B.
- The result of thinning A by one pass is the union of the results obtained by thinning by B^i by one pass.

Thinning Procedure



a / b c d / e f g / h i j / k l

FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A. (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to m-connectivity.

Thinning algorithm is based on the hit-or-miss transform.

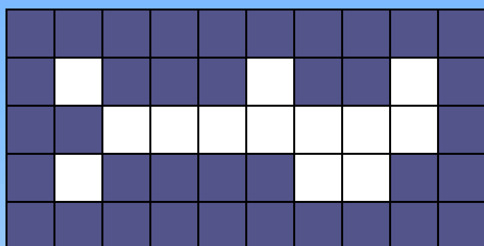
Thickening

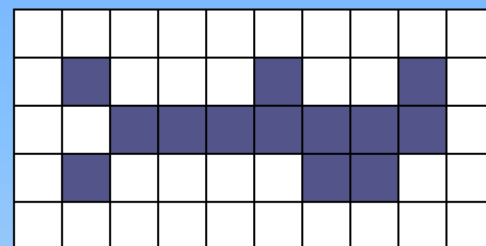
- The thickening of a set A by a structuring element B, is defined by

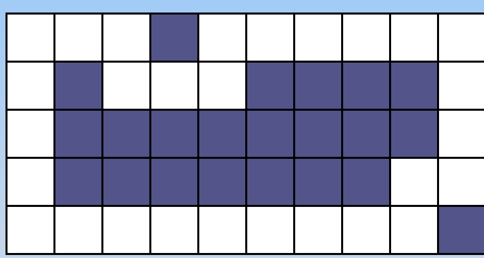
$$A \odot B = A \cup (A \circledast B)$$

- A more efficient scheme is to obtain the complement of A, say A^c, and then to compute C^c, where C is the thinned result of A^c and C^c is its complement.

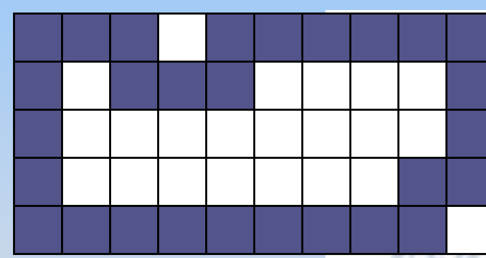
$A \odot B = A \cup (A \circledast B)$ where B is a structuring element suitable for thickening.

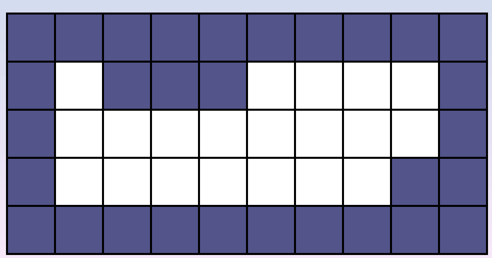

A


A'


(c)

Result of thinning A'


thickened set obtained by complementing (c)


Final result, showing removal of disconnected points.

Skeletons

- The dot line : the skeleton of *A*, *S*(*A*).

$$S(A)=\bigcup_{k=0}^{K} S_k(A) \text{ with } S_k(A)=(A\ominus kB)\circ B$$

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

a b
c d

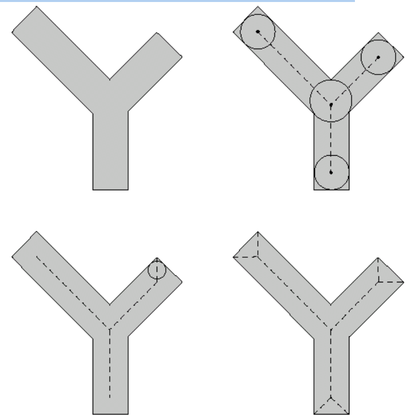


FIGURE 9.23 (a) Set *A*. (b) Various positions of maximum disks with centers on the skeleton of *A*. (c) Another maximum disk on a different segment of the skeleton of *A*. (d) Complete skeleton.

The Procedure of Skeletonization

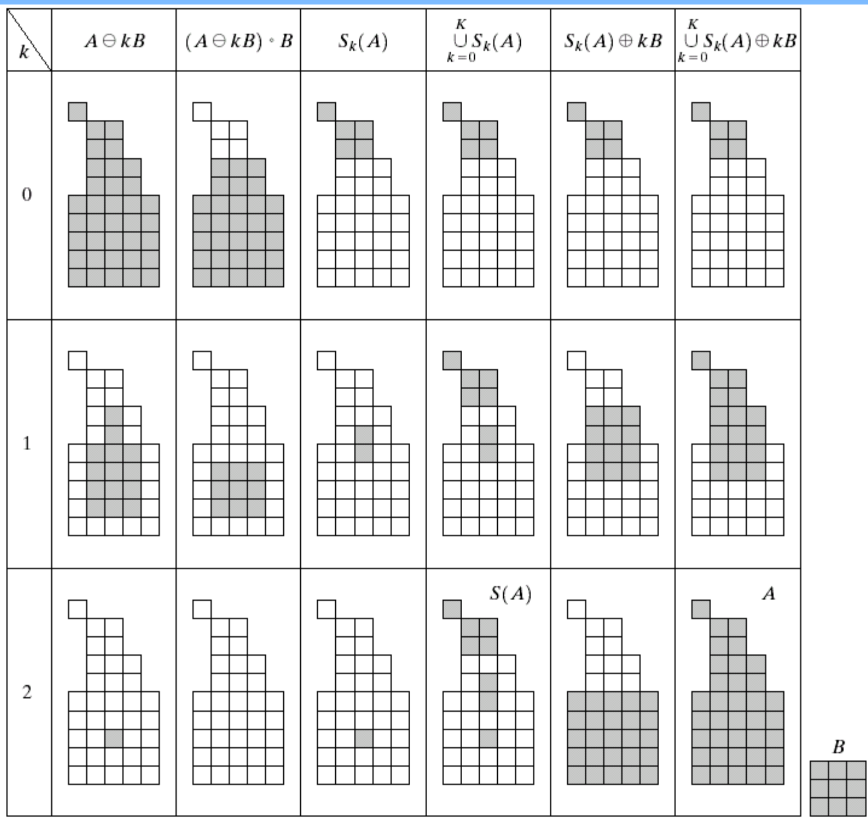
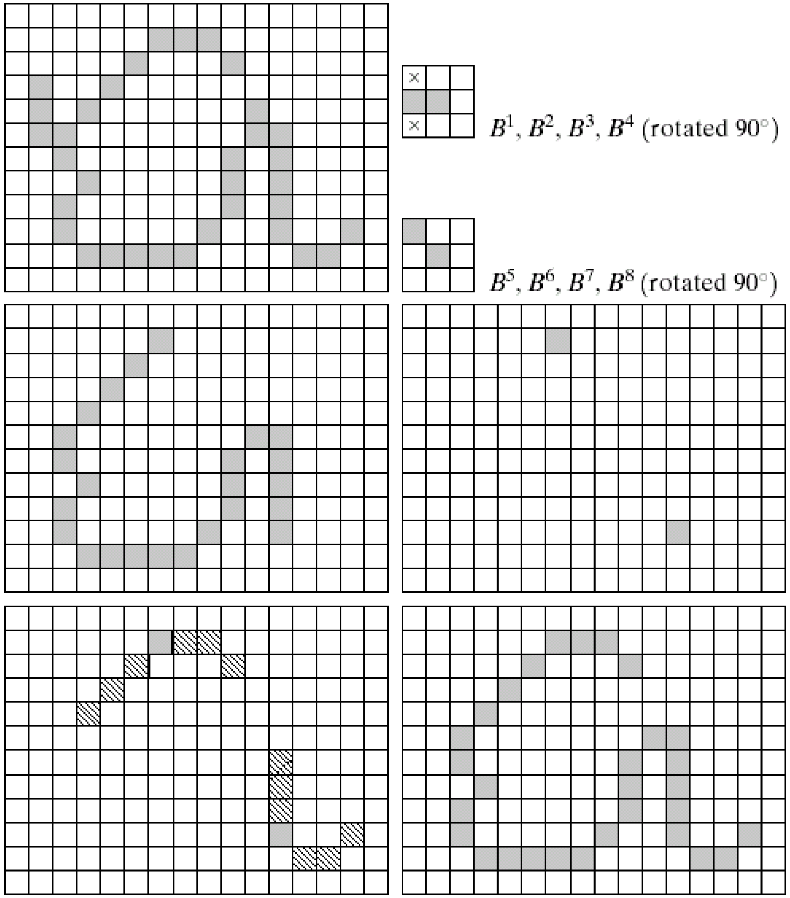


FIGURE 9.24 Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.

Pruning



a b
c
d e
f g

FIGURE 9.25 (a) Original image. (b) and (c) Structuring elements used for deleting end points. (d) Result of three cycles of thinning. (e) End points of (d). (f) Dilation of end points conditioned on (a). (g) Pruned image.

The Procedure of Pruning

- Thinning an input set *A* to eliminate the short line segment by

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

- To restore the character to its original form:
  - Find the set containing all the end points by

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

  - Dilate the end points and find the intersection with *A*:

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

  - The union of X_3 and X_1 yields the desired result:

$$X_4 = X_1 \cup X_3$$

Outline

- Preliminaries
- Dilation and erosion
- Opening and closing
- Hit-or-miss transformation
- Some basic morphological algorithms
- Morphological operations on gray images

Dilation $A \oplus B$ of gray images

Let A be the gray image dilated by the structuring element B, G be the resultant gray image and $b(x, y) \in B$, $g(s,t) \in G$, then

$$g(s,t) = (f \oplus b)(s,t) = \max\{f(s-x,t-y)+b(x,y) \mid (s-x),(t-y) \in D_A; (x,y) \in D_B\}$$

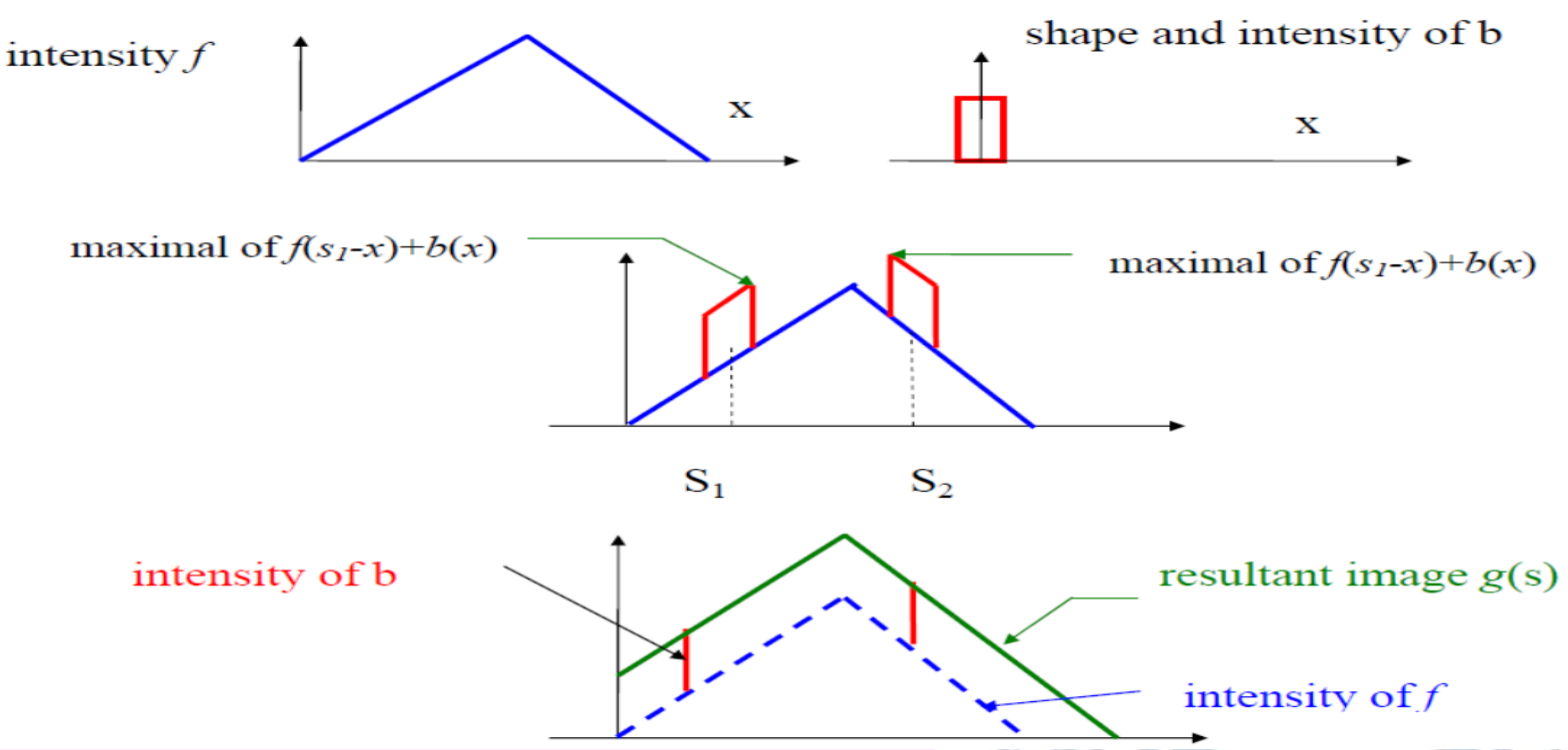
One may follow three steps to obtain the intensity of g(s, t) at a pixel coordinate (s, t)

(i) For each given (s, t), translate A around B (i.e., change x and y) such that $(s-x),(t-y) \in D_A$ and $(x,y) \in D_B$

(ii) at each position (x, y), obtain A(s – x, t – y) + b(x, y) with the given (s, t)

(iii) For the given (s, t), compare all intensity value A(s – x, t – y) + b(x, y) under the condition that $(s-x),(t-y) \in D_A$ and $(x,y) \in D_B$, and choose the maximum value of A(s – x, t – y) + b(x, y) as the value of g(s, t).

- Please notice that Resultant intensity is maximal of A(s – x,t – y) + b(x, y) , hence dilation generally increase the intensity of a gray image.
- In the limiting case when the gray image is binary, the above definition reduces exactly to that defined for binary images.
- However, in the case of gray images, all pixels are changed (in terms of intensity) while binary operations changes only those pixels on the boundary.
- The following example is a 1-D case



Erosion A ⊖ B of gray images

Let A be the gray image eroded by the structuring element B, G be the resultant gray image and b(x, y) ∈ B, g(s,t) ∈ G , then

$$g(s,t) = (f \Theta b)(s,t) = \min\{A(s+x,t+y) - b(x,y) \mid (s+x),(t+y) \in D_A;(x,y) \in D_B\}$$

One may follow three steps to obtain the intensity of g(s, t) at a pixel coordinate (s, t)

(i) For each given (s, t), translate A around B (i.e., change x and y) such that $(s-x),(t-y) \in D_A$ and $(x,y) \in D_B$

(ii) at each position (x, y), obtain A(s – x, t – y) + b(x, y) with the given (s, t)

(iii) For the given (s, t), compare all intensity value A(s – x, t – y) + b(x, y) under the condition that $(s-x),(t-y) \in D_A$ and $(x,y) \in D_B$, and choose the minimum value of A(s – x, t – y) + b(x, y) as the value of g(s, t).

- Please notice that Resultant intensity is minimal of A(s – x, t – y) + b(x, y) , hence dilation generally decrease the intensity of a gray image.
- In the limiting case when the gray image is binary, the above definition reduces exactly to that defined for binary images.
- However, in the case of gray images, all pixels are changed (in terms of intensity) while binary operations changes only those pixels on the boundary.
- The following example is a 1-D case

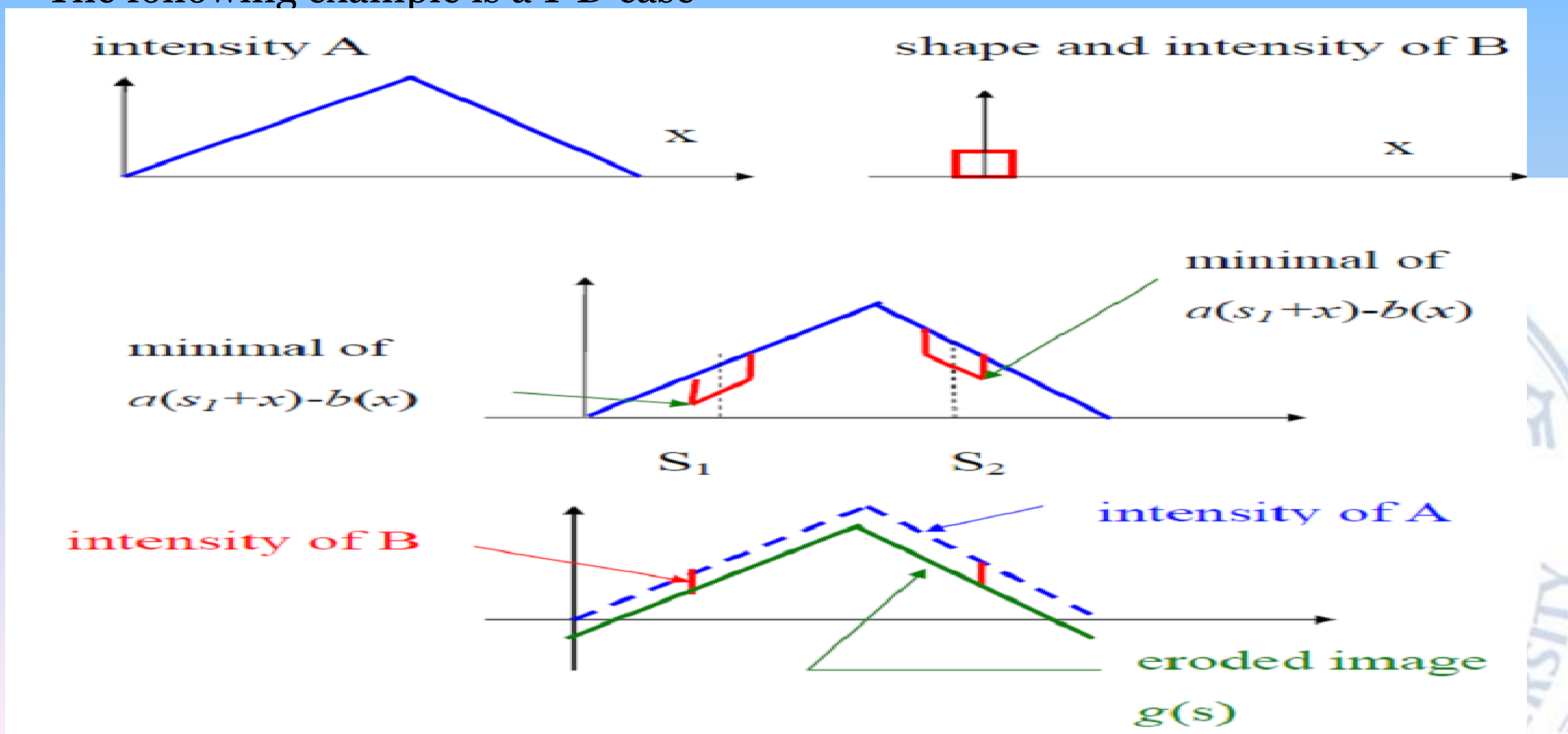
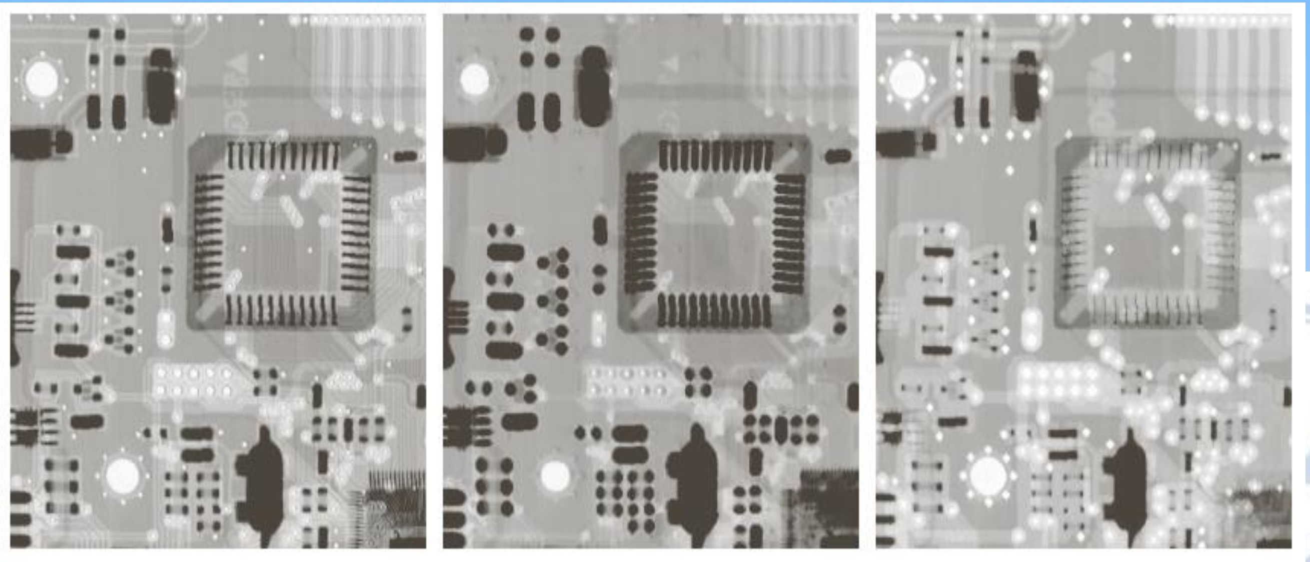


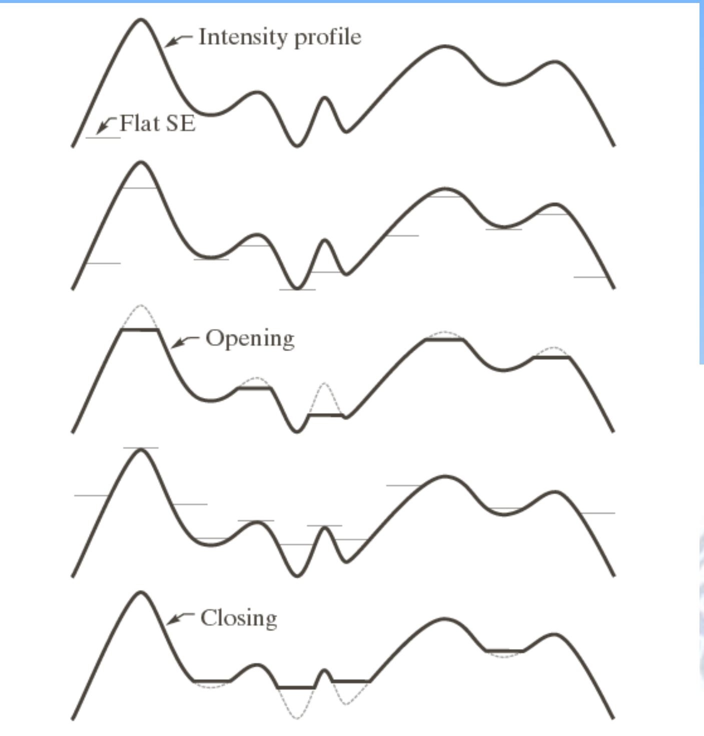
Illustration of gray-scale erosion and dilation



a b c

FIGURE 9.35 (a) A gray-scale X-ray image of size 448 × 425 pixels. (b) Erosion using a flat disk SE with a radius of two pixels. (c) Dilation using the same SE. (Original image courtesy of Lixi, Inc.)

Opening and closing in one dimension

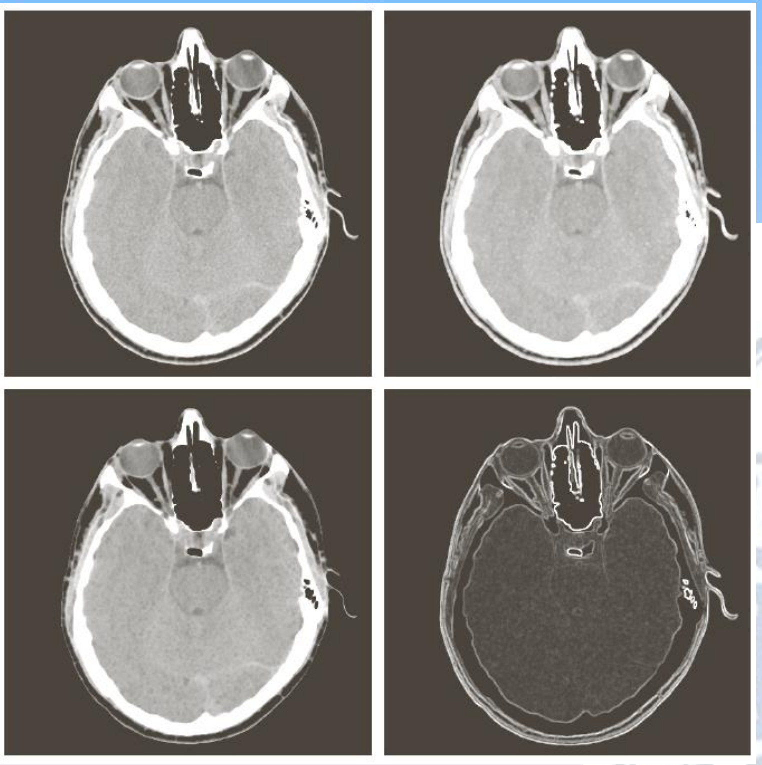


a
b
c
d
e

FIGURE 9.36 Opening and closing in one dimension. (a) Original 1-D signal. (b) Flat structuring element pushed up underneath the signal. (c) Opening. (d) Flat structuring element pushed down along the top of the signal. (e) Closing.

Morphological gradient

Morphological gradient = $(A \oplus B) - (A \ominus B)$ can be used for edge detection.



a b
c d

FIGURE 9.39
(a) 512 × 512 image of a head CT scan.
(b) Dilation.
(c) Erosion.
(d) Morphological gradient, computed as the difference between (b) and (c).
(Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Top-hat and bottom-hat transformations

- Combing image subtraction with openings and closings results in so-called top-hat and bottom-hat transformations.

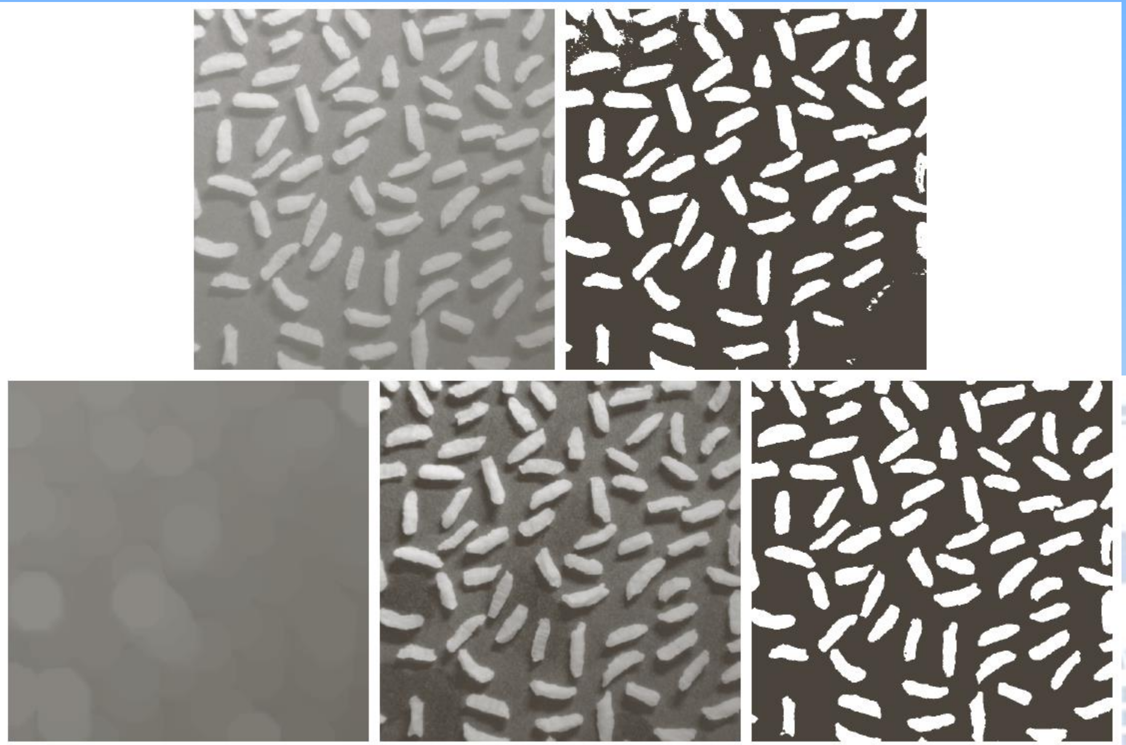
Which is defined as f minus its opening:

$$T_{hat}(f) = f - (f \circ b)$$

And closing of f minus f:

$$T_{hat}(f) = (f \text{¥} b) - f$$

Top-hat transformation for shading correction

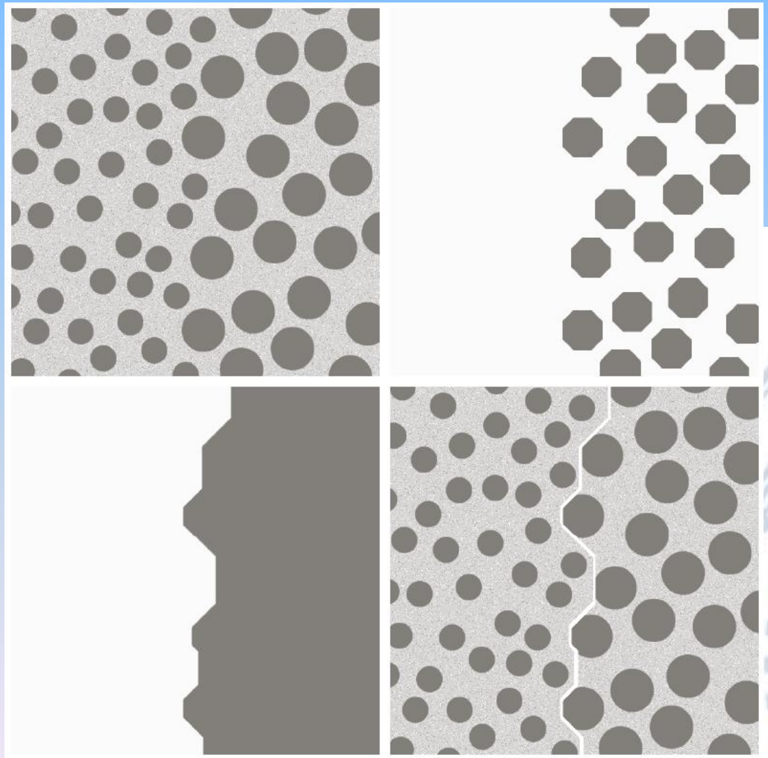


a b
c d e

FIGURE 9.40 Using the top-hat transformation for *shading correction*. (a) Original image of size 600 × 600 pixels. (b) Thresholded image. (c) Image opened using a disk SE of radius 40. (d) Top-hat transformation (the image minus its opening). (e) Thresholded top-hat image.

Textural segmentation

To find the boundary of large blobs and small blobs



a b
c d

FIGURE 9.43 Textural segmentation. (a) A 600 × 600 image consisting of two types of blobs. (b) Image with small blobs removed by closing (a). (c) Image with light patches between large blobs removed by opening (b). (d) Original image with boundary between the two regions in (c) superimposed. The boundary was obtained using a morphological gradient operation.

Requirements of Project Three now posted!

Thank You!