Morphological operators

Part I

Morphology

What is morphology?

- Commonly denotes a branch of biology that deals with the **form and structure** of animals and plants

What is mathematical morphology?

- A tool for extracting image components that are useful in the
  - representation and description of **region shapes** (boundaries, skeletons, convex hull, ...)
  - pre- or post-processing (morphological filtering, thinning, and pruning)
- Mathematical morphology deals with ***set theory***

When is it useful?

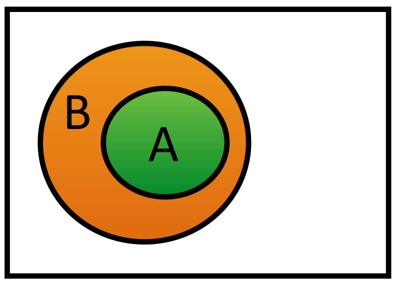
- pre-processing (noise filtering, shape simplification, boundaries smoothing, ...)
- enhancing object structure (skeletonization, convex hull, ...)
- segmentation (watershed, ...)
- quantitative description (area, perimeter, ...)

Recall mathematical tools

What is a set?

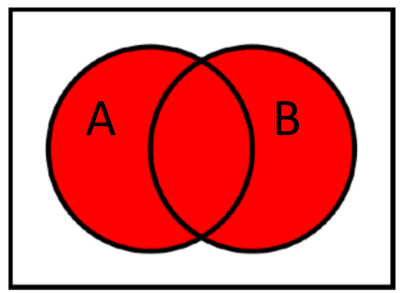
- **Collection** of distinct objects (e.g. for images collection of pixels)

Basic set operations



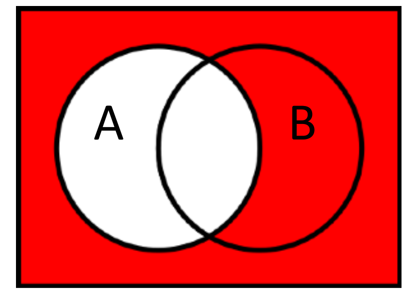
Subset

$A \subseteq B$



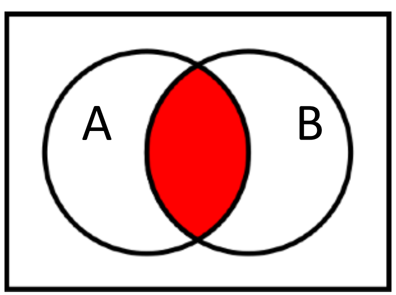
Union

$A \bigcup B$



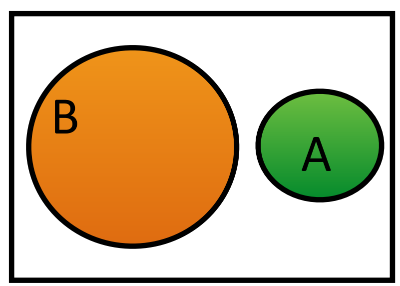
Complement

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



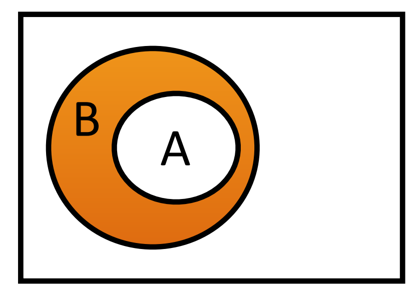
Intersection

$A \bigcap B$



Disjoint

$A \bigcap B = \emptyset$

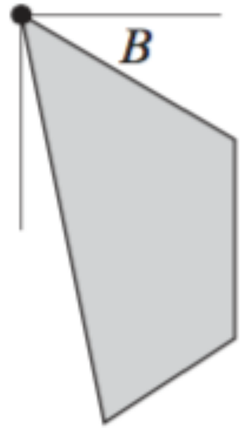


Difference

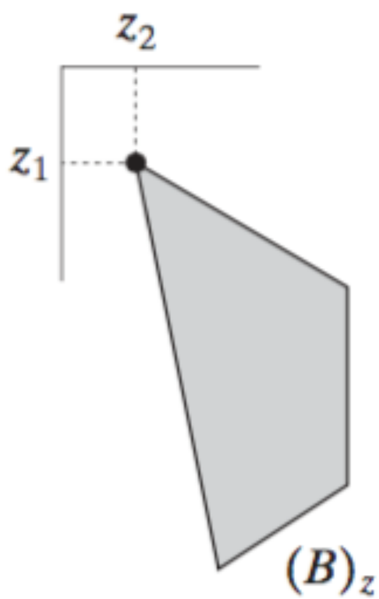
$B \setminus A = \{w \mid w \in B, w \notin A\}$

Recall mathematical tools

- Original set

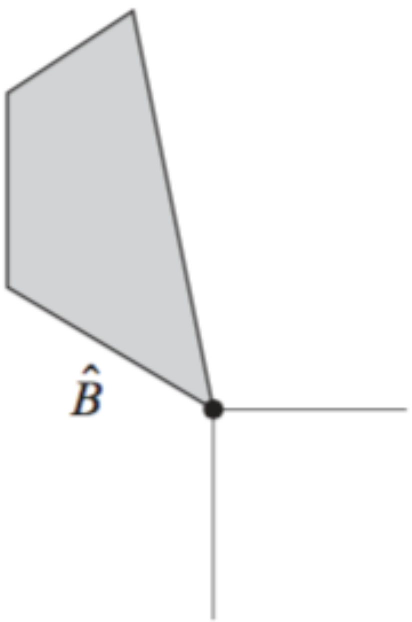


- Translation



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

- Reflection



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

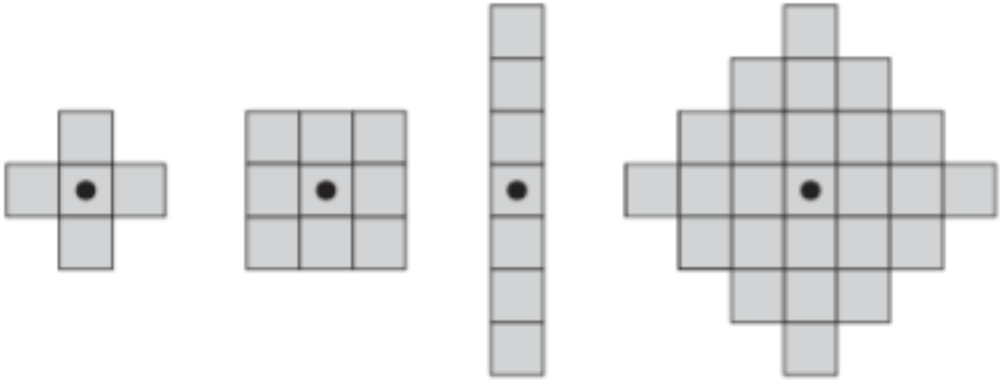
What are morphological operators?

- ***Local*** pixel ***transformation*** for processing region shapes
- Most often used on ***binary images***
- Logical transformation based on comparison of pixel neighbourhoods with a specific pattern (***structuring element***)
  - Erosion
  - Dilation
  - Opening
  - Closing
  - Hit-or-miss transformation

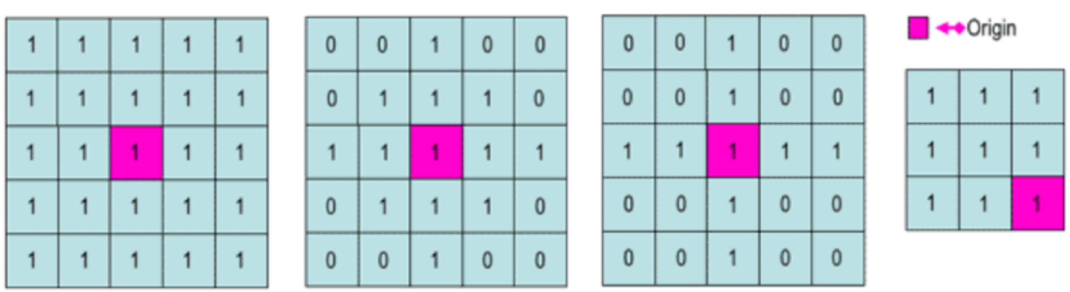
Binary morphology

Structuring element (SE)

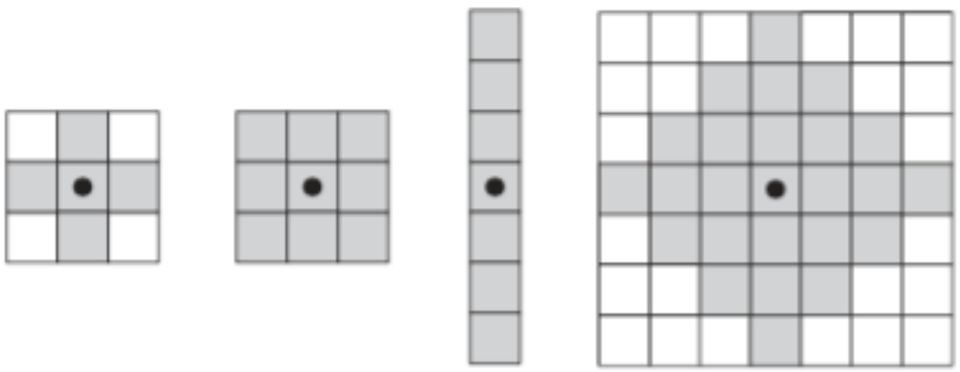
- Small set with a predefined shape to probe the image under study
  - **simple** shape
  - binary image
  - well defined ***origin*** (usually the center of gravity or symmetry)



- It is useful to adapt the shape to be rectangular



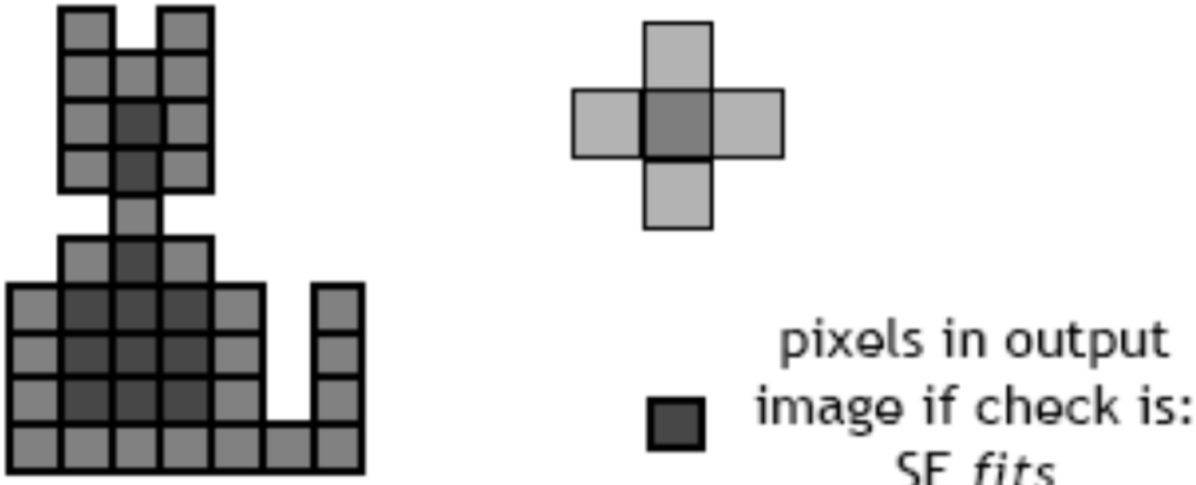
Square 5x5 element — Diamond-shaped 5x5 element — Cross-shaped 5x5 element — Square 3x3 element



How we use SE?

- We compare the structuring element to the neighbourhood of each pixel
  - In parallel for each pixel we
    - check if SE is “satisfied”
    - we mark the pixels that satisfy the SE and then we perform the desired morphological operation (erosion, dilation, …)

- Example



Fitting, hitting and missing

Given a structuring element *S* and a binary image *I*

- ***S* fits *I* at *x* if**

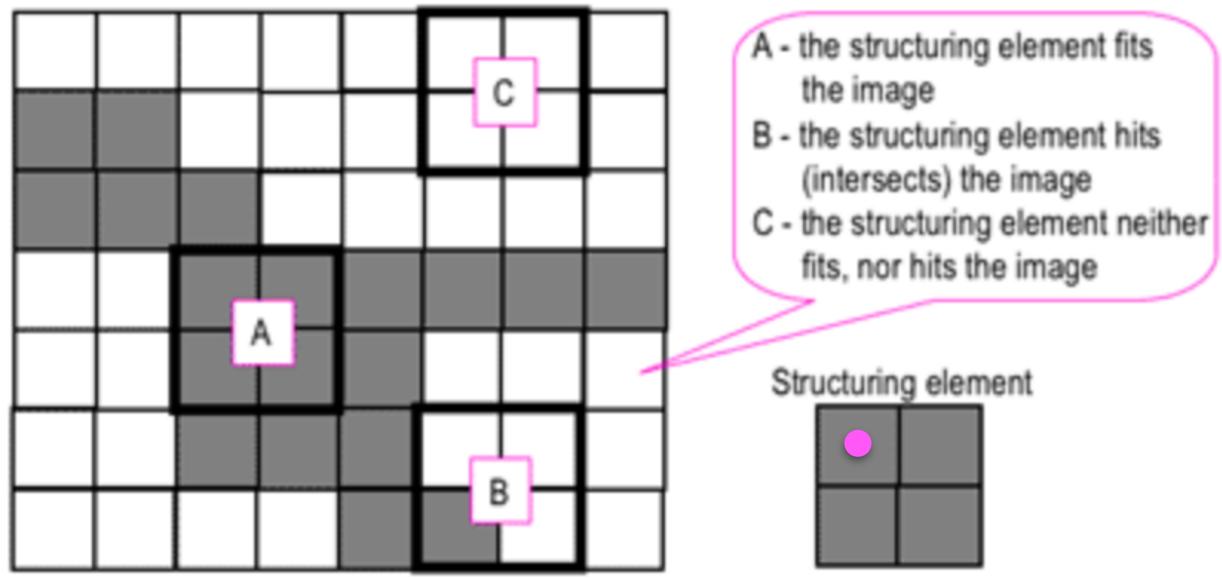
  $$\{y \mid y = x + s,\ s \in S\} \subset I$$

- ***S* hits *I* at *x* if**

  $$\{y \mid y = x - s,\ s \in S\} \bigcap I \neq \emptyset$$

- ***S* misses *I* at *x* if**

  $$\{y \mid y = x - s,\ s \in S\} \bigcap I = \emptyset$$



$$s_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad s_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

		A	B	C
fit	s_1	yes	no	no
	s_2	yes	yes	no
hit	s_1	yes	yes	yes
	s_2	yes	yes	no

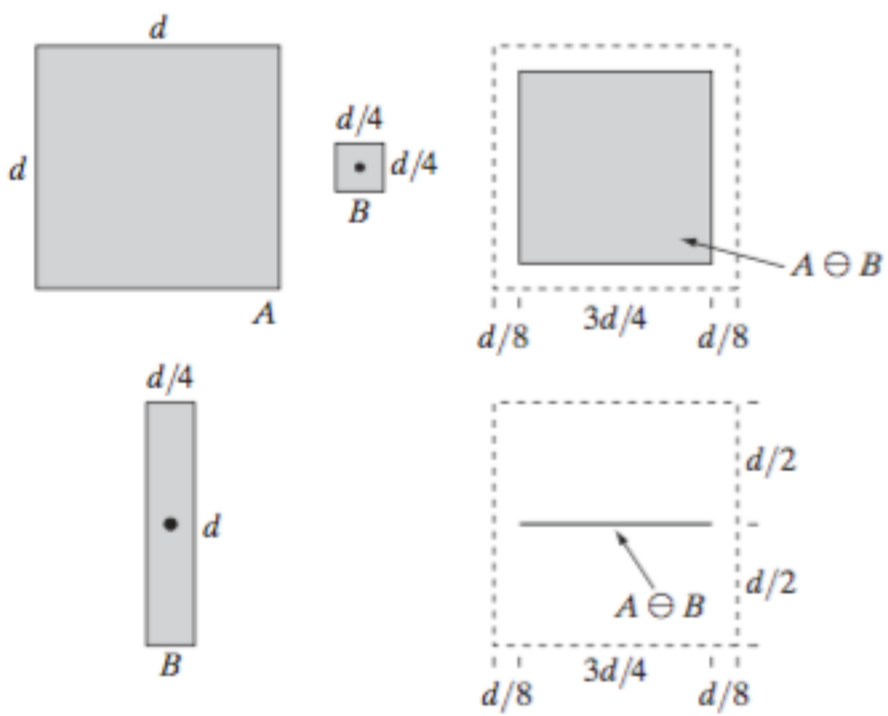
Erosion $\ominus$

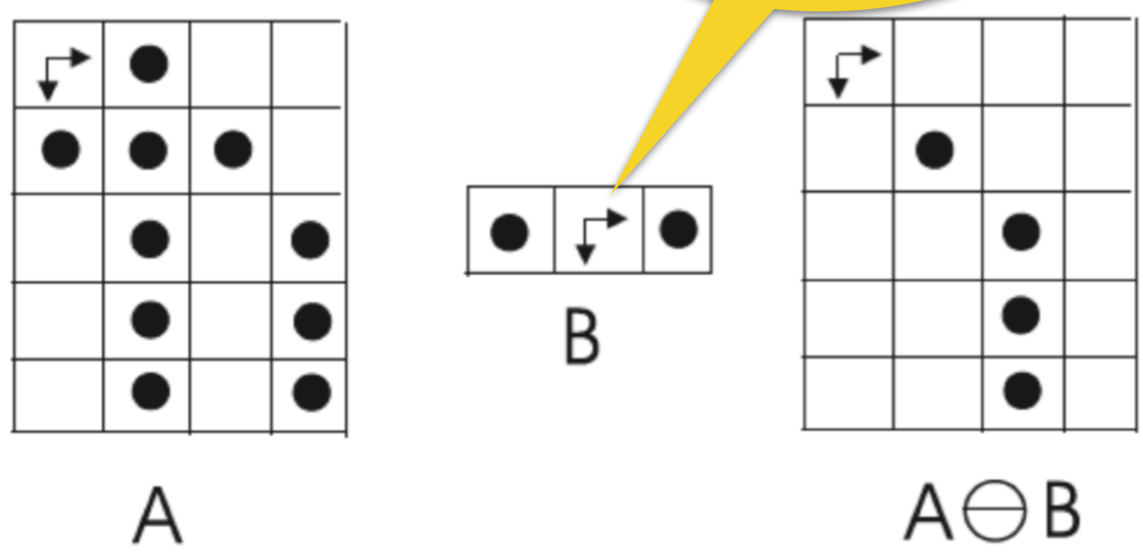
Erosion of a set *A* by a structuring element *B*

- all the point *z* in *A* such that *B* is included in *A* when the origin of *B* coincides with *z* (i.e. all the points in *A* at which *B* **fits** *A*)

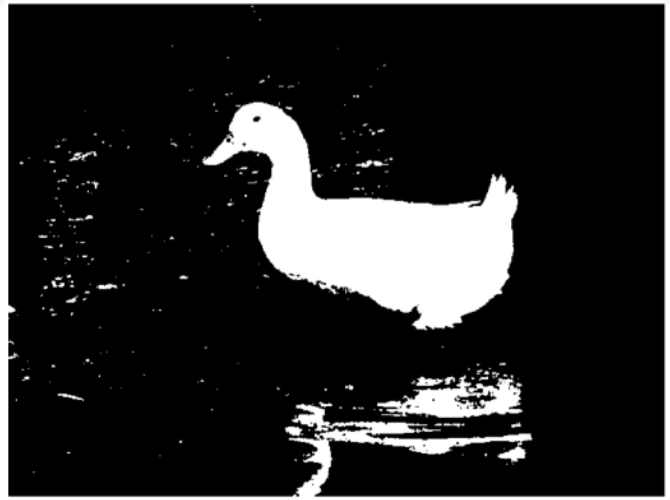
$$A \ominus B = \{z \mid (B)_z \subseteq A\} = \{z \mid (B)_z \bigcap A^c = \emptyset\}$$
$$= \{z \mid z + b \in A \text{ for every } b \in B\}$$

N.B. Here "white" means that the value can be 0 or 1





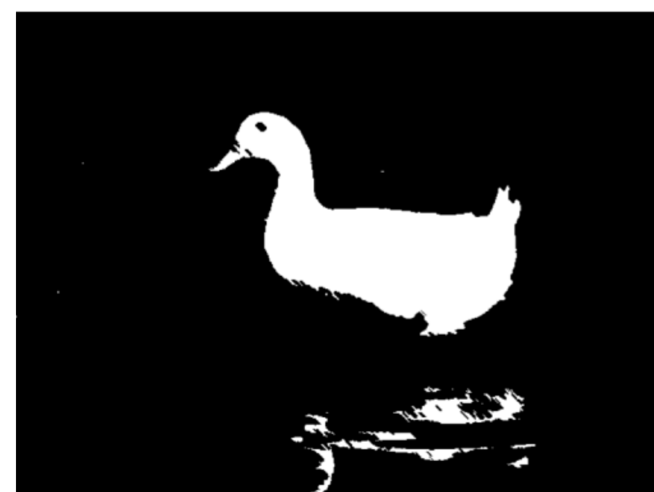
Erosion example 1



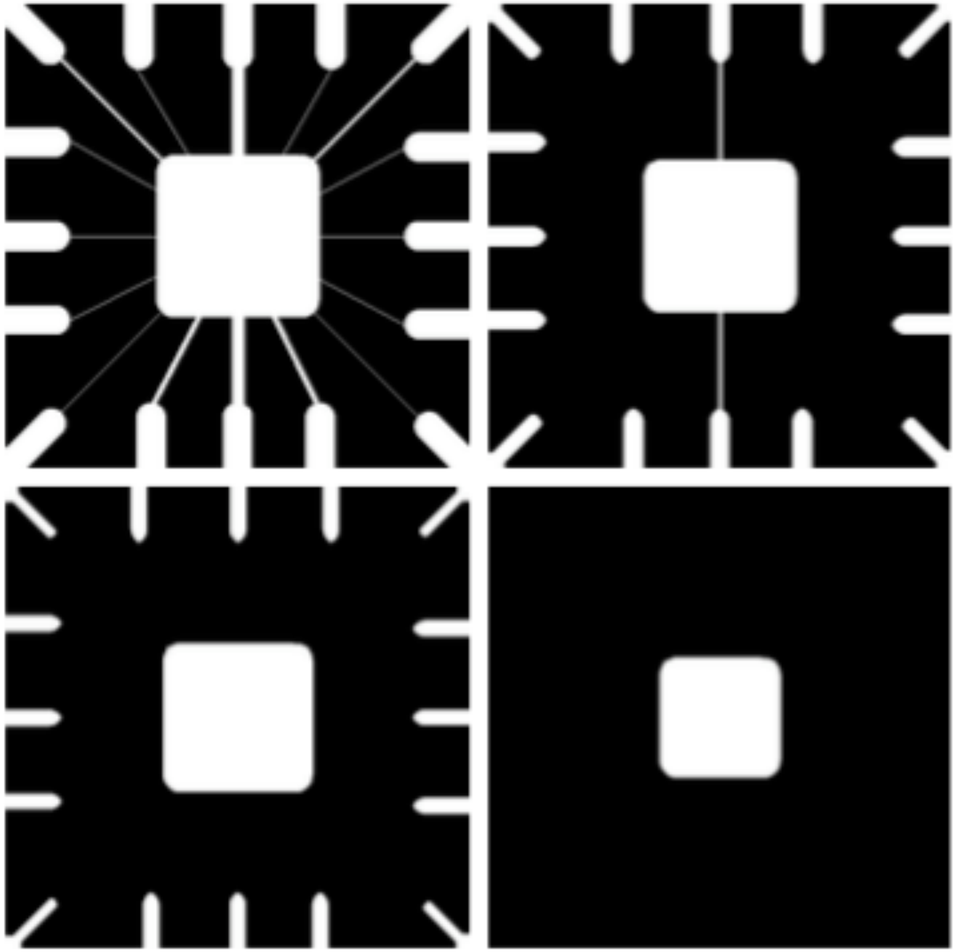
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

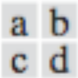
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	0

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



Erosion example 2





a b
c d

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

Erosion properties

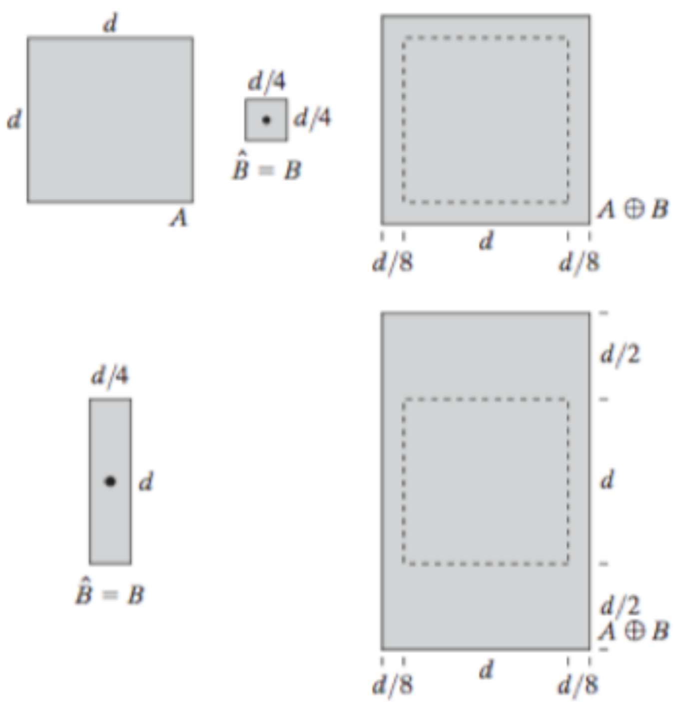
- it shrinks or thins objects in a binary images
  - acts like a “filter”
- it is not commutative
  - $A \ominus B \neq B \ominus A$
- It is an increasing transformation
  - $A \subseteq C \Rightarrow A \ominus B \subseteq C \ominus B$

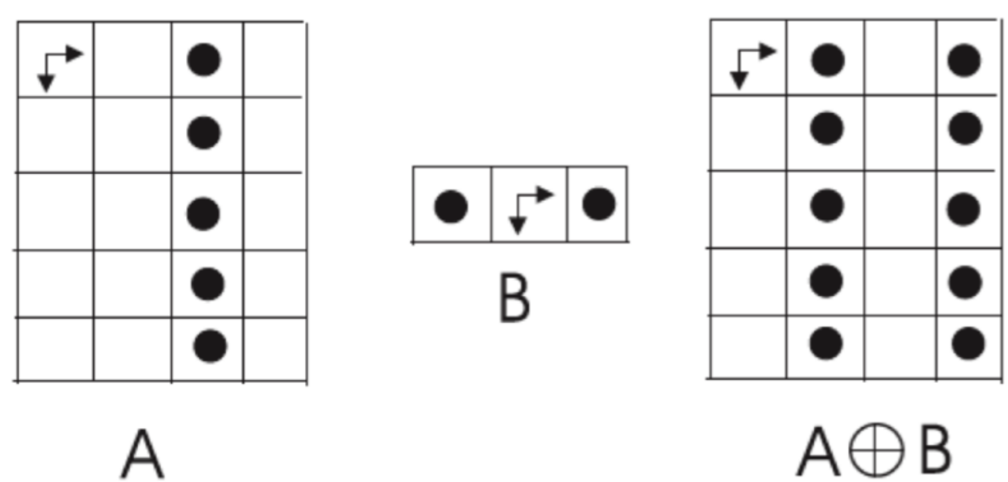
Dilation $\oplus$

Dilation of a set *A* by a structuring element *B*

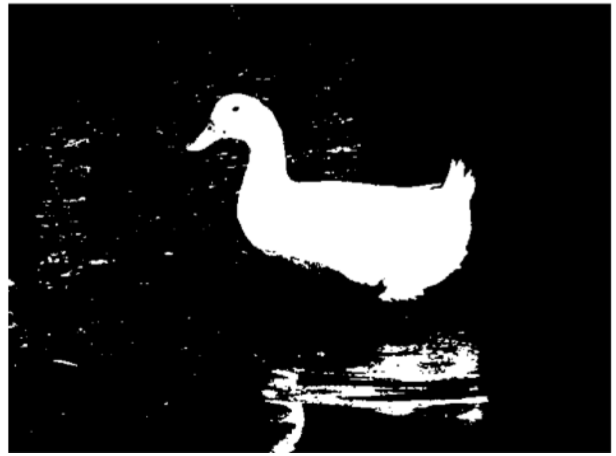
- all the point *z* in *A* such that *B* flipped and translated by *z* has non-empty intersection with *A* (i.e. all the points in *A* at which <u>flipped</u> *B* **hits** *A*)

$$A \oplus B = \left\{ z \mid (\hat{B})_z \bigcap A \neq \emptyset \right\} = \left\{ z \mid \left[(\hat{B})_z \bigcap A \right] \subseteq A \right\}$$
$$= \{ z \mid z = a + b,\ a \in A \text{ and } b \in B \}$$





Dilation example 1

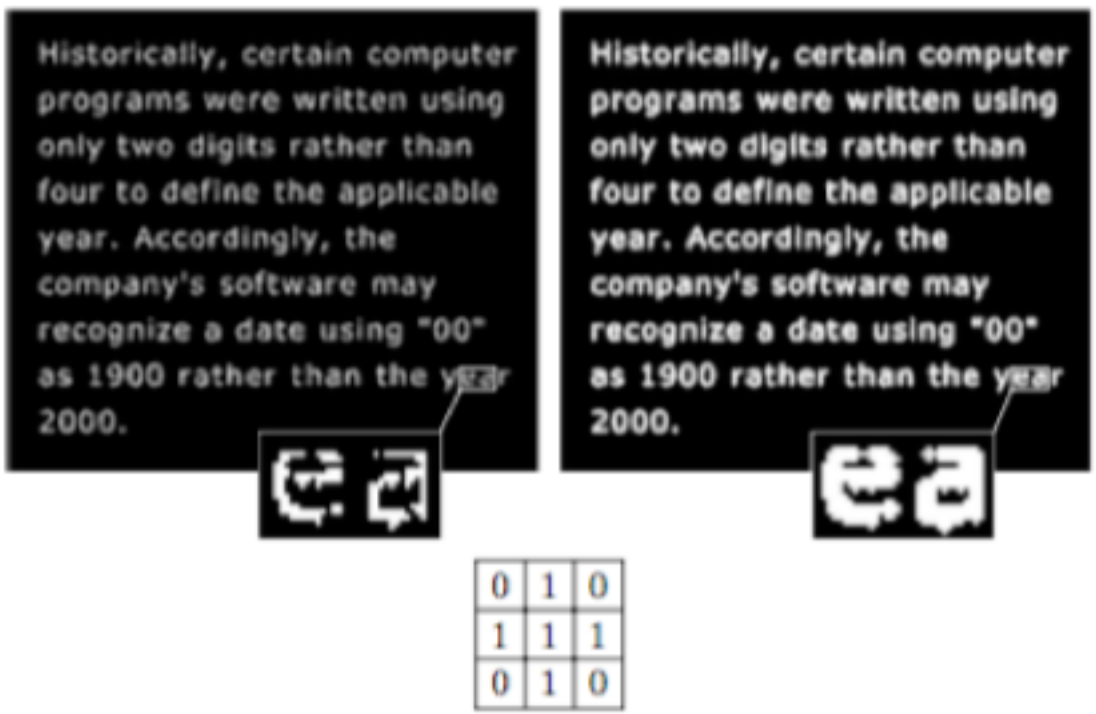


1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

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

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

Dilation example 2



a c
b

FIGURE 9.7
(a) Sample text of poor resolution with broken characters (see magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

Dilation properties

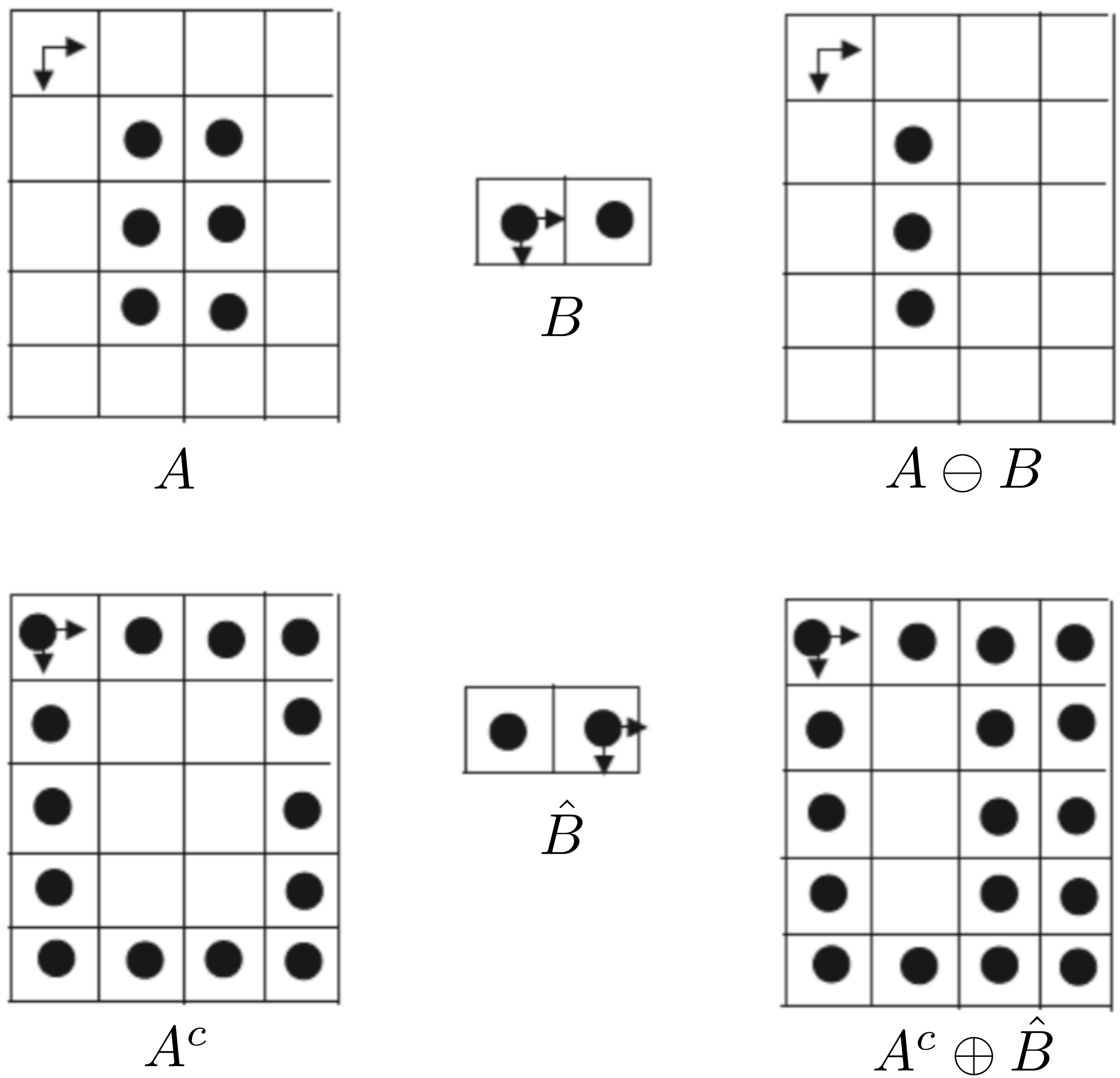
- it grows or thickens objects in a binary images
  - acts like a “spatial convolution” using the SE as convolution mask
- it is commutative
  - $A \oplus B = B \oplus A$
- it is associative
  - $A \oplus (B \oplus C) = (A \oplus B) \oplus C$
- It is an increasing transformation
  - $A \subseteq C \Rightarrow A \oplus B \subseteq C \oplus B$

Erosion vs Dilation

- Erosion
  - Removal of structures of certain shape and size given by the SE
- Dilation
  - Filling of holes or adding borders of certain shape and size given by the SE
- Erosion and dilation are dual operations
  - $(A \ominus B)^c = A^c \oplus \hat{B}$
  - $(A \oplus B)^c = A^c \ominus \hat{B}$

If B is symmetric with respect to its origin, we can obtain the erosion of the image A by B simply by dilating the background of the image with the same B and complementing the result.

Duality example



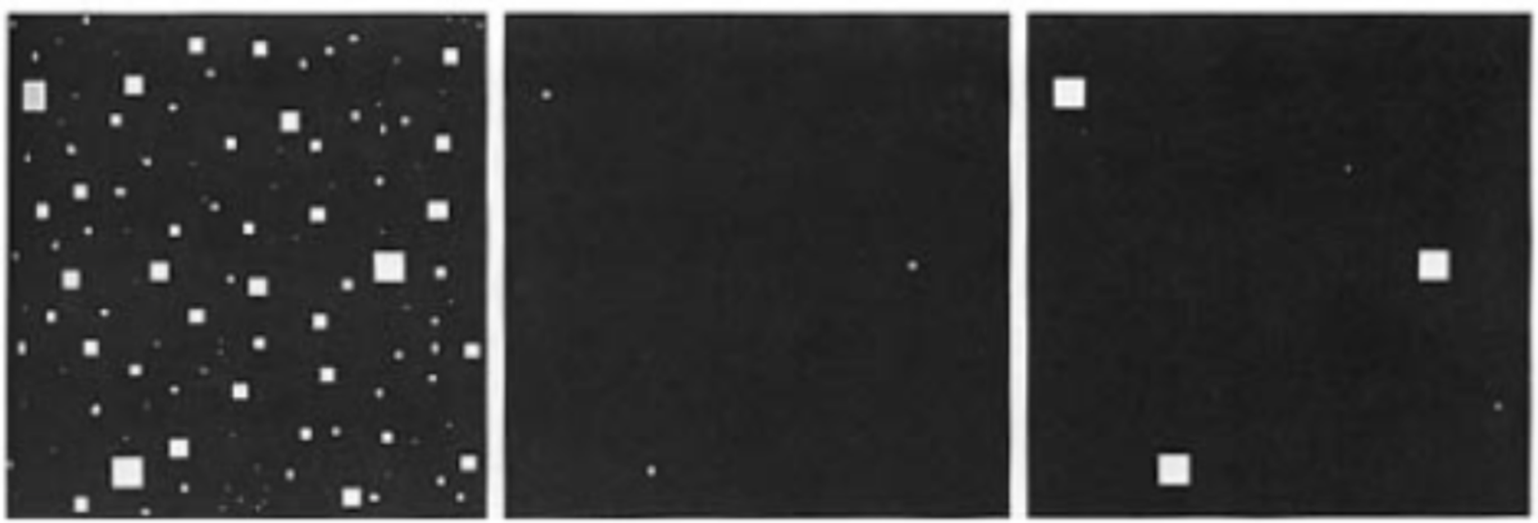
Combining erosion and dilation

Why?

- To remove structures and/or fill holes without affecting the remaining parts

How?

- by combining erosion and dilation using the same structuring element



a b c

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

Opening $\bigcirc$

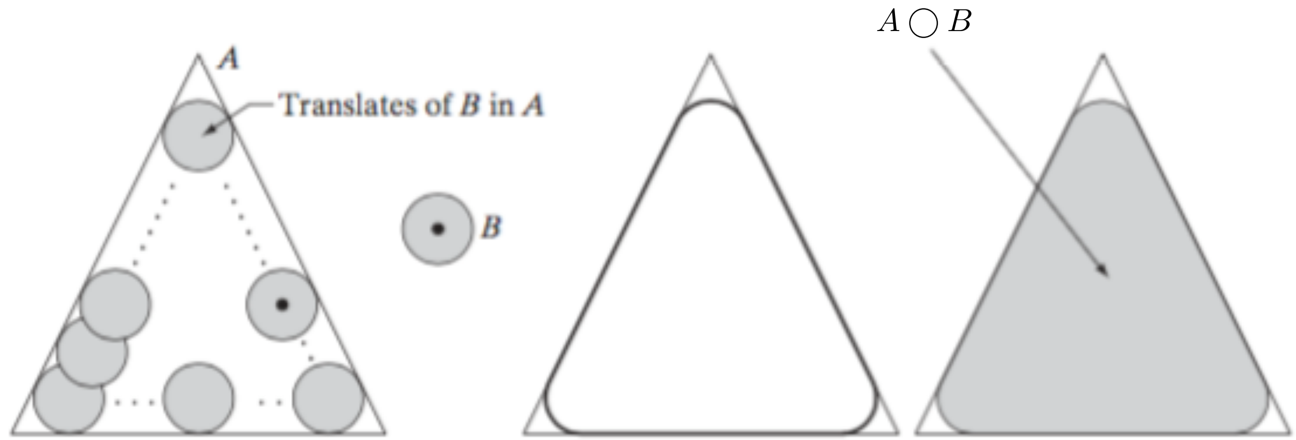
Opening of a set *A* by the structuring element *B*

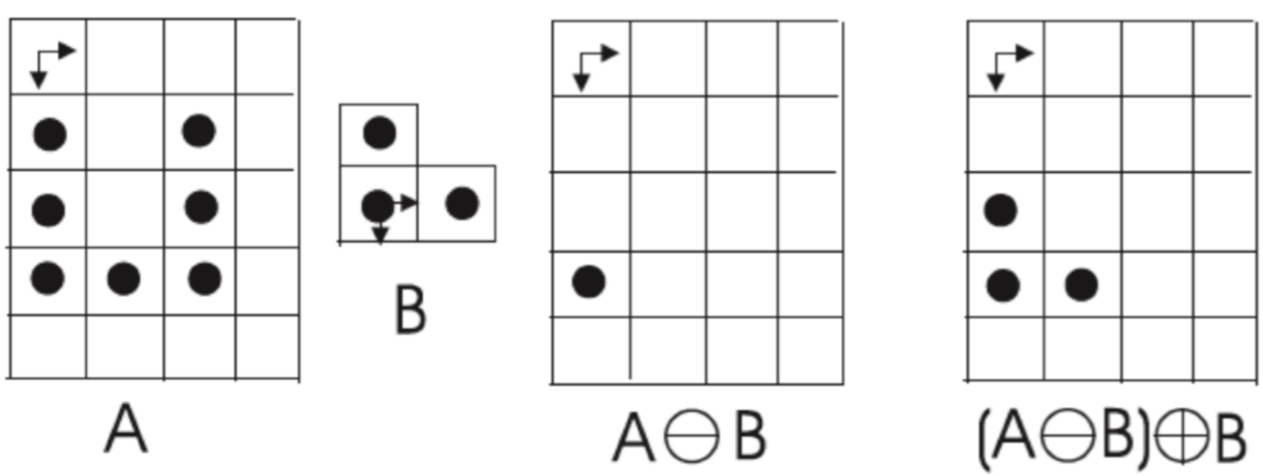
- erosion of *A* by *B* followed by a dilation of the result by *B*

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

Properties

- smoothes the contour
- breaks narrow isthmuses
- eliminates protrusions





Closing ●

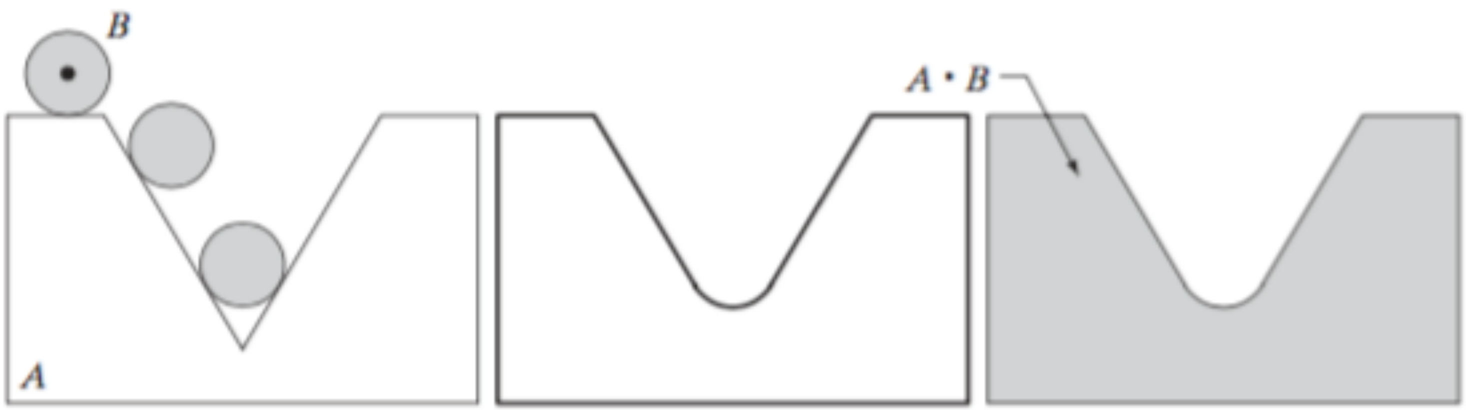
Closing of a set *A* by the structuring element *B*

- dilation of *A* by *B* followed by an erosion of the result by *B*

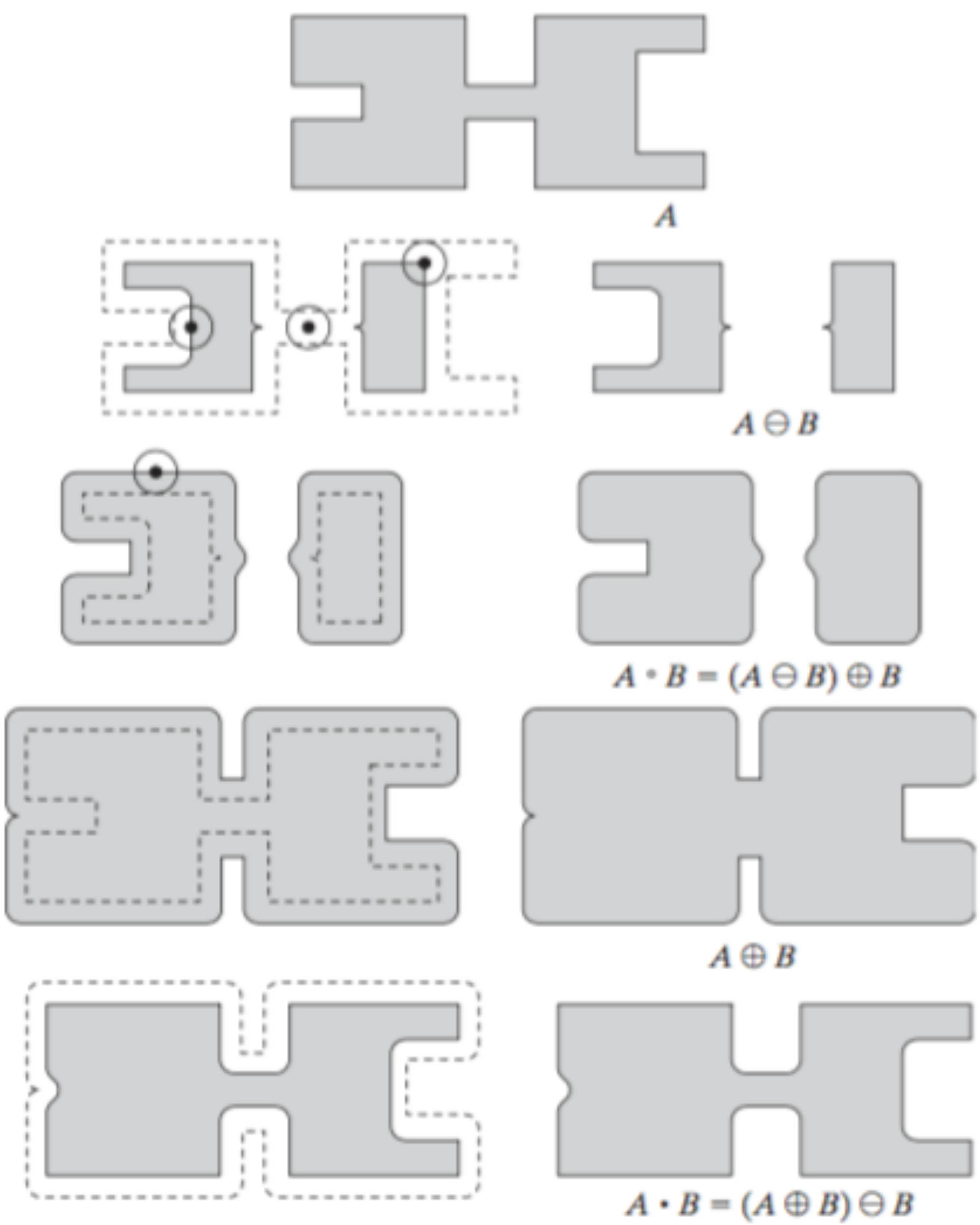
$$A \bullet B = (A \oplus B) \ominus B$$

Properties

- smoothes sections of the contour
- fuses narrow breaks and long thin gulfs
- eliminates small holes
- fills gaps in contour

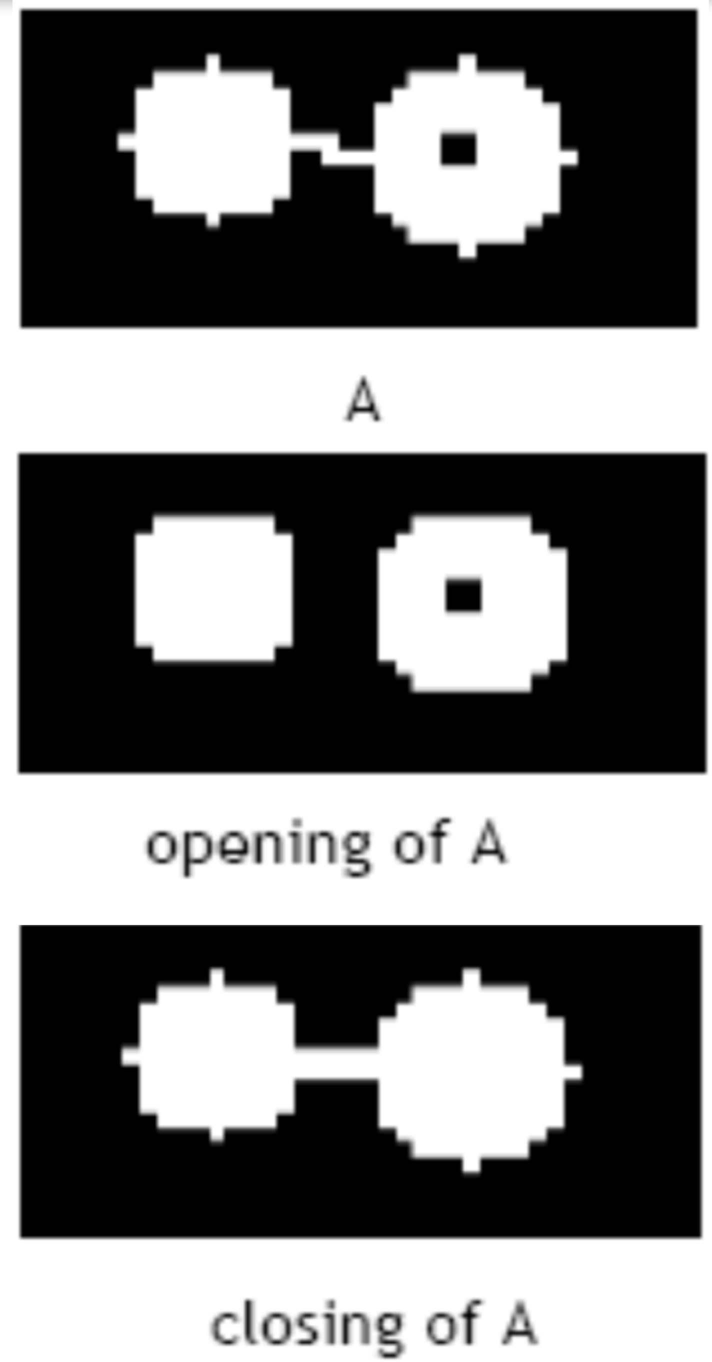


Opening/Closing example 1

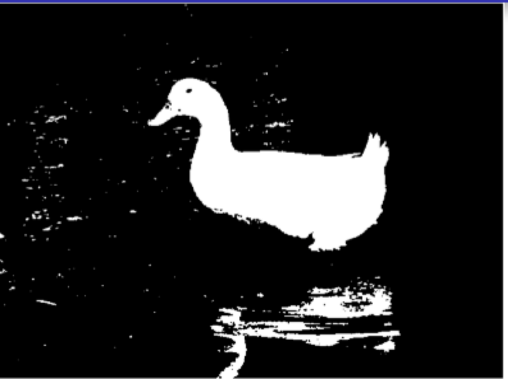


a
b c
d e
f g
h i

FIGURE 9.10 Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The SE was not shaded here for clarity. The dark dot is the center of the structuring element.



Opening/Closing example 2



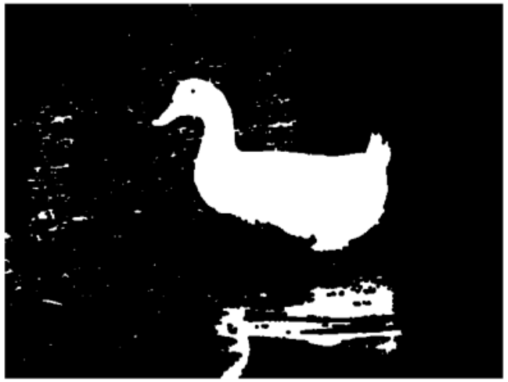
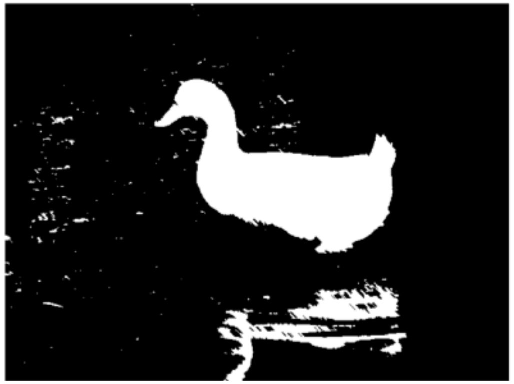
Structuring element

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

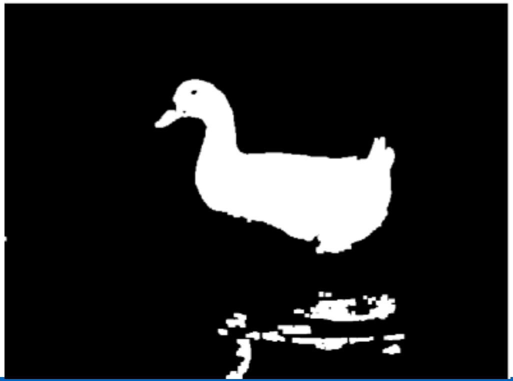
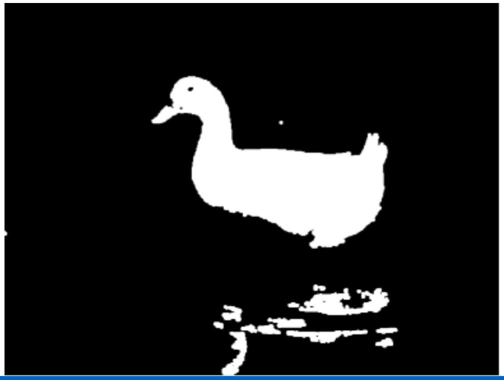
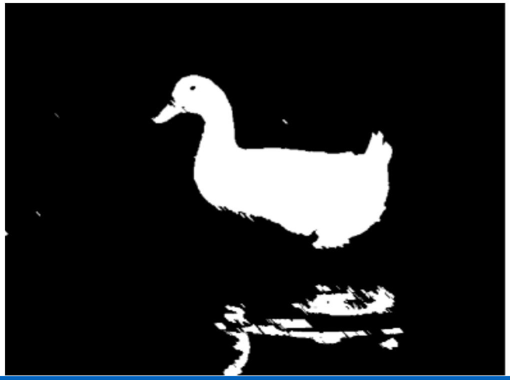
0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	0

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

Closing

Opening

Properties

Opening

- $A \bigcirc B$ is a subset (subimage) of *A*
- It is an increasing function $C \subseteq D \Rightarrow C \bigcirc B \subseteq D \bigcirc B$
- It is ***idempotent***, i.e. $(A \bigcirc B) \bigcirc B = A \bigcirc B$

Closing

- *A* is a subset (subimage) of $A \bullet B$
- It is an increasing function $C \subseteq D \Rightarrow C \bullet B \subseteq D \bullet \mathrm{B}$
- It is ***idempotent***, i.e. $(A \bullet B) \bullet B = A \bullet B$

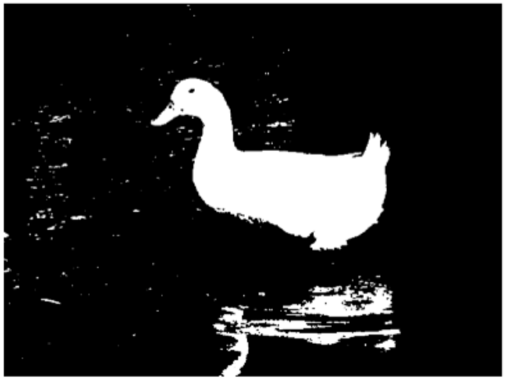
Opening and closing are dual operations

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

Morphological filtering

- To **remove holes** in the foreground **and islands** in the background
  - do both opening and closing
- The **size and shape of the structuring element** determines which feature will survive after the filtering
- In the **absence of knowledge** about the shape of the features to remove, use a **circular SE**

Morphological filtering example



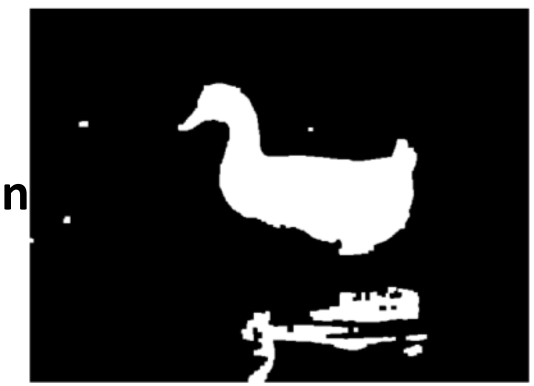
Structuring element

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

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

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

Close and then open



Open and then close



Hit-or-Miss Transformation $\circledast$

- Searches for an **exact match** of the structuring element
- Given a structuring element *B=(B₁ U B₂),* the **match of *B* in *A*** is

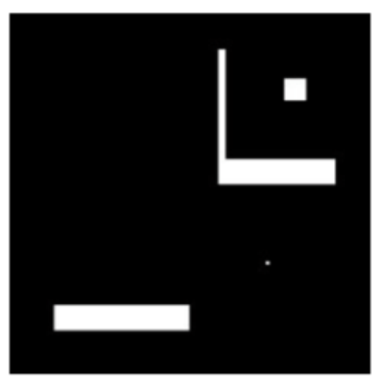
$$A \circledast B = (A \ominus B_1) \bigcap (A^c \ominus B_2)$$

- B_1 finds a match in A (***hit***)
- B_2 finds a match in A^c (***miss***)

Original Image A

$(A \ominus B_1)$

B_1 and B_2



$(A^c \ominus B_2)$

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

Hit-or-Miss Transformation example 1

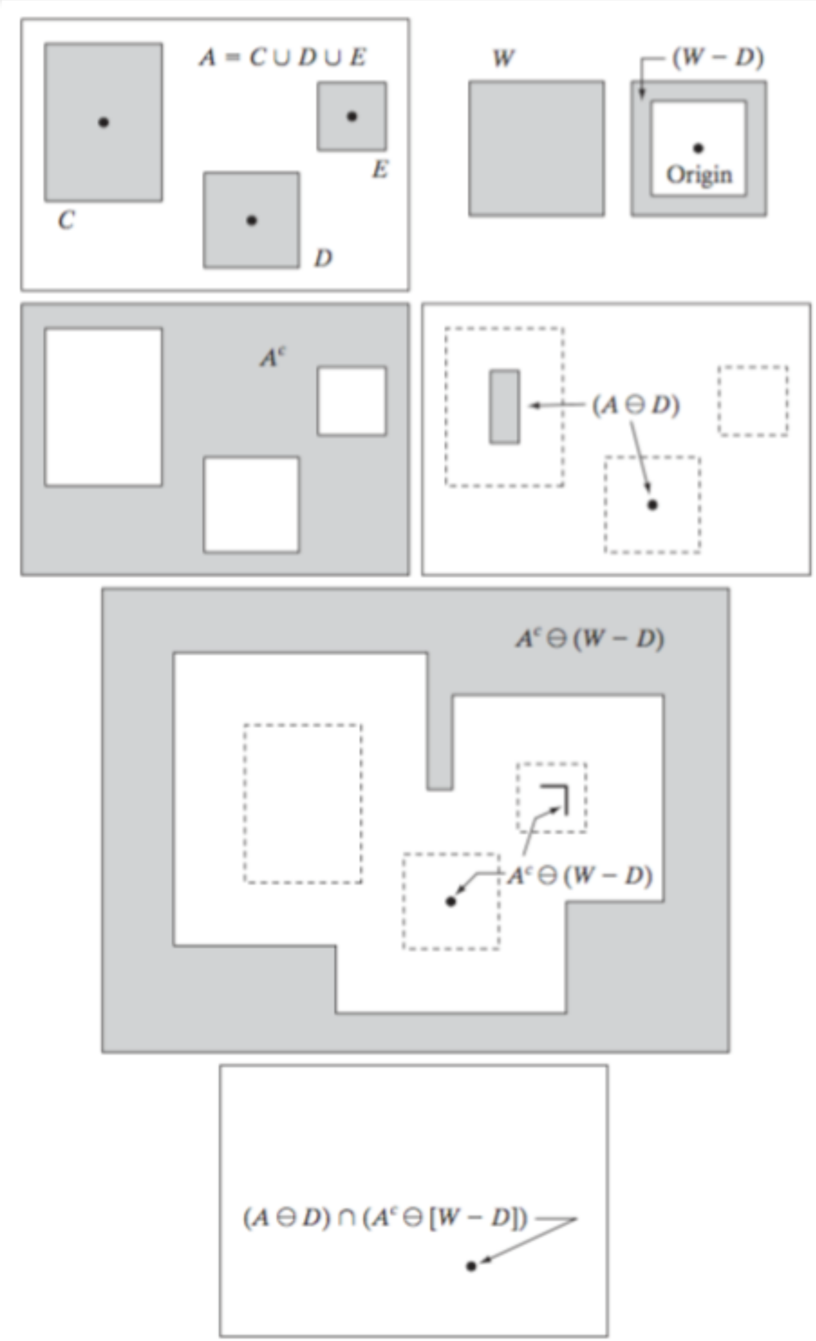
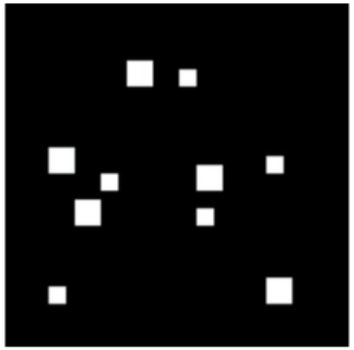


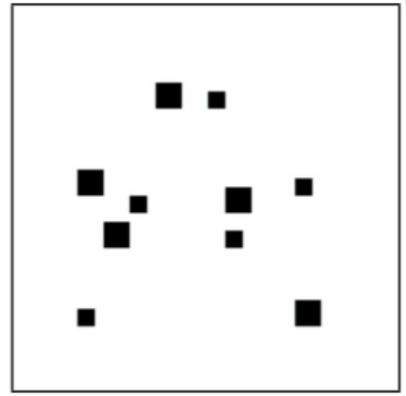
FIGURE 9.12 (a) Set A. (b) A window, W, and the local background of D with respect to W, $(W - D)$. (c) Complement of A. (d) Erosion of A by D. (e) Erosion of A^c by $(W - D)$. (f) Intersection of (d) and (e), showing the location of the origin of D, as desired. The dots indicate the origins of C, D, and E.

Hit-or-Miss Transformation example 2



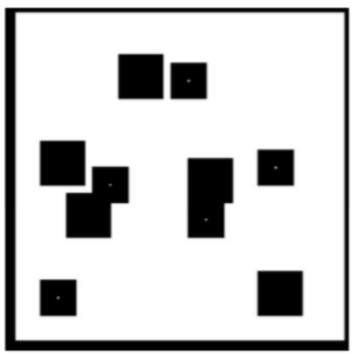
Original Image A

$C_1 = (A \ominus B_1)$



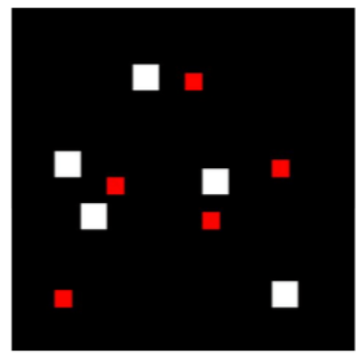
A^c

B_1 and B_2



$C_2 = (A^c \ominus B_2)$

$(A \circledast B) = C_1 \cap C_2$

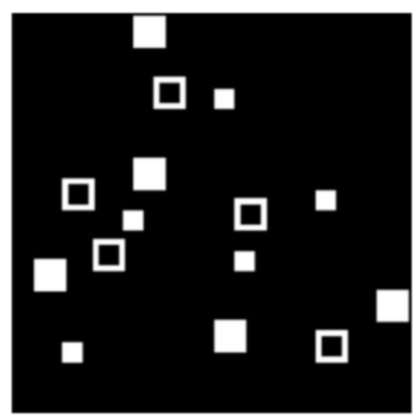


Detected objects shown in red.

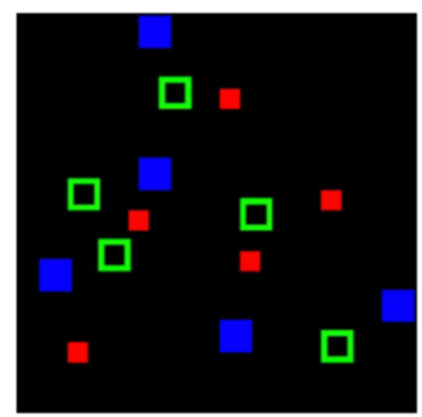
$$B = (S_1, S_2)$$

$$B = (S_2, S_3)$$

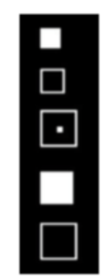
$$B = (S_4, S_5)$$



Original image



Search results



S_1
S_2
S_3
S_4
S_5

Morphological operators

Part II

Applications

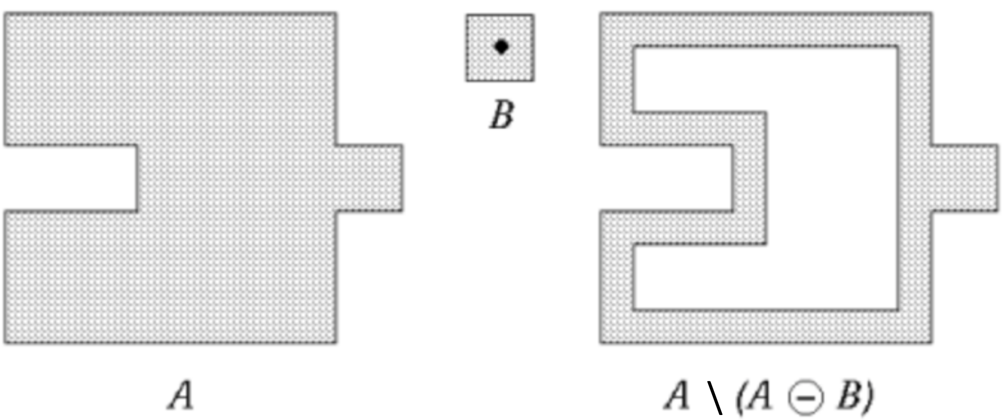
Why do we use morphological operators?

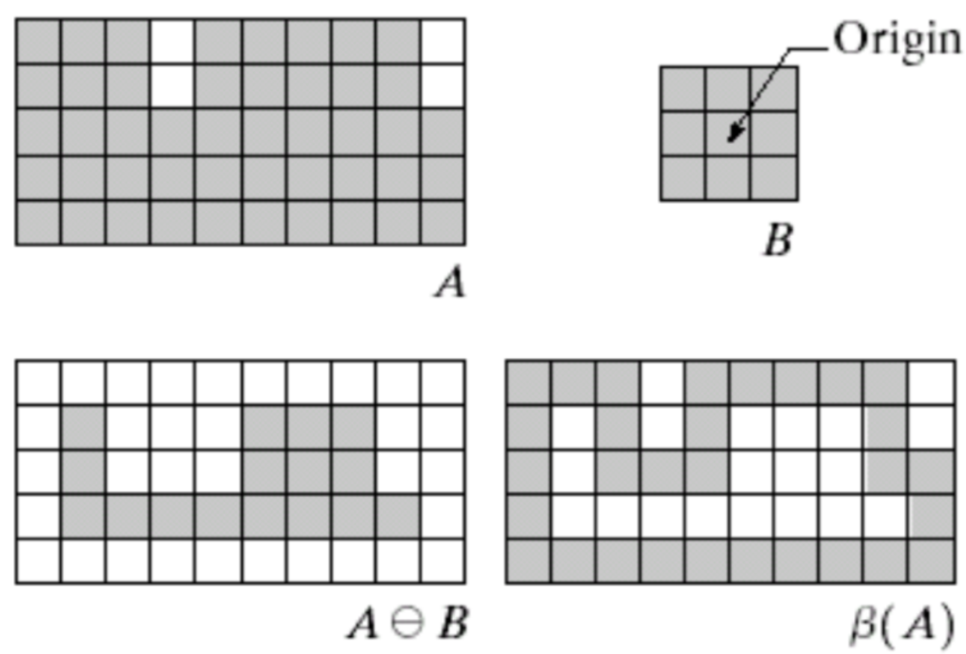
- Representation and description of a shape
  - extracting boundaries
  - extracting connected components
  - extracting the convex hull
  - skeletonize a region
- Pre/Post-processing steps
  - region filling
  - region thinning
  - region thickening

Boundary extraction

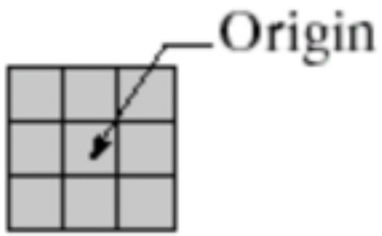
- Combining erosion by a structuring element B and difference between sets

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





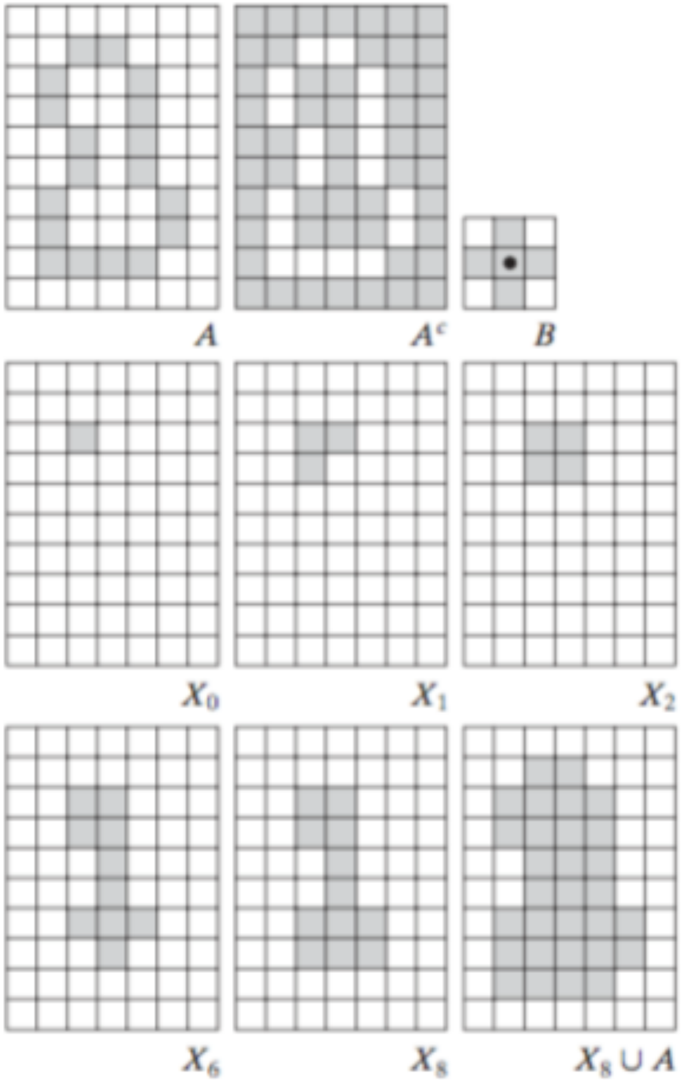
Boundary extraction example



Holes filling

Combining set dilation, complementation and intersection

- A ***hole*** is defined as a background region surrounded by a connected border of foreground pixels



Algorithm

- Given a point *P* inside a hole set $X_0 = P$
- while $X_k \neq X_{k-1}$ do

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

  end
- $X_F = X_k \bigcup A$

We call this ***conditional dilation***

Holes filling example

a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

Extraction of connected components

Combining set dilation and intersection

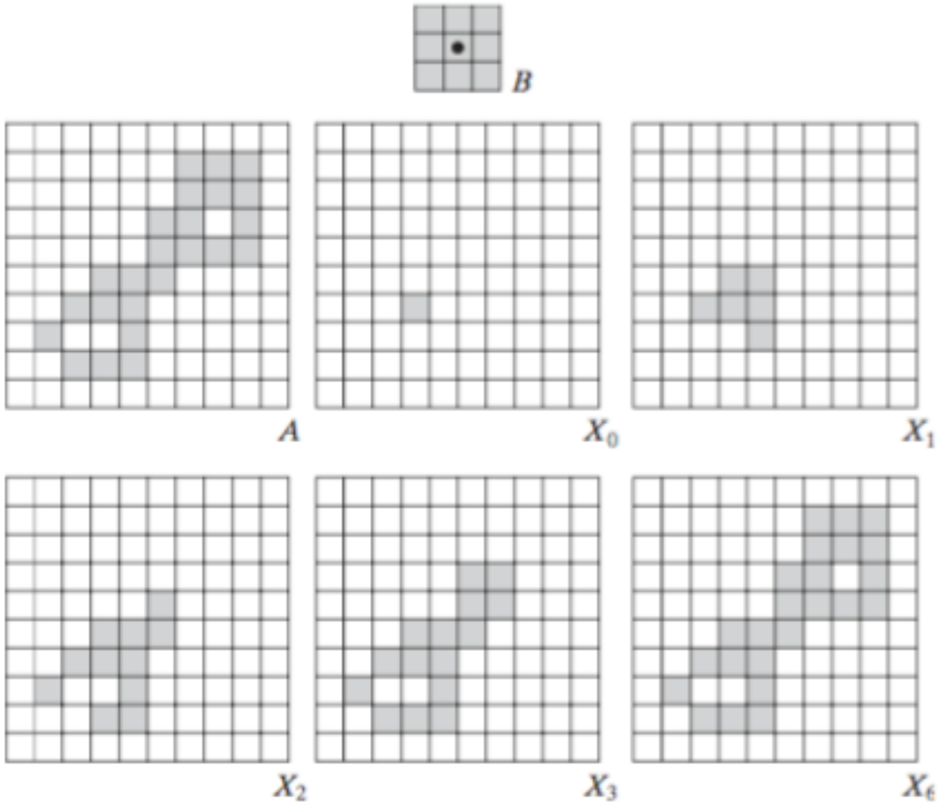
- A ***connected component S*** is defined as all the pixels in *S* for which there exists a path that connect them consisting entirely of pixels in *S*

Algorithm

- Given a point *P* inside *A* set $X_0 = P$
- while $X_k \neq X_{k-1}$ do

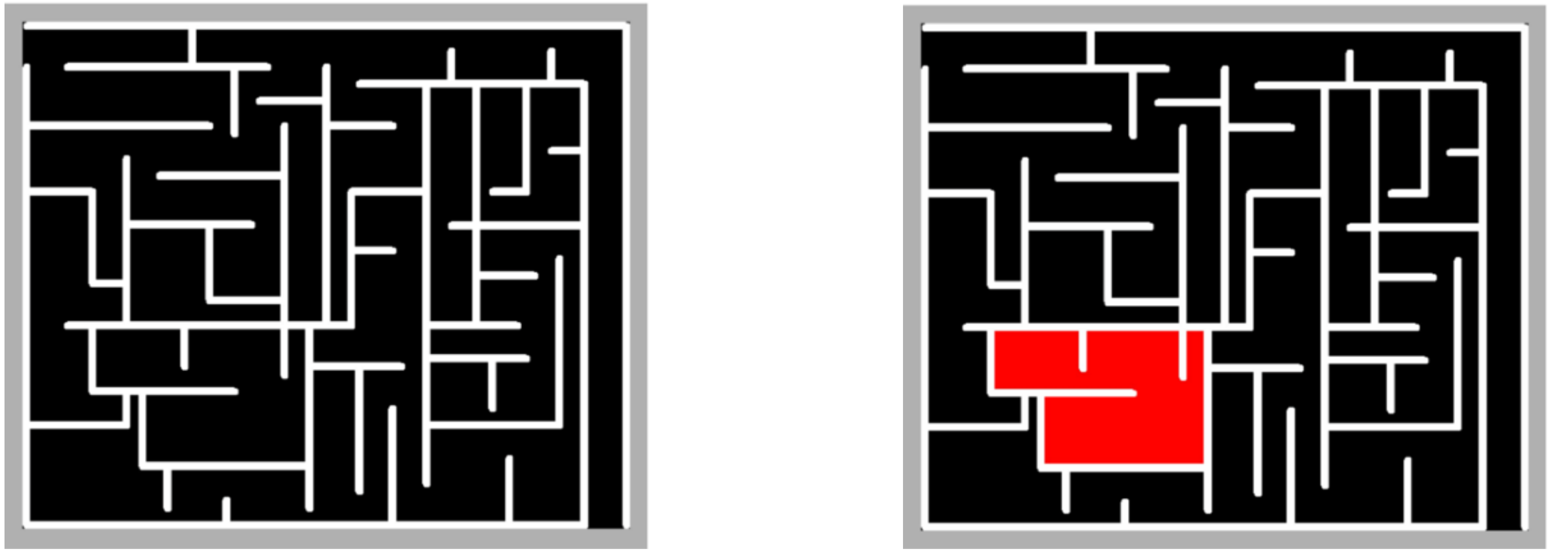
$$X_k = (X_{k-1} \oplus B) \bigcap A$$

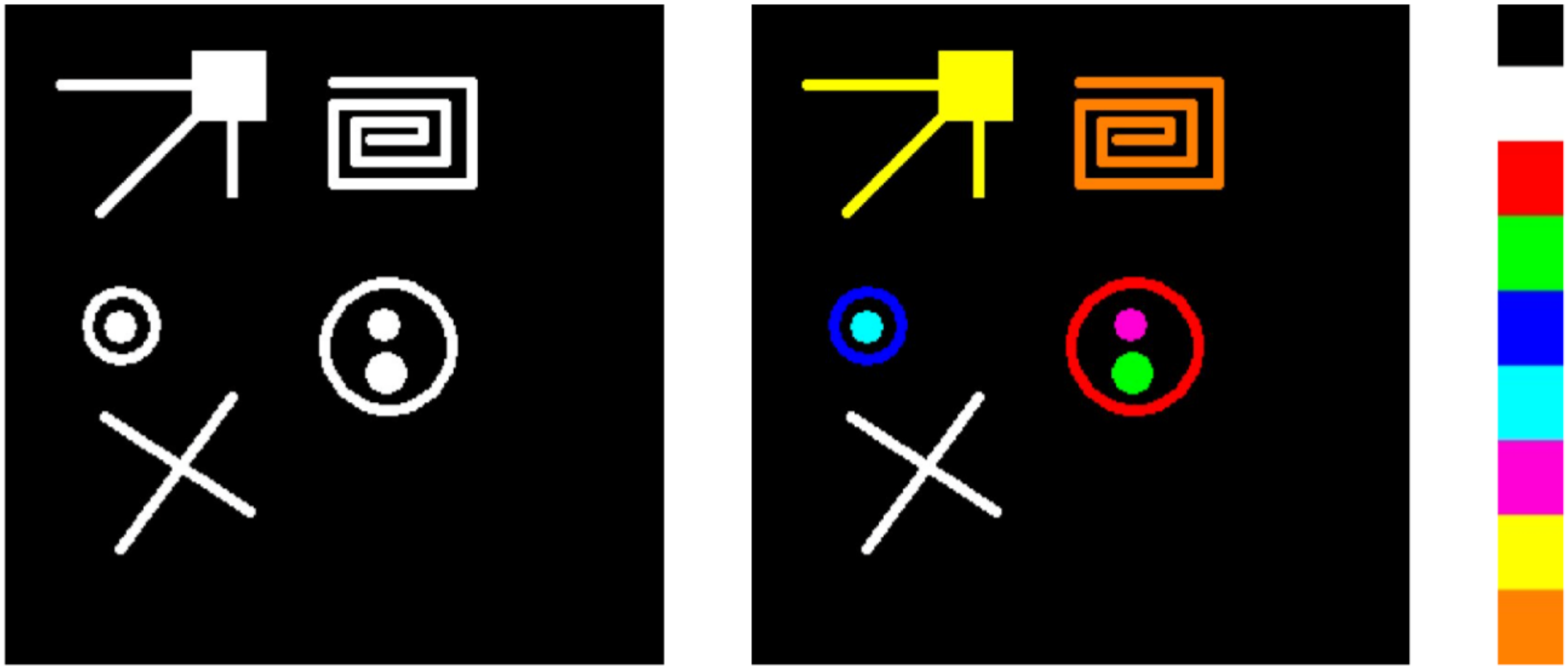
end



It is again a ***conditional dilation***

Connected components example 1





Convex Hull

Combining hit-or-miss transformation and union

- A set *A* is said to be ***convex*** if the straight line segment joining ***any*** two points in *A* lies ***entirely*** within *A*
- The ***convex hull C(A)*** of an arbitrary set *A* is the ***smallest*** convex set containing *A*

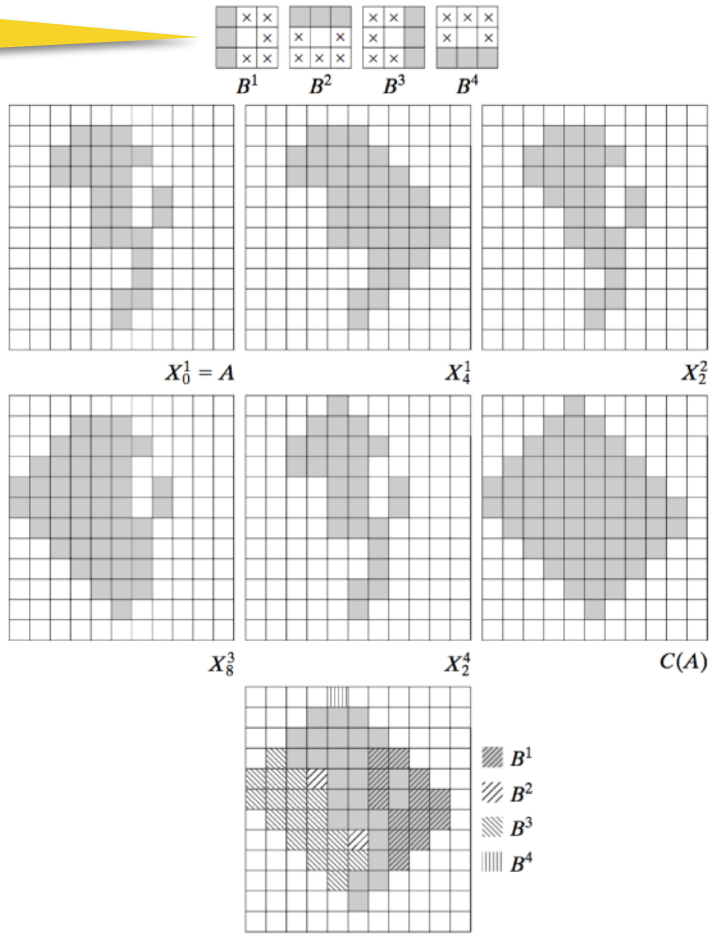
N.B. Here "x" means that the value can be 0 or 1

Algorithm

- Define 4 structuring elements $B^i \; i = 1,2,3,4$
- Initialize $X_0^i = A \; i = 1,2,3,4$
- while $X_k^i \neq X_{k-1}^i$ do

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

  end
- $C(A) = \bigcup_{i=1}^{4} X_k^i$
- To get the ***smallest*** convex set containing A we limit growth to the min and max coordinates of A



Thinning $\otimes$

Combining hit-or-miss transformation and intersection

- removes pixels from the set *A* using a structuring element *B* until only a narrow set remains

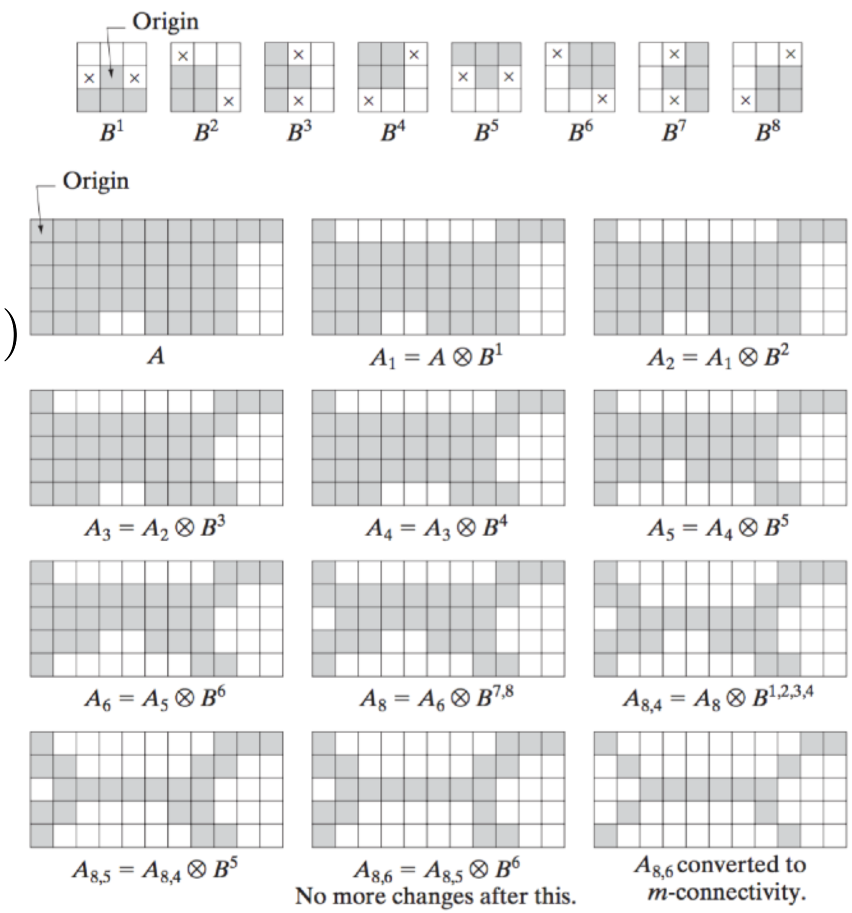
$$A \otimes B = A \bigcap (A \circledast B)^c$$

Algorithm

- Given a sequence of n SEs $\{B\} = (B^1, \ldots, B^n)$
- Let $X_n = A$ and $Y = X_0 = 0_{N \times M}$
- while $X_n \neq Y$ do

$$Y = X_n$$
$$X_0 = X_n$$
$$X_i = X_{i-1} \bigcap (X_{i-1} \circledast B^i)^c \; i = 1, \ldots, n$$

end



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

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 four elements again. (l) Result after convergence. (m) Conversion to *m*-connectivity.

Thinning example

The basic structure is captured by the thinned objects (red). The small ends could be removed by further processing to refine the result.

Thickening $\odot$

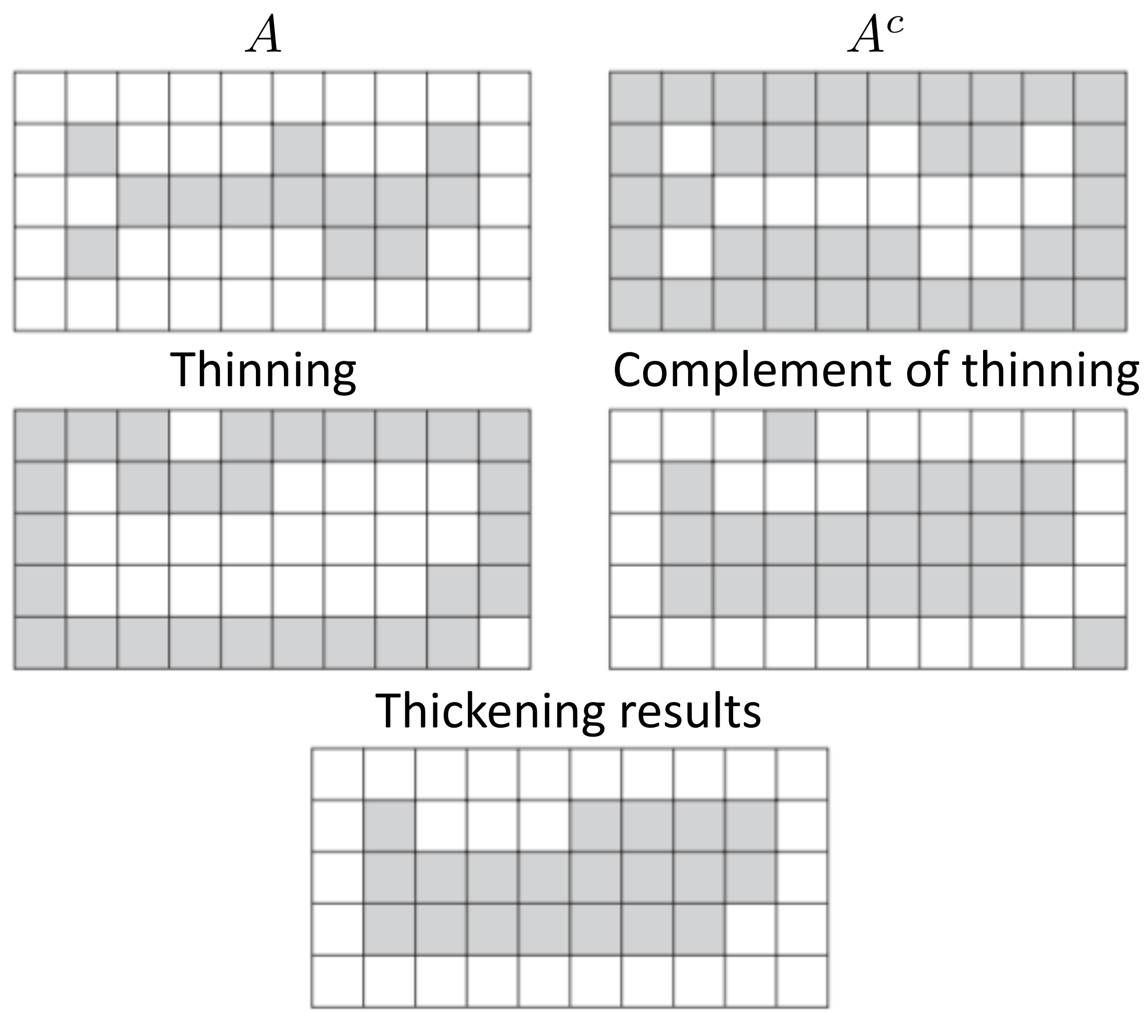
Hit-or-miss transformation and union

- is the morphological **dual of thinning** and is defined by

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

Algorithm

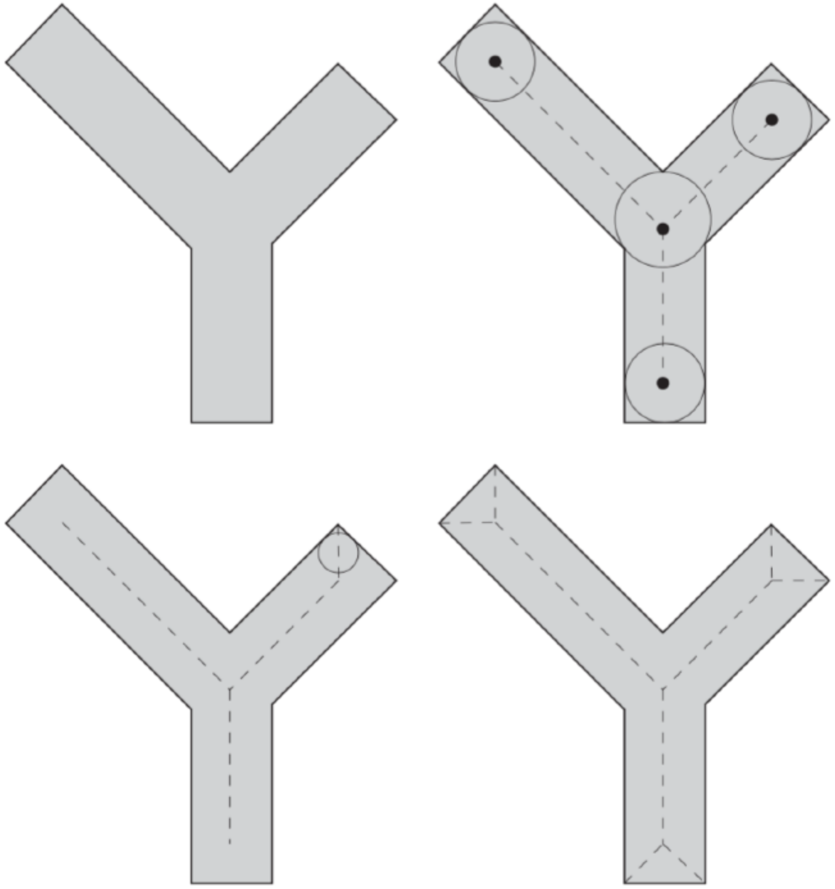
- Compute A^c
- Perform thinning algorithm of using the sequence of n SEs $\{B\} = (B^1, \ldots, B^n)$
- Take only the connected component containing *A*



Skeletonization

It is expressed in terms of erosions and openings

- ***Maximum disk***: largest disk included in *A*, touching the boundary of *A* at two or more different places
- ***Skeleton***: set of centers of the maximum disks



Algorithm

- Using e structuring element *B*
- The k-th skeleton subset is

$$S_k(A) = (A \ominus kB) \setminus (A \ominus kB) \bigcirc B$$

- The skeleton is the union of all the subsets

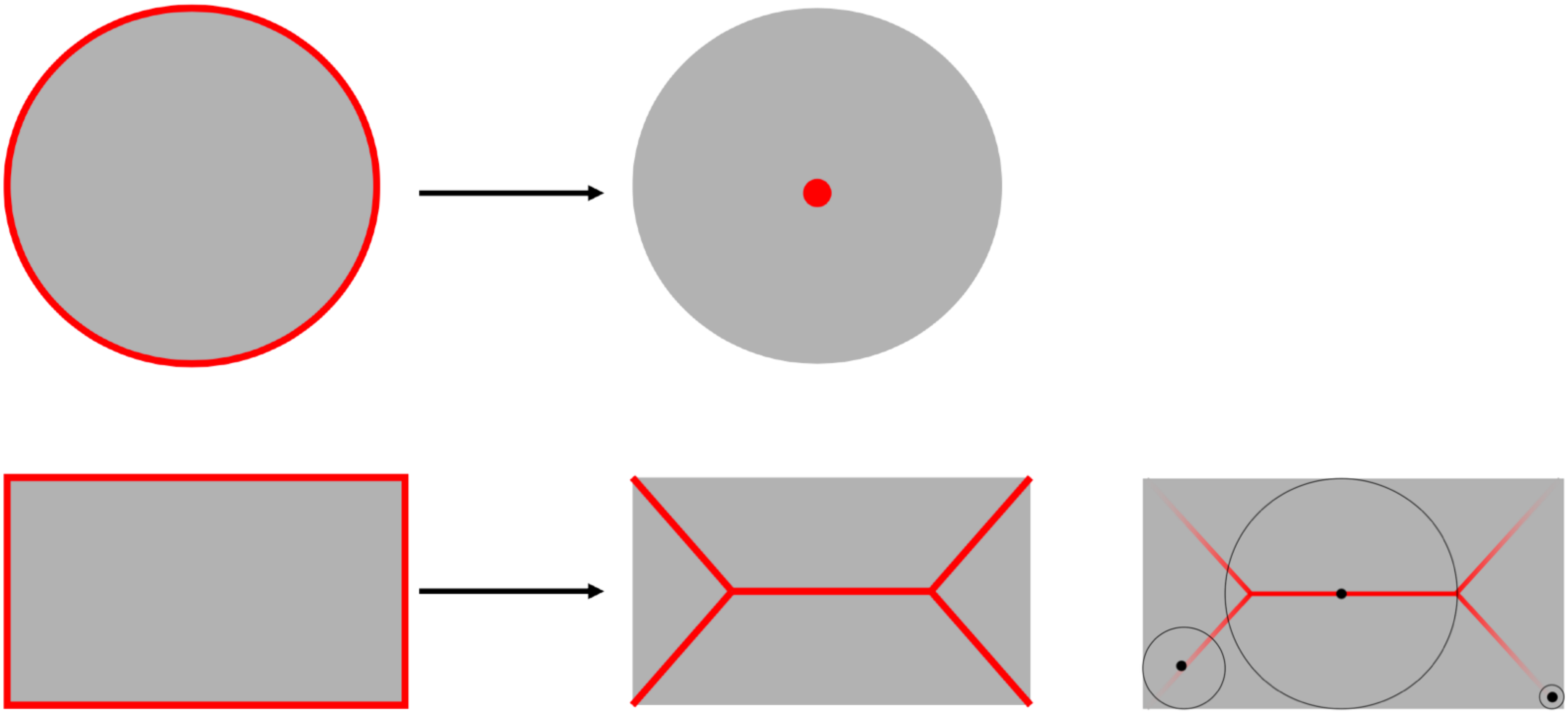
$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

$$K = \max\{k \mid (A \setminus kB) \neq \emptyset\}$$

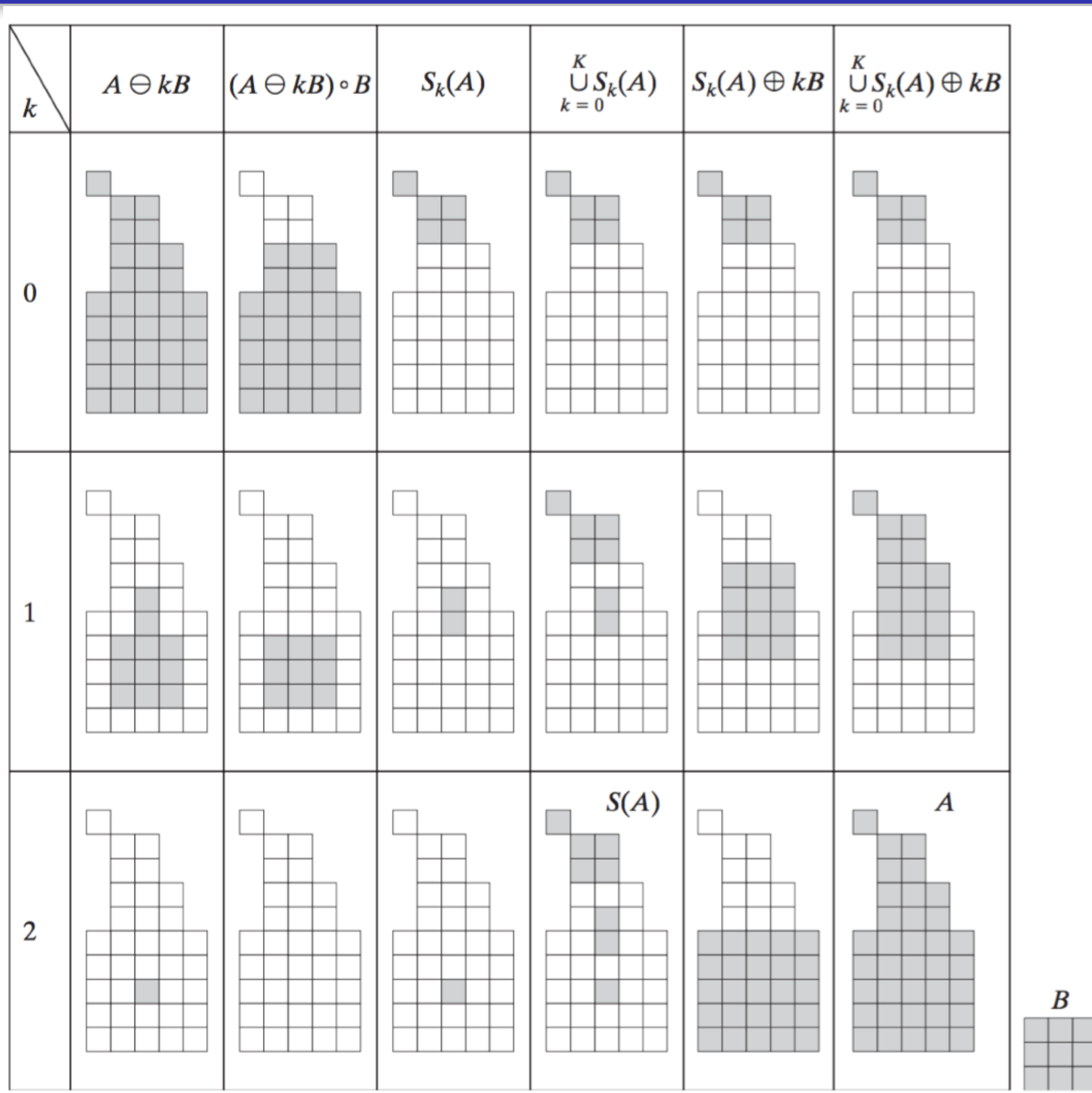
- We can obtain the original image using dilation

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

Skeleton example 1

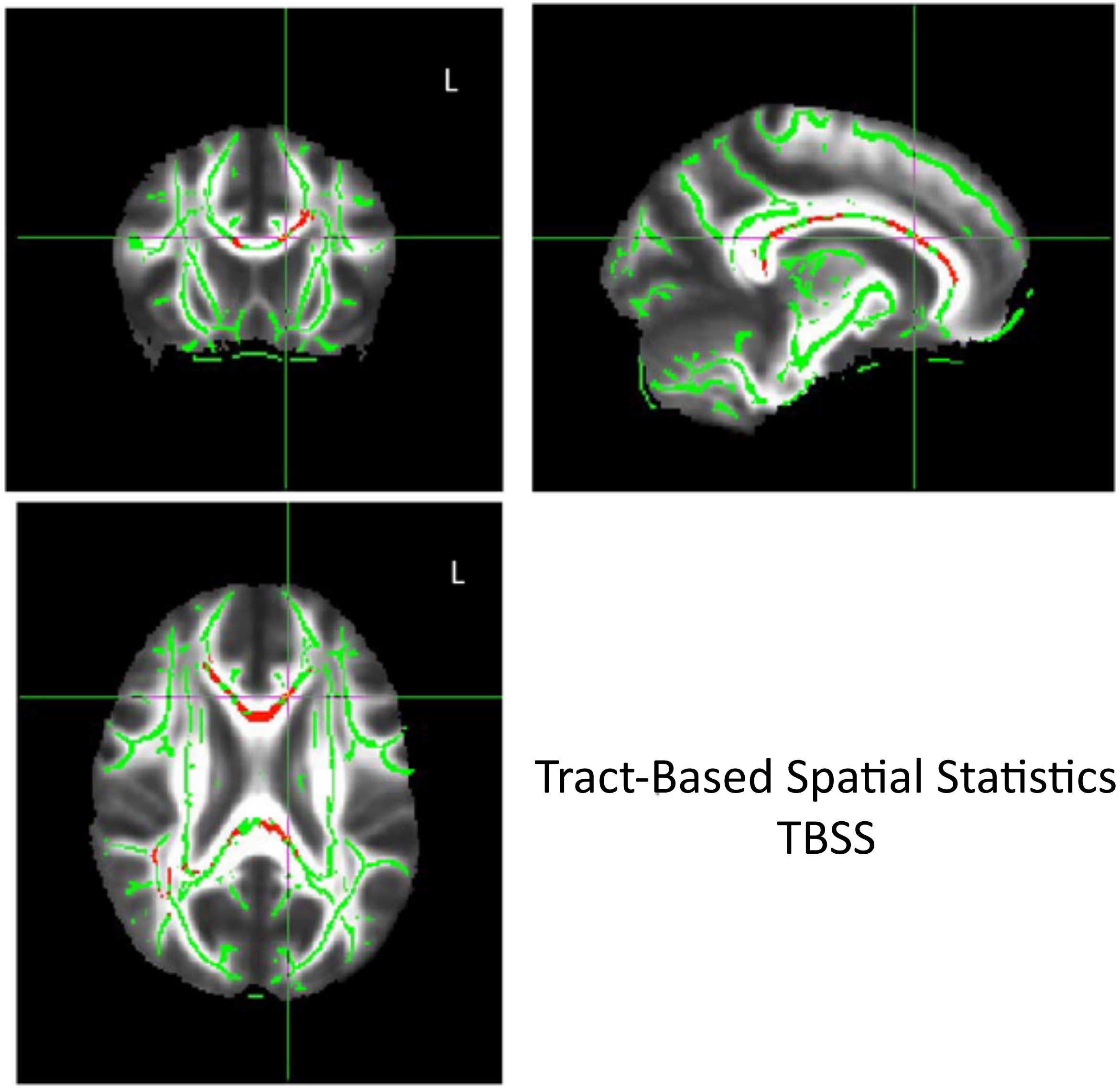


Skeletonization example 2



Application to medical imaging

- Very useful in medical imaging to compare patients and healthy controls brain structures.



Tract-Based Spatial Statistics
TBSS