Edge Detection and Morphology

01

Edge Detection

Edge Detection

Edge detection

- It is a manner for image segmentation with lower computation cost
- There are two gradient magnitude (g) types:
  - Template matching (TM)
    - Prewitt, Kirsch, and Robinson
  - Differential gradient (DG)
    - Roberts, Sobel, and Frei-Chen

TM Method

$g = \max(g_i : i = 1, \ldots, n)$

DG Method

$g = (g_x^2 + g_y^2)^{1/2}$

or

$g = |g_x| + |g_y|$ or $g = \max(|g_x|, |g_y|)$

Edge Detection

Edge detection

- How to calculate the g_x and g_y? On the other word, how to obtain the first-order derivation at an arbitrary point x of one-dimensional function f(x)
- A Taylor series

$$f(x+\Delta x) = f(x) + \Delta x \frac{\partial f(x)}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \cdots = \sum_{n=0}^{\infty} \frac{(\Delta x)^n}{n!}\frac{\partial^n f(x)}{\partial x^n}$$

where Δx is the separation between samples of f and is measured in pixel units

$$if\ \Delta x = 1 \rightarrow f(x+1) = f(x) + \frac{\partial f(x)}{\partial x} + \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \cdots = \sum_{n=0}^{\infty} \frac{1}{n!}\frac{\partial^n f(x)}{\partial x^n}$$

$$if\ \Delta x = -1 \rightarrow f(x-1) = f(x) - \frac{\partial f(x)}{\partial x} + \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2} + \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!}\frac{\partial^n f(x)}{\partial x^n}$$

Edge Detection

Edge detection

- According to the Taylor series, intensity differences can be computed using just a few terms
- For first-order derivatives, the difference can be formed in one of three ways

1. Forward difference

$$\frac{\partial f(x)}{\partial x} = f'(x) = f(x+1) - f(x)$$

2. Backward difference

$$\frac{\partial f(x)}{\partial x} = f'(x) = f(x) - f(x-1)$$

3. Central difference

$$\frac{\partial f(x)}{\partial x} = f'(x) = \frac{f(x+1) - f(x-1)}{2}$$

Edge Detection

Edge detection

- For the second order derivative

$$\frac{\partial^2 f(x)}{\partial^2 x} = f''(x) = f(x+1) - 2f(x) + f(x-1)$$

- For the third order derivative
  - Need the Taylor expansions for f(x+2) and f(x-2)

$$\frac{\partial^3 f(x)}{\partial^3 x} = f'''(x) = \frac{f(x+2) - 2f(x+1) + 0f(x) + 2f(x-1) - f(x-2)}{2}$$

Edge Detection

Edge detection

- The first four central derivatives

	$f(x+2)$	$f(x+1)$	$f(x)$	$f(x-1)$	$f(x-2)$
$2f'(x)$		1	0	–1	
$f''(x)$		1	–2	1	
$2f'''(x)$	1	–2	0	2	–1
$f''''(x)$	1	–4	6	–4	1

Edge Detection

Edge detection

- For two variables

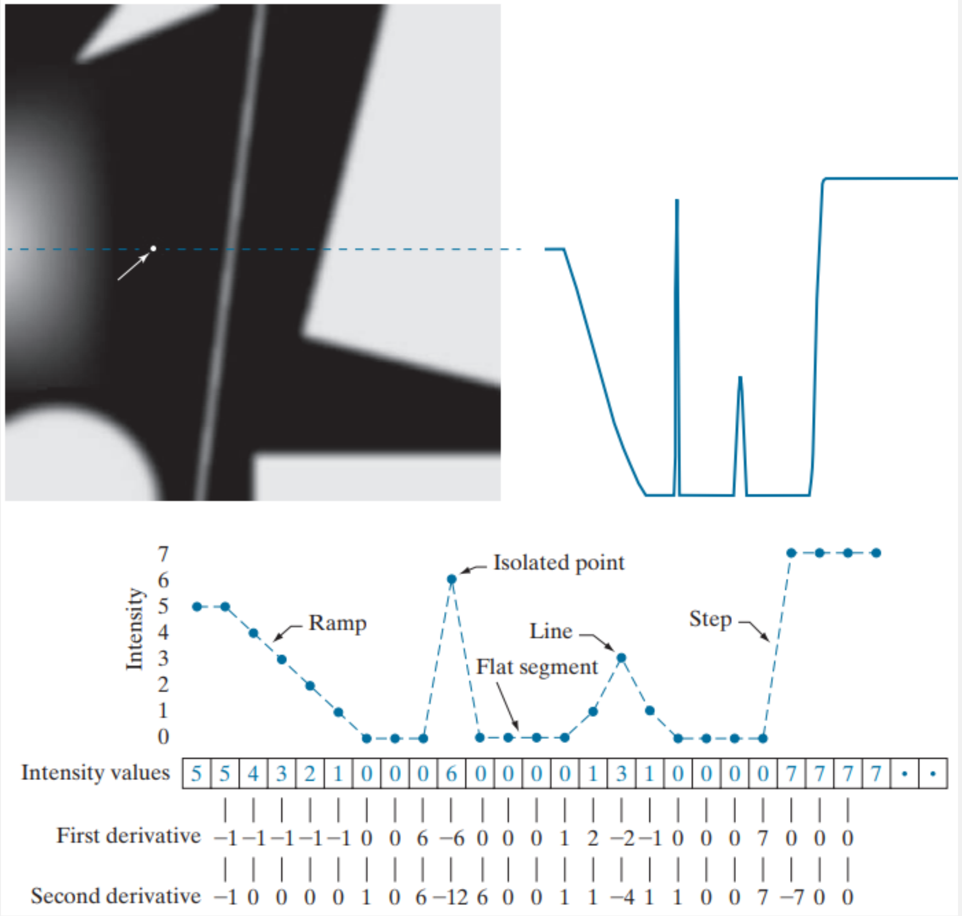
$$\frac{\partial^2 f(x,y)}{\partial^2 x} = f(x+1,y) - 2f(x,y) + f(x-1,y)$$

$$\frac{\partial^2 f(x,y)}{\partial^2 y} = f(x,y+1) - 2f(x,y) + f(x,y-1)$$

Edge Detection

Edge detection

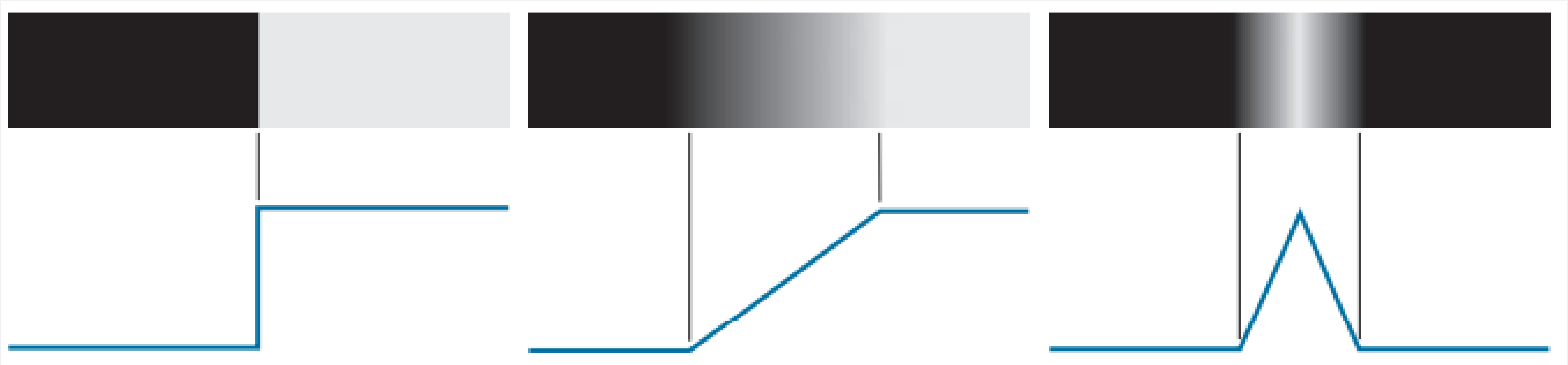
- Conclusions
  1. First-order derivatives generally produce thicker edges
  2. Second-order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise
  3. Second-order derivatives produce a double-edge response at ramp and step transitions in intensity
  4. The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light



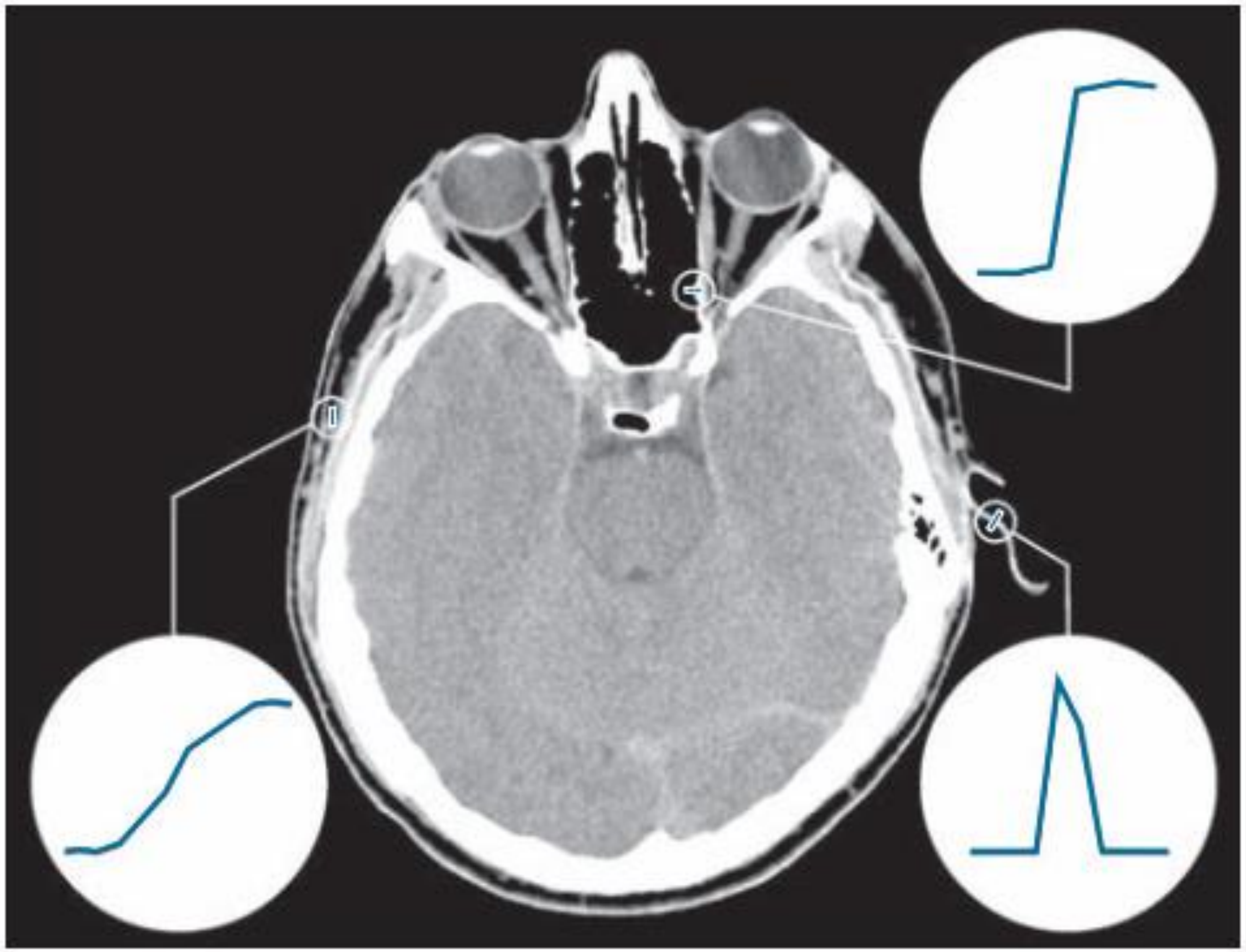
Edge Detection

Edge models are classified according to their intensity profiles

- The step edge is a characterized by a transition between two intensity levels occurring ideally over the distance of one pixel



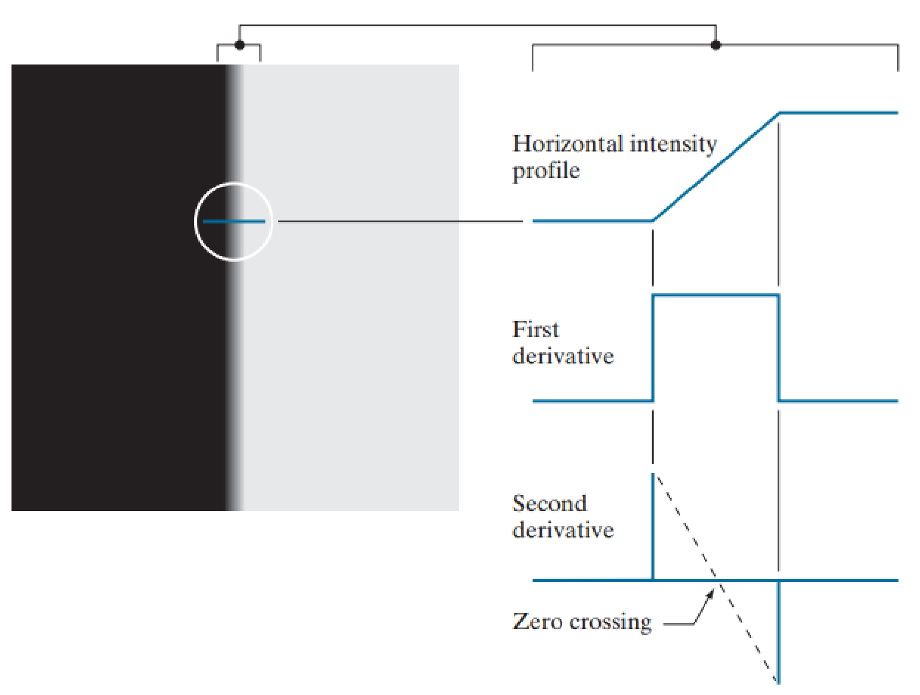
Edge Detection



Edge Detection

Edge detection

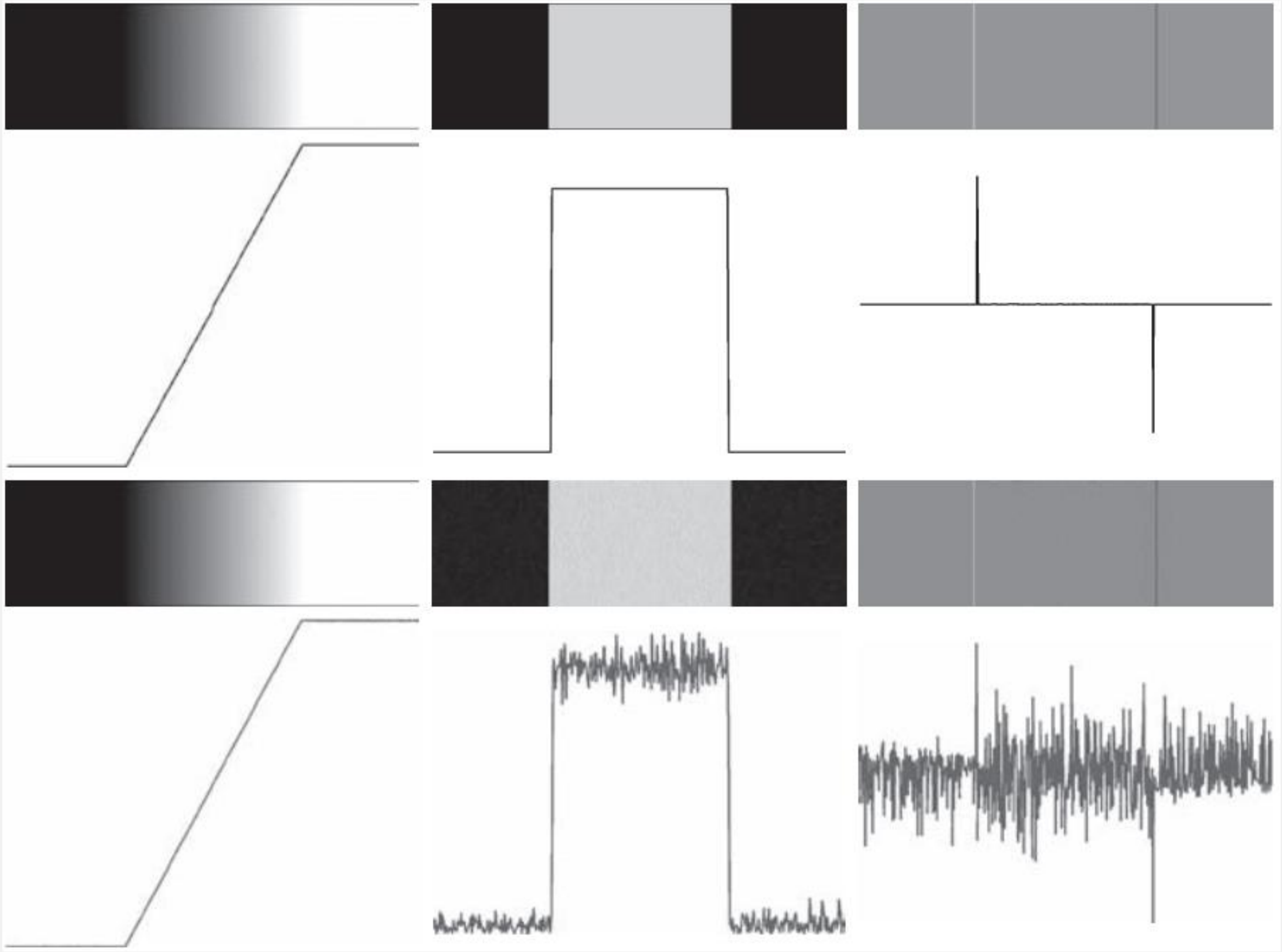
- Conclusions
  1. The magnitude of the first-order derivative can be used to detect the presence of an edge at a point in an image
  2. The sign of the second derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge
  3. Two additional properties of the second derivative around an edge:
     a) It produces two values for every edge in an image
     b) Its zero crossings can be used for locating the centers of thick edges



Edge Detection

Edge detection

- In noisy image
  1. Image smoothing for noise reduction
  2. Detection of edge points
  3. Edge localization



Edge Detection

Basic edge detection

- Image gradient and properties

vector

Gradient vector

$$\nabla f(x,y) \equiv grad[f(x,y)] \equiv \begin{bmatrix} g_x(x,y) \\ g_y(x,y) \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f(x,y)}{\partial x} \\ \dfrac{\partial f(x,y)}{\partial y} \end{bmatrix}$$

Magnitude

$$M(x,y) = ||\nabla f(x,y)|| = \sqrt{g_x^2(x,y) + g_y^2(x,y)}$$

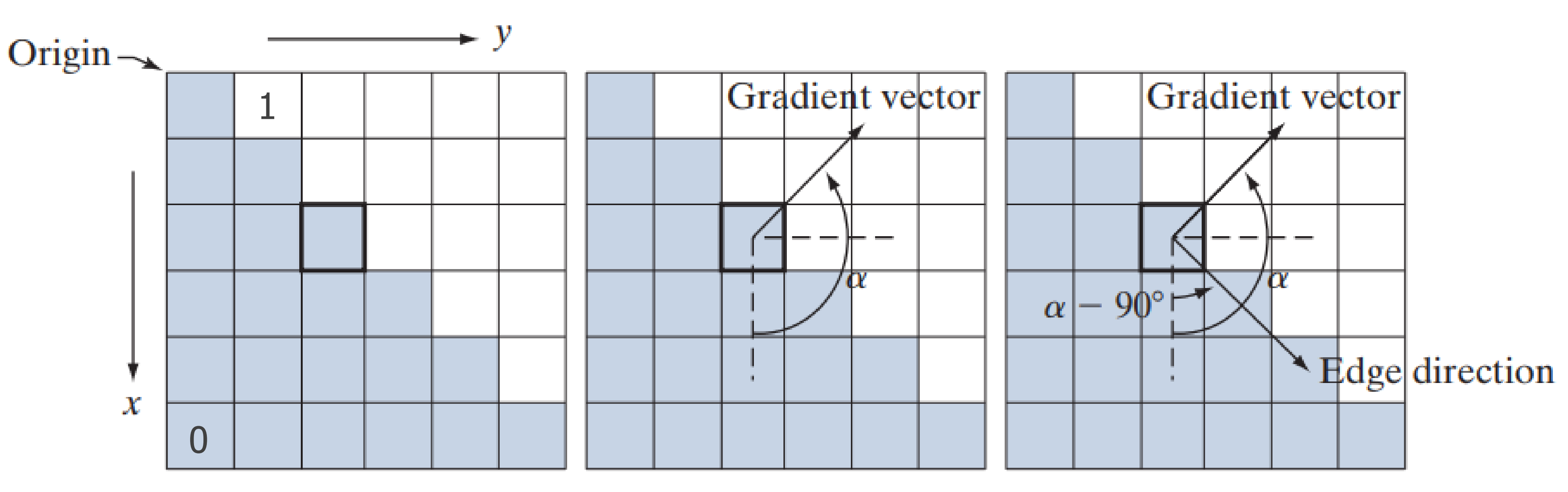
Direction of gradient vector

$$\alpha(x,y) = tan^{-1}\left[\frac{\dfrac{\partial f(x,y)}{\partial y}}{\dfrac{\partial f(x,y)}{\partial x}}\right]$$

Edge Detection

Basic edge detection

➢ Image gradient and properties



Edge Detection

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Basic edge detection

- Gradient operators in 1-D kernels

$$g_x(x,y) = \frac{\partial f(x,y)}{\partial x} = \boxed{f(x+1,y) - f(x,y)} = z_8 - z_5$$

$$g_y(x,y) = \frac{\partial f(x,y)}{\partial y} = \boxed{f(x,y+1) - f(x,y)} = z_6 - z_5$$

-1
1

-1	1

Edge Detection

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Basic edge detection

- Gradient operators in 2-D kernels

Roberts

$$g_x(x,y) = z_9 - z_5$$

$$g_y(x,y) = z_8 - z_6$$

-1	0
0	1

0	-1
1	0

Prewitt

$$g_x(x,y) = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y(x,y) = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Sobel

$$g_x(x,y) = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y(x,y) = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Edge Detection

Edge detection

- Prewitt

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

- Sobel

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Edge Detection

Basic edge detection

- The results of Sobel kernel

Edge Detection

Laplacian detection

- It is a second-order differential operator for edge detection
- It is very sensitivity to noise
- Processes
  - Noise reduction (low-pass spatial frequency filtering)
  - Laplacian mask

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial^2 x} + \frac{\partial^2 f(x,y)}{\partial^2 y} = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Laplacian mask

1	1	1
1	-8	1
1	1	1

Edge Detection

Marr-Hildreth edge detection [paper]

- They suggest that an operator used for edge detection have two salient features
  1. It should be a differential operator capable of computing a digital approximation of the first or second derivative at every point in the image
  2. It should be capable of being “tuned” to act at any desired scale, so that large operators can be used to detect blurry edges and small operators to detect sharply focused fine detail

$$G(x,y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

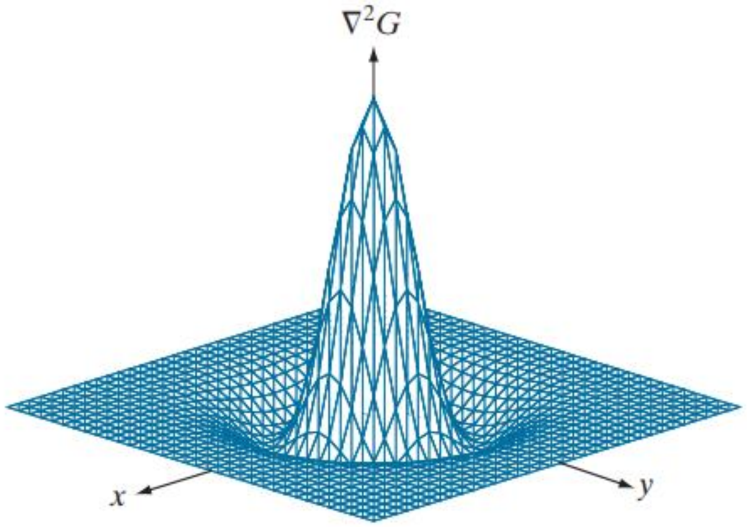
$$\begin{aligned}
\nabla^2 G(x,y) &= \frac{\partial^2 G(x,y)}{\partial x^2} + \frac{\partial^2 G(x,y)}{\partial y^2} \\
&= \frac{\partial}{\partial x}\left(\frac{-x}{\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}\right) + \frac{\partial}{\partial y}\left(\frac{-y}{\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}\right) \\
&= \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right) e^{\frac{-(x^2+y^2)}{2\sigma^2}} + \left(\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right) e^{\frac{-(x^2+y^2)}{2\sigma^2}} \\
&= \left(\frac{x^2 + y^2 - 2\partial^2}{\sigma^4}\right) e^{\frac{-(x^2+y^2)}{2\sigma^2}}
\end{aligned}$$

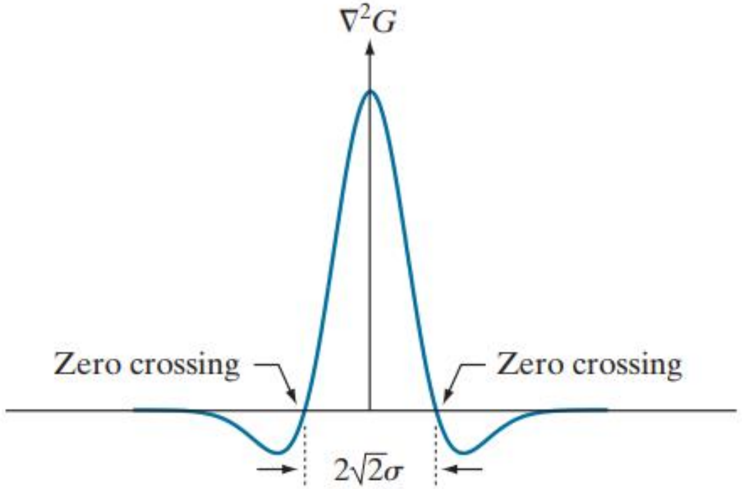
Laplacian of a Gaussian (LoG)

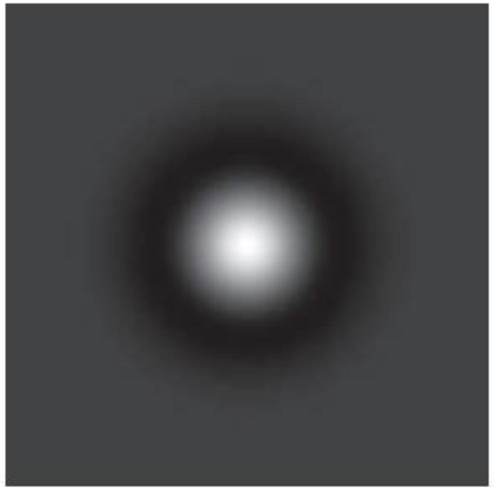
Edge Detection

LoG

Mexican hat operator







0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

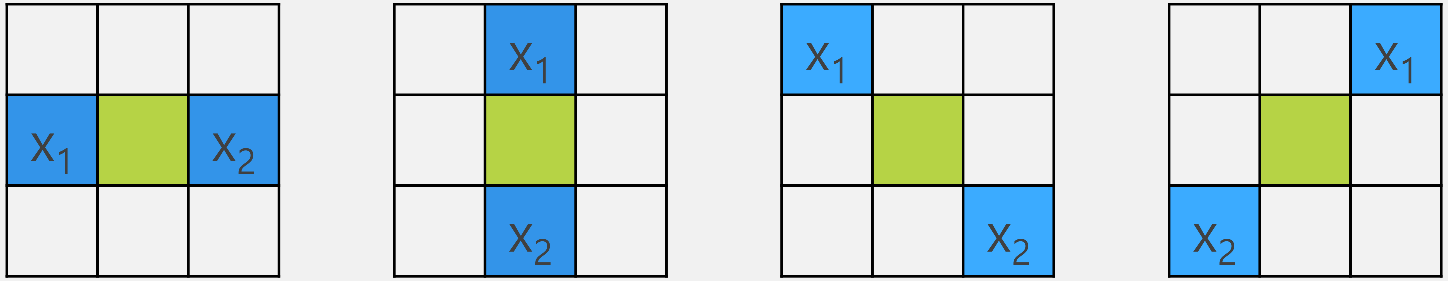
Edge Detection

Marr-Hildreth edge detection summary

$$g(x, y) = [\nabla^2 G(x, y)] \star f(x, y) = \nabla^2 [G(x, y) \star f(x, y)]$$

1	1	1
1	-8	1
1	1	1

1. Filter the input image with an n×n Gaussian kernel by $G(x, y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}$
2. Compute the Laplacian of the image resulting from step 1 using the 3×3 kernel
3. Find the zero crossings of the image from step 2
   - Using a 3×3 neighborhood centered at point p -> a zero crossing at p implies that the signs of at least two of its opposing neighboring pixels must differ
   - For cases to test: left/right, up/down, and the two diagonals
     - ✓ If the values of g(x, y) are being compared against a threshold (a common approach), then not only must the signs of opposing neighbors be different, but the absolute value of their numerical difference must also exceed the threshold before we can call p a zero-crossing pixel

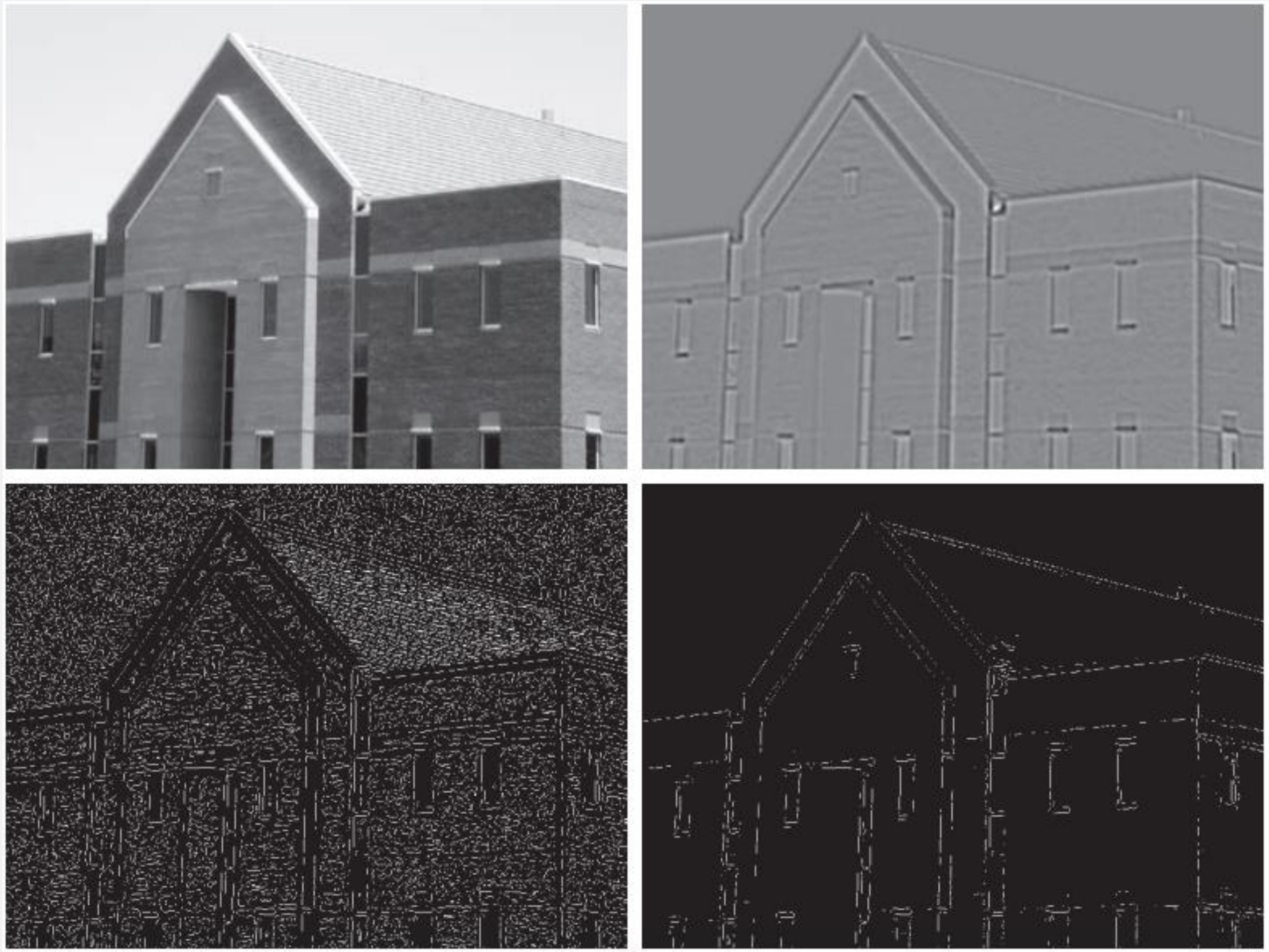


Edge Detection

Marr-Hildreth edge detection

Intensity values scaled to the range [0, 1]

Results of step1 and step2 using σ=4 and n=25



Zero crossings using threshold=0

threshold=4% of the maximum value

Edge Detection

Canny detection

- It is proposed by John F. Canny in 1986 to get rid of mask template for edge detection [paper]
- Three objectives
  1. Low error rate
  2. Edge points should be well localized
  3. Single edge point response
- How to achieve?
  - Smooth the image (Using Gaussian filter is a good method)
  - Compute gradient magnitude and angle
  - Preserve edge points
  - Edge determination

Edge Detection

Canny detection

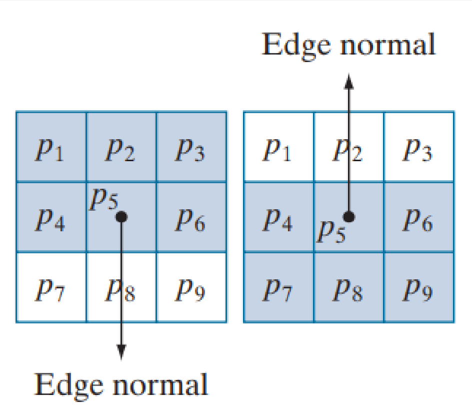
- Smooth the image (Gaussian filter)

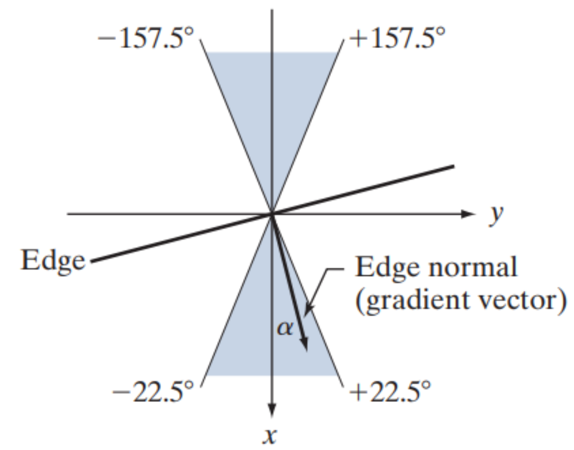
$$f_s(x,y) = G(x,y) \star f(x,y) \qquad G(x,y) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

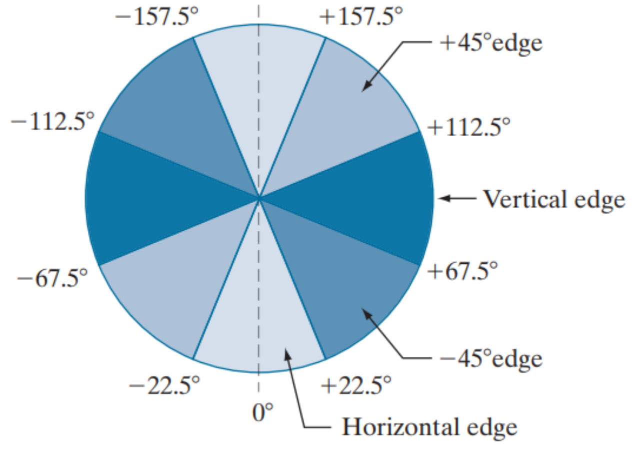
- Compute gradient magnitude and angle

$$M_s = ||\nabla f_s(x,y)|| = \sqrt{g_x^2(x,y) + g_y^2(x,y)} \qquad \alpha(x,y) = tan^{-1}\left[\frac{g_y(x,y)}{g_x(x,y)}\right]$$

- Preserve edge points
  - Non-maxima suppression



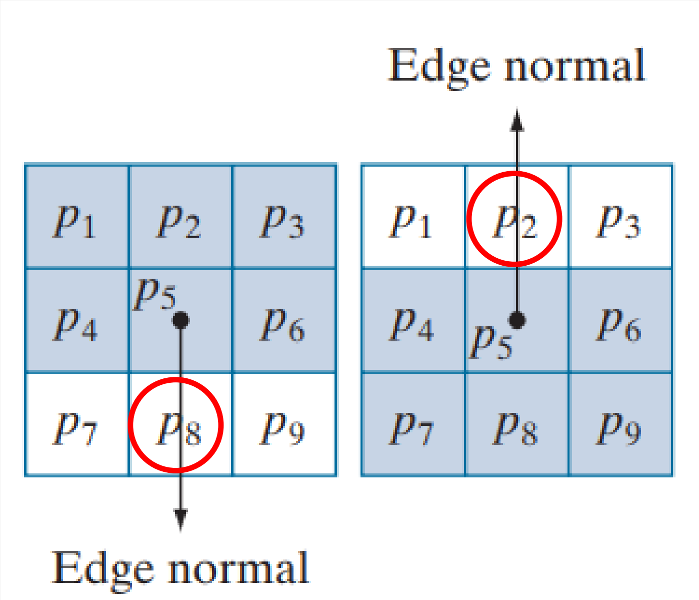


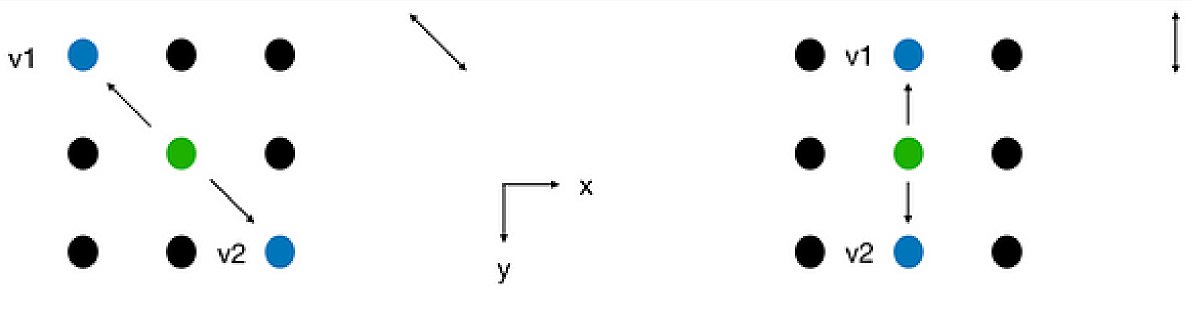


Edge Detection

Canny detection

- Preserve edge points
  - Formulate the non-maxima suppression for a 3×3 region centered at point (x, y) in α
    1. Let d_1, d_2, d_3, and d_4 denote the four basic edge directions: horizontal, -45°, vertical, and +45°
    2. Find the direction d_k that is closest to $\alpha(x, y)$
    3. K denotes the value of $||\nabla f_s||$ at (x, y), and if K is less than the value of $||\nabla f_s||$ at one or both of the neighbors of point (x, y) along d_k, let $g_N(x, y) = 0$ (suppression); otherwise, $g_N(x, y) = K$



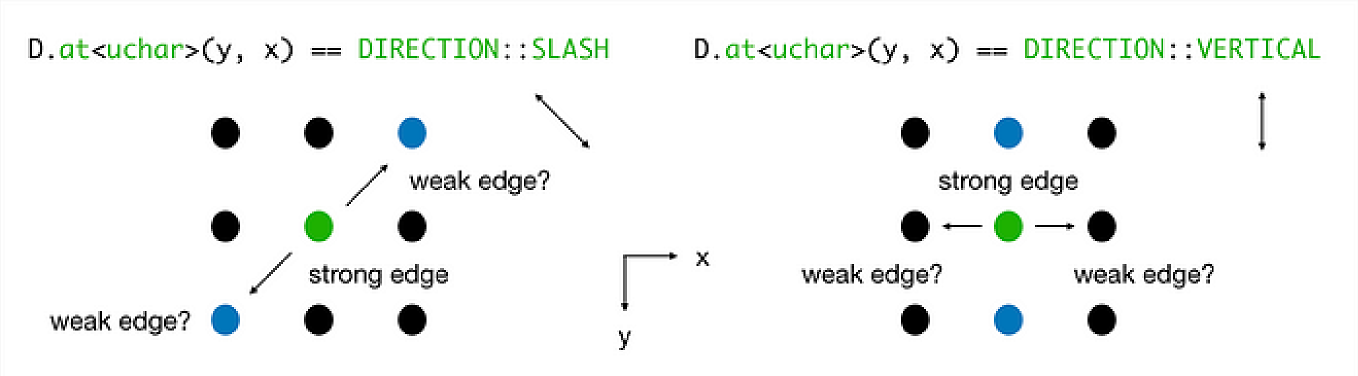


Link

Edge Detection

Canny detection

- Edge determination
  - Using two thresholds: T_H and T_L (Experimental evidence suggests $T_H : T_L$ in the range of 2:1 to 3:1)
    1. Define the strong edge pixels $g_{NH}(x, y) = g_N(x, y) \geq T_H$
    2. Define the weak edge pixels $g_{NL}(x, y) = g_N(x, y) \geq T_L$
    3. Eliminate from $g_{NL}(x, y)$ all the nonzero pixels from $g_{NH}(x, y)$: $g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y)$
    4. Longer edge are formed using
       a) Locate the next unvisited edge pixel, p, in $g_{NH}(x, y)$
       b) Mark as valid edge pixels all the weak pixels in $g_{NL}(x, y)$ that connected to p using 8-connectivity
       c) If any nonzero pixel in $g_{NH}(x, y)$ has not been visited, return to step a)
       d) Set to zero all pixels in $g_{NL}(x, y)$ that are not marked as valid edge pixels



Link

Edge Detection

Canny edge detection

Intensity values scaled to the range [0, 1]

Thresholded gradient of the smoothed image

Using Marr-Hildreth

Using Canny

02

Morphology

Morphology

Morphology in image processing is a technology based on shape processing

- It performs a series of nonlinear operations by structuring elements
  - Dilation and erosion
  - Opening and closing
  - Morphological hit-or-miss transform
  - Some basic morphological algorithms
  - Grayscale morphology
- Application
  - Noise reduction
  - Edge detection
  - Object segmentation
  - Skeletonization

Morphology

Set theory

- Complement

$$A^c = \{\omega | \omega \notin A\}$$

- Difference

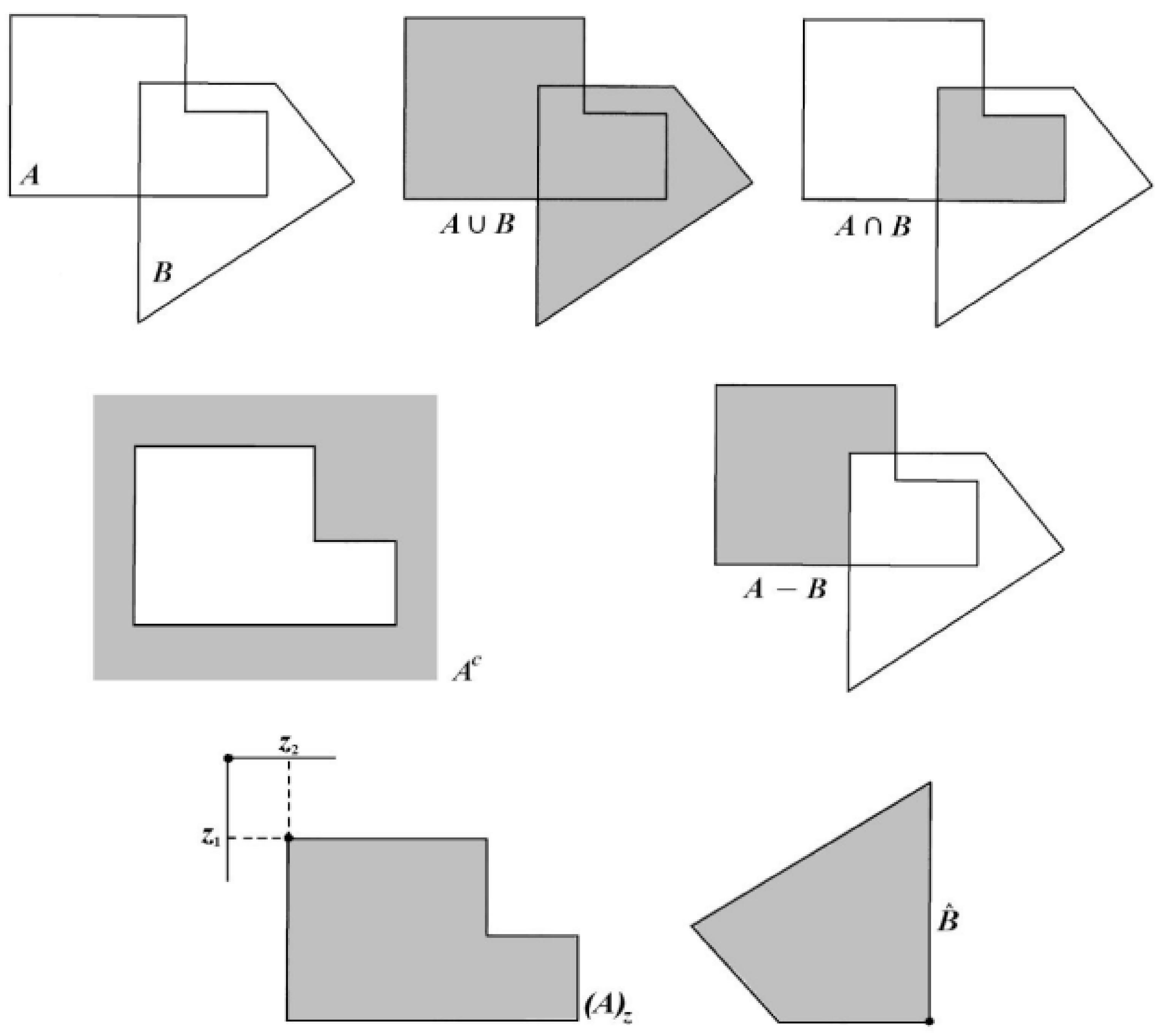
$$A - B = \{\omega | \omega \in A \; and \; \omega \notin B\}$$

- Translation

$$A_z = \{c | c = a + z, for \; a \in A\}$$

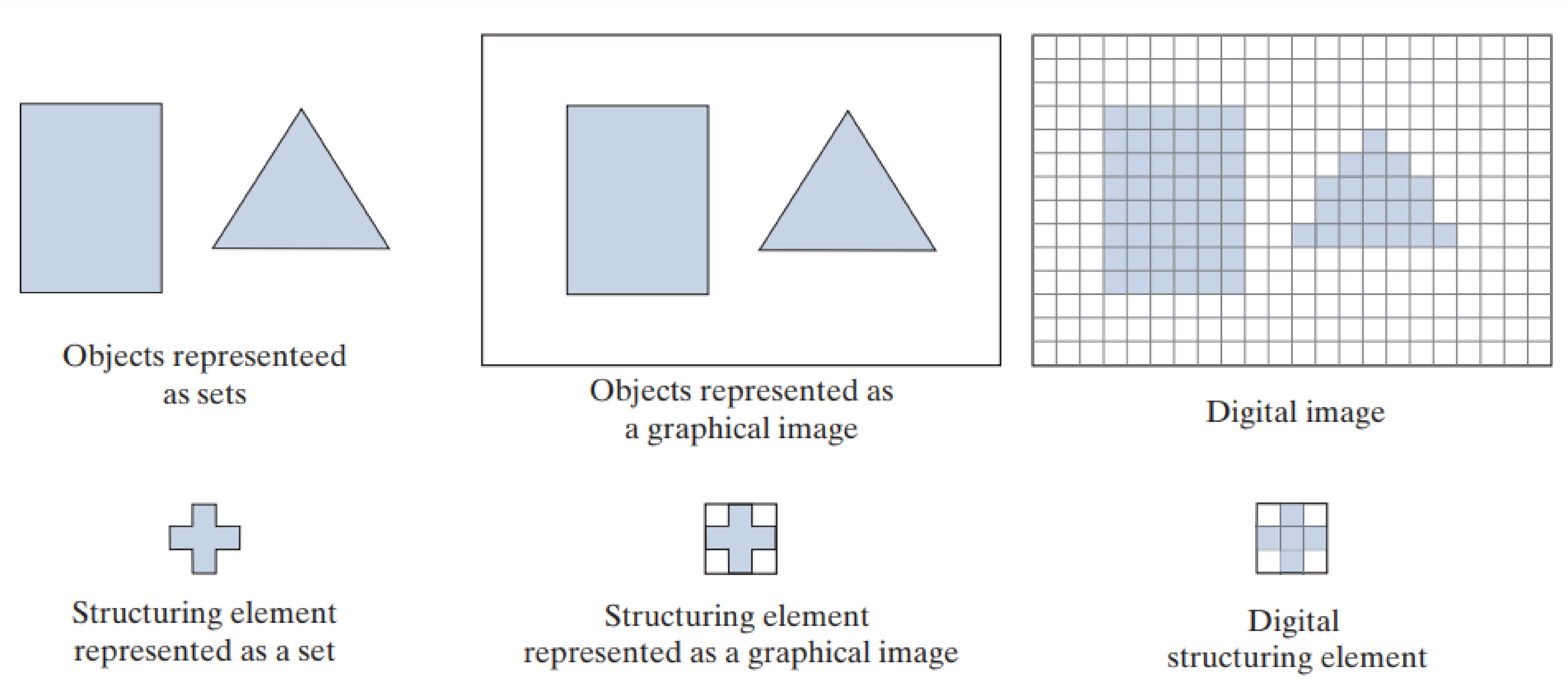
- Reflection

$$\hat{B} = \{\omega | \omega = -b, b \in B\}$$



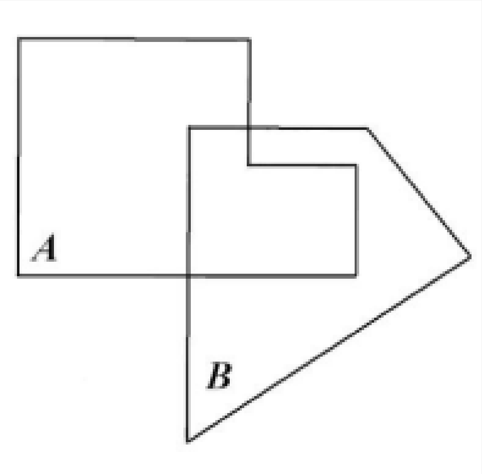
Morphology

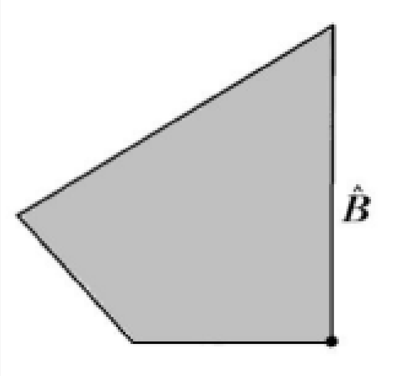
Set theory

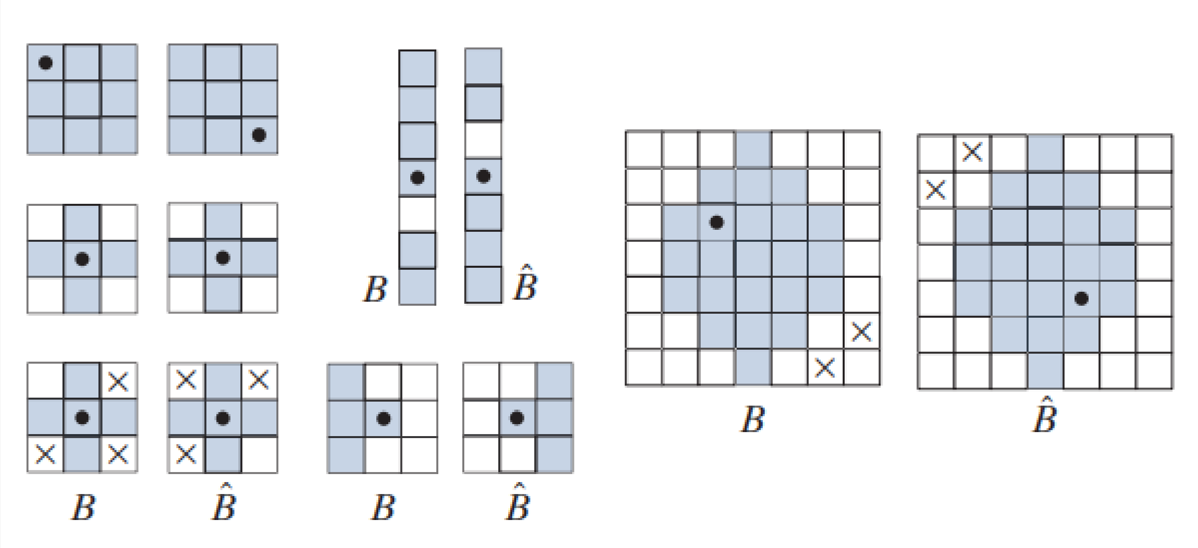


Morphology

Set theory







Morphology

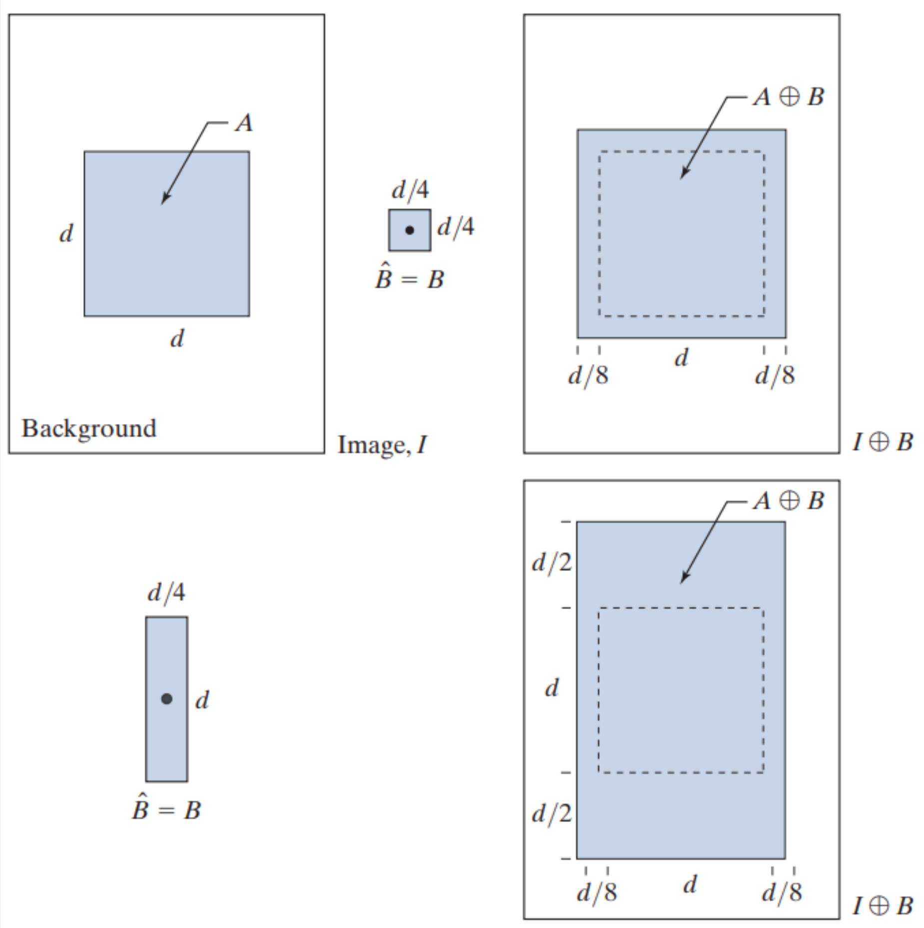
Dilation and erosion are two basic operations in morphology

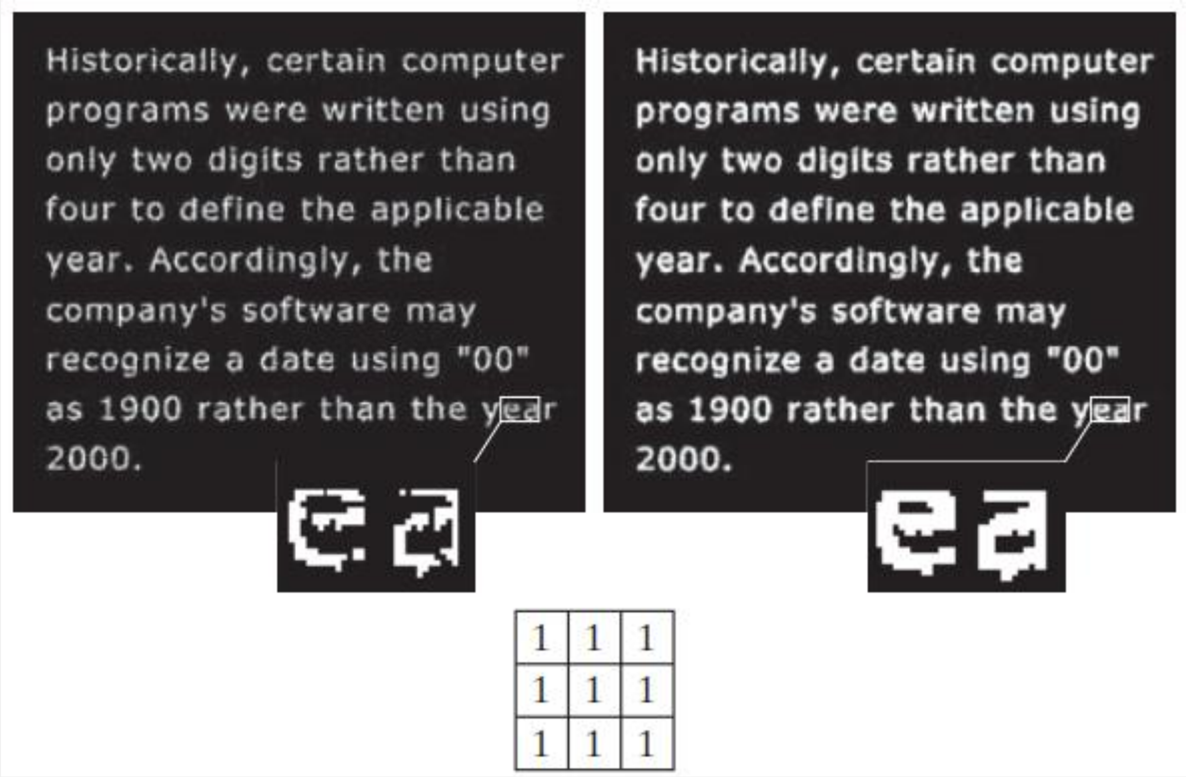
- Dilation
  - It is an operation that expands objects (white areas) in an image and commonly used to fill in small holes, connect broken parts, and enhance features in the image
  - Principle
    - ✓ Structuring elements
      - Define the shape and size of the pixel neighborhood
      - Be typically a rectangular, circular, or cross-shaped matrix
    - ✓ Operation
      - The center of the structuring element is aligned with each pixel in the image
      - If any part of the structuring element overlaps with a white pixel in the image, the corresponding pixel in the output image is set to white
    - ✓ Mathematical representation
      - $A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\} = \{z | [(\hat{B})_z \cap A] \subseteq A\}$

Morphology

Dilation and erosion are two basic operations in morphology

- Dilation





Morphology

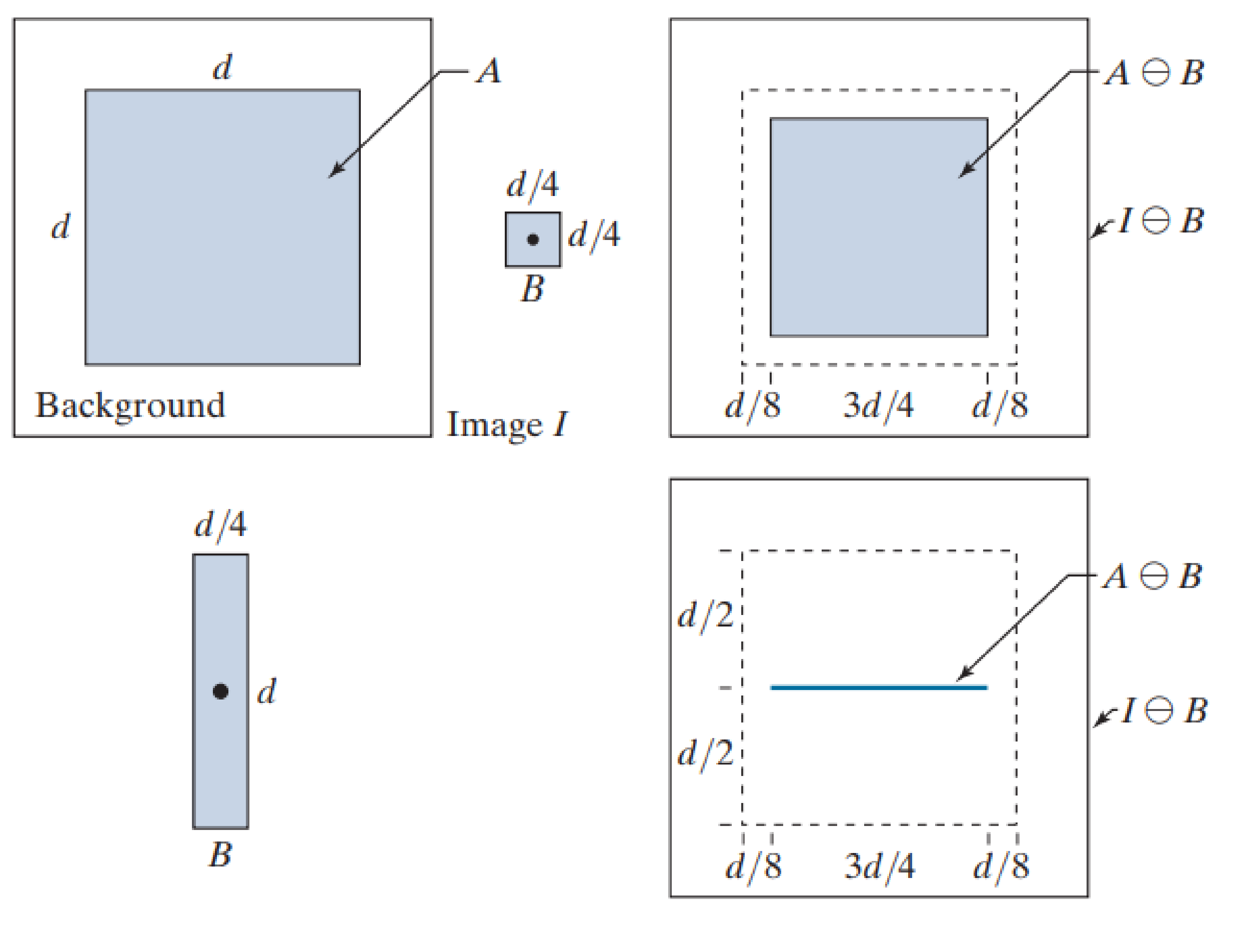
Dilation and erosion are two basic operations in morphology

- Erosion
  - It is an operation that shrinks objects (white areas) in an image and commonly used to remove small objects, separate connected objects, and thin object boundaries
  - Principle
    - ✓ Structuring elements
      - Define the shape and size of the pixel neighborhood
    - ✓ Operation
      - The center of the structuring element is aligned with each pixel in the image
      - If all part of the structuring element overlaps with white pixels in the image, the corresponding pixel in the output image is retained as white
    - ✓ Mathematical representation
      - $A \ominus B = \{z | (B)_z \subseteq A\} = \{z | (B)_z \cap A^c = \emptyset\}$

Morphology

Dilation and erosion are two basic operations in morphology

- Erosion



Morphology

Duality

- Dilation and erosion are duals of each other with respect to set complementation and reflection

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

  - Prove (if the structuring element values are symmetric with respect to its origin -> $\hat{B} = B$)

$$(A \ominus B)^c = \{z|(B)_z \subseteq A\}^c = \{z|(B)_z \cap A^c = \emptyset\}^c = \{z|(B)_z \cap A^c \neq \emptyset\} = A^c \oplus \hat{B}$$

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

Morphology

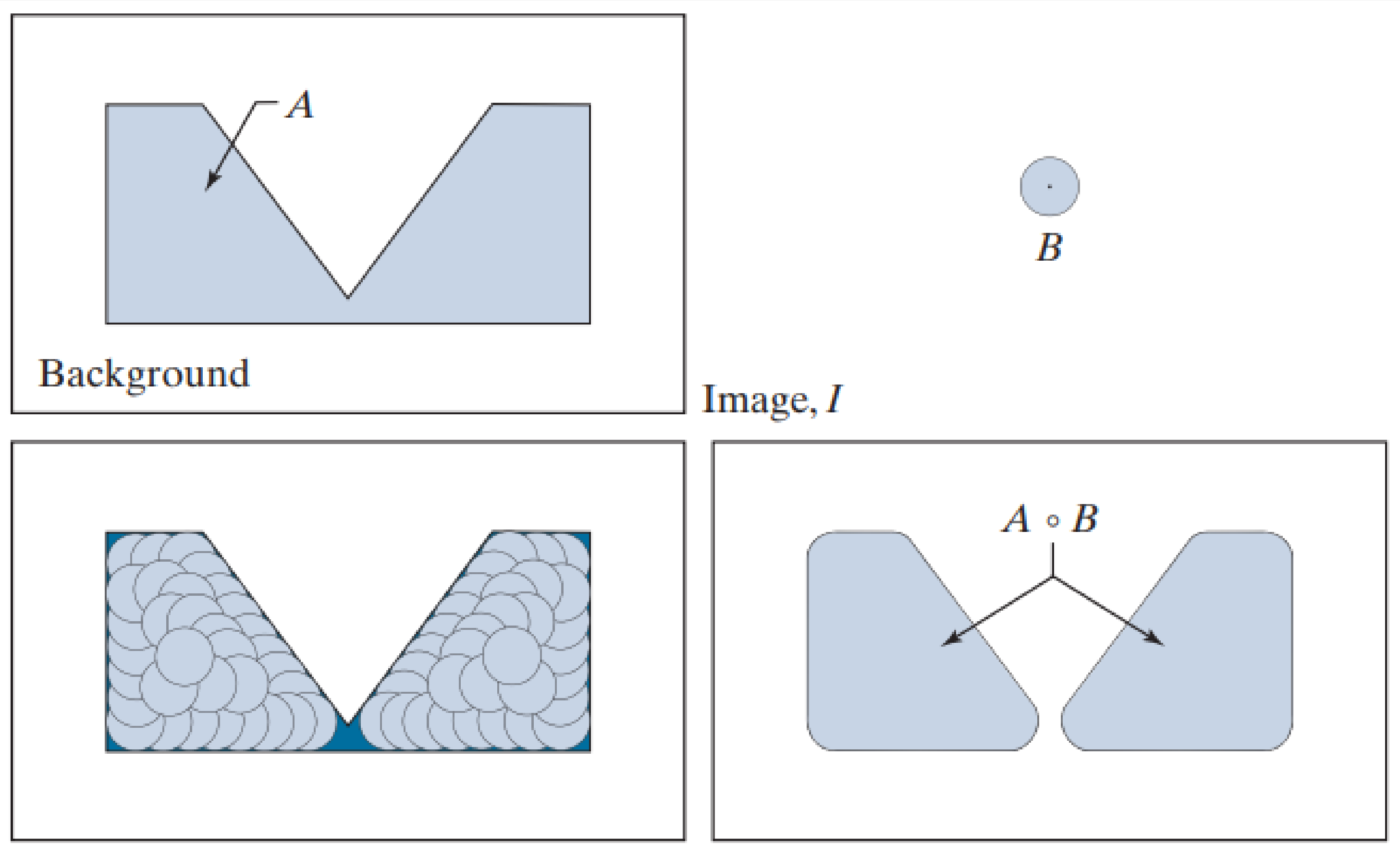
Opening and closing are fundamental operations in morphological image processing defined using dilation and erosion

- Opening
  - It is an operation consisting of an erosion followed by a dilation, using the same structuring element for both operations
  - To remove small objects or noise while preserving the shape and size of larger objects
  - To smooth the boundaries of objects and separate close but distinct objects
  - Principle
    - ✓ Erosion
    - ✓ Dilation
    - ✓ Mathematical representation
      - $A \circ B = (A \ominus B) \oplus B = \cup \{(B)_z | (B)_z \subseteq A\}$

Morphology

Opening

- $A \circ B = (A \ominus B) \oplus B = \cup \{(B)_z | (B)_z \subseteq A\}$



Morphology

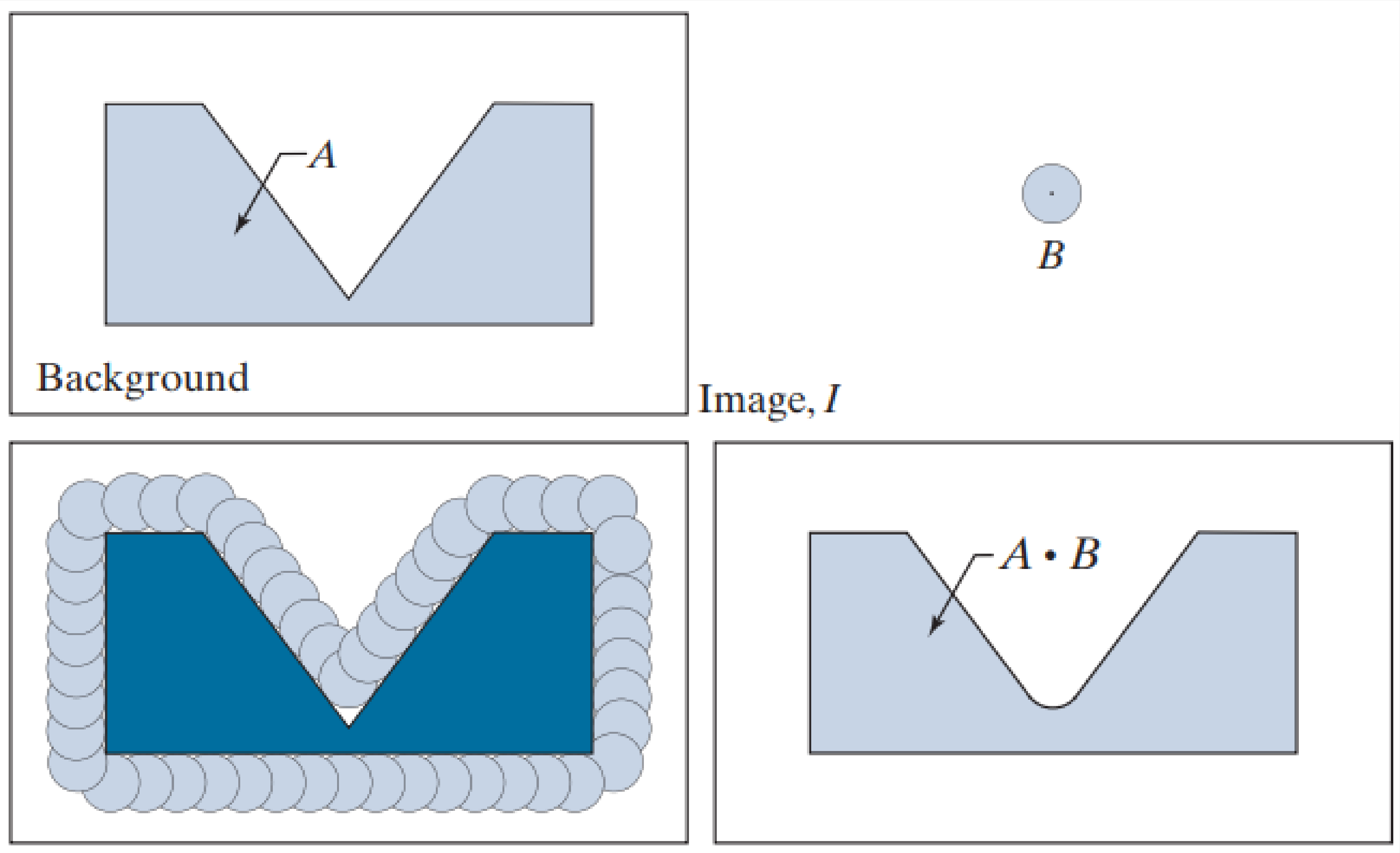
Opening and closing are fundamental operations in morphological image processing defined using dilation and erosion

- Closing
  - It is an operation consisting of a dilation followed by an erosion, using the same structuring element for both operations
  - To fill small holes and in objects
  - To smooth the boundaries of objects and connect adjacent objects
  - Principle
    - ✓ Dilation
    - ✓ Erosion
    - ✓ Mathematical representation
      - $A \bullet B = (A \oplus B) \ominus B = [\cup \{(B)_z | (B)_z \cap A = \emptyset\}]^c$

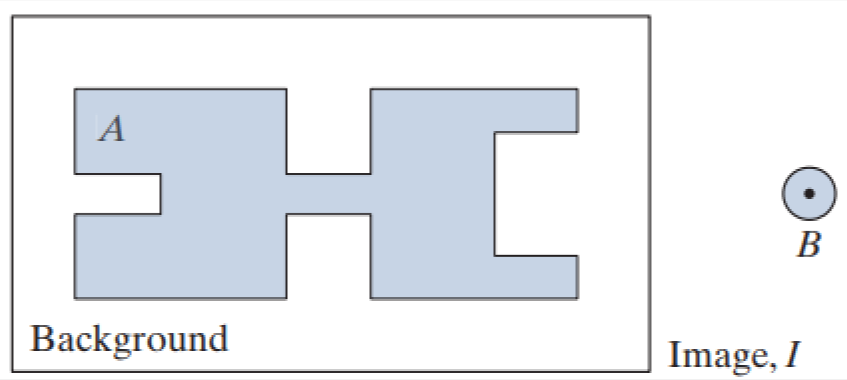
Morphology

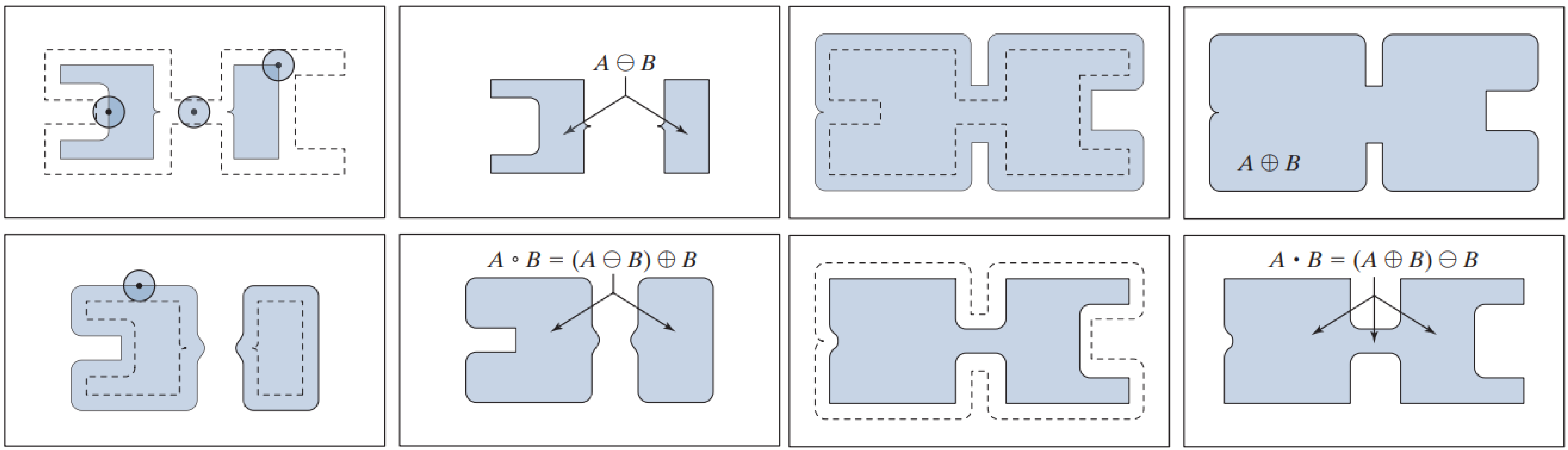
Closing

- $A \bullet B = (A \oplus B) \ominus B = [\cup \{(B)_z | (B)_z \cap A = \emptyset\}]^c$



Morphology





Morphology

Properties

- Opening
  a) $A \circ B$ is a subset 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 of $A \bullet B$
  b) If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$
  c) $(A \bullet B) \bullet B = A \bullet B$
- Note from condition (c) in both cases that multiple openings or closings of a set have no effect after the operation has bee applied once

Morphology

Duality

- Opening and closing are duals of each other with respect to set complementation and reflection

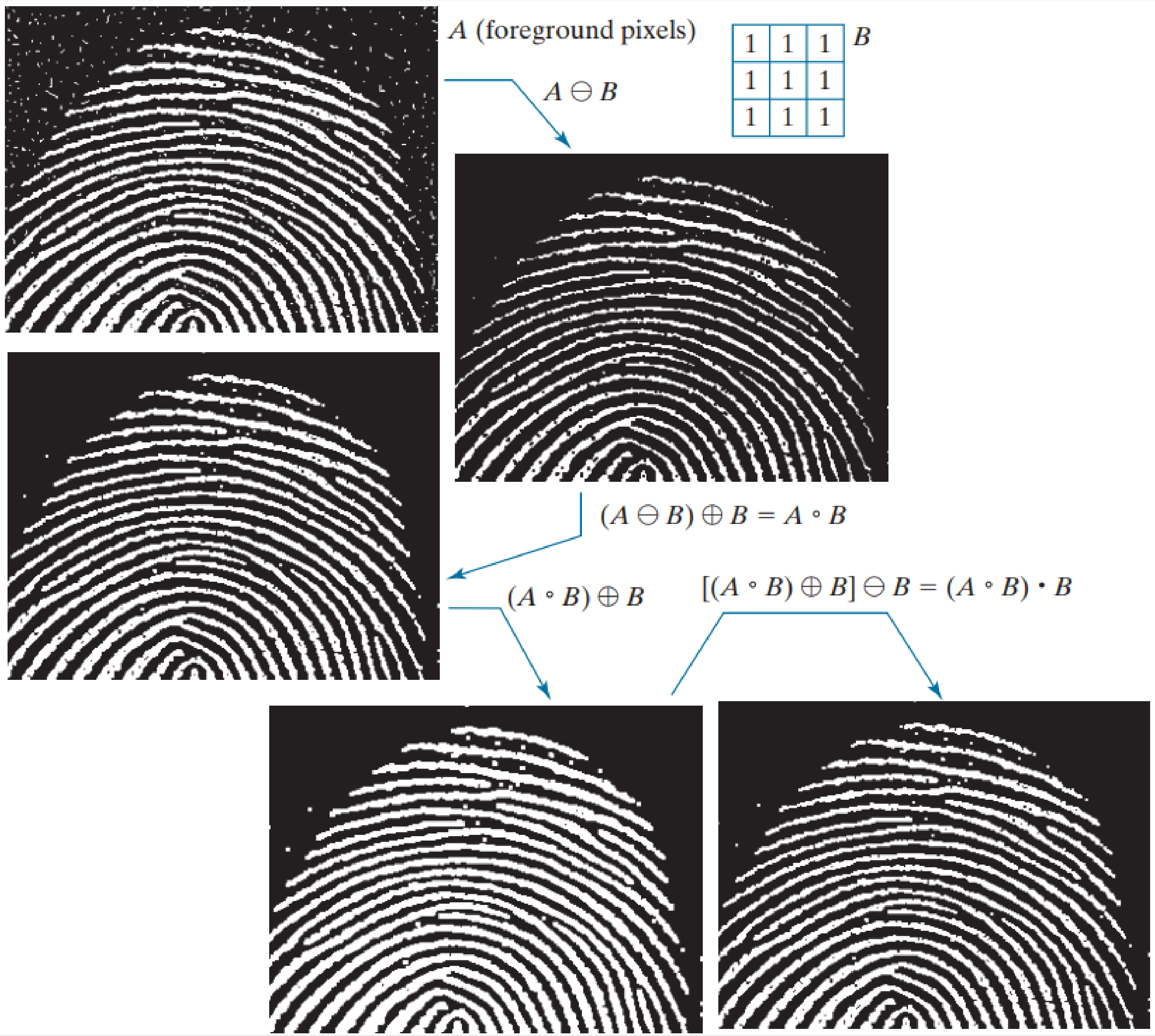
$$(A \circ B)^c = A^c \bullet \hat{B} \qquad (A \bullet B)^c = A^c \circ \hat{B}$$

  - Prove

$$(A \circ B)^c = [(A \ominus B) \oplus B]^c = [(A \ominus B)^c \ominus \hat{B}] = [(A^c \oplus \hat{B}) \ominus \hat{B}] = A^c \bullet \hat{B}$$

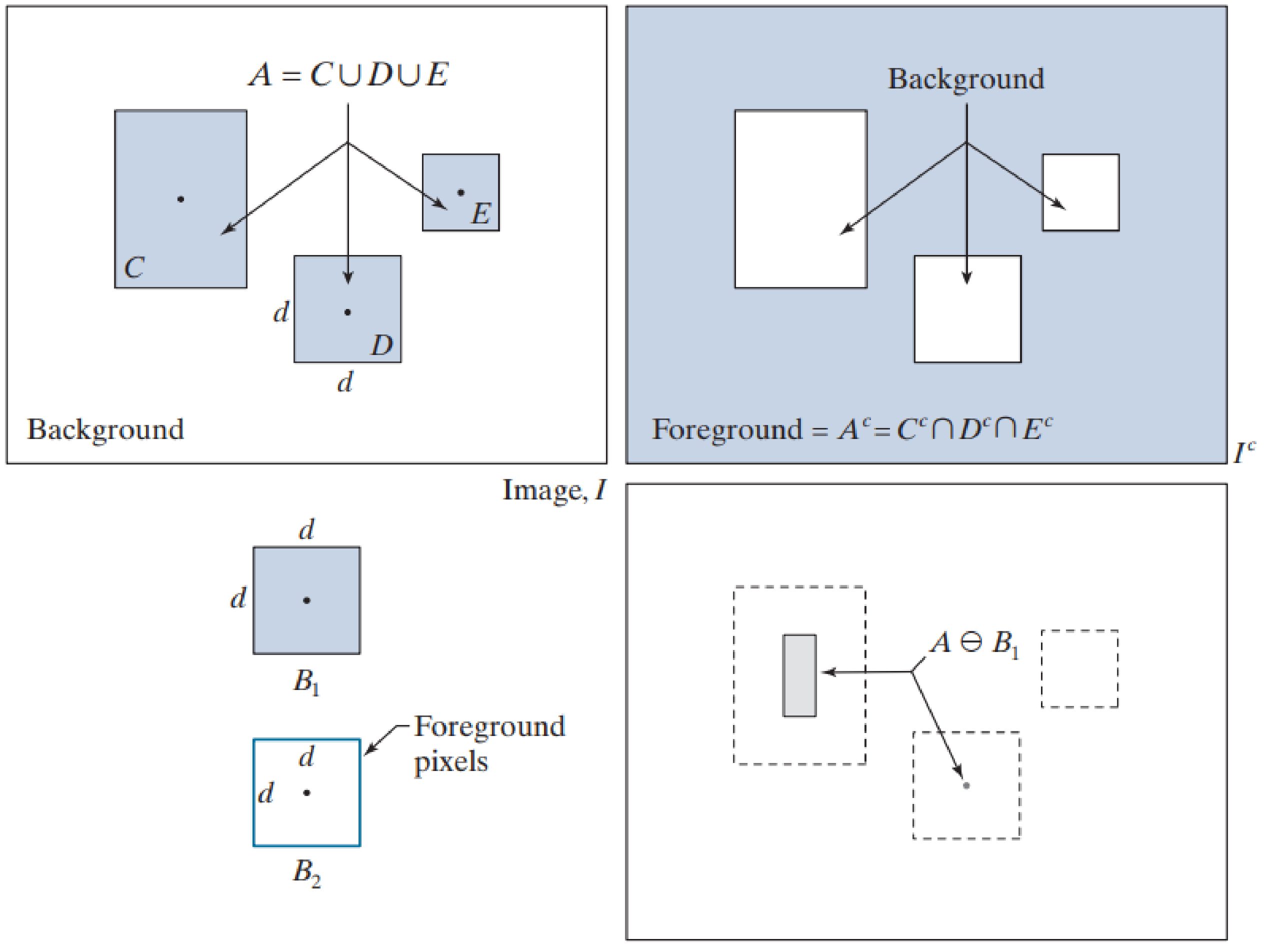
$(A \bullet B)^c$ → $A^c \circ \hat{B}$

Morphology

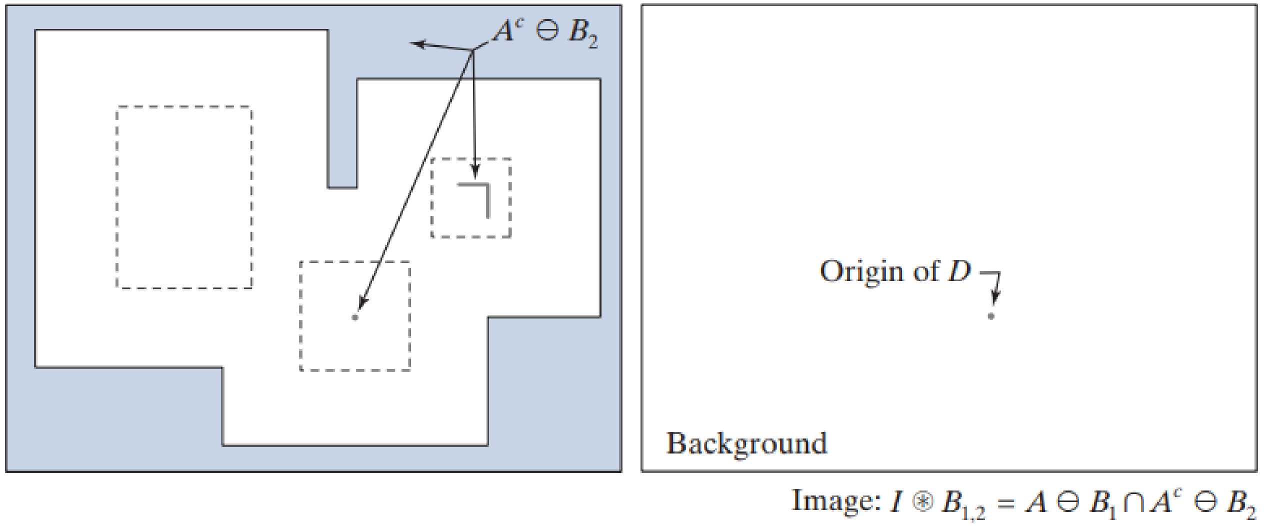


Morphology

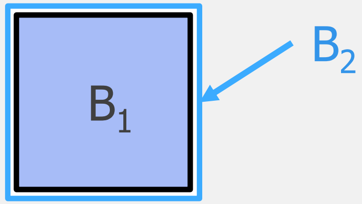
Morphological hit-or-miss transform (HMT) is a basic tool for shape detection



$$I \circledast B_{1,2} = \{z|(B_1)_z \subseteq A \text{ and } (B_2)_z \subseteq A^c\} = (A \ominus B_1) \cap (A^c \ominus B_2)$$



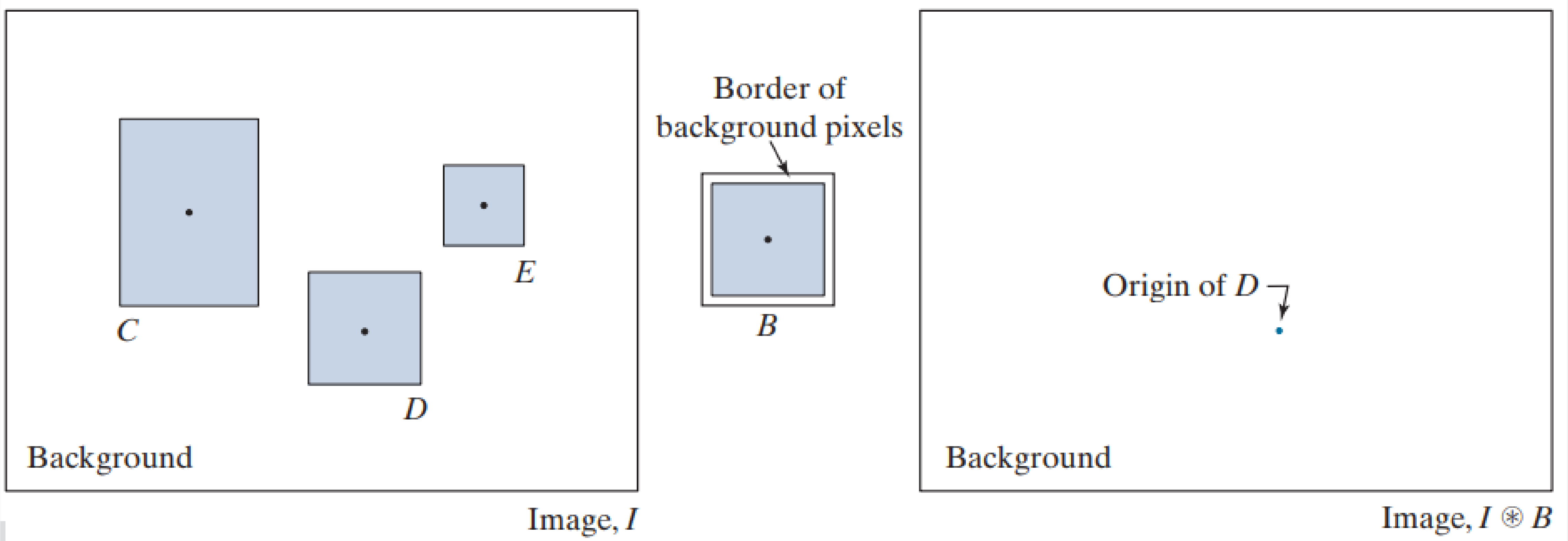
$B=\{B_1, B_2\}$



Morphology

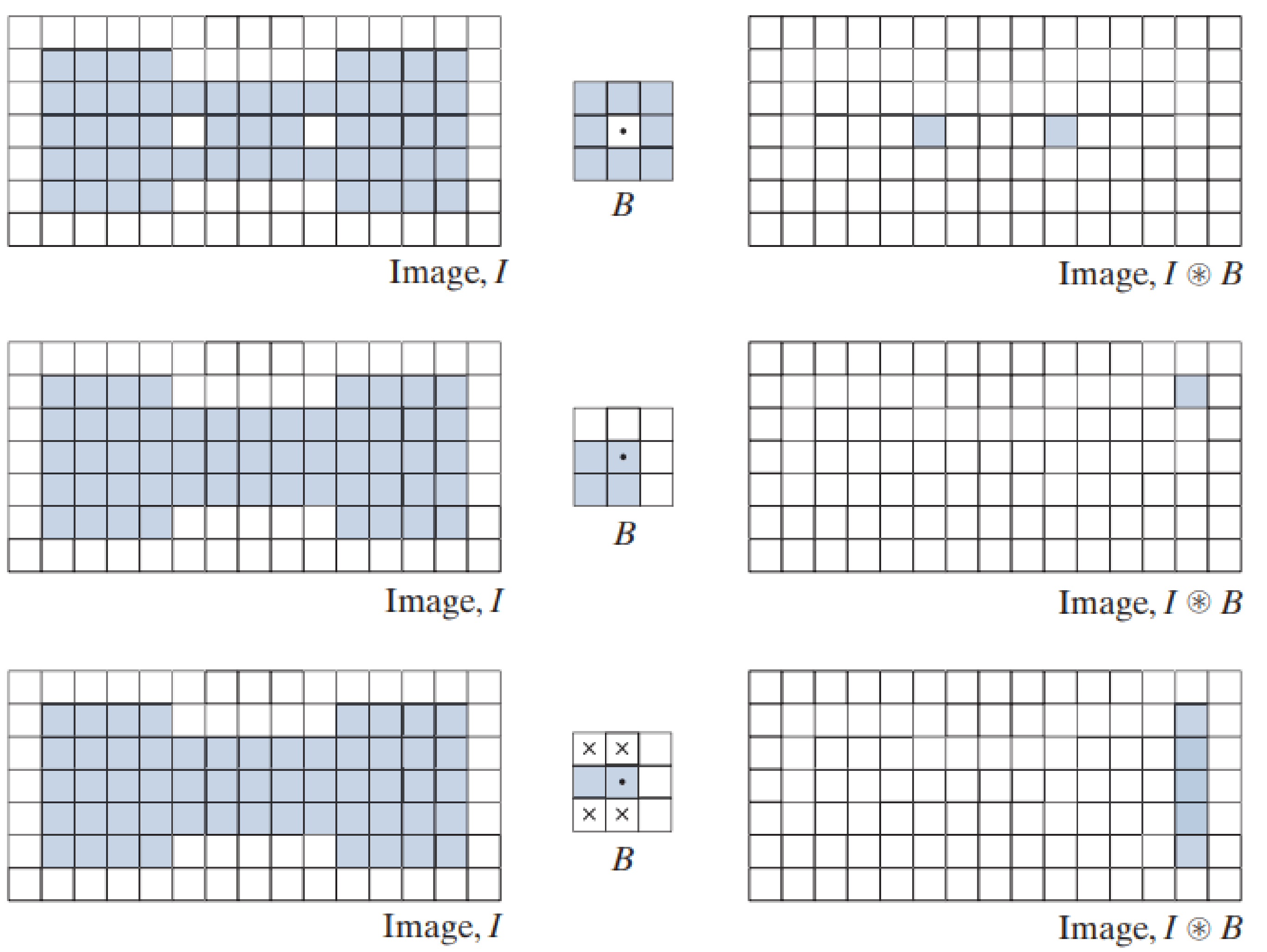
Morphological hit-or-miss transform (HMT) is a basic tool for shape detection

$$I \circledast B = \{z|(B)_z \subseteq I\}$$



Morphology

Some basic morphological algorithm

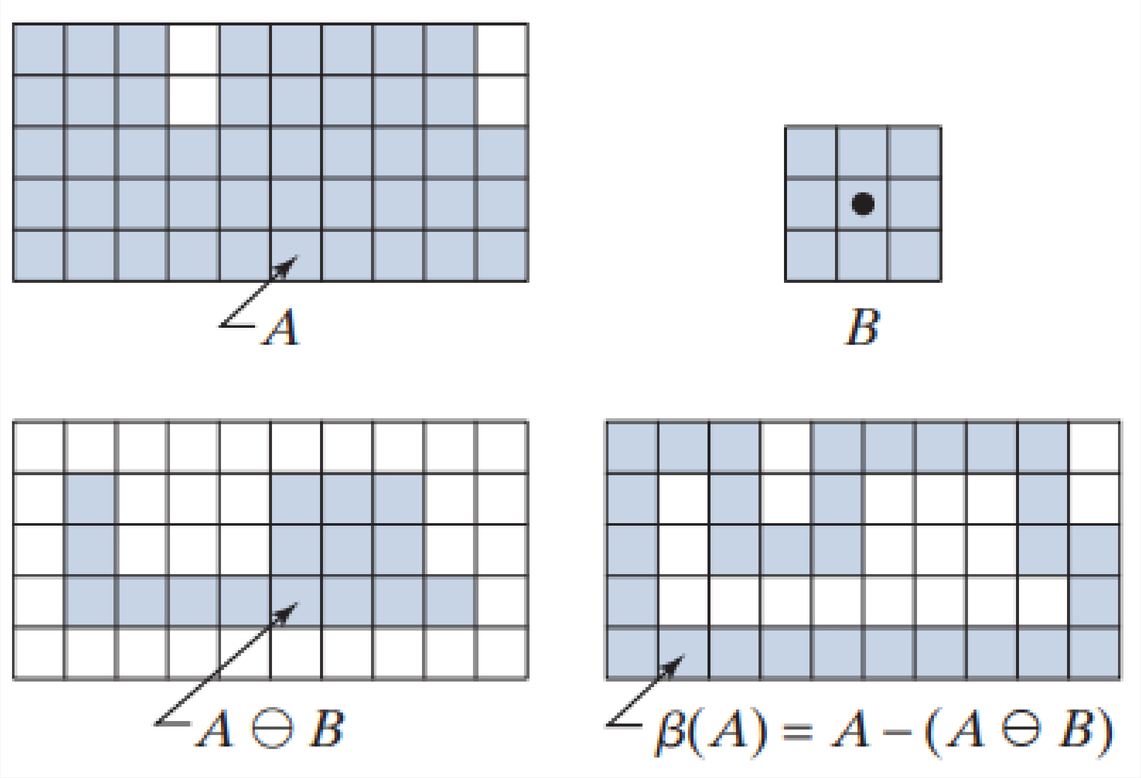


Morphology

Some basic morphological algorithm

- Boundary extraction

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

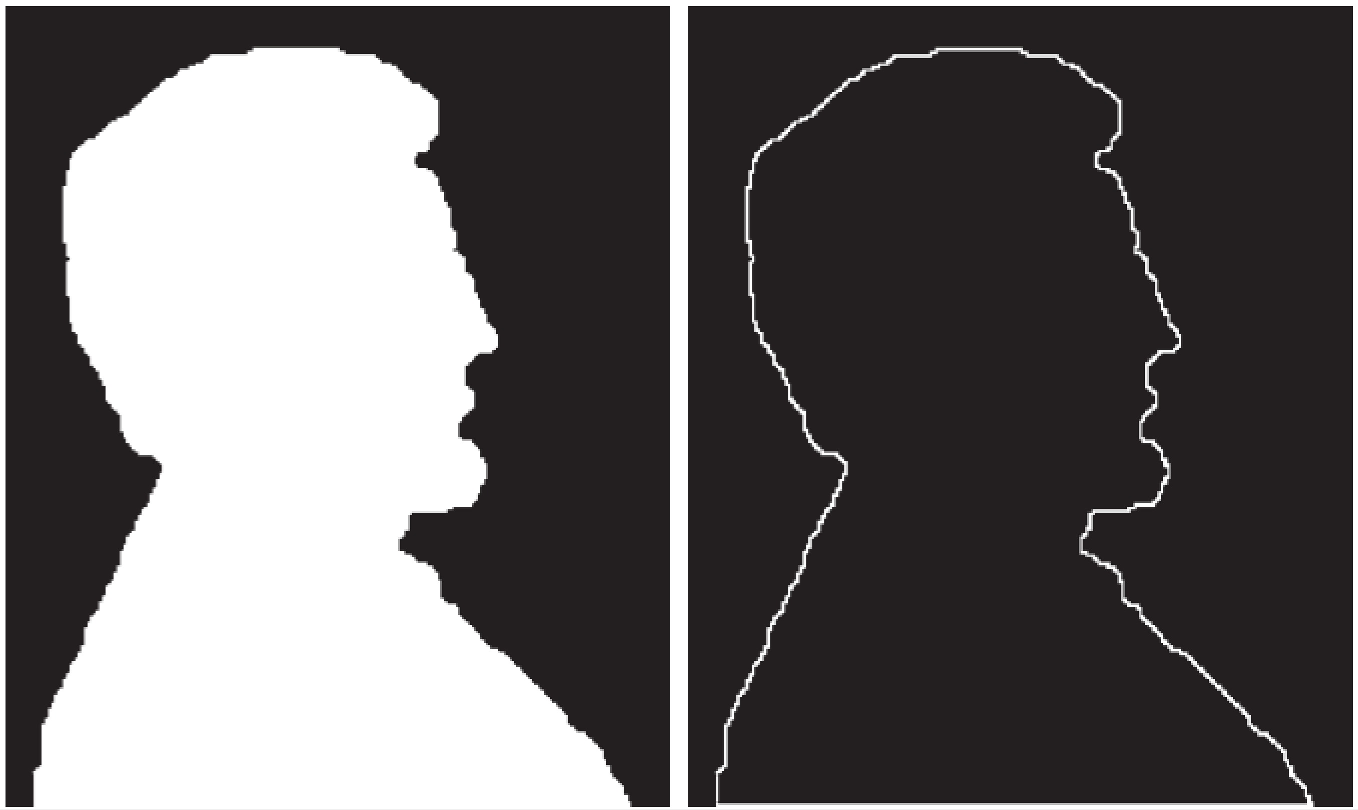


Morphology

Some basic morphological algorithm

- Boundary extraction

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

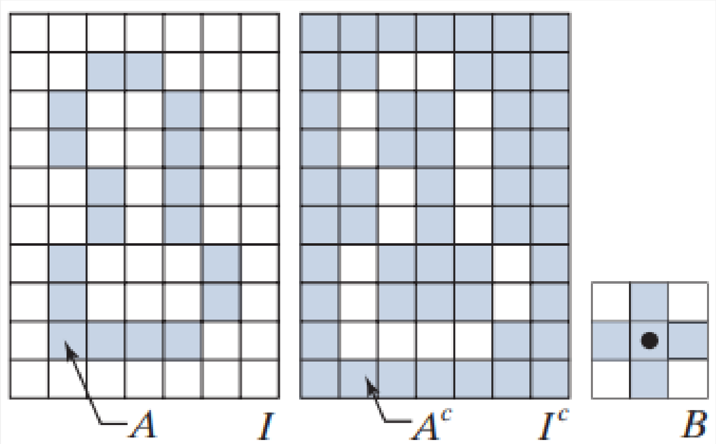


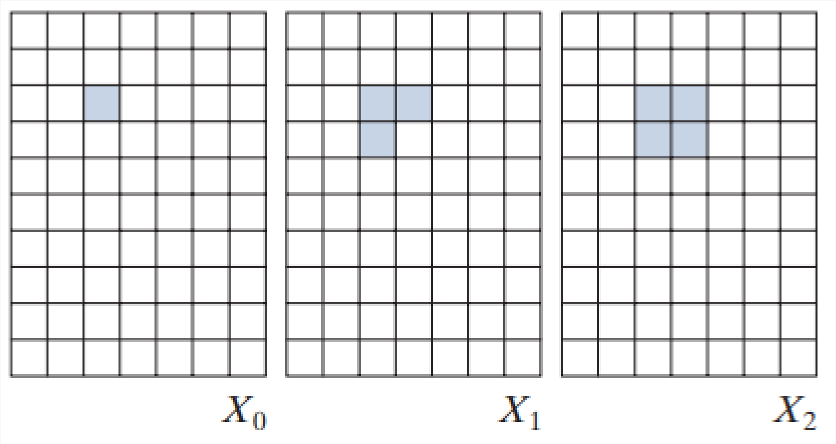
Morphology

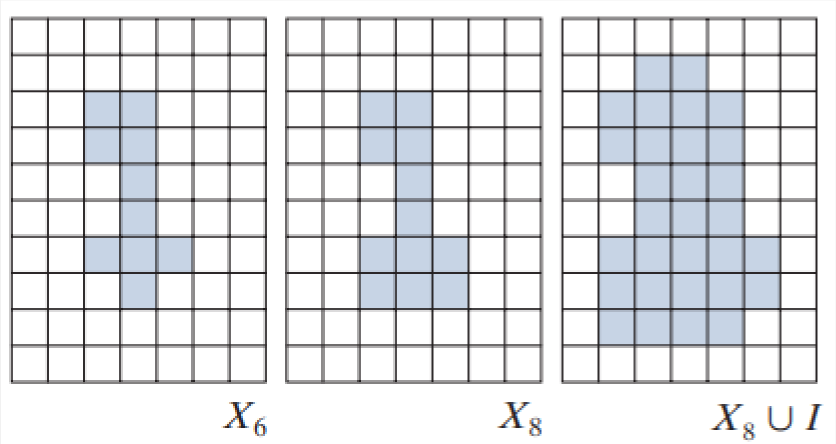
Some basic morphological algorithm

- Hole filling

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





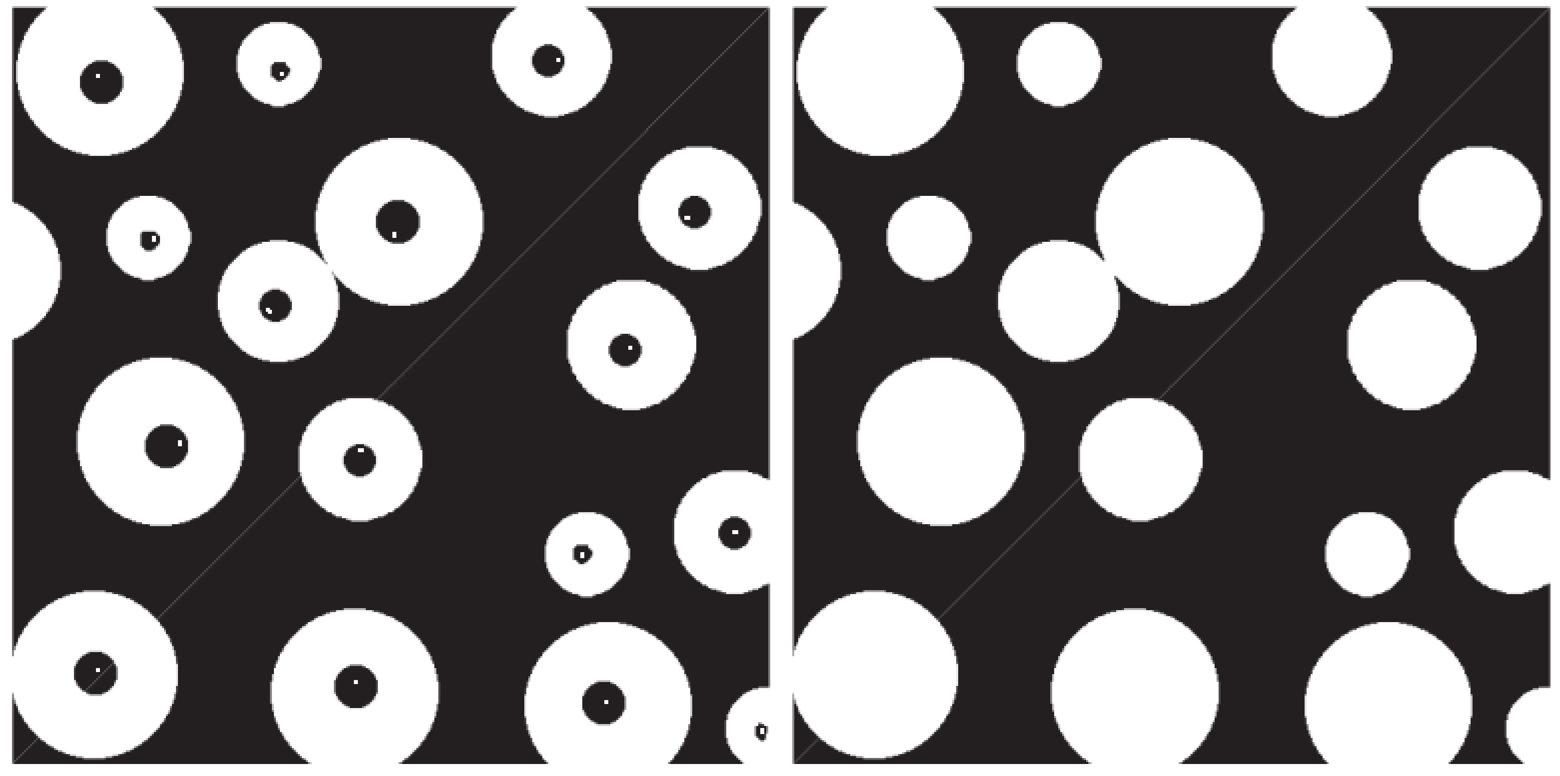


Morphology

Some basic morphological algorithm

- Hole filling

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

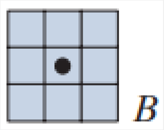


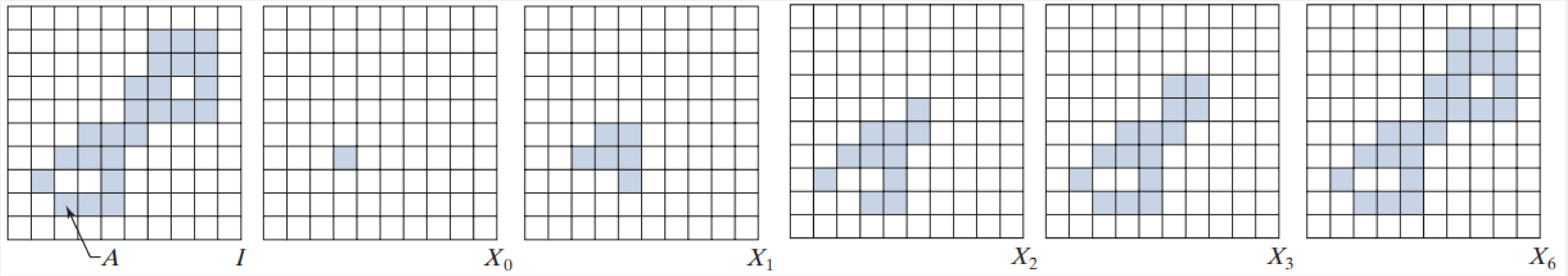
Morphology

Some basic morphological algorithm

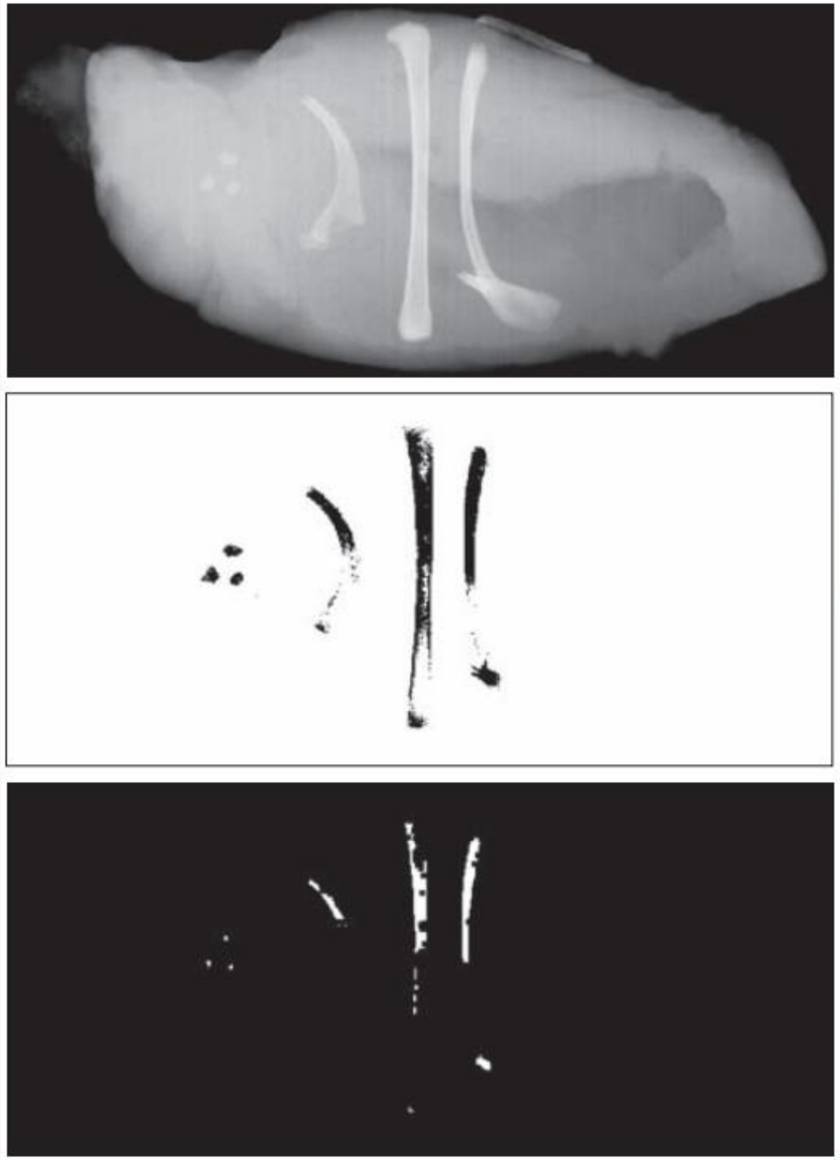
- Extraction of connected components

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





Morphology

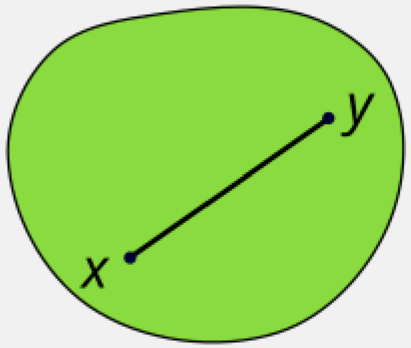


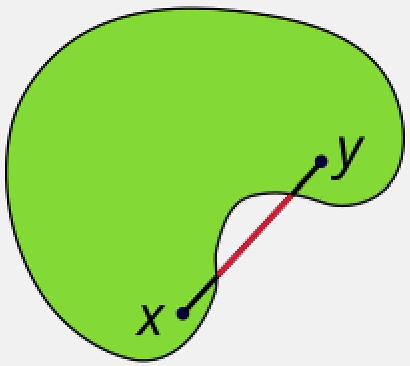
Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

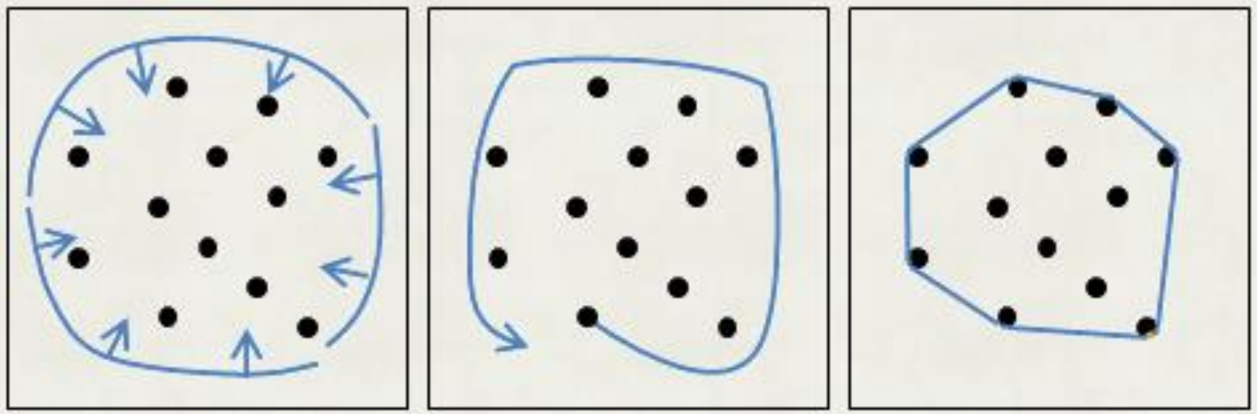
Morphology

Some basic morphological algorithm

- Convex hull
  - A set, *S*, of points in the Euclidean plane is said to be convex if and only if a straight line segment joining any two points in *S* lies entirely within *S*
  - Convex hull, *H*, of *S* is the smallest convex set containing *S*







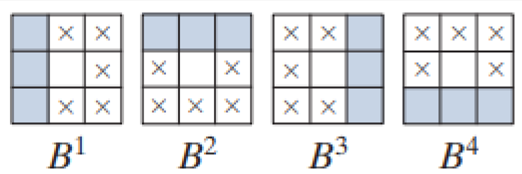
Morphology

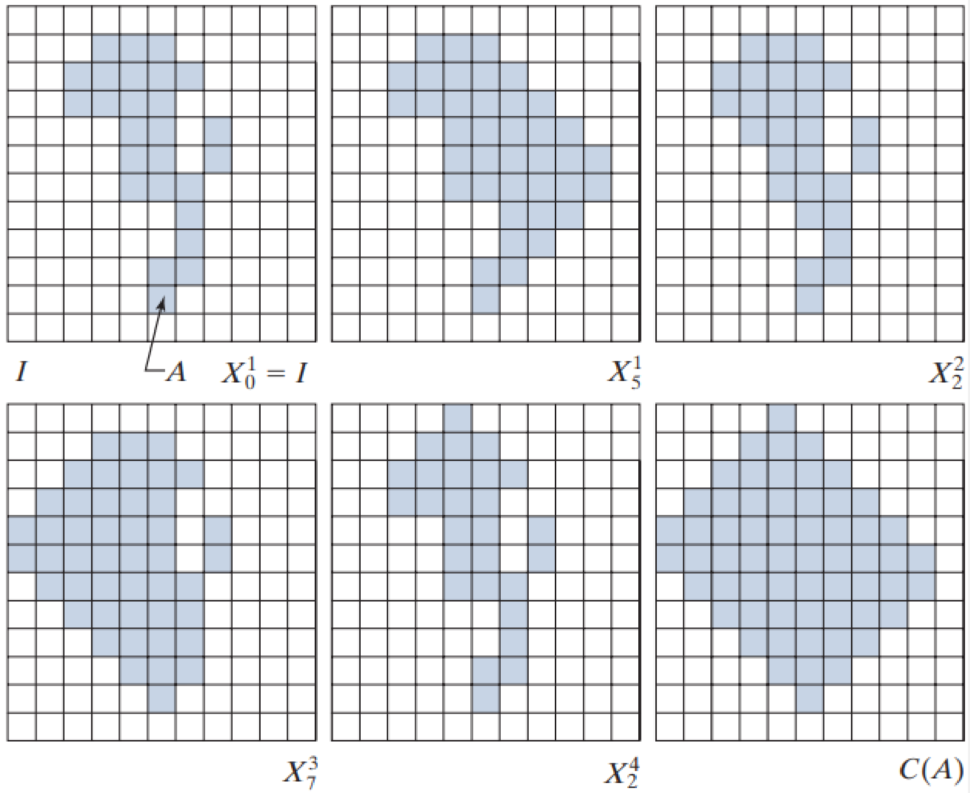
Some basic morphological algorithm

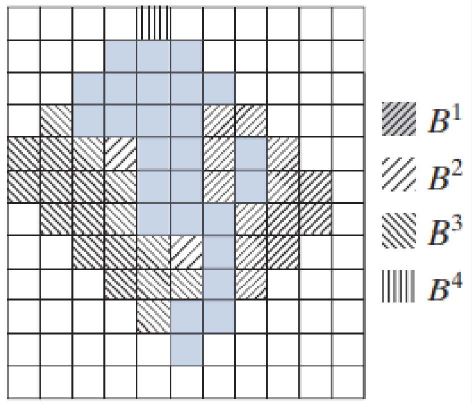
- Convex hull

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

$$C(A) = \bigcup_{i=1}^{4} D^i \quad \text{where } D^i = X_k^i$$



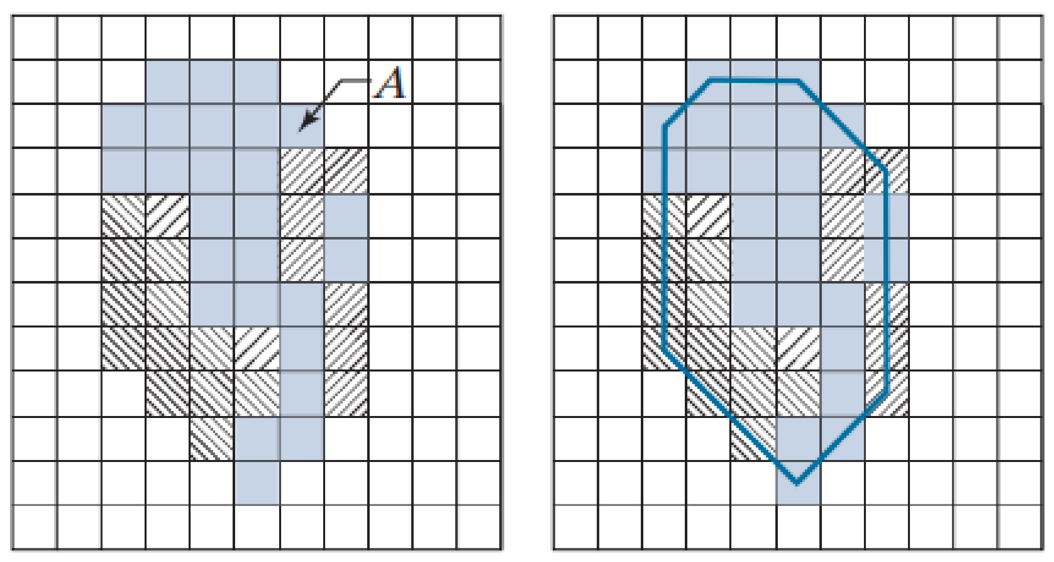




Morphology

Some basic morphological algorithm

- Limiting growth of convex hull algorithm



Morphology

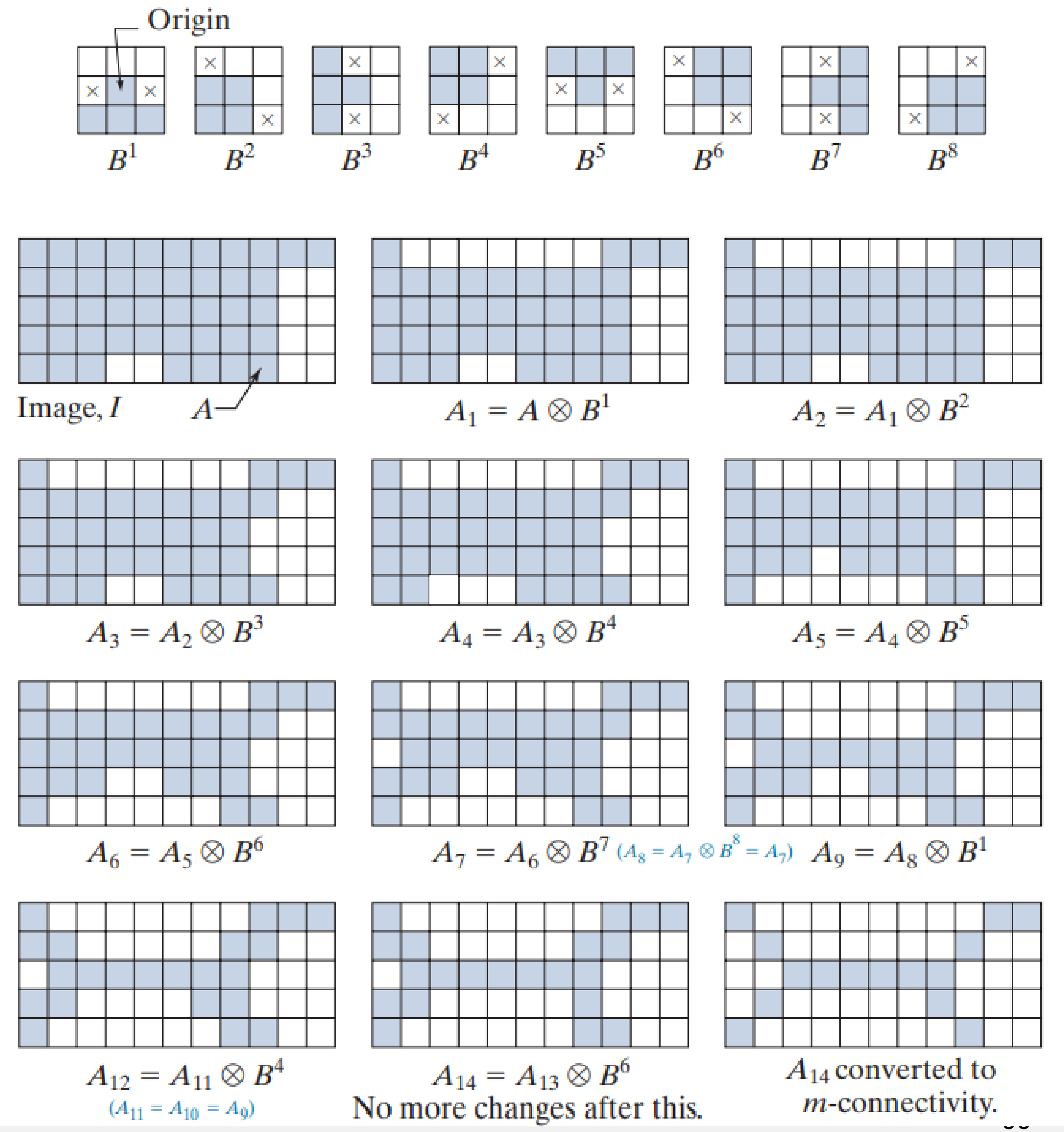
Some basic morphological algorithm

- Thinning

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

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

$$A \otimes \{B\} = \left(\left(\dots\left(\left(A \otimes B^1\right) \otimes B^2\right)\dots\right) \otimes B^n\right)$$



Morphology

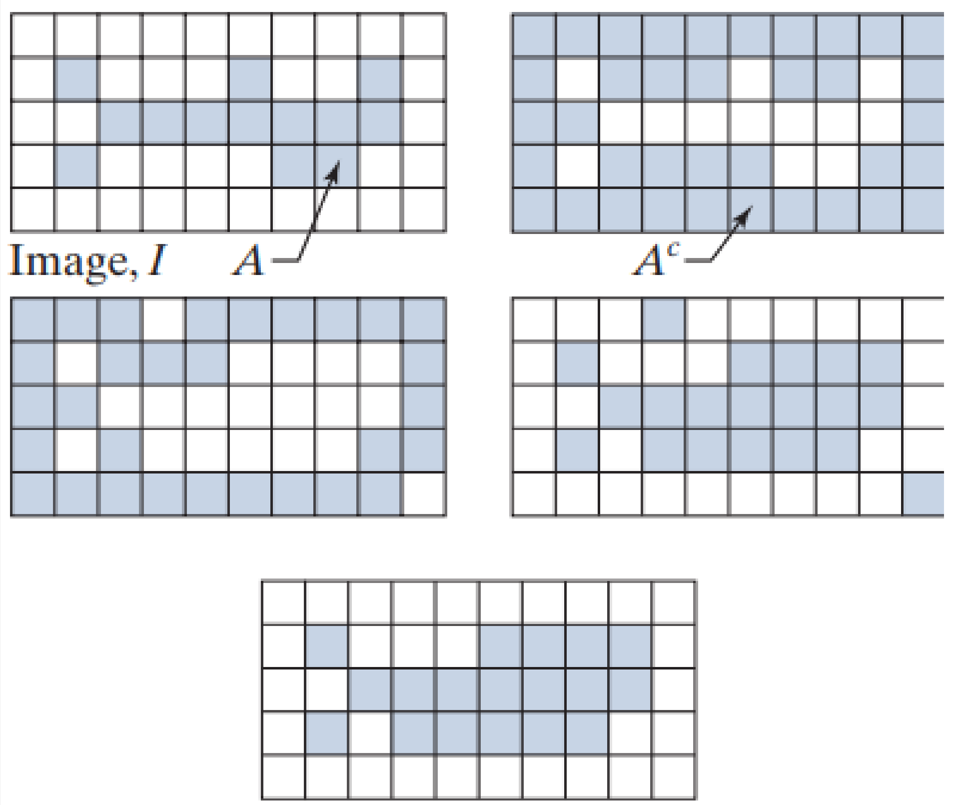
Some basic morphological algorithm

- Thickening (i.e. thinning the background)

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

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

$$A \odot \{B\} = \left(\left(\dots\left(\left(A \odot B^1\right) \odot B^2\right)\dots\right) \odot B^n\right)$$



Morphology

Some basic morphological algorithm

- Skeletons
  - Principles (skeleton of A: S(A))
    - ✓ Maximum disk $(D)_z$: 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 simultaneously included in A
    - ✓ If $(D)_z$ is a maximum disk, it touches the boundary of A at two or more different places

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

$(A \ominus kB)$ indicates k successive erosions starting with A

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

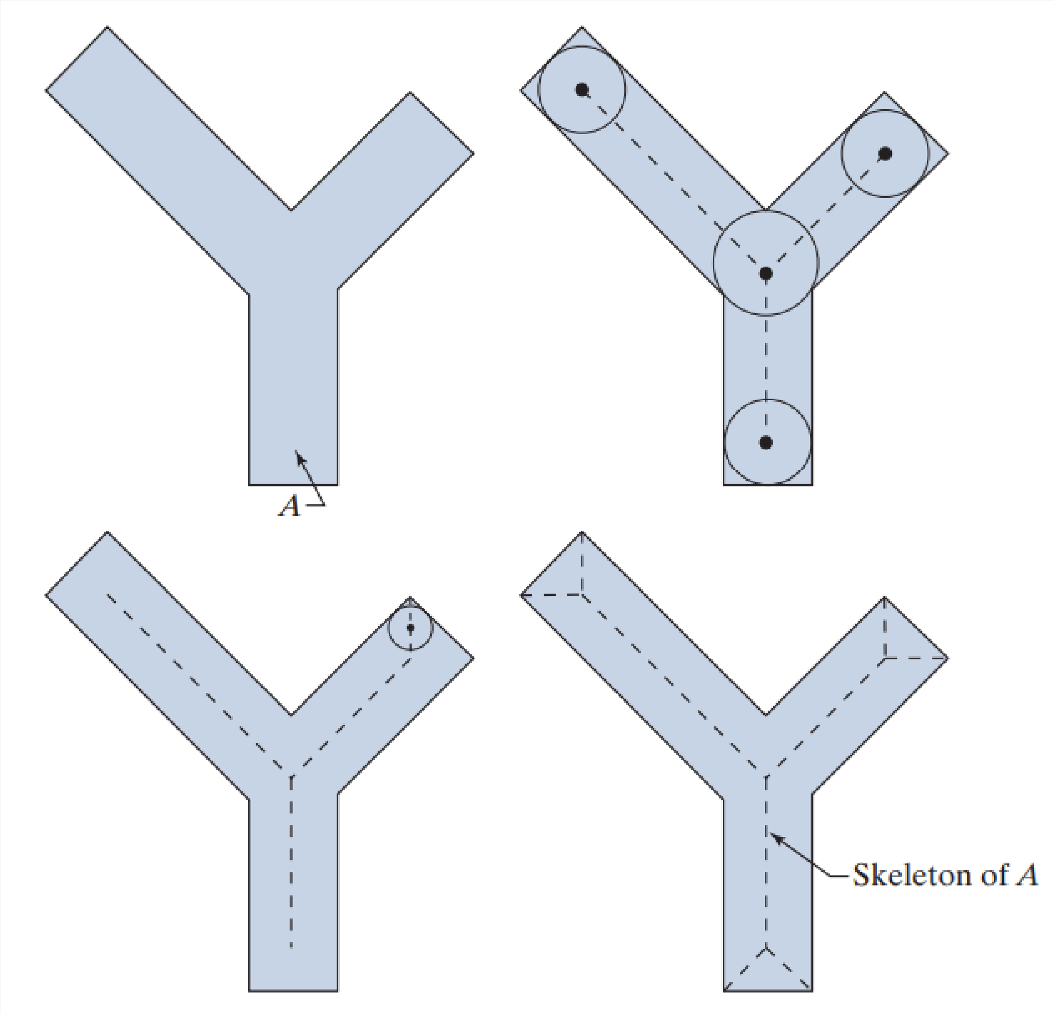
$$K = \max\{k | (A \ominus kB) \neq \emptyset\}$$

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

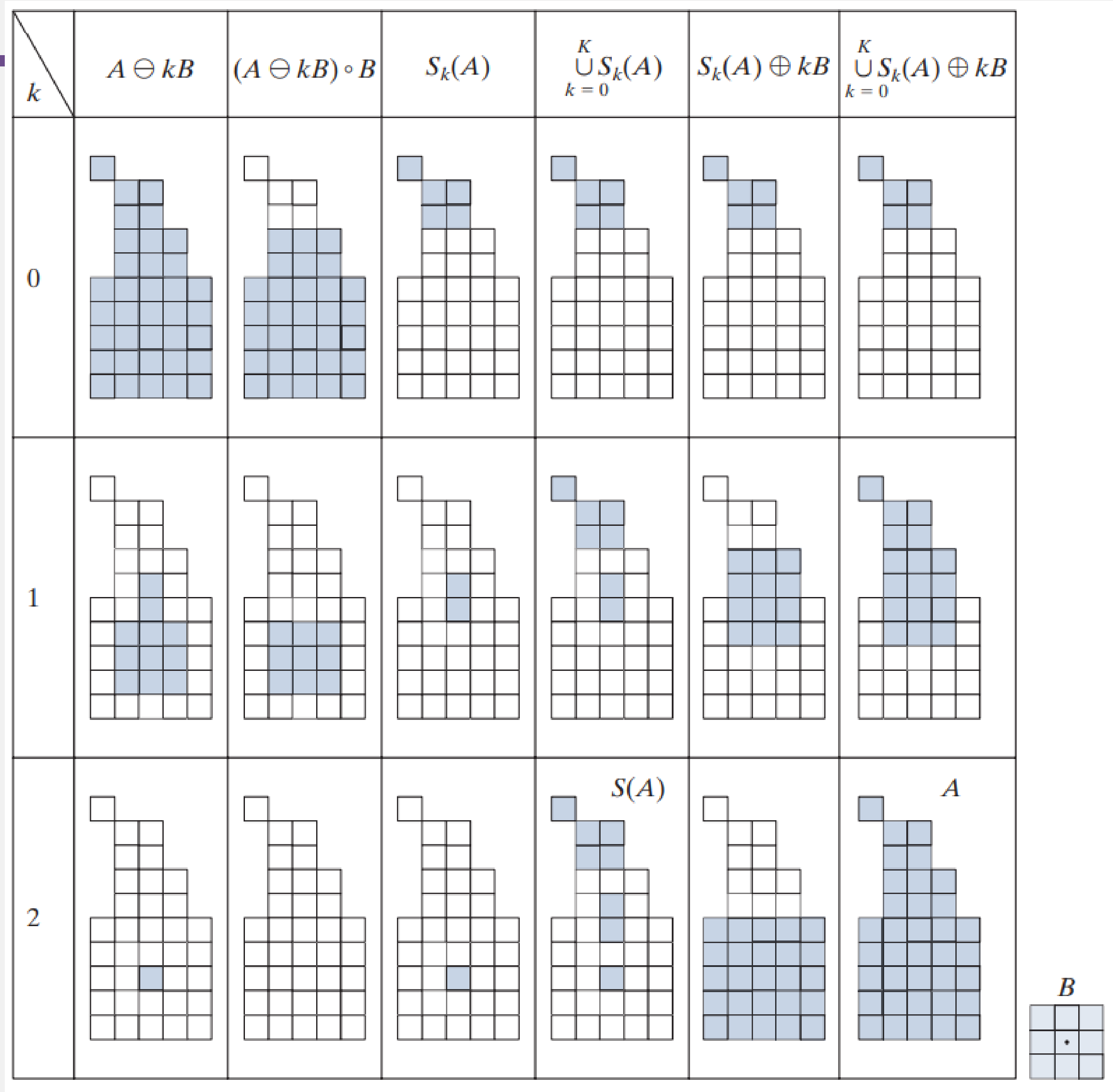
Morphology

Some basic morphological algorithm

- Skeletons



Morphology



k
$A \ominus kB$
$(A \ominus kB) \circ B$
$S_k(A)$
$\bigcup_{k=0}^{K} S_k(A)$
$S_k(A) \oplus kB$
$\bigcup_{k=0}^{K} S_k(A) \oplus kB$
0
1
2
S(A)
A
B

Morphology

Summary of binary morphological operations

Translation	$(B)_z = \{c \mid c = b + z, \text{ for } b \in B\}$	Translates the origin of B to point z.
Reflection	$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\}$	Reflects B about its origin.
Complement	$A^c = \{w \mid w \notin A\}$	Set of points not in A.
Difference	$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c$	Set of points in A, but not in B.
Erosion	$A \ominus B = \{z \mid (B)_z \subseteq A\}$	Erodes the boundary of A. (I)
Dilation	$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \varnothing\}$	Dilates the boundary of A. (I)
Opening	$A \circ B = (A \ominus B) \oplus B$	Smoothes contours, breaks narrow isthmuses, and eliminates small islands and sharp peaks. (I)
Closing	$A \bullet B = (A \oplus B) \ominus B$	Smoothes contours, fuses narrow breaks and long thin gulfs, and eliminates small holes. (I)

Morphology

Summary of binary morphological operations

Hit-or-miss transform	$I \circledast B = \{z \mid (B)_z \subseteq I\}$	Finds instances of B in image I. B contains *both* foreground and background elements.
Boundary extraction	$\beta(A) = A - (A \ominus B)$	Set of points on the boundary of set A. (I)
Hole filling	$X_k = (X_{k-1} \oplus B) \cap I^c$ $k = 1, 2, 3, \ldots$	Fills holes in A. X_0 is of same size as I, with a 1 in each hole and 0's elsewhere. (II)
Connected components	$X_k = (X_{k-1} \oplus B) \cap I$ $k = 1, 2, 3, \ldots$	Finds connected components in I. X_0 is a set, the same size as I, with a 1 in each connected component and 0's elsewhere. (I)
Convex hull	$X_k^i = (X_{k-1}^i \circledast B^i) \cup X_{k-1}^i;$ $i = 1, 2, 3, 4 \quad k = 1, 2, 3, \ldots$ $X_0^i = I;\ D^i = X_{conv}^i;\ C(A) = \bigcup_{i=1}^{4} D^i$	Finds the convex hull, $C(A)$, of a set, A, of foreground pixels contained in image I. X_{conv}^i means that $X_k^i = X_{k-1}^i$. (III)

Morphology

Summary of binary morphological operations

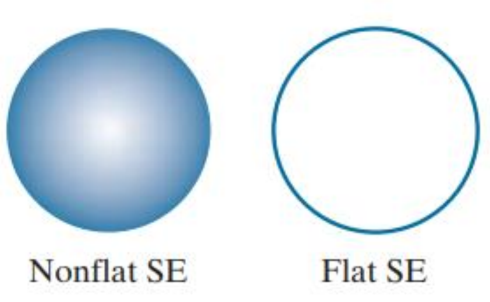
Thinning	$A \otimes B = A - (A \circledast B)$ $= A \cap (A \circledast B)^c$ $A \otimes \{B\} =$ $((\ldots((A \otimes B^1) \otimes B^2)\ldots) \otimes B^n)$ $\{B\} = \{B^1, B^2, B^3, \ldots, B^n\}$	Thins set A. The first two equations give the basic definition of thinning. The last two equations denote thinning by a sequence of structuring elements. This method is normally used in practice. (IV)
Thickening	$A \odot B = A \cup (A \circledast B)$ $A \odot \{B\} =$ $((\ldots((A \odot B^1) \odot B^2)\ldots) \odot B^n)$	Thickens set A using a sequence of structuring elements, as above. Uses (IV) with 0's and 1's reversed.

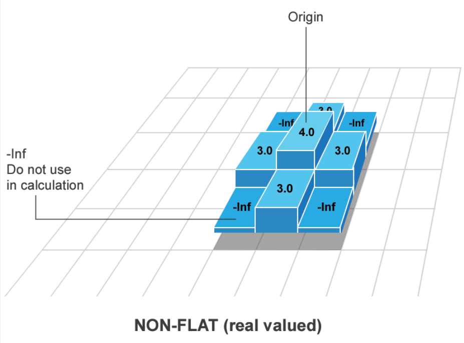
Skeletons	$S(A) = \bigcup_{k=0}^{K} S_k(A)$ $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$ Reconstruction of A: $A = \bigcup_{k=0}^{K} (S_k(A) \oplus kB)$	Finds the skeleton $S(A)$ of set A. The last equation indicates that A can be reconstructed from its skeleton subsets $S_k(A)$. K is the value of the iterative step after which the set A erodes to the empty set. The notation $(A \ominus kB)$ denotes the kth iteration of successive erosions of A by B. (I)
Pruning	$X_1 = A \otimes \{B\}$ $X_2 = \bigcup_{k=1}^{8} (X_1 \circledast B^k)$ $X_3 = (X_2 \oplus H) \cap A$ $X_4 = X_1 \cup X_3$	X_4 is the result of pruning set A. The number of times that the first equation is applied to obtain X_1 must be specified. Structuring elements (V) are used for the first two equations. In the third equation H denotes structuring element. (I)

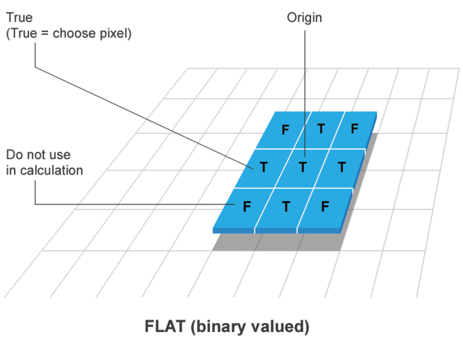
Morphology

Grayscale morphology

- Grayscale image: $f(x, y)$
- Structuring element: $b(x, y)$
  - Two categories: nonflat and flat

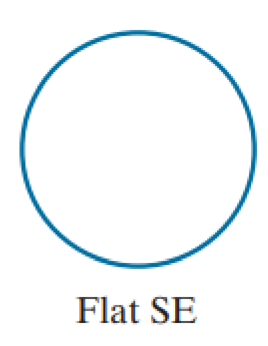


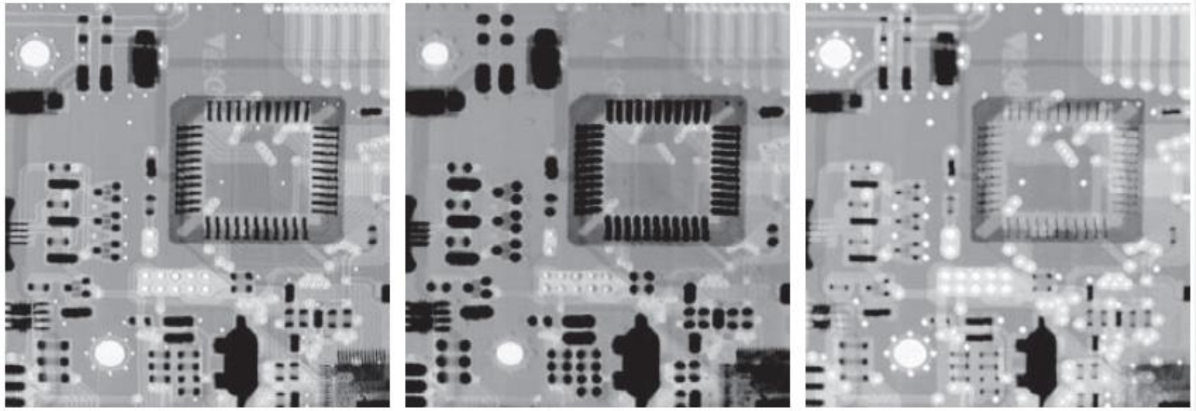




Some basic morphological operations with flat SE

- Dilation: $[f \oplus b](x,y) = \max_{(s,t)\in\hat{b}} \{f(x-s, y-t)\}$
- Erosion: $[f \ominus b](x,y) = \min_{(s,t)\in b} \{f(x+s, y+t)\}$
- Opening: $f \circ b = [f \ominus b] \oplus b$
- Closing: $f \bullet b = [f \oplus b] \ominus b$





Morphology

Some basic morphological operations with nonflat SE

- Dilation: $[f \oplus b_N](x,y) = \max_{(s,t)\in \hat{b}_N} \{f(x-s, y-t) + \hat{b}_N(s,t)\}$
- Erosion: $[f \ominus b_N](x,y) = \min_{(s,t)\in b_N} \{f(x+s, y+t) - b_N(s,t)\}$
- Opening: $f \circ b = [f \ominus b] \oplus b$
- Closing: $f \bullet b = [f \oplus b] \ominus b$

Nonflat SE

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	7	4	5	6	7	0
0	6	4	7	8	6	0
0	7	9	8	7	7	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

0	3	0
3	3	3
0	3	0

SE

Dilation

0	0	0	0	0	0	0
0	10	7	8	9	10	0
0	10	7	10	11	10	0
0	10	12	11	11	11	0
0	12	12	12	11	10	0
0	10	12	11	10	10	0
0	0	0	0	0	0	0

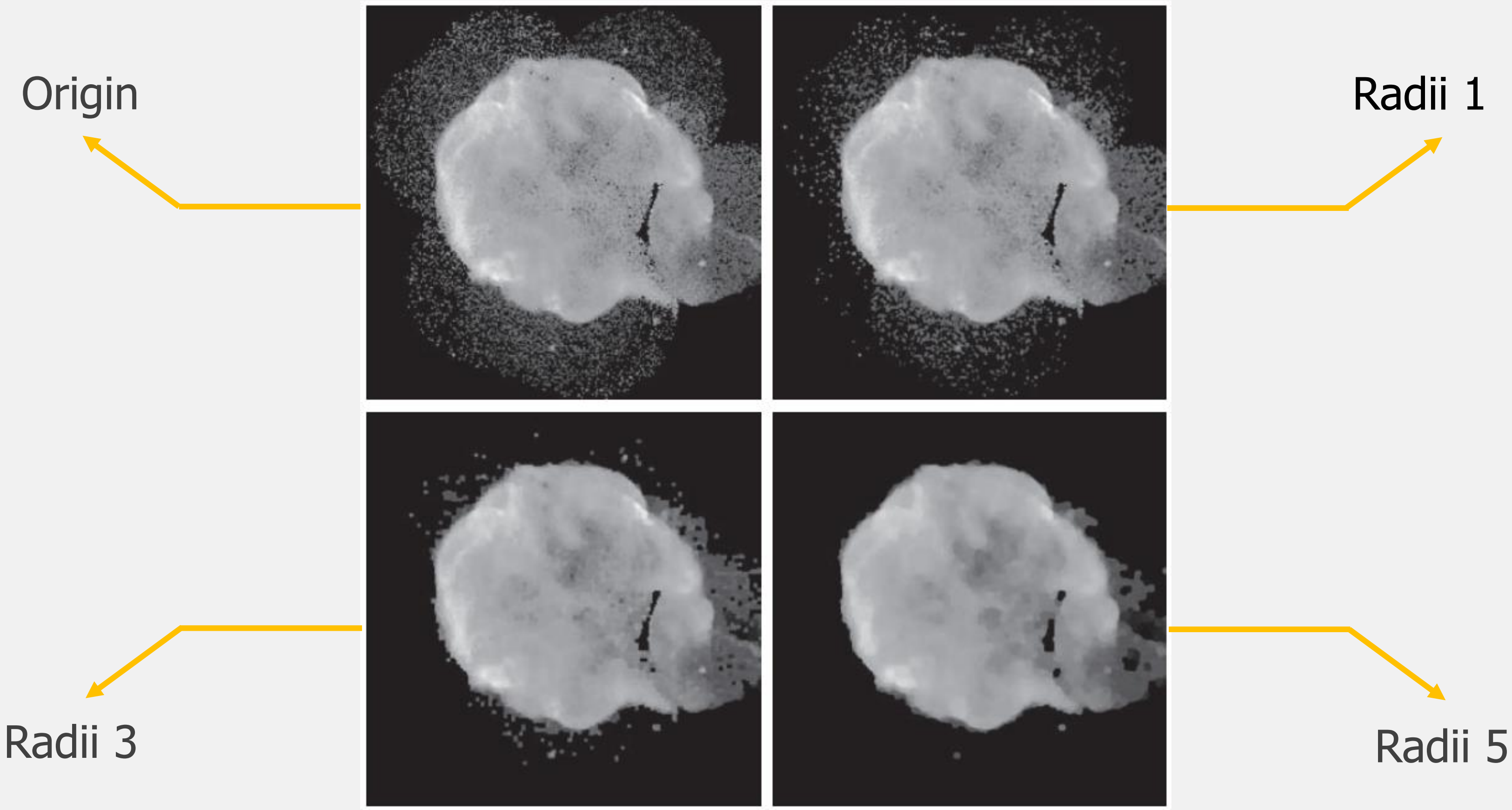
Erosion

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	3	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Morphology

Grayscale morphological smoothing

- Performing opening and closing sequences on the original image with disk structuring elements of radii, 1, 3, and 5, respectively

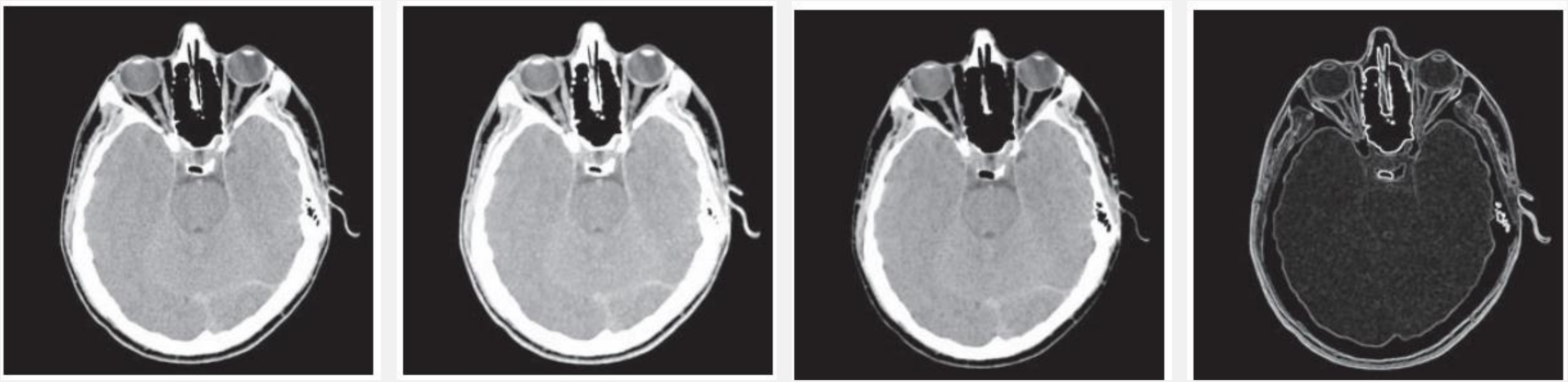


Morphology

Grayscale morphological gradient

- Dilation and erosion can be used in combination with image subtraction to obtain the morphological gradient

$$g = (f \oplus b) - (f \ominus b)$$



Morphology

Top-hat and bottom-hat transformations

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

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

Origin | Thresholded origin | Open using a disk SE of radius 40 | Top-hat | Thresholded Top-hat

Morphology

Using opencv to perform edge detection and morphology operations

[Data]

Original Image

Sobel

Canny

Morphology

Using opencv to perform edge detection and morphology operations

Data

Original Image Erosion Dilation Opening Closing