Lec. 6

Morphology Operator

Assist. Prof. Dr. Saad Albawi

INTRODUCTION

- Mathematical morphology is a branch of image processing that has been successfully used to provide tools for representing, describing, and analyzing shapes in images.

- It is class of neighborhood operations on binary images, the morphological operators that modify and analyze the form of objects.

- In addition to providing useful tools for extracting image components, morphological algorithms have been used for pre- or postprocessing the images containing shapes of interest.

- The basic principle of mathematical morphology is the extraction of geometrical and topological information from an unknown set (an image) through transformations using another, well-defined, set known as *structuring element*.

- In morphological image processing, the design of SEs, their shape and size, is crucial to the success of the morphological operations that use them.

Morphology relates to the structure or form of objects. Morphological filtering simplifies a segmented image to facilitate the search for objects of interest. This is done by smoothing out object outlines, filling small holes, eliminating small projections, and using other similar techniques. Even though this section will focus on applications to binary images,

The two principal morphological operations are dilation and erosion. Dilation allows objects to expand, thus potentially filling in small holes and connecting disjoint objects. Erosion shrinks objects by etching away (eroding) their boundaries. These operations can be customized for an application by the proper selection of the structuring element, which determines exactly how the objects will be dilated or eroded. The dilation process is performed by laying the structuring element on the image and sliding it across the image in a manner similar to convolution.

➢ Neighborhood Operations on Binary Images

❑ Binary Convolution

- **The result of operation in binary images can only be a zero or a one. Consequently, neighborhood operators for binary images will work on the shape of object, adding pixels to an object or deleting pixels from an object.**

- **Neighboring pixels of gray value images: convolution (“weighting and summing up”) and rank value filtering (“sorting and selecting”). With binary images, we do not have much choice as to which kind of operations to perform.**

- **We can combine pixels only with the logical operations of Boolean algebra. We might introduce a *binary convolution* by replacing the multiplication of the image and mask pixels with an *and operation* and the summation by and, *or operation*:**

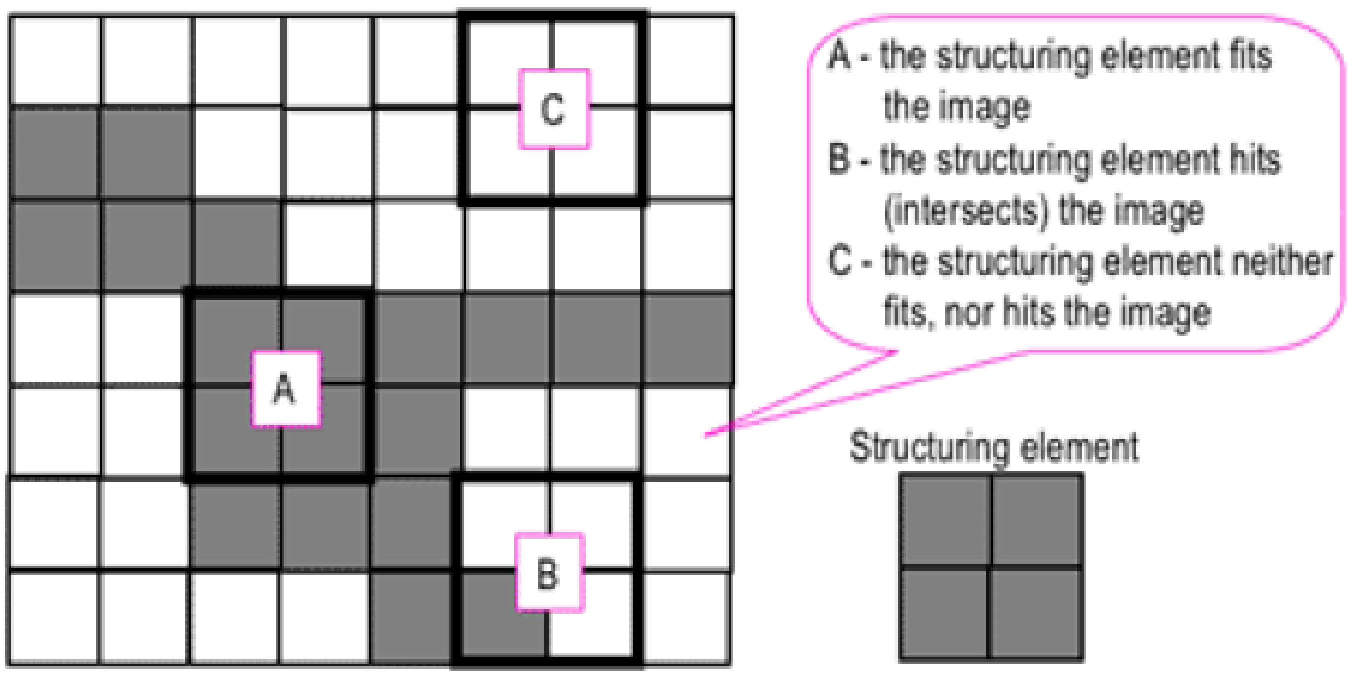
- The $\wedge$ and $\vee$ denote the logical (AND & OR) operations, respectively. The binary image G is convolved with a symmetric $2R + 1 \times 2R + 1$ mask M.

- Let us assume that all the coefficients of the mask are set to 'one'. If one or more object pixels, i. e., 'ones', are within the mask, the result of the operation will be one, otherwise it is zero .

- Hence, the object will be dilated. Small holes or cracks will be filled and the contour line will become smoother, .

The Structuring Element

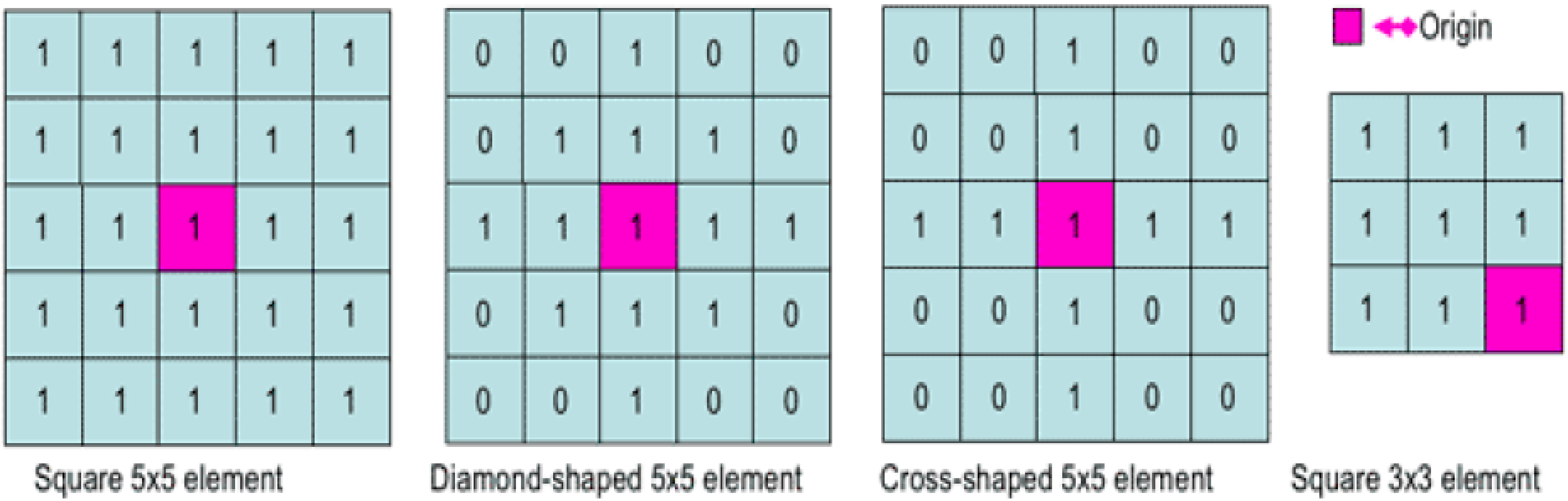
The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels. Some operations test whether the element "fits" within the neighbourhood, while others test whether it "hits" or intersects the neighbourhood:



Probing of an image with a structuring element
(white and grey pixels have zero and non-zero values, respectively).

- **The structuring element is the basic neighborhood structure associated with morphological image operations.**
- **It is usually represented as a small matrix, whose shape and size impact the result of applying a certain morphological operator to an image.**
- **Although a structuring element can have any shape, its implementation requires that it should be converted to a rectangular array.**
- **For each array, the shaded squares correspond to the members of the SE, whereas the empty squares are used for padding, only.**

When a structuring element is placed in a binary image, each of its pixels is associated with the corresponding pixel of the neighbourhood under the structuring element. The structuring element is said to **fit** the image if, for each of its pixels set to 1, the corresponding image pixel is also 1. Similarly, a structuring element is said to **hit**, or intersect, an image if, at least for one of its pixels set to 1 the corresponding image pixel is also 1.



Examples of simple structuring elements.

FUNDAMENTAL CONCEPTS AND OPERATIONS

The basic concepts of mathematical morphology can be introduced with the help of set theory and its standard operations: *union* (∪), *intersection* (∩), and *complement*, defined as

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

and the *difference* of two sets *A* and *B*:

$$A - B = \{z | z \in A,\ z \notin B\} = A \cap B^c$$

Let *A* be a set (of pixels in a binary image) and *w* = (*x, y*) be a particular coordinate point. The *translation* of set *A* by point *w* is denoted by *Aw* and defined as

$$A_w = \{c | c = a + w, \quad \text{for} \quad a \in A\}$$

The *reflection* of set *A* relative to the origin of a coordinate system, denoted *A*ˆ , is defined as

$$\hat{A} = \{z | z = -a, \quad \text{for} \quad a \in A\}$$

The following figure shows a graphical representation of the basic set operations defined above. The black dot represents the origin of the coordinate system.

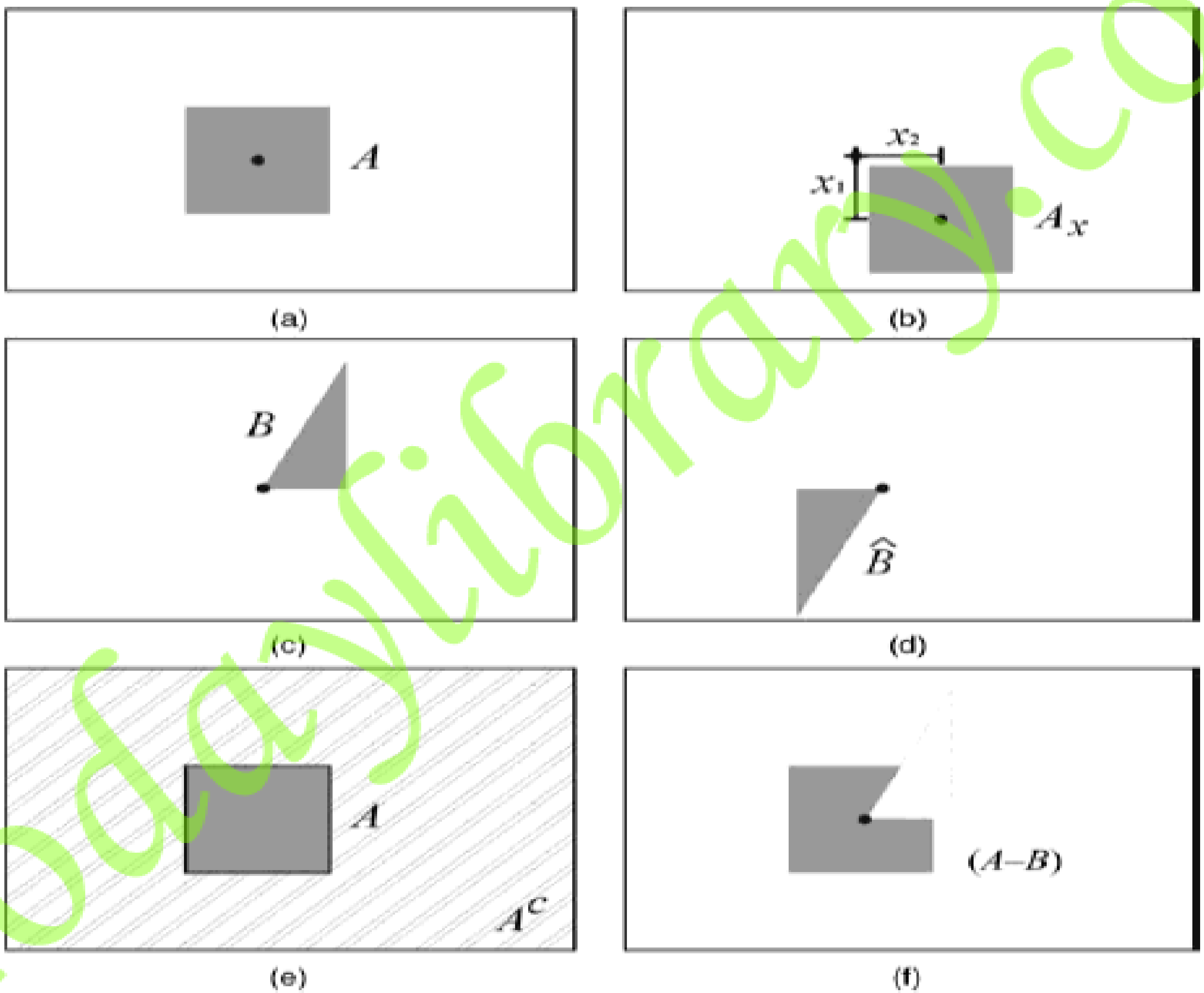


FIGURE 13.1 Basic set operations: (a) set A; (b) translation of A by $x = (x_1, x_2)$; (c) set B; (d) reflection of B; (e) set A and its complement A^c; (f) set difference $(A-B)$.

Binary mathematical morphology theory views binary images as a set of its foreground pixels (whose values are assumed to be 1).

Classical image processing refers to a binary image as a function of *x* and *y*, whose only possible values are 0 and 1. To avoid any potential confusion that this dual view may cause, here is an example of how a statement expressed in set theory notation can be translated into a set of logical operations applied to binary images:
The statement $C = A \cap B$, from a set theory perspective, means

$$C = \{(x, y) | (x, y) \in A \text{ and } (x, y) \in B\}$$

The equivalent expression using conventional image processing notation would be

$$C(x, y) = \begin{cases} 1 & \text{if } A(x, y) \text{ and } B(x, y) \text{ are both } 1 \\ 0 & \text{otherwise} \end{cases}$$

This expression leads quite easily to a single MATLAB statement that performs the intersection operation using the logical operator AND (&). Similarly, complement can be obtained using the unary NOT (˜) operator, set union can be implemented using the logical operator OR (|), and set difference (*A* − *B*) can be expressed as (A & ˜B).

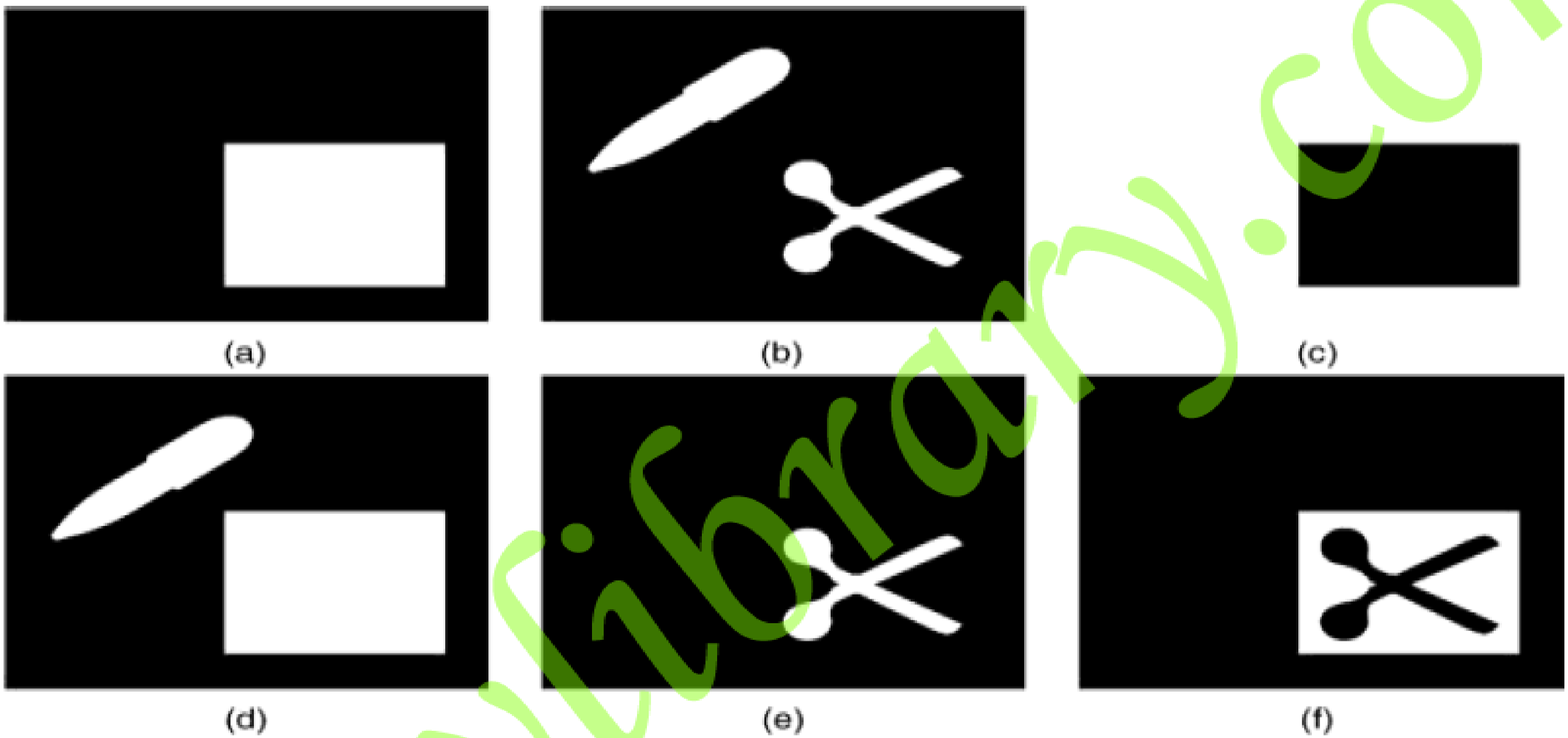


FIGURE 13.2 Logical equivalents of set theory operations: (a) Binary image (A); (b) Binary image (B); (c) Complement (A^c); (d) Union ($A \cup B$); (e) Intersection ($A \cap B$); (f) Set difference ($A - B$).

Erosion and dilation

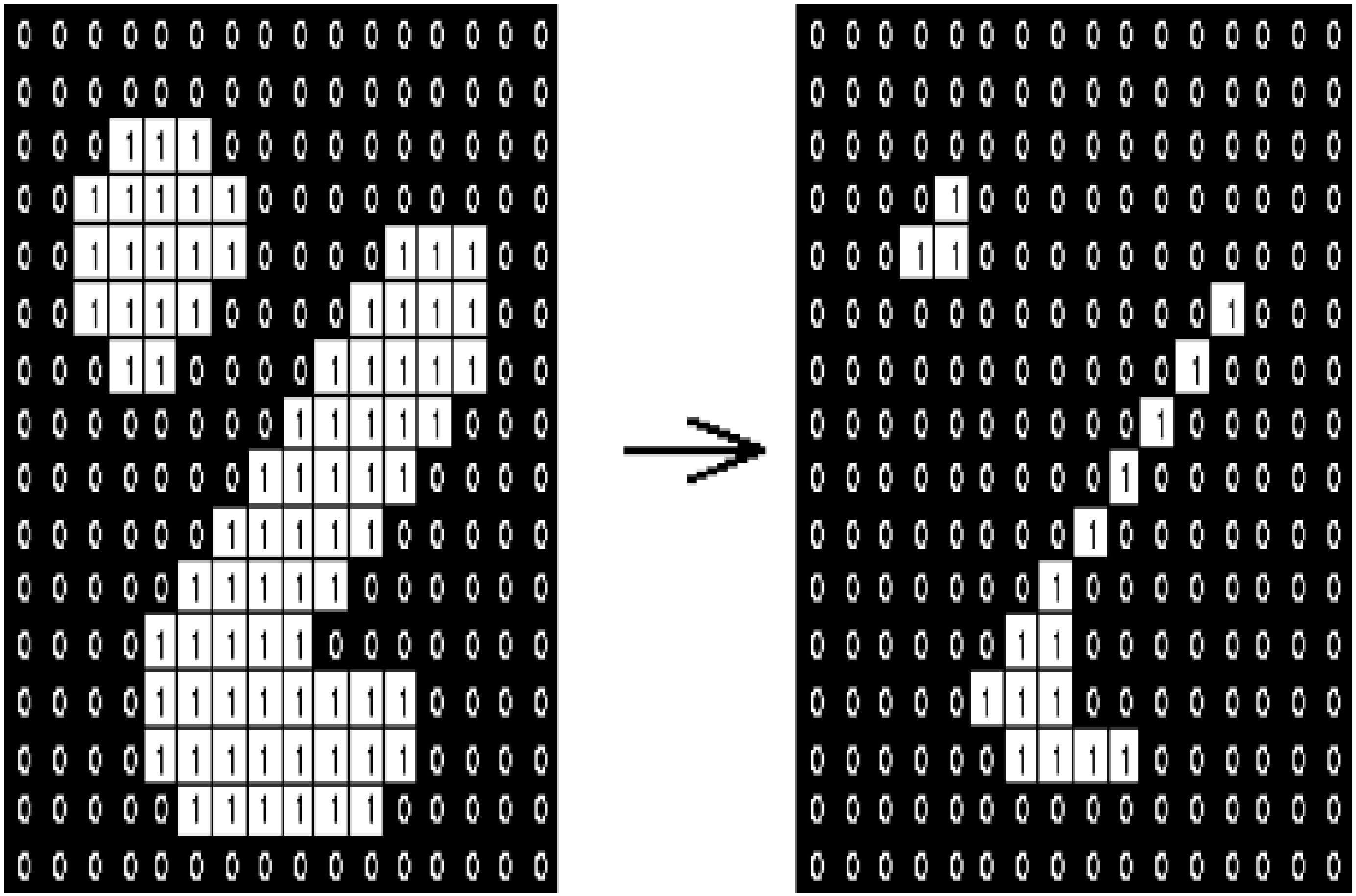
The **erosion** of a binary image f by a structuring element s (denoted $f \ominus s$) produces a new binary image $g = f \ominus s$ with ones in all locations (x,y) of a structuring element's origin at which that structuring element s fits the input image f, i.e. $g(x,y) = 1$ is s fits f and 0 otherwise, repeating for all pixel coordinates (x,y).

Greyscale image

Binary image by thresholding

Erosion: a 2×2 square structuring element

Erosion with small (e.g. 2×2 - 5×5) square structuring elements shrinks an image by stripping away a layer of pixels from both the inner and outer boundaries of regions. The holes and gaps between different regions become larger, and small details are eliminated:



Erosion: a 3×3 square structuring element
(www.cs.princeton.edu/~pshilane/class/mosaic/).

Larger structuring elements have a more pronounced effect, the result of erosion with a large structuring element being similar to the result obtained by iterated erosion using a smaller structuring element of the same shape. If s_1 and s_2 are a pair of structuring elements identical in shape, with s_2 twice the size of s_1, then

$$f \ominus s_2 \approx (f \ominus s_1) \ominus s_1.$$

Erosion removes small-scale details from a binary image but simultaneously reduces the size of regions of interest, too. By subtracting the eroded image from the original image, boundaries of each region can be found: $b = f - (f \ominus s)$ where f is an image of the regions, s is a 3×3 structuring element, and b is an image of the region boundaries.

The erosion process turn pixels to 'O', not '1'. Slide the structuring element across the image and then follow these steps:

1. If the origin of the structuring element coincides with a '0' in the image, there is no change; move to the next pixel.
2. If the origin of the structuring element coincides with a '1' in the image, and any of the '1' pixels in the structuring element extend beyond the object ('1' pixels) in the image, then change the '1' pixel in the image to a '0'.

Erosion is a morphological operation whose effect is to "shrink" or "thin" objects in a binary image. The direction and extent of this thinning is controlled by the shape and size of the structuring element. Mathematically, the erosion of a set *A* by *B*, denoted *A* g *B*

$$g'_{mn} = \bigwedge_{m'=-R}^{R} \bigwedge_{n'=-R}^{R} m_{m',n'} \wedge g_{m+m',n+n'}$$

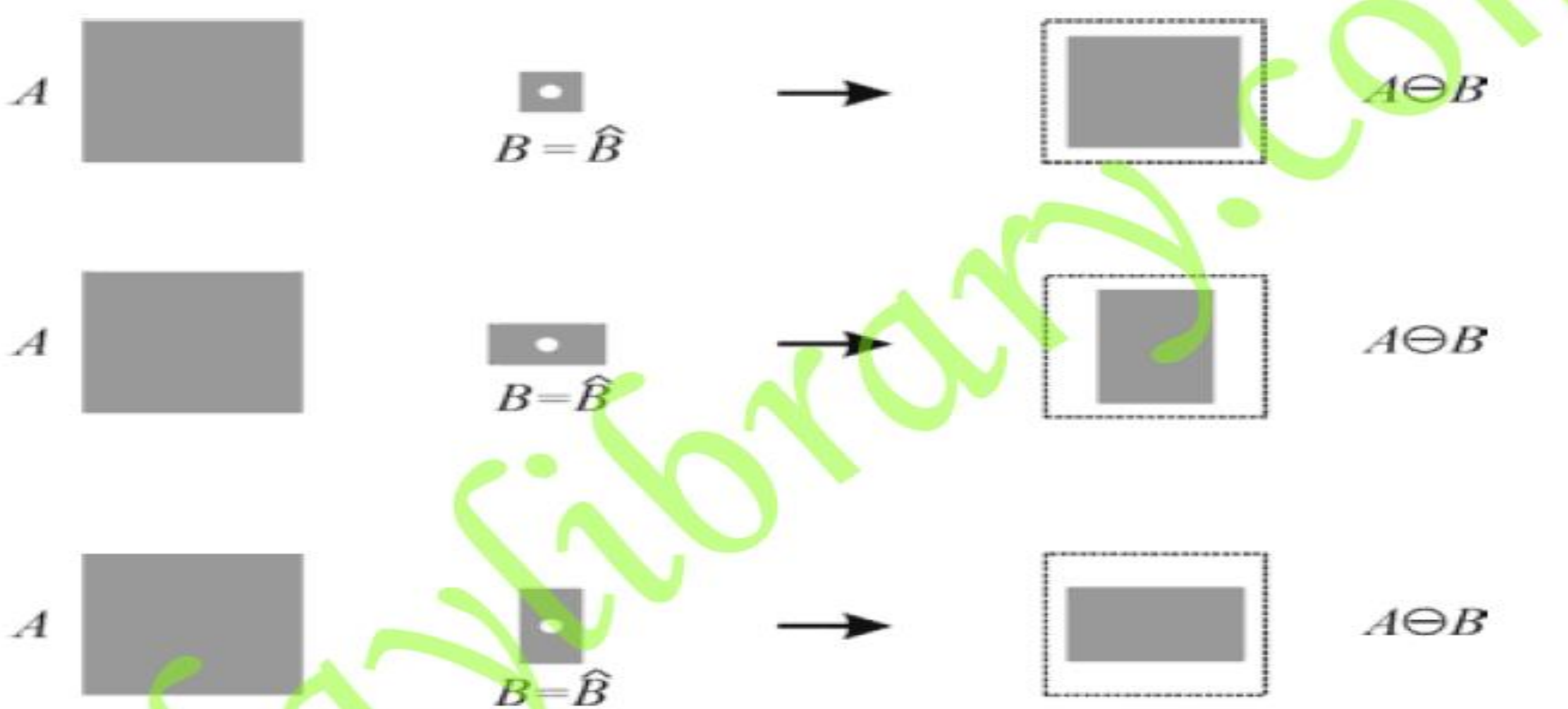


FIGURE 13.5 Example of erosion using three different rectangular structuring elements.

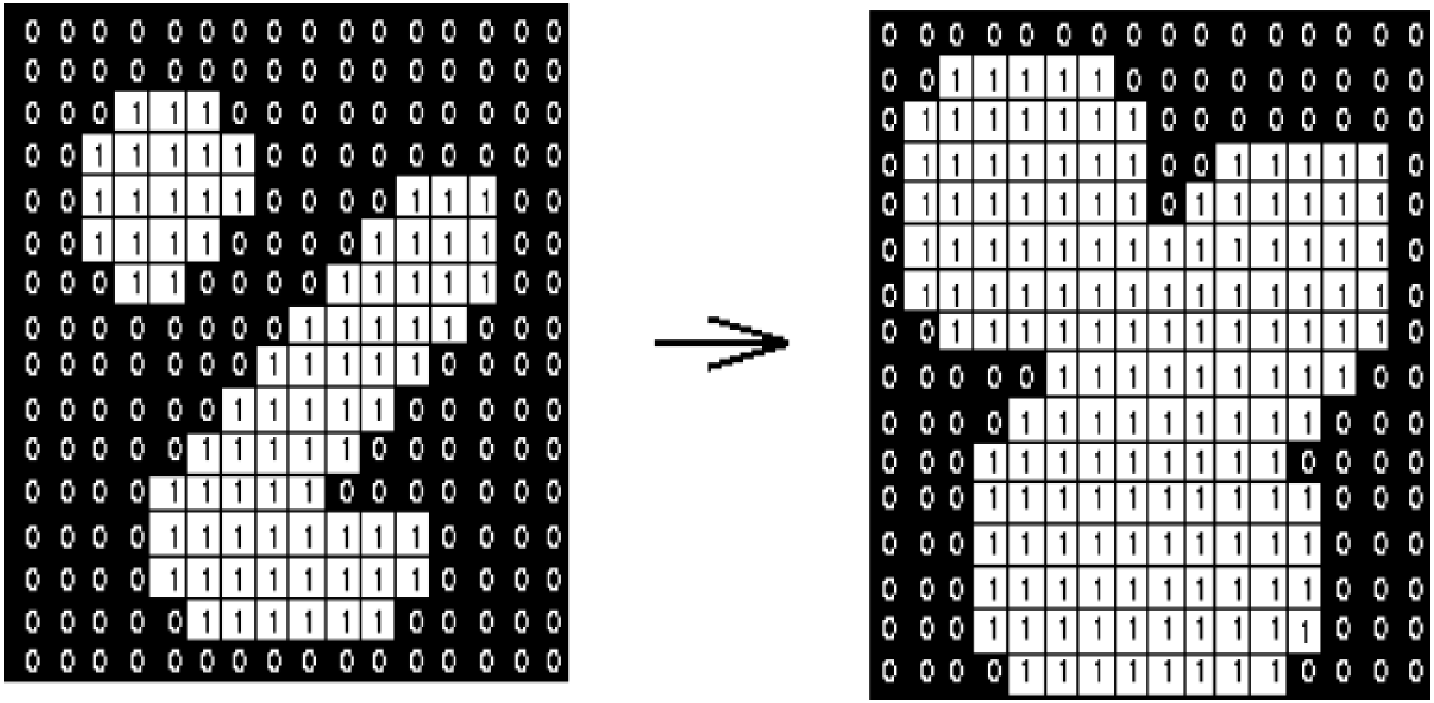
The **dilation** of an image f by a structuring element s (denoted $f \oplus s$) produces a new binary image $g = f \oplus s$ with ones in all locations (x,y) of a structuring element's orogin at which that structuring element s hits the the input image f, i.e. $g(x,y) = 1$ if s hits f and 0 otherwise, repeating for all pixel coordinates (x,y). Dilation has the opposite effect to erosion -- it adds a layer of pixels to both the inner and outer boundaries of regions.

Binary image

Dilation: a 2×2 square structuring element

http://documents.wolfram.com/applications/digitalimage/UsersGuide/Morphology/ImageProcessing6.3.html

The holes enclosed by a single region and gaps between different regions become smaller, and small intrusions into boundaries of a region are filled in:



Dilation: a 3×3 square structuring element
(www.cs.princeton.edu/~pshilane/class/mosaic/).

Results of dilation or erosion are influenced both by the size and shape of a structuring element. Dilation and erosion are *dual* operations in that they have opposite effects. Let f^c denote the complement of an image f, i.e., the image produced by replacing 1 with 0 and vice versa. Formally, the duality is written as

$$f \oplus s = f^c \ominus s_{\mathrm{rot}}$$

where s_{rot} is the structuring element s rotated by 180°. If a structuring element is symmetrical with respect to rotation, then s_{rot} does not differ from s. If a binary image is considered to be a collection of connected regions of pixels set to 1 on a background of pixels set to 0, then erosion is the fitting of a structuring element to these regions and dilation is the fitting of a structuring element (rotated if necessary) into the background, followed by inversion of the result.

Dilation is a morphological operation whose effect is to "grow" or "thicken" objects in a binary image. The extent and direction of this thickening are controlled by the size and shape of the structuring element. Mathematically, the dilation of a set *A* by *B*, denoted *A* ⊕ *B*, is defined as (*A* ⊕ *B*)

$$g'_{mn} = \bigvee_{m'=-R}^{R} \bigvee_{n'=-R}^{R} m_{m',n'} \wedge g_{m+m',n+n'}.$$

The dilation process is performed by laying the structuring element on the image and sliding it across the image in a manner similar to convolution. The difference is in the operation performed. It is best described in a sequence of steps:

1. If the origin of the structuring element coincides with a '0' in the image, there is no change; move to the next pixel.
2. If the origin of the structuring element coincides with a '1' in the image, perform the OR logic operation on all pixels within the structuring element.

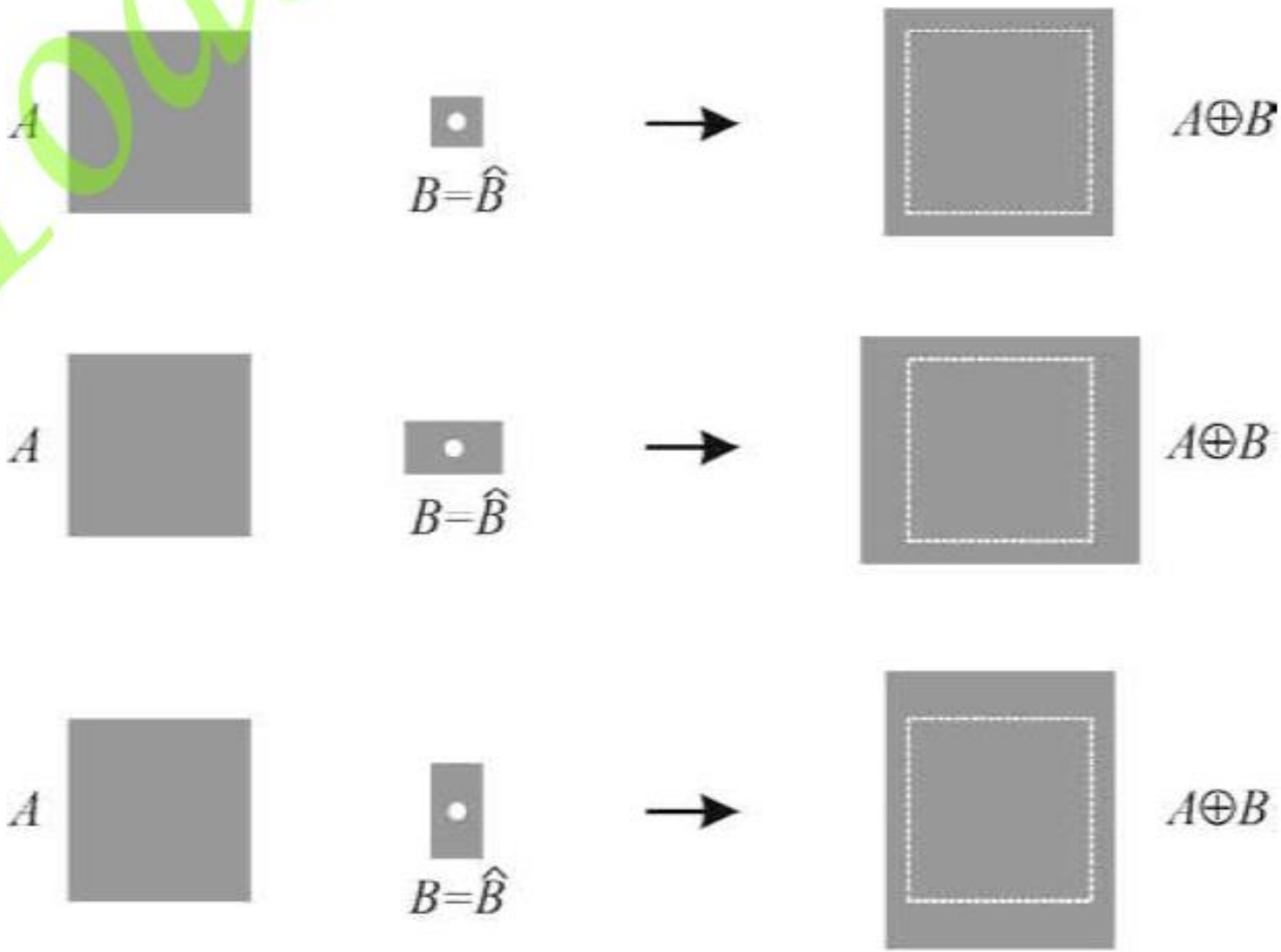


FIGURE 13.4 Example of dilation using three different rectangular structuring elements.

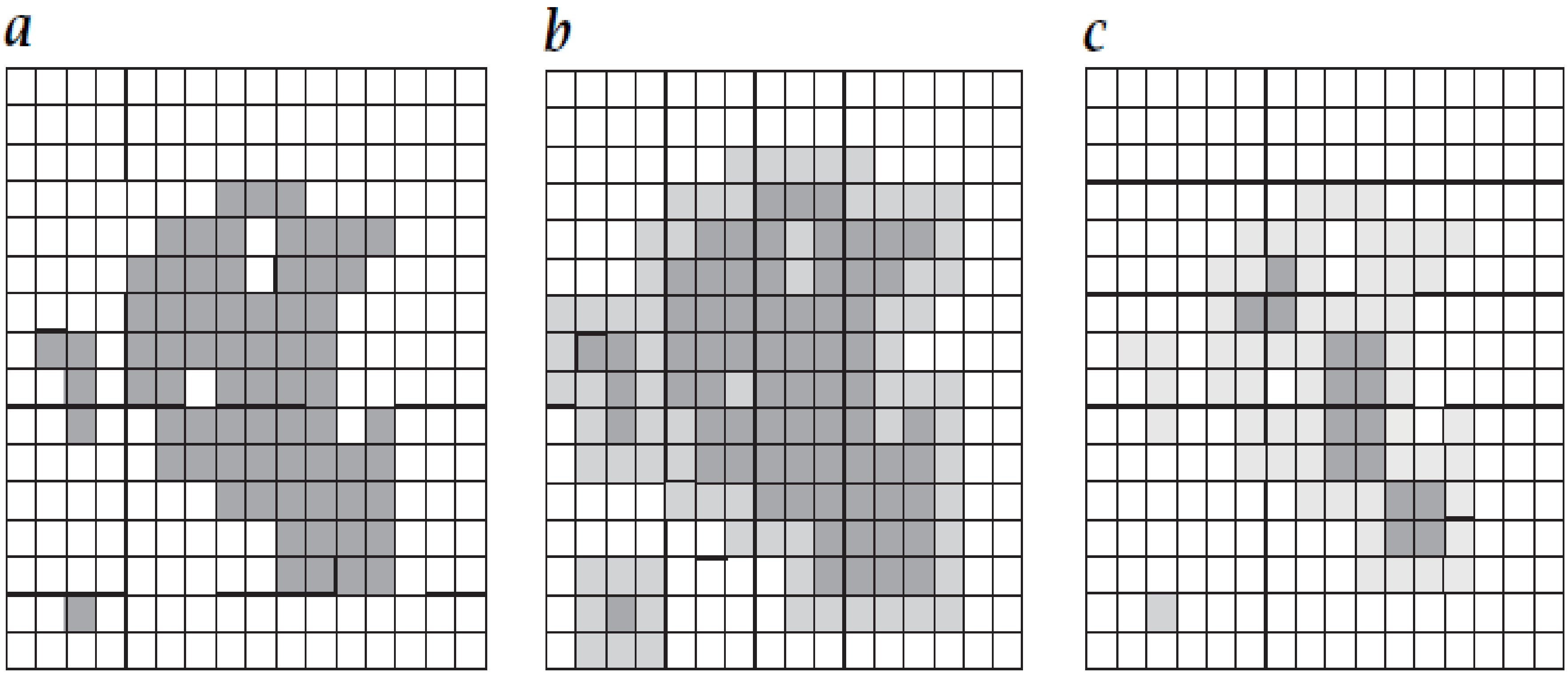


Figure 18.1: ***b*** *Dilation and* ***c*** *erosion of a binary object in* ***a*** *with a* 3×3 *mask. The removed (erosion) and added (dilation) pixels are shown in a lighter color.*

Operations on Sets

We used a rather unconventional way to introduce morphological operations. Normally, these operations are defined as operations on sets of pixels. We regard *G* as the set of all the pixels of the matrix that are not zero. *M* is the set of the non-zero mask pixels. With *Mp* we denote the mask shifted with its reference point (generally but not necessarily its center) to the pixel *p*.

Erosion is then defined as

$$G \ominus M = \{p : M_p \subseteq G\}$$

and dilation as

$$G \oplus M = \{p : M_p \cap G \neq \emptyset\}.$$

General Properties

Morphological operators share most but not all of the properties we have discussed for linear convolution operators . The properties discussed below are not restricted to 2-D images but are generally valid for *N*-dimensional image data.

1. Shift Invariance

***Shift invariance* results directly from the definition of the erosion and dilation operator as convolutions with binary data . Using the *shift operator S* and the operator notation, we can write the shift invariance of any morphological operator *M* as**

$$\mathcal{M}\left({}^{mn}S\boldsymbol{G}\right) = {}^{mn}S\left(\mathcal{M}\boldsymbol{G}\right).$$

2. Commutativity and Associativity

Morphological operators are generally not *commutative*:

$$\boldsymbol{M}_1 \oplus \boldsymbol{M}_2 = \boldsymbol{M}_2 \oplus \boldsymbol{M}_1, \text{ but } \boldsymbol{M}_1 \ominus \boldsymbol{M}_2 \neq \boldsymbol{M}_2 \ominus \boldsymbol{M}_1.$$

We can see that erosion is not commutative if we take the special case that $M1 \supset M2$. Then the erosion of $M2$ by $M1$ yields the empty set. However, both erosion and dilation masks consecutively applied in a cascade to the same image G are commutative:

$$
\begin{aligned}
(G \ominus M_1) \ominus M_2 &= G \ominus (M_1 \oplus M_2) &= (G \ominus M_2) \ominus M_1 \\
(G \oplus M_1) \oplus M_2 &= G \oplus (M_1 \oplus M_2) &= (G \oplus M_2) \oplus M_1.
\end{aligned}
$$

These equations are important for the implementation of morphological operations. generally, the cascade operation with k structure elements $M1, M2, \ldots, Mk$ is equivalent to the operation with the structure element $M = M1 \oplus M2 \oplus \ldots \oplus Mk$. In conclusion, we can decompose large structure elements in the very same way as we decomposed linear shift-invariant operators. An important example is the composition of separable structure elements by the horizontal and vertical elements $M = Mx \oplus My$. Another less trivial example is the construction of large one-dimensional structure elements from structure elements including many zeros:

$$
\begin{gathered}
[1\ 1\ 1] \oplus [1\ 0\ 1] = [1\ 1\ 1\ 1\ 1] \\
[1\ 1\ 1\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 1] = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1] \\
[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1] \oplus [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1] \\
= [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]
\end{gathered}.
$$

3. Monotony

Erosion and dilation are monotonic operations

$$
\begin{array}{lll}
G_1 \subseteq G_2 & \rightsquigarrow & G_1 \oplus M \subseteq G_2 \oplus M \\
G_1 \subseteq G_2 & \rightsquigarrow & G_1 \ominus M \subseteq G_2 \ominus M.
\end{array}
$$

Monotony **means that the subset relations are invariant with respect to erosion and dilation**

4. Distributivity

Linear shift-invariant operators are ***distributive*** **with regard to addition. The corresponding distributivities for erosion and dilation with respect to the union and intersection of two images** ***G*1 and *G*2 are more complex:**

$$
\begin{array}{lll}
(G_1 \cap G_2) \oplus M & \subseteq & (G_1 \oplus M) \cap (G_2 \oplus M) \\
(G_1 \cap G_2) \ominus M & = & (G_1 \ominus M) \cap (G_2 \ominus M)
\end{array}
$$

$$
\begin{array}{lll}
(G_1 \cup G_2) \oplus M & = & (G_1 \oplus M) \cup (G_2 \oplus M) \\
(G_1 \cup G_2) \ominus M & \supseteq & (G_1 \ominus M) \cup (G_2 \ominus M).
\end{array}
$$

Erosion is distributive over the intersection operation, while dilation is distributive over the union operation.

5. Duality

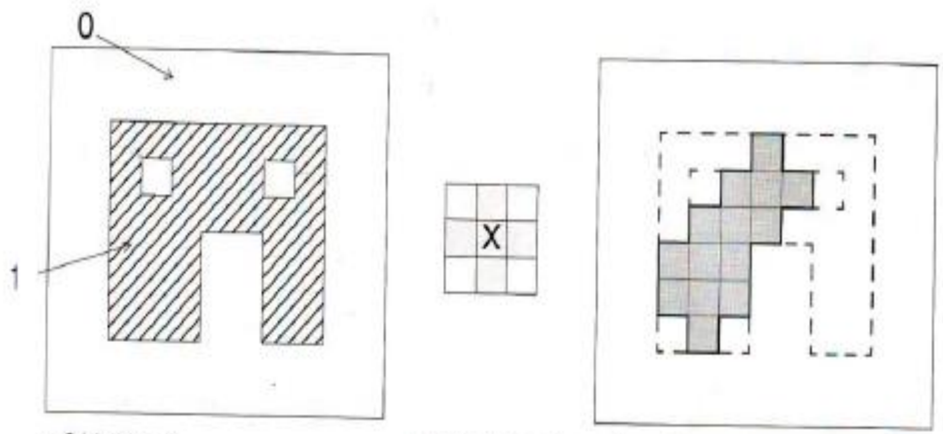
Erosion and dilation are *dual operators*. By negating the binary image, erosion is converted to dilation and vice versa.

$$\begin{aligned} \overline{G} \ominus M &= \overline{G \oplus M} \\ \overline{G} \oplus M &= \overline{G \ominus M}. \end{aligned}$$

❖ Composite Morphological Operators

➢ Opening and Closing

These two basic operations, dilation and erosion, can be combined into more complex sequences. The most useful of these for morphological filtering are called opening and closing. Opening consists of an erosion followed by a dilation and can be used to eliminate all pixels in regions that are too small to contain the structuring element. See Figure below for an example of opening.



a. Original image.

b. Structural element; x = origin.

c. Image after opening; erosion followed by dilation.

❖Opening

The opening sieves out all objects which at no point completely contain the structure element, but avoids a general shrinking of object size .

It is also an ideal operation for removing lines with a thickness smaller than the diameter of the structure element. Note also that the object boundaries become smoother.

The opening operation is *idempotent*, that is, once an image has been opened with a certain SE, subsequent applications of the opening algorithm with the same SE will not cause any effect on the image.

The erosion operation is useful for removing small objects. However, it has the disadvantage that all the remaining objects shrink in size . We can avoid this effect by dilating the image after erosion with the same structure element. This combination of operations is called an *opening operation*

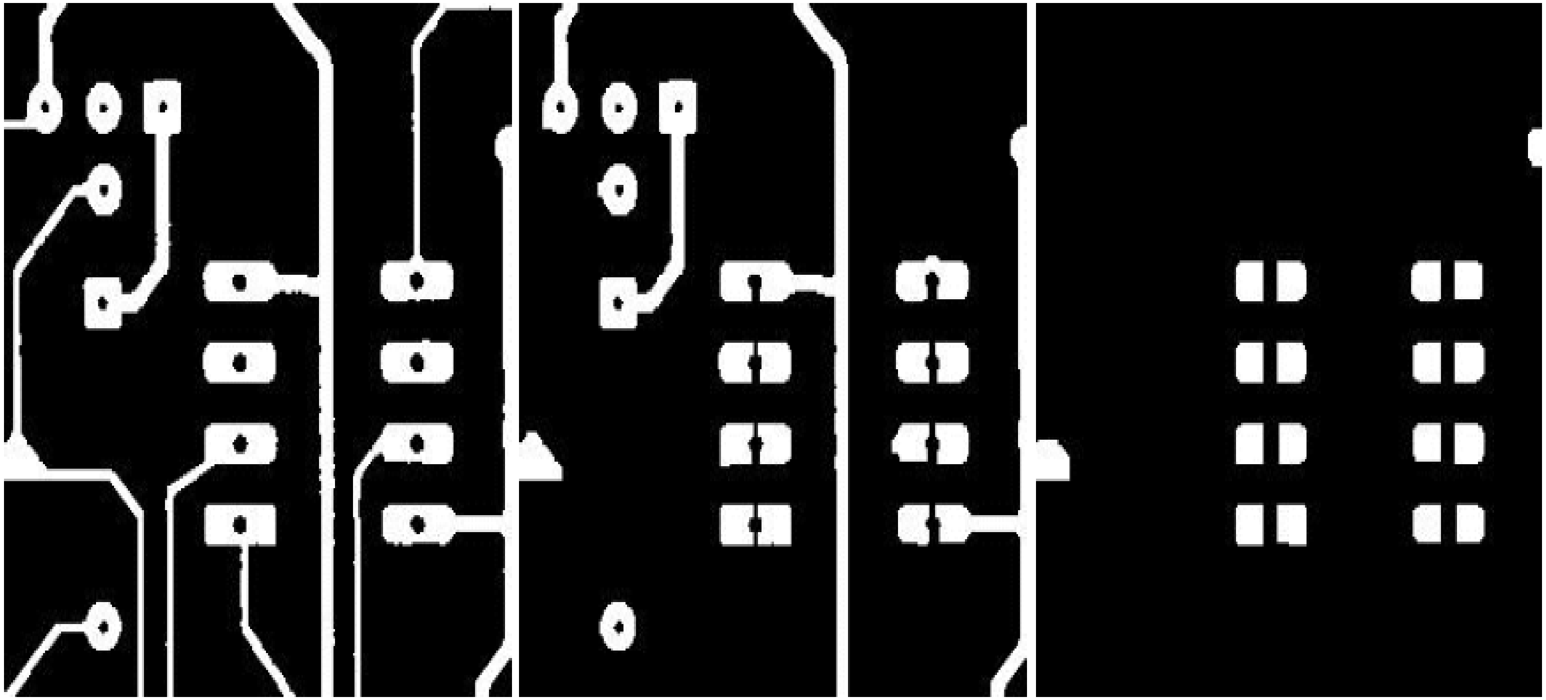
$$G \circ M = (G \ominus M) \oplus M.$$

Binary image

Opening: a 2×2 square structuring element

http://documents.wolfram.com/applications/digitalimage/UsersGuide/Morphology/ImageProcessing6.3.html

Opening is so called because it can open up a gap between objects connected by a thin bridge of pixels. Any regions that have survived the erosion are restored to their original size by the dilation:



Binary image f | $f \circ s$ (5×5 square) | $f \circ s$ (9×9 square)

Results of opening with a square structuring element (www.mmorph.com/html/morph/mmopen.html/).

Opening is an **idempotent** operation: once an image has been opened, subsequent openings with the same structuring element have no further effect on that image:

$$(f \circ s) \circ s) = f \circ s.$$

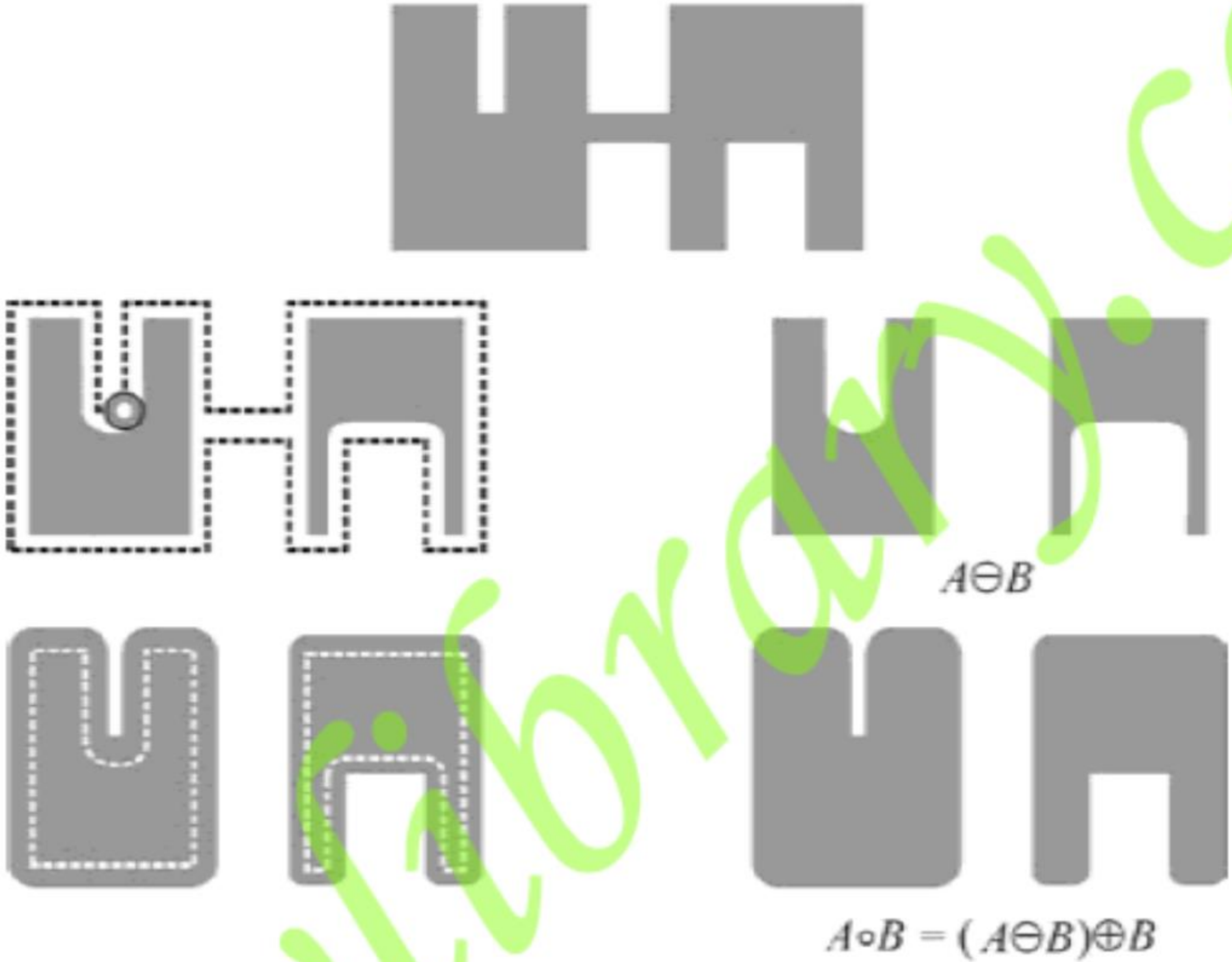


FIGURE 13.6 Example of morphological opening.

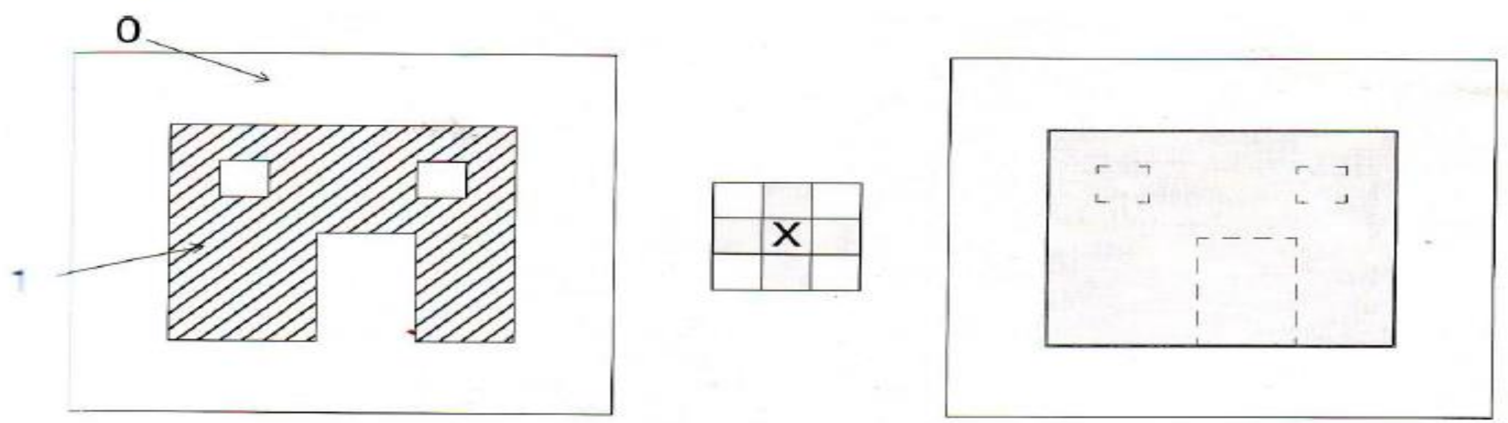
❖ Closing

In contrast, the dilation operator enlarges objects and closes small holes and cracks. General enlargement of objects by the size of the structure element can be reversed by a following erosion . This combination of operations is called a *closing operation*

$$G \bullet M = (G \oplus M) \ominus M.$$

Closing consists of a dilation followed by erosion and can be used to fill in holes and small gaps. In Figure below we see that the closing operation has the effect of filling in holes and closing gaps.

Closing



a. Original image.

b. Structural element; x = origin.

c. Image after closing; dilation followed by erosion; original in dashes.

Just as with opening, the closing operation is *idempotent*, that is, once an image has been closed with a certain SE, subsequent applications of the opening algorithm with the same SE will not cause any effect on the image. Mathematically,

The area change of objects with different operations may be summarized by the following relations:

$$G \ominus M \subseteq G \circ M \subseteq G \subseteq G \bullet M \subseteq G \oplus M.$$

Opening and closing are *idempotent operations*:

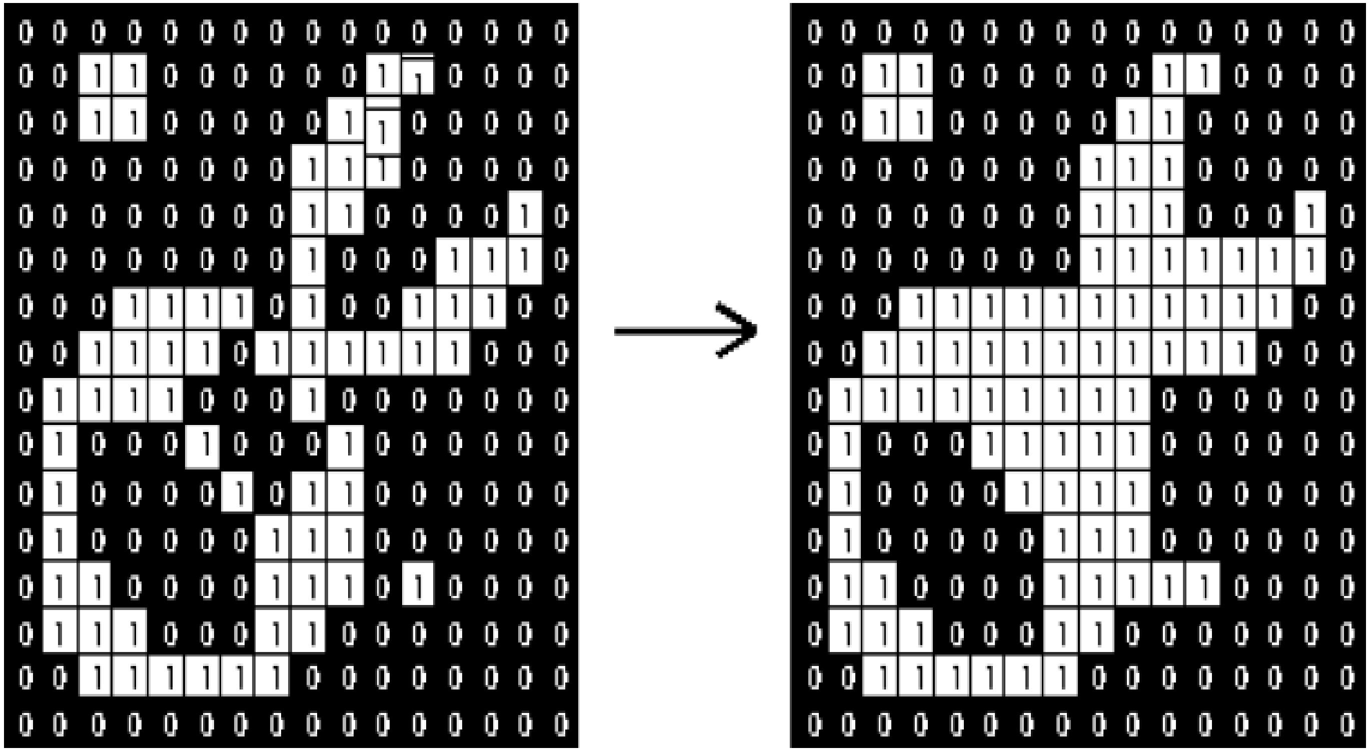
$$\begin{aligned} G \bullet M &= (G \bullet M) \bullet M \\ G \circ M &= (G \circ M) \circ M, \end{aligned}$$

Binary image

Closing: a 2×2 square structuring element

http://documents.wolfram.com/applications/digitalimage/UsersGuide/Morphology/ImageProcessing6.3.html

In this case, the dilation and erosion should be performed with a rotated by 180° structuring element. Typically, the latter is symmetrical, so that the rotated and initial versions of it do not differ.



Closing with a 3×3 square structuring element (www.cs.princeton.edu/~pshilane/class/mosaic/).

Closing is so called because it can fill holes in the regions while keeping the initial region sizes. Like opening, closing is idempotent: $(f \bullet s) \bullet s = f \bullet s$, and it is dual operation of opening (just as opening is the dual operation of closing):

Morphological closing is typically used to fill small holes, fuse narrow breaks, and close thin gaps in the objects within an image, without changing the objects' size (as dilation would have done). It also causes a smoothening of the object's contour (Figure 13.8).

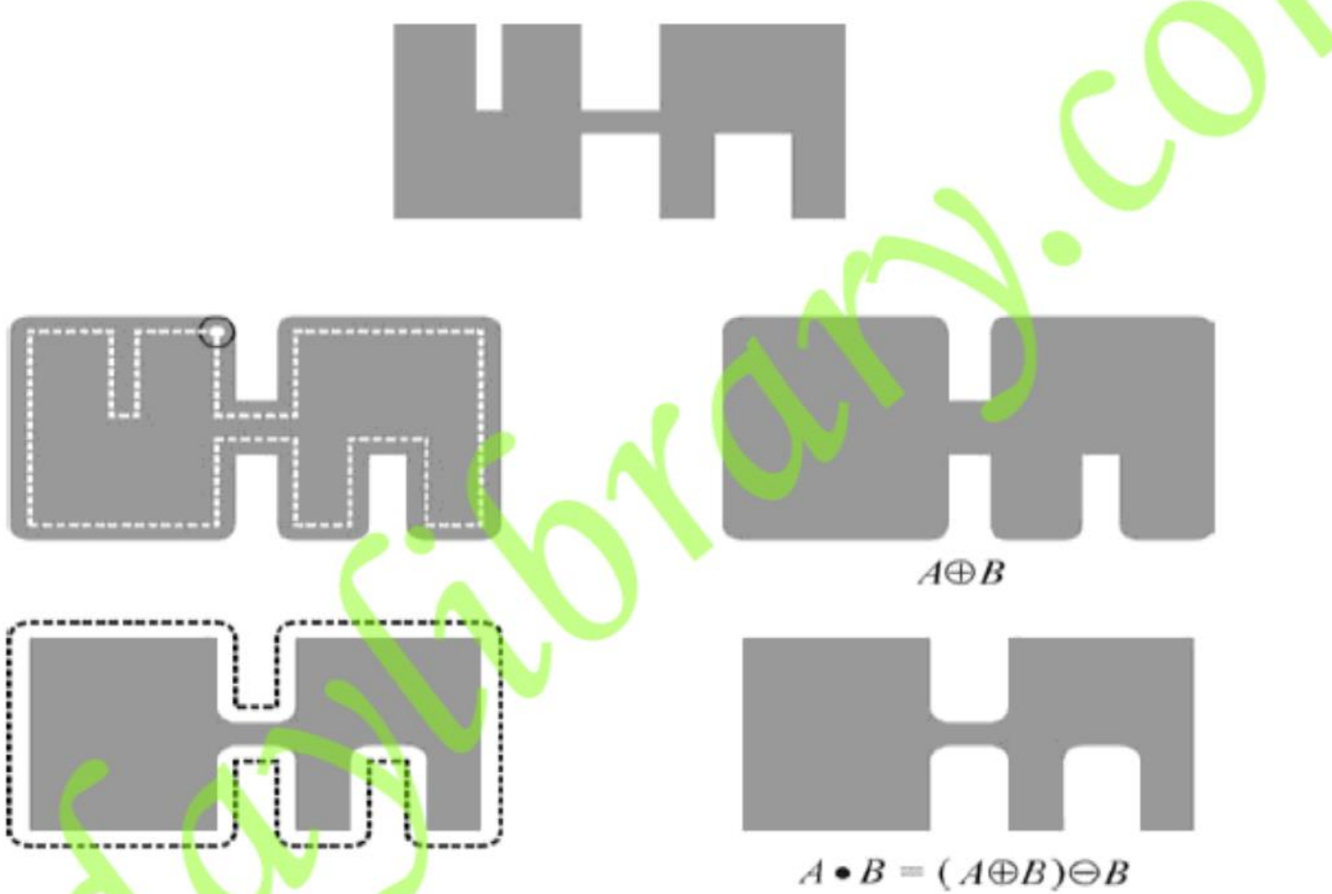


FIGURE 13.8 Example of morphological closing.

The geometric interpretation of the closing operation is as follows: $A \bullet B$ is the complement of the union of all translations of B that do not overlap A (Figure 13.9). Closing is the dual operation of opening and vice versa:

$$A \bullet B = (A^c \circ B)^c$$

$$A \circ B = (A^c \bullet B)^c$$

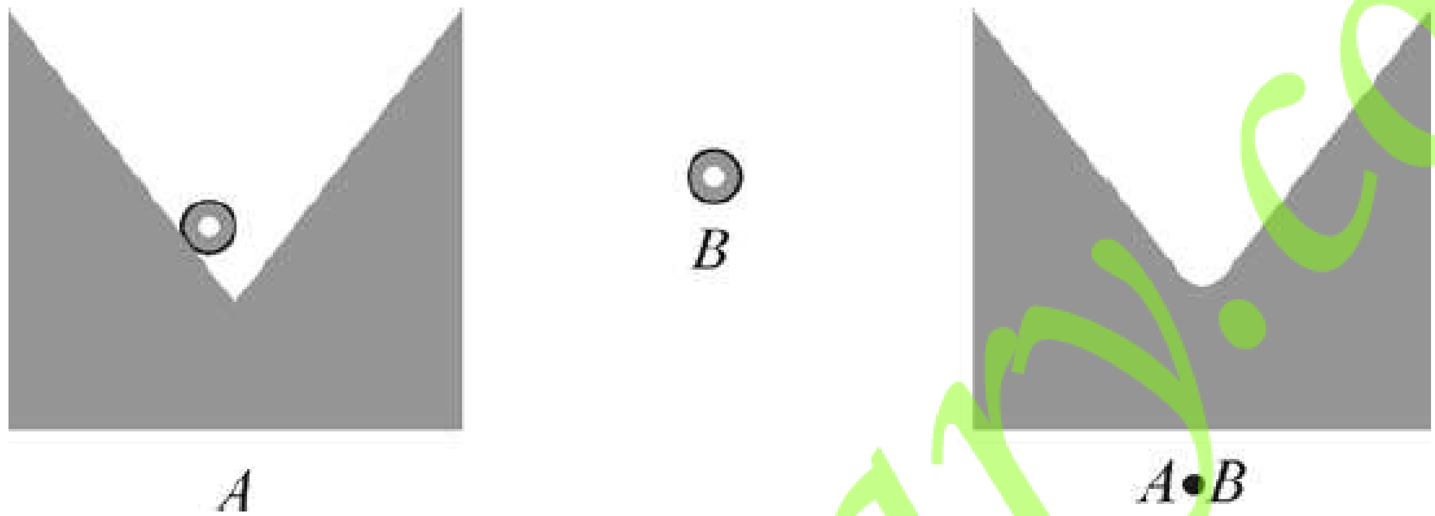


FIGURE 13.9 Geometric interpretation of the morphological closing operation. Adapted and redrawn from [GW08].