7. Morphological operations on binary images

7.1. Introduction

Morphological operations are affecting the form, structure or shape of an object. They are usually applied on binary images (black & white images – images with only 2 colors: *black* and *white*). They are used in pre- or post- processing (filtering, thinning, and pruning) or for getting a representation or description of the shape of objects/regions (boundaries, skeletons convex hulls).

7.2. Theoretical considerations

The two principal morphological operations are *dilation* and *erosion* [1]. 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.

Notations:

Object / foreground pixels: pixels of interest (on which the morphological operations are applied)

Background pixels: the complementary set of the object / foreground pixels

7.2.1. The dilation

The *dilation* process is performed by laying the structuring element **B** on the image **A** and sliding it across the image in a manner similar to convolution (will be presented in a next laboratory). 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 'background' pixel in the image, there is no change; move to the next pixel.
2. If the origin of the structuring element coincides with an 'object' pixel in the image, make (label) all pixels from the image covered by the structuring element as ‘object’ pixels.

Notation:

$A \oplus B$

The structuring element can have any shape. Typical shapes are presented below:

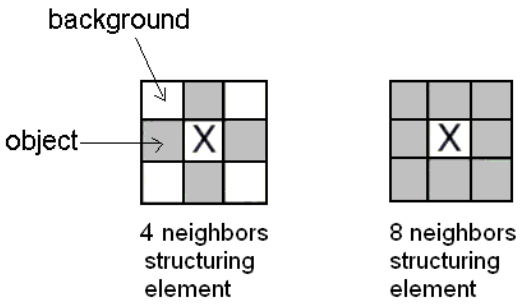


Fig. 7.1 Typical shapes of the structuring elements (B)

An example is shown in Fig. 7.2. Note that with a dilation operation, all the 'object' pixels in the original image will be retained, any boundaries will be expanded, and small holes will be filled.

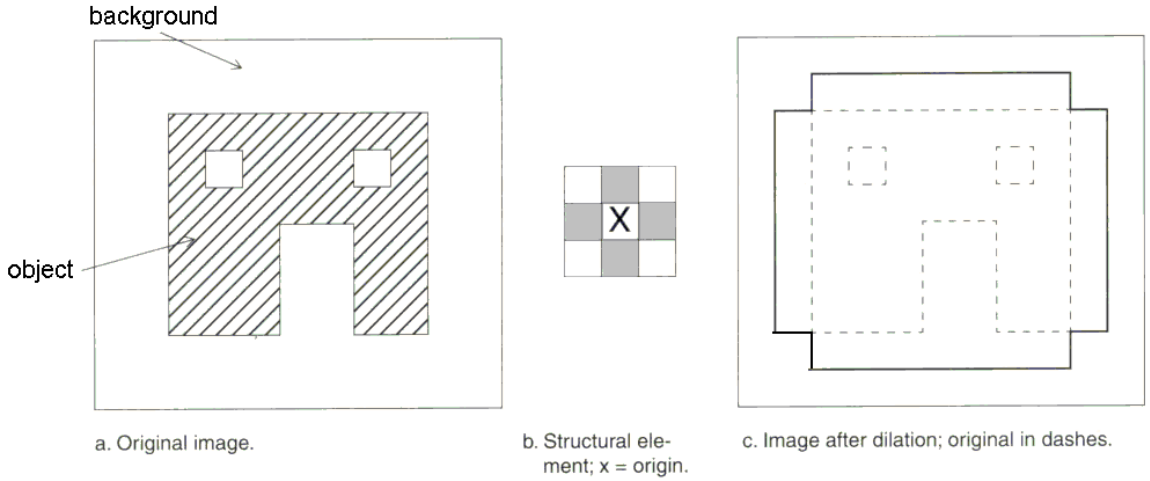


Fig. 7.2 Illustration of the dilatation process

a. b.

Fig. 7.3 Example of the dilation (object = black / background = white): a. Original image **A**; b. The result image: $\mathbf{A} \oplus \mathbf{B}$.

7.2.2. The erosion

The *erosion* process is similar to dilation, but we turn pixels to 'background', not 'object'. As before, slide the structuring element across the image and then follow these steps:

1. If the origin of the structuring element coincides with a 'background' pixel in the image, there is no change; move to the next pixel.

2. If the origin of the structuring element coincides with an 'object' pixel in the image, and any of the 'object' pixels in the structuring element extend beyond the 'object' pixels in the image, then change the current 'object' pixel in the image (above you have positioned the structuring element center) to a 'background' pixel.

Notation:

$A \ominus B$

In Fig. 7.4, the only remaining pixels are those that coincide to the origin of the structuring element where the entire structuring element was contained in the existing object.

Because the structuring element is 3 pixels wide, the 2-pixel-wide right leg of the image object was eroded away, but the 3-pixel-wide left leg retained some of its center pixels.

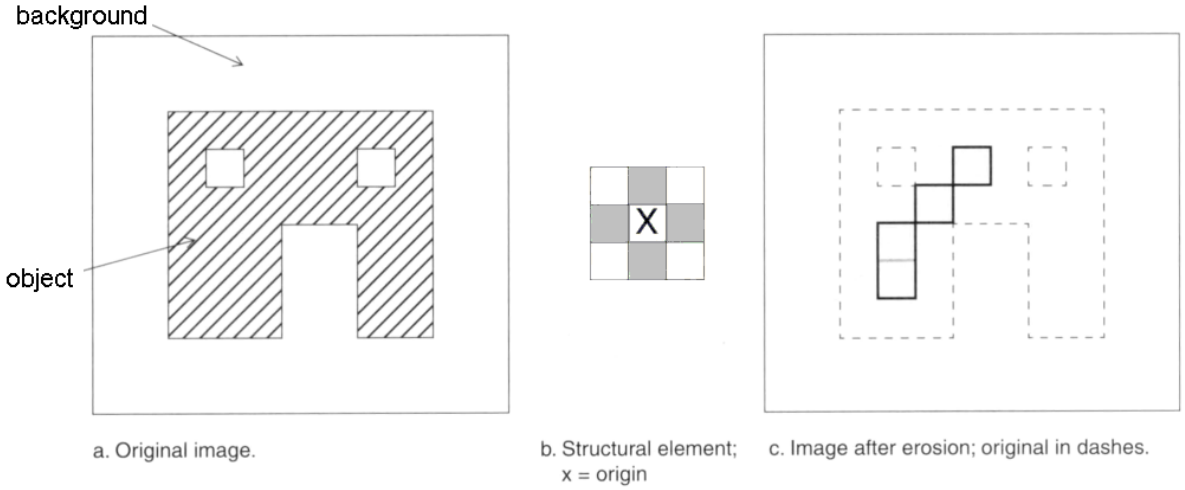
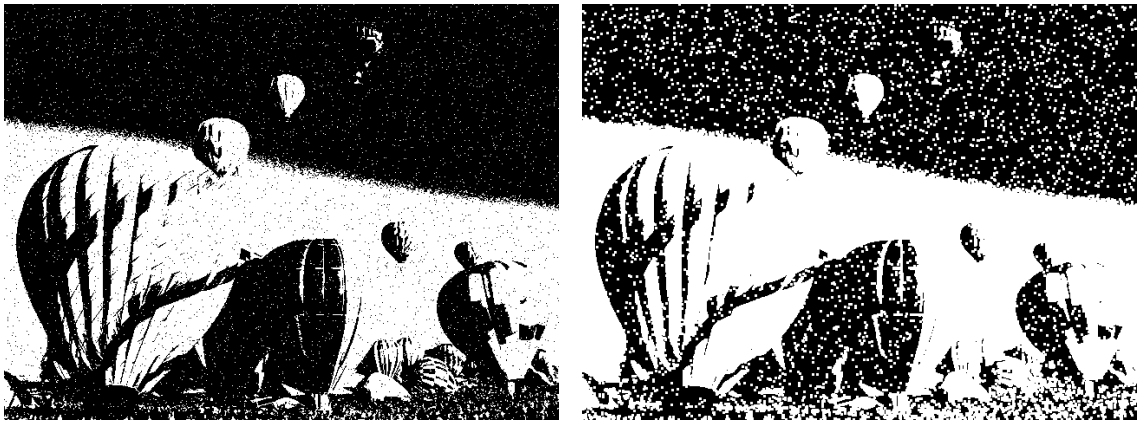


Fig. 7.4 Illustration of the erosion process



a. b.

Fig. 7.5 Example of the erosion (object = black / background = white): a. Original image **A**; b. The result image: **A** Θ **B**.

7.2.3. 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 [1]. *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. In this case the structuring element is often called a probe, because it is probing the image looking for small objects to filter out of the image. See Fig. 7.6 for the illustration of the opening process.

Notation:

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

Closing consists of a dilation followed by erosion and can be used to fill in holes and small gaps. In Fig. 7.7 we see that the closing operation has the effect of filling in holes and closing gaps. Comparing the left and right images from Fig. 7.8, we see that the order of

operation is important. Closing and opening will generate different results even though both consist of erosion and dilation.

Notation:

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

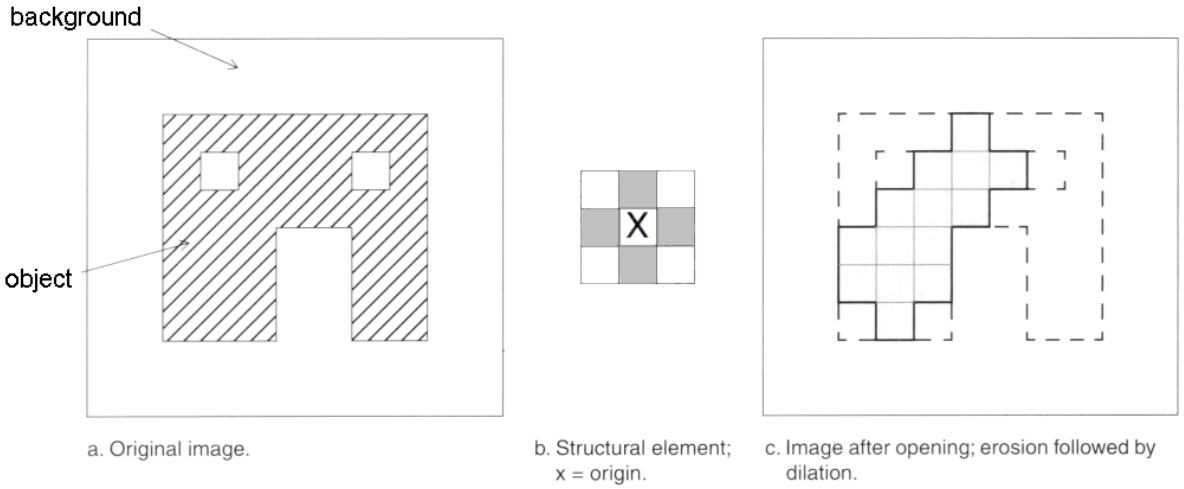


Fig. 7.6 Illustration of the opening process

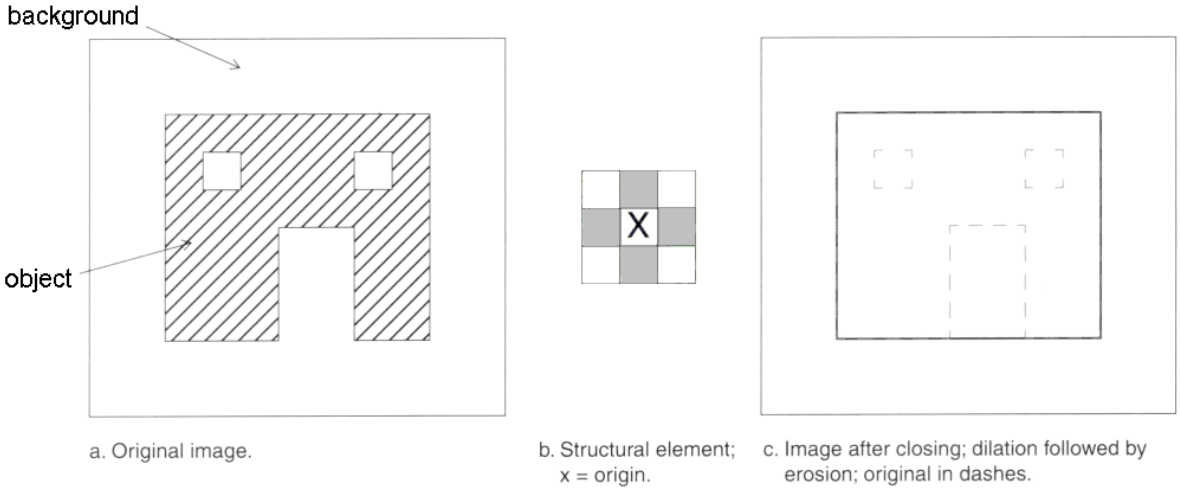


Fig. 7.7 Illustration of the closing process

a. b.

Fig. 7.8 Results of the opening (a) and closing (b) operations applied on the original image from Fig. 7.5a (object = black / background = white).

7.2.4. Some basic morphological algorithms [2]

7.2.4.1. Boundary extraction

The boundary of a set A, denoted by $\beta(A)$, can be obtained by first eroding A by B and then performing the set differences between A and its erosion. That is,

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

where

B is a suitable structuring element.

'–' is the difference operation on sets (illustrated in Fig. 7.10)

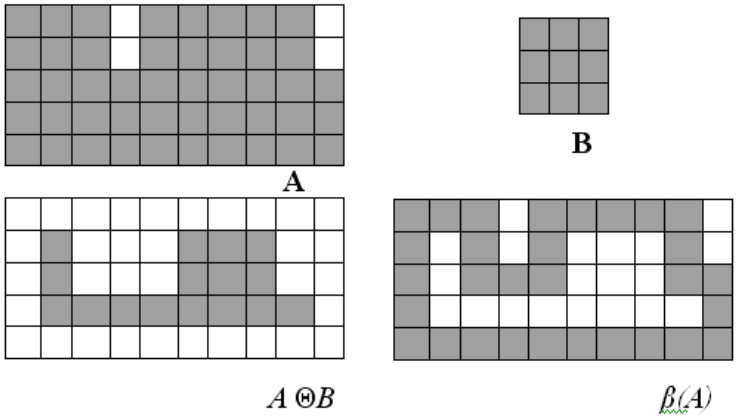


Fig. 7.9 Illustration of the boundary extraction algorithm

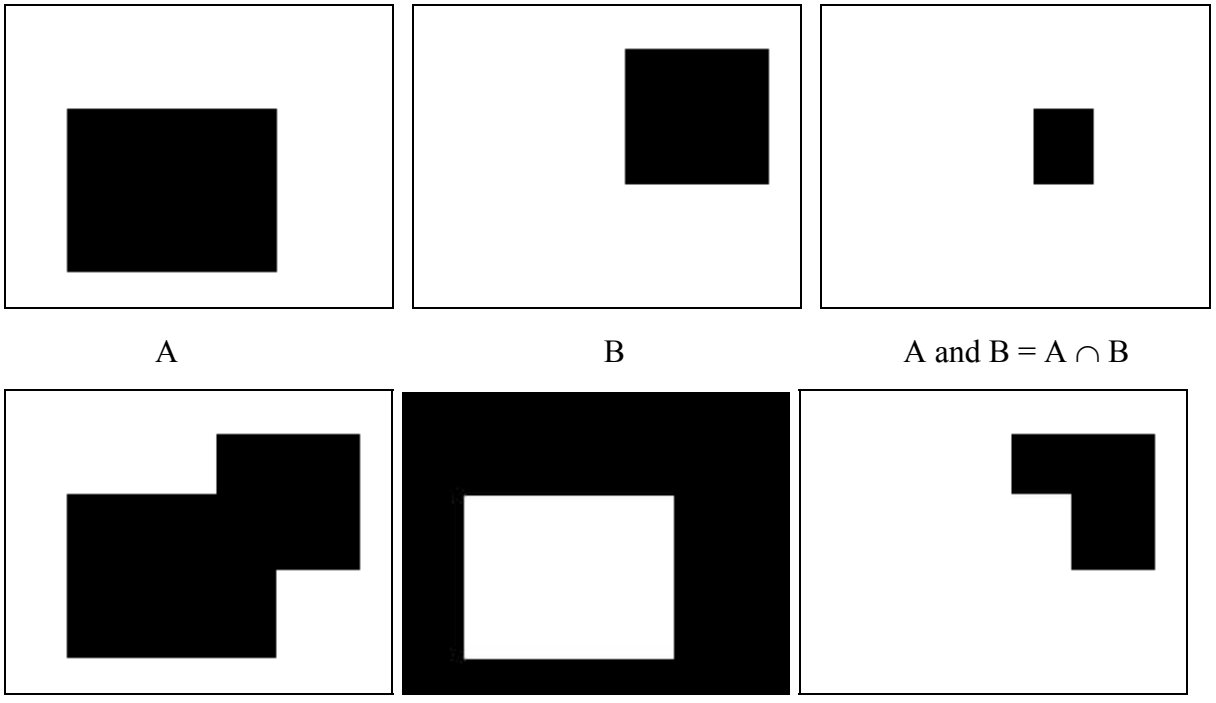


Fig. 7.10 Illustration of the main operations on sets

7.2.4.2. Region filling

Next we develop a simple algorithm for region filling based on set dilations, complementation, and intersections.

Beginning with a point p inside the boundary, the objective is to fill the entire region with 'object' pixels. If we adopt the convention that all non-boundary points are labeled 'background', then we assign the value/label 'object' to p to begin. The following procedure then fills the region with 'object' pixels:

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

where

$X_0=p$,

B is the symmetric structuring element

$\cap$ - is the intersection operator (see Fig. 7.10)

A^C – is the complement of set A (see Fig. 7.10)

The algorithm terminates at iteration step k if $X_k=X_{k-1}$. The set union of X_k and A contains the filled set and its boundary.

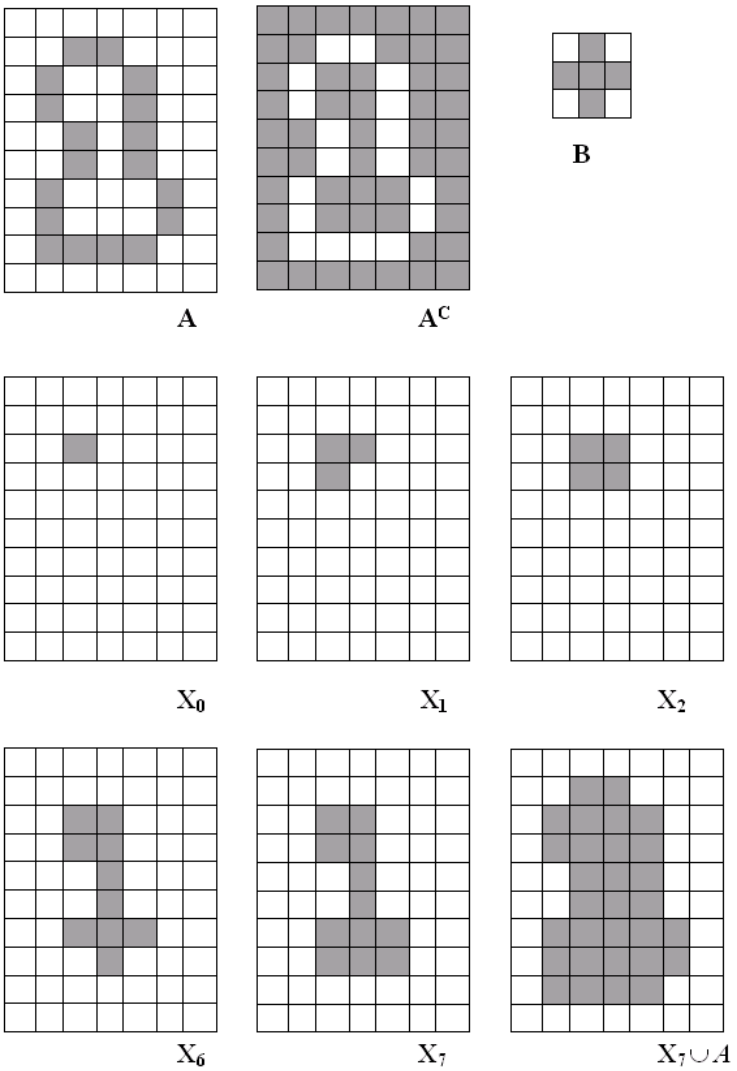


Fig. 7.11 Illustration of the region filling algorithm

7.3. Implementation hints

7.3.1. Using a supplementary image buffer for chain processing

The results of the basic morphological operations (dilation and erosion) should be applied in the following manner:

Destination image = *Source image* (operator) *Structuring element*

The source image shouldn't be affected in any way!

For the implementation of the combined morphological operations (opening and closing) or of the repeated operations (for example: *n* consecutive erosions) in a single processing function a supplementary image buffer should be created and used.

7.4. Practical work

1. Add to the OpenCV Application framework processing functions which implement the basic morphological operations.
2. Add the facility to apply the morphological operations repeatedly (*n* times). Input the value of *n* from the command line. Remark the 'idempotency' property of the opening/closing operations (therefore there is no use to apply them repeatedly).
3. Implement the boundary extraction algorithm.
4. Implement the region filling algorithm.
5. **Save your work. Use the same application in the next laboratories. At the end of the image processing laboratory you should present your own application with the implemented algorithms!!!**

References

[1]. Umbaugh Scot E, *Computer Vision and Image Processing*, Prentice Hall, NJ, 1998, ISBN 0-13-264599-8

[2] R.C.Gonzales, R.E.Woods, *Digital Image Processing. 2-nd Edition*, Prentice Hall, 2002.