Ioannis Pitas

Digital Image Processing Fundamentals

Chapter 7
Shape description

Lab exercises in EIKONA

Thessaloniki 1998

Chapter 7: Shape description.

Exercise 7.1: Object characteristics.

We can obtain the characteristics of a binary image by selecting the menu option “***Black and White→Analysis→Shape Analysis→Find Char***“. Let us suppose that the binary image RECT.RAW is loaded in BW buffer 0. We choose the ROI to be the area surrounding the rectangle. By selecting the menu option mentioned above and specifying BW buffer 0 as the source buffer and 0 as the threshold value, EIKONA will respond with a pop-up window which gives information regarding the characteristics of the first object met in a row-wise manner from the upper left corner (the rectangle in our example) inside the defined ROI.

By applying this procedure on image RECT.RAW, we get the following results:

Object area: 20400.000000
Center of gravity X: 127.000000
Center of gravity Y: 77.000000
Inertia axis angle: 0.005437
Object volume: 20400.0000000

Exercise 7.2: Chain coding.

In this exercise we perform chain coding on an image.We assume that the image SHAPE.RAW is loaded in BW buffer 0. We choose the ROI to be the area surrounding the rectangle. Next, we choose the menu option “***Black and White→Analysis→Shape Analysis→Chain Coding***“ and in the dialog box that appears ,we specify BW buffer 0 as the source buffer, **<NewBuffer>** as the destination buffer and a moderate value for the maximum number of chain nodes (in this case 2000). After pressing the “OK” button the output image is displayed. It is obvious that chain coding can be used as another edge detection algorithm for binary images. Figure 1 depicts the result of chain coding of the image SHAPE.RAW.

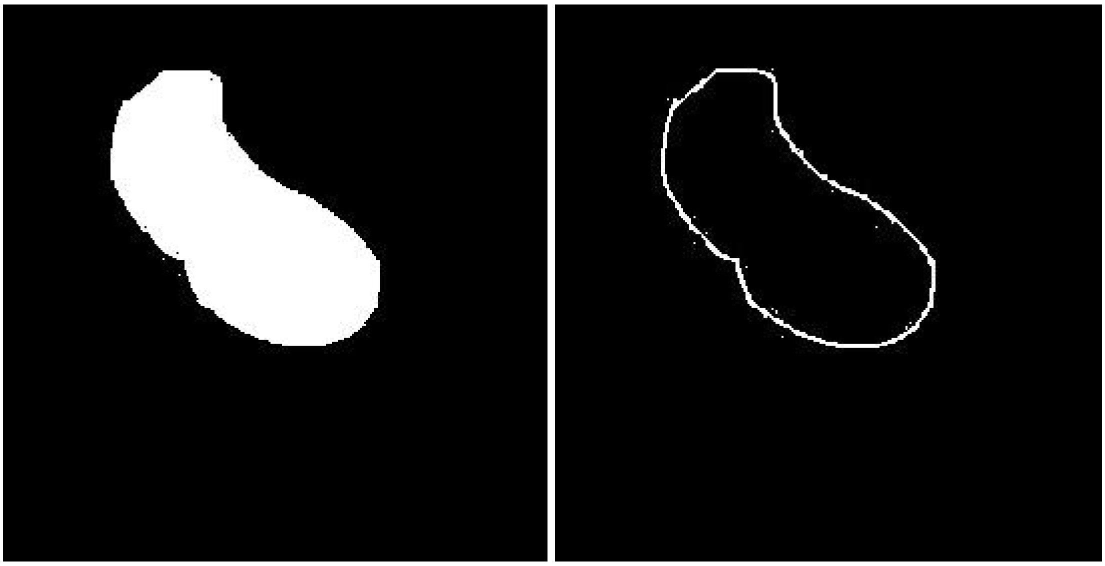


Figure 1: Original and chain-coded image of SHAPE.RAW.

Exercise 7.3: Image pyramid.

In this example, we use EIKONA to compute the pyramid of an image. Pyramids are commonly used in image compression and analysis techniques. We assume that the thresholded version of BABOON is loaded in BW buffer 0. If we select the menu option "***Black and White→Analysis→Pyramid***", we are prompted with a dialog box. By specifying the appropriate input and output image buffers we come up with the result shown in Figure 2b, which shows the pyramid of BABOON.

Figure 2: Thresholded BABOON and its pyramid.

Exercise 7.4: One–pass thinning.

In this exercise, we demonstrate the use of the one–pass thinning algorithm. In general, thinning algorithms recursively delete pixels in a binary image where transitions from 0 to 1 occur, until a set of unit-width connected lines appears.

We threshold the image “circuit” before thinning. To perform one-pass thinning, we select the menu option “***Black and White→Analysis→Thinning→One Pass Thinning”*** from EIKONA’s menu. We should note that, in one-pass thinning, the input and output image buffers are the same. As a result, we must be very careful when using it, because our original image can be destroyed. An example of one–pass thinning is given in Figure 3.

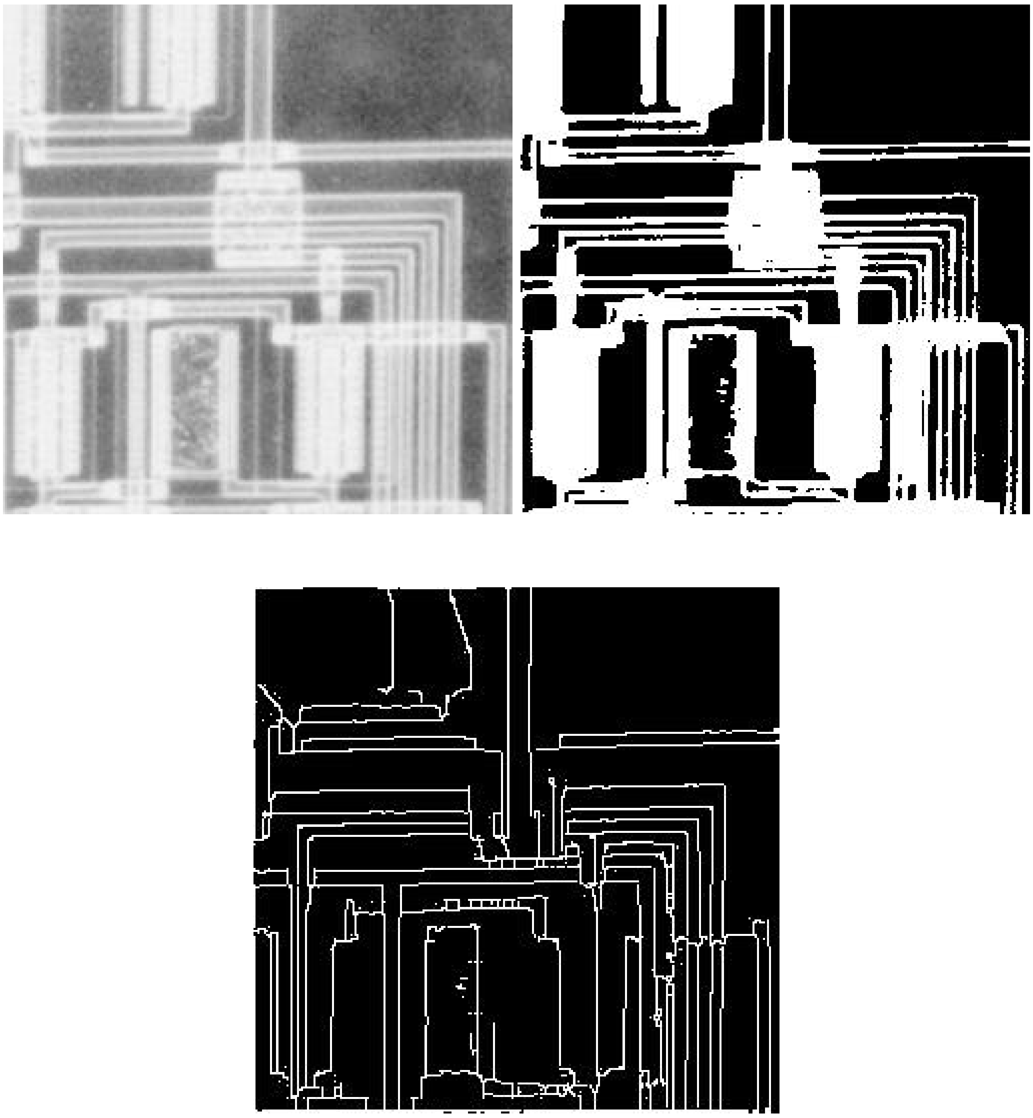


Figure 3: (a) The grayscale image “Circuit”.
(b) Thresholded version of image “Circuit” (Threshold=200).
(c) Result of one–pass thinning on the binary image of Figure 3b.

Exercise 7.5: Two–pass thinning.

The one–pass thinning algorithm has the disadvantage of not thinning image objects symmetrically. Moreover, one-pass thinning is not efficient in case of images that contain relatively large and curved objects. A more efficient algorithm is two-pass thinning.

We threshold the image “circuit” before thinning. The two–pass thinning algorithm is accessible through the menu option “***Black and White→Analysis→Thinning→Two Pass Thinning”***. In the dialog box that appears, we specify the appropriate Source and Destination buffers and then we press the “OK” button. The result of two-pass thinning on the image of Figure 3b is shown in Figure 4.

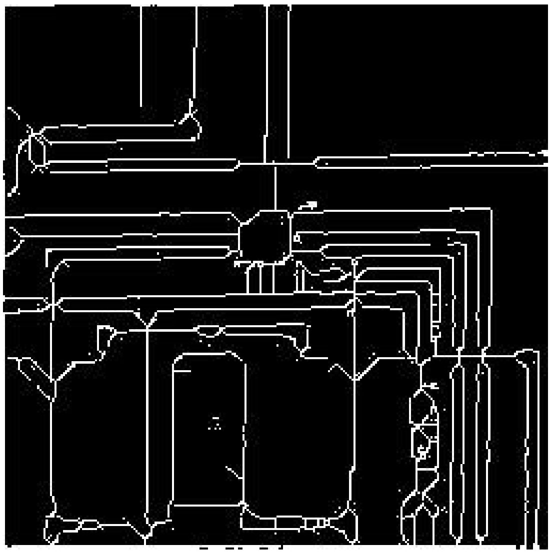


Figure 4: Result of two-pass thinning on the binary image of Figure 3b.

Exercise 7.6: Morphological operations.

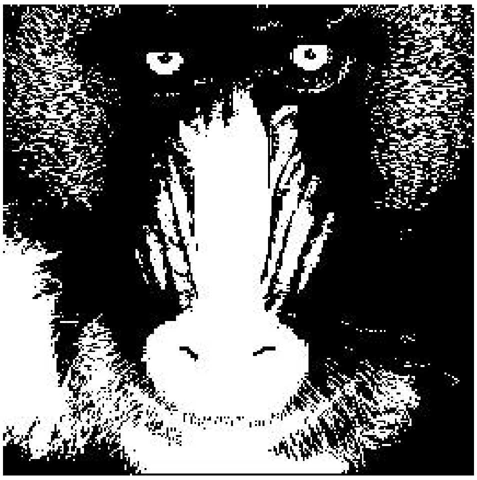
Many morphological operations are also supported in EIKONA. These include both binary and grayscale morphological operators. In this exercise we present examples of binary opening, skeleton extraction and top hat transformation.

Section 7.6.1: Binary opening and closing.

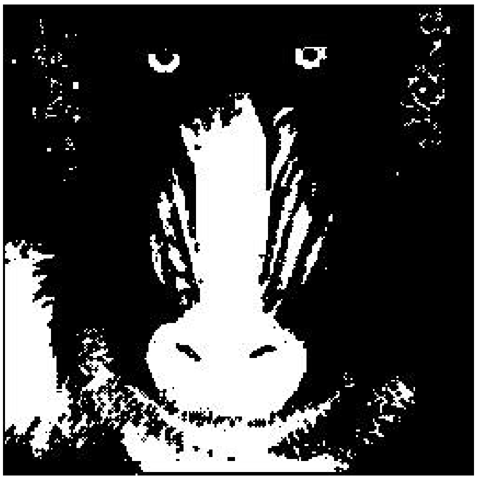
Assuming that the thresholded binary image of BABOON, shown in Figure 5a, is loaded in BW buffer 1, we can perform a binary opening operation by performing a binary erosion followed by a binary dilation. In the same way, we can perform a binary closing operation by eroding the dilated image. Erosion shortens object limbs, while dilation expands the object. The first step can be performed by selecting the menu option “***Black and white→Non-linear filtering→Morphology→Binary Erode***“ and specifying the thresholded image as the source and **<NewBuffer>** as the destination. We use a cross type structuring element, by entering 2 in the *structuring element* box. The intermediate eroded image is shown in Figure 5b.

The opened image can be obtained by performing a binary dilation operation on the eroded image. We select the menu option "***Black and white→Non-linear filtering→Morphology→Binary Dilate"*** and enter the eroded image as the source and **<NewBuffer>** as the destination. The final opened image is shown in Figure 5c.

The closed image can be obtained by performing a binary erosion on the dilated image. We select the menu option "***Black and white→Non-linear filtering→Morphology→Binary Erode***" and enter the dilated image as the source and **<NewBuffer>** as the destination. The final closed image is shown in Figure 5d.

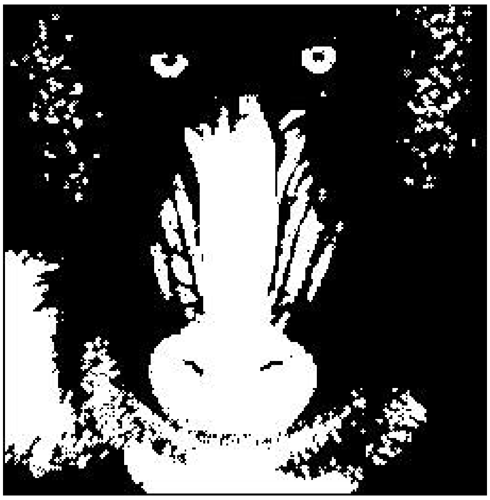


(a)



(b)

(c)



(d)

(e)

Figure 5: (a) Thresholded BABOON. (b) Result of erosion on Image 5a. (c) Result of dilation on Image 5a. (d) Final opened image. (e) Final closed image.

Section 7.6.2: Skeletons.

A well known morphological operation is skeletonization. It is used to extract the skeleton of an image, which is used as a means for shape description. EIKONA supports this operation for binary images. Let us suppose that the binary image RECT.RAW is loaded in BW buffer 0. We select the menu option "***Black and white→Non-linear filtering→Morphology→Skeleton***". We specify BW buffer 0 and **<NewBuffer>** as source and destination image respectively and 2 as the parameter. The original image and its skeleton are shown in Figure 6.

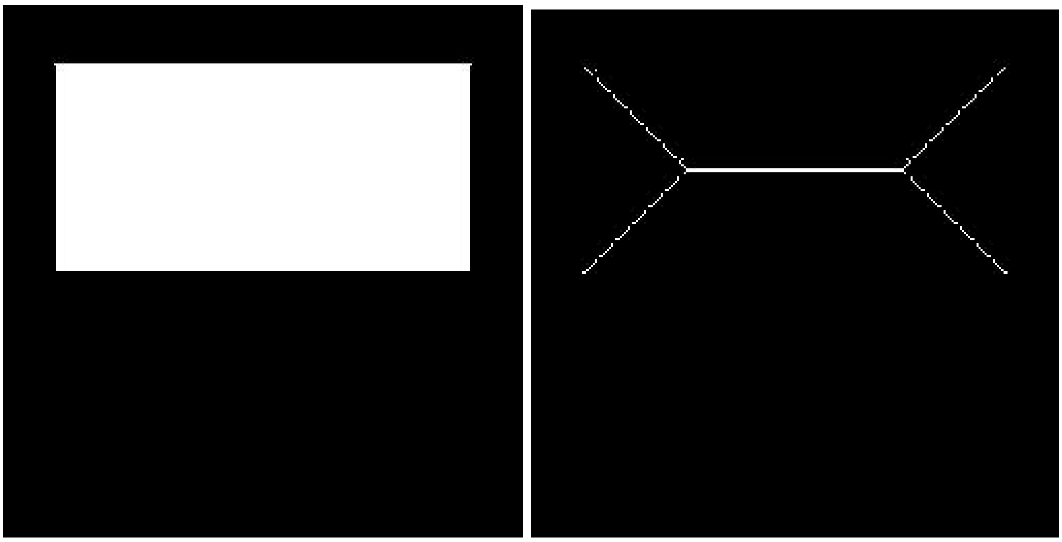


Figure 6: The binary image RECT and its skeleton.

Section 7.6.3: Top hat transformation.

Small details of a binary image can be extracted using the top hat transformation. We will use the binary image of Figure 7a, for the demonstration of the transformation. To perform the transformation, we select the menu option "***Black and white→Non-linear filtering→Morphology→Top Hat***". We specify the thresholded image and **<NewBuffer>** as source and destination respectively and 3,3 as the window dimensions. The result is depicted in Figure 7. It is obvious that this morphological transformation yields many small details of the source binary image.

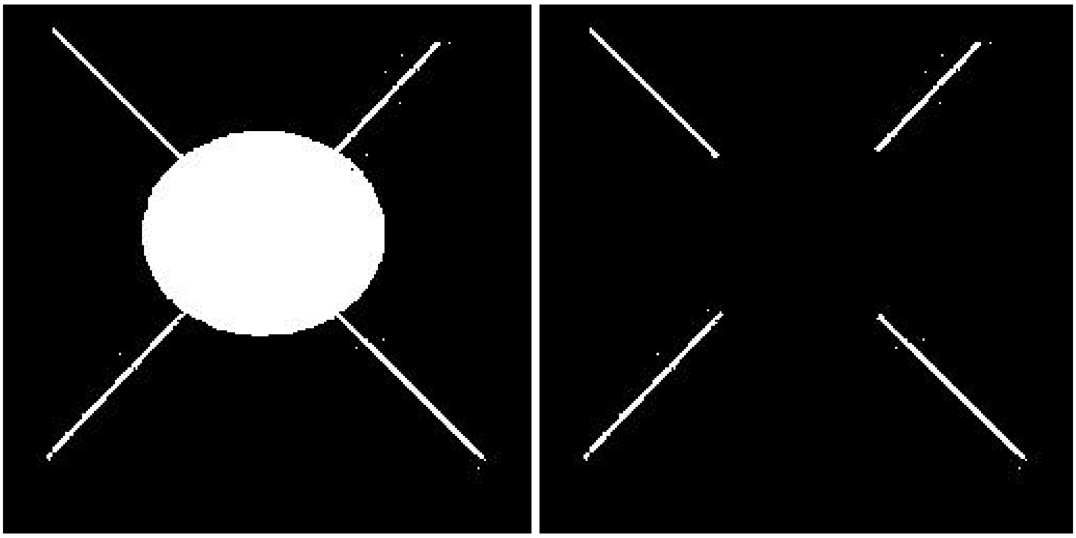


Figure 7: (a)Original image. (b) Result of the top hat transformation.