

Morphological Operators

Objective:

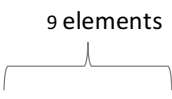
To implement the following morphological operators and test them on test images

Language used: Python

Running the code (mp02.py): (Requires *ccl.py*)

In a terminal, enter:

`./mp02.py [image name] [SE(3x3)] [operation(s)]`



Example usage:

`./mp02.py gun.bmp 111010111 D2CE1NB 50`

where,

- 'D2CE1B' → **D2** = 'Dilation' x 2, followed by, **C** = 'Closing', followed by, **E1** = 'Erosion x 1', followed by, **N** = De-noise, followed by, **B** 'Boundary Extraction';
- '50' → Size-filter threshold for de-noising

Description of morphological operators and their implementation:

1. Erosion:

- $A \ominus B$, where A is original image and B is some Structural Element (SE)
- In this problem, 'Erosion' is applied to white images on black background
- A SE is assumed to be any combination of a 3 x 3 element as shown in **Figure 1**
- Implementation:
 - a. Scan image from left to right, top to down
 - b. Only set the pixels that correspond to the center of the (3x3) SE to be 'white' (This will 'erode' the surrounding pixels and keep them 'black')
 - c. The output image is obtained once all the pixels have been scanned

2. Dilation:

- $A \oplus B$, where A is original image and B is some SE
- In this problem, 'Dilation' is applied to white images on black background
- A SE is assumed to be any combination of a 3 x 3 element as shown in Figure 1
- Implementation:
 - a. Scan image from left to right, top to down
 - b. Set the surrounding pixels in the actual image data to be 'white' if the corresponding surrounding pixel(s) in the SE is 'white'
 - c. The output image is obtained once all the pixels have been scanned

3. Opening:

- Defined as a combination of Erosion followed by Dilation
- $(A \ominus B) \oplus B$, where A is original image and B is some SE

4. Closing:

- Defined as a combination of Dilation followed by Erosion
- $(A \oplus B) \ominus B$, where A is original image and B is some SE

5. Boundary:

- Boundary extraction is carried out by first doing a 'Erosion' and then subtracting the eroded image from the original image
- $A - (A \ominus B)$, where A is original image and B is some SE
- In this problem, SE8 was used for all boundary extraction

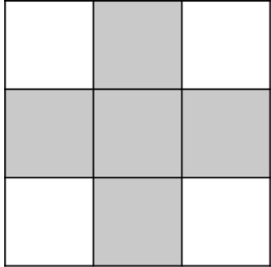
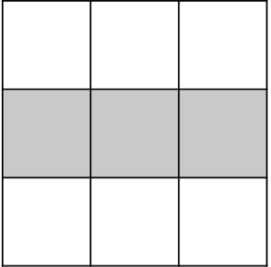
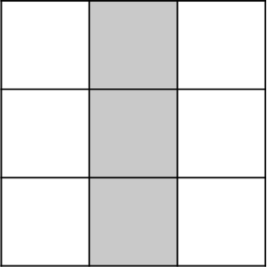
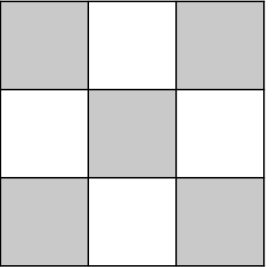
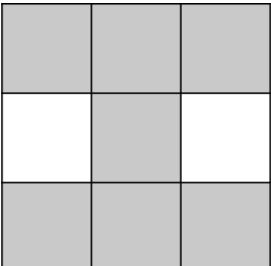
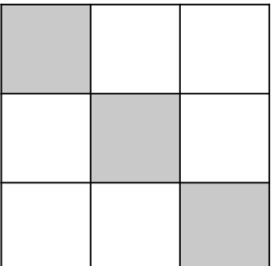
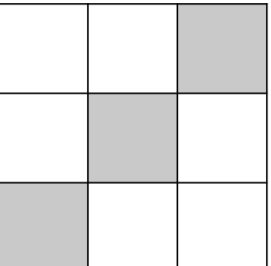
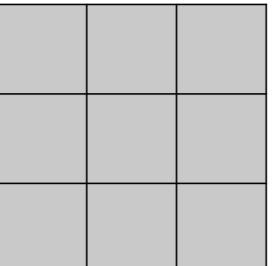

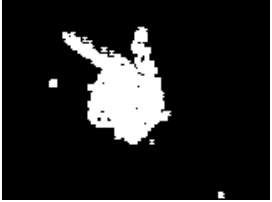
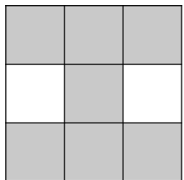


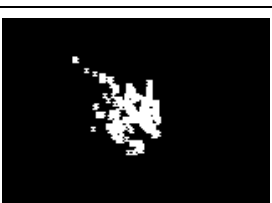


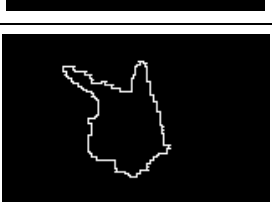



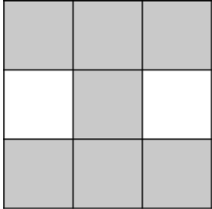
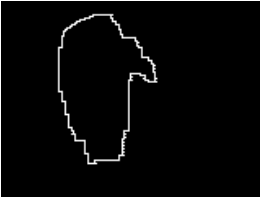
			
SE1	SE2	SE3	SE4
			
SE5	SE6	SE7	SE8

Figure 1: Different types of Structural Elements (SE) used for testing

Results Analysis:

The following table summarizes the results obtained using two test images:

Image No.	Input Image	Operation	Output Image	SE
1a (gun)		D1		SE5  (111010111)
1b (gun)		E1		
1c (gun)		C		
1d (gun)		O		
1e (gun)		D2CE1		
1f (gun)		D2CE1N 50		
1g (gun)		D2CE1NB* 50		

2a (palm)		D3CE1		SE5
2b (palm)		D3CE1N 50		
2c (palm)		D3CE1NB* 50		(111010111)

**In the above cases, boundary extraction was carried out using SE8*

All the input images produced the desired output images. Noise filtering was required for both the images to achieve the final images (1c and 2c).

After trial & error, the SE5 from Figure 1 was determined to be optimal for both images to achieve the desired boundary images.

Image 1 required 2 iterations of dilation, followed by closing, followed by 1 iteration of erosion, followed by de-noising using size filter and followed by boundary extraction to achieve the desired 'boundary' image.

Image 2 required 3 iterations of dilation, followed by closing, followed by 1 iteration of erosion, followed by de-noising using size filter and followed by boundary extraction to achieve the desired 'boundary' image.

Various other SE and combinations can be easily carried out by varying the input arguments when running the code in order to achieve different results.