EECS: 332 Intro to Computer Vision, MP #2

Digital Image Analysis, Due on 10/05/2017

Karan Shah

Introduction

Morphological operators are a collection of nonlinear operators related to the shape and morphology of the image. These operators rely primarily on the relative ordering of pixel values and not on the actual numerical values of the pixel. Therefore, it is more suited for binary images. A structuring element is used to probe the image at all locations and is compared to the actual image pixels after which a decision is made based on the match.

Algorithm to various Morphological Operators

1. Erosion: function img_output_erosion = Erosion (image_input, struct_element);

- This operation is used for erosion of an image
- The Structuring element starts from the first pixel of the input image, and checks its neighbors, if it contains all the elements of the SE (in their corresponding position)
- If this is true, then the first pixel is set to be 1. If not, it is set to 0
- This process is done for every pixel in the Input Image and the output image is determined based on this procedure
- This could be generalized as an "AND" operation between the pixels of the structuring element and Input Image. If all the pixels match, then the pixel from which the match was started is set to 1.

2. Dilation: function image_output_dilation = Dilation (image_input, struct_element);

- This operation is used for dilation of an image
- The structuring element starts from the first pixel of the input image and checks if there is a match between any of the input image and the structuring element, even if there is one match, then the first pixel in the output image is set to 1
- This process is done for every pixel in the input image and the output image is determined based on this procedure
- This could be generalized as an "OR" operation between the pixels of the structuring element and Input Image. If any of the pixels match, then the pixel from which the match was started is set to 1.

3. Opening: function image_output_open = Open (image_input, struct_element):

- In this operation, the structuring element is rolled along the inner boundary of the Input Image, in order to give a smoother image
- Here, there is an erosion operation performed between the input image and the structuring element
- Then, the result of the erosion is then used to perform a dilation operation with the SE to get the output image
- Therefore, it is a combination of an erosion and dilation (in that order)

4. Closing: function image_output = Close (image_input, struct_element);

- In this operation, the structuring elemen is rolled along the outer boundary of the Input Image, in order to get a slightly better image
- Firstly, there is a dilation operation performed between the input image and the structuring element

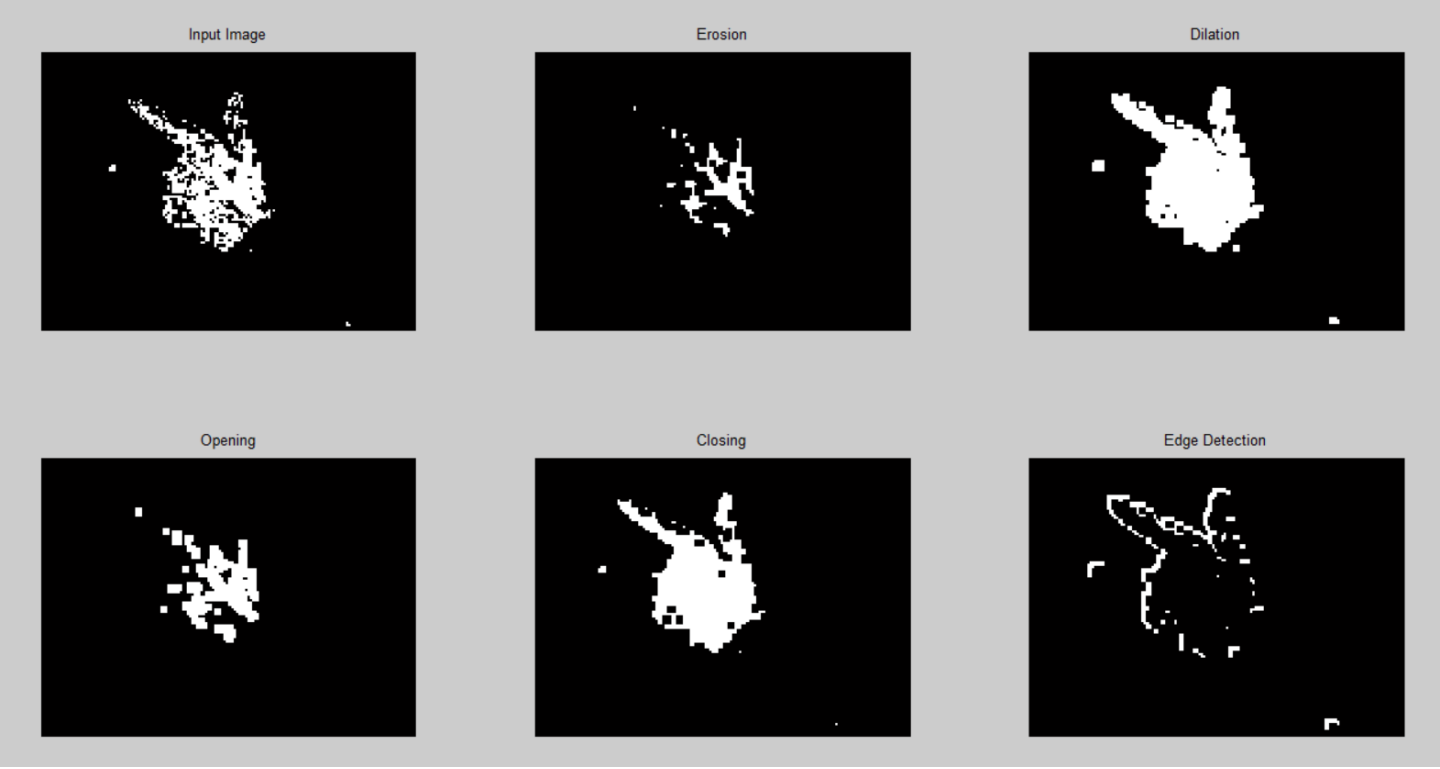
- Then the result of this dilation is then used to perform an erosion operation with the structuring element to get the output image
- Therefore, it is a combination of dilation and erosion (in that order)

5. Edge Detection: function image_output = Edge_Detection (image_input, struct_element);

- This operation is used to get the boundary of the input image
- For obtaining an accurate boundary of the image, there are many ways to perform this task
- One way is to take the input image and perform a dilation with the structuring element, then take the result of the dilation operation and perform another dilation operation with the structuring element.
- Then, take this result and subtract it from the input image to get an accurate image boundary
- This method gives an accurate boundary for most sizes of the structuring element

Result Analysis

Running this code on the image "gun.bmp", with a Structuring Element of the matrix [1 1 1; 1 1 1; 1 1 1], the following results are obtained



The Results can be analyzed as follows:

1. Fig 1. Shows the Input Image of the Gun

2. In Fig 2. It can clearly be seen how the Erosion operation has been performed and the image has much fewer pixels than the input image. The structuring element starts off from the first pixel and checks if all the pixels match between the two, if this happens, then the pixel in the output image is set to 1. Therefore, there are much fewer pixels in the output image. Thereby, an Erosion operation has been performed.

3. In Fig 3. It can clearly be seen how the Dilation operation has been performed and the image has much is much more dilated than the input image. The structuring element starts from the first pixel and checks if any of the pixels match between the two, if this happens, then the pixel in the output image is set to 1. Therefore, there are many more pixels in the output image. Thereby, a Dilation operation has been performed.

4. In Fig 4. The opening operation is being performed, it can clearly be seen how there are no pixels outside the boundary of the image, the structuring element was just rolled inside the boundary of the image. For obtaining image, there was first an Erosion operation performed, where the pixels reduce, and then a dilation operation is performed, to get a smoother image. Thereby, an Opening operation is performed.

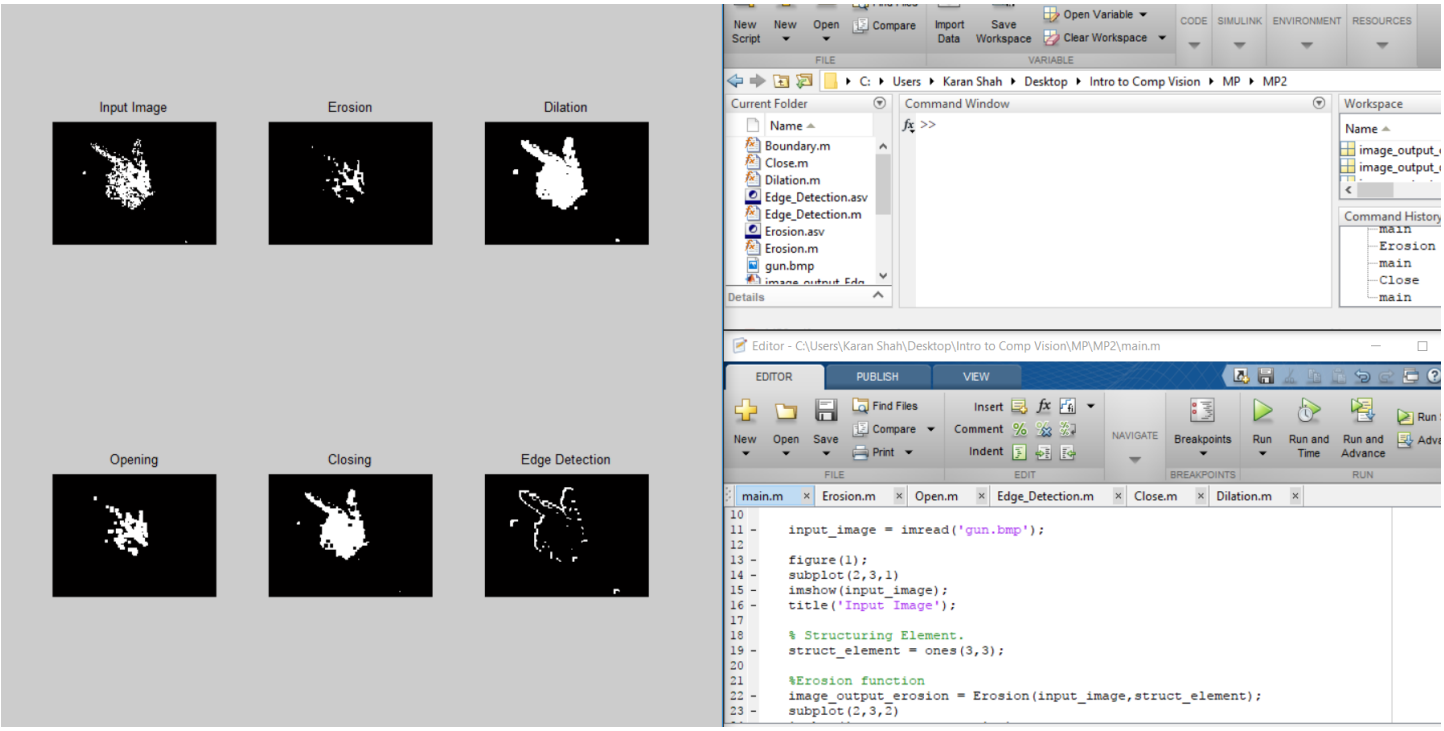
5. In Fig 5. The closing operation is being performed and it can be seen how there is a dilation operation being performed first and then an erosion operation with the structuring element. For obtaining the image, there was first a Dilation operation performed, where the image has more pixels, and then an Erosion operation is performed to get a better image. Thereby, a Closing Operation has been performed.

6. Lastly, in Fig 6, the boundary of the gun is clearly being shown which is a result of performing two dilations and one subtraction.

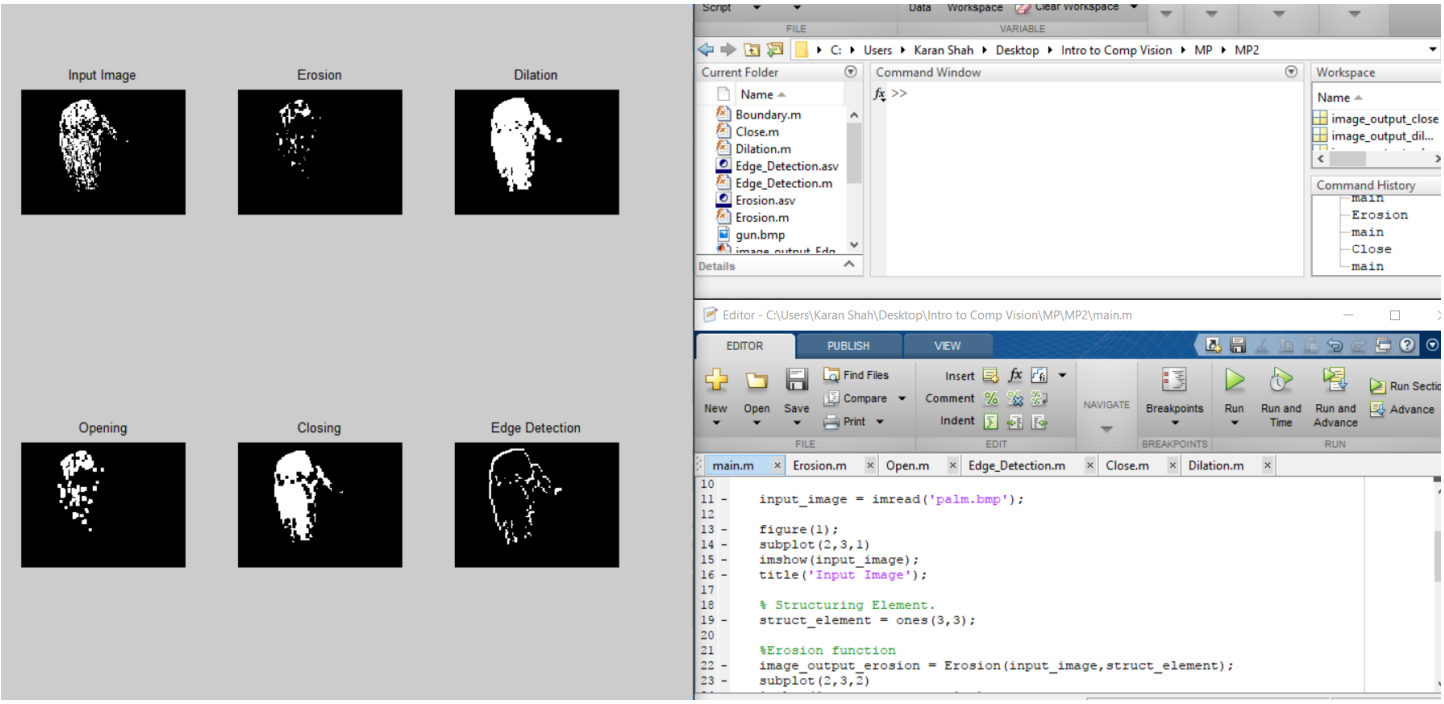
MP#2 Results for test images

Results:

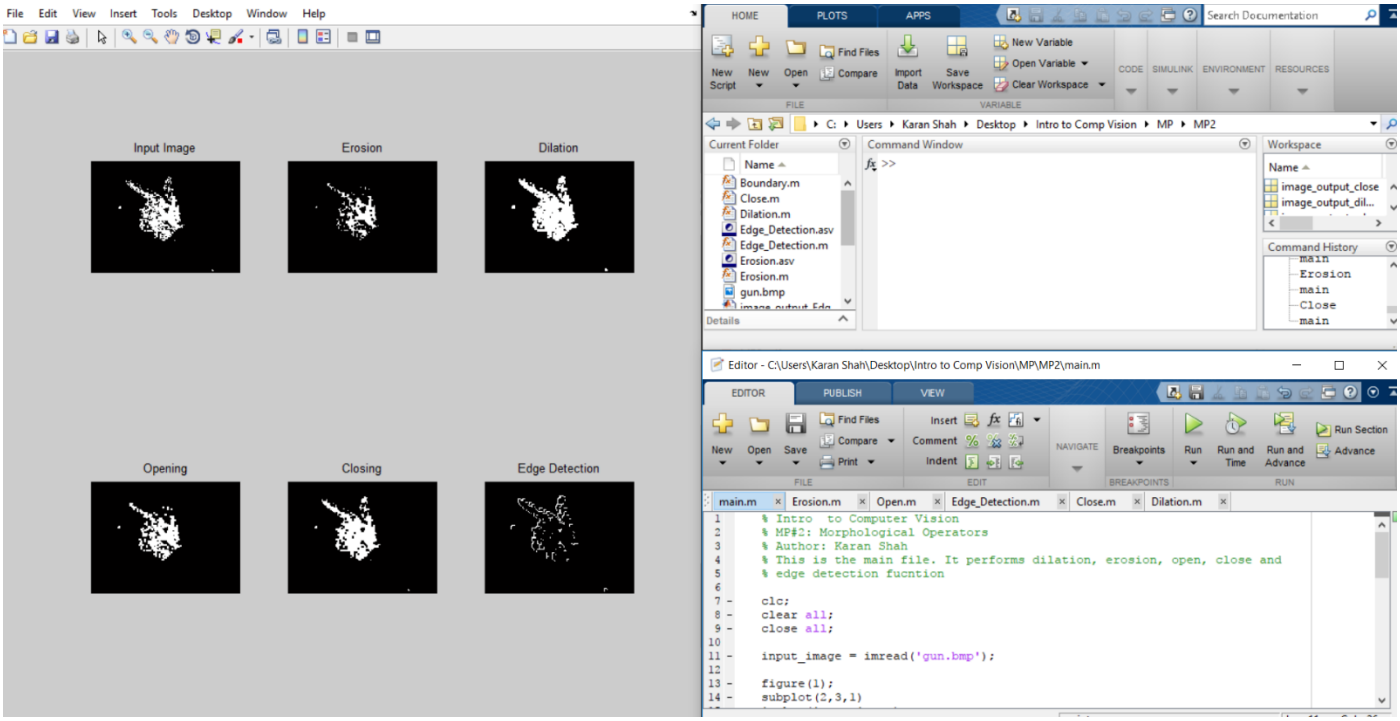
1. Gun.bmp



2. Palm.bmp



3. Trying different Structure Elements
   - Using Gun.bmp with a SE of [1 1; 1 1]



- Using palm.bmp with a SE of [1 1; 1 1]