IMAGE PROCESSING LABORATORY | SPRING '15

ASSIGNMENT - 5

GROUP-7

ANKIT BANSAL | 12EC35024
AMRUT RAJKARNE | 12EC35017

ASSIGNMENT – 4
MORPHOLOGICAL OPERATIONS

Aim:

Write C++/Image-J modular functions to perform the following operations on the given test image, ricegrains_mono.bmp. All functions must support **binary images** –

Make separate functions for erosion, dilation, opening, and closing of binary images

a. **ErodeBinary**, **DilateBinary**
Input: Binary image, structuring element
Output: Eroded/dilated image

b. **OpenBinary**, **CloseBinary**
Input: Binary image, structuring element
Output: Opened/closed image

Use structuring elements:

1	1

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

and 9 × 9, 15 × 15 kernels of grayvalue = 1 (reference point – centre pixel).

Theory:

The word **Morphology** denotes a branch of biology that deals with the form and structure of animals and plants. Here, we use the same word in the context of Mathematical Morphology, which means as a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons etc. We use Morphology for shape analysis & shape study. Mathematical Morphology is used to extract image components that are useful in the representation and description of region shape.

Mathematical morphology involves a convolution-like process using various shaped kernels, called structuring elements. Every Operation has two elements are present – Input Image (almost Binary) and Structuring element. The operation's results depend upon the structuring element that is chosen. The structuring elements are mostly symmetric: squares, rectangles, and circles. Most common morphological operations are – Dilation and Erosion. The operations can be applied iteratively in selected order to effect a powerful process - Opening and Closing.

- The most basic morphological operations are two: Erosion and Dilation. They have a wide array of uses, i.e. :
    - Removing noise
    - Isolation of individual elements and joining disparate elements in an image.
    - Finding of intensity bumps or holes in an image
- We will explain dilation and erosion briefly, using the following image as an example:

Dilation:

- This operations consists of convoluting an image A with some kernel (B), which can have any shape or size, usually a square or circle.
- The kernel B has a defined *anchor point*, usually being the center of the kernel.
- As the kernel B is scanned over the image, we compute the maximal pixel value overlapped by B and replace the image pixel in the anchor point position with that maximal value. As you can deduce, this maximizing operation causes bright regions within an image to "grow" (therefore the name *dilation*). Take as an example the image above. Applying dilation we can get:

The background (bright) dilates around the black regions of the letter.

Erosion:

- This operation is the sister of dilation. What this does is to compute a local minimum over the area of the kernel.
- As the kernel B is scanned over the image, we compute the minimal pixel value overlapped by B and replace the image pixel under the anchor point with that minimal value.
- Analogously to the example for dilation, we can apply the erosion operator to the original image (shown above). You can see in the result below that the bright areas of the image (the background, apparently), get thinner, whereas the dark zones (the “writing”) gets bigger.

Opening:

- It is obtained by the erosion of an image followed by a dilation.
- Useful for removing small objects (it is assumed that the objects are bright on a dark foreground)
- For instance, check out the example below. The image at the left is the original and the image at the right is the result after applying the opening transformation. We can observe that the small spaces in the corners of the letter tend to disappear.

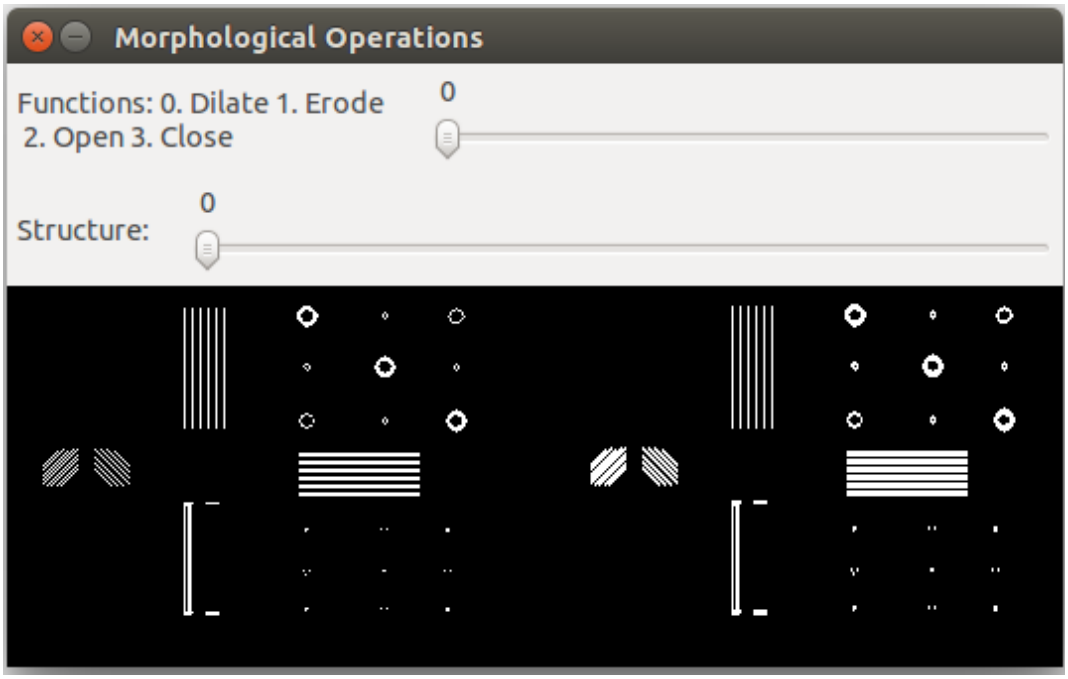
Closing:

- It is obtained by the dilation of an image followed by an erosion.
- Useful to remove small holes (dark regions).

Results:

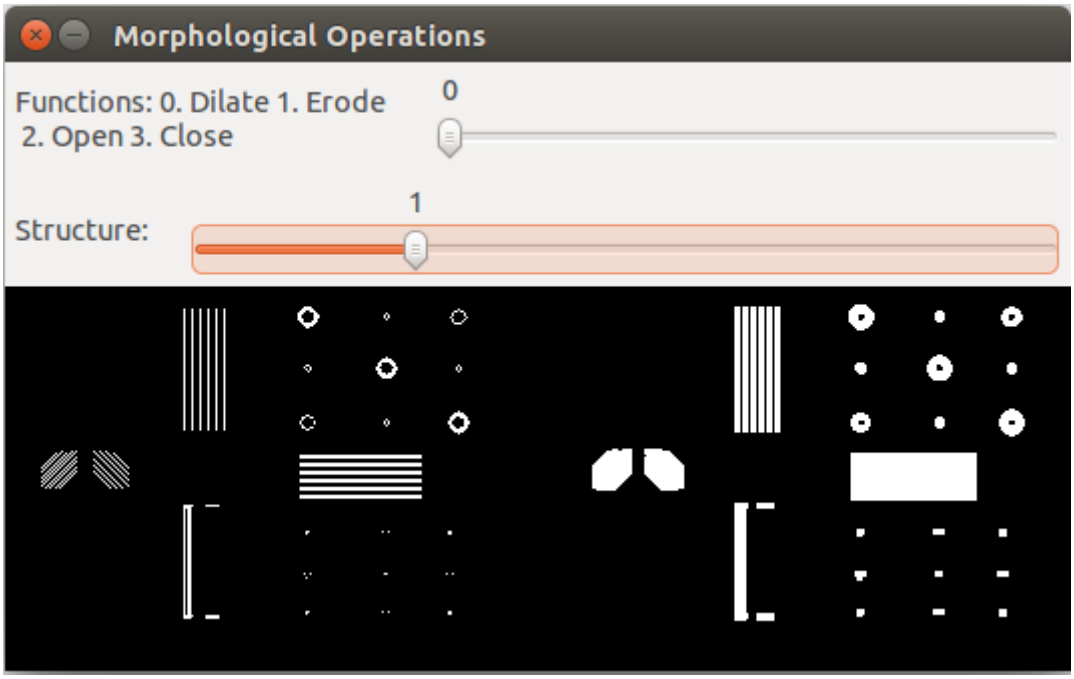
- **Dilation** –

Structuring Element: 1 1



Structuring Element:

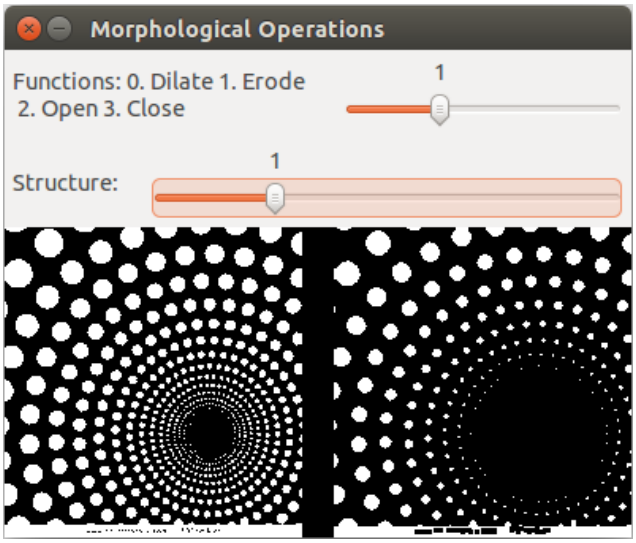
1	1	1
1	1	1
1	1	1



- **Erosion** –

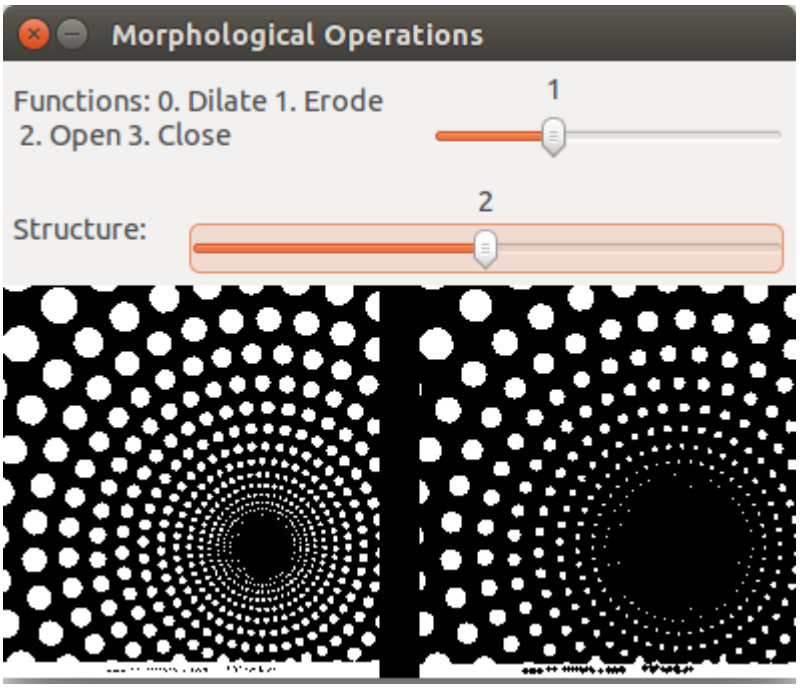
Structuring Element:

1	1	1
1	1	1
1	1	1



Structuring Element:

0	1	0
1	1	1
0	1	0



Similarly the output results for the other morphological operations are documented in the **'Result Images'** folder.

The different Structuring Elements correponding to the Structure Code of the Trackbar –

- Structure 1 -

1	1

- Structure 2 - 3X3 kernel

1	1	1
1	1	1
1	1	1

- Structure 3 – 3X3 Cross kernel

0	1	0
1	1	1
0	1	0

- Structure 4 – 9X9 kernel
- Structure 5 – 15X15 kernel